FlowVision

User's guide

Version 3.13.01

Contents

1 Introduction

The *FlowVision* software is intended for numerical simulation of three-dimensional laminar and turbulent, steady and unsteady flows of liquids and gases. The software is based on the finite volume method, high precision difference schemes, high-performance numerical methods and reliable mathematical models of physical processes. The numerous models allow simulation of complex flow processes with flow swirling, movement of free/contact surfaces, shock waves, conjugate heat transfer, combustion, etc. A Cartesian, locally-adaptive computational grid is used by *FlowVision*. Local dynamic adaptation of the initial grid is performed according to user-defined criteria. The initial grid consists of rectangular cells. Near the border of the computational domain, Boolean subtraction of non-computed volumes from rectangular cells is performed, resulting in polygonal cells of arbitrary shape. No simplification of boundary cells is performed. Grid generation is completely automated.

The *FlowVision* software runs on mixed architecture computers, combining inter-node MPI parallelization with intra-node thread parallelization, as on a shared memory computer. Using mixed parallelization allows high-quality scaling of the software when running on a large numbers of processors.

FlowVision permits flow simulation near moving and deforming bodies in conjunction with the *Abaqus* finite-element software.

Being connected to the *IOSO* optimizing software, *FlowVision* solves object shape optimization problems.

Using state-of-the-art data visualization and processing implemented in *FlowVision*, it is possible to perform quick and effective analysis of computation results and receive the required numeric data.

The *FlowVision* capabilities, based on modern computing technologies and effective customer support, enable *FlowVision* to compete successfully with other fluid- and gas-flow simulation software products.

Date and time of the document's creation: 12/15/2022, 4:46 PM.

1.1 What is FlowVision?

Purpose

The *FlowVision* software is a tool for simulating the flow of fluid and/or gas in technical devices or in natural objects with subsequent analysis of simulation results.

The software is primarily intended for simulation of flows with the following features:

- complex shape of the flow-region boundaries, which allow simulation of real technical devices
- turbulence
- arbitrary flow speeds (from incompressible fluid flows to supersonic ones)
- convection and radiation heat exchange, thermal conductivity and conjugate heat transfer
- non-Newtonian flow medium rheology
- presence of moving bodies (e. g., parts of complex construction) in the computational domain
- presence of liquid/gas or liquid/liquid phase interface surfaces
- diffusion, chemical reactions between components, combustion

Brief technical description

The *FlowVision* software is equipped with various project building tools, including:

- tools for the import of geometric objects (which are boundaries of the computational domain) from CAD software products
- tools for creation elementary (*standard*) geometric objects
- tools for automatic generation of a computational grid, taking into account the shapes of objects located in the computational domain
- tools, which specify boundary and initial conditions for simulation of flow and heat exchange in the computational domain

The computational block of the software provides quantitative solutions to systems of equations describing movement of fluid and/or gas in the computational domain, including:

- mass, moment and energy conservation equations
- state equations
- turbulence model equations
- special model equations (describing the motion of a fluid/gas phase interface surface, combustion, radiation heat transfer, flow in narrow gaps)

The system of equations is solved on a Cartesian grid that is automatically locally downscaled. Grid refinement may be concentrated in areas of high gradients or of complex geometric shape. Grid cells intersecting computational domain boundaries and computational subregions are clipped by boundary surfaces.

FlowVision implements a flow computation technology with bodies moving in the computational domain relative to immobile bodies.

The computational block of *FlowVision* works on computers of various configurations: personal computers, local area networks or clusters. Parallelization of computations is performed automatically and provides effective use of multiprocessor hardware.

FlowVision includes various utilities for analyzing the simulation results:

- means of control and express analysis of intermediate computation results
- means of accumulating the integral characteristics of various processes
- means of analyzing and visualizing computation results, including widespread means for displaying scalar and vector variables in the computational domain volume and on its boundary

Applicability

The *FlowVision* software is mainly used for simulation and analyzing flow processes in technical devices of various purposes. Flow simulation is usually performed to solve two types of tasks:

- evaluation of project solutions
- computational support for technical device design

Also *FlowVision* can be a kind of laboratory test bench for students during aerodynamics, gas dynamics and heat exchange courses.

Evaluation of project solutions is performed by modeling flows in a technical device based on a project proposed by a designing organization. Comparing the parameters of a simulated flow with the expected values confirms the correctness of the proposed project or provides information on the need for project modification.

The use of *FlowVision* in design tasks frequently goes deeper than the evaluation of single solutions. In many cases when the design methods used are not able to guarantee the necessary parameters of the device designed, a refinement stage in technical object development is necessary. *FlowVision* permits multivariate searching of the required technical solution before it is implemented in hardware.

Optimization problems can also be solved in automated mode, using advanced optimization algorithms. For this purpose, interaction with the *IOSO* software is implemented (see the Optimization section).

1.2 Use of the documentation

Locations in hierarchy structures

In the hierarchy structures (project tree, parameter groups in the **Properties** window, multi-level menus), the levels are separated by the "**>**" symbol. Upper levels may be omitted if they are known from the context.

Examples:

- **View > Toolbars > Network** – the **Network** command in the **Toolbars** submenu, which opens from the **Main Menu**'s submenu **View**
- **Preprocessor > Region > Objects > Cone/Cylinder #0** - a standard geometric object of **Cone/Cylinder** type, created in the **Pre-Postprocessor** project tree, in the **Preprocessor** tab, in sub-folder **Objects** of root folder **Region** and possessing the standard name **Cone/Cylinder #0**
- **Location > Axis X > Z** - parameter in the geometric object's **Properties** window, the direction of the **X** axis of its local coordinate system along the **Z** axis of the absolute coordinate system

Values of parameters

For conditions applied to values of parameters, notation **Name = Value** or **Name ≠ Value** is used.

For assigning a value to a parameter, notation **Name = Value** is used.

Alternatives for values of parameters are separated with **|** symbol(s) or they are listed in a bullet list.

Default values are underlined or noted explicitly.

Examples:

- Parameter **Color value** is available when **Color from object = No**.
- The total number and the available number of licenses that grant use of **Moving bodies** with enabled update (**Update > Type ≠ Disabled**)
- Set **Variable > Variable = Temperature**.
- Possible options are: **Yes** | **No**.
- **Appearance > Mode = Lines | Fill | Lines and fill**

Fonts

Names of files and folders in the operating system are `indicated by font`.

Example: `ReleaseNotes_eng.txt`

Italic is used to identify logical selections and input of definitions. Also italic is used for names of companies, products and technologies.

Interface elements and names of objects and parameters are marked in **bold**. The names may be changed in the description, the project tree, parameters, etc. according to grammar rules.

Example: "Assign **Binder condition #0** to **Binder #0**. To do so, select command **Add/Remove** in the context menu of sub-folder **Boundary links > Binder conditions > Binder condition #0 > Binders**, select **Binder #0** on the **Non-selected** panel in the dialog that opens, move it to the **Selected** panel and click **OK**."

Commentaries in code examples can be indicated by gray font color.

Example:

```
# command that saves data after calculation stops
echo "SS_PRJDATASAVE" >> $PROJECT_DIR/.commandfile-auto
# command that finishes Solver and unloads it from memory
echo "SS_SHUTDOWNSOLVER" >> $PROJECT_DIR/.commandfile-auto
```

Links

Hyperlinks are used for navigation to other sections of the documentation.

Links to numbers of sections will be replaced by hyperlinks in future versions of the documentation. Now these links are not applicable.

Tables

Tables with descriptions:

Column title	Column title	Column title
Element	Description for this line	Description for this line
Element	Common description for several lines	Common description for several lines
Element		

Tables describing some procedures or user's actions:

Step	Actions
1	Description of Step 1.
2	Description of Step 2.
3	Description of Step 3.

Text box notes

Aspects that require special attention are presented as text boxes.

Example:

Text

Greek/Latin character replacement

Greek characters may be replaced with Latin ones in the program interface.

For example, the Greek character epsilon (ε) is used in names of certain turbulence models. It may be replaced with the character "**e**" in the program user interface in order to avoid possible encoding issues.

Interface elements (selection options in lists, parameter values, field names) are mentioned in the documentation the same way as they are displayed in the interface. Greek characters may be used in descriptions of models themselves, variables and physical values.

Notation for numbers

The traditional notation for powers of 10 is used in the documentation (for example 10^{-6}=1/1000000).

The *FlowVision* interface uses the following indication: the power of 10 is indicated by the character "**e**" followed by an optional sign and three digits. Thus, **1e-010** in a program's numeric field indicates 10^{-10}.

The integral and the fractional part of decimal fractions are separated by a period ("**.**").

Abbreviations

ACS - absolute coordinate system
BC - boundary condition
CAD - computer added design
CFD - computational fluid dynamics
CS - coordinate system
FESP -finite-element software package
FSI - fluid-structure interaction
GUI - graphical user interface
LCS - local coordinate system
BL - boundary layer
OS - operating system
SGA - side gradient approximation

Translation

Many pages of the documentation have been automatically translated to English. Some fragments are highlighted with blue background which can designate human translation.

Other

- For the purpose of brevity, the project files are usually identified by their extension only.

Example: `glo` file

- Dimensions of quantities are indicated in square brackets []. Dimension of an arbitrary variable f is indicated as [f].

Example: [ms] - milliseconds

- The program's documentation is regularly improved, but might omit descriptions of some program's functionality, interface settings, recommendations of use, and other descriptions. Also the documentation might have typos or errors.
- Sections "What's new" are not edited even if their contents become not actual.
- The last minute updates that were not included into the last release are available online in the web version of the documentation.

1.3 Documentation under construction

Updates of the documentation are available on the **flowvisioncfd.com/en/** web site (in the section **Support and Blog > Documentation**, https://flowvisioncfd.com/en/support-page-en/blog-en).

You can contact the technical support service for any questions about this functionality.

1.4 Recommendations for learning the program

The following steps are recommended to get acquainted with the program at the initial learning stage:

1. Study the Quick Start
2. Study the basic examples in the guidebook supplied with the software
3. Study the examples related to your fields of interest from the second part of the guidebook
4. Familiarize yourself with the various settings described in the Pre-Postprocessor Interface Reference section
5. It is strongly recommended to have an understanding of process equations and boundary values calculation equations. See Theory.

2 What's new in FlowVision

Software package *FlowVision* is constantly improving and new versions are regularly issued.

This section gives a brief overview of the most significant innovations to versions of the *FlowVision* software.

Note: Some innovations require changes of project's settings if the project was prepared in an earlier version of *FlowVision.*

2.1 What's new in FlowVision 3.13.01

Important changes

- The *FlowVision*'s distribution kit for *Linux* now includes **Pre-Postprocessor**, so all *FlowVision* modules can now operate under *Linux*.
- Functionality of the boundary layer grid has been totally renewed. The boundary layer grid is now totally curvilinear, able to fit to the main grid, and adaptable. Computation of the previously calculated projects that include old boundary layer grid cannot be resumed because these projects require rebuilding their boundary layer grids.
- System requirements changed, and some very old operating systems are not supported now:
  - *Linux* distribution kits with version *glibc* below *2.12* are not supported anymore.
  - *Windows server 2003* and older versions of this operating system are not supported anymore. *Windows Vista* and older operating systems are not supported.
  - Requirements to version of *OpenGL* were increased. Minimal requirements are now: *OpenGL* version *3*.
- Update of the **License Manager**. To make the changes take effect, you have to install the new version of the **License Manager**. This update includes the following features:
  - **Pre-Postprocessor** displays a warning if the license expires soon.
  - More detailed information in the dialog box with information about the license.

New

FlowVision 3.13.01 contains new features; the most significant of them are listed below.

What's new in physical models

- The icing model supports connected boundary conditions.
- **Internal characteristics** now include a new data line, **Film flow time**, for the icing model.
- The icing model takes into account splashing caused by collisions of supercooled drops and a wall and returning the satellite drops into the two-phase flow.
- The film motion model takes into account shedding and partial transfer of the film by the flow.
- Now it is possible to simulate motion of the dispersed phase through the phase interface surface between two continuous phases. Example of this: falling sand from air on water and further falling the send within the water.
- Now it is possible to specify source of a phase on the phase interface surface.

New features in Pre-Postprocessor

- Now it is possible to import reactions and substances from ChemKin.
- In settings of **Models** it is possible to select local variables, for which averaged values in specified time interval will be additionally calculated. This allows you to visualize fields of variables averaged by time.

New features in Postprocessor

- Now it is possible to save **Views** from the **View** window and switch between them quickly. This allows you to change visualization settings of **Layers** quickly and conveniently for post-processing a large number of similar projects with a large number of **Layers**.
- **Internal characteristics** now includes new data lines, which include numbers of cells and time steps.

Improvements of the user interface

- Now you don't have to click the **Apply** button to apply changes in the **Properties** window. Your changes are saved automatically when you navigate to other elements of the project tree. The **Apply** button remains in the user interface; it allows quick changing in settings of **Layers** when **Pre-Postprocessor** operates being connected to **Solver**.
- The changed parameters in the **Properties** window are visually highlighted.
- In the project tree, in the **Postprocessor** tab, now it is possible to select multiple elements to carry out group operations with them.
- The dialog box for creating **Layers** and **Characteristics** has been improved. The new window allows you to set the basic parameters quickly. Also you can specify there the name of the **Layer** or **Characteristics** or keep the default name (based on the type and properties of the element being created).
- Ability to create a **Layer** or **Characteristic** at once on multiple objects just selecting the appropriate mode in the dialog box.
- The context menus now might include icons for better navigation when you select commands there.
- **Pre-Postprocessor** now includes functionality for tuning **Palettes** and new ready-made **Palettes** were added (see Parameters for defining a palette).

- User interface for connecting to Solver-Agent was improved in all program modules. The key improvements are:
  - If you generally run computations on the same computer, you can tune automatic background connection to **Solver-Agent** at start of **Pre-Postprocessor**.
  - You can save setting of connection to **Solver-Agent**. This is useful when you work with large number of computers via one workstation.
- User interface for starting **Solver** from **Pre-Postprocessor** has been improved. *FlowVision 3.13.01* requires less clicks for frequent operations. Starting **Solver** and connecting **Pre-Postprocessor** to it are done by clicking one button only.
- Now it is possible to look for elements in the project tree. This simplifies your work with complex assemblies.
- Now it is possible to assign hot keys. Number of commands that can be called by hot keys has substantially increased.
- **Pre-Postprocessor** doesn't include the **Viewport** tab anymore. Its contents has been moved to the **Postprocessor** tab (to the **Lighting** folder and to properties of the **3D-scene** root folder).

Other changes

- In **Terminal** it is possible now to decimate data records that were saved with history. This feature is useful when you work with supercomputers as it allows you to decimate data records without downloading a large client project from **Solver** on **Pre-Postprocessor**.
- It is possible to use implicit linking for FSI tasks and *Abaqus* tasks. This approach substantially improves stability of the FSI solutions in cases when liquid has significant influence on the surface, which is simulated in the FSI software.
- A new dialog box with settings of the FSI computation has been added (used when external connections are created).

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.2 What's new in FlowVision 3.12.05

Release *FlowVision 3.12.5* is a hot fix.

Important changes

- In the **Formula editor** a new operator `prev` (`prevvec`) has been added, which allows you to receive value of a scalar or vector global user variable from the previous time step. Use of this operator is shown in the updated exercise *"Time-varying flow in a tube"* in the *Tutorial: Examples of typical tasks* document, which is included in the distributive pack of *FlowVision*.
- The **Initial grid editor** now supports one-dimensional (1D) problem settings. See description of the new user interface in the section Element «Initial grid».
- For **Modifiers** (except **Moving bodies**), which are set on finite-volume **Objects**, a new parameter **Scope** was added that allows you to specify more precisely the area where the Modifier acts near surface of the **Object**.

Improvements

Calculations of radiation flux on the boundary condition **Connected** are more accurate now.

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.3 What's new in FlowVision 3.12.04

Important changes

The release *FlowVision 3.12.04* is a hot fix and it fixes some errors in release *3.12.03*:

- In the **Formula editor**, functions `root`, `sec`, `cosec`, `arcsec`, and `arccsc` operated incorrectly.
- Velocities were calculated incorrectly on exchange surfaces in FSI simulations if some initial rotation angle was set in settings of the initial position of the **Moving body**.

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.4 What's new in FlowVision 3.12.03

Important changes

The release *FlowVision 3.12.03* is a hot fix and it fixes some errors in release *3.12.02* related to user Substance Databases:

- When properties of a **Substance** were set in **Pre-Postprocessor** by downloading from a user Substance Database, dependency of specific heat C_p on temperature was not downloaded. Now *FlowVision 3.12.03* download data from user Substance Databases correctly.
- **Substance Database Editor** now uses the *UTF-8* character set. So, when old substance databases are opened, there are issues with displaying Cyrillic text. To work in new **Substance Database Editor** with old substance databases that include Cyrillic text, you can apply workarounds:
    - remove translation file **`strings.lst`** from the user substance database
    - *or* convert the file **`strings.lst`** to the *UTF-8* format.

2.5 What's new in FlowVision 3.12.02

Here the most important changes and new possibilities, added to the version *FlowVision 3.12.02*, are listed.

Important changes

Before installing and using *FlowVision 3.12.02* ensure that you have read this section.

- The **libFvMPI** library changed. If you compiled your own version of this library to interact in *Linux* with a specific version of MPI, you have to make a compilation again using new source codes from the distribution kit. See Compilation a library for interconnection with various MPI implementations.

New

FlowVision 3.12.02 contains new features; the most significant of them are listed below.

New physical models

- The **KEQ** quadratic turbulence model was replaced with the **KENL** nonlinear turbulence model.
- New boundary conditions **Radiation flux density** and **Point Source** were added to the Discrete-ordinates method radiation model.
- In the **Motion** physical process of a **Particles** phase it is possible now to set a user-defined force that acts on particles (it is set by parameters D and F, see Theory > Physical processes > Processes in the presence of dispersed medium > Parameters > Process 'Motion').
- In the **Mass transfer** physical process of a **Particles** phase it is possible now to set a user-defined source of mass (it is set by parameters D and F, see Theory > Physical processes > Processes in the presence of dispersed medium > Parameters > Process 'Mass transfer').
- In the general mass transfer model of a **Particles** phase now you can also set a model of transforming the particles (**Model for particles = Variable diameter | Constant diameter**). This setting was only available for the **Coal** mass transfer model before. Old values of the **Model for particles** parameter were renamed (**Shrinking particle** was renamed as **Variable diameter**, **Shrinking core** was renamed as **Constant diameter**).
- Ablation of thermal-protective coating caused by thermochemical destroying the matrix of porous structure (the VOF method is used).
- An ablation model **Carcass**, was implemented, which simulates destruction of porous thermal-protective system (TPS) and mass loss from its surface. Gases, which are generated during the pyrolysis process, move through pores and blown into the boundary layer near the surface of the TPS.
- Specific heat C_p, molar mass, and enthalpy of formation h_0 can be set as values that depend on the **Oxid. excess factor rec.** variable (α).

New possibilities of preprocessing

- **Pre-Postprocessor** now includes a convenient setting **2D direction**, which makes the computational grid be 2D. When **2D direction** is set, adaptation of the computational grid is also done in 2D; this substantially saves computational resources.
- Now it is possible not to take into account the pressure gradient in calculation of the explicit time step (this is set by the **Pressure gradient** parameter in properties of the **Time step** element).
- A new setting for computation of gradient of surface tension (**Multiphase C > Surf. Sigma gradient** in advanced settings of Solver).
- Now it is possible to disable mandatory adaptation to the same level within gaps (this is set by the **Adapt through gap** parameter in advanced settings of Solver).

New possibilities of postprocessing

- **Viewer** now displays layers **Plot along line**, **Plot along curve**, **Plot along ellipse**.
- In **Characteristics** over a surface, when the **Extended data** setting is turned on, additional scalar and vector integral values will be calculated.
- Rendering of the **Solids** layer in carried out more quickly now in situations with large number of **Subregions**.
- The **Nodal loadings** layer now provides coloring its vectors according to their length and visualizes the loads in the coordinate system of the **Moving body**.

Improvements of the user interface

- In **Terminal**, in the **Projects** menu, a new command has been added, **Run solver, load project and start**. This essentially saves your time when you start a project from **Terminal**.

- When an **Initial condition** is selected in the project tree, the **Object**, on which the **Initial condition** is set, it highlighted in the **View** window.
- In the **Plot** tab in the **Monitor** window in **Pre-Postprocessor** and in the **Plot** tab in **Viewer** you can now copy names and current values of plots using **Ctrl+C** hot keys.

Other changes

- Now it is possible to register a **Solver-Agent** user without a license name. This is useful when you wish to create a user to analyze only results of the computation.
- Better compatibility of files in the *VTK* format and programs *Paraview* and *Ensight*.
- It is possible to export loadings for a FSI project based on saved results of the computation (in previous versions you had to repeat the computation).
- In FSI simulations with mapping, accuracy of mapping now is calculated automatically; you don't have to specify this accuracy manually.
- **Configurator** can now enable/disable use of the *TCP/IP v6* protocol for network communications between modules.
- Now it is possible to select all regions for data export in the CGNS format.
- The **Very viscous flow** parameter was relocated to properties of the element **Simulation controls > Limiters > Limiters for calculation > Phase limiters > Phase #N**.
- Some changes in the algebraic solver to improve convergence so settings of the algebraic solver were changed.

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.6 What's new in FlowVision 3.12.01

Here the most important changes and new possibilities, added to the version *FlowVision 3.12.01*, are listed.

Important changes

Before installing and using *FlowVision 3.12.01* ensure that you have read this section.

- 32-bit versions of *Windows* are not supported anymore:
  - So new versions of main *FlowVision* modules and **License manager** by default are now installed on 64-bit *Windows* into the folder `Program Files`, not into `Program Files (x86)`.
  - *Note:* before installing the new **License manager** version *3.12.01*, you have to uninstall its previous version.
- *MPICH2* for *Windows* is not supported anymore.
- In the user interface of **Pre-Postprocessor**, instead of **Dispersed Phase**, you can create a **Phase** of either **Particles** type or **Carcass** type. The first one is used to simulate particles and the second to simulate flows in porous media.

New

FlowVision 3.12.01 contains new features; the most significant of them are listed below.

New physical models

- Coal combustion model. Accordingly, **Coal** mass transfer models were added for **Continuous Phases** and **Phases Particles**.
- Crystallization model for solid substance on a surface (for simulating of of aircraft icing).
- A new radiation model, **Discrete-ordinates method**, allows simulating radiation with taking spectra and shadows into account.
- The program's interface includes now the new type of dispersed phases, **Carcass**, which is used to simulate porous media.
- The **Darcy model** for stable and quick simulating of flow's motion through porous medium (**Carcass**) with high hydraulic resistance.
- Simple cavitation model (beta version).
- Simulating of multiple size groups of dispersed particles. Prepared sets of size groups can be tuned in **Size spectra**; this allows the program to simulate motion of particles of different-sized particles in one **Phase** of the **Particles** type.
- New power law **PL2** for simulating non-Newtonian liquids.

New possibilities of preprocessing

- New settings **Method = Replace in cropped volume**, **Method = Add in cropped volume**, and **Method = Average in cropped volume** for **Initial conditions** and **Modifiers**. This new feature allows taking into account the difference between the volume of the computational grid and the volume of the geometric object, in which the modifier or initial conditions are activated.
- Simulating of very viscous flows (it is turning on by the **Very viscous flow**) allow the program to improve stability and accuracy of simulations of high viscosity liquids (for example, in simulations of forming rubber parts).
- Support of *FSI* computations done by *CSE* method with *Abaqus 2020*.
- Now it is possible to exclude the equation for the electric potential from EMHD equations or apply them with a **Correction factor**. This is set in **MHD parameters** in the Advanced settings of Solver.
- In the interface of connectors **External Connections >** **Arbitrary external connection** it is possible now to specify arguments for running an external program.
- Accuracy of simulating bubbles in the **Particles** phase was improved. Simulations of the force pushing bubbles off a wall and the lifting force acting on rotating particles were added (they are enabled by parameters **Repulsion force** and **Lift force** in properties of the **Phase interaction** element for pair of **Phases** "continuum-particles").

New possibilities of postprocessing

- Displaying streamlines on a surface.
- Export of loading from **Supergroups** into a `glo` file.
- A new physical variable, **Conservative velocity** was added.
- The **Color contours** layer with interpolation is now built more quickly.
- A new standard **Palette**, `inferno.fvpal`, has been added to **Postprocessor**.

- Exporting results of the computation into the *VTK* format for *paraview* or other software for scientific data visualization.

Improvements of the user interface

- New **Initial grid editor** with convenient graphical user interface, which allows you to tune sizes of cells of the initial grid. The old **Initial grid editor** and the **Alternative Initial Grid Editor** have been removed.
- New uniform design of the **Table editor** (see Editor of tables of only one argument and Editor of tables of several arguments).
- **Viewer** now have new buttons in its toolbars,  (**Rotate the graphics window view 90 degrees counterclockwise**) and  (**Rotate the graphics window view 90 degrees clockwise**) that allow you to rotate the graphical window 90 degrees.
- An ability to select and copy into the clipboard texts of most of the error messages and warnings (using **Ctrl+C** hot keys).
- In **Preprocessor** now it is possible to set the color of a **Boundary condition** in its properties (not only by the **Set color >** ***(color)*** command from the context menu as it was before).
- In element selection window (for example, the window for selecting **Substances** of a **Phase** or the window for selecting **Phases** of a **Model**) you can drag by the mouse elements in lists from one pane to another.
- More convenient user interface of the dialog box for loading an assembly.

Other changes

- Heat source now acts on the dispersed **Phase** too.
- Particles can now be simulated with sliding surfaces and conjugated boundary conditions.
- It is possible to receive license statistics from Terminal.
- The **FvLicenseUtil** utility is now also available in *Windows*. This utility allows you, from a command line, to examine a license status, register a license or obtain information for generating a license.
- The **Cell debug** layer now includes parameters **Information window > ...** that allow you to set the volume of data that will be transferred into the **Info** window.
- In the **Cell debug** layer data save into a `glo` file is available now.
- In **License Manager** gathering statistics of license use is now turned on by default for new installations of *FlowVision*.
- Improvements in the documentation and in the *Tutorial* (new examples of typical problem settings).
- Improvements in the user interface for selection of algebraic solvers to be used (*Aggregation AMG*, *Selective AMG*, *TParFBSS*). Now it is possible to specify individually the maximal numbers of iterations for the solvers. These parameters are set in Advanced settings of Solver.

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.7 What's new in FlowVision 3.11.02

Here the most important changes and new possibilities, added to the version *FlowVision 3.11.02*, are listed.

Important changes

Before installing and using *FlowVision 3.11.02* ensure that you have read this section.

Since the next version of the program, support of 32-bit *Windows* operating systems is entirely disabled. No *FlowVision* modules will be able to run on 32-bit versions of *Windows*. Please update your operating system in advance if this is required.

New

FlowVision 3.11.02 contains new features; the most significant of them are listed below.

- Export to a **`glo`**-file, in the STL format, the layers **Isosurface** and **VOF (visualization of a free surface)** is now available. This can be used to specify initial conditions for a new simulation based on variable fields that were preliminary calculated in another simulation.
- The **Viewer** module now can playback of **`fvvis`** files in an infinite loop. Such loop playback can be used to show results of the simulation, for example, at presentations or exhibitions. **Viewer** doesn't require license and can be installed on any demo-computer. The loop displaying is turned on by the **loop_playback** key when when Viewer is started from a command line or in the batch mode.
- Specifying criteria of **Small cells** has been relocated to **Phase Limiters**. This allows you to specify different criteria of **Small cells** for different **Phases**.
- Heat capacity in properties of a **Substance** can be specified dependent on the temperature of the dispersed **Phase**.
- Specifying properties of a **Substance** depending on its concentration.
- **Voltage Drop** can now be set on a connected boundary condition for electrical potential.
- A new type of connectors, **Arbitrary external connection**, has been added to **External Connections**. It allows you to link a user's connector for a joint FSI simulation with an arbitrary third party's software. Also a test support for joint FSI computations with *Nastran* has been added.

See the full list of changes in the file **`ReleaseNotes_eng.txt`**.

2.8 What's new in FlowVision 3.11.01

Here the most important changes and new possibilities, added to the version *FlowVision 3.11.01*, are listed.

Important changes

Before installing and using *FlowVision 3.11.01* ensure that you have read this section.

- Update the **License Manager**.
- Functionality of FSI simulations has been substantially rearranged:
  - Module **MP manager** has been excluded from the delivery because its functionality is now included in **Pre-Postprocessor**.
  - You don't have to edit text file of the project to prepare the project for connection to *Abaqus*. All necessary operations are now done in the user interface of **Pre-Postprocessor** in the **External Connections** folder. Now the program supports connection to *Abaqus*, besides *Direct Coupling*, also via the *Co-simulation Engine* (*CSE*) protocol.
  - FSI projects that have been prepared in previous versions of *FlowVision*, will be automatically converted when they are opened in **Pre-Postprocessor**. The program will create all necessary elements in the **External Connections** folder.
  - If an FSI project contains more then one exchange surface, you have to carry out additional steps. Follow instructions displayed by the program.

Starting from the next version of the program after *FlowVision 3.11.02*, support of 32-bit *Windows* operating systems is entirely closed. No *FlowVision* modules will be able to run on 32-bit versions of *Windows*. Please update your operating system in advance if this is required.

New

FlowVision 3.11.01 contains new features; the most significant of them are listed below.

New physical models and boundary conditions

- Now it is possible to simulate influence of not only electric fields but also magnetic fields on flow of fluid.
- Models of breakup and coalescence of droplets allow the program to calculate changes of sizes of liquid droplets in aerosols during breakup and coalescence processes.
- On the boundary condition **Wall** it is possible now to set parameters of radiation equilibrium. This provides a simplified method for taking into account heat losses on surfaces due to radiation without brute computation of the radiation over the whole computational domain.
- On the boundary condition **Inlet/Outlet** the **Velocity** variable can be specified via **Supersonic inlet**.
- Coefficient k_{HE} of heat exchange between continuous and dispersed **Phases** has been added. This coefficient is set by the **Heat exchange coef. (D-C)** parameter in properties of the element **Models > Phase interaction > Continuum-dispersed**.
- Simulation now can be done with disabled calculation of velocities (only pressure is calculated). This mode is set by the **Off speed** parameter in advanced settings of Solver. Simulation without calculation of velocities is useful when you don't need calculate velocities and you are interested in pressure only, for example, when you solve a steady-state FSI problem that includes transfer of loads on complex and changing surface.

New abilities of preprocessing

- The **New project** dialog box has been added, which allows you to select either creation an empty project or loading the project's geometry model from a file (as it was implemented in previous versions of the program).
- You don't have to save a new project on a disk immediately. You are may either not to save the project or save it at a time, which is convenient to you.
- A new **Geometry** tab, which facilitates preliminary preparing of geometry elements of a project, has been added into **Pre-Postprocessor**. The prepared geometry elements can be used as geometry of the computational domain, of **Subregions**, and of **Imported objects**.
- Functionality of *FlowVision* can be expanded by user modules from third-party manufacturers.
- In properties of geometric **Objects** and **Movements** you can easily change their local coordinate systems (LCS) specifying its parameters explicitly or copying the data from LCS of another **Object** in the **Coordinate system adjustment** dialog box, which opens when you click on the  (**Coordinate system adjustment**) button. See General properties of Objects.
- Now the program can refine the computational grid depending on curvature of surfaces and sharp edges that is set in properties of adaptations by parameters **Adaptation to curvature** and **Adaptation to sharp edges**.
- For the **Wall** boundary condition, a new option for specifying the velocity has been added, **Track**.

- Now you can set **Stopping conditions** not by time only, but also by steps of the computation and you can also specify their activity not from beginning of the computation only, but also from the specified time moment or since the specified step. This provides you with flexibility is setting the stopping conditions and you can filter initial disturbances of the solution.

New abilities of postprocessing

- The "Read only" mode of **Pre-Postprocessor**'s operation has been added, which allows you to view results of computations and the client part of a project even when the license expired or all license options are consumed.
- It is possible now to connect **Pre-Postprocessor** to a **Solver**, on which FSI computation is running.
- A new geometric object **Set of sensors** has been added, which is a convenient tool for reading off local characteristics over several points in the computational area or on a surface.
- Now you can display in the **View** window custom captions and current time of step number of the physical process; this is set in properties of the root folder **3D-scene**.
- A new variable has been added, **Wall temperature**.
- **Characteristics** over a surface now include coordinates of **Center of pressure**.

Other changes

- A new parameter of **Solver-Agent** has been added, **NumaThreadNum**, which specifies the number of numa cores per processor. The number of cores on the remote (relating to **Solver-Agent**) computers can be differ from the number of cores on the computer, on which **Solver-Agent** runs. To run computations via a queue in **Terminal** on a remote computer, you can use the **NumaThreadNum** parameter to specify how many cores are available on the remote computer.
- *FlowVision* now is tested for compatibility with *AltLinux* family of operating systems (*Alt Workstation 8* and *Alt Server 8.2*).
- New hot keys for work in the Properties window were added.
- Icons of hidden or non-active elements are now displayed dimmed or gray.
- Now the program can work with addresses in the *IPv6* format. To enable operation in *IPv6* networks, you have to specify **UseIPv6=Yes** in configuration files and specify correct addresses in settings of all modules.

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.9 What's new in FlowVision 3.10.02

Here the most important changes and new possibilities, added to the version *FlowVision 3.10.02*, are listed.

Important changes

Before installing and using *FlowVision 3.10.02* ensure that you have read this section.

- The **Implicit new** computational scheme is now renamed as **Implicit**, and the former **Implicit** scheme has been removed.

New features in Solver

The following new settings were added to the **Advanced settings of Solver**:

- The **Matrix Solver**, which can be used for computations in the **Chemistry** mass transfer model. **Matrix Solver** provides more stable development of solution at CFL>1 in simulations with large number of components (**Substances**) and large number of chemical reactions.
- The **Skew scheme**, which computes convective flows on faces of cells with taking into account the mass flow through edges of the cells. Use of **Skew scheme** essentially reduces the scheme diffusion for flows that substantially deviate from directions of Cartesian axes (that are particularly vortex-like flows). Meantime **Skew scheme** is being beta-tested.

Improvements

- When an **Adaptation** is created, if the **Region** has only one computational **Subregion**, this **Subregion** will automatically be added to conditions of the adaptation.
- Now it is not possible to connect from **Pre-Postprocessor** or **Viewer** to a **Solver** with another version number. This protects user's data from damage if a module with incorrect version number was launched.

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.10 What's new in FlowVision 3.10.01

Here the most important changes and new possibilities, added to the version *FlowVision 3.10.01*, are listed.

Important changes

Before installing and using *FlowVision 3.10.01* ensure that you have read this section.

- To use *FlowVision 3.10.01*, you have to update **License Manager** to its version *3.10.01*.
- When a project is restarted from scratch (if the **Continue calculation** checkbox is unselected in the **Starting solve** dialog box), the existing `glo` and `fvvis` files will be removed.
- The user interface for work with adaptations and computational grid has been rearranged. All kinds of adaptation are now specified in a single place. The specified adaptations can now be applied on multiple objects. Please note, that:
  - After the project's conversion, it is strongly recommended that you check the specified parameters of adaptations and its activity; also it is recommended to use new abilities for improving the setting of the adaptation.
  - Similarly, after conversion of old projects, you should examine correctness of specification of boundary layer grids and, if necessary, respecify the boundary layer grids so that there would be no simultaneously several active boundary layer grids (with **Enabled=Yes** specified in their properties) as the program can use only one boundary layer grid at a time.
  - Reducing the number of objects of adaptation allows the program to reduce the time of the computation. For this we recommend you to group surfaces and volumes with the same parameters of adaptation.
  - A new type of adaptation has been added (**Adaptation by condition**), which is convenient for solving shocks or free surfaces.
- *For simulations with the mass transfer:* please note that in the version *3.10* the definition of coefficients **F** and **D** has changed in the mass transfer equation. If coefficients **F** and **D** are used in the simulation, then in projects, which were made in an old version, these coefficients were additionally multiplied by the **Density**. Starting version *3.10,* when specifying them, you have yourself multiply a numerical value by the **Density**. If you wish to continue a computation, which has been prepared in a previous version, you have to change the coefficients **F** and **D**.
- Licensing for **Moving bodies** has been changed. Now you can use **Moving bodies** even if you don't have the **Moving Bodies** license option, if the **Moving bodies** don't move (**Update > Type = Disabled** is set in the properties of such **Moving bodies**). This allows you to use **Moving bodies** for quick replacement of the geometry model and for a simple change of the configuration of the computational domain, for example, by rotating a **Moving body**.

New models and boundary conditions

- A new mass transfer model has been added, **Chemistry**, which allows computation of successive reactions. The old, a simpler mass transfer model, **Mixing+Chemistry**, is now renamed as **Mixing**.
- Now it is possible to simulate **Ablation**.
- Now it is possible to specify the **Anisotropic thermal conductivity** modifier also in the dispersed medium with ability to specify **Phases** where the modifier will act. This allows the program to simulate flow of a liquid through a bundle of hot pipes or through other porous areas with a heat source and anisotropic thermal conductivity.
- Now it is possible to specify heat transfer from a free surface to **Vacuum**. This allows taking into account heat losses from the free surface even if the second phase is not simulated. See the subsection *Folder "Models > Model #N > Phase interaction"*.
- A new boundary condition **Fixed velocity** (it can be set when simulating subsonic motion on BCs **Inlet/Outlet** and **Free outlet**) has been added, which allows you to specify strictly a known profile of velocities on an inlet or outlet.
- Non-Newtonian fluid model are now specified by a law for viscosity in the properties of a substance; viscosity of a non-Newtonian fluid can be set by a formula, by a table, or by a non-Newtonian fluid's model *Power Law*, *Herschel-Bulkley*, or *Bird-Carreau*.
- In k-ε turbulence models (**KES**, **KEAKN**, **KEFV**, **KEQ**) the turbulent viscosity now can be calculated using Bradshaw formula.
- In the SST turbulence model now it is possible to enable a new model of the laminar-turbulent transition $\gamma - \tilde{R}e_{\theta t}$ that can be used for a more accurate prediction of the laminar-turbulent transition (this is enabled by the parameter **Transition=Yes** in the properties of the **Turbulence** physical process.

Postprocessing

- The **Monitor** window is now significantly modified. Now the **Plots** can be tuned flexibly and it is possible to display more information in the **Status** tab.
- The **Formula editor** now contains a random value generator (the `random` operation), which can be used, for example, for specifying fluctuating turbulent initial conditions.
- The layer **Streamlines** can now be built not only in the **Computational space** but also on surfaces. This allows you to view motion of the flow along the surfaces.
- Now it is possible to integrate the main variable and the coloring variable of the **Layers** **Plot along line**, **Plot along curve**, and **Plot along ellipse** when these **Layers** are built on a **Line**, see General properties of Layers.
- For your convenience, to relieve you of frequent use of reference books, several new constants have been added into the **Formula editor**:
  - **Gas constant**
  - **Stefan–Boltzmann constant**
  - **Boltzmann constant**
  - **Planck constant**
  - **Vacuum permittivity**
  - **Vacuum permeability**
  - **Gravitational constant**
  - **Speed of light in vacuum**
- New physical variables have been added:
  - **Rad. energy flux** (radiative heat flux on a wall)
  - **Shear stress (vector)**
- In the **Internal characteristics** now there are available value of the reference temperature and pressure and also the gravity vector. This allows you, for example, to create in **Postprocessor** user variables **Tabs** and **Pabs** that correspond to the reference temperature and pressure, and these variables will allow you to automatically take into account changes in settings of the reference values.
- In FSI computations now it is possible to calculate loads in nodes not only proportionally to angles of triangle area elements, but also using Voronoi diagrams. This allows more precise calculation of transferred loads.
- In the layer **Distributed characteristics** now it is possible to select **User variables**.

Improvements of the user interface

- Now it is possible to display the **Layer**'s name in the legend of a palette (this is set in the general properties of the Layer as **Palette > Appearance > Title = Yes**).
- When you select a line in the **Initial grid editor**, this line will be highlighted in the **View** window. Also the same highlighting is done when you select a line in the properties of the **Initial grid**.
- Now it is possible to create complex **Supergroups** consisting of several boundary conditions and arbitrary surfaces. This is useful when you need to receive characteristics from different parts of an object, which consists of various combinations of surfaces.
- You don't have to set a specific order of **Substances** in the **Combustion** mass transfer model anymore. Now you have to explicitly specify (in the properties of the physical process) what is **Fuel**, **Oxidizer**, and **Product(s)**.
- Hot keys can be used for some actions in the **Properties** window.
- Icons of permanently inactive objects are displayed in faint colors.
- You can change the background color in **Viewer** using the  button in the **Views** toolbar.

Other

- **Viewer** now supports the **Computational grid section** layer.
- **Terminal** displays the assembly date of the running **Solver**.
- The **Log** window of **Pre-Postprocessor** now displays additional information about found errors in the settings of the project.
- The **Explicit new** computational scheme is now renamed as **Explicit** (the computational scheme is specified in the **Advanced settings of Solver** by the **Numerical method > Type of scheme** parameter).
- *FlowVision 3.10.01* is tested in *Windows 10* and now *Windows 10* is officially supported.
- One possible value of the parameter **Variable > Component** the properties of **Characteristics** has been renamed (from **Source** to **Vector**).
- Advection schemes **Upwind scheme** and **Smooth reconstruction** have been renamed as **1st order scheme** and **2nd order scheme**.

- Project tree object **Local frames of reference** is renamed as **Local coordinate systems** and **Local FR #N** are renamed as **Local CS #N**.

See the full list of changes in the file **`ReleaseNotes_eng.txt`**.

2.11 What's new in FlowVision 3.09.05

Important changes

Before installing and using *FlowVision 3.09.05* ensure that you have read this section.

- Changes have been made in **License Manager**, it is recommended to update it up to version *3.09.03*. In simulations with conjugate heat transfer a message might be displayed informing about lack of licenses. If you see such message, please, contact our sales manager, which works with you, or write about this problem to the technical support service.
- Important changes have been made relating to compressibility:
  - The artificial compressibility near surfaces of movable boundaries is now completely defined by parameters of **Moving bodies**, it can be enables or disabled there (in the group of parameters **FSI**).
  - Parameters **Artificial compressibility** and **Sonic speed** in the properties of the element **Phase #N > Physical processes > Motion** and parameter **Compressibility** (χ) in the properties of liquids have been removed. Be careful when transferring projects from older versions, make appropriate changes. Instead of parameters of compressibility, you can specify the partial derivative of **Density** with respect to **Pressure** (see below) or specify **Density** by a formula, a table, or a physical law.
  - When density of a **Substance** is specified as a constant, you can, in properties of the **Substance**, specify partial derivatives of **Density** with respect to **Pressure** and to **Temperature** ($\frac{\partial \rho}{\partial P}$ and $\frac{\partial \rho}{\partial T}$), which are used for approximation of **Density** in a neighborhood of reference values of **Pressure** and to **Temperature**, see section Folder «Substances».
- Default values of the following advanced settings of the Solver have changed:
  - **Numerical method > Type of scheme**. Now, by default, the **Implicit new** scheme is used, which has better performance and convergence.
  - **Numerical method > Pressure gradient**. Now the default value of this parameter is **Simple**. This option provides much better convergence in many simulations, but can provide worse accuracy comparing to other options of this setting.
- For continuous **Phases** it is necessary now to specify the method of solution of the equations for the **Heat transfer** physical process. You can select the following options:
  - **Heat transfer via h** is convective and diffusive energy transfer via thermodynamic enthalpy.
  - **Heat transfer via H** is convective and diffusive energy transfer via total enthalpy.

  When old project are converted for a new version, the following rules are used: For a gas, which density is defined by the ideal gas equation, this parameter is converted to **Heat transfer via H**. For a liquid, solid body or gas with constant density, this parameter is converted to **Heat transfer via h**. Recommendation for use and details see in sections Folder «Phases» and Theory > Physical processes > Heat transfer > Equations.
- Rules for calculations of **Characteristics** by the variable **VOF** changed radically (you have to take this into account if you calculated characteristics by **VOF** in your previous projects). Calculation of **Characteristics** by **VOF** is done over the whole volume or over all specified surfaces of the **Object**, where not only the **VOF** variable's **Phase** is defined, but also where the other **Phase** is defined too. See details in the section Specifics of calculating Characteristics by the variable VOF and illustrations in the subsection *"Domain of integration (for calculation Characteristics) depending on the Variable's category"* of the section Characteristics.
- **Autorotation** is now a property of a **Rotation** (see section Folder «Local coordinate systems»), but not a property of a **Boundary condition**.
- Since release *3.09.05*, *Windows XP* and *Windows 2000 Service Pack 3* are not officially supported.
- Operation with *Microsoft MPI (MS MPI)* is possible with version *7* only or more recent version of *MS MPI*. If an older version of *MS MPI* is installed, you have to update it or remove it and install a supported version.
- If in an old project in a some **Layer** the **MachNumber** variable has been visualized, you have to specify it again in parameters of this **Layer**. Such situation is marked by the "**!**" symbol near the icon of the **Layer**.

New features

FlowVision 3.09.05 contains many new features, the most significant of them are listed below.

In Pre-Postprocessor

- The program's functionality now enables a new sector-sliding problem setting, which assumes binding the radial surfaces of sectors by periodic surfaces, and binding the cylindrical surfaces by sliding surfaces. See the section Sector-sliding setting for details and see an example of such simulation in the *Tutorial: Examples*

of typical tasks document, section *"Sector of axial compressor"*. This technique allows the program to substantially reduce the time of simulation rotor-stator settings with periodical structures.

- The variable **Velocity** on the boundary condition **Wall** can be now specified as **Slip** (when **Wall > Wall interaction > Phase #N** = **No wall functions**). See details in the section Theory > Physical processes > Motion > Boundary conditions > Template 'Wall'.
- When installing *FlowVision*, now you can also install *3DTransVidia*, see section *Installation of 3DTransVidia along with FlowVision*.
- Now you can copy text from **Info** windows using the **Ctrl+C** hot keys.
- The parameter **V*nabla_P for h** in the properties of the physical process **Heat transfer** allows you to enable or disable taking into account the convective derivative of pressure (the term $\mathbf{V}\cdot\nabla P$ for the time derivative $dP/dt = \partial P/\partial t + \mathbf{V}\cdot\nabla P$) in the energy equation written through h.
- It is possible now to use a boundary layer grid and VOF phase transfer together, in this case calculations and data exchange are done in only those boundary layer grid's cells that are filled by liquid.
- A new **Modifier** has been provided, **Anisotropic thermal conductivity**.
- For the boundary condition **Inlet/Outlet** it is possible now to specify **Velocity** as **Total pressure and velocity direction** (the user specifies the direction of the velocity's vector). See details in the section Theory > Physical processes > Motion > Boundary conditions > Template 'Inlet/Outlet'.
- New **Physical variables** have been added:
  - **Random value** returns a random value in the range [0,1].
  - **S-criterion** is the invariant of the strain rate tensor.
  - **W-criterion** is the contraction of the velocity gradient tensor.
  - **Q-criterion** is a half-difference of squares of **W-criterion** and **S-criterion**. **Q-criterion** allows visualization of areas where either shear flow or vortical flow dominates.
  - **Vorticity** is curl of the velocity.
- In the **Formula editor** a new function has been added, **defined()**, which depends on whether its argument is defined in a cell.
- Now it is possible to specify the method for calculating the total pressure, either **Simple** or **Exact**. See detailed information and recommendations for selecting the method in section Element «Advanced settings» (advanced settings of Solver).

In Solver

- 64-bit **Solvers** in a single-processor mode do not require an installed MPI anymore.
- Export of loadings can be done now not only for loading on nodes, but also on faces (triangles) of geometry models of objects. This is set by the parameter **Export loadings > Export subject** in the properties of **Advanced settings** of **Solver**. See details in the section Parameters of loadings export.
- A new type of the algebraic solver has been added, **AST**, which is a combination of *Aggregation AMG*, *Selective AMG* and *TParFBSS* algebraic solvers. Use of the **AST** solver generally speeds up the simulation. Type of the algebraic solver is set by the parameter **Algebraic solver > Solver type** in the properties of **Advanced settings** of **Solver**.

In other modules

The **Terminal**'s **License monitor** now displays a queue of requests waiting for free licenses. This queue is only formed when a non-zero parameter **LicenseTimeout** is specified in the configuration file of **Solver** (`FvSolver.cfg`), which defines the Solver's timeout for waiting a license.

Improvements

FlowVision 3.09.05 contains many improvements, the most significant of them are listed below.

In Pre-Postprocessor

- As the variable **Y_plus** is defined for walls only, it is calculated for those **Layers** and **Characteristics** only, which are built on surfaces of walls.

Translation to English

- The *Tutorial: Examples of typical tasks* manual has been completely translated to English (human translation).
- Translation of the *User's guide* has also been improved, but many pages of the manual are still machine translated.

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.12 What's new in FlowVision 3.09.04

Important changes

Before installing and using *FlowVision 3.09.04* ensure that you have read this section.

64-bit versions of Pre-Postprocessor and Viewer

64-bit versions of **Pre-Postprocessor** and **Viewer** were implemented, which can work with volumes of memory more than 4 Gbytes, which is required, for example, for work with complex geometries (when the geometry model contains large number of facets) and, in some cases, with complex visualizations.

New

FlowVision 3.09.04 contains new features; the most significant of them are listed below.

In Pre-Postprocessor

- In the **Formula Editor** new operations were added, which allow defining a local variable on a surface of a boundary condition by the value from the center of the cell, which is adjacent to the surface (**ignore_bc**, **ignore_bc_vector**), and not immediately from the surface of the boundary condition as it is doing by default. If on a boundary condition it is required to calculate a local variable's value, which depends on the value from the previous time step, then use of these operations is the only correct approach, because the value from the previous time step is not stored in the memory but calculated at each time step and so must not depend on itself. Also these operations allow use of values of velocity near a wall in characteristics, because velocity's values is zero immediately on the wall.
- In the turbulence model the constants, which control the compressibility, are now available for being set by the user.

In other modules

- Installation files of the **License Manager** and the main installation pack are provided in individual archives.

Improvements

FlowVision 3.09.04 contains many improvements, the most significant of them are listed below.

In Pre-Postprocessor

- Now in multiphase problems for each **Phase** its own **Limiters** are specified. The **Limiters** are now moved from the **Advanced settings** object into a new object **Limiters** in the project tree. If a project from an old version is converted, a separate set of **Limiters** would be created for each its **Phase**.
- To protect projects from being damaged, the program prohibits now connection of **Pre-Postprocessor** to a **Solver** with a different version number.
- In simulations where only one **Phase** exists, there are no **Phase volume** variable.
- In simulations with a dispersed **Phase**, the phase volume of the continuous **Phase** is calculated automatically and is not specified by the user.
- Variables **VOF** and **Phase volume** are now excluded from the categories of variables **Common and phase-unrelated variables**. They are now presented in the lists of variables in categories **Variables of phase "Phase #N"**.
- There no closed gap cells anymore. To make leaks from the gap be small to negligible, specify large effective viscosity in gap, which should exceed the molecular viscosity.

In Solver

- The automatic grid generator was substantially improved, so it is possible now to work with complex geometry models with large number of facets.
- Now uni-processor solvers are not linked to *numa* nodes by default. This prevents loss of performance when several uni-processor solvers start to run on a single computer (node). If you wish, you can turn the linking on by setting `affinity=true` in the solver settings.

In other modules

- Several minor improvements were made in **Viewer**, including icons consistent to the icons in the toolbar of **Pre-Postprocessor**.

Translation to English

Translation was improved in many pages, but many pages of the manual are still translated using machine translation.

See the full list of changes in the file **`ReleaseNotes_eng.txt`**.

2.13 What's new in FlowVision 3.09.03

Important changes

Before installing and using *FlowVision 3.09.03* ensure that you have read this section.

Errors fixed, stability increased

The release *FlowVision 3.09.03* does not include changes in functionality but provides more stable and precise solutions.

Valuable improvements were added, which allow preventing disadvantage existing in version *3.09.02* that caused substantial decreasing of accuracy of solution in projects, which apply the following technologies:

- **Moving bodies** with an enabled **Update** option
- solution of the "rotor-stator" problem with rotating coordinate system (either with a sliding surface or without it)
- boundary layer grid

Installation and update

If you don't have the *FlowVision 3.09.02* version installed, then before installation *3.09.03* you have to read the section "What's new in FlowVision 3.09.02". All described there specifics of upgrading from versions *3.08.xx* or *3.09.01* are also actual.

If you have the *FlowVision 3.09.02* version installed, then you have to refuse of its use. You do NOT have to update the **License Manager** when upgrading from *3.09.02* to *3.09.03*.

Updates of License Manager are not so frequent now

Starting from version *3.09.02* we do not update the **License Manager** with each new release of other *FlowVision* modules. Installer of the **License Manager** is supplied now as a stand-alone executable file.

The **License Manager** is updated much less frequently then other *FlowVision* modules, so the actual version of the **License Manager** will have smaller version number then the version number of the main distribution package of *FlowVision*.

Translation to English

Please note, that many pages of the manual are still translated using machine translation software. Chapter Theory has been translated manually.

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.14 What's new in FlowVision 3.09.02

Important changes

Before installing and using *FlowVision 3.09.02* ensure that you have read this section.

Changes in user interface of Pre-Postprocessor:

- for the **Phase Transfer** physical process for *continuous* **Phases** the **Time step coefficient** parameter is not available for editing. Use the **Advanced settings** **> Multiphase C > Relaxation** parameter instead in the **Solver** tab.

Changes in interaction between License Manager and Solver:

- To use **Solver** version *3.09.02* you need upgrade the **License Manager**. Starting from version *3.09.02* **Solver** correctly inter-operates only with **License Manager** version *3.09.02* or newer.
- **Solver** of *FlowVision* of versions *3.09.01*, *3.08.xx* do correctly work with **License Manager** version *3.09.02* or newer.
- To upgrade **License Manager** from version *3.09.01* to version *3.09.02* it is enough to install **License Manager** version *3.09.02* over existing version of **License Manager**. The installer will automatically do the upgrade and will not ask you about the file system paths.
- Procedure of transferring the license from version *3.08.xx* to version *3.09.02* is described in section Transferring a license from version 3.08.xx to version 3.09.xx.

English translation of the user documentation:

- The English version of the documentation has been made. Please note: many pages of the documentation have been automatically translated. Chapter Theory has been translated manually.

New

FlowVision 3.09.02 contains a number of improvements and new features; below the most significant of them are listed.

Visualization:

- New layer **Volume visualization** allows you to visualize changing variables in the whole space (for example, view zones with high substance concentration or density). Before use of the **Volume visualization** layer ensure that you have installed the last versions of video card drivers.
- Layer **Vectors**, specified in volume, allows you to visualize the boundary layer grid. A new group of parameters, **Computational grids** allows you to include or exclude visualization of vectors in the boundary layer grid and in the main grid. This group of parameters is only available when parameter **On regular grid = No**.

New capabilities of data export:

- Implemented data export for visualization in a third-party visualization software *EnSight*. See section Data export for visualization in EnSight.
- Implemented the ability to export data to `glo` and `fvvis`-files based on previously calculated and saved data. This function allows you to upload a newly created visualization layer or characteristics, as well as data into files for *LMS*, *EnSight* or *TORT* without recalculation of the project. See section Data export after computation.

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.15 What's new in FlowVision 3.09.01

Important changes

Before installing and using *FlowVision 3.09.01* ensure that you have read this section.

Some objects in Pre-Postprocessor were renamed

- Since version *3.09*, the variable **Volume of phase** represents the relative volume of a dispersed or continuous phase (when interaction between a dispersed and continuous phases is simulated). Please note that in versions *3.08* and earlier version the variable **Volume of phase** represented the volume of the continuous phase in problems with a free surface, see Theory > Physical processes > Phase transfer. Now this variable is called **VOF** ("volume of fluid").
- Limiters for the time step have been added (parameters **Max step** and **Min step** in the **Properties** window of the **Time step** element in the project tree). Please note that the parameter, which was called **Min step** in the older versions of the program, is now called as **Explicit time step limit**.

Protection against incorrectly specified project

- In this case, the incorrect elements in the project tree are marked with a "**!**" symbol. If, for example, in a **Substance** the specific heat, which is used to calculate the Mach number during simulation of gas flows, is not specified, then, when you try to start such a project to computation from **Pre-Postprocessor**, the **Substance** object will be marked with a "**!**" symbol and the project will not run. When you try to start the project without **Pre-Postprocessor**, **Solver** also will not start the computation and will write a message into its `.err` file.

Support of *MPI* in *Linux*

- List of supported *MPIs* has been changed. Now libraries for operation with *MPI* are not compatible with earlier versions. However, there is an opportunity to compile a library by your own for any version of your *MPI*. It is possible to make a compilation for a combination of *FlowVision* with any implementation of *MPI*.

The Installer

- Installation of *FlowVision 3.09* does not require the removal of the previous versions of the program; now you can have several different versions of *FlowVision* installed on one computer (but these versions have to use different ports).
- Now installations in the command line mode (text mode) and in the graphical mode (GUI mode) are supported both for *Windows* and for *Linux*.
- All *FlowVision* modules for *Linux* are now 64-bit only, since most of the computational servers do not have pre-installed 32-bit libraries.

Transferring settings near-surface artificial compressibility from of old projects

- Setting the subsurface artificial compressibility for moving bodies (parameters **Flexibility** and **Mobility**) is now set for a **Moving bodies** (used to be in the configuration of the **Physical processes**). So now for the different **Moving bodies** can set different parameters subsurface artificial compressibility. Subsurface artificial compressibility is used usually in problems FSI, or when the density of the floating body substantially less than the density of the liquid in which it floats. **Flexibility** and **Mobility** values specified in version *3.08* for the physical process of **Motion**, while porting old projects into version *3.09*, will *not* be automatically transferred into the properties of **Moving bodies**, so when porting the old projects, these parameters need to be specified manually for those **Moving bodies** to which they have to be applied.

New

FlowVision 3.09.02 contains a number of improvements and new features; the most significant of them are listed below.

New models

- Now it is possible to calculate particles in dispersed medium with a phase transfer (see section Theory > Physical processes > Processes in the presence of dispersed medium). Simulation of processes in a dispersed medium can be done with computational schemes **Implicit new** and **Explicit new** only.
- Simulation of electromagnetic hydrodynamics (EHD) in the framework of the implemented electrodynamic model. Simulation of EHD can be done with computational schemes **Implicit new** and **Explicit new** only.
- The model of anisotropic porosity has been implemented (within the model of dispersion medium), which takes into account the heat transfer process between of the porous carcass with the heat-transfer fluid. An example of this problem is a pipe with cold water, which is filled with multiple tubes through those hot water flows (cold

water is being heated during this process). To avoid doing computations for hundreds of tubes, you can be replace this computations by an equivalent hydraulic resistance and take the heat transfer equations into account indirectly. This approach allows you to solve problems accurately, having a significant economy on building the computational grid. An anisotropic resistance is specified using the **Anisotropic resistance Modifier**.

- The implemented model of a sliding surface, **Frozen rotor**, which doesn't simulate slipping of cells on the boundary between **Subregions** (rotor and stator). **Frozen rotor** allows you to get an initial approximation spending less computational resources.

Boundary conditions

- A new type of connected boundary conditions has been added, **Sliding surfaces**, which allows you to connect a rotating and a fixed **Subregions**. This feature is useful, for example, for solving problems of interaction between rotor and stator, it substantially saves the computational time compared with solutions, which use **Moving bodies**.
- A new type of connected boundary conditions has been added, **Periodic surface**, which allows you to connect repeating boundaries of identical fragments of a model and replace a problem's solution in a large computational domain by a solution in a small computational domain (consisting of one fragment only).
- Enabling and disabling of wall functions is now specified on a **Boundary condition**, not in the settings of **Physical processes**; this allows you to solve problems, which has surfaces with very different Y+, for example, when you need to solve a boundary layer on one surface and use of near-wall function is desirable on another surface.
- Translational motion in local coordinate systems now allows you to specify the tangential component of velocity on a **Boundary condition**.
- Now it is possible to simulate autorotation while using **Sliding surface**, when the rotation is caused by the influence of the incoming flow (a differential equation is solved with respect to the angular velocity, taking into account the moment of inertia J, the additional moment of inertia J_{sub} of liquid or gas near the rotor, the external torque T_{ext} and hydraulic torque T_{hyd}), see section Folder «Local coordinate systems». Simulation of autorotation is used in problems of interaction between rotor and stator with a rotating coordinate system.

Computational grid

- It is possible now to specify an additional *boundary layer grid* (BL grid) for the explicit resolution of the boundary layer and use of Low-Re turbulence models (Y+ is below 1). See sections Theory > Physical processes > Boundary layer grid and Boundary layer grid.
- Now you can specify levels and number of layers of adaptation on boundary conditions by formulae. This allows you to specify your project in such a way that it will start on a coarse grid with a coarse step, and then, during the computation, after some time (set by a formula) elapses, the program will refine the grid by an adaptation.
- A new method of editing the initial grid has been added, now it is possible to add manually grid lines into the array of lines. This provides you with additional flexibility of customizing the existing computational grid.

Turbulence parameters

- Users can now specify themselves many numerical parameters (constants) of turbulence models, which were hard-coded in algorithms in old versions of the program. Normally this is not required, but it might be necessary when solving some specific problems. These parameters are specified in the properties of the **Phases** in the parameters of the **Turbulence** physical process (see subsection *Parameters of physical process "Turbulence"* in the section Folder «Phases»).
- Set of turbulence settings in the advanced settings of **Solver** has changed (see Element «Advanced settings»).

Settings in the tab Solver

- A new computational scheme **Explicit new** has been developed. This scheme, as the **Implicit new** scheme, has a better performance and stability. When using **Explicit new** scheme you should specify the values of CFL numbers as equal to 1.
- Now you can specify the time step by formulas or tables. This allows the program to reduce or increase the time step at required times.
- New multiphase parameters were added, which are specified in the advanced settings of **Solver** in groups of settings **Multiphase C** and **Multiphase D** (see Element «Advanced settings»).

Geometry models with multiconnection

- Now it is possible to use Boolean operations on assemblies with overlapping bodies and run projects with multiconnection. *Multiconnection* is connection of three or more surfaces on one edge; a particular case of multiconnection is connection of surfaces with a T-connection.

Post-processing

- The layer **Computational grid section** now can be built not only on **Planes**, but also on surfaces of other geometric **Objects**.
- Exporting images from **Pre-Pre-Postprocessor** is now available in the format `png`, which requires much less disk space than the `bmp` format.

Appearance and user interface

- Icons of *FlowVision* modules have been redesigned. Now it is simple to select the required executable from the **Start** menu or from a folder by the color of its icon.
- New features for the layout of toolbars and windows have been added. Now it is possible to arrange the panels and windows in any desired layout and automatic hide is also available.
- Turbulence models are now presented in the program's interface by their convenient mnemonic names (for example, names of kε turbulence models start with "KE").

Test 64-bit versions of programs

- 64-bit versions of **Pre-Pre-Postprocessor** and **Viewer** have been implemented. These versions are included into the distribution pack for the purposes of testing and collecting reports from users to our technical support service. Stable operation of these modules is not be guaranteed, the modules are not mentioned in the documentation and the installer. The 64-bit modules allow you to work with complex geometries and, in some cases, with complex visualizations.

Balancing of the computing power

- Balancing of the computing power can significantly accelerate multiprocessor computations due to redistribution of cells with overloaded processors to underloaded during in the process of computation. The balancing is configured in the advanced settings in **Solver** in the group of settings **Dynamic balance** (see Element «Advanced settings»).

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.16 What's new in earlier versions of FlowVision

See sections:

- What's new in FlowVision 3.08.05
- What's new in FlowVision 3.08.04
- What's new in FlowVision 3.08.03
- What's new in FlowVision 3.08.02
- What's new in FlowVision 3.08.01

List of changes common for all versions see in file **`ReleaseNotes_eng.txt`**.

2.16.1 What's new in FlowVision 3.08.05

Important changes

- Libraries **`libFvMPI.so`** in this release are not compatible with the libraries from any other releases, including *3.09.xx*. Also the list of *MPI*s, which are supported on *Linux*, has changed.
- *MPI*, which is included into the distribution pack, has changed.
- Since *FlowVision 3.08.05*, the distribution pack for *Linux* comes with *Intel MPI* instead of *OpenMPI*.
- By default, in the directory **`lib64`** the library **`libFvMPI.so`** is installed to work with *IntelMPI*, which is installed into the a directory **`impi`**.
- A new stable computational scheme, **Implicit new**, has been implemented. This scheme allows simulating supersonic flows with a large time steps (CFL > 10). The scheme is applicable to all problems, it has high stability and considerably faster than the old implicit scheme. Using the new implicit scheme is set in the **Properties** window of the **Simulation controls > Advanced settings** element in the project tree (in **Solver** tab), in the parameter **Type of scheme**. See details in section Element «Advanced settings».

New

- Two new operators have been added into the Formula Editor: gradient and rotor.
- A new *special model of the gap* can be used for simulating screw compressors, see section Gap model. This model requires obtaining a separate license.
- A new Low-Re k-ε turbulence model *KOLOKOL* has been implemented. This model allows simulating the laminar-turbulent transition.

See the full list of changes in the file **`ReleaseNotes_eng.txt`**.

2.16.2 What's new in FlowVision 3.08.04

Important changes

The following important changes in **Solver** have been implemented:

- Support for the supersonic outlet mode through the **Inlet/Outlet** boundary condition has been added when **Velocity = Total pressure** is selected in the **Properties** window.
- Settings have been added, which tune logging level for the data recorded in the **`log`**-file of the project (this allows you to avoid excessive logging and not reduce the calculation's performance due to too detailed logging).

New

New features in **Solver**:

- Output of the **Maximum Gauss** value into the information window and into the **`glo`**-file of the layer **Computational grid** in the volume has been implemented. Also information is displayed about coordinates of the cell, where this value is achieved. If **Maximum Gauss** is greater than 10^{-7}, the corresponding cell, has been probably built with an error and it is necessary to fix the cause of the error.
- Now it is possible to specify a pause for *MPI*-processes in the standby mode. The **SleepMPIProcess** parameter is added into the file **`FvSolver.cfg`** and its default value is 10ms.
- Improvements have been made in turbulence models and in wall functions.
- Computation time per iteration has been significantly reduced for problems, which has computation time of one iteration less than 10 seconds.

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.16.3 What's new in FlowVision 3.08.03

Important changes

For all **Physical processes** the characteristic time of diffusion transfer is taken into account in calculation of the explicit diffusion time step.

New

New models of physical processes

The algebraic Smagorinsky turbulence model has been implemented.

New settings

Now it is possible to select a method of calculating the pressure gradient near the surface, which boundary condition is **Wall**.

New schemes for integrating

New **Explicit** and **Explicit supersonic** schemes for integrating the equations in time have been implemented.

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.16.4 What's new in FlowVision 3.08.02

Important changes

Before installing and using *FlowVision 3.08.02* ensure that you have read this section.

The law of motion of a Moving body

The law of motion **Moving body** has been changed. Some new parameters have been added into the user interface. When working with a project, which was prepared in previous versions of the program, you have to check the correctness of how correctly the laws of motion of **Moving bodies** have been specified.

Severe error in the documentation of previous versions of the program

If you continue your work with previous versions *FlowVision*, keep in mind that there the values V_o and w_o are added, respectively, to the translational and rotational speeds of a body at each time step. Make appropriate changes into your projects with **Moving bodies**, on which hydro forces and hydro torques act.

Changes in the user interface of Preprocessor

Now it is possible to move geometrical **Objects** of **Preprocessor** using a mouse in the **Object editing mode** (similarly as it is possible in **Postprocessor**).

Changes in limitations of Moving bodies' motion

The **Body above plane** limitation for movement of **Moving bodies** now works with saving the tangential component of the velocity. In previous versions, the **Moving body** stopped any its movement if the acting forces had to move the **Moving bodies** beyond the limiting **Plane**. Now, if the body falls on the plane, it will slide on the plane.

Work of limitations of other types did not change: a **Moving body** stops if it bumps into a limitation.

New

Formats for downloaded and uploaded geometries

1. The ability to import geometry in the format *CEL/VRT* (*StarCD*) has been added.
2. The ability to export geometry in the format *VTK* has been added.

New models of physical processes

1. The Herschel-Bulkley's non-Newtonian model of the fluid's motion has been implemented.
2. The *EDC* combustion model has been implemented.

New features of multiprocessor computing

1. Running on an arbitrary number of processors (in previous versions only N^2 was allowed as the number of processors)
2. Running on *Linux* in the multiprocessor mode using the following implementations of *MPI*:
   a. *Platform MPI 8*
   b. *Intel MPI 3*

Improved abilities of command and batch modes for running Solver

Now it is possible to transfer parameters into command files of **Solver**.

New abilities for data visualization and calculation of Characteristics

1. The ability to build the **Vectors** layer in a moving coordinate system has been added.
2. The ability to calculate **Characteristics** by user variables and by coordinates has been added.
3. The ability to build the center of **Characteristics** as a formula has been added.

New variables and ability to specify them

1. Now physical variables can be specified by tables.
2. The **Explicit time step** variable has been added.

New parameters of Solver

1. A new stopping condition has been added, which stops the computation when the time step becomes less than the specified.
2. The ability to correct touching surfaces in *FSI* problems (which are solved in conjunction with the *Abaqus* software) has been added.
3. The ability to export loads on a selected **Subregion** has been added.

Communication with other packages

Connection with the *LMS* software package has been added.

See the full list of changes in the file `ReleaseNotes_eng.txt`.

2.16.5 What's new in FlowVision 3.08.01

Important changes

Before installing and using *FlowVision 3.08.01* ensure that you have read this section.

Installation

When upgrading from *3.07* to *3.08* the conversion settings is required.

User directory

Starting from with version *3.08*, one user directory is used for all server modules and another user directory is used for client modules (these directories store licenses, information about users of **Solver-Agent**, `cfg` and `log` files). The directories are specified during the installation. It is recommended to make a backup copy of these *FlowVision*'s user directories.

Settings of Solver-Agent

In version *3.08*, all executable files are located in the installation directory next to each other, so the relative path from **Solver-Agent** to **Solver** has changed. In settings of **Solver-Agent** for all command lines of starting **Solver** it is required to check and, if required, correct the path to **Solver**.

Conversion of projects

After conversion of a project, the following changes will take place, visible in the user in interface:

- A volume heat source in a **Subregion** will be converted into a **Volume heat source** modifier.
- Ability to chose a schema type will disappear: all calculations will be performed by the implicit scheme.

Turbulence parameters

The meaning of the **Pulsations** parameter has changed in **Initial conditions** and **Boundary conditions**. When using projects that have been prepared in previous versions of the program, you need to change setting of this parameter.

Boundary condition Inlet/Outlet

Meaning of the **Boundary conditions "Velocity with pressure"** and "**Normal velocity with pressure**" for the variable **Velocity** has been changed: now these boundary conditions work according a new algorithm.

The thermal conductivity of a gap

Now, this parameter is called **Gap heat-transfer coef.**. Calculation of the of thermal conductivity coefficient in a **Gap** is done for non-negative values of this parameter on the gap-forming walls.

License Manager

A new possibility to administrate and monitor the existing license options has been added. You cannot use *FlowVision 3.xx* with **License Manager** from versions *FlowVision* 2.xx. For normal operation of these two versions of *FlowVision*, replace **License Manager** from *FlowVision 2.xx* with **License Manager** from *FlowVision 3.08*.

Visualization

Names of some variables changed. Now the names are:

TemperatureTotal = $T_{\text{tot, abs}}$-T_{ref}

PressureTotal
EnthalpyTotal
TurbViscosity

Now the variables' names have the following meanings:

- **Prandtl** is molecular Prandtl number
- **PrandtlTurb** is turbulent Prandtl number
- **Schmidt** is molecular Schmidt number
- **ShmidtTurb** is turbulent Schmidt number

New

Formats of loaded geometry

No the program can load the geometry in the *NGEOM* format.

Tools for changing the geometry

1. The ability to create the geometry of the computational domain and **Moving bodies** based on **Preprocessor**'s **Objects** has been added.
2. The following operations for editing the geometry of **Region** have been added:
   a. Adding geometries of **Subregions** after creating the project.
   b. Equidistant offset of the geometry.
   c. Removing too small facets.
   d. Correction of self-intersections.
3. The following operations for editing the geometry of **Imported objects** have been added:
   a. Splitting an object into separate objects (if the object consists of several surfaces).
   b. Equidistant offset of the geometry.
   c. Correction of self-intersections.
4. A new group of elements, **Surfaces**, has been added into the **Geometry** folder added. They correspond to individual surfaces of the computational domain and **Moving bodies**. The following operations for editing the individual surfaces of the geometry are available:
   a. Removing a surface.
   b. Replacing a surface.
   c. Adding a surface.
   d. Assigning a boundary conditions on a surface.
   e. Regrouping a surface.
   f. Transforming a surface.
   g. Export of a surface.

New models of physical processes

1. **Mixing+Chemistry**
2. **Combustion**
3. **Radiation**

New features in the models of the physical processes

1. Ability to disable the equations for some physical processes (that can be useful for debugging).
2. Anisotropic thermal conductivity for the **Heat transfer** process.
3. AKN and S&S models of turbulent heat transfer in the **Turbulence** process.
4. Non-Newtonian Herschel-Buckley's model of motion in the **Motion** process.

User variables

A new type of **User variables** has been added: **Constants**.

Functional dependencies

Abilities have been added to specify functional dependencies for:

1. **Initial data** (see the section Folder «Models»)
2. The **Gravity vector** (see the section Element «General settings»)

Boundary conditions

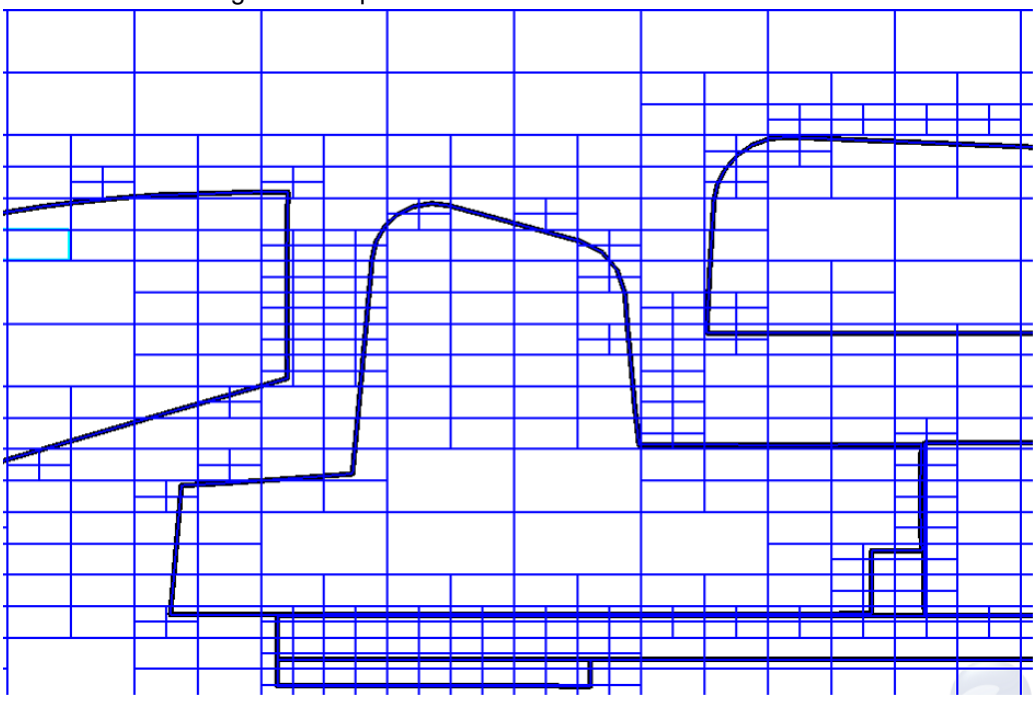
The boundary condition **External heat exchange** allows us to calculate the variable temperature of the outer wall, which is determined by the temperature of the simulated environment, by the specified temperature of the environment, and by the specified heat transfer coefficient.

New Modifiers

- **Volume heat source**
- **Setting variable**
- **Volume force**
- **Resistance**
- **Ignition/extinction zone**

Automatic grid

A new criterion for adaptation has been added, **Improve** (see the section Folder «Adaptation»), which allows automatic generation of the computational grid in thin-walled assemblies and in models with complex geometry, where resolution of narrow subregions is required.



Moving body

1. Limitation of **Moving body**'s motion by two degrees of freedom: along and around a specified axis.
2. Carrying out a calculation inside a **Moving body**
3. Replacing the geometry of a **Moving body** and continuation of the computation

4. Elimination of fluid mixing from different sides of a thin **Moving body** (the size of which in some direction is less than the cell size)
5. New methods of specifying the initial rotation of a **Moving body**

Boundary links

1. Automatic creation of **Binders** for all **Free BCs** (free boundary conditions)
2. Automatic detection of **Binders** corresponding to **Boundary Condition**
3. Removal a **Boundary condition** when **Boundary links** exist
4. Copying a **Linked** boundary condition

Characteristics

1. Building **Characteristics** in enclaves
2. Advanced capabilities for work with **Characteristics**, which are built by vector variables

Parameters of calculation

1. The ability to limit the minimum time step has been added
2. The dynamic criterion for smallness of cells has been added
3. Emulation of 128-bit representation of numbers has been added

Visualization

1. In the **Palette** group of parameters (which is presented in properties of **Layers**), parameters **Appearance** have been added that control displaying of the **Palette**'s legend directly in the **View** window (see the section General properties of Layers).
2. Ability of specifying default values for all parameters of the **Solids** layer has been added
3. A new layer, **Distributed characteristics**, has been added
4. Ability of specifying **Materials** for visualization of surfaces has been added
5. Ability of building a **Vectors** layer in a moving coordinate system has been added

Viewer

1. The ability to load/save the scene has been added
2. The ability to save images of the specified scene using a command line has been added
3. The ability to display the coordinate system has been added

Communication with other software packages

1. Connection with IOSO using the *FlowVision*'s interface has been implemented:
   a. Specification of parameters and optimization criteria in the *FlowVision*'s project
   b. Starting *IOSO* from **Postprocessor**
2. Connection with TORT has been implemented via text files.

See the full list of changes in the file `ReleaseNotes_eng.txt`.

3 Known limitations

FlowVision is a complex software product, which constantly changes and is improved daily. In this section the most significant limitations are listed, which will be fixed in one of future releases.

Code(s)	Limitation	Workaround
#900, #4222, #1454, #1465, #1453	Surface tension is simulated incorrectly when two phases interact with the phase interface surface (this limitation is not applied to vacuum).	It is recommended not to use the surface tension.
#1366	If you run the **Solver** to continue the computation while **Moving body > Activation > Type = Inactive** has been specified, then this **Moving body** will not be removed from the computational domain.	*Workaround 1:* Start calculation from the beginning (in this case the **Moving body** will be removed). *Workaround 2:* Create another **Moving body** modifier (in this case inactive **Moving body** will also be removed from the computational domain).
	Solver stops responding the commands when building the **Streamlines** layer. The problem arises when a streamline bumps on the surface on which the boundary condition **Wall** is set at very low speeds in near-border cells.	Temporary solution is not available, the problem will be solved in future versions of *FlowVision*. *Recommendations:* a) do not create the **Streamlines** layer before the end of the computation b) use the **Streamlines** layer only for the final analysis of the results c) sometimes changing of the number of streamlines or moving their **Emitter** to another position helps.
-	When you load a saved project with another number of processors, values of flows of the values on **Characteristics** are not displayed (their values are displayed as "**(none)**").	This limitation appeared because of the new version of the program supports changing of data formats. To display the values of flows it is necessary to run the computation for continue and make at least one iteration.
-	In visualization of a rotor with a sliding surface, the not-moving part of the 3D model outside the rotating subregion is also visualized with rotation. This visualization effect appears if before the inserting of the sliding surface the moving and not-moving parts are combined in the same surface. The solution, in distinction from the visualization, is correct and the not-moving subregion does not move.	A workaround of this visualization issue you can separate a rotor a not-moving part in the CAD system and build the sliding surface around the rotor only, do not touching the not-moving part.
	Substantial slow-down of project, which have been saved in old version of the program (30905 and older versions), that have numerous objects with adaptation.	Open the project in a new version of the program and specify settings of **Adaptation**, **Adaptation by condition**, and **Adaptation to solution**, and then save the project.
	Incomplete converting of all projects containing several objects with adaptation.	

4 Installation, setting up, and administration

See sections:

- General system requirements
- FlowVision's architecture
- Versions of *FlowVision*
- FlowVision technical support
- Installation of *FlowVision* (under *Windows* and under *Linux*)
- Recovery of damaged installation and/or changing the list of installed modules
- Setting up
- Administration

4.1 General system requirements

Operating systems

Only 64-bit versions of operating systems are supported:

- *Linux*: the program's operation is tested on *CentOS 6.10* and *CentOS 7.9*. Fully functional and stable operation in other *Linux* versions is possible, but not guaranteed. Performance on other *Linux* versions is not guaranteed (it is not tested), but it is known that *FlowVision* effectively operates in a many of various *Linux* versions, including *Ubuntu, Open SUSE*, *RedHat,* etc. The program's operation in versions of *Alt Linux* and in *CentOS 7* is not only supported but also tested. See also sections Installation on Linux and Specific system requirements for Linux.
- *Windows HPC Server 2008*
- *Windows Server 2008*
- *Windows Vista*
- *Windows 7*
- *Windows 10*

FlowVision is not tested on *Windows 8* and *Windows 8.1*, and operation of *FlowVision* on these versions of *Windows* is not guaranteed. However, it is known that *FlowVision* is successfully used on these operating systems. Since version *3.11.01*, operation of *FlowVision* on *Windows 8* requires installation of the update *Windows #2999226,* see https://support.microsoft.com/en-us/help/2999226.

System requirements of all*FlowVision*components except Pre-Postprocessor and Solver

Device	Minimal system requirements
Network card	Any card supporting *TCP/IP v.4* and/or *v.6*
Processor	*x86*-architecture: • *Intel Core i3* • *AMD Ryzen 3* or with higher performance
RAM	4 GB
Video accelerator	Any video card

System requirements of Pre-Postprocessor

Device	Minimal system requirements	Recommended system requirements
Processor	*Intel Core i3* *AMD Ryzen 3* or faster *x86*-architecture processor	*Core i7, AMD Ryzen 5* or *7* or faster *x86*-architecture processor
RAM	4 GB	12 GB
Network card	100 Mbit/s with*TCP/IP v.4/6*	1 Gbit/s with *TCP/IP v.4/6*
Video accelerator	Any video card, preferably a card on *GeForce* (256 Mb) chips*)	Video cards with support of *Shader Model 2* and *OpenGL 3.0* (more then 1500 MB). It is recommended to use: • discrete video adapters *NVidia* and *AMD* • *OpenGL* versions *4.5* or higher (use of *OpenGL* versions *<4.5* some functionality of post-processing might be limited).

*) Support of *OpenGL* is required for operation of **Pre-Postprocessor**. Professional video adapters *Nvidia Quadro*, which support hardware acceleration in *Windows Remote Desktop*, are recommended for work with **Pre-Postprocessor** via *Windows Remote Desktop*.

System requirements of Solver (workstation based on a personal computer)

Device	Minimal system requirements	Recommended system requirements
Network card	100 Mbit/s with support of *TCP/IP v.4/6*	1000 Mbit/s or better, support of *TCP/IP v.4/6*
RAM	8 GB	From 2 to 4 GB per core*)
Processor	*Intel Core i5* *AMD Ryzen 5* or faster *x86*-architecture processor	*Intel Core i7*, *i9* *AMD Ryzen 5* (high-end modifications), *7*, *9* or faster *x86*-architecture processor
Video accelerator	is not required for **Solver**	

*) See *Recommendations for amount of RAM*.

Recommended configuration for work of Solver on a high-performance cluster system

Device	Recommended system requirements
Interconnect	It is recommended that a combination of *Infiniband* and *Ethernet, 1 Gb / s and faster (Ethernet*for*TCP / IP* v.4/6) or analogues
RAM	From 2 to 4 GB per core *)
Video accelerator	is not required for **Solver**

*) See *Recommendations for amount of RAM*.

Recommendations for amount of RAM

To estimate requirements for RAM, you can use the ratio: 200 ... 300 thousand of cells require about 1 GB of RAM (depending on the bitness of **Solver** and complexity of the problem).

Best scalability*) is achieved when on core is provided to 50 ... 100 thousand cells. Scalability depends on many factors such as: number of memory channels, bandwidth of memory buses, processors's architecture, type of interconnection, dimension of the problem.

The bigger number of RAM memory channels, and the higher frequency of the memory, the speed of simulation is usually faster.

*) *Scalability* is the ability of the system to increase its capacity when resources are added; ideally, increase the number of processors twice should reduce the calculation time twice.

Software

For *Windows*:

- *Microsoft MPI (MS MPI)* of version 7 or more recent version. This is required to run **Solver** in multiprocessor mode. The installation is to be done as a user with Administrator's permissions.
- *Microsoft Visual C++ Redistributable 2015-2019* (it is included into the archive with the distribution pack and it is installed automatically)

For *Linux*:

To run programs with a graphical interface, *Xserver* is required.

Amount of free space on your hard disk

Depending on the components selected, installation and operation of the software requires up to 300 MB of free space on your hard disk.

When you save the results of the computation it may require a significant amount of free hard disk space (depending on the dimension of the problem). To estimate the disk space required for one saved iteration, you can use the following ration: 100,000 cells require about 50 MB of disk space.

Names of computers

If names of computers, on which **Solver-Agent** and **License Manager** modules are installed, contain non-Latin letters (for example, Cyrillic or Greek ones), then access to these computers is only possible by their IP addresses (not by their names).

Operation of the *MS MPI* library requires that the name of the computer, on which the library is installed, contains only Latin characters (**a**...**z**, **A** ... **Z**), digits (**0**, **1** ... **9**), minus sign (**-**) or underline sign (_). If the computer name contains non-Latin-alphabetic language letters (for example, Cyrillic or Greek letters), it would be impossible to run **Solver** in multiprocessor mode with use of *MS MPI*.

Specific system requirements for installation on *Linux*

See section Installation on Linux.

4.2 FlowVision's architecture

This section describes location of *FlowVision* components on computers and inter-operation of components, computers and data files.

See sections:

- Modular architecture of FlowVision
- FlowVision modules and their purposes
- Variants of FlowVision's deployment
- Network inter-operation between modules

4.2.1 Modular structure of FlowVision

The *FlowVision* software complex contains several modules and utilities. *FlowVision*'s modules can inter-operate via a network.

The modular structure of *FlowVision* allows effective use of computational resources of your organization and provides embedding into various organizational structures.

User's and computing computers

We will make a distinction between two types of computers: *user* and *computing* ones.

The user performs project preparation and analysis of results on a *user computer*.

The user performs computations on a *computing computer*.

There may be several user and computing computers. The same computer can be both user and computing one.

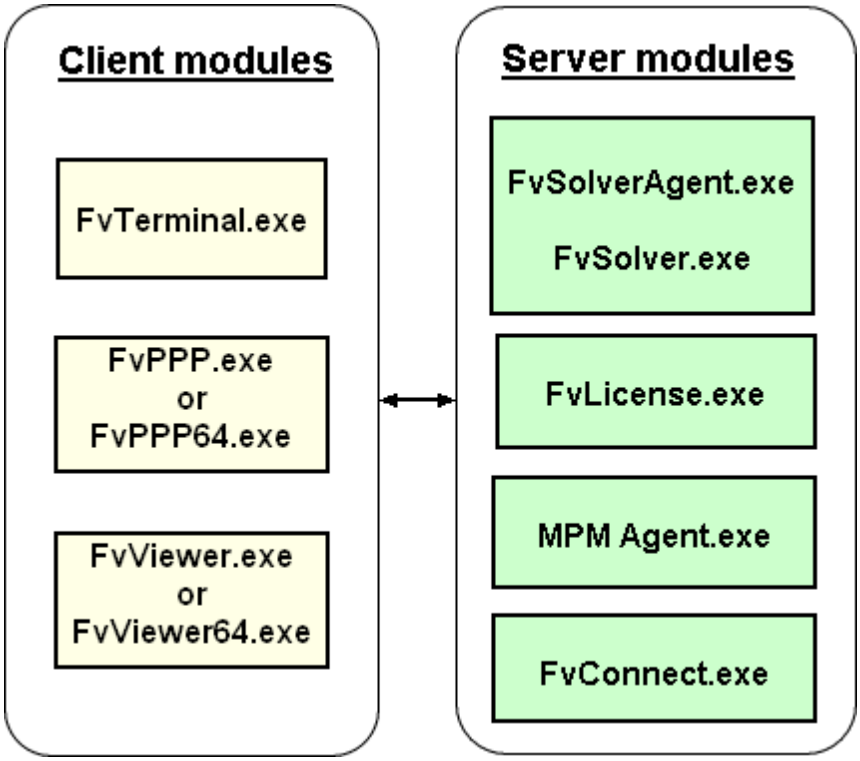
Single-processor and multiprocessor computers, clusters

Generally, computation is performed on several computers united into a network. A computer in the network may be a *single-processor*, a *multiprocessor* or a *cluster*.

A *multiprocessor* computer has more than one processor.

A *cluster* consists of several computers united into a single processing unit; several computation processes may be running on them simultaneously. One of the computers of the cluster is the *controller* (*FlowVision*'s server modules are installed on it). In addition, the **License Manager** can be installed on any computer accessible to the cluster. Usually it is installed on the *cluster controller computer*.

4.2.2 FlowVision modules and their purposes



All *FlowVision* modules may be subdivided conveniently into two groups: *client modules* and *server modules*. Modules from both groups can be installed on a same computer or on different computers. See also section FlowVision modules.

It is not possible to establish a connection between *FlowVision* modules with different version numbers (except **License Manager (FvLicense.exe)**, which has its own numbering system for its versions).

FlowVision's client modules

A *client module* is a program that has some user interface. Users interact with these programs directly.

Client modules can be installed either on a same computer or on different computers.

FlowVision includes the following client modules:

- **Pre-Postprocessor**
- **Terminal**
- **Viewer**

All the listed above client modules are available in both operating systems *Windows* and *Linux*.

Module	Description
Pre-Postprocessor (`FvPPP.exe`)	Provides user and license registration, creation of projects, starting project's computation and viewing calculation results.
Terminal (`FvTerminal.exe`)	Provides management of project and **Solvers**, user and license registration. Does not provide editing projects or viewing of calculation results.
Viewer (`FvViewer.exe`)	Allows the user to view results of computation or watch the simulation results during the computation process. Only the visualization layers are visualized, which were previously created in **Pre-Postprocessor**.

FlowVision's server modules

The *server modules* have no graphical user interface (GUI). The user does not work with these modules directly, but interacts through *client modules*.

Since server modules do not have a user interface, no dialog box opens when they are started.

To see whether a server module is running or not, it is necessary to find it in the process list.

The *server modules* include the following modules: two interdependent (**Solver-Agent** and **Solver**), two independent (**License Manager** and **MPM-Agent**), and one service module (**Retranslator**).

The **Solver-Agent** and **Solver** modulus are always installed on the same computer.

The **License Manager** may be installed on the same computer or on any other computer to which all running *FlowVision* modules have constant network access.

MPM-Agent must be installed on the same computer as *Abaqus*.

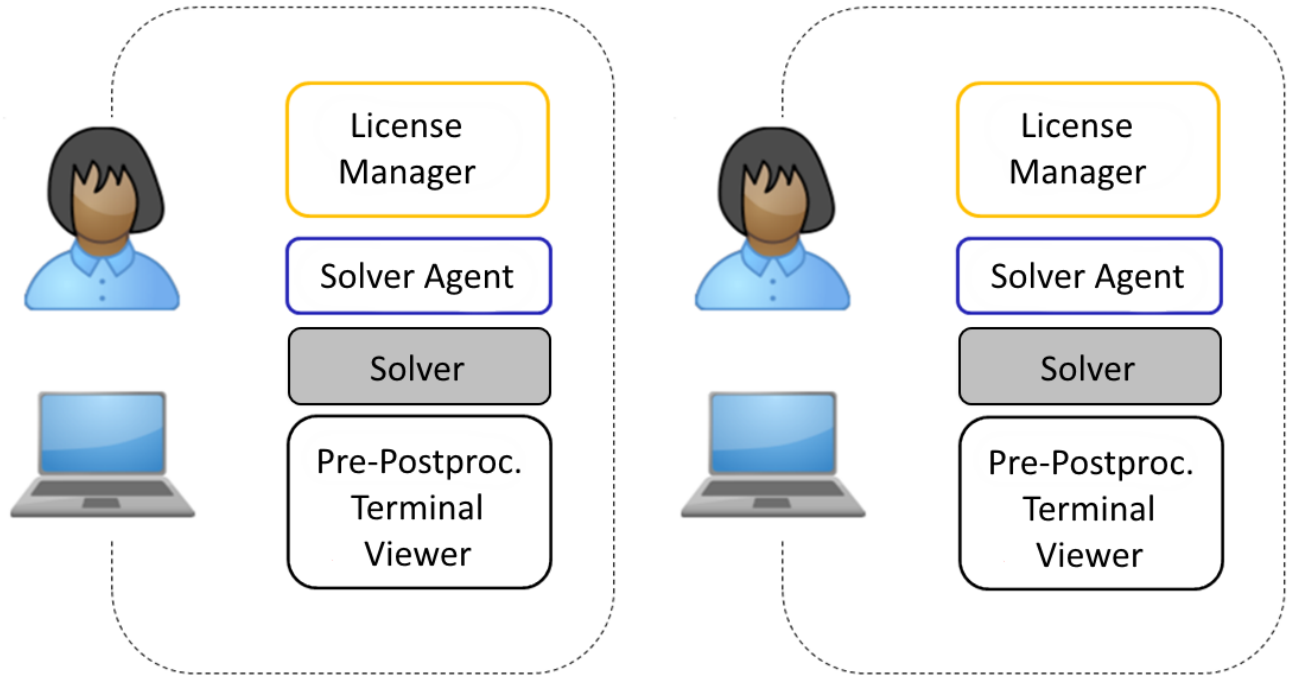
Retranslator is always installed on all computers where at least one other *FlowVision* module is installed.

Module	Description
Solver (file `FvSolver64.exe` in *Windows*, `FvSolver64` in *Linux*)	Performs computations.
Solver-Agent (file `FvSolverAgent.exe` in *Windows*, `FvSolverAgent` in *Linux*)	Performs loading/unloading of solvers and sends information about solvers running and existing projects in the server directory to client modules.
License Manager (file `FvLicense.exe` in *Windows*, `FvLicense` in *Linux*)	Manages licenses.
MPM-Agent (file `MpmAgent.exe` in *Windows*, `MpmAgent` in *Linux*)	Starts computation in *Abaqus*.
Retranslator (file `FvConnect.exe` in *Windows*, `FvConnect` in *Linux*)	It provides data exchange between *FlowVision* modules when they locate in different address spaces. Actually, **Retranslator** is a software proxy server.

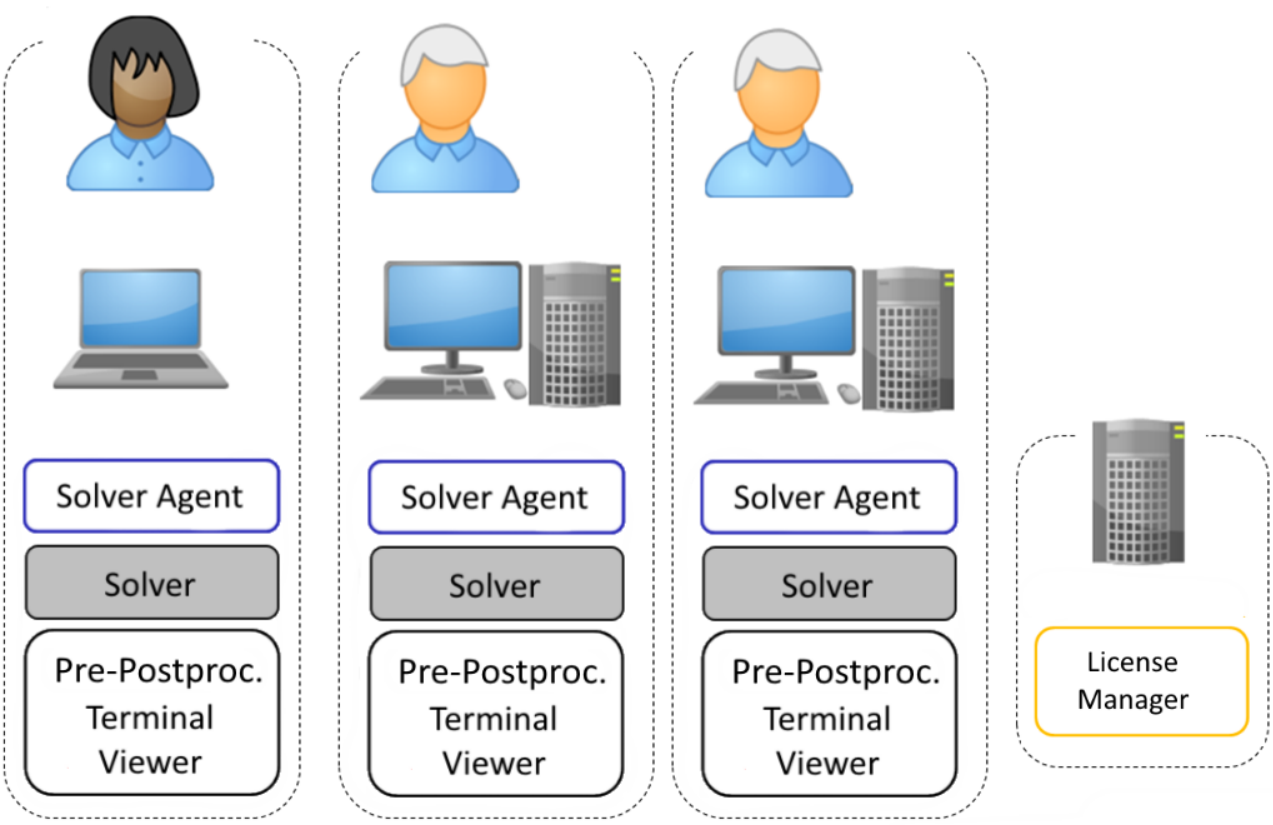
4.2.3 Variants of FlowVision's deployment

FlowVision has a modular structure and allows you to implement various schemes of operation in your organization.

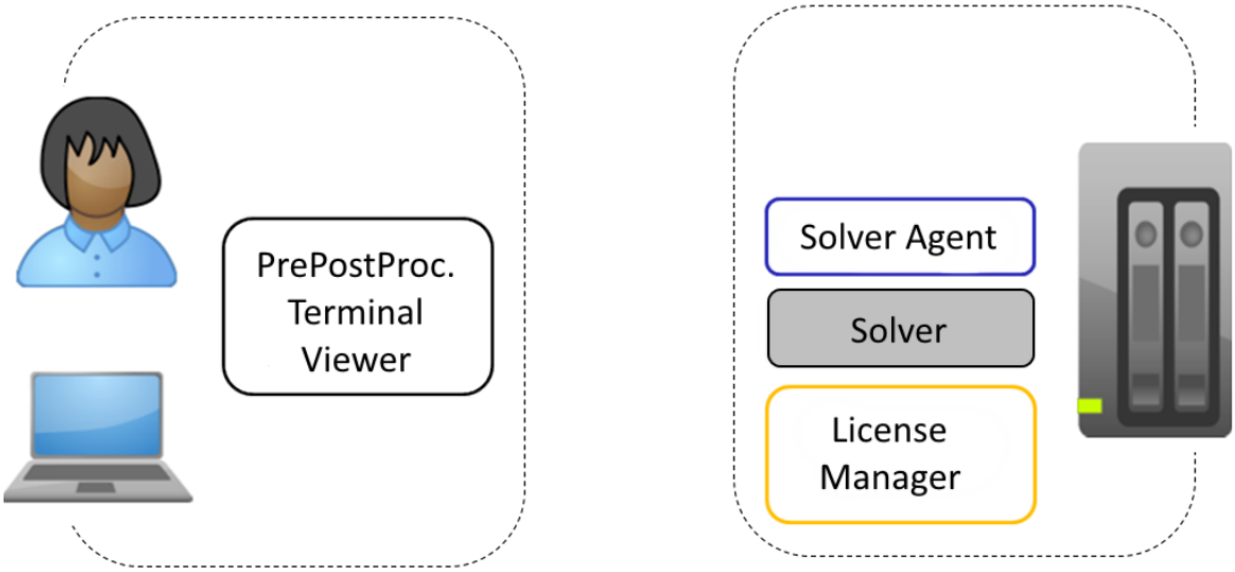
The illustrations below schematically show possible *FlowVision*'s deployment scenarios:



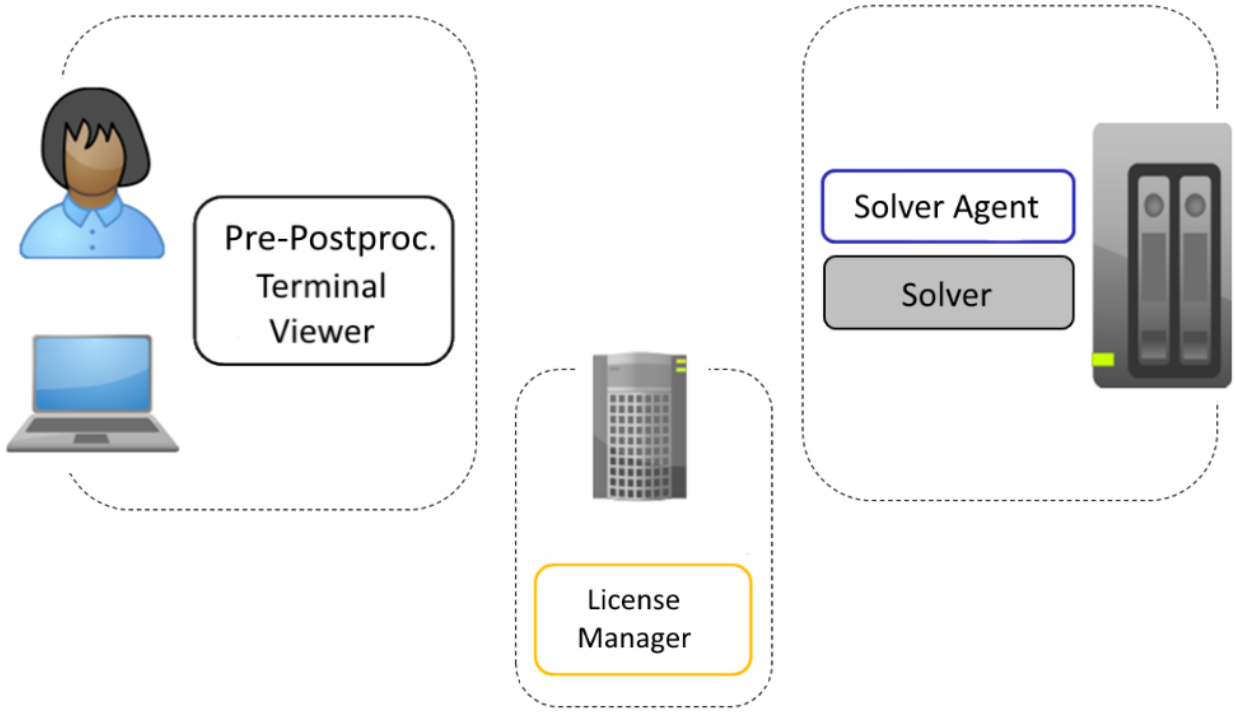
Variant 1. *FlowVision* is entirely installed on individual workstations.



Variant 2. **License Manager** operates on a remote computer.



Variant 3. An individual remote server is allocated for performing the calculations and for **License Manager**.



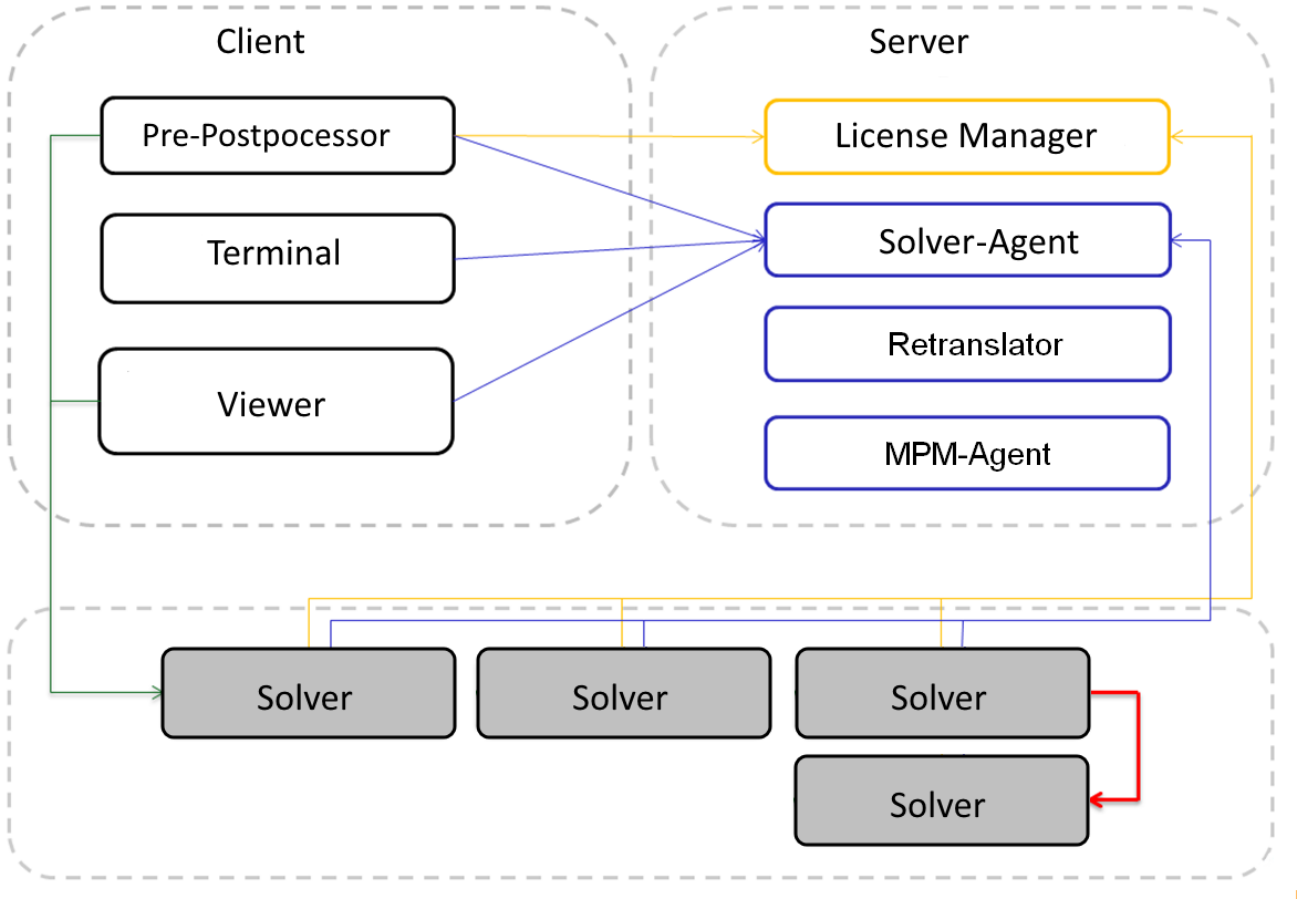
Variant 4. An individual workstation is allocated for calculations, and the third computer is allocated for **License Manager**.

Other deployment variants are possible. Variants 3 and 4 are suitable for deployment FlowVision on clusters.

4.2.4 Network inter-operation between modules

Client *FlowVision* modules are usually located on the user's computer, while server modules (including **Solvers**) are usually located on computing clusters and servers.

Below is a diagram of network inter-operations between *FlowVision* modules:



Comments to the diagram:

- Blue arrows present requests to **Solver-Agent**.
- Yellow arrows present requests to **License Manager**.
- Red arrow presents inter-operation between **Solvers** via *MPI* at multiprocessor operation.
- Green arrow presents a direct connection between **Pre-Postprocessor** (or **Viewer**) and **Solver**.

Solver-Agent controls **Solvers**. That is the module, to which all client modules send requests when they need to receive the list of **Solvers**, or need to run a **Solver**, etc. After receiving from **Solver-Agent** all necessary information about a **Solver**, client modules interacts with this **Solver** directly.

Each **Solver** is both a server module (client modules connect to it) and a client module (it connects to **Solver-Agent** to report about its existence and to **License Manager**, to get permission to compute). **Solvers**, which compute a same problem in parallel mode, communicate with each other via *MPI*.

Also **Retranslator** might play special role in the network inter-operation, see section Use of Retranslator.

4.3 Versions of FlowVision

Format of the version number

FlowVision version number has the following format:

version 3.XX.XX build YYYYMMDD

where:

- **3.XX.XX** is the version number
- **YYYYMMDD** is the build date

Each program in *FlowVision* can have its own version.

Starting from *FlowVision 3.09.01*, **License Manager** has its own distributive pack and version numbering system independent from the main distributive pack. **License Manager** updates occur much less frequently then updates of the main distributive pack.

How to find the version number

There are several ways to find the version number of the program:

1. In **Configurator**, in the tab General, click **Versions**.
2. Run the program in the console with the parameter `-v`, for example, `FvSolver.exe -v`. As a result, you will see information about the program's version and about the place, where it looks for its settings.
3. If *FlowVision* is installed under *Linux*, run the file `GetVersions` in the installation directory of *FlowVision*. After that the information about the version will be output on the console.

If the components of *FlowVision* are installed on several computers, do these steps on each of them.

Since version *3.09* it is not required to remove old versions of *FlowVision* from the computer before installing a new version.

4.4 Technical support

About the technical support service

The technical support is a paid service. Technical support allows you to download and use updates of *FlowVision* and consult with qualified experts if errors in the *FlowVision* software complex were found.

Engineers from the technical support service will analyze errors in the program and, when possible, help you to use a workaround, which allows you to continue the work until a new version is released where the error is fixed.

The technical support service also support the communication between users and developers. Send to us your wishes and suggestions concerning improvements of user interface, operation and expanding the program's functionality.

Responsibility of the technical support service does NOT include:

- development of methodology of solving your computational problem
- analysis of the quality of the formulation of your computational problem
- other consultations relating to the formulation of your computational problem
- solving your computational problem

Detailed information about the provision of technical support services, information on how to access the technical support, FAQs, etc. can be found at https://flowvisioncfd.com/en/support-page-en.

Contacts of the technical support service

For written requests to the technical support service, use email address support@flowvisioncfd.com.

How your requests are processed in the technical support service

When a user sends an email to the technical support service, his message goes into an automated system, which is based on the powerful open software *OTRS* (www.otrs.org). This system guarantees answering on users' messages and provides controlled time of response and quality of the support.

Immediately after your email comes into the system, an answer on it will be sent informing you that your mail has been received and it is being processed. If you didn't receive an answer on your email within 30 minutes, this means that your email might be not delivered to the technical support. In such situation make a telephone call and ask about the status of your request.

After this automatic answer is sent, an engineer of the technical support service will contact you usually within a day and will tell you a decision or ask additional questions for better diagnostics.

During further communication do not change subject of the email received from the technical support service engineer.

If you wish to create a new request, which has no concern to the previously asked request, send an email with different subject (do not include the new request into communication concerning to another request).

How to reduce the time required for fixing the error

Unfortunately, many requests to the technical support service appear mostly because of incorrect use of the *FlowVision* software complex. Your preliminary analysis of the problem would help to substantially reduce the time required to fix the error.

So, before your request to the technical support service, please follow the steps:

- **If the error relates to a specific computational project or to Solver:**
  - Make sure that your setting of the computational problem corresponds to the domain of applicability of mathematical models and numerical methods implemented in *FlowVision*. Domains of applicability of models (for example, the gap model or model of the boundary condition **Inlet/Outlet**), if they are limited, are described in chapter Theory. When you use such commonly used physical abstractions as perfect gas, ideal liquid, etc., then you should follow limitations, which are based on these physical abstractions.
  - Make sure that your geometry model conforms to system requirements of *FlowVision*.
  - If correction of the found defects did not fix the error, provide the technical support service with:
    - `log` and `err`-files from the server part of the project
    - client part of the project
    - detailed description of the error

- **If the error consists of unsatisfactory accuracy of the computation:**
  - Research the convergence of the solution by the computational grid.
  - Make sure that distance to boundary conditions does no influence on the solution
  - Research the convergence depending the time step.
  - If removing the found defects did not fix the error, then provide the technical support service with:
    - description of your research of the convergence of the solution by the computational grid including illustration of dependency of the key characteristics from the size of the computational grid and detailed description or illustration of examined computational grids
    - description of your research of the convergence by the time step including illustration of dependency of the key characteristics from the time step
    - client part of the project
    - information about the reference data, which does not comply to the *FlowVision* results (quotation from scientific works along with references on the works, or detailed description of the experiment and its results including accuracy evaluation)

- **If the error is connected with work of Solver, Pre-Postprocessor, License Manager, Solver-Agent, or other module and does not depend on the project:**
  - Turn on the **DebugLog** parameter (see section Parameters in configuration files) for the module, which relates to the error.
  - Restart the module.
  - Reproduce the error.
  - Create a diagnostics archive.
  - Send this diagnostics archive to the technical support service.
  - Send to the technical support service a detailed description of your actions, that cause the error. It is necessary that you inform the technical support service about the module(s), which relates to your actions on each step.
  - Send to the technical support service the exact texts of error messages or (which is preferable) screenshots of the error messages. Specify a software module, which generates the error message.

- **General recommendations:**
  - Examine all required log files of the software complex (see subsection *"Log files of FlowVision modules"* below).
  - Try to find information about fixing the error in the user manual. Try the search by the texts of error messages or key words relating to the error.
  - Always specify the exact version of the program including the version number and the build date.

Analysis of *FlowVision* logs

There are several types of *FlowVision*'s logs:

- **`log`**-files of *FlowVision*'s modules
- log files of the project's computation
- standard output of the **Solver**

Log files of *FlowVision* modules

All logs are stored in the user directory **`logs`**, which locates near the directory with settings of the appropriate module (see section User directories).

Examine the contents of the log files of the module, which relates to the error. The interesting lines are those, that have time stamps corresponding to the time of arising the error.

In general, description of the error is human-readable. Errors generally are indicated by the word "**`Error`**".

If you can not understand the error's description in the log, contact the technical support service, adhering to recommendations from subsection *"How to reduce the time required for fixing the error"* (see above).

Log files of the project's computation

These are **`err`**, **`log`**, and **`sta`**-files. These files are located in the directory, which contains the server part of the project.

`sta`-file contains general statistics of the work of **Solver** and does not contain error messages.

err-files contain error messages and warnings. Lines indicated by the text "**[WARNING]**" generally are not interesting. Lines indicated by the text "**[ERROR]**" generally indicate a critical error, which makes no sense the further computation. Also such lines might indicate an inappropriately built computational grid, which in most of cases means that it is no sense to continue this computation. If you can not interpret the message from an **err**-file, you should contact the technical support service.

log-file contains information about procedures, which are carried out by **Solver**. The **log** -file may contain a human-readable error message. If you can not interpret this, contact the technical support service.

Standard output stream of Solver

FlowVision modules output their data into standard output stream and error stream. The standard output stream is the console window of **Solver**, which you could see, where the report of the program's operation is visible.

It is possible to redirect this stream into a file or the console, to do so you have to start the program from the operation system's command line (ask for details your system administrator if required).

If errors are connected to start of *FlowVision* modules, but not to computation, the standard output stream often contain human-readable error messages, which might indicate, for example, absence of some libraries.

What can you do when the solution diverges. What is a divergent solution

When solving computational problems using numerical methods, users of any software may see unstable computation results, which can be caused by:

- Neglecting of the domain of applicability of mathematical models
- Bad first approximation
- Bad resolution by the computational grid
- Coarse representation of the geometry surface
- Bad computational grid, for example:
  - errors or bad quality of geometry models: self-intersections in surface grids, too big or too small facets, etc.
  - coincidence of surfaces
  - generation of isolated very small volumes caused by intersections of surfaces
  - too big difference between facets of computational cells
  - too big difference between sizes of adjacent cells
- Coarse time step

Instability of the solution shows itself in the following:

- Extremal values of physical variables appear.
- When using the **Implicit** scheme, equations for a physical process might require a greater number of iterations for convergence (more then 100). You can see the number of iterations in the **Status** tab of the **Monitor** window.
- Explicit time step has abnormally small values, which do not correspond to normal velocities (see sections Window «Monitor» and Time step).

If you encounter a bad convergence, then, your primarily should evaluate which physical process must define the time step. You should try to use an explicit time step (with convective CFL = 1) and make computation in this mode.

Then you should localize the place with abnormal velocity, pressure, temperature and density arise. Most convenient way to do this, is using characteristics calculated in the whole volume and plots displayed in the **Plot** tab of the **Monitor** window. Prepare the computational project in such a way:

1. Do computation until several time steps before anomalies appear.
2. Turn on saving the data with history at each time step.
3. Create **Characteristics** in **Preprocessor** in the whole **Computational space** by **Pressure**, **Temperature**, **Density**, and **Velocity**.
4. Create user stopping conditions to output maximal and minimal values of **Pressure**, **Temperature**, **Density**, and **Velocity**.
5. Continue the computation and view the moving of plots, which relate to the user stopping conditions.
6. When you find an abnormal increase or decrease of the extremal values of variables, stop the calculation before the anomaly spreads on the whole **Computational space**.
7. In the **Info** window of the **Characteristics** find coordinates of the cell with abnormal maximal or minimal value of a physical variable.
8. Build a plane with a reference point with coordinates found on the previous step.

9. Examine the calculation grid near the reference point using the **Computational grid section** layer. Using contours of intersection the geometry model and the plane evaluate the correctness of the geometry model near the reference point (examine if there are coincidences of surfaces, micro volumes, or self-intersections).
10. Build a **Vectors** or **Color contours** layer by the required physical variable. Disable the **Interpolation** and enable the **Double resolution** parameters of the layers.
11. Based on the collected data, make a decision concerning improvement of the geometry model, condensing the computational grid in the area of high gradients, or changing initial conditions or time step.

First experience of working with *FlowVision*

The technical support service has no enough resources to teach users to work with *FlowVision*, so education is a separate service. You can contact your sales manager to discuss providing this service to you.

The most effective way of quality and quick learn (or remember) many useful functions, is sequential, project after project, solving all exercises from our tutorial, which is included in the distributive pack of *FlowVision*. Even if you don't plan to solve the computational problems relating to some model, we recommend you to solve all exercises from the tutorial, as they can illustrate various useful and convenient universal functionality.

Links

- Useful links (https://flowvisioncfd.com/en/support-page-en#useful-external-links)
- Feedback form of the technical support (https://flowvisioncfd.com/en/contacts-en/contacts-cto-en)
- Quick menu for access to pages of the technical support (https://flowvisioncfd.com/en/support-page-en)

4.5 Installation of FlowVision

Since version *3.09*, you do NOT have to remove older versions of *FlowVision* when you install new versions.

You can install *several various versions* of *FlowVision* on your computer, this was not possible before. So when you install version *309* you do not have to remove version *308*. And when you install version *3.09.02* you do not have to remove the previously installed version *3.09.01*.

FlowVision can be installed on 64-bit computers on the following operation systems:

- *Windows*
- *Linux*

Installation directory

The *installation directory* is the directory where *FlowVision* or *FlowVision*'s **License Manager** is installed. The installation directory is specified during the installation of *FlowVision* or *FlowVision*'s **License Manager**.

By default, **License Manager** is installed into the following directory:

- `C:\Program Files\FlowVisionLM` (for *Windows*)
- `/home/user/FlowVisionLM` (for *Linux*, and "`user`" corresponds to the user's system name)

By default, other *FlowVision* modules are installed into a directory, corresponding to the version of the program and (for *Linux*) the system user's name, for example:

- `C:\Program Files\FlowVision-3.13.01` (for *Windows*)
- `/home/user/FlowVision-3.13.01` (for *Linux*)

In these directory names the numbers (`3.13.01`) correspond to the program's version number and `user` corresponds to the system name of the user.


The *installation directory* of *FlowVision* must be *available for write* by the user, who is installing *FlowVision*.

Installers

FlowVision is installed by two installer programs:

- installer for **License Manager**
- installer for other modules of *FlowVision*

When a new version of *FlowVision* is installed, you do not have to upgrade **License Manager** to the actual version when such upgrade is not explicitly specified in the documentation in the release notes. **License Manager** comes as a stand-alone installer and can be installed on another server, which is available to client computers via the network.

Installation *FlowVision* components on different computers

Different components of *FlowVision* can be installed on different computers*:*

1. If a single computer is planned to be used for both creating projects and calculations, then all components are installed on this computer.
2. If different computers are planned to be used for creating projects and for calculations, then
   - **Pre-Postprocessor** must be installed on client computers
   - **Solver** and **Solver-Agent** must be installed on server computers
   - location of other components depends on specific configuration

See detailed descriptions of various configurations in section Typical configurations.

4.5.1 Installation on Windows

Files in the Windows distributive package

The *Windows* distributive pack contains three archives (digits in their names depend on the program's version):

- `309_LM_Windows_20160430_Fv.zip` for installation **License Manager**
- `309_Windows_20160430_Fv.zip` for installation all other *FlowVision* modules
- `309_3DTV_Windows_20160430_Fv.zip` for combined installation of FlowVision and 3DTransVidia (this is for installation *3DTransVidia* and all *FlowVision* modules except **License Manager**)

Digits in the beginning of these names encipher a part of program's version number (for example, **309** means version number *3.09*) and digits in the middle encipher the build date of the distributive package (for example, **20160430** means 30th of April 2016).

Archive **309_LM_Windows_20160430_Fv.zip**[*)] contains one file:

File	Description
FlowVisionLM-windows-installer.exe	Executable file of the installer for **License Manager**

Archive **309_Windows_20160430_Fv.zip**[*)] contains the following files:

File	Description
FlowVision-3.13.01-windows-installer.exe [*)]	Executable file of the main installer of *FlowVision*
MSMpiSetup.exe	Installation program for the *MS MPI* library
ReadMe_enu.pdf	Documents in the *pdf* format containing sections of documentation required to install *FlowVision*
ReleaseNotes_eng.txt	Information about changes in all versions of *FlowVision*
requirements_eng.txt	Information about system requirements
vcredist_x64.exe	Program for installation/restoration distributable libraries *Microsoft Visual C++* (for 64-bit computers)

[*)] Number in the filename depends on the software version's number and/or release date.

Archive **309_3DTV_Windows_20160430_Fv.zip**, comparing to archive **309_LM_Windows_20160430_Fv.zip**, contains also all *3DTransVidia*'s files that are necessary for the installation.

Work modes of the installers under *Windows*

The installers can work in one of the following modes:

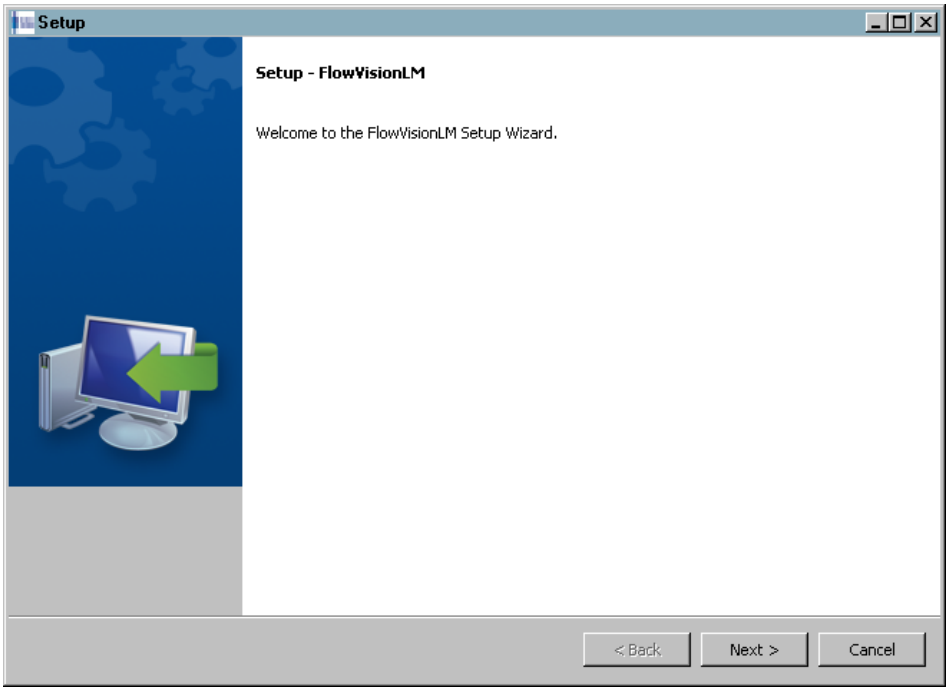
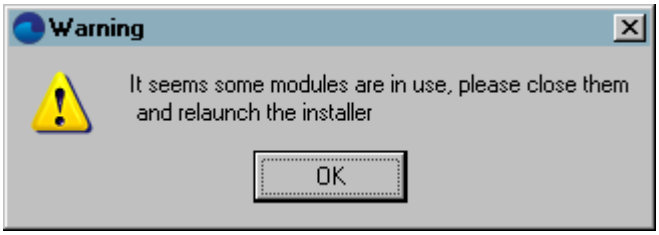
- *Installation of a new version* is started automatically when *FlowVision* of the installed version is *not* installed on the computer.
- *Recovery/changing* is started automatically when *FlowVision* of the version, which is being installed, has already been installed in the same directory. This mode is used for recovery of damaged installation and/or changing the list of installed modules.

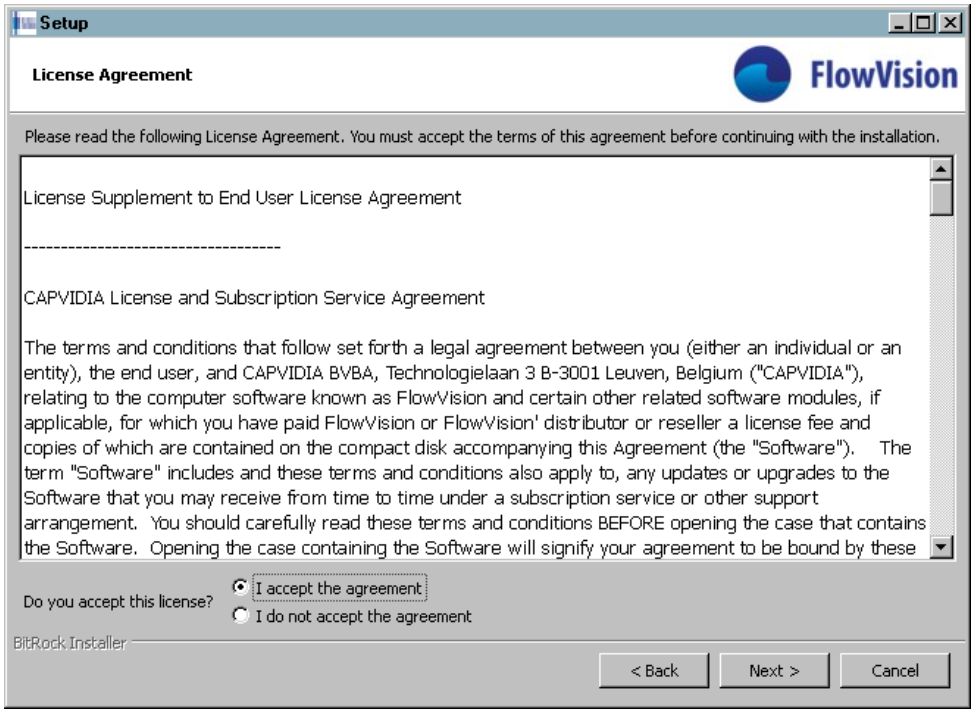
See step-by-step procedures of installation **License Manager** and other *FlowVision* modules under *Windows* in sections below:

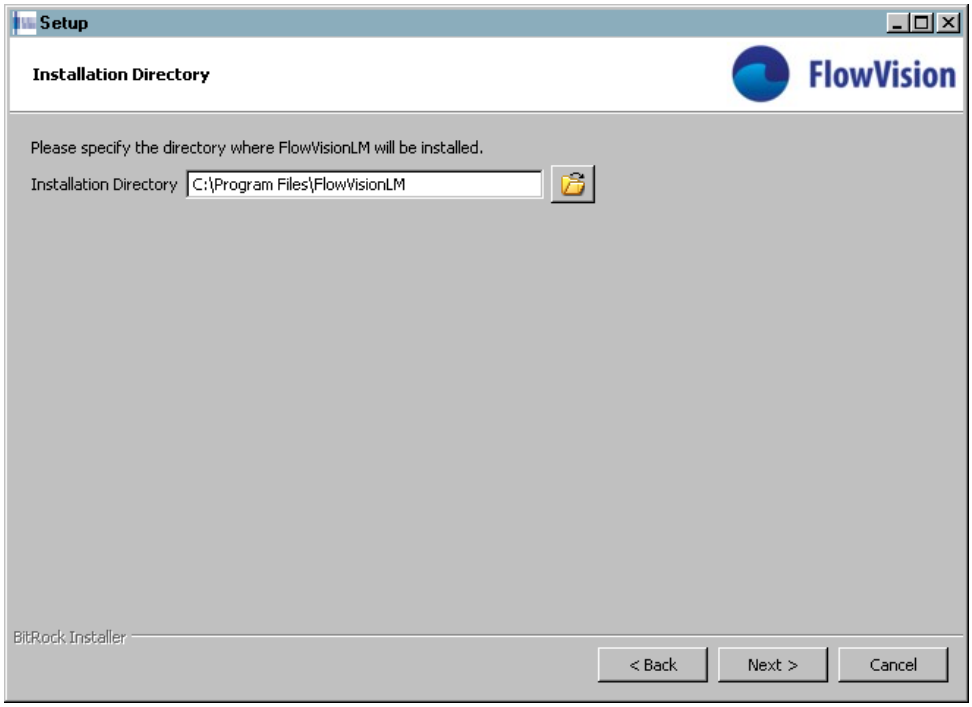
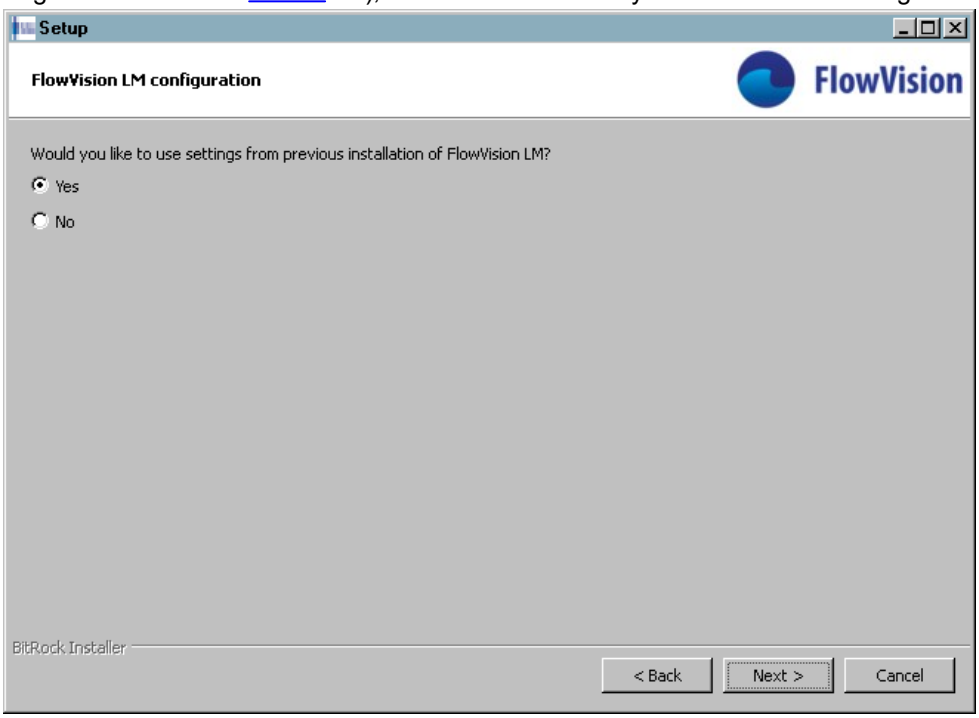
- Installing License Manager under Windows
- Installing FlowVision modules (except License Manager) under Windows

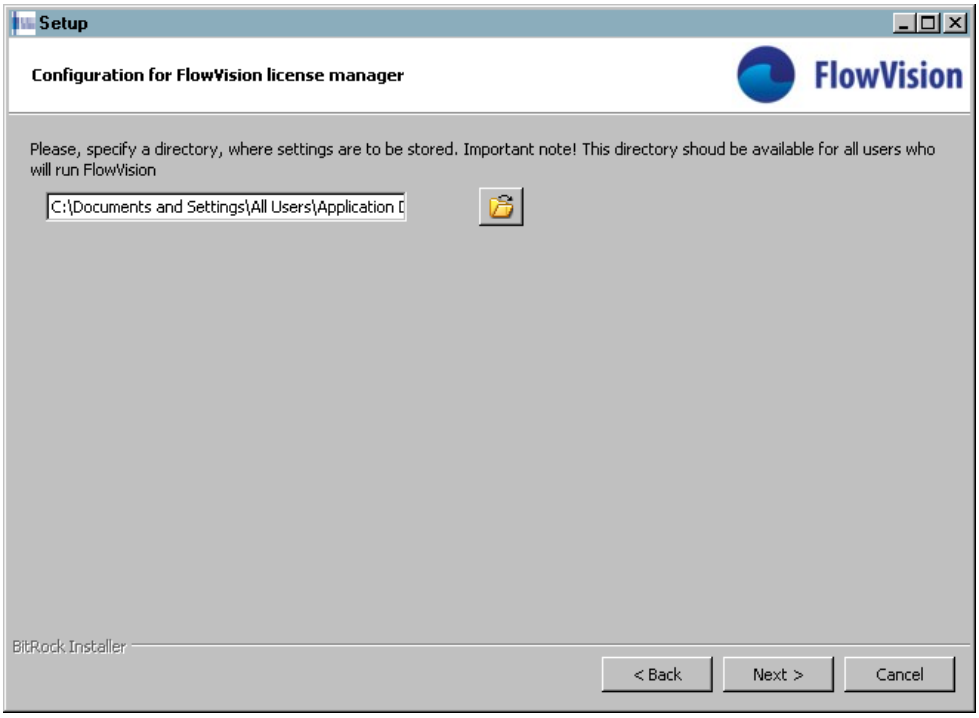
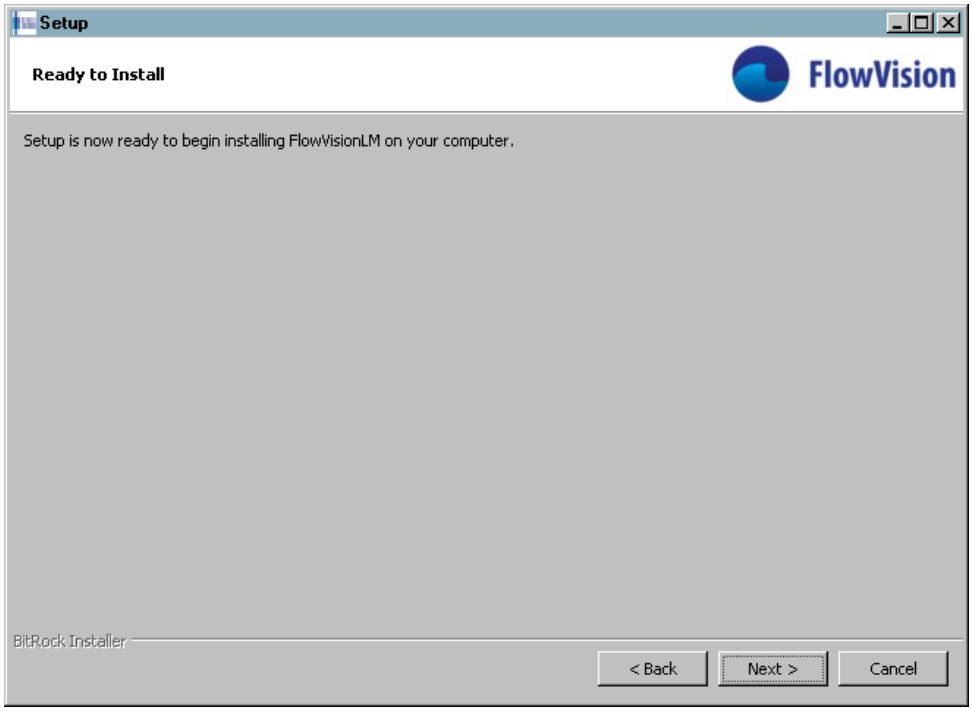
4.5.1.1 Installing License Manager on Windows

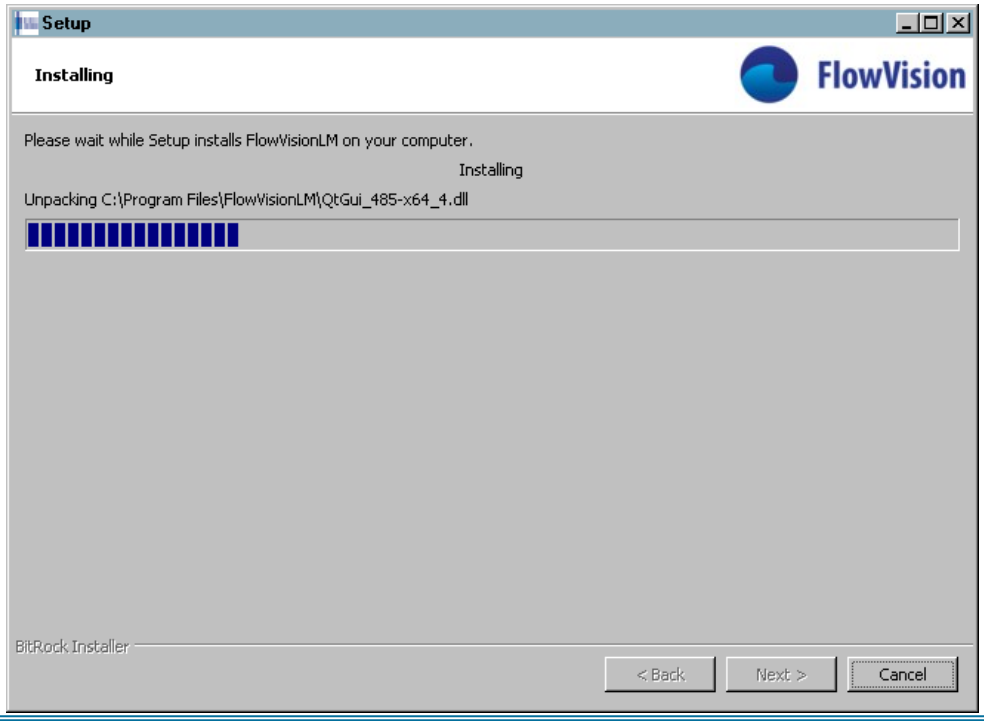
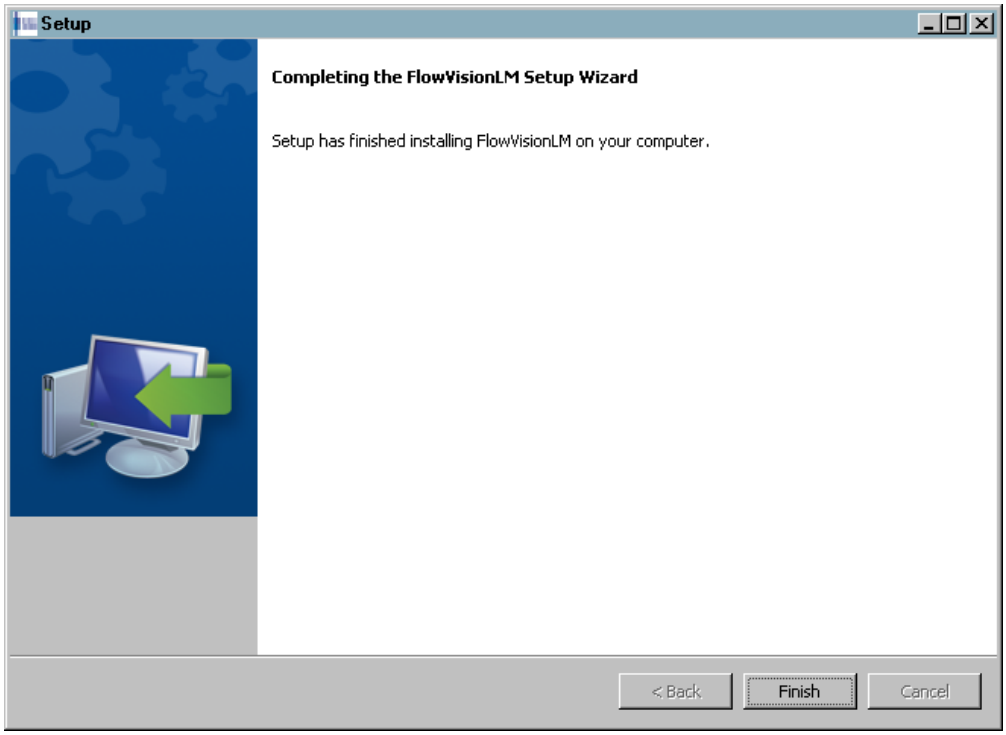
Step	Actions
1	Run, as a user with **Administrator**'s permissions, the executable file of the installer of the **License Manager** (**FlowVisionLM-windows-installer.exe**). A welcome dialog box will open:

Step	Actions
	 Click the **Next** button there. ***Note:*** If a component of previous or current version of *FlowVision* is working and this prevents the installation, a message ("**It seems some modules are in use, please close them and relaunch the installer.**") will open:  If so, click **OK**, then close the working component(s) of *FlowVision* and start the installer again.
2	A dialog box opens with the text of the license agreement:

Step	Actions
	 Carefully read the license agreement and select the **«I accept the agreement»** option if you agree with conditions of the license agreement. If you do not agree with conditions of the license agreement then, select the **«I do not accept the agreement»** option and abort the installation confirming your decision in a new dialog box, which opens. After you select your agreement with conditions of the license agreement, click **Next**. ***Note:*** this step might be skipped if the installer finds an existing previously installed **License Manager** (for example, when the installer is working to recover damaged files).
3	A dialog box will open, in which you have to specify an installation directory for the **License Manager**. (by default this directory is `C:\Program Files\FlowVisionLM`). The  button allows you to select the directory in the file system's tree.

Step	Actions
	 The **License Manager** can be installed in a location, which is closed for ordinary users of operating system because all other modules of *FlowVision* interact with **License Manager** through network connection(s). Click **Next** to continue the installation.
4	If in the directory, which you have specified for installation the **License Manager**, the installer detects settings from the previous installation of **License Manager** (information about location of the settings is stored in the **Fv.cfd** file), the installer will offer you to use these settings:  If you answer "**Yes**", then the question about a directory, where configuration files of the **License Manager** are stored (Step 5), will be skipped and the installer will immediately inform you about its readiness to install the **License Manager** (will go to Step 6). If you answer "**No**", then the installer will not use settings from the previous installation of the **License Manager**. Click **Next** to continue the installation.

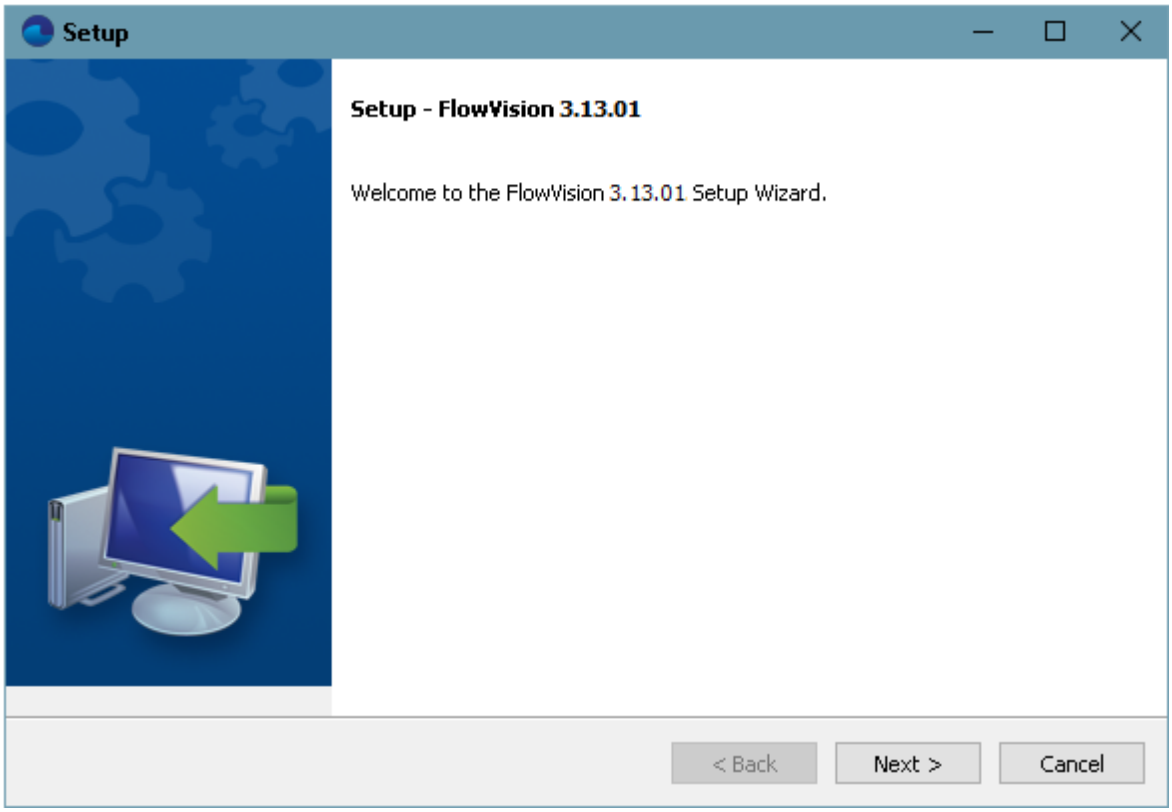
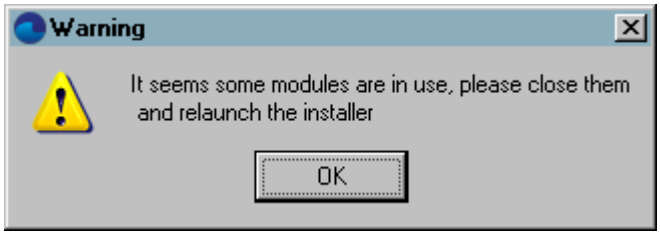
Step	Actions
5	A dialog box will open where you have to specify a directory, which will store settings and log files (the user's directory) of the **License Manager**:  This directory must be available for *reading and writing* for all operation system's users, as them the **License Manager** will run. Click **Next** to continue the installation. ***Note:*** this step might be skipped if use of old settings was selected.
6	A dialog box will open with a message about readiness to install the **License Manager** on your computer:  Click **Next** to continue the installation.
7	A dialog box will open with a progress bar indicating the installation:

Step	Actions
	
8	After completing the installation of the **License Manager** a dialog box opens with a message about it:  Click **Finish**.

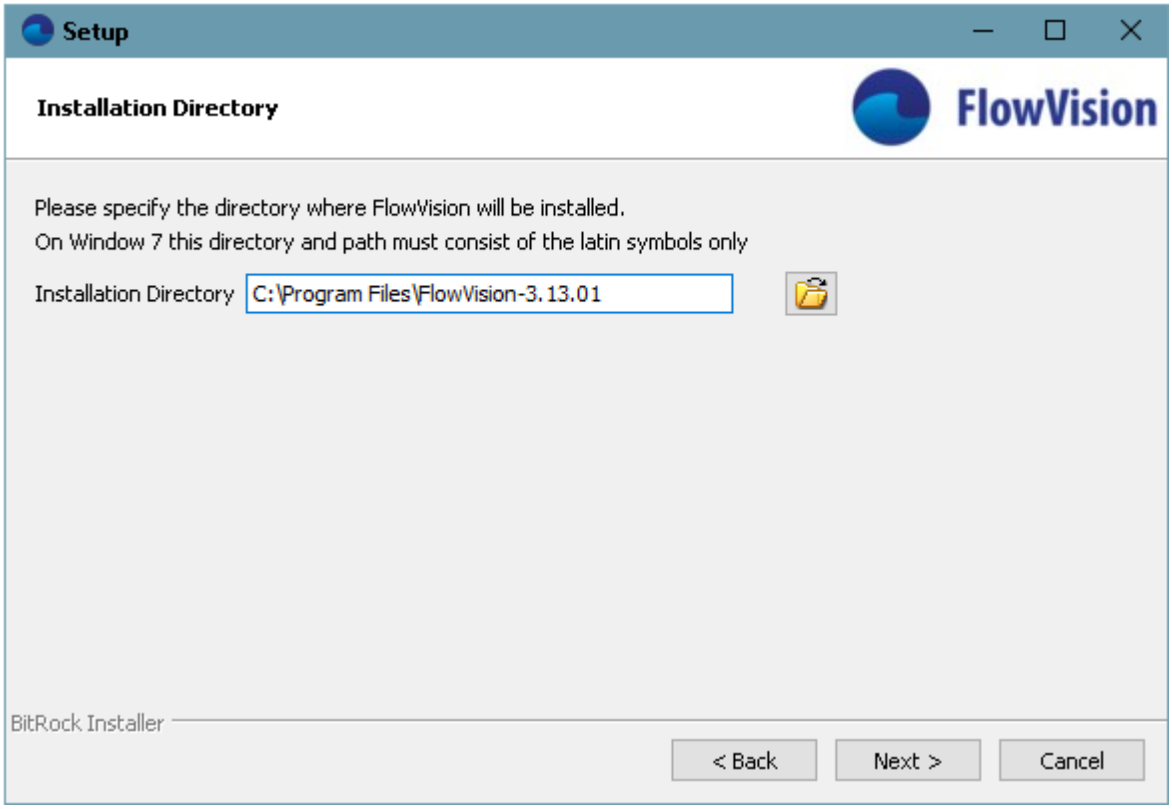
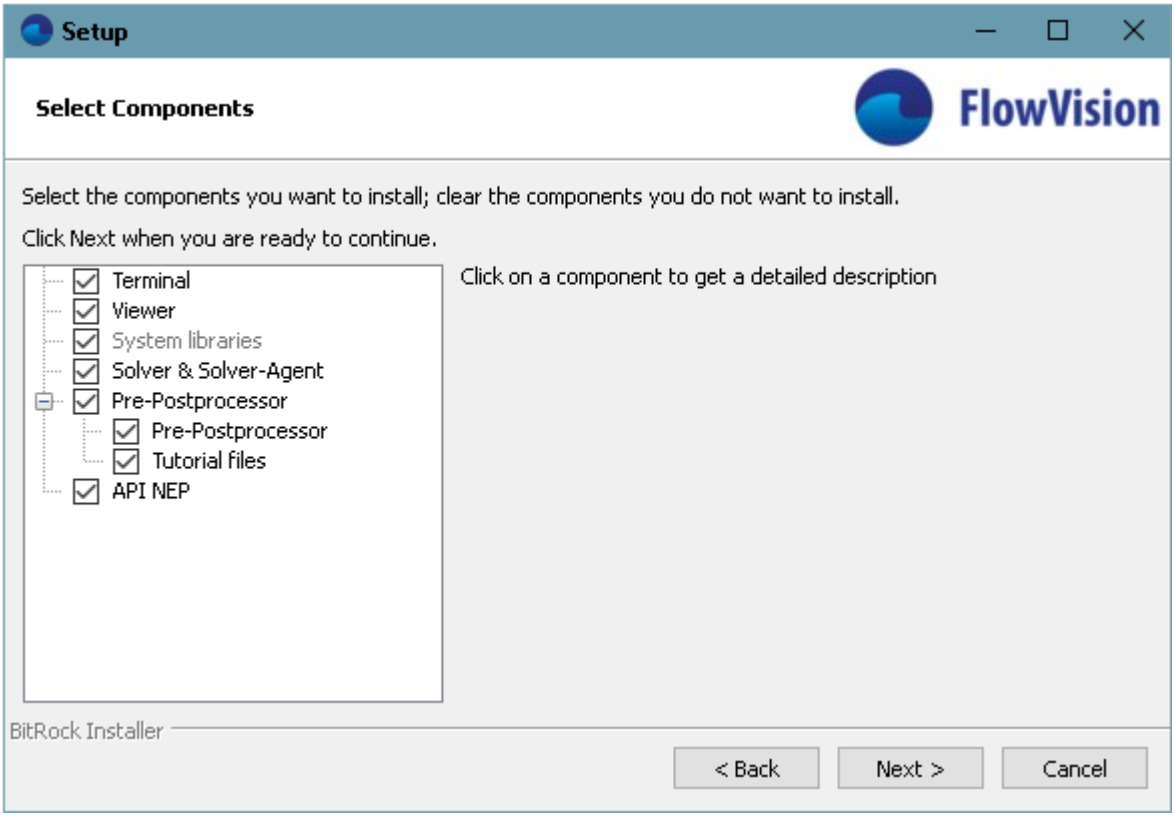
4.5.1.2 Installing FlowVision modules (except License Manager) under Windows

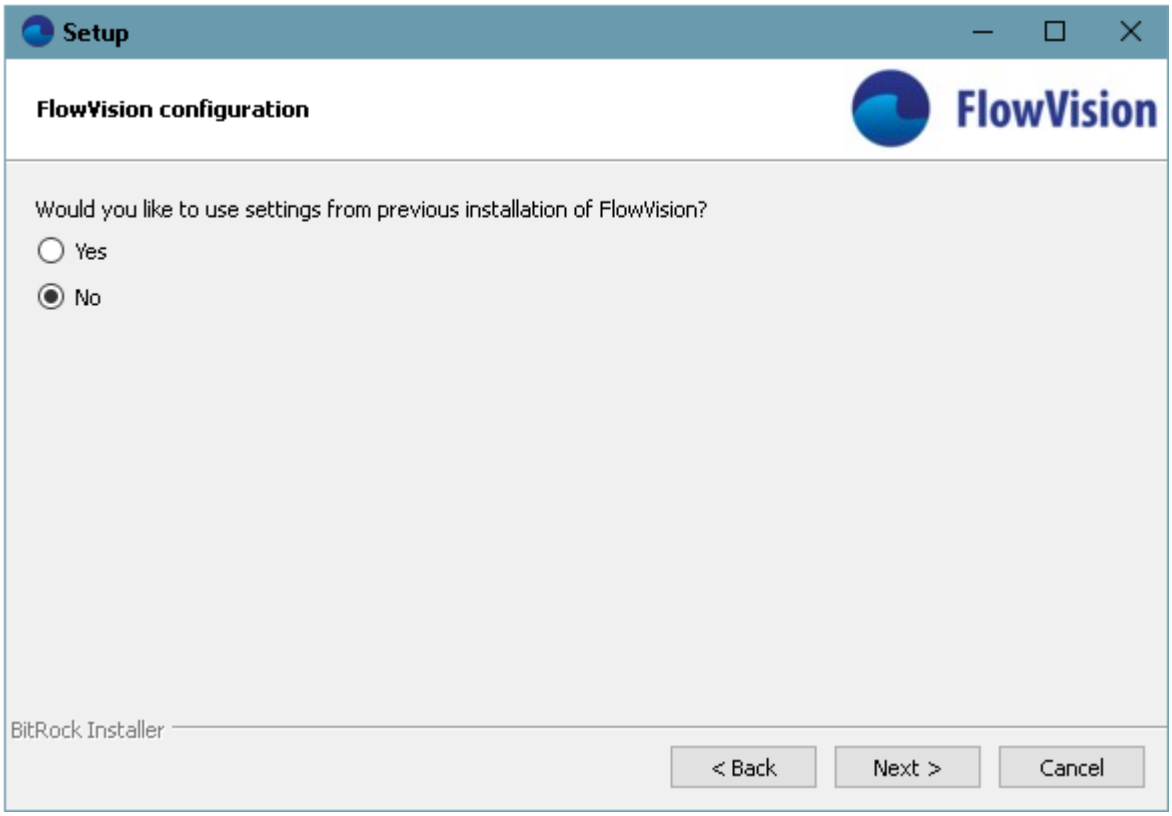
Here, the "installation *FlowVision*" refers to the installation of all the *FlowVision*'s components, except **License Manager**.

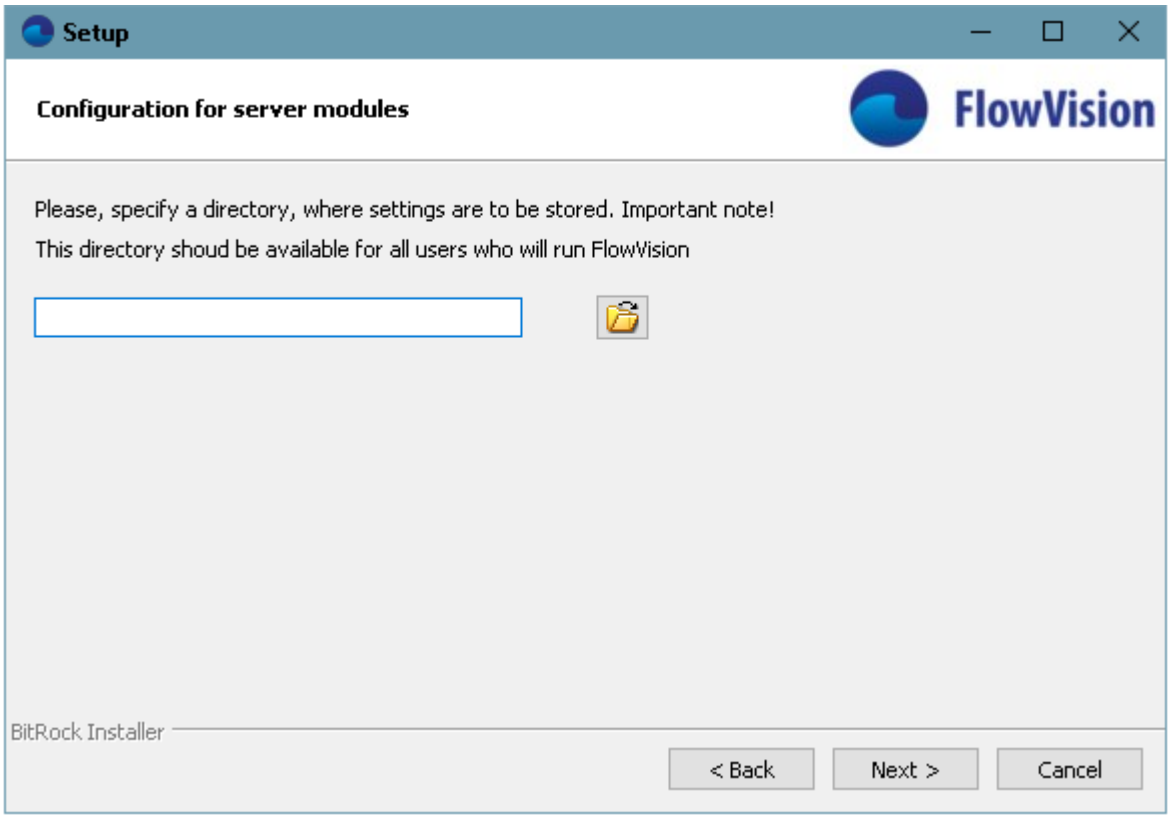
Step	Actions
1	Run, as a user with **Administrator**'s permissions, the executable file of the main *FlowVision*'s installer (`FlowVision-3.13.01-windows-installer.exe`, the digits in the file name indicate the version

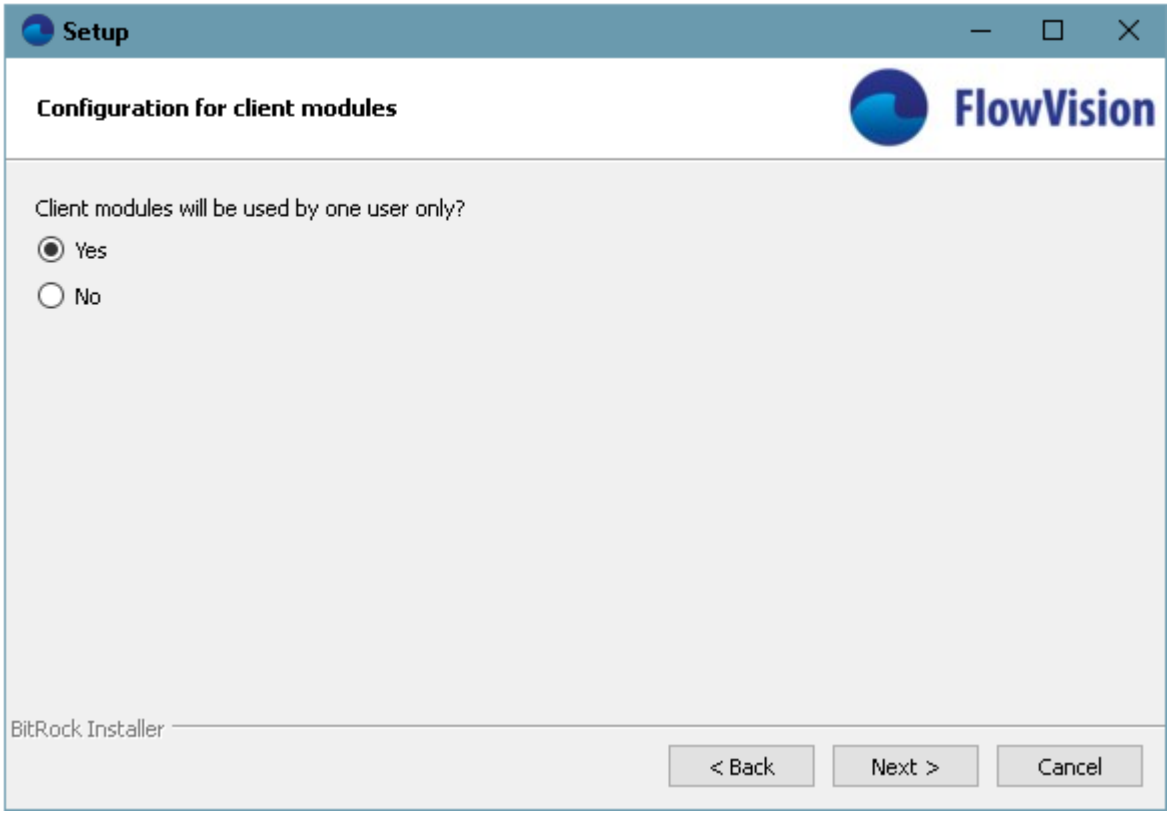
Step	Actions
	number). A welcome dialog box will open:  Click **Next**. ***Note:*** If any component of previous or current version of *FlowVision* is working and this prevents the installation, a message ("**It seems some modules are in use, please close them and relaunch the installer.**") will open:  If so, click **OK**, then close the working component(s) of *FlowVision* and start the installer again.
2	A dialog box with the text of the license agreement will open:

Step	Actions
	 Carefully read the license agreement and select the **«I accept the agreement»** option if you agree with conditions of the license agreement. If you do not agree with conditions of the license agreement then, select the **«I do not accept the agreement»** option and abort the installation confirming your decision in a new dialog box, which opens. After you select that you accept the conditions of the license agreement, click **Next**. ***Note:*** this dialog box will not open if the installer finds an existing previously installed *FlowVision* (for example, when the installer is working to recover damaged files).
3	A dialog box will open in which you have to specify the installation directory for *FlowVision* modules except **License Manager**. The  button allows you to select the directory in the file system's tree.

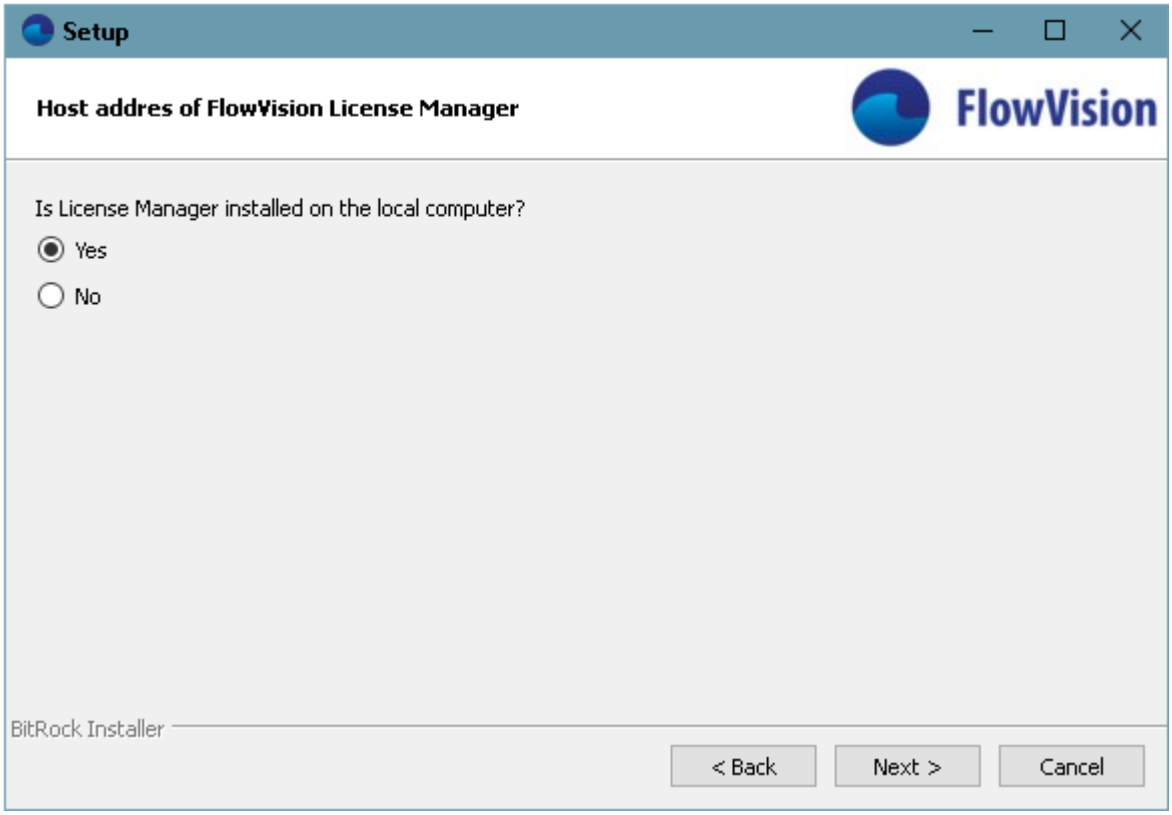
Step	Actions
	 Click **Next** to continue the installation. ***Note:*** this dialog box will not open if the installer finds an existing previously installed *FlowVision* (for example, when the installer is working to recover damaged files).
4	Select *FlowVision* modules, which will be installed:  Brief description of the highlighted module is displayed in the right part of the dialog box. Click **Next** to continue the installation.

Step	Actions
5	If in the directory, which you have specified for installation of the *FlowVision*'s modules, the installer detects settings from the previous installation (stored in the file **Fv.cfd**), the installer will offer you to use these settings:  If you answer **"Yes"**, some of the next steps will be skipped and the installer will pass directly to the dialog box about readiness to start the installation of *FlowVision*'s modules (Step 12). If you answer **"No"**, the installer will not read settings from the previous installation. Click **Next** to continue the installation.
6	Specify a directory, which will store settings of server modules and log files (the user's directory):

Step	Actions
	 This directory must be accessible for *reading and writing* for all users, as them *FlowVision*'s modules will run. Click **Next** to continue the installation. ***Note:*** this step is skipped if use of settings from the previous version of *FlowVision* has been selected.
7	Specify whether *FlowVision*'s client modules would be used by a single user or by multiple users:

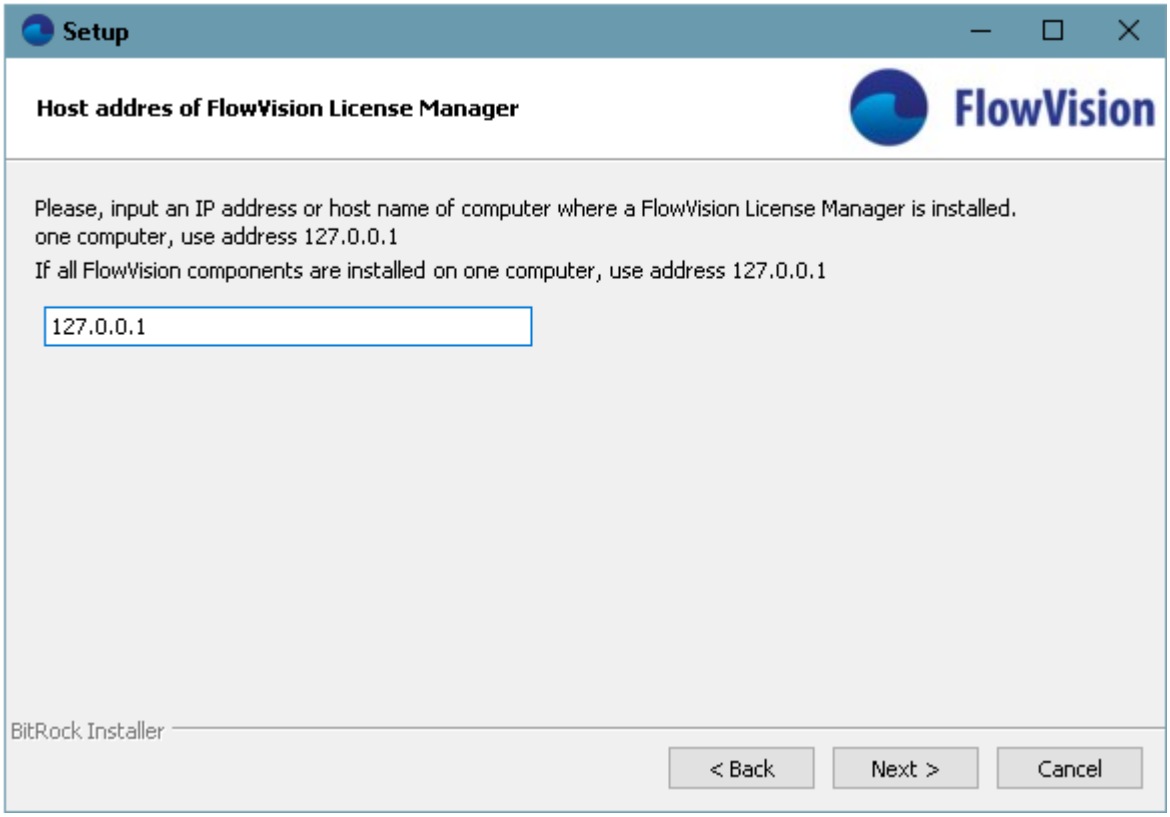
Step	Actions
	 If client modules (**Pre-Postprocessor**, **Terminal**, and/or **Viewer**) will be used by *several different* users of the operating system, then it is recommended to select **No**. In this case, all client module settings are stored in users' home directories. This selection allows use of *FlowVision* modules with individual user's settings. If you select **Yes**, then settings of the client modules will be stored in the directory, which was previously selected to store settings of the server modules. Click **Next** to continue the installation. ***Note:*** this step is skipped if use of settings from the previous version of *FlowVision* has been selected.
8	Specify the IP address or host name of the computer, on which **License Manager** is installed. The program will ask you first if **License Manager** is installed on the same computer:

Step	Actions



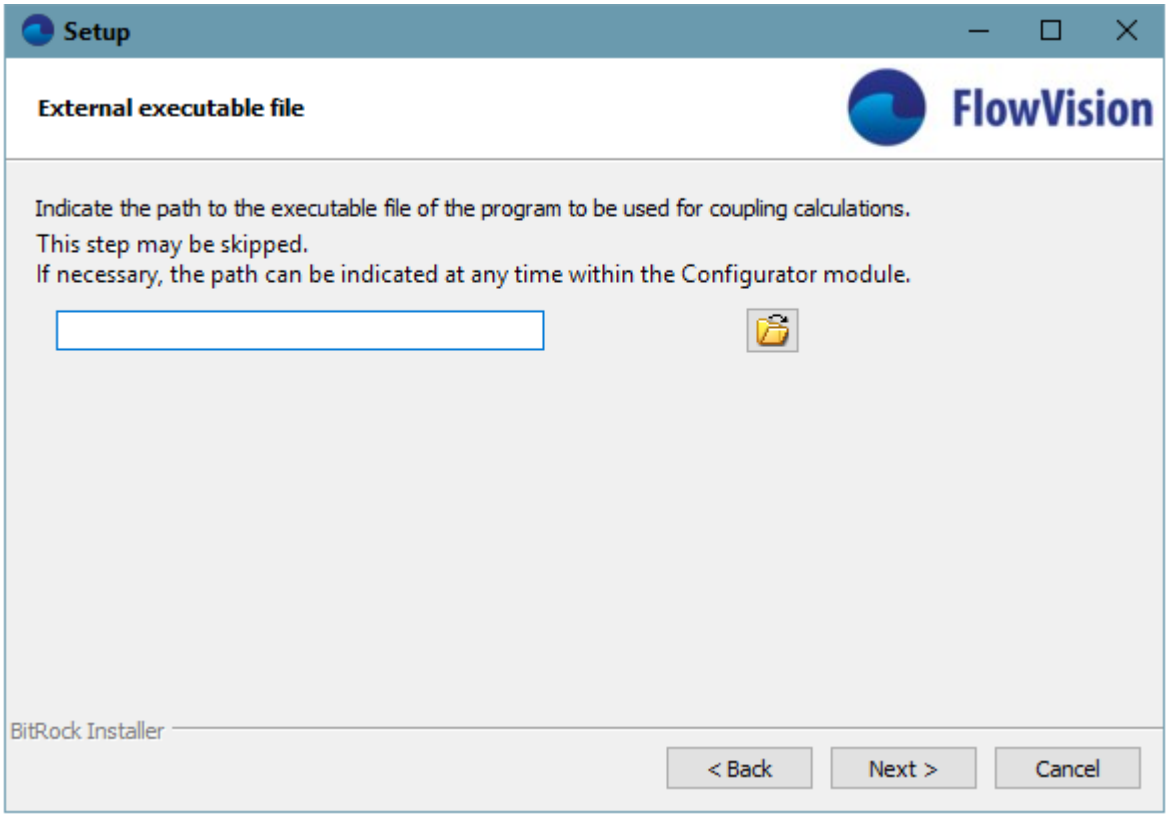
If you answer **Yes**, the IP address will be automatically set as **127.0.0.1**.

If you answer **No**, the program will request the IP address or host name of the computer, on which **License Manager** is installed:

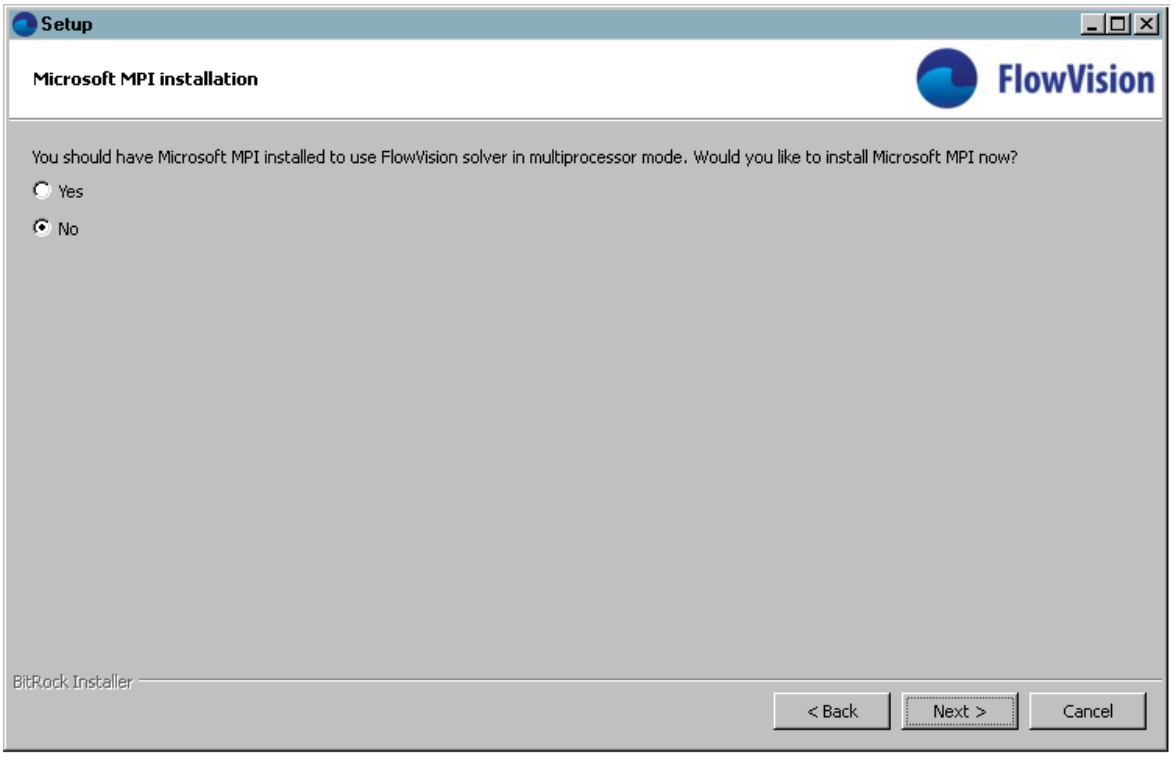


(by default the program will prompt IP address **127.0.0.1**, which corresponds to the computer, on which the installation of *FlowVision* is running)

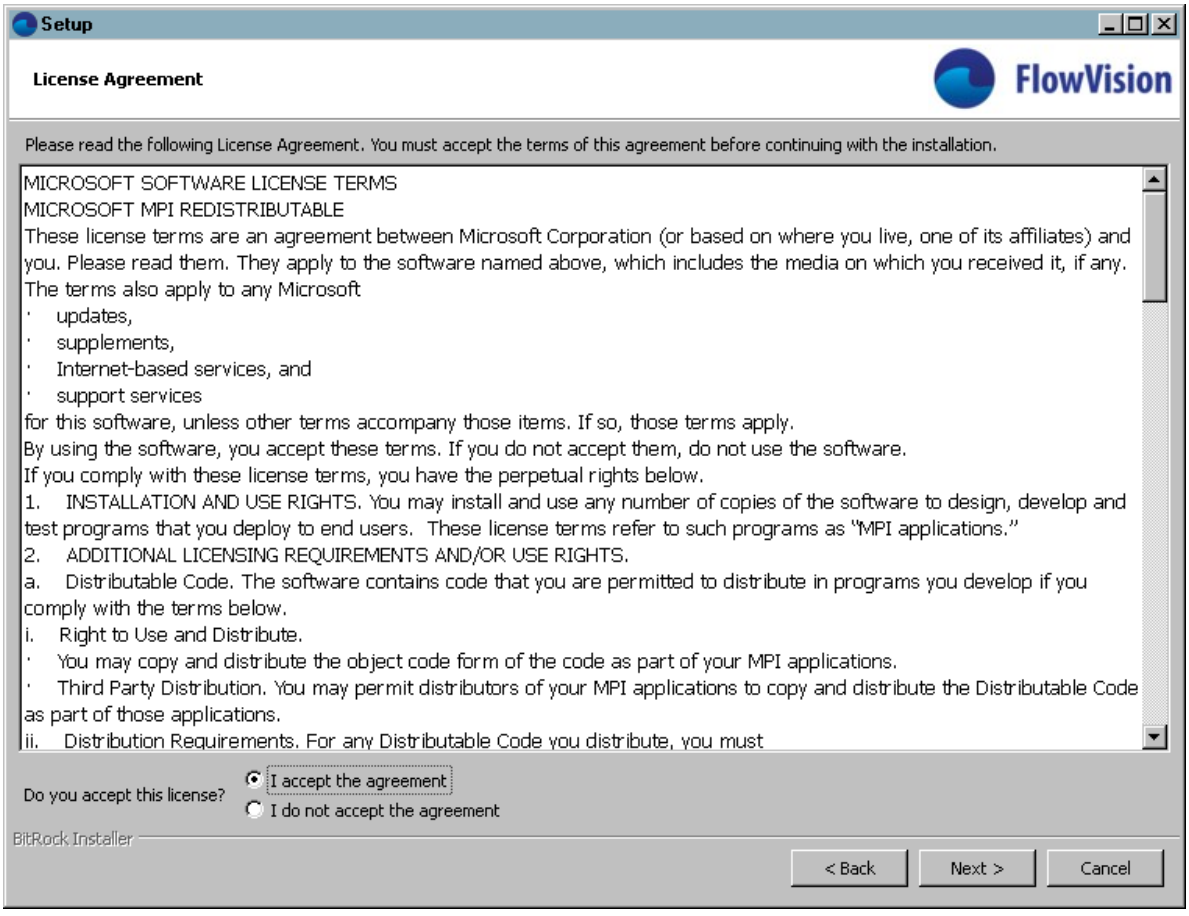
Enter the host name or IP address and click **Next** to continue the installation.

Step	Actions
	When you use a license with per-minute charge, specify the IP address, which you received after purchasing the license. ***Notes:*** • This step is only performed if the following *FlowVision*'s modules has been selected for the installation: ○ **Solver** and **Solver-Agent** ○ *and/or* **Pre-Postprocessor** • This step is skipped in the case when use of settings from the previous version of *FlowVision* has been selected.
9	The **External executable file** screen will open. If you plan to use *FlowVision* for joint computations with an external software (for example, *Abaqus FEA*) specify the path to the executable file, which is to be run by the **MPM-Agent** module:  Click **Next** to continue the installation (if use of external software is not planned or if the external software has not been installed yet, you don't have to enter the data in this dialog box). ***Notes:*** • This step is only performed if **MPM-Agent** has been selected for the installation. • This step is skipped if use of settings from the previous version of *FlowVision* has been selected.
10	For multiprocessor work of **Solver**, it is necessary to have an installed *Microsoft MPI* (*MS MPI*) library. The installer will prompt you to install the *MS MPI* library (**You should have Microsoft MPI installed to use FlowVision solver in multiprocessor mode. Would you like to install Microsoft MPI now?**):

Step	Actions

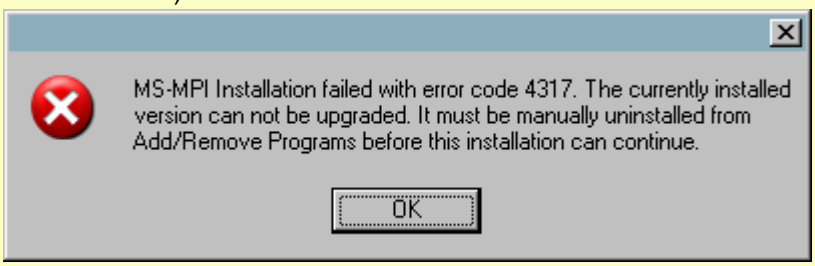
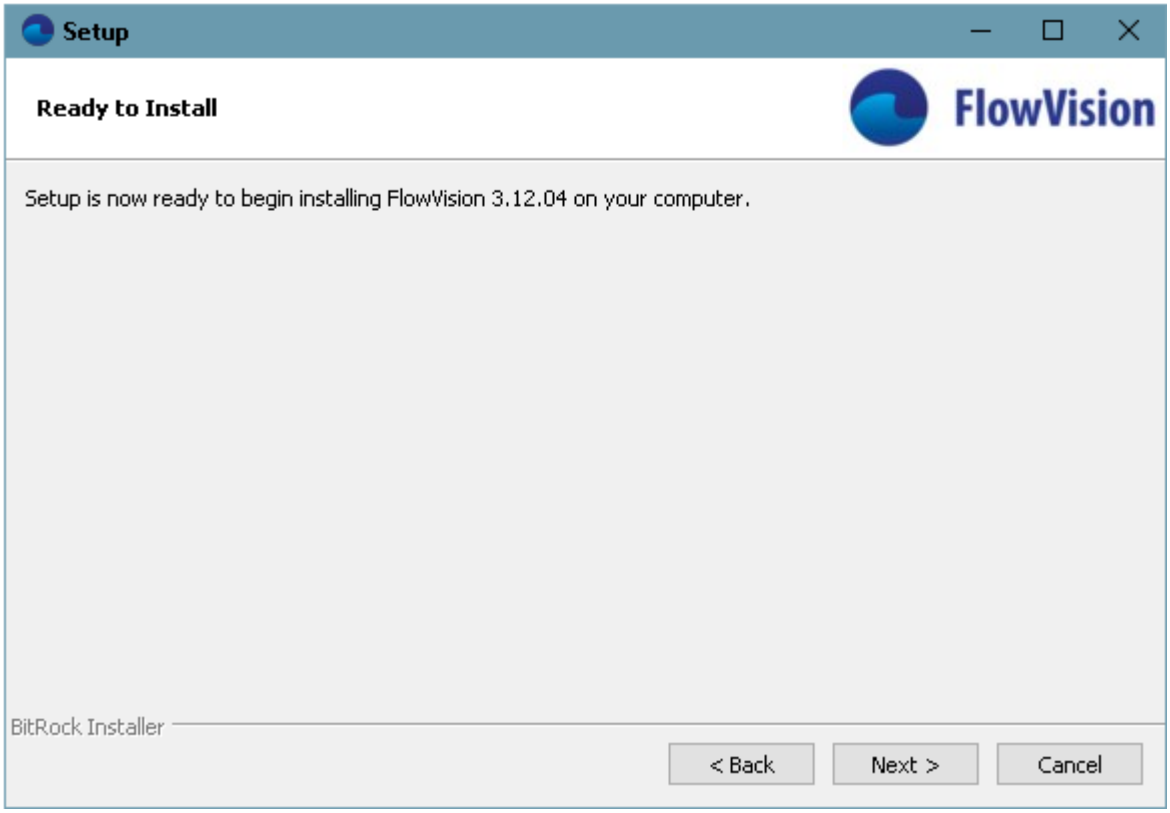


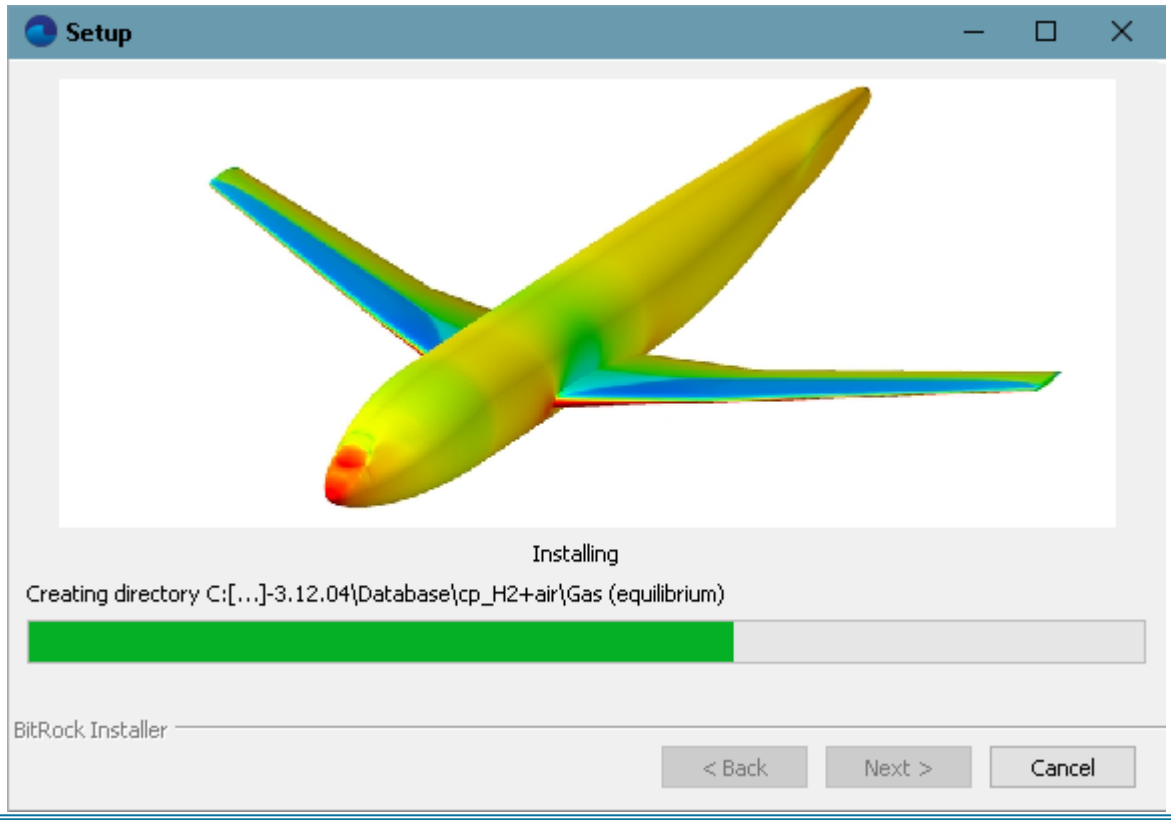
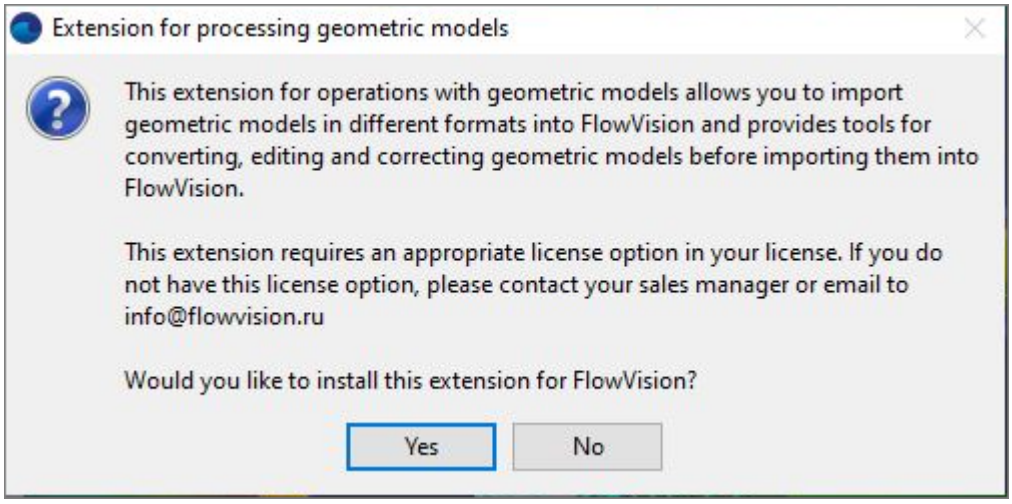
After you positive answer, the installer will prompt you to read and accept the license agreement:

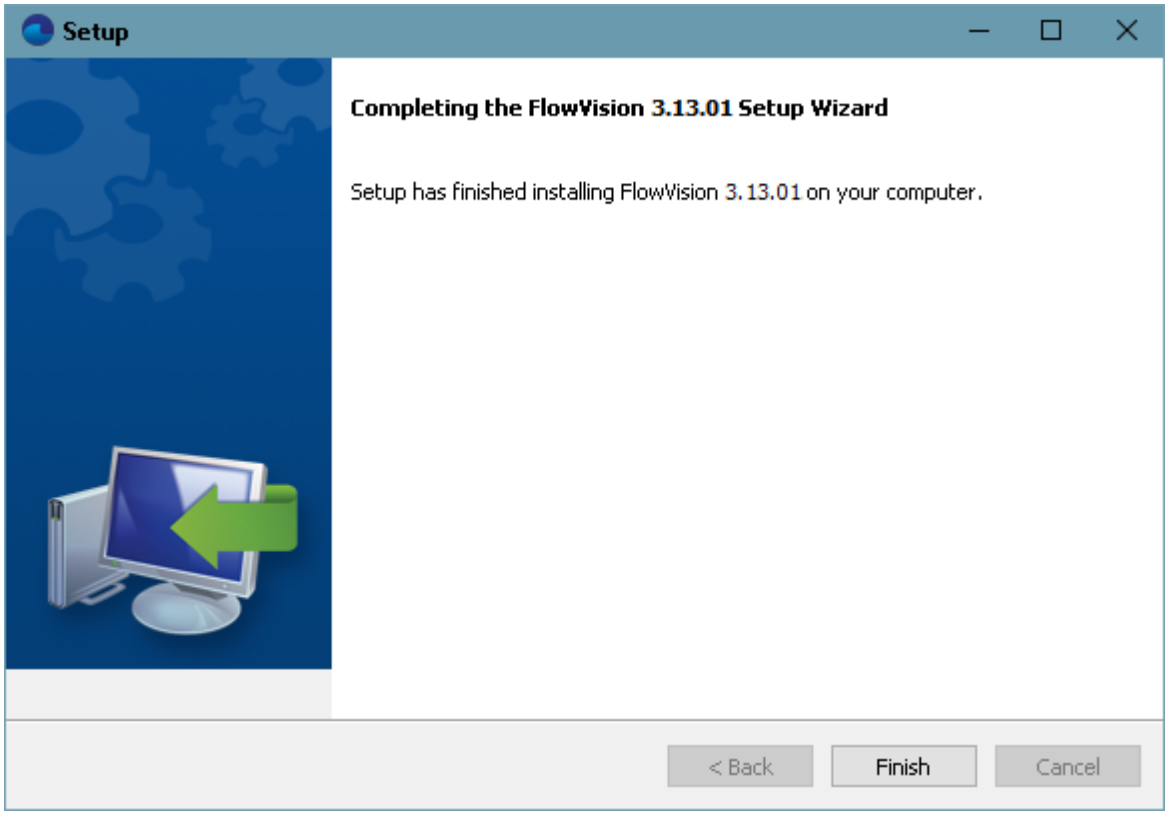


Carefully read the license agreement and select the **«I accept the agreement»** option if you agree with conditions of the license agreement.

If you do not agree with conditions of the license agreement then, select the **«I do not accept the agreement»** option and abort the installation confirming your decision in a new dialog box, which opens.

Step	Actions
	After you select that you accept the conditions of the license agreement, click **Next**. During the installation of *MS MPI v.7*, if an earlier version of *MS MPI* is installed on this computer, the installer might prompt you to run an update or display an error message (**MS-MPI Installation failed with error code 4317. The currently installed version can not be upgraded. It must be manually uninstalled from Add/Remove Programs before this installation can continue.**):  If you see this message, remove the old version of *MS MPI* or *MS HPC pack* using the standard *Windows*' feature for adding and removing programs and then run the installer of *MS MPI v.7* again.
11	A dialog box will open with a message about readiness to start the installation of *FlowVision*'s modules:  Click **Next** to continue the installation.
12	A dialog box will open with changing illustrations of the *FlowVision*, displaying the installation process:

Step	Actions
	
13	A dialog box will open, which prompt you to install the extension for processing geometric models:  This extension might require the following license options: • **Preparing of Geometry models** • **Import .mesh, .stp, .igs geom.models** • **Import extension** Click **Yes** to install the extension or click **No** to refuse installing the extension.
14	Upon completion of the installation, a dialog box will open:

Step	Actions
	 Click **Finish**.

4.5.2 Installation on Linux

FlowVision can only be installed on 64-bit versions of *Linux* (see details is section Specific system requirements for Linux).

FlowVision installer for *Linux* has two functions:

1. It allows you to select the required software modules and unpacks archive with their files.
2. It makes a preliminary setup of *FlowVision*.

FlowVision installer does not interact with operation system's functionality for management of installed packages and applications.

Unlike installation on *Windows*, on *Linux* you can install an unlimited number of *FlowVision* copies on one computer, even using the same installer. This also is true for the installer of **License Manager**, but you cannot run more then one copy of **License Manager** at the same time.

When an installer attempts to install a program into a directory, where the program with the same version is already installed, the installer works the mode of Recovery of damaged installation and/or changing the list of installed modules.

See details in sections below:

- Specific system requirements for Linux
- Distribution pack for Linux and methods for installing FlowVision's modules
- Installing License Manager under Linux (in graphical mode)
- Installing License Manager under Linux (in text mode)
- Installing FlowVision modules (except License Manager) under Linux in graphical mode
- Installing FlowVision modules (except License Manager) under Linux in text mode
- Preparing Solver for work with an installed MPI
- Compilation a library for interconnection with various MPI implementations

4.5.2.1 Specific system requirements for Linux

System requirements of the installer

The installer can be run either in a console mode or in a graphical environment. Only 64 bit systems are supported.

System requirements to operating system of the installer

- Graphical window interface or command line interface (terminal)

System requirements of *FlowVision* modules

FlowVision can only be installed on 64-bit versions of *Linux*.

Hardware requirements are the same in both *Windows* and *Linux*. Any *FlowVision*'s module can only be run on a 64-bit operating system.

FlowVision is tested on the following *Linux* operating systems:

- *CentOS 6.10*
- *CentOS 7.9*

FlowVision has certificates of compatibility with *RedOS* and various versions of *Alt Linux*. Functionality on other *Linux* versions is not guaranteed, but it is knows that *FlowVision* effectively operates in a many of versions of *Linux* including:

- *Ubuntu*
- *Open SUSE*
- *RedHat*
- etc.

Also use of *Unity* and *GNOM* was successful.

When working with graphical *FlowVision*'s modules, it is recommended to use the graphical environment *KDE*.

The following libraries *must* be installed in the system:

- *glibc* (version *2.12* or newer)
- *OpenGL 64bit* (version *3.0* or newer)

Support of *MPI*

By default, **Solver** is ready to run in multiprocessor mode (with *MPI*) using *Intel MPI*. However *FlowVision* can also work with other versions of *MPI*.

Support of *MPI* is implemented using the library **`libFvMPI.so`**, which is located in the directory **`lib64/`** in the installation directory of *FlowVision*. In the same directory there are several precompiled libraries for various implementations of *MPI*, which are included with the *FlowVision*'s distribution pack:

- **`libFvMPI_impi2018.so`** - *Intel MPI 2018*
- **`libFvMPI_impi2019.so`** - *Intel MPI 2019*
- **`libFvMPI_impi403_static.so`**
- **`libFvMPI_mpich3.so`**
- **`libFvMPI_mvapich19.so`**
- **`libFvMPI_ompi142.so`**
- **`libFvMPI_ompi164.so`**
- **`libFvMPI_ompi184.so`**
- **`libFvMPI_ompi216.so`** - *OpenMPI 2.1*
- **`libFvMPI_ompi401.so`** - *OpenMPI 4.0*

To tune the **Solver**'s work some of *MPI* implementations, you should copy the appropriate library and save it by the name of **`libFvMPI.so`** (see details in section Preparing Solver for work with an installed MPI).

You can also work with other versions of *MPI*, see details in section Compilation a library for interconnection with various MPI implementations.

Directory names

It is recommended not to use in installer directory names (including the names that are included in specified file system's paths), which contain non-English letters. If you really need to use such directories, type their names into appropriate installer' GUI fields manually, do not select the directories and paths using the directory selection button ().

See also: section General system requirements)

4.5.2.2 Distribution pack for Linux and methods for installing FlowVision's modules

Distribution pack for *Linux*

Linux distribution pack distributive pack contains two archives, one for **License Manager** and another for other modules, with names like these:

- `309_LM_Linux_20130320_Fv.tar.gz`*) for **License Manager**. This archive contains the file `FlowVisionLM-linux-x64-installer.run`, which is installer for **License Manager**.
- `309_Linux_20150430_Fv.tar.gz`*) for other *FlowVision*'s modules. This archive contains the file `FlowVision-3.09.04-linux-x64-installer.run`*), which is installer for main *FlowVision*'s modules.

*) Digits in the beginning of these names encipher a part of program's version number (for example, `309` means version number *3.09*) and digits in the middle encipher the build date of the distributive package (for example, `20150430` means 30th of April 2015).

If the **tar** program is installed in your system, you can unpack the archives using the commands (*digits in the file names correspond to the version number and the build date, and might differ from digits given in the example*):

```
tar -xzf 309_LM_Linux_20130320_Fv.tar.gz
tar -xzf 309_Linux_20150430_Fv.tar.gz
```

Work modes of *FlowVision* installers for *Linux*

FlowVision's installer can operate in one of three modes:

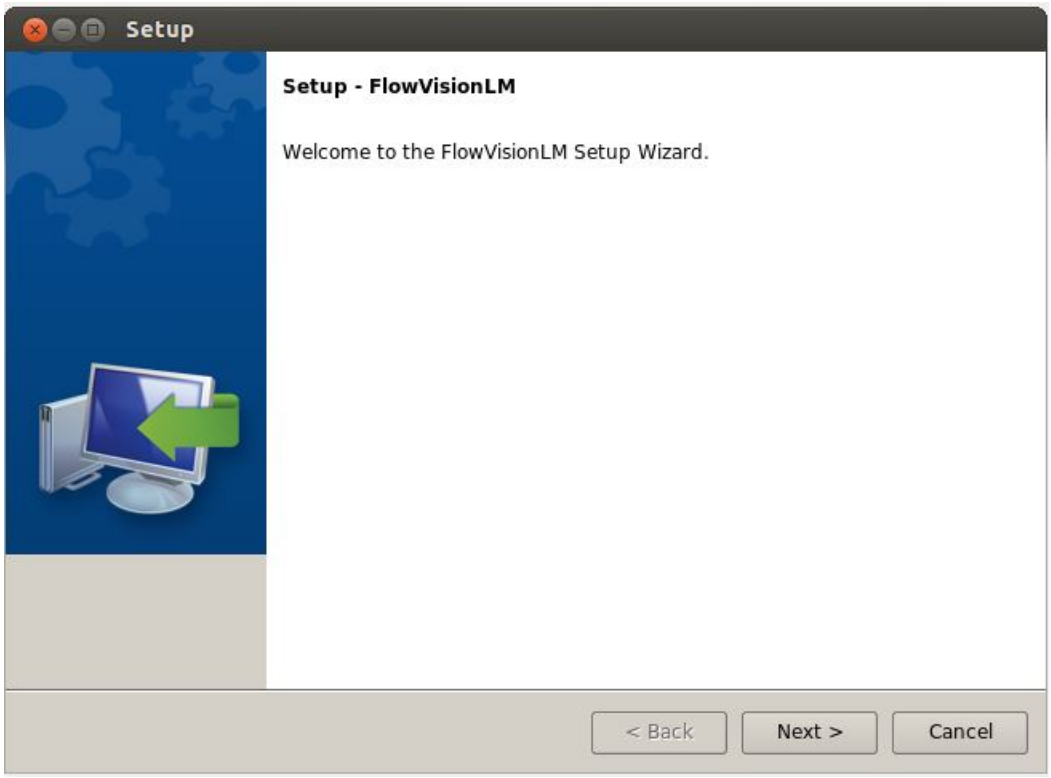
1. *Graphic mode*, which uses a dialog-based graphical interface of the operating system. Settings are specified in dialog boxes.
2. *Text mode*, which is the console mode of installation. This mode is convenient for remote installation *FlowVision* using *SSH*, it allows you to install the software package via the terminal.
3. *Automatic installation*, which neither displays any dialog boxes no ask questions; this mode requires no user's intervention. This mode is convenient for automated deployment *FlowVision* on a large number of computers.

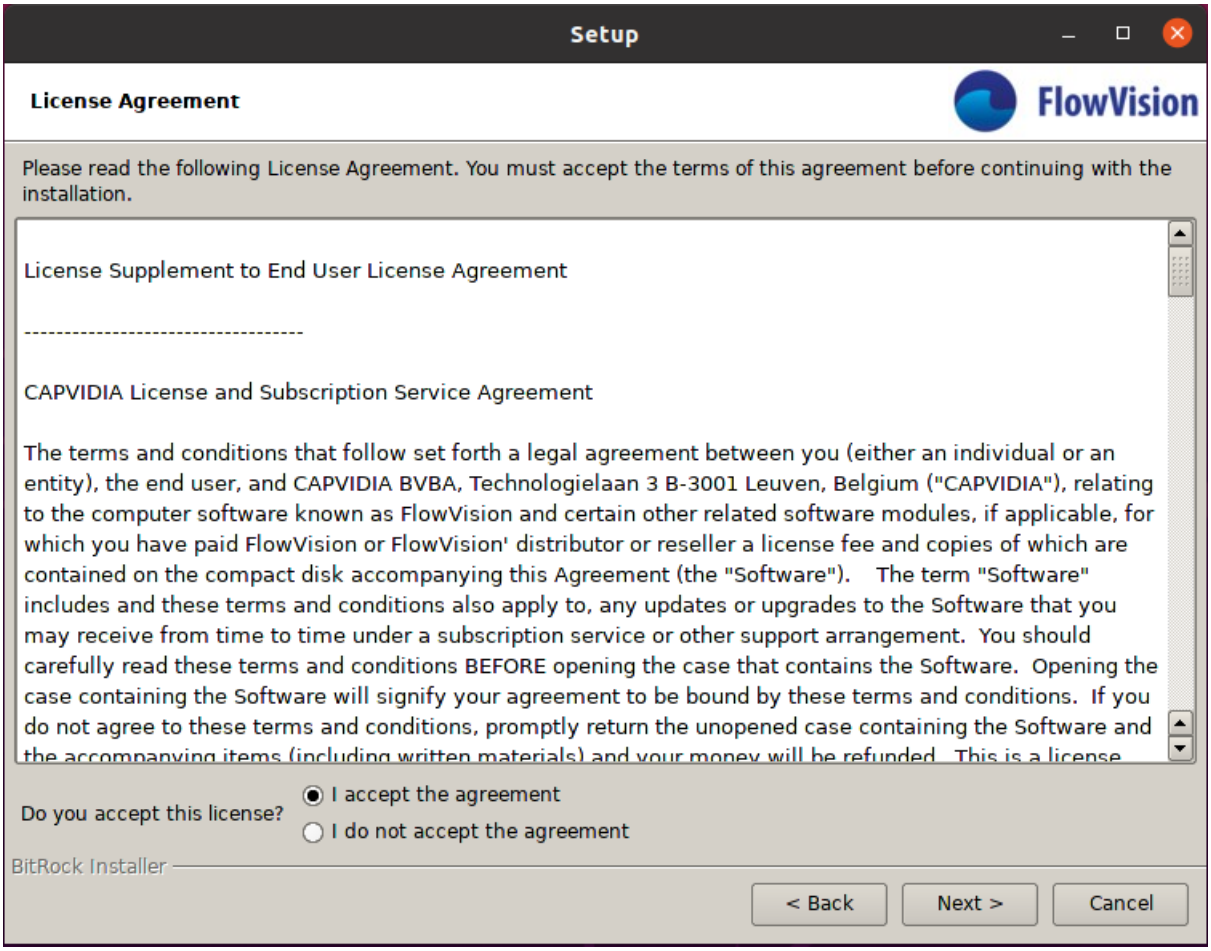
Also it is possible to install *FlowVision* by copying its directory to another computer. In this case, after copying of the files of the already installed *FlowVision*, you have to run the **Configurator** module to configure the storage of configuration files and to configure the *FlowVision*'s modules.

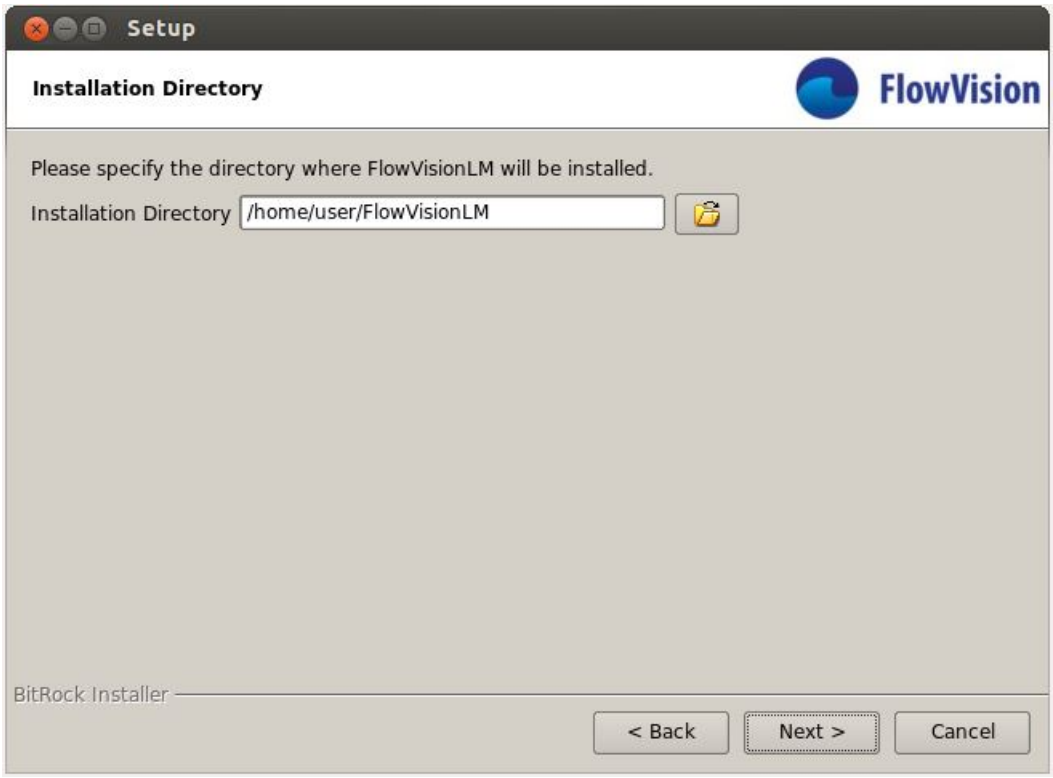
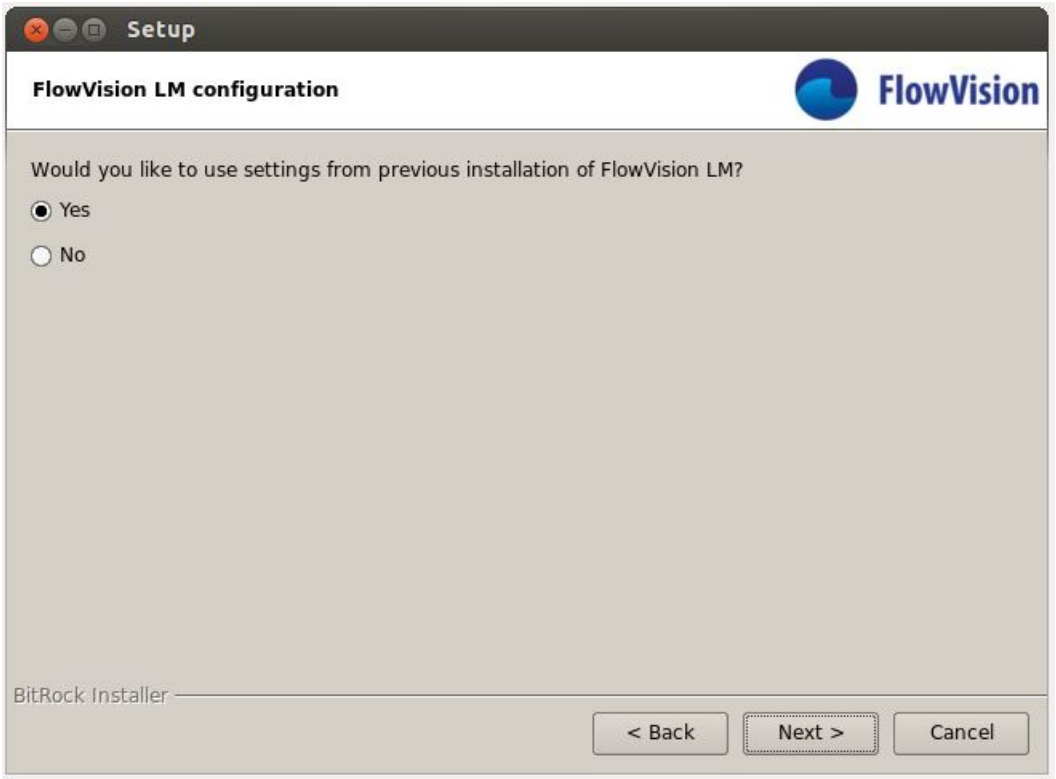
To run the installer you might need to change attributes of files from the archives, to make the installer file executable. This can be done by configuring properties of the file in the *Linux*'s graphical interface or specifying them from the command line, as in the example below:
`chmod 755 FlowVision-3.13.01-linux-x64-installer.run` (digits in the file name correspond to the version number)

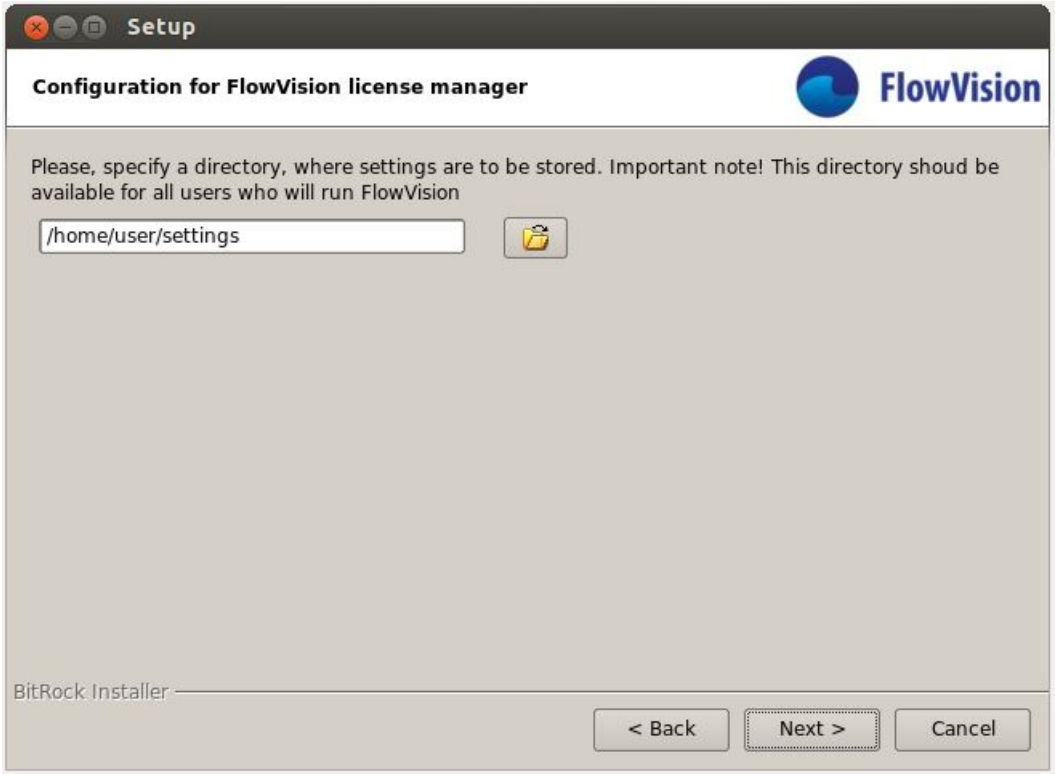
4.5.2.3 Installing License Manager on Linux (in graphical mode)

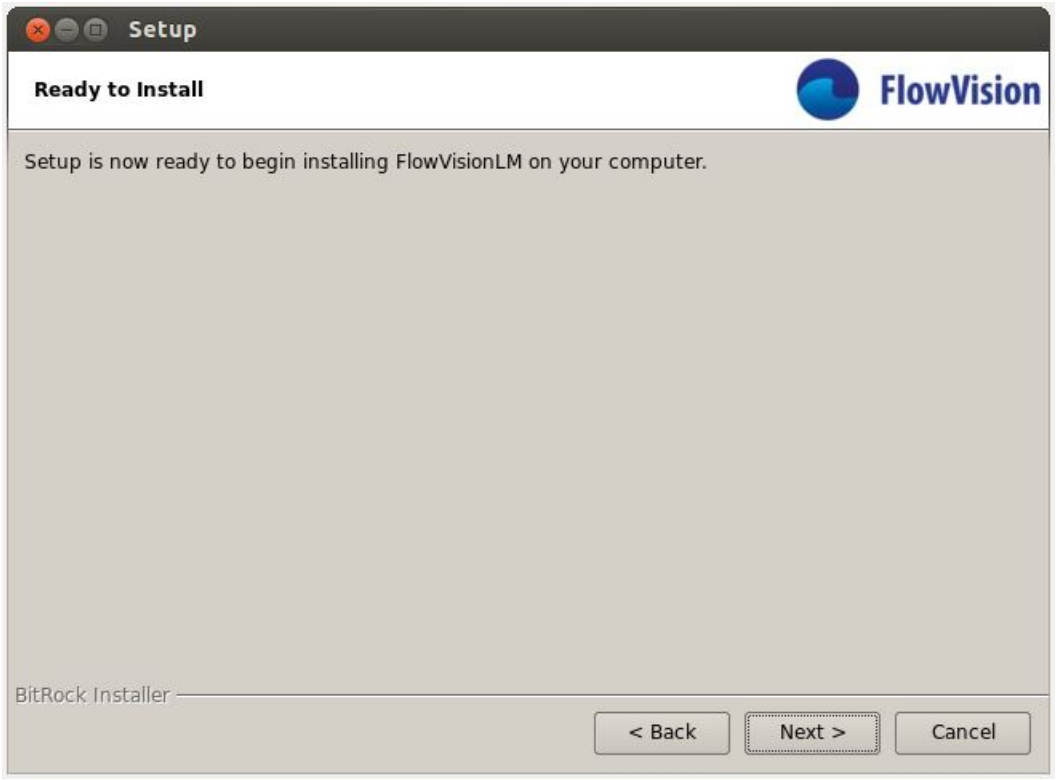
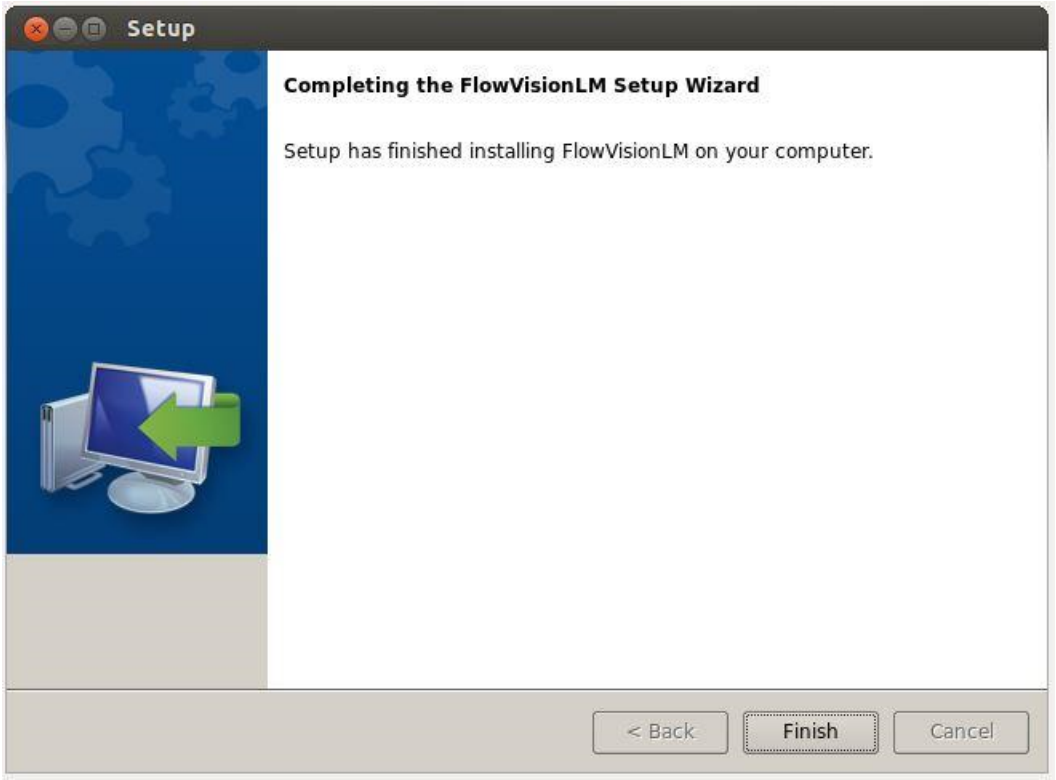
Step	Actions
1	Run the installer for **License Manager** (`FlowVisionLM-linux-x64-installer.run`). The welcome dialog box will open:

Step	Actions
	 Click **Next** to continue the installation. ***Note:*** if the installer does not run, make sure that its file (`FlowVisionLM-linux-x64-installer.run`) is executable. If no, grant appropriate permission for it.
2	A dialog box opens with the text of the license agreement:

Step	Actions
	 Carefully read the license agreement and select the **«I accept the agreement»** option if you agree with conditions of the license agreement. If you do not agree with conditions of the license agreement then, select the **«I do not accept the agreement»** option and abort the installation confirming your decision in a new dialog box, which opens. After you select your agreement with conditions of the license agreement, click **Next**. Click **Next**. ***Note:*** this step might be skipped if the installer finds an already installed **License Manager** (for example, when the installer runs to recover damaged files).
3	A dialog box opens in which you have to specify the installation directory for the **License Manager** (the default directory's name is `/home/user/FlowVisionLM`):

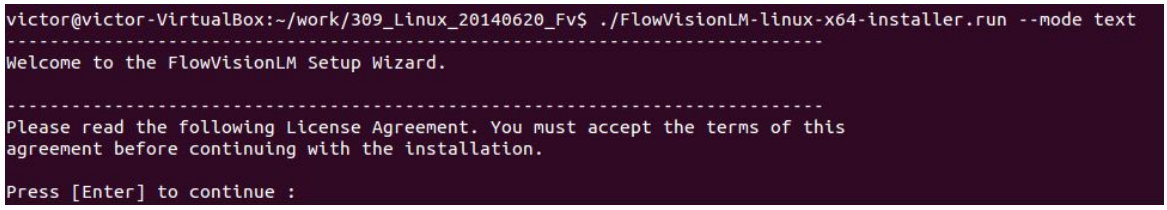
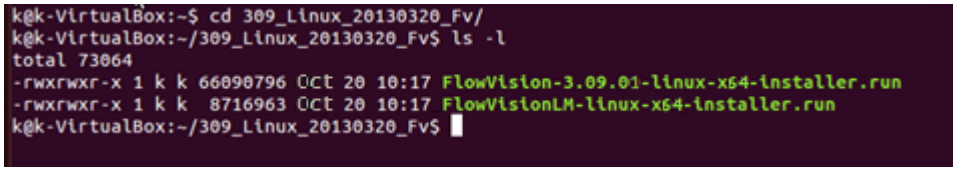
Step	Actions
	 **License Manager** can be installed in an network location, which is closed for ordinary users of the operating system, and all other *FlowVision*'s modules interact with it via a network connection. Click **Next** to continue the installation.
4	If in the directory, which you have specified for installation the **License Manager**, the installer detects settings from the previous installation of **License Manager** (information about location of the settings is stored in the Fv.cfd file), the installer will offer you to use these settings:  If you answer "**Yes**", then the question about a directory, where configuration files of the **License Manager** are stored (Step 5), will be skipped and the installer will immediately inform you about its readiness to install the **License Manager** (will go to Step 6).

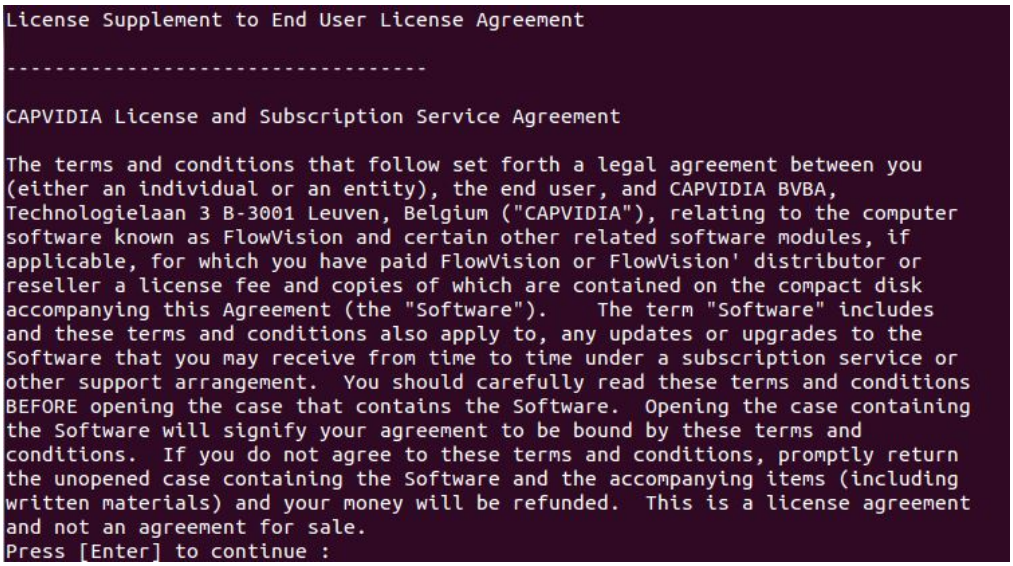
Step	Actions
	If you answer "**No**", then the installer will not use settings from the previous installation of the **License Manager**. Click **Next** to continue the installation
5	A dialog box will open where you have to specify a directory, which will store settings and log files (the user's directory) of the **License Manager**:  This directory must be available for *reading and writing* for all operation system's users, as them the **License Manager** will run. Click **Next** to continue the installation. ***Note:*** this step might be skipped if use of old settings was selected.
6	A dialog box will open with a message about readiness to install the **License Manager** on your computer:

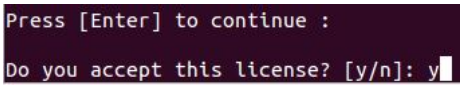
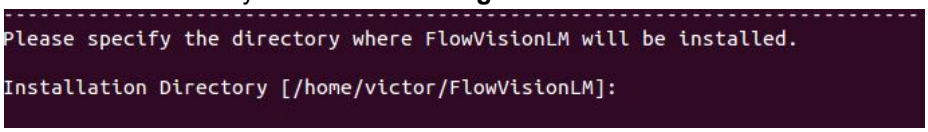
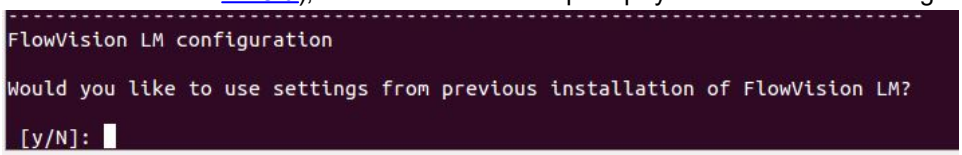
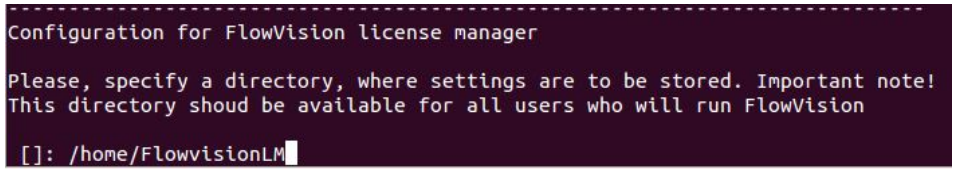
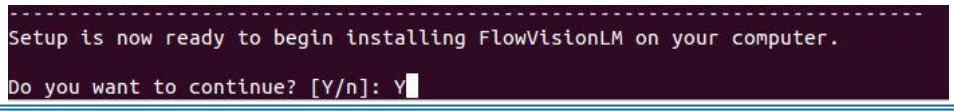
Step	Actions
	 Click **Next** to continue the installation.
7	After completing the installation of the **License Manager** a dialog box opens with a message about it:  Click **Finish**.

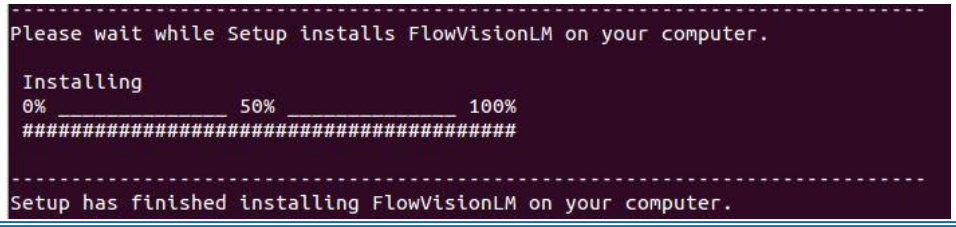
4.5.2.4 Installing License Manager on Linux (in text mode)

Step	Actions

1	Run the installer of the **License Manager** (`FlowVisionLM-linux-x64-installer.run`) in the text mode:
	`./FlowVisionLM-Linux-x64-installer.run --mode text`
	The program will prompt you to read the license agreement. To continue, press the **Enter** key on your computer's keyboard.
	 
	Note 1: If the installer does not run, make sure that its file (`FlowVisionLM-linux-x64-installer.run`) is executable. If no, grant appropriate permission for it.
	To view and grant the permissions, you can use commands of *Linux*. Navigate to the directory, which contains the installer's file and run the command for displaying detailed information:
	`ls -l`
	Permissions of the file of the **License Manager**'s installer (`FlowVisionLM-linux-x64-installer.run`) must contain symbols `"x"`, as in the illustration below:
	 
	To make the file executable, use the command `chmod +x`. Example:
	`chmod +x FlowVisionLM-linux-x64-installer.run`
	Note 2: If your system has no graphical mode, you can omit the `--mode text` parameter in the command line, which runs the installer.

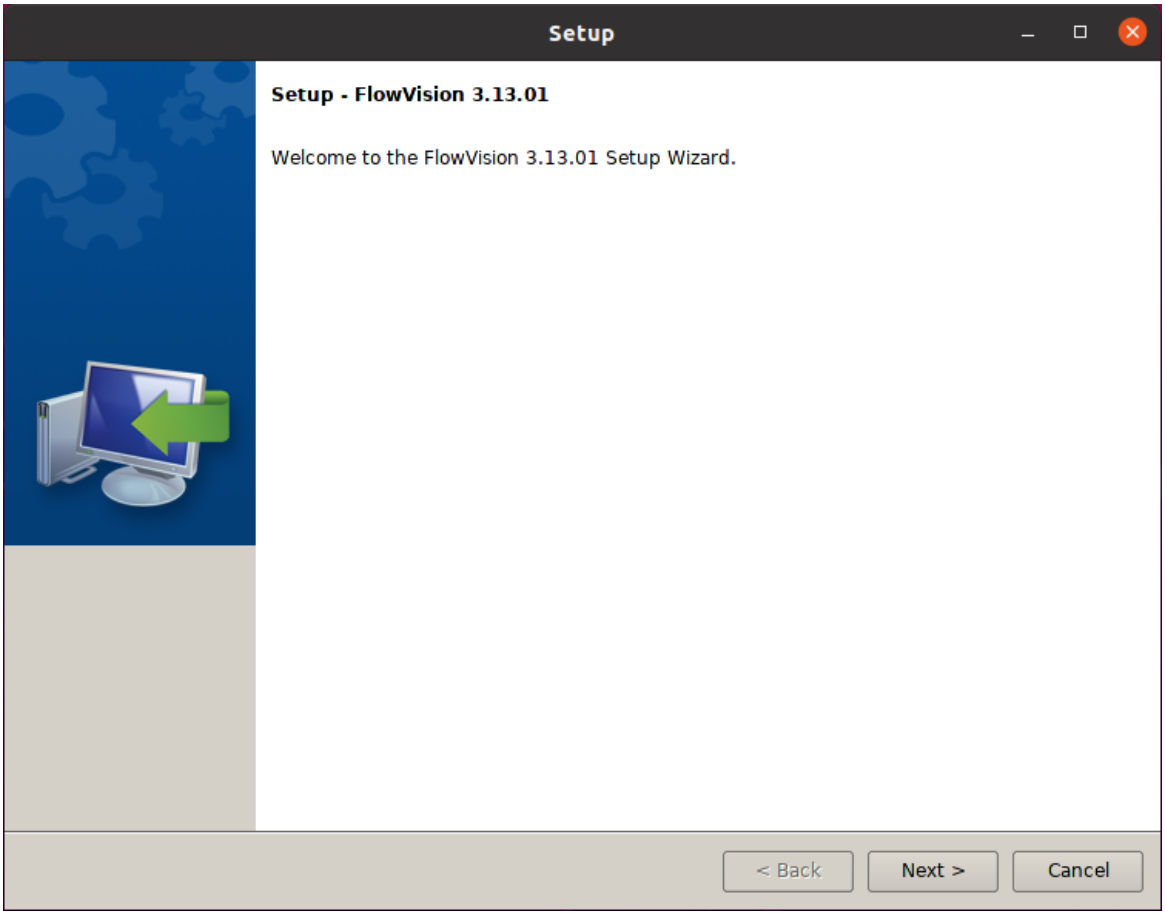
2	The license agreement will be displayed on the screen page by page. To display a next page of the license agreement, press the **Enter** key on the keyboard. Read the license agreement carefully.  ... After the last page of the license agreement, a question about your consent to the its conditions will be displayed:

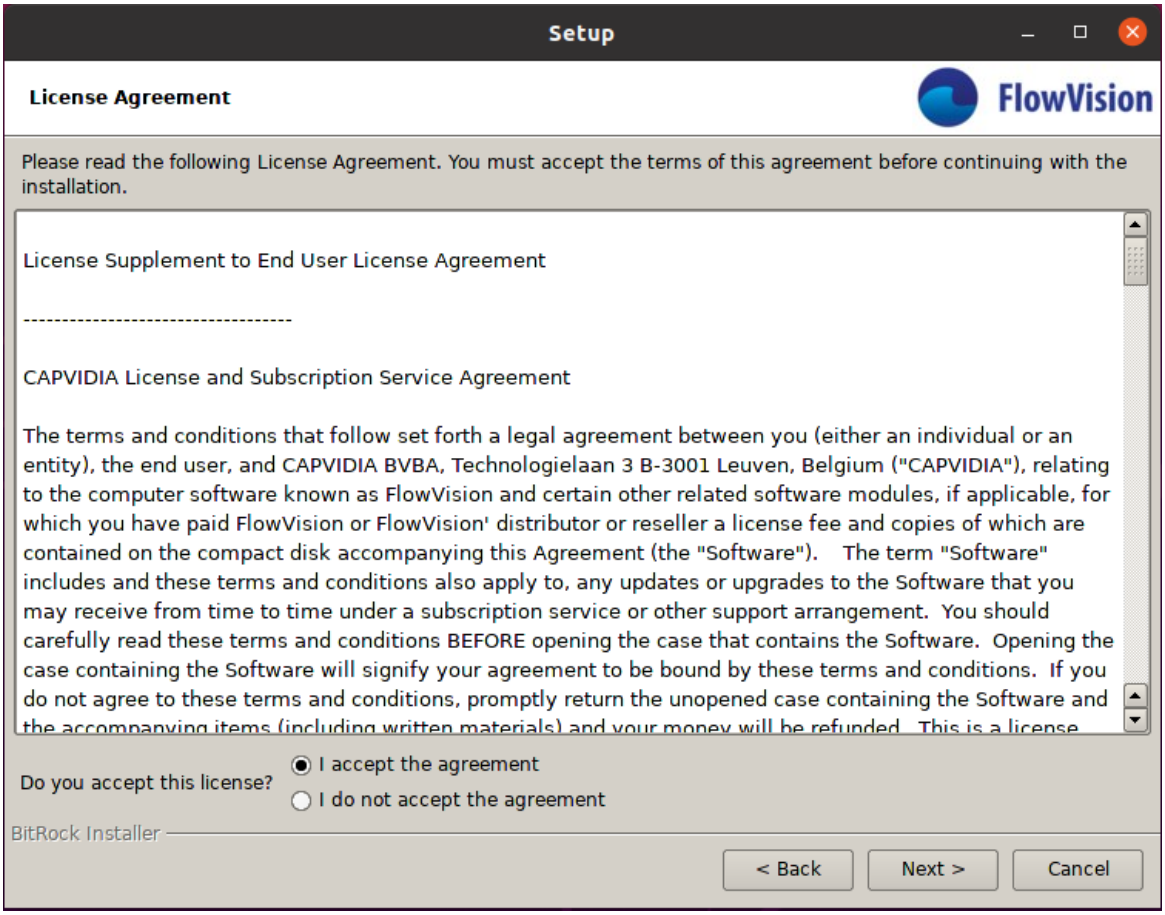
Step	Actions
	 If you agree with conditions of the license agreement, type "**y**" and press the **Enter** key on the keyboard. ***Note:*** this step might be skipped if the installer finds an already installed **License Manager** (for example, when the installer runs to recover damaged files).
3	Specify the installation directory for **License Manager**:  **License Manager** can be installed in an network location, which is closed for ordinary users of the operating system, as all other *FlowVision*'s modules interact with it via a network connection. The default directory's name (**/home/user/FlowVisionLM**) is displayed in square brackets. It will be used if you enter nothing but just press the **Enter** key.
4	If in the directory, which you have specified to install **License Manager**, the installer detects settings from the previous installation of **License Manager** (information about location of these settings is stored in the file **Fv.cfd**), then the installer will prompt you to use these settings:  If you answer positively to the question about use of the old settings (enter "**y**"and then press the **Enter** key), then *Step 5* will be skipped, and the installer will inform you that the installation is ready to start and will go to *Step 6.* If you answer negatively (enter "**N**" and then press **Enter**), the installer will not read settings from the previous installation of **License Manager**.
5	Specify a directory, which will store settings and log files (the user's directory) of the **License Manager**:  This directory must be available for *reading and writing* for all operation system's users, as them the **License Manager** will run. The default name of this directory is displayed in square brackets. It will be used if you enter nothing but just press the **Enter** key. ***Note:*** this step might be skipped if use of old settings was selected.
6	Confirm the start of the installation (type **y** and press the **Enter** key on your computer's keyboard): 
7	Progress of the installation will be displayed on the screen. Then a message will be displayed informing you about completion of the installation:

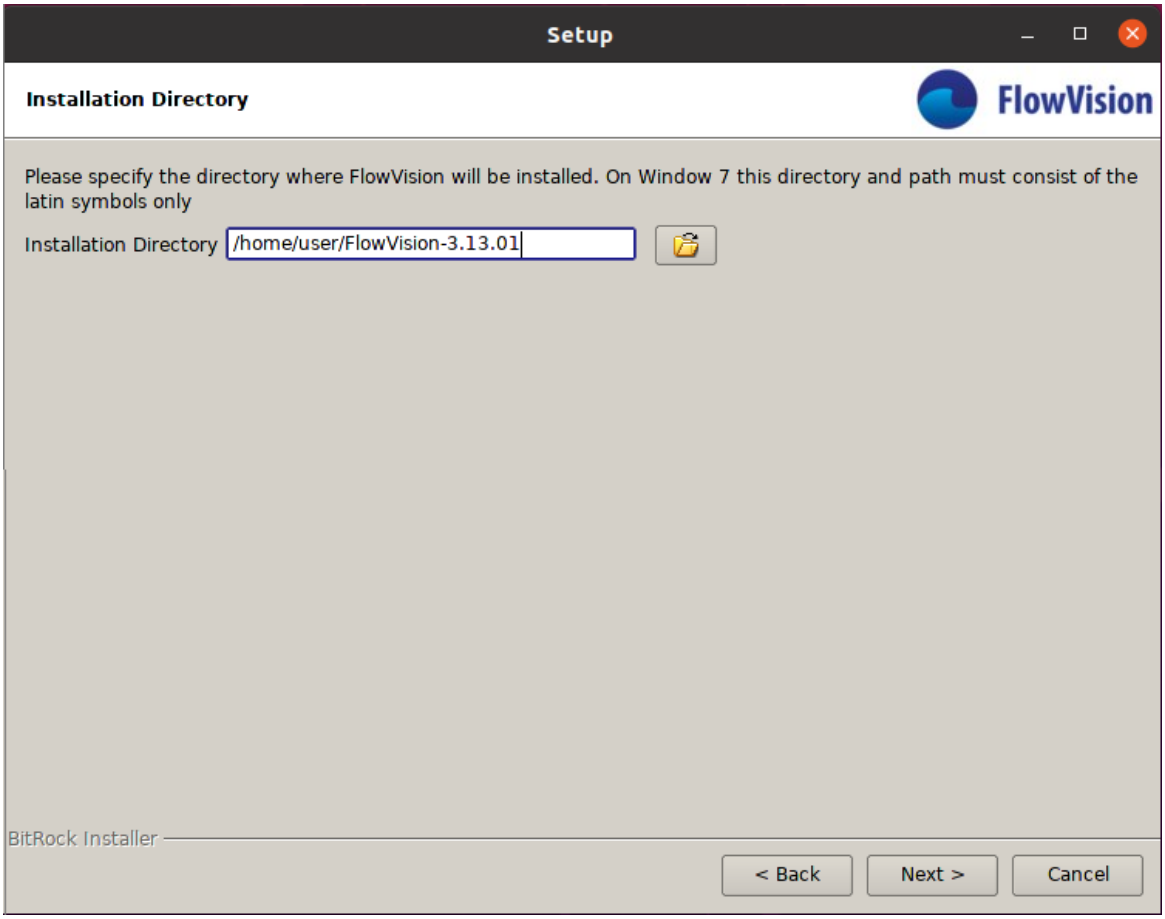
Step	Actions
	

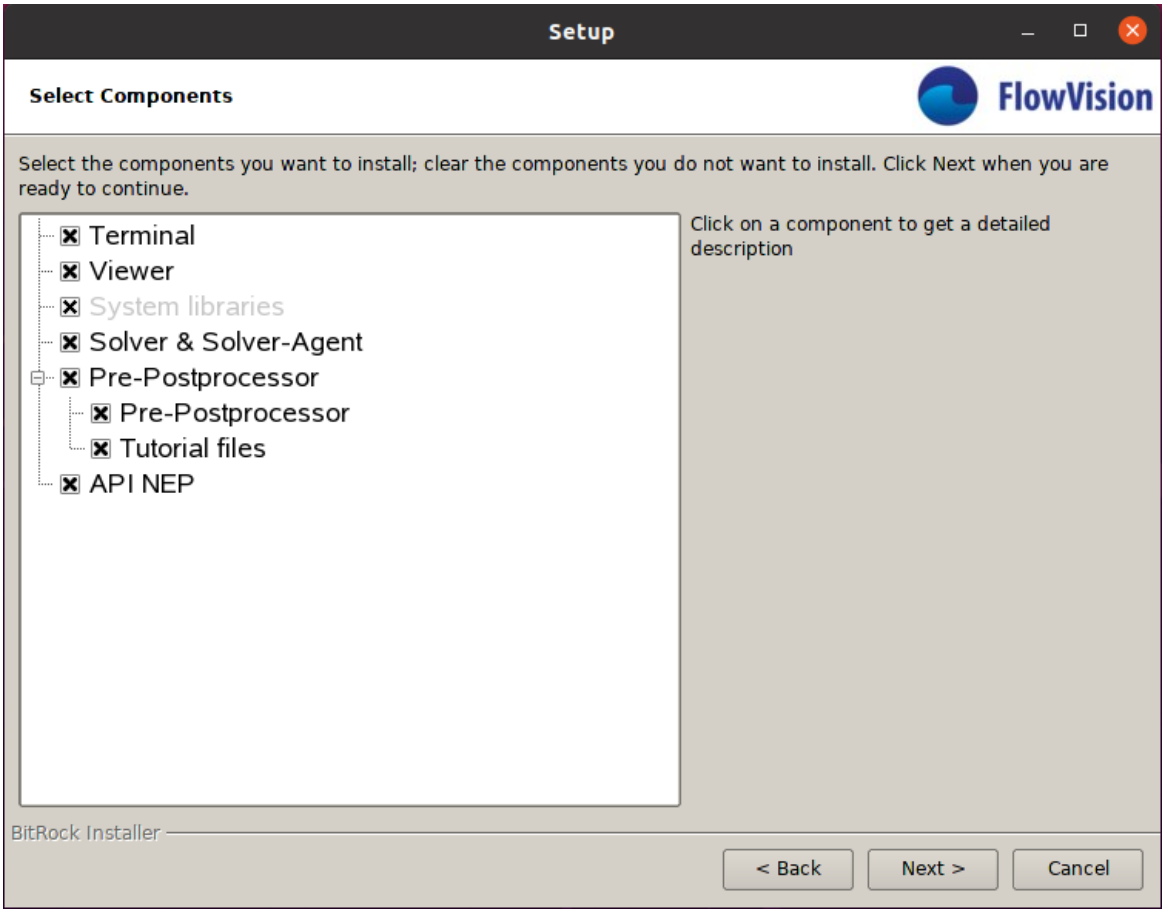
4.5.2.5 Installing FlowVision modules (except License Manager) under Linux in graphical mode

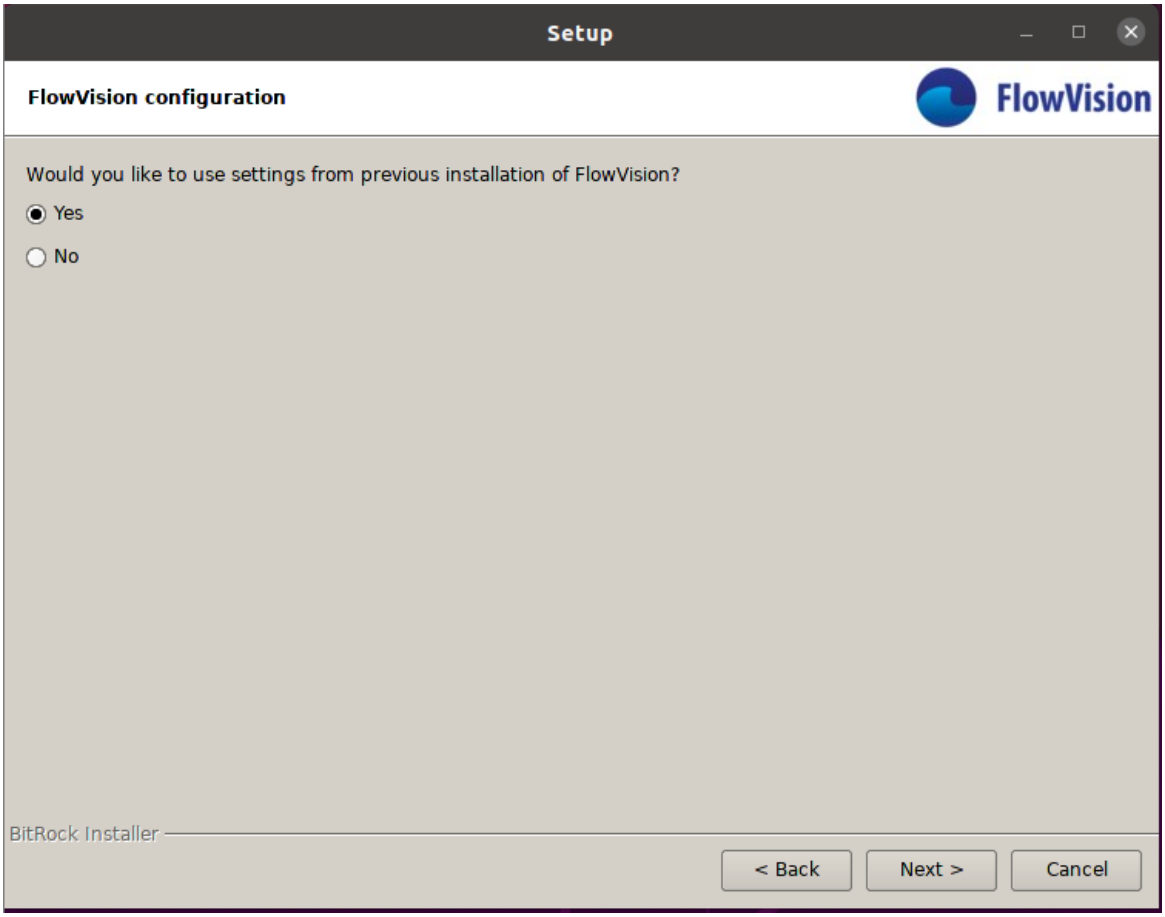
Here, the term "installation of *FlowVision*" refers to the installation of all *FlowVision*'s components except **License Manager**.

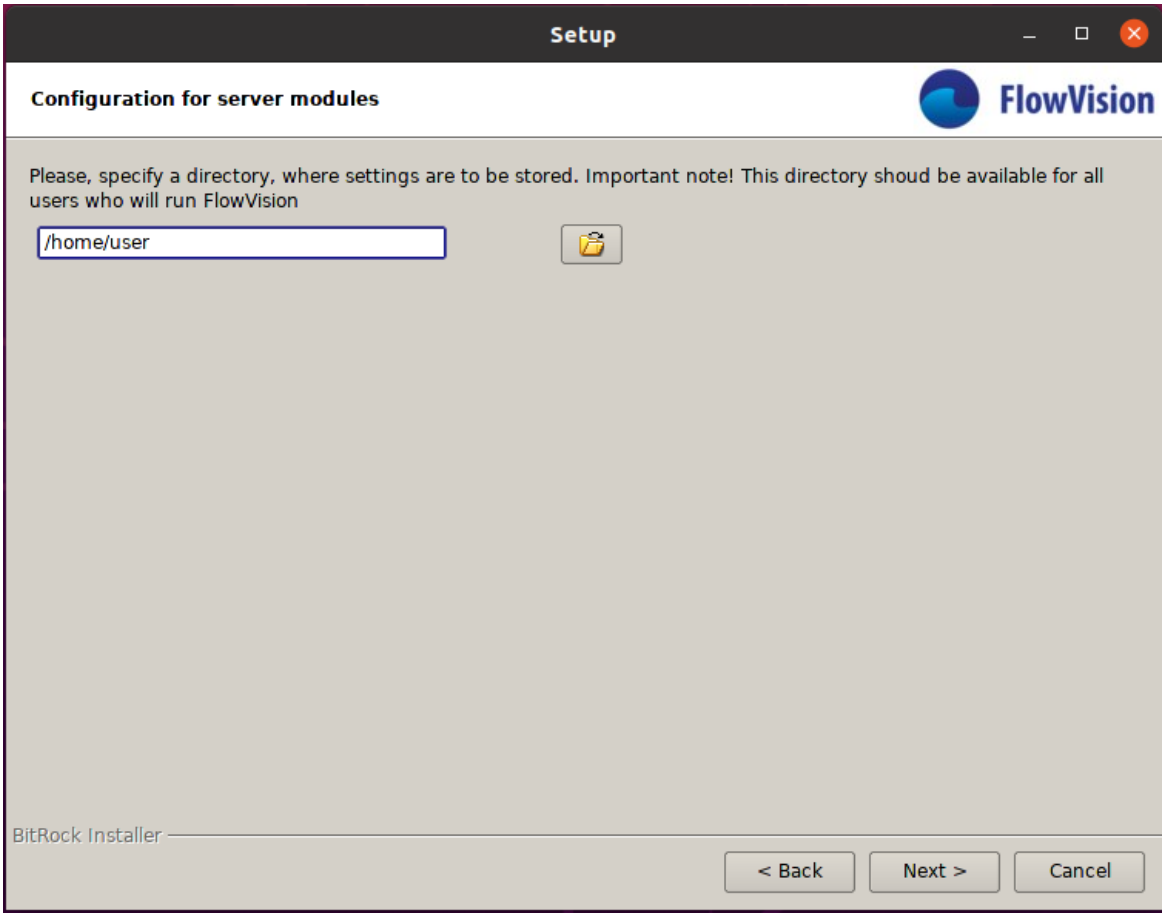
Step	Actions
1	Run the main *FlowVision*'s installer (`FlowVision-3.13.01-linux-x64-installer.run`, the digits in the file name indicate the version number). A welcome dialog box will open:  Click **Next** to continue the installation. ***Note:*** if the installer does not run, make sure that its file is executable. If no, grant appropriate permission for it.
2	A welcome dialog box will open displaying the license agreement for *FlowVision*:

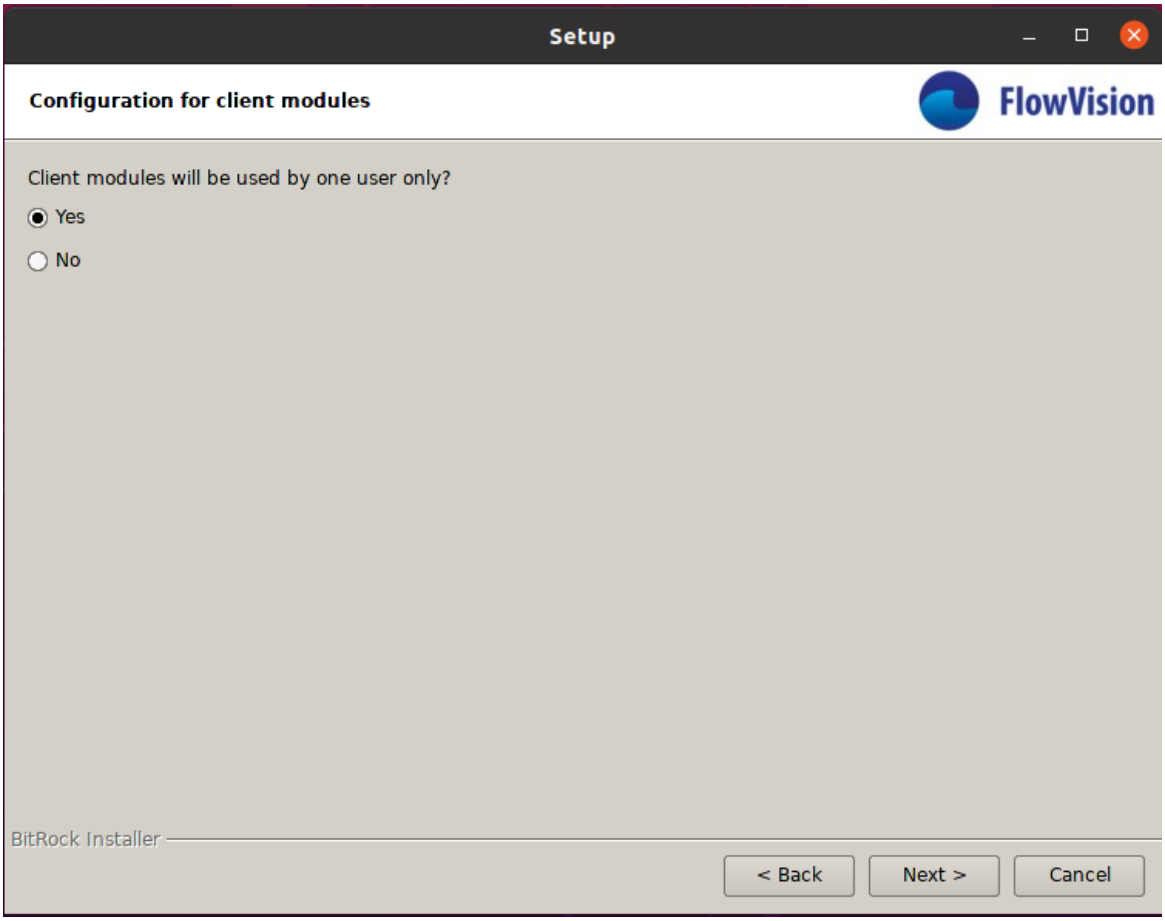
Step	Actions
	 Carefully read the license agreement for *FlowVision* and select the **«I accept the agreement»** option if you agree with conditions of the license agreement. If you do not agree with conditions of the license agreement then, select the **«I do not accept the agreement»** option and abort the installation confirming your decision in a new dialog box, which opens. After you select your agreement with conditions of the license agreement, click **Next**. Then a dialog box will open with the license agreement for *Cluster OpenMP Library* from *Intel*. Confirm you acceptance of the agreement in a similar way.
3	A dialog box will open in which you have to specify the installation directory for *FlowVision* modules except **License Manager** (by default this is a directory, which name is formed by the version number of the program, for example, `/home/user/FlowVision-3.13.01`):

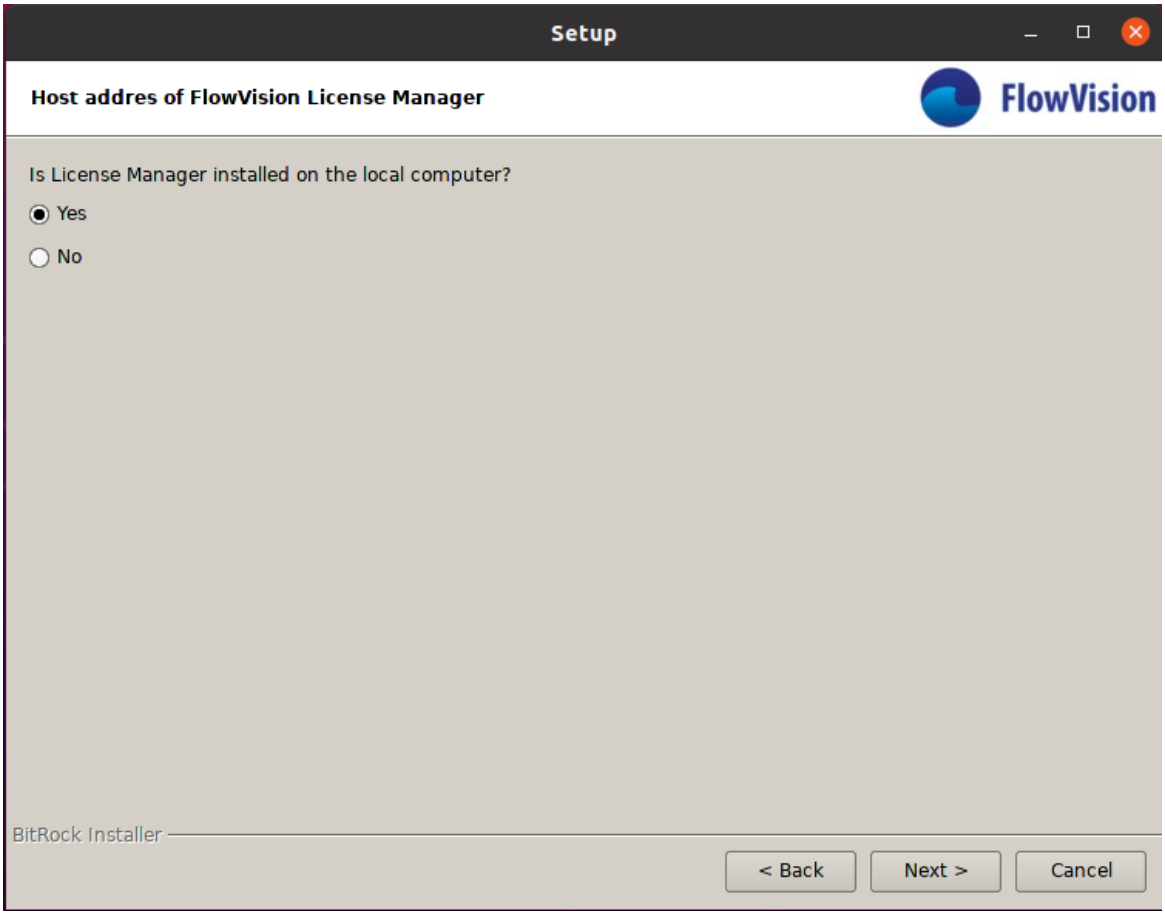
Step	Actions
	 Click **Next** to continue the installation. If *FlowVision* is yet installed in the specified directory, the installation will go in the update mode.
4	Select *FlowVision* modules, which will be installed:

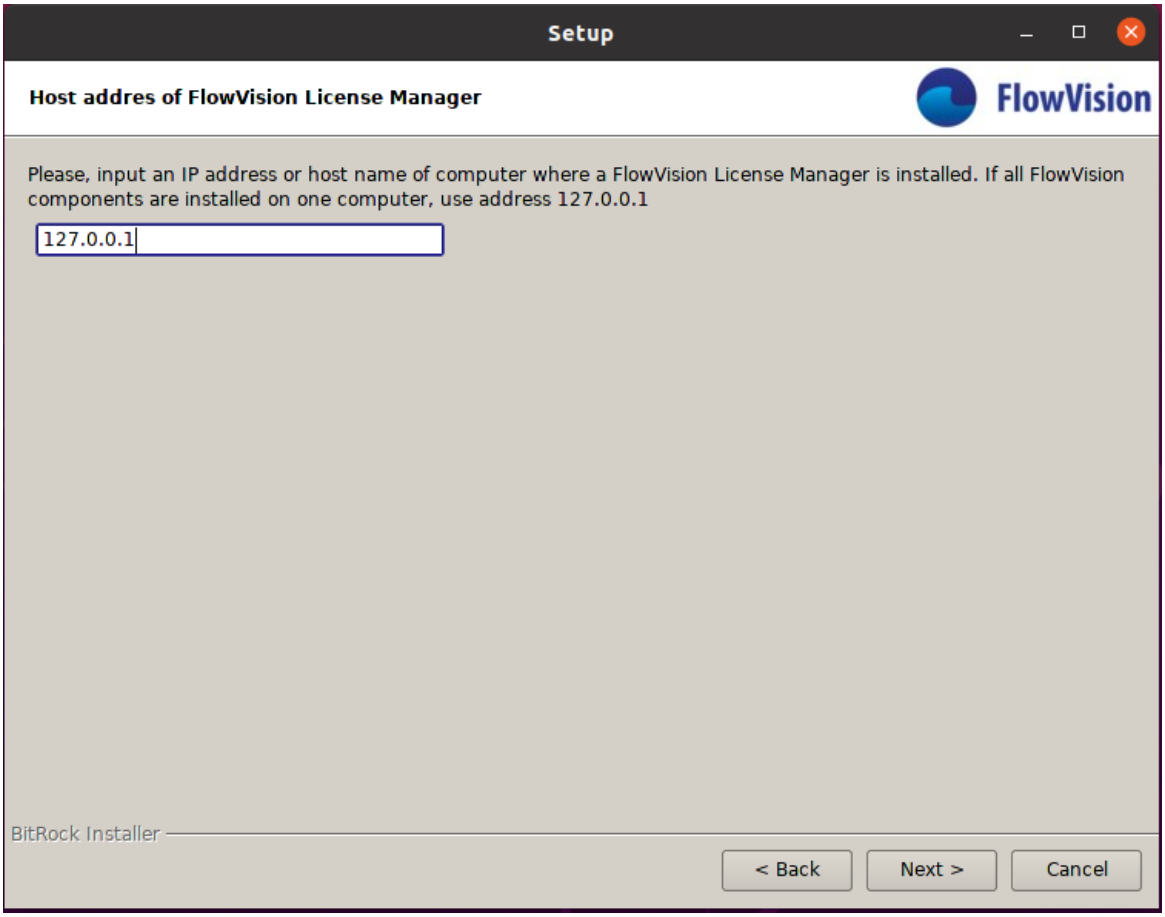
Step	Actions
	 ***Note:*** the **Pre-Postprocessor** module is not available for installation under *Linux*. Brief description of the highlighted module is displayed in the right part of the dialog box. Click **Next** to continue the installation.
5	If in the directory, which you have specified for installation of the *FlowVision*'s modules, the installer detects settings from the previous installation (stored in the file **Fv.cfd**), the installer will offer you to use these settings:

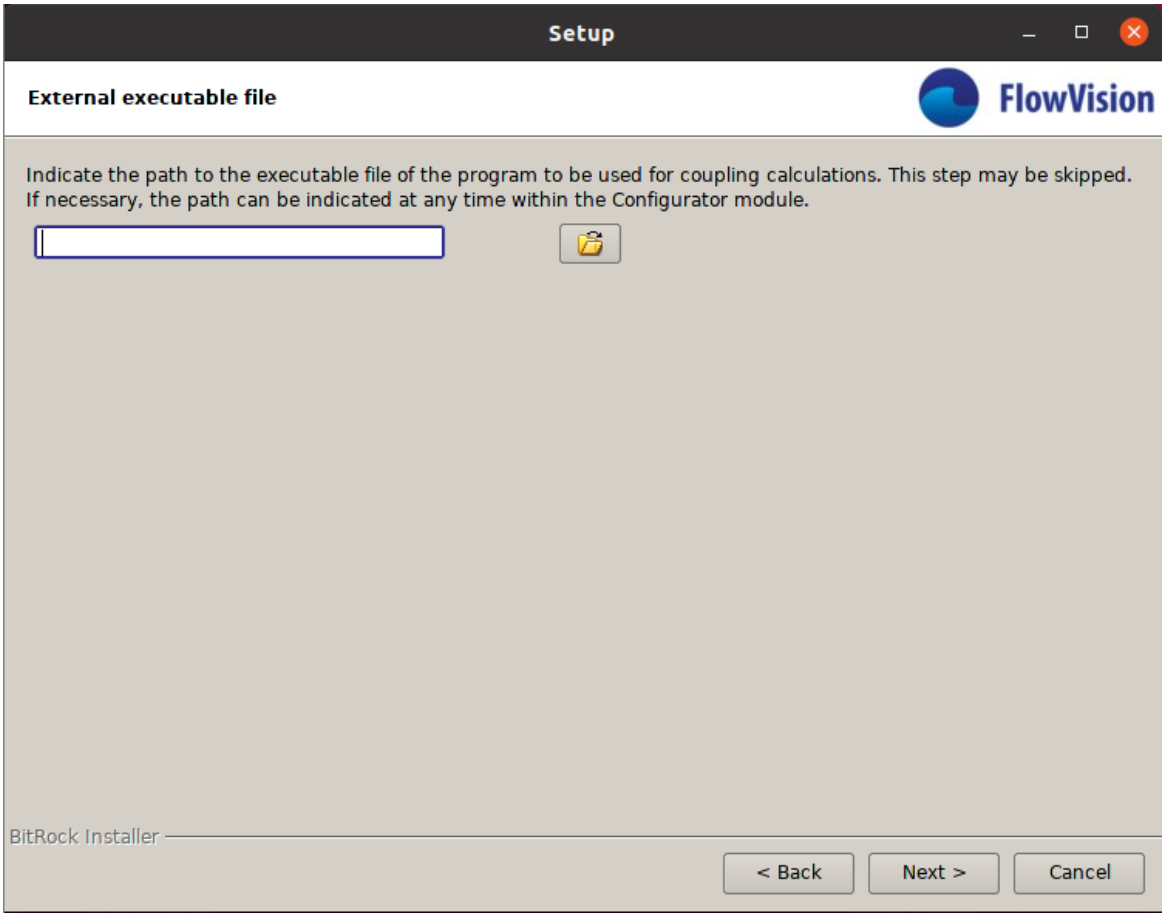
Step	Actions
	 If you answer **"Yes"**, some of the next steps might be skipped. If you answer **"No"**, the installer will not use settings from the previous installation. Click **Next** to continue the installation.
6	Specify the directory, which will store settings of server modules and log files (the user's directory):

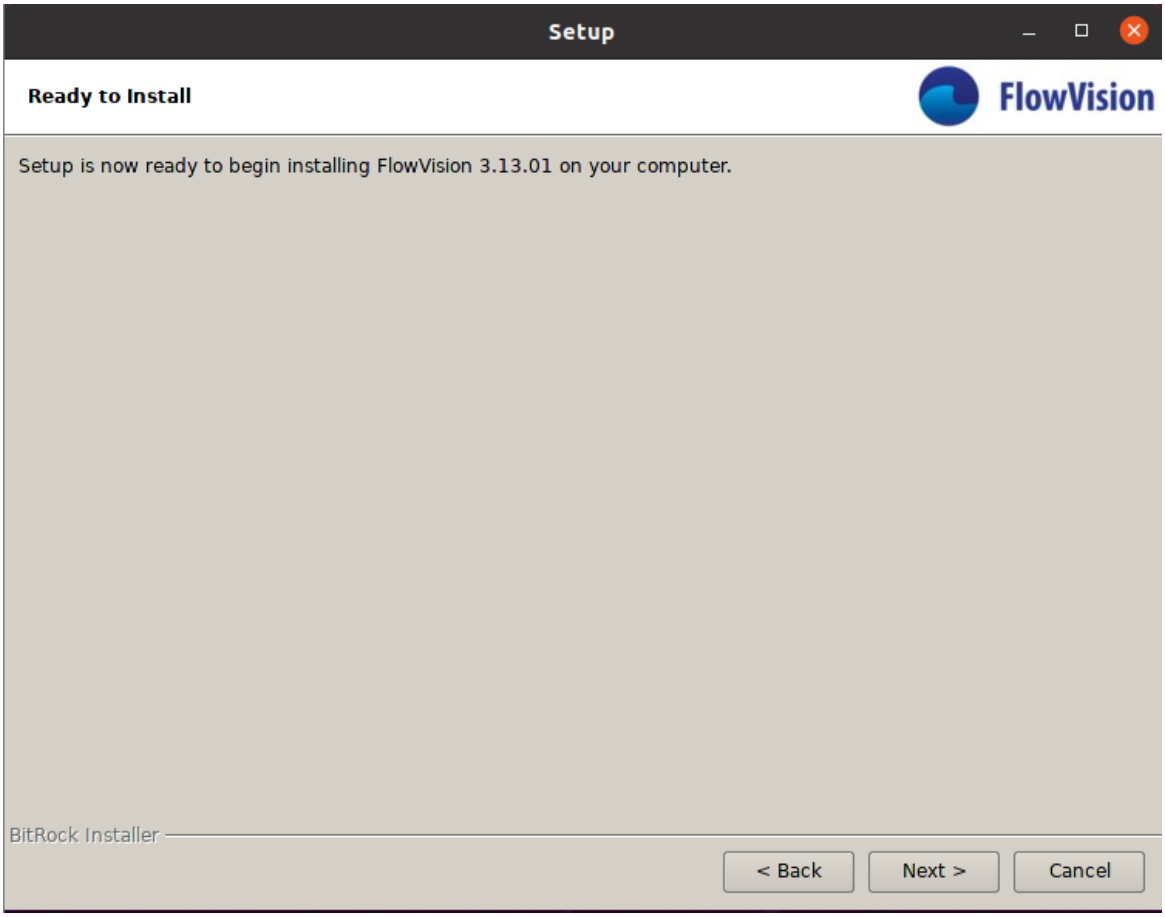
Step	Actions
	 This directory must be accessible for *reading and writing* for all users, as them *FlowVision*'s modules will run. Click **Next** to continue the installation. ***Note:*** this step might be skipped if use of old settings has been selected.
7	Specify whether *FlowVision*'s client modules would be used by a single user or by multiple users:

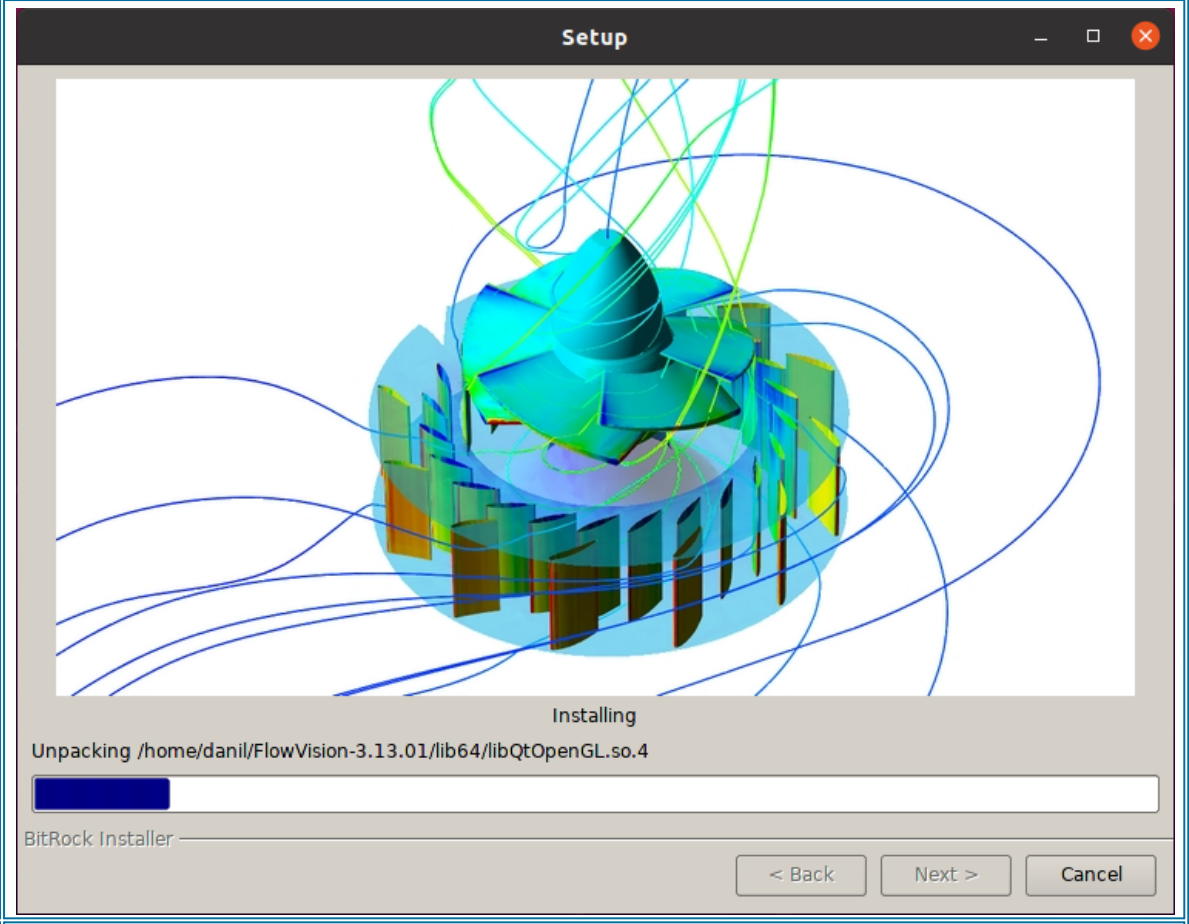
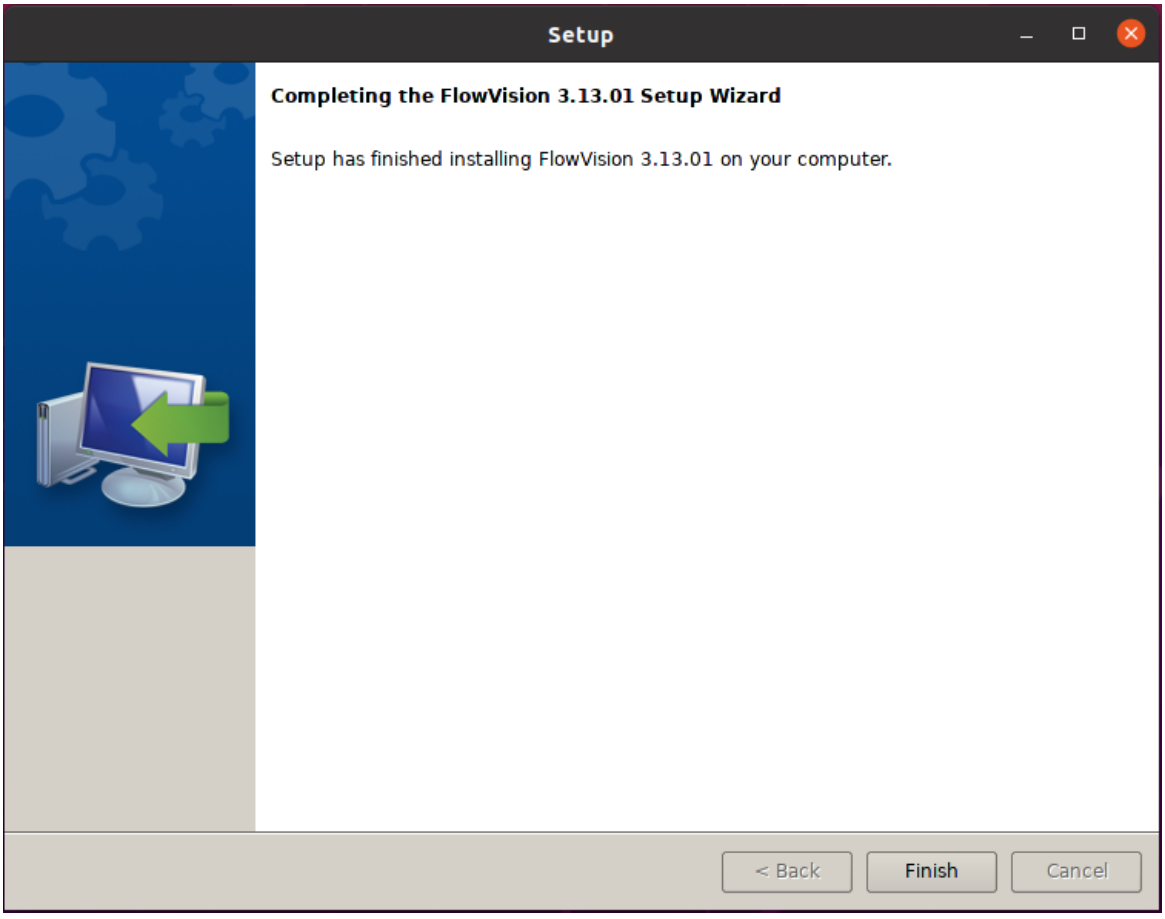
Step	Actions
	 If client modules will be used by *several different* users of the operating system, then it is recommended to select **No**. In this case, all client module settings are stored in users' home directories. This selection allows individual user's settings for *FlowVision* modules. If you select **Yes**, then settings of the client modules will be stored in the directory, which was previously selected to store settings of the server modules. Click **Next** to continue the installation. ***Note:*** this step might be skipped if use of old settings has been selected.
8	Specify the IP address or host name of the computer, on which **License Manager** is installed. The program will ask you first if **License Manager** is installed on the same computer:

Step	Actions
	 If you answer **Yes**, the IP address will be automatically set as **127.0.0.1**. If you answer **No**, the program will request the IP address or host name of the computer, on which **License Manager** is installed:

Step	Actions
	 (by default the program will prompt IP address **127.0.0.1**, which corresponds to the computer, on which the installation of *FlowVision* is running) Enter the host name or IP address and click **Next** to continue the installation. When you use a license with per-minute charge, specify the IP address, which you received after purchasing the license. ***Notes:*** • This step is only performed if **Solver** and **Solver-Agent** have been selected for the installation. • This step is skipped in the case when use of settings from the previous version of *FlowVision* has been selected.
9	If you plan to use *FlowVision* for joint computations with an external software (for example, with *Abaqus FEA*), specify the path to the executable file, which is to be run by the **MPM-Agent** module:

Step	Actions
	 Click **Next** to continue the installation (if use of *Abaqus FEA* is not planned or if *Abaqus FEA* has not been installed yet, then you don't have to enter the data in this dialog box). ***Notes:*** • This step is only performed if **MPM-Agent** has been selected for the installation. • This step might be skipped if use of old settings has been selected.
10	A dialog box will open with a message about readiness to start the installation of *FlowVision*'s modules:

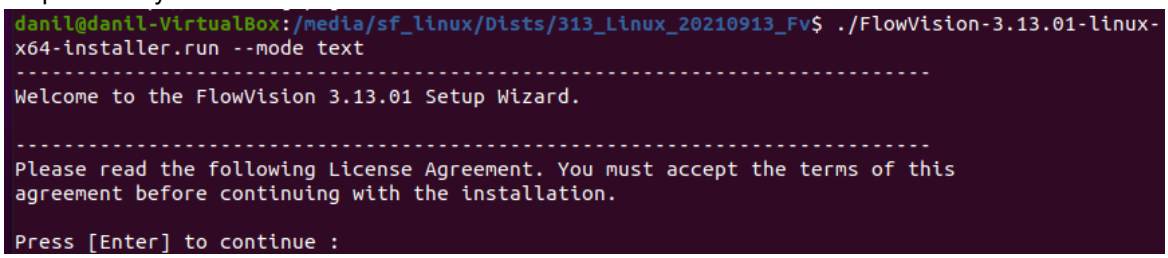
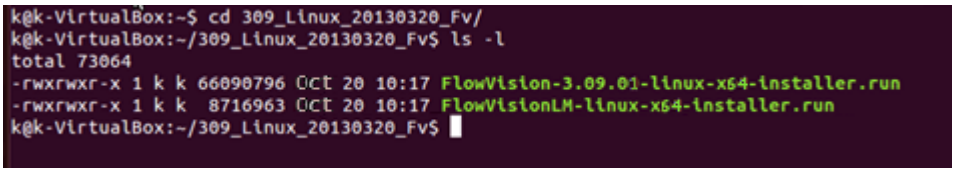
Step	Actions
	 Click **Next** to continue the installation.
11	A dialog box will open with changing illustrations of the *FlowVision*, displaying the installation process:

Step	Actions
	
12	Upon completion of the installation, a dialog box will open:  Click **Finish**.

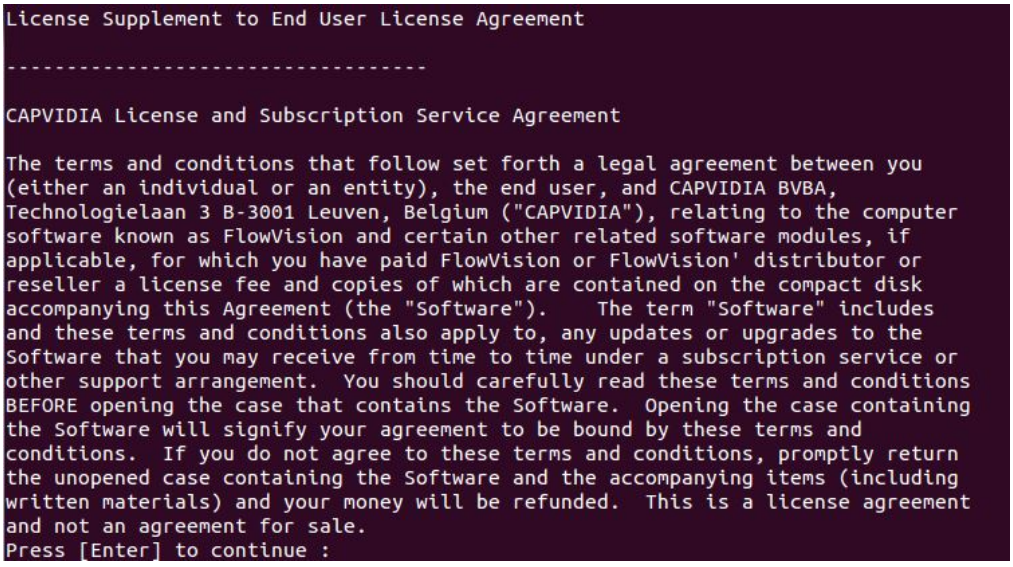
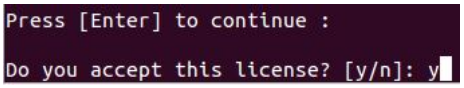
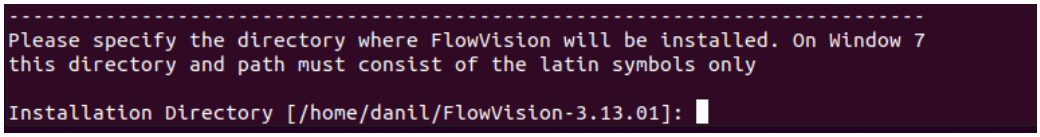
4.5.2.6 Installing FlowVision modules (except License Manager) under Linux in text mode

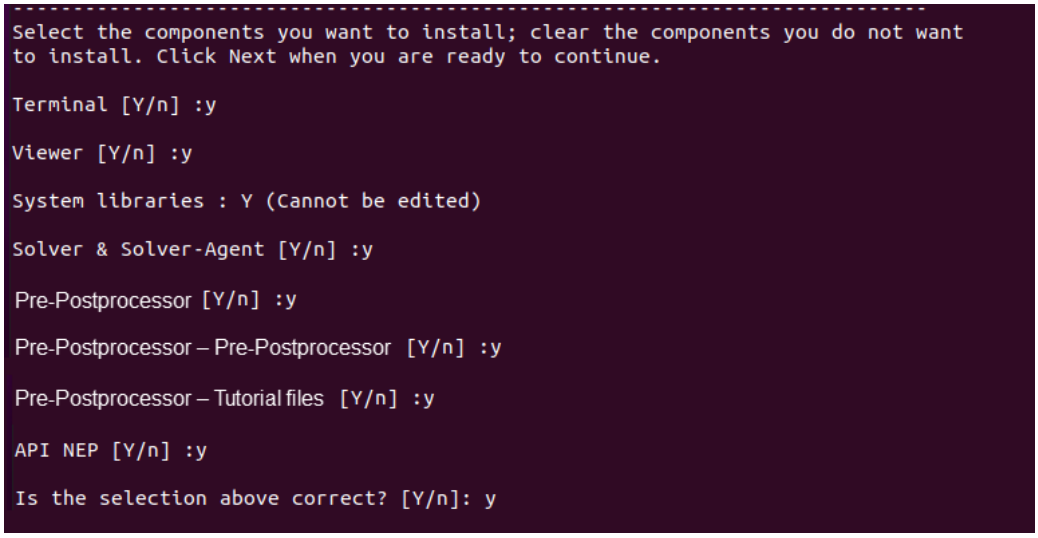
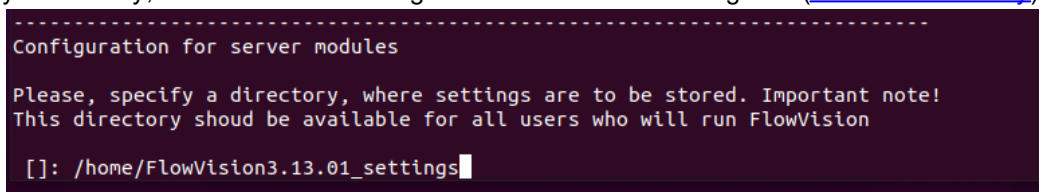
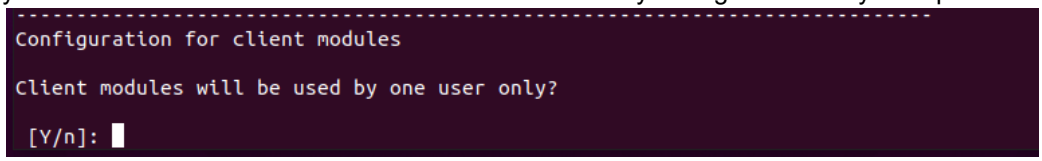
Here, the term "installation of *FlowVision*" refers to the installation of all *FlowVision*'s components except **License Manager**.

Step	Actions

1	Run the main *FlowVision*'s installer in the text mode *(digits in the file name depend on the version)*:
	`./FlowVision-3.13.01-linux-x64-installer.run --mode text`
	The program will prompt you to read the license agreement. To continue, press the **Enter** key on your computer's keyboard.
	 
	Note 1: If the installer does not run, make sure that its file is executable. If no, grant appropriate permission for it.
	To view and grant the permissions, you can use commands of *Linux*. Navigate to the directory, which contains the installer's file and run the command for displaying detailed information:
	`ls -l`
	Permissions of the file of the main installer (`FlowVision-3.13.01-linux-x64-installer.run`, digits in the file name depend on the version) must contain symbols `"x"`, as in the illustration below:
	 
	To make the file executable, use the command `chmod +x`. Example *(digits in the file name depend on the version)*:
	`chmod +x FlowVision-3.13.01-linux-x64-installer.run`
	Note 2: If your system has no graphical mode, you can omit the `--mode text` parameter in the command line.

2	The license agreement will be displayed on the screen page by page. To display a next page of the license agreement, press the **Enter** key on the keyboard. Read the license agreement carefully.

Step	Actions
	 ... After the last page of the license agreement, a question about your consent to the its conditions will be displayed:  If you agree with conditions of the license agreement, type "**y**" and press the **Enter** key on the keyboard. ***Note:*** this step might be skipped if the installer finds an already installed *FlowVision* (for example, when the installer runs to recover damaged files).
3	Specify the installation directory for *FlowVision* modules (except **License Manager**):  To continue, press the **Enter** key on your computer's keyboard. The default directory's name is displayed in square brackets. It will be used if you enter nothing but just press the **Enter** key.  If *FlowVision* is yet installed in the specified directory, the installation will go in the update mode.
4	Select *FlowVision* modules, which will be installed. For each module answer either "**y**" to install it or "**n**" to not install, and then press **Enter**:

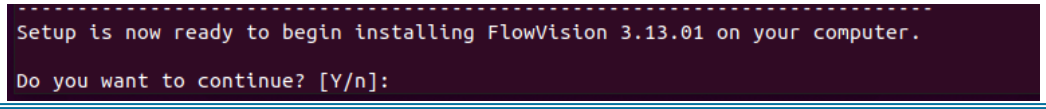
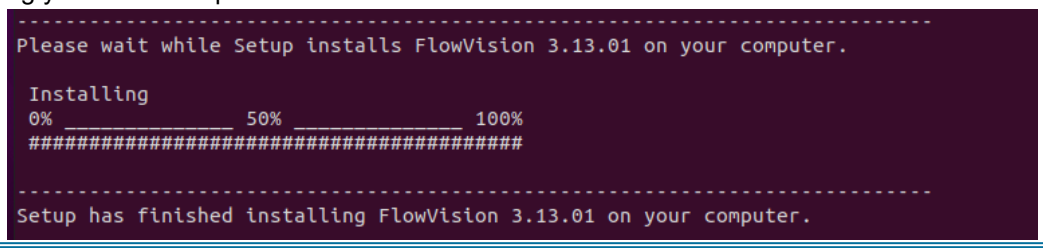
Step	Actions
	 Then, if the list of the modules, which you wish to install, is correct, confirm your selection by typing "**y**"and pressing the **Enter** key as response to the question "**Is the selection above correct? [Y/n]**". ***Note:*** the **Pre-Postprocessor** module is not available for installation under *Linux*.
5	If in the directory, which you have specified for installation of the *FlowVision*'s modules, the installer detects settings from the previous installation (stored in the file **Fv.cfd**), the installer will offer you to use these settings:  After your positive answer, some of the next steps might be skipped. After your negative answer, the installer will not use settings from the previous installation.
6	Specify a directory, which will store settings of server modules and log files (the user's directory):  The default directory's name is displayed in square brackets. It will be used if you enter nothing but just press the **Enter** key. This directory must be accessible for *reading and writing* for all users, as them *FlowVision*'s modules will run. ***Note:*** this step might be skipped if use of old settings has been selected.
7	Specify whether *FlowVision*'s client modules would be used by a single user or by multiple users:  If client modules will be used by *several different* users of the operating system, then it is recommended to answer "**n**". In this case, all client module settings are stored in users' home directories. This selection allows individual user's settings for *FlowVision* modules. If you answer "**y**", then settings of the client modules will be stored in the directory, which was previously selected to store settings of the server modules.

Step	Actions
	Enter your selection and then press **Enter** to continue the installation. ***Note:*** this step might be skipped if use of old settings has been selected.

8	Specify the IP address or host name of the computer, on which **License Manager** is installed.
	The program will ask you first if **License Manager** is installed on the same computer:
	`Host addres of FlowVision License Manager`
	`Is License Manager installed on the local computer?`
	`[Y/n]:`
	If you answer is positive (`Y`), the IP address will be automatically set as **127.0.0.1**.
	If you answer is negative (`n`), the program will request the IP address or host name of the computer, on which **License Manager** is installed:
	`Host addres of FlowVision License Manager`
	`Is License Manager installed on the local computer?`
	`[Y/n]: n`
	`Host addres of FlowVision License Manager`
	`Please, input an IP address or host name of computer where a FlowVision License Manager is installed. If all FlowVision components are installed on one computer, use address 127.0.0.1`
	`[127.0.0.1]:`
	(By default the program will prompt IP address **127.0.0.1**, which corresponds to the computer, on which the installation of *FlowVision* is running. The program will use this address if you enter nothing and just press the **Enter** key.)
	Enter the host name or IP address and press **Enter** to continue the installation.
	  When you use a license with per-minute charge, specify the IP address, which you received after purchasing the license.
	Notes:
	• This step is only performed if **Solver** and **Solver-Agent** have been selected for the installation.
	• This step might be skipped if use of old settings has been selected.

9	If you plan to use *FlowVision* for joint computations an external software (for example, with *Abaqus FEA*), specify the path to the executable file, which is to be run by the **MPM-Agent** module, for example `/home/6.12-3/code/bin/abq6123.exe` and then press **Enter**.
	If you don't plan to use *FlowVision* for joint computations with *Abaqus FEA* or if *Abaqus FEA* has not been installed yet, then you don't have to enter any data (just press the **Enter** key).
	`External executable file`
	`Indicate the path to the executable file of the program to be used for coupling calculations. This step may be skipped. If necessary, the path can be indicated at any time within the Configurator module.`
	`[]:`
	Notes:
	• This step is only performed if **MPM-Agent** has been selected for the installation.
	• This step might be skipped if use of old settings has been selected.

10	Confirm the start of the installation (type `y` and press the **Enter** key on your computer's keyboard):

Step	Actions
	
11	Progress of the installation will be displayed on the screen. Then a message will be displayed informing you about completion of the installation: 

4.5.2.7 Preparing Solver for work with an installed MPI

By default, **Solver** is ready to run in multiprocessor mode (with *MPI*) using *Intel MPI*.

However *FlowVision* can work with other versions of *MPI*. To use another version of *MPI*, do the following:

Step	Actions
•	Copy the *MPI* library, which you wish to use, into subdirectory **`lib64`** in the *FlowVision*'s installation directory, with name **`libFvMPI.so`**. (The subdirectory **`lib64`** contains several ready-made libraries for work with 64-bit *MPI*s. When you wish to use one of them, make its copy in this subfolder and then rename this copy as **`libFvMPI.so`**.)

See also subsection *"Support of MPI"* in section Specific system requirements for Linux and section Compilation a library for interconnection with various MPI implementations.

4.5.2.8 Compilation a library for interconnection with various MPI implementations

The distribution pack contains several ready-made libraries for interconnection with various versions of *MPI* (see details in section Specific system requirements for Linux). If in your system another version of *MPI* is installed, you can compile by yourself a library for interconnection with a required *MPI* version.

The distribution pack also contains the source code of the library **`libFvMPI.so`** (in the directory **`./additional/LibFvMPI`** in the *FlowVision*'s installation directory). To compile an *MPI* library it is necessary that an appropriate *MPI* distribution pack is installed in your system.

To work with other *MPI* implementations, you can independently compile the required version **`libFvMPI.so`** by yourself.

The supplied makefile assumes use of wrappers of system compilers (**`mpicc`**, **`mpiicc`**), which automatically write in the environment paths to header files and libraries.

For normal operation of the library is not required an accurate (up to the last digit) coincidence of version's number of *MPI*, installed in the system, with version's number of *MPI*, with which the library has been compiled. But the library and *MPI* and also the compiler that was used to compile the *MPI* and library are to have the same number of bits.

Compilation a library	
Step	**Actions**
1	Unpack the archive, which contains the source code and makefile: **`tar -xvf 309_distrib.tar.gz`** If necessary, specify in the makefile a path to the compiler (**`mpicc`**) and desired flags.
2	Navigate to the directory, which contains the makefile, and run **`make`**: **`cd ./309_distrib`** **`make`**

Compilation a library	
Step	**Actions**
	As a result, two directories will be created. The compiled library has name **`libFvMPI.so`** and it is located in the directory **`lib64`**.

Editing the makefile

The supplied distribution pack, along with source codes of *libFvMPI*, also contains a makefile to compile libraries. If you don't wish to use *MPI*, which is specified in your environment, you can specify the following variables:

- **`MPI_DIR`** specifies the path to the directory, where the *MPI* has been installed (optional)
- **`MPICC`** specifies the path to the **`mpicc`** wrapper, which will be used for the compilation. By default, **`MPICC = mpicc`** that means use of *MPI*, which is available in the environment. Other possible value is **`mpiicc`** for a combination of *Intel MPI* and *Intel compiler.*

When you use *Intel MPI*, you can pass specific options for compilation and linking (see their description in the compiler's documentation) using variables **`I_CCFLAGS`** and **`I_LFLAGS`**. Among these options there is the static linking with runtime libraries from the *Intel* compiler (**`-i-static`**, **`-static_mpi`**).

4.5.3 Automatic installation

Installers of *FlowVision* (both for *Windows* and for *Linux*) have many keys, which allow an unattended installation of *FlowVision*, for example, when you run the installer using a file with a set of keys.

For a list of all the keys of the installer, run the installer with the key **`--help`**.

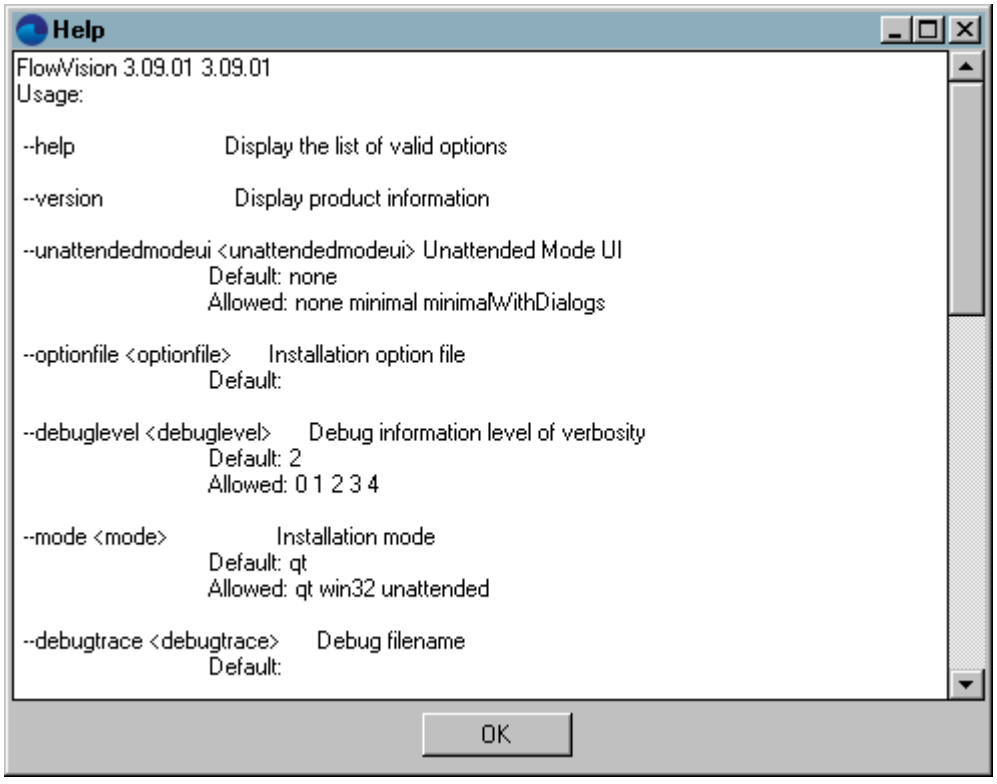
Key	Possible arguments	Description
`--help`	-	Displays a list of possible keys.
`--version`	-	Displays information about the program.
`--unattendedmodeui <unattendedmodeui>` (this key is applied when the **`--mode`** key is used with the argument **`unattended`**, see below)	**`none`** *(the default value)* **`minimal`** **`minimalWithDialogs`**	Settings for unattended installation: • **`none`** - quiet run mode without any graphical output • **`minimal`** - with displaying the installation as a progress bar • **`minimalWithDialogs`** - with displaying the installation as a progress bar and displaying error messages in graphical windows boxes In all these three cases the error messages are also recorded into the file **`InstallationErr.log`** located in the directory, from which the installer was started.
`--optionfile <optionfile>`	*path to a file*	Allows the program to run an unattended installation with keys specified in the referred text file. See details in subsection *"Format of files specified by the --optionfile key"* below.
`--mode <mode>`	**`qt`** *(the default value)* **`unattended`**	Operating mode of the installer: • **`qt`** `is applied` for a windowed (graphical) mode • **`unattended`** specifies an automatic installation (with no user interface) that is convenient for automatic installation, which is done by scripts. When this mode is selected, you can use the **`--unattendedmodeui`** key (see above). When this mode is used, the error messages about the installation progress are recorded into the file **`InstallationErr.log`**

Key	Possible arguments	Description
		located in the directory, from which the installer was started.

`--disable-components <disable-components>`	*A list containing possible arguments separated by commas:*
	`Terminal`
	`FvViewer`
	`FvSolver`
	`PPPGroup`
	`PPP`
	`Tutorial`
	`SDK`
		List of *FlowVision* modules and components, which *must not* be installed.
		You can not disable installation of system libraries (`Libraries`).
		By default all *FlowVision* modules will be installed. When you wish to exclude one or several modules from the installation, use this key with argument(s) referring to the module(s):
		• `Terminal`- **Terminal**
		• `FvViewer`- **Viewer**
		• `FvSolver`- **Solver** and **Solver-Agent**
		• `PPPGroup` - **Pre-Postprocessor** and tutorial (the "*Tutorial: Examples of typical tasks*" document)
		• `PPP`- **Pre-Postprocessor**
		• `Tutorial` - tutorial (the "*Tutorial: Examples of typical tasks*" document)

`--prefix <prefix>`	*In Windows by default the* `C:\Program Files\FlowVision-x.xx.xx` *directory is used, where* `x.xx.xx` *is the version of FlowVision. If you try to install the same version as the existing (this is updating of the program), then by default the directory from the previous installation will be used.* *In Linux by default the* `/home/user/FlowVision- x.xx.xx` *directory is used, where* `user` *is the system name of the user, see section Installation of FlowVision.*	The installation directory
`--usePrevSettingsDialog <usePrevSettingsDialog>`	`0` `1` *(the default value)*	If the program is installed in the directory, where it has already been installed (this is detected by the presence of the file **Fv.cfd**), the this parameter specifies how to proceed: • `0` - override settings from the previous installation. In this case it is necessary to use the `--homeSvrDialog` key (see below). • `1` - use settings from the previous installation
`--homeSvrDialog <homeSvrDialog>`	*Path to the directory, where settings and log files of server modules are stored.*	Directory for storing settings and log files of server modules (user directory). This key must be specified if the `--usePrevSettingsDialog` key is set as `0`.
`--homeClnDialog <homeClnDialog>`	`0` `1` *(the default value)*	The mode of use of *FlowVision* client modules: • `0`- multiuser • `1`- single user

Key	Possible arguments	Description
		If client modules (**Pre-Postprocessor**, **Terminal**, and/or **Viewer**) will be used by *several different* users of the operating system, then it is recommended to specify this key as `0`. In this case, all client module settings are stored in users' home directories. This selection allows use of *FlowVision* modules with individual user's settings. If this key is specified as `1`, then settings of the client modules will be stored in the directory, which was previously selected to store settings of the server modules.
`--lsHostDialog <lsHostDialog>`	*the default value is* `127.0.0.1`	IP address or host name of the computer, on which **License Manager** is installed.
`--AbaqusExecutable <AbaqusExecutable>`	*the default value is an empty string*	The path to an executable file, which starts *Abaqus*.
`--InstallMSMPIDialog <InstallMSMPIDialog>`	`0` *(default value)* `1`	This key specifies if it necessary to install *Microsoft MPI*: • `0` - do not install • `1` - install
`--enable-components`		This key is not used and has no effect on the installation



Result of a running the *FlowVision*'s installer in *Windows* with the **`--help`** key

The most popular cases of use the installer in an automatic mode in the *Windows* environment

In the listed below cases we give examples of installation of all available *FlowVision* modules in the quiet mode (with the key **--unattendedmodeui none**) in *Windows*. The omitted keys are assumed as set to their default values (see description in the first example).

For each of the listed cases you can specify a custom set of unnecessary components (by the **--disable-components** key) and use any desired mode of automatic installation (**--unattendedmodeui**).

Version *FlowVision x.xx.xx* is installed on a computer first time or after its total removal with the installation directory

User installs the version *FlowVision-x.xx.xx*. This version of the program has never been installed on the computer or has been totally removed with the installation directory. If you don't specify the path to the installation directory, to which the program should be installed, after the **--prefix** key, then the program will be installed into the default directory (**C:\Program Files\FlowVision-x.xx.xx**). Also after the **--homeSvrDialog** key you have to specify the directory, which will store settings and log files of server modules.

Here is an example of quiet installation of the program into the directory **C:\Flowvision** with the directory for storing the settings **C:\Flowvision\Settings** if **License Manager** is installed on the computer with the IP address 192.168.1.3 in the local network:

```
FlowVision-3.10.01-windows-installer.exe --prefix C:\Flowvision --mode unattended --
usePrevSettingsDialog 0 --homeSvrDialog C:\Flowvision\Settings --lsHostDialog 192.168.1.3
```

The omitted keys are assumed as set to their default values. For example, in this case, the omitted keys will be applies as follows:

--unattendedmodeui none	quiet mode
--homeClnDialog 1	single user installation
--InstallMSMPIDialog 0	do not install *Microsoft MPI*

If you need other values for these keys, you have to specify them explicitly in the command line, which runs the installer.

Installation of version *FlowVision x.xx.xx* after its removal

Version *x.xx.xx* was installed on the computer and then removed by standard means for removing programs. In this case you can use the settings that remained from the previous installation because neither the settings itself no the file **Fv.cfd** (which points to the location where the settings are stored) have not been deleted. The user can specify only the installation directory where the **Fv.cfd** file remaining from the previous installation locates and it is not necessary to specify the directory where settings and log files of server modules are stored.

Here is an example of quiet installation of the program into the directory **C:\Flowvision**, which already contains the file **Fv.cfd**, which remained from the previous installation:

```
FlowVision-3.10.01-windows-installer.exe --prefix C:\Flowvision --mode unattended
```

The omitted keys are assumed as set to their default values as in the previous case. It should be noted that as the omitted key **--usePrevSettingsDialog**, similarly to other omitted keys, assumes its default value, the program uses remaining settings from the previous installation if the **Fv.cfd** file will be found in the directory **C:\Flowvision**.

Updating the existing version *x.xx.xx*

Version *FlowVision x.xx.xx* has been installed on the computer and you install the same version again (update it). In this case you don't have to specify neither the installation directory no the directory for storing settings and log files of server modules as the installation will go over the existing installation.

Example of quiet update of the program in *Windows*:

```
FlowVision-3.10.01-windows-installer.exe --mode unattended
```

The omitted keys are assumed as set to their default values (as in the first case).

Use a text file containing a set of keys

In this case the keys, which define run mode of the installer, are stored in the file **`keysfile.txt`**, which locates in the directory, from which the installer is launched (see details in the subsection *"Format of files specified by the --optionfile key"* below).

Example:

```
FlowVision-3.10.01-windows-installer.exe --optionfile keysfile.txt
```

Format of files specified by the `--optionfile` key

You can store installation options in a file, which is specified by the **`--optionfile`** key. The options are set as records in the "**`name=value`**" format, with one record per line. Lines, which have # as their first non-blank character, are commentaries.

Example of a file with keys for installation all FlowVision components on Linux:

```
################################################################################
# options for a standard installation, but without Pre-Postprocessor           #
# (written by J.Smith, 30 Apr 2017)                                            #
################################################################################
mode=unattended
unattendedmodeui=minimalWithDialogs
prefix=/home/FlowVision-3.10.01
usePrevSettingsDialog=0
homeSvrDialog=/home/FlowVision-3.10.01_settings
lsHostDialog=server12345
```

4.6 The 3DTransVidia software and its installation along with FlowVision

About the *3DTransVidia* software

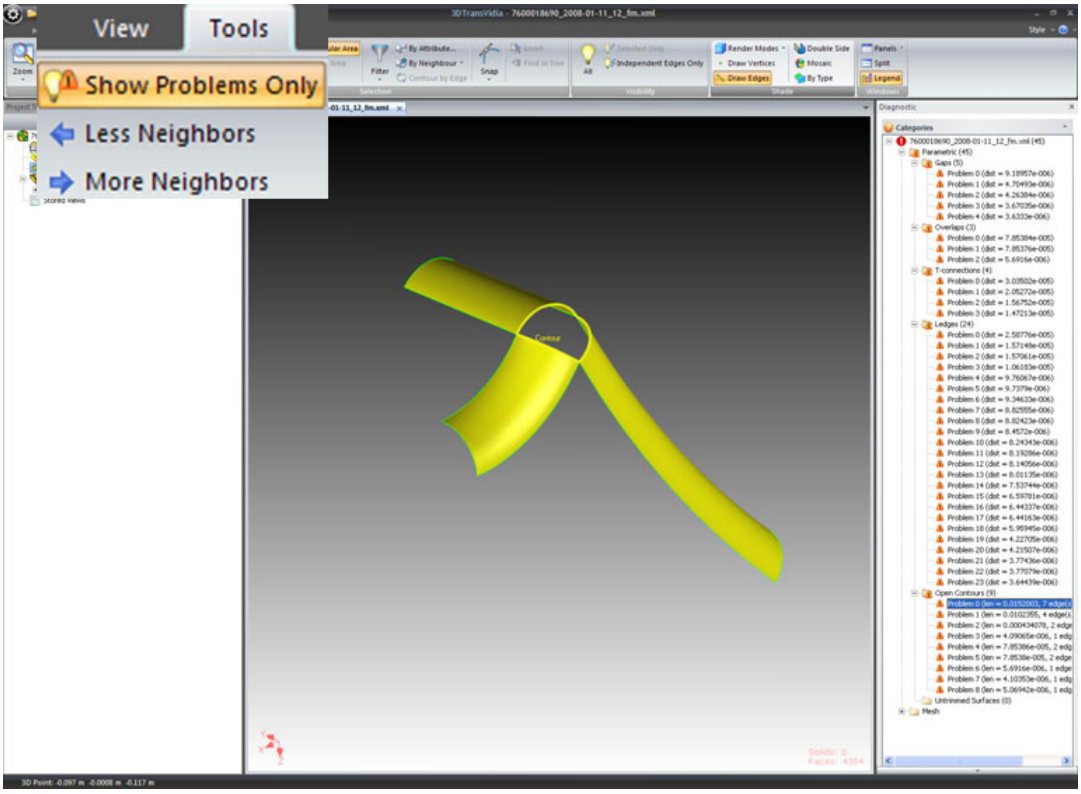
The software package *3DTransVidia* is developed to provide better translation of geometry 3D models between various CAD systems, it allows diagnosing and correcting the 3D models. The diagnostics and fixing process is highly automated and can be applied to separate solids (bodies) and also to assemblies of various degrees of complexity. Correction is always carried out taking into account the model's precision; this keeps the initial integrity of the presentation and prevents any deformations.

3DTransVidia supports importing of geometry and fixing their triangulation errors at transferring the models from CAD systems to *FlowVision*. *3DTransVidia* reads geometry models from neutral formats *IGES, STEP, xt, SAT*.

The models can be prepared for their import into *FlowVision*. *3DTransVidia* allows you to fix errors in CAD models and also in mesh models like *VRML* and *STL*, if they appear. You can fix a geometry model in *3DTransVidia*, check it for errors and then load it into *FlowVision* using a special format **`mesh`**, which is used for data exchange between *FlowVision* and *3DTransVidia*.

3DTransVidia also provides access to the technical requirements' data (*Product Manufacturing Information*) stored in a file.

The style for controlling in the view window in *3DTransVidia* is the same as the style in *FlowVision*.



Window of the *3DTransVidia* software

3DTransVidia is included into the delivery pack of *FlowVision*, but you need appropriate license options to use it.

Use of *3DTransVidia* with *FlowVision*

The version of *3DTransVidia*, which is included into the *FlowVision*'s distribution pack, is optimized and intended for correction of the geometry model and to prepare the model for import into *FlowVision*. Using *3DTransVidia*, you can:

- correct errors of unsatisfactory triangulated geometry model (fix self-intersections, sew holes in the net, remove overlays and small triangles)
- change triangulation of the net surface (smooth it or reduce the number of triangles)
- convert parametric model of any of formats supported in *3DTransVidia* into a triangulated three-dimensional surface with specified accuracy and quality

3DTransVidia, when it is integrated with *FlowVision*, allows exporting surfaces into the **mesh** format only, which is the native format of *FlowVision*. *3DTransVidia* in *FlowVision* is configured so its mouse control for the view would be the same as those, which is used in *FlowVision*.

Combined installation of *3DTransVidia* and *FlowVision*

At the end of the installation *FlowVision* on *Windows*, the *FlowVision*'s installer prompts the user to install also *3DTransVidia*. When the user agrees, *3DTransVidia*'s installer starts.

When started by this method, the *3DTransVidia*'s installer operates in its normal mode and displays all its dialog boxes (except the dialog box where the installation path is selected). The installation is done into the *FlowVision*'s installation directory.

It is possible to install several copies of *3DTransVidia* when you install several versions of *FlowVision*.

Licensing *3DTransVidia* when it operates together with *FlowVision*

When *3DTransVidia* operates together with *FlowVision*, it receives its license information from the *FlowVision*'s module **License Manager**.

Support of various file formats

Depending on the license, which *3DTransVidia* receives from **License Manager**, *3DTransVidia* supports import of the following file formats:

- **STL**, **VRML**, **mesh**, **STEP**, **IGES**, and **AF**
- all supported formats (when a special license is granted)

In both cases the export is possible into the **mesh** format only.

Uninstallation of *3DTransVidia* along with uninstallation of *FlowVision*

At uninstallation of *FlowVision*, if *3DTransVidia* has been installed, *3DTransVidia*'s uninstaller also will start.

The uninstaller will work in its normal mode and will display all its dialog boxes.

Technical support for *3DTransVidia*

For technical support of *3DTransVidia*, please contact the technical support service of FlowVision.

4.7 Recovery of damaged installation and/or changing the list of installed modules

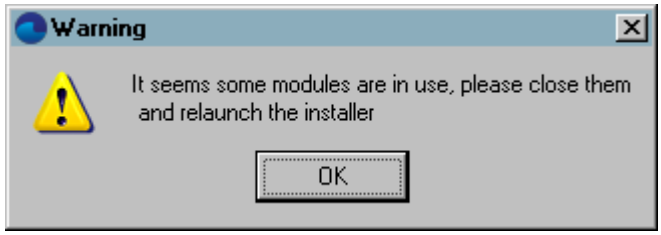
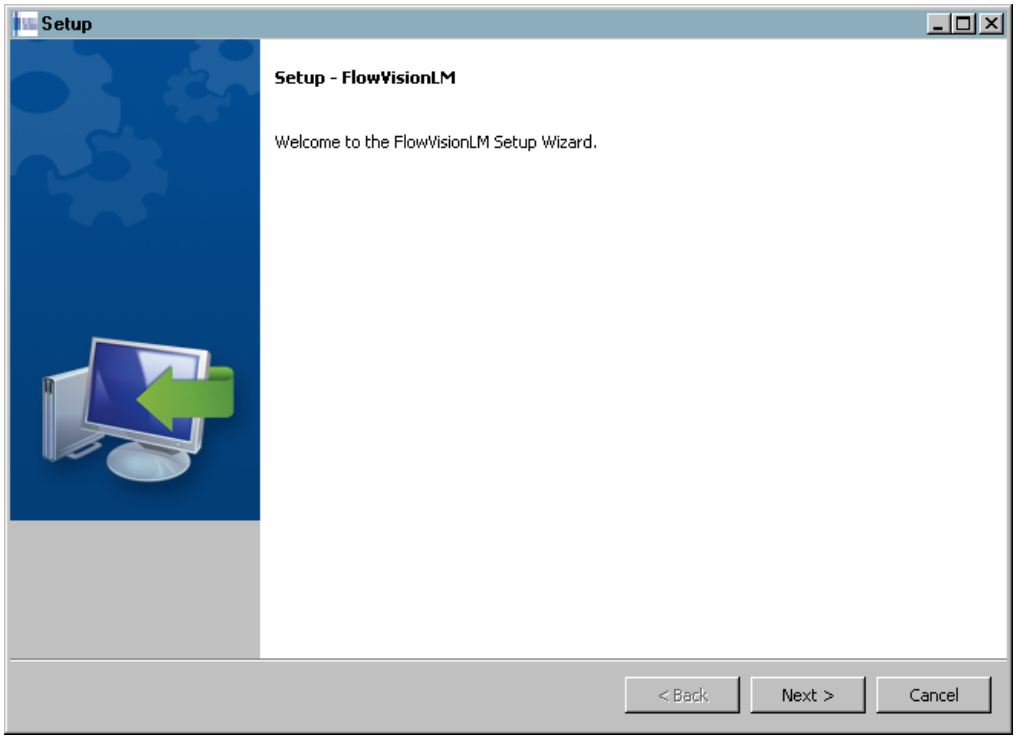
If an installation of **License Manager** or other *FlowVision* modules has been damaged or removed, they can be restored by an appropriate installer (**License Manager**'s installer or the main installer).

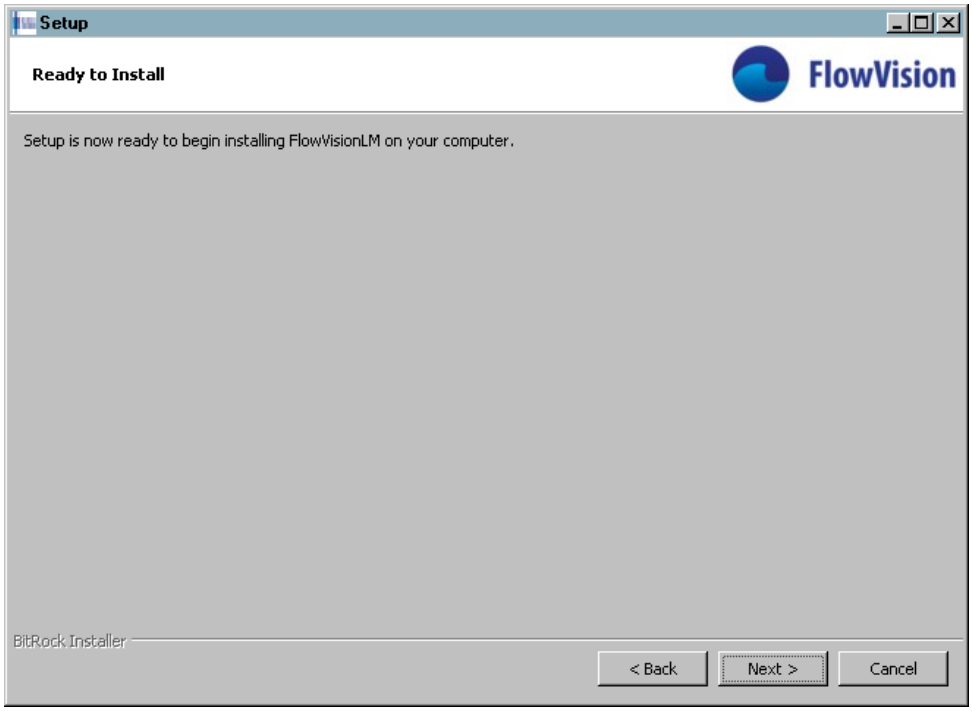
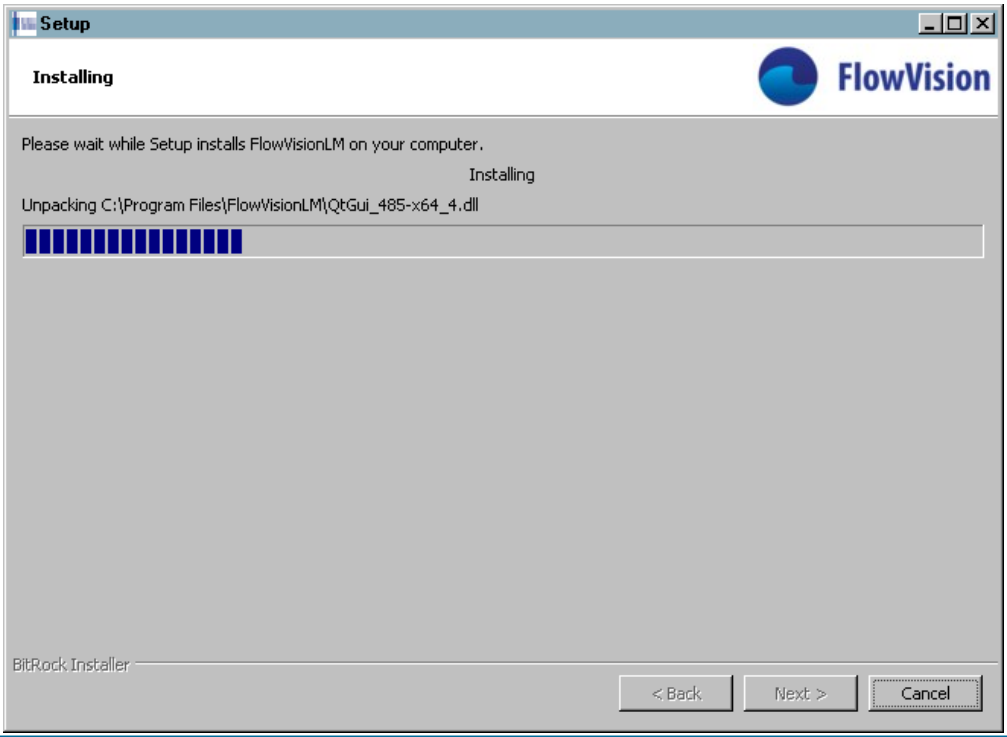
Also, using the *FlowVision*'s main installer it is possible to change the list of the installed *FlowVision* modules.

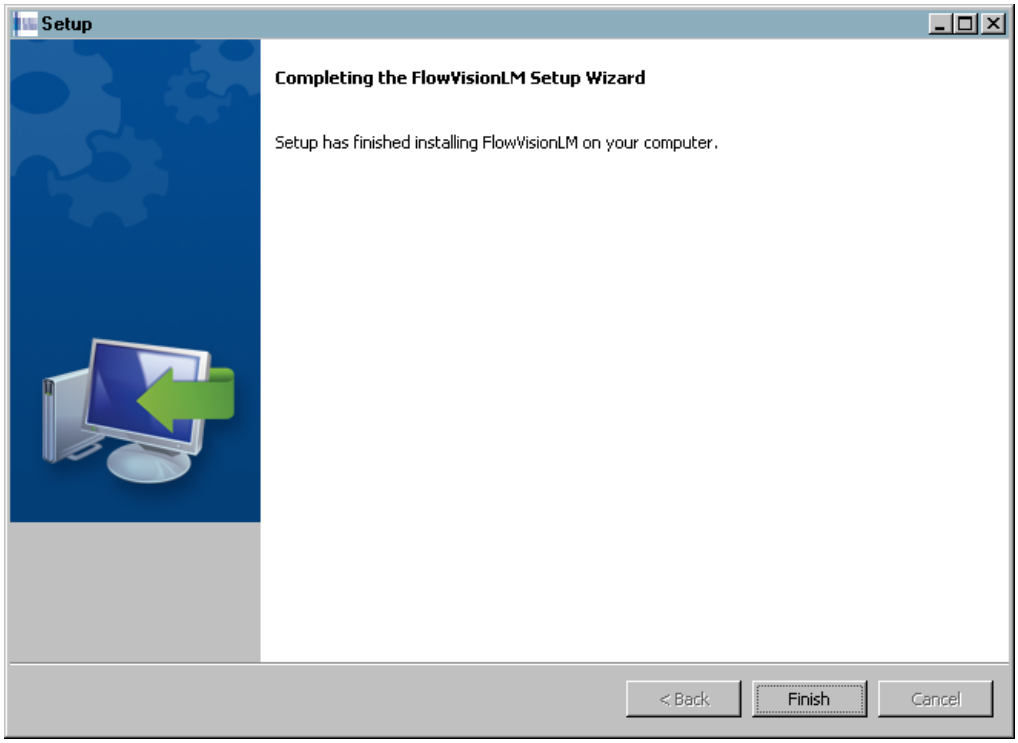
In this case, the installer of **License Manager**, after its welcome dialog box, immediately will go to the installation, and the main installer prompts you to specify the list of modules to be installed.

See step-by-step procedures below (for *Windows*; installers for *Linux* work similarly).

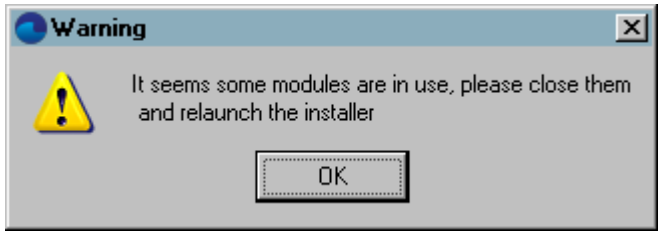
Recovery of damaged files of License Manager

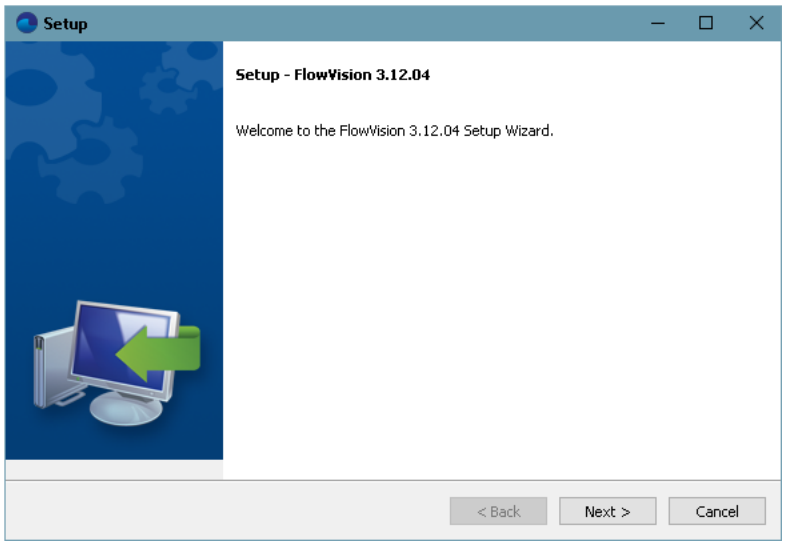
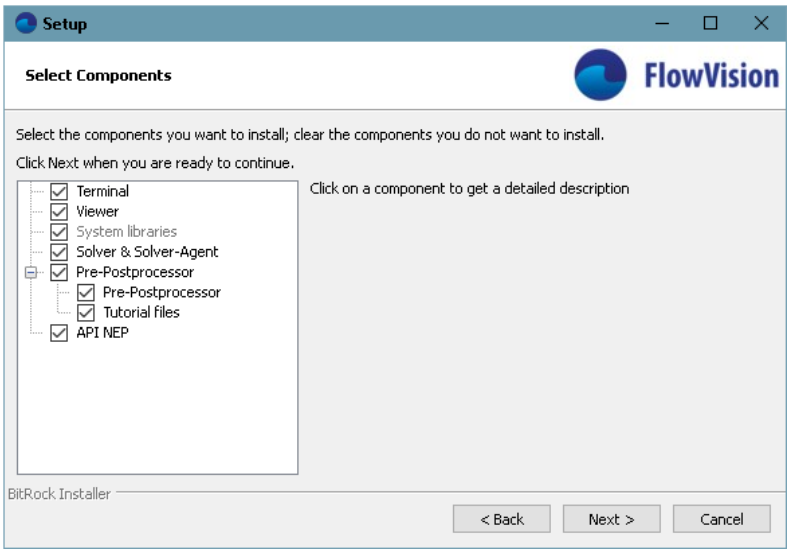
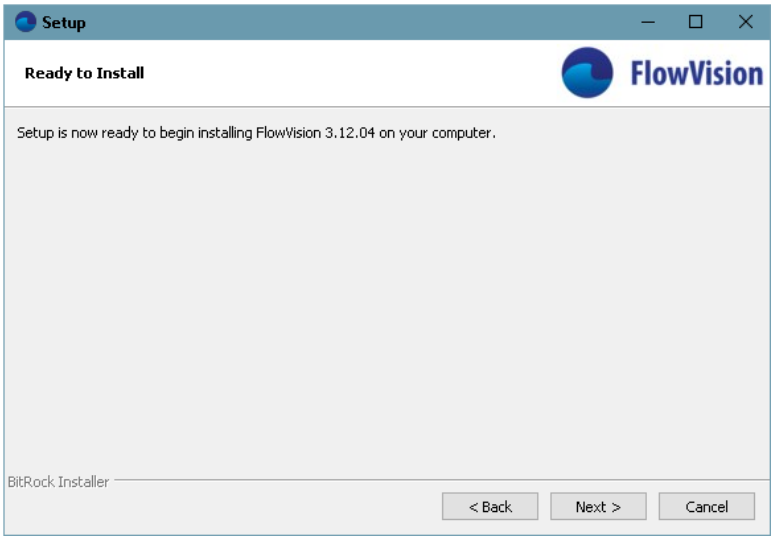
Step	Actions
1	Stop all running *FlowVision* modules. If any component of previous or current version of *FlowVision* is working when installation is started, and this prevents the recovery of the installation, a message ("**It seems some modules are in use, please close them and relaunch the installer**") will open:  If so, click **OK**, then close the working component(s) of *FlowVision* and proceed to the next step.
1	Run the executable file of the **License Manager**'s installer (`FlowVisionLM-windows-installer.exe`). This welcome dialog box will open:  Click **Next** to continue.
2	A dialog box opens with a message about readiness to install:

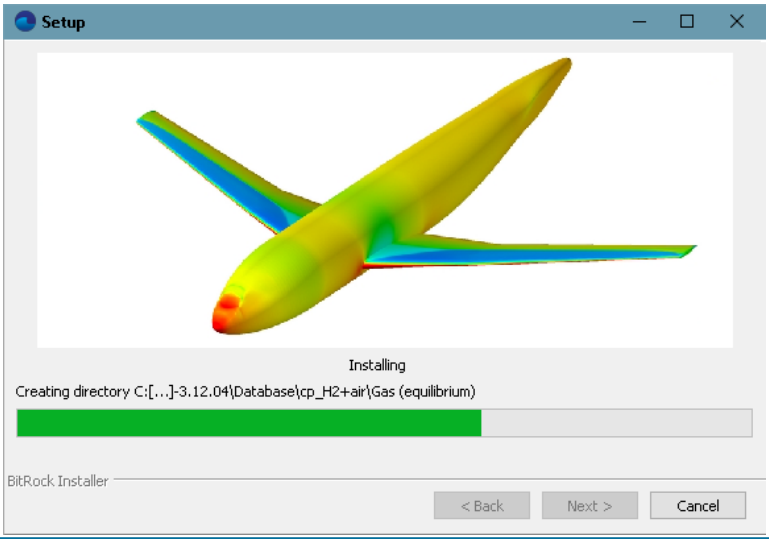
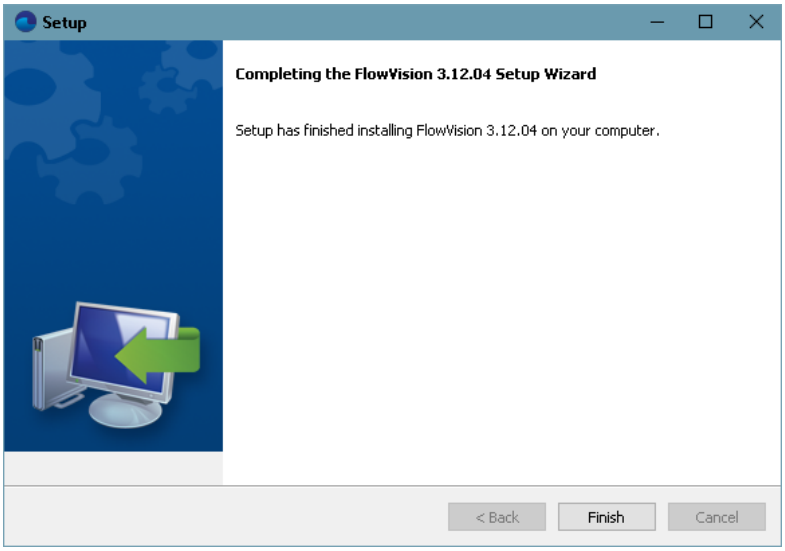
Step	Actions
	 Click **Next** to continue the installation.
3	A dialog box opens displaying the process of file recovery: 
4	A dialog box opens upon completion of the file recovery:

Step	Actions
	 Click **Finish**.

Repair corrupted modules FlowVision (except License Manager)

Step	Actions
1	Stop all running *FlowVision* modules. If any component of previous or current version of *FlowVision* is working when installation is started, and this prevents the recovery of the installation, a message ("**It seems some modules are in use, please close them and relaunch the installer**") will open:  If so, click **OK**, then close the working component(s) of *FlowVision* and proceed to the next step.
1	Run the executable file of the main *FlowVision*'s installer (`FlowVision-3.13.01-windows-installer.exe`, the digits in the file name correspond ti the version number). This welcome dialog box will open:

Step	Actions
	 Click **Next**.
2	Select *FlowVision* modules, which are to be installed:  Click **Next** to continue.
3	A dialog box opens with a message about readiness to install *FlowVision* modules:  Click **Next** to continue.
4	A dialog box with changing illustrations of the *FlowVision* opens displaying the installation process:

Step	Actions
	
13	A dialog box opens upon completion of the file recovery:  Click **Finish**.

4.8 Setting up

See sections:

4.8.1 Initial Configuration

After installing *FlowVision* it is necessary to:

- check the base correctness of *FlowVision* modules installation
- set up the FlowVision modules

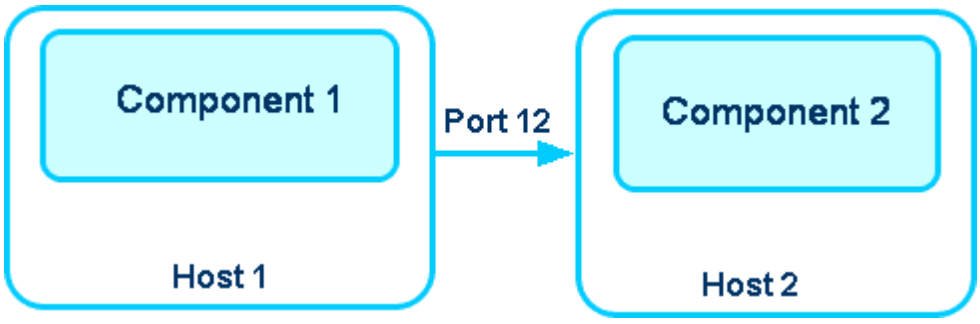
Check the base correctness of FlowVision modules installation

To check functionality of *FlowVision* modules:

- Launch **Configurator** and, in the **General** tab, click the **Get versions** button.
- If there is an error message for any module instead of version information, create a diagnostic information file and send it to *FlowVision* technical support with a description of the problem encountered.

Setting up modules

The setting process includes input of parameters required for module interaction, as well as of log file writing parameters.



During operation, FlowVision modules send requests to each other. The requests are sent using a dedicated channel. Different modules may be run on different computers while doing so (with some exceptions). If direct network communication is not possible, communication is maintained through the **Retranslator**.

To provide the operation of a *FlowVision* module, set:

- IP addresses of the computers where the modules it interacts with are installed
- numbers of ports through which the interaction is carried out
- necessary additional settings

Some modules require setting only port numbers and do not require setting IP addresses.

Initial setting up in Windows

In *Windows*, initial settings are set during the installation process. By default, every *FlowVision* module is set up to interact with modules installed on the local computer. If **Solver Agent** or **License Manager** is not installed on the same computer, the IP addresses or network names of computers where they are installed must be set during installation. If necessary, the settings entered during the installation process can be changed.

In *Windows*, the default settings must be changed after installation if:

- the default ports on the computer are closed or busy
- the default ports do not match the ports used for module communication on other computers
- the default commands must be changed

Initial setting up in Linux

In *Linux*, all modules are set up to interact with modules installed on the local computer.

The default settings must be modified after installation on *Linux* if:

- **Solver Agent** or **License Manager** are not installed on the local computer
- the default ports on the local computer are closed or busy
- the default ports on the local computer do not match the ports used for module communication on other computers
- the default commands must be changed

After the installation, setting up the *FlowVision* modules can be done using the *configuration files*.

Configuration files

A *configuration file* is a text file with extension **`cfg`**. Each configuration file corresponds to one *FlowVision* module.

A configuration file contains parameters for connection and interaction of the given module with the others and log file recording.

The configuration files are edited:

- manually, in the user directory
- in the **Configurator** module

FlowVision has the following configuration files:

File	Program module
`FvLicense.cfg`	**License Manager**
`FvSolverAgent.cfg`	**Solver Agent**
`FvSolver.cfg`	**Solver**
`FvTerminal.cfg`	**Terminal**
`FvPPP.cfg` To edit settings of **Pre-Postprocessor**, apply its user interface, do *not* edit the file **`FvPPP.cfg`** manually using a text editor. The file **`FvPPP.cfg`** is coded in the XML format. Manual editing this file can destroy its structure and cause incorrect operation of **Pre-Postprocessor**.	**Pre-Postprocessor**
`FvViewer.cfg`	**Viewer**
`MpmAgent.cfg`	**MPM-Agent**
`FvConnect.cfg`	**Retranslator**

Log file

The *log file* is a text file with **`log`** extension. Every log file corresponds to a module of *FlowVision*. Information about the operation of a module and errors encountered during operation is stored in the log file.

Fixing problems

In case of problems during *FlowVision* module configuration and starting:

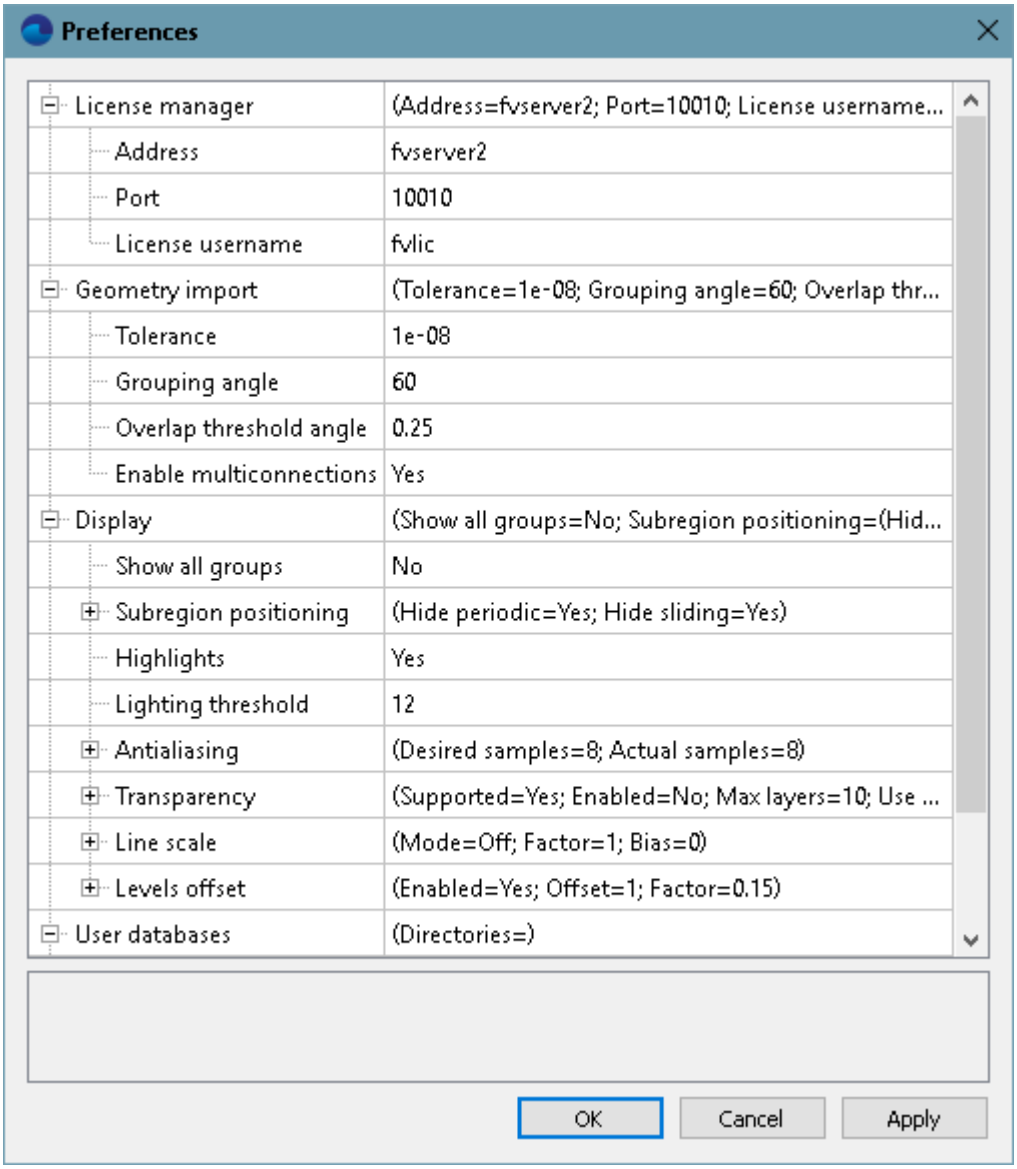
- Read the description of typical configuration errors and try to fix them yourself.

- If the problems are not fixed, create a diagnostic information file using the **Configurator** and send the file to *FlowVision* technical support with a description of the problems encountered.

4.8.2 Basic settings of Pre-Postprocessor

Basic settings of *FlowVision* are set in **Pre-Postprocessor**, in the dialog box **Preferences**, which opens by the command **File > Preferences** from the main menu.

The specified settings take effect after you click **Apply** or **OK**.

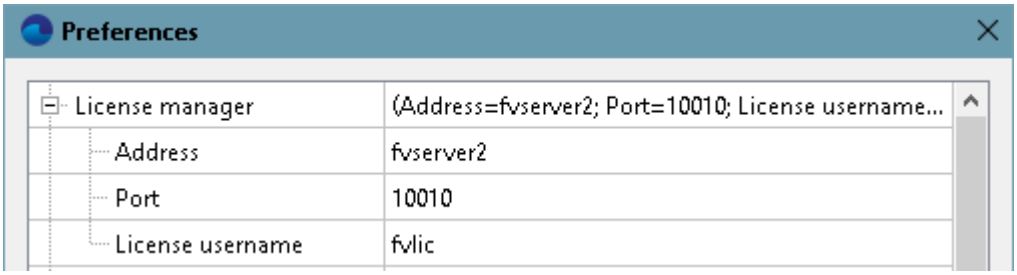


The **Preferences** dialog box

To close the dialog box, click **OK** (your the changes will be saved) or **Cancel** (without saving the changes).

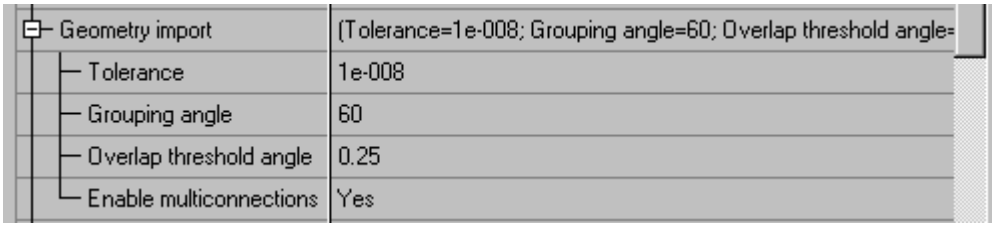
The dialog box displays options organized into several groups of settings (folders). These settings are described in subsections below.

Group of settings "License manager"



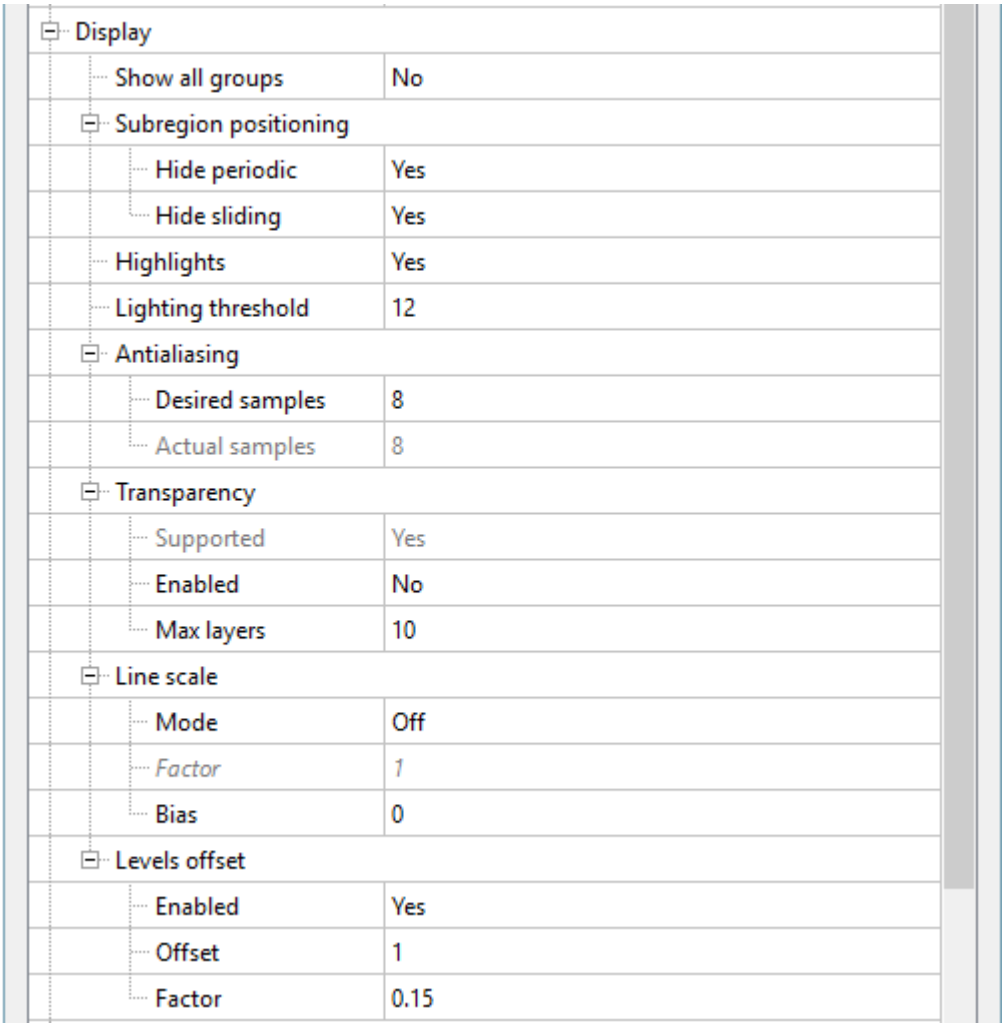
Setting	Description
Address	IP address or network name of the computer, on which **License Manager** is installed (See section "Initial Configuration").
Port	Port of **License Manager** (see section Initial Configuration).
License username	Name, on which the license has been issued (see section Initial Configuration).

Group of settings "Geometry import"



Setting	Description
Tolerance	The accuracy with which the geometric parameters are determined (coordinates of points of project's geometric objects).
Grouping angle	This is a criterion for merging facets into groups while their grouping: adjacent facets belong to the same group when angles between their normals are smaller then the **Grouping angle**. The smaller is the **Grouping angle**, the larger will be the number of groups.
Overlap threshold angle	The threshold (minimal) angle between adjacent facets, from which such a pair is considered as an overlap, [degree].
Enable multiconnection	Determines whether import of geometries with multiconnection is allowed. *Multiconnection* is connection of three or more surfaces by at least one edge; a special case of the multiconnection is a T-connection of surfaces (see section Multiconnection). When multiconnection is used, a **Model** must be specified in the project in all **Subregions** that are adjacent to any surface forming the multiconnection. If this requirement is violated, then all **Subregions**, which are adjacent to the multiconnection, will be marked in the project tree with a "**!**" symbol and the project will not be able to run for calculation. You *can not* create a geometry with multiconnection using an *assembly* (assembly is loading a geometry model from several files). If you attempt to do this, the program will make an error message indicating on the first outer geometry.

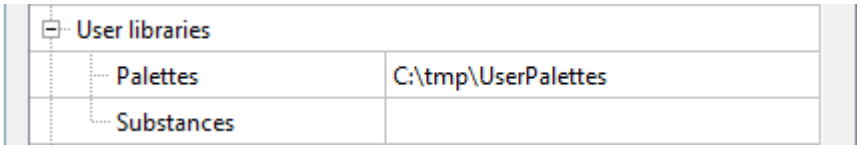
Group of settings "Display"



Setting	Description
Show all groups	Possible options are: • **Yes** – display all geometric groups of facets in the folder **Geometry** on the project tree, in the **Preprocessor** tab (the list of all groups is displayed in the element **Subregions > SubRegion #N > Geometry > Region - Surface #N**) • **No** – the groups are not displayed
Subregion positioning > Hide periodic **Subregion positioning > Hide sliding**	Displaying or hiding periodic and/or sliding boundary conditions in the image, formed when you click the button (**Enable/disable duplication and overlapping of subdomains up to the complete model supplied with sector-sliding**) in the **Rendering** toolbar. See illustrations in the section Sector-sliding setting.
Highlights	Possible options are: • **Yes** – display glares on surfaces • **No** – do not display glares This setting is also toggled by the button (**Turn on/off glare on the surface**) in the **Rendering Toolbar**.
Lighting threshold	Maximum lighting level for the geometry model of the computational domain (can be set from 0 to 180)
Antialiasing > Desired samples	The desired number of points to smooth lines
Antialiasing > Actual samples	The actual number of points to smooth lines. It is determined automatically by abilities of your the video card.

Setting	Description
Transparency > Supported	The ability to support transparency. Is set automatically in accordance with your video card. Possible options are: **Yes** | **No**.
Transparency > Enabled	Possible options are: • **Yes** – surfaces can be semi-transparent • **No** – surfaces are either completely transparent (and so invisible) or completely opaque This setting is also toggled by the button (**Enable/disable transparency support**) in the **Rendering** Toolbar.
Transparency > Max layers	The number of layers that are visible through a semi-transparent surface
Line scale > Mode	Scaling of thickness of lines depending on size of the **View** window. Possible options are: • **Off** – scaling of lines is disabled • **Automatic** – scaling of lines is automatic • **Constant** – line width is multiplied by a constant
Line scale > Factor	The constant factor, on which the thickness of lines is multiplied by, when **Mode = Constant**
Line scale > Bias	Permanent supplement to the thickness of lines (from 0 to 2)
Levels offset > ...	Parameters that control the displacement of the visualization **Layers**, which were built on the same surface.
Levels offset > Enabled	Possible options are: • **Yes** – layers, which were built on the same surface, are displayed with offset • **No** – layers are displayed without offset
Levels offset > Offset	A constant distance between adjacent layers (it is defined in conditional units from 0 to 100)
Levels offset > Factor	A factor for additional offset of layers depending on the angle of view

Group of settings "User libraries"



Setting	Description
Palettes	Paths to a directory or several directories that contain files of user **Palettes**. If you specify several directories, separate them with a semicolon symbol (";"). The program automatically adds a directory in this list when you, in the **Properties** window, apply the command **Palette > Operations > (Save palette to file)** and answer positively to the appropriate request ("**Do you want to add this folder in palette library?**"), see section Parameters for defining a palette.
Substances	The path to the user's **Substances Database**. Specify one or more directories from which user's databases are opened. If you specify several directories, separate them with a semicolon symbol (";"). (See section Substance Database Editor)

Group of settings "Formula editor"

The group of settings **Formula editor** contains parameters of fonts to display various elements of formulae in the dialog box **Formula editor**.

a)

Formula editor	(Syntax highlighting=(Enabled=Yes; Constants=(Color=Green; Bol
Syntax highlighting	(Enabled=Yes; Constants=(Color=Green; Bold=No; Italic=No; Und
Enabled	Yes
Constants	(Color=Green; Bold=No; Italic=No; Underline=No; Strikeout=No)
Variables	(Color=Blue; Bold=Yes; Italic=No; Underline=No; Strikeout=No)
Functions	(Color=Black; Bold=Yes; Italic=No; Underline=No; Strikeout=No)
Externals	(Color=Olive; Bold=Yes; Italic=No; Underline=No; Strikeout=No)
Operators	(Color=Black; Bold=Yes; Italic=No; Underline=No; Strikeout=No)
Local expressions	(Color=Purple; Bold=Yes; Italic=No; Underline=No; Strikeout=No)
Comments	(Color=Silver; Bold=No; Italic=Yes; Underline=No; Strikeout=No)
Brackets	(Color=Black; Bold=No; Italic=No; Underline=No; Strikeout=No)
Separators	(Color=Black; Bold=No; Italic=No; Underline=No; Strikeout=No)
Errors	(Color=Red; Bold=No; Italic=No; Underline=No; Strikeout=No)

b)

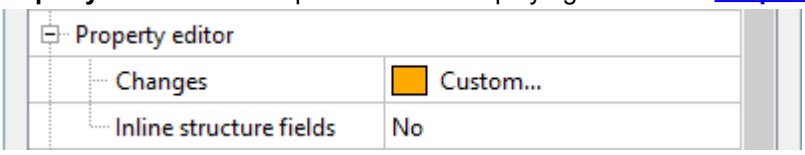
Constants	(Color=Green; Bold=No; Italic=No; Underline=No; Strikeout=No)
Color	Green
Bold	No
Italic	No
Underline	No
Strikeout	No

The **Formula editor** group of settings in the **Preferences** dialog box:
a - subfolders are collapsed, b - an example of an expanded subfolder

Setting	Description	
Syntax highlighting > Enabled	Possible options are: • **Yes** – highlighting of syntax is enabled • **No** – highlighting of syntax is disabled	
Syntax highlighting> Constants **Syntax highlighting> Variables** **Syntax highlighting> Functions** **Syntax highlighting> Externals** **Syntax highlighting> Operators** **Syntax highlighting> Local expressions** **Syntax highlighting> Comments** **Syntax highlighting> Brackets** **Syntax highlighting> Separators** **Syntax highlighting> Errors**	Parameters for displaying various syntax elements in the **Formula editor**.	
	... > Color	Font color
	... > Bold	Possible options are: **Yes** – the font is bold, **No** – the font is not bold.
	... > Italic	Possible options are: **Yes** – the font is italic, **No** – the font is roman
	... > Underline	Possible options are: **Yes** – the font is underlined, **No** – the font is not underlined
	... > Strikeout	Possible options are: **Yes** – the font is strikeout, **No** – the font is not strikeout

Group of settings "Property editor"

The group of settings **Property editor** contains parameters of displaying data in the **Properties** window.



Setting	Description
Changes	The color, which will be used to highlight the fields in the **Properties** window that have been changed but not saved yet. By default it is orange color. See details in the subsection *"Highlighting the changes"*.
Inline structure fields	This setting determines if lines in the "**Property=Value; ...**" format will be displayed in fields that corresponds to groups of properties in the **Properties** window. Possible options: **Yes** | **No**. See details in the subsection *"Displaying structure of groups of properties"*.

4.8.3 User directories

User directory is a directory, which stores configuration files, log files and other files with settings.

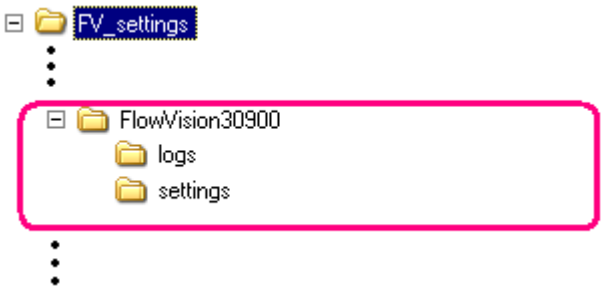
There are two user directories:

- for **License Manager**
- for other *FlowVision* modules.

The location of each of them is determined by answers to the questions of **License Manager**'s installer and the main installer, when were given at the Installation of FlowVision. These directories can be the same.

In the user directory of **License Manager** the subdirectory **`FlowVisionLM`** is created.

When installing an each new version of *FlowVision* (since version *3.09*) a subdirectory corresponding to the version being installed appears in the user directory, with a name matching to the version number (**`FlowVision3xxxx`**).



For each version of *FlowVision* in the user directory an appropriate **`FlowVision3xxxx`** subdirectory is created.

User directory of the **License Manager** can be same or not same as the user directory of other *FlowVision* modules.
In the user directory of the **License Manager** a subdirectory `FlowVisionLM` is created.

The directory `FlowVisionLM` contains the following subdirectories and files:

Name	Description
Subdirectories	
`licHistory`	Contains files with names like `mmyyyy.hst` (where `mmyyyy` means a month and a year), containing the license history. The history of use the licenses is available for viewing through **License monitor** when you click there on the **History** button.
`logs`	The directory, which stores log files
`settings`	The directory, which stores configuration files
Files	
`FvLicense.dat`	Information about licenses

Directories `FlowVision3xxxx` contain the following subdirectories and files:

Name	Description
Subdirectories	
`logs`	The directory, which stores log files
`settings`	The directory, which stores configuration files
Files	
`FvUsers.dat`	The file, which stores the user's credentials for **Solver-Agent** (see section Registration data (profile) of Solver-Agent's user and their change)
`FvTerminal.hst`	Service data (history of IP address, port, name, that were entered into dialog boxes of **Terminal** and **Viewer** to facilitate their re-entry)
`FvViewer.hst`	
`FvTerminal.lhs`	
`FvTerminal.lvs`	
`MpManagerD.bgg`	

The **User directory** must be writable by the user as which the module runs. If the **User directory** is not writable or an invalid path has been specified, then the model will not be able to run.

To edit user directories it is often necessary to run **Configurator** or text editor as Administrator (even when the user has been obtained administrative access rights). Settings of the user directories are stored in the file `Fv.cfd`, which locates in the *FlowVision*'s installation directory. So, to edit the settings it is necessary to have access rights for writing into the file `Fv.cfd`. The `Fv.cfd` file often locates in the directory `Program Files` or another directory with restricted access, which requires more powerful access rights when you run **Configurator**.

Paths to *FlowVision*'s user directories are specified in files `Fv.cfd`. Here is an example of the `Fv.cfd` file:

```
HOMESRV=C:\FV_settings
HOMECLN=HOMESRV
```

Viewing and changing of the contents of the `Fv.cfd` files can be done using a text editor, or (for *FlowVision* modules except **License Manager**) using **Configurator**, in the tab **Configuration/Logs** after clicking on the **User directories** button.

When you start a *FlowVision* component, it carries out the following actions:

a. It determines the position of the **User directory** (by parameters **HOMECLN** and **HOMESRV**).
b. It looks for its configuration file (`cfg`) in the **User directory**.
c. If an appropriate configuration file is absent in the **User directory**, it is created with standard parameters.

If *FlowVision* is installed first time on this computer, then the configuration files have not been created yet. At the first start of each component its configuration files are created in appropriate directories.

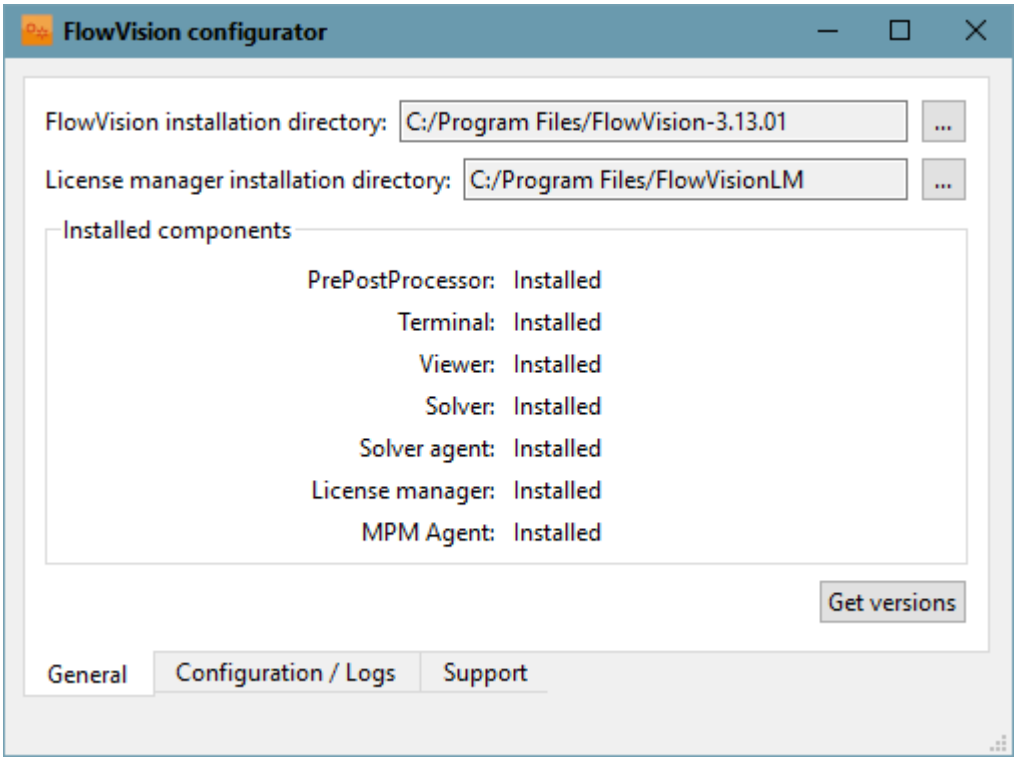
4.8.4 Settings defined in Configurator

Configurator (the **FvConfigurator** module) is designed to:

- provide general information about components of *FlowVision*, which are installed on this computer
- set user directories
- view and edit configuration files
- view log files using a graphical interface
- automatically generate archive for technical support

Search of all files is done automatically.

FvConfigurator module runs either under *Windows* or under *Linux*, it is set automatically, regardless of set of the installed components. It is located in the root installation directory of *FlowVision*.

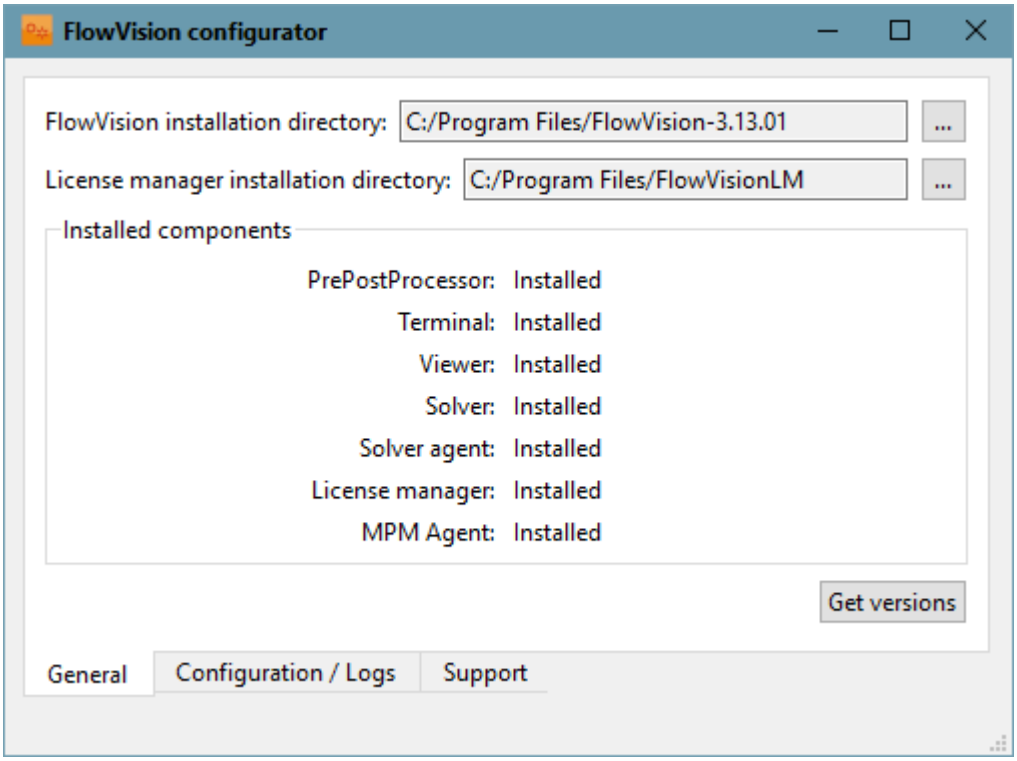


Configurator has the following tabs, described in separate sections:

- **General**
- **Configuration/Logs**
- **Support**

4.8.4.1 Configurator's tab "General"

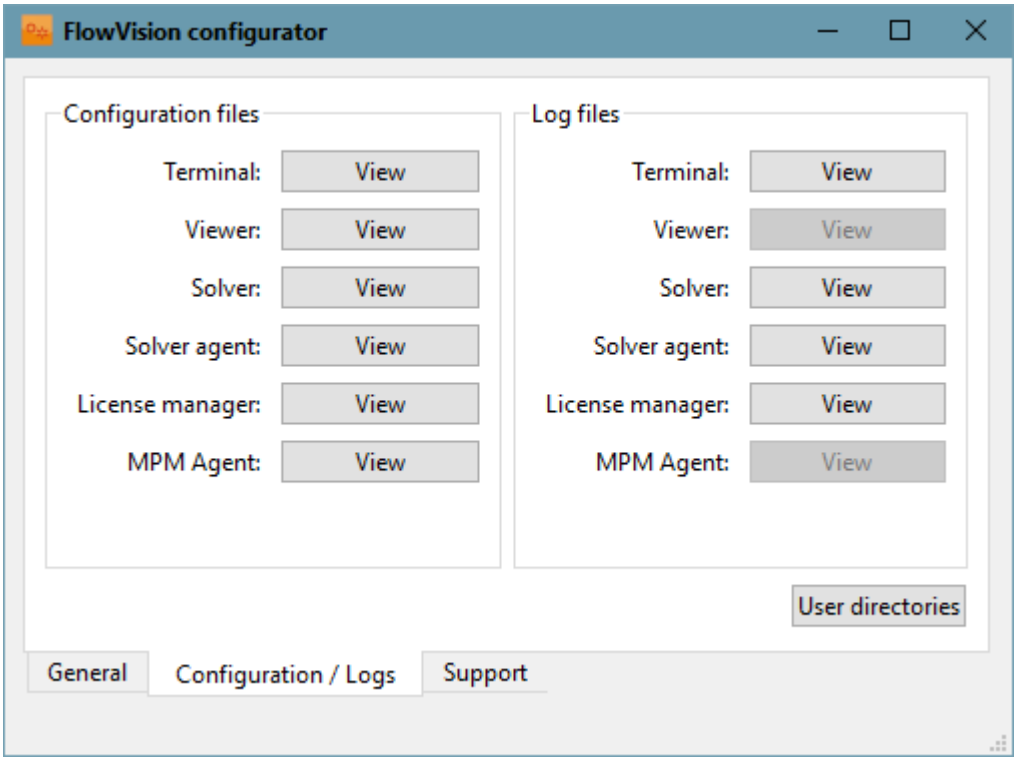
The **General** tab contains information about installed components.



Contents of the **General** tab	
FlowVision installation directory	The directory, in which *FlowVision* is installed (modules except **License Manager**)
License manager installation directory	The directory, in which **License Manager** is installed
Installed components	List of FlowVision modules with indication of whether they are installed on this computer
Button **Get versions**	When you click this button, the number of version for each installed component is displayed.

4.8.4.2 Configurator's tab "Configuration/Logs"

The **Configuration/Logs** tab is used for viewing and editing configuration (`cfg`) files and for viewing log (`log`) files.



Viewing or changing settings in configuration files, viewing log files

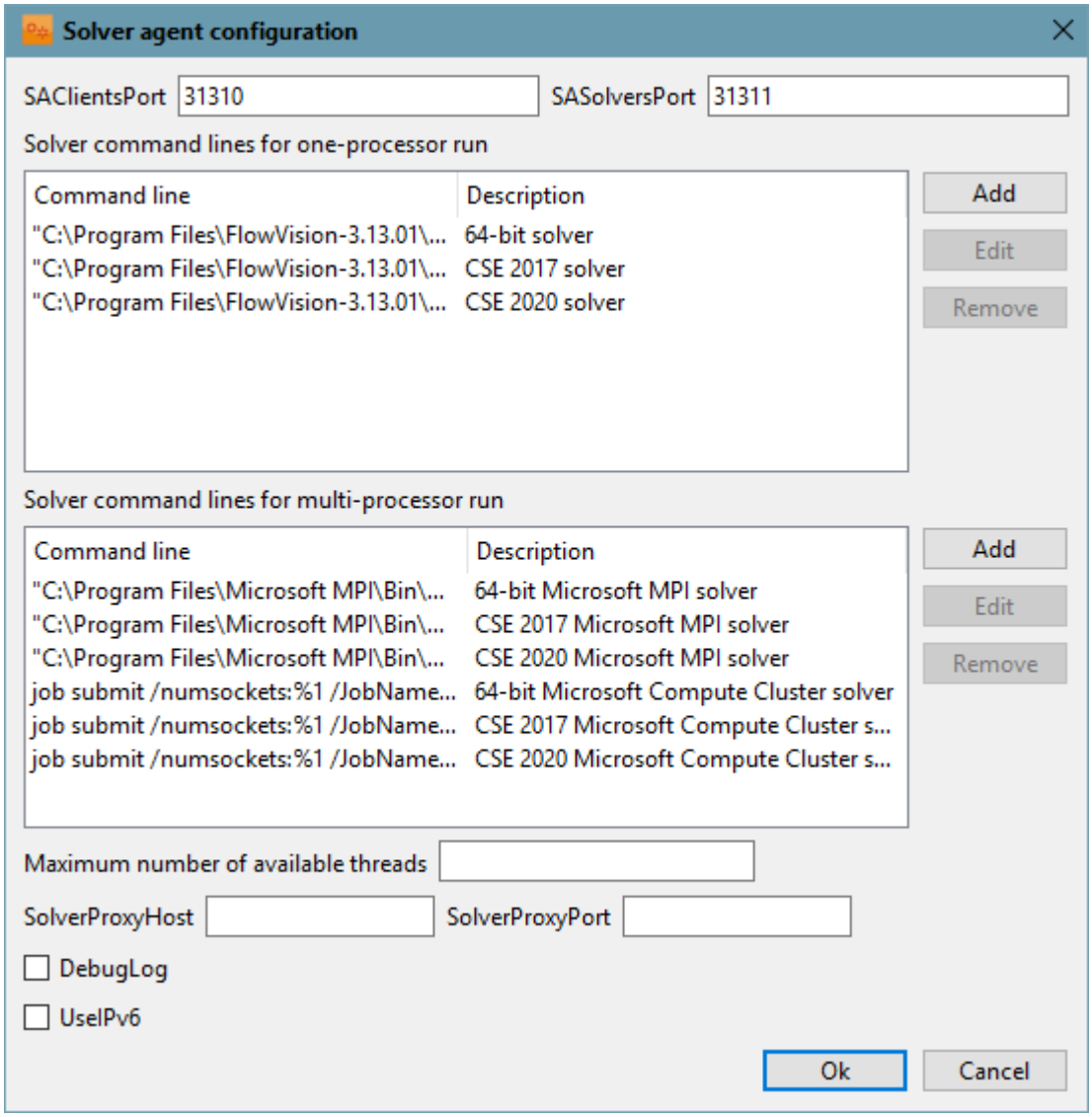
View buttons in tab **Configuration/Logs** are used for:

Configuration files		
	Terminal	View or change settings stored in the Configuration file of Terminal[1)]
	Viewer	View or change settings stored in the Configuration file of Viewer[1)]
	Solver	View or change settings stored in the Configuration file of Solver[1)]
	Solver agent	Open the **Solver agent configuration** window, where you can view or change settings stored in the Configuration file of Solver-Agent[1.2)]
	License Manager	View or change settings stored in the Configuration file of the License manager[1)]
	MPM Agent	View or change settings stored in the Configuration file of MPM-Agent (MpmAgent.cfg)[1)]. The **MPM Agent configuration** dialog box will open.
Log files		
	Terminal	View log file of **Terminal**[1)]
	Viewer	View log file of **Viewer**[1)]
	Solver	View log file of **Solver**[1)]
	Solver agent	View log file of **Solver-Agent**[1)]
	License Manager	View log file of **License Manager**[1)]
	MPM Agent	View log file of **MPM-Agent**[1)]

Notes:

[1)] To view the settings or contents of the file, click the **View** button. To see the path to the file, move the mouse pointer over the **View** button, but do not click. A **View** button in the **Log files** pane is only enabled if the corresponding file exists.

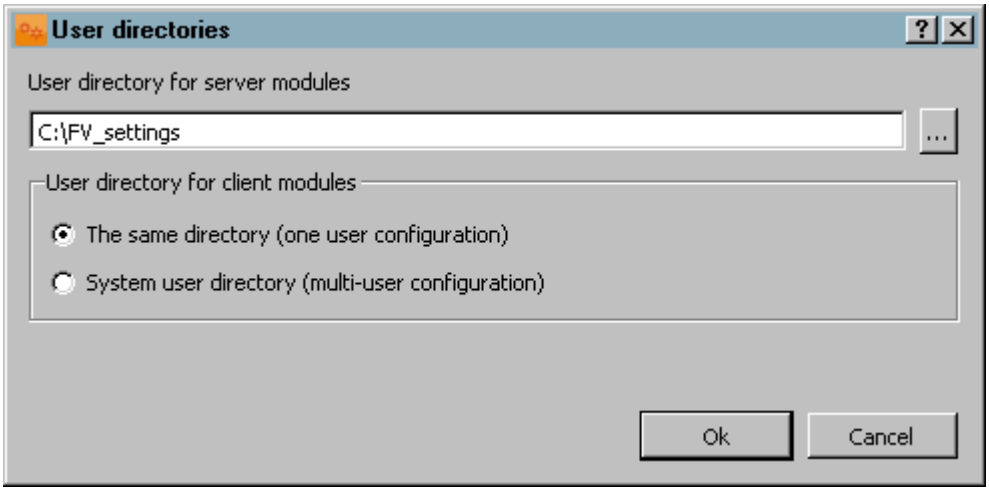
[2)] In the **Solver agent configuration** dialog box you can configure command lines to run **Solver**:



Button	Description
Add	Add a new command line to run **Solver**. Note that the command lines to run **Solver** in multiprocessor mode always contain symbols `%1` (these symbols will be replaced by the number of processors). You can not enter a command line, which does not contain symbols `%1` (the **OK** button would be gray (inactive) in an appropriate dialog box).
Edit	Changing the selected command line (command line for **Solver** working in multiprocessor mode should contain symbols `%1`).
Remove	Deleting the selected command line. The line will only be deleted after your positive answer to the program's prompt for confirmation this action.

Specifying paths to user directories

Button **User directories** opens the **User directories** window in which the path to user directories for client and server modules of *FlowVision* is specified:



The contents of the **User directories** window:

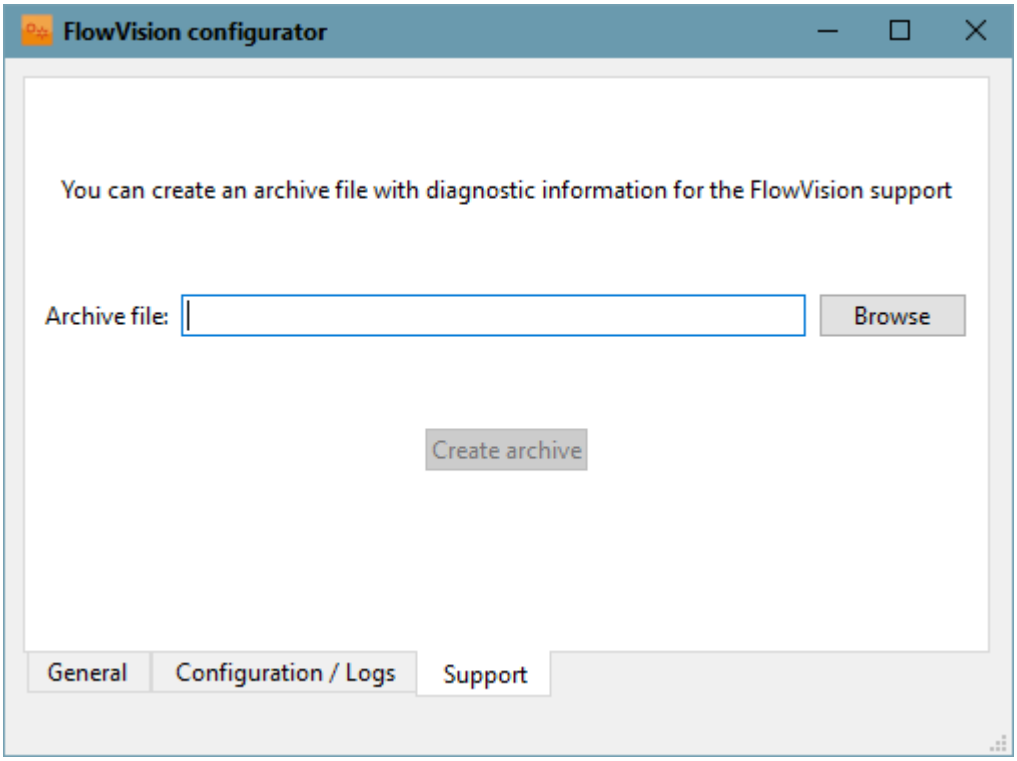
User directory for the server modules	This setting defines the path to user directory of server modules.
User directory for client modules	This setting defines the path to user directory of client modules.
The same directory (one user configuration)	Use as the configuration directory for client applications the configuration directory for server modules.
System user directory (multi-user configuration)	Use as the configuration directory for client applications the system user directory.

The **User directory** must be writable by the user as which the module runs. If the **User directory** is not writable or an invalid path has been specified, then the model will not be able to run.

To edit user directories it is often necessary to run **Configurator** or text editor as Administrator (even when the user has been obtained administrative access rights). Settings of the user directories are stored in the file `Fv.cfd`, which locates in the *FlowVision*'s installation directory. So, to edit the settings it is necessary to have access rights for writing into the file `Fv.cfd`. The `Fv.cfd` file often locates in the directory `Program Files` or another directory with restricted access, which requires more powerful access rights when you run **Configurator**.

4.8.4.3 Configurator's tab "Support"

The tab **Support** is designed to create an archive file with diagnostic information for the technical support services of *FlowVision*.



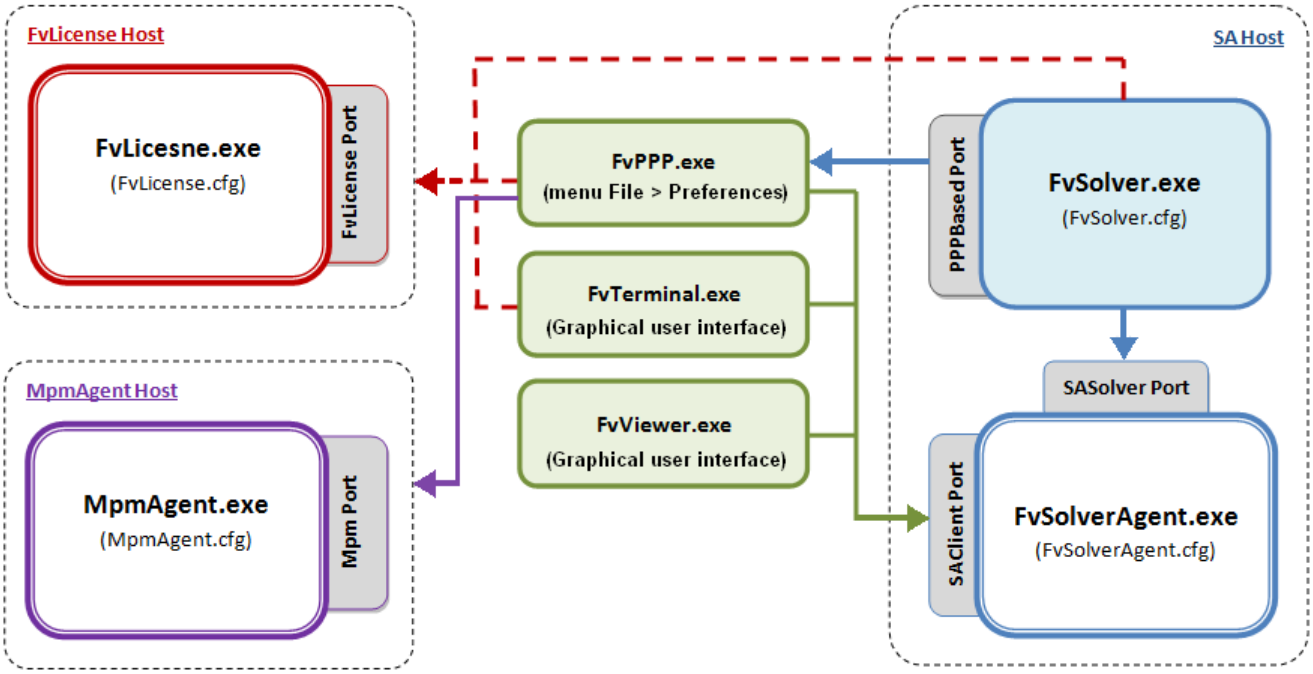
To create a file with diagnostic information, follow the steps:

- Click the **Browse** button.
- In the field **Archive file** specify the path (directory) and the name of the archive file.
- Click the **Create archive** button.

You can also generate the diagnostic information using the **`FvDiagUtil.exe`** utility, see section Generating the diagnostics information.

4.8.5 Parameters in configuration files

Parameters, which are stored in configuration files, can be edited in the GUI of **Configurator** and directly in the text files.



Format of configuration files

Configuration files are text files consisting of lines of the form:

```
Parameter=Value [;Comment]
```

All configuration files of *FlowVision* modules must be saved in the *UTF-8* format. Use of another format can cause malfunction of the modules.

General parameters in the configuration files

Parameter	Description	The default value
PPPBasePort	Base port for direct connection between**Solver**and**Pre-Postprocessor**[1)]	**11000**
FvLicenseHost	IP address or network name of the computer, on which **License Manager** is running[2)]	**127.0.0.1**
FvLicensePort	Port for connection **License Manager** and licensed components[3)]	**10010**
SAHost	IP address or network name of the computer, on which **Solver-Agent** is running[2)]	**127.0.0.1**
SAClientsPort	Port for connection **Solver-Agent** and client components (**Pre-Postprocessor**, **Terminal**, **Viewer**)[3)]	abce**0** *)
SASolversPort	Port for connection **Solver** and **Solver-Agent**.	abce**1** *)
DebugLog	Recording debug messages into log files: • **Yes** - the recording is enabled • other value or absence of this parameter means that the recording is disabled	**No**

*) Digits **abce** in the port number are the 1st, 2nd, 3rd and 5th digits of the version number. For example, for version *30901* ports **SAClientsPort** and **SASolversPort** will be **30910** and **30911** by default.

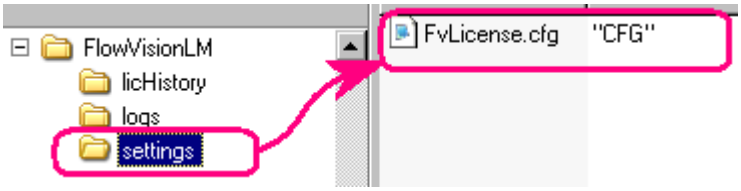
[1] If the base port for connection **Pre-Postprocessor** and**Solver** is already occupied, then the connection is done using a port with a successive number. If direct connection between **Solver** and **Postprocessor** is not possible, you have to configure a connection through **Retranslator** and **Solver-Agent**.
[2] If this *FlowVision* component (**License Manager** or **Solver-Agent**) is not available for communication directly, then it is necessary to specify IP address or host name of the computer, on which **Retranslator** runs.
[3] If this *FlowVision* component (**License Manager** or **Solver-Agent**) is not available for communication directly, then it is necessary to specify a port for connection to **Retranslator**.

Settings of the firewall

Work of *FlowVsion* requires that the following ports be open:

- **PPPBasePort** and 10 ports after it
- **FvLicensePort**
- **SAClientsPort**
- **SASolversPort**

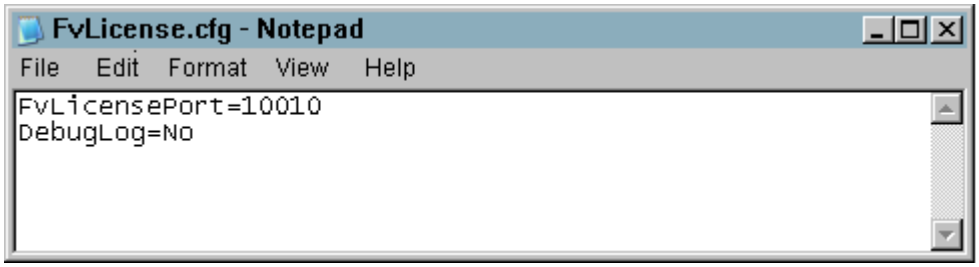
4.8.5.1 Configuration file of the License Manager (FvLicense.cfg)



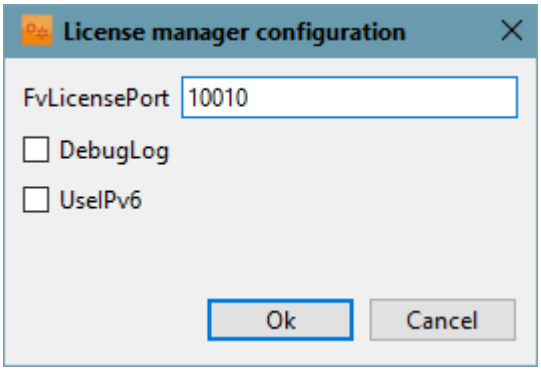
Configuration file of **License Manager** (`FvLicense.cfg`) is located in the subdirectory `settings` in the **License Manager**'s user directory and contains the following settings:

Parameter	Description	Default value
FvLicensePort	Port for connection **License Manager** and licensed components. (If **License Manager** is not available for direct connection, then IP address or host name of the computer, on which **Retranslator** runs, is specified.	**10010**
DebugLog	Recording the debug messages into log files: • **Yes** means that the logging is on • any other value or absence of the parameter means that the logging is off	**No**
UseIPv6	Operation using the *TCP/IPv6* protocol. Possible options are: **Yes | No**. For operation in *IPv6* networks you have to specify **UseIPv6=Yes** and addresses in settings of other modules have to be specified using the *IPv6* format or as domain names.	**No**
StatSave	Gathering statistics of license use	**Yes**

You can view or change these settings directly in the file (for example, using *Windows*' *Notepad*) or use the **License manager configuration** window, which opens in **Configurator** after clicking on the button **View** near the "**License Manager**" text in the **Configuration/Logs** tab.



Viewing and/or editing the `FvLicense.cfg` file using *Windows*' *Notepad*



Viewing and/or editing the settings stored in the `FvLicense.cfg` file, using **Configurator**

4.8.5.2 Configuration file of Solver-Agent (FvSolverAgent.cfg)

Configuration file `FvSolverAgent.cfg`

Parameter	Description	The default value
SAClientsPort	See Parameters in configuration files	abce0[1)]
SASolversPort	See Parameters in configuration files	abce1[1)]
DebugLog	See Parameters in configuration files	`No`
MaxThreadNum	The limit on the maximal number of threads, which are available for running projects from the FlowVision's queue system for projects. This parameter is only applied when the queue system is tuned manually.	*By default this parameter is absent in the configuration file and the **MaxThreadNum** value is determined automatically based on the information about the processor of the computer, on which **Solver-Agent** is installed.*
NumaThreadNum	The number of numa cores per processor. This parameter is only applied when the queue system is tuned manually.	*By default this parameter is absent in the configuration file and the **NumaThreadNum** value is determined automatically based on the information about the processor of the computer, on which **Solver-Agent** is installed.*

SolverCmdLine[2) 3) 6)]	Parameters for starting Solver in the *one-processor mode* (without use of *MPI*).
	The text from the comment field from symbol `";"` to the end of the line (the reference to this command line) is displayed as an option in the **Solver type** drop-down list in the **Select solver** dialog box. These options are available for selection when the **Procs** value in this dialog box is **1**.
		Windows
		`FvSolver64.exe;64-bit solver`
		`SolverCmdLine="D:\FlowVision-3.13.01\FvSolver64+CSE_2017.cmd";CSE 2017 solver`
		`SolverCmdLine="D:\FlowVision-3.13.01\FvSolver64+CSE_2020.cmd";CSE 2020 solver`
		Linux
		`./FvSolver64; 64-bit Solver`
		`./FvSolver64+CSE; CSE Solver`[4)]

SolverCmdLineMPI[2) 3) 6)]	Parameters for starting Solver in the *multiprocessor mode* (with use of *MPI*).	**Windows**

Parameter	Description	The default value
	The text from the comment field from symbol ";" to the end of the line (the reference to this command line) is displayed as an option in **Solver type** drop-down list in the **Select solver** dialog box. These options are available for selection when the **Procs** value in this dialog box is **≥2**.	`"C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n %1 "C:\Program Files\FlowVision-3.13.01\FvSolver64.exe" mpi=libFvMPI_MSMPI_v7_x64.dll;64-bit Microsoft MPI solver`
		`job submit /numsockets:% 1 /JobName:FlowVisionHPCSolver mpiexec.exe FvSolver64.exe mpi=libFvMPI_MSMPI_v7_x64.dll;64-bit Microsoft Compute Cluster solver`
		`SolverCmdLineMPI="C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n %1 "D:\FlowVision-3.13.01\FvSolver64+CSE_2017.cmd" mpi=libFvMPI_MSMPI_v7_x64.dll;CSE 2017 Microsoft MPI solver`
		`SolverCmdLineMPI="C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n %1 "D:\FlowVision-3.13.01\FvSolver64+CSE_2020.cmd" mpi=libFvMPI_MSMPI_v7_x64.dll;CSE 2020 Microsoft MPI solver`
		`SolverCmdLineMPI=job submit /numsockets:% 1 /JobName:FlowVisionHPCSolver mpiexec.exe "D:\FlowVision-3.13.01\FvSolver64+CSE_2017.cmd" mpi=libFvMPI_MSMPI_v7_x64.dll;CSE 2017 Microsoft Compute Cluster solver`
		`SolverCmdLineMPI=job submit /numsockets:% 1 /JobName:FlowVisionHPCSolver mpiexec.exe "D:\FlowVision-3.13.01\FvSolver64+CSE_2020.cmd" mpi=libFvMPI_MSMPI_v7_x64.dll;CSE 2020 Microsoft Compute Cluster solver`
		Linux
		`./impi/bin64/mpirun -np %1 ./FvSolver64; 64-bit Predefined MPI Solver`
		`./impi/bin64/mpirun -np %1 ./FvSolver64+CSE; CSE Predefined MPI Solver` [4]
		`mpirun -np %1 ./FvSolver64; 64-bit System MPI Solver`
		`mpirun -np %1 ./FvSolver64+CSE; CSE System MPI Solver` [4]
SolverProxyHost[4]	IP address or host name of the computer, on which **Retranslator** is running	*no default value*
SolverProxyPort[4]	Port for communication between a client *FlowVision* module (**Pre-Postprocessor**, **Viewer**) and **Retranslator** (it must be same as the port, which is specified in the configuration file of Retranslator (FvConnect.cfg))	*no default value*

Parameter	Description	The default value
UseIPv6	Operation using the *TCP/IPv6* protocol. Possible options are: **Yes | No**. For operation in *IPv6* networks you have to specify **UseIPv6=Yes** and addresses in settings of other modules have to be specified using the *IPv6* format or as domain names.	**No**

Notes:

[1)] Digits **abce** in the port number are the 1st, 2nd, 3rd and 5th digits of the version number. For example, for version *30901* ports **SAClientsPort** and **SASolversPort** will be **30910** and **30911** by default.

[2)] There can be several lines with parameter **SolverCmdLine** and/or **SolverCmdLineMPI**.

[3)] In the command line **Solver** will replace symbols `%1` by the number of processors.

[4)] These lines are to be used in *Linux* to run a joint computation with *Abaqus* when CSE connector is used. The line runs a script that specifies paths, which are necessary for CSE, into the environment variable `LD_LIBRARY_PATH` and then starts **Solver**. Also use this wrapper script when you tune running **Solver** for FSI computations on a supercomputer.

[5)]

If **Retranslator** is not used, do not specify parameters **SolverProxyHost** and **SolverProxyPort**! Otherwise, the connection between **Solver** and client modules will not be installed

[6)] Command lines with references that include a substring "`CSE`" (examples: "`CSE 2017 solver`" and "`CSE 2020 solver`") can be used only for starting FSI computations with use a CSE-connector with the *Abaqus* version that corresponds to digits in the reference of the command line. Command lines with "`CSE 2017`" can be used with all versions of *Abaqus* below *2020*. Command lines with "`CSE 2020`" can be used with *Abaqus 2020* and newer versions.

Example of the file `FvSolverAgent.cfg`

```
SAClientsPort=31110
SASolversPort=31111
SolverCmdLine=FvSolver64.exe;64-bit solver
SolverCmdLineMPI="C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n %1 "C:\Program
Files\FlowVision-3.13.01\FvSolver64.exe" mpi=libFvMPI_MSMPI_v7_x64.dll;64-bit Microsoft MPI
solver
SolverCmdLineMPI=job submit /numsockets:%1 /JobName:FlowVisionHPCSolver mpiexec.exe
FvSolver64.exe mpi=libFvMPI_MSMPI_v7_x64.dll;64-bit Microsoft Compute Cluster solver
DebugLog=No
SolverProxyHost=
SolverProxyPort=
```

4.8.5.2.1 Parameters for starting Solver

Methods of starting **Solver** are defined by lines in configuration file of Solver-Agent (FvSolverAgent.cfg) with parameters `SolverCmdLine` and `SolverCmdLineMPI`, containing various variants of command lines for starting **Solver** and, in the comment field, names of these variants that will be displayed as option for selection by the user in client modules.

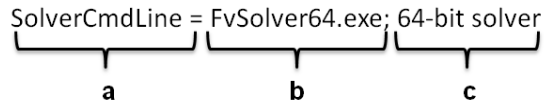
String from the **Solver-Agent**'s configuration file is converted by **Solver-Agent** into a command line, which is clear to the operating system.

Lines in the **Solver-Agent**'s configuration file for starting **Solver** are of two types: single-processor lines (`SolverCmdLine=...`) and multiprocessor lines (`SolverCmdLineMPI==...`).

Specifying a single-processor command line

These lines are designated in the **Solver-Agent**'s configuration file by the `SolverCmdLine` parameter.

Example:



in this example:

a (from the beginning of the line to the symbol "=") is the *name* of a parameter in the **Solver-Agent**'s configuration file, i.e. it is SolverCmdLine

b (from the symbol "=" to the symbol "*;*") is the *value* of the parameter in the **Solver-Agent**'s configuration file, which is actually the command line to start **Solver**. In the simplest case, this is a path to the **Solver**'s executable file.

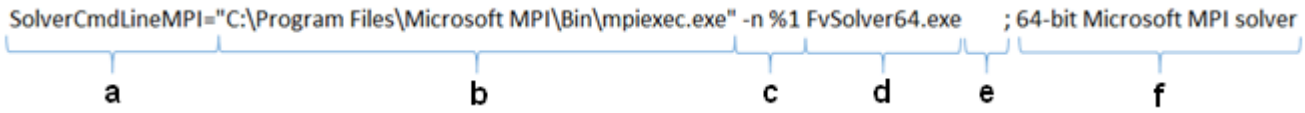
c (from the symbol "*;*" to the end of the line) is a comment, which will also be displayed in client modules in drop-down lists as an option for selection parameters of **Solver** to be started (for example, in the **Type of solver** list in the **Select solver** dialog box).

Command lines specified in lines `SolverCmdLine=...`, define types of starting **Solver** that are available for selection by the user when **Procs=1** is set in the **Select solver** dialog box.

Specifying a multiprocessor command line

These lines are designated in the **Solver-Agent**'s configuration file by the `SolverCmdLineMPI` parameter.

Example:



in this example:

a (from the beginning of the line before the **"=")**- the keyword (parameter name) in the configuration file**Solver-Agent**

b, **c**, **d** and **e** (from the symbol "=" to the symbol "*;*") are the *value* of the parameter in the **Solver-Agent**'s configuration file, it defines the command line to start **Solver** via *MPI*. In this command line, part **b** is the path to the executable file of *MPI*, parts **c** and **d** are parameters of *MPI*, and **e** contains parameters of **Solver**. More details about this:

c contains symbols **"-n"**, which form an *MPI*'s flag (keyword) that defines number of processors, running by *MPI*. **Solver-Agent** automatically replaces symbols **"%1"** with number of processors, specified by the user in the interface of the client module when **Solver** is started in multiprocessor mode (for example, this is specified in the **Number of processors** field in the **Select solver** dialog box).

d is the path to the executable file, which will run for parallel computing through *MPI*. Generally, this is a path to the **Solver**'s executable file.

e contains parameters of **Solver**.

f (from symbol "*;*" to the end of the line) is a comment, which will also be displayed in client modules in drop-down lists as an option for selection parameters of **Solver** to be started (for example, in the **Solver type** list in the **Select solver** dialog box).

Command lines specified in lines `SolverCmdLineMPI=...`, define types of starting **Solver** that are available for selection by the user when **Procs≥2** is set in the **Select solver** dialog box.

Multiprocessor command line for *Windows*

To run **Solver** in multiprocessor mode in *Windows*, it is necessary to place at the end of the command line the **mpi=** parameter, after which goes the name of the the library for connection with *MPI* (the library must locate in the same directory as the executable file **`FvSolver64.exe`**).

Example of a command line, which runs multiprocessor **Solver** using *Microsoft MPI*:

```
SolverCmdLineMPI="C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n %1 FvSolver64.exe
mpi=libFvMPI_MSMPI_v7_x64.dll;64-bit Microsoft MPI solver
```

Full command line to run Solver

After the user selected in the client module's interface the type of **Solver**, specified number of processors and cores, and sent to **Solver-Agent** a command to start **Solver**, **Solver-Agent** modifies the command line for starting **Solver** and executes it using the operating system.

Modification of the command line is done according the following rules:

1. When **Solver-Agent** finds symbols "**%1**" in a multiprocessor command line, it replaces them with the number of processors, which has been specified by the user in the interface of a client module.
2. **Solver-Agent** adds several parameters for **Solver** at the end of the command line (into the part **e** after the path to **Solver**):
   - **Solver-Agent**'s username in the form of `sa_user=username`.This parameter specifies to which user this **Solver** will be available in the list of **Solvers**.
   - unique identifier for **Solver** in the form of sa_ID=111-111111
   - the number of cores in the form of:
     - `threads=numa` (if in the interface of the client module the checkbox "**Use all cores of each processor**" is checked, then, using the *NUMA* technology, **Solver** will automatically determine the optimal number of threads per processor)
     - *or* `threads=`***n***, where an integer value ***n*** is number of nuclei

For example, if the **Solver-Agent**'s user with name **johnsmith** run **Solver** on two processors using all cores, the operating system will execute a command like this:

```
"C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n 2 "C:\Program Files\FlowVision-
3.13.01\FvSolver64.exe" mpi=libFvMPI_MSMPI_v7_x64.dll sa_ID=111-111111 sa_user=johnsmith
threads=numa
```

NUMA is a technology that allows to determine the optimal number of threads on which **Solver** is to be run. Typically, the optimal amount of threads corresponds to the number of cores on one processor. In some cases, the optimal number of threads might be less than optimal number of cores. If the *NUMA* mode is not supported on a computer, then when the parameter `threads=numa` is used, number of cores on a processor will be determined incorrectly (it will be equal to the total number of cores on the computer).

Placing Solver into a queue when using *Microsoft HPC Pack (Windows HPC Server)*

To interact with the queue system in *Microsoft HPC Server* it is possible to use the following syntax of the command line:

```
SolverCmdLineMPI=job submit /numsockets:%1 /JobName:FlowVisionHPCSolver mpiexec.exe
FvSolver64.exe mpi=libFvMPI_MSMPI_v7_x64.dll;64-bit Microsoft Compute Cluster solver
```

Here `job submit` is a system command for placing into a queue and `/numsockets` and `/JobName` are parameters of this command.

Use of *MS MPI* is supposed, so the `mpi` parameter specifies the name of the appropriate library for connection with *MPI*.

More details about placing tasks into the *Microsoft HPC Server*'s queue system you can find in the documentation for *Microsoft*'s products.

4.8.5.2.2 Command line for calculations on several computers (on a cluster)

When running in multiprocessor mode on a single computer, the command line in settings of **Solver-Agent** looks like this:

```
"SolverCmdLineMPI="C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n %1 FvSolver64.exe
mpi=libFvMPI_MSMPI_v7_x64.dll;64-bit Microsoft MPI solver
```

Here *MPI* receives the following parameters: the number of processors (`-n %1`) and the path (relative in this example) to the executable file of **Solver** (`FvSolver64.exe`), and also the name of the library for connection with *MPI*. The parameter "`mpi=`" specifies the name of the library for connection with *MPI* (*MS MPI* in this example), which locates in the same directory, as the executable file `FvSolver64.exe`.

If parallel computations are performed on several computers (on a cluster), *MPI* has to receive additional information, which is specified in the command line by additional parameters.

When using *MS MPI*, it is necessary to specify the following additional parameters:

- `-pwd` specifies a user (registered in the operating system), as which **Solver-Agent** runs and as which **Solvers** will be started.
- `-machinefile` specifies the path to the machine MPI file, which contains list of IP addresses of computers, on which **Solvers** are to be run.
- `-wdir` specifies the network path to the directory with the executable file of **Solver** (`FvSolver64.exe`). Generally this is the installation directory of *FlowVision*, which is accessible by its network path from any computer (cluster node), on which **Solver** will run.

Example of the command line in the Configuration file of Solver-Agent (FvSolverAgent.cfg) for *MS MPI* in *Windows*:

```
"C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n %1 -machinefile "C:
\DirForMachinefileOnHeadMachine\hosts.txt" -pwd SharedUserPassword -wdir "\
\HeadMachine\SharedWorkDirectory" FvSolver64.exe mpi=libFvMPI_MSMPI_v7_x64.dll ; 64-bit
Multihost MS MPI solver
```

Directories with settings of **Solver** and **Solver-Agent** have to be accessible for reading and writing from all nodes by the same network path (for example, `\\HeadMachine\SharedWorkDirectory\Settings`). This network path is to be specified in the file `Fv.cfd`, for example, like this: `HOMESRV=\\HeadMachine\SharedWorkDirectory\Settings`.

After execution of this command line, on each compute that is listed in the file `hosts.txt`, *MPI* will authorize as the user, which has started the command line, and which is registered on each of these computers (for example, user **SharedUser**), with password **SharedUserPassword** and will run copies of the process `\\HeadMachine\SharedWorkDirectory\FvSolver64.exe`.

For successful start of copies of **Solver**'s process on each computer listed in the file `hosts.txt`, it is necessary that on each of these computers the *MS-MPI Launch Service* (*MsMpiLaunchSvc*) service has been launched.

> ***Please note:*** During the automatic installation of *MS MPI v.7*, the *MS-MPI Launch Service* (*MsMpiLaunchSvc*) service will also be installed but not launched. You have to launch it manually. Usually, it is convenient to tune an automatic launch of this service at the start of *Windows*. You can tune this in in the *Windows'* **Control Panel** (**Administrative Tools > Services**).

For details on machine files and user files, see sections Machine MPI file and User MPI file and documentation of *MS MPI*.

4.8.5.2.2.1 Machine file MPI

Machine MPI file is a text file (`txt`), lines of which contain network names of the computers, on which **Solver** will run. In each of these lines, after the computer's name (separated by a space for *MS MPI*) the number of *MPI* processes, which will run on this computer, might be specified. Generally, it is optimal to specify the numbers of *MPI* processes as be equal to the number of processors (or *NUMA* nodes) that are available on the computers (see section Hybrid approach to parallel computations). If the number of processes is not specified in a some line, then only one *MPI* process will run on the corresponding computer.

The computers' names might repeat in the machine file. Processes of **Solver** are started on the computers according the sequence as the computers are specified in the machine file and, after run on the last computer, if some requested processes have not been allocated, the processes will continue be started again on the first computer, then on the next, etc. (in a cycle).

Example of the machine *MPI* file to run Solver on a single computer:

```
192.168.1.2
```

Sample of the machine *MPI* file to run Solver on a cluster with one processor per node:

```
192.168.1.1
192.168.1.2
192.168.1.3
192.168.1.4
```

Contents of the machine *MPI* file when there are multiple processors per node

When there are multiple (more then one) processors on a node, your should explicitly specify in the appropriate line of the machine file the number of processors. See below an example of the of the machine *MPI* file for *MS MPI*, where the number of processors is specified after the node's name or address, separated by a space:

```
192.168.1.1 2
```

```
192.168.1.2 2
192.168.1.3 2
192.168.1.4 2
```

If you don't specify numbers of processors on the nodes, then, at the attempt to run several processes on one node, **Solver** will run on the first processor only (to be more exact, on the first *NUMA* node, see section Hybrid approach to parallel computations). For example, if two processes are started on a node with two processors, both processes will use only the first processor, while the second processor will be idle.

4.8.5.2.2.2 User file MPI

User MPI file is a text *file* (`txt`) containing the name and password of the user, as whom **Solver** will run on computers listed in the machine MPI file.

Structure of the user *MPI* file

Parameter	Description
USERNAME	Username as which **Solver** will run. If the user belongs to a domain, the username must also include the domain's name.
PASSWORD	Password of the user, as which **Solver** will run. It is recommended to use a password that only contains digits and Latin characters. The password is mandatory.

Sample of the user *MPI* file

```
DOMAIN/user
123456
```

4.8.5.2.3 Starting Solver through Solver-Agent by user script

It is typical situation when it is necessary to start **Solver**, doing some preliminary steps in the operating system (setup environment variables, form on the fly a list of hosts for *MPI*, etc.). In such situation, you can run **Solver** by a user script, in which all the necessary steps a done and then **Solver** is started.

User script can be run either manually from the operating system's console or through **Solver-Agent**. In the latter case, the end user will run **Solver** with a simple and convenient interface of client modules.

Example of a line from Configuration file of Solver-Agent, which starts script `runsolver.bat`:

```
SolverCmdLineMPI=C:\runsolver.bat %1 ; User script
```

From the viewpoint of **Solver-Agent**, it does not matter what starts (**Solver**, *MPI* or user script).

An example of the contents of the script `runsolver.bat` for *Windows*:

```
rem This script will run Solver
"C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n %1 -machinefile "C:
\DirForMachinefileOnHeadMachine\hosts.txt" -pwd SharedUserPassword -wdir "\
\HeadMachine\SharedWorkDirectory" FvSolver64.exe mpi=libFvMPI_MSMPI_v7_x64.dll sa_ID=%3
threads=%5
```

Parameters `%1`, `%3`, `%5` in this script are parameters that are passed to the script from **Solver-Agent**.

Parameter `%1` has been specified by us in the line of the configuration file of Solver-Agent, which define command line with parameters for starting Solver. In this example it is the number of processors specified in the user interface at starting **Solver**.

Parameters `%2` and `%4` always contain values **sa_ID** and **threads** respectively (due to specifics of transferring and handling parameters in the program `cmd`).

In *Linux* the similar script is the following:

```
#!/bin/sh
mpiexec -n $1 FvSolver64 $2 $3
```

Using one script and several lines from the configuration file of **Solver-Agent** (with different parameters), it is possible to implement, for example, starting different versions of **Solver** from a single startup script.

4.8.5.2.4 General recommendations for use of command lines

- When computation is run on multiple computers, you have to specify the full path to executable files.
- When you run **Solver** in the multiprocessor mode, **Solver-Agent** starts *MPI*, which, in turn, runs **Solver**.
- When computation is run on a cluster in the batch mode, you have to specify the path to the batch file that is accessible to all nodes of the cluster.
- If the commentary (part of the line after the symbol "`;`" to the end of the line) in the file's line is absent, then the appropriate command line will not be visible when you start **Solver** from **Pre-Postprocessor** or **Terminal**.
- If there is a space in a path, then it is recommended enclose the path with a file name in double quotes: `"path/filename"`.
- If, on your computer, the *NUMA* mode is not supported or it is not configured, then, when you use the `threads=numa` parameter, number of cores per processor will be determined incorrectly (equal to the total number of cores on the computer).
- When you run **Solver** from any client module, **Solver** receives its identifier automatically (see *Note* in section Configuration file of Solver (FvSolver.cfg)). When **Solver** is started from the command line, it is necessary to specify the identifier manually (with a unique value to differentiate this **Solver** from other **Solvers**).

4.8.5.3 Configuration file of Solver (FvSolver.cfg)

Configuration file `FvSolver.cfg`		
Parameter	**Description**	**The default value**
PPPBasePort	See Parameters in configuration files	**11000**
FvLicenseHost	See Parameters in configuration files	**127.0.0.1**
FvLicensePort	See Parameters in configuration files	**10010**
SAHost	See Parameters in configuration files	**127.0.0.1**
SASolversPort	See Parameters in configuration files	abce**1** where abce - is the 1st, 2nd, 3rd and 5th digits of the version number. ***Example:*** **30911**
DebugLog	See Parameters in configuration files*)	**No**
CompressLayers	• **Yes** - pack layers before sending to **Pre-Postprocessor** and **Viewer** • **No** - do not pack the layers before sending to **Pre-Postprocessor** and **Viewer**	**Yes**
SleepMPIProcess	Latency for *MPI*-processes in the standby mode, [ms]. In some systems, **Solver**, even when it is not computing, may consume resource of *CPU*, which is especially critical when using the `mpi_skip` parameter. The presence of such a delay allows the program to avoid this negative effect, but this can also slow down the computations in problems with a relatively short duration of iterations. If when the parameter is zero and **Solver** is idle, CPU is not loaded by **Solver**, then setting a zero value of this parameter can give a small advantage of **Solver**'s performance.	*by default these parameters are absent in the configuration file*
LicenseTimeout	The time in minutes, during which **Solver** will wait appearing a vacant (if all licenses are used). The requests, which are waiting for a license, are displayed in **License monitor** in the **Waiting requests** pane.	

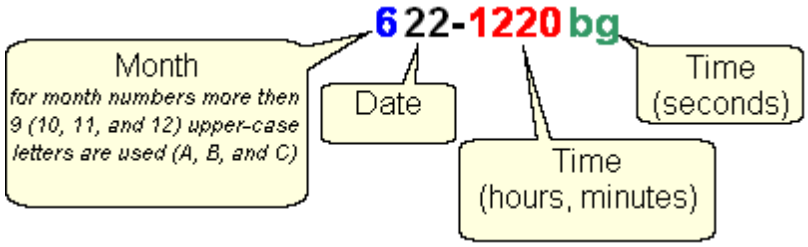
Configuration file `FvSolver.cfg`		
Parameter	**Description**	**The default value**
	If the required license(s) will not be freed during this time, then **Solver** will output an error message about no free licenses. If **LicenseTimeout=0**, then **Solver** will output the error message about no free licenses immediately and will not wait possible freeing of licenses.	
SaveAfterAbaqus	When **SaveAfterAbaqus=Yes**, it is possible to repeat the last step of the computation before crash of **Solver** with unknown error (without using *Abaqus* but using the geometry, which came on this step). This can help to troubleshoot the project and find errors that cause the crash of the computation. To use the **SaveAfterAbaqus** parameter, you have to carry out the following actions when the joint computation crashes (except crashes caused by self-intersections of the geometry or errors arisen in *Abaqus*): • In `fvproj` file, write `*DebugMpmConnector` into the line, which contains the `<ConnectorID>` tag as in the example below: `<ConnectorID>*DebugMpmConnector</ConnectorID>` • Save and close the `fvproj` file • Restart (continue) the computation from the last step and obtain the crash of **Solver**	
UseIPv6	Operation using the *TCP/IPv6* protocol. Possible options are: **Yes** | **No**. For operation in *IPv6* networks you have to specify **UseIPv6=Yes** and addresses in settings of other modules have to be specified using the *IPv6* format or as domain names.	**No**
DeformLog **LoadingsLog**	These parameters tune output of additional information during a joint computation. Possible options for these parameters are: **Yes** | **No**. The **DeformLog=Yes** enables output of files with information that *FlowVision* receives from the external software. The **LoadingsLog=Yes** enables output of files with information that *FlowVision* transfers to the external software. *FlowVision* outputs the additional information only when joint computation is running. In the case of the single computation(when *FlowVision* only operates), the additional information will not be output.	**No** (because the output files are big and recording to them is done at each step)
FinalRTBLAutoSave	When this setting is enable, then, when the table of external input parameters is used and **Solver** terminates normally, *FlowVision* creates an `fvrtbl_f`-file. This is useful for arranging controlling *FlowVision* by some external software, for example, such as *IOSO*. Creation the `fvrtbl_f`-file means that *FlowVision*'s **Solver** terminated due to fulfillment its stopping condition and unloaded correctly.	**No**

Configuration file `FvSolver.cfg`		
Parameter	**Description**	**The default value**
	If the **Solver** terminated abnormally, then the `fvrtbl_f-` file would not be created and the controlling external software would understand this situation. Possible options are: **Yes** | **No**.	

Note:

*) `FvSolverComm_XXX-XXXXXX.log` contains messages about connecting to server modules of *FlowVision*. `Commander_XXX-XXXXXX.log` contains messages about execution of commands for **Solver** where `XXX-XXXXXX` is **Solver**'s identifier.

Each **Solver** at its start receives an identifier. The identifier is formed based on the date and time of the **Solver**'s start. The identifier has the following format:



Solvers' identifiers are used in names of the log files of corresponding **Solvers** and you can see them in **Terminal** and **Pre-Postprocessor** in lists of active **Solvers**.

4.8.5.4 Configuration file of Terminal (FvTerminal.cfg)

Configuration file FvTerminal.cfg

Parameter	Description	The default value
ViewerDir	Path to the directory where **Viewer** is installed. The path may be absolute or relative (relatively to location of the `FvTerminal.exe` file). If this parameter is absent or has an empty value, **Terminal** assumes that **Viewer** locates in the same directory as **Terminal**.	By default this parameter is absent
DebugLog	See Parameters in the configuration files	**No**

Example file FvTerminal.cfg

```
ViewerDir=C:\Program Files\FlowVision\FlowVision HPC
DebugLog=No
```

4.8.5.5 Configuration file of Pre-Postprocessor (FvPPP.cfg)

Configuration file `FvPPP.cfg`

The file `FvPPP.cfg` stores current settings of **Pre-Postprocessor**.

To edit the settings, apply user interface of Pre-Postprocessor, do *not* edit the file `FvPPP.cfg` manually using a text editor.

The file `FvPPP.cfg` is coded in the XML format. Manual editing this file can destroy its structure and cause incorrect operation of **Pre-Postprocessor**.

4.8.5.6 Configuration file of Viewer (FvViewer.cfg)

Configuration file FvViewer.cfg

Parameter	Description	The default value
DebugLog	Recording the debug messages into the log files: • **Yes** - recording is enabled • any other value or absence of this parameter means that recording is disabled	**No**

Example of the FvViewer.cfg file

```
DebugLog = No
```

4.8.5.7 Configuration file of MPM-Agent (MpmAgent.cfg)

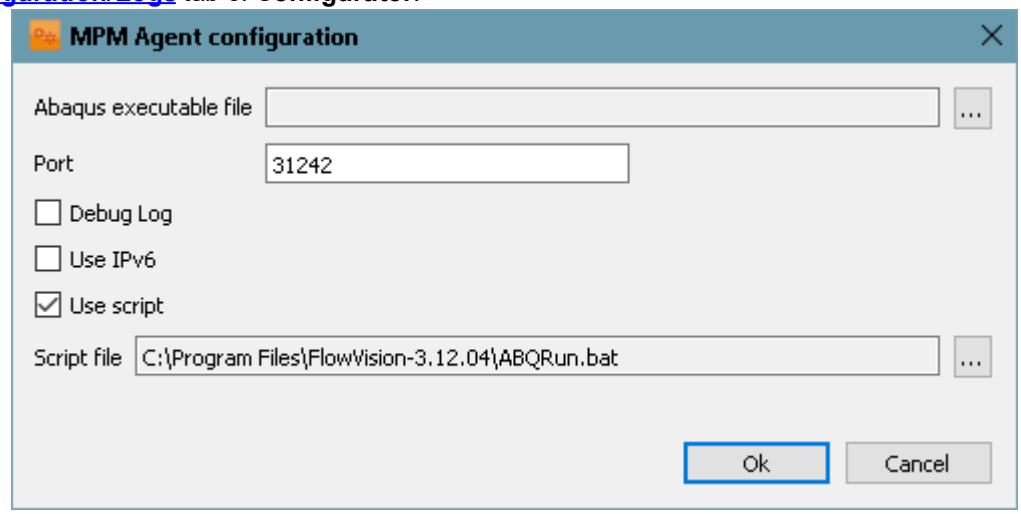
Configuration file MpmAgent.cfg

Parameter	Description	The default value
Port	The port number that will be created and on which **MPM-Agent** will listen for connections from the *FlowVision* server.	abce**2** where abce are the 1st, 2nd, 3rd and 5th digits of the version number of the program. For example, for version *30901* the default value for the port number will be `30912`.
Abaqus	An *Abaqus* executable file (its network path and file name are specified)	
DebugLog	Recording debug messages into log files. See section Parameters in configuration files	`No`
UseIPv6	Operation using the *TCP/IPv6* protocol. Possible options are: `Yes` | `No`. For operation in *IPv6* networks you have to specify `UseIPv6=Yes` and addresses in settings of other modules have to be specified using the *IPv6* format or as domain names.	`No`
UseScript	Use of a script to run a joint computation. Possible options are: `Yes` | `No`. When `UseScript=Yes` is specified in the configuration file, a script will be used, which is set by the **Script** parameter (see below). When `UseScript=No` is specified, then an *Abaqus* executable file will be used, which is set by the **Abaqus** parameter (see above).	`Yes`
Script	A file with a script to run a joint computation (its network path and file name are specified). By default the program uses a script from its distribution pack that allows you to run a joint computation for **DC** (direct coupling) and for **CSE** connectors. You can specify here your own script to which parameters for starting the joint computation will be transferred: • network path and name of the *Abaqus* executable file	`C:/Program Files/FlowVision-3.13.01/ABQRun.bat` *(digits here correspond to the FlowVision's version number)*

Parameter	Description	The default value
	• communication port that is set in *FlowVision* • the **ask_delete** parameter for running *Abaqus*	

Configuring parameters of MPM-Agent in graphical user interface of Configurator

You can set parameters of **MPM-Agent** from graphical user interface of **Configurator**, in the **MPM Agent configuration** dialog box, which opens on clicking the **Configuration files > MPM Agent > View** screen button in the Configuration/Logs tab of **Configurator**.



The **MPM Agent configuration** dialog box has the following interface elements that correspond to parameters in the configuration file of **MPM-Agent**:

Interface element	Correspondence to a parameter in the configuration file of MPM-Agent	Description
Abaqus executable file	**Abaqus**	An *Abaqus* executable file
Port	**Port**	The port number at which **MPM-Agent** will wait for connection from server of *FlowVision*.
Debug Log	**DebugLog**	Recording debug messages into log files
Use IPv6	**UseIPv6**	Operation using the *TCP/IPv6* protocol
Use script	**UseScript**	Use a script for running the joint computation
Script file	**Script**	A file with a script to run a joint computation

Example of the MpmAgent.cfg file

```
DebugLog=No
Port=31242
Abaqus=
```

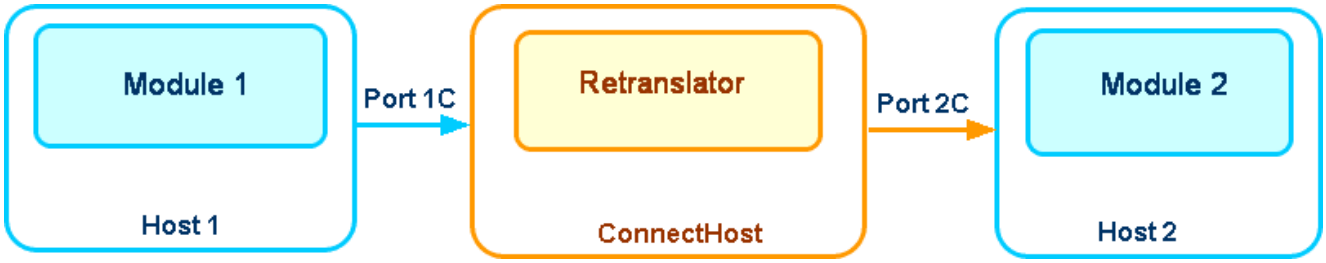
4.8.5.8 Configuration file of Retranslator (FvConnect.cfg)

Configuration file FvConnect.cfg

Parameter	Description	The default value
Channel	Possible values: • **Port1C:Host2:Port2C**, if direct re-translation is used. • **Port1**, if re-translation through Solver-Agent is used.	*no default value*
DebugLog	Recording debug messages into log files: • **Yes** means that the recording is enabled	**No**

Parameter	Description	The default value
	• any other value or absence of this parameter means that the recording is disabled	
UseIPv6	Operation using the *TCP/IPv6* protocol. Possible options are: **Yes | No**. For operation in *IPv6* networks you have to specify **UseIPv6=Yes** and addresses in settings of other modules have to be specified using the *IPv6* format or as domain names.	**No**

4.8.5.8.1 Direct re-translation



To redirect a request from **Module 1** software, which is installed on the computer **Host 1**, to **Module 2**, which is installed on the computer **Host 2**, it is necessary before running of these modules to do the following:

- In the interface or in the configuration file (**cfg**-file) of **Module 1** specify, as parameters for communication with **Component 2**, an IP address or host name of the computer on which **Retranslator** is installed (the computer **ConnectHost**) and the number of any port (**Port1C**), which is free on both computers. It is possible to use the default port if it is free.
- In the configuration file (**cfg**-file) of **Module 2** set the number of any free port (**Port2C**). It is possible to use the default port if it is free.
- In Configuration file of Retranslator (FvConnect.cfg) it is necessary to specify `Channel=Port1C:Host2:Port2C`.

4.8.5.8.2 Re-translation via Solver-Agent

If it is necessary to organize the interaction between **Solver** and a client module, and **Solver** is running on a computer, which does not have a direct connection with the client module, and **Solver-Agent** is running on a computer, which has connection with the client module, then it is necessary to provide re-translation through **Solver-Agent**:

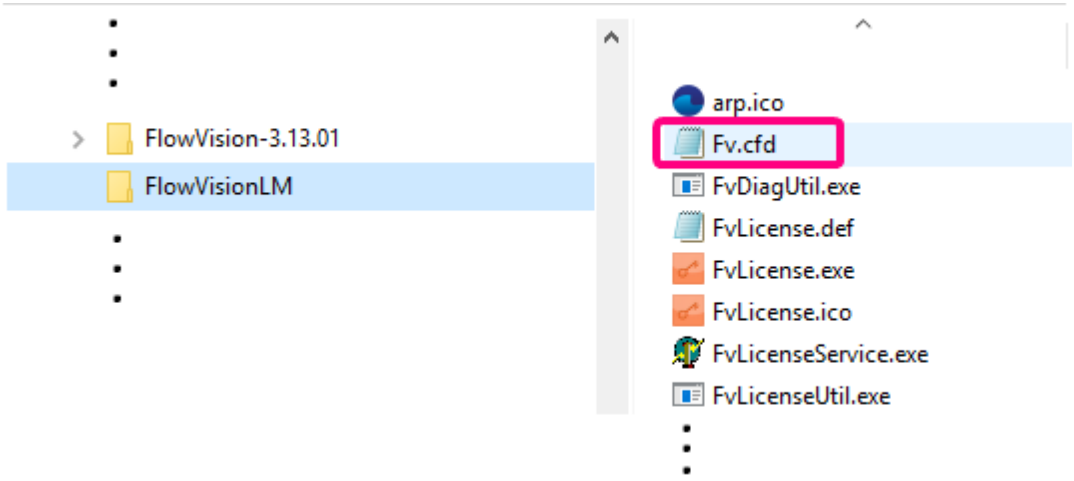
- In the configuration file (cfg-file) of Solver-Agent specify `SolverProxyHost` as an IP address or host name of the computer where **Retranslator** is installed, and specify `SolverProxyPort` as the number of any free port on the computer where **Retranslator** is installed.
- In the configuration file (cfg-file) of Retranslator specify a string `Channel=Port1`, where `Port1` must be same port as specified in `SolverProxyPort`.

4.8.5.9 Files Fv.cfd

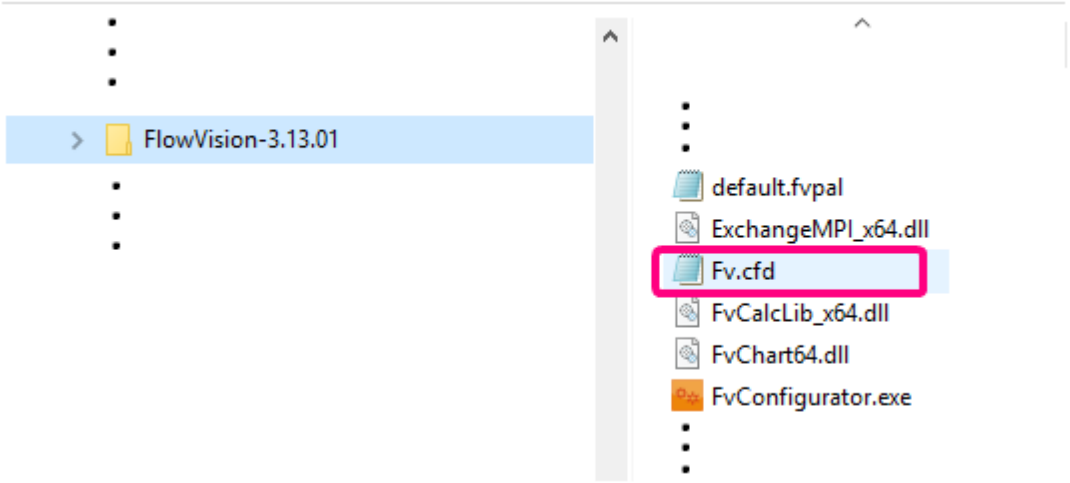
Files `Fv.cfd` locate in:

- the installation directory of **License Manager**
- the installation directory of other modules of *FlowVision*

In each of these directories there is an one file `Fv.cfd`, which contains information about location of the corresponding user directories.



File **`Fv.cfd`** in the installation directory of **License Manager**



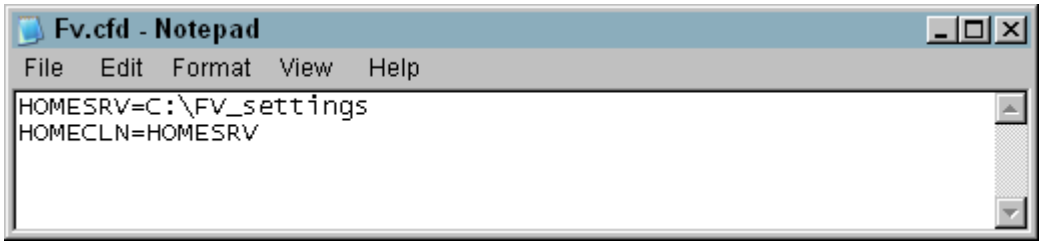
File **`Fv.cfd`** in the installation directory of other *FlowVision* modules

File **`Fv.cfd`** indicates user directories and contains the following parameters:

Parameter	Description
HOMESRV	Absolute path in the file system to the user directory of server modules of *FlowVision.* **HOMESRV =*(path)***
HOMECLN	The path to the user directory of client modules of *FlowVision.* Possible options: **HOMECLN = SYS** User directory of client modules is located in the system directory of the user: • for *Windows* - **`C:\Documents and Settings\username`** • for *Linux* - **`/home/username`** *Each user has its own system directory!!! When you run the component as different users, different system directories will be used.* **HOMECLN = HOMESRV** The user directory of the client modules will be the same as the user directory of server modules.

Viewing and changing of the contents of the **`Fv.cfd`** files can be done using a text editor, or (for *FlowVision* modules except **License Manager**) using **Configurator**, in the tab **Configuration/Logs** after clicking on the **User directories** button.

Example of the Fv.cfd file



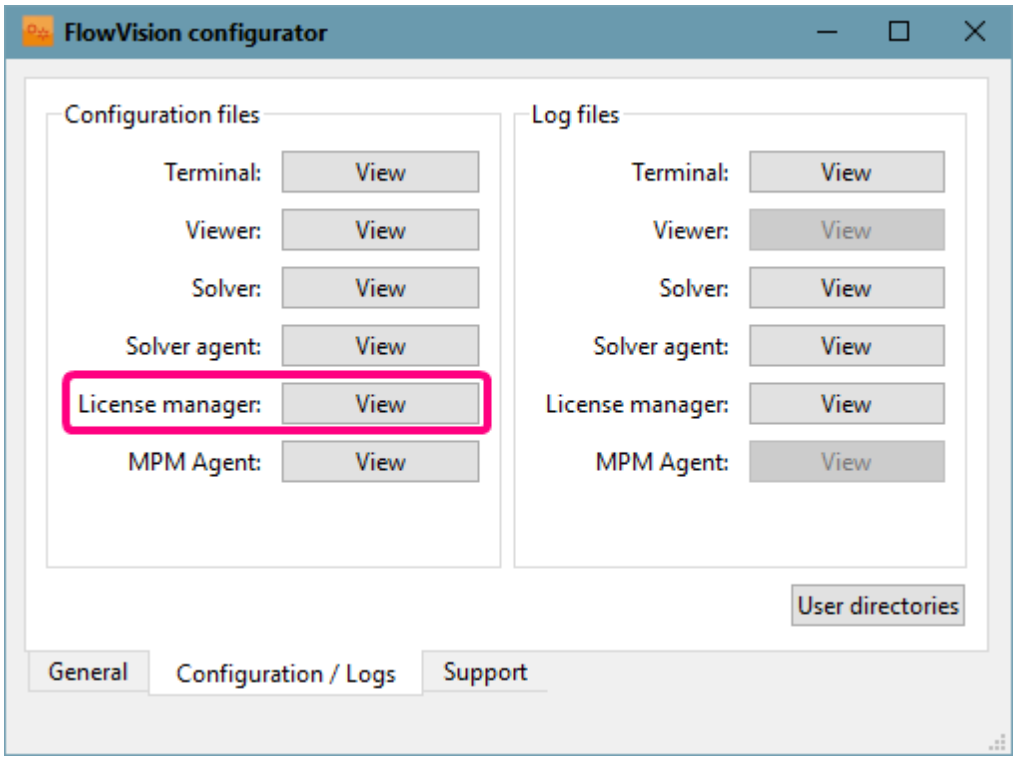
4.8.6 Errors during setup

Error messages that occur during setting up (configuring) are recorded into log files. The log files can be viewed using **Configurator** (tab **Configuration/Logs**, group of settings **Log files**). You can also see there locations of these files (see section User directories).

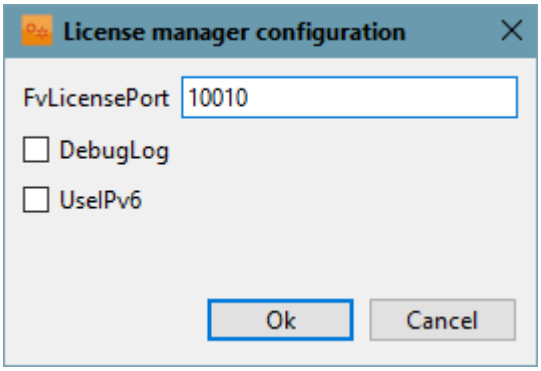
Error when starting License Manager (`FvLicense.exe`)

If you had tried to run **License Manager**, have no success, and in its log file `FvLicense.log` information appeared about an error of unsuccessful creating a socket *"Cannot create a server socket, port (port number)*", this means that the default port **10010** is busy (might be because of previous run of **License Manager**). In this case you have to specify another port for **License Manager**, for example, **10011** and, if the error repeats (if the new port it is also busy), increase the port number by **1** and repeat attempts to run **License Manager** until success.

To specify the port for **License Manager**, run **Configurator**, open the **Configuration/Logs** tab, and in the **Configuration file** pane click the **License manager: View** screen button:

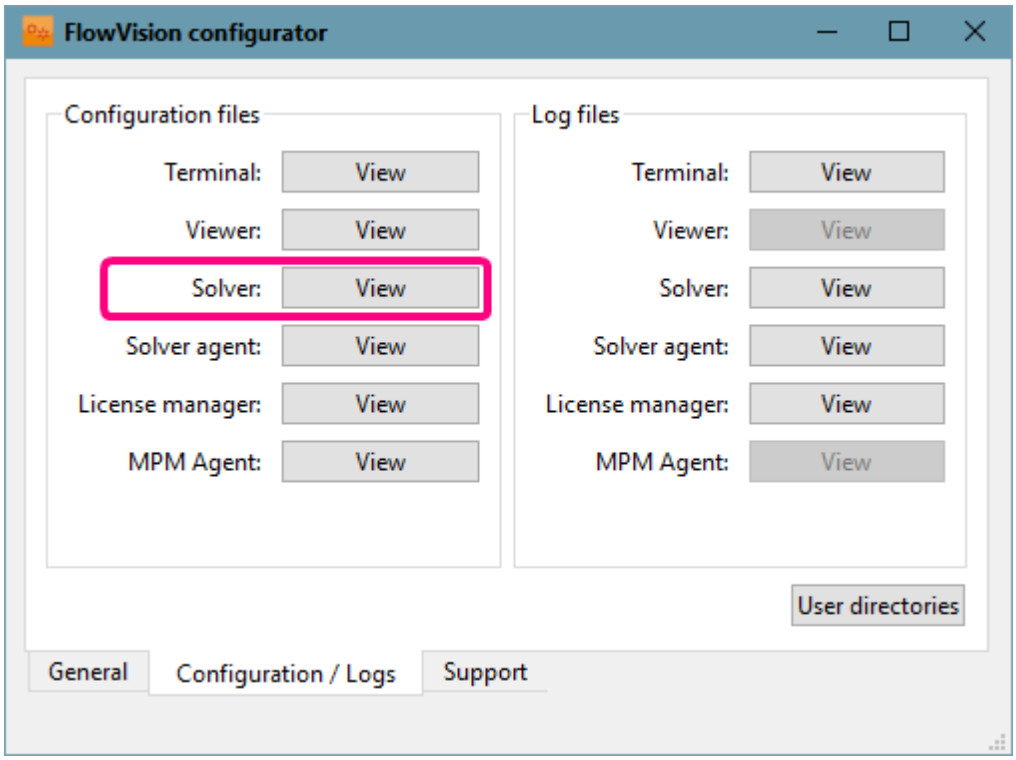


In the **License manager configuration** dialog box, which opens, specify the port number in the **FvLicensePort** field and click **Ok**:



The port number of **License Manager** and other settings of **License Manager** are stored in the configuration file of **License Manager** `FvLicense.cfg`.

When you find out the correct port number, specify this value in settings of other modules:

Module	Access to setting from the user interface
Solver	In Configurator open the Configuration/Logs tab, and in the **Configuration file** pane click the **Solver: View** screen button:  In the **Solver configuration** dialog box, which opens, specify the port number in the **FvLicensePort** field and click **Ok**. The new port number will be saved in the **Solver**'s configuration file (`FvSolver.cfg`).
Pre-Postprocessor	Open basic settings of Pre-Postprocessor (apply the **File > Preferences** command from the main menu of **Pre-Postprocessor**) and specify the correct port number in the **License manager > Port** setting. The new port number will be saved in the **Pre-Postprocessor**'s configuration file (`FvPPP.cfg`).

Error when starting Solver-Agent (`FvSolverAgent.exe`)

If you had tried to start **Solver-Agent**, and it did not run, and in the log file of **Solver-Agent** (`FvSolverAgent.log`) appeared information about an error of creating a client socket *"Cannot create a client socket, port (port number)*", then it is required in the configuration file of **Solver- Agent** (`FvSolverAgent.cfg`) set the parameter `SAClientsPort` as **10100**. If the error persists, increase the parameter's value by **1** and repeat attempts to start. After you find the correct port number, this value must be set up in other configuration files:

Configuration file	Location of settings in the user interface	Parameter
-	**Terminal** (user authentication dialog)	**Port**
`FvPPP.cfg`	**Pre-Postprocessor** (**File > Preferences**)	**Solver-Agent >Port**

If you had tried to start **Solver-Agent**, and it did not run, and in the log file of **Solver-Agent** (`FvSolverAgent.log`) appeared information about an error of creating a server socket *"Cannot create a server socket, port (port number)*", then it is required in the configuration file of **Solver- Agent** (`FvSolverAgent.cfg`) set the parameter `SASolversPort` as **10100**. If the error persists, increase the parameter's value by **1** and repeat attempts to start. After you find the correct port number, this value must be set up in other configuration files:

Configuration file	Customize the user interface	Parameter
`FvSolver.cfg`	Configurator (Configuration/Logs > **Configuration files > Solver**)	**SASolversPort**

4.8.7 Inspection of correctness of installation and setup

Preparing for inspection

To check correctness of settings of client modules, **Solver-Agent** and **License Manager** have to be run before this.

The inspection assumes that you have a registered license.

To check the whole software complex and network interactions, it is recommended to follow sequentially instructions in subsections of this section. For the check you will need to use the project **Lam_tube** from the tutorial, which is located, in the subdirectory `Tuturial/Samples/EnuProjects/Lam_tube` of the directory where **Postprocessor** is installed.

Important! The **Lam_tube** project must be copied into a directory where you have all rights to read and write.

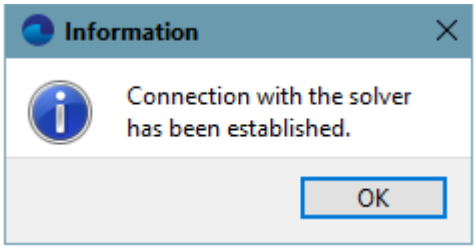
Inspection of Pre-Postprocessor

Step	Subject for the check	Actions	The expected result in the case of the correct operation of the program	Possible errors, their causes and ways for fixing
1	Overall integrity of the installation	Start **Pre-Postprocessor**.	The program's window opens. The **File** menu must be available. Error message(s) must not be displayed.	**The program does not run or fails:** Try restoration of damaged installations or reinstall the program by running the installer as Administrator. Make sure that the program that runs as the current user of the operating system has rights to read and write in user directory of client modules.
2	Check of settings for connection to **License Manager**	Open the project **Lam_tube**.	In the window **Project** the project tree will appear. In the window **View** an image of a tube will appear. In the window **Log** the latest message must include the phrase **"Information: New project has been successfully loaded from ...**". Appearance of some	Possible causes of the errors: 1. There are no permissions to read and write to a temporary system directory. 2. There are no permissions to read and write to the directory with the client part of the project. 3. Outdated drivers for your video card are installed (update the drivers and read information in the section Known limitations).

Step	Subject for the check	Actions	The expected result in the case of the correct operation of the program	Possible errors, their causes and ways for fixing
			warnings in the **Log** window is acceptable.	
3	Check of correctness of installation of some libraries	In the window **Project** click the element **Region> General settings** and in its **Properties** window, set for the parameter **g-Density** its value as **1**. Click in the **Properties** window the **Apply** button.	The **g-Density** parameter in the **Properties** window will change its value.	If the **Properties** window is empty, then the installation was incorrect. Do the installation again, running the installer with Administrator's rights.
4	Check of settings for connection to **Solver-Agent** (**Solver-Agent** must be running and has been configured correctly)	Connect to **Solver-Agent**: click the button (**Solver agent log in**) in the **Network** toolbar, and then select a connection **Configuration** in the **Solver agent connection** dialog box (it opens if at least one connection has been created). If no **Configuration** has been created yet, create a **Configuration** as described in the section Connection to Solver-Agent and user authentication on Solver-Agent. If some **Configuration** is set as default one and contains correct settings, then connection to **Solver-Agent** will be done using this **Configuration** automatically at each start of **Pre-Postprocessor**. When a connection to **Solver-Agent** is established, click the button (**Open solver selection window**) in the **Network** toolbar. The **Select solver** dialog box will open.		If a message appears informing about fail of connection to **Solver-Agent**, try to remove the following possible causes of the error: 1. A firewall blocks the connection between modules. 2. **Solver-Agent** has not been started. 3. Incorrect data were specified for connection to **Solver-Agent** (wrong host or port in settings of **Pre-Postprocessor**, the port in settings of **Solver-Agent** does not correspond to the appropriate port specified in settings of **Pre-Postprocessor**), invalid username and/or password of the **Solver-Agent** user.
5	Check of whether the system rights are enough for work of the software	Click the button (**Save changes to the client side of the project**) in the toolbar **Standard** (the button might be unavailable if you skipped step 3). Close **Pre-Postprocessor** by clicking on the symbol "x" in the upper right corner of its window.	Neither after saving a project no after closing the program, error message(s) must not be displayed.	

Inspection of Solver-Agent, Retranslator (if used) and Solver by Pre-Postprocessor

Step	Subject for the check	Actions	The expected result in the case of the correct operation of the program	Possible errors, their causes and ways for fixing
1	Check of general	Follow steps 1-4 from the previous	The **Select solver** dialog box will open.	See steps 1-4 from the previous

Step	Subject for the check	Actions	The expected result in the case of the correct operation of the program	Possible errors, their causes and ways for fixing
	functionality	subsection (*"Inspection of Pre-Postprocessor"*).		subsection (*"Inspection of Pre-Postprocessor"*).
Proceed to step 2a or 2b				
2a	Check of correctness of the command for a one-processor multicore run and **Retranslator** if the latter is used	In the **New solver launch** group of settings in the **Select solver** dialog box do: • In the **Procs** field specify value **1**. • Select the desired type of **Solver** from the **Solver type** drop-down list. • Click the  **Connect** screen button.	A message box will open informing you about successful uploading the project on **Solver** (**"Connection with the solver has been established"**):  Click **OK**. If after this you click the button  (**Open solver selection window**) in the **Network** toolbar, you will see in the table in the **Select solver** dialog box the new running **Solver** with status **CONNECTED**.	If the message **"Waiting for solver response has timed out"** was displayed, try to refresh the list after half a minute. If the new line with information about the **Solver** is not appear, the following causes of the error are possible: 1. The command line, which starts **Solver**, is incorrect. 2. Settings for for connection **Solver** to**Solver-Agent** are incorrect. 3. The network connection is blocked by a firewall. If the status of the new solver is **INACCESSIBLE**, it means that the program failed to establish a direct connection between **Postprocessor** and **Solver**, or settings of **Retranslator** are incorrect. Error messages indicate that there are no rights to read or write in the folder with the client part of the project and/or in the folder with the server part of the project. Possibly an incorrect path has been specified in the user settings of **Solver-Agent**. For example, for a multiprocessor start of **Solver**, different copies of **Solver** might be

Step	Subject for the check	Actions	The expected result in the case of the correct operation of the program	Possible errors, their causes and ways for fixing
or **2b**	Check of correctness of the command for a multiprocess or run and/or settings of Retranslator	In the **New solver launch** group of settings in the Select solver dialog box do: • In the **Procs** field specify value **2** or greater (for cluster systems it is recommended to specify the number of processors that are available when you use more than one node). • Select the desired type of **Solver** from the **Solver type** drop-down list. • Click the  **Connect** screen button.		placed on different nodes. Accordingly, on all of these nodes the same network directory has to be mounted, which must be available for reading and writing from all nodes using the same path.
3	Check of correctness of **Solver**'s settings	Click the button  (**Start computation**) in the toolbar **Solver**. In the **Starting solve** dialog box, which will open, uncheck all checkboxes and click **OK**.	After some time, within a minute, a message will open informing that the computation has been successfully started. In the Monitoring window the values should change at least once within a few next minutes.	If almost immediately after starting the computation a message about insufficient number of licenses will be displayed, then check in the license information whether there is enough enough number of license options for this run mode (number of processors and cores) and for the model used in the project. If a message appears informing that the license server (**License Manager**) has not been found, then it is required to check: 1. Whether the correct license name is specified in the user settings of **Solver-Agent**. 2. Whether **License Manager** is running and whether the settings of **Solver**

Step	Subject for the check	Actions	The expected result in the case of the correct operation of the program	Possible errors, their causes and ways for fixing
				are correct (connection to **License Manager**). 3. Whether the network connection between**Solver** and **License Manager** is not blocked if by a firewall.

4.8.8 Copying settings from previous version of FlowVision

In order not to repeat the procedure of configuring of all *FlowVision* modules when installing a new version of the program, you can use configuration files from the previous version of the program. Perform the following steps, which also do full copying of **Solver-Agent**'s user settings:

Step	Actions
1	Install a new version of the software.
2	Shut down all the modules of the old version of the software.
3	Find a directory with settings of server modules of the *old* version of the software, its name should contain the word **`FlowVision`** and the version number (this directory was specified during the installation of *FlowVision*, see sections Installation on Windows and Installation on Linux). Copy from it the subdirectory **`settings`**.
4	Find a directory with settings of server modules of the *new* version of the software, its name should contain the word **`FlowVision`** and the version number (this directory was specified during the installation of *FlowVision*, see sections Installation on Windows and Installation on Linux). Replace there its **`settings`** subdirectory with the **`settings`** subdirectory, which was copied on the previous step.
5	If you don't plan to use the old version of the software, remove it using standard features of the operating system and skip the remaining steps.
6	If you plan to use of both the old and new versions of the program simultaneously, it is necessary to change settings of **Solver** and **Solver-Agent**.
6.1.	In settings of **Solver-Agent** of the new version of the software, specify **SAClientsPort** as **`xxxy0`**, and **SASolversPort** as **`xxxy1`**.*)
6.2.	In settings of **Solver** of the new version of the software, specify **SASolversPort** as **`xxxy1`**.*)

*) **`xxxy0`** and **`xxxy1`** are port number where **`xxx`** are the first three digits from the version number of *FlowVision*, and **y** is the last digit from the version number.
For example, for version *FlowVision 3.11.04* these ports will be **31140** and **31141**.

Notes:

1. This instruction does not assume transfer of the license.
2. These instruction is only applicable for versions since *3.09.00*.

4.8.9 Specifics of configuring FlowVision in Windows with UAC

If *FlowVision* has been installed into a system directory with restricted access rights (for example, in the directory **`Program Files`**), then changing a directory, which stores settings of server modules, can be done by **Configurator**, which was started only with Administrator's access rights. Also any editing of the file **`Fv.cfd`**, if it is located in a *Windows'* system directory with restricted access, can be only done by a program (a text editor), which runs as Administrator.

Rules of installing *FlowVision* on *Windows* with *UAC*

When installing *FlowVision* on *Windows Vista* or newer versions of *Windows* with *User Account Control* (*UAC*) enabled, you have to keep the following rules:

- Installer and uninstaller must be run as Administrator.
- Do not store settings of any modules in directories, which are write-protected for users without Administrator's rights (**`Program Files`**, **`Windows`**, etc.).
- If *FlowVision* is installed into the directory **`Program Files`** or another protected system directory, then:
  - The file **`Fv.cfd`** and any other files in the installation directory can be edited only by a program (a text editor), which runs as Administrator.
  - **Configurator** can be used only when it runs as Administrator (because files, which are stored in system directories, can be changed only by a program, which runs as Administrator).

The correct operation of *FlowVision* modules is not guaranteed when these rules are not kept.

Fixing errors caused by violation of the rules of installing *FlowVision* on *Windows* with *UAC*

To fix errors caused by violation of the above rules do the following:

Step	Actions
1	Uninstall *FlowVision* (run as Administrator).
2	Delete a directory, where *FlowVision* was installed, using a program, which runs as Administrator.
3	Install *FlowVision*, starting the installer, which runs as Administrator.

4.8.10 Removing FlowVision

Installation of *FlowVision* does not assume deep integration with the operating system. Therefore, the removal of *FlowVision* consists of the following manual actions with of directories.

Do the following steps:

Step	Actions
1	Remove directories with the settings and log files of *FlowVision* modules (the path to these directories can be found using **Configurator** or view in the text file **`Fv.cfd`**, which locates in the root of the directory where *FlowVision* is installed).
2	Remove the directories where *FlowVision* is installed.

Attention!

Removing the configuration directory of server modules will cause loss of all *FlowVision* settings.

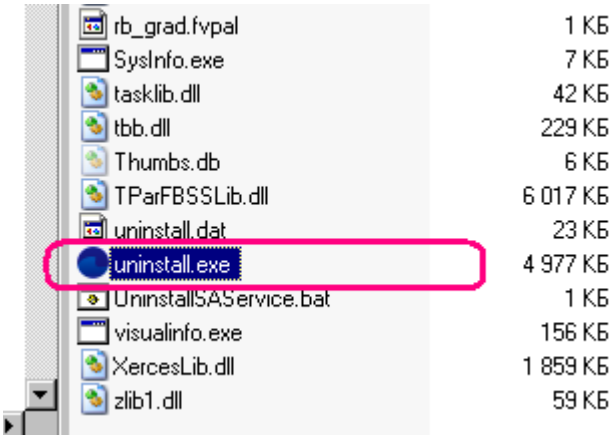
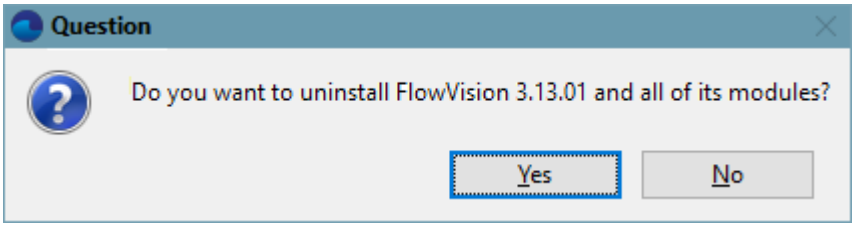
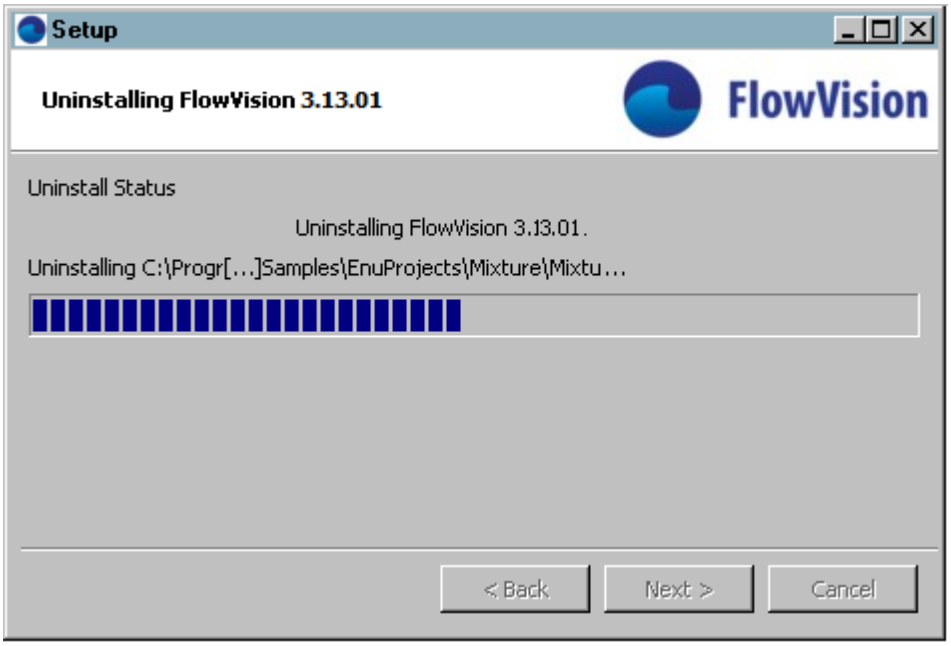
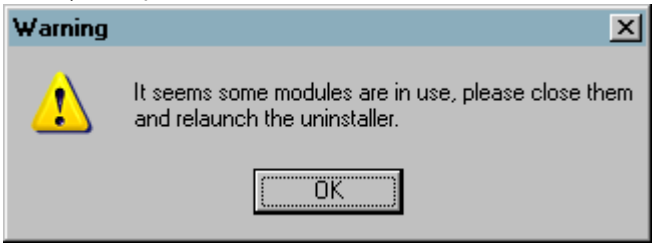
Removing the configuration directory of **License manager** *will cause loss of the license, which is registered on this computer*.

It is recommended to save the license and settings, see details in the section Saving and transferring the license and settings.

Use of the uninstaller program `uninstall.exe` (if *FlowVision* is installed on *Windows*)

To remove *FlowVision* (on *Windows*) you can also use the uninstaller program **`uninstall.exe`**, which is included with the distribution pack. Do the following steps:

Step	Actions
1	Find in the installation directory, where you have installed *FlowVision* (by default this is the directory **`C:\Program Files\FlowVision-3.13.01`**, digits indicate here the version number), the file **`uninstall.exe`**:

Step	Actions
	 Run this file.
2	A window opens prompting you to confirm your decision to remove *FlowVision*:  Confirm your decision to delete *FlowVision*, clicking on **Yes**.
3	The progress of removing *FlowVision* will be displayed in a dialog box:  ***Note:*** If a component of previous or current version of *FlowVision* is working and this prevents the uninstallation, a message ("**It seems some modules are in use, please close them and relaunch the uninstaller.**") will open:  If so, click **OK**, then close the working component(s) of *FlowVision* and start the uninstaller again.

Step	Actions
4	After *FlowVision* is removed, a message is displayed:  Click **OK**.

Removing the License Manager

Removing the **License Manager** is done similarly.

On *Windows*, it is recommended to remove **License Manager** using the uninstaller program `uninstall.exe` or the *Windows' Control Panel* **Add or Remove Programs** (**Start Windows** button **> Settings > Control Panel > Add or Remove Programs**).

Before use of the uninstaller, you also have to stop all running **License Manager**'s programs. Otherwise the uninstaller will not work and will display an error message ("**It seems some modules are in use, please close them and relaunch the installer**").

Removing *3DTransVidia* installed along with *FlowVision*

If the *3DTransVidia* program has been installed along with *FlowVision*, then removing *FlowVision* causes also removing *3DTransVidia*.

4.9 Administration

See sections:

- License management
- Using Solver-Agent
- Typical configurations
- Generating the diagnostics information

4.9.1 License management

A license must be received and registered to begin using *FlowVision*.

Principal terms used during the licensing procedure

- ***Registration information*** is the information provided by the user to the *FlowVision* support service. This information is required for issuing of a license.
- ***Registration file*** is a file with `rdb` extension containing the registration information.
- ***License*** is a permission granted to the user to use certain features of *FlowVision*.
- ***License file*** is a file with `lic` extension containing everything necessary for license registration. The license file is issued to the user by the *FlowVision* support service. It must be registered by the **License Manager**, following which the license can be used for running *FlowVision*.
- ***Information file*** is a file with `txt` extension issued to the user by the *FlowVision* support service. It contains information about the user and the validity dates of the license and technical support.
- ***License username*** is the name to which the license is issued (set when receiving the registration information stored in the information file). Several licenses can be registered to one license username.

License names can contain from 3 to 16 symbols and may contain only case-sensitive uppercase and lowercase Latin characters (`A`, `B`, ..., `Z`, `a`, `b`, ..., `z`), digits (`0`, `1`, ..., `9`) and the underscore symbol "_".

License password is a password enabling the user to make use of the license after registration of the license file by **License Manager** (stored in the information file). **License username** and **License password** are the credentials confirming the user's rights to the given license. They will be necessary later when registering new users on **Solver-Agent**.

License expiration term is the date until which the license is valid. It is not possible to open projects in **Pre-Postprocessor** or start computations after this date.

Technical support expiration term is the date until which technical support is provided. Technical support includes the ability to use software updates and contact *FlowVision* technical support. Software updates issued after the technical support expiration date cannot be used. To prolong the license or technical support, a new license must be received.

License management

License management is performed:

1. From Pre-Postprocessor using menu commands **File > Licenses**.
2. From Terminal using menu command **Licenses**.
3. When starting the **FvLicenseUtil** utility with various keys.

The **License Manager** (`FvLicense.exe`) must be running at the time.

Licensing procedure

- Receive registration information from the **License Manager** through **Pre-Postprocessor**, **Terminal** or the **FvLicenseUtil** utility.
- Send the registration information file (`*.rdb`) to your dedicated manager.
- Receive the license file and information file from your manager.
- Register the license file using **License Manager** in **Pre-Postprocessor**, **Terminal** or with the **FvLicenseUtil** utility *not later than the indicated date*.

Receiving information about status of your license

You can receive information from **License Manager** on the status your license during further operation. This information is displayed in the **License information** window. It is accessible only during the term of the license.

Support of hardware tokens

Version *309* of *FlowVision* does not support linking licenses to hardware tokens.

Work with *FlowVision* without a license

In the absence of a license, the following functionality of **Pre-Postprocessor** is available:

- **Pre-Postprocessor** operates in the view mode. It is possible to create projects, change and view them, but it is impossible to save or run a project.
- It is possible to register users of Solver-Agent without specifying a license. Such users will obtain reduced functionality, they will be able to load projects to **Solver**, connect to projects from **Pre-Postprocessor**, view results (including viewing history of saved data), create new visualization **Layers** and **Characteristics**, but these users cannot save changes and run computation of projects.

Operation of **Viewer** doesn't require a license. Users can view `fvvis`-files.

Fixing errors

In case of errors during license management:

- See the description of typical license management errors and try to fix them yourself.
- In case of license issues in **Pre-Postprocessor**, also see section Errors in Pre-Postprocessor because of licenses and try to fix them yourself.
- If the issues cannot be resolved, create a diagnostic information file using **Configurator** and send this file to *FlowVision* technical support with a description of the issues encountered.

4.9.1.1 Operations with licenses

After installing *FlowVision* on your computer, you should perform several operations:

- get the file in the `*.rdb` format with the registration information on the computer, where **License Manager** locates (this computer is the license server).
- send this registration file to the license service.
- after receiving from the license service a file in the `*.lic` format and a text file with your data (license name of the user, their password, period of validity of the license, and period of activation of the license), register the license.

You can also get information about the statuses of the existing licenses.

Receiving registration information (`.rdb` file)

To receive the registration information (in a `.rdb` file), use:

- in **Pre-Postprocessor** the menu command **File > Licenses > Get registration info** (see details in section Receiving registration information from Pre-Postprocessor)
- in **Terminal** the menu command **Licenses > Get registration info** (see details in section Getting registration information from Terminal)
- the utility `FvLicenseUtil` with the key `/R` (see details in section Getting registration information from FvLicenseUtil)

Registration of a license

To register a licenses based on the data from a `.lic` file received from the license service, use:

- in **Pre-Postprocessor** the menu command **File > Licenses > Register new license**(see details in section Registering a license in Pre-Postprocessor)
- in **Terminal** the menu command **Licenses > Register new license** (see details in section Registering a license in Terminal)
- the utility `FvLicenseUtil` with the key `/L` (see details in section Registering a license in FvLicenseUtil)

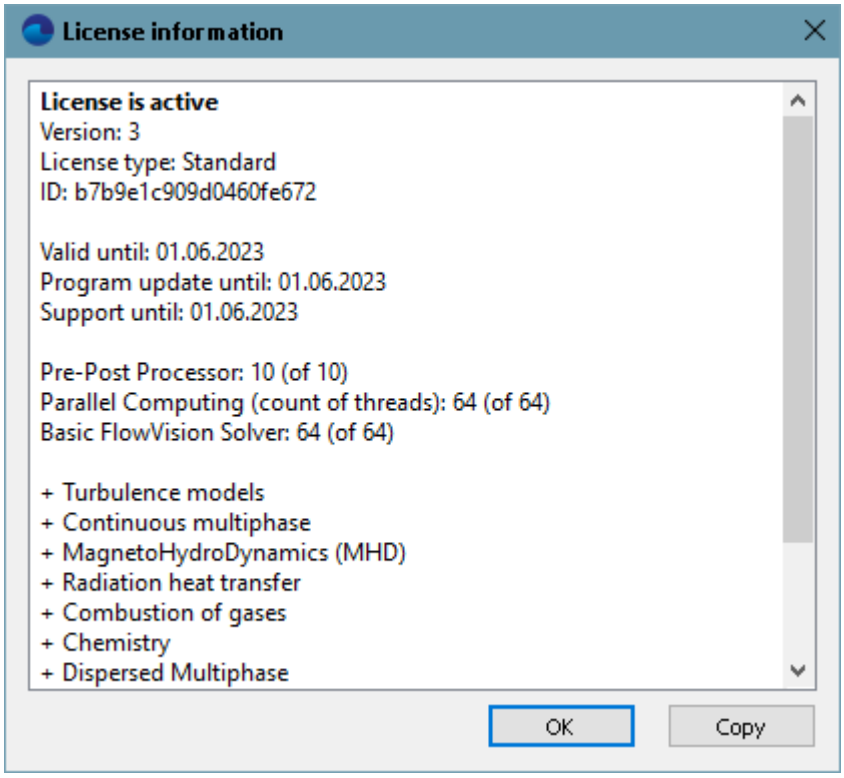
Getting information about statuses of existing licenses

To get information about existing licenses, use:

- in **Pre-Postprocessor** the menu command **File > Licenses> Get license info** (see details in section Getting license information from Pre-Postprocessor)

- in **Terminal** the menu command **Licenses > Get license info**(see details in section Getting license information from Terminal)
- the **`FvLicenseUtil`** utility with the key **`/I`** (see details in section Getting information about licenses from FvLicenseUtil)

4.9.1.2 License information



The **License information** dialog box (how to open it, see sections Getting license information from Pre-Postprocessor and Getting license information from Terminal) displays information about terms of licenses, license type, and about the number of licenses for specific modules, issued for the previously specified **License username**.

This dialog box contains the following lines:

Parameter or license option	Description
Version	Version of *FlowVision* (for example, **3**)
License type	Type of the license. Possible values are: • **Standard** for industrial use • **Academic** for academic use
ID	Identifier of the license
Valid until	The date, until which the license is valid. You will not be able to create new *FlowVision* projects and run *FlowVision* computations after this date.
Program update until	You cannot use (create projects and run computations) versions *FlowVision*, which are issued after this date.
Support until	The date, until which the technical support is provided.
Pre-Post Processor	The number of workplaces for **Pre-Postprocessor**. The number of unused licenses is shown in brackets.
Parallel Computing (count of threads)	The number of parallel threads. Generally, this value corresponds to the total number of cores, on which it is possible to run a **Solver**. If several **Solvers** run, they spend as many licenses, as the total number of cores that are occupied by all running **Solvers**. The number of unused licenses is shown in brackets.
Basic FlowVision Solver	The number of computations that you can run simultaneously. The number of unused licenses is shown in brackets.
Turbulence models	Ability to simulate turbulence

Parameter or license option	Description
Continuous multiphase	Ability to simulate multiphase flows with two continuous phases in one model, when the phase interface surface is simulated using the VOF method, see Two-phase medium models.
Moving Bodies	Ability to simulate **Moving bodies** with enabled update (**Update > Type ≠ Disabled**)
Gap (Clearance) Modeling	Ability to use the gap model
Cosimulation with FEA software	Ability to connect to *Abaqus* or other finite element software
Preparing of Geometry models	Ability to import geometry in the format **`mesh`** (the internal format). This option enables import of *3DTransVidia* files (their file extensions are **`.mesh`**) into *FlowVision*.*)
SIMULIA Abaqus format import	Ability to import geometry in the format **`inp`** (format of *Abaqus*). This option enables import of *Abaqus* files (their file extensions are **`.inp`**) into *FlowVision* and export load distribution on nodes from *FlowVision*.*)
Nastran format import	Ability to import geometry in the *Nastran data file* format. This option enables import of *NASTRAN* files (their file extensions are **`.bdf`**, **`.dat`** or **`.nas`**) into *FlowVision* and export load distribution on nodes from *FlowVision*.*)
ANSYS format import	Ability to import geometry in the format **`cdb`** (format of *Ansys*). This option enables import of *ANSYS* files (the file extension is **`.cdb`**) into *FlowVision* and export load distribution on nodes from *FlowVision*.*)
Radiation heat transfer	Ability to simulate radiation
Combustion of gases	Ability to simulate combustion of gases
Sliding mesh	Ability to simulate sliding surfaces
MagnetoHydroDynamics (MHD)	Ability to simulate electromagnetic hydrodynamics (EHD)
Dispersed Multiphase	Ability to use dispersed medium models
Import .mesh, .stp, .igs geom.models	Ability to use the *3DTransVidia* software with set of its license options **eBaseMeshRead**, **eBaseMeshNativeRead**, **eBaseIgsRead**, **eBaseStepRead** and **eBaseMeshNativeWrite**.
Import extension	Ability to use the *3DTransVidia* software with additional data formats.
Subregion Conjugation	Ability to use conjugation of all variables or conjugation of temperature.
Chemistry	Ability to simulate chemistry
Ablation	Ability to simulate ablation
Dispersed phase crystallization	Ability to simulate dispersed phase crystallization (icing)
Cavitation models	Ability to simulate cavitation

*) If the geometry model is already imported in project, you do not need this license option for opening this project. But if you use geometry replacing during the simulation (FSI simulation or batch script with replacing geometry, or *IOSO* optimization), you will need this license.

To copy the text from the **License information** dialog box into the *Windows* clipboard, select a fragment of the text or the whole text in the box and press **Ctrl+C** on your keyboard or click the **Copy** screen button (in the last case the whole text will be copied).

4.9.1.3 Saving and transferring the license and settings

We strongly recommend periodically backup the *FlowVision* settings. We also recommend backing up settings before making any technical or preventive actions on the hardware/software hosting the *FlowVision* modules.

This will help to reset the settings and license in case of damage to the hard disk or files. Also, backing up settings makes it possible to reinstall the operating system with the license preserved.

To back up the settings and active license, copy folder FlowVision3xxxx, which is located in the user directory of the server modules, to safe storage. To find the location of that folder, use the **Configurator** module or file Fv.cfd.

Restoring the saved settings and license

To restore your settings and license, after installing *FlowVision* using **Configurator** or directly in file Fv.cfd, indicate the path to the new user directory of the server modules when installing FlowVision modules and place in this directory a subdirectory with name **`FlowVision3xxxx`**. Replace subdirectory **`FlowVision3xxxx`**, replacing all the files in it.

> ***Note:***
>
> The user directory for the **License Manager** can be modified after installation only in file Fv.cfd, located in the same directory as the executable file of **License Manager** (file **`FvLicense.exe`**).
>
> If settings are transferred to a new version of *FlowVision*, keep in mind that the name of the **`FlowVision3xxxx`** subfolder depends on the version, so it will need to be renamed for *FlowVision* be able to find it in the user subdirectory of the server modules.

4.9.1.4 Transferring a license from version 3.08.xx to version 3.09.xx

Starting with *FlowVision* version *3.09.01* the **License Manager** is available as a separate application with its own installer. This allows you to quickly deploy the **License Manager** on a separate computer. This approach also reduces the risks and costs of the service, because the **License Manager** can be upgraded much less frequently than the other modules of *FlowVision*.

To transfer a license from the **License Manager** *3.08.xx* to **License Manager** *3.09.xx* carry out the following steps:

Step	Actions
1	Manually back up settings of *FlowVision* version *3.08.xx*, under which the **License Manager** works. a. Using **Configurator** create a diagnostic archive, which will contain the settings and license. Start the **Configurator** as an Administrator user. b. *Either* manually copy to a safe place the directory *FlowVision*, which contains the settings and license. The location of this directory is specified in the text file **`Fv.cfd`**, which locates in the directory with the installed *FlowVision*.
2	There can NOT work several **License Manager** on a single computer. Therefore, you have to remove the **License Manager** *3.08.xx*: a. do any of the following actions: • remove **License Manager** using the **Change** option of the installer of the **License Manager** 3.08.xx: • *or* completely remove *FlowVision 3.08.xx*. • *or* remove the system service of *Windows*, which starts the **License Manager** b. In *Linux* you can simply close the application and delete the executable file **`FvLicense`**.
3	Install the **License Manager** version *3.09.02* or newer version (since **License Manager** version *3.09.02* **Solvers** do not support the **License Manager** *3.09.01*). During the installation you have to specify the directory where the settings will be stored. In this directory after the installation the **`FlowVisionLM`** directory will be created.
4	In the directory **`FlowVisionLM`** (mentioned in Step 3) you must copy the **`FvLicense.dat`** file from the *FlowVision* directory or from the diagnostic archive (see Step 1). Also in the **`FlowVisionLM`** directory there will be the **`settings`** directory, where you must place the **`FvLicense.cfg`** file from the diagnostic archive (see Step 1).

Step	Actions
5	If the **License Manager** is installed as a system service, restart your computer or start the service **FvLicenseSrv**. Or just start the **License Manager**.
6	To check the correctness of the **License Manager**'s operation, request the license information using **Pre-Postprocessor**, **Terminal**, or the **FvLicenseUtil** utility.

If you encounter difficulties or questions during this procedure, please contact our technical support service.

4.9.1.5 Operations with licenses in Pre-Postprocessor

In **Pre-Postprocessor** licenses are managed by using the following commands in the main menu:

- **File > Licenses >** Get registration info
- **File > Licenses >** Register new license
- **File > Licenses >** Get license info

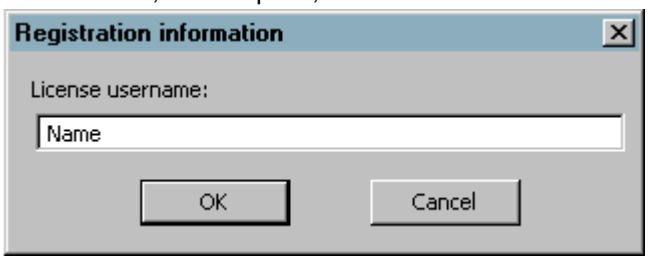
Before you begin managing the licenses in **Pre-Postprocessor**, you have to configure the connection with License Manager. Open the configuration window by the menu command **File > Preferences** and then specify the following settings:

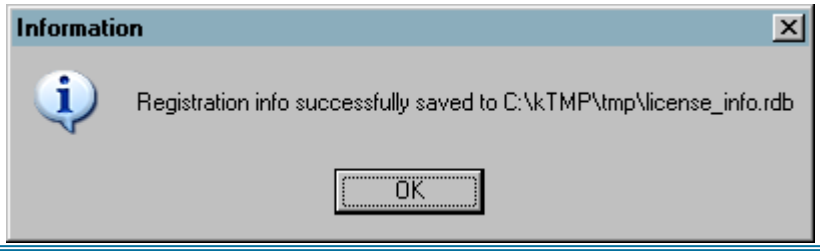
Setting	Description
License manager > Address	IP address or host name of the computer on which **License Manager** is running. If **License Manager** is not available for direct connection, then specify an IP address or host name of the computer, on which **Retranslator** is running.
License manager > Port	The port for connection between **License Manager** and licensed components. If **License Manager** is not available for direct connection, then specify a port for connection with **Retranslator**.
License manager > License username	The license username, see section License management.

When you start **Pre-Postprocessor** at the first time, the **License username** field is empty. If the license has been obtained, it is necessary to set the license name once in **Preferences**, after that it is remembered by the program and is used automatically at next starts. If receiving the registration information or registering a new license is done in **Pre-Postprocessor**, then the license username will be remembered automatically.

4.9.1.5.1 Receiving registration information from Pre-Postprocessor

Do the following steps:

Step	Actions
1	Make sure that **License Manager** is running.
2	Use the menu command **File > Licenses > Get registration info**.
3	In the form **Registration information**, which opens, enter the **License username**:  After clicking on **OK** the license information will be requested from **License Manager**.
4	Save the registration information in the registration (`.rdb`) file, specifying it in a standard operation system's window for access to files.

Step	Actions
	After a successful saving the information in the registration (**.rdb**) file, a message about the completed operation (**Registration info successfully saved to** ...) will be displayed: 
5	Click **OK**.

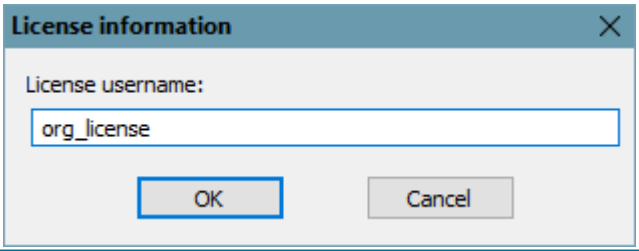
4.9.1.5.2 Registering a license in Pre-Postprocessor

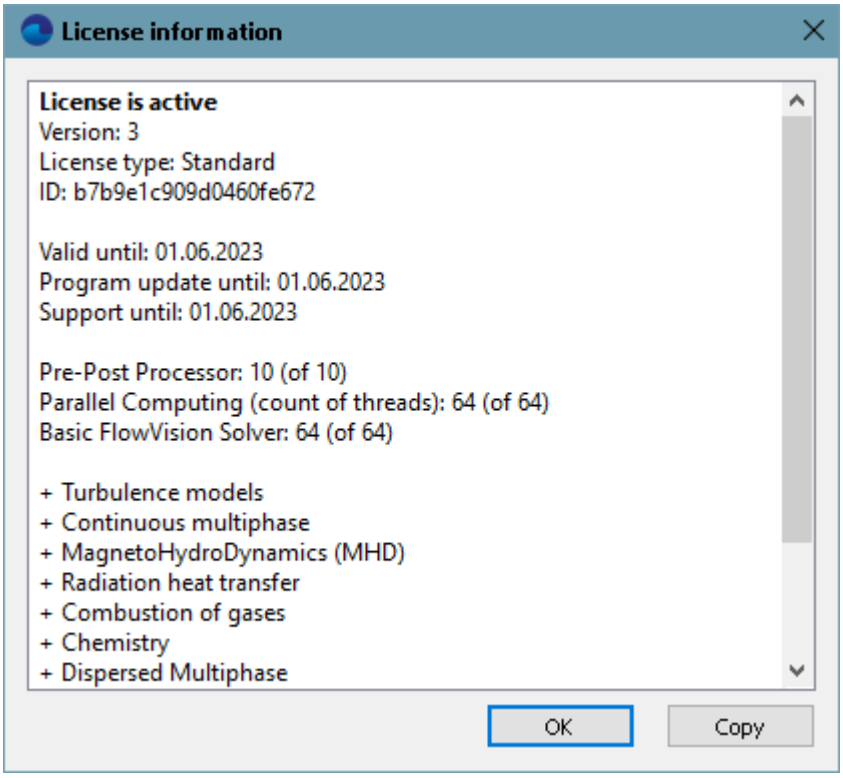
Do the following steps:

Step	Actions
1	Make sure that **License Manager** is running.
2	Apply the menu command **File > Licenses > Register new license**.
3	In operation system's window for opening files, which opens, select license file. After a successful registration of the license, a message about this will be displayed.
4	If an error occurs, write down the error code (it will be displayed in the error message) and contact the Technical support service.

4.9.1.5.3 Getting license information from Pre-Postprocessor

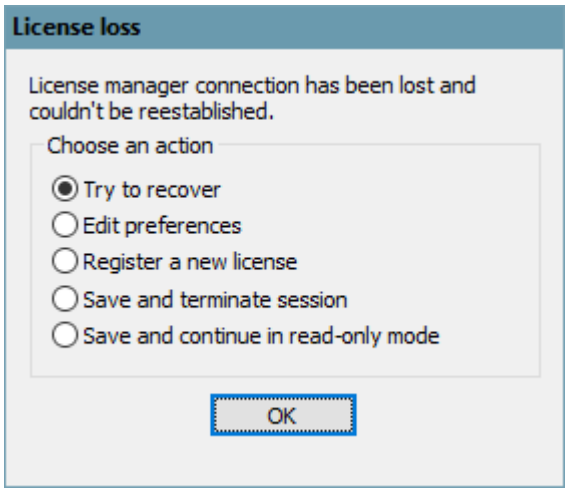
Follow the steps:

Step	Actions
1	Make sure that **License Manager** is running.
2	Apply the menu command **File > Licenses > Get license info**.
3	In the **License information** dialog box, which opens, define **License username** for which information is requested. 
4	After entering the license name and clicking on **OK** the **License information** dialog box opens:

Step	Actions
	 To copy the text from the **License information** dialog box into the *Windows* clipboard, select a fragment of the text or the whole text in the box and press **Ctrl+C** on your keyboard or click the **Copy** screen button (in the last case the whole text will be copied).

4.9.1.5.4 Errors in Pre-Postprocessor because of licenses

Error of connection to License Manager

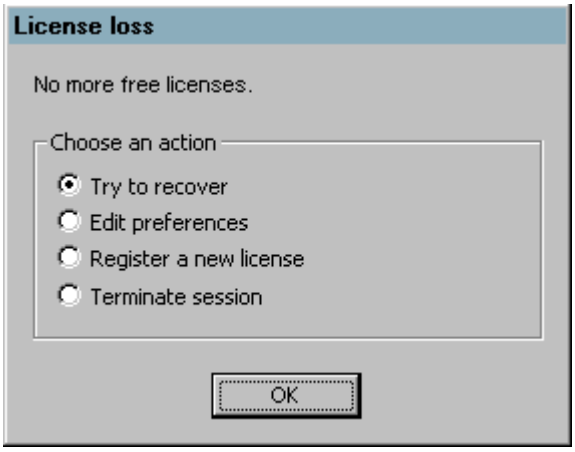


This error message means that *connection with **License Manager** is lost*.

When this message appears, try to do the following:

- Check whether **License Manager** is running and, if no, restart it and then reconnect with it (use the **Try to recover** option)
- Check the network connection to the computer on which **License Manager** is installed, and then, after the network connection restores, reconnect with **License Manager** (use the **Try to recover** option).
- If necessary, set up a connection to another **License Manager** (select the option **Edit preferences**).

Absence of available licenses



This error message means that *there are no free licenses*.

When this message appears, try to do the following:

- check the period of validity of your license
- check the number of currently available licenses for **Pre-Postprocessor** (see section Getting license information from Pre-Postprocessor)

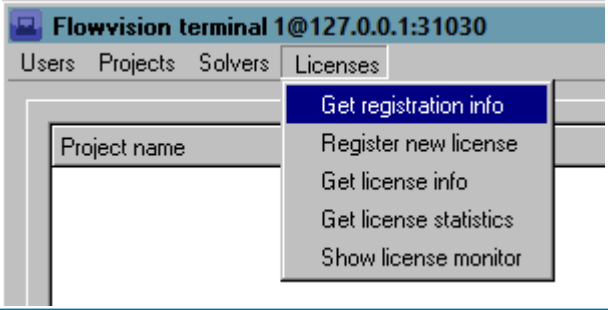
4.9.1.6 Operations with licenses in Terminal

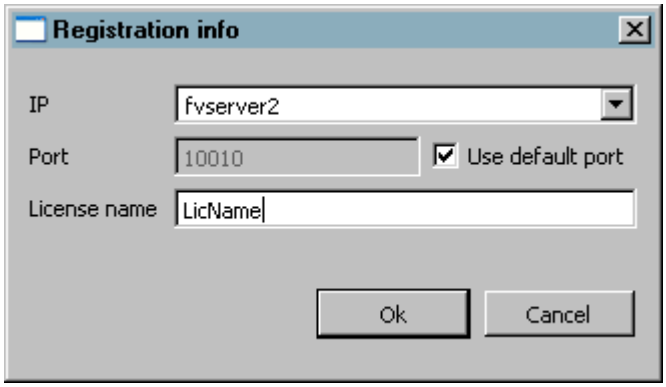
In **Terminal** licenses are managed by the following commands from the menu **Licenses**:

- **Get registration info**
- **Register new license**
- **Get license info**
- **Get license statistics**
- **Show license monitor**

4.9.1.6.1 Getting registration information from Terminal

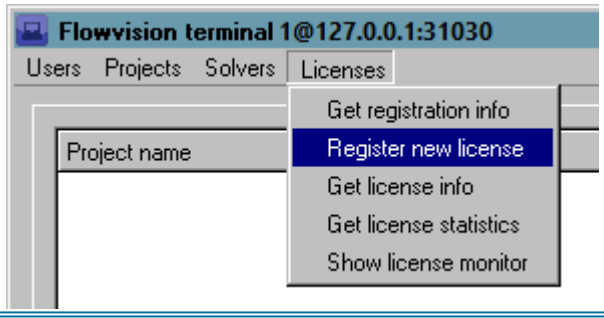
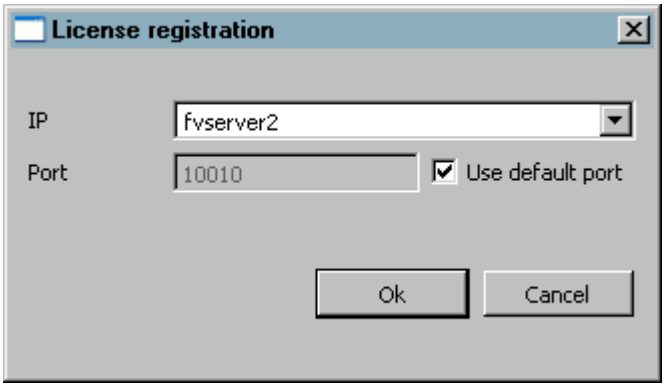
Do the following steps:

Step	Actions
1	Make sure that **License Manager** is running.
2	Use the menu command **Licenses** > **Get registration info**: 
3	The **Registration info** window opens:

Step	Actions
	 Specify there: • an IP-address or host name of the computer, on which the appropriate **License Manager** is installed • a port for connection to **License Manager** • the license name The entered values are remembered. After clicking on **OK** the license information will be requested from **License Manager**.
4	This opens a standard operating system's window for access to files. Specify there a registration (`.rdb`) file where the registration information will be saved.

4.9.1.6.2 Registering a license in Terminal

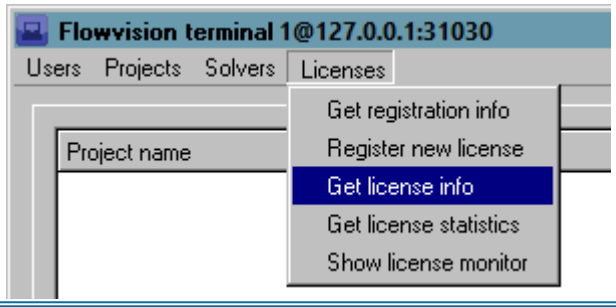
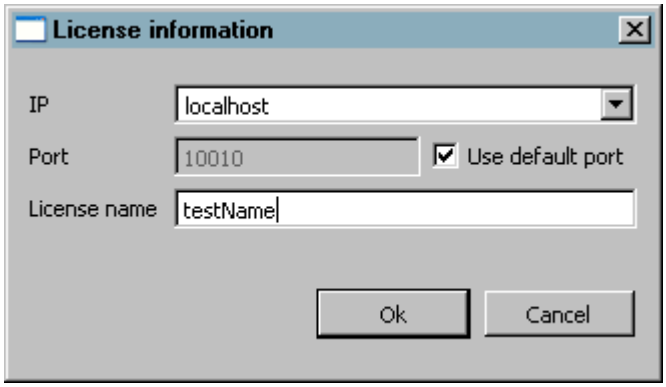
Do the following steps:

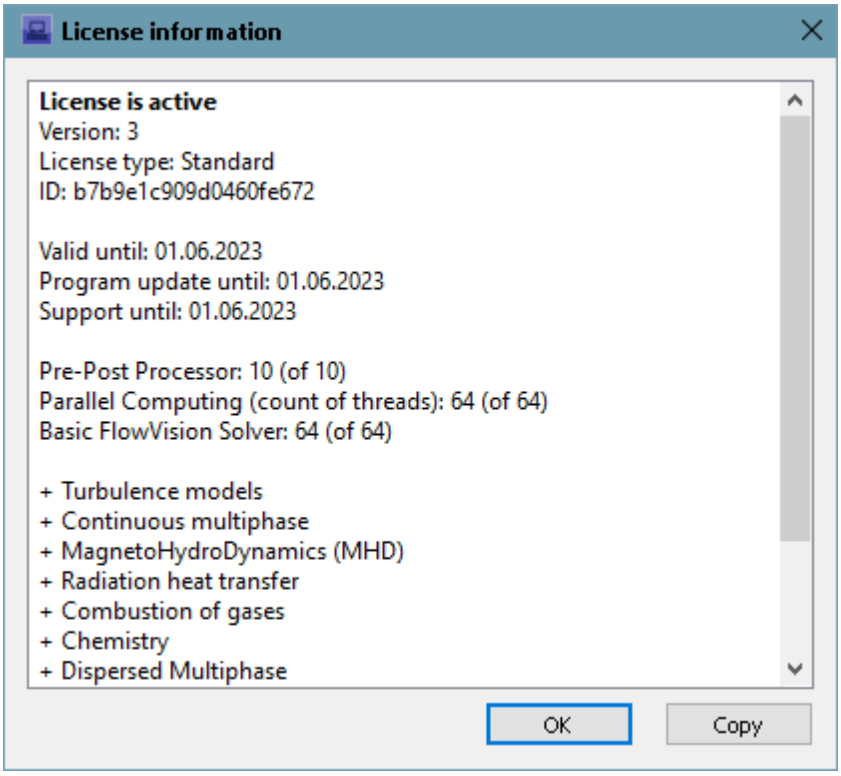
Step	Actions
1	Make sure that **License Manager** is running.
2	Use the menu command **Licenses** > **Register new license**: 
3	The **License registration** window opens:  Specify there: • an IP-address or host name of the computer, on which **License Manager** is installed • a port used for connection to **License Manager**

Step	Actions
4	In the window, which opens, specify the *license file*. After successful registration you will receive a message that the license has been successfully registered.
5	If an error occurs, write down the error code (it will be displayed in the error message) and contact the Technical support service.

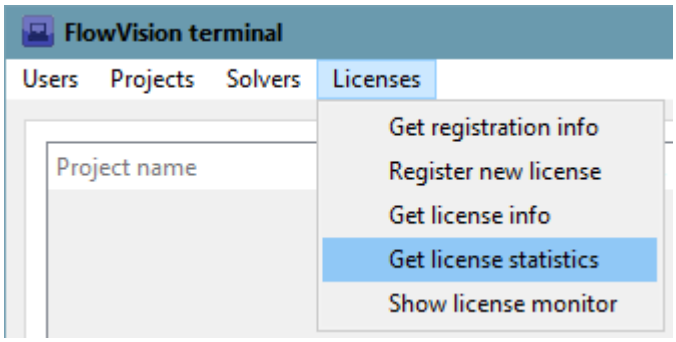
4.9.1.6.3 Getting license information from Terminal

Do the following steps:

Step	Actions
1	Make sure that **License Manager** is running.
2	Use the menu command **Licenses** > **Get license info**: 
3	The **License information** window opens:  Specify there: • an IP-address or host name of the computer, on which **License Manager** is installed • a port for connection to **License Manager** • the license name for which the information is requested Click **Ok**.
4	The License information dialog box opens:

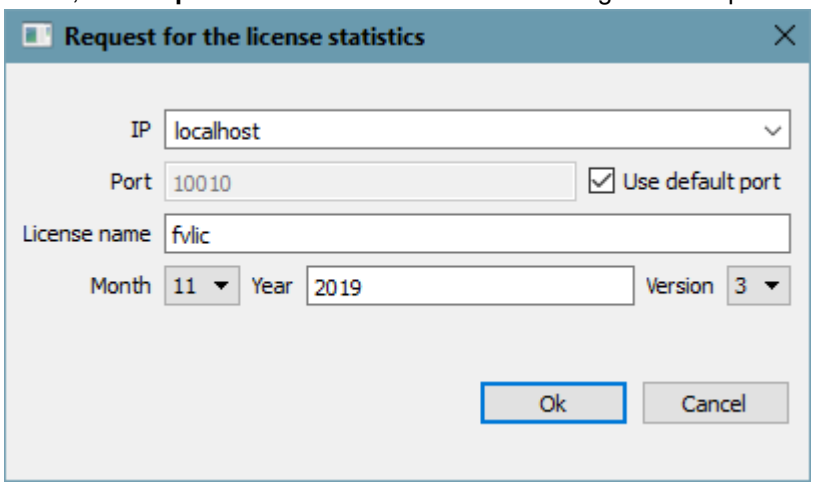
Step	Actions
	 To copy the text from the **License information** dialog box into the *Windows* clipboard, select a fragment of the text or the whole text in the box and press **Ctrl+C** on your keyboard or click the **Copy** screen button (in the last case the whole text will be copied).

4.9.1.6.4 Getting license statistics from Terminal (information for developers only)



Using the menu command **Licenses** > **Get license statistics**, it is possible to get statistics about use of licenses.

After applying the command, the **Request for the license statistics** dialog box will open:



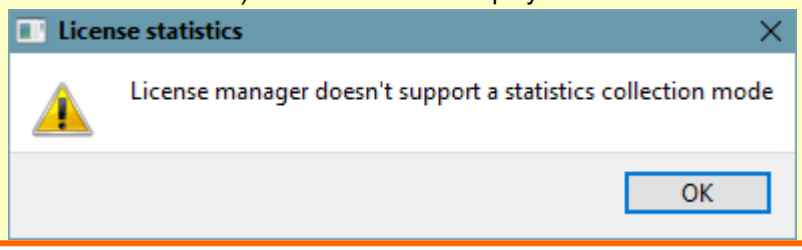
Specify the required information in this dialog box:

Interface element	Description
IP	An IP-address or host name of the computer, on which **License Manager** is installed (see Initial Configuration).
Port	A port for connection to **License Manager** (see Initial Configuration).
Use default port	Select this checkbox when **License Manager** uses the default port.
License name	Username for which the license is granted (see Initial Configuration).
Month	Month and year for which the statistics is gathered
Year	
Version	Version of *FlowVision*

Then click **Ok**.

Gathering of statistics is enabled when configuration file of the License Manager (FvLicense.cfg) includes **StatSave=Yes**.

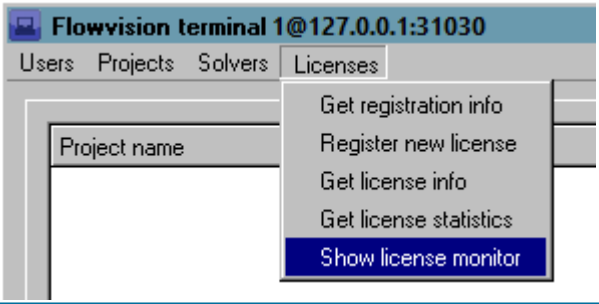
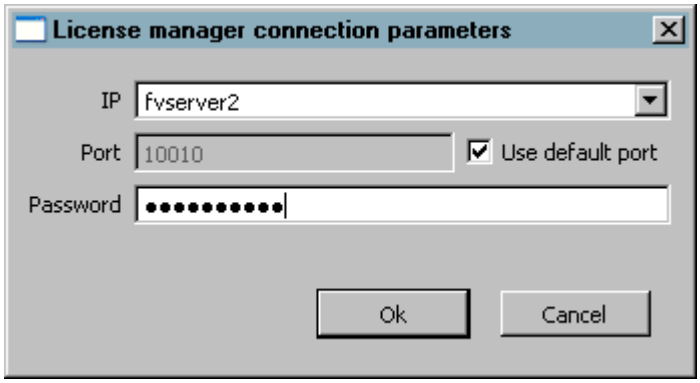
If gathering of statistics is disabled (**StatSave=No**), then a message (**License manager doesn't support a statistics collection mode**) about this will be displayed:

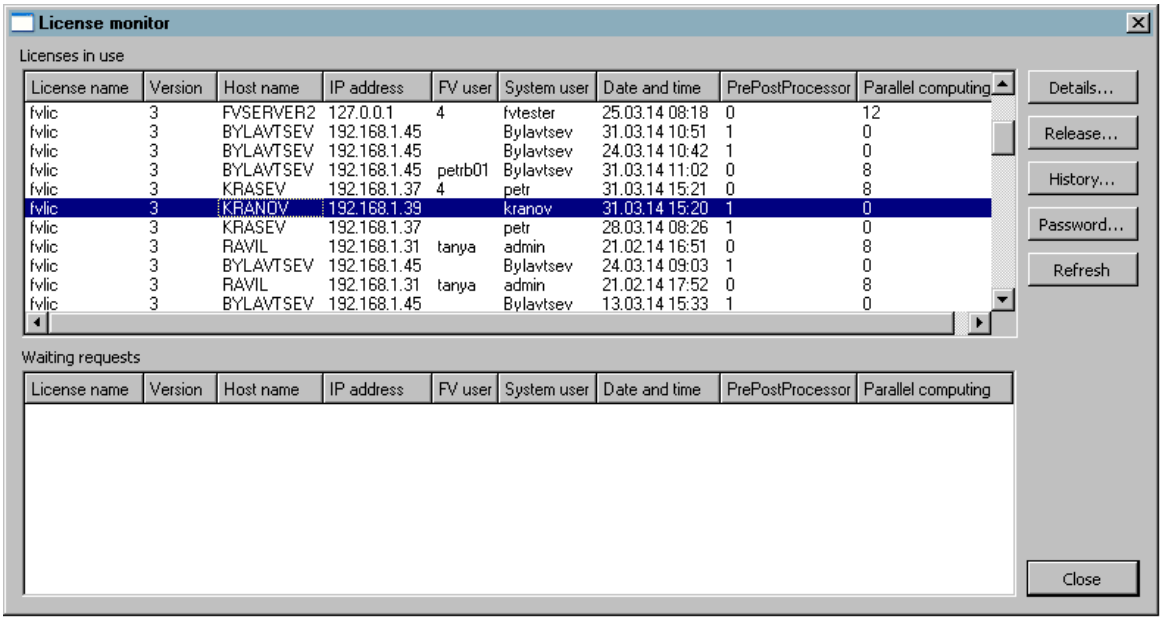


4.9.1.6.5 License monitor

License monitor is used to monitor the captured licenses, and for viewing the history of the use of licenses.

To access **License monitor**, do the following steps:

Step	Actions
1	Make sure that **License Manager** is running.
2	Use the menu command **Licenses** > **Show license monitor**: 
3	The **License manager connection parameters** window opens:  Specify there:

Step	Actions
	• an IP-address or host name of the computer, on which **License Manager** is installed • a port for communication with **License Manager** • the password for access to **License monitor** [1)] Click **Ok**.
4	Information about the captured licenses is displayed in the **License monitor** window. The used licenses are displayed in the pane **Licenses in use** and requests for licenses are displayed in the pane **Waiting requests**.  The queue of waiting requests can only be formed if a non-zero **`LicenseTimeout`** parameter is specified in the configuration file of Solver (FvSolver.cfg).

Both panes (**Licenses in use** and **Waiting requests**) in the **License monitor** window have the same columns:

Column	Description
License name	License name, on which the license is captured
Version	The version number of *FlowVision*
Host name	Name of the computer from which the license has been captured
IP address	IP address of the computer from which the license has been captured
FV user	The username for **Solver-Agent** under which the license has been captured[2)]
System user	The username for operating system as which a *FlowVision*'s module, which has captured the license (**Pre-Postprocessor** or **Solver**), is running
Date and time	The date and time of when the license was captured
PrePostProcessor	The number of licenses captured on **Pre-Postprocessor**
Parallel computing	The number of processor cores on which **Solver** is running

The **License monitor** window has the following buttons:

Button	Description
Details[3)]	This button opens the **License capturing details** window, which displays details about the selected captured license.
Release[3)]	This button releases the selected captured license.
History	This button allows viewing the history of the use of licenses.
Password	This button allows changing the password for access to **License monitor**[1)]
Refresh	This button refreshes the list of captured licenses.

Button	Description
Close	This button closes the **License monitor**.

Notes:

[1] By default, the password for access to **License monitor** is `FlowVision`. If necessary, you can change it.

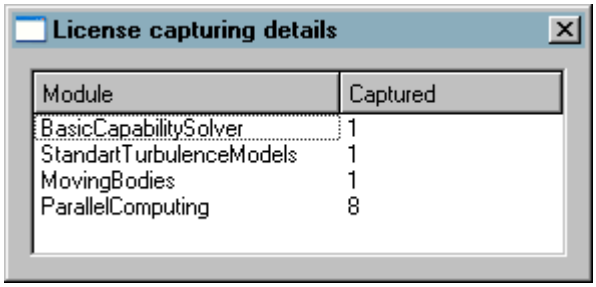
[2] This is only displayed when a project is loaded to the **Solver**.

[3] This is only available if some captured license is selected in the **License monitor** window.

4.9.1.6.5.1 License capturing details

The **License capturing details** window opens by clicking on the button **Details** in License monitor.

It contains the list of modules and functionality codes, for which the licenses are captured, and the number of captured licenses.



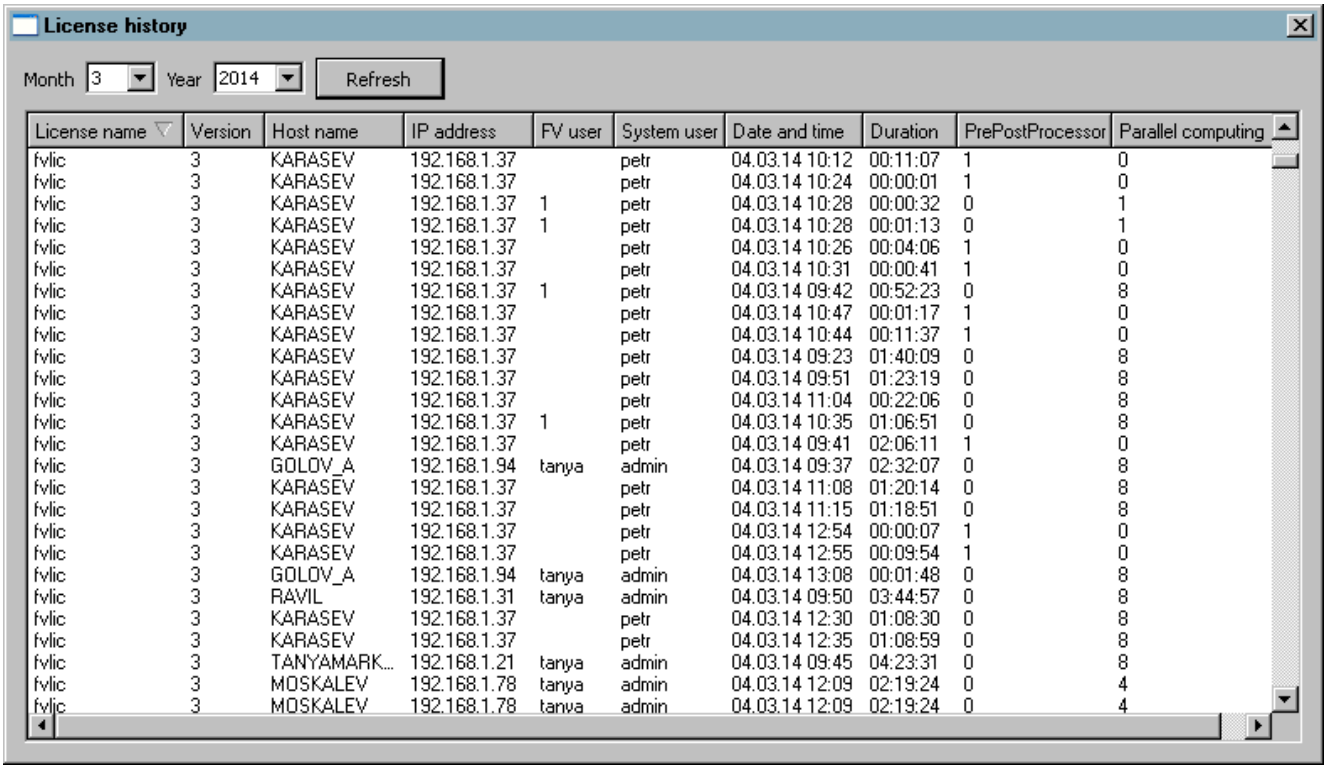
The **License capturing details** window has the following columns:

Column	Description
Module	Name of the module or functionality code, for which the licenses are captured. See details in section License information.
Captured	The number of licenses, captured by this module or functionality code.

4.9.1.6.5.2 License history

The **License history** window opens when you click the **History** button in License monitor.

The **License history** window displays information about the history of use of licenses.



The window contains the following elements at its top:

Element	Description
Month (drop down list)	Month and year for which you wish to view the history of use of licenses
Year (drop down list)	
Refresh (button)	This button updates the displayed data according to the specified **Month** and **Year**.

Information about the history of use of licenses is displayed in the following columns:

Column	Description
License name	License username that captured the license
Version	The version number of *FlowVision*
Host name	Name of the computer from which the license has been captured
IP address	IP address of the computer from which the license has been captured
FV user	The username for **Solver-Agent** under which the license has been captured (it is only displayed when a project is loaded on **Solver**)
System user	The username for operating system as which a *FlowVision*'s module, which has captured the license (**Pre-Postprocessor** or **Solver**), is running
Date and time	The date and time of when the license was captured
Duration	Period for which the license has been captured
PrePostProcessor	The number of licenses captured on **Pre-Postprocessor**
Parallel computing	The number of processor cores on which **Solver** is running

4.9.1.7 Operations with licenses using FvLicenseUtil

Operations with licenses can be also done using the **FvLicenseUtil** utility, which can be run with keys:
- `/R` provides registration information
- `/L` does registering a license
- `/I` provides information about licenses

The **FvLicenseUtil** utility for *Linux* locates in the directory with installed License Manager (`/home/user/FlowVisionLM` where `user` corresponds to the user's system name).

The **FvLicenseUtil** utility for *Linux* locates in the folder with installed License Manager (by default it is `C:\Program Files\FlowVisionLM`) and also in the folder where main *FlowVision* modules are installed (by default it is `C:\Program Files\FlowVision-N.NN.NN` where "N.NN.NN" corresponds to the version of *FlowVision*).

4.9.1.7.1 Getting registration information from FvLicenseUtil

Do the following steps:

Step	Actions
1	Make sure that **License Manager** is running.
2	Run from the command line the `FvLicenseUtil` utility with the following syntax: `FvLicenseUtil /R IP_ADDRESS PORT LICENSE_NAME FILE_NAME`

`IP_ADDRESS`	IP address of the computer, on which the appropriate **License Manager** is installed
`PORT`	The port, which is used for connection to **License Manager**
`LICENSE_NAME`	The license username
`FILE_NAME`	The name of the registration (`.rdb`) file, where the registration information will be saved

Step	Actions
	After executing the command line, a registration (`.rdb`) file will be created having the specified name.

4.9.1.7.2 Registering a license in FvLicenseUtil

Do the following steps:

Step	Actions
1	Make sure that **License Manager** is running.
2	Run from the command line the `FvLicenseUtil` utility with the following syntax: `FvLicenseUtil /L IP_ADDRESS PORT FILE_NAME` `IP_ADDRESS` — IP address of the computer, on which the appropriate **License Manager** is installed `PORT` — The port, which is used for connection to **License Manager** `FILE_NAME` — Name of the license file (`.lic`) If an error occurs, write down the error code (it will be displayed in the error message) and contact the Technical support service.

4.9.1.7.3 Getting information about licenses from FvLicenseUtil

Do the following steps:

Step	Actions
1	Make sure that **License Manager** is running.
2	Run from the command line the `FvLicenseUtil` utility with the following syntax: `FvLicenseUtil /I IP_ADDRESS PORT LICENSE_NAME` `IP_ADDRESS` — IP address of the computer, on which the appropriate **License Manager** is installed `PORT` — The port, which is used for connection to **License Manager** `LICENSE_NAME` — The license username As a result of the execution of the command line, in the console window information about the licenses issued for the specified license name will be displayed.

4.9.1.8 Licenses with per-minute charge

Since version *FlowVision 3.11.01* it is possible to purchase licenses with per-minute charge.

How this works

The user purchases some fixed prepaid amount of node*minutes.

After purchasing the license the user receives an notification with parameters of the license:

- IP-address for connection to a web-based **License Manager**
- port for connection to the **License Manager**
- license username
- license password

When the computation starts, countdown of available node*minutes begins. Spending the node*minutes suspends when the computation finishes.

The **Solver**, on which the project has been loaded but which is not calculating, do not request a license. So node*minutes are being spent only when the computation is in progress.

License on **Pre-Postprocessor** is included and is not paid individually. Duration of use of Pre-Postprocessor is not paid, i.e. use of **Pre-Postprocessor** doesn't cause spending of node*minutes.

If available node*minutes finish during the computation, **Solver** will inform that it has no enough license options (an appropriate line will be recorded in the `err`-file in the projects folder) and then **Solver** will stop the computation.

You can run several **Solvers** simultaneously.

How to purchase

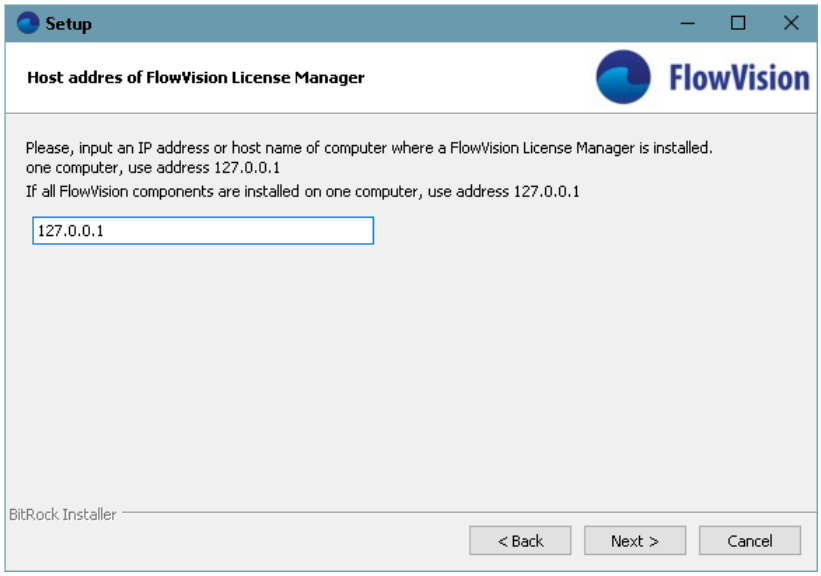
To find out the cost of licenses with per-minute charge, contact your regional sales managers.

Connecting *FlowVision* to a license with per-minute charge

Setting up during installation of *FlowVision*

When you use a license with per-minute charge, you don't have to install **License Manager** because a web-based **License Manager** will be used.

During the main installation of *FlowVision*, you will see a dialog box with a question about location of **License Manager**:



Here you have to enter the IP-address, that you received after purchasing the license with per-minute charge.

After finishing the installation you can check the status of the license to make sure that *FlowVision* is ready to use this license.

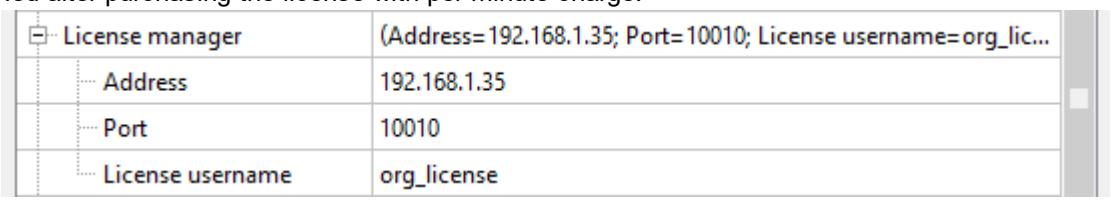
Connecting an installed *FlowVision* to a license with per-minute charge

If *FlowVision* is already installed and tuned for operation with a local license, it can be tuned to using a license with per-minute charge.

To do so, you have to change settings of two modules, **Pre-Postprocessor** and **Solver**, and also set or change parameters of the user of **Solver-Agent**.

Setting Pre-Postprocessor

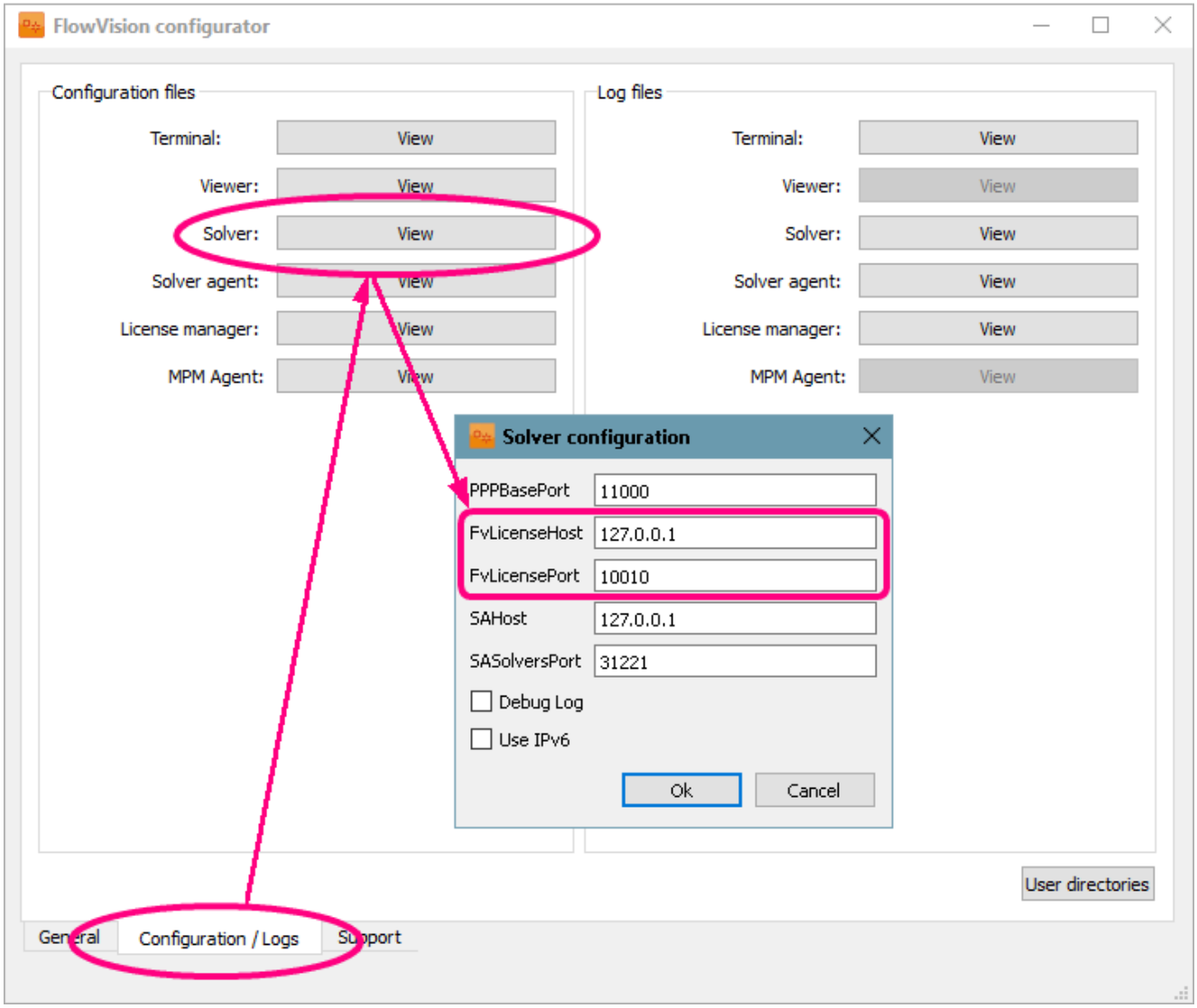
From the main menu of Pre-Postprocessor select the command **File > Preferences** and in the dialog box of the base settings, in the "License manager" group of settings, specify **Address**, **Port**, and **License username**, which you received after purchasing the license with per-minute charge:



Setting Solver

To set **Solver**, run **Configurator**.

In the **Configuration/Logs** tab, in the section **Configuration files**, click the **View** button next to the **Solver:** line. In the **Solver configuration** dialog box, which opens, specify the **Address** and **Port** for connection to **License Manager**, which you received after purchasing the license with per-minute charge.



Note that you don't enter the license name in settings of **Solver**, because this name is specified in parameters of the user of **Solver-Agent**. So the next step will be editing of these parameters or creation of another user of **Solver-Agent**.

When **Solver** is installed in the system with no graphical user interface, you can edit its configuration file. See Configuration file of Solver (FvSolver.cfg).

Creating a new user of Solver-Agent or changing parameters of an existing user

Editing parameters of a user of **Solver-Agent** is possible either from **Terminal** or from **Pre-Postprocessor**, see Registration data (profile) of Solver-Agent's user and their change.

You have to specify the **License username** and **License password** that you received after purchasing the license with per-minute charge.

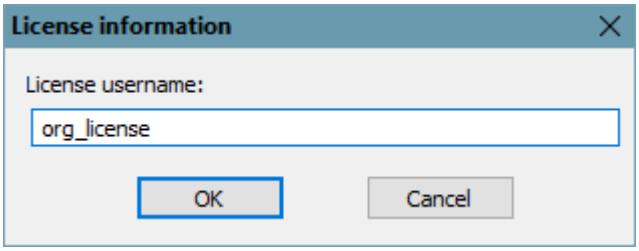
Also you can registering a new user of Solver-Agent, if *FlowVision* is newly installed or you wish to use *FlowVision* with several different licenses.

Checking status of a license

To check correctness of connection to a license or examine how many unspent node*minutes are available, you can use standard features for viewing license information from **Pre-Postprocessor** or **Terminal**.

In **Pre-Postprocessor** open the **File > Licenses > Get license info** menu item.

In the **License information** dialog box, which opens, enter the license username that you received after purchasing the license with per-minute charge:



After clicking **OK** the License information dialog box opens. Amount of unspent node*minutes and the total amount of cores that are being used by **Solvers** at the current time moment are displayed in the `Parallel Computing (count of threads)` line.

Example: `Parallel Computing (count of threads): 999618336 core/minutes left, 12 cores are using`

In **Terminal** information about the status of the license is available from its "Licenses" menu.

4.9.1.9 License management errors

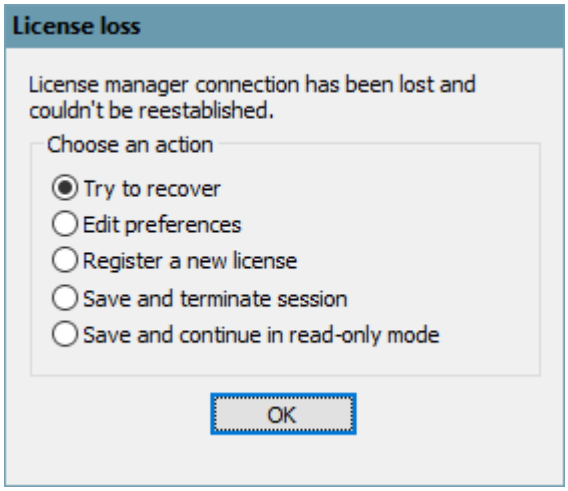
Error	Possible causes
Errors of getting the registration information:	
Unable to connect to License Manager (**Terminal**'s message is: `Connection to ... (port ...) refused.`)	Possible causes of this error: • **License Manager** is not running. • Incorrectly specified address of the computer, on which the License Manager is running. • Ports has been configured incorrectly.
Errors of registration of the license:	
Unable to connect to License Manager (**Terminal**'s message is: `Connection to ... (port ...) refused.`)	Possible causes of this error: • **License Manager** is not running. • Incorrectly specified address of the computer, on which the License Manager is running. • Ports has been configured incorrectly.
Wrong Local ID (**Terminal**'s message is: `License registration error (code 4). Please, contact to support.`)	Possible causes of this error: • The local ID is used, which has been obtained on another computer. • The hardware was updated after obtaining the local ID. • The USB key has not been inserted into a connector (if the license is linked to the USB key). • The USB key does not operate (if the license is linked to the USB key).
Time for registration of the license has been expired (**Terminal**'s message is: `License registration error (code 5). Please, contact to support.`)	There was an attempt to register the license after the date, when the registration period expired (the time period, which was specified in the text file that you have received from the service service along with the license key, expired).
Incorrect version of the license file (**Terminal**'s message is: `License registration error (code 8). Please, contact to support.`)	Incorrect version of the lic-file, which is used to register the license. This error appear if the license has been generated by an old or, alternatively, a very new version of the license generator.
A license of another type has been found for this license name (**Terminal**'s message is: `License registration error (code 13). Please, contact to support.`)	A license of another type has been found for this license name.
Write error of the file with the license information	Write error of the file `FvLicense.dat`. This might occur, for example, when the user directory, specified in the configuration, is not available for writing.

Error	Possible causes
(**Terminal**'s message is: `License registration error (code 14). Please, contact to support.`)	
Unknown error (**Terminal**'s message is: `License registration error (code 100). Please, contact to support.`)	Something went wrong on the technical level; analysis of the log-file is required to find the correct cause of the error.
Errors of getting information about licenses:	
Unable to connect to License Manager (**Terminal**'s message is: `Connection to ... (port ...) refused.`)	1. **License Manager** is not running. 2. Incorrectly specified address of the computer, on which the License Manager is running. 3. Ports has been configured incorrectly.
No registered licenses. (**Terminal**'s message is: `No license information for this user.`)	**License Manager** couldn't find at the specified address a registered license for the current license name (it is defined either in parameters of the **Solver-Agent**'s user or in the parameter, which was used for starting the **Solver**). Possible reasons are: 1. All licenses, which have been issued for this license name, are busy. 2. The license has expired. 3. The license name is incorrect
Error because of an incorrect license name:	
Bad symbols in license name **License name may contain (3-16 symbols):** **- Latin characters (case-sensitive)** **- digits 0-9** **- underscore symbol "_"**	An incorrect license name encountered. License names can contain from 3 to 16 symbols and may contain only case-sensitive uppercase and lowercase Latin characters (`A`, `B`, ..., `Z`, `a`, `b`, ..., `z`), digits (`0`, `1`, ..., `9`) and the underscore symbol (_).

See also: sections Pre-Postprocessor's error messages and warnings and Solver's error messages and warnings.

What to do if connection to License Manager has been lost

If you are using **License Manager** and connection to it has been lost, the **License loss** dialog box (with the message **License manager connection has been lost and couldn't be reestablished**) will open:



You can select the following options:

- **Try to recover**
- **Edit preferences**
- **Register a new license**
- **Save and terminate session**
- **Save and continue in read-only mode**

If the connection to **License Manager** has been lost because of network errors, then, after the cause of the error is fixed, select in this window the option **Try to recover** and then click **OK**.

When the connection to **License Manager** is restored, a message about this will be displayed.

If connection to **License Manager** fails, you can select either **Save and terminate session** or **Save and continue in read-only mode** option, which allows you to save your project (the last option allows you to view the project but not to modify or run it).

Pre-Postprocessor will not escape the read-only mode even after fixing the cause of entering this mode. All the project's not saved data will be lost on exiting the project.

Take this into consideration and make a responsible choice in the **License loss** dialog box (**Try to recover** the connection or **Edit preferences** might be better alternatives).

See also: Work of Pre-Postprocessor in the read-only mode.

4.9.2 Using Solver-Agent

Solver-Agent loads and unloads **Solvers** and delivers information to client modules about running **Solvers** and existing projects in the user's server directory.

Principal terms that are used when you work with **Solver-Agent**:

- ***Solver-Agent's** user* is the user under whose account interaction with **Solver-Agent** takes place. A user profile includes the user name and password, path to the user's server directory, license username used for the given user's licenses and the license password.
- *Server directory* is the directory where the server parts of user projects are stored.
- *New user registration* is creation of a new user profile.
- *User authorization* is connection to **Solver Agent** with an existing user profile.

Before starting use of **Solver-Agent**, it is necessary to set up the connection with it from **Pre-Postprocessor**. Change the settings with the **File > Preferences** command of the Main menu:

Parameter	Value
Solver-Agent > Address	**SAHost**
Solver-Agent > Port	**SAClientsPort**

Registering a new Solver Agent user

After installing *FlowVision* and receiving a license, register a new user with **Solver-Agent**. To do so:

- Open the user authentication on Solver-Agent dialog box from **Pre-Postprocessor** or from **Terminal**.
- Create a new **Solver-Agent** user user:
  - in **Pre-Postprocessor**, in the **Network** toolbar click the (**Solver agent log in**) button, and then in the **Solver agent connection** dialog box, which opens, click the (**Edit configurations**) button, and then, in the **Solver agent connection configurations** dialog box, which opens, click the **(Register a new user)** button and enter the required data.
  - in **Terminal** the **Solver agent connection** dialog box opens automatically at start (you also can open it any time later by the menu command **Users > Login**). In this dialog box click the (**Edit configurations**) button and then, in the **Solver agent connection configurations** dialog box, which opens, click the **(Register a new user)** button and enter the required data.

(see details in the section Connection to Solver-Agent and user authentication on Solver-Agent)

After user is *registered* in **Solver Agent**, it is possible to continue with user authentication on Solver-Agent.

Changing a Solver-Agent user profile

To change a **Solver-Agent** user profile:

- perform user authentication on Solver Agent in **Pre-Postprocessor** or **Terminal**, running as the same user whose profile must be modified
- open the user registration information dialog box and make necessary changes

Deleting a Solver-Agent user profile

To delete a **Solver-Agent** user profile:

- perform user authorization with Solver Agent in **Terminal**, running as the same user whose profile should be deleted
- use the **Users > Delete current user** menu command

Fixing problems

If problems are encountered during **Solver-Agent** operation, create a diagnostic information file using **Configurator** and send it to the *FlowVision* technical support service with a description of the problems encountered.

4.9.2.1 Connection to Solver-Agent and user authentication on Solver-Agent

Connection to **Solver-Agent** (which is done along with user authentication on **Solver-Agent**) is almost similar in **Pre-Postprocessor**, in **Viewer**, and in **Terminal**.

Some differences in its functionality, depending on the module, from which the connection is done, are:

- in **Viewer** and in **Terminal** you cannot enable automatic connection to **Solver-Agent**
- in **Viewer** you cannot register a new Solver-Agent user

Automatic connection to Solver-Agent and user authentication on Solver-Agent from Pre-Postprocessor

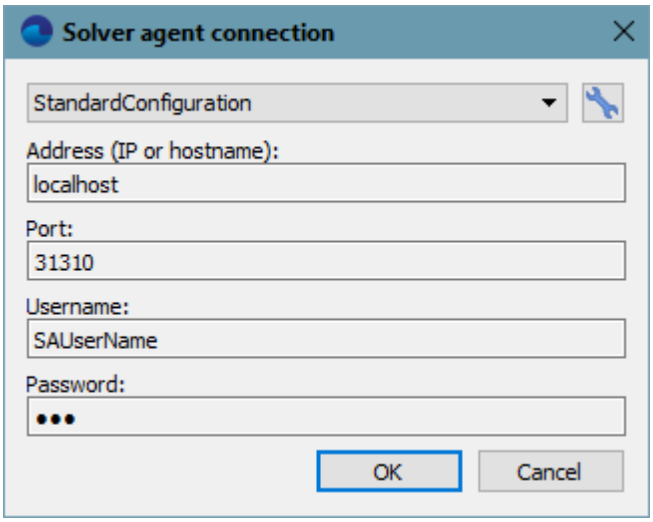
In **Pre-Postprocessor** you can enable automatic connection to **Solver-Agent** (and user authentication on it) that will be done at start of **Pre-Postprocessor** without displaying any dialog boxes for the user authentication.

If some **Configuration** for connection to **Solver-Agent** has already been set as default in **Pre-Postprocessor** (when the **Use for automatic connection** checkbox has been selected in the **Solver agent connection configurations** dialog box, see details in the subsection, *"Non-automatic connection to Solver-Agent and user authentication on Solver-Agent"* below), then connection to this **Configuration** is done automatically, just after starting of **Pre-Postprocessor**.

In this case, if you need to connect to **Solver-Agent** via another **Configuration**, or to create or tune other **Configurations**, click the  (**Solver agent log in**) screen button in the **Network** toolbar and follow instructions in the subsection *"Non-automatic connection to Solver-Agent and user authentication on Solver-Agent"*) below.

Non-automatic connection to Solver-Agent and user authentication on Solver-Agent

If automatic connection to **Solver-Agent** at start of **Pre-Postprocessor** is not set, then clicking the  (**Solver agent log in**) screen button in the **Network** toolbar opens the **Solver agent connection** dialog box (on condition that at least one configuration was created before):



Viewer and **Terminal** don't provide automatic connection and user authentication on **Solver-Agent**, so the **Solver agent connection** dialog box always opens at start of these modules.

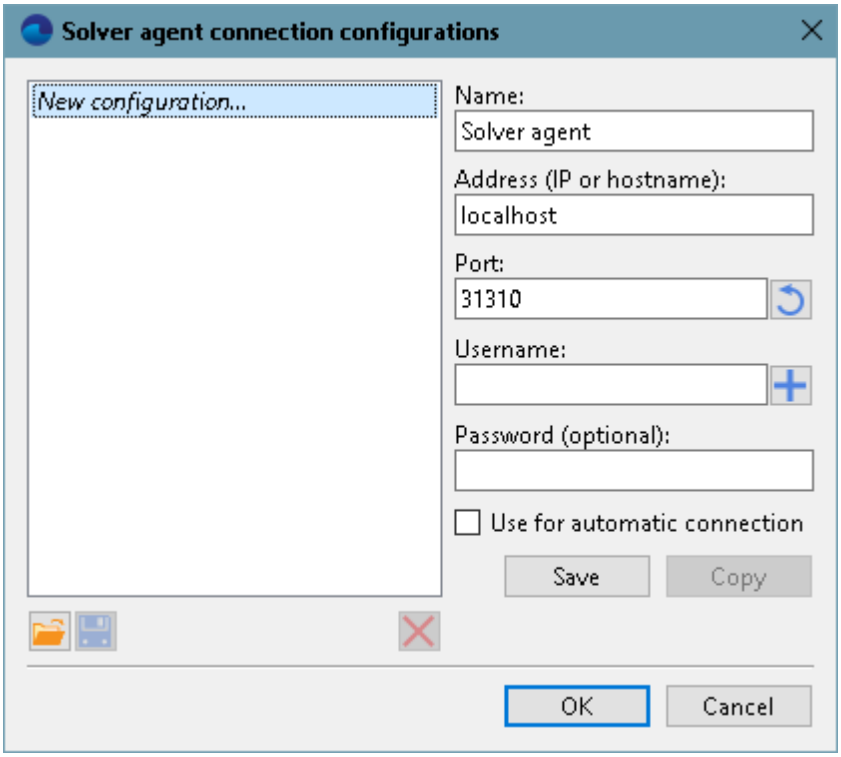
When you work in **Viewer** or **Terminal**, you can open the **Solver agent connection** dialog box any time later:

- in **Viewer** – by clicking the button  (**Connect**) in the **Connection** toolbar
- in **Terminal** – by the menu command **Users > Login**

The **Solver agent connection** dialog box contains the following interface elements:

Interface element	Description
List of connection configurations	It is a drop-down list, from which you select a desired configuration for connection to **Solver-Agent**.
 (**Edit configurations**) *(screen button)*	Clicking this button opens the **Solver agent connection configurations** dialog box (see below), in which you can create new configurations for connection to **Solver-Agent** or edit existing configurations.
Address (IP or hostname)	IP address or network name of the computer, on which **Solver-Agent** is installed. This field is read-only. Its data are entered in the **Solver agent connection configurations** dialog box.
Port	Port of **Solver-Agent**. This field is read-only. Its data are entered in the **Solver agent connection configurations** dialog box.
Username	User name for authentication on **Solver-Agent**. This field is read-only. Its data are entered in the **Solver agent connection configurations** dialog box.
Password	User's password for authentication on **Solver-Agent**. You can enter a password here if it is *not* specified in the **Solver agent connection configurations** dialog box. When the password is set in the **Solver agent connection configurations** dialog box, you cannot enter it here.
OK *(screen button)*	Connecting to **Solver-Agent** with parameters of the selected configuration.
Cancel *(screen button)*	Clicking this button closes the dialog box without connecting to **Solver-Agent**.

If no **Configuration** exists yet, the program will automatically open an empty **Solver agent connection configurations** where you have to create at least one configuration:

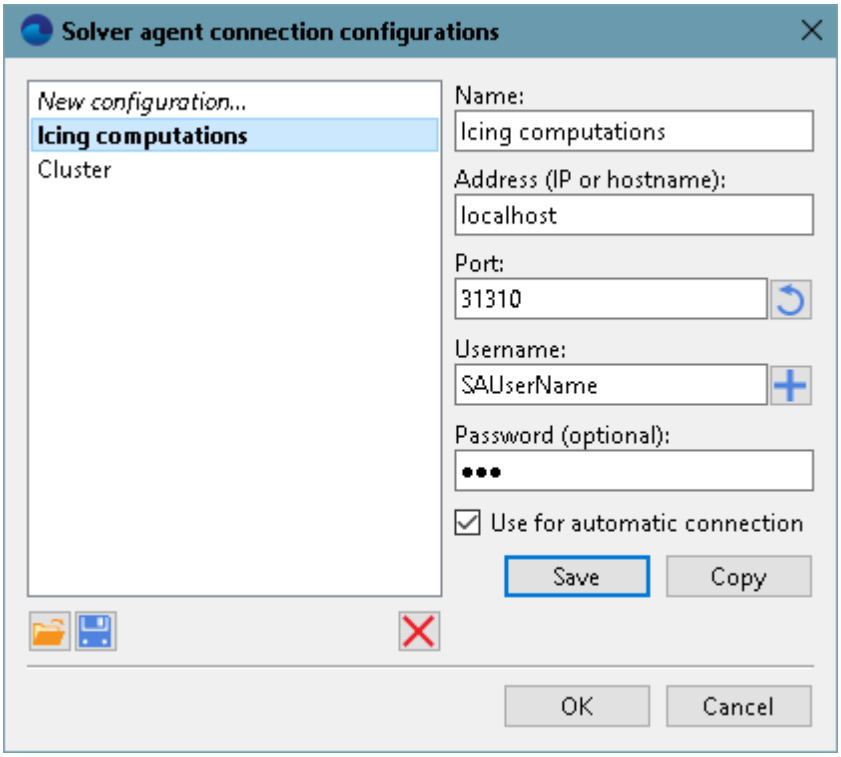


The **Solver agent connection configurations** dialog box contains the following user interface elements:

Interface element	Description
List of connection configurations	List of **Solver-Agent** connection **Configurations**. When the **New configuration** line is selected, after filling all fields and clicking on the **Save** screen button, a new **Configuration** will be created. You can change order, in which **Configurations** are listed, by dragging and dropping them by the mouse. ⚠ If you don't click the **Save** button (see below), then selecting another list line will cancel your changes.
Name	Name of the **Configuration** that is currently selected from the list. When you create a new **Configuration**, the program prompts "**Solver agent**" as its default name.
Address (IP or hostname)	IP address or network name of the computer, on which **Solver-Agent** is installed (see section "Initial Configuration").
Port	Port of the computer, on which **Solver-Agent** runs, to which the connection will be done (see section Initial Configuration).
(Default port) *(screen button)*	When you click this button, the **Port** field is filled with the default port number that is used in the current version of *FlowVision*.
Username	User name for authentication on **Solver-Agent**
(Register a new user) *(screen button)*	Registering a new user on **Solver-Agent**. The **User Registration** dialog box will open where you can register a new Solver-Agent user. Title of this dialog box will include the name of the configuration. ⚠ This functionality is available in **Pre-Postprocessor** and **Terminal** only. In **Viewer** this functionality is disabled. See details in sections Registering a new Solver-Agent user and Registration data (profile) of Solver-Agent's user and their change.
Password (optional)	User's password for authentication on **Solver-Agent**. If you don't specify the password here, it will be necessary to enter the password at each connection to **Solver-Agent**

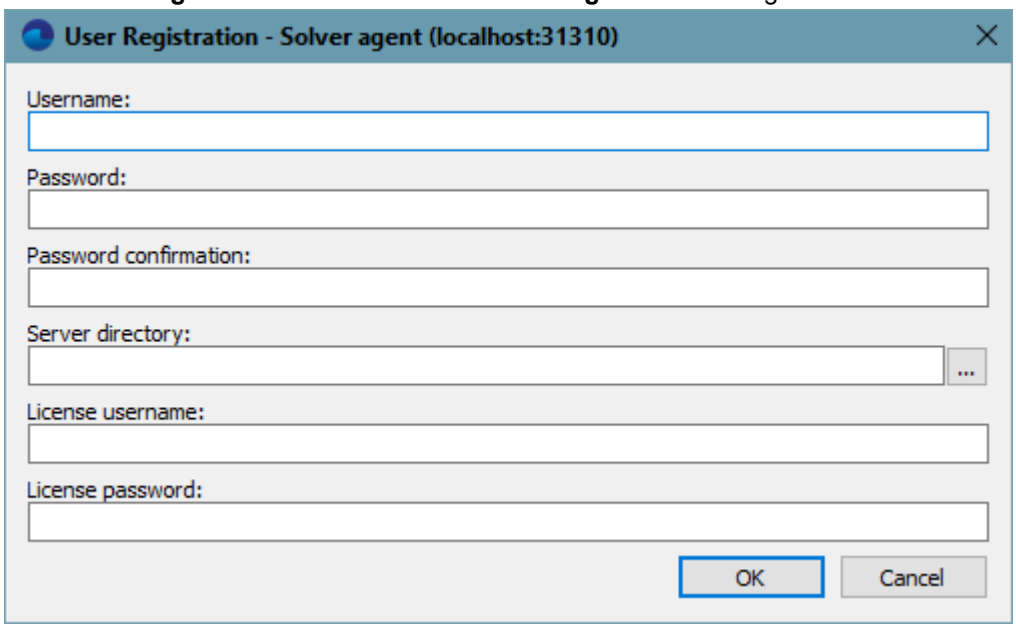
Interface element	Description
	using the selected configuration.
Use for automatic connection *(check box)*	Use the selected **Configuration** as default one (for automatic connection). The default **Configuration** is displayed in the list in boldface. ⚠ This functionality is available in **Pre-Postprocessor** only. In **Viewer** and in **Terminal** this functionality is disabled.
Save *(screen button)*	Clicking this button cause applying and saving changes that were made for the selected **Configuration**. Saving the changes to the current line is also done when you click screen buttons **Copy**,  **(Export selected configurations)**, and **OK**. ⚠ Until you save changes entered to the current **Configuration** (by clicking either **Save**, **Copy**, or ), selecting another line from the list of **Configurations** will cancel these changes.
Copy *(screen button)*	Clicking this button copies the selected **Configuration** to a new line in the list. Also the changes entered to the current **Configuration** will be saved.
 **(Import configurations)** *(screen button)*	Importing one or several **Configurations** from a text file.
 **(Export selected configurations)** *(screen button)*	Exporting one or several **Configurations** selected from the list to a text file. Also the changes entered to the current **Configuration** will be saved. The program does *not* export user passwords.
 **(Remove selected configurations)** *(screen button)*	Removing one or several **Configurations** selected from the list. You can also use the **Del** key on the keyboard. The program will request you to confirm this.
OK *(screen button)*	Closing the dialog box and saving the changes. Also the changes entered to the current **Configuration** will be saved.
Cancel *(screen button)*	Closing the dialog box and canceling the changes that were made after opening this dialog box.

Here is an example of the **Solver agent connection configurations** dialog box that contains several previously created configurations. The default **Configuration** is shown in the list in boldface:

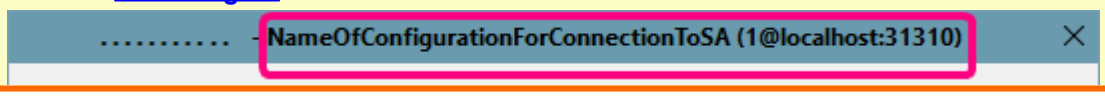


4.9.2.2 Registering a new user

Registering a new **Solver-Agent** user is done from the **User Registration** dialog box:

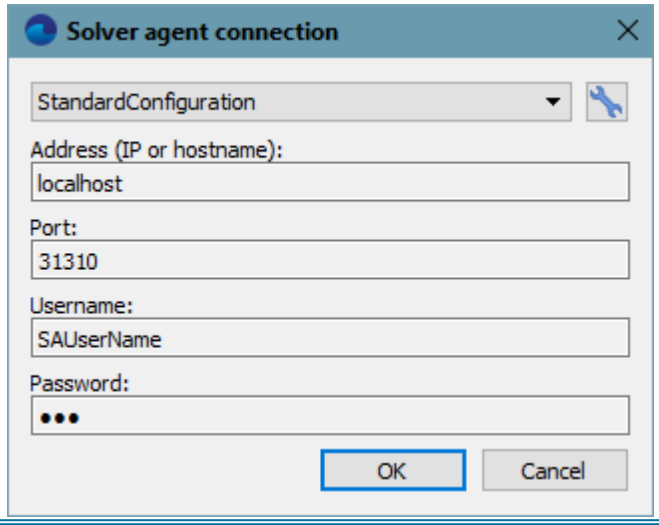
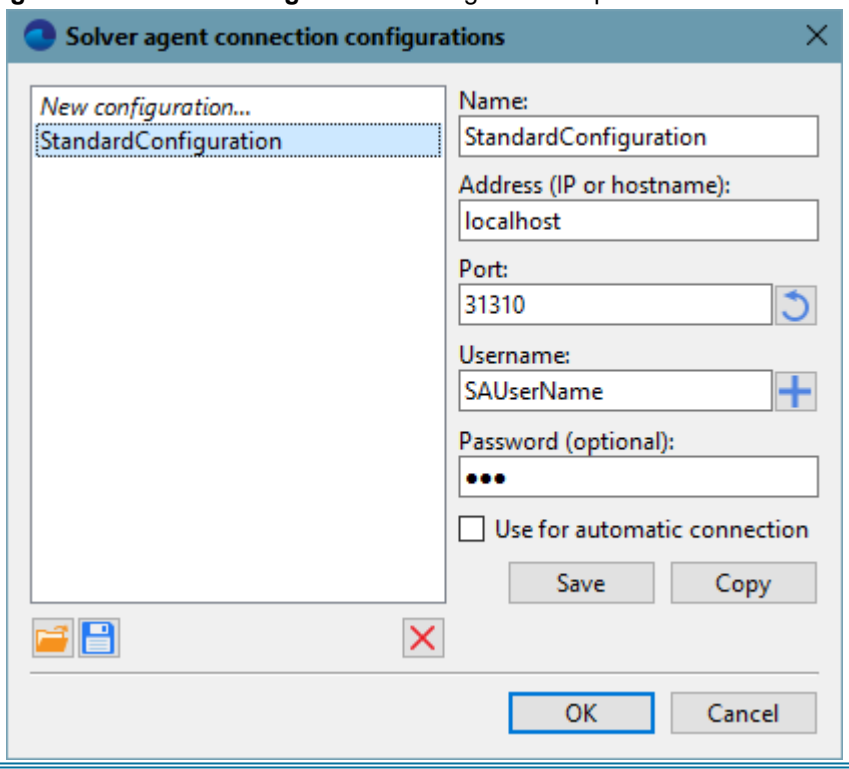


The header of this dialog box contains the name and parameters of the used **Configuration** for connection to **Solver-Agent**:



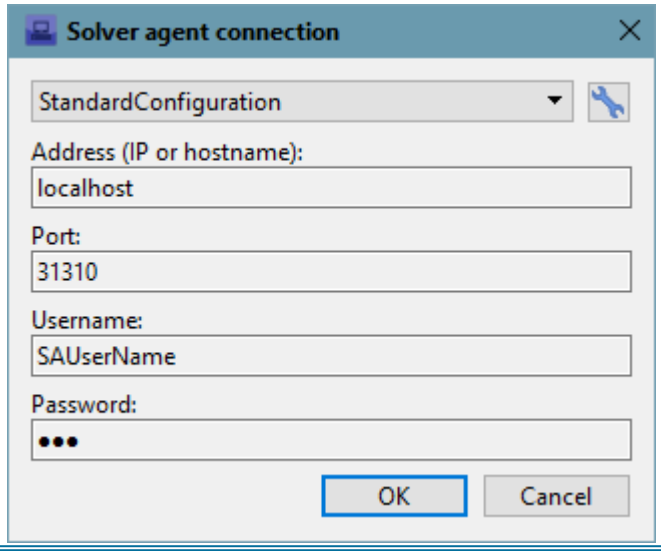
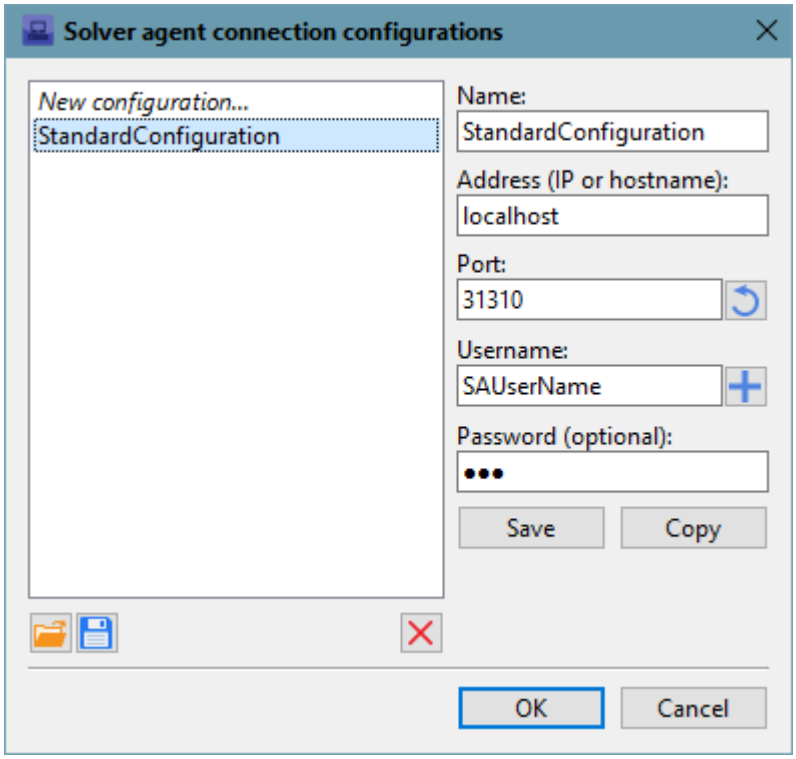
To do this from **Pre-Postprocessor**, follow the steps:

Step	Actions
1	The **Solver agent connection** dialog box might open automatically at start of **Pre-Postprocessor**. You can also open it by clicking the (**Solver agent log in**) screen button in the **Network** toolbar.

Step	Actions
	
2	In this dialog box click the  (**Edit configurations**) screen button. The **Solver agent connection configurations** dialog box will open: 
3	In this dialog box click the  **(Register a new user)** screen button. An empty **User Registration** dialog box will open. Fill in its fields and click **OK**.

To do this from **Terminal**, follow the steps:

Step	Actions
1	The **Solver agent connection** dialog box might open automatically at start of **Terminal**. You can also open it by the menu command **Users > Login**.

Step	Actions
	
2	In this dialog box click the  (**Edit configurations**) screen button. The **Solver agent connection configurations** dialog box will open:  When the **Solver agent connection configurations** dialog box is opened from **Terminal**, it does not contain the **Use for automatic connection** checkbox.
3	In this dialog box click the  **(Register a new user)** screen button. An empty **User Registration** dialog box will open. Fill in its fields and click **OK**.

The **User Registration** dialog box contains the following fields:

Field	Description
Username	Name for user's authorization on **Solver-Agent**. The name **Admin** is reserved for the *FlowVision*'s administrator. The *FlowVision*'s administrator's password is suitable to any user profile.
Password	Password for user's authorization on **Solver-Agent**.

Field	Description
Password confirmation	Enter the **Password** again (to avoid typos).
Server directory	This is the absolute path to the user's server directory on the computer, on which **Solver** and **Solver-Agent** are installed. You can specify not only one directory but a list, which specify several directories separated by semicolons (";"). Each time when a new project is loaded on **Solver**, a subdirectory with the is created in the first directory from list. If you wish, you can manually move the server part of the project into another server directory that is specified in the list (so you can group the projects according to your preferences). When the program accesses to the server part of the project, it inspects *all* server directories that are specified in the list. It is strongly recommended *not* to copy the server part of one project in multiple server directories specified in the list. The button ... allows you to select the server directory from a standard operating system's dialog box for access to files.
License username **License password**	License username and license password.It is possible to register multiple users, that share one license username. If multiple users share the same license username, each of the users captures a certain number of licenses from the total number. And, in this situation, users do not see each other's **Solvers**. When you keep these both fields blank, you registered a **Solver-Agent**'s user with reduced functionality. Such user will be able load projects to **Solver**, connect to projects from **Pre-Postprocessor**, view results (including viewing history of saved data), create new visualization **Layers** and **Characteristics**. But such user cannot save projects or run computation of projects.

License username and **License password**, and **Username** and **Password** for **Solver-Agent** are two different user names and passwords !!!

License username is specified by the user when receiving the registration information, a license is issued for this username, and the **License password** is issued with the license.

User name and **Password** for **Solver-Agent** are specified by the user when registering on **Solver-Agent**, it is necessary for different users, which use a single **Solver-Agent**, do not be confused with their projects.

When registering a new user on **Solver-Agent**, the **License username** and **License password** are only used as information confirming the right to register this user, but the **Username** and **Password** for **Solver-Agent** should differ from **License username** and **License password**.

The server directory must not be same as the installation directory and must be writable for the user, as which **Solver-Agent** runs.

Notes:

[1] The name **Admin** is reserved for the *FlowVision*'s administrator. The administrator's password is suitable to any user profile.

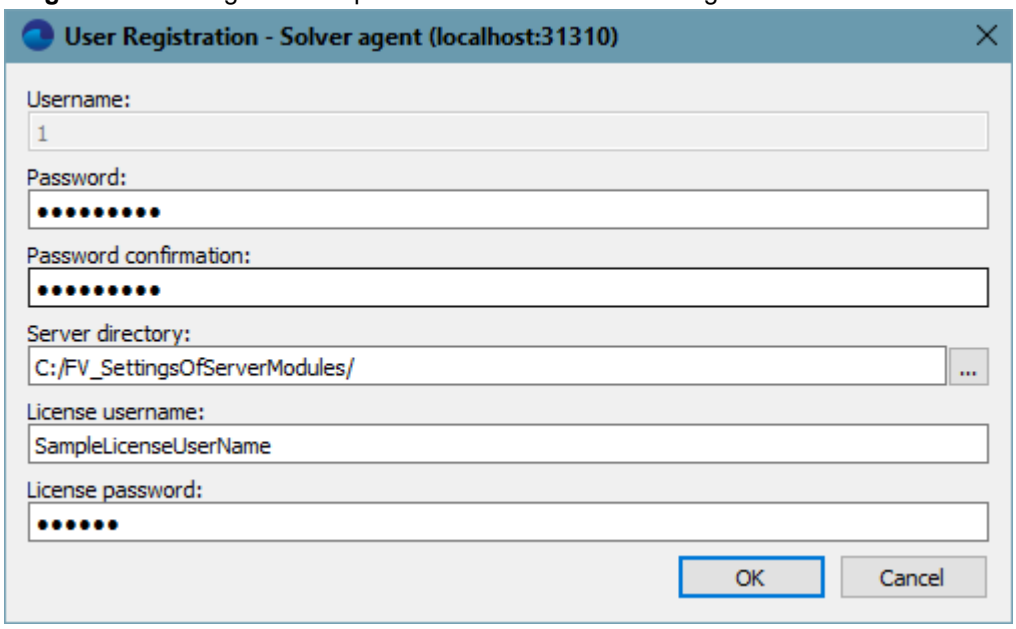
[2] It is possible to specify multiple directories separated by semicolons (";").

[3] It is possible to register multiple users, that share one license username. If multiple users share the same license username, each of the users captures a certain number of licenses from the total number. And, in this situation, users do not see each other's **Solvers**.

4.9.2.3 Registration data (profile) of Solver-Agent's user and their change

To change the profile of the **Solver-Agent** user, do the following steps:

Step	Actions
1	In **Pre-Postprocessor** click the button (**Edit solver agent user information**) in the **Network** toolbar.
	In **Terminal** use the menu command **Users > Edit user info**.

Step	Actions
2	The **User registration** dialog box will open that contains the user's registration data:  See section Registering a new Solver-Agent user for description of the registration data. Make changes to the user profile and then click **OK** (you can change any parameter except **Username**). The header of this dialog box contains the name and parameters of the used **Configuration** for connection to **Solver-Agent**: 

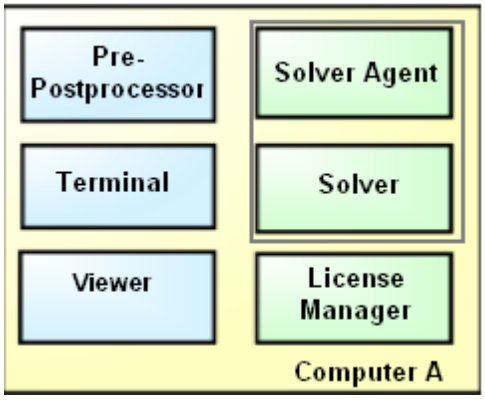
You can change the **Solver-Agent** user's profile only after your Solver-Agent user authentication and if you have system administrator's privileges.

4.9.3 Typical configurations

The following typical situations are considered:

1. Calculations on a user computer (available for *Windows* only)
2. Existing network connection between user computer and calculation computer
3. No network connection between user computer and calculation computer
4. Existing network connection between user computer and cluster; License Manager is installed on cluster
5. Existing network connection between user computer and cluster; License Manager is installed outside cluster
6. No network connection between user computer and cluster
7. No network connection between user computer and cluster; no graphical user interface on cluster

4.9.3.1 Calculations on a user computer



Installation

Step	Actions
1	Install on the computer all *FlowVision*'s modules. In this case, all network connections are configured automatically.

Licensing

Step	Actions
1	Start **License Manager** (in *Windows*' Task Manager **FvLicense** will appear).
2	Receive and register a license.

Registering

Step	Actions
1	Start **Solver-Agent** (in *Windows*' Task Manager **FvSolverAgent** will appear).
2	Register a new user on **Solver-Agent**.

Preparing a project

Step	Actions
1	Start **License Manager**.
2	Start **Pre-Postprocessor**. Create a new project or open an existing one.
3	Start **Solver-Agent**. Load the project on a **Solver**.

Starting a computation

Step	Actions
1	Start **License Manager**.
2	Start **Solver-Agent**.

Step	Actions
3	Load the project on **Solver** from **Pre-Postprocessor** or from **Terminal**. Run the project for computation.

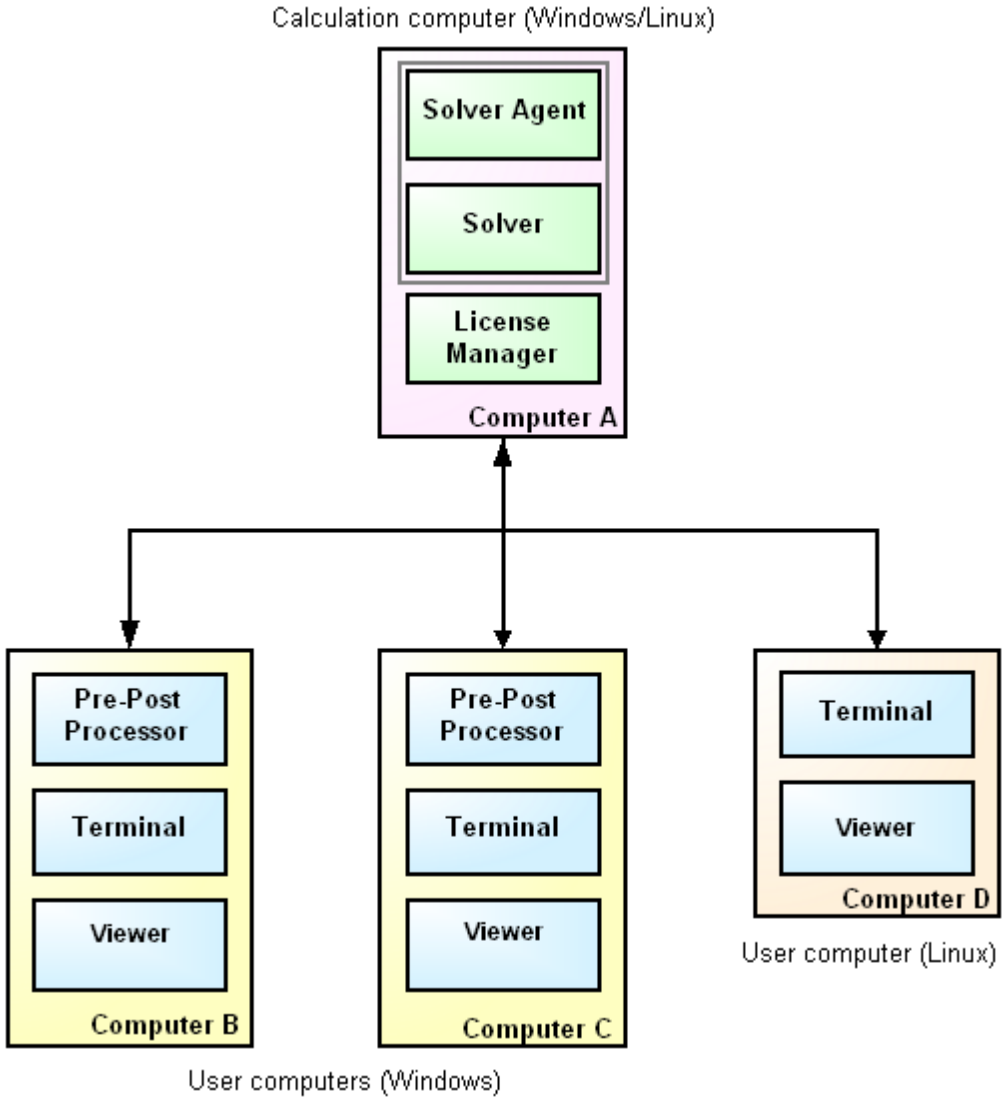
Displaying results

Step	Actions
1	Start **License Manager**.
2	Start **Solver-Agent**.
3	Displaying results in **Pre-Postprocessor**: a. Open the project in **Pre-Postprocessor** and then load it on **Solver**.
4	Displaying results in **Viewer**: a. Load the project on **Solver** using **Terminal**. b. In **Viewer** connect to a **Solver**, on which the project is loaded.

Notes:

1. *FlowVision* modules (`FvSolverAgent.exe`, `FvLicense.exe`, `FvTerminal.exe`, `FvRRR.exe`, `FvViewer.exe`) can be started in an arbitrary order.
2. Information about users, projects, and licenses available on a computer, can be received from **Terminal** or **Pre-Postprocessor**.
3. When you start **Pre-Postprocessor** first time, it is necessary to enter the **License username**. You can also specify the **License username** using the menu command **File > Preferences**. It is not necessary to enter the **License username** at next starts.

4.9.3.2 Existing network connection between user computer and calculation computer



Installation

Step	Actions
1	Install the following modules on **Computer A**: • **Solver** and **Solver-Agent** • **License Manager**
2	Install the following modules on each of user computers (**B**, **C**, **D**): • **Pre-Postprocessor** (only on computers **B** and **C**, which run under *Windows*) • **Terminal** • **Viewer** During installation the modules on *Windows*'s user computers (**B** and **C**) specify the IP address or host name of the computer **A** (on which **License Manager** and **Solver-Agent** are installed).

Licensing

Step	Actions
1	Start **License Manager** on computer **A**.
2	Receive and register a license on any of the user computers (**B**, **C**, **D**) from **Terminal** or **Pre-Postprocessor**.

Registering

Step	Actions
1	Start **Solver Agent** on computer **A**.
2	On any of the user computers (**B**, **C**, **D**) register a user on **Solver-Agent** from **Terminal** or **Pre-Postprocessor**.

Preparing a project

Step	Actions
1	Start **License Manager** on computer **A**.
2	On any of *Windows*'s user computers (**B** and **C**) create in **Pre-Postprocessor** a new project or open an existing project.
3	Start **Solver Agent** on computer **A**.
4	Load the project on **Solver**.

Starting solve

Step	Actions
1	Start **License Manager** on computer **A**.
2	Start **Solver-Agent** on computer **A**.
3	Load the project on **Solver** (from **Pre-Postprocessor** or **Terminal**) from any user computer (**B**, **C**, **D**). Run the project on computation.

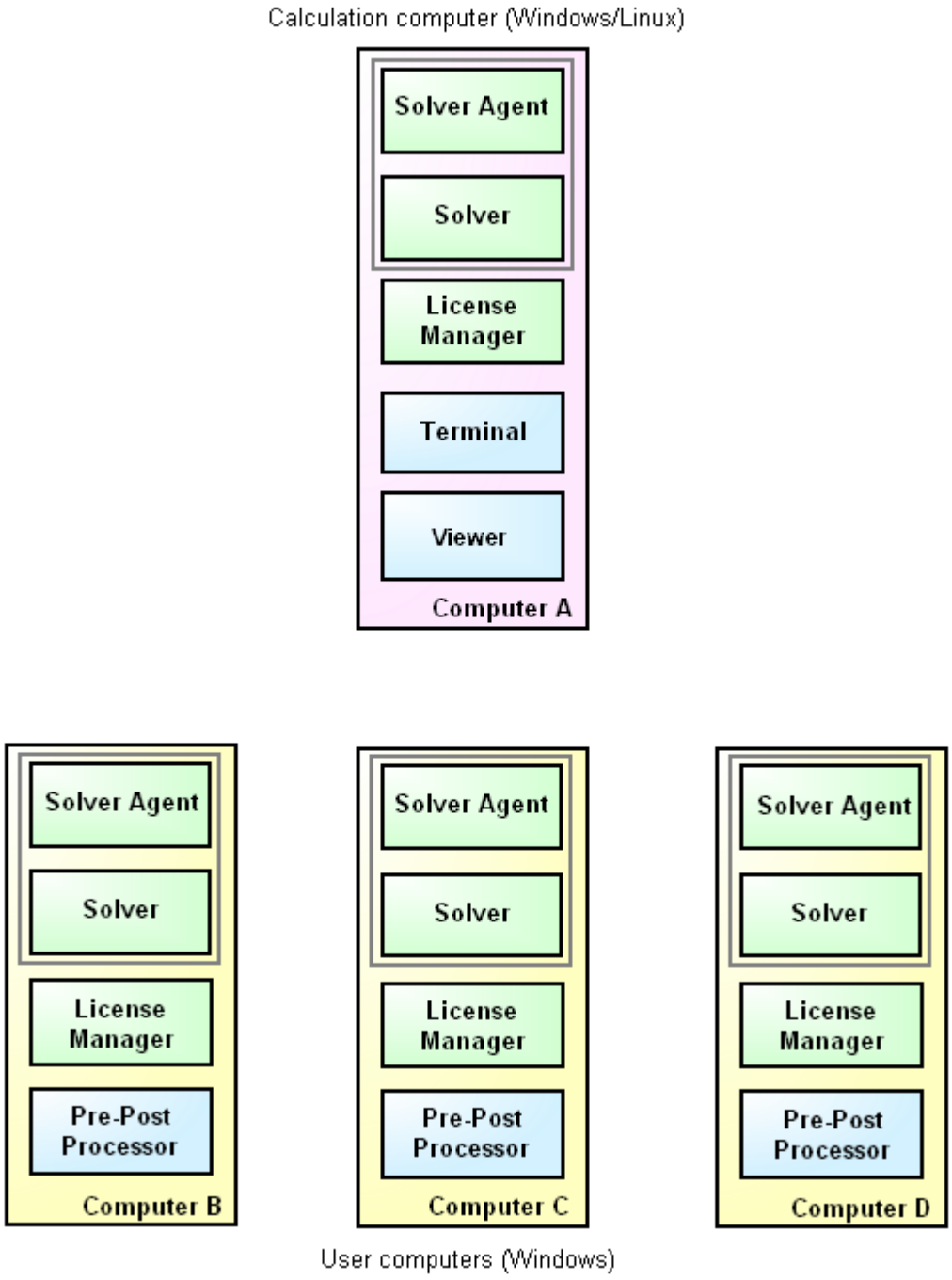
Displaying results

Step	Actions
1	Start **License Manager** on computer **A**.
2	Start **Solver Agent** on computer **A**.
3	Displaying results in **Pre-Postprocessor**: a. Open the project in **Pre-Postprocessor** and upload it to **Solver**.
4	Displaying results in **Viewer**: a. Load the project to **Solver** using **Terminal**. b. In **Viewer** connect to the **Solver**, on which the project has been loaded.

Notes:

1. **License Manager** can be installed on a separate computer, which is accessible to other computers via the network. In this case, on computers **A**, **B**, **C** you have to specify the address of the computer on which **License Manager** is installed. On *Windows*-based computers, this can be done during installation. On *Linux*-based computers this is to be done after the installation during the configuration (see section Parameters in configuration files).
2. See notes for a typical configuration described in section Calculations on a user computer.

4.9.3.3 No network connection between user computer and calculation computer



Installation

Step	Actions
1	Install the following modules on **Computer A**: • **Solver** and **Solver-Agent** • **License Manager** • **Terminal** • **Viewer**
2	Install the following modules on each of user computers (**B**, **C**, **D**): • **License Manager** • **Solver** and **Solver-Agent** • **Pre-Postprocessor** When installing on *Windows*-based computers, all network connections are configured automatically.

Licensing

Step	Actions
1	Start **License Manager** on each computer.
2	On each computer receive and register a license using **Terminal** (on **Computer A**) or **Pre-Postprocessor** (on computers **B**, **C**, **D**). The number of licenses must match to the number of individual computers!

Registering

Step	Actions
1	Start **Solver-Agent** on each computer.
2	On each computer register a new user of **Solver-Agent** using **Terminal** (on **Computer A**) or **Pre-Postprocessor** (on computers **B**, **C**, **D**).

Preparing a project

Step	Actions
1	On one of the user computers (**B**, **C**, **D**), follow the steps for the case Calculations on a user computer.
2	Copy the server part of the project from the server directory on the user's computer (**B**, **C**, **D**) into the server directory on the calculation computer (**A**).

Starting solve

Step	Actions
1	Start **License Manager** on the calculation computer (**A**).
2	Start **Solver Agent** on the calculation computer (**A**).
3	Run the project on computation from **Terminal**.

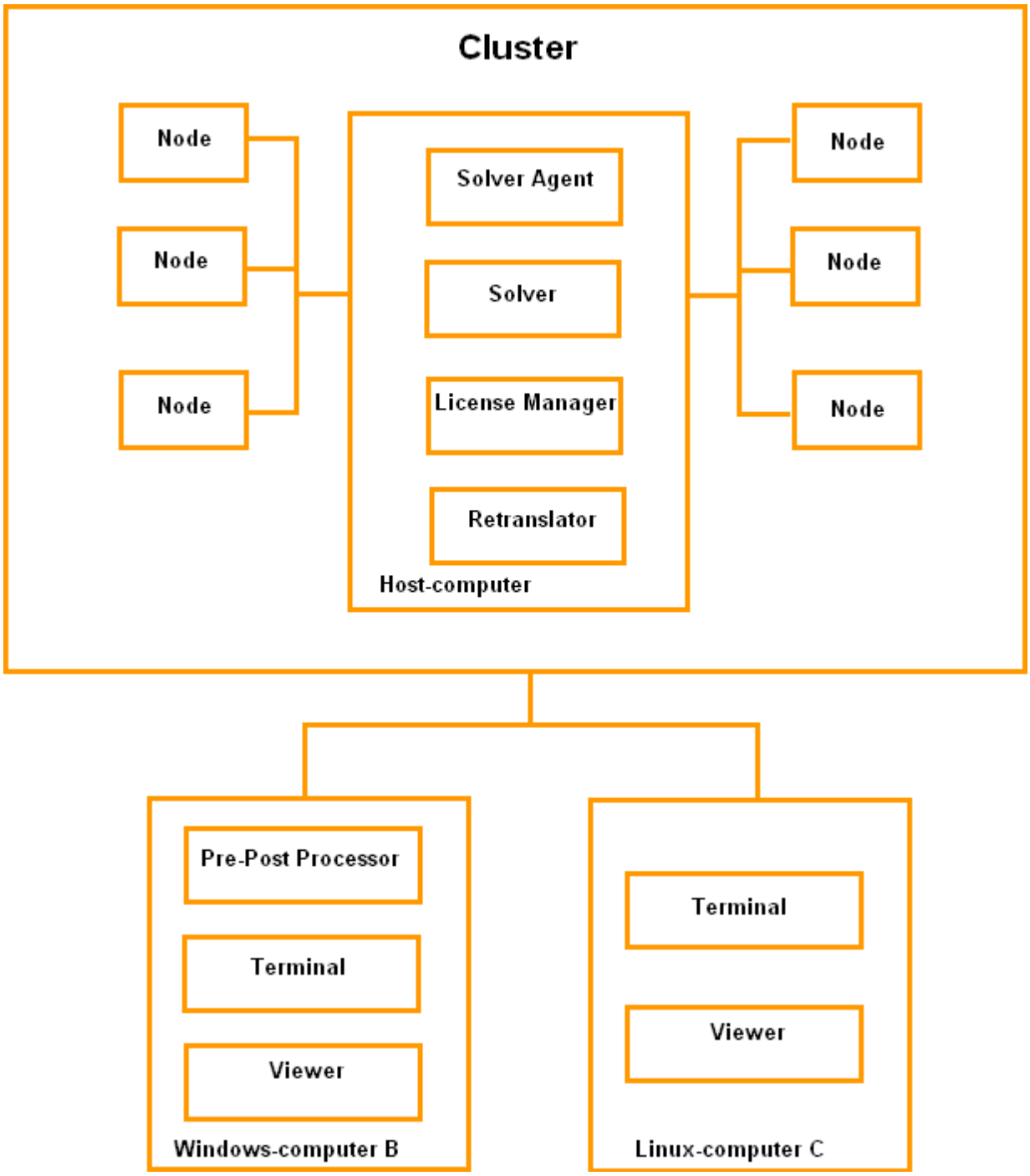
Displaying results

Step	Actions
1	Displaying results in **Pre-Postprocessor**: a. Copy the server part of the project from the server directory on the calculation computer (**A**) to the server directory on a user computer (**B**, **C**, **D**). b. Start **License Manager** on the user computer, where the project has been copied. c. Start **Solver-Agent** on the user computer, where the project has been copied. d. Open the project in **Pre-Postprocessor** and connect to **Solver**.
2	Displaying results in **Viewer**: a. Start the **License Manager** on the calculation computer (**A**). b. Start **Solver Agent** on the calculation computer (**A**). c. Load the project on **Solver** and **Terminal**. d. Connect to the **Solver**, on which the project is loaded.

Notes:

1. *FlowVision* modules (**`FvSolverAgent.exe`**, **`FvLicense.exe`**, **`FvTerminal.exe`**, **`FvRRR.exe`**, **`FvViewer.exe`**) can be started in an arbitrary order.
2. Information about users, projects, and licenses available on a computer, can be received from **Terminal** or **Pre-Postprocessor**.
3. When you start **Pre-Postprocessor** first time, it is necessary to enter the **License username**. You can also specify the **License username** using the menu command **File > Preferences**. It is not necessary to enter the **License username** at next starts.

4.9.3.4 Existing network connection between user computer and cluster; License Manager is installed on cluster



Installation

Step	Actions
1	On the host-computer of the cluster install*): • **Solver** and **Solver-Agent** • **License Manager** • *MPI*
2	On each cluster's node install: • system libraries (for *Windows*-based cluster) • *MPI*
3*)	On the user's *Windows*-based computer **B** install:

Step	Actions
	• **Pre-Postprocessor** • **Terminal** • **Viewer**
4*)	On the user's *Linux*-based computer **C** install: 1. **Terminal** 2. **Viewer**

*) During the installation on computers **B** and **C** (where **License Manager** and **Solver-Agent** are installed), specify the IP address or host name of the host-computer of the cluster.

Configuring

Step	Actions
1	Open access to *FlowVision*'s installation directory on the host-computer for all nodes of the cluster.
2	Specify directories for user files on client computers and on host-computer of the cluster for each *FlowVision* module prior to its first run. On the cluster, specify the absolute network path to directories with user files of modules installed on the host-computer of the cluster. For all nodes of the cluster, open access to these directories for reading and writing.
3	In the configuration file of Solver-Agent in the parameter **SolverProxyHost** specify the IP address or host name of the host-computer of the cluster and specify in the parameter **SolverProxyPort** any free port on the host-computer.
4	In the configuration file of Solver-Agent in all command lines for starting Solver specify the absolute network path to the *machinefile*, *pwdfile* and **Solver**.
5	In the configuration file of Solver in the parameters **FvLicenseHost** and **SAHost** specify the IP address or host name of the host-computer of the cluster.
6	In the configuration file of Retranslator specify **Channel = (Port1)**, where the port **(Port1)** must match to the value of **SolverProxyPort** in the configuration file of Solver-Agent.

Licensing

Step	Actions
1	Start **License Manager** on the host-computer of the cluster.
2	Receive and register a license on the cluster using **Terminal** or **Pre-Postprocessor**.

Registering

Step	Actions
1	Start **Solver Agent** on the host-computer of the cluster
2	Register a new user of **Solver-Agent** using **Terminal** or **Pre-Postprocessor**. Specify an absolute network path to the Server directory. Open access to the directory for reading and writing from each cluster's node.

Preparing a project

Step	Actions
1	Start **Retranslator** on the host-computer of the cluster.

Step	Actions
2	See case Existing network connection between user computer and calculation computer.

Starting solve

Step	Actions
1	Start **Retranslator** on the host-computer of the cluster.
2	See case Existing network connection between user computer and calculation computer taking into account the cluster's configuration (presence of queues, etc.).

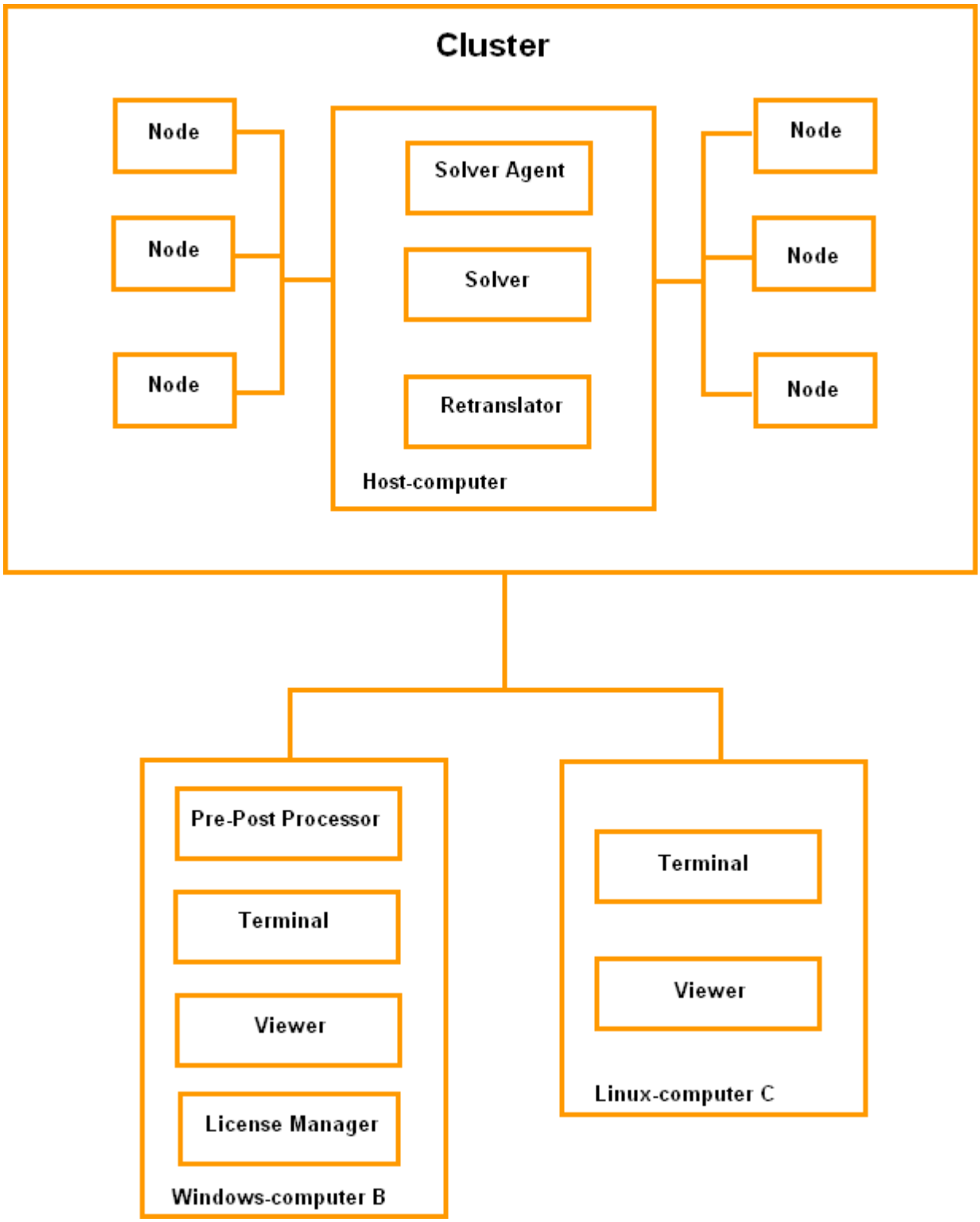
Displaying results

Step	Actions
1	Start **Retranslator**.
2	See case Existing network connection between user computer and calculation computer.

Note:

*) **Retranslator** is always installed automatically.

4.9.3.5 Existing network connection between user computer and cluster; License Manager is installed outside cluster



Installation

Step	Actions
1	On the host-computer of the cluster install: • **Solver** and **Solver-Agent** • **Retranslator** • *MPI*

Step	Actions
2	On each cluster's node install: • system libraries (for *Windows*-based cluster) • *MPI*
3	On the user *Windows*-based computer **B** install: • **Pre-Postprocessor** • **Terminal** • **Viewer** • **License Manager** During installation the modules on *Windows*'s user computers **B**, specify the IP address or host name of the host-computer of the cluster as the IP address or host name of the computer, on which **Solver-Agent** is installed.
4	On the user *Linux*-based computer **C** install: • **Terminal** • **Viewer**

Configuring

Step	Actions
1	Open access to *FlowVision*'s installation directory on the host-computer for all nodes of the cluster.
2	Specify directories for user files on client computers and on host-computer of the cluster for each *FlowVision* module prior to its first run. On the cluster, specify the absolute network path to directories with user files of modules installed on the host-computer of the cluster. For all nodes of the cluster, open access to these directories for reading and writing.
3	In the configuration file of Solver-Agent (FvSolverAgent.cfg) in the parameter **SolverProxyHost** specify the IP address or host name of the host-computer of the cluster and specify in the parameter **SolverProxyPort** any free port on the host-computer.
4	In the configuration file of Solver-Agent in each command line specify the absolute network path to machinefile, pwdfile and **Solver**.
5	In the configuration file of Solver in the parameters **FvLicenseHost** and **SAHost** specify the IP address or host name of the host-computer of the cluster.
6	In the configuration file of Retranslator specify **Channel = (Port1)**, where the port **(Port1)** must match to the value of **SolverProxyPort** from configuration file of Solver-Agent and specify **Channel = LicensePortCluster:HostB:LicensePortB**, where **LicensePortCluster** is the value of **FvLicensePort** from configuration file of Solver, **HostB** is theIP address or host name of the computer **B**, on which **License Manager** is running, **LicensePortB** is the value of **FvLicensePort** from configuration file of License Manager.

Licensing

Step	Actions
1	Start **License Manager**.
2	Receive and register a license using **Terminal** or **Pre-Postprocessor**.

Registering

See section Existing network connection between user computer and cluster; License Manager is installed on cluster.

Preparing a project

See section Existing network connection between user computer and cluster; License Manager is installed on cluster.

Starting solve

See section Existing network connection between user computer and cluster; License Manager is installed on cluster.

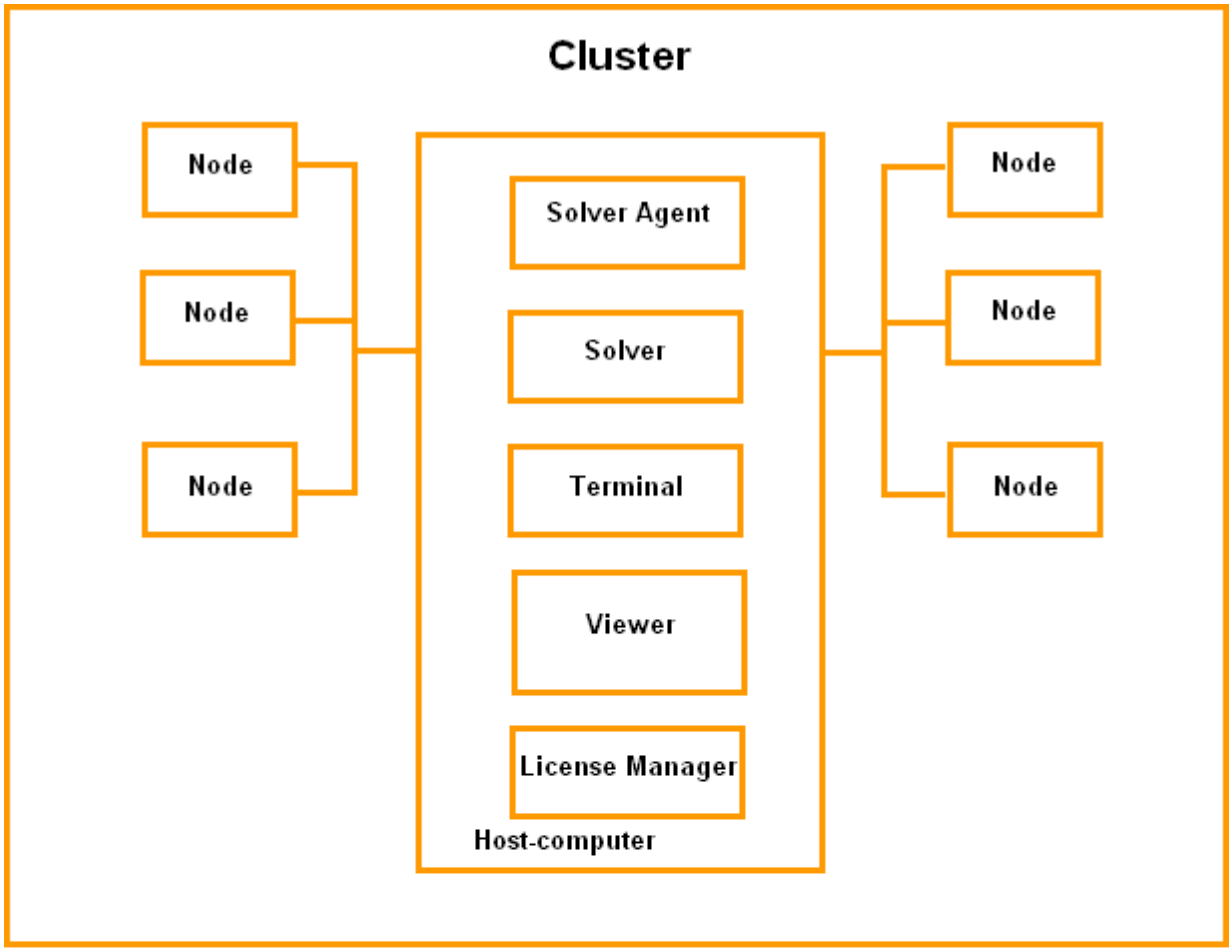
Displaying results

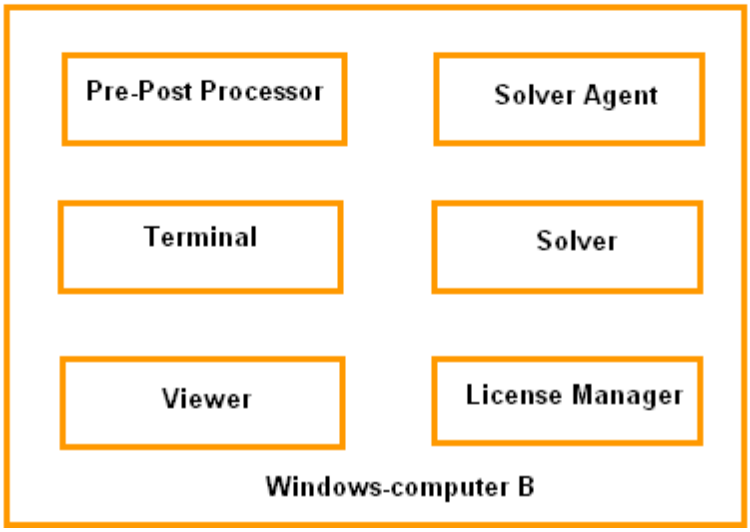
See section Existing network connection between user computer and cluster; License Manager is installed on cluster.

Note:

*) **Retranslator** is always installed automatically.

4.9.3.6 No network connection between user computer and cluster





Installation

Step	Actions
1	On the host-computer of the cluster install: • **Solver** and **Solver-Agent** • **License Manager** • **Terminal** • **Viewer** • *MPI*

Step	Actions
2	On each cluster's node install: • system libraries (for *Windows*-based cluster) • *MPI*
3	On the user *Windows*-based computer **B** install: • **Pre-Postprocessor** • **Terminal** • **Viewer** • **Solver** and **Solver-Agent** • **License Manager**

Configuring

Step	Actions
1	Do configuring of the cluster, as described in section The connection between the user's computer and a cluster, steps 1-5.
2	Do configuring of the user computer, as described in section Calculations on a user computer.

Licensing

See section No network connection between user computer and calculation computer.

Registering

Step	Actions
1	See section No network connection between user computer and calculation computer.
2	On the cluster, specify the absolute network path to the network server directory. Open access to this directory from all nodes of the cluster for reading and writing.

Preparing a project

See section No network connection between user computer and calculation computer.

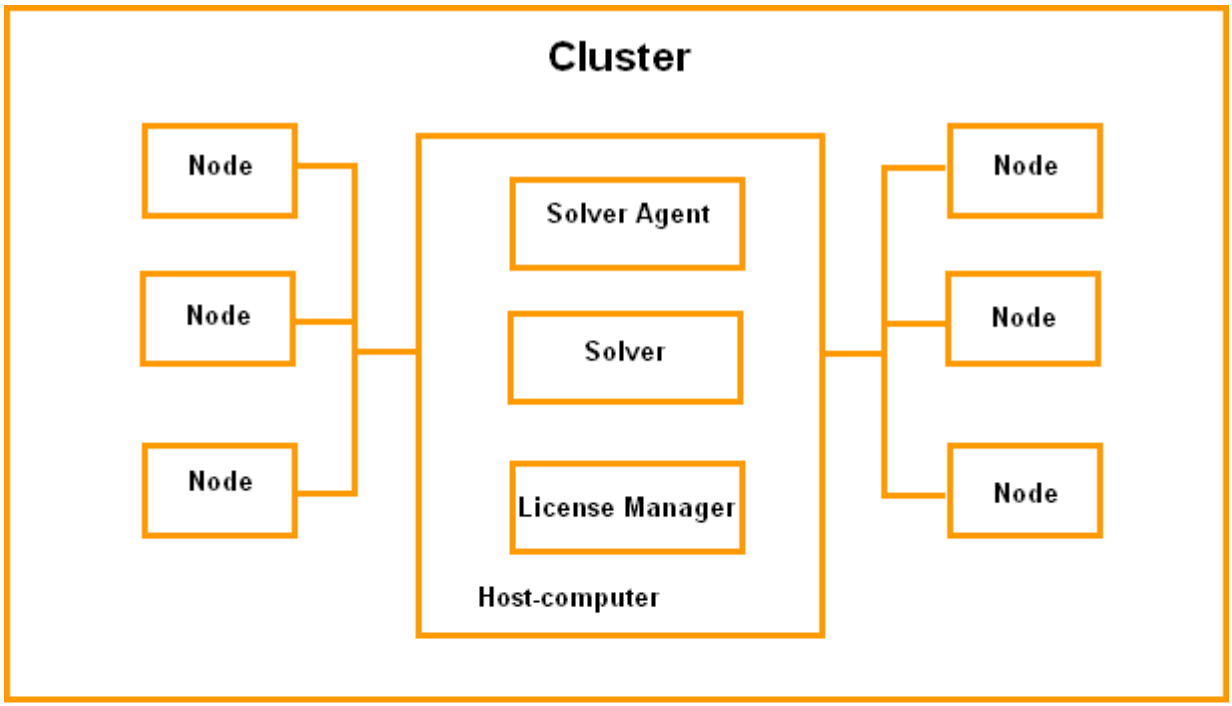
Starting solve

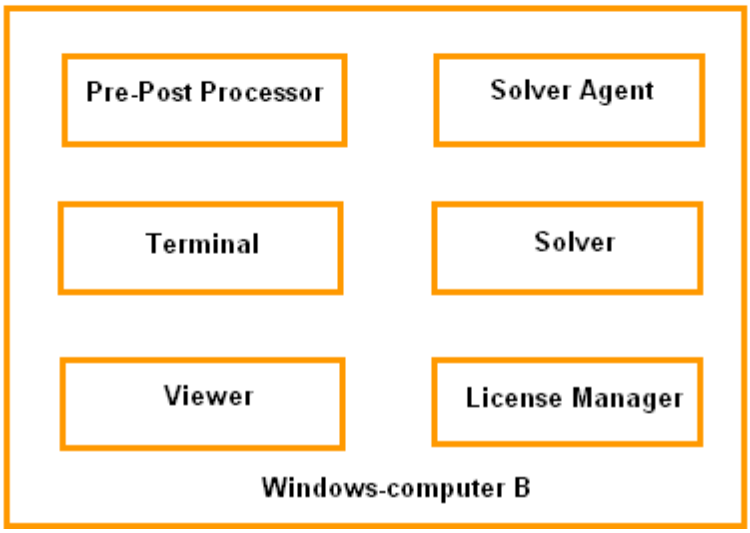
See section No network connection between user computer and calculation computer taking into account the cluster's configuration (presence of queues, etc.).

Displaying results

See section No network connection between user computer and calculation computer.

4.9.3.7 No network connection between user computer and cluster; no graphical user interface on cluster





Installation

Step	Actions
1	On the host-computer of the cluster install: • **Solver** and **Solver-Agent** • **License Manager** • *MPI* • the `FvLicenseUtil` utility
2	On each cluster's node install *MPI*.
3	On the user *Windows*-based computer **B** install: • **Pre-Postprocessor** • **Terminal**

Step	Actions
	• **Viewer** • **Solver** and **Solver-Agent** • **License Manager**

Configuring

See section No network connection between user computer and cluster.

Licensing

Step	Actions
1	For licensing on *Windows*-based computer **B** see section Carrying out calculations on the user's computer.
2	For licensing on the cluster: a. Start **License Manager** on the cluster. b. Register the license using the `FvLicenseUtil` utility.

Date

Step	Actions
1	For registering a user on *Windows*-based computer **B** see section Calculations on a user computer.
2	Registering a user on the cluster, which has not GUI, is not required.

Preparing a project

See section No network connection between user computer and calculation computer.

Starting solve

Step	Actions
1	Start **License Manager**.
2	Run the project on the computation using the command line.

Displaying results

See section No network connection between user computer and calculation computer.

4.9.4 Generating the diagnostics information

There are 2 methods for generating the diagnostics information: using either **Configurator** or the `FvDiagUtil` utility.

Generating the diagnostics information using Configurator

See section Configurator's tab "Support".

Generating the diagnostics information using the FvDiagUtil diagnostic utility

In some situations generating the diagnostics information using **Configurator** is impossible or difficult:

- in environments without graphical user interface (for example in *Linux*)

- on computers where installation of **Configurator** is unreasonable (for example on a computer with **License manager**)

Use in such situations the **FvDiagUtil** utility (file **FvDiagUtil** in *Linux* or **FvDiagUtil.exe** in *Windows*, it is located in the root of the directory where *FlowVision* is installed). Also, the **FvDiagUtil** utility has extended functionality compared to **Configurator**. The utility:

- stores in the archive a log of its own operation, which provides information about presence of files in the diagnostic archive and time of the archive's creation
- stores in the archive **Solver**'s logs
- outputs into a file not only the structure of the directory where *FlowVision* is installed, but also the directory with settings and logs of server modules
- stores in the archive all files from the installation directory with extensions **cft** and **def**
- can enable or disable writing into diagnostics log information from separate modules

Specific features of the **FvDiagUtil** utility are:

- the utility is started using a command console
- it is cross-platform
- it is independent on *Linux*'s graphical libraries

Under *Windows*, the **FvDiagUtil** utility is to be run as Administrator.

Parameters of the FvDiagUtil diagnostic utility

Parameters in the command line	Description
The parameter, which is not preceded by a key	The path and filename of the file where the archive will be saved (see example). If this parameter is not specified, then the utility attempts to save the archive near itself in a file with the name: **fvsupp_ddmmyyhhss.zip** where **ddmmyyhhss** is the time when the utility started.
-sb ddmmyyyy *)	**ddmmyyyy** - starting date for archiving the logs
-se ddmmyyyy *)	**ddmmyyyy** - ending date for archiving the logs
-dbgon **)	Enable the debug log in settings of **Solver**, **Solver-Agent**, **License Manager**, **MPM-Agent** and **Retranslator**.
-dbgoff **)	Disable the debug log in settings of **Solver**, **Solver-Agent**, **License Manager**, **MPM-Agent** and **Retranslator**.

*) If the **FvDiagUtil** utility runs without **-sb** and/or **-se** key, then only the last 10 MB of the **Solver**'s log data are recorded into the log. If you wish to log more data, use **-sb** and/or **-se** key(s).

) After use of these commands a messages are output into the console with information about which settings were changed or not (for example if a module is not installed or a file is inaccessible). After changing a configuration file of any module, the message will be displayed: "Please restart *<name of the module>* to apply new settings**". When the utility is started from *Windows* it makes no output into the console. When you need to read this information, redirect the console output into a file, for example: «**FvDiagUtil.exe > log.txt**».

A date, specified in parameters with keys **-sb** and **-se**, indicates file modification date. In most versions of *Linux* this date is the date of the last change of the file status, which when the file is created, but it also occurs whenever the user writes or sets inode information (for example, changing the file permissions).

Examples of use of parameters of the FvDiagUtil diagnostic utility

FvDiagUtil /home/user/myarchive.tar.gz

Save the archive in the file **myarchive.tar.gz** in the directory **/home/user/**.

FvDiagUtil -sb ddmmyyyy

Archive all logs starting date **ddmmyyyy** up to the current time. The archive is saved in a file with the default name near the utility.

FvDiagUtil -sb ddmmyyyy -se ddmmyyyy

Archive all logs with creation date from **sb** to **se**. The archive is saved in a file with the default name near the utility.

4.10 Parallel computations in FlowVision

In these sections the technologies that are used in *FlowVision* for parallel computation are described.

Understanding of specifics of this technology allows you to configure and use *FlowVision* with maximal efficiency.

See sections:

- Architecture of the processor-memory system
- Parallel calculations in UMA
- FlowVision and Hyper-Threading Technology (HTT)
- Parallel calculations using MPI in systems with NUMA
- Hybrid approach to parallel calculations
- Scalability and reasons of bad scalability
- Some errors in parallel start of FlowVision
- Diagnostics of errors of parallel start of FlowVision

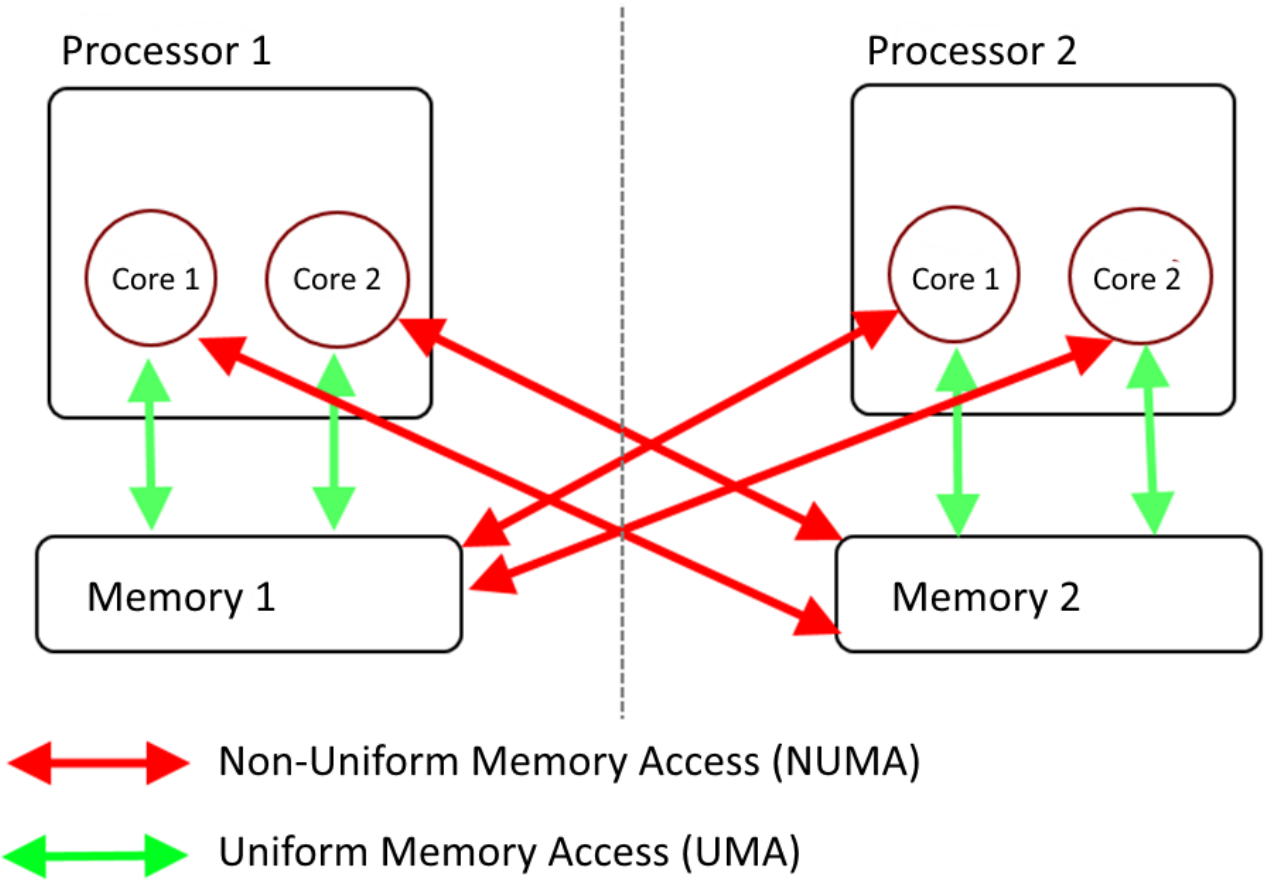
4.10.1 Architecture of the processor-memory system

Parallel computation is done on two types of hardware systems:

- With uniform memory access (UMA - uniform memory access)
- With non-uniform memory access (NUMA - non-uniform memory access)

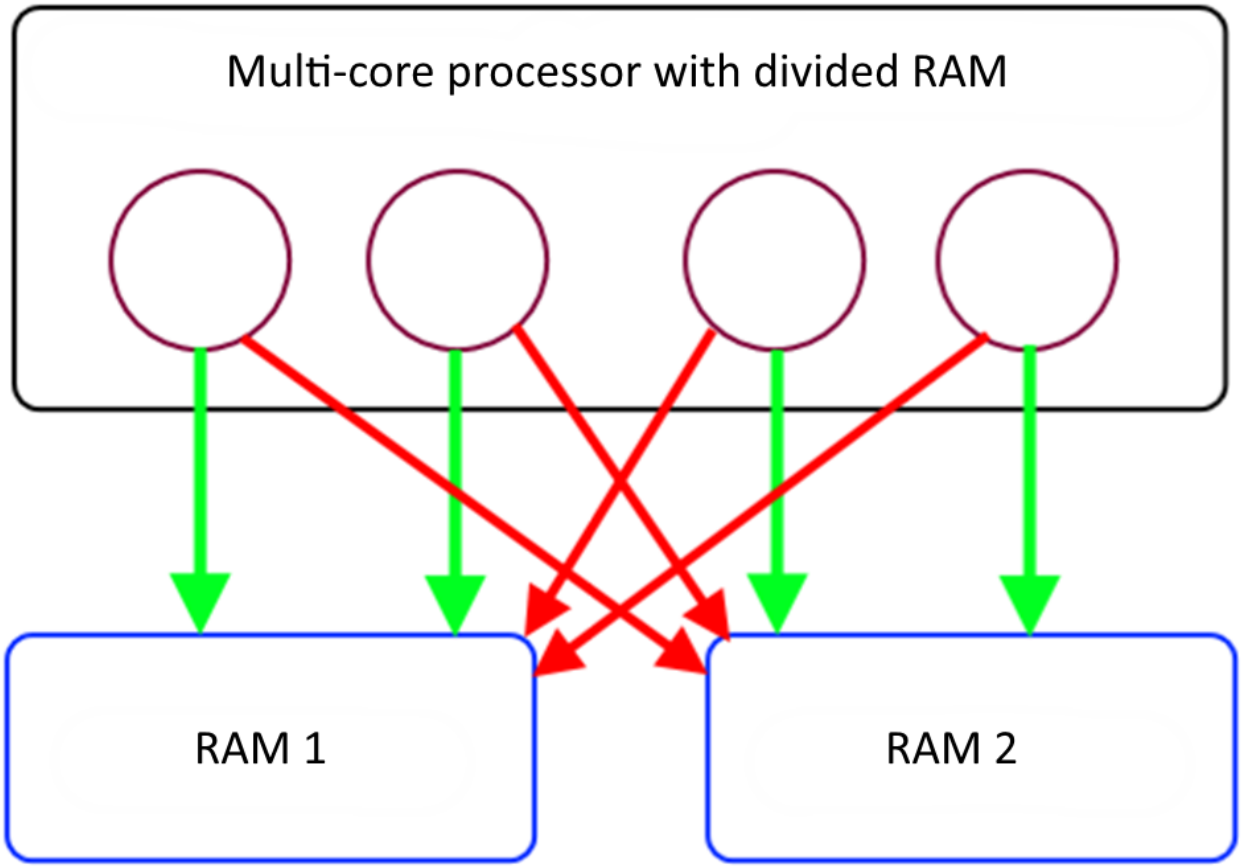
In a system with uniform memory access, access speed to the RAM (random access memory) is the same from any computational core.

In a system with non-uniform memory access, there are several adjacent groups of cores/processors having shared RAM memory. However, the access to different blocks of the shared memory is not identical.



A system with heterogeneous random access memory, consisting of two processors, each of them contains two computational cores

Situations are possible, when even cores from the same processor have non-uniform memory access:



Single-processor system with non-uniform memory access

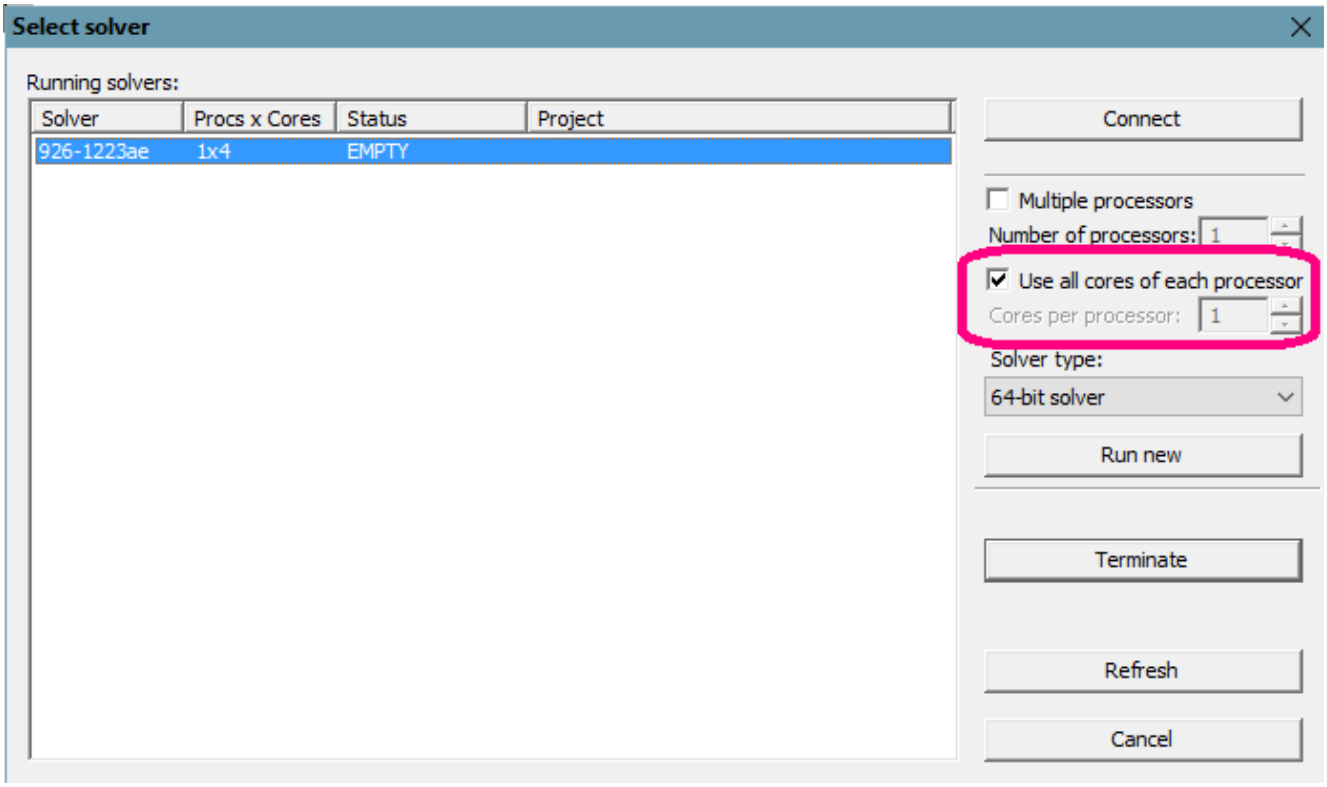
4.10.2 Parallel calculations in UMA

Within a single processor with a uniform memory access (UMA) the most effective way of parallelization is the parallelization on shared memory using threads. This technology provides a uniform and constant load on all CPU cores, the balancing of the computational load over multiple cores is done automatically.

To improve the efficiency of this method of parallelization, it is necessary to bind computing threads to specific cores, i.e. do not permit the flows fluctuate from core to core. This can be done by transferring to **Solver**, at its start, the following parameter:

```
affinity=true
```

When you start **Solver** using **Solver-Agent** with the option **Use all cores of each processor** (see illustration), this parameter is enabled automatically.



Running **Solver** with automatic detection of the number of cores in a single *NUMA* node

When you start **Solver** from a command line, the number of threads is specified using the parameter **`threads`**:
`threads=n` where ***n*** is number of threads

When you start **Solver** using **Solver-Agent** with the option **Use all cores of each processor**, **Solver-Agent** automatically appends the parameter **`threads=numa`** into the command line for starting **Solver**. In such situation the optimal number of cores is determined using the library *NUMA*. Also an automatic binding of threads to cores is turned on (the **`affinity=true`** parameter is not mandatory in this situation.

4.10.3 FlowVision and Hyper-Threading Technology (HTT)

Hyper-Threading Technology (*HTT*) as a technology from *Intel*, which increases the efficiency of processors. When the hyper-threading is turned on, the physical processor core is divided into two logical cores (operating system displays a double number of cores).

When using hyper-threading, you should be aware of its features:

- Each logical core captures one parallel license. So doubling the number of cores results in doubled consumption of licenses.
- Use of hyper-threading gives an insignificant performance gain in tasks for numerical hydrodynamics simulation (about 10% depending on the used processor and the simulation task).

The results of investigations have also demonstrated that partial use of the processor with the hyper-threading is enabled might either increase or decrease the speed of the computation.

To obtain a maximal benefit from the hyper-threading, the program should use all logical cores. In case of lack of parallel licenses it is advisable to disable hyper-threading, if the loss of 8-10% percent of the processor time is not critical.

Important note: at partial load on logical cores (with hyper-threading enabled), the loss of time can be critical if there are other processes except *FlowVision*, that compete for CPU time. This is caused by the fact that hyper-threading does not double the computational power of the CPU, but only allows use of CPU time during forced outage of one of the threads (for example, when waiting of receiving information from the RAM memory).

4.10.4 Parallel calculations using MPI in systems with NUMA

In systems with non-uniform memory access it is necessary to use parallelization with *MPI*, since parallelization using threads becomes extremely inefficient.

When parallelization is done using *MPI*, the computational grid is allocated for processors (it is geometric decomposition of the grid). Thus, queries to the slow distributed memory occurs only for data exchange on boundaries of parts of the grid but not randomly for any cell as in the case of parallelization using threads.

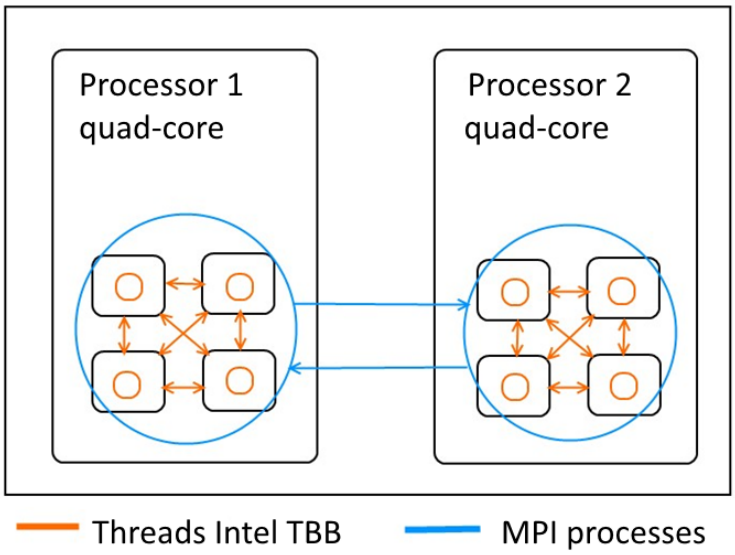
Running *FlowVision* using *MPI* is fundamentally similar to running any other program using *MPI*.

4.10.5 Hybrid approach to parallel computations

In *FlowVision* were taken into account the advantages and disadvantages of the two methods of parallelization, resulting in a hybrid approach, which has been implemented.

For multiprocessor computations *MPI* is used, and at the same time within each processor (inside a *NUMA* node) threads are used.

This hybrid approach can significantly reduce the computation time compared to the solution, which uses *MPI* only.



Scheme of parallelization used in *FlowVision*

4.10.6 Scalability and reasons of bad scalability

Scalability is the property of a system, which characterize how rate of computation depends on increase of the number of processors.

Ideally, the scalability is linear, so how many times we increase the number of processors, as many times the computation accelerates. In reality, there are many of factors that affect the scalability:

- The grid is poor.
- Dimension of the problem does not correspond to the number of processors (if number of cells per processor is not optimal).
- Poor architecture of the computing system.
- Errors of setting up the running mode of *FlowVision*.

4.10.7 Some errors in parallel run of FlowVision

The following errors are possible when *FlowVision* runs in parallel mode:

1. Threads are not linked to nuclei. Run **Solver** with the parameter `affinity`.
2. Threads request to the "foreign" memory (this is possible when, within a single processor, non-uniform access to the memory exists, and the NUMA library is not supported by the processor). Provide running of threads only with uniform access to the memory.
3. Queue management system makes incorrect allocation of *MPI* processes:
   a. *MPI* processes runs not by one per processor, but by one per core.

b. *MPI* processes runs not by one per processor, but by one per node.
c. *MPI* processes runs not by one per processor, but by several per processor.
d. One node is used by multiple programs simultaneously (the resource had not been locked and was granted to several processes simultaneously).

4. The **`libFvMPI.so`** library used, which does not comply to used version of *MPI* (see sections Specific system requirements for Linux, Preparing Solver for work with an installed MPI, Compilation a library for interconnection with various MPI implementations).
5. When using **`threads=numa`** , the number of cores is determined incorrectly (when running with the option "**Use all cores of each processor**"), the processor does not support the NUMA technology or incorrect version of the NUMA library is used).
6. Conflict between configuration of *MPI* and **Solver**'s parameter **`mpi_skip`**.See details in section Methods of running FlowVision in a queuing system.
7. If the number of processors is not specified in a line in the machine MPI file, then *MPI* processes are run on only the first processors of the appropriate computers. So, if a node has multiple (more then one) processors, then you should explicitly specify in the machine file the number of processors (see details and examples in the section Machine MPI file).

4.10.8 Diagnostics of errors of parallel start of FlowVision

The main source of information for analysis of correctness of *FlowVision*'s run in parallel mode is the *MPI* log. Access to the *MPI* log is described in the documentation of your *MPI* or the queuing system.

Information about run mode of **Solver** is located in the first lines of the *MPI* log. Here is an example of such a log:

```
dirname = /opt/software/FlowVision/30803
Initialize MPI
dirname = /opt/software/FlowVision/30803
dirname = /opt/software/FlowVision/30803
Initialize MPI
Initialize MPI
dirname = /opt/software/FlowVision/30803
Initialize MPI
Setting signal handlers
Setting signal handlers
Setting signal handlers
Setting signal handlers
Setting affinity
Setting affinity
Setting affinity
Setting affinity
1: Setting NUMA affinity to node 1 of 2 available
2: Setting NUMA affinity to node 0 of 2 available
0: Setting NUMA affinity to node 0 of 2 available
3: Setting NUMA affinity to node 1 of 2 available
Proc0 Processor name = node6-168-15 Pid 8618 8 threads
Proc3 Processor name = node6-169-05 Pid 16565 8 threads
Proc1 Processor name = node6-168-15 Pid 8619 8 threads
Proc2 Processor name = node6-169-05 Pid 16564 8 threads
Create a communicator thread
Create a communicator thread
Create a communicator thread
```

You have to analyze the following information from the *MPI* log:

1. The number of lines starting from **`Proc0`**, **`Proc1`**, **`Proc2`**, etc. must match the number of ordered processors (number of *MPI* processes).
2. All lines starting from "**`Proc`**" must have unique numbers after "**`Proc`**". If some of these lines have the same identifiers, for example, **`Proc0`**, **`Proc0`**, **`Proc0`**, etc., it likely means that of the version of the library **`fvlibmpi.so`** does not match to the version of used *MPI*.
3. A line of the form "**`Proc0 Processor name = node6-168-15 Pid 8618 8 threads`**" contains the following information:
   a. Node on which the current *MPI* process has been started, in our example, this is the node with the host name **`node6-168-15`**. If on a node the number of running MPI processes is more than the number of core blocks with shared memory, it means that this run mode is not optimal.

b. The number of threads that have been started in this *MPI* process. In our example, this is 8 threads. The number of threads must match the number of nuclei with shared memory.

4. Lines "**`Setting affinity`**" mean that threads have been linked to cores.

The above recommendations cannot be applied when the **Solver**'s parameter **`mpi_skip`** has been used; number of *MPI* processes will not match the number of active *MPI* processes, see details in section Methods of running FlowVision in a queuing system.

Additionally, it is necessary to inspect the contents of the **`sta`** file, which is created in the process of the project's computation in *FlowVision* and locates in the server part of the project.

Besides other information (see section Format of sta files), this file provides statistics of memory usage.

```
================================ Iter number 0 ==================================
CALCULATION TIME STATISTICS
Iter calculation time:    7.29107
Total calculation time:   26.9015

SOLVING STATISTICS
Tcur:    1e-05
Texpl:   3.86158e-07
Tstep:   1e-05
Nstep:   0

SYSTEM STATISTICS
          PhM_total   PhM_free   Proc_mem   Peak_mem   CPU_time   Wait_CPU Wait_Wall
Proc0    :  24158.92  21874.82     454.01     667.28      37.42       0.31       0.15
Proc1    :  24158.92  21874.82     461.93     899.84      36.58       0.50       0.31
Proc2    :  24158.92  21815.04     503.08     936.84      34.26       0.89       0.78
Proc3    :  24158.92  21815.04     453.79     646.12      34.74       0.75       0.70
Proc4    :  24158.92  21838.04     500.18     800.84      35.66       0.80       0.77
Proc5    :  24158.92  21838.04     493.11     750.12      34.11       0.71       0.69
...
```

In the example above the computation was done on 6 processors. There were two processors per each node. In the log section **`SYSTEM STATISTICS`** you see information about the memory consumption:

- The column **`PhM_total`** is equal for each processor, because each node has the same amount of RAM memory.
- The column **`PhM_free`** has three pairs of equal values. This is because the amount of free memory is determined for a whole node rather than for a separate processor.

An incorrect run might be indicated by an excessive number of absolutely identical values in the column **`PhM_free`**, as in the example below:

```
SYSTEM STATISTICS
          PhM_total   PhM_free   Proc_mem   Peak_mem   CPU_time   Wait_CPU Wait_Wall
Proc0    :  24158.92  21274.82     454.01     667.28      37.42       0.31       0.15
Proc1    :  24158.92  21274.82     461.93     899.84      36.58       0.50       0.31
Proc2    :  24158.92  21274.82     503.08     936.84      34.26       0.89       0.78
Proc3    :  24158.92  21274.82     453.79     646.12      34.74       0.75       0.70
Proc4    :  24158.92  21838.04     500.18     800.84      35.66       0.80       0.77
Proc5    :  24158.92  21838.04     493.11     750.12      34.11       0.71       0.69
```

These statistics means that four *MPI* processes ran on the same node, and remaining two 2 *MPI* processes ran on the other node, which is not optimal, because one node has two but not four processors.

4.11 Specifics of deploying FlowVision on clusters

See sections:

- Network interoperations between modulus. Requirements to settings of FlowVision's environment.
- Use of Retranslator
- Methods of running FlowVision in a queuing system
- Multiuser installation FlowVision on cluster

4.11.1 Requirements to settings of FlowVision's environment.

There are following requirements for configuring the operating system and *FlowVision*:

1. Client modules need to know where (IP address and port) are **License Manager** and **Solver-Agent**, and should have access to them via the *TCP/IP* protocol.
2. After **Solver-Agent** tells where **Solvers** locate (on which cluster nodes, defined by IP and a port), client modules must have access to **Solvers** via the *TCP/IP* protocol.
3. If some modules are in a hidden network (for example, **Solvers** on the computational nodes can be not accessible from Internet, as computing nodes, usually locate in a separate local network of the cluster without access to Internet), then they are accessible via the module **Retranslator** module, which actually is a software proxy server.
4. **Solvers** must know where (IP address and port) **Solver-Agent** and **License Manager** locates and must have access to them via the *TCP/IP* protocol.
5. There should be organized a *SSH* access (with no passwords) for the operating system's user, as who **Solver-Agent** runs. This is a standard requirement for *MPI* computations.
6. For **Solvers** it is necessary to open (for connections from client modules) many ports starting from **PPPBasePort** (its default value is **11000**). When **Solver** starts, when it is waiting for a connection from a client module, it looks for any free port, which it can use, starting from the port **PPPBasePort**.

Also there are requirements to directories and files of *FlowVision*:

1. When using *FlowVision* on a cluster, **Solver** and **Solver-Agent** must be installed into a directory that is mounted on all computing nodes of the cluster. This directory and its contents must be available from all computing nodes of the cluster by the same path.
2. A directory with configuration files of server modules must be mounted on all computing nodes and be available from all computing nodes by the same path. This directory must be accessible for reading and writing for the operating system's user, as which **Solver-Agent** runs.
3. The system temporary directory must be accessible for reading and writing for all operating system's users by which *FlowVision* modules run.
4. The directory with calculation data must mounted on all computing nodes and be available from all computing nodes by the same path. Access to this directory for reading and writing must be granted for the operating system's user, as which **Solver-Agent** runs. This directory should be created by a user and later, during configuring **Solver-Agent**, be specified as a server directory project (the directory for server parts of projects). See section Using Solver-Agent.

4.11.2 Use of Retranslator

The **Retranslator** module (`FvConnect`) is actually a software proxy server and allows redirecting traffic from **Solvers** to **License Manager** and client modules, if these modules locate in a network, which is not accessible from the computing nodes.

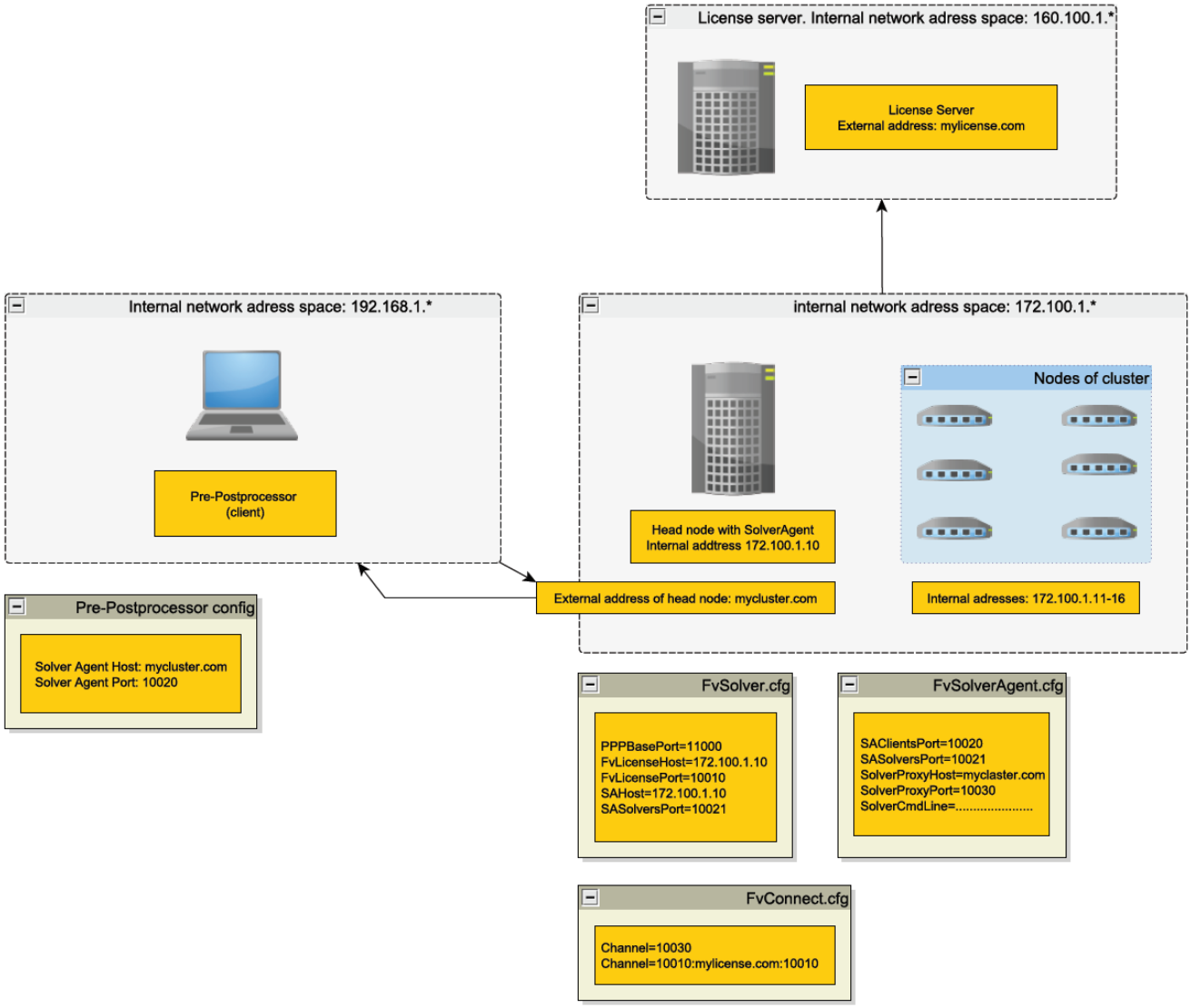
As a rule, the head node of the cluster has access to the Internet. **Retranslator** should be installed on this node.

In the settings of **Retranslator** each new line defines a method for traffic forwarding between modules.

To configure the redirection of traffic between **Solver** and client modules it is also required to edit settings of **Solver-Agent**.

For details about configuring **Retranslator** see in section Configuration file of Retranslator (FvConnect.cfg).

The diagram below gives an example of configuration of **Retranslator** and other *FlowVision* modules when the software package is deployed on a cluster:



If the computational nodes do not have a direct way out to the external world, then it is impossible to connect a client module (**Viewer** or **Pre-Postprocessor**) to **Solver** without **Retranslator**.

The connection between a client module and **Solver** without **Retranslator** is done as follows:

1. The client module connects to **Solver-Agent**.
2. The user selects a **Solver** from the list provided by **Solver-Agent**.
3. **Solver-Agent** sends the IP address and port for connection the client module to **Solver**.
4. If **Solver** locates in a closed network, the client module will not be able to find it based on the data received from **Solver-Agent**.

When **Solver-Agent** is configured, connection between a client module and **Solver** is done as follows:

1. The client module connects to **Solver-Agent**.
2. The user selects a **Solver** from the list provided by **Solver-Agent**.
3. **Solver-Agent** sends the IP address and port for connection to **Retranslator**. Also, the client module receives from **Solver-Agent** the IP address and port of the **Solver** in the closed network.
4. As **Retranslator** is running on the node, which has access to the global network, the client module with no problem connects to **Retranslator** and gives to it the IP address and port of **Solver**.
5. **Retranslator** begins to transmit traffic between the client module and **Solver**.

4.11.3 Methods of running FlowVision in a queuing system

There are several approaches to start *FlowVision*:

1. **Solver** is started using **Solver-Agent**. Then **Pre-Postprocessor** is connected to **Solver**. A project is loaded on **Solver**, then user runs the project for computation.
2. **Solver** is started by the user from a script, which contains commands for automatic loading some project on **Solver** and for automatic starting the **Solver** for computation.

The first method is convenient when resources of the cluster are entirely under the control of the user.

The second method is convenient when competition for the resources exists (queues are used). In this case, we recommend to make a fully automatic start of the computation.

Running Solver through Solver-Agent

Running **Solver** through **Solver-Agent** is made directly by **Solver-Agent** using a line for starting **Solver**.

This line can be seen in the configuration file of **Solver-Agent**:

```
SolverCmdLineMPI=mpiexec -n %1 FvSolver64.exe;64-bit MPI solver
```

In this line, symbols `%1` will be replaced by the number of processors, which has been ordered in the interface of *FlowVision* client modules.

This line can be rewritten in another way:

```
SolverCmdLineMPI=myscript.sh %1;64-bit MPI solver
```

Here, instead of *MPI*, we start the script `myscript.sh` and pass to it the parameter `%1` (which, as mentioned before, will be replaced by the number of processors).

And in the script `myscript.sh` it is possible to do many things, for example, it is possible to select a specific *MPI*, to reserve the required number of nodes in a cluster, to send the task to the queue management system, etc.

Important note: no matter what a kind is of the line for starting a **Solver**, **Solver-Agent** automatically adds to the end of the line the following parameters:

- `threads=n`, where **n** is the number of ordered cores or the word `numa` (for automatic selecting the number of cores). This parameter is to be transferred to the **Solver**, to inform it how many threads must be started.
- `sa_ID=XXX-XXXXXX`, which is a unique identifier of the started **Solver**. This parameter is to be transferred to the **Solver**, to distinguish this **Solver** from other **Solvers**.

After its start, a **Solver** connects to **Solver-Agent** and passes its `sa_ID`. **Solver-Agent**, after receiving this `sa_ID` and having in memory, which user has started this **Solver**, transfers to this **Solver** the user's license name (`sa_license`) and the user's name at **Solver-Agent** (`sa_user`).

If a `myscript.sh` script is used, it must contain all required parameters to pass them to the **Solver**.

Example of starting Solver using *SLURM* and Solver-Agent

Sample of the `myscript.sh` script:

```
#!/bin/bash
# Variable CPUS is 1% from a start line in the settings of Solver-Agent
CPUS=$1
# If a node has 2 processors, then the number of reserved nodes = CPUS/2. In the expression
  below the bc calculator program is used.
NODES=`echo "$1 / 2" | ./bc`
# uncomment the line below, if Solver cannot find its libraries or there is a conflict with
  libraries in the operating system
# export LD_LIBRARY_PATH=/home/users/myuser/FlowVision3.09/lib64
# here we start the queue system module, which places the task into the queue,
# where -p, --sockets-per-node, and --ntasks-per-socket are parameters of this module
sbatch -p myqueue -N $NODES 2 --sockets-per-node=2 --ntasks-per-socket=1
  mpirun /home/users/myuser/FlowVision3.09/FvSolver64 $2 $3 $4 $5 $6 $7 $8 $9
```

In the last line **Solver** receives variables `$2 - $9`, which are all possible variables that were has been received **Solver-Agent** (they were transferred to the script `myscript.sh`). Actually, the number of transferred variables is less, but we add more variables then required to ensure that the script will work even if the future versions of *FlowVision* contain new hidden parameters to control **Solver**.

Using such scripts, you can create quite complex logic for starting **Solver**, you can even select a queue on the cluster, depending on the username of the **Solver-Agent**'s user.

The main advantage of this method of starting, is that the user is able not to have a contact with the *Linux*' command to start **Solver**. All the work will be take place through the user interface of *FlowVision.*

But this method of starting **Solver** has substantial disadvantage. If there is competition on the cluster for computing resources, then the **Solver**'s startup script goes into a queue and the can not load a project on the **Solver** and run the project for computation. The user will have to wait until **Solver** starts after moving through the queue. A situation is possible when **Solver** will start at the night, outside the user's working hours. It will be idle

while waiting for the command for loading a project and running the project, wasting the computing resources of the cluster. Therefore, in an environment with high competition for the computing resources it is recommended not to use this method of starting **Solver**.

Another example of starting Solver (using *TORQUE*, *openmpi* and Solver-Agent)

The general principle of operation:

1. Using **Solver-Agent** start the **sa_run.sh** script and transfer many parameters to it.
2. In the script **sa_run.sh** a script file will be formed for *TORQUE* and *qsub* will be started, transferring to it the script **qsub_script.sh**.
3. *qsub* will execute the script **qsub_script.sh**.

Line in the settings of **Solver-Agent** (in the file **FvSolverAgent.cfg**):

```
SolverCmdLineMPI=sa_run.sh -np %1;torque solver
```

Script **sa_run.sh**:

```
#!/bin/sh
##############
# Here you can place loading of required libraries and modules, and defining the necessary
  environment variables
# For example, load a module with MPI
module load openmpi/1.6.5
# For example, specify the path to the FlowVision libraries
export LD_LIBRARY_PATH=/shared/flowvision/FlowVision3.0/lib64:$LD_LIBRARY_PATH
##############
##############
# prepare the necessary variables
FVID=`date +%d%m%H%M%S`
PROCS=$1
# assume that there are two processors per node
PROCSPERNODE=2
# Define how many nodes have to be ordered for the specified number of processors. Use the bc
  calculator program for this.
NODES=`echo "${PROCS:-2} / $PROCSPERNODE" | bc`
##############
##############
# here the qsub_script.sh script file for qsub is formed
echo '#!/bin/sh' > qsub_script.sh
# here the home directory is specified
echo '#PBS -d /share/flowvision/FlowVision3.0' >> qsub_script.sh
# here the name of the task is specified
echo "#PBS -N fv.$FVID" >> qsub_script.sh
# here the resources we need are specified; here nodes is the number of nodes, ppn is the
  number of processors per node
echo "#PBS -l nodes=$NODES:ppn=$PROCSPERNODE" >> qsub_script.sh
# configuring of output for MPI logs
echo "#PBS -o ./$FVID.out" >> qsub_script.sh
echo "#PBS -e ./$FVID.err" >> qsub_script.sh
# forming a line for starting Solver, here $2 - $9 are all the possible parameters, that might
  be transferred to the sa_run.sh script from Solver-Agent
# PBS_NODEFILE is the variable of torque, which contains a list of the selected nodes
echo "/share/mpi/openmpi-1.6.5/bin/mpirun -np $PROCS -machinefile
  \${PBS_NODEFILE} ./FvSolver64 $2 $3 $4 $5 $6 $7 $8 $9" >> qsub_script.sh
##############
##############
# granting permissions to the created script for execution
chmod +x qsub_script.sh
# run qsub
qsub qsub_script.sh
##############
```

Here is an example of a final script **qsub_script.sh**, generated as a result of the work of the above script started through **Solver-Agent** where 8 processors were specified:

```
#!/bin/sh
```

```
#PBS -d /share/flowvision/FlowVision3.0
#PBS -N fv.1009124949
#PBS -l nodes=4:ppn=2
#PBS -o ./1009124949.out
#PBS -e ./1009124949.err
/share/mpi/openmpi-1.6.5/bin/mpirun -np 8 -machinefile ${PBS_NODEFILE} ./FvSolver64 sa_ID=910-
  1249ez threads=numa
```

Starting Solver using a script

In some cases it is convenient to use starting **Solver** without **Solver-Agent**. As was mentioned above, this is especially important in the case when there is competition in the queue on the cluster.

This start-up method uses the following principle of work:

1. Script starts and receives parameters for **Solver**'s startup.
2. The script handles the parameters and starts **Solver**, and **Solver** will automatically load the project and begin to calculate using the batch mode of **Solver**.

Example of the line starting the script runsolver.sh:

```
./runsolver.sh 8 queue1 7 /home/projects/rotor
```

where:

`8` is the number of nodes

`queue1` is the queue in the queue management system

`7` is the mode of starting **Solver** (`0` means start from the beginning, `7` means start for continuation)

`/home/projects/rotor` is the directory with the project, which is to be launched

Sample of the script `runsolver.sh` to start Solver within *SLURM* without Solver-Agent:

```
#!/bin/bash
# giving meaningful names to the received parameters
NODES=$1
QUEUE_PBS=$2
START_MODE=$3
PROJECT_DIR=$4
# Prepare a batch file of Solver, which will be located in a directory called $PROJECT_DIR and
  has the name.commandfile-auto
# Defining a name for the project, located in this folder with the project (in a folder only
  one project can locate). Write into the batch file the command for loading the project on
  Solver
echo "SS_PROJECTLOAD<"`find $PROJECT_DIR -name "*.fvproj" -exec echo {} \;`"><><><>" >
  $PROJECT_DIR/.commandfile-auto
# Write the command for start indicating the start mode: 0 - from beginning; 7 - for
  continuation.
echo "SS_SOLVESTART<$START_MODE >" >> $PROJECT_DIR/.commandfile-auto
# Write the command that makes Solver compute and do not respond to external commands, until
  stopping condition is satisfied in the parameters of the project itself
echo "SS_DO_NOT_DISTURB" >> $PROJECT_DIR/.commandfile-auto
# The command that saves the data after stopping of the computation
echo "SS_PRJDATASAVE" >> $PROJECT_DIR/.commandfile-auto
# The command that finishes the work of Solver and unload it from the memory
echo "SS_SHUTDOWNSOLVER" >> $PROJECT_DIR/.commandfile-auto
# uncomment the line below, if Solver cannot find its libraries or there is a conflict with
  libraries in the operating system
# export LD_LIBRARY_PATH=/shared/flowvision/FlowVision3.0/lib64
# Run the task in the specified queue on the specified number of nodes
sbatch -p $QUEUE_PBS -N $NODES --sockets-per-node=2 --ntasks-per-socket=1
    mpirun /shared/flowvision/FlowVision3.0/FvSolver64 sa_ID=`date | sed s/\ //g`
    sa_user=sausername sa_license=licensename threads=numa cmdfile=$PROJECT_DIR/.commandfile-
    auto
```

As a result, the task is placed into the queue `QUEUE_PBS` of the *SLURM* queue system on the specified number of nodes. For autonomous work of **Solver** it is necessary to specify several parameters:

- `threads=n`, where **n** is the number of ordered cores or the word `numa` (for automatic selecting the number of cores). This parameter has to be transferred to **Solver**, to inform it how many threads must be started.

- **sa_ID=XXX-XXXXXX**, which is a unique identifier of the started **Solver**. This parameter has to be transferred to a **Solver**, to distinguish this **Solver** from other **Solvers**.
- **sa_user**, which is the username of the **Solver-Agent**'s user
- **sa_license**, which is the license username

For the above example the following **Solver**'s command file **commandfile-auto** will be generated:

```
SS_PROJECTLOAD<"/home/projects/rotor/rotor.fvproj"><><><>
SS_SOLVESTART<7>
SS_DO_NOT_DISTURB
SS_PRJDATASAVE
SS_SHUTDOWNSOLVER
```

When it is run with this batch file, **Solver** will be displayed in **Pre-Postprocessor** with the **UNAVAILABLE** status. This is because of the fact that **Solver** can not accept commands from the client modules until a stopping condition is satisfied as it has been defined in the project. Therefore, this script allows you to start **Solver**, view it in **Terminal** and view data using **Viewer**, but not to connect from **Pre-Postprocessor**.

To provide the possibility to connect to the project from **Pre-Postprocessor**, you can create such a file, which would imply stopping **Solver** only by the command from the user; the contents of such a file is given below:

```
SS_PROJECTLOAD<"/home/projects/rotor/rotor.fvproj"><><><>
SS_SOLVESTART<7>
```

In this case, **Solver** will not stop until receives an explicit instruction from **Pre-Postprocessor** or **Terminal**.

A situation is possible when **Solver-Agent** is not used at all neither for starting **Solver** nor for viewing a list of running **Solvers**. All computations are started from the command line, and analysis of the results is done by methods that do not imply a direct connection of the *FlowVision* modules to **Solver** on the cluster. In this case, there is no need to specify for **Solver** parameters that relate to **Solver-Agent**. But you have to specify the parameters relating to licensing. Below we give an example of the last line from the script **runsolver.sh** for the case when **Solver-Agent** is not used and is not running on the cluster:

```
# Running the task in the specified queue on the specified number of nodes
sbatch -p $QUEUE_PBS -N $NODES --sockets-per-node=2 --ntasks-per-socket=1
  mpirun /shared/flowvision/FlowVision3.0/FvSolver64 sa_ID=`date | sed s/\ //g`
  sa_license=licensename threads=numa cmdfile=$PROJECT_DIR/.commandfile-auto
```

Specifics of running Solver in a cluster without division into processors and cores

In some clusters the queue management system allows you to order a certain number of processors. *PBS* is configured there so that each core is considered as one separate processor, so an *MPI* process is run on each core of a processor.

From the *FlowVision*'s point of view this is not an optimal starting method, because parallelization using threads is more efficient then *MPI*, which is used in this method.

In *FlowVision*'s **Solver** a special mode of operation is implemented, which provides work with settings of the queue management system.

Suppose that we run **Solver** on two processors and each processor has four cores. Because of the settings of the queue management system we have to order 2x4 *MPI* processes. To use the hybrid parallelization for **Solver**, it is necessary to set the **mpi_skip** parameter. In our example it looks like this:

```
FvSolver64 mpi_skip=4 threads=numa
```

When **mpi_skip=4** is set, the work with **Solver** will go according the following scenario:

1. We order from the queue management system a reserve of 8 cores.
2. The queue management system starts 8 *MPI* processes with **Solver** on 8 cores.
3. **Solver** reads the parameter **mpi_skip=4** and allocates data for each one of the four *MPI* processes, while other *MPI* processes remain inactive during the whole computation.
4. Two active *MPI* processes (each running on a separate processor) use parallelization of four threads per processor.

4.11.4 Multiuser installation FlowVision on cluster

Rules of setup and installation *FlowVision* imply the presence of a few directories that should be accessible by all users of *FlowVision*. At a minimum, this is the directory with **log**-files and settings.

In some situations, the deployment of applications on a cluster involves installing the application in a read-only directory.

You can configure *FlowVision* so that each user of the cluster (assuming that the user's home directories are mounted on all compute nodes) could use their own unique settings of **Solver** and **Solver-Agent**.

Settings of **License Manager** should not be visible and accessible to the system administrator only.

Possible problems: If *MPI* on the cluster is configured so that threads are linked to the core, on which *MPI* process is running, then it turns out that all the **Solver**'s threads will work on one core, while other cores will be idle. You can detect such situation when analyzing loading of kernels during the computation (or loading of the entire processor, it will correspond to the part 1/N of 100%, where N is the number of cores per processor). Also, in this situation, the calculation in the mode without the pure *MPI* thread is significantly more efficient than in the `mpi_skip` mode with streams.

Methods of storing settings

There are two ways to store settings of *FlowVision* modules:

- indicate a directory with the settings in the file `Fv.cfd`
- indicate a directory with the settings in a `*.cft` file

The first method assumes that all settings of server modules are stored in the same directory, the path to which is specified in the file `Fv.cfd` as an absolute path. The path can not be specified as an environment variable. Thus, this method of storing settings is single-user.

The second method assumes that there is no `Fv.cfd` file and each *FlowVision* module looks near it a file with the same name as the module's file has, and with the extension `cft`, which contains information about location of the module's settings. Also, in the `cft` file, it is possible to specify that the settings are stored in the home directory of the user, as which the module will be running.

Multiuser configuration of Solver

To implement the second operating mode, follow the steps:

Step	Actions
1	Delete the `Fv.cfd` file from the directory with *FlowVision.*
2	Rename files `FvSolver.def`, `FvLicense.def`, `FvSolverAgent.def` etc. as `FvSolver.cft`, `FvLicense.cft`, `FvSolverAgent.cft` etc.

3	Make changes in the `cft` files.
	Into `cft` files of all modules that which will be available to users for configuring, add two lines in the beginning of a file:
	`HOME=SYS`
	`TEMP=SYS`
	Here the `HOME` parameter defines the directory where settings of the appropriate *FlowVision* will be stored, and `TEMP` is the parameter that defines at which directory temporary files created the *FlowVision* module will be stored.

Example of the file `FvSolver.cft`:

```
HOME=SYS
TEMP=SYS
PPPBasePort=11001
FvLicenseHost=172.160.1.1
FvLicensePort=10010
SAHost=172.160.1.1
SASolversPort=10021
DebugLog=No
CompressLayers=Yes
```

The parameters listed below the parameter `TEMP`, are the default parameters. If the user starts **Solver** at the first time, in his/her home directory the `FlowVision` will be created, with subdirectories `settings` and `logs`. In the subdirectory `settings` the `FvSolver.cfg` file will be created, which will contain the following:

```
PPPBasePort=11001
FvLicenseHost=172.160.1.1
FvLicensePort=10010
```

```
SAHost=172.160.1.1
SASolversPort=10021
DebugLog=No
CompressLayers=Yes
```

Thus, the administrator is able to provide multiuser start of correctly preconfigured **Solver**. And a user, if he/she wish, can change settings of **Solver** to use it, for example, with his/her own copy of **Solver-Agent**, which is not available to other users, or connecting an external **License Manager**.

In all `cft` files of those modules that do not assume changing their settings by users, it is necessary to specify, instead of `SYS`, the absolute path to the directory, which is available for the user as which the module will be running.

Example of the file `FvLicense.cft`:

```
HOME=/opt/FVSettings
TEMP=/opt/tmp
FvLicensePort=10010
DebugLog=No
```

5 Quick start

This chapter contains educational examples for the full project's life cycle from its formation to results analysis. As an example, we will consider the simulation of mixing of a liquid in a mixer.

Before you start the exercises, make sure that:

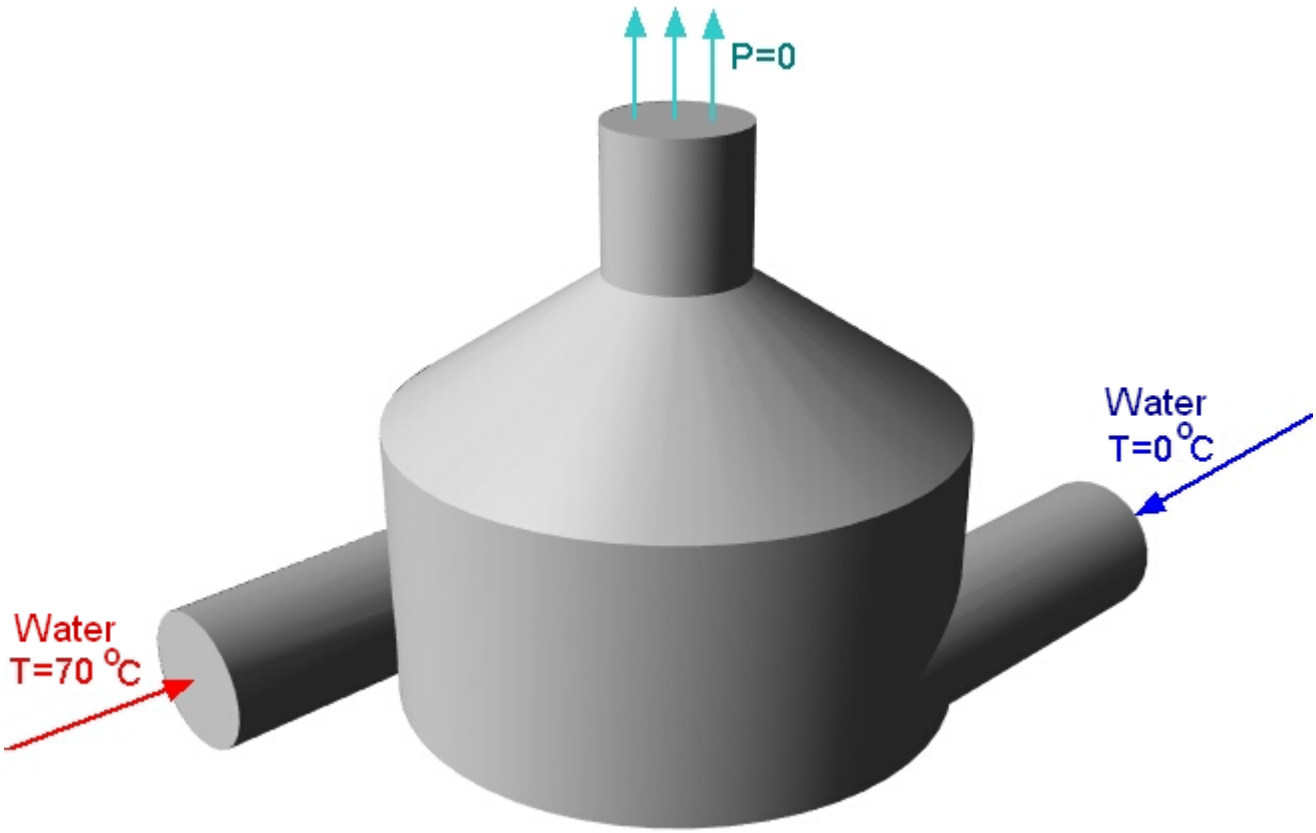
- client and server FlowVision modules have been installed
- **License Manager** is installed
- the license is obtained and registered
- all settings of **License Manager** and **Solver-Agent** are tuned
- **Solver-Agent** and **License Manager** are running
- directories for server parts and client parts of projects have been created
- the **Solver-Agent**'s user have been created (see sections Using Solver-Agent, Connection to Solver-Agent and user authentication on Solver-Agent, Registering a new Solver-Agent user, Registration data (profile) of Solver-Agent's user and their change).

Follow these step-by-step instructions to get the basic experience of creating a *FlowVision* project, including how to run the simulation, viewing the results, and and gaining familiarity with the interface:

- Load a geometry model of the mixer
- Basic operations in the View window
- Cross-section of a geometry model with a Plane
- Specifying physical parameters of the project:
  - Specifying general settings of the project
  - Specifying Substances and their parameters
  - Specifying Phases and their parameters
  - Specifying a Model and its parameters
  - Specifying a Model in the computational domain
  - Specifying Boundary conditions
  - Specifying Initial conditions
- Specifying a computational grid:
  - Specifying the initial grid
  - Specifying an adaptation
- Specifying simulation controls:
  - Specifying the time step
  - Specifying the autosave parameters
  - Specifying the stopping conditions
- Specifying the visualization:
  - Creating characteristics
  - Creating layers
- Starting the project's computation
- Viewing results of the computation:
  - Viewing data in the Monitor window
  - Viewing the layers during the computation
  - Viewing characteristics in the Info window
  - Making an animation

5.1 Example of a problem (mixing a liquid)

Consider the flow of water in a mixer, which is a reservoir with two supply tubes at the base and one outlet tube on the top:



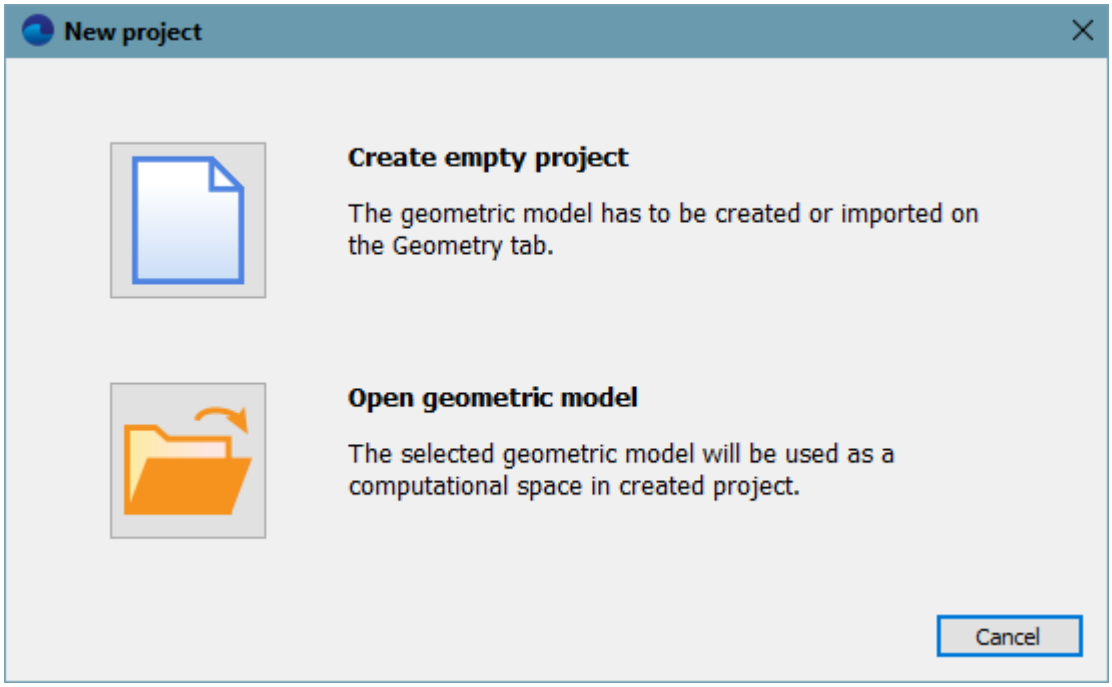
The reservoir's diameter is 0.04m, the height of its cylindrical part is 0.02m, the height of the top cone-shaped part is 0.01m, and the diameter of the tubes is 0.01m. One of the tubes supplies cold water (T= 0°C), and another tube supplies hot water (T=70°C). The flow rate is same for both tubes and it is equal to 0.1 [$kg \cdot s^{-1}$].

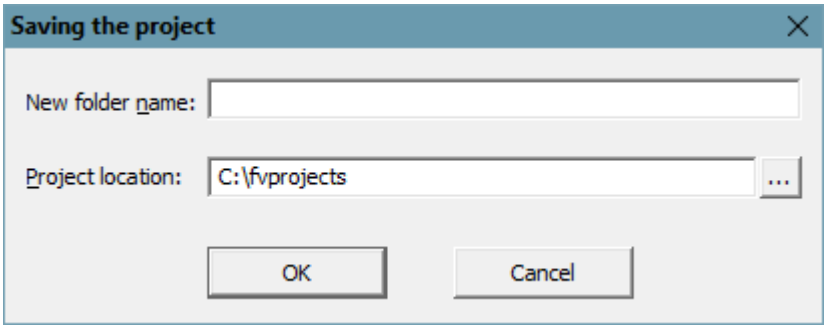
The purpose of the computation is to obtain patterns of flow and temperature equalization of water in the mixer.

5.2 Loading a geometry model of the mixer

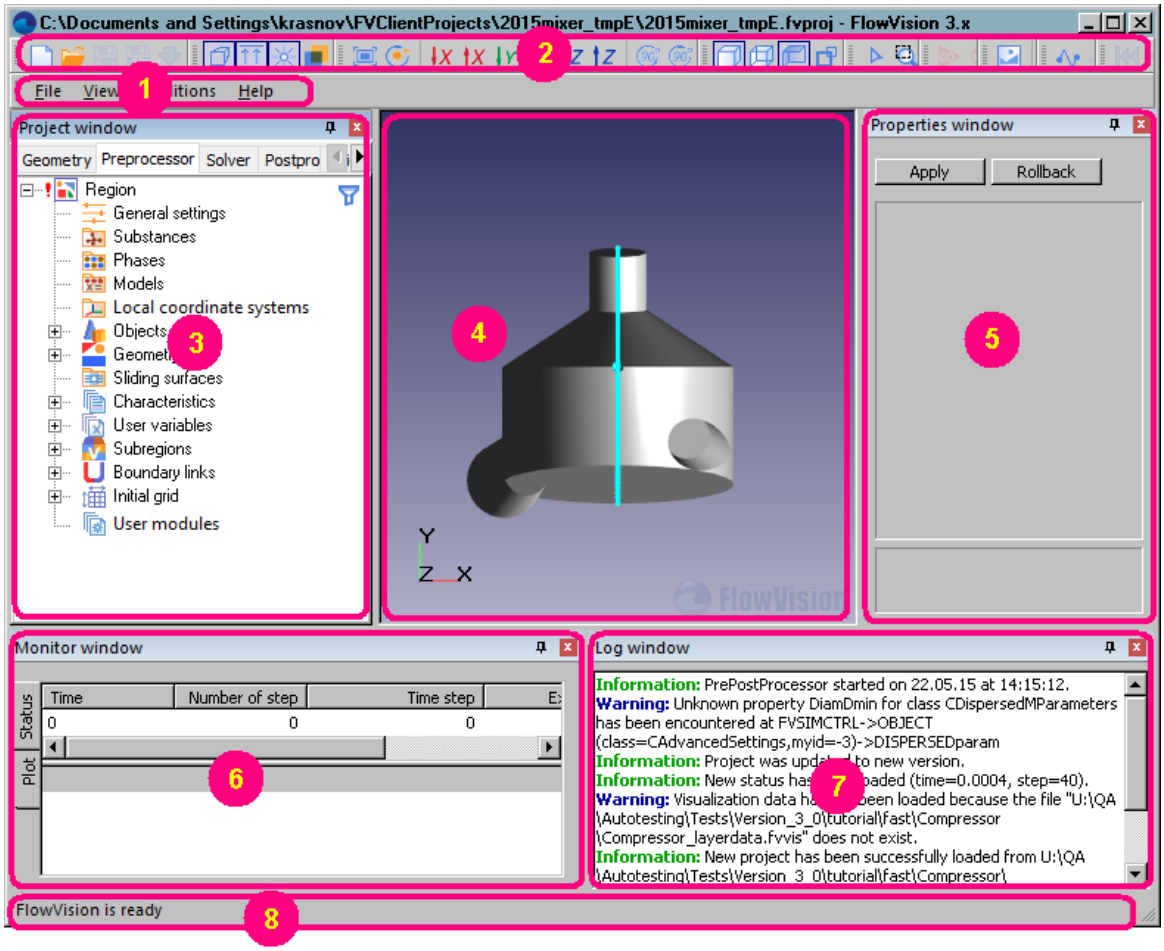
The geometry be used in the project, is created in a geometry modeling software or by a CAD system (outside the *FlowVision* software).

The project begins by loading the geometry model as follows:

Step	Actions
1	Start **Pre-Postprocessor** (*Windows'* **Start Button > Programs > FlowVision > Pre-Postprocessor**). Window of **Pre-Postprocessor** will open. Before the start of **Pre-Postprocessor**, the **License Manager**, on which the has been registered, must be started. The **License Manager**'s start can occur automatically at operating system's startup. If necessary, the **License Manager** can be started manually via the *Windows'* **Start Button** (**Start > Programs > FlowVisionLM > License Manager**), or by running its executable file (`FvLicense.exe` in *Windows* or `FvLicense` in *Linux*).
2	In the main menu select the command **File > Create**. The **New project** dialog box will open:  In this dialog box click the  (**Open geometric model**) button. A standard operating system's dialog box for selecting a file will open.
3	Select the file `Mixer.wrl` from the `Tutorial\Samples\Geom` subdirectory in the installation directory, so the default location of this file is: • `C:\Program Files\FlowVision-3.13.01\Tutorial\Samples\Geom` (for *Windows*) • `/home/user/FlowVision-3.13.01/Tutorial/Samples/Geom` (for *Linux*) with numbers (`3.13.01` as in the examples) in the name of the installation directory corresponding to the program's version number. After the geometry model loads, it will be displayed in the **View** window, which initially was empty.
4	In the main menu select the command **File > Save**. A dialog box for specifying the name and location of the project will open:

Step	Actions
	 Specify the name of the project in the field **New folder name** (and, if necessary, the project's location in the **Project location** field), then click **OK**. In the specified directory a subdirectory with the specified name will be created; this subdirectory will store files of the client part of the project. In the **Project** window, in the **Preprocessor** tab, the project tree will appear containing loaded elements of the project. Until at least one **Subregion** has a **Model** specified, the root **Region** folder in the project tree will be marked with the symbol "**!**". If the geometry contains errors, the root **Region** folder will be marked with "**!**" and the reason will be indicated. In such situation, you should check the geometry model for self-intersections. If self-intersections are found, you can try to fix the errors by the following means, for example: • load the geometry with another **Tolerance** (it is specified in the base settings of FlowVision by the parameter **Geometry import > Tolerance**) • in the context menu of the **Region** click **Fix self-intersections** • use a specialized software (for example, *3DTransVidia*) to correct errors in the geometry Loading of a previously created *FlowVision* project is done using the **File > Open** menu command, which opens a dialog box where you select the appropriate project (select first a directory, then a file with extension `fvproj`). If you can not download the file with geometry, then: • check whether the file corresponds to the *FlowVision* supported format and its elements correspond to the requirements • send the geometry file to the FlowVision's technical support with a description of the problems encountered
5	In the **Pre-Postprocessor**'s **View** window (4) the loaded geometry model of the computational domain is displayed:

Step	Actions



The **Pre-Postprocessor**'s window

In the illustration, the numbers refer to the elements of the **Pre-Postprocessor**'s window:

- the main menu (1)
- toolbars (2)
- the **Project** window (3)
- the **View** window (4)
- the **Properties** window (5)
- the **Monitor** window (6)
- the **Log** window (7)
- status bar (8)

In the **Project** window (3) the project tree is displayed, which contains four tabs: **Geometry**, **Preprocessor**, **Solver**, **Postprocessor**.

After creating the project and loading the geometry model, all the folders of the project tree are empty except for folders with the geometry model and parameters of its visualization:

- the geometry model is displayed in the **Preprocessor** tab in the folder **Region > Subregions > SubRegion #0 > Geometry**
- the boundary conditions, which are formed automatically when loading the geometry model, are displayed in the **Preprocessor** tab in the folder **Region > Subregions > SubRegion #0 > Boundary conditions**
- the boundary conditions visualization parameters are displayed in the **Postprocessor** tab in folders **3D-scene > Objects > Computational space > Solids > Subregions > SubRegion #0 > Boundary conditions** and **3D-scene > Layers > Solids > Subregions > SubRegion #0 > Boundary conditions**

In addition to the geometry model, in the **Postprocessor** tab of the **Project** window, an object **Plane #0** is created automatically in the folder **3D-scene > Objects > Computational space**. This **Plane** is displayed in the **View** window (4) as a vertical line (by default it is light blue; when the **Plane** is selected in the project tree, it is displayed in an inverted color, by default in red). **Plane #0** is oriented perpendicular to the X-axis and passes through the center of the computational domain.

Step	Actions
	This plane is not a part of the geometry model and has no influence on the flow of the fluid. The **Plane** is used to display the distribution of calculated values, for calculation of **Characteristics** on it, and to perform other auxiliary operations.

5.3 Basic operations in the View window

Before you generate a project, consider the loaded geometry model (see section Loading a geometry model of the mixer above).

Operations in the **View** window are performed using the toolbar buttons and a mouse. The contour of the cross section of the mixer by **Plane #0** is displayed in a light blue color.

Setting up the image using toolbar buttons

Step	Actions
1	In the project tree, in the tab **Preprocessor**, select the tab **Subregions > SubRegion #0 > Boundary conditions**.
2	Click and release the button  (**Enable/disable the use of perspective**) in the **Rendering** toolbar. View how the image changes (the perspective projection will be enabled or disabled).
3	Click and release the button  (**Enable/disable surface fill**) in the **Solids** toolbar. View how the image changes (the surfaces will be filled with a color or will be transparent).
4	Click and release the button  (**Enable/disable the display of facet edges**) in the **Solids** toolbar. View how the image changes (the wireframe consisting of edges of individual facets will be displayed or will not).
5	Using the button  (**Enable/disable surface fill**) in the **Solids** toolbar, enables the surface fill. Click and release the button  (**Show/hide surface facing viewer**) in the **Solids** toolbar. View how the image changes (when the button is pressed, the surface, closest to the observer, becomes transparent).
6	Click and release the button  (**Enable/disable display of face group borders**) in the **Solids** toolbar. View how the image changes (when the button is pressed, outlines of groups of facets are outlines with heavy lines).

Orientating the view direction along coordinate axes (in positive or negative direction)

Step	Actions
•	By clicking on buttons  , in the **Sights** toolbar, orient the display of the mixer along coordinate axes of the absolute coordinate system (ACS) in positive or negative direction.

Rotating the image

Step	Actions
1	Click the button  (**View transformation mode**) in the **Work modes** toolbar.
2	Place the mouse pointer into the **View** window.
3	Press the *left* mouse button and, holding the button, move the mouse. For orienting, use the image of the absolute coordinate system's unit vectors (by default it is displayed in the lower left corner of the **View** window).

Parallel shift of the image

Step	Actions
1	Click the button (**View transformation mode**) in the **Work modes** toolbar.
2	Place the mouse pointer into the **View** window.
3	Press the *right* mouse's button and, holding the button, move the mouse.
4	To return to the initial position of the image in the **View** window and display the whole geometry model, click the button (**Fit calculation region to window**) in the **Sights** toolbar.

Scaling the image

Step	Actions
1	Click the button (**View transformation mode**) in the **Work modes** toolbar.
2	Place the mouse pointer into the **View** window.
3	Rotate the mouse wheel. The image will be scaled relative to the center of the **View** window.
4	To return to the initial position of the image in the **View** window and display the whole geometry model, click the button (**Fit calculation region to window**) in the **Sights** toolbar.

Displaying individual groups of facets

The geometry model consists of several groups of facets. To display their outlines, click (**Enable/disable display of face group borders**) in the **Solids** toolbar. To display an individual group of facets, do the following steps:

Step	Actions
1	Click the button (**Enable facet group selection mode**) in the **Work modes** toolbar.
2	Click some point in the image of the geometry model. Image of only one group of facets (located below the mouse pointer) will remain. One of sides of the group of facets is the surface's *front side* and it is filled with an appropriate color when the button is pressed. The *back side* of the surface (when the button is pressed) is also filled with the same color but it also has the pattern of the "back side" symbols .
3	When you repeatedly click, other groups of facets below the mouse pointer will be displayed (one group per click).
4	To display the entire geometry model (according to its display settings), click in the **View** window outside the geometry model.

5.4 Cross-section of a geometry model with a Plane

When loading the geometry model, the program automatically creates a **Plane** (the geometric object **3D-scene > Objects > Computational space > Plane #0** located in the **Postprocessor** tab of the project tree), which can visually clip some part of the geometry model.

Making a visual cut of a part of the geometry model by a Plane

Visually clipping a part of the geometry model by a **Plane** requires two conditions:

- the **Plane** must be a clipping object
- the geometry model must be able to be clipped

To provide these conditions, do the following:

Step	Actions
1	Open the context menu of the element **3D-scene > Objects > Plane #0** in the **Postprocessor** tab of the project tree, and place a mark near the **Clipping object** option.
2	Open the context menu of the element **3D-scene > Objects > Computational space > Solids** in the **Postprocessor** tab of the project tree, and place a mark near the **Apply clipping** option.
3	Select the element **3D-scene > Objects > Plane #0** in the **Postprocessor** tab of the project tree. In the **View** window you will see the **Plane**, its **Reference point**, and the **Plane**'s normal vector originating from the **Reference point**. You will see that the **Plane** cuts off the negative half-subspace (the normal vector is directed into the positive half-subspace).

Changing position of the Plane using the mouse

Step	Actions
1	Select the element **3D-scene > Objects > Plane #0** in the **Postprocessor** tab of the project tree.
2	Click the button (**Enable edit mode for selected object**) in the **Work modes** toolbar.
3	Place the mouse pointer into the **View** window.
4	To rotate the **Plane** around its **Reference point**, press the *left* mouse button and, holding the button, move the mouse.
5	To move the **Reference point** on the **Plane**, press the *left* mouse button and, holding the button, move the mouse.
6	To move the **Plane** back and forth along its normal vector, press both left and right mouse buttons and, holding them, move the mouse.

Orientation the Plane using buttons in its Properties window

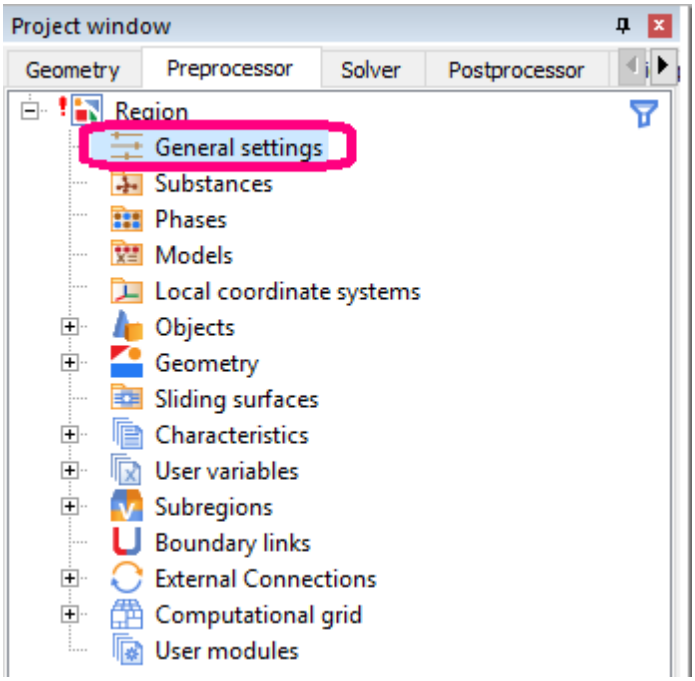
Step	Actions
1	Select the element **3D-scene > Objects > Plane #0** in the **Postprocessor** tab of the project tree.
2	Use buttons in the **Properties** window: • to orient the **Plane** along the axis X (Y, Z), click the button **Object > Operations >** (, ,), and then click **Apply** • to orient the **Plane** against the axis X (Y, Z), click the button **Object > Operations >** (, ,) again, and then click **Apply**
3	Rotate the **Plane** arbitrarily with a mouse as described in subsection *"Changing position of the Plane using the mouse"* above.
4	Reverse the orientation of the **Plane**: in its **Properties** window. Click the button **Object > Operations >** (**Invert**), and then click **Apply**.

5.5 Specifying physical parameters of the project

See sections below:

- Specifying general settings of the project
- Specifying Substances and their parameters
- Specifying Phases and their parameters
- Specifying a Model and its parameters
- Specifying a Model in the computational domain
- Specifying Boundary conditions
- Specifying Initial conditions

5.5.1 Specifying general settings of the project

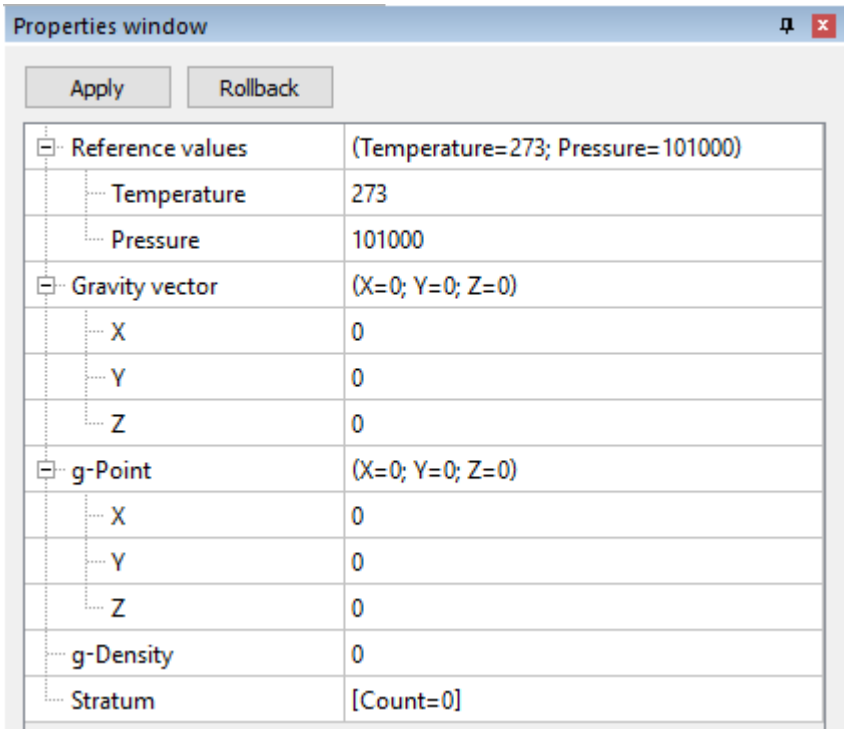


Element **General settings** in the project tree

The **General settings** are specified in the **Properties** window of the element **Region >** **General settings** in the project tree (in the tab **Preprocessor**).

The **General settings** of the project include the following parameters:

- Reference values of **Temperature** T_{ref} and **Pressure** P_{ref} (by default they are 273 [K] and 101000 [Pa])
- Vector of *gravity acceleration* in the absolute coordinate system associated with the geometry model (the **Gravity vector**)
- **g-Point** is a point at which the hydrostatic component of pressure is zero. The point is defined by its coordinates in the absolute coordinate system.
- **g-Density** is density of a heavy fluid, located above the **g-point** (and if the layers of a heavy fluid are not specified, then also the density below the **g-point**), which is taken into account in the hydrostatic component of pressure.
- Thickness and density of the fluid in the layers of the heavy fluid located below the g-point, specified individually for each of these layers. The density, specified for the lowest layer, is also applied for the fluid below the layer. See details in sections Hydrostatic component of pressure and Hydrostatics.



The **Properties** window of the **General settings** element

In our example the gravity affects negligibly the solution, so you don't have to change the default values of **General settings**.

Navigate to the next element of the project tree, **Substances**.

5.5.2 Specifying Substances and their parameters

Substance is determined by its aggregative state and properties.

Substances are created in the folder **Substances** in the **Preprocessor** tab of the project tree. There can be several **Substances** in a project. The aggregative state of a **Substance** is defined in the **Substance**'s **Properties** window.

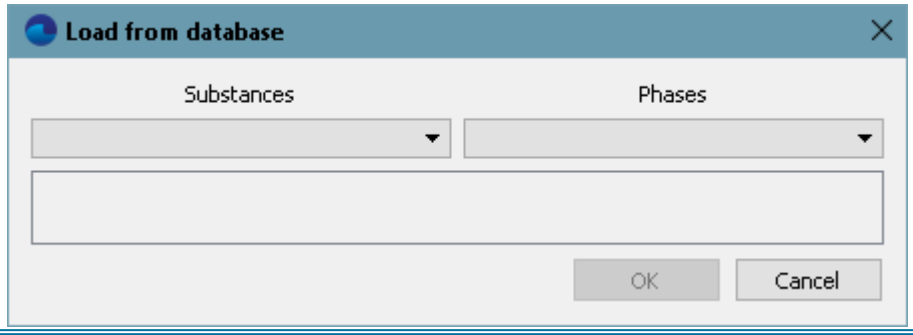
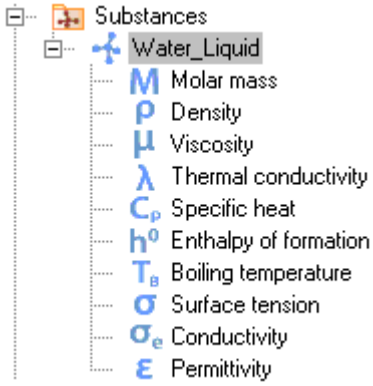
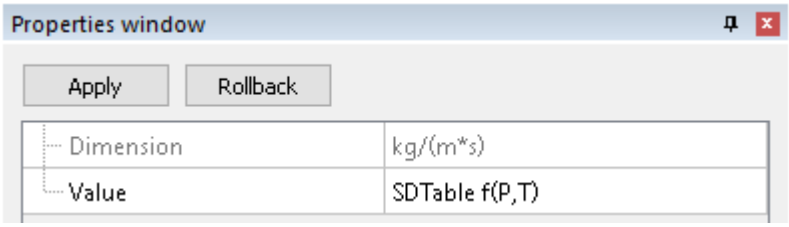
Each **Substance** is presented in the project tree as a folder containing several child elements, which correspond to *physical properties* of this **Substance** (**Molar mass**, **Density**, **Viscosity**, etc.). The values of the physical properties are specified in the **Properties** windows of these child elements.

Parameters of a **Substance** in the project can be defined manually or loaded from the **Substance Database**.

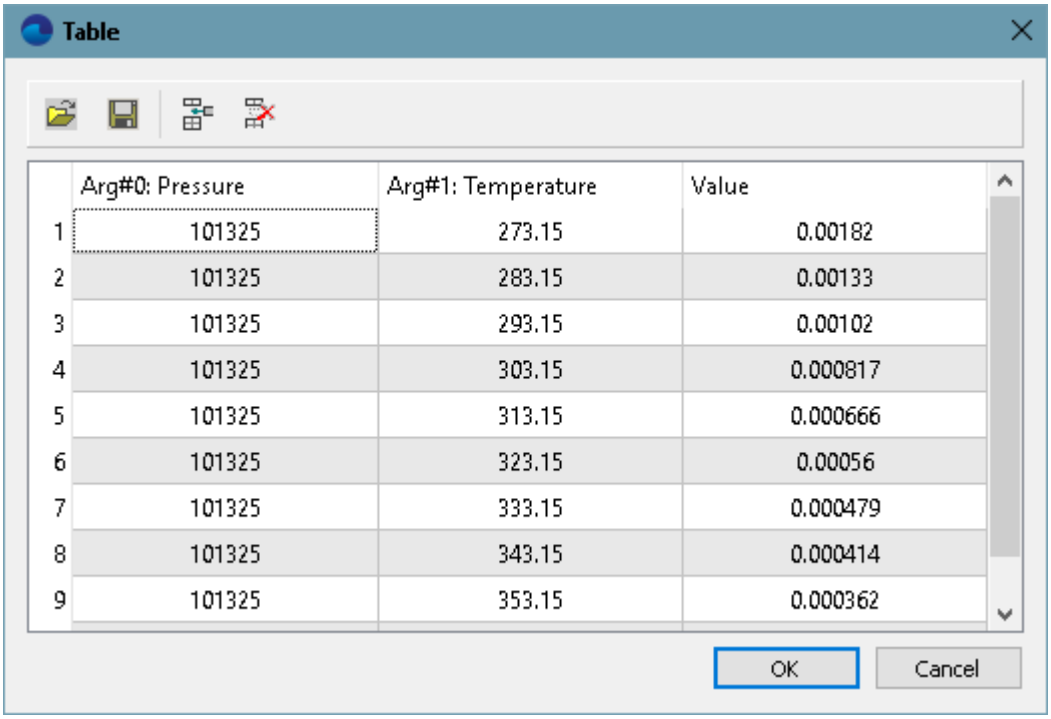
Let us define the physical properties of the fluid, which is a single-phase in this project and has only one substance (water), considered as an incompressible liquid.

Specifying parameters of a substance

Step	Actions
1	With a right-click, open the context menu of the folder **Substances** in the **Preprocessor** tab and select the command **Create**. In the folder **Substances** a new subfolder **Substance #0** will appear.
2	Open the context menu of the subfolder **Substance #0**, and then select **Load from SD > Standard**. The **Load from database** dialog box will open:

Step	Actions
	
3	Select: • **Water** from the **Substances** drop-down list • **Liquid** from the **Phases** drop-down list Click **OK**. In the folder **Substances** a new element (subfolder) **Water_Liquid** will appear.
4	Expand the **Water_Liquid** element (click the symbol "+"). A list of physical properties of this substance will be displayed: 
5	To view values of the **Substance**'s physical property that has been loaded from the **Substance Database** as a table with dependency on pressure and temperature (for example, to view values of **Viscosity**), do the following: a) In the project tree select the element, which corresponds to the **Substance**'s physical property, which you wish to view, for example, **Water_Liquid > Viscosity**. Parameters of **Viscosity** will be displayed in its **Properties** window:  b) In the **Properties** window, click the **Value** field (on the data entry field itself), and then click the icon (**SDTable f(P,T)**), which will appear near the field. The dialog box **Table** will open with a tabulated function of the viscosity depending on pressure (X) and temperature (Y):

Step	Actions



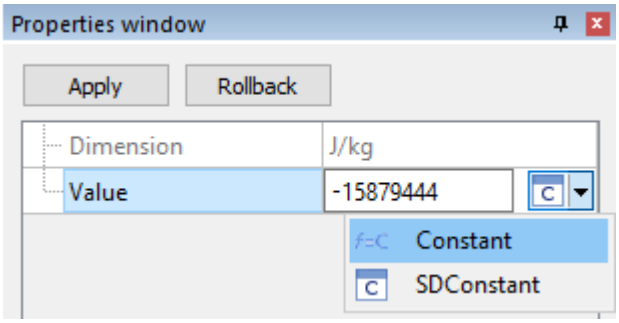
c) Close the dialog box **Table**.

6

Specify a zero value for **Enthalpy of formation** (because this parameter is only used in the model of mass transfer with chemical reactions or combustion):

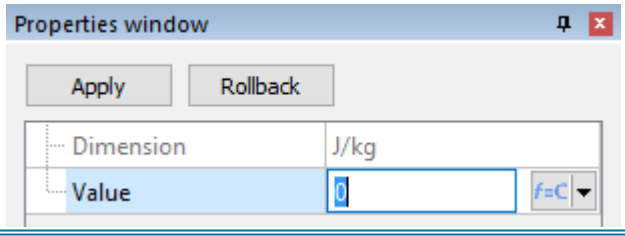
a) Select the **Water_Liquid > Enthalpy of formation** element from the project tree.

b) In the **Properties** window, click the line **Value** and then click :



c) A drop-down list will open. Select **Constant**.

d) In the input field type the value **0** and click **Apply**:

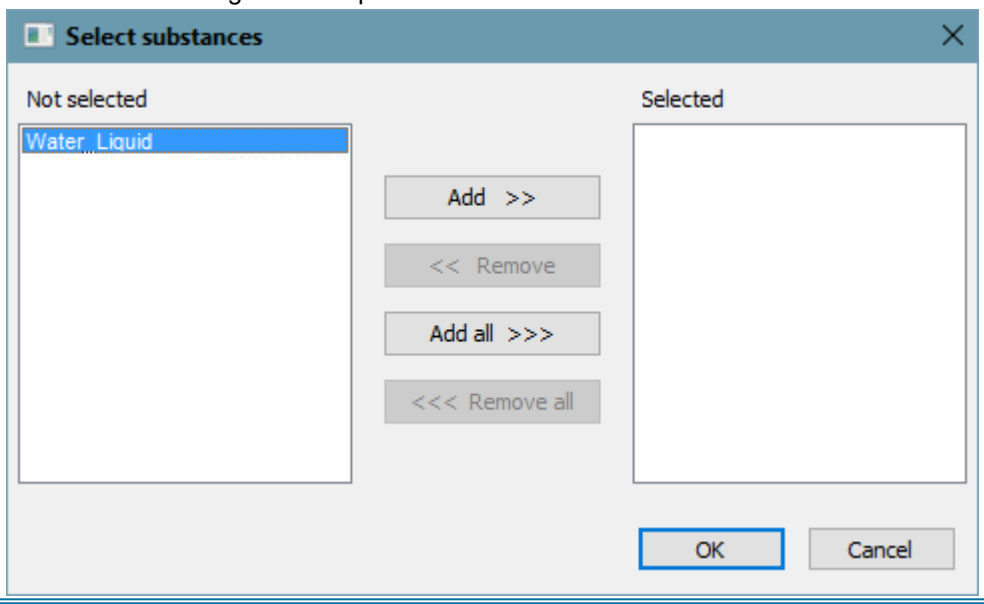
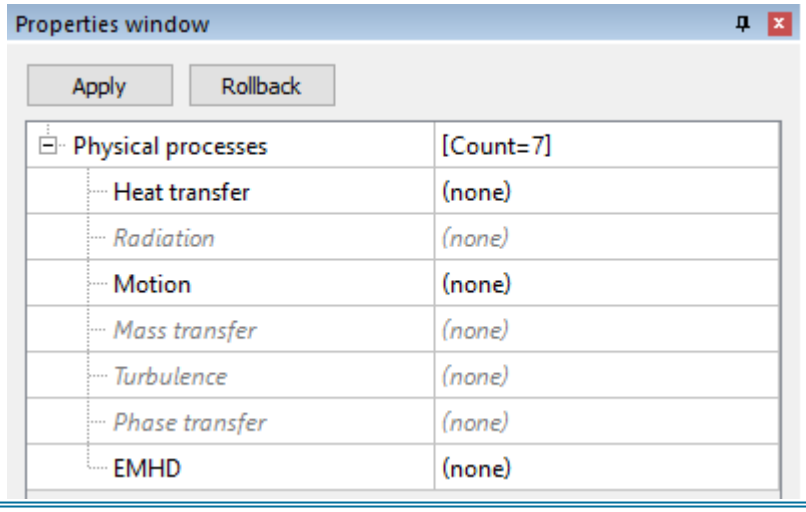


5.5.3 Specifying Phases and their parameters

Phase is a set of **Substances** and the simulated **Physical processes**.

Phases are defined in the **Preprocessor** tree in folder **Phases**. The project must have at least one **Phase**. The number of phases is unlimited.

We will specify, which **Substance** will form the **Phase** in our example, and specify in this **Phase**, what **Physical processes** will be used for the simulation.

Step	Actions
1	Open the context menu of the folder **Phases** (in the **Preprocessor** tab of the project tree) and select **Create continuous**. A new element (folder) **Phase #0** will appear in the folder **Phases**.
2	Expand the newly created folder **Phase #0**, select its subfolder **Substances** and in the context menu select the command **Add/Remove**. The **Select substances** dialog box will open: 
3	Select from the list **Not selected** (on the left) the substance **Water_Liquid** and click **Add**. The selected substance will appear in the list **Selected** (on the right).
4	Click **OK**. In the subfolder **Phase #0 > Substances** a new element, **Water_Liquid**, will appear.
5	Select the element **Phase #0 > Physical processes**. In its **Properties** window a list of physical processes is displayed: 
6	Select from drop-down lists the required physical processes in the following order: • for the parameter **Heat transfer** select its value as **Heat transfer via h** • for the parameter **Motion** select its value as **Navier-Stokes model** • for the parameter **Turbulence** select its value as **KES** (the standard k-ε turbulence model). Please note that the turbulence model cannot be specified while **Motion= (none)**.
7	Click **Apply**. The specified elements for physical processes of **Phase #0** will appear in the project tree:

Step	Actions
	

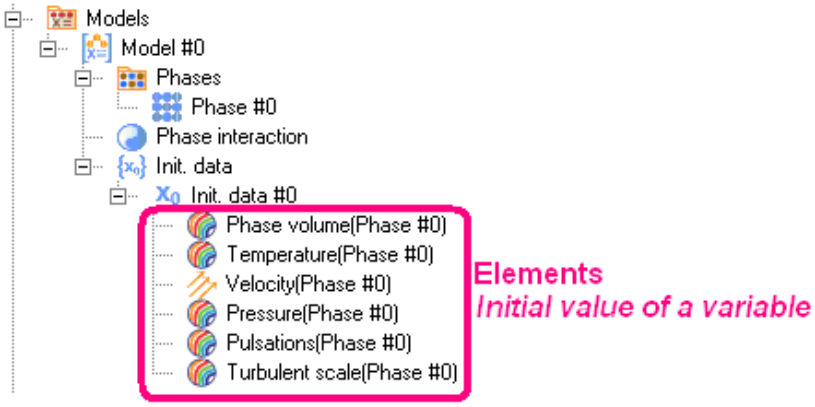
5.5.4 Specifying a Model and its parameters

A *model* is a set of phases and phase interactions.

Models are created in the project tree, in the tab **Preprocessor**, in the folder **Models**. Any project should contain at least one **Model**. The number of models in a project is unlimited.

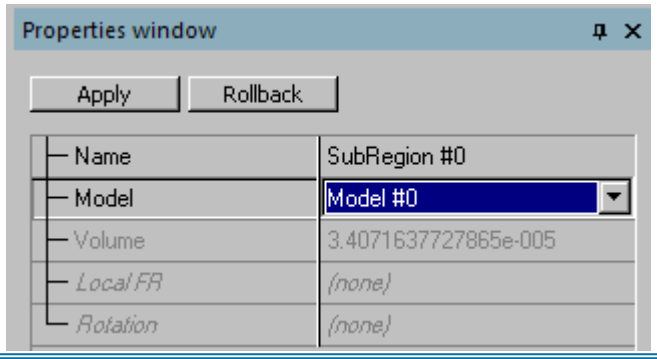
If a **Model** contains several **Phases**, you can specify **Phase interaction**.

At this stage of the formation of the project, let's specify, which **Phase** is included with the **Model**, and define the initial data, which will be used later to specify the initial conditions for the **Model**'s equations.

Step	Actions
1	Open the context menu of the folder **Models** and select the command **Create**. In the folder **Models** a new element (a folder) will appear, **Model #0**.
2	Expand the folder **Model #0**, open a context menu of the subfolder **Model #0 > Phases**, and select from the menu the command **Add/Remove**. The **Select phases** dialog box will open.
3	Select the desired phase from the **Not selected** list in the left pane (in our case this phase is **Phase #0**), and then click **Add**. The selected phase will be displayed in the **Selected** list in the right pane. Click **OK**.
4	In the folder **Phases** the newly added element **Phase #0** will appear. In the folder **Init. data > Init. data #0** the child elements ***Initial value of a variable*** are located: 
5	When you select in the project tree an element ***Initial value of a variable***, then you can see in its **Properties** window the default value of the initial data. At the initial moment of time the liquid in the mixer is not moving and its temperature is 0 degrees Celsius (=273 Kelvin), which is equal to the reference value of the temperature (which is specified by the **Reference values > Temperature** parameter in properties of the **Region > General settings** element). The *difference* between the initial temperature and the reference temperature is 0 [K].

5.5.5 Specifying a Model in the computational domain

Let's specify, which **Model** will be used in the computational **Subregion** (in non-computational **Subregions** no **Model** is specified).

Step	Actions
1	Select in the **Preprocessor** tab of the project tree the folder **Subregions > SubRegion #0**.
2	In its **Properties** window, specify: • for the parameter **Model**, select its value as **Model #0** from the drop-down list 
3	Click **Apply**.

5.5.6 Specifying Boundary conditions

A *boundary condition* is the condition imposed on the calculated variables on borders of a **Subregion**.

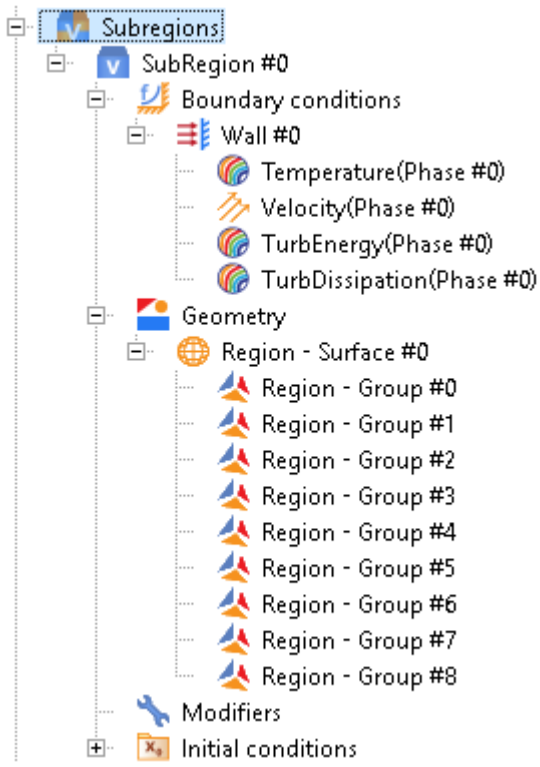
Boundary conditions are specified in the folder **Boundary conditions** of an appropriate **Subregion**.

Before you specify the **Boundary conditions** in the **Properties** window of the folder **Subregion**, you have to select some pre-specified **Model**.

When loading a geometry model, the model's surface is divided into groups of facets which are presented in the project tree in the **Preprocessor** tab by the elements **Subregions > SubRegion #0 > Geometry > Region- Surface #0 > Region- Group #N** (see illustration).

When you use the default settings, the elements **Region- Group #N** are not displayed in the project tree (in folder **Subregions > SubRegion #N> Geometry > Region - Surface #N**), because in real-world tasks the program might create tens or even hundreds of elements **Region- Group #N**.

If you wish to display the elements **Region- Group #N**, specify **Display > Show All Groups = Yes** in the basic settings (which are opened by the command **File > Preferences** from the main menu).



Folder **Subregions** in the project tree

In order to assign boundary conditions to these groups of facets, you firstly have to create the boundary conditions in the folder **Subregions > SubRegion #0 > Boundary conditions**. When loading a geometry model, one boundary condition, **Wall #0** (thermally insulated), is automatically created in this folder.

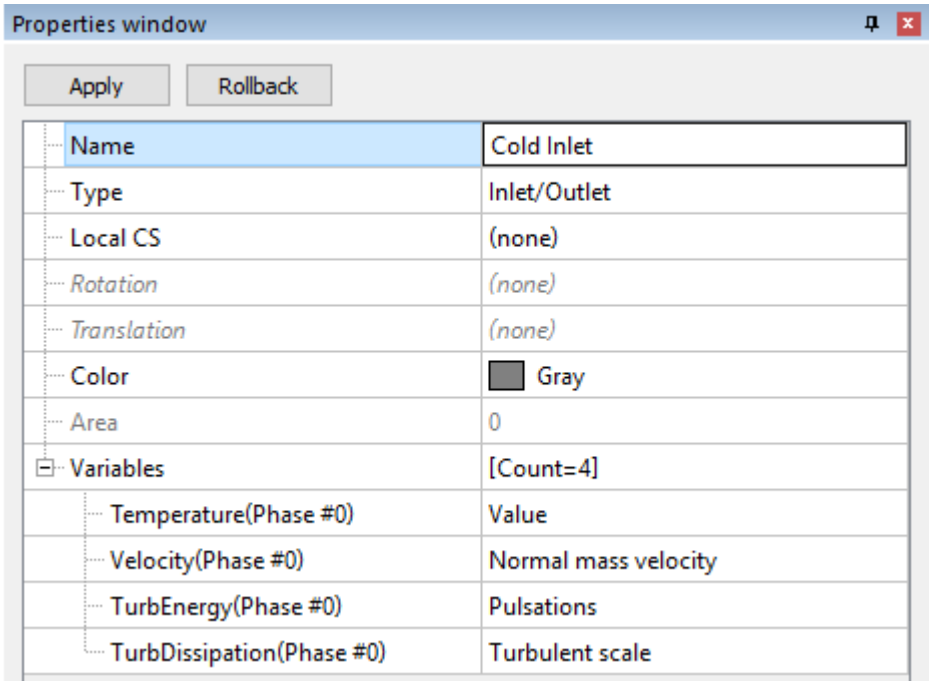
The boundary conditions which you will create are then assigned to groups of facets.

After assigning the **Boundary conditions**, you have to specify parameters of each **Boundary condition**.

Create boundary conditions

According the step-by-step procedure given below, you have to create four **Boundary conditions** that will be set on walls of the mixer, on inlets for cold and hot water, and on the outlet from the mixer.

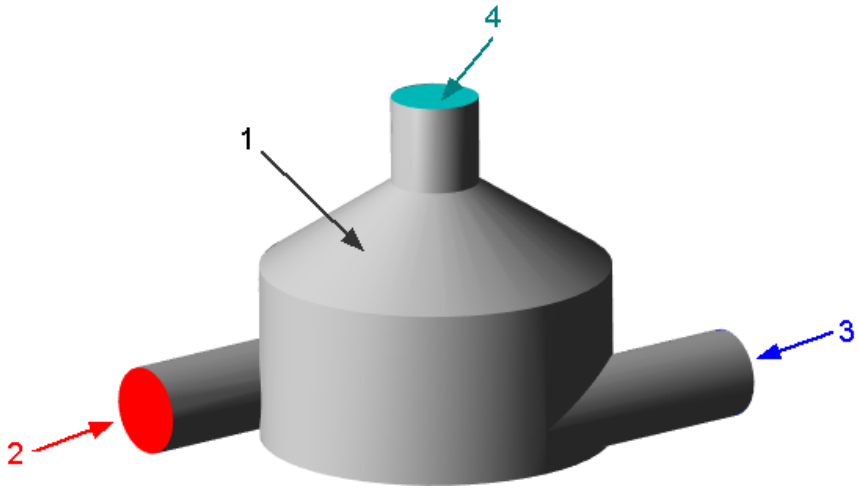
Step	Actions
1	In the project tree, in the folder **Subregions > SubRegion #0 > Boundary conditions**, there is an automatically created **Boundary condition**, which is presented by the element **Wall #0**. Select from the project tree the element **Wall #0**, and, in its **Properties** window, specify **Name** = **Wall**, and then click **Apply**. So, you remove the number code "**#0**" from the standard name **Wall #0** because you don't need it as this exercise has only one wall boundary condition. This **Boundary condition** will be used for wall of the mixer.
2	Create another **Boundary condition**. Open the context menu of the folder **Subregions > SubRegion #0 > Boundary conditions** and select the command **Create**. The **Wall #0** element will appear in the **Boundary conditions** folder (the program again uses the standard name **Wall #0** because the element **Wall #0** from the previous step has been renamed to just **Wall** without a number.
3	Change the name and the type of the newly created element **Wall #0**. Select from the project tree the **Wall #0** element and in its **Properties** window specify: • **Type = Inlet/Outlet** (select it from a drop-down list) • **Name = Cold Inlet**

Step	Actions
	 Then click **Apply**.
4	Make *two more* copies of the boundary condition **Cold inlet**, copying twice the element **Cold inlet**: • Open the context menu of the element **Cold inlet** and select the command **Copy**. In the **Boundary conditions** folder a new element **Inlet/Outlet #0** will appear. • Again open the context menu of the element **Cold inlet** and select the command **Copy**. In the **Boundary conditions** folder another new element **Inlet/Outlet #1** will appear. So at this step you create two boundary conditions, **Inlet/Outlet #0** and **Inlet/Outlet #1**. At next steps names and/or properties of these boundary conditions will be changed.
5	Give a correct name for the **Boundary condition**, which corresponds to the hot water inlet. Select in the project tree the element **Inlet/Outlet #0** and in its **Properties** window specify: • **Name = Hot inlet** Then click **Apply**.
6	The last **Boundary condition** will be used for the water outlet from the mixer. Select in the project tree the element **Inlet/Outlet #0** and in its **Properties** window specify: • **Type = Free Outlet** (select from a drop-down list) • **Name = Outlet** Then click **Apply**.
7	Specify colors for the created **Boundary conditions**: • From the context menu of the boundary condition **Cold inlet** select the **Set color** command and then select the blue color. • From the context menu of the boundary condition **Hot inlet** select the **Set color** command and then select the red color. • From the context menu of the boundary condition **Outlet** select the **Set color** command and then select the aqua color.

Assignment of the boundary conditions on groups of facets

Assignment of the **Boundary conditions** on groups of facets is done by item-by-item examination the groups and assigning each group a required boundary condition. You must do this for each group of facets. As the **Wall** boundary condition is already assigned, you only have to change the **Boundary conditions** on two inlets and one outlet.

Assignment of the **Boundary conditions** is done according the illustration below:



1- **Wall**, 2 - **Hot inlet**, 3- **Cold inlet**, 4 - **Outlet**

Step	Actions
1	If the elements in the folder **Subregions > SubRegion #0 > Geometry > Region- Surface #0 > Region - Group #N** are not displayed in the project tree, then adjust their visibility in the basic settings of FlowVision. To accomplish this, open the command menu **File > Preferences** and set **Display > Show all groups = Yes**.
2	Select the element **Subregions > SubRegion #0 > Geometry > Region - Surface #0 > Region - Group #1**, and in its **Properties** window specify **Boundary condition = Outlet** (select the value from the drop-down list), and then click **Apply**.
3	Select the element **Subregions > SubRegion #0 > Geometry > Region - Surface #0 > Region - Group #6**, and in its **Properties** window specify **Boundary condition = Cold inlet** (select the value from the drop-down list), and then click **Apply**.
4	Select the element **Subregions > SubRegion #0 > Geometry > Region - Surface #0 > Region - Group #8**, and in its **Properties** window specify **Boundary condition = Hot inlet** (select the value from the drop-down list), and then click **Apply**.

Specifying parameters of the boundary conditions

The parameters of the **Boundary conditions** are specified in their own **Properties** windows and in **Properties** windows of their child elements (which correspond to variables).

In the step-by-step procedure below you see the values of parameters, that should be specified. You have to enter *or* select these values, *or* ensure that the default values are the same as required by the step-by-step procedure (so you don't have to enter or select them).

Step	Actions
1	For the element **Subregions > SubRegion #0 > Boundary conditions > Wall** in its **Properties** window specify: a) for the parameter **Variables > Velocity**, select its value as **Logarithm law** b) for the parameter **Variables > Temperature**, select its value as **Zero gradient** c) for the parameter **Variables > TurbEnergy**, select its value as **Value in cell near wall** d) for the parameter **Variables > TurbDissipation**, select its value as **Value in cell near wall** Click **Apply**.
2	For the element **Subregions > SubRegion #0 > Boundary conditions > Cold inlet**: a) In its **Properties** window specify **Variables > Temperature = Temperature**. Click **Apply**. b) In its **Properties** window specify **Variables > Velocity = Normal mass velocity**. Click **Apply**.

Step	Actions
	c) In the **Properties** window of its child element **Velocity** specify **Mass velocity = 1285.7**[*)] [kg·m^{-2}s^{-1}]. Click **Apply**. d) In the **Properties** window of its child element **Temperature** specify **Value = 0** [K] (which means that the temperature is 0 [K] above the reference temperature 273[K], i.e. it is 0°C). Click **Apply**. e) In its **Properties** window specify **Variables > TurbEnergy = Pulsations**. Click **Apply**. f) In the **Properties** window of its child element **TurbEnergy** specify **Value = 0** (it is dimensionless quantity). Click **Apply**. g) In its **Properties** window specify **Variables > TurbDissipation = Turbulent scale**. Click **Apply**. h) In the **Properties** window of its child element **TurbDissipation** specify **Value = 0** [m]. Click **Apply**.
3	For the element **Subregions > SubRegion #0 > Boundary conditions > Hot inlet**: a) In its **Properties** window specify **Variables > Temperature = Temperature**. Click **Apply**. b) In its **Properties** window specify **Variables > Velocity = Normal mass velocity**. Click **Apply**. c) In the **Properties** window of its child element **Velocity** specify **Mass velocity = 1285.7**[*)] [kg·m^{-2}s^{-1}]. Click **Apply**. d) In the **Properties** window of its child element **Temperature** specify **Value = 70** [K] (which means that the temperature is 70 [K] above the reference temperature 273[K], i.e. it is 70°C). Click **Apply**. e) In its **Properties** window specify **Variables > TurbEnergy = Pulsations**. Click **Apply**. f) In the **Properties** window of its child element **TurbEnergy** specify **Value = 0** (it is dimensionless quantity). Click **Apply**. g) In its **Properties** window specify **Variables > TurbDissipation = Turbulent scale**. Click **Apply**. h) In the **Properties** window of its child element **TurbDissipation** specify **Value = 0** [m]. Click **Apply**.
4	For the element **Subregions > SubRegion #0 > Boundary conditions > Outlet**: a) In its **Properties** window specify **Variables > Velocity = Pressure**. Click **Apply**. b) In the **Properties** window of its child element **Velocity** specify **Value = 0** [Pa] (above the reference pressure). Click **Apply**. c) In its **Properties** window specify **Variables > Temperature = Zero gradient**. Click **Apply**. d) In its **Properties** window specify **Variables > TurbEnergy = Zero gradient**. Click **Apply**. e) In its **Properties** window specify **Variables > TurbDissipation = Zero gradient**. Click **Apply**.

[*)] The value **1285.7** [kg·m^{-2}s^{-1}] is derived as division of the mass flow 0.1 [kg·s^{-1}] by the area of the section of the supply tube 7.7777591212273E-05 [m^2]. The area of the section of the tube is displayed by the parameter **Area** in the **Properties** window of the appropriate **Boundary condition**.

5.5.7 Specifying Initial conditions

Initial conditions specify that on a given **Object** at the initial moment of time some **Initial data** (fields of values of calculated variables) are applied.

By default, all variables at the initial time moment are zero.

Initial conditions are defined in a **Subregion**.

Prior to specifying the **Initial conditions**, you must specify a **Model** in the **Properties** window of the folder **Subregion #N** (select it from the drop-down list of existing **Models**).

For our example problem (water flow in the mixer) you don't have to specify **Initial conditions** because the default values can be used. Anyway, for this exercise, we will give a step-by-step procedure of specifying **Initial conditions**:

Step	Actions
1	Create **Initial data**, which will be specified in the properties of the **Initial conditions**. Open the context menu of the folder **Models > Model #0 > Init. data** and select the command **Create**. A new folder will appear in the project tree, **Models > Model #0 > Init. data > Init. data #1**. (Note, that the folder **Model #0 > Init. data** already contains the element **Init. data #0**, which is created automatically)
2	Specify values of child elements ***Initial value of a variable*** of the newly created element **Init. data #1**.
3	Create an **Object**, on which the **Initial conditions** will be applied. To do this, open the context menu of the folder **Objects** and select the command **Create**, and then specify the type of the new **Object**. (Note, that the folder **Objects** already contains the element **Computational space**, which is created automatically. You can also use it as an **Object**, upon which the **Initial conditions** will be applied.)
4	Specify the required parameters of the newly created **Object** (in its **Properties** window).
5	Create a new element **Initial condition**. To do this, open the context menu of the folder **Subregions > SubRegion #0 > Initial conditions** and select the command **Create**. A new element will appear in the folder **Initial conditions**, **Init. condition #1**. (The folder **Initial conditions** already contains the element **Init. condition #0**, which is created automatically by the program.)
6	Specify properties of the new element **Init. condition #1**, by matching it and the required **Initial data** and **Object**. In the **Properties** window of the newly created element **Init. condition #1** specify: • for the parameter **Object**, specify its value (select from a drop-down list) as any **Object** from the list (you can also select the **Computational space** as an **Object**) • for the parameter **Init. data**, specify its value (select from a drop-down list) as **Init. data #1** • if you wish to change the standard name of the element **Init. condition #1**, enter a new name as the value of the parameter **Name** Click **Apply**.
7	In our example with the mixer we will use zero initial values so we do not need this newly created element **Init. data #1** so delete it using the command **Delete** from its context menu.

Specifying non-zero initial conditions accelerates the convergence of the solution (for example, in the problems of external flow it is convenient to define the field of initial velocity as corresponding to the velocity of the forward flow).

5.6 Specifying a computational grid

The *computational grid* is a set of cells into which the computational domain is divided. The values of variables within a single cell are constants.

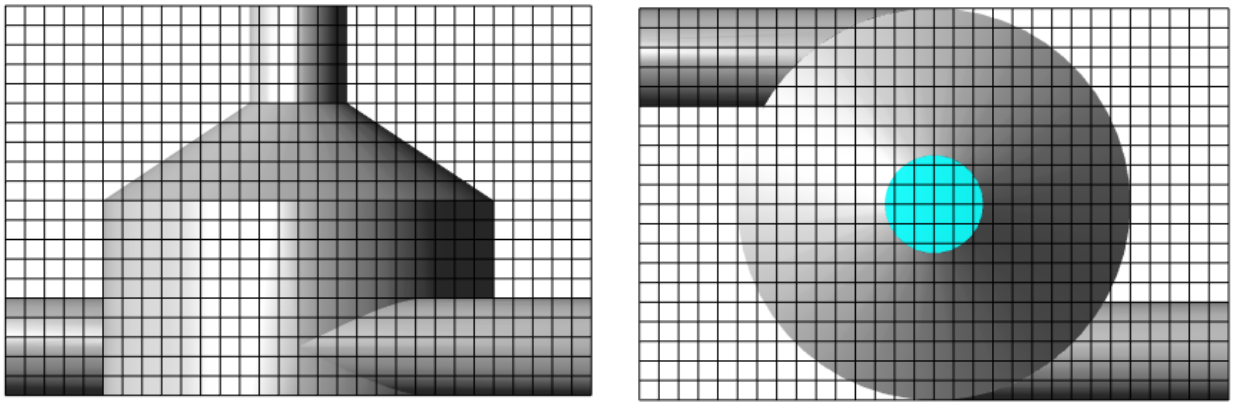
In our example, specifying the **Computational grid** consists of two stages:

a) specifying the **Initial grid**. A uniform initial computational grid will be specified, i.e. a grid with constant steps along each of the coordinate axes with a number of steps N_X=20, N_Y=20, N_Z=30. See the section Specifying the initial grid.

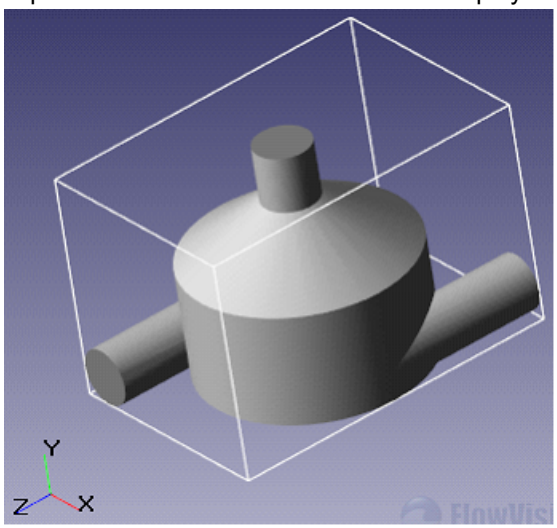
b) specifying an **Adaptation** with two layers and one level on walls of the mixer. See section Specifying an adaptation. In our example **Adaptation by condition** and **Adaptation to solution** are not specified.

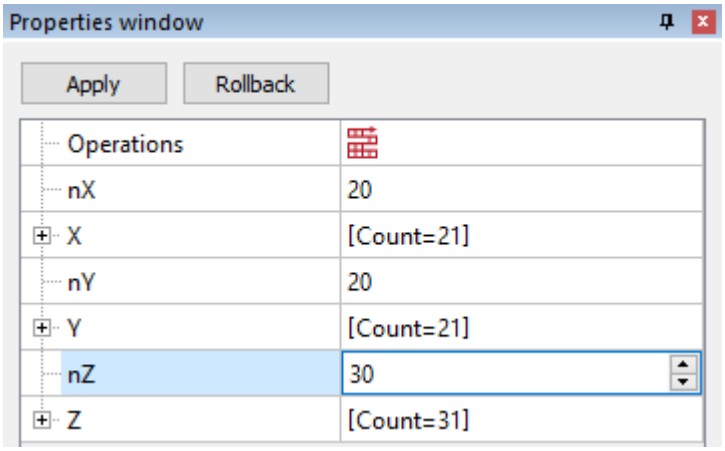
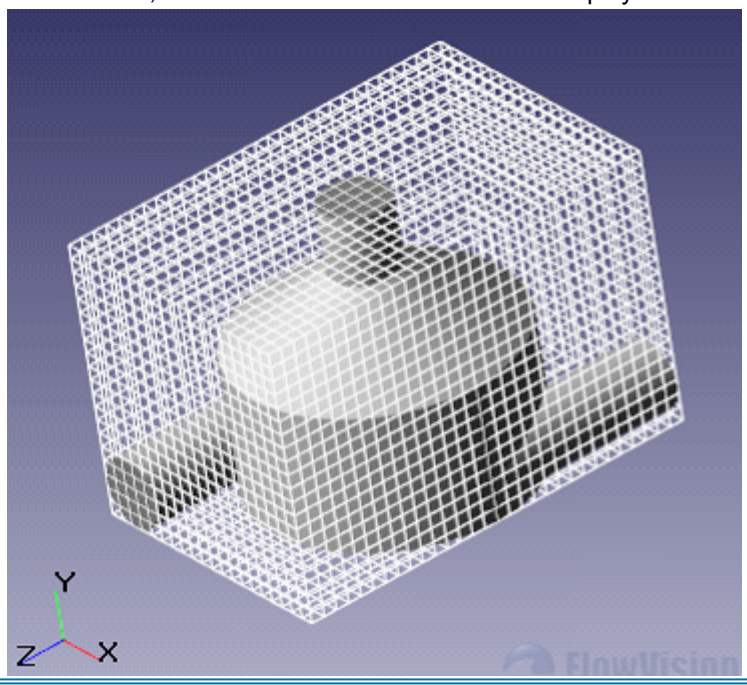
5.6.1 Specifying the initial grid

We will specify a uniform **Initial grid** with number of cells 20, 20, and 30 along axes X, Y, Z respectively.



Do the following steps:

Step	Actions
1	In the project tree, in the **Preprocessor** tab, select the folder **Initial grid**. In the **View** window the computational domain in a "box" will be displayed: 
2	In the **Properties** window of the folder **Initial grid**, specify numbers of steps for each coordinate axis: • for the parameter **nX**, specify its value as **20** • for the parameter **nY**, specify its value as **20** • for the parameter **nZ**, specify its value as **30**

Step	Actions
	 Then click **Apply**. The **Initial grid** will be created, and in the **View** window it will be displayed: 

5.6.2 Specifying an adaptation

Adaptation is accomplished by splitting the grid cells into smaller cells by half in direction of each coordinate axis X, Y and Z (so each splitting divides a cell into 8 parts).

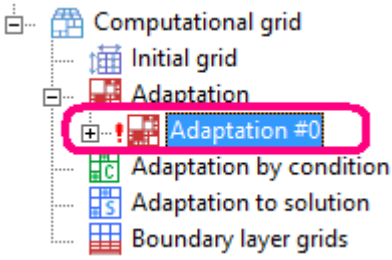
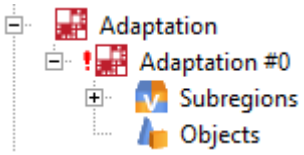
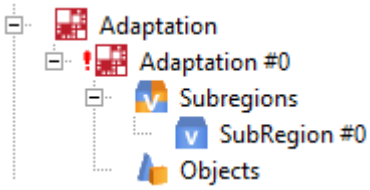
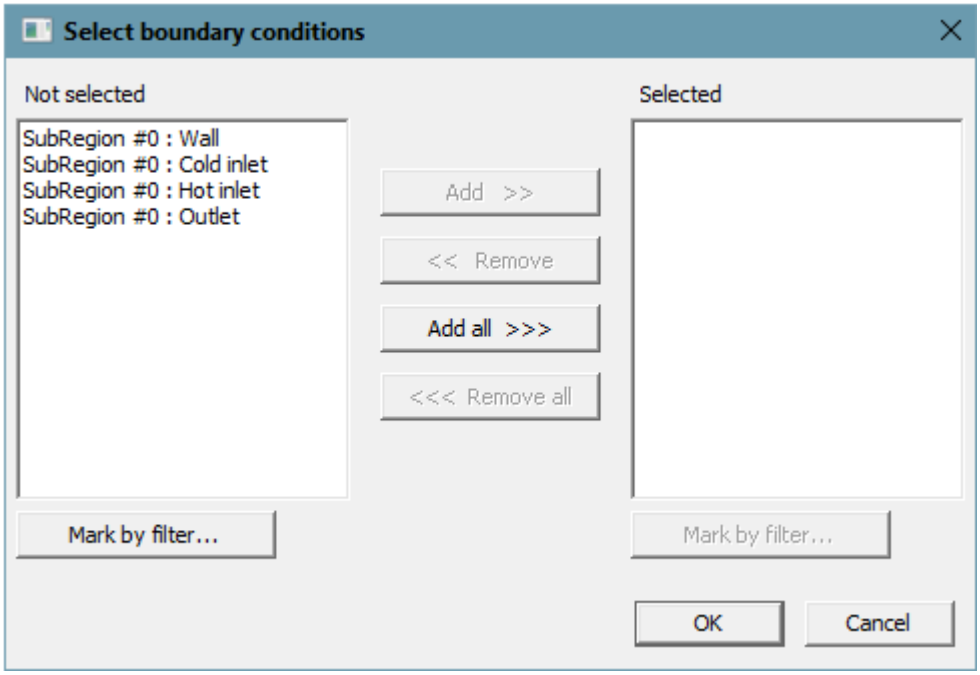
The *Level of adaptation* is the number of consecutive divisions of cells in the grid.

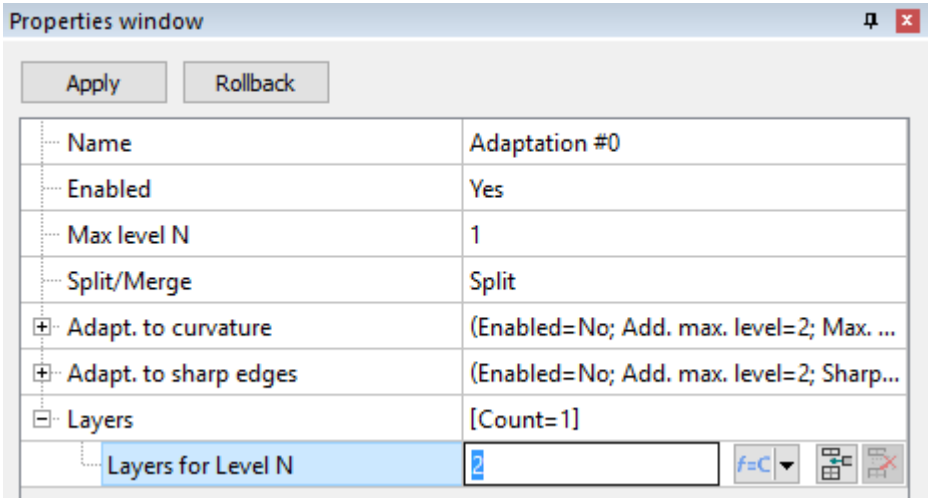
The **Adaptation** can be enabled in several parts of the computational domain: on surfaces of **Boundary conditions** or on **Objects**, in **Computational space**, or within specified **Objects**. Also it is possible to specify an *adaptation to the solution*, which activates when the specified values or gradients of values reach specified levels.

In our example, the **Adaptation** will be specified on the surfaces of the mixer's walls, and division of the cell will not be done more than one level, with two layers of adapted cells near the mixer's walls.

Do the following steps:

Step	Actions
1	Create an **Adaptation**, which will be applied on the walls of the mixer. To do so, select the **Create** command from the context menu of the subfolder **Computational grid > Adaptation**. The element **Adaptation #0** will appear in the project tree:

Step	Actions
	
2	Tune the area where **Adaptation #0** will act. Start with expanding the folder **Adaptation #0**, clicking on the "+" sign near the **Adaptation #0** element in the project tree: 
3	As our sample project contains only one subregion, **SubRegion #0**, this subregion will be automatically presented as a child element in the **Adaptation > Adaptation #0 > Subregions** subfolder. Ensure that it is so by clicking on the "+" sign near this subfolder. The subfolder **Adaptation > Adaptation #0 > Subregions** will expand and you will see that it contains the **SubRegion #0** element. 
4	From the context menu of the element **Adaptation > Adaptation #0 > Objects** select the **Add/Remove Boundary Conditions** command. The **Select boundary conditions** dialog box will open:  Select **SubRegion #0 : Wall** from the **Not selected** pane, click **Add**. **SubRegion #0 : Wall** will move into the **Selected** pane. Click **OK**.
5	In the **Properties** window of the element **Computational grid > Adaptation > Adaptation #0**: • for the parameter **Enabled** keep the existing value **Yes** • for the parameter **Max level N**, specify its value as **1** • for the parameter **Layers > Layers for Level N**, specify its value as **2**

Step	Actions
	 Click the **Apply** button.

5.7 Specifying simulation controls

Simulation controls are specified in the **Solver** tab of the project tree.

In the sections below we review specifying the following simulation controls, which will be used for our example of the mixer:

- the time step of integration of the model's equations
- stopping conditions for the computation
- specifying the data autosave parameters

5.7.1 Specifying the time step

The **Time step** is specified in the **Properties** window of the element Time step.

The **Time step** can be specified:

- to be a constant
- by the Courant-Friedrichs-Lewy number

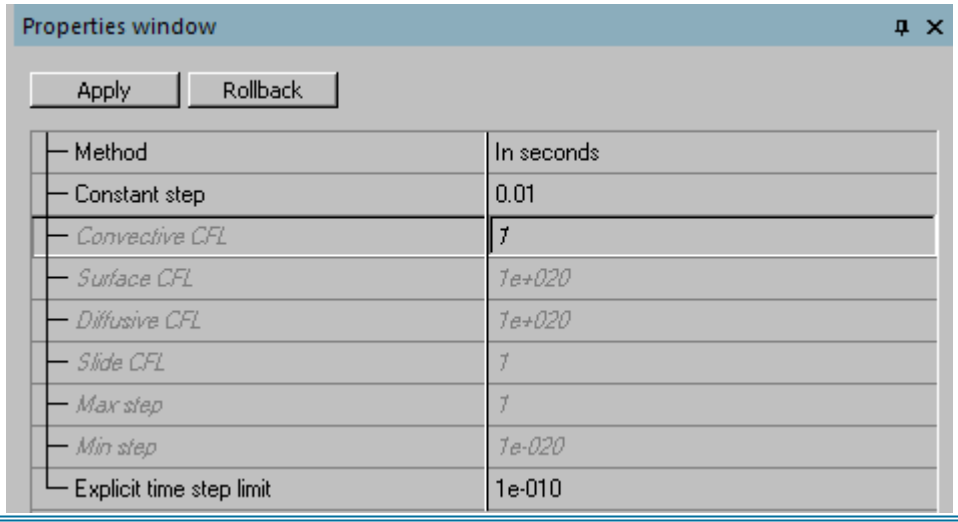
It is possible to specify the **Constant step**, which is equal to 1/10 of the transit time.

The transit time is the time that is required for a hypothetical particle, which moves with an average velocity of the flow, to pass the characteristic dimension.

The characteristic dimension for a flow in a tube is the length of the tube. And for a problem of flow around a flight device, the characteristic dimension can be assumed to be equal to the size of the flight device along the flow, and the characteristic velocity would be the velocity of the forward flow.

When specifying the **Time step**, you define:

- a **Method** of specifying the **Time step**
- parameters for the selected **Method**

Step	Actions
1	In the **Solver** tab, select the element **Time step**, and, in its **Properties** window, specify: • for the parameter **Method** select its value as **In seconds** • for the parameter **Constant step** specify its value as **0.01** [s] 
2	Click **Apply**.

5.7.2 Specifying the stopping conditions

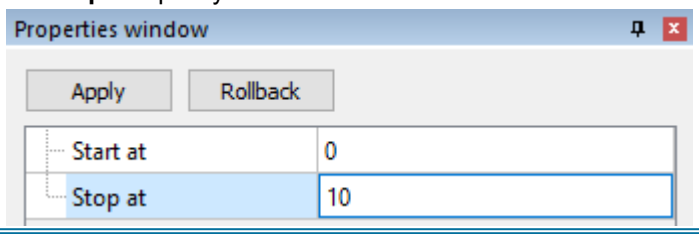
Computation can be stopped:

- manually by the button **(Stop computation)** in the toolbar **Network**.
- *or*, if you want, you can specify **Stopping conditions** to stop the computation automatically (see sections Stopping conditions for the calculation and Folder «Stopping conditions»). Particularly you can define the following **Stopping conditions**:
  - **Length of time** stops the computation after specified time
  - Number of **Time steps** stops the computation after specified number of steps
  - **Residuals** stops the computation after all residuals of calculated variables are below the specified levels*)

- **Custom values** stops the computation after all residuals of user variables are below the specified levels*)

*) You can view plots of residuals and/or values of calculated or user variables in the **Plot** tab of the **Monitor** window.

In our example the computation stops by time. Specify the **Stopping conditions**:

Step	Actions
1	In the project tree, in the **Solver** tab, select the element **Stopping conditions** > **Time span** and in its **Properties** window: • ensure that parameter **Start at** is **0** (its default value) • for the parameter **Stop at** specify its value as **10** 
2	Click **Apply**.

5.7.3 Specifying the data autosave parameters

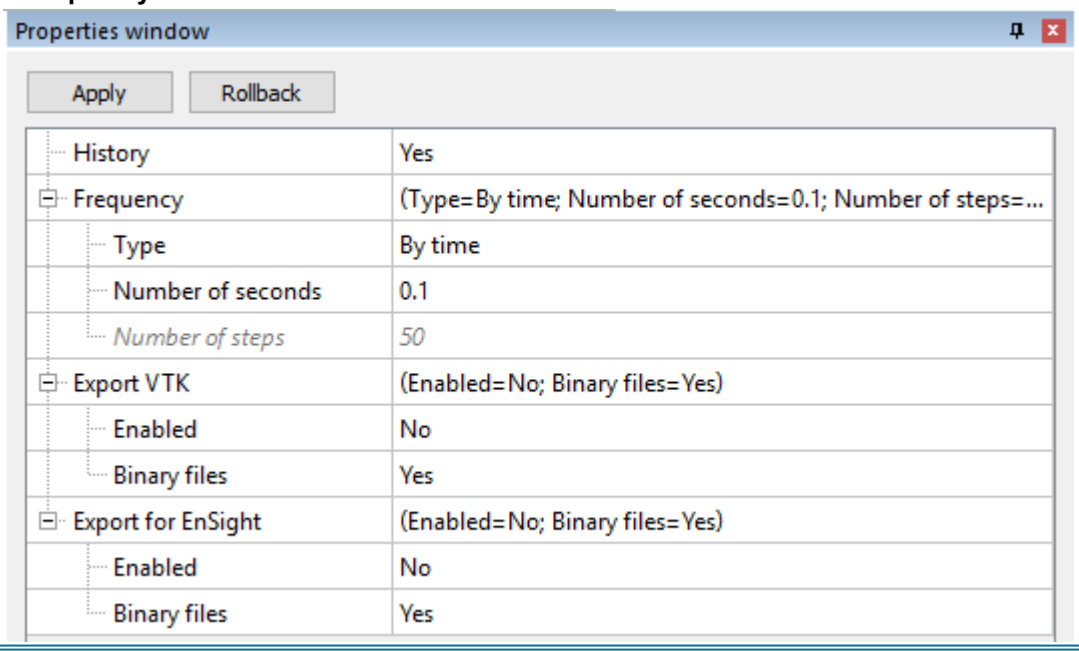
The results of the computation can be saved automatically (with a specified frequency) during the computation.

It is possible to save the data and/or layers.

Saving the data is configured in the **Data autosave** element of the project tree.

Saving the layers is configured in the **Layers autosave** element of the project tree.

Specify the data autosave parameters for our example of the flow in a mixer:

Step	Actions
1	In the project tree, in the **Solver** tab, select the element **Data autosave** and in its properties specify: • **History = Yes** • **Frequency >Type = By time** • **Frequency > Number of seconds = 0.1** 
2	Click **Apply**.

5.8 Specifying the visualization

To visualize results of the computation, we specify **Characteristics** in the **Preprocessor** tab of the project tree and **Layers** in the **Postprocessor** tab.

See sections below:

5.8.1 Creating characteristics

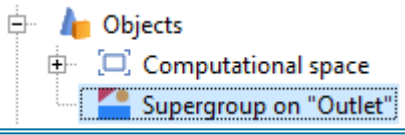
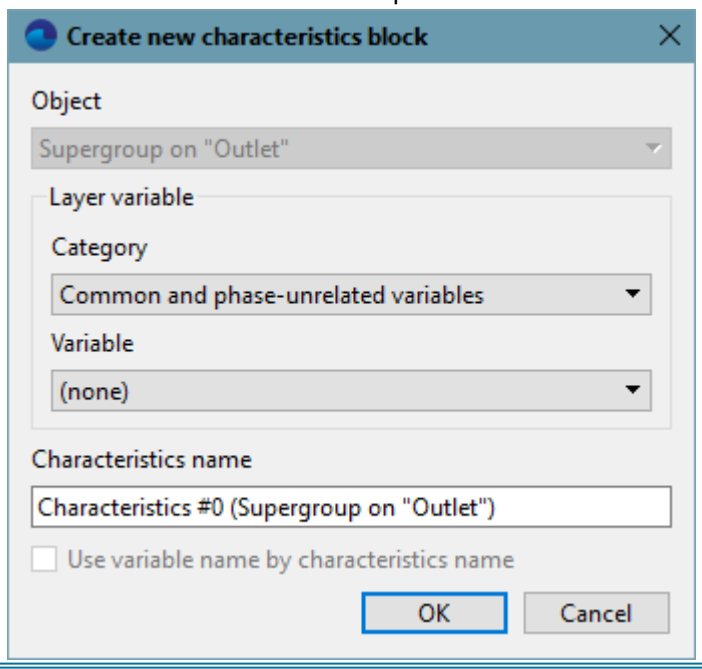
To monitor the transition of the output flow to the steady-state mode, we will watch temperature at the outlet of the mixer.

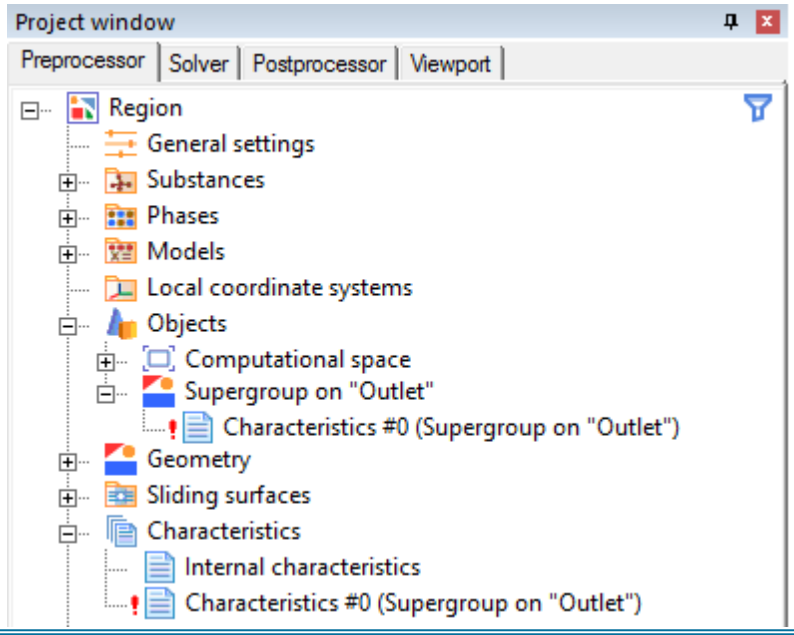
First you have to specify an **Object**, on which the **Characteristics** is to be built. To do so you should select a section of the mixer's outlet tube and define a Supergroup element on it that will contain facets, on which the **Outlet** boundary condition is to be set.

Then you have to specify the **Characteristics** to be calculated. **Characteristics** are a set of integral values (including averages), which are calculated as integrals of some **Variable** by surface or volume of some **Object**. So, when you create a **Characteristics** object, you have to specify appropriate **Object** and **Variable**.

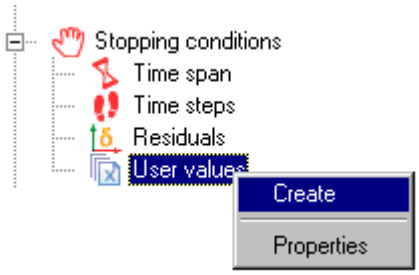
Using the fact that the component of the **Characteristics**, which is used in the stopping criterion, is displayed as a plot in the **Monitor** window, let's specify a stopping condition based on average value of the **Variable** (the average is a component of the **Characteristics**).

Creating a Supergroup and a Characteristics object on it

Step	Actions
1	Open the context menu of the element **Subregions > SubRegion #0 > Boundary conditions > Outlet** in the **Preprocessor** tab of the project tree, and select the command **Create supergroup > In Preprocessor**. A new **Supergroup** will appear in the project tree (the element **Objects > Supergroup on "Outlet"**): 
2	Open the context menu of the element **Supergroup on "Outlet"** and select the command **Create characteristics**. A dialog box **Create new characteristics block** will open: 

Step	Actions
3	Click **OK**. In the project tree a new element **Characteristics #0 (Supergroup on "Outlet")** will appear: 
4	Select the element **Characteristics #0 (Supergroup on "Outlet")**, and in its **Properties** specify **Variable > Variable** = **Temperature**, and then click **Apply**.

Specifying a stopping condition with a criterion based on the Characteristics

Step	Actions
1	Open the context menu of the element **Stopping conditions > User values** in the **Solver** tab of the project tree, and select the command **Create**:  In the project tree a new stop criterion **Stopping conditions > User values > Stop criterion #0** will appear: 
2	Select the element **Stop criterion #0**, and, in its **Properties** window, do the following: • For the parameter **Object** select its value as **Characteristics #0 (Supergroup on "Outlet")** • For the parameter **Variable** select its value as **<f mass->**, thereby you set the calculation of average temperature of the water, which outflows from the mixer's outlet). Selection of **<f mass+>** would cause calculating the average temperature of the water, which *inflows* from the mixer's outlet, and would give a zero result. • Specify **Color=Teal** (select Teal from the drop-down list). Click **Apply**.

Step	Actions
	This **Stop criterion** has been created only to display a plot in the **Monitor** window. If you need to stop the computation, you have to specify parameters **Level**, **Averaging**, and **Period** of the **Stop criterion** (see section Folder «Stopping conditions»).

5.8.2 Creating layers

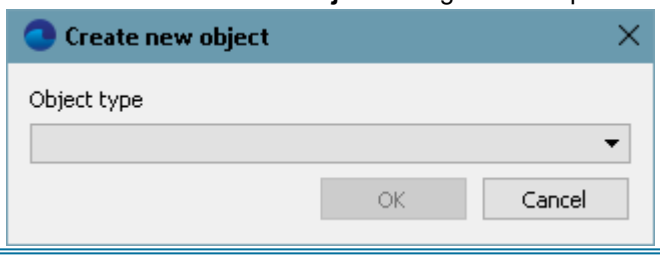
To visualize the computation, let's view **Temperature** and vectors of **Velocity** vectors in two cross-sections of the mixer.

First we need to create two **Planes**:

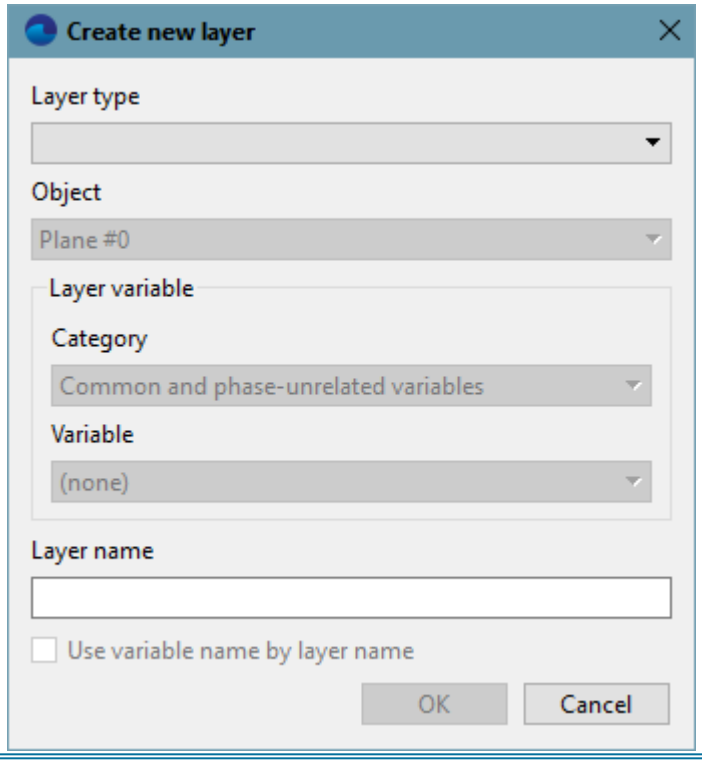
- one of the **Planes** is the vertical plane, which is parallel to the supplying tubes and passes through the mixer's axis of symmetry
- another **Plane** is a horizontal plane that passes through central axes of the supplying tubes

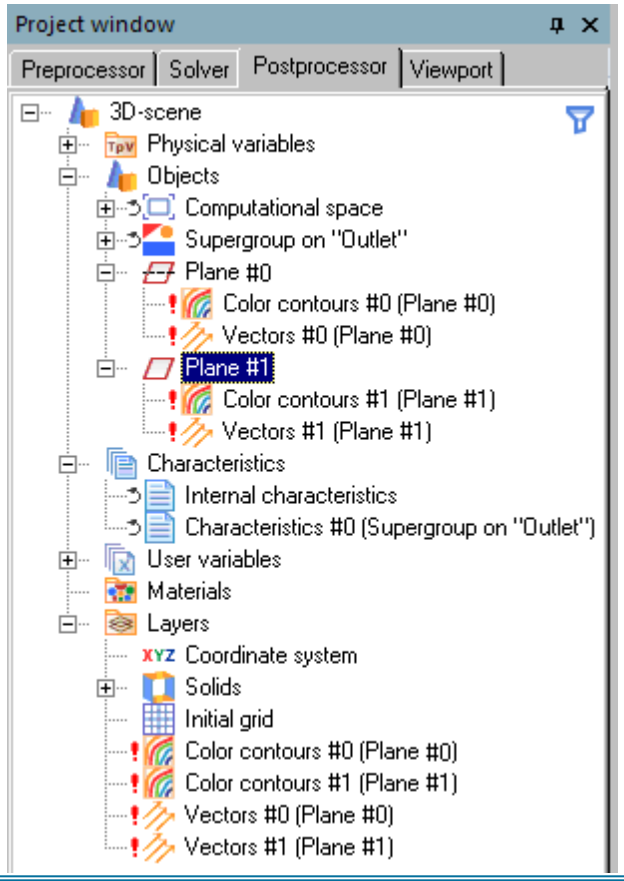
Then, on each of these **Planes** we create **Layers** upon which we define visualizations of **Temperature** (as color fill) and **Velocity** (as vectors). Each **Layer** displays in a specific way only one variable on only one object. So you have to create two **Layers** on one **Plane** and two other **Layers** on another **Plane**. In each of these pairs of **Layers**, one of the **Layers** will display a field of **Temperature**, and the other **Layer** will display the **Velocity**'s vectors.

Creating the second Plane and changing its position; restoring position of Plane #0

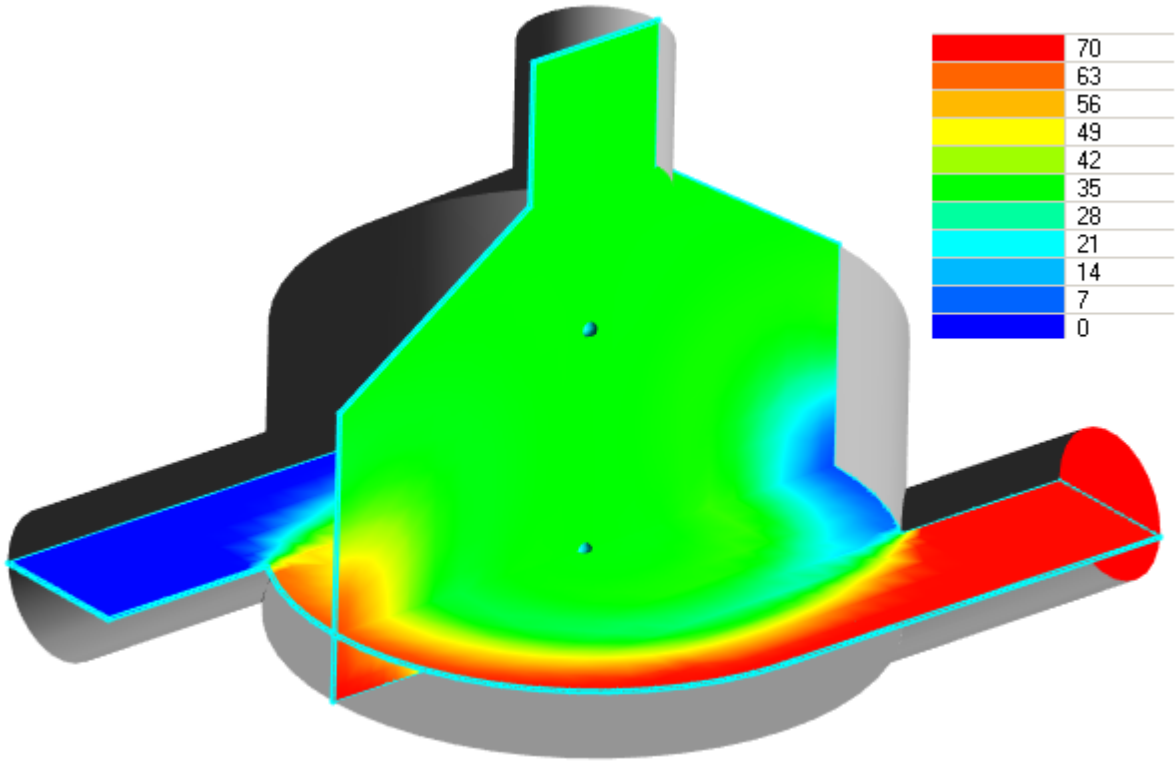
Step	Actions
1	Open the context menu of the element **Objects** in the **Postprocessor** tab of the project tree, and select the command **Create**. The **Create new object** dialog box will open: 
2	Select **Plane** from the drop-down list **Object type**, and then click **OK**. A new plane will appear in the project tree, the element **Plane #1** in the folder **Objects**, which can coincide with **Plane #0**, which has been automatically created by the program.
3	Select the element **Plane #1** and in its **Properties** window do the following: • Click the button **Object > Operations >** and then click **Apply**: **Plane #1** will become horizontal. • For the parameter **Object > Reference point > Y** specify its value as **0.005** and then click **Apply**: **Plane #1** will move along the Y-axis and will pass through axes of supplying tubes.
4	When making exercises in section Cross-section of a geometry model with a Plane, you possibly changed the position of **Plane #0**. To restore the initial position of **Plane #0**, do the following in its **Properties** window: • For the parameter **Object > Reference point > X** specify its value as **0**. • For the parameter **Object > Reference point > Y** specify its value as **0.02**. • For the parameter **Object > Reference point > Z** specify its value as **0**. Click **Apply**. Click the button **Object > Operations >** . Click **Apply**. Make sure that the parameter **Normal > X** has value **1**. If this parameter has value "**-1**", click again and click **Apply** again.

Creating and specifying the Layers

Step	Actions
1	Open the context menu of the element **Objects > Plane #0** in the **Postprocessor** tab of the project tree and select the command **Create layer**. A dialog box **Create new layer** will open: 
2	Select **Color contours** from the **Layer type** drop-down list. Click **OK**.
3	Open the context menu of the element **Objects > Plane #0** in the **Postprocessor** tab of the project tree and select the command **Create layer**.
4	In the dialog box, which will open, select **Vectors** from the drop-down list **Layer type**, and then click **OK**.
5	Repeat steps **1-4** for the element **Objects > Plane #1**. The four recently created **Layers** will be visible in the project tree:

Step	Actions
	
6	Specify parameters of both layers **Color contours**, selecting each of them in the project tree, and specifying in its **Properties** window: • for the parameter **Variable > Variable** select its value as **Temperature** • for the parameter **Value range > Mode** select its value as **Manual** • for the parameter **Value range > Max** specify its value as **70** • for the parameter **Value range > Min** specify its value as **0** Click **Apply**.
7	Specify parameters of both layers **Vectors**, selecting each of them in the project tree, and specifying in its **Properties** window: • for the parameter **Variable > Variable** select its value as **Velocity** • for the parameters **Grid > Size 1** and **Grid > Size 2** specify their values as **21** • for the parameter **Scaling > Mode** select its value as **Manual** Click **Apply**.
8	To display in the **View** window a legend with correspondence of colors in the **Color contours** to the temperature, specify in properties of any of the **Color contours** the following parameters: • **Palette > Appearance > Enabled = Yes** • **Palette > Appearance > Horiz. alignment = Right**

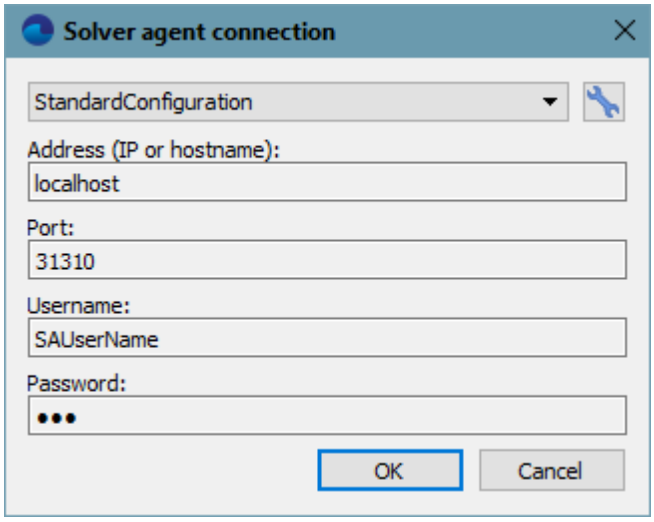
For both **Layers**, displaying distribution of **Temperature**, a manual mode is specified for the temperature range (from 0 to 70 degrees). This allows you to display the temperatures in one scale, which is common for both these **Layers** (see illustration below; you can see that temperatures on both **Planes** near the line of intersection of the **Planes** are displayed with the same colors).

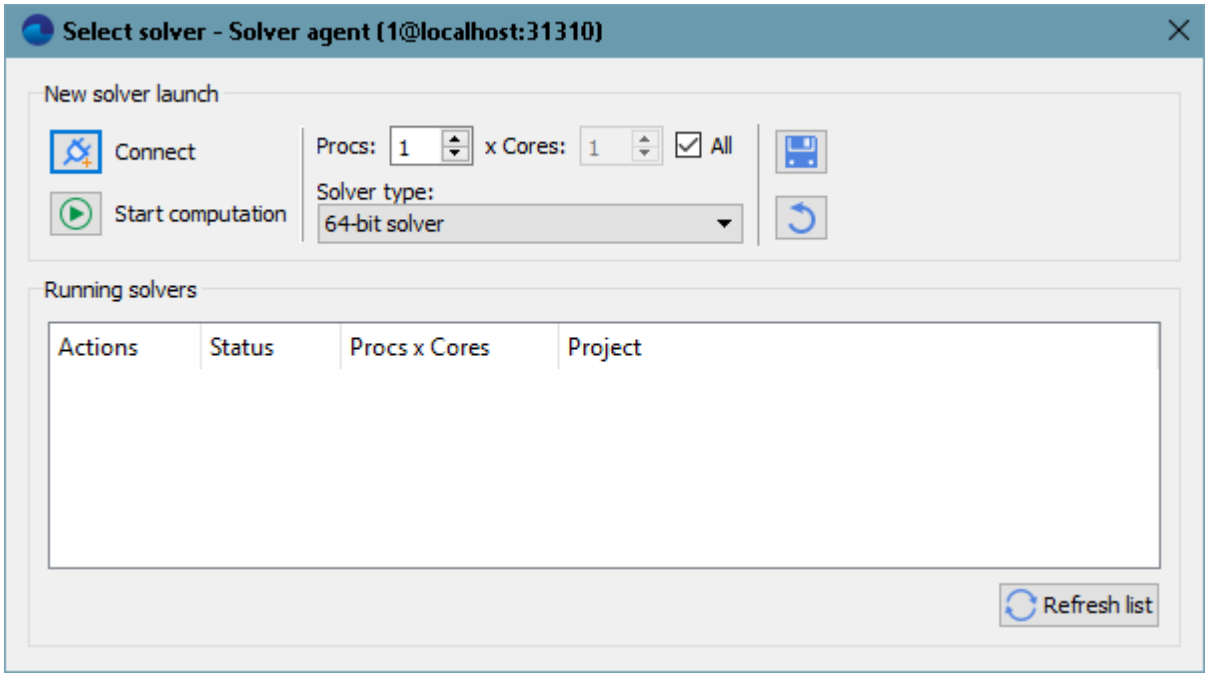
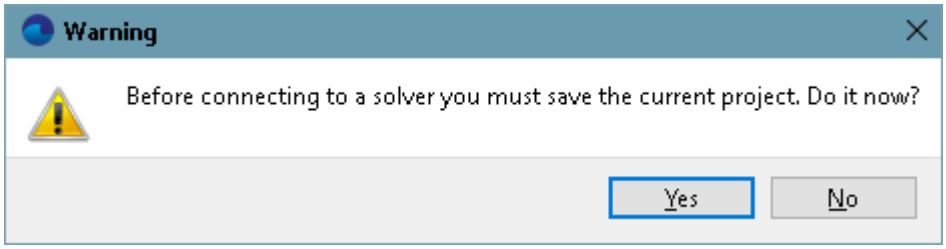
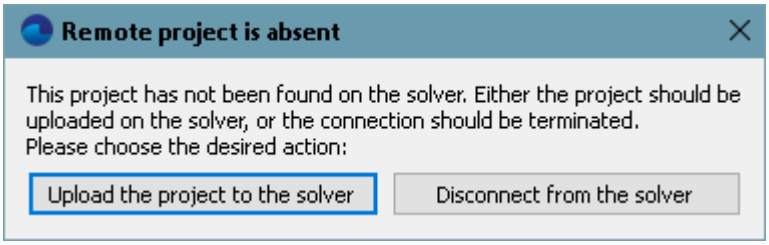


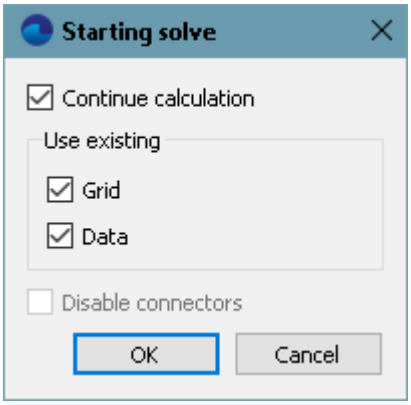
Visualization of the temperature distribution on two intersecting planes with a common value range from 0 to 70 degrees

5.9 Starting the project's computation

Run the project for computation:

Step	Actions
1	Connect to **Solver-Agent**: • Click the button (**Solver agent log in**) in the **Network** toolbar. • The **Solver agent connection** dialog box will open; select there a **Configuration** for connection to **Solver-Agent**  • Click **OK**. When a correctly tuned **Configuration** is set as the default one, connection to **Solver-Agent** will be done automatically at start of **Pre-Postprocessor**. In this case proceed with the next step. Connection of a user to **Solver-Agent** is only possible after registering the user on it. The connection is only possible when **Solver-Agent** is running (it has been already started). See details in sections: • Connection to Solver-Agent and user authentication on Solver-Agent • Registering a new Solver-Agent user • Registration data (profile) of Solver-Agent user and their change
2	Load the project on **Solver**: • Click the button (**Open solver selection window**) in the **Network** toolbar. The **Select solver** dialog box will open:

Step	Actions
	 • In this dialog box, in the **New solver launch** group of settings, specify parameters of the new **Solver**, which will be launched. • Then click there the  (**Start computation**) screen button. This will cause running a new **Solver** and loading the current project on it.
3	The program will request (**Before connecting to a solver you must save the current project. Do it now?**) to save the current project:  Click **Yes**.
4	A message about the absence of the remote part of the project on the server (**Remote project is absent**).  Click **Upload the project to the solver**.
5	The **Starting solve** dialog box will open:

Step	Actions
	 Uncheck all checkboxes there and click **OK**. Computation of the current project will start on the **Solver**.
6	When necessary, you can stop the computation by clicking the **(Stop computation)** button in the toolbar **Network**.

See section Starting solve, stop and resuming the project's computation for details.

5.10 Viewing results of the computation

Let's get to know how to view the computation results:

- viewing the basic parameters of the computation in the Monitor window (including making a plot of a stopping condition over time)
- viewing the layers during the computation
- viewing characteristics in the Info window
- creating an animation

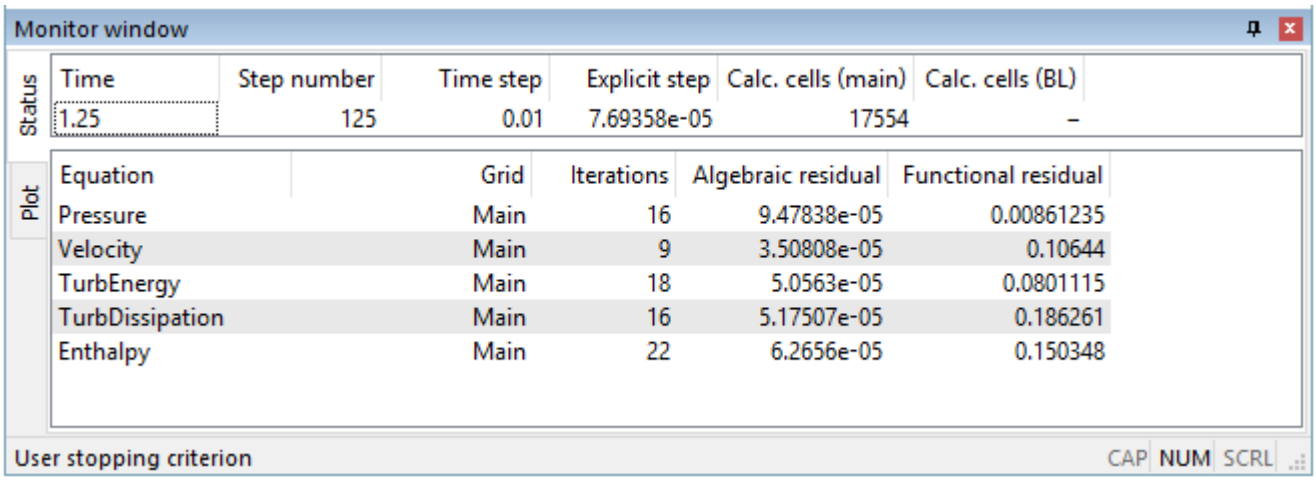
5.10.1 Viewing data in the Monitor window

The **Monitor** window has two tabs with data which you can view during the computation:

a) In the **Status** tab you can track changes of the following parameters:

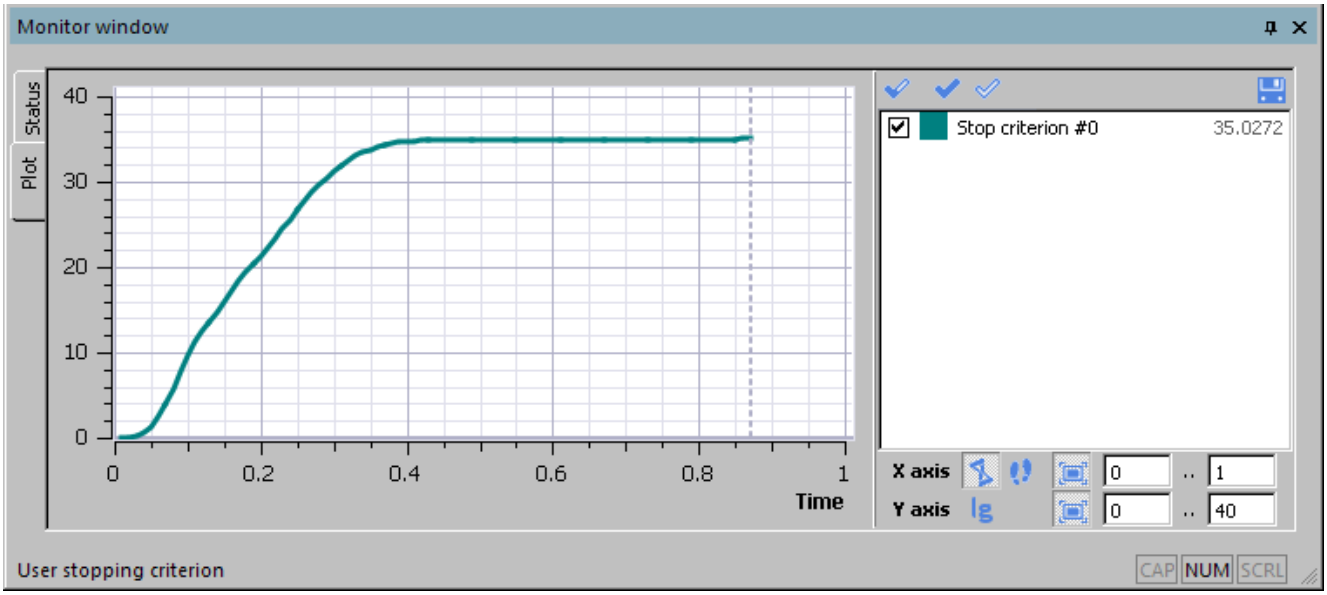
- **Time**
- **Step number**
- other data, list of which is tuned from the context menu opened when you right-click the table's header (**Time step**, **Explicit step**, **Surface step**, **Diffusive step**, **Slide step**, **Exchange step (FSI)**, **Exchange number (FSI)**, **Cells (main)**, **Calc. cells (main)**, **Cells (BL)**, **Calc. cells (BL)**)
- **Equation** (displays the list of main calculated variables)
- **Grid** (informs about which computational grid is used, the main grid or the boundary layer grid)
- **Iterations** (number of iterations that were done)
- **Algebraic residual**
- **Functional residual**

(see details in the section Window «Monitor»)



Displaying parameters in the "Status" tab

b) In the **Plot** tab you can track changes of a reference parameter over time as a plot.

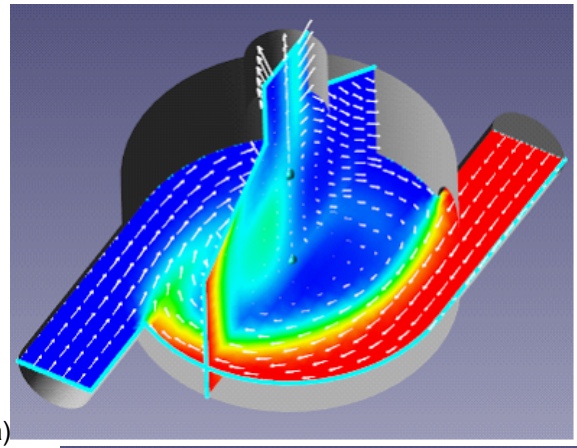
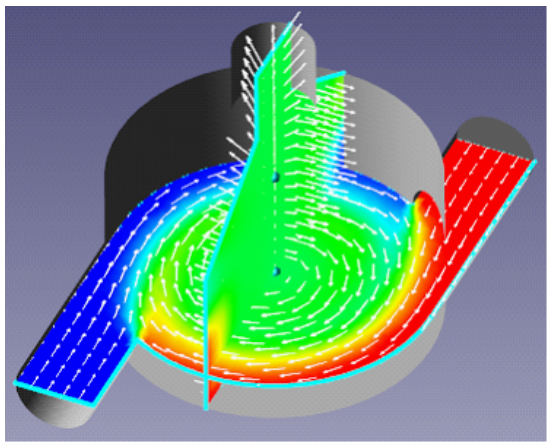


Displaying a plot of a reference parameter over time

You can shift and/or scale the plot, display the axis of ordinates in either uniform or logarithmic scale, display the dependency on either time or number of steps (see details in the section Window «Monitor»).

5.10.2 Viewing the layers during the computation

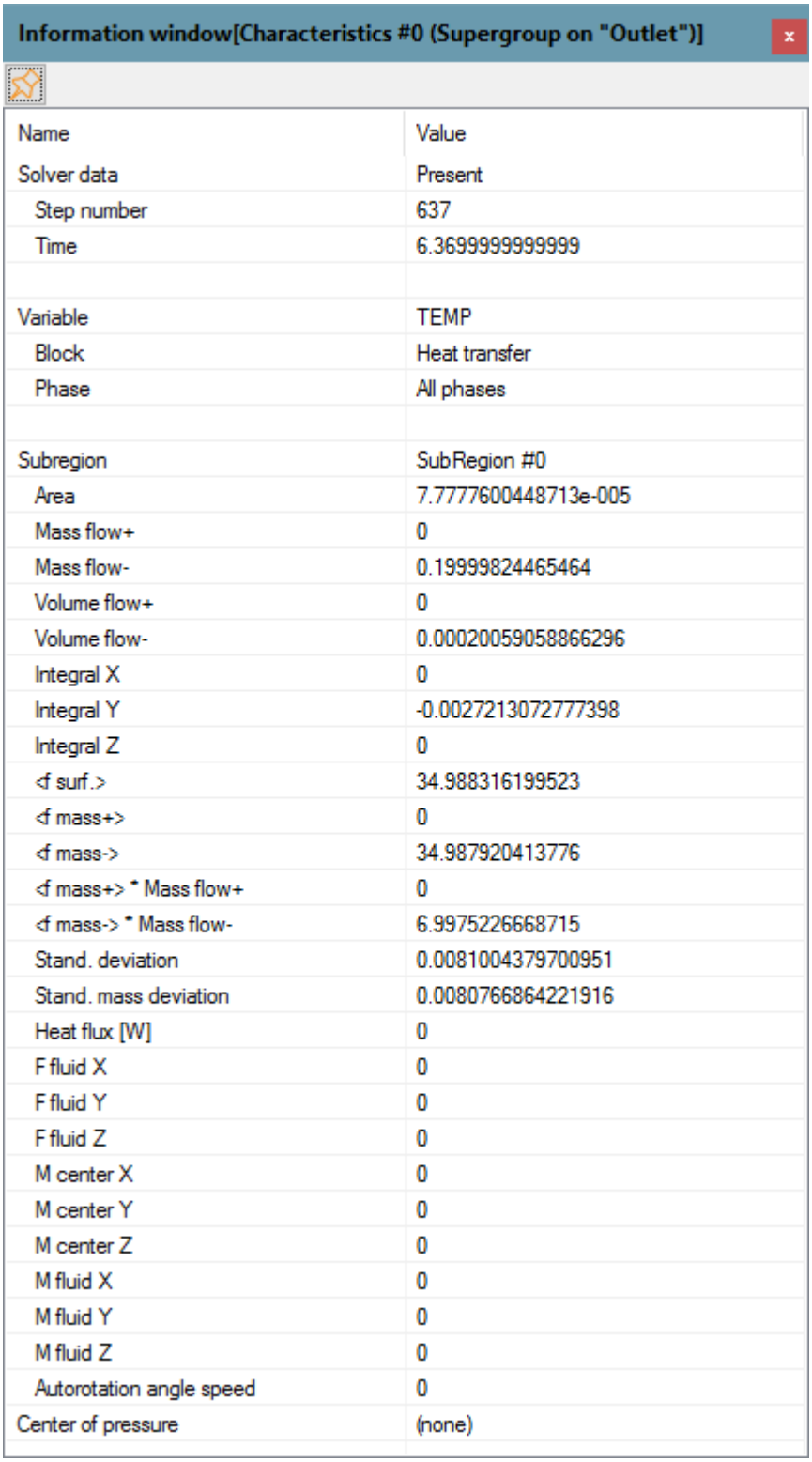
Step	Actions
1	Click the button (**Show/hide surface facing viewer**) in the toolbar **Solids**, to make the object's outer surfaces (which enclose the interior space) transparent. This allows you to see the layers on the cut **Planes** (see illustration).

Step	Actions
	 a) b)  Displaying layers with the temperature field and with vectors of velocity: a) at an intermediate time point; b) the final result
2	Click the button (**View transformation mode**) in the toolbar **Work modes** and adjust the most convenient perspective for watching the flow.

5.10.3 Viewing characteristics in the Info window

Select the element Characteristics #0 in the project tree and then click the button (**Show info window for selected object**) in the toolbar **Work modes**.

The **Info** window will open displaying parameters of these Characteristics calculated on the last time step (values of the parameters are updated at the end of the next time step).



Information window[Characteristics #0 (Supergroup on "Outlet")]

Name	Value
Solver data	Present
Step number	637
Time	6.3699999999999
Variable	TEMP
Block	Heat transfer
Phase	All phases
Subregion	SubRegion #0
Area	7.7777600448713e-005
Mass flow+	0
Mass flow-	0.19999824465464
Volume flow+	0
Volume flow-	0.00020059058866296
Integral X	0
Integral Y	-0.0027213072777398
Integral Z	0
<f surf.>	34.988316199523
<f mass+>	0
<f mass->	34.987920413776
<f mass+> * Mass flow+	0
<f mass-> * Mass flow-	6.9975226668715
Stand. deviation	0.0081004379700951
Stand. mass deviation	0.0080766864221916
Heat flux [W]	0
F fluid X	0
F fluid Y	0
F fluid Z	0
M center X	0
M center Y	0
M center Z	0
M fluid X	0
M fluid Y	0
M fluid Z	0
Autorotation angle speed	0
Center of pressure	(none)

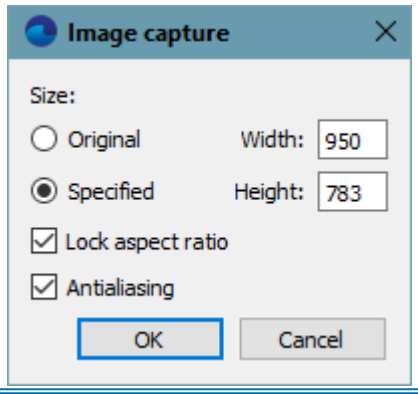
The **Info** window with parameters of **Characteristics #0**

5.10.4 Making an animation

You can make an animation from a set of graphic files created either during the computation or when viewing the saved data. These graphic files contain images from the **View** window.

Here we describe making an animation in the **Pre-Postprocessor** during the computation.

Note: There are also similar animation possibilities in **Viewer**.

Step	Actions
1	Create the **Layers** which images you want to save (see section Creating layers; you can leave the **Layers**, displaying the temperature and velocity, which were created there, or you can delete them and create other **Layers**). On the **Layers**, it is recommended to specify **Value range > Mode** = **Manual**, so the range and coloring of the **Layers** do not change from step to step.
2	Click the button (**Start capturing an image sequence**) in the toolbar **Capture**. The **Image capture** dialog box will open: 
3	In the **Image capture** dialog box specify the parameters of the image (size, antialiasing), and then click **OK**. (see the detailed description of the elements in this dialog box in section Controlling visual capture).

4	A standard operating system's window for access to files will open. Specify the location, type, and filename prefix for the files, which will store the images. Then click the **Save** button.
	The images will be saved in the files at each change of the **View** window.
	The file names are formed as follows:
	`(prefix)_00000.(type)`
	`(prefix)_00001.(type)`
	`(prefix)_00002.(type)`
	etc.

5	Create a video from this sequence of saved graphic files using an external (third party's) video software.

6 Principal concepts of FlowVision

See sections:

6.1 Operating procedures with FlowVision

Generally the simulation, which is done by any computational fluid dynamics software package, assumes the following steps:

1. Analysis of setting of the problem, simplification, selecting a physical model
2. Preparing a computational project
3. Finding correct parameters of the mathematical model (research of the solution's convergence depending on the time step, computational grid, size of the computational domain, details of the geometry model). As a rule, this step is done once for a group of similar problems.
4. Making a final simulation or optimization research
5. Analysis of obtained results

FlowVision makes it possible to simulate a large class of continuous-medium flows. Any CFD problem is basically defined by:

- properties of the medium
- physical processes occurring in the medium
- conditions on the boundaries of the analyzed zone of the flow

Simulation a certain variant of flow, which assumes forming and carrying out a computational project. This consists of the following steps:

- creation initial data for the computation
- carrying out the computation, monitoring the calculated values
- analysis of computation results
- if necessary, changing the project and either continuing the computation or starting a computation of a new project

The initial data for computation are set by doing the following:

1. Creating, in an external CAD system, a geometry model of the computational domain (referred to as "geometry" for brevity), which defines boundaries of the computational domain and initial conditions
2. Creating a project or opening an existing project
3. Setting parameters of computational models in subregions
4. Setting boundary conditions and boundary links
5. Setting moving bodies and modifiers
6. Setting the computational grid and parameters of adaptation
7. Setting parameters to control the computation (time step, computation method parameters)

Performing the computation consists of the following steps:

- starting the project computation
- monitoring of the computation's parameters and results of the project's computation (this is done using facilities for analysis of the results). *FlowVision* can analyze the results online and does not require to wait the ending of the computation.

On the step of the results' analysis you need to specify and configure the graphical and numerical analytical facilities provided by **Pre-Postprocessor**.

In **Pre-Postprocessor** the user carries out the steps of forming the initial data for the computation and the step of results' analysis.

The computation is performed by the **Solver** module, which has no graphical user interface. Monitoring of the computation's parameters and project computation's results is done in **Pre-Postprocessor** and in **Viewer**.

6.2 Variables (physical, integral, and user)

FlowVision can use the following types of *variables*:

- **Physical variables** that are defined in the computational domain in cells or on boundaries of cells. These variables are defined locally, this means that they depends on coordinates.
- **Integral variables** that do not depend on coordinates (they are defined globally or are calculated in some volume or on some surface). Particularly, the values that are calculated in characteristics, are integral variables.
- **User variables** can be created by users themselves from physical, integral, and other user variables. To create user variables, the **Formula editor** is applied, in which you can use mathematical formulae, logical operators, conditional expressions, and mathematical constants (e, π).

Based on physical and user variables, the program can calculate time-averaged variables.

6.2.1 Physical variables

In *FlowVision* the **Physical variables** are the following values defined in the computational domain (or in some its **Subregions**):

- the main variables of computational models
- algebraic combinations of the main variables
- coordinates

The *physical variables* (except coordinates) are calculated when solving the model's equations at every time step *in every cell of the computational grid*, so they are referred as local variables as opposed to the integral variables.

The physical variables can be used:

- for visualization of the simulation's results
- for calculation of **Characteristics**
- in formulae (for example, to define **User variable** or **Boundary conditions**)

Physical variables used in visualizations, for calculation of Characteristics and for calculations in formulae

Name in *FlowVision*	Description	Dimension
Q-criterion	Q-criterion is defined by the formula: $$Q = \frac{1}{2}\left(W^2 - S^2\right)$$ where S is **S-criterion** and W is **W-criterion** (see below). **Q-criterion** allow you to visualize areas where either shear flow or vortical flow dominates. Positive values of **Q-criterion** correspond to areas where vortical flow dominates, while negative values correspond to areas where shear flow dominates.	$[s^{-4}]$
S-criterion	Invariant of the strain rate tensor: $$S = 2\sum_{i,j} S_{ij}S_{ji}$$ $$S_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right)$$	$[s^{-2}]$
W-criterion	Contraction of the velocity gradient tensor: $$W = 2\left\lvert\sum_{i,j} W_{ij}W_{ji}\right\rvert$$	$[s^{-2}]$

Name in *FlowVision*	Description	Dimension
	$$W_{ij}=\frac{1}{2}\left(\frac{\partial V_i}{\partial x_j}-\frac{\partial V_j}{\partial x_i}\right)$$	
Y_plus	Dimensionless distance from the wall. The variable **Y_plus** is defined for walls only, so it is calculated for those **Layers** and **Characteristics** only, which are built on surfaces of walls.	
Viscosity	Molecular dynamic viscosity	[Pa s] = [kg·m^{-1}·s^{-1}]
TurbViscosity	Turbulent dynamic viscosity	
Pressure	Relative pressure	[Pa]
Pressure (+hydrostatic)	Relative pressure + hydrostatic pressure	
PressureTotal	Total pressure	
Shear stress **Shear stress (vector)**	These are scalar and vector values of the mechanical shear stress (the viscous force acting on a solid surface specific per unit of area). The scalar variable **Shear stress** is calculated as length of the vector of tangential stress on the section of a computational grid's cell, which was cut by the specified surface (this value is calculated only on sections formed by the geometry). ⚠ When account of the pressure gradient in settings of the turbulence model is enabled, **Shear stress**, in some rare cases, can be negative (this effect appears in some cells because of some specifics of the used numerical method). The vector variable **Shear stress (vector)** is calculated as weighted vector sum of shear stress vectors by all sections of a computational cell that were cut by the geometry, normalized by the total area of sections cut by all surfaces, using the formula: $$\boldsymbol{\tau}=\frac{\sum_i \boldsymbol{\tau}_i S_i}{\sum_i S_i}$$ where: $\boldsymbol{\tau}_i$ is the shear stress vector formed by one of sections of the computational cell by geometry surfaces, S_i is the area of a section of the computational cell that was cut by one of the geometry surfaces, and the sum is calculated over all sections of the computational cell that are cut by the geometry surfaces.	
Molar mass	Molar mass	[kg·mol^{-1}]
VOF	The variable **VOF** (volume of fluid) presents the part of the cell's volume occupied by the continuous phase and dispersed phases contained in it (see section Theory > Physical processes > Phase transfer).	
Radiation density	Density of radiation's energy	[W·m^{-2}] = [kg·s^{-3}]

Name in *FlowVision*	Description	Dimension
Prandtl	Molecular Prandtl number	
PrandtlTurb	Turbulent Prandtl number	
Distance to wall	Distance to the wall	[m]
Velocity	Velocity (vector)	[$m \cdot s^{-1}$]
Velocity (disp.)	Velocity of dispersed phase (vector)	
Conservative velocity	Conservative velocity is calculated based on conservative velocities on faces of computational cells. If **Conservative velocity** differs too much from ordinary **Velocity**, this means that the time step is too large or you have some issues with problem setting. Examine the computational grid and parameters of the mathematical model. Big difference between **Velocity** and **Conservative velocity** can appear due to **Viscosity**.	
Velocity divergence	Velocity divergence	[s^{-1}]
Reaction rate	Rate of the combustion reaction	[kg m^{-3} s^{-1}]
Temperature	Temperature (relative) in a cell	[K]
Wall temperature	Temperature (relative) on a part of a surface that goes through a cell **Wall temperature** is defined on the following surfaces: • on a boundary condition **Wall**, in properties of which **Variables > Temperature = Flux | External heat exchange | Radiation equilibrium** • on a boundary condition **Wall, ablation** • on a connected boundary conditions **Conjugated ablation** • on a phase interface surface between a solid body and gas when ablation is simulated using the VOF method Note that values of the variable **Temperature** on the above mentioned surfaces might differ from values of the variable **Wall temperature** because **Temperature** is taken from centers of cells adjacent to the surface while **Wall temperature** is taken immediately from the surface.	
Temperature (disp.)	Temperature of the dispersed phase (relative)	
TemperatureTotal	Total temperature (relative)	
HeatFlux	Specific (per unit of area) heat flow, passing through the surface (*not* including the radiative heat flux, **Rad. energy flux**). It is only calculated on the surface of a **Supergroup**.	[W m^{-2}] = [$kg \cdot s^{-3}$]
Rad. energy flux	Radiative heat flux on a wall (see section Theory > Physical processes > Radiation)	
Specific heat	Specific heat	[$J \cdot kg^{-1} \cdot K^{-1}$] = [$m^2 \cdot s^{-2} \cdot K^{-1}$]
Thermal conductivity	Molecular thermal conductivity	[$kg \cdot m \cdot s^{-3} \cdot K^{-1}$]
TurbDissipation	Rate of dissipation of turbulent energy	[$m^2 \cdot s^{-3}$]
TurbDissipation specific	Specific rate of dissipation of turbulent energy	[s^{-1}]
TurbEnergy	Turbulent energy	[$m^2 s^{-2}$]
Density	Density	[$kg \cdot m^{-3}$]

Name in *FlowVision*	Description	Dimension
MachNumber	The Mach number	
Weber number	The Weber number	
Lighthill tensor convolution	These vector variables present convolution and divergence of ordinary and simplified Lighthill tensor. The Lighthill tensor in the 3D space consists of 9 components (in a 3x3 matrix). Components of this tensor, [kg·m^{-1}·s^{-1}], are defined as: $$\boldsymbol{T}_{ij} = \rho \cdot \boldsymbol{V}_i \cdot \boldsymbol{V}_j + \mu(\nabla \boldsymbol{V})_{ij}$$, where ρ is density, $\boldsymbol{V}$ the velocity vector, μ the viscosity. The simplified version of the Lighthill tensor doesn't take into account the viscosity: $$\boldsymbol{T}_{ij} = \rho \cdot \boldsymbol{V}_i \cdot \boldsymbol{V}_j$$ Convolution of the Lighthill tensor is a 3-component vector, i-th component of which is sum of all components of the i-th row of the tensor. Divergence of the Lighthill tensor is a 3-component vector, i-th component of which is calculated as follows: i-th component of gradient of the tensor's convolution is taken and then components of the gradient are summarized.	[kg·m^{-1}·s^{-1}]
Lighthill tensor convolution (simple)		
Lighthill tensor divergence		[kg·m^{-2}·s^{-1}]
Lighthill tensor divergence (simple)		
EnthalpyTotal	Total enthalpy	[m^2·s^{-2}]
Mass frac. [Substance #N]	Mass concentration of the specified **Substance**	
Molar frac. [Substance #N]	Mole concentration of the specified **Substance** (for a mix of gases this is equal to the volume concentration)	
Fuel mass frac. true	True mass fraction of **Fuel**	
Oxidizer mass frac. true	True mass fraction of **Oxidizer**	
Product mass frac. true	True mass fraction of combustion's **Products**	
Oxid. excess factor rec.	Recovered factor of **Oxidizer**'s excess	
Oxid. excess factor true	True factor of **Oxidizer**'s excess	
Variance of fuel	Variance of recovered mass fraction of **Fuel**	
Phase volume	Volume fraction of the cell occupied by the **Phase** (continuous or dispersed)	
Area	Area of a boundary condition or a surface, over which an integration is done	[m^2]
Concentration	Concentration of particles. This variable is limited from below by the value of 1 [m^{-3}], which is 1 particle per cubic meter.	[m^{-3}]
Conductivity	Specific conductivity	[S·m^{-1} = Ohm^{-1}·m^{-1}]
Diameter	Diameter of particles	[m]
Lorentz force	The Lorentz force, which is the volume force (vector) of electromagnetic field acting on a unit volume of the media	[N m^{-3}]
Electric field intensity	Electric field intensity (vector)	[V m^{-1}]
Film thickness	Thickness of film of **Substance** of the dispersed phase **Particles** on a solid surface. Physical meaning of this variable depends on what is set in properties of the	[m]

Name in *FlowVision*	Description	Dimension
	element **Physical processes** of the dispersed **Particles** phase: • when **Crystallization = (none)**: This is thickness of a static film, which is calculated in assumption that particles deposit on the surface. Calculation of this variable is done only when the **Permeable surface** boundary condition for velocity of dispersed particles and volume of the **Particles** phase. The static film is calculated only on the surface of the **Supergroup** that corresponds to the appropriate boundary condition. • when **Crystallization = Film model**: This is thickness of a dynamic film, for which, in the wet icing model, processes of motion on the surface, evaporation and crystallization are specified.	
Joule heat	Joule heat	[W m^{-3} = kg m^{-1} s^{-3}]
Mass	Average mass of a particle within the cell	[kg]
Normal	Normal vector to a surface	[m]
Permittivity	Permittivity (the dielectric capacity)	
Electrical potential	Potential of the electrical field	[V]
Magnet Potential	Potential of the magnetic field (vector)	[T·m = Wb/m = = kg·m·s^{-2}·A^{-1}]
Saturation pressure	Saturation pressure	[Pa]
Volume of cell	Volume of a cell of the computational grid	[m^3]
Clearance	Width of the gap (this variable is defined for gap cells only)	[m]
Vorticity	Curl of the velocity, $\boldsymbol{\omega} = rot(\mathbf{V})$	[s^{-1}]
Mass blow-in from boundary	The total specific mass flow rate of a **Substance** from the body surface at ablation (see section Theory > Physical processes > Ablation)	[kg s^{-1} m^{-2}]
Flow of particles	The **Flow of particles** variable can be visualized on surfaces, on which the **Permeable surface** boundary condition is set for velocity of particles. This scalar physical variable is the specific mass flow of the **Dispersed Phase** that moves through the surface or condenses on the surface. **Flow of particles** is calculated as follows: $F_d = \varphi_d \rho_d V_{nd}$ if $V_{nd} < 0$ (when the particles move towards to the surface) $F_d = 0$ if $V_{nd} > 0$ (when the particles move away from the surface) It is assumed here that the local normal to the surface is directed into the computational domain. The **Flow of particles** variable *cannot* be used for adaptations of the computational grid, for this purpose use the variable **Flow of particles (adapt.)** instead of it (see below).	[kg s^{-1} m^{-2}]
Flow of particles (adapt.)	The **Flow of particles (adapt.)** variable is identical to the **Flow of particles** variable, but it can be for adaptations of the computational grid.	

Name in *FlowVision*	Description	Dimension
Vaporization mass rate	Mass rate of vaporization or sublimation, $\dot{m}_{evap(subl)}$ according to calculation of the thermal balance on the inter-phase surface in simulation of the dispersed phase crystallization.	
Crystallization mass rate	Mass rate of the solid phase (the ice) formation $\dot{m}_s$ according to calculation of the thermal balance on the inter-phase surface in simulation of the dispersed phase crystallization.	
Icing intensity	Ratio of the mass velocity of the crystallization to density of the solid phase (the ice), $\frac{\dot{m}_s}{\rho_{ice}}$ See section Crystallization.	[m/s]
Freezing coefficient	Ratio of the mass flow of moisture (dispersed phase plus the liquid film) on the contact surface to the sum of the mass rate of crystallization and the mass rate of vaporization or sublimation, $f = \frac{\dot{m}_d + \dot{m}_f}{\dot{m}_s + \dot{m}_{evap(subl)}}$ See section Crystallization.	dimensionless
ExplicitTimeStep	The explicit time step (this is a physical variable, which is calculated individually for each computational cell)	[s]
Radiation wavelength	The wavelength λ of the radiation. This variable is used in simulations of radiation energy transfer	[m]

See also:

Correspondence of physical variables to notations used in the chapter **Theory** see in the Basic notations section and in subsections *"Notations"* of sections *"Theory > Physical processes"*:

- Motion
- Heat transfer
- Radiation
- Electro-dynamics
- Mass transfer > Mixing
- Mass transfer > Combustion
- Mass transfer > Chemistry
- Mass transfer > Ablation
- Turbulence
- Phase transfer
- Processes in the presence of dispersed medium
- Processes in clearance

6.2.2 Integral variables

Integral variables are not dependent on coordinates in the computational domain. They include:

- parameters, which are set for the whole computational domain:
  - Current time
  - Time step
  - Explicit time step
  - Time step number

- values of components of some user-defined **Characteristics**, which are set in a volume or on a surface of some **Object**

6.2.3 User variables

User variables are the user-defined functions of
- physical variables
- integral variables created as a result of calculation
- other user variables and constants

These functions are set in the **Formula Editor** which has a variety of mathematical functions.

User variables can be of the following types:
- local user variables that may depend on coordinates and time
- global user variables, not dependent on coordinates (dependent only on time)
- constant user variables, not dependent on coordinates and time

User variables can be specified in the project tree in two different tabs: in **Preprocessor** and in **Postprocessor**. The **User variables**, which has been specified in **Postprocessor**, can be used in this tab only (in **Postprocessor**).

See also: section Folder «User variables».

6.2.4 Averaged variables

Based on physical and user variables that are included in the project, the program can calculate their averaged values per time period from the specified time moment t_0 till the current time moment t.

At each point with coordinates (x, y, z) in the computational domain, value of the averaged variable f_{avg} is calculated based on value of the source (not averaged) variable f by the formula:

$$f_{\text{avg}}(x,y,z)=\frac{1}{(t-t_0)}\int_{t_0}^{t} f(x,y,z)dt$$

Calculation of averaged variables will start at the moment when the project's current time reaches the value t_0.

Correct calculation requires that you either specify t_0 greater then the project's current time or the project is starts from its beginning.

Both scalar and vector variables can be averaged.

You can use averaged variables in the same way as ordinary physical and user variables, for example:
- to visualize them in **Layers**
- to calculate **Characteristics**
- to use them in the **Formula editor**

See section Folder «Models» for details about the user interface for specifying the list of averaged variables and the t_0 value and specific details of calculating the averaged variables.

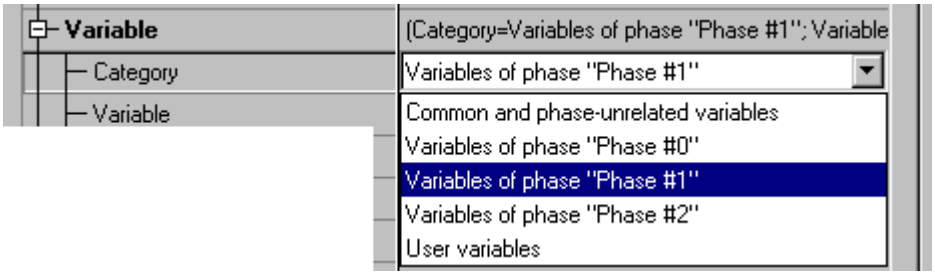
6.2.5 Categories of variables

Physical values (physical variables, integral variables, user variables) can be calculated either in a context of some specific **Phase** or without of such context.

For example, in a simulation with interaction of water and air, you can either watch the average temperature of the water or, without taking into account any phase, watch the common average temperature.

Some physical variables can be phase ones and also be general ones. Other variables can be phase ones only.

The *category* of a physical variable allow you to specify in which phase or phases the calculation of this variable will be done for a visualization **Layer**, some **Characteristics** or in a formula for calculation a user variable.



When you select a **Category**, the following options can be selected:

- **Variables of phase "Phase #N"** are all the variables, which are created by some specific **Phase #N**. Building a **Layer** or calculating **Characteristics** will be done only for those cells, where the selected **Phase** exists. For example, to calculate the average temperature of a steel tube and do *not include* into the averaging the temperature of the water within the tube, select the category **Variables of phase "Phase #N"**. The computational domain for the variables of the category **Variables of phase "Phase #N"** are the computational cells, in which the selected **Phase #N** is defined.
- **Common and phase-unrelated variables** are the variables that can be defined in several existing **Phases** or are not connected to any **Phase** at all. Computation of such a variable is done in every cell (independently on a **Phase** in it), where the variable is defined. For example, **Temperature** can be defined both within steel tube's wall and within the water in the tube, but it can be not defined in the phase outside the tube and is not defined in a **Moving body** and in vacuum. Examples of variables that are not connected to any phase at all are **Coordinates** and **Time step**.
- **User variables** are calculated based on other variables, either connected with specific **Phases** or not connected with them. **User variables** are calculated everywhere in the **Computational space** and are not associated with any **Phase**.

The category of a variable influences on the domain of integration over which **Characteristics** are calculated, see details and illustrations in the section Characteristics.

Calculation **Characteristics** for the variable **VOF** has specifics: the **Volume** value is calculated for the whole region, where the **Model** is specified, which contains the **Phase**, in which the **VOF** variable is specified (so the region includes the cells, which are occupied by another **Phase** of the **Model**).

6.3 Characteristics

A *characteristic* is a set of integral values obtained by integrating a certain variable (physical or user local) for the surface or in the volume of an object. Setting a characteristic is limited to indicating the variable and an **Object** for which surface or volume integration is to be performed.

Numeric values of a characteristic's modules are displayed in the Info window and are updated according to the results of every time step computation. Observing in the **Info** window how a characteristics is being changed allows you to control over the computation procedure.

Characteristics can be set in different tabs of the Project window tree (either **Preprocessor** or **Postprocessor**), but their feasibility vary for different tabs (see the table below).

Characteristics can be specified in the project tree in two different tabs: in **Preprocessor** and in **Postprocessor**. The **Characteristics**, which has been specified in **Postprocessor**, can be used in this tab only (in **Postprocessor**).

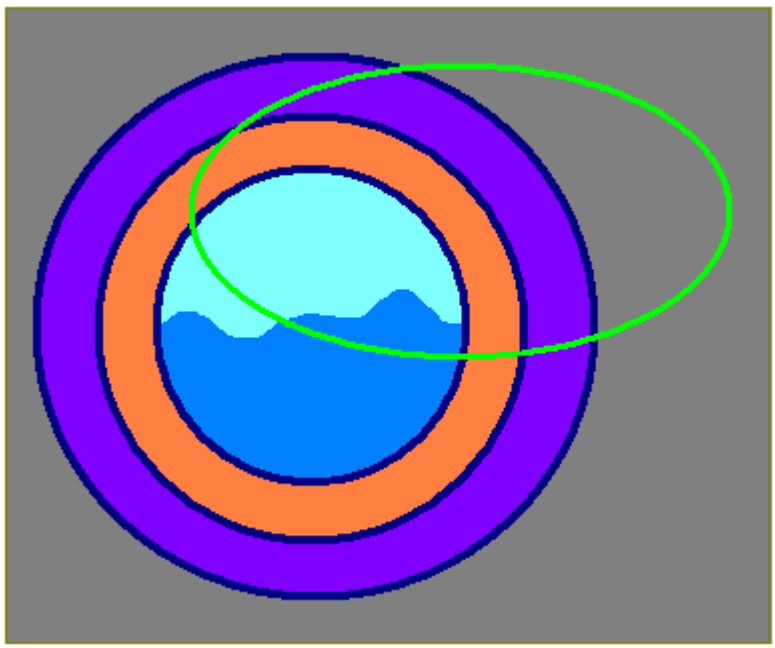
Internal characteristics

The predefined element **Internal characteristics** is a special case of the **Characteristics** element with the following particular properties:

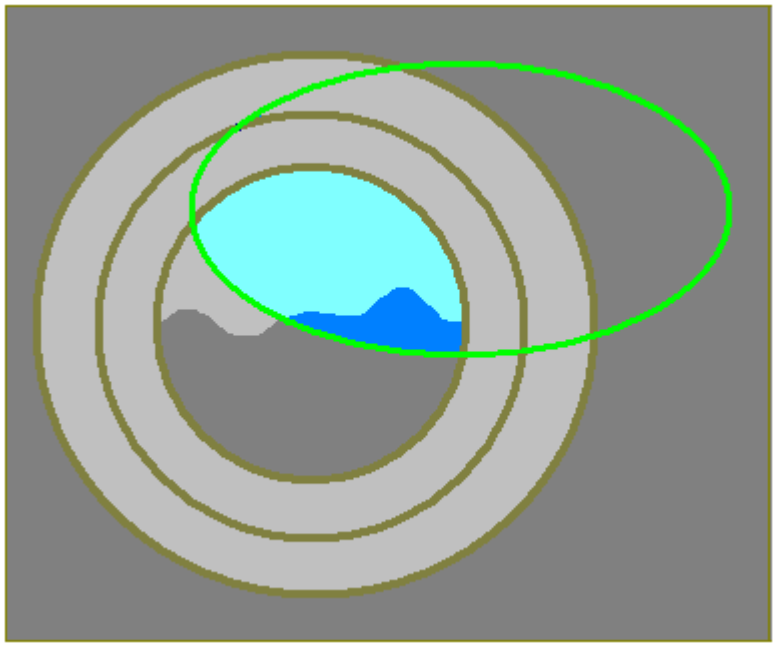
- The **Internal characteristics** element is linked to **Computational space** and it is displayed both in folder **Characteristics** and in folder **Objects > Computational space**.
- Integral variables, access to which and control of which are linked to the **Internal characteristics** element, are preset in the software and are not editable; these are the current time, the time step number, values of the time step and the explicit time step, the reference temperature, the reference pressure, components of the **Gravity vector**, **The phase transfer time**, etc. See details in subsections *Element «Internal characteristics»* and *Information in the Info window for the "Internal characteristics" element*.

Domain of integration (for calculation Characteristics) depending on the Variable's category

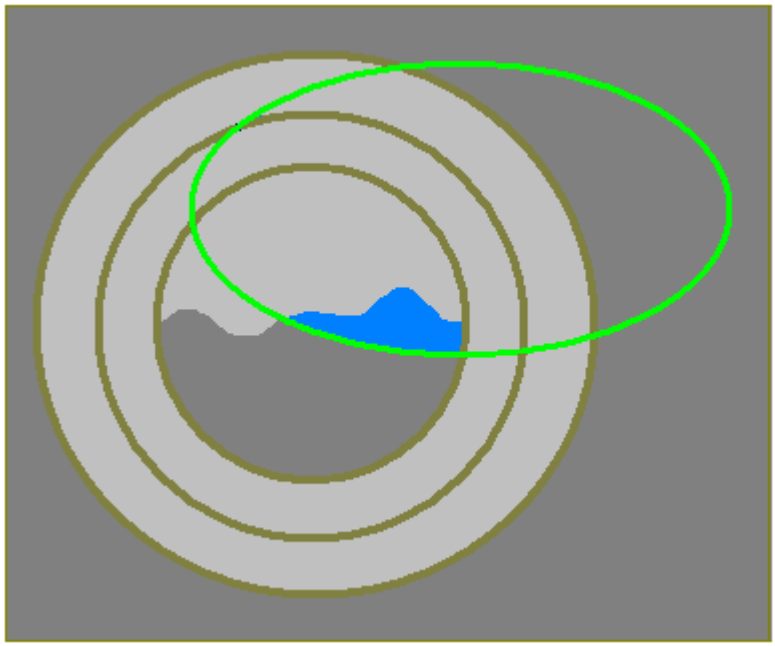
Depending on the **Variable**, based on which the **Characteristics** are calculated, the domain of integration is different, see illustrations below.



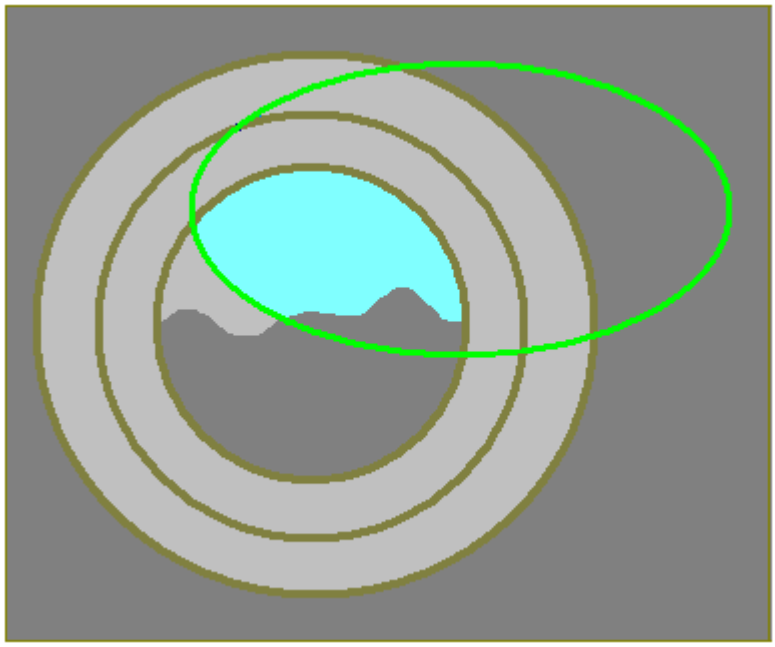
Example of specifying **Characteristics** in the volume of an **Ellipsoid** (displayed by the green contour), which outlines a part of a flow of liquid (**Phase #0**) and gas (**Phase #1**) within a thick-walled tube (**Phase #2**), submerged into other liquid (**Phase #3**). The non-computational area is shown as gray. The part of the volume of the **Ellipsoid**, over which the integrating will be done, depends on the selection of the **Variable**. The selected **Variable** can be specified in a some specific **Phase** or it can be independent on any **Phase** at all. See examples in illustrations below.



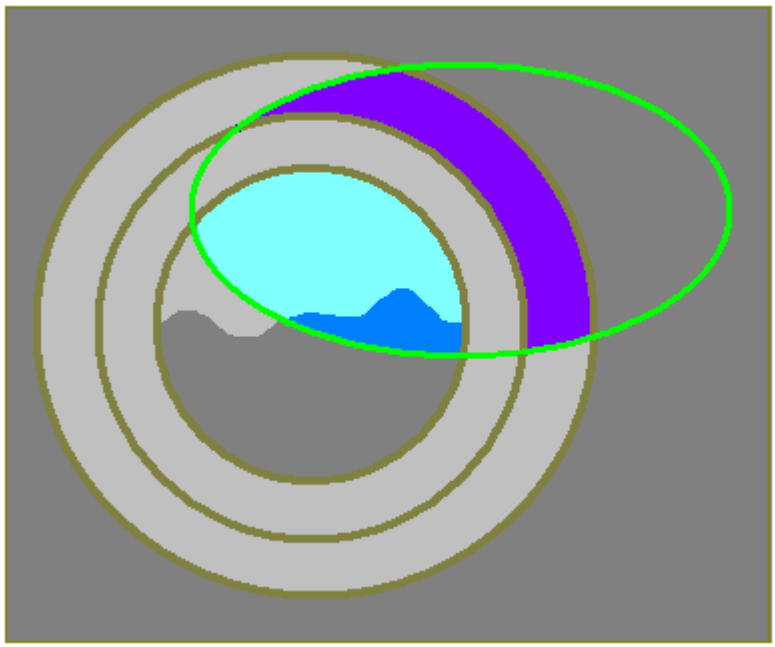
The domain of integration, when **Characteristics** are calculated for the variable **VOF** of some **Phase** in the tube (either **VOF** of **Phase #0** or **VOF** of **Phase #1**), is shown as both blue and light blue colors.
The volume of the whole domain (blue + light blue) is displayed in the Info window for **Characteristics** that has been built either for **VOF** of **Phase #0** or **VOF** of **Phase #1**.



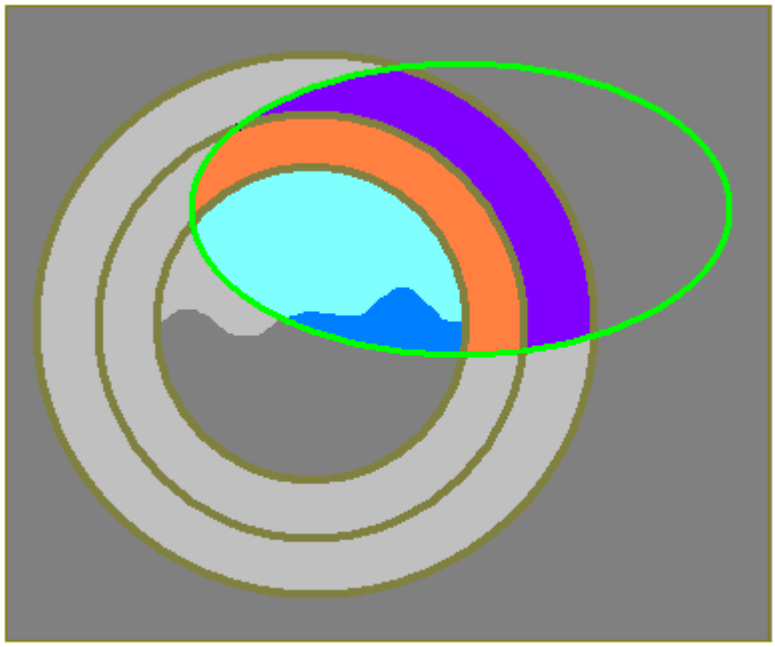
The domain of integration, when **Characteristics** are calculated for some phase-dependent variable (except **VOF**), which is specified in **Phase #0**. It is shown as blue.



The domain of integration, when **Characteristics** are calculated for some phase-dependent variable (except **VOF**), which is specified in **Phase #1**. It is shown as light blue.



The domain of integration, when **Characteristics** are calculated for the non-phase variable Velocity, which is specified in **Phases #0**, **#1**, and **#3**
(the variable **Velocity** is not specified in **Phase #2**, because **Motion** is not specified there). It is shown as colored areas.



The domain of integration, when **Characteristics** are calculated for the non-phase variable **Temperature**, which is specified in all **Phases**.
It is shown as colored areas. The same domain of integration will be used for calculation of **Characteristics** by **User variables**.

See also: Folder «Characteristics»

6.4 Reference parameters, absolute and relative values

In many cases, changes of pressure and temperature in the flow field from hydrodynamic causes are significantly less than the absolute pressure and temperature values. So, it makes sense to present the absolute values of pressure and temperature as sums of reference and relative values to improve computing precision:

$$P_{abs} = P_{ref} + P$$

$$T_{abs} = T_{ref} + T$$

The state equation includes absolute (thermodynamic) values of pressure P_{abs} and temperature T_{abs}.

The movement and heat transfer equations includes pressure with the hydrostatic component P_{+hs1} and relative temperature T.

The reference values of pressure P_{ref} and temperature T_{ref} are specified in the **Pre-Postprocessor**'s project tree in the **Properties** window of the **General Settings** element in the following settings:

- **Reference values > Temperature** (its default value is **273** [K])
- **Reference values > Pressure** (its default value is **101000** [Pa])

Usually, the relative values of **Temperature** and **Pressure** are specified and displayed in the program.

Absolute values of **Temperature** and **Pressure** are used:

- when specifying properties of a **Substance** (also when specifying by a formula or a table using **Formula editor** or **Table editor**; in other cases relative **Temperature** and **Pressure** are used in formulae and tables).
- In **Limiters for calculation**

Note: Make sure not to confuse *absolute static* pressure P_{abs} and temperature T_{abs} with *absolute total* pressure $P_{tot,abs}$ and temperature $T_{tot,abs}$ (deceleration pressure and deceleration temperature of the flow). See Theory > Basic notations.

6.5 Hydrostatic component of pressure

For convenience, the boundary conditions and initial conditions for the problems with pressure changing along the **Gravity vector** (in problems with the hydrostatic) the relative pressure $P_{+\mathrm{hst}}$ is presented as a sum of the hydrostatic pressure component P_{hst} and pressure P without the hydrostatic component:

$$P_{+\mathrm{hst}} = P + P_{\mathrm{hst}}$$

$$P_{\mathrm{hst}} = g \int_0^{h_g} \rho(h)\, dh$$

where:

h_{g} is the coordinate, directed along the **Gravity vector** $\boldsymbol{g}$

h_{g}=0 is the level, on which P_{hst} is assumed to be zero.

Then the contribution of the hydrostatic component of pressure is described explicitly as an individual term of the equation of motion ($-\rho\vec{g}h_g$), and the relative pressure without hydrostatic (P) is included into the gradient term of the equations of motion ($-\nabla P$).

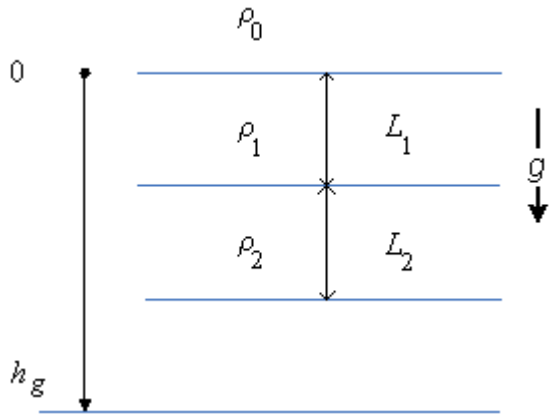
In *FlowVision* the hydrostatic pressure is approximated by a piecewise linear function

$$P_{\mathrm{hst}} = \begin{cases} \rho_0 g h_g, & h_g \le 0; \\ \rho_1 g h_g, & 0 \le h_g < L_1; \\ \rho_1 g L_1 + \rho_2 g (h_g - L_1), & L_1 \le h_g < L_1 + L_2; \\ \cdots & \\ \rho_1 g L_1 + \cdots + \rho_N g (h_g - L_{N-1}), & L_1 + \cdots + L_{N-1} \le h_g, \end{cases}$$

which describes a set of layers with thicknesses L_{i} with constant densities ρ_{i}, where i=0,1,...,N.

Here ρ_0 is the density of the layer above the zero level of the hydrostatic pressure, the layer's thickness is measured in the direction of the **Gravity vector g**, the N-th layer extends from the layer L_{N-1} to infinity. If the layers L_i, where i≥1, are not set (which means that no one layer is specified), it is assumed that the density of the heavy fluid, which determines the hydrostatic pressure, is constant and is equal to ρ_0.

Results of calculations displays the relative pressure P and the pressure $P_{+\mathrm{hst}}$, which includes the hydrostatic component of pressure.



Scheme of layers of a heavy fluid used for approximation of the hydrostatic component of pressure

If hydrostatics is not specified in the project (if $\mathbf{g}$ or ρ_i are zero), then $P_{+\mathrm{hst}}=P$.

Please note that in problems, where hydrostatics is specified, when integral characteristics are calculated and quantitative analysis of values of absolute pressure is done, it is necessary to use the physical variable P_{hst}, as it corresponds to the actual pressure in the point, whilst P is an artificial value.

Parameters of hydrostatic pressure are specified in the project tree in the properties of the element **General settings**.

6.6 Structure of the FlowVision's project

A *project* is a set of:
- files, which contain source data for the computation
- *and* files, which contain the obtained results of the computation

Any *FlowVision*'s project contains several files.

FlowVision has modular structure, so the project files are stored in two separate directories, which are referred as:
- *client part of the project* is intended for working with the project in **Pre-Postprocessor**. Its location is specified by the user.
- *server part of the project* is intended for computations, which are done by the **Solver**. Its location is specified by the registration data (profile) of the **Solver-Agent**'s user.

The directories, which stores the client and the server parts of the project must be different and must not be located one within another (even they must not be nested one within another in several levels separated by other levels of directories).

Source data for the project are formed in **Pre-Postprocessor** and stored in the directory of the **Client part** of the project.

Before starting the computation on **Solver**, the project has to be copied into the **Server directory**. This is done automatically at the connection of **Pre-Postprocessor** to **Solver**. If the connection of **Pre-Postprocessor** to **Solver** is impossible (for example, when working with a remote cluster), you can manually copy the files of the **Client part** into the **Server directory**.

Using the source data, which were copied into the **Server part** of the project, the module **Solver** performs the computations and results of the computations are saved into new files in the same location (in the **Server part** of the project).

Splitting a project source data into two parts and presence the source data in both **Client** and **Server** parts of the project, enables the user:
- to visualize current results of computation during the computation without waiting until the computation finishes
- to work independently with both parts of the project: while **Solver** is doing the computation, the user can change parameters of the visualization of the computation's results

At any attempts of saving or opening a project, **Pre-Postprocessor** examines correctness of the project and, in the case of incorrectness, displays an appropriate message.

See also: section Connecting Pre-Postprocessor to Solver. Work with client and server parts of the project.

The client part of the project

The **Client part** contains only *source data of the project*, consisting of:
- the main geometry model and geometry models of **Imported objects**
- source data for the computation (initial and boundary conditions, computational grid, etc.)
- computation control parameters
- scene (parameters of visualization layers)
- status and history of residuals

The **Client part** is loaded into **Pre-Postprocessor** by menu command **File > Open**. The catalog with the **Client part**'s files can be located anywhere as the user wishes.

Client part's files are saved when using the following commands of the **Pre-Postprocessor**'s main menu:
- **File > Save**
- **File > Save copy**
- **File > Save as**

The **Client directory** *must be open for reading and writing* to the user, as which **Pre-Postprocessor** runs.

In the modern versions of *Windows* the directory **`Program Files`** is protected from writing by non-privileged users. So the normal work with examples from the *Tutorial: Examples of typical tasks* is impossible if you open the examples immediately from their subdirectory in **`Program Files`**. Before the work with the *Tutorial*, your have to copy projects from it into a directory, which is open for writing.

The server part of the project

The **Server part** contains the project's *source data* (a copy of the **Client part**) and computation results with log files. The **Server part** is loaded to **Solver** by a corresponding command from **Pre-Postprocessor**, or **Terminal** or from command line.

The location of the server directory is specified during user's registration on **Solver-Agent**. Each user can have his/her own server directory. This allows different users, working with the same computational server, to separate their work spaces.

Files of the **Server part** generally are created automatically at the first load of the project on **Solver**, and also they can be copied manually from another directory.

Opening the server part of the project using **Pre-Postprocessor** is forbidden!

Files of the **Server part** are saved:

- by **Pre-Postprocessor**:
  - manually (by the menu command **File > Remote save**)
  - *or* automatically during the project's computation (according to settings specified in the properties of the **Data autosave** element)
- by **Terminal** (by the command **Projects > Save project data**)

During the calculation goes, not only new file are created but also the source data can be changed. In this case, the **Client part** of the project differs from the **Server part** and, when **Pre-Postprocessor** connects to **Solver**, synchronization between parts of the project will require.

6.6.1 Project files

Source data of the project include:

- geometry model of the computational domain
- boundary and initial conditions, the initial computational grid, grid adaptation options
- control parameters for the computation
- parameters for visualization of the computation's results

Results of project's computation include:

- fields of the calculated variables
- log file of the computation
- files of error messages
- file with statuses of the computation on each time step
- file with the history of changes of the initial data
- files with results of the computation at previous time moments:
  - when automatic saving of the computation's results is enabled, then these files contain fields of the calculated variables saved at certain time steps
  - when automatic saving of the computation's results is disabled, they are two full sets for the project's files, saved at certain time steps
- files of the visualized values saved at certain time steps, if automatic saving of the visualization data is enabled

Purpose of project files			
Extension	**Format**[*)]	**Contents of the file**	**Notes**
`fvproj`	XML	The main file of the project (contains project ID, versions of the project, parameters of scene and computation, information about the number and numbering of unsteady steps of the record). [1)]	Files are created and edited by means of *FlowVision* in the client part of the project, and then they are copied to the server part of the project.
`fvgeom`	binary	The geometry model of the computational domain (the basic geometry)	
`fvgobj`	binary	Geometry models of imported objects	
`fvbcs`	binary	Mapping of the boundary conditions on groups of facets	
`fvinp`	XML	The input data for the computation	
`fvgprep`	XML	Description of objects in the **Geometry** tab	

Purpose of project files			
Extension	**Format**[*)]	**Contents of the file**	**Notes**
fvgprep_b	binary	Geometry data of objects in the Geometry tab	
fvctrl	XML	Parameters for control of the computation and for stopping criteria	
fvview	XML	Data for visualization a **Layer** in **Pre-Postprocessor** (scene in **Pre-Postprocessor**).	
fvresd	binary	History of residuals and stop criteria. This file contains: • history of residuals obtained in iteration processes at each time step • user Stop criteria, which can be displayed as plots in the Monitor window.	Files are created automatically by **Solver** in the server part of the project, and then they are copied into the client part of the project.
fvstat	XML	The status file, which stores: • information about the status of the computation, which is displayed in **Pre-Postprocessor** in the Monitor window • the current position of **Moving bodies**	
fvtmps	XML	**fvtmps** file is similar in structure and content to **fvstat** file, but contains the *current status* of the computation.	The file is created automatically by **Solver** in the server part of the project during the project's computation (it is not copied automatically into the client part of the project).
fvmind	binary	Index of the grid for projects [2)]	These files are created automatically by **Solver** in the server part of the project.
fvgrid	binary	Computational grid [2)]	
fvdata	binary	Data for projects created in version *FlowVision 3.08.00* or higher (with a higher version number) [2)]	
err	text file	Error files, which contain lists of **Solver**'s error messages. For each process, a separate error file is created and the process number (along with the project's name) is displayed in the file name.	
backup	binary or XML	A file containing information from **fvmesh**, **fvmind**, **fvgrid**, **fvdata**, **fvgobj**, **fvstat** the files of the previous saves.	
glo	text table	A file with data of characteristics and visualization layers, and the file for export loads. When a project is restarted from zero (if the **Continue calculation** checkbox is unselected in the Starting solve dialog box), the existing **glo** files are removed.	These files are created automatically by **Solver** in the server part of the project. The files can be copied into the client part of the project.
fvvis	binary	Data for visualization of characteristics and layers, which were saved during the computation (if the automatic save mode for the visualization data is enabled).	

Purpose of project files			
Extension	**Format**[*)]	**Contents of the file**	**Notes**
		These data can be viewed in **Pre-Postprocessor** with no connection to **Solver**.[3)] When a project is restarted from zero (if the **Continue calculation** checkbox is unselected in the **Starting solve** dialog box), the existing `fvvis` files are removed.	
`log`	text file	An information file (log file) that contains input data of the project and message about operation of **Solver**.	
`fvcam`	XML	Scene in **Viewer**	The files are automatically created by **Solver** in the directory specified by the user.
`fvdtbl`	XML	Exposed parameters (external parameters). It is assumed that this file is generated by some external program. Data from this file allow replacing the values of parameters included in the **Exposed parameters window**.	See sections: • External input parameters • Optimization
`fvrtbl`	XML	The file of exported results. In this file, the values of parameters, which are included in the **Exposed parameters window**, are recorded at each step of the computation. It is assumed that this file can be used by some external program that runs in conjunction with *FlowVision*.	See sections: • Exported results • Optimization
`fvpost`	XML	Settings of the visualization (zooming the visible area, settings for the background, settings of light sources).	
`sta`	text file	Information about time of calculation of each iteration, the total time of computation since the start of the computation and about the use of random access memory of processors on each iteration. See details in section Format of sta files.	The file is created automatically by **Solver** in the server part of the project.
`fvibin`	binary	This file is used to restore the geometry when combining subregions that were separated by a sliding surface and for forming a mapping surface (mapper).	The file is automatically created by **Pre-Postprocessor** in the client part of the project and automatically copied to the server part.

Notes:

*) Binary files cannot be opened using text editors.

1) The project identifier is assigned at creating a project and saving a copy of the project (from **Pre-Postprocessor**). Versions in the project's file changes:

- *client*, when saving a project:
  - if changes have occurred in the input data, then version of the input data changes
  - if changes have occurred in the scene, then version of the scene changes

- *server:*
  - when saving the project, being connected to **Solver**
  - when making a non-steady recording

[2] Since version *3.8.00*, information about the computational grid is stored in the following files: **fvmind**, **fvgrid**, **fvdata**. The file **fvmind** contains indexing information about how the computational grid is allocated among the processors. The file **fvgrid** contains the computational grid. The file **fvdata** contains data. In multiprocessor mode the grid for each processor is stored in a separate file **fvgrid**, and the relevant data are stored in a separate file **fvdata**. If the grid does not change, the file **fvgrid** is not overwritten. The frequency of data saving is specified in the **Project** window in the **Solver** tab in the parameters of the element **Data autosave**. Prior the version *3.8.00* **fvmesh** files were used instead of files **fvgrid**.

[3] Frequency of saving the visualization information is specified in the **Project** window in the **Solver** tab in the parameters of the element **Layers autosave**.

Use of project files on the client and server parts of the project		
File extension	**The client part**	**The server part**
fvproj	+	+
fvgeom	+	+
fvgobj	+	+
fvbcs	+	+
fvinp	+	+
fvctrl	+	+
fvview	+	+
fvresd	+	+
fvstat	+	+
fvmind	-	+
fvgrid	-	+
fvdata	-	+
err	-	+
backup	-	+
glo	-	+
fvvis	-	+
log	-	+
fvcam	depends on the location defined at saving	depends on the location defined at saving
fvdtbl	+	+
fvrtbl	+	+
sta	-	+
fvibin	+	+

6.6.2 Format of sta files

sta-file of the project contains information about time of calculation of each iteration, the total time of computation since the start of the computation and about the use of random access memory of processors on each iteration. This file is a text file and stores data in groups similar to the data group in the example below:

```
============================== Iter number 43 ==================================
CALCULATION TIME STATISTICS
Iter calculation time:    67.9012
Total calculation time:   3252.05

SOLVING STATISTICS
Tcur:    0.0100753
Texpl:   7.48709e-05
Tstep:   9.33215e-05
Nstep:   43

SYSTEM STATISTICS
          PhM_total   PhM_free   Proc_mem   Peak_mem   CPU_time   Wait_CPU Wait_Wall
Proc0    :  16056.09   12092.91    2624.93    3644.18     177.91      37.50      37.52
Proc1    :  16056.09   10057.96    2066.38    3798.94     232.95      16.35      16.44
Proc2    :  16056.09   10057.96    2899.29    3193.89     184.21      21.78      22.09
Proc3    :  16056.09   10320.22    2679.92    3800.99     154.74      38.13      38.38

Total memory requirement estimate : 10270.52 14438.00
```

See descriptions of the **sta** file's data in the table below:

Parameters, groups of parameters	Descriptions, dimensions
Iter number	Iteration number
CALCULATION TIME STATISTICS	Statistics of the physical duration of the computation
Iter calculation time	Physical duration (wall time) of computation of the iteration, [s]
Total calculation time	Total duration of the computation time since the last start of **Solver** for computation, [s]
SOLVING STATISTICS	Statistics on the operation of **Solver**
Tcur	Current time of the project, [s]
Texpl	Explicit convective time step, [s]
Tstep	Duration of the time step, [s]
Nstep	Number of of the current time step
SYSTEM STATISTICS	System statistics
Proc0, ..., **ProcN**	Lines with individual statistics by used processors
PhM_total	Amount of physical memory on the computational node (computer), on which the **Solver**'s copy runs, [Mb]
PhM_free	Amount of free physical memory on the computational node, on which the **Solver**'s copy runs, [Mb]
Proc_mem	Memory consumption by the **Solver**'s copy on this node at the end of the time step, [Mb]
Peak_mem	Maximal memory consumption by the **Solver**'s copy on this node since the **Solver**'s start, [Mb]
CPU_time	The total processor time of all threads during the period of computation of one time step, [s]. As this parameter sums up the time, which has been spent by all nodes (threads), **CPU_time** might be greater then the physical (wall) time.

Parameters, groups of parameters	Descriptions, dimensions
Wait_CPU	Processor time of waiting of all threads during computation of one time step, [s]. This is the time, which is spent for execution the *MPI* operations (that are operations, which provide data transfer between processors). The more is the disbalance between processors, the more is the value of **Wait_CPU**.
Wait_Wall	Physical time of waiting of a processor, [s]. The more is the fraction of the waiting time in the physical duration of computation of the whole iteration (**Iter calculation time**), the more is the fraction of exchange operations in the calculation and the less is the efficiency of the calculations. Some possible reasons of long time of waiting are: • non-optimal mode of run (too many processors for this number of cells) • the operating memory is too slow • incorrect configuration of the software, which causes use of another's memory If in one iteration two processors have substantially different values of **Wait_Wall**, this means large disbalance of calculations over processors. To reduce the large disbalance, you should try to specify an optimal run mode, enable the automatic balancing and, might be, change the generation of the computational grid (for example, reduce the level of the local adaptation).
Total memory requirement estimate	Total values of the values in the columns **Proc_mem** and **Peak_mem**, [Mb]

Notes:

CPU Time is the time, which is spent totally by all processor's nodes directly for the calculations. If the computational task suspends waiting new data, the *CPU Time* counter pauses.

Wall time is the real (physical) time, which is spent by a processor for carrying out the operation, including all down times during the waiting. This is the time measured by the wall clock from beginning till ending of carrying out the task.

Specifics of measuring the time-related data stored in sta-files

Measuring of the parameters such as **Iter calculation time**, **Total calculation time**, **CPU_time**, **Wait_CPU**, **Wait_Wall** is done only during making calculations exclusively during carrying out the calculation of one computational step itself. So, the program does not measures the time, which is spent on:

- data exchange with other software (for example, when making *FSI* simulations)
- saving the results (computational data, **glo**-files, **fvvis**-files) on the disk
- calculating the data, which are necessary to build visualization layers displayed at connections of **Pre-Postprocessor** or **Viewer**, and stored at saving of **fvvis**-files

All events, which are registered in the **log**-file outside the block

```
===== New Time Step=====, Iter number =  …
End:
===== Time Step=====, Iter number = …
```

are not taken into account in the **sta**-file.

See also:

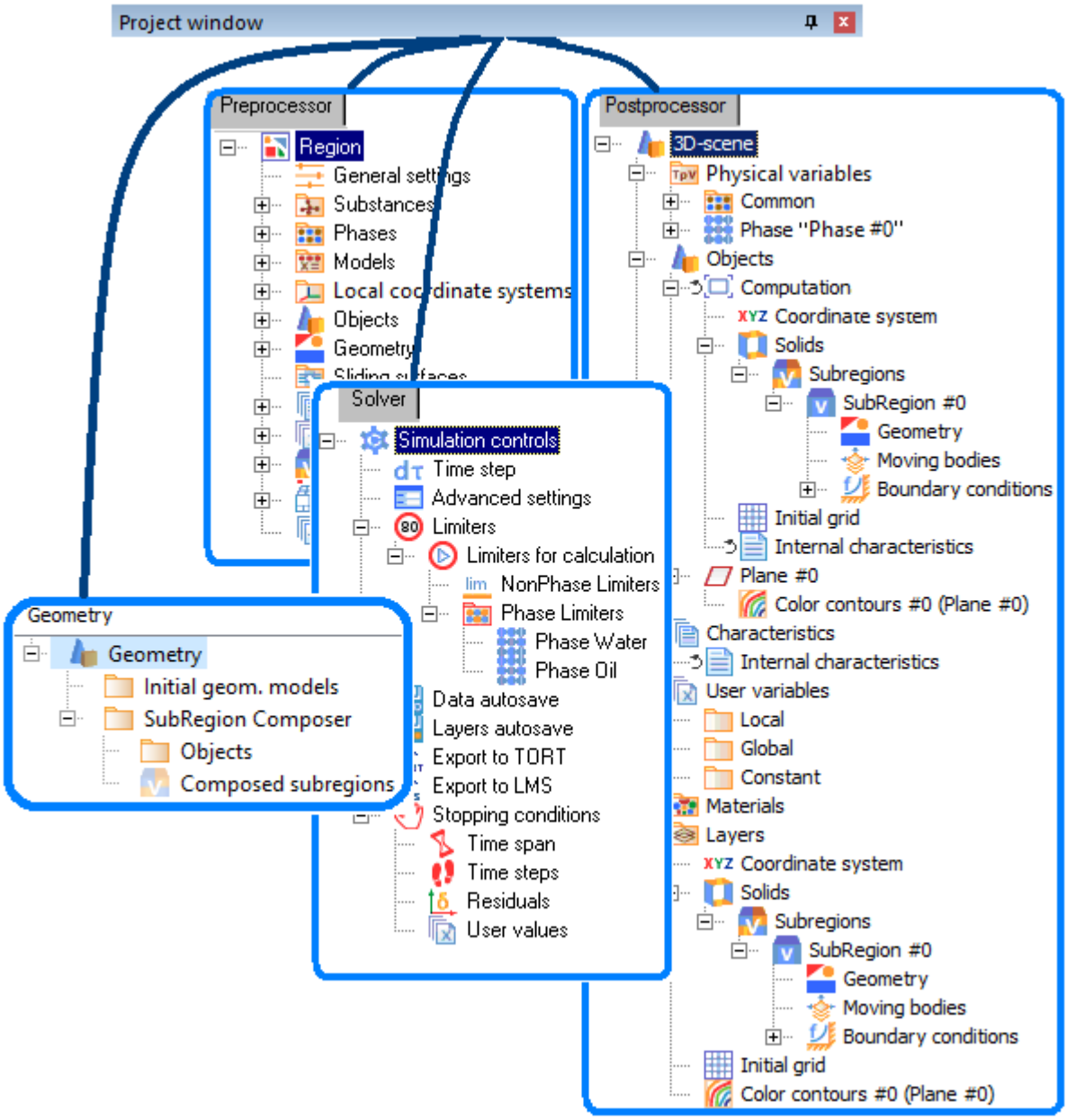
- Section Parallel computations in FlowVision
- Subsection *"Configuring the Dynamic Balancing"* in the section Element «Advanced settings» (advanced settings of Solver)
- Section Element «Data autosave»
- Description of setting for saving the data from **Layers** into **glo**-files, which are defined by the **Save to file** parameters (see section General properties of Layers)
- Description of setting for saving the data from **Characteristics** into **glo**-files, which are defined by the **Save to file** parameters (see section The "Characteristics" folder)
- Section Time step

- Sections about interaction with third-party's software (External input parameters, Exported results, Data export after computation, Data export for visualization in EnSight, Use of connectors, Optimization, Neutron transfer (TORT), Acoustics (LMS))
- Section Layers and displaying the computation's results

6.7 General sequence of project creation, project tree

A project is presented in **Pre-Postprocessor** as a set of nodes combined in a tree-like structure, the *project tree*. Nodes of the project tree can be either folders (containing other nodes) or nodes, which don't contain child elements.

The project tree consists of four main branches, which are displayed in tabs **Geometry**, **Preprocessor**, **Solver**, **Postprocessor**:



See details in the section Window «Project».

Sequence of project's preparing

When you click the **Create a project** button, you can select either creation a project without any geometry model or start creation a project from import of a geometric object that encloses the computational domain (in the latter case the operating system's dialog box for access to files will open).

When a new project is created, you have to modify the project tree according to the selected setting of the problem.

Modification of the project tree generally consists of the following actions:

- creation new elements
- editing parameters of elements

We recommend to go through the project tree downwards in each tab, and from from left to right by tabs. This sequence facilitates forming an integer and correct structure of the project.

The **Geometry** tab might not be used when you load the geometry model from a file.

6.7.1 Recovery of the data from the previous saving, if the program terminates abnormally

Sometimes, because of failures of hardware, file system or third party's software, operation of *FlowVision* might terminate during the data saving.

In such situations, because of the data recording has not been fulfilled, some project files become damaged and it would be not possible to load the project on **Solver**. Along this, information about missing and damaged files is written into the `.err` file in the server part of the project.

It is often possible to restore data from the previous successful saving of the project. The recovery algorithms are different for the cases:

- if the project's data, before the problem arose, was being autosaved with **History**, i.e. if the calculation's results have not been overwritten (**History=Yes** is set in the properties of the **Data autosave** element)
- if the project's data has been autosaved without **History**, i.e. if the calculation's results have been overwritten (**History=No** is set in the properties of the **Data autosave** element)

In these situations, the substantial is only how *the last saving* has been done (if history has been enabled or disabled) regardless of how the saving was done before it.

Recovery of the data, if the last data autosave has been done with History enabled

To recover a project, if the last data autosave has been done with **History** enabled, follow the steps:

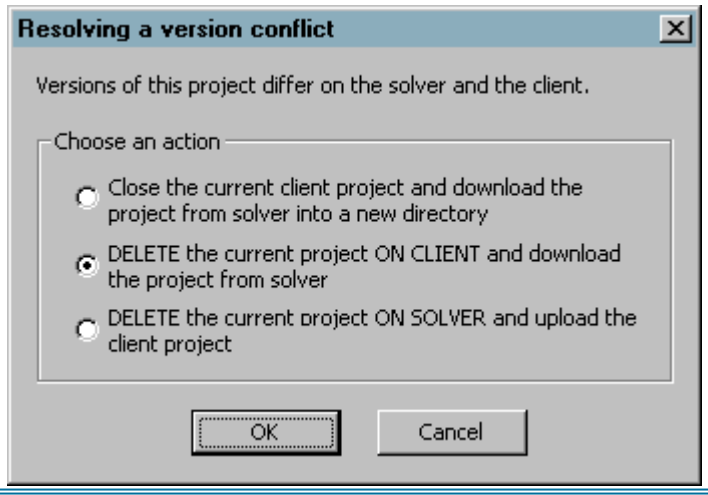
Step	Actions
1	Make backup copies of the server and client parts of the project. Do not change anything in these copies, then work with the server part of the original project. These backup copies will be useful in the case, if you, trying to follow the recovery procedure, make a mistake, which destroy the project's data. ***Note about location of the backup copies:*** Storing projects with same ID is not allowed neither in the client directory no in the server directory, so you must save backup copies in some other directories.
2	Make sure that the file `.fvproj` in the server part of the original project is not empty and contains typical data. If the is empty, try to recover the `.fvproj` file from its copy, which has been made at the previous saving (the file `.fvproj.backup`). To do this, delete the damaged `.fvproj` file, and remove the extension `.backup` from the name of the file `.fvproj.backup`. If the file `.fvproj.backup` also contains incomplete data or is damaged, try to recover using the file `.fvproj` from the client part of the project.

3	In the file `.fvproj` remove *the last* block `<ITEM>`…`<ITEM>`, containing the `<Intermediate>` tag, if any. Such block looks like this:
	`<ITEM>`
	`<Number>16</Number>`
	`<Iteration>7598</Iteration>`
	`<Time>0.20775437958965703</Time>`
	`<Intermediate>1</Intermediate>`
	`</ITEM>`

4	Try to load the project on **Solver**.
5	If the project again can not be loaded on **Solver**, try to remove *another* block `<ITEM>`…`<ITEM>` (might be just this saving is damaged).
6	Repeat step **4**.

Recovery of the project from the previous saving, if the last data autosave has been done with History disabled

To recover a project, if the last data autosave has been done with **History** disabled, follow the steps:

Step	Actions
1	Make backup copies of the server and client parts of the project. Do not change anything in these copies, then work with the server part of the original project. These backup copies will be useful in the case, if you, trying to follow the recovery procedure, make a mistake, which destroy the project's data. ***Note about location of the backup copies:*** Storing projects with same ID is not allowed neither in the client directory no in the server directory, so you must save backup copies in some other directories.
2	Try to recover the previous saving, all files of which are stored in the **`.backup`** files. To do so, you have to replace the project files in the server directory by their copies, which were created on the previous saving (**`backup`** files). Follow the steps: **2a.** Sort the files by date. Find the last saved files with extension **`backup`**: **`name.fvproj.backup`** **`name_xxxxx.fvstat.backup`** **`name_xxxxx.fvmind.backup`** **`name_xxxxx_yyy.fvdata.backup`** **`name_xxxxx.fvgobj.backup`** (can be absent) **`name_xxxxx_yyy.fvgrid.backup`** (can be absent) Here **`xxxxx`** is the number of the last saving. **2b.** Delete the files **`name_xxxxx.fvstat`**, **`name_xxxxx.fvmind`**, **`name_xxxxx_yyy.fvdata`**, **`name_xxxxx.fvgobj`** (if any), **`name_xxxxx_yyy.fvgrid`** (if any) that have number of saving, which is equal to the number **`xxxxx`** in the found **`backup`** files (from the step **2a**). Also delete the file **`name.fvproj`**. **2c.** Remove the extension **`.backup`** from names of the **`backup`** files, which has been found at step **2a**.
3	Try to load the project on **Solver**. When connecting to **Solver**, select in the **Resolving a version conflict** window, which opens, the option "**DELETE the current project ON CLIENT and download the project from solver**" and click **OK**: 

6.8 Geometric elements of the project

Geometric elements of the project include:

- Geometry model of the computational domain
- Geometric objects:
  - standard geometric objects (**Box**, **Cone/cylinder**, **Ellipsoid/sphere**, **Plane**, **Line**)
  - **Imported objects**
  - **Supergroups**
  - **Groups** of facets
  - **Computational space**
  - **Sets of sensors**
- Coordinate systems

FlowVision allows you to import, transform, and create various geometric objects. Size values of geometric objects are specified in meters, [m].

The geometric objects are used for:

- forming the computational domain (geometry boundaries), including those that are used for simulating moving walls and bodies.
- specifying in the space of the computational domain parameters of the computation:
  - **Initial conditions**
  - values of physical variables (**Velocity**, **Pressure**, **Temperature**, etc.)
  - specifying volume forces, heat sources, resistances, etc.
- visualization
- calculation of integral values (**Characteristics**)
- specifying areas where local refining (adaptation) of the computational grid will be applied

See the next sections for details about coordinate systems and geometry elements of the project.

6.8.1 Geometry model of the computational domain: surfaces and subregions

Geometry model of the computational domain is defined geometry boundaries of the space forming an enclosed volume, in which the simulating is done. This is geometry model of the **Region**.

For simulations of internal flow, the geometry model of the computational domain is the volume of the flow channel with «blank flanges» that simulate inlets and outlets.

For simulations of external flow, this is the outer volume (for example, a box), on which inlets and outlets are set, and a body inside that is flown around.

To form a computational domain it is necessary to import its geometry model, which is presented by a faceted surface. Further specifying parameters of simulation for the project can be done only after loading the **Region**'s geometry model.

To form a computational domain, it is necessary to import its geometry model.

Meanwhile, a computational domain can be formed by:

- importing the main geometry of Region:
  - immediately from a file
  - from the **Geometry** folder
  - from standard geometric objects, created in *FlowVision*, and from **Imported objects**
- by surfaces of inserted **Moving bodies**

Geometry model of the **Region** is one or several closed surfaces that form enclosed volumes. These surfaces might be nested inside each other or can be spatially separated.

Intersections or coinciding of **Region**'s surfaces are not allowed. See Requirements to geometry models.

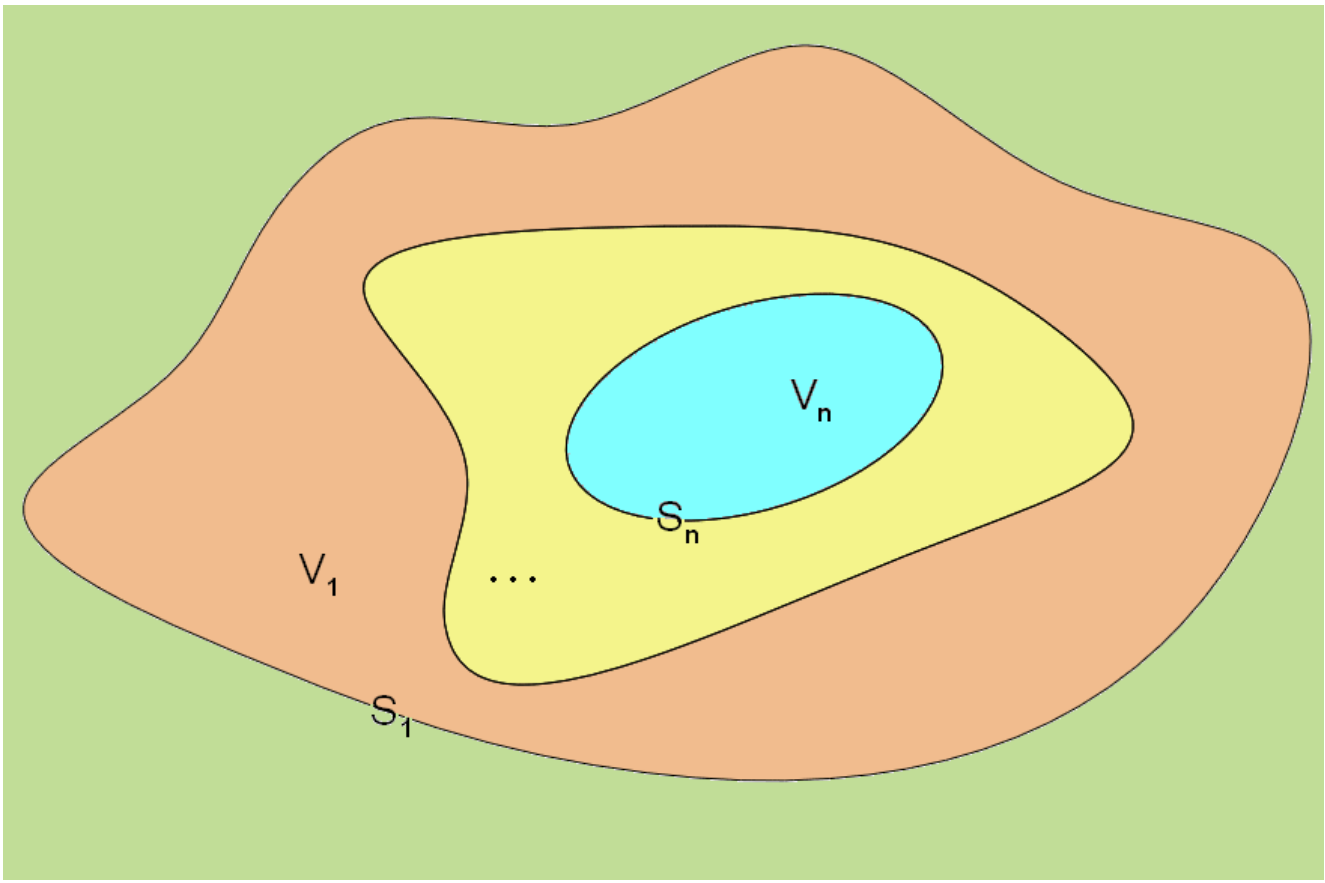
A particular case of the **Region**'s geometry model is assembling with union that appears when intersection of two surfaces causes multiconnection (T-connection).

Surfaces of the **Region**'s geometry model might be intersected by **Moving bodies**.

Each enclosed volume of the the **Region**'s geometry model forms a **Subregion**. Volume of a **Subregion** can be formed by one or several closed surfaces.

Subregions can be mutually linked by a computation of some or other physical variables (conjugation by **Temperature**, **Velocity**, etc.). Calculations in different **Subregions** can go independently; also some

Subregions can be non-computational. In assemblies with multiconnection (T-connection) *all* **Subregions** that are contiguous to the multiconnection must be computational.

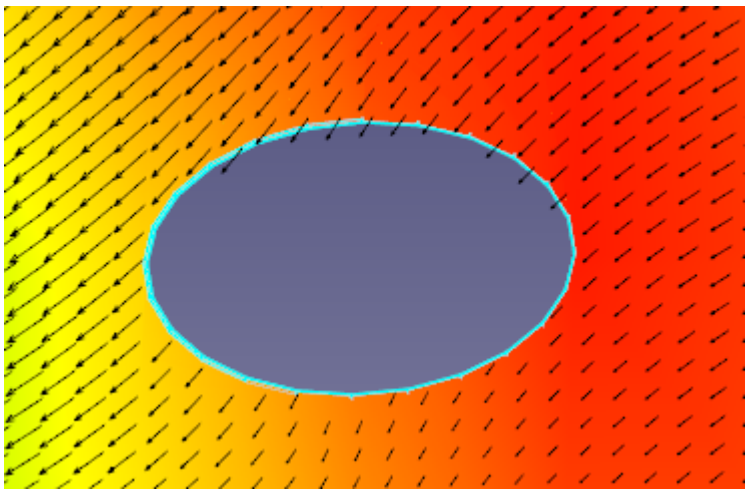


Example: surfaces S_i form volumes of **Subregions** V_i, i=1,...,n

Computation within the volume of a **Subregion** is only performed if a model of physical processes and physical properties of the media are specified in the **Subregion** (you have specified **Substances**, **Phases**, **Model**).

On the boundaries of **Subregions** (both on outer and internal ones) **Boundary conditions** are set that allow you to specify physical parameters outside the computational domain. You don't need to specify anything of that in non-computational domains.

In a non-computational **Subregion**, the program does not calculate flows and/or heat and mass transfers. An example of a non-computational **Subregion**: space within a non-moving and non-heat-conducting body streamed by fluid (see the illustration below).



Example of a non-computational area (within the body that is flown around)

When a surface separates two **Subregions**, **Boundary conditions** are set on both sides of the surface (individually for each of the **Subregions**). When necessary, **Boundary conditions** can be mutually bound to provide, for example, mass or heat transfer between **Subregions**.

When the geometry model of the computational domain is partially formed by surfaces of **Moving bodies**, you have to specify **Boundary conditions** also on **Moving bodies** in appropriate **Subregions** (and **Moving bodies** don't form **Subregions** and no **Models** are specified within them).

See also:

- Import of the geometry model of the computational domain
- Assembling with separation
- Algorithm of assembling with separation
- Algorithm of assembling with union
- Example of assembly with union
- Possible errors when making an assembly
- Adding or removing subregions in geometry models
- Embedding a Moving body's surface into the computational domain
- Operations with the geometry model of the computational domain
- Loading a geometry model of the computational domain into a project

6.8.1.1 Import of the geometry model of the computational domain

FlowVision carries out simulations in limited space only. Boundaries of the computational domain are specified by geometry surfaces. The unit of measurement for geometry models is meter, [m].

FlowVision does not include a built-in CAD system, but it allows an engineer user to import a geometry model from those CAD system, in which the user works.

The import file for *FlowVision* contains the mesh presentation of one or several closed surfaces, see Requirements to contents and formats of geometry files.

Creation and loading a geometry model can be done using one of these methods:

- Creation a project based on a imported geometry model that is imported from a file.
- Creation an empty project and preparing a geometry model in the **Geometry** tab. In this tab it is possible to import a source parametric CAD-geometry and to form a mesh surface for **Region** or **Imported object** from it.
- Creation standard and imported geometric objects and embedding them into the project as the geometry model of the whole **Region** (for an initially empty project) or embedding additional geometry surfaces (for **Subregions** that are added to the existing geometry model of the **Region**).
- Creation an assembly made of several geometry models (batch import of several files).

During loading a geometry model of the computational domain, the following data are imported:

- the coordinate system, in which the element is specified (in *FlowVision* this coordinate system is treated as the absolute coordinate system of the computational domain)
- coordinates of nodes of facets, data about edges and faces of the facets
- when an element in imported in the *VRML* format, the data about colors of facets are transferred, if the facets were colored in the CAD-system. All surfaces with the same color are automatically grouped and assigned to individual boundary conditions. This provides automatic allocation of boundary conditions to surfaces.

During the loading the program splits the facets of the imported model into groups of triangular facets, see Grouping the facets and Regrouping the facets.

After the surface, which is to be included into the geometry model of the computational domain or into the surface of an imported object, is loaded, the quality of imported surface can be checked. This check consists of two steps:

- check the imported surface for self-intersection (with correction the found self-intersections)
- removal of too-small facets

The geometry model can be loaded either from a single file or from several files (the latter case is called *assembly*).

See also:

- requirements to contents and formats of geometry files
- step-by-step instructions in the section Loading a geometry model of computational domain into a project

6.8.1.2 Forming the geometry model of the computational domain

Let's take a closer look at methods of creation (forming) the geometry model of the computational domain.

The geometry model of the computational domain might be formed by:

- importing the main geometry of the **Region** (this is a mandatory step in forming the project's computational domain):
  - from a single file
  - from several files (forming an assembly)
  - from the **Geometry** tab, in which the geometry model of the computational domain was preliminary prepared
  - by creating and embedding standard and imported geometric objects of *FlowVision*
- adding surfaces of the embedded **Moving bodies**, which can either be moving or remain stationary, and which can also be replaced during the computation (see section Embedding a Moving body's surface into the computational domain).

6.8.1.2.1 Importing a geometry model from a file

The geometry models is created in a CAD system. The model presents a volume formed by closed surfaces, between and within which the computational volume locates.

Geometric objects from a CAD model are exported into a file with mesh representation of outer and inner surfaces. Then objects from the file are loaded into *FlowVision* forming the computational volumes (**Subregions**).

A single file can contain several closed surfaces, which form several **Subregions** at loading them into *FlowVision*.

To create a project, it is enough to open a file with a geometry model.

Also it is possible to import surfaces with multiconnection, when multiple surfaces contact each other at an edge (a special case of multiconnection is a T-connection of surfaces). Multiconnection can appear when a project is created with assembling with union and further exporting the **Region**'s geometry model into a file.

See section Requirements to geometry models for information about the supported file formats.

6.8.1.2.2 Preparing a geometry model in the Geometry tab

In *FlowVision* you can preliminary view a geometry model in a file, make changes in the model and even after this use the model to form the computational domain.

To do so, you should create an empty project and then you begin your work from preparing the geometry model in the **Geometry** tab: the model is arranged in the space, it is shifted, transformed, and then a mesh geometry surface is formed.

Thus, in the **Geometry** tab, it is possible to:

- load one or several files with the geometry model
- load a parametric geometry model from the CAD-system (this is possible when you installed the *3DTransVidia* software for *FlowVision* and you have a license for its use)
- create 3D standard geometric objects (**Boxes** **Cones/cylinders**, **Ellipsoids/spheres**)
- form the mesh geometry surface based on the parametric model and standard geometric objects
- group geometry elements in folders of the project tree
- mutually arrange all the geometry elements (move them, scale, delete)
- use selected geometry elements to form the **Region**, i.e. use them as the geometry model of the computational domain

See details about the **Geometry** tab in sections:

- The Project window, tab «Geometry»
- Root folder «Geometry»
- Folder «Initial geom. models»
- Folder «SubRegion Composer»

6.8.1.2.3 Creating assemblies

In *FlowVision* it is possible to create geometry models of complex surfaces, obtaining them from several parts (each of them is specified by a simple surface without multiconnection).

In this case the program applies batch import of several pats (importing each part from its own file) with possible further fixing self-intersections. The files, from which the import is done, are specified in the **Assembly creation** dialog box, see subsection Loading geometry model of multiple files (assembly).

If surfaces of different parts do not contact and have substantial gap between them, then the parts are assembled without fixing self-intersections.

If surfaces of the parts contact each other or gap between them is too narrow, then algorithms for fixing self-intersections are applied:

- *either* assembling with separation (elements will be transformed equidistantly, they will be shifted from each other)
- *or* assembling with union (in this case the surfaces join together and form a contact patch with T-connection of the surfaces)

Creating assemblies (geometry models loaded from multiple files) is described in sections below:

- Assembling with separation
- Algorithm of assembling with separation
- Assembling with union
- Algorithm of assembling with union
- Example of assembly with union
- Possible errors when making an assembly

See also:

- Loading a geometry model of the computational domain into a project (step-by-step instructions)
- Checking a geometry model of computational domain and moving bodies for self-intersections
- Fixing self-intersections of a surface in a geometry model

6.8.1.2.3.1 Assembling with separation

Assembling with separation is applied when each part is saved in its separate file and parts in the CAD assembly have contact between them. As surfaces must not coincide or intersect in *FlowVision*, they can be inserted into a project using the batch import with providing at the same time an *ensured gap between the parts*. We refer this functionality as "*assembling with separation*".

When the program finds a contact between surfaces, it separates these surfaces (makes an equidistant shift i.e. offset of the geometric closed surface of a part to one or another direction) and so a gap appears instead of contact of surfaces.

It is recommended to place the external geometry on the first position in the list of files that will be loaded for making an assembly.

Parameters that are specified for assembling with separation:

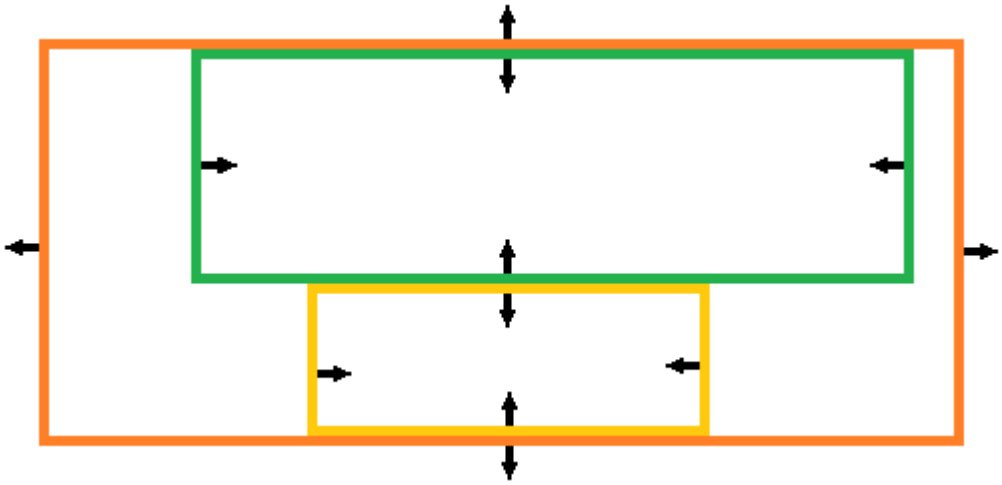
- **Max. iteration number** – the maximum number of iterations in the procedure, which searches intersections of the separated surfaces
- **Surface offset** – the maximum displacement of nodes of facets per one iteration

These parameters influence the maximal number of possible program's attempts to make an equidistant shift of the surface (to create a gap instead of contact of surfaces).

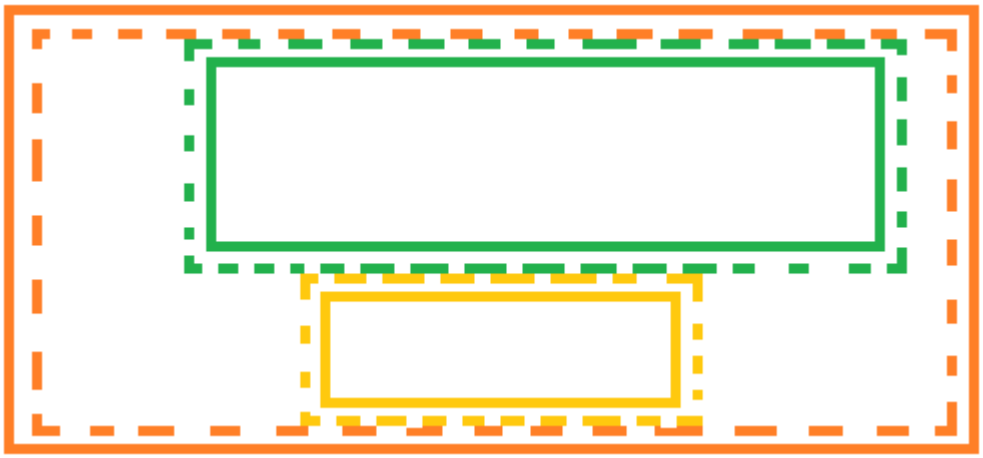
If the total offset, which appears due to the separations, becomes insufficient to separate the parts, the assembly is either not created or created with self-intersections. In this case we recommend to change slightly the parameters of the assembly separation algorithm or apply the equidistant shift manually (immediately in the CAD system or element-by-element in *FlowVision* using the geometry transformation). If even this didn't help, contact our technical support service.

6.8.1.2.3.2 Algorithm of assembling with separation

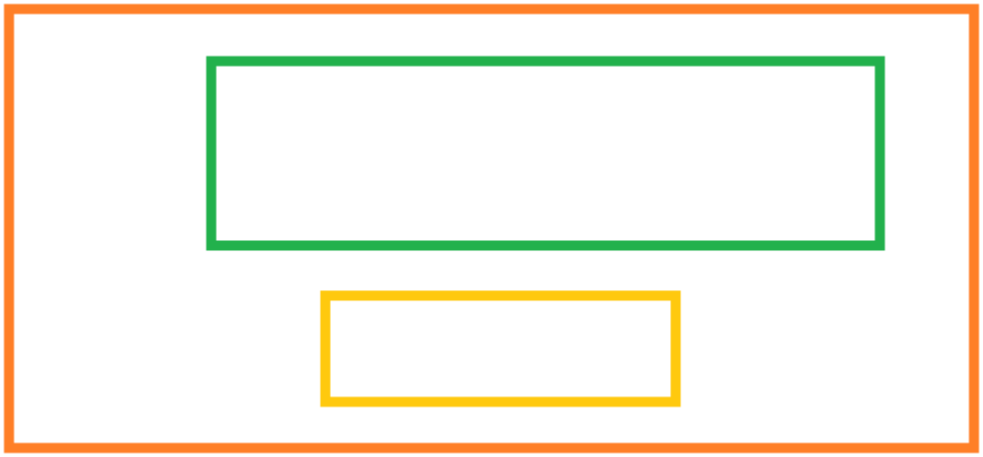
When assembling with separation is used, with surfaces are inverted so that normals of internal surfaces will be directed inside, and normals of the external surface will be directed outside:



Then the program checks if the geometries intersect. If the geometries intersect, then each surface will be shifted in the direction of its normal (an internal shift is done for internal surfaces, and an external shift for external surfaces). Then each surface is checked for self-intersection, which might appear because of the shift. If self-intersections exist, the program makes another attempt to separate the surfaces (the maximal number of such attempts is specified by the parameter **Max. iteration number** in the Part intersection fix).



Then the whole geometry is checked for self-intersection. If self-intersections exist, they are displayed in **Postprocessor**. Self-intersections can appear because of impossibility in a particular problem's formulation to make a shift to the specified distance and prevent self-intersections of the surface.



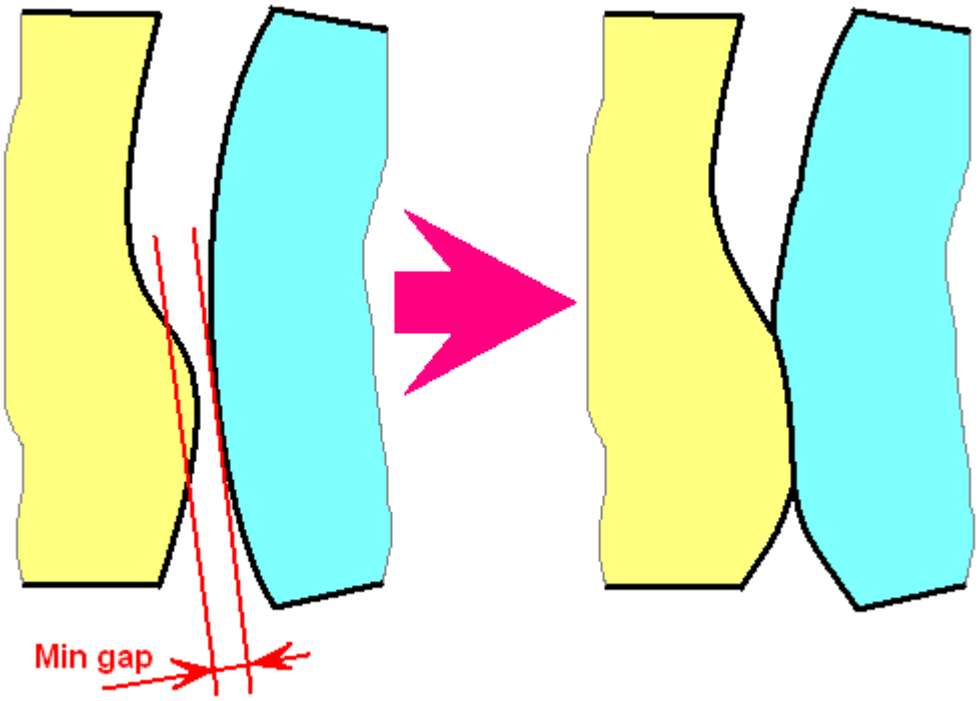
6.8.1.2.3.3 Assembling with union

Assembling with union is applied when each part is saved in its separate file and parts in the CAD assembly have contact or overlapping. As surfaces must not coincide or intersect in *FlowVision*, they can be inserted into a project using the batch import with providing at the same time a *contacts between the parts* with forming the multiconnection. We refer this functionality as "*assembling with union*".

When the program finds closely-spaced surfaces, it eliminates the gaps and forms the contact patches. Assembling with union is possible in the mode when multiconnection is allowed (i.e. when **Geometry import > Enable multiconnection = Yes** is set in the basic settings of Pre-Postprocessor).

Parameters that are specified for assembling with union:

- **Min gap**: at places where the distance between the surfaces is less then **Min gap**, merging of the surfaces occurs (see the illustration below)
- **Remove internals**: the program automatically removes the internal volumes, which appeared after merging the surfaces



Assembly with union of parts causes multiconnection.
Merging of the surfaces occurs there, where the distance between them is less than the **Min gap** value.

Coupling the parts is provided due to equidistant shift of geometric surfaces of the parts. The first part in the list of parts that form the assembly don't change its geometry and the shift will be applied only to other pats in the list.

It is recommended to place the external geometry or the largest part on the first position in the list of files that will be loaded for making an assembly.

Parts can change to a certain extent their geometry during the assembling, so you have to examine the results after the assembling is done.

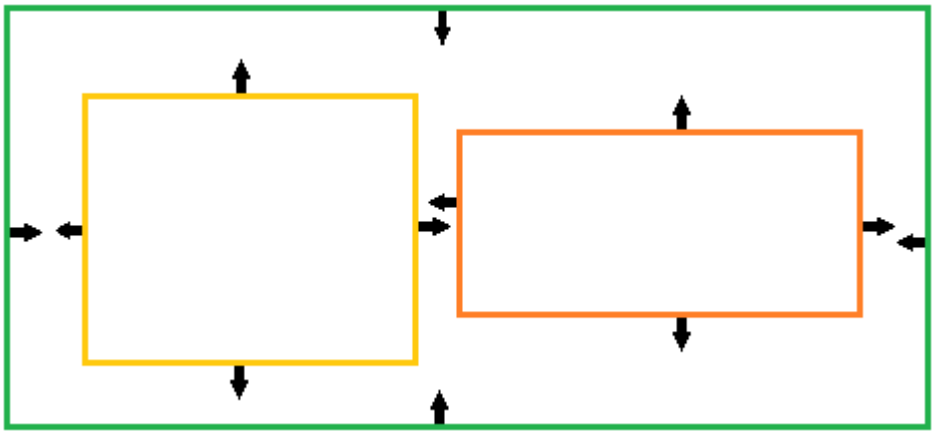
When the gap between parts is greater the the specified minimal value, the program will do nothing with the geometry, the parts will not unite and the gap between the parts will remain as before.

When you set **Remove internals = No**, the shifts can cause appearing of small volumes, which, due to the assembling, would form small **Subregions** in the project. As these **Subregions** are unnecessary for the computation, you can delete them manually.

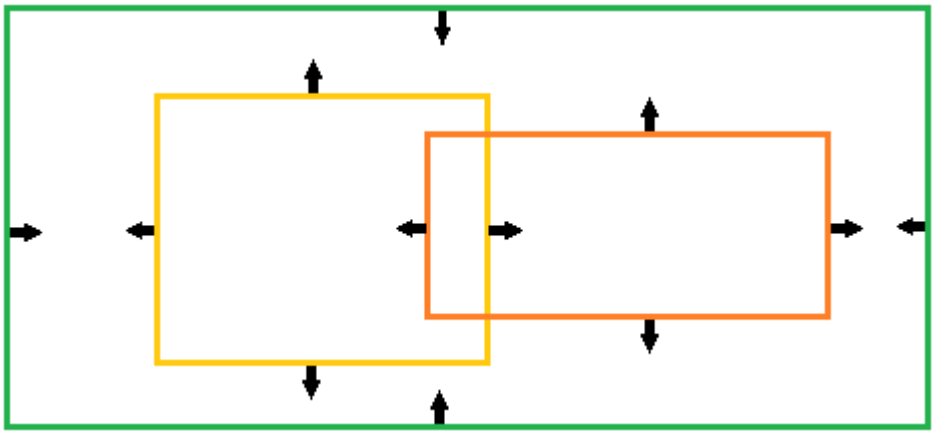
6.8.1.2.3.4 Algorithm of assembling with union

This type of assembly allows you to create multiconnected surfaces. This is convenient for simulations with conjugate heat transfer. This algorithm makes an equidistant shift of surfaces until their intersection and then it tries to seam the original surfaces and form multiconnected surfaces.

When this method of assembling is used, normals of all internal surfaces are directed outwards and normal of the external surface is directed inwards. If required, the algorithm might automatically toggle the orientation of surfaces.

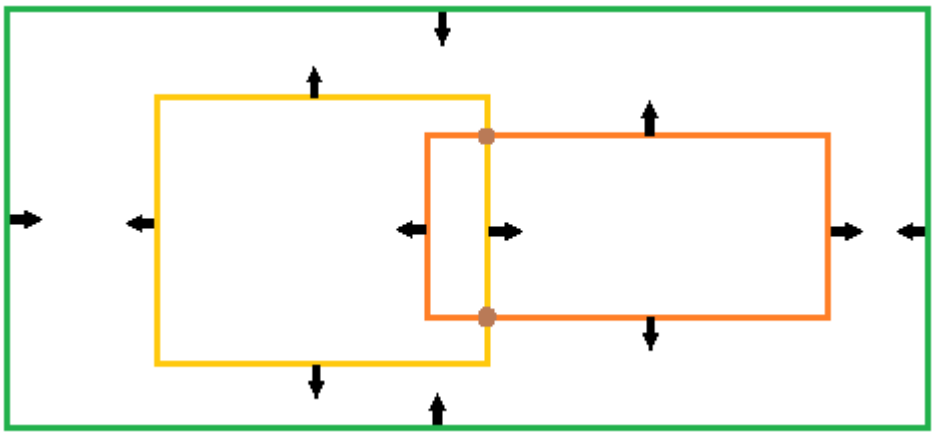


Then, the program finds the contact face and performs a local shift (offset) of surfaces (even if the surfaces are already intersecting each other). This causes intersection of the internal surfaces. The shift is done only in the local area of the contact contact, this area includes the nearby facets according to the value of the **Min gap** parameter, see the section Loading a geometry model of computational domain into a project).

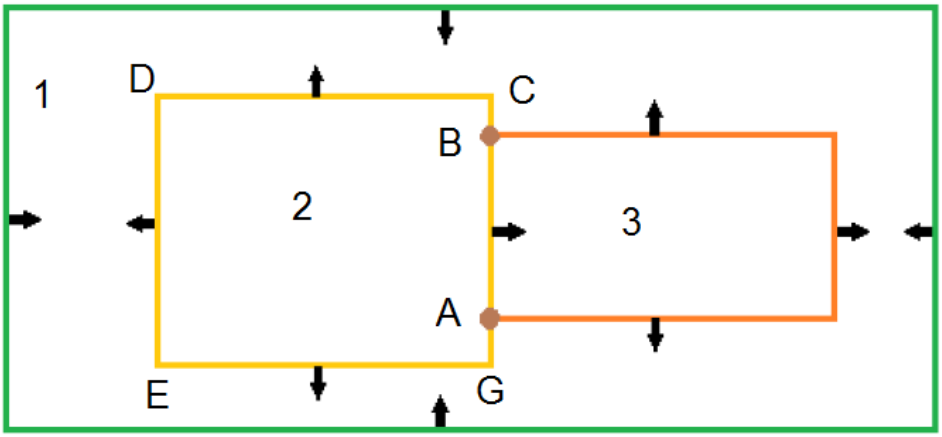


The **Min gap** parameter is the criterion, which determines overlap of two surfaces. If the distance between the surfaces is less then **Min gap**, they will be joined using shift (offset) even when they intersect each other, because this is necessary for making a correct Boolean sum of sets.

Then, if **Remove internals = Yes** is specified in the **Part intersection fix** window, the program cuts out the intersections of internal parts.



When removing unnecessary internal volumes, the program removes the surface, which has the greater area. The unnecessary internal volumes are those, which have at least two back sides of their surfaces.

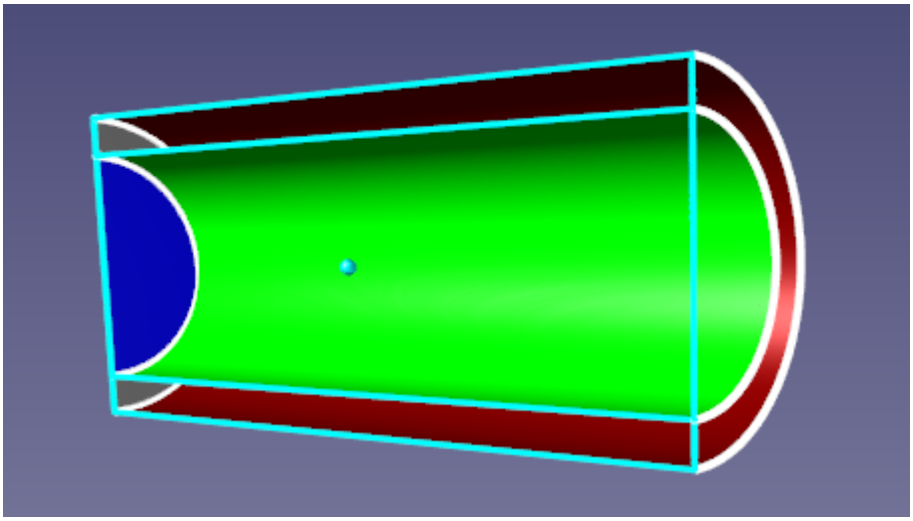


As a result of successful creation an assembly with union and removing internal volumes, surfaces of **Subregions** will be ready to setting boundary conditions on them and conjugating the **Subregions**. At this step, in our example, between **Subregions 2** and **3**, a surface **AB** appears, which is common to both **Subregions** and is ready tor setting a **Connected** boundary condition on it and for creation a **Binder**. Similarly, it is possible to set a conjunction of **Subregions 1** and **2** after creation **Connected** boundary condition for the group of surfaces **BCDEG**. So, it is *not* necessary to split the surface **CG** into fragments **AB**, **BC**, and **AG** beforehand in the CAD-system.

Without automatic removal of the internal volumes, you have to remove them manually.

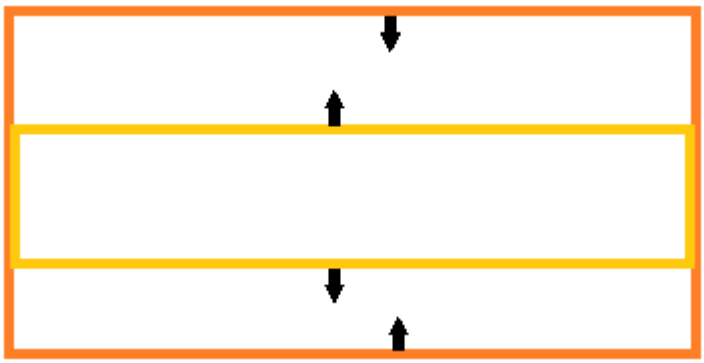
6.8.1.2.3.5 Example of assembly with union

Below you can see an example of creating a thick-walled tube with closed ends.

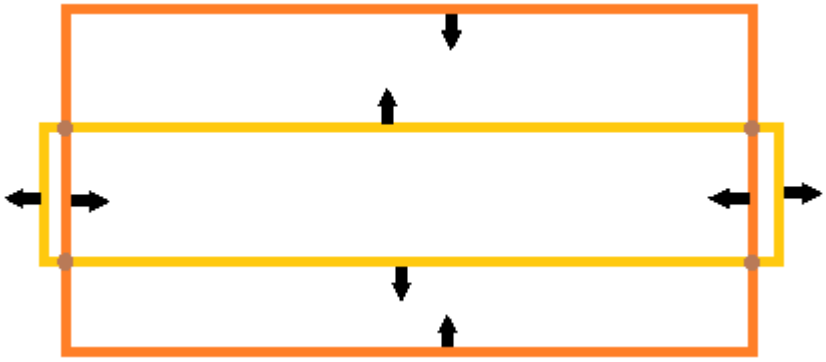


Example of a surface with multiconnection (a thick-walled tube with liquid inside)

Let's load two tubes with coupling ends. The large tube is loaded as an outer surface *(it is the first in the list of surfaces!*).



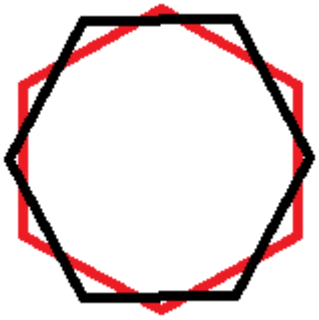
After determining the contact face, shifting a contacting surface, and mutual cutting the surfaces, there will be:



After removing the volumes, which have two back sides of their surfaces, we will obtain the desired geometry with multiconnection:

6.8.1.2.3.6 Possible errors when making an assembly

1. The geometry model, which has been created as a result of assembling with the specified parameters, does not meet the *FlowVision*'s requirements to geometry surfaces.
2. An incorrect value of **Min gap** (for example, when **Min gap** is less then the distance between surfaces, which should be connected).
3. If an unwanted contact of surfaces was not prevented, this could be caused by too small value of the **Surface offset** parameter.
4. You cannot create assemblies from multiconnected surfaces.
5. If the geometry model is too complicated, the algorithms of separation and union can output an error message. Unfortunately the algorithms are not all-powerful.



An example of bad approximation of curved surfaces by facets, which causes difficulties during the assembling if the offset is insufficient

6.8.1.2.4 Adding or removing subregions in geometry models

You can add a geometry surface into a project, so a **Subregion** is created, in the following cases:

- in an empty project
- in a project with existing geometry model of the **Region**

To do so, you have to follow the steps:

1) Create a standard geometric objects (**Box**, **Cone/cylinder**, **Ellipsoid/sphere**) or load an **Imported object** from a file.
2) Embed this **Object** into the main geometry, so a new **Subregion** will be created.

Thus you can make an assembly as you load geometry files one-by-one and embed them into the main geometry. Before the embedding, you can previously transform the geometry (make shifts, rotations, scaling, or equidistant shift), and after this embed the geometry as a **Subregion**.

If an unnecessary **Subregion** appeared during the assembling, you can delete it. To do this, you have to delete its geometry surface from the project tree.

See also:

Description of the user interface and step-by-step instructions for adding/deleting surfaces see in sections:

- Adding a surface into a geometry model of computational domain
- Deleting a surface from a geometry model of computational domain

6.8.1.2.5 Embedding a Moving body's surface into the computational domain

Besides the geometry model of the **Region**, **Moving body** modifiers can also take part in forming the computational domain. Surfaces of **Moving bodies** are added into the computational domain and are used in the computation, and **Boundary conditions** are set on them.

Use of **Moving bodies** provides the following possibilities:

- **Moving bodies** can move in the computational domain according to the specified movement law with ability to take into account actions of the hydrodynamic force F_{hydr} and the hydrodynamic torque T_{hydr}.
- **Moving bodies** can be fixed (be stationary), forming either a body that is flown around or geometry of a flow channel.
- Simulation of physical processes can be done either outside or inside a **Moving body**. In the latter case **Inside out = Yes** is set in properties of the **Moving body** (this is set by the **Turn inside out** command from the context menu of the Moving body).
- **Moving bodies** can be replaced during the computation (you can change their geometry and continue the simulation). Note, that it geometry model of the **Region** also can be replaced, but after this you can run the computation only starting from scratch.
- A **Moving body** can intersect geometry surfaces of the **Region** and geometry surfaces of other **Moving bodies**.
- Various boundary conditions (**Wall**, **Inlet/Outlet**, **Symmetry**, etc.) can be set on surfaces of **Moving bodies**.

A **Moving body** can be formed:

- on an **Imported object** loaded from a file that meets the requirements to contents and formats of geometry files.
- on an **Imported object** that has been created based on a standard geometric object (**Box**, **Cone/cylinder**, **Ellipsoid/sphere**).

6.8.2 Requirements to geometry models

Geometry models are imported to *FlowVision* mainly for the following goals:

- for forming boundaries of the computational domain (the computational region) consisting of **Subregions** and **Moving bodies**.
- for postprocessing, when integral **Characteristics** are calculated on a geometry surfaces or fields of physical values are visualized on the surfaces.

General requirements to geometry models imported to *FlowVision*

Sizes in the geometry models are set in meters, [m].

Surfaces for forming geometry models of the **Region** and **Moving bodies** are to be closed. These surfaces might be nested inside each other or be spatially separated. For a **Moving body** only the outmost surface is used and all the inner surfaces are not used in the computation.

Intersections and/or coinciding of model's surfaces are not allowed with exception of **Moving bodies** that can intersect the geometry model of the **Region** and intersect each other. Coinciding of surfaces in the latter case is also not allowed.

Self-intersections of surfaces (when facets of the same surface intersect or conjunct) are not allowed.

Degenerate facets are not allowed (when the angle between two edges of a facet is near zero).

Near conjunctive facets with same faces are not allowed (when the spatial angle between two adjacent facets is near zero).

Import of a geometry model requires specifying the geometry tolerance (the **Tolerance** parameter in the **Geometry import** group of parameters in the basic settings of **Pre-Postprocessor**) taking into account the order of magnitudes of coordinates of nodes.

Accuracy of presenting a surface depends on its triangulation (sizes of facets) and it is set in the CAD-system at exporting the file with mesh representation of surfaces.

Accuracy of the solution can depend not only on resolution of the space by the computational grid, but also on quality of the surface triangulation. For example, in simulations of a flow around an airfoil, it is generally required to provide very accurate triangulation of the surface.

Supported formats of loaded geometry files of computational domain and imported objects

File name extension	Format	Type
wrl	ASCII	*VRML*
stl	ASCII, binary	*STL*
mesh	binary	*3DTransVidia*
dat, **bdf**, **nas**	ASCII	*NASTRAN data*
inp	ASCII	*ABAQUS input*
cdb	ASCII	*ANSYS Text Database*
ngeom	ASCII	*CEDRE NGEOM*
vtl	ASCII, binary	*VTK file*
cel	ASCII	*STAR CD CEL file*

But *FlowVision* does not support all possibilities of various formats. See details about requirement to files in various formats in the section Requirements to contents and formats of geometry files.

See also:

- Self-intersections of surfaces and their correction
- Removal of too-small facets

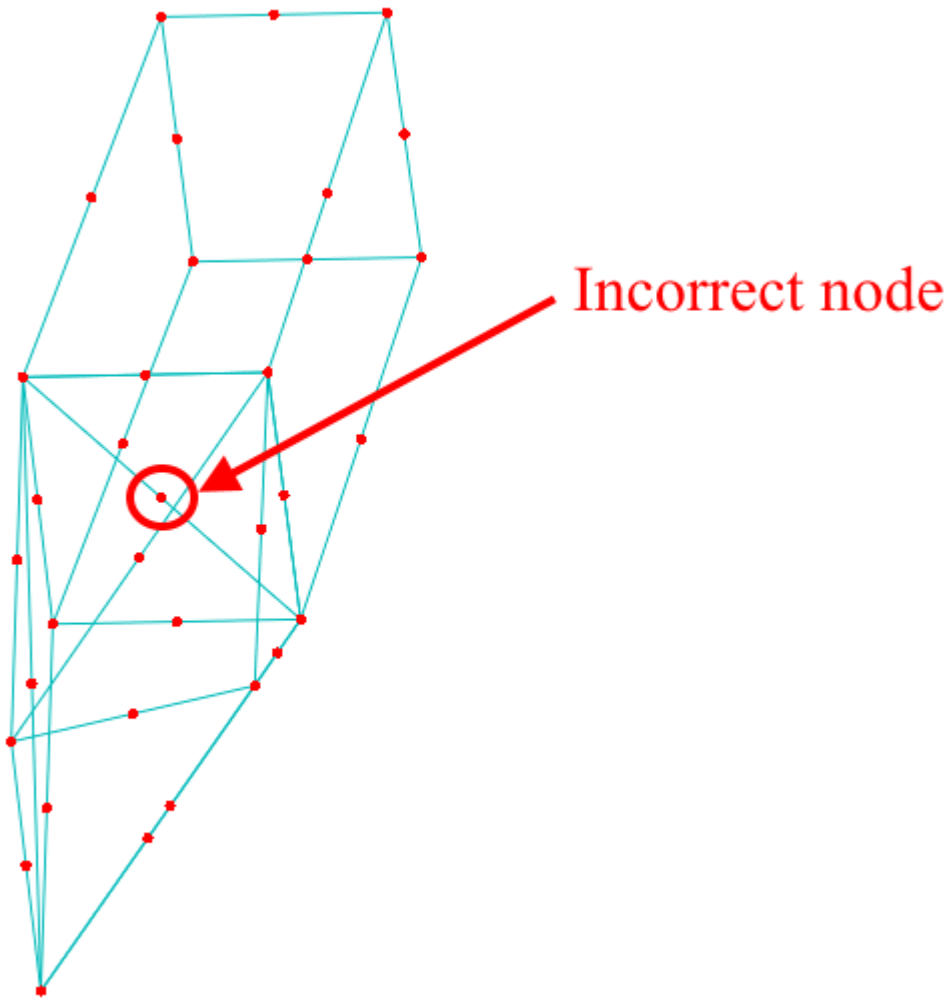
6.8.2.1 Requirements to contents and formats of geometry files

Geometry files contain information about nodes of the mesh representation of the geometry model.

Requirements to geometry from FEA software

It is not possible to use both hexahedral 20-node elements and tetrahedral 10-node elements because an edge of the tetrahedral elements appears a node, which doesn't match to any node of the hexahedron.

In this case the generated grid will get a tie-link between nodes. Use of such grid is not allowed in *FlowVision*.



Example of not allowed use of hexahedral 20-node elements and tetrahedral 10-node elements. The node in the middle of the tetrahedron's edge located on the hexahedron's face doesn't contact any hexahedron's node so this node becomes an "orphaned" one.

When you create an interface surface, you have to select only outer surfaces of the transferred body. Sometimes, when an interface surface is selected, internal planes come into the grid, if the grid has been created by cutting the geometry.

Elements of *VRML*

FlowVision can import all the elements of *VRML*, except the following ones:

- nodes of the **proto** type
- links that assume loading some geometry from other files
- interactive elements

Elements of *Abaqus*

FlowVision can import 2D elements of the following format:

<type><number of nodes><any letter suffix>

where **<type>** can be:

- **C2D**
- **CPE**
- **CPS**
- **C2DG**
- **CPEG**
- **CPSG**
- **S**

- **DS**
- **STRI**
- **SFM3D**
- **M3D**
- **R3D**
- **SC6R**
- **SC8R**

and **<number of nodes>** can be **3**, **4**, **6**, **8** or **9** (**3** or **6** only for the type **STRI**; **3** or **4** only for the type **R3D**).

FlowVision can import 3D elements of the following format:

C3D<number of nodes><any letter suffix>

where **<number of nodes>** can be **4**, **6**, **8**, **10**, **15**, **15V** (18 nodes), **20**, **27**.

Elements of *Ansys*

FlowVision can import *2D* and *3D* elements listed in the tables below:

2D-elements

6-node triangles	4-node quadrangles	8-node quadrangles
PLANE2	PLANE13	PLANE53
PLANE35	PLANE25	HYPER74
PLANE146	PLANE42	PLANE77
	PLANE55	PLANE78
	HYPER56	PLANE82
	PLANE67	PLANE83
	PLANE75	HYPER84
	VISCO106	VISCO88
	PLANE162	VISCO108
	PLANE182	HF118
		PLANE121
		PLANE145
		PLANE183

3D-elements

8-node hexahedrons	10-node tetrahedrons	20-node hexahedrons
SOLID5	SOLID87	VISCO89
SOLID45	SOLID92	SOLID90
SOLID46	SOLID98	SOLID95
HYPER58	HF119	SOLID117
SOLID62	SOLID123	HF120
SOLID64	SOLID127	SOLID122
SOLID65	SOLID148	SOLID128
SOLID69	HYPER158	SOLID147
SOLID70	SOLID187	SOLID186
HYPER86		SOLID191
SOLID96		
SOLID97		

8-node hexahedrons	10-node tetrahedrons	20-node hexahedrons
VISCO107		
SOLID164		
SOLID185		

Elements of *Nastran*

FlowVision imports from the file the contents of the section *BULK*. If the *BULK* is absent, the program cannot recognize the file signature and select a matching library for data import.

FlowVision can import 2D-elements of the following types:

- 3-node triangles: *CTRIA3*, *CTRIAR*
- 6-node triangles: *CTRIA6* (additional nodes can be absent)
- 4-node quadrangles: *CQUAD4*, *CQUADR*, *CSHEAR*, *CTWIST*
- 8-node quadrangles: *CQUAD8* (additional nodes can be absent)

FlowVision can import 3D-elements of the following types (additional nodes can be absent):

- *CHEXA*: a box (8 - 20 nodes)
- *CPENTA*: a prism with a triangle base (6 - 15 nodes)
- *CPYRA*: a pyramid with a quadrangle base (5 - 13 nodes)
- *CTETRA*: a tetrahedron (4 - 10 nodes)

6.8.2.2 Multiconnection

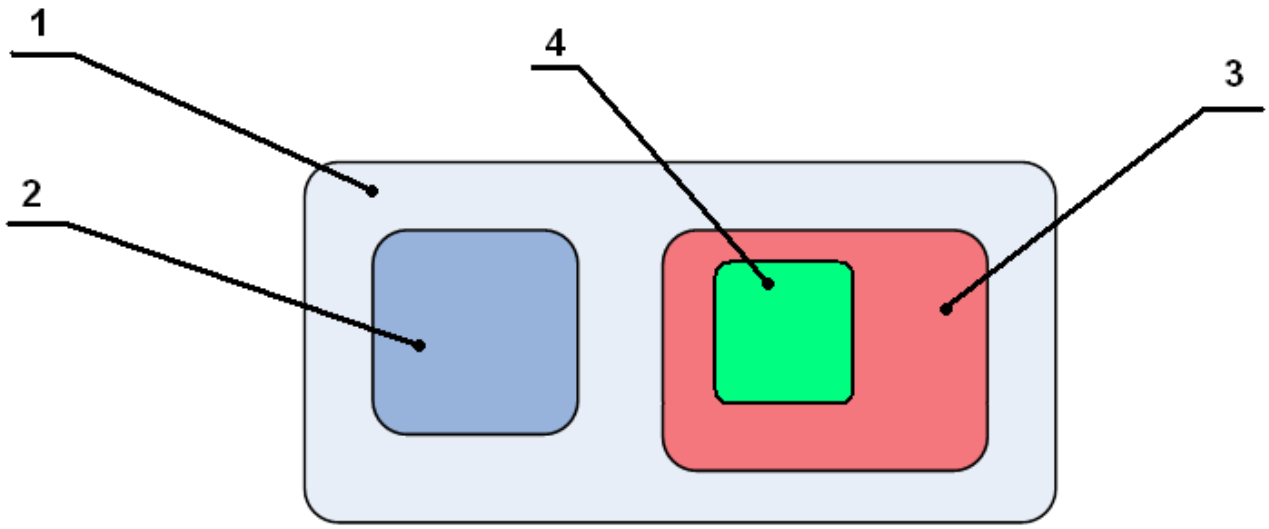
Geometry model without multiconnection

A geometry model of the computational domain *without multiconnection* is a set of closed surfaces, which meet to the following requirements:

- the surfaces must close a finite volume and be closed
- the surfaces must not touch or intersect each other
- the surfaces must not intersect themselves

For example, the Klein bottle does not meet to these requirements.

Such computational domain is a sum of subregions separated by closed non-intersecting surfaces.



Example of a computational domain without multiconnection (it consists of **Subregions** 1, 2, 3, and 4)

The whole set of **Subregions** can be created by:

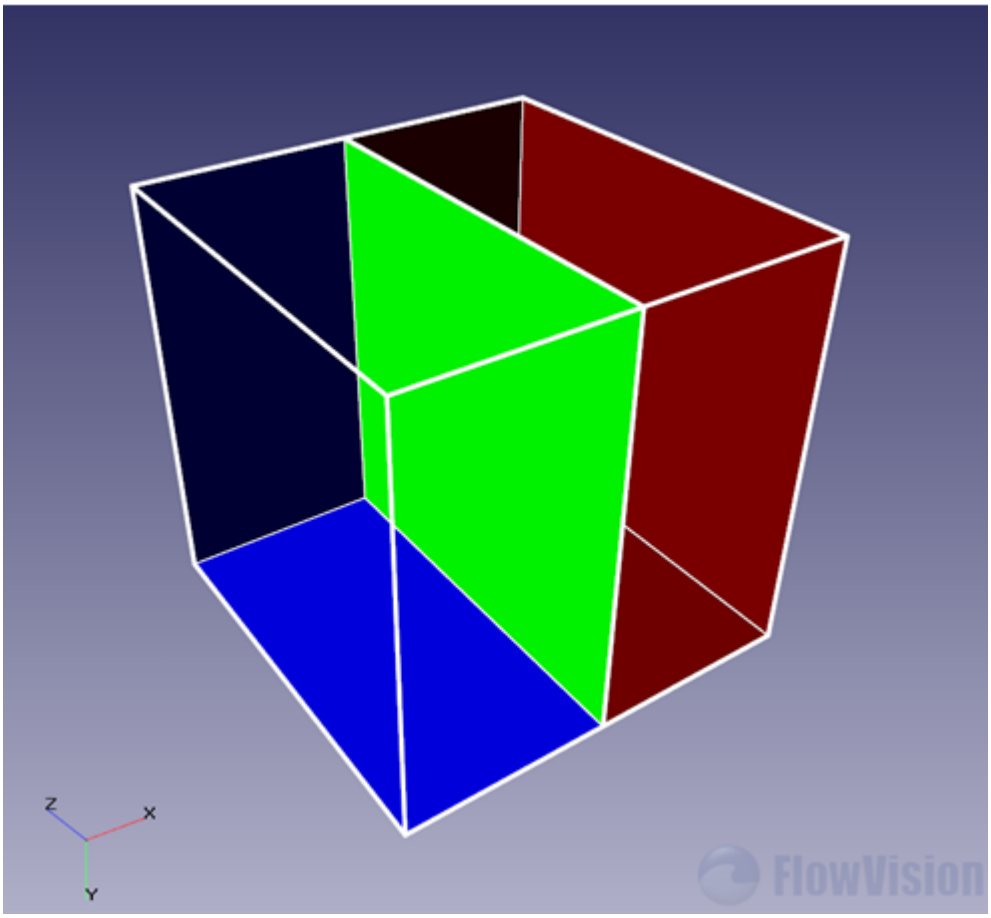
- loading from a single file that contains the geometry model, which include consists of several surfaces
- successive adding surfaces from separate files
- a batch import from several files when assembling is done, see section Creating assemblies and step-by-step instructions in subsection Loading geometry model from multiple files (assembly).

Geometry model with multiconnection

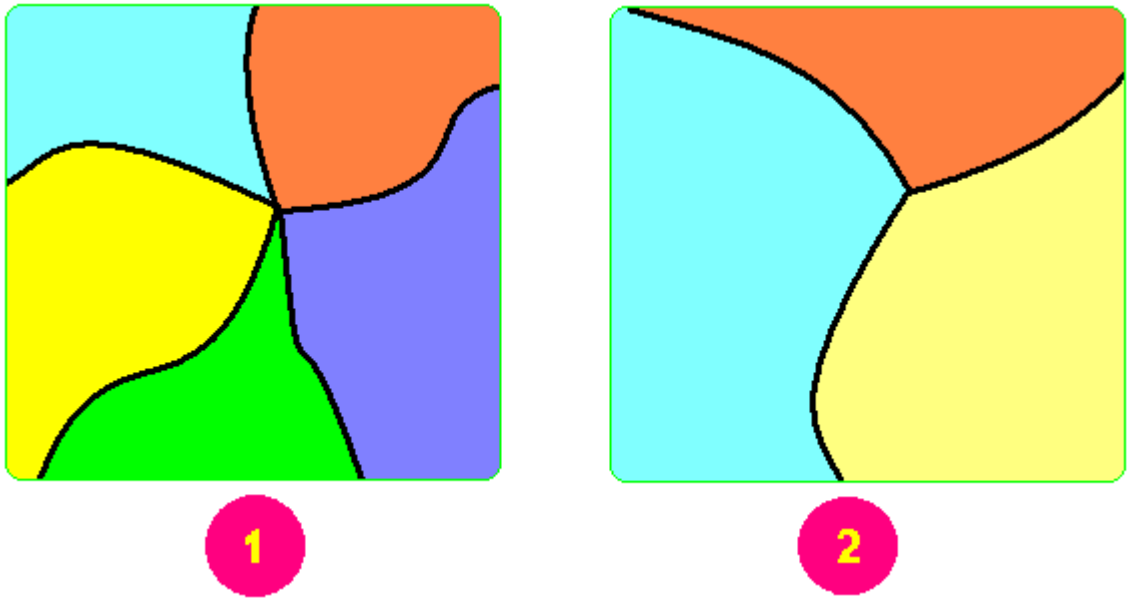
Multiconnection is connection of three or more surfaces by a single edge. A special case of multiconnection is a T-connection of surfaces.

You can enable or disable the multiconnection using the basic settings of Pre-Postprocessor.

A geometry model with multiconnection can be available in *FlowVision* via loading files with parts in the "assembling with union" mode. It is possible to export a geometry model with multiconnection into a file with its further use in *FlowVision*.



Example of multiconnection



Examples of connection of surfaces with multiconnection:
1 - connection of five surfaces; 2 - T-connection of three surfaces

When multiconnection is used, the project should have **Models** specified in all **Subregions** that are adjacent to any surface forming the multiconnection. If this requirement is not met, then all **Subregions**, adjacent to the multiconnection, will be marked in the project tree with a "**!**" symbol and the project will not be able to run.

In geometry models with multiconnection, **Subregions** are formed between the surfaces. The same operations can be applied to them, as those, which can be applied to **Subregions** without multiconnection (so the **Subregions** can be removed, transformed, exported into files).

The surfaces, which form a multiconnection, correspond to the elements **Complex surface** in the project tree (see section Folder «Geometry»).

6.8.3 Surfaces and groping of facets

The geometry model is presented by a set of nested and/or spatially separated surfaces.

Surfaces of the **Region**, of **Imported objects**, and of standard geometric objects are mesh surfaces formed by flat facets.

For convenient work with facets they are grouped by some criterion (by color, angles of normals of the facets, by boundary conditions that are set on the facets, etc.). Also it is possible to regroup the facets (do their merging and/or splitting).

See also: Groups of facets as geometric objects.

6.8.3.1 Facets

All surfaces in *FlowVision* are represented by polyhedra with flat faces (facets). Typically, the facets are triangles.

Surfaces of the standard geometric objects might also include tetragonal facets, but when the program creates **Imported objects** from the standard objects, all rectangular facets are converted into pairs of adjacent triangular ones.

Terminology note: For facets that form surfaces of the *FlowVision*'s geometry models and objects, the "triangle" term can be used, because such facets are always triangles.

Facets of geometry models of external CAD-systems can be not triangles only.

The geometry model of the computational domain is created in an external CAD-system and is loaded into **Pre-Postprocessor**.

A geometry model, which is imported into *FlowVision*, can be in one of the following formats:

- *VRML 1.x*
- *VRML 2.0* (***.wrl**)
- *STL* (***.stl**)
- *Abaqus* (***.inp**)
- *ANSYS* (***.cdb**)
- *NASTRAN* (***.dat**, ***.bdf**, ***.nas**)
- *3DTransVidia Mesh* (***.mesh**)
- *Legacy VTK* (***.vtk**)
- *StarCD cel/vrt* (***.cel** + ***.vrt**)
- *CEDRE NGEOM* (***.ngeom**)

Surfaces of the geometry model are created in an external CAD system and are imported into *FlowVision* as a set of triangular facets.

During the import of the geometry model of the computational domain, the program performs several checks of the surface's quality, including the following ones:

- check if surfaces are closed
- search for facets, which are set at very acute angles to each other
- search for more than two facets with a common edge

If such errors are found, it is necessary to correct the geometry model in the external CAD system and reload the geometry model. Also it is recommended to check the surfaces for self-intersections.

6.8.3.2 Grouping the facets

The surfaces of a geometry model of the computational domain consist, as a rule, of a large number of triangular facets, which, for convenient work with the program, are joined into **Groups**.

Facets automatically are grouped at importing of the geometry model according to some or other grouping (see subsection *"Grouping at importing the geometry"* below). During the further work with the project you can regroup the geometry.

Grouping the facets allows you, when it is necessary, to select not individual facets but whole fragments of surfaces (for example, when boundary conditions are set).

Standard geometric objects also can form groups of facets: these are faces of a **Box**, bases and lateral surface of a **Cone/cylinder**, and the whole surface of an **Ellipsoid/sphere**. These groups of facets are used to analyze the results of the project's computation (to create **Characteristics** and **Layers**).

Visualization of outlines of groups of facets in the **View** window is turned on by button  (**Enable/disable display of face group borders**) in the **Solids** toolbar. Color and thickness of these outlines is set by parameters **Outlines >...** in properties of the layer **Solids**.

Grouping at importing the geometry

When importing into the project a geometry model of the computational domain or an imported object, subdivision into groups is performed automatically, using as a criterion the angle between facets, which is compared to the *grouping angle*, which is set by the parameter **Geometry import > Grouping angle** in the basic settings of **Pre-Postprocessor**. When the angle between adjacent facets is less then the specified angle, the facets are grouped.

Splitting is performed according to the color of the facets when importing from *VRML* and *3DTransVidia Mesh* formats if the facets from which the model facets are being formed were colored during the creation of the geometry model of the computational domain or the imported object in a CAD system.

During import from *STL*, *Abaqus* and *StarCD sel/vrt*, the facets of the object from which the model facets are formed are split automatically into groups.

When importing an object from other formats, grouping information is either read directly from the source file or the splitting into groups is performed according to the materials and boundary conditions set in the source file:

- when importing from *VTK* format, grouping information (if present) is used, or automatic splitting is performed according to the default set grouping angle;
- when importing from *VRML* format, splitting is performed according to indicated colors/"materials";
- when importing from *NGEOM* format, splitting is performed according to set boundary conditions;
- when importing from *ANSYS* and *NASTRAN* formats, splitting is performed according to the set materials of finite elements;
- when importing from *Abaqus* format, automatic grouping is performed.

The following criteria are used for automatic grouping of facets into a single group:

- facet color - facets of same color are grouped together

- angle between neighboring facets - the facets with an angle between normals less than critical are united into a group (the value of the critical angle is set by the user)

Note: All such facets are split into triangles. The facets may be not triangles, but, for example, quadrangles or other polygons in the imported model. All such facets will be split into triangles.

See also:

- Regrouping of geometry model of the computational domain (and moving bodies)
- Procedure of regrouping a geometric model of computational domain (and moving bodies)
- Groups of facets as geometric objects
- Folder «SubRegion #N > Geometry»

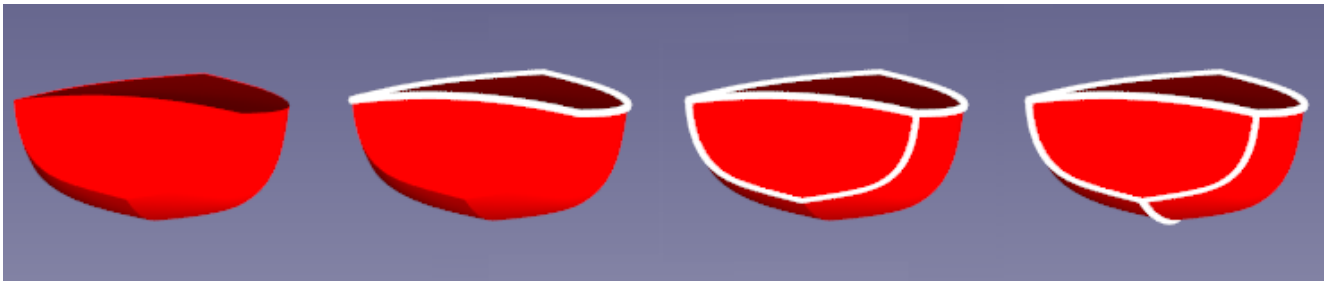
6.8.3.3 Regrouping the facets

Regrouping the geometry is change of splitting/merging geometric surfaces into **Groups** of facets.

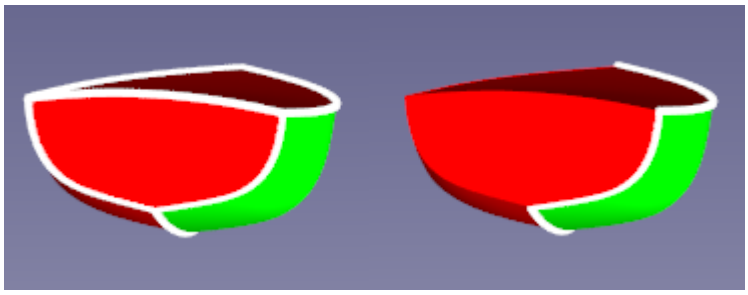
The regrouping is done according to parameters, which you set in the **Geometry regrouping** window. The most important criteria, which define the regrouping, are values of angles between adjacent facets.

Regrouping forms a new set of **Groups**:

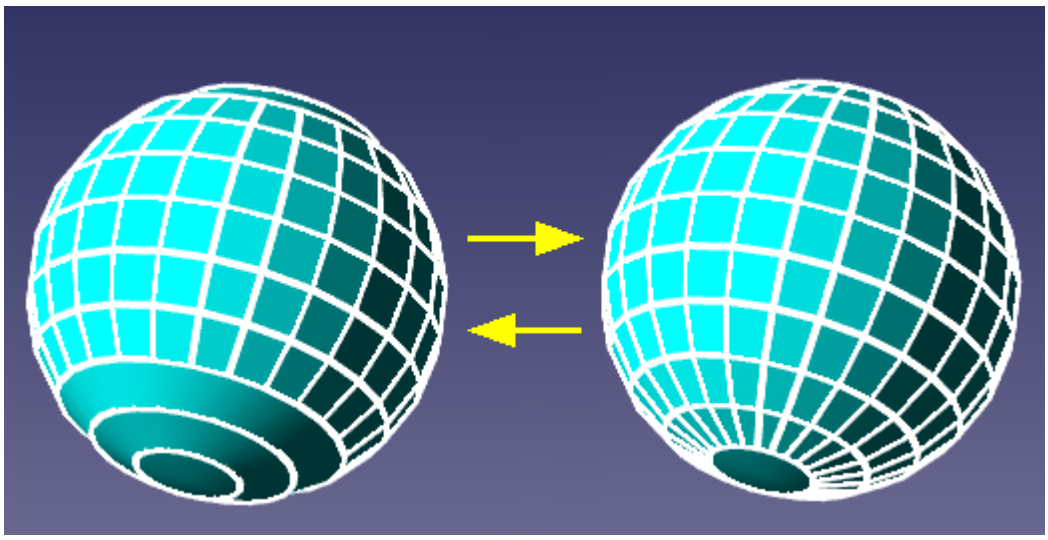
- if only one **Group** of facets is regrouped, then the new set of **Groups** is formed by splitting the initial **Group**
- if two or more **Groups** of facets are regrouped, then the new set of **Groups** is formed by merging and/or splitting the initial set of **Groups**



Depending on the settings, grouping of facets gives different results



You can enable or disable merging facets, on which different boundary conditions are set



Example of regrouping of facets of a **Sphere** with polar approximation

See details:

- Regrouping a geometry model of computational domain (and moving bodies); this section contains step-by-step procedures and details about the user interface of the **Geometry regrouping** window.
- Grouping the facets
- Groups of facets as geometric objects
- Folder «SubRegion #N > Geometry»

6.8.4 Troubleshooting and fixing geometry issues

Issues with the geometry appear when the geometry model doesn't meet the requirements, for example, when it has intersections or contacts of surfaces.

FlowVision includes several features for troubleshooting the geometry:

- Automatic troubleshooting of the geometry model at creating the project from a file.
- Manual troubleshooting the geometry model of the **Region** and/or **Moving bodies** and **Imported objects** for self-intersections (see the step-by-step instruction in the section Checking a geometry model of computational domain and moving bodies for self-intersections).
- It is possible to enable validating the computational grid structure at each its change (by default the program examines correctness of the computational grid only when the grid is created). Defects of the computational grid in most cases are caused by issues with the geometry model.

Resolution of issues with the geometry might include:

- Fixing the self-intersections that were found in the *FlowVision* project (see the step-by-step instruction in the section Fixing self-intersections of a surface in a geometry model).
- Troubleshooting and fixing the problems caused by overall dimensions and value of **Tolerance**. Examine the units that have been used (the program uses meters, [m]). Check that **Tolerance** has similar magnitude as the minimal size of facets (this size in shown as value of the **Minimal edge** parameter in properties of the **Region** and in properties of **Imported objects**). When the program simulates large objects, you don't have to use the small default value of **Tolerance**. And otherwise, when small objects are simulated, smaller value of **Tolerance** should be used.
- Removal of too-small facets.
- Fixing defects of the geometry model in the CAD-system.
- Tuning parameters of accuracy of exporting the file from the CAD-system (both too rough and too precise accuracy of presenting the surface by triangles can cause self-intersections).
- Use software products that work with mesh geometry models and are able to fix their defects (for example, the *3DTransVidia* software).

6.8.4.1 Self-intersections of surfaces and their correction

Self-intersection of a surface appear due to different causes. *FlowVision* has features to find the self-intersections (see step-by-step instruction in the section Checking a geometry model of computational domain and moving bodies for self-intersections).

Checking the geometry for self-intersections can be done for:

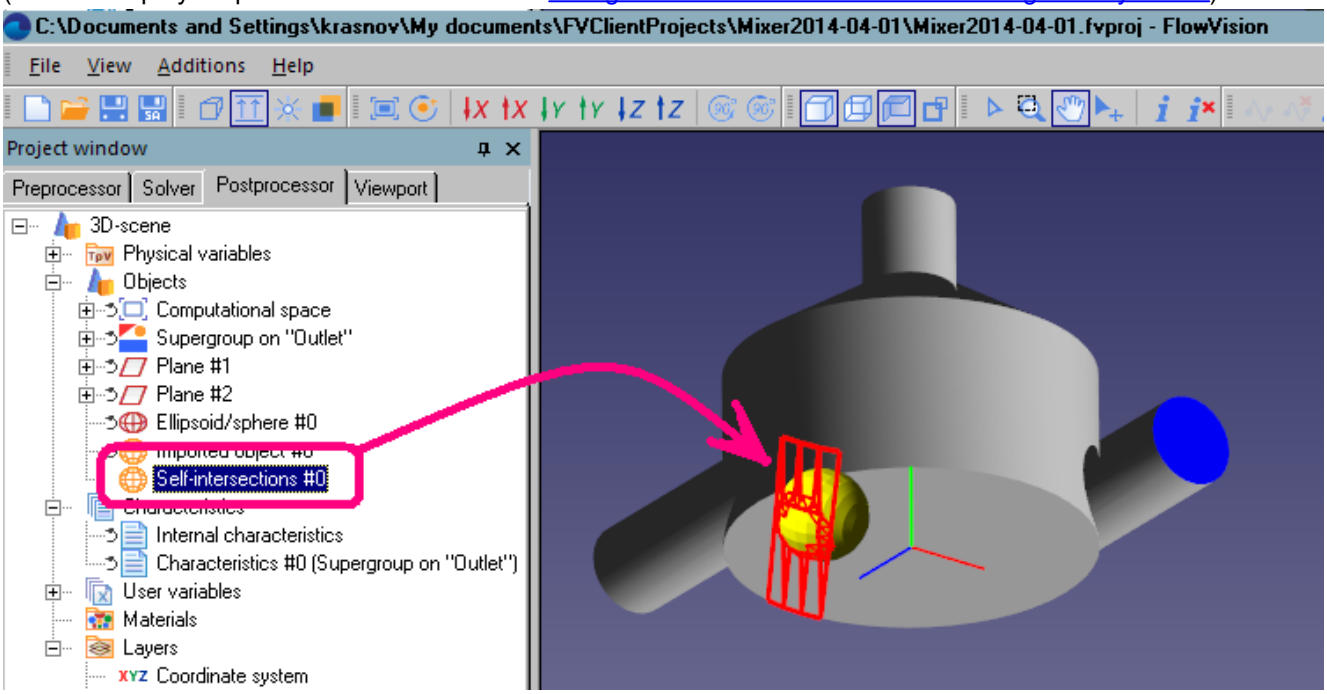
- **Region**
- **Region** along with **Moving bodies**
- an **Imported object** (note, that surface of the **Imported object** is allowed to be not closed until its geometry is used in the computation)

Checking for self-intersections can give the following results:

- Self-intersections are not found (but anyway it is recommended to check the model's overall dimensions specified in meters, [m], and **Tolerance**).
- Self-intersections are found. In this case a **Self-intersections #N** object will be created (this is a special **Imported object** consisting of self-intersecting facets). Facets of the object **Self-intersections #N** correspond to facets of the geometry model, which has been checked). The **Self-intersections #N** object is highlighted in the **View** window with red lines.

> Absence of found self-intersections doesn't guarantee that there are no problems with the geometry. The troubleshooting algorithm is improved at each new version of the program.
>
> The check doesn't detect contacts of **Moving bodies**, but the contacts will cause issues at the stage when the computational grid is built.

If the found self-intersections of the surface have been caused by minor overlaps of facets, so the volume formed by the self-intersections is negligible, it is possible to use *FlowVision*'s feature for precise removal of such areas (see the step-by-step instruction in the section Fixing self-intersections of a surface in a geometry model).



Correction self-intersections

FlowVision has the feature of automatic *correction self-intersections* of surface of the computational domain or the selected geometrical object.

> The functionality of automatic fix of self-intersections does not guarantee positive results, because its algorithm is not applicable for all situations.
>
> After checking for self-intersections examine correctness of the geometry model an, after fixing its defects (if any), run checking for self-intersections once again.

Fixing of self-intersections can be run:

- for the whole **Region**
- individually for selected **Imported objects**

- for elements **Self-intersections #N**

For step by step procedures of fixing self-intersections see descriptions in the sections:

- Fixing self-intersections of a surface in a geometry model
- Fixing self-intersections of the surface of an imported object

The procedure of fixing self-intersections procedure for an element **Self-intersections #N** is similar.

If the fixing of self-intersections failed, we recommend you to:

- Examine overall dimensions of the geometry model (they are specified in meters, [m]). At its export, the geometry model might be saved in millimeters, [mm], or in inches. In this case you have to carry out scaling of the geometry model in *FlowVision* (see Transformation of geometry model of the computational domain and an imported object).
- Examine **Tolerance**. When simulated objects are very large (kilometers) or very small (microns), change the default value of **Tolerance**.
- Remove too-small facets.
- Make corrections of the geometry model in the CAD-system (for example, eliminate coincidence of surfaces so an ensured gap will appear between the surfaces).
- Change parameters of exporting the geometry model from the CAD-system.
- Or to fix defects of the mesh in a specialized software (for example, in *3DTransVidia*).

6.8.4.2 Removal of too-small facets

The procedure of *removing too-small facets* is necessary for cleansing the model of too-small facets and subsequent geometry correction. A facet is considered as small if one of its edges is smaller than the largest edge of any triangle in the geometry by six orders of magnitude, i.e., the minimal edge is less than 10^{-6} of the maximal edge size of any facet in the geometry model. Such small facets can cause errors in building the computational grid and it is recommended to remove them.

Removing a too-small facet on a surface is accomplished by merging it with a neighboring larger one. This is performed when deleting the common node of large and small facets, which belongs to the small edge of the small facet.

The minimal value of an edge in the surface of an **Imported object** is displayed its property **Minimal edge**.

After removing too-small facets check correctness of the geometry model once again because removing facets changes the mesh surface.

See also: a step-by-step procedure in the section Removal of too-small facets of geometry model of computational domain.

6.8.5 Visualization of surfaces

In *FlowVision* all geometry elements are presented in:

- the project tree of **Pre-Postprocessor** by individual elements and folders with elements (the root folder **Region**, folders **Objects**, **Boundary conditions**, etc.)
- the **View** window of **Pre-Postprocessor**. The program provide many visualization settings for geometry elements (standard geometric objects, **Imported objects**, elements of the geometry model, surfaces of **Moving bodies** and some **Layers**).

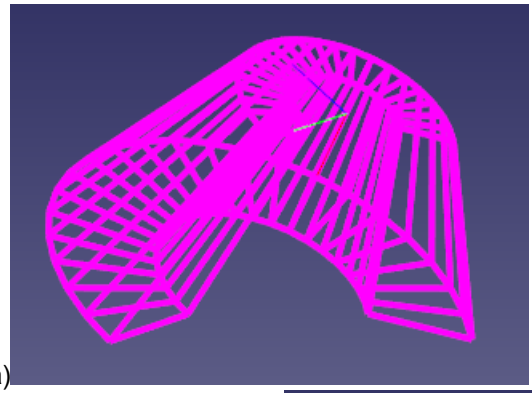
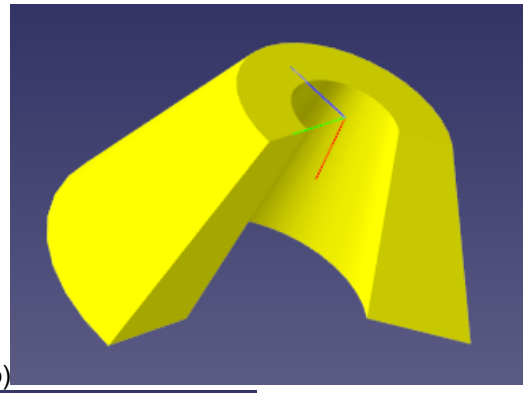
The visualization is set in **Postprocessor**:

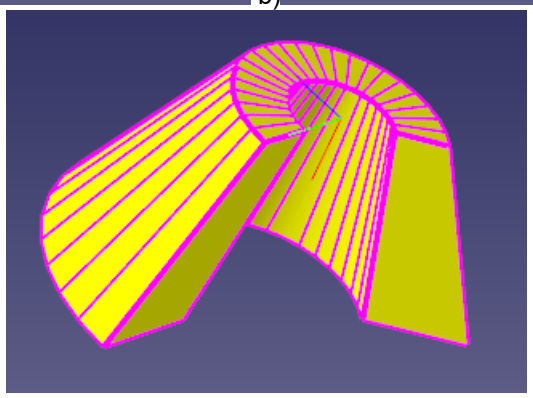
- local visualization settings for the selected object are set in its **Properties** window
- global visualization settings for all objects are set by buttons in **Toolbars**

Main methods of visualization of surfaces

The main methods of visualization of surfaces are:

- visualization by lines (by a wireframe)
- visualization by fill
- visualization by fill and lines

a)  b)  c)



Visualization of a surface: a) by lines (by a wireframe); b) by fill; c) by fill and lines

You can also set:

- Transparency
- Lighting and reflections of the surface
- Use of the perspective projection for 3D visualization
- Hiding the surfaces
- Cutting parts of surfaces by planes
- Adding a mirror reflection of the surface by a plane
- Forming the whole image of an axially symmetric body of sectors or combine images of individual **Subregions** (this is used to visualize sliding boundary conditions and rotating **Subregions**)
- Hide outer side of the surface (located towards to the observer)
- Colors of the fill and wire-frame used to visualize the surfaces
- Properties of **Materials** that determine gloss of the surfaces and ability to reflect the image of the "outer world" (by default the "outer world" is a landscape picture with the sky, mountains, lake and woodland)

The visualization settings can be set:

- globally
- individually
- be inherited from properties of the object, which locates a level up (this is applied when value of a setting is **Default**)

You can specify visualization settings individually to each geometry object.

Visualization of standard and imported geometric objects

By default standard geometric objects and **Imported objects** are visualized by lines (by wireframe that form the facet presentation of the surfaces). When the object is not selected in the project tree, its lines by default are displayed in aqua color, and when the object *is* selected in the project tree, its lines are displayed in red color (which is inverted color related to aqua).

Individual visualization settings of standard and imported geometric objects are specified in the objects' **Properties** windows in the **Postprocessor** tab:

- **Visible** (defines if the object's image will be visible)
- **Clipped** (defines if the object's image can be cut by a **Plane**)
- **Lighting** (defines if the object's image will be lit by light sources and have specular highlights)
- **Appearance > Mode = Lines | Fill | Lines and fill** (method of visualization)
- **Appearance > Lines > Color** (color of lines when the object is not selected in the project tree)

- **Appearance > Lines > Width** (thickness of lines)
- **Appearance > Fill > Color** (color of the image's surface when the object is not selected in the project tree. When the object is selected in the project tree, the inverted color is used.)
- **Appearance > Fill > Opaqueness =** ***value*** (this parameter specifies semitransparency of the object's surface; the semitransparency can be applied only when it is enabled in global visualization settings of **Pre-Postprocessor**).

Some objects have specific visualization settings, such as properties **Clipping object** and **Mirror** of a **Plane**.

Visualization of geometry model of the computational domain (surfaces of the Region and Moving bodies)

When surfaces of the **Region** and **Moving bodies** are imported into the computational domain they by default are visualized by fill. Color of a surface corresponds to the color of the **Boundary condition** that is set on it.

You can specify the following visualization settings:

- In the **Preprocessor** tab:
  - color of the **Boundary condition** (it also defines color of the surface)
- In the **Postprocessor** tab in properties of child elements of the folder **Layers > Solids**:
  - in properties of the folder **Solids > Subregions**
  - in properties of folders **Solids > Subregions > SubRegion #N**
  - in properties of the element **Solids > Subregions > SubRegion #N > Geometry**
  - in properties of elements **Solids > Subregions > SubRegion #N > Moving bodies > Moving body #M**
  - in properties of elements **Solids > Subregions > SubRegion #N > Boundary conditions > Boundary condition #M**

Individual settings of child elements, when they are set as **Default**, override the parent and global settings.If some visualization setting is specified as **Default**, then appropriate setting of the parent element or global setting of **Pre-Postprocessor** is applied.

By default the surfaces are opaque (they are displayed without semitransparency). The semitransparency can be applied only when it is enabled in global visualization settings of **Pre-Postprocessor**).

The **Visible**, **Clipped** and **Lighting** settings are inherited from properties of the folder **Layers > Solids**.

When **Moving bodies** are displayed, visualization settings of their appropriate **Boundary conditions** have the higher priority.

Global visualization settings

The global visualization settings are applied to all displayed surfaces except those ones, to which different settings are specified (i.e. except surfaces with local settings).

The global visualization settings are loaded automatically at loading the geometry model of the **Region** or a **Moving body** and at creating a standard geometric object or an **Imported object**. By default the following rules are applied:

- the surfaces, on which **Boundary conditions** *are not* set (i.e. the surfaces that don't influence on the flow, standard geometric objects and **Imported objects**), are visualized by lines.
- the surfaces, on which **Boundary conditions** *are* set (surfaces of the **Region** and **Moving bodies**), are visualized by fill.

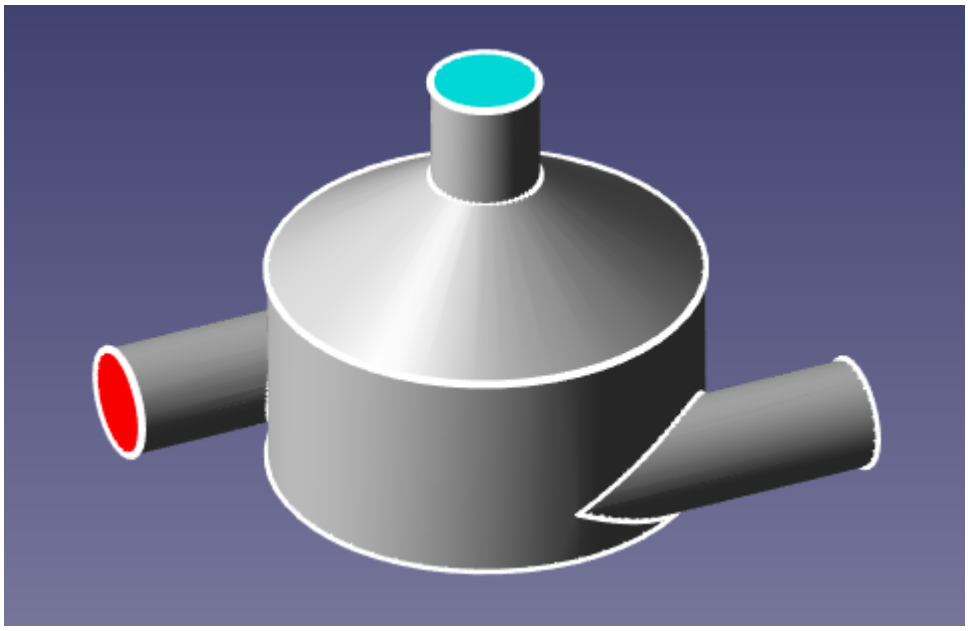
There are the following global visualization settings:

- Parameters **Display > ...** in the basic settings of **Pre-Postprocessor**. Here, for example, you can specify hiding of sliding and/or periodic surfaces at subregion positioning.
- Properties of **Materials** that allow you to set some visualization parameters (glares, etc.) of all surfaces, on which a specific **Material** is set.
- Parameters of light sources that are specified in the **Lighting** folder in the **Postprocessor** tab of the **Project** window.
- Images of outer world that are reflected by glassy surfaces (these images are stored in files `env_xp.bmp`, `env_xn.bmp`, `env_yp.bmp`, `env_yn.bmp`, `env_zp.bmp`, `env_zn.bmp` in the subdirectory `Textures`).
- Properties of the folder **Layers > Solids** in the **Postprocessor** tab:
  - **Visible**
  - **Clipped**
  - **Lighting**
  - **Outlines > Color** (color of outlines of groups of facets)
  - **Outlines > Width** (thickness of outlines of groups of facets)

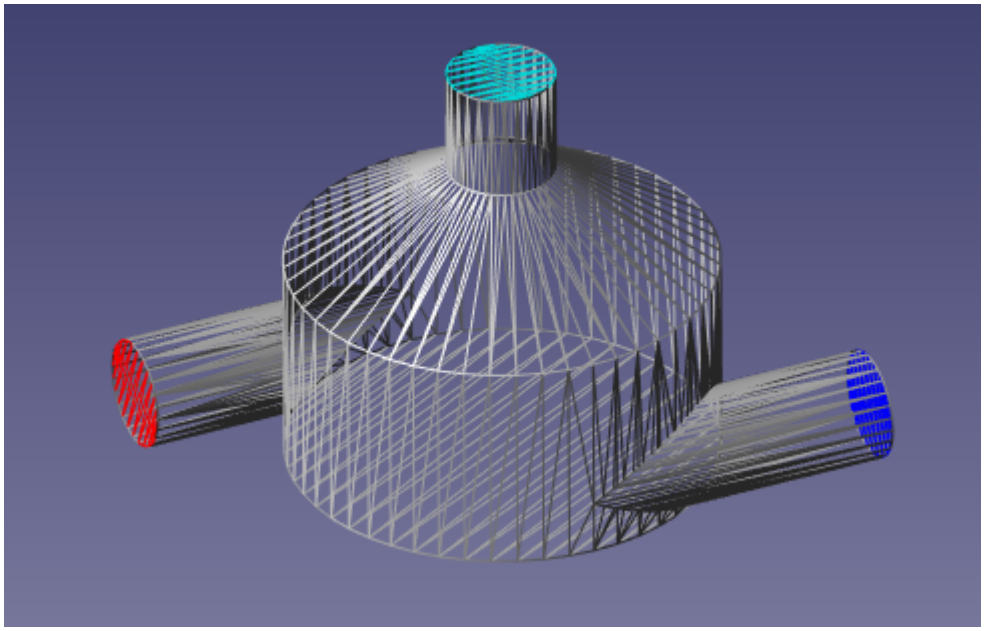
- **Outside > ...** (settings for visualization of the outer side of the surface in the case when the outer side is to be displayed in a different manner)

- Settings that are toggled by buttons in **Toolbars**:
  - displaying with perspective (this is toggled by the button in the **Rendering** toolbar)
  - small offset of the displayed surfaces to avoid their visual coincidence and caused by this difficulties of visualization, for example, when several **Layers** overlay each other on one **Plane** (this is toggled by the button in the **Rendering** toolbar)
  - displaying of specular highlights (this is toggled by the button in the **Rendering** toolbar)
  - support of semi-transparency (this is toggled by the button in the **Rendering** toolbar)
  - visualization the whole image of the simulated physical phenomenon in problems with a rotating **Subregion** and a sliding **Boundary condition**, for example, in simulations with a sector-sliding problem setting (this is toggled by the button in the **Rendering** toolbar)
  - visualization of surfaces by fill (this is toggled by the button in the **Solids** toolbar)
  - visualization of surfaces by wireframe lines (this is toggled by the button in the **Solids** toolbar)
  - visualization of outer surfaces that are oriented towards to the observer (this is toggled by the button in the **Solids** toolbar)
  - displaying or hiding outlines of facet groups of the main geometry (this is toggled by the button in the **Solids** toolbar)

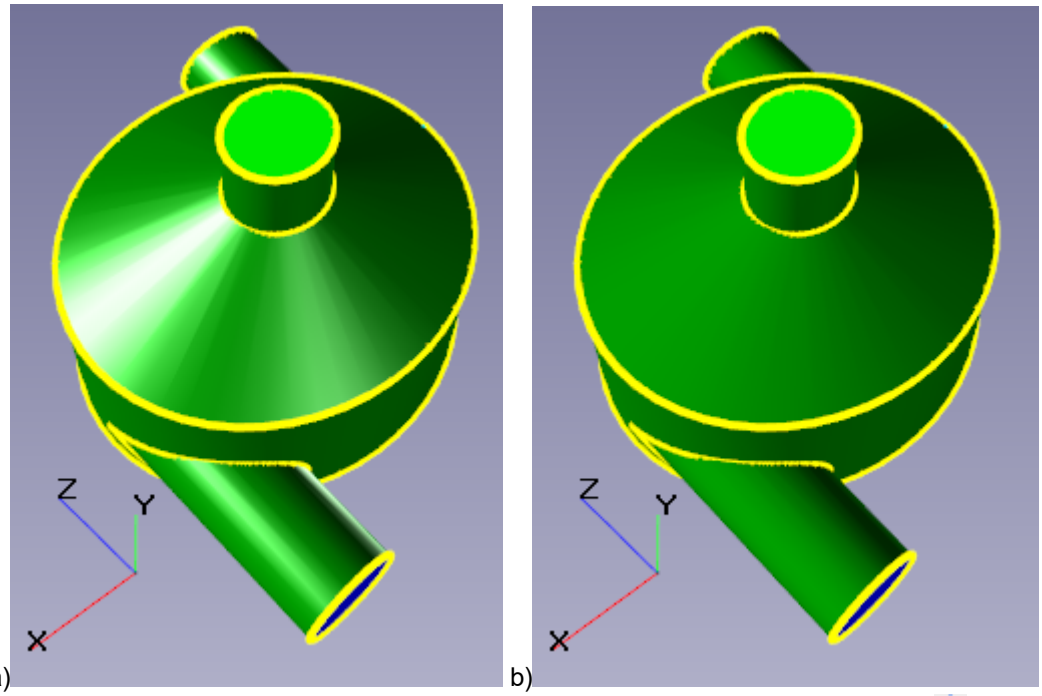
Illustrations



Outlines of groups of facets are visualized by white lines. Also fill of the surfaces is enabled.

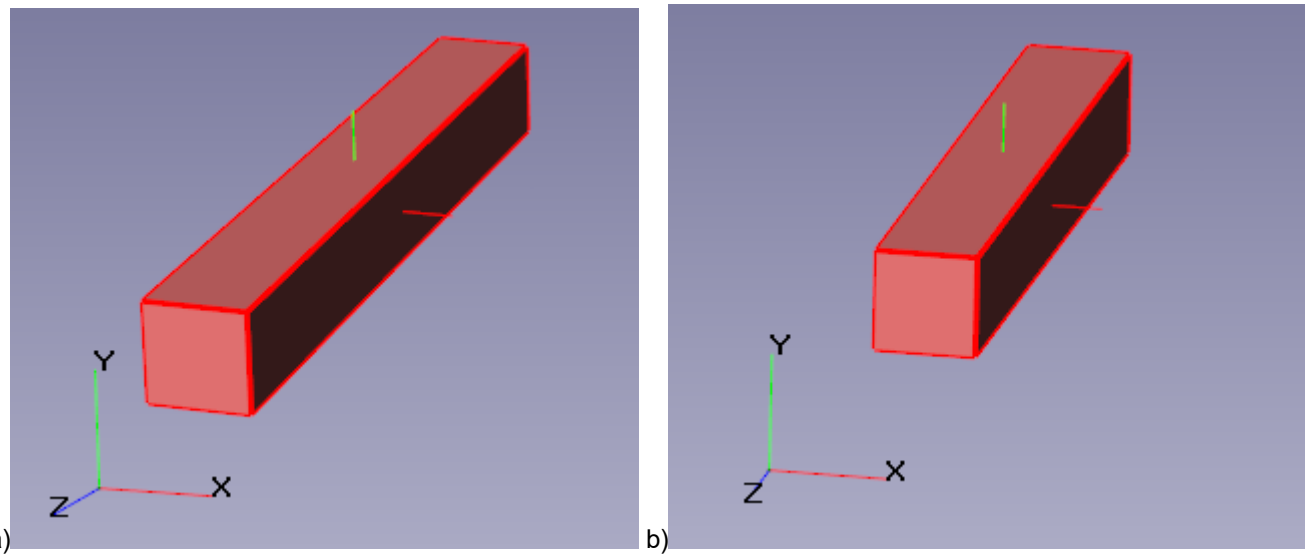


Wireframe visualization is enabled. Visualization of outlines and fill of surfaces are disabled.



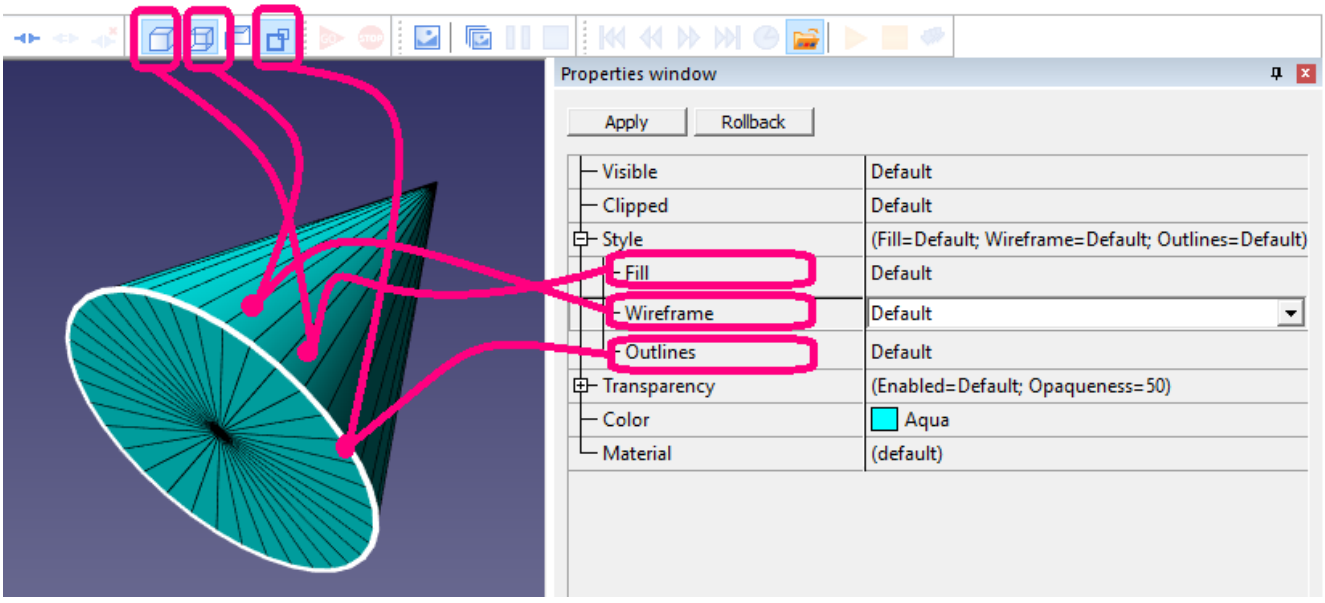
a) b)

Examples of visualization with enabled (a) and disabled (b) glares. This is toggled by the button in the **Rendering** toolbar.

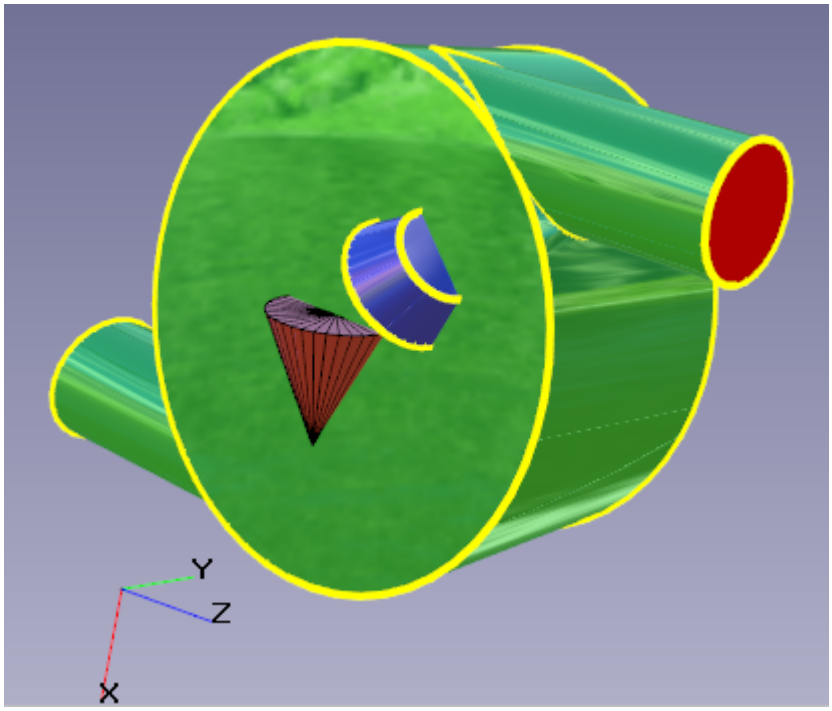


a) b)

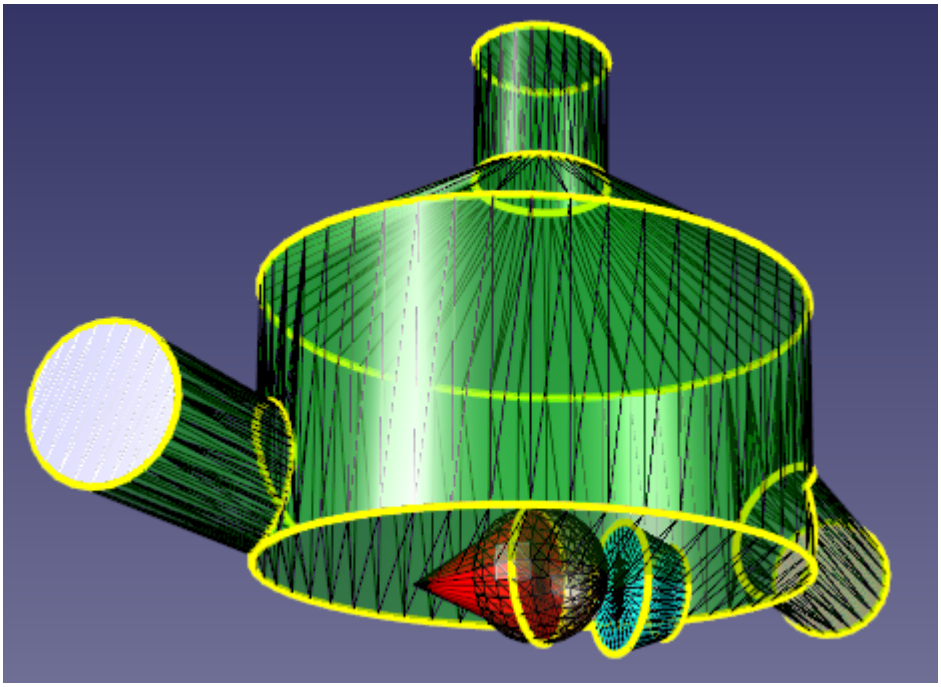
Visualization with enabled (a) and disabled (b) perspective. This is toggled by the button in the **Rendering** toolbar.



Individual visualization settings, which are specified by parameters **Style >...** in the **Properties** window, global settings



Example of individual visualization setting. The default visualization of facet groups (which has been enabled by the button in the **Solids** toolbar) is individually disabled for the purple **Moving body** but not disabled for the blue **Moving body**. Displaying of the wireframe is enabled individually for the purple **Moving body**. Displaying of the fill, wireframe and outlines of facet groups is set by parameters **Style >...**.



Transparency can also be set individually for different surfaces

6.8.6 Transformation of geometric model of the computational domain and an imported object

Transformation (conversion) of the geometry model includes operations of *translation*, *rotation*, and *scaling* the geometry model of the computational domain relating to the absolute coordinate system (ACS) or an **Imported object** relating to its the local coordinate system (LCS-O).

Transformation of geometry is available for **Region**, **Subregions** and **Moving bodies**.

Particular cases of the scaling are:
- mirror reflection
- mirror axial symmetry
- central symmetry

(see description of the **Scaling** parameter and illustrations below).

The **Geometry transformation** dialog box is opened from the context menu of the project tree's element corresponding to the object, which is to be transformed, by the following menu command:
- **Transform geometry** or **Transform**
- *or* **Transform geometry + moving bodies** - in this case, the transformation is also applied to the **Moving bodies**, which exist in the **Object**. This command might be absent in the context menu.

The transformation is carried out for the following elements of the geometry model:

Element of the geometry model	Element(s) in the project tree	Availability of the a command "Transform geometry + moving bodies"
The whole computational domain	**Region**	**Yes**
One of **Subregions**	**Region > Subregions > SubRegion #N** **Region > Geometry > Complex surface #N > SubRegion #N**	**Yes**
One of surfaces of a **Subregion**	**Region > Subregions > SubRegion #N > Geometry > Region- Surface #N**	**No**
Imported object	**Region > Objects > Imported object #N**	**No**
Moving body[*)]	**Region > Objects > Imported object #N >Moving Body #N** **Region > Subregions > SubRegion #N > Modifiers > Moving Body #N**	**No**

[*)] When a **Moving body** is transformed, the same transformation is also applied to the **Imported object**, on which the **Moving body** is built.

Transformation of geometry model of the computational domain is done in the absolute coordinate system and it allows you to change the location of the geometry model of the computational domain relatively the absolute coordinate system. This, in particular, allows you to change the position of the absolute coordinate system within the computational domain (for example, to shift its origin within the computational domain or turn the computational domain in the absolute coordinate system). Transformation of the geometry model of the computational domain can be done including or not including surfaces of moving bodies.

You can perform many operations with the geometry model of an **Imported object** in the local coordinate system. The local coordinate system is not considered as being rigidly connected with the object (unlike local coordinate systems of standard final-volume geometric objects).

At the import of the object, the origin of the local coordinate system locates there, where it has been specified in the CAD-system, which can be inconvenient for work in *FlowVision.* So, for your convenience, you can specify, in the group of parameters **Transformation pivot**, coordinates of the *center of operations*, relating which **Scaling** and **Rotation** will be done.

Transformation of geometry includes the following transformations of the surface, which are done in the absolute or local coordinate system (which coordinate system will be used, depends on the transformed object):
- **Scaling**
- **Rotation**
- **Translation**

Operations of **Rotation** and **Scaling** are performed relatively to the specified *center of operations* (the **Transformation pivot** group of parameters). By default, the center of operations locates at the origin of the coordinate system, but you can move it to another location.

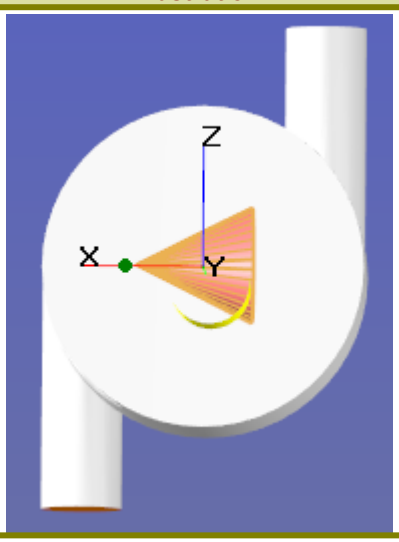
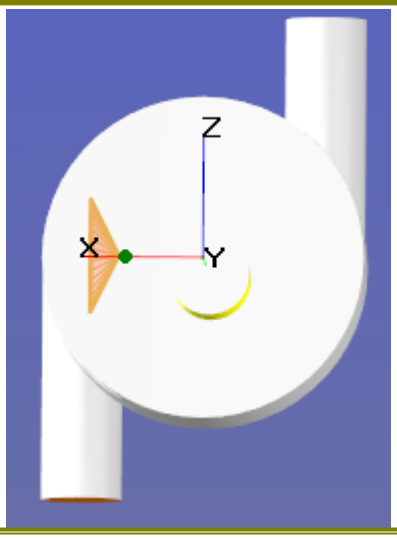
Scaling of an object is done by multiplying on the specified factor coordinates of facets' nodes relatively to the center of operations (the **Transformation pivot**), which can differ from the coordinate system's origin.

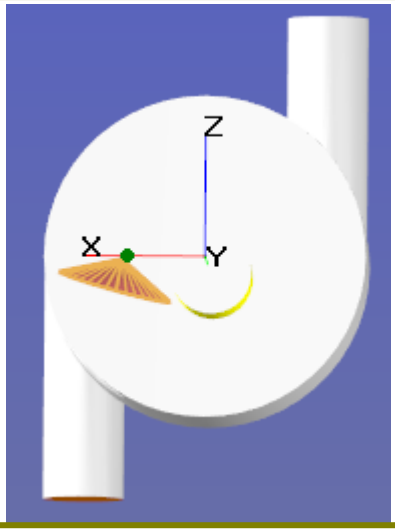
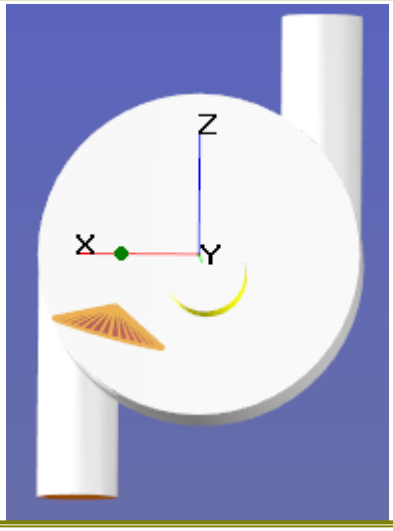
Rotation is performed around an axis, which is codirectional with an axis of the coordinate system or a custom axis specified by the user. The axis of rotation goes through the center of operations, which can differ from the local coordinate system's origin.

Translation allows you to shift the object at the specified distance in custom direction, which are specified by components along axes of the coordinate system.

Example of the geometry transformation

An example of transformation of geometry is presented below:

Application of the transformation to the cone	
Action	**Illustration**
The initial position and shape of the cone. In next rows of the table, you will see steps of the transformation, which includes successive operations of: • **Scaling** (with a negative coefficient, which causes inverting of the cone's orientation) • **Rotation** relatively the center of operations, which is indicated by a green point • **Translation** that, in our example, is a shift opposite to the axis Z	
The **Scaling** is done along the coordinate axes, relatively to the center of operations. The scaling is done with a negative factor, so it inverts orientation of the cone. The absolute magnitude (modulus) of the scale factor is less than 1, so the height of the cone decreases.	

Application of the transformation to the cone	
Action	**Illustration**
The **Rotation** is done around the axis of rotation, which goes through the center of operations.	
The **Translation** is done as a shift in the direction opposite to the axis Z.	

Important note: geometry transformation on Solver

Please note:

If **Update > Type = Disabled** is set in the properties of a **Moving body**, the geometry transformation on **Solver** does not happen.

Therefore, immediately after the geometry transformation of a **Moving body** (which update is disabled), you have to carry out at least one iteration of the calculation with enabled update of the **Moving body**.

If the update of the **Moving body** is disabled, enable it (by specifying, for example, **Update > Type = Automatic**) and, after connecting to **Solver**, carry out the computation until the computational grid rebuilds. Then you can again disable the update of the transformed **Moving body** by specifying **Update > Type = Disabled**.

See also:

Description of the user interface for specifying parameters of the geometry transformation and step-by-step procedures, see in the section Transformation of geometry model of the computational domain (and moving bodies).

6.8.7 Geometry replacement

An important feature of *FlowVision* is its possibility to replace the geometry of a selected element of the geometry model by another geometry *(geometry replacement)*. The geometry replacement is possible for objects **Region**, **Imported object** and **Moving body**.

Geometry replacement can be useful because of:

- change of the geometry model does not requires creating a new project, this saves user's efforts
- changing the geometry makes it possible to run the computation not from scratch but use existing computational results obtained for another geometry, which can reduce the time computation required for the new geometry

Please note:

Replacement of geometry of **Region** makes the whole project be calculated from the scratch (even if continuation of the computation has been selected).

When the geometry is replaced for an **Imported object**, on which a **Moving body** is defined, or just for a **Moving body**, the computation can be run for continuation, but, in this case, it is necessary to enable updating in the properties of the **Moving body** (it is set by the **Update > ...** parameters), or else the new geometry would not replace the old geometry on **Solver**. If **Update > Type > Disabled** is set in the properties of the **Moving body**, then the geometry will not be replaced on **Solver**.

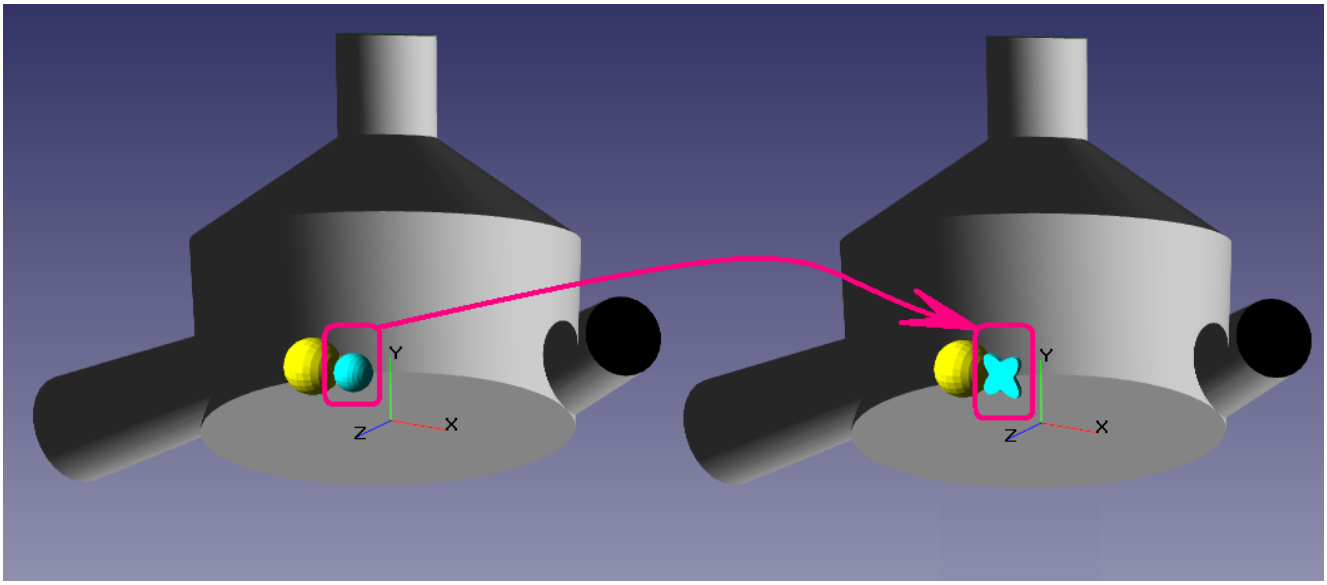
Therefore, immediately after changing the geometry of a **Moving body**, which update is disabled, you have to make at least a single update, that is, to make at least one iteration of the computation with enabled update of the **Moving body**. If the update of the **Moving body** has been disabled, enable it before the geometry replacement (by specifying, for example, **Update > Type = Auto**) and then, after connection to **Solver** and making the geometry replacement, do the computation until the computational grid is rebuilt. Then you can again disable the update of the just modified **Moving body** by setting **Update > Type > Disabled**.

When the computation is started from scratch, you do not have to enable updates of the **Moving body** after the geometry replacement.

When FSI computations are carried out with geometry replacement, updates of **Moving bodies** must be enabled.

Why sometimes it is desirable to disable updates of a Moving Body:

The computational grid is rebuilt at each update of a **Moving body**. This process takes a certain CPU time. Therefore, in simulations where the **Moving body** does not move, it is recommended to disable the updates. If the project has several **Moving bodies**, then update of one of them will cause update of all other **Moving bodies**.



Example of the geometry replacement

See also: section Replacing a geometry model of the computational domain in the project.

6.8.8 Exporting a geometric model into a file

A geometry model can be saved (exported) into a file.

The program supports export to files in the following formats:

- *WRML* (**`*.wrl`**)
- *VTK* (**`*.vtk`**)
- *3DVision* (**`*.mesh`**)

See also:

Step-by-step procedures for export a geometry model into a file see in the section Exporting a geometric model into a file.

6.8.9 Geometric objects

FlowVision uses *geometric objects*, surfaces of which are not parts of the geometry model of the computational domain. In other words, those geometric objects are not perceived by the medium as material bodies, that is, objects do not influence the flow and other physical processes as real bodies; they do not influence the flow and other processes solely by the fact of their existence.

Geometric objects can be classified as:

- *standard geometric objects* created by means of *FlowVision*:
  - **Line**
  - **Plane**
  - **Box**
  - **Cone/cylinder** (which can also be a truncated cone or a cone or a cylinder with an internal channel and a sectoral cutout)
  - **Ellipsoid/sphere** (which can also be a Platonic solid with triangle facets)
- **Imported objects** are geometric objects created in an external CAD system and imported into *FlowVision* similarly to importing a geometry model of the computational domain. Unlike a geometry model of the computational domain, they may be open surfaces. They can also be obtained as copies of standard geometric objects.
- **Supergroups**
  - **Groups** of facets
  - the **Computational space** object
- **Sets of sensors**

The principal purpose of the objects is to define a certain surface or volume in the computational domain within which other project elements will be acting or will be defined.

Geometric objects are used for setting volumes or surfaces, within/on which will be performed:

- computational grid refinement
- setting values of variables
- setting initial conditions
- computing the set of variables, averaged over a surface or volume
- displaying a visualization layer

Geometric objects are displayed in two tabs of the project tree: **Preprocessor** and **Postprocessor**, see section Objects in project tree.

Direction of the normal to a surface. Specifics of calculation of integral variables over surfaces of geometric objects.

Geometric objects can be used to define an integration surface required to calculate **Characteristics**. For many integral variables (flows), direction of the normal to the object's surface is an important parameter (see details in the section The Info window for Characteristics).

FlowVision uses the following rules to define direction of the normal to a surface:

- normals of standard geometric objects (**Cone/cylinder**, **Ellipsoid/sphere**, **Box**) are directed inside the **Objects**
- normals of imported **Objects** are directed according to the settings that were made at creation of these **Objects** in a CAD system

- normal of a **Plane** is defined at creation of the **Plane**

6.8.9.1 Standard geometric objects

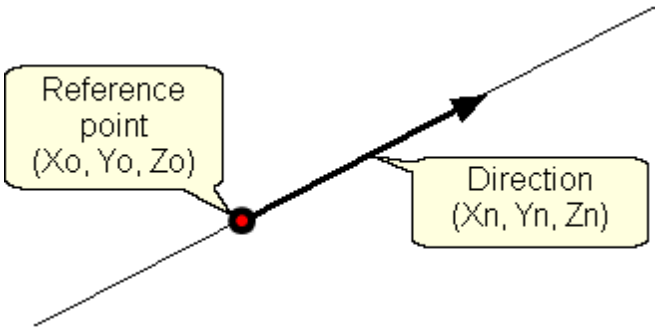
In next sections the following standard geometric objects are described:

- **Line**
- **Plane**
- **Box**
- **Cone/cylinder**
- **Ellipsoid/sphere**

6.8.9.1.1 Object «Line»

A **Line** object is defined by the following parameters:

- a point, through which the **Line** goes
- a directing vector of the **Line**



A **Line** is defined by its reference point and direction

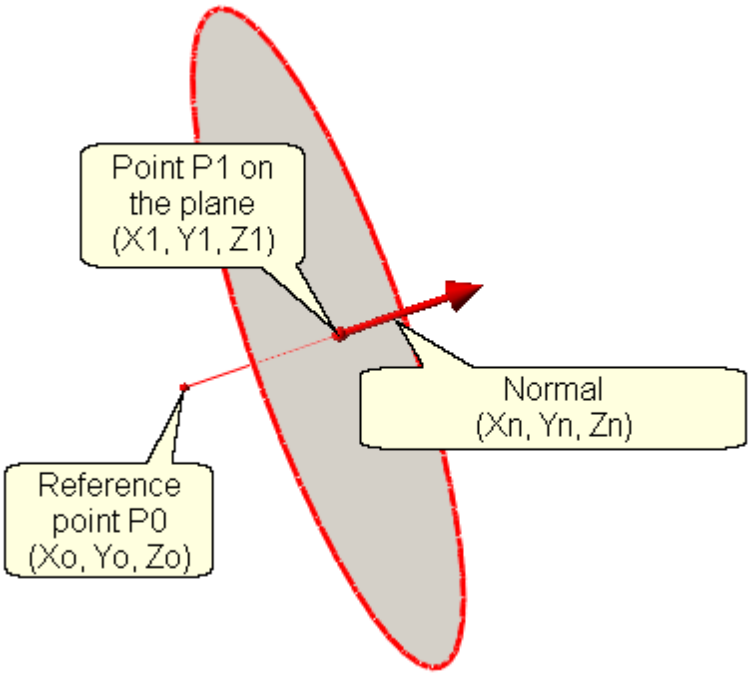
See also:

Description of the user interface, where you can specify these parameters, in the section Object «Line» (user interface).

6.8.9.1.2 Object «Plane»

A **Plane** object is defined by the following parameters:

- the reference point P0 of this **Plane**
- direction of the normal to the **Plane**
- shift of the **Plane** relatively the reference point P0 (this shift is done in the direction of the **Plane**'s normal and moves the point P0 to point P1, which lies in the **Plane**)



A **Plane** is defined by its reference point, by direction of its normal and by the **Shift**. When **Shift** is non-zero, the **Reference point** doesn't lie in the **Plane**.

A **Plane** splits the space into two halves:

- *positive semi-space* (to which the **Plane**'s normal is directed)
- *negative semi-space*

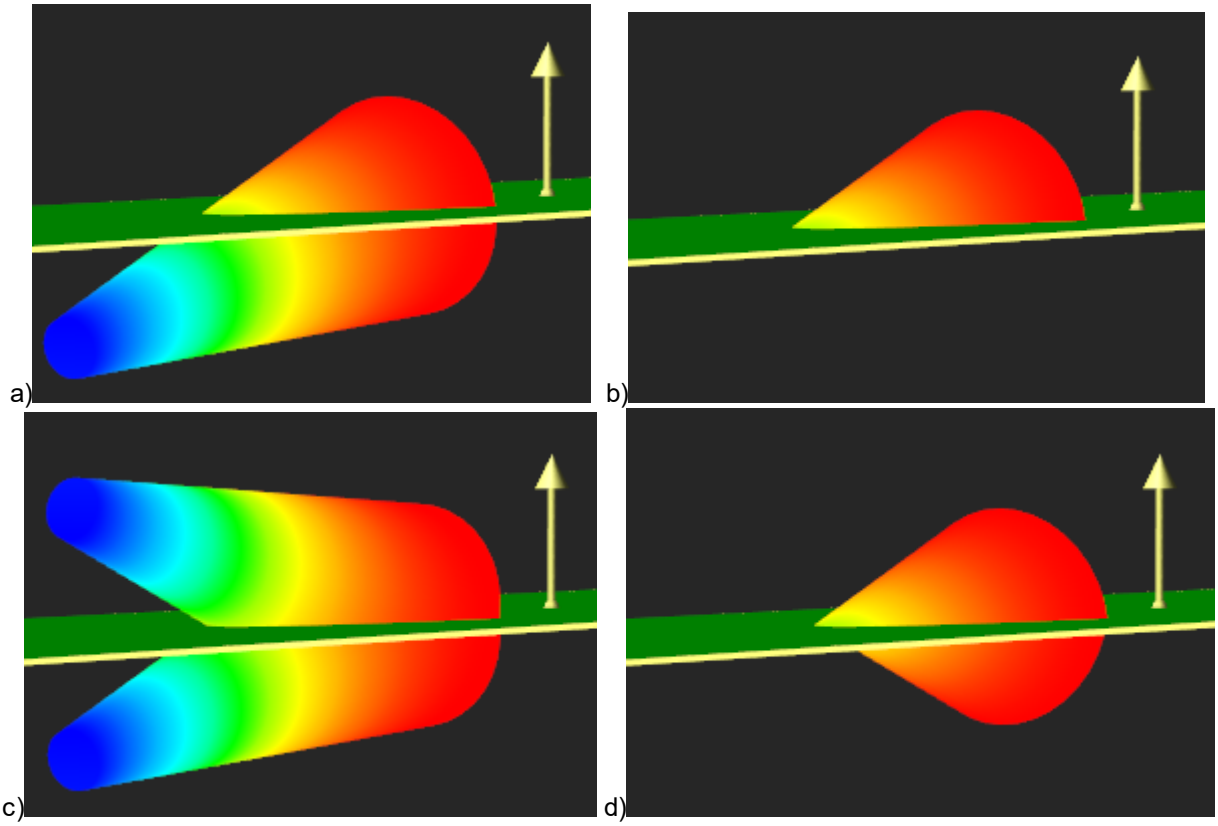
Parameters **Clipping object** and **Mirror** affect the whole visualization, which is displayed in the **View** window:

- A *clipping* plane cuts images of **Objects** and **Layers**, for which the **Clipped=Yes** is set in their properties. When the clipping is applied, those parts of are displayed only that locate in in the positive semi-space.
- A *mirror* plane reflects layers within the computational domain (from the positive semi-space to the negative semi-space, and vice versa, i.e., a mirror **Plane** reflects like a double-sided mirror). The reflected layers are displayed in those order in which the mirror **Planes** are specified in **Postprocessor**.

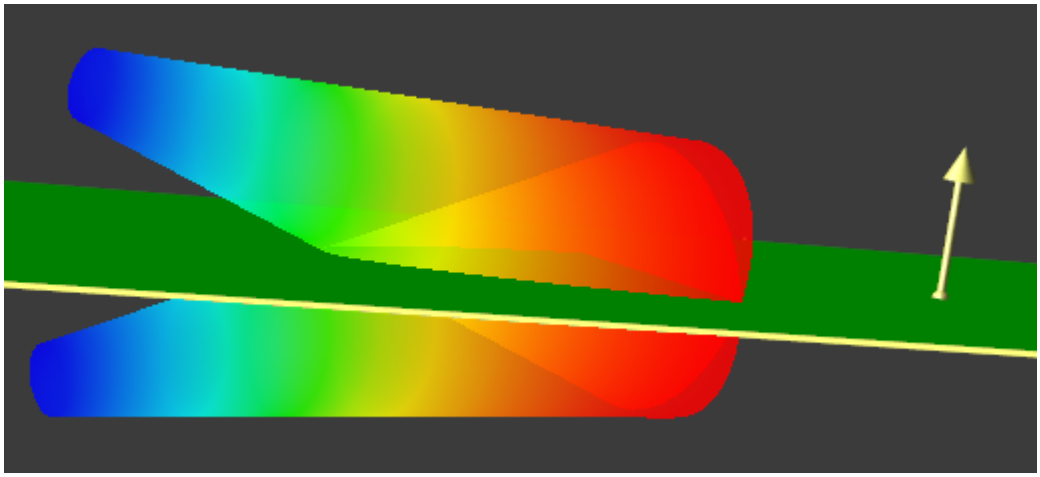
Because of limitations of *OpenGL*, it is *not* recommended to specify:

- more then six clipping **Planes** in a project
- more then three mirror **Planes** in a project

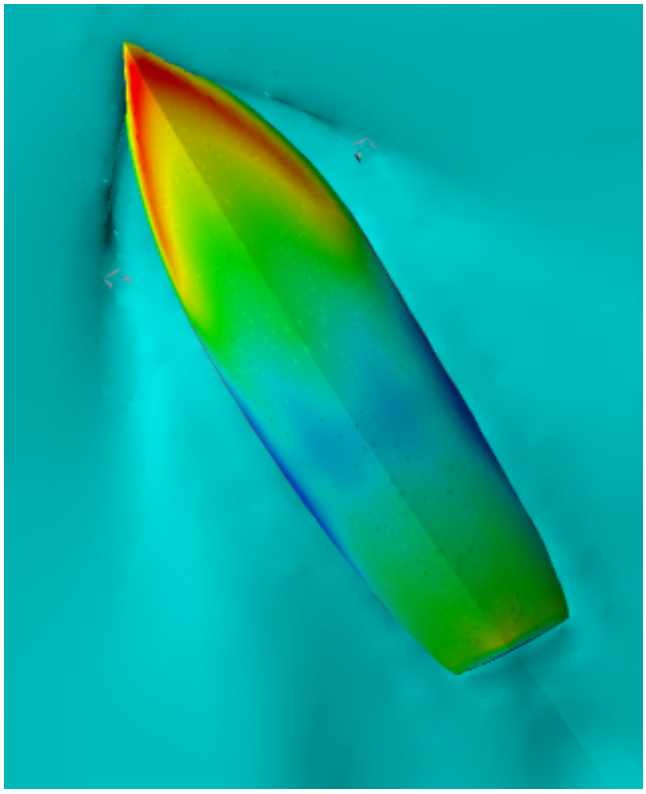
The **Shift** parameter is useful when the **Plane**'s normal is specified and you just need to displace the **Plane** into a new position by a parallel shift with now displacement of the **Reference point** from the specified **Normal**. Instead of calculating a new position of the **Reference point** with taking into account the **Normal**'s inclination, you can just specify the **Shift**.



Examples of use of the **Plane**'s properties **Clipping object** and **Mirror**; the illustration shows their influence on the visualization of a **Color contours** layer, which has been built on a **Cone**: a) **Clipping object = No**, **Mirror=No**; b) **Clipping object = Yes**, **Mirror=No**; c) **Clipping object = No**, **Mirror=Yes**; d) **Clipping object = Yes**, **Mirror=Yes**.



When the **Mirror** property is enabled, the **Plane** reflects **Layers** as a double-sided mirror. To show this effect, semitransparency has been enabled for the reflected **Layer**.



Mirror **Planes** are useful to visualize simulations, which are done in symmetry settings (for example, movement of a boat)

See also:

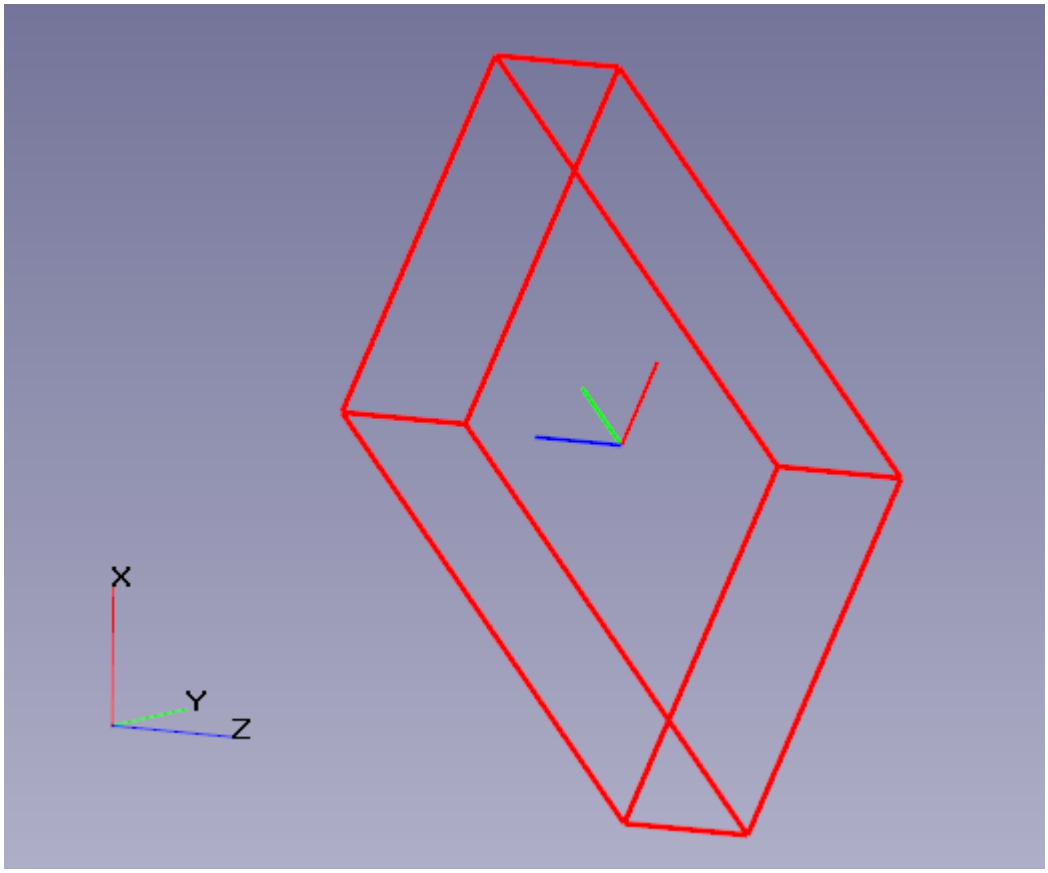
Description of the user interface for specifying parameters of a **Plane** in the section Object «Plane» (user interface).

6.8.9.1.3 Object «Box»

A standard geometrical object **Box** is defined by the following parameters:

- a point located in its center
- lengths of its edges
- orientation of its local coordinate system (LCS-O, objec's local coordinate system), the unit vectors of which are perpendicular to the faces of the **Box**. The origin of the LSC-O locates in the center of the **Box**.

Normals to the faces of the **Box** are directed inside.



A **Box** with its local coordinate system located in its center

See also:

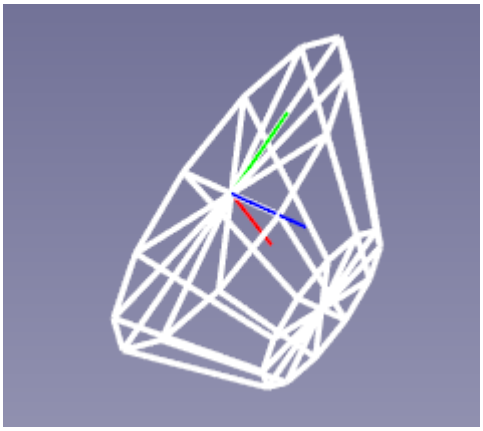
Description of the user interface for specifying parameters of a **Box** in the section Object «Box» (user interface).

6.8.9.1.4 Object «Cone/cylinder»

A standard geometrical object **Cone/cylinder** can be an approximation of the following geometric solids:

- cone
- frustum of cone
- cylinder
- any of the listed above solids with a central channel
- a 3-dimensional sector cut out from any of the listed above solids along the axis connecting the centers of the solid's bases or connecting the center of the solid's single base and the solid's vertex

A **Cone/cylinder** object has an associated local coordinate system (LCS-O), which origin locates in the center of a base, the X-axis is directed along the object's axis, and axes Y and Z lie in the plane of the object's base.



A **Cone/cylinder** with its local coordinate system (with its origin in the center of the **Cone/cylinder**'s base)

Bases of a **Cone/cylinder** are generally coaxial proportional ellipses that lie in parallel planes. In a particular case, bases of a **Cone/cylinder** can be circles.

A **Cone/cylinder** object can have an inner channel having the shape of a cone/cylinder coaxial with the **Object**. The channel has elliptical bases with semiaxes that are proportional to the semiaxes of the **Object** (i.e. the ratio r_1/r_2 of semiaxes of the channel's bases is equal to the ratio R_1/R_2 of semiaxes of the **Object**'s bases).

The line connecting the centers of the bases of the **Cone/cylinder** (or the center of a single base and the vertex of a non-truncated cone) may be *not perpendicular to* the bases (base).

Also a **Cone/cylinder** object can be a *three-dimensional sector*, which has been cut from the **Cone/cylinder**.

Normals to the surface of the **Object** are directed inside.

See also:

Description of the user interface for specifying parameters of an **Cone/cylinder** in the section Object «Cone/cylinder»(user interface).

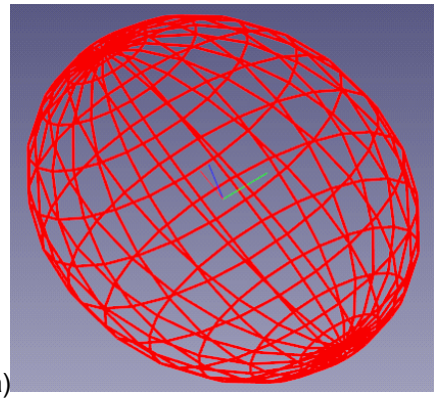
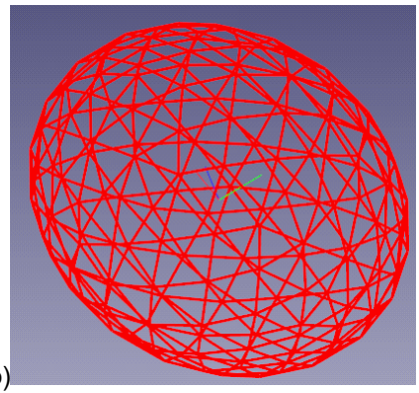
6.8.9.1.5 Object «Ellipsoid/sphere»

A standard geometrical object **Ellipsoid/sphere** is defined by the following parameters:

- its reference point (the center)
- orientation, i.e. direction of axes of its local coordinate system (LCS-O), the origin of which locates in the center of the **Object**
- radius R of the sphere or three values of length of semiaxes of the ellipsoid R_1, R_2, R_3 (radii of the ellipsoid), which specify the distances from the center to the surface of the **Object** along the axes of the local coordinate system (**X, Y** and **Z**, respectively)
- scale ratio, which can be used to change the size of the **Object**
- the method, which has been used to approximate the surface of the **Ellipsoid/sphere** by a polyhedron (see illustration below)

Normals to the surface of the **Object** are directed inside.

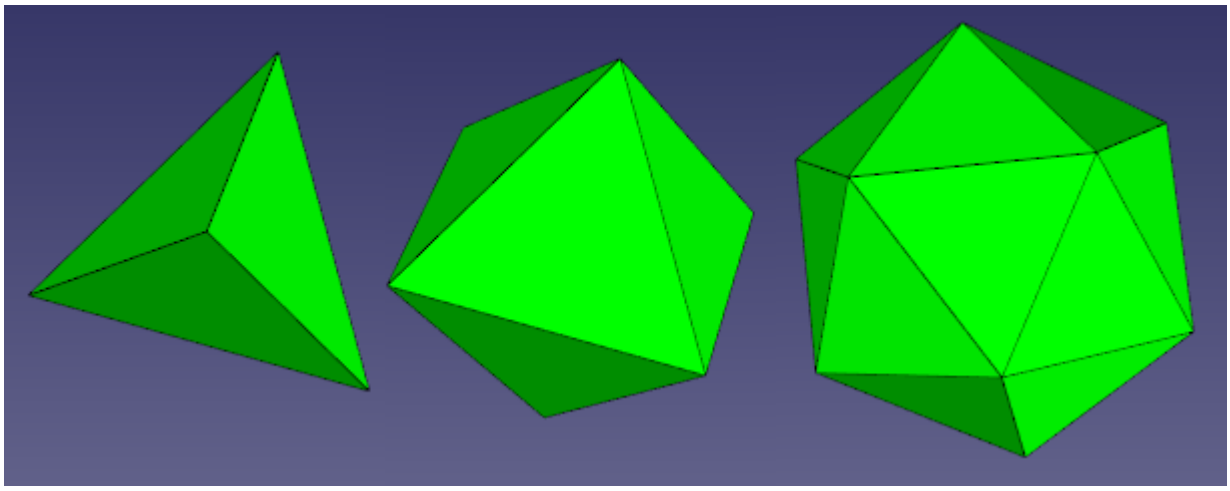
When you create an **Ellipsoid/sphere**, you specify the method of its approximation by polar or geodesic approximation.

a)  b) 

Approximation of an **Ellipsoid/sphere** by facets: a) polar; b) geodesic

When the geodesic approximation is used and the number of iterations is zero, then the **Sphere** is approximated by a Platonic solid with triangle facets:

- **Tetrahedron**
- **Octahedron**
- **Icosahedron**



When number of iterations is zero, the geodesic approximation of a **Sphere** is a regular tetrahedron, octahedron, or icosahedron

See also:

Description of the user interface for specifying parameters of an **Ellipsoid/sphere** in the section Object «Ellipsoid/sphere» (user interface).

6.8.9.2 Imported objects

Imported objects are used to:

- specify area of action of **Adaptations**, **Adaptations by condition** and **Adaptations to solution**
- specify visualization **Layers**
- create **Moving bodies**

An **Imported object** is added to the project in the following ways:

- It is imported from an object created in an external CAD system. Besides, the following data is imported from the CAD system:
  - the coordinate system (CS) where the object was defined (in *FlowVision* that CS becomes the objects' local coordinate system, LCS)
  - coordinates of facet nodes, data on edges and sides of facets
  - facet color data (when importing the object in formats *VRML* and *3D TransVidia Mesh*)

- It is formed based on a standard geometric object of finite volume (**Box**, **Cone/cylinder**, **Ellipsoid/sphere**). To do this, use the command **Copy as imported object** from the context menu of the standard geometric object.

If the **Imported object** has been created in a CAD system, it must be a closed or non-closed surface *without self-intersections*. During it is being imported from a CAD system, the object's facets are grouped.

An **Imported object** consisting of several unconnected surfaces can be split into separate Imported objects. An **Imported object**, which is limited by a close surface, can be used as a base for creation a **Moving body** on it (if no **Movement** is set on this **Imported object**).

A special case of **Imported objects** are objects **Self-intersections #N**, which are generated at checking the geometry for self-intersections.
Objects **Self-intersections #N** are presented in the **Postprocessor** tab only.
See section Self-intersections of surfaces and their correction.

See also: Imported objects (user interface).

6.8.9.3 Object «Supergroup»

The **Supergroup** object is formed from groups of facets according to the needs of the user, for example, it forms from facet groups, on which a certain boundary condition is set.

Using a **Supergroup**, you can apply operations to parts of of several geometry **Objects** simultaneously.

A group of facets *can* belong simultaneously to different **Supergroups**.

Even if the boundary conditions are been changed on some groups of the **Supergroup**, the composition of the **Supergroup** will not change, so groups of facets included in a **Supergroup** still will form the same single **Supergroup**.

During geometry transformation, the structure of a **Supergroup** may be violated. In this case, it will be necessary to delete this **Supergroup** with violated structure and create a new **Supergroup**.

Examples of use of **Supergroups**:

- they can be used to select surfaces where control parameters are computed (for example, flow through certain inlets/outlets)
- calculated characteristics can be displayed on surfaces of a **Supergroup** (for example, distribution of pressure over the surfaces of a streamlined body)

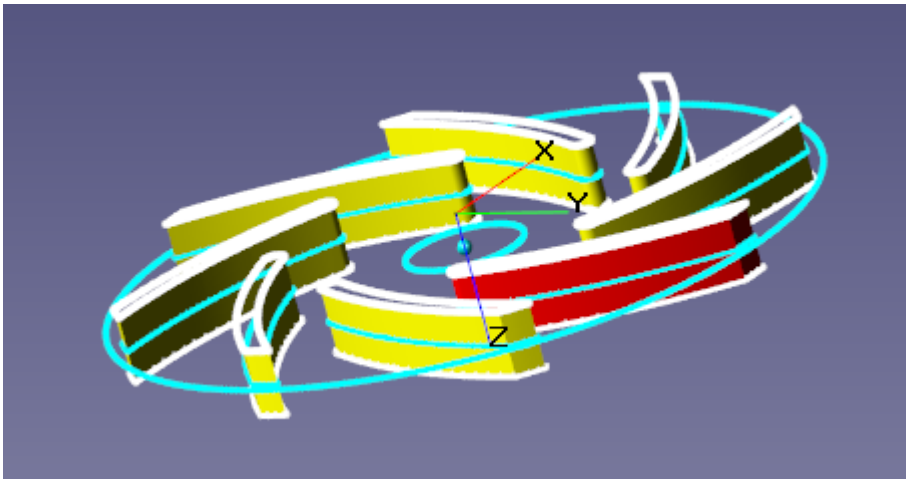
Groups of facets can be combined into a **Supergroup** in *FlowVision*, if they share a common boundary condition. In other words, a **Supergroup** built from a group of facets at the moment of creation satisfies the following criteria:

- a **Supergroup** completely includes one or several groups of facets
- a common boundary condition is set for all groups of facets included in a **Supergroup**; this boundary condition is not set for other groups of facets

After creation a **Supergroup**, it can be changed quite arbitrary according to the needs of the user.

Example:

It is possible to create a **Supergroup** including *all the rotor blades* for which a corresponding boundary condition is set (see illustration below). If later another boundary condition is set for one of these blades (shown in red), that blade remains in the **Supergroup** where it was initially included.



Even if the boundary condition for it changes, a group of facets is not deleted from a **Supergroup**

See also: Object «Supergroup» (user interface).

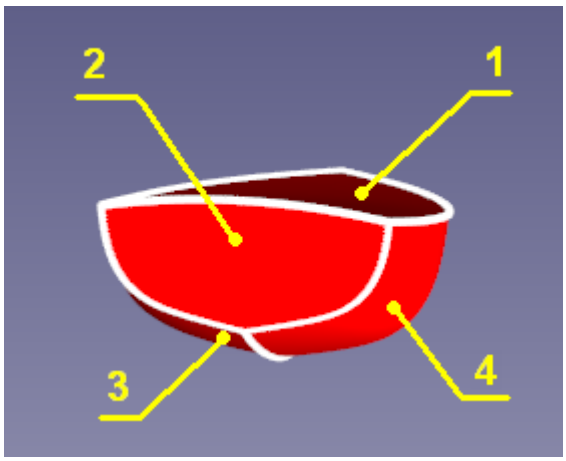
6.8.9.4 Groups of facets as geometric objects

Groups of facets are used to select surfaces and to match the surfaces to, for example, boundary conditions.

Generally, the number of facets in geometry models is quite large, so it is inconvenient for the user to carry out operations with individual facets (for example, setting **Boundary conditions** on them). It is much more convenient to work with *groups* of neighboring facets, which form quite large fragments with different purposes in the model.

Groups of facets are formed during the process of grouping of facets, which is is done:

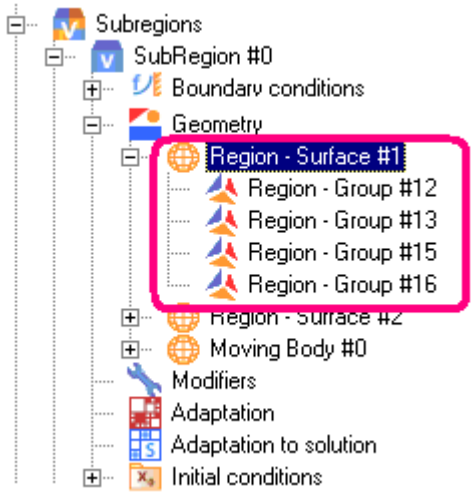
- automatically during import of geometry models
- *or*, if the user wishes, the process can be done again with other parameters



Example of grouping of facets of the geometry models of a boat. The following **Groups** have been formed: 1 - deck, 2 - aft, 3 and 4 - sides.

Presentation of groups of facets in the project tree

Groups of facets are presented in the project tree in the **Preprocessor** tab in elements **Subregions > SubRegion #N>Geometry > Region- Surface #N > Region- Group #N**:



Depending on the value of the **Display > Show all groups** parameter in the basic settings of **Pre-Postprocessor**, either all **Groups** are presented or that **Group** only, which is selected in the **View** window by mouse clicking with the **Ctrl** key pressed.

Setting boundary conditions on groups of facets

In the project tree or in the **View** it is possible to select only a whole group of facets. So when **Boundary conditions** are set, the same boundary condition is assigned to all the facets of the selected group.

If it is necessary to set different boundary conditions to different facets within the same group, then do the following:

- first regroup (split) the group into the required number of groups
- then assign the required boundary conditions to the resulting groups of facets

Boundary conditions as such are created as a separate element of the project and can be assigned to different sets of facets.

Forming a Supergroup

Groups of facets that have been selected according to the user's choice (for example, those, which have the same **Boundary condition** on them) can be included into a **Supergroup**, which is can be useful to analyze results of the computation or to set adaptations of the computational grid on the selected surface.

See also:

- Grouping the facets
- Regrouping the facets
- Procedure of regrouping a geometric model of computational domain (and moving bodies)
- Folder «SubRegion #N > Geometry»

6.8.9.5 Object «Computational space»

Often, when you need to create a visualization **Layer** or specify an **Adaptation** (or an **Adaptation by condition**, or an **Adaptation to solution**), you might wish to specify them in the entire computational domain.

Also the **Computational space** object may be used to set **Modifiers**, **Characteristics** and **Initial conditions** within the entire volume of a **Subregion**.

Specifics of use the *Computational space* object

Note that **Modifiers** and **Initial conditions** are always related to some specific **Subregion** and act only there.

Even if the **Computational space** object is specified as the area of action of a **Modifier** or **Initial conditions**, they will act only in the set intersection of geometric volumes of the **Computational space** and the appropriate **Subregion**.

For example, if you wish to specify the same **Initial conditions** in several **Subregions** in the entire **Computational space**, you have to specify them separately in each of the **Subregions**.

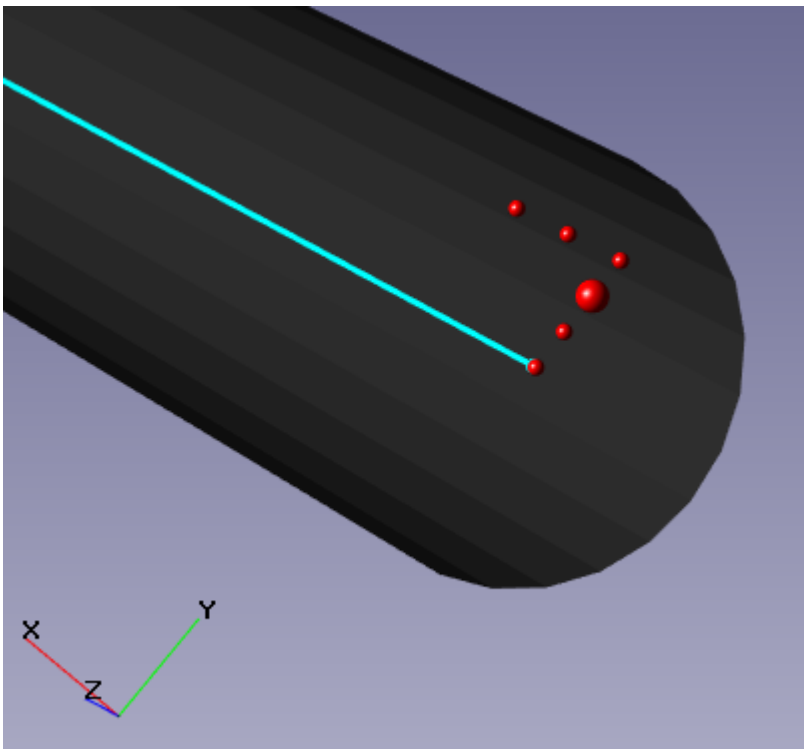
See also descriptions of user interface:

- Element «Computational space» in the Preprocessor tab
- Element «Computational space» in the Postprocessor tab

6.8.9.6 Object «Set of sensors»

Set of sensors is a zero-volume geometrical object, which is a set of mathematical points.

Sets of sensors are useful to calculate **Characteristics** on desired points in the computational domain.



Example of **Set of sensors** consisting of 6 points. The selected **Sensor** is displayed bigger then others.

You can save a **Set of sensors** in a file for future use.

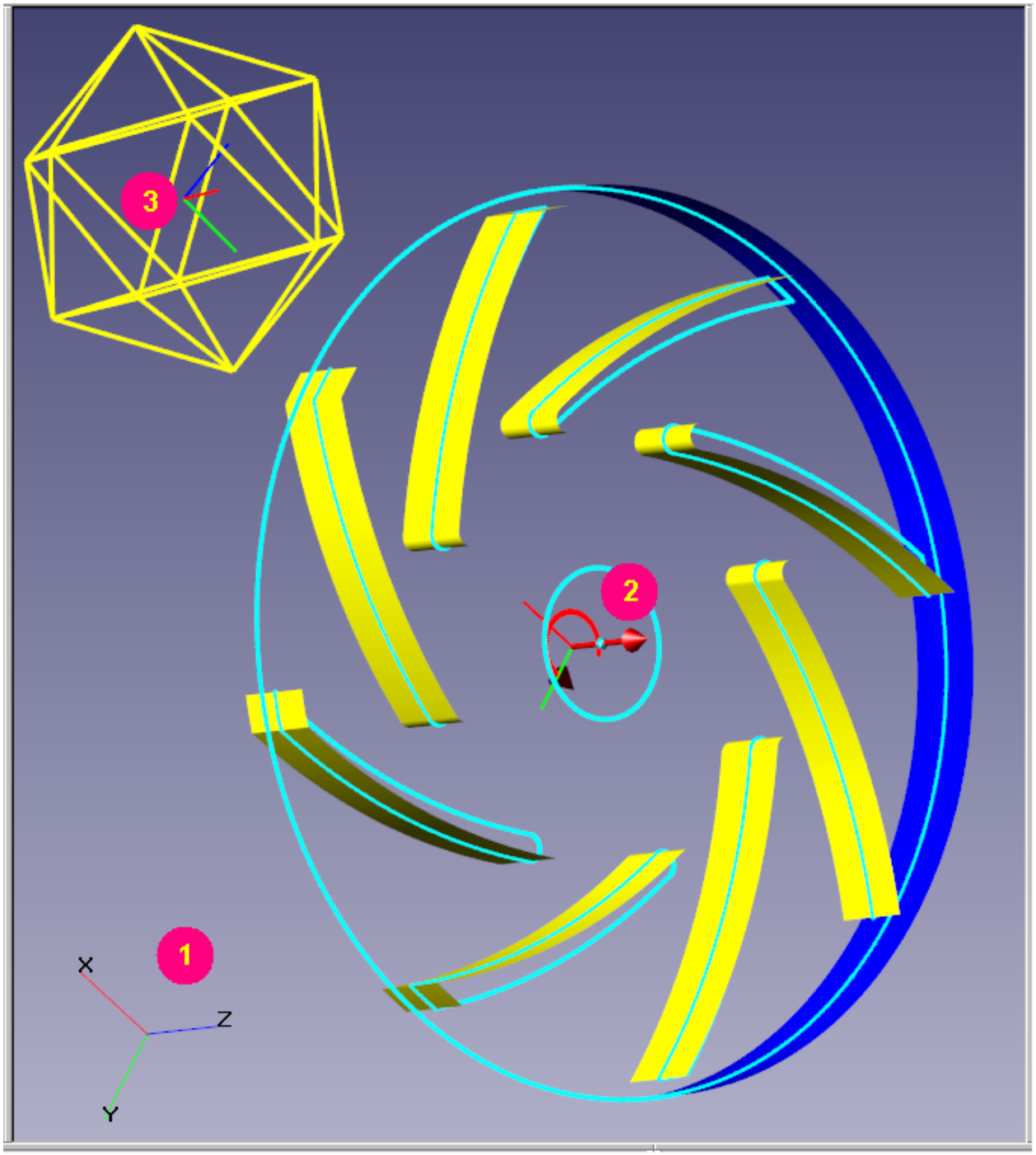
All or any **Sensors** from a **Set** can be bound (linked, connected) either to the LCS (local coordinate system) of the **Set of sensors** or to facets of a geometry model (for example, to facets of a **Moving body** or a **Boundary condition**). Respectively the **Sensors** might move either at changes of the LCS or changes (movements, rotations, scaling) of the geometry model.

See also: Object «Set of sensors» (user interface).

6.8.10 Coordinate systems

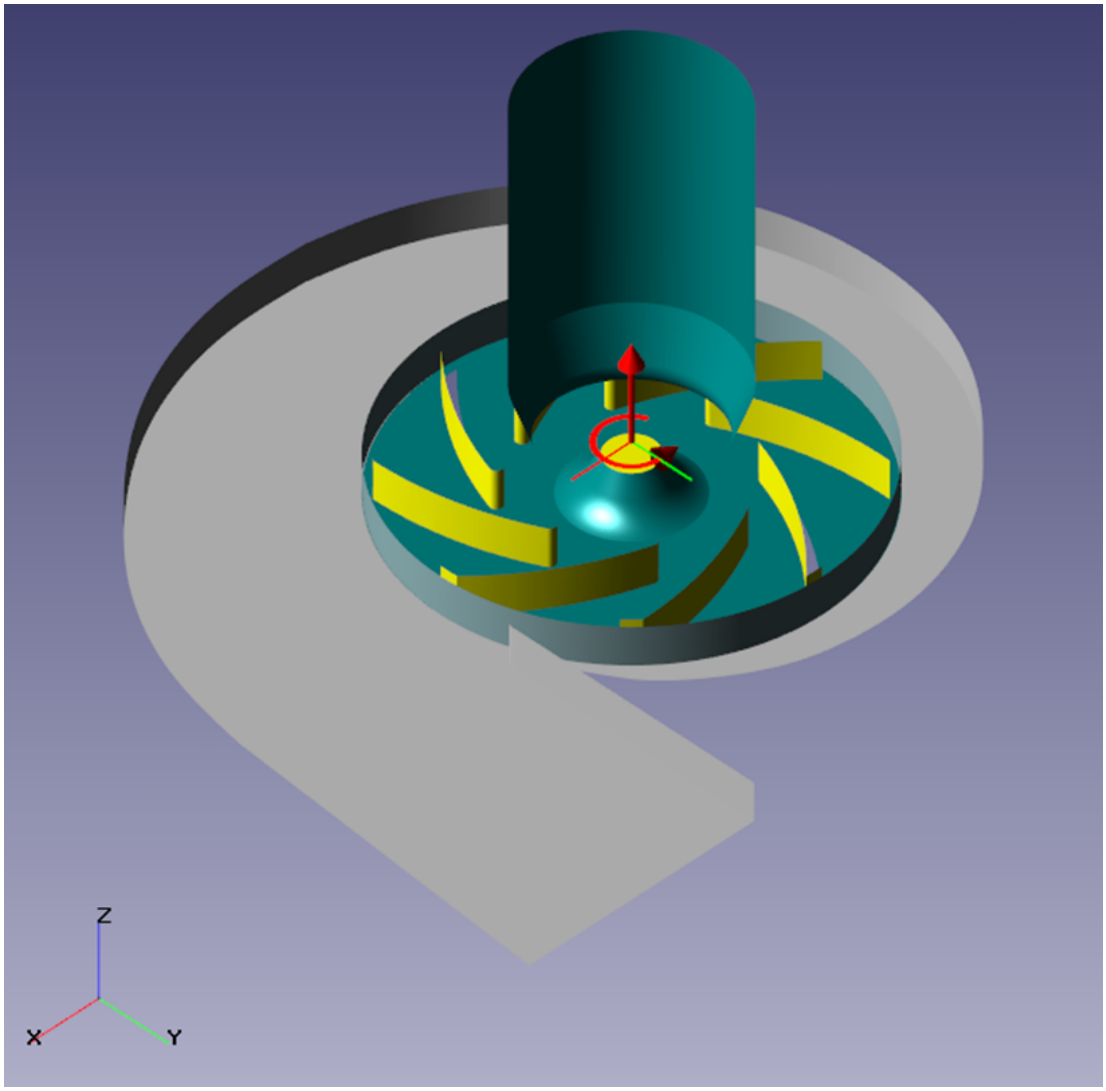
FlowVision uses the **Absolute** and **Local** coordinate systems (CS):

- Absolute coordinate system (ACS) is the coordinate system of the geometry model of the **Region**. By default, ACS is the operational coordinate system for specifying input data and for visualization. ACS is imported to the *FlowVision*'s project along with the geometry model.
- Local coordinate systems (LCS) can be:
  - a Movement LCS (LCS-M) that is created by the user in the *FlowVision*'s project and stores position and orientation of the coordinate system, in which rotation and translation movement relating ACS or another LCS-M (for nested coordinate systems). LCS-M are used to specify motion parameters of a subregion, a surface of a boundary condition, a geometry object, and also for visualization.
  - a Geometric object LCS (LCS-O) that defines location and orientation of a geometric object relating the ACS. A LCS-O is imported along with its geometric object.

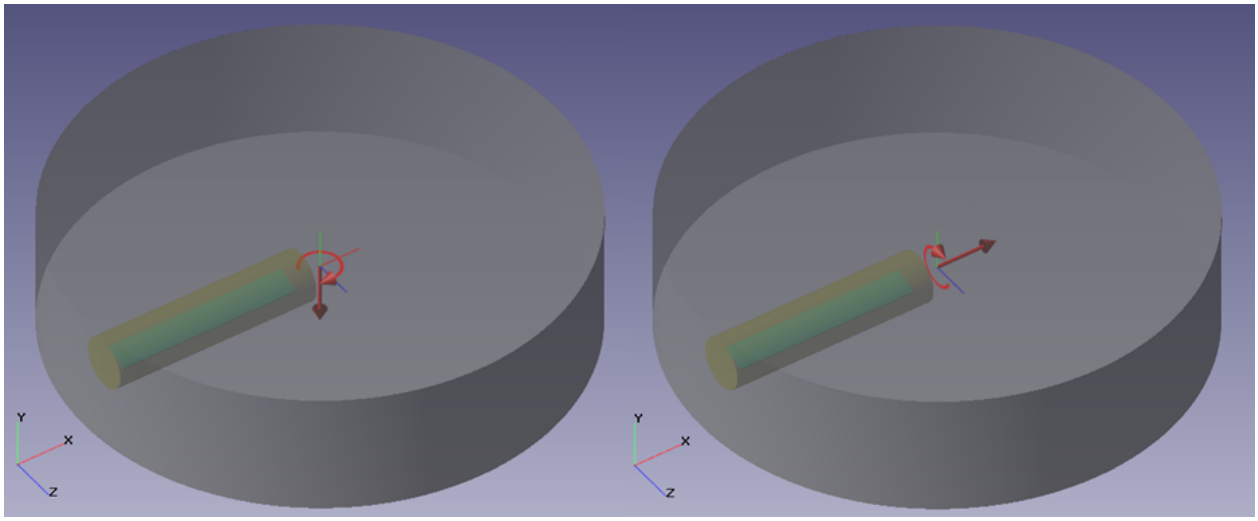


Coordinate systems:

1: ACS; 2: Movement LCS (LCS-M); 3: Geometric object's LCS (LCS-O)



Origin of a LCS locates in the center of the rotor, origin of the ACS locates at the left bottom corner of the illustration



Example of nested rotation of blades relating two axes: rotation of all blades relating local axis Y of the wide cylinder (shown on the left) plus rotation of an individual blade along local axis X of the narrow cylinder (shown on the right)

6.8.10.1 Absolute coordinate systems (ACS)

The *absolute coordinate system* (ACS) is set in an external CAD system during creating the geometry model of the computational domain and then it is imported into the *FlowVision* project along with this geometry model.

FlowVision allows you to change position of the geometry model of the computational domain relating the ACS. It is done by means of *transformation* of geometry model of the computational domain, which is displacement and rotation it relating to the ACS, see sections "Transformation of geometry model of the computational domain and imported object" and "Transformation of geometry model of the computational domain (and moving bodies)". The transformation unlinks the initial position of the geometry model from the ACS.

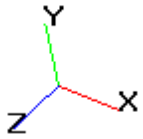
Most of values and parameters in *FlowVision* are specified, calculated, and visualized relatively to the ACS. There are some exclusions:

- if an LCS is set on a **Subregion**, then all parameters are set and displayed in this LCS.
- mass-inertia properties of a **Moving body** are specified in its local coordinate system (LCS-O)
- for visualization layers **Vectors** and **Nodal loadings** it is possible to select visualization in some local movement coordinate system (LCS-M) or specify in the layer's properties its own local moving coordinate system.

By default, ASK is displayed in the bottom left corner of the **View** window as the **Coordinate system** layer. The standard display of the ACS can be changed, see details in the section Layer «Coordinate system», user interface.

Displaying the ACS in the View window

By default, ACS is displayed in the left bottom corner of the **View** window as an image of coordinate axes (this image is the **Coordinate system** layer):



By default, this image displays only directions of coordinate axes of the ACS and is *not linked* to the origin of the ACS.

You can change displaying of the ACS in the **View** window. When you change appropriate properties of the **Coordinate system** layer, you can:

- display the image of the ACS in another corner the **View** window
- link the image of the ACS to the origin of the ACS
- change size of the image of the ACS

See details in the section Layer «Coordinate system», user interface.

6.8.10.2 Movement local coordinate systems (LCS-M)

The movement local coordinate systems (LCS-M) are used to define:

- tangential velocity on surfaces of boundary conditions
- method of calculation for the selected computational **Subregion**
- parameters of motion of geometric objects
- visualization of layers **Vectors** and **Nodal loadings** when **Coordinate System = Local** is set in their properties.

LCS-M is created in the *FlowVision* project by the user and initially it is not linked to any geometric object. Later, an LCS-M can be linked to one or several geometric objects; this is specified in properties of these objects.

If necessary, when the problem setting requires, you can specify several LCS-Ms within one *FlowVision* project.

Each LCS-M (specified by coordinates of its origin and directions of its axes in the absolute coordinate system ACS) is used in the program in combination with one of **Rotations** and one of **Translations** that are selected in the program's user interface from elements, which locates in child subfolders **Rotation** and **Translation** (see description of the user interface in the section Folder «Local coordinate systems»).

Details about how **Rotations** and **Translations** are specified in a LCS-M, see in sections below:

- Rotations in LCS-M
- Translations in LCS-M
- Specifics of use Rotation and Translation

6.8.10.2.1 Rotations in LCS-M

Rotations are used to:

- specify problem setting in a rotating local coordinate system (see the subsection *"Problem settings with rotating local coordinate system"* below)
- specify the rotational speed of a boundary condition (see the subsection *"Specifying a rotation of a Boundary condition"* below)
- create a sliding surface for forming a rotating subregion
- specify rotations of geometric objects (to specify dynamic adaptations, to set variables, to calculate integrated values)
- create layers **Vectors** and **Nodal loadings** in rotating coordinate systems

Each **Rotation** is defined by its *vector of rotational speed*.

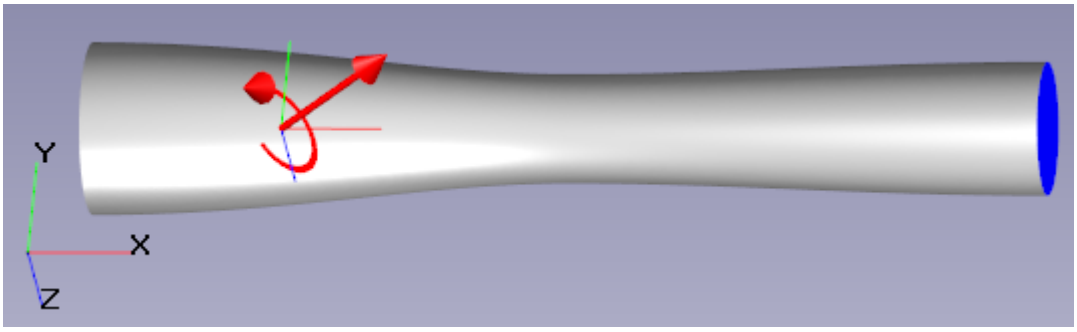
The length of the vector of rotational speed ([rad/s]) can be defined as a:

- constant
- harmonic function
- formula
- table

Also the rotational speed can be calculated with taking into account the moment of inertia and external torques, see Autorotation (changing the rotation speed by the flow).

The direction of the vector of rotational speed is defined relatively to a LCS-M.

A **Rotation** is displayed in the **View** window as two arrows, one of which represents the axis of rotation, and the other arrow shows the direction of rotation if the rotation speed if it is set as a constant and is not zero (see the illustration below).



A **Rotation** element is displayed as two arrows depicting the axis and direction of the rotation

If the length of the rotational speed is defined not as a constant but as any other method, then the visualization of the rotation's direction is displayed figuratively and might not correspond to the actual direction of the rotation.

Problem settings with rotating local coordinate system

In problem settings with rotating surfaces that form some rotating volume (a rotating subregion), it is possible to do simulating in relative rotating coordinate system. This approach allows the program not to rearrange the grid in the rotating subregion and considerably save the computational resources.

Transition to a rotating relative coordinate system is done by specifying an LCS-M and a **Rotation** for:

- a rotating **Subregion** (for problem settings with a sliding grid where the rotating subregion is only a part of the whole computation **Region**)
- *or for* a rotating **Region** (i.e. for the whole rotating subregion).

In the specified computational volume the program enables taking into account forces of inertia (centrifugal and Coriolis forces). Additionally, to take into account the tangential velocity of the rotating surface, you have to specify an LCS-M and a **Rotation** on the **Boundary condition** (see subsection *"Specifying a rotation of a Boundary condition"* below).

See an example of the problem setting with a rotating coordinate system in the *Tutorial: Examples of typical tasks* document, section *"Sector of axial compressor"*.

Specifying a rotation of a Boundary condition

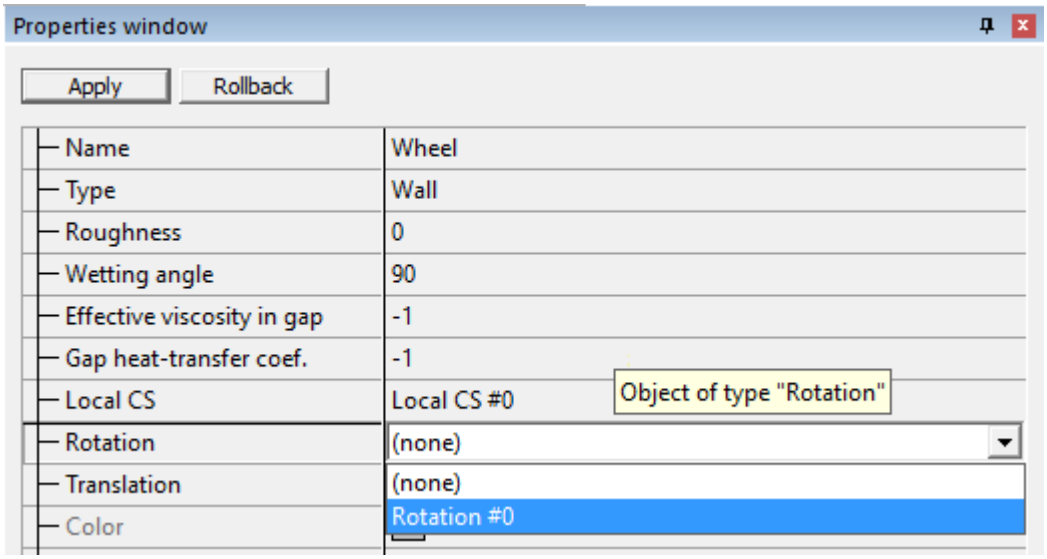
To specify a rotating surface in a problem setting in a rotating coordinate system (see subsection *"Problem settings with rotating local coordinate system"* above) you have to specify an LCS-M and a **Rotation** on the **Boundary condition**. This enables taking into account the tangential velocity on the surface of the **Boundary condition**.

When the rotating surface in the *FlowVision* project is an axially symmetric body and rotates around its axis, the program can take this rotation into account without use of the rotating coordinate system. An example of such problem setting is rotation of a car's wheel (if the wheel disk is also presented as an axially symmetrical body or just neglect effects of the disk). The rotating boundary condition **Wall**, which is set on a wheel, unlike other walls, will have a non-zero velocity of the surface (the tangential velocity of this surface is $\omega \cdot R$, where ω is the rotational speed and R is the radius, and the radial velocity of the surface is zero).

Another example of use a **Rotation** on a **Boundary condition** – specifying it on a boundary condition **Inlet** to simulate a rotating flow.

To specify a rotation of a **Boundary condition**, you have to select from drop-down lists in properties of the **Boundary condition** values of the following parameters:

- **Local CS** (select one of existing LCS-Ms presented in the project tree by folders **Local CS #N**)
- **Rotation** (select one of existing **Rotations** presented in the project tree by elements **Local CS #N > Rotation > Rotation #N**, belonging to the LCS-M selected before).



The velocity vector on such **Boundary condition** will be defined by the formula:

$$\mathbf{V} = \mathbf{V}_{BC} + \boldsymbol{\omega} \times \mathbf{r}$$

where $\mathbf{V}_{BC}$ is the velocity, which is defined by the **Boundary condition** (it depends on the boundary condition itself and on the velocity of a moving body, if the **Boundary condition** belongs to a **Moving body**), **ω** is the rotational speed that is defined in the **Rotation**, and **r** is the radius vector from the origin of the **Rotation**'s vector to the center of the cell's face, which is formed by the surface of the **Boundary condition**.

Problem setting with rotating surfaces by means of Moving bodies

For problems with rotating surfaces you can also use **Moving bodies**. However, during the movement of the **Moving body** the computational resources would be spent on each time step because of rebuilding the computational grid. Also in such problem settings the time step would be limited by the velocity of the surface and by the surface CFL.

6.8.10.2.2 Translations in LCS-M

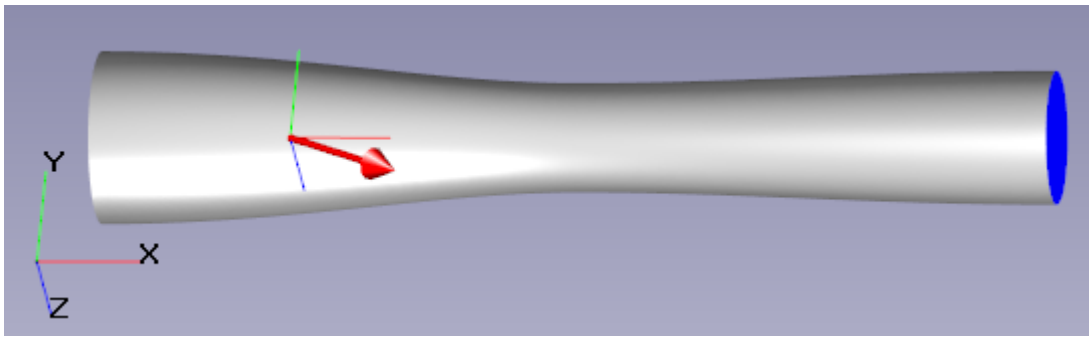
A **Translation** (a translational motion) can be assigned to a **Boundary condition** to set additional velocity of the surface of the **Boundary condition**.

An example of use of such **Boundary condition** is simulating of external flow around a car, with specifying the forward air flow and the surface of the road under the car. In this example, the **Boundary condition**, which relates to the road's surface, is linked to a certain LCS-M and its element **Translation #N**.

A **Translation** can be specified to standard and/or imported geometric objects (but not to **Moving bodies**). Such geometric objects might move according to the solution and follow other moving objects. You can use this to specify motion of areas where adaptation operates, to set variables, to calculate integral values.

Also it is possible to use visualization layers **Vectors** and **Nodal loadings** (when **Coordinate System = Local** is set in their properties) in moving relative coordinate system. So, it is possible to visualize velocity vectors not in the absolute coordinate system only, but also in the local coordinate system of the body, which is flown around.

Each **Translation** is determined by components of its *velocity* along the axes of the LCS-M. Velocity of the translational motion can be defined as a constant, as a formula or as a table. If a **Translation** is specified by a non-zero constant, it will be displayed in the **View** window as an arrow pointing along the direction of the movement (see the illustration below).



The **Translation** element is displayed as an arrow pointing along the direction of the movement

6.8.10.2.3 Specifics of use Rotation and Translation

If the geometry model of the computational domain moves at a *constant* speed relatively to the ACS, then displaying the computation's results in a movable local coordinate system causes only adding the speed of the relative movement, since the local coordinate system is inertial.

Otherwise, if the geometry model of the computational domain *rotates* and/or moves with *nonuniform* velocity relatively to the ACS, then the local coordinate system will be non-inertial.

FlowVision displays the velocity $\vec{V}$ and other vector variables in the *concurrent coordinate system*. This is the absolute coordinate system matching the local coordinate system at the current moment of time.

So, the user views the geometry model of the computational domain in the local coordinate system, and the vector variables in the absolute coordinate system.

Computations of flow and heat and mass exchange are performed with taking into account the movement, which is defined in the LCS-M (for example, when rotation is present, the heat equations are stated with taking into account the rothalpy), and then the computation's results are converted into the ACS.

Example: In a project that computes the flow moving around a complex shape rotor rotating in a stationary cylindrical casing, it is convenient to introduce a rotating local coordinate system, in which the rotor is stationary and the casing rotates. The results of the computation are visualized as a field of velocities, which values measured in the absolute coordinate system, displayed over the background of the computational domain where the rotor isn't moving.

When diameter of the rotating subregion is large, on the outer boundary conditions, which permit inflows or outflows, loss of accuracy can occur. This loss of accuracy manifests in artificial curls appearing on the boundary. The loss of accuracy is caused by correlation of velocities on the boundary, which isn't rotating in the absolute coordinate system, and velocities in the volume of the rotating subregion when the distances from the axis of rotation are large.

Simulations with rotating coordinate systems have limitation on the size of the rotating **Subregion**. If the rotating subregion is too big, on its boundaries away from the axis of rotation, nonphysical fluctuations of variables might occur due to loss of accuracy.

To avoid such fluctuations you have to reduce size of the computational domain. When this is impossible, you have to use a sliding surface around the rotor and the remaining part of the computational is to be simulated in a non-rotating coordinate system.

6.8.10.2.4 Autorotation (changing the rotation speed by the forward flow)

When the rotation speed is the sought value, i.e. when the rotation speed of the rotor in a simulation with rotating coordinate system is unknown and determined by the forward flow, the **Autorotation** parameters in properties of the **Rotation** are applied.

The rotation speed will be so calculated based on the torque from the fluid acting on the rotor.

To calculate the autorotation you have to specify the initial rotation speed of the rotor, the rotor's moment of inertia, and the external torque (for example, torque due to mechanical losses or the service load).

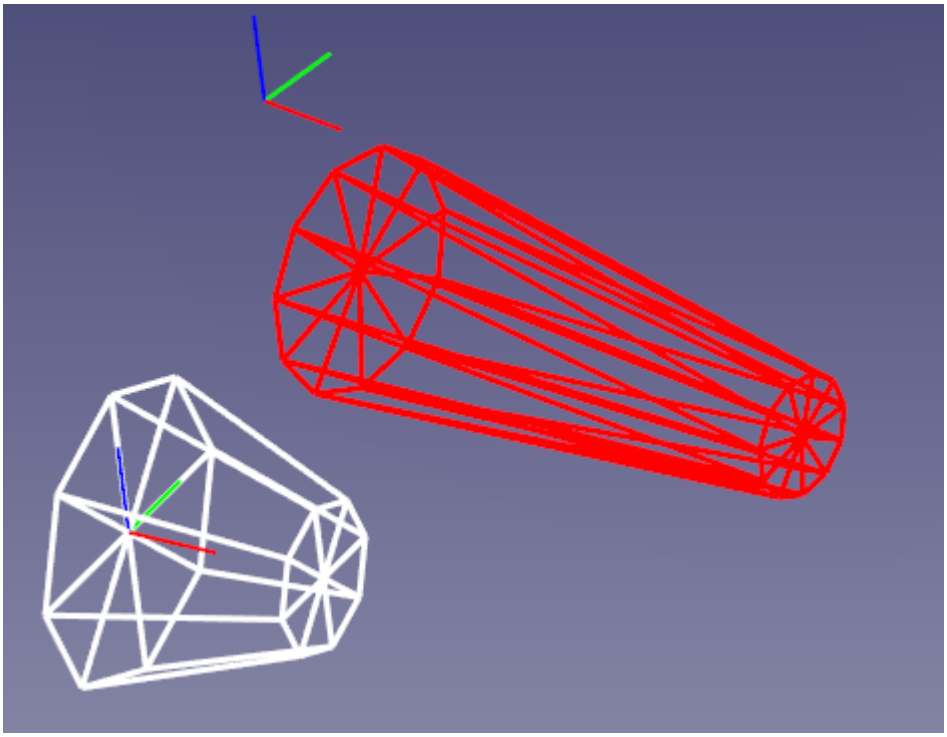
To receive the steady-state solution quickly, you can set:

- zero rotor's moment of inertia (**Autorotation > Moment Inertia = 0**)
- coefficient of additional moment of inertia that equals to 1 to avoid infinitely great increases of the speed (**Autorotation > Coeff. for additional moment inertia = 1**)

When you need a non-steady-state solution (time-dependent problems), when you wish to know how the rotor's speed changes depending on time, you have to specify the actual value of the rotor's moment of inertia and zero coefficient of additional moment of inertia.

6.8.10.3 Geometric object's LCS (LCS-O)

A local coordinate system of a geometric object (LCS-O) defines the location and orientation of a geometric object.



Local coordinate systems of geometric objects (LCS-O)

All geometric objects created in **Pre-Postprocessor** or imported into the project from an external CAD system (*imported objects*) are aligned with their own local coordinate systems (LCS-O).

These local coordinate systems set the location and orientation of geometric objects. Due to this, operations of translation and rotation of geometric objects in the computational domain are operations of translating and rotating the LCS-O relative to ACS.

Position of a LCS-O (and, accordingly, position of its **Object**) can be changed using of the following methods:

- by changing parameters in the group **Location** in the properties of standard geometric objects
- by changing parameters in the group **Initial position** in the properties of **Moving bodies**
- using a **Transformation**

LCS-O is only displayed when its **Object** is selected in the project tree.

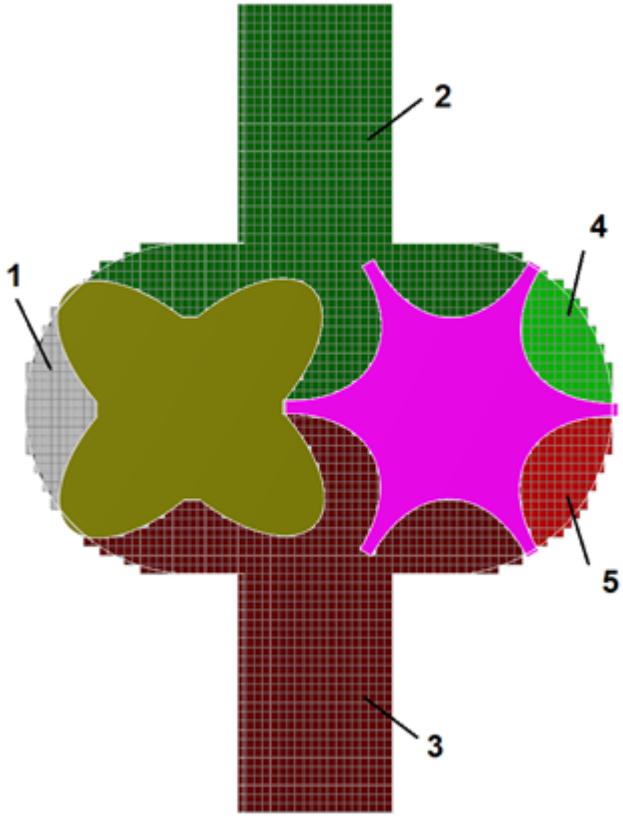
6.8.11 Enclaves

An *enclave* is a connected set of cells within the computational volume of an **Object** separated from other such sets.

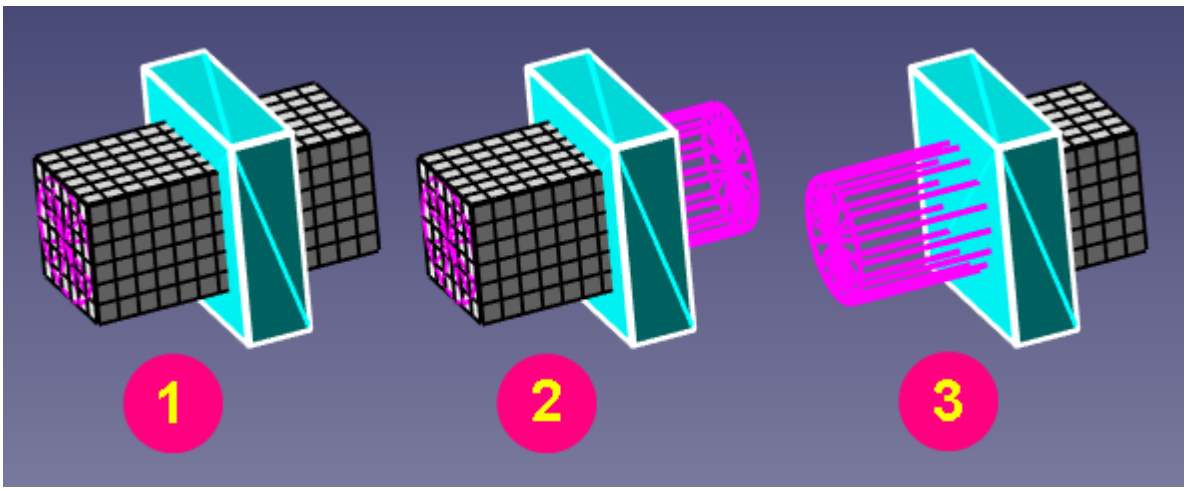
Use of Enclaves

- Separate calculation of **Characteristics** in individual enclaves can be useful when you need to receive values of variables in enclosed volumes, for example, when operation of a compressor is simulated. See also Folder «Characteristics».
- A **Cell set** layer, displaying all cells in the volume of a **Box**, **Cone/cylinder**, or **Ellipsoid/sphere**, can be built on all or individual enclaves. See also Layer «Cell set», user interface.

Illustrations



Housing and gears of a compressor form five enclaves



A **Moving body** splits a **Cylinder** into two enclaves. You can build a **Cell set** on both enclaves (1), or on individual enclaves (2, 3) of the **Cylinder**.

6.9 Movement of geometric objects

FlowVision can simulate three-dimensional motion of geometric objects. You can use the following possibilities:

- use the **Moving body** modifier to simulate movable walls.
- use the **Movement** element to move geometric objects, which have no boundary conditions (standard geometric objects, **Imported objects**).

Simulation of movement of geometric objects with no boundary conditions is useful when in the volume or on the surface of the moving object:

- an **Adaptation** and/or **Modifier** (**Setting variable**, **Resistance**, etc) acts
- *and/or* **Characteristics** are calculated

To set the law of motion of a geometric object, you have to create for it a child element **Movement** in the project tree and specify parameters of the movement (velocity, acceleration, parameters of rotation, initial position).

If a geometric object has to follow a **Moving body** that moves with a law, which is not yet known, then, using the **Formula editor**, you can specify those velocities, which were received from **Characteristics** on the **Moving body**.

See also:

- Element «Movement»
- Modifier «Moving body»
- Folder «Modifiers»

6.10 Computational models

After specifying the geometry model of the computational area, you have to specify the computational **Model**, i.e. to specify:

- which physical processes will be simulated;
- how (with which assumptions) the selected physical processes will be simulated;
- in which geometry volumes particular physical processes will be simulated or not simulated.

Computational models are defined by **Substances**, **Phases**, and models of **Physical processes**:

- a **Substance** specifies properties and the aggregate state of the simulated substance.
- a **Phase** defines the list of **Physical processes** that will be simulated. A **Phase** can include one **Substance** or several **Substances** (to simulate mixtures with calculating the concentrations).
- a **Model** defines how many **Phases** will be simulated (one or two). When the problem includes two phases (for example, interaction between liquid and gas), parameters of inter-phase interaction are defined. Also a **Model** includes a set of initial data, which can be used for specifying the initial conditions.

You can create many **Substances**, **Phases**, and **Models**, but use only some of them in the simulation. This is useful when you have to do simulations for several different **Substances** and computational **Models**.

Creating a computational Model

Specifying calculated media and physical processes in **Subregions** is done in the following order:

1. Specifying the **Substances**, which will be simulated.
2. Creating **Phases**, to which the **Substances** are loaded. Then for each **Phase** with a set of **Substances** the **Physical processes**, which will be simulated in this **Phase**, are specified.
3. Creating **Models** and loading **Phases** into them.

The specified parameters are presented in the project tree in the **Preprocessor** tab.

Assigning Models to Subregions

The computational domain can consist of several volumes, which are **Subregions**.

To each **Subregion** a **Model** can be assigned. If a **Model** is assigned to a **Subregion**, then in this **Subregion** the processes, which are specified in this **Model**, are simulated. If no **Model** is assigned to a **Subregion**, then volume of this **Subregion** will be non-computational.

See sections:

- Substances
- Phases (Continuous and dispersed Phases, One-component continuous media and multicomponent continuous mixtures, Simulating a multi-phase problem with an inter-phase surface using a single Phase)
- Phase transfer
- Turbulence
- Gap model
- Radiation
- Time step

6.10.1 Substances

In the project tree a **Substance** element contains information about the aggregative state and physical properties of the substance.

Aggregative state of a Substance

When a **Substance** is created in *FlowVision*, you have to specify its aggregative state: **Liquid | Gas | Solid**. The selected aggregative state has influence on the set of properties of the **Substance**.

For example, no motion equations can be used for a solid body, so viscosity is not specified in properties of a solid **Substance**.

Also the aggregative state has influence on the way of calculating the sonic speed, total pressure and total temperature, the state equation and the energy conservation equation (see Theory > Properties of Substances).

Computation of the total pressure for liquid is simpler then for gas and the system of equations for liquid might be more numerically stable (rapidly converging) and the solution be more accurate. So, when flows of gas are low-subsonic (when variation of gas density is negligible), it is reasonable to select the **Liquid** aggregative state for the **Substance** and specify **Density** as a constant.

Methods of specifying parameters of a Substance

You can create a **Substance** by two methods:

- manually, when you specify each property of the **Substance**.
- by loading properties of the **Substance** from a **Substance Database**. After the loading you are able to change any of the loaded properties of the **Substance**.

The *FlowVision*'s **Substance Database** stores the most frequently used substances and their properties for various aggregative states.

For example, the water in a **Substance Database** is presented in four phases:

- gas
- gas (equilibrium)
- liquid
- solid (ice)

The "**(equilibrium)**" note in the selection option of a gas phase means that the properties of the **Substance** assume dissociation processes and changing properties of the gas at high temperatures due to changing the ratio of the mixture.

The standard *FlowVision*'s **Substance Database** is created based on generally available reference books and third-party thermodynamical software. The **Substance Database** contains minimal set of the must used substances for majority of cases. Most of properties are set for normal pressure and for a narrow range of temperatures.

When you use **Substances** from the standard **Substance Database**, you have to attend to their properties, as the properties might differ from properties of your **Substance** of the same name due to admixtures/impurities or different operating range of temperatures and pressures.

When you need, you can create your own (user) **Substance Database** with required properties in the required range of pressures/temperatures.

See also: Substance Database Editor.

Important specifics of properties of Substances

Properties of **Substances** generally depend on pressure and temperature. *FlowVision* allows specifying these dependencies by tables or formulae.

Also viscosity of non-Newtonian fluids can be set by one of the three predefined laws.

In any dependencies of the **Substance** properties, pressure and temperature are always set as absolute values!

If chemical reactions, combustion or phase transitions are not taken into account in the computation, then it is recommended for the used **Substance**, which has been loaded from the **Substance Database**, to set zero **Enthalpy of formation** (h_0). This will improve accuracy of the simulation.

If simulation of some physical process is not planned, you don't have to specify the corresponding properties of the **Substance**. So, for example, if heat transfer is not to be simulated, you don't have to specify thermophysical properties of the **Substance**.

If some property of a **Substance** is used in the selected equations and has an incorrect (for example, zero) value, it will be marked in the project tree with a "**!**" symbol.

References

- Folder «Substances» (user interface of elements **Substance** in the project tree)
- Theory > Properties of Substances
- Theory > Physical processes
- Loading a Substance from the Substance Database

- Group of settings "User libraries" in the basic settings of Pre-Postprocessor (the network path to the user **Substance Databases** is set there by the **Substances** parameter)
- Substance Database Editor
- Settings for simulating non-Newtonian fluids

6.10.2 Phases

The **Phase** element defines:
- composition of the simulated single-phase mixture
- physical processes (set of equations), which will be computed for this **Phase**
- parameters of the physical processes, which have been selected

Navigate to sections below:
- Continuous and dispersed Phases
- One-component continuous media and multicomponent continuous mixtures
- Simulating a multi-phase problem with an inter-phase surface using a single Phase
- Spectra of particle sizes

6.10.2.1 Continuous and dispersed Phases

When you create a **Phase**, you have to specify its type, which can be either:
- **Continuous**
- dispersed of the **Particles** type
- dispersed of the **Carcass** type

A dispersed phase can be used in two-phase computational models. You can use a dispersed phase to simulate:
- motion of solid particles in continuous medium
- combustion of particles (for example, coal particles in coal-air mixture)
- spraying the liquid with breakup and coalescence of droplets
- floating-up bubbles in liquid with changing the size of the bubble particles
- evaporating droplet particles at their motion in continuous gas medium with changing the size of the particles
- motion in porous medium (these are simulations of liquid's flow through porous medium or across a bundle of tubes)

See also:
- Physical processes > Phase transfer
- Specifics of phase transfer for particles
- Spectra of particle sizes
- Folder «Phases»
- Processes in the presence of dispersed medium (in the chapter *Theory*)
- Parameters (in the chapter *Theory*, parameters of dispersed phases)

6.10.2.2 One-component continuous media and multicomponent continuous mixtures

To simulate an one-component continuous media, specify a single **Substance** in the **Phase** (using the **Add/Remove** command from the context menu of the subfolder **Phase #N > Substances**).

To simulate flows of mixtures you can apply either one of the described below methods.

Simulating a Phase with one Substance, which is a mixture

For example, air is a mixture of gases. To simulate flow of air generally it is not necessary to calculate air components separately and calculate their concentrations due to the mixture of the components is homogenous and can be described by a single set of the **Substance**'s properties.

Simulating a Phase with several Substances

If concentration of components in the mixture is nonuniform, then several **Substances** are added into the **Phase**. In this case the mixture components take part in chemical reactions or several flows of different **Substances** are mixing within the computational domain.

6.10.2.3 Simulating a multi-phase problem with a phase interface surface using a single Phase

FlowVision allows solving multiphase problems with phase interface surfaces, which are simulated using the VOF method. But sometimes the multiphase problems with phase interface surfaces can be solved with appropriate accuracy even in an one-phase setting.

For this a **Phase** containing two or more **Substances** is to be created. To minimize diffusion (mixing the phases simulated by the **Substances**), it is necessary to specify very large Schmidt number Sc in properties of the Mass transfer physical process.

This approach has the following advantages:

- It is possible to simulate more then two components (the VOF method works for two phase problems only by now).
- Sometimes it provides higher simulation performance for complex shaped inter-phase surfaces when multiprocessor computations are used.
- Sometimes it provides better stability of the solution.
- It is possible to use substantially larger time step comparing to the use of the VOF method.

The main disadvantage of this approach: it is not possible to completely eliminate diffusion (mixing) of the components. This affects accuracy of simulating the phase interface surface.

6.10.3 Physical processes

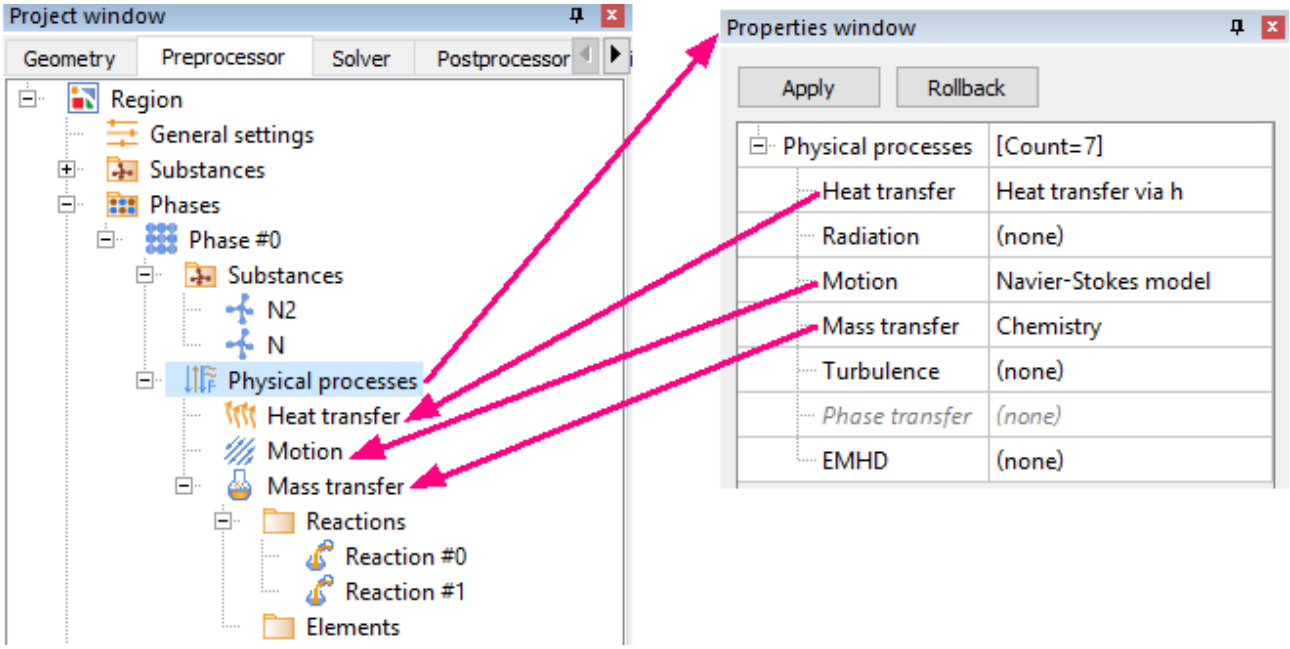
In properties of the folder **Phases > Phase #N > Physical processes** you specify, which equations will be used for simulating the **Phase #N**.

You can specify simulating the following **Physical processes**:

- **Heat transfer**
- **Radiation**
- **Motion**
- **Phase Transfer**
- **Mass transfer**
- **Turbulence**
- **EMHD** (electrohydrodynamics)

To enable calculating a **Physical process**, you have to select from the drop-down list a mathematical model that will be used for the process. Some physical processes can be computed by different methods and be described by different mathematical models. For some processes you can select only one option from the list.

After selection the model of the physical processes, an appropriate child element corresponding to the **Physical process**, which has been specified, will appear in the **Physical processes** folder. Properties of this element contain parameters of the selected mathematical model of the physical process.



6.10.3.1 Heat transfer

The physical process **Heat transfer** includes computation of the energy conservation equations.

Form of this equation depends on the aggregate states of **Substances** included into the **Phase** (for example, simulating the heat transfer in a solid substance uses a simplified formula with no convective term because velocities in a solid body are zero).

Simulating the **Heat transfer** can be enabled by selecting one of the following options:

- Energy equation written via the thermodynamic enthalpy (h); it is selected for simulating incompressible media and slow flows (Mach number < 0.1).
- Energy equation written via the total enthalpy (H); it is selected for simulating compressible media and, particularly, for problems with shock waves.

Accounting of heat generation due to viscous friction

Viscous liquids or rapid flows can cause intensive heating of the liquids due to viscous friction between the flow and the wall or between flows with different velocities. To take this process into account you have to specify the parameter **All terms = Yes** in properties of the element **Phase #N > Physical processes > Heat transfer**.

Specifying user's heat sources

FlowVision has several tools, which can be used to specify heat sources/sinks in computational cells:

- The **Volume heat source** modifier
- Heat sources on **Boundary conditions**
- The source term in an energy equation determined by coefficients D and F – see Eq. (13). This source is used for simulation of the heat exchange between fluid and porous carcass of arbitrary nature. Specifying properly coefficients D and F, one can allow for an energy source (positive or negative) due to the processes which are not simulated directly in the given problem, for instance, the source of energy due to chemical reactions or phase transition. The source term in an energy equation determined by coefficients D and F is calculated in all the computational cells. However, using logical expressions for Cartesian coordinates, one can localize the action of this source term.

6.10.3.2 Radiation

To calculate radiation energy transfer, two main models are used in computational fluid dynamics (CFD):

- Diffusion model
- Surface-surface model

Comparing to surface-surface models, diffusion models require far less computational resources (their computation is faster). But domain of applicability of the diffusion models is quite limited because they cannot take into account direction of radiation, i.e. take into account shades, and they also cannot calculate re-reflection from surfaces. For example, it is not possible to use diffusion models to calculate nonuniform heating of surfaces in a room by sunlight because the model doesn't take into account shadows from from furniture and walls.

The diffusion approach, can generally be used at relatively weak anisotropy of the radiation field. The requirement of weak anisotropy is fulfilled when free-path length of a photon is much smaller then the characteristic dimension in the simulated problem. However, as practice shows, the diffusion approach very often gives good results for simulations of radiation gas dynamics (RGD).

FlowVision implements the following models of radiation heat transfer:

- P1 (diffusion model)
- optically thin layer (diffusion model)
- discrete-ordinates method (surface-surface model)

Radiation model P1

An example of a task that is solved by the P1 model is radiation between very closely spaced surfaces.

Radiation model "Optically thin layer"

The model of an optically thin layer is used only for simulations of combustion to take into account heat losses of the burning medium due to radiation. Here it is assumed that the heat quits the computational domain due to

radiation having no dispersion of radiation in translucent medium. So, for the optically thin layer model, you don't need to specify parameters of radiation on boundary conditions.

Radiation model "Discrete-ordinates method"

The discrete-ordinates method is the most comprehensive model of radiation energy transfer. This model is applicable for the whole range of optical depth and allows to solve such problems as:

- surface to surface radiation energy transfer
- taking radiation into account in simulations of combustion
- radiation transfer in semi translucent media

This method has moderate requirements to computational resources when discretization is typical (no more then 32 solid angles are used) but when number of solid angles is large, substantial computational resources will be required.

See details in the chapter **Theory**:

- Radiation
- Notations
- Parameters
- P1
  - Equations for method P1
  - Boundary conditions
  - Template 'Wall'
  - Template 'Symmetry'
  - Templates 'Inlet/Outlet', 'Free outlet', 'Nonreflecting'
  - Template 'Connected'
- Optically thin layer
- Discrete-ordinates method
  - Equations for the discrete-ordinates method
- References

6.10.3.3 Motion

The **Motion** physical process is simulated by:

- Navier-Stokes equations
- Darcy equations (for porous media)

Navier-Stokes model

When viscosity is set as zero and the **Heat transfer** physical process is not enabled, the Navier-Stokes equation (the mass conservation law) reduces to the Euler's equation (the perfect-fluid equation).

To take the turbulent losses into account, the system of Navier-Stokes equations can be appended with appropriate equation by enabling the **Turbulence** physical process.

See also: Theory > Physical processes > Motion > Equations.

Darcy model

The Darcy model describes flows in porous media.

The Darcy model's system of equations is simpler then system of equations of the Navier-Stokes model and it is more stable for simulations with high volume resistances.

This model has some limitations, it *doesn't support*:

- anisotropic heat exchange
- use of conjugated boundary conditions

If the hydrodynamic resistance is not set (using a **Resistance** or **Anisotropic resistance** modifier), the Darcy motion model has no meaning. The program applies the lower limit for hydrodynamic resistance coefficient as 10^{-6} [$kg \cdot m^{-3} \cdot s^{-1}$].

The same value is automatically applied when the resistance is not set. So the result obtained without specifying the hydrodynamic resistance will be same as when hydrodynamic resistance 10^{-6} [$kg \cdot m^{-3} \cdot s^{-1}$] is applied.

6.10.3.4 Phase transfer

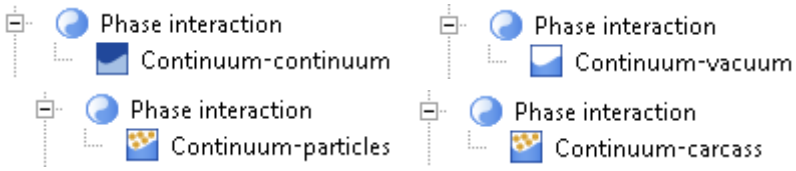
FlowVision allows simulation of two-phase processes:

- in dispersed media:
  - motion of liquid in porous medium
  - motion of particles or bubbles in continuous medium
- interaction of two immiscible phases with an inter-phase surface (free surface)

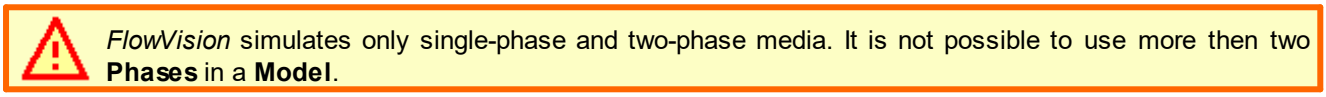
The model of phase interaction turns on automatically when more then one **Phase** are presented in a **Model**.

Each time when two **Phases** are presented in a **Model**, in the project tree in the folder **Model #N > Phase interaction** a new element will appear that can be one of the following types:

- **Continuum-continuum**, simulating of two immiscible phases.
- **Continuum-vacuum**, simulating of two immiscible phases where one of the two **Phases** doesn't contain **Substances**. The **Phase**, which has no **Substances**, is referred as **Vacuum**. The **Continuum-vacuum** phase interaction is used when shape of the inter-phase surface is defined mostly by the heavier phase while motion of the lighter phase may be omitted. In this case simulating of cavitation is possible.
- **Continuum-particles**, simulating of motion of either particles or bubbles within a continuous phase.
- **Continuum-carcass**, simulating of interaction between a flow of a continuous phase and a porous carcass.



In these elements you specify interaction of two **Phases**.


FlowVision simulates only single-phase and two-phase media. It is not possible to use more then two **Phases** in a **Model**.

Also you can use an approximate method of simulating multiphase flows, which have more then two phases. See section Simulating a multi-phase problem with a phase interface surface using a single Phase.

See details in sections below:

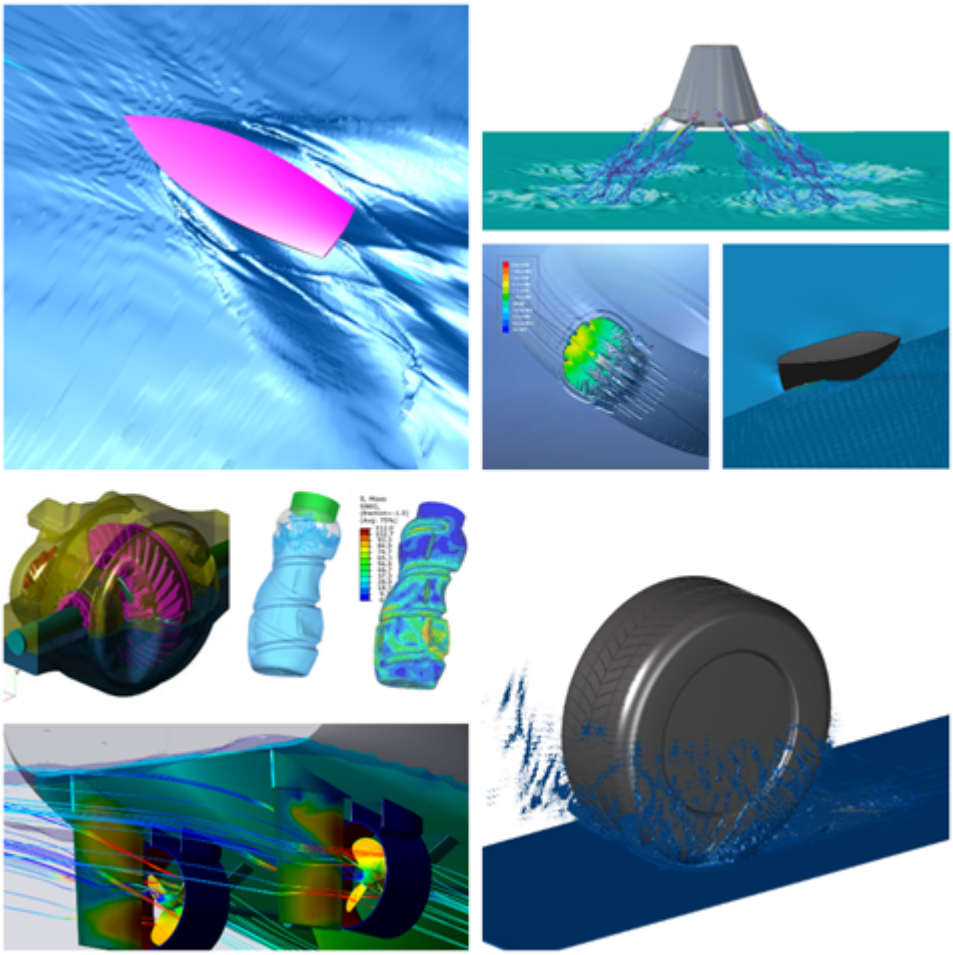
- Two phase media with an inter-phase surface
- Applicability of the VOF model
- Specifics of phase transfer for particles

6.10.3.4.1 Two phase media with an inter-phase surface (VOF model)

The **VOF model** is designed for simulation of flows of immiscible fluids with contact boundaries or free surfaces.

Below some examples of problems, in which activation of the process **Phase Transfer** is required, are listed:

- water flow over a ship,
- filing a mold with liquid metal,
- lifting a liquid in a capillary.



Examples of calculation of the phase interface surface between two continuous **Phases**

To apply the VOF model, you have to:

1. Create two continuous **Phases**.
2. Create a **Model** and add these two **Phases** into it.

After adding the two continuous **Phases** into some **Model #N**, the following changes will occur:

- for these **Phases**, in properties of their child elements **Physical processes** it will be set **Phase transfer** = **VOF model** (this parameter is set by the program automatically and cannot be changed by the user)
- in the folder **Model #N > Phase interaction** an element **Continuum-continuum** or **Continuum-vacuum** will appear.

FlowVision allows simulating of no more then two immiscible continuous **Phases** within one **Model**.

Note: If densities of the **Phases** substantially differ, then it is recommended to place the **Phases** in the project tree in decreasing order by their densities to improve stability of the solution.

In certain problem settings motion of the low-density continuous **Phase** can be neglected (for instance, in simulation of the water flow around a slow vessel). In such cases it is recommended to specify a non-calculated phase **Vacuum** (a phase without physical processes) as the second continuous **Phase**. It must be placed on the last position in the folder **Preprocessor > Models > Model #N > Phases**.

Note: The term "**Vacuum**" is a nickname for the **Phase** that has no physical processes. Pressure on the free surface is equal not to absolute zero but to the value on the boundary from the side of the continuous **Phases** with physical processes.

Values of surface tensions of liquids are set in properties of related **Substances**.

Developers discover that implementations of the VOF model cause issues with poor mass conservation. So in many CFD products substantial loss of mass in an enclosed volume can be observed. *FlowVision* applies a stack of technologies that limit the loss of mass on the level below 5% even in the case of complex high-curvature shapes of the free surfaces.

See also:

- Description of the physical process Phase transfer (for continuous **Phases**) in the chapter Theory.
- Applicability of the VOF model
- Mass conservation and the VOF model
- Time step in simulations with VOF, frozen free surface
- Heat exchange between a continuous Phase and Vacuum through the free surface
- Recommendations on specifying initial conditions for VOF
- Folder «Phases» (description of the user interface)
- Folder «Models > Model #N > Phase interaction» (description of the user interface)
- VOF-particles

6.10.3.4.1.1 Applicability of the VOF model

There are the following limitations for applicability of the VOF model:

- In one problem setting it is possible to simulate no more then two continuous **Phases**.
- *FlowVision* doesn't simulate intersection of the inter-phase surface with surfaces of connected boundary conditions conjugated by speed (i.e. boundary conditions **Conjugate all variables**, **Periodic surface**, and **Sliding surface**).

Interaction with dispersed particles

FlowVision can do simulations with three **Phases** where two **Phases** are continuous which an inter-phase surface between them and the third dispersed **Phase** of the **Particles** type.

In such simulations the mass of the dispersed **Phase** transfers to the mass of a continuous **Phase** when a particle intersects the inter-phase surface between the continuous **Phases**. An example of such problem setting is rain. Drops of the rain (the dispersed **Phase**) move through the first continuous **Phase** (air). The second continuous **Phase** is the water in a puddle. Drops disappear when they contact the surface of the puddle (the inter-phase surface) and volume of the puddle increases.

Another example of a simulation with three **Phases** is aircraft icing when water particles sediment on a non-moving free surface of the ice body on the wing. See details in sections Boundary conditions «Wall, film», Crystallization, and in the subsection Parameters of the physical process "Crystallization" (simulating the icing).

6.10.3.4.1.2 Mass conservation and the VOF model

Simulation of motion of an inter-phase surface using the VOF model generally includes some mathematical issues that cause loss of the fluid's mass. This loss can be critical in problem settings with enclosed volumes and in settings that are sensitive to precise values of the fluid's mass in the computational domain.

This issues related to the numerical simulation can be divided into the following two categories:

- Infinitely small relative changes of cell volumes.
- Drops or bubbles that are much less then cells.

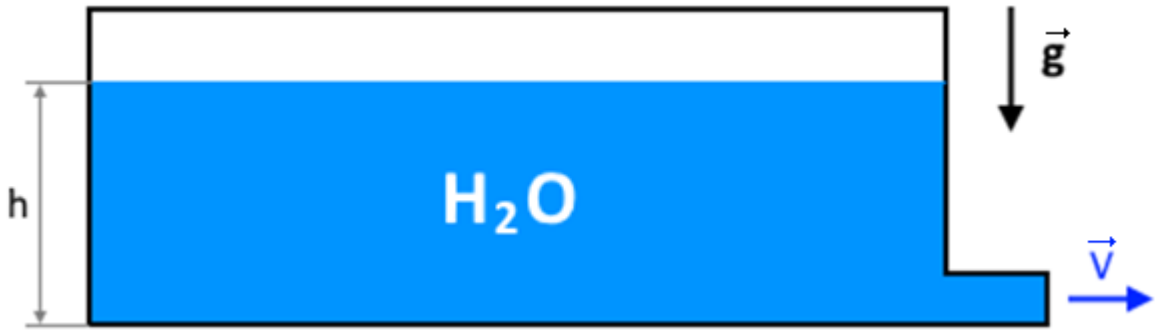
Correct and accurate simulating using the VOF model requires taking into account these numerical issues and apply their workarounds that are available in *FlowVision*.

To monitor accuracy of simulations with VOF, it is recommended checks the mass of one of the **Phases** in the computational volume and compare change of he mass to flows of the **Phase** on inlets and outlets.

Small change of the Phase's relative volume in cells

If the **Phase**'s relative volume in a cell changes to infinitely small value per a time step, the machine precision do not provide the correct calculation of changing the fluid's volume. If area of the inter-phase surface is large in the circumstances, the total (by all cells) error becomes substantial and the mass loss becomes visible.

This issue is most visible in problem settings that include vessels with large area of the free surface and small flow of the fluid at an inlet or outlet as shown on the illustration below:



In a such problem setting you can observe constant flow on the outlet and stationary level (h) of the inter-phase surface all the time. Due to infinitely small changes of the phase volume per one time step, position of the free surface doesn't change. The program just ignores the infinitely small changes to provide convergence of the solution.

There are several methods to workaround this issue:

- Increase the time step to make the change of the **Phase**'s relative volume per time step substantial.
- Do not simulate motion of the free surface that has large area if this motion is not interesting in this problem setting (you can also replace this free surface with a boundary condition or a moving wall).
- Refine the computational grid near the free surface. Then change of the relative volume of the **Phase** in the cells per time step will be substantial.
- When you don't need high accuracy of the mass conservation, you can disable the mechanism of the mass conservation: specify **Multiphase C > Phase conservative = No** in the advanced settings of **Solver**.

If you really need simulation of the inter-phase surface, the grid refining will require too much resources and increasing the time step cause impossibility to solve important physical processes, you can try to change the **VOF level** parameter (the cut level for **VOF**) in the advanced settings of **Solver**. This parameter tells **Solver** what change of the relative volume per time step will be ignored. The less is the value of the **VOF level** parameter, the more accurate the solution will be.

Excessive change the cut level for **VOF** can cause nonconvergence of the solution or incorrect solution. Use this method only in exceptional cases and carefully monitor the process of the solution to detect anomalies.

Droplets and bubbles

In simulations with complex shaped inter-phase surface, droplets (volumes of liquid enclosed by gas or other liquid) and bubbles (volumes of gas enclosed by liquid) can appear.

When size of the droplets/bubbles becomes much less then size of a computational cell, then, to keep stability of the solution, such droplets and bubbles are excluded from the computation or their mass is added to larger volumes.

When the mass conservation is enabled (when **Multiphase C > Phase conservative = Yes** is set in the advanced settings of **Solver**) the droplet/bubble will be excluded from the computation and mass of the droplet/bubble will be added to larger volume of the same **Phase**. But this approach can cause incorrect volume distribution of the mass, when substantial shift of the mass in some direction appears due to the droplets/bubbles.

When the mass conservation is disabled (**Multiphase C > Phase conservative = No**), then the mass of the infinitely small volumes of the **Phase** will be excluded from the computation and will be lost; large incorrect value of the mass will be eventually accumulated.

To do approximate computations of very small liquid volumes, *FlowVision* can use VOF-particles that are enabled by the **Multiphase C > VOF-particles** parameter in the advanced settings of **Solver**). The program doesn't calculate shapes of such particles but their motion is simulated using the Lagrange equations.

Use of VOF-particles substantially improves the mass conservation in simulations with complex shapes of free surfaces and doesn't require much additional consumption of computational resources.

6.10.3.4.1.3 Time step in simulations with VOF, frozen free surface

To obtain a correct solution of a simulation with a free surface, value of **Surface CFL** is to be ≤**1** (see sections Time step and Element «Time step»).

It is recommended to use the implicit numerical scheme and in properties of the time step specify:

- **Surface CFL = 1**

- **10 < Convective CFL < 100**

Frozen free surface

Specifying **Surface CFL=1** can dramatically reduce the time step. This limitation is required for more accurate solution of motion the free surface, but it is is excessive for other physical processes.

Taking this into account, the problem of liquid friction around a boat can be solved in two stages:

1. In the first stage the phase transfer is to be disabled and the free surface becomes "frozen". The time step in this situation will be limited by **Convective CFL** only. This computation will give steady-state solution for the main flow of the liquid with an immobile inter-phase surface. Thus we obtain a good initial approximation.
2. "Unfreeze" the free surface (specify **Surface CFL = 1**) and run the computation. As a result, a steady-state solution will be found for the wave component of the friction force and the value of viscous friction force on the boat's surface will be calculated more accurately.

6.10.3.4.1.4 Heat exchange between a continuous Phase and Vacuum through the free surface

When interaction of a continuous **Phase** and **Vacuum** (which is a continuous **Phase** without physical processes) is simulated, it is possible to take into account the heat exchange through the free surface. To do so, you have to specify **Ext. heat exchange = Yes** in properties of the element **Model #N > Phase interaction > Continuum-vacuum**.

Also in properties of the element **Model #N > Phase interaction > Continuum-vacuum** you have to specify:

- The heat exchange coefficient of the heat transfer to the external medium (from the continuous **Phase** to **Vacuum**). It is set by the **Heat-transfer coef.** parameter.
- Temperature of the external medium (temperature of the **Vacuum** phase). It is set by the **T of external medium** parameter.

6.10.3.4.1.5 Recommendations on specifying initial conditions for VOF

When you specify **Initial conditions** for the **VOF** variable, it is recommended to set **Method = Replace in cropped volume**. This allows the program to spatially limit the **Phase** by a geometry object accurately.

If you set **Method = Replace in full volume**, then level of the liquid will differ from the desired level by size of one computational cell. In simulations with hydrostatics (for example, in simulations of flow around a boat) this cause that the hydrostatic zero and the initial level of the liquid will differ and this can increase the required time for convergence of the solution.

See also:

- Details about use of the **Method** parameter in properties of **Modifiers** and **Initial conditions** see in sections: Modifiers, Initial conditions, Folder «Modifiers», Folder «Initial conditions».
- Details about hydrostatics see in sections: Hydrostatic component of pressure, Hydrostatics.

6.10.3.4.2 Porous media and carcasses

FlowVision can simulate flows of liquid or gas in porous media.

Here are some examples that we refer as porous media:

- ground and soil
- filters
- any porous and permeable medium
- radiators and heat exchangers with many pipes or channels, throw or across which the continuous medium flows

Specific feature of such media is high resistance to motion of the the continuous medium, depending on its velocity.

Let's consider motion of liquid through volume filled with heat exchanger's pipes. Direct simulation of the flow is a resource-intensive task. Experimental data give us resistance coefficients for flows through grids and radiators with pipes of various diameters, so we can present pipes of heat exchangers as a dispersed medium, a **Phase** of the **Carcass** type. This will allow the program not to solve each pipe and vortex by the computational grid. The pipes will be replaced by the resistance coefficient of the carcass.

To run simulations of flow in a porous medium, you have follow the steps:

1. Create a **Continuous Phase**, which will be used for simulating the continuous medium.
2. Create a **Phase** of the **Carcass** type, which will be used for simulating the porous medium.
3. Create a **Model**, which will include these two **Phases**.
4. Specify a volume resistance by a **Modifier**, which will determine the low of how the resistance depends on velocity.

5. If necessary, specify the process of the heat exchange and its parameters (if the carcass contains heat conductor, for example, if the carcass is a bundle of pipes with hot water) and a **Modifier** for a **Volume heat source** in the **Carcass**.

6.10.3.4.2.1 Specifics of phase transfer for particles

The implemented model of the dispersed particles allows you to simulate the following problems:

- motion of solid particles in liquid/gas
- motion of droplets in liquid/gas
- motion of bubbles in liquid

A **Continuous Phase** and a **Particles Phase** are simulated by the 'Euler-Euler' method.

Motion of particles and transfer processes are simulated by the 'Euler-Euler' method. This approach assumes solving of "continuum" equations that describe conservation laws of momentum, energy and mass for all **Phases** presented in the **Model**. As a result of this simulation, we receive concentrations of the particles. When sizes of the particles are set by a spectrum, concentrations are calculated separately for each size spectrum group. See details in sections Spectra of particle sizes and Theory > ... > Equations for particles > Process 'Phase transfer'.

Size of particles

In problem settings with a **Phase** of the **Particles** type without simulating the **Mass transfer** physical process, it is necessary to set diameter of particles because mass of the particles is determined by diameter of the particles and density of the substance.

Sizes of particles in each *size group* in a **Size spectrum** are set by parameters **Size groups > [N] > Diam. particles**.

When **Mass transfer** is simulated, size of particles can vary within the computational domain but it is specified on boundary conditions and in initial conditions by some of **Size spectra**.

See also:

- section Spectra of particle sizes
- description of the parameter **Number of size groups** in properties of the folder Phases > Phase #N
- subsection *Folder «Phases > Phase #N > Size spectra» and elements «Size spectrum #N»*

Interaction of phases

To set interaction of a **Continuous** phase and a phase **Particles**, you have to create a **Model**, add into it one **Continuous** phase and one phase of the **Particles** type. An element **Phase interaction > Continuum-particles** will be created automatically in the project tree. In properties of this element you can specify parameters that have influence on the heat transfer, mass transfer, properties of the particles, etc.

Limitations of applicability of Dispersed Phases

- The program doesn't support calculation of several **Dispersed Phases** in one **Model**. To simulate particles of different sizes, use **Size spectra**.
- Simulation of **Dispersed Phases** can *not* be done along with simulation of free or contact surfaces (i.e. use of the VoF method when **Phase transfer** in the continuous **Phase** is enabled).

6.10.3.4.2.2 Spectra of particle sizes

For dispersed **Phases** of the **Particles** type, *FlowVision* supports simulation of one or several *size spectra* of particles. Diameters of particles and volume fractions of the size groups are specified by discrete **Size spectra** that are stored in the project tree in the folder **Phase #N > Size spectra**.

When size spectra are used, you have to specify for **Boundary conditions** and **Initial data** links to some of existing **Size spectra** instead of numerical values.

Each size group in a **Size spectrum** specifies the following parameters:

- diameter d^i of particles of this group
- volume fraction φ^i of particles of this group in the dispersed **Phase**

Sum of volume fractions of all size groups in a **Size spectrum** is to be **1**:

$$\sum_i \varphi^i = 1$$

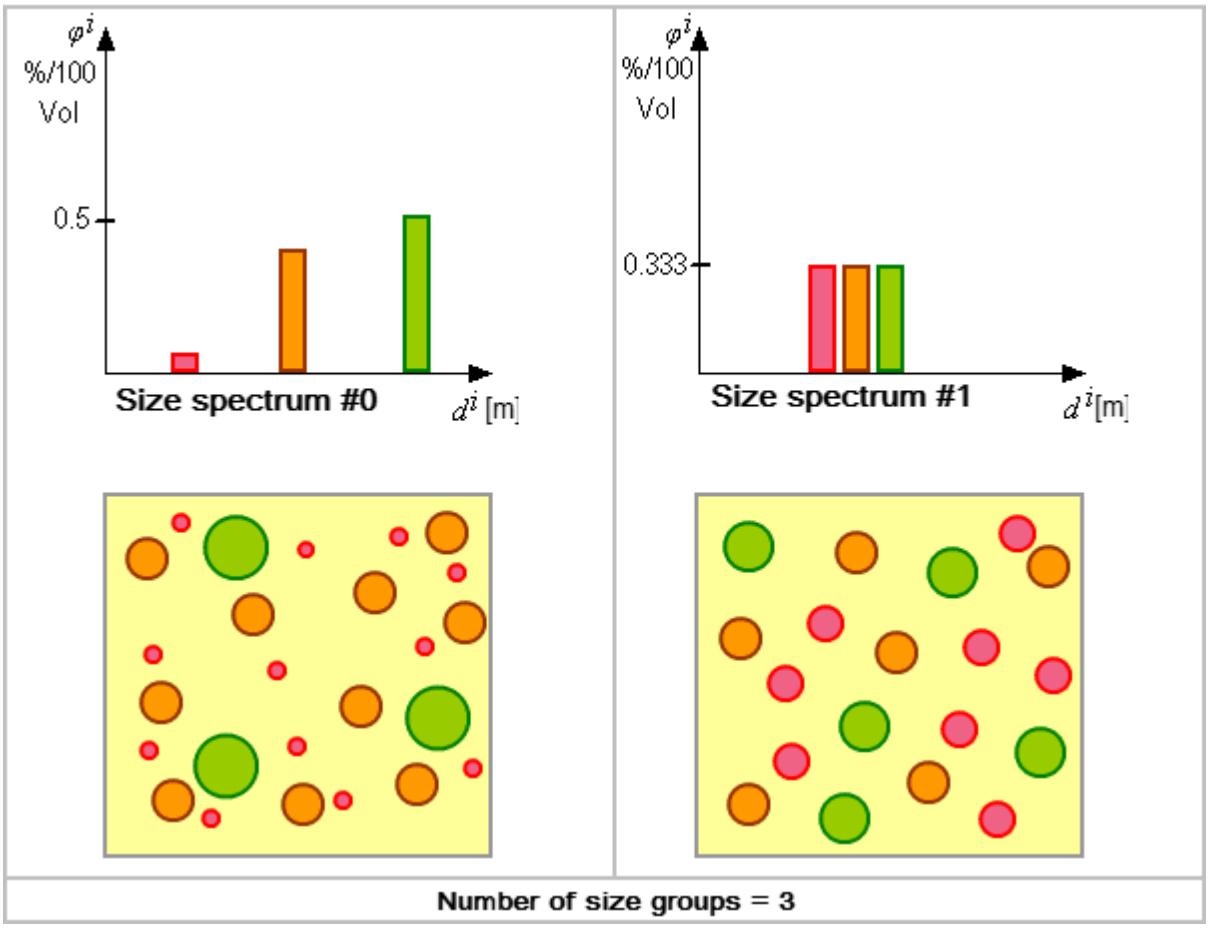
Each size spectrum has its own equations that are solved by the program (phase transfer, mass transfer (evaporation), motion, energy transfer) with taking interactions with the continuous phase into account.

The program simulates no interaction between size spectra (mass transfer between the spectra, momentum and energy transfer due to breakup and coalescence of droplets).

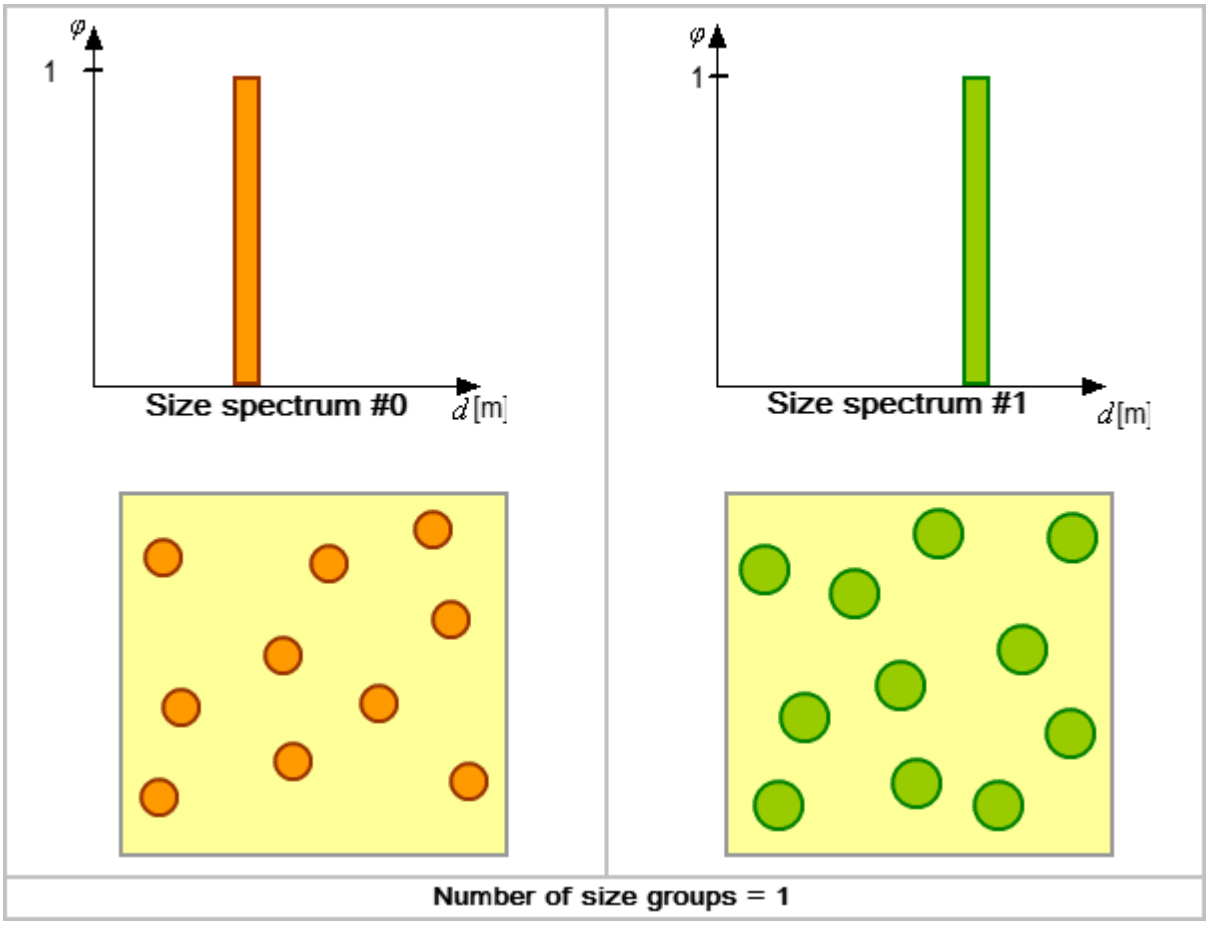
Applying the size spectra makes sense in cases when:

- simulating motion of particles of several constant sizes
- in simulating of aircraft icing; this improves accuracy of computations for mass transfer (aviation standards require taking the size spectra into account)
- in simulating of coal combustion (parameters of combustion change depending on diameters of the particles)

Conception of **Size spectra** is shown on illustrations below:



Number of size groups = 3



Number of size groups = 1

Variables of a **Particles** dispersed phase that relate to different size groups are marked in the program's interface by indexes of their size groups in square brackets.

Example: **Velocity (disp.) [0]**, **Velocity (disp.) [1]**, etc. (numeration of size groups starts from **0**).

Specifics of use Size spectra in simulations with mass transfer

When **Mass transfer** is enabled, diameters of the dispersed particles can vary around the computational domain and change during the computation. The problem setting requires that you specify diameters of dispersed particles on Boundary conditions and in Initial data:

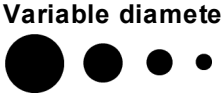
- on **Boundary conditions** the **Diameter** element is either associated with some **Size spectrum #N** or set as **Symmetry** or as **Permeable surface**.
- in **Initial data** the **Diameter** element is associated with some **Size spectrum #N**.

In simulations with *disabled* **Mass transfer**, the **Size spectrum #N** is used, which is set by the **Used drop size spectrum** parameter from properties of the element Phase #N > Physical process > Phase transfer.

To calculate masses of the particles, constant values of particle sizes and concentrations from this **Size spectrum** will be used.

Simulations with enabled **Mass transfer** assume changing of diameters of particles in each size group during the computation. When flows with different diameters of particles in the same size group are mixing, values of concentrations and diameters of particles of this group without taking into account specific physics of coalescence and breaking up the particles.

Depending on the specified model of dispersed mass transfer (it is set by the **Mass transfer** parameter in properties of the element Phase #N > Physical processes) and the model of dispersed particles (it is set by the **Model for particles** parameter in properties of the element Phase #N > Physical processes > Mass transfer of the dispersed **Phase Particles**), some or other Size spectra of particles are used in simulations of dispersed mass transfer.

Mass transfer =	Model for particles =	User interface and operation of the program
(none)	*(when the dispersed mass transfer is disabled, the **Model for particles** parameter is not presented in the user interface)*	Individual **Size spectra** cannot be set in initial conditions and on boundary conditions. For the whole dispersed **Phase**, that **Size spectrum #N** is applied, which is set by the **Used drop size spectrum** parameter in properties of the element Phase #N > Physical processes > Phase transfer.
Mass transfer *(the general model of dispersed mass transfer)*	**Variable diameter** 	Individual **Size spectra** are set in initial conditions and on boundary conditions and are applied in the computation. The **Used drop size spectrum** parameter (in properties of the element Phase #N > Physical processes > Phase transfer) is absent.
	Constant diameter 	Technically, the program's user interface allows you to set individual **Size spectra** in initial conditions and on boundary conditions, but they *are ignored* in the computation. Instead of them, the program uses the **Size spectrum #N**, which is set by the **Used drop size spectrum** parameter in properties of the element Phase #N > Physical processes > Phase transfer.
Coal	**Variable diameter** 	Individual **Size spectra** are set in initial conditions and on boundary conditions and *are used* in the computation. The **Used drop size spectrum** parameter (in properties of the element Phase #N > Physical processes > Phase transfer) *is used to calculate limiters* in the algorithm that computes mass loss from the **Particles** phase. It is recommended to select **Used drop size spectrum** as that **Size spectra #N**, which corresponds to the maximal by mass fraction of coal particles supplying to a burner or boiler.

	Constant diameter 	Technically, the program's user interface allows you to set individual **Size spectra** in initial conditions and on boundary conditions, but they *are ignored* in the computation. Instead of them, the program uses the **Size spectrum #N**, which is set by the **Used drop size spectrum** parameter in properties of the element **Phase #N > Physical processes > Phase transfer**.

See also:

- section Folder «Phases» and subsection *Folder «Phases > Phase #N > Size spectra» and elements «Size spectrum #N»* (description of the user interface)
- Processes in the presence of dispersed medium (in the chapter *Theory*)
- Parameters (in the chapter *Theory*, parameters of dispersed **Phases**)

6.10.3.5 Mass transfer

The **Mass transfer** allows the program to calculate changing of concentrations of several **Substances** in a **Phase**.

For continuous **Phases** *FlowVision* implements the following mass transfer models:

- **Mixing**, which simulates mixing of several substances, with taking into account diffusion and simple (single-stage) chemical reactions.
- **Combustion**, which simulates an irreversible single-stage reaction with two reagent substances (fuel and oxidant) and generating one ore two combustion products.
- **Chemistry**, which simulates successive (multistage) chemical reactions. This model can be recommended for use when there are many reactions with kinetics conforming to the Arrhenius law.
- **Coal**, which simulates combustion of coal (in a continuous **Phase**).

Combustion of multiple types of fuel is a complex multistage reaction. But the simple **Combustion** model is enough for solving many cases with adequate accuracy. The **Chemistry** model can be recommended to simulate only those complex chemical reactions, which require very high accuracy of the simulation.

All these mass transfer models for continuous **Phases** can include simulation of **Ablation**.

Calculation of thermal conductivity and viscosity of the mixture of substances in a **Phase** is set by parameters **Viscosity of mixture** and **Thermal conductivity of mixture** in properties of the Phase.

For dispersed **Phases** of the **Particles** type the general mass transfer model (**Mass transfer**) and the coal combustion model (**Coal**) are provided, see Mass transfer in dispersed media.

For dispersed **Phases** of the **Carcass** type the only specialized mass transfer model is provided, see Mass transfer in dispersed media.

See also:

- User interface for specifying parameters of the "Mass transfer" physical process"
- Theory > Physical processes > Mass transfer > Mixing > Equations
- Theory > Physical processes > Mass transfer > Combustion > Equations
- Theory > Physical processes > Mass transfer > Chemistry > Equations
- Theory > Physical processes > Processes in the presence of dispersed medium

6.10.3.5.1 Mixing

The **Mixing** mass transfer model is a simple model for calculating concentrations in homogeneous mixture of several substances.

Creative use of the Mixing model

Simple (single-stage) chemical reaction

When you set parameters D and F by formulae (using the **Formula editor**), you can set a simple single-stage reversible or complete chemical reaction (see examples "Radioactive decay of isotopes" and "Dissociation of oxygen" in chapter Theory).

Multiphase flows

You can use the **Mixing** model to simulate several immiscible liquids without using phase interface surfaces. This can be implemented using a single **Phase** with several **Substances**. And you have to specify a very large value of

the molecular Schmidt number (the **Schmidt** parameter of the Physical processes > Mass transfer element) to minimize the diffusion.

However, you have take into account that diffusion will not be completely excluded; also the diffusion will be worse (will increase) when the computational grid is changing near the boundary between the surfaces.

Use Mixing to visualize a flow

The **Mixing** model can be used for a sophisticated visualization. If you split one **Substance** to several **Substances** with same thermodynamic properties, you can visualize their concentrations by color contours on planes and view dynamics of the flow. Such method is absolutely identical to coloring the liquid for visualization of its flow in real experiments.

Such visualization can be done also after finishing the main computation and receiving a converging solution. To do so, follow the steps:

Step	Actions
1	Specify a coloring **Substance** and enable the **Mass transfer** physical process with the **Mixing** model.
2	In physical processes **Motion** and **Heat transfer** specify **Time step coefficient = -1**, which will stop calculation of the appropriate equations.
3	Continue the computation specifying a source for the new **Substance** (for example, on an inlet boundary condition).
4	Visualize concentration of the new **Substance** using **Color contours** or **Volume visualization**.

See also: Theory > Physical processes > Mass transfer > Mixing > Equations.

6.10.3.5.2 Combustion

The **Combustion** mass transfer model allows you to simulate an irreversible single-stage reaction with two reagent substances (fuel and oxidant) and generating one ore two combustion products. The mixture can also contain other substances that will not participate in the reaction (these are neutral substances).

The following models are used to simulate the combustion:

- **Zeldovich**: this model assumes infinite rate of the combustion reaction. This means that the mixture in a cell burns instantly when matches to the ignition characteristics. This is the fastest model but the least accurate.
- **Arrhenius**: in this model, the rate of the reaction is defined by value of the temperature of the gas mixture. This model is suitable for laminar flow of fuel and oxidant and when the combustible mixture is previously stirred.
- **Magnussen**: in this model, the rate of combustion is defined by rate of turbulent mixing the fuel and the oxidant. This model is applicable only for turbulent flow of fuel and oxidant that were not preliminary stirred.
- **Arrhenius-Magnussen**: this model is combination of the models **Arrhenius** and **Magnussen**. The **Arrhenius-Magnussen** can be used to simulate combustion in previously stirred mixtures and in non-stirred mixtures. When turbulence is low, this model is similar to the **Arrhenius** model, and when turbulence is large this model is similar to the **Magnussen** model.
- **EDC**: this is the most accurate from combustion models implemented in *FlowVision*. This model allows calculating the combustion in previously stirred mixtures and in non-stirred mixtures.

Compatibility of combustion models with turbulence models

Combustion models **Magnussen**, **Arrhenius-Magnussen** and **EDC** can only be used when a turbulence model is turned on (otherwise the program would compute using the **Arrhenius** combustion model).

The combustion models **Magnussen**, **Arrhenius-Magnussen** and **EDC** are only compatible with k-ε turbulence models (**KES**, **KEAKN**, **KEFV**, **KENL**) and the **SST** turbulence model (in the **SST** model ε is calculated from ω). When **SA** or **Sm** turbulent model is selected, the **Arrhenius** model will be used.

Initiating the combustion

In all combustion models you specify the ignition temperature of the mixture (**T ignition**) in properties of the **Mass transfer** physical process. The combustion reaction is simulated in zones where temperature of the mixture exceeds the specified ignition temperature.

Also, in properties of the **Mass transfer** physical process, you specify the minimal and the maximal values of the oxidizer-fuel equivalence ratio (**Alpha min.** and **Alpha max.**). The combustion reaction is simulated in zones where the value of the oxidizer-fuel equivalence ratio falls within the specified range.

If the computational model includes no heat source, which is capable to heat the gas to the ignition temperature, the combustion can be initiated by an **Ignition/extinction zone** modifier (with **Type= Ignition** specified). Such modifier activates the **Zeldovich** combustion model within the specified geometry **Object** (even if another combustion model is selected for calculating the **Mass transfer** physical process).

Combustion can also be initiated by a **Setting variable** modifier. To do so, you have to set temperature more then the ignition temperature within the specified geometry **Object**.

The **Ignition/extinction zone** modifier with **Type=Extinction** specified blocks simulation of the combustion reaction in the specified volume. Such modifiers are used in zones where combustion in real practice is absent but is simulated in the numerical modes due to such reasons as, for example, simplified representation of the combustion mechanism (brutto-reaction only is simulated).

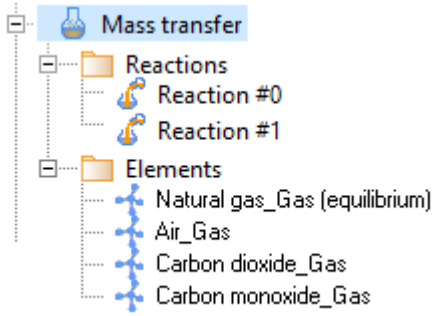
See also: Theory > Physical processes > Mass transfer > Combustion > Equations.

6.10.3.5.3 Chemistry

The **Chemistry** mass transfer model is used to simulate the mass transfer in the form of successive (multistage) chemical reactions. This model can be recommended for use when there are many reactions with kinetics conforming to the Arrhenius law.

The **Chemistry** mass can be used for simulations of complex reactions including multistage combustion of fuel. But the more is the number of equations, the longer will be the time of computation of each iteration, so use of computation of successive reactions should be justifiable due to aims of the simulation.

Each reaction is specified in **Preprocessor** in a separate element in the project tree:



For each chemical reaction you have to specify its constants and also stoichiometric coefficients with a negative sign (minus) for substances that locate on the left side of the chemical formula and with a positive sign (plus) for substances that locate on the right side of the formula.

For dissociation/recombination reactions it is necessary to specify efficiency of «third bodies» («third particles»).

The **Elements** folder contains **Conservative scalars** (**Elements**). Their specifying is not mandatory but this substantially improves the computational stability (convergence). See chapter Theory, section Equations, for details on calculating the coefficients that define the **Elements**.

6.10.3.5.4 Ablation

Simulation of ablation allows the program to calculate the thermochemical process of the ablative mass loss from the surface of a physical body. It is assumed that chemical reactions are going and gas substances (ablative products) are being blown into the gas phase.

Examples of problems with ablation are simulating the thermal ablative protection of space landers (descent vehicles) or reaction chambers of liquid-fuel rocket engines.

Simulating of ablation is only possible when physical processes **Heat transfer** and **Mass transfer** are enabled.

The applied model of ablation is specified in parameters of Mass transfer by the **Ablation** parameter. The following ablation models can be selected:

- **Carcass**, which is a complex model of destruction of porous thermal-protective system (TPS) and mass loss from surface of the TPS. The TPS is simulated as indestructible matrix (silicone matrix or carbon matrix) filled with filling material (phenolic resin). The TPS is exposed to thermal load. Heating the TPS causes destruction of the thermal-protective material (pyrolysis of the resin). Gases, which are generated during the pyrolysis

process, move through pores and blow out from the TPS. On the surface of the TPS, homogeneous and heterogeneous reactions proceed resulting in mass loss (ablation) and motion of the surface.

- **Chemistry**, which allows you to specify multiple simple single-stage chemical reactions for blowing several substances from the body's surface into the computational domain.
- **Boiling**, which assumes that temperature on the surface is specified by the user as a property of the boundary condition (this is the boiling temperature), and the ablation has a single product only. The mass flow here is defined basely by the temperature drop between the wall and the gas flow near the wall.
- **Sublimation**, in which mole fraction of the ablation product on the surface is defined by saturated-vapor pressure, which is function of the wall temperature.

Specifying a surface of Ablation

A surface of ablation can be a wall or a **Conjugated ablation** connected boundary condition.

To set ablation on a wall, you have to specify **Type = Wall, ablation** in properties of the boundary condition.

To set ablation on a "Conjugated ablation" connected boundary condition, you have to create for the connected boundary condition (**Type=Connected**) a **Binder conditions** with **Connection type = Ablation**.

6.10.3.5.5 Mass transfer in dispersed media

This physical process allows the program to take into account changing of mass and size of particles of a dispersed phase due to evaporation, combustion, ablation and condensation.

For dispersed **Phases** of the **Particles** type the following mass transfer models are available:

- **Mass transfer**: general mass transfer model
- **Coal**: this model is used to simulate combustion of coal

Simulating of mass transfer is only possible when the **Phase transfer** physical process is enabled.

For **Phases** of the **Carcass** type the only general mass transfer model is available.

See section Theory > Physical processes > Processes in the presence of dispersed medium.

6.10.3.6 Turbulence

A flow, in which many vortices arise, is called a turbulent flow. When vortices are very small, it is unpractical to simulate them using the computational grid resolution because this consumes very large computational resources (see subsection *Direct numerical simulation (DNS) of turbulence* below).

Due to viscous friction in the fluid vortices the kinetic energy converts to the internal energy. The turbulence models consider losses of kinetic energy without direct simulating the vortices.

In *FlowVision* a turbulent flow is taken into account by introducing additional terms, which specify the turbulent viscosity and turbulent thermal conductivity, into the Navier-Stokes equations. The coefficients of turbulent viscosity μ_t and turbulent heat transfer μ_t/Pr_t are calculated from the solution of additional equations of semi-empirical turbulence models.

The turbulent viscosity in this approach measures the kinetic energy losses due to viscous friction and it is added to the molecular viscosity in the Navier-Stokes equations.The more intensive the turbulence is, the turbulent viscosity is larger.

The following turbulence models are available in *FlowVision*:

- SST (the Shear Stress Transport turbulence model)
- Sm (the algebraic Smagorinsky turbulence model)
- SA (the Spalart-Allmaras turbulence model)
- KES (the standard k-ε turbulence model)
- KEAKN (the Abe, Kondoh, Nagano Low-Re k-ε turbulence model)
- KEFV (the "FlowVision" Low-Re k-ε turbulence model)
- KENL (the nonlinear k-ε turbulence model based on works of E.Baglietto)

Direct numerical simulation (DNS) of turbulence

This approach to simulating the turbulent flows assumes that all vortices is the flow are resolved by the computational grid.

To simulate a turbulent flow using the DNS method, you have to disable a turbulence model in *FlowVision*, because the method requires only the Navier-Stokes equations and a computational grid that will resolve the smallest vortices. Also you have to use a very small time step, which will be able to resolve the life time of any vortex.

The DNS is not convenient for practical application because it requires very large computational resources due to very large number of computational cells and long computations due to the small time step. But this method is applied, for example, to examine the turbulence models.

Simulating turbulent flows using the scheme viscosity

Numerical methods include the so-called scheme viscosity, which appears due to inexact nature of numerical methods. Cumulative errors of the numerical scheme cause losses of accuracy and can cause losses of calculated energy of the flow.

When quite small computational cells are used without applying a turbulence model, this makes the simulated flow to be highly turbulent i.e. there the computation will resolve well its pressure gradients, which generate vortices, which generate new pressure gradients. Real turbulent flows do not have so many middle-scale and large-scale vortices, because their generating is reduced due to energy losses cased by generating small-scale vortices.

But the scheme viscosity can cause losses that are comparable with losses for generating small-scale vortices. So you it is possible to obtain a realistic pattern of a turbulent flow without a turbulence model and without resolving small-scale vortices.

As the source of the scheme viscosity is the quantum nature of the model space and time in numerical methods, limitations of accuracy in machine computations, and other specifics of numerical computing. So it is difficult to forecast the scheme viscosity and accuracy of this approach to simulating turbulent flows.

6.10.3.7 EMHD (electromagnetohydrodynamics)

The **EMHD** physical process allows simulating the electromagnetohydrodynamics.

The following models can be selected:

- **Electrodynamics** for simulating interaction of liquid and electrostatic fields. This model does not take into account the effect of the induced magnetic field.
- **MHD Potential model** for simulating the interaction of liquids and electromagnetic fields.

To simulate **EMHD**, the program solves steady state Maxwell's equations.

The **EMHD** model also allows you to set the voltage drop on the metal-gas (plasma) boundary depending on the current by specifying the **Voltage Drop** parameter on the connected BC **Conjugation for potential**.

See also: Theory > Physical processes > Electromagnetohydrodynamics.

6.10.4 Models

Prepared computational models are presented in the program by elements **Model #N** in the folder **Models** in the project tree.

The **Model #N** elements are used for referring to computational models that are used in **Subregions**. In every **Model #N** the following parameters for computation are set:

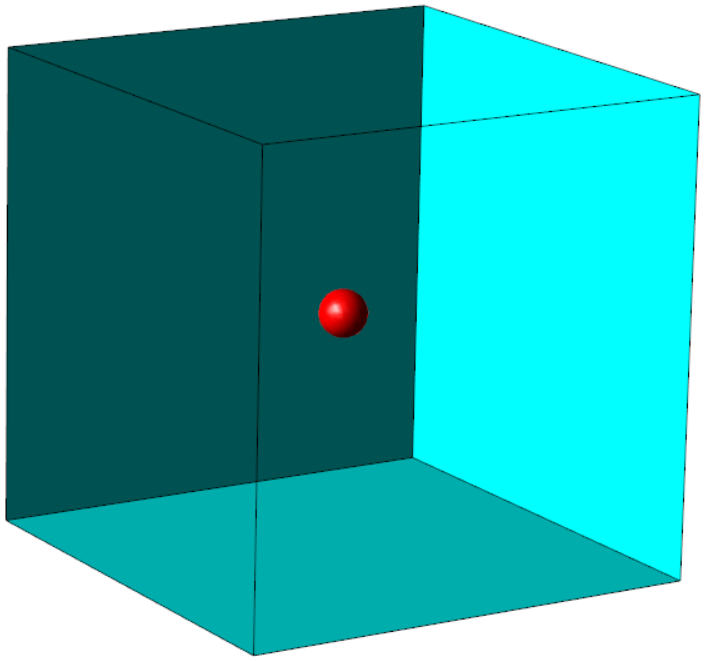
- Simulated **Phases** (a **Model** may contain one or two **Phases**)
- Parameters of **Phase interaction** (when two **Phases** are presented in the **Model**). Also here you specify use of special models such as the cavitation model).
- **Initial data**
- Use of the **Gap model** (for analytical solution of flow in gaps without resolution the gaps by a computational grid).

To apply a **Model #N** computational model, with all **Substances** and **Phases** that are set in it, in some computational **Subregion**, you have to specify **Model = Model #N** in properties of this **Subregion**.

When you need to exclude volume of some **Subregion** from the computation, you mustn't specify a **Model** in this **Subregion**, so **Model = (none)** will remain in properties of the **Subregion**.

Example: subsonic flow around a sphere by air

In this example the computational region consists of two enclosed volumes – subregions **Cube** and **Sphere**:



In the **Subregion Cube** we set a **Model** with one **Phase**, which contains one **Substance** (**Air**) and has specified **Physical processes** (**Motion** and **Turbulence**). Also we set in the **Cube**, on its boundary with the **Sphere**, we specify a **Wall** boundary condition.

Within the **Subregion Sphere** there are no physical processes that are interesting to us in this problem setting. So we specify this volume as non-computational. To do this, we don't change **Model = (none)** in properties of the **Sphere**, thus no computational model is set in the **Sphere**. So also no any boundary and initial conditions will be set in the **Sphere** and no computations will be done in the **Sphere**.

6.10.4.1 Gap model

The *gap model* is an analytical model, which provides a solution in thin gaps between boundary conditions.

This model allows to spend less resources due to the computational grid is not applied in the gaps. Use of the gap model allows the program not to resolve the gap by the grid along the normal to gap-forming surfaces.

FlowVision automatically detects gap computational cells that locate between close boundary conditions. Then flows in the gaps of any shape are approximated as flows in flat gaps (see details in the section Theory > Physical processes > Processes in clearance).

The program doesn't apply the gap model if less then two computational cells locate between the gap-forming surfaces.

To visualize the gap cells, use the **Cell set** layer and specify in its properties **Type = Gap**.

To enable the gap model, specify **Use Gap model = Standard Gap model** in properties of the element **Models > Model #N**. Parameters of the gap model (**Min. clearance** and **Max. clearance**) are specified in properties of the element **Models > Model #N > Standard Gap model**.

FlowVision computes the following physical processes in gases, fluids and solid bodies:

- **Motion**
- **Radiation**
- **Heat transfer**

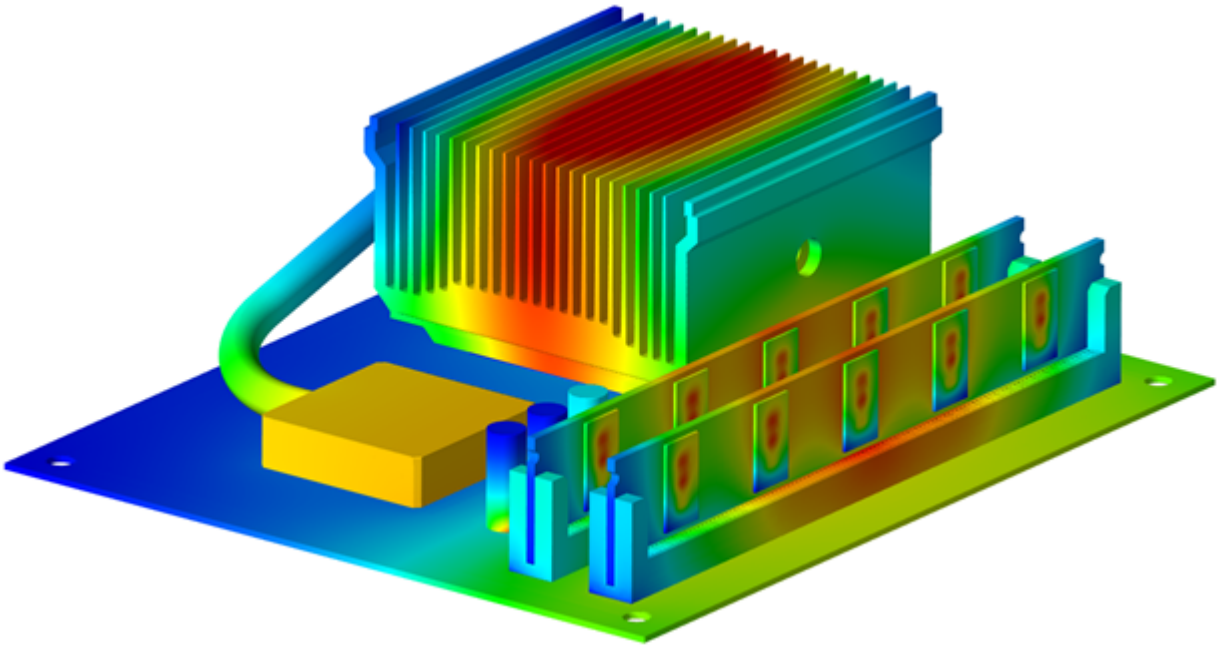
For the correct operation of the gap model, the gap cells must be of the same level of adaptation. If in the gap there are cells with different adaptation level, then the program will automatically apply adaptation in the gap so all cells in the gap will have the same (the maximal) adaptation level.

Applying gap cells to calculate heat transfer in assemblies

Simulations of heat transfer, for example in electrical assemblies, is carried out in the following steps:

1. Batch import of geometry objects is done to create the computational domain. At this import the surfaces are shifted equidistantly to avoid contacts between parts of the assembly (gaps are created).

2. When boundary conditions are being set on all surfaces of the assembly, set connected boundary conditions for conjugated heat transfer.
3. Due to applying the gap model, it is not required to resolve gaps between parts of the assembly by a grid. But the gap model has also another functionality in such simulations: it is possible to change thermophysical properties of the **Phase** in gap cells by tuning parameters of the connected boundary conditions that correspond to the contact pairs. You specify gap heat transfer coefficients on the contact pairs (parameter **Gap heat-transfer coef.** in properties of the boundary condition). These values allows you to change heat transfer coefficients of air gaps between parts of the assembly so you can set individual heat transfer properties for different contact pairs.



Effective viscosity in the gap

In the above mentioned example the program *does* simulate motion of the air in gaps. This can cause convective heat losses in contact pairs in the simulation while convection is actually absent in reality.

Using the **Effective viscosity in gap** parameter (in properties of the boundary condition) you can set large viscosity in the **Phase** in the gap and so radically reduce mobility of the fluid.

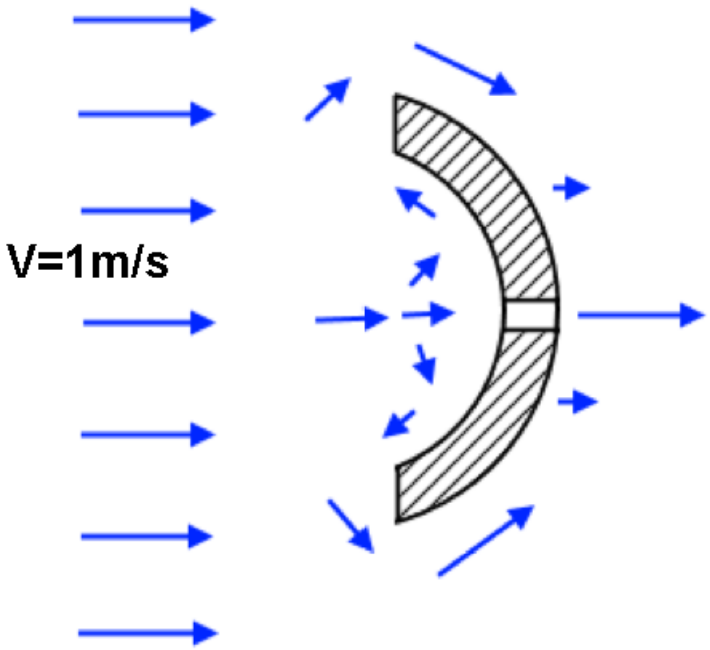
Limitations and specifics of use of the gap model

The gap model predicts well the pressure drop in the gap under the following conditions:

- A laminar flow forms in the gap. That means that over the gap's thickness velocities of the real flow are directed along walls of the gap.
- The gap is resolved by its length at least by 4 cells (the more is number of cells along the gap, the more accurate is the computation, but good integral results of the simulation often doesn't require high accuracy of the solution in the gap).

The gap model might be useful also in cases when the above mentioned conditions are not met.

Let's take a look on simulating of flow around thick-walled semisphere with an aperture, as shown on the illustration below:



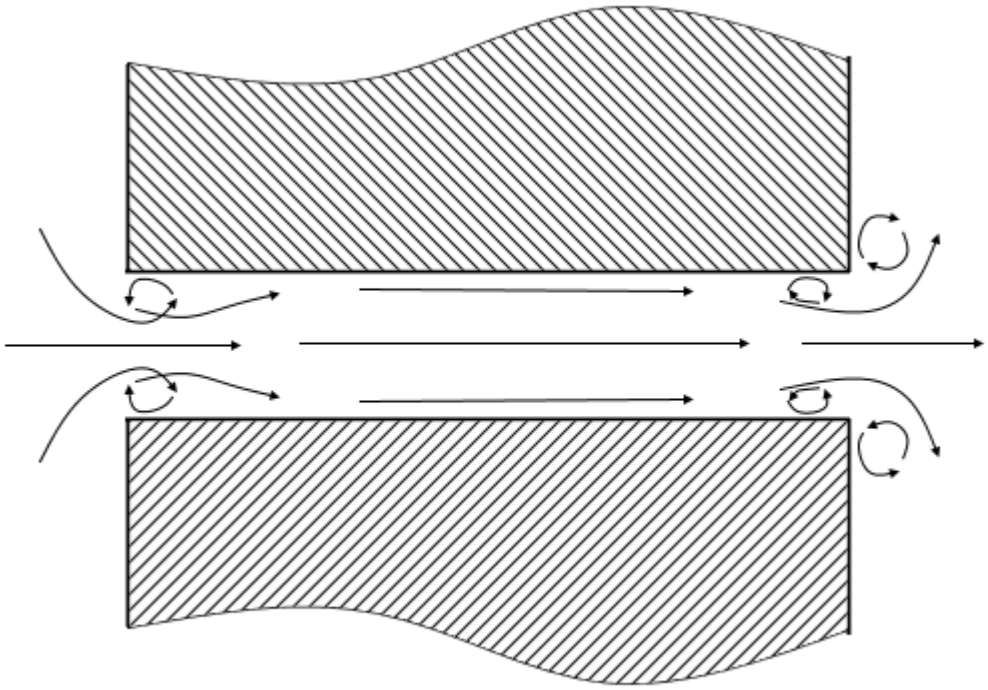
The sought quantity in this problem is the force from the gas flow acting on the semisphere.

When the aperture is very small relating to the size of the semisphere, it is possible to neglect loss of the pressure caused due to flow of the gas through the aperture. Influence of the aperture on the force applied by the gas on the surface of the semisphere will be minimal.

But on the left and on the right from the semisphere there will be high pressure drop connected by the aperture. And, if the channel formed by the aperture is not resolved by the computational grid, the solution will not converge due to too high pressure drop within one cell.

When we enable the gap model in the aperture, the solution will successfully converge because of simple analytical solution in the gap. The gap's will contain one or two cells along it length, and flow in the gap might be turbulent; moreover, at the inlet and at the outlet of the gap losses caused by vortex formation (flow at the ends will be not laminar). But accuracy of the pressure drop on the aperture doesn't matter because of negligible influence of this geometry detail for the integral solution.

You shouldn't use the gap model when the gap itself is the main subjects for study. For example, when you have to calculate the pressure drop in a channel with pressure losses due to vortex formation at the channel's inlet and outlet as shown on the illustration below:



The gap model is not able to take these edge effects into account as it assumes that the flow is laminar in all gap cells and velocity vectors are directed strictly along the gap-forming surfaces.

Providing adaptations in a gap

For the correct operation of the gap model, the gap cells must be of the same level of adaptation.

When the **Adapt through gap** setting is enabled, if the gap contains cells with different adaptation levels, then automatic adaptation of all cells in the the gap is applied up to the same (the maximal) adaptation level of all cells in the gap.

See also

Theory > Physical processes > Processes in clearance

6.11 Boundary conditions

Boundary conditions are used to specify parameters of medium located on the boundary of the computational domain, as well as to bind two different volumes.

Types of boundaries and boundary conditions

Boundary conditions arise on real existing boundaries (for example, on the surface of a body in a gas flow) or on artificially drawn boundaries that separate the computational domain from the rest of the environment.

In the latter case, the boundary is set so as to satisfy two opposing requirements:

- to minimize errors caused by the appearance of a boundary in the simulated processes, on one hand
- to minimize the computational domain for accelerating computations, on the other hand

The complexity of this task predetermines the variety of boundary conditions that may be set on artificially drawn boundaries.

*FlowVision*includes the following types of boundaries and boundary conditions *(templates):*

- **Wall**: conditions on boundaries of solid body - liquid, solid body - gas, solid body - solid body
- **Symmetry**: boundary conditions on a plane of symmetry; they can also be set on a hard surface, on which conditions of impermeability and slipping for the **Velocity** variable are set (**Zero gradient** conditions are set for other variables).
- **Inlet/Outlet**: Boundary conditions setting a convective flow through a surface by indicating flow velocity or pressure (full pressure)
- **Free outlet** specifies a surface through which the flow exits from the computational domain (and inflow of the medium into the computational domain is also available)
- **Non-reflecting** are boundary conditions based on the assumption that the equations relative to all variables on the boundary assume the form of a wave equation.
- **Wall, ablation** simulates ablation at a surface
- **Wall, film** simulates crystallization of the dispersed phase's substance on a surface of the geometry model and/or the solid phase. This boundary condition is used to simulate aircraft icing.
- **Connected** are the boundary conditions that are set for simulation of a physical process, which is *common for two subregions* separated by a boundary. There are the following types of connected boundary conditions (boundary links):
  - Conjugate all variables
  - Conjugate temperature
  - Periodic surface
  - Sliding surface
  - Conjugated ablation
  - Conjugated electrical potential
  - Conjugated Maxwell equations

Forms of boundary conditions

Most boundary conditions are limited to two principal forms of boundary conditions for a vector variable ($\vec{V}$) or a scalar variable P, k, ε,ν_t, T, Yi, gF, F:

- the following is set on the boundary or in the center of a boundary cell:
  - scalar variable value
  - vector variable value
  - value of the vector variable component normal to the boundary
- the following is set on the boundary:
  - flow value of the scalar variable
  - flow value of the vector variable component normal to the boundary

Variables for which boundary conditions are set

Boundary conditions are set for the principal computed variables, the set of which is defined by the model.

For the **Movement** model, the boundary conditions define the pressure or velocity at the boundary; the boundary condition for the principal variable can be defined by:

- normal mass velocity
- full pressure

- static pressure

For the **Turbulence** model, the boundary conditions define the principal turbulent variables at the boundary (which depend on the particular turbulence model):

- for energy of turbulent pulsations k and dissipation speed ε (when using all turbulence models except **SA**)
- for turbulent viscosity factor ν_t (when using the **SA** turbulence model)

For the **Heat transfer** model,the boundary conditions define the temperature at the boundary. Boundary conditions are set for temperature T, which can be defined by:

- heat flow
- temperature value
- etc.

In cases where the principal variable in the heat transfer model equation is enthalpy or total enthalpy, the conversion of boundary conditions from temperature to the principal variable is performed by the software automatically.

For the **Mass transfer** model, boundary conditions define the mass fractions of used **Substances** (see details in the sections Theory > Physical processes > Mass transfer > Mixing > Boundary conditions, Theory > Physical processes > Mass transfer > Chemistry > Boundary conditions, Theory > Physical processes > Mass transfer > Combustion > Boundary conditions and Theory > Physical processes > Mass transfer > Ablation > Boundary conditions).

For the **Phase transfer** model, boundary conditions are set for concentration of phase F.

For the **Radiation** model, boundary conditions are set for the radiation flux. Boundary conditions are set by:

- radiation flux value
- radiation factor (emissivity coefficient)

Setting boundary conditions in the Pre-Postprocessor

Generally, setting boundary conditions is performed in two stages:

Step	Actions
1	Creating **Boundary condition** elements in the project and defining their parameters
2	Assigning **Boundary conditions** to groups of facets

If automatic facet grouping is performed when loading the geometry model of the computational domain, a boundary condition will be created on each group of facets on its surface. By default, all boundary conditions are of the **Wall** type.

In this case, setting boundary conditions in the project is limited to redefinition by the user of the boundary conditions, which were automatically created during loading loading of geometry modes.

See also:

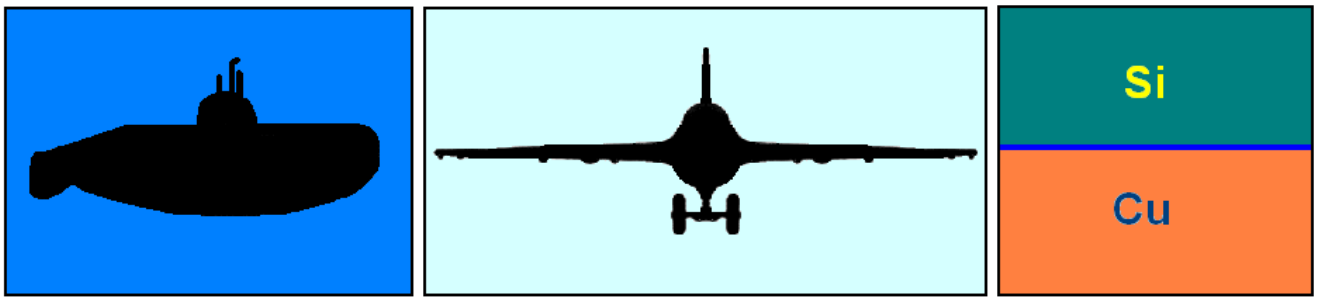
- subsection *Properties of element «B.Cond. #N»* in the section Folder «Boundary conditions»
- chapter Theory

6.11.1 Boundary conditions «Wall»

Boundary conditions **Wall** are used to simulate at the boundaries between:

- solid and liquid
- solid and gas
- solid and solid

The boundary condition **Wall**, in its physical meaning, is a solid surface on which the conditions of non-flow and adherence are fulfilled for the variable **Velocity**. For other variables the conditions are determined by physics of the simulated processes.

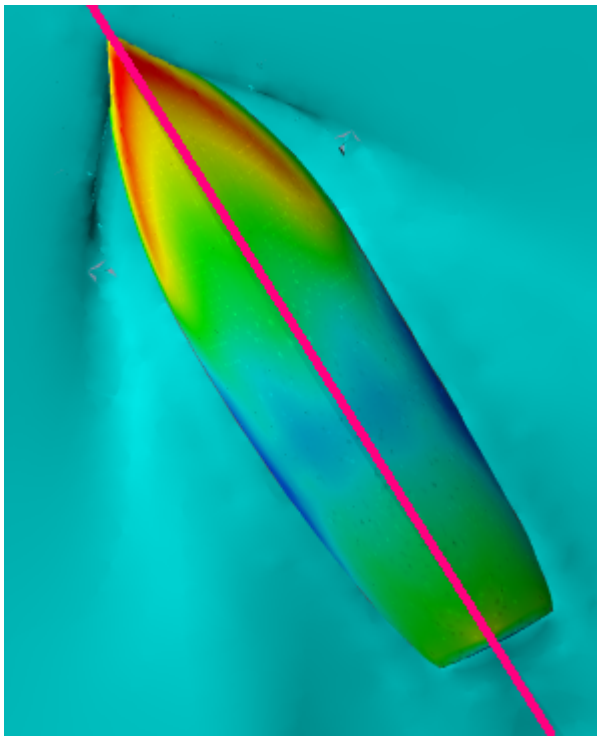


Boundary condition **Wall** is used to simulate the boundaries:
"solid - liquid", "solid - gas" and "solid - solid"

You can configure multiple parameters, including:

- wall functions for turbulent parameters of the specified **Phase**
- surface roughness (size of microroughnesses on a rough wall)
- contact angle of wetting the wall's surface by a liquid
- effective viscosity in the gap
- the heat transfer coefficient of the wall
- the heat transfer coefficient of the gap
- conditions applied to the variables on the wall or near the wall
- simulate a moving belt like a chain track (see **Track**)

6.11.2 Boundary conditions «Symmetry»



The **Symmetry** boundary condition is set on a plane of symmetry; also it can be set on a hard surface, for which conditions of impermeability and slip conditions for the **Velocity** variable are complied (for other variables the **Zero gradient** conditions are applied).

6.11.3 Boundary conditions «Inlet/outlet»

Boundary conditions **Inlet/Outlet** define a convective flow through the boundary, with setting of the following conditions:

- For flow velocity (**Velocity**) you can select one of the following options (see section Theory > Physical processes > Movement > Boundary conditions > Inlet/outlet):
  - **Normal mass velocity**
  - **Normal velocity with pressure**

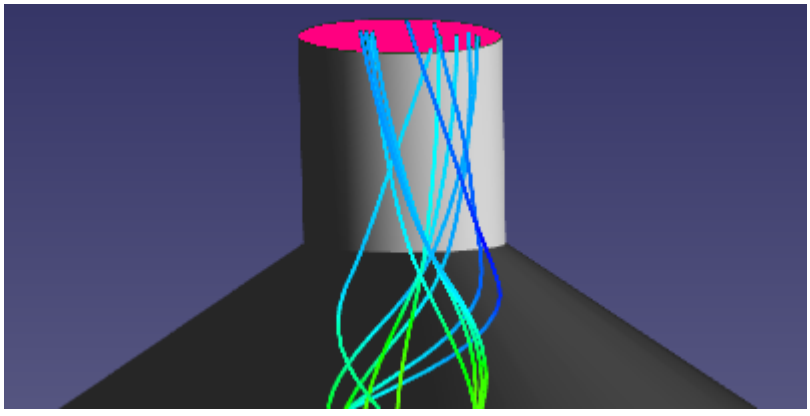
  - **Velocity with pressure** (when selecting this option, the indicated velocity vector must have a normal component directed inside the computational domain). If it is necessary to set a velocity tangential to the boundary, it is recommended to set a minimal normal component (for example, 10^{-6} m/s.)
  - **Inlet pressure**
  - **Total pressure**
  - **Total pressure and velocity direction** (the user specifies the direction of the velocity's vector). See section Theory > Physical processes > Motion > Boundary conditions > Template 'Inlet/Outlet' for details.
  - **Fixed velocity**
  - **Supersonic inlet**

- For turbulent energy (**TurbEnergy**, k) you can select one of the following options (see section Theory > Physical processes > Turbulence > Boundary conditions > Inlet/outlet):
  - **Pulsations**
  - **Value**
- For turbulent dissipation (**TurbDissipation**, ε) or specific turbulent dissipation (**TurbDissipation specific**, ω) you can select one of the following options (see section Theory > Physical processes > Turbulence > Boundary conditions > Inlet/outlet):
  - **Turbulence scale**
  - **Value**
- For turbulent kinetic viscosity (**TurbKinViscosity**) you can only set the value (see section Theory > Physical processes > Turbulence > Boundary conditions > Inlet/outlet).
- For **Temperature** you can select one of the following options (see section Theory > Physical processes > Heat transfer > Boundary conditions > Inlet/outlet):
  - **Total temperature**
  - **Value**
- For **Variance of fuel** and **Mass fraction** you can select one of the following options:
  - **Value at the inlet**
  - **Zero gradient**
- For **Radiation density**, you can set only the **Radiation flux density** (see section Theory > Physical processes > Radiation > P1 > Boundary conditions > Inlet/outlet, Free outlet, Non-reflecting).

Boundary conditions for variables of the dispersed phase on a real physical outlet

On a surface corresponding to a real physical outlet, for all the variables of the dispersed phase (for example, **Velocity (disp)** or **Phase volume**), specify boundary conditions **Permeable surface** (select them from a drop-down list in the **Properties** window of element **Subregions > SubRegion #N > Boundary conditions > B.Cond. #N** in parameters **Variables >** ***(name of variable)***).

6.11.4 Boundary conditions «Free outlet»

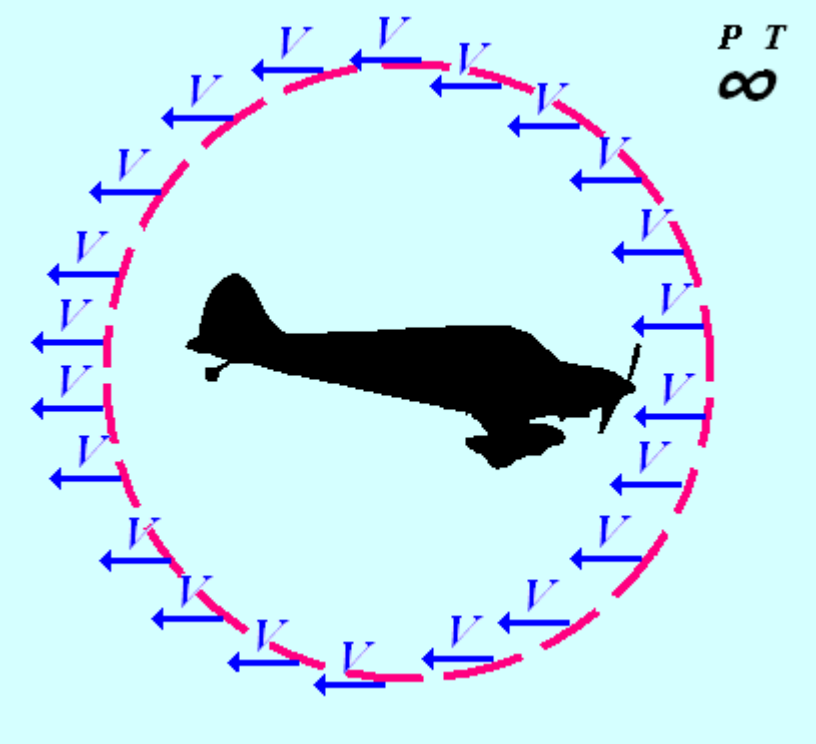


The **Free outlet** boundary conditions set a surface, through which flow exits out of the computational domain.

The **Free outlet** BC has the following properties:

- when the medium moves outside the computational domain, zero gradient is used for all variables (except pressure) along the normal to the boundary
- this boundary condition *allows incoming flow* of the medium into the computational domain
- when the medium moves inside the computational domain, the values or boundary conditions for the quantities, which were specified by the user, are applied.

6.11.5 Boundary conditions «Non-reflecting»



The **Non-reflecting** boundary condition suggests that equations equation in all variables on the boundary take the form of the wave equation.

This BC is set on the outer surface in simulations of external flows. These boundary conditions take into account spread of information outside of the computational domain.

To simulate motion, the user specifies x, y, z components of the velocity at infinity and the static pressure at infinity.

To simulate heat transfer, the user specifies the value of the static temperature at infinity.

The **Non-reflecting** BC causes unstable and incorrect solution in the case if convective CFL is more then 1 in cells that are adjacent to the BC (this requirement relates to the CFL, which is calculated by size of the boundary cells and and boundary normal velocity).

Before starting the computation, you can evaluate the time step, which corresponds to the near-boundary CFL=1:

$$t_w = h_{min} / V_\infty$$

where:

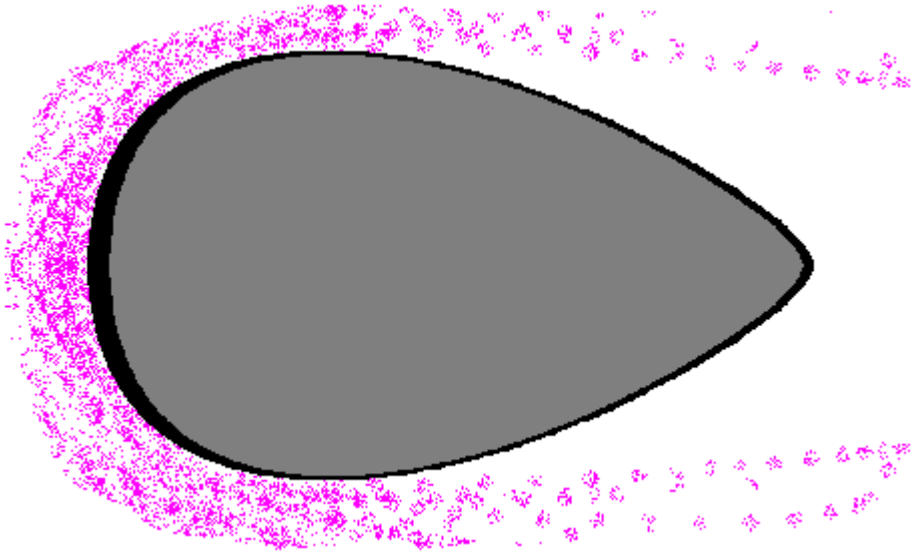
h_{min} – minimal size of the boundary cells,

V_∞ – value of the normal velocity on the boundary condition.

Stable computation requires that the value t_w must not be less then the time step of the computation.

See details in the chapter Theory.

6.11.6 Boundary conditions «Wall, ablation»



Boundary conditions **Wall, ablation** are used for simulation of ablation (thermochemical process of mass loss from a surface).

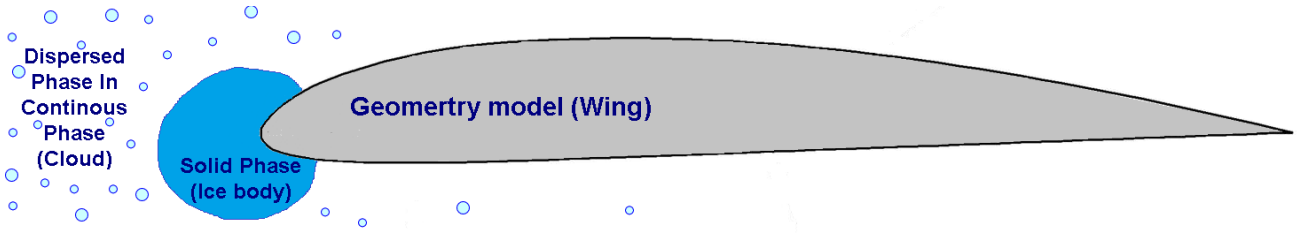
These boundary conditions can only be set when simulating of ablation is specified in the project.

When the **Wall, ablation** boundary conditions are selected, the program automatically set boundary conditions **Ablation** for **Temperature**, **Velocity** and **Mass fractions** of substances.

See also: details in the chapter Theory (section Template 'Wall, ablation').

You cannot set a boundary layer grid on the boundary condition **Wall, ablation**.

6.11.7 Boundary conditions «Wall, film»



Boundary conditions **Wall, ablation** are used for simulation of crystallization of the dispersed phase's substance on the surface of the geometry model and/or solid phase. These boundary conditions are applied to simulate aircraft icing.

For such simulations you have also set some model of **Crystallization** in physical processes of the dispersed **Phase**.

See also: details in the chapter Theory (section Crystallization).

Use of the boundary layer grid is *not* recommended on boundary conditions **Wall, ablation**.

6.11.8 Connected boundary conditons (boundary links)

Connected boundary conditions (boundary links)can be set at the *boundary between two* subregions.

Boundary conditions of that type allow the linking of two subregions by coordination of variables at the common boundary of the two subregions: values of the principal variables on the boundary of one subregion must be equal to the values of the principal variables on the boundary of the other subregion.

The connected boundary conditions have the following specifics:

- unlike usual boundary conditions, they are coordinated (identical) at the boundary between subregions
- they are set once as a pair of boundary conditions (once for two different sides of the boundary between subregions)

The following types of connected boundary conditions can be set in *FlowVision:*

- *conjugate all variables*, when boundary conditions are linked for all variables that are calculated in models and are common for models in the neighboring subregions (that is, values of those variables in neighboring subregions are identical at the boundary surface)
- *conjugate temperature* (conjugated heat exchange), when only the boundary conditions for temperature andheatflux are linked (the values of those variables in neighboring subregions are identical at the boundary surface)
- *periodic surface* is the boundary between repeating identical fragments, which permits the project to be limited to a single fragment, saving computational resources
- *sliding surface* is linking performed for all variables taking into account the rotation of one **Subregion** relative to another **Subregion**. Sliding surface is permitted only for the connection of one **Model** and the boundaries obtained from separating a subregion by a **Sliding surface**.
- *ablation* simulates thermochemical process of ablation at a surface that separates subregions.
- *conjugated electrical potential*, which can also simulate the voltage drop on the boundary
- *conjugated Maxwell equations* to simulate equations of electromagnetohydrodynamics
- *user boundary conditions* implemented by **API Binder** modules

In problem settings with a free surface, it is not allowed to simulate intersections of the free surface and connected boundary conditions of the following types:

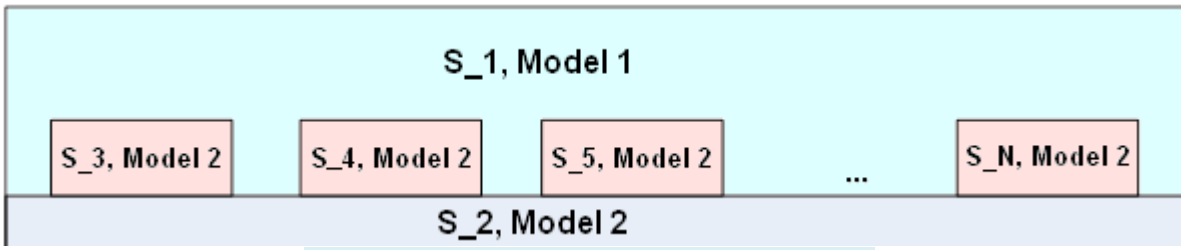
- **Conjugate all variables**
- **Periodic surface**
- **Sliding surface**

Examples of the use of connected boundary conditions

Modeling the air cooling of a circuit board with working heat-producing chips mounted on it can be an example of a project using boundary links (see illustration below).

The computational domain in this project is divided into the following **Subregions**:

- subregion **S_1** (moving air) for which **Model 1** is set (physical processes: **Movement**, **Heat transfer**)
- subregion **S_2** (circuit board) for which **Model 2** is set (physical process: **Heat transfer**)
- subregions **S_3**, **S_4**, ... , **S_N** (the chips), for which **Model 2** is set (physical process: **Heat transfer**)



Simulating air cooling of a circuit board with chips

Setting the project's initial data regarding the connected boundary conditions consists of several steps, after which the structure shown in illustration below is formed:

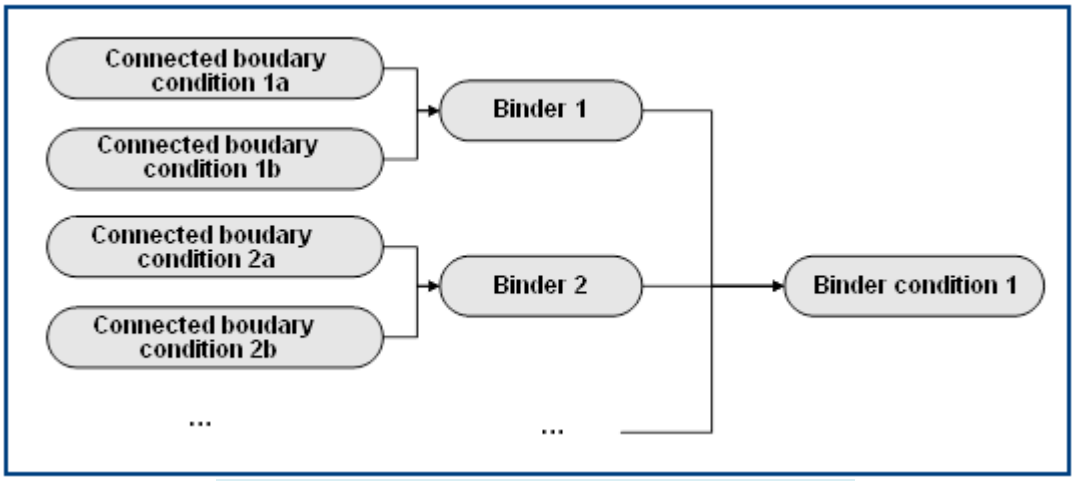


Diagram of connected boundary conditions organization

At the beginning, steps of the *preliminary stage* are performed:

1. Splitting geometry models of computational subregions into groups of facets, through which the conjugated heat transfer flows between different subregions.
2. Setting boundary conditions of the **Connected** type at each subregion. There must be as many such boundary conditions as how many connections are there. For example, two boundary conditions must be set in our example with chips for subregion **S_2** (the circuit board):
   - one for the boundary with the air (subregion **S_1**)
   - another for the boundary with chips (subregions **S_3**, **S_4**, **S_5**, ..., **S_N**)
3. Assigning boundary conditions to groups of facets.

Groups of facets, on which one set of connected boundary conditions is set in a **Subregion**, must be the same as groups of facets in the other **Subregion** (on which also one set of connected boundary conditions is placed).

In other words, the following situations are not allowed:

- a boundary condition of the **Connected** type is set on one side of a group of facets and a boundary condition of another type is set on the other side
- a boundary condition of the **Connected** type is set on one side of a group of facets and two or more boundary conditions of the **Connected** type are set on the other side
- a boundary condition of the **Connected** type is set on one side of a group of facets with surface area that differs from surface area of the group of facets with boundary condition of the **Connected** type on the other side

As a result of the preliminary stage, the program creates a list of the connected boundary conditions, which have *not been united* into pairs yet and on which the model variables to be connected are not indicated. This list corresponds to the first column in the diagram (see illustration above), that contains pairs of elements "**Connected boundary condition Na**" and "**Connected boundary condition Nb**" where N = 1, 2 ...

At the *main stage* the following steps are done:

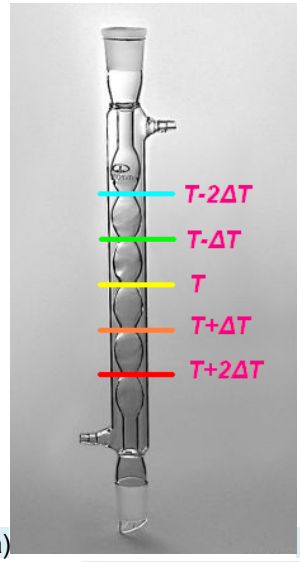
4. The connected boundary conditions are combined into pairs (binders), thus geometric linking of groups of facets from the two sides of boundaries separating the different subregions is done. On the illustration those binders are displayed in the middle column (elements **Binder1**, **Binder2**, ...). In the example with chips, there will be only three such binders - binder **A** at the *air-chip* boundary, binder **B** on the *air-board* boundary and binder **C** on the *board-chip* boundary.
5. Binder conditions are created with the following parameters defined:
   - linked models, that is, two models defined at different sides of the boundary surface
   - linking type - **Conjugate temperature** or **Conjugate all** variables. The choice of linking type actually indicates the type of variables to be linked: temperature only or all variables.

   On the illustration the binder conditions are displayed in the right column (element "**Binder condition 1**"). In our example with chips there will be two binder conditions - one for binders **A** and **B**, and another for binder **C**.
6. Binders are added to the created binder conditions, thus the geometric connection of surfaces achieved by binders is supplemented with binding by variables indicated during the creation of binder conditions.

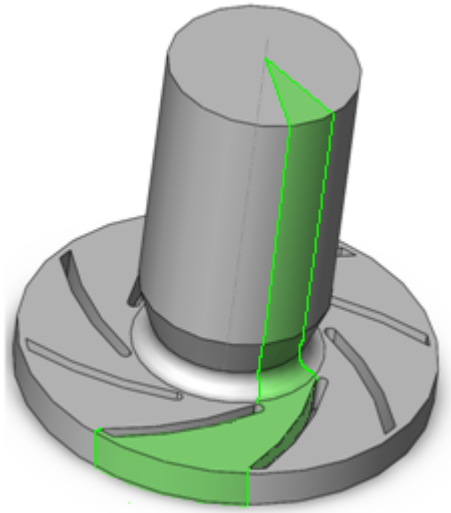
Connected boundary conditions "Periodic surface"

Periodic surface separates boundaries of repeating identical fragments (can be the same differential pressure and/or temperature).

Such boundary conditions conveniently used in the simulation of multiple-exchangers and fragments machine comprising a rotor rotating inside the stator (for simulation of only one of several identical sectors rotor).



a)



b)

Examples of use of the **Periodic surface** boundary condition
a) heat exchanger of several identical sections; b) a rotor that consists of several identical sectors

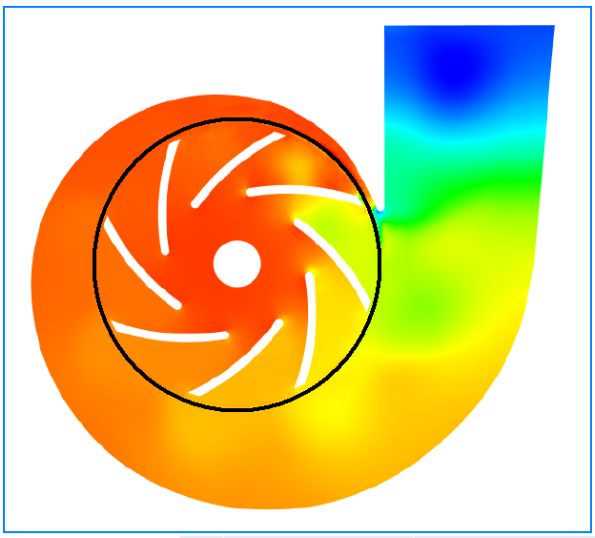
See details in sections:

- Periodic surface (Details about this boundary condition)
- Operations with boundary links (Step-by-step procedure for creating a periodic surface)
- Folder «Boundary links»

Connected boundary conditions "Sliding surface"

Connection is done *over all the variables* and with taking into account *rotation* of one **Subregion** relatively to the other one.

It is convenient to use such boundary conditions in simulations of machines that have a rotor that rotates within a stator (for example, centrifugal compressors or turbines).



An example of use the **Sliding surface** boundary condition

See details in sections:

- Sliding surface (Details about this boundary condition)
- Operations with boundary links (Step-by-step procedure for creating a sliding surface)

Applicability of connected boundary conditions

Application-related boundary conditions are allowed under the following conditions:

Type of a connected boundary condition	The condition of applicability
Conjugate temperature	These can be set on different sides of a surface that separates two different **Subregions**
Conjugate all variables	
Sliding surface	
Periodic surface	Can be set to the same shape surfaces partially bounding the same subregion, and these surfaces must be of the same size and shape and they must be well defined *snap points*.

FlowVision, in its user interface (in selection fields and in lists), does not display connected boundary conditions and **Binders** (pairs of matched connected boundary conditions), in accordance to the above table.

It should however be noted that the choice of boundary conditions, and binders in the forms and lists does not guarantee the validity of some variants (incorrect choices lead to miscalculation of the project or to the impossibility of calculating the project). Therefore, specify formation of **Binders** (from pairs of boundary conditions), as well as matching of **Binders** and **Binder conditions**, meaningfully.

The program's behavior if the Model is changed or removed

If the Model is changed

When the user attempts to change a model in a subregion with connected BCs, the program outputs a warning informing the user that appropriate binders would lose binder conditions and requests the user to confirm his/her decision (**With this model change the following binders will lose their binding conditions:... Continue with the model change?**). After the user's positive answer all that binders would be removed from lists of their binder conditions and would be indicated in the project tree with the "**!**" sign.

If the Model is removed

When the user attempts to remove a model (selecting the "**(none)**" value as the model) in a subregion with connected BCs, the program outputs a warning informing the user that appropriate binders would be removed and requests the user to confirm his/her decision (**With this model change the following binders will be deleted:... Continue with the model change?**). After the user's positive answer all that binders would be

removed and connected BCs in the subregion without a **Model** would be BCs of the **Wall** type. The linked BCs in the other subregion would be indicated in the project tree with the "**!**" sign.

See also sections:

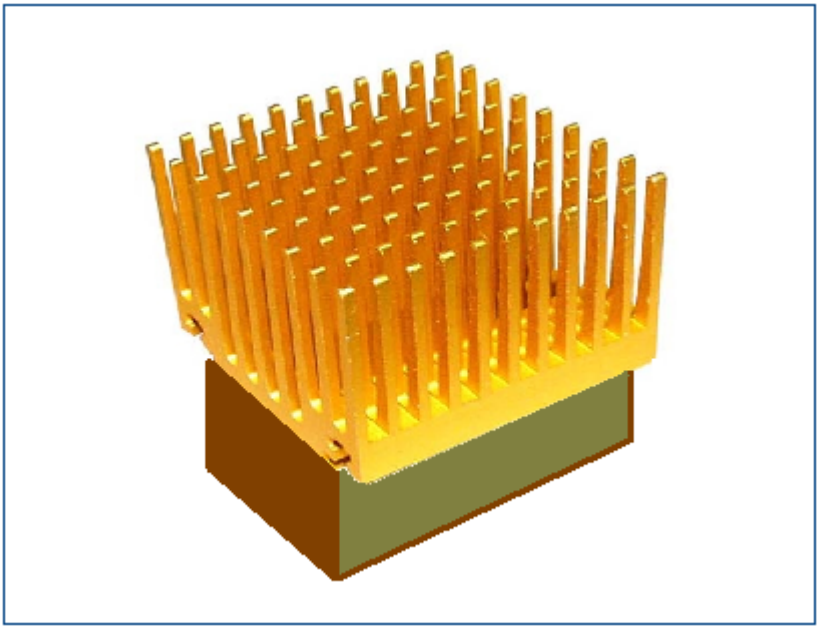
- Periodic surface
- Sliding surface
- Folder «Sliding surfaces»
- Folder «Boundary links»
- Operations with boundary links

6.11.8.1 Conjugate all variables

The **Conjugate all variables** connected boundary condition binds on the boundary between subregions all the variables that are common to models in the adjacent subregions (i.e., the values of these variables in the adjacent subregions are equal on the boundary's surface).

For these variables the **Conjugate all variables** boundary condition is similar to an ideal completely transparent membrane that affects neither the flow nor heat and mass transfer. For each variable continuity of it and its fluxes is provided.

6.11.8.2 Conjugate temperature



When using a connected **Conjugate temperature** boundary condition (conjugate heat transfer), only boundary conditions for temperature and for heat flux are bound (values of these parameters in adjacent subregions are equal on the surface of the boundary).

For the **Temperature** variable, continuity of temperature and heat flux through the surface is provided.

For all other variables, the same boundary conditions as for the **Wall** boundary condition are applied (and, if necessary, the same parameters as for the **Wall** are tuned: **Roughness**, **Wetting angle**, etc.).

The **Conjugate temperature** boundary condition represents a thin wall that is absolutely transparent for heat.

In problems with conjugate heat transfer between a solid and a fluid, a non-convergence of the computation can occur, if a small time step is set when the fluid is not moving and the parameter Advanced settings **> Numerical method > Time integration > Method** parameter is specified as **Steady-state**.

See an example in section Connected boundary conditions (boundary links).

6.11.8.3 Periodic surface

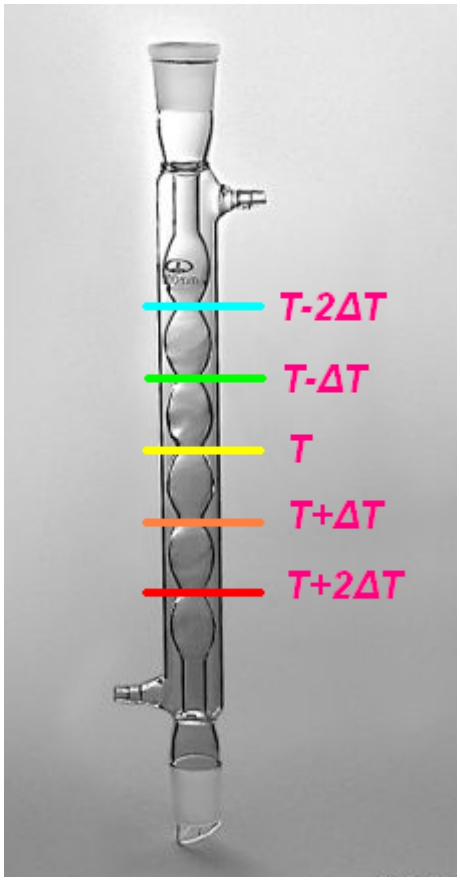
Periodic surface shares borders repeating identical fragments, just as some of the fragments formed pattern:

Examples of repeating units

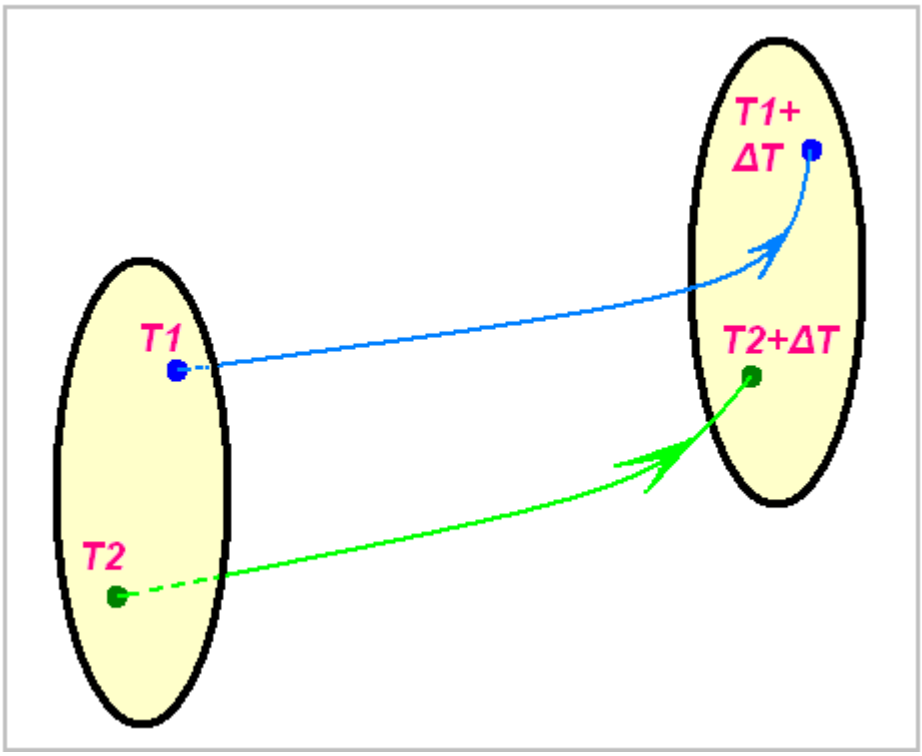
Using **periodic surface** allows *not perform* the calculation for all the tracks, and confine ourselves to a single fragment, thus saving computational resources.

Combination of periodic and sliding surfaces can be used to simulate problems in the sector-sliding setting.

All variables (except the pressure and/or temperature) are transferred from one **surface** to another of the **Periodic** unchanged. When transferring the pressure and/or temperature may be added or subtracted pressure drop and/or temperature between the fragments.



On periodic surfaces can be specified temperature and/or pressure

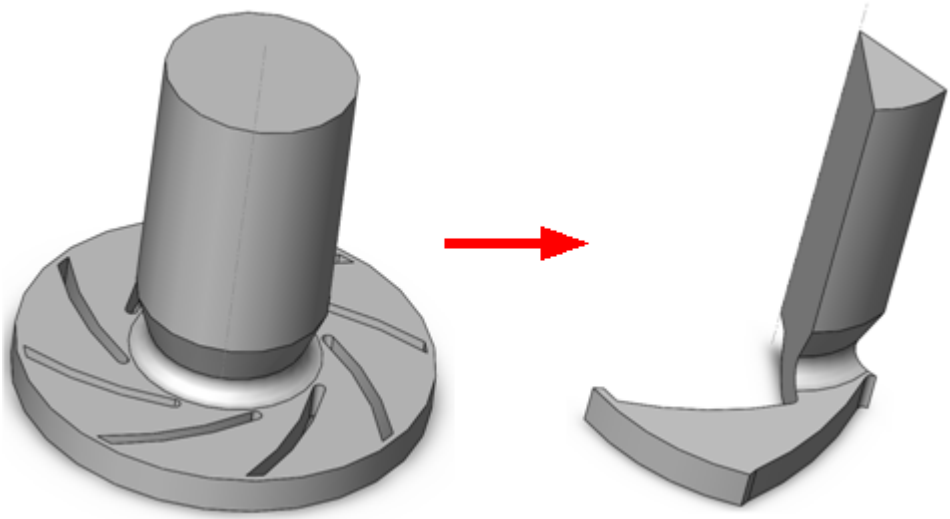


Scheme of the temperature drop to add maps the point of periodic surfaces (for the pressure - the same way)

The values of temperature and/or pressure are set in parameters of the variables **Temperature (Phase #N)** and **Velocity (Phase #N)**, which are elements of the folder **Boundary links > Binder conditions > Binder condition #N > Variables > Connected**.

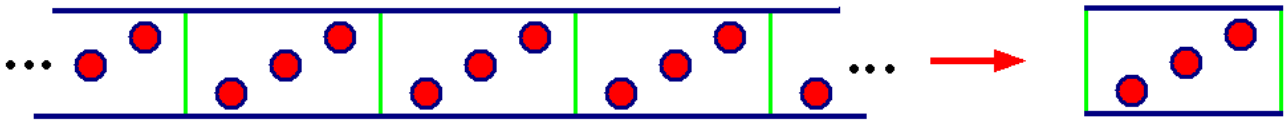
Examples of problems with the use of periodic surfaces

Example 1. The rotor of a turbomachine



In the geometry model of the rotor is allocated one of the rotor. The calculation is carried out for one sector, as other sectors - are identical.

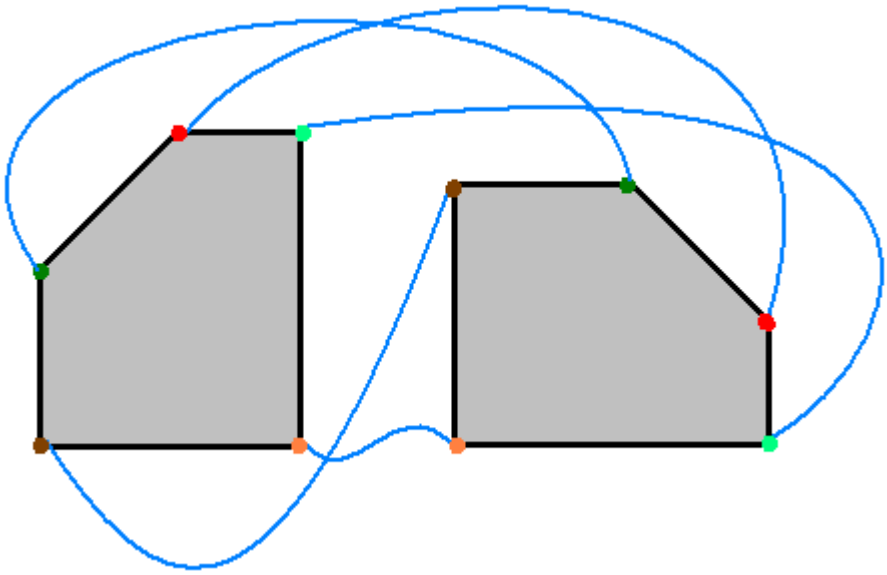
Example 2. Heat exchanger with a large number of identical sections



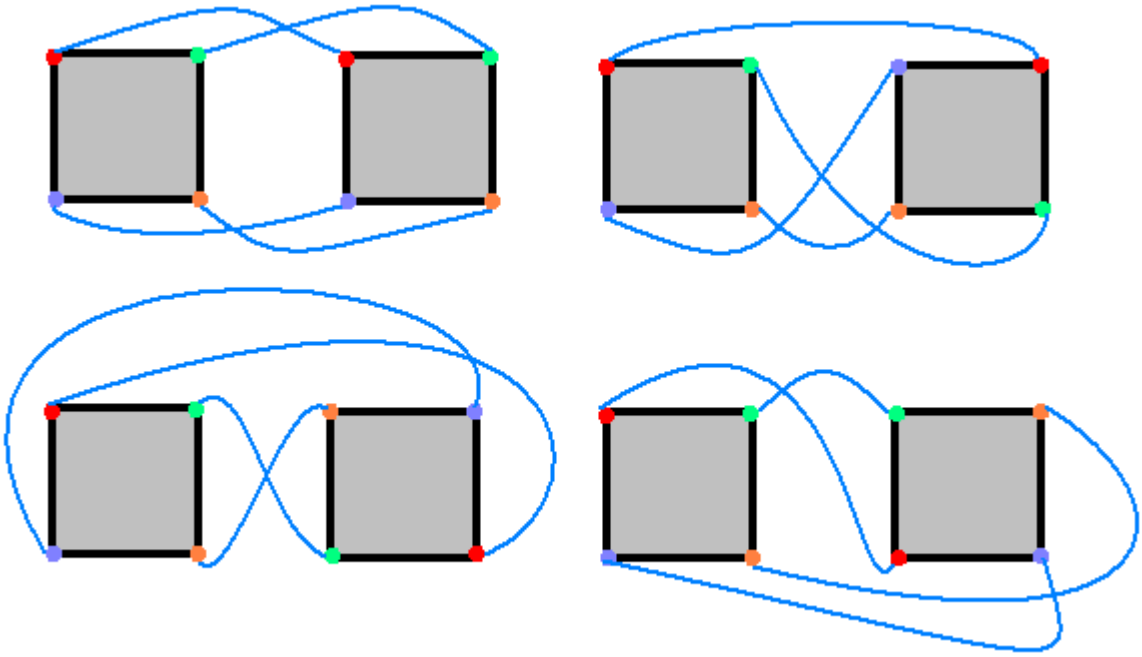
If the heat exchanger is of uniform section, the section of the middle portion can be considered identical, and neglect the effect of the end elements. On the input and output sections of each section of the variables (except pressure and temperature) are distributed equally. Pressure and temperature differ by a constant amount.

Snap points of periodic surfaces

Periodic surface can bind to both single and multiple ways (see illustrations below).



Example of a pair of periodic surfaces being connected only way



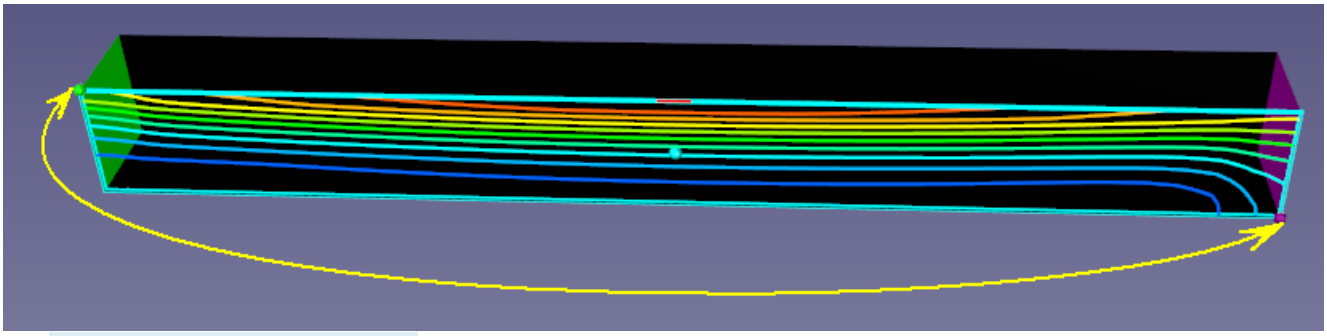
Sometimes periodic surface can be linked in different ways (only one of which is correct)

The program was able to correctly associate the periodic surface, they are placed*as fixed* points, one snap point on each of the two binding periodic surfaces. Snap point defines the direction of the geometric center of gravity of the surface used in the construction of an orthonormal coordinate system used to map the surfaces. Typically, the snap points are set on the ground contours fracture surfaces periodic (but, if necessary, they can be set in arbitrary points, which may be necessary upon binding surfaces divided into unequal ways polyhedra).

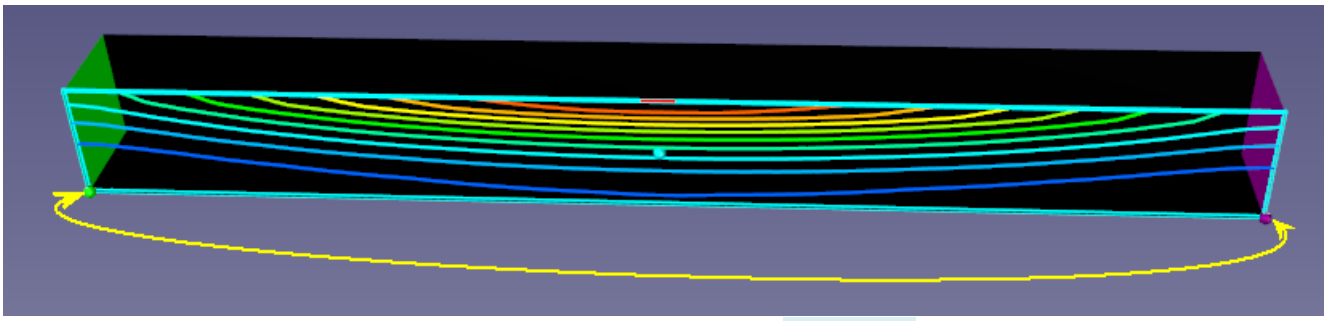
The program requires the use of snap points even when the only options for comparing periodic surfaces, as This requires matching algorithm.

At the wrong setting of snap points following situations may occur:

- binding periodic surfaces would be impossible in general
- *or*periodic surface will be connected incorrectly, leading to incorrect results of the analysis (see design)



Snap points are set incorrectly. This has led to improper binding of periodic surfaces and wrong calculation results.

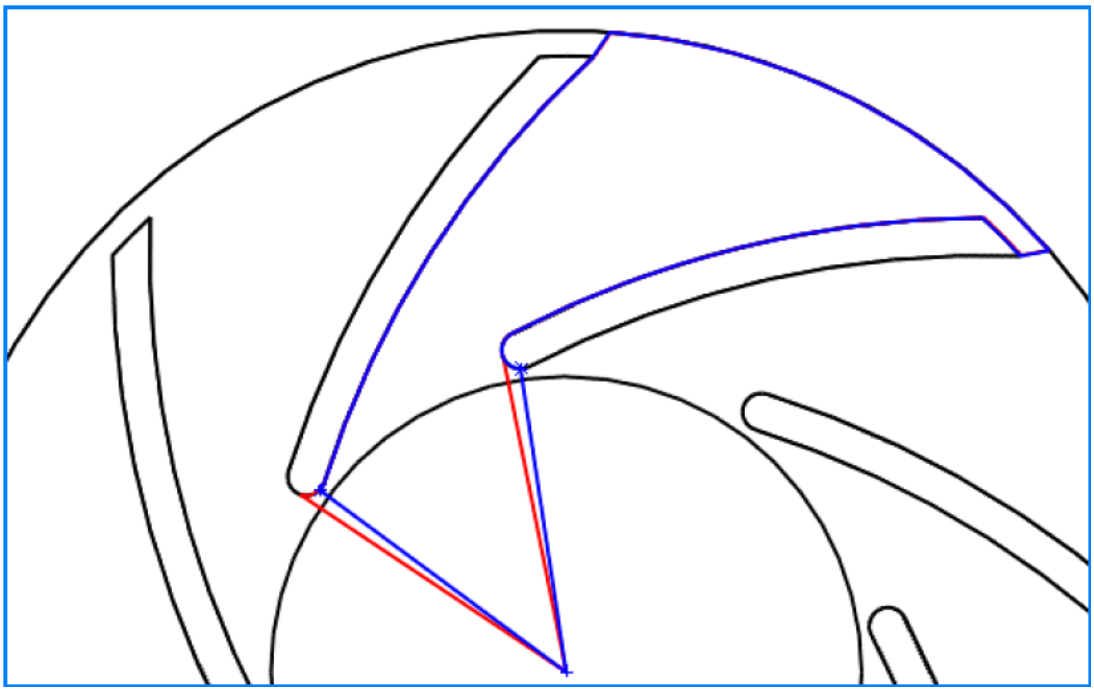


An example of correct setting of snap points

Recommendations on the selection of periodic fragments

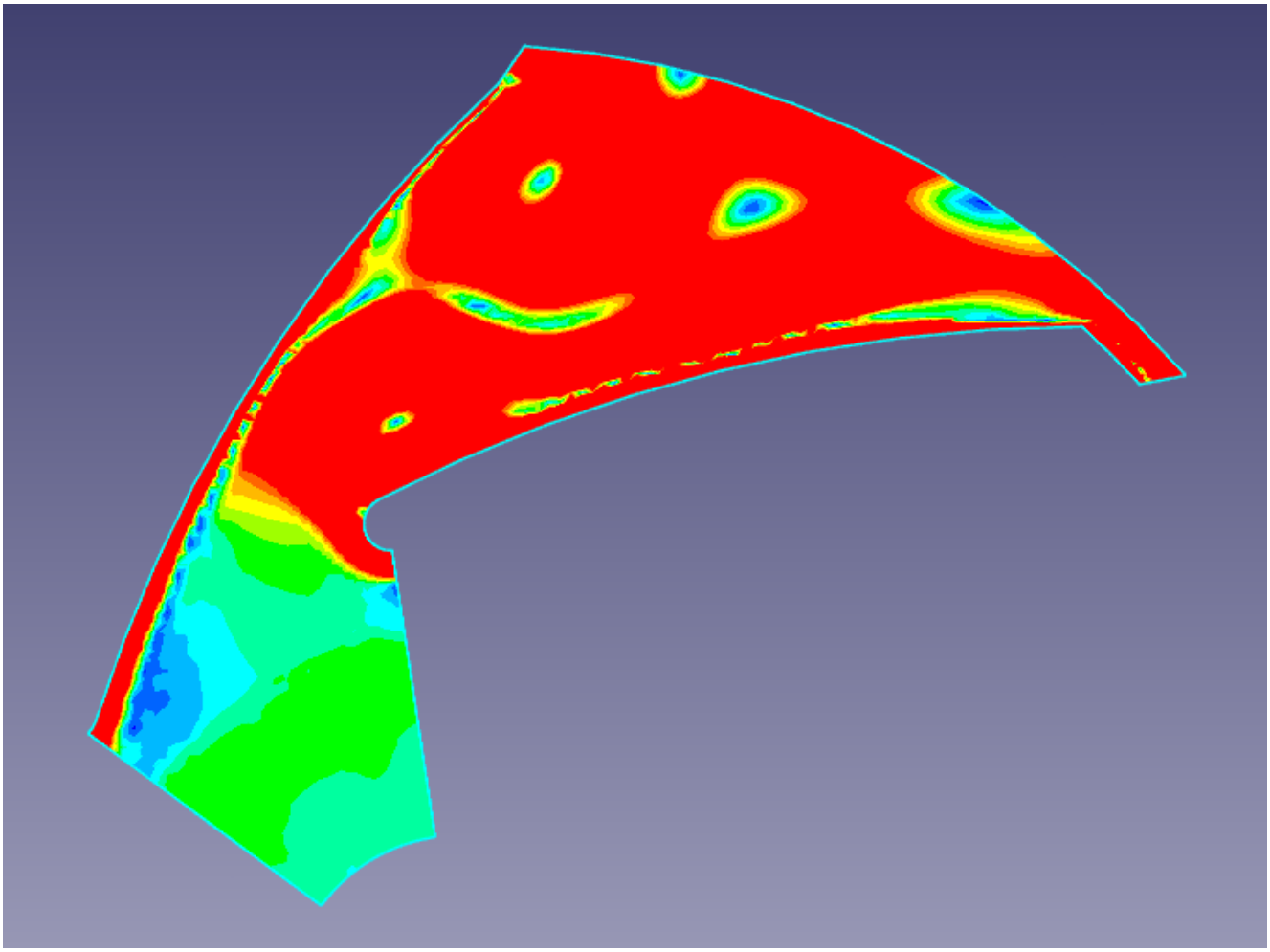
In preparing the geometry model in an external CAD-system, when the selection is made periodic fragment, you need to make sure that on a periodic surface did not appear "poor little cell." The appearance of such cells may lead to the fact that the flow through the periodic surface will be properly recognized. To avoid this situation, try to avoid sharp corners when creating periodic surfaces. In the example shown in the figure below:

- blue lines - the correct version of the periodic allocation of fragment
- red lines - the wrong option (as produced narrow protruding parts)



Follow these recommendations or give guidance to people preparing their geometry model in an external CAD-system.

Illustration



Example of calculation with periodic surfaces

Special aspects of the periodic surface's use

Prohibition of use a periodic surface for problems with a free surface

The boundary conditions of the **Periodic surface** type cannot be used in problems with a free surface.

Prohibition of crossing a periodic surface by a Moving Body

You cannot apply crossing a periodic surface by a **Moving body**.

Slight difference between areas of Periodic surfaces, which were calculated by different methods

Areas of corresponding **Periodic surfaces**, which were calculated by different methods (based on their source geometries or by **Characteristics**, specified on them) might differ by several percents (because in **Characteristics** the program calculates areas of the surfaces, which are built as projections of one **Periodic surface** on another).

Step by step instructions for defining a periodic surfaces

See subsection *"Specifying and using periodic surfaces"* in section Operations with boundary conditions.

6.11.8.4 Sliding surface

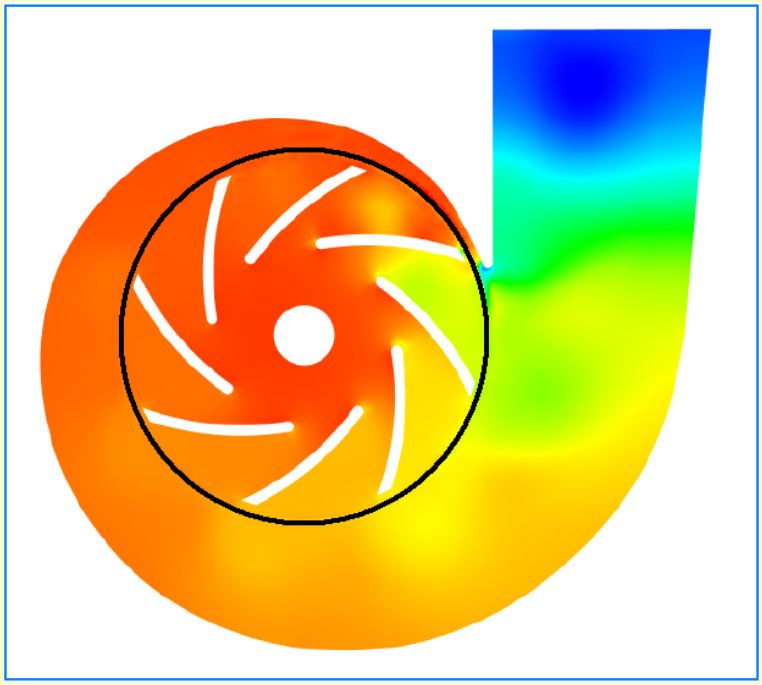
Sliding surface is a kind of **connected boundary conditions** in which the connection is done *over all variables* taking into account *rotation* of one **Subregion** relatively to another one.

Sliding surface separates two subregions that can rotate one relatively the other.

Combination of sliding and periodic surfaces can be used to simulate problems in the sector-sliding setting.

Example:

Simulation of the motion of air in the centrifugal pump, between the rotating rotor blades and the stator channel. Subregion near the rotor - rotating, it is within the fixed stator subregion. **Sliding surface** in this example is a wide lateral surface of the cylinder encompassing the rotor blades.



Sliding surface (black circle) separates the subregion of rotating rotor the fixed subregion of stator

Sliding surfaces are created:

- *either* in the CAD system and it is saved as separate geometric files
- *or* in **Preprocessor** from the surfaces of an axisymmetric standard geometric **Object** (**Cone/cylinder** or **Ellipsoid/sphere**)

When forming a geometry model of the project, initially a a single **Subregion** is loaded, forming the **Region** (without division into mobile and fixed parts), and then **Sliding surfaces** are inserted into the main geometry. As a result of each of these inserts some **Subregion** is split into two new **Subregions**.

For those surfaces that are to be connected, the user specifies the type of the boundary condition as **Connected**, and later connects pairs of such surfaces by a **Sliding surface** binder condition. After this, the user must set a rotation for those **Subregion**, which is considered as rotating (the rotor).

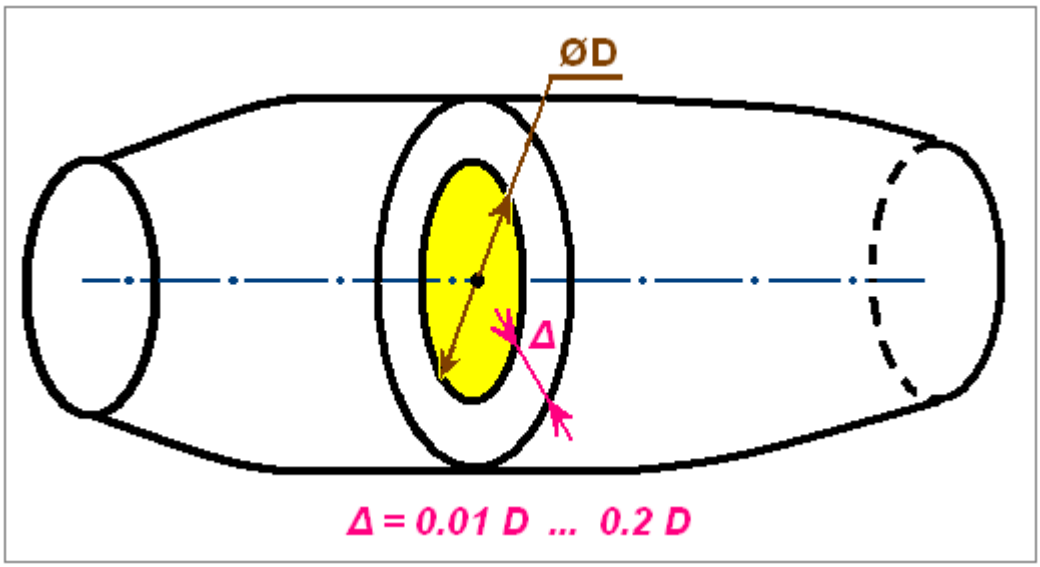
Note: until a sliding connected boundary condition is specified, a rotation cannot be set in a **Subregion**.

Requirements to use of sliding surfaces

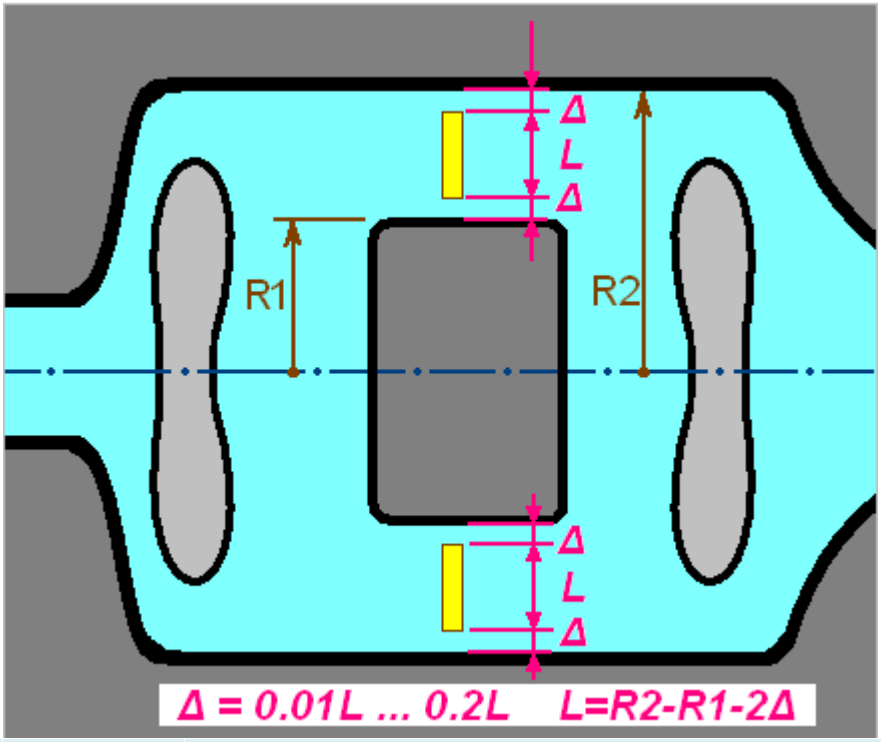
Main rules:

1. Boundary conditions of the type **Sliding surface** can not be used in problems with a free surface.
2. Crossing a sliding surface by a **Moving body** is prohibited, you cannot simulate this.
3. A **Sliding surface** can only be built using an **Imported object**.
4. The **Imported object** must be an axially symmetric body (i.e., must be a surface of rotation).
5. The **Imported object** must lie within the **Subregion**, into which you plan to insert the sliding surface. Intersection of this **Imported object** and the **Subregion**'s boundaries (**Boundary conditions**) is not allowed.

6. The sliding surface should be inserted in a place where surface of the main geometry is smooth enough (has no angles, roundings, and connections of shapes).
7. We recommend to specify the **Imported object** in such a way, when the gap(s) between its edge(s) and the main geometry is/are in a range from 1 to 20 percent of the total width of the section of **Imported object** by a plane passing through the axis of rotation (see illustrations below).
8. A **Sliding surface** can be only inserted into the project's geometry, if in its properties a **Local coordinate system** and **Rotation** are specified. Otherwise, the appropriate command from the context menu is inactive.
9. A **Sliding surface** can be only inserted into the project's geometry, if no **Model** is specified in any of the **Subregions** of the project.



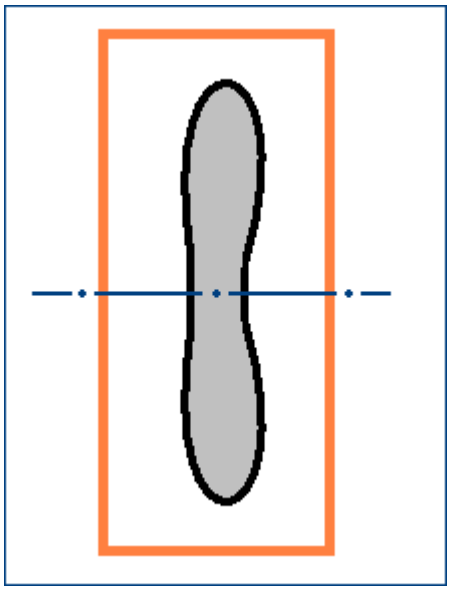
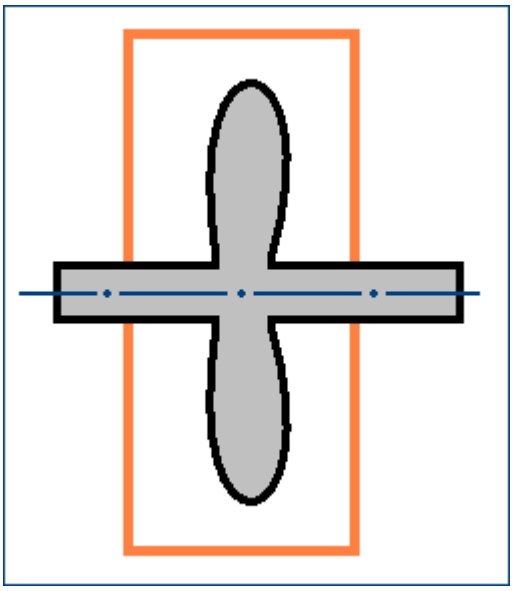
Recommendation for specifying the distance between an edge of an *Imported object* and the surface of the main geometry when the *Imported object* is a disk.



When inserting an *Imported object* into a circular channel, the distances between its edges and the surface of the main geometry are recommended to specify in a range from 1 to 20 percent of the thickness of the *Imported object* (which is a ring).

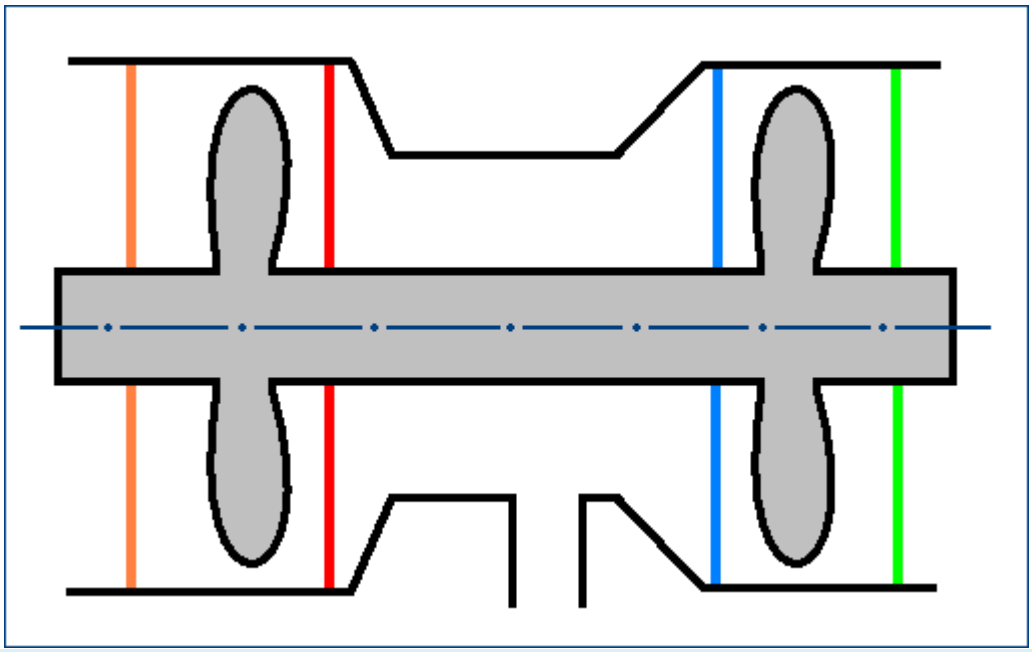
Other specifics:

1. Sliding surface can be adjacent to another BC if the other BC is also an axially symmetric. This is not to avoid crossing **Imported object**, on which will be built the sliding surface, and the other BC. An edge of the **Imported object** should come close to another BC (but within the accuracy of the import geometry).
2. According to the Gauss' formula (the divergence theorem, Gauss-Ostrogradsky theorem), for a closed polyhedron, the sum of products of areas of its faces multiplied by normal vectors to these faces must be zero ($\sum s_i \boldsymbol{n}_i = 0$ where s_i are areas of the faces and $\boldsymbol{n}_i$ are the normal vectors to the faces). When this parameter, normalized by dividing by the surface area of the cell, is greater then 10^{-7} , this usually indicates an error in building of cells but, when **Sliding surfaces** are used, this parameter can increase up to no more then 0.01, which is not an error but the effect of not precise mapping of facets of the unturned and turned positions of the **Sliding surface**. You can check the correctness of the generated grid by this parameter calculated before inserting the **Sliding surfaces**.
3. Particular cases of the construction of the sliding surface.
   a. If **Imported object**, which is constructed on the basis of the sliding surface is a closed surface of revolution, it will create a new **sub-region** is fully proportional to the volume of **Imported object** and all surface of the **Imported object** will be a sliding surface.
   b. If **Imported object** to an unclosed surface, there must be other **Boundary conditions** (presented necessarily bodies of revolution), which, joined with the surface of **Imported object**, create a closed surface of revolution. An exception may be the rotor, which is not required to be a body of revolution (as it rotates itself). But in any case the place where the closing is to occur and the sliding surface of the existing boundary conditions must be a line rotation. Examples of such a statement given in the figures below.

a)  b) 

Examples of specifying a **Sliding surface** (indicated by orange contours)
a) an initially closed sliding surface;
b) an unclosed surface should closely adjacent to the shaft that the combination therewith form a closed surface

A more complex example:



A more complex example of use of sliding surfaces (they are marked with orange, red, blue and green)

A feature of this statement is that in the middle of the volume of the computational domain is not a body of revolution. Accordingly, the average should be separated from the rest of the computational domain.

This can be achieved by inserting a non-closed four (or even flat) surface that together with the walls of the stator and rotor form a new 5 volumes. Two of these volumes will contain the rotor blades and non-rotating surface in the middle and two outer subregions adjacent to the rotating must be bodies of rotation.

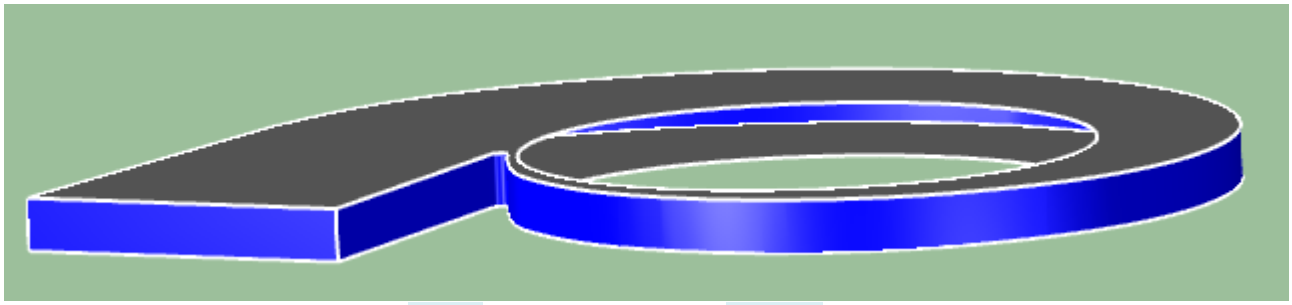
An algorithm for constructing the sliding surface

Sliding surface for the construction of the program uses the following algorithm:

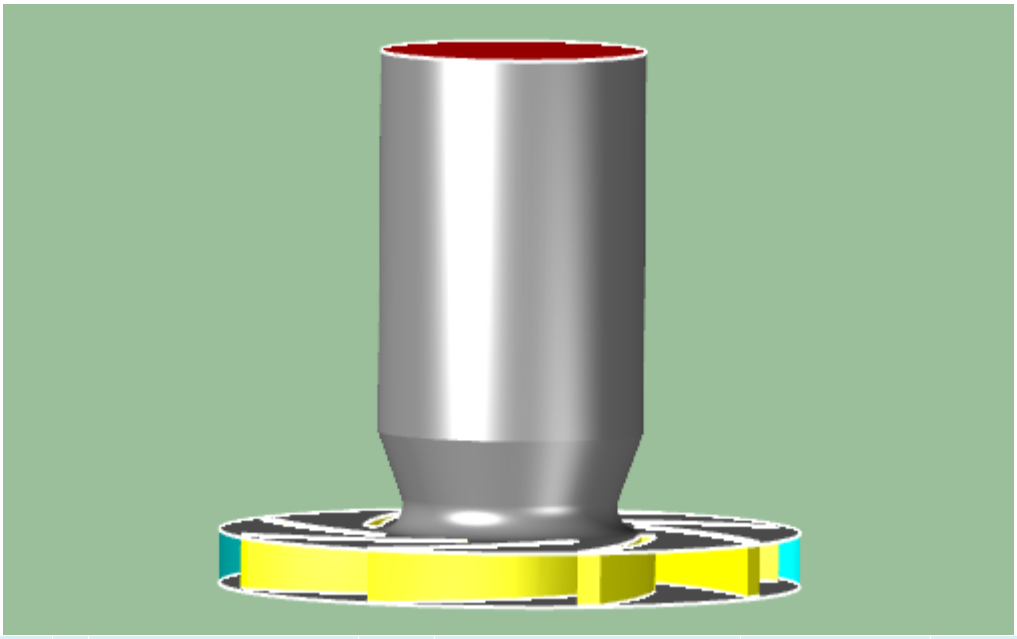
1. **Imported object** intersected by the plane passing through the axis of rotation and one of the points on its surface.
2. If the resulting cross-section is not closed, the lines continue to the intersection with the nearest **Boundary condition**.
3. The resulting cross section is rotated about an axis of rotation, thereby forming a surface which forms a new **Sub-region**. Thus, a new sub-region limits not an **Imported object** but an object that is constructed using:
   - axis of rotation
   - sectional **Imported objects**

Specifics of visualization of the sliding surface (its transparency from the observer when one of the subregions is hidden)

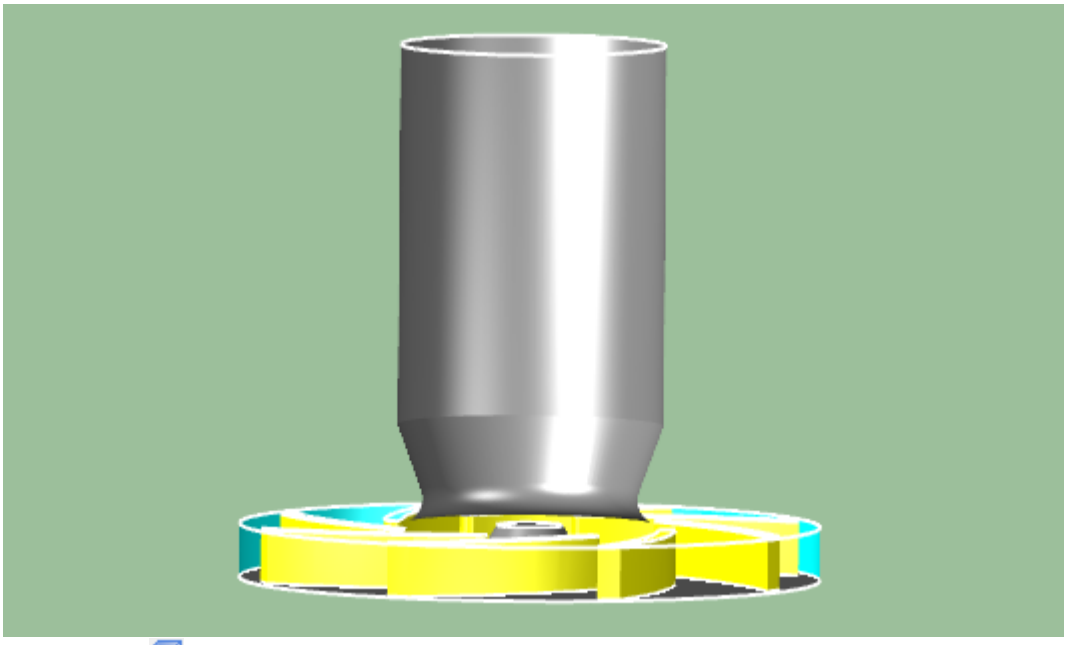
If you set hiding of a subregion, which has a sliding surface, then, at visualization of the computational domain, this sliding surface in the image of the other subregion becomes transparent in the direction from the observer, as if the button  (**Show/hide surface facing viewer** in the toolbar **Solids**) be pressed and acting on it, even when this button has not been pressed. See examples on the illustrations below.



Subregion **Rotor** hidden subregion **Stator** is not hidden. Parts **Sliding** surface that are proximal to the observer and not obstructed by other surfaces of the **Stator** are transparent, even when the button is not pressed.



Subregion **Stator** is hidden while subregion **Rotor** is not hidden. Parts of the **Sliding surface**, which are proximal to the observer, are transparent, even when the button is not pressed. Distal (from the observer) parts of the **Sliding surface** are opaque (they are colored blue).



When the button is pressed, all surfaces, which are proximal to the observer, become transparent

Step by step instructions for setting sliding surfaces

See the subsection *"Setting and use sliding surfaces"* in the section Operations with boundary conditions.

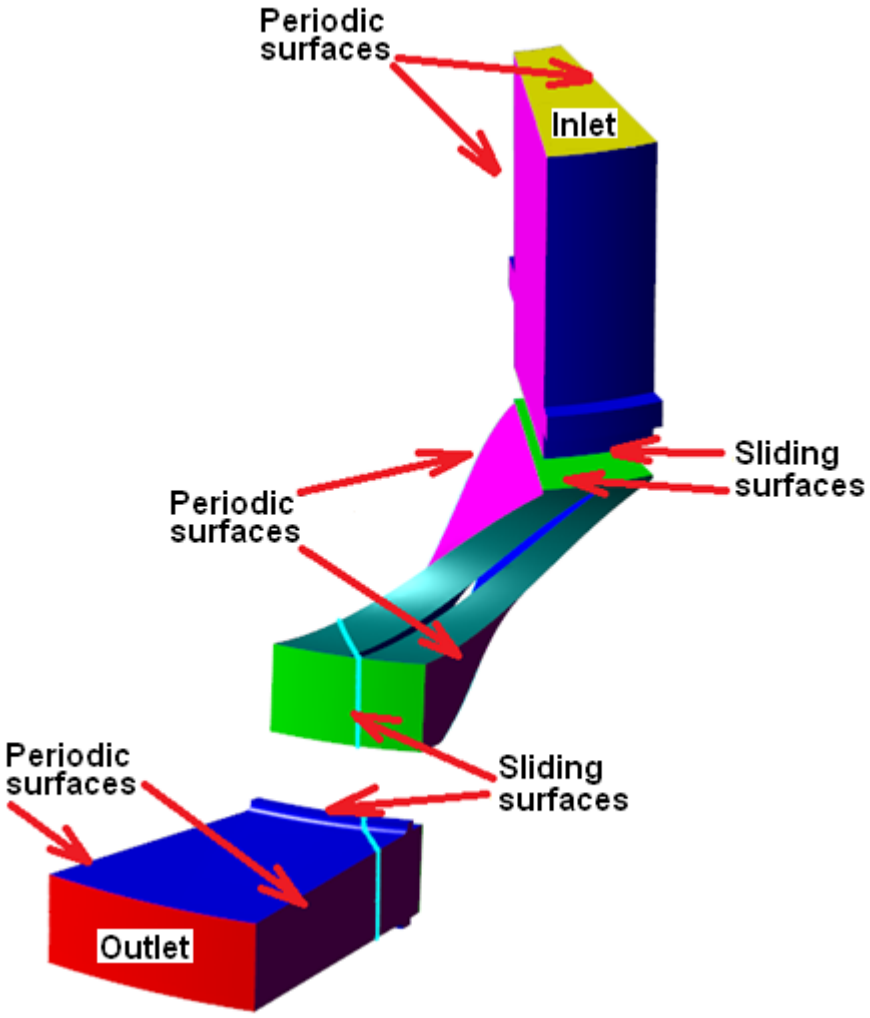
6.11.8.5 Sector-sliding setting

The *sector-sliding problem setting* combines periodic and sliding connected boundary conditions.

Preparing a project in a sector-sliding problem setting

When you need to simulate a problem in a sector-sliding problem setting, you have to prepare the geometry is such a way, which provides that computational domain would contain both periodic and sliding surfaces.

In the sector-sliding problem setting, you *cannot* insert a sliding surface into the source geometry.



Example of computational domain (a sector of an axial compressor)

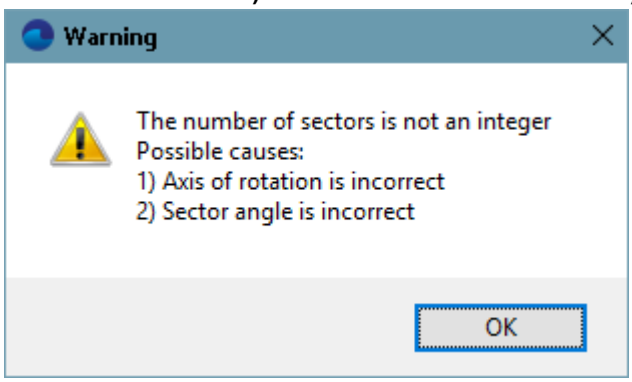
When working with the geometry in a CAD software system, to prepare the geometry do the following steps:

Step	Actions
1	Make volumes (as on the illustration above) and, not displacing them, save each of them in a separate surface mesh file (format `stl` or other format).
2	Load any of these volumes into an empty project.
3	Other volumes load as **Imported objects**.
4	Move each of these **Imported objects** in such a way that they don't intersect each other and the source volume (**Region**).
5	Embed the **Imported objects** into the main geometry.

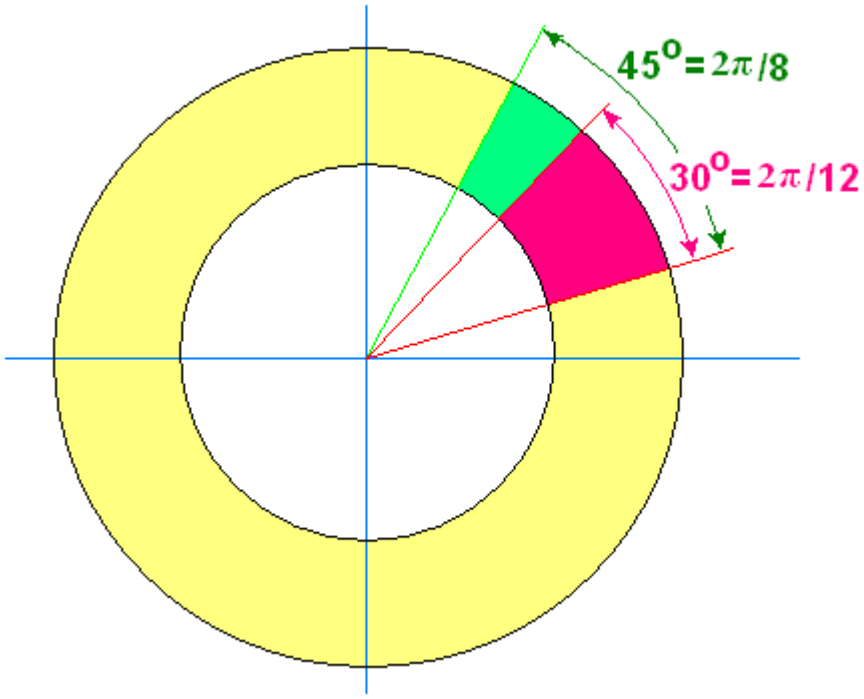
Requirements to the connected sliding surfaces of a project in a sector-sliding setting

- After the rotation axis is specified for a sector, the program checks whether the sector's geometry can fit into a circle around the specified axis, so that when the circle closes, the last sector accurately fits the first one. The necessary but not sufficient condition for the correct fit is the requirement that values of angles of the sectors must be 360 degrees (2π radian) divided by an integer number. But, even if values of the angles are acceptable, the fit will be impossible if position and/or direction of the rotation axis is specified incorrectly.

FlowVision evaluates these parameters and, if they are not acceptable, displays a message (**The number of sectors is not an integer Possible causes: 1) Axis of rotation is incorrect 2) Sector angle is incorrect**):

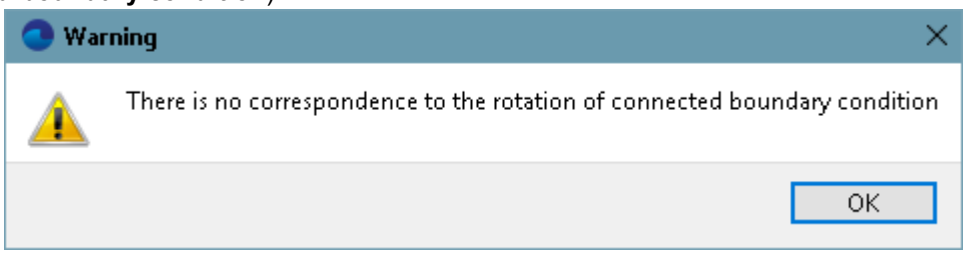


- Angles of sectors, corresponding to different surfaces of a pair of matching surfaces, do not have to be the same, for example, in simulation a rotor and a stator with different and aliquant numbers of blades. So areas of the matched connected boundary conditions also can differ, see the illustration below.



Example of a correct setting sectors of moving surfaces. The matching surfaces, when they are multiplies around a whole circle, form the same rings. Each sector is formed from the whole circle by splitting it by dividing by an integer number. The sectors are formed from the circle by splitting it into 8 and into 12 parts, the sector's angles are 45 and 30 degrees respectively. Areas of 45-degree sectors differ from the areas of 30-degree sectors.

- Axes of rotation of sliding surfaces and corresponding **Subregions** *must be the same*. If this requirement is not fulfilled, the program does not allow you to set a **Rotation**. If you try to set inappropriate **Rotation** for the corresponding **Subregion**, the program outputs a warning (**There is no correspondence to the rotation of connected boundary condition**):



This message means that the axis of rotation, which is selected on the **Subregion**, and the axis of rotation of the connected **Boundary condition** are not coincide (their vectors do not lie along the same right line).

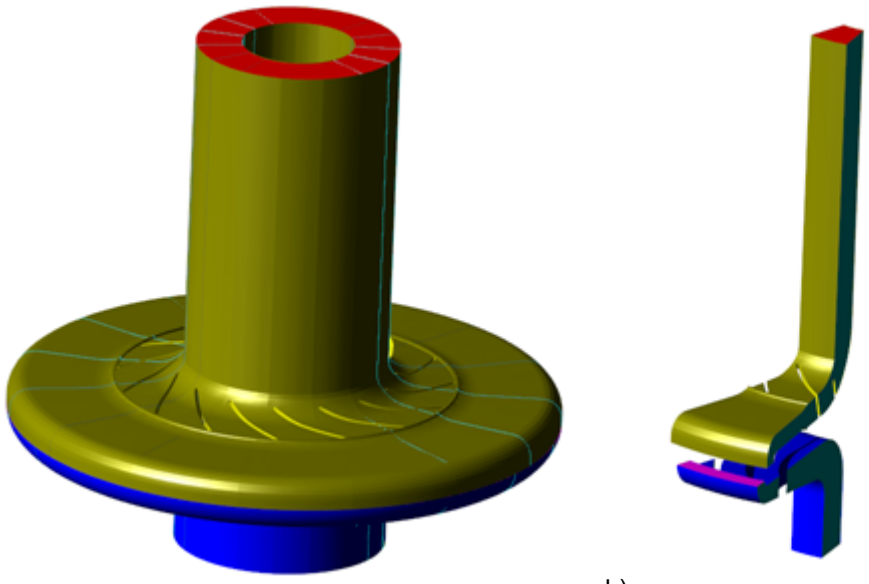
- The computational subregions must not intersect. If they intersect, you have to displace them, not changing their axes of rotation, using the following methods:

- move a **Subregion** along its axis of rotation
- *and/or* rotate a **Subregion** around its axis of rotation

- For correct matching of surfaces in a pair of sliding surfaces, each of the surfaces, after it is multiplied around a whole circle, must form the *same* axially symmetric surface (ring, disk, cone, side surface of a cylinder), as the other sliding surface. **Pre-Postprocessor** checks that areas of these axially symmetric surfaces be the same (this is an necessary but not a sufficient condition that the sliding surfaces are set correctly).

Step-by-step procedure of preparing a project in a sector-sliding setting

As an example, lets consider an axial compressor.



a) b)

An axial compressor: a) in a full setting; b) in a sector-sliding setting

Step	Actions
1	On sliding and periodic surfaces set the **Connected** type of their boundary conditions. As a result, all these boundary conditions will be placed into the folder **Boundary links > Free BCs**.
2	On each of the periodic surfaces, specify a **Snap point** in such a way that for each pair of matching surfaces, when they matches, the **Snap points** correctly correspond to each other (with no twisting). See details in the section Periodic surface.
3	On sliding surfaces, specify their axes of rotation (by specifying a **Local CS #N** and a **Rotation #N**). When you specify an axis of rotation, you inform the program that this surface is a sliding one.
4	Form the **Binders**. Incorrect specification of the bound surfaces prevents forming the **Binders**. If the **Binders** would not be formed, check: • for periodic surfaces: correctness of setting the **Snap points** and areas of the bound surfaces (the areas must be equal) • for sliding surfaces: correctness of specifying their axes of rotation ⚠ If creation a **Binder** for some sliding surfaces failed, check representation of their geometry in the CAD system. The CAD system multiplies (copies several times) the volumes, which are not able to be bound, and checks them for intersections. If the multiplied volumes do not intersect, contact the technical support service and attach the illustration of the multiplied subregion.
5	Create a periodic and a sliding **Binder conditions**.
6	Add **Binders** into appropriate **Binder conditions**.

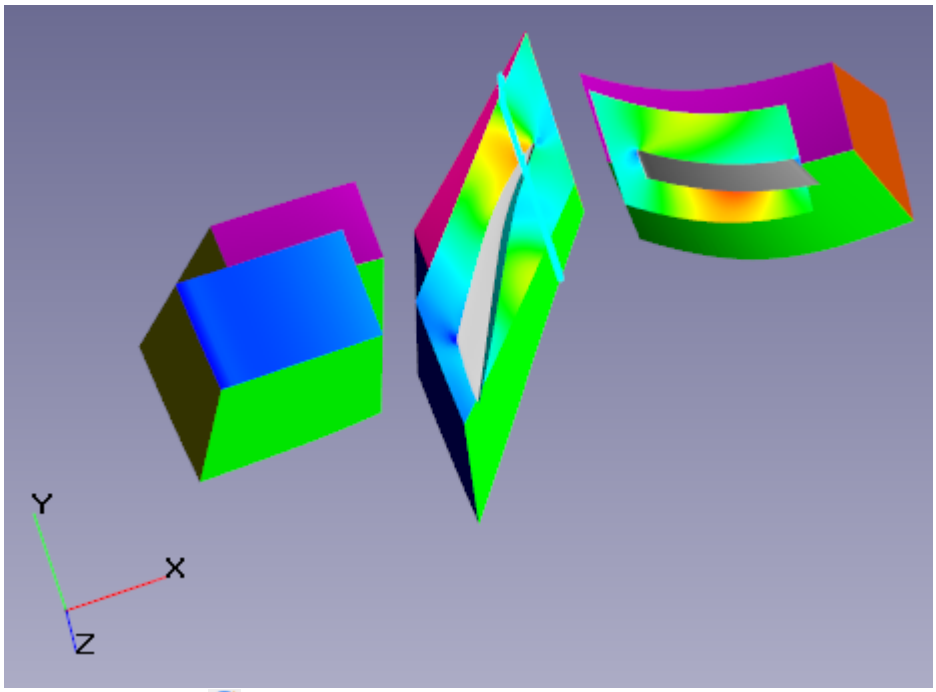
7	In the advanced settings of Solver specify the parameter **Sliding surfaces > Method**. Possible options are: • **Frozen Rotor**: this value is recommended for most of the cases. Use this setting, when sectors of different **Subregions** are not are not divisible by (this is so in most of the cases). Other advantages of the **Frozen Rotor** method are good computational speed and no limiting the time step by the value **Slide CFL = 1**. • **Sliding**: this value can only be used when sectors of **Subregions**, matched by the **Sliding surfaces**, are divisible by. This selection can provide better accuracy of the computation, but it has no advantages of the **Frozen Rotor** method. Also, when **Method=Sliding** is selected, the time step is limited by the value **Slide CFL = 1**.
8	Tune the visualization (see section *"Tuning the visualization"* below).

See also a detailed step-by-step procedure in the *Tutorial: Examples of typical tasks* document, in the section *"Sector of axial compressor"*.

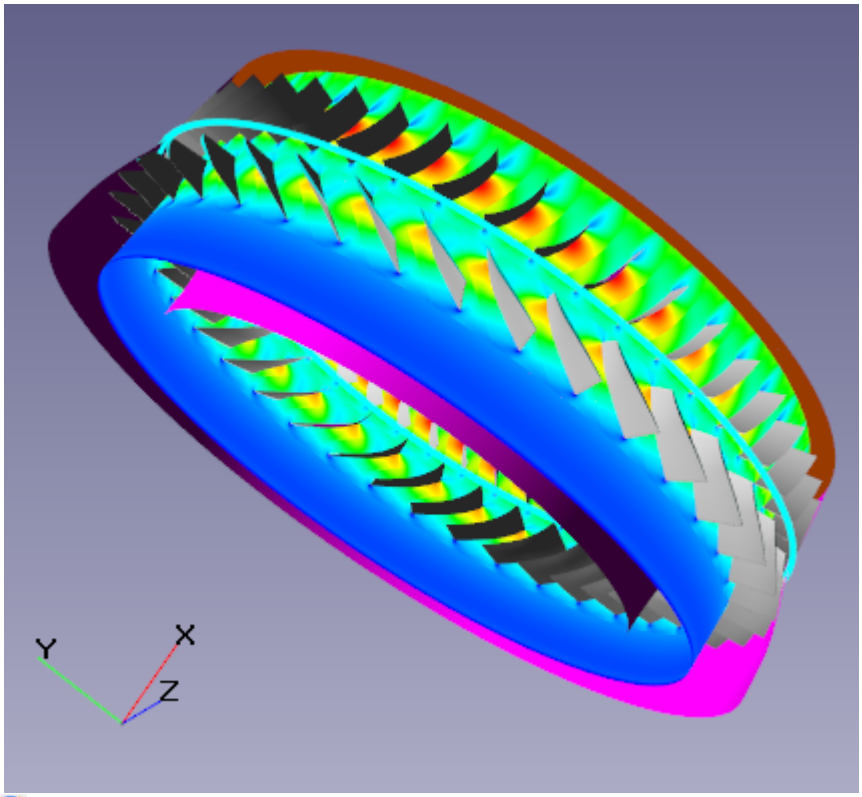
Tuning the visualization

If you wish, you can view the geometry and solution's visualization in **Postprocessor** in the full setting, for this the program multiplies (copies many times and places circularly) images of sectors. This visualization is toggled by the button (**Enable/disable duplication and overlapping of subdomains up to the complete model supplied with sector-sliding**) in the **Rendering** toolbar.

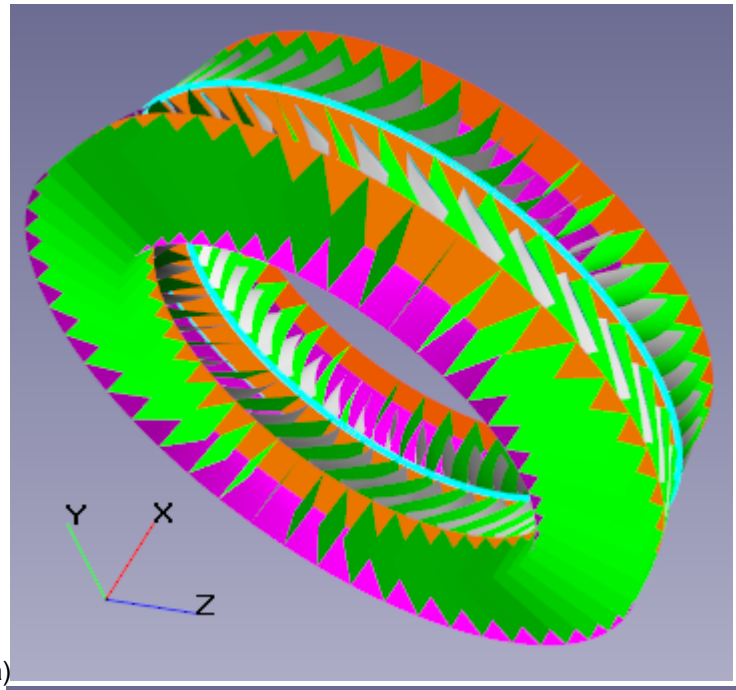
Displaying of periodic and sliding surfaces is enabled or disabled by parameters **Subregion positioning > Hide periodic** from basic settings of FlowVision (they are opened by the **File > Preferences** command from the main menu).

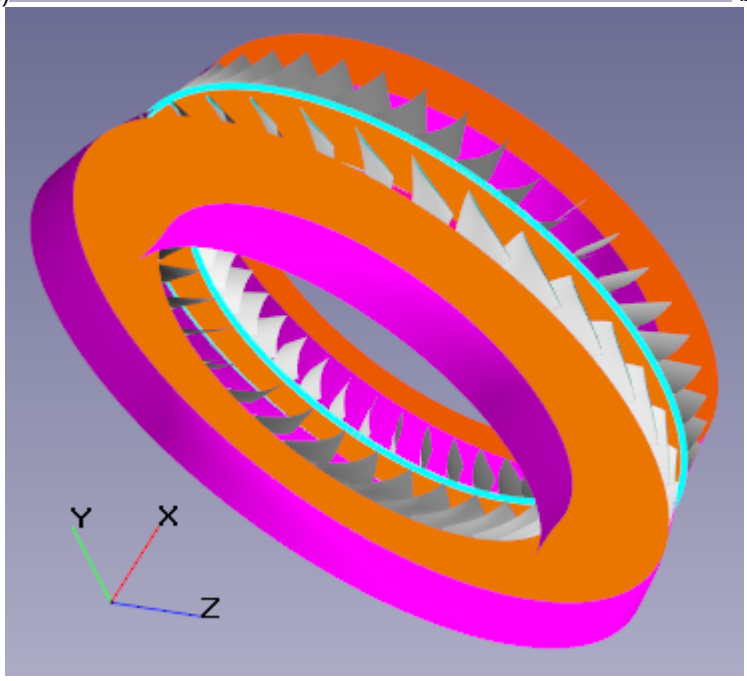


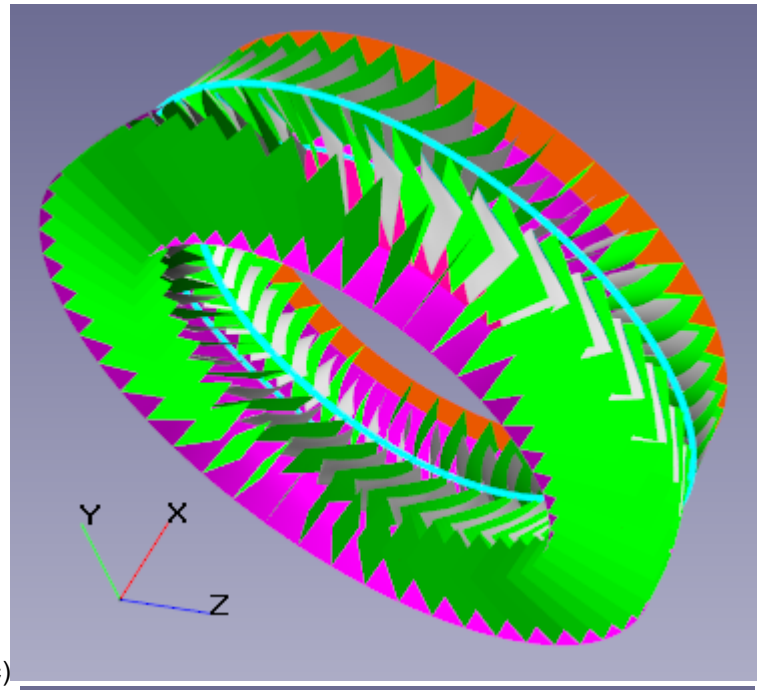
When the button is released, only one sector is displayed in **Postprocessor**



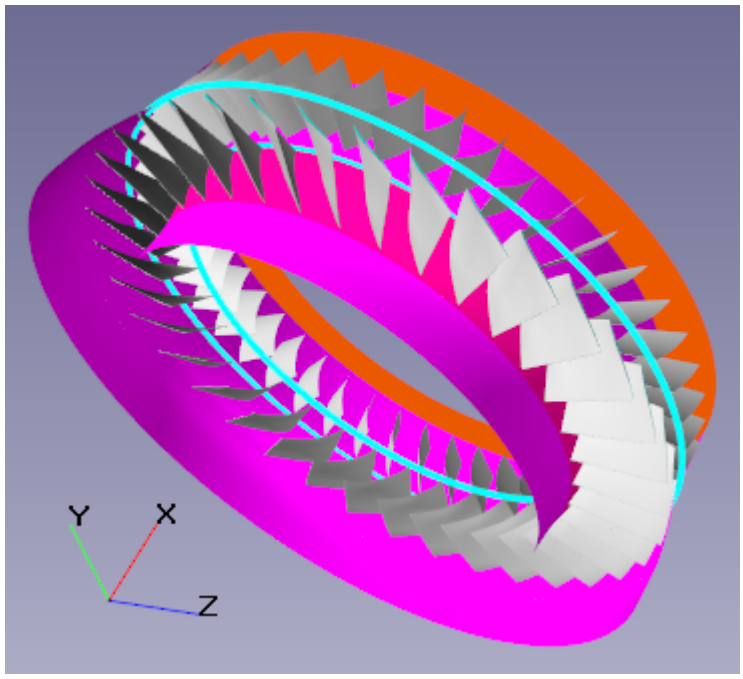
When the button is pressed, the program will build in Postprocessor images of all sectors around the wheel

a) 

b) 



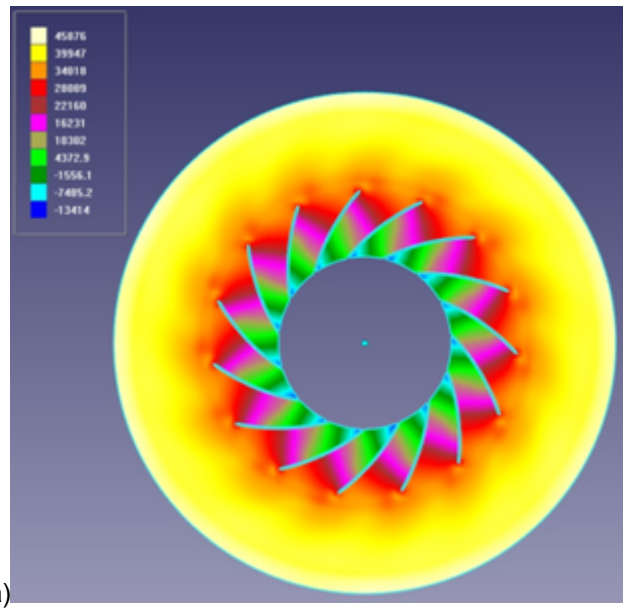
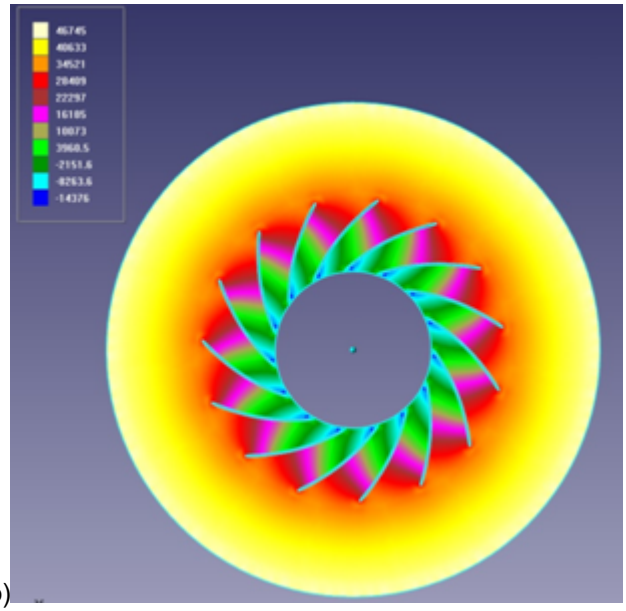
c) d)



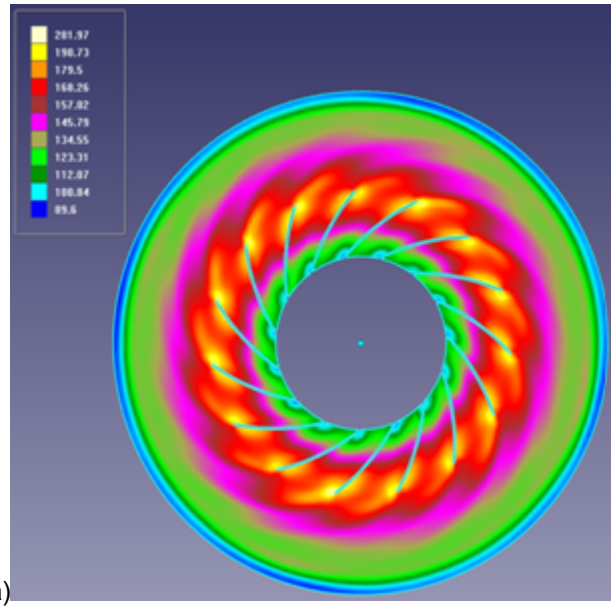
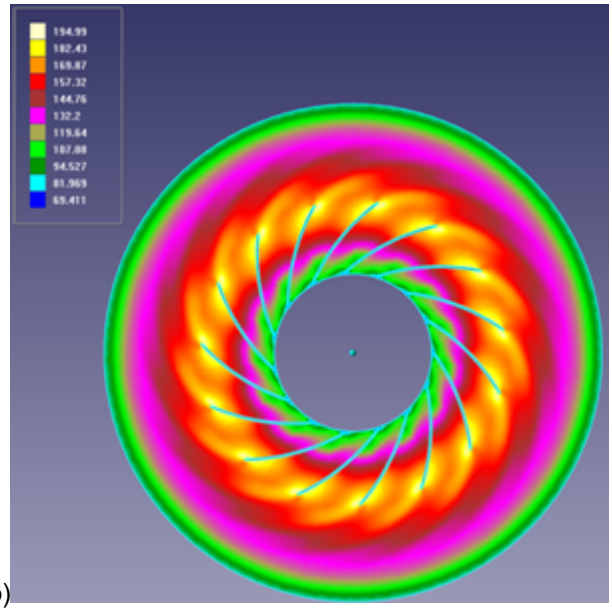
How the **Display > Subregion positioning** base settings act:
a: displaying of both sliding and periodic surfaces is enabled;
b: displaying of only sliding surfaces is enabled;
c: displaying of only periodic surfaces is enabled;
d: neither sliding no periodic surfaces are displayed

Examples of results of simulations in the sector-sliding setting

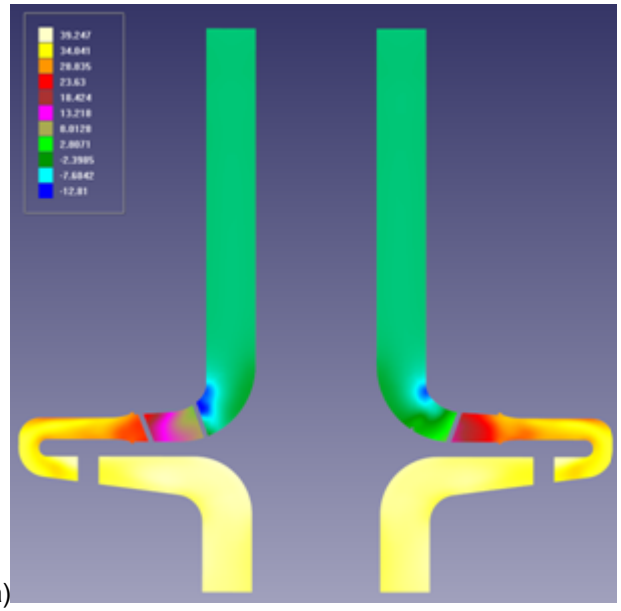
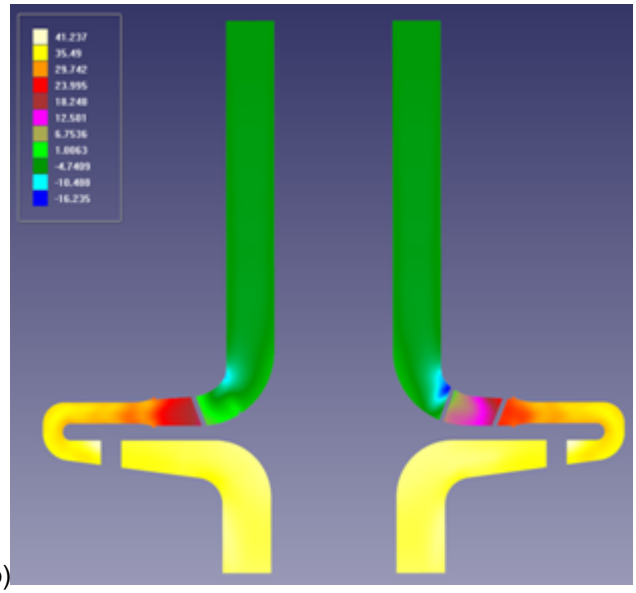
On the illustrations below you can see results of simulation an axial compressor in the sector-sliding setting (on the first illustrations you can compare the results with results of simulation in the full setting).

a)  b) 

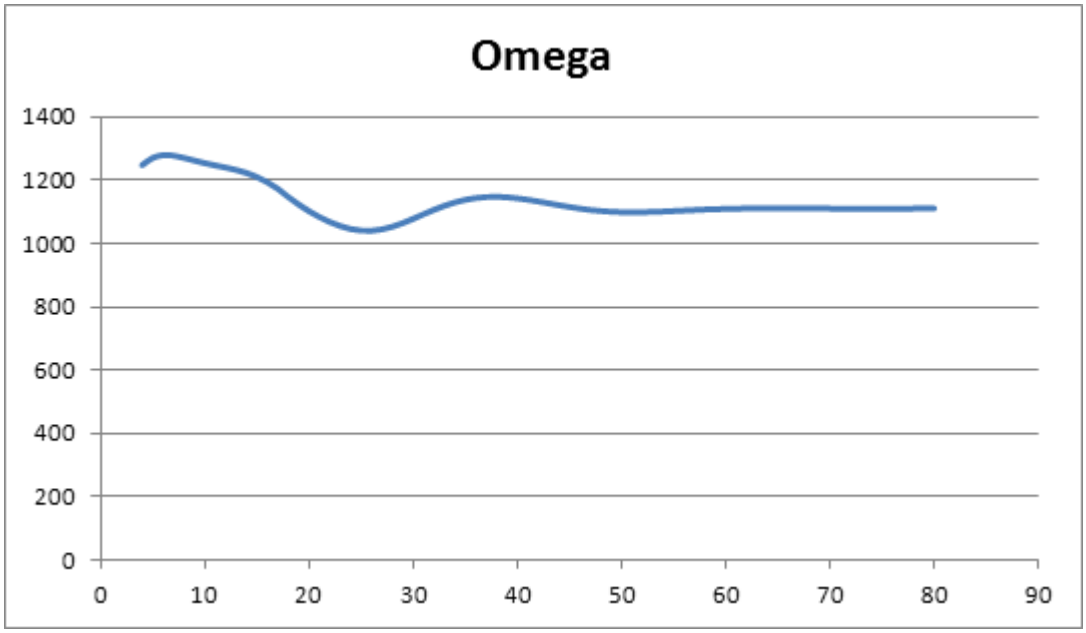
Pressure distribution: (a) in the full setting; (b) in the sector-sliding setting

a)  b) 

Distribution of the absolute value (module) of the velocity: (a) in the full setting; (b) in the sector-sliding setting

a)  b) 

Temperature distribution: (a) in the full setting; (b) in the sector-sliding setting



Plot of process stabilization of rotational speed when autorotation is simulated

6.11.8.6 Conjugated ablation

This connected boundary condition is used to simulate ablation (a thermochemical process of mass loss) from a surface, which separates **Subregions**.

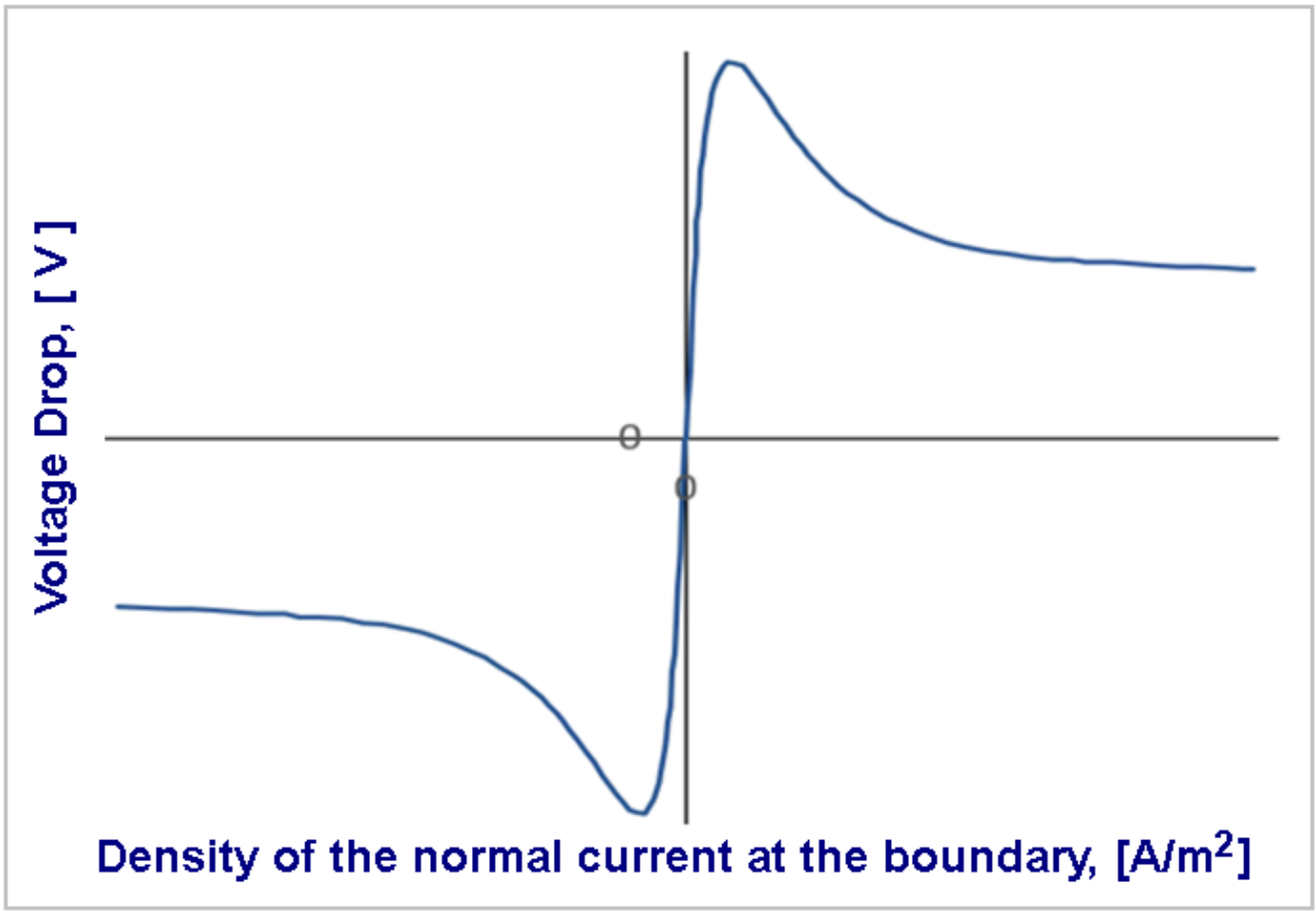
See also: Boundary conditions «Wall, ablation».

6.11.8.7 Conjugated electric potential

The **Conjugation for potential** connected boundary condition is used for values of the **Electrical potential** variable.

In properties of this boundary condition, you can specify voltage drop on the inter-phase boundary.

The **Voltage Drop** is specified in [V] as a constant or as a formula or a table value that depends on other value(s). Usually, dependency on density of the normal current on the boundary is specified:



See details in the section Theory > Physical processes > Electromagnetohydrodynamics > Boundary conditions >Template "Connected".

6.11.8.8 Conjugated Maxwell equations

This connected boundary condition is used for conjugation equations of electromagnetodynamics.

See details in the section Theory > Physical processes > Electromagnetohydrodynamics > Boundary conditions >Template "Connected".

6.12 Modifiers

A **Modifier** is a project element that allows:

- setting the values of computational variables during the solution process (or adding values of variables to existing values)
- setting regions where volume force, volume heat source, flow resistance force are acting
- adding (removing) elements setting additional physical processes (*ignition/extinction of combustion*)

A modifier is created as a combination of "**Object + Modifier type + Modifier properties**", and the modifier is acting in a certain computational subregion and is not acting outside the subregion's limits.

Several **Modifiers** may be linked to an **Object**, but only one of them can be a **Moving body**.

Types of modifiers

In *FlowVision* there are **Modifiers** of following types:

- Moving body
- Setting variable
- Volume force
- Volume heat source
- Ignition/extinction zone (available for creation only when **Phase #N > Physical processes > Mass transfer = Combustion**)
- Resistance
- Anisotropic resistance
- Anis. therm. conductivity
- Volume External charge
- External Current
- External Induction

Set of **Modifier**'s properties depends on chosen type of **Modifier**, but some parameters are applied for all types of **Modifiers**.

Modifier time of operation (activity)

The **Modifier**'s time of activity can be set using a group of parameters **Activation** in the **Properties** window:

Property in the Activation group	Description
Type	Defines how the **Modifier**'s activity time is set. The following options are available: • **Inactive** - the modifier is disabled and has no effect • **Only once by time** - the modifier is active from the moment of **Start in seconds** during the **Duration in seconds** period • **Only once by step** - the modifier is active from the moment of **Start in steps** during the **Duration in steps** period • **Permanent** - the modifier is constantly active • **Repetitive by time** - the modifier is activated periodically. The first activation of the modifier occurs at the moment of **Start in seconds**, the modifier remains active during the **Duration in seconds** period. The modifier activation procedure occurs with a periodicity defined by parameter **Period in seconds**, counted from the moment of **Start in seconds**. • **Repetitive by step** - the modifier is active from the moment of **Start in steps** during the period of **Duration in steps**. Then the activation/deactivation procedure is repeated after a **Period in steps** counted from the moment of **Start in steps**.
Start in seconds	Moment(s) of time when the **Modifier** starts (becomes active): • time of the first start (when **Type = Repetitive by time**) • time of the single start (when **Type** = **Only once by time**)
Duration in seconds	Duration of modifier activation for single or periodic start (**Type** = **Only once by time** or **Type = Repetitive by time**).
Period in seconds	Duration (in seconds) of the modifier activation/deactivation period. This setting is accessible when **Type = Repetitive by time**.

Property in the Activation group	Description
Start in steps	Step(s) at which the **Modifier** becomes active: • first time (when **Type = Repetitive by step**) • single time (when **Type = Only once by step**)
Duration in steps	Number of steps when the **Modifier** is active (if a single or periodic start is used). (**Type = Only once by step** or **Type = Repetitive by step**).
Period in steps	Period of modifier activation/deactivation, measured in steps. This setting is accessible when **Type = Repetitive by step**.

Scope of a Modifier

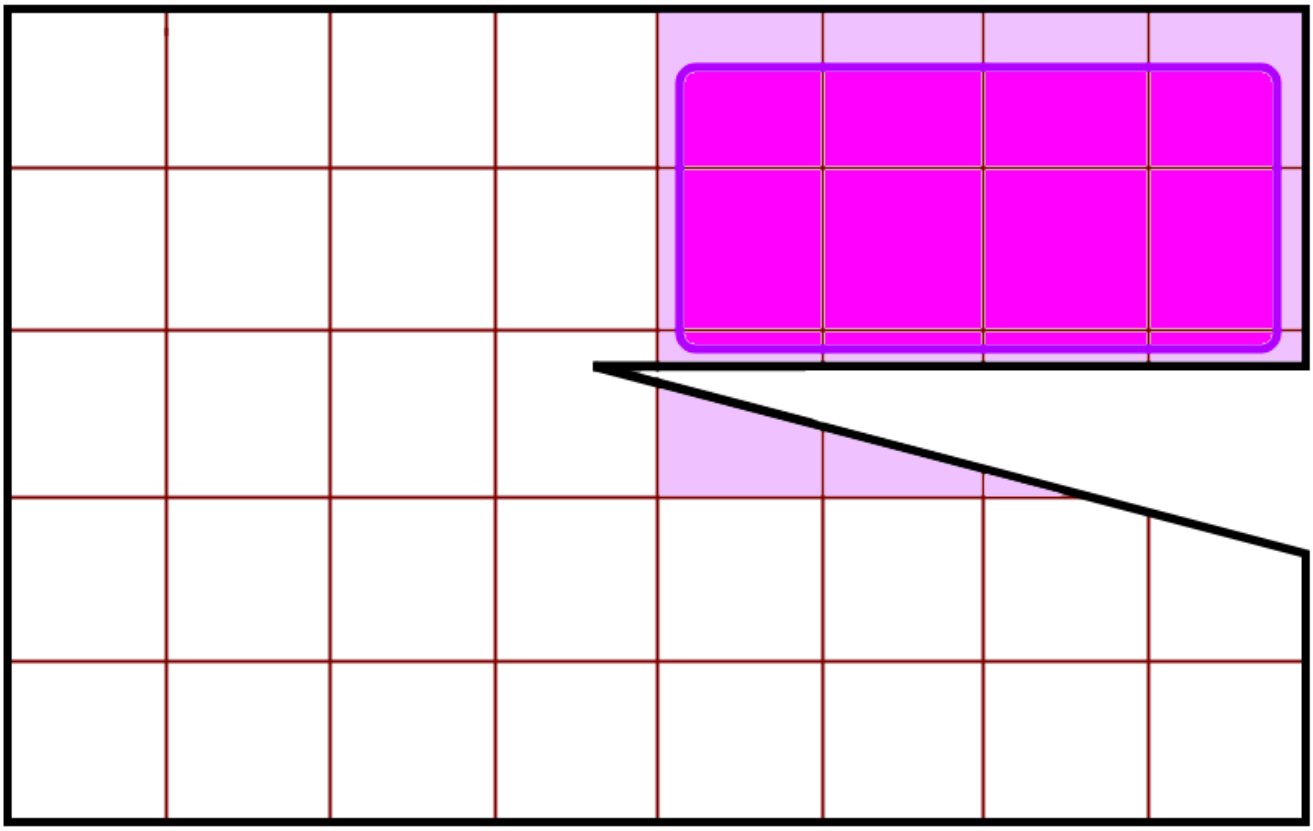
The scope of a **Modifier**, except **Moving body**, is the volume where the **Modifier** acts. The scope is formed specifically in cells near a finite-volume **Object**'s surface. These specifics are set by the **Scope** parameter in properties of the **Modifier**. You can select:

- **Scope = All cells** – in this case the **Modifier** acts in all cells that contain even a small fragment of the **Object**. The area, in which the **Modifier** acts, will be slightly extended (within the computational domain) around the **Modifier**'s **Object**, because the **Modifier** will act in *all cells* that are even partially occupied by the **Object**. This setting is used by default.
- **Scope = With centers inside** – in this case the **Modifier** acts in cells with centers that locate inside the **Object**.

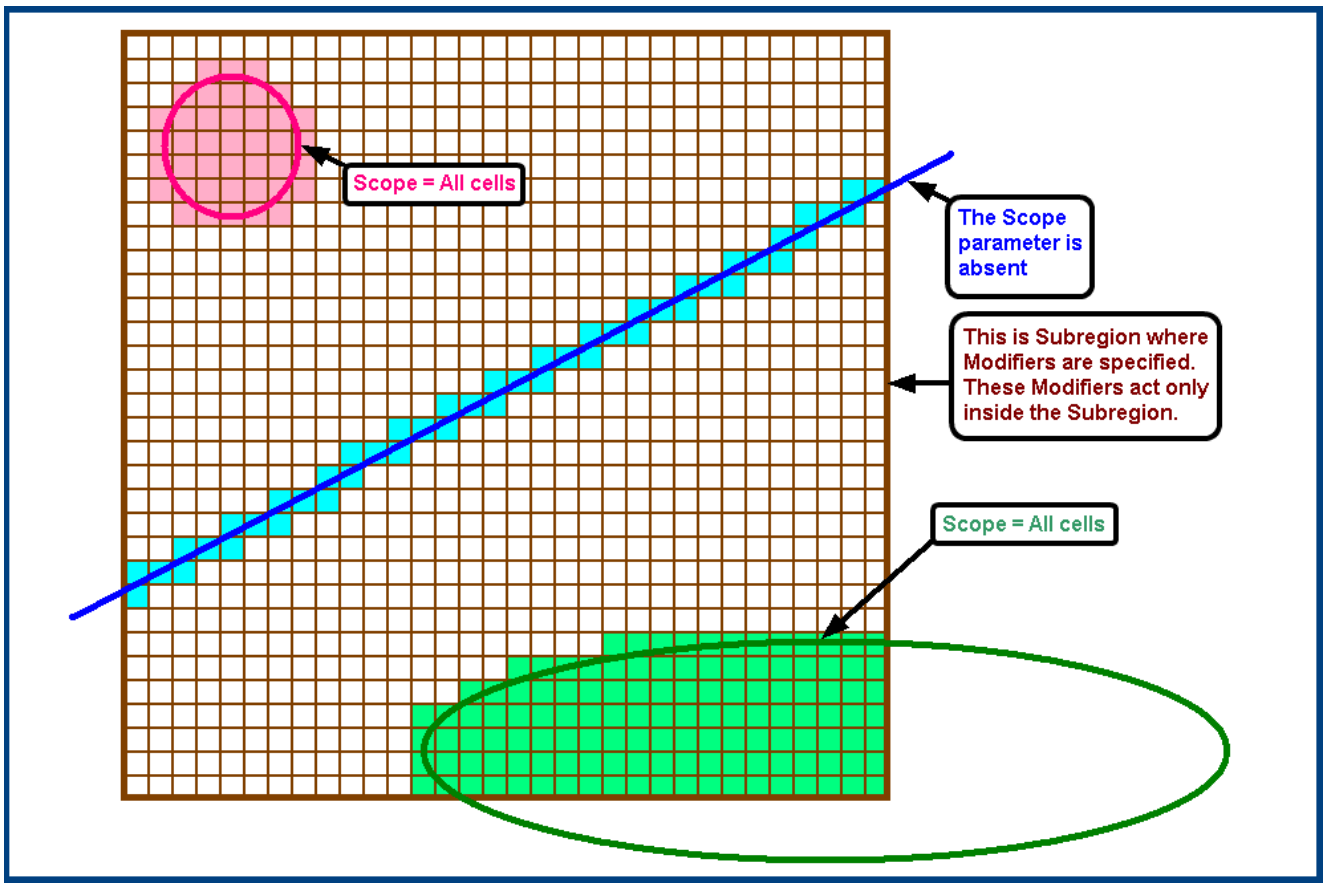
The scope of a **Modifier** is formed based on initial boxes of cells of the computational grid before the boxes might be split by surfaces of the computational domain and/or **Moving bodies**.

In some cases a cell might be split to new cells in such a way that some new cells would not contact the **Object** but nevertheless would be included in the scope of the **Modifier** (see the illustration below).

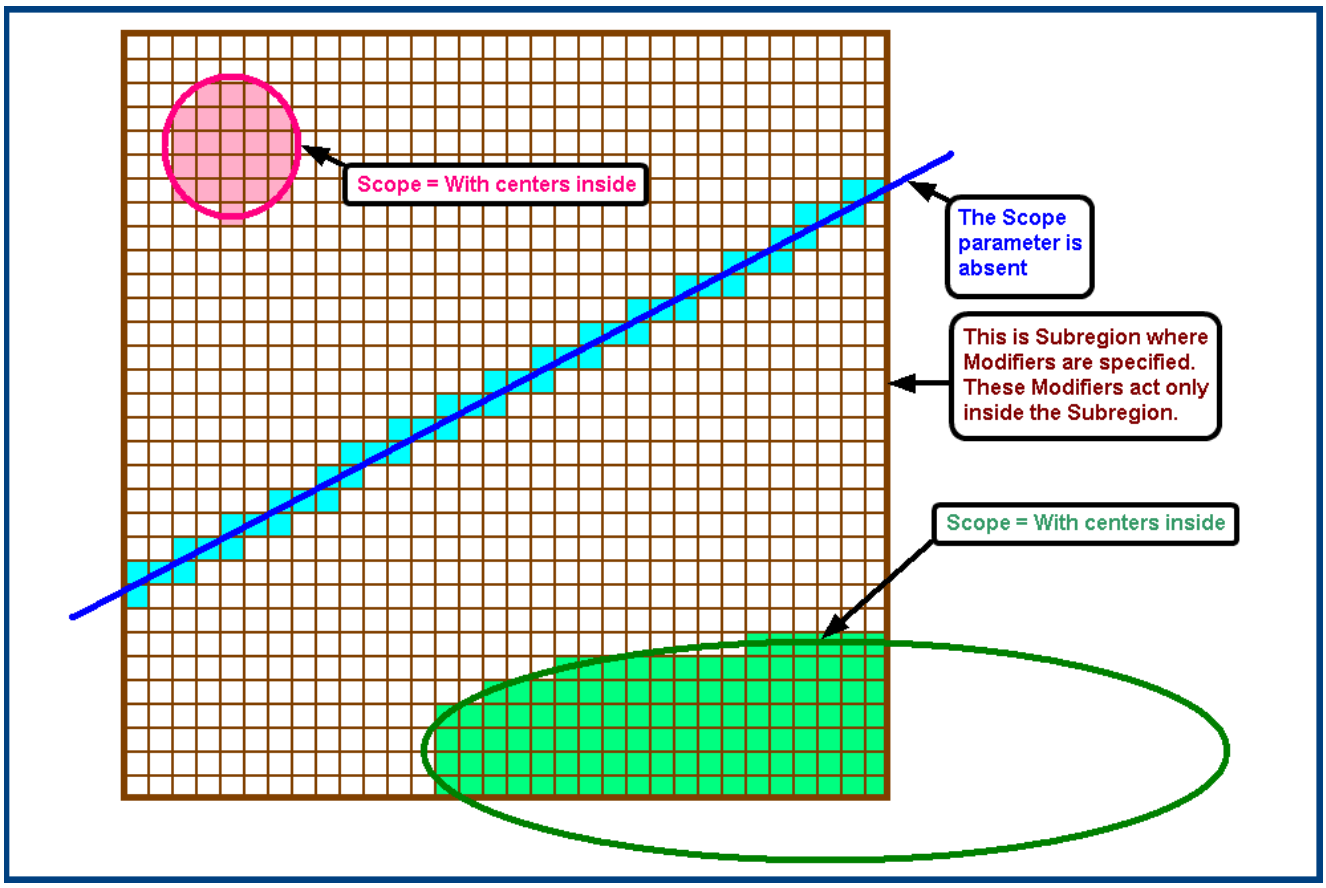
Any **Modifier** acts only in the computational **Subregion**, where it was specified. Space of a cell is defined by external borders of the cell *before the cell is split by surface of the computational subregion*.



The set of cells, where a **Modifier** is active, is determined *before the cells are split* by surface of the computational domain or by surface of some **Moving body**



Scope = All cells was set for **Modifiers** that are specified on the pink **Sphere** and the green **Ellipsoid**. The **Modifier**, which is specified on the blue **Line**, doesn't have the **Scope** parameter.



Scope = With centers inside was set for **Modifiers** that are specified on the pink **Sphere** and the green **Ellipsoid**.
The **Modifier**, which is specified on the blue **Line**, doesn't have the **Scope** parameter.

If a **Modifier** is set on an **Imported object**, then geometry of this **Imported object** must be closed or else an error occurs or the action of the **Modifier** (for example, **Setting variable**, **Volume force**, or **Resistance**) will spread through the whole computational **Subregion**.

Specifics of applying some Modifiers in cells that are either partially or completely filled by the Object

Results of applying some **Modifiers** might be corrected in cells that are *partially* filled by the **Object**.

Also it is possible to tune applying a **Modifier** so values specified in it would be added to the existing values in the **Object** instead of replacing them.

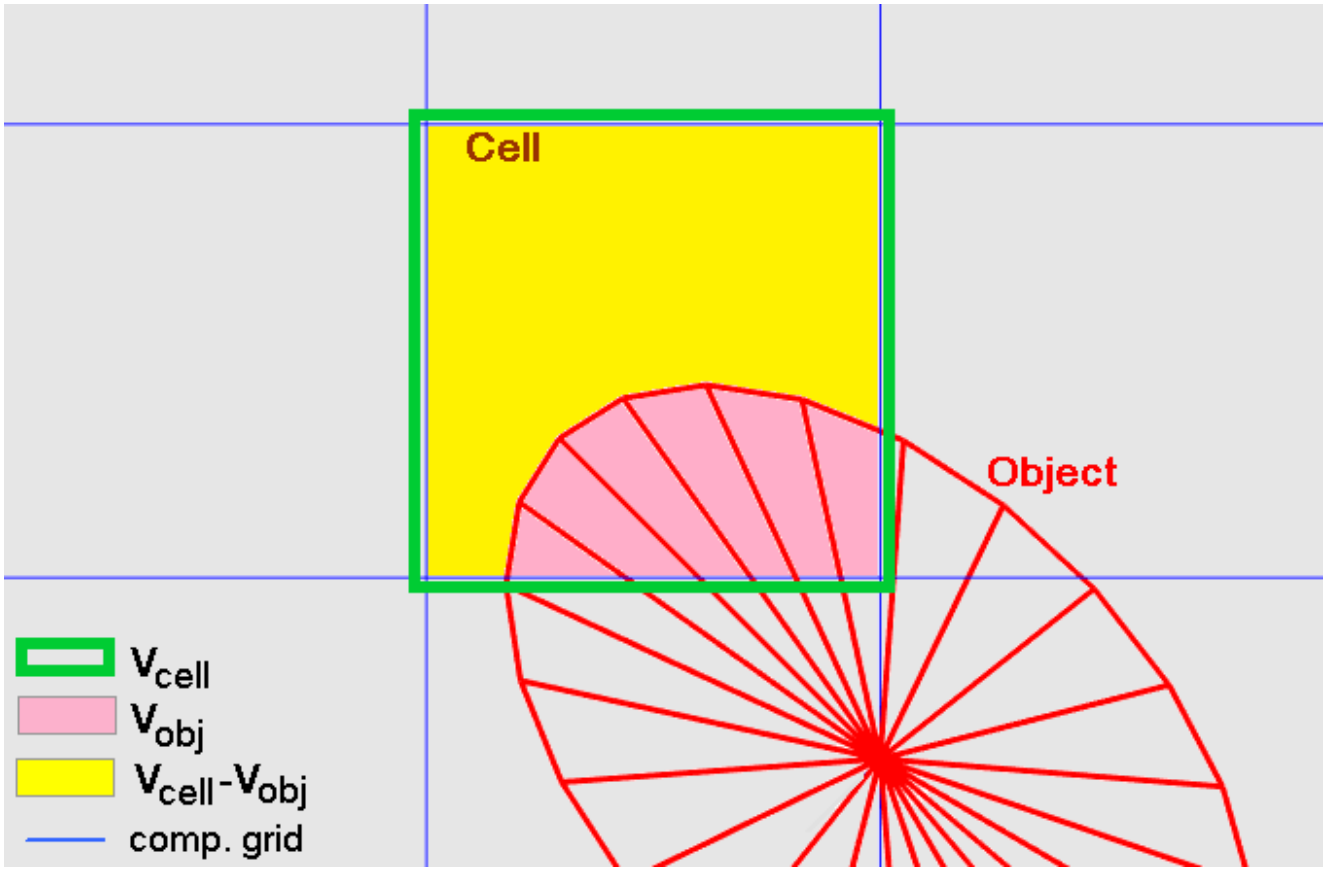
The following methods of applying some **Modifiers** within cells can be set:

- Completely replacing the **Variable**'s value in cells that locate fully filled by **Object** and also in cells that are even partially filled or just contacted by the **Object**. This is the "**Replace in full volume**" method.
- Changing the **Variable**'s value in the cell proportionally to the volume of the cell filled by **Object**. If the cell is filled by the **Object** fully, the new value will be the same value as specified in the **Modifier**, and when the **Object** partially fills the cell, the value will be replaced proportionally to the volume filled by the **Object**. This is the "**Replace in cropped volume**" method.
- Value of the **Variable** from the **Modifier** is added to the existing value in the cell, proportionally to the volume filled by the **Object** in the cell. This is the "**Add in cropped volume**" method.

This setting is specified in properties of the **Modifier** by the **Method** parameter and can be applied to **Modifiers** of the following types: **Setting variable I Volume force I Volume heat source I Resistance I Volume External charge I External Current I External Induction**.

Adequate setting the **Method** parameter allows you to improve accuracy of the simulation. For example, when **Method = Replace in full volume**, the total power of a **Volume heat source** will be equal to the specific volume heat power multiplied by volume of all concerned cells but not by volume of the **Object**, because heat will be produced at boundaries of the **Object** (in cells that are partially filled with the **Object**) with the same power as within the **Object**. When you set **Method = Replace in cropped volume**, heat production in cells at boundaries of the **Object** will be calculated proportionally to volumes of cell parts that are filled by the **Object** (so the total power of the **Volume heat source** will be equal to the specific volume heat power multiplied by volume of the **Object**).

Let's take a closer look at these methods. To gain a better understanding of our notation, see the illustration below:



On the illustration a cell is shown partially filled with a geometry **Object** of non-zero volume (in this example it is a cylinder with an elliptic base). Boundaries of the cell are marked with green contour.

The whole cell has volume V_{cell}. The part of the cell that is *within* the **Object**, is highlighted with pink and has volume V_{obj}. The part of the cell that is *out of* the **Object**, is highlighted with yellow and has volume V_{cell}-V_{obj}.

Value of the **Variable**, which is set by parameters of the **Modifier**, will be referred as f_{new}. Value of the same **Variable** that was in the cell before applying the **Modifier**, will be referred as f_{old}. Value of the **Variable** that will be result of applying the **Modifier**, will be referred as f.

Depending on the selected **Method** the following formulae will be used:

Method = ...	The general formula	Formula for cells that are fully filled by the Object (V_{obj}=V_{cell}, so the formula becomes simpler)
Replace in full volume	$f = f_{new}$ Value from the **Modifier** replaces the existing value.	$f = f_{new}$ The cell is fully filled by the **Object**, so methods **Replace in full volume** and **Replace in cropped volume** give the same result (value from the **Modifier** just replaces the existing value).
Replace in cropped volume	$f = f_{old}\frac{V_{cell} - V_{obj}}{V_{cell}} + f_{new}\frac{V_{obj}}{V_{cell}}$	

Method = ...	The general formula	Formula for cells that are fully filled by the Object ($V_{obj}=V_{cell}$, so the formula becomes simpler)
	Value from the **Modifier** replaces the existing value proportionally to the volume filled by the **Object** in the cell.	
Add in cropped volume	$f = f_{old} + f_{new}\frac{V_{obj}}{V_{cell}}$ To the existing value in the cell the value from the **Modifier** is added multiplied by the fraction of the volume filled by the **Object** in the cell.	$f = f_{old} + f_{new}$ In cells that are fully filled by the **Object** the value from the **Modifier** is added to the existing value.

For cells, in which a **Modifier** is applied built on a **Line**, a **Plane**, a **Set of sensors**, the same rules are used as for cells that locate entirely within an **Object** (see the table *"Formula for the Variable's value in cells in cells where the Modifier is applied and that locate entirely within the Object"* above). The same is also true for **Modifiers** that are built on the **Computational space**.

The similar parameter you can also find in properties of **Initial conditions**.

Order of applying Modifiers (priority of modifiers of the same type)

When two or more **Modifiers** should be applied to the *same cell of the computational domain*, the following rules are used:

- action of **Modifiers** of *different types* is combined
- when there are several **Modifiers** of the *same type*, only the **Modifier** is applied, which is below the others in the list of modifiers in the project tree (so as a **Modifier** is lower in the list, the higher is its priority).

User can change position of a **Modifier** in the list when required.

Examples:

- When there are one modifier **Resistance** and one modifier **Anisotropic resistance** (in any order) in the list, their actions are combined (both **Modifiers** are applied).
- When there are two modifiers **Resistance** in the list, then the modifier is applied, which is located lower then the other.
- Similarly, when there are two modifiers **Anisotropic resistance** in the list, then the modifier is applied, which is located lower then the other.

See also: section Folder «Modifiers».

6.12.1 Modifier «Moving body»

The modifier **Moving body** is set on an **Imported object** - the carrier of a moving body. The modifier that provides the **Object** with additional properties:

- ability to set boundary conditions on the **Object**'s surface
- movement and rotation abilities
- ability to set powers and torques affecting the **Object**

The resulting geometric body is regarded as part of the geometry model of the computational domain; in particular, a sub-grid resolution of the moving body surface is performed to take into account features of its shape.

Geometry of **Moving bodies** must conform to the same requirements that are applied to the *main geometry* (see section Geometry model of the computational domain: surfaces and subregions).

It is also not allowed a coincidence of surface of some **Moving body** and a surface of another **Moving body** or the main geometry.

Imported object, on which the **Moving body** is defined, can be created:

- based on a standard geometric object of finite volume (**Box**, **Cone/cylinder**, **Ellipsoid/sphere**)
- based on a file with geometric data. Also a *batch import* is possible, which simultaneously imports many geometry files and creates many **Imported objects** and **Moving bodies**.

Movement of the body relative to the computational grid makes it necessary to perform automatic computational grid plotting during project computation to coordinate the grid with new computational domain boundaries. Cells of the computational domain located within the moving body are considered as non-computed, i.e., for those cells, the computational model of physical processes is not set.

The parameters of a moving and rotating body can depend on time and other variables. Those dependencies are set as:

- constants
- functions defined by tables
- formulae

Parameters of updating a moving body

Computation of the actual moving body parameters is performed during updating of the **Moving body** modifier. The settings of that update are controlled by the **Update** group of parameters in the **Modifier**'s **Properties** window.

A **Moving body** update means the following:

- computation of the moving body's new position and its movement parameters (forces, torques)
- plotting the computational grid

Updating is performed for all bodies if it has been set for at least one moving body in a project containing several moving bodies.

Since calculation grid plotting takes significant time, it is recommended to disable updating all **Moving body** modifiers if none of the moving bodies move. If at least one **Moving body** update is enabled, the computational grid will be replotted for the entire volume independently of **Updates** of other moving bodies.

Also, for every update of at least one **Moving body** presented in the computational domain, the computational grid will be replotted for the *entire computational domain volume*.

Forces and torques affecting a moving body from the medium

Moments of inertia and rotation centers of bodies are set in the local CS of the moving body (that is, in the LCS-O of the **Imported object**).

It must be kept in mind that during body movement computation, the moment inertia matrix is inverted, so it is necessary to set all diagonal elements of the matrix, even if the body rotates around only one axis and setting inertia moments for the rotation other axes seems unnecessary.

Moving body's movement law

If the task statement presumes movement of a **Moving body** under the effect of forces or torques, then moment of inertia matrix must be defined correctly, so all its diagonal elements must be non-zero and must correspond to appropriate realistic mass characteristics. Otherwise, the movement of a **Moving body** cannot be computed correctly. If the action of torques is not taken into account, you can use a zero matrix.

The **Moving body**'s movement law, projected onto every axis of coordinates (**X**, **Y**, **Z**) is set in three time intervals:

- at the first interval, the body is at rest
- at the second interval, the body moves with linear or angular velocity set by the user
- at the third interval, the body movement is defined by effect of forces $\boldsymbol{F}_{ext} + \boldsymbol{F}_{hydr}$ (for translation movement) and moments of forces $\boldsymbol{T}_{ext} + \boldsymbol{T}_{hydr}$ (for rotation) and damping ratios (O_V for translation movement, O_ω for rotation). The user sets:
  - force $\boldsymbol{F}_{ext}$
  - moment of forces $\boldsymbol{T}_{ext}$
  - use of hydrodynamic force $\boldsymbol{F}_{hydr}$ or declining to use hydrodynamic force $\boldsymbol{F}_{hydr}$ for some or all axes of the absolute coordinate system
  - use of hydrodynamic torque $\boldsymbol{T}_{hydr}$ or declining to use hydrodynamic torque $\boldsymbol{T}_{hydr}$ for some or all axes of the absolute coordinate system
  - damping ratio O_V for translation movement
  - damping ratio O_ω for rotation

Damping ratios O_V and O_ω are used for hastening the onset of the stable flow mode near moving bodies. When those ratios equal 1, no damping is performed. The most efficient damping of moving body oscillations takes place with damping ratio values within 0.3 to 0.7.

See the detailed explanation of the **Moving body** movement law in the tables below:

Translation movement of a Moving body		
Interval	**The time (*t*) is within the range:**	**Formula and notes**
1	$0 < t < t_{V,i}$	0 The **Moving body** is at rest (not moving along axis i)
2	$t_{V,i} \le t < t_{F,i}$	$V_{0,i}(t)$ The **Moving body** is moving with linear velocity explicitly defined by the user (as a constant, formula or a table function)
3	$t \ge t_{F,i}$	$V_{(n+1),i} = O_V\left(\frac{m_n V_{n,i} + (F_{ext,i}(t) + F_{hydr,i}(t))\tau}{m_{(n+1)}}\right)$ The **Moving body** is moving in accordance with a formula taking into account the effects of selected forces and the O_V damping coefficient.

Notation:

i=x, y, z - index of the axis for which the velocity projections are set

$t_{V,i}$ - moment in time that designates the start of the body's movement along the i axis. Starting here and up to the time $t_{F,i}$, the velocity of the body is explicitly set by the user as a constant, formula or a table function.

$t_{F,i}$ - moment of time that designates the start of the body's movement in accordance with an iterative formula

$V_{0,i}(t)$ - linear velocity along the i axis, in the time interval $t_{V,i} \le t < t_{F,i}$

n - time step number

$F_{ext,i}(t)$ - the external force defined by the user (as a constant, formula or a table function, so it generally depends on time)

$F_{hydr,i}(t)$ - hydrodynamic force effecting the body in an absolute coordinate system, it calculates automatically. The user can set up hydrodynamic force application or cancel it on several or all of the absolute coordinate system axes.

τ - time step duration

O_V - damping coefficient

m - mass of the **Moving body** as a function of time

Note:

If $t_{F,i} = t_{V,i}$, then at the first time step after this limit is exceeded, the motion occurs at the speed of $V_{0,i}(t_{F,i})$. This is also true for the special case if $t_{F,i} = t_{V,i}=0$.

Rotation of a moving body		
Interval	**The time (*t*) is within the range:**	**Formula and notes**
1	$0 < t < t_{\omega,i}$	0 The **Moving body** is static (not rotating around the i axis)
2	$t_{\omega,i} \le t < t_{T,i}$	$\omega_{0,i}(t)$ The **Moving body** is rotating with angular velocity explicitly set by user (as a constant, formula or a table function)
3	$t \ge t_{T,i}$	$\boldsymbol{\omega}_{(n+1),i} = O_{\omega}(\mathbf{J}^{-1}{}_{(n+1)}\mathbf{J}_n\boldsymbol{\omega}_{n,i} + \mathbf{J}^{-1}{}_{(n+1)}(\mathbf{T}_{ext}(t) + \mathbf{T}_{hydr}(t))\tau)$ The **Moving body** is rotating in accordance with a formula allowing for torque moment action and damping coefficient O_{ω}.

Notation:

i=x, y, z - index of the axis for which the angular velocity projections are set

$t_{\omega,i}$ - the time until which the moving body does not rotate. Starting here and up to the time $t_{T,i}$ the angular velocity of the body is explicitly set by the user as a constant, formula or a table function.

$t_{T,i}$ - the time that designates the start of the body's rotation in accordance with the formula defining the damping coefficient effect and T_{user} and T_{hydr} torque moments

$\omega_{0,i}(t)$ - angular velocity in the time interval $t_{\omega,i} \le t < t_{T,i}$

n - time step number

$T_{ext}(t)$ - an external torque moment that is user-defined as a constant, formula or a table function, i.e., is commonly time-dependent

$T_{hydr}(t)$ - the hydrodynamic torque moment that is applied to a body in an absolute coordinate system and is calculated automatically. Users can toggle accounting for the hydrodynamic force on several or all of the absolute coordinate system axes.

τ - time step duration

O_{ω} - damping coefficient

J - the moment of inertia of the **Moving body**

Note:

If $t_{T,i}=t_{\omega,i}$, then, on the first time step after the time passes this value, the angular velocity of the rotation will be as of $\omega_{0,i}(t_{T,i})$. This is also true for the special case if $t_{T,i}=t_{\omega,i}=0$.

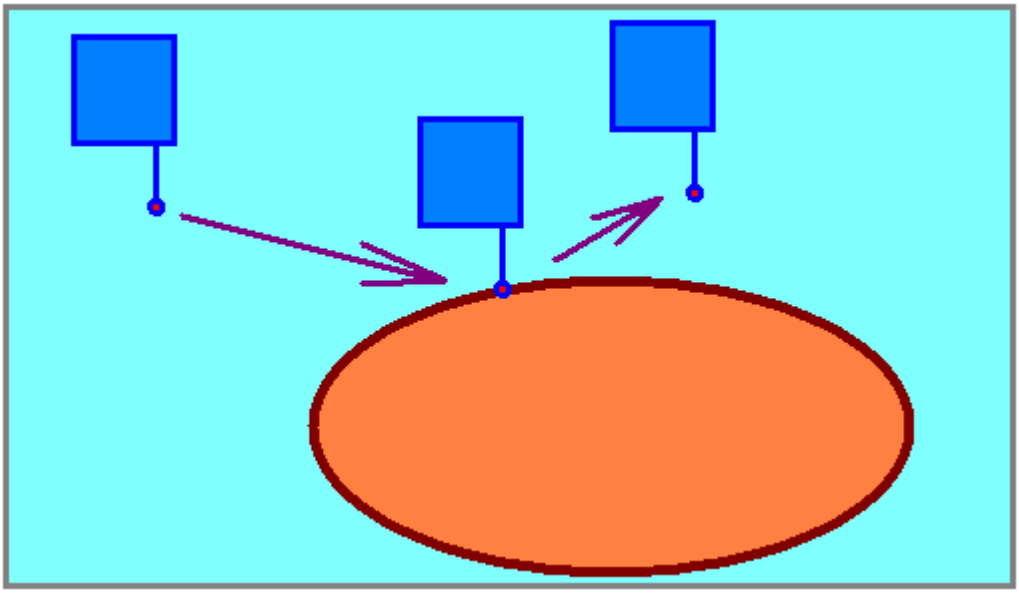
Limitations of the Moving body's movements

Movement of a **Moving body** can be limited by a **Plane** or another geometric **Object** by any of the following methods:

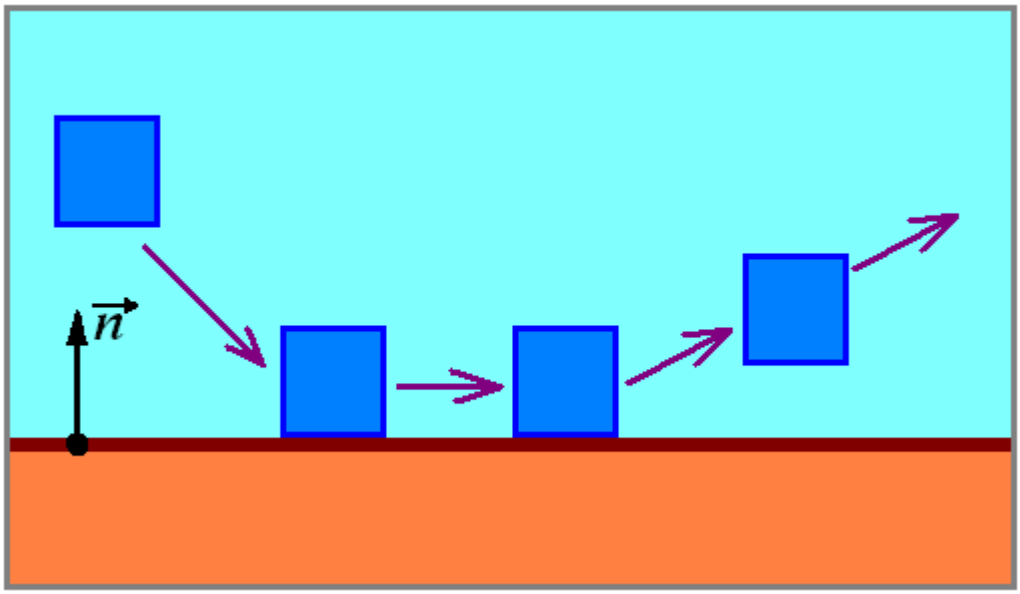
- A *limiting point* of the **Moving body** is specified, which is rigidly attached to the local coordinate system of the **Moving body**. This point must not intersect the surface of the specified standard geometric **Object** of finite size (i.e. a **Cone/Cylinder**, **Box**, or **Ellipsoid/sphere**) during the movement of the **Moving body**. Correct work of this limitation is not provided if the limiting **Object** moves (if a **Movement** is set on the **Object**). If a **Moving body**'s motion makes the limiting point contact with the limiting **Object**, then the motion stops until the calculated velocity vector points away from limiting surface (see the illustration). Thus the limiting point *cannot slide* along the surface of the limiting **Object**. If the **Point** initially was inside the **Object**, then it will continue to be inside it, and if it was outside the **Object**, it will continue to be outside. This limitation is set by the setting **Limitation > Method = Point-Object** in the properties of the modifier **Moving body**.
- A **Plane** is specified, which cannot be intersected by the **Moving body**. The **Moving body**'s movement is only possible in the positive half-subspace (where the normal vector of the **Plane** is directed). If the **Moving body** contacts the limiting **Plane**, it can slide along the **Plane**. Then after a time, the **Moving body**, driven by forces, can continue its movement away from the **Plane** (see the illustration). This limitation is set by the setting **Limitation > Method = Body above plane** in the properties of the modifier **Moving body**.

Also the movement of **Moving bodies** can be unrestricted having six degrees of freedom (movements and rotations) or restricted by two degrees of freedom, which assumes movement only along some axis and rotation only around this axis. In the latter case, the axis is defined by coordinates of two points in the absolute coordinate system.

See descriptions of user interface parameters, which specify these limitations, in the subsection *"Parameter groups «Limitation» and «Degrees of freedom» (limiters for movement and degrees of freedom of the Moving Body)"* of the section Folder «Modifiers».



Limiting of the movement of a **Moving body**, which is specified by the setting **Method = Point-Object**. When the limiting **Point** contacts the limiting **Object**, movement of the **Moving body** stops until the **Moving body**, driven by forces, goes away from the **Object**.



Limiting of the movement of a **Moving body**, which is specified by the setting **Method = Body above plane**. After a contact with the limiting **Plane**, the **Moving body** continues its movement as sliding along the **Plane**. Then after a time, the **Moving body**, can move away from the **Plane**.

Simulation of moving bodies in adjacent subregions

Moving bodies can intersect borders of **Subregions** and they can locate out of limits of **Subregions**.

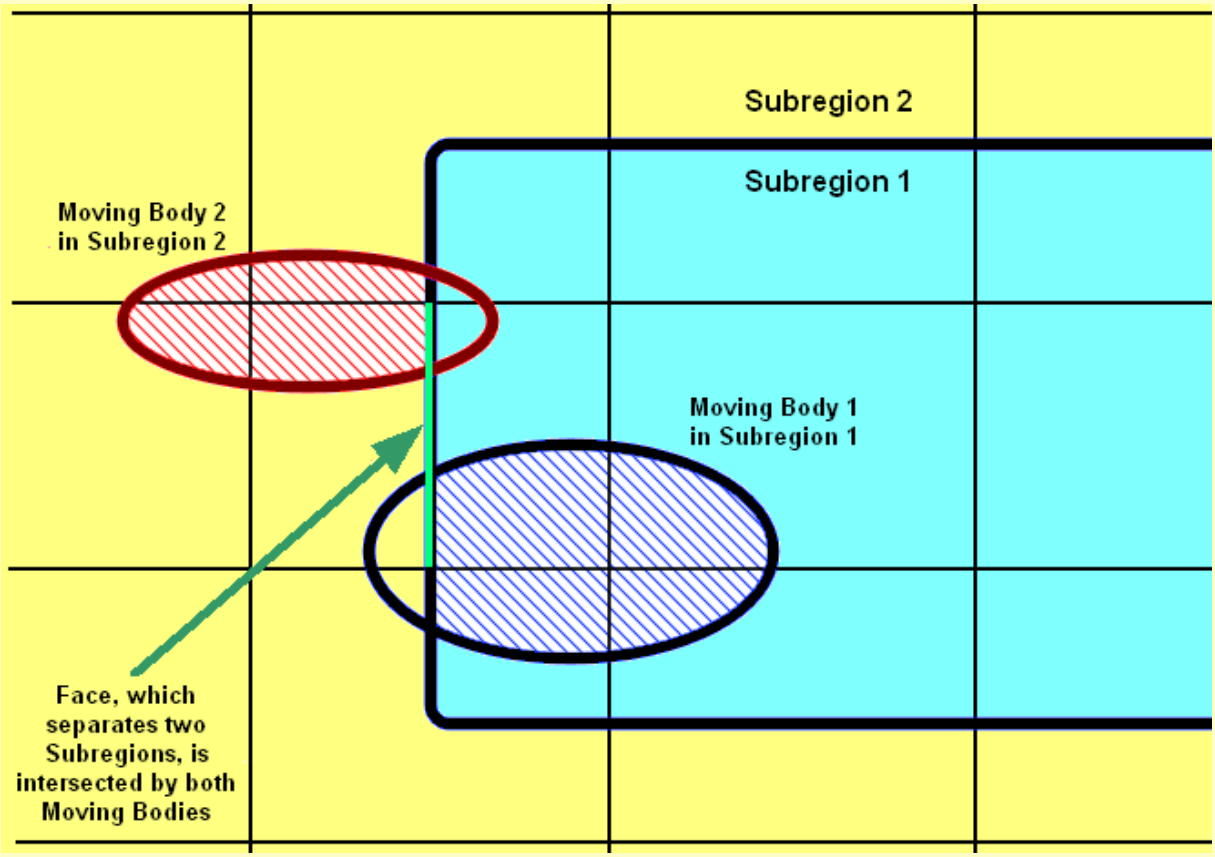
A **Moving body** can influence only that **Subregion**, where it locates.

The space within **Moving bodies** does not form new **Subregions** and does not require setting **Models**. But you have to set **Boundary conditions** on outer surfaces of **Moving bodies**.

You can use **Moving bodies** to simulate movable walls.

Moving bodies can reside (do not move); they are useful to form and modify the computational domain. For example, a wing that is inserted into a **Subregion** as a **Moving body**, can be promptly turned by a required for the computation angle without use of a CAD software.

If in both of adjacent **Subregions** there are **Moving bodies**, then *both surfaces* of the **Moving bodies** from different **Subregions** *must not intersect the common face between two cells, which separates these* **Subregions**. See the illustration below:



If this limitation is not complied, the normal work of the program is not guaranteed.

Requirements to the geometry of Moving Bodies

The geometry of an **Imported object**, on which a **Moving body** is created, conform the following requirements:

- it has to consist of one or several close surfaces
- *and* these surfaces can not intersect

If the **Imported object** consists of several *non-intersecting* close surfaces, then, when a **Moving body** is created on it, the following procedures will be done:

- all internal *nested* surfaces will be removed and will not be used in the computation
- if the surfaces are *not nested*, then they form a single **Imported object** consisting of several fragments, and all properties, which are to be given to the **Moving body**, will be allocated to all of these fragments.

If you attempt to create a **Moving body** on an **Imported object**, which consists of *intersecting* surfaces, the program will remove some of the surfaces so the remaining surfaces would not intersect. So the program will try to form a **Moving body**, but this might cause inappropriate results.

Prohibition of crossing some Boundary conditions by a Moving Body

You cannot apply crossing by a **Moving body** a surface on which the following **Boundary conditions** are set:

- **Conjugate all variables**
- **Periodic surface**
- **Sliding surface**

Special case: Displacing substance by a Moving Body from a volume, from which the substance cannot go out

If a **Moving body** displaces substance from a volume, from which the substance cannot go out, transferring the data can be blocked and multiple records about this situation are written into the **err**-file of the project.

Example:

```
...
23.10.15 00:36:50 ****** Iteration is = 190 ******
[Error]: [proc=0, code=0x80090006]  Error in Build Grid:
MovingBody::TransferData: Not all the data were transfered. Vol=3.8067573147781e-010
CellCount=8
23.10.15 00:37:20 ****** Iteration is = 192 ******
[Error]: [proc=0, code=0x80090006]  Error in Build Grid:
MovingBody::TransferData: Not all the data were transfered. Vol=1.842944922036e-009
CellCount=19
...
```

Such records in many cases can be considered as result of the simulation's definition and they do not prevent the computation of the project.

See also:

- Description of user interface for setting properties of a **Modifier** see in the section Folder «Modifiers».
- Operations with **Moving bodies**, including procedures of their creation, see in the section Operations with Modifiers.

6.12.2 Modifier «Setting variable»

The **Setting variable** modifier specifies a certain value of the variable (from the variables presented in the phase, which is set in the computational subregion) in the **Object** of the **Modifier**.

At each iteration, the value, which is set by the modifier, replaces the value, which was obtained from the previous iteration. This replacement is doing prior to starting the calculation on the current iteration. Upon completion of the current iteration, the value in the cells may vary and differ from the values that were set by the **Modifier**.

Parameters of the modifier settings are configured in the **Properties** window of the corresponding element in the **Modifiers** folder in the project tree, see section Folder «Modifiers».

6.12.3 Modifier «Volume force»

The modifier **Volume force** specifies a certain value of the force acting on a moving medium in the **Object** of the **Modifier**.

This power is included in the motion equation as a separate term $\boldsymbol{F}_{\text{user}}$.

Modifier parameters are set in the **Properties** window of the corresponding element **Modifiers** folder in the project tree (see section Folder «Modifiers»).

6.12.4 Modifier «Volume heat source»

The modifier **Volume heat source** specifies a certain intensity of a source or drain in a volume of some thermally conductive area.

This source/drain is included in the equation of heat transfer by a separate term Q_{user}.

The user specifies the heat generation per unit of volume [W/m^3] for the volume of the **Modifier**, see section Folder «Modifiers».

Please note, that if in properties of this **Modifier** you specify **Method = Replace in full volume**, then the total power of the **Volume heat source** will be equal to the specific volume heat generation multiplied by the volume of the **Modifier** but not by the volume of the **Object**, on which the **Modifier** has been built (see explanations in subsection *"Specifics of applying some Modifiers in cells partially or completely filled by the Object"*).

6.12.5 Modifier «Ignition / extinction zone»

The modifier **Ignition/extinction zone** starts/stops combustion in the volume of the **Object**.

This modifier can only be set when the set parameter **Combustion Mass transfer** elements of the **Physical process**.

Modifier acts either in a specific phase or all phases, in which the specified physical process of combustion.

When you select **Type= Ignition**, the **Ignition/extinction zone** modifier is used to initiate the combustion in the case where the ambient temperature is below the temperature of combustion, for example, to simulate a spark or flame ignition.

When you select **Type = Extinction**, the **Ignition/extinction zone** modifier is used to turn off the combustion, i.e. does not allow the chemical reaction of a compound of fuel with an oxidant (its corresponding equations do not apply).

Recommendation: The **Ignition/extinction zone** modifier does not make sense to apply the use of the combustion model **Zeldovich**. In which combustion takes place in all cells, where is there an oxidizer and fuel.

Modifier settings are configured in the **Properties** window of the corresponding element **Modifiers** folder in the project tree, see section Folder «Modifiers».

6.12.6 Modifier «Resistance»

The modifier **Resistance** specifies the value of the volume defined hydrodynamic resistance force acting on a moving medium.

This force is proportional to the velocity vector V (Darcy's law), is included in the motion equation individual term (-DV).

The result of the application of this modifier is identical to the **Volume force** modifier i.e. modifier **Resistance** may be replaced by a **Volume force** modifier. **Resistance** modifier introduced for the convenience of reference resistance forces in problems with the flow in porous media.

The coefficient D can be defined by a constant formula or table.

For example, in the simulation of flow in a homogeneous isotropic porous medium at high Reynolds numbers, the coefficient D can be set by this formula:

$$D = \frac{\mu}{\alpha} + C\frac{1}{2}\rho|\mathbf{V}|$$

where:

μ- The dynamic viscosity

α- Porosity

C- coefficient of inertia.

In the calculation of the equations of motion, this ratio is multiplied by the speed, resulting in a non-linear formula will take the form of Darcy's law (see [13], [14]).

Modifier settings are configured in the **Properties** window of the corresponding element **modifiers** folder in the project tree, see section Folder «Modifiers».

When modifiers **Resistance** and/or **Anisotropic resistance** are used, a stable and accurate solution is only possible when the order of magnitude of diffusion CFL is 1.

See also: Modifier «Anisotropic resistance»

6.12.7 Modifier «Anisotropic resistance»

Modifier **Anisotropic resistance** is defined by 6 elements of symmetric matrix of anisotropic resistance $\hat{D}$:

$$\begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix}$$

This matrix determines the volume force of anisotropic resistance exerted onto a moving medium (term $-\hat{D}V$ in the motion equation).

Anisotropic resistance can manifest itself in structures characterized by non-uniform spatial orientation of pores, micro-channels, or layers such as:

- fibrous and tubular materials,
- sandwiched materials.

In *FlowVision*, the matrix of anisotropic resistance is specified in the absolute coordinate system (ACS). Its elements are specified in the **Properties** window of the corresponding element of the Preprocessor tree (created in the Modifiers folder).

Obtaining the matrix of anisotropic resistance from the matrix specified in another coordinate system

The matrix of anisotropic resistance is defined in the absolute coordinate system (ACS). In the local coordinate system (LCS), related to the fibers and layers of a porous material, this matrix often has diagonal form. In order to obtain the matrix in ACS from the matrix in LCS, it is necessary to sequentially perform rotation operations around the ASC axes. Each rotation operation is determined by a *rotation matrix* (see the table below):

Matrices of rotation around coordinate axes of ACS	
Coordinate axis	**Rotation matrix A and inverse matrix A^T**
X	$A_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}$ $A_x^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{pmatrix}$
Y	$A_y = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix}$ $A_y^T = \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix}$
Z	$A_z = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $A_z^T = \begin{pmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Rotation around coordinate axis i assumes transformation:

$$\hat{D}' = A_i \hat{D} A_i^T \qquad (1)$$

Here

D is the matrix of anisotropic resistance in LCS

$\hat{D}'$ is the matrix of anisotropic resistance in ACS

Rotation around two or three axes assumes sequential execution of corresponding transformations (1).

When modifiers **Resistance** and/or **Anisotropic resistance** are specified, stable and accurate solution is only possible when diffusion CFL is 1.

See also: Modifier «Resistance».

6.12.8 Modifier «Anisotropic thermal conductivity»

Modifier **Anisotropic thermal conductivity** (**Anis. therm. conductivity** in the interface) is defined by 6 elements of symmetric matrix of anisotropic thermal conductivity $\hat{\boldsymbol{D}}$:

$$\begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix}$$

When anisotropic heat transfer is specified in certain region of computational domain, the heat flux is computed as follows:

$$\boldsymbol{J}_q = -\lambda \hat{\boldsymbol{D}} \cdot \nabla T + \sum_{i=species} h_i \boldsymbol{J}_i$$

Here

λ is the coefficient of molecular thermal conductivity

h_i is the thermodynamic enthalpy of **Substance** i

$\boldsymbol{J}_i$ is the diffusive flux of **Substance** i

Anisotropic thermal conductivity can manifest itself in structures characterized by non-uniform spatial orientation of pores, micro-channels, or layers such as:

- fibrous and tubular materials
- sandwiched materials

In *FlowVision*, the matrix of anisotropic thermal conductivity is specified in the absolute coordinate system (ACS). Its elements are specified in the **Properties** window of the corresponding element of the Preprocessor tree (created in the Modifiers folder).

Obtaining the matrix of anisotropic thermal conductivity from the matrix specified in another coordinate system

The matrix of anisotropic thermal conductivity is defined in the absolute coordinate system (ACS). In the local coordinate system (LCS), related to the fibers and layers of a porous material, this matrix often has diagonal form. In order to obtain the matrix in ACS from the matrix in LCS, it is necessary to sequentially perform rotation operations around the ASC axes. Each rotation operation is determined by a *rotation matrix* - see Modifier «Anisotropic resistance». The corresponding mathematical transformations are also described in that section.

6.12.9 Modifiers «Volume External charge», «External Current», «External Induction»

Modifiers **Volume External charge**, **External Current** and **External Induction** are used in simulations of electromagnetic hydrodynamics (EMHD).

They specify respectively:

- volume electric charge, [C/m^3]
- vector of the electric current density, [A/m^2]
- vector of magnetic induction, [T]

6.13 Initial conditions

Initial conditions are used to specify fields of variables in the computational domain to accelerate convergence of the solution.

Initial conditions specify values of variables only once, before starting the computation. By default, **Initial conditions** are set in the whole computational space of the **Subregion**. **Initial conditions** become active when a computational **Model** is set for the **Subregion**.

By default (if no **Initial conditions** are applied), values in the **Subregion** will obtain zero values (zero relative values for temperature and pressure, for diameters of dispersed particles the default values are specified in properties of the dispersed Phase). Thus, in absence of **Initial conditions**, the computation starts from the initial state of stable medium with reference values of temperature and pressure (T_{ref} and P_{ref}).

The user can set different initial conditions in various parts of the computational domain.

Initial conditions are created as a combination "**Object + Initial data**", where:

- **Object** defines the volume where the initial conditions are set
- **Initial data** define the numeric values of variables

The following can be as **Objects**:

- **Computational space** - the internal volume of the computational subregion
- imported or standard geometric objects

If **Initial conditions** are set on an **Imported object**, then geometry of this **Imported object** must be closed or else the action of the **Initial conditions** might spread through the whole computational **Subregion**.

If the **Initial conditions** are linked to a *non-zero-volume* **Object** (when surface of the **Object** forms a closed space), they are applicable in all cells of the computational subregion that completely or partially locate inside the **Object**.

If the **Initial conditions** are linked to a standard *non-zero-volume* **Object** (a **Line**, a **Plane**, or a **Set of sensors**), they are applicable in those only cells of the computational subregion that are intersected by that **Line** or **Plane** or contain **Sensor(s)**.

The variables, for which **Initial conditions** are set, are determined by a model specified in the **Subregion**.

Outside the **Object**, on which the **Initial conditions** are set, the computation will start from default (zero) values.

Order of applying Initial conditions and Moving bodies

Initial conditions are applied before the first step of the computation and before inserting **Moving bodies**.

In the first iteration after inserting **Moving bodies**, the volume occupied by the **Moving bodies** is totally removed from the computation, rather than displaced by the **Moving bodies** into the surrounding space.

Order of applying Initial conditions and the computational grid Adaptation

Adaptation on the main geometry and on object of adaptation (except **Moving bodies**) is done before applying **Initial conditions**.

Adaptation on **Moving bodies** is done after applying **Initial conditions**.

Sequence of application of initial conditions

The initial conditions in volumes are applied according to the top-to-bottom list in the **Initial conditions** folder in the project tree. Thus, if volumes of initial conditions match or intersect, the **Initial conditions** placed lower in the list are of higher priority.

Because of that, it is necessary to take into account the order used for elements **Initial conditions #N** in the project tree.

Initial condition values

Values of **Initial conditions** are set in elements **Initial data #N** as constants, functions in table presentation or by formulae according to which the values in computational cells are computed. **Initial data #N** can only be a function of coordinates, since other variables at the moment of project computation start are not defined.

Also, it is possible to use user-defined constants when creating formulas for **Initial data #N**.

When **Initial conditions** are set, you also have to specify exactly, which of the predefined **Initial data > Initial data #N** will be used.

If **Initial conditions** are *not* set in some part of a **Subregion**, then they will be assumed zero there.

Applying Initial conditions in cells partially or completely filled by the Object

Properties of elements **Initial condition #N** include the parameter **Method** that specify the method of the computation in cells located entirely within the **Object** and in cells that are filled by the **Object** partially. The following options are possible:

- **Method = Replace in full volume**: replacing existing values of variables in the cell, no matter if the cell is entirely locates within the **Object** or is partially filled by the **Object** (even touching the **Object**).
- **Method = Replace in cropped volume**: replacing existing values of variables in the cell proportionally to the volume that is occupied by the **Object** in the cell (this is similar to **Method = Replace in full volume** in all cells except the border cells that are *partially* filled by the **Object**).
- **Method = Average in cropped volume**: averaging the values, which are set by the initial conditions, and existing values in the cells, proportionally to the volume that is occupied by the **Object** in the cell.

Non-zero existing values that are present in cells before applying an **Initial condition #N**, are determined by applying other elements **Initial condition #N** that are specified above in the list in the **Initial conditions** folder.

The following formulae are applied (see illustration, notations, and explanations for the similar **Method** parameter in properties of **Modifiers**, *"Specifics of applying some Modifiers in cells partially or completely filled by the Object"*):

Method = ...	General formula	Formula for cells that are fully filled by the Object ($V_{obj}=V_{cell}$, so the formula becomes simpler)
Replace in full volume	$f = f_{new}$	$f = f_{new}$
Replace in cropped volume	$f = f_{old}\frac{V_{cell} - V_{obj}}{V_{cell}} + f_{new}\frac{V_{obj}}{V_{cell}}$	
Average in cropped volume	$f = \frac{f_{old}V_{cell} + f_{new}V_{obj}}{V_{cell} + V_{obj}}$	$f = \frac{f_{old} + f_{new}}{2}$

For standard **Objects** **Line**, **Plane**, **Set of sensors**, and **Computational space** for all cells, contacting with the **Object**, the same algorithms will be applied as those, which are used for cells that locate entirely within the **Object**.

See also: section Folder «Initial conditions».

6.14 Computational grid

The computational grid is built in *FlowVision*in in two stages:

- at the first stage the initial computational grid is specified in **Pre-Postprocessor**;
- at the the second stage **Solver** refines the initial computational grid in accordance to the user-specified criteria.

Criteria of the adaptation

Criterion of adaptation is a condition that determines splitting or merging computational cells in the specified object up to the specified level.

If in the same part of the computational domain several criteria for adaptation are active, the higher priority has the criterion, which splits the cells, comparing to the criterion, which merges the cells.

Use of grids with non-computational directions (2D and 1D grids)

For some simulations it makes sense to specify directions, along which the computational grid will always contain only one cell.

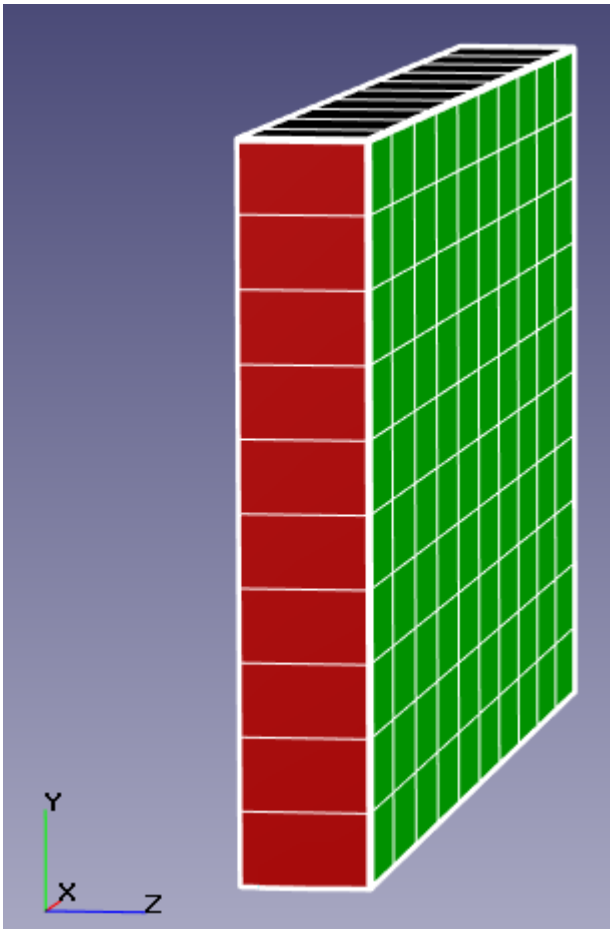
Such directions (along some of axes of the absolute coordinate system) are specified in properties of the element **Initial grid** by either parameter **Plane** or **Direction** that are available when **Grid structure = 2D | 1D** is selected there.

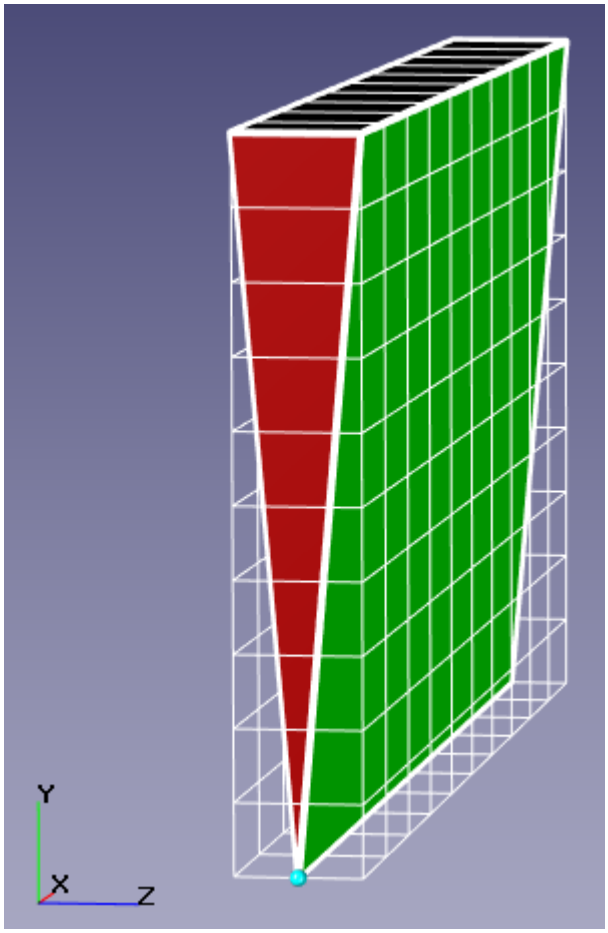
Also in the non-computational directions adaptations of the computational grid will be blocked.

Recommendations for use of a 2D grid (one direction is non-computational)

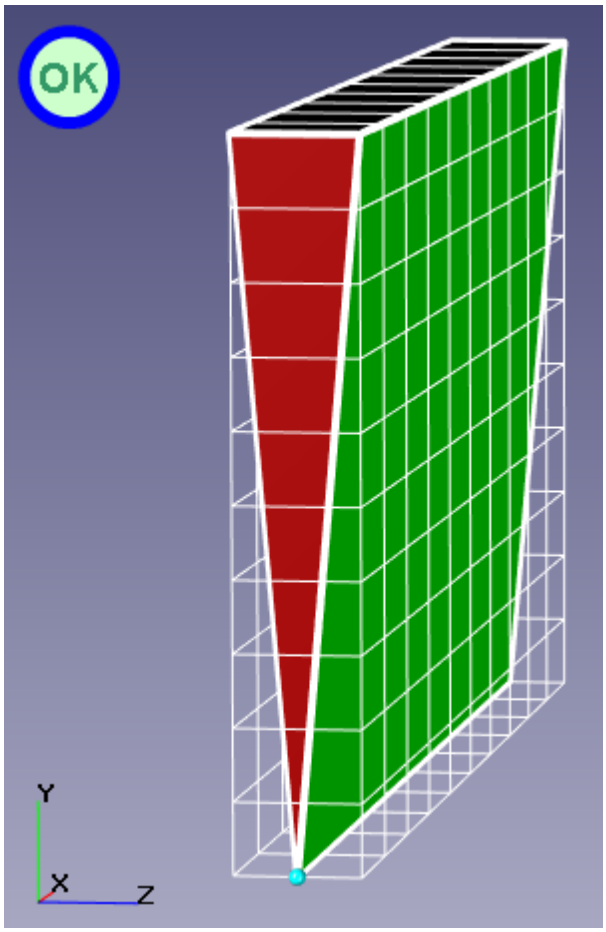
2D grids can be applied in planar and sectoral problem settings. Recommendations of their use are:

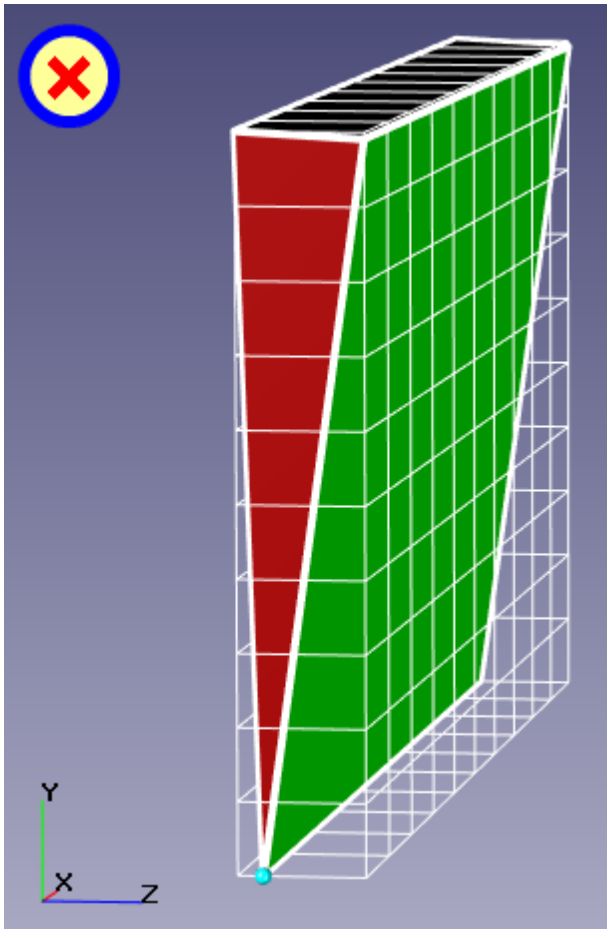
- The computational domain be symmetrical relating to the non-computational direction.
- In a planar problem setting, boundaries of the computational domain should be either perpendicular or parallel to the non-computational direction.
- In a sectoral problem setting, the plane of symmetry of the sector should be perpendicular to the non-computational direction.
- The type of **Boundary conditions** on surfaces limiting the computational domain from the side of the non-computational direction are to be either **Symmetry** or **Connected**.
- You should use the **Absolute** criterion for revealing small cells (**Small Cells > Criterion = Absolute** in properties of the appropriate **Phase Limiter**) in all cases except simulating the icing and some other cases when there are explicit recommendations to specify the **Relative** criterion for revealing small cells.

a) 

b) 

A 2D grid can be set for projects with planar (a) and sectoral (b) problem setting.
Grid structure = 2D and **Plane = XY** are set in properties of the **Initial grid** (so the non-computational direction is **Z**).
On the surfaces that are shown in green, only either **Symmetry** or **Connected** boundary conditions are to be set.

a) 

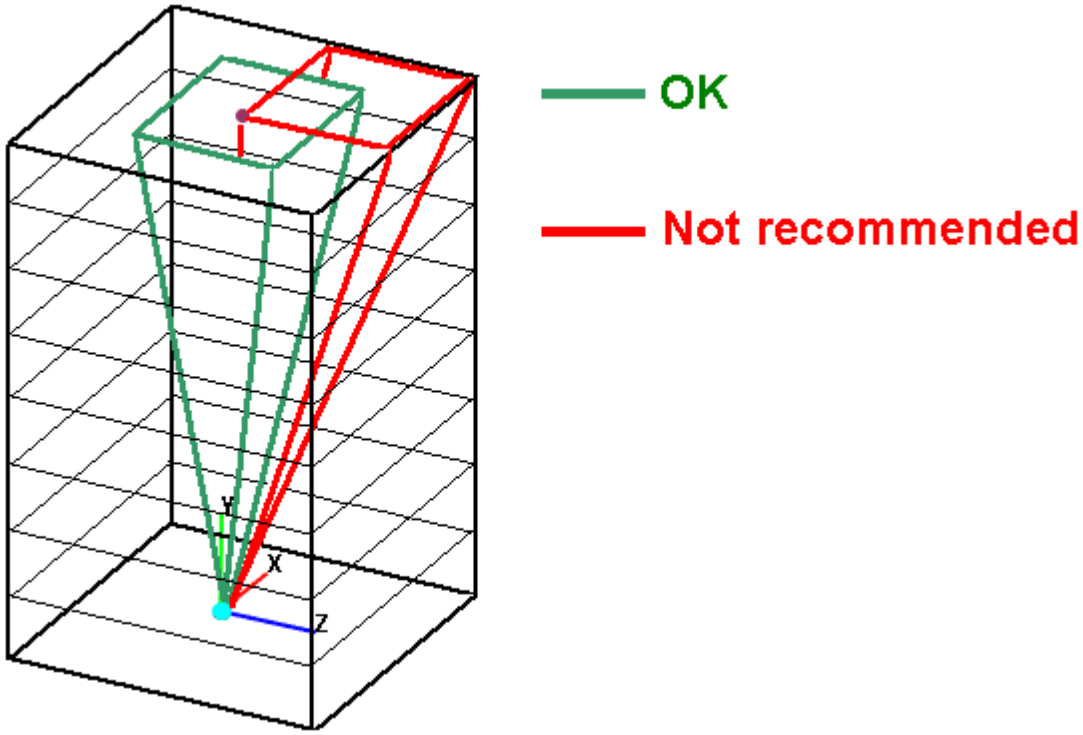
b) 

Examples of correct and incorrect geometry model in a 2D sectoral problem setting (**Grid structure = 2D** and **Plane = XZ** are set in properties of the **Initial grid**):
a) correct problem setting, the computational domain is symmetrical; б) incorrect problem setting, the computational domain is not symmetrical

Recommendations for use of an 1D grid (two directions are non-computational)

1D grids can be applied in problem settings with axially symmetric geometry.

When you use 1D grids follow the same recommendations as for 2D grids taking into account that problem setting has now two non-computational directions.

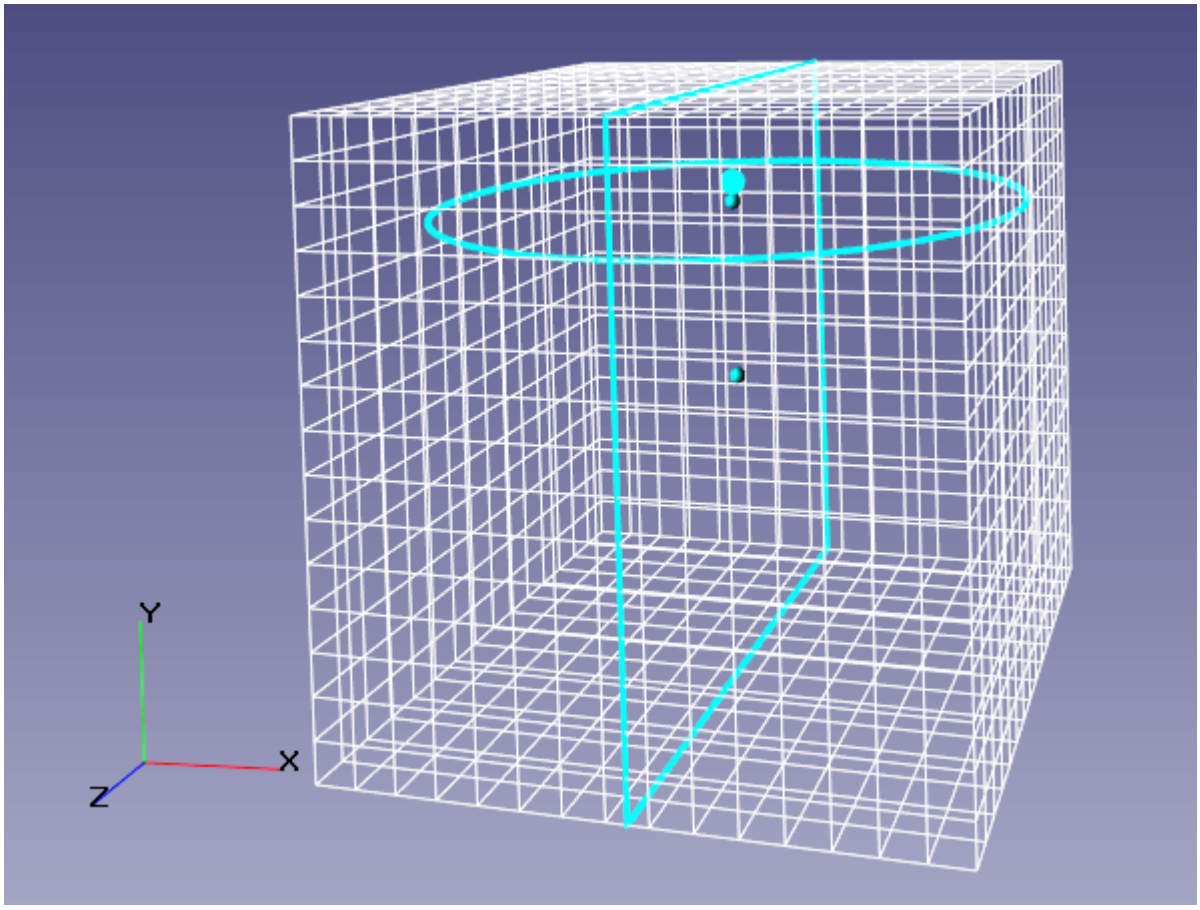


Examples of correct and incorrect geometry model in an 1D problem setting.
Grid structure = 1D and **Direction = Y** are set in properties of the **Initial grid** (so the non-computational directions are **X** and **Z**).

6.14.1 Initial computational grid

Initial grid is generated (built) in a rectangular parallelepiped, which includes the computational domain.

Faces of this parallelepiped are parallel to axes of the absolute coordinate system.



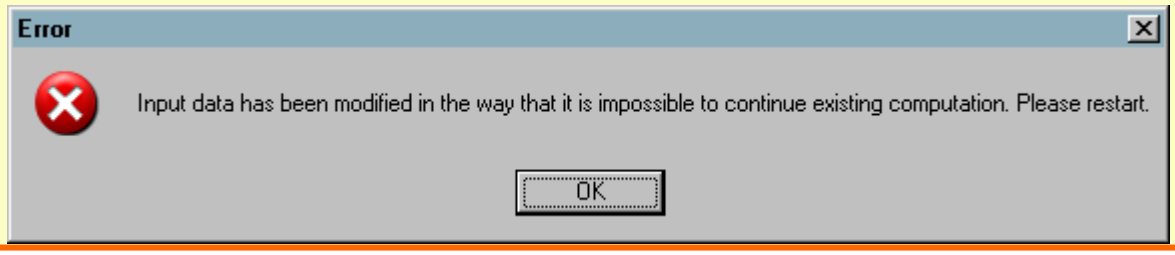
This is the volume of a box in which to build initial grid. Initial grid cell, trapped in the computational domain, are calculated, the cells that lie outside the boundaries of the computational domain - a non-computational domain.

In *FlowVision* solution of the system is performed on a Cartesian grid, which automatically locally milled in accordance with predetermined criteria. Such grid is created in two steps. At the first stage is given initial Cartesian grid, and the second - the area and the local grid refinement criteria, after which the system automatically creates a computational grid.

Initial Cartesian grid is defined as a three-dimensional grid along each axis of the Cartesian coordinates. Dimensional grid can be given a uniform (constant step) or irregular variable pitch.

When the initial grid for the project, which had already been initiated computation, renewal calculation impossible. You can only run the calculation "from scratch."

When you try in such a situation to continue where it left off computation, the message **"Input data has been modified in the way that it is impossible to continue existing computation. Please restart."**:



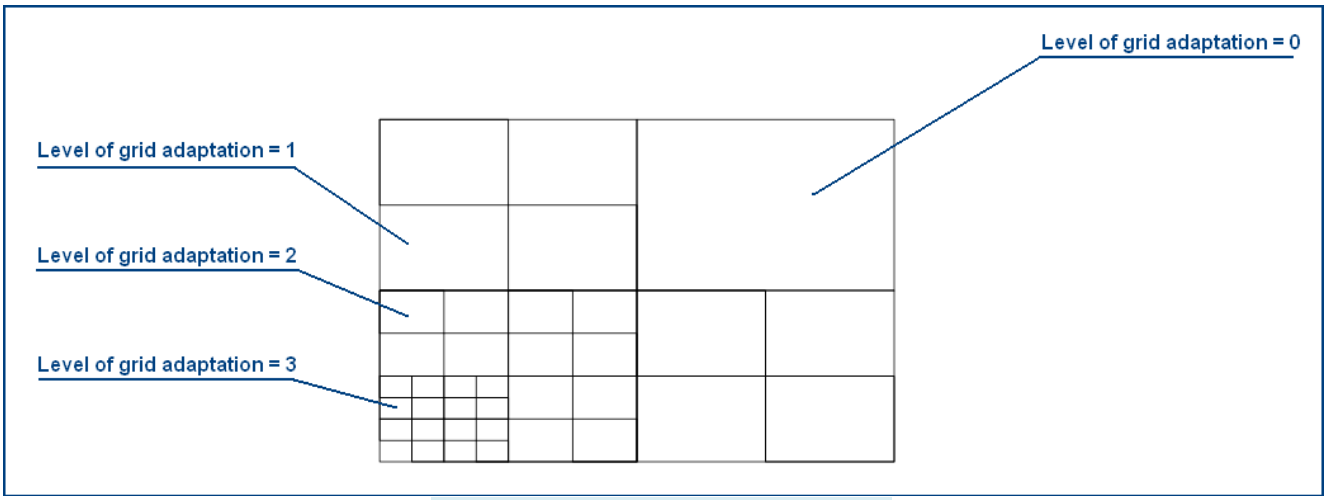
6.14.2 Splitting and merging cells of grid

Partition of the grid cell divides in half of each edge of the parallelepiped cell so as to obtain 8 equal boxes cell next level of smallness.

Initial cell of the computational grid cells are considered to be*zero, the result of a single partition - the cells of the first* level,etc. Example thus constructed a two-dimensional grid is shown on the illustration.

Operation, return to the partition, is the fusion of cells, which is produced by combining the neighboring 8 boxes in one.

In automatic mesh refinement monitored requirement that the common edges and ribs may have cells only at levels of splitting which differ by no more than one (i.e., for example, the cell level 3 of splitting may have common edges and the edges only with cells 2nd, 3rd and 4th levels of splitting).

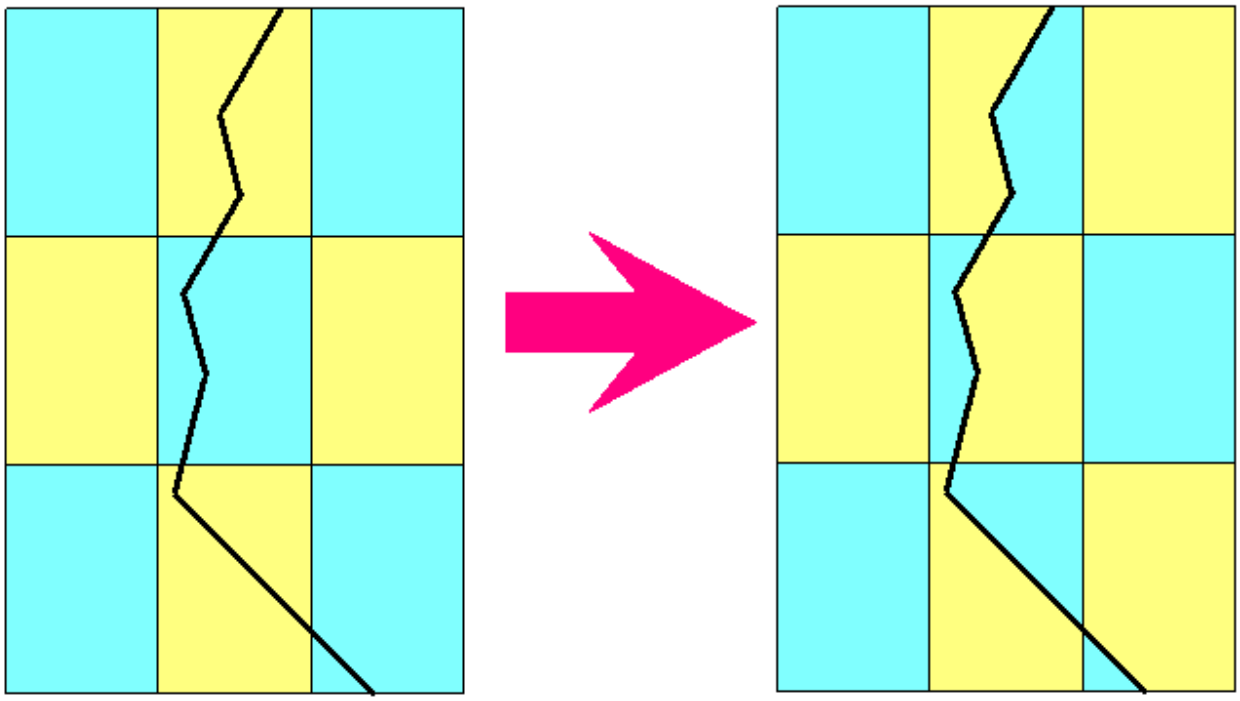


Levels of splitting the computational grid

6.14.3 Subgrid geometric computation region model resolution

The geometry model of the computational domain is a set of surfaces formed by facets.

With the application of a geometry model of the computational domain into the calculated Cartesian grid facet surfaces intersect the calculated cell-dividing cells parallelepipeds the multifaceted cell of arbitrary shape. The number of cells thus increases.



Splitting cells of the computational grid surface of the geometry model of the computational domain increases the number of cells.

6.14.4 Grid refinement control

FlowVision implements the following methods of grid refinement:

- **Adaptation** (a simple one) according user's adaptation criteria, regardless of calculation results (see sections Subgrid geometric computation region model resolution and Adaptation). This method is employed to build the computational grid before the computation process or while it is in progress (if the grid is being rebuilt).
- **Adaptation by condition** - This method allows to apply adaptation in areas where values of specified variables fall into the specified ranges.
- **Adaptation to solution** - This method is used only if the solution is in progress and certain conditions are met. The conditions are related to specific features of the computational variables' fields and are referred to as properties of adaptation to the solution.

All types of adaptations can be set in volumes or on surfaces of geometry **Objects** and their action is limited by specified **Subregions**.

Simple **Adaptations** and **Adaptations by condition** can also be set on **Boundary conditions**.

If an adaptation is set in the volume of an **Imported object**, then geometry of this **Imported object** must be closed or else the action of the adaptation could go out and spread through all **Subregions** where the adaptation is set.

6.14.5 Adaptation

Adaptations are elements of a project that define the splitting or merging parameters of those cells of the computational grid that:

- belong to the specified object volume
- are adjacent to the specified object surface
- are adjacent to a surface that has certain boundary conditions defined

Adaptation always impacts the volume or the surface of an **Object**.

An **Object** can be an **Imported object** or a standard geometric object. Surfaces may be represented by open imported objects or groups of facets, on which some boundary conditions are defined. One object or a surface may be linked to several **Adaptations**.

An adaptation is created inside a specific computational **Subregion** and has no effect outside its boundaries. Thus, if an object that has an adaptation attached to it lies outside the **Subregion** boundaries, this adaptation is only applicable to those parts of an object that are located within the computation **Subregion**.

If an adaptation is connected to an object of a finite volume, it has an effect on all cells of the computational subregion that are partially or fully associated with this body. If an adaptation is connected to a **Line** or **Plane**, it has an effect on all cells of the computational **Subregion** that are intersected by this **Line** or **Plane**.

The main parameter of **Adaptation** is the grid refinement level. The number of cells of the computational grid, which will be after the adaptation, is not defined beforehand (compare to the Adaptation to solution).

The two main adaptation types: splitting and merging

The main adaptation types perform computational grid refinement or agglomeration to a specific level N via *splitting* or *merging* cells.

The adaptation type is defined in the project when selecting the **Split** or **Merge** value of the **Split/Merge** property (see detailed GUI description in section Folder «Adaptation»).

Setting an adaptation to split (merge) cells up to level N means that every cell's adaptation levels in the adaptation zone will not be lower (higher) than N.

With several adaptation conditions in effect, the split condition has higher priority then the merge condition. The cells can only be merged if no split conditions are defined for them.

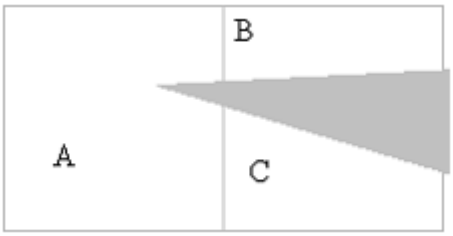
Additional adaptation types: enhancement

FlowVision has *additional adaptations* available to automatically resolve gaps and multiple complex geometric regions smaller than the original cell. This improves convergence as it enables avoidance of negative effects imposed by concave cells.

These adaptations are defined in the project after selecting the **Improve** value of the **Split/Merge** parameter.

The **Several neighbors** adaptation is applied to circumvent a situation when the edge of cell **A** is split by a body that results that cell **A**, is adjacent to both cells **B** and **C** through at edge.

The **Several neighbors** adaptation sequentially splits **A**-type cells until this cell type is eliminated.



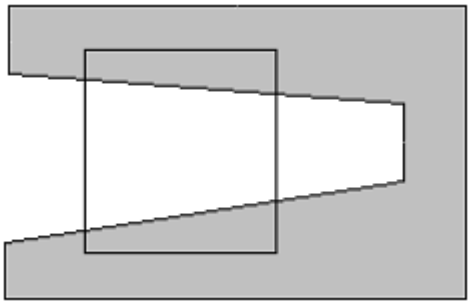
Adaptation **Several neighbors**

The **Concavity** adaptation is used when a cell is concave. The **Concavity** adaptation sequentially splits concave cells until they are eliminated.

Concave cells may cause solution divergence.

Adaptation **Concavity**

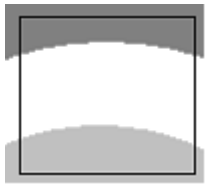
The **Identic BCs** adaptation is used for automatic gap resolution. The adaptation is performed until the distance between two edges is valid in more than one cell or until the user-defined adaptation level is reached.



Adaptation **Identic BCs**

This adaptation is applied when a cell contains two body faces with *identical* boundary conditions; this prevents performing an adaptation on cells with the Gap model because the Gap model is applied when two body faces with *different* boundary conditions are present in the *same* cell.

The **Different ConBCs** adaptation is employed to resolve the gaps between two connected boundary conditions when the cell contains two surfaces with connected boundary conditions (BCs) defined while the first surface has its BCs connected to one subregion and the second surface to another.



Adaptation **Different ConBCs**

This adaptation splits the cells until those containing different connected boundary conditions are eliminated.

Adaptation schedule setup

In addition to plain on/off controls for **Adaptations**, a schedule fine-tuning is available (specified by seconds or step-numbers). These schedule settings can are common for all active **Adaptations**.

All enabled **Adaptations** will be launched according to this schedule and every time when an **Adaptation by condition** or **Adaptation to solution** starts.

Providing adaptations in a gap

For the correct operation of the gap model, the gap cells must be of the same level of adaptation.

When the **Adapt through gap** setting is enabled, if the gap contains cells with different adaptation levels, then automatic adaptation of all cells in the the gap is applied up to the same (the maximal) adaptation level of all cells in the gap.

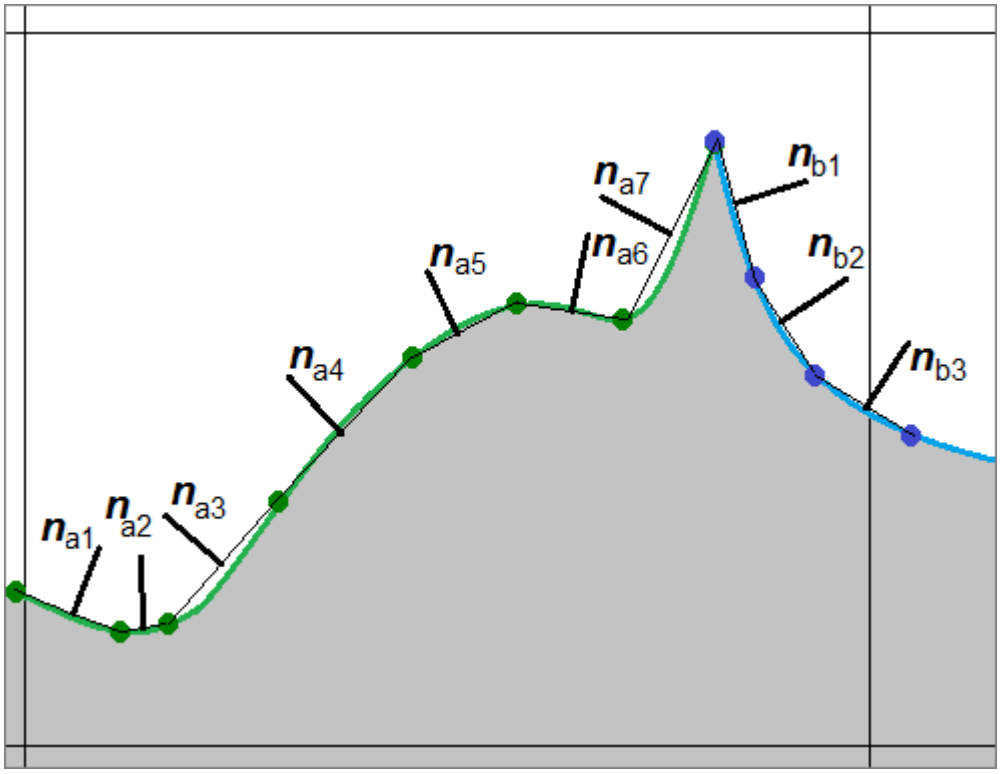
Automatic increase of the maximal level of adaptation near curved surfaces and sharp edges

For simple **Adaptations** and **Adaptations by condition** you can set automatic increase of the maximal level of adaptation near curved surfaces and sharp edges.

When angles between normals to some facets exceed the user-defined threshold values, **Adaptation to curvature** and/or **Adaptation to sharp edges** will activate.

Adaptation to curvature activates when angles between normals fall into the interval **[Max. angle, Upper limit]** for normals of any facets within a group of facets of a surface, on which the adaptation is set, within a cell.

Adaptation to sharp edges activates when angles between normals to adjacent facets of the surface, on which the adaptation is set, independently on belonging to facet groups, within a cell, exceed the specified threshold value **Sharp edge angle**.



n_{a1} ... n_{a7} and n_{b1} ... n_{b3} are normal vectors to facets belonging two facet groups

Tables below show which angles between normals to facets are compared to activate **Adaptation to curvature** and/or **Adaptation to sharp edges** on (relating to the situation shown on the illustration above).

Comparison of angles between normals to facets that can activate *Adaptation to curvature*

✓ – examine that the angles fall in the interval [Max. angle, Upper limit].

⊘ – do not examine that the angles fall in the interval [Max. angle, Upper limit].

	n_{a1}	n_{a2}	n_{a3}	n_{a4}	n_{a5}	n_{a6}	n_{a7}	n_{b1}	n_{b2}	n_{b3}
n_{a1}		✓	✓	✓	✓	✓	✓	⊘	⊘	⊘
n_{a2}	✓		✓	✓	✓	✓	✓	⊘	⊘	⊘
n_{a3}	✓	✓		✓	✓	✓	✓	⊘	⊘	⊘
n_{a4}	✓	✓	✓		✓	✓	✓	⊘	⊘	⊘
n_{a5}	✓	✓	✓	✓		✓	✓	⊘	⊘	⊘
n_{a6}	✓	✓	✓	✓	✓		✓	⊘	⊘	⊘
n_{a7}	✓	✓	✓	✓	✓	✓		⊘	⊘	⊘
n_{b1}	⊘	⊘	⊘	⊘	⊘	⊘	⊘		✓	✓
n_{b2}	⊘	⊘	⊘	⊘	⊘	⊘	⊘	✓		✓
n_{b3}	⊘	⊘	⊘	⊘	⊘	⊘	⊘	✓	✓	

Comparison of angles between normals to facets that can activate *Adaptation to sharp edges*

✓ – examine that the angles exceed the threshold value Sharp edge angle.

⊘ – do not that the angles exceed the threshold value Sharp edge angle.

	n_{a1}	n_{a2}	n_{a3}	n_{a4}	n_{a5}	n_{a6}	n_{a7}	n_{b1}	n_{b2}	n_{b3}
n_{a1}		✓	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘
n_{a2}	✓		✓	⊘	⊘	⊘	⊘	⊘	⊘	⊘
n_{a3}	⊘	✓		✓	⊘	⊘	⊘	⊘	⊘	⊘
n_{a4}	⊘	⊘	✓		✓	⊘	⊘	⊘	⊘	⊘
n_{a5}	⊘	⊘	⊘	✓		✓	⊘	⊘	⊘	⊘
n_{a6}	⊘	⊘	⊘	⊘	✓		✓	⊘	⊘	⊘
n_{a7}	⊘	⊘	⊘	⊘	⊘	✓		✓	⊘	⊘
n_{b1}	⊘	⊘	⊘	⊘	⊘	⊘	✓		✓	⊘
n_{b2}	⊘	⊘	⊘	⊘	⊘	⊘	⊘	✓		✓
n_{b3}	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘	✓	

If both **Adaptation to curvature** and **Adaptation to sharp edges** activate, their additions to the maximal level of adaptation are not applied cumulatively (the bigger addition is applied).

In the program's user interface the settings of adaptation to curvature and to sharp edges are set by parameters **Adapt. to curvature > ...** and **Adapt. to sharp edges > ...**.

See also section Folder «Adaptation».

6.14.6 Adaptation by condition

FlowVision can launch adaptations when values of the selected variables meet the specified conditions.

For each **Adaptation by condition** you can specify:

- **Boundary conditions** and/or geometry **Objects** in/on which the adaptation will act
- **Subregions**, which will limit the area where the adaptation will act
- the **Range** that turns the adaptation on, when the values of the **Variable** come into it
- numbers of adaptation layers for various levels of the adaptation
- use adaptation to curvature and to sharp edges

Description of the interface and all parameters see in the section Subfolder «Adaptation by condition».

6.14.7 Adaptation to solution

Adaptation to solution allows automatic rebuilding of the computational grid (split and/or merge cells):

- in the volume or on surfaces of the specified geometry **Objects**
- being limited by the specified **Subregions**
- acting according to the specified ***Conditions of adaptation to solution***: near the the specified value or near maximal gradient of the specified the **Variables**

In the project tree the adaptation to solution is presented by one or several elements **Adaptations to solution #N** (with their individual parameters). Also there are parameters, which are common for all elements **Adaptations to solution #N** (they are set in the properties of the folder **Adaptation to solution**).

You can specify:

- Activity of *all* **Adaptations to solution #N**. This allows you to include all the **Adaptations to solution #N** permanently or start them at the desired time or calculation step, or turn them periodically according to time or steps. All **Adaptations to solution #N** together can be turned on either permanently or since a specified time moment or step, or you can set periodic turning them on/off by time or by steps.
- Activities of *specific* **Adaptations to solution #N**. These allow the user to set individual rules for turning separate **Adaptations to solution #N** on or off.
- **Cell number**, the maximal total number of cells allocated to *all* elements **Adaptation to solution #N**. Refining of the computational grid will continue until this limit is reached; the new cells are allocated proportionally to *weighting factors* of **Subregions** (see below).

- **Subregion Weights**, which set proportions for allocation of new cells over **Subregions**.
- The maximal level of grid refinement, which is set individually for each **Adaptation to solution #N**.
- **Objects**, on which adaptations to the solution act, are set individually for each **Adaptation to solution #N**. Such **Object** can be either imported or standard geometric object or it can be the whole **Computational space** (the whole internal volume of the computational **Subregion**). One **Object** can be matched to either one or multiple **Adaptations to solution #N**.
- **Subregions**, which limit action of individual **Adaptations to solution #N**. Each **Adaptation to solution #N** is created in the specified **Subregions** and does not act beyond their limits. Therefore, if an **Object**, on which **Adaptations to solution #N** goes out of the limits of the **Subregions**, then **Adaptation to solution #N** acts only in the intersection of the **Subregions** and the **Object**.
- **Conditions** of an adaptation to solution are specified in the project tree as child elements of elements **Adaptation to solution #N** and specify either values or gradients of specified variables are applied to enable the adaptation. Also these **Conditions** include weighting factors that determine allocation of new cells between the **Conditions** (i.e. **Conditions** compete for new cells in **Subregion**, which are allocated for **Adaptation to solution #N**). List of calculated variables, which are used in the **Conditions**, is defined by the given **Model** that is set in the subregion and this list cannot be extended by the user. Allocation of new cells over **Conditions** is done according to **Weights**, which are set to the **Conditions**.

See details about parameters of the adaptation to solution in the section Subfolder «Adaptation to solution».

Adaptation to solution is applied until it spends limits for new cells in subregions; these limits are calculated according to the total **Cell number** limit and **Subregion Weights** (these limits might be not reached when are spent of its it restricts the number of cells in the subregion (this limit might be not reached if the specified maximal adaptation levels specified for specific **Adaptations to solution #N** are not big enough).

It is recommended to examine how the limitation on the number of cells in **Subregions** affects the results of the computation. The total number of cells in a **Subregion** includes all the cells in the **Subregion**, both computational and non-computational cells. The non-computational cells can be, for example, those, which are are cut out by a **Moving body** or located in a non-computational **Phase** (in vacuum).

Allocation of new cells over several **Adaptations to solution #N** is done proportionally to **Weights** that are specified in **Conditions** of these **Adaptations to solution #N**.

An **Adaptation to solution #N** acts in those cells of computational **Subregions**, which (either fully or partially) locate into the **Object**, on which the **Adaptation to solution #N** is set.

Applying an adaptation to solution in a gap

For the correct work of the gap model, the gap cells have to be of the same adaptation level. If the gap contains cells with different adaptation level, then the program will automatically apply adaptation in the entire gap with a same (the maximal) adaptation level.

See also section Subfolder «Adaptation to solution».

6.14.8 Cell's number (index)

The **Pre-Postprocessor** uses the computational cell number (index) to acquire computational parameters data.

The cell number can be displayed in two formats:

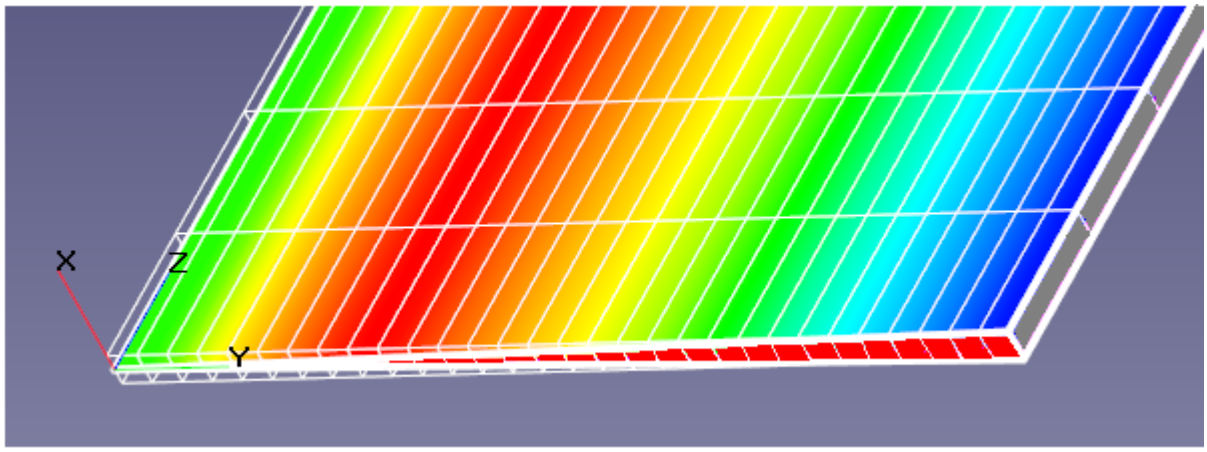
- the two-level format displays two numbers: *number of hypercell* and *number of cell*. A hypercell is a cell set created automatically by *FlowVision* to provide a convenient calculation process.
- the multilevel format displays the cell path in the cell tree indicating different levels of cells: **N0:N1: … NK**, where **N0** is the zero level cell number (in the initial grid), **N1** is the first level cell number, etc.

Sometimes the cell description in the project's log files is represented by three coordinates (**X**, **Y** and **Z**) that correspond to the cell's mass center.

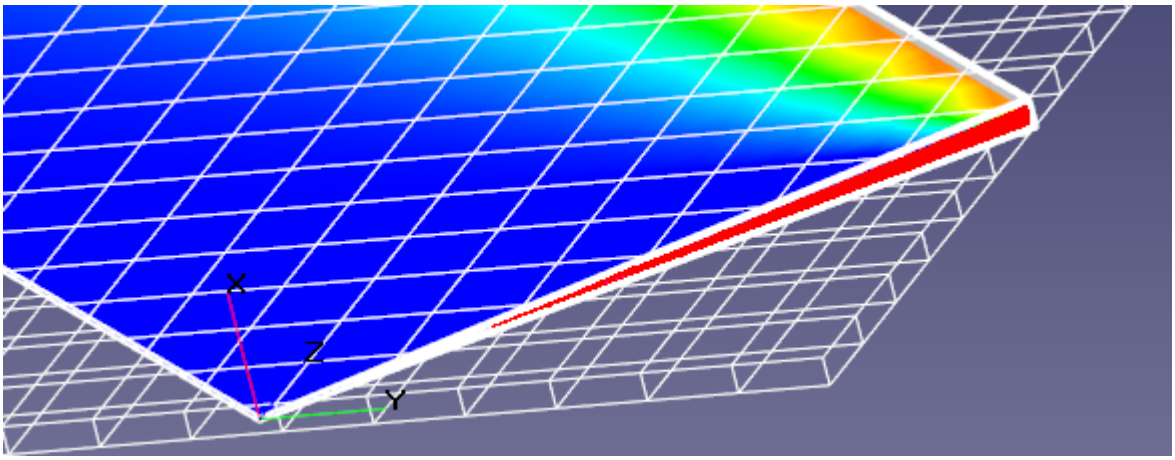
If a cell has an irregular shape, its mass center may lie outside the cell.

6.14.9 Orientation of the initial grid in axisymmetric problems

When solving axisymmetric problems, especially when they are formulates in a sector, it is recommended to position the computational domain in in such a way that its axis be parallel to the lines of the initial grid, i.e. be parallel to one of the coordinate axes of the Absolute coordinate system (ACS).



An example of correct positioning: the axis of symmetry of the computational domain is parallel to the axis OZ of the absolute coordinate system



An example of not recommended positioning: the axis of symmetry of the computational domain is not parallel to any axis of the absolute coordinate system

See also: section Specifics of solving axisymmetric problems with periodic boundary conditions.

6.14.10 Specifics of solving axisymmetric problems with periodic boundary conditions

When solving axisymmetric problems with periodic boundary conditions (for example, a sector of a turbo machine), some additional requirements are applied to the computational grid:

Size of computational cells on a periodic BC is to be nearly same to the size of corresponding sells on the other BC. That is to say that when binding surfaced it is necessary to avoid binding the cells with substantially different sizes.

When sizes of cells on the bound surfaces are substantially different, serious losses of accuracy can occur.

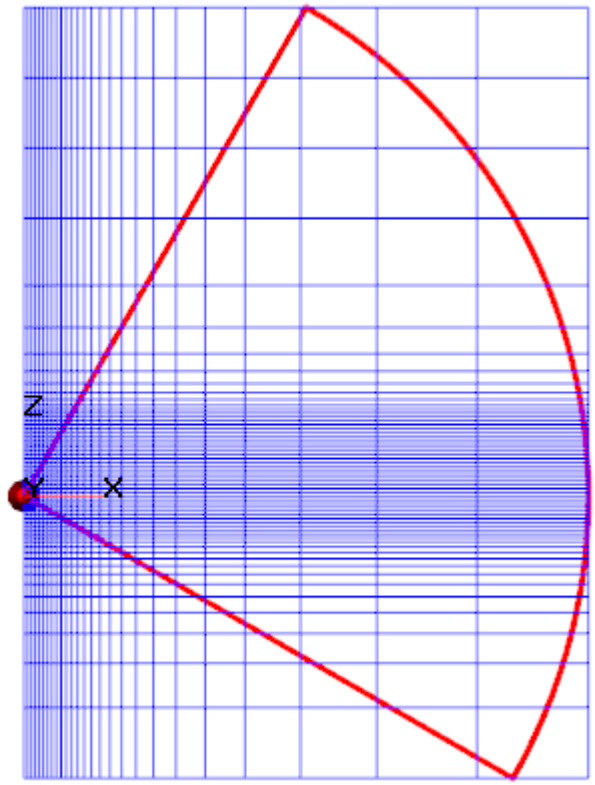
To obtain high accuracy at transferring the data between periodic BCs, use all or some of the recommendations listed below:

- When using adaptation, provide the same level of adaptation on the bound periodic BCs.
- When solving a problem with a sector of 90 degrees, position the plane periodic surfaces so they be parallel to two axes of ACS (and the axis of the sector will automatically become parallel to the third axis of the ACS).
- When solving a problem with a sector, which is not of 90 degrees, position the sector so one of the axes of ACS be coincide with the axis of the sector, and another axis of ACS be in the bisectrix of the sector. If the

computational grid is symmetric relatively to the bisectrix of the sector, then the matching cells near the periodic surfaces will be of near the same size.

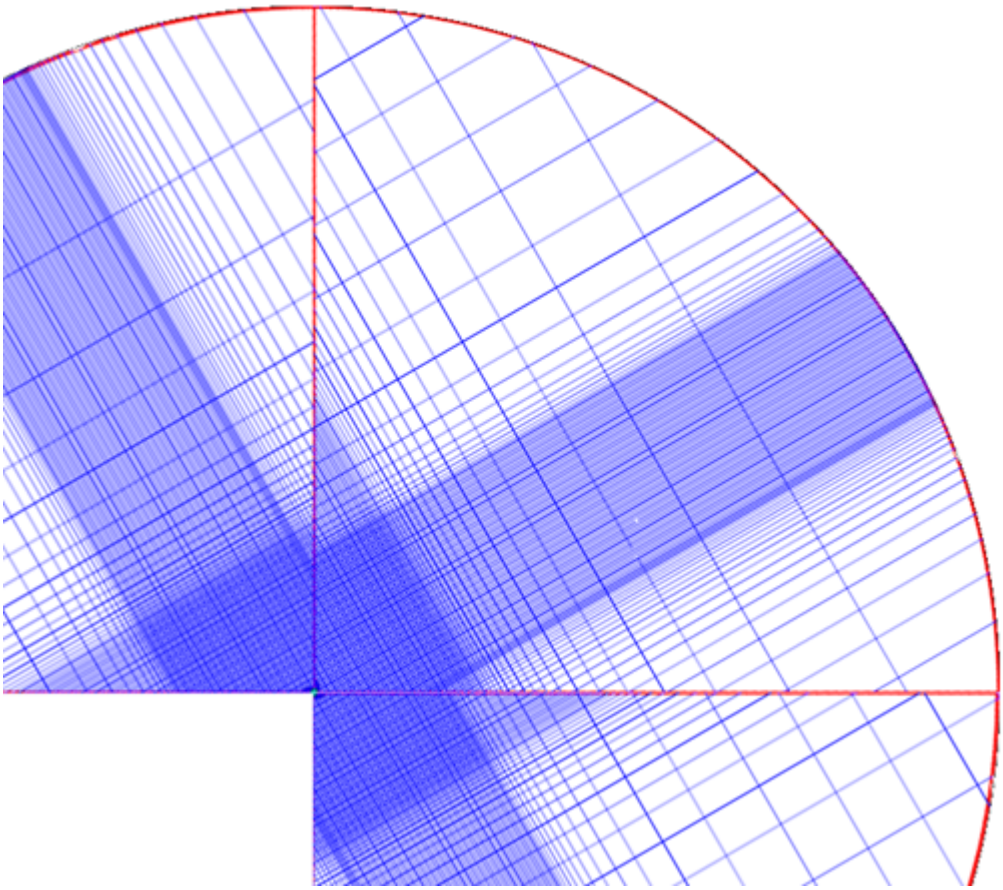
- If periodic surfaces are not flat, we recommend you to use a uniform initial grid.
- To improve the accuracy, apply adaptation on the periodic surfaces.

On the illustration below you can see a problem setting with a sector of 90 degrees and a non-uniform computational grid. In this case, the grid lines are not oriented neither along the edges of the sector nor along the bisectrix.



An example of a not recommended positioning: the sector is turned around to the axis X at 30 degrees. The sector is positioned not symmetrically relatively to the computational grid.

On the next illustration you can see as the matching cells, which are adjacent to the bound periodic surfaces, will vary greatly in size:



An example of unbalanced sizes of the matching cells, which are adjacent to the bound periodic surfaces, if the sector is positioned incorrectly

See also: section Orientation of the initial grid in axisymmetric problems.

6.14.11 Overlapping boundary layer grid (BL grid)

The *Boundary layer grid* (BL grid) is an one-dimensional adaptation, which allows solving a *boundary* layer, that provide substantial economy of computational *grid* comparing to 3-dimensional adaptation. An example of use of the BL grid is resolution of a thin viscous boundary layer near a surface of a streamlined object.

Boundary layer grid consists of almost flat cells, therefore good results are achieved when using BL grids on smooth surfaces and when flows are unseparated.

The BL grid is built on the *current main grid*, which exists at the moment of creation of the BL grid. Faces of cells of the main grid cut the surface, which is presented by connected triangles. As a result of this, polygons are formed in each cell. When a border between groups of facets overlaps on these polygons, the polygons will be split by these borders; this can increase the number of the polygons and, eventually, the number of cells of the boundary layer grid.

On these polygons the program builds averaged *effective sites*, which will be bases of prismatic stacks of thin cells of the BL grid.

If the main grid changes later, the BL grid *is not rebuilt*. This allows to save the solution, which has been computed on the BL grid. So changes of the main grid caused, for example, by moving bodies, adaptation to solution, etc., will not affect the already built BL grid.

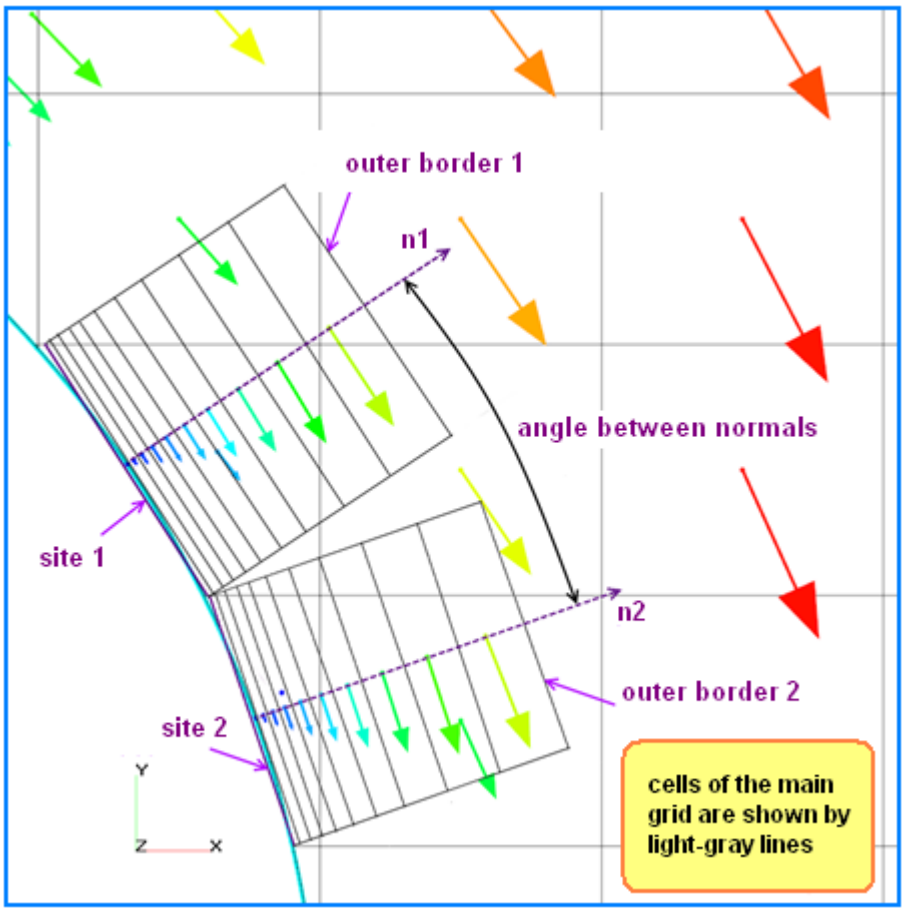
The BL grid is only rebuilt when:

- its parameters are changed
- *or* when it is enabled again after it has been disabled

A stack of cells of the BL grid would not be built when:

- center of the *outer border* of the stack does not lie in the computational domain
- *and/or* when the angle between normals of the neighboring stacks is greater then the specified maximal angle (both stacks with such angle between them would not be built)

(see the illustration below)



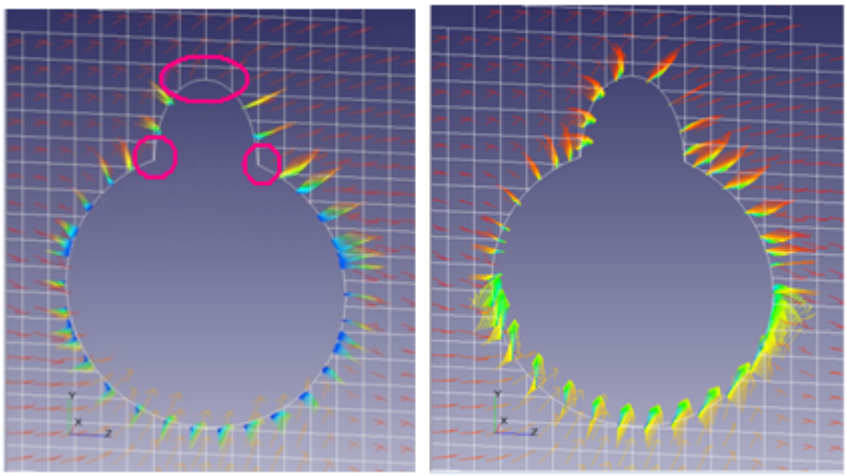
Fragments of a BL grid built on adjacent sites

BL grid on curve surfaces

For successful work of the BL grid, the surface on which it is built, should be quite smooth. You can specify the *maximal normals divergence angle*, which limits the acceptable flection of the surface. When the flection is large (and the angle between the normals is also large), the stream breaks away from a bend or an edge of the surface.

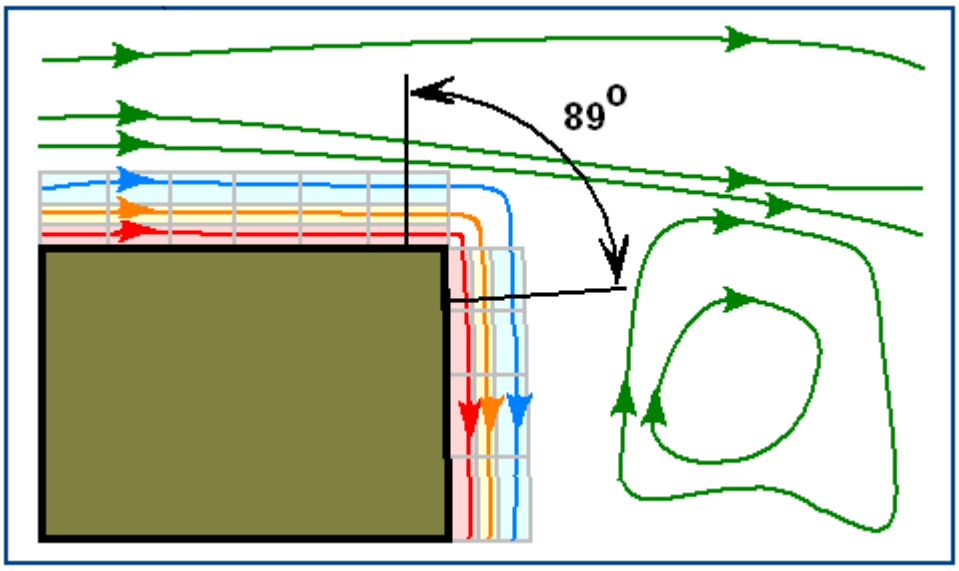
The angle between normals depends on size of cells of the main grid; when these cells are smaller, the sites under prismatic stacks also are smaller, and orientation of these sites is changing so angles between them become smaller too.

We recommend you to specify the maximal normals divergence angle in a range from 20 to 30 degrees.



When the maximal normals divergence angle is not large, the BL grid will not be built on highly bent surfaces (these areas are marked with pink contours on the left illustration). If you increase the maximal normals divergence angle, you can force the program to built the BL grid on all areas (see the illustration on the right), but this can cause a non-physical simulation.

Example of an incorrectly specified maximal normals divergence angle: If a near vertical back step has angle between normals 89º, and you, specified the *maximal normals divergence angle*, force the program to build the BL grid on the edge of the step, which assumes flowing the fluid over the step through cells of one prismatic stack to another over the edge (this is non-physical because on this edge a breakaway appears in reality), see the illustration.



If you specify the maximal normals divergence angle too large, then the solution might become non-physical (on the illustration you can see that the flow in the boundary layer after the edge moves oppositely to the real flow). We recommend you to specify this angle in a range from 20 to 30 degrees.

Algorithm of junction the solutions on two grids

One of features of the algorithm is that the volume of the BL grid is not subtracted from the volume of the main grid. Equations of continuity, momentum, energy, mass transfer, turbulence and other equations are integrated on both grids (computation is going on both grids simultaneously).

Values of desired variables are transferred from the main grid to the outer border of the BL grid (these values are interpolated from a center of the main grid's cell to which the outer border of the BL grid falls).

When the equations are solved on the BL grid, on the wall the boundary conditions, specified in the program's interface, are used. When transferring a momentum from cells of one prismatic stack to another (to appropriate cells of the neighboring stacks) the tangential component of the velocity changes its direction (rotates).

Solutions on the BL grid received on the wall (shear stress, etc.) are transferred into the main grid in which they are assumed as fixed values. If turbulence parameters in the boundary condition are specified as **Value in cell near wall**, then turbulence parameters from the BL grid are also transferred into the main grid's cell located near to the wall.

Briefly this algorithm is illustrated by the following scheme:

Main grid → Outer border of the BL grid → Computation in the BL grid → In the BL grid a solution is obtained on the wall and in the outer border's cell → Values are fixed on the wall and then used for solution in the main grid

Limitations and recommendations for use the BL grid

Using the BL grid follow these limitations and recommendations:

- The BL grid can be specified on boundary layer on boundary condition **Wall** only.
- On complex surfaces some cells near the BL grid become inactive. With a large number of edges (non-smooth surface) all or almost all of cells in the BL grid may become inactive.
- Each **Boundary condition**, on which a BL grid is built, must consist of either only one **Group** of facets or several *non-contacting* **Groups** of facets. If necessary, follow the procedure of regrouping the facets that are included into the **Boundary condition** to merge contacting **Groups** of facets if any. In the parameters of the regrouping, increase the value in the field **Threshold angle (in degrees)** and keep the **Prevent changing boundary conditions on triangles** checkbox checked, so the existing contacting **Groups** will merge and become larger (the non-contacting **Groups** still remain to be non-contacting and this doesn't prevent building a BL grid on them).
- You can not use the BL grid on **Moving bodies** on which motion is specified (you have to disable motion of the **Moving body** or else the BL grid would be rebuilt at each iteration and all values calculated on it would be cleared and recalculated on the new grid).
- In the "liquid-vacuum" problem settings, when the BL grid and the **VOF** phase transfer (see section Two phase media with an inter-phase surface) are used together, calculations and data exchange are done in only those BL grid's cells that are filled by liquid.
- The BL grid must enclose the whole boundary layer. The recommended thickness of the boundary layer on a curved surface is from 1.5 to 2 cells of the main grid. If the thickness is more, it can cause divergence between stacks of BL grid cells while angle between normals of base sites is not zero and on a some distance from the surface and the divergence exceeds one cell of the main grid (this would cause non-physical solution because data for the BL grid would be taken not from all cells but only from cells where centers of outer border BL grid cells would fall in). So we recommend you to limit the boundary layer by 1.5 to 2 cells of the main grid, because it is desirable that upper border of BL grid fall in to the next main grid's cell after the cell, which is adjacent to the surface.
- Thickness of the BL should not be less then a half of the size of the adjacent cell of the main grid. This is so because values on the outermost cells of the BL are taken from the cells of the main grid, in which centers of the outermost cells of the BL locate. When this recommendation is violated, the values of pressure and velocity would be not smooth and, therefore, values of viscous stress would also be not smooth.
- For the BL grid also the requirements are applied, as those for the main grid. For example, the distance between grid lines cannot be less then the geometric precision (the Tolerance parameter in the Basic settings of FlowVision). Thicknesses of adjacent cells in a same stack should not differ more then twice. Ratio of the length of the longest facet to the length of the shortest facet should not exceed 1000. Compliance with these rules helps to avoid non-convergence and insufficient accuracy.
- It is not recommended to use BL grid for solving axi-symmetric problems (sector geometry, 2D grid).
- You can specify several BL grids, but the program can use only one of them at each run. If in your project several BL grids are active (**Enabled=Yes** is specified in their properties), then a message will be output in the **Log** window informing you that only one (the first) of the grids will be built (`Error in Build Grid: Restriction: Only the first active BL-grid will be built. All others will be ignored.`).
- A BL grid cannot be specified on the boundary condition **Wall, ablation**.
- A BL grid does not follow changes in the main grid. It must be activated only after the adaptation of the main grid is completed.
- A BL grid is not rebuilt until user changes its settings (including **Activation**). In the course of rebuilding, all the data, stored in the old BL grid, are lost and computed afresh on the new grid.
- A BL grid is built only in the near-boundary cells whose adaptation level does not exceed the level specified in the BL grid's settings. Checkup of this condition is performed in the program only when the BL grid is activated or rebuilt.

- When an additional physical process is toggled on at a non-zero time step, new variables are introduced into the calculations. This happens, for instance, when process **Turbulence** is activated after a laminar solution is obtained or when the transition model is toggled on after a solution with use of turbulence model SST k-ω is obtained. The new variables are not initialized on the BL grid in these and similar cases. In order to continue calculations, a) activate the required physical (sub-) process, b) if necessary, specify the initial and boundary conditions for the new variables, c) make the BL grid inactive, d) perform one time step, e) make the BL grid active, f) continue the calculations.

Numbering and thickness of layers in the BL grid

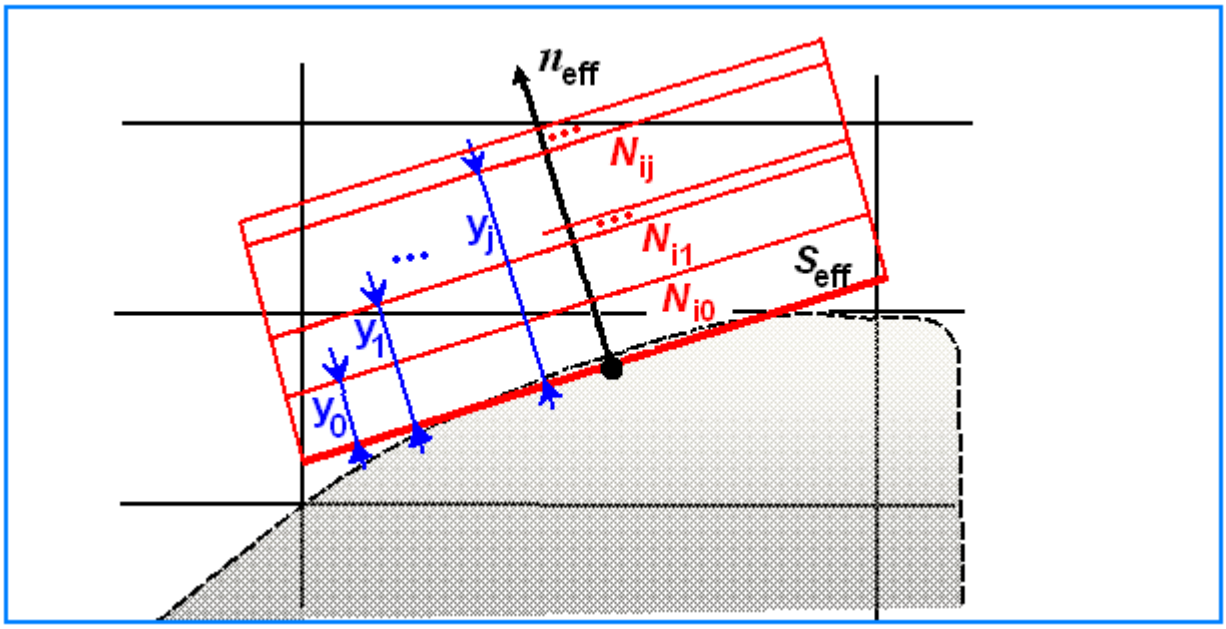
On each effective site a prismatic stack of cells is built, and layers in these stacks can have different thickness (see the illustration). Thickness of corresponding layers in all stacks are however the same for corresponding layers (read details in the note).

Layers of the BL grid are enumerated starting from the surface on which the boundary condition is defined; the layers' indexes start from zero.

Layers with identical indexes have the same thickness throughout the boundary condition, so all N_{ij} cells with equal j indexes have same thicknesses equal to:

- y_j-y_{j-1}(with $j \neq 0$)
- $y_{=0}$(with $j = 0$)

(see the illustration)



Numbering and thickness of layers in the BL grid (one effective site is shown)

Visualization of the solution in the BL grid

Solution in cells of the BL grid is only used for the following visualizations:

- **Characteristics** on a surface on which the BL grid is built.
- **Plot along curve**, on which the BL grid is built (shift is not used).
- **Vectors** in a volume of the BL grid. The **Vectors** can be colored according to values of a variable; this coloring is applied to the solution in the boundary layer's cells. You can save this colored **Vectors** layer in a `glo` file.

All other **Layers**, **Characteristics**, **Plots**, etc. display solution from the main grid only.

Specifying parameters of the BL grid

Parameters of BL grids are specified in the **Properties** windows of the elements **Computational grid > Boundary layer grids > Boundary layer grid #N** in the project tree.

See also:

Theory > Physical processes > Boundary layer grid

6.15 Calculation control parameters

See sections below:

- Parameters of the numerical method
- Limiters
- Parameters of small cells
- Validating the computational grid structure
- Turbulence model parameters
- Parameters of loadings export
- Settings for automatic saving of calculation results and visualization data
- Monitoring of project calculation's progress
- Stopping conditions for the calculation

6.15.1 Time step

Specifying the time step τ, which is common for the whole project, in done using either one of the ways:

- in the form of a dimensional constant value
- *or* in the form of a dimensionless parameter, which is used to calculate a dimensional time step (in this case, the time step changes during the computation, depending on the results of the computation).

Simulation of individual physical processes in **Phases** allows specifying for each*) of them individual **Time step coefficients** (see description of the interface in the section Folder «Phases»). Specifying a negative value for **Time step coefficients** can even stop the computation of a physical process, i.e. "freeze" it without resetting fields of appropriate physical variables.

Setting different time steps for different physical processes can sometimes speed up the convergence to the steady-state solution.

*) Except the physical process **Phase transfer** for *continuous* **Phases** (in such a case this parameter is not available for defining by a user and has a default value **1**). Use the **Solver > Advanced settings > Multiphase C > Relaxation** instead.

Dimensional constant time step

When you specify the value of the dimensional constant time step, you should be guided by the characteristic time τ_0, which is the ratio of the characteristic dimension of the computational domain to the characteristic velocity in it, i.e. it is equal to the time, during which a perturbation of the flow passes through the whole or a key part of the computational domain (the transit time).

Examples:

- For an internal flow in a pipe, the characteristic time would be the transit time.
- For an external flow around a body, the characteristic time would be the ratio of body size to the free stream velocity.
- For a closed flow, the characteristic time can be assumed as the time in which a fluid's particle encircles within the computational domain.

In *FlowVision* integration of the equations of motion and heat and mass transfer is performed using an implicit method. When a steady-state flow is calculated using the relaxation method, you should specify a time step, which is to be small relatively to the time τ_0, for example, specify $\tau=0.1\tau_0$.

Calculation of unsteady flows requires a smaller step τ, than those, which is used in calculation of a steady-state flow. In this case, you can specify the time step with the same order of magnitude as 0.1 ... 0.01 multiplied by the period or on the characteristic time of the unsteady flow.

It is recommended to examine the dependency of the solution relating to the time step, for example, decrease the time step repeatedly until this decrease causes no significant influence on the solution (but not less than the explicit time step for the implicit scheme).

Dimensionless time step (CFL numbers)

The dimensionless time step is specified by the user in the project tree in properties of the **Time step** that set appropriate CFL (Courant-Friedrichs-Lewy) numbers.

A CFL number represents the ratio of the time step to the time for which a perturbation of the flow is transferred by the stream within a cell.

CFL number	Usage
Convective CFL number, CFL_{conv}	This value is set by the **Convective CFL** parameter. It is used to calculate the convective time step τ_{conv}: $\tau_{conv} = CFL_{conv}\ \tau_{expl,conv}$, where $\tau_{expl,\ conv}$ is the explicit convective time step*), which is calculated as follows: $$\tau_{expl,conv} = 0.5 \min \frac{1}{\frac{V_x}{h_x} + \frac{V_y}{h_y} + \frac{V_z}{h_z}}$$ The minimal value is looked for over all cells in the subregion, and, for stability of the calculation, the found value is multiplied by the factor 0.5. At the beginning of the iteration all time steps (explicit, surface, diffusion) are calculated, and, based of the specified CFLs, the time step τ is calculated (see the formula and explanation after this table). Then motion of the moving bodies and free surfaces is simulated. After this explicit convective time step $\tau_{expl,\ conv}$ is *computed again* with taking into account the changed positions of the moving bodies and the free surface, but calculation τ is *not* done the second time again. The new value of the explicit convective time step $\tau_{expl,\ conv}$ is displayed in the column **Explicit step** in the **Status** tab of the **Monitor** window. *) The explicit convective time step is limited by the gravitational step (see subsection *"Gravitational time step"* below).
Diffusion CFL number, CFL_{diff}	This value is set by the **Diffusive CFL** parameter. It is used to calculate the diffusion time step τ_{diff}: $\tau_{diff} = CFL_{diff}\ \tau_{expl,diff}$ where the explicit diffusion time step $\tau_{expl,diff}$ is calculated as the ratio of the square of the cell size h_i^2 to the diffusion coefficient of the corresponding process. It is calculated as the minimum over all processes and cells in the subregion. For the **Implicit** computation scheme: $$\tau_{expl,diff} = \min_i \left(\min_{1,2,\ldots} \left(\frac{h_i^2}{D_{1i}}, \frac{h_i^2}{D_{2i}}, \ldots \right) \right)$$ For the **Explicit** scheme: $$\tau_{expl,diff} = 0.25 \min_i \left(\min_{1,2,\ldots} \left(\frac{h_i^2}{D_{1i}}, \frac{h_i^2}{D_{2i}}, \ldots \right) \right)$$ The 0.25 coefficient in the latter formula provides a steady computation when the **Explicit** scheme is used. The diffusion coefficients (D_{1i}, D_{2i}, ...) are calculated for all blocks of equations that correspond to the user-selected physical processes. Here are examples of diffusion coefficients: • μ/ρ is the kinematic-viscosity coefficient • $\lambda/(\rho C_p)$ is the temperature conductivity coefficient The diffusion CFL characterizes the ratio of the time step (τ_{diff}) to the minimum (over all blocks of diffusion equations and over all cells) time (h^2/D), in which the velocity perturbation diffusely distributes within a cell.

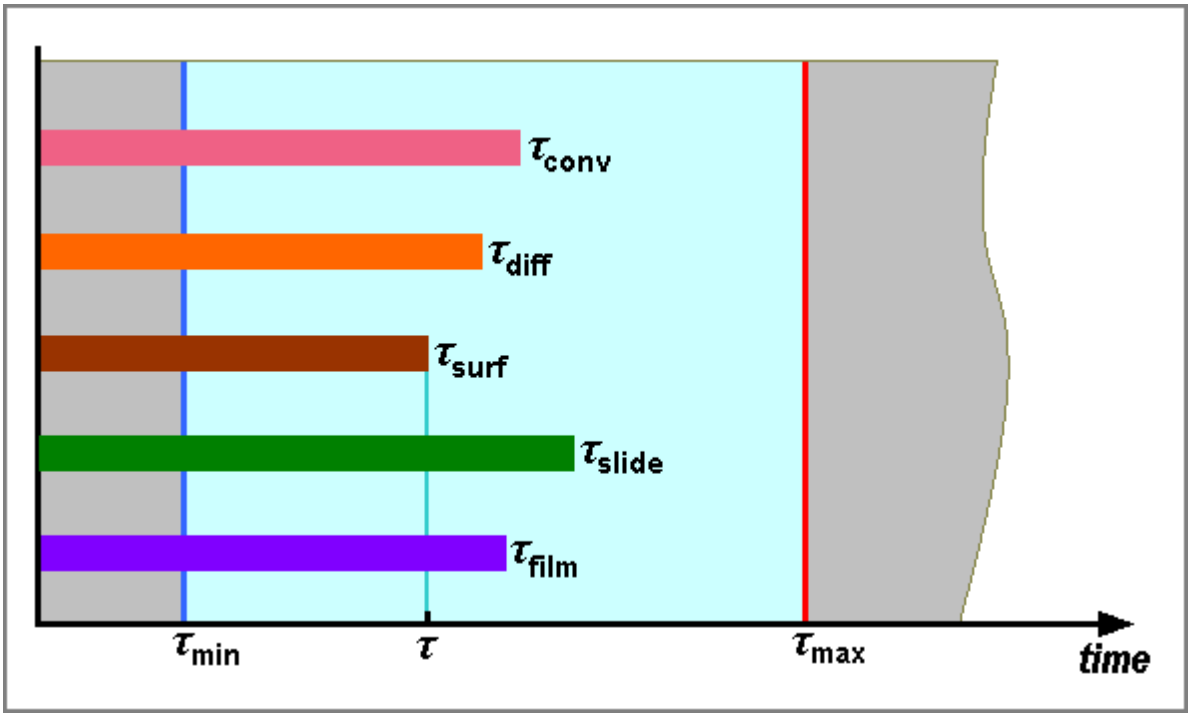
CFL number	Usage
Surface CFL number, CFL_{surf}	This value is set by the **Surface CFL** parameter. It is used to calculate the surface time step τ_{surf}. When **Multiphase C > Use VOF source for time step = No** is set in the advanced settings of **Solver**, the following formula is applied: $\tau_{surf} = CFL_{surf}\ \tau_{expl,surf}$ where $\tau_{expl,surf} = \min_j \frac{Rh_j}{V_j}$ is the explicit surface time step calculated as the minimum of the ratio of size of cells containing the interface surface or surface of a body to the absolute velocity of the interface surface or a body's surface body in this cell multiplied by the coefficient R. The coefficient R is 0.5 for a moving free surface or is 1 for surfaces of bodies. When **Multiphase C > Use VOF source for time step = Yes**, calculation of the surface time step τ_{surf} takes into account motion of the free surface caused by action of the source Q_{VOF} of the variable **VOF** in Eq. (PhTr.1). The following formula is applied in this case: $\tau_{surf} = CFL_{surf}\ \min(\tau_{expl,surf}, \tau_{expl,src})$ where $\tau_{expl,src} = \min_j \frac{R}{Q_{VOF,j}}$ is the explicit surface time step determined by Q_{VOF}. Q_{VOF} is the source of the variable **VOF** in Eq. (PhTr.1). --- The explicit surface time steps ($\tau_{expl,surf}$ and $\tau_{expl,src}$) are limited from above by the gravitational time step (see subsection *"Gravitational time step"* below). The minimum if formulae for $\tau_{expl,surf}$ is taken by all cells of the moving free surface or over surfaces of bodies. The surface CFL differs from the convective CFL in that only, that it use as a characteristic velocity the flow's velocity in near-surface cells (the velocity of a free surface), or the velocity of a body's surfaces.
Sliding CFL number, CFL_{slide}	This value is set by the **Slide CFL** parameter. It is used to calculate the time step for a sliding surface, τ_{slide}: $\tau_{slide} = CFL_{slide}\ \tau_{expl,slide}$ where $\tau_{expl,slide} = 0.5 \min_i \frac{h_i}{V_i}$ is an explicit time step for a sliding surface V_i is the linear speed on the **Sliding surface**, h_i is the cell size on the **Sliding surface** in the direction of the linear rotational speed, 0.5 is the coefficient for stability of the computation.
CFL for the dispersed phase crystallization's program block, CFL_{film}	This value is set by the **Film CFL** parameter. This is the Courant-Friedrichs-Lewy number for determining the time step for spreading the liquid film over a surface. When icing is simulated, this parameter determines the number of computational cycles that are required to obtain a quasi-stationary solution in simulating of spreading the film and its crystallization on a surface.

CFL number	Usage
	It is used to calculate the time step for the film, τ_{film}: $\tau_{film} = CFL_{film}\,\tau_{expl,film}$ where $\tau_{expl,film} = 0.5 \min_i \frac{h_i}{V_i}$ is an explicit time step for the film V_i is the linear velocity of the film, h_i is the cell size in the direction of the film's spreading, 0.5 - is the coefficient for stability of the computation. Values above **1** define the number of computational circles of the quasi-stationary process of "substance fell-out – spreading the film – crystallization of the substance" with explicit step of convective mass transfer in the film. Such values are recommended for use only for simulating icing of aircraft or other closed contours. When icing of an aircraft is simulated, **Film CFL** is recommended to be set in the range from **3** to **5**. When the film's spreading without icing is simulated, we recommend you to specify **Film CFL** as ≤**1** and also specify **Multiphase D > Film step is limited by task step = No** in advanced settings of Solver.

The time step τ is determined based on CFL numbers by following formula:

$$\tau = \min\{\max[\min(\tau_{conv}, \tau_{diff}, \tau_{surf}, \tau_{slide}, \tau_{film}), \tau_{min}], \tau_{max}\}$$

where τ_{min} and τ_{max} are the minimal and maximal allowable time steps, that is, the time step τ is calculated as the minimum of the values τ_{conv}, τ_{surf}, τ_{diff}, τ_{slide},
shifted within limits of the range [τ_{min} ... τ_{max}], if it occurs out of this range.



The time step τ is selected as the minimum value of τ_{conv}, τ_{diff}, τ_{surf}, τ_{slide}, τ_{film} and it is limited by the range [τ_{min}... τ_{max}]

Gravitational time step

Convection and surface explicit time steps $\tau_{expl,\ conv}$ and $\tau_{expl,\ surf}$ are limited from above by the gravitational time step τ_{gr}, which is calculated by the formula:

$$\tau_{gr} = \sqrt{\frac{2(|g_x|\Delta x + |g_y|\Delta y + |g_z|\Delta z)}{|\boldsymbol{g}|^2}}$$

where

$\boldsymbol{g}$ is the gravity vector (it is specified in the element **General settings** of the project tree),

g_x, g_y, g_z are components of the gravity vector along axes X, Y, Z,

Δx, Δy, Δz are minimal sizes of cells along axes X, Y, Z.

6.15.2 Parameters of the numerical method

FlowVision allows you to tune parameters of the numerical method.

Order of accuracy of the difference scheme

Convection terms in transfer equations are approximated in first- or second-order schemes.

A first-order scheme is coarser, so a second-order accuracy scheme is normally recommended for calculation. In cases when the solution is unstable, it is desirable to try to stabilize it by choosing the first order of accuracy scheme and then revert to second order of accuracy when the situation is corrected.

Selection of the first- or second-order schemes is defined by the **Numerical method > Advection scheme** parameter in the **Advanced settings** element in the project tree.

Types of the numerical method scheme

The numerical method scheme is selected by the **Numerical method > Type of scheme** parameter in the **Advanced settings** element in the project tree. Possible options are:

- **Implicit**
- **Explicit**

When **Explicit** scheme is used in calculations, the program can automatically switch to a hybrid scheme, which implies that the convection term of the convection-diffusion equation is calculated explicitly, while the diffusion term is calculated implicitly. This happens when diffusion becomes large in the solved equation, and, consequently, the explicit diffusion time step becomes small. In doing so, the integration step remains equal to the explicit convection time step.

Time integration method

As mentioned before, the integration of equations of motion and heat and mass transfer is performed in*FlowVision* using the implicit method. The time integration method is selected by the **Numerical method > Time integration > Method** parameter of the **Advanced settings** element in the project tree. The following implicit method versions are implemented:

- **Standard**: solving the equations of the mathematical model with use all terms of the equations
- **Steady-state**: solving the equations of the mathematical model with the following specifics:
  - When solving the energy equation, the term $\partial(\rho h)/\partial t$ or $\partial(\rho H)/\partial t$ is omitted if the first **Substance** in the **Phase** has aggregative state **Liquid** or **Solid**.
  - When solving the energy equation, the term $\partial p/\partial t$ is omitted.
  - The **Symmetry** boundary condition is automatically set on the free surface.
  - A **Moving body**, which moves within the computational grid, does not exert dynamic load onto the fluid (it neither pushes the medium nor pulls it).

The number of cycles of successive solution of equations of the mathematical model on one time step is specified by the parameter **Numerical method > Time integration > Number of iterations** of the **Advanced settings** element in the project tree.

Side gradient approximation of diffusion terms in equations of transfer (SGA)

Diffusion and gradient terms in equations of transfer can also be approximated using a scheme of a higher or lower order of accuracy. Normally, it is recommended to make calculations with a scheme of higher order of accuracy (use of the *side gradient approximation*, SGA). Abandoning the high order of accuracy is rational only when facing impassable non-steadiness of a solution. After stabilizing the solution process, it is advisable to return to the scheme with a high order of accuracy. Use of SGA is enabled by the**Numerical method > Use SGA** parameter of the **Advanced settings** element in the project tree.

Algebraic residual

After approximation, differential equations are transformed into a system of algebraic equations that are solved at each time step through the iterative method. A solution is considered as achieved when ratio of residual norm of the algebraic equation's solution to the norm of the the right-hand side of the equation becomes lower than the specified value.

For a system of algebraic equations $Af^{n+1} = b$, which is solved iteratively relative to variable f, at the time step t^{n+1}, the algebraic residual is equal to:

$$R_A(t^n) = \frac{\left\| b - Af^n \right\|_2}{\left\| b \right\|_2}$$

Accuracy of the equation's convergence, the maximum iteration number of the algebraic **solver is selected** in the **Pre-Postprocessor** in the **Algebraic** solver group of parameters of the **Advanced settings** element.

Besides the algebraic *residual, FlowVision* also uses *functional* residual - the maximal (within in the computational domain) speed of changing of the main calculated variable . Functional residual is displayed in the**Monitor window**and should decrease when a steady flow is calculated with the pseudo-viscosity method.

6.15.3 Limiters

Limiters (constraining parameters) define the minimal and maximal limits for calculated values (for example, pressure, temperature, density, or velocity), available during the project's computation.

If some of these values goes out of the range, it becomes equal to the value of an appropriate limiter.

These limiters prevent an uncontrollable solution's oscillation caused by local disturbances of fields of calculated values.

The **Limiters** are specified in the **Solver** tab of the project tree:

- in the properties of the element **Advanced settings** in the group of parameters **Computation of loads**
- in the folder **Limiters** in the properties of the elements:
  - **Limiters for calculation > NonPhase Limiters** (these are service settings, we do not recommend to change them)
  - **Limiters for calculation > Phase Limiters > Phase #N**

Parameters from the group **Computation of loads** limit the range of pressure, which is used to calculate hydrodynamic forces acting on a surface. These parameters influence on, for example, calculation of:

- loads transferred into a finite-element analysis (FEA) software
- motion of **Moving bodies**
- forces in **Characteristics**

6.15.4 Parameters of small cells

Surfaces of the computational domain's geometry model split the cells of the initial or adapted Cartesian computational grid into polyhedrons. Small cells can be created near borders of the computational domain's geometry model. If a cell's size is smaller than a certain threshold, it is merged with an adjacent big one (which is not small in the specified sense), so further calculations are performed in this merged big cell. This rule has an exception: if a gap model is defined, then the small cells in the gap will not merge into adjacent big cells.

FlowVision offers two criteria for detecting small cells due for merging with an adjacent one:

- the absolute criterion, according to which a cell is considered small if its volume is smaller than the specified fraction of its volume before intersecting the surface of the computational domain

- the relative criterion, according to which a cell is considered small if its volume is smaller than the specified fraction of the biggest adjacent cell value. The relative criterion is recommended for calculations in a narrow sector, for example, when performing axisymmetric flow simulation.

Moreover, the **Dynamic criterion** is used to rebuild the grid cells on-the-fly and considers a cell to be small if its volume ratio at the current and the previous time step is smaller than the specified fraction.

See also:

In the program's user interface parameters of small cells are set in the **Solver** tab of the project tree in properties of elements **Limiters > Limiters for calculation > Phase Limiters > Phase #N**, in the group of parameters **Small Cells**.

6.15.5 Validating the computational grid structure

FlowVision has a computational grid structure validation procedure. This validity check may require for a noticeable amount of calculation time, for example, in a project with a complex computational grid that has to be frequently rebuilt due to moving bodies, which locate in the computational domain. That is why this check is disabled by default. It is possible to enable computational grid structure validation through the user interface.

Computational grid structure validation errors are written to the project `*.err` log. The error entry contains coordinates or cell indexes pointing to errors found.

Performing project calculation using a computational grid with errors yields erroneous results.

The most common computational grid structure defects are:

- self-intersections
- too small facets
- wrong accuracy of import of geometry model of computational domain (it is set by the **Geometry import > Tolerance** parameter in the **Preferences** dialog box, which is opened by the **File > Preferences** command in the main menu).

Computational grid validation is performed each time it is modified, so this may significantly slow down the calculation progress.

Validation of the computational grid's structure is toggled by the **Check grid** parameter in **Advanced settings** of **Solver**.

6.15.6 Turbulence model parameters

In FlowVision, flow turbulence is taken into considerations by introducing additional terms to the Navier-Stokes equations that describe the turbulent viscosity and turbulent thermal conductivity.

Turbulent viscosity μ_t and turbulent thermal conductivity coefficients μ_t/Pr_t are calculated by solving additional equations that are defined by the semi-empirical turbulence models (see section Theory > Physical processes > Turbulence).

Calculation control parameters include the following:

- distance from the cell center to the nearest wall calculation method
- near-wall functions used to calculate parameters in near-wall cells
- limitations to prevent a non-steady process of flow turbulence calculation:
  - on the turbulent viscosity value
  - on the minimum dimensionless distance from the cell center to the wall
  - on dimensionless roughness

These parameters are set in **Pre-Postprocessor** in the project tree on the **Solver** tab in the **Turbulence** group of parameters of the **Advanced settings** element.

6.15.7 Parameters of loadings' export

FlowVision allows you to export loadings into a text file for their later analysis and use. This file contains:

- Mechanical loadings (in nodes / on faces)
- Thermal loadings (in nodes / on faces)
- Coordinates of nodes or identifiers of nodes that are vertices of faces.

Example: unidirectional stationary calculation with a finite-element software. *FlowVision* provides the calculation and exports loadings from a body. A file with loadings can be imported by external programs such as *Abaqus*, *APM WinMachine*, *Nastran*, *Ansys*, etc.

External programs might require different formats of input data. Data export to an external program might require converting the file.

Export of loadings is possible on:

- an **Imported object**, on which a **Moving body** modifier is set
- a **Subregion**

If loadings are exported from a **Subregion**, it is important that equations for motion and/or heat transfer were enabled in the **Model** of this **Subregion**. Otherwise this **Subregion** cannot be used for export of loadings.

Parameters of export of loadings are set in the properties of the element **Advanced settings** in the **Solver** tab of the project tree, in the group of parameters **Export loadings**.

The **Export subject** parameter allows you to tune export of loadings from *FlowVision* either from nodes or from faces of the geometry model.

When export of loadings from nodes is selected (**Export subject = Nodal loadings**), *FlowVision* interpolates loadings from elements to nodes, see details in the section Node loadings interpolation.

When export of loadings from faces is selected (**Export subject = Facial loadings**), *FlowVision* obtains loading on a face by integrating from the cells that are adjacent to the face.

Other parameters of export of loadings define the type and timing of recordings to the file, the file name and the list of **Subregions** and/or **Imported objects**, loadings from which are saved in the file.

Please note, that *FlowVision* is also able to do conjugate calculation of a flow near a body, which is being deformed, and the body's deformation caused by the loadings. This process is organized by alternating calculations of *FlowVision* and an external program (for example, *Abaqus*) and exchanging calculations' results after each step.

See also section Folder «External Connections».

FlowVision has also functionality for visualization and export of loadings on the surface of an **Imported object** in nodes of the computational grid (which has been built in *FlowVision*). For this use the **Nodal loadings** visualization layer.

See sections:

- Layer «Nodal loadings»
- Layer «Nodal loadings», user interface

Format of the loadings export file

See below examples of loadings export files.

With each record for each object, on which the loads are calculated, a set of data is recorded into the file.

At the beginning the following information is recorded:

XX.YY.ZZ AA:BB:CC	Date and time of recording into the file
NODE LOADINGS FOR "*Object name*" ***or*** **FACE LOADINGS FOR "*Object name*"**	The name of the object, on which the loadings are calculated. When loadings *on nodes* are exported, the object's name is preceded with the text "**NODE LOADINGS FOR**". When loadings *on faces* are exported, with the text "**FACE LOADINGS FOR**".
Step	Number of the time step
Time	Time

When the program exports loadings *on facets*, then information about grid nodes, which are corners of faces (corners of triangles). Each node receives its own number and its coordinates are recorded:

NodeNum	Number of a node
Node.X	Coordinates of the node
Node.Y	
Node.Z	

Then lines corresponding to nodes or faces are written, containing the following information.

When loadings on nodes are exported:

PartName(optional)	Name of a part in the finite-element geometry assembly (in the naming system of the program, into which the loadings export is done, for example, *Abaqus*). This column is optional.
NodeNumber	Number of a node in the part.
Area	Area of the node
Node.X	Coordinates of the node
Node.Y	
Node.Z	
Normal.X	Components of the normal vector on the node
Normal.Y	
Normal.Z	
Pressure	Value of pressure on the node
Force.X	Projections of forces on coordinate axes
Force.Y	
Force.Z	
Temperature	Value of temperature on the node
HeatFlux	Heat flux on the node
Formats for recording for objects from *Abaqus* and objects from other sources differ: • For objects from *Abaqus*, the numbering of nodes starts from "1", while, for objects from other sources, the numbering starts from "0". • For objects from *Abaqus*, the separators between columns are spaces, while, for objects from other sources, the separators at tabs. • For objects from *Abaqus*, the column **PartName(optional)** can be presented.	

When loadings on faces are exported:

Triangle Number	Number of a face (triangle) in the detail
Area	Area of the face
NodeID 1	Identifiers of the nodes, which are vertices of the face
NodeID 2	
NodeID 3	
Pressure	Pressure on the face
Force.X	Projections of forces on coordinate axes
Force.Y	
Force.Z	
Temperature	Value of temperature on the face
HeatFlux	Heat flux on the face

See details about calculating the loads in the section Node loadings interpolation. For those nodes, which are not solved by the grid, zero values are written.

Examples of the loadings export files

Some examples of fragments of loadings export files are given below. It is convenient to use *Microsoft Excel* or similar software to view such files or their fragments.

An example of a file for exporting the loads *on nodes* (if **Export loadings > Export subject = Nodal loadings** is set in **Advanced settings** of **Solver**):

==========

```
"01.08.16 17:00:56 NODE LOADINGS FOR ""Imported object #0"""
Step=3
Time=3
PartName(optional) NodeNumber Area Node.X Node.Y Node.Z Normal.X Normal.Y Normal.Z Pressure Force.X Force.Y Force.Z Temperature HeatFlux
================================================================================
CUBE-1 1 0.1875 -0.25 -0.25 0.5          -0.57735 -0.57735  0.57735 1000      62.5    62.5   62.5 0.00E+00 0
CUBE-1 2 0.1875 -0.25  0.25 0.5          -0.57735  0.57735  0.57735 1000      62.5   -62.5  -62.5 0.00E+00 0
CUBE-1 3 0.1875 -0.25 -0.25 2.60E-08 -0.57735 -0.57735 -0.57735 1333.33   62.5     62.5  125  4.16667    0
CUBE-1 4 0.1875 -0.25  0.25 2.65E-08 -0.57735  0.57735 -0.57735 1666.67   62.5   -125    125  8.33333    0
CUBE-1 5 0.1875  0.25 -0.25 0.5           0.57735 -0.57735  0.57735 1333.33 -125     62.5  -62.5 4.16667    0
CUBE-1 6 0.1875  0.25  0.25 0.5           0.57735  0.57735  0.57735 1666.67 -125    -125   -62.5 8.33333    0
CUBE-1 7 0.1875  0.25 -0.25 2.80E-08  0.57735 -0.57735 -0.57735 1666.67 -125     62.5   125  8.33334    0
CUBE-1 8 0.1875  0.25  0.25 2.85E-08  0.57735  0.57735 -0.57735  2333.33 -125 -187.5    125 16.6667     0
...
```

An example of a file for exporting the loads *on faces* (if **Export loadings > Export subject = Facial loadings** is set in **Advanced settings** of **Solver**):

```
================================================================================
02.08.16 12:40:32 FACE LOADINGS FOR "SubRegion #0"
Step=3
Time=3
Geometry data
NodeNumber      Node.X    Node.Y    Node.Z
================================================================================
0         -1        -1        -1
1         -1        -1        1
2         -1        1         -1
3         -1        1         1
4         1         -1        -1
5         1         -1        1
6         1         1         -1
7         1         1         1
================================================================================

Face loadings data
Triangle Number   Area      NodeID 1  NodeID 2  NodeID 3  Pressure  Force.X   Force.Y   Force.Z   Temperature       HeatFlux
================================================================================
0         2         2         0         3         2000      -4000     0         0         12.5      0
1         2         3         0         1         2000      -4000     0         0         12.5      0
2         2         0         4         1         1000      0         -2000     0         0         0
3         2         1         4         5         1000      0         -2000     0         0         0
4         2         4         6         5         1000      2000      0         0         0         0
5         2         5         6         7         1000      2000      0         0         0         0
6         2         6         2         7         1000      0         2000      0         0         0
7         2         7         2         3         1000      0         2000      0         0         0
8         2         1         5         3         1000      0         0         2000      0         0
9         2         3         5         7         1000      0         0         2000      0         0
10        2         2         6         0         1000      0         0         -2000     0         0
11        2         0         6         4         1000      0         0         -2000     0         0

================================================================================
02.08.16 12:40:32 FACE LOADINGS FOR "Imported object #0"
Step=3
Time=3
Geometry data
NodeNumber      Node.X    Node.Y    Node.Z
================================================================================
0         0.25      0.75      2.50E-08
1         0.25      0.25      2.55E-08
2         0.25      0.75      -0.5
3         0.25      0.25      -0.5
4         0.75      0.75      2.70E-08
5         0.75      0.25      2.75E-08
6         0.75      0.75      -0.5
7         0.75      0.25      -0.5
================================================================================

Face loadings data
Triangle Number   Area      NodeID 1  NodeID 2  NodeID 3  Pressure  Force.X   Force.Y   Force.Z   Temperature       HeatFlux
================================================================================
0         0.125     2         3         0         1000      125       0.00      0.00      0         0
```

1	0.125	3	1	0	3000	375	0.00	0.00	25	0
2	0.125	0	1	4	2000	0.00	0.00	-250	12.5	0
3	0.125	1	5	4	2000	0.00	0.00	-250	12.5	0
4	0.125	2	0	6	1000	0.00	-125	0.00	0	0
5	0.125	0	4	6	1000	0.00	-125	0.00	0	0
6	0.125	1	3	5	3000	0.00	375	0.00	25	0
7	0.125	3	7	5	1000	0.00	125	0.00	0	0
8	0.125	3	2	7	1000	0.00	0.00	125	0	0
9	0.125	2	6	7	1000	0.00	0.00	125	0	0
10	0.125	5	7	4	1000	-125	0.00	0.00	0	0
11	0.125	7	6	4	1000	-125	0.00	0.00	0	0

6.15.8 Settings for automatic saving of calulation results and visualization data

Results acquired from the calculation process can be visualized and analyzed.

Data visualization and analysis is performed:

- in **Pre-Postprocessor**
- in **Viewer**

FlowVision has two calculation results and data analysis and visualization formats:

- analysis and visualization of data acquired *at the last time step* is available to the user if the project is loaded on the **Solver**
- analysis and visualization of data acquired *at several time steps* is available to the user if the *calculation results or visualization data (layers) save mode* was defined before the calculation start.

Saving of *computational data* requires a significant amount of system resources but enables to get more information than saved layers provide, which are set up before the calculation start. When it is unclear which layers would provide exhaustive information about the flow simulation, it is convenient to save complete calculation results.

You can set the computational and visualization data saving frequency (see sections Data autosave and Layers autosave).

The calculation results and layer visualization data can be accessed dynamically and used to create animation with frames corresponding to specific time steps.

Computational data are saved to files with the following extensions:

- `fvmind`
- `fvgrid`
- `fvdata`
- `fvstat`

Visualization data (*layers*) are saved to a file with extension `fvvis`.

Files are saved to the server part of the project from where they can be accessed by **Pre-Postprocessor** if **Pre-Postprocessor** is connected to the **Solver** and the project is loaded on the **Solver**.

If **Pre-Postprocessor** is not connected to the **Solver**, then, if you want to view the saved layer visualization data, `fvvis` files have to be copied from the server part of the project to the client part.

Available operations with saved results in Pre-Postprocessor:

Operation	Data	Layers
Starting the computation from a saved step	Yes	No
Visualization with connection to **Solver**	Yes	No
Visualization without connection to **Solver**	No	Yes
Visualization of **Layers** created before the calculation		
Parameters of **Layers** did not change after saving	Yes	Yes
Parameters, which affect on building the **Layer**, changed	Yes	No[1)]
Parameters, which affect on displaying the **Layer**, changed	Yes	Yes
Visualization of layers, which were created after the calculation was done	Yes	No
Automatic sequential loading results in connection with **Solver**	Yes	No

Note:

[1] The visualization is performed using parameters, which were set before the saving.

Available operations with saved results in Viewer:

Operation	Data	Layers
Visualization with connection to **Solver**	Yes	No
Visualization without connection to **Solver**	No	Yes
Visualization of layers, which were created after calculation	Yes	No
Automatic sequential loading of the results without connection to **Solver**	No	Yes

6.15.9 Monitoring of project calculation's progress

Project calculation progress control is performed using a variety of tools.

General process characteristics are displayed in the **Monitor** window, there are also tools to define characteristics that can be used to analyze the computational data with the project solution in progress.

Functional residual

Functional residual $R_{norm}(t^n)$ at the t^n time step is the maximum f variable rate of change in the computational domain:

$$R_{norm}(t^n) = \frac{R(t^n)}{\max\limits_{1<k<n} R(t^k)}, \qquad R(t^n) = \sqrt{\frac{1}{V}\int_V \left(\frac{f^n - f^{n-1}}{\tau}\right) dV},$$

where V is the computational domain's volume and τ is the time step.

The functional residual value in steady flow calculation must decrease over time. If the residual value is stably high, the calculated process is non-steady.

See also section Project calculation progress control (operations).

6.15.10 Stopping conditions for the calculation

Project solution can be stopped:

- manually by clicking the **(Stop computation)** button in the toolbar **Network**
- *or* automatically if one of the stopping conditions (specified by the user) is fulfilled

The following stopping conditions can be defined:

- by expiration of the computational period (time of integration of non-steady equations of flow movement and heat transfer)
- after a specific number of time steps
- by the value of functional residuals of calculation variables
- by stopping criteria defined for convergence of user variables or characteristics

In some cases, it is practical to use a mean value of the target function used for stopping the computation, and use not the target function $f(t)$ itself, but its *sliding average* by some averaging period T. The sliding average is derived from $f(t)$ via the following formula:

$$\bar{f}(t) = \frac{1}{T}\int_{t-T}^{t} f(\tau)d\tau$$

The time dependency of the functional residual, variable, or characteristic used as the stopping condition is displayed as a plot in the **Plot** tab of the **Monitor** window.

It is convenient for plotting a specific value's dependency on time (don't specify the actual stop of the calculation, specify a zero value for the **Level** parameter of an appropriate element in the project tree corresponding to a residual, user variable or characteristic see subsections *Subfolder «Residuals» and elements in it* and *Subfolder "User values" and elements "Stop criterion #N"* in the section Folder «Stopping conditions»).

See also: Description of user interface for stopping the calculation see m section Folder «Stopping conditions».

6.16 Layers and displaying the computation's results

Specifics of data visualization in *FlowVision*

A unique specific of *FlowVision* is the ability to visualize data not only after finishing the computation, but also during computation.

A user can pause the computation to analyze intermediate results. Also it is possible to receive the visualization data without stopping the computation.

Important note: The visualization data are prepared by **Solver**. This procedure of preparing the visualization data is resource consuming and requires some processor's time. Each time when **Characteristics** or **Layers** are saved in files (see details in sections The "Characteristics" folder and General properties of Layers), and when transferring the date to a client module, **Solver** carries out the procedure of preparing the data. So the permanent connection of a client module to **Solver** and also the excessive frequency of saving the data in files could substantially slow down the computation.

Recommendation: If it is not required to view the course of the computation at each time step, we recommend disconnect client modules some time after the start of the computation. To do so:

- in **Pre-Postprocessor** click the (**Disconnect from the solver**) icon in the **Network** toolbar or terminate the work of this module
- in **Viewer** terminate the work of this module

Forms of displaying the computation's results

Project computation's results can be displayed in the following forms:

- as a diagram of a computational variable's distribution in the computational domain (in the volume, on a surface or along a line)
- as quantitative integral characteristics of the flow and the heat and mass transfer
- as a time dependency of those variables and characteristics, which are specified as stopping criteria for the computation
- as values of variables in the center of a computational cell

Distributions of variables in the computational domain are represented by the **Layer** elements with computational data.

The quantitative integral characteristics of the simulated flow are displayed in the **Info** window as a list of components of the **Characteristic** element.

The time dependence of the variable or characteristic that is set as the stopping condition is displayed as a plot on the **Plot** tab of the **Monitor** window.

Conditions for visualization

Conditions, under which visualization is possible, might depend on the manner how the saving of the results has been done (as data or as **Layers**).

If the data have been saved, you can display the obtained results by the following steps:

- open the project in **Pre-Postprocessor**
- connect to **Solver**
- if required, synchronize the client and the server parts of the project.

After that, **Pre-Postprocessor** will display the last saved data. If necessary, you can create additional elements in **Postprocessor**.

If the data were saved with the history, you can view intermediate results (using the **Navigation** toolbar) and create animations (using the **Capture** toolbar) with displaying the time dependency of the simulation.

Selection of visualization elements depending on the type of the displayed data	
Data type	**Name of the element**
Integral values:	
A variable's value on a surface and in a volume	**Characteristics**
Local values:	

Selection of visualization elements depending on the type of the displayed data	
Data type	**Name of the element**
Local values of a scalar variable along a line	**Layer «Plot along line»** **Layer «Plot along curve»** **Layer «Plot along ellipse»**
Local values of a scalar variable on a surface	**Layer «Color contours»**
Local values of a scalar variable in a volume	**Isosurface** **VOF** (Only for variable VoF, see section Theory > Physical processes > Phase transfer)
Local values and directions of a vector variable on a surface and in a volume	**Layer «Vectors»** **Layer «Streamlines» and element «Emitter for Streamlines»**

Displaying various values				
The value of	**Object**	**Characteristics / Layer**	**Variable in the Properties window**	**Variable in the Info window**
Integral values:				
The flow through the input or output	**Supergroup** created on the surface of the input or output	**Characteristics** on the corresponding **Supergroup**	any from the list	Mass flow + Mass flow- Volumetric Flow + Volumetric flow-
The average speed in the section	**Plane** section	**Characteristics** on the corresponding **Plane**	**Velocity**	<f surf.>
Average temperature section	**Plane** section	**Characteristics** on the corresponding **Plane**	**Temperature**	<f surf.>
The average temperature of the flow through the section	**Plane** section	**Characteristics** on the corresponding **Plane**	**Temperature**	<f mass+> <f mass->
The average surface temperature	**Supergroup** created on a given surface	**Characteristics** on the corresponding Supergroup	**Temperature**	<f surf.>
Force or torque acting on a streamlined body	**Supergroup** created on the surface of the body	**Characteristics** on the corresponding Supergroup	any from the list	- F liq. X, Y, Z1) - M liq. X, Y, Z1,2)
Position of the center **Moving body**	**Imported object**, on which a **Moving body** is created	**Characteristics** on the corresponding **Imported object**	any from the list	The center of rotation X, Y, Z
Velocity of movement **Moving body**	**Imported object**, on which a **Moving body** is created	**Characteristics** on the corresponding **Imported object**	any from the list	Velocity X, Y, Z
Local values:				
Plot of the pressure distribution along the contour of the cross section	**Plane** of a section	**Plot along curve** on the corresponding **Plane**[3)]	**Pressure**	
The pressure distribution in the cross section	**Plane** of a section	**Color contours** on the corresponding **Plane**	**Pressure**	No

Displaying various values				
The value of	**Object**	**Characteristics / Layer**	**Variable in the Properties window**	**Variable in the Info window**
The velocity distribution in the cross section	**Plane** of a section	**Vectors** on the corresponding **Plane**	**Velocity**	No
The pressure distribution on the body surface	**Supergroup** created on the surface of the body	**Color contours** on the corresponding **Supergroup**	**Pressure**	No
Streamlines	**Computational space**	**Streamlines** in the **Computational space**[4)]	**Velocity**	No
Surface phase (only in problems with the contact surfaces)	**Computational space**	**VOF**	No	No
Pressure in a given volume (sensor)	**Object** (**Box** or **Ellipsoid/sphere** or **Cone/cylinder**) corresponding sensor	**Characteristics** on an appropriate **Object**	**Pressure**	<f vol.>

Notes:

[1)] These are F_{liq}. and T_{liq}, respectively, the force and the torque, with which the body acts on the flow. The force and the torque, with which the flow acts on the body, are equal in magnitude and opposite in sign to F_{liq}. and T_{liq}.

[2)] The torque is defined relatively to the **Center**, which coordinates are specified in the **Characteristics**.

[3)] When you create a **Plot along curve** it is required that the curve be located in the first quarter (quadrant) of the coordinate system of the plot.

[4)] You have to specify an **Emitter** for **Streamlines**. Any **Object** can be uses as an **Emitter**.

Creation of a **Layer** or changing its settings can be done at any time.

6.16.1 Layers

Visualization of calculation results is performed by **Layers**.

Layers compose the image in the **View** window of the **Pre-Postprocessor**.

Layers, like other entities, are located in the project tree (the **Layer** element).

The image in the **View** window is composed of **Layers** that are displayed in a certain order. Each **Layer**'s image can be set up independently from other **Layers**, and the resulting image can be composed from **Layers** with toggling visibility.

All **Layers** are displayed over a *background* that is displayed first.

Layers can display:

- *initial data* available before the calculation start (geometry model body images, the initial grid, the absolute coordinate system icon)
- *computational data* compiled in the calculation process

Every **Layer** is composed based of certain information. This information, which represents the **Layer** element of the project tree, is composed of processed project data (initial data or calculation results). For example:

- a layer displaying the geometry model of the computational domain (layer **Solids**) contains facet node coordinates from the surfaces of the computational region and moving bodies.
- a layer with flow velocity vectors (the layer **Vectors**) contains node coordinates with calculated velocity and velocity vector components in these nodes.

The **Layer**'s image can be customized with *display parameters* (for example, specifying the palette that collates the colors with the displayed variable values).

The **Layer**'s display mode is user-defined. A **Layer** can be displayed constantly or only when it is selected it in the project tree.

Layer types

The layer type defines the display format for the layer data. The following layer types are available:

- **Coordinate system** (always presented in the project tree and cannot be removed)
- **Solids** (automatically added to the project tree when a geometry model is loaded and cannot be removed)
- **Initial grid** (always presented in the project tree and cannot be removed)
- **Vectors**
- **Plot along line**
- **Plot along curve**
- **Plot along ellipse**
- **Isosurface**
- **Streamlines**
- **Nodal loadings**
- **VOF** (visualization of a free surface)
- **Distributed characteristics**
- **Color contours**
- **Cell set**
- **Computational grid**
- **Computational grid section**
- **Cell debug**
- **Volume visualization**
- **Mapping surface**

The *layer with computational data* is defined on a certain geometric object on which or in which the image is composed. These objects include:

- the computational space (for example, when building the initial grid or a phase interphase boundary)
- a line (for example, when plotting certain parameter, which changes on a tube axis)
- a surface of a standard or imported geometric object (for example, when calculating pressure or heat distribution)
- a volume of a standard or imported object with a finite volume (for example, when calculating velocity vectors or heat isosurfaces)

Layers with computational data (all layers except **Solids**, **Initial grid** and **Coordinate system**) have a similar structure.

Most **Layers** with computational data display the values of some variable, but some layer types display information about certain cells of the computational domain instead:

- **Cell set**
- **Computational grid**
- **Computational grid section**
- **Cell debug**

The main parameters of these layers are:

- an object defining the spacial orientation of the layer
- the variable to be displayed
- other parameters of the layer's appearance

The information necessary to create **Layers** with computational data is acquired only after the **Solver** has started calculations.

Displaying of the Layers

Layer elements with computational data both display their images in the **View** window and provide quantitative data in their Info windows.

The layer image in the **View** window is:

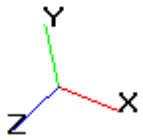
- a visualization of the variable distribution calculated during project solution (for the computation variable information layer)
- an image of the computational grid or a cell (for layers with data in cells of the computational grid)

The **Info** window displays:

- visualization of the variable distribution during project solution (for the computation variable information layer)
- characteristics of the computational grid or a cell (for the computational grid cells information layer)

6.16.1.1 Layer «Coordinate system»

The **Coordinate system** layer represents an icon image of the three coordinate axes **X**, **Y** and **Z** absolute coordinate system (ACS).



The image of the three coordinate axes X, Y and Z of the absolute coordinate system (ASC)

The coordinate axes are shown in different colors:

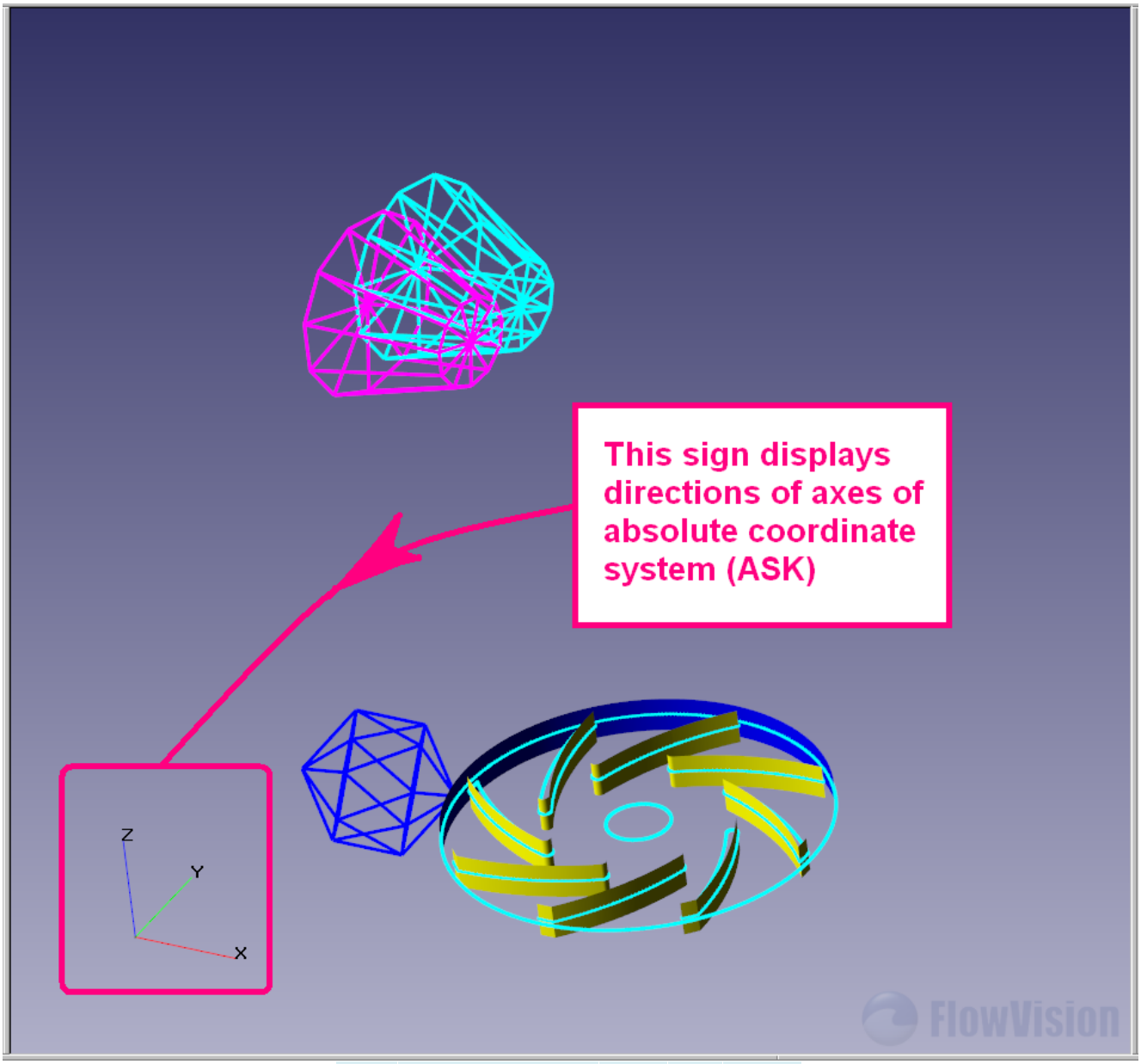
- the axis **X** is red
- the axis **Y** is green
- the axis **Z** is blue

and they are marked with letters **X, Y** and **Z**.

The icon of the coordinate axes can be configured:

- place it in the desired angle of the **View** window
- place it in the beginning absolute coordinate system (ACS)
- select size

Coordinate system layer displayed on top of other layers.



Layer **Coordinate system** in the **View** window

See also: Layer «Coordinate system», user interface.

6.16.1.2 Layer «Solids»

The **Solids** layers are used to display the surfaces that make up the computational domain and moving bodies.

Every component of the computational domain's geometry model corresponds to an element in the project tree.

Visualization parameters can be set individually for each component of the computational domain in the **Properties** window, for example you can specify the following settings:

- toggle component visibility
- toggle component clipping
- toggle component lighting
- specify color and width of outlines of groups of facets
- use of the inner surface of bodies
- setting for visual semi-transparency

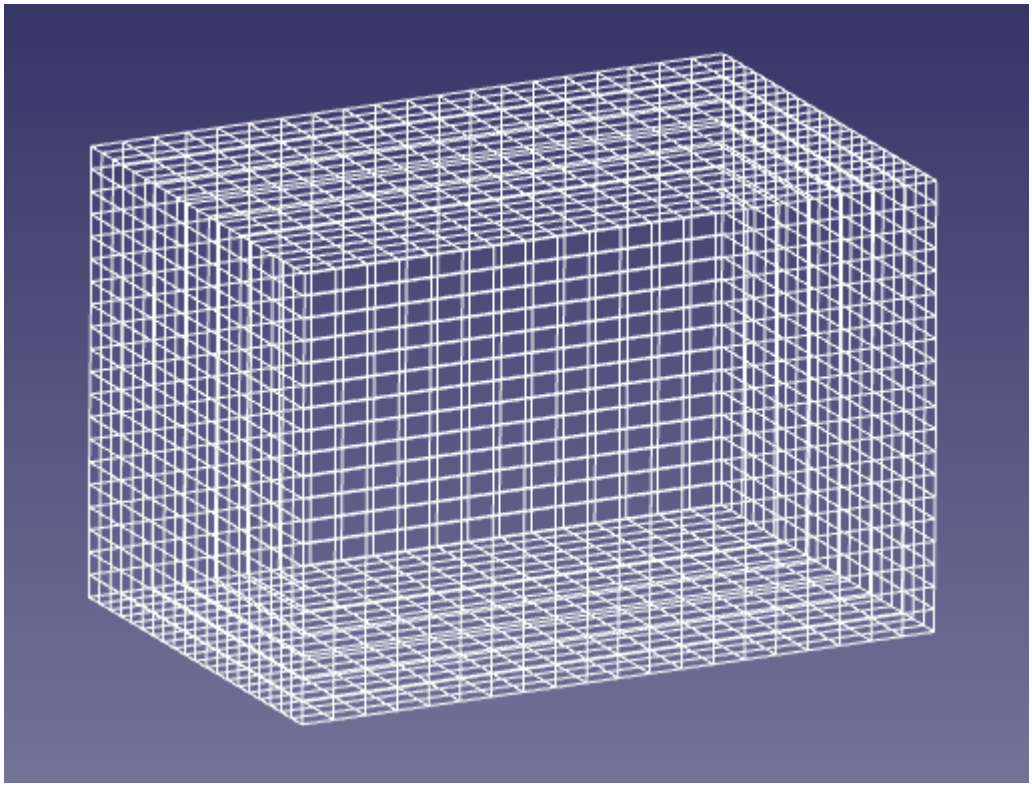
Parameters are *inheritable*, so parameters, which were set for elements of a higher level automatically apply to elements of lower levels. When so, parameters that are explicitly set for lower-level modules are override by higher-level module parameters.

See also: section Layers in the «Solids» folder, user interface.

6.16.1.3 Layer «Initial grid»

The **Initial grid** layer displays projections of the initial calculation grid (which is the computational grid of *level* zero) on faces of the «box», which contains the computational domain.

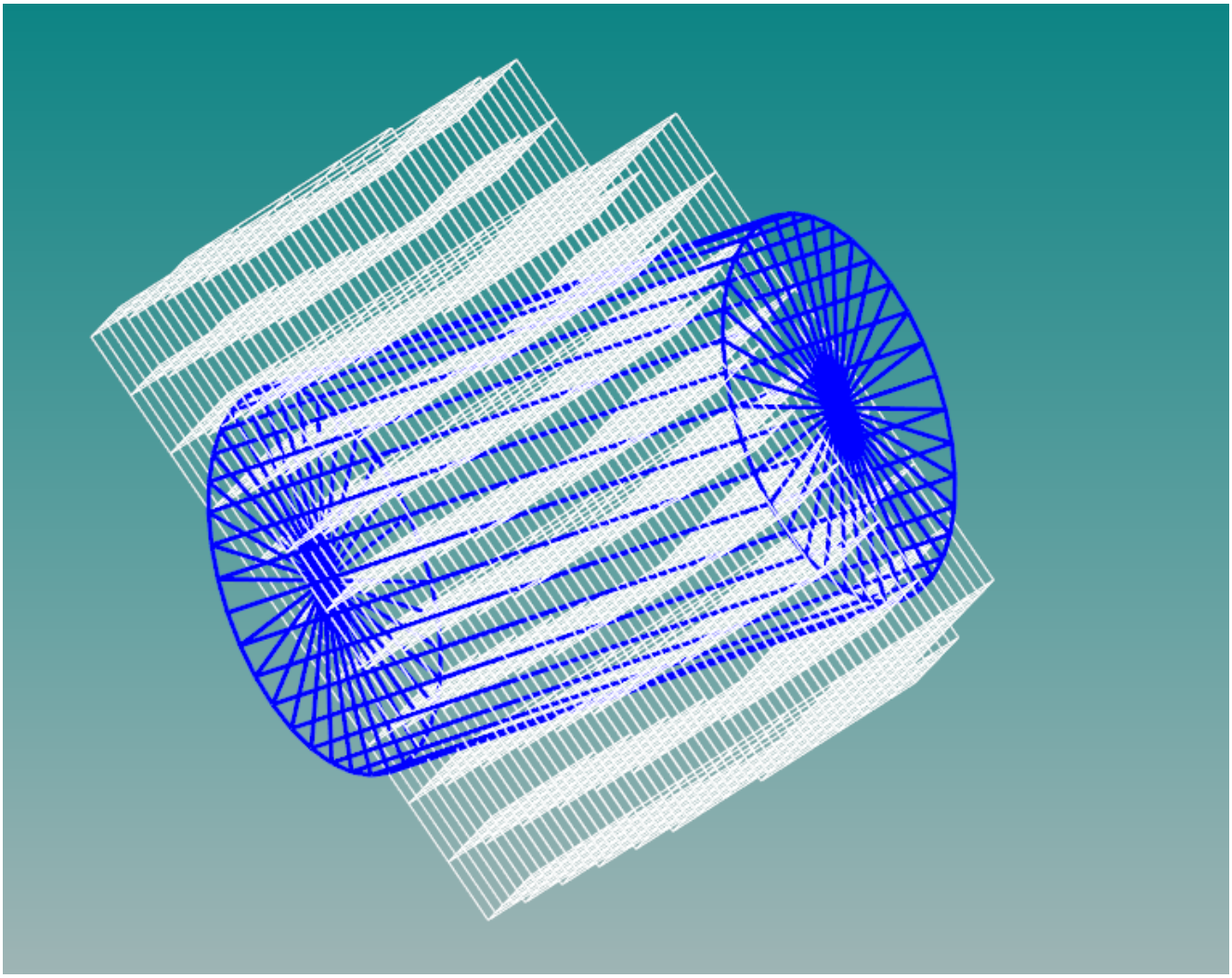
Data for building this layer is taken from the client part of the project, so the **Initial grid** layer *can* be built before running the calculation.



Layer **Initial grid**

See also: section Layer «Initial grid», user interface.

6.16.1.4 Layer «Computational grid»



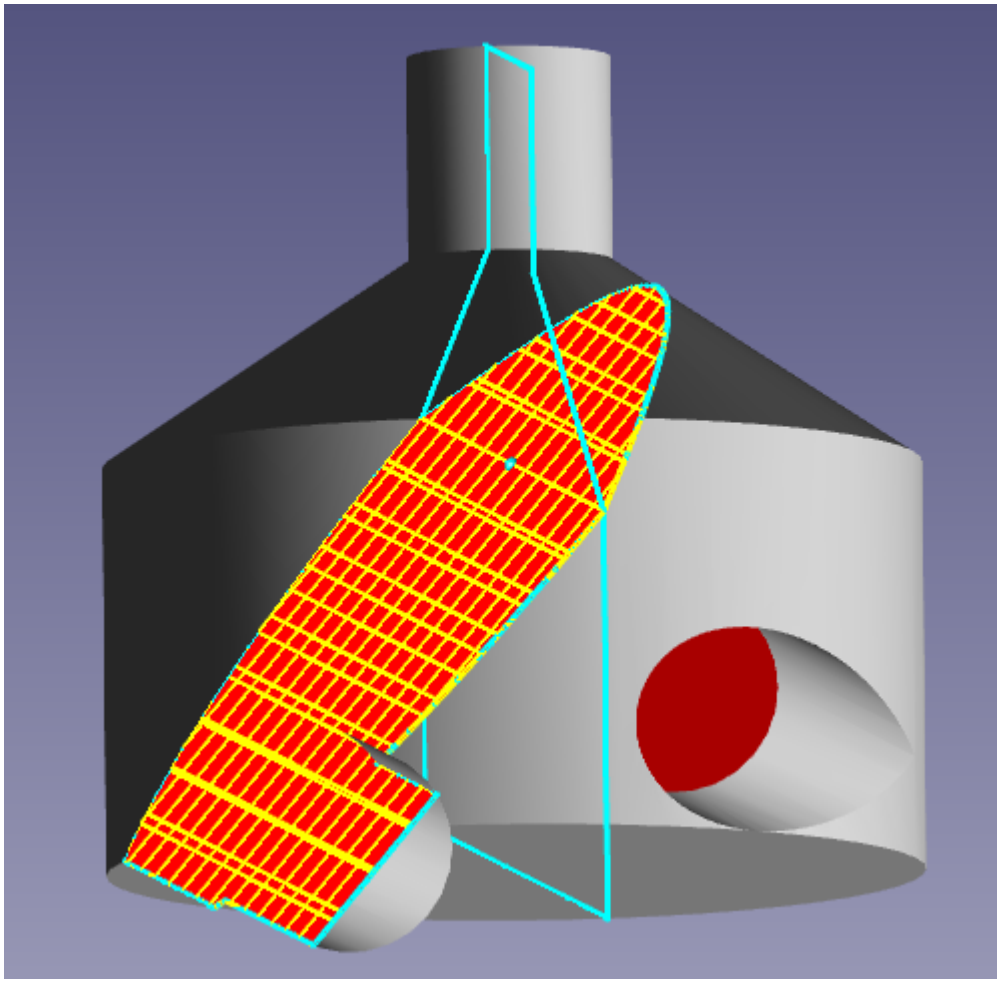
The **Computational grid** layer displays the computational grid in the volume of the whole computational domain or on a some **Object** (for example, on a **Plane**).

Cells can be displayed smaller than their real size, this can be useful for better visualization.

See also: section Layer «Computational grid», user interface.

6.16.1.5 Layer «Computational grid section»

The **Computational grid section** layer displays the cross section of the computational grid by a plane or a surface:



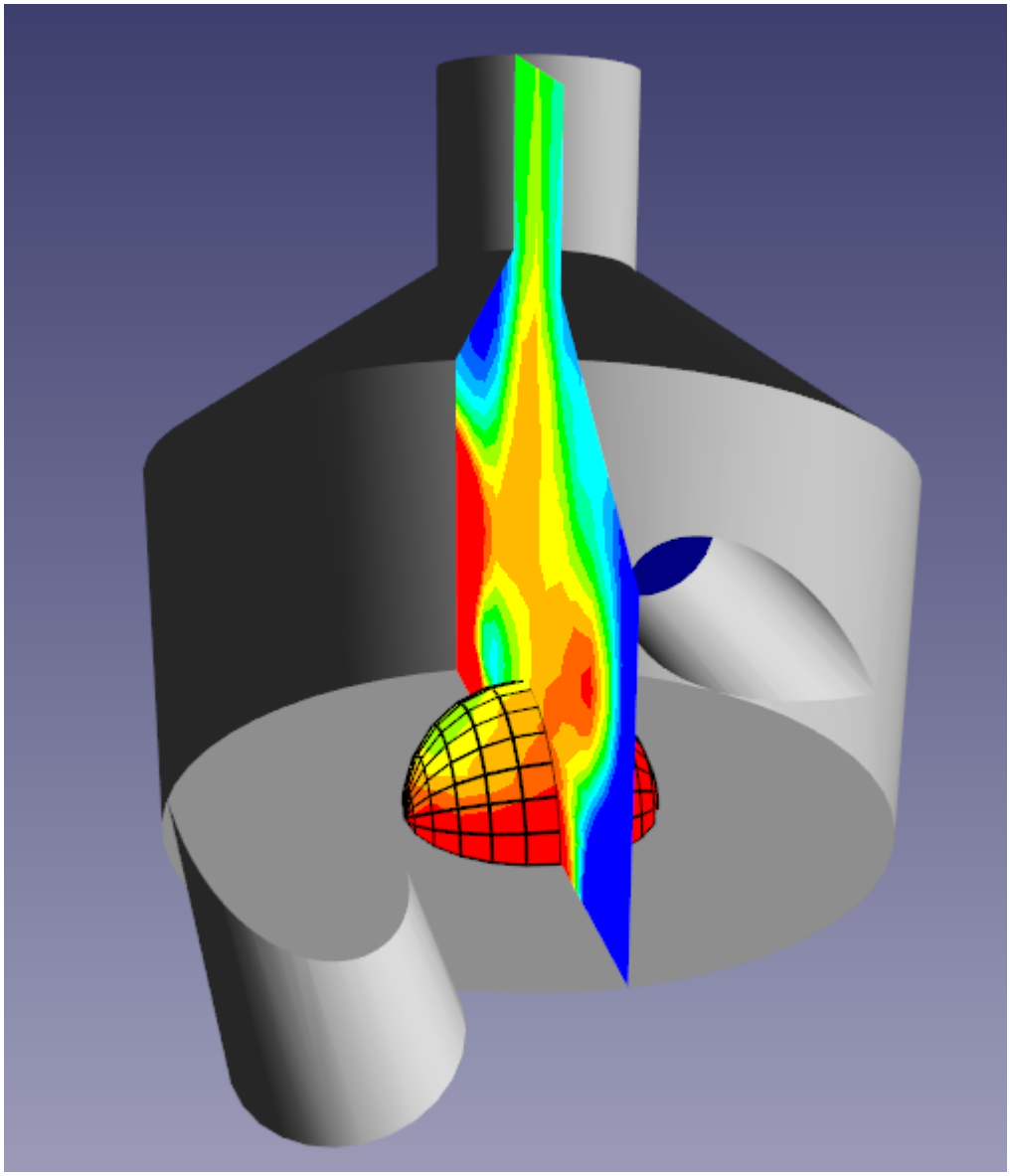
Layer **Computational grid section**

The **Computational grid section** layer allows you to view the subgrid resolution of the computational domain's geometry model.

See also: section Layer «Computational grid section», user interface.

6.16.1.6 Layer «Color contours»

The **Color contours** layer displays is the distribution of a variable over the surface. In the layer, you can construct the distribution of a scalar variable, component or module vector variable.



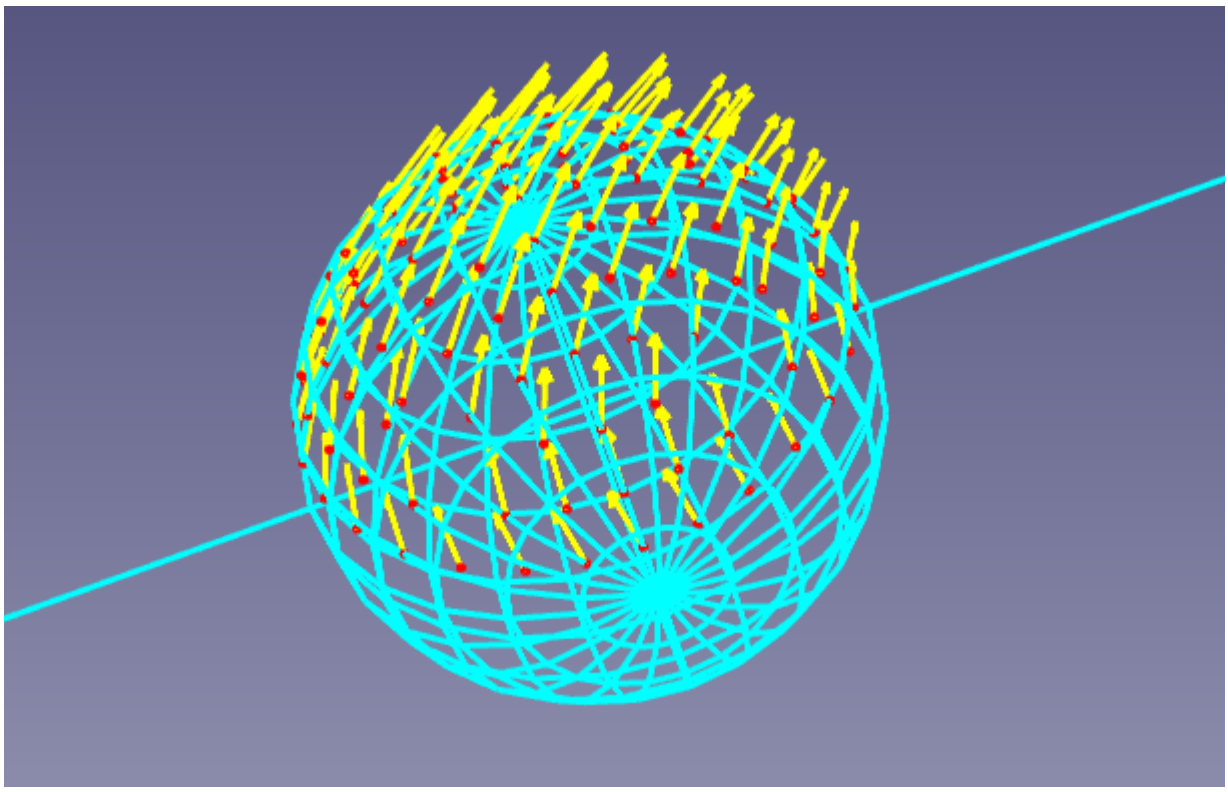
The visualization can be of one of three types (color corresponds to the given values of the palette):

- isolines variable
- a fill color changeable
- a wireframe facet surface with painted edges

See also: section Layer «Color contours», user interface.

6.16.1.7 Layer «Vectors»

The **Vectors** layer displays field of a vector variable in some volume or on a surface.



The "Vector" layer, which is built on the surface of a sphere

Accordingly, the **Layer**'s **Object** can be:

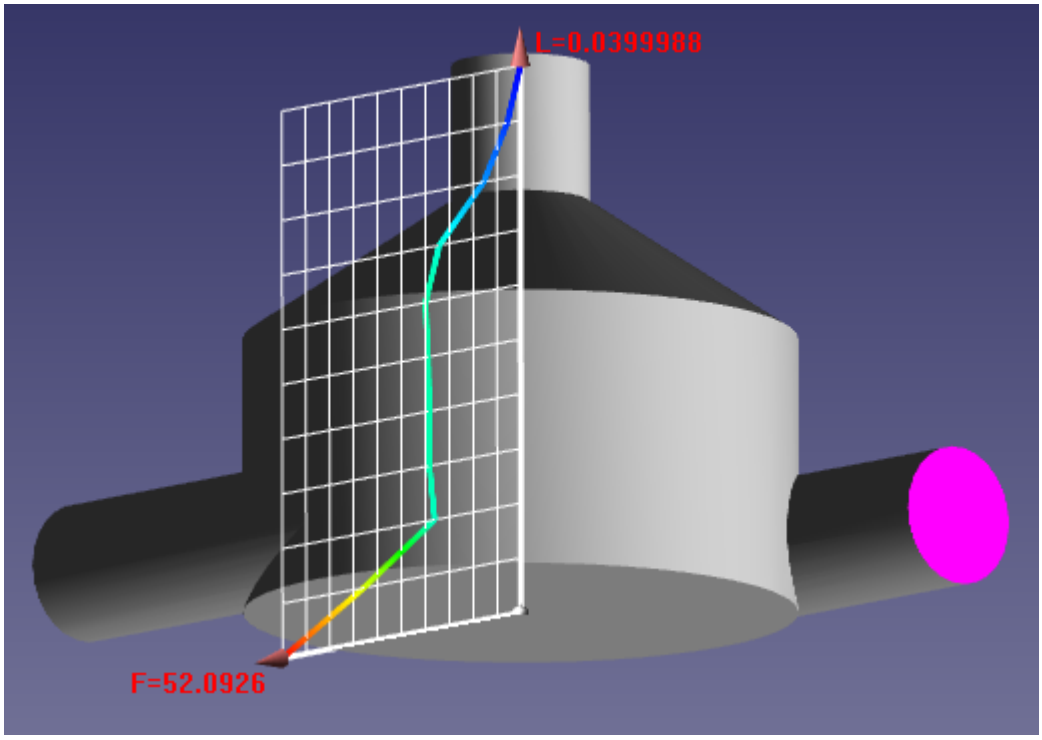
- the whole space of the computational **Subregion**
- objects of finite volume (standard or imported ones)
- **Planes**
- **Supergroups**

Vectors can be colored according to the value of another variable (this is set in the **Coloring** group of parameters).

See also: section Layer «Vectors», user interface.

6.16.1.8 Layer «Plot along line»

The **Plot along line** layer displays the image of the selected variable along this line.



Displayed on the plot variable can be:

- scalar
- component of a vector variable
- unit vector variable

The x-axis plot is specified by a layer:

- **Line** object
- line in the plane coming out of the reference point of the object **Plane**

this line is directed along the projection of one of the coordinate axes of the absolute coordinate system on the plane (in the direction of the coordinate axis absolute coordinate system or against it).

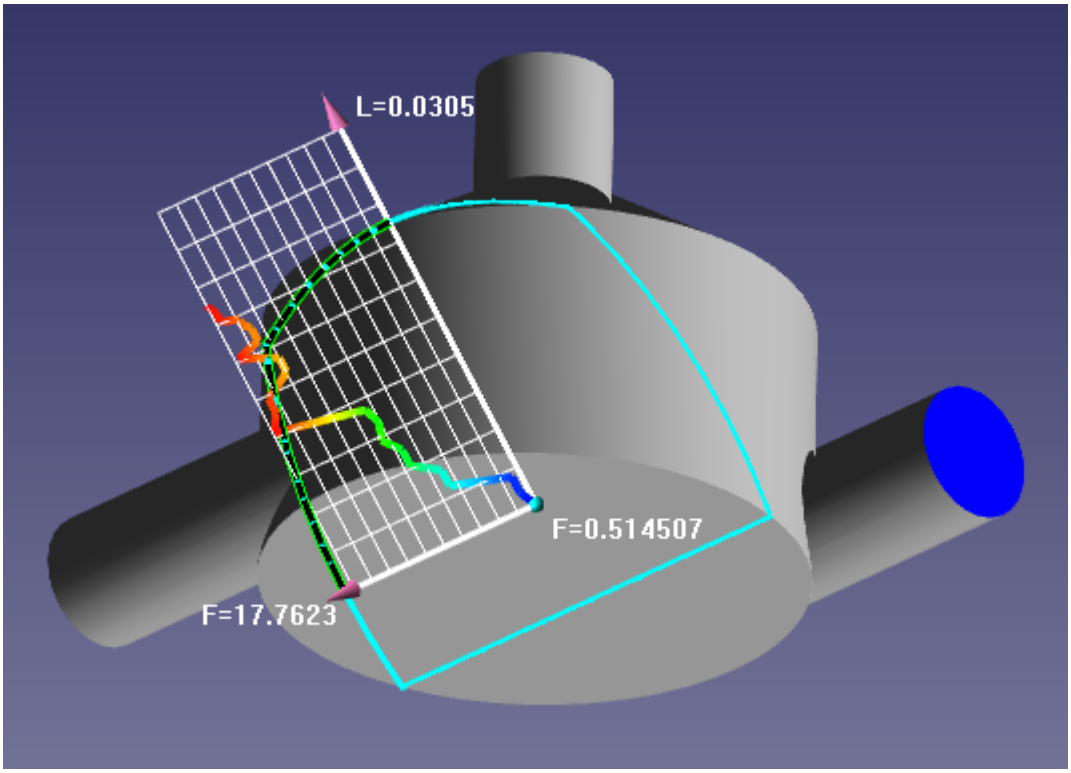
If the dates specified on the object **plane**, the change in direction of the x-axis (the line along which a plot) is made using parameter angle of rotation in the **Properties** window (interface description see in section Layer «Plot along line», user interface).

The origin coincides with the plot reference point **Line** or **Plane**. Plot is based on a straight line to its intersection with the boundary of the computational domain.

Line displayed on the plot can be colored according to the value of another variable.

6.16.1.9 Layer «Plot along curve»

The **Plot along curve** layer displays a plot of a given variable along a curved path formed by the intersection of the plane (object layer) and the boundaries of the computational domain.



Displayed on the plot variable can be:

- scalar
- component of a vector variable
- unit vector variable

Such curves (contours) can be several in this case, the values on the contours are displayed as multiple curves on the plot. These curves are numbered: [0] [1], ...

The surface on which the curve is, can separate two calculated subregion. In this case, each curve should specify a subregion data to be displayed on the plot. This choice is made in parameter **Subregion** layer.

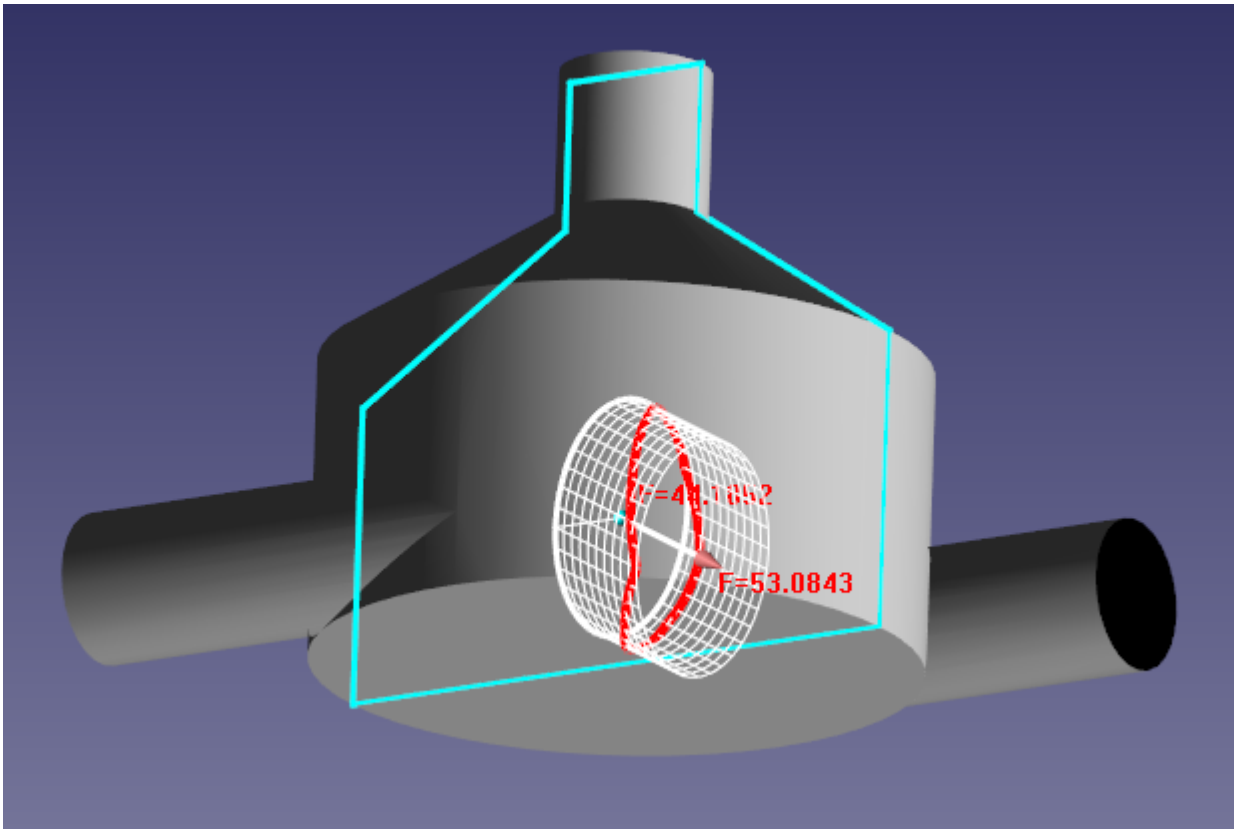
The object is to layer plane. Starting point of the plot coincides with the reference point of the plane. The abscissa axis is directed along the lines of one of the coordinate plane. On the x-axis is projected only a part of the curve that falls in the first quarter of the coordinate plot. Displayed on the chart portion of the curve is highlighted.

Line shown in the chart can be colored according to the value of another variable (the group of parameters **Coloring**).

See also: section Layer «Plot along curve», user interface.

6.16.1.10 Layer «Plot along ellipse»

The **Plot along ellipse** layer displays the plot of a given variable along the ellipse / circle.



Displayed on the plot variable can be:

- scalar
- component of a vector variable
- unit vector variable

Ellipse / circle is given by one of two ways:

- if an object layer selected **Line**:
  - a plane is built perpendicular to the direction vector and the line passing through datum line
  - in this plane carried ellipse / circle centered at the reference point of the line
- if an object layer selected **Plane**:
  - from the datum line is drawn perpendicular to the plane;
  - carried out in the plane ellipse / circle centered at the reference point of the plane

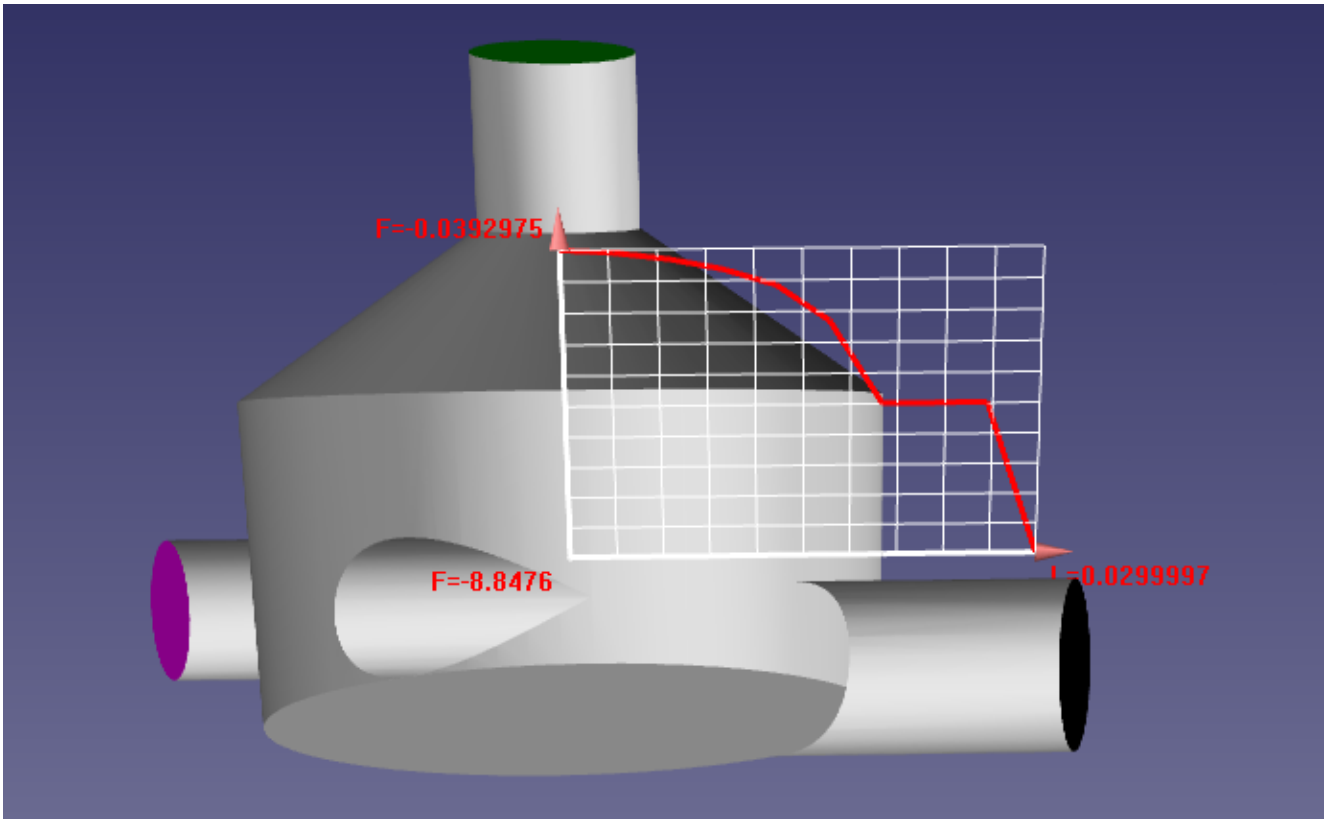
Further, through an ellipse / circle as the image is held cylindrical surface with its axis coinciding with the line. On this surface is plotted, assuming that the horizontal axis extends along an ellipse / circle, and the vertical axis - along the axis of the cylindrical surface.

Line shown in the chart can be colored according to the value of another variable.

See also: section Layer «Plot along ellipse», user interface.

6.16.1.11 Layer «Distributed characteristics»

The **Distributed characteristics** layer displays a plot of a certain function (see description below).



The main parameters of the **Distributed characteristics** layer have the following structure:

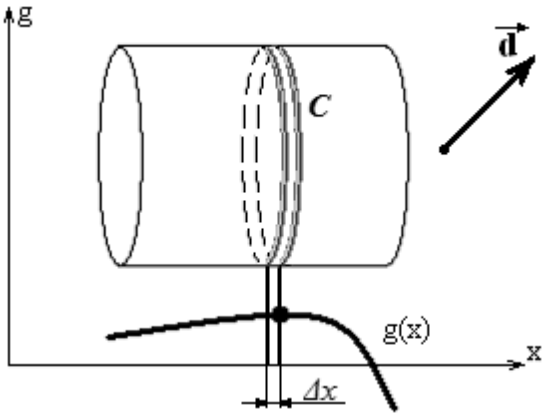
Object + Surface/Volume + Function of the variable + Direction

where:

- **Object** is a **Line** or a **Plane** that define the x-axis of the plot
- **Surface/Volume** - the following options are possible:
  - *surface* of an **Imported object** or **Supergroup**
  - *surfaces* of all **Imported objects**, all **Supergroups**, and boundaries of the computational domain
  - *volumes* limited by surfaces of all **Imported objects**, all **Supergroups**, and boundaries of the computational domain, and adjacent cutting planes that are orthogonal to the x-axis of the plot.
- **Function of the variable** - the following options are possible:
  - The surface integral of a scalar or vector **Variable** over the strip, which is formed by two adjacent **Surface**'s sections that pass orthogonally through adjacent points on the x-axis of the plot. This integral defines either a *scalar or a vector function* along the x-axis of the plot.
  - The volume integral of a scalar or vector **Variable** over a slice (thin plate) formed within surfaces of all **Imported objects**, **Supergroups**, boundaries of the computational domain, and adjacent cutting planes that are orthogonal to the x-axis of the plot. This integral defines either a *scalar function* along the x-axis of the plot.
- **Direction** (specified by a direction vector), on which the computed **Function of the variable** is projected if it is a *vector function*. This setting is not available and is not applied for scalar **Functions of the variable**.

In the layer of the x-axis is specified by a layer:

- when the **Object** is a **Line** then the x-axis is the **Line** itself
- when the **Object** is a **Plane**, then the x-axis is a line in the plane, which goes through the base point of the **Plane** object and goes along a projection of one of the coordinate axes of the absolute coordinate system on the **Plane** (along direction of the coordinate axis of the absolute coordinate system or against it). Also the user can change the direction of x-axis using the **Rotation angle** parameter.



Layer **Distributed characteristics**.

x is the x-axis of the plot; $g(x)$ is the scalar function displayed by the **Layer**;

C is a contour of surface's section or a volume slice, over which the integrating is done;

$\boldsymbol{d}$ is the vector that defines the **Direction**, on which **Function of the variable** is projected (when this function is a vector one)

The origin of the plot is the base point of the **Line** or **Plane**.

The plot shows the function g (x), which is constructed as follows:

1. The program finds an interval in the abscissa axis x of the plot, where **Function of the variable** will be calculated. Ends of this interval are found by projecting the **Surface/Volume** to the abscissa axis x of the plot.
2. The found interval of the abscissa axis x of the plot is split into equal sub-intervals (number of these sub-intervals is set by the **Number of points** parameter).
3. From the ends of these sub-intervals planes are build orthogonally to the abscissa axis x of the plot. These planes:
   - split surface of **Imported objects** / **Supergroups** / boundaries of the computational domain to strips Δx
   - *or* split volume **Imported objects** / **Supergroups** / boundaries of the computational domain to slices Δx
4. On each of the stripes or slices the program calculates:
   - a scalar function $F(x)$ as an integral over the contour C
   - *or* a vector function $\boldsymbol{F}(x)$ as an integral over the contour C
   - *or* a vector function $\boldsymbol{F}(x)$ as a volume integral in the slice C.

When the **Variable** is a scalar one and integrating is vector and it is done over a surface, then the **Variable** will be multiplied on normal vectors **n** to the surface.

Only scalar integrating is implemented over a volume slice L.

A scalar integrals over the contour C is divided by length of the contour, so the result of the computation will be the average value on the contour C.

Here are the possible cases:

$$\boldsymbol{F}(x) = \oint_C f(c)\cdot \boldsymbol{n}\, dc$$

(for scalar variables and vector integrating along the contour C)

$$\boldsymbol{F}(x) = \oint_C f(c)\, dc$$

(for vector variables and vector integrating along the contour C)

$$F(x) = \frac{1}{C}\oint_C f(c)\, dc$$

(for scalar variables and scalar integrating along the contour C). The calculated integral is divided by length of the contour C, so result will be average value over the contour.

$$F(x) = \frac{1}{C}\oint_C \left|\boldsymbol{f}(c)\right| dc$$

(for vector variables and scalar integrating along the contour C). The calculated integral is divided by length of the contour C, so result will be average value over the contour.

$$F(x)=\iiint_C f(c)\,dc$$ (for scalar variables and volume integrating over the slice C)

$$F(x)=\iiint_C |\boldsymbol{f}(c)|\,dc$$ (for vector variables and volume integrating over the slice C)

5. Function $g(x)$, which is displayed by the **Layer**, is calculated as follows:
   - when **Function of the variable** $\boldsymbol{F}(x)$ is a vector function, $g(x)$ is calculated as a scalar product of $\boldsymbol{F}(x)$ and a user-defined directing vector $\boldsymbol{d}$.
   - when **Function of the variable** $F(x)$ is a scalar function, $g(x)$ is equal to $F(x)$.
6. The plot of $g(x)$ is drawn until its abscissa axis x goes out of the computational domain.

In visualization of the **Distributed characteristics** layer, the abscissa axis x of the plot is marked with letter **L**, and the ordinate axis $g(x)$ of the plot is marked with letter **F**.

See also: section Layer «Distributed characteristics», user interface.

6.16.1.12 Layer «Isosurface»

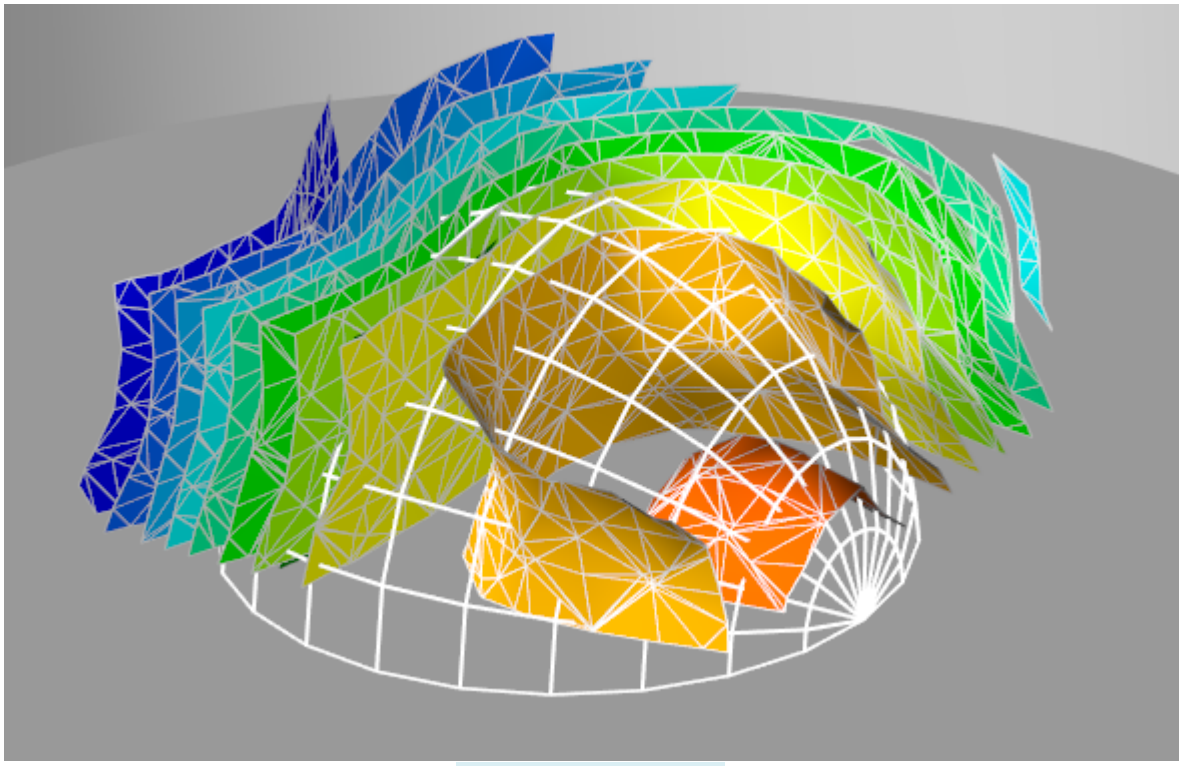
The **Isosurface** layer displays the image of one or more isosurfaces (iso-surface - the surface on which some variable is constant set point).

On the isosurface given variable has the same value, while variable can be:
- scalar
- component of a vector variable
- unit vector variable

The **Layer**'s **Object** can be all the space or an **Object** of finite volume (standard or imported).

Isosurface can be colored according to the value of another variable.

In constructing the isosurface is used to smooth the corners relaxation method that leads to a small deviation from the calculated surface. The amount of deflection can be adjusted (it increases with the **relaxation** parameter).

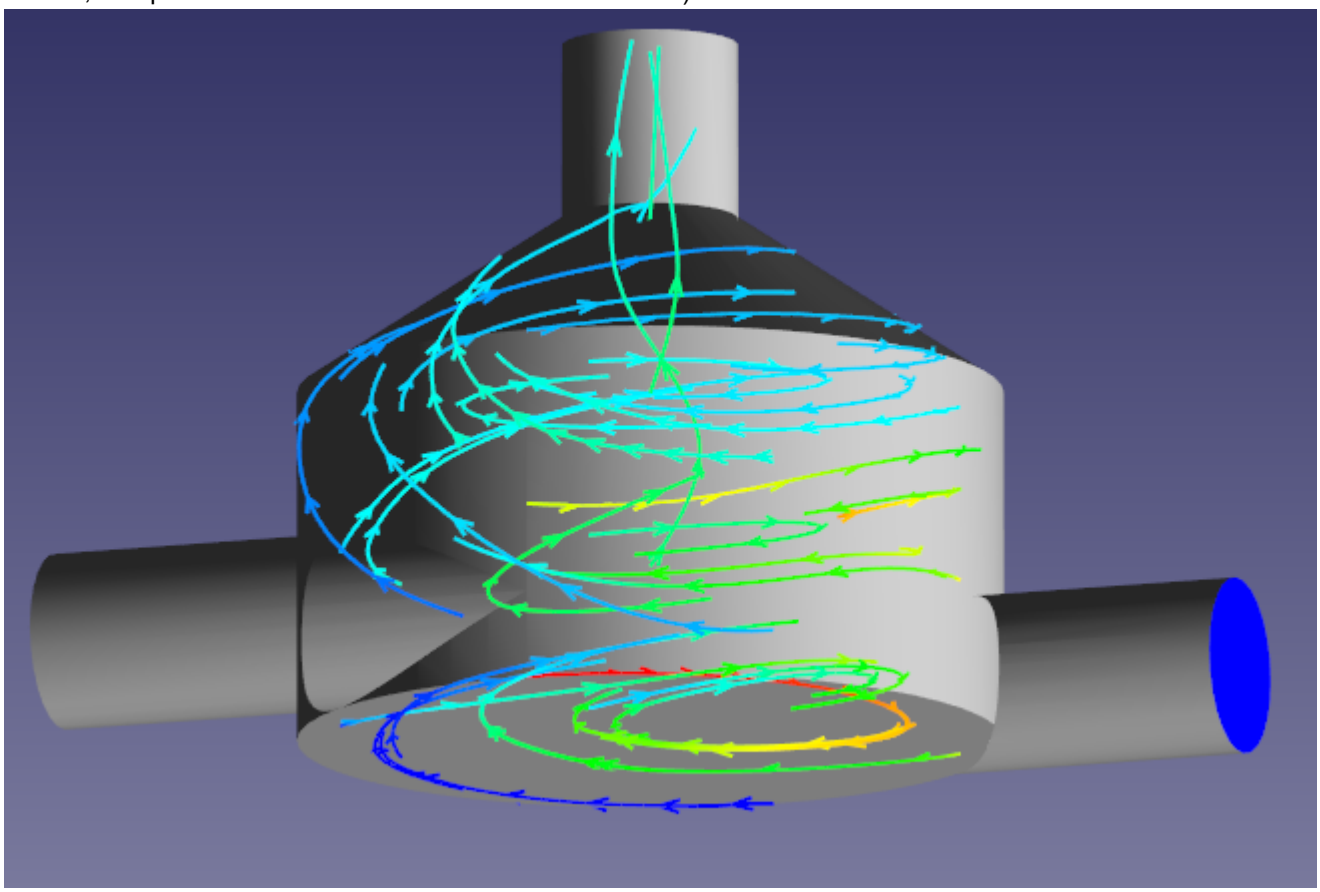


Example of isosurfaces

See also: section Layer «Isosurface», user interface.

6.16.1.13 Layer «Streamlines»

The **Streamlines** layer displays the field of a vector variable (for example speed) as streamlines located in the **Computational space** or on the surface of a **Supergroup** (in the latter case only the tangential, relating to the surface, component of the visualized vector variable is used).



Streamlines, which have been built on the surface of a **Supergroup**, terminate when they go away or substantially deviate form the surface.

Streamlines can be colored according to the value of another variable.

It should be remembered that streamlines coincide with the trajectories of fluid particles only in a steady stream. **Streamlines** in an unsteady stream has no physical meaning.

See also: section Layer «Streamlines» and element «Emitter for streamlines», user interface.

6.16.1.14 Layer «Nodal loadings»

The **Nodal loadings** layer displays vectors of loadings in the nodes of the computational grid (which has been built in *FlowVision*) on the surface of an **Imported object**. Data on this layer are calculated *only* during a joint computation of the flow (*FlowVision*) and stress-strain analysis of the construction *(Abaqus)*.

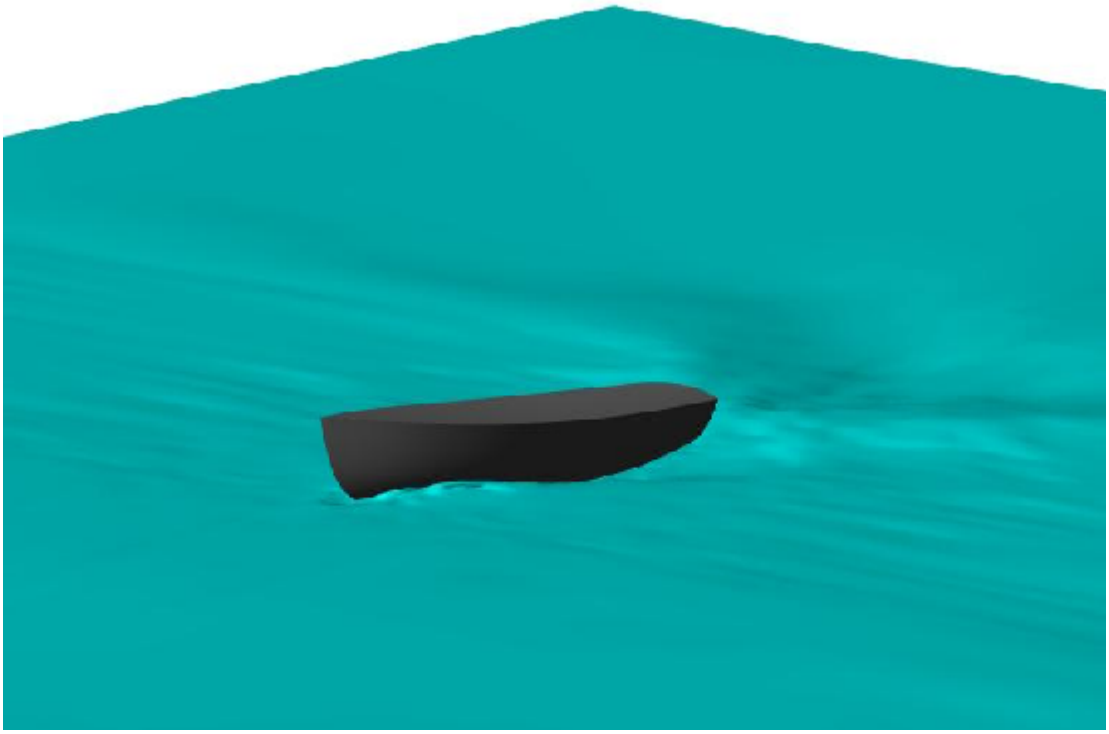
During the joint computation with *Abaqus*, the **Nodal loadings** layer is not available in **Pre-Postprocessor**.

The vectors can be colored according to the values of temperature or heat flux.

See also: section Layer «Nodal loadings», user interface.

6.16.1.15 Layer «VOF» (visualization of a free surface)

The **VOF** layer displays the phase interface surface, generated by the VoF calculation method, in the space of the computational domain.



The *Volume of Fluid (VoF)* method can *not* be applied in the following cases:

- When subregions are conjugated by velocities, if the inter-phase surface intersects the boundary of the conjugation
- To simulate **Dispersed Phases**
- When using sliding surfaces, if the inter-phase surface intersects the sliding surface
- To simulate interaction of two **Continuous Phases** if no **Motion** physical process is specified in both these **Phases**.

See also: section Layer «VOF», user interface.

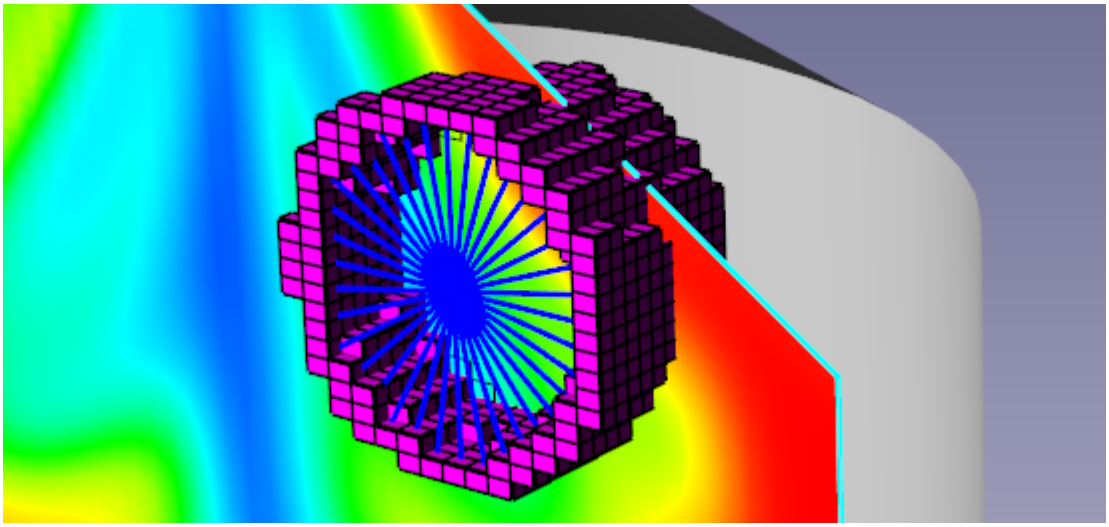
6.16.1.16 Layer «Cell set»

The **Cell set** layer displays computational grid's cells of certain types.

The **Cell set** layer, together with the **Cell debug** layer, is used to evaluate the quality of the grid, which has been built.

The **Cell set** layer displays cells of some type, which has been specified by the user, and which are within a specified geometric **Object** and/or contacting the **Object**.

Data for building this layer are received from **Solver**, so the layer cannot be built before the computation is started.



Example of the **Cell set** layer, which has been built on a side surface of a cylinder (the displayed cells are highlighted with pink color)

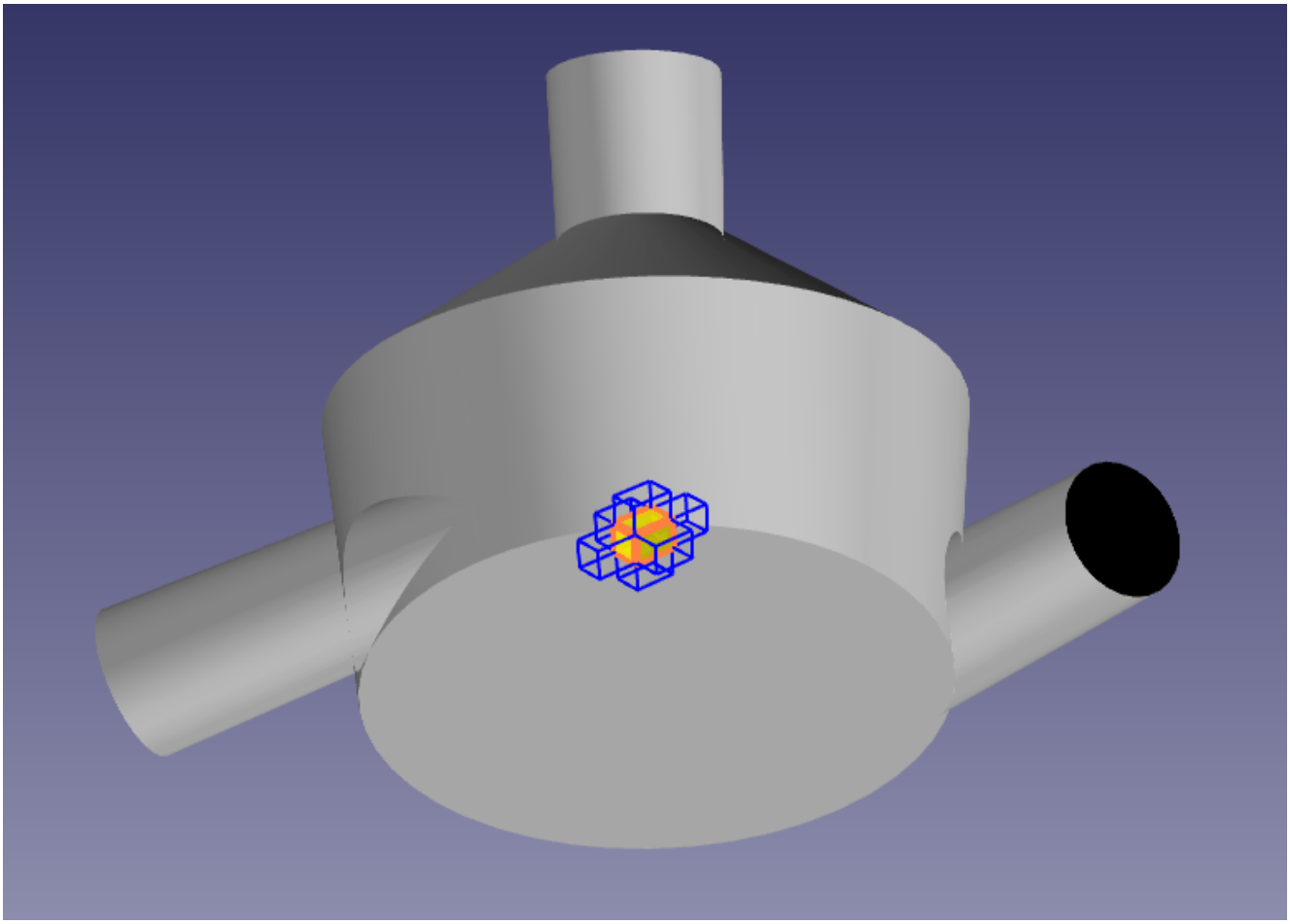
The **Cell set** layer can be built on the following geometric **Objects**:

- **Computational space**
- **Line**
- **Plane**
- **Box**
- **Cone/cylinder**
- **Ellipsoid/sphere**
- **Imported object**
- **Set of sensors**

See also: section Layer «Cell set», user interface.

6.16.1.17 Layer «Cell debug»

The **Cell debug** layer displays the location of a cell of the computational grid in the subregion. Additionally, the **Info** window displays values of a variable in the cell.



The **Cell debug** layer is used for the following purposes:

- in the analysis of the quality of meshing together with a layer of **Cell set**
- to obtain the values of all variables at the computational domain - the center of the computational cell

Adjacent cells can be displayed smaller than the real, which shows locations of cells near the selected cell.

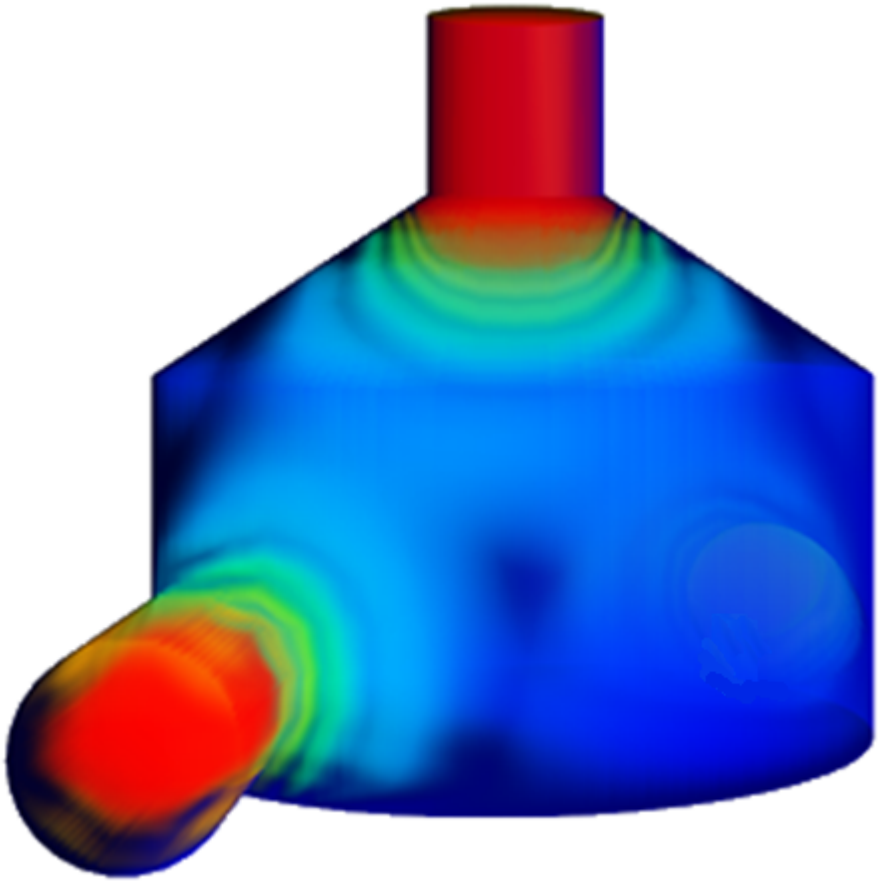
When you create this layer on a surface or a line **Object**, the debug cell will be the cell, which locates in the reference point of the **Object**. When the reference point is moved, the debug cell also moves along the **Line** or in the **Plane**.

Chip called the face of the cell surface, a cut after applying a geometry model of the surface or the moving body, and after the construction of the free surface.

See also: section Layer «Cell debug», user interface.

6.16.1.18 Layer «Volume visualization»

Layer **Volume visualization** displays distribution of some variable in a volume.



Layer **Volume visualization** allows you to visualize changing variables in the whole space (for example, view zones with high substance concentration or density).

The visualized variable can be:

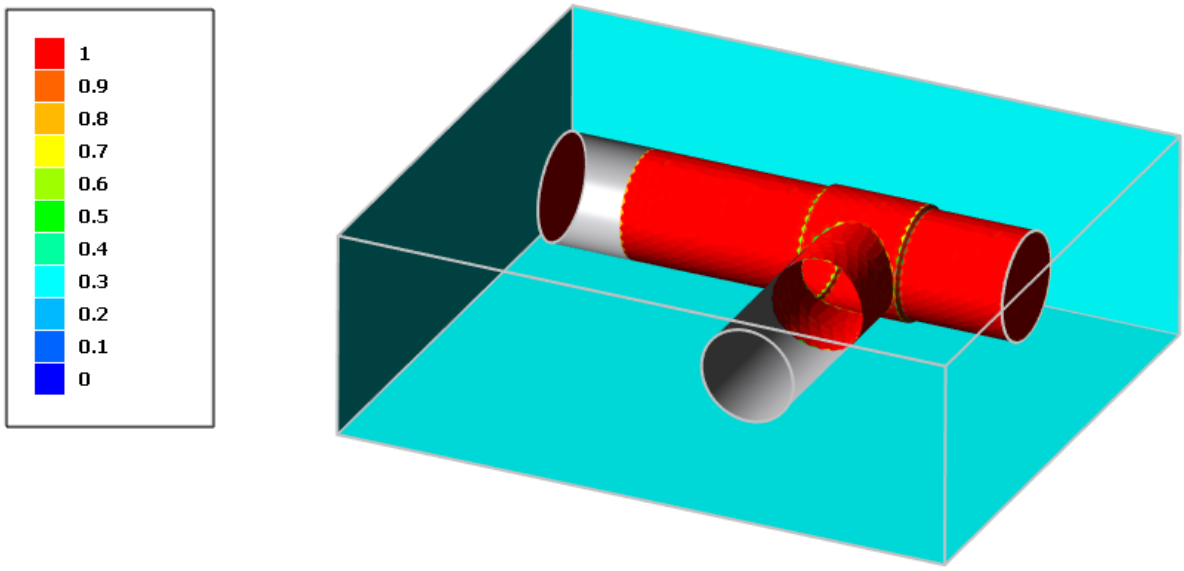
- scalar
- a component of a vector variable
- the modulus (absolute magnitude) of a vector variable

Before use of the **Volume visualization** layer ensure that you have installed the last versions of video card drivers.

See also: Layer «Volume visualization», user interface.

6.16.1.19 Layer «Mapping surface»

The **Mapping surface** layer displays distribution of the value $f_i^{(1)}$ over the source surface when mapping is used.



When $f_i^{(1)}$ = 0, appropriate triangles of the surface will be invisible.

See also sections:

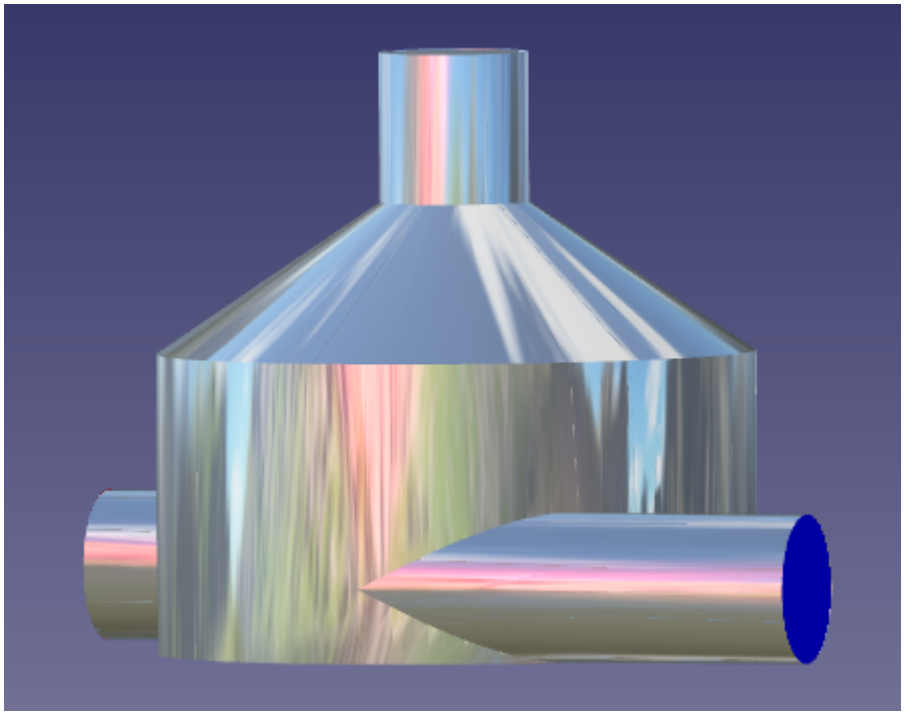
- Mapping
- Layer «Mapping surface», user interface

6.16.2 Displaying surface's material

To achieve a more realistic image of the surface of solids in *FlowVision* is possible to create a separate set of display properties of a particular material. These sets of properties, referred to as the tree of the "materials" are stored in the folder **Materials** Project Tree tab **Postprocessor**. When forming images of the surface, the user can take the "materials" folder **Materials**.

If, after comparing the display surface of the appropriate "materials" change the "materials" folder **Materials**, the visualization in the **View** window will change accordingly.

On the inner surface of the "box" attached picture (a landscape picture), which may be reflected in a mirror surface geometry model of the computational domain.



Specifying the properties of the material can be obtained high gloss surfaces and reflections of a given color (in the example on the illustration they are red)

6.16.3 Data analysis

There are two ways of computational and visual data analysis:

- analyzing data acquired at the last time step
- as layer visualization data

Parameter values in the specified computational cell and their changing can be monitored in the **Info** window.

6.16.3.1 Analysis of the data saved at the last time step

The *project's computation results* and *layer visualization data* in the last time step:

- locate in the server part of the project
- are full data related to a *single time moment*
- are available for analysis in **Pre-Postprocessor** only when **Pre-Postprocessor** has connection to the **Solver**, on which the project is loaded

Analysis of the calculated data is done by:

- forming and displaying layers (including those layers, which were defined before start of the project's computation)
- specifying the parameters of characteristics and displaying their components in the **Info** window
- displaying in the **Monitor** window plots of dependencies (against time) of variables or characteristics that were specified as criteria for stopping the computation.

6.16.3.2 Analysis of the data saved at several time steps

The computational data can be saved at a set of time steps in two formats:

- as full calculation results (the calculated variables fields)
- as layer visualization data

Calculation results and/or layers autosave must first be enabled to allow the saving of computational data.

A deficiency of the analysis method of computational data saved at a set of time steps is its resource-intensive nature.

Calculation results saved at intermediate time steps can be partially removed in the **Non-steady-state steps decimation** dialog box.

Analyzing a set of saved calculation results

Calculation results saved at a set of time steps:

- are located in the server part of the project
- represent full data characterizing several moments in time according to the autosave settings of calculation results
- can be automatically viewed sequentially, after which a set of animation frames can be compiled
- are available for analysis in **Pre-Postprocessor** only if it is connected to the **Solver**, on which the project has been loaded.

To analyze the saved calculation results:

- setup the calculated data save mode mode before the start of the calculation
- after the calculation is finished, set the parameters of layers displaying the calculation results
- launch automatic layer display at the saved set of time steps; changing time-steps updates all layer images; set the automatic screenshot mode for the **View** window to get the animation frames.

Analyzing a saved sequence of layer visualization data

Layer visualization data that were saved at a set of time steps:

- are located in the client part of project (the `*.fvvis` files must be copied from the server part)
- are partial and constrained by the layer parameter defined in the **Pre-Postprocessor** before the start of calculation
- are available for analysis in **Pre-Postprocessor** only if it is NOT connected to the **Solver** on which the project is loaded

Viewing predefined layers with data saved at a set of time steps is a less resource-intensive analysis method but, in this case, only the predefined data is available for viewing.

Layer visualization data saved at a set of time steps can be automatically viewed sequentially in **Viewer**.

Pre-Postprocessor has two layer display modes:

- one by one, loaded from `*.fvvis` files
- sequentially in the automatic mode

In both cases, it is necessary:

- before the start of calculation:
  - set up the saved layers parameters
  - set up the image layers save mode
- when the calculation is finished, copy the layer data saved during the calculation process from the server part of project to the client part.

Viewing separate layers comes to loading data from `*.fvvis` files in the **View** window of the image created based on the loaded data.

The automatic sequential layer view mode is available only when certain conditions are met. **Pre-Postprocessor** supports the automatic view mode only for sets of saved calculation results. This set is a list that can be navigated from the **Navigation** toolbar. If some of the time steps in this list have layer visualization data, they will be displayed in the **View** window. The `*.fvvis` files saved at time steps with no calculation results will not be displayed in this mode. To view the whole `*.fvvis` file set, the calculation results and the layer visualization data must be saved synchronously.

While viewing layer data, the automatic screenshot saving mode for the **View** window can be set in order to get animation frames.

Viewing calculation results in dynamic

If the automatic screenshot mode for the **View** window was enabled while viewing the sequence of saved calculation results, the resulting animation frames can be compiled into standalone animation in a suitable graphics editor.

Viewing the animation can help in the analysis of project solution results.

6.16.3.3 Viewing values of parameters in a cell

If you create a **Cell debug** visualization layer and specify in its properties a cell number, then in the **Info** window of this layer will display values of all of this layer's variables.

See details in section Layer «Cell debug», user interface.

6.17 Storing input data and settings

Autosave during calculation, backup

The results of the project tasks in **Pre-Postprocessor** are stored in file sets in the client part of project. When the project calculation is started for the first time, these files are copied from the client part of project to the server part.

If the project task is interrupted by **Solver**, the project input data are modified and the modified project is restarted for calculation, then in the servers part of the project a file is created with a copy of a new input data.

These changes may affect:

- the main project parameters (specified on the **Preprocessor** tab)
- the simulation controls (specified on the **Solver** tab)
- display parameters (specified on the **Postprocessor** tab)

Also the user can refuse to continue the calculation and restart it from scratch using some initial data rather than the data from the last time step.

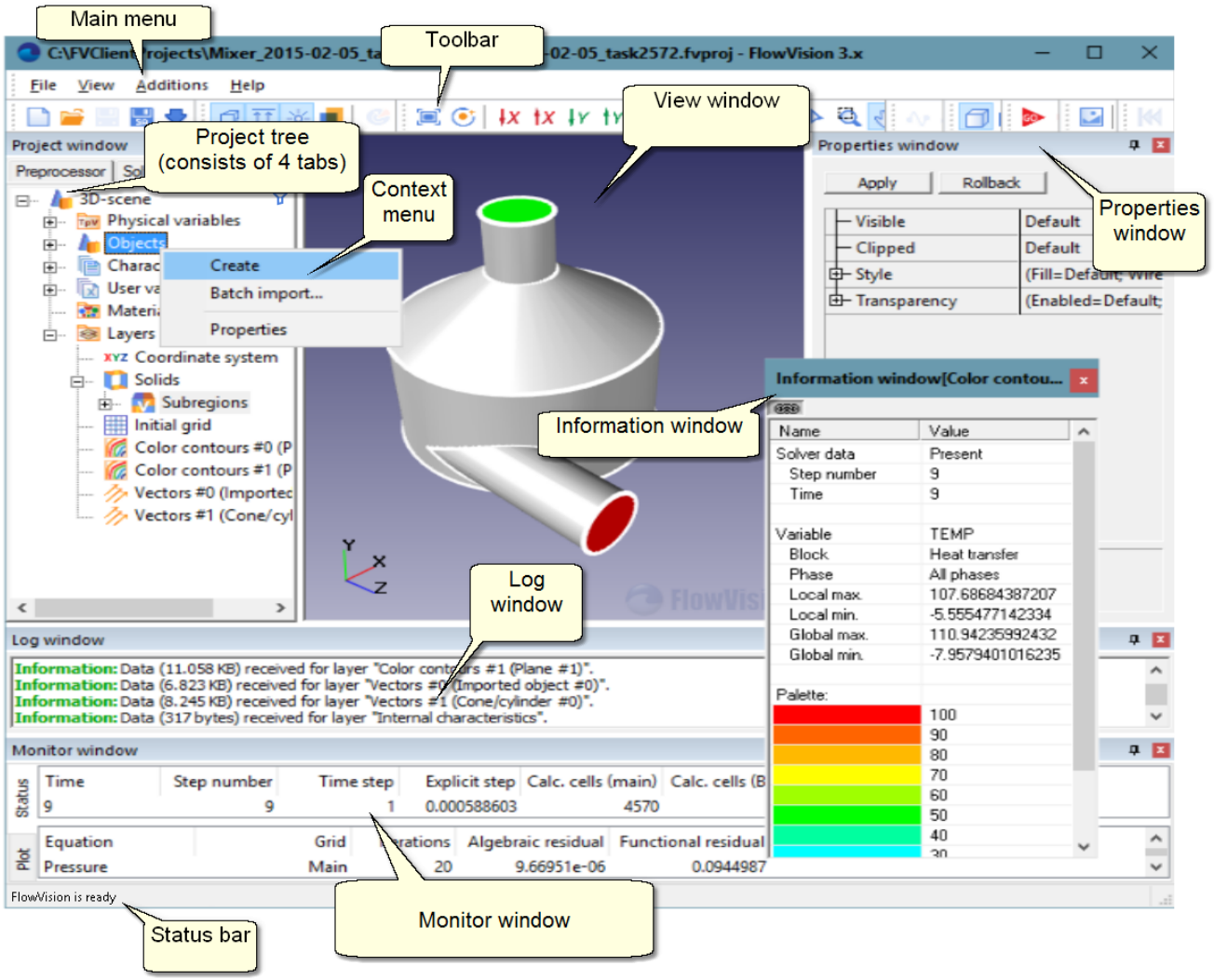
As a result, the server part of project has a complete set of input data received from **Pre-Postprocessor** - the input data history. This history is automatically copied to the client part of the project.

See also: Operations with backup.

7 FlowVision modules

Sections below contain descriptions of *FlowVision* modules:

7.1 Pre-Postprocessor



Pre-Postprocessor is intended to create projects, starting the project's computation and displaying the results.

See sections Work with Pre-Postprocessor and Quick start.

7.1.1 Pre-Postprocessor's error messages and warnings

Pre-Postprocessor's *error messages*[*)] are displayed on the screen in pop-up dialog boxes and/or in the **Log** window and they are recorded into the **Pre-Postprocessor**'s `log`-file.

Errors of loading project files usually arise because of missed or damaged project files.

Note:

[*)] For brevity, this term is applies to *errors*, *warnings* and *info messages*, unless explicitly stated otherwise.

Error messages of Pre-Postprocessor	
Post	**Description**
`Bad geometry`	The loaded geometry model of the computational domain contains an error: • **Boundary edges**- in the model there are edges, forming the boundary of an open surface • **Overlap**- in the model there are edges belonging to overlapping facets • **Hanging edges**- the edges in the model is not owned by a single facet

Error messages of Pre-Postprocessor	
Post	**Description**
	• **Multiconnection**- in the model there are edges belonging to more than two facets. Use of multiconnection can be allowed or blocked in basic settings of Pre-Postprocessor. *Possible cause:* The geometry model does not meet to requirements or has been damaged.
`Failed to load main geometry`	Error of loading the file **`fvgeom`**. *Possible cause:* file is missing or corrupted.
`Failed to load head`	Error of loading the file **`fvinp`**. *Possible cause:* file is missing or corrupted.
`Failed to load boundary conditions`	Error of loading the file **`fvbcs`**. *Possible cause:* file is missing or corrupted.
`Failed to load simulation controls`	Error of loading the file **`fvcrtl`**. *Possible cause:* file is missing or corrupted.
`Failed to load object geometries`	Error of loading the file **`fvgobj`**. *Possible cause:* file is missing or corrupted.
`Failed to load scene`	Error of loading the file **`fvview`**. *Possible cause:* file is missing or corrupted.
`Failed to load scene object geometries`	Error of loading the file **`fvgobj`**. *Possible cause:* file is missing or corrupted.
`Failed to load status`	Error of loading the file **`fvstat`**. *Possible cause:* file is missing or corrupted.
`Failed to apply status data to the project`	The **`fvstat`** file does not match to the file **`fvproj`**. *Possible cause:* the files are missing, damaged, or does not correspond to each other.
`Failed to load temporary status`	Error of loading the file **`fvtmps`**. *Possible cause:* file is missing or corrupted.
`Failed to apply temporary status data to the project`	The **`fvtmps`** file does not match to the file **`fvproj`**. *Possible cause:* the files are missing, damaged, or does not correspond to each other.
`Failed to load postprocessor viewport settings`	Error of loading the file **`fvpost`**, which contains display settings. *Possible cause:* file is missing or corrupted.
`Failed to load design table`	Error of loading the input file of external parameters, **`fvdtbl`**. *Possible cause:* file is missing or corrupted.
`Failed to load result table`	Error of loading the output file of exported results **`fvrtbl`**. *Possible cause:* file is missing or corrupted.
`Read-only mode alone is available without license. Continue?`	Error of access to the **License Manager**. After a confirmation the project will be loaded in the read-only mode. *Possible cause:* Incorrect settings of access to the **License Manager**. Examine and, if necessary, correct the settings in the group of settings "License manager" in the Basic settings of Pre-Postprocessor.
`Directory is write-protected. Project has been opened for viewing only.`	The project has been opened for viewing only and cannot be changed. *Possible causes:*

Error messages of Pre-Postprocessor	
Post	**Description**
	• The project's client directory has been placed in a system folder, for example, in **Program Files**, which is write protected. Allow writing to the directory or move the project to another directory that is not write protected. • There is no access to the license. See License management.
Failed to allocate memory block of requested size	*Possible cause:* This message can be caused by a layer, which is too difficult for building, in a large project (with several millions of cells).
Error in assembly process. Contact tech. support.	Errors of assembling a geometry model of the computational domain.
Multiconnection is detected in some input geometries. Check documentation for details.	
Error in assembly process: Geometry ... has self-intersection.	
Error in assembly process: Cannot remove proximity of surfaces.	
It is necessary to specify a model for subregions with T-connection.	When multiconnection is used, you have to specify in the project a **Model** in all **Subregions** that are adjacent to any surface forming the multiconnection.
It is necessary to specify a model for one of the subregions.	A **Model** is to be specified at least for one **Subregion**.
There is no correspondence to the rotation of connected boundary condition	This message means that the axis of rotation, which is selected on the **Subregion**, and the axis of rotation of the connected sliding **Boundary condition** are not coincide (their vectors do not lie along the same right line). See details in the section Sector-sliding setting.
The number of sectors is not an integer Possible causes: 1) Axis of rotation is incorrect 2) Sector angle is incorrect	Wrong orientation of an axis of rotation or wrong angle of a sector specified in the sector-sliding setting. See details in the section Sector-sliding setting.
Task structure integrity violation	**Pre-Postprocessor** has read the project's data and found that their structure doesn't meet the program's requirements. Possible reasons are: • the project's file was damaged during its recording, storing, or reading • the project's file has been incorrectly manually edited by a user • the project's file has been saved by a newer version of the program
No registered licenses	The program failed to find a license that is registered for the current license name. This message is output at attempts to obtain a license immediately by **Pre-Postprocessor**. *Possible causes of this error:* On **License Manager** no license has been found at the specified address for the current license name (it is defined in the parameters of the **Solver-Agent**'s user). Possible causes: 1. Incorrect license name has been specified. 2. The license for the license name has been expired. 3. Address of the **License Manager** (IP and port) has been specified incorrectly for this license name, so the program attempts to connect to an another program's **License Manager**. *Recommendations:*

Error messages of Pre-Postprocessor	
Post	**Description**
	• Register the license. To obtain a license, contact your sales manager. • Ensure that **Pre-Postprocessor** connects to those **License Manager**, on which the license has been activated. • Ensure that a correct license name is specified in user's settings of **Solver-Agent**.
`No free licenses`	There are no free licenses; all licenses for this user are busy. To find out the cause of the error, use the command **File > Licenses > Get license info**. *Possible causes of this error:* • **Pre-Postprocessor** 's build date is later than last day of technical support, specified in the license. • the project uses models or functionality that are not provided by the current license • lack of licensing options (for example, there are no enough options for parallel computing) *Recommendations:* • check the validity end date of your technical support; it must be later than the date of your solver's version, otherwise contact your sales manager to renew the license • check the availability licenses for models, which are used in your project • check the number of available options and the number of options allocated for the calculation • When the hyper-threading (*Hyper-Threading Technology, HTT*) is enabled, each run on any logical core will require one parallel option. To prevent wasting license parallel options, you can disable the hyper-threading.
`Cannot release license`	Unsuccessful attempt to release a free license. Contact the technical support service.
`Start of solution failed. License manager is not found.`	See description of errors with code **1111** in the section Solver's error messages and warnings.
`Solve stopped. License manager is not found.`	
`Start of solution failed. Unknown response from license manager.`	See description of errors with code **1112** in the section Solver's error messages and warnings.
`Solve stopped. Unknown response from license manager.`	
`Start of solution failed. License information for this user is absent.`	See description of errors with code **1113** in the section Solver's error messages and warnings.
`Solve stopped. License information for this user is absent.`	
`Start of solution failed. Not enough licenses for this user.`	See description of errors with code **1114** in the section Solver's error messages and warnings.
`Solve stopped. Not enough licenses for this user.`	
`Only one active BL grid may be specified in a project.`	Incorrect specification of the boundary layer grid
`Minimum 3 layers must be specified in a BL grid.`	
`Maximal ratio of the BL grid thickness to a BL cell height is 1.e+5.`	
`A BL grid can be built on boundary template WALL only:`	

Error messages of Pre-Postprocessor	
Post	**Description**
The same boundary condition cannot enter different BL grids:	
If the same boundary condition is assigned to different moving bodies or a moving body and the main geometry, a BL grid cannot be built on these surfaces:	
Incompatible license manager version. Please update this license manager.	Version of the License Manager is outdated and does not meet to the used version **Pre-Postprocessor**. To continue your work, you have to update **License Manager**.
It is necessary to specify properties required for the substance (Substance: ; Property:).	Correct the referred errors in specification of a **Substance**.
If process Turbulence is activated, viscosity of every Substance present in the Phase must be nonzero (Phase: ; Substance:).	
It is necessary to create a binder condition and include a binder in it when using boundary condition of Connected type.	Correct the referred errors in specification of a connected boundary condition.
It is necessary to create a binder for boundary condition of Connected type (Subregion: ; Boundary condition:).	
It is necessary to include a binder in binder condition when using boundary condition of Connected type (Subregion: ; Boundary condition:).	
It is necessary to specify heat transfer calculation model when using a Combustion model.	Correct the referred errors in specification of the **Combustion** mass transfer model.
It is necessary to specify properties required (Fuel, Oxidizer, Product-1) when using a Combustion model.	
It is necessary to specify Product-2 property if i_3 coefficient is not null when using a Combustion model.	
It is necessary to create reaction if elements are specified when using a Chemistry model.	Correct the referred errors in specification of the **Chemistry** mass transfer model.
Number of elements must be less then number of phase's substances when using a Chemistry model.	
It is necessary to specify not null value of one of the coefficients for every element when using a Chemistry model (Element:).	
It is necessary to specify at least one reaction when using an Ablation->Chemistry model.	Correct the referred errors in specification of **Ablation**.
It is necessary to select a Reactant (from the list of Substances) for specifying reaction within Ablation->Chemistry model.	

Error messages of Pre-Postprocessor	
Post	**Description**
`It is necessary to select a Product (from the list of Substances) for specifying reaction within Ablation->Chemistry model.`	
`Reactant and Product must be different Substances in specifying reaction within Ablation->Chemistry model.`	
`It is necessary to specify 'Reactant stoichiometric coeff.' in specifying reaction within Ablation->Chemistry model.`	
`It is necessary to specify 'Product stoichiometric coeff.' in specifying reaction within Ablation->Chemistry model.`	
`It is necessary to select an Object for Limitation in specifying Modifier 'Moving body' in the given Subregion (Subregion: ; Modifier: ).`	Correct the referred errors in specification of **Modifiers**.
`The number of Modifiers, affecting the value of a variable in Subregions with the same Model, must be less then 52`	
`While activating process Heat transfer for a solid Phase, select 'Heat transfer via h' (Phase: ).`	Correct the referred errors in specification of **Physical processes**.
`Selection of model 'Heat transfer via H' assumes that process Motion is activated (Phase: ).`	
`If Cp is specified by polynom, molar mass must be constant`	If **Specific heat** is set by a polynome, the **Molar mass** must be set by a constant. Make the required changes in the project.
`Error: Volume visualization is not supported by graphical subsystem`	This error can appear in the following cases: • your computer doesn't meet to the recommended system requirements to the video card • your driver failed to compile the shaders on the user's computer (this problem can be caused due to the video card driver) It is recommended to use a video card with support of *OpenGL* with version *3.1* or higher and update the video card driver.
`Failed to upload the project files to the solver`	This error message will be displayed in the window of **Pre-Postprocessor** if the path to the server directory specified for the user of **Solver-Agent**, does not exist. This can occur if the specified directory was deleted or renamed or moved to another place. Also you should check availability of the disk and correctness of the parent directory's name.
`In the phase interaction, a pair of substances with a vapor-forming substance for the film is not defined (Continuum-particles).`	When evaporation/sublimation of the film's substance is simulated, you have to specify pairs of **Substances** that correspond to the liquid and vapor states of the film's substances. These pairs of **Substances** are specified as elements of the **Substance pair** array in properties of the **Phase interaction** element for the **Continuum-particles** phase pair, which correspond to the carrier phase (when icing is simulated: the air) and the dispersed phase of particles (when icing is simulated: water droplets).

Error messages of Pre-Postprocessor	
Post	**Description**
	If the film and the particles contain only one **Substance**, you fill only one element in the **Substance pair** array. If the film and the particles contain multiple **Substances** (for example, they contain mix of water and ethanol), you have to fill in several elements of the array. For example, when the film and particles consist of mix of water and ethanol, you specify water and water vapor in one of the elements of the **Substance pair** array, and specify liquid and gaseous ethanol in another element of the array. If the **Substance pair** is not set in such situation, then the "**!**" symbol will appear near the appropriate **Phase interaction** element, and the error message "**In the phase interaction, a pair of substances with a vapor-forming substance for the film is not defined (Continuum-particles).**" will be displayed in the **Log** window.

7.2 Solver and Solver-Agent

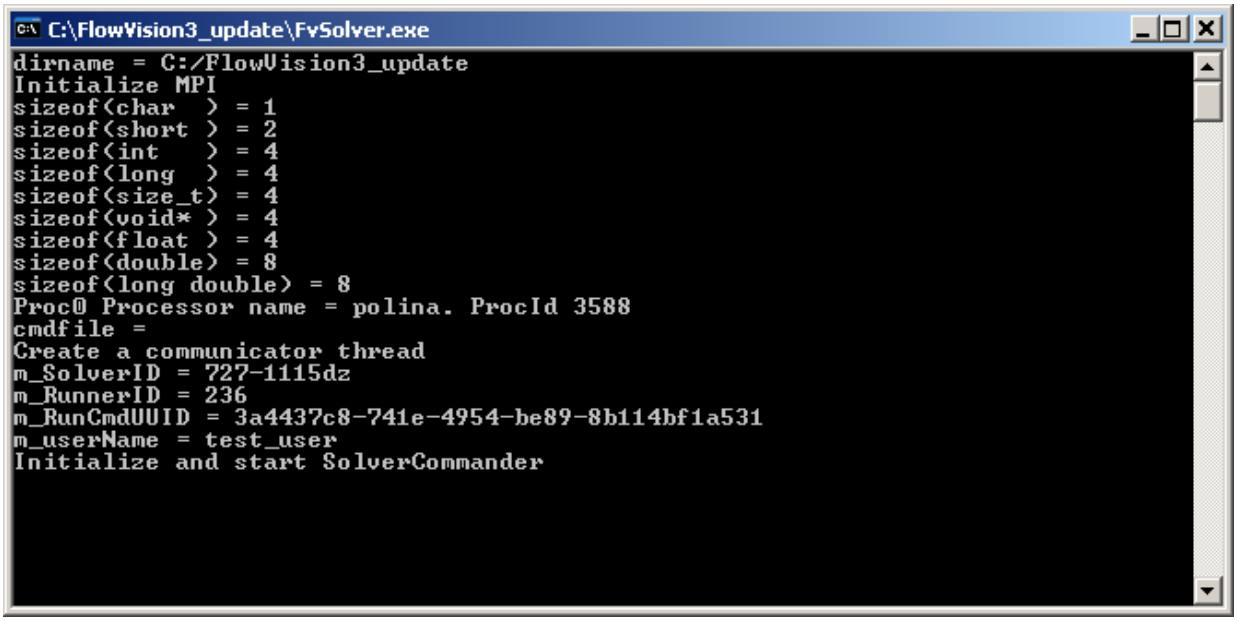
Solver

Solver makes the calculations and transfers results of the calculations to client applications. This module has no graphical user interface.

Solver can be run:

1. from **Pre-Postprocessor** or from **Terminal** via **Solver-Agent**
2. from a command line of the operating system or a user script
3. from a command line in the Batch mode (this is a special case of starting from a command line or a user script)

After starting the **Solver** in *Windows*, you might view the *console window* of **Solver**:



Console window of **Solver**

The console window outputs messages about operation of **Solver**, including error messages.

The console window is displayed in the graphical environment of those user, as which the process was started.

Closing the console window causes termination of the **Solver**'s operation.

Solver-Agent

Solver-Agent is a server application of *FlowVision*, it has no graphical user interface.

Solver-Agent provides communication between client applications and **Solver**, and also it is used for starting **Solver** and transferring files from the server side to the side of client applications.

In *Windows*, **Solver-Agent** is started from **Start menu > Programs > FlowVision N.NN.N**.

Solver Agent loads and unloads **Solvers** and delivers information to client modules about running **Solvers** and existing projects in the user's server directory.

Starting **Solver** from client modules is made by **Solver-Agent**, which runs (in the system where **Solver-Agent** is installed) a special command line (see section Parameters for starting Solver). A list of several possible command line **Solver** configured in configuration file of **Solver-Agent** (`FvSolverAgent.cfg`) manually or using the **Configurator** module.

Prepared projects can be run on the calculation:

- from **Pre-Postprocessor**
- from **Terminal**

Visualization of the obtained data can be done:

- in **Pre-Postprocessor**

- in **Viewer**

At the first start **Solver-Agent**:

1. **Solver-Agent** is to be be configured.
2. At least one user is to be registered on **Solver-Agent**.

Solver-Agent also provides the multiuser mode with a common computing resource by separating the user workspace (see section Registration data (profile) of Solver-Agent's user and their change), where each user can see only his/her own **Solvers** and projects, separately from his/her colleagues.

Automatic start of Solver-Agent at *Windows*' startup

Windows has methods of automatic start of applications at the startup of the operating system, these methods should be known to the system administrator.

System administrator can use any convenient method, but he/she must keep in mind access rights of the started **Solver-Agent**, because it is necessary to provide permissions for reading and writing in directories with **Solver-Agent**'s settings.

The distributive pack of *FlowVision* includes scripts `InstallSAService.bat` and `UninstallSAService.bat` that are intended for installing and uninstalling the system service `FvSAService.exe`, which provides automatic start of **Solver-Agent** as system service at *Windows*' start. Installing and uninstalling the `FvSAService` service is to be run as an Administrator.

Advantages:

- **Solver-Agent** will start even if the user does not login into the system
- **Solver-Agent** cannot be unloaded from the system by a user if the user has no permissions to terminate system processes

Limitations:

- Using these scripts it is possible to install on a computer *only one* system service for starting **Solver-Agent**.
- If several *FlowVision* versions are installed on a computer, then you have to choose some main version, **Solver-Agent** of which will start as a system service. For automatic start of other **Solver-Agents**, you have to use other methods of their startup.

To change an automatic **Solver-Agent**'s startup of one version to another version, you have to run the script `UninstallSAService.bat` (to discontinue startup of the old version) and then to run the script `InstallSAService.bat` from the directory of the required version of **Solver-Agent**, which will start as a system service.

7.2.1 Starting Solver without Solver-Agent

Solver can be started without **Solver-Agent**, using an operating system's command line or a user's script.

This might be required in the cases listed below.

Starting Solver from a command line with automatic connection to Solver-Agent:

In this case you have to specify:

- in the `sa_user` parameter, the user's name at **Solver-Agent**
- in the `sa_ID` parameter, some identifier for the **Solver** (to distinguish different **Solvers**)

Example:

```
"C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n 2 FvSolver64.exe
mpi=libFvMPI_MSMPI_v7_x64.dll sa_user=username sa_ID=111-111111 threads=numa
```

Starting in a system where starting Solver-Agent is not assumed and Solvers are started in the batch mode:

In this case:

- you don't have to specify the `sa_user` parameter (the user's name at **Solver-Agent**) because it is not used
- in the `sa_license` parameter specify the license name
- in the `sa_ID` parameter specify some identifier for the **Solver** (to distinguish different **Solvers**)

See also the section Batch mode.

Example:

```
"C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n 2 FvSolver64.exe
mpi=libFvMPI_MSMPI_v7_x64.dll sa_ID=111-111111 sa_license=licensename threads=numa
cmdfile=cmdfile.txt
```

Starting Solvers in the batch mode when connection to Solver-Agent is assumed:

In this case you have to specify all the three parameters:

- `sa_user`
- `sa_license`
- `sa_ID`

Example:

```
"C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n 2 FvSolver64.exe
mpi=libFvMPI_MSMPI_v7_x64.dll sa_user=username sa_license= licensename sa_ID=111-111111
threads=numa cmdfile=cmdfile.txt
```

7.2.2 Batch mode

The *batch mode* of the **Solver**'s operation assumes that all **Solver**'s activity is controlled not by **Solver-Agent** but a specially composed file containing a sequence of commands. Thus **Solver** can be connected to **Solver-Agent** later, after **Solver**'s startup.

The batch mode allows you to arrange an ongoing serial calculation of several projects at the same **Solver**. In this case, the duration of each is determined by calculating the stopping conditions.

To start **Solver** in the *batch mode* as well as when running through the **Solver-Agent**, use the command line setting in the configuration file **Solver-Agent** (see section Parameters for starting Solver). However, when running in batch mode from the command line, you must specify in the parameter `cmdfile` the path to the batch file.

Batch mode is also used to start the work queues in **Terminal**.

When you run **Solver** in the batch mode, the command line will have the following additional parameters:

Parameter	Description	Example of syntax
`sa_ID`	Identifier of the **Solver**[1], which starts	`sa_ID=111-111111`
`sa_user`	Username of the user of **Solver-Agent** as which it runs	`sa_user=username`
`sa_license`	The license name. This parameter should only be specified for running **Solver** in the batch mode without **Solver-Agent** (see also the section Starting Solver without Solver-Agent).	`sa_license=licensename`
`cmdfile=path/cmdfilename.txt`	Name of the Batch file including the absolute path to it [2]	`cmdfile=path/cmdfilename.txt`

Notes:

[1] When you run **Solver** from any client identifier assigned automatically **Solver** (see *Note* in the section Configuration file of Solver (FvSolver.cfg)). When run from the command line, you must specify the ID manually (giving it a unique value) to differentiate from other **Solvers**.

[2] When you run the calculation on a cluster must be the path accessible to all nodes in the cluster.

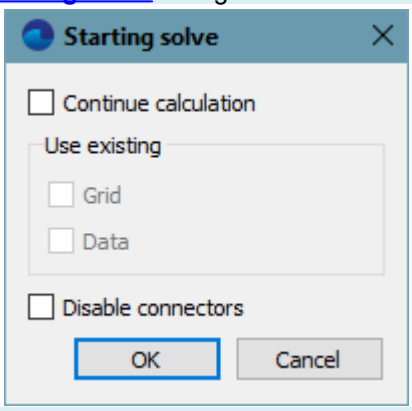
Example of a command line to run Solver in the batch mode without Solver-Agent and subsequent connection to Solver-Agent

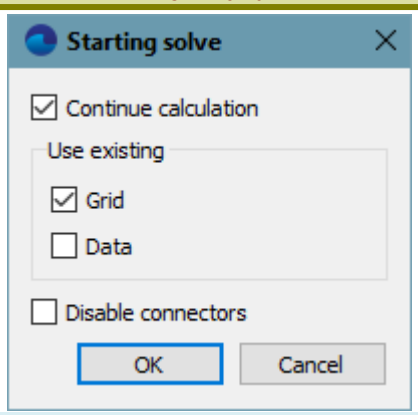
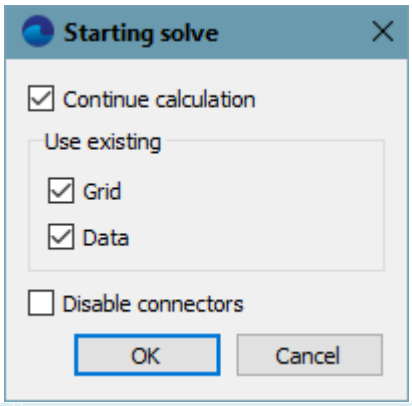
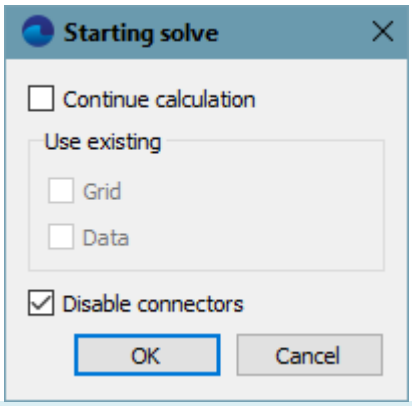
```
"C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n 2 FvSolver64.exe
mpi=libFvMPI_MSMPI_v7_x64.dll sa_user=username sa_license= licensename  sa_ID=111-111111
threads=numa cmdfile=cmdfile.txt
```

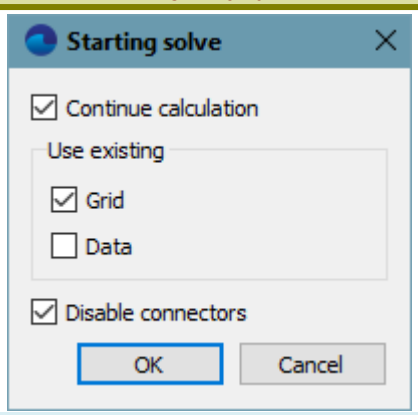
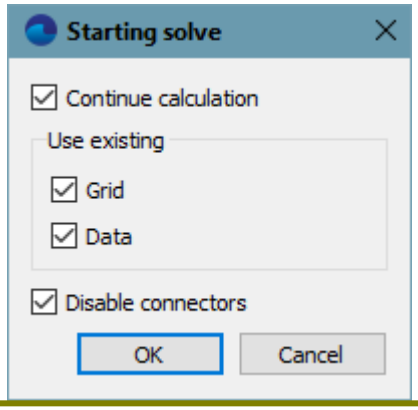
7.2.2.1 Command file

Batch file- a file containing a set of commands for controlling the running solver.

The contents of a batch file

Command	Description
SS_PROJECTLOAD<path to the project file for solver[1]**><ProjectID><VersionID><>** **ProjectID** - the unique identifier of the project (can be skipped) **VersionID** - a unique version identifier (can be skipped)	Load the project on the solver
SS_APPLYNEWGEOMETRY<OBJECT[identifier**]><**path to a file with the geometry**>**	Replacing the geometry of an imported object. The identifier in the parameter **OBJECT** corresponds to the value of the parameter **myid** in the line from the **`fvinp`** file corresponding to the selected imported object. Example of this lile: **`UIName="Imported object #0" class="CMeshEntity" myid="10"`** See an example of use of the **SS_APPLYNEWGEOMETRY** command in the section Automatic replacement of the geometry during the computation.
SS_CUSTOM_COMMAND<TORT_EXPORT_2.0><><><><>	Export results of *FlowVision*'s computation to the TORT software package.
SS_SOLVESTART<startup parameter for the calculation**>** Startup parameter for the calculation: • **0** or **2** (both these values give the same effect) – starting the computation from scratch. These values correspond to the situation when all checkboxes in the **Starting solve** dialog box are not selected:  • **3** – starting the computation from scratch, but do *not* rebuild the existing computational grid. This corresponds to the situation when in the **Starting solve** dialog box checkboxes **Continue calculation** and ***(Use existing)* Grid** are selected and checkboxes ***(Use existing)* Data** and **Disable connectors** are not selected:	Running the solver for calculation according to the specified settings.

Command	Description
 • **7** – continue the computation. This corresponds to the situation when in the Starting solve dialog box the **Continue calculation**, ***(Use existing)*** **Grid**, and ***(Use existing)*** **Data** checkboxes are selected and the **Disable connectors** checkbox is not selected:  • **8** or **10** – starting the computation from scratch with disabling connectors. This corresponds to the situation when in the Starting solve dialog box the **Continue calculation**, ***(Use existing)*** **Grid**, and ***(Use existing)*** **Data** checkboxes are not selected and the **Disable connectors** checkbox is not selected:  • **11** – starting the computation from scratch with disabling connectors and do not rebuild the existing computational grid. This corresponds to the situation when in the Starting solve dialog box the **Continue calculation** and ***(Use existing)*** **Grid** checkboxes are selected, the ***(Use existing)*** **Data** checkbox is not selected, and the **Disable connectors** checkbox is selected:	

Command	Description
 • **15** – continue the computation with disabling connectors. This corresponds to the situation when all checkboxes (**Continue calculation**, ***(Use existing)* Grid**, ***(Use existing)* Data**, **Disable connectors**) in the Starting solve dialog box are selected: 	
SS_SOLVESTART_EX<computation startup parameter**><**type of stop**><**time or number of steps**>** Startup option for computation, this is similar to **SS_SOLVESTART** Type of stop: • **AT** - stop at a specified time • **AI** - stop after a specified step • **RT** - stop after a specified time after the start of the calculation • **RI** - stop after a specified number of steps after the beginning of the calculation The time or number of steps is time or number of steps used for stopping.	Running the solver and stopping the computation at the specified time.
SS_DO_NOT_DISTURB	Do not process other requests until the program is counted to the end (makes sense only after the command **SS_SOLVESTART**)
SS_SOLVESTOP	Stop through solver
SS_PRJDATASAVE	Save the results of the calculation of the project
SS_PROJECTUNLOAD	Unload project solver
SS_SHUTDOWNSOLVER	Unload solver

Note:

[1)] The path can be either absolute or relative. A relative path is specified relatively to the location of the command file, and in both *Windows* and *Linux* the relative path is started from the symbol "." and the delimiter between directory names is the symbol "/". Example: `./projectname.fvproj`.

Sample of a batch file

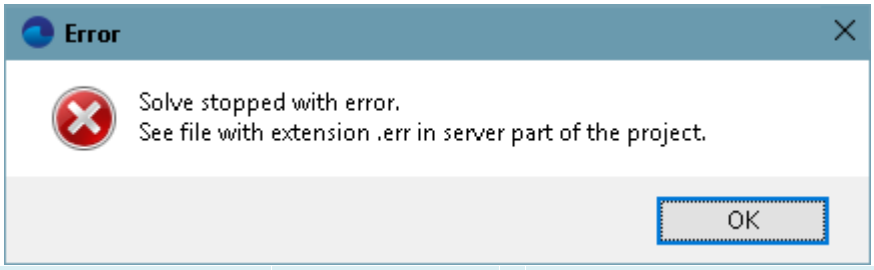
```
SS_PROJECTLOAD</home/user/fvproj1/proj1.fvproj><><><>
SS_SOLVESTART<0>
SS_DO_NOT_DISTURB
SS_PRJDATASAVE
SS_PROJECTUNLOAD
SS_PROJECTLOAD</home/user/fvproj2/proj2.fvproj><><><>
SS_SOLVESTART<7>
SS_DO_NOT_DISTURB
SS_PRJDATASAVE
SS_SHUTDOWNSOLVER
```

7.2.3 Solver's error messages and warnings

Solver's error messages*) are sent over the network to other *FlowVision* modules, which, if necessary, can display them to the user.

The **Terminal**'s messages displayed error codes (either with or without explanations), while **Pre-Postprocessor** displays texts of the messages without their error codes.

Some errors can cause displaying additional information in the console window of **Solver** and recording the information into errors files (`.err` files, see section Project files); this information has its own codes and text messages.



An example of an error message in **Pre-Postprocessor** ("**Solve stopped with error. See file with extension.err in server part of the project.**")

Some reports may include the computational cell's number to give information about the cell where the failure occurred.

Note:

*) For the sake of brevity, this term is applies to errors, warnings and info messages, unless explicitly stated otherwise.

Codes and text messages of Solver's errors and warnings

Errors (they are transferred to other *FlowVision* modules through the network and are not displayed in the Solver's console window and usually are not recorded into .err files)

Code	Text of the message ***P:*** **inPre-Postprocessor** ***T:*** **in Terminal** **err: in an `err`-file in the server part of the project**	Description
100	***P:*** `Solution is not in progress`	An attempt to perform an operation that allowed only during the calculation.
101	***P:*** `Solution is in progress`	An attempt to perform an operation that is prohibited during the calculation.
102	***P:*** `No project loaded`	An attempt to perform an operation on the loaded project in its absence.

Code	Text of the message *P:* inPre-Postprocessor *T:* in Terminal err: in an `err`-file in the server part of the project	Description
103	*P:* `Project is already loaded`	Attempting to download the project when it was already loaded.
104	*P:* `Project mismatch`	Non-compliance projects. This error is handled by **Pre-Postprocessor** when its synchronizing with **Solver**.
105	*P:* `Input data version mismatch`	Version mismatch of the input data. This error is handled by **Pre-Postprocessor** when its synchronizing with **Solver**.
106	*P:* `The solver is unable to load this project.` *T:* `Can not load project` err: *the record depends on which file is missing*	Unexpected situation during the boot project **solvers**.
107	*P:* `Path does not exist or is inaccessible`	Invalid or not available for solver file system path.
108	*P:* `I/O Error`	Uncertain input or output error. *Possible reasons are:* • the client part of the project has been damaged (damage to individual files, no individual files) • folder or individual files the client part of the project is not writable or locked by another application *Recommendations:* • Do not place or open projects in the system directory, such as `C:/Windows` or `C:/Program Files`. For example problems from the textbook is recommended before you start working with them to copy the directory to which there is full access from the current user's operating system.
109	*P:* `Invalid format`	Wrong format of data transferred to **Solver**.
110	*P:* `Invalid parameters`	Invalid parameters of the operation.
111	*P:* `Invalid operation`	Invalid operation in this situation.
112	*P:* `On the solver has unsaved data`	Operation is not allowed to save on solver counted data.
113	*P:* `Scene version mismatch`	Mismatch between the version of the scene. This error is handled by **Pre-Postprocessor** when its synchronizing with **Solver**.
114	*P:* `Simulation controls version mismatch`	Version mismatch calculation parameters. This error is handled by **Pre-Postprocessor** when its synchronizing with **Solver**.
115	*P:* `Bad geometry`	The geometry contains topological errors.
116	*P:* `Bad geometry coloring`	When replacing the geometry can not be set to-one correspondence of the boundary conditions on the coloring.

Code	Text of the message *P:* inPre-Postprocessor *T:* in Terminal *err:* in an `err`-file in the server part of the project	Description
117	*P:* `Can not find suitable import library`	You can not import existing geometry libraries imports.
118	*P:* `The solver was unable to load grid with data`	Failed to load the grid with data on the solver - files are missing or damaged.
119	*P:* `One of necessary files was not found. (The solver is unable to load this project.)` *T:* `Error loading project` *err:* *the record depends on which file is missing*	File not found or is not available **Solver**. *Possible reasons are:* • one or more files download project available *Recommendations:* • restore the project from backup
120	*P:* `Error loading library connector`	Failed to load the library connector, for example `MpmConnector.dll`.
121	*P:* `Solver does not support this feature.`	Request may not support the current version of **Solver**.
122	*P:* `Error during processing of the extended command solver`	Error executing query `SS_CUSTOM_COMMAND`.
123	*P:* `Start of solution failed. Some necessary parameters are not defined.`	Run solver for the calculation failed. Some of the required parameters are not set.
124	*P:* `Solver has no data to save`	The **Solver** has no data to be saved. This text is output at attempt to save the server part of a project (when you click the  button (**Save solution on the solver**) in the **Standard** toolbar).
	P: `Cannot continue solve: the server part does not contain results of the previous calculation (start the calculation with no "Continue calculation" checked) or files with calculation results are damaged (try to restore the data from *.backup files).` *T:* `Solve starting error (124)` *err:* `[Error]: [proc=0, code=0x80010002] No data, grid files are absent`	The **Solver** has no data to continue the calculation. This text is output at unsuccessful attempt to continue the calculation due to absent or damaged data.
126	*P:* `Input data has been modified in the way that it is impossible to continue existing computation. Please restart.`	Input data is changed so that the continuation of calculation is impossible.
136	*P:* `The solver is unable to load this project because the Decimate.cmd or Decimate.err files exist in server part of the project.`	Errors due to decimation of the non-steady-state record of the project. **Solver** cannot load the selected project because of files `Decimate.cmd` or `Decimate.err` exist in the server part of the project. The error code is ERR_CANNOT_LOAD_PROJECT_DECIMATE = 136
1111	*Messages that are displayed at starting of the project:* *P:* `Start of solution failed. License manager is not found.`	**Solver** failed to connect to the license server (**License Manager**). *Possible causes of this error:* • **License Manager** has not been started

Code	Text of the message ***P:*** **inPre-Postprocessor** ***T:*** **in Terminal** **err: in an `err`-file in the server part of the project**	Description
	Messages that are displayed during the project's running: ***P:*** `Solve stopped. License manager is not found.`	• Invalid port or host of **License Manager** is specified in settings of **Solver** • no connection to the computer, on which **License Manager** is running • the connection to **License Manager** is blocked by a firewall *Recommendations:* • check whether **License Manager** is running • check the settings of **Solver** (host and port for connection to **License Manager**)
1112	*Messages that are displayed at starting of the project:* ***P:*** `Start of solution failed. Unknown response from license manager.`	Wrong answer from **License Manager** (violation of the communication protocol). Contact the technical support service.
	P: `Solve stopped. Unknown response from license manager.`	
1113	*Messages that are displayed at starting of the project:* ***P:*** `Start of solution failed. License information for this user is absent.`	There is no license for this user. *Possible causes of this error:* On **License Manager** no license has been found at the specified address for the current license name (it is defined either in the parameters of the **Solver-Agent**'s user or in the parameter of the **Solver**'s run). Possible causes: 1. Incorrect license name has been specified. 2. The license for the license name has been expired. 3. Address of the **License Manager** (IP and port) has been specified incorrectly for this license name, so the program attempts to connect to an another program's **License Manager**. *Recommendations:* • Register the license. To obtain a license, contact your sales manager. • Ensure that **Solver** connects to those **License Manager**, on which the license has been activated. • Ensure that a correct license name is specified in user's settings of **Solver-Agent**.
	Messages that are displayed during the project's running: ***P:*** `Solve stopped. License information for this user is absent.`	

Code	Text of the message ***P:*** **inPre-Postprocessor** ***T:*** **in Terminal** **err: in an `err`-file in the server part of the project**	Description
1114	*Messages that are displayed at starting of the project:* ***P:*** `Start of solution failed. Not enough licenses for this user.`	There are no free licenses; all licenses for this user are busy. To find out the cause of the error, use in **Pre-Postprocessor** the command **File > Licenses > Get license info** or in **Terminal** the command **Licenses > Get license info**. *Possible causes of this error:* • **Solver**'s build date is later than last day of technical support, specified in the license. • the project uses models or functionality that are not provided by the current license • lack of licensing options (for example, there are no enough options for parallel computing) *Recommendations:* • check the validity end date of your technical support; it must be later than the date of your solver's version, otherwise contact your sales manager to renew the license • check the availability of licenses for models, which are used in your project • check the number of available options and the number of options allocated for the calculation • When the hyper-threading (*Hyper-Threading Technology, HTT*) is enabled, each run on any logical core will require one parallel option. To prevent wasting license parallel options, you can disable the hyper-threading.
	Messages that are displayed during the project's running: ***P:*** `Solve stopped. Not enough licenses for this user.`	
1115	***P:*** **Can not release license**	Unsuccessful attempt to release a free license. Contact the technical support service.
1116	***P:*** **The number of cells exceeds the upper limit for the given license**	The number of computational cells exceeds the limit for this license.
65533	***P:*** **Unknown solver response**	Unknown response solver (violation of the communication protocol).
65534	***P:*** **Invalid query parameter**	Invalid request parameter (communication protocol violation).

65535	***P:*** a message depending on the situation or "`Unknown error`"
	err:
	`[Error]: [Code = 0x80090002] Error in matrix solver: Explicit time step limit reached: XXXXX`
	`[Error]: An error (0x0000FFFF) during solution step, stopping solution.`
		Unexpected situation, fall prevention program.
		Possible cause: The current time step is less than the predetermined minimum limiter in the time step, indicating that the divergence of equations - the breakup of the solution.
		Recommendations:

Code	Text of the message *P:* inPre-Postprocessor *T:* in Terminal err: in an `err`-file in the server part of the project	Description
		• check the correctness of the specified minimum time step (the default value is **1e-20**) • analyze the solution, check the input data, if necessary, grind computational grid, to reduce the time step

Warnings displayed in the Solver's console window and written into the `.err` file

Code	Text of the message	Description
40090000	`Could not concentrate VOF`	Unable to adjust VoF (see section Theory> Physical processes> Phase transfer). In such a way that conservative.
40090001	`Near fluid cell is gas cell`	During the reconstruction of the free surface there is an error in which the liquid near the surface was the gas cell.
40090002	`Undefined cell, making it as Gas cell`	During the reconstruction of the free surface there is an error in which cell type is not specified, the cell becomes a gas cell.
40090003	`Fluid cell, no sides`	During the reconstruction of the free surface there is an error in which the liquid from the surface of the cell, or did not have borders. This cell will be defined as a gas.

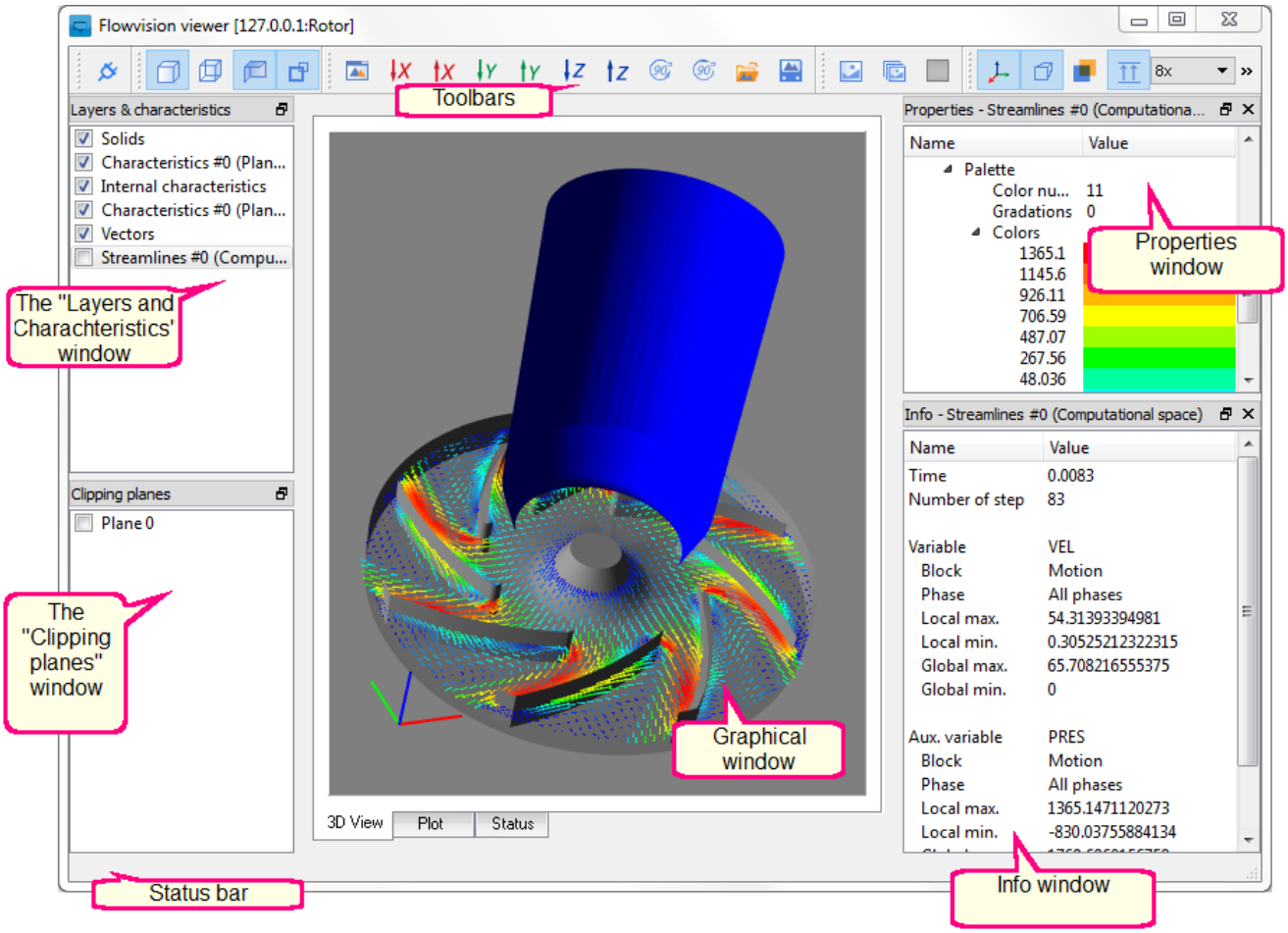
Errors displayed in the Solver's console window and written into the `.err` file

Code	Text of the message	Description
80090000	`Could not build free surface`	Failure in constructing the free surface. Most likely an error occurred while generating the computational grid.
80090001	`No reconstruction for this cell`	Most likely there was an error in determining the cell type (liquid, gas, surface), can not be built in the free surface of the cell
80090002	`Error in matrix solver:`	Error in the algebraic solver, the specific error is reported in the same line after this header.
80010003	`No FlowVision geometry is linked to Abaqus region "ABAQUS_REGION_NAME"!`	*Abaqus* gave a region name, which doesn't matches to any exchange surface (connection region) in the *FlowVision*'s project.
80090004	`Error in Check Cell: Cp < 8.31441 / Molar mass ||||| Check Cp and Molar mass`	An incorrect combination of values of **Specific heat** (Cp) and **Molar mass** has been specified. This causes termination of the **Solver**. *Recommendations:* Check and correct values of **Specific heat** and/or **Molar mass**.
80090004	`Heat transfer is not activated`	Inappropriate parameters of physical processes on a **Supersonic inlet**.
80090004	`Supersonic problems must be solved with 'Heat transfer via H'`	
80090004	`Absolute static temperature is negative. Probably, the specified total temperature is too low.`	
80090004	`Velocity modulus is less than sonic speed`	

Code	Text of the message	Description
80090005	`Error in Check Grid: Read BL-grid with old version. BL-grid will be destroyed due to different proc number`	This message outputs when the program makes an attempt to open a project, which has been created in an old version of *FlowVision*, if number of processors changed. In this situation the boundary layer grid (BL grid) rebuilds with loss of data.
8009000b	`Dispersed Solver: Algebraic case CALC_FITA_D: Relative volume of the Dispersed phase (particles) in the given cell exceeds 0.99. As a consequence, porosity < 0.01 => There is no room for the Continuous phase.`	The relative volume of the **Dispersed Phase** (particles) in a cell exceeded the value 0.99. So the value of porosity becomes less then 0.01 and there is no room for the flow of the **Continuous Phase**. This problem arises when the number of particles in the volume is too big. This causes termination of the **Solver**.

7.3 Viewer

The **Viewer** module is used to visualize results of computations according to specifications of **Layers**, which have been set in the project.



Viewer's window contains the following user interface elements:

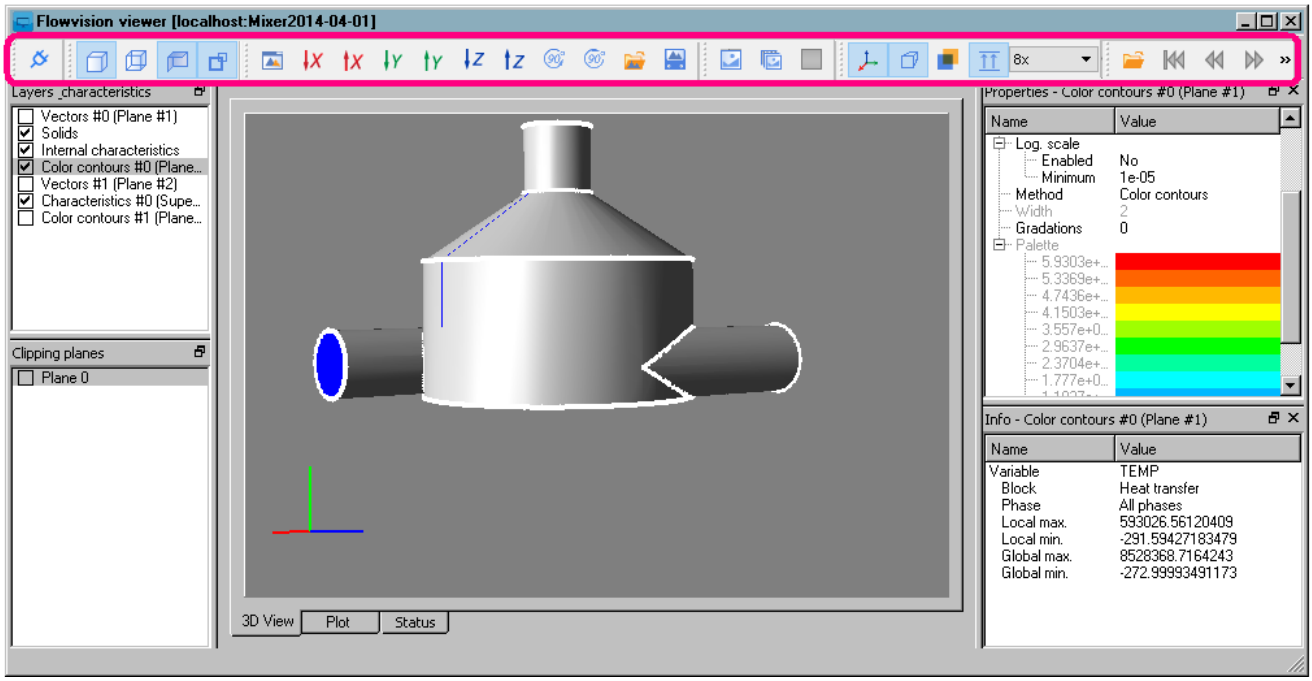
1. **Toolbars**
2. **Status bar**
3. **Graphical window**
4. **Layers and characteristics window**
5. **Clipping planes window**
6. **Properties window**
7. **Info window**

The key features of **Viewer** are:

1. Ability to connect to a running **Solver** and visualize the current results of the simulation. (During the calculation, **Solver** processes the visualization data and transfers them to **Pre-Postprocessor** and **Viewer** at the end of each time step. Therefore the user can view the visualization not immediately, but only after the time step finishes.)
2. Visualization of the current results through layers and stop conditions, predefined in the project.
3. Recording of the current image to a graphic file, or a series of graphic files (recording to a series of files can be used to create an animation).
4. Loading and visualization of files that were previously recorded by **Solver** (these files has extensions **`fvvis`**).

Viewer might not display some of visualization **Layers**. Their visualization will be enabled as new versions of the program are released.

7.3.1 Toolbar of Viewer



Viewer contains the following toolbars:

1. **Connection**
2. **Solids**
3. **Views**
4. **Capture**
5. **Rendering**
6. **Navigation**

(See their descriptions in subsections below)

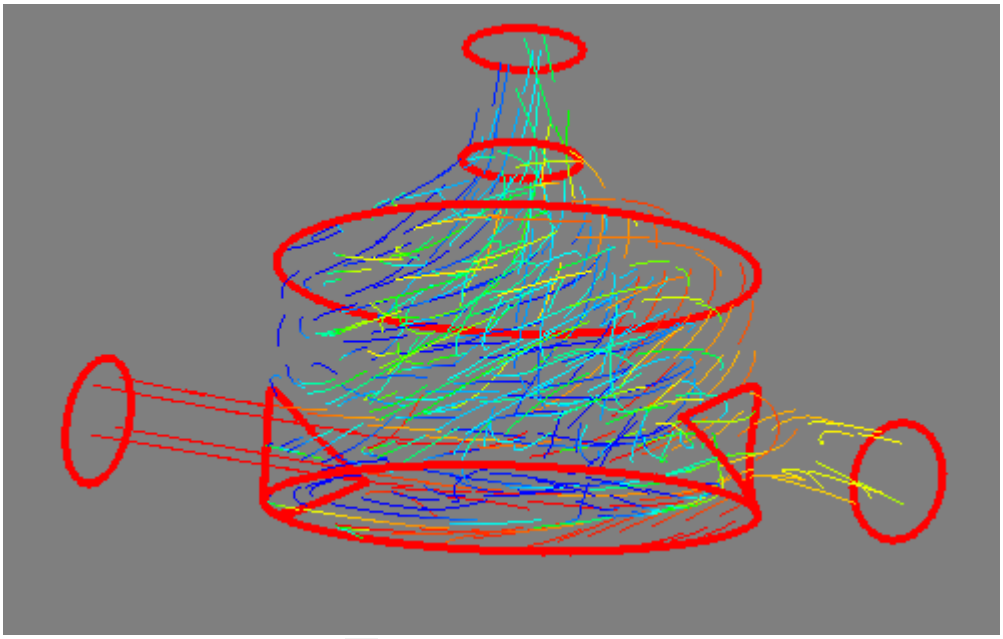
Toolbar "Connection"

Button	Name	Description
	Connect to solver	Connection to **Solver** and the result of the calculation of the project, loaded or calculated on this**Solver**

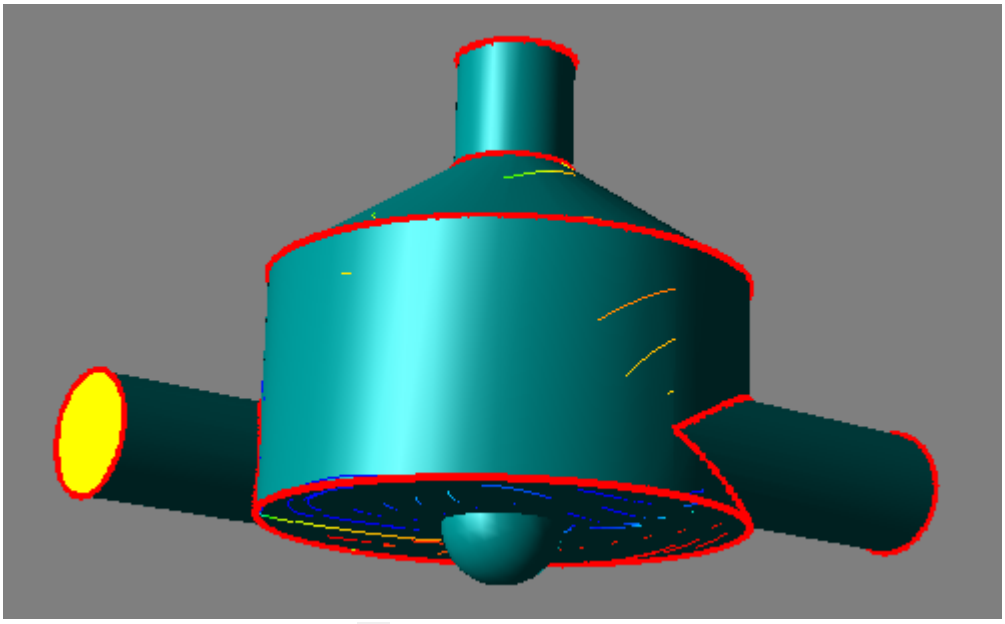
Toolbar "Solids"

Button	Name	Description
	Enable/disable surface fill	Displaying surface groups facets geometry model (i.e., to paint these surfaces, rather than leaving transparent).
	Enable/disable the display of facet edges	Displaying edge facet geometry
	Show/hide surface facing viewer	Hiding surfaces of the geometry, which are oriented to the observer by their non-computational sides[1)]
	Enable/disable display of face group borders	Displaying border groups geometry

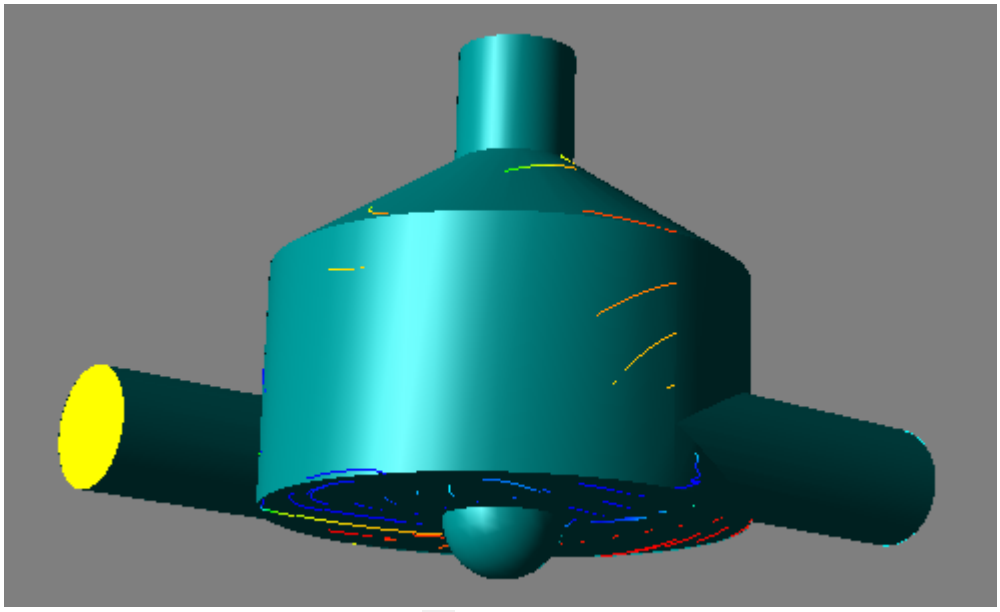
See examples of these buttons in the illustrations below (in all the examples, except the last one is displayed as a layer **Streamlines** a mixer built in speed and with additional coloring temperature).



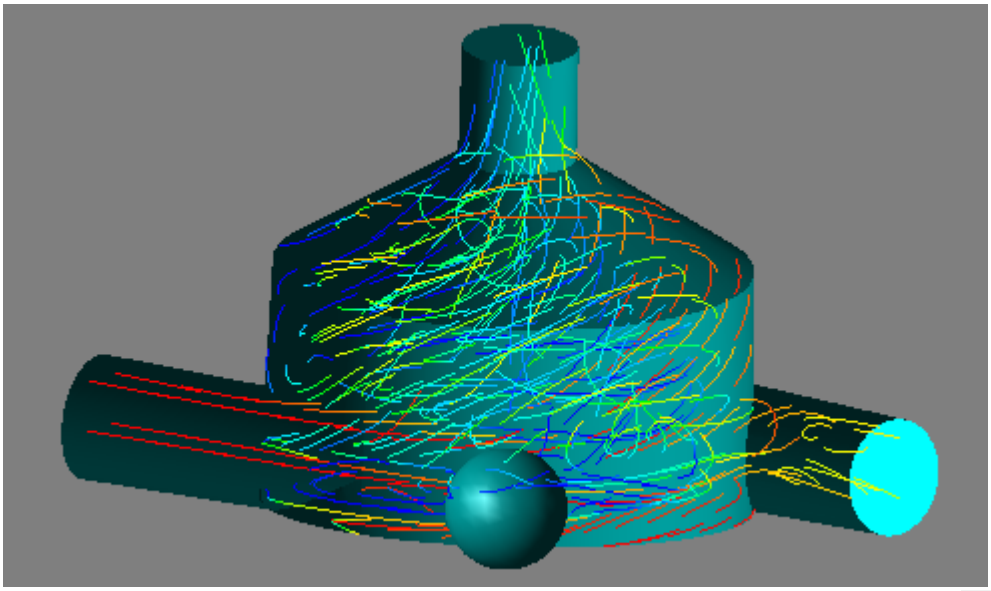
Borders of facet groups are visible (button is pressed), surfaces of facet groups are transparent (button is released)



Borders of facet groups are visible (button is pressed), surfaces of facet groups are filled (button is also pressed)

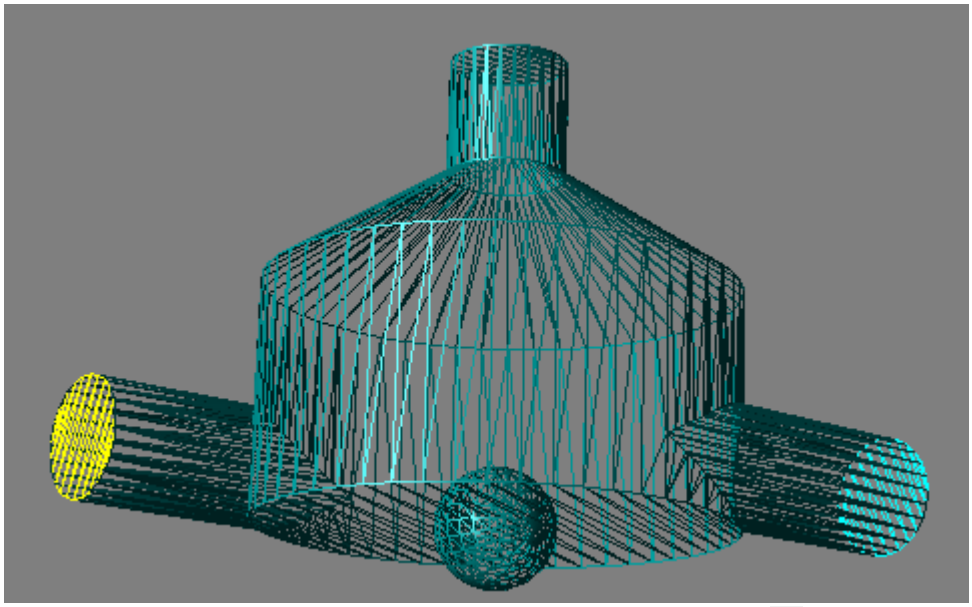


Borders of facet groups are not displayed (button is released), surfaces of facet groups are filled (button is pressed)



Surface of facet groups are filled filled, except external ones aimed at the observer (the buttons and are pressed).

Borders of facet groups are not displayed (the button is released).



Wireframe (edges of individual facets) is visible when the button is pressed.

Toolbar "Views"

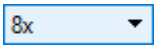
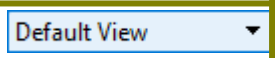
Button	Name	Description
	Change background color	Opens a dialog box for selecting a new background color. The background color and the button's color will change according to your selection.
	View against axis X	Orient the view in the graphical window opposite to the axis X.
	View along axis X	Orient the view in the graphical window along the axis X.
	View against axis Y	Orient the view in the graphical window opposite to the axis Y.
	View along axis Y	Orient the view in the graphical window along the axis Y.
	View against axis Z	Orient the view in the graphical window opposite to the axis Z.
	View along axis Z	Orient the view in the graphical window along the axis Z.
	Rotate the graphics window view 90 degrees counterclockwise	Rotating the view in the graphical window 90 degrees either counterclockwise or clockwise
	Rotate the graphics window view 90 degrees clockwise	
	Load view	Load the view from a **fvcam** file. This opens a standard operating system's window for access to files in which you want to specify a file that will be loaded form (saved earlier using the button).
	Save view	Save the view in a **fvcam** file. This opens a standard operating system's window for access to files in which you want to specify a file in which to save the view.

Toolbar "Capture"

Button	Name	Description
	**Save the contents of the graphics window to an image file**	Save image from the **Viewer**'s graphics window to a graphic file.
	**Start capturing an image sequence**	Start sequentially saving images from the graphics window into files.
	**Finish capturing an image sequence**	Stop sequentially saving from the graphics window into files. *)

*) See description of the application of section Creating animation in Viewer.

Toolbar "Rendering"

Button	Name	Description
	**Toggle coordinate system**	Enable/disable displaying the coordinate system
	**Toggle perspective**	Enable/disable the perspective projection
	**Enable/disable transparency support**	When the button is released, the image is created with no taking into account the transparency. When the button is pressed, the image is created with taking into account the transparency.
	**Toggle layers offset**	Enable/disable displacement of the layers
	**Selector of smoothing modes**	Specifying the mode of smoothing the pixels. Elimination of aliasing is performed using the technology of multiple selection: for each pixel is calculated is not a single value but several probes distributed over its area, which are then averaged. This result and a list of the available modes depend on the hardware and software graphical subsystem of a computer that is running the **Viewer**. Work modes are displayed as • **Nx** • *or* **Nx/Mx** where N and M denote the number of different samples per pixel. Than they are, the greater the smoothing, but lower productivity and higher demands on graphics memory.
	**Selector of the View**	This is a drop-down list, where you can select the **View**, which will be used for displaying the image in the **3D View** tab of the **Viewer**'s graphical window and for applying other settings saved in the **View**.

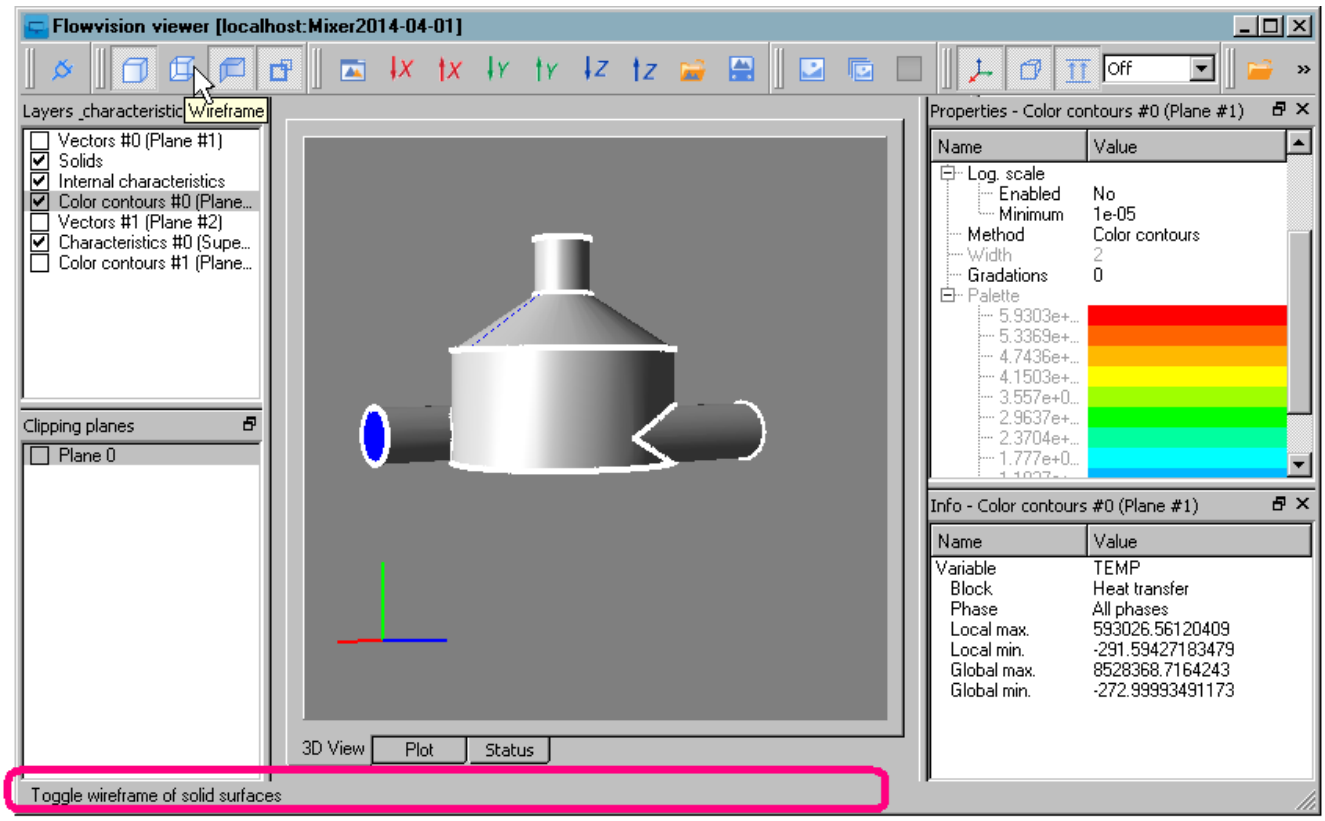
Toolbar "Navigation"

Button	Name	Description
	**Select directory for visualization**	Load the visualization data from a specified folder.
	**Load first step**	Load the first stored visualization data
	**Load previous step**	Load previous visualization data
	**Load next step**	Load the next visualization data

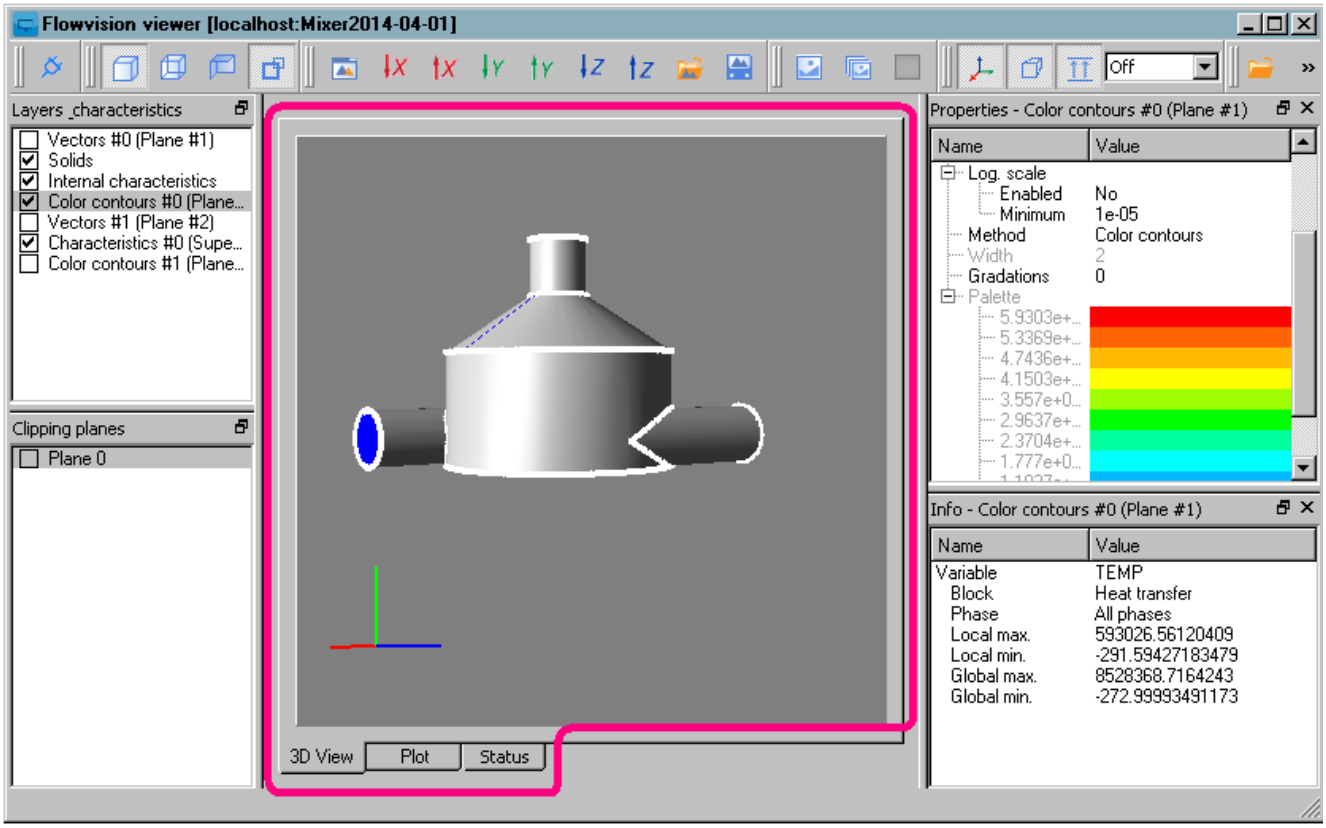
Button	Name	Description
	Load last step	Load the last visualization data
	Selecting the calculation history step to download	Load some arbitrary visualization data from a specified step
	Start automatic sequential transition through the steps of the calculation history	Start an automatic displaying the saved visualization data, with transition from the current data to the last recorded data. The data can also be displayed in an infinite loop. When the directory for visualization contains several **`fvvis`**-files, their contents will be displayed sequentially. To display the data in a loop (nonstop), run Viewer from a command line or in the bath mode with the **loop_playback** key.
	Stop automatic sequential transition through the steps of the calculation history	Stop automatic displaying the data.

7.3.2 Status bar of Viewer

Status bar of **Viewer** displays a short description of the button in the toolbar, over which the mouse cursor locates.



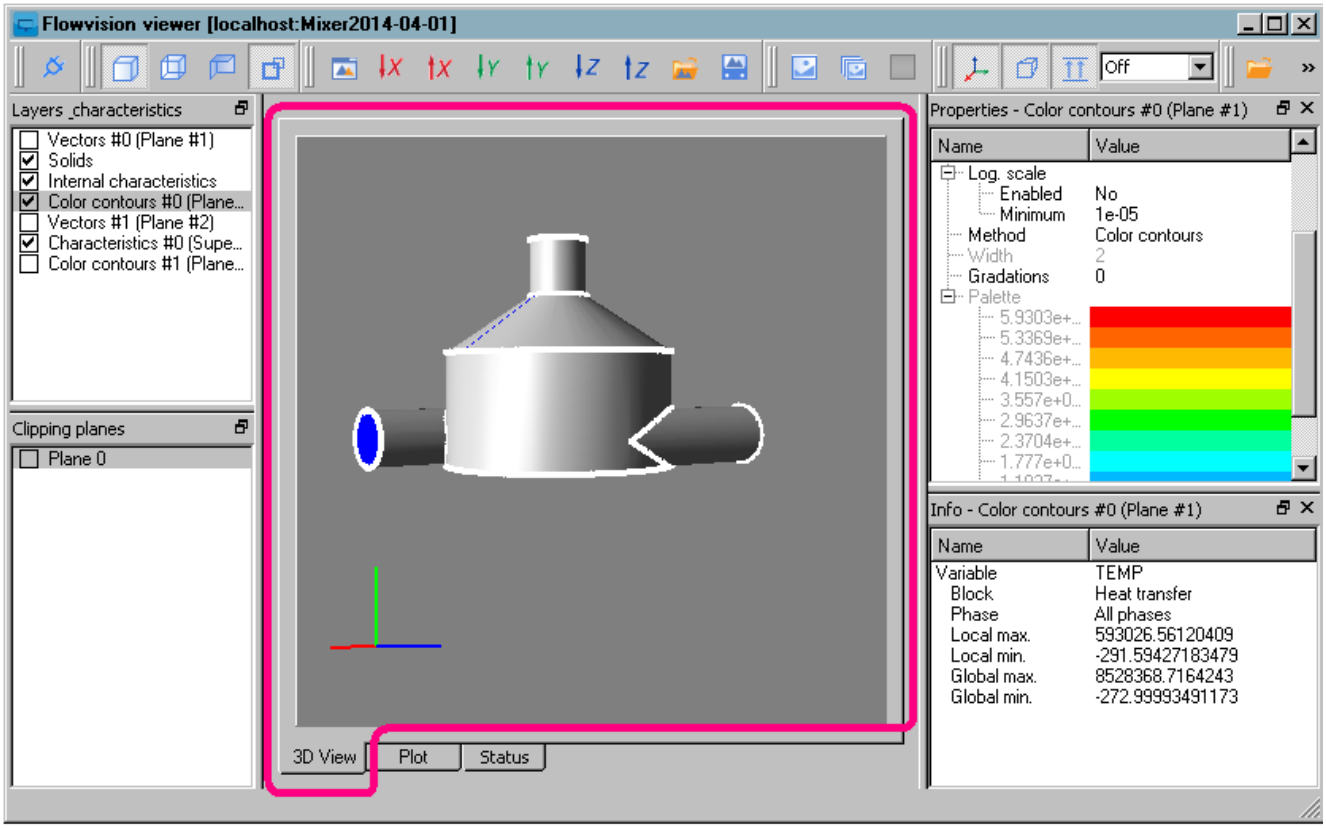
7.3.3 The graphical window of Viewer



The **Viewer**'s *graphical* window contains the following tabs:

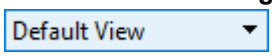
1. **3D View**
2. **Plot**
3. **Status**

7.3.3.1 The "3D View" tab

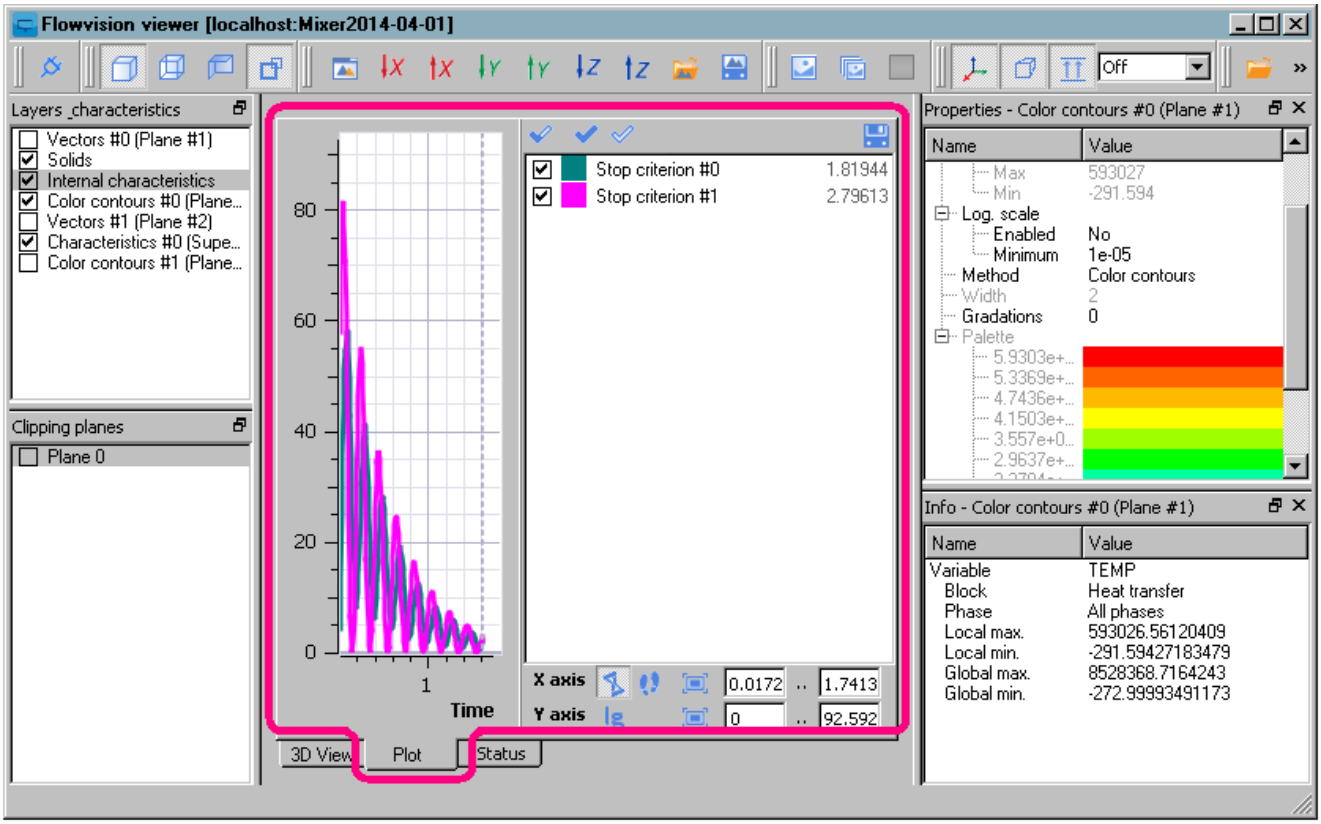


The **3D View** tab is used to display graphical data from **Layers**. In the **3D View** tab you can rotate, move, zoom in the scene as well as you do these in **Pre-Postprocessor** in the changing sight angle and scene orientation mode.

Initially the image is formed accordingly to the **View**, which is currently used in **Pre-Postprocessor**. If you wish to use another **View**, select it from the drop-down list in the **Rendering** toolbar:



7.3.3.2 The "Plot" tab

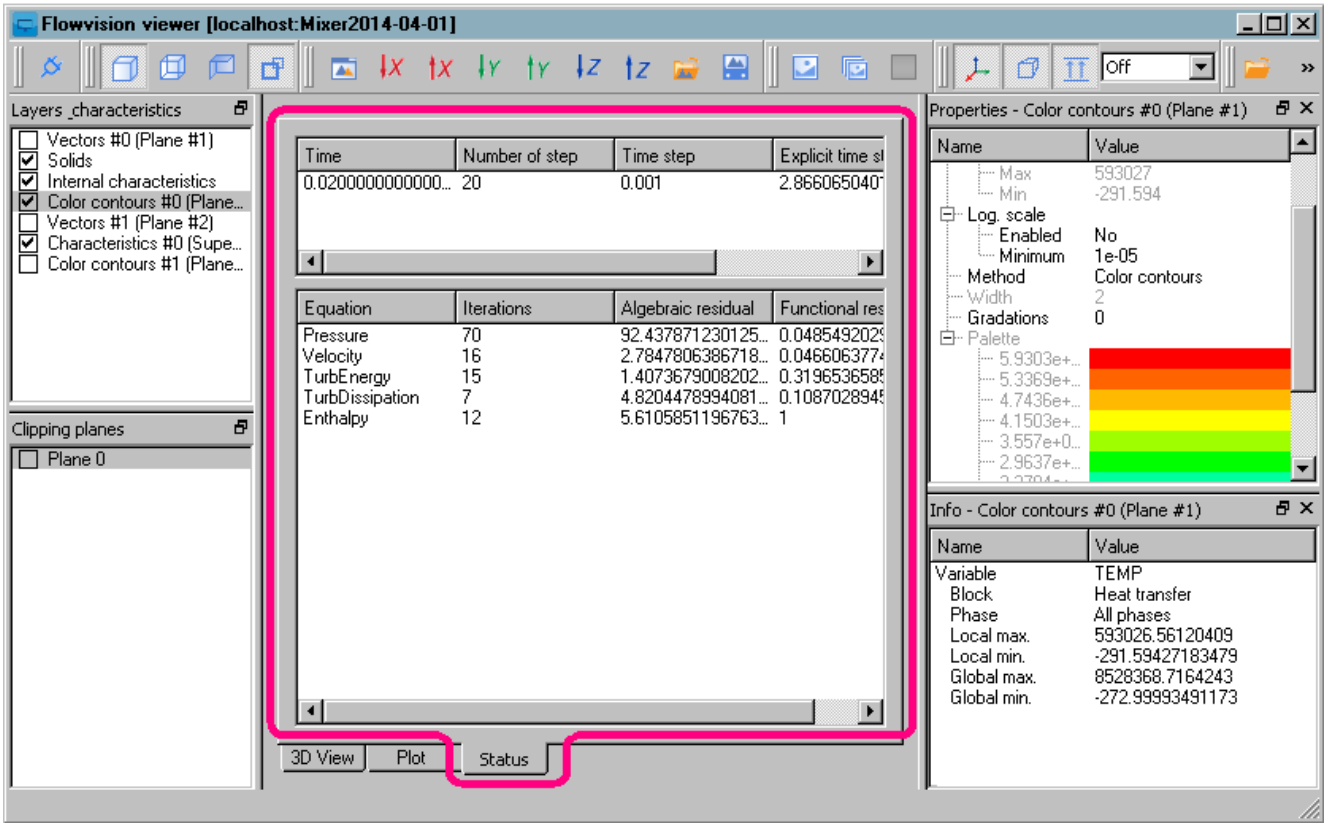


The **Plot** tab displays the plot of functional residuals and user values (characteristics), for which **Stop criteria** have been defined in **Pre-Postprocessor**.

The plot can be adjusted similarly as the plot in the **Monitor** window in **Pre-Postprocessor** (you can shift the plot, scale it, select either logarithmic or uniform scale at the axis of ordinates, view the data depending on either time or number of steps, etc.).

When name of the plot line or value is selected, pressing **Ctrl-C** keys cause copying the name or value into the *Windows* clipboard.

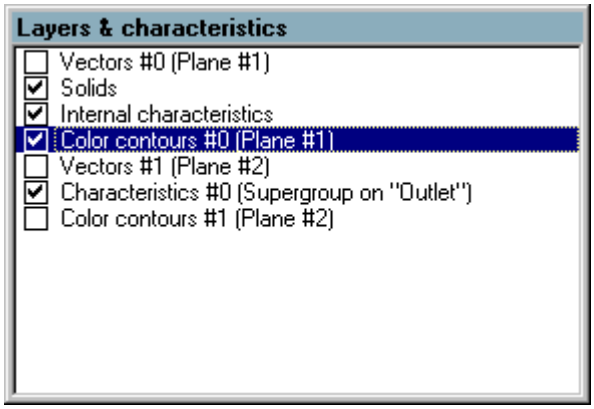
7.3.3.3 The "Status" tab



The **Status** tab is similar to the **Status** tab of the **Monitor** window in **Pre-Postprocessor**.

7.3.4 Viewer's "Layers & characteristics" window

The **Layers & characteristics** window contains list of **Layers** and **Characteristics**, created in the project.



7.3.4.1 Layers in Viewer's "Layers & characteristics" window

The **Layers** are created in **Pre-Postprocessor** when the project is prepared.

Viewer can display the following **Layers**:

- **Solids**
- **Computational grid section**
- **Color contours** (if a variable is selected in the layer)
- **Vectors** (if a variable is selected in the layer)
- **Plot along line**
- **Plot along curve**
- **Plot along ellipse**
- **Isosurface** (if a variable is selected in the layer)
- **Streamlines** (if a variable is selected in the layer)
- **Nodal loadings**
- **VOF**
- **Volume visualization**

- **Screen text layer**, which displays the **Title** in the **View** window in **Pre-Postprocessor** (the title is specified in the **Postprocessor** tab by parameters **Title > ...** in properties of the **3D-scene** root folder).

In the list of **Layers**, each line contains a checkbox. When this checkbox is checked, the appropriate **Layer** is displayed in the **Graphical window** in the **3D View** tab. When the checkbox is unchecked, the **Layer** is not displayed.

When you select a **Layer**'s line in the list, the **Viewer's Properties window** for this **Layer** opens. The **Viewer**'s **Properties window** contains a reduced set of parameters compared to the **Properties window in Postprocessor**. The **Viewer**'s **Properties window** allows you to specify visualization of the **Layer**.

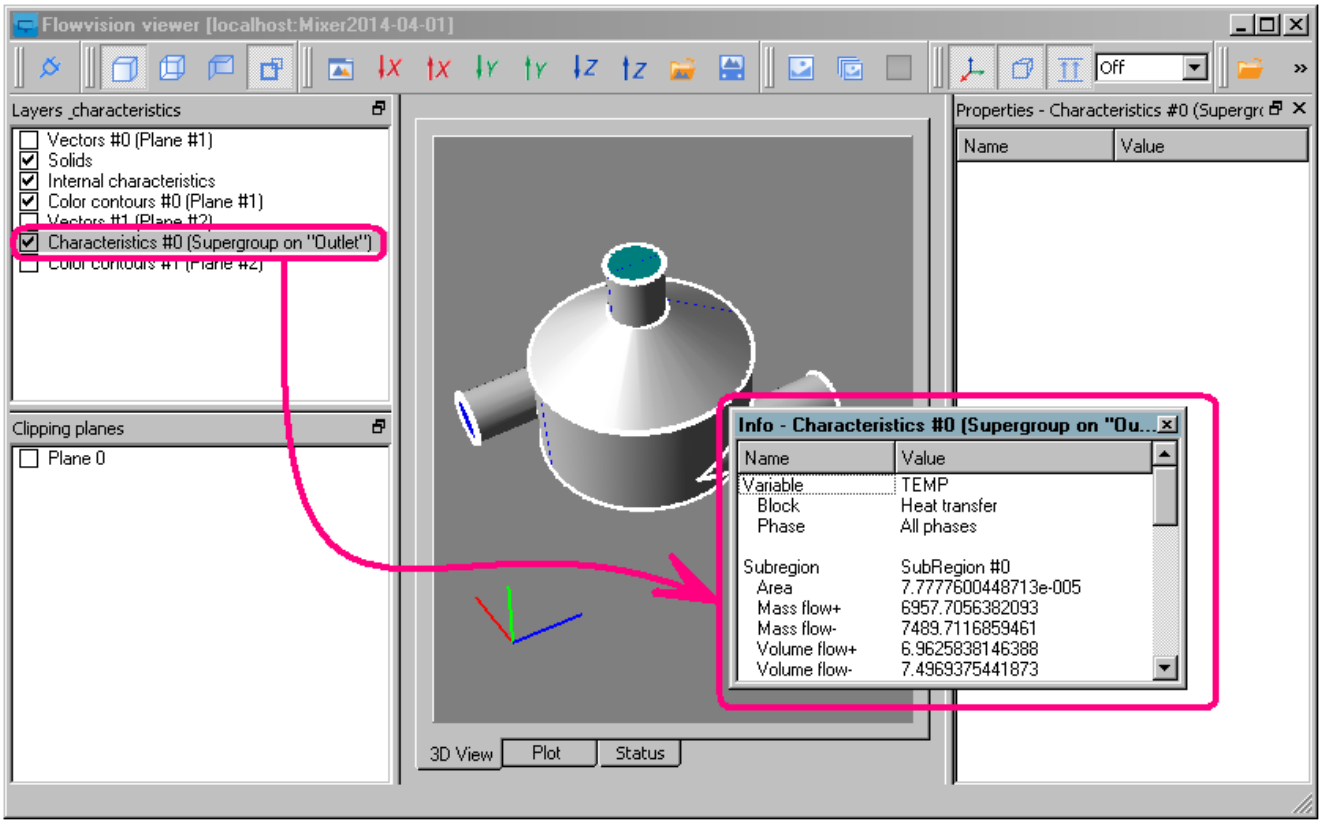
A *double-click* on a **Layer**'s line in the list opens the **Info window of Viewer** that contains information received from this **Layer**. Contents of the **Info window of Viewer** is similar to the contents of the **Info window in Postprocessor**.

7.3.4.2 Characteristics in the Viewer's "Layers & characteristics" window

Characteristics are created and defined in **Pre-Postprocessor** during preparing the project.

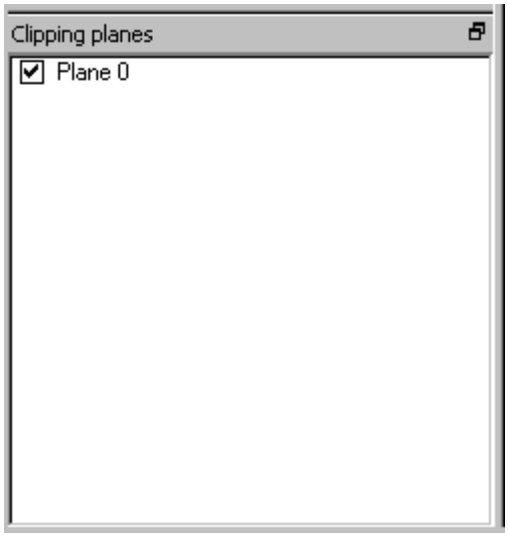
When you select a **Characteristic**'s line in the list, the **Viewer's Properties window** for this **Characteristic** opens.

A *double-click* on a **Characteristic**'s line in the list opens the **Info window of Viewer** that contains information received from this **Characteristic**. Contents of the **Info window of Viewer** is similar to the contents of the **Info window in Postprocessor**.



7.3.5 The "Clipping planes" window of Viewer

The **Clipping planes** window displays the list of clipping planes, containing in the project.



The planes are created in**Pre-Postprocessor**.

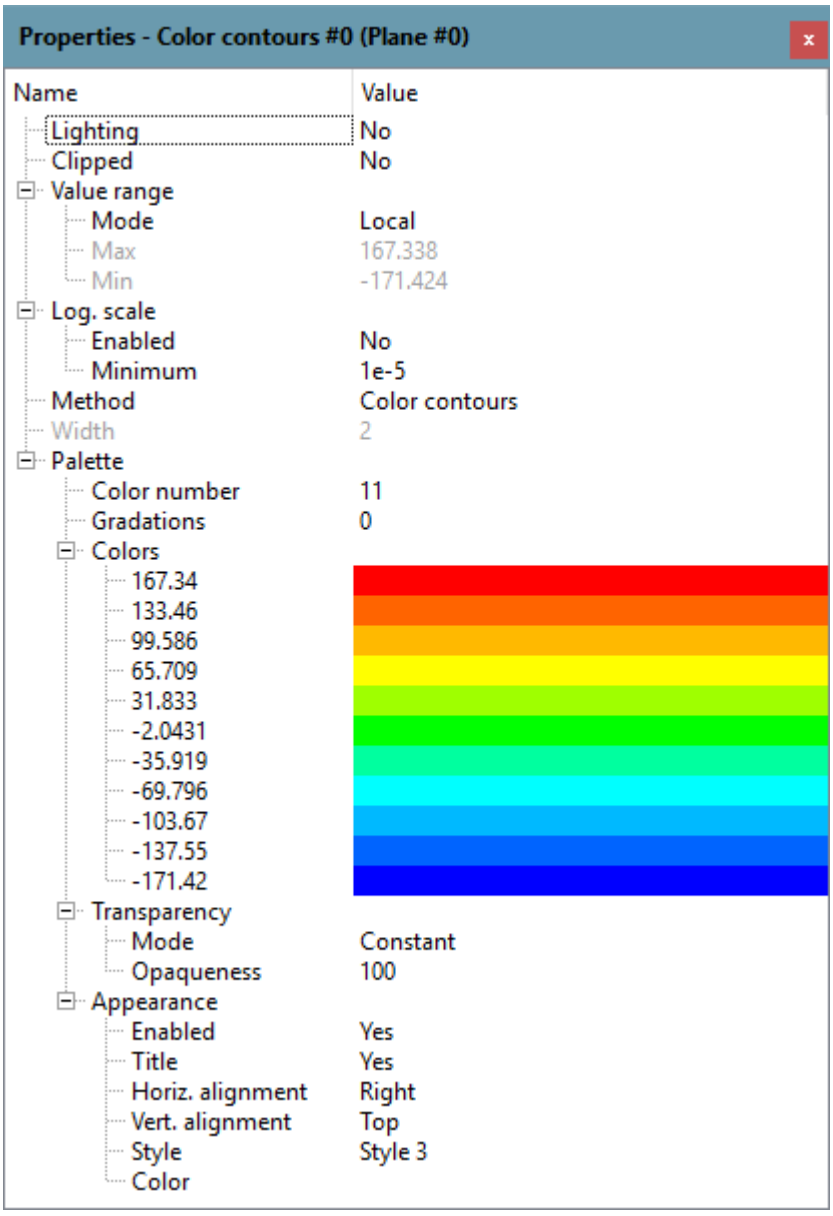
In **Viewer** displayed only plane which **Pre-Postprocessor** defined as cutting off objects, and in the absence of such a plane is shown having a reference point in the center box and normal oriented along the axis OX. Reference point and the orientation of this plane can be changed in its **Properties** window.

When you select a plane in the **clipping plane** displays the properties of the plane. **Properties** window planes in **Viewer** contain truncated set of properties planes compared with a set of properties existing in **Postprocessor**.

The list of planes before each plane is checkbox. If this box is checked, then the plane is clipping object in the graphical window of Viewer.

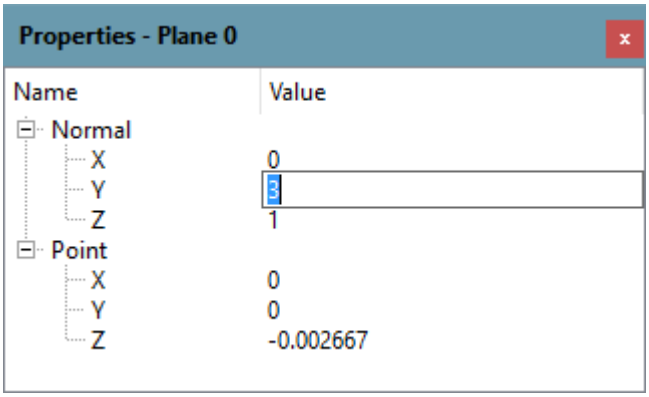
7.3.6 The "Properties" window of Viewer

The **Properties** window of **Viewer** allows you to view and/or change parameters of **Layers** and **Planes**.



The **Properties** window of **Viewer**

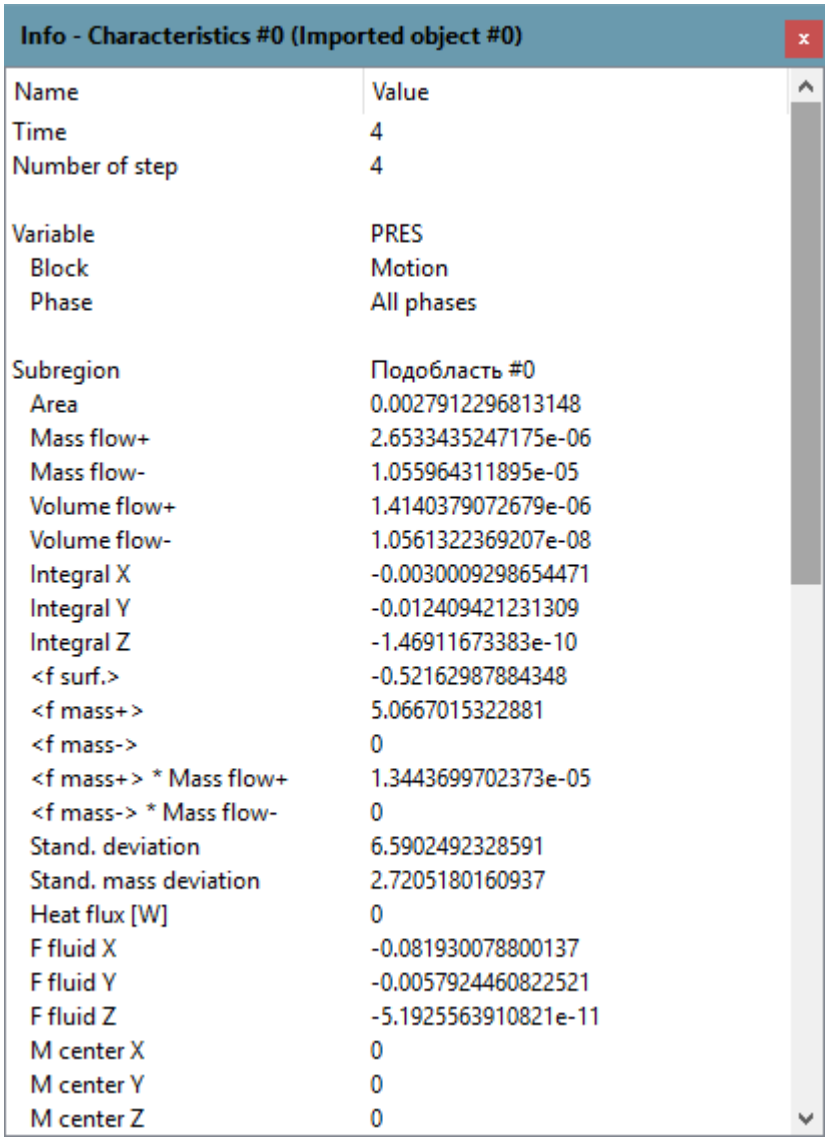
To change a parameter, double-click its value, then change the value, and press the **Enter** key on the keyboard or click outside this input field.



To change a parameter, double-click its value

7.3.7 The "Info" window of Viewer

The **Info** window of **Viewer** displays the data, which are being received from the selected **Layer** or **Characteristic**.



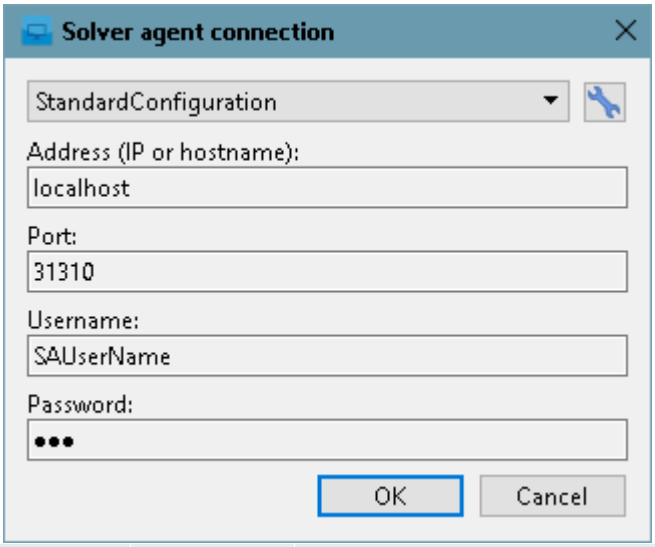
To open the **Info** window, double-click the line of a **Layer** or **Characteristic** in the **"Layers & characteristics" window of Viewer**.

7.3.8 Receiving data from Solver

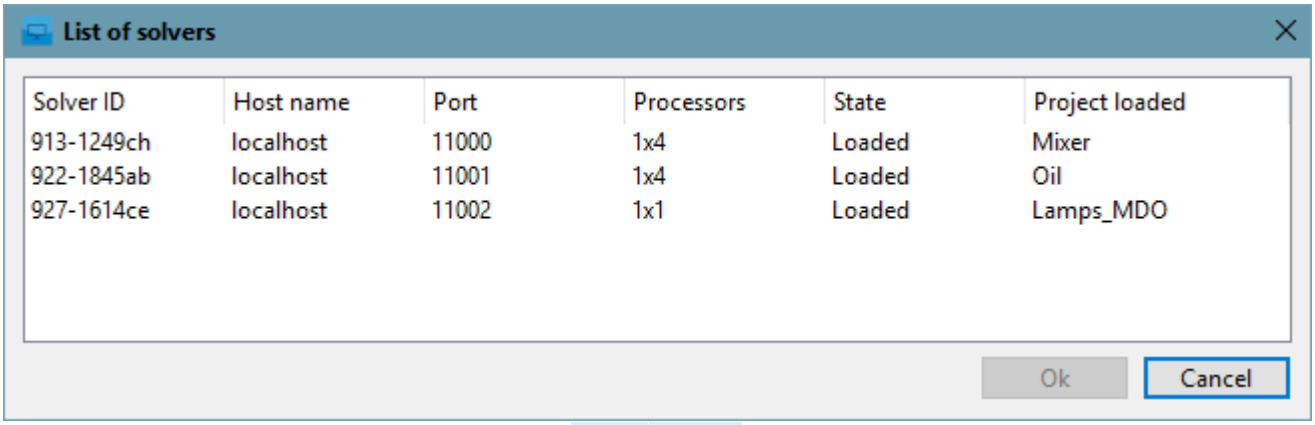
Retrieving data from **Solver** occurs after the user logins and selects a **Solver**, to which **Viewer** connects, from the list of available **Solvers**.

The **List of solvers** window is used to connect to running **Solvers**, on which projects are loaded.

The **List of solvers** window opens by clicking the button in the toolbar **Connection** and access to will require user authentication on Solver-Agent.



User authentication on **Solver-Agent** is required to access to the list of **Solvers**



List of **Solvers**

The **List of solvers** dialog box displays information about running **Solvers**. The information is displayed in the following columns:

Column	Description
Solver ID	Identifier of the **Solver** in **Solver-Agent**
Host name	Identifier of the computer, on which the **Solver** is running
Port	The port for connecting **Pre-Postprocessor** and **Viewer** to the **Solver**
Processors	The number of processors and the number of nuclei that are running **Solver**
State	Status of the **Solver**: • **Loaded** – a project is loaded on the **Solver**, and the **Solver** is not calculating now • **Calculates** – the **Solver** is calculating now

Column	Description
	• **Standalone** – the **Solver** is executing a batch file now
Project loaded	Name of the project, which is loaded on the **Solver**

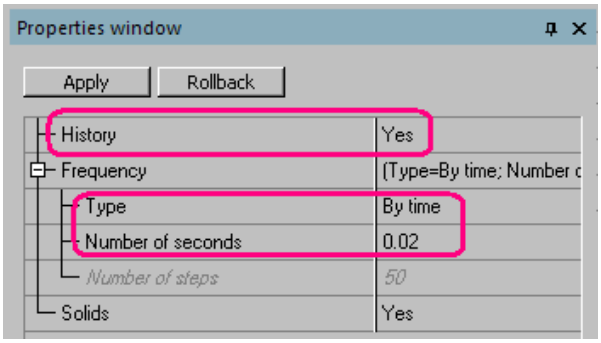
After connecting to the **Solver** to obtain data visualization in the files becomes unavailable (so **Viewer**'s toolbar "**Navigation**" is inactive).

7.3.9 Creating animation in Viewer

Animation is created using an external program that generates a video file from multiple graphic image files received from *FlowVision*. Animation presents non-stationary processes (turbine wheel rotation, waves on a water surface, etc.) more vividly.

Since **Viewer** can display both data received directly from a running **Solver** or stored visualization data, the processes of creating animation for the two data types will differ.

Steps for creating animation in Viewer

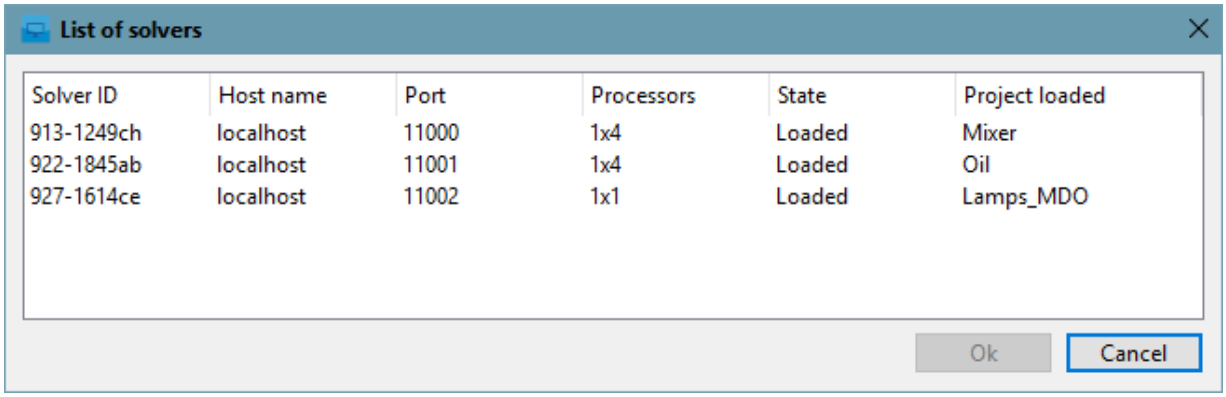
Step	Creating from stored visualization data	Creating from data received directly from a running Solver
1	Set up saving data visualization (layers) history in **Pre-Postprocessor** before starting computation. To do so, set the following in the **Solver** tab in **Pre-Postprocessor** using the properties window of the Layers autosave element: • **History = Yes** • **Frequency > Type = By time** • **Frequency > Number of seconds = *(desired autosave interval)***  Properties window Apply Rollback History Yes Frequency (Type=By time; Number c Type By time Number of seconds 0.02 Number of steps 50 Solids Yes ***Note:*** *) **Type = By time** is recommended to autosave history in case it is planned to create animation. You can also use **Type = Automatic** or **Type = By Step**, but in this case if time step during process calculation is not constant, creating animation with linear physical process flow timing will not be possible.	Skip this step. Setting up saving of visualization data (layers) is not required.
2	Start project computation and wait for it to complete. Find the folder where the visualization data (layers) created by **Pre-Postprocessor** were saved as `fvvis` files. These `fvvis` files are stored during calculation in the server part of project, but can be moved elsewhere automatically (using command **File > Download additional files > Visualization files** in **Pre-Postprocessor**) or manually (for example, using the *ftp* protocol or a removable drive). If the *visualization directory* is located in the server part of the project is inaccessible for the **Viewer**	Connect to **Solver** in **Pre-Postprocessor** and, if there are several **Solvers** running, make sure it is the one running the project computations. To record animation beginning not from the start of computation but from another moment of time, start the computation by the (**Start computation**) button, let the **Solver** run until the necessary moment and stop the computation by the (**Stop computation**) button. Do not run project computation before image recording is activated in **Viewer** (otherwise images

<table>
<tr><th colspan="3">Steps for creating animation in Viewer</th></tr>
<tr><th>Step</th><th>Creating from stored visualization data</th><th>Creating from data received directly from a running Solver</th></tr>
<tr><td></td><td>(or its location is not known), provide access to <code>fvvis</code> files for the Viewer.

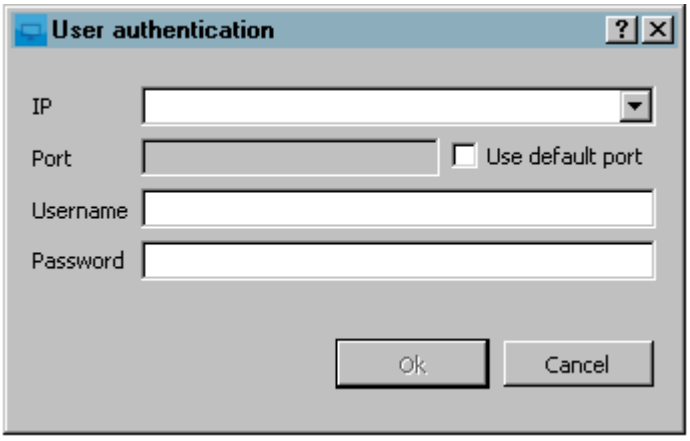
This can be done the following ways:
If the client part of the project is accessible for Viewer and there is a network connection between Pre-Postprocessor and Solver, then:
open the project in Pre-Postprocessor and connect to Solver
select File > Download additional files > Visualization files in the Pre-Postprocessor menu. The files will be downloaded into the client part of the project.
If there is no connection between Pre-Postprocessor and Solver, copy the <code>fvvis</code> files manually from the server part of the project to some directory accessible for Viewer.</td><td>generated before starting recording will not be saved to files by the Viewer).</td></tr>
<tr><td>3</td><td>Open Viewer without connecting to any Solver, i.e., click the Cancel button in the Solver selection window.

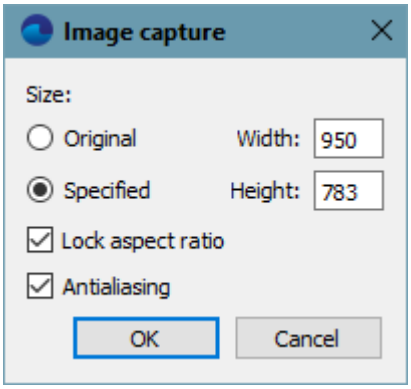
User authorization to connect to Solver is not required to load visualization data, so it is not necessary to enter credentials in the user authorization window (click Cancel).</td><td>Open the Viewer and connect to the Solver, which will run the project computation.

User authorization will be necessary when connecting to the Solver.

Start (or restart) the computation in Pre-Postprocessor.</td></tr>
<tr><td colspan="3">Illustrations:
 

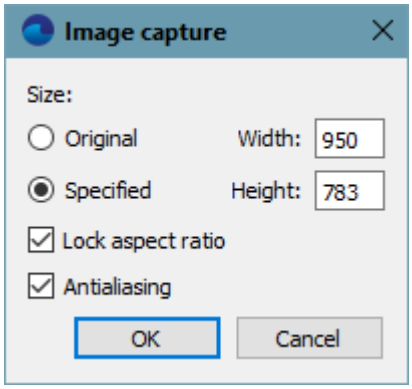
The List of solvers window for selecting a Solver</td></tr>
</table>

Steps for creating animation in Viewer		
Step	**Creating from stored visualization data**	**Creating from data received directly from a running Solver**
	 User authorization window	
4	In **Viewer** indicate the folder containing **`fvvis`** files from which visualization data created in **Pre-Postprocessor** will be derived (the folder selection was to be performed at step 2). To do so click the button and select the necessary folder from the window that opens.	Skip this step (since **Viewer** will receive data directly from **Solver** during project computation and not from **`fvvis`** files).
5	Set up the scene in the **Viewer**'s **View** window - select layers to display and their parameters, perspective settings, smoothing, layer offset; select camera position (point of view). To preserve the identical camera position for different projects, or to restore the camera position if it has been accidentally moved, use the (**Load View**) and (**Save View**) buttons.	
6	Go to: • first step of recording (using the button) • *or* to the necessary step of recording to begin playback from (using buttons , , ,)	Skip this step.

<table>
<tr><th colspan="3">Steps for creating animation in Viewer</th></tr>
<tr><th>Step</th><th>Creating from stored visualization data</th><th>Creating from data received directly from a running Solver</th></tr>
<tr><td>7</td><td colspan="2">Start sequential image capture from the Graphic window to file using the (Start capturing an image sequence) button.
In the Image capture and in the operation system's Save image windows, set the parameters of graphics files, their location, format and name prefix:
 

Setting image parameters (size, anti-aliasing)
The Image capture window is described in detail in a separate subsection below.</td></tr>
<tr><td>8</td><td>Start sequential loading of visualization data by clicking the (Start automatic sequential transition through the steps of the calculation history) button.</td><td>Start the project computation by clicking the (Start computation) button in the Pre-Postprocessor.</td></tr>
<tr><td>9</td><td>To stop loading visualization data before completion, click the (Stop automatic sequential transition through the steps of the calculation history) button.
If this button is not pressed, the visualization will go on displaying until the last <code>fvvis</code> file is reached.</td><td>Skip this step.</td></tr>
<tr><td>10</td><td colspan="2">To stop creating graphic files, click the (Finish capturing an image sequence) button.</td></tr>
<tr><td>11</td><td>Skip this step.</td><td>If there is no need to continue with the process computation, it can be stopped using (Stop computation) button in Pre-Postprocessor.</td></tr>
<tr><td>12</td><td colspan="2">Convert the sequence of stored graphics files into animation using any third party software for creating videos from image files.</td></tr>
</table>

The «Image capture» dialog box



The **Image capture** dialog box sets the parameters for saving an image or series of images (after clicking (**Save the contents of the graphics window to an image file**) or (**Start capturing an image sequence**) in the **Capture** tool panel).

The window contains the following elements:

Element		Description
Size		
	Original	Size of the saved image is equal to the size of the **Viewer**'s **View** window.
	Specified	Size of the saved image is set by the user.
	Width	Image width
	Height	Image height
Lock aspect ratio		Preserve image ratio relative to the initial **Viewer** window. When this option is turned on, the user can set only one image dimension (either **Width** or **Height**), and the second dimension will be automatically set by the program according to initial ratio.
Antialiasing		Anti-aliasing the lines in the saved image. If anti-aliasing is turned off in the **Rendering** toolbar, the anti-aliasing in the saved image will be selected automatically. Otherwise, the same mode will be selected as for the **View** window.

7.3.10 How to start Viewer

Starting Viewer from the "Programs" menu of *Windows*

To start **Viewer** from the menu of the *Windows'* **Start** button, open this menu and select **Programs > FlowVision (*version number*) > Viewer**.

Starting Viewer from a command line or in the batch mode

You can run **Viewer** from a command line or in the batch mode with the following keys:

Key	Description	Default value
sshost=HOST ssport=PORT ssname=NAME	Connect to **Solver** after start NAME - name or IP address of the computer on which **Solver** is running PORT - port number used by **Solver** to receive connections from **Pre-Postprocessor** and **Viewer** NAME - user name of the user, who started the **Solver**	
ssproxyhost=PROXYHOST	If these parameters are specified, then connection to **Solver** is provided via **Retranslator** (**FvConnect**)	

Key	Description	Default value
ssproxyport=PROXYPORT	PROXYHOST - name or IP address of the computer on which **Retranslator** is running PROXYPORT - port number used by **Retranslator** to receive connections	
visfile="path/file.fwis"	Open the defined **`fvvis`** file after the start	
camera=VIEW *or* camera="path/file.fvcam"	Define view or load a specified **`fvcam`** file after the start Possible values of the parameter VIEW: • **x+** = Orientation along axis OX • **x-** = Orientation opposite axis OX • **y+** = Orientation along axis OY • **y-** = Orientation opposite axis OY • **z+** = Orientation opposite axis OZ • **z-** = Orientation opposite axis OZ	**z+**
perspective=yes *or* perspective=no	Turn the perspective on or off: • **yes** = Turn on the perspective • **no** = Turn off the perspective	**yes**
scale=value	Scale the view with defined factor	**1.0**
xpan=value	Move the view horizontally on the defined distance *)	**0.0**
ypan=value	Move the view vertically on the defined normalized distance *)	**0.0**
rotate=value	Rotate the view on an angle defined in degrees (positive direction is clockwise)	**0.0**
coordsys=yes *or* coordsys= no	Turn on/off the symbol of coordinate system: • **yes** = display the symbol • **no** = does not display the symbol	**yes**
fill=yes *or* fill=no	Turn on/off displaying of faces of triangles of the geometry: • **yes** = Turn the displaying on • **no** = Turn the displaying off	**yes**
wireframe=yes *or* wireframe=no	Turn on/off displaying of edges of triangles of the geometry: • **yes** = Turn the displaying on • **no** = Turn the displaying off	**no**
no_outside=yes *or* no_outside=no	Hide or show the outer side of the geometry: • **yes** = Hide the outer side of the geometry • **no** = Show the outer side of the geometry	**yes**
contours=yes *or* contours=no	Turn on/off displaying of borders of **Groups**: • **yes** = Turn the displaying on • **no** = Turn the displaying off	**yes**
bkcolor=RRGGBB	Define the background color (in hex)	
screenshot="path/file.*"	Save the image from the **View** window in a file. Possible file extensions: **`BMP`**, **`JPG`**, **`PNG`**, **`PPM`**, **`XBM`**, **`XPM`** Default extension is **`JPG`** If this parameter is specified, then **Viewer** stops (terminates) after creation of the file.	
sswidth=value	Define width of the image saved in a file (in pixels)	**640**
ssheight=value	Define height of the image saved in a file (in pixels)	**480**
ssaa=yes	Turn smoothing on/off	**no**

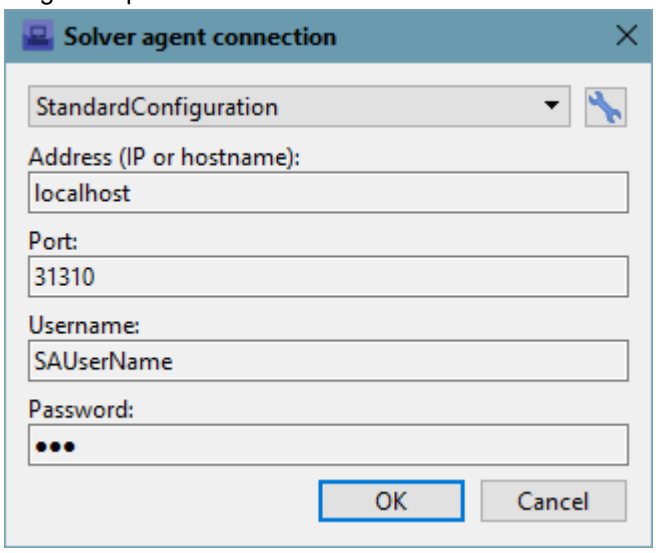
Key	Description	Default value
or ssaa=no		
loop_playback	Displaying results of a simulation(s) in a loop mode. *Note:* this key is used without a value.	

*) Normalized distance 1.0 corresponds to a half of the maximal size of the outer box embracing the geometry model.

7.4 Terminal

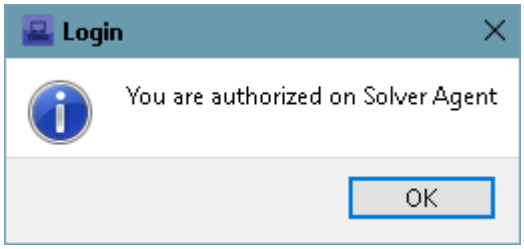
Terminal is one of the client components of *FlowVision*. **Terminal** provides licensing and registering a user, as well it allows you to manage computations of prepared projects that locate in the Server directory.*)

When you start **Terminal**, the program prompts you to carry out user authentication on Solver-Agent and the **Solver agent connection** dialog box opens:



Authorization is required if you plan to work with projects. If the user has not been registered yet, you have to follow a new Solver-Agent user registering procedure.

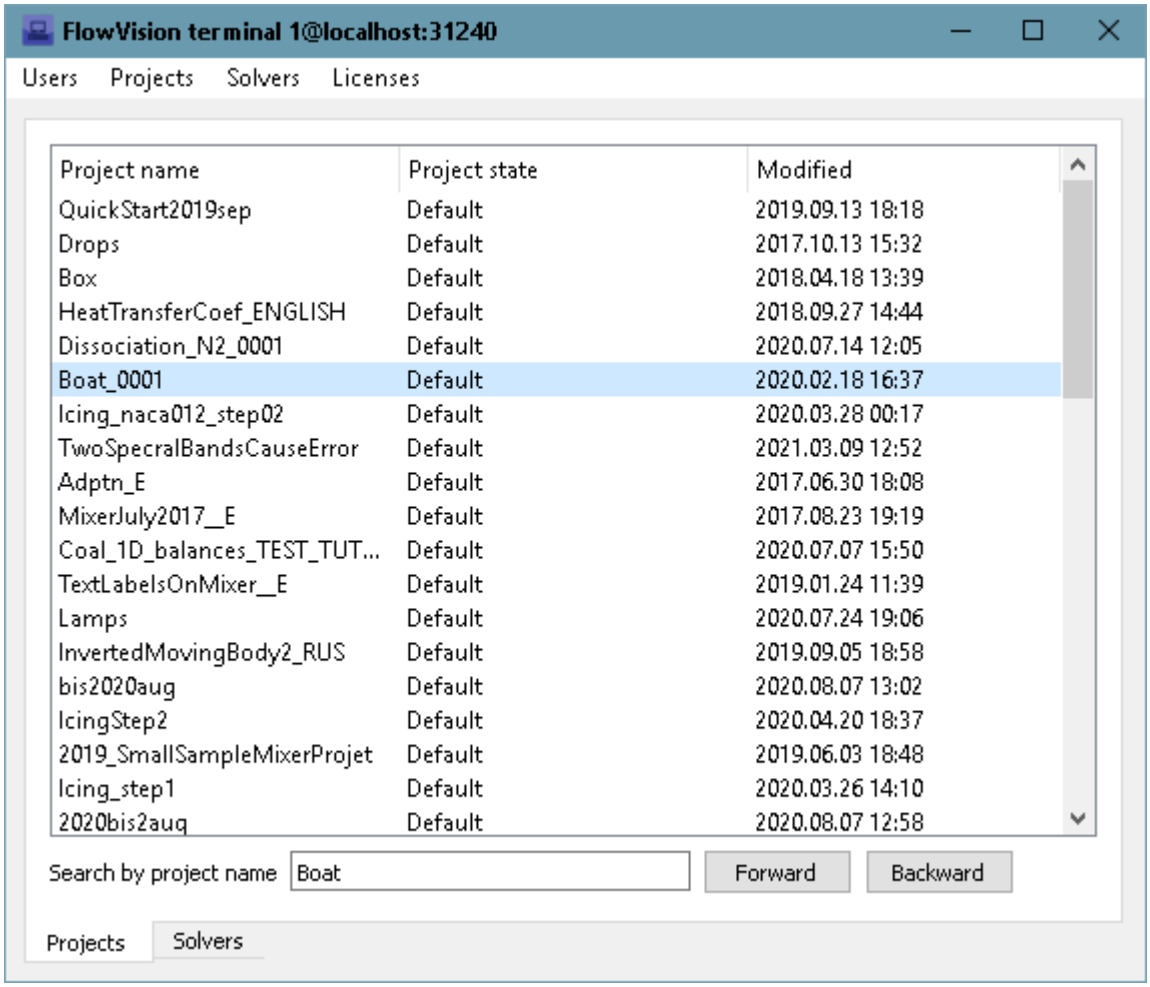
When the data are specified correctly, a successful authentication will occur and the program display the message "**You are authorized onSolver Agent**":



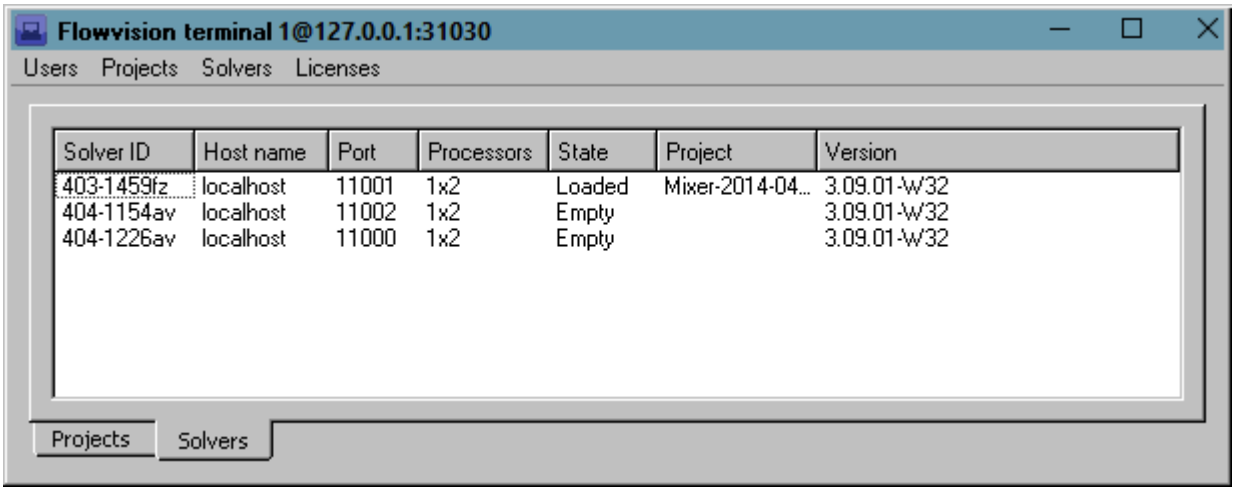
If the authentication fails, the program also displays a message about this (see subsection *"Messages when user's authorization on Solver-Agent is unsuccessful"* below).

After a successful authentication the **Terminal**'s window opens, which consists of two tabs:

- **Projects** (it can be opened using **Ctrl+1** hot keys)
- **Solvers** (it can be opened using **Ctrl+2** hot keys)



Terminal's window, tab **Projects**



Terminal's window, tab **Solvers**

Title bar of the **Terminal**'s window displays the user's name, the computer's IP address or host name, and the port number in the **User@Host:Port** format.

The upper part of the **Terminal**'s window contains menus:

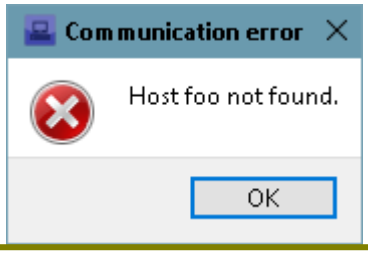
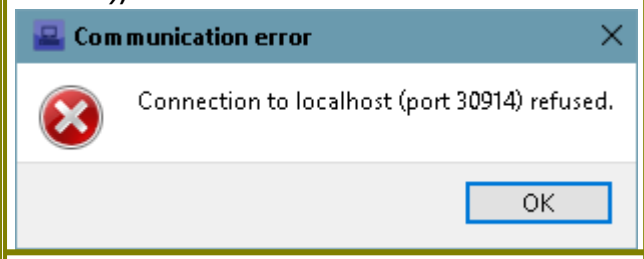
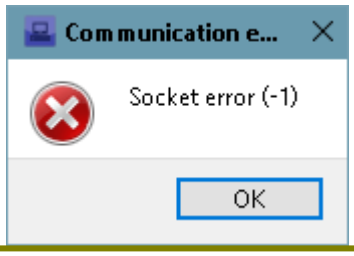
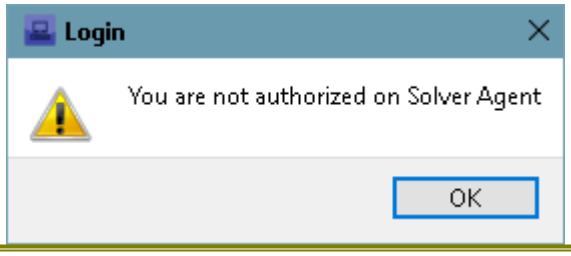
- **Users**
- **Projects**
- **Solvers**
- **Licenses**

See also: Configuration file of Terminal (FvTerminal.cfg).

Note:

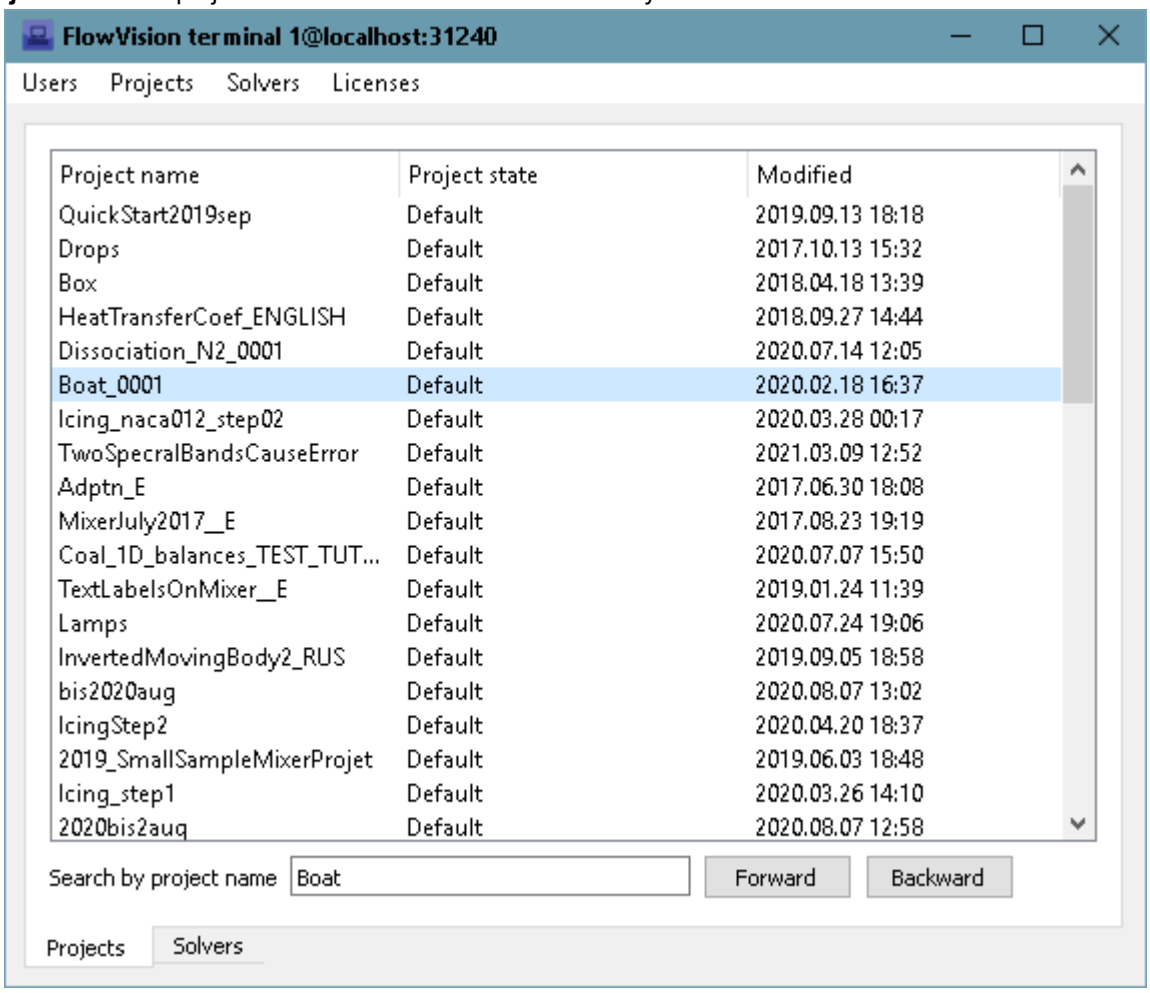
*) To ensure that the project is presented in the server directory, you have to upload it to **Solver** from **Pre-Postprocessor** or copy it there manually.

Messages when user's authorization on Solver-Agent is unsuccessful

Error message	Cause of Error	Correction
Host *(address/network_name)* not found. 	Incorrectly specified IP address or host name of the computer.	Specify a correct IP address or host name of the computer and make sure that this computer is running and available.
Connection to (*address/network_name)* (port *(port number)*) refused. 	Incorrectly specified the port number (except than the port **0**)	Specify a valid port number (see section Parameters in configuration files).
Socket error 	An attempt to connect to the reserved port **0**	
You are not authorized on Solver Agent 	Incorrect user name and/or password	Enter the correct user name and password

7.4.1 Terminal's tab "Projects"

The **Projects** tab lists projects that locate in the Server directory.



To view the current list of projects, you have to update it by the menu command **Projects** > **Refresh list of projects** (or press the key **F5** key on your keyboard).

The information is displayed in the following columns; by default some of the columns might be hidden:[*)]

Column	Description
Project ID	The project's identifier
Project name	Name of the project (defined by the user when they creates the project)
Project state	The project's status: • **Default**: the project is not loaded on a **Solver** • **Loaded**: the project is loaded on a **Solver**, computation is not running • **Calc**: calculation of the project is running • **Saved**: the project is loaded on a **Solver**, the calculation has been done, results of the calculation has been saved • **Unsaved**: the project is loaded on a **Solver**, the calculation has been done, results of the calculation has *not* been saved • **In queue**: the project is placed into the queue for computation • **Waits for decimation** – decimation (that was run from **Terminal**) a project, from the moment of clicking the **Save** button in the **Non-steady-state steps decimation** dialog box until finishing the decimation. See menu Projects. • **Decimation error** – unsuccessful attempt of decimation a project from **Terminal** in situation when the project is connecting to a **Solver** that was opened from **Pre-Postprocessor**.
Size, Mb	The project's size (in megabytes)

Column	Description
	When you are working with large projects, displaying this column might slow down work of **Terminal**, so this column is not displayed by default*).
Modified	Date of the last change of the project
Full name	Name and the full path of the project's file in the Server directory
Solver ID	The **Solver agent**'s identifier of the **Solver**, on which the project is loaded (this identifier is a property of projects that are loaded on a **Solver**)
Plugin	Name of the plugin, which was used at creating the project (this data item is available for projects that were created using a plugin)
History	This column displays how many intermediate data save occurred
Connector	This column displays if there is a connection with a third-party program (for example, connection with *Abaqus*)
Host name	IP address or the name of the computer, on which the **Solver** is running, on which the project is loaded
Port	Port for connection the **Solver**, on which the project is loaded, to **Pre-Postprocessor**
Processors	Number of processors and number of nuclei, on which the **Solver** is running (these numbers are separated by "x"), on which the project is loaded
Process ID	Identifier of the **Solver**, on which the project is loaded, in the operating system
Version	Version of the **Solver**, on which the project is loaded
Build	Assembly date of the **Solver**, on which the project is loaded, in the YYYYMMDD format and the **Solver**'s changeset number separated by the symbol "-"

Note:

*) Displaying of columns is set in the **Projects list view settings** dialog box, which opens by the menu command **Projects** > **Project list view settings**

Sorting

Clicking on a title of a column causes sorting the list of projects by contents of this column. Sorting by the project's name is not case-sensitive. The specified sorting will be saved and used in further runs of **Terminal**.

Search by project name

You can search projects by a string that is included in their names. To do so, enter the desired string into the **Search by project name** field and click either **Forward** or **Backward** screen button one or several times:

If the table doesn't contain a project with a name that includes the specified string, buttons **Forward** and **Backward** are not active.

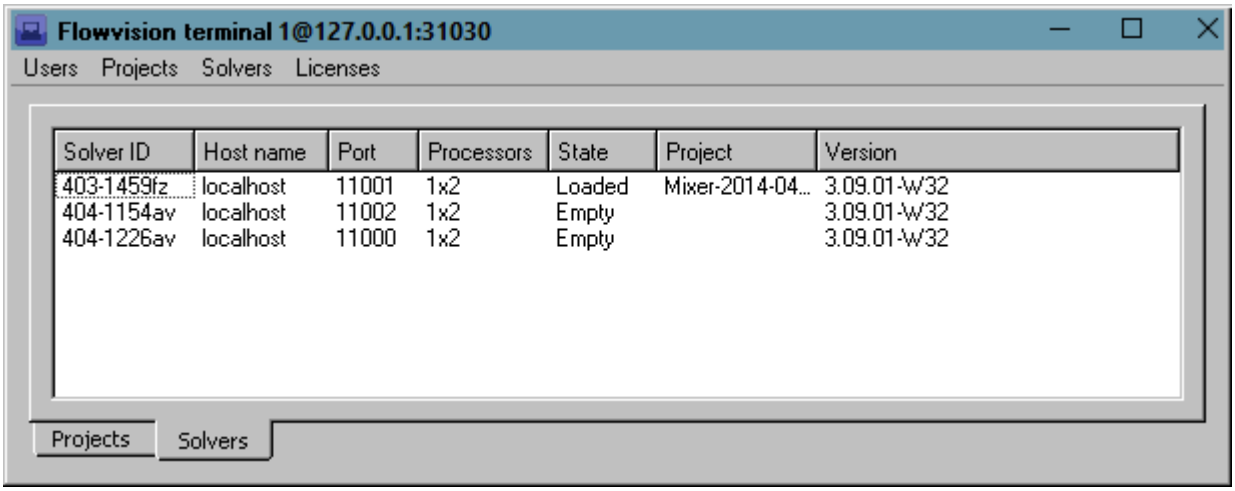
When the cursor locates in the **Search by project name** field, pressing the **Enter** key on the keyboard acts the same as clicking the **Forward** screen button.

Context menu in the list of projects

When you right-click in the list of projects, a context menu, same as the Projects menu, opens.

7.4.2 Terminal's tab "Solvers"

The **Solvers** tab displays information about running **Solvers**.



To view the actual list of **Solvers**, you have to update it using the **Solvers** > **Refresh list of active solvers** menu command (or press the **F5** key on your keyboard).

The information is displayed in the following columns, some of which by default can be hidden:*)

Column	Description
Solver ID	Identifier of the **Solver** in **Solver-Agent**
Host name	IP address or the name of the computer, on which the **Solver** is running
Port	Port for connection the **Solver** to **Pre-Postprocessor**
Processors	Number of processors and number of nuclei, on which the **Solver** is running (these numbers are separated by "x")
State	Condition **Solver**: • **Empty** - no project is uploaded on the **Solver** • **Loaded** - a project is uploaded on the **Solver**, and the **Solver** is not calculating • **Solving** - the **Solver** is calculating • **Autonomous** - the **Solver** is processing a batch file
Project	Name of the project, which is uploaded on the **Solver** (the information is presented if a project is uploaded on the **Solver**)
Process ID	Identifier of the **Solver** in the operating system
Version	Version of the **Solver**
Build	Assembly date of the **Solver** in the YYYYMMDD format and the changeset number separated by the symbol "-"

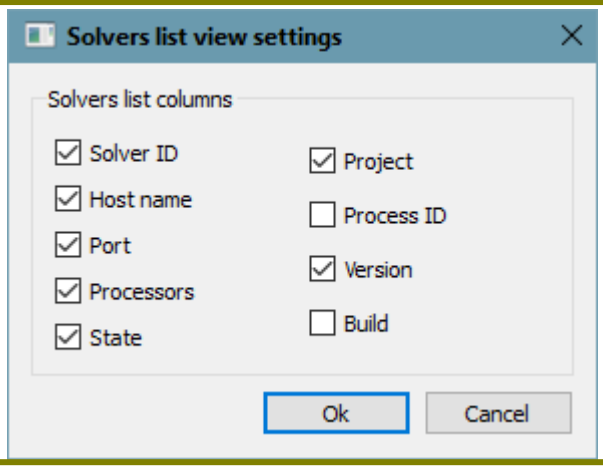
Note:

*) Displaying or hiding the columns is defined in the dialog box, which opens by the menu command **Solvers** > **Solvers list view settings**.

Context menu of the "Solvers" tab

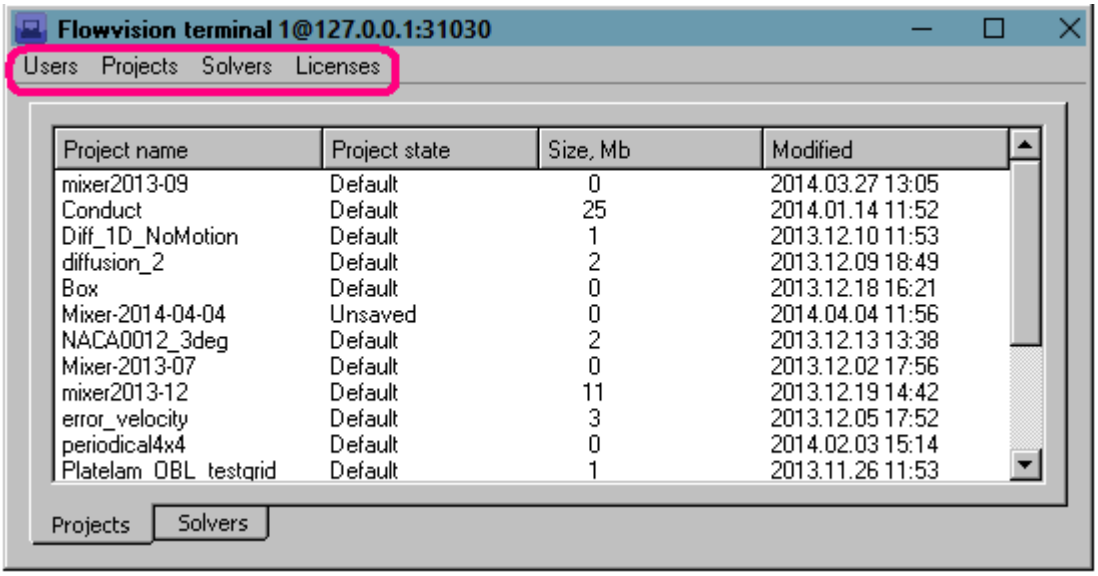
The context menu, which opens by right-clicking in the **Solvers** tab, contains the following commands:

Menu item	Description	Hot key
Start project solving	Starts computation of the project, which is loaded on the selected **Solver**. The **Starting solve** dialog box will open.	
Stop project solving	Starts computation of the project, which is loaded on the selected **Solver**.	
Save project data	Saves results of computation of the project, which is loaded on the selected **Solver**, in the user's **Server directory**.	
Unload project from solver	Unloads a project from the selected **Solver**. If the project is not saved, the program requests confirmations in two dialog boxes for this action to avoid data loss ("**Project is not saved and all data will be lost. Continue?**").	
Stop, save and unload	Stops computation of the project, which is loaded on the selected **Solver**, and then saves results of the computation and unloads the project from the **Solver**.	
Export results	Exports results of computation of the project, which is loaded on the selected **Solver**, not only during the computation, but also after the end of the computation using the saved data from the grid. See section Data export after computation.	
Run new solver*)	Run a new **Solver** (the **Solver running** dialog box will open).	**Ctrl+R**
Kill solver*)	Terminate the selected **Solver**.	
Attach viewer*)	Start **Viewer** and connect it to the selected **Solver** (this command is available when the **Solver** is running with a project loaded on it).	
Solvers list view settings*)	Configure a set of displayed columns in the **Solvers** tab. This opens the **Solvers list view settings** dialog box where you can select the columns, which will be displayed:	

Menu item	Description	Hot key
		
Refresh list of active solvers[*)]	Update information about active **Solvers** and opens the Solvers tab (if it isn't already opened).	**F5**

[*)] These commands are also available from the "Solvers" menu.

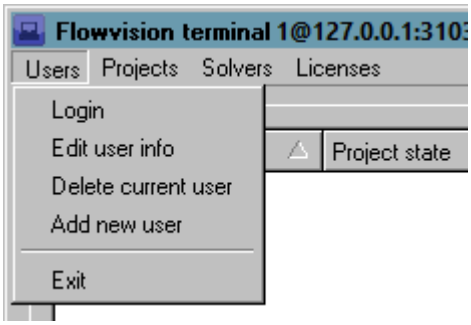
7.4.3 Terminal's menu



Terminal's menu consists of the following sub-menus that are described in separate sections:

- **Users**
- **Projects**
- **Solvers**
- **Licenses**

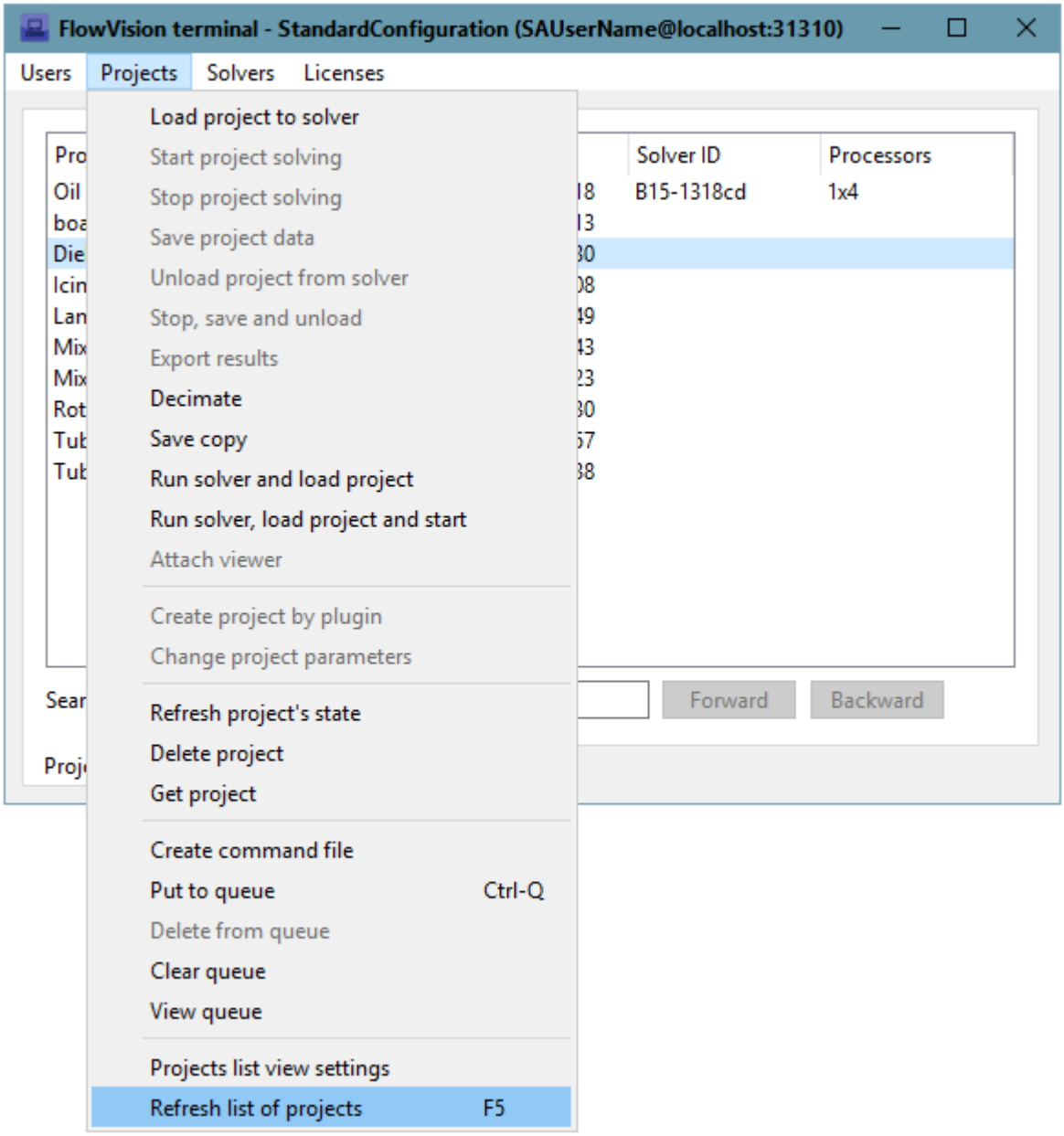
7.4.3.1 Menu Users



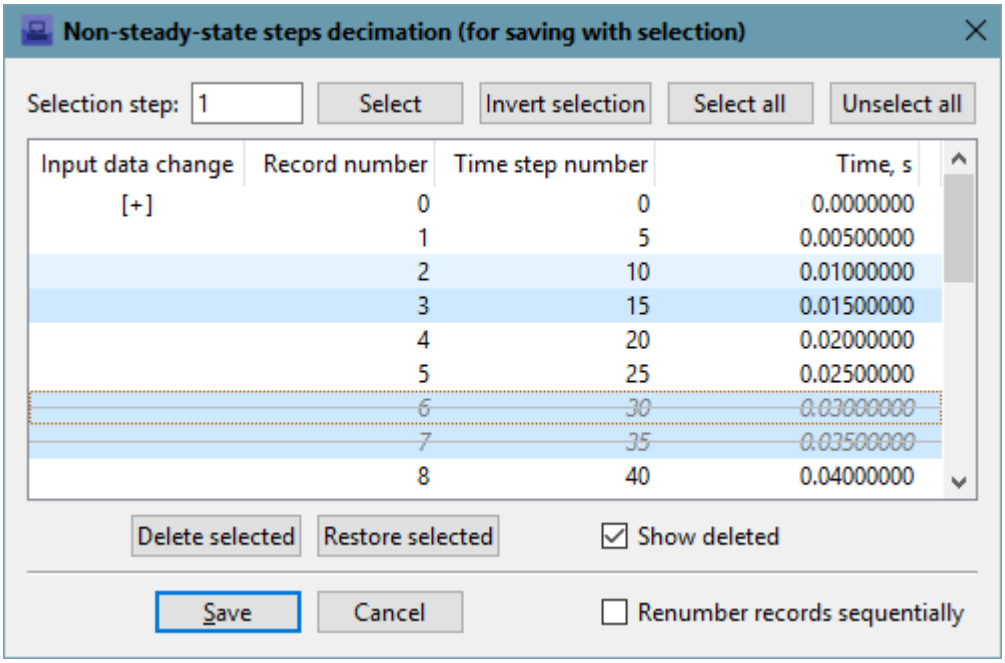
Command	Description
Login	Carry out user authentication on Solver-Agent

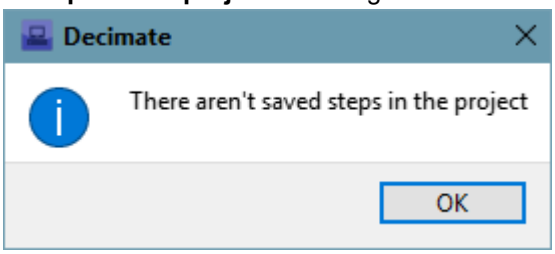
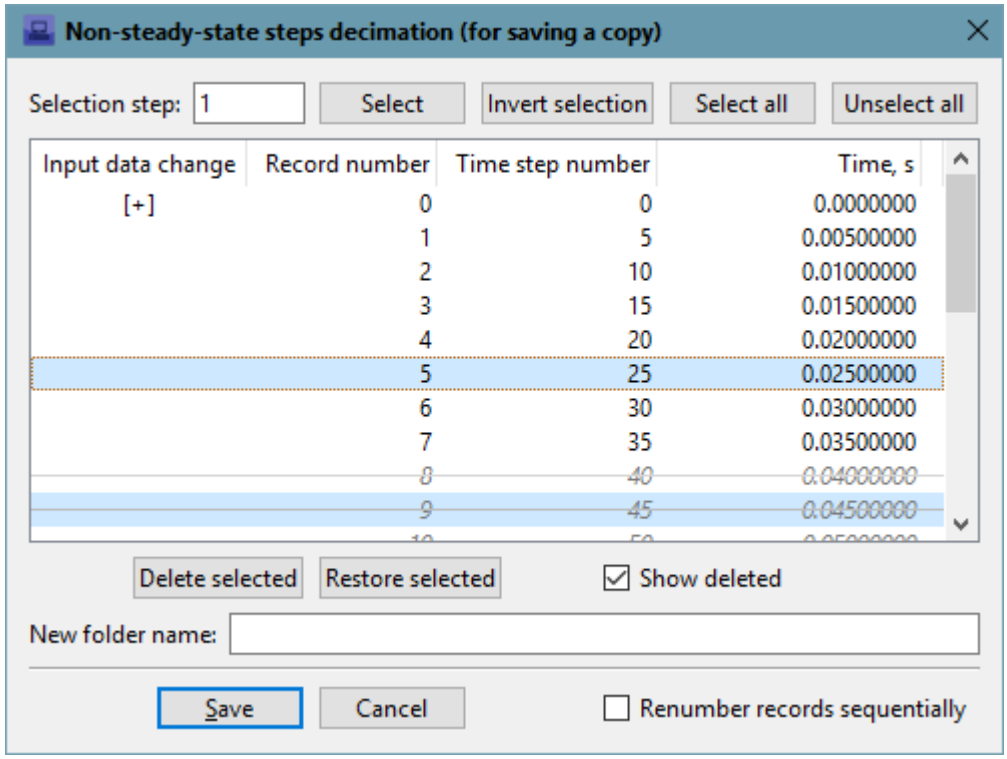
Command	Description
Edit user info	Changing profile (registration data) of a Solver-Agent user (for system administrators only)
Delete current user	Removing the current **Solver-Agent** user (for system administrators only)
Add new user	Creating a new **Solver-Agent** user (for system administrators only)
Exit	Exiting from **Terminal**

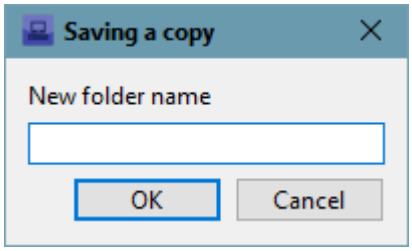
7.4.3.2 Menu Projects

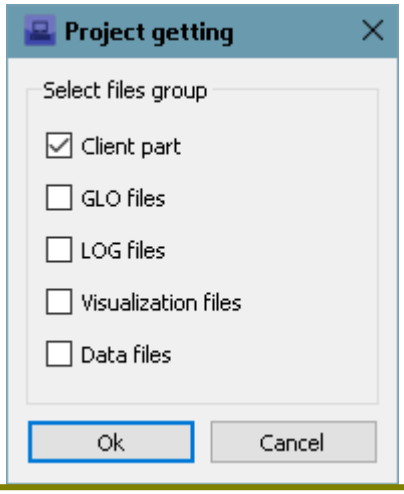
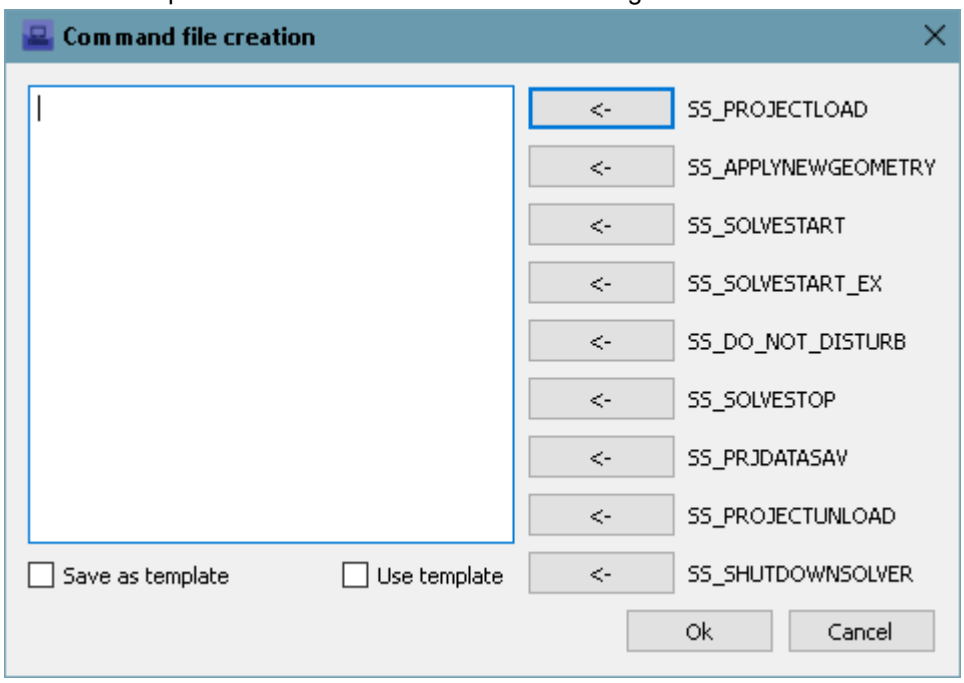


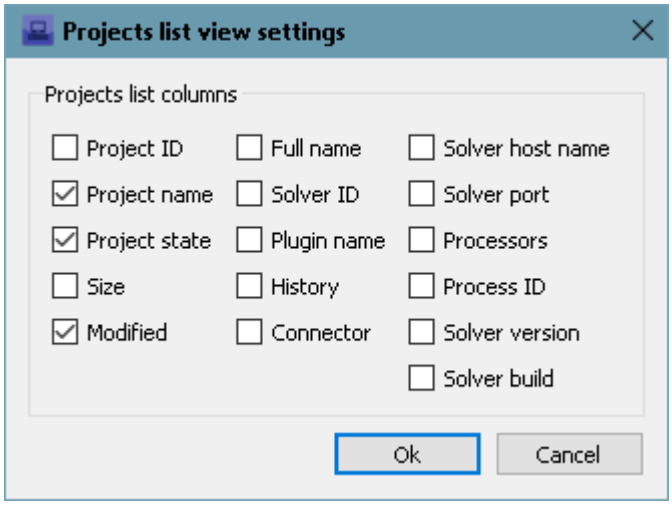
Menu item	Description
Load project to solver	This command uploads the project to a **Solver**, which is to be selected in the **List of solvers** dialog box, which opens. After a successful loading the project, the list of **Solvers** in the Solvers tab will be refreshed automatically.
Start project solving	This command starts the computation (opens the **Starting solve** dialog box).
Stop project solving	This command stops computation of the project.
Save project data	Saving the results in the user's **Server directory**.

Menu item	Description
Unload project from solver	Unloading the project from **Solver**. If the project is not saved, the program requests confirmations in two dialog boxes for this action to avoid data loss ("**Project is not saved and all data will be lost. Continue?**").
Stop, save and unload	This command stops computation of the project, and then saves results of computation and unloads the project from a **Solver**.
Export results	Exporting results of computation, not only during the computation, but also after the end of the computation using the saved data from the grid. See section Data export after computation.
Decimate	Decimating the saved steps of the non-steady-state record of the project, similarly as it done in **Preprocessor**. The **Non-steady-state steps decimation** dialog box will open:  See details in the section Saving a project, subsection Dialog box «Non-steady-state steps decimation». If the project *is* loaded on a **Solver**, the decimation is done by this **Solver** itself. If the project *is not* loaded on a **Solver**, then a new **Solver** will run in the mode "*(1 MPI process x 1 core)*". This **Solver** will consume no license options. You will have to select the type of this **Solver** from the **Solver to use** field (drop-down list) that will appear in the **Non-steady-state steps decimation** dialog box.  If the configuration file of Solver-Agent (FvSolverAgent.cfg) contains no line for running a **Solver** in the one-processor mode, as it can be, for example, on clusters, then the **Solver to use** list will be empty and decimation without loading the project on a **Solver** will be impossible.

Menu item	Description
	Also the decimation will not be done if the user runs one or several **Solvers** that totally consume the number of threads *N* more then maximal number of threads specified in settings of **Solver-Agent** (this is specified by the **MaxThreadNum** parameter in the configuration file of Solver-Agent (FvSolverAgent.cfg). To avoid this problem, you can intentionally overestimate the **MaxThreadNum** parameter in settings of **Solver-Agent**. If this parameter is not specified in the settings, its value is obtained automatically based on characteristics of the used computer. In the graphical user interface the **MaxThreadNum** parameter is set in **Configurator** as value from the **Maximum number of available threads** field in the **Solver agent configuration** dialog box (see Configurator's tab "Configuration/Logs"). If the project doesn't contain the non-steady-state record, the program displays the "**There aren't saved steps in the project**" message:  This command is not available for projects, which state is **Calc**. Also this command is not available for projects that contain no saved steps of non-steady-state record.
Save copy	This command allows saving a copy of the project (and also it can decimate the saved steps of the non-steady-state record of the new project). If the project contains the non-steady-state record, similarly as in the case of the **Decimate** command (see above), the **Non-steady-state steps decimation** dialog box will open, but in this case it will contain the **New folder name** field where you have to specify new location of the saved project:  If the project *is* loaded on a **Solver**, the decimation is done by this **Solver** itself. If the project *is not* loaded on a **Solver**, then a new **Solver** will run in the mode "*(1 MPI process x 1 core)*". This **Solver** will consume no license options. You will have to

Menu item	Description
	select the type of this **Solver** from the **Solver to use** field (drop-down list) that will appear in the **Non-steady-state steps decimation** dialog box.  If the configuration file of Solver-Agent (FvSolverAgent.cfg) contains no line for running a **Solver** in the one-processor mode, as it can be, for example, on clusters, then the **Solver to use** list will be empty and decimation without loading the project on a **Solver** will be impossible. Also the decimation will not be done if the user runs one or several **Solvers** that totally consume the number of threads *N* more then maximal number of threads specified in settings of **Solver-Agent** (this is specified by the **MaxThreadNum** parameter in the configuration file of Solver-Agent (FvSolverAgent.cfg). To avoid this problem, you can intentionally overestimate the **MaxThreadNum** parameter in settings of **Solver-Agent**. If this parameter is not specified in the settings, its value is obtained automatically based on characteristics of the used computer. In the graphical user interface the **MaxThreadNum** parameter is set in **Configurator** as value from the **Maximum number of available threads** field in the **Solver agent configuration** dialog box (see Configurator's tab "Configuration/Logs"). If the project doesn't contain the non-steady-state record, the **Saving a copy** dialog box will open, in which the program will request only the new location of the saved project:  See details in the section Saving a project. This command is not available for projects, which state is **Calc**.
Run solver and load project	This command starts a **Solver** and loads the selected project on this **Solver**.
Run solver, load project and start	This command carries out the following actions: 1) starts a **Solver** 2) loads the selected project on this **Solver** 3) runs computation of the selected project
Attach viewer	This command opens **Viewer** where you can view result of the project's computation.
Create project by plugin	Creating a new project based on a template plugin.
Change project parameters	Changing parameters of the project (that was created using a plugin).
Refresh project's state	Updating information about status of the project.
Delete project	Deleting the selected project from the **Server directory**.
Get project	Copying the selected part of the project: • the client part • `glo` files • `log` files • visualization files

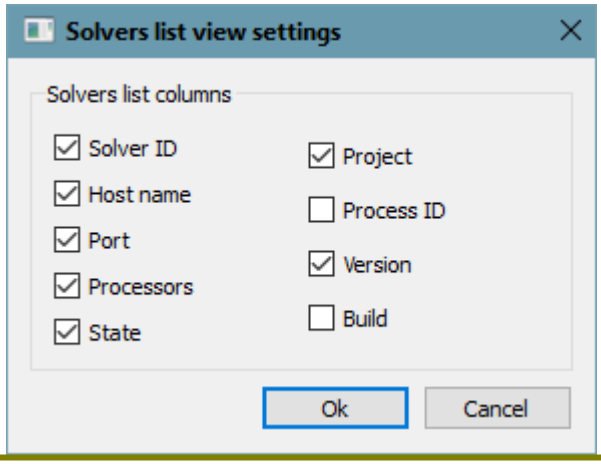
Menu item	Description
	• data files The **Project getting** dialog box will open: 
Create command file	This command creates a **Solver**'s command file and places it into the server's part of the project. The command opens the **Command file creation** dialog box:  The **Command file creation** dialog box contains the following interface elements: • the editing field for the command file – it contains the command file's text, which can be typed in manually or you can apply buttons with arrows to insert standard commands to the end of the command file. This field can be used as a simple text editor. If the server part of the project already includes a command file with the name **`command.txt`**, then contents of the file will be automatically displayed in this field. • button "**<- SS_PROJECTLOAD**" – adding the **SS_PROJECTLOAD** command with an argument with the relative path to the project's file on the server and three other empty arguments. • button "**<- SS_APPLYNEWGEOMETRY**" – adding the **SS_APPLYNEWGEOMETRY** command with empty arguments • button "**<- SS_SOLVESTART**" – adding the **SS_SOLVESTART** command with an argument that corresponds to values of the setting **Continue calculation** and **Use existing grid/data** that are specified in the **Starting solve** dialog box, which opens.

Menu item	Description
	• button "**<- SS_SOLVESTART_EX**" – adding the **SS_SOLVESTART_EX** command with the first argument that corresponds to values of the setting **Continue calculation** and **Use existing grid/data** that are specified in the **Starting solve** dialog box, which opens and empty the second and the third arguments. • button "**<- SS_DO_NOT_DISTURB**" – adding the **SS_DO_NOT_DISTURB** command • button "**<- SS_SOLVESTOP**" – adding the **SS_SOLVESTOP** command • button "**<- SS_PRJDATASAVE**" – adding the **SS_PRJDATASAVE** command • button "**<- SS_PROJECTUNLOAD**" – adding the **SS_PROJECTUNLOAD** command • button "**<- SS_SHUTDOWNSOLVER**" – adding the **SS_SHUTDOWNSOLVER** command • the **Save as template** checkbox allows you to save the created command file as a template that will be used by default. The program saves the template when you click **Ok**. • the **Use template** checkbox allows you to load the command file's template, which was saved before by the **Save as template** checkbox. When no template has been saved, this checkbox in unavailable. • button **Ok** – creating a command file with the specified contents, its transferring into the server part of the project and saving with the name **`command.txt`**. The old contents of the file **`command.txt`** will be replaced by the new contents. • button **Cancel** – closing the dialog box without saving the data
Put to queue	This command places the project into the queue of projects for calculation (opens the **Adding project to the projects queue** window). You can also use the **Ctrl+Q** hot keys to execute this command.
Delete from queue	This command removes the project from the queue of projects for calculation.
Clear queue	This command removes all projects from the queue of projects for calculation.
View queue	This command displays the queue of projects for calculation (opens the **Projects queue** dialog box).
Project list view settings	This command configures displaying columns in the **Projects** tab. This command opens the **Projects list view settings** dialog box with checkboxes corresponding to displayed or hidden columns:  When you are working with large projects, displaying the **Size** column might slow down work of **Terminal**, so this column is not displayed by default.
Refresh list of projects	This command updates the displayed list of projects in the server user's directory and opens the Project tab (if it isn't opened). You can also use a **F5** hot keys to execute this command.

Menu item	Description
	Updating the displayed list of projects can be a time-consuming operation (for example if the server directory contains many projects or if the projects consume a large amount of disk space). When waiting the list of projects from **Solver-Agent**, the mouse pointer changes its status to "Busy" and no operation with Terminal can be done during this updating.

The **Projects** menu can be opened from the list of menus or as a context menu on the **Projects** tab.

7.4.3.3 Menu Solvers

Menu item	Description	Hot key
Run new solver	Runs a new **Solver** (the **Solver running** dialog box will open).	**Ctrl+R**
Kill solver	Terminates the selected **Solver**.	
Attach viewer	Starts **Viewer** and connect it to the selected **Solver** (this command is available when the **Solver** is running with a project loaded on it).	
Solvers list view settings	Configures a set of displayed columns in the **Solvers** tab. This opens the **Solvers list view settings** dialog box where you can select the columns, which will be displayed: 	
Refresh list of active solvers	Updates information about active **Solvers** and opens the Solvers tab (if it isn't already opened).	**F5**

Commands of the **Solvers** menu are also available from the context menu on the "Solvers" tab.

7.4.3.4 Menu Licenses

Menu item	Description
Get registration info	Getting the registration information
Register new license	Registering a new license
Get license info	Getting information about licenses
Get license statistics	Getting statistics about use of licenses
Show license monitor	Getting information about the captured license

7.4.4 Interoperation between Terminal and Solver and Solver-Agent

There are two methods of managing projects in **Terminal**:

1. manually
2. using the queue of projects for calculation

7.4.4.1 Manual starting and stopping project calculation from Terminal

Launching a project on computation

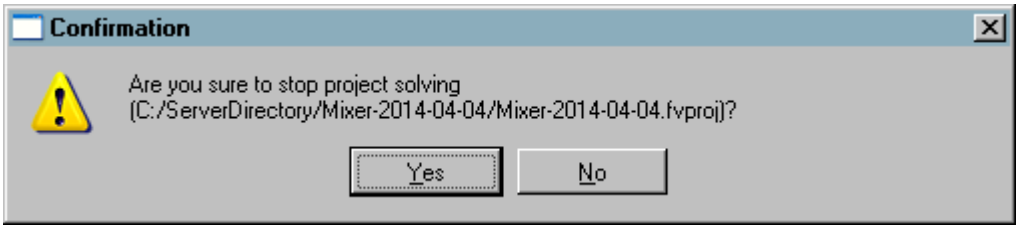
In order to run the project on computation, follow the steps:

- Run **Solver**
- Download project **Solver**:
  - menu **Projects** update the list of projects (press **F5** on the keyboard)
  - in the **Projects** tab select the **Project**, which should be sent to the computation
  - select **Load project to solver** from the **Projects** menu
- Start calculation

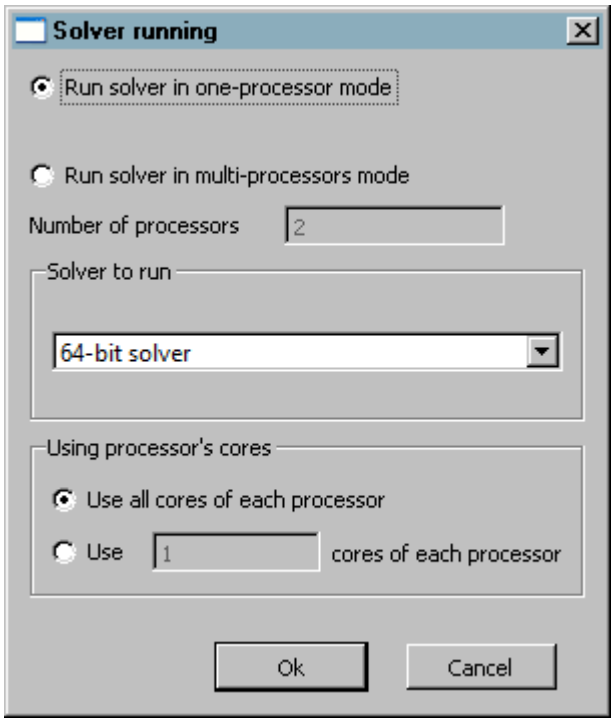
Stop a project

In order to stop a project, launched manually, follow the steps:

- in the **Projects** tab select the project
- select **Stop project solving** from the **Projects** menu.
- confirm your decision to stop the project in the dialog box, which opens (**Are you sure to stop project solving ... ?**):



7.4.4.1.1 Window "Solver running"



The **Solver running** dialog box is used to run **Solver** with the specified parameters. The **Solver running** dialog box is opened the **Solvers** >**Start a new solver** menu.

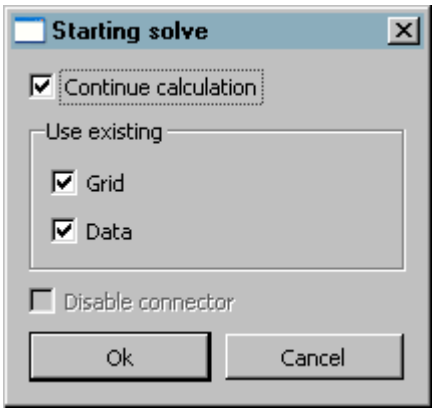
The **Solver running** dialog box contains the following interface elements:

Run solver in one-processor mode	**Solver** runs on a single processor
Run solver in multi-processors mode	**Solver** runs on multiple processors[1)]
Number of processors	The number of processors on which to run the **Solver** (field is available only when you run**Solver**in multiprocessor mode).
Solver to run	The command line used to start the **Solver**[1)]
Using processor's cores	
Use all cores of each processor	**Solver** runs on each core of each processor used
Use *(N)* cores of each processor	Run **Solver** with specified number of cores of each processor used

Note:

[1)] See Dialog box "Select solver".

7.4.4.1.2 Window "Starting solve"



Starting solve box is designed to run on the calculation of the project loaded on the **Solver**, with the specified parameters.

Starting solve window called by the menu **Projects > Start project solving**.

The **Starting solve** dialog box:

Checkbox	Actions when the checkbox is selected	Actions when the checkbox is unselected
Continue calculation	continuation of the calculation with the following parameters:	start the calculation again
Use existing		
Grid	without adjustment of the computational grid	rebuild the computational grid
Data	use the current estimates	use the initial data (the initial conditions)
Disable connector (this checkbox is available only for joint simulations, see Connectors)	Start the computation without connection to an external software	Start the computation when an external software connects

7.4.4.2 Queue of projects for calculation

Queue of projects for calculation is **Solver-Agent**'s functionality that allows forming a list of of projects, which will be automatically run for sequential computation.

Suppose that we have several projects that are ready for computation. Their calculation can be done in the following steps:

1. Load a project on a **Solver**.
2. Run the computation.
3. Examine periodically if the calculation has stopped.
4. Once the calculation stopped, save its results and unload the **Solver**.
5. Repeat previous steps for the next project until calculations of all projects are made.

If calculation of some project finishes outside of user's working hours then user will not able to start another project so computational resources will be idle.

An alternative approach is use of queue of projects. In order to do this, the user should open **Terminal** and create the queue of projects. Each project placed into the queue calculates sequentially until its stopping conditions activate. Then the next project from the queue starts.

How the queue of projects works

Initially the user should prepare the projects that will be calculated and load them into the server directory.

Using **Terminal**, the projects are placed into the queue with information about how many cores will be used to run which projects.

Solvers will start automatically until they exhaust the allocated resources (until **Solvers** consume all available compute cores). Then projects from the queue will be automatically loaded on the running **Solvers**.

If a stopping condition activates during the project's calculation, then, applying the commands specified in the command file, the results of the computation will be saved and the **Solvers**, which calculated the project, will finish its work and will free the computational resources so they will be available for the next project from the queue.

Tuning the queue of projects

For the correct work of the queue of projects you have to specify its parameters in the configuration file of Solver-Agent:

- **MaxThreadNum**, this parameter specifies how many cores can be used by all **Solvers** together that are started from the queue of projects. The next project from the queue will not start if the number of cores, requested (in the **Adding project to the projects queue** dialog box) for its running, exceeds the number, which is calculated as difference (**MaxThreadNum** - *Ncurrent*), where *Ncurrent* is the number of cores used by **Solvers** that will be operating at that time moment. If the **MaxThreadNum** parameter is not specified, the number of available cores will be determined automatically.
- **NumaThreadNum**, this parameter allows you to specify explicitly manually the available number of cores per processor in the case when **Solver** is started with the option **Use all cores of each processor** enabled. When the **NumaThreadNum** parameter is not set, the number of cores per processor will be set automatically (this is the system value **numa**). Use of the **NumaThreadNum** parameter allows you to compulsory manually specify the number of cores per processor.
  Example: Let's assume that each processor has 8 cores and the **NumaThreadNum** parameter is not specified. Then, when **Solver** is started with the option **Use all cores of each processor** enabled, the 1x8 solver will run. If, in the same situation, you specify **NumaThreadNum=5**, then the 1x5 solver will run.

The queue system of **Solver-Agent** can only operate when the **Solver**, which is started from it, uses cores of one computer.
So the queue system might operate incorrectly on clusters due to cores occupied by the **Solver** may belong to different computers.

Placing projects into the queue

To place projects into the queue, follow the steps:

Step	Actions
1	Prepare several projects with stopping condition.
2	Place directories with the projects into the server directory.
3	For each project, prepare a command file that will control the calculation. It is recommended to use the standard name for the command file, `command.txt`. Each command file is to be placed into the directory of its project, i.e. near the `*.fvproj` file.
4	In the **Projects** tab, update the list of projects (use the menu command **Projects** > **Refresh list of projects** or press the **F5** key on your keyboard).
5	In the tab **Projects** select the first project, which will be placed into the queue.
6	Apply the the menu command **Projects** > **Put to queue**. The **Adding project to the projects queue** dialog box will open, specify the required parameters there.
7	Repeat steps 5-6 for other projects that you wish to place into the queue.

Deleting a project from the queue

To delete a project from the queue, follow the steps:

Step	Actions
1	Select the project in the tab **Projects**.

Step	Actions
2	Use the menu command **Projects** > **Delete from queue**.

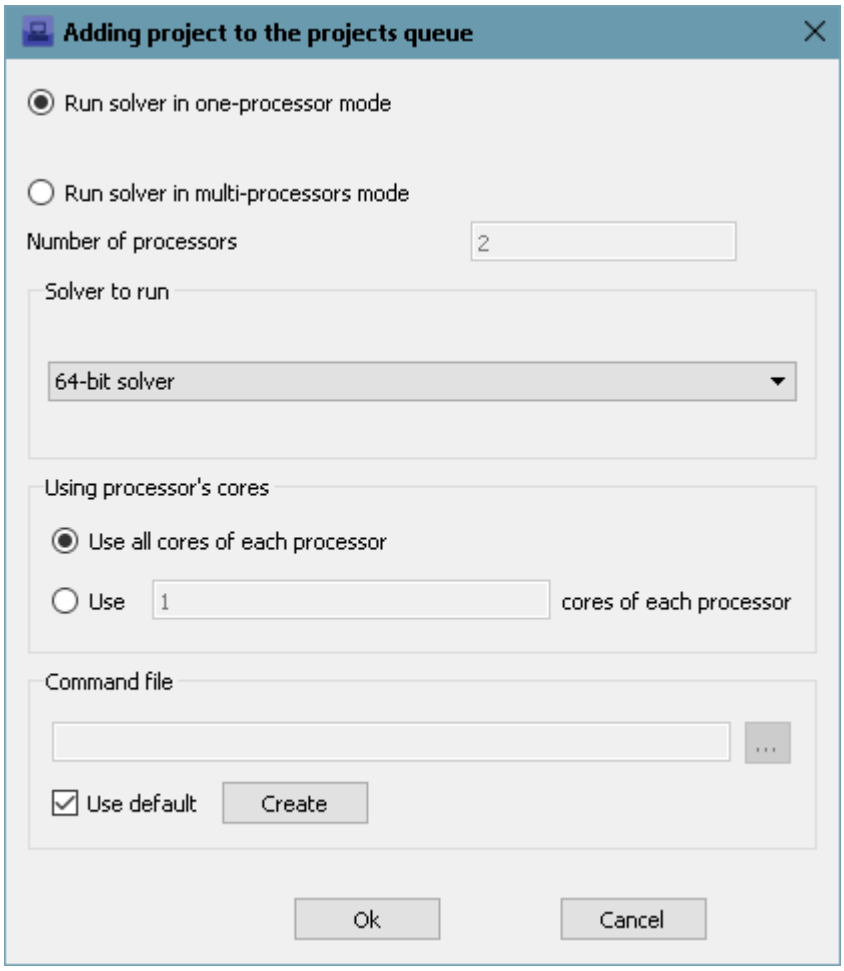
Viewing the queue

To view the queue of projects, use the menu command **Projects** > **View queue**.

Clearing the queue

To clear the queue of projects, use the menu command **Projects** > **Clear queue**. Restart of **Solver-Agent** also causes clearing the queue.

7.4.4.2.1 Dialog box "Adding project to the projects queue"



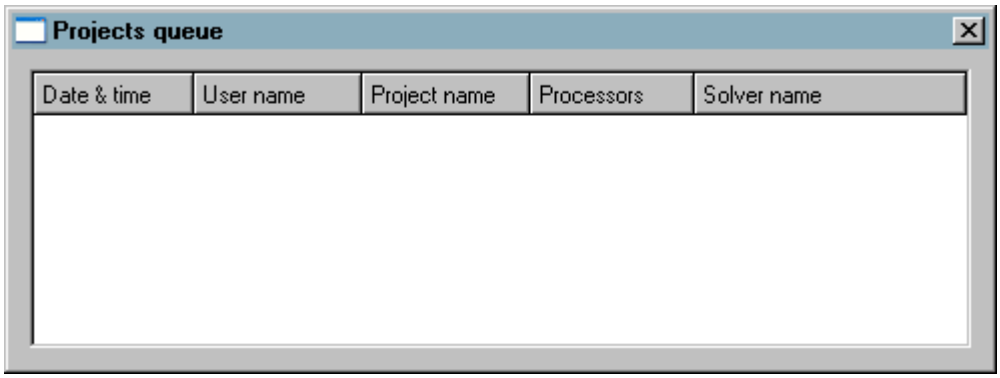
The **Adding project to the projects queue** window is intended to add a **Project** into the queue for computation on **Solver** with specified parameters and specified command file. The **Adding project to the projects queue** window is opened by the menu command **Projects** > **Put to queue**.

The **Adding project to the projects queue** window contains the following user interface elements:

Interface element	Description
Run solver in one-processor mode	See Dialog box "Solver running".
Run solver in multi-processor mode	
Number of processors	
Solver to run	
Using processor's cores	
Use all cores of each processor	

Interface element		Description
	Use *(N)* cores of each processor	
Command file		This specifies a **Command file** for the running **Solver**. See an example of use the **Command file** in the section Automatic replacement of the Moving body's geometry during the computation.
	file selection field	Specify a **Command file** for launching. This field is only available when the **Use default** checkbox (see below) is off.
	Use default	Use as a **Command file** the default file, which has the name **`command.txt`** and locates in the project's directory. Note that you have create this file yourself and save it in the project's directory with the name **`command.txt`**. If the program does not find the **`command.txt`** file in the project's directory, an error message will be output: "**Appending project to queue error: default command file command.txt is absent in the project directory. Please, create it by yourself.**".
	Create	Opening the **Command file creation** dialog box to create a **Solver**'s command file and place it into the server's part of the project. The same actions are done by the **Projects > Create command file** menu command.

7.4.4.2.2 Dialog box "Projects queue"



The **Projects queue** dialog box is used to view the list of projects, which are waiting for computation in the queue. The **Projects queue** window is opened by the menu command **Projects > View queue**.

The **Projects queue** dialog box displays the following information:

Column	Description
Date & Time	Date and time of setting the project in all
User name	The user has moved the project to the queue
Project name	The project name is placed in the queue
Processors	The number of processors and the number of nuclei, which project will be launched
Solver name	The name of the command line, using the **Solver** which will be launched on the computation

7.4.5 Plugins

Plugins are *Dynamic Link Libraries*, which extend functionality of **Terminal**.

Plugins provides a technology for simplified creation and tuning a *FlowVision* project, which is intended for solving a highly-specialized problem. When such problems are solved, only few parameters are changed in a prearranged project.

A plugin provides a simple interface for specifying a limited number of key parameters of the computation. Routine calculation are made there without use of the interface of **Pre-Postprocessor**. This technology allows you to

reduce resources of the company's computational department and decrease requirements to skills of computing engineers.

A project is created based on a template project as its simple copy. Actually the plugin is a combination of a template project and an expansion module (`dll`) for the **Terminal** module. A plugin includes a graphical user interface, which allows you to specify changeable parameters of the project. A specific layout of the graphical user interface and the list of parameters depend on the task's specifics.

The procedure of creating a project in **Terminal** consists of the following steps:

- Creating a project based on a template project (by copying).
- Loading the project on **Solver**.
- Specifying parameters of the project using the plugin's graphical user interface.
- Saving the project on a disk.

After this, the project's parameters can be viewed and/or edited in the plugin's user interface at any time, including the time after finishing the project's computation.

The project's computation is started in the usual manner. If the project has been created using a plugin, **Terminal** will display the plugin's name in the **Projects** tab (in the **Plugin** column).

Plugins can be developed on a by-order basis.

Create a project using a plugin

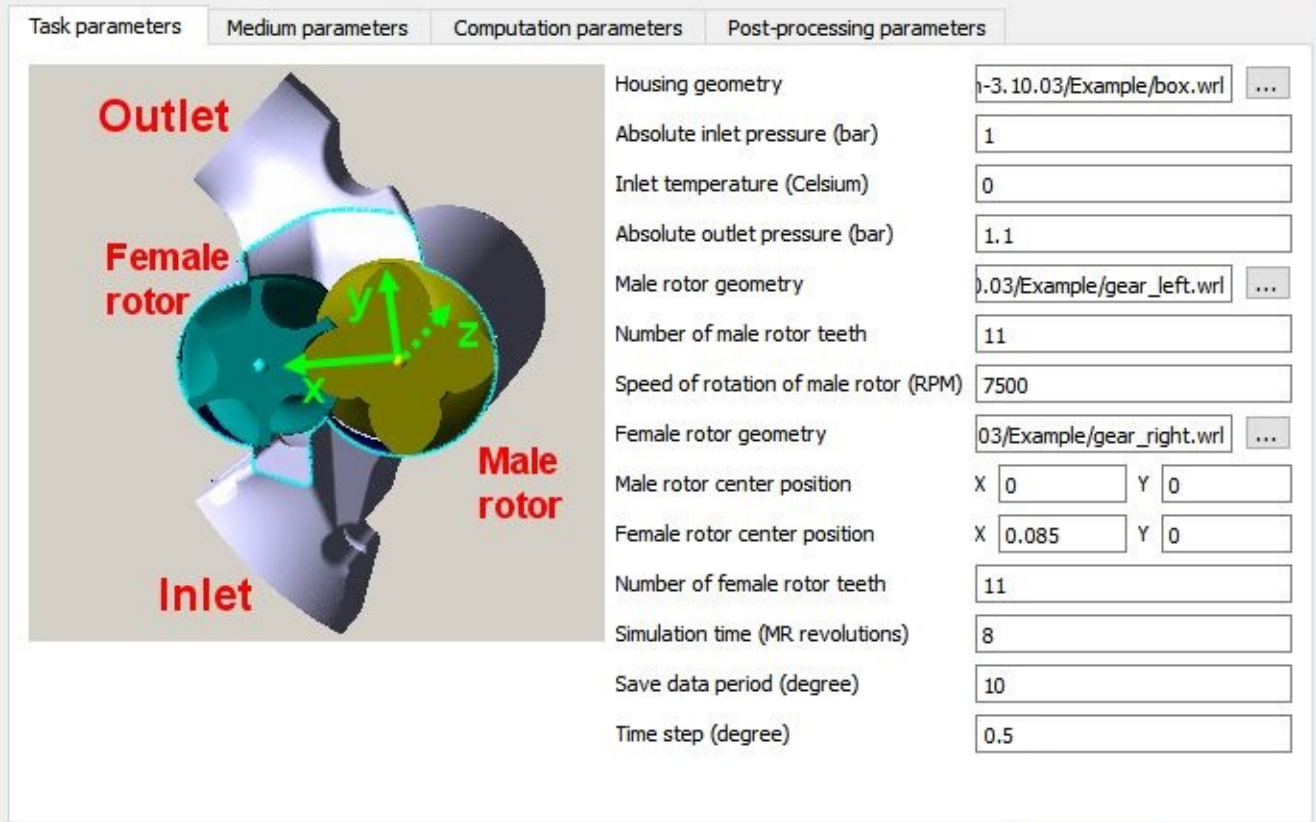
Step	Actions
1	Copy files of the plugin into the directory of **Terminal**.
2	Start **Terminal**.
3	Start a new **Solver** (apply the menu command **Solvers > Run new solver** or the command **Run new solver** from the context menu on the tab **Solvers**).
4	Create a new project applying the menu command **Projects > Create project by plugin** or the command **Create project by plugin** from the context menu on the tab **Projects**.
5	Fill in the fields: • Enter the name of the project. • Select a plugin, which will be used, from the list. • Select a **Solver** from the list (you can select any free **Solver**) and click the **OK** button. • Click **Yes** in the window, which prompts you to update the list of projects. A new project with the specified name will appear in the list of projects of **Terminal**. The **Plugin** will display the name of the used plugin.
6	Select the created project from the list and upload it to **Solver**: • Apply the command for uploading a project to **Solver** (apply menu command **Projects > Load project to solver** or the command **Load project to solver** from the context menu on the **Projects** tab). • Select a free **Solver** from the list **Solvers** in a dialog box and then click **OK**. • Click **Yes** in the dialog box, which prompts you to update the project's status in the **Terminal**'s window.
7	Select the project in the project list, and change its parameters: • Call the plugin to change parameters of the project (apply command **Projects > Change project parameters** or the command **Change project parameters** from the context menu on the **Projects** tab). • Set the problem's parameters manually or load them from a file with extension `.pcs`, that store the default values. Click **"Ok"**. • Click **Yes** in the dialog box, which prompts you to update the project's status in the **Terminal**'s window.

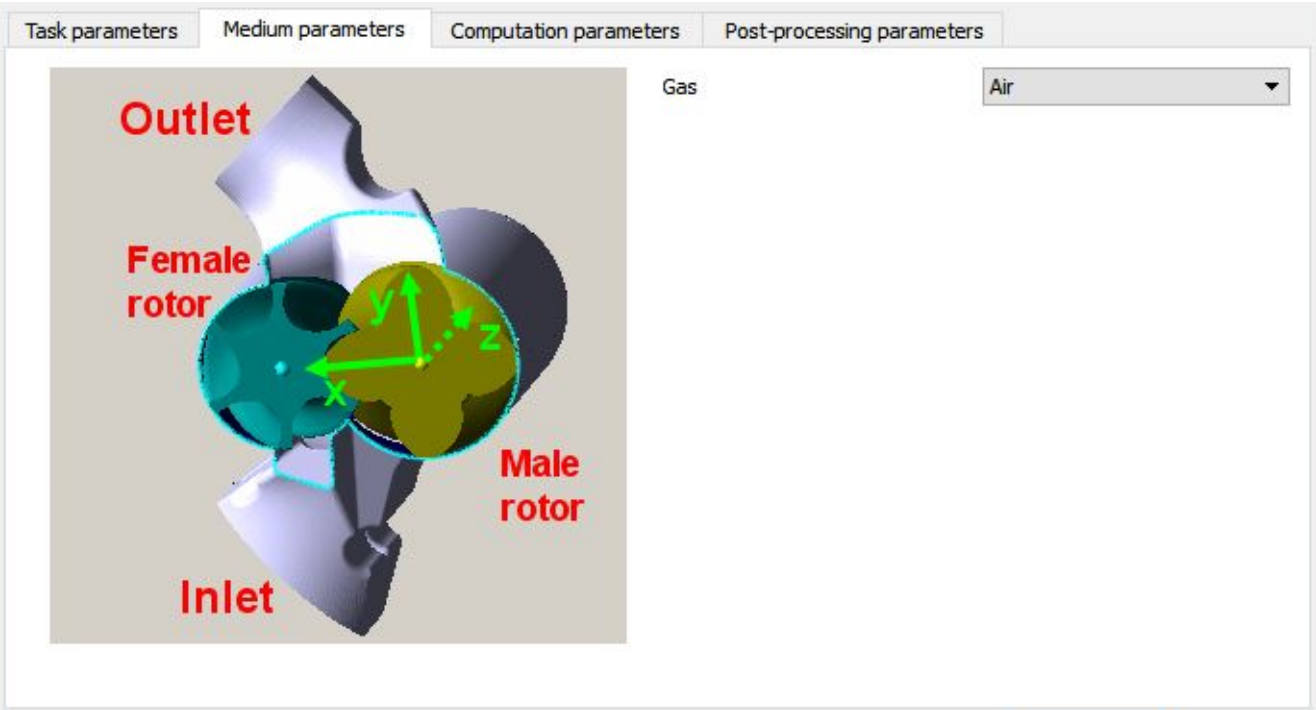
Example: user interface of the Screw Compressor plugin

On the illustrations below you can see the user interface of the **Screw Compressor** plugin, which simulates a screw compressor.

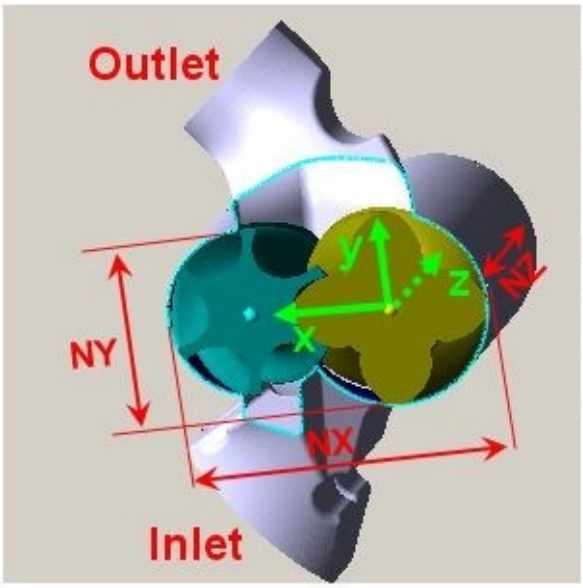
Project parameters are grouped in the following tabs:

- **Task parameters** contains parameters of the problem setting (geometry of housing and rotors, boundary conditions, simulation time, step, save data period).
- **Medium parameters** contains parameters of the simulated medium.
- **Computation parameters** contains parameters of the computational grid and its adaptation, limiters.
- **Post-processing parameters** contains parameters of planes, on which results of the computation are calculated.


Task parameters
Medium parameters
Computation parameters
Post-processing parameters
Outlet
Female rotor
Male rotor
Inlet
y
z
x
Housing geometry
-3.10.03/Example/box.wrl
Absolute inlet pressure (bar)
1
Inlet temperature (Celsium)
0
Absolute outlet pressure (bar)
1.1
Male rotor geometry
.03/Example/gear_left.wrl
Number of male rotor teeth
11
Speed of rotation of male rotor (RPM)
7500
Female rotor geometry
03/Example/gear_right.wrl
Male rotor center position
X 0 Y 0
Female rotor center position
X 0.085 Y 0
Number of female rotor teeth
11
Simulation time (MR revolutions)
8
Save data period (degree)
10
Time step (degree)
0.5


Task parameters
Medium parameters
Computation parameters
Post-processing parameters
Outlet
Female rotor
Male rotor
Inlet
y
z
x
Gas
Air

Task parameters | Medium parameters | Computation parameters | Post-processing parameters



The number of cells in rotor compartment

X direction 20

Y direction 10

Z direction 10

Aspect ratio of cells outside rotor compartment

X direction 1.2

Y direction 1.2

Adaptation level in rotor compartment 0

Adaptation level over housing boundary 0

Temperature limiters

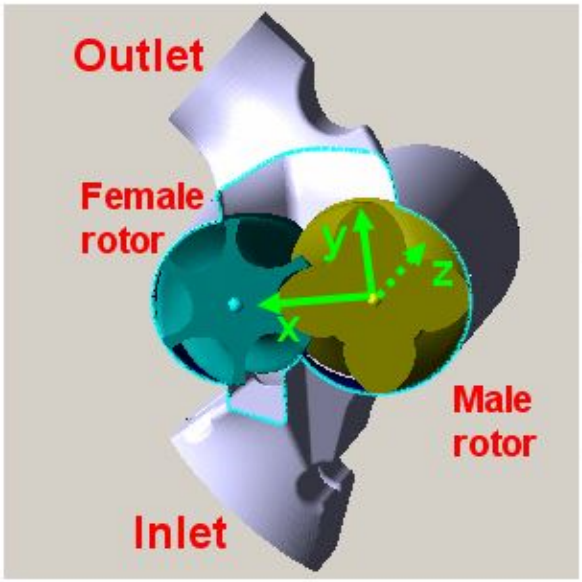
Min 100 Max 1000

Pressure limiters

Min 10000 Max 1000000

Maximum speed limiter 1000

Task parameters | Medium parameters | Computation parameters | Post-processing parameters



Post-processing plane for density

Position

X 0 Y -0.1 Z 0

Normal

X 1 Y 0 Z 0

Post-processing plane for pressure out and temperature out

Position

X -0.1 Y 0 Z 0

Normal

X 0 Y 0 Z -2

7.5 Retranslator

Retranslator (`FvConnect`) is a module intended for transmitting data between modules of *FlowVision* installed on computers without a direct network connection, but connected through a third computer. **Retranslator** is installed on a computer that has a direct network connection with both computers where the modules are installed. **Retranslator** is controlled through its config file.

Retranslator is a software proxy server that transfers traffic from **Solvers** to **License manager** and to client modules if those modules are located in a network not accessible from the computation modules.

As a rule, the head node of a cluster is equipped with Internet access. **Retranslator** must be installed on such a node.

In **Retranslator**'s settings, each line defines one the method of routing the traffic between modules is indicated in every new line.

To set up traffic routing between **Solver** and client modules, the settings of the **Solver-Agent** must be edited as well.

Routing will require a starting the **Retranslator** module.

See also:

About configuring **Retranslator** see sections:

- Configuration file of Retranslator (FvConnect.cfg)
- Direct re-translation
- Re-translation via Solver-Agent

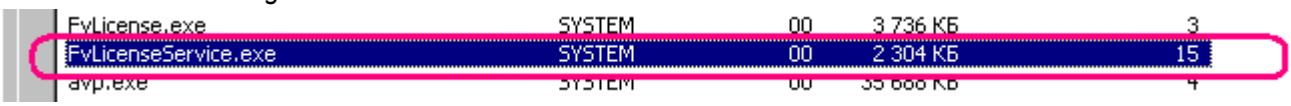
7.6 License Manager

License Manager is a server application, which manages the licenses.

License Manager is launched at start of the operating system file, from file **`FvLicense.exe`** (for *Windows*) or **`FvLicense`** (for *Linux*).

License Manager has no user interface. To make sure that it is running, use the *Task Manager Windows* (check for line **`FvLicense.exe`** in the process list):

It is possible to configure the automatic start **License Manager** each time you start *Windows*, which is used to program **`FvLicenseService.exe`**, working as a service of *Windows*. This service, when set up its application, as shown in *Task Manager Windows*:



License Manager settings are stored in the configuration file **`FvLicense.cfg`**.

Starting the **License Manager** is installed as a service (service *Windows*) **`FvLicenseService.exe`**, it is done automatically every time you boot your computer.

When necessary, **License Manager** can be started manually using the **Start** button of *Windows* (**Start > Programs > FlowVisionLM > License Manager**), or run it executable **`(FvLicense.exe`** *Windows* or **`FvLicense`** on *Linux*).

Location of License Manager

License Manager is to be installed on the computer to which all working components *FlowVision* be permanent access (over the network, or it can be installed on the same computer on which you installed the other components *FlowVision).*

When using a cluster, **License Manager** can be installed on any computer that has access to the cluster. Usually **License Manager** installed on the host PC cluster.

If you have more, not connected by a network of computers (for example, project preparation and visualization of the results produced on the same computer, and computing - on the other), then in each such computer must be set to **License Manager** and registered its own license.

Installation of License Manager

See sections:

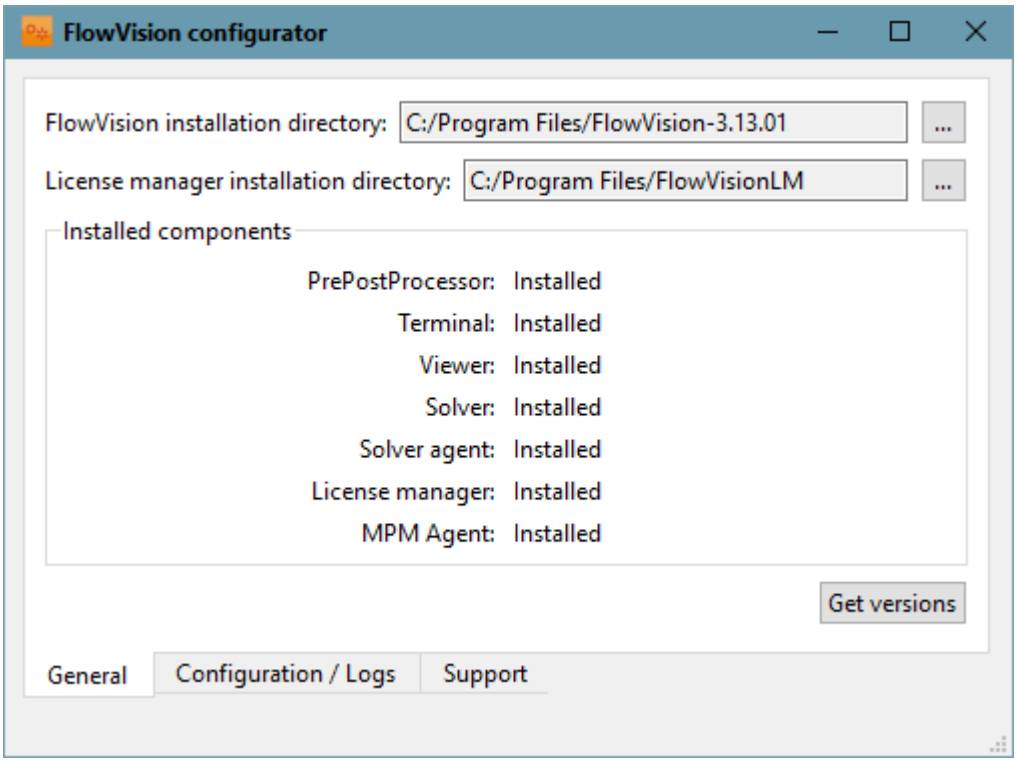
- Installation of FlowVision
- Installation on Windows
- Installing License Manager on Windows
- Installation on Linux
- Installing the License Manager under Linux (in graphical mode)
- Installing the License Manager under Linux (in text mode)
- Automatic installation

Compatibility with older versions of License Manager

FlowVision 3.xx can not be used with the **License Manager** from *FlowVision 2.xx*. For normal operation of the two versions *FlowVision* replace the **License Manager** *FlowVision 2.xx* on the **License Manager** *FlowVision* 3.08.

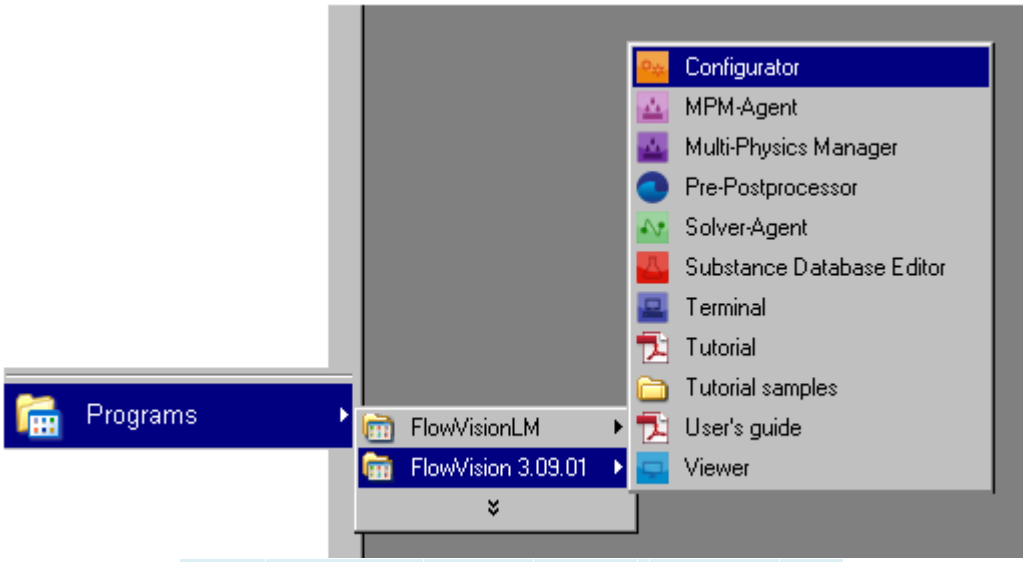
7.7 Configurator

Configurator is a module designed to configure the modules.



Configurator's window

You can call **Configurator** in *Windows* from the **Start button > Programs > FlowVision *(version number)*** menu.



Calling **Configurator** from the *Windows*' **Programs** menu

Description of **Configurator** see in the section Installation, configuration, administration > Settings defined in Configurator.

7.8 Substance Database Editor

Substances Database is a database containing information about properties of substances in various states of aggregation. A **Substance Database** can be either *standard* or *user*.

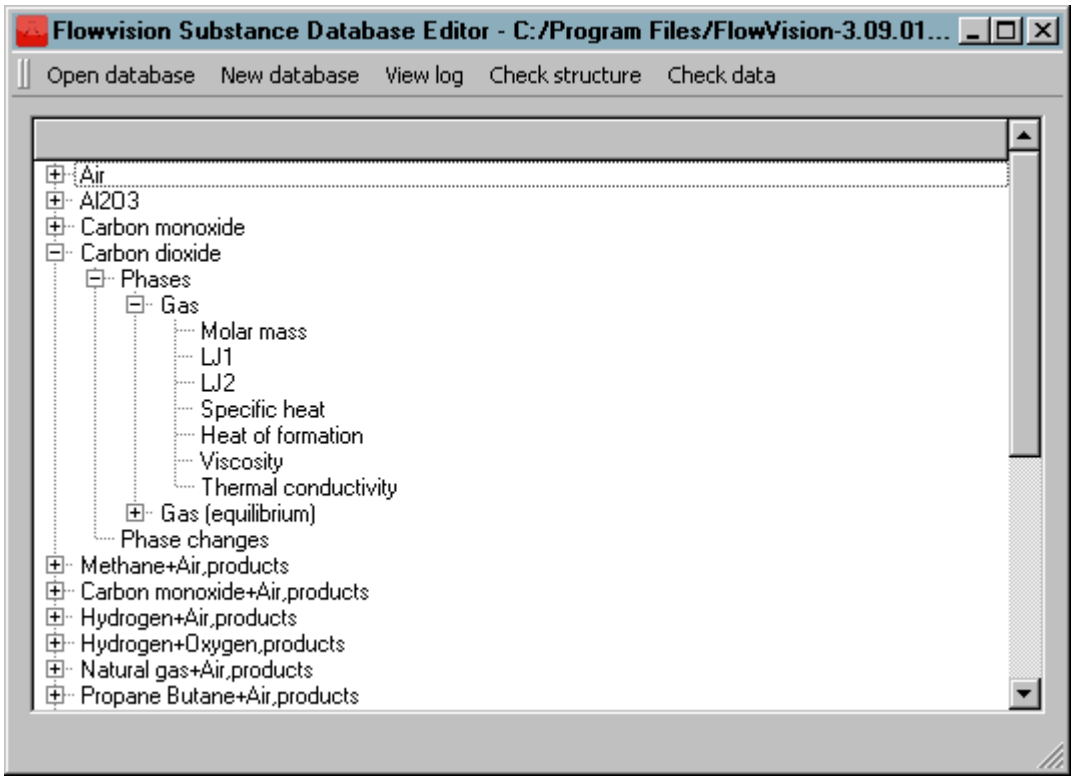
Substance Database Editor is a separate module required for editing a user **Substance Database**. **Substance Database Editor** allows you to create and delete **Substances**, as well as edit their properties.

Standard Substance Database	User Substance Database
comes with *FlowVision*	is created by the user as a copy of another **Substance Database** or in the **Substance Database Editor**
always locates in the `Database` subdirectory, which locates in the same directory as **Pre-Postprocessor** does	can locate in any place, which is accessible to **Pre-Postprocessor**
cannot be edited	is editable

The standard *FlowVision*'s **Substance Database** is created based on generally available reference books and third-party thermodynamical software. The **Substance Database** contains minimal set of the must used substances for majority of cases. Most of properties are set for normal pressure and for a narrow range of temperatures.

When you use **Substances** from the standard **Substance Database**, you have to attend to their properties, as the properties might differ from properties of your **Substance** of the same name due to admixtures/impurities or different operating range of temperatures and pressures.

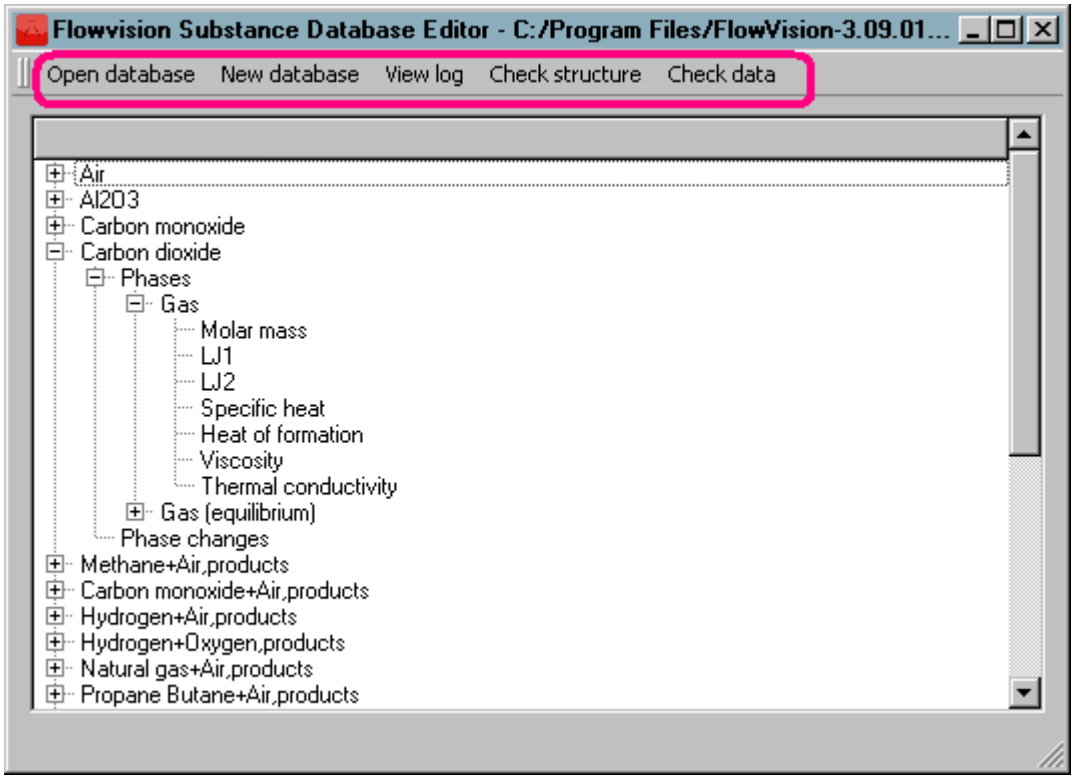
When you need, you can create your own (user) **Substance Database** with required properties in the required range of pressures/temperatures.



Window of the **Substance Database Editor**

You can start **Substance Database Editor** in*Windows* from the menu **Start button > Programs > FlowVision (version number)**.

7.8.1 Menu of the Substance Database Editor

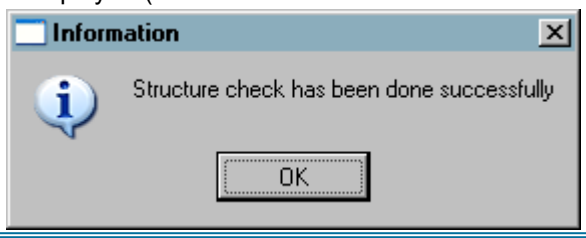


Menu of the **Substance Database Editor** contains the following commands:

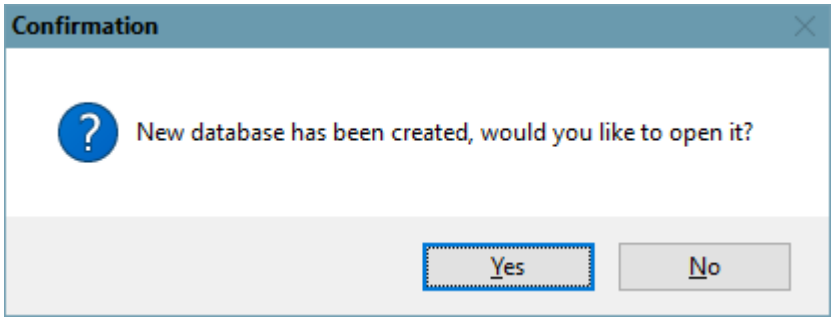
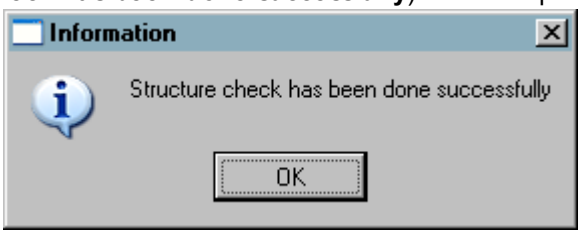
Command	Description
Open database	Opens an existing **Substance Database**
New database	Creates an new empty user **Substance Database**
View log	Displays the log file of **Substance Database Editor** This log does not locate in the user directory (where logs of *FlowVision* modules locate) but locates in the directory of the **Substance Database** itself.
Check structure	Checks the structure of a **Substance Database**
Check data	Checking the properties of a **Substance Database**

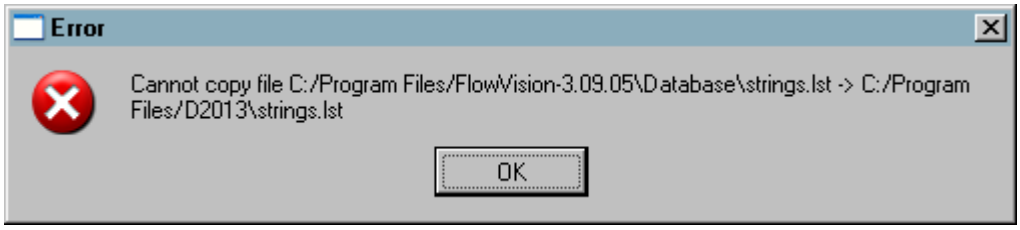
Opening an existing Substance Database

Step	Actions
1	Apply the **Open database** menu command. A standard operating system's window for access to directories will open.
2	In the window for access to directories select a directory, where the required **Substance Database** locates (for example, the `Database` subdirectory, which contains the **Standard Substance Database** and locates in the same directory as **Pre-Postprocessor** does).
3	The program will read the **Substance Database**, which locates in the selected directory. If the **Standard Substance Database** has been selected, the warning that the database is read-only will be displayed ("**Standard database is read-only**"):

Step	Actions
	 Then the structure of the **Substance Database** will be checked for correctness. If the structure is correct, a message will be displayed ("**Structure check has been done successfully**"): 
4	If a directory is selected that does not contain a **Substance Database**, a warning will be displayed ("**Directory doesn't contain a database**"): 

Creation a new Substance Database

Step	Actions
1	Apply the **New database** menu command. A standard operation system's dialog box will open with access to folders.
2	Select a folder to which writing of files is enabled. The program will output a message about successful creation of a new **Substance Database** in the selected folder (**New database has been created, would you like to open it?**) prompting you to open this **Substance Database**:  Click **Yes**.
3	Then the program checks the structure of the **Substance Database**; when the structure is correct, a message (**Structure check has been done successfully**) will be output:  Click **OK**.

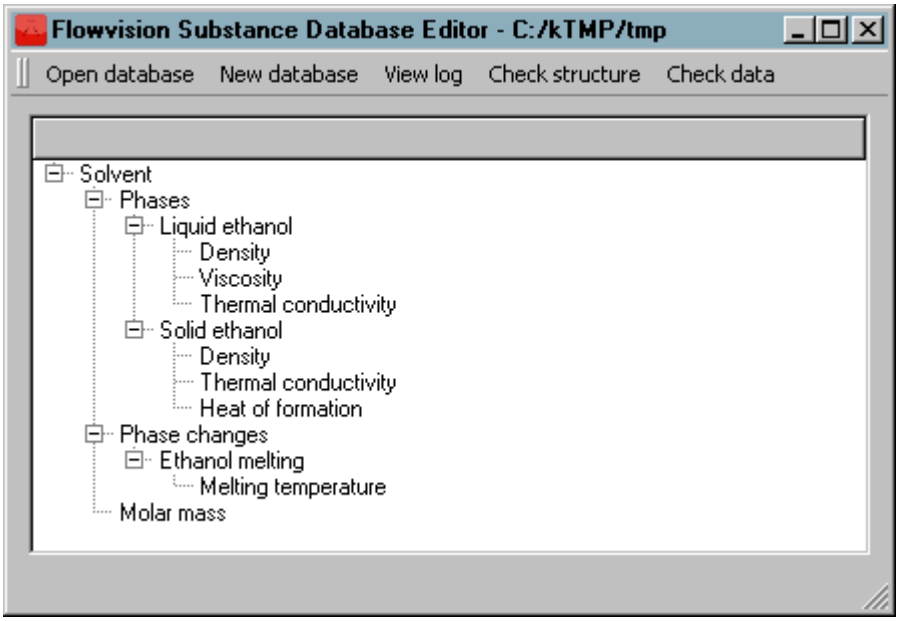
	At attempt of create the **Substance Database** in a folder, which already contains another **Substance Database** and/or if there are problems with writing files, an error message (**Cannot copy file ... -> ...**) will be output:  Click **OK** and try to create **Substance Database** in another folder.
4	Create in the new **Substance Database** the required **Substances** and specify their **Phases** and properties as described in in the section Defining a Substance, its Phases and properties). To create the first **Substance** in an empty **Substance Database**, right-click in the empty window of the **Substance Database Editor** and select **Create substance** from the context menu that opens.

7.8.2 Defining a Substance, its phases and properties

Definition of a **Substance** in the **Substance Database Editor** has the following structure:

Substance identifies a substance, which may consist of several phases			
	The **Phases** folder contains definitions of **Phases** of the **Substance**. Each **Phase** corresponds to some aggregative state of the **Substance**.		
		Phase is a folder corresponding to one of phases of the **Substance**.	
			Property of a phase
			Property of a phase
			...
		Phase is similar for other phases of the **Substance** (if any)	
			Property of a phase
		...	**...**
	The **Phase changes** folder contains definitions of phase changes between **Phases** of the **Substance** (for example, melting, boiling, sublimation) ⚠ Functionality of **Phase changes** is not supported in the current version of *FlowVision*		
		A ***Phase change*** is a node, which corresponds to one of the phase changes (it is defined by two **Phases**)	
			Property of the phase change
		...	
	Name of a common property is a node corresponding to one of properties of a **Substance** that is common for all **Phases** of the **Substance** (optional)		
	Name of a common property is similar for other common properties of a **Substance** (if any) **...**		
	...		

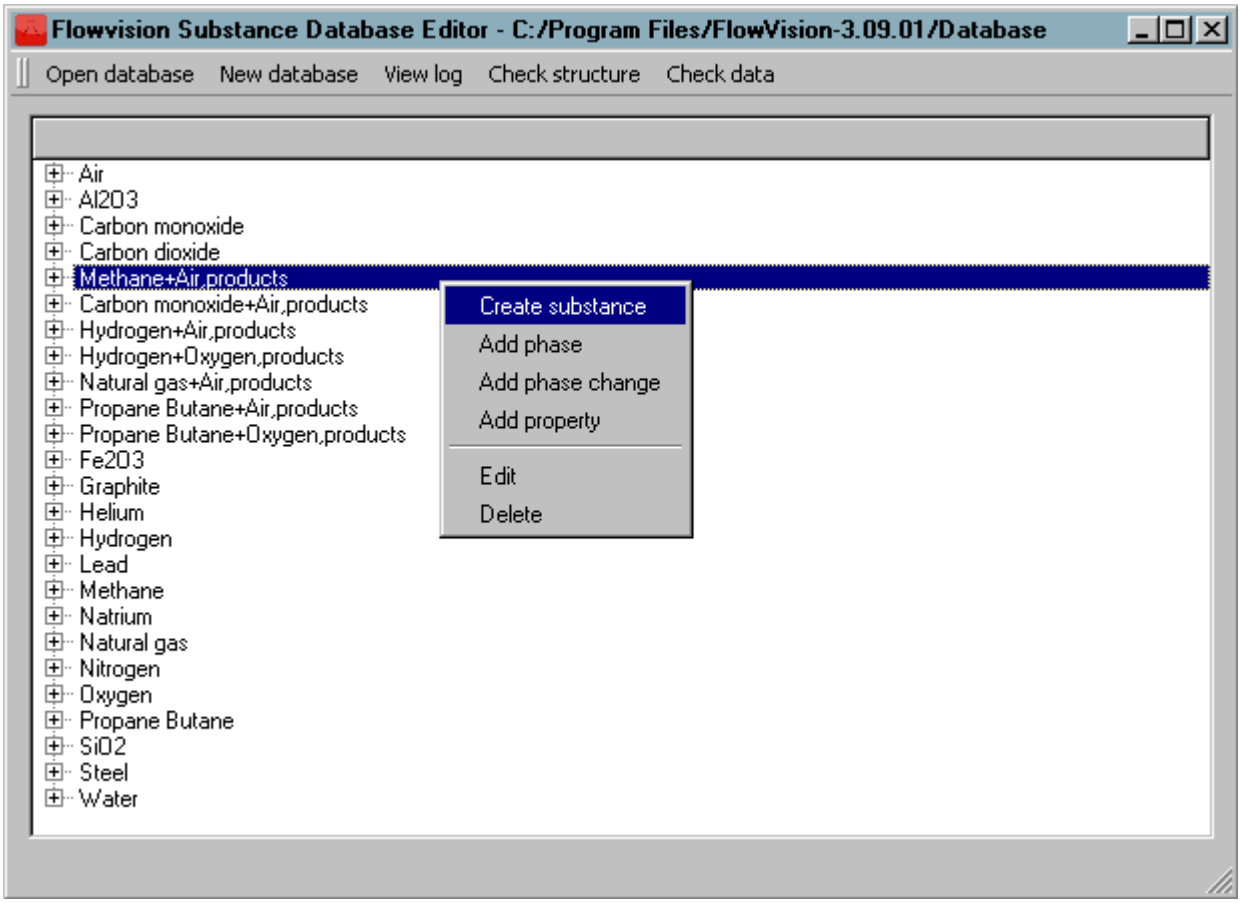
This structure is displayed in the window of **Substance Database Editor**:



A property can be as common to the whole substance, and owned by only one **Phase**. If the material is the same and the property is specified in the list, and the selected **Phase**, the values of the properties in **Phase** have a higher priority.

If the editor **base Substances** set and enthalpy and specific heat, the **Pre-Postprocessor** defined in the **Base Substances** specific heat capacity is ignored. Specific heat of a substance in **Pre-Postprocessor** in this case is calculated using a predetermined enthalpy.

Context menu of a Substance



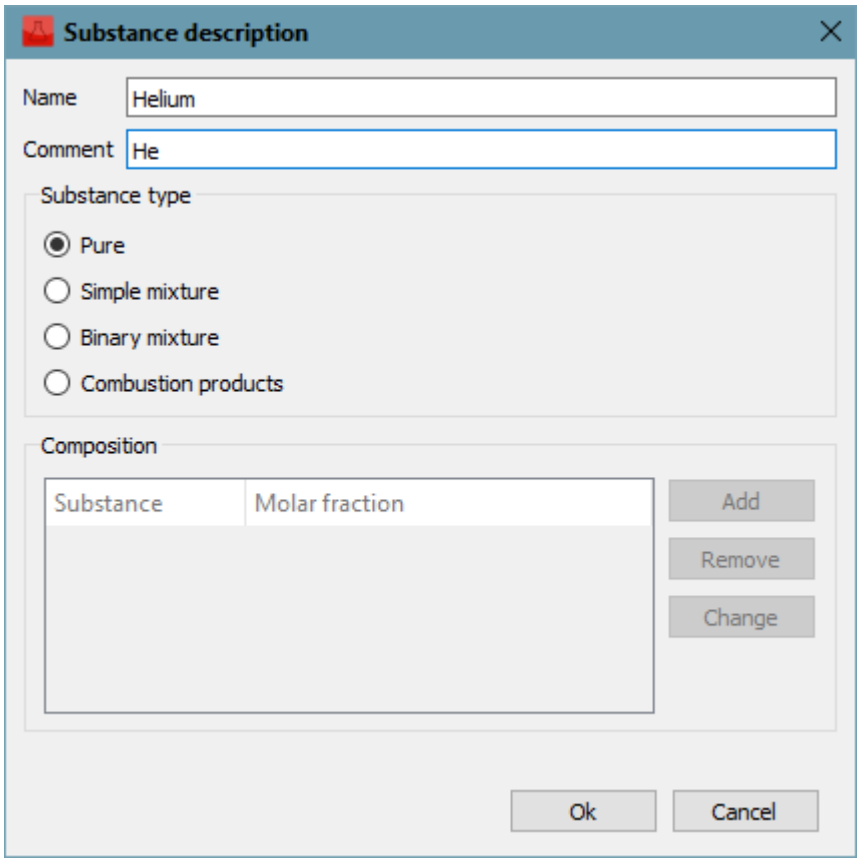
Context menu of a **Substance** contains the following commands:

Command	Description
Create substance [1)]	Creation of a new **Substance**.

	The **Substance description** dialog box will open where you specify the name, a short description (comment), and type of the new **Substance**. After creation of the **Substance** you can create its **Phases**, **Phase changes**[2] and common properties.
Add phase	Creation of a new **Phase** for the selected **Substance** (you define the name and the aggregative state of the **Phase** in the **Phase description** dialog box). A new created **Phase** should be later filled with properties.
Add phase change[2]	Creating a new phase change, determined by a pair of two different **Phases** (in the **Phase change description** set the name of the phase change and two **Phases**).
Add property	Add to definition of the **Substance** some property, which is common for all **Phases** of the **Substance** (it is defined in the **Add new property** dialog box, see its description in the appropriate subsection).
Edit	Opens the **Substance description** dialog box of the substance,which can change the name, a short description (comment), type and composition of the **Substance**.
Delete	Deletes the **Substance** from the **Substance Database**. The program will request you to confirm this action.

[1] If the **Substance Database** has no substance, the context menu with this command appears when you right-click in the empty pane of the main **Substance Database Editor** window.
[2] Functionality of **Phase changes** is not supported in the current version of *FlowVision*.

The "Substance description" dialog box

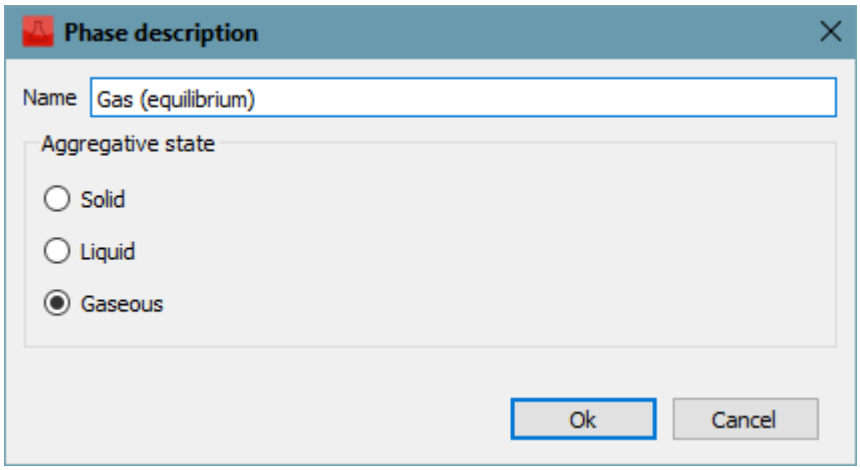


The **Substance description** dialog box has the following elements:

Element	Description
Name	Name of the substance
Comment	Brief description of the substance

Substance type	Type of substance; the following options are possible: • **Pure** -a substance consisting of one component • **Simple mixture** - a substance consisting of several components. **Phase changes** are specified for one (the dominant) component. • **Binary mixture** - a substance consisting of two components. **Phase changes** are specified by phase diagrams. • **Combustion products** - a substance consisting of several components. Contains additional information on the substance of **fuel** and **oxidizer** (for a simplified model of combustion).
Composition	Set of components constituting the substance, and their mole fractions. It is specified for the following **Substance types**: • **Simple mixture** • **Binary mixture** • **Combustion products** ***Note:*** The sum of molar fractions has to be equal to 1.

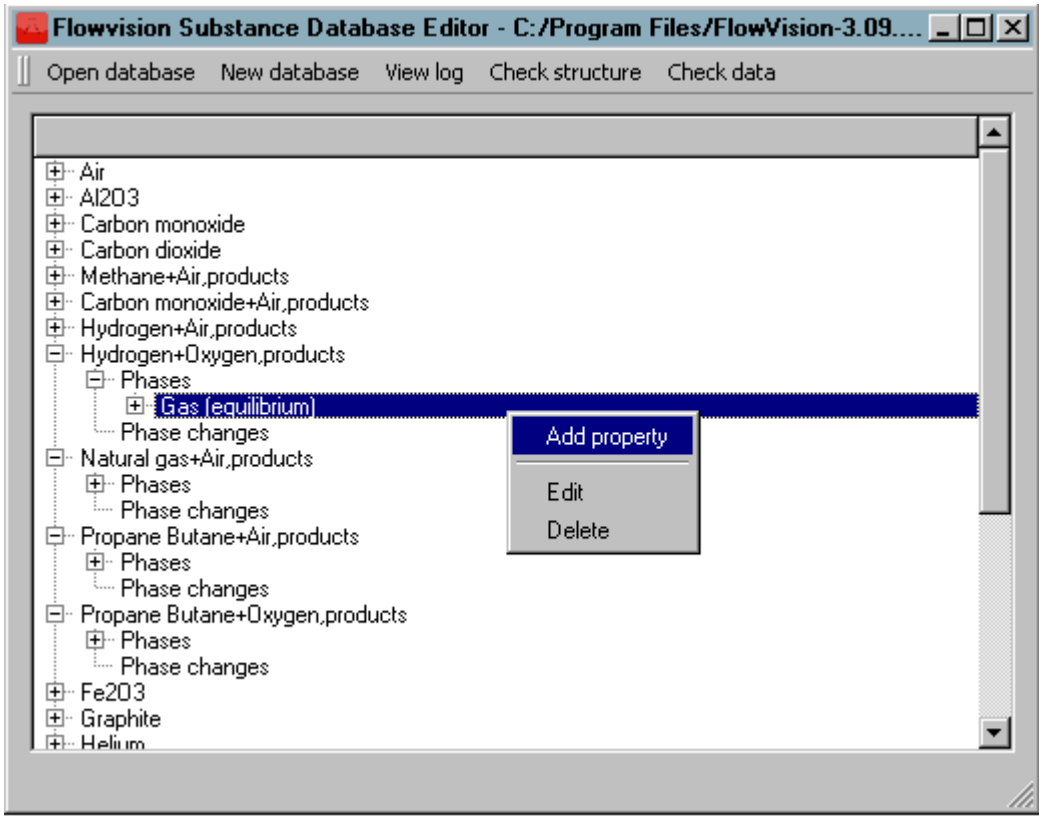
The "Phase description" dialog box



The **Phase description** dialog box allows you to specify the name and state of aggregation of a given phase. Possible options of **Aggregative state** are:

- **Solid**
- **Liquid**
- **Gaseous**

Context menu of a Phase

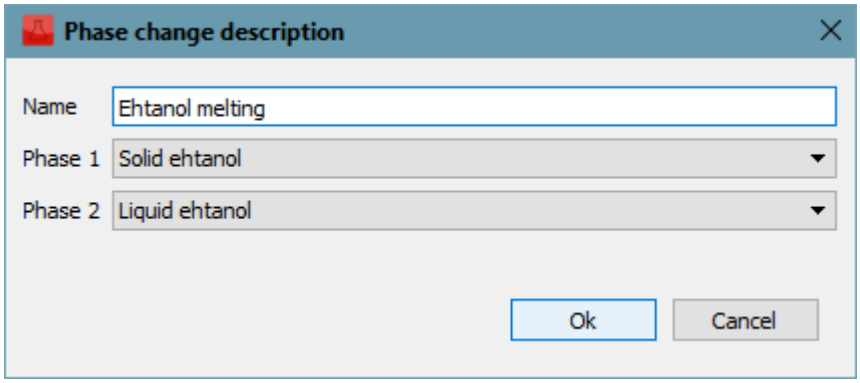


Context menu of a **Phase** contains the following commands:

Command	Description
Add property	Adding into a **Phase** a new property related only to this **Phase**, and not being a common property of all **Phases** of the **Substance**. This action is carried out using the **Add new property** dialog box, see its description in the appropriate section. You can then specify numerical parameters of the added property using the **Edit** command from the property's context menu (see an appropriate section below.) For adding a property that is common for all **Phases** of the **Substance**, use the **Add property** command from the context menu of the **Substance**.
Edit	Changing the name and/or the aggregation state of the **Phase** (in the **Phase description** dialog box, see subsection above)
Delete	Delete the **Phase**. The program will request you to confirm this action.

The "Phase change description" dialog box

Functionality of **Phase changes** is not supported in the current version of *FlowVision*.



The **Phase change description** dialog box opens when you:

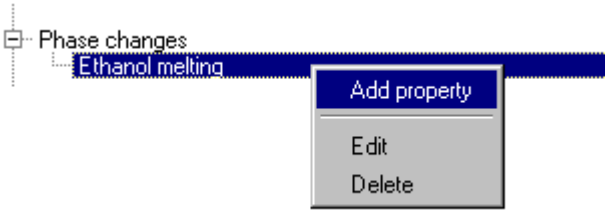
- add a new **Phase change** (by the **Add phase change**command from the context menu of the **Substance**)
- modify an existing **Phase change** (by the **Change** command from the context menu of the **Phase change**)

It contains the following fields:

Field	Description
Name	Name of the **Phase change**
Phase 1	Two different **Phases**, between which the phase change occurs (they are selected from drop-down lists)
Phase 2	

Context menu of a phase change

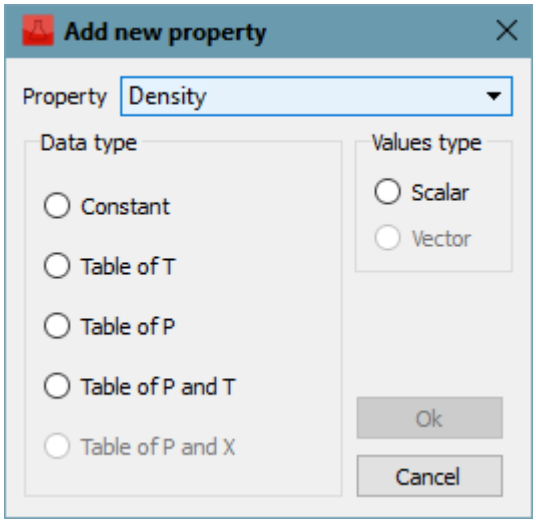
Functionality of **Phase changes** is not supported in the current version of *FlowVision.*



The context menu of a **Phase change** has the following commands:

Command	Description
Add property	Adding a property that characterize the **Phase change**. This action is carried out using the dialog box add a property, see its description in the appropriate section. You can then specify the numerical parameters of the added properties using the **Change** command from its context menu (see an appropriate section below).
Edit	Changing the name and the phase of the **Phase change** using **Phase change** dialog box, see subsection below.
Delete	Deleting the **Phase change**. The program will request you to confirm this action.

The "Add new property" dialog box



The **Add new property** dialog box is used for:

- adding properties, common for all **Phases** of the **Substance**
- adding properties, specific for the selected **Phase** of the **Substance**
- adding properties of a **Phase change**

It contains the following elements:

Property	The choice of properties from the drop-down list. Possible options (but the set of properties available for selection depends on the aggregate state): • **Molar mass** • **Density** • **Viscosity** • **Thermal conductivity** (can be specified as a vector value) • **Specific heat**[**)] • **Enthalpy**[**)] • **Enthalpy of formation** • **Range of solidification** • **Surface tension**
Data type	Data type, the following options: • **Constant** • **Table of T** • **Table of P** • **Table of P and T** • **Table of P and X** (availability of this option depends on the version of the program)
Values type	Value type, the following options: • **Scalar** • **Vector** (a choice is possible only if the property can be a vector quantity)

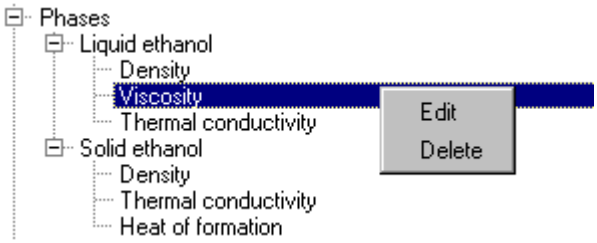
Notes:

[*)] If some property of the **Substance** is not defined in the **Substance Database**, then when this **Substance** is loaded into **Pre-Postprocessor** in will have type **Constant** and a default value.

[**)] If in the **Substance Database Editor** you set both enthalpy and specific heat, then the specific heat capacity, which was set in the **Substance Database**, will be ignored in **Pre-Postprocessor**. Specific heat of a substance in **Pre-Postprocessor** in this case is calculated using a predetermined enthalpy.

After the addition of properties can be set its numerical parameters using the **Edit** command from the context menu that appears when you right-click the corresponding node in the **Substance Database Editor** window.

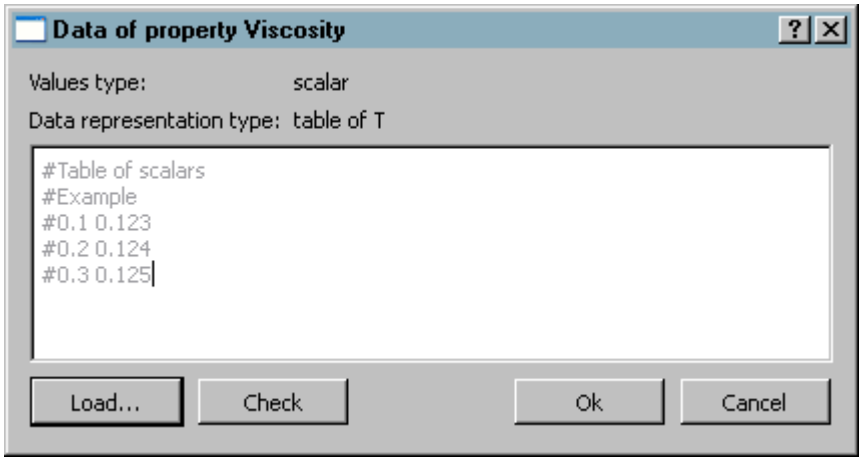
Context menu of a Substance's property



The context menu of a substance's property contains the following commands:

Command	Description
Edit	Changing the numerical parameters of the property using the **Data of property** dialog box (described in the next subsection).
Delete	Deleting the property. The program will ask you to confirm this action.

Dialog box "Data of property"



The **Data of property** dialog box opens with the command **Edit** from the context menu of the property, and double-clicking on a node of the property.

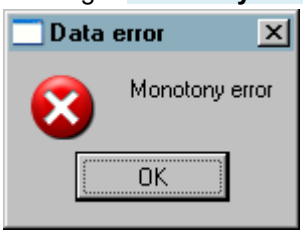
The **Data of property** dialog box contains the following options:

Values type	scalar or vector
Data representation type	corresponds to the data type specified in the **Add properties** by the addition of properties

In the main pane of the **Data of property** dialog box you enter the text that specifies the numerical parameters defining the properties in the format illustrated several lines of comments that are already in the field. Comments given by the **#** symbol at the beginning of the line.

The format is specified depending on the data type (data type representation).

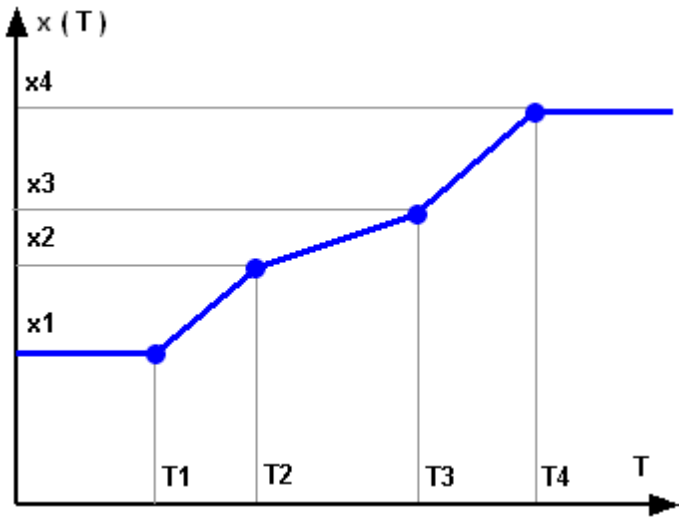
Type of data	Presentation of data	Example
Constant	A single value is specified: x	**0.123**
Table of T	Table of **Values** depending on the temperature is given by pairs of numbers: T_1 x_1 T_2 x_2 ...	**0.1 0.123** **0.2 0.124** **0.3 0.125**
Table of P	Table of **Values** depending on the pressure given by the pairs of numbers: P_1 x_1 P_2 x_2 ...	
Table of P and T	Table of **Values** depending on the pressure and temperature of groups of rows. Each group corresponds to a pressure and comprises a pair of numbers, each of which corresponds to a temperature of: $P = P_1$ T_{11} x_{11} T_{12} x_{12} ... $P = P_2$ T_{21} x_{21} T_{22} x_{22} ...	**P = 1** **0.1 0.123** **0.2 0.124** **0.3 0.125** **P = 2** **0.1 0.223** **0.2 0.224** **0.3 0.225**

	Pressure values (indicated in the opening lines of the groups) and temperature values within groups should follow in ascending order, otherwise when you try to input the data, the message "**Monotony error**":  If you enter an incorrect value or use incorrect data format, an error message will be also displayed.	

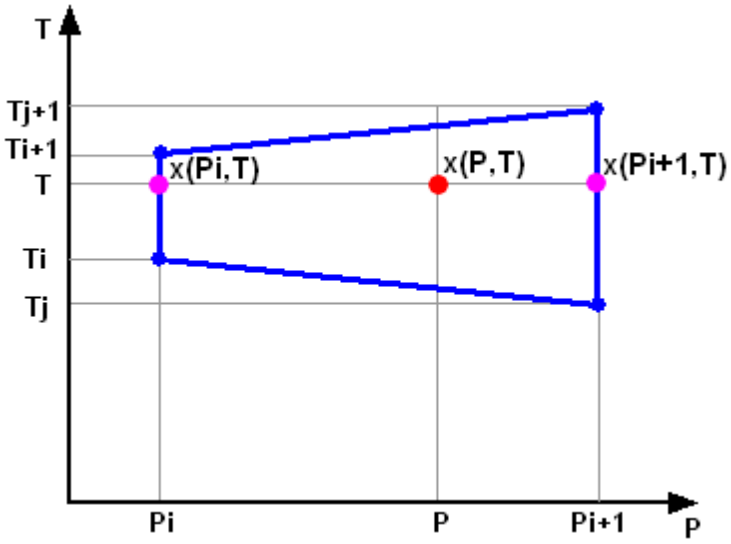
It is possible to download numerical properties from a `.txt` or `.dat` file using the **Upload** button. Contents from these files must match the described data format. Use the space symbol as a delimiter in these files.

When the property is set by the table, its values at intermediate points are calculated using linear or bilinear interpolation of table values.Beyond the table range, a value from the nearest table point is used.

When the variable depends on the pressure only or on the temperature only, linear interpolation is used. When the variable depends both on pressure and temperature, bilinear interpolation is applied (see https://en.wikipedia.org/wiki/Bilinear_interpolation).



Interpolation of a variable x, which depends on the temperature only



Bilinear interpolation of a variable x, which depends both on pressure and temperature

7.9 User modules

FlowVision's functionality can be expanded with *user modules* that are written by external developers.

The following types of user modules are supported:

- **API Evaluator** for calculating values
- **API Binder** for binding connected boundary conditions

User modules contain in files with extensions **`fvdll`**.

User modules, which have been loaded into the project, are presented in the project tree as child elements in the **User modules** folder in the **Preprocessor** tab. To load user modules into the project, use the **Load** command from the context menu of the **User modules** folder.

7.9.1 API Evaluator

User modules of the **Evaluator** type are intended to connect user modules that implement calculation of values. API **Evaluator** allows you to use in *FlowVision* functions from libraries of external developers.

To obtain values that are calculated by an API **Evaluator**, which is loaded into the project, click the  button in the expanded data input field in the **Properties** window and enter required input data into the dialog box, which opens.

7.9.2 API Binder

User modules of the **Binder** type are applied to connect user modules that implement connection of boundary conditions. API **Binder** connects two surfaces, on which special boundary conditions are set that implement functionality of the user module.

After a user module of the **Binder** type is loaded into the project, it becomes possible to select it as **Connection type** for elements **Binder condition #N** (see section Folder «Boundary links»).

8 Work with Pre-Postprocessor

This section describes the work with **Pre-Postprocessor**.

First, the guide on the user interface is given, then descriptions of typical user's step-by-step procedures follow, which you can do in **Pre-Postprocessor**.

We recommend you to read materials from the section Quick Start before starting to use this section.

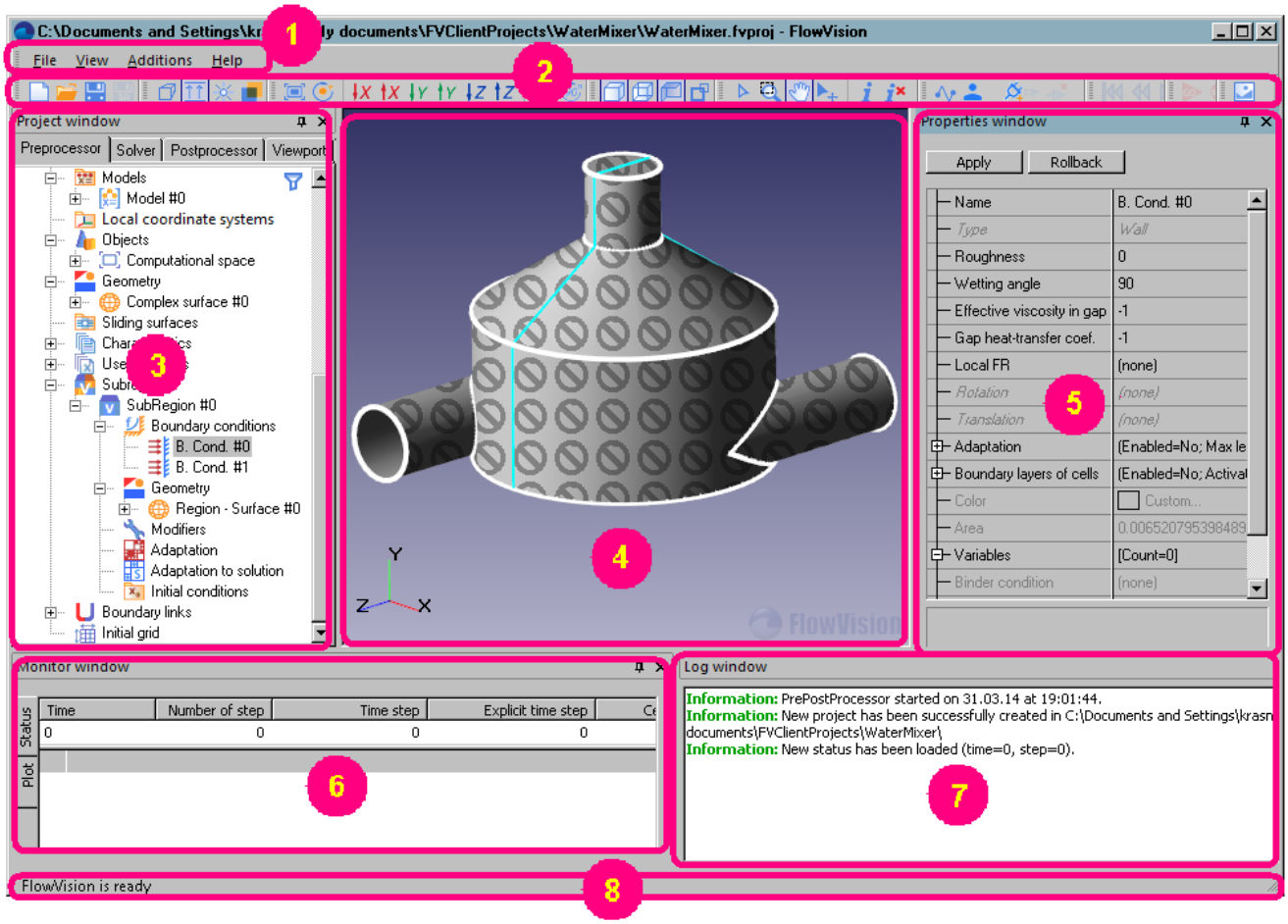
8.1 Interface guide on Pre-Postprocessor

The sections below describe the user interface of **Pre-Postprocessor**:

- Window of Pre-Postprocessor
- Main menu
- Context menu
- Toolbars
- The status bar
- Element selection window
- Window **View**
- Window **Project**
- Window **Properties**
- Window **Monitor**
- Window **Log**
- **Exposed parameters window**
- Window **Info**
- Formula editor
- Table editor

8.1.1 Window of Pre-Postprocessor

Pre-Postprocessor is used to create a project, the project's launch on the calculation and display the results.



Interface elements in **Pre-Postprocessor**

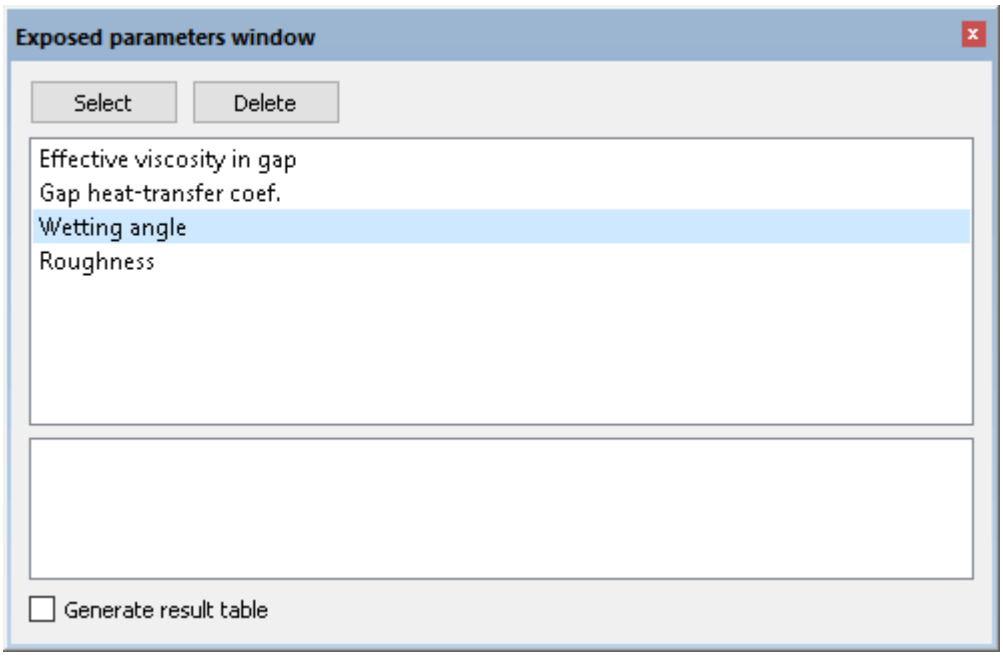
The **Pre-Postprocessor**'s window contains the following elements (see the illustration):

- Main menu (1)
- Toolbars (2)
- window **Project** (3), which displays the project tree
- window **View** (4) (also it is called as "graphics window")
- window **Properties** (5)

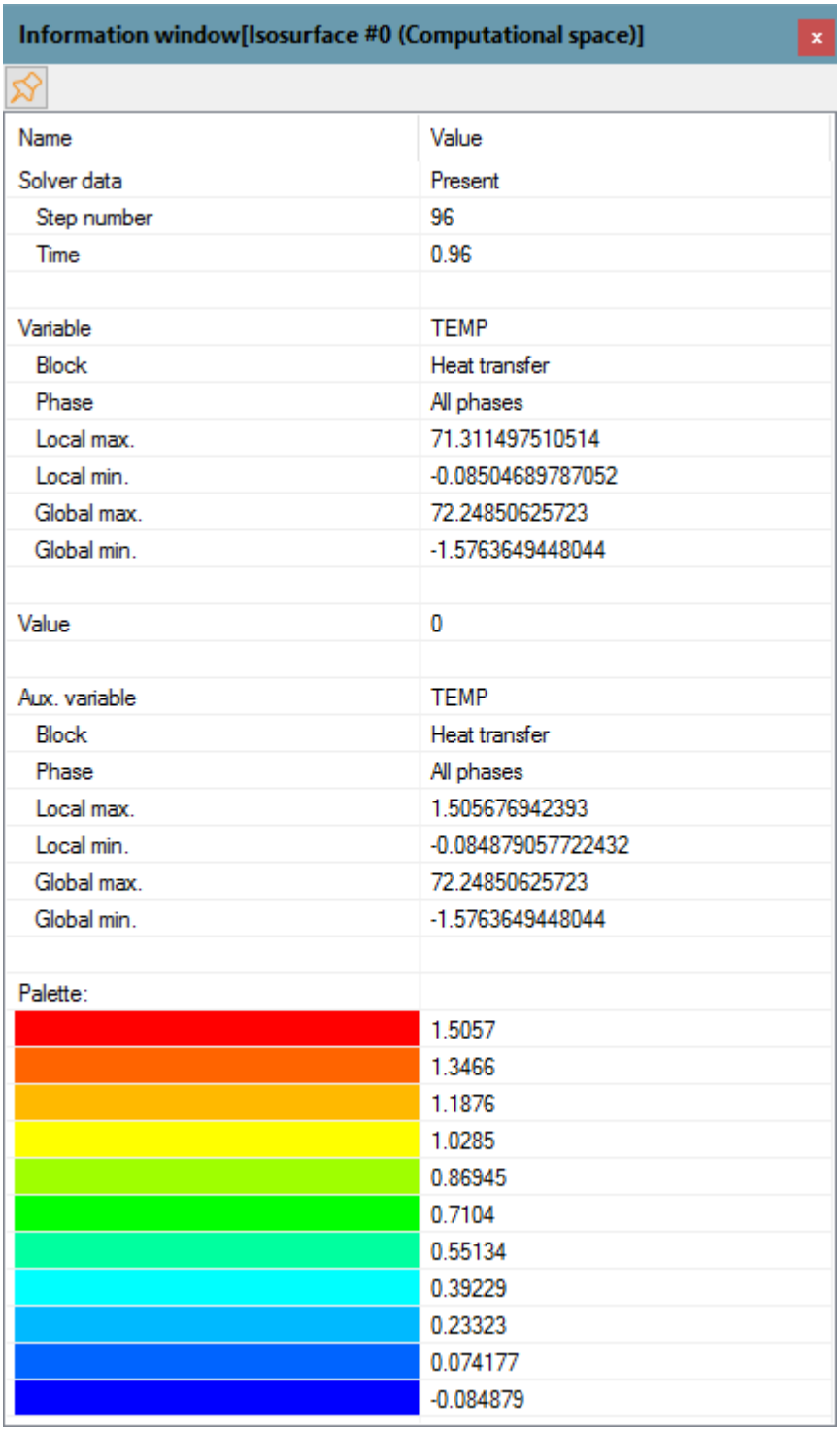
- window **Monitor** (6)
- window **Log** (7)
- status bar (8), which indicates the current status of the program.

Layout of the windows can be changed by the user.

Except the windows that are shown on the illustration, there are also the **Exposed parameters window** and the **Info** window (see illustrations below).



The **Exposed parameters window** (example)

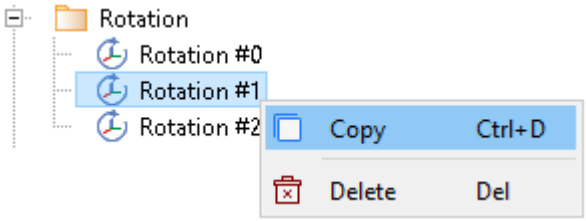


Information window[Isosurface #0 (Computational space)]

Name	Value
Solver data	Present
Step number	96
Time	0.96
Variable	TEMP
Block	Heat transfer
Phase	All phases
Local max.	71.311497510514
Local min.	-0.08504689787052
Global max.	72.24850625723
Global min.	-1.5763649448044
Value	0
Aux. variable	TEMP
Block	Heat transfer
Phase	All phases
Local max.	1.505676942393
Local min.	-0.084879057722432
Global max.	72.24850625723
Global min.	-1.5763649448044
Palette:	
	1.5057
	1.3466
	1.1876
	1.0285
	0.86945
	0.7104
	0.55134
	0.39229
	0.23323
	0.074177
	-0.084879

Window **Info** (example)

Right-clicking opens the context menu that contains commands, which is specific to the selected element from the project tree.



Example of a context menu

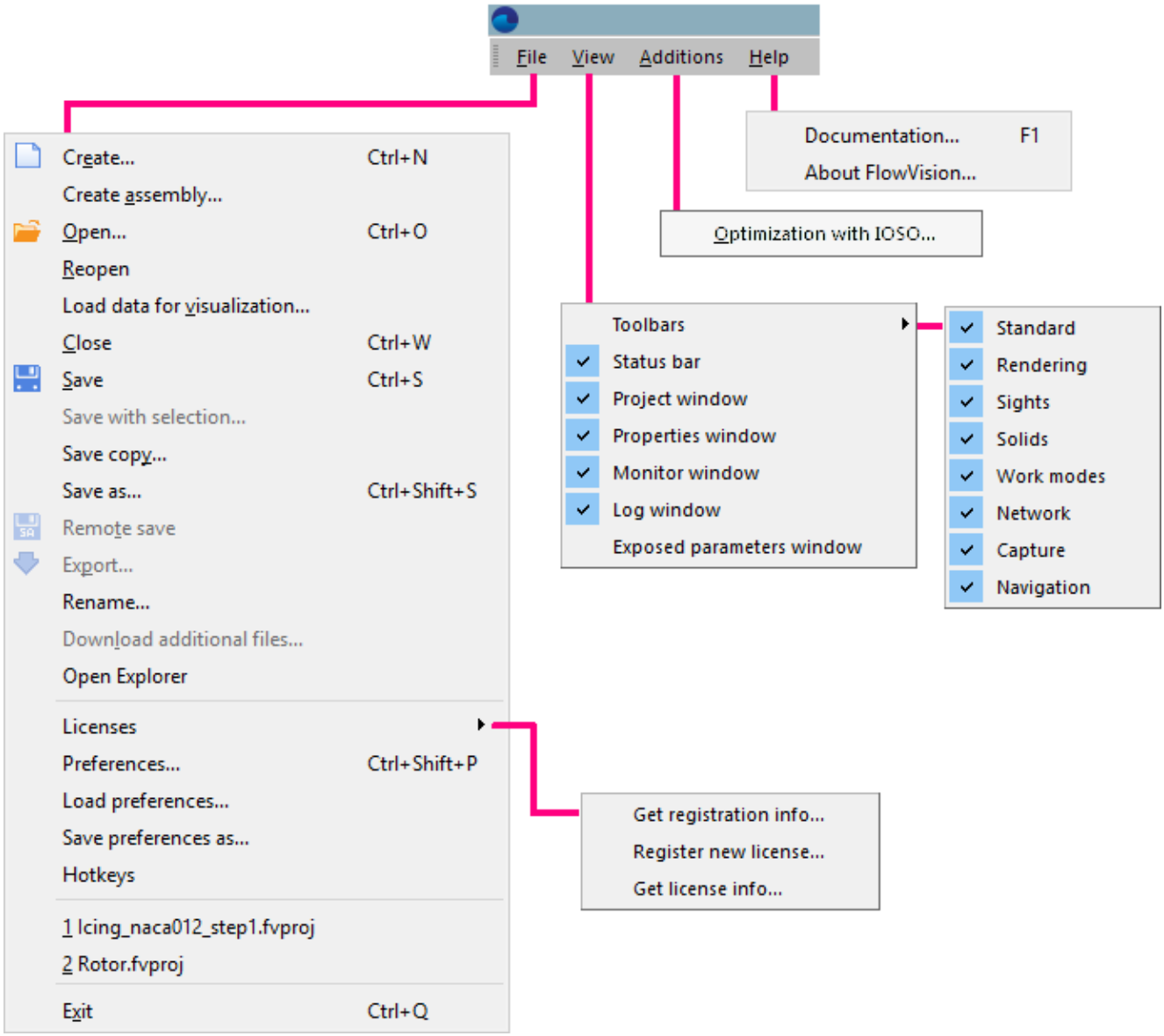
Note: displaying of operation windows **Pre-Postprocessor** can be switched on or off; you can also place them in different locations **Pre-Postprocessor** window and display in the "floating window".

Task information blocks consists of two steps:

- filling folders tree project required elements; add a new item to a folder follows the same - on the command **Create** context menu on line folder. In some cases, in the dialog box that opens by this command, you must select the desired type of item to be added.
- set and change the properties of the added elements (in the **Properties** window)

When you first start **Pre-Postprocessor** a window for entering license name. If the license name is defined, you must type it in this box. After that, the license name will be saved in the **Preferences Pre-Postprocessor** and drive it is no longer required. If the license name is not specified, then the box should be left blank.

8.1.2 Main menu



Scheme of the main menu

The main menu consists of the following submenus:

- **File** contains commands to load/save the project, settings, and load/save the configuration
- **View** contains commands to configure layout of Toolbars and the program's windows
- **Additions**: contains the command to run *FlowVision* with *IOSO*
- **Help**: contains commands for access to reference information

See descriptions below.

The "File" menu

Menu item	Hot keys (by default, you can chenge them))	Description
Create	**Ctrl+N**	Creating a new project, which contains either no geometry model of the computational domain or a geometry model based on a single file depending your selection in the **New project** dialog box, which opens.

Menu item	Hot keys (by default, you can chenge them))	Description
Create assembly		Creating a new project *based on several files* that contain a geometry model of the computational domain.  You *can not* create a geometry with multiconnection using an *assembly*. If you attempt to do this, the program will make an error message indicating on the first outer geometry.
Open[1)]	**Ctrl+O**	Closes the current project and the discovery of a previously created project.
Reopen[1)]		Opening the last saved version of the project with the loss of the last unsaved changes, i.e. closing without saving and opening the disc current proposal.
Load data for visualization		Loading in **Pre-Postprocessor** data layers stored in the calculation of the back end of the project
Close	**Ctrl+W**	Project Closure
Save[4)]	**Ctrl+S**	Saving the client part of the project in the client directory. Connected with **SOLVER** this action also serves to send changes **Solver** and use them there.
Save with selection[2) 4)]		Saving the client and server parts of the current project with ability to decimate records of the computation's history.
Save copy[2) 4)]		Copying the client and the server parts of the loaded project as a new project with another name. The current (original) project remains loaded on **Pre-Postprocessor** and **Solver**.
Save as[2) 4)]	**Ctrl+Shift+S**	This is also copying the client and the server parts of the loaded project as a new project with another name. But the new (the just created) project's copy becomes loaded on **Pre-Postprocessor** and **Solver**.
Remote save		Saving the results of the calculation (the server part of the project) in the server directory. This operation is available only in the calculation of the project on the **Solver**.
Export		This command allows you to export the results of calculation, not only during the computation, but also after the calculation is finished using the saved data from the grid. See section Data export after computation.
Rename		Renaming a client and server parts of the loaded project
Download additional files[3)]		Copying to the client part of the project files from the server-side project. Can be copied files visualization (`*.glo`, `*.fvvis`) or log files.
Open Explorer		Start *Windows Explorer* and open it in the directory of the client part of the project.
Licenses		Menu for licensing. There are commands: • **Get registration info** • **Register new license** • **Get license info** *(See descriptions in subsequent rows of the table)*

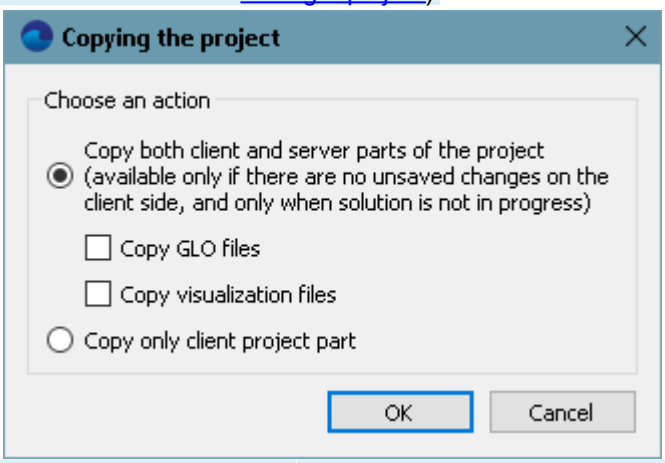
Menu item	Hot keys (by default, you can chenge them))	Description
Licenses > Get registration info		See section "Getting license information from Pre-Postprocessor".
Licenses > Register new license		See section "Registering a license in Pre-Postprocessor".
Licenses > Get license info		See section "Receiving license information from Pre-Postprocessor".
Preferences	**Ctrl+Shift+P**	Opening the **Preferences** dialog box, where you can configure the basic settings of Pre-Postprocessor.
Load preferences		Loading configuration settings from a file format `*.cfg`.
Save preferences as		Saving the current configuration settings in a file format `*.cfg`.
1 **2** **3** **...**		List of last opened project files
Exit	**Ctrl+Q**	Closing the **Pre-Postprocessor**'s window

Notes:

[1] The current project can also be accessed from *Windows Explorer*, when you drag the project's folder or the `fvproj` file into the graphics window.

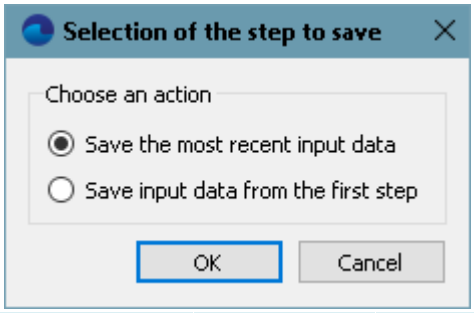
[2] If the project contains more than one record of the history of calculation:

- if the project is connected to **Solver**, the **Copying the project** dialog box opens with options of saving the project's copy (see details in the section Saving a project):

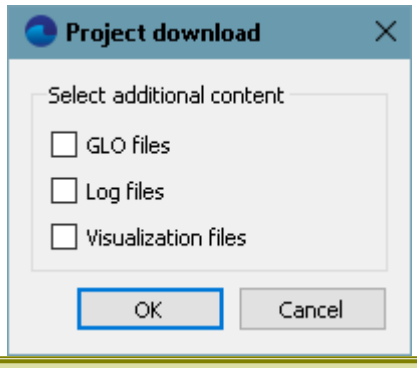


Then the **Non-steady-state steps decimation** dialog box opens. If after the last transfer of changes to **Solver** the project's input data change, only the client part of the project can be saved.

- if the project is *not connected* to **Solver**, the **Selection of the step to save** dialog box opens with options:



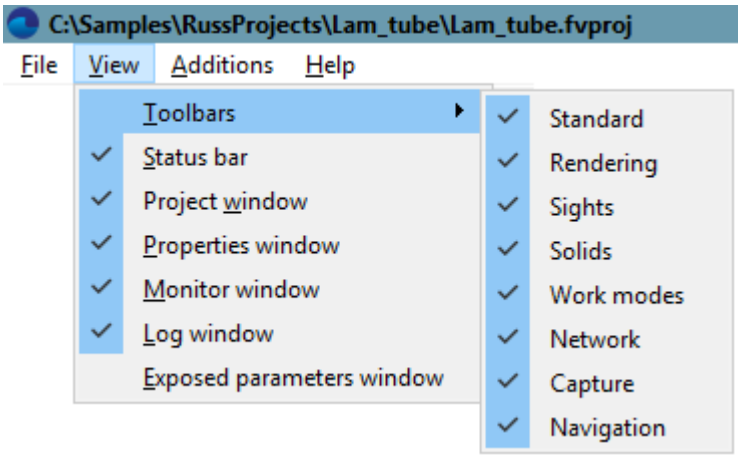
[3] When you apply the **Download additional files** command, the **Project download** dialog box appears where you select the desired files that will be downloaded:



Checkbox	Actions
GLO files	Download from the server part of the project in client part all **glo**-files
Log files	Download from the server part of the project to the client part all the log files
Visualization files	Download from the server part of the project to the client part all files with information for visualization (**fvvis**)

[4] When **Pre-Postprocessor** is in the read-only mode, these commands are not available.

The "View" menu



Menu item	Description
Toolbars > Standard	Displaying the appropriate toolbar: ☐ – the toolbar is not displayed ☑ – the toolbar is displayed
Toolbars > Rendering	
Toolbars > Sights	
Toolbars > Solids	

Menu item	Description
Toolbars> Work modes	
Toolbars > Network	
Toolbars > Capture	
Toolbars > Navigation	
Status bar	Determines whether or not the status bar is displayed ☐ – the status bar is not displayed ☑ – the status bar is displayed
Project window	Determines whether or not a given window ☐ – is not displayed ☑ – a window is displayed
Properties window	
Monitor window	
Log window	
Exposed parameters window	

The "Additions" menu

Menu item	Description
Optimization with IOSO	This command opens the dialog box for starting *IOSO*. This command is only available when: • *IOSO* version *2.0* or higher is installed on the computer • a project is opened in **Pre-Postprocessor** • **Pre-Postprocessor** is not connected to **Solver**

The dialog box for starting *IOSO* contains:

Element	Description
Optimization	Select a version of *IOSO* for the joint project
IOSO NM - non-parallel	Start *IOSO NM*
IOSO PM - parallel	Start *IOSO PM*
Timeout	Limit for duration of the computation
Close PrePostProcessor	Closes **Pre-Postprocessor** before starting *IOSO*
Solver	Startup parameters of **Solver** under the joint computation based on *IOSO-FlowVision*
Procs	See section Dialog box "Select solver".
Cores	
All	
Solver type	

Element	Description
OK	Start *IOSO* and create the project
Cancel	Back to the task of the project *FlowVision*

The "Help" menu

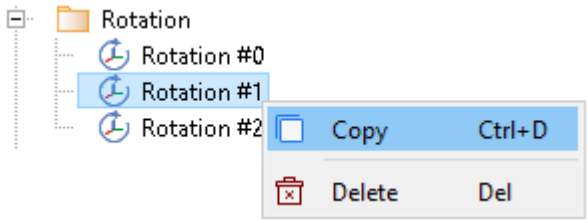
The **Help** menu contains commands for reference.

Menu item	Hot keys	Description
Documentation	**F1**	Opening the Help system.
About FlowVision		Displays information with information about the current version (version number, build date of release)

8.1.3 Context menu

The *context menu* is opened by right-clicking (when mouse settings are standard right-handed).

The context menu contains standard commands and commands that depend on the element on which it was opened.



Example of a context menu

Standard context menu commands

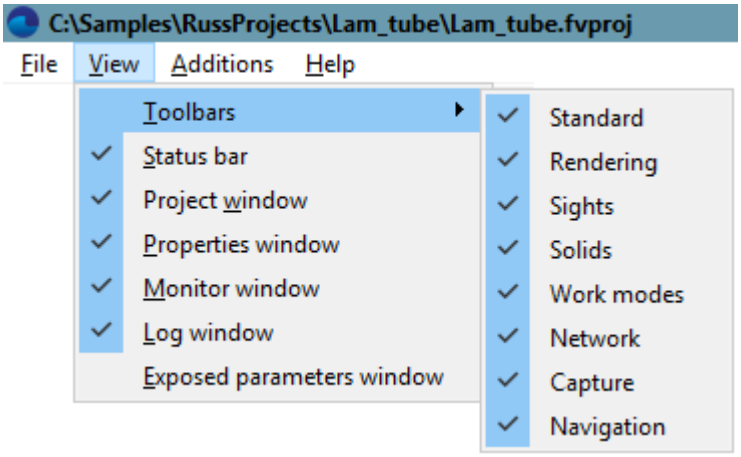
Command	Description
Create	Creating a new element
Copy	Copying the current element
Delete	Deleting the current element
Add/Remove	Adding a reference to an element that is already created before (a selection dialog box will open)

8.1.4 Toolbars

Toolbars locate at the upper part of window of **Pre-Postprocessor**. Some toolbar buttons duplicate corresponding commands of the Main Menu or items of context menus of the project tree elements.

If you drag the toolbar to the edge of the *FlowVision*'s window, it will stick to the edge automatically. If you drag the toolbar away from the edge of the window, the toolbar becomes floating, i.e. it is displayed in a separate window that can be moved freely within the screen.

Displaying or hiding **Pre-Postprocessor**'s toolbars is set in the main menu by the sub-menu **View > Toolbars**:

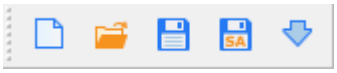


Settings for displaying the toolbars

There are the following toolbars:

- **Standard**
- **Rendering**
- **Sights**
- **Solids**
- **Work modes**
- **Network**
- **Capture**
- **Navigation**

Toolbar «Standard»



The **Standard** toolbar contains the following buttons:

Button	Name	Description
	**Create a project**	Creating a new project
	**Open a project**	Opening an existing project that was saved before
	**Save changes to the client side of the project**	Saving the current project
	**Save solution on the solver**	Saving the current project on **Solver** (saving results of the computation). This causes saving the results of the computation in the directory of the server part of the project.
	**Export data on solver**	Export the results of calculation, not only during the computation, but also after the calculation is finished using the saved data from the grid. See section Data export after computation. This button is equivalent to the command **File > Export** of the Main menu.

Toolbar "Rendering"

The **Rendering** toolbar contains the following buttons:

Button	Name	Description
	Enable/disable the use of perspective	When the button is released, the image is created with no perspective projection. When the button is pressed, the image is created with perspective projection.
	Enable/disable offset displacement of facet group boundaries	When the button is released, the image is created without intentional offset of boundaries of groups of facets. When the button is pressed, the image is created with intentional offset of borders groups facets.
	Turn on/off glare on the surface	When the button is released, the image is created with no glares on surfaces. When the button is pressed, the image is created with glares on surfaces.
	Enable/disable transparency support	When the button is released, the image is created with no taking into account the transparency. When the button is pressed, the image is created with taking into account the transparency.
	Enable/disable duplication and overlapping of subdomains up to the complete model supplied with sector-sliding	Enable/disable displaying of the full image of simulation defined in a sector statement with periodic and sliding surfaces. See illustrations in the section Sector-sliding setting.

Toolbar «Sights»

The **Sights** toolbar contains the following buttons:

Button	Name	Description
	Fit calculation region to window	Moving and zooming to the full display of the computational domain in the **View** window
	Set a new rotation center for the scene	Moving the center of rotation of the image (pivot) to a point of an applicable object selected in the project tree. As an object for changing the pivot you can specify: • A geometric object, which has a **Reference point**. The pivot is moved to the **Reference point** or (for a **Plane**) to the point P1 of the **Plane**. • A layer **Cell debug**. The pivot is moved to the center of the cell. • A local coordinate system for movements (Movement LCS (LCS-M)) or specified in it some **Rotation** or **Translation**. The pivot is moved to the origin of the local coordinate system. If no applicable object is selected, the pivot is moved to the center of the computational domain.
	View against axis X	Positioning the line of sight of the observer of the scene in the **View** window along or against

Button	Name	Description
	View along axis X	one of the coordinate axes
	View against axis Y	
	View along axis Y	
	View against axis Z	
	View along axis Z	
	Rotate the graphics window view 90 degrees counterclockwise	Rotating the image in the scene by 90 degrees either clockwise or counter-clockwise
	Rotate the graphics window view 90 degrees clockwise	

Toolbar «Solids»

The **Solids** toolbar contains the following buttons:

Button	Name	Description
	Enable/disable surface fill	When the button is pressed, the facets are filled. When the button is released, the facets are not filled.
	Enable/disable the display of facet edges	When the button is pressed, edges of the facets are displayed. When the button is released, edges of the facets are not displayed.
	Show/hide surface facing viewer	Hide surfaces of the geometry, which are oriented by their non-computational sides[1] towards to the observer. When the button is pressed, *only* internal surfaces of subregions, in which models are specified, are visible (external surfaces that are oriented to the observer, become transparent). When the button is released, *all* internal and external surfaces of subregions, in which models are specified, are visible. So, from the viewpoint of an outer observer, the internal surfaces are obscured by the external surfaces.
	Enable/disable display of face group borders	When the button is pressed, borders of facet groups are displayed. When the button is released, borders of facet groups are not displayed. The color and thickness of the outlines are set by parameters **Outlines >...** in the properties of the folder **Solids**.

Note:

[1] A non-computational subregion is a **Subregion**, on which no **Model** is specified.

Toolbar "Work modes"

The **Work modes** toolbar contains the following buttons:

Button	Name	Description
	Enable facet group selection mode	Selection mode for groups of facets in the **View** window (see Selecting a geometry element).
	Enable window zoom mode	Zoom in or zoom out mode for displaying the **View** window (see Zooming and unzooming the visible area).
	View transformation mode	Setting mode the angle and location of the scene (see Changing sight angle and scene orientation)
	Enable edit mode for selected object	Edit mode settings for the selected object (see Changing position, orientation and scale of an Object by the mouse)
	Show info window for selected object	Opening the **Info** window for the element, which is selected in the project tree
	Close all information windows	Closing all **Info** windows

Toolbar "Network"

The **Network** toolbar contains buttons for working with **Solver** and **Solver-Agent**:

Button	Name	Description
	Solver agent log in	When you click this button the program will open dialog boxes for a user authorization on **Solver-Agent**. See details in the section Connection to Solver-Agent and user authentication. When **Pre-Postprocessor** is connected to a **Solver**, this button is not available.
	Edit solver agent user information	This button opens the **Solver-Agent** user registration information, which is displayed in the **User Registration** dialog box, see section Registration data (profile) of Solver-Agent's user and their change. (This button is only available when you are already authorized on **Solver-Agent**)
	Connect to solver	Creating and running a new solver with default parameters and connecting to it. After this, the button (**Connect to solver**) disappears and on its place the button (**Disconnect from the solver**) appears, see below.
	Disconnect from the solver	Disconnecting the project from the **Solver**, to which it is connected now. (This button is only available when **Pre-Postprocessor** is connected to a **Solver**)
	Close the solver and disconnect from it	Disconnecting the project from **Solver** and terminates operation of the **Solver**. (This button is only available when **Pre-Postprocessor** is connected to **Solver**)
	Open solver selection window	Opening the **Select solver** dialog box.
	Create solver and start computation	Creating a **Solver** with default parameters and starting computation on it
	Start computation	Starting computation on the **Solver**, to which the project is connected.

Button	Name	Description
		The button is available if the project is loaded into the server part and computation of the project is not running.
	Stop computation	Stopping computation on the **Solver** at the end of the current iteration. The button is available if computation of the project is running.

Depending on your current situation, some buttons can be invisible or unavailable.

See sections:

- Starting and stopping project's computation
- Dialog box «Starting solve»

Toolbar "Capture"

The **Capture** toolbar contains the following buttons:

Button	Name	Description
	Save the contents of the graphics window to an image file	Opening the **Image capture** dialog box[1.2)]. Saving to a file the current image from the the **View** window.
	Start capturing an image sequence	Opening the **Image capture** dialog box. Running sequentially saving the images from the **View** window to files.[1.2)] • at each time step in the calculation, • when loading a previously saved data (when switching from step to step).
	Pause capturing an image sequence	Suspension or resumption save an image file of the **View** window
	Finish capturing an image sequence	Stop storing an image file of the **View** window

Notes:

[1)] After the call to save the image to a file, the **Image capture** dialog box opens, see section Controlling visual capture.

[2)] Save a picture from the graphics window to a file made in one of the following formats: *Windows bitmap* (`*.bmp`), *Targa* (`*.tga`).

Toolbar "Navigation"

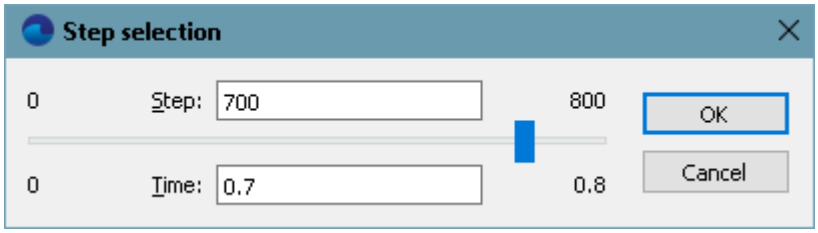
The **Navigation** toolbar is designed to handle the results of calculations carried out in the mode enabled recording of data autosave or layers autosave with **History** turned on.

Loading data is available only in connection with the **Solver**. In the transition from one to another saved result updated all the layers.

Loading layers is only available without connection to the **Solver** in the presence of the client part of the project the files to the information for visualization and enabled the automatic download of files with information for visualization. In the transition from one to another saved result updated only the layers that were created before the computation occurs.

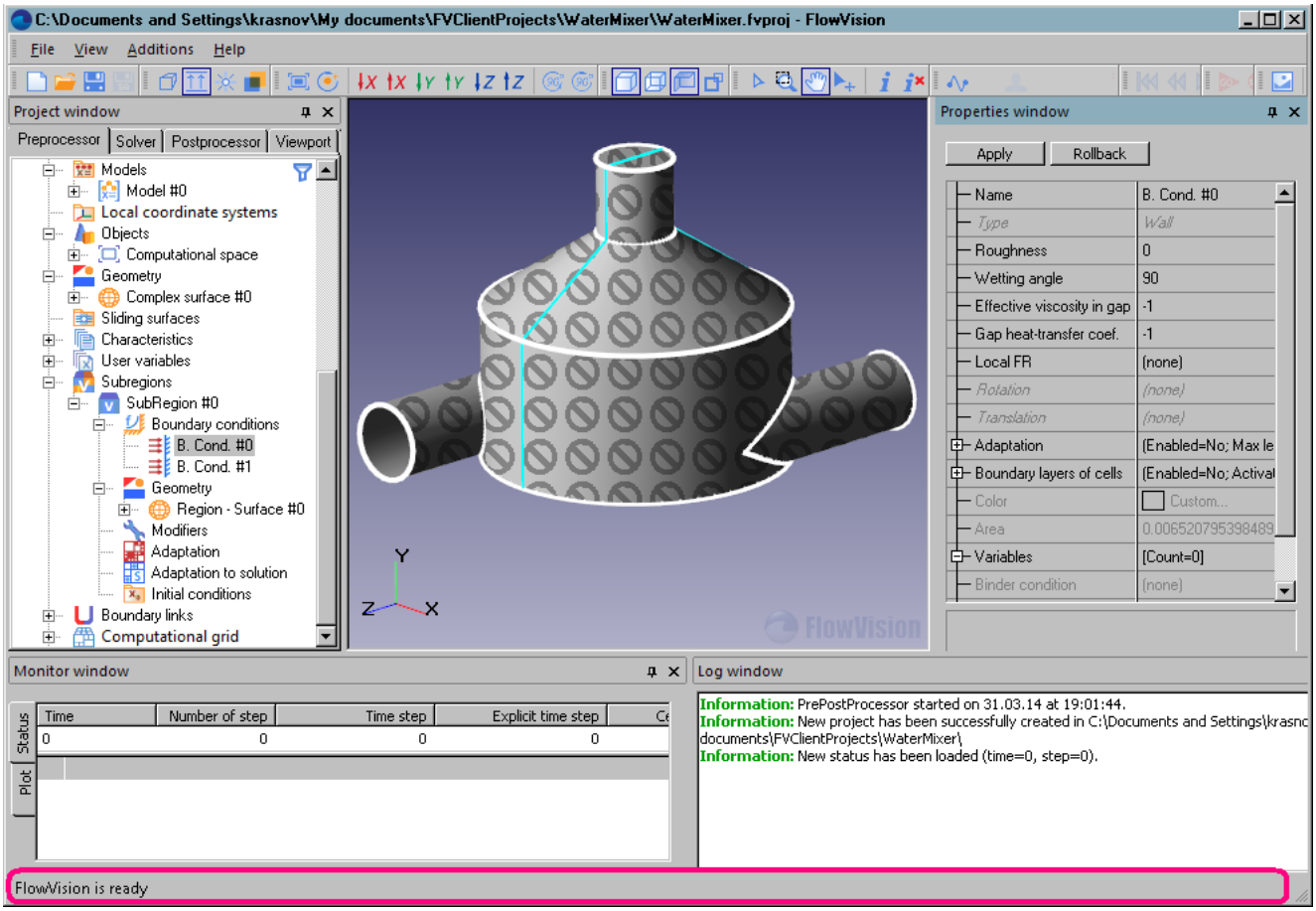
If the mode is **sequentially storing** images,it is also written a sequence of files with the contents of the graphics window.

Navigation toolbar contains the following buttons:

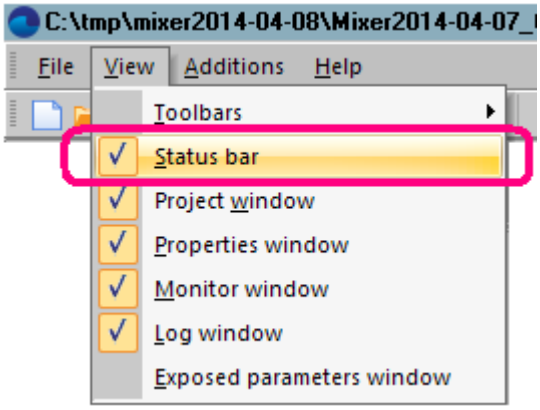
Button	Name	Description
	**Load first step**	Loading in **Pre-Postprocessor** the first saved result of the computation
	**Load previous step**	Loading in **Pre-Postprocessor** the previous saved result of the computation
	**Load next step**	Loading in **Pre-Postprocessor** the next saved result of the computation
	**Load last step**	Loading in **Pre-Postprocessor** the last saved result of the computation
	**Selecting the calculation history step to download**	Opens the **Step selection** dialog box where you can select a saved record, specifying its step or time:  Select the step or time by: • moving the slider • specifying the desired time step (field **Step** over the slider) • specifying the desired time (field **Time** under the slider)
	**Autoload visualization files**	Enable/disable automatic download of data files to render the scene
	**Start automatic sequential transition through the steps of the calculation history**	Starting automatic transitionfrom the current stored result to last recorded or to stop
	**Stop automatic sequential transition through the steps of the calculation history**	Stop automatic transition to the next saved result
	**Automatically export data during playback of saved steps as during calculation**	This button can be either pressed or released, this state is toggled when you click the button. When the button is in the pressed state, then the program will make output data into files for all objects, for which the data saving into files is specified. So you can record required data into the files during the playback and you do not have to carry out the computation again.

8.1.5 Status bar

The *status bar* locates at the bottom edge of the **Pre-Postprocessor**'s window and displays the program's status.



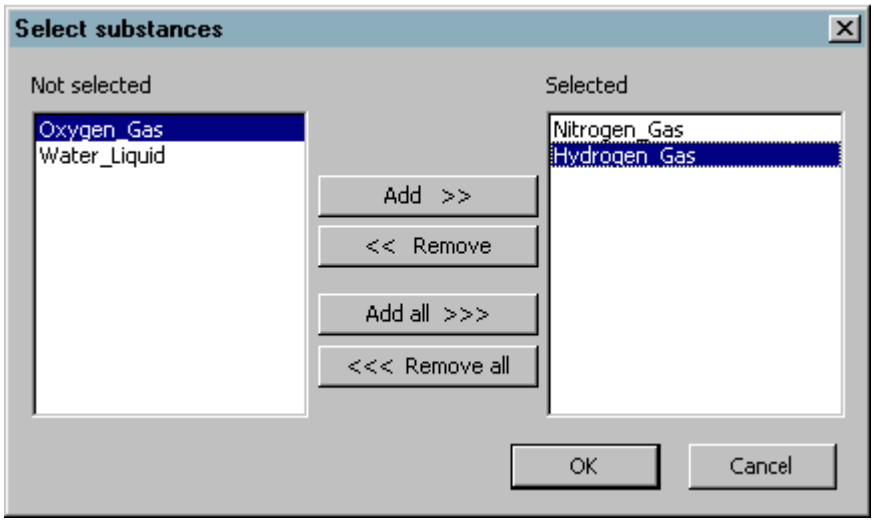
Availability of the **Status Bar** is defined by the **View > Status Bar** command from the **Main menu**:



8.1.6 Element selection window

Window select items to add a pre-existing elements in the selected folder. This dialog box is opened by the **Add/Remove** command from the context menu of the selected folder.

Such windows are used, for example, to specify, which **Substances** are present in a **Phase**:



Similar windows are applied to select **Phases** that are included into a **Model**.

Such dialog boxes contain elements:

Not selected	List of items available to be added to the specified folder
Selected	List of items that have been added to the specified folder
Add	Move the highlighted items from the pane **Not selected** to the pane **Selected**.
Remove	Move the highlighted items from the pane **Selected** to the pane **Not selected**.
Add all	Move all items from the pane **Not selected** to the pane **Selected**.
Remove all	Move all items from the pane **Selected** to the pane **Not selected**.
OK *(screen button)*	Close the dialog box and elements in the selected folder will correspond to items listed in the pane **Selected**.
Cancel *(screen button)*	The dialog box will be closed without changing the contents of the selected folder.

Selecting of items is done by the mouse. You can select multiple items (to do so, hold down the **Ctrl** key).

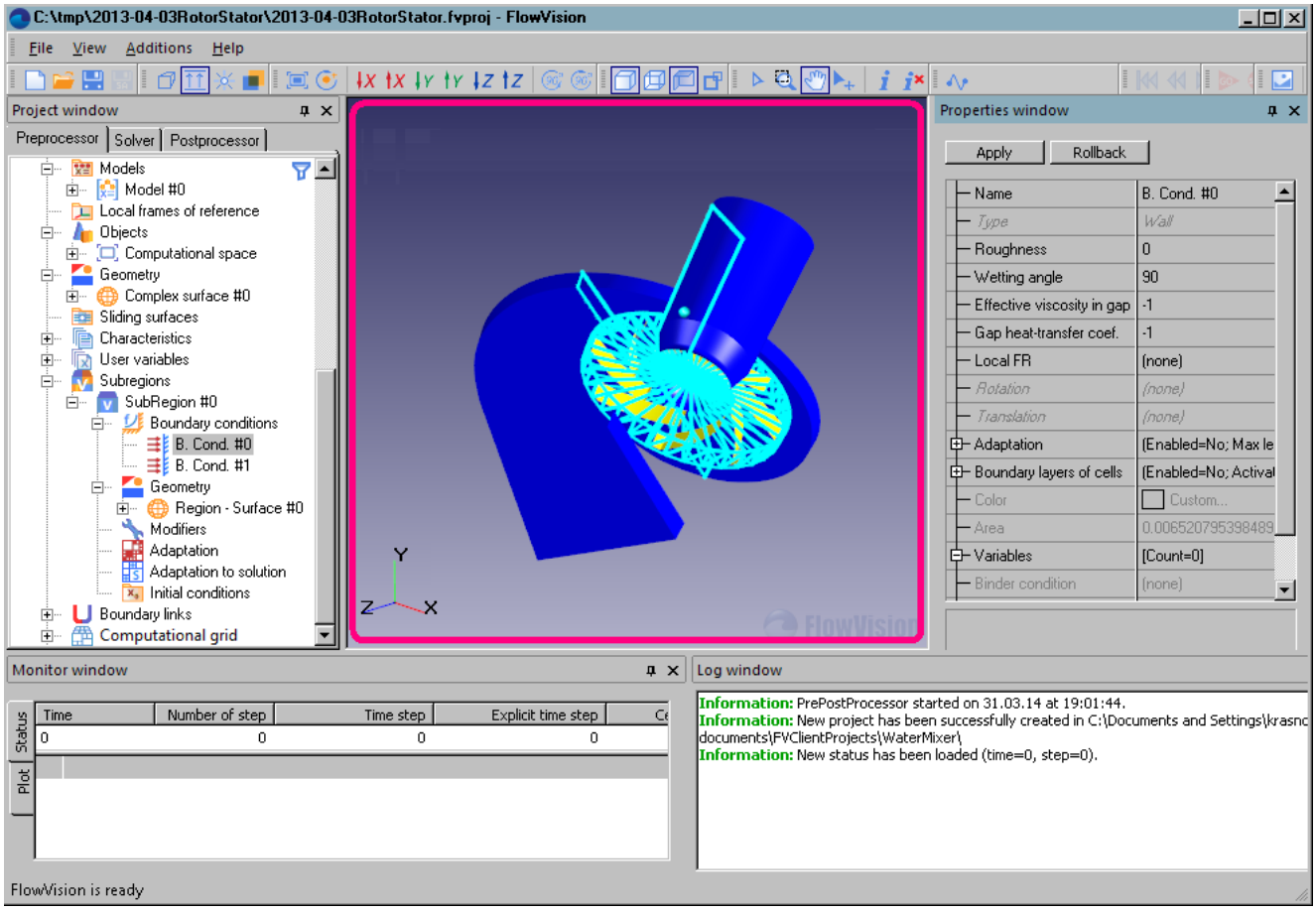
You can also move items to the other pane by dragging them by the mouse or double-clicking them.

8.1.7 Window «View»

The **View** window is part of the working window **Pre-Postprocessor**.

The **View** window displays the geometry model of the computational domain, objects and layers.

Actions on objects and layers in the **View** window are performed using mouse, toolbar buttons and keyboard keys.



The **View** window of **Pre-Postprocessor** (marked by a pink outline)

Graphics window is designed to display the statement of the problem and the results of calculation.

In the graphics window displays visible **Objects** and **Layers**.

Background and title for the **View** window is set in properties of the root folder **3D-Scene** in the **Postprocessor** tab of the **Project** window. Lighting for the **View** window is set in the **Lighting** folder in the **Postprocessor** tab of the **Project** window.

The colors match the colors of the surfaces of the boundary conditions.

FlowVision tools allow you to perform the following steps in the **View**:

1. select elements of geometry
2. increase the selected area
3. adjust the angle of view and the location of the scene
4. changing position, orientation and scale of an Object by the mouse

8.1.7.1 Selecting a geometry element

If there is a layer **Solids** in **Postprocessor**, then in the **View** window, you can select *the group* with the mouse. You can select only **Groups** belonging to **Subregions**, in which some **Model** is set. Selection of **Groups** from the screen is used for placement of the boundary conditions.

To select a **Group** in the **View**, you have to:

- select (**Enable facet group selection mode**) in the **Work modes** toolbar or press and hold down **Ctrl**,
- move the cursor to the **Group** in the **View** window and press the left mouse button.

When you select the **Group** in the **View** window, it becomes visible, and the remaining groups invisible. The selected group is colored in color as the corresponding boundary conditions are given on this group. The reverse side of the surface on which set the selected group are marked with .

After selecting a **Group** in the **View** window, you can call its context menu by clicking the right mouse button.

The context menu of a **Group** in the **View** window:

Command selected from the context menu	Description
Boundary condition > ***Name of a boundary condition***	Set a **Boundary condition** on the **Group**
Select next group	Select group located in this
Select other side	Select the other side of the surface
Regroup	Regroup the **Group**

In order to highlight the occlusion element geometry using the context menu:

- move the cursor to the position of the image, under which there is an item
- open the context menu
- using a serial command calls **Select the next group**, select the desired surface
- if the surface of the two sides, in order to choose the right direction, use the **Select the other side**.

In order to highlight the occlusion element geometry using the left mouse button:

- move the cursor to the location of the image, under which there is an item
- by sequentially pressing the left mouse button to move to the desired element geometry

8.1.7.2 Zooming and unzooming visible area

In order to switch to zoom into a specific part of the image, follow these steps:

Step	Actions
1	Turn on the instrument  in the toolbar **Work modes**.
2	Place the cursor at the place where you want one corner of the box, and click the left mouse button.
3	With the left mouse button, move the cursor diagonally to the opposite corner of the bounding box.
4	Release the left mouse button.
5	Region, captured frame will be stretched on all window **View**.

In order to switch to reduce the image visible in the graphics window,follow these steps:

Step	Actions
1	Turn on the instrument  in the toolbar **Work modes**.
2	Right-click the point of the graphics window, which becomes the center of large field of view.
3	As a result: • part of the scene, visible in the window, decrease and become visible parts of the scene, hidden behind the window frame, • click point will be the center of a new kind.

8.1.7.3 Changing sight angle and scene orientation

Setting the angle and location of the scene can be performed using standard tools in toolbar **Sights** or manually.

In order to enter the manual mode the angle and location of the scene, follow these steps:

Step	Actions
1	Turn on the tool  (**View transformation mode**) in the **Work modes** toolbar.
2	To rotate an image in space, move your mouse pointer around the screen by holding down the left mouse button. Also you can click by the mouse within the **View** window and then use the computer's keyboard: • to rotate around the horizontal axis, which lies in the image plane, press keys ↑ (up arrow) or ↓ (down arrow) • to rotate around the vertical axis, which lies in the image plane, press keys → (right arrow) or ← (left arrow)

Step	Actions
3	To rotate the image around the axis, which is perpendicular to the image plane: • move the mouse pointer within the **View** window to the left or to the right while holding down both mouse buttons • *or* click by the mouse within the **View** window and then use keystrokes **Ctrl**+→ or **Ctrl**+← (this means that you hold down **Ctrl** and press an arrow key → or ←).
4	To move the image, move the mouse pointer over the screen with the right mouse button pressed.
5	To zoom the image in/out: • Move the mouse pointer on the screen up/down, hold down both the buttons • Rotate the mouse scroll wheel (if your mouse has it)
6	To change the center of rotation of the image - in the tree **Pre-Postprocessor** select **Object**, the center of which will be used as the center of rotation, and click (**Set a new rotation center for the scene**) in the **Sights** toolbar.

8.1.7.4 Changing position, orientation and scale of an Object by the mouse

In interactive mode, using the mouse, you can dynamically change the following parameters of the elements folder **Objects** in **Preprocessor** and **Postprocessor**:

- position of the reference point (shown as a ball)
- direction of the vector

In order to switch to dynamic settings,performthefollowing steps:

Step	Actions
1	Turn on the (**Enable edit mode for selected object**) tool in the **Work modes** toolbar.
2	Select an editable object in the project tree.
3	Follow the mouse: **Action Mouse** / **Object** / **Result** move the mouse around the screen by holding down the right button — **Line**, **Plane**, **Box**, **Cone/cylinder**, **Ellipsoid/sphere**, **Imported object** — moving the reference point in a plane perpendicular to the direction of the vector move the mouse around the screen by holding down the left mouse button — **Line**, **Plane**, **Box**, **Cone/cylinder**, **Ellipsoid/sphere**, **Imported object** — vector rotation around the reference point move the mouse around the screen by holding down both the buttons — **Plane** — moving the reference point in the direction of move the mouse around the screen by holding down both the buttons — **Box**, **Cone/cylinder**, **Ellipsoid/sphere**, **Imported object** — scaling of the object relative to the origin
4	The relevant parameters of the selected object will change (as can be seen in the **Properties** window of the **Object**).

Note:

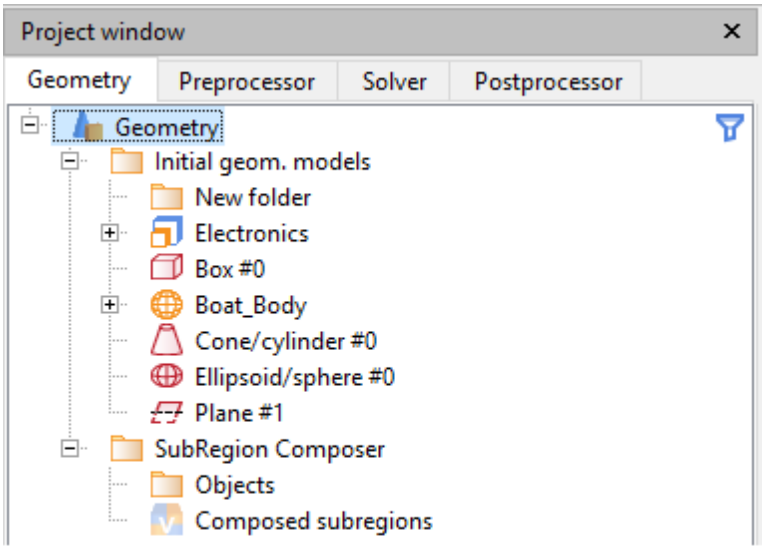
If you select another item in the project tree, or switch to online mode,edit the parameters of the object (includes buttons) automatically change to setting mode angle (includes buttons).

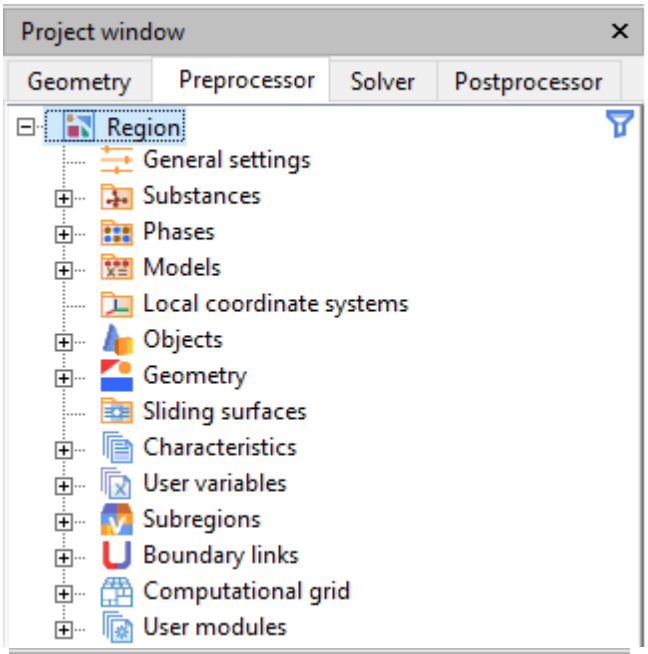
8.1.8 Window «Project»

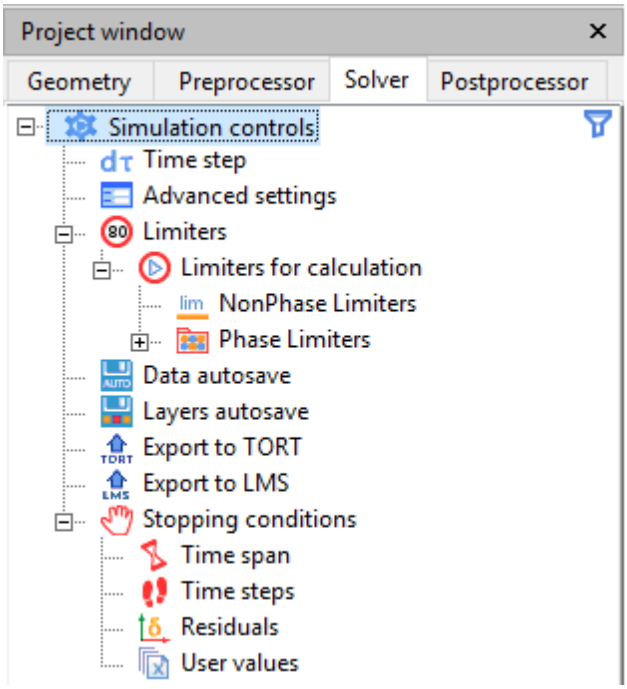
The **Project** window contains the following tabs, which collectively present the *project tree*:

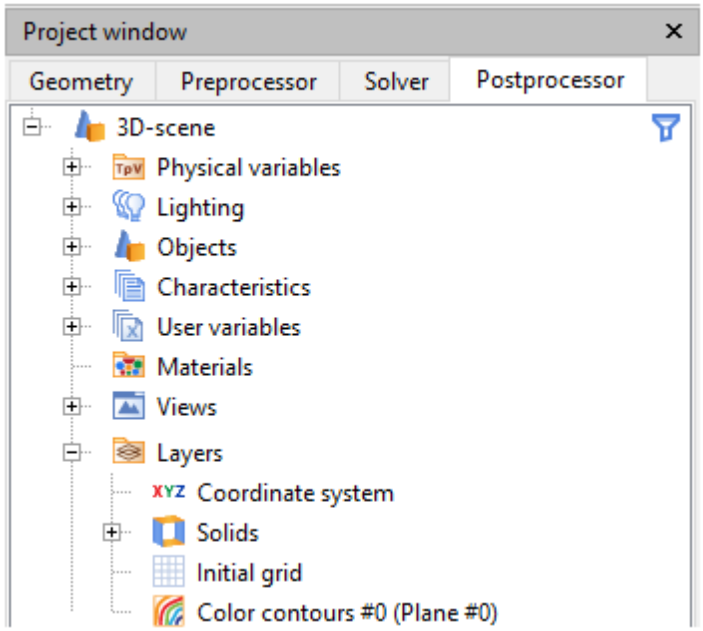
- **Geometry**
- **Preprocessor**
- **Solver**
- **Postprocessor**

(See illustrations below)


Project window
Geometry
Preprocessor
Solver
Postprocessor
Geometry
Initial geom. models
New folder
Electronics
Box #0
Boat_Body
Cone/cylinder #0
Ellipsoid/sphere #0
Plane #1
SubRegion Composer
Objects
Composed subregions


Project window
Geometry
Preprocessor
Solver
Postprocessor
Region
General settings
Substances
Phases
Models
Local coordinate systems
Objects
Geometry
Sliding surfaces
Characteristics
User variables
Subregions
Boundary links
Computational grid
User modules


Project window
Geometry
Preprocessor
Solver
Postprocessor
Simulation controls
Time step
Advanced settings
Limiters
Limiters for calculation
NonPhase Limiters
Phase Limiters
Data autosave
Layers autosave
Export to TORT
Export to LMS
Stopping conditions
Time span
Time steps
Residuals
User values



The elements of the project tree include:

- objects:
  - standard objects
  - imported objects
  - special objects
- custom variables
- characteristics and internal characteristics
- other elements

The **Project** window is visible when the **View > Project window** item is selected in the main menu.

To close the **Project** window, either:

- unselect the **View > Project window** item is selected in the main menu
- *or* click the **"x"** in the upper right corner of the window

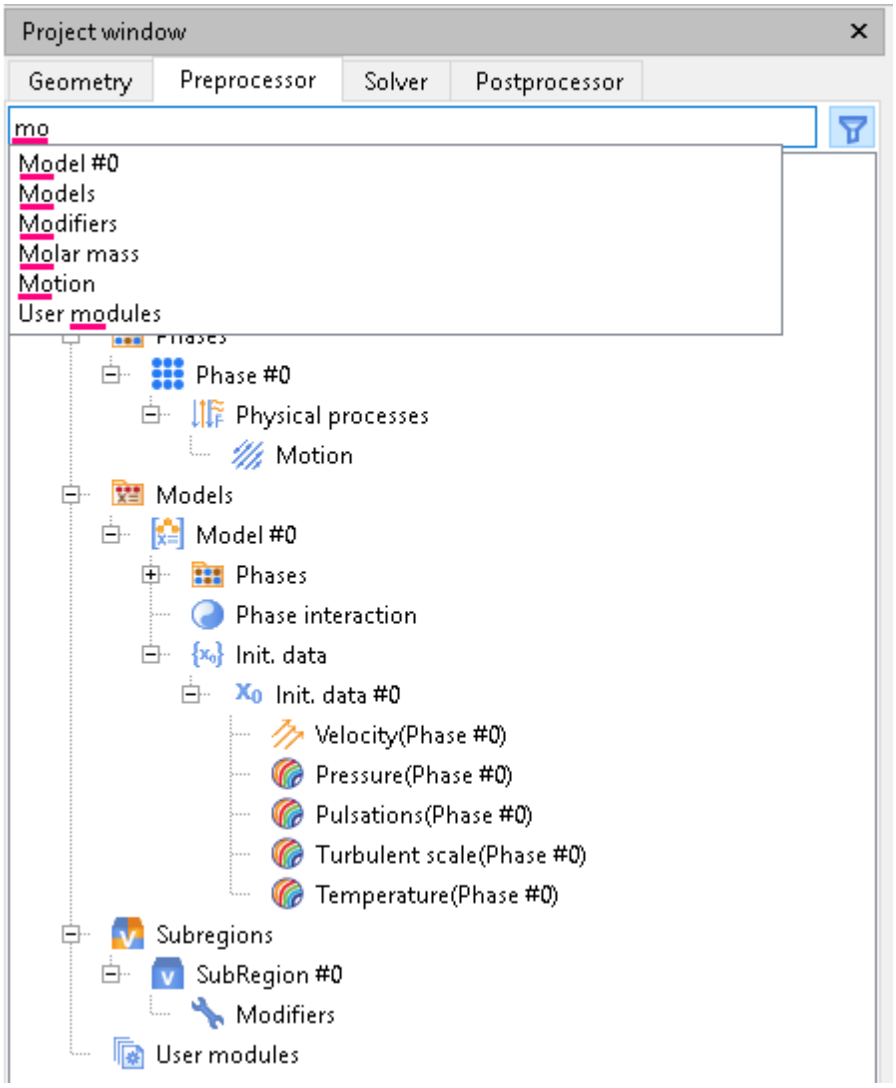
At the beginning of work with a project, after the project's creation and loading the geometry model of the computational domain, all folders of the project tree are empty, except folders that contains information that relates to the geometry model of the computational domain and settings of its displaying in the **View** window:

- In the **Preprocessor** tab the geometry model of the computational domain is presented in folders **Region > Subregions > SubRegion #0 > Geometry**; the automatically generated (at loading) boundary conditions are placed into the folder **Region > Subregions > SubRegion #0 > Boundary conditions**
- **Postprocessor** tab display options boundary conditions are placed in the folder:
  - **3D-scene > Objects > Computational space > Solids > Subregions > SubRegion #0 > Boundary conditions**
  - **3D-scene > Layers > Computational space > Solids >Subregions > SubRegion #0 > Boundary conditions**

Looking elements in the project tree by name

For quick navigation to an element in the project tree, you can use filtering by name. To enable filtering, click the (**Filter by name**) screen button, which is presented in each tab of the **Project** tree.

When this button is pressed, the window contains a text field, in which you can specify a fragment of an element's name that you are looking for:



This filtering by name is being done together with entering the text, case-insensitively. To finalize the filtering, press the **Enter** key on the keyboard or click by your mouse somewhere within free space near the project tree.

Under the text field, list of names, which contain the search string, will be displayed. If you see the required line in this list, you can navigate to it by mouse clicking or by mouse or by pressing keys ↑ (up arrow) or ↓ (down arrow).

The project tree will be changing as you enter the search text – the tree will contain only the elements, which contain the search text, and their parent and child elements.

The pressed (active) **Filter by name** button is displayed as . To turn the filtering off, click again.

The filtering by name is set individually for each tab of the **Project** window.

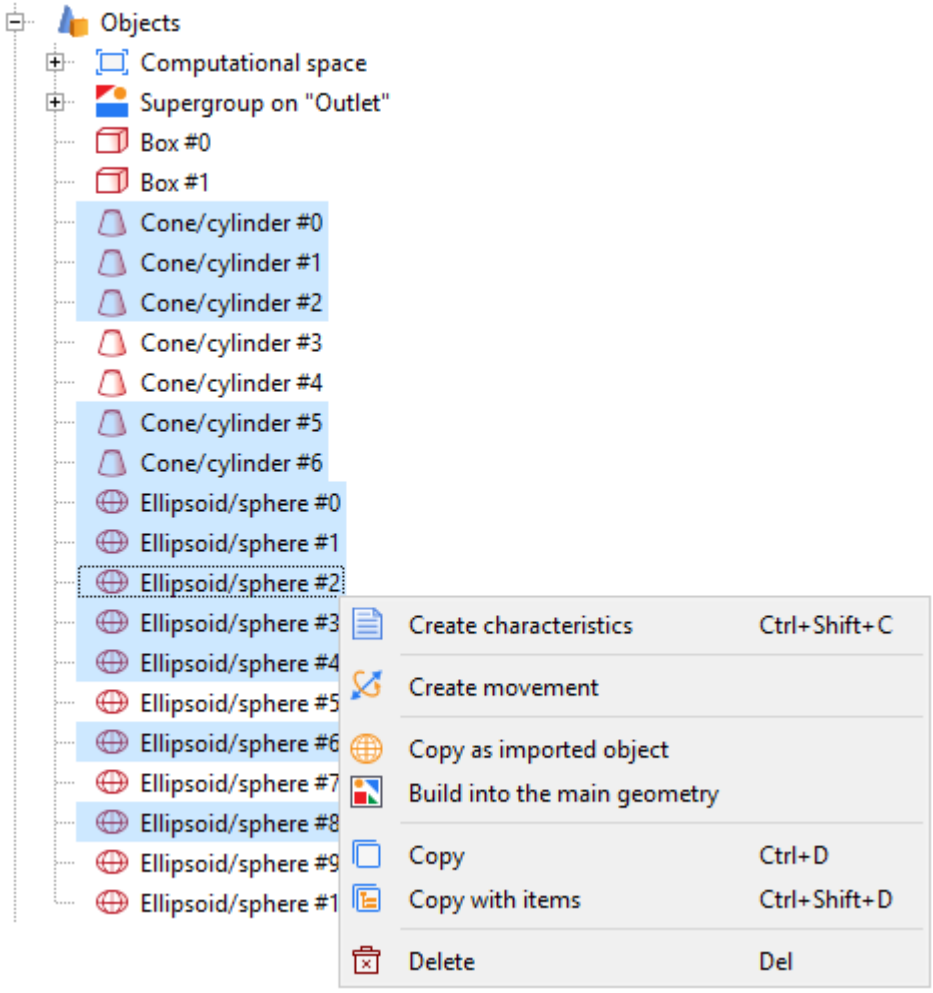
Clicking the released (not active) (**Filter by name**) button is duplicated by hot keys **Ctrl+F**.

Group operations with elements of the project tree (multiselect)

You can select multiple elements in the project tree and then apply to all the selected elements together:

- a command from the context menu
- changing properties that are common for all the selected elements (for example, specifying variables, changing names of the elements themselves, changing names of files, etc.)

To select multiple elements hold pressed the key **Ctrl** (to select or unselect elements individually) or **Shift** (to select/unselect all elements from the current cursor position to the element, on which you click):



8.1.8.1 Objects in the project tree

The project includes a number of elements that are displayed and can be edited in the project tree in both **Preprocessor** and **Postprocessor** tabs, so their descriptions are given in separate sections.

Objects are such elements.

Classes and types of Objects

Objects in *FlowVision* can be classified in the following classes:

- standard geometry objects
- **Computational space**
- **Imported objects**
- special objects

Standard objects may be of the following types:

- **Line**
- **Plane**
- **Box**
- **Cone/cylinder**
- **Ellipsoid/sphere**

Standard objects **Box**, **Cone/cylinder** and **Ellipsoid/sphere** are *objects of finite volume*.

Imported objects are used:

- to define the boundaries of the computational domain and subregions in it
- to set the boundary conditions

In *FlowVision* can be made a copy of the standard object format **Imported object**, which can then be added to the geometry model of the computational domain.

The special objects include:

- **Supergroup**, which are used to control calculations on the results of the computational domain and the boundaries of the subregions therein

- **Sets of sensors** (these objects present sets of several mathematical points, on which you can calculate **Characteristics**)
- the **Movement** element, which sets translation motion and rotation of the standard object of finite volume or imported object (if a **Moving body** modifier it is not specified on it).

Specifics of Objects created in tabs «Preprocessor» and «Postprocessor»

Objects are created, displayed and edited in the folders **Objects** in the **Preprocessor** tab and the **Postprocessor** tab. The order of operations with objects in the tabs **Preprocessor** and **Postprocessor** is the same, but the set of available operations are different.

Objects, created tabs **Preprocessor** and **Postprocessor**, different set of parameters that are displayed in the **Properties** window:

- an object created on the tab **Preprocessor**, is only the basic parameters
- the object (except the object **Supergroup**), created on the tab **Postprocessor**, in addition to basic, there are parameters display object in the **View**.

An **Object** created in the **Preprocessor** tab, is listed also on the **Postprocessor** tab, where parameters of its visualization in the **View** window can be edited (these parameters are not presented in the **Object**'s properties when the **Object** is selected in the **Preprocessor** tab).

In the Project Tree tab **Postprocessor** different icons of the objects created on the **Preprocessor** tab,and the tab **Postprocessor**: icon in the object created in the tab **Preprocessor**, adds an additional element . Below is an example of a display of two objects in the **Project** tab **Postprocessor**: top created in the tab **Preprocessor**, lower - the tab **Postprocessor**:

Object created tab **Postprocessor**, does not appear on the tab **Preprocessor**.

An object can only be removed in the same tab **(Postprocessor** or **Preprocessor**), in which it was created.

Parameters of location of finite volume *Objects* and *Sets of sensors*

Finite volume **Objects** (**Boxes**, **Cones/cylinders**, **Ellipsoids/spheres**, and **Imported objects**) and **Sets of sensors** have in the **Properties** window the same parameters and screen buttons that set their position, orientation, and scaling. These parameters and buttons locate in the **Properties** window in the **Location** group of parameters. A pane with these buttons locates in the **Operations** element.

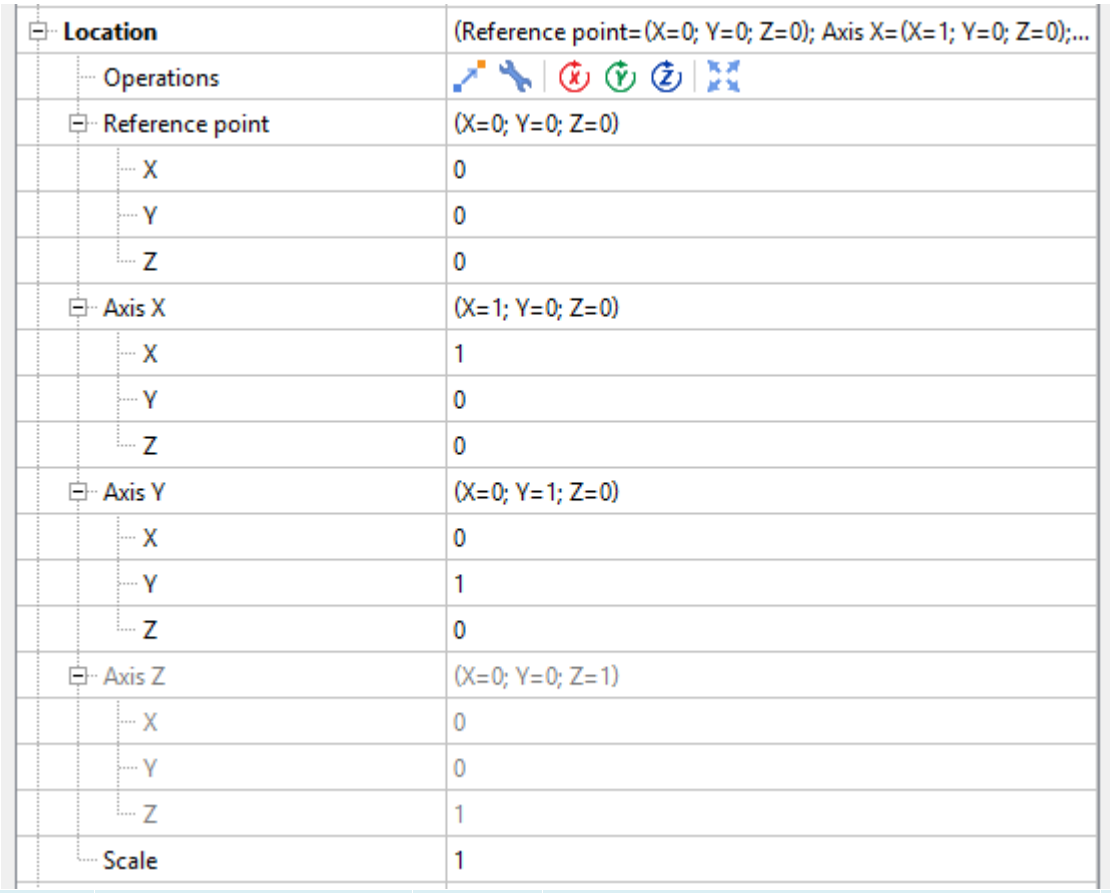
You can use the same functionality to set the initial position of **Moving bodies**.

Objects locate in the space of the computational domain, where an absolute coordinate system (ACS) is specified. Each **Object** also has its own, rigidly attached to it, local coordinate system (LCS).

When an **Object** is **Box**, **Cone/cylinder**, **Ellipsoid/sphere**, **Imported object**, or **Sets of sensors**, you can perform the following operations:

- operation of translation and rotation of the **Object** in the ACS, which are defined as operations of translation and rotation of the LCS relatively to the ACS;
- operation of scaling the **Object** so all the LCS coordinates of the **Object**'s facets are multiplied by specified number of times.

Operations and parameters in the **Location** group in properties of a **Set of sensors** are not applied to *surface* **Sensors** (that are linked to surfaces of geometry models).
Surface **Sensors** change their locations together with the surfaces, to which the are connected; along with this changes coordinates of such **Sensors** in the LCS (local coordinate system) of the **Set of sensors** are changed.



The **Location** group of parameters in the **Properties** window of a finite volume geometric object (and **Set of sensors**)

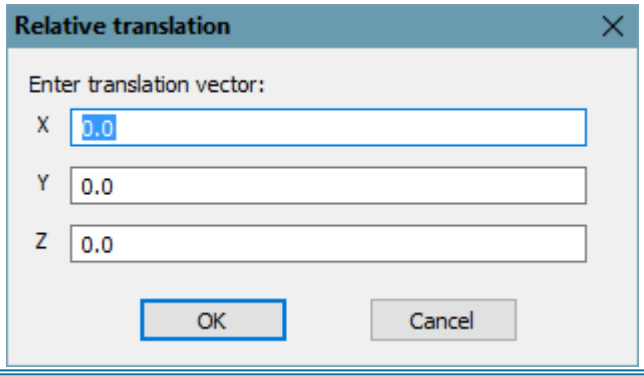
Parameters of an **Object**'s location ("**Location > ...**")

Parameter or button	Description
Operations > (Relative translation)	Opens the **Relative translation** dialog box (see below)
Operations > (Coordinate system adjustment)	Opens the **Coordinate system adjustment** dialog box (see below)
Operations > (Relative rotation around local axis X)	Opens the **Relative rotation around local axis X** dialog box (see below)
Operations > (Relative rotation around local axis Y)	Opens the **Relative rotation around local axis Y** dialog box (see below)
Operations > (Relative rotation around local axis Z)	Opens the **Relative rotation around local axis Z** dialog box (see below)
Operations > (Relative scaling)	Opens the **Relative scaling** dialog box (see below)
Reference point > X	Coordinates of the reference point (the center) of the **Object** in the ACS. In the **View** window, unit vectors of the the LCS are displayed in this point.
Reference point > Y	
Reference point > Z	
Axis X > X	Coordinates (in the absolute coordinate system) of the X-axis unit vector of the LCS. After clicking the **Apply** button the program automatically carries out normalization of the LCS' unit vectors.
Axis X > Y	
Axis X > Z	

Parameter or button	Description
Axis Y > X	Coordinates (in the absolute coordinate system) of the Y-axis unit vector of the LCS. After clicking the **Apply** button the program automatically carries out normalization of the LCS' unit vectors.
Axis Y > Y	
Axis Y > Z	
Axis Z > X	Calculated coordinates (in the absolute coordinate system) of the Z-axis unit vector of the LCS. Information in these fields is read-only.
Axis Z > Y	
Axis Z > Z	
Scale	Scaling factor of the **Object**

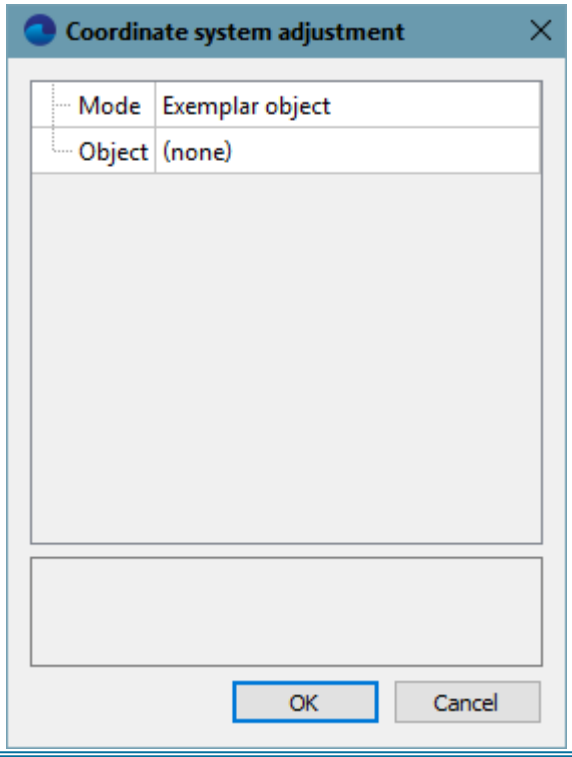
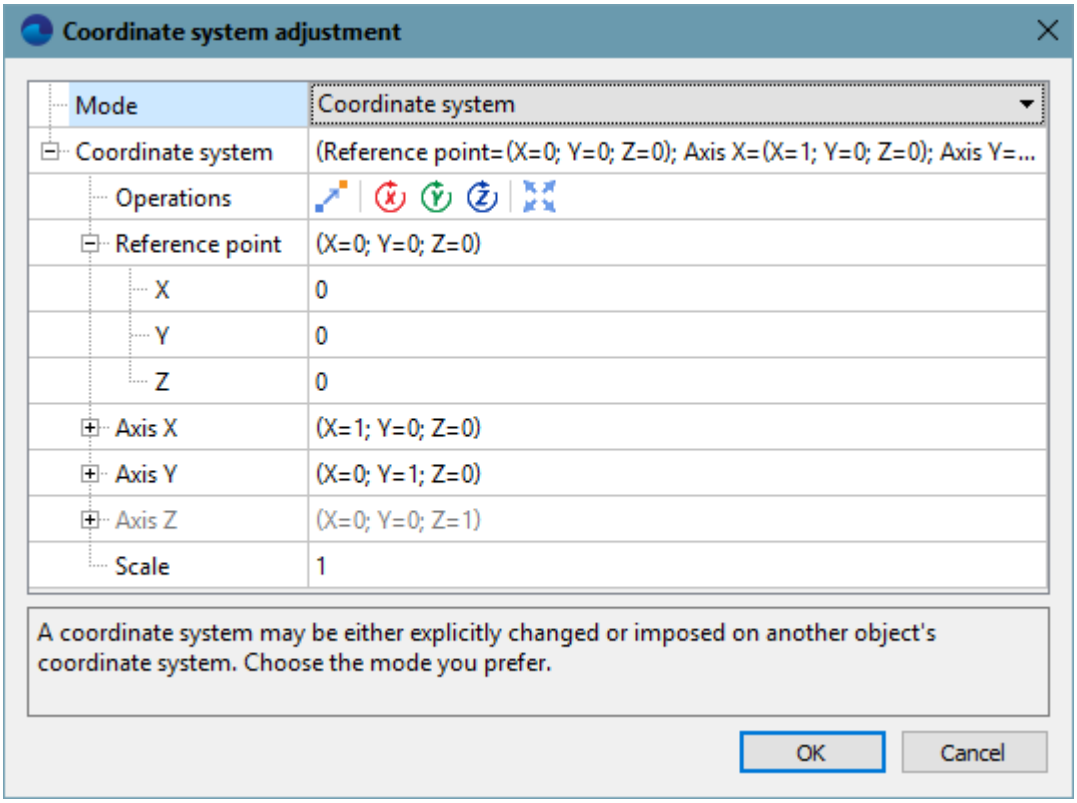
If a **Movement** is set on an **Object**, then the **Object**'s location parameters are not available, but you can use the similar parameters of the **Movement** that specify location of the **Object** at the initial time moment of the **Movement** (these parameters locate in the **Movement**'s **Properties** window in the **Initial position** group of parameters).

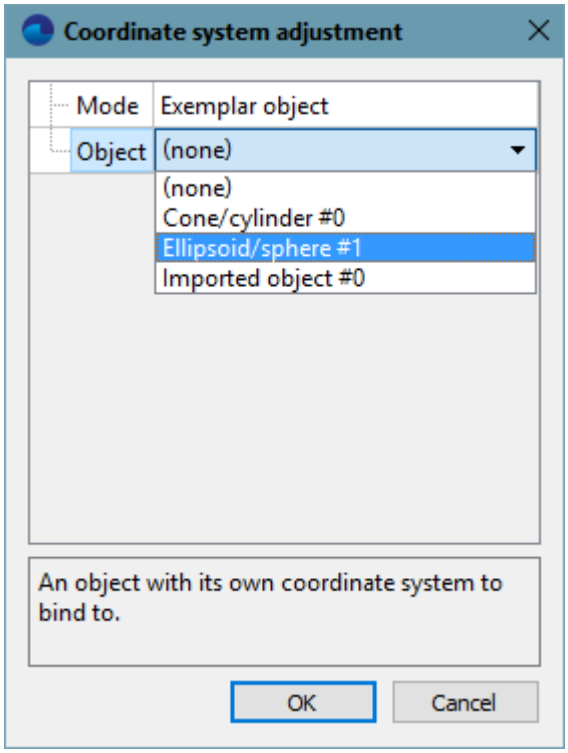
Specifying a relative shift of an Object

Step	Actions
1	In the **Properties** window, click **[Object >] Location > Operations >** (**Relative translation**). The **Relative translation** dialog box will open: 
2	Specify components of the object's translation vector in the absolute coordinate system and then click **OK**. In the **Properties** window, coordinates of the **Object**'s **Reference point** will change.
3	In the **Properties** window, click the **Apply** button to activate your changes. Position of the **Object** in the **View** window will change.

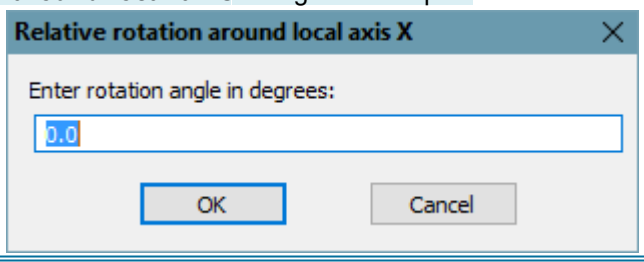
Adjustment of an Object's LCS

Step	Actions
1	In the **Properties** window, click **[Object >] Location > Operations >** (**Coordinate system adjustment**). The **Coordinate system adjustment** dialog box will open, using which you can change the local coordinate system (LCS) of the **Object**:

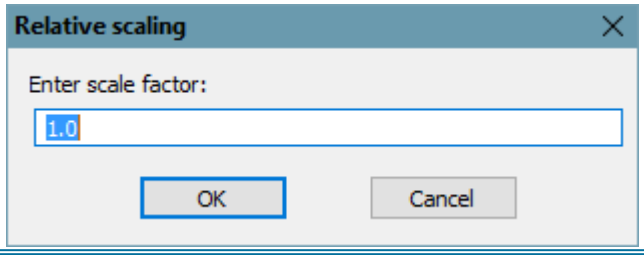
Step	Actions
	
2a	You can either specify the new LCS either explicitly or copy it from the LCS of another **Object**. To specify the LCS explicitly, select **Mode = Coordinate system** and specify parameters of the coordinate system (you can also use screen buttons of operations):  After specifying parameters of the LCS, click **OK**. In the **Properties** window, parameters of the **Object**'s location will change.
2b	To copy the LCS of another **Object**, select **Mode = Exemplar object** and select the desired exemplar object from the **Object** field:

Step	Actions
	 You can select an **Object** from the list of geometry **Objects** (Boxes, Cones/cylinders, Ellipsoids/spheres, Imported objects, Sets of sensors) that are specified in either **Preprocessor** or **Postprocessor**. You *cannot* select neither Movements nor Moving bodies. Then click **OK**. In the **Properties** window, parameters of the **Object**'s location will change.
3	In the **Properties** window click **Apply** to activate your changes. Location of the **Object** in the **View** window will change.

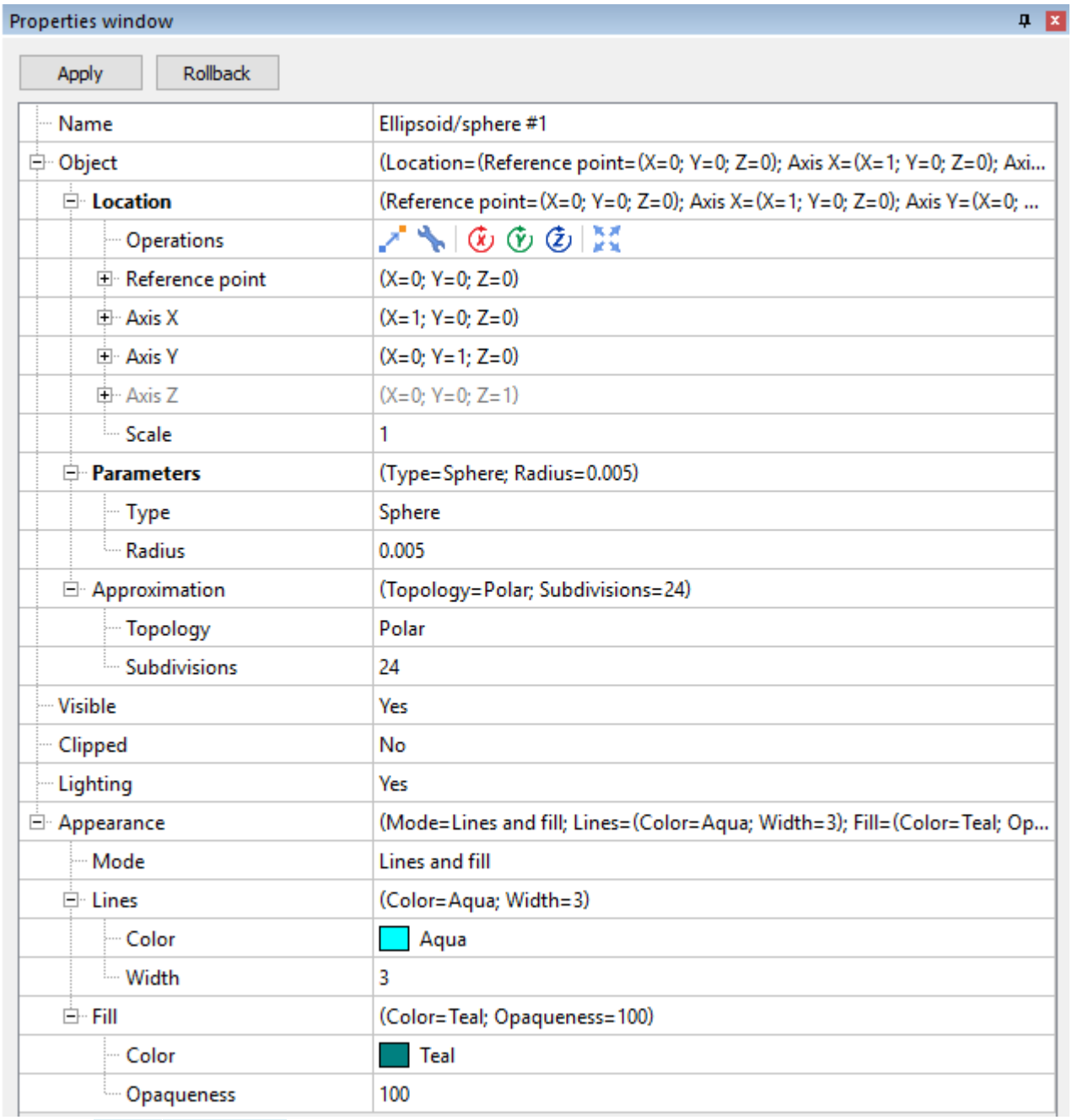
Specifying a relative rotation of an Object around an LCS axis

Step	Actions
1	In the **Properties** window, click **[Object >] Location > Operations >** (, ,) (**Relative rotation around local axis**). The **Relative rotation around local axis** dialog box will open: 
2	Specify rotation angle around the axis X (Y, Z) of the local coordinate system, and then click **OK**. In the **Properties** window, components of some LCS' unit vectors (**Axis X > ...**, **Axis Y > ...**, **Axis Z > ...**) will change.
3	In the **Properties** window, click the **Apply** button to activate your changes. Orientation of the **Object** in the **View** window will change.

Specifying the relative scaling of an Object

Step	Actions
1	In the **Properties** window, click **[Object >] Location > Operations >** (**Relative scaling**). The **Relative scaling** dialog box will open: 
2	Specify the factor by which all LCS coordinates of the **Object** will be multiplied, and then click **OK**. In the **Properties** window, the value of the **Scale** parameter will change.
3	In the **Properties** window, click the **Apply** button to activate your change. Size of the **Object** in the **View** window will change.

8.1.8.1.1 General properties of Objects



Object's properties (some options are available only in the tab **Postprocessor** project tree)

This section describes general parameters that are properties in properties of all or many types of **Objects**.

Object's parameters are displayed and specified in the **Properties** window. Some parameters can be specified using the mouse (see section *"Changing some properties of an Object using the mouse"* below).

General parameters of **Objects**

Parameter or screen button	Description
Name	Name of the **Object**. It is generated automatically, but you can change it if you wish.
Location	See subsection *"Parameters of location of finite volume Objects and Sets of sensors"* in the section Objects in the project tree. If a **Movement** is set on an **Object**, then the **Object**'s location parameters are not available, but you can use the similar parameters of the **Movement** that specify location of the **Object** at the initial time moment of the **Movement** (these parameters locate in the **Movement**'s **Properties** window in the **Initial position** group of parameters).

Parameter or screen button	Description
	Operations and parameters in the **Location** group in properties of a **Set of sensors** are not applied to *surface* **Sensors** (that are linked to surfaces of geometry models). *Surface* **Sensors** change their locations together with the surfaces, to which the are connected; along with this changes coordinates of such **Sensors** in the LCS (local coordinate system) of the **Set of sensors** are changed.
Operations	Screen buttons to easily perform actions that change location of the **Object**: • (**Relative translation**) moves the **Object** relatively to its current position in the ACS (absolute coordinate system). This operation changes values of parameters **Reference point > ...** (see below). • (**Coordinate system adjustment**) changes the **Object**'s LCS (local coordinate system) using the **Coordinate system adjustment** dialog box, which opens. This operation changes values of other parameters **Location > ...** (see below). • (**Relative rotation around local axis X**) rotates the **Object** around the X axis of its LCS (local coordinate system). This operation changes values of parameters **Axis Y > ...** and **Axis Z > ...** (see below). • (**Relative rotation around local axis Y**) rotates the **Object** around the Y axis of its LCS (local coordinate system). This operation changes values of parameters **Axis X > ...** and **Axis Z > ...** (see below). • (**Relative rotation around local axis Z**) rotates the **Object** around the Z axis of its LCS (local coordinate system). This operation changes values of parameters **Axis X > ...** and **Axis Y > ...** (see below). • (**Relative scaling**) scales the **Object** relatively to the current size relatively to the center of its LCS (local coordinate system). This operation changes the **Scale** parameter (see below). You can cancel all your changes (including those that were done at previous uses of this operation) by setting **Scale=1**.
Reference point > X	Coordinates of the origin of the **Object**'s LCS (local coordinate system) in the ACS (absolute coordinate system). For most **Objects**, this point is their geometric center. By default, the **Reference point** locates in the center of the computational domain and coincides with the origin of the ACS (absolute coordinate system).
Reference point > Y	
Reference point > Z	
Axis X > X	Coordinates of the X-axis unit vector of the **Object**'s LCS (local coordinate system) in the ACS (absolute coordinate system).
Axis X > Y	
Axis X > Z	
Axis Y > X	Coordinates of the Y-axis unit vector of the **Object**'s LCS (local coordinate system) in the ACS (absolute coordinate system).
Axis Y > Y	
Axis Y > Z	
Axis Z > X	Coordinates of the Z-axis unit vector of the **Object**'s LCS (local coordinate system) in the ACS (absolute coordinate system).
Axis Z > Y	
Axis Z > Z	
Scale	The scaling factor of the **Object** relating to the origin of its LCS (local coordinate system). This scaling factor is applied to geometry parameters of the **Object** that are set relating to the **Object**'s LCS.
Visible	• **Yes** = The **Object** is always displayed in the **View** window. • **No** = The **Object** is displayed in the **View** window only when it is selected in the project tree.
Clipped	• **Yes** = The **Object** is cut by clipping **Planes**. • **No** = clipping **Planes** do *not* affect the **Object**.

Parameter or screen button	Description
Lighting	This parameter allows you to toggle lighting of the **Object**.
Appearance	
Mode	Method of displaying the **Object**: • **Lines** = only wireframe lines of the **Object** are visible • **Fill** = only surface of the **Object** is visible • **Lines and fill** = both wireframe lines and surface of the **Object** are visible
Lines	
Color	Color of the **Object**'s wireframe lines
Width	Thickness of the **Object**'s wireframe lines
Fill	(Object **Line** does not have these parameters)
Color	Color of the **Object**'s surface
Opaqueness	Opaqueness of the **Object**'s surface

Changing some properties of an Object using the mouse

In interactive mode (using the mouse), you can dynamically change parameters of geometric **Objects** and so do:

- changing position of the **Object**'s reference point (the reference point is displayed as a small ball)
- changing direction of the **Object**'s directing vector or orientation of the **Object**'s LCS (local coordinate system)
- shifting along the directing vector
- scaling

Use the  (**Enable edit mode for selected object**) tool in the toolbar **Work modes**. See section Changing position, orientation and scale of an Object by the mouse for details.

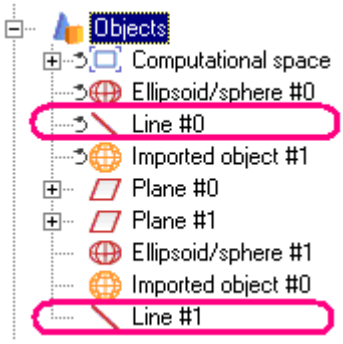
8.1.8.1.2 Standard Objects

Standard geometrical objects are **Objects** of the following types:

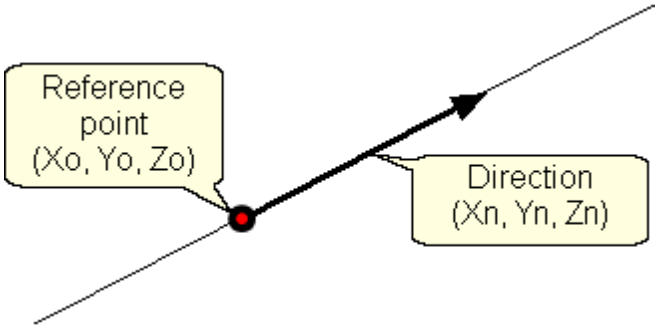
- **Line**
- **Plane**
- **Box**
- **Cone/cylinder**
- **Ellipsoid/sphere**

In some situations the **Computational space** object can also be used as a standard geometrical object.

8.1.8.1.2.1 Object «Line» (user interface)



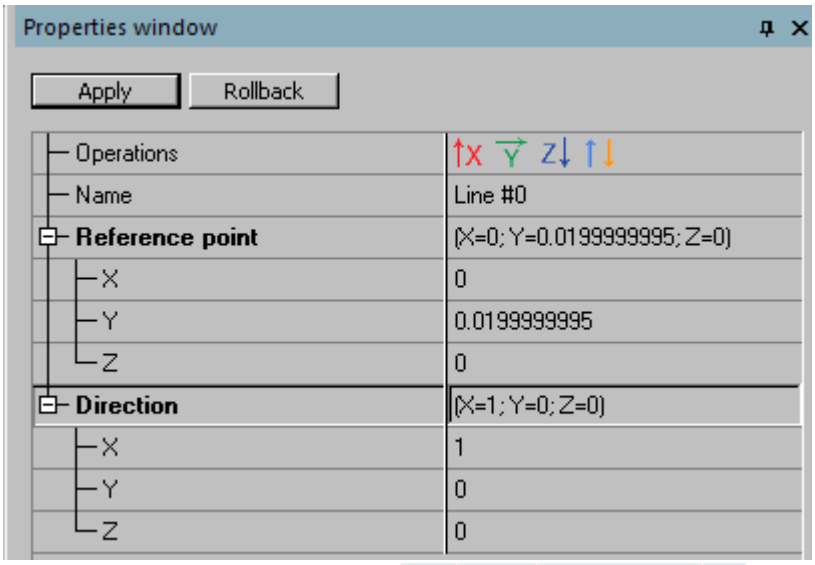
Objects **Line** in the project tree



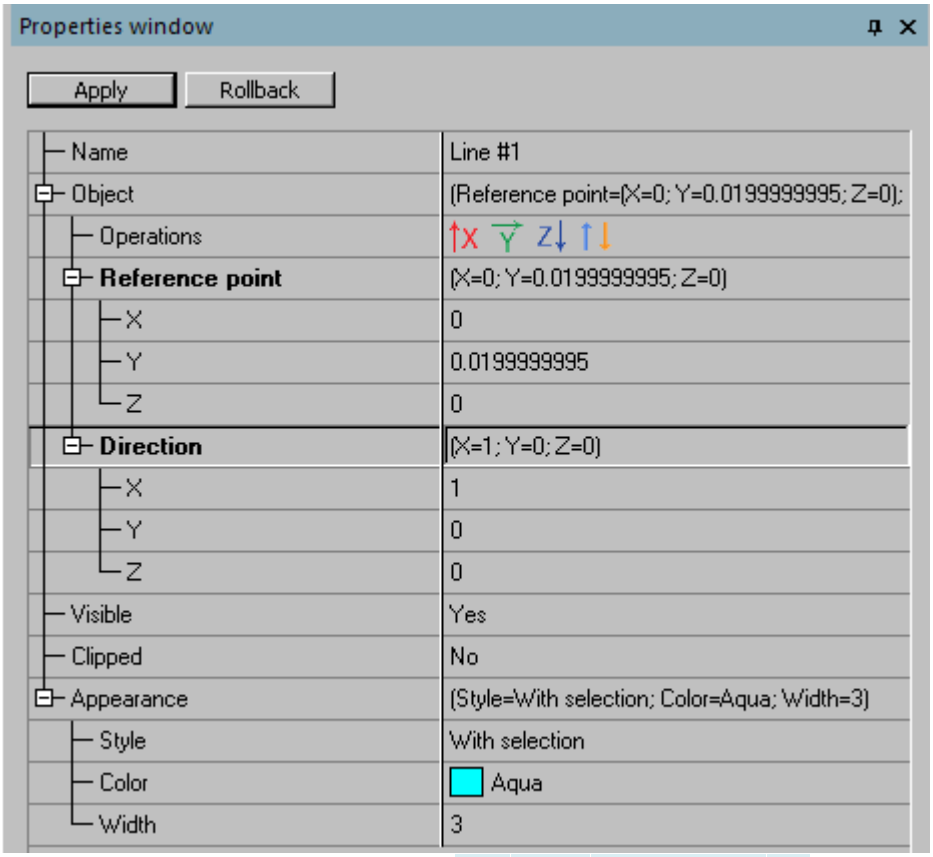
A **Line** object is defined by its reference point and direction

Operations on the object **Line** are performed using the context menu that appears in the project tree, and using screen buttons in the **Properties** window.

Parameters of object "Line"



Properties window of object **Line** in the **Preprocessor** tab



Properties window of object **Line** in the **Postprocessor** tab

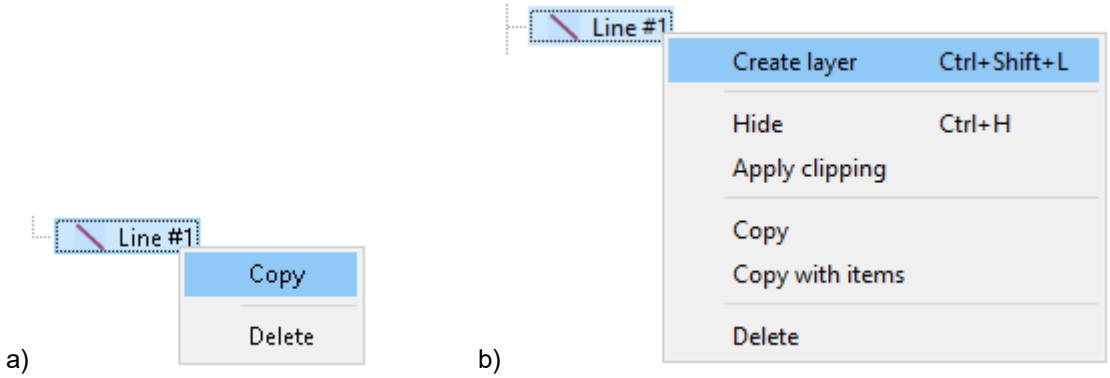
Parameters of object **Line**

Parameter screen buttons	Description
Name	Name of the **Object** in the project tree. The name is generated automatically as **Line #N** and you can change it if you wish.
Object	
Operations	Screen buttons to quickly change the orientation and direction of the **Line**: • – orient the **Line** along axis **X** • – orient the **Line** along axis **Y** • – orient the **Line** along axis **Z** • - invert direction of the **Line** Pressing the button again , , or inverts the direction of the **lines** (or counterclockwise along the axis).
Reference point > X	Cartesian coordinates X_0, Y_0 and Z_0 reference point on the line. The reference point is the center of rotation of the line. By default, the reference point is taken as the center of the computational domain.
Reference point > Y	
Reference point > Z	
Direction	Direction **lines** in three-dimensional space. After pressing the **Apply** button performs self-normalized direction vector **lines**.
Direction > X	Xn - a component of the direction vector along the X axis
Direction >Y	Yn - a component of the direction vector along the Y axis
Direction >Z	Zn - a component of the direction vector along the Z axis

Parameter screen buttons	Description
Visible	See General parameters of objects.
Clipped	
Appearance	
Style	Display style line: • **Solid** - the object appears with a bold line in the whole **space** ("box"). For this choice of the construction **lines** faster. • **With selection** - the object is displayed by a thick line within the computational domain, regardless of the computational domain to be displayed by a thin line. Thus, the fatty **line segments** correspond to the regions located in the computational domain. For this choice of construction **lines** is slower.
Color	Color of the **Line**
Width	Thickness of the **Line**

Note: The location of some parameters in groups in the tabs **Preprocessor** **Postprocessor** and may be slightly different.

Context menu of the "Line" object



a) b)

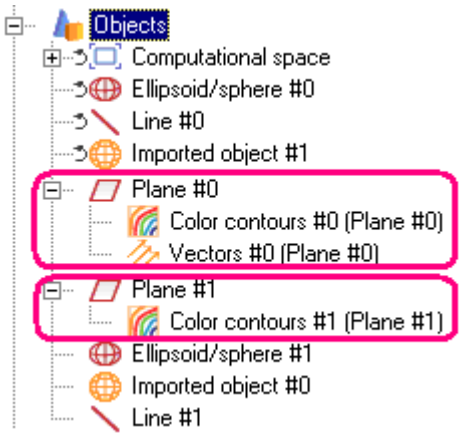
Context menu in the project tree on the **Line** object:
a) on the tab **Preprocessor**; b) - on the tab **Postprocessor**

The context menu of the **Line** object in the project tree

Menu item	Description
Create layer[*]	Creates a new **Layer** on this **Line**. This command is duplicated by the **Ctrl+Shift+L** hot key (by default; you can change this hot key).
Hide[*]	☐ - the **Object** is always displayed in the **View** window ☑ - the **Object** is displayed in the **View** window, only when it is selected in the project tree The **Visible** parameter in the **Properties** window will change accordingly. This command is duplicated by the **Ctrl+H** hot key (by default; you can change this hot key).
Apply clipping[*]	☐ - clipping **Planes** do not affect the **Object** ☑ - clipping **Planes** cut the **Object** The **Clipped** parameter in the **Properties** window will change accordingly.
Copy	Creating an element, which is a copy of the selected element
Copy with items[*]	Copy the selected object with the child elements
Delete	Deleting the selected element from the project tree

[*] these commands are only available in the context menu in **Postprocessor**

8.1.8.1.2.2 Object «Plane» (user interface)

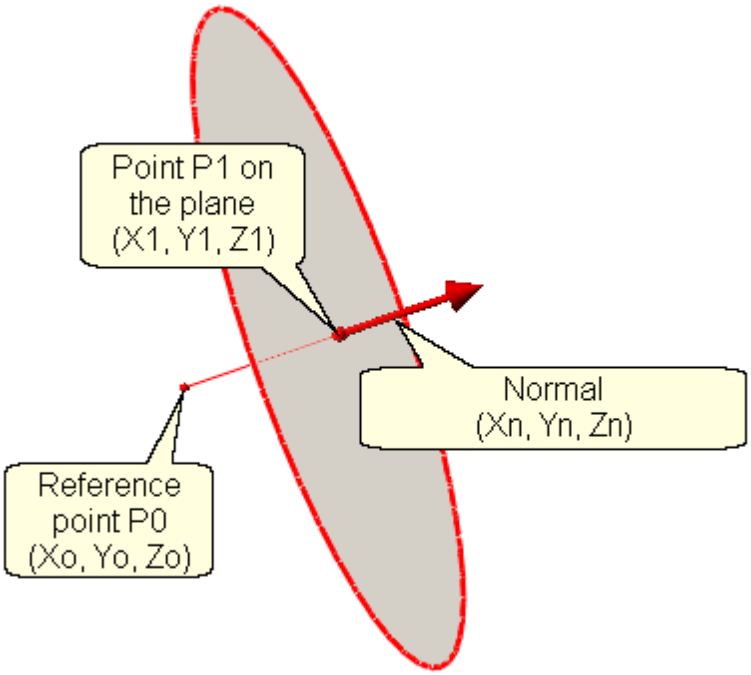


Objects **Plane** in the project tree

A **Plane** object is defined by the following parameters:

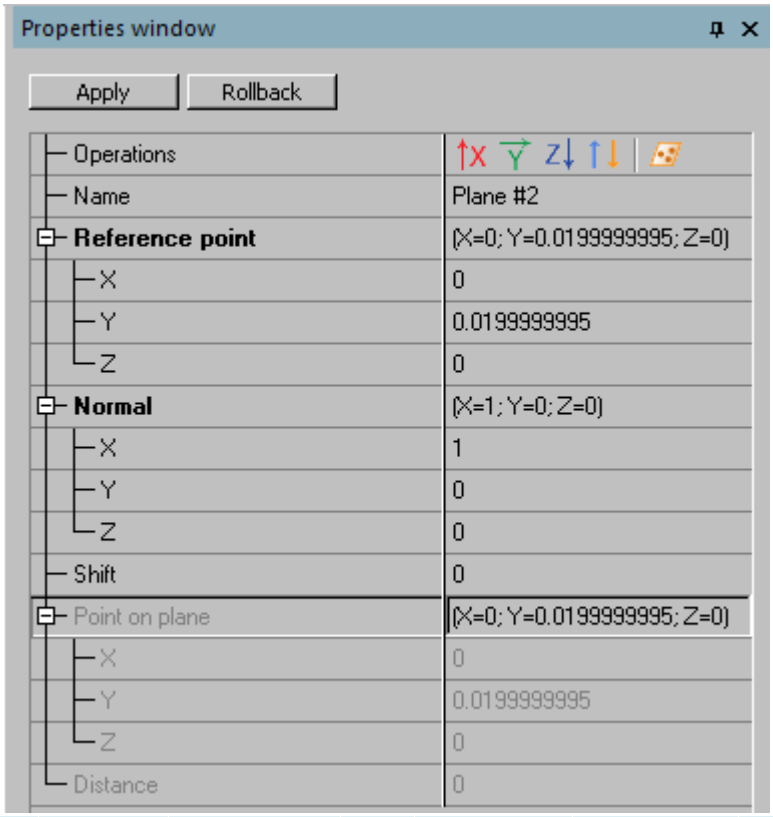
- *reference point*
- *direction of the normal*
- *shift* of the plane relating to its reference point (the shift directed along the normal moves the P0 point to the point P1, which belongs to the plane)

Operations on the object **Plane** are performed using context menu items to be opened in the project tree, and with the help of on-screen buttons in the **Properties** window.

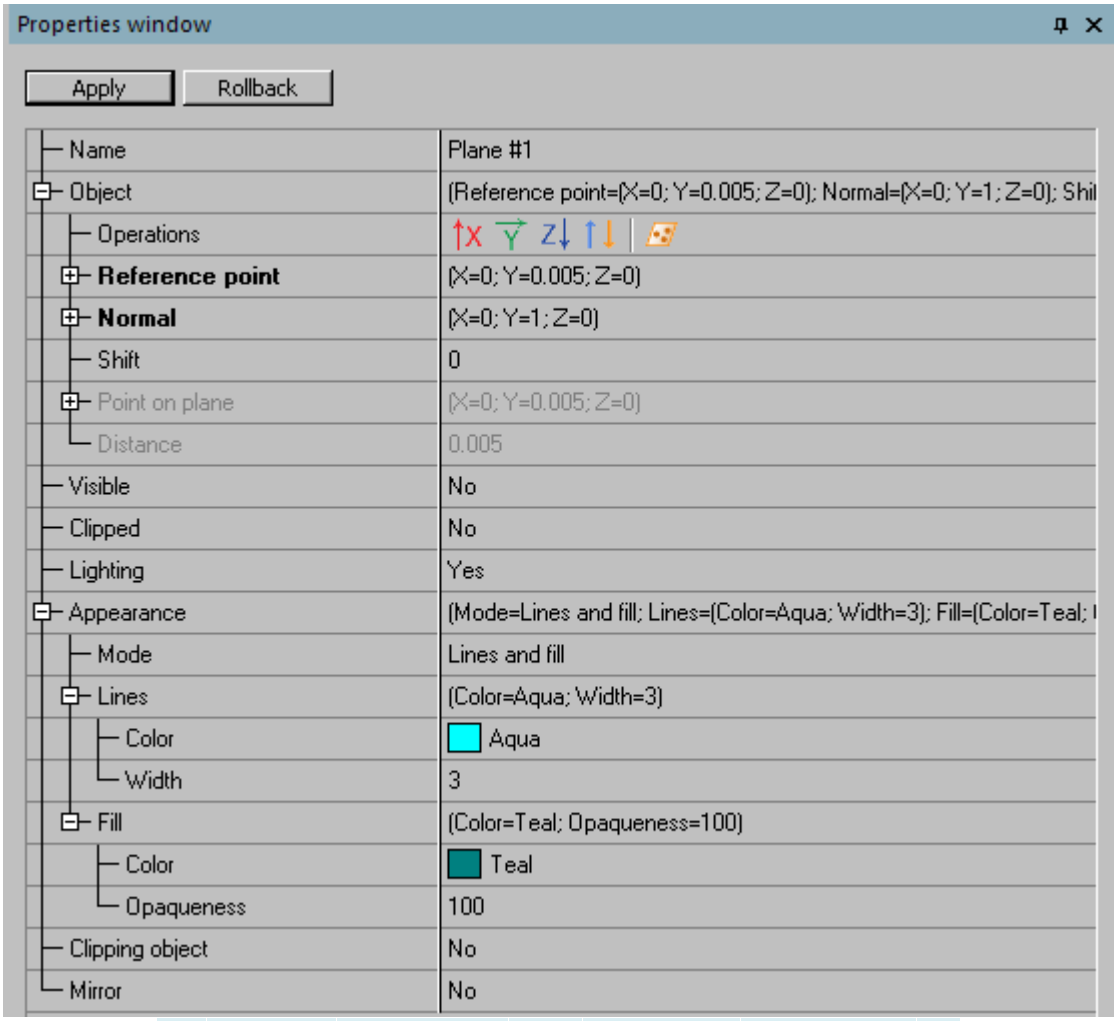


A **Plane** is defined by its reference point, normal vector and shift

Parameters of object «Plane»



The **Properties** window of the **Plane** object in the **Preprocessor** tab



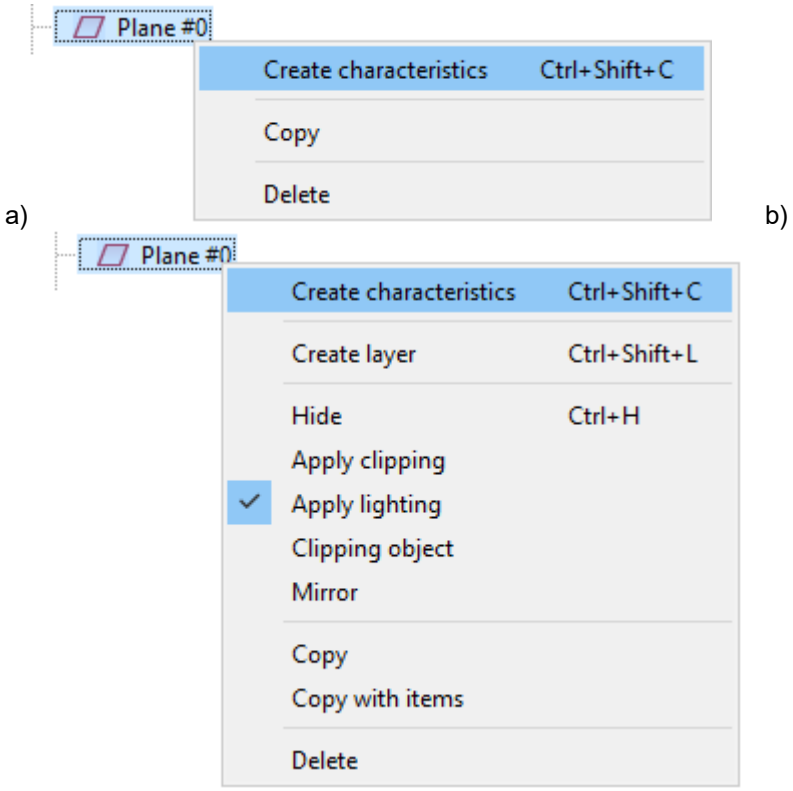
The **Properties** window of the **Plane** object in the **Postprocessor** tab

Parameters of object **Plane**:

Parameter button	Description
Operations	Screen buttons to quickly change the orientation and direction of the normal vector of the **Plane**: • (**Along X**): orienting the normal of the **Plane** along axis **X** • (**Along Y**): orienting the normal of the **Plane** along axis **Y** • (**Along Z**): orienting the normal of the **Plane** along axis **Z** • (**Invert**): inverting direction of the normal of the **Plane** • (**By 3 points**): specifying the **Plane** by three points (the **Plane definition by three points** dialog box will open, see subsection *"Plane definition by three points"* below) Clicking the (, ,) button again inverts the direction of the **Plane**'s normal vector.
Name	Name of the **Object** in the project tree. The name is generated automatically as **Plane#N** and you can change it if you wish.
Reference point> X	Coordinates X_0, Y_0 and Z_0 of the **Plane**'s **Reference point** (P0). A **Plane** can either pass through the **Reference point** or not pass through it (in the latter case the **Plane** is shifted relatively the **Reference point** on the value of the parameter **Shift**, see below). In the **View** the **Reference point** is displayed in the form of a ball and the center of rotation and the object shift performed when interactive editing.

Parameter button	Description
Reference point> Y **Reference point> Z**	By default, the **Reference point** is the center coordinates of the computational domain.
Normal > X **Normal > Y** **Norma l> Z**	Components of X_n, Y_n, Z_n of the normal vector of the **Plane**. The normal vector is displayed In the **View** window as an arrow. After clicking the **Apply** button the program performs automatic normalization of the vector.
Shift	Displacement of the **Plane** along the normal vector relating to the **Reference point**. When **Shift** is non zero, the **Plane** doesn't passes through the **Reference point** (X_0, Y_0, Z_0), i.e. the **Reference point** doesn't lie in the **Plane**.
Point on plane > X **Point on plane > Y** **Point on plane > Z**	Coordinates X_1, Y_1 and Z_1 of the point P1, obtained as a result of shifting the **Reference point** P0 along the normal. Like the **Reference point**, the point P1 is also displayed in the **View** window as a ball. This field in for information only and cannot be edited.
Distance	Calculated distance from the origin of the absolute coordinate system to the **Plane** (along to the normal to the **Plane**). This field in for information only and cannot be edited.
Visible **Clipped** **Lighting** **Appearance > Mode** **Appearance > Lines> Color** **Appearance > Line> Width**	See General properties of Objects.
Clipping object	If this option is enabled, then the space is divided into two half. In the half-subspace, which looks normal plane, clipped by drawing layers and objects allowed in the other - is prohibited. Clipping plane affect all layers, which enabled **Clipped**. • **No**- not a clipping plane • **Yes**- the plane is clipping object Depending on value of this parameter, the **Plane**'s icon in the project tree in the **Postprocessor** tab is: • - for a **Plane**, which *is not* a clipping object • - for a **Plane**, which *is* a clipping object See illustration in section Specific settings of the Plane object.
Mirror	If enabled, the layers all the layers are mirrored with respect to this plane[1,2]. • **No**- not a plane mirror; • **Yes**- the plane is mirrored See illustration in section Specific settings of the Plane object. ***Notes:*** [1] Mirror images of layers work in the same sequence in which they were set [2] In one project may not work more than three mirror images

Context menu of the «Plane» object



Context menu of the **Plane** object in the project tree:
a) in the **Preprocessor** tab, b) in the **Postprocessor** tab

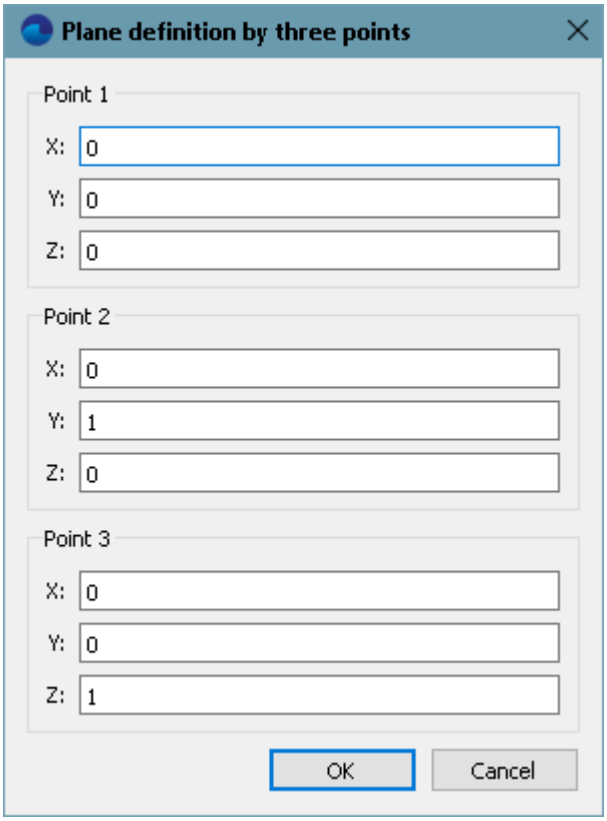
The context menu of the object **Plane** in the project tree:

Menu item	Description
Create characteristics	Creates **Characteristics** on this **Object**. This command is duplicated by the **Ctrl+Shift+C** hot key (by default; you can change this hot key).
Create layer*)	Creating a new **Layer** on this **Plane**. This command is duplicated by the **Ctrl+Shift+L** hot key (by default; you can change this hot key).
Hide*)	☐ - the **Object** is always displayed in the **View** window ☑ - the **Object** is displayed in the **View** window, only when it is selected in the project tree This command is duplicated by the **Ctrl+H** hot key (by default; you can change this hot key).
Apply clipping*)	☐ - clipping **Planes** do not affect the **Object** ☑ - clipping **Planes** cut the **Object**
Apply lighting*)	☐ - the **Object** is *not* lit by light sources ☑ - the **Object** is lit by light sources
Clipping object*)	☐ - The plane is not a cross-section ☑ - the plane is a cutting plane
Mirror*)	☐ - The plane is not a mirror; ☑ - the plane is a mirror
Copy	Creating an element, which is a copy of the selected element
Copy with items*)	Copying the selected object with its child elements
Delete	Deleting the selected element from the project tree

*) These commands are only available in the context menu in **Postprocessor**.

Plane definition by three points

Clicking the button **Operations >**  (**By 3 points**) in properties of a **Plane** opens the **Plane definition by three points** dialog box, which allows you to specify position of the **Plane** by three points, through which it passes.



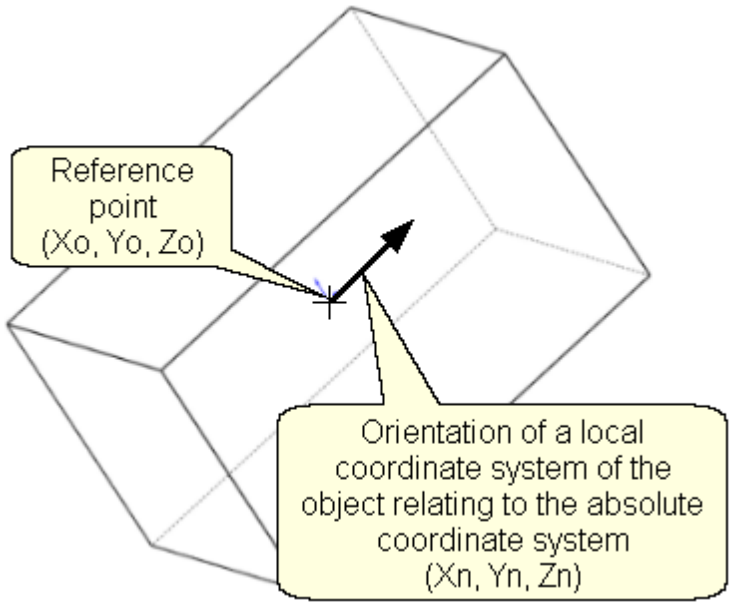
The **Plane definition by three points** dialog box has the following fields:

Point 1		The coordinates of the first point
	X	
	Y	
	Z	
Point 2		Coordinates of the second point
	X	
	Y	
	Z	
Point 3		Coordinates of the third point
	X	
	Y	
	Z	

Enter coordinates of the three points and click **OK**. The **Cancel** button closes the dialog box without entering data.

8.1.8.1.2.3 Object «Box» (user interface)

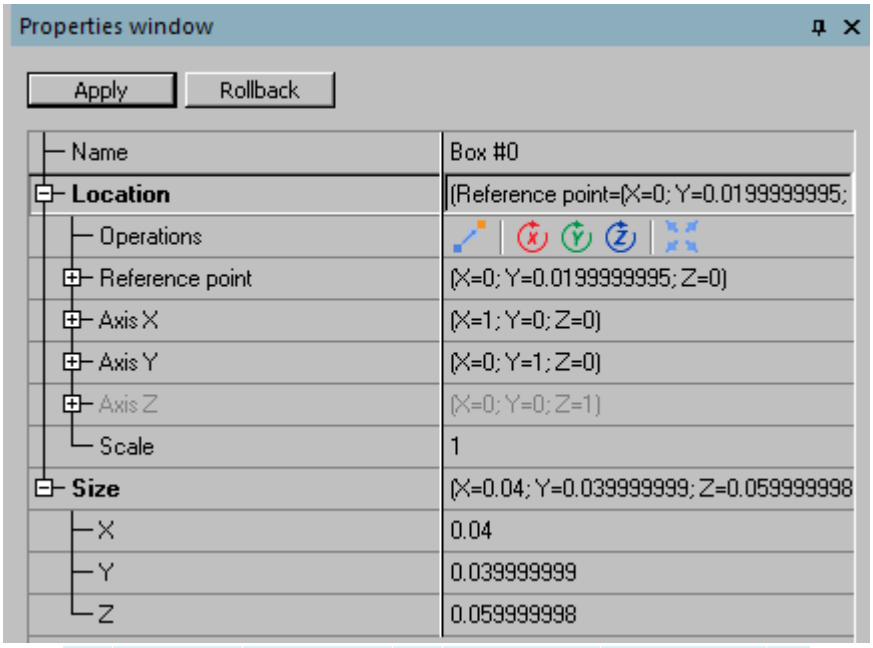
Object **Box** in the project tree



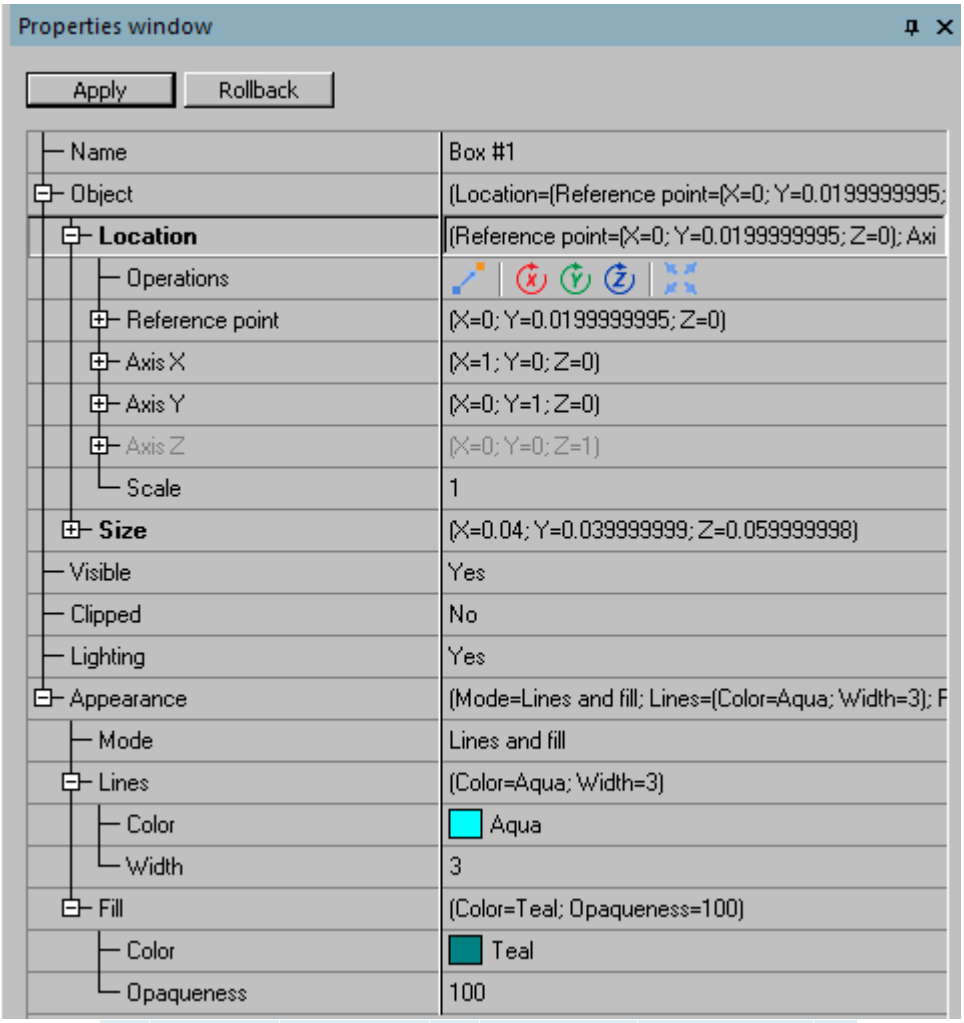
Object **Box**

Parameters of object "Box"

All active parameters may be external parameters.



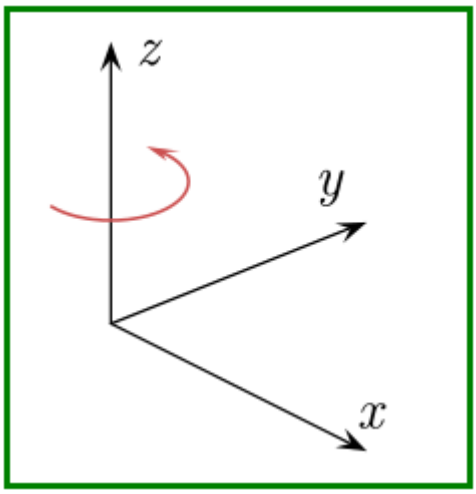
The **Properties** window of a **Box** object in the **Preprocessor** tab



The **Properties** window of a **Box** object in the **Postprocessor** tab

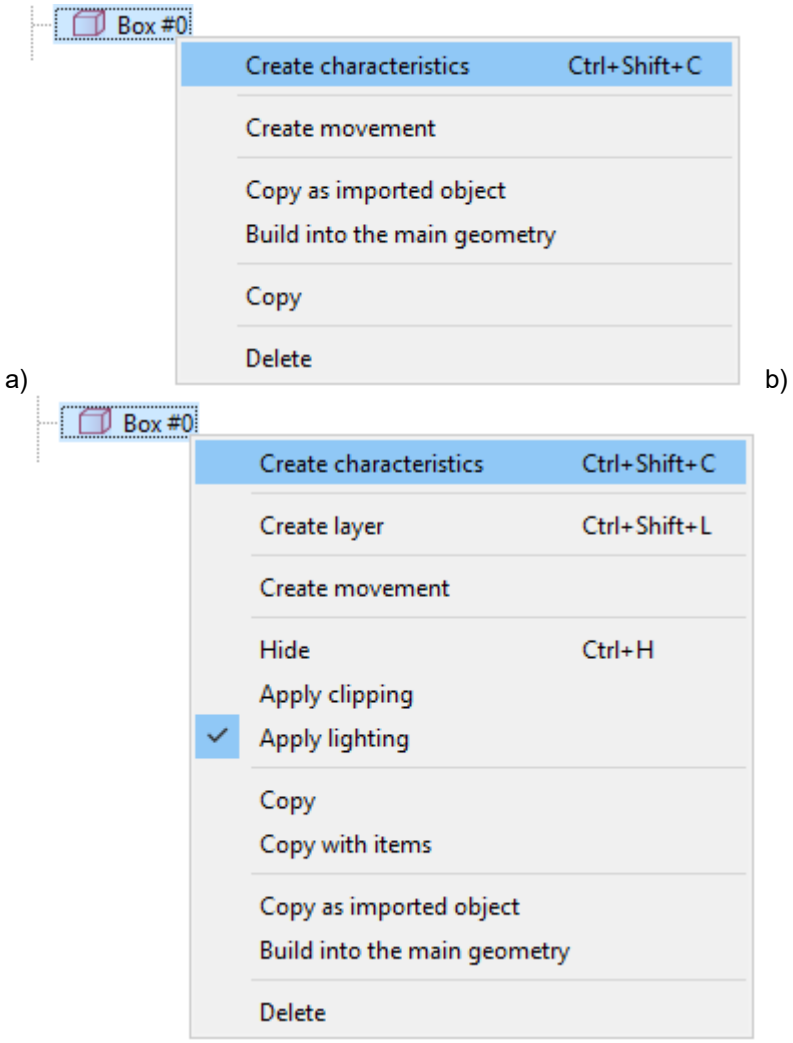
Parameters of a **Box** object (see also General properties of Objects):

Parameter	Description	Dimension
Name	Name of the **Object** in the project tree. The name is generated automatically as **Box#N** and you can change it if you wish.	
Object > Location > Operations	See General parameters of objects.	
Object > Location> Reference point> X	Coordinates of the center of the parallelepiped, defined in the absolute coordinate system (ACS)	[m]
Object > Location > Reference point> Y		
Object > Location> Reference point > Z		
Object > Location> Axis X > X	The orientation of the local coordinate system of the object (LCS-O) givenin the absolute coordinate system (ACS). The orientation of the axes X and Y coordinates are set by the user as vectors (not necessarily a unit length) along these axes, the coordinate axis LCS ACK, but the *check FlowVision orthogonal axes X and Y and, if necessary, corrects the input* data. When entering data for the directions of the axes X and Y, the program automatically adjusts the other axis (Y or X,	

Parameter	Description	Dimension
Object > Location> Axis X > Y **Object > Location> Axis X > Z** **Object > Location> Axis Y > X** **Object > Location> Axis Y> Y** **Object > Location> Axis Y> Z** **Object > Location> Axis Z> X***) **Object > Location> Axis Z> Y***) **Object > Location> Axis Z > Z***)	respectively), and the input data for the Z-axis is not provided at all (it is built automatically by the data entered for the X and Y). After each data entry *FlowVision* automatically normalizes them (replaces on the projection of the unit vectors along the axes). Z-axis orientation is calculated automatically so that it is perpendicular to the X and Y axis and formed with them the right to an orthonormal basis:  Right basis of the coordinate axes	
Object > Location> Scale	Scale factor applied to set the size of the **Box**	
Object > Size> X **Object > Size> Y** **Object > Size> Z**	Dimensions of the **Box** along the axes of the local coordinate system (LCS-O)	[m]
Object > Visible **Object > Clipped** **Object > Lighting**	See General parameters of objects.	
Appearance> ...	See General parameters of objects.	

*) The parameter is calculated automatically and can not be entered by the user.

Context menu of the "Box" object



Context menu of the **Box** object in the project tree:
a) in the **Preprocessor** tab, b) in the **Postprocessor** tab

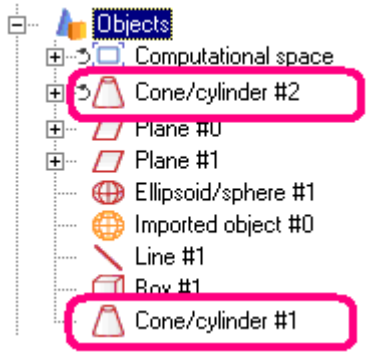
The context menu of the **Box** object in the project tree:

Menu item	Description
Create characteristics	Creates **Characteristics** on this **Object**. This command is duplicated by the **Ctrl+Shift+C** hot key (by default; you can change this hot key).
Create movement	Creating a movement of the **Object**
Create layer*)	Creating a new **Layer** on/in this **Box**. This command is duplicated by the **Ctrl+Shift+L** hot key (by default; you can change this hot key).
Hide*)	☐ - the **Object** is always displayed in the **View** window ☑ - the **Object** is displayed in the **View** window, only when it is selected in the project tree This command is duplicated by the **Ctrl+H** hot key (by default; you can change this hot key).
Apply clipping*)	☐ - clipping **Planes** do not affect the **Object** ☑ - clipping **Planes** cut the **Object**
Apply lighting*)	☐ - the **Object** is *not* lit by light sources ☑ - the **Object** is lit by light sources

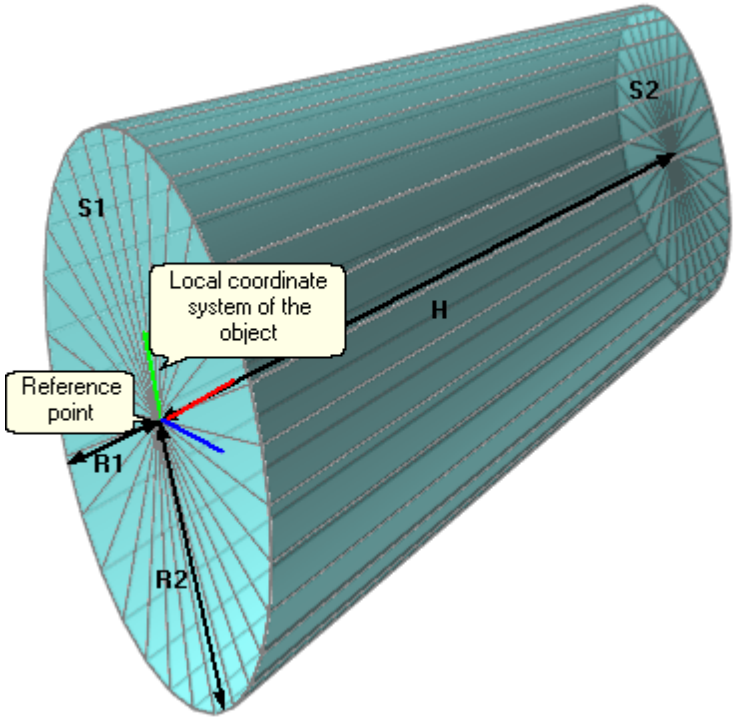
Menu item	Description
Copy	Creating an element, which is a copy of the selected element
Copy with items[*)]	Copying the selected element with its child elements
Copy as imported object	Copying the selected **Object** with converting to an **Imported object**
Built into the main geometry	Add the surface of a selected object in the computational domain
Delete	Deleting the selected element from the project tree

[*)] these commands are only available in the context menu in **Postprocessor**

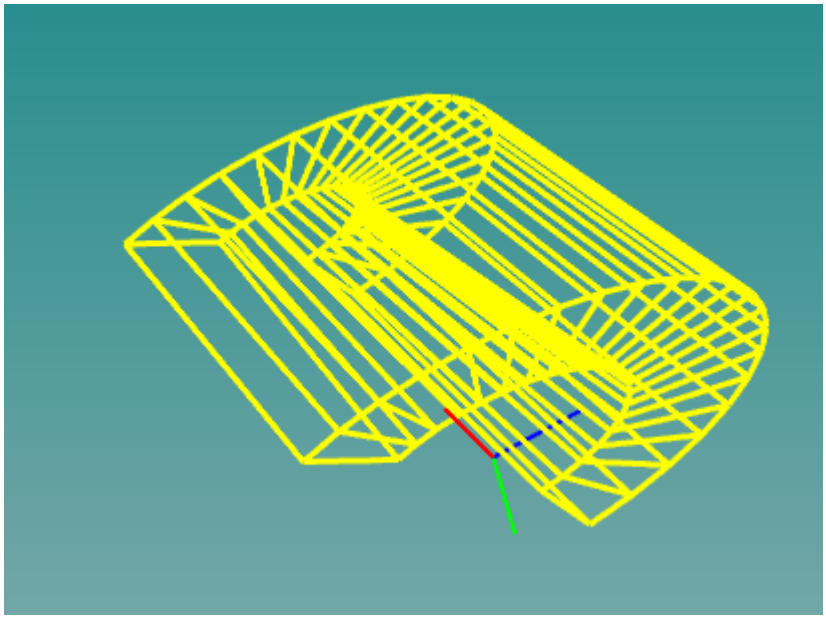
8.1.8.1.2.4 Object «Cone/cylinder» (user interface)



Objects **Cone/cylinder** in the project tree



Reference point, the semi-axis height of the truncated cone bases



Sector of a truncated cone with the channel

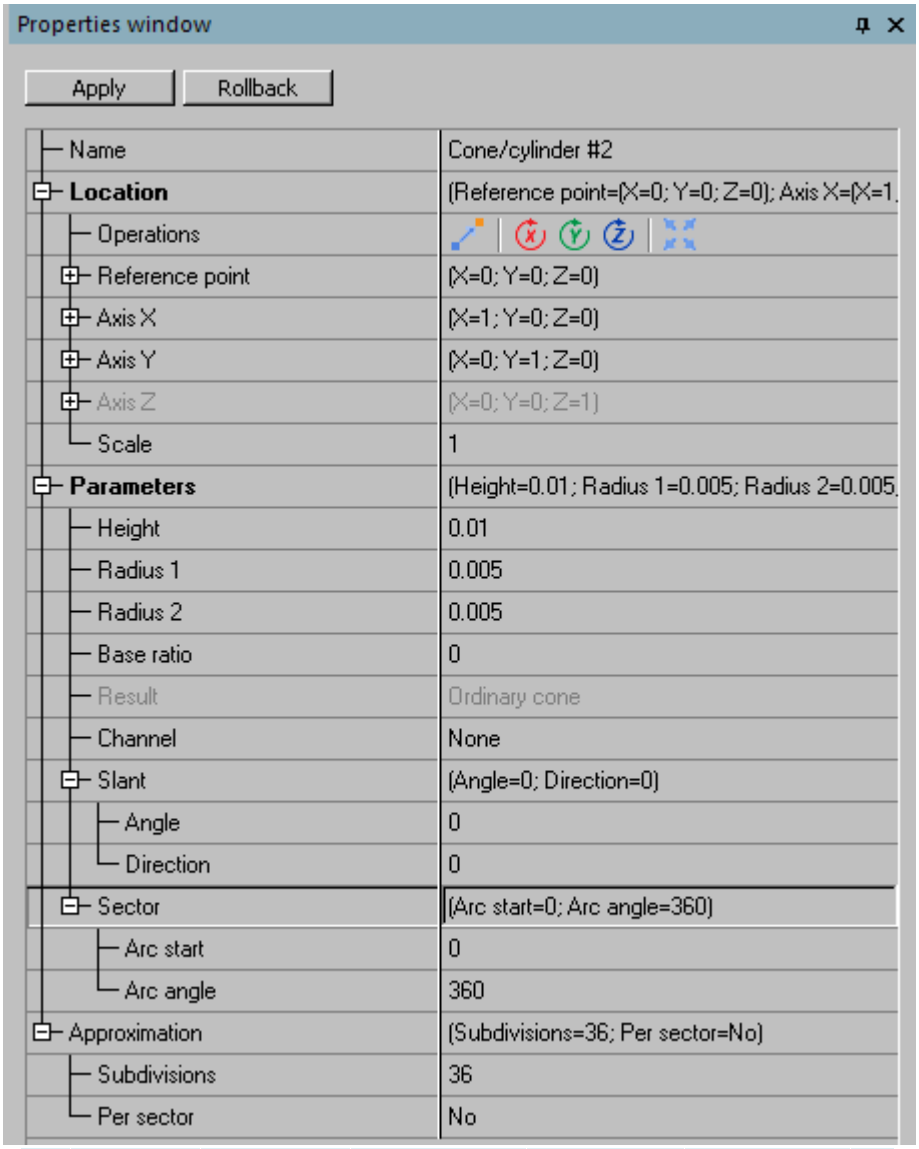
Object **Cone/cylinder** is a truncated cone with two proportional elliptical bases. Larger base is defined by two radii. Smaller base is defined by the ratio of the areas. A special case of the truncated cone are simple cone (the area of the smaller base is 0) and the cylinder (square bases are equal).

Object **Cone/cylinder**has parameters that are displayed in the **Properties** window.

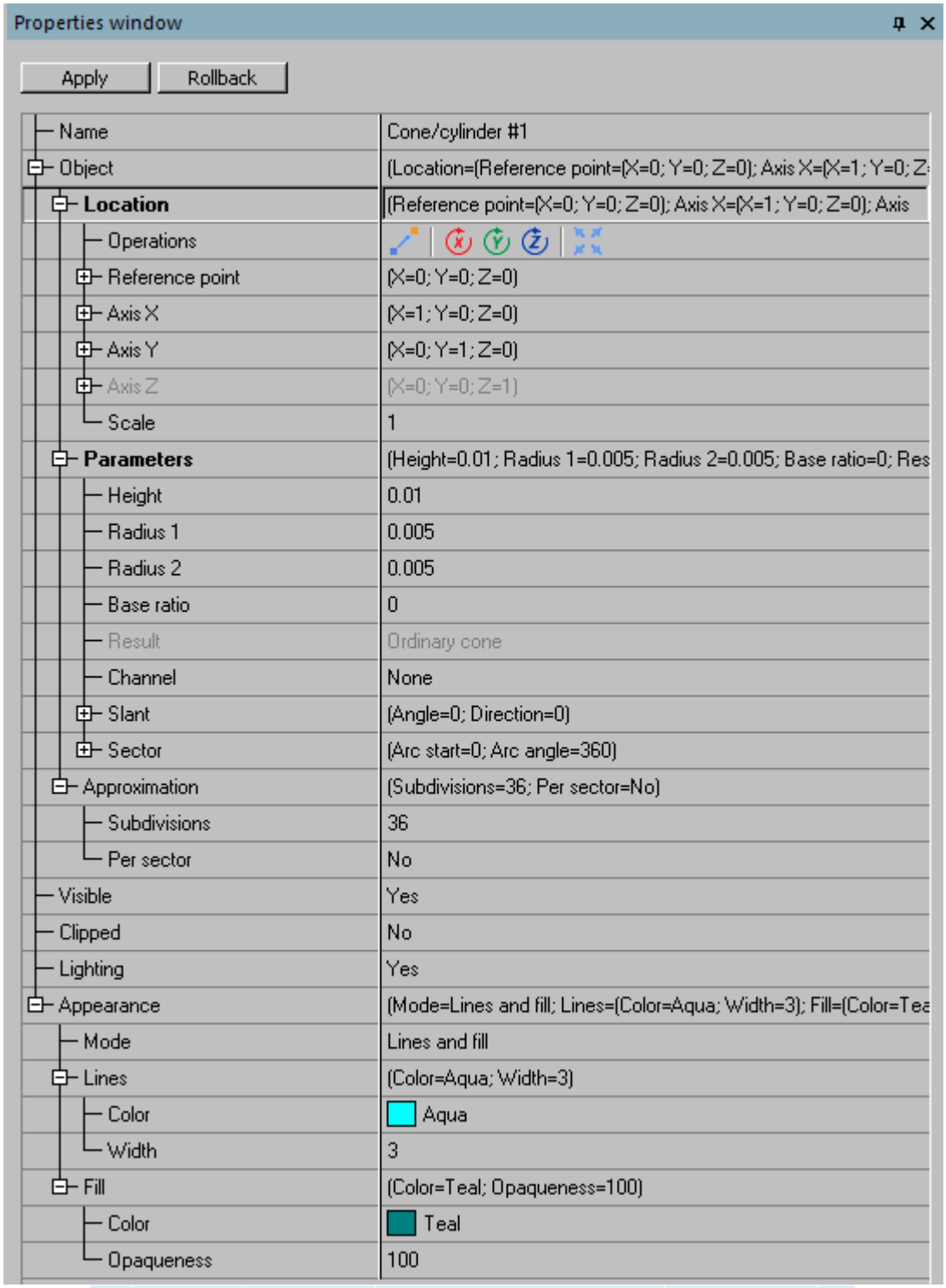
Operations on the object **Cone/cylinder** are performed using context menu items to be opened in the project tree and with the help of on-screen buttons in the **Properties** window.

Object parameters "Cone/cylinder"

All active parameters may be external parameters.



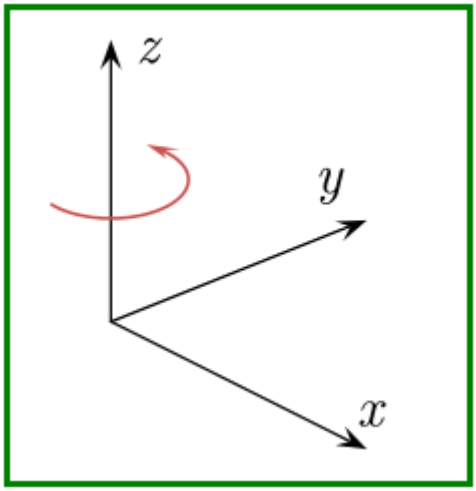
The **Properties** window of a **Cone/cylinder** object in the **Preprocessor** tab



The **Properties** window of a **Cone/cylinder** object in the **Postprocessor** tab

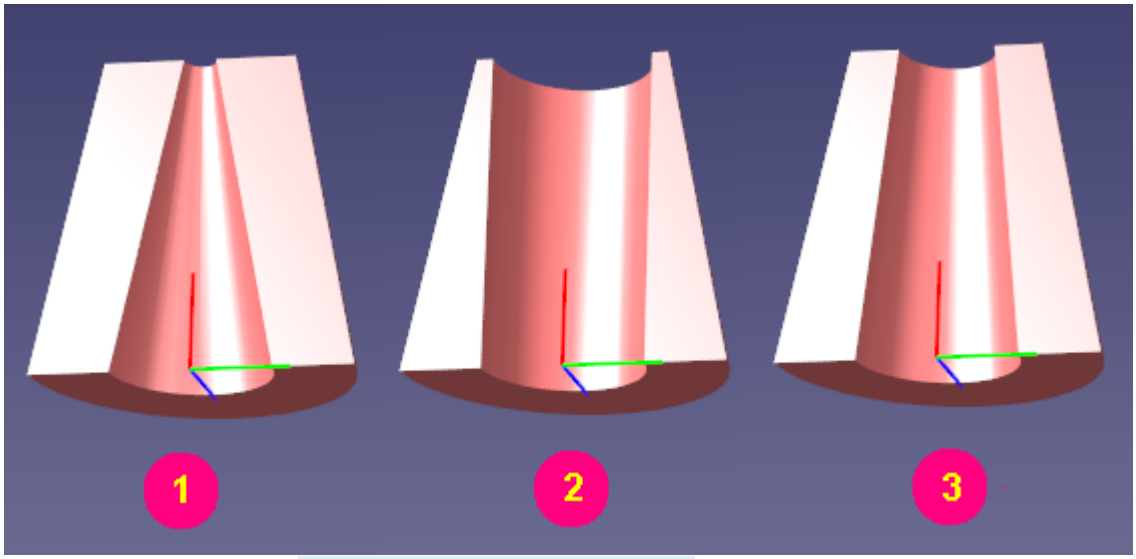
Parameters of object **Cone/cylinder**:

Parameter	Description
Name	Name of the **Object** in the project tree. The name is generated automatically as **Cone/cylinder #N** and you can change it if you wish.
Object > Location> Operations	See General parameters of objects.
Object > Location > Reference point > X	The center of one of the grounds of a truncated cone or cylinder or the center of the (unique) non-truncated cone base,defined in the absolute

Parameter	Description
Object > Location > Reference point > Y	
Object > Location> Reference point > Z	
Object > Location > Axis X > X	The orientation of the local coordinate system of the object (LCS-O) givenin the absolute coordinate system (ACS). The orientation of the axes X and Y coordinates are set by the user as vectors (not necessarily a unit length) along these axes, the coordinate axis LCS ACK, but the *check FlowVision orthogonal axes X and Y and, if necessary, corrects the input* data. When entering data for the directions of the axes X and Y, the program automatically adjusts the other axis (Y or X, respectively), and the input data for the Z-axis is not provided at all (it is built automatically by the data entered for the X and Y). After each data entry *FlowVision* automatically normalizes them (replaces on the projection of the unit vectors along the axes). Z-axis orientation is calculated automatically so that it is perpendicular to the X and Y axis and formed with them the right to an orthonormal basis:  Right basis of the coordinate axes
Object > Location > Axis X > Y	
Object > Location > Axis X > Z	
Object > Location > Axis Y > X	
Object > Location > Axis Y > Y	
Object > Location > Axis Y > Z	
Object > Location > Axis Z > X*)	
Object > Location > Axis Z > Y*)	
Object > Location > Axis Z > Z*)	
Object > Location > Scale	Scale factor applied to set the size of **Cone/Cylinder**
Object > Parameters > Height	The height H of the **Cone/cylinder**, i.e. the distance between the upper and lower base, [m]
Object > Parameters > Radius 1	The values of R_1 and R_2 of the first semi-axes of the elliptic base (that centered on the **Reference point**), [m]
Object > Parameters > Radius 2	
Object > Parameters > Base ratio	The ratio of the corresponding linear dimensions of the two bases of a truncated cone or cylinder. If the object is a non-truncated cone, this parameter is set to 0 (second base shrinks to a point, the top of the non-truncated cone).
Object > Parameters > Result	A shape, which occurs according to the specified **Base ratio**. It can be: • **Cylinder** - when **Base ratio = 1** (i.e., when the bases are of the same size) • **Cone** - when the **Base ratio = 0** (i.e., when the second base shrinks to a point) • **Truncated cone**- when the bases are different and non-zero size (The field of information depends on the value of the parameter **Base ratio** and can not be edited)

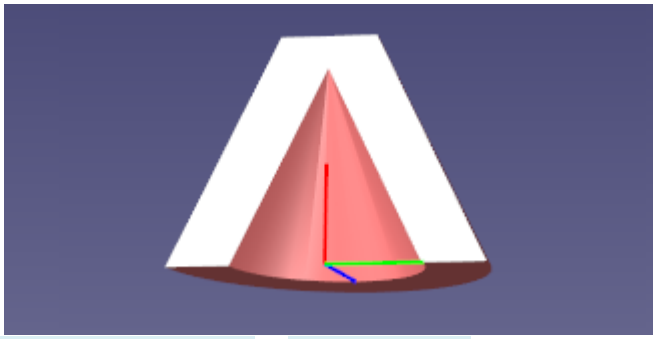
Parameter	Description
Object > Parameters > Channel	The presence of the notch in the **cone/cylinder** in the form of a cone or a truncated cone with an elliptical (s) Reason (s). The coordinate system of the channel and the ratio of the radii of the same, with the same parameters of the **object**. When you set the channel the following options: • **None**- there is no channel • **Constant walls**- the wall thickness **Cone/cylinder** (between the channel and the outer surface of the **cone/cylinder**) is constant (in the cross section perpendicular to the axis of the channel). At certain ratios of the values of this parameter and the parameter **Channel ratio** and the **height of** one of the openings of the channel may be closed (see illustration). • **Constant radii** - radius of the channel is constant (in the cross section perpendicular to the axis of the channel), i.e. channel is elliptical cylinder • **Constant ratio** - the ratio between the radius of the **cone/cylinder** and the radius of the channel is constant along the entire axis of the **cone/cylinder**
Object > Parameters > Channel ratio	The ratio of the linear size of the base of the channel to the base of the **cone/cylinder**, i.e. the ratio of the semi-axis of the ellipse $=r_1$ at the base of the channel to the corresponding semi-axis R_1 ellipse in the base of the **cone/cylinder**.
Object > Parameters > Slant > Angle **Object > Parameters > Slant > Direction**	Slope of the line connecting the base of the **Cone/cylinder** to the first ground plane, [degree], and orientation of the projection of this line on the first base relative to the Y-axis of the local coordinate system **Cone/cylinder**, [degree]
Object > Parameters > Sector > Arc start **Object > Parameters > Sector > Arc angle**	Parameters sector, cut out from the **Cone/cylinder**. Defined angle sector at the starting point of the LCS-O (about axis Y), [degree], and the arc length of the sector, [degree].
Object > Approximation> Subdivisions **Object > Approximation> Per sector**	Settings on the partition of the facets. Surface of the **cone/cylinder** is approximated by facets, and forming bases are divided into equal arcs, which are under construction sector, and the lateral surface is divided into rectangular or triangular (for non-truncated cone) facets. Number of partitions forming a base defined based on the whole or ellipse, or a sector (user selectable). This parameter determines the number of **triangles number** of facets into which the lateral surface of the **object**. Parameter**on the** sector,the following options: • **No**- given the number of partitions of the circle (arc of 360°) • **Yes**- set the number of partitions sector
Object > Visible **Object > Clipped** **Object > Lighting** **Appearance > ...**	See General properties of Objects.

*) The parameter is calculated automatically and can not be entered by the user.



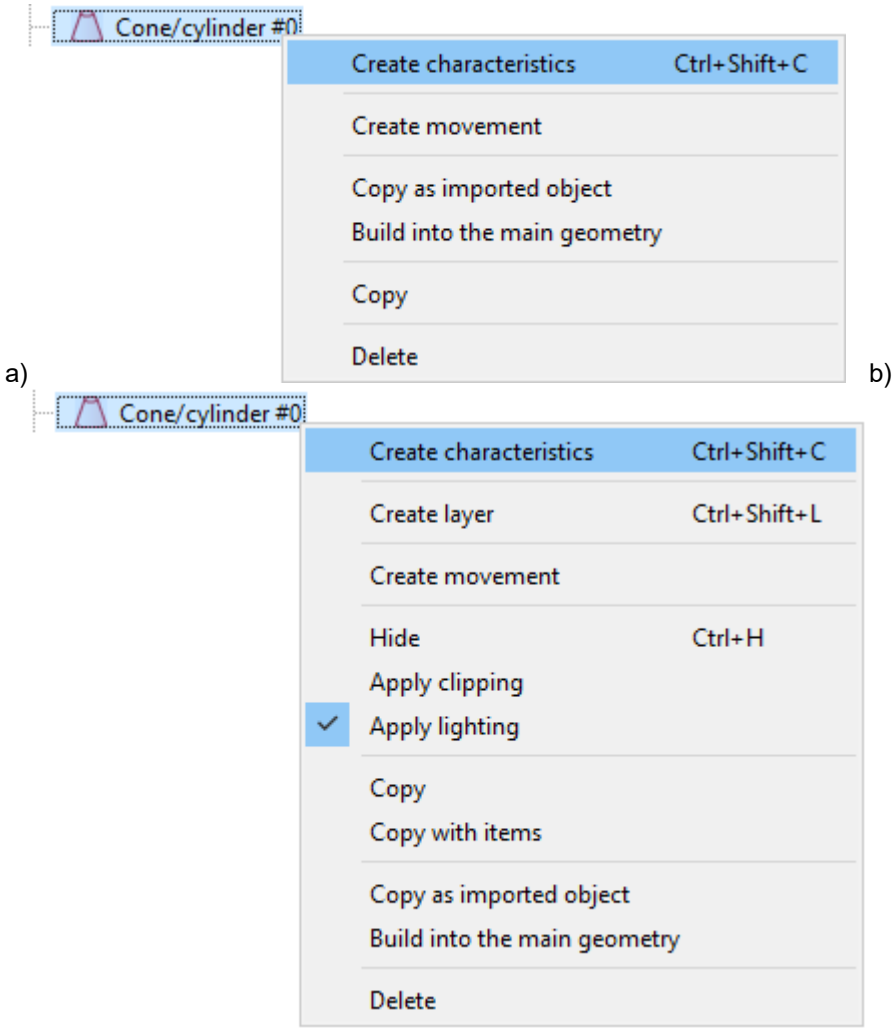
Influence of the parameter **Object > Parameters > Channel** on the shape of the channel inside the **Cone/cylinder**:

1 - **Constant walls**; 2 - **Constant radii**; 3 - **Constant ratio**



When choosing **Object > Parameters > Channel = Constant walls** and some relations between the parameters and the **Base ratio**. One of the openings of the channel may be closed.

Context menu of the "Cone/cylinder" object



Context menu of the **Cone/cylinder** object in the project tree:
a) in the **Preprocessor** tab, b) in the **Postprocessor** tab

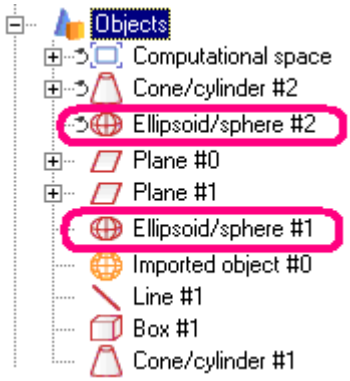
The context menu of the **Cone/cylinder** object in the project tree:

Menu item	Description
Create characteristics	Creating characteristics based on the **Object**. This command is duplicated by the **Ctrl+Shift+C** hot key (by default; you can change this hot key).
Create movement	Creating a movement of the **Object**
Create layer*)	Creates a new **Layer** on/in this **Cone/cylinder**. This command is duplicated by the **Ctrl+Shift+L** hot key (by default; you can change this hot key).
Hide*)	☐ - the **Object** is always displayed in the **View** window ☑ - the **Object** is displayed in the **View** window, only when it is selected in the project tree This command is duplicated by the **Ctrl+H** hot key (by default; you can change this hot key).
Apply clipping*)	☐ - clipping **Planes** do not affect the **Object** ☑ - clipping **Planes** cut the **Object**
Apply lighting*)	☐ - the **Object** is *not* lit by light sources ☑ - the **Object** is lit by light sources
Copy	Creating an element, which is a copy of the selected element

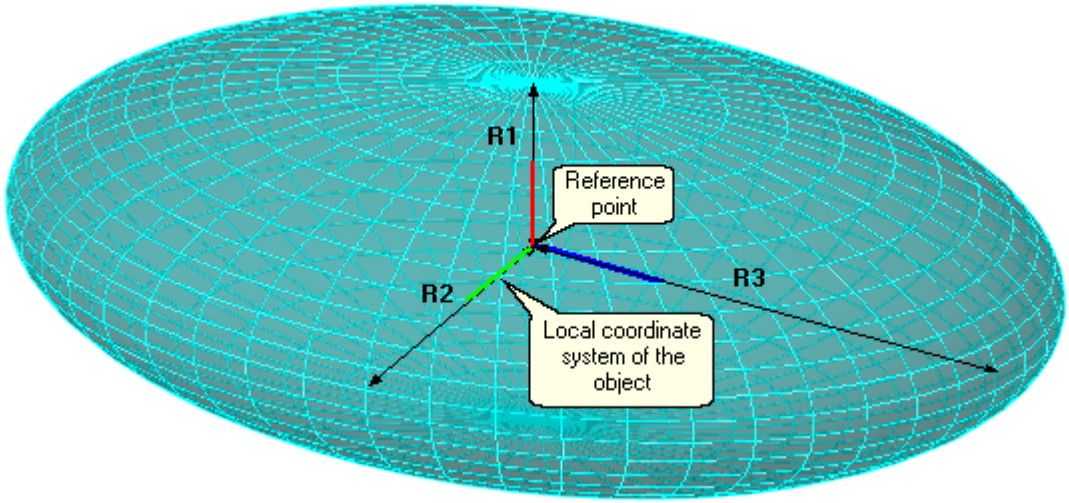
Menu item	Description
Copy with items[*)]	Copying the selected element with its child elements
Copy as imported object	Copying the selected **Object** with converting to an **Imported object**
Built into the main geometry	Add the surface of the selected **Object** into the computational domain
Delete	Deleting the selected element from the project tree

[*)] these commands are only available in the context menu in **Postprocessor**

8.1.8.1.2.5 Object «Ellipsoid/sphere» (user interface)



Objects **Ellipsoid/sphere** in the project tree



Object **Ellipsoid/sphere**

Object **Ellipsoid/sphere** is an ellipsoid defined by three radii. A special case of the ellipsoid is a sphere are (all radii are equal).

Object parameters "Ellipsoid/sphere"

a)

Properties window

Apply | Rollback

Name	Ellipsoid/sphere #2
Location	(Reference point=(X=0; Y=0; Z=0); Axis X=(X=1
Operations	
Reference point	(X=0; Y=0; Z=0)
Axis X	(X=1; Y=0; Z=0)
Axis Y	(X=0; Y=1; Z=0)
Axis Z	(X=0; Y=0; Z=1)
Scale	1
Parameters	(Type=Sphere; Radius=0.005)
Type	Sphere
Radius	0.005
Approximation	(Topology=Polar; Subdivisions=24)
Topology	Polar
Subdivisions	24

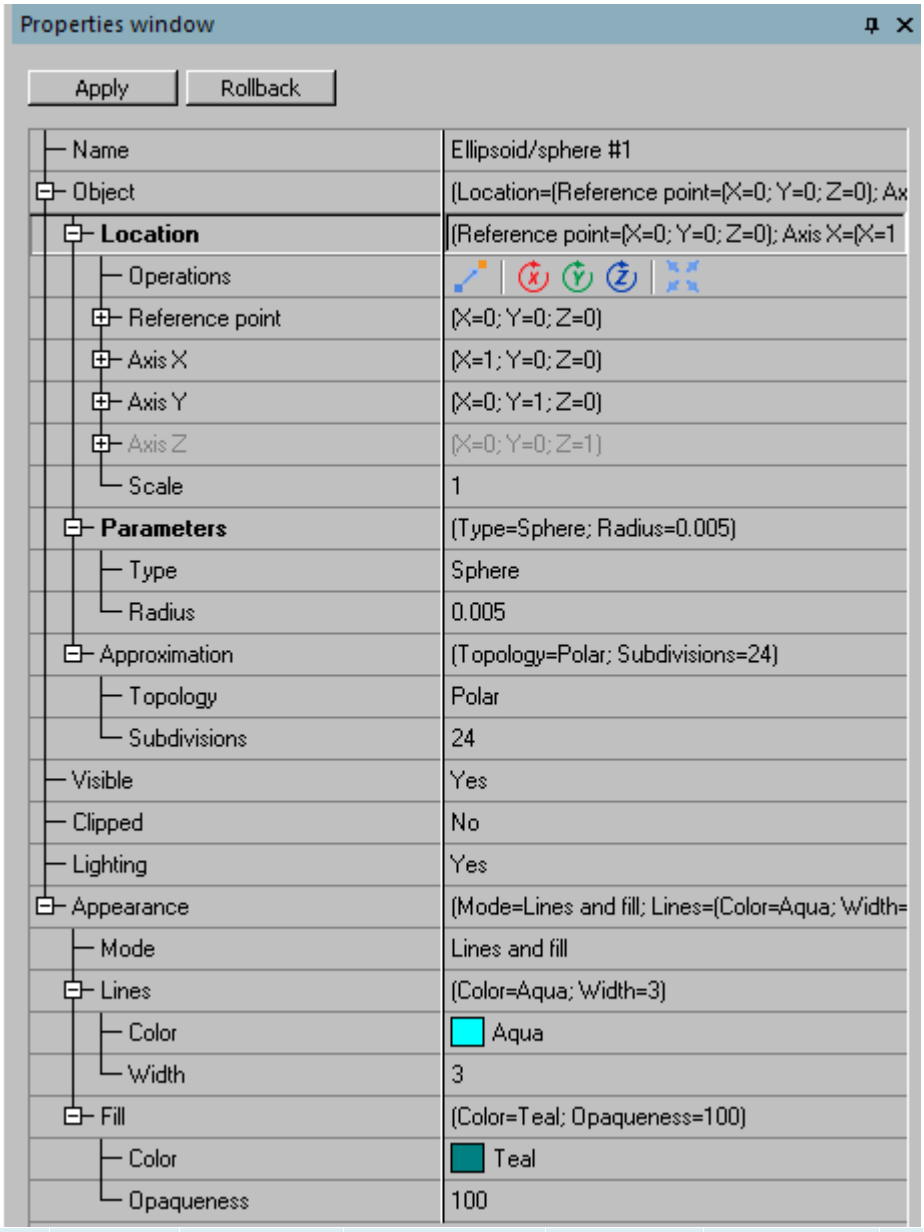
b)

Properties window

Apply | Rollback

Name	Ellipsoid/sphere #2
Location	(Reference point=(X=0; Y=0; Z=0); Axis X=(X=1
Operations	
Reference point	(X=0; Y=0; Z=0)
Axis X	(X=1; Y=0; Z=0)
Axis Y	(X=0; Y=1; Z=0)
Axis Z	(X=0; Y=0; Z=1)
Scale	1
Parameters	(Type=Sphere; Radius=0.005)
Type	Sphere
Radius	0.005
Approximation	(Topology=Geodesic; Base=Icosahedron; Iteratic
Topology	Geodesic
Base	Icosahedron
Iterations	2

The **Properties** window of an **Ellipsoid/sphere** object in the **Preprocessor** tab: a) the polar approximation; b) the geodesic approximation



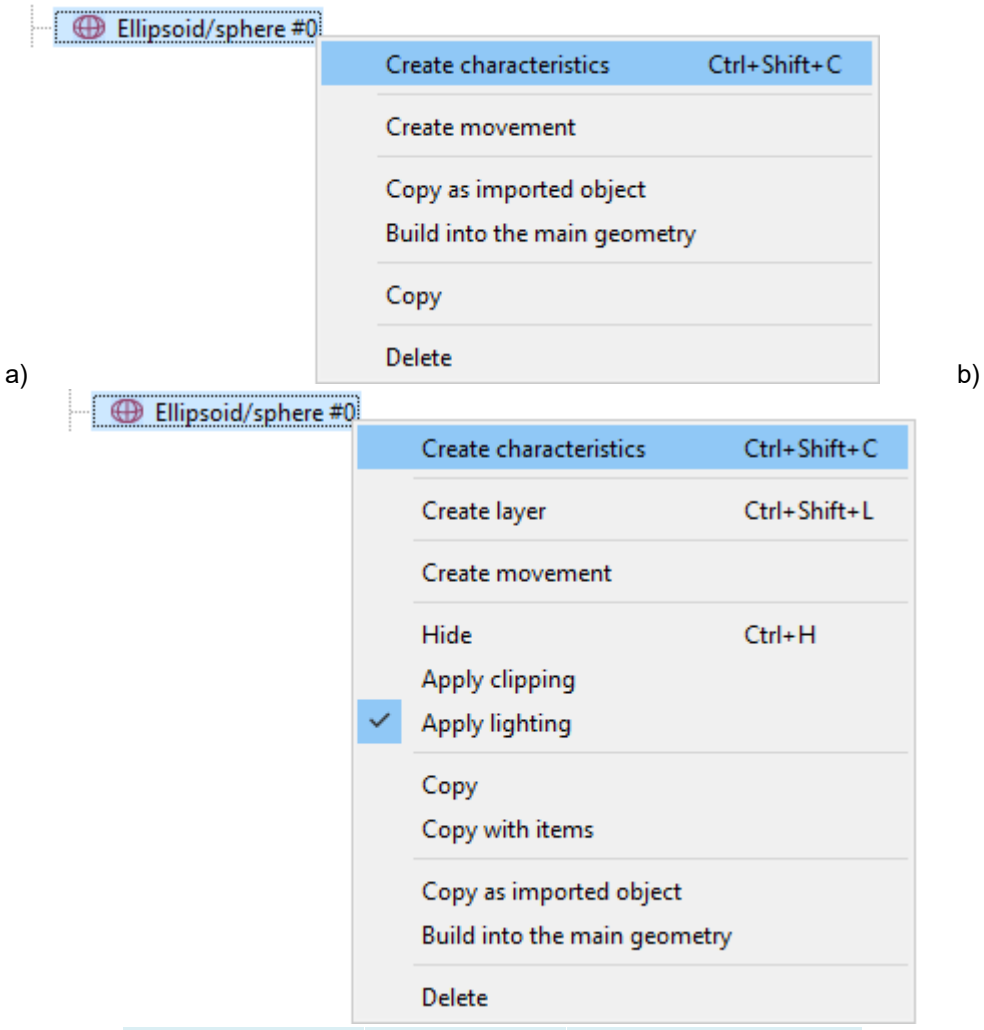
The **Properties** window of an **Ellipsoid/sphere** object in the **Postprocessor** tab

Parameters of object «Ellipsoid/sphere»	
Parameter	**Description**
Name	Name of the **Object** in the project tree. The name is generated automatically as **Ellipsoid/sphere #N** and you can change it if you wish.
Object > Location > Operations	See General parameters of objects.
Object > Location > Reference point > X	
Object > Location > Reference point > Y	
Object > Location> Reference point > Z	
Object > Location > Axis X > ...	

Parameters of object «Ellipsoid/sphere»	
Parameter	**Description**
Object > Location > Axis Y > ...	
Object > Location> Axis Z > ...	
Object > Location > Scale	
Object > Parameters > Type	Determines whether the object is a sphere or ellipsoid. Possible options are: • **Sphere** - the object in the form of a sphere (defined by a single radius) • **Ellipsoid**- an object in the shape of an ellipsoid (defined three radii)
Object > Parameters > Radius *(for a sphere)*	Radius of the sphere (before application of the scaling factor), [m]
Object > Parameters > Radius 1 *(for an ellipsoid)*	The radii of the ellipsoid - the value of R_1, R_2, R_3 (before the application of the scaling factor), [m]
Object > Parameters > Radius 2 *(for an ellipsoid)*	
Object > Parameters > Radius 3 *(for an ellipsoid)*	
Object > Approximation> Topology	Determines the way of approximation of the **Ellipsoid/sphere** by a polyhedron. Possible options are: • **Polar** -on the surface of the object is constructed the same number of parallels and meridians: generators in the XY and XZ are divided into equal arcs, which are built quadrangular and near the poles, triangularfacets (variant "a" in the figure) Size facets depends on the parameter **Subdivisions**,with the growth of facets which are reduced. • **Geodesic** -the surface of the object are divided into triangular facets, using as an initial approximation stretched Platonic body with triangular faces (tetrahedron, octahedron, icosahedron).
Object > Approximation > Subdivisions *(for the polar approximation)*	This parameter sets the size and number of facets for polar approximation of the **Ellipsoid/sphere**. Size facets decreases when the **Subdivisions** parameter increases. This parameter is an integer number greater or equal **4**. Let *n*=**Subdivisions**, then: • Number of meridional sectors of the approximation will be *n* • Number of parallel slices (which are rings and polar areas) of the approximation will be: ○ *n*/2 for even *n* ○ (*n*+1)/2 for odd *n*
Object > Approximation > Base *(for the geodesic approximation)*	This parameter specifies the basis for building a geodesic approximation **Ellipsoid/sphere**. Possible options are: • **Tetrahedron** • **Octahedron** • **Icosahedron**
Object > Approximation > Iterations *(for the geodesic approximation)*	This parameter will determine the number of iterations applied in the construction of the geodesic approximation **Ellipsoid/sphere**. Possible values are: an integer from 0 to 6 inclusive. Size facets decreases rapidly with increasing the **Subdivisions** parameter. When you set **Iterations = 0** to be a tetrahedron, octahedron, icosahedron, or in accordance with the value of the parameter **Approximation > Base** (see an illustration in the section Object «Ellipsoid/sphere»).

Parameters of object «Ellipsoid/sphere»	
Parameter	**Description**
Visible	See General parameters of objects.
Clipped	
Lighting	
Appearance	

Context menu of the "Ellipsoid/sphere" object



Context menu of the **Ellipsoid/sphere** object in the project tree:
a) in the **Preprocessor** tab, b) in the **Postprocessor** tab

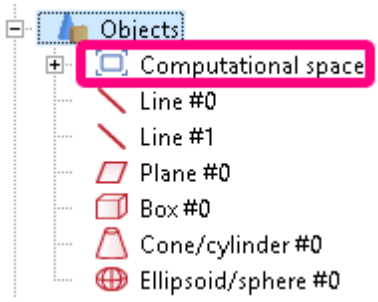
The context menu of the **Ellipsoid/sphere** object in the project tree:

Menu item	Description
Create characteristics	Creating characteristics based on the **Object**. This command is duplicated by the **Ctrl+Shift+C** hot key (by default; you can change this hot key).
Create movement	Creating a movement of the **Object**
Create layer[*)]	Creating a new **Layer** on/in this **Ellipsoid/sphere**. This command is duplicated by the **Ctrl+Shift+L** hot key (by default; you can change this hot key).
Hide[*)]	- the **Object** is always displayed in the **View** window

Menu item	Description
	☑ - the **Object** is displayed in the **View** window, only when it is selected in the project tree This command is duplicated by the **Ctrl+H** hot key (by default; you can change this hot key).
Apply clipping[*)]	☐ - clipping **Planes** do not affect the **Object** ☑ - clipping **Planes** cut the **Object**
Apply lighting[*)]	☐ - the **Object** is *not* lit by light sources ☑ - the **Object** is lit by light sources
Copy	Creating an element, which is a copy of the selected element
Copy with items[*)]	Copying the selected element with its child elements
Copy as imported object	Copying the selected **Object** with converting to an **Imported object**
Built into the main geometry	Add the surface of a selected object in the computational domain
Delete	Deleting the selected element from the project tree

[*)] these commands are only available in the context menu in **Postprocessor**

8.1.8.1.3 Object «Computational space»



Computational space is a special **Object** that is created along with the creation of the project and it cannot be deleted nor edited because it does not have any properties. This object does not impose spatial restrictions on the rendering.

See also:

- Element «Computational space» in the Preprocessor tab
- Element «Computational space» in the Postprocessor tab

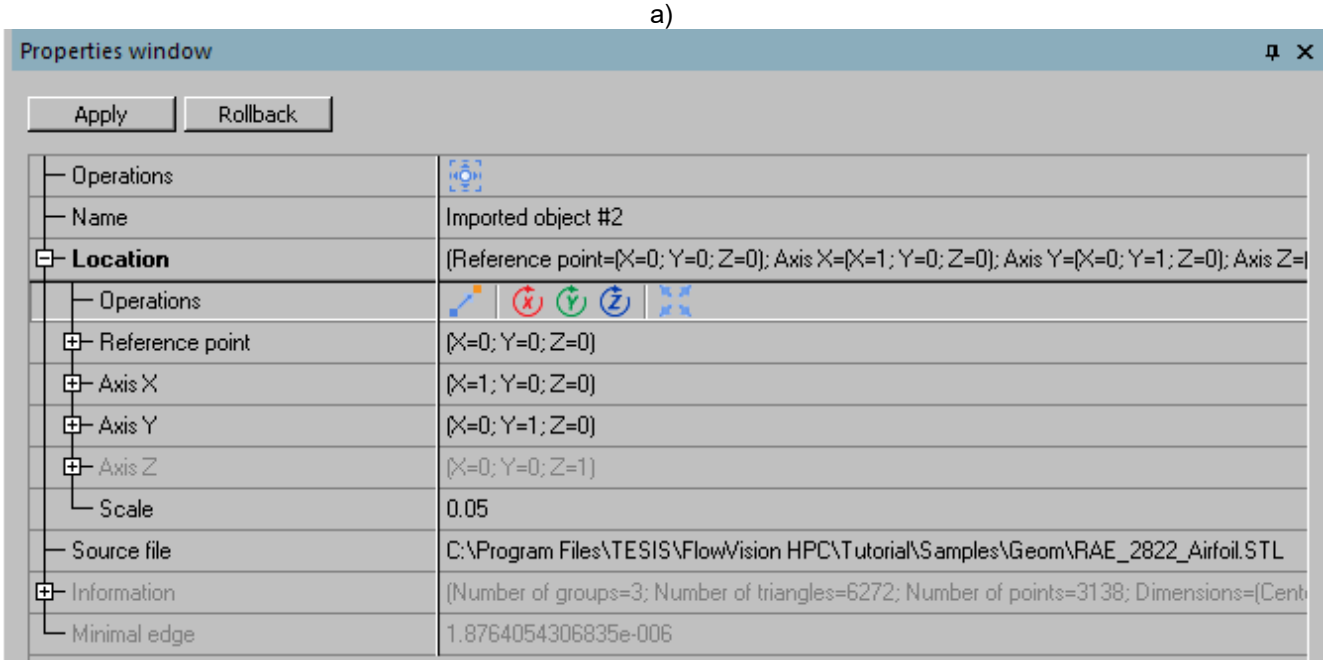
8.1.8.1.4 Imported objects (user interface)

Imported object is an arbitrary geometric object that is defined by a set of triangles. **Imported object** can be created on the basis of any other entity having a finite area (using the **Copy as imported object** command from the context menu), or loaded from a stand-alone file with geometry.

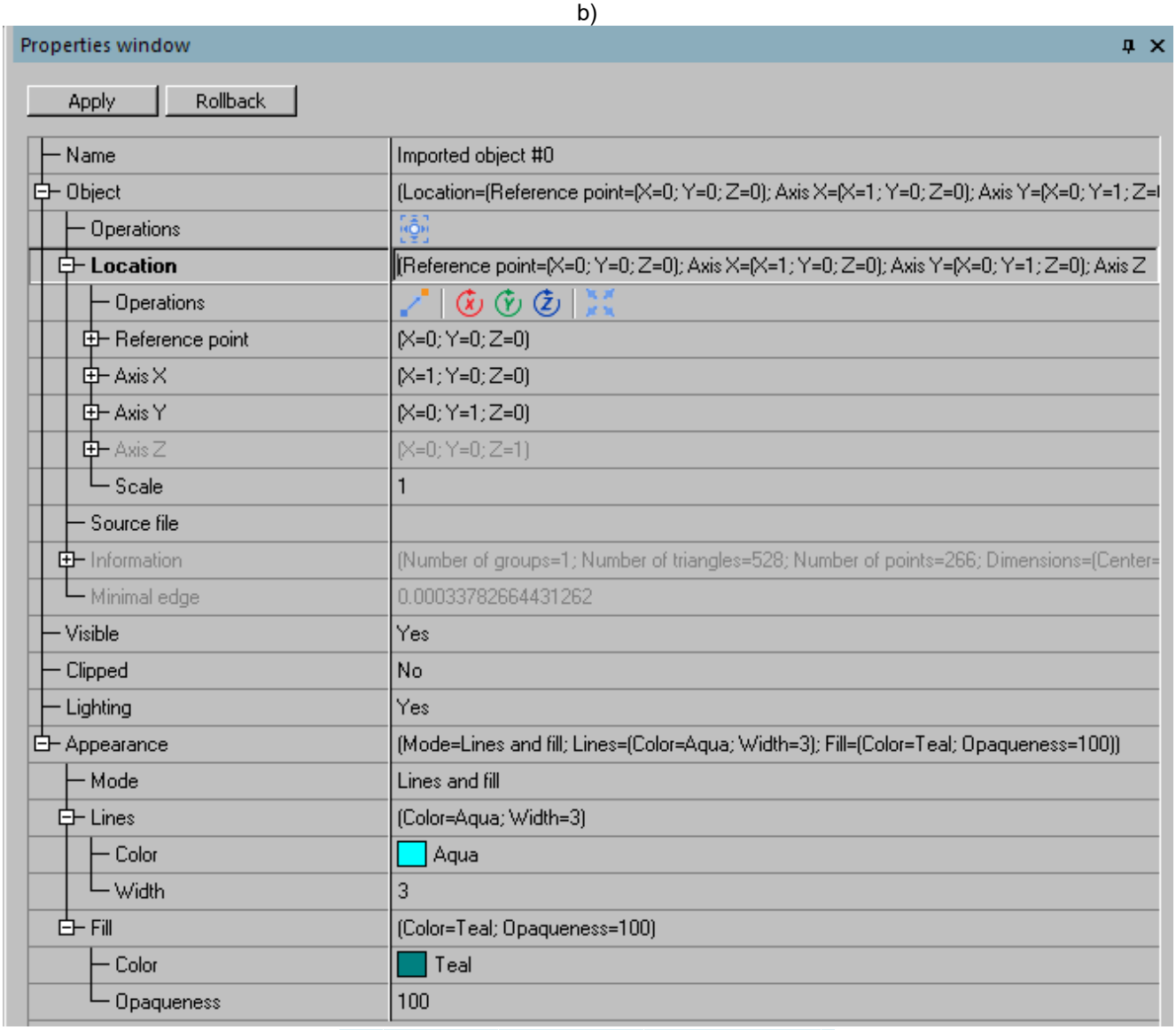
A special case of **Imported objects** are objects **Self-intersections #N**, which are generated at checking the geometry for self-intersections.
Objects **Self-intersections #N** are presented in the **Postprocessor** tab only.
See section Self-intersections of surfaces and their correction.

The Properties window of an Imported object

a)



b)

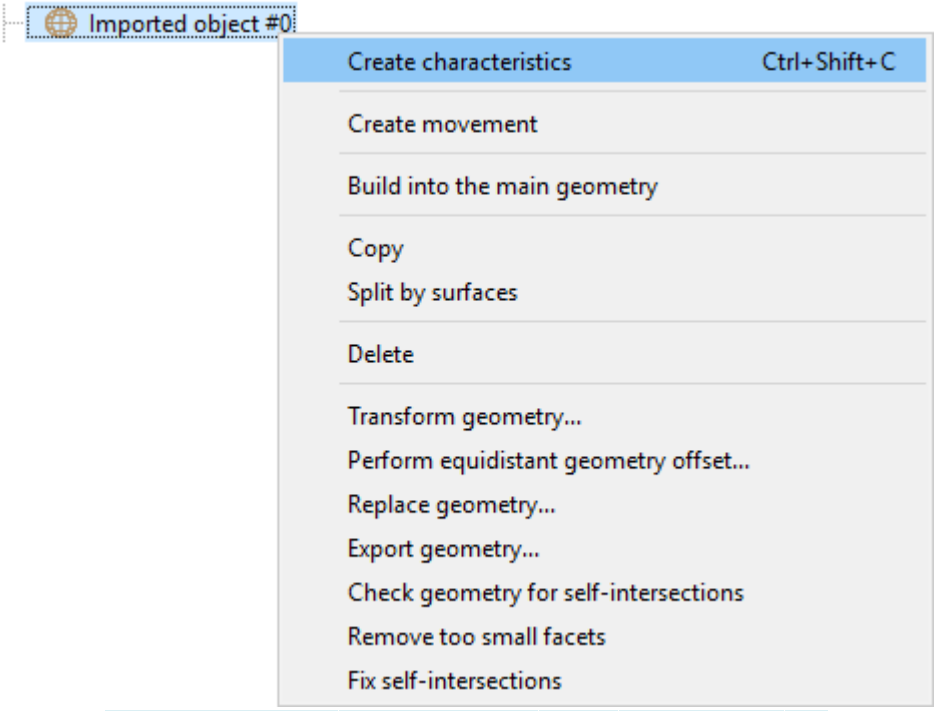


The **Properties** window of an **Imported object**:
a) in the **Preprocessor** tab; b) in the **Postprocessor** tab

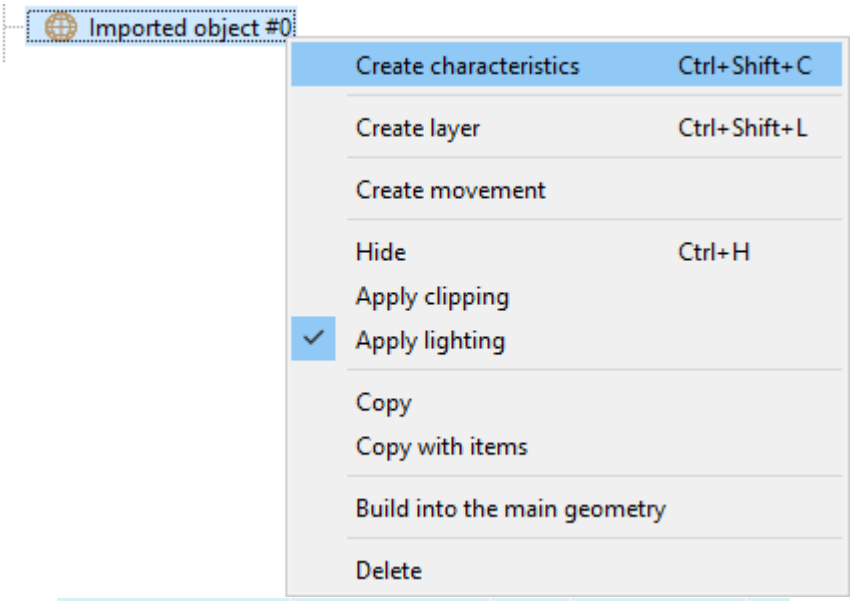
Parameters of object **Imported object**:

Parameter, button	Description
Operations > 	The object is placed in the center of the computational domain and scaled so that the **Object** was inscribed in the computational domain.
Name	Name of the **Object** in the project tree. This name is generated automatically and you can change it.
Location	See General properties of Objects.
Source file	The path to the file from which the **Object** has been imported
Information > Number of groups	The number of groups of facets that make up the surface of the **Object**. This field is read-only and is not editable.
Information > Number of triangles	The number of facets that make up the surface of the **Object**. This field is read-only and is not editable.
Information > Number of points	The number of nodes facets constituting the surface of the **Object**. This field is read-only and is not editable.
Information > Dimensions > Center > X	Coordinates of the origin of the local coordinate system object ACS. This field is read-only and is not editable.
Information > Dimensions > Center > Y	
Information > Dimensions > Center > Z	
Information > Dimensions > Size > X	Size of the object along the axes of ACS. This field is read-only and is not editable.
Information > Dimensions > Size > Y	
Information > Dimensions > Size > Z	
Information > Dimensions > Min > X	The minimum values of the coordinates of the object surface in the ACS. This field is read-only and is not editable.
Information > Dimensions > Min > Y	
Information > Dimensions > Min > Z	
Information > Dimensions > Max> X	The maximum values of the coordinates of the object surface in the ACS. This field is read-only and is not editable.
Information > Dimensions > Max> Y	
Information > Dimensions > Max> Z	
Information > Minimal edge	The minimum edge of the surface's facets. This field is read-only and is not editable.
Visible	See General properties of Objects.
Clipped	
Lighting	
Appearance	

Context menu of Imported object



Context menu of an **Imported object** in the **Preprocessor** tab



Context menu of an **Imported object** in the **Postprocessor** tab

The context menu of the **Imported object** object in the project tree:

Command	Description
Create characteristics	Creating **Characteristics** on this **Imported object**. This command is duplicated by the **Ctrl+Shift+C** hot key (by default; you can change this hot key).
Create movement	Creates **Movement** of this **Imported object**[1,2]. The operation is available if no **Moving body** modifier is specified on the **Imported object**. If a **Movement** is specified, then it is impossible to set a **Moving body** on this **Imported object**.
Create layer[*]	Creating a new **Layer** on this **Imported object**.

Command	Description
	This command is duplicated by the **Ctrl+Shift+L** hot key (by default; you can change this hot key).
Hide[*)]	☐ - the **Object** is always displayed in the **View** window ☑ - the **Object** is displayed in the **View** window, only when it is selected in the project tree This command is duplicated by the **Ctrl+H** hot key (by default; you can change this hot key).
Apply clipping[*)]	☐ - clipping **Planes** do not affect the **Object** ☑ - clipping **Planes** cut the **Object**
Apply lighting[*)]	☐ - the **Object** is *not* lit by light sources ☑ - the **Object** is lit by light sources
Copy	Creating an item, which is a copy of the selected item
Copy with items[*)]	Copying the selected element with its child elements
Split by surfaces	Split the **Imported object** that consists of several non-connecting to each other surfaces to individual **Imported Objects** (this command is available when no **Moving body** is created on the **Imported object**).
Build into the main geometry	Add the surface of a selected object in the computational domain
Delete[3)]	To delete an object from the tree project
Transform geometry	Transform the geometry of the **Imported object**. These operations are performed in the **Geometry transformation** dialog box.
Perform equidistant geometry offset	Make equidistant shift geometry of **Imported object** (available, if no **Moving body** is defined on the **Imported object**). These actions are made in the dialog box **Equidistant offset of surfaces**.
Replace geometry	Replace geometry of the**Imported** object. The **Open** dialog box to import the object, replacing the selected **Imported object**.
Export geometry	Exporting geometry of the **Imported object**. The **Open** dialog box will open where you can export the surface of the **Imported object** into a file.
Check geometry for self-intersections[*)]	Checking for self-intersection of geometry of the **Imported object**
Remove too small facets[**)]	Removing too small facets on the surface of the object
Fix self-intersections	Correct geometry of the self-intersection of **Imported object** - removing self-intersecting facets on the surface of the object.

[*)] These commands are only available in the context menu in **Postprocessor**.
[**)] This command is only available in the context menu in **Preprocessor**.
[1)] Motion of a **Preprocessor**'s **Object** can be defined only in **Preprocessor**.
[2)] **Movement** can be set on a **Box**, **Cone/cylinder**, **Ellipsoid/sphere**, and **Imported object** (if no **Moving body** is created based on it).
[3)] Deleting an **Object**, which has been created in **Preprocessor**, is possible in **Preprocessor** only.

8.1.8.1.5 Special objects (user interface)

The special objects include:

- object **Supergroup**
- object **Set of sensors**
- element **Movement**

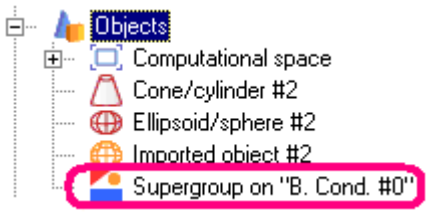
8.1.8.1.5.1 Object «Supergroup» (user interface)

A *Supergroup* is a geometric object consisting of a set of geometric groups of facets.

A **Supergroup** is created initially:

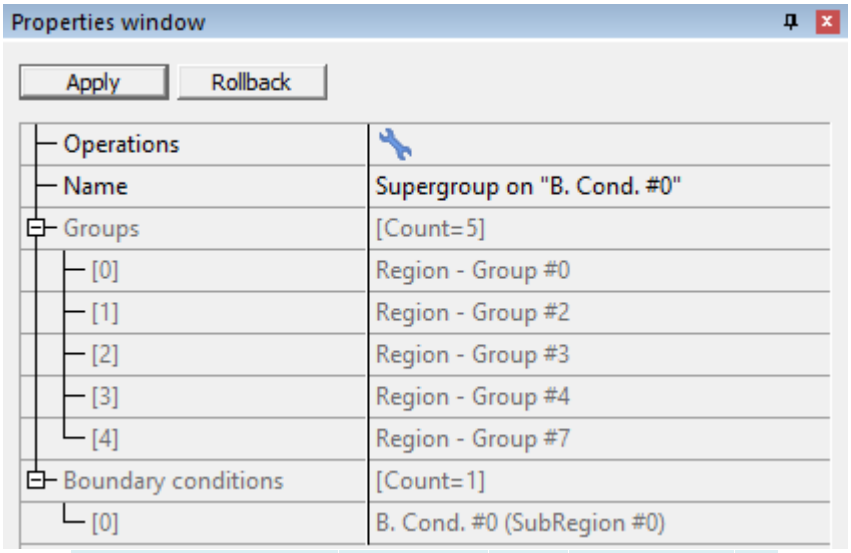
- using the **Create supergroup > In Preprocessor** or **Create supergroup > In Postprocessor** command from the context menu of some **Boundary condition** (in this case the program creates a **Supergroup** that has all facet groups, on which the **Boundary condition** is set)
- using the **Create** command from the context menu of the **Objects** folder and selecting **Object type = Supergroup** in the **Create new object** window (in this case an empty **Supergroup** is created)

The user can edit a **Supergroup** so it will contain groups of facets, on which different **Boundary conditions** are set and/or belonging to different **Imported objects**.

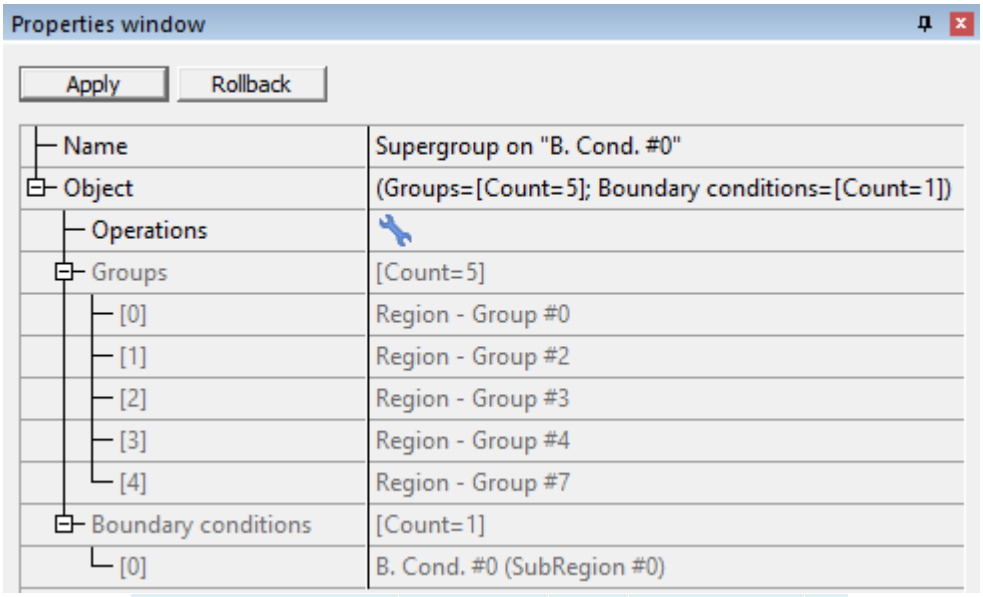


Object **Supergroup** in the project tree

Parameters of object "Supergroup"



Properties window of a **Supergroup** in the **Preprocessor** tab

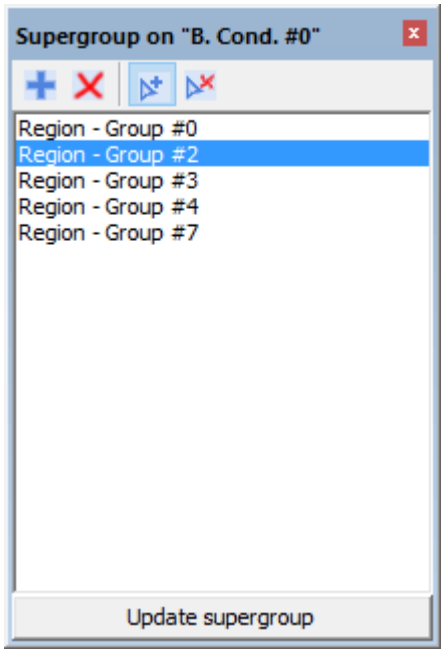


Properties window of a **Supergroup** in the **Postprocessor** tab

Parameters of object **Supergroup**:

Parameter	Description
Name	Name of the **Supergroup** in the project tree
Operations *or* **Object > Operations**	Line of this parameter contains the button (**Change list of groups**). Clicking this button opens the window for editing the **Supergroup**, see subsection "*The Supergroup editing window*" below.
Groups > [N] *or* **Object > Group> [N]**	Group of facets, on which the **Supergroup** is built. Information field, not editable.
Boundary conditions > [N] *or* **Object > Boundary conditions > [N]**	List of boundary conditions, which are set on facet groups that are included in the **Supergroup**. Information field, not editable.

The Supergroup editing window



The **Supergroup** editing window opens when you click on the (**Change list of groups**) icon in the line of the **Operations** (or **Object > Operations**) parameter in the Properties window of the **Supergroup**.

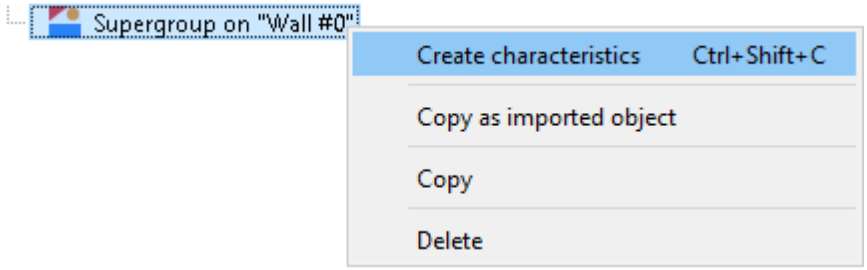
This window works jointly with the **View** window and contains the following interface elements:

Interface element	Description
(**Add selected group to the list**)	This button is active if the facet group, which is selected in the **View** window (by the left mouse button when the **Ctrl** key is pressed) and/or in the project tree (in the element **Subregions > SubRegion #N > Geometry > Region - Surface #N** when the **Pre-Postprocessor**'s setting **Display > Show all groups** is enabled), is absent in the list. Clicking this button adds the selected group into the list.
(**Remove selected group from the list**)	This button is active if some facet group is selected in the list. Clicking this button removes the selected group from the list. Pressing the **Del** key has the same effect.
(**Automatically add groups to the list when they are selected in the graphics window**)	If this button is pressed, then, at each selecting a facet group in the **View** window (but not in the project tree), this facet group will be added to the list. Pressing this button releases the button .
(**Automatically remove groups from the list when they are selected in the graphics window**)	If this button is pressed, then, at each selecting a facet group in the **View** window (but not in the project tree), this facet group will be removed from the list. Pressing this button releases the button .
List of the facet groups	This list contains facet group that are included in the **Supergroup**. When a group is selected in the **View** window or in the project tree, this group will be also automatically selected in the list (if it is included in the list).
Update supergroup *(screen button)*	Clicking this button closes the window and your changes become actual.
(an icon located in the top right corner of the window)	Clicking this icon closes the window and discard your changes.

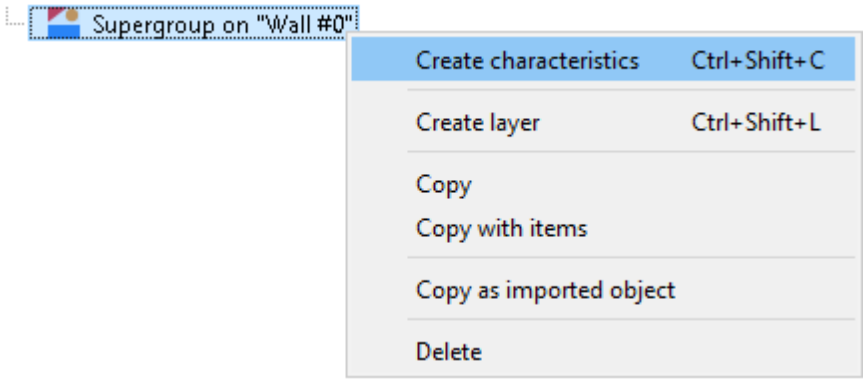
If you select another **Supergroup** in the project tree and click the button in its properties, then the current **Supergroup** editing window will be replaced by another one without saving the entered data.

If a facet group that is selected in the **View** window separates two **Subregions**, then selection a side of the group (and an appropriate element in the project tree) depends on which side was selected first in the the **View** window.

Context menu of a Supergroup



The context menu of the **Supergroup** object in the **Preprocessor** tab



The context menu of a **Supergroup** in the **Postprocessor** tab

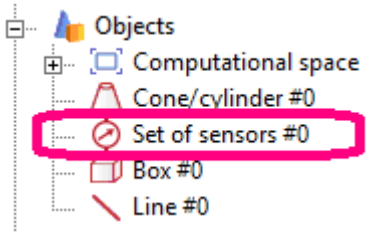
Context menu item **Supergroup** in the project tree:

Menu command	Description
Create characteristics	Creating characteristics based on the **Object**. This command is duplicated by the **Ctrl+Shift+C** hot key (by default; you can change this hot key).
Create layer*)	Creates a new **Layer** on this **Supergroup**. This command is duplicated by the **Ctrl+Shift+L** hot key (by default; you can change this hot key).
Copy	Creating an element, which is a copy of the selected element
Copy with items*)	Copying the selected element with its child elements
Copy as imported object	Copying the selected **Object** with converting to an **Imported object**
Delete	Deleting the selected element from the project tree

*) these commands are only available in the context menu in **Postprocessor**

8.1.8.1.5.2 Object «Set of sensors» (user interface)

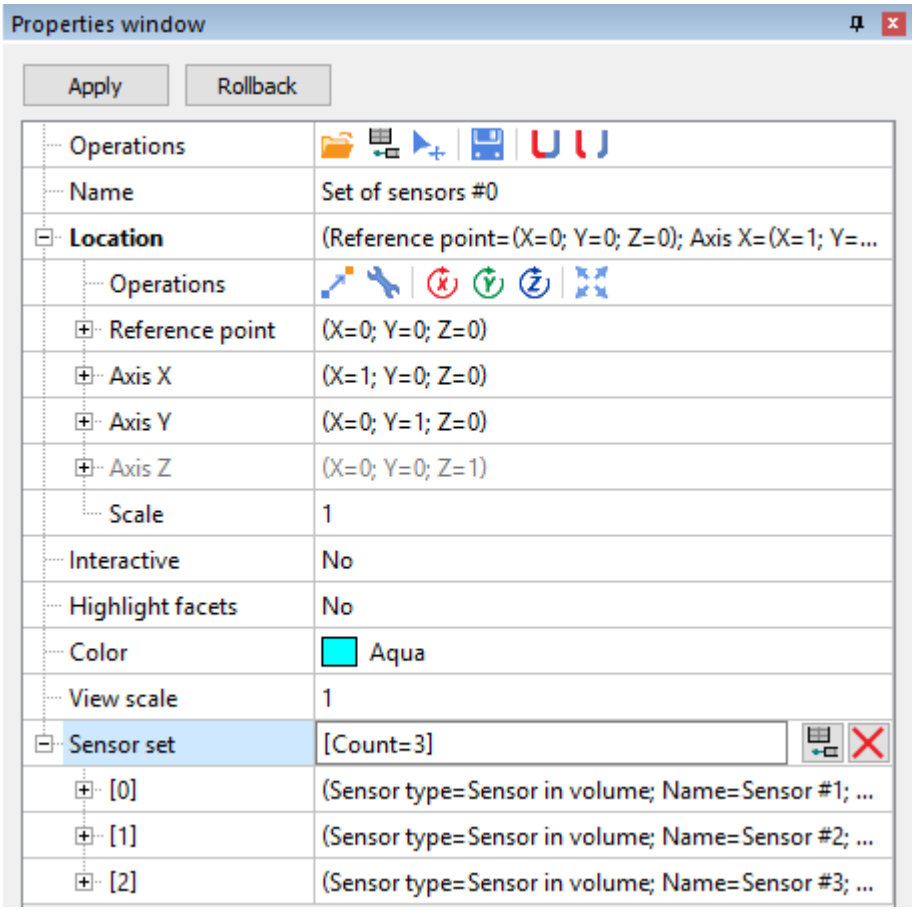
Set of sensors is a geometric object consisting of several mathematical points.



Sets of sensors are useful for calculating **Characteristics** in the points that are placed in desired locations.

Also you can use **Sets of sensors** to build a [Cell set] layer on it.

Parameters and icons in the Properties window of the "Set of sensors" object



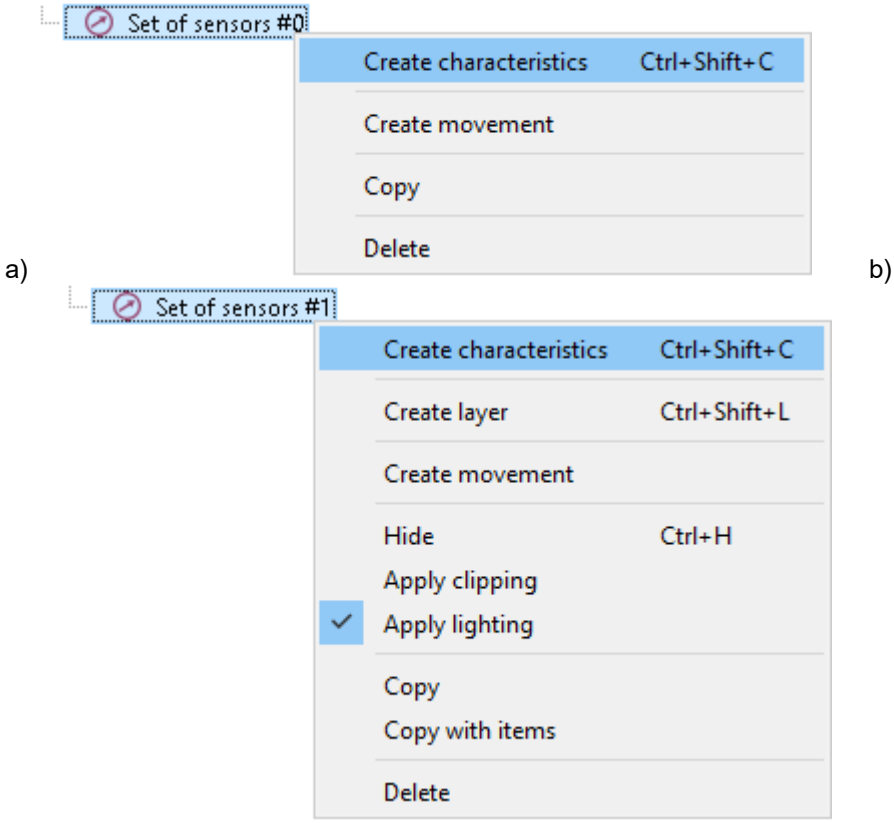
Parameters and icons in the Properties window of the "Set of sensors" object	
Parameter	**Description**
Operations*)	This field contain icons that you can use for actions with the **Set of sensors**: • **Overwrite sensors from file** replaces all existing **Sensors** with **Sensors** that were previously saved in a file • **Add sensors from file** takes **Sensors** that were previously saved in a file and adds them to existing **Sensors**. • **Add sensor(s) with cursor** - add **Sensors** using your computer mouse. See subsection *"Adding sensors on a surface clicking in the View window with mouse clicks"* below. • **Export sensors to file** saves existing **Sensors** into a text file. The file will have the `txt` file name extension. • **Bind sensor(s)** sticks all sensors of the **Set of sensors** on the surface of the geometry model, a **Moving body**, or a **Boundary condition**. The **Bind sensor(s)** dialog box will open, which contains the same fields as the **Add**

Parameters and icons in the Properties window of the "Set of sensors" object	
Parameter	**Description**
	sensor(s) with cursor dialog box and also the **Binding radius** field, where you specify the maximal radius for searching surfaces, on which the **Sensors** can be bound (stuck). • **Reset binding** disconnects (releases) all sensors of the **Set of sensors** from the surface(s), on which they were stuck. Binding of **Sensors**, which are loaded from a file, is defined by settings that are specified for the whole **Set**. See also subsection *Format of files for saving Sensors* below.
Name*)	Name of the **Object** in the project tree. The name is generated automatically as **Set of sensors #N** and you can change it if you wish.
Location > Operations*)	See General parameters of objects.
Location > Reference point > X*)	Similarly to other geometric **Objects**, a **Set of sensors** has its own its own local coordinate system (LCS). When the LCS of the **Set** is shifted, rotated, or scaled, all *volume* **Sensors** in the **Set** will move, but *surface* **Sensors** will *not* move. Operations and parameters in the group **Location > ...** in properties of the **Set of sensors** will not be applied to *surface* **Sensors**. *Volume* **Sensors** *don't* change their locations at geometry transformations because their coordinates are bound to the LCS of the **Set**. *Surface* **Sensors** *do change* their locations at geometry transformations and their coordinates in the LCS of the **Set** are changed. Surface sensors move together with the surface(s), to which they are bound, for example, together with a surface of a **Moving body**.
Location > Reference point > Y*)	
Location > Reference point > Z*)	
Location > Axis X > ...*)	
Location > Axis Y > ...*)	
Location > Axis Z > ...*)	
Location > Scale*)	
Interactive	This parameter specifies if position of the **Set of sensors** can be changed interactively during editing the object. Possible values: **Yes | No**.
Highlight facets	Highlighting the facets, to which the surface sensors from this set are bound (see an illustration). Possible values: **Yes | No**.
Color	The color, which is used to highlight the **Set of sensors** in the **View** window.
View scale*)	Size of images of **Sensors** in the **View** window (see an illustration). This parameter is set in the range (**0** ... **2.1**) excluding limits of the range; the default value is **1**.
Sensor set > ...*)	An array of groups of parameters that contain properties of separate **Sensors**. When a **Sensor** is created by clicking the **Append item to the array (Ins)** icon, its location will automatically continue the sequence in direction from the two previous **Sensors** (see an illustration).
Sensor set > [N]	Group of parameters corresponding to one **Sensor** (**N = 0**, **1**, **2**, ...). When a **Sensor** is created by clicking the **Insert item before current (Ins)** icon, its location will be automatically set between the current and previous **Sensors**.
Sensor set > [N] > Operations	Icons for operations with this **Sensor**: • **Bind sensor(s)** • **Reset binding** These icons work for the selected **Sensor** as the similar icons for the whole **Set of sensors**.
Sensor set > [N] > Sensor type	Type of this **Sensor** depending on binding to a surface (for example, because the the **Bind sensor(s)** icon has been applied for this **Sensor**).

Parameters and icons in the Properties window of the "Set of sensors" object	
Parameter	**Description**
	Possible values: **Sensor in volume | Sensor on surface**. This field is informational only and it is filled by the program automatically.
Sensor set > [N] > Name	Name of the **Sensor**. By default, **Name=Sensor #N** and you can change it if you wish.
Sensor set > [N] > Shift	Shift of the point, from which data for the **Sensor** will be received, relating to the surface, to which the **Sensor** is bound, in direction along the normal to the surface. This parameter is available only for **Sensors** that are bound to a surface (**Sensor type = Sensor on surface**).
Sensor set > [N] > Global coordinates > X	Coordinates of the **Sensor** in the absolute coordinate systems (ACS). These parameters are available only for **Sensors** that are bound to a surface (**Sensor type = Sensor on surface**).
Sensor set > [N] > Global coordinates > Y	
Sensor set > [N] > Global coordinates > Z	
Sensor set > [N] > Coordinates > X	Coordinates of the **Sensor** in the local coordinate system of the **Set of sensors**. These parameters are available only for **Sensors** that are *not* bound to a surface (**Sensor type = Sensor in volume**).
Sensor set > [N] > Coordinates > Y	
Sensor set > [N] > Coordinates > Z	
Visible	See General parameters of objects.
Clipped	

*) in **Postprocessor** these parameters locate in the group of parameters **Object > ...**

Context menu of the "Set of sensors" object



Context menu of the **Set of sensors** element in the project tree:
a) in the **Preprocessor** tab; b) in the **Postprocessor** tab

The context menu of the **Set of sensors** object in the project tree

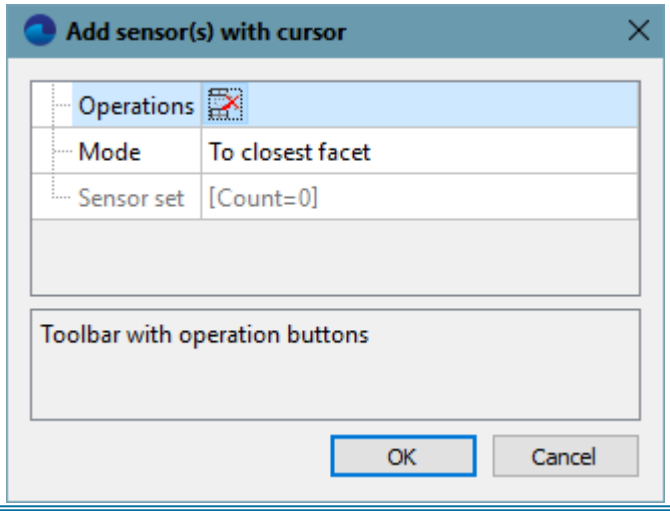
Menu item	Description
Create characteristics	Creating **Characteristics** on the **Object**. This command is duplicated by the **Ctrl+Shift+C** hot key (by default; you can change this hot key).
Create movement	Creating **Movement** of the **Object**
Create layer[*)]	Creating a new **Cell set** **Layer** on the cells that correspond to this **Set of sensors**. This command is duplicated by the **Ctrl+Shift+L** hot key (by default; you can change this hot key).
Hide[*)]	☐ - the **Object** is always displayed in the **View** window ☑ - the **Object** is displayed in the **View** window, only when it is selected in the project tree This command is duplicated by the **Ctrl+H** hot key (by default; you can change this hot key).
Apply clipping[*)]	☐ - clipping **Planes** do not affect the **Object** ☑ - clipping **Planes** cut the **Object**
Apply lighting[*)]	☐ - the **Object** is *not* lit by light sources ☑ - the **Object** is lit by light sources
Copy	Creating an element, which is a copy of the selected element
Copy with items[*)]	Copying the selected **Object** with its child elements
Delete	Deleting the selected element from the project tree

[*)] these commands are only available in the **Postprocessor** tab

Adding sensors on a surface clicking in the View window with mouse clicks

The program implements a convenient user interface that allows you to place sensors on surfaces using mouse clicks in the **View** window.

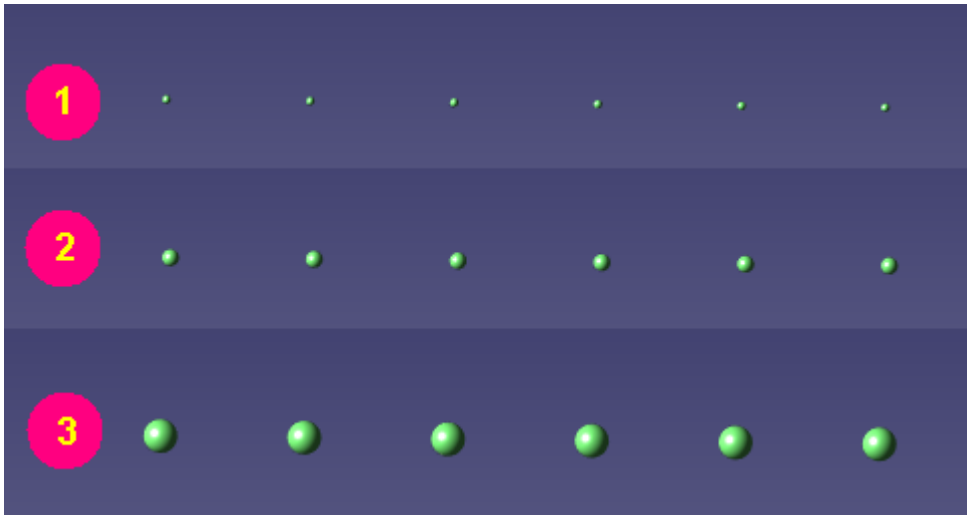
Follow the steps:

Step	Actions
1	In the **Properties** window of the **Set of sensors** click the **Operations** > **Add sensor(s) with cursor** button. The **Add sensor(s) with cursor** dialog box will open: 
2	If necessary, specify parameters of the algorithm of adding sensors on a surface: • **Mode** defines the surface, on which **Sensors** would be placed. Possible options are: **To closest facet | To boundary condition**. • **Subregion**, that contain the **Boundary condition**, on which **Sensors** would be placed. The value is selected from a list. This parameter is available when **Mode = To boundary condition**. • **Choose BC** specifies the **Boundary condition**, on which **Sensors** would be placed. The value is selected from a list. This parameter is available when **Mode = To boundary condition**. The **Operations >** button removes the last added **Sensor**.
3	Keeping the **Add sensor(s) with cursor** dialog box opened, place the mouse pointer into to the **View** window and, clicking the left mouse button, add new **Sensors** that will be placed on a surface according to the specified parameters.
4	In the **Add sensor(s) with cursor** dialog box click **OK**.
5	In the **Properties** window of the **Set of sensors** click **Apply**.

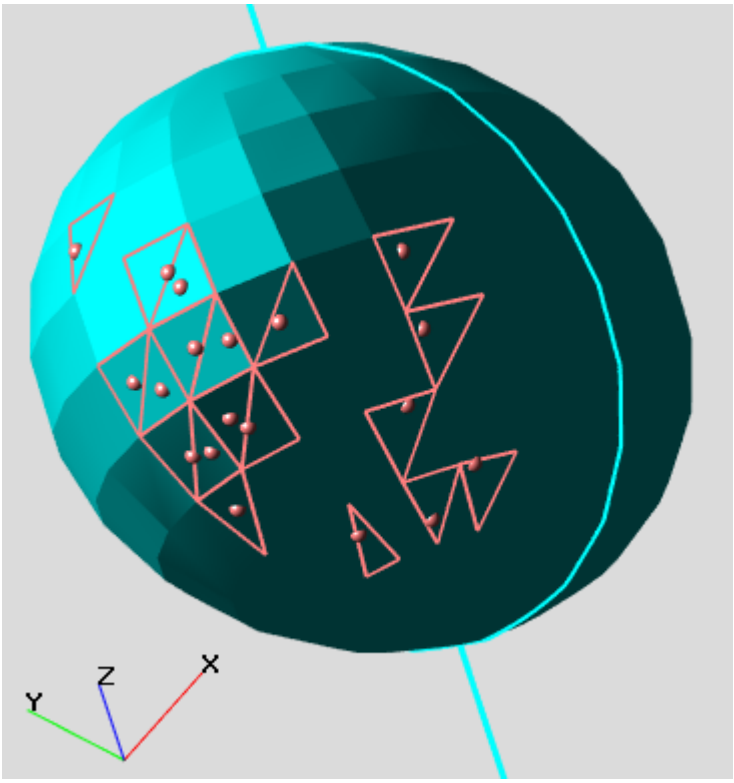
A **Sensor**, which stuck to the surface of a **Moving body**, will move along with this **Moving body** during the computation.

If a surface, to which some **Sensors** are bound, is deleted, binding of these **Sensors** is defined by settings that are specified for the whole **Set**.

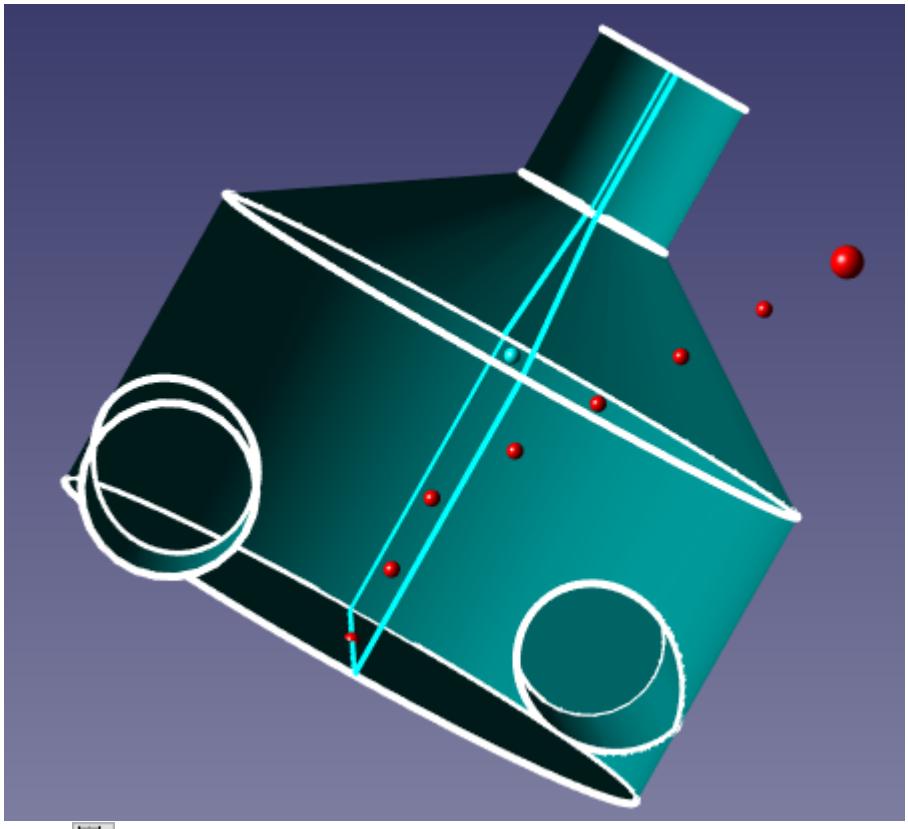
Illustrations



The **View scale** parameter changes sizes of **Sensors'** images in the **View** window (1 - **View scale = 0.5**; 2 - **View scale = 1**; 3 - **View scale = 2**)



When **Highlight facets = Yes**, the **Sensors** are displayed in the **View** window along with the facets, on which they locate



When you click the "  **Append item to the array (Ins)**" icon, location of a new **Sensor** will automatically continue the sequence determined by two previous **Sensors**. To change direction or step of this sequence, change location of the last preceding **Sensor** manually.

Format of files for saving Sensors

Each line of a text file, which stores **Sensors**, corresponds to one **Sensor** and contain the following data:

- coordinates X, Y, Z of the **Sensor** in the local coordinate system of the **Set of sensors**
- name of the **Sensor** enclosed in double quotation marks (")
- color of the **Sensor** encoded in the RGB format.

URL encoding is used.

Here is an example of a text file, in which a **Set of sensors** has been saved:

```
0.01 0 0 "N" 0xffff00
0 0 0 "N" 0xffff00
0 0 0 "N" 0xffff00
0 0 0 "N" 0xffff00
0 0 0 "N" 0xffff00
0 0.03 0 "N" 0xffff00
0 0.04 0 "N" 0xffff00
0 0 0 "N" 0xffff00
0 0 0 "N" 0xffff00
0 0 0 "N" 0xffff00
0 0 0 "N" 0xffff00
0 0 0 "N" 0xffff00
0 0 0 "N" 0xffff00
0 0 0 "N" 0xffff00
```

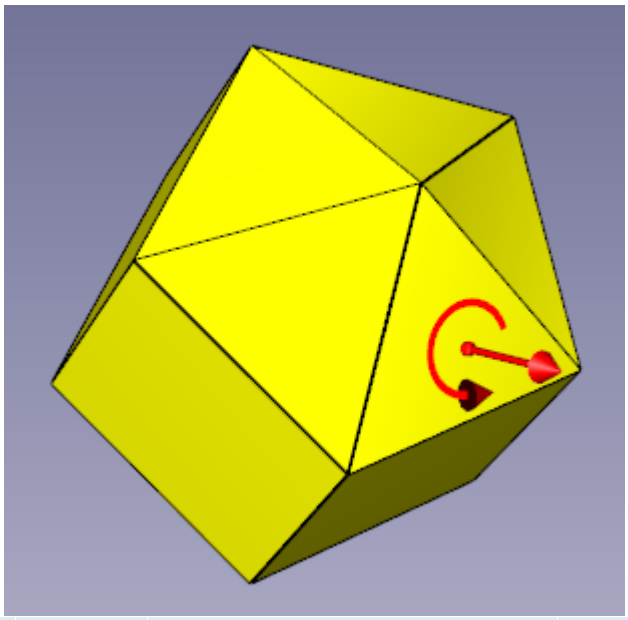
8.1.8.1.5.3 Element «Movement»

Movement is the element, which defines the law of motion of the **Object**, on which it has been created.

The **Movement** element in the project tree

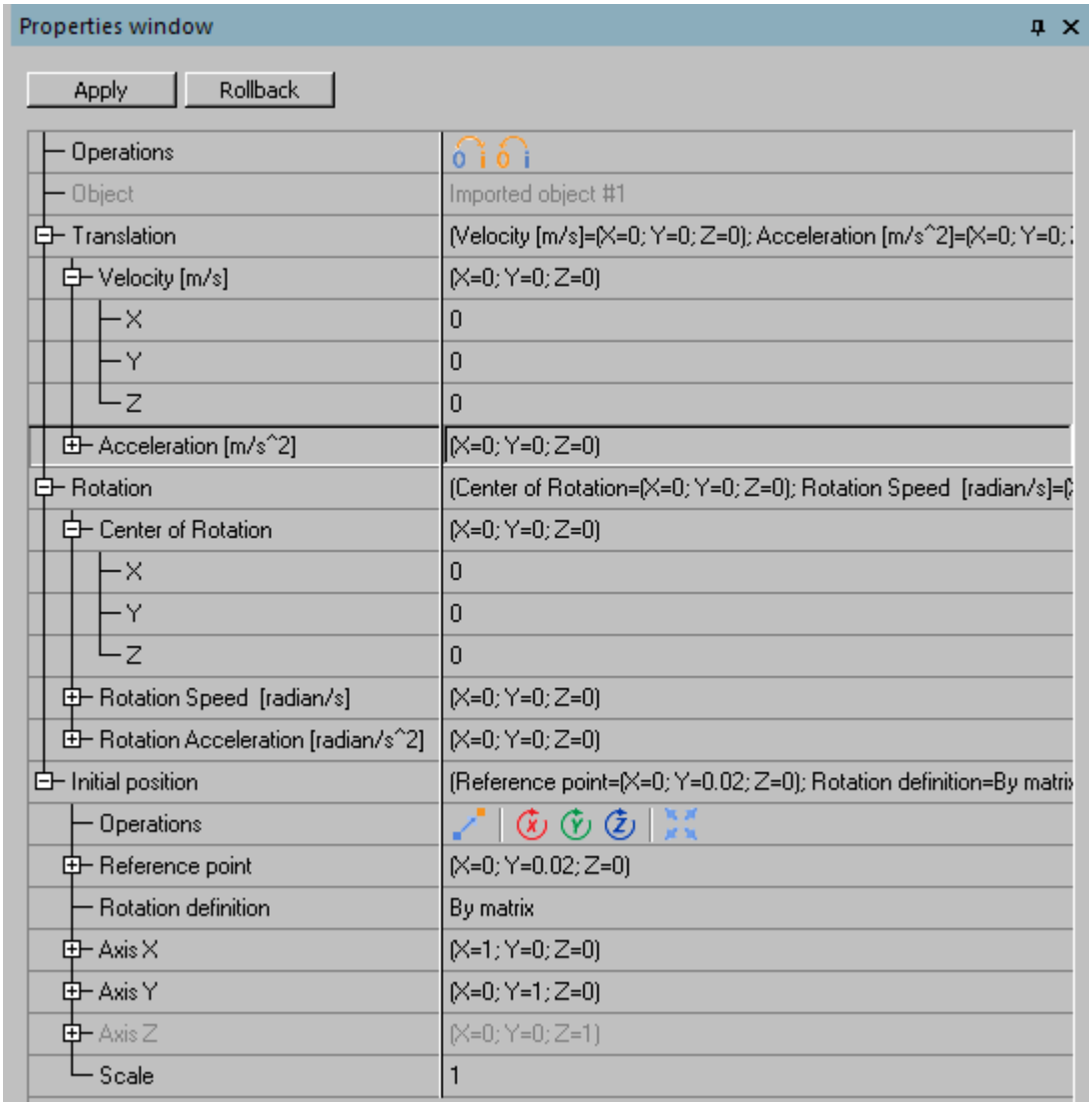
For an **Object** that has been created in the **Preprocessor** tab, the **Movement** element can only be created in the **Preprocessor** tab and it will be displayed in the project tree in the **Preprocessor** tab only.

When a rotational speed is specified for a **Movement**, and acceleration of the rotation is *not* specified, then, when the **Movement** is selected in the **Pre-Postprocessor**'s tree, its direction is displayed in the **View** window:



Rotation, which is set for a **Movement** element, is displayed by arrows (if *no acceleration* of the rotation is specified)

Parameters of the "Movement" element

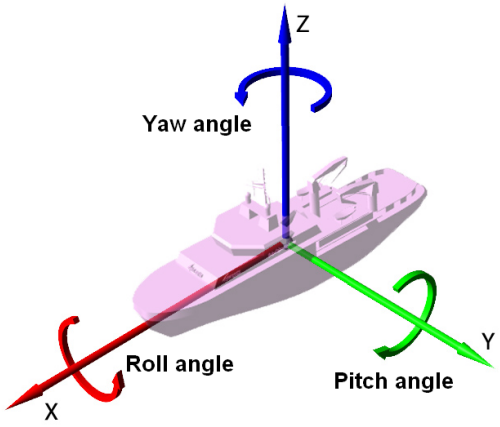
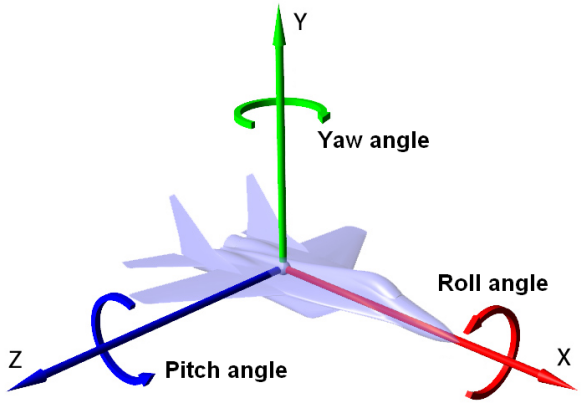


The **Properties** window of the **Movement** element

Parameters of the **Movement** element:

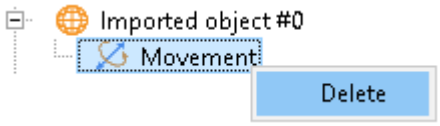
Parameter	Description
Operations > 	The **Object** is placed into its initial position.
Operations > 	The current position of the **Object** becomes its initial position.
Object	Name of the **Object**, which motion is specified. This field is informational only and is not editable.
Translation > Velocity > X	Components of velocity of the **Object**'s translational motion, specified in the absolute coordinate system (ACS), [m s^{-1}]
Translation > Velocity > Y	
Translation > Velocity > Z	
Translation > Acceleration > X	Components of the acceleration vector of the **Object**'s translational motion, specified in the absolute coordinate system (ACS), [m s^{-2}]
Translation > Acceleration > Y	

Parameter	Description
Translation > Acceleration > Z	
Rotation > Center of Rotation > X	Coordinates of the **Object**'s center of rotation (a point on the axis of rotation) in the local coordinate system of the**Object** (LCS-O)
Rotation > Center of Rotation > Y	
Rotation > Center of Rotation > Z	
Rotation > Rotation speed > X	Components of the vector of the rotation speed of the **Object** in the absolute coordinate system, [rad s^{-1}]. The direction of the vector defines the rotation axis, the module of the vector is the speed of rotation around this axis.
Rotation > Rotation speed >Y	
Rotation > Rotation speed >Z	
Rotation > Rotation Acceleration > X	Components of the vector of the angular acceleration of rotation of the **Object** in the absolute coordinate system, [rad s^{-2}]
Rotation > Rotation Acceleration > Y	
Rotation > Rotation Acceleration > Z	
Initial position	This group of parameters is similar to the Location group of parameters of the finite volume **Objects** and **Sets of sensors** (while the **Object**'s **Location** group of parameters becomes inactive).
Initial position > Operations	Screen buttons that allows you to easily perform actions with initial location of the **Object**: • (**Relative translation**) moves the **Object** relatively its current location in the absolute coordinate system • (**Coordinate system adjustment**) changes the **Object**'s local CS using the **Coordinate system adjustment** dialog box, which opens. • (**Relative rotation around local axis X**) rotates the **Object** around the X-axis of its local coordinate system • (**Relative rotation around local axis Y**) rotates the **Object** around the Y-axis of its local coordinate system • (**Relative rotation around local axis Z**) rotates the **Object** around the Z-axis of its local coordinate system • (**Relative scaling**) scales the **Object**'s current size relatively to the origin of the **Object**'s local coordinate system
Initial position > Reference point > X	Coordinates of the origin of the **Object**'s local coordinate system in the absolute coordinate system, [m]. For most **Objects**, this point is the geometric center. By default, the **Reference point** locates in the center of the computational domain and coincides with the origin of the absolute coordinate system.
Initial position > Reference point > Y	
Initial position > Reference point > Z	
Initial position > Rotation definition	Method of specifying the rotational displacement of the local coordinate system (LCS) in the absolute coordinate system (ACS). Possible options are: • **By matrix**: the rotational displacement is set by the matrix of projections of LCS' axes on ACS' axes. • **Around custom axis**: the rotational displacement is set by specifying the axis of rotation and angle of rotation around it.

Parameter	Description
	• **Euler angles (Naval)**: the rotational displacement is set by three angles (naval): pitch, roll and yaw:  • **Euler angles (Aviation)**: the rotational displacement is set by three angles (aviation): yaw, pitch, roll: 
Initial position > Axis X > X **Initial position > Axis X > Y** **Initial position > Axis X > Z**	The unit vector **i** of the local coordinate system in the absolute coordinate system (these parameters are available when **Rotation definition = By matrix**)
Initial position > Axis Y > X **Initial position > Axis Y > Y** **Initial position > Axis Y > Z**	The unit vector **j** of the local coordinate system in the absolute coordinate system (these parameters are available when **Rotation definition = By matrix**)
Initial position > Axis Z > X **Initial position > Axis Z > Y** **Initial position > Axis Z > Z**	The unit vector **k** of the local coordinate system in the absolute coordinate system (these parameters are available when **Rotation definition = By matrix**) These are read-only parameters and cannot be edited here.
Initial position > Rotation angle	Rotation angle of the **Object** around the given axis (this parameter is available when **Rotation definition = Around custom axis**)

Parameter	Description
Initial position > Axis of rotation > X	These parameters specify components of a vector, which defines the **Axis of rotation** (these parameters are available when **Rotation definition = Around custom axis**).
Initial position > Axis of rotation > Y	
Initial position > Axis of rotation > Z	
Initial position >Pitch angle *(naval)*	The angle of the first rotation, around initial position of the local axis Y, [degree] (this parameter is available when **Rotation definition = Euler angles (Naval)**)
Initial position >Roll angle *(naval)*	The angle of the second rotation, around initial position of the local axis X, [degree] (this parameter is available when **Rotation definition = Euler angles (Naval)**)
Initial position >Yaw angle *(naval)*	The angle of the third rotation, around initial position of the local axis Z, [degree] (this parameter is available when **Rotation definition = Euler angles (Naval)**)
Initial position >Yaw angle *(aviation)*	The angle of the first rotation, around initial position of the local axis Y, [degree] (this parameter is available when **Rotation definition = Euler angles (Aviation)**)
Initial position >Pitch angle *(aviation)*	The angle of the second rotation, around initial position of the local axis Z, [degree] (this parameter is available when **Rotation definition = Euler angles (Aviation)**)
Initial position >Roll angle *(aviation)*	The angle of the third rotation, around initial position of the local axis X, [degree] (this parameter is available when **Rotation definition = Euler angles (Aviation)**)
Initial position > Scale	The scaling factor affecting the reference point of the **Movement**

Context menu of the element "Movement"



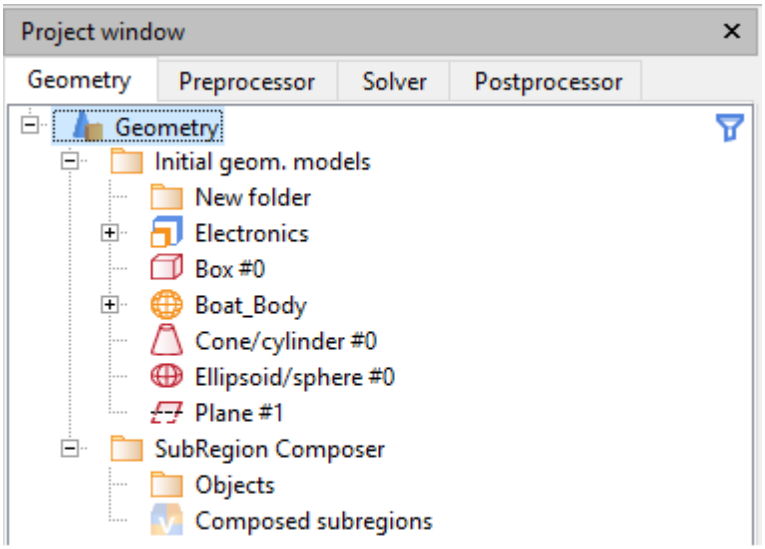
Context menu of the **Movement** element

Context menu of the element **Movement** in the project tree:

The menu command	Description
Delete	Deleting the selected element from the project tree

See also: Movement of geometric objects.

8.1.8.2 The Project window, tab «Geometry»



The **Geometry** tab is used for previous preparation of the project's geometry elements. The prepared geometry elements can be used as geometry of the computational domain, **Subregions** and **Imported objects**.

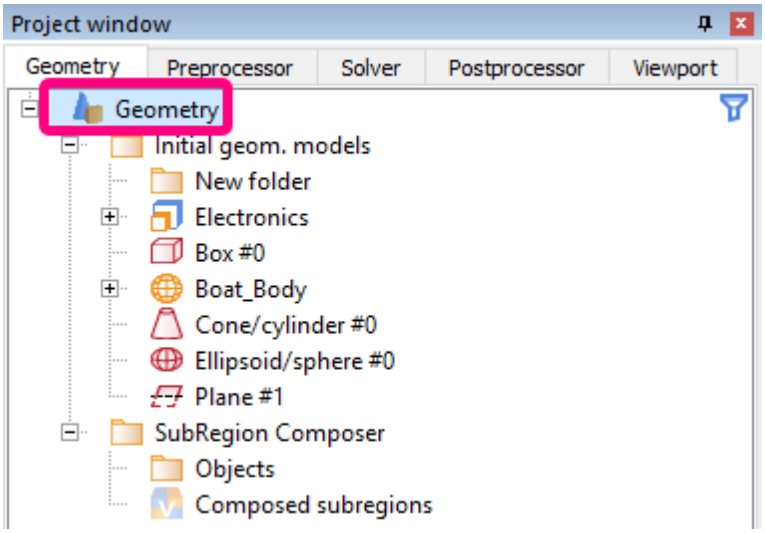
The **Geometry** tab allows you to start your work with a project from preparing geometry of the computational domain or of a part of the computational domain. Here you can load the geometry in the mesh format and in the parametric format (the last is possible when the *3DTransVidia* program for *FlowVision* is installed), and also you can create some standard geometric objects (**Boxes**, **Cone/cylinders**, **Ellipsoids/spheres**). You can use these standard geometric objects and parametric geometric objects to form mesh geometric objects. If a mesh surface consists of several disconnected fragments, you can split the surface.

Any mesh geometry can be exported into **Preprocessor** or into **Postprocessor** as **Imported objects**. If the geometry is formed for the **Region** (the whole computational domain consisting of several nested **Subregions**), then you have to compose mesh surfaces, which require to carry out scaling and mutual shifts of the surfaces, and the surfaces, which will be used in the new *FlowVision* project as geometry models for the **Region**'s **Subregions**, are also have to be examined for intersections.

Contents of the **Geometry** tab is grouped into the following folders and elements:

- the **Geometry** root folder
- the **Initial geom. models** folder, which might also contain user-defined folders that are useful for convenient work with several geometries (assemblies)
- the **SubRegion Composer** folder with subfolders **Objects** and **Composed subregions**.

8.1.8.2.1 Root folder «Geometry»



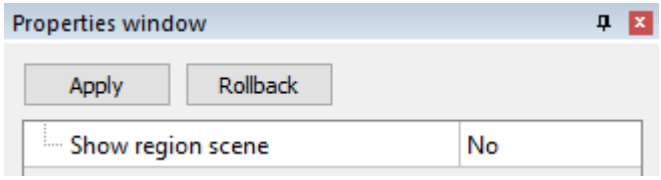
The root folder **Geometry** contains:

- the **Initial geom. models** folder, which contains geometry models
- the **SubRegion Composer** folder, which contains subfolders **Objects** and **Composed subregions**

The root folder **Geometry** has no context menu.

Parameters of the «Geometry» root folder

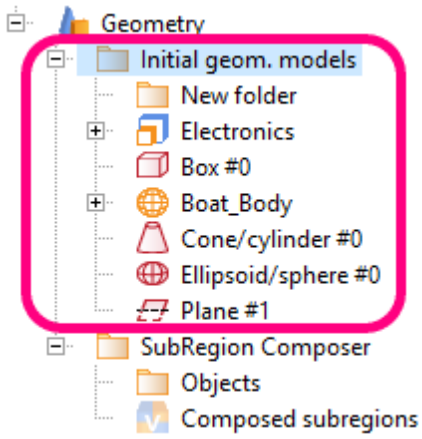
The **Geometry** root folder contains only one parameter in its **Properties** window, **Show region scene**:



Parameters of the root folder **Geometry**:

Parameter	Description
Show region scene	This parameter enables or disables displaying the objects that are presented in the **Preprocessor** tab of the **Project** window. Possible options are: **Yes** | **No**.

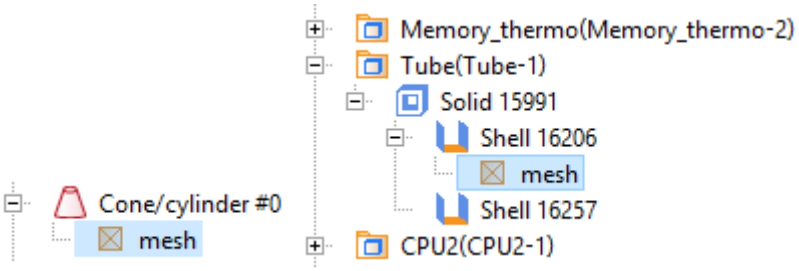
8.1.8.2.2 Folder «Initial geom. models»



The **Initial geom. models** folder in the project tree

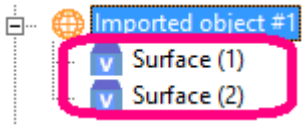
The **Initial geom. models** folder contains child elements that present clipping **Planes**, analytical **Geometric objects** (**Boxes**, **Cone/cylinders**, **Ellipsoids/spheres**), imported geometric **Elements of CAD models** specified in a parametric format. (they are available for import only when the *3DTransVidia* program is installed), **Imported objects** with mesh geometry and also user-defined **folders** with similar contents.

It is possible to generate consistent **meshes** on surfaces of analytical **Geometric objects** and imported geometric **Elements of CAD models** specified in a parametric format. The **meshes** are presented in the project tree as child elements:



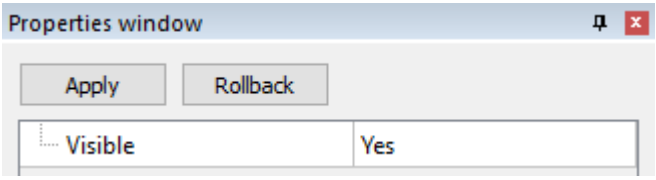
Presenting an Imported object with mesh geometry in the project tree

If an **Imported object** with mesh geometry consists of several unconnected surfaces, these surfaces are presented in the project tree as child elements with default names **Surface (1)**, **Surface (2)**, etc.:



Parameters of the «Initial geom. models» folder

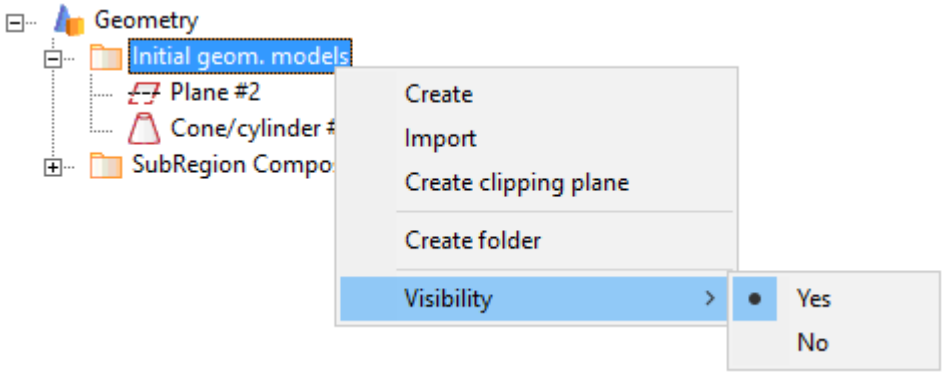
The **Initial geom. models** folder contains only one parameter in its **Properties** window, **Visible**:



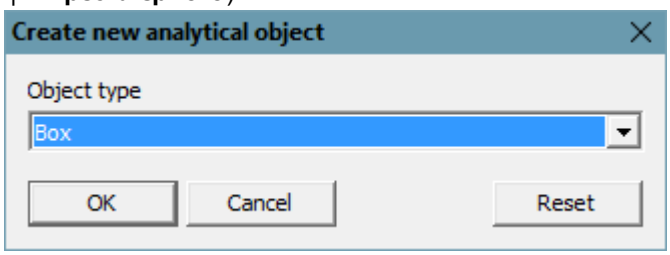
Parameters of the folder **Initial geom. models**:

Parameter	Description
Visible	This parameter enables or disables displaying the objects that are presented in the **Initial geom. models** folder if this setting isn't overridden by individual settings of user-defined folders and objects. Possible options are: **Yes** | **No**. This parameter can also be set by the **Visibility** item from the context menu of the **Initial geom. models** folder (see below).

Context menu of the folder "Initial geom. models"

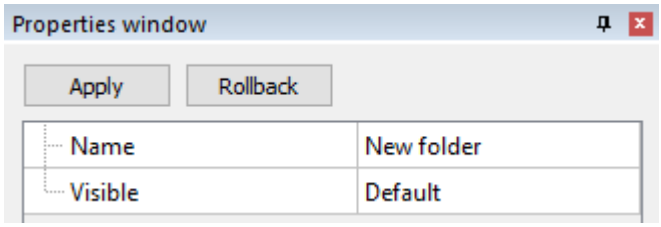


The context menu of the folder **Initial geom. models** contains the following items:

Menu item	Description
Create	Creates a new analytical geometric object. The **Create new analytical object** dialog box will open where you can select the **Object type** (possible options are **Box** | **Cone/cylinder** | **Ellipsoid/sphere**):  After clicking **OK** a new analytical geometric object will appear in the project tree as a child element in the **Initial geom. models** folder.
Import	Imports a geometric object from a file. A standard operating system's window for access to files will open where you should select a file, from which the object will be imported. The **Imported object** will be presented in the project tree as a child element in the **Initial geom. models** folder.

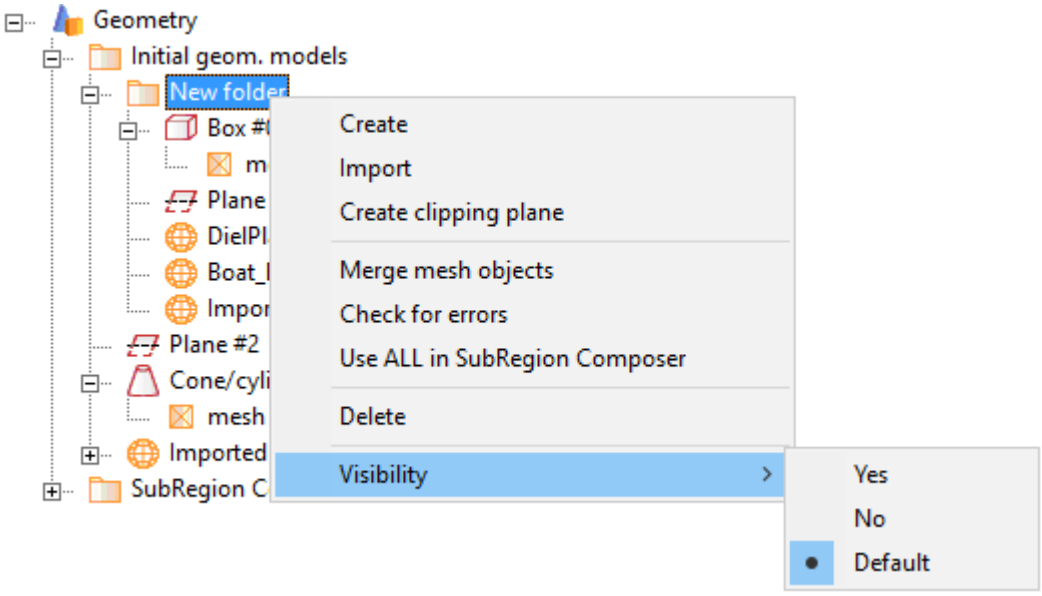
Menu item	Description
	Depending on type of the file, which contains a mesh or parametric geometry model, the program might request parameters of the import.  Import of geometry allows selection of multiple files so batch import in this case will be done. If the **Import** command is selected from the context menu of the **Initial geom. models** folder, then a new user-defined folder will be created, into which objects from files will be imported. If the **Import** command is selected from the context menu of a user-defined folder, then the **Imported objects** will be added into the same user-defined folder.
Create clipping plane	Creates an new clipping **Plane** that will be presented as a child element in the **Initial geom. models** folder. The **Planes**, which are created using this command, affect only those images in the **View** window, which are displayed when the **Geometry** tab is opened.
Create folder	Creates an new user-defined folder in the **Initial geom. models** folder. You can place into the user-defined folders the same types of objects as are placed into the **Initial geom. models** folder itself. User-defined folders cannot be nested (this means that a user-defined folder cannot locate into another user-defined folder).
Visibility	This menu item contains a sub-menu with options **Yes** and **No** that specify visibility of initial geometry models if this isn't overridden by individual settings of user-defined folders and objects. The visibility is also toggled by the **Visible** parameter in the **Properties** window of the **Initial geom. models** folder.

Parameters of user-defined folders that are nested into the «Initial geom. models» folder



Parameter	Description
Name	Name of the user-defined folder. At their creation, user-defined folders are named as **New folder**, **New folder (2)**, **New folder (3)**, etc. If you wish, you can change the name of a user-defined folder by changing the value of this parameter.
Visible	Displaying objects that are presented in the user-defined folder (if some else isn't set individually in properties of the objects). Possible options are: • **Yes**: enable the displaying • **No**: disable the displaying • **Default**: apply the mode of displaying, which is set in properties of the parent folder **Initial geom. models** This parameter can also be set by the **Visibility** item from the context menu of the user-defined folder (see below).

Context menus of user-defined folders that are nested into the «Initial geom. models» folder



Context menus of user-defined folders that are nested into the **Initial geom. models** folder contain the following items:

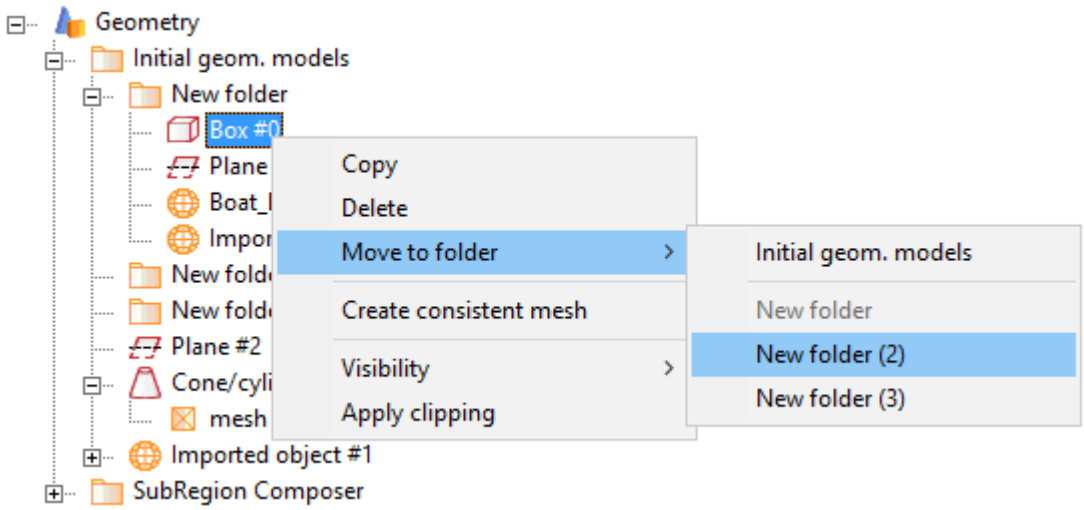
Menu item	Description
Create	Adds a child object into the folder (such object can be: **Box**, **Cone/cylinder**, **Ellipsoid/sphere**, **Imported object**, clipping **Plane**). These menu items are similar to the having the same name menu items from the context menu of the **Initial geom. models** folder.
Import	
Create clipping plane	
Merge mesh objects	Merges **Imported object** that are presented in the user-defined folder (imported from files and created from **meshes** by the command **Copy as mesh geometry**, see subsection *"Context menus of objects of initial geometry models"* below). As a result of applying this command, a new **Imported object** will be created and presented in the **Initial geom. models** folder, which is the parent folder for this user-defined folder.
Check for errors	Checks geometry of **Imported object** that are presented in the user-defined folder for errors.
Use ALL in SubRegion Composer	Uses *all* **Imported objects**, which are presented in the folder, to compose **Subregions**. After applying this command all these **Imported objects** will be presented in the folder **SubRegion Composer > Objects**.
Delete	Deletes the selected user-defined folder including its contents. If the folder isn't empty, the program request you to confirm this decision.
Visibility	This menu item contains a sub-menu with options **Yes | No | Default** that specify visibility of initial geometry models if this isn't overridden by individual settings of the objects. This visibility setting can also be set by the **Visible** parameter in the **Properties** window of the user-defined folder.

Parameters of objects of initial geometry models

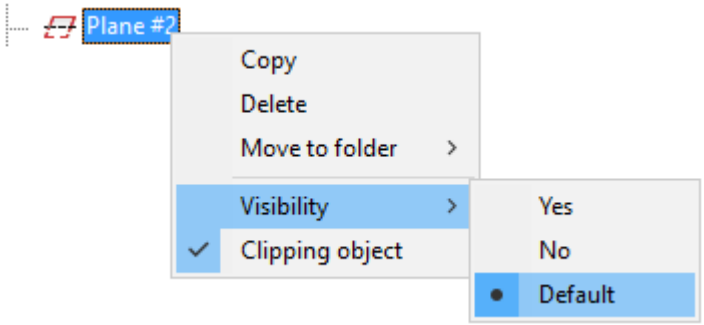
Objects of initial geometry models have in their **Properties** windows the same parameters as **Geometric objects** in other tabs of the project tree. See details in sections:

- Object «Plane» (user interface)
- Object «Box» (user interface)
- Object «Cone/cylinder»(user interface)
- Object «Ellipsoid/sphere» (user interface)
- Imported objects (user interface)

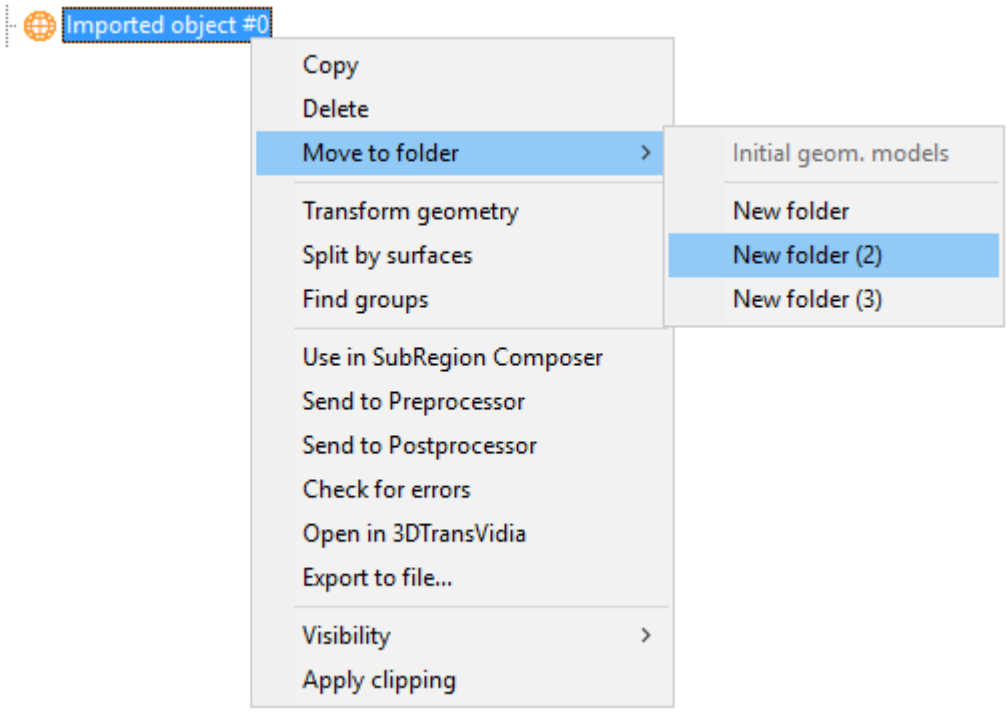
Context menus of objects of initial geometry models



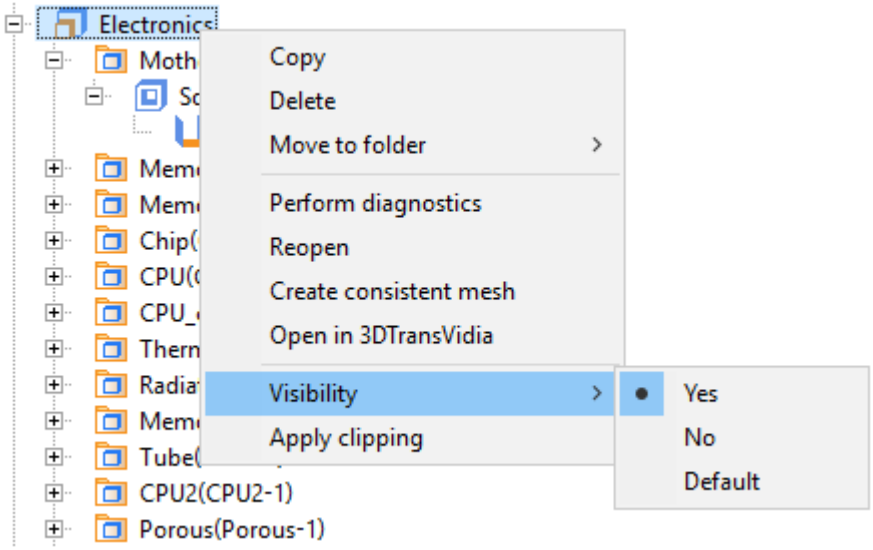
Context menu of an analytical geometric object presented in the folder **Initial geom. models** or in a nested user-defined folder



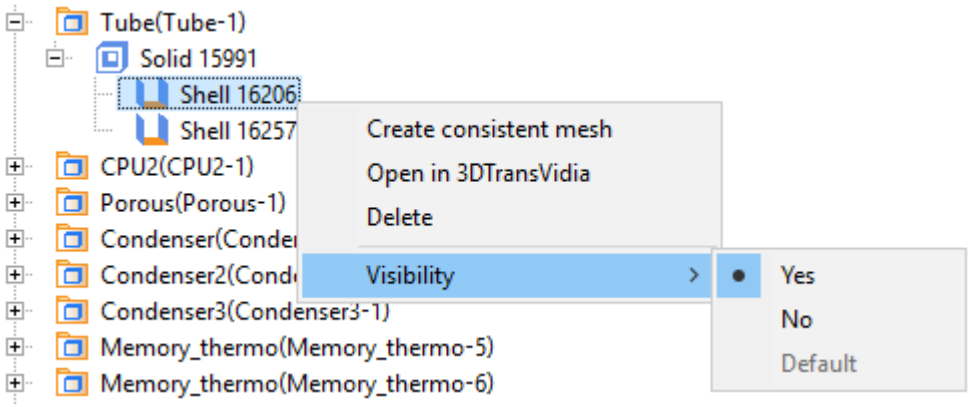
Context menu of a **Plane** in the folder **Initial geom. models** or in a nested user-defined folder



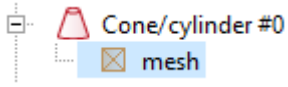
Context menu of an **Imported object** in the folder **Initial geom. models** or in a nested user-defined folder

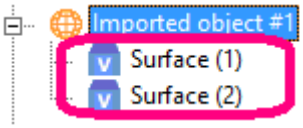
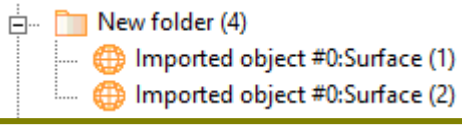


Context menu of a root node of an element of a CAD model in the folder **Initial geom. models** or in a nested user-defined folder



Context menu of a non-root node of an element of a CAD model in the folder **Initial geom. models** or in a nested user-defined folder

Menu item	Description
Copy	Copy the selected object within the same folder.
Delete	Delete the selected object.
Move to folder > ...	Move the selected object into the folder, which is selected from a sub-menu.
Perform diagnostics[4)]	This command runs diagnostics of the selected Element of a CAD model.
Reopen[4)]	This command imports the selected Element of a CAD model again with ability to specify the geometry tolerance. Before the import the **Choose tolerance** dialog box opens with the following options: • **From file** • **Estimated (recommended)** • **User defined** Near selectors of these options the dialog box contains numerical fields that display values of the geometry tolerance.
Create consistent mesh[3)]	Create a mesh by the surface of the selected analytical geometric object or Element of a CAD model. For **Cones/cylinders** and **Ellipsoids/spheres** the mesh is generated based on parameters from the **Approximation** group in the properties of the geometric object. When the mesh is generated on a **Box**, the **Approximation parameters** dialog box opens where density of the mesh along the local coordinate axes of the **Box** is specified. The resulting **mesh** is presented in the project tree as a child element of the geometric object, on which it has been generated:  See also subsections *"Parameters of consistent mesh"* and *"Context menus of the consistent mesh"* below.
Transform geometry[*)]	Transform geometry of the selected **Imported object** according to parameters that are set in the **Geometry transformation** dialog box, which opens. See details in the section Transformation of geometry model of the computational domain and imported object.
Split by surfaces[*)]	Create new **Imported objects** based on **Surfaces** that are presented in the project tree as child elements of the selected **Imported object**:

Menu item	Description
	 As a result of applying this command, a new user-defined folder will be created, in which the new created **Imported objects** will be presented with names formed from name of the source **Imported object** and names of the **Surfaces**: 
Find groups[*)]	Looking for groups of facets of the selected **Imported object**.
Use in SubRegion Composer[*)]	Use the selected **Imported object** for composing **Subregions**.As a result of applying this command, the selected **Imported object** will be presented in the folder **SubRegion Composer > Objects**.
Send to Preprocessor[*)]	Copy the selected **Imported object** to the folder **Objects** in the **Preprocessor** tab. The copied **Imported object** will have a new name there. Work with the copied object is carried out independently on actions in the **Geometry** tab.
Send to Postprocessor[*)]	Copy the selected **Imported object** to the folder **Objects** in the **Postprocessor** tab. The copied **Imported object** will have a new name there. Work with the copied object is carried out independently on actions in the **Geometry** tab.
Check for errors[*)]	Check the selected **Imported object** for errors in its geometry.
Open in 3DTransVidia[*)]	Open the selected **Imported object** in the *3DTransVidia* program. This allows you to correct the geometry in the *3DTransVidia* program and then return to work with the geometry in **Pre-Postprocessor**.
Export to file[*)]	Export the selected **Imported object** to a file, see details in the section Exporting a geometric model into a file (step-by-step procedure).
Visibility > ...	This menu item contains a sub-menu with items **Yes | No | Default** that specify visibility of the selected object. When visibility of an object is disabled, the object's icon in the project tree is faded.
Clipping object[**)]	This menu item defines if the selected **Plane** is a clipping object (see details in the section Object «Plane»).
Apply clipping[3)]	This menu item defines if clipping by clipping objects (**Planes**) is applied to the selected object.

[*)] This item contains in context menus of **Imported objects** and **Elements of CAD models** only. To do similar actions for analytical geometric objects (**Boxes**, **Cones/cylinders**, **Ellipsoids/spheres**), you have apply to them the **Create consistent mesh** command and then apply the required command from the context menu of the created consistent **mesh** (see subsection *"Context menu of the consistent mesh"* below).

[**)] This menu item is included in context menus of **Planes** only.

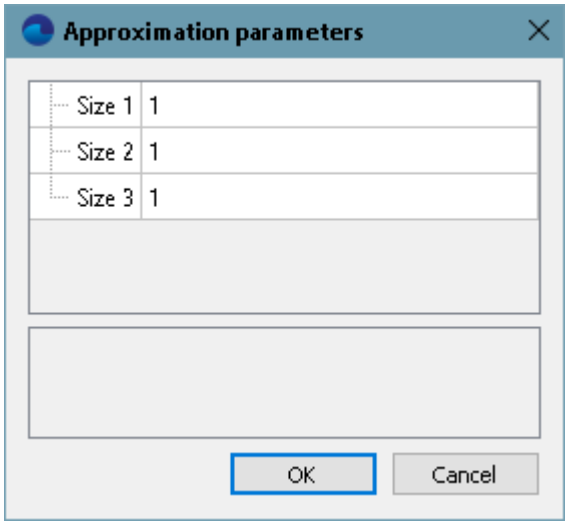
[3)] This menu item is absent in context menus of **Planes**.

[4)] This menu item is included in context menus of **Elements of CAD models** only.

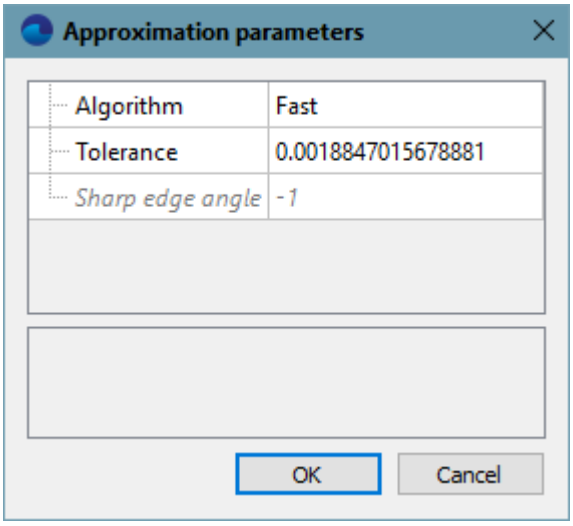
Parameters of the consistent mesh

The consistent **mesh** at its creation is formed differently depending on the type of the geometric object, on surface of which it has been generated:

- for **Cones/cylinders** and **Ellipsoids/spheres**: by approximation parameters of the geometric object (**Subdivisions**, **Per sector**, **Topology**, **Base**, **Iterations**);
- for **Boxes** according to density of the **mesh** along axes of the object's local coordinate system, specified by the user in the **Approximation parameters** dialog box.
- for imported **Elements of CAD models** and its components (parts, solid bodies, surfaces) according to parameters **Algorithm**, **Tolerance**, and **Sharp edge angle** specified by the user in the **Approximation parameters** dialog box.



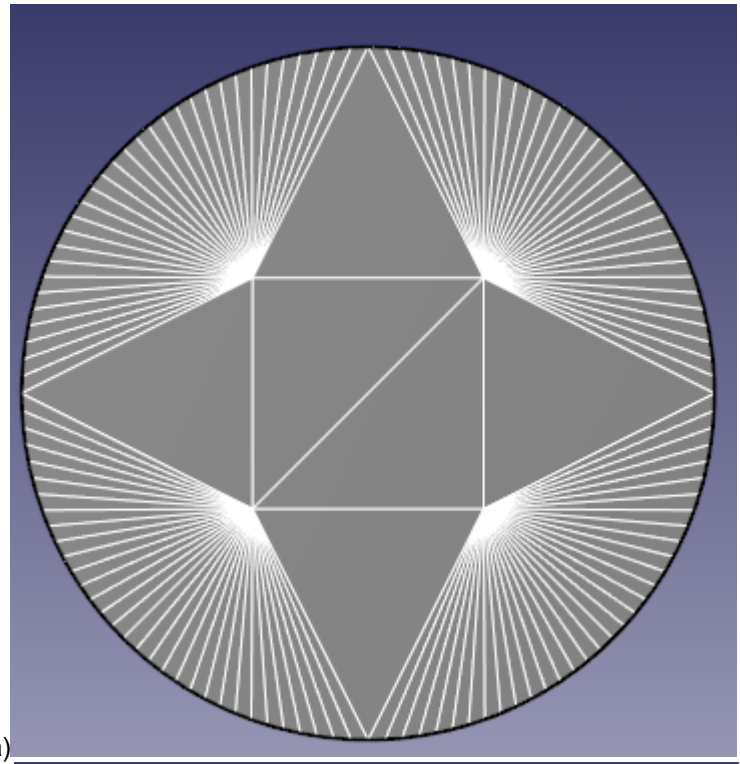
The **Approximation parameters** dialog box for specifying parameters of a consistent mesh on a **Box**

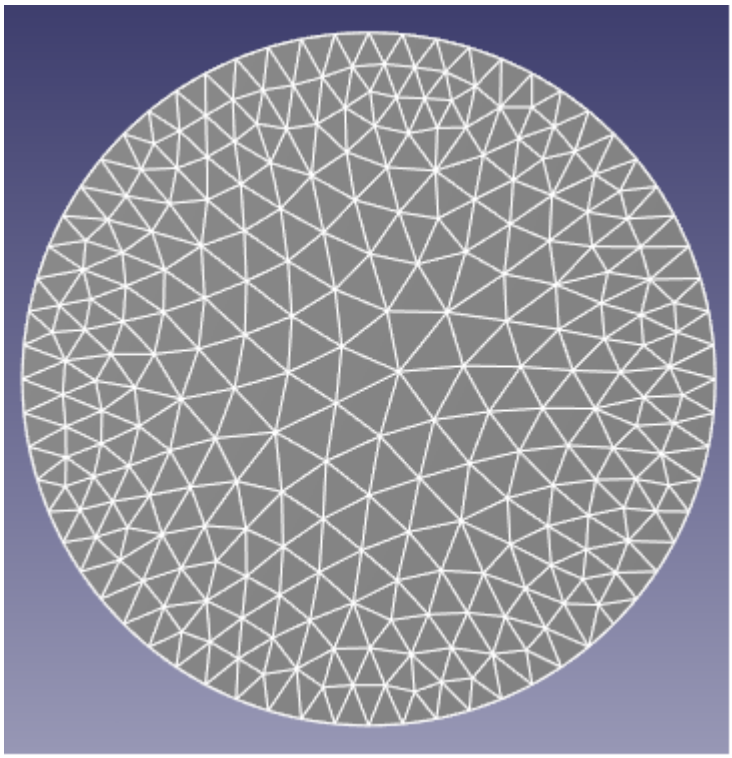


The **Approximation parameters** dialog box for specifying parameters of a consistent mesh on an element of a CAD model

When a consistent mesh is generated on an element of a CAD model, the **Approximation parameters** dialog box contains the following parameters:

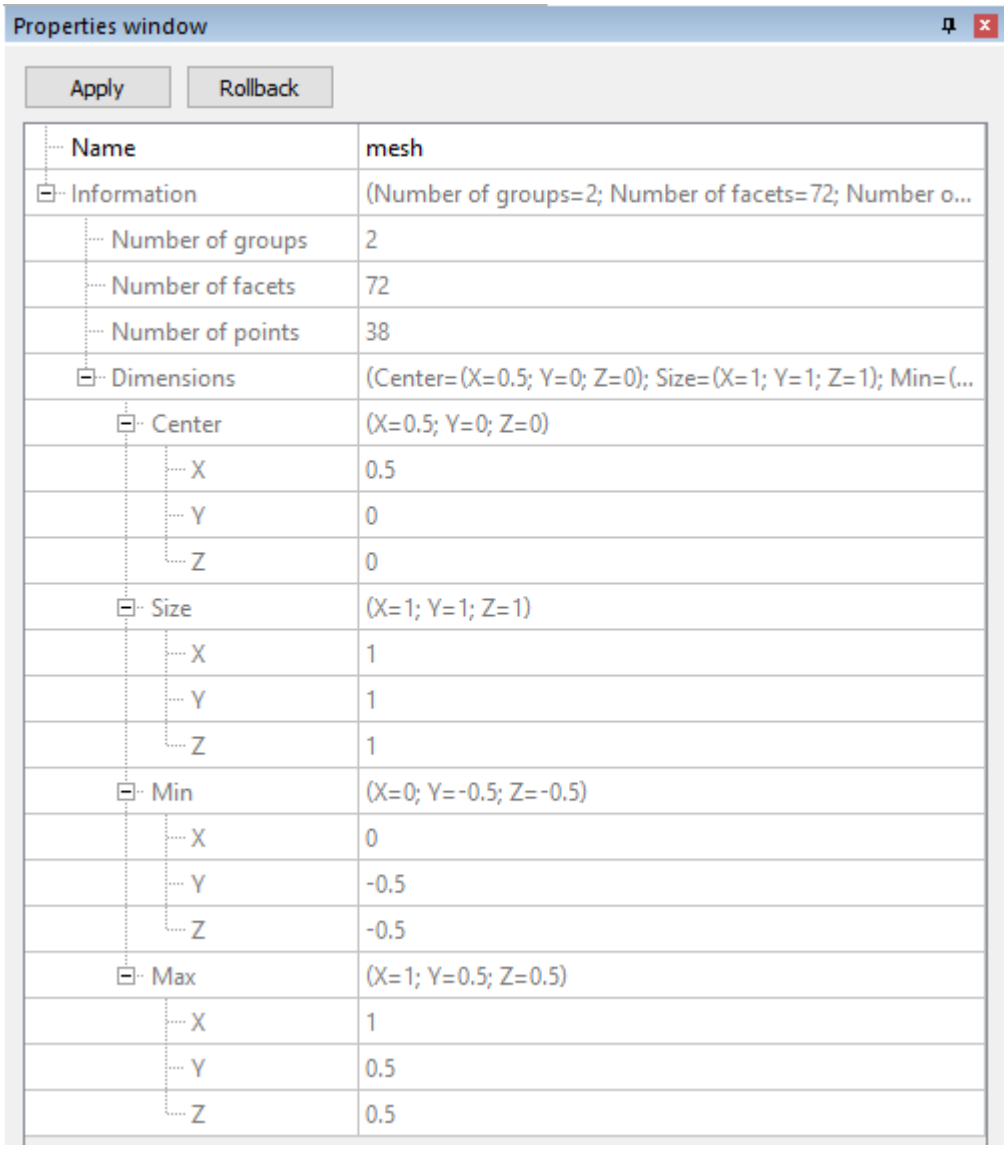
Parameter	Description
Algorithm	An algorithm that will be used to generate a consistent mesh. Possible options are: **Fast** | **Uniform**. The **Fast** algorithm, comparing to the **Uniform** algorithm, generates meshes with generally less number of facets. The **Uniform** algorithm allows you to limit the sharp angles of facets and generates mesh geometry with better quality and relatively uniform facets for all surfaces. The **Uniform** algorithm generally requires more time and generates more facets so this causes more consumption of the random access memory during the computation. See the illustrations below this table.
Tolerance	This is the linear deviation of the approximating surface from the original surface.
Sharp edge angle	This is the threshold degree between edges, [degrees]. This parameter is only available when **Algorithm = Uniform**.

a)  b)



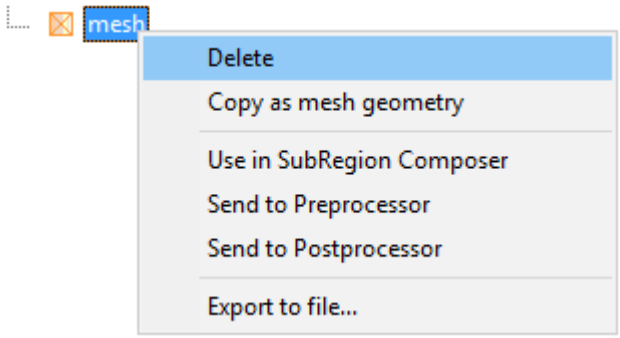
Consistent mesh generated on an element of a CAD model: a) by the **Fast** algorithm; b) by the **Uniform** algorithm

Parameters of a **mesh** are displayed in its **Properties** window and they, except **Name**, are read-only:



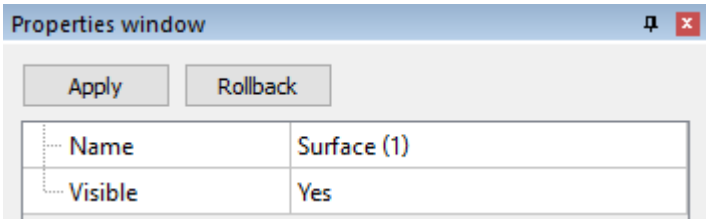
Parameter	Description
Name	Name of the consistent **mesh**. By default, names **mesh**, **mesh (2)**, **mesh (3)**, etc. are used. If you wish, you can change the **mesh**'s name.
Information > Number of groups	Number of groups of facets in the **mesh**
Information > Number of facets	Number of facets in the **mesh**
Information > Number of points	Number of points in the **mesh**
Information > Dimensions > Center > ...	Coordinates of the center of the **mesh** on axes X, Y, Z
Information > Dimensions > Size > ...	Sizes of the **mesh** along axes X, Y, Z
Information > Dimensions > Min > ...	Minimal coordinates of the **mesh** on axes X, Y, Z
Information > Dimensions > Max > ...	Maximal coordinates of the **mesh** on axes X, Y, Z

Context menu of the consistent mesh



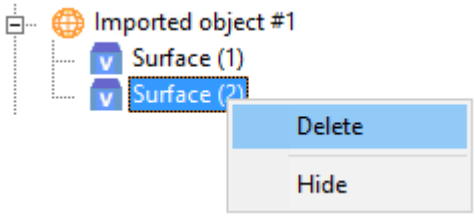
Menu item	Description
Delete	Delete the **mesh**.
Copy as mesh geometry	Create an **Imported object** based on the **mesh** and place it in the project tree in the folder of the parent analytical geometric object.
Use in SubRegion Composer	This commands are similar to the commands from context menus of **Imported objects** (see subsection *"Context menus of objects of initial geometry models"* above).
Send to Preprocessor	
Send to Preprocessor	
Export to file	

Parameters of a Surface of an Imported object

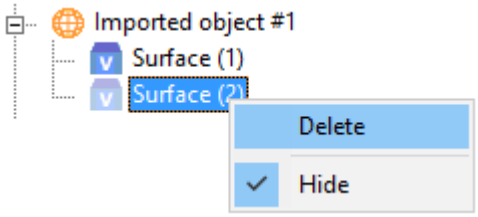


Parameter	Description
Name	Name of the **Surface**. By default, the following names are used: **Surface (1)**, **Surface (2)**, etc. If you wish, you can change the **Surface**'s name.
Visible	Displaying the **Surface**. Possible options are:**Yes** | **No**. This parameter can also be set by the **Hide** item from the context menu of the **Surface** (see subsection *"Context menu of a Surface of an Imported object"* below).

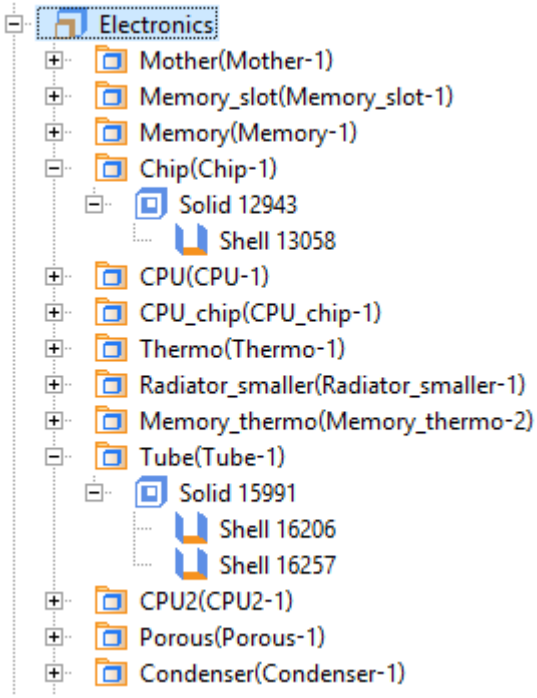
Context menu of a Surface of an Imported object



Menu item	Description
Delete	Delete the **Surface**.

Menu item	Description
Hide	This menu item toggles visibility of the **Surface** (this can also be set by the **Visible** parameter in properties of the **Surface**, see subsection *"Parameters of a Surface of an Imported object"* above). When the **Surface** is hidden, this menu item is checked and the **Surface**'s icon is faded: 

Geometric elements of CAD-models

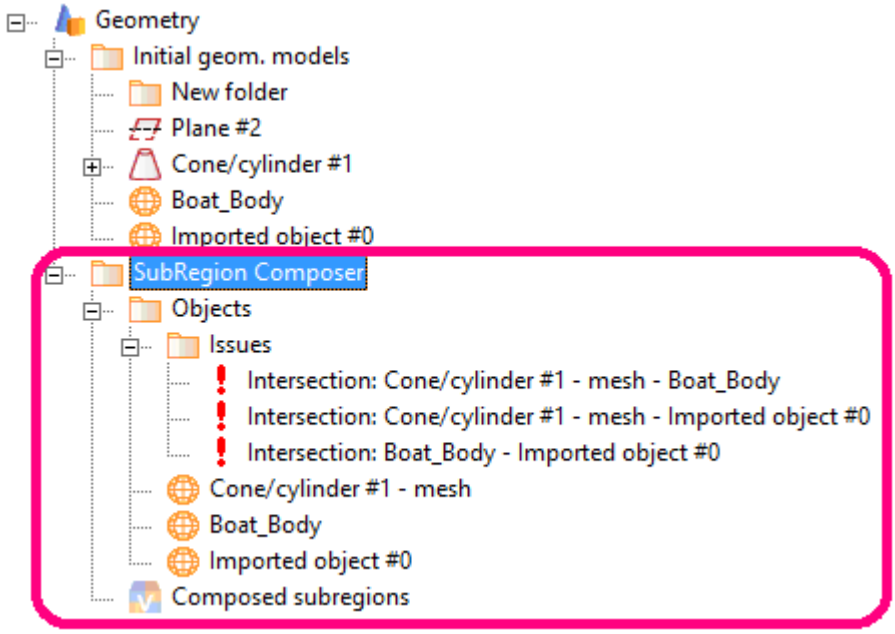


Geometric objects in a parametric format that were imported to the **Geometry** tab from a CAD model using the *3DTransVidia* software, are presented in the project tree as subtrees with following types of nodes:

Icon in the project tree	Node type
	The root node of an imported geometric **Element of CAD model** specified in a parametric format. This node corresponds to an assembly of several parts.
	A part, which can consist of one or several solids
	A solid which is included in a part. A solid can consist of one or several surfaces.
	A surface that forms (by itself or combining with other surfaces) a solid that is included in a part.

8.1.8.2.3 Folder «SubRegion Composer»

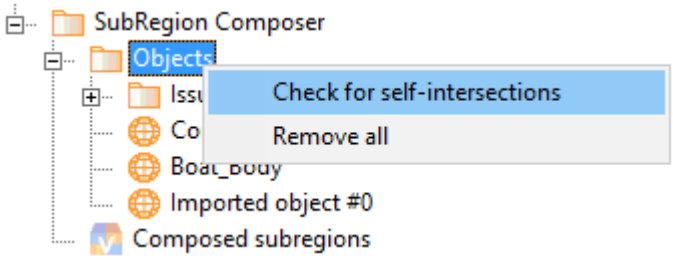
The **SubRegion Composer** folder represents structure of subregions formed by mesh geometries that are presented in the **Objects** subfolder.



The **SubRegion Composer** folder contains the following subfolders:

- the **Objects** subfolder, which contains **Imported objects** that can be used to compose **Subregions**, and optionally the **Issues** folder, which contains information about errors encountered at attempts of composing the **Subregions**. The **Objects** folder has no parameters in its **Properties** window. When an element, which contains in the **Issues** folder, is selected, contour of the appropriate intersection of surfaces is highlighted in the **View** window with red color.
- the **Composed subregions** subfolder, in which surfaces that are already used to compose **Subregions**, are presented.

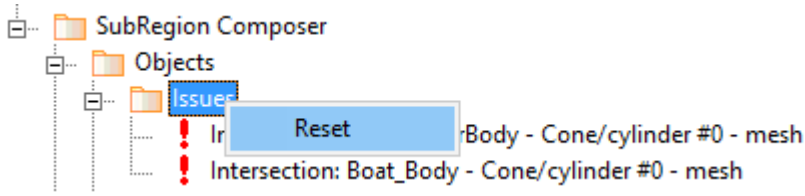
Context menu of the "SubRegion Composer > Objects" folder



Context menu of the **SubRegion Composer > Objects** folder contains the following items:

Menu item	Description
Check for self-intersections	Check all **Imported objects** that are presented in the **Objects** folder for intersections of their geometry. If intersections are found, appropriate elements will appear in the **Issues** folder. If no intersections are found, the "**No issues found**" message will be displayed. Applying this command doesn't cause presenting the checked **Imported objects** in the **Composed subregions** folder.
Remove all	Clear all contents from the **Objects** folder totally. This command removes all **Imported objects** that are presented in the **Objects** folder and the **Issues** folder if any including its contents.

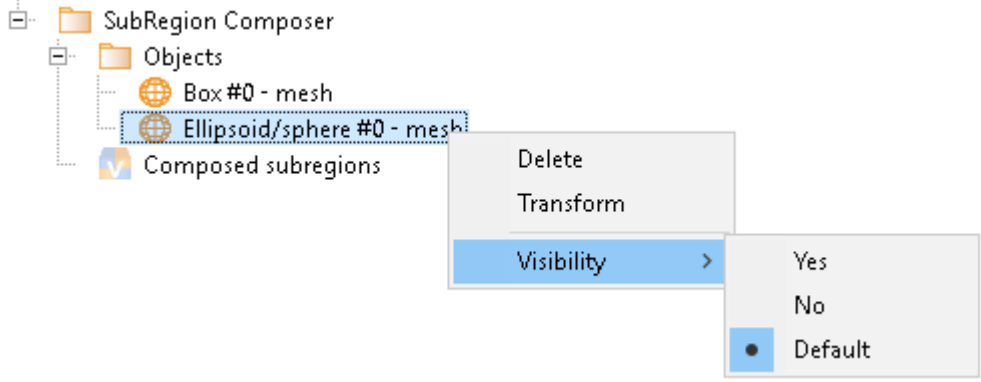
Context menu of the "Objects > Issues" folder



Menu item	Description
Reset	Delete the **Objects > Issues** including its contents. If, during a new search, intersections of **Imported objects**' surfaces are found, the **Objects > Issues** folder appear again with elements that correspond to the intersections of surfaces.

Child elements that correspond to issues, have no parameters in the **Properties** window and their context menus have the **Properties** command only, which opens the **Properties** window if it isn't already opened.

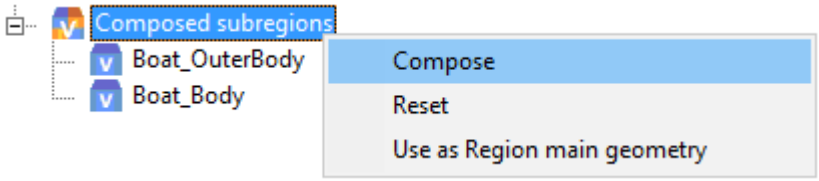
Context menu of the "SubRegion Composer > Objects > *(Imported object)*" element



Context menu of the element **SubRegion Composer > Objects > *(Imported object)*** contains the following items:

Menu item	Description
Delete	Delete the selected **Imported object**.
Transform	Transformation geometry of the selected **Imported object**. The **Geometry transformation** dialog box will open to specify parameters of the transformation.
Visibility > Yes	Specifying visibility parameters of the selected **Imported object**.
Visibility > No	
Visibility > Default	

Context menu of the "Composed subregions" folder



The context menu of the **Composed subregions** folder contains the following items:

Menu item	Description
Compose	Make an attempt of composing **Subregions** using surfaces of all **Imported objects** that are presented in the **Objects** folder. After applying this command, surfaces of the

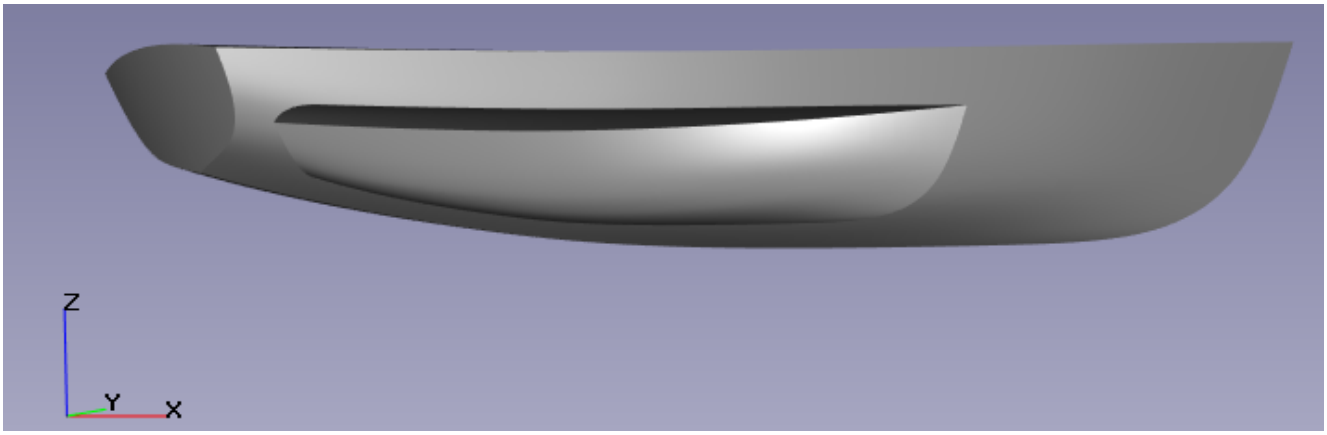
Menu item	Description
	Imported objects will be presented as child elements in the **Composed subregions** folder. If intersections of the surfaces are found, the **Objects > Issues** folder will be created or updated with elements that correspond to the intersections. If the next composing cause no intersections, the **Objects > Issues** folder will disappear.
Reset	Delete contents of the **Composed subregions** folder.
Use as Region main geometry	Use the surfaces, which are presented in the **Composed subregions** folder, as a geometry model of the project's computational domain that will be presented in the **Preprocessor** tab. If a geometry model is already presented in the **Preprocessor** tab, then the program will display a warning about possible lost of data (**The region already contains geometry. Want to replace it? Some data may be lost!**). After applying this command, the **Preprocessor** tab opens. The program does *not* check repeated attempts of transferring the mesh geometry.

Properties window of the "Composed subregions" folder

Button	Description
Operations > (Compose)	These screen buttons work similarly as context menu commands of the **Composed subregions** folder (see above).
Operations > (Reset)	
Operations > (Use as Region main geometry)	

Examples and illustrations

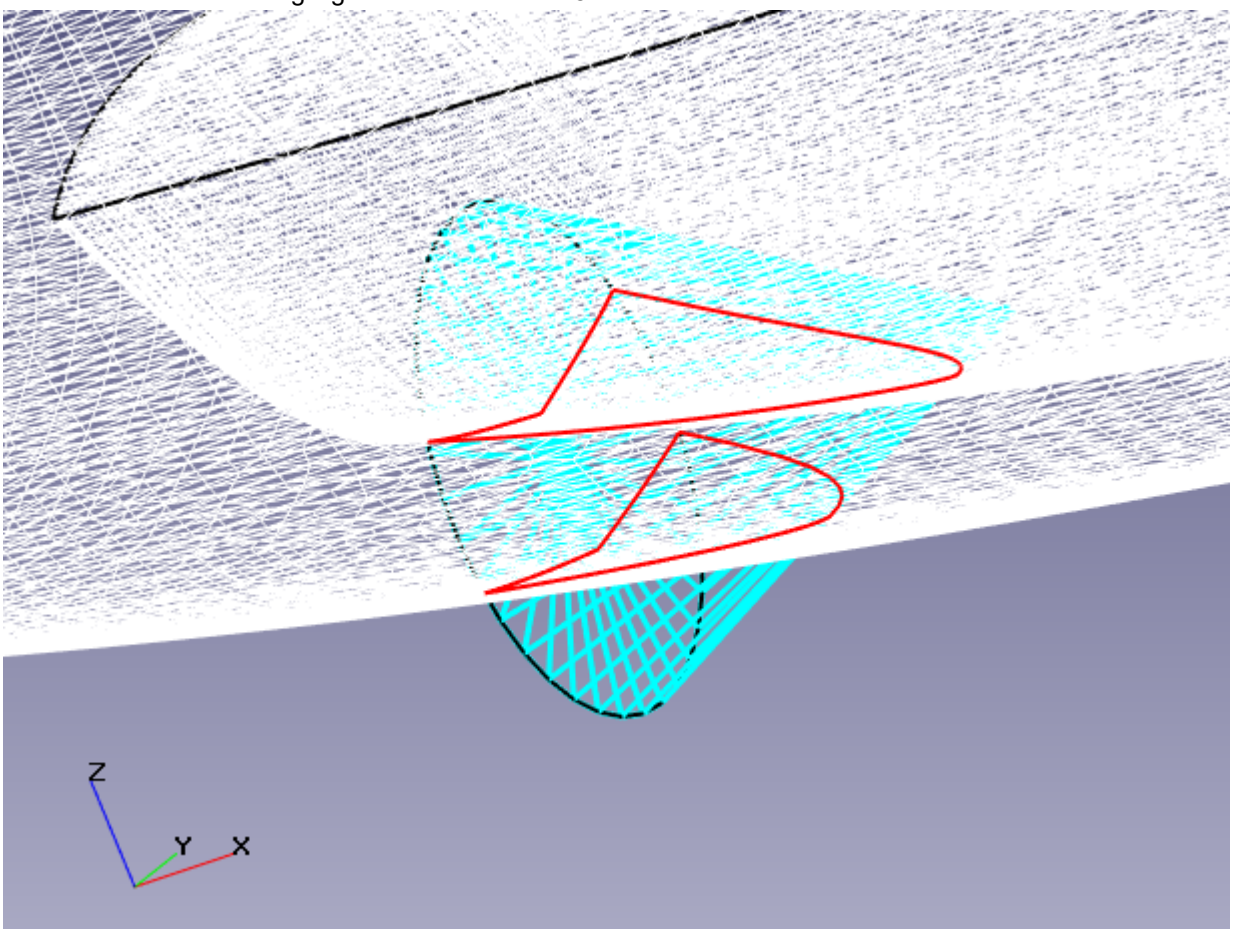
On the illustration below you can see composed subregions assembled from surfaces of two **Imported objects** that were created from the geometry model of a boat shell `Boat_Body.wrl` (geometry of one of the **Imported objects** was processed by a **Transformation**, during which the linear dimensions of the **Imported object** were increased):



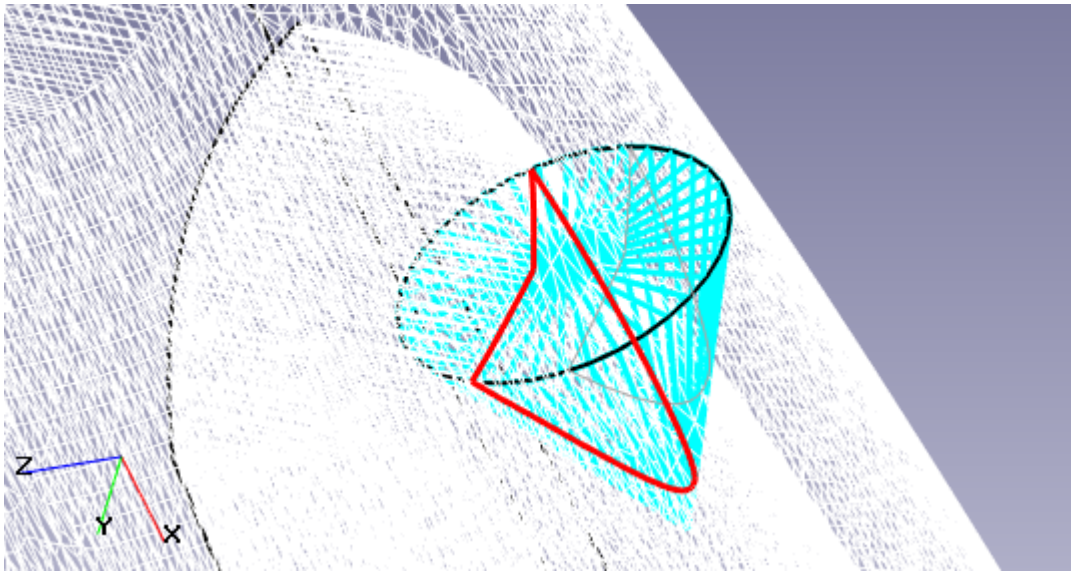
There are no intersections of surfaces in this example.

In the next example an **Imported object** that was created from a **Cone**, which intersects both previous **Imported objects**, was added into the **SubRegion Composer > Objects** folder. As a result, the **Objects > Issues** folder

was created, which contains two elements that correspond to contours of intersections of the **Cone** and two boat shells. These contours are highlighted in red in the **View** window:



When in the project tree an individual child element is selected in the the **Objects > Issues** folder, highlighting in red is applied to those contour only, which corresponds to the selected element (other contours are displayed in gray):

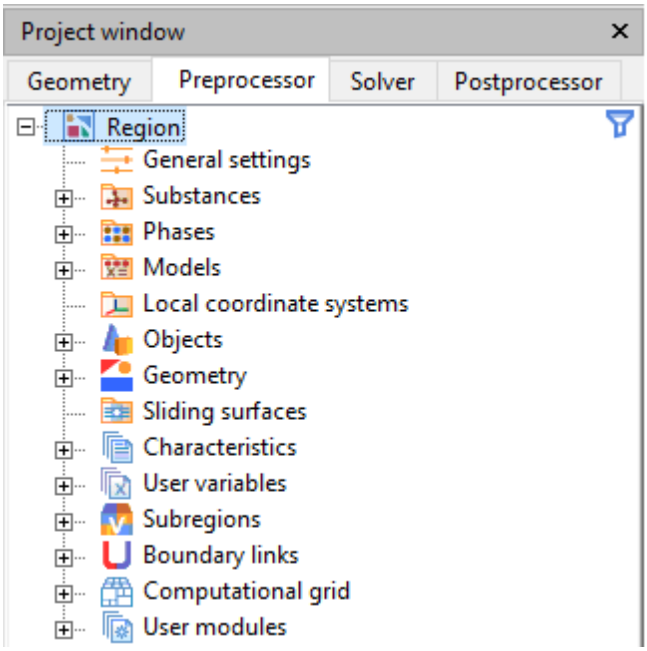


When intersections of geometry are found, the program displays an error message ("**Errors in the arrangement of objects. Unable to compose!**").

Check for intersections of **Imported objects** presented in the **Objects** folder (without attempts to compose **Subregions**), which is carried out by the **Check for self-intersections** command from the context menu of the **Objects** folder, when intersections are found, also the program displays an error message ("**There are issues in the arrangement of bodies.**").

In this case the **Objects > Issues** folder will be created and filled but intersecting surfaces will not be added into the **Composed subregions** folder.

8.1.8.3 The Project window, tab «Preprocessor»



The **Project** window, the **Preprocessor** tab

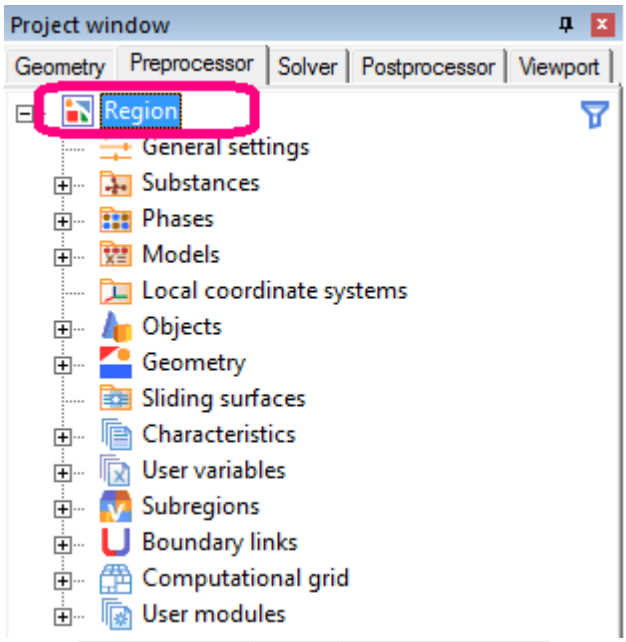
The **Preprocessor** tab displays a part of the project tree, in which main elements of the original project's data locate.

The **Preprocessor** tab of the project tree is the constructor of the source project's data. The elements, which have been created in the **Preprocessor** tab, are copied, with their parameters, into the **Postprocessor** tab of the project tree. Visualization in the **View** windowis tuned by parameters of the project tree elements displayed in the **Postprocessor** tab.

Contents of the **Preprocessor** tab is grouped into the following folders and elements:

- root folder **Region**
- element **General settings**
- folder **Substances**
- folder **Phases**
- folder **Models**
- folder **Local coordinate systems**
- folder **Objects**
- folder **Geometry**
- folder **Sliding surfaces**
- folder **Characteristics**
- folder **User variables**
- folder **Subregions**
- folder **Boundary links**
- folder **Computational grid**
- folder **User modules**

8.1.8.3.1 Root folder «Region»



The root folder **Region** in the project tree

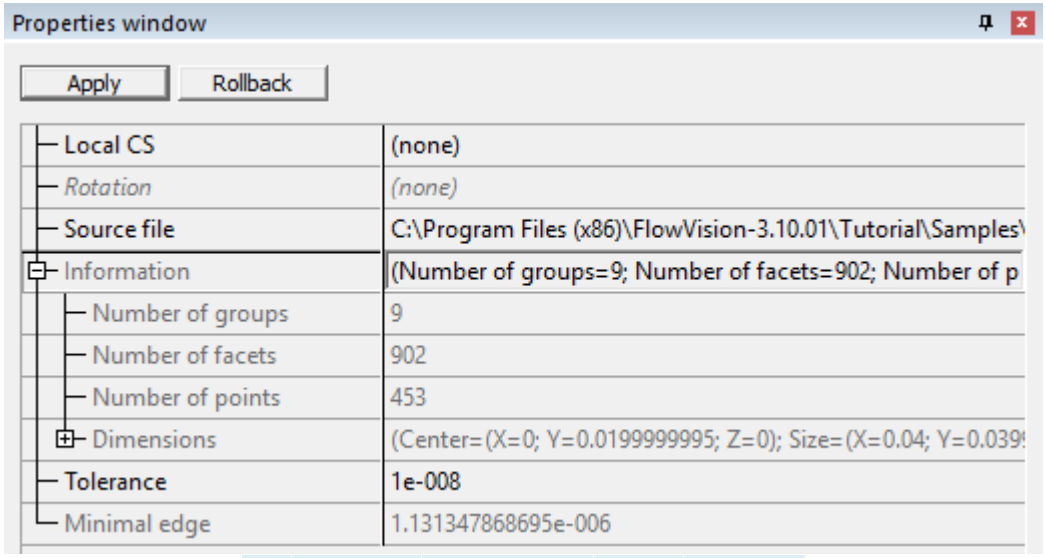
The root folder **Region** presents the geometry model of the computational domain, which:

- is created in a CAD-system and is imported into the project
- may be amended with surfaces of imported objects

When a new project is created from a geometry file (by the **File > Create** command), the **Region** folder is initially marked with the "**!**" symbol, which means that no **Model** is specified in each **Subregion**.

Parameters of the «Region» root folder

Parameters of the **Region** root folder are displayed in its **Properties** window.



The **Properties** window of the **Region** root folder

The **Properties** window of the **Region** folder:

Parameter	Description
Local CS	The choice of the local coordinate system. Drop-down list of defined elements **local coordinate system**.

Parameter	Description
	If you select **No** absolute coordinate system is used.
Rotation	Selection of rotation in a predetermined local coordinate system. Drop-down list of defined elements **Rotation**. Specifying rotation on the **Region** means "inclusion" of the centrifugal and Coriolis forces in the Navier-Stokes equations. In this case, the boundary conditions are set by default in the absolute coordinate system. For the correct formulation of the problem outside the boundary of the region should be a surface of revolution.
Source file	This is the source, from which the geometry of the computational domain has been loaded. Possible options are: • network path and name of the file, from which the geometry has been loaded • the "**From Geometry tab**" text, which means that the geometry has been prepared in the **Geometry** tab
Information > Number of groups	This is the number of groups of facets belonging the basic geometry model (the total number of groups presented in **SubRegion #N > Geometry** folders of all **Subregions**). This field is read-only and is not editable.
Information > Number of triangles	This is the number of facets belonging to the basic geometry model. This field is read-only and is not editable.
Information > Number of points	This is the number of nodes of facets that make up the surface of the basic geometry model. This field is read-only and is not editable.
Information > Dimensions > Center > X	These are coordinates of the center of the computational domain, [m]. These fields are read-only and are not editable.
Information > Dimensions > Center > Y	
Information > Dimensions > Center > Z	
Information > Dimensions > Size > X	These are sizes of the computational domain along axes X, Y, Z, [m]. These fields are read-only and are not editable.
Information > Dimensions > Size > Y	
Information > Dimensions > Size > Z	
Information > Dimensions > Min > X	These are minimum values of the coordinates of the surface of the basic geometry model in the absolute coordinate system (ACS), [m]. These fields are read-only and are not editable.
Information > Dimensions > Min > Y	
Information > Dimensions > Min > Z	
Information> Dimensions > Max > X	These are maximum values of the coordinates of the surface of the basic geometry model in the absolute coordinate system (ACS), [m]. These fields are read-only and are not editable.
Information > Dimensions > Max > Y	
Information > Dimensions > Max > Z	

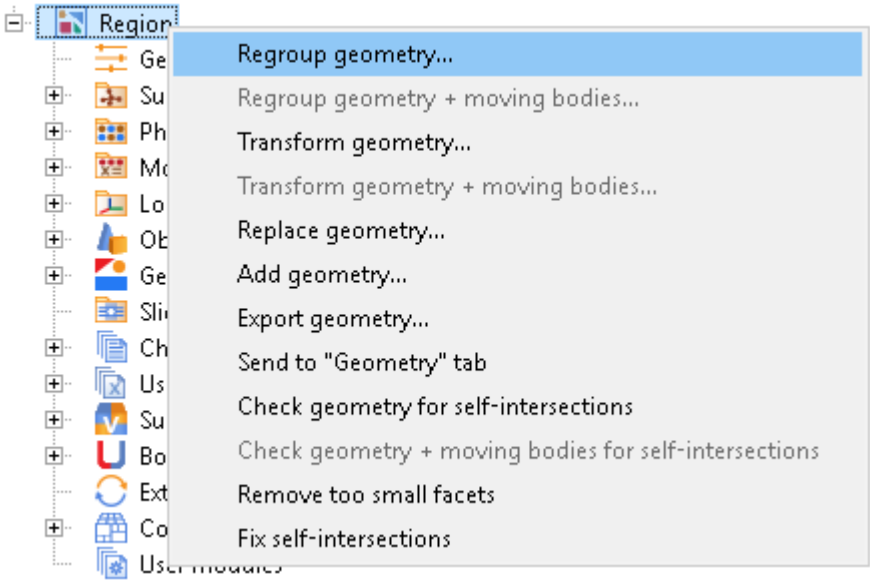
Parameter	Description
Tolerance	Geometric quantity ε, which characterizes the length of the maximum distance at which can be shifted vertex geometry in the construction of the computational grid. Values must satisfy the inequalities $\varepsilon > 10^{-8} d_{max}$ and $\varepsilon < d_{min}$ where d_{min} is the minimum value of the minimum size of an edge of the triangle and the minimum distance between the surfaces, d_{max} is the maximum value of the coordinates of the basic geometry model If the geometrical tolerance is set too small, this may lead to errors in the construction of the computational grid. If this value is set too high it can cause a self-intersection geometry on the solver. It is recommended that before starting the calculation to inspect the geometry on the self-intersection-adjusted solver (i.e., responding positively to the request **"Do you want to modify the geometry to the solver state before checking?"**, see section Checking a geometry model of computational domain and moving bodies for self-intersections) to make sure that the value of the geometric tolerance is not set too high. By default, this parameter is equal to the parameter **Geometry import > Tolerance**, which is set in the basic settings of Pre-Postprocessor.
Minimal edge	The minimum edge of the surface's facet, [m]. This field is read-only and is not editable.

Notes:

Specifying the LCS without specifying the rotation has no effect on the solution.

Specifying rotation for the **Region** is only a transition to a rotating coordinate system (in this case the boundary conditions are set by default in the absolute coordinate system).

Context menu of the «Region» root folder



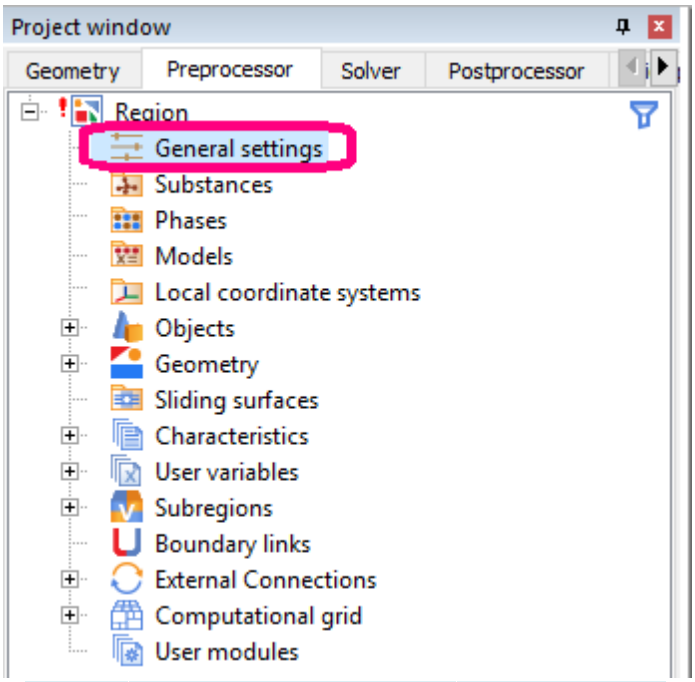
Context menu of the folder **Region**

The context menu of the folder **Region**:

Menu item	Description
Regroup geometry	Open the **Geometry regrouping** dialog box to change the composition of groups of facets geometry model of the

Menu item	Description
Regroup geometry + moving bodies	computational domain (and mobile cases, when you select "**...+ Moving bodies**"), see section Regrouping the facets.
Transform geometry **Transform geometry + moving bodies**	Open the **Geometry transformation** dialog box to transform the geometry model of the computational domain (and mobile cases, when you select "**...+ Moving bodies**"), see section Transformation of geometry model of the computational domain and an imported object.
Replace geometry	Open dialog box to import the object, replacing the geometry model of the computational domain (see section Geometry replacement).
Add geometry	Open dialog box to import the surface added to the computational domain (see the section Adding or removing Subregions in geometry models).
Export geometry	Open dialog box to export the file in the surface of a geometry model of the computational domain (see section Exporting a geometry model into a file).
Check geometry for self-intersections **Check geometry + moving bodies for self-intersections**	Check for self-intersection of the surface of a geometry model of the computational domain(and mobile cases, when you select "**...+ Moving bodies**"),see section Check for self-intersection of the surface of a geometry model of the computational domain and moving bodies.
Remove too small facets	Removing too small facets on the surface of a geometry model of the computational domain (see paras. Removal of too-small facets and Removal of too-small facets of geometry model of computational domain).
Fix self-intersections	Removal of self-intersecting facets on the surface of a geometry model of the computational domain (see sections Self-intersections of surfaces and their correction and Fixing self-intersections of a surface in a geometry model).

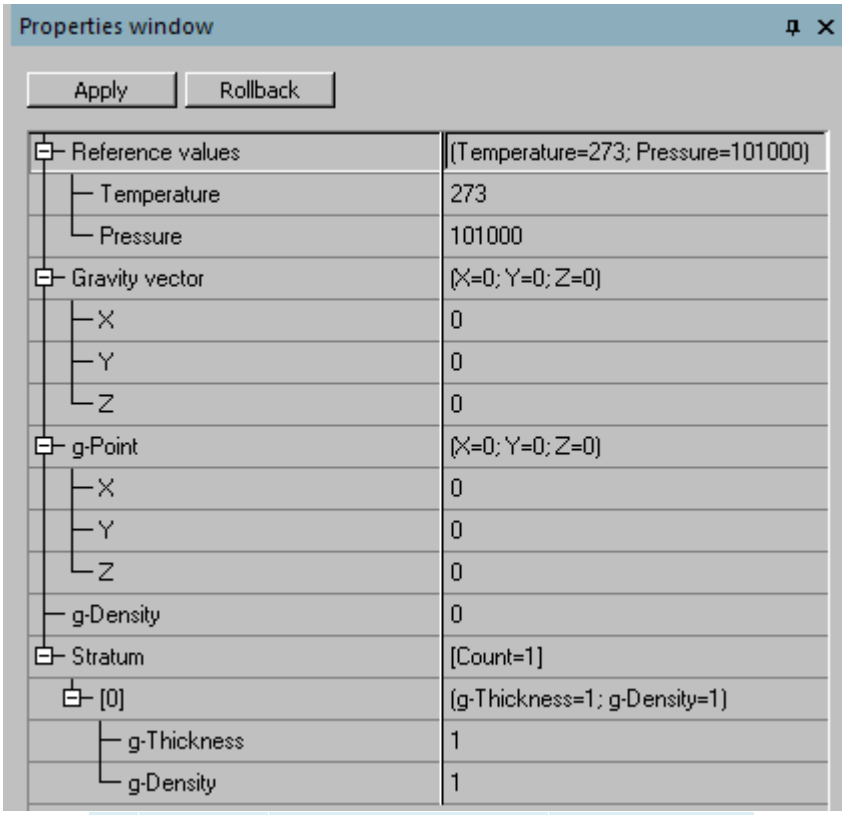
8.1.8.3.2 Element «General settings»



Element **Region > General settings** in the project tree

The **General settings** element contains parameters, which are common to the whole computational domain.

The **General settings** element has no context menu.



The **Properties** window of the element **General settings**

Parameters of the element **General settings**:

Parameter	Description	The dimension of
Reference values	The absolute values of the variables corresponding to zero values in the calculation and interface	
Reference values > Temperature	The value of the reference temperature T_{ref}. (see section Reference parameters, absolute and relative variables). The default value is **273** [K].	[K]
Reference values > Pressure	The value of the reference pressure P_{ref}. (see section Reference parameters, absolute and relative variables). The default value is **101000** [Pa].	[Pa]
Gravity vector > X **Gravity vector > Y** **Gravity vector > Z**	The projection of the acceleration of gravity ***(g)*** on the axis of the absolute coordinate system	[$m \cdot s^{-2}$]
g-Point > X **g-Point> Y** **g-Point> Z**	Point through which is perpendicular to the direction of the gravity vector passes a plane corresponding to a zero hydrostatic pressure. Coordinates**g-points**are given in the coordinate system of the computational domain.	[m]
g-Density	Hydrostatic fluid density, located above the zero hydrostatic level (if there are no other layers - and then below zero hydrostatic level)	[kg m^{-3}]
Stratum	List the layers of a heavy fluid, below the **g-point**, describes an array of hydrostatic layers located below the zero hydrostatic level.	

Parameter	Description	The dimension of
Stratum> [N] > g-Thickness	Layer parameters numbers N (countdown runs down from the **g-point**): ▪ **g-Thickness** is thickness of a heavy fluid; ▪ **g-Density** is density of a heavy fluid	[m]
Stratum> [N] > g-Density		[kg m^{-3}]

Notes:

1. The absolute values of the variables, for which a predetermined reference value is calculated by the formula $f_a=f+f_{ref}$, where
   f_a is the absolute value of the variable,
   f is calculated the relative value of the variable
   f_{ref} is the reference value.
2. Objectives of the reference values:
   Use familiar values
   Troubleshooting loss of accuracy in the calculation of variables varying weakly on the background of a large average.
3. All variables, unless otherwise specified, are defined and visualized in *relative* terms.
4. By default, the following values are selected:
   T_{ref}= 273 K (which means that as the temperature scale, the scale Celsius)
   P_{ref}= 101000 Pa (this means that the pressure is measured from about 1 atm).

It is recommended to set these values reference values, which are close to the average expected value. In many problems, such a task can significantly increase the accuracy of calculations.

8.1.8.3.3 Folder «Substances»

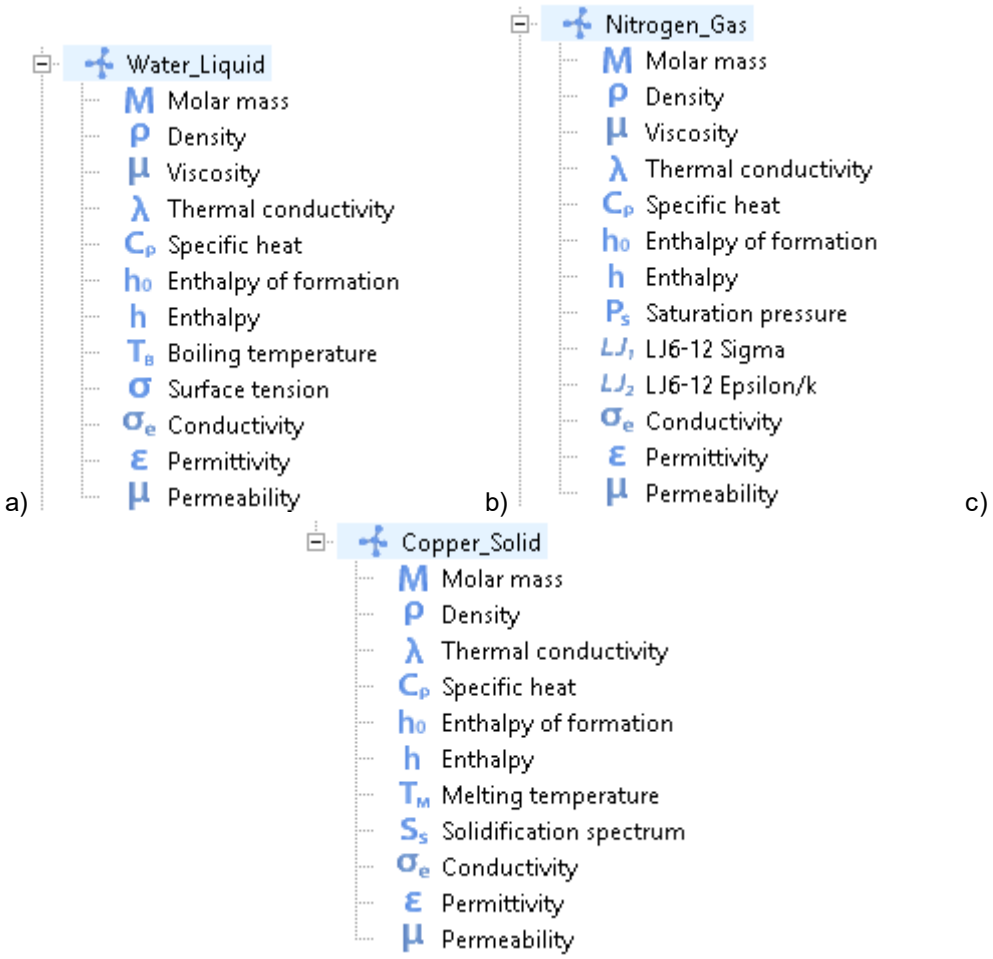
Folder **Substances** stores elements **Substance #N** that contain elements corresponding to **Physical properties**.

The context menu of the **Substances** folder contains the **Create** command only, which creates a new **Substance #N** element.

Element Substance

An element **Substance #N** is presented as a folder, located in the **Substances** folder.

The set of physical properties, which is available for a **Substance #N**, depends on the **Substance**'s **Aggregative state**.

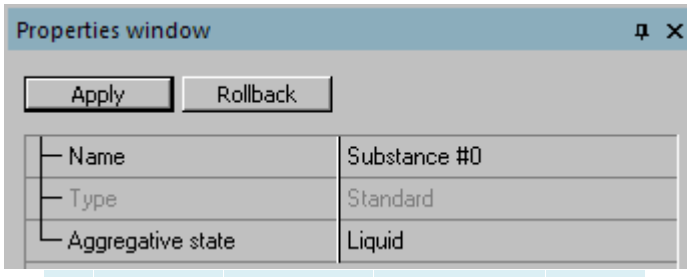


Physical properties of materials in different states of aggregation: a - liquid; b - gas; c - solid

Parameters of the «Substance #N» element

The **Substance** element has the following parameters:

- parameters in the **Properties** window of the folder itself
- parameters in the **Properties** windows of child elements "physical property"



The **Properties** window of a **Substance #N** element

Parameters from the Properties window of the **Substance** element:

Parameter	Description
Name	Name of the **Substance**.
Type	Possible values are: • **Standard**: for a substance, parameters of which are specified in **Preprocessor** • **From Substance Database**: for a substance that has been loaded from a database This field is informational only and cannot be changed by the user.

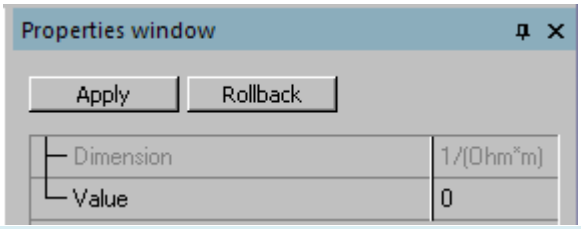
Parameter	Description
Aggregative state	Possible values are: • **Solid** • **Liquid** • **Gas** For the **Substances**, which have been loaded from the standard base, this field is informational only and cannot be changed by the user.

The properties of the substance are specified in the child elements "Physical property" (see subsection *"Elements corresponding to physical properties of a Substance"* below).

Elements corresponding to physical properties of a Substance

An element corresponding to physical properties of a **Substance** has two parameters in its **Properties** window:

- **Dimension**: this is a not editable parameter (the dimension of all physical quantities in *FlowVision* expressed in the *SI* system units);
- **Value**: this parameter is displayed and edited in the extended entry field and can be a numerical constant, given by the formula or function in a tabular form. The parameter **value** can be included in the number of external parameters.



The **Properties** window of an element corresponding to a physical property of a **Substance**

The **Value** parameter of a physical property, which is specified "from scratch", can only be specified as a constant and, for a substance in the **Gas** aggregate state, by the ideal gas law formula.

If you need to use in the calculations are not constants, but functions of pressure and/or temperature, you should:

- use substances, which has been defined in the standard substance database
- *and/or* create substances with required properties in your own user's **Substance Database**.

When specifying properties of a **Substance**, *absolute* **Temperature** and **Pressure** are used (see section Reference parameters, absolute and relative variables for details).

At the base substances property values may be a function of temperature and/or pressure, which is given in the form of a table. In calculating the values of the project is determined by linear interpolation between the nodes of the table (see details in the section Defining a Substance, its Phases and Properties).

A set of physical properties of a substance depends on the aggregative state of the substance. The user is able to set not all properties, and only those necessary for the solutions of the equations. If any required physical property will not be given, it will be marked with the "**!**" sign in the project tree.

In the project tree physical properties of the substance are displayed as child elements of the element **Substance #N** (the list of the physical properties depends on the aggregative state of the **Substance**):

Icons and names of physical properties of the Substance	Aggregative state of a Substance		
	Solid	Liquid	Gas
M **Molar mass** [5)]	Yes	Yes	Yes
ρ **Density** [*)]	Yes	Yes	Yes
λ **Thermal conductivity** [4)]	Yes	Yes	Yes
C_p **Specific heat** [5)]	Yes	Yes	Yes

Icons and names of physical properties of the Substance	Aggregative state of a Substance		
	Solid	Liquid	Gas
h^0 **Enthalpy of formation**	Yes	Yes	Yes
h **Enthalpy** (thermodynamic enthalpy)	Yes	Yes	Yes
T_M **Melting temperature**[**)]	Yes	No	No
S_s **Solidification spectrum**	Yes	No	No
μ **Viscosity**[***)]	No	Yes	Yes
T_B **Boiling temperature**[**)]	No	Yes	No
σ **Surface tension**	No	Yes	No
P_s **Saturation pressure**[**)]	No	No	Yes
LJ_1 **LJ6-12 Sigma**	No	No	Yes
LJ_2 **LJ6-12 Epsilon/k**	No	No	Yes
σ_e **Conductivity** (specific electrical conductivity)	Yes	Yes	Yes
ε **Permittivity** (relative dielectric capacity)	Yes	Yes	Yes
μ **Permeability** (relative magnetic permeability)	Yes	Yes	Yes

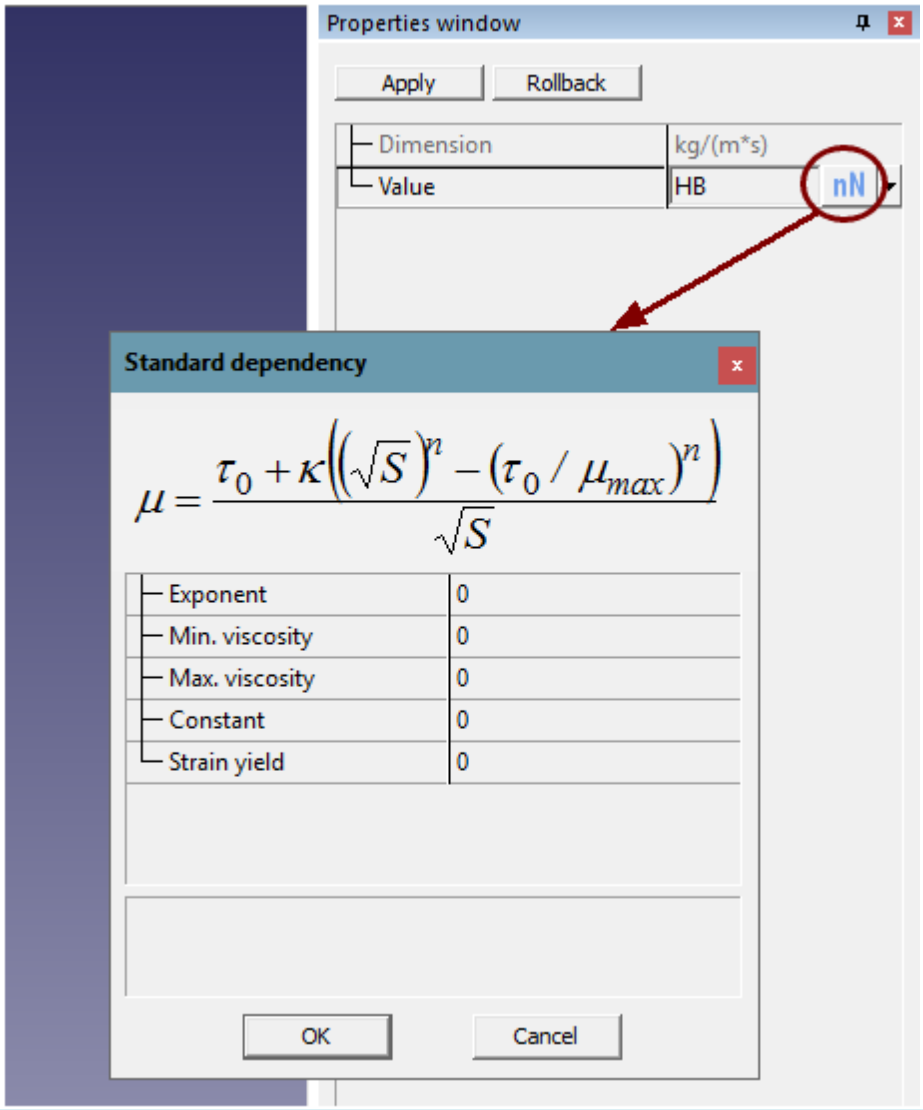
[*)] For liquid and gas **Substances**, if their **Density** is specified as a constant, it is possible to specify partial derivatives of **Density** with respect to **Pressure** and to **Temperature**.

[**)] These physical properties are specified by *absolute* **Temperature** and **Pressure** (see Reference parameters, absolute and relative variables).

[***)] Viscosity of liquid can be set by a constant, a formula, a table, or by models of non-Newtonian fluid **PL** and **PL2** (*Power Law* and *Power Law 2*), **HB** (the *Herschel-Bulkley* model), **BC** (the *Bird-Carreau* model). Viscosity of gas can be set by a constant, a formula, a table, a polynome, or by Blottner formula.

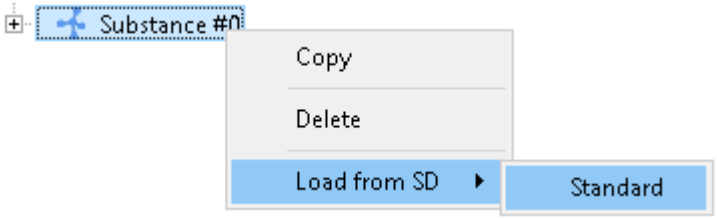
[4)] Thermal conductivity of a gas can be set by a constant, a formula, a table, by a polynome or by a $f(\mu, C_p)$ formula, see details in the section Theory> Substance properties > Gas.

[5)] If **Specific heat** is set by a polynome, the **Molar mass** must be set by a constant, else an error message will be output (**If Cp is specified by polynom, molar mass must be constant**).



Viscosity can be specified by the Herschel-Bulkley model of non-Newtonian liquid

Context menu of the element «Substance #N»

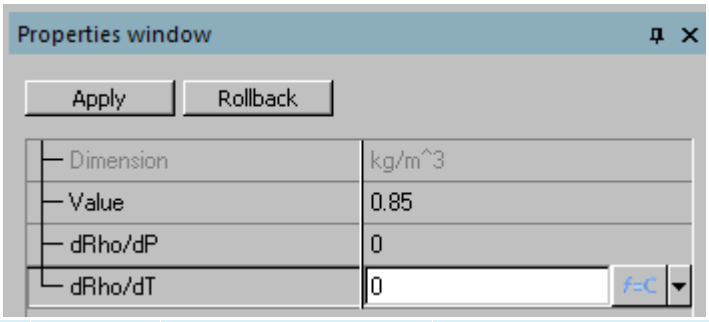


Context menu of the **Substance #N** element

Context menu of the **Substance #N** element in the project tree:

Menu item	Description
Copy	Creating a **Substance**, which is a copy of the selected item
Delete	Deleting the selected **Substance** item from the project tree
Load from SD > Standard	Opening the **Load from database** dialog box to select a **Substance** from the Substance Database.
Load from SD > *(name of a substance database)*	

Properties of the element «Substance #N > *(Physical property)*»



The **Properties** window of the element **Substance #N > Density**

Parameters of the element «Substance #N > (*Physical property*)»	
Parameter	**Description**
Dimension	Dimension of the physical property of the **Substance**. This field if for information only, it is not editable.
Value	A value of the physical property of the **Substance**; it can be specified, for example, by a constant, formula, table, of physical law.
dRho/dP	For liquid and gas **Substances**, if their **Density** is specified as a constant, it is possible to specify partial derivatives of **Density** with respect to **Pressure** and to **Temperature**. $\frac{\partial \rho}{\partial P}$ and $\frac{\partial \rho}{\partial T}$ These derivatives can be specified as constants, formulae or tables. By default these parameters have zero value, this corresponds to no compressibility. When the derivatives are specified, **Density** is calculated using the Taylor series of the first order:
dRho/dT	$\rho = \rho_{ref} + \frac{\partial \rho}{\partial P}(P_{abs} - P_{ref}) + \frac{\partial \rho}{\partial T}(T_{abs} - T_{ref})$ This method allows the program to approximate **Density** in a neighborhood of reference values of pressure P_{ref} and temperature T_{ref} if **Density** depends on **Pressure** and **Temperature** smoothly enough. When a more precise dependency of **Density** versus **Pressure** and **Temperature** is required, specify **Density** by a formula, table, or physical law. The derivatives are additive, i.e. the derivatives of a mix (a linear combination) of **Substances** are linear combinations of derivatives of appropriate component **Substances**.

Loading a Substance from the Substance Database

In order to load a **Substance** from a **Substance Database**, you have to select from the context menu of the **Substance** the command **Load from SD > *(name of the Substance Database)***. Then, in the **Load from database** dialog box, select the **Substance** and its **Phase**. After that, all values of the substance's property will be filled in accordance with the contents of the **Substance Database**. The user can change the value of the substance in the **Properties** window. In order to create a new **Substance** or change parameters of an existing **Substance** directly in a user **Substance Database**, apply Substance Database Editor.

Note:

By default, a user can load **Substances** from the standard **Substance Database** only. In order to be able to load **Substances** from a user **Substance Database**, you have to specify in the basic settings of **Pre-Postprocessor** the path to the directory of the user Substance Database.

Loading Substances from *CHEMKIN* files

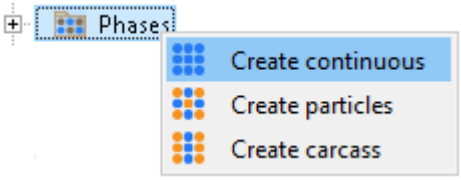
When chemistry is simulated, and chemical processes are imported from files in the CHEMKIN format, when the project contains **Substances** with the same names, properties of such **Substances**, which are set in the *CHEMKIN* file, will replace the properties, which are set in the project.

When **Substances** in project are replaced by **Substances** from the *CHEMKIN* file, the program requests you to confirm this ("**Replace properties of the existing substance ... ?**").

8.1.8.3.4 Folder «Phases»

The **Phases** folder contains subfolders **Phase #N**.

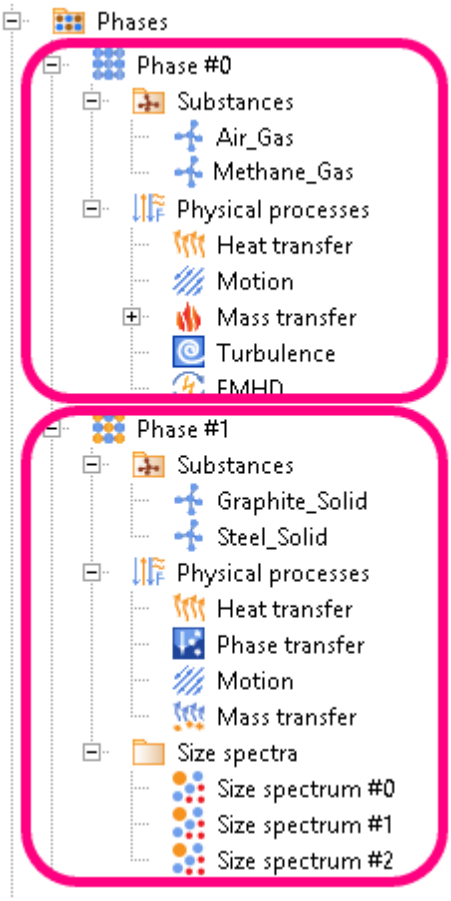
Context menu of the folder Phases



Context menu of the folder **Phases** contains the following commands:

Menu item	Description
Create continuous	Adding a new continuous **Phase #N** into the folder **Phases**
Create particles	Adding a new dispersed **Phase #N** of the **Particles** type into the folder **Phases**
Create carcass	Adding a new dispersed **Phase #N** of the **Carcass** type into the folder **Phases**

Folder «Phases > Phase #N»



Folders **Phase #N** in the project tree

Folder **Phases > Phase #N** contains:

- folder **Substances**
- folder **Physical processes** with child elements corresponding to various *physical process*.
- folder **Size spectra** with child elements **Size spectrum #N** (for dispersed **Phases** of the **Particles** type)

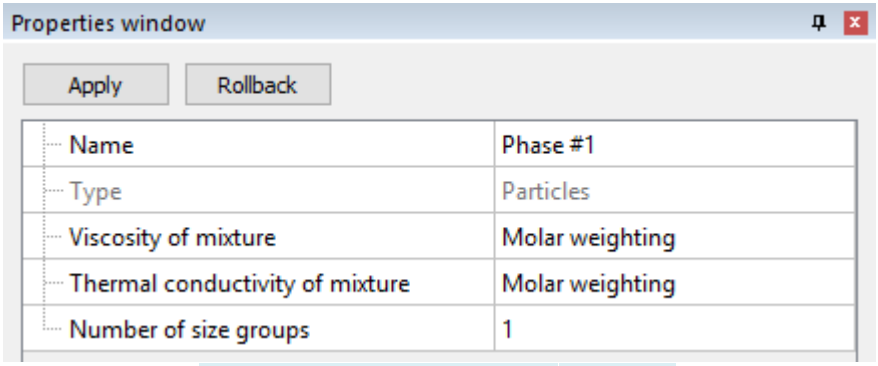
Presence of an element **Substance** in a **Phase** means that some properties of the **Substance** will be used during calculations for this **Phase**.

If no **Substance** is included in a **Phase**, then such a **Phase**, when it is loaded into a **Model**, is considered as vacuum. Loading such **Phase** into a computational model allows to calculate only the movement of boundary between continuous medium (gas or liquid) and vacuum.

Parameters of the element **Phase #N** are displayed in its **Properties** window.

If the folder **Phase #N > Substances** contains two or more **Substances**, then, in the **Properties** window of the element **Physical processes**, you have to select some of mass transfer models (**Mixing | Chemistry | Combustion**).

If the model of mass is not selected, this **Phase** in the project tree is marked with a symbol "**!**":



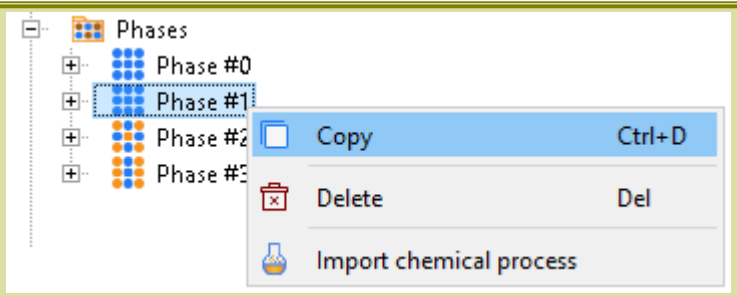
Properties window of the folder **Phase #N**

Parameters of the folder **Phase #N**:

Parameter	Description
Name	Name of the **Phase**
Type	Possible values are: • **Continuous** – a continuous phase that contains one or more **Substances** • **Particles** – a dispersed phase for simulating particles of one or more **Substances** • **Carcass** – a dispersed phase for simulating a carcass structure of one or more **Substances** This field is informational only, it is not editable.
Viscosity of mixture	Mixing rule for the dynamic coefficient of viscosity of the mixture. Possible options are: • **Mass weighting** – mixture viscosity is the sum of individual viscosities multiplied by mass fractions. • **Molar weighting** – mixture viscosity is the sum of individual viscosities multiplied by molar fractions • **Wilke** – use the Wilke formula for gas mixture
Thermal conductivity of mixture	Mixing rule for the coefficient of thermal conductivity of the mixture. Possible options are: • **Mass weighting** – thermal conductivity is the sum of individual thermal conductivities multiplied by the **Substance** mass fractions. • **Molar weighting** – thermal conductivity is the sum of individual thermal conductivities multiplied by the **Substance** molar fractions • **Mason-Saxena** – use the formula of Mason-Saxena for gas mixture

Parameter	Description
Number of size groups	This is number (an integer numerical value from **1** to **100**) of **Size groups** in **Size spectra** of particles. This parameter is available for dispersed **Phase** of the **Particles** type. Some **Variables** of the dispersed **Phase** are calculated individually for each of the size spectrum groups. Names of such variables include "**[N]**", which indicates the corresponding size group (**N=0**, **1**, ...). For example, concentrations of particles of different diameters will be presented by variables **Concentration [0]**, **Concentration [1]**, etc. instead of a single **Concentration** variable. This doesn't concern to variables that are common for all size groups of the **Phase** (for example, to variables **Molar mass** and **Density**). When **Mass transfer** *is enabled* in the dispersed **Phase**, you can use any existing **Size spectra** on a boundary condition and/or in initial conditions (the desired **Size spectra** is selected from a list in properties of the element **Diameter**). When **Mass transfer** *is disabled* in the dispersed **Phase**, the **Size spectrum #N** is used, which is set by the **Used drop size spectrum** parameter from properties of the element **Phase #N > Physical process > Phase transfer**. In this case elements **B. Cond. #N > Diameter** and **Init. data #N > Diameter** will be absent in the project tree. See details in subsection *Folder «Phases > Phase #N > Size spectra» and elements «Size spectrum #N»* below and in the section Spectra of particle sizes.

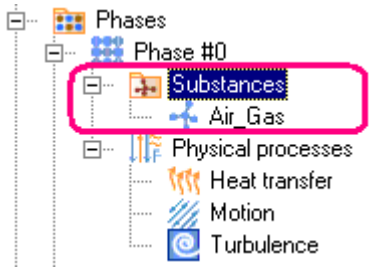
Context menu of the folder "Phase #N"



Context menu of the "Phase #N" folder

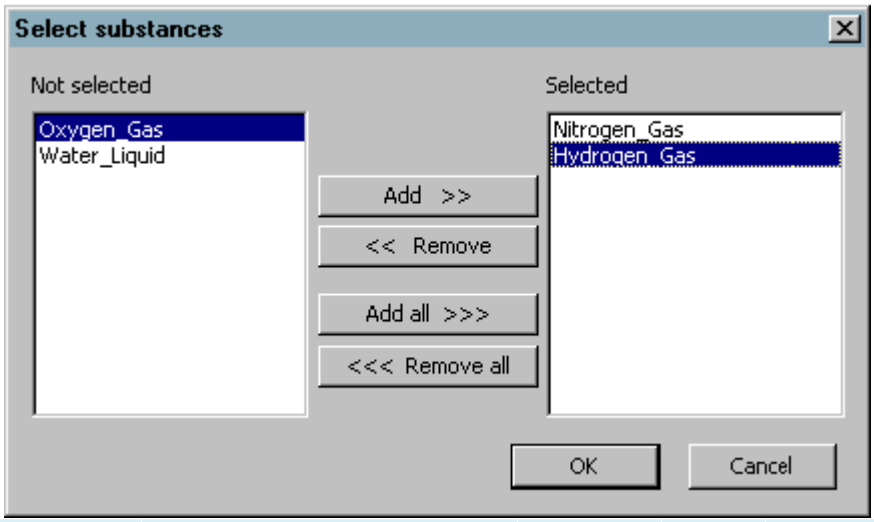
Menu item	Description
Copy	Creating an element, which is a copy of the selected element
Delete	Deleting the selected element from the project tree
Import chemical process	The **Import chemical process** dialog box will open, from which you can import a chemical process from files in the *CHEMKIN* format. As result of this import, the **Mass transfer** physical process of this **Phase** will be set as **Chemistry**, and **Substances**, which are reactants in chemical reactions, and their properties will be imported from the files. This command is available for continuous phases only. If the **Mass transfer** physical process is already set for the **Phase**, then attempt of importing chemical processes cause that the program requests you to confirm the import ("**Replace existing Mass-transfer process?**").

Folder «Phases > Phase #N > Substances»



The folder **Phases > Phase #N > Substances** contains child elements that correspond to **Substances**, which are simulated in **Phase #N** (the **Substances** themselves, which can be added into **Phases**, have to be specified beforehand in the folder **Substances**).

Context menus of the folder **Phases > Phase #N > Substances** and elements **Phases > Phase #N > Substances >** ***(substance)*** contain the **Add/remove** command, which opens the **Select substances** dialog box, which is used to specify the **Substances** presented in the **Phase**. You can select **Substances** among those that are specified in the folder **Substances**; click on the desired **Substances** in the **Not selected** pane and move them to the **Selected** pane by clicking the **Add** button. If you wish to delete a **Substance** from the list of the selected ones, select it there and click the **Remove** button.



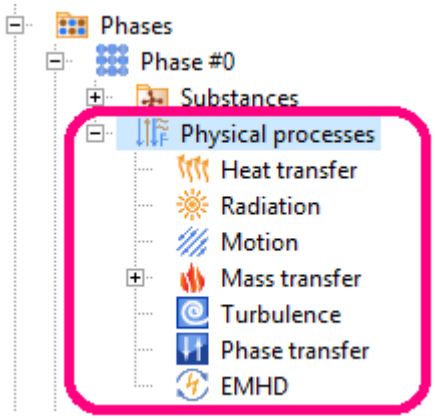
The **Select substances** dialog box allows you to specify **Substances** that are presented in a **Phase**

If a **Phase** contains several **Substances** and the **Mixing** model of mass transfer is used, it is strongly recommended to enter the **Substances** in the **Select substances** dialog box so that the **Substance** with the maximal amount in the simulated process would be placed on the last position in the **Selected** list list.

(when the **Chemistry** or **Combustion** model of mass transfer is used, it doesn't matter in which order the **Substances** are listed)

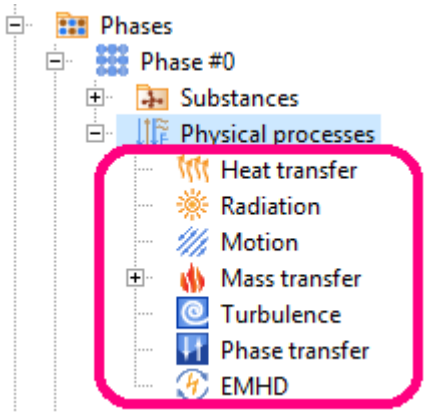
A **Phase** cannot contain more then 100 **Substances**.

Folder «Phases >Phase #N > Physical processes" and its elements



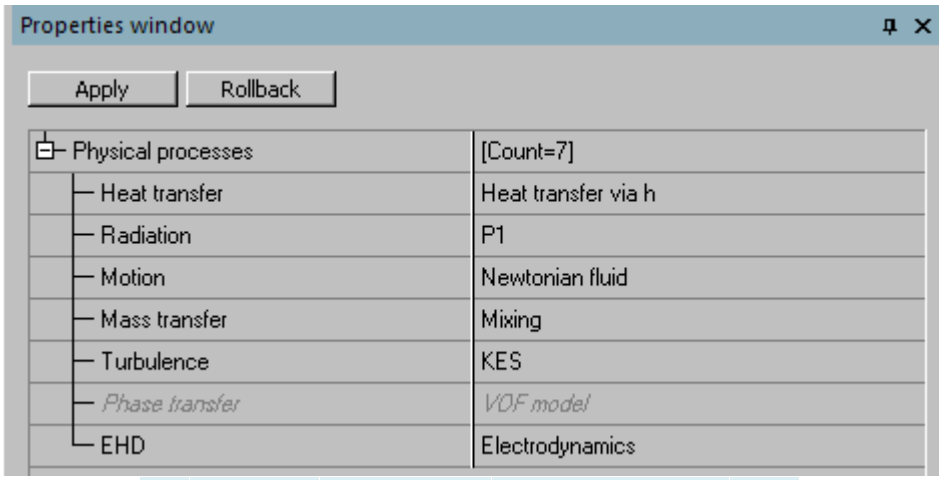
Folder **Phases > Phase #N > Physical processes** in the project tree

The folder **Phases > Phase #N > Physical processes** contains child elements that correspond to separate physical processes. These elements have different names and are referred in the documentation as **Physical process** elements.



Elements in the folder **Phases > Phase #N > Physical processes** correspond to the individual physical processes

The list of **Physical process** elements is determined by properties of the folder **Phases > Phase #N > Physical processes**.



The **Properties** window of the **Physical processes** folder

When a **Phase** has some **Physical process** element, this means that, when the project is calculated for this **Phase**, appropriate equations will be solved simulating the specified **Physical process**. If a **Phase** has no **Physical process** elements, you cannot use it to simulate any meaningful physically model.

Parameters of the **Physical processes** folder:

Parameter	Description
Heat transfer	The applied model of heat transfer in the form of convection and thermal conductivity. --- For *continuous* **Phases** the following values are possible: • **(none)**: Calculation without heat transfer in the form of convection and heat conduction • **Heat transfer via h**: Convective and diffusive energy transfer via the thermodynamic enthalpy. • **Heat transfer via H**: Convective and diffusive energy transfer via the total enthalpy. When flows of incompressible liquids and slow flows of gases ($M \le 0.1$) are simulated, it is recommended to use the simpler energy equation, which is formulated via the thermodynamic enthalpy (h). When flows of compressible fluids are simulated, particularly when solving problems with shock waves, it is recommended to use the energy equation formulated via the total enthalpy (H). Please note, that shock waves can appear and propagate in liquids after high local pressure surges. To simulate shock waves in liquid, you have to correctly specify the liquid's compressibility. See details in section Theory> Physical processes> Heat transfer > Equations. --- For *dispersed* **Phases** of the **Particles** type the following values are possible: • **(none)** - Calculation without heat transfer taken into account. • **Convection & conduction** - Taking into account the heat transfer. See details in section Processes in the presence of dispersed medium. --- For *dispersed* **Phases** of the **Carcass** type the following values are possible: • **(none)** - Calculation without heat transfer taken into account. • **Conduction** - Taking into account the heat transfer. See details in section Processes in the presence of dispersed medium.
Radiation	The applied model of the heat transfer by radiation. Possible options are: • **(none)**: Calculation is done with no heat transfer caused by radiation. • **Discrete-ordinates method** • **P1**: taking into account the heat transfer by radiation in the *P1* model (see section Theory> Physical processes> Radiation> P1) • **Optically thin layer**: taking into account the heat transfer by radiation in the approximation of optically thin layer (see section Theory> Physical processes> Radiation> Optically thin layer)
Motion	The applied model of the fluid's flow. For *continuous* **Phases** the following options are possible: • **(none)**: motion of the medium is not simulated • **Navier-Stokes model**: simulation of motion using the Navier-Stokes equations • **Darcy model**: simulation of flows in porous medium (see Motion). --- For *dispersed* **Phases** of the **Particles** type the following options are possible: • **(none)**: motion of the medium is not simulated • **Motion**: the program uses the motion model of dispersed phases (see Processes in the presence of dispersed medium)
Mass transfer	The applied model of mass transfer. --- For *continuous* **Phases** the following options are possible:

Parameter	Description
	• **(none)**: mass transfer is not simulated. • **Mixing** is for calculation, which takes into account the convection mass transfer. See section Theory > Physical processes > Mass transfer > Mixing > Equations. • **Chemistry** is for calculation, which takes into account the mass transfer in the form of chemically reacting components. See section Theory > Physical processes > Mass transfer > Chemistry > Equations. • **Combustion** is for calculation, which takes into account mass transfer in the form of chemically reacting components and combustion. See section Theory > Physical processes > Mass transfer > Combustion > Equations. • **Coal**: simulating the coal combustion. See section Theory > Physical processes > Processes in the presence of dispersed medium > Coal combustion. When chemical processes are imported from files in the CHEMKIN format, the existing **Mass transfer** physical process will be replaced. The program will request you to confirm this action ("**Replace existing Mass-transfer process?**"). For *dispersed* **Phases** of the **Particles** type the following values are possible: • **(none)**: mass transfer is not simulated • **Mass transfer**: the program uses the general mass transfer model for phases of the **Particles** type (see Processes in the presence of dispersed medium) • **Coal**: simulating the coal combustion. For *dispersed* **Phases** of the **Carcass** type the following values are possible: • **(none)**: mass transfer is not simulated • **Mass transfer**: the program uses the mass transfer model for phases of the **Carcass** type (see Processes in the presence of dispersed medium)
Turbulence	The applied model of the turbulent flow. Possible options are: • **(none)**: turbulent motion is not simulated • **SST**: turbulence is simulated using the SST (Shear Stress Transport) model. See Model SST. • **Sm**: turbulence is simulated using the algebraic Smagorinsky model. See Model Sm. • **SA**: turbulence is simulated using the SA (Spalart-Allmaras) model. See Model SA. • **KES**: turbulence is simulated using the standard k-ε model. See Model KES. • **KEAKN**: turbulence is simulated using the Low-Re k-ε model AKN (Abe-Kondoh-Nagano, model Abe - Kondoha - Nagano). See Model KEAKN. • **KEFV**: turbulence is simulated using the Low-Re k-ε model "FlowVision". See Model KEFV. • **KENL**: turbulence is simulated using the nonlinear k-ε model based on works of E.Baglietto. See Model KENL. Take into account compatibility between turbulence models and combustion models. Combustion models **Magnussen**, **Arrhenius-Magnussen** and **EDC** can only be used if a turbulence model is turned on (if no turbulence model is specified, the program would compute using the **Arrhenius** combustion model). The combustion models **Magnussen**, **Arrhenius-Magnussen** and **EDC** are only compatible with k-ε turbulence models (**KES**, **KEAKN**, **KEFV**, **KENL**) and the **SST** turbulence model (in the **SST** model ε is calculated from ω). To avoid possible problems with displaying Greek letters, the ε letter may be displayed as **e** in the program's user interface.
Phase Transfer	For *continuous* **Phases** this parameter informs you about either use of the VoF phase transfer model (when two continuous phases are presented in the model) or about absence of phase transfer (when only only one continuous phase is presented in the model). Possible options are: • **(none)**: movement of the phase interface surface between the two phases is not simulated in computation of the project

Parameter	Description
	• **VOF model**: the project is computed with calculation of movement of the phase interface surface between the two phases using the VoF method (see section Theory> Physical processes> Transfer phase> Equations) This field in informational only and cannot be edited here (it is filled by the program automatically depending on presence of one or two **Phases** in the computational model).
	For *dispersed* **Phases** of the **Particles** type the following options are possible: • **(none)**: phase transfer is not simulated in the project • **Convection & diffusion**: the project is calculated with taking phase transfer into account (see section Processes in the presence of dispersed medium) This field in informational only and cannot be edited here (it is filled by the program automatically depending on presence of one or two **Phases** in the computational model).
EMHD	• **(none)**: electrohydrodynamics is not simulated in the project • **Electrodynamics**: the project is calculated using the standard electrodynamic model • **MHD Potential model**: the project is calculated using the potential magnetohydrodynamics (MHD) model See details in the section Electromagnetohydrodynamics.
Crystallization	This parameter is used in *dispersed* **Phases** of the **Particles** type when icing is simulated. Possible options are: • **(none)**: icing is not simulated in the project • **Dry model**: drops, which fall the solid phase's (ice) surface, freeze immediately • **Film model**: subcooled waterdrops form a film on the solid surface. See details in the chapter Theory, in the section Crystallization.

Parameters of **Physical process** elements are individual to various physical processes, see their descriptions in subsections below and also in the Theory chapter.

1. Specifying physical processes is possible only after adding at least one **Substance** into the **Phase**.
2. Selection of the mass transfer model is possible only after adding at least two **Substances** into the **Phase**.
3. Selection of the turbulent transfer model is possible only when the **Movement** physical process is enabled.
4. Selection of the model of motion of the phase interface contact surface is possible only after a mathematical model of the **Movement** process is selected and the second **Phase** is added into the **Model**.
5. Selection of the radiation model is possible only after a **Heat transfer** model is selected.

See detailed descriptions of the mathematical models of physical processes in the chapter Theory.

Parameters of the 'physical process' elements, which are common to all physical processes

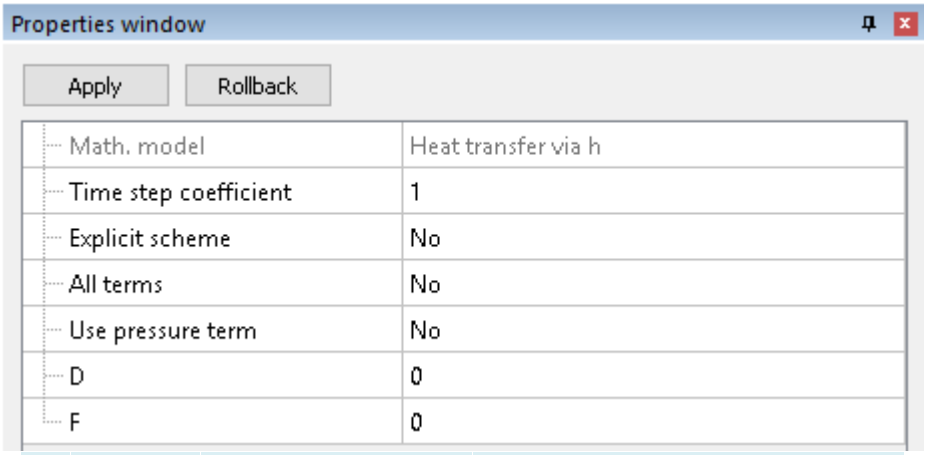
Parameter	Description
Math. model	Name of the mathematical model. This field is informational only and is not editable (its contents is determined by the value of the appropriate parameter of the element **Phase #N > Physical processes**).
Time step coefficient	This is a coefficient that is equal to the ratio of the time step, which is used to calculate the physical process, to the common time step of the whole project (τ). Possible values are: • value **1** means that computation of this physical process is carried out *at the same pace* as computation of the whole project is done • value greater than**1** means *accelerated* computation of this physical process in comparison with computation of the whole project • value in the range (**0** ... **1**) means *decelerated* computation of this physical process in comparison with computation of the whole project • value less than **0** means stopping computation of this physical process i.e. "freezing" (without resetting to zero) the field of the physical variable (traditionally the "**-1**" value is used for this purpose).

Parameter	Description
	Thus: Own time step of a physical process = τ x **Time step coefficient** ***Note:*** for the **Phase Transfer** physical process for *continuous* **Phases**, this parameter is not available for editing. Use the **Advanced settings** > **Multiphase C** > **Relaxation** parameter (in the **Solver** tab) instead.

For brevity's sake, we don't provide duplicated descriptions parameters for specific physical processes.

Parameters of the physical process "Heat transfer"

The **Physical processes >Heat transfer** element in the project tree



The **Properties** window of the element **Physical processes > Heat transfer**

Parameters of the element **Physical processes > Heat transfer**:

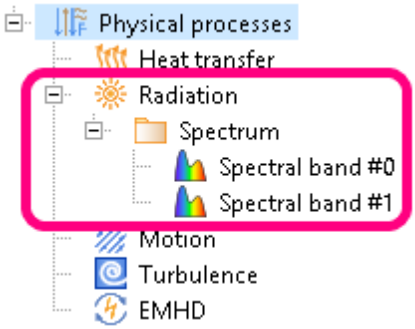
Parameter*)	Description
Math. model **Time step coefficient**	*See table "Parameters of the 'physical process' elements, which are common to all physical processes"*
All terms	Taking into account the additional terms of the heat transfer equation. Possible options are: • **Yes**: the heat transfer equation is calculated with the term that describes the heat generation due to viscous dissipation. • **No**: calculation of the heat transfer equation is performed without this term.
Use pressure term	Taking into account the pressure term in the energy equation. Possible options are: **No** | **Yes**. When **Use pressure term = No**, the following terms are removed from the the energy equation: • If the **Heat transfer via h** mathematical model is used, terms $\partial P / \partial t$ and $V \cdot \nabla P$ are removed from the equation (HeatTransfer.4). • If the **Heat transfer via H** mathematical model is used, term $\partial P / \partial t$ is removed from the equation (HeatTransfer.9). In most of the cases it is recommended to specify **Use pressure term = Yes** after several time steps after beginning the computation.

Parameter*)	Description
	The default value is: • **No**: if, at creation of the **Phase**, the **Heat transfer** physical process was specified as **Heat transfer via h** • **Yes**: if, at creation of the **Phase**, the **Heat transfer** physical process was specified as **Heat transfer via H**
D	Expression for D in the source term D * TEMP + F
F	Expression for F in the source term D * TEMP + F

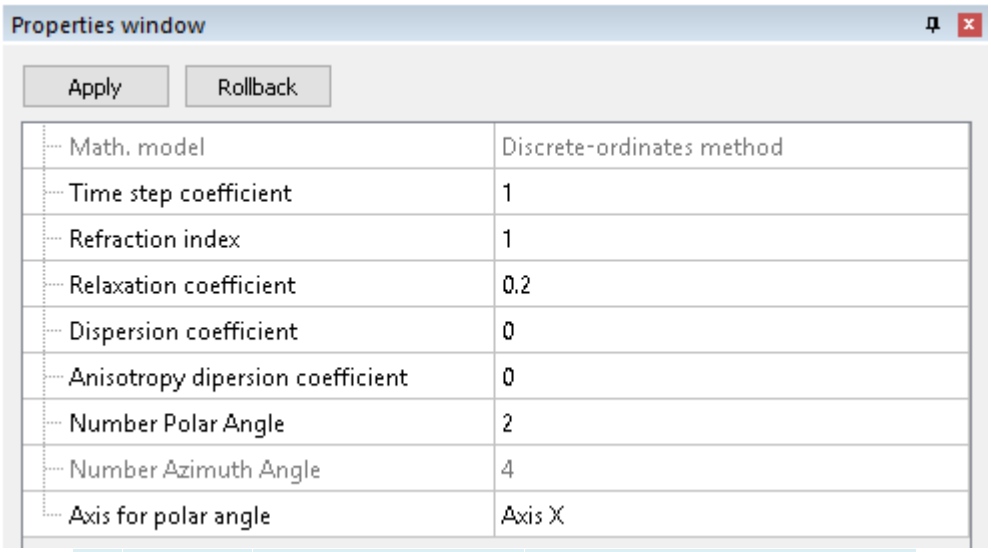
*) Only parameters **D** and **F** are specified for dispersed (**Particles** and **Carcass**) phases.

Note that when the mathematical model of heat transfer (**Heat transfer via h** or **Heat transfer via H**) is changed, the program keeps existing properties of the **Physical processes > Heat transfer** element and these properties can differ from the default properties of the new heat transfer model.

Parameters of the physical process "Radiation"



The **Physical processes > Radiation** element in the project tree (the **Discrete-ordinates method** radiation model is applied)



The **Properties** window of the element **Physical processes > Radiation**

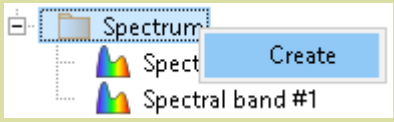
Parameters of the element **Physical processes > Radiation**:

Parameter	Description
Math. model	*See table "Parameters of the 'physical process' elements, which are common to all physical processes"*
Time step coefficient	
Refraction index	The refraction coefficient n of the medium

Parameter	Description
Absorption coefficient	The absorption coefficient of the medium. This parameter is not available for the **Discrete-ordinates method** radiation model (**Absorption coefficient** for this model is set individually for each **Spectral band #N**, see below).
Relaxation coefficient	The weighting factor, which is used to damper the term $divQ$, given by the radiation model, before substituting into the heat transfer equation at the current time step
Dispersion coefficient	Parameters of the **Discrete-ordinates method** radiation model. See details in sections Theory > Physical processes > Radiation > Parameters and Discrete-ordinates method.
Anisotropy dispersion coefficient	
Number Polar Angle	
Number Azimuth Angle	
Axis for polar angle	

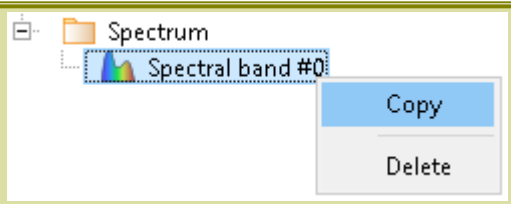
See details in the section Radiation.

When the **Discrete-ordinates method** radiation model is used, the **Radiation** element in the project tree has subfolder **Spectrum**, which can contain child elements **Spectral band #N**. The **Spectrum** subfolder has no parameters in its **Properties** window.



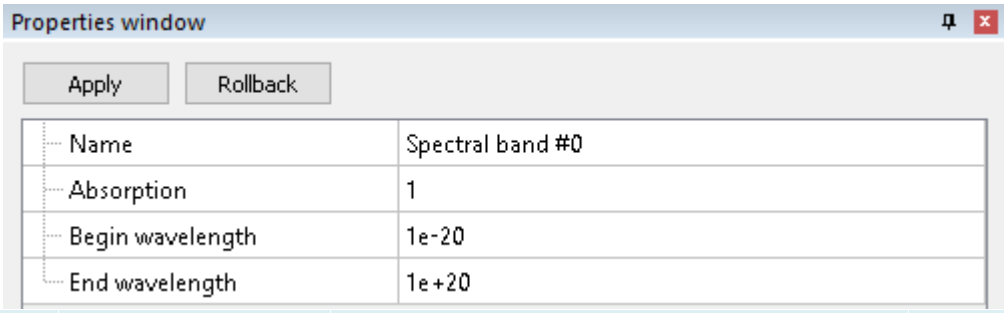
Context menu of the "Spectrum" folder

Menu item	Description
Create	Create a new **Spectral band #N** element in this folder



Context menu of the "Spectral band #N" element

Menu item	Description
Copy	Creating an element, which is a copy of the selected element
Delete	Deleting the selected element from the project tree



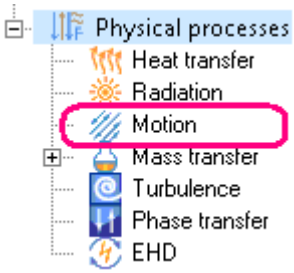
The **Properties** window of the element **Physical processes > Radiation > Spectrum > Spectral band #N**

Parameters of the element **Physical processes > Radiation > Spectrum > Spectral band #N**:

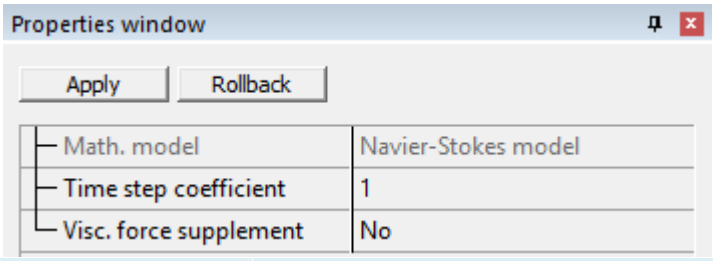
Parameter	Description
Name	Name of the spectral band

Parameter	Description
Absorption	Absorption coefficient, [1/m]
Begin wavelength	Beginning of the spectral band, [m]
End wavelength	End of the spectral band, [m]

Parameters of the physical process "Motion"



The **Physical processes > Motion** element in the project tree



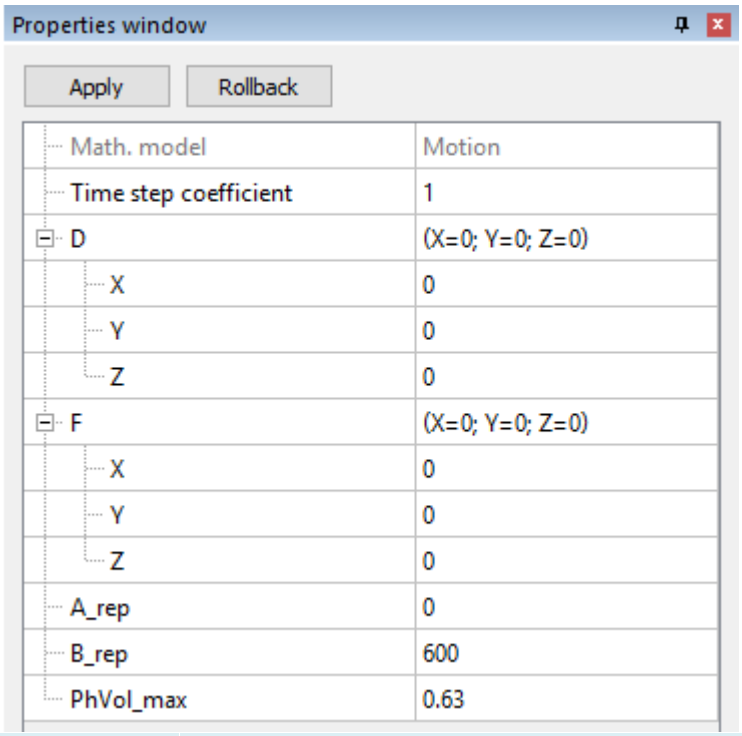
The **Properties** window of the element **Physical processes > Motion (for continuous Phases)**

Parameters of the element **Physical processes > Motion** (for continuous **Phases**):

Parameter	Description
Math. model **Time step coefficient**	*See table "Parameters of the 'physical process' elements, which are common to all physical processes"*
Visc. force supplement	This parameter tunes adding a supplement to viscous forces (which act on the fluid), due to spatial heterogeneity of viscosity and density. Possible options are: • **No**: solution of the Navier-Stokes equations is carried out with a homogenous viscosity factor • **Yes**: solution of the Navier-Stokes equations with terms that take into account heterogeneity of the viscosity factor This parameter specifies if the full equation for forces, caused by viscosity, will be calculated. By default **Visc. force supplement = No** and the simplified equation is solved to calculate viscosity forces. In most cases this simplification gives no substantial inaccuracy. You don't have to enable **Visc. force supplement** when: • no turbulence model is used or gradients of turbulent viscosity are not large • the molecular viscosity has no large spatial gradients You should enable **Visc. force supplement** when: • there are large gradients of viscosity (for example, when non-Newtonian fluids are simulated) • values of viscosity are large • some turbulence model is used (because a turbulent flow has heterogeneous field of viscosity)

Parameter	Description
	The only way to ensure that you don't have to enable **Visc. force supplement** is comparison of two calculations, one with enabled **Visc. force supplement** and the other with disabled **Visc. force supplement**, on a converged grid. By default **Visc. force supplement** is disabled because solving a simulation with a full equation of viscosity might cause excessive instability. In this case, if gradients are large, divergence of the computation can occur. This means that if the initial approximation is coarse and gradients of physical variables are large the computation is unstable. So it is recommended to start the computation without **Visc. force supplement** and then, when the solution converges, you can refine the results with **Visc. force supplement** enabled. Do not start the computation from scratch with **Visc. force supplement** enabled. If turning **Visc. force supplement** on causes divergence of the computation, even after a preliminary computation, try to reduce the time step.

See also: section Theory > Physical processes > Motion > Parameters.



The **Properties** window of the element **Physical processes > Motion (for dispersed Phases of the Particles type)**

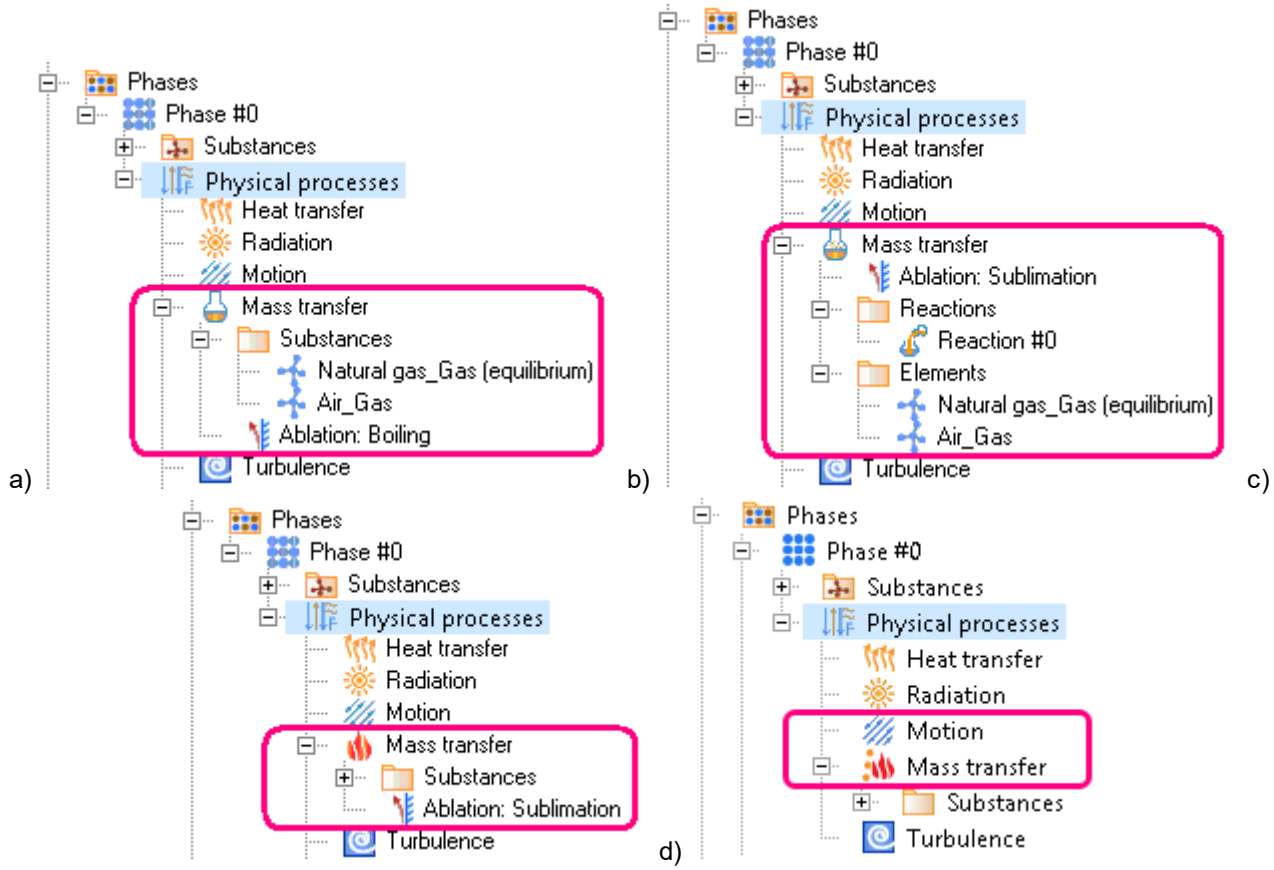
Parameters of the element **Physical processes > Motion** (for dispersed **Phases** of the **Particles** type):

Parameter	Description
Math. model **Time step coefficient**	*See table "Parameters of the 'physical process' elements, which are common to all physical processes"*
D > X **D > Y** **D > Z** **F > X** **F > Y** **F > Z**	A user-defined force that acts on particles. The force is is specified by parameters D and F in the source term **D*TEMP_D+F**.

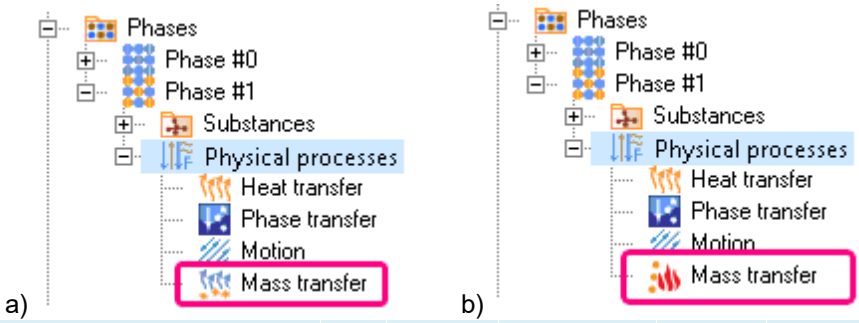
Parameter	Description
A_rep	Coefficients in the implemented model of particles repulsion - see Eq. (DispParticles.4).
B_rep	
PhVol_max	Maximum permissible fraction of the cell volume occupied by the **Dispersed Phase** - see Eq. (DispParticles.4).

See also: section Theory > Physical processes > Processes in the presence of dispersed medium > Parameters > Process 'Motion'.

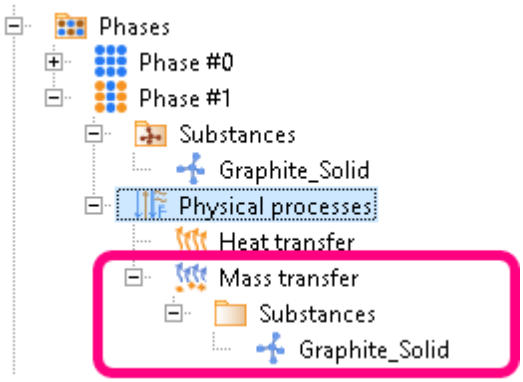
Parameters of the physical process "Mass transfer"



The **Physical processes > Mass transfer** element for a continuous phase in the project tree:
a) the **Mixing** model; b) the **Chemistry** model; c) the **Combustion** model; d) the **Coal** model



Element **Physical processes > Mass transfer** for a dispersed phase of the **Particles** type in the project tree (the icon depends on the model of **Mass transfer**):
a) the general mass transfer model; b) the **Coal** model



Element **Physical processes > Mass transfer** for a dispersed phase of the **Carcass** type in the project tree

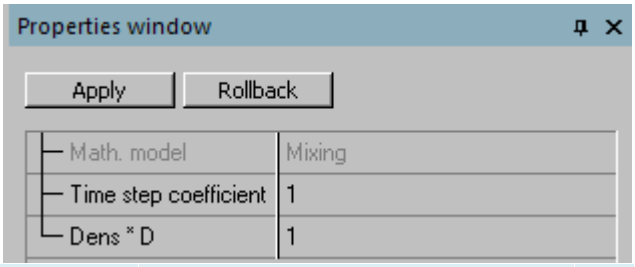
Parameters of the mass transfer parameters are specified in the **Properties** window of the element **Physical processes > Mass transfer**, the set of the parameters depends on the used mathematical model and if the **Motion** physical process is enabled.

Properties window

Apply | Rollback

Math. model	Combustion
Time step coefficient	1
Schmidt	1
Explicit scheme	No
Ablation	Sublimation
Combustion model	Zeldovich
i_1	17.167
i_2	18.167
i_3	0
T ignition	0
Alpha min.	0
Alpha max.	10000000000
New combustion limits	No
A	10000000000
n	0
B	18400
n_f	1
n_o	1
C	23.6
Fuel	Natural gas_Gas (equilibrium)
Oxidizer	Air_Gas
Product-1	Carbon dioxide_Gas
Product-2	Carbon monoxide_Gas

Example of the **Properties** window of the element **Physical processes > Mass transfer**, the **Combustion** model is applied in *continuous* phases, the **Motion** physical process is enabled



The **Properties** window of the element **Physical processes > Mass transfer**, the **Mixing** model is applied in *continuous* phases, the **Motion** physical process is disabled

Parameters of the "Physical processes > Mass transfer" element (for continuous phases)	
Parameter	**Description**
Math. model **Time step coefficient**	*See table "Parameters of the 'physical process' elements, which are common to all physical processes".* The **Math. model** parameter can be: **Mixing | Chemistry | Combustion | Coal**. (the value is determined by the value of the **Mass transfer** parameter of the element **Phase #N > Physical processes**)
Dens * D	This is ρD, the product of density ρ and the molecular diffusion coefficient D. This parameter is only available for the mass transfer models **Mixing** and **Combustion** when the physical process **Motion** is *disabled*.
Schmidt *or* **Schmidt > ...**	The molecular Schmidt number of the medium (Sc). For the **Chemistry** mass transfer model an array is specified with Schmidt numbers for each of the **Substances** from the **Phase #N**. For the **Mixing** and **Combustion** mass transfer models, when the physical process **Motion** is *disabled*, this parameter is absent (in such cases use the **Dens * D** parameter, see above).
Explicit scheme	Solution of the equations of mass transfer with use of the **Explicit** computational scheme. Possible options are: **Yes | No**. For the **Mixing** model the explicit scheme is recommended for modeling of radiative decay predecessors. For the **Combustion** model it is recommended not to change the default value, which is **No**.
Ablation	Selection of the ablation model. Possible options are: **(none) | Carcass | Chemistry | Boiling | Sublimation**. When an ablation model is enabled, a subfolder appears in the project tree, **Physical processes > Mass transfer > Ablation: *Model of ablation***, see subsection *"Subfolder «Mass transfer > Ablation: Model of ablation»"* below. An ablation model can only be selected if a **Heat transfer** is selected in physical processes of the **Phase**.
Combustion model[*)]	Selection of the combustion model. Possible options are: **Zeldovich | Arrhenius | Magnussen | Arrhenius-Magnussen | EDC**.

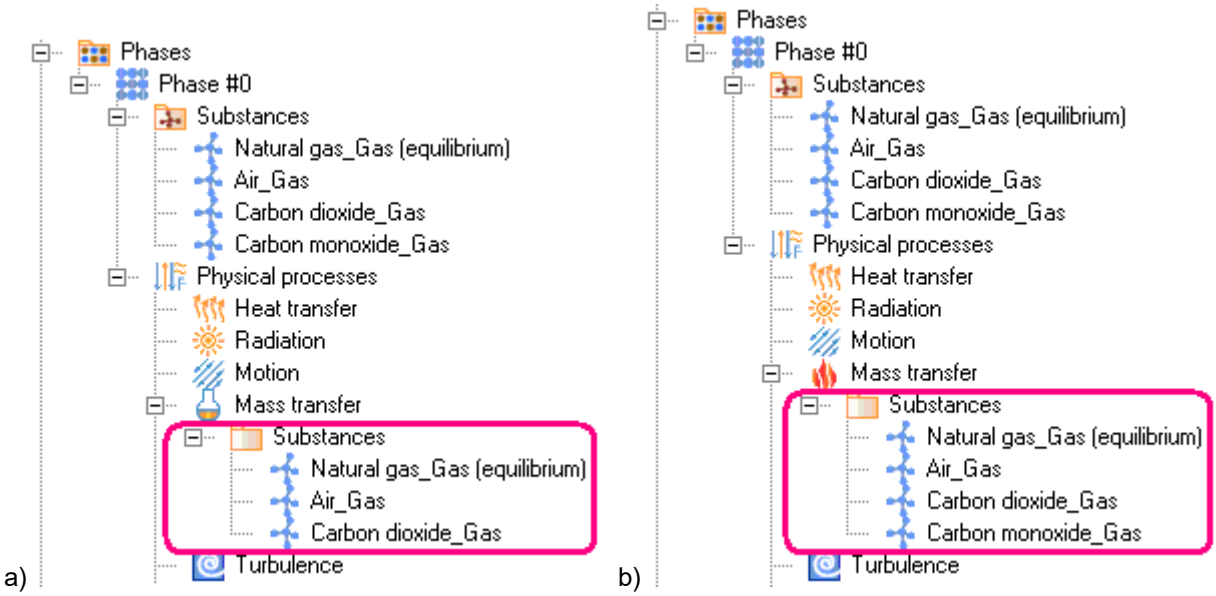
Parameters of the "Physical processes > Mass transfer" element (for continuous phases)	
Parameter	**Description**
	Combustion models **Magnussen**, **Arrhenius-Magnussen** and **EDC** can only be used if a turbulence model is turned on (if no turbulence model is specified, the program would compute using the **Arrhenius** combustion model). The combustion models **Magnussen**, **Arrhenius-Magnussen** and **EDC** are only compatible with k-ε turbulence models (**KES**, **KEAKN**, **KEFV**, **KENL**) and the **SST** turbulence model (in the **SST** model ε is calculated from ω). Any other turbulence models cannot be used with these combustion models because computation results would be substantially incorrect.
i_1 *) **i_2** *) **i_3** *)	Stoichiometric coefficients of the chemical reaction: • i_1 - stoichiometric coefficient of the **Oxidant**. • i_2 - stoichiometric coefficient of **Product 1**. • i_3 - stoichiometric coefficient of **Product 2**. The coefficient i_3 may be equal to **0**.
T ignition *)	Inflammation temperature (point of ignition), [K]
Alpha min. *) **Alpha max.** *)	Lower and upper limits of combustion in the **oxidant** excess coefficient,the limit values of the reduced coefficient of excess oxidant α^*_{min}, α^*_{max} to indicate the limits of combustion
New combustion limits *)	Activation of new limits instead of burning **Alpha min.** and **Alpha max.** Possible options are: • **No** - use the "old" limits of combustion • **Yes** - use the "new" limits of combustion
T lean *)	The temperature limit of lean combustion mixture, calculated on the assumption that all the fuel burned (Available only if the **New combustion limits = Yes**)
T rich *)	The temperature limit of rich combustion mixture, calculated on the assumption that all the oxidant consumed (Available only if the **New combustion limits = Yes**)
A *) **n** *) **B** *) **n_f** *) **n_o** *)	Rate parameters of the combustion reaction in the **Arrhenius** model (parameters of the kinetic reaction) occurring in the formula: $$W_f = W_{kin} = A\,T_{abs}^{n} e^{-B/T_{abs}} (\rho Y_f)^{n\_f} (\rho Y_o)^{n\_o}$$ (See section Theory > Physical processes > Mass transfer > Combustion> Equations)
C *)	Rate parameter of the combustion reaction in the model **Magnussen** (turbulent reaction), included in the formula: $$W_f = W_{turb} = C \left(\frac{\mu\varepsilon}{\rho k^2}\right)^{0.25} \rho \frac{\varepsilon}{k} \min\left(Y_f, \frac{Y_o}{i_1}\right)$$ (See section Theory > Physical processes > Mass transfer> Combustion> Equations)
Fuel *)	The **Substance**, which is fuel (selected from the list)
Oxidizer *)	The **Substance**, which is oxidizer (selected from the list)
Product-1 *) **Product-2** *)	The **Substances**, which are combustion products (selected from the list)

*) These parameters are only available only when standard or coal combustion is simulated (**Phases > Phase #N > Physical processes > Mass transfer = Combustion | Coal**).

Parameters of the "Physical process > Mass transfer" element (for dispersed phases of the Particles type)	
Parameter	**Description**
Math. model **Time step coefficient**	*See table "Parameters of the 'physical process' elements, which are common to all physical processes".* The **Math. model** parameter can have the following values: • **Mass transfer**: the general mass transfer model for dispersed phases • **Coal**: the coal combustion mass transfer model (the value is determined by the value of the **Mass transfer** parameter of the element **Phase #N > Physical processes**)
D **F**	A user-defined source of mass. The source is is specified by parameters D and F in the source term **D*TEMP_D+F**.
Model for particles	Model of transforming particles. Possible options are: • **Variable diameter** - density of a particle remains constant while the diameter changes according to the mass loss of the **Particles Phase**. • **Constant diameter** - diameter of a particle remains constant while the density changes according to the mass loss of the **Particles Phase**. For Euler (non-coal) particles: • The **Variable diameter** model simulates natural processes of evaporation/sublimation of liquid or solid substance and vapor condensation on surfaces of particles. Density of the liquid or solid substance is set in properties of the **Substance** (the density can be set either by a constant or a formula or a table). • The **Constant diameter** model assumes permanency of sizes of particles (so sizes of particles that were specified in initial and on boundary conditions will not change during the simulation). Densities of particles will be calculated by their masses taking into account that diameters are known. Density of the dispersed **Phase** in this model corresponds to a porous particle filled with liquid, which can transfer to the continuous **Phase** (evaporate) or come from continuous **Phase** (condense on surfaces of the particles).
Parameters for the coal combustion model	These parameters are used when the coal combustion model is applied (**Math. model = Coal**). See description in section Theory> Physical processes > Processes in the presence of dispersed medium > Coal combustion > Parameters.

Parameters of the "Physical process > Mass transfer" element (for dispersed phases of the Carcass type)	
Parameter	**Description**
Math. model **Time step coefficient**	*See table "Parameters of the 'physical process' elements, which are common to all physical processes".* The **Math. model** parameter is determined by the value of the **Mass transfer** parameter of the element **Phase #N > Physical processes**.
See also subsection *"Child elements of the subfolder «Physical processes > Mass transfer» for dispersed Phases of the Carcass type"*. Some settings for **Phases** of the **Carcass** type (blackness of the carcass, coefficient of heat exchange with the continuous **Phase**) are specified in properties of the element **Phase interaction > Continuum-carcass**.	

Subfolder «Physical processes > Mass transfer > Substances» for mass transfer models "Mixing" and "Combustion" in continuous phases

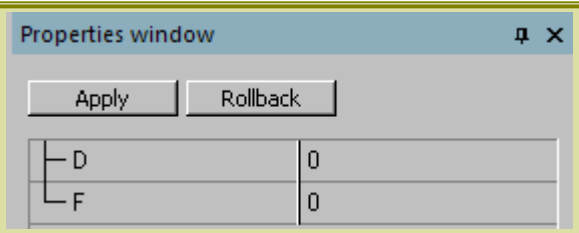


Subfolder **Physical processes > Mass transfer > Substances**
a) the **Mixing** model; b) the **Combustion** model

When Mixing or Combustion mass transfer model is selected for a continuous*) phase, the **Substances** subfolder appears in the **Mass transfer** element; this subfolder contains child elements ***Mass transfer substance*** that correspond to the following **Substances**:

- for the Mixing mass transfer model: all **Substances** from the folder **Phases > Phase #N > Substances**, except the **Substance**, which is listed there as the last one.
- for the Combustion mass transfer model: all **Substances** from the folder **Phases > Phase #N > Substances**.

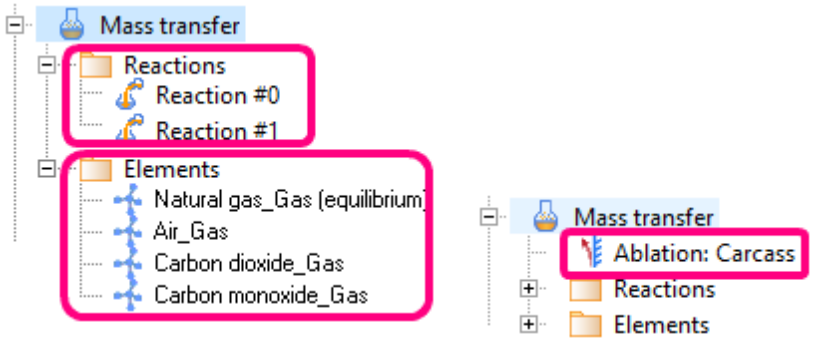
*) In a *dispersed* **Phase**, the element **Physical processes** > Mass transfer has no child elements ***Mass transfer substance***.



Parameters of the "Physical processes > Mass transfer > Mass transfer substance" element for the mass transfer models "Mixing" and "Combustion" in continuous phases

Parameter	Description
D	The coefficient of the mass fraction of **Substance #N** in the source term of convection-diffusion equation for **Substance #N**
F	The free term in the source term of the convective-diffusion equation for **Substance #N**

Subfolders «Physical processes > Mass transfer > Reactions» and «Physical processes > Mass transfer > Elements» for the mass transfer model "Chemistry"



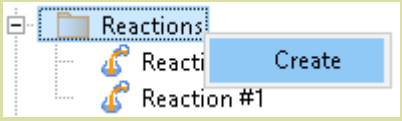
Subfolders **Mass transfer > Reactions** and **Mass transfer > Elements**

When the **Chemistry** mass transfer model is selected for a continuous phase, the subfolders **Reactions** and **Elements** appear in the **Mass transfer** element.

The subfolder **Mass transfer > Reactions** contains elements **Reaction #0**, **Reaction #1**, etc., which correspond to chemical reactions (you can change their standard names **Reaction #N**).

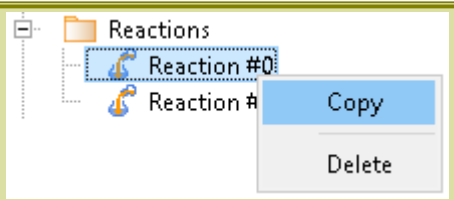
The subfolder **Mass transfer > Elements** contains ***Mass transfer substances*** (conservative scalars), which take part in the chemical reactions; they are selected from the **Substances** that locate in the folder **Phases > Phase #N > Substances**, see details in the section Chemistry.

The **Properties** windows of the subfolders **Reactions** and **Elements** are empty (but the **Properties** windows of the elements **Reactions > Reaction #N** and **Elements > *Mass transfer substance*** are *not* empty).



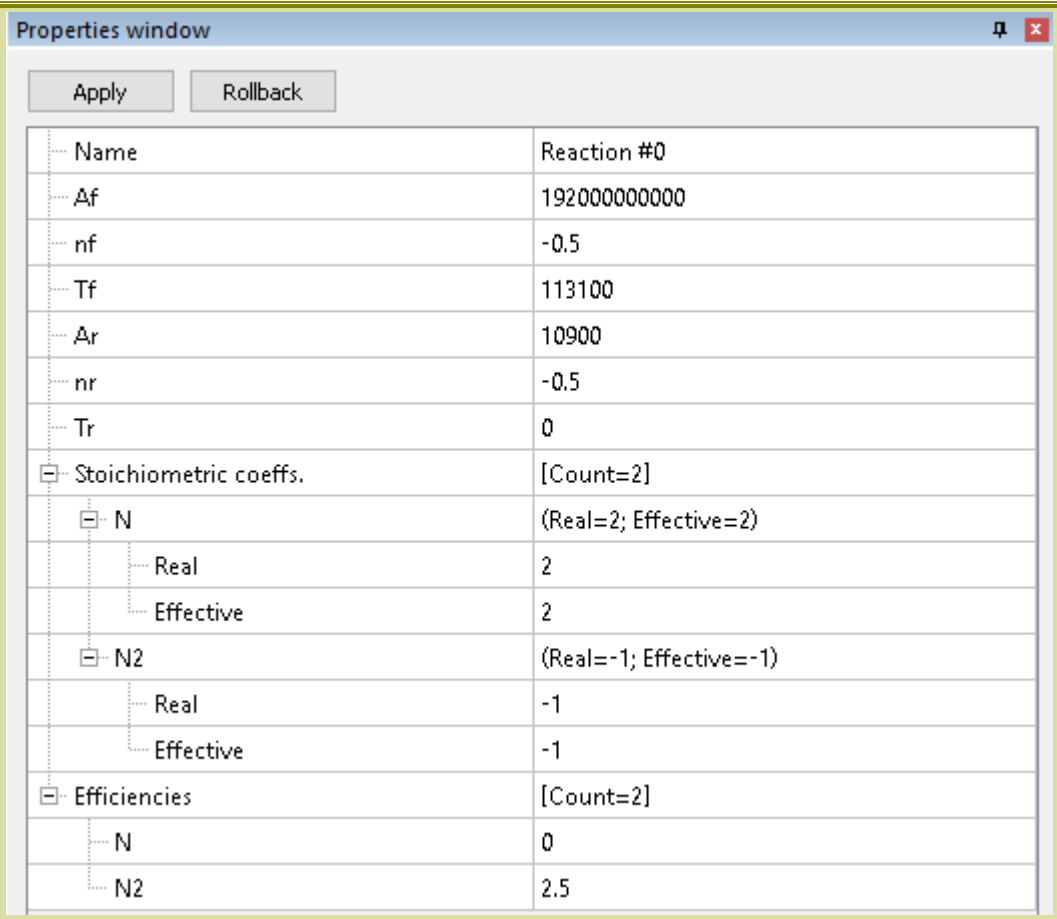
Context menu of the folder «Mass transfer > Reactions» (the "Chemistry" mass transfer model is used)

Menu item	Description
Create	Creates a new element **Reaction #N**



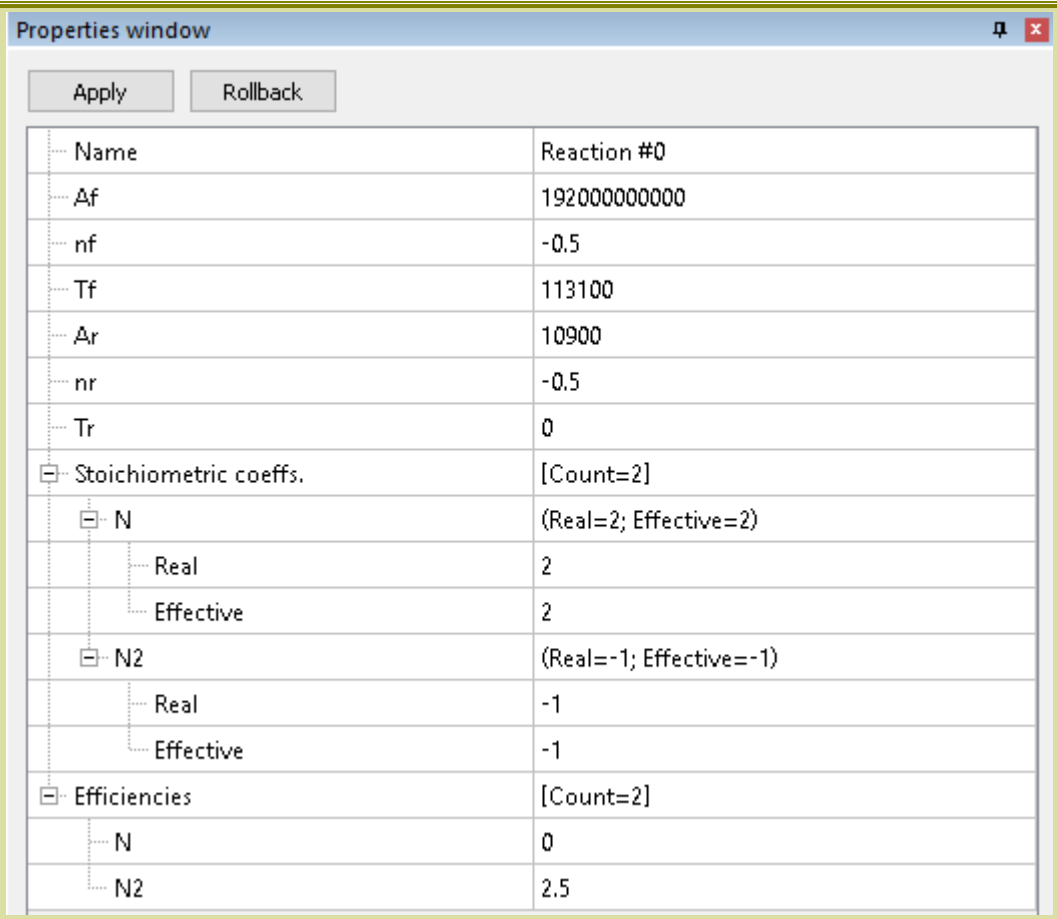
Context menu of the element «Mass transfer > Reactions > Reaction #N» (the "Chemistry" mass transfer model is used)

Menu item	Description
Copy	Creates a copy of the element **Reaction #N**
Deletes	Deletes the element **Reaction #N**



Parameters of the element «Physical processes > Mass transfer > Reactions > Reaction #N» for the "Chemistry" mass transfer model in continuous phases

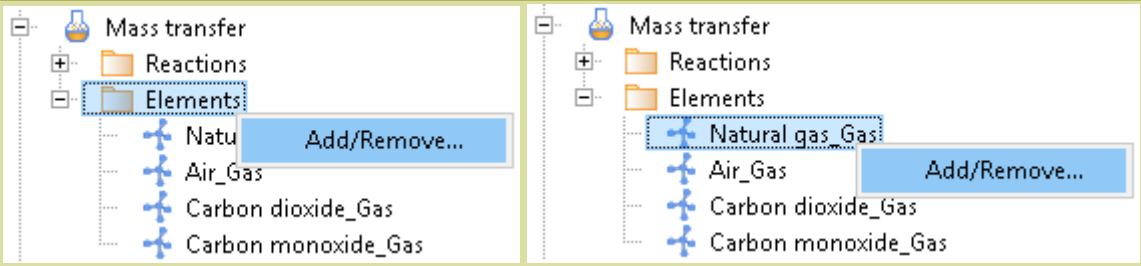
Parameter	Description
Name	Name of chemical reaction. You can enter your own name instead of the standard name **Reaction #N**.
Af **nf** **Tf**	Parameters of the forward reaction rate
Ar **nr** **Tr**	Parameters of the reverse reaction rate
Stoichiometric coeffs. > *(Substance)* > Real	Real (as in the molar equation of the reaction) and effective stoichiometric coefficients of **Substances**. These values determine the molar formula of the **Reaction**. Real stoichiometric coefficients in an elementary reaction determine balance of atoms. These coefficients are used for assembling source terms in equations for mass fractions of **Substances** in the **Phase**. Also these values determine powers of mass fractions of **Substances** in expressions for reaction rates. When elementary chemical reactions are simulated, you can specify only real stoichiometric coefficients (the effective stoichiometric coefficients coefficients are to be set the same). But often, to save computational resources, artificial brutto reactions are simulated with fractional powers of mass fractions of reacting **Substances** in expressions for reaction rates. In such cases you have to specify *real* stoichiometric coefficients for correct automatic assembling source terms the source terms and *effective* stoichiometric coefficients (that are be different from the real ones) according to empiric expressions for reaction rates. When real and effective stoichiometric coefficients of a reaction specified in the



Parameters of the element «Physical processes > Mass transfer > Reactions > Reaction #N» for the "Chemistry" mass transfer model in continuous phases

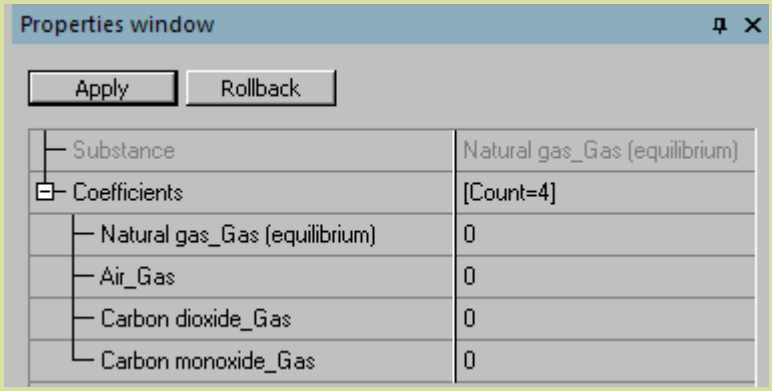
Parameter	Description
Stoichiometric coeffs. > *(Substance)* > Effective	*FlowVision*'s user interface are the same, this reaction is elementary.
Efficiencies > *(Substance)*	Coefficients, which determine the efficiencies of the **Substances** "third bodies" in the dissociation-recombination reactions

See details in the sections Chemistry, Theory > Physical processes > Mass transfer> Chemistry > Parameters and Theory > Physical processes > Mass transfer> Chemistry > Equations.



Context menus of the folder «Mass transfer > Elements» and elements «Mass transfer > Elements > Mass transfer substance» (the "Chemistry" mass transfer model is used)

Menu item	Description
Add/Remove	Opens the **Select substances** dialog box, which allows you to add or remove the ***Mass transfer substance*** elements that correspond to the **Substances** in the **Phase #N**.

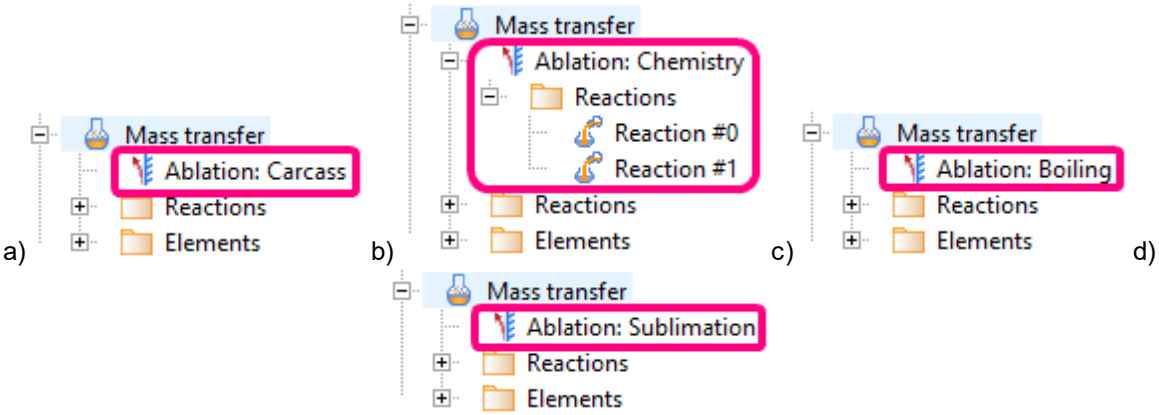


Parameters of the element «Physical processes > Mass transfer > Elements > *Mass transfer substance*» for the "Chemistry" mass transfer model in continuous phases

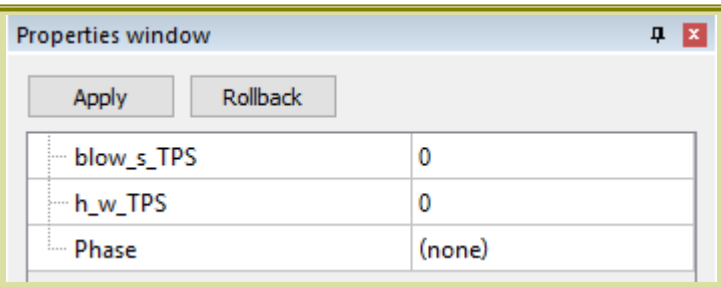
Parameter	Description
Substance	Name of the ***Mass transfer substance***. This field is only for your information, you cannot edit it here.
Coefficients > ...	Coefficients, which determine the weight of each **Substance** of **Phase #N** in the conservative scalar.

See details in the sections Theory > Physical processes > Mass transfer> Chemistry > Parameters and Theory > Physical processes > Mass transfer> Chemistry > Equations.

Subfolder «Physical processes > Mass transfer > Ablation:*Model of ablation*»



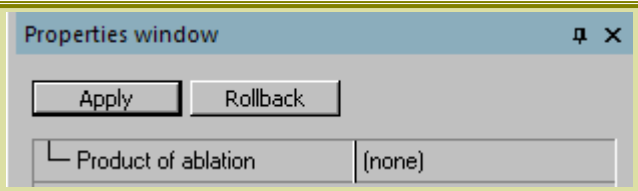
Subfolder **Ablation:*Model of ablation***
b) **Ablation:Carcass**; b) **Ablation:Chemistry**; c) **Ablation:Boiling**; d) **Ablation:Sublimation**



Parameters of the element «Physical processes > Mass transfer > Ablation:Carcass»

Parameter	Description
blow_s_TPS	Dimensionless blow-in rate of the ablation products of the thermal-protective coating's matrix into boundary layer
h_w_TPS	Thermodynamic enthalpy of the ablation products of the thermal-protective coating's matrix, [m^2/s^2]
Phase	Select here the **Phase**, which corresponds to the ablation products of the thermal-protective coating.

The **Properties** window of the element **Physical processes > Mass transfer > Ablation:Chemistry** has no parameters.

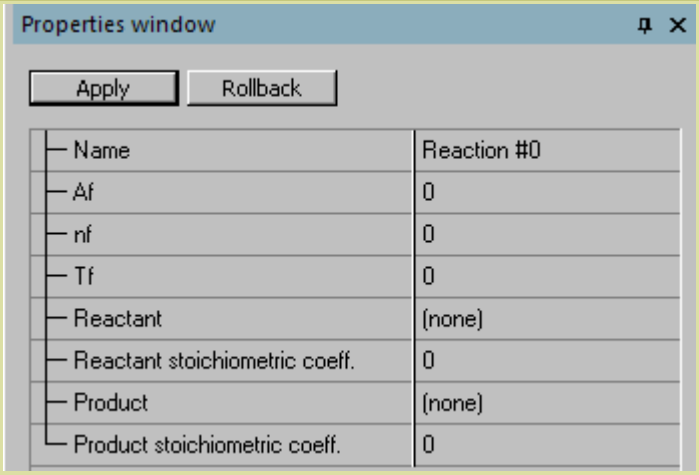


Parameters of the element «Physical processes > Mass transfer > Ablation:Boiling» and «Physical processes > Mass transfer > Ablation:Sublimation»

Parameter	Description
Product of ablation	Product of the ablation (this is selected from a list, you can select a **Substance** from those ones that are presented in **Phase #N**).

When the **Chemistry** ablation model is used, the element **Ablation:Chemistry** contains the **Reactions** folder, which consist elements corresponding to certain chemical reactions of ablation, similarly as it is done for the element **Physical processes > Mass transfer > Reactions** (see above).

However the **Properties** windows of these reactions of ablation contain slightly different sets of parameters:

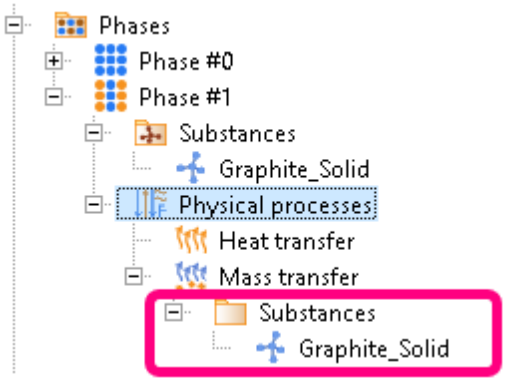


Parameters of the element «Physical processes > Mass transfer > Ablation:Chemistry > Reactions > Reaction #N»

Parameter	Description
Name	Name of chemical reaction. You can enter your own name instead of the standard name **Reaction #N**.
Af	Parameters of the forward reaction rate
nf	
Tf	
Reactant	The reactant substance, it is selected from the list of **Substances** of **Phase #N**
Reactant stoichiometric coeff.	The stoichiometric coefficient of the reactant, it defines the molar formula of the reaction
Product	The product substance, it is selected from the list of **Substances** of **Phase #N**. The **Product** substance and the **Reactant** substances have be not same.
Product stoichiometric coeff.	The stoichiometric coefficient of the product, it defines the molar formula of the reaction

See details in the sections Theory > Physical processes > Mass transfer> Ablation > Parameters and Theory > Physical processes > Mass transfer> Ablation > Equations.

Child elements of the subfolder «Physical processes > Mass transfer» for dispersed Phases of the Carcass type

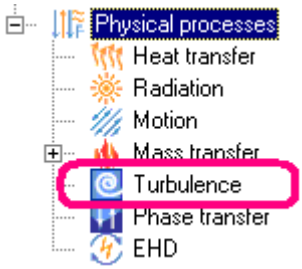


In dispersed **Phases** of the **Carcass** type the element **Physical processes > Mass transfer** contains the subfolder **Substances** with elements corresponding to all **Substances** of the **Phase**. In properties of these elements coefficients **D** and **F** are specified that determine the decomposition rate of the appropriate substances in the carcass. If some substance isn't decomposed, zero coefficients are to be set for it (**D=0** and **F=0**).

For substances of the carcass initial-value problems are solved (*not* initial boundary value problems), see Equations for porous carcass > Process "Mass transfer", so no boundary conditions are set for **Substances** of the carcass.

Coefficients **D** and **F** can be set by formulae.

Parameters of the physical process "Turbulence"



The **Physical processes > Turbulence** element in the project tree

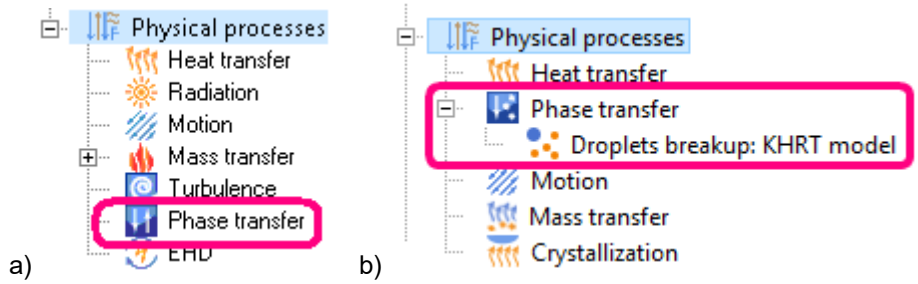
Parameters of the **Physical processes > Turbulence** element:

Parameter	Description
Math. model **Time step coefficient**	*See table "Parameters of the 'physical process' elements, which are common to all physical processes"*
PrandtlTurb	Turbulent Prandtl number Pr_t
SchmidtTurb	Turbulent Schmidt number Sc_t
E, log.law	Constant E, which is included in the formula for the velocity profile of the wall in the standard model of wall functions
von Karman constant	Karman constant κ, which is included in the formula for the velocity profile of the wall in the standard model of wall functions (logarithmic velocity profile near the wall).
Roughness constant	The turbulence model's parameter, which specifies influence of the of the wall roughness. See Theory> Physical processes> Turbulence> Parameters.
Therm. expansion	Enable/disable the account of the buoyancy force in turbulence models.

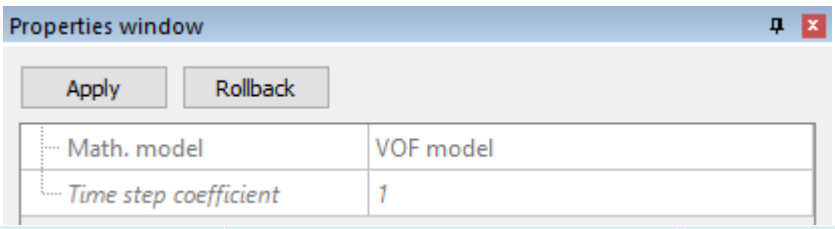
Parameter	Description
	The temperature expansion coefficient β, which is present in terms of equations for k and ε.
KEteta model	Here you select a model of turbulent heat transfer: • **(none)**: the calculation is carried out without using any heat transfer model • **AKN**: the calculation uses the AKN turbulent heat transfer model • **S&S**: the calculation uses the SS turbulent heat transfer model • **LMS**: the calculation uses a specific model of turbulent heat transfer, which requires a special license
other parameters of the turbulence model	Numerical parameters (constants) of turbulence models, the composition of these parameters depends on what kind of turbulence model is selected (it is set by the **Turbulence** parameter in properties of the folder **Phases > Phase #N > Physical processes**). You might need to change these settings when solving specific tasks, for example, when there are recommendations to do so in scientific literature. If you do not clear the meaning of these parameters do not change them, use the default values. See description of these parameters in section Theory> Physical processes> Turbulence> Parameters.

Interaction of a **Phase** and a wall (wall function for turbulent parameters) is specified in properties of individual **Boundary conditions** that are of the **Wall** type. See section Folder «Boundary conditions».

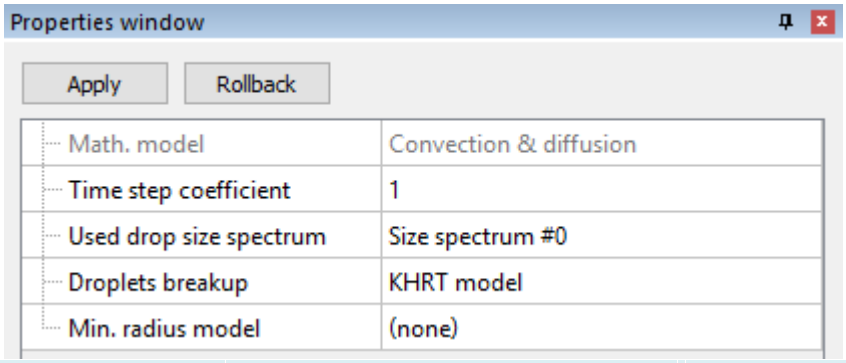
Parameters of the physical process "Phase transfer"



The **Physical processes > Phase transfer** element in the project tree:
a) for continuous **Phases**;
b) for dispersed **Phases** of the **Particles** type.



The **Properties** window of the element **Physical processes > Phase transfer** (for interaction of two continuous phases)

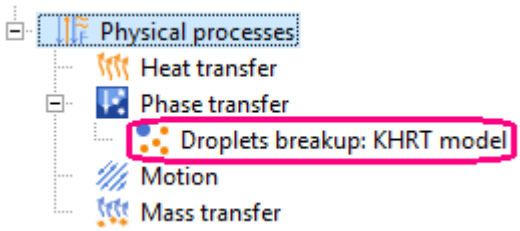


The **Properties** window of the element **Physical processes > Phase transfer** (for transfer of a dispersed phase)

Parameters of the "Physical process > Phase transfer" element	
Parameter	**Description**
Math. model	Name of the mathematical model (it is not editable). For interaction of two continuous phases, **Math. model = VOF model**. For transfer of a dispersed phase, **Math. model = Convection & diffusion**. The **Math. model** parameter is informational only, it is filled by the program automatically and you cannot change it here.
Time step coefficient	*See table "Parameters of the 'physical process' elements, which are common to all physical processes".* This parameter is not applied and is not available for continuous **Phases**.
Used drop size spectrum	This parameter specified a **Size spectrum #N**, which will be used in simulation of the **Phase #N** in the following cases: • When the dispersed mass transfer is *disabled* (i.e. when **Mass transfer = (none)** is set in properties of **Phase #N > Physical processes**). The selected **Size spectrum #N** is used in the computation while no **Size spectrum** can be set in initial and on boundary conditions. • *Or* the dispersed mass transfer is *enabled* (i.e. when **Mass transfer = Mass transfer** or **Mass transfer = Coal** is set in properties of **Phase #N > Physical processes**). The selected **Size spectrum #N** is used to define non-changing sizes of particles with changing density. The **Size spectra**, which are set in initial and on boundary conditions, are *not applied*.
Droplets breakup[*)]	A model of breakup of droplets, it is selected from a list. Possible options are: • **(none)** • **KHRT model** • **RD model** • **WAVE model** See details in the section Theory > Physical processes > Processes in the presence of dispersed medium > Equations for particles > Process 'Phase transfer'.
Min. radius model[*)]	A model of coalescence of droplets. Possible options are: • **(none)** • **MW model** • **Hiroyasu model** This parameter is not available when **Droplets breakup = (none)**. See details in the section Theory > Physical processes > Processes in the presence of dispersed medium > Equations for particles > Process 'Phase transfer'.

[*)] These parameters are available for dispersed **Phases** of the **Particles** type with *liquid particles* and the **Mass transfer** physical process enabled.

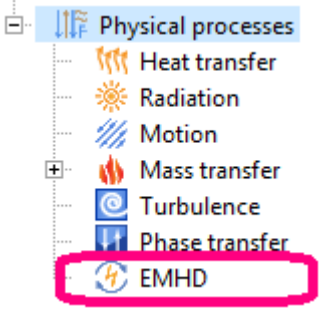
Parameters of the droplets breakup model



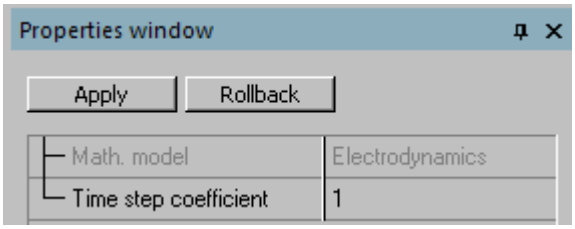
When a droplets breakup model is selected in properties of the dispersed phase transfer (see subsection *'Parameters of the physical process "Phase transfer"'* above), the **Phase transfer** element obtains a child element **Droplets breakup: *Model of droplets breakup***, in properties of which parameters of the droplets breakup model are set (see details in the section Theory > Physical processes > Processes in the presence of dispersed medium > Equations for particles > Process 'Phase transfer').

Parameter	Description
Small drops account	This parameter enables or disables taking generation of small particles into account in the the droplets breakup model. Possible options are: **Not account | Account (hybrid model)**.
Hyb. model %	The $SD^{(\%)}$ parameter, which is used in taking generation of small particles into account. This parameter is available when **Small drops account = Account (hybrid model)**.
B_1 coef.	Constants of droplets breakup models. Availability of these parameters depends on the selected droplets breakup model.
C_t coef.	
C_rt coef.	
B_2 coef.	

Parameters of the physical process "EMHD" (electromagnetohydrodynamics)



The **Physical processes > EMHD** element in the project tree

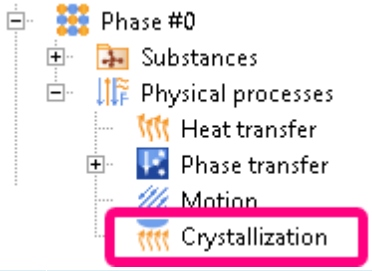


The **Properties** window of the element **Physical processes > EMHD**

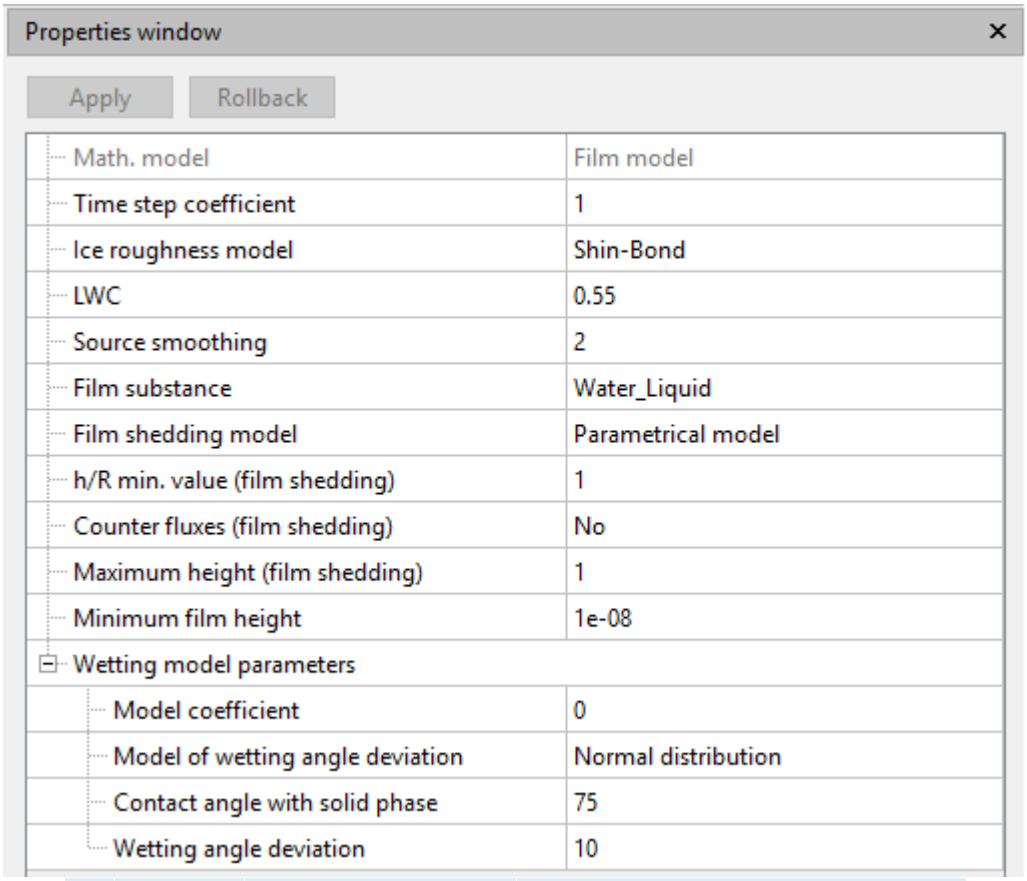
Parameters of the "Physical process > EMHD" element	
Parameter	**Description**
Math. model	*See table "Parameters of the 'physical process' elements, which are common to*

Parameters of the "Physical process > EMHD" element	
Parameter	**Description**
Time step coefficient	

Parameters of the physical process "Crystallization" (simulating the icing)



The **Physical processes > Crystallization** element in the project tree (for **Phases** of the **Particles** type) in the project tree



The **Properties** window of the element **Physical processes > Crystallization**

The **Crystallization** physical process can be enabled in **Phases** of the **Particles** type. Here parameters of icing are set.

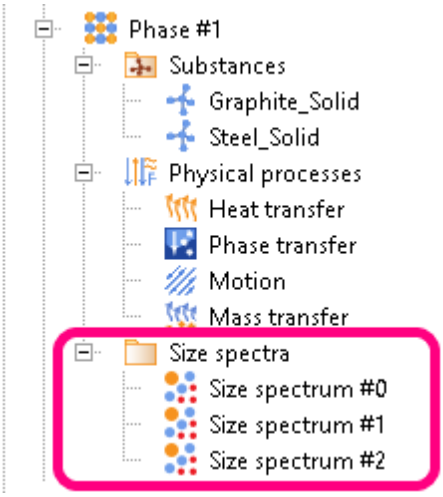
Parameters of the "Physical process > Crystallization" element	
Parameter	**Description**
Math. model	*See table "Parameters of the 'physical process' elements, which are common to all physical processes".* The **Math. model** parameter can have values: **Dry model** | **Film model**.
Time step coefficient	

Parameters of the "Physical process > Crystallization" element	
Parameter	**Description**
	(the value is determined by the value of the **Crystallization** parameter of the element **Phase #N > Physical processes**)
Ice roughness model	This is an empiric model of roughness applied to calculate the equivalent grit roughness on a surface of the geometry or on a surface of the solid phase. The surface becomes rough due to sequential growth and crystallization of subcooled drops of the dispersed phase on the surface. The empiric models of roughness are recommended for use only when icing of aircraft is simulated. The used ice roughness model. Possible options are: **(none)** | **Shin-Bond** | **Shin-Bond (local)**.
LWC	Liquid water content, which is amount of condensed water containing in a unit volume of a cloud, in an undisturbed two-phase flow. The **LWC** parameter specifies the initial liquid water content in cells. This parameter is specified as a constant only. This parameter is available when **Ice roughness model = Shin-Bond**. In the program's user interface the **LWC** parameter is set in [g/m^3]. A numerical value specified as a constant only
Source smoothing	Number of iterations of surface smoothing of the volume source of the dispersed phase's substance that settle on the surface from the two-phase flow. This is an integer number specified as a constant only. Specifying values above **0** allows obtaining more smooth shape of the ice body.
Film substance	This **Substance**, which will be used in calculations of thermal balances and the film's motion. The **Substance** of the dispersed phase might be not liquid only; it can be solid (as snowflakes or ice particles). The value is selected from the list.
Film shedding model	The model of shedding the film from the surface (loss of the dispersed phase's substance from the film). Forming a new dispersed phase due to the film shedding is not taken into account yet. This parameter is available when **Math. model = Film model**. Possible values: • **(none)** • **Parametrical model** – the film's shedding from the surface occurs when any of the following conditions takes place: ○ ratio of the film's thickness h to the radius R of curvature of the surface in the near-surface cell exceeds the value of the **h/R min. shedding value** parameter (see below) ○ *or* projections of vectors of the film's velocity in adjacent near-surface cells to the normal to the face between these cells are directed towards to each other.
h/R min. shedding value	The minimal value h/R for the film shedding model. This parameter specifies the threshold ratio of the film's thickness h to the radius R of curvature of the surface. When the h/R ratio exceeds the specified is value, film shedding occurs (film shedding can also occur due to another reason, see subsection *Film shedding model "Parametrical model"*). This is a numerical value that can be specified as a constant only. The default value is **1**.
Minimum film height	This is an empirical estimate of the minimal height of the film (thickness), [m], below which the film is stable in this specific problem setting. In simulations of aircraft icing, when you have no empirical estimate for this minimal height of the film, it is recommended to keep the default value of this parameter.

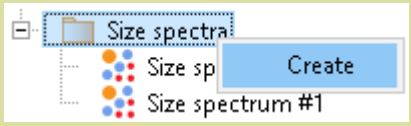
Parameters of the "Physical process > Crystallization" element	
Parameter	**Description**
	This is a numerical value that can be specified as a constant only. The default value is **1e-8**.
Wetting model parameters	The group of parameters of the wetting model
Wetting model parameters > Model coefficient	The wetting model's coefficient β, see formula Cryst.12. The default value is **0**.
Wetting model parameters > Model of wetting angle deviation	Distribution law of the random value θ, which is local wetting angle on the solid surface, see formula Cryst.12. The **Normal distribution** value only is possible now.
Wetting model parameters > Contact angle with solid phase	The mathematical expectation of the random value θ, which is local wetting (contact) angle on the solid surface, [degree], see formula Cryst.12. The default value is **75**.
Wetting model parameters > Wetting angle deviation	The mean-square deviation of the random value θ, which is local wetting (contact) angle on the solid surface, [degree], see formula Cryst.12. The default value is **10**.

See details in the chapter Theory, in the section Crystallization.

Folder «Phases > Phase #N > Size spectra» and elements «Size spectrum #N»



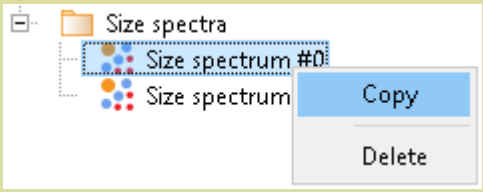
The folder **Phase #N > Size spectra** contains child elements **Size spectrum #N**. Each element **Size spectrum #N** contains information about size distribution of particles in groups (number of groups is set in properties of the **Phase** by the **Number of size groups** parameter).



Context menu of the folder «Size spectra»

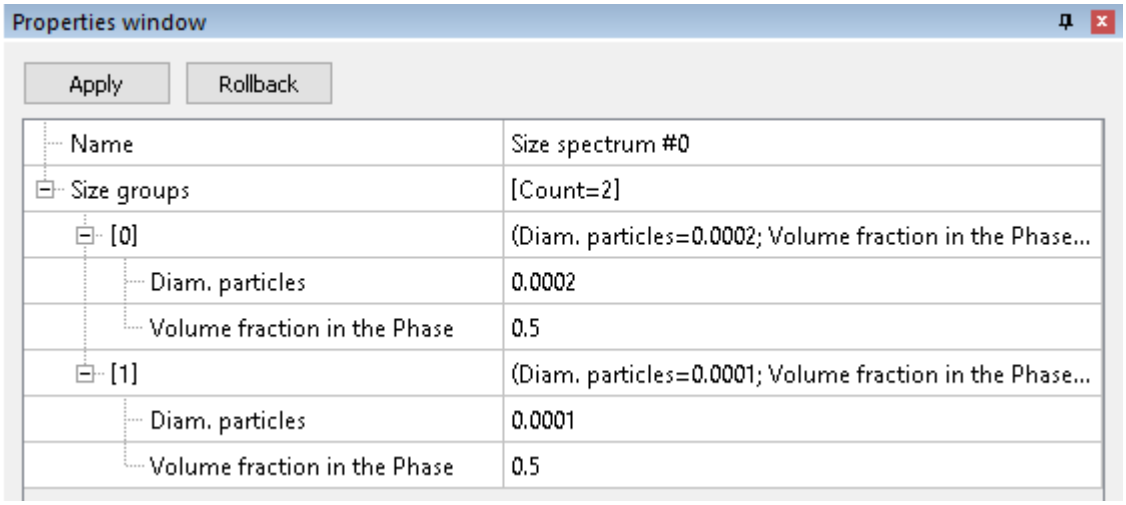
Menu item	Description
Create	Creates a new element **Size spectrum #N**

The folder **Phase #N > Size spectra** has no parameters in its **Properties** window.



Context menu of the element «Size spectrum #N»

Menu item	Description
Copy	Creating an element, which is a copy of the selected element
Delete	Deleting the selected element from the project tree



Elements **Size spectrum #N** have the following parameters:

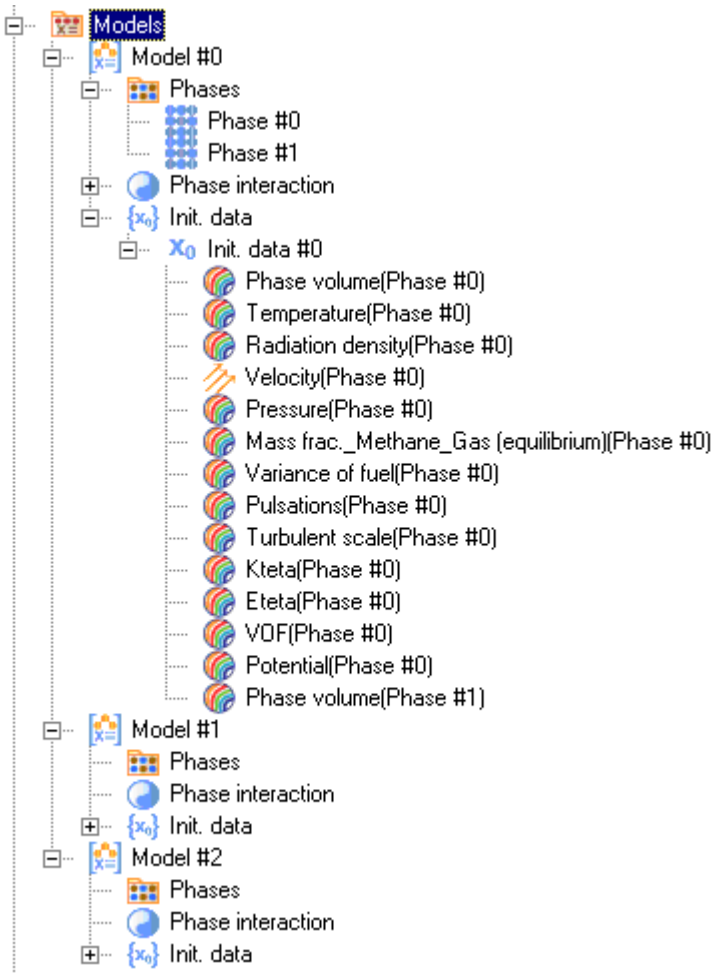
Parameter	Description
Name	Name of the **Size spectrum** in the project tree. By default this name is **Size spectrum #N** and you can change it if you wish.
Size groups > [N] > ...	Array of size spectra of disperse particles. Number of elements in this array is specified by the **Number of size groups** parameter in properties of the **Phase**. Number of elements in this array is set by the **Number of size groups** parameter in properties of the **Phase**.
Size groups > [N] > Diam. particles	Diameter d^i of particles in this size spectra, [m]. The default value is **1e-09**, which means 10^{-9} [m].
Size groups > [N] > Volume fraction in the Phase	Volume fraction φ^i of particles in this size spectra in the whole volume of the dispersed **Phase**. The sum of volume fractions of all size spectra is to be 1. If this requirement is not complied, an error will be displayed in the **Log** window, you will not be able to save and run the project. When **Number of size groups = 1** is set in properties of the **Phase**, the **Volume fraction in the Phase** has the value **1** and cannot be edited.

Variables of a **Particles** dispersed phase that relate to different size groups are marked in the program's interface by indexes of their size groups in square brackets.

Example: **Velocity (disp.) [0]**, **Velocity (disp.) [1]**, etc. (numeration of size groups starts from **0**).

See illustrations, notations, and details in the section Spectra of particle sizes.

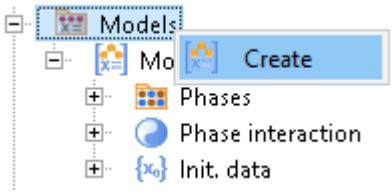
8.1.8.3.5 Folder «Models»



Model is a set of **Phases** with specified *inter-phase interactions*.

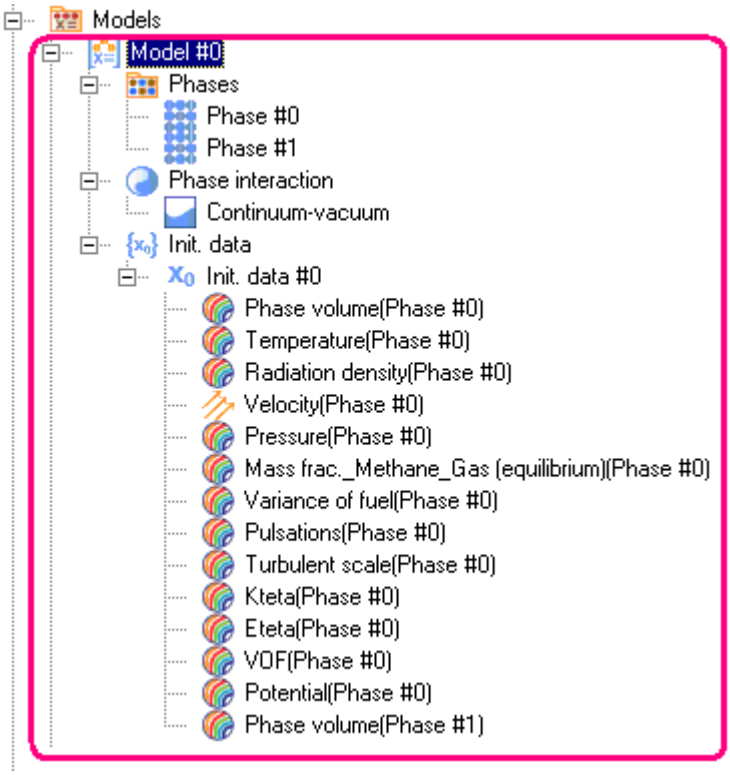
Specifying a **Model** becomes possible only after specifying **Phases**.

The **Models** folder contains subfolders **Model #N**, which are created by the **Create** command selected from the context menu:



The context menu of a folder "Models"	
Menu item	**Description**
Create	Add a new **Model** (a **Model #N** subfolder) into the **Models** folder

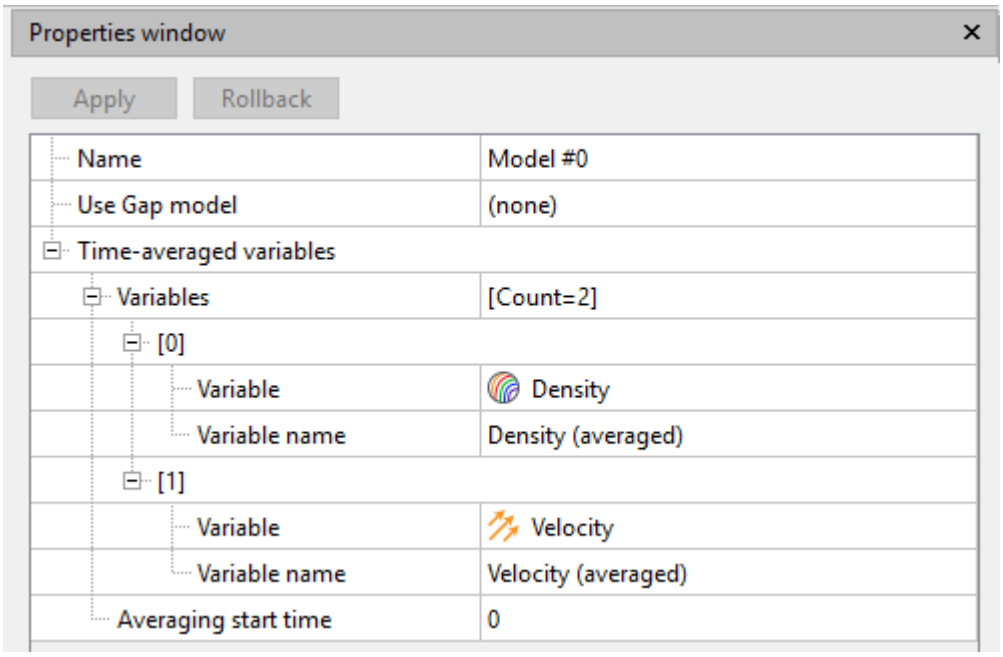
Folder "Models> Model #N"



Element **Model #N** in the project tree

The folder **Models > Model #N** contains:

- the folder **Phases** with elements **Phase #N**
- the folder **Phase interaction**
- the folder **Init. data** with subfolders **Init. data #N**
- an element, which corresponds to the selected gap model if any (for example, **Standard Gap model**) depending on the value of the parameter **Use Gap model** in the **Properties** of the folder **Models > Model #N**



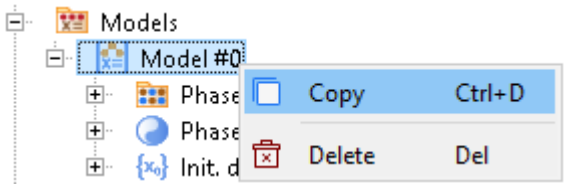
The **Properties** window of the element **Model #N**

Parameters of the element **Model #N**:

Parameter	Description
Name	The **Model**'s name
Use Gap model	Possible options are: • **(none)** – no gap model is used in the project's computation • **Standard Gap model**
Time-averaged variables > ...	This is the list of the averaged variables and the time t_0 when the averaging period begins. See section Averaged variables.
Time-averaged variables > Variables > ...	This is the array of the averaged variables.
Time-averaged variables > Variables > [N] > Variable	This is the source (not averaged) variable, which will be used to calculate the averaged variable. It is selected from a drop-down list that includes all the model's and user variables existing in the project (but this list doesn't include previously averaged variables that were created before).
Time-averaged variables > Variables > [N] > [Variable name]	Name of an averaged variable. By default the program forms this name based on the name of the source variable appended with word " **(averaged)**". If you wish, you can specify another name here.
Time-averaged variables > Averaging start time	The time moment t_0, [s], when the averaging period begins. Calculation of the averaged variables will start when the project's current time reaches t_0. Correct calculation requires that you either specify t_0 greater then the project's current time or the project starts from its beginning. If you specify t_0 less then the project's current time, the program outputs a warning: "**Found incorrect values for averaging variables: - The new value of the variable averaging start time is less than the current time in the calculation. It is recommended to restart the calculation.**". The default value is **0**.

Specific details of calculating the averaged variables:

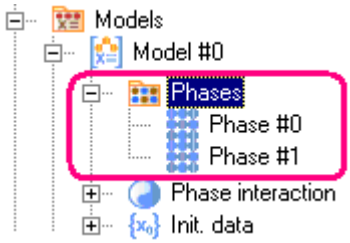
- If you need to restart the averaging, you have to specify the **Averaging start time** (t_0) greater then the project's current time and the program has to do at least one computational step before the specified **Averaging start time** (otherwise previously calculated values of averaged variables would not be reset to zero).
- If, during the calculation, after the **Averaging start time** is reached, the **Averaging start time** would is changed so it would be less or equal to the project's current time (i.e. if the beginning of the averaging period is moved within the period), the averaging procedure would continue its previous work and previously calculated values of averaged variables would *not* be reset to zero.
- If, during the calculation, after the **Averaging start time** is reached, a new variable would be added into the list of the averaged variables, then this variable would be averaged only the current computational step only (and the program would warn you about this).
- If, during the calculation, after the **Averaging start time** is reached, a source variable would be replaced, based on which an existing averaged variable was created, then this averaged variable would be calculated from scratch strarting from the next computational step (and the program would warn you about this).



Context menu of the element **Model #N**

Context menu of the element "Model #N"	
Menu item	**Description**
Copy	Creating an element, which is a copy of the selected element
Delete	Deleting the selected element from the project tree

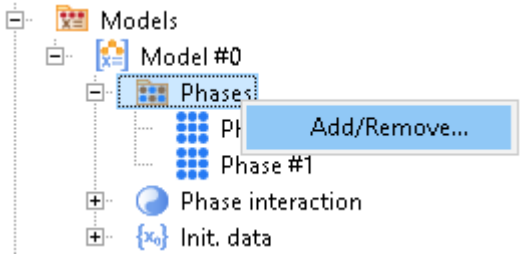
Folder "Models > Model #N > Phases" and elements of the "Models > Model #N > Phases > Phase #N"



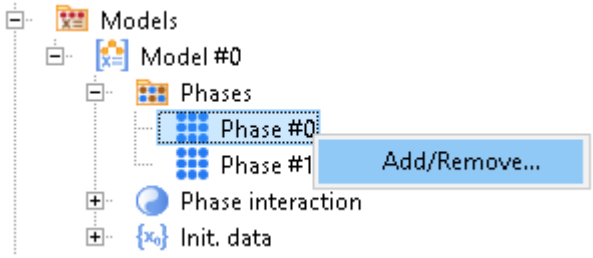
Folder **Models > Model #N >Phases**

The folder **Models > Model #N > Phases** stores references to the **Phases**, which are included into **Model #N**. These **Phases** are represented by elements **Phase #N**.

Context menu of the folder **Models > Model #N > Phases** and context menus of elements **Models > Model #N > Phases > Phase #N** contain the item **Add/remove** only, which allows you to add or remove a **Phase** into/from the **Model**.

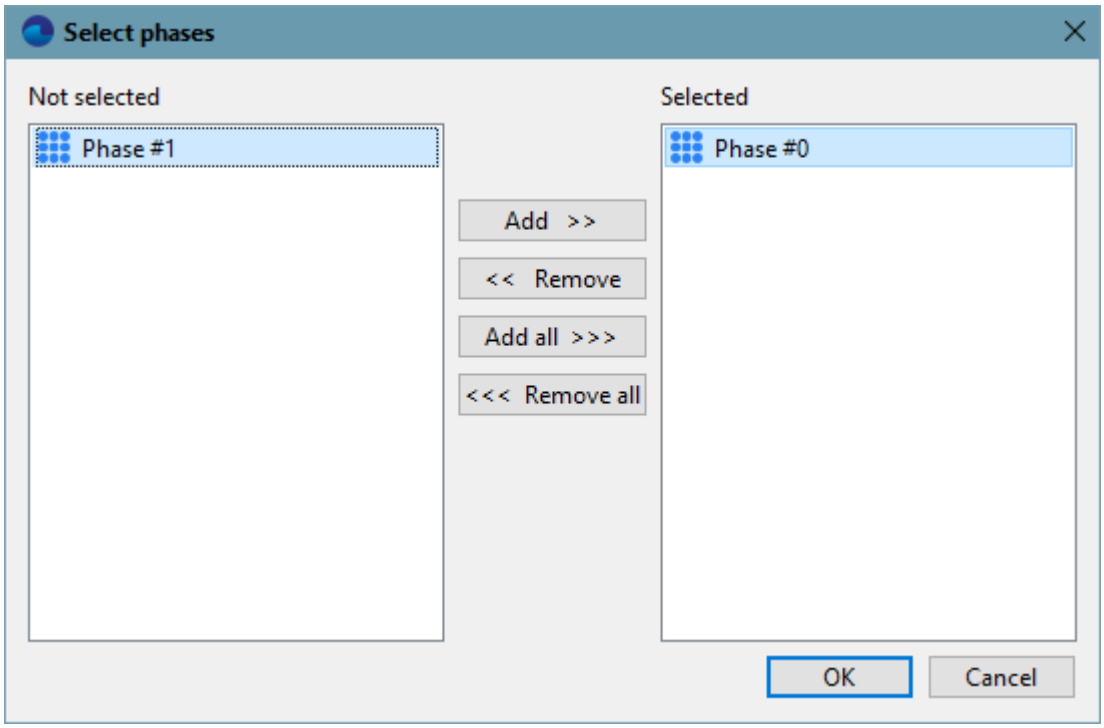


Context menu of the folder **Models > Model #N >Phases**



Context menu of the element **Models > Model #N > Phases > Phase #N**

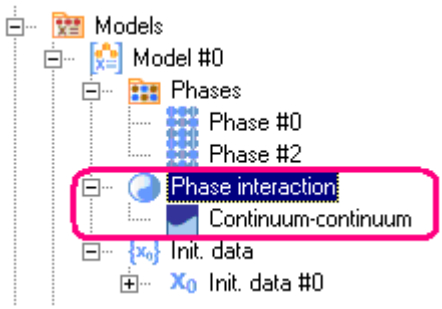
Computation of multiple dispersed **Phases** in one **Model** is not possible.

The context menu of the folder "Models > Model #N > Phases" and elements "Models > Model #N > Phases > Phase #N"	
Menu item	**Description**
Add/ Remove	This command opens the**Select phase** window where you can add **Phase #N** elements into **Model #N** or remove them from there:  To improve stability of the solution, you should to arrange the **Phases**, which you have selected in the **Select phases** window, by descending of their densities. For example, if one of the **Phases** is liquid and the other is gas or vacuum, then you should place on the first place the **Phase** with liquid and on the second place the **Phase** with gas or vacuum. If densities of the **Phases** are of the same order, you can arrange the **Phases** in any order. ***Note:*** You can select all the **Phases** that were created before in the folder Phases.

Folder "Models > Model #N > Phase interaction"

The folder **Models > Model #N > Phase interaction** contains child element(s) ***Phase interaction***, which define parameters of interaction of specific *two* **Phases** in **Model #N**.

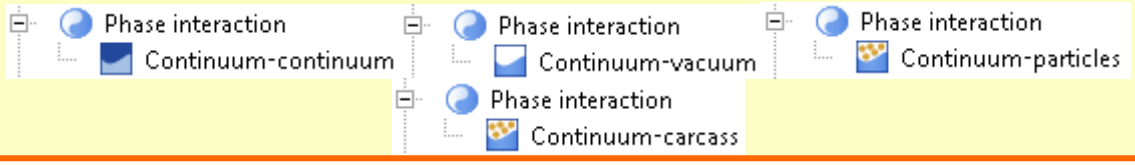
A ***Phase interaction*** element is created in the project tree automatically and it is automatically deleted from the project tree if any of its **Phases** is removed from the **Model**.

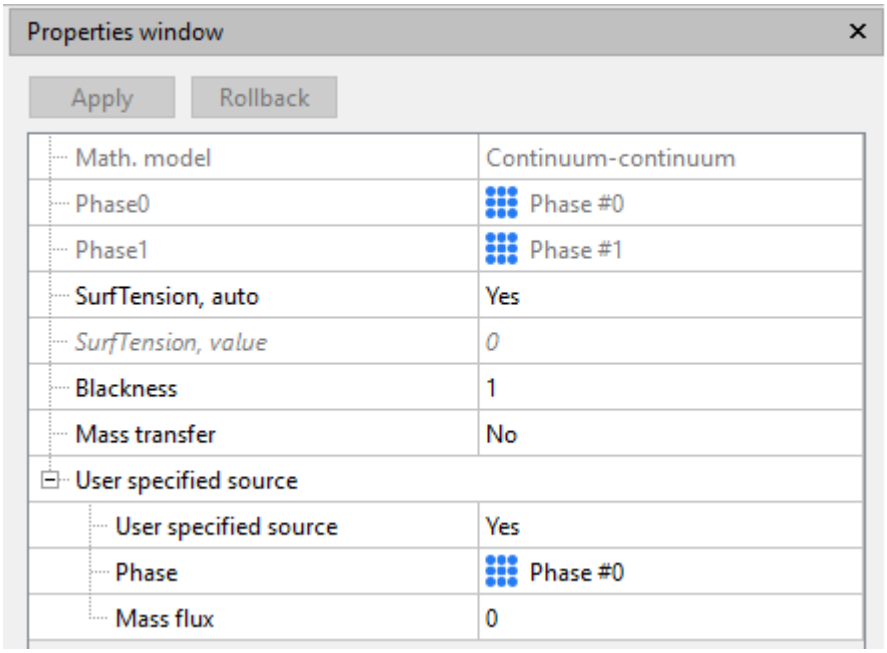


The **Phase interaction** folder in the project tree

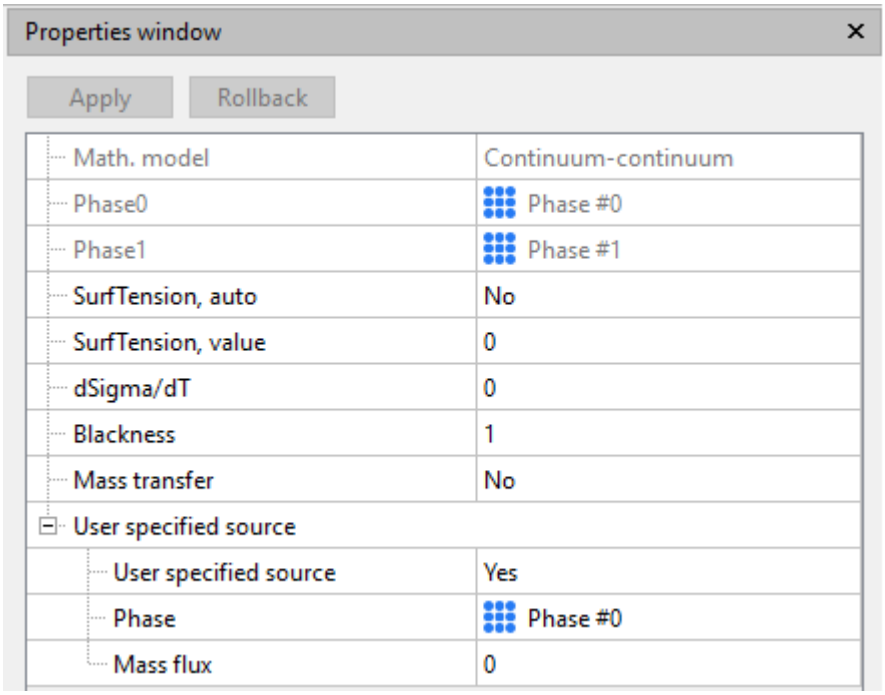
If you wish to simulate the motion in only one of the **Phases** and movement of the phase interface surface, you don't have to specify a **Substance** and **Physical processes** in the second **Phase**.

Names and icons of elements **Phase interaction** depend on which **Phases** they relate:

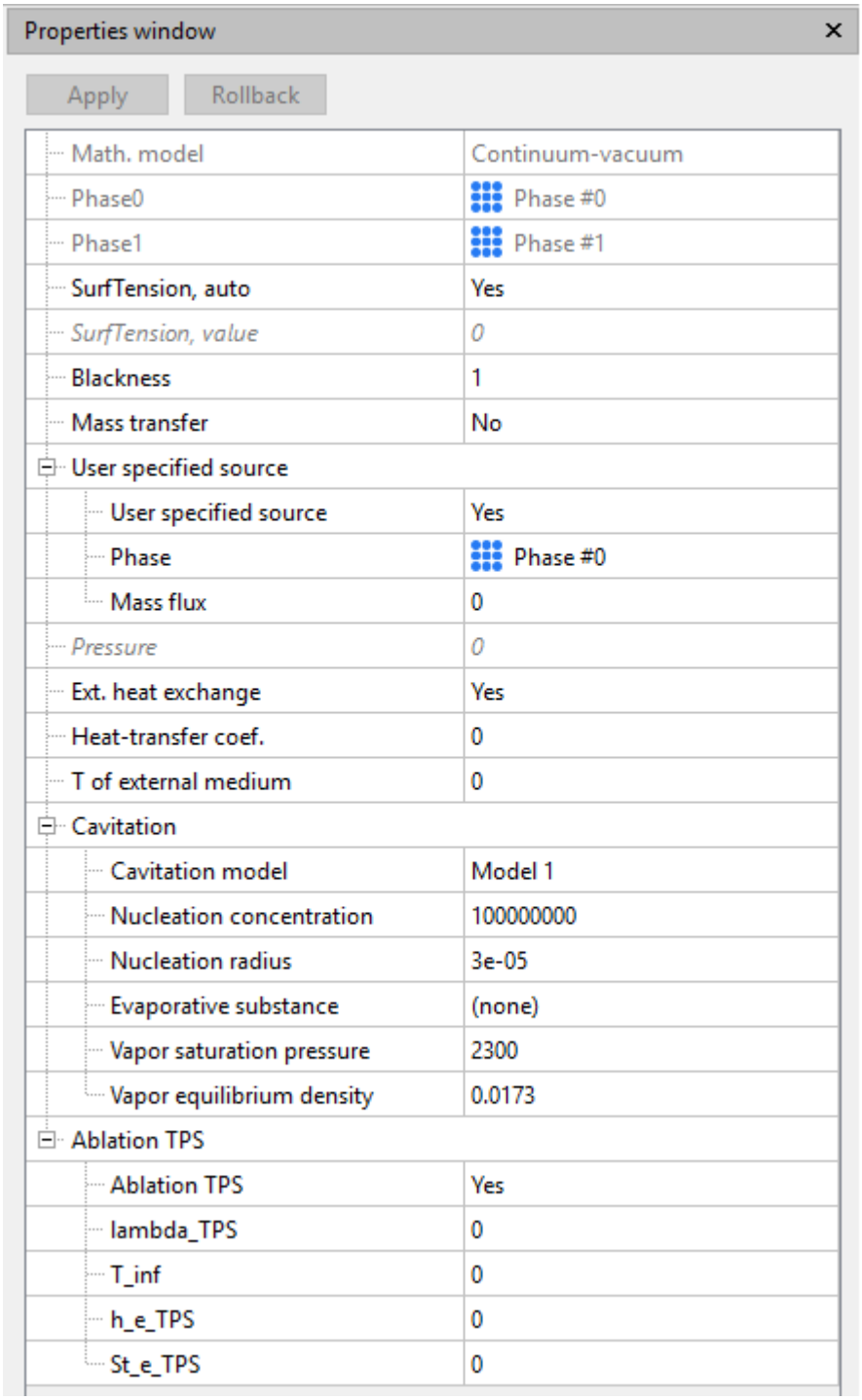




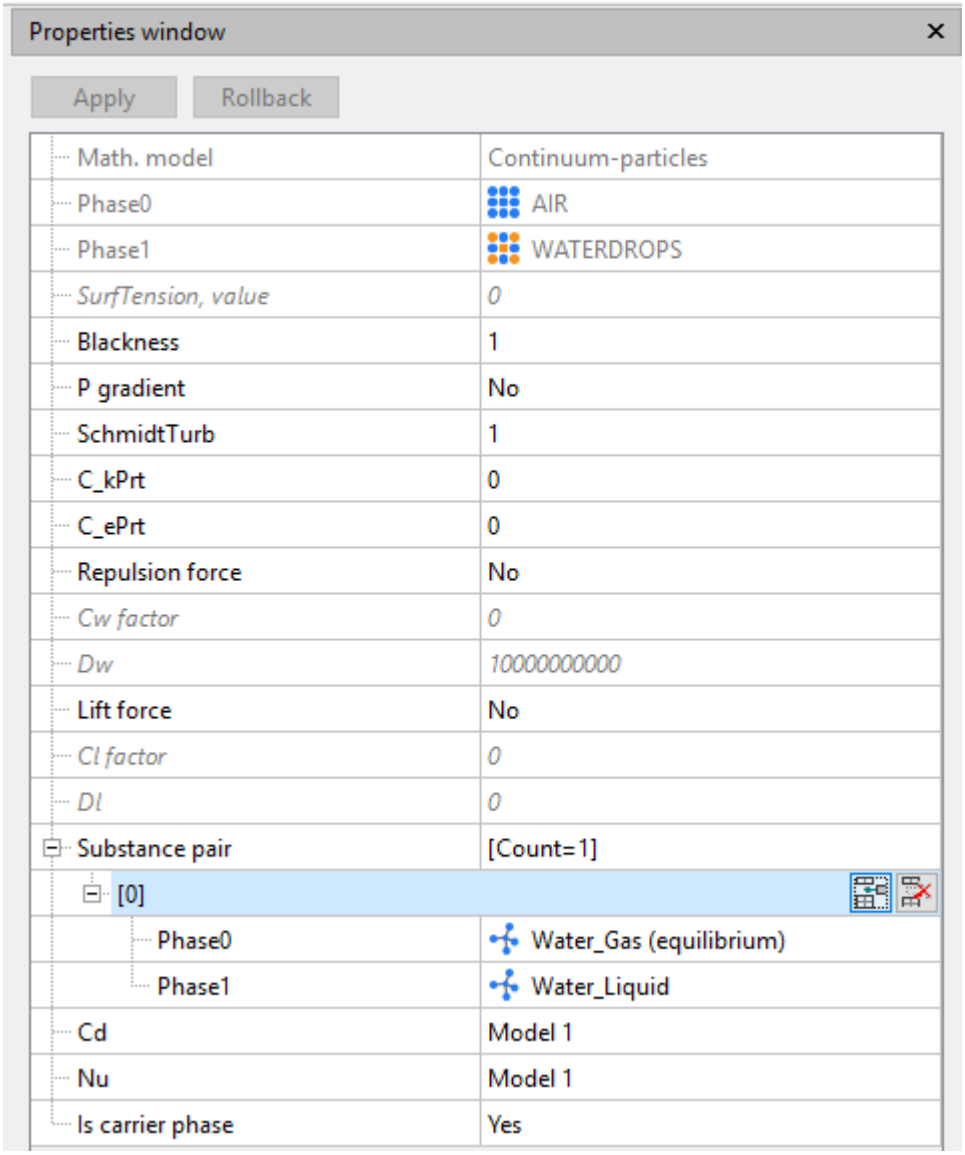
Properties of the element **Phase interaction** for a pair of phases **Continuum-continuum** with automatic taking into account the surface tension



Properties of the element **Phase interaction** for a pair of phases **Continuum-continuum** with user-defined surface tension



Properties of the element **Phase interaction** for a pair of phases **Continuum-vacuum**



Properties of the element **Phase interaction** for a pair of phases **Continuum-particles**

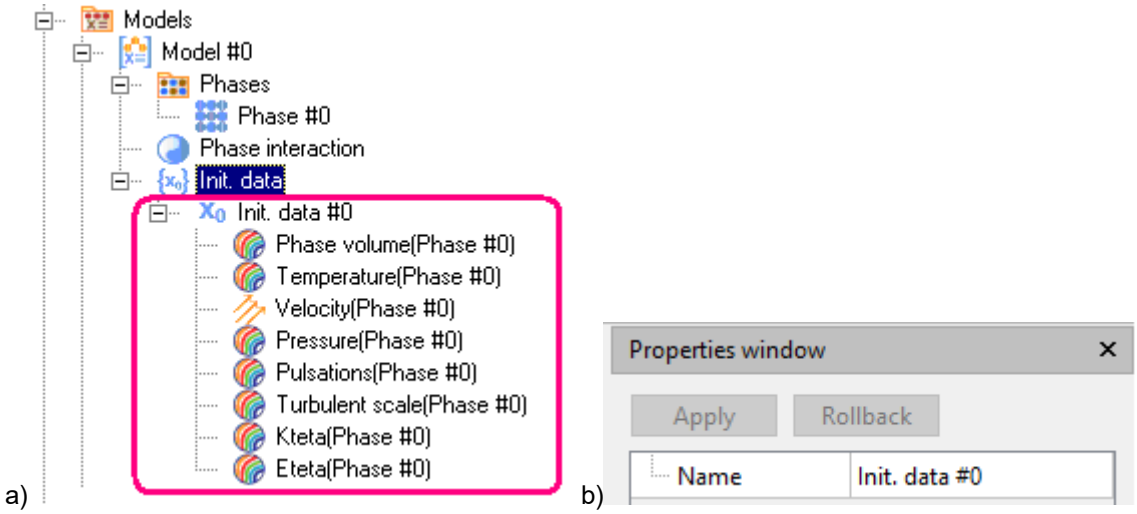
Properties of the element "Phase interaction"	
Parameter[*)]	**Description**
Math. model	The name of a mathematical model of the interaction between the phases. Is set automatically according to the types of interacting **Phases**. For interaction between *two continuous phases*, the value is: • **Continuum-continuous**, if the physical process **Movement** is specified for both **Phases** • **Continuum-vacuum**, if no **Movement** is specified on one of these **Phases** For interaction between *a continuous phase and a dispersed phase* this parameter has value **Continuum-particles** or **Continuum-carcass**. This parameter is informational and can not be changed by the user.
Phase0	The first **Phase** in the pair of interacting phases
Phase1	The second **Phase** in the pair of interacting phases
SurfTension, auto	Possible options are: • **Yes** - calculate the surface tension automatically • **No** - specify the surface tension manually using parameters **SurfTension, value** and **dSigma/dT** (see below)

Properties of the element "Phase interaction"	
Parameter[*)]	**Description**
	When selecting **SurfTension, auto = Yes** surface tension is calculated as the absolute difference of the surface tensions of the **Phases**, specified by the **Surface tension** parameters of the **Substances** forming the two **Phases**, which are included in this pair of interacting phases.
SurfTension, value	The surface tension coefficient σ, [N/m] (it is available when **SurfTension, auto = No**). The **SurfTension, value** parameter can be set by a constant, a formula or a table. When **SurfTension, value** is set by a constant, you specify here the reference value σ_{ref}, which is applied at the reference temperature T_{ref}, while the actual value of the surface tension coefficient is calculated with use of the temperature coefficient σ_T, which is set by the **dSigma/dT** parameter (see below). The surface tension coefficient is calculated using the formula: $$\sigma = \sigma_{ref} - \sigma_T(T_{abs} - T_{ref}) = \sigma_{ref} - \sigma_T T$$
dSigma/dT	$\sigma_T = -d\sigma/dT$, the temperature coefficient of the surface tension, [N·m^{-1}·K^{-1}]. The **dSigma/dT** parameter can be set by a constant, a formula or a table. This parameter is available when **SurfTension, value** is set by a constant (see above).
Blackness	The blackness (emissivity) of the free inter-phase surface, or surface of particles or carcass
Mass transfer	Enable simulation of mass transfer in interaction of **Phases**. Possible options are: **Yes | No**. Simulation of mass transfer on the phase interface surface of continuous phases is actually simulation of ablation on the phase interface surface.
User specified source > ...	This group of parameters specifies a user-defined source of mass (on the inter-phase surface) of one of continuous **Phases**, presented in the phase interaction. This mass source influences the **VOF** variable and motion of the inter-phase surface. These parameters are available for **Continuum-continuum** and **Continuum-vacuum** phase interactions only.
User specified source > User specified source	This parameter specifies if the user-defined source of the mass flow on the inter-phase surface exists. Possible options are: **Yes | No**.
User specified source > Phase	The **Phase**, which user-defined source of the mass flow is specified. For a **Continuum-continuum** phase interaction you can select any of the **Phases** presented in the interaction. For a **Continuum-vacuum** phase interaction the only non-vacuum **Phase** is automatically selected as value of this parameter.
User specified source > Mass flux	Mass flow of the above specified **Phase**, [kg/(m^2 s)], generated on the inter-phase surface. It is specified as specific mass flow on the inter-phase surface per unit of surface [m^2] per unit of time [s].
Pressure	Pressure on the inter-phase surface. This parameter is only available if **Motion** is *not* set in one of the **Phases**. This **Pressure** is specified in relative units in those **Phase**, in which **Motion** is set.

Properties of the element "Phase interaction"	
Parameter*)	**Description**
Ext. heat exchange	This parameter enables heat transfer between a continuous **Phase** and **Vacuum** through a free surface. Possible options are: **Yes** | **No**.
Heat-transfer coef.	The heat exchange coefficient of the heat transfer to the external medium (from the continuous **Phase** to **Vacuum**). You can set this parameter if **Ext. heat exchange = Yes**.
T of external medium	Relative temperature of the external medium (of **Vacuum**), [K] You can set this parameter if **Ext. heat exchange = Yes**.
P gradient **Substance pair > ...** **SchmidtTurb** **C_kPrt** **C_ePrt** **Repulsion force** **Cw factor** **Dw** **Lift force** **Cl factor** **Dl** **Cd** **Nu** **Heat exchange coef. (D-C)** **Evaporation model** **Sh** **Is carrier phase**	Parameters of interaction between dispersed and continuous **Phases**. See details in section Theory> Physical processes> Processes in a dispersion medium > Parameters.
Cavitation > ...	Parameters for simulating the cavitation (available for **Continuum-vacuum** phase interaction). Simulating of cavitation is being beta-tested now, contact the Technical support service (support@flowvisioncfd.com) when required.
Cavitation > Cavitation model	The used cavitation model. Possible options: • **(none)** - cavitation is not simulated • **Model 1** • **Model 2**
Cavitation > Nucleation concentration	Cavitation nucleation concentration, [1/m^3]
Cavitation > Nucleation radius	Cavitation nucleation radius, [m]
Cavitation > Evaporative substance	Substance which fills cavitation bubbles
Cavitation > Vapour saturation pressure	Saturation pressure of the vapor which fills cavitation bubbles, absolute [Pa]
Cavitation > Vapour equilibrium density	Equilibrium density of the vapor, which fills cavitation bubbles, [kg/m^3]

*) Presence of specific parameters in properties of a **Phase interaction** element depends on combination of **Phases** (continuum-vacuum, continuum-continuous, continuum-particles, continuum-carcass) in the **Model**, and on physical processes specified in the **Phases**.

Folder "Models > Model #N > Init. data" and its subfolders «Init. data #N»



Element **Init. data #N**: a - in the project tree; b - it's **Properties** window

The folder **Models > Model #N > Init. data** contains subfolders **Init. data #N**, corresponding to the initial data used in the **Model**.

Initial data is a set of values of calculated variables at the initial time. The initial data may be specified as constants or formulas using coordinates, constant user variables and constants. The **Init. data #N** folder is used to specify parameters of **Initial conditions**. Scope of use of **Initial data** is defined in **Initial conditions**.

Specifying the initial data is useful to accelerate the convergence of the solution (for example, in simulations of external flow it is convenient to set the initial velocity field, which corresponds to the velocity of the free stream).

The **Properties** window of the folder **Init. data #N** contains the parameter **Name** only. You can replace this standard name (**Init. data #N**) by a custom mnemonic name as you wish.

Child elements in the folder **Models > Model #N > Init. data > Init. data #N** correspond to initial values of various variables and are generated automatically after loading phases into the model. Set of the variables corresponds to the set of physical processes defined for phases that were loaded into this computational model. In other words, for each equation, which is included in the computational model by enabling a physical process in the medium, the system automatically generates an element **Models > Model #N > Init. data > Init. data #N > *Initial value of a variable***.

Values of the variables are specified in the **Properties** windows of the corresponding elements ***Initial value of a variable***.

A **Model** has always to contain at least one set of initial data.

Originally the **Init. data** folder contains only the subfolder **Init. data #N** with zero values of variables.

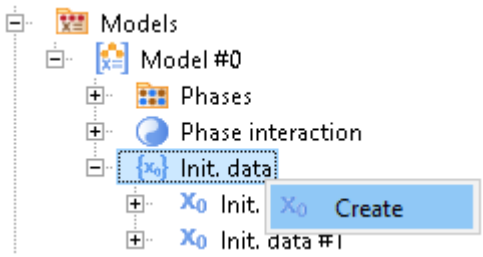
The variable **Phase volume** for the continuous phase is not displayed in the **Initial data**, it is not specified by the user but calculated by the formula:

$$\varphi_c = 1 - \Sigma\varphi_d$$

where $\Sigma\varphi_d$ is the sum of volumes of all dispersed phases presented in the **Model**.

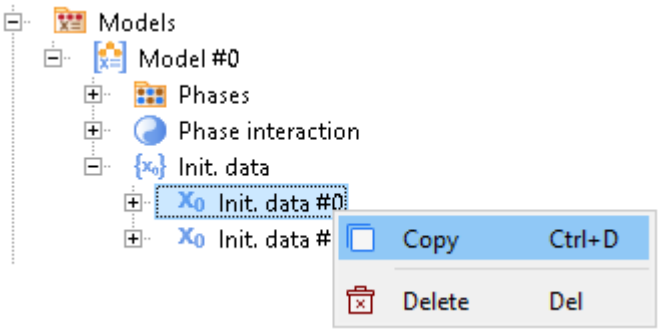
When chemistry is simulated, **Init. data #N** obtain child elements that correspond to variables **Mass frac.** not only for **Substances**, but also for **Elements** (conservative scalars). True mass fractions of **Substances**, which are selected as **Elements**, replaced in the user interface of *FlowVision* by mass fractions of **Elements**. In this situation you have to specify mass fractions of **Elements**. For example, in simulation of air dissociation, mass fraction of the **Oxygen Element** at the initial time moment in the computational domain is always to be equal **0.233333**, no matter on whether the computational domain contains some atomic oxygen or not. So, no matter if mass fraction of atomic oxygen is zero there or if it is not zero, mass fraction of the **Oxygen Element** is to be set equal **0.233333**.

Mass fractions of **Elements** on **Boundary conditions** are specified similarly.



The context menu of the folder **Init. data**

Context menu of the element "Init. data" in the project tree	
Command	**Description**
Create	Creates a new subfolder **Init. data #N**.



The context menu of the folder **Init. data #N**:

Context menu of the element "Init. data #N" in the project tree	
Command	**Description**
Copy	Creating an element, which is a copy of the selected element
Delete	Deleting the selected element from the project tree

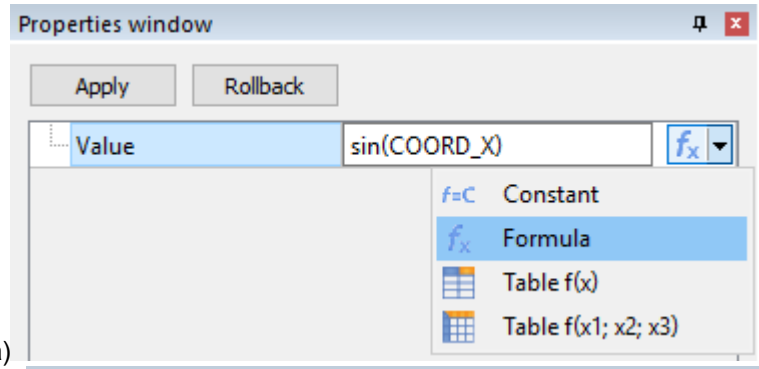
Elements «*Initial value of a variable*»

Elements **Models > Model #N > Init. data > Init. data #N > *Initial value of a variable*** have different forms for scalar and vector variables:

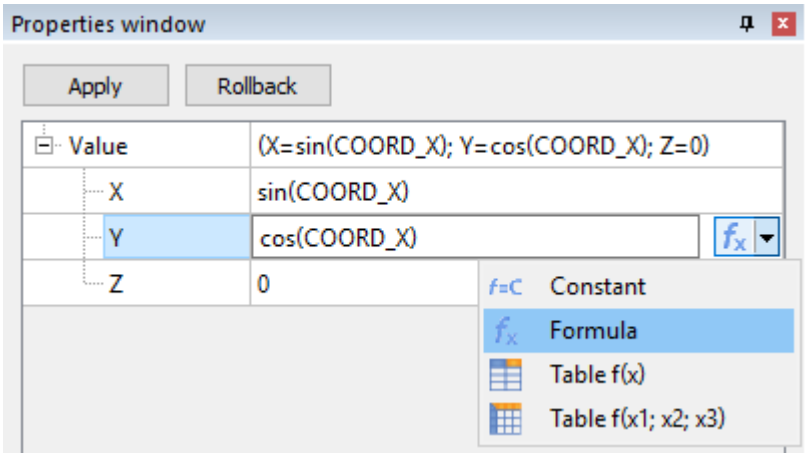
- scalar variables are marked with the icon and their **Properties** window contains the **Value** parameter only
- vector variables (for example, speed) are marked with the icon and their **Properties** window contains components of the vector (in the absolute coordinate system), which are displayed as parameters **Value > X**, **Value > Y**, and **Value > Z**.

Values of elements ***Initial value of a variable*** can be set by constants, formulae (using Formula editor) or tables (using Table editor). These values can be external parameters.

a)

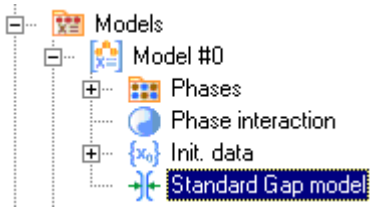


b)

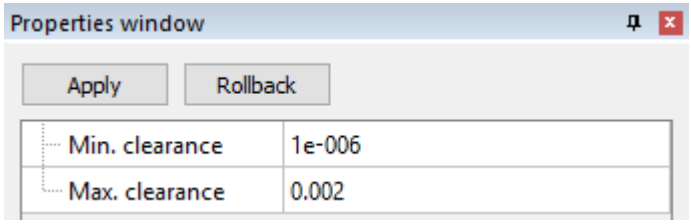


The **Properties** window of the element ***Initial value of a variable***:
a - for scalar variables, b - for vector variables

Element «Models > Model #N > Standard Gap model»



Element **Models > Model #N > Standard Gap model** contains parameters of the gap model specified by the parameter **Use Gap model** in the **Properties** window of the folder **Models > Model #N**.



The **Properties** window of the element **Standard Gap model**

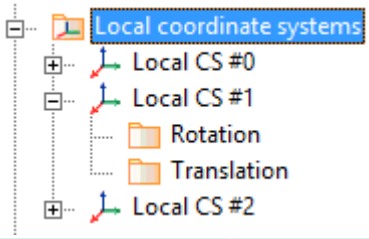
Parameters of **Standard Gap** model:

Parameter	Description
	Distance d_{min} and d_{max}, which specify use of the gap model in cells. The gap model is used in computational cells, where the distance d between the surfaces with different boundary conditions falls in the interval $d_{min} \le d \le d_{max}$.

Parameter	Description
Min. clearance	For all the cells of the gap, in which the distance between the gap-forming surfaces (d) is less than this value, the size of the gap will be limited to the minimum value ($d = d_{min}$).
Max. clearance	For all cells having two gap-forming surfaces, and distance between them (d) is less than this value, the gap model is applied.

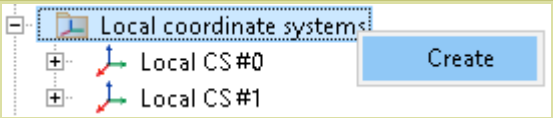
Note: A cell is a gap one if it meets to conditions specified in the section Theory > Physical processes > Processes in clearance > Gap cells.

8.1.8.3.6 Folder «Local coordinate systems»



The **Local coordinate systems** folder in the project tree

The folder **Local coordinate systems** contains subfolders **Local CS #N** (described in the next section).



Context menu of the folder «Local coordinate systems»

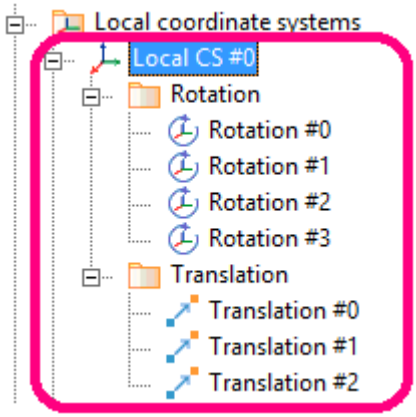
Menu item	Description
Create	Adding a new subfolder **Local CS #N**

Folder "Local CS #N"

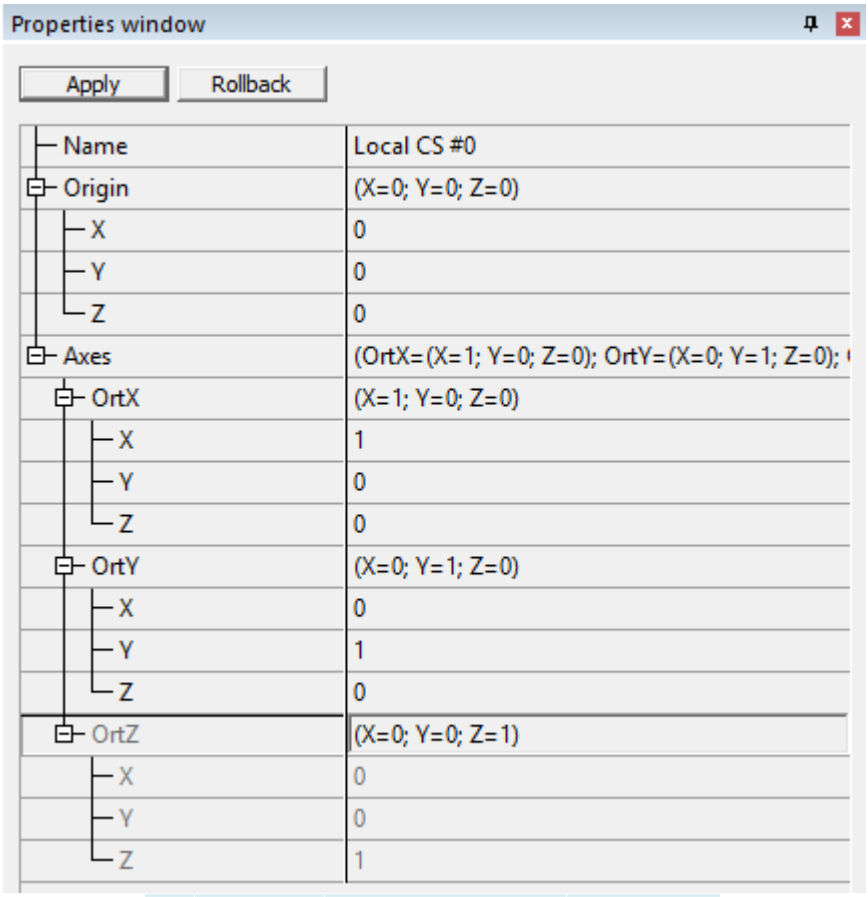
Elements (that are folders) **Local CS #N** locate in the folder **Local coordinate systems** and contain subfolders:

- **Rotation**, which contains elements **Rotation #N**
- **Translation**, which contains elements **Translation #N**

Folder **Local CS #N** has properties (defined in absolute coordinate system), which determine the point of origin and the orientation of the axes of the LCS-M, see section Movement local coordinate systems (LCS-M).

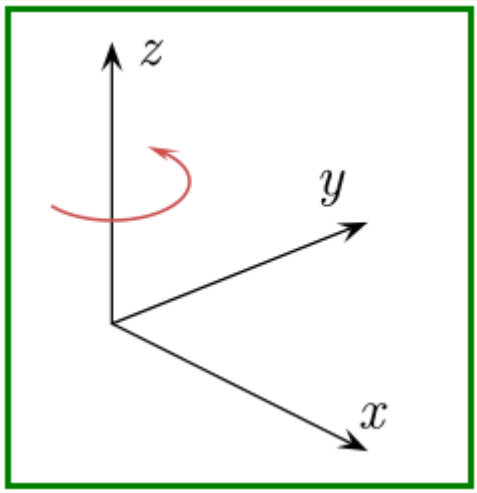


Folder **Local CS #N** in the project tree

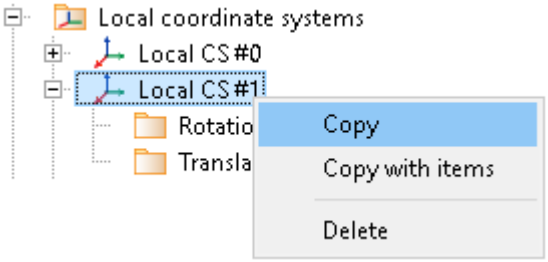


The **Properties** window of the folder **Local CS #N**

Properties of element "Local CS #N"	
Property	**Description**
Name	Name of LCS-M. Standard names are **Local CS #N** where N = 0, 1, 2, ... You can change this standard name if required.
Origin > X **Origin > Y** **Origin > Z**	Coordinates of origin of local CS, defined in the absolute CS.
Axes > OrtX > X	Orientation of the local CS axes relative to the absolute CS is set by projection of unit vectors along the axis of LCS to the axis of ACS. Orientation of the X and Y axes is set by the user as coordinates of vectors (not necessarily of unit length) along those axes on the coordinate axis of ACS, but in doing so *FlowVision checks the orthogonality of the X and Y axes and, if necessary, corrects the input* data. When data are entered for directions X and Y the program automatically corrects other axis (either Y or X respectively). Data entry for the Z axis in not provided (axis Z is oriented automatically based on data entered for axis X and Y). After every data input, *FlowVision* automatically normalizes it (replaces it with projections of unit vectors along the axes). Orientation of the Z axis is calculated automatically so as to be perpendicular to the X and Y axes and form a right normal orthogonal basis:

Properties of element "Local CS #N"	
Property	**Description**
Axes > OrtX > Y	 Right normal orthogonal basis
Axes > OrtX > Z	
Axes> OrtY > X	
Axes > OrtY > Y	
Axes > OrtY > Z	
Axes > OrtZ > X[*)]	
Axes > OrtZ > Y[*)]	
Axes > OrtZ > Z[*)]	

[*)]These parameters are computed automatically and can not be entered by the user.

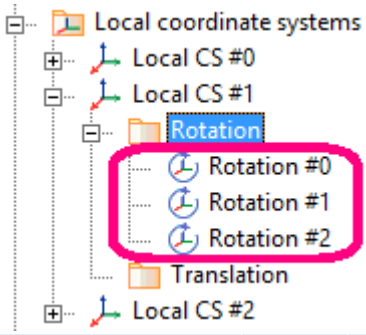


Context menu of the element **Local CS #N**

Context menus of items **Local CS #N**, in addition to standard commands include commands to add items **Rotational** and **Translational** motion.

Context menu of the item "Local CS #N"	
Menu item	**Description**
Copy	Copy item **Local CS #N** in the project tree.
Copy with items	Copy item **Local CS #N** in the project tree with the elements **Rotation #N** and **Translation #N**.
Delete	Deleting the selected element from the project tree

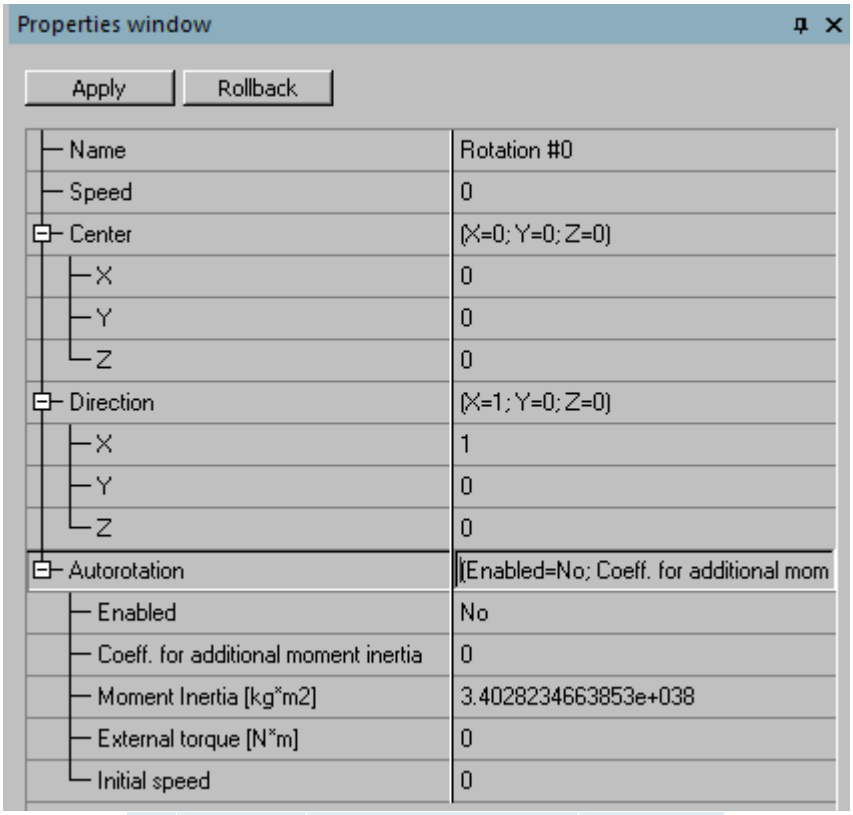
Folder "Rotation" and elements "Rotation #N"



Elements **Rotation #N** in the folder **Rotation** in the project tree

Rotation is the element, which allows you to specify the rotation of elements of the geometry. Rotation can be set in the **Region** or on a **Boundary conditions**.

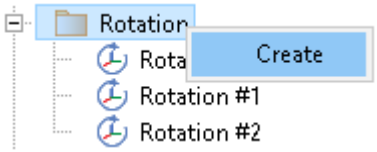
Element **Rotation #N** is a child element in the **Local coordinate systems > Local CS #N>Rotation** folder.



The **Properties** window of the element **Rotation #N**

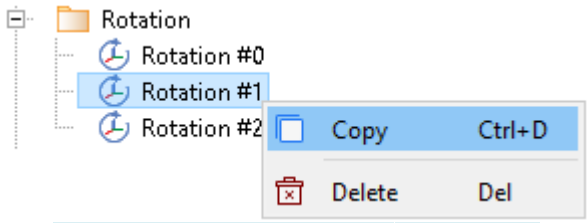
Properties of element "Rotation #N"		
Parameter	**Description**	**The dimension of**
Name	A user-defined name of the **Rotation**. You can change the standard name (**Rotation #N**, where N = 0, 1, 2, ...)	
Speed	Angular rotation speed (if **Autorotation** is disabled). When **Autorotation** is enabled, this parameter is not active and displayed with gray font.	[rad s^{-1}]
Center > X **Center > Y** **Center > Z**	Coordinates of the rotation center (a point on the rotation axis), set in the LCS-M.	[m]
Direction > X **Direction > Y** **Direction > Z**	Projections of the guiding vector of the rotation axis on the axis of the LCS-M.	[m]
Autorotation	This group of parameters is used for simulation of autorotation (when rotation of the rotor is influenced by the torque of the incoming flow and by the external torque). When the autorotation is enabled, the initial angular speed of the rotor is specified by the **Autorotation > Initial speed** parameter; then the angular speed of the rotor is calculated based on the differential equation: $$(J + \alpha J_{sub})\frac{d\omega}{dt} = T_{ext} + T_{hyd}$$ where ω is the angular speed of the rotor	

Properties of element "Rotation #N"		
Parameter	**Description**	**The dimension of**
	t is the time J is the moment of inertia of the rotor (user-defined value) α is the coefficient of the additional moment of inertia J_{sub}, which allows the user to specify the extent to which the program shall take into account the moment of inertia of the liquid/gas around the rotor. Set it within the range from 0 to 1. If you set α=0, the additional moment of inertia J_{sub} will not be taken into account. If you set to α=1, the additional moment of inertia is calculated as the moment of inertia of all liquid/gas in the subregion around the rotor. J_{sub} is the additional moment of inertia, which allows you to take into account the moment of inertia of the liquid or gas around the rotor. The additional moment of inertia J_{sub} is calculated as the volume integral within the subregion around the rotor of the density ρ of the liquid/gas around the rotor multiplied on the square of the distance r from the axis of rotation: $$J_{sub} = \int_{Volume} \rho r^2 dv$$ T_{ext} is the external torque (a user-defined value, for example, which arise because of frictional load or useful load) T_{hyd} is the hydraulic torque caused by the free-stream Recommendations: • for the stationary solution set J= 0 and α = 1 • to receive a non-stationary solution, specify J as the actual value of the moment of inertia of the rotor (available, for example, from the CAD-system, where the geometry has been prepared) and α=0.	
Autorotation > Enabled	This parameter specifies whether the autorotation is used. Possible options are: **Yes** | **No**.	
Autorotation > Coeff. for additional moment inertia	α, which is the coefficient of additional moment of inertia J_{sub} (see description above). It is specified in the range from 0 to 1 inclusive.	
Autorotation > Moment Inertia	J, which is the moment of inertia of the rotor	[kg m^{-2}]
Autorotation > External torque	T_{ext}, which is the external torque	[N m]
Autorotation > Initial speed	The initial angular speed of the rotor	[rad/s]



Context menu of the **Rotation** folder

The context menu of the folder "Rotation"	
Menu item	**Description**
Create	Adding into the folder **Rotation** a new element **Rotation #N**



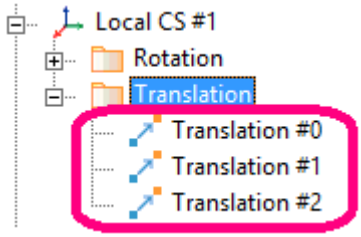
Context menu of the element **Rotation #N**

Context menu item "Rotation #N»	
Menu item	**Description**
Copy	Adding a new **Rotation #N** element, which is a copy of the selected element, into the **Rotation** folder
Delete	Deleting the selected element from the project tree

Rotation of a **Boundary condition** is specified independently from rotation of the **Region**.

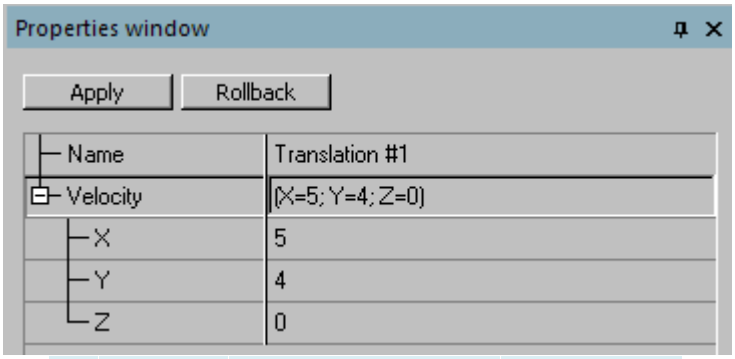
Initial conditions are specified in the absolute coordinate system only.

Folder "Translation" and elements "Translation #N"



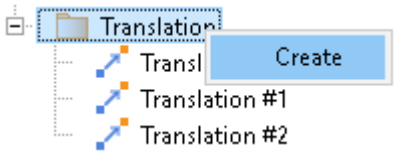
Elements **Translation #N** in the folder **Translation** in the project tree

Elements **Translation #N** are child elements in the folder **Local coordinate systems > Local CS #N > Translation**.



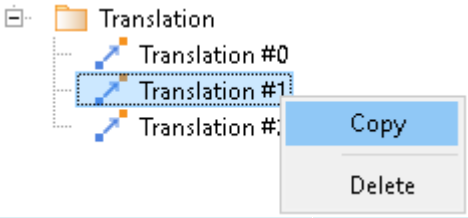
The **Properties** window of the element **Translation #N**

Item Options "Translational motion #N»	
Property	**Description**
Name	Name of translation movement. You can change the standard name (**Translation #N**, where N = 0, 1, 2, ...)
Velocity > X	Components of translation velocity, set in the LCS-M. Can be set as a constant or as a formula or as a table.
Velocity > Y	
Velocity > Z	



Context menu of the folder **Translation**

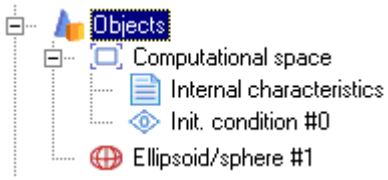
Context menu of the folder "Translation"	
Menu item	**Description**
Create	Adding a new element **Translation #N** into the folder **Translation**



Context menu of the element **Translation #N**

Context menu of the element "Translation #N»	
Menu item	**Description**
Copy	Adding an element, which is a copy of the selected element, into the **Translational** subfolder
Delete	Deleting the selected element from the project tree

8.1.8.3.7 Folder «Objects»



In the **Preprocessor** tree, new geometric objects are created in the **Objects** folder.

These objects can then be used in any of the subareas Calculated for setting them in the initial conditions, the adaptation computational grid, creating layers, etc.

Types of geometric objects and their properties window in the **Preprocessor** and **Postprocessor** the same, but the objects themselves are different.

The following folders and elements of the project tree are described below in this section:

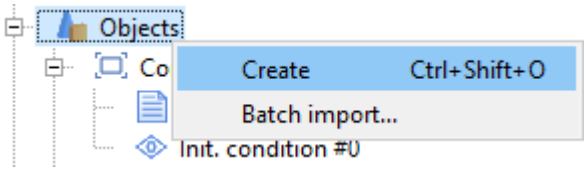
- the **Objects** folder
- the **Objects > Computational space** folder
- the **Objects > Object #N** folder
- the **Computational space** element

When you create a new project folder **Objects** contains only the object **Computational space**.

Folder **Objects** contains subfolders:

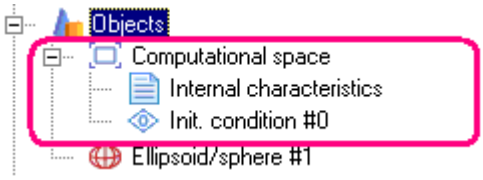
- **Computational space**
- ***(geometric object)***

The context menu of a folder «Objects» (loading geometry import objects from files)



Menu item	Description
Create	Creating a new **Object** (creating a standard geometric object, loading an imported object from a file or creating a **Supergroup**). This command is duplicated by the **Ctrl+Shift+O** hot key.
Batch import	Loading multiple **Imported objects** from several files (this command saves time and effort the user). This opens the standard operating system's window for access to files, select there the files you want, hold down the **Ctrl** key on your computer keyboard and click the button to read files.

Folder «Objects > Computational space»

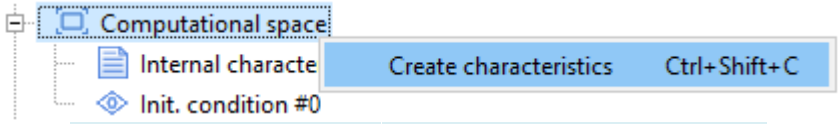


Folder **Objects > Computational space** in the project tree

Folder **Objects > Computational space** is automatically created with the project and contains the characteristics and initial conditions are specified for the entire computational domain.

Folder **Objects > Computational space** may contain elements:

- **Internal characteristics**
- **Characteristics**
- **Init. condition #N** (that is a duplicate of the same-name element in the «Initial conditions» folder)



Context menu of the folder **Objects >Computational space**

The context menu of the folder **Objects > Computational space**:

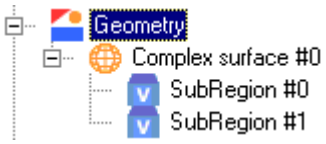
Menu item	Description
Create characteristics	Creating a **Characteristics** object on the **Computational space**. This command is duplicated by the **Ctrl+Shift+C** hot key (by default; you can change this hot key).

Folders «Objects > *(object)*»

Folders **Objects > *(object)*** contain representations of geometric objects (each folder represents one **Object**).

Objects includes two tabs at once in the project tree, **Preprocessor** and **Postprocessor**, so their description is given in a separate section, Objects in the Project Tree.

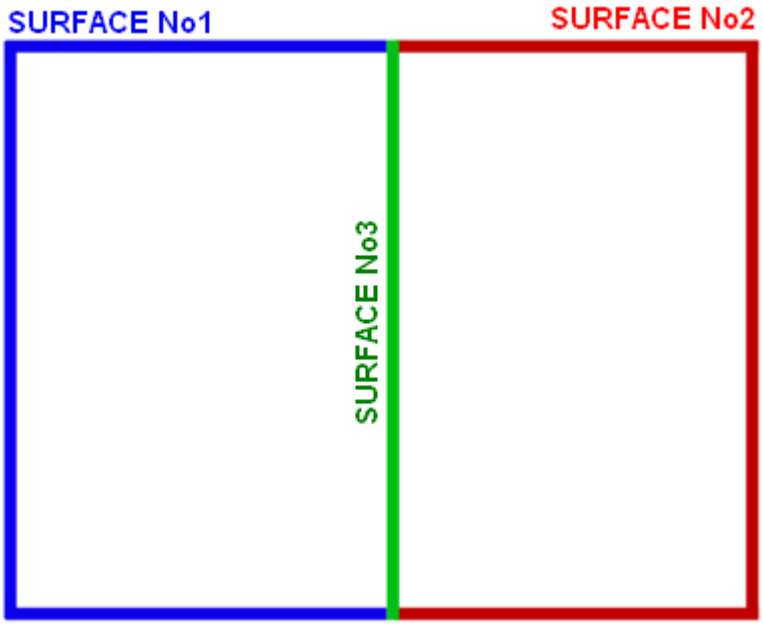
8.1.8.3.8 Folder «Geometry»



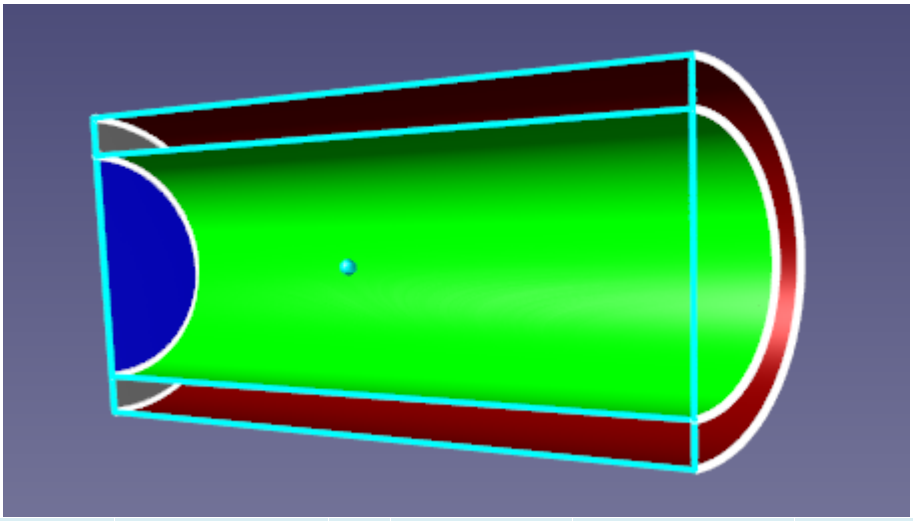
Folder **Region > Geometry** in the project tree structure contains subregions using a multiconnection geometry. (Multiconnection is connection of three or more surfaces at least one common edge, a particular case of multiconnection is T-connection of surfaces.)

In the project tree, the surfaces, which form one or several subregions, are presented as **Complex surfaces**, which, in turn, consist of simple surfaces (groups of facets), see illustrations. A typical particular case should be noted, when a **Complex surface** consists of a single surface, which limits, with no multiconnection, some **Subregion**.

In the project tree the **Complex surfaces** folder contains child elements corresponding to their adjacent **Subregions**.



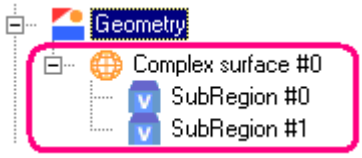
A **Complex surface** consists of several surfaces (groups of facets)



Example: a **Complex surface** with multiconnection (T-joint) forms two **Subregions**

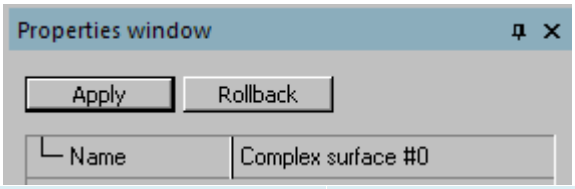
The **Region > Geometry** folder does not contain parameters (its **Properties** window is empty) and has no context menu.

Subfolders "Complex surface #N"



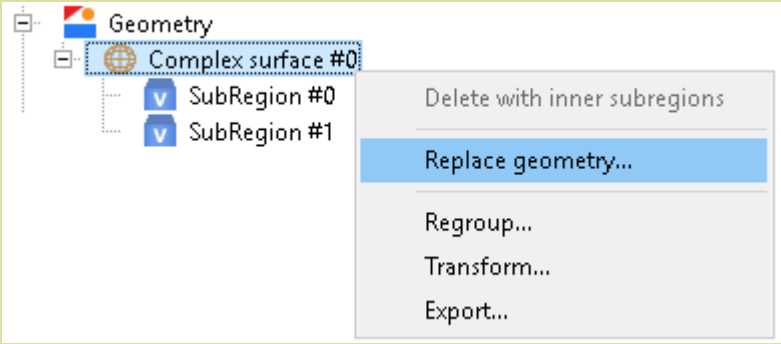
Folder "**Complex surface #N**" in the project tree

Subfolders **Complex surface #N** correspond to individual complex surfaces.



Properties window of subfolder **"Complex surface #N"**

Parameters element "Complex surface #N"	
Parameter	**Description**
Name	Name of the complex surface. If desired, you can change its default name of **Complex surface #N** to another.



Context menu of subfolder "Complex surface #N" in the project tree

Menu item	Description
Delete with inner subregions	Remove the **Complex surface** with its internal **Subregions**.
Replace geometry	Remove the **Complex surface** with its internal **Subregions**, then build in a new geometry, which is loaded from a file.
Regroup	Regroup the **Complex surface**.
Transform	Transform the **Complex surface**.
Export	Export the **Complex surface** into a file.

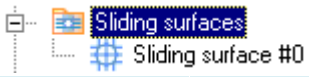
Elements "Complex surface #N > SubRegion #M"

Child elements **Complex surface #N > SubRegion #M** correspond to subregions, limited by appropriate **Complex surfaces** (so these elements correspond to one or several internal **Subregion(s)** and, might be, one external **Subregion**).

The **Properties** windows of elements **SubRegion #M** are empty, for viewing and/or changing parameters of the **Subregions** use **Properties** windows of elements **Subregions > Subregion #M**.

Context menus of elements **Complex surface #N > SubRegion #M** contain the same commands as context menus of elements **Subregions > Subregion #M**.

8.1.8.3.9 Folder «Sliding surfaces»



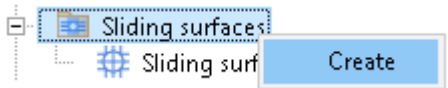
Folder **Sliding surfaces** in the project tree

Elements of this folder in the project tree present *sliding surfaces*, which are special cases of connected boundary conditions, see section Sliding surface.

Their standard names are **Sliding surface #N**, and colors of their icons depend on whether the appropriate **Sliding surface** is built into the geometry (in faint colors when is not built into the geometry).

You can find a step-by-step example of work with sliding surfaces in the subsection *"Specifying and using a sliding surface"* in Operations with boundary conditions.

The **Properties** window of the folder **Sliding surfaces** is empty.

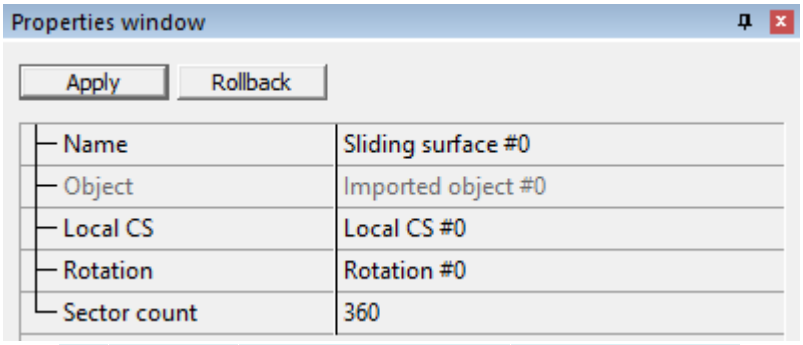


Context menu of the folder **Sliding surfaces**

The context menu of the folder **Sliding surfaces** contains commands:

The menu command	Description
Create	Adding a new element **Sliding surface #N** to the **Sliding surfaces** folder

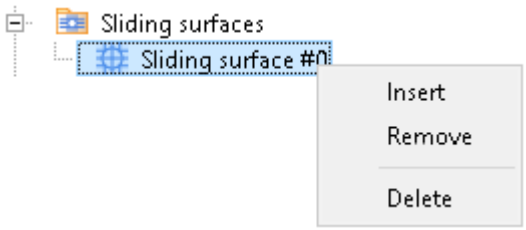
Parameters of the element «Sliding surface #N»



The **Properties** window of the element **Sliding surface #N**

Parameters of the element **Sliding surface #N**:

Parameter	Description
Name	Name of the sliding surface, the default is set to **Sliding surface #N**
Object	Object on which was built the sliding surface. This field is for information only, its contents can not be changed here.
Local CS	The local coordinate system, which was built by the sliding surface
Rotation	Rotation in the local coordinate system applied to the sliding surface
Sector count	Number of sectors of the sliding surface



Context menu of the element **Sliding surface #N**

The context menu of the **Sliding surface #N** element contains the following commands:

Menu command	Description
Insert[*]	Split the **Subregion**, where the **Sliding surface #N** locates, in two **Subregions**, separated by the **Sliding surface #N**. A **Sliding surface** can be only inserted into a **Subregion** if in the properties of the **Sliding surface** its local coordinate system and rotation are specified. Otherwise, this command is inactive.
Remove[**]	Reconnect separated **Subregions**, removing the sliding surface, which corresponds to the element **Sliding surface #N**.
Delete[*]	Deleting the element **Sliding surface #N**

[*] When the sliding surface is already inserted in the geometry (dividing the subregion into two subregions), the **Insert** and **Delete** commands are inactive.

[**] The **Remove** command is only active when the sliding surface is inserted into the geometry

8.1.8.3.10 Folder «Characteristics»

The folder **Characteristics** is located in two tabs of the **Project** window: **Preprocessor** and **Postprocessor**. *Characteristics* contain information about a set of *integral values*, which have been calculated on some selected **Object**.

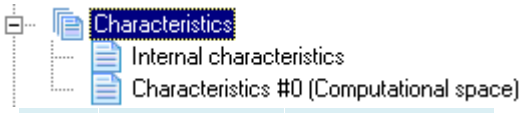
The information is displayed in the **Info** window, and can be written into a text `glo`-file.

Set of the values depends on the base **Object**.

Links to sections with descriptions of some specifics:

- Specifics of calculating Characteristics on a Plane or on another surface
- Specifics of calculating Characteristics by the variable VOF
- The Info window for Characteristics
- Components of a text file for recording data from Characteristics

See also: section Characteristics.



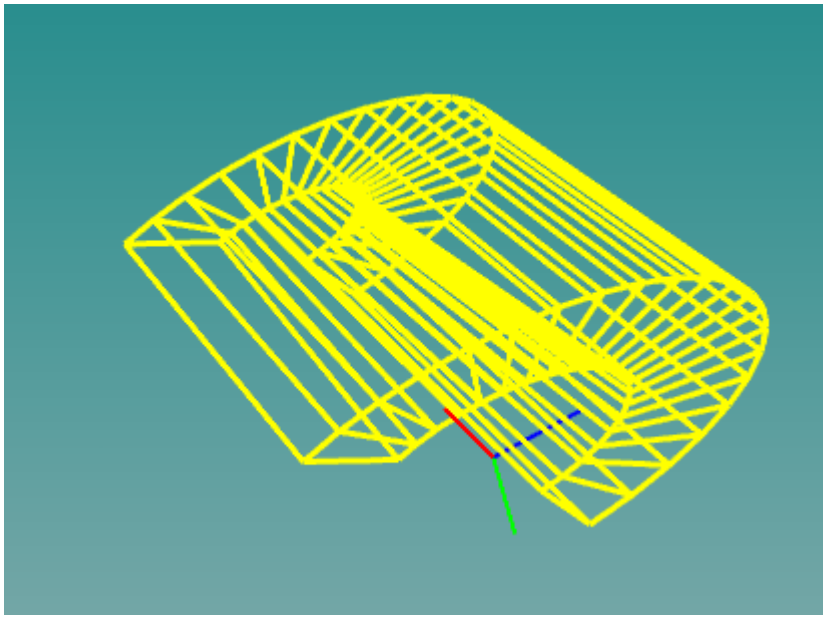
Folder **Characteristics** in the project tree

The **Characteristics** folder contains:

- The **Internal characteristics** element, which contains information about the current time, the time step number, values of the time step and the explicit time step, the reference temperature, the reference pressure, components of the **Gravity vector**, etc. The **Internal characteristics** element is always presented in the **Computational space** geometry object.
- Elements **Characteristics #N** that represent various integral values calculated for the corresponding **Objects**. These elements are created by the **Create** command of the context menu of the **Characteristics** folder.

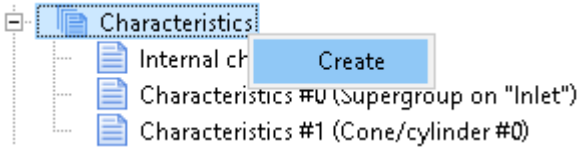
If an element **Characteristics #N** has been created in **Preprocessor**, you can see all its parameters in **Postprocessor**, but you cannot change there the parameters that was set in **Preprocessor**.

Geometry Objects	Type of variable
Computational space Plane Box Cone/cylinder Ellipsoid/sphere Imported object Supergroup Set of sensors	scalar vector, the characteristic can be calculated by: • scalar product of the vector variable and vectors of elementary sites • absolute length of a vector variable • values of any of the components of the vector variable



Characteristics can be set on the individual surfaces of an **Object**
(for example, substrates and the lateral surface or surfaces of the channel sections with the channel of the cone sector)

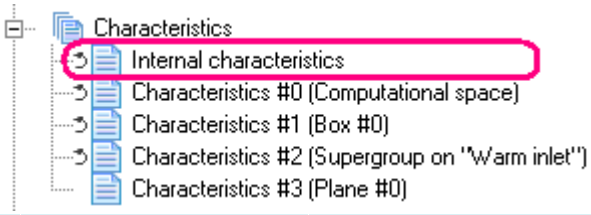
Context menu of the "Characteristics" folder



Context menu of the **Characteristics** folder

Menu item	Description
Create	Creating a new element **Characteristics #N**

Element «Internal characteristics»



The **Internal characteristics** element in the project tree

The **Internal characteristics** element is associated to the the **Computational space** geometry object.

The values, which are calculated in **Internal characteristics**, are predefined in the program and cannot be changed; these values are:

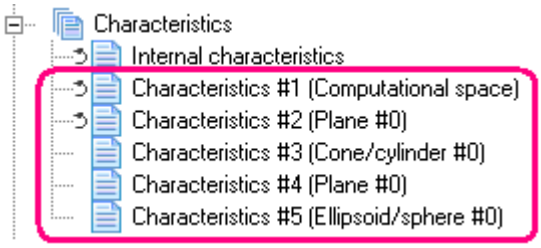
- the time step number
- the current time
- duration of the time step
- duration of the explicit time step
- etc.

Particularly, **Internal characteristics** contain all data that you can be displayed in the upper table from the **Status** tab of the **Monitor** window.

The **Internal characteristics** element has no context menu.

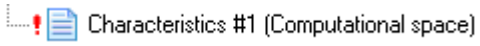
Elements "Characteristics #N (*Object's name*)"

Elements **Characteristics #N** are child elements in the folder **Characteristics** and in the folder of those **Objects**, on which the characteristics are calculated.



Elements **Characteristics #N** in the project tree

If the **Variable** parameter in properties of **Characteristics #N** is not set, it will be marked in the project tree by a symbol "**!**":



The numerical values of components of an element **Characteristics #N** are displayed in the **Info** window and are updated by the results of the calculation at each time step.

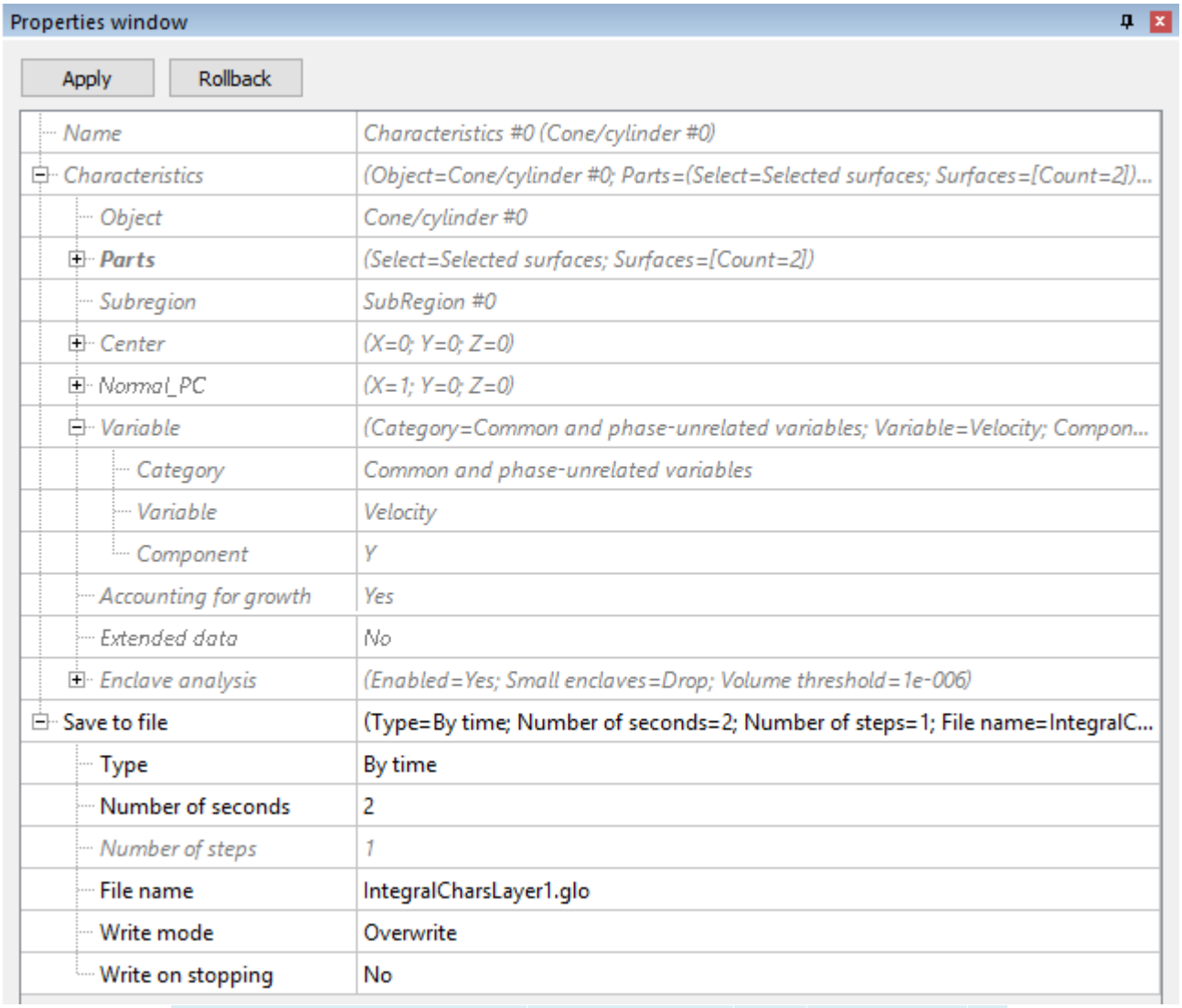
The parameters in the **Properties** window of the element **Characteristics #N** displayed in the tabs **Preprocessor** and **Postprocessor**, differ by the presence of a group of parameters **Saving a file** (only available in the tab **Postprocessor**).

Properties window

Apply | Rollback

Name	Characteristics #0 (Cone/cylinder #0)
Object	Cone/cylinder #0
Parts	(Select=Selected surfaces; Surfaces=[Count=2])
Select	Selected surfaces
Surfaces	[Count=2]
Lateral surface	Yes
Bottom base	No
Subregion	SubRegion #0
Center	(X=0; Y=0; Z=0)
X	0
Y	0
Z	0
Normal_PC	(X=1; Y=0; Z=0)
X	1
Y	0
Z	0
Variable	(Category=Common and phase-unrelated variables; Variable=Velocity; Compon...
Category	Common and phase-unrelated variables
Variable	Velocity
Component	Y
Accounting for growth	No
Extended data	No
Enclave analysis	(Enabled=Yes; Small enclaves=Drop; Volume threshold=1e-006)
Enabled	Yes
Small enclaves	Drop
Volume threshold	1e-006

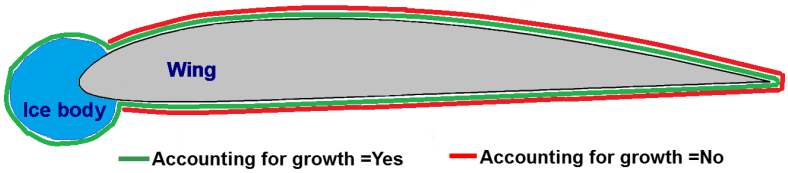
Properties window of the element **Characteristics #N** in the **Preprocessor** tab (the characteristic is built on one of the surfaces of a **Cone**)



Properties window of the element **Characteristics #N** in the **Postprocessor** tab

Parameters of the element "Characteristics #N *(Object's name)*"	
Parameter	**Description**
Name	Name of the **Characteristics**. Optionally, you can specify a different name for the item instead of the standard name "**Characteristics #N *(Object's name)***"
Object	Name of the **Object** that is associated with the characteristic. Field information is not editable.
Parts > Select	The domain of the **Characteristics**. Possible options depend on which **object** is computed **features**. In particular, the possible options: • **Volume** means the volume of the **Object** • **Whole surface** means the whole surface of the **Object** • **Selected surfaces** means some selected surfaces the **Object** (eg, faces of a box or the base of a cylinder) • **Whole plane***) - This option is the calculation of **Characteristics** on the whole **plane**, identified as the **object** on which to evaluate the **features**, within the **Subregion**, to which the P1 point of the **Plane** falls. • **Selected contour***) - This option is only calculation **Characteristics** on the contour **of** the **plane defined** as **an object** that encloses the P1 point of the **Plane** obtained from

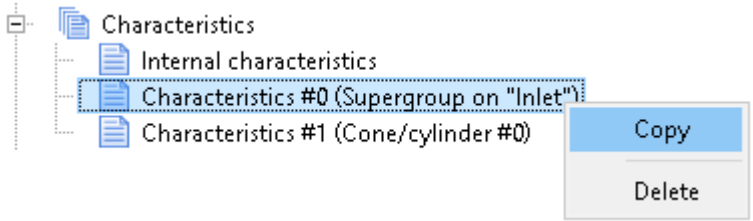
Parameters of the element "Characteristics #N *(Object's name)*"	
Parameter	**Description**
	the **Reference point** P0 set the **Shift.** This option is similar to the same parameter in the general properties of the **Layers** (see Illustration in the section Layer «Cell set»).
Parts > Surfaces > X- **Parts > Surfaces > X+** **Parts > Surfaces > Y-** **Parts > Surfaces > Y+** **Parts > Surfaces > Z-** **Parts > Surfaces > Z+**	Calculation of the characteristics specified in the faces of the parallelepiped. For each face, you can specify: • **Yes** - calculate the variable on the said faces • **No** - do not calculate the variable on the said faces (these parameter can be set when **Select = Selected surfaces**)
Parts > Surfaces > Lateral surface **Parts > Surfaces > Bottom base** **Parts > Surfaces > Top base** **Parts > Surfaces> Channel** **Parts > Surfaces > First-section** **Parts > Surfaces > Second-section**	Calculation of the characteristics specified in the surface of the **Cone/cylinder** (and/or the surfaces and edges of the channel obtained from the edges of the sector of a **Cone/cylinder**, see the illustration). For each surface, you can specify: • **Yes** - calculate the variable on said surfaces • **No** - do not calculate the variable on said surfaces (these parameter can be set when **Select = Selected surfaces**)
Subregion[*)]	A **Subregion**, in which the **Characteristics** is calculated. (The value is selected from a drop-down list.) This parameter is set in the surface characteristics, calculated *on* a *curved surface* (not on a **Plane**).
Center > X **Center > Y** **Center > Z**	Coordinates of a point (in the absolute coordinate system), relating which torque of the force acting on the liquid by the **Object**'s surface(s) is calculated. Also parameters **Center > ...** are used to define the plane, which cuts the section where the sought **Center of pressure** locates.
Normal_PC > X **Normal_PC > Y** **Normal_PC > Z**	Normal vector of the plane, in which section the sought **Center of pressure** locates. These parameters are available and make sense only for **Characteristics** that are built on: • **Boxes** • **Cones/cylinders** • **Ellipsoids/spheres** • geometrically closed **Supergroups** • **Imported objects**, on which **Moving bodies** are set
Variable	Variable, which is calculated on or in an object.
Variable > Category	Selection of a category for the **Variable**. Possible options are: • **Common and phase-unrelated variables** • **Variables of phase "Phase #N"** • **User variables** See details in the section Categories of variables.
Variable > Variable	The **Variable**, which is selected from the drop-down list of variables of the selected **Category**
Variable > Component	This parameter defines use of the **Variable** for calculation of characteristics. The possible options are: • **Vector**: the program integrates scalar product of the vector of the **Variable** and the normal vector to an elementary area of the surface, over which the integrating is done. Actually, this is the normal (to the surface)component of the vector **Variable**. When the integrating is done over a volume, this option is unavailable.

Parameters of the element "Characteristics #N *(Object's name)*"	
Parameter	**Description**
	This option is unavailable for some geometric objects, if so, if you need, create appropriate **Imported objects** by the command **Copy as imported object**. • **Length**: the absolute magnitude of the vector **Variable** (the length of the vector) is used. • **X**: the component of the vector **Variable** along the X axis is used. • **Y**: the component of the vector **Variable** along the Y axis is used. • **Z**: the component of the vector **Variable** along the Z axis is used.
Accounting for growth	The **Accounting for growth** parameter has influence on calculating **Characteristics** when icing is simulated. Possible options are: • **Yes**: the **Characteristics** will be calculated over the whole surface *including the ice-covered areas*, over the outer surface of the ice body. • **No**: the **Characteristics** will be calculated over those areas of the surface only that are *not covered by ice*. This parameter is available for those **Subregions** only, in which the specified **Model** includes the "continuous-continuous" phase interaction. 
Extended data	This parameter turns on/off calculating additional components of **Characteristics**, see details in the section The Info window for Characteristics. Possible options are: **Yes** | **No**.
Enclave analysis	Enclaves of the **Subregion** are included in calculation of the characteristics. *Enclaves* are isolated volumes of the **Subregion** generated due to presence of a **Moving body** in the **Subregion**.
Enclave analysis > Enabled	Enable calculation of the characteristics in enclaves. Possible options are: • **Yes**: the variable is calculated not only for the whole **Subregion**, but also separately for individual enclaves. • **No**: the variable is only calculated for the entire subregion
Enclave analysis > Small enclaves	Small enclaves have volume that is less then **Volume threshold** (see below). Possible values of this parameter are: • **Keep**: the variable is calculated in the enclaves of any size. • **Drop**: the characteristics will be calculates in those enclaves only, which have amount greater than **Volume threshold**.
Enclave analysis > Volume threshold	The threshold value of volume of enclaves, [m^3]
Save to file	Parameters controlling the storage of data into a text file (with extension `glo`). This group of parameters is only available in the **Postprocessor** tab.
Save to file > Type	Type of saving the data into a file: • **Disabled**: no data are recorded

Parameters of the element "Characteristics #N *(Object's name)*"	
Parameter	**Description**
	• **Automatic**: the data are recorded on each time step • **By time**: the data are recorded after a specified time interval • **By step**: the data are recorded after a specified number of steps Regardless of the value of this parameter, the program will write into the file at stopping the computation, if **Save to file > Write mode = Overwrite** and **Save to file > Write on stopping = Yes** are selected (see below).
Save to file > Number of seconds	Period of time, after which there is a record in the data file (used when **Type = By time**)
Save to file > Number of steps	The number of time steps, through which there is a record in the data file (used when **Type = By step**)
Save to file > File name	Name of the text filein the server part of the project, in which data is stored. The user is prompted formed the default file name with extension `glo`. If desired, you can change the name and file extension.
Save to file > Write mode	Selection the recording mode: • **Overwrite**: overwrite the file each time you save the data in the file contains only the most recent data and old data is overwritten (the default mode). • **Append**: at each record are appended to the file, all data are stored.
Save to file > Write on stopping	This parameter instruct the program to make an additional writing into the file after triggering the specified **Stopping conditions** or manual stopping the computation. The program makes writing into the file at each stopping of the project regardless of the value of the **Save to file > Type** parameter. Possible options: **Yes** | **No**. This parameter is available when **Save to file > Type mode = Overwrite**.

*) Surface characteristics are *always* considered for one **Subregion**. For curved surfaces, the subregion is set manually by the user in the parameter **Subregion**. For **Characteristics** on a **Plane** the subregion is automatically selected by location of the point P1, obtained from the **Reference point** of the **Plane** by applying the **Shift**.

Context menu of the element "Characteristics #N"



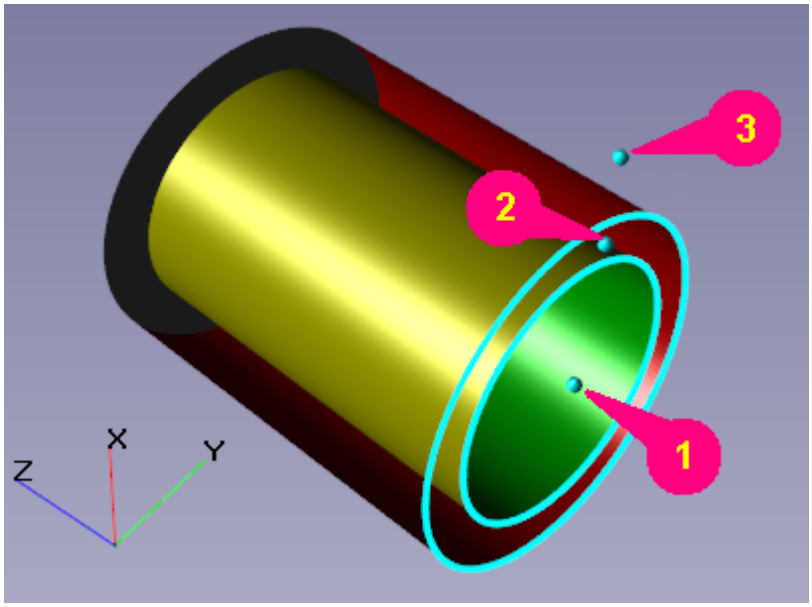
Context menu of the element **Characteristics #N** in the project tree

Context menu of the element "Characteristics #N" in the project tree	
Menu item	**Description**
Copy	Creating an item as a copy of the selected item
Delete	Removing the item from the project tree

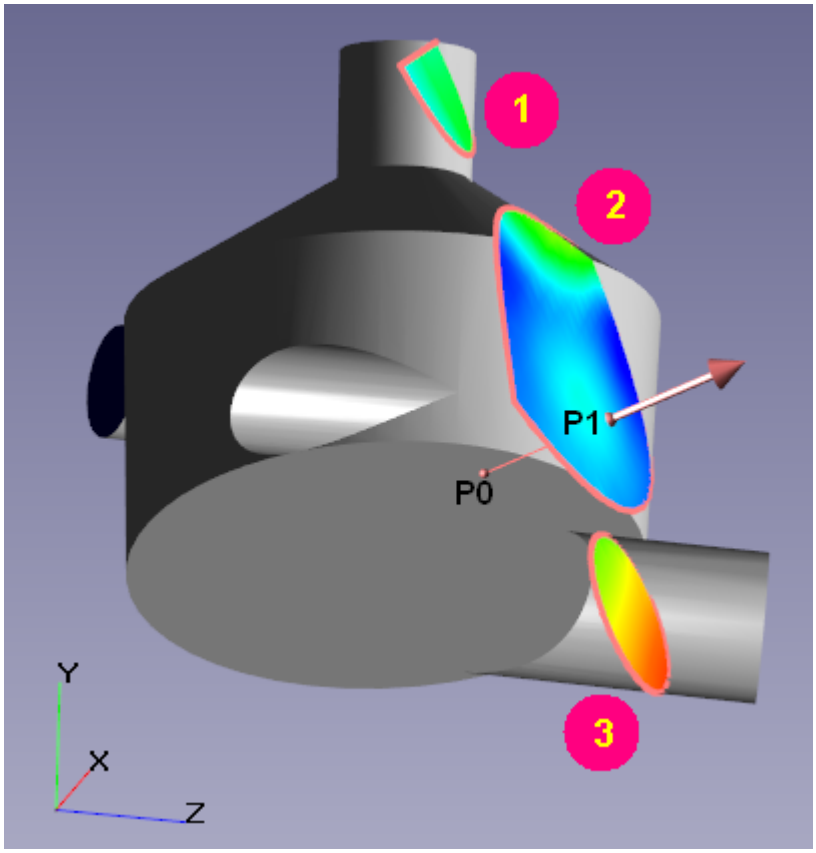
8.1.8.3.10.1 Specifics of calculating Characteristics on a Plane or on another surface

Characteristics on a **Plane**, as well as other surface characteristics, are calculated in only one **Subregion**.

Which of **Subregions** will be used, is determined by position of the **Plane**'s *P1 point*, which is obtained from the **Plane**'s **Reference point P0** by the specified **Shift** (see section Object "plane" (interface description)).



Characteristics on a **Plane** are calculated for those **Subregion**, into which the **P1** point falls. The illustration above shows a **Plane**, which orthogonally intersects a thick metal tube with flow inside and outside the tube, with three variants (1, 2, or 3) of location of the point **P1**. Depending on the location of the point **P1**, **Characteristics**, which are defined on the **Plane**, will be calculated in one of three different **Subregions** (1: for the flow inside the tube; 2: for the tube itself; 3: for the flow outside the tube).



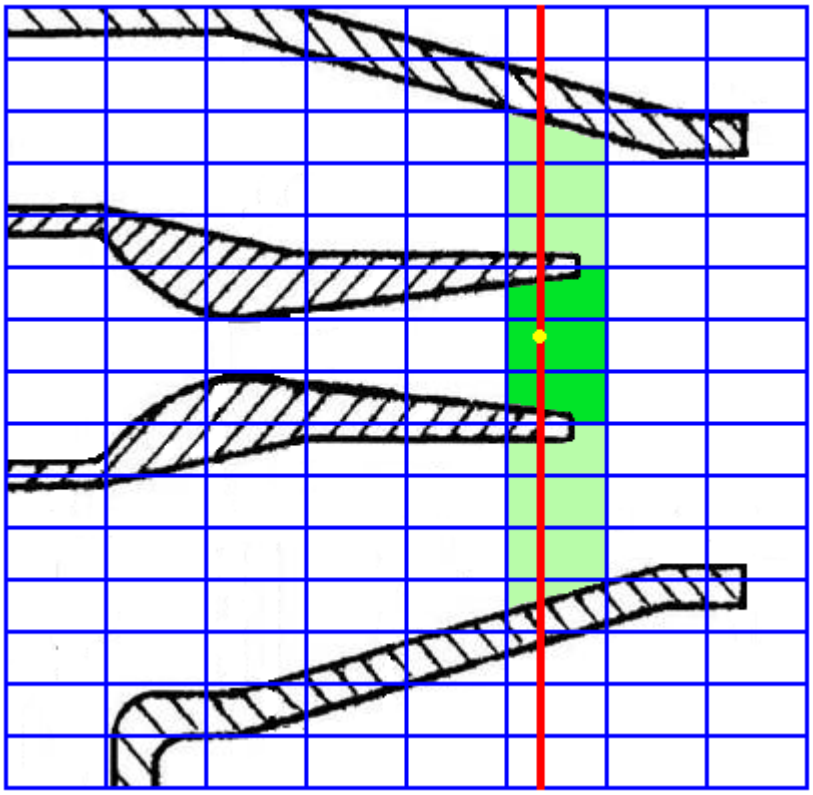
When **Characteristics** are calculated on a **Plane**, the **Subregion** is selected according to position of the **Plane**'s point **P1**. On the illustration the point P1 falls into the mixer, so the **Characteristics** will be calculated for the **Subregion**, which locates inside the mixer. If you specify **Parts > Select =Whole plane**, then the **Characteristics** will be calculated on *all three* contours of intersection between the **Plane** and the **Subregion** (these are contours 1, 2 and 3). If you specify **Parts > Select = Selected contour**, then the **Characteristics** will be calculated on the contour 2 *only*, as the **Plane**'s point **P1** falls into this contour.

Specifics of forming sections near edges

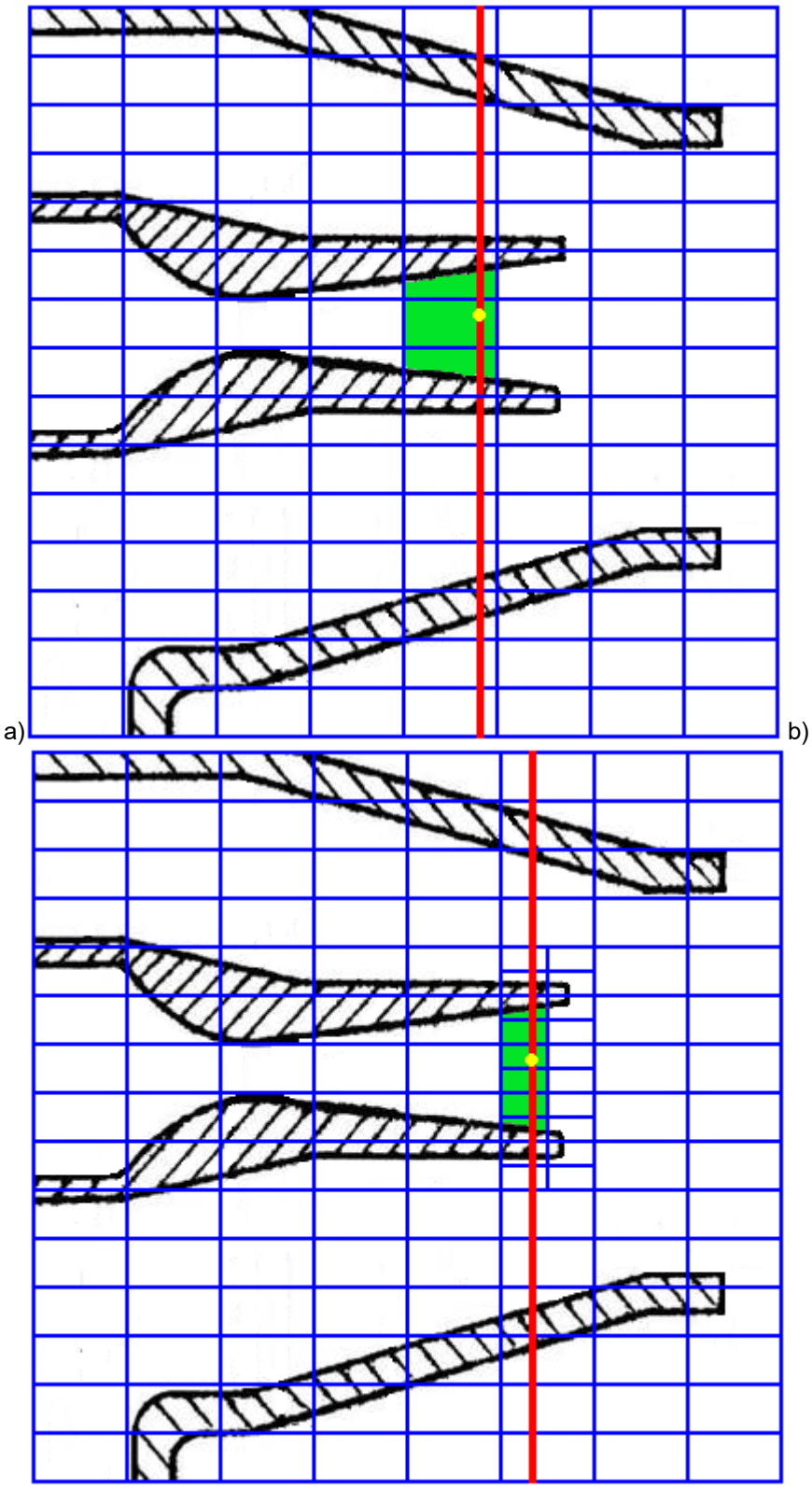
Forming a **Subregion**'s section, on which **Characteristics** are calculated, has specifics near edges. If you don't take these specifics into account, you might have unexpected results.

Attempting to build a section, the program recurrently includes in it all adjacent cells, through which the intersecting surface goes. And, even if the intersecting surface falls on the boundary of the**Subregion**, the forming section might go around obstacles through adjacent cells and spread much wider then the contour, on which you wish to calculate the **Characteristics**.

On the illustration below you can see as a section (dark green and light green cells), which is formed near an edge, differs from the section, which the user needs (dark green cells only). On the next illustrations you can see how you can fix this undesirable situation.



An example of forming a surface, on which **Characteristics** will be calculated, near an edge. The forming section includes dark green and light green cells. The section, which the user needs, includes dark green cells only. The intersecting **Plane** is shown as a red line, and its point **P1** is colored in yellow.



How you can form the desired section: *a)* move the intersecting surface away from the edge; *b)* split the cells near the edge using an **Adaptation**.

Specifics of calculation of characteristics by the total pressure on boundary conditions

When **Characteristics** are calculated by **Total pressure** on a **Supergroup**, which is built on a **Boundary condition**, the **Total pressure** values are taken not from the surface of this **Boundary condition** but from the nearest to it (adjacent) cells of the computational grid.

8.1.8.3.10.2 Specifics of calculating Characteristics by the variable VOF

The variable **VOF** can be only a variable of some **Phase #N** and it is always included into the category **Variables of phase "Phase #N"**.

When calculating an integral of phase variables (except **VOF**) the domain of integration is formed by the cells, which contain the selected **Phase #N**. But for the variable **VOF** the domain of integration is formed by computational cells of the whole **Model** in the **Subregion**.

For the variable **VOF**, the **Mass** value is calculated for those only **Phase**, to which the variable **VOF** belongs. When **Mass** is calculated, during the volume integrating, in each cell the density ρ is multiplied on the value of **VOF** (denoted by F), which is zero in the cells where the **Phase** is absent:

$$m = \iiint_V \rho F \, dV$$

Therefore in the **Info** window for **Characteristics**, which has been built by the variable **VOF**, the **Mass** value is calculated for the selected **Phase #N** only, while **Volume** is displayed for all computational cells of of the **Subregion**.

For any variable, except **VOF**, **Volume** will be calculated similarly as it is done for **Mass**.

Examples of calculation **Volume** and **Mass** for different categories of variables:

- if **Characteristics** are built by the variable **VOF** of some **Phase #N**:
  - **Volume** is calculated over all computational cells of the **Subregion** (non-computational cells belong to the **Vacuum** phase or are located inside a **Moving body**). So, the calculation of **Volume** includes all computational cells of the **Subregion**, independently on their **Phases**.
  - **Mass** is calculated in those cells only, where the variable **VOF** of the **Phase #N** is defined
- if **Characteristics** are built by a **User variable** (**UV #N**), created based on the variable variable **VOF** of the **Phase #N**, then:
  - **Volume** is calculated over all computational cells of the **Subregion**, independently on their **Phases**.
  - **Mass** is calculated over all computational cells of the **Subregion**, independently on their **Phases**.

See also illustrations in the subsection *"Domain of integration (for calculation Characteristics) depending on the Variable's category"* of the section Characteristics.

8.1.8.3.10.3 The Info window for Characteristics

The **Info** window with components of the **Characteristics** that was calculated at the current time step, is opened when you select in the project tree the **Characteristics** element and then click on the icon  (**Show info window for selected object**) in the toolbar **Work modes**.

When **Solver** is connected, this window displays a list of the characteristics' components along with their values. Values of the components are updated after each time step. For the element **Internal characteristics** you can see the data on the time step.

The components are grouped into blocks in a tree structure.

Information window[Characteristics #1 (Supergroup #0)]

Name	Value
Solver data	Present
Step number	56
Time	0.56
Variable	PRES
Block	Motion
Phase	All phases
Subregion	SubRegion #0
Area	0.0014132484424354
Mass flow+	0
Mass flow-	0
Volume flow+	0
Volume flow-	0
Integral X	0.0020304801831612
Integral Y	-11.159682161267
Integral Z	0.00028655428507296
<f surf.>	9495.7232905933
<f mass+>	0
<f mass->	0
<f mass+> * Mass flow+	0
<f mass-> * Mass flow-	0
Stand. deviation	1279.7474183208
Stand. mass deviation	0
Heat flux [W]	0
F fluid X	0.001969173409309
F fluid Y	-11.161869363795
F fluid Z	0.00039659292338046
M center X	0
M center Y	0
M center Z	0
M fluid X	4.0053444417959e-006
M fluid Y	0.00023464592993919
M fluid Z	-4.3096228250709e-005
Autorotation angle speed	0
Center of pressure X	0
Center of pressure Y	0.021885440889552
Center of pressure Z	0.11915960718844

Example of the **Info** window for **Characteristics**

Method for calculating the components of the **Characteristics** element depends on the **Characteristics**' parameters **Parts > Select** and **Parts > Surfaces > ...**:

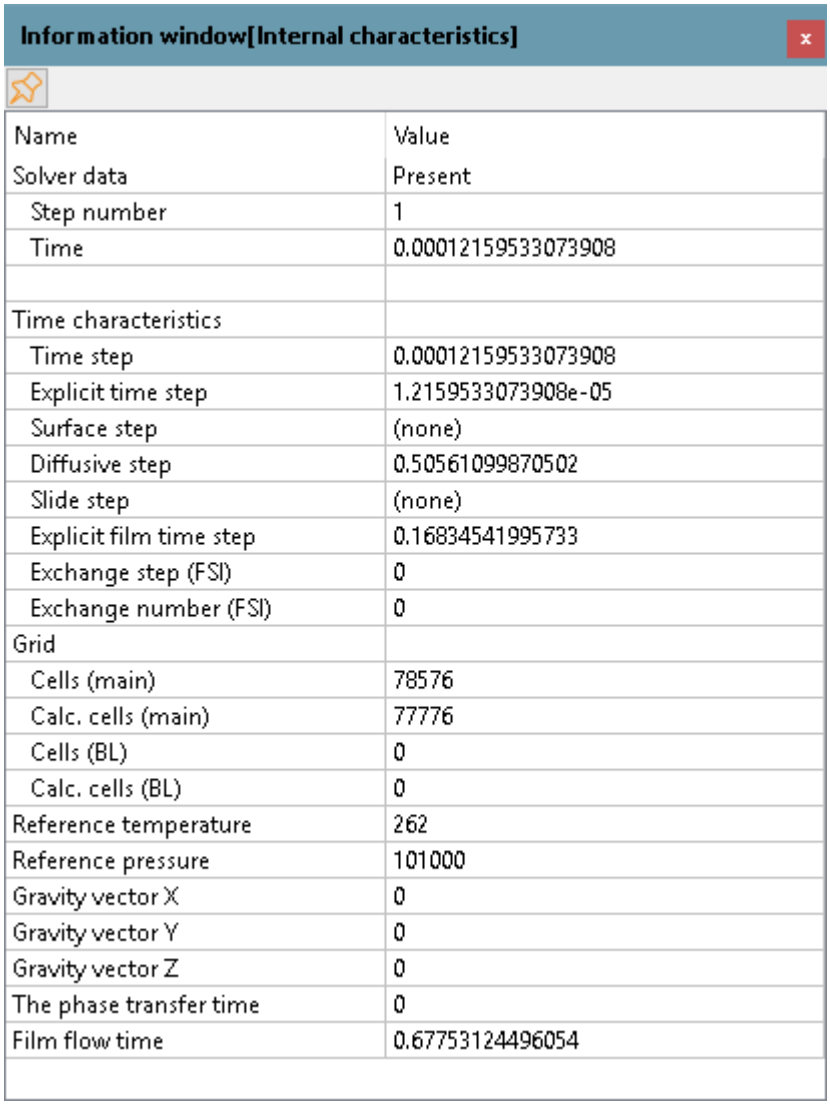
- if the **Parts > Select** parameter is selected as **Volume**, then components of the **Characteristics** are calculated in the volume of the subregion
- if the **Parts > Select** parameter is selected as **Whole surface** (or as **Whole plane** for **Characteristics** on a **Plane**), then components of the **Characteristics** are calculated on the surface of an object, which locates in the subregion.

- if the **Parts > Select** parameter is selected as **Selected surfaces** (or as **Selected contour** for **Characteristics** on a **Plane**) and the parameters **Part > Surfaces > ...** specify the required surfaces (faces of a box, bases of a cone, etc.), on which you wish to calculate the variable, then components of the **Characteristics** are calculated on the specified surfaces (or within the appropriate contour of a plane, where the plane's point **P1** locates).

General information presented in any Info windows for Characteristics

Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current simulated time
Variable	Name of the displayed variable
Variable > Block	The physical process, which is described by the variable
Variable > Phase	The **Phase**, to which the variable belongs
Subregion	Name of the **Subregion**, on which characteristics are calculated

Information in the Info window for the "Internal characteristics" element



Information window[Internal characteristics]

Name	Value
Solver data	Present
Step number	1
Time	0.00012159533073908
Time characteristics	
Time step	0.00012159533073908
Explicit time step	1.2159533073908e-05
Surface step	(none)
Diffusive step	0.50561099870502
Slide step	(none)
Explicit film time step	0.16834541995733
Exchange step (FSI)	0
Exchange number (FSI)	0
Grid	
Cells (main)	78576
Calc. cells (main)	77776
Cells (BL)	0
Calc. cells (BL)	0
Reference temperature	262
Reference pressure	101000
Gravity vector X	0
Gravity vector Y	0
Gravity vector Z	0
The phase transfer time	0
Film flow time	0.67753124496054

Line with data	Description
Time characteristics	Group of parameters describing the temporal parameters of the current step
Time step	
Explicit time step	
Surface step	
Diffusive step	
Slide step	
Explicit film time step	
Exchange step (FSI)	
Exchange number (FSI)	
Grid	Group of parameters with information about the computational grid
Cells (main)	Total number of cells in the main grid
Calc. cells (main)	Number of computational cells in the main grid
Cells (BL)	Total number of cells in the boundary layer (BL) grid
Calc. cells (BL)	Number of computational cells in the boundary layer (BL) grid

Line with data	Description
Reference temperature	The reference temperature
Reference pressure	The reference pressure
Gravity vector X	Components of the **Gravity vector**
Gravity vector Y	
Gravity vector Z	
The phase transfer time	This is the total (within the whole period of the simulation when the dispersed phase crystallization's program block is active) duration of forming the solid phase (i.e. ice in simulations of icing).
Film flow time	This is the total (within the whole period of the simulation when the dispersed phase crystallization's program block is active) duration of the the film's spreading. This is the total duration of the film's spreading with taking into account all iterations that were made ate the stage of obtaining the quasi-stationary process. This value is informative for non-stationary problem settings only.

Information in the Info window for characteristics defined in volume

Depending on the **Variable**, based on which the characteristics are calculated, the volume, on which the integrating is done, is different:

Variable	The volume, on which the integrating is done
VOF*) (it is defined for a some specific **Phase**)	The volume of the **Object** where the **Model** is specified, which contains the **Phase**
Another variable that is defined for a some specific **Phase***)	The volume of the **Object*****), which is filled by the **Phase**
A variables that is defined as not associated to any specific **Phase****)	The volume of the **Object*****), which is filled by any **Phases**, in which the variable is defined
A user variable	The volume of the **Object** within the computational domain. So, if you create a user variable **UV #N**, which is equal to the variable **VOF**, then **Characteristics** by **UV #N** and **VOF** will be calculated over different regions of integration.

*) These variables can be selected when **Variable > Category = Variables of phase "Phase #N"** is specified in the properties of the **Characteristics**

) These variables can be selected when **Variable > Category = Common and phase-unrelated variables is specified in the properties of the **Characteristics**

***) Not exceeding the **Subregions**, where the **Model** with this variable is set

(See also the information about categories of variables and descriptions and illustrations in the subsection *"Domain of integration (for calculation Characteristics) depending on the Variable's category"* in the section Characteristics.)

Line with data	Description
Volume	V The volume, over which the integrating has been done, [m^3]. For the variable **VOF** this is the total volume of all computational cells in the subregion.
Mass	m The mass of the medium in the volume, [kg] $m = \iiint_V \rho dV$ For the variable **VOF** the mass is calculated only for the **Phase**, to which the **VOF** variable belongs.

Line with data	Description
<f vol.>	<f> $_V$ The volume averaged value of a scalar variable f, [f]. $\langle f \rangle_V = \frac{1}{V} \iiint_V f dV$ For the variable **VOF** this is the fraction of the volume, which is filled by the **Phase**, in the total volume, over which the integrating has been done (which is displayed in the line **Volume**).
<f mass>	$\langle f \rangle_m$, the mass averaged value of a scalar variable f, [f] $\langle f \rangle_m = \frac{1}{m} \iiint_V f \rho dV$
<f vol.> * Volume	<f> $_V V$ Amount of scalar variable f in the volume, [f] For the variable **VOF** this is the volume of the liquid, which partially fills the volume, over which the integrating has been done.
Maximum	The maximal value of the variable f in volume of the subregion, [f]
Maximum > Hypercell max **Maximum > Cell max**	The two-level number (in the <Hypercell: Cell> format) of the cell in which the maximum of the variable is reached.
Maximum > Point max. X	Coordinates, [m], (in the absolute coordinate system) of the point, at which the maximum of the variable f is achieved
Maximum > Point max. Y	
Maximum > Point max. Z	
Minimum	The minimal value of the variable f in volume of the subregion, [f]
Minimum > Hypercell min **Minimum > Cell min**	The two-level number (in the <Hypercell: Cell> format) of the cell in which the minimum of the variable is reached.
Minimum > Point min. X	Coordinates, [m], (in the absolute coordinate system) of the point, at which the minimum of the variable f is achieved
Minimum > Point min. Y	
Minimum > Point min. Z	

When calculating **Characteristics** for individual enclaves is enabled (**Enclave analysis > Enabled = Yes** is set in properties of the **Characteristics**), data blocks in the **Info** window that relates to individual enclaves are marked as "**[* *n* *]**", where n is the enclave's number (0, 1, 2, ...).

Information in the «Info» window for Characteristics that are specified on a surface

Depending on the **Variable**, by which the characteristics are calculated, the surfaces, over which the integrating is done, are different. The rules are used similar to those, which are used for calculating **Characteristics** in the volume (see description in the previous subsection, *"Information in the Info window for characteristics defined in volume"* and illustrations in the subsection *"Domain of integration (for calculation Characteristics) depending on the Variable's category"* in the section Characteristics).

Line with data	Description
Part	Possible options are: • **Whole surface**: the whole surface of the geometry **Object**, on which the **Characteristic** is built • **All contours**: all sections of the **Subregion** cut by the **Plane**, on which the **Characteristic** is built

Line with data	Description
	• **Selected contour**: those section of the **Subregion** cut by the **Plane**, on which the **Characteristic** is built • ***Surface part's name***: some specified part of the surface, on which the variable is calculated (for example, **Bottom base**, **Lateral surface**, **X-**, **X+**, **Y-**, **Y+**, ...)
Area	S Area of the surface, over which the integrating has been done, [m^2]
Mass flow+	G_+ is the convective mass flow across the surface in the direction of the normal to the surface, [kg s^{-1}] $G_+ = \oiint_S \rho(\vec{V}_{abs}\vec{n})_+ dS, \quad (a)_+ = \begin{cases} 0, & a \le 0 \\ a, & a > 0 \end{cases}$
Mass flow-	G_- is the convective mass flow across the surface in the direction opposite to the normal to the surface, [kg s^{-1}] $G_- = \oiint_S \rho(\vec{V}_{abs}\vec{n})_- dS, \quad (a)_- = \begin{cases} a, & a < 0 \\ 0, & a \ge 0 \end{cases}$
Volume flow+	N_+ is the flow across the surface in the direction of the normal to the surface, [m^3 s^{-1}] $N_+ = \oiint_S (\vec{V}_{abs}\vec{n})_+ dS, \quad (a)_+ = \begin{cases} 0, & a \le 0 \\ a, & a > 0 \end{cases}$
Volume flow-	N_- is the flow across the surface in the direction against to the normal to the surface, [m^3 s^{-1}] $N_- = \oiint_S (\vec{V}_{abs}\vec{n})_- dS, \quad (a)_- = \begin{cases} a, & a < 0 \\ 0, & a \ge 0 \end{cases}$
Integral X **Integral Y** **Integral Z**	$\Sigma_{f,i}$ i^{th} component of the integral of a scalar variable f over the oriented surface, [f m^2] $\Sigma_{f,i} = \oiint_S f\vec{n}_i dS, \quad i = 1(X), 2(Y), 3(Z)$ $\Sigma_{\vec{F},i}$ integral of i^{th} component of the vector variable $\vec{F}$ over the surface, [f m^2] $\Sigma_{\vec{F},i} = \oiint_S \vec{F}_i dS, \quad i = 1(X), 2(Y), 3(Z)$
<f surf.>	$\langle f \rangle_S$ is the average (by the surface) value of a scalar variable f, [f] $\langle f \rangle_S = \frac{1}{S} \oiint_S f\, dS$ $\langle \vec{F} \rangle_S$ is the average (by the surface) value of a vector variable $\vec{F}$, [f] $\langle \vec{F} \rangle_S = \frac{1}{S} \oiint_S (\vec{F}\vec{n})\, dS$ For the variable **VOF** this line displays the fraction of the surface moistened by the liquid in the total area of the surface, over which the integrating has been done.

Line with data	Description
<f mass +>	$\langle f \rangle_{G+}$ is the average (by the mass) value of a scalar variable f, transferred in the direction of the normal to the surface, [f] $\langle f \rangle_{G+} = \frac{1}{G_+} \oiint_S \rho \cdot f \cdot (\vec{V}_{abs}\vec{n})_+ dS$, $(a)_+ = \begin{cases} 0, & a \le 0 \\ a, & a > 0 \end{cases}$ $\langle \vec{F} \rangle_{G+}$ is the average (by the mass) value of a vector variable $\vec{F}$ transferred in the direction of the normal to the surface, [f] $\langle \vec{F} \rangle_{G+} = \frac{1}{G_+} \oiint_S \rho \cdot \vec{F} \cdot (\vec{V}_{abs}\vec{n})_+ dS$, $(a)_+ = \begin{cases} 0, & a \le 0 \\ a, & a > 0 \end{cases}$
<f mass ->	$\langle f \rangle_{G-}$ is the average (by the mass) value of a scalar variable f, transferred in the direction opposite to the normal of the surface, [f] $\langle f \rangle_{G-} = \frac{1}{G_-} \oiint_S \rho \cdot f \cdot (\vec{V}_{abs}\vec{n})_- dS$, $(a)_- = \begin{cases} a, & a < 0 \\ 0, & a \ge 0 \end{cases}$ $\langle \vec{F} \rangle_{G-}$ is the average (by the mass) value of a vector variable $\vec{F}$ transferred in the direction opposite to the normal of the surface, [f] $\langle \vec{F} \rangle_{G-} = \frac{1}{G_-} \oiint_S \rho \cdot \vec{F} \cdot (\vec{V}_{abs}\vec{n})_- dS$, $(a)_- = \begin{cases} a, & a < 0 \\ 0, & a \ge 0 \end{cases}$
<f mass +> * Mass flow +	$\langle f \rangle_{G+} G_+$ flow of scalar variable f across the surface in the direction of the normal to the surface, [f kg s^{-1}]
<f mass -> * Mass flow-	$\langle f \rangle_{G-} G_-$ flow of scalar variable f across the surface in the direction opposite to the normal to the surface, [f kg s^{-1}]
Stand. deviation	σ_S standard deviation of a scalar variable f on a surface, [f] $\sigma_S = \sqrt{\frac{1}{S}\oiint_S (f - \langle f \rangle_S)^2 dS}$
Stand.mass deviation	σ_G mass-weighted average standard deviation of a scalar variable f on a surface, [f] $\sigma_G = \sqrt{\frac{\oiint_S (f - \langle f \rangle_S)^2 \rho(\vec{V}_{abs}\vec{n}) dS}{\oiint_S \rho(\vec{V}_{abs}\vec{n}) dS}}$
Heat flux [W]	Q, which is the heat flux through the surface *(integral, not specific)*, [W]. It is only calculated on the surface of a **Supergroup** on a BC **Wall** or **Connected**. $Q = -\oiint_S (\lambda + \lambda_t)(\nabla T_S \vec{n}) dS$
F fluid X **F fluid Y** **F fluid Z**	Components of the vector $\vec{F}_{fluid}$ of the force of acting the surface on the medium, [N]. $\vec{F}_{fluid} = \oiint_S (P + P_{hst})\vec{n} dS - \oiint_S (\mu + \mu_t)\frac{\partial \vec{V}_\tau}{\partial n} dS$ This is calculated in the absolute coordinate system (ACS).

Line with data	Description
M center X **M center Y** **M center Z**	Coordinates of the point $\vec{r}_M$, relatively to which the torque $\vec{T}_{fluid}$ is calculated, [m], in the absolute coordinate system (ACS). These coordinates are specified in properties of the element **Characteristics**, in the parameters **Center**.
M fluid X **M fluid Y** **M fluid Z**	Components of the vector $\vec{T}_{fluid}$, which is torque of the force $\vec{F}_{fluid}$ relatively to the point $\vec{r}_M$, [N m] $\vec{T}_{fluid} = \oiint_S (P + P_{hst})[\vec{r}_M \times \vec{n}]dS - \oiint_S (\mu + \mu_t)\left[\vec{r}_M \times \frac{\partial V_\tau}{\partial n}\right]dS$ This is calculated in the absolute coordinate system (ACS).
Moving body*)	The **Moving body**, which is specified on the **Imported object** on which the **Characteristics** are calculated.
Rotation center X*) **Rotation center Y***) **Rotation center Z***)	Coordinates X, Y, Z of the rotation center of the **Moving body**, specified in the absolute coordinate system (ACS), [m]
Velocity X*) **Velocity Y***) **Velocity Z***)	Components of velocity of translation movement of the **Moving body**, specified in the absolute coordinate system (ACS), [m s^{-1}]
Rotation velocity X*) **Rotation velocity Y***) **Rotation velocity Z***)	Components of rotational speed of the **Moving body**, specified in the absolute coordinate system (ACS), [rad s^{-1}]
Rotation X X*) **Rotation X Y***) **Rotation X Z***) **Rotation Y X***) **Rotation Y Y***) **Rotation Y Z***) **Rotation Z X***) **Rotation Z Y***) **Rotation Z Z***)	Elements of rotation matrix of the **Moving body** (dimensionless quantities). For example, **Rotation X Y** is the projection of the unit vector OX of the local coordinate system of the **Moving body** on the axis OY of the absolute coordinate system (ACS).
F body X*) **F body Y***) **F body Z***)	Components of the total vector of the force acting on the **Moving body**, [N], which is calculated in the absolute coordinate system (ACS) by the formula: $\boldsymbol{F}_{body} = -\boldsymbol{F}_{fluid} + \boldsymbol{F}_{ext} + m\boldsymbol{g}$ where $\boldsymbol{F}_{ext}$ and $m\boldsymbol{g}$ are the external force and the gravitational force acting on the **Moving body**.
M body X*)	Components of the total vector of the torque acting on the **Moving body**, [N m], which is calculated in the absolute coordinate system (ACS) by the formula: $\boldsymbol{T}_{body} = \int_S \mathbf{r} \times d\boldsymbol{F}_{body} + \boldsymbol{T}_{ext}$ where $\boldsymbol{T}_{ext}$ is the external torque acting on the **Moving body**.

Line with data	Description
M body Y[*)]	The torque T_{body} is calculated relatively to the center of rotation of the **Moving body**.
M body Z[*)]	
Autorotation angle speed	Angular speed ω of rotation of a **Subregion**, if the computation was done with autorotation (see section Folder «Local coordinate systems»)
Center of pressure X	Coordinates of the center of pressure. Center of pressure is the point, in which line, along which the line of action of the resultant pressure forces applied to the stationary or moving body by the fluid intersects with a selected plane within the body. For example, for an airplane wing the center of pressure is defined as the point, in which line of action of the aerodynamic force with the plane of the wing's chords; for an axially symmetrical body (rocket case, dirigible balloon, etc,) it is defined as the point, in which line of action of the aerodynamic force intersects the plane of symmetry passing the center line and vector of the body's mass center's velocity. See [Лойцянский Л.Г. Механика жидкости и газа. Изд. 3. М.:Наука,1970] (Russian source). The plane, on which the center of pressure is sought, is specified in properties of the **Characteristics** by the following parameters: • it passes through the point, which is specified by parameters **Center > ...** • its normal is specified by parameters **Normal_PC > ...** To calculate the center of pressure of a wing you should specify the **Center** parameter of the characteristics on the leading edge of the wing.
Center of pressure Y	
Center of pressure Z	
Integral surf. +	Integral over the surface, on which the **Characteristics** is calculated, with positive and negative values integrated separately. For vector values, the program integrates their scalar product by local normal to the surface. In the text (glo) file, the lines **Integral surf. +** and **Integral surf. -** are saved in fields **P_SurfaceIntegralPlus** and **P_SurfaceIntegralMinus**. These data are calculated when **Extended data = Yes** is set in properties of the **Characteristics**.
Integral surf. -	

[*)] These data lines are displayed when the **Characteristics** are specified on an **Imported object**, on which a **Moving body** is specified. Values of components of the force and the torque acting on the **Moving body** are displayed in the table as zeros if the **Moving body**, due to its settings, cannot move under the action of these force and/or torque.

Specific information in «Info» windows for Characteristics that are specified on Sets of sensors

Info windows for **Characteristics**, which are specified on **Sets of sensors**, contain groups of lines with data where each group corresponds to one **Sensor** from the **Set**. These groups contain the following lines:

Line with data	Description
Part	Name of the **Sensor** (it is set in properties of the **Set of sensors** by the **Sensor set > [N] > Name** parameter; the default name is **Sensor #N**).
Sensor index	Index of the **Sensor** in the **Set**
Sensor type	This parameter informs if the **Sensor** is bound (stuck) to a surface. Possible options are: • **SURF**: the **Sensor** is bound (stuck) to a surface • **VOL**: the **Sensor** is specified in the volume (it is not stuck to any surface)

8.1.8.3.10.4 Components of a text file for recording data from Characteristics

At each recording into the text file (with extension `glo`), which has been specified in the **Properties** window in the parameter **Save to file > Name of the file** (in the **Postprocessor** tab), the information is recorded corresponding to the lines that are displayed in the **Info** window (see descriptions of the data, which are given in the previous sections):

Step	Step
Time	Time
Variable	Parameters of the variable, which is used to built the characteristics
id	Variable's identifier on **Solver**
block	Block
phase	Phase
SubRegion	Subregion

Internal characteristics:

Time step	The time step
ExpTimeStep	The explicit time step
RefTemperature	The reference temperature
RefPressure	The reference pressure
GravityVector.x	Components of the **Gravity vector**
GravityVector.y	
GravityVector.z	

Surface characteristics (are output when **Parts > Select > All surface** or **Surface Selected**):

Area	Area, [m2]
MassFlowP	Mass flow +, [kg s-1]
MassFlowN	Mass flow -, [kg s-1]
VolumeFlowP	Volume flow +, [m3 s-1]
VolumeFlowN	Volume flow -, [m3 s-1]
Integral.x	Integral X, [f m2]
Integral.y	Integral Y, [f m2]
Integral.z	Integral Z, [f m2]
Agv	<f surf.>, [f]
AgvFlowP	<f mass+>, [f]
AgvFlowN	<f mass->, [f]
ValFlowP	<f mass+> * Mass flow +, [f kg s-1]
ValFlowN	<f mass-> * Mass flow-, [f kg s-1]
StdDeviation	Standard deviation over a surface, [f]
StdMassDeviation	Standard deviation over a stream, [f]
HeatFlux	The heat flux Q through the surface (the integral one, not specific) [W]. It corresponds to the line Heat flux [W] in the Info window. It is only calculated on the surface of a **Supergroup**.
F_Fluid.x	F liq. X, [N]
F_Fluid.y	F liq. Y, [N]
F_Fluid.z	F liq. Z, [N]

M_Center.x	M Center X, [m]
M_Center.y	M center Y, [m]
M_Center.z	M center Z, [m]
M_Fluid.x	M liq. X, [N m]
M_Fluid.y	M liq. Y, [N m]
M_Fluid.z	M liq. Z, [N m]

Additional surface characteristics (they are output when a **Moving body** is built on an **Imported object**):

MovingBody	The **Moving body**, which has been built on the **Imported object**	
RotCenter.x	The center of rotation X, [m]	
RotCenter.y	The center of rotation Y, [m]	
RotCenter.z	The center of rotation Z, [m]	
Velocity.x	Velocity X, [m s^{-1}]	
Velocity.y	Velocity Y, [m s^{-1}]	
Velocity.z	Velocity Z, [m s^{-1}]	
RotVelocity.x	Angular speed X, [rad s^{-1}]	
RotVelocity.y	Angular speed Y, [rad s^{-1}]	
RotVelocity.z	Angular speed Z, [rad s^{-1}]	
RotationX.x	Rotate X X	Elements of rotation matrix of the **Moving body** (dimensionless quantities). For example, **Rotation X.y** is the projection of the unit vector OX of the coordinate system of the **Moving body** on the axis OY of the absolute coordinate system.
RotationX.y	Rotate X Y	
RotationX.z	Rotate X Z	
RotationY.x	Rotate Y X	
RotationY.y	Rotate Y Y	
RotationY.z	Rotate Y Z	
RotationZ.x	Rotate Z X	
RotationZ.y	Rotate Z Y	
RotationZ.z	Rotate Z Z	
F_Body.x	F body X	Components of the total vector of the force acting on the **Moving body**, [N]
F_Body.y	F body Y	
F_Body.z	F body Z	
M_Body.x	M body X	Components of the total vector of the torque acting on the **Moving body**, [N m]
M_Body.y	M body Y	
M_Body.z	M body Z	
Omega	Angular speed of rotation (when computation is carried out with autorotation).	
P_SurfaceIntegral Plus	Integral over the surface, on which the **Characteristics** is calculated, with positive and negative values integrated separately. For vector values, the program integrates their scalar product by local normal to the surface. These data are calculated when **Extended data = Yes** is set in properties of the **Characteristics**.	
P_SurfaceIntegral Minus		

Volumetric characteristics (they are output when a **Parts > Select = Volume**):

Vol	Volume, [m^3]
Mass	Weight, [kg]

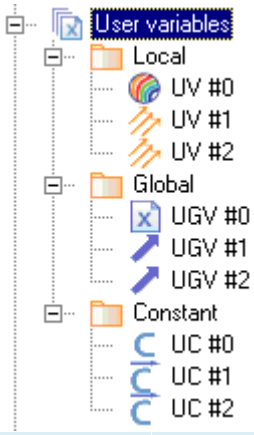
Agv	<f vol.>, [f]
AverageByMass	<f mass>, [f]
AvgMulVol	<f vol.> * Volume, [f]
Max	Maximum, [f]
MaxHC	Hypercell max.
MaxCell	Cell max.
MaxPt.x	The point of max. X, [m]
MaxPt.y	The point of max. Y, [m]
MaxPt.z	The point of max. Z, [m]
Min	Minimum, [f]
MinHC	Hypercell min.
MinCell	Cell min.
MinPt.x	The point of min. X, [m]
MinPt.y	The point of min. Y, [m]
MinPt.z	The point of min. Z, [m]

8.1.8.3.11 Folder «User variables»

User variables are created by the user based on **Physical variables** and **Constants**. **User variables** can be created in both **Preprocessor** or **Postprocessor**.

The **User variables**, which has been created in **Postprocessor**, can be used:

- on **Characteristics**, which are created in **Postprocessor**
- in **Layers**



Folder **User variables** in the project tree

A **User variable** element is displayed in the project tree as an element in one of the following folders:

- **User variables > Local** contains *local user variables* that may depend on coordinates and time
- **User variables > Global** contains *global user variables* that do not depend on coordinates (but that may depend on time)
- **User variables > Constant** contains *constant user variables* that depend on neither coordinates no time

Local user variables

Local user variables are user-defined variables that can depend on the coordinates and time. When setting the local user variables, you can use constants, local physical variables, integral variables previously created local, global and permanent user-defined variables.

User variables,in the folder **Local**, formed from local physical variables, constants, variables, integral with a local, global and permanent user-defined variables. These components are available in panels **Variables and Constants** **Formula editor**.

Name of an element created in the folder **User variables > Local** has a format **UV #N** (it is mnemonic for "*User Value*").

In the project tree for **Local User variables**, the following icons are used:

 - for scalars

 - for vectors

Global user variables

Global user variables are user-defined variables that do not depend on the coordinates (but may depend on time). When you set the global user variables, you can use constants, variables, and integrated pre-existing global user variables and constants.

User variables,in the folder **Global**, formed from the global physical variables, constants, variables, integral, permanent and available global user variables. These components are available in panels **Variables and Constants** **Formula editor**.

Name of an element created in the folder **User variables > Global** has a format **UGV #N** (it is mnemonic for "*User Global Value*").

In the project tree for **Global User variables**, the following icons are used:

 - for scalars

 - for vectors

For **Global User variables** you can specify their default values that will be used for calculating functions `prev` and `prevvec` at the first step of the computation.

Constant user variables

Persistent user variables - user-defined variables that do not depend on time and coordinates. When setting permanent, you can use user-defined variables and constants previously created custom variables constant.

User variables, **constants** in the folder, formed from the constant integral variables, which have a uniform user-defined variables. These components are available in panels **Variables and Constants** **Formula editor**.

Name of an element created in the folder **User variables > Constant** is formatted **UC #N** (it is mnemonic for "*User Constant*").

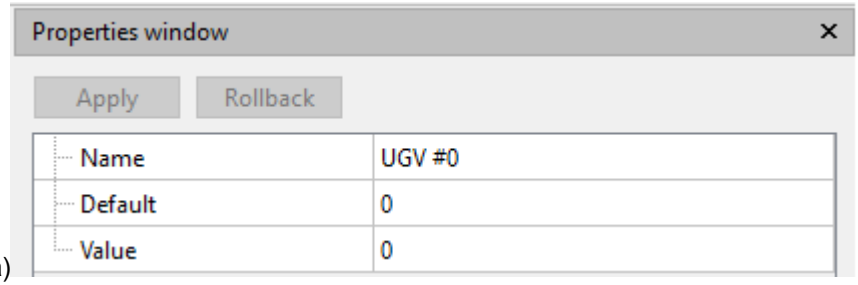
In the project tree for **Loyal user** variables,the following icons:

 - for scalars

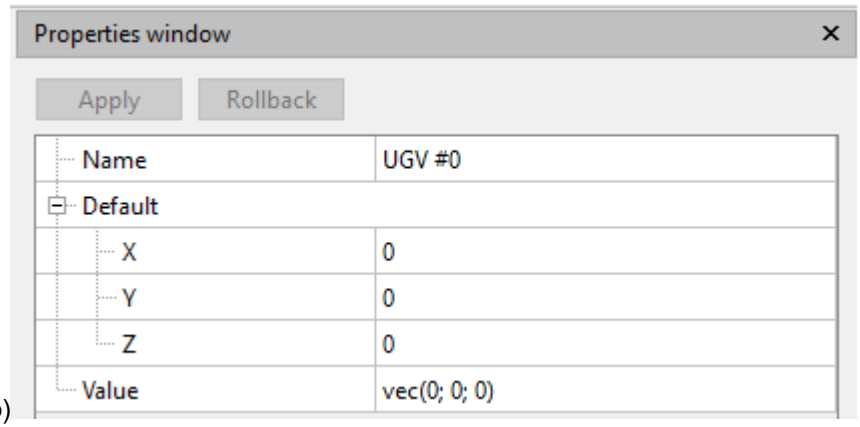
 - for vectors

Parameters of User variables

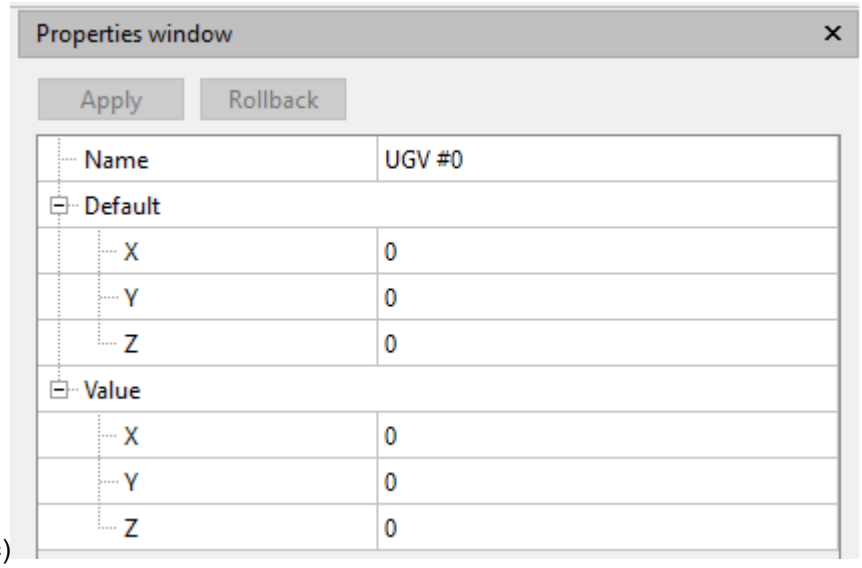
a)



b)



c)



The **Properties** window of a **Global User variable** element: a - for a scalar, b - for a vector, c - for a vector defined by components.

Default values are not set for **Local** user variables and **Constants**.

Parameters of scalar **User variables**:

Parameter	Description
Name	Name of the scalar variable
Default[*)]	The default value of the scalar variable
Value	Value of the scalar variable.

[*)] Only **Global User variables** have this parameter.

Parameters of vector **User variables**, except defined by components:

Parameter	Description
Name	Name of the vector variable
Default > X[*)]	Default values of components of the vector variable
Default > Y[*)]	
Default > Z[*)]	
Value	Value of the vector variable defined as `vec(value1, value2, value3)`.

[*)] Only **Global User variables** have these parameters.

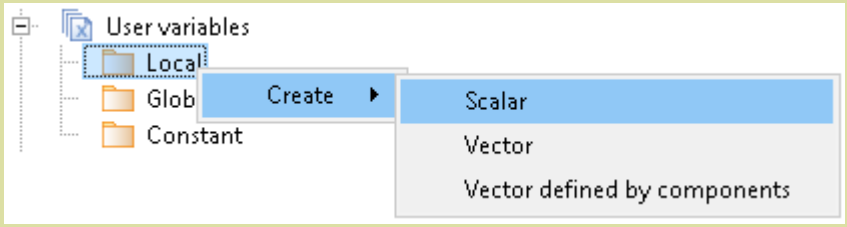
Parameters of vector **User variables** defined by components:

Parameter	Description
Name	Name of the vector variable
Default > X[*)]	Default values of components of the vector variable
Default > Y[*)]	
Default > Z[*)]	
Value > X	Values of components of the vector variable
Value > Y	
Value > Z	

[*)] Only **Global User variables** have these parameters.

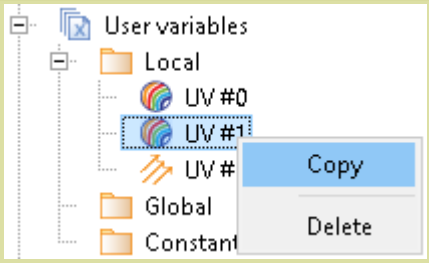
Context menus

The **User variables** folder had no context menu but its child subfolders do have context menus.



Context menu of a child subfolder in the "User variables" folder

Menu item	Description
Create	Submenu with commands to create a new custom variable
Scalar	Create a new scalar variable
Vector	Create a new vector variable
Vector defined by components	Create a new vector variable as defined by specifying three scalar components



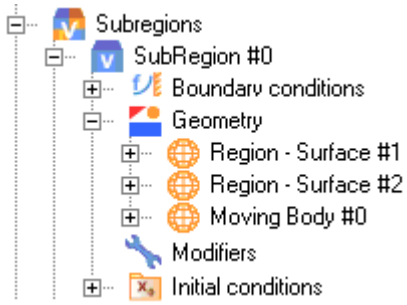
Context menu of a "User variable" element

Menu item	Description
Copy	Creating an element, which is a copy of the selected element
Delete	Deleting the selected element from the project tree

8.1.8.3.12 Folders «Subregions» and «SubRegion #N»

Subregion is a closed geometric volume, which is modeled for. A predefined **Model** is to be loaded into a **Subregion**.

Folder «Subregions»



Folder **Subregions** in the project tree

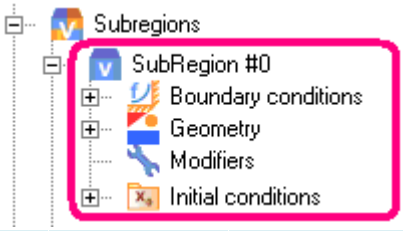
Subregions are automatically generated by the program after loading the geometry model of the computational domain into the project.

Each subfolder **Subregions > SubRegion #N** stores information about one **Subregion** and contains child subfolders:

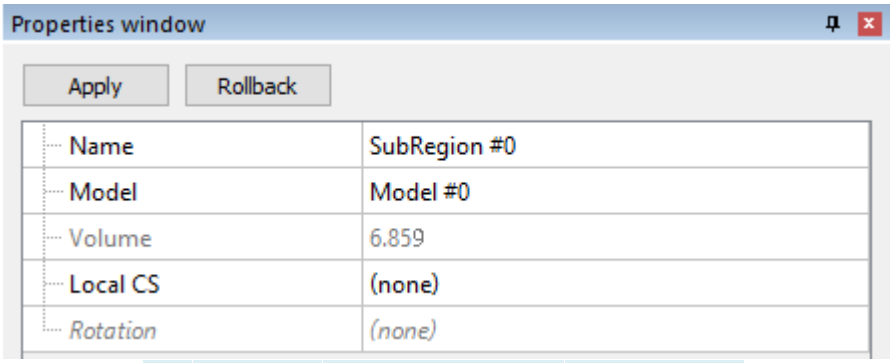
- **Boundary conditions**
- **Geometry**
- **Modifiers**
- **Initial conditions**

The **Subregions** folder does not contain parameters (its **Properties** window is empty) and has no context menu.

Subfolders «Subregions> SubRegion #N»



Folder **SubRegion #N** in the project tree

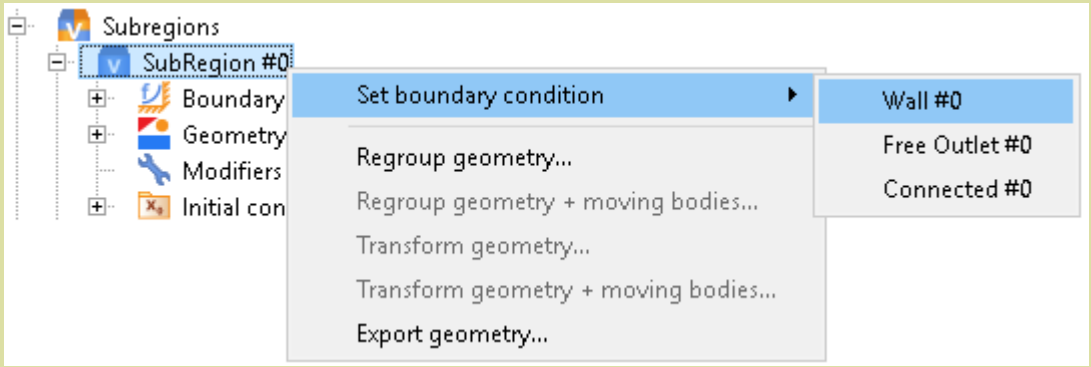


The **Properties** window of the folder **SubRegion #N**

Parameters of **Subregion #N**:

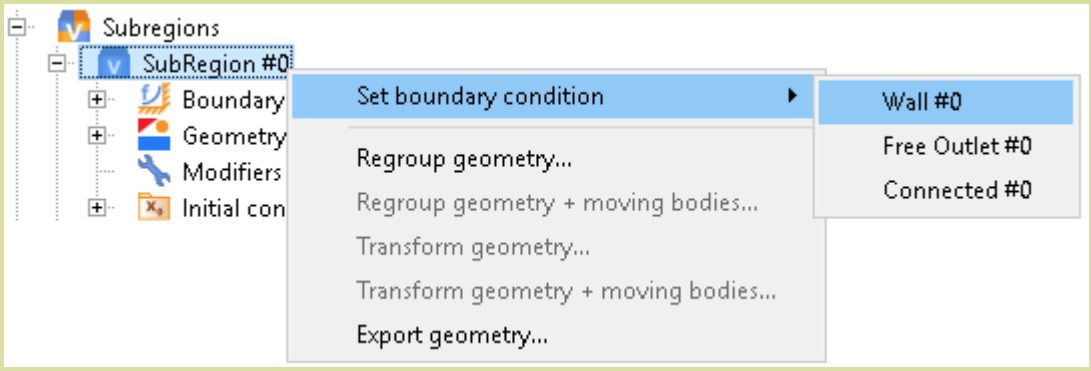
Parameter	Description
Name	Subregion name
Model	A **Model**, that is set for the **Subregion**. It is selected from the list of models presented in the **Models** folder. The initial conditions, **boundary conditions and modifiers** is possible only after the specifying **Models** in a **Subregion** !!! If in a **Subregion** no **Model** is defined, then calculations in this **Subregion** are not carried out !!!
Volume	Volume of the **Subregion**, [m^3], excluding volumes of **Moving bodies**. This field is informational and it is not editable.
Local FR	Local coordinate system of the **Subregion**
Rotation	**Rotation** of the local coordinate system that is set for the **Subregion**

Context menu of subfolders «Subregions > SubRegion #N»



Context menu of a folder "Subregions > Subregion #N"

Menu command	Description
Set boundary condition > *Boundary condition #N*	Setting a certain boundary on the boundary of the whole **Subregion**.
Regroup geometry	Opens the **Geometry regrouping** dialog box to change the composition of groups of facets of the boundary of the computational domain, see section Procedure of regrouping a geometric model of computational domain (and moving bodies).
Regroup geometry + moving bodies	Opens the **Geometry regrouping** dialog box to change the composition of groups of facets of the boundary of the computational domain *and* **Moving bodies**, see section Procedure of regrouping a geometric model of computational domain (and moving bodies).



Context menu of a folder "Subregions > Subregion #N"

Menu command	Description
Transform geometry	Opens the **Geometry transformation** dialog box to transform boundaries of the computational **Subregion**. When multiconnection exists on the surfaces, which limit the **Subregion**, a warning will be displayed informing you that the geometry transformation will be applied to *all* **Complex surfaces**, which limit the **Subregion**.
Transform geometry + moving bodies	Opens the **Geometry transformation** dialog box to transform boundaries of the computational **Subregion** *and* **Moving bodies**. When multiconnection exists on the surfaces, which limit the **Subregion**, a warning will be displayed informing you that the geometry transformation will be applied to *all* **Complex surfaces**, which limit the **Subregion** and *all* **Moving bodies** located in the affected **Subregions**.
Export geometry	This command saves in a file the information about boundaries of the computational **Subregion**. File selection is done using a standard operating system's dialog box.

8.1.8.3.12.1 Folder «Boundary conditions»

Boundary condition is presented by a set of parameters defined on a selected surface.

Boundary conditions have to be matched to the respective surfaces. This is done automatically: if the loaded geometry model used a format that supports color information, then **Boundary conditions** are created and arranged automatically according to the colors. By default, in this case all boundary conditions will be of the type **Wall**.

If boundary conditions have not been assigned automatically, you have to assign them manually.

Assignment of boundary conditions is done:

- in the **View** window using a mouse
- in the properties of a **Group**
- in context menus of a **Subregion**, a **Surface** or a **Moving body**[1)]

Do not forget to assign boundary conditions!

The folder **Subregions > SubRegion #N > Boundary conditions** contains elements **B. Cond. #N** that correspond to individual boundary conditions.

The context menu of the folder **Boundary conditions** includes the **Create** command, which creates a new element **B. Cond #N**.

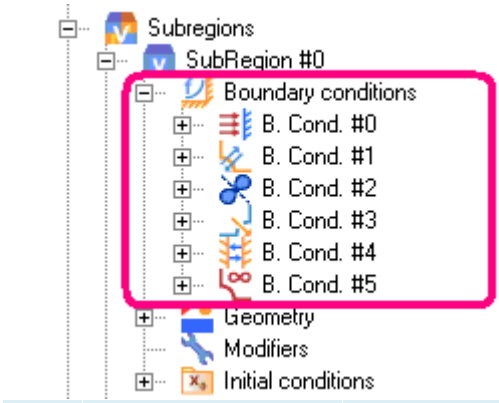
Each boundary condition **B. Cond. #N** contains a set of elements, the composition of which is determined by the model specified in the subregion.

Element «B. Cond. #N»

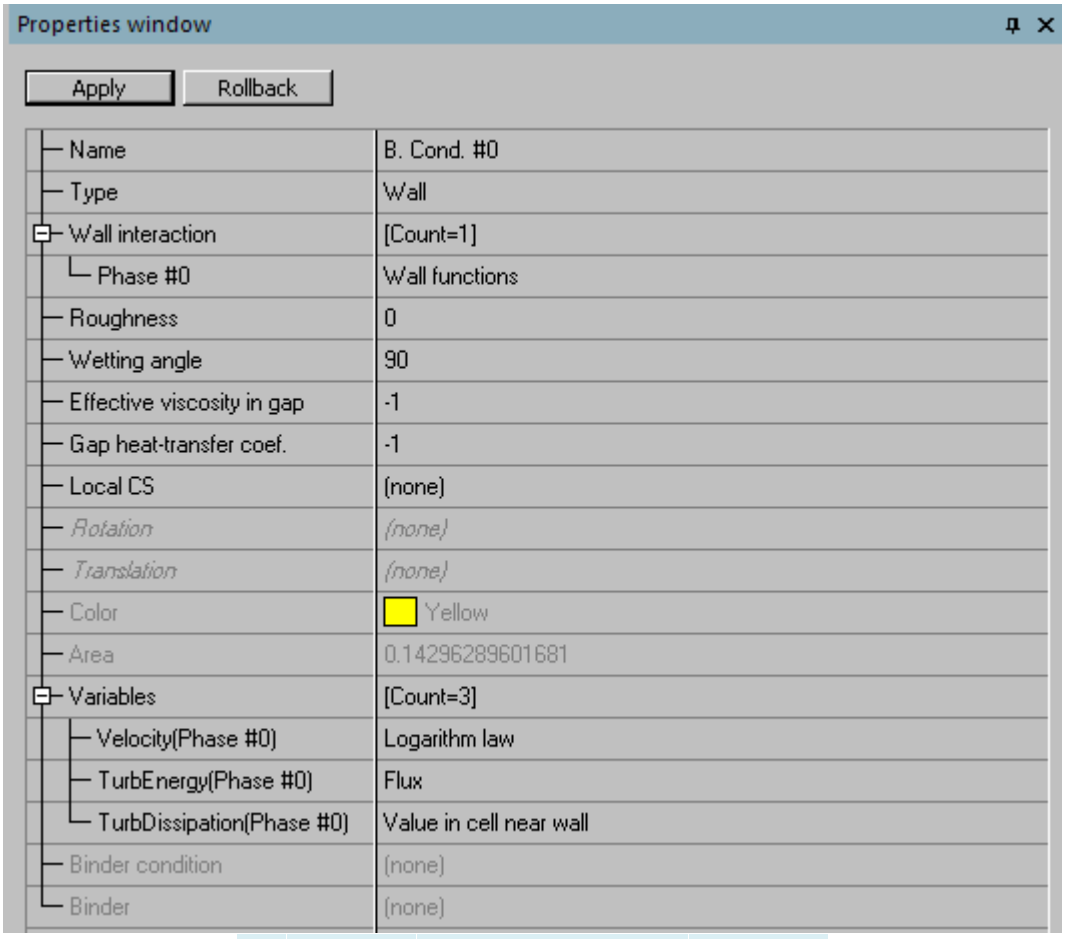
Boundary conditions correspond in the project tree to folders **Subregions > SubRegion #N > Boundary conditions > B. Cond. #N**.

A boundary condition of the **Connected** type that is in an *unbound* state, is marked specially in the project tree:

- in the folder **Subregions > SubRegion #N > Boundary conditions** the element, which corresponds to the boundary condition, is displayed with a "**!**" symbol (for example, ГУ #1)
- such boundary conditions are also listed in the folder **Boundary links > Free BCs**.



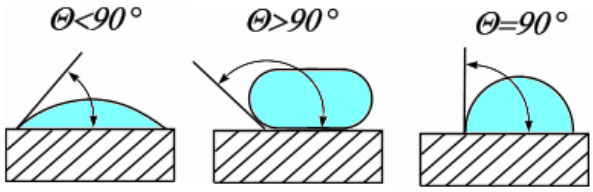
Folder **Boundary conditions** in the project tree



The **Properties** window of the element **B.Cond. #N**

Properties of element «B.Cond. #N»

Parameter	Description	Dimension
Name	Name the boundary condition. The default name depends on the type of the boundary condition. We will refer such names as **B. Cond #N**, where N = 0, 1, 2, ...	

Parameter	Description	Dimension
Type	Type of the boundary condition. Possible options are: • **Wall** for boundary conditions on the surface of solid body • **Symmetry** for symmetry boundary conditions or slipping on a solid surface • **Inlet/Outlet** for boundary conditions at an inlet to the computational domain or at outlet from it • **Free outlet** for free boundary conditions at the outlet from the computational domain • **Connected**[1)] for connected boundary conditions, see section Connected boundary conditions (boundary links) • **Non-reflecting** for non-reflecting boundary conditions at removing from the computational domain • **Wall, ablation** for simulating the ablation	
Wall interaction > Phase #N	The choice of the wall functions for turbulent parameters of the **Phase #N**: • **Wall functions, equilibrium** - boundary conditions are imposed by means of equilibrium wall functions • **Wall functions, nonequilibrium** - boundary conditions are imposed by a non-equilibrium wall functions • **No wall functions** - boundary conditions are imposed without the use of wall functions This parameter is available only if: • the boundary condition is of type: ○ **Wall** ○ *or* **Connected** for which set of **Binders** and related variables missing **Velocity** • *and* **Phase #N** given any physical process to turbulence.	
Roughness	This is the equivalent sand roughness (see Account of wall roughness). Traditional metrics for roughness (Ra, Rz, Rmax) can be converted to the equivalent sand roughness using empiric formulae. To determine maximal influence of the roughness on results of the simulation, you can neglect this conversion and specify just the maximal measured size of the teeth on the rough wall. This parameter is available only when type of the boundary condition is: • **Wall** • *or* **Connected** for which **Binders** are set and there is no **Velocity** in related variables	[m]
Wetting angle	Contact angle of wetting the wall's surface by a liquid. This angle measured in degrees corresponds to the following cases: • 0 - ideal wettability • from 0 to 90 - hydrophilic wettability • 90 - neutral wettability • from 90 to 180 - hydrophobic wettability  Option is available only if the boundary condition is of type: • **Wall** • *or* **Connected**, for which **Binders** are configured and of related variables missing **Velocity**	[Degree]

Parameter	Description	Dimension
Effective viscosity in gap	Effective dynamic viscosity of the medium in the gap[2]. If the specified value is zero or negative (≤0), then the dynamic viscosity of the medium will be used. *Recommendation*: If you need to make leaks from the gap be small to negligible, specify large effective viscosity in gap, which should exceed the molecular viscosity. This parameter is only available when: • the Gap model is enabled • type of the boundary condition is: ○ **Wall** ○ *or* **Connected**, for which **Binders** are configured and related variables don't include missing **Velocity**	[kg $m^{-1}s^{-1}$]
Wall heat-transfer coef.	The heat transfer coefficient of the wall, α_w. If you specify a negative value, then α_w=0. Option is available only if theboundary condition is of type **Connected**, for which **Binders** are configured and of related variables missing **Velocity**.	[W m^{-2} K^{-1}]
Gap heat-transfer coef.	The heat transfer coefficient of the gap on the side of the wall. Effective heat transfer coefficient of the wall gap α_w. If you specify a value of ≤ 0, then the coefficient of thermal conductivity in the gap equal to the coefficient of thermal conductivity of the medium. Option is available only if: • set model clearance • boundary condition is of type: ○ **Wall** ○ *or* **Connected**, for which **Binders** are configured and related variables missing **Velocity**	[W m^{-2} K^{-1}]
Local CS	The following options are available: • **(none)** means that the local coordinate system, which defines the rotation of the boundary conditions, is not specified; the absolute coordinate system will be used • **Local CS #N** specifies a local coordinate system, which is selected among local coordinate systems presented in **Local coordinate systems** By default, the boundary conditions are given in the absolute coordinate system.	
Rotation	This parameter allows you to select a **Rotation**, which has been created before in **Local coordinate systems**. The rotation of the border involves adding to the speed set in the **boundary conditions**, the vector $\omega \times \mathbf{r}_{boundary}$ On the surface of the rotor must be set **Rotation** defined on the **Region**. On the stationary outer cylindrical boundary set **Rotation** is not necessary. The following options are available: • **No** - the rotation of the boundary conditions is not specified; • ***<Rotation>***- the rotation of the boundary conditions to be displayed in the folder of the local coordinate system, which is defined in the **local coordinate system**	

Parameter	Description	Dimension
Translation	This parameter allows you to select a **Translation**, which has been created before in **Local coordinate systems**.	
Color	The color used to highlight groups of facets, on which this boundary condition is set.  To set a color for a boundary condition, you cal also use the **Set color> (color)** command from the context menu of the "B. Cond. #N" element.	
Area	Area of all groups of facets, on which this boundary condition is set. This field is informational only and is not editable.	[m^2]

Variables > *(name of a variable)*	The **Variables** group of parameters contains parameters (sub-elements), which correspond to key variables, calculated in accordance with the given **Model**, and boundary conditions[3]. are specified for them.
	For each variable a *method of its specifying* is selected from the drop-down list (but sometimes the choice is limited by an only one option).
	The list of possible methods depends on the kind of the value, the computational model and type of the boundary condition.
	Examples of variables:
	• **Temperature**
	• **Radiation density**
	• **Velocity** (the options depend on the selected boundary condition). *Examples:* **No slip**, **Slip**, **Track**, **Normal mass velocity**, **Normal velocity with pressure**, **Inlet pressure**, **Total pressure**, **Velocity with pressure**, **Pressure**, **Supersonic outlet**, **Pressure at porous surface**, **Fixed velocity**[4].
	• **Mass frac. *name of the substance***
	• **Variance of fuel**
	• **TurbEnergy**
	• **TurbDissipation**, which can be specified for the following turbulence models:
	  ○ **KES** (standard k-ε model)
	  ○ **KEAKN** (Low-Re k-ε model)
	  ○ **KENL** (nonlinear k-ε model based on works of E.Baglietto)
	• **TurbDissipation specific**, which can be specified for the **SST** turbulence model (model Shear Stress Transport, the boundary condition for ω)
	• **TurbKinViscosity**, which can be specified for the **SA** turbulence model (the Spalart-Allmaras model)
	• **Phase volume**
	• **VOF**
	The actual values of the variables are specified as the **Value** parameters of the elements **B. Cond. #N > *(Name of the variable)*** in the project tree.	

Binder conditions	Names of specified communication conditions	
Binder	The names given ligament	

Notes:

[1] If the alignment of the boundary conditions is carried out after setting the **Binder**, changing the placement of the boundary conditions associated **Binders** deleted. Thus, there is an inquiry whether to delete all connections corresponding to these boundary conditions:

- **Yes**: delete all connections
- **No**: there are requests for the removal of each individual ligament
- **Cancel**: cancel the entire operation

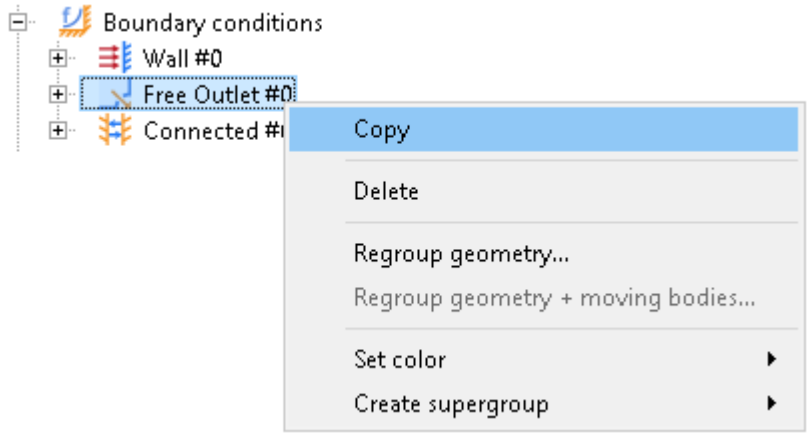
A connected boundary condition is completely determined only after setting the binder. You can *not* specify a connected boundary condition on a **Moving body**. Crossing a surface of a connected boundary condition by a **Moving body** is not allowed, this can cause an incorrect solution of the problem.

[2] This makes sense only when the *Gap model* is enabled.

[3] A set of variables that are defined by the boundary conditions, defined by a set of **physical processes**. See details in the descriptions of the physical processes in the section Theory > Physical processes.

[4] If you set **Velocity** as **Velocity with pressure** then defined velocity vector should have a normal component directed into the computational domain. If desired set speed, directed tangentially to the border, is recommended to set a very small normal component, for example, 10^{-6}m/s.

Context menu of the "B. Cond. #N" element



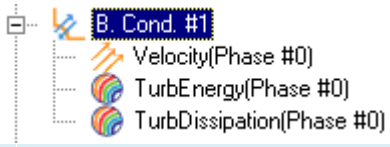
Context menu of the **B. Cond. #N** element in the project tree

Context menu of the **B. Cond. #N** element:

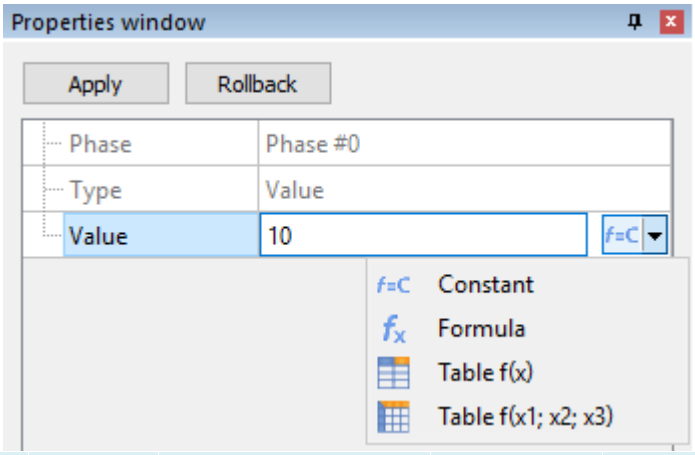
Menu item	Description
Copy	Creating an item, which is a copy of the selected item
Delete	Removing the selected item from the project tree
Regroup geometry	Opening the **Geometry regrouping** dialog box for regrouping the facets, on which certain boundary conditions are set
Regroup geometry + moving bodies	Opening the **Geometry regrouping** dialog box for regrouping the facets, on which certain boundary conditions are set, and also moving bodies
Set color > *(color)*	Specifying the color of groups of facets, on which this boundary condition is set. You can also use the **Color** parameter in properties of the **B. Cond. #N** element.
Create supergroup > In Preprocessor	Creating a **Supergroup** in **Preprocessor** based on the selected boundary condition
Create supergroup > In Postprocessor	Creating a **Supergroup** in **Postprocessor** based on the selected boundary condition

Elements "B.Cond #N > *(variable)*"

Elements **B. Cond #N** are folders, which contain elements corresponding to the variables defined in the boundary condition. Elements **B.Cond #N > *(variable)*** have no context menus.



Folder **B.Cond #N** contains elements corresponding to the variables



The **Properties** window of the element **B. Cond #N >** ***(variable)***

Properties of the element **B. Cond #N >** ***(variable)***:

Parameters that are always present in the Properties window of the element:	
Parameter	**Description**
Phase	A **Phase**, to which the given variable belongs. This field is informational only and is not editable.
Type	Method of specifying the variable, which was specified in the corresponding parameter **Variables >** ***(name of variable)*** in the **Properties** window of the element **Cond. #N** (see above). This field is informational only and is not editable. To change it, make changes in appropriate parameter **Variables >** ***(name of variable)*** in **Properties** of the element **Cond. #N**.
Other parameters (their presence depends on the selected method of specifying the variable):	
Value	This is the variable's value on the **Boundary condition**. This parameter can be used as an external parameter. The **Value** can be specified as: • a constant • a formula (use **Formula editor**) • a table (use **Table Editor**)
Temperature	Specifying the **Temperature** by a specific value
Energy flux **Blackness** **T_inf**	Parameters for simulating the heat transfer when **Temperature** is specified via **Radiation equilibrium**.
Coef. norm. **Coef. tang.**	Parameters for simulating the motion of dispersed particles: restitution coefficients for the normal and tangential components of the particles momentum. These parameters are used when the **Velocity (disp.)** variable is specified via parameters of the **Contact with wall**.
P (BC) type **P static or total** **V (BC) type**	Parameters of **Supersonic inlet** (for BC **Inlet/outlet**).

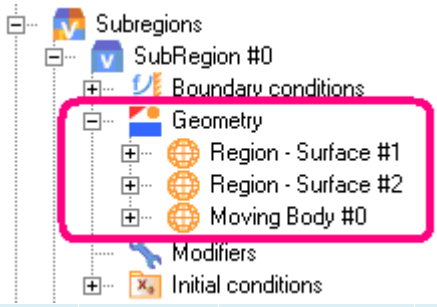
Parameters that are always present in the Properties window of the element:	
Parameter	**Description**
|V| or M	
V or direction > ...	
Voltage Drop	This parameter is specified for the **Electrical potential** variable and specifies the voltage drop in the electrical field on the boundary. The **Voltage Drop**, [V], can be set as a constant, a function or a table. Usually, dependency on the normal current's density on the boundary is specified: See details in the section Theory > Physical processes > Electromagnetohydrodynamics > Boundary conditions >Template "Connected".
Size spectrum	Spectrum of particle sizes of a dispersed **Phase**.

When chemistry is simulated, **Boundary condition #N** obtain child elements that correspond to variables **Mass frac.** not only for **Substances**, but also for **Elements** (conservative scalars). True mass fractions of **Substances**, which are selected as **Elements**, replaced in the user interface of *FlowVision* by mass fractions of **Elements**. In this situation you have to specify mass fractions of **Elements**. For example, in simulation of air dissociation, mass fraction of the **Oxygen Element** at the inlet to the computational domain is always to be equal **0.233333**, no matter on whether atomic oxygen is supplied or not at the inlet. So, no matter if mass fraction of atomic oxygen at the inlet is zero or if it is not zero, mass fraction of the **Oxygen Element** is to be equal **0.233333**.

Mass fractions of **Elements** in **Initial data** are specified similarly.

8.1.8.3.12.2 Folder «SubRegion #N > Geometry»

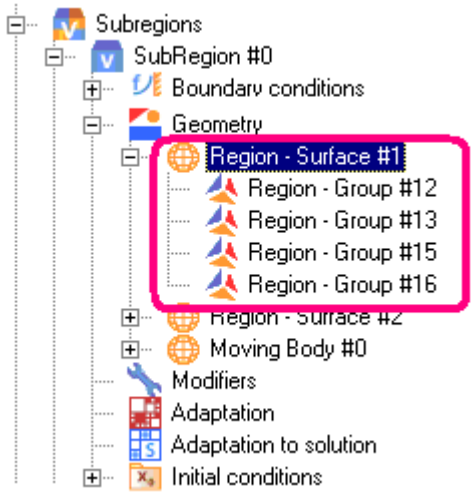
Folder **Subregions > SubRegion #N>Geometry** contains information about the geometry of a **Subregion**. It contains the **Surfaces** of the computational domain, forming this sub-region, and **Surfaces Moving bodies**, created in this **Subregion**.



Folder **Subregions > SubRegion #N > Geometry** in the project tree

The **Subregions > SubRegion #N > Geometry** folder stores **Region- Surface #N** and **Moving Body #N** subfolders.

Folder «Region- Surface #N»

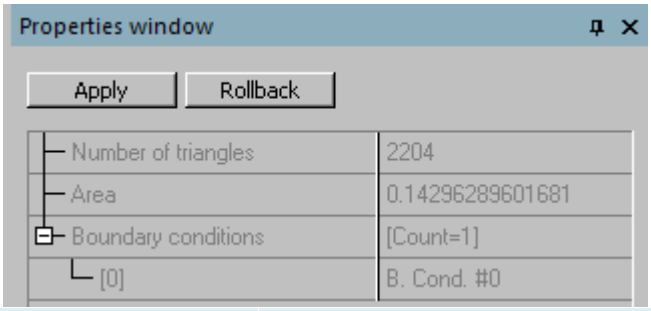


Folder **Geometry > Region - Surface #N** in the project tree

Folder **Subregions > SubRegion #N >Geometry > Region- Surface #N** contains data on one of the surfaces that make up the geometry model of computational **Subregion**.

Folder **Region- Surface #N** created automatically when you create a new subregion.

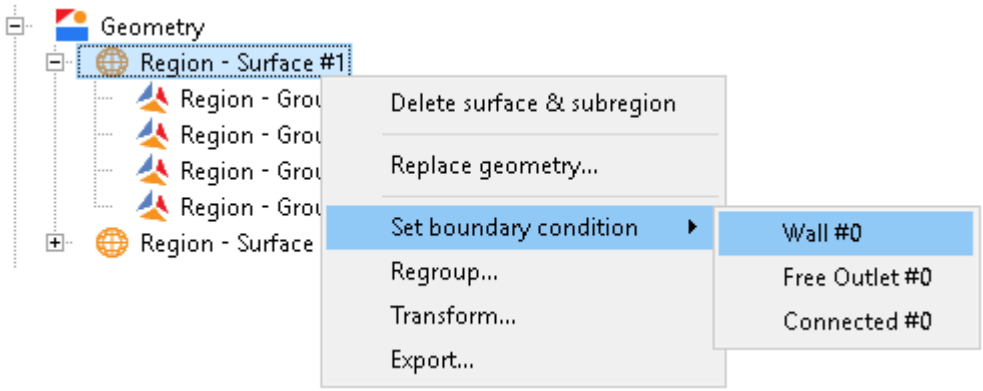
Each folder **Region- Surface #N** contains elements **Region- Group #N**, which correspond to **Groups** of facets.



Properties of the folder **Geometry > Region- Surface #N**

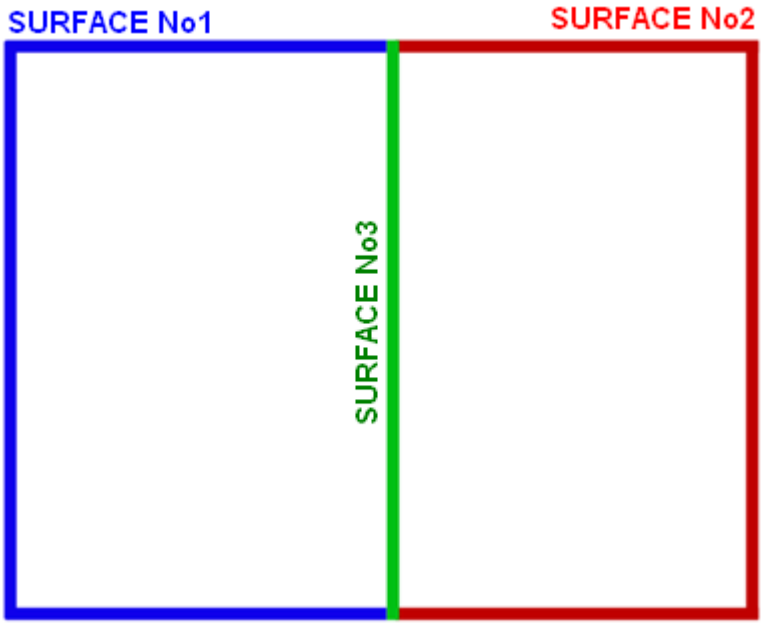
Properties of the folder **Geometry > Region - Surface #N**:

Parameter	Description
Number of triangles	Number of facets on the surface. Field information is not editable
Area	Area of the surface, [m^2]. Field information is not editable
Boundary conditions > [N]	List of boundary conditions which are defined by groups of facets constituting the surface. Field information is not editable



Context menu of the folder **Geometry > Region- Surface #N**

Context menu of the folder **Geometry > Region- Surface #N**:

Menu item	Description
Delete surface & subregion	Removal of the **Surface** that separates two **Subregions**, and removal one of these **Subregions**. This command is available only for the **Surfaces** of the computational domain, if the computational domain contains more than one **Surface**). When multiconnection exists, this command might cause merge of **Surfaces**. For example, if you delete any of the **Surfaces** in the diagram below, the remaining two **Surfaces** will merge into a single closed **Surface**, and the **Complex surface** will become trivial. 
Replace geometry	Replacing the **Surface** with an imported surface. When multiconnection exists in the **Complex surface**, which contains the **Surface**, a warning will be displayed informing you that *all* the geometry of the **Complex surface** will be replaced.
Set boundary condition > B. Cond. #N	Here you can select a boundary condition from the list, which was specified in the folder **Subregions > SubRegion #N> Boundary conditions**. The selected boundary condition will be applied to the **Surface**.
Regroup	Opens the **Regrouping geometry** dialog box to change the facet groups on the **Surface**.

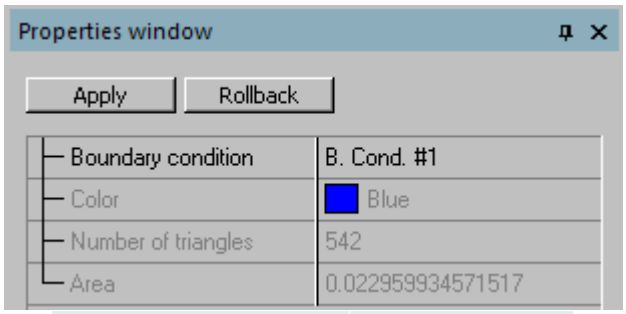
Menu item	Description
Transform	Opens the **Geometry transformation** dialog box to convert the surface. When multiconnection exists in the **Complex surface**, which contains the **Surface**, a warning will be displayed informing you that *all* the geometry of the **Complex surface** will be transformed.
Export	Exporting the geometry model of the surface into a file

Element «Region- Group #N»

Element **Region- Group #N** is displayed in the folder element **Region- Surface #N**.

Element **Region- Group #N** is created automatically when you create an element **Region- Surface #N**.

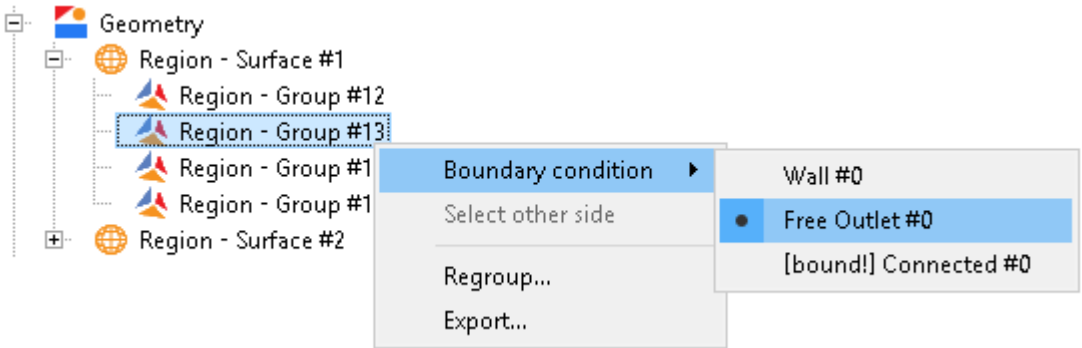
Operations with the element **Region- Group #N** are performed using context menu items to be opened in the project, as well as with the mouse in the **View** window.



Properties of the element **Region- Group #N**

Parameters of the element **Region - Group #N**:

Parameter	Description
The boundary condition	The boundary condition set on the group of facets. Selected from the drop-down list of the boundary conditions.
Color	Color group facets. Field information is not editable
The number of triangles	The number of facets in the group. Field information is not editable.
Area	The surface area of the group of facets, [m^2] Field information is not editable.



Context menu of the element **Region- Group #N** in the **Preprocessor** tab

Context menu of the element **Region- Group #N** in the project tree:

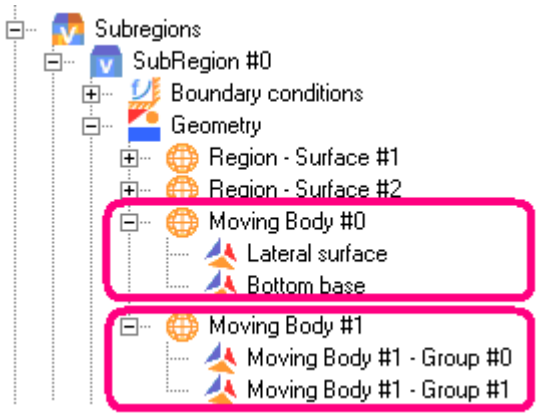
Menu item	Description
Boundary condition > *B. Cond #N*	The choice of the boundary condition from the list means the boundary conditions on the group of facets
Select other side	Reverse the direction normal to the surface groups of facets
Regroup	Opening the **Geometry regrouping** dialog box to regroup facets of this group of facets
Export	Exporting the geometry model of the surface into a file

Element «Moving Body #N»

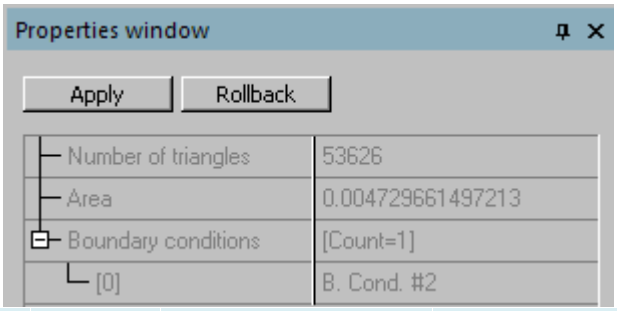
Moving Body #N element contains data on one of the surfaces constituting the geometry model is calculated subregion.

Item is displayed in the folder **Subregions > subregion #N > Geometry** as a folder that contains the elements of **Moving Body #N - Group #M** (these items are displayed only when the setting **Display > Show all group**, see section Basic settings FlowVision; when creating **Moving body** on the basis of **Imported objects** created from standard geometric object, these elements have names that correspond to the surfaces of the geometric object, for example, **Lateral surface** or **Bottom base**).

Moving Body #N element is created automatically when you create a new modifier **Moving body**.



Elements **Moving body #N** in the project tree

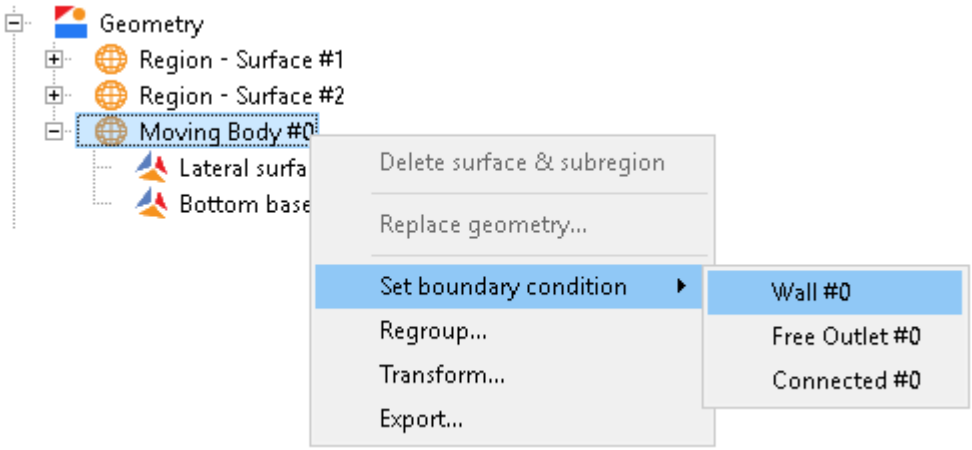


The **Properties** window of the element **Moving body #N**

Parameters of the element **Moving body #N**:

Parameter	Description
Number of triangles	The number of facets on the surface. Field information is not editable.
Area	Surface Area [m^2]. Field information is not editable.

Parameter	Description
Boundary conditions > [K]	List of boundary conditions which are defined by groups of facets constituting the surface. Field information is not editable.



Context menu of the element **Geometry > Moving Body #N**

Context menu of the element **Geometry > Moving Body #N** in the project tree:

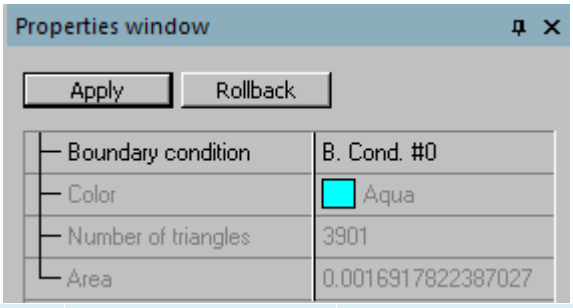
Menu item	Description
Delete surface & subregion	Removal of the surface with the union of the subregions. To the sliding surface, this command is not available.
Replace geometry	Replacing surface separating a subregion.
Set boundary condition > B. Cond. #N	List of boundary conditions from the list in the folder **Subregions > SubRegion #N > Boundary conditions**, which is selected from the boundary condition on the surface
Regroup	Opening the **Geometry regrouping** dialog box to change the facet surface groups
Transform	Opening the **Geometry transformation** dialog box to convert the surface
Export	Exporting the geometry model of the surface into a file

Element «Moving Body #N - Group #M»

Element **Moving Body #N - Group #M** displayed in the folder element **Moving Body #N**.

Element **Moving Body #N - Group #M** created automatically when you create an element **Moving Body #N**.

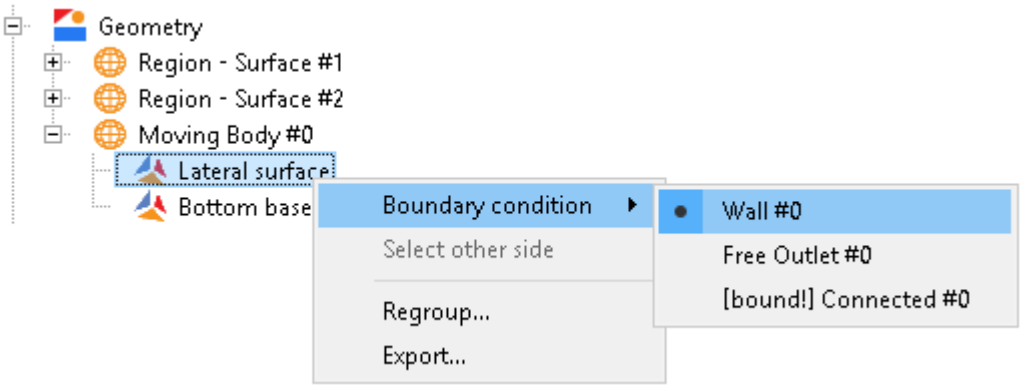
Operations with element **Moving Body #N - Group #M** are performed using context menu items to be opened in the **Project**, as well as with a mouse in the **View** window.



The **Properties** window of the element **Moving Body #N - Group #M**

Parameters of the element **Moving Body #N - Group #M**:

Parameter	Description
Boundary condition	Drop-down list of the boundary conditions, in which you select the boundary condition on this group of facets
Color	Color facet group. Field information is not editable.
Number of triangles	The number of facets in the group. Field information is not editable.
Area	The surface area of the group of facets. Field information is not editable.



Context menu of the element **Moving Body #N - Group #M** in the **Preprocessor** tab

Context menu of the element **Moving Body #N - Group #M**:

The menu command	Description
Boundary condition > *B. Cond #N*	Choice of a boundary condition from the list means setting this boundary conditions on the group of facets
Select the other side	Selecting the other side of the group of facets.
Regroup	Opening the **Geometry regrouping** dialog box to regroup facets of the selected group of facets
Export	Exporting the geometry model of the surface into a file

8.1.8.3.12.3 Folder «Modifiers»

Modifiers are the elements, which allow you to change the solution, and the computational domain.

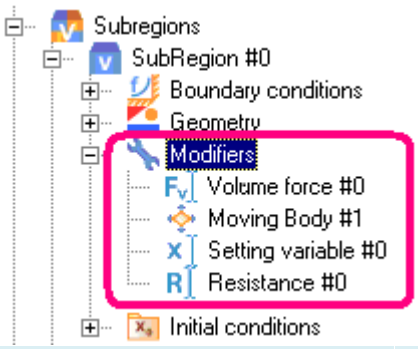
Folders **Subregions > SubRegion #N > Modifiers** contain elements **Modifier #N**.

A **Modifier** element contains its parameters; it is displayed in two folders:

- in the folder **Subregions > SubRegion #N > Modifiers**
- in the folder of the object (**Objects > Imported object #N**), on which the modifier has been built.

There are following types of modifiers:

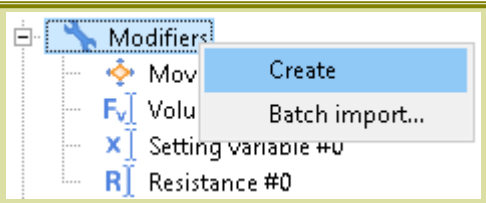
- Moving body
- Setting variable
- Volume force
- Volume heat source
- Ignition/extinction zone (it is available to be created only when **Phase #N > Physical processes > Mass transfer = Combustion**)
- Resistance
- Anisotropic resistance
- Anisotropic thermal conductivity
- Volume External charge
- External Current
- External Induction



Folder **Subregions > SubRegion #N > Modifiers** in the project tree

Priority of a **Modifier** is determined by its position in the list of modifiers in the project tree: the lower a **Modifier** is in the list - the higher is its priority.

For **Moving body** modifiers, their priorities do no matter.



Context menu of the "Modifiers" folder

Command	Description
Create	Create a new **Modifier**
Batch import	Upload multiple imported objects and create **Modifiers** on them

General properties of various types of Modifiers

Parameters of a **Modifier** element are displayed in its **Properties** window; set of these parameters depends on the type of the modifier. Any **Modifier** has the **Activation** section in its **Properties** window.

Activation	(Type=Semiexpendable by time; Start in seconds=1; Duration in seconds
Type	Only once by time
Start in seconds	1
Duration in seconds	2
Period in seconds	*0*
Start in steps	*1*
Duration in steps	*1*
Period in steps	*1*

Section **Activation** in the **Properties** window of the element **Modifier**

Parameters of a Modifier, section **Activation**:

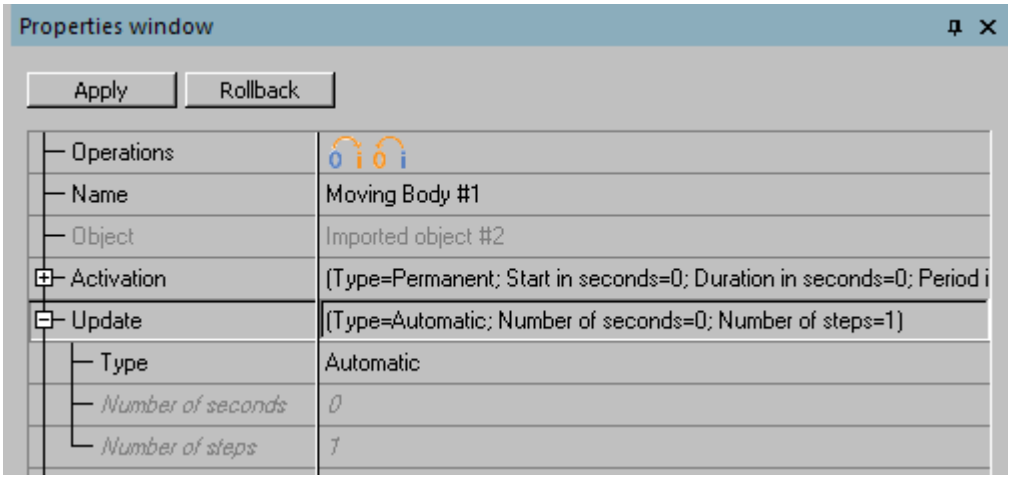
Parameter	Description
Activation > Type	Possible options are: • **Inactive**: the modifier is disabled and has no effect. When this option is selected, the **Modifier**'s icon is displayed in the project tree in faint colors. • **Only once by time**: The modifier is active from the moment of **Start in seconds** during the **Duration in seconds** period

Parameter	Description
	• **Only once by step**: the modifier is active from the moment of **Start in steps** during the **Duration in steps** period • **Permanent**: the modifier is constantly active • **Repetitive by time**: the modifier is activated periodically. The first activation of the modifier occurs at the moment of **Start in seconds**, the modifier remains active during the **Duration in seconds**: period. The modifier activation procedure occurs with a periodicity defined by parameter **Period in seconds**, counted from the moment of **Start in seconds**. • **Repetitive by step**: the modifier is active from the moment of **Start in steps** during the period of **Duration in steps**. Then the activation / deactivation procedure is repeated after a **Period in steps** counted from the moment of **Start in steps**.
Activation > Start in seconds	The start time of the **Modifier** (if **Type** = **Only once by time** or **Type** = **Repetitive by time**).
Activation > Duration in seconds	Duration of activity of the **Modifier**, defined in seconds. This option is available if **Type** =**Only once by time** or **Type** = **Repetitive by time**.
Activation > Period in seconds	Activation time of the **Modifier**, in seconds. This option is available if **Type** = **Repetitive by time**.
Activation > Start in steps	A step in which activation of the **Modifier** occurs. This option is available if **Type** =**Only once by step** or **Type** =**Repetitive by step**.
Activation > Duration in steps	Duration of activity of the **Modifier**, defined in steps. This option is available if **Type** =**Only once by step** or **Type** =**Repetitive by step**.
Activation > Period in steps	Activation period of the **Modifier**, defined in steps. This option is available if **Type**=**Repetitive by step**.

Modifier type «Moving body»

See also section Modifier «Moving body».

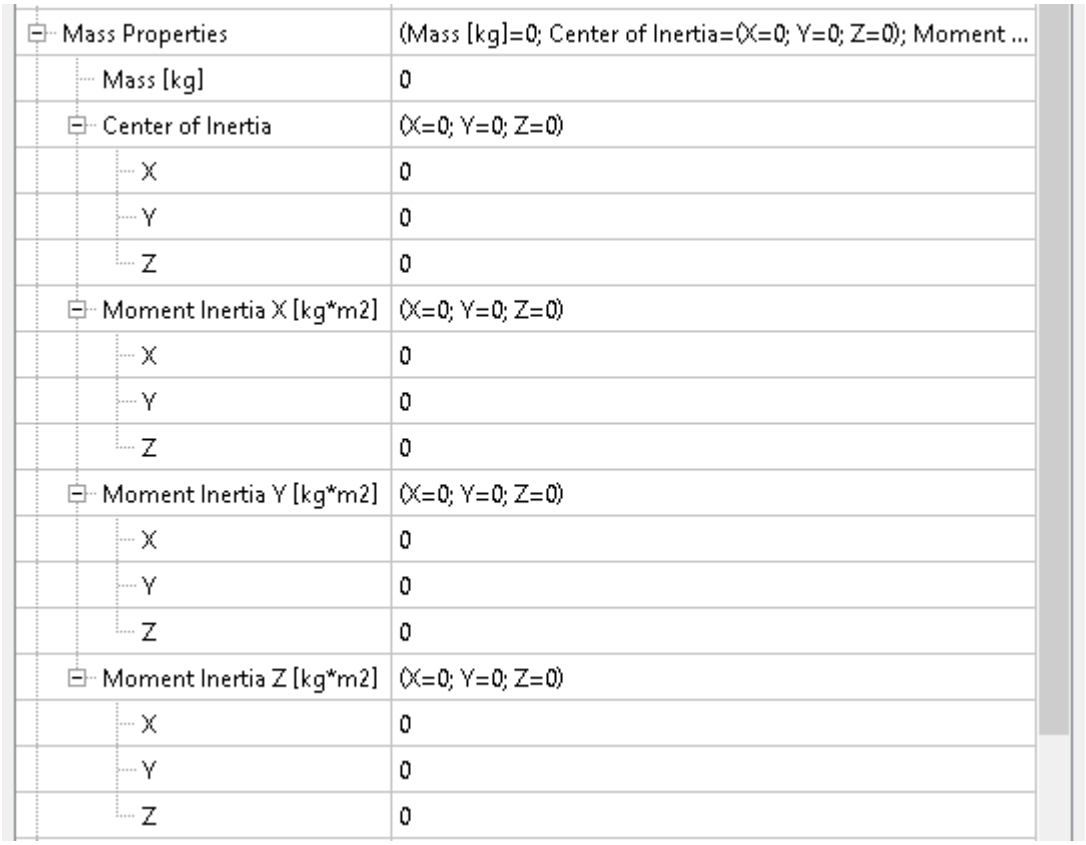
Buttons for operations in the Properties window, parameters «Name», «Object», and parameter groups «Activation» and «Update»



Elements in the Properties window of the modifier «Moving body»: buttons for operations, parameters «Name», «Object», and parameter groups «Activation» and «Update»	
Parameter or button	**Description**
Operations > (Place to initial position)	Puts the **Moving body** into the starting position.

Elements in the Properties window of the modifier «Moving body»: buttons for operations, parameters «Name», «Object», and parameter groups «Activation» and «Update»	
Parameter or button	**Description**
Operations >  (Make current position to be initial)	Makes the initial position of the **Moving body** same as its the current position
Name	Name of the **Modifier**.
Object	**Imported object**, on which the **Moving body** is built. This field is informational only and is not editable.
Activation > ...	Activity parameters of the **Modifier**. See subsection *"General properties of various types of Modifiers"* above.
Update > Type	This update means: • calculation of the new position of the moving body and its movement parameters (forces, torques) • the construction of the computational grid Possible options are: • **Disabled** - update is not made • **Automatic** - updates are made at each time step • **By time** - update is performed regularly by number of seconds • **By step** - updating is performed regularly by the number of steps If **Update > Type = Disabled**, then geometry transformation is *not* done on **Solver** (see recommendations in the section Transformation of geometry model of the computational domain and imported object). If at least **Moving body** has enabled update (**Update > Type ≠ Disabled**), this requires the **Moving Bodies** license. If **Update > Type = Disabled** for all **Moving bodies**, you do not have this license.
Update > Number of seconds	Update period in seconds, if **Type = By time**
Update > Number of steps	Update period in steps, if **Type = By step**

Parameter group «Mass Properties»



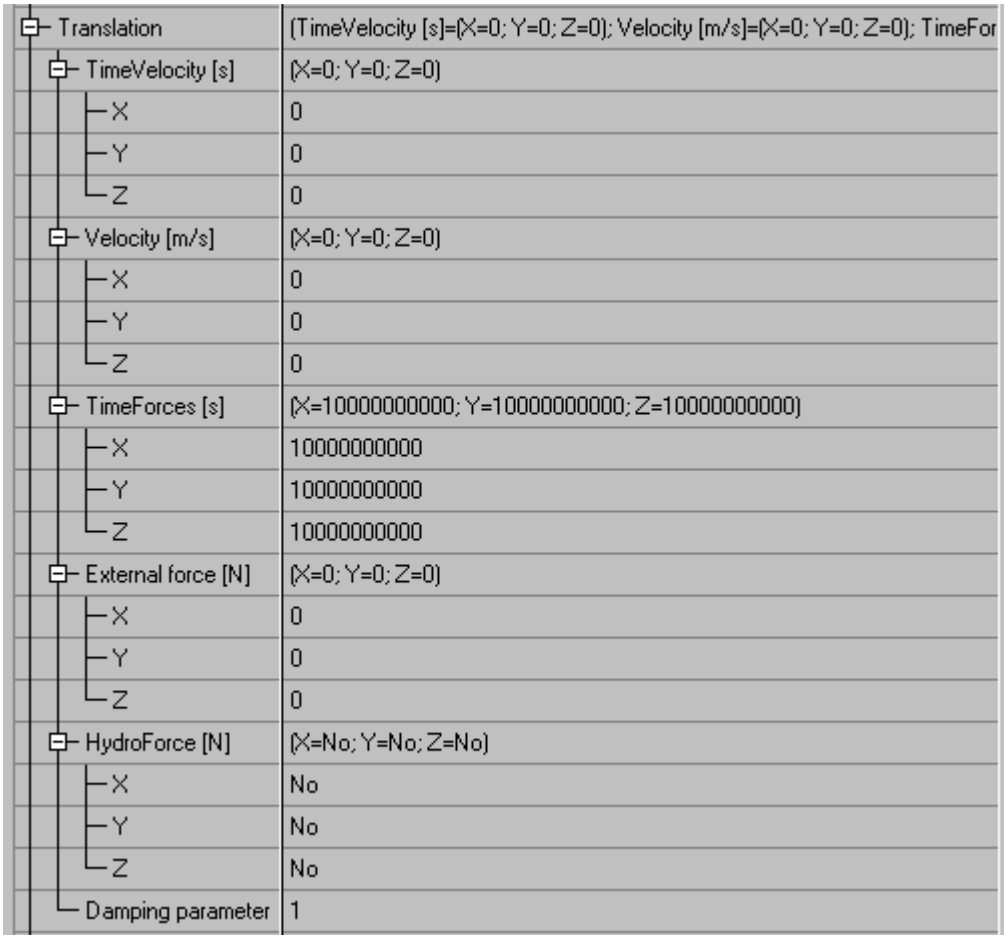
Mass Properties	(Mass [kg]=0; Center of Inertia=(X=0; Y=0; Z=0); Moment ...
Mass [kg]	0
Center of Inertia	(X=0; Y=0; Z=0)
X	0
Y	0
Z	0
Moment Inertia X [kg*m2]	(X=0; Y=0; Z=0)
X	0
Y	0
Z	0
Moment Inertia Y [kg*m2]	(X=0; Y=0; Z=0)
X	0
Y	0
Z	0
Moment Inertia Z [kg*m2]	(X=0; Y=0; Z=0)
X	0
Y	0
Z	0

Parameters of the modifier «Moving body»: parameter group «Mass Properties»

Parameter	Description
Mass Properties > Mass	Mass of the **Moving body**
Mass Properties > Center of Inertia > X	Coordinates, [m], in the local coordinate system of the center of mass of the **Imported object**, which is associated to the **Moving body** modifier
Mass Properties > Center of Inertia > Y	
Mass Properties > Center of Inertia > Z	
Mass Properties > Moment Inertia X > X	Projections on axes X, Y, Z of the local coordinate system of moment of inertia, [kg·m^2], of the **Imported object**, which is associated with the **Moving body** modifier, around axes X, Y, Z of the coordinate system, the center of which is located in the center of mass, and axes are parallel to axes of the local coordinate system
Mass Properties > Moment Inertia X > Y	
Mass Properties > Moment Inertia X > Z	
Mass Properties > Moment Inertia Y > X	
Mass Properties > Moment Inertia Y > Y	
Mass Properties > Moment Inertia Y > Z	
Mass Properties > Moment Inertia Z > X	
Mass Properties > Moment Inertia Z > Y	

Parameters of the modifier «Moving body»: parameter group «Mass Properties»	
Parameter	**Description**
Mass Properties > Moment Inertia Z > Z	

Parameter group «Translation»

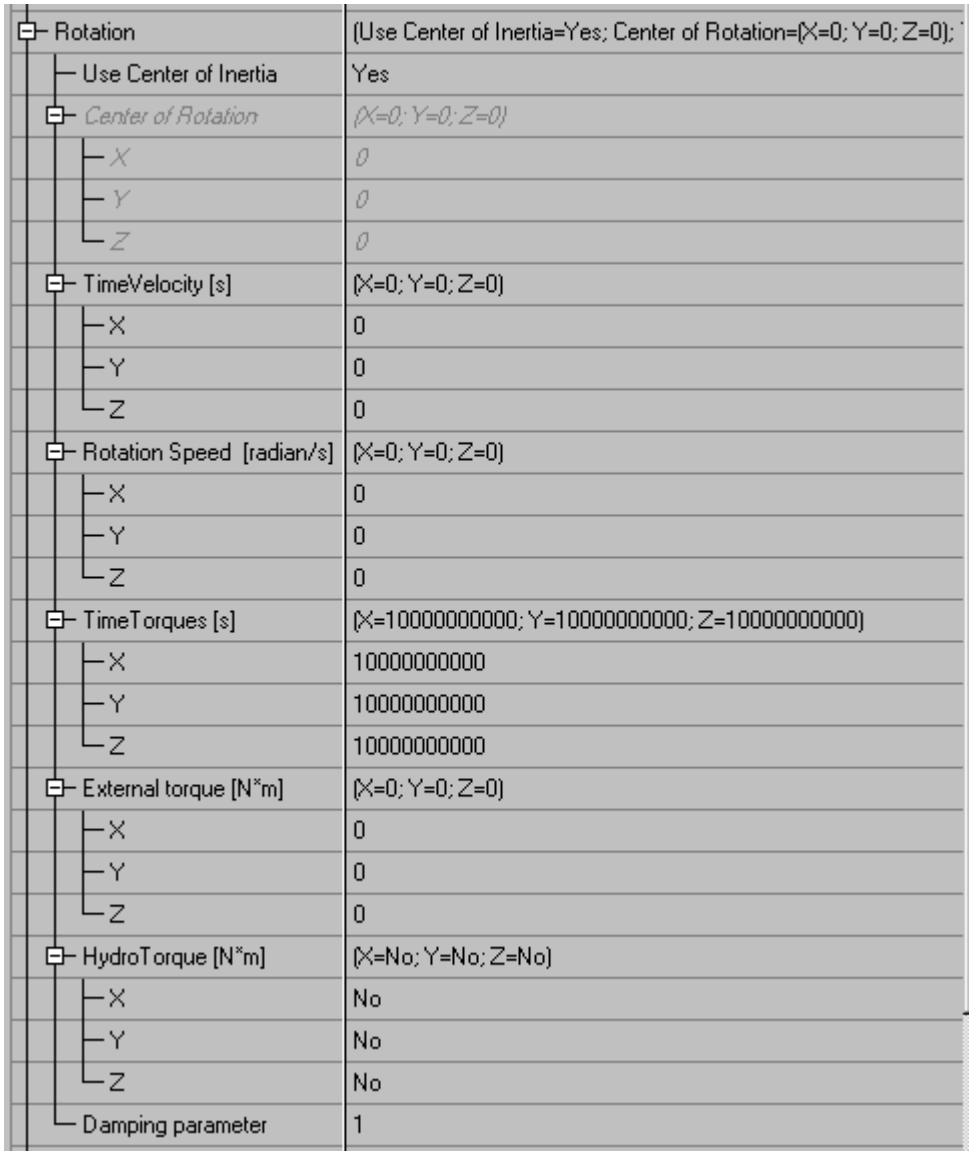


Parameters of the modifier «Moving body»: parameter group «Translation»*)	
Parameter	**Description**
Translation > TimeVelocity > X **Translation > TimeVelocity > Y** **Translation > TimeVelocity > Z**	Parameters $t_{V,\,i}$ for the law of motion of the **Moving body***)
Translation > Velocity > X **Translation > Velocity > Y** **Translation > Velocity > Z**	Projections on the axes (X, Y, Z) of the absolute speed coordinate system $V_{user,i}$ object **Imported object**,is associated with the modifier **Moving body***)
Translation > TimeForces > X **Translation > TimeForces > Y** **Translation > TimeForces > Z**	Parameters $t_{F,\,i}$ for the law of motion of the **Moving body***)
Translation > External force > X **Translation > External force > Y** **Translation > External force > Z**	The projection on an axis (X, Y, Z) the absolute coordinate system force $F_{ext,i}$ acting on the object **Imported object**, which is associated with modifier **Moving body***)
Translational > HydroForce > X	Possible options are:

Parameters of the modifier «Moving body»: parameter group «Translation»*)	
Parameter	**Description**
Translation > HydroForce > Y **Translation > HydroForce > Z**	• **No** - do not take into account the calculated motion along the axis (X, Y, Z) the absolute coordinate system force acting on the object **Imported object**, which is associated modifier **Moving body** • **Yes** - take into account in calculating the motion along the axis X (Y, Z) of the power $F_{hydr,i}$, in the absolute coordinate system acting on the object **Imported object**, which is associated with the modifier **Moving body***)
Translation > Damping parameter	Parameter O_V for damping the forces acting on the **Moving body***)

*) See formulae in the section Modifier «Moving body».

Parameter group «Rotation»



Parameters of the modifier «Moving body»: parameter group «Rotation»*)	
Parameter	**Description**
Rotation > Use Center of Inertia	Possible options are: • **No** - the rotation is around the axes X, Y, Z, parallel to the axes of the absolute coordinate system, and the center of rotation specified by **Rotation > The center of rotation** • **Yes**- rotation occurs around axes X, Y, Z, parallel to the axes of the absolute coordinate system and the center of mass of the **Imported object**, which is associated to the **Moving body** modifier

Parameters of the modifier «Moving body»: parameter group «Rotation»*)	
Parameter	**Description**
Rotation > Center of Rotation > X	Coordinates in the local coordinate system center of rotation of the **Imported object**, which is associated to the **Moving body** modifier
Rotation > Center of Rotation > Y	
Rotation > Center of Rotation > Z	
Rotation > TimeVelocity > X	Parameters $t_{\omega,i}$ for the law of motion of the **Moving body***)
Rotation > TimeVelocity > Y	
Rotation > TimeVelocity > Z	
Rotation > Rotation Speed > X	Projections on the axes (X, Y, Z) of the absolute angular velocity of the object coordinate system **Imported object**, which is associated to the **Moving body***) modifier
Rotation > Rotation Speed > Y	
Rotation > Rotation Speed > Z	
Rotation > TimeTorques > X	Parameters $t_{T,i}$ for the law of motion of the **Moving body***)
Rotation > TimeTorques > Y	
Rotation > TimeTorques > Z	
Rotation > External torque > X	Projections on axes (X, Y, Z) of the absolute coordinate system of external torques $\boldsymbol{T}_{ext}$, acting on the **Imported object**, which is associated to the **Moving body***) modifier
Rotation > External torque > Y	
Rotation > External torque > Z	
Rotation > HydroTorque > X	Possible options are: • **No**: do not take into account in the calculation of the rotation axis (X, Y, Z) the absolute coordinate system torque $\boldsymbol{T}_{hydr}$, acting on the **Imported object**, which is associated to the **Moving body***) modifier • **Yes**: considered in the calculation of the rotation axis (X, Y, Z) the absolute coordinate system torque $\boldsymbol{T}_{hydr}$, acting on the **Imported object**, which is associated to the **Moving body***) modifier
Rotation > HydroTorque > Y	
Rotation > HydroTorque > Z	
Rotation > Damping parameter	Parameter O_{ω} for damping the torques acting on the **Moving body***)

*) See formulae in the section Modifier «Moving body».

Parameter group «Initial position»

Initial position	(Reference point=(X=0; Y=0; Z=0); Rotation definition=By matrix; ...
Operations	
Reference point	(X=0; Y=0; Z=0)
Rotation definition	By matrix
Axis X	(X=0.13887301496588; Y=0.026726124191242; Z=0.98994949366117)
X	0.13887301496588
Y	0.026726124191242
Z	0.98994949366117
Axis Y	(X=0.97050879928687; Y=-0.20257402568564; Z=-0.13067683277564)
X	0.97050879928687
Y	-0.20257402568564
Z	-0.13067683277564
Axis Z	(X=0.19704556889472; Y=0.97890218020149; Z=-0.054070004399175)
X	0.19704556889472
Y	0.97890218020149
Z	-0.054070004399175
Scale	1

Parameter group **Initial position** when **Rotation definition = By matrix**

Initial position	(Reference point=(X=0; Y=0; Z=0); Rotation definition=Around custom ...
Operations	
Reference point	(X=0; Y=0; Z=0)
Rotation definition	Around custom axis
Rotation angle	0
Axis of rotation	(X=1; Y=0; Z=0)
X	1
Y	0
Z	0
Axis X	*(X=0.13887301496588; Y=0.026726124191242; Z=0.98994949366117)*
Axis Y	*(X=0.97050879928687; Y=-0.20257402568564; Z=-0.13067683277564)*
Axis Z	*(X=0.19704556889472; Y=0.97890218020149; Z=-0.054070004399175)*
Scale	1

Parameter group **Initial position** when **Rotation definition = Around custom axis**

Initial position	(Reference point=(X=0; Y=0; Z=0); Rotation definition=Euler angles (Na...
Operations	
Reference point	(X=0; Y=0; Z=0)
Rotation definition	Euler angles (Naval)
Pitch angle	0
Roll angle	0
Yaw angle	0
Axis X	*(X=0.13887301496588; Y=0.026726124191242; Z=0.98994949366117)*
Axis Y	*(X=0.97050879928687; Y=-0.20257402568564; Z=-0.13067683277564)*
Axis Z	*(X=0.19704556889472; Y=0.97890218020149; Z=-0.054070004399175)*
Scale	1

Parameter group **Initial position** when **Rotation definition = Euler angles (Naval)**

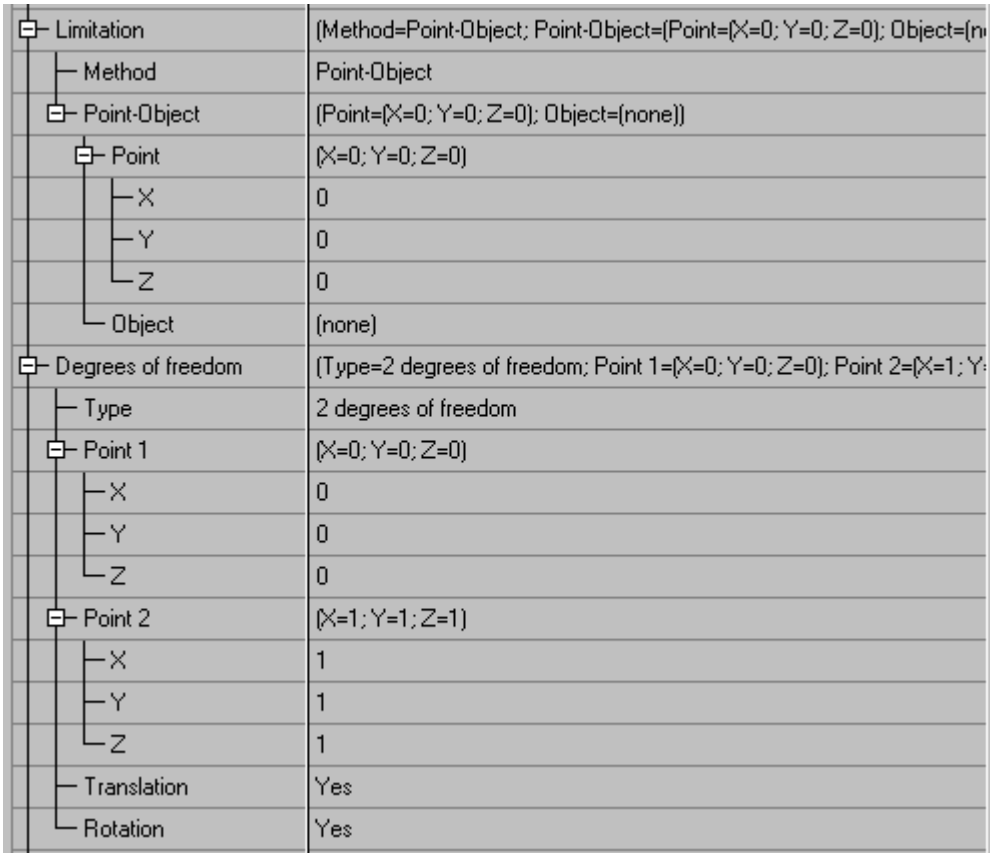
Initial position	(Reference point=(X=0; Y=0; Z=0); Rotation definition=Euler angles (Av...
Operations	
Reference point	(X=0; Y=0; Z=0)
Rotation definition	Euler angles (Aviation)
Yaw angle	0
Pitch angle	0
Roll angle	0
Axis X	*(X=0.13887301496588; Y=0.026726124191242; Z=0.98994949366117)*
Axis Y	*(X=0.97050879928687; Y=-0.20257402568564; Z=-0.13067683277564)*
Axis Z	*(X=0.19704556889472; Y=0.97890218020149; Z=-0.054070004399175)*
Scale	1

Parameter group **Initial position** when **Rotation definition = Euler angles (Aviation)**

Parameters of the modifier «Moving body»: parameter group «Initial position»	
Parameter or button	**Description**
Initial position > Operations >	Opens the dialog box **Relative translation** (see below)
Initial position > Operations >	Opens the dialog box **Coordinate system adjustment** (see below)
Initial position > Operations >	Opens the dialog box **Relative rotation around local axis X** (see below)
Initial position > Operations >	Opens the dialog box **Relative rotation around local axis Y** (see below)
Initial position > Operations >	Opens the dialog box **Relative rotation around local axis Z** (see below)
Initial position > Operations >	Opening the dialog box **Relative scaling** (see below)
Initial position > Reference point > X	Coordinates of the reference point (the center) of the object in the ACS (absolute coordinate system).

Parameters of the modifier «Moving body»: parameter group «Initial position»	
Parameter or button	**Description**
Initial position > Reference point > Y	In the **View** window, the reference point is displayed unit vectors of the LCS (local coordinate system).
Initial position > Reference point > Z	
Initial position > Rotation definition	This parameter defines a method how the rotation is specified. Possible options are: • **By matrix** • **Around custom axis** • **Euler angles (Naval)** • **Euler angles (Aviation)**
Initial position > Axis X> X	Coordinates of the X-axis unit vector in the local coordinate system absolute coordinate system (when **Rotation definition = By matrix**). Clicking the **Apply** button causes automatic normalization of coordinates.
Initial position > Axis X> Y	
Initial position > Axis X> Z	
Initial position > Axis Y> X	Coordinates of the Y-axis unit vector in the local coordinate system absolute coordinate system (when **Rotation definition = By matrix**). Clicking the **Apply** button causes automatic normalization of the normal vector.
Initial position > Axis Y> Y	
Initial position > Axis Y> Z	
Initial position > Axis Z> X	Calculates the coordinates of the unit vector of the local Z axis in absolute coordinate system coordinate system (when **Rotation definition = By matrix**). This field is informational only and is not editable.
Initial position > Axis Z> Y	
Initial position > Axis Z> Z	
Initial position > Rotation angle	Angle of rotation around the axis of rotation in degrees (when **Rotation definition = Around custom axis**)
Initial position > Axis of rotation	Coordinates of the axis of rotation of the unit vector in the local coordinate system coordinate system absolute (when **Rotation definition = Around custom axis**). Clicking the **Apply** button causes automatic normalization of coordinates.
Initial position > Pitch angle *(naval)*	The naval angles of pitch, roll, and yaw. These parameters are available when **Rotation definition = Euler angles (Naval)**. See illustration and detailed description in the section Element «Movement».
Initial position > Roll of heel *(naval)*	
Initial position > Yaw angle *(naval)*	
Initial position > Yaw angle *(aviation)*	The aviation angles of yaw, pitch, and roll. These parameters are available when **Rotation definition**= **Euler angles (Aviation)**. See illustration and detailed description in the section Element «Movement».
Initial position > Pitch angle *(aviation)*	
Initial position > Roll angle *(aviation)*	
Initial position > Scale	Scaling factor of the **Object**'s size. When the scaling factor is changed, the **Mass Properties > ...** parameters will *not* corrected automatically. When necessary, you have to manually calculate and enter new values of these parameters.

Parameter groups «Limitation» and «Degrees of freedom» (limiters for movement and degrees of freedom of the Moving Body)

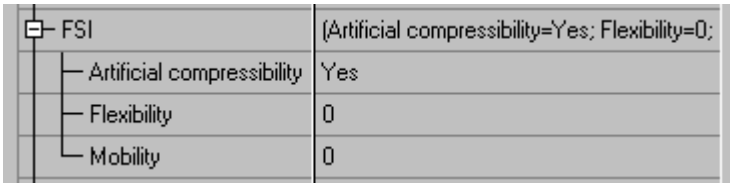


Parameters of the modifier «Moving body»: parameter groups «Limitation» and «Degrees of freedom»	
Parameter	**Description**
Limitation > Method	Possible options are: • **No limits**: the **Moving body** movement is not limited by any geometrical **Object** (including a **Plane**) • **Point-Object**: a point of the **Moving body** is specified, and this point can not go over the surface of some specified **Object** during the **Moving body**'s movement. • **Body above plane**: a **Plane** is specified, which can not be crossed by the **Moving body**. After contact with the **Plane**, the **Moving body** can slide along the **Plane** as well as can move away from it. See illustrations and details in the subsection *"Limitations of the Moving body's movements"* of the section Modifier «Moving body».
Limitation > Body above plane > Object	A **Plane** (selected from a drop-down list), which can not be crossed by the **Moving body**. This parameter is only available when **Limitation > Method = Body above plane**.
Limitation > Point-Object > Point > X	Coordinates of a **Point** (specified in the local coordinate system of the **Imported object**, which is associated with the **Moving body** modifier).
Limitation > Point-Object > Point > Y	This **Point** will not be able to pass over the surface of the **Object**, which is specified by **Limitation > Point-Object > Object**.
Limitation > Point-Object > Point > Z	The **Point** must be specified in the position where the distance from the **Point** to the surface of the **Object** is not less then four times the value of the **Tolerance**. These parameters are only available when **Limitation > Method = Point-Object**.
Limitation > Point-Object > Object	A *standard* geometric **Object** of finite size (i.e. this is a **Cone/Cylinder**, **Box**, or **Ellipsoid/sphere**), the surface of which can not be passed by the

Parameters of the modifier «Moving body»: parameter groups «Limitation» and «Degrees of freedom»	
Parameter	**Description**
	point specified by the parameters **Limitation > Point-Object > Point > X (Y, Z)**. The **Object** is selected from a drop-down list. This parameter is only available when **Limitation > Method = Point-Object**.
Degrees of freedom > Type	Possible options are: • **6 degrees of freedom**: movement and rotation of the **Moving body** without limitations • **2 degrees of freedom**: the movement of the **Moving body** along some axis with a rotation around the axis. The axis is defined by the coordinates of two points in the absolute coordinate system.
Degrees of freedom > Point 1> X **Degrees of freedom > Point 1> Y** **Degrees of freedom > Point 1> Z** **Degrees of freedom > Point 2> X** **Degrees of freedom > Point 2> Y** **Degrees of freedom > Point 2> Z**	Coordinates (specified in the absolute coordinate system) of two points through which the axis of rotation goes. These parameters are only available when **Degrees of freedom > Type = 2 degrees of freedom** is selected.
Degrees of freedom > Translation	Possible options are: • **No**: the **Moving body** cannot move along the specified axis • **Yes**: the **Moving body** can move along the specified axis This parameter is only available when **Degrees of freedom > Type = 2 degrees of freedom** is selected.
Degrees of freedom> Rotation	Possible options are: • **No**: the **Moving body** cannot rotate around the specified axis • **Yes**: the **Moving body** can rotate around the specified axis This parameter is only available when **Degrees of freedom > Type = 2 degrees of freedom** is selected.

See also details in the subsection *"Limitations of the Moving body's movements"* of the section Modifier «Moving body».

Parameter group «FSI»



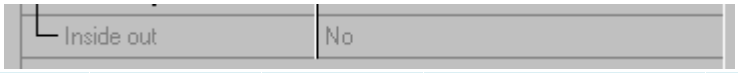
The **FSI** group of parameters of the **Moving body** modifier defines the settings of the subsurface's artificial compressibility in the interaction of the **Moving body** and the simulated fluid. The near-surface artificial compressibility is usually applied in **FSI** problems or when density of the floating body is substantially less than the density of the liquid, in which the body floats.

Parameters of the modifier «Moving body»: parameter group «FSI»	
Parameter	**Description**
Artificial compressibility	This parameters toggles simulation of artificial compressibility.

Parameters of the modifier «Moving body»: parameter group «FSI»	
Parameter	**Description**
	Other parameters in the **FSI** group are available when **Artificial compressibility = Yes**.
Flexibility	The response of the body to the load [m Pa^{-1}] $C = \frac{dl}{dP}$
Mobility	This parameter defines the ratio of transferring the momentum from the fluid to the deformable/mobile body, [m^2 kg^{-1}]. $B = \frac{A_w}{m}$

See details in the section Theory > Physical processes > Motion > Equations.

Parameter «Inside out»

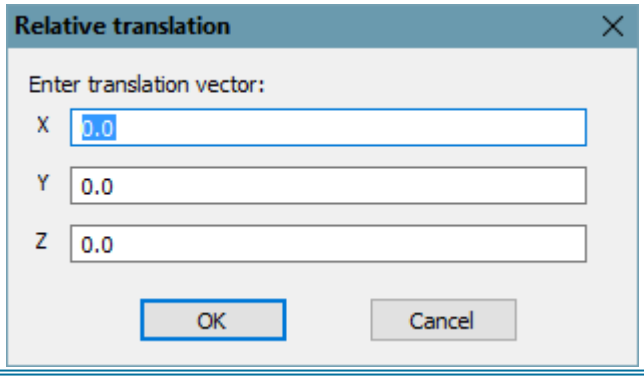


The **Properties** window of the **Moving body** modifier, the parameter **Inside out**

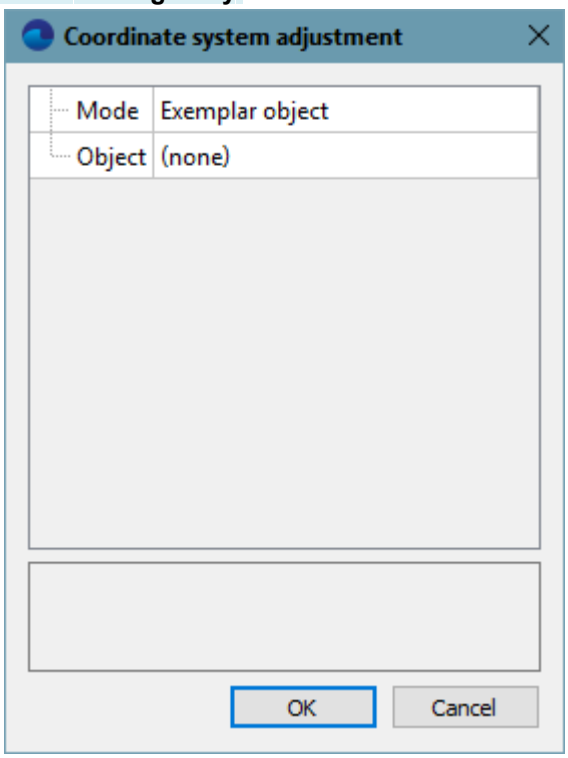
Inside out is the parameter of modifier **Moving body**, which indicates whether the body is inverted (turned inside out, the computational domain is inside the body, not outside).

Parameters of the modifier «Moving body»: parameter «Inside out»	
Parameter	**Description**
Inside out	Possible options are: • **No** - normal to the surface of the body is directed as specified by its creation • **Yes** - normal to the surface of the body is directed in the opposite way as specified in its creation

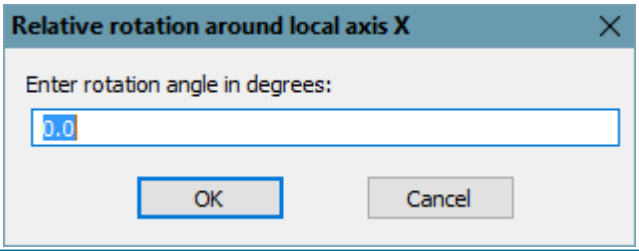
Specifying the relative movement Moving body

Step	Actions
1	In the **Properties** window, click **Initial position > Operations >** . A dialog box **Relative translation** will open: 
2	Specify components of the translation vector of displacement of the **Moving body** (the local coordinate system) in the absolute coordinate system and then click **OK**. In the **Properties** window, coordinates of the reference point will change.
3	Click the **Apply** button. Position of the **Moving body** will change (you will see this in the **View** window).

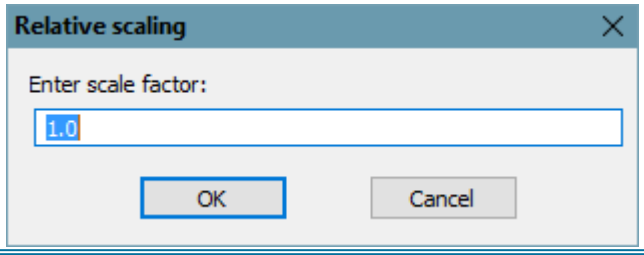
Adjustment the local coordinate system of a Moving body

Step	Actions
1	In the **Properties** window, click **Initial position > Operations >** (**Coordinate system adjustment**). The **Coordinate system adjustment** dialog box will open, using which you can change the local coordinate system (LCS) of the **Moving body**:  Use this dialog box similarly as its use for adjustment of a geometric Object's LCS.
2	In the **Properties** window click **Apply** to activate your changes. Position of the **Moving body** in the **View** window will change.

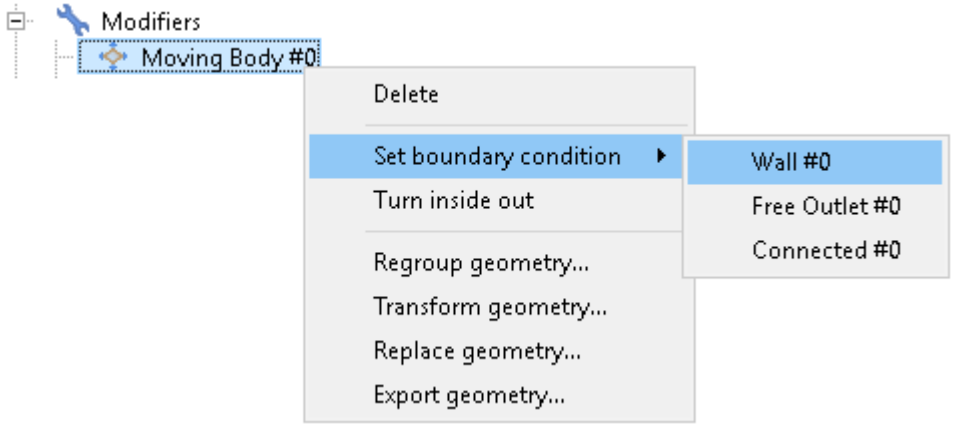
Specifying relative rotation Moving body around an axis

Step	Actions
1	In the **Properties** window, click **Initial position > Operations >** (**Y**, **Z**). A dialog box with respect to rotation around the local axis: 
2	Specify the angle of rotation of the moving body relative to the axis X (Y, Z) of the local coordinate system, and then click **OK**. In the **Properties** window, values of components of the unit vectors of the local coordinate system (**Axis X > ...**, **Axis Y > ...**, **Axis Z > ...**) will change.
3	Click the **Apply** button. In the **View** window, change the position of the moving body.

Specifying the relative scaling Moving body

Step	Actions
1	In the **Properties** window, click **Initial position > Operations >** . A dialog box **Relative scaling** will open: 
2	Specify the factor by which multiply all the coordinates of the moving body in the local coordinate system, and then click **OK**.
3	Click the **Apply** button.

Context menu of the «Moving body» modifier



Context menu of the **Moving body** modifier in the project tree

Context menu of the «Moving body» modifier in the project tree	
Menu item	**Description**
Delete	Removes the **Moving body** from the project tree.
Set boundary condition > *B. Cond. #N*	Sets the selected boundary conditions on the surface of **Moving body** (actually this is setting the **Boundary conditions** on the surface of the **Imported object**, on which this **Moving body** has been created).
Turn inside out	Reverses the direction of the normal to the surface of the **Imported object**, which is associated with the **Moving body** modifier. The program will calculate inside the **Moving body** (and the part of the **Subregion**, which locate outside the **Moving body** will be non-computational).
Regroup geometry	Opens the **Geometry regrouping** dialog box to change of the facets of the object groups the **Imported object**, which is associated with the **Moving body** modifier.
Transform geometry	Opens the Geometry transformation dialog box to transform the surface of the **Imported object**, which is associated with the **Moving body** modifier.
Replace geometry	Opens the **Open** dialog box to import the object, replacing the **Imported object**, which is associated with the **Moving body** modifier.
Export geometry	Exports the geometry model of the surface of the **Imported object**, which is associated with the **Moving body** modifier, into a file in the *WRML* (`*.wrl`) or

Context menu of the «Moving body» modifier in the project tree	
Menu item	**Description**
	3DVision (`*.mesh`) format.

Modifiers of other types (except "Moving body")

Modifiers of different types are displayed in the project tree as nodes with default names and icons, corresponding to their types.

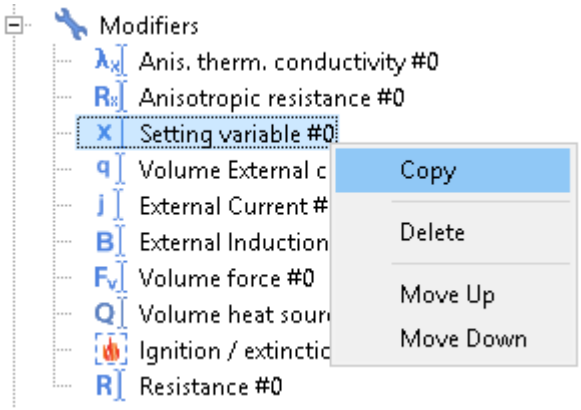
Parameters of "Modifier" elements (except "Moving body")	
Parameter	**Description**
Name	Name of the **Modifier**
Object	The **Object** on which the modifier is set. This field is informational only and is not editable.
Scope	This parameter defines the scope where the **Modifier** acts, which is set on a finite-volume **Object**, in cells near the **Object**'s surface. Possible options are: • **All cells** – in this case the **Modifier** acts in all cells that contain even a small fragment of the **Object**. The area, in which the **Modifier** acts, will be slightly extended (within the computational domain) around the **Modifier**'s **Object**, because the **Modifier** will act in *all cells* that are even partially occupied by the **Object**. • **With centers inside** – in this case the **Modifier** acts in cells which centers locate inside the **Object**. This parameter is available for all **Modifiers** (except **Moving bodies**) that are set on finite-volume **Objects**. See details in the section Modificators, subsection *"Scope of a Modifier"*.
Activation > ...	Setting of the **Modifier**'s activity. See subsection *"General properties of various types of Modifiers"* above.
Variable > ...	This group of parameters defines the variable, volume of which is set by the **Modifier**. This parameters are used for the modifier **Setting variable**.
Variable > Category	Selection of a category for the variable, which will be set by the **Modifier**. Possible options are: • **Common and phase-unrelated variables** • **Variables of phase "Phase #N"** See details in the section Categories of variables.
Variable > Variable	Possible options are: • **(none)**: there is no variable for which the value is set • **<*Variable*>**: a variable from the selected category (**Variable > Category**). The value is set for this variable.
Value	This parameter sets the value of the **Variable**. It is used for the modifier **Setting variable**.
Method[1)]	Method of applying some **Modifiers** in cells. Possible options are: • **Replace in full volume** • **Replace in cropped volume** • **Add in cropped volume** See details in subsection *"Specifics of applying some Modifiers in cells partially or completely filled by the Object"*.
Volume[1)]	This is the real volume, in which the **Modifier** was applied. This field is informational only and is not editable. This field will be filled with data after starting the computation.

Parameters of "Modifier" elements (except "Moving body")	
Parameter	**Description**
<f vol.>*Volume [1)]	This is the integral value of the variable induced by the **Modifier**. This field is informational only and is not editable. This field will be filled with data after starting the computation.
Volume force> X **Volume force> Y** **Volume force> Z**	Vector components of the volume force F_{user} along axes **X**, **Y**, **Z** of the absolute coordinate system, [N m^{-3}]. These parameters are only used for the **Volume force** modifier.
Volume heat source	Distributed heat generation per unit of the **Modifier**'s volume, [W m^{-3}]. When you specify this parameter, please note, that the **Modifier**'s volume is greater then the volume of the **Object**, on which the **Modifier** is built, because the **Modifier** includes not only the cells within the **Object** but all the cells that even partially intersect the **Object** or contact it (see details in the subsection *"Scope of a Modifier"* in the section Modifiers). This parameter is used only for the modifier **Volume heat source**.
Type	Possible options are: • **Ignition** - an infinite rate of combustion is set • **Extinction** - a zero rate of combustion is set This parameter is used for the modifier **Ignition/extinction zone**.
Phase [*)]	The **Phase**, in which the **Modifier** acts. Possible options are: • **(all)**: the **Modifier** acts in all **Phases** where **Physical processes** are enabled, for which the **Modifier** can be applied • **<*Phase*>**: the **Modifier** acts only within the specified **Phase**
Resistance coef.	Coefficient of isotropic resistance D for the volume force proportional to velocity, [kg·m^{-3}·s^{-1}] This parameter is used for the **Resistance** modifier.
Resistance coef. 1-1 [**)]	Element Dxx in the matrix of the anisotropic resistance.The default value is **0**.
Resistance coef. 2-2 [**)]	Element Dyy in the matrix of the anisotropic resistance.The default value is **0**.
Resistance coef. 3-3 [**)]	Element Dzz in the matrix of the anisotropic resistance.The default value is **0**.
Resistance coef. 1-2 [**)]	Element Dxy in the matrix of the anisotropic resistance.The default value is **0**.
Resistance coef. 1-3 [**)]	Element Dxz in the matrix of the anisotropic resistance.The default value is **0**.
Resistance coef. 2-3 [**)]	Element Dyz in the matrix of the anisotropic resistance.The default value is **0**.
Rel.Therm.Conductivity coef. 1-1 [***)]	Element Dxx in the matrix of the anisotropic thermal conductivity.The default value is **1**.
Rel.Therm.Conductivity coef. 2-2 [***)]	Element Dyy in the matrix of the anisotropic thermal conductivity.The default value is **1**.
Rel.Therm.Conductivity coef. 3-3 [***)]	Element Dzz in the matrix of the anisotropic thermal conductivity.The default value is **1**.
Rel.Therm.Conductivity coef. 1-2 [***)]	Element Dxy in the matrix of the anisotropic thermal conductivity.The default value is **0**.
Rel.Therm.Conductivity coef. 1-3 [***)]	Element Dxz in the matrix of the anisotropic thermal conductivity.The default value is **0**.
Rel.Therm.Conductivity coef. 2-3 [***)]	Element Dyz in the matrix of the anisotropic thermal conductivity.The default value is **0**.

Parameters of "Modifier" elements (except "Moving body")	
Parameter	**Description**
Volume External charge	Volume external electric charge, [C/m^3]. This parameter is used for the modifier **Volume External charge**.
Vector of current density > X	Components of the vector of external electric current density, [A/m^2]. These parameters are used for the modifier **External Current**.
Vector of current density > Y	
Vector of current density > Z	
Induction vector > X	Components of the vector of external magnetic induction, [T]. These parameters are used for the modifier **External Induction**.
Induction vector > Y	
Induction vector > Z	

[1)] These parameters are used for modifiers **Setting variable**, **Volume External charge**, **External Current**, **External Induction**, **Volume force**, **Volume heat source**, **Ignition / extinction zone**, **Resistance**.
[*)] This parameter is used for modifiers **Ignition/extinction zone** and **Anisotropic thermal conductivity**.
[**)] These parameters are used for the modifier **Anisotropic resistance** (see details in the section Modifier «Anisotropic resistance»).
[***)] These parameters are used for the modifier **Anisotropic thermal conductivity** (see details in the sectionModifier «Anisotropic thermal conductivity»).

Context menus of Modifiers (except the «Moving body» modifier)



Context menu of a **Modifier** (except **Moving bodies**)

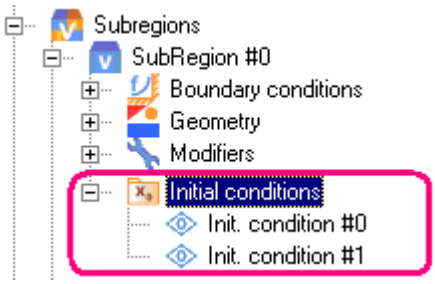
Context menu of a modifier (except the **Moving body** modifier):

Command	Description
Copy	Create a copy of the selected item.
Delete	Remove this **Modifier** from the **Modifiers** folder.
Move Up	Move the **Modifier** one position up in the list, thereby lowering its priority.[*)]
Move Down	Move the **Modifier** one position down in the list, thus increasing its priority.[*)]

[*)] for details of order of applying the **Modifiers** see section Modifiers, subsection *"Order of applying modifiers (priority of modifiers of the same type)"*.

8.1.8.3.12.4 Folder «Initial conditions»

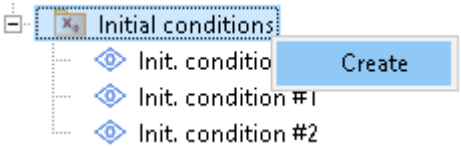
Initial conditions are initial values of variables set in the volume or on the surface of the specified **Object**. Values of the variables are set in some of **Initial data** defined for the **Model**, applied in the **Subregion**.



The **Initial conditions** folder in the project tree:

Folder **Subregions > SubRegion #N > Initial conditions** contains elements **Init. condition #N**.

Context menu of the folder «Initial conditions»



Context menu of the folder **Initial conditions**

Context menu of the folder «Initial conditions»	
Menu item	**Description**
Create	Adding an **Init. condition #N** element into the **Initial conditions** folder

Elements "Init. condition #N"

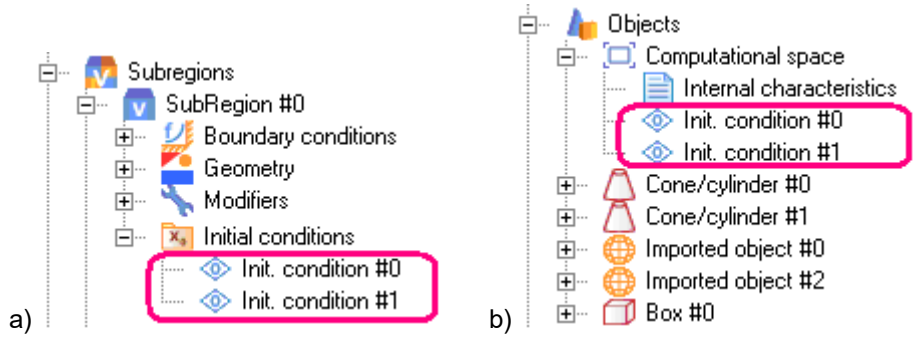
Elements **Init. condition #N** contain the initial conditions for variables. These elements are presented in two locations:

- folder **Subregions > SubRegion #N > Initial conditions**
- folder of the **Object**, in volume or on surface of which the **Initial conditions** act

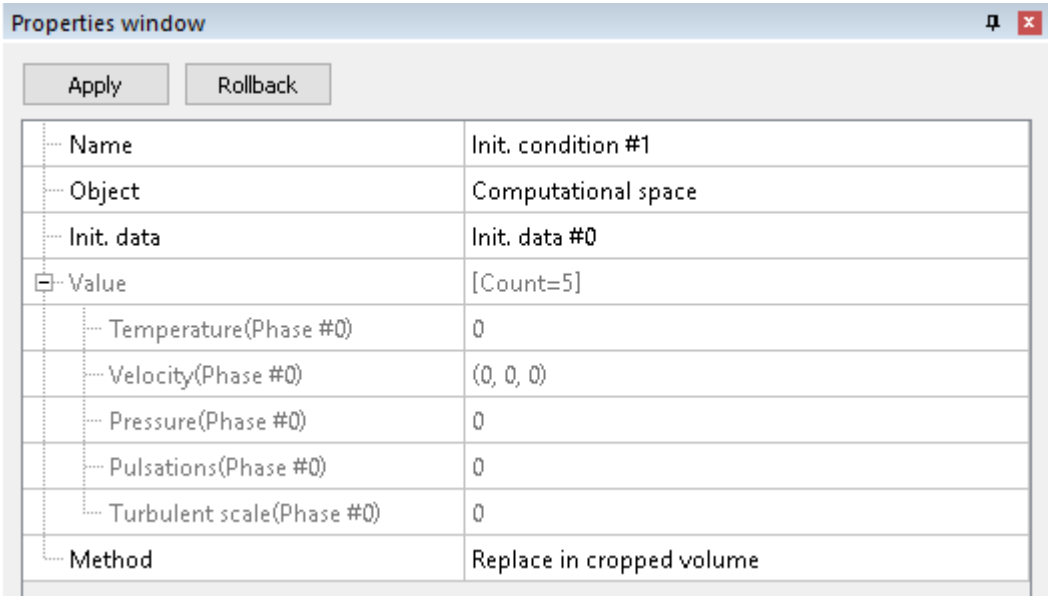
As soon as a **Model** is set in a **Subregion** (after **Substances**, **Phases**, and **Physical processes** are correctly specified), the program automatically creates **Init. condition #0** filled by default with zero data. In properties of **Init. condition #0** the program will automatically set **Object** = **Computational space**, **Init. data** = **Init. data #0**, and **Method** = **Replace in full volume**.

When an **Init. condition #N** element is selected in the project tree, the **Object** (except **Computational space**), on which the **Initial condition** is set, is highlighted in the **View** window.

If **Initial conditions** are not set in some part of the **Subregion**, they will be assumed as zero there.



Elements **Init. condition #N** in the project tree:
a) in the folder **Subregions > SubRegion #N > Initial conditions**; b) in the **Object**, on which the initial conditions are set

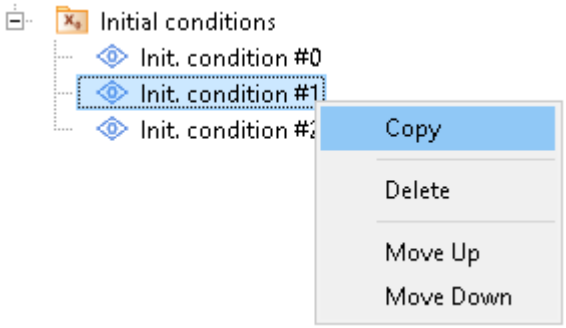


Properties window of the element **Init. condition #N**

Parameters of the element **Init. condition #N**:

Parameters, parameter groups	Description
Name	Name of the initial condition (allows you to change the default name of **"Initial condition #N"**)
Object	**Object** on which the **initial conditions** (chosen from the drop-down list)
Init. data	**Initial data #N** element (selected from the drop-down list). See section Folder Models.
Value > ...	Group of parameters, which reflect the values of the initial data. Information field, not editable.
Method	Method of data initialization. This parameter specifies the algorithm of applying **Initial data #N** in computational cells. Possible options are: • **Replace in full volume** (for Init. condition **#0** that is set on the **Computational space** and for operating in the legacy mode that was used in program's versions before appearing the **Method** parameter) • **Replace in cropped volume** (this option is recommended when **Objects** of different Init. conditions are overlapping)

Parameters, parameter groups	Description
	• **Average in cropped volume** (this option is recommended when **Objects** of different Init. conditions are contacting at some surface between them without gaps or overlapping). This parameter works similarly to the **Method** parameter in properties of the Setting variable modifier. By default, the *automatically created* **Init. condition #0** will be set with **Method = Replace in full volume**. Other **Init. conditions #N** that will be *created manually* will be set with **Method = Replace in cropped volume**. In old project (that did not have this parameter before), **Method = Replace in full volume**. See details in section Initial conditions (subsection *"Applying Initial conditions in cells partially or completely filled by the Object"*).



Context menu of the element **Init. condition #N**

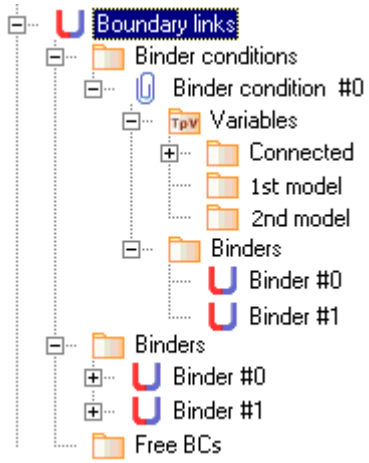
Context menu of the element **Init. condition #N**:

Menu item	Description
Copy	Creating an element, which is a copy of the selected element
Delete	Deleting the selected element from the project tree
Move Up	Moving the **Initial condition #N** element either up or down in the list of elements in the folder **Subregions > SubRegion #N > Initial conditions**.
Move Down	Application of the initial conditions in volumes of **Objects** occurs from top to bottom on the list in the tree of the **Initial conditions**. Thus, if **Objects** of the initial conditions coincide or overlap, the priority is given to **Initial conditions**, which are lower in the list (because they are "overwritten" treated previously **Initial conditions**, which are higher on the list).

See also: section Initial conditions.

8.1.8.3.13 Folder «Boundary links»

Boundary links are intended to define the connection between two subregions using *connected boundary conditions*.



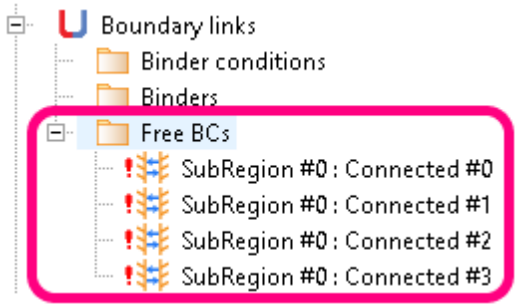
Folder «Boundary links» in the project tree

Folder **Boundary links** contains the following subfolders:

- **Binder conditions**
- **Binders**
- **Free BCs**

They are described in subsections below (description of the subfolder **Free BCs** comes first).

Folder «Boundary links > Free BCs»



Folder **FreeBCs** in the project tree

Folder **Boundary links > Free BCs** contains elements **SubRegion #N: *B.Cond. #M***, appropriate boundary conditions unrelated.

Free (unbound) boundary conditions are the boundary conditions, which have the type **Linked** and which are set on some surface and are *not* used in any **Binder #N**. Boundary conditions are placed into this folder automatically.

Boundary conditions in this list are used to create **Binders**. Once the boundary condition from my **Free BCs** included in any **Binder #N**, it is excluded from the list of **Free BCs**.


Until folder **Free BCs** is not empty, the project can not be started on the computation!

Element **Free BCs > SubRegion #N: *B.Cond. #M*** contains links:

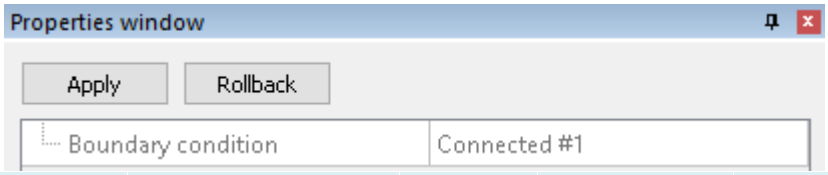
- on **Subregion #N**, where is the surface on which the supplied boundary condition ***B.Cond #M***
- on the boundary condition ***B.Cond #M***, which:
  - has type **Connected**
  - assigned to a group of facets on the surface
  - not connected through any **Binder #N** with another boundarycondition

Element **Free BCs > SubRegion #N: *B.Cond. #M*** created automatically by the program when the requirements listed above.

Element **Free BCs > SubRegion #N: *B.Cond. #M*** deleted from the **Free BCs** in two cases:

- on binding to another element **SubRegion #N: *B.Cond. #M***
- in violation of therequirementslisted above,for example, when you change the type of boundary condition ***B.Cond #M*** or removing the boundary condition ***B.Cond #M*** from a group of facets

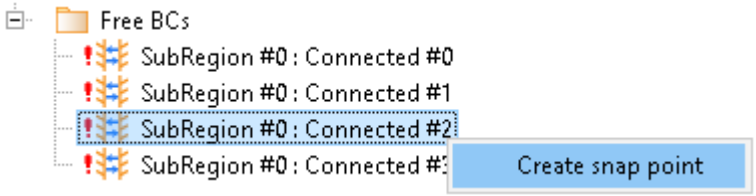
Before sending the project to computation, it is necessary to achieve that folder **Free BCs** become empty and all the boundary conditions of the type **Connected** would correct pairs are connected to the elements **Binder #N**.



The **Properties** window of the element **FreeBCs > SubRegion #N: *B.Cond. #M***

Properties of the element corresponding to an unbound (free) **Boundary condition**:

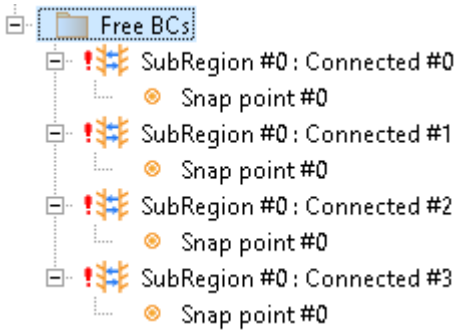
Parameter	Description
Boundary condition	Name of the **Boundary condition**. Field information is not editable



Context menu of the element **Free BCs > SubRegion #N: B.Cond. #M**

Context menu item Free BCs > SubRegion #N: *B.Cond. #M*	
Menu item	**Description**
Create snap point	Create a snap point for an unrelated periodic surface. After applying this command element **Free BCs > SubRegion #N: *B.Cond. #M*** child element will **Snap point #0**, see description in subsection below. After you create the snap points on the bonding surfaces of periodic ensure the correct placement in accordance with the physical meaning of the model and, if necessary, adjust the alignment in the properties window of **Snap points**. See details in the subsection *"Snap points of periodic surfaces"* in the section Periodic surface.

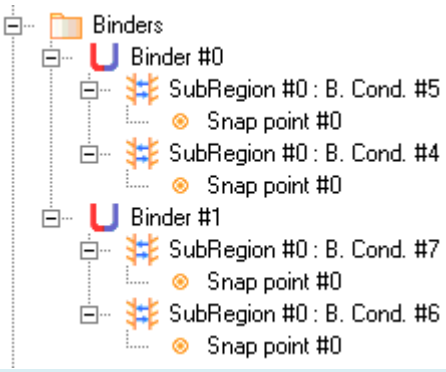
Snap points (needed for periodic surfaces)



Snap points in the elements **Boundary links > Free BCs > SubRegion #N: *B.Cond. #M*** after they are created in the project tree

Elements **SubRegion #N: *B.Cond. #M*** may be present subsidiaries **Snap point #0**,which must be specified on the periodic surfaces. Related to the boundary conditions of other types of fixed points set is not required.

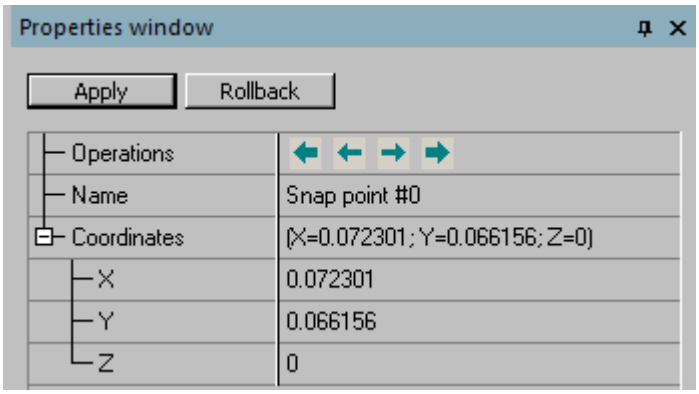
Snap points are created by the user in the elements **Boundary links > Free BCs > SubRegion #N: *B.Cond. #M***,and after the formation of binders are present as child elements in the elements **Boundary links >Binders > Binder #N > SubRegion #N: *B.Cond. #M***:



Snap points in the elements **Boundary links > Binders > Binder #N > SubRegion #N: *B.Cond. #M*** in the project tree

Snap points are used for the correct binding of periodic surfaces.

Location of snap points on the boundary conditions set in the **Properties** window elements **Snap point #0**.



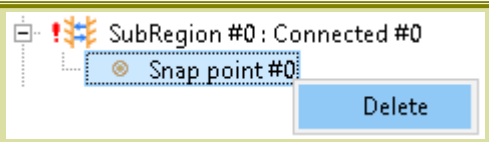
Properties window of a **Snap point**, you can change its location on the boundary condition

Buttons and (thick arrows) move the snap point along the contour of the periodic surface to the next sharp bend of the contour. It is recommended to use these buttons when it allowed by the contour's shape of the periodic surface (if the contour has sharp bends).

Buttons and (thin arrows) move the snap point along the contour of the periodic surface to the next vertex of the polygon, which approximates the curvilinear contour (so these buttons move the snap point relatively slowly and without selecting positions of sharp bends of the contour; this increases the risk of errors.

The choice of snap points must provide a valid comparison of periodic surfaces, corresponding to the physical meaning of the problem being solved (see subsection *"Snap points of periodic surfaces"* in the section Periodic surface).

Also, if desired, in the **Properties** window, the snap point in the **Name** field, you can specify a meaningful name of the snap point instead of the default name **Snap point #0**.



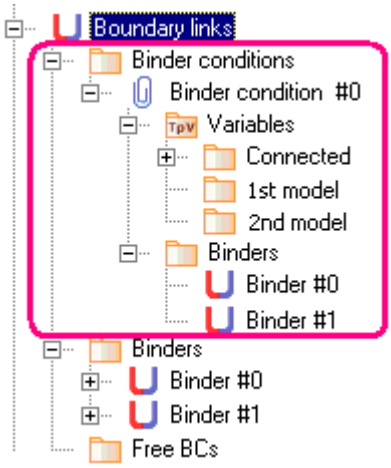
Context menu of the item "Snap point #0"

Menu item	Description
Delete (Only present when the boundary condition on which a snap point has not yet been connected, i.e. located in the **Free BCs**)	Delete the snap point

See also:

- See the "snap point of periodic surfaces" section Periodic surface
- Step by step procedure how to create a periodic boundary condition; see the section Operation with the boundary conditions

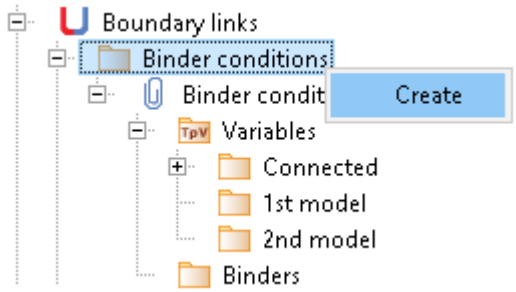
Folder «Boundary links > Binder conditions»



Folder **Boundary links** in the project tree

Binder condition - communication parameters calculated variables two binding models.

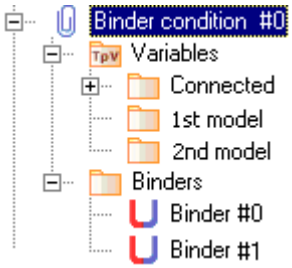
The **Boundary links > Binder conditions** folder stores subfolders **Binder condition #N**. The **Properties** window of the **Boundary links > Binder conditions** is empty (has no parameters).



Context menu of the folder **Binder conditions**

Context menu of the folder "Binder conditions"	
Command	**Description**
Create	Creates a new item **Binder condition #N**. The **Create binder condition** window will open where you should select **Connection type** (possible options are: **Conjugate all variables | Ablation | Conjugate temperature | Periodic surface | Sliding surface |** ***Name of a user module of the Binder type***) and both **Models**, between which the binder conditions is specified.

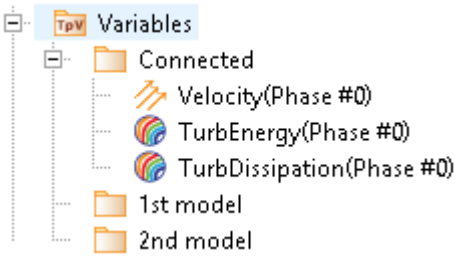
Element «Boundary links >Binder conditions >Binder condition #N»



Element **Boundary links > Binder conditions > Binder condition #N** in the project tree

Element **Binder condition #N** is displayed in the project tree as a folder containing two subfolders:

- **Variables**
- **Binders**



Subfolder **Boundary links > Binder conditions > Binder condition #N > Variables** in the project tree

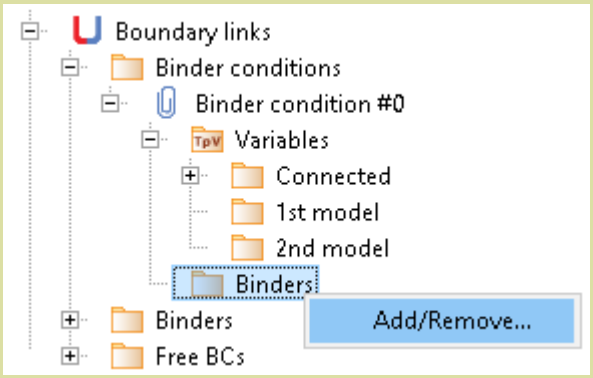
Subfolder **Boundary links > Binder conditions > Binder condition #N > Variables** is used to display the variables on which binding of the boundary conditions has been done (the folder **Connected**) or hasn't been done (folders **1st model** and **2nd model**). This folder contains three subfolders:

- **Connected** - the list of connected variables
- **1st model**- a list of unbound variables in the first model
- **2nd model**- a list of unbound variables in the second model

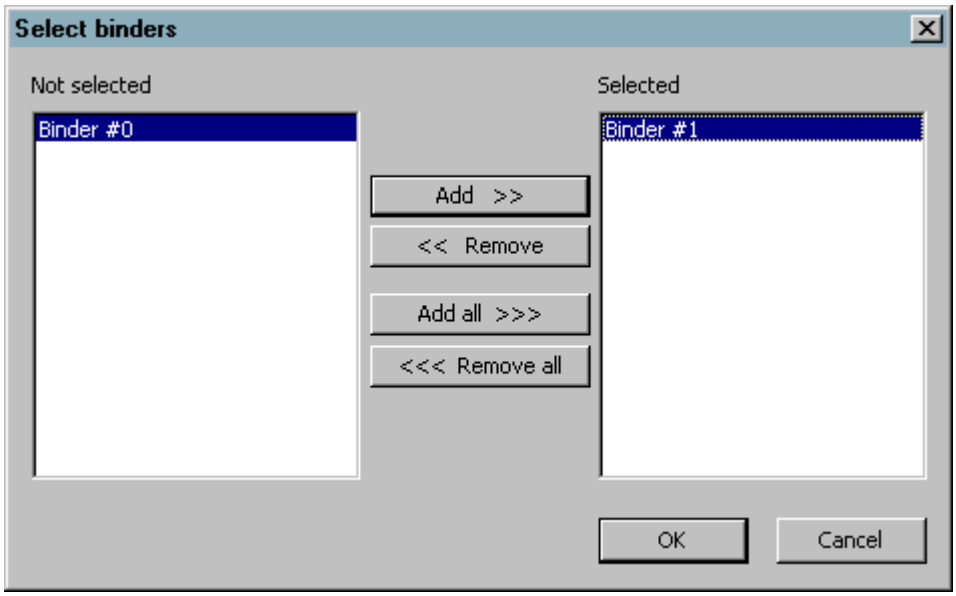
Depending on the type of the connected boundary condition **variable** may have specific parameters, such as drop in temperature and/or pressure periodic surface.

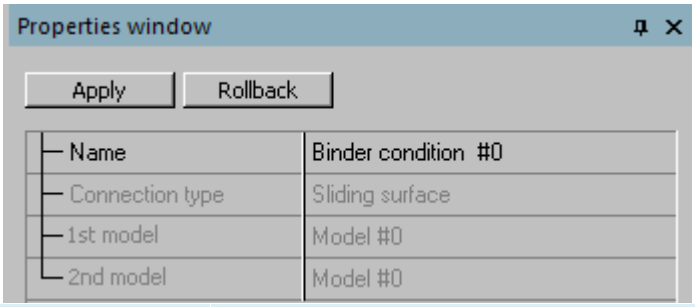
Subfolder **Boundary links > Binder conditions > Binder condition #N > Binders** in the project tree

Subfolder **Boundary links > Binder conditions > Binder condition #N > Binders** contains items **Binder #N**, on which a **Binder condition #N**.



The context menu of the folder "Boundary links > Binder conditions > Binder condition #N > Binders"

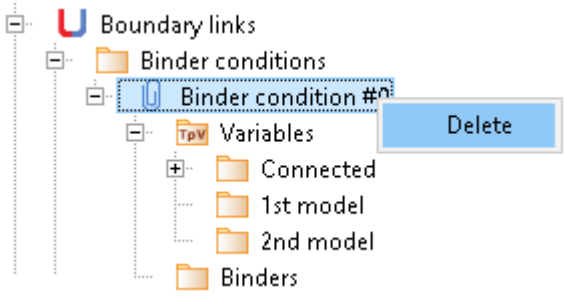
Command	Description
Add/remove	Select to open the **Select binders** dialog box where you can specify which binder will use **Binder condition #N**:  Generate lists of bundles: • select the added bundles from the **Not selected** list, and click **Add**; to add all bundles at once, click **Add all** • select excluded ligament in the **Selected** and click **Remove**; to exclude from all binders, click **Remove all** When complete the distribution of the binders on the **Add** and **Remove**, click **OK**.  In some cases, the conditions for connection type **Sliding surface** when comparing the **conditions due** to **a bunch of** program can offer in the **Select binders** of acceptable and unacceptable **Binders** options when the boundary conditions do not correspond to some **Sliding surface #N** in the project tree. Selecting this option will make the project unacceptable incorrect and will lead to an inability to start the project on the computation. Do not make mistakes when choosing a ligament in the **Select** ligament.



The **Properties** window of the element **Boundary links > Binder conditions > Binder condition #N**

Parameters of the element **Binder condition #N**:

Parameter	Description
Name	Name of the communication conditions (the default name **Binder condition #N**,which, if desired, can be changed to another, such as a **light** or **surface ends of the section of the heat** exchanger).
Connection type	Link type, is being created communication conditions. Possible options are: • **Conjugate all variables**- communication through the permeable surface of the medium, does not affect the flow and heat and mass transfer. For all variables put the condition of continuity and variable flow. • **Ablation** • **Conjugate temperature**- communication through a solid surface. For all variables except temperature are set the boundary conditions on the solid surface (**Wall**), for the temperature - the continuity condition of temperature and heat flux through the surface. • **Periodic surface** - the link between repeated identical segments, which are conventionally divided the original geometry model (possibly with a given change in temperature or pressure), see section Periodic surface. • **Sliding surface** is the relationship between sub-areas, which are rotated relative to each other, sliding along a surface,see section Sliding surface. • ***Name of a user module of the Binder type*** is name of a **User module** of the **Binder** type, which has been loaded into the project This field is read-only and can not be changed.
1st model **2nd model**	Name models in neighboring subregions (for the **Periodic surface** model used in the periodic fragment). Information field, not editable.



Context menu of the element **Binder condition #N** in the project tree

Context menu of the **Binder condition #N** element in the project tree:

Menu item	Description
Delete	Deleting the selected element from the project tree

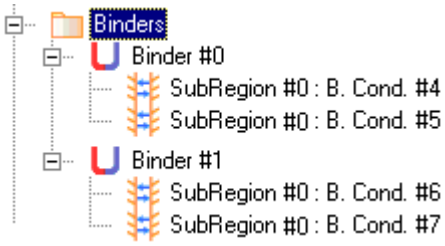
The procedure for creating a binder condition:

Step	Actions

1	From the context menu of the **Binder conditions** folder, select **Create**.
2	In the **Create binder condition** window that opens, specify: • **Connection type** - the type of the binder condition (possible options are: **Conjugate all variables | Ablation | Conjugate temperature | Periodic surface | Sliding surface |** ***Name of a user module of the Binder type***) • **1st model** - the first model, which is bound • **2nd model** - the second model, which is bound

After creation a binder condition, it is impossible to edit its parameters!

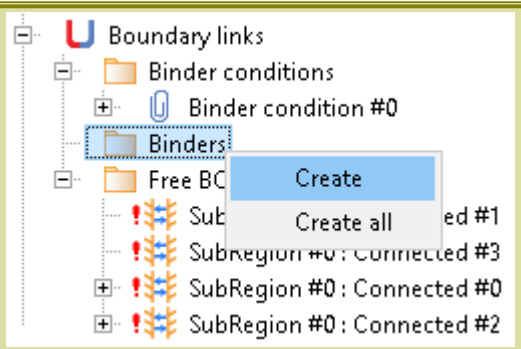
Folder «Boundary links > Binders»



Folder **Binders** in the project tree

Binder is an element that establishes a correspondence between the two surfaces. You can specify a **Binder** between **Free BCs** only.

A **Binder** can be set between two sides of a surface and between spatially spaced *periodic surfaces*.



Context menu of the folder "Binders"

Command	Description
Create	Creating a new element **Binder #N**. A dialog box will open where you have to select a pair of boundary conditions that are present in the list of **Free BCs**. Selection of the first boundary condition limits ability for selection the second boundary condition, this reduces your chances to make a mistake.
Create all	Automatic creation of the binders of all the boundary conditions in the folder **Boundary links > Free BCs**

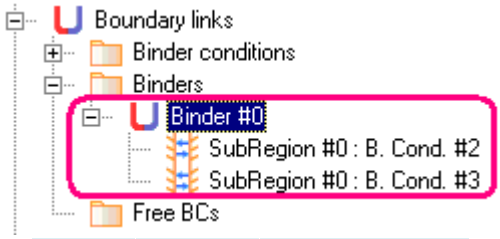
In the window for creation a new **Binder #N** pairs of boundary conditions are selected from a list.

After you create a bunch of editing its parameters can not be!

You can only delete a bunch that are not used in a connection!

If the alignment of related boundary conditions changed after the creation of the bunch, a bunch of deleted automatically.

Element «Boundary links > Binders > Binder #N»



Element **Binder #N** in the project tree

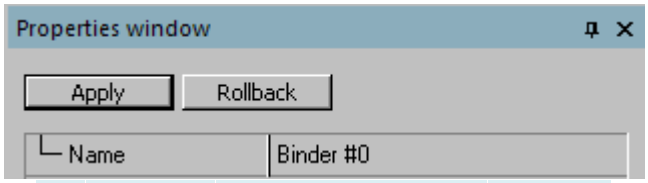
Element **Binder #N** displayed in a tree in the Project Folder **Boundary links > Binders** as a folder comprising two elements, each of which represents the attachment surface. The standard names of these objects (**SubRegion #N: *B.Cond. #M***) indicate:

- **Subregion #N**, where is the surface on which delivered connected boundary condition
- boundary condition **B. Cond #M**, which:
  - has type **Bound**
  - assigned to a group of facets on the surface
  - not connected through the element **together** with other boundary condition

Element **Boundary links> Binders> Binder #N**, which does not set the constraintDisplayed with a warning symbol (**!**).

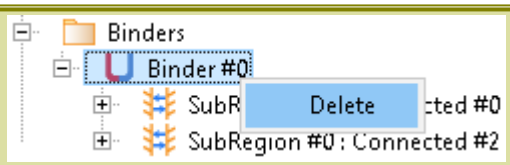
Before starting the calculation is necessary to achieve the lack of binders, which are not set connection.

In the**Properties** window element **Bundle #N** there is only one parameter - the **name**.



The **Properties** window of the element **Binder #N**

Properties of the element "Binder #N"	
Parameter	**Description**
Name	Name of the binder. If desired, you can change the default name **Binder #N** to another.



Context menu of the element "Binder #N" in the project tree

Menu item	Description
Delete	Deleting the selected element from the project tree

Procedure to bind the boundary conditions

- In a **Subregion Boundary conditions** in the folder created by the boundary condition which is the type of **Connected**.
- The boundary condition **Bound**mounted on some surface (if it was not automatically placed on the surface).
- Once some **Connected** boundary condition is set on the surface, in the folder **Free BCs** appears corresponding element.
- For periodic surfaces need to set snap points (while they are still in the folder **Free BCs**).

Of a pair of elements present in the folder **Free BCs**, creates an element **Bundle #N**, which combines a pair of two boundary conditions[1)]

- In the folder is created **Conditions connection** element **Binder condition #N** and determined its properties:
  - Type of connection
  - **1st model** and **2nd model** (they can be the same)
  - A set of related variables and variables models
  - Type of boundary conditions for each of the bound or unbound variable
- For each **Binder Binder condition** specified in the list of pre-built

Until the project is unrelated boundary conditions or ligament, which is not set the constraint, the project will not run on computation!

Boundary conditions appropriate and **Binders** are marked with a "**!**" tree **Preprocessor**.

Note:

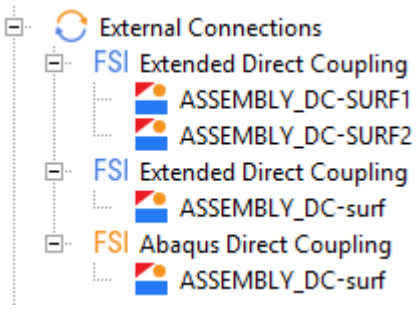
When a **Binder** is created the both uncoupled boundary conditions included in the **Binder**, are automatically removed from folder **Free BCs** After all the settings in the**Properties**window of the **Binder**.

See also:

- Connected boundary conditions (boundary links)
- Periodic surface
- Sliding surface

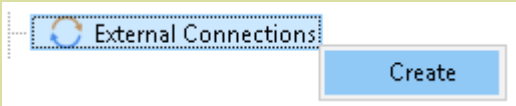
8.1.8.3.14 Folder «External Connections»

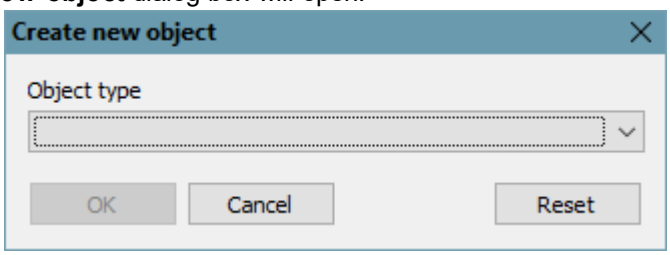
The **External Connections** folder contains elements that correspond to connections (*connectors*) between *FlowVision* and external programs.



Folder **External Connections** in the project tree

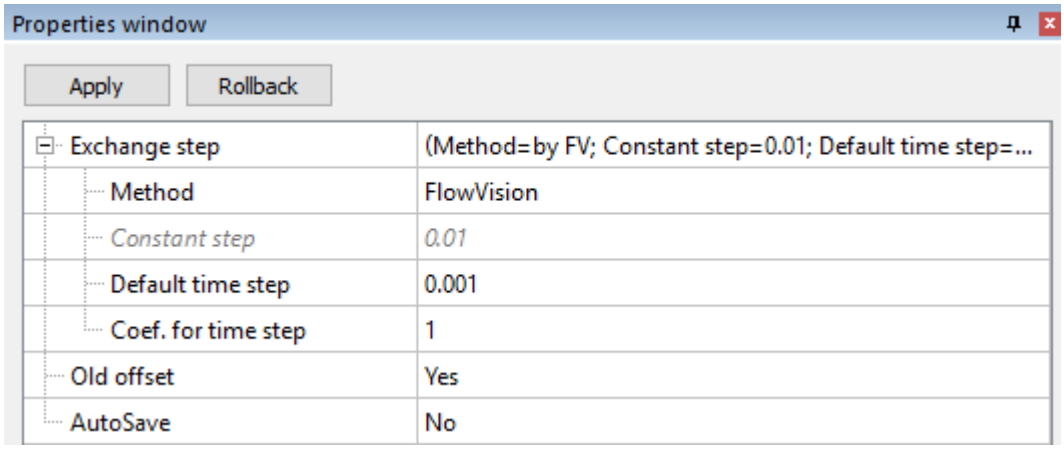
Context menu of the folder "External Connections"



Menu item	Description
Create	Add into the **External Connections** an element that corresponds to a connector between *FlowVision* and an external program. The **Create new object** dialog box will open:  In this dialog box you have to select the type of the connector, which is to be created, from the **Object type** field, and click **OK**.

Menu item	Description
	For connectors with types **Abaqus CSE**, **Abaqus Direct Coupling**, **Extended Direct Coupling** you will have to select an `inp` file, which contain the *Abaqus* project, using an operation system's dialog box for access to files. For connectors with type **Arbitrary external connection** you will have to create **Exchange surfaces** manually.

Parameters of the folder "External Connections"



The **Properties** window of the folder **External Connections**

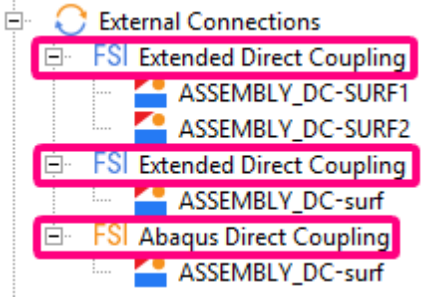
Parameter	Description
Exchange step	Parameters that specify settings of the exchange step between *FlowVision* and external programs.
Exchange step > Method	Method of specifying time step for interaction. Possible options are: • **FlowVision**: the exchange step is set based on the *FlowVision*'s time step • **User value**: the exchange step is set by the user as a constant, formula or table value • ***(connector)***: the exchange step is set by the external program
Exchange step > Constant step	Exchange time step value, [s], predefined by the user. This value can be set by a constant, formula, or table. This parameter is available when **Exchange step > Method = User value**.
Exchange step > Default time step	Exchange time step that will be used if no exchange time step is specified by any of the connectors. Generally this value might be used at the beginning of interaction between *FlowVision* and the external program.
Exchange step > Coef. for time step	Coefficient for the exchange time step. This value can be set by a constant, formula, or table.
Old offset	Use the old offset method for removal of self-intersections. Possible options are: **Yes | No**. The default value is **Yes**.
AutoSave	This parameter is used for debugging the project, it allows you to automatically save the project after receiving data from the external program. *FlowVision* saves the whole computational grid and geometry from the external program *at each exchange with rewriting the previous steps*. If the computation is unsuccessful, it is possible to reproduce the error, which occurs in the iteration before the crash. When you contact the support service, provide them with the appropriate project.

Parameter	Description
	If self-intersections of the geometry were found, *FlowVision* automatically uploads the geometry, which has been received from the external program, in formats **`*.wrl`** and **`*.ngeom`**. Possible options are: **Yes** | **No**. The default value is **No**.

8.1.8.3.14.1 Connectors

Each connector is a DLL (a dynamic link library) and includes a set of attributes that are visible in the user interface of **Pre-Postprocessor**.

Connectors are presented in the project tree by folders (within the **External Connections** folder), which icons and names correspond to types of the connectors:



Child elements in folders of connectors present exchange surfaces.

The following connector types are supported:

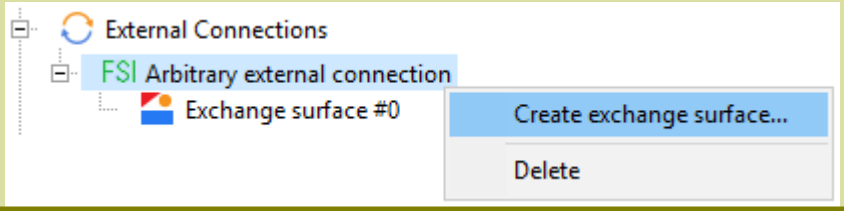
- **Abaqus CSE**
- **Abaqus Direct Coupling**
- **Extended Direct Coupling**
- **NASTRAN**
- **Arbitrary external connection**

Collection of the connector's parameters depends on the type of the connector.

Connectors of some types are specified by their **`inp`** files that store descriptions of exchange surfaces (connection regions), parameters of processes on the surfaces (deformations, heat), and transferred variables. The exchange surfaces are presented in the project tree as child elements of the connectors.

If the **`inp`** file doesn't contain description of exchange surfaces or contains a description, which doesn't correspond to the type of the connector, than an error message will be displayed (for example: "**The selected connector type does not match the one specified in the Abaqus inp-file!**") and no connector will be created.

Context menu of a connector



Menu item	Description
Create exchange surface	Create a new **Exchange surface**. This menu item is presented in context menus of connectors of the **Arbitrary external connection** type only.
Delete	Delete the connector from the **External Connections** folder.

Connectors Abaqus Direct Coupling, Abaqus CSE, and Extended Direct Coupling

Connectors of different types have some general parameters in their **Properties** windows:

Parameter	Description
Activation	This parameter specifies if the connector is active. Possible options are: **Yes | No**.
Source ABAQUS project	Network path and name of the `inp` file, which contains the *Abaqus*' project, that were specified at creation of the connector. This field is read-only and cannot be changed here.
ABAQUS > ...	Parameters of connection to *Abaqus*.
ABAQUS > Run ABAQUS	Run *Abaqus* automatically when **Solver** starts running the calculation. Automatic run of *Abaqus* required that **MPM-Agent** is set and started. Possible options are: **Yes | No**.
ABAQUS > MPM Agent > ...	Parameters of connection to **MPM Agent** (which is the module that launches computations in *Abaqus*). These parameters are available when **ABAQUS > Run ABAQUS = Yes**.
ABAQUS > MPM Agent > Address	Host name or IP address of the computer, on which **MPM-Agent** runs. The default value is **127.0.0.1**.
ABAQUS > MPM Agent > Port	The port number for connection to the computer, on which **MPM-Agent** runs. The default value is formed depending on the version number of *FlowVision*.
ABAQUS > ABAQUS-project	Network path and name of the `inp` file, which contains the *Abaqus*' project. This parameter is available when **ABAQUS > Run ABAQUS = Yes**.
ABAQUS > Simulation controls	This parameter contains user settings of the *Abaqus*' computation, for example, `cpus`, `interactive`, and `mp_mode`. These settings are presented as a space separated string. See descriptions of these settings in the *SIMULIA User Assistance 2019 Abaqus > Execution* document (or similar documents). This parameter is available when **ABAQUS > Run ABAQUS = Yes**.
ABAQUS > IP Source	Method of specifying the IP address of the computer, on which *Abaqus* will run. Possible options are: • **IP of MPM Agent**: the IP address is taken as the value of the **ABAQUS > MPM Agent > Address** parameter (see above) • **User**: the IP address is specified explicitly by the user • **From File**: the IP address is specified in a file, which is specified by the user. This option is useful when the computation runs on multiple nodes.
ABAQUS > Address	Host name or IP address of the computer, on which *Abaqus* will run. The default value is **127.0.0.1**. This parameter is available when **ABAQUS > IP Source = User**.
ABAQUS > File with IP	Name of the file containing the IP address of the computer, on which *Abaqus* will run. This parameter is available when **ABAQUS > IP Source = From File**. The default value is **AbqHost.txt**.
ABAQUS > Port	The port number for connection to the computer, on which *Abaqus* will run. The default value is **5555**.
ABAQUS > Timeout [s]	Timeout for connection to the external program. This timeout is specified in seconds. The default value is **300** [s].
Loads relaxation > ...	Relaxation parameters for loads. These parameters are used to provide a smooth transfer of loads from *FlowVision* to the external program during the co-simulation.

Parameter	Description
	The loads that are transferred to the external program are multiplied by coefficient $k_{sc}R_n$, which depends on the exchange step, where: $R_n = \begin{cases} R_1, & n < N_1 \\ R_1 + \frac{n - N_1}{N_2}(R_2 - R_1), & N_1 \le n \le N_2 \\ R_2, & n > N_2 \end{cases}$ is the relaxation coefficient at the exchange step n n is the number of the exchange step k_{sc} is the scaling factor. It is set by the **Scale factor** parameter (see below). N_1 is the exchange step, since which the relaxation coefficient R_n begins its linear growth from R_1 to R_2. It is set by the **Start in steps** parameter (see below). N_2 is the exchange step, since which the relaxation coefficient R_n stops its linear growth from R_1 to R_2. It is set by the **End in steps** parameter (see below). R_1 is the initial relaxation coefficient; it is set by the **Initial coefficient** parameter (see below) R_2 is the final relaxation coefficient; it is set by the **Final coefficient** parameter (see below) *Example of use:* Suppose that at the beginning of the computation the external program had received high loads from *FlowVision* that can cause divergence of the finite-element computation. If the computation had been able to be convergent under these loads, then *FlowVision* received higher deformations of the simulated body that would cause further increasing of loads and divergence of the computation. Smooth increase of the loads during several steps allows the program to avoid the divergence and to obtain a stable solution.
Loads relaxation > Scale factor	The coefficient can be used for scaling the loads that are transferred from *FlowVision* to the external program. Generally this coefficient is used to recalculate measurement units of the transferred loads. The default value is **1**.
Loads relaxation > Start in steps	The exchange step, starting from which relaxation of loads from *FlowVision* is applied. Often this value is zero, which means applying a growing relaxation coefficient since the the very first exchange step. The default value is **0**.
Loads relaxation > End in steps	The exchange step, since which the relaxation coefficient for loads from *FlowVision* stops its grow. This parameter is set depending on the task's specifics. The default value is **0**.
Loads relaxation > Initial coefficient	The coefficient R_1, on which loads from *FlowVision* are multiplied before beginning of the relaxation. When the relaxation is used, this value is generally set as **0**. The default value is **1**.
Loads relaxation > Final coefficient	The coefficient R_2, on which loads from *FlowVision* are multiplied after finishing of the relaxation. Generally this value is set as **1**. The default value is **1**.
Heat relaxation > ...	Relaxation parameters for thermal loads. These parameters are similar to the **Loads relaxation > ...** parameters (see above).

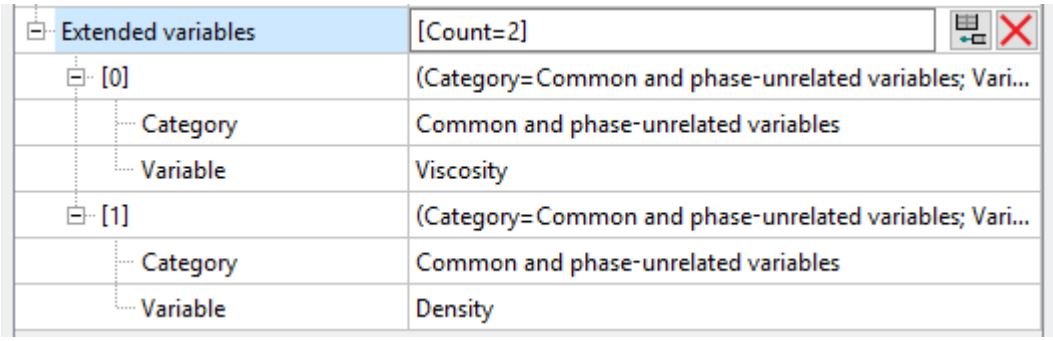
Parameters that are specific to the connector's type are described in separate subsections below.

Parameters of connectors of the "Abaqus Direct Coupling" type

Connectors **Abaqus Direct Coupling** have general connector parameters of types "Abaqus Direct Coupling", "Extended Direct Coupling", and "Abaqus CSE" only and have no specific parameters.

Parameters of connectors of the "Extended Direct Coupling" type

Connectors of the **Extended Direct Coupling** type have general connector parameters of types "Abaqus Direct Coupling", "Extended Direct Coupling", and "Abaqus CSE" and also the **Extended variables** array, which contains variables transferred via the *DirectCoupling* protocol.



Parameters of connectors of the "Abaqus CSE" type and export to the configuration file

Use of connectors of this type, requires starting **Solver** by a special command line with a reference that includes substring "`CSE`" (examples: "`CSE 2017 solver`" and "`CSE 2020 solver`").

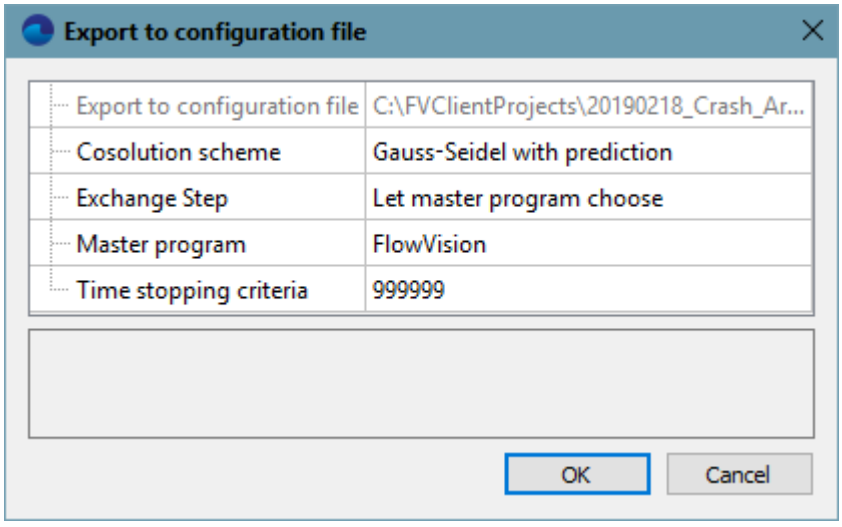
Digits in the command line's name correspond to the version of *Abaqus*.

Command lines with "`CSE 2017`" can be used with all versions of *Abaqus* below *2020*.

Command lines with "`CSE 2020`" can be used with *Abaqus 2020* and newer versions.

An attempt to run FSI simulation with a connector of the **Abaqus CSE** type and regular (not CSE) **Solver** would cause an abnormal termination of the computation.

Connectors **Abaqus Direct Coupling**, in their **Properties** windows, have general connector parameters of types "Abaqus Direct Coupling", "Extended Direct Coupling", and "Abaqus CSE" and also the screen button **Operations > (Save configuration file for CS-service)**, which opens the **Export to configuration file** dialog box:



This dialog box provides a convenient user interface for forming a configuration *Abaqus* `xml` file, so you will not have create this file manually, which requires time and efforts.

The **Export to configuration file** dialog box contains the following fields:

Parameter	Description
Export to configuration file	This is the network path and name of the file, into which the configuration will be saved. This file is saved in the project's client folder, it has the `xml` file name extension and the name, which corresponds to the name of the `inp` file that was used for creation the connector. This field is read-only and cannot be changed.
Cosolution scheme	Possible options are: • **Gauss-Seidel with prediction**: this is an explicit scheme that is similar to the Gauss-Seidel algorithm but the delayed program can predict the results for

Parameter	Description
	the next joint step. This reduces the delay between the programs if the solution is smooth enough for the accurately calculated values. • **Gauss-Seidel**: this is an explicit scheme that assume delayed exchange between the programs. One of the programs leads another program, which calculates with delays between the programs. This scheme of joint computations is well suited for week joint physics and uses small time steps. • **Jacobi**: this is an explicit scheme of parallel simulation, which is done by both programs with exchanges in specified points. This scheme is less resource consuming but also less stable. It is recommended for simulating tasks with week joint physics. • **Implicit iterative**: this is an implicit iterative co-simulation scheme that can be used for strong joint physics. The solution is iterated during the joint computation to prevent any delays between the programs.
Exchange Step	Possible options are: • **Let master program choose**: here you select a program, which will set the exchange step • **Minimum step**: the exchange step is selected as the minimal step of the programs that are used in the computation • **Maximum step**: the exchange step is selected as the maximal step of the programs that are used in the computation
Master program	This is the program, which sets the exchange step. Possible options are: **FlowVision | ABAQUS**. This field is available when Exchange Step = **Let master program choose**.
Time stopping criteria	The simulated time, at which the computation terminates, [s]

The export into the configuration file is carried out when you click **OK**.

Connector NASTRAN

Parameters of connectors of the "NASTRAN" type

Parameter	Description
Activation	This parameter specifies if the connector is active. Possible options are: **Yes | No**.
NASTRAN > ...	Parameters of connection to *Nastran*.
NASTRAN > Run remote program	This parameter specifies if the external program is to be started using **MPM-Agent** or it is assumed that it has already been started. Possible options are: **Yes | No**.
NASTRAN > MPM Agent > ...	Parameters of connection to **MPM Agent** (which is the module that launches computations in *Nastran*). These parameters are available when **NASTRAN > Run remote program = Yes**.
NASTRAN > MPM Agent > Address	Host name or IP address of the computer, on which **MPM-Agent** runs. The default value is **127.0.0.1**.
NASTRAN > MPM Agent > Port	The port number for connection to the computer, on which **MPM-Agent** runs. The default value is formed depending on the version number of *FlowVision*.
NASTRAN > Command Line	Additional parameters that are to be specified in the command line for starting the external program in **MPM-Agent**
NASTRAN > IP Source	Method of specifying the IP address of the computer, on which *Nastran* will run. Possible options are: • **IP of MPM Agent**: the IP address is taken as the value of the **NASTRAN > MPM Agent > Address** parameter (see above) • **User**: the IP address is specified explicitly by the user • **From File**: the IP address is specified in a file, which is specified by the user. This option is useful when the computation runs on multiple nodes.

Parameter	Description
NASTRAN > Address	Host name or IP address of the computer, on which *Nastran* will run. The default value is **127.0.0.1**. This parameter is available when **NASTRAN > IP Source = User**.
NASTRAN > File with IP	Name of the file containing the IP address of the computer, on which *Nastran* will run. This parameter is available when **NASTRAN > IP Source = From File**. The default value is **Host.txt**.
NASTRAN > Port	The port number for connection to the computer, on which *Nastran* will run. The default value is **5555**.
NASTRAN > Timeout [s]	Timeout for connection to the external program. This timeout is specified in seconds. The default value is **300** [s].
Loads relaxation > ...	These parameters are similar to parameters with same names in general parameters of connectors of types "Abaqus Direct Coupling", "Extended Direct Coupling", and "Abaqus CSE".

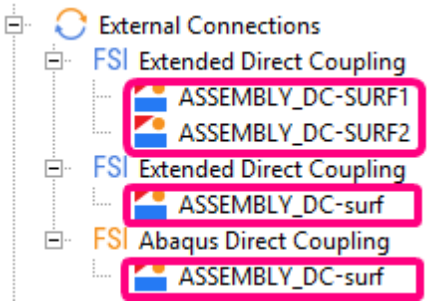
Connector "Arbitrary external connection"

Parameters of connectors of the "Arbitrary external connection" type

Parameter	Description
Activation	This parameter specifies if the connector is active. Possible options are: **Yes** | **No**.
Connector name	Name of the dynamic library involved in external connection
Remote software properties > ...	Parameters of connection to the external software, which is used for co-simulation. These parameters are similar to parameters **ABAQUS > ...** in general parameters of connectors of types "Abaqus Direct Coupling", "Extended Direct Coupling", and "Abaqus CSE".
Loads relaxation > ... **Heat relaxation > ...**	These parameters are similar to parameters with same names in general parameters of connectors of types "Abaqus Direct Coupling", "Extended Direct Coupling", and "Abaqus CSE".

8.1.8.3.14.2 Exchange surfaces

Exchange surfaces are presented in the project tree as child elements of connectors:



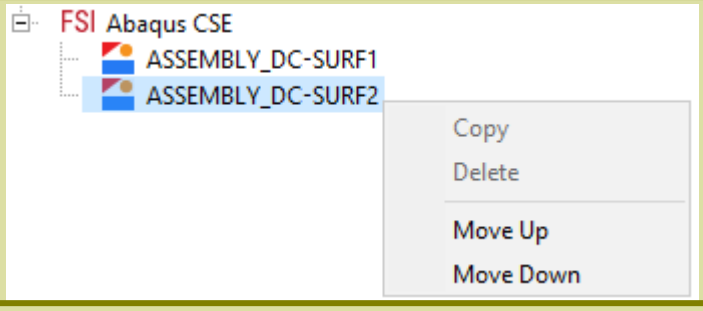
Parameters of exchange surfaces

Parameter	Description
Name	Name of the exchange surface. This name is taken from the `inp` file of the connector (when co-simulation with *Abaqus* is used, this name is the name of the appropriate region in *Abaqus*).
Moving body	Moving body, which will be used for connection. When mapping is enabled, the you should specify the **Moving body** that corresponds to the CFD surface, on which the mapping is to be done.

Parameter	Description
Mapping > ...	Settings of mapping (matching corresponding geometry surfaces during interaction between *FlowVision* and the external program)
Mapping > Enabled	This parameter defines if the mapping is enabled. Possible options are: **Yes** | **No**. *When the mapping is disabled:* The data exchange is carried out using a mesh surface that is received from a finite element software (FE code). In the *FlowVision*'s simulation geometry of the **Moving body** will be replaced. *When the mapping is enabled:* Surfaces in the FE code and in *FlowVision* might differ. During the data exchange the data will be interpolated from one surface to another. The interpolation is done using the *FlowVision*'s embedded surface matching algorithm. In the *FlowVision*'s simulation the **Moving body**'s surface will be used itself. --- See details in the section Mapping.
Mapping > Surface	This parameter specifies what surface will be used for building the mapping surface (mapper), either initial or deformed surface of the finite element object. Possible options are: **Initial** | **Deformed**. If no preliminary computation is carried out in the FE software, the initial surface should be used, and if a preliminary computation is carried out, then the deformed surface should be used.
Mapping > Rebuild mapper	This parameter specifies if the program will rebuild the saved mapping surface (mapper) or will remove the saved mapping surface and build a new mapping surface. A mapper is built if **Mapping > Enabled = Yes** but there is no mapping file (the `*.fvibin` file, which locates in the server part of the project). A mapper is rebuilt if the geometry, for which it has been created, doesn't conform to the current geometry. A mapper is built at the beginning of the analysis only. If running of the computation is resumed after a stop (caused by loss of connection to **Solver**, etc.), the mapper might be rebuilt. Possible options are: **Yes** | **No**. If a project has history and the user wishes to start the computation from the zero step, it is necessary either update the geometry to the initial at the current step or remove all saves except the first one, which contains the initial geometry. Otherwise geometry from the current step will not match the mapper and the mapper will be rebuilt automatically.
Mapping > FE-object	The **Imported object**, which has been received from the FE software and which corresponds to the surface used for data exchange transferring the geometry, deformations and temperature. This field is read-only and cannot be changed here.
Body correction > ...	Due to the nature of processing of **Moving bodies** in the external program, mutual overlaps of **Moving bodies** or unwanted gaps between **Moving bodies** could occur. Specifying **Body correction** settings can sometimes overcome these issues. The **Body correction** parameters specify the additional correction of the **Moving body**, which corresponds to the exchange surface. Expanding or shrinking of the **Moving body** occurs during this correction. Location of the exchange surfaces (and corresponding **Moving bodies**) in the list of the connector's child elements in the project tree has influence on the process of the correction. The **Moving bodies** are processed according their position in the list in the direction from the top downward.

Parameter	Description
	See details and illustrations in the section Correction of moving bodies. *Note:* Correction of **Moving bodies** is done once even if the computation runs in the absence of connection to *Abaqus*.
Body correction > Enabled	This parameter defines if the **Body correction** is enabled. Possible options are: **Yes** | **No**.
Body correction > Amount	An approximate value of the offset of the **Moving body**'s surface, [m]
Body correction > By proximity	This parameter defines the method of how the correction of the **Moving body** will be done: to all directions evenly (not depending on other **Moving bodies**) or towards **Moving bodies** that are presented on lower positions in the list or aren't included into the list at all (see illustrations in the section Correction of moving bodies). Possible options are: **Yes** | **No**.
Self-intersections removal > ...	Settings of automatic removal of self-intersections using the offset method
Self-intersections removal > Enabled	This parameter defines if the automatic self-intersections removal is enabled. Possible options are: **Yes** | **No**.
Self-intersections removal > Initial factor	The multiplier for the minimal edge's length of the facets participating in the self-intersection. The product of this multiplication is the initial offset value.
Self-intersections removal > Max number of tries	This is the maximal number of tries of increasing the offset value. Its value is an integer number in the range [1...19].
Node loadings interpolation	When mapping is disabled (**Mapping > Enabled = No**), the data exchange uses a mesh surface, which is received from the finite element (FE) software, and surfaces in the FE software and in *FlowVision* can be different. During the exchange, the data will be interpolated from one one surface to another surface. The interpolation employs an internal *FlowVision*'s algorithm of matching the surfaces. There are the following methods of interpolation the loadings from faces to nodes: • **Angle based** - the loads are calculated proportionally to angles at nodes. • **Voronoi diag.** - the loads are calculated proportionally to areas in the Voronoi diagram; this allows more precise calculating of the loading transfer, but this algorithm requires more computational resources. See details in the section Node loadings interpolation.
Send data	This parameter defines if sending data from *FlowVision* to the external program will be done. Possible options are: **Yes** | **No**.
Receive data	This parameter defines if receiving data from the external program to *FlowVision* will be done. Possible options are: **Yes** | **No**.
Send data *(array)*	Array of codes of variables that are transferred from *FlowVision* to the external program. This array is read from the `inp` file, which has been used for creating the connector. This fields are read-only and cannot be changed here. See the codes of variables in the section Variables for import and export.
Receive data *(array)*	Array of codes of variables that are transferred from the external program to *FlowVision*. This array is read from the `inp` file, which has been used for creating the connector. This fields are read-only and cannot be changed here. See the codes of variables in the section Variables for import and export.

Context menu of an exchange surface



Menu item	Description
Copy	This menu item is not available.
Delete	Delete the selected exchange surface from the connector. This menu item is available for connectors of the **Arbitrary external connection** type only.
Move Up **Move Down**	These menu items are used to change position of the selected exchange surface (and its corresponding **Moving body**) in the list of the connector's child elements. This has an effect on sequence of corrections of **Moving bodies** (see section Correction of moving bodies).

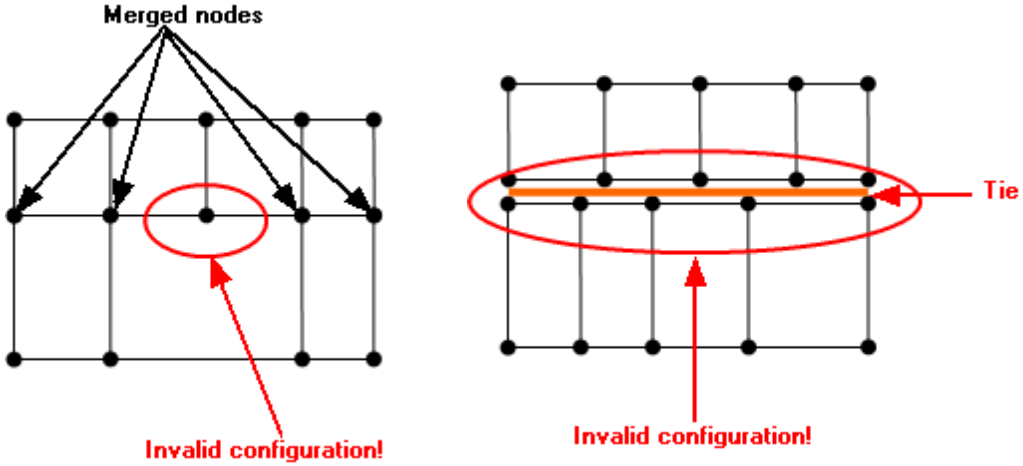
See also:

- Requirements to exchange surfaces
- Mapping
- Correction of moving bodies
- Variables for import and export
- Node loadings interpolation

Requirements to exchange surfaces

Requirements to the geometry model in an external finite-element analysis software when mapping is not applied

1. The model's details, which take part in the coupling analysis, are to be simulated by volume or flat elements.
2. Interface surface(s) (in *Abaqus* they are interface regions of the **SURFACE** type) for the coupling analysis (the direct coupling interface) must be defined in the model.
3. An interface surface must be:
   - closed
   - determined on a continuous mesh. The mesh must have no "duplicate" nodes, modified elements and must not contain a surface of two meshes connected using a TIE-contact (see the illustration)

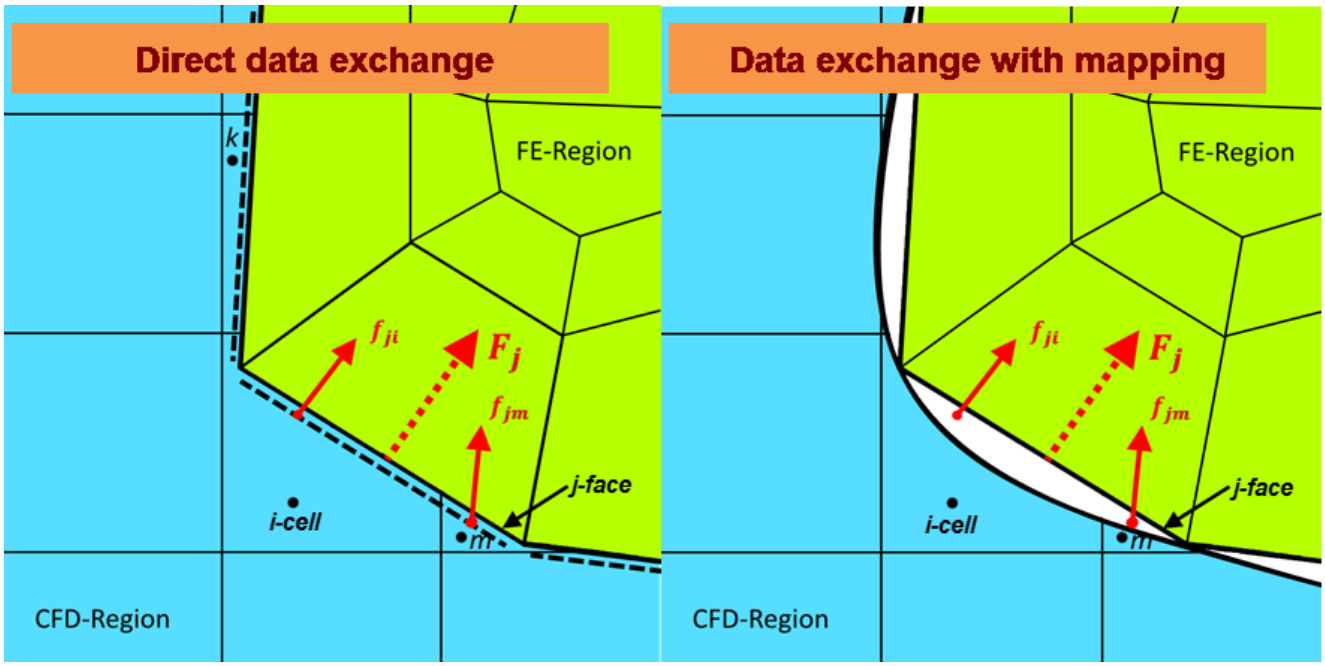


Mapping

When the data exchange is done directly from a mesh to mesh, *FlowVision* demands some requirements to quality of the mesh of the external finite-element analysis software (FE-mesh), see subsection *"Requirements to the geometry model in an external finite-element analysis software when mapping is not applied"*.

When mapping is applied, *FlowVision* uses its own mesh and interpolates the data to the FE-mesh. This approach allows the user to have less strict requirements to the FE-mesh and to manage the following issues:

- When the FE-mesh is courser then the *FlowVision*'s mesh: the data interpolation will be applied between nodes of both meshes. *Example:* simulation of a wing's aeroelasticity.
- When the FE-mesh is a surface with non-closed elements and the *FlowVision*'s mesh is closed geometry: non-closed elements will be interpolated depending on boundary nodes and locations of nodes on the mapping surface. *Examples:* a sail, a balloon.
- When the FE-mesh has defects (self-intersections, overlays, etc.) and the *FlowVision*'s mesh has no such defects: it is possible to find a mapping surface that has no these defects. *Example:* a heart where the common surface from the whole assembly is used (elements of assembly contact each other, so self-intersections appear).



Use of the mapping surface

In the project tree **Imported objects** will appear, which correspond to surfaces in the FE software used for the data exchange. The appropriate **Imported object** is displayed in properties of the exchange surface in the parameter **Mapping > FE-object**.

The external FE software can carry out a preliminary computation, during which the initial geometry might be substantially deformed. The **Mapping > Surface** parameter in properties of the exchange surface allows you to select the geometry, which *FlowVision* will receive on the zero exchange step: either the initial geometry (that was before the preliminary computation in the FE software) or the deformed geometry (after the preliminary computation).

Using the **Mapping > Rebuild mapper** parameter in properties of the exchange surface, you can specify if the program would rebuild the saved mapping surface or it would remove it and create a new mapping surface.

Details on the mapping procedure

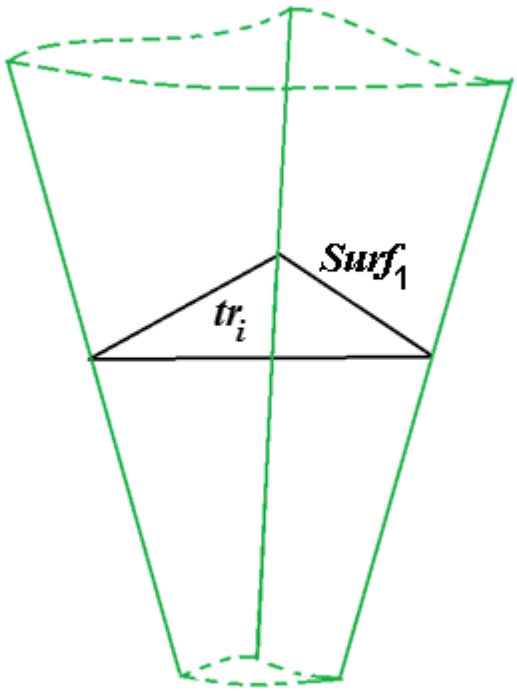
Below the mapping procedure is outlined, which is used in simulations with **External Connections** when mapping is applied.

Mapping is matching of two surfaces, $Surf_1$ and $Surf_2$. Each of these surfaces is presented by a set of triangles:

$$Surf = \bigcup_i tr_i$$

Each of triangles of the source surface $Surf_1$ is mapped on the other (target) surface $Surf_2$. Let's take a closer look at this process.

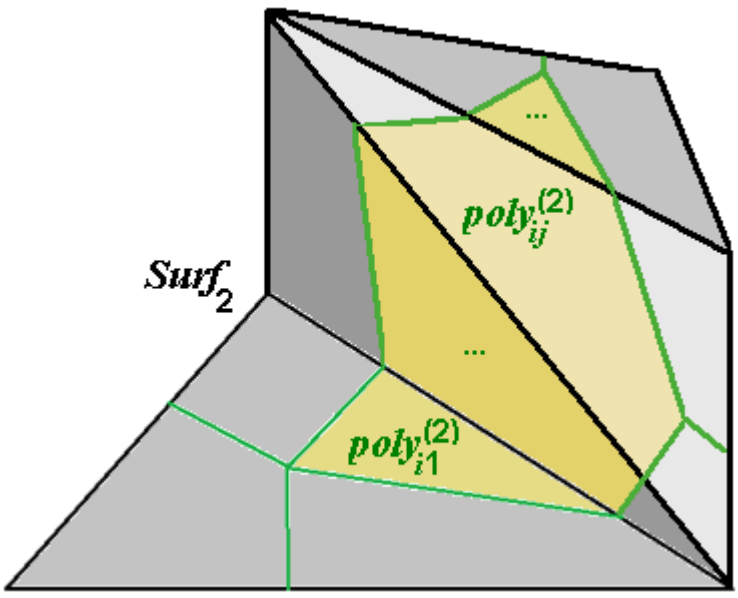
For each side of a such triangle let's define its mean normal as half-sum of normals of this and adjacent triangles. Let's draw, through each side of the triangle, a plane, which is parallel to the mean normal of this side. Set of three such planes determines the *solid angle* of this triangle:



Solid angle of a triangle of the surface $Surf_1$

Solid angles of one surface will split triangles of the other surface into fragments.

Solid angles of the first surface $Surf_1$ will split triangles of the second surface $Surf_2$ to fragments:



Splitting a triangle from the surface $Surf_2$ by solid angles of triangles of the surface $Surf_1$

Each of these fragments is a plane convex polygon. So each triangle of the surface $Surf_1$ can be matched to a set of polygons of the surface $Surf_2$.

Let us denote:

$\boldsymbol{r}_i^{(1)}$ – center of the source triangle $tr_i^{(1)}$

$\boldsymbol{n}_i$ – normal of the plane of the source triangle $tr_i^{(1)}$

$\boldsymbol{r}_{ij}^{(2)}$ – centers of projections (polygons) $poly_{ij}^{(2)}$

Among all polygons that fall within the solid angle of the source triangle we will take into account only those, for which the condition of *proximity of the projection* to the source triangle is fulfilled:

$$|(\boldsymbol{r}_i-\boldsymbol{r}_{ij})\boldsymbol{n}_i| < \varepsilon$$

The value ε is not set by the user, it is determined by the program automatically.

Let us denote:

$s_i^{(1)}$ – area of the i-th triangle $tr_i^{(1)}$

$s_{ij}^{(2)}$ – area of projection $poly_{ij}^{(2)}$

$S_i^{(2)}$ – total area of the surface, on which the triangle $tr_i^{(1)}$ has mapped, i.e.:

$$S_i^{(2)} = \sum_j s_{ij}^{(2)}$$

Let's take a closer look at the value $f_i^{(1)}$, which is ratio of values $S_i^{(2)}$ and $s_i^{(1)}$:

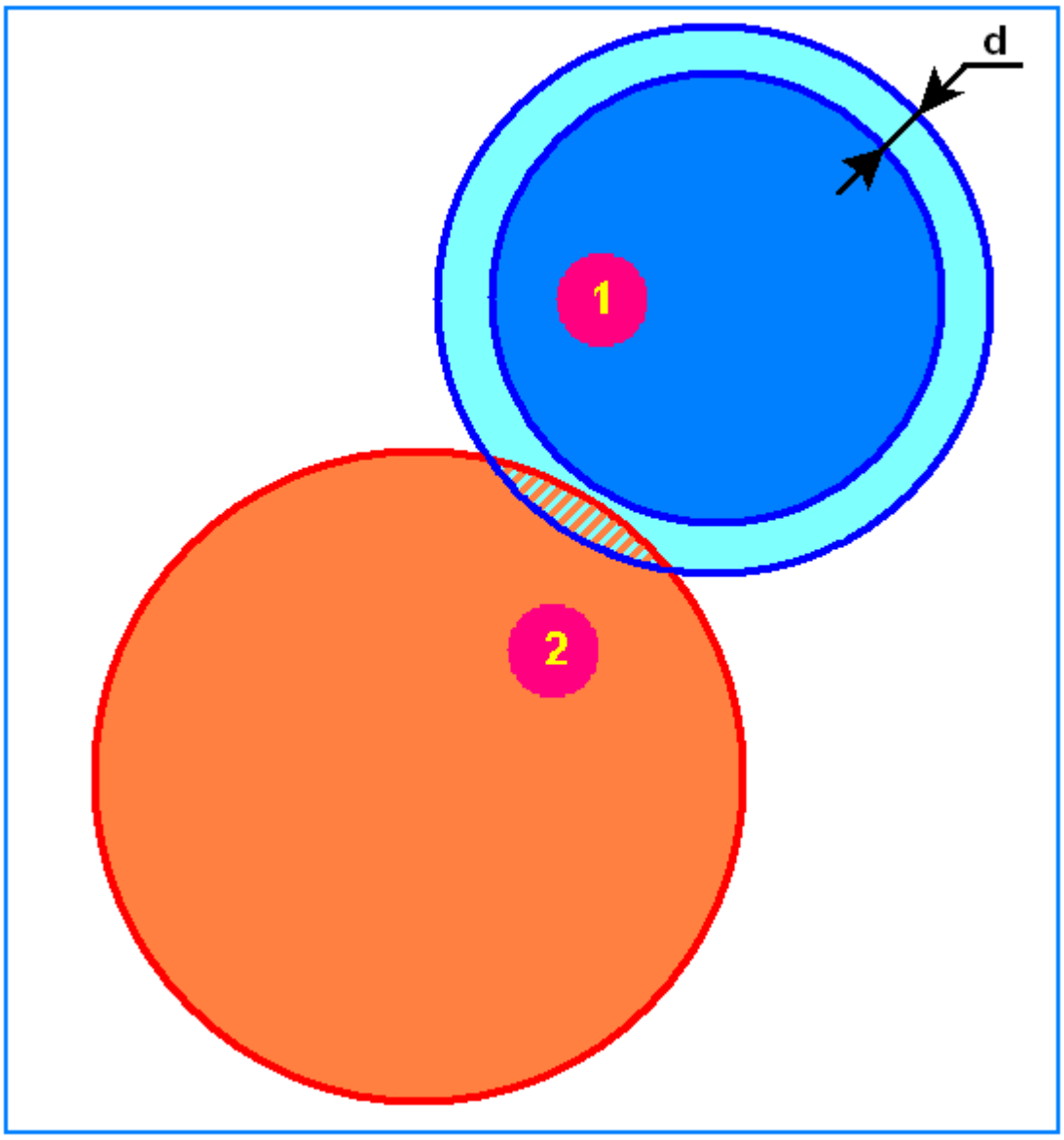
$$f_i^{(1)} = \frac{S_i^{(2)}}{s_i^{(1)}}$$

The value $f_i^{(1)}$ is used for coloring triangles of the source surface in the visualization layer **Mapping surface**.

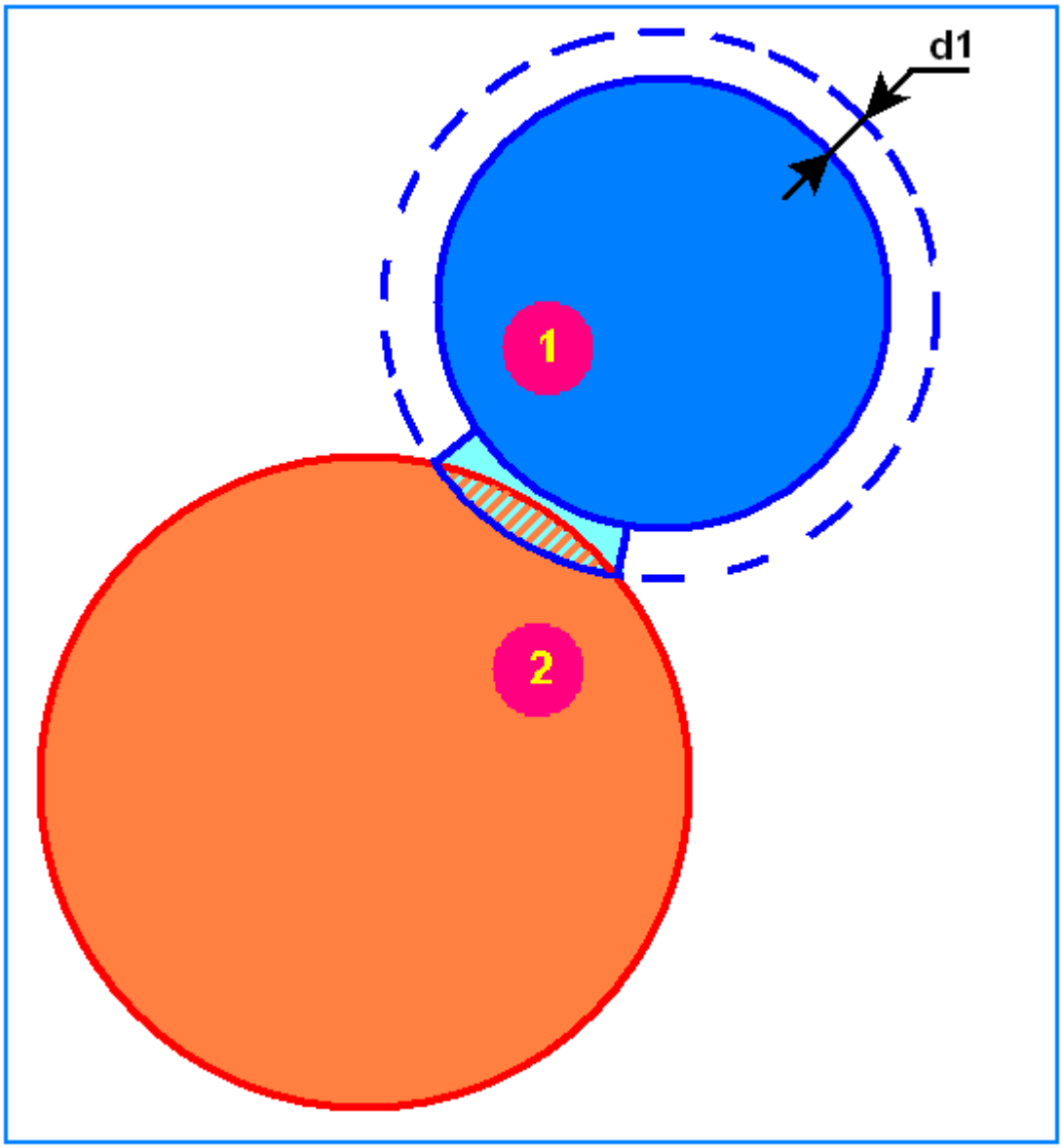
In cases when $f_i^{(1)}$ = 0, the appropriate triangles in the **Mapping surface** layer will not be displayed (such triangles will be invisible).

Correction of moving bodies

Operation of the algorithm for correction **Moving bodies** and their corresponding exchange surfaces depends on positions of the exchange surface among child elements of their connecter in the project tree. These positions can be changed using the **Move Up** and **Move Down** commands from the context menus of the exchange surfaces.

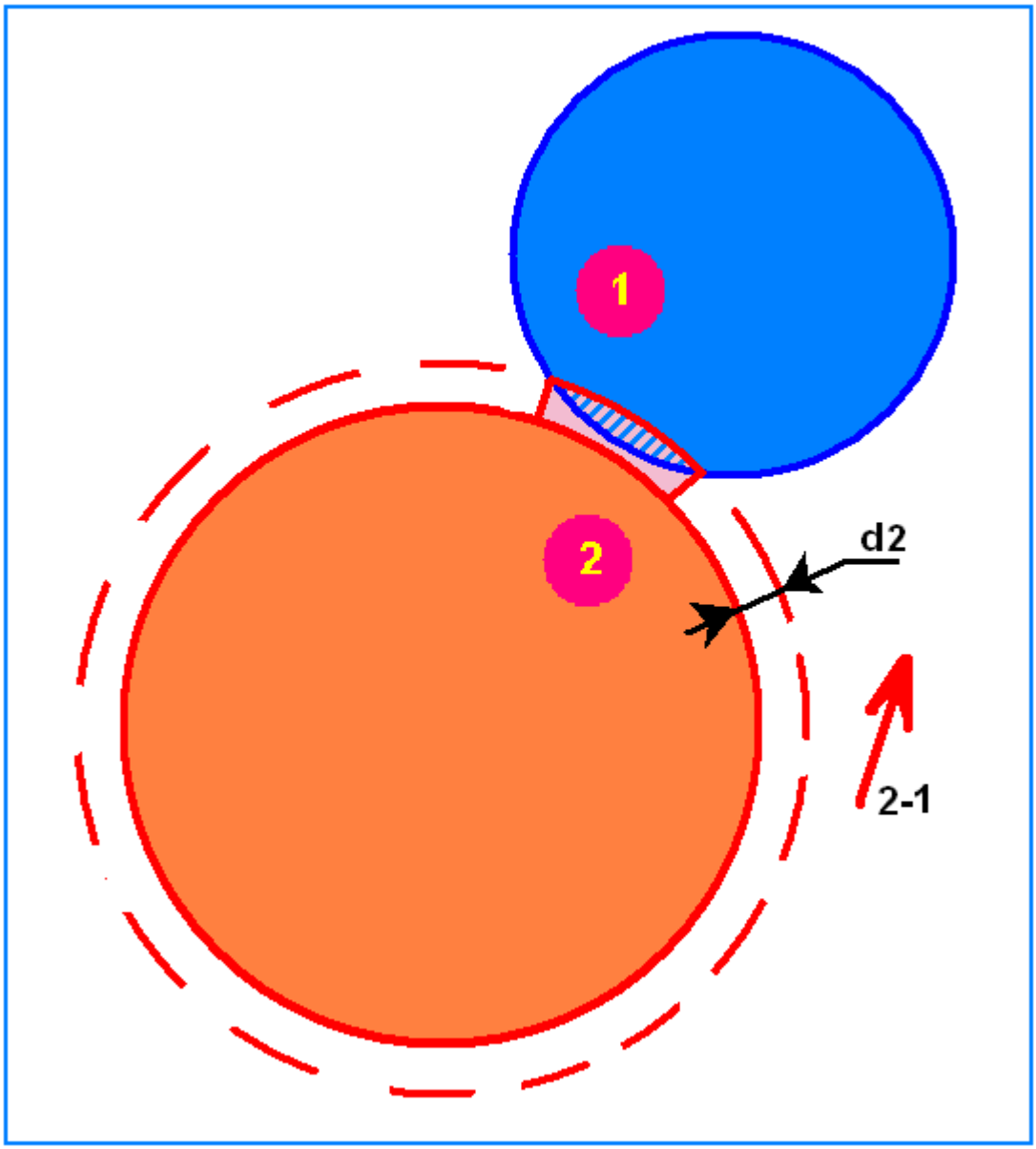


Correction of **Moving body** "1" when its **By proximity = No**.
d is the value of the **Amount** parameter of **Moving body** "1".
Moving body "2" does not affect the correction.



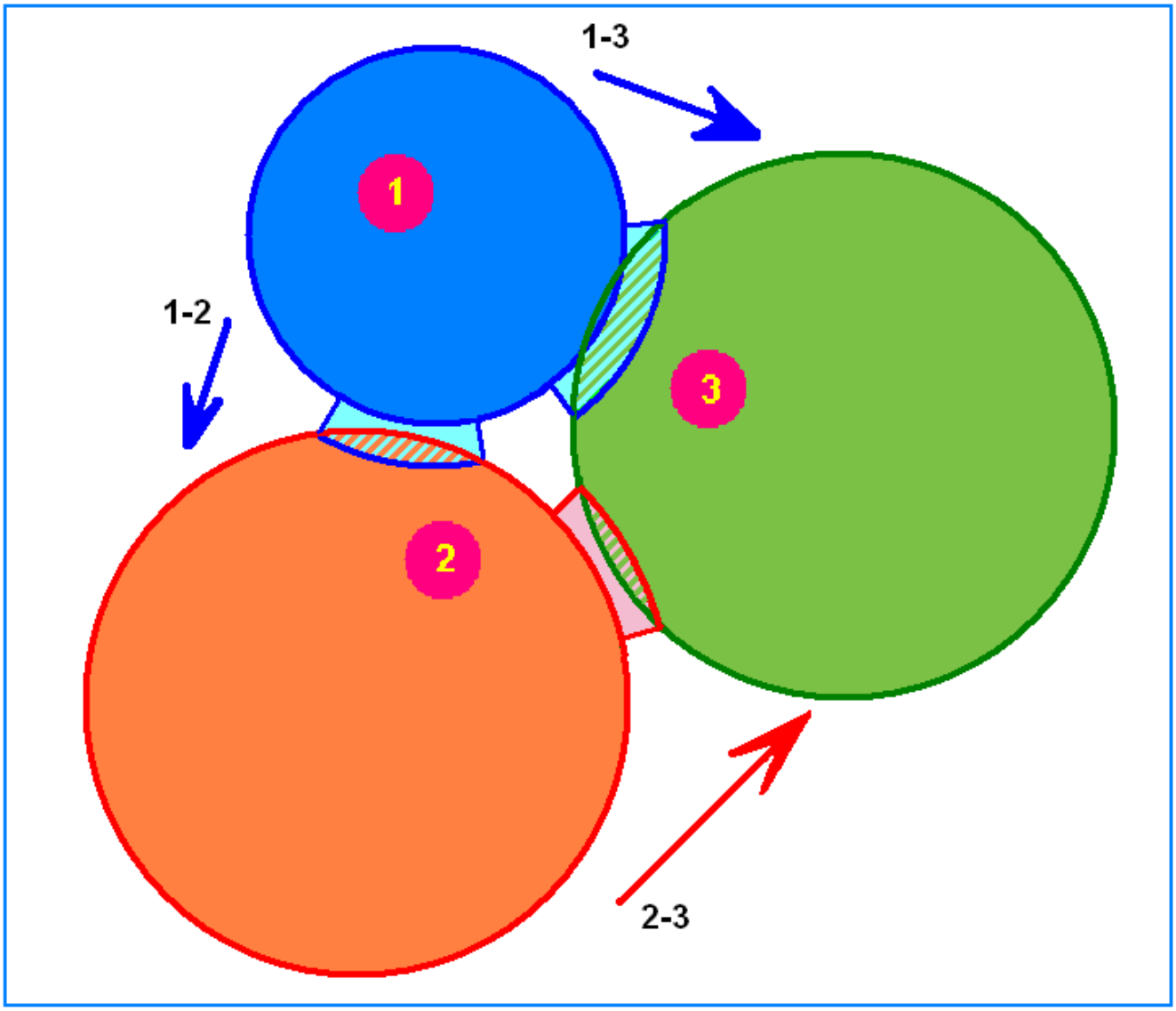
Correction of **Moving body** "1" when its **By proximity = Yes** and **Moving body** "2" locates in the list lower then **Moving body** "1" or isn't included into the list.

d1 is the value of the **Amount** parameter of **Moving body** "1".



Correction of **Moving body** "2" when its **By proximity = Yes** and **Moving body** "1" locates in the list lower then **Moving body** "2" or isn't included into the list.

d2 is the value of the **Amount** parameter of **Moving body** "2".



Correction of **Moving bodies** "1" and "2" when **Moving body** "2" locates in the list lower then **Moving body** "1" and **Moving body** "3" locates in the list lower then **Moving bodies** "1" and "2" or isn't included into the list. Both **Moving bodies** "1" and "2" have **By proximity = Yes** specified.

Variables for import and export

The table below lists codes of variables in `inp` files, used to form connectors, and in configuration *Abaqus* `xml` files.

Codes of variables in `inp` files	Codes of variables in `xml` files	Physical variable, dimension*)	Is supported
UT or **U**	**displacement**	Displacement	Yes
VT or **V**	**velocity**	Velocity	No
AT or **A**	**acceleration**	Acceleration	No
UR	**rotation**	Rotational displacement	No
VR	**rotational_velocity**	Rotational velocity	No

Codes of variables in inp files	Codes of variables in xml files	Physical variable, dimension*)	Is supported
AR	**rotational_acceleration**	Rotational acceleration	**No**
COORD	**coordinates**	Coordinates	**Yes**
CF	**force**	Force	**Yes**
CM	**moment**	Torque	**No**
PRESS	**pressure**	Pressure	**Yes**
NT	**Temperature**	Temperature	**Yes**
CFL	**heat_flux**	Heat flux	**Yes**
HFL	**flux**	Flux	**No**

*) The dimensions are defined by settings of the external program:

- L is the unit of length
- T is the unit of time
- F is the unit of force
- θ is the unit of temperature
- J is the unit of energy

Node loadings interpolation

To calculate values in nodes, the following steps are done:

1. Integral values on 2D elements (triangles) are calculated by integrating the values from computational cells that are adjacent to the surface.
2. Each node receives a linear combination (with weight coefficients) of values from surface elements that are adjacent to this node:

$$VALUE_{nodal} = \sum_i VALUE_{element,i} \cdot k_i$$

The influence area $AREA_{nodal}$ is a some «area of the node» that can be used for binding the force and the pressure:

$$PRESSURE_{nodal} = \frac{FORCE_{nodal}}{AREA_{nodal}}$$

Interpolation of the force is done either proportionally to angles of adjacent facets or using the Voronoi diagram. The applied method is specified by the **Node loadings interpolation** parameter in properties of the exchange surface.

Interpolation of the force proportionally to the angle

The area is calculated similarly as sum of areas of adjacent elements (with weight coefficients):

$$AREA_{nodal} = \sum_i AREA_{element,i} \cdot k_i$$

The same weight coefficients are used to calculate the force:

$$FORCE_{nodal} = \sum_i FORCE_{element,i} \cdot k_i$$

The coefficients k_i are selected in such a way that sum of influence areas of all nodes would be the equal to the sum of areas of all elements:

$$\sum AREA_{nodal} = \sum AREA_{element}$$

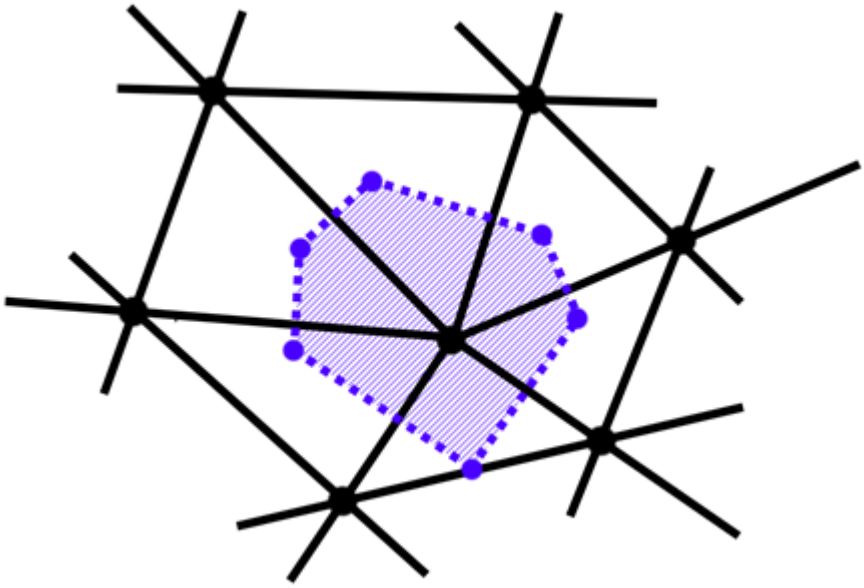
The current version of the program calculate coefficients k_i proportionally to angles α_i of elements adjacent to the node:

$$k_i = \frac{\alpha_i}{\pi}$$

The angle α_i is the angle of adjacent facet at the node, for which the values $PRESSURE_{nodal}$, $FORCE_{nodal}$, and $AREA_{nodal}$ are calculated.

Interpolation of the force using the Voronoi diagram

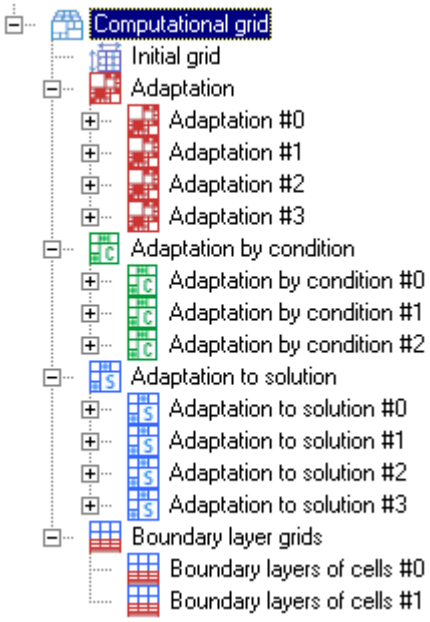
For the triangles the program constructs a set of points, where each point provides a minimal perpendicular with other points. Right lines between the points form the area $AREA_{nodal}$.



The force on the node is calculated by the following formula:

$$FORCE_{nodal} = \sum_i FORCE_{element,i} \cdot \frac{AREA_{nodal,i}}{AREA_{element,i}}$$

8.1.8.3.15 Folder «Computational grid»



Folder **Computational grid** in the project tree

The **Computational grid** folder contains the following element and subfolders:

- element **Initial grid**
- subfolder **Adaptation**
- subfolder **Adaptation by condition**
- subfolder **Adaptation to solution**
- subfolder **Boundary layer grids**

For the sake of brevity, to avoid repetitive descriptions, we describe below some common settings and user interface elements.

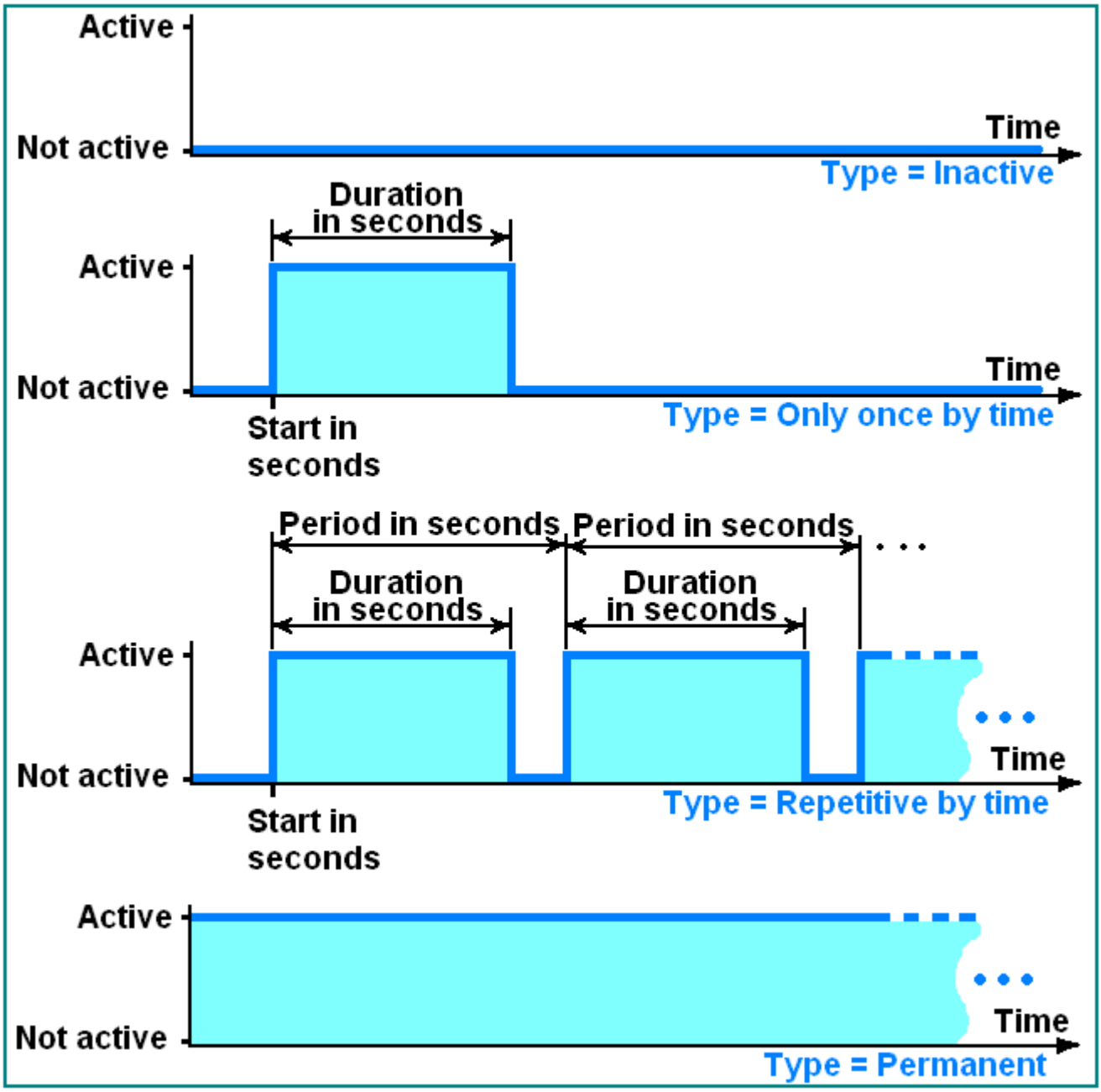
Activity of adaptations and boundary layer grids

All the *subfolders* in the **Computational grid** folder (the simple **Adaptation**, **Adaptation by condition**, **Adaptation to solution**, **Boundary layer grids**) in their **Properties** windows contain groups of parameters **Activation**, which define timing of turning on/off the appropriate adaptations and boundary layer grids. This parameters are also similar to the appropriate parameters of **Modifiers**.

Parameters	Descriptions
Activation > ...	This settings specify the activity (time and frequencies of the action), applied to all child elements, for which **Enabled**=**Yes** is set in their **Properties** windows. Adaptation to solution can mandatory enable **Adaptation by condition** and the simple **Adaptation**; and **Adaptation by condition** can mandatory enable the simple **Adaptation**. This is done according the following rules: • **Adaptations #N**, which have **Enabled=Yes** in their properties, will start not only according their **Activation** settings, but also each time when any **Adaptation by condition #N** or **Adaptation to solution #N** starts. • **Adaptations by condition #N**, which have **Enabled=Yes** in their properties, will start not only according their **Activation** settings, but also each time when any **Adaptation to solution #N** starts. Such mandatory start is done even if the started adaptation has **Activation >Type = Inactive**. You can turn off this mandatory start, only if you set **Enabled=Yes** in the properties of the adaptation (when you do so, the appropriate icon in the project tree is displayed in faint colors).
Activation > Type	Type of timing of running the adaptations or boundary layer grids:

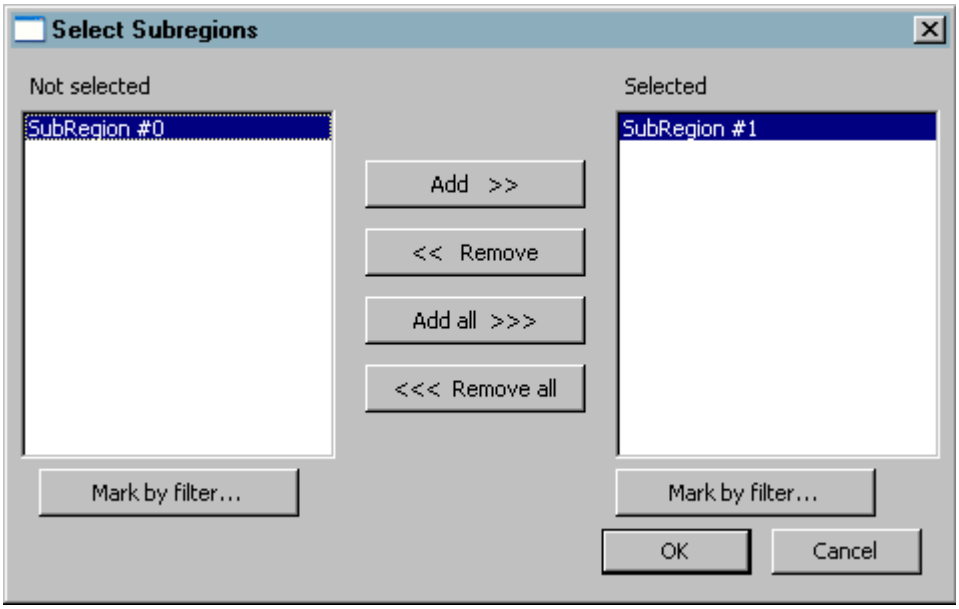
Parameters	Descriptions
	• **Inactive** - the adaptations (boundary layer grids) are not scheduled (at this selection the icon of the appropriate subfolder is faded:  ,  ,  or ) • **Only once by time** - the adaptations (boundary layer grids) are active once from the moment of **Start in seconds** during the **Duration in seconds** period, then they turn off. • **Only once by step** - the adaptations (boundary layer grids) are active once from the moment of **Start in steps** during the **Duration in steps** period, then they turn off. • **Permanent** - the adaptations (boundary layer grids) are always active. • **Repetitive by time**[5] - the adaptations start periodically from time moments **Start in seconds** + *N** **Period in seconds** (where *N* = 0, 1, ...) and then are active during **Duration in seconds**, after which they suspends until a new cycle begins. • **Repetitive by step**[5] - the adaptations start periodically from time steps **Start in steps** + *N* * **Period in steps** (where *N* = 0, 1, ...) and act several steps, the number of which is specified by the **Duration in steps** parameter, then the adaptations become inactive until a new cycle begins.
Activation > Start in seconds[*]	The start time for activity of the adaptations or boundary layer grids.[*]
Activation > Duration in seconds[*] [6]	The duration of activity of the adaptations (single or within each period), specified in seconds.
Activation > Period in seconds[**] [6]	Period of the cycle of enabling/disabling the adaptations, specified in seconds.
Activation > Duration in steps[***] [6]	The number of steps of activity of the adaptations (single or within each period).
Activation > Period in steps[4]	Period of the cycle of enabling/disabling the adaptations, measured in number of steps.

[*] This parameter is available if **Type = Only once by time** or **Repetitive by time**.
[**] This parameter is available if **Type = Repetitive by time**.
[***] This parameter is available if **Type = Only once by step** or **Repetitive by step**.
[4] This parameter is available if **Type = Repetitive by step**.
[5] These values of the parameter **Type** are not available for **Boundary layer grids**.
[6] These parameters are absent for **Boundary layer grids**.



Activity depending on the parameter **Type**
(diagrams for **Type = Only once by step** and **Type = Repetitive by step**
are similar to diagrams for **Type = Only once by time** and **Type = Repetitive by time**)

The dialog box "Select Subregions"

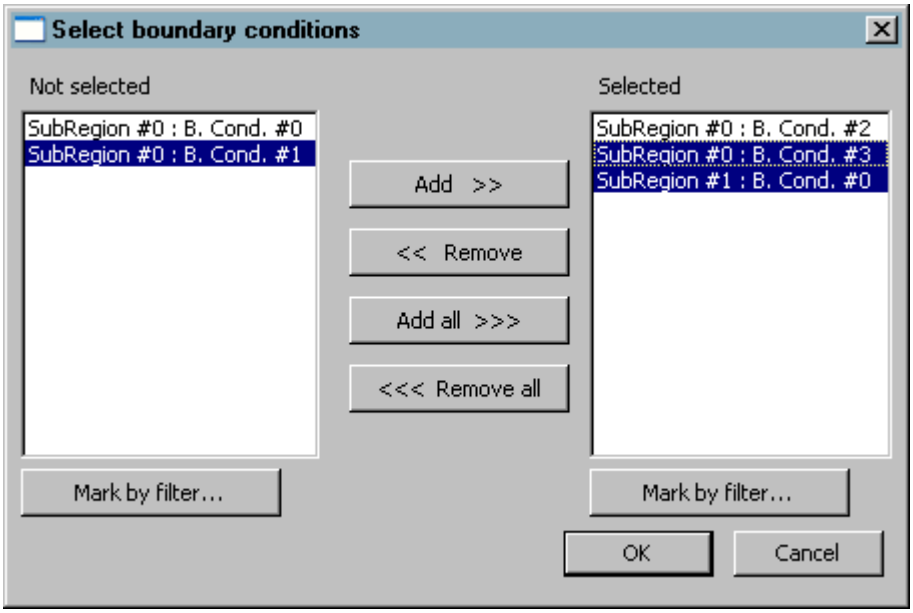


The **Select Subregions** dialog box is used to specify **Subregions**, in which an adaptation can act. Only those **Subregions** are presented in this dialog box, in which some **Model #N** is specified.

Select the desired **Subregions** from the **Not selected** pane and move them to the **Selected** pane by clicking the **Add** button. To remove a **Subregion** from the list of selected ones, select it from the**Selected** pane and then click on the **Remove** button. Clicking by the mouse, you can select several **Subregions**, and you also use the **Mark by filter** button to select the desired **Subregions** according to how they meet the specified conditions (see the subsection *"Filtering the list items (use of the "Mark by filter" button)"* below). Also you can use buttons **Add all** and **Remove all**.

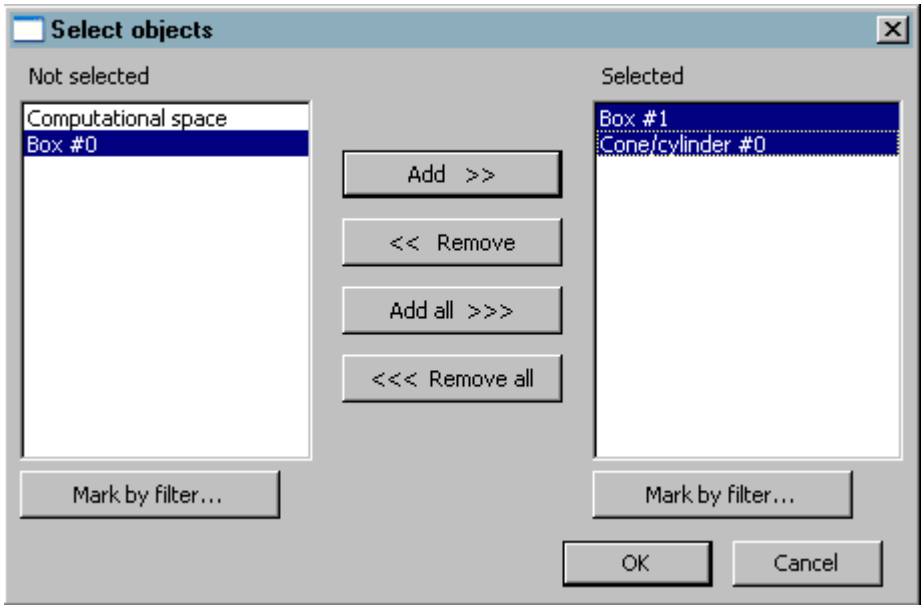
When you finish forming the list of the selected **Subregions**, click on the **OK** button. To reject your changes and close the dialog box, click **Cancel**.

The dialog box "Select boundary conditions"



The **Select boundary conditions** dialog box is used to specify **Boundary conditions**, on which an adaptation or a boundary layer grid can be set. The interface controls in this dialog box are the same as those, which are used in the **Select Subregions** dialog box (see above).

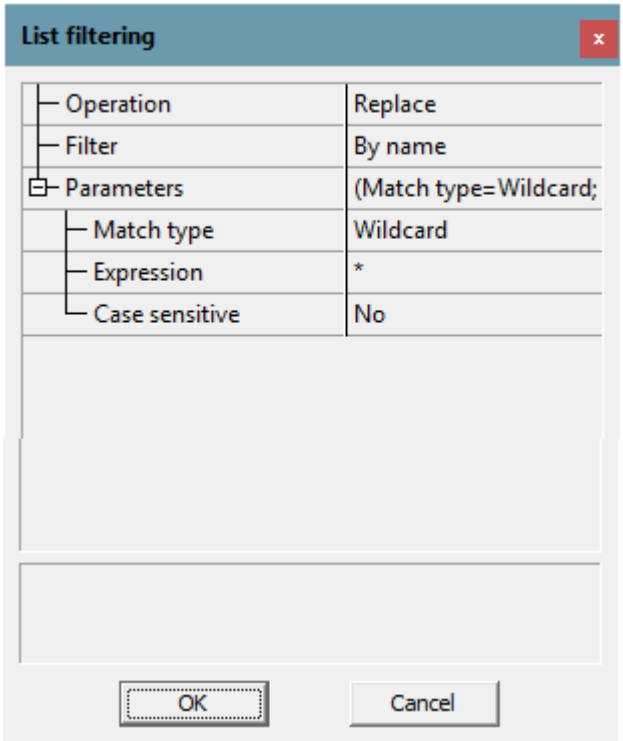
The dialog box "Select objects"



The **Select objects** dialog box is used to specify **Objects**, where (for example, in their volumes or on all or selected their surfaces) you can set an adaptation. The interface controls in this dialog box are also the same as those, which are used in the **Select Subregions** dialog box (see above).

Filtering the list items (use of the "Mark by filter" button)

When a dialog box **Select Subregions**, **Select boundary conditions**, or **Select objects** (see above) contains many items in the **Not selected** or **Selected** panes, it might be convenient to use the **Mark by filter** buttons. When you click them, the **List filtering** dialog box opens:



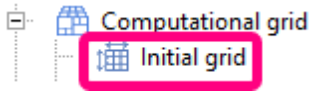
The **List filtering** dialog box contains the following parameters:

Parameter	Description
Operation	The action, which will be done with the found list items. The possible options are: • **Replace** - replace the existing list items by the found ones • **Add** - add the found list items to the existing ones • **Subtract** - subtract the found list items from the existing ones

Parameter	Description
	• **Intersect** - create the new list as an intersection of the existing and the found items
Filter	The type of the filter. The possible options are: • **By name** - filtering by the item's name. This option is available for any list items. • **By volume** - filtering **Subregions** by their volumes • **By area** - filtering **Boundary conditions** by their areas • **By color** - filtering **Boundary conditions** by their colors
Parameters > ...	Parameters of the filtering. The set of possible parameters depends on the selected filter type (the **Filter** parameter).
Parameters > Match type	The type of matching a text string to the specified sample. The possible options are: • **Substring** - looking for the specified substring in the names of the list items • **Wildcard** - looking for the specified substring with use of wildcard symbols "**?**" and "*****" • **Regular expression** - looking for the matching the list items to the specified regular expression
Parameters > Expression	A substring or regular expression that is to be match to names of the desired list items
Parameters > Case sensitive	This setting defines if the search of the names of the desired list items will be case sensitive.The possible options are: **Yes | No**.
Parameters > Predicate	This is a condition for filtering by numerical values (volumes of **Subregions**, areas of **Boundary conditions**) or by colors of **Boundary conditions**. The possible options are: **Less or equal | Greater or equal | In interval | Equal | Not equal**.
Parameters > Maximum	The maximal numerical value. It is used when **Predicate = Less or equal | In interval**.
Parameters > Minimum	The minimal numerical value. It is used when **Predicate = Greater or equal | In interval**.
Parameters > Color	The color used for filtering **Boundary conditions** by their colors. It is used when **Predicate = Equal | Not equal**.

Specify the settings and click **OK**. If you click **Cancel**, the dialog box will close without applying the entered settings.

8.1.8.3.15.1 Element «Initial grid»

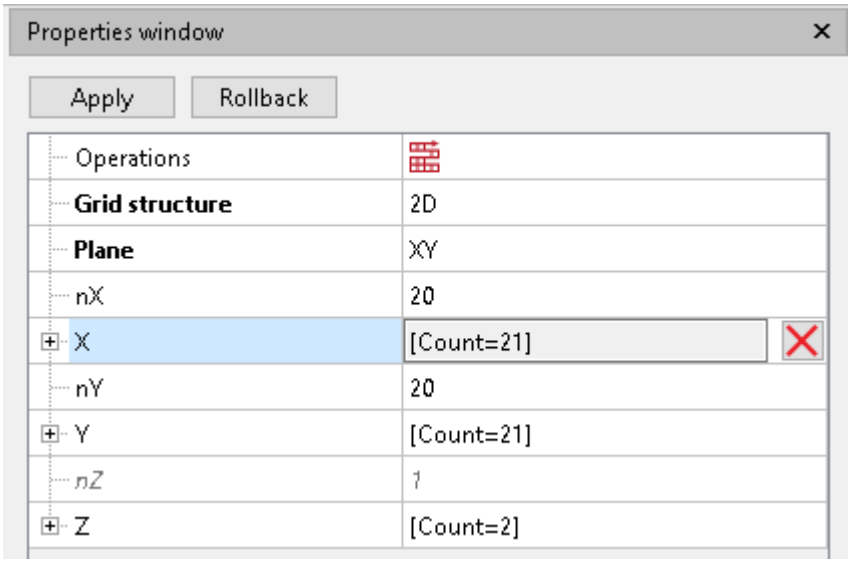


The **Initial grid** element contains parameters of the initial grid (which is the *zero level of refining* of the computational grid).

You can do the following operations with an **Initial grid**:

- Specify the number of cells in a uniform **Initial grid**'s along axes X, Y, Z.
- Specify either one or two non-computational directions, along which the computational grid will always contain only one cell.
- Set a nonuniform **Initial grid** using the **Initial grid editor**, which opens by the button in the **Properties** window.
- Add and remove grid lines by inserting/appending/deleting the appropriate elements in arrays of grid lines in the **Properties** window.
- Export and import an **Initial grid** to/from a text file using commands **Export to a text file** and **Import from a text file** from the context menu of the **Initial grid** element in the project tree.

Properties window of the element «Initial grid»



The **Properties** window of the element **Initial grid**

Parameters of the element **Initial grid**:

Parameters and buttons	Description
Operations >	Opens the **Initial grid editor**
Grid structure	This parameter determines dimension of the computational grid. Possible options are: • **3D** - a 3D grid is used, which can have multiple cells along each direction X, Y, Z. • **2D** - a 2D grid is used, which is limited by only one cell along one of the directions. • **1D** - an 1D grid is used, which is limited by only one cell along two of the directions. Also any adaptations of the computational grid will be blocked along non-computational directions. See details and illustrations in the section Computational grid. Recommendations when you use 2D or 1D grid structure: • the computational domain should be symmetrical along the non-computational direction(s); • boundaries of the computational domain should be either perpendicular or parallel to the non-computational direction(s); • in a sector problem setting, the plane of symmetry of the sector should be perpendicular to the non-computational direction(s); • type of **Boundary conditions** on surfaces limiting the computational domain from the side of a non-computational direction should be either **Symmetry** or **Connected**. In problem settings with 2D or 1D grid structure you should use the **Absolute** criterion for revealing small cells (**Small Cells > Criterion = Absolute** in properties of the appropriate **Phase Limiter**) in all cases except simulating the icing and some other cases when there are explicit recommendations to specify the **Relative** criterion for revealing small cells. It is possible to use 2D or 1D problem settings with minor asymmetry of the geometry model that can appear due to small inaccuracies or rounding errors. See details and illustrations in the section Computational grid.

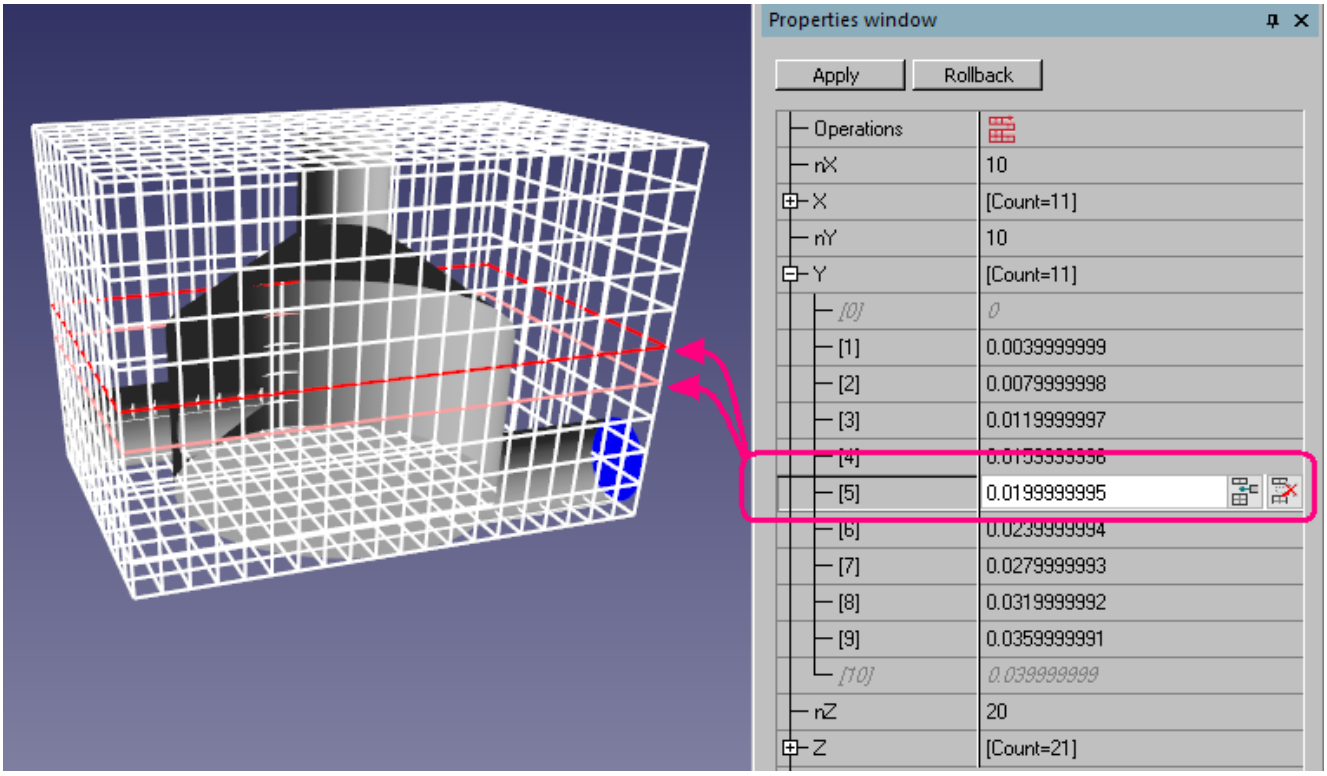
Parameters and buttons	Description
Plane	Selection of the computational plane. In the third direction the grid is limited by one cell only. Possible options are: **XY** | **YZ** | **XZ**. This parameter is available only when **Grid structure =2D**.
Direction	Selection of the computational direction. In other two directions the grid is limited by one cell only. Possible options are: **X** | **Y** | **Z**. This parameter is available only when **Grid structure =1D**.
nX	The total number of cells of the initial grid along the **X** axis.
X[*)]	Array of elements corresponding to the grid lines across the **X** axis.
X > [N] [**)] (where N= 0, 1, ..., **nX**)	Coordinates of grid lines across the **X** axis. Coordinates of the outermost line (with numbers **[0]** and **[nX]**) are not available for changing by the user.
nY	The total number of cells of the initial grid along the **Y** axis.
Y[*)]	Array of elements corresponding to the grid lines across the **Y** axis.
Y > [N] [**)] (where N= 0, 1, ..., **nY**)	Coordinates of grid lines across the **Y** axis. Coordinates of the outermost line (with numbers **[0]** and **[nY]**) are not available for changing by the user.
nZ	The total number of cells of the initial grid along the **Z** axis.
Z[*)]	Array of elements corresponding to the grid lines across the **Z** axis.
Z > [N] [**)] (where N= 0, 1, ..., **nZ**)	Coordinates of grid lines across the **Z** axis. Coordinates of the outermost line (with numbers **[0]** and **[nZ]**) are not available for changing by the user.

[*)] These fields contain the  (**Clear the array**) screen button for clearing the data array (deleting all child elements except elements corresponding to the outermost grid lines, which after the clearing the array will be designated as **[0]** and **[1]**), also you can press the **Del** key.

[**)] These fields (except fields corresponding to outermost grid lines) contain screen buttons:

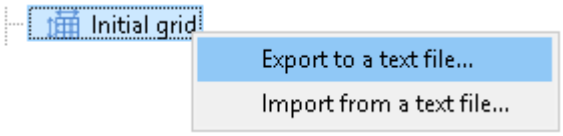
-  (**Insert item before current**) for adding a new grid line between the selected line and the previous line, also you can press the **Ins** key.
-  (**Delete this item**) for deleting the selected grid line, also you can press the **Del** key.

When some grid line (parameter **X > [N]**, **Y > [N]**, or **Z > [N]**) is selected, then, in the **View** window, the selected line is highlighted with a color and the adjacent line (between which and the selected line you can add a new line by clicking the  button) is also highlighted with another color shade.



The selected grid line and the adjacent grid line are highlighted with colors in the **View** window

Context menu of the element «Initial grid»



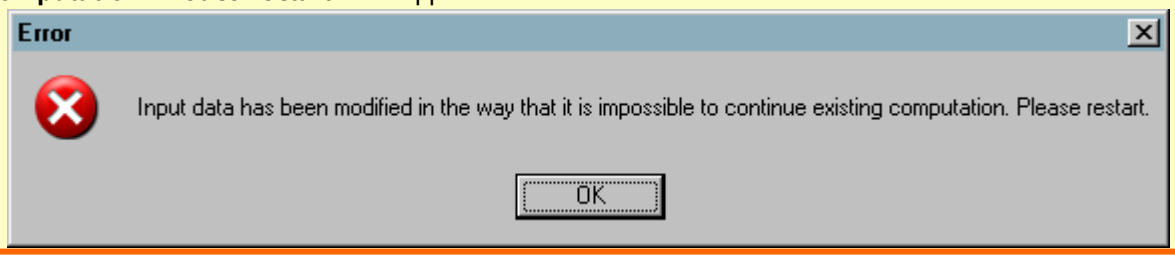
Context menu of the element **Initial grid** in the project tree

Context menu of the element **Initial grid** in the project tree:

Menu item	Description
Export to a text file	Saving the initial grid in a text file
Import from a text file	Loading a previously saved initial grid from a text file

Changing the **Initial grid** of the project, which has already been started, prevents resumption of the computation. You can only run the computation from scratch.

When you in such a situation try to continue the computation from the moment when it was suspended, the message **"Input data has been modified in the way that it is impossible to continue existing computation. Please restart."** will appear:

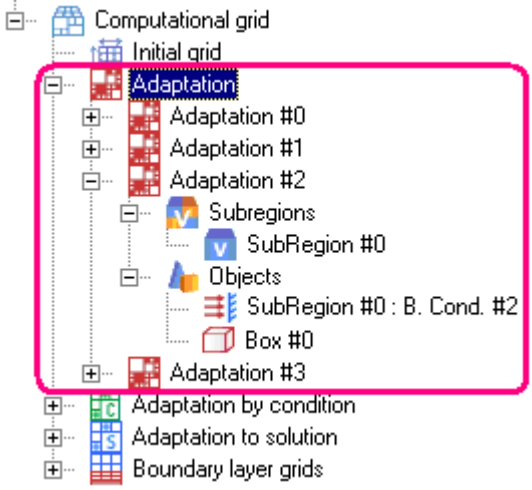


See also: sections Computational grid and Operations with initial grid.

8.1.8.3.15.2 Subfolder «Adaptation»

Adaptation is an element that defines splitting or merging of computational grid's cells up to the specified level on the surface of the specified **Boundary conditions** and/or in the volume or on the surface of the specified **Objects** within the specified **Subregions** (see the section Adaptation).

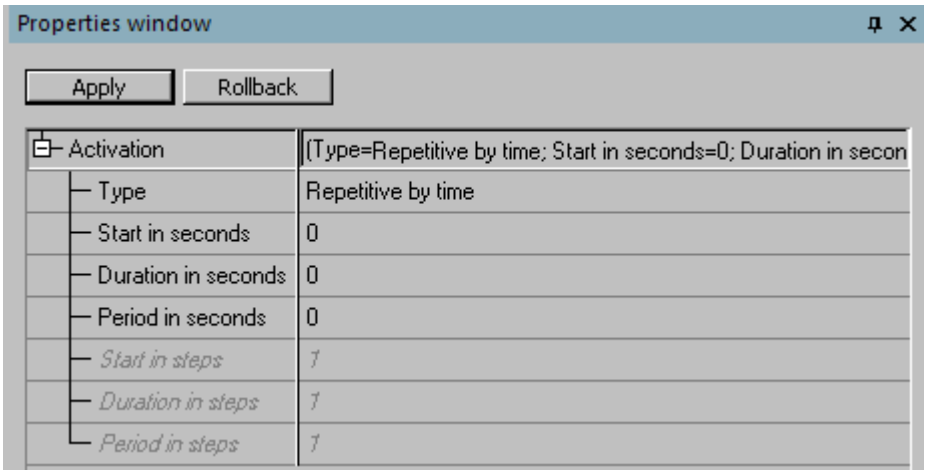
In the project tree the elements **Adaptation #N** locate in the folder **Computational grid > Adaptation**:



Each element **Adaptation #N** contains subfolders **Subregions** and **Objects** that contain elements defining the area where the adaptation is applied. If these elements are not set, the **Adaptation #N** will be marked in the project tree with the symbol "**!**".

The Properties window of the folder "Adaptation"

In the **Properties** window of the folder **Computational grid > Adaptation** the general settings for all **Adaptations #N** are configured, which define the general settings of time and frequency of their running (these settings are applied to individual **Adaptations #N**, if their running is not disabled).



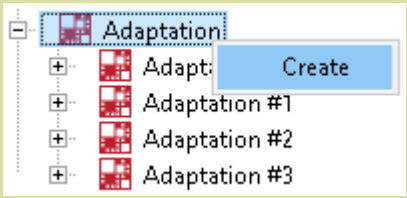
The **Properties** window of the folder **Computational grid > Adaptation**

Properties of the folder **Adaptation**:

Parameter	Description
Activation > ...	Setting of timing of the **Adaptation**. See details in the subsection *"Activity of adaptations and boundary layer grids"* of the section Folder «Computational grid».
Activation >Type	
Activation > Start in seconds	
Activation > Duration in seconds	

Parameter	Description
Activation > Period in seconds	
Activation > Start in steps	
Activation > Duration in steps	
Activation > Period in steps	

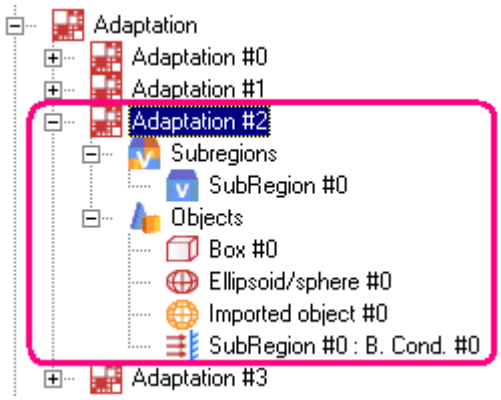
Context menu of the "Adaptation" folder



Context menu of the "Adaptation" folder

Menu item	Description
Create	This menu item creates a new element **Adaptation #N** in the folder **Adaptation**.

Elements "Adaptation #N"

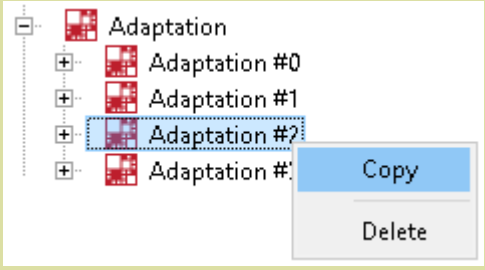


Element **Adaptation #N** in the project tree

Elements **Computational grid > Adaptation > Adaptation #N** specify separate simple **Adaptations**.

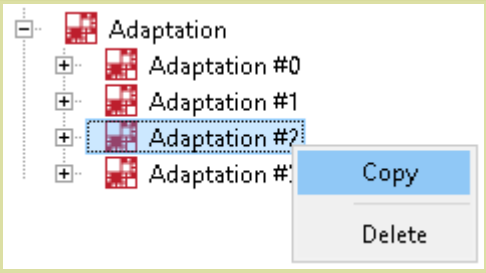
An element **Adaptation #N** contains subfolders with elements, which define the area where the adaptation acts:

- **Subregions** defines the **Subregions**, in which the adaptation acts.
- **Objects** defines on which geometry **Objects** and on which **Boundary conditions** the adaptation acts. When an adaptation is set on geometry **Objects**, you can specify where exactly it will act (in the **Object**'s volume or on all or some of **Object**'s surfaces).



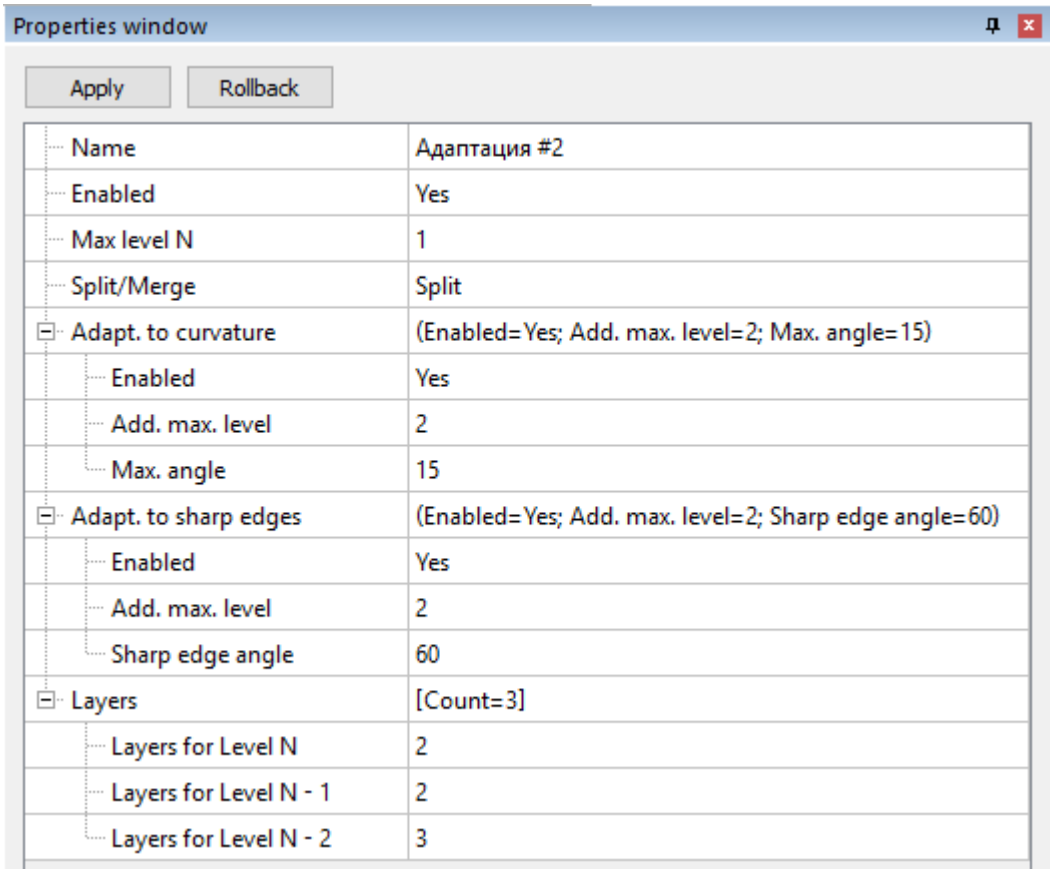
Context menu of the element "Adaptation #N"

Menu item	Description
Copy	Creates a copy of **Adaptation #N**



Context menu of the element "Adaptation #N"

Menu item	Description
Delete	Removes **Adaptation #N** from the project tree



The **Properties** window of the element **Adaptation #N**

Parameters of the element **Adaptation #N** are displayed in its **Properties** window:

Parameters of the element «Adaptation #N»

Parameter	Description
Name	Name of the **Adaptation**. By default the standard names are used: **Adaptation #N**, where N = 0, 1, 2, ...
Enabled	Use of the **Adaptation #N**. Possible options are: • **No**: specifies that the **Adaptation #N** is *not used* for building the computational grid. The **Adaptation #N**'s icon will be faded : . • **Yes**: specifies that the **Adaptation #N** *is used* for building the computational grid in accordance with the parameters of the parent folder **Adaptation**, and also every time when **Adaptation by condition** or **Adaptation to solution** runs (see details in the subsection *"Activity of adaptations and boundary layer grids"* of the section Folder «Computational grid»).

Parameters of the element «Adaptation #N»	
Parameter	**Description**
Max level N	The maximal level of adaptation of the grid cells (the maximal level of splitting), without taking into account additional levels**) appearing because of applying **Adaptation to curvature** and/or **Adaptation to sharp edges** (see below).  You can specify this value not only as a constant (≥**1**), but also as a formula or a table. This is useful to specify dependency on step number or time. If the specified value is not integer, the fractional part is dropped. If a value, which has been specified by a formula or table, is less then **1**, then the program will use the value **1**.
Split/Merge	Possible options are: • **Split**: all the cells that have the level of adaptation less than **Max level**, will be refined (split) so that the level of adaptation of new cells will be equal to **Max. level**. • **Merge**: all the cells that have the level of adaptation more then **Max level**, will be merged so that the level of adaptation of new cells will be equal to **Max. level**. • **Improve**: all cells that meet additional criteria (defined by parameters **Several neighbors**, **Concavity**, **Identic BCs** and **Different ConBCs**, see their descriptions below), will be split up the **Max. level**.
Adapt. to curvature***)	These parameters specify adaptation to curvature. **Adaptation to curvature** activates when angles between normals to facets (to any facets within one group of facets of the surface on which the **Adaptation** is set, within one cell) fall in the interval from **Max. angle** to **Upper limit** (see below). See details in the subsection *"Automatic increase of the maximal level of adaptation near curved surfaces and sharp edges"* of the section Adaptation.
Adapt. to curvature > Enabled***)	This parameter enables adaptation to curvature. Possible options are: **Yes** | **No**.
Adapt. to curvature > Add. max. level***)	Addition to the maximal level of adaptation, which is set by the **Max level N** parameter, that is added when adaptation to curvature activates.**) This parameter is available when **Adapt. to curvature > Enabled = Yes**.
Adapt. to curvature > Max. angle***)	The lower endpoint of interval, which, when the angle of normal's spreading falls in it, causes applying the adaptation to curvature. This parameter is available when **Adapt. to curvature > Enabled = Yes**.
Adapt. to curvature > Upper limit***)	The upper endpoint of interval, which, when the angle of normal's spreading falls in it, causes applying the adaptation to curvature. This parameter is available when **Adapt. to curvature > Enabled = Yes**.
Adapt. to sharp edges***)	These parameters specify adaptation to sharp edges. **Adaptation to sharp edges** activates when angles between normals to adjacent facets of the surface, on which the **Adaptation** is set, independently on belonging to facet groups, within a cell, exceed the threshold value specified by the **Sharp edge angle** parameter. See details in the subsection *"Automatic increase of the maximal level of adaptation near curved surfaces and sharp edges"* of the section Adaptation.
Adapt. to sharp edges > Enabled***)	This parameter enables adaptation to sharp edges. Possible options are: **Yes** | **No**.
Adapt. to sharp edges > Add. max. level***)	Addition to the maximal level of adaptation, which is set by the **Max level N** parameter, that is added when adaptation to sharp edges activates.**) This parameter is available when **Adapt. to sharp edges > Enabled = Yes**.
Adapt. to sharp edges > Sharp edge angle***)	This is the threshold value of the angle between normals to adjacent facets (either facets that contact the boundary between groups or those ones that

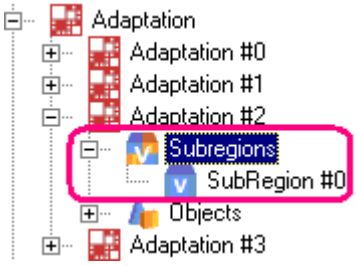
Parameters of the element «Adaptation #N»	
Parameter	**Description**
	belong to the same group); when this value is exceeded, the adaptation to sharp edges is applied. This parameter is available when **Adapt. to sharp edges > Enabled = Yes**.
Several neighbors[*)]	Specifies whether the program will split boundary cells that contact by one their side to several cells of the same or larger size.
Concavity[*)]	Specifies whether the program will split concave border cells.
Identic BCs[*)]	Specifies whether the program will split cells that have the same boundary condition on opposite faces.
Different ConBCs[*)]	Specifies whether the program will split cells that have different connected boundary conditions on opposite faces.
Layers > ...	Array of parameters, which defines the numbers of layers of different levels of the adaptation. For volumes the layers are built outwards, for surfaces they are built to both directions from the surface.
Layers > Layers for Level N	Number of adaptation layers for these levels
Layers > Layers for Level N-1	 You can specify these values not only as constants (≥**1**), but also as formulae and tables. Formulae and tables are useful to specify dependencies of these values on step number or time. If the specified value is not integer, their fractional part is dropped. If a value, which has been specified by a formula or table, is less then **1**, then the program will use the value **1**.
Layers > Layers for Level N-2 **...**	

[*)] These parameters are only available when **Split/Merge = Improve**. Their possible options are: **Yes | No**. See details in the section Adaptation.

[**)] If both **Adaptation to curvature** and **Adaptation to sharp edges** activate, their additions to the maximal level of adaptation (**Max level N**) are not applied cumulatively (the bigger addition is applied).

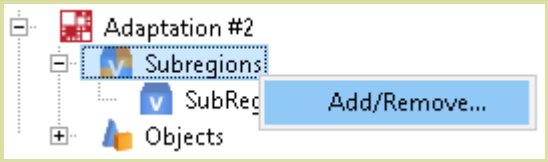
[***)] These parameters are available when **Split/Merge = Split** and **Adaptation** is set on a surface of some geometry **Object** and/or of a boundary condition (except a connected boundary condition).

Subfolder «Adaptation #N > Subregions»



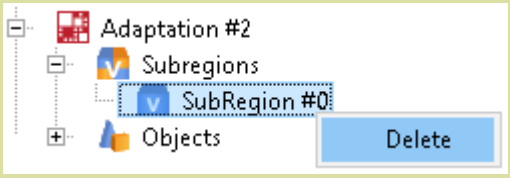
The subfolder **Adaptation #N > Subregions** contains child elements, which correspond to the **Subregions**, in limits of which **Adaptation #N** can act.

The **Properties** windows of the subfolder **Adaptation #N > Subregions** and its child elements are empty. To add or remove the child elements **Adaptation #N > Subregions > *Subregion***, use the **Add/Remove** command from the context menu.



Context menu of the subfolder «Adaptation #N > Subregions»

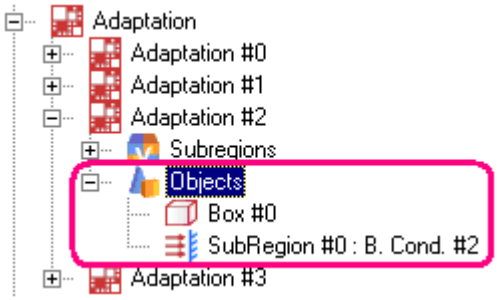
Menu item	Description
Add/Remove	Opens the **Select Subregions** dialog box, which allows you to specify the **Subregions** where **Adaptation #N** will act.



Context menu of the element «Adaptation #N > Subregions > SubRegion #N»

Menu item	Description
Delete	Removes the selected **Subregion** from the **Adaptation #N > Subregions** subfolder, so the adaptation will not act within this **Subregion**.

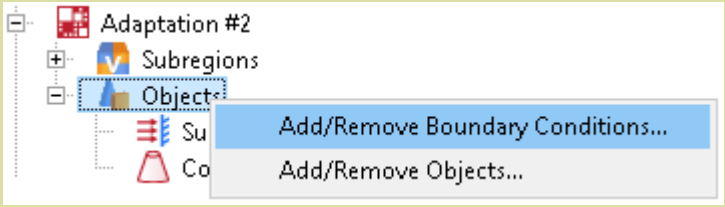
Subfolder «Adaptation #N > Objects»



The subfolder **Adaptation #N > Objects** contains child elements, which correspond to the **Objects** and **Boundary conditions**, in which (on which) **Adaptation #N** acts.

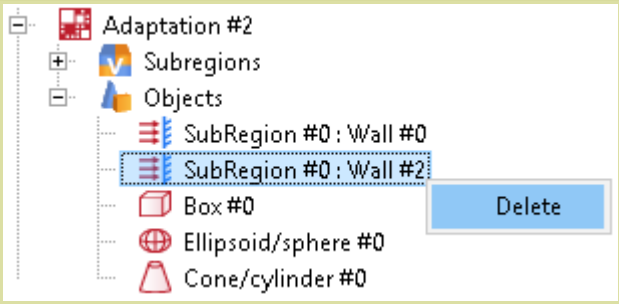
The **Properties** window of the subfolder **Adaptation #N > Objects** is empty. To add or remove the child elements to or from the subfolder **Adaptation #N > Objects**, use the commands **Add/Remove Boundary Conditions** and **Add/Remove Objects** from the context menu.

In the **Properties** windows of elements **Adaptation #N > Objects > *Object*** for some geometry **Objects** you can specify where the adaptation will act (in the volume or on the surface of the **Object**, on all or only on the selected surfaces).



Context menu of the subfolder «Adaptation #N > Objects»

Menu item	Description
Add/Remove Boundary Conditions	Opens the **Select boundary conditions** dialog box, which allows you to specify the **Boundary conditions**, on which **Adaptation #N** will act.
Add/Remove Objects	Opens the **Select objects** dialog box, which allows you to specify the **Objects**, where **Adaptation #N** will act.



Context menu of the element «Adaptation #N > Objects > Object»

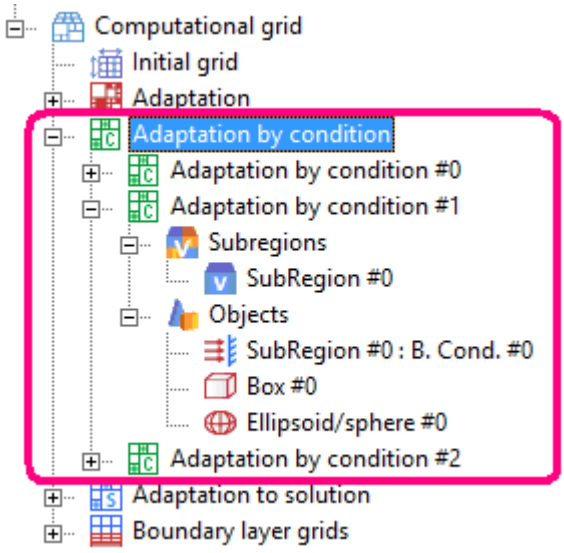
Menu item	Description
Delete	Removes the selected **Boundary Condition** or geometry **Object** from the subfolder **Adaptation #N > Objects**, so the adaptation will not act on this **Boundary Condition** or in/on this **Object**.

8.1.8.3.15.3 Subfolder «Adaptation by condition»

Adaptation by condition allows the program to split cells of the computational grid depending on conditions that are set to the variables.

As for the simple **Adaptation**, action of an **Adaptation by condition** is set on **Boundary conditions** and/or in/on geometry **Objects** and is limited by the specified **Subregions**.

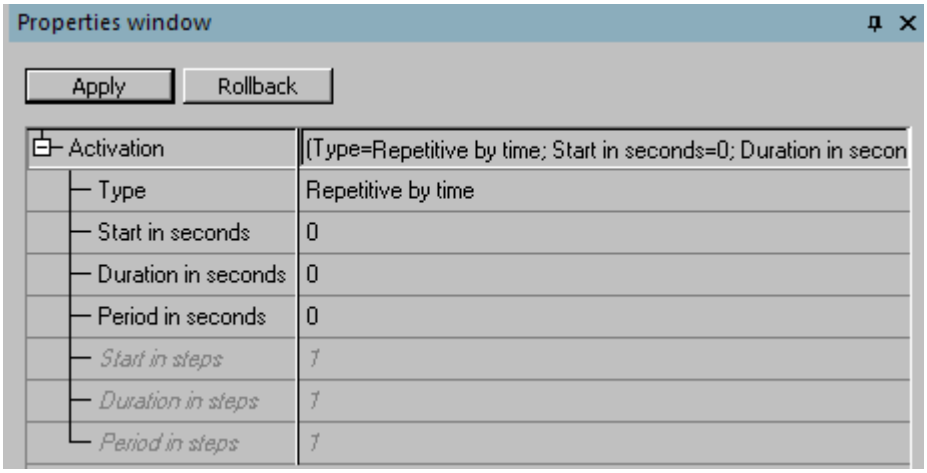
In the project tree the elements **Adaptation by condition #N** locate in the folder **Computational grid > Adaptation by condition**:



Each element **Adaptation by condition #N** contains subfolders **Subregions** and **Objects** that contain elements defining the area where the adaptation is applied. If these elements are not set, the **Adaptation by condition #N** will be marked in the project tree with the symbol "**!**".

The Properties window of the folder "Adaptation by condition"

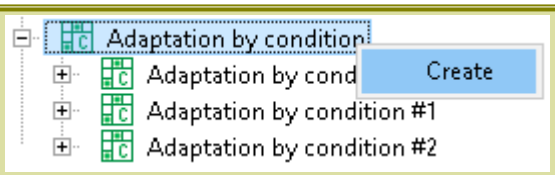
In the **Properties** window of the folder **Computational grid > Adaptation by condition** the general settings for all **Adaptations by condition #N** are configured, which define the general settings of time and frequency of their running (these settings are applied to individual **Adaptations by condition #N**, if their running is not disabled).



The **Properties** window of the folder **Computational grid > Adaptation by condition**

Parameters of the folder "Adaptation by condition"	
Parameter	**Description**
Activation > ...	Setting of timing of the **Adaptation by condition**. See details in the subsection *"Activity of adaptations and boundary layer grids"* of the section Folder «Computational grid».
Activation >Type	
Activation > Start in seconds	
Activation > Duration in seconds	
Activation > Period in seconds	
Activation > Start in steps	
Activation > Duration in steps	
Activation > Period in steps	

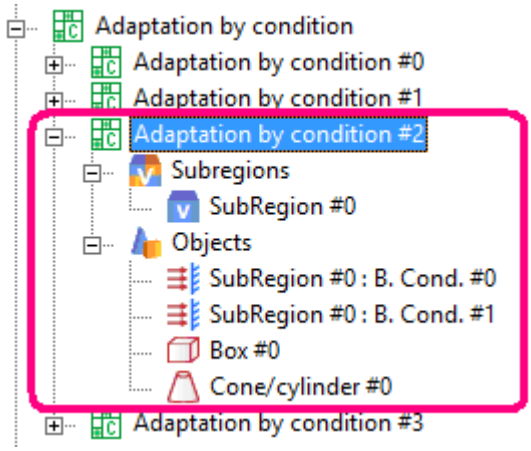
Context menu of the "Adaptation by condition" folder



Context menu of the "Adaptation by condition" folder

Menu item	Description
Create	Creates a new element **Adaptation by condition #N** in the folder **Adaptation by condition**

Elements "Adaptation by condition #N"

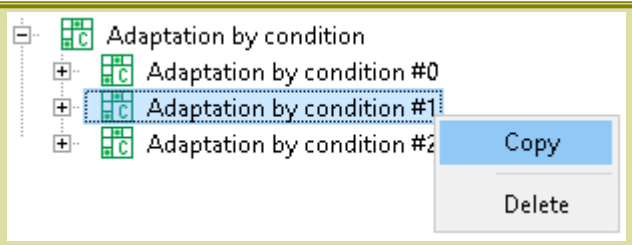


Element **Adaptation by condition #N** in the project tree

Elements **Computational grid > Adaptation by condition > Adaptation by condition #N** specify individual **Adaptations by condition**.

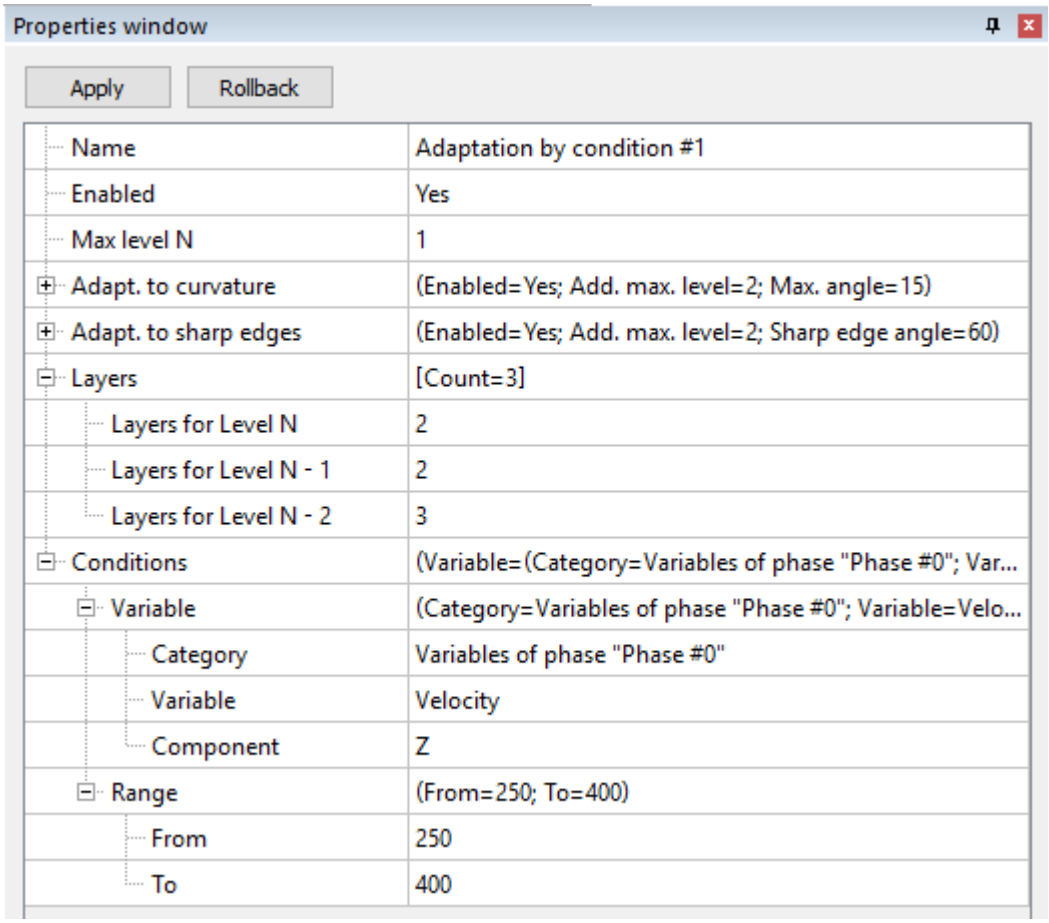
An element **Adaptation by condition #N** contains the following subfolders with elements, which define the area where the adaptation by condition acts:

- **Subregions** defines the **Subregions**, in which the adaptation acts.
- **Objects** defines in/on which geometry **Objects** and on which **Boundary conditions** the adaptation by condition acts. When an adaptation by condition is set on geometry **Objects**, you can specify where exactly it will act (in the **Object**'s volume or on all or some of **Object**'s surfaces).



Context menu of the element "Adaptation by condition #N"

Menu item	Description
Copy	Creates a copy of the **Adaptation by condition #N**
Delete	Removes the **Adaptation by condition #N** from the project tree



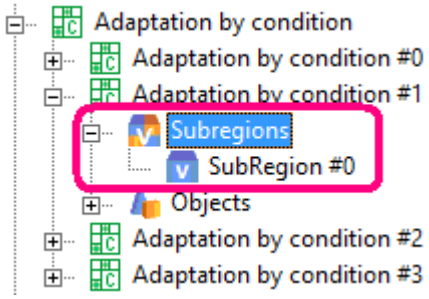
The **Properties** window of the element **Adaptation by condition #N**

Parameters of the element **Adaptation by condition #N** are displayed in its **Properties** window:

Parameters of the element «Adaptation by condition #N»	
Parameter	**Description**
Name	Name of the **Adaptation by condition**. By default the standard names are used: **Adaptation by condition #N**, where N = 0, 1, 2, ...
Enabled	Use of the **Adaptation by condition #N**. Possible options are: • **No**: specifies that the **Adaptation by condition #N** is *not used* for building the computational grid. The **Adaptation by condition #N**'s icon will be faded:  . • **Yes**: specifies that the **Adaptation by condition #N** *is used* for building the computational grid in accordance with the parameters of the parent folder **Adaptation**, and also every time when **Adaptation to solution** runs (see details in the subsection *"Activity of adaptations and boundary layer grids"* of the section Folder «Computational grid»).
Max level N	The maximal level of the adaptation of the grid cells (the maximal level of splitting).
Adapt. to curvature > ...	Settings of adaptation to curvature and to sharp edges that are similar to settings applied for simple Adaptations. See details in the subsection *"Automatic increase of the maximal level of adaptation near curved surfaces and sharp edges"* of the section Adaptation.
Adapt. to sharp edges > ...	
Layers > ...	Array of parameters, which defines the numbers of layers of different levels of the adaptation. For volumes the layers are built outwards, for surfaces they are built to both directions from the surface.
Layers > Layers for Level N	Number of adaptation layers for these levels

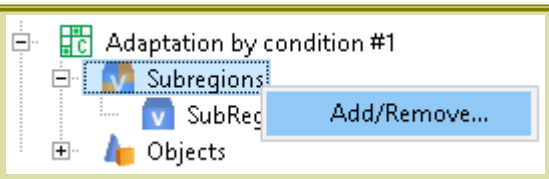
Parameters of the element «Adaptation by condition #N»	
Parameter	**Description**
Layers > Layers for Level N-1 **Layers > Layers for Level N-2** **...**	You can specify these values not only as constants, but also as formulae and tables.
Conditions > ...	Conditions for applying the adaptation
Conditions > Variable > ...	Variable, which can meet the conditions, that cause applying the adaptation
Conditions > Variable > Category	Selection of a category of the **Variable**. Possible options are: • **Common and phase-unrelated variables** • **Variables of phase "Phase #N"** • **User variables** See details in the section Categories of variables.
Conditions > Variable > Variable	The **Variable**, which is selected from the drop-down list of variables of the selected **Category**
Conditions > Variable > Component	A component or the absolute magnitude of a vector **Variable**. The possible options are: • **Length**: the absolute magnitude of the vector **Variable** (the length of the vector) • **X**: the component of the vector **Variable** along the X axis • **Y**: the component of the vector **Variable** along the Y axis • **Z**: the component of the vector **Variable** along the Z axis
Conditions > Range > From **Conditions > Range > To**	The range, into which the value of a scalar **Variable** or vector **Variable**'s **Component** is fall to cause applying the adaptation.

Subfolder «Adaptation by condition #N > Subregions»

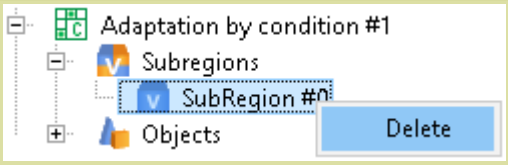


The subfolder **Adaptation by condition #N > Subregions** contains child elements, which correspond to the **Subregions**, in limits of which **Adaptation by condition #N** can act.

The **Properties** windows of the subfolder **Adaptation by condition #N > Subregions** and its child elements are empty. To add or remove the child elements **Adaptation by condition #N > Subregions > *Subregion***, use the **Add/Remove** command from the context menu.



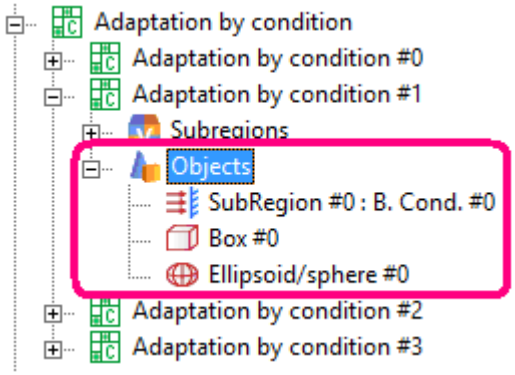
Context menu of the subfolder «Adaptation by condition #N > Subregions»	
Menu item	**Description**
Add/Remove	Opens the Select Subregions dialog box, which allows you to specify the **Subregions** where **Adaptation by condition #N** will act.



Context menu of the element «Adaptation by condition #N > Subregions > SubRegion #N»

Menu item	Description
Delete	Removes the selected **Subregion** from the **Adaptation by condition #N > Subregions** subfolder, so the adaptation will not act within this **Subregion**.

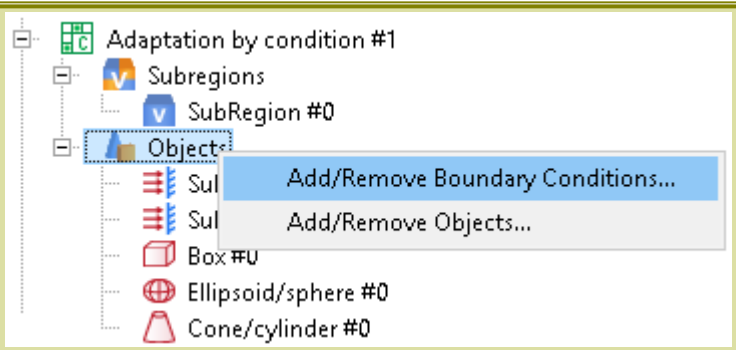
Subfolder «Adaptation by condition #N > Objects»



The subfolder **Adaptation by condition #N > Objects** contains child elements, which correspond to the **Objects** and **Boundary conditions**, in which (on which) **Adaptation by condition #N** acts.

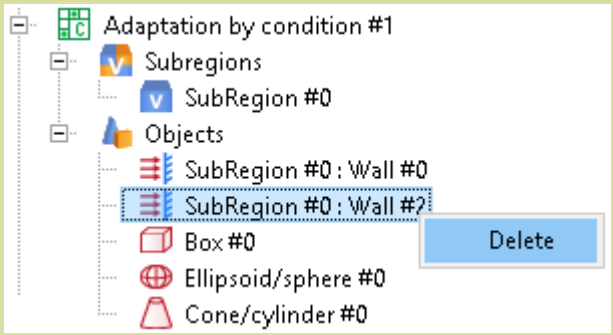
The **Properties** window of the subfolder **Adaptation by condition #N > Objects** is empty. To add or remove child elements to or from the subfolder **Adaptation by condition #N > Objects**, use the commands **Add/Remove Boundary Conditions** and **Add/Remove Objects** from the context menu.

In the **Properties** windows of elements **Adaptation by condition #N > Objects > *Object*** for some geometry **Objects** you can specify where the adaptation will act (in the volume or on the surface of the **Object**, on all or only on the selected surfaces).



Context menu of the subfolder «Adaptation by condition #N > Objects»

Menu item	Description
Add/Remove Boundary Conditions	Opens the **Select boundary conditions** dialog box, which allows you to specify the **Boundary conditions**, on which **Adaptation by condition #N** will act.
Add/Remove Objects	Opens the **Select objects** dialog box, which allows you to specify the **Objects**, where **Adaptation by condition #N** will act.



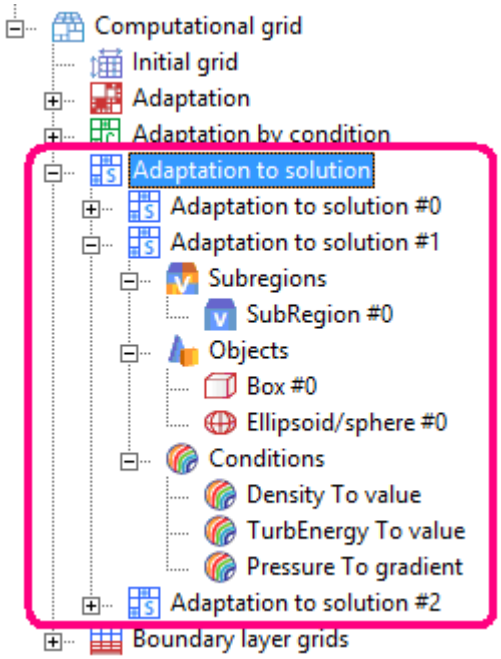
Context menu of the element «Adaptation by condition #N > Objects > *Object*»

Menu item	Description
Delete	Removes the selected **Boundary Condition** or geometry **Object** from the subfolder **Adaptation by condition #N > Objects**, so the adaptation will not act on this **Boundary Condition** or in/on this **Object**.

8.1.8.3.15.4 Subfolder «Adaptation to solution»

Adaptation to solution is the element, which splits or merges the cells within the specified **Object** in the area of the specified values of a variable value or of the maximum gradient of a variable.

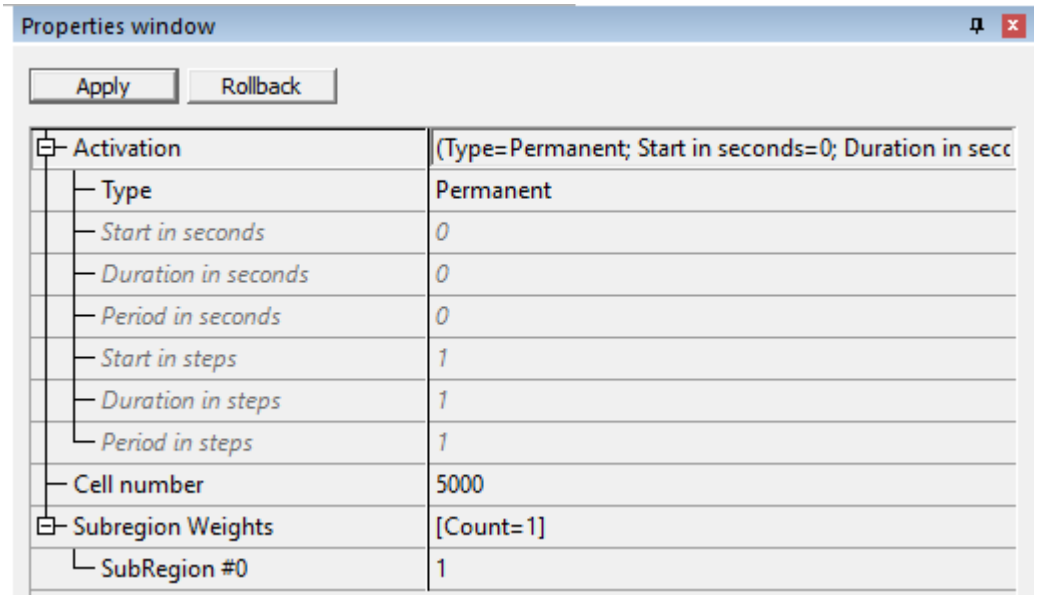
Condition of adaptation to solution is the value or gradient of the variable, by which the adaptation is set. One adaptation can have several **Conditions**.



See also: the section Adaptation to solution.

The Properties window of the folder "Adaptation to the solution"

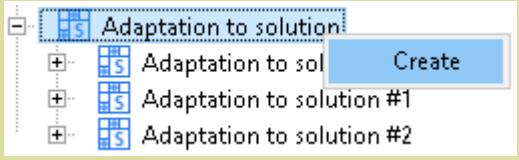
Parameters, which are set in the **Properties** window of the folder **Computational grid > Adaptation to solution**, contain general settings for *all* child elements **Adaptation to solution #N**.



The **Properties** window of the folder **Adaptation to solution**

Parameters of the folder "Adaptation by condition"	
Parameter	**Description**
Activation > ... **Activation >Type** **Activation > Start in seconds** **Activation > Duration in seconds** **Activation > Period in seconds** **Activation > Start in steps** **Activation > Duration in steps** **Activation > Period in steps**	Setting of timing of the **Adaptation to solution**. See details in the subsection *"Activity of adaptations and boundary layer grids"* of the section Folder «Computational grid».
Cell number	The maximum allowable total number of computational cells
Subregion Weights > ... **Subregion Weights > SubRegion #N**	Weights of individual **Subregions**, in which the **Adaptations to solution** act, used for limiting numbers of cells

Context menu of the "Adaptation to the solution" folder



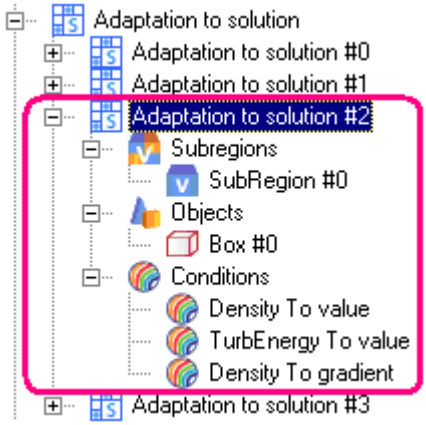
Context menu of the folder "Adaptation to the solution"	
Menu item	**Description**
Create	Creates a new element **Adaptation to solution #N** in the folder **Adaptation to solution**

Element "Adaptation to solution #N»

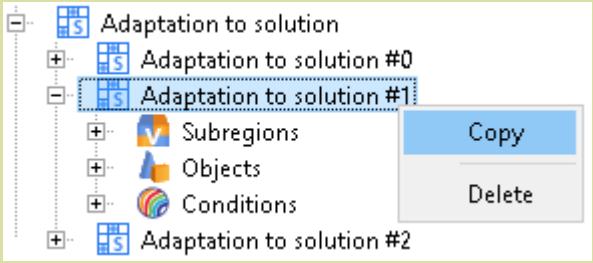
Elements **Computational grid > Adaptation to solution > Adaptation to solution #N** specify individual **Adaptations to solution**.

An element **Adaptation to solution #N** contains the following subfolders:

- **Subregions** - defines the **Subregions**, in which the adaptation to solution acts.
- **Objects** - defines in/on which geometry **Objects** the adaptation to solution acts. In the properties of the child elements of this subfolder you specify where exactly the adaptation will act (in the **Object**'s volume or on all or some of **Object**'s surfaces).
- **Conditions** - defines the conditions, when the adaptation to solution will act. Each child element of this subfolder define a condition on the value or on the gradient of some variable.



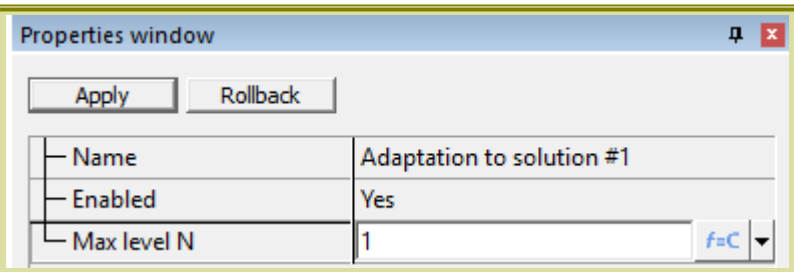
Element **Adaptation to solution #N** in the project tree



Context menu of the element "Adaptation to solution #N"

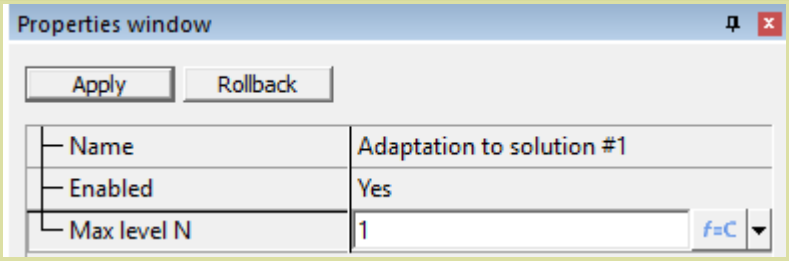
Menu item	Description
Copy	Creates a copy of the **Adaptation to solution #N**
Delete	Removes the **Adaptation to solution #N** from the project tree

Parameters of the element **Adaptation to solution #N** are displayed in its **Properties** window:



Parameters of the element «Adaptation to solution #N»

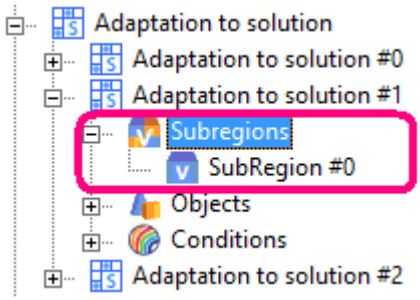
Parameter	Description
Name	Name of the **Adaptation to solution**. By default the standard names are used: **Adaptation to solution #N**, where N = 0, 1, 2, ...
Enabled	Use of the **Adaptation to solution #N**. Possible options are:



Parameters of the element «Adaptation to solution #N»

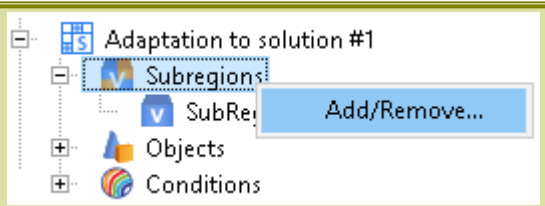
Parameter	Description
	• **No**: specifies that the **Adaptation to solution #N** is *not used* for building the computational grid. The **Adaptation to solution #N**'s icon will be faded:  . • **Yes**: specifies that the **Adaptation to solution #N** *is used* for building the computational grid in accordance with the parameters of the parent folder **Adaptation**.
Max level N	The maximal level of the adaptation of the grid cells (the maximal level of splitting).

Subfolder «Adaptation to solution #N > Subregions»



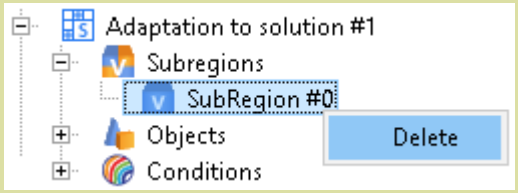
The subfolder **Adaptation to solution #N > Subregions** contains child elements, which correspond to the **Subregions**, in limits of which **Adaptation to solution #N** can act.

The **Properties** windows of the subfolder **Adaptation to solution #N > Subregions** and its child elements are empty. To add or remove the child elements **Adaptation to solution #N > Subregions > *Subregion***, use the **Add/Remove** command from the context menu.



Context menu of the subfolder «Adaptation to solution #N > Subregions»

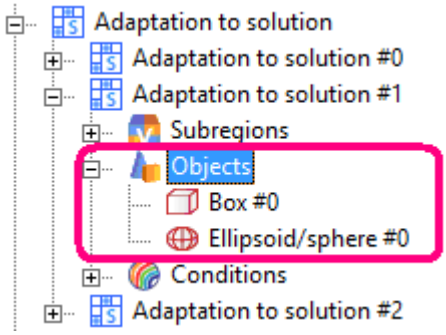
Menu item	Description
Add/Remove	Opens the Select Subregions dialog box, which allows you to specify the **Subregions** where **Adaptation to solution #N** will act.



Context menu of the element «Adaptation to solution #N > Subregions > SubRegion #N»

Menu item	Description
Delete	Removes the selected **Subregion** from the **Adaptation to solution #N > Subregions** subfolder, so the adaptation will not act within this **Subregion**.

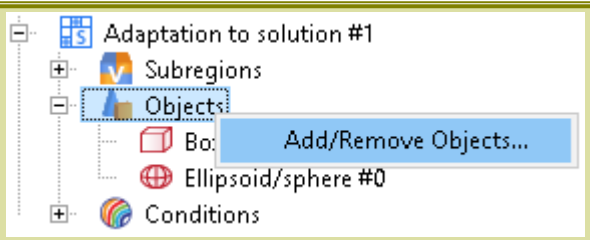
Subfolder «Adaptation to solution #N > Objects»



The subfolder **Adaptation to solution #N > Objects** contains child elements, which correspond to geometry **Objects** in/on which the **Adaptation to solution #N** acts.

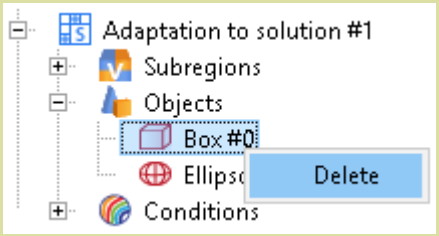
The **Properties** window of the subfolder **Adaptation to solution #N > Objects** is empty. To add or remove child elements to or from the subfolder **Adaptation to solution #N > Objects**, use the command **Add/Remove Objects** from the context menu.

In the **Properties** windows of the child elements **Adaptation to solution #N > Objects > *Object*** you can specify where specifically the adaptation to solution will act (in the volume or on the surface of the **Object**, on all or only on the selected surfaces).



Context menu of the subfolder «Adaptation to solution #N > Objects»

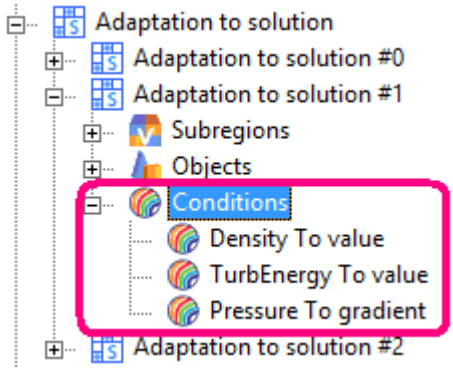
Menu item	Description
Add/Remove Objects	Opens the Select objects dialog box, which allows you to specify the **Objects**, where **Adaptation to solution #N** will act.



Context menu of the element «Adaptation to solution #N > Objects > *Object*»

Menu item	Description
Delete	Removes the selected geometry **Object** from the subfolder **Adaptation to solution #N > Objects**, so the adaptation to solution will not act in/on this **Object**.

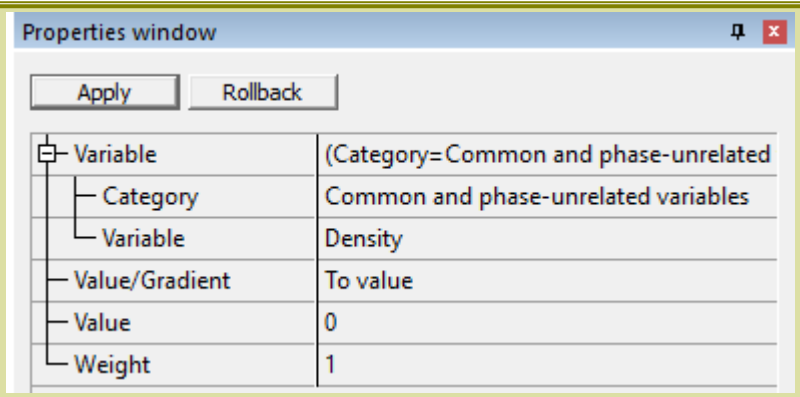
Subfolder «Adaptation to solution #N > Conditions»



The subfolder **Adaptation to solution #N > Conditions** contains child elements, which correspond to individual conditions on values or gradients of variables, defining where the adaptation to solution will act.

The **Properties** window of the subfolder **Adaptation to solution #N > Conditions** is empty.

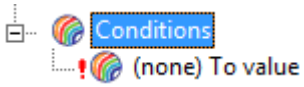
The **Properties** window of the of the child elements **Adaptation to solution #N > Conditions > *Condition of adaptation to solution*** contains the following parameters:



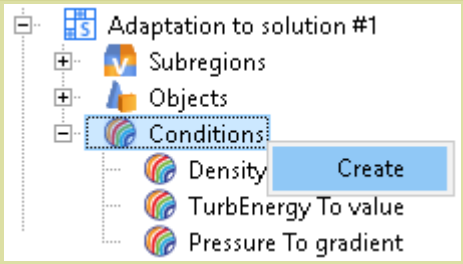
Parameters of the element «Adaptation to solution #N > Conditions > *Condition of adaptation to solution*»

Parameter	Description
Variable > ...	Parameters that define the **Variable**, the value or gradient of which cause applying of this condition of adaptation to solution
Variable > Category	Selection of a category of the **Variable**. Possible options are: • **Common and phase-unrelated variables** • **Variables of phase "Phase #N"** • **User variables** See details in the section Categories of variables.
Variable > Variable	The **Variable**, which is selected from the drop-down list of variables of the selected **Category**
Value/Gradient	Possible options are: • **To value** - the adaptation is done near the specified value of the **Variable** • **To gradient** - the adaptation is done near maximal gradients of the **Variable**
Value	Value of the **Variable**, near which the **Adaptation to solution** will act (this parameter is available if **Value/Gradient = To value**)
Weight	Weight of this condition of adaptation to solution. Cells for individual ***Condition of adaptation to solution*** are allocated proportionally to the specified **Weights**: the bigger is **Weight**, the more cells will be adapted according to this ***Condition of adaptation to solution***.

Initially, at the time of its creation, the subfolder **Adaptation to solution #N > Conditions** already contains an element **"(none) To value"** (which is marked by the "**!**" sign because no data is specified in it):

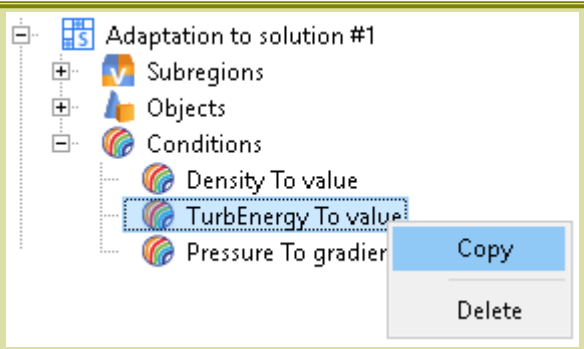


The "**(none) To value**" element is created automatically for your convenience, so you don't spend time on creating an element ***Condition of adaptation to solution*** and can immediately fill its **Properties**.



Context menu of the subfolder «Adaptation to solution #N > Conditions»

Menu item	Description
Create	Creates a new element ***Condition of adaptation to solution*** into the subfolder **Adaptation to solution #N > Conditions**. An element "**(none) To value**", marked with "**!**" and which is similar the element automatically initially created at creation of the **Adaptation to solution #N**, will be added: (none) To value



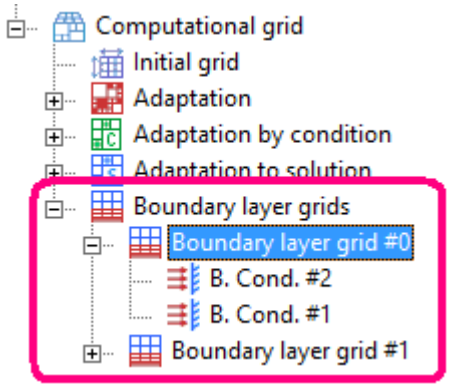
Context menu of the element «Adaptation to solution #N > Conditions > *Condition of adaptation to solution*»

Menu item	Description
Copy	Adds into the subfolder **Adaptation to solution #N > Conditions** a copy of the element ***Condition of adaptation to solution***
Delete	Remove the element ***Condition of adaptation to solution*** from the subfolder **Adaptation to solution #N > Conditions**

8.1.8.3.15.5 Subfolder «Boundary layer grids»

The *Boundary layer grid (BL grid)* is an one-dimensional adaptation, which allows solving a boundary layer by an additional computational grid with flat cells.

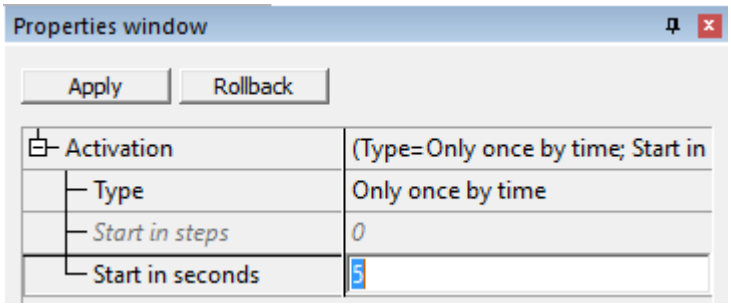
You can set several BL grids, and each of them can act in its own **Subregion** and on specified **Boundary conditions** (but the program can run only on of them at at one time). Individual BL grids are presented in the project tree by elements **Computational grid > Boundary layer grids > Boundary layer grid #N**.



If a **Subregion** and **Boundary conditions**, in/on which a **Boundary layer grid #N** should act, are not specified, then this **Boundary layer grid #N** will be marked in the project tree with the symbol "**!**".

The Properties window of the folder «Boundary layer grids»

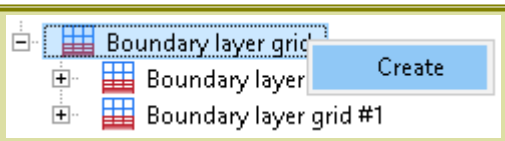
In the **Properties** window of the folder **Computational grid > Boundary layer grids** contains general settings for all **Boundary layer grid #N** that are specified. These settings define timing when the **Boundary layer grid #N** elements act (these settings are applied to every **Boundary layer grid #N**).



The **Properties** window of the folder **Computational grid > Boundary layer grids**

Parameters of the folder "Boundary layer grids"	
Parameter	**Description**
Activation > ...	Setting of timing of the **Boundary layer grids**. See details in the subsection *"Activity of adaptations and boundary layer grids"* of the section Folder «Computational grid».
Activation >Type	
Activation > Start in seconds	
Activation > Start in steps	

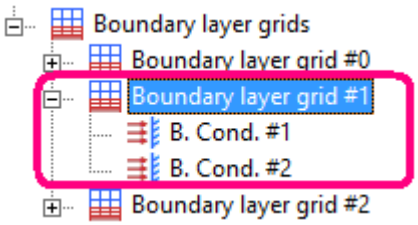
Context menu of the "Boundary layer grids" folder



Context menu of the "Boundary layer grids" folder

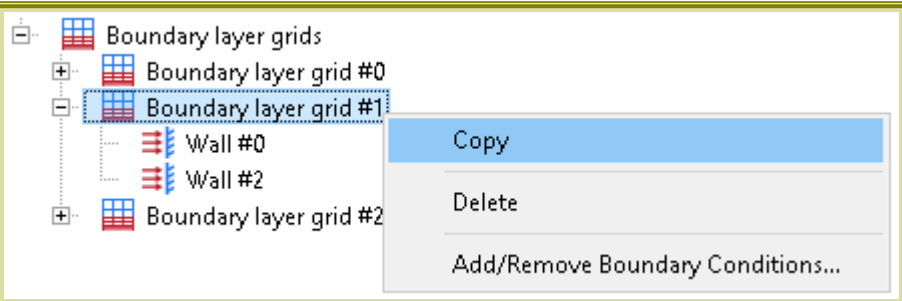
Menu item	Description
Create	Creates a new element **Boundary layer grid #N** in the folder **Boundary layer grids**

Elements «Boundary layer grid #N»

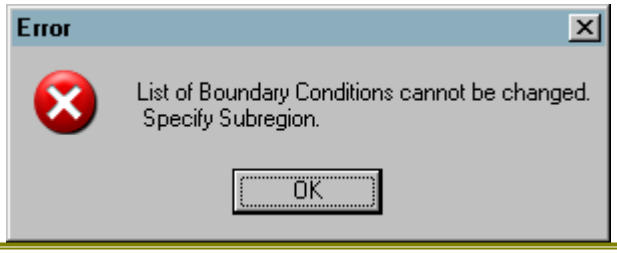


Elements **Computational grid > Boundary layer grids > Boundary layer grid #N** specify individual **Boundary layer grids**.

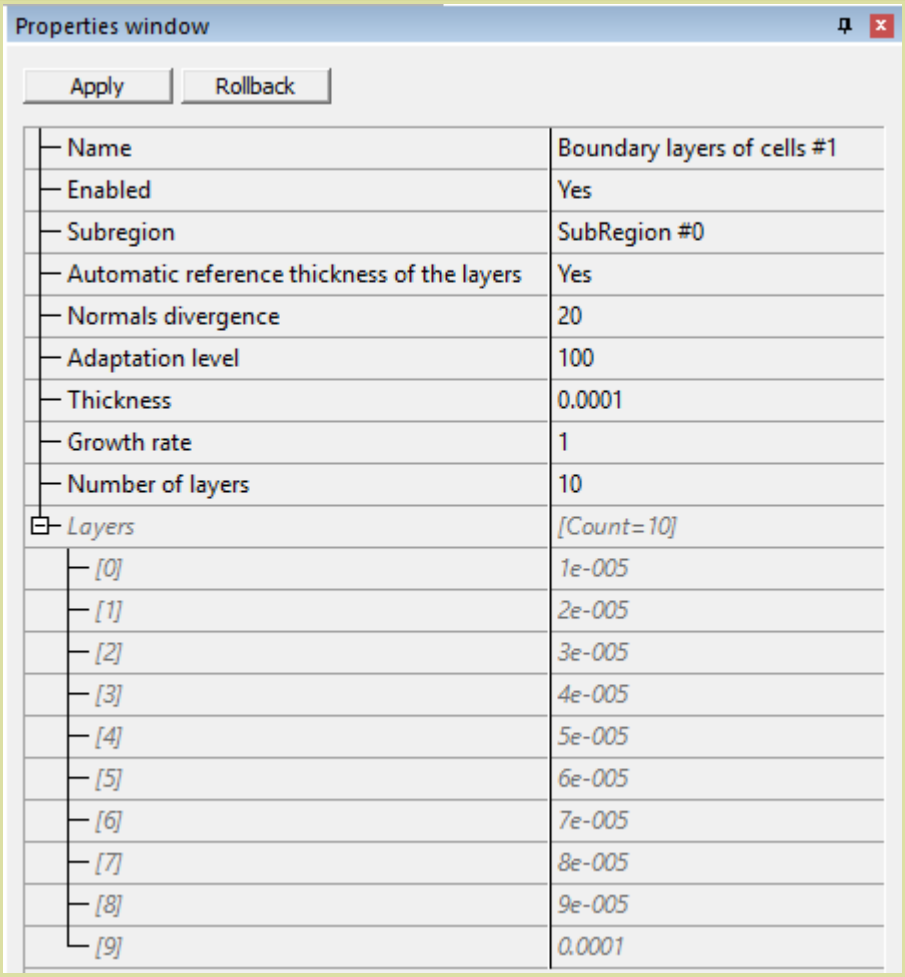
An element **Boundary layer grid #N** contains child elements **B.Cond #N**, which correspond to the **Boundary conditions**, on which the BL grid will be built (these **Boundary conditions** can be only **Walls**). The **Properties** windows of the child elements **Boundary layer grid #N > B.Cond #N** are empty and context menus of these elements contains commands **Delete** and **Properties** only.



Context menu of the element «Boundary layer grid #N»

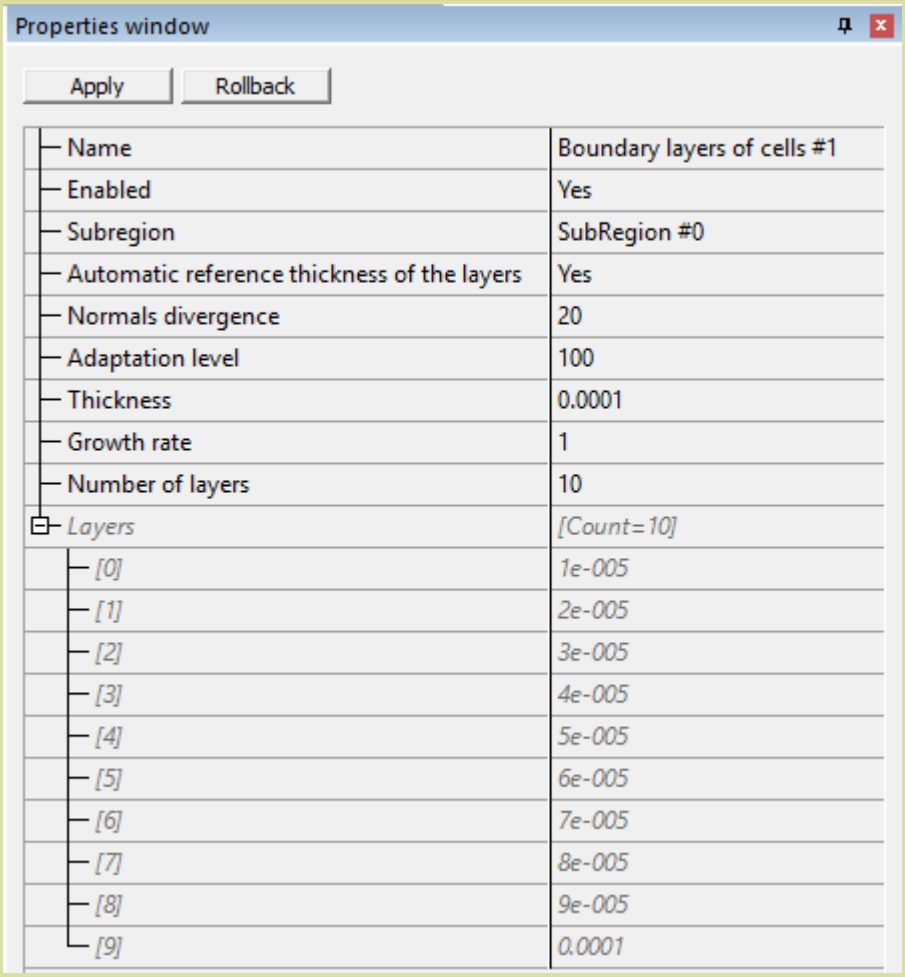
Menu item	Description
Copy	Creates a copy of the **Boundary layer grid #N** element
Delete	Removes the **Boundary layer grid #N** element from the project tree
Add/Remove Boundary Conditions	Allows you to add or remove child elements that correspond to the **Boundary conditions**, on which this BL grid (**Boundary layer grid #N**) will be built (these **Boundary conditions** can be only **Walls**). The command opens the **Select boundary conditions** dialog box, where you can add or remove **Boundary conditions**. This command can only be selected if some **Subregion** is specified in the properties of the **Boundary layer grid #N**, otherwise an error message (**List of Boundary Conditions cannot be changed. Specify Subregion.**) would be displayed: 

Parameters of a **Boundary layer grid #N** element are displayed in its **Properties** window:



Parameters of a "Boundary layer grid #N" element

Parameter	Description
Name	Name of the BL grid object. By default the standard names are used: **Boundary layer grid #N**, where N = 0, 1, 2, ...
Enabled	This parameter defines usage of this boundary layer grid. Possible options: • **Yes** • **No** (at this selection the icon of **Boundary layer grid #N** is faded:)
Subregion	The **Subregion** where the **Boundary layer grid #N** acts
Automatic reference thickness of the layers	This parameter defines the algorithm for layer distance definition in the boundary layer. The possible options are: • **Yes**- the distances between adjacent layers of the boundary layer are identical or vary in geometric progression with a ratio set by the **Growth rate** parameter. The distance to every layer is calculated by the program automatically. • **No** - distances to each layer are entered manually
Normals divergence	Maximal value of divergence of normals on the surface in one cell [Degree]. When this value is exceeded, the boundary layer grid is not used for calculations in this and adjacent cells.
Adaptation level	The maximal adaptation level of the main grid at which the BL grid will be built. The BL grid is built on those only areas of the surface, which are adjacent to the cells with adaptation level that is not greater then this parameter.
Thickness	The overall *thickness of the boundary layer* (i.e., the *summary thickness of all layers*); it is defined when thickness of layers is calculated automatically. This parameter is available when **Automatic reference thickness of the layers** = **Yes**.



Parameters of a "Boundary layer grid #N" element

Parameter	Description
	The boundary layer grid must include the whole boundary layer.
Growth rate	The geometric progression ratio that defines the layer thickness variation in the boundary layer (when automatic setting of thickness of the layers is enabled). If this parameter is set to **1**, the layers go with a same (constant) step. This parameter is available when **Automatic reference thickness of the layers = Yes**.
Number of layers	The number of layers in the BL grid.
Layers > N N=0, 1, 2, ...	The distances from the surface to the outer boundaries of the boundary layers (they are labeled as y_0, y_1, ... in the illustration in the sectionBoundary layer grid (BL grid). If **Automatic reference thickness of the layers = Yes**, these distances are calculated automatically. If **Automatic reference thickness of the layers = No**, distances to the layers can be specified manually. ***Note:*** The layers are enumerated starting from zero, so the first layer has index 0 (see examples).

See also: sections Boundary layer grid (BL grid), Theory > Physical processes > Boundary layer grid.

Examples of setting the thickness of layers of the boundary layer grid

Example 1.

Values of parameters:

- **Automatic reference thickness of the layers = Yes**

- **Thickness = 0.05**
- **Growth rate = 1**
- **Number of layers = 5**

All layers will have the same thickness.

Layer number counting from the surface with a boundary condition	Layer j index	y_j (distance to the layer's boundary)	Layer thickness
1st	0	0.01	0.01
2nd	1	0.02	0.01
3rd	2	0.03	0.01
4th	3	0.04	0.01
5th	4	0.05	0.01

Example 2.

Values of parameters:

- **Automatic reference thickness of the layers = Yes**
- **Thickness = 0.31**
- **Growth rate = 2**
- **Number of layers = 5**

Each next layer is twice thicker.

Layer number counting from the surface with a boundary condition	Layer j index	y_j (distance to the layer's boundary)	Layer thickness
1st	0	0.01	0.01
2nd	1	0.03	0.02
3rd	2	0.07	0.04
4th	3	0.15	0.08
5th	4	0.31	0.16

Example 3.

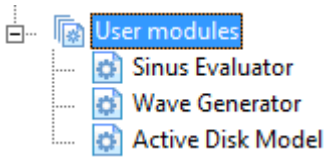
Values of parameters:

- **Automatic reference thickness of the layers = No**
- **Number of layers = 4**
- **Thickness**- the parameter is not available
- **Growth rate**- the parameter is not available
- **Layers >[0]**, **Layers > [1]**, **Layers > [2]**, **Layers > [3]** - are now available for entering values

Distances to outer surfaces of layers are defined by user manually individually for each layer.

Layer number counting from the surface with a boundary condition	Layer j index	y_j (distance to the layer's boundary)	Layer thickness
1st	0	Defined by parameter **Layers > [0]**	Defined by parameter **Layers > [0]**
2nd	1	Defined by parameter **Layers > [1]**	Difference between **Layers > [1]** and **Layers > [0]**
3rd	2	Defined by parameter **Layers > [2]**	Difference between **Layers > [2]** and **Layers > [1]**
4th	3	Defined by parameter **Layers > [3]**	Difference between **Layers > [3]** and **Layers > [2]**

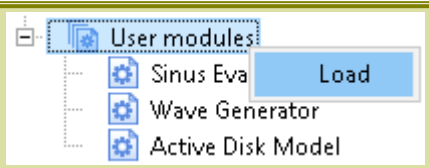
8.1.8.3.16 Folder "User modules"



The **User modules** folder contains elements that correspond to user modules from external developers, which have been loaded into the current project.

The **Properties** window of the **User modules** folder has no parameters.

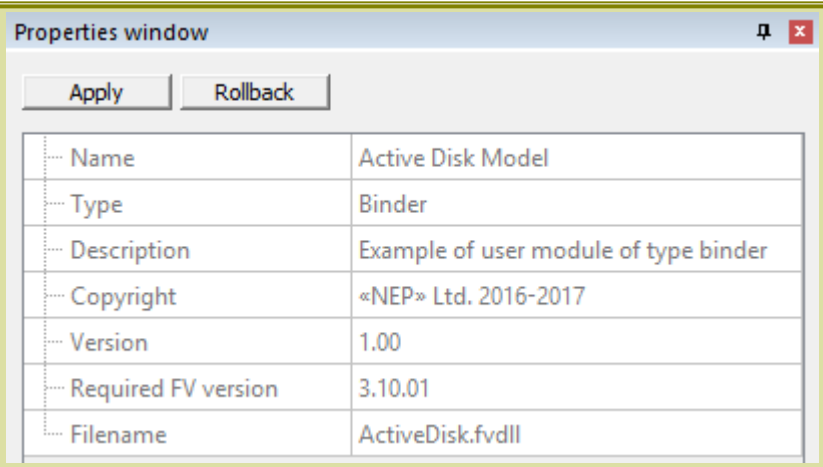
Context menu of the "User modules" folder



Context menu of the "User modules" folder

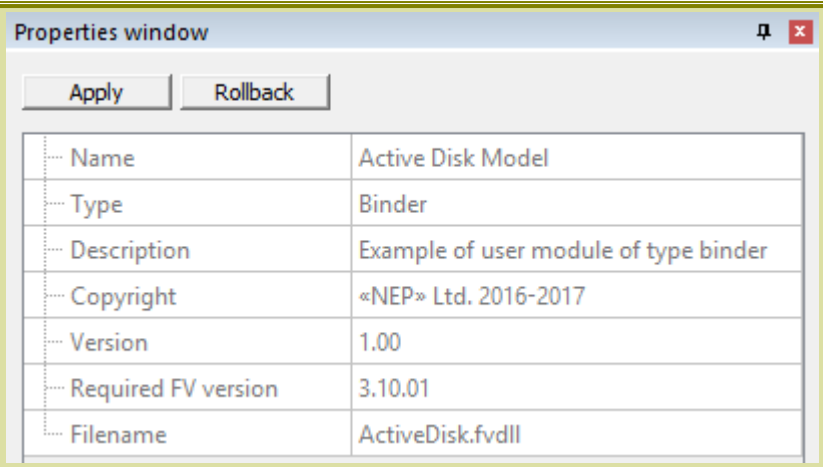
Menu item	Description
Load	This menu item loads one more user module into the project. A standard operating system's window for access to files will open. Specify there the file of the required user module (with `fvdll` extension). Then a dialog box will open containing information about the user module, which is to be loaded (its name, type, short description, copyright information, version number, requirements to **Solver**, supported computational platforms). Click **OK** to load the module, or click **Cancel** to refuse. Loading the user module includes copying its file into the project's directory.

The Properties window of a *User module* element



Parameters of a *User module* element

Parameter	Description
Name	Name of the user module
Type	Type of the user module. Possible options: • Evaluator • Binder
Description	Short description of the user module
Copyright	Copyright information related to the user module
Version	Version number of the user module

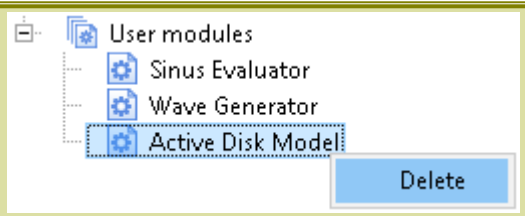


Parameters of a *User module* element

Parameter	Description
Required FV version	The lowest version of the *FlowVision*'s solver that is required to run this user module
Filename	Name of the module's file (including the `fvdll` extension), which has been copied to the project's directory

These parameters are read-only and cannot be changed in the **Properties** window.

Context menu of a *User module* element

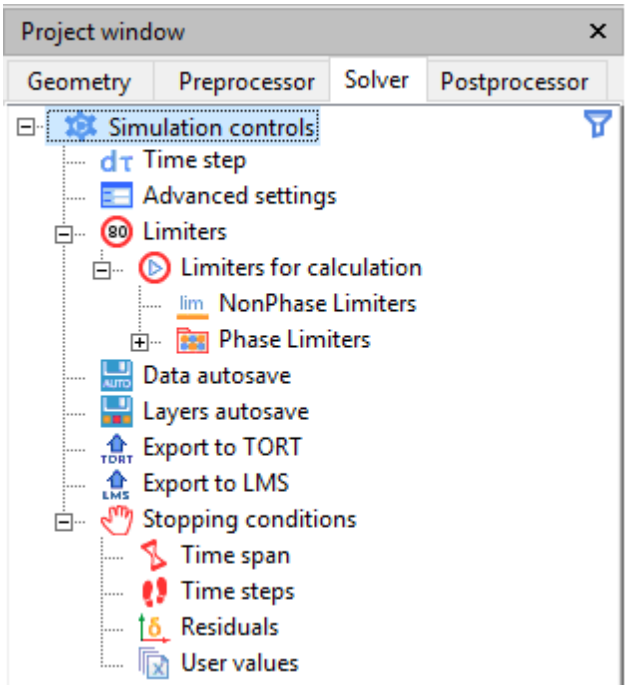


Context menu of a *User module* element

Menu item	Description
Delete	Exclude the selected user module from the list of user modules that are loaded into the project. This action does not delete the `fvdll`-file of the user module.

8.1.8.4 The Project window, tab «Solver»

The **Solver** tab displays the part of the project tree, which contains control parameters of the computation.

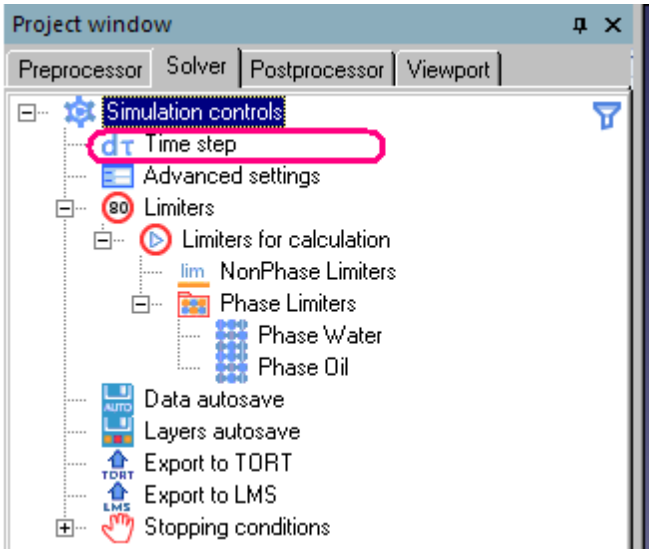


The **Project** window, the **Solver** tab

Elements and folders in the project tree in the tab **Solver** are grouped into the following folders and items within the folder **Simulation controls**:

- element **The time step**
- element **Advanced settings**
- folder **Limiters**
- element **Data autosave**
- element **Layers autosave**
- element **Export to TORT**
- element **Export to LMS**
- folder **Stopping conditions**

8.1.8.4.1 Element «Time step»

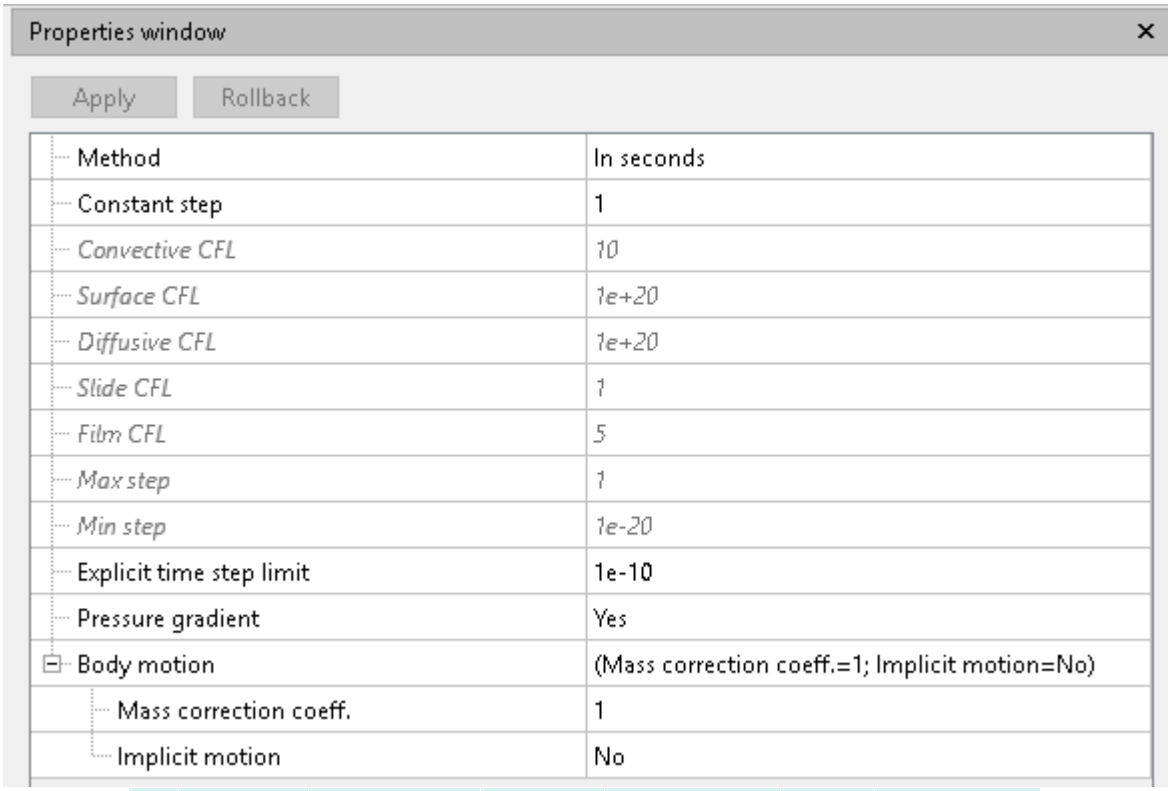


The **Time step** element is used to display and set the calculation's time step.

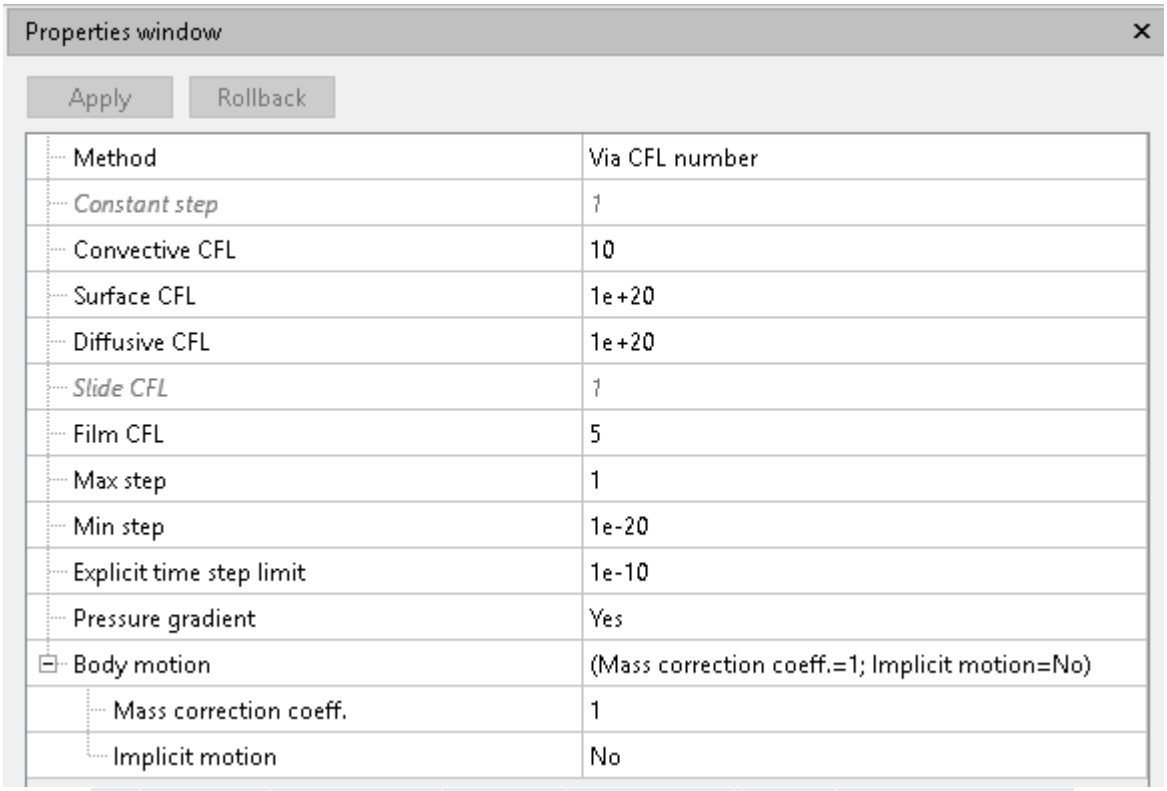
Step time can be set in one of two ways:

- *constant* (when **Method = In seconds**, the step size is defined by the **Constant step**)

- *depending on the given by the Courant-Friedrichs-Lewy numbers and other parameters* (when **Method = Via CFL number**).



The **Properties** window of the **Time step** element when **Method = In seconds**



The **Properties** window of the **Time step** element when **Method = Via CFL number**

Parameters of the **Time step** element:

Parameter	Description
Method	The method of specifying the time step: • **In seconds**: the time step is specified as a fixed value • **Via CFL number**: the time step is specified by the Courant-Friedrichs-Lewy number (CFL)
Constant step*)	The value of the constant time step. This parameter is active when **Method = In seconds** is selected.
Convective CFL*)	The Courant-Friedrichs-Lewy number for convective transfer, CFL_{conv}. This parameter is active when **Method = Via CFL number** is selected.
Surface CFL*)	The Courant-Friedrichs-Lewy number for surface transfer, CFL_{surf}. This parameter is active when **Method = Via CFL number** is selected.
Diffusive CFL*)	The Courant-Friedrichs-Lewy number for diffusive transfer, CFL_{diff}. This parameter is active when **Method = Via CFL number** is selected.
Slide CFL*)	The Courant-Friedrichs-Lewy number on a **Sliding surface**, $CFL_{slide.}$ This parameter is active when **Method = Via CFL number** is selected and in the presence of a **Sliding surface**.
Film CFL*)	This is the Courant-Friedrichs-Lewy number for determining the time step for spreading the liquid film over a surface. When icing is simulated, this parameter determines the number of computational cycles that are required to obtain a quasi-stationary solution in simulating of spreading the film and its crystallization on a surface. This parameter is not available when **Crystallization=(none)** in properties of **Physical processes** of the dispersed phase of the **Particles** type. When **Film CFL** is zero, the dispersed phase crystallization's program block is inactive but the program makes calculations of the volume source of the dispersed phase on the solid surface. This is useful at the preliminary computation of the project to make adaptation of the grid in the area where drops fell out. Values above **1** define the number of computational circles of the quasi-stationary process of "substance fell-out – spreading the film – crystallization of the substance" with explicit step of convective mass transfer in the film. Such values are recommended for use only for simulating icing of aircraft or other closed contours. When icing of an aircraft is simulated, **Film CFL** is recommended to be set in the range from **3** to **5**. When the film's spreading without icing is simulated, we recommend you to specify value of this parameter as ≤**1** and also specify **Multiphase D > Film step is limited by task step = No** in advanced settings of Solver. The default value is **5**.
Max step*)	Limitation on the maximal time step (t_{max}), [s], when the step is specified **Via CFL number**. This parameter is active when **Method = Via CFL number** is selected.
Min step*)	Limitation on the minimal time step (t_{min}), [s], when the step is specified **Via CFL number**. This parameter is active when **Method = Via CFL number** is selected. In older versions of the program, there was a parameter with the same name, which limited the minimum explicit time step at which computation is stopped. Now this parameter is renamed as **Explicit time step limit** (see below).
Explicit time step limit	Limitation on the minimum explicit time step at which computation is stopped.

Parameter	Description
	If the solution of the equations does not converge, the explicit time step can take very small values, which is a sign of the need to stop the computation. Application parameter **Explicit time step limit** is convenient when you run **Solver** in the batch mode. In older versions this parameter was named as **Min. step**.
Pressure gradient	This parameter specifies whether the pressure gradient is taken into account in calculation of the explicit time step. Possible values: **Yes** | **No**. Use of the default **Pressure gradient = Yes** value is recommended. Specify **Pressure gradient = No** for very viscous flows only.
Body motion > Mass correction coeff.	Mass correction coefficient, from **0** (no correction) to **1** (maximum correction)
Body motion > Implicit motion	This parameter specifies whether implicit calculation of moving body motion enabled.

*) The parameters can be specified either as constants or formulae (using Formula editor) or tables (using Table Editor). Specify step formula or table allows, for example,start a task, first with a coarse step, and then, after a specified formula or a table of time, the program will reduce the time step.

By default, **Method = Via CFL number** and **Convective CFL = 1**.

If you wish actually do not specify an upper limit to time step by some parameter, specify this parameter's value as a very large number, for example 10^{20}.

The calculation of the time step by means of numbers CFL

When choosing a method of calculating the step by the CFL (Courant-Friedrichs-Lewy) number, a step is calculated by the following formula:

$$\tau = \min \{\max [\min (\tau_{conv}, \tau_{diff}, \tau_{surf}, \tau_{slide}, \tau_{film}), \tau_{min}], \tau_{max}\}$$

where:

τ_{conv} is the convective time step

τ_{diff} is the diffusion time step

τ_{surf} is the surface time step

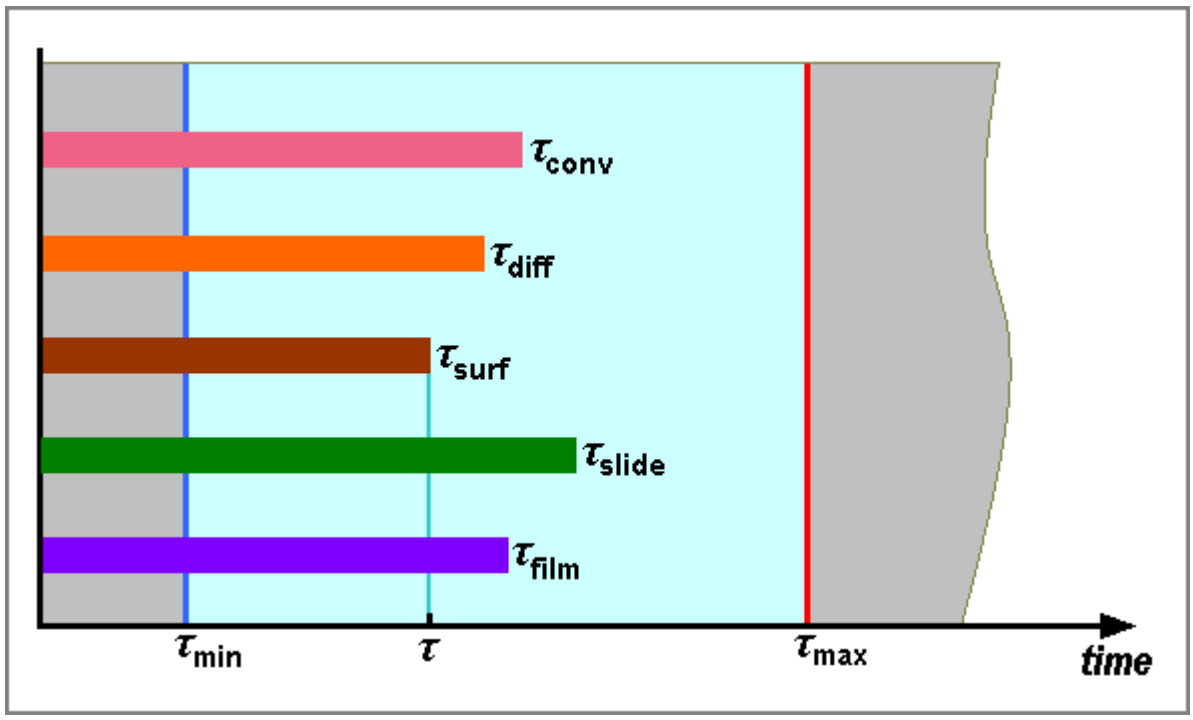
τ_{slide} is the time step along the sliding surface (using the sliding surface)

τ_{film} is the time step of the dispersed phase crystallization (when this process is simulated)

τ_{max} is the the maximum time step

τ_{min} is the the minimum time step

i.e. it is the minimum of variables $\tau_{conv}, \tau_{diff}, \tau_{surf}, \tau_{slide}, \tau_{film}$, shifted within the range $[\tau_{min} \ldots \tau_{max}]$, if this minimum goes out this range.



The time step τ is chosen as the minimum number of τ_{conv}, τ_{diff}, τ_{surf}, τ_{slide}, τ_{film} and is limited to the range [τ_{min}... τ_{max}]

Values of τ_{conv}, τ_{diff}, τ_{surf}, τ_{slide}, τ_{film} are obtained by multiplying the user-defined CFL (Courant-Friedrichs-Lewy) numbers – CFL_{conv}, CFL_{diff}, CFL_{surf}, CFL_{slide}, CFL_{film} – to their corresponding explicit time steps $\tau_{expl,\ conv}$, $\tau_{expl,diff}$, $\tau_{expl,surf}$, $\tau_{expl,slide}$.

When **Multiphase C** > **Use VOF source for time step = Yes** is set in the advanced settings of **Solver**, the formula for calculating τ_{surf} includes also $\tau_{expl,src}$, the explicit surface time step determined by action of the source Q_{VOF} of the variable **VOF** in Eq. (PhTr.1).

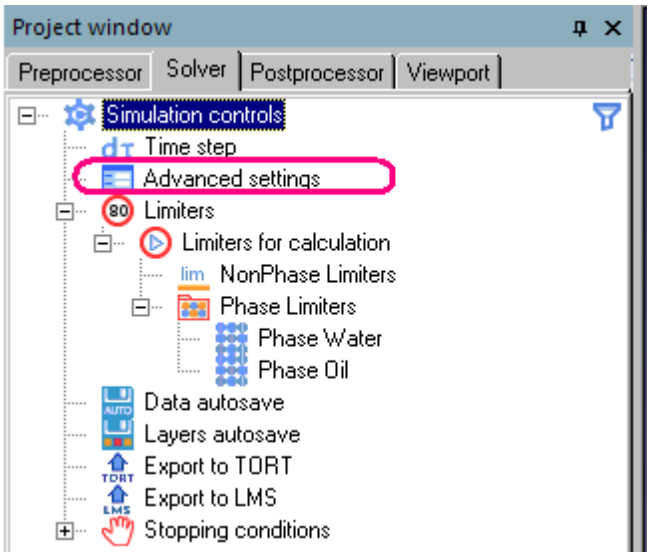
See formulae in the section Time step.

Recommendations for use the CFL numbers:

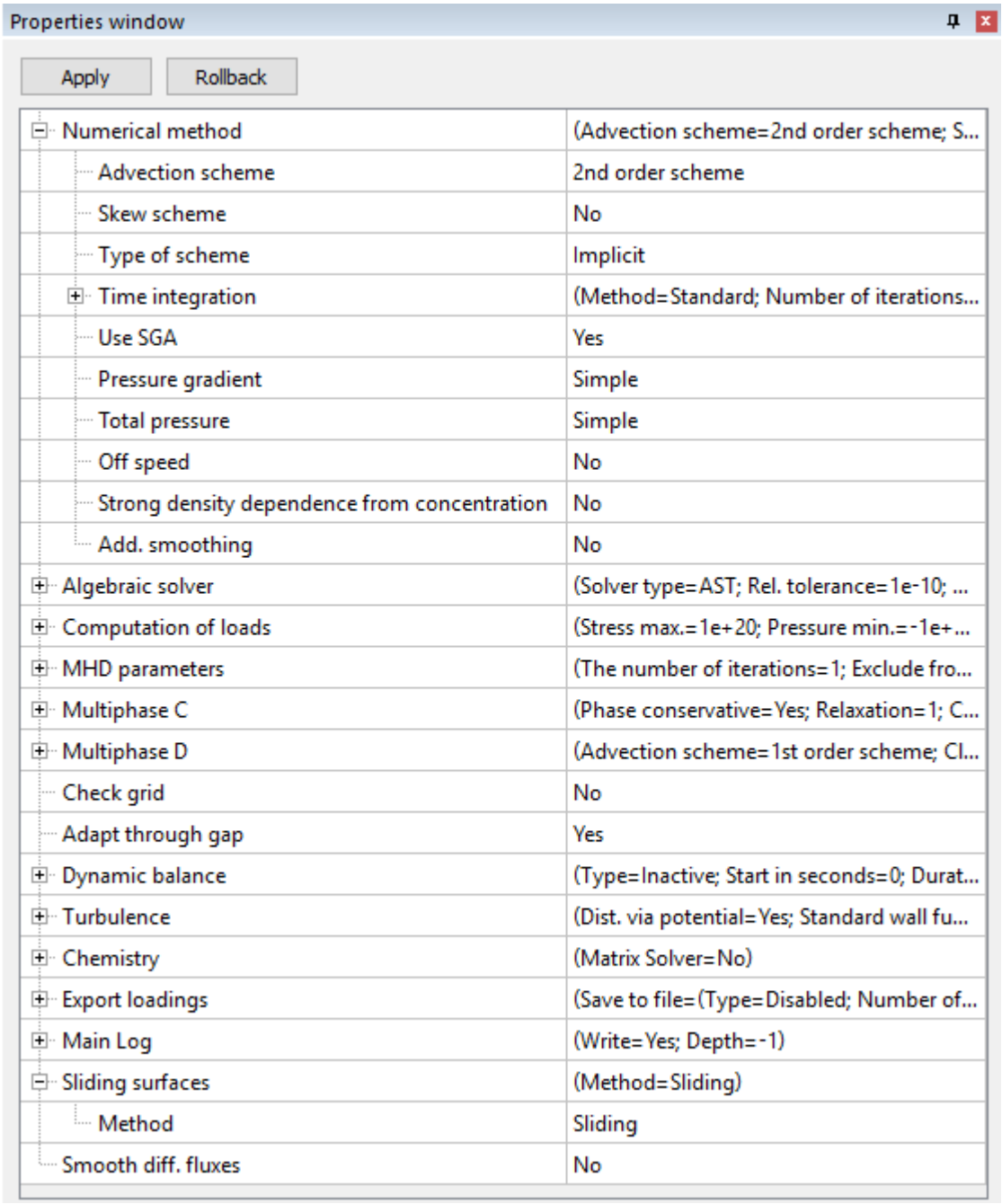
1. CFL_{conv} recommended for use in applications where the determining process is the convective transfer of substances, energy and momentum (subsonic, transonic and supersonic gas flow, unsteady flow of fluid, etc.)
2. CFL_{surf} is to be used in applications where the determining process is the movement of the free surface and moving bodies.
3. CFL_{diff} recommended for use in applications where the determining process is the diffusive transfer of substances, momentum and energy (movement of a very viscous liquid).
4. In problems with low Reynolds numbers (Re≤1), for example, when simulating a flow of a viscous liquid, it is necessary to limit the time step by the diffusion Courant-Friedrichs-Lewy number (specifying CFL_{diff}=1), because otherwise the splitting method for solving the Navier-Stokes equations does not work well and the solution becomes unphysical.
5. When using a sliding surface we recommend to use CFL_{slide}.

See also: Time step.

8.1.8.4.2 Element «Advanced settings» (advanced settings of Solver)



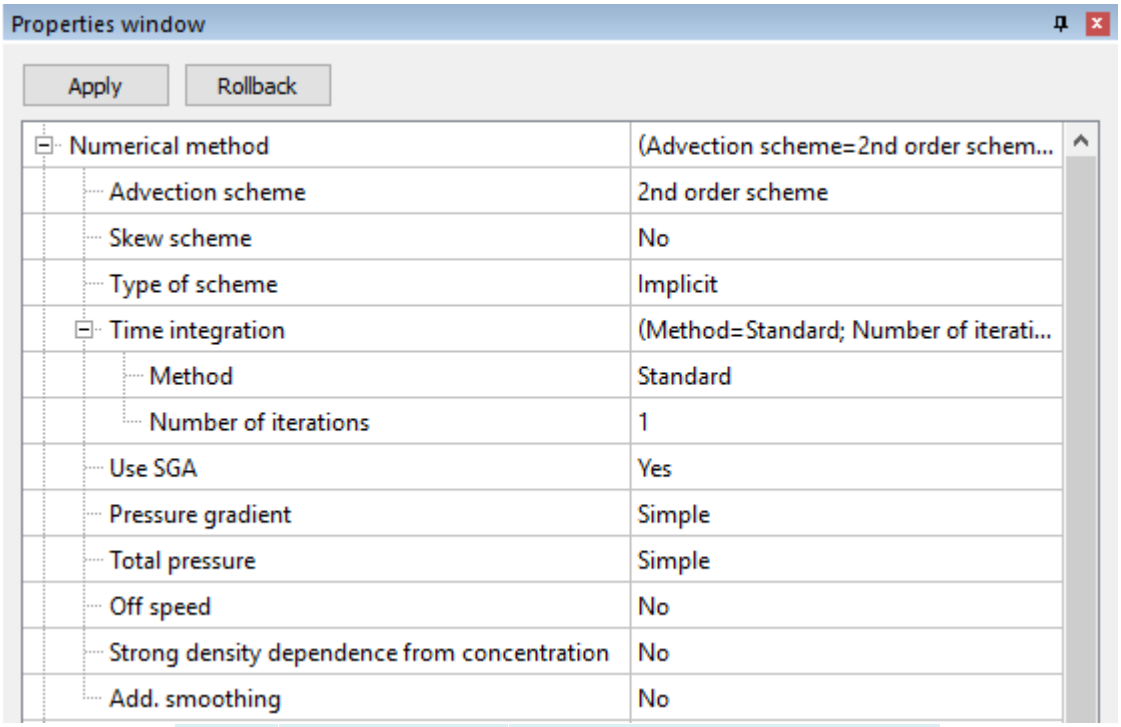
The **Advanced settings** element is designed to display and/or specify a large number of different calculation's parameters (referred as *advanced **Solver**'s settings*). These parameters, grouped in several groups, are displayed and specified in the **Properties** window.



The **Properties** window of the **Advanced settings** element (parameters are grouped in several groups)

Parameters of the numerical method

The parameters of the numerical method are specified and displayed in the **Properties** window, in the group of settings **Numerical method**:



Element **Advanced settings**, parameters of the numerical method

Parameters of the numerical method:

Parameter	Description
Advection scheme	Approximation scheme for convective terms in the integrated equations: • **1st order scheme**: defines use of the first-order accuracy of approximation of convective terms in transfer equations. • **2nd order scheme**: defines use of the second-order accuracy of approximation of convective terms in transfer equations.
Skew scheme	Use of the **Skew scheme** functionality. Possible options are: **Yes** | **No**. The **Skew scheme** functionality assumes calculation of convective flows of the target value (either scalar or vector one) on faces of a cell with taking into account the mass flow of the fluid through edges, which limit the selected face of the cell. Therefore, during calculation of convective flows, not only the "face-adjacent" cells but also the "edge-adjacent" ones are taken into account at the current time step. Use of the skew scheme essentially reduces the scheme diffusion for flows that substantially deviate from directions of Cartesian axes (that are particularly vortex-like flows). ⚠ The **Skew scheme** is in the process of development and was enabled in the *FlowVision* version *3.10.02* for testing in industrial simulations. Please, inform the *FlowVision* technical support service about any issues that you encounter when using the **Skew scheme** functionality.
Type of scheme	Type of computational scheme. Possible options are: **Implicit** *) | **Explicit**. See details in the section Parameters of the numerical method.
Time integration > Method	Possible options are:**Standard** | **Steady-state**. See details in the section Parameters of the numerical method. For most purposes, it is recommended to set **Method = Standard**. For the simulation of steady-state processes in solids you can set **Method = Steady-state**. This speeds up the convergence to the steady-state solution.

Parameter	Description
	For a gas that follow the ideal gas law, the **Steady-state** method works the same way as the **Standard** one.  In problems with conjugate heat transfer between a solid and a fluid media nonconvergence of the computation can occur, when a small time step is defined and the fluid is not moving and the **Steady-state** value of this parameter.
Time integration > Number of iterations	Number of successive cycles of solving the equations of the mathematical model within one time step: if **Number of iterations = 1**, then the equationsare are solved once, if **Number of iterations = 2**, then the same equations are solved twice, etc. Each time before solving the equations, their coefficients, which depend on the solution, are recalculated.
Use SGA	Use of the higher order of approximation (SGA, side gradient approximation) for diffusion terms of the transfer equations: • **Yes**: use high approximation order for diffusion terms of the transfer equations • **No**: use low approximation order for diffusion terms of the transfer equations
Pressure gradient	Method of approximation of the pressure gradient in cells adjacent to the boundaries on which static pressure is not specified explicitly. It is applied when on a surface a boundary condition is specified defining the value of velocity. This setting is not applied on boundary conditions **Free Outlet**, **Non-reflecting**, and when **Velocity** is specified as **Velocity with pressure**, **Normal velocity with pressure**, or **Total pressure**. Possible options are: • **Simple**: The static pressure on the boundary equals to its value in the cell's center (so the pressure gradient is assumed to be zero). This method makes an inaccuracy in calculation of the pressure gradient along the normal to the boundary condition's surface, but it is very stable and can be recommended for simulations with poor convergence at transient phases of the computation. • **With interpolation**: The static pressure at the surface is calculated with taking into account the local distribution of the pressure (it is linearly extrapolated from values in nearby cells). • **With velocity consideration**: Taking into account a loss of the momentum caused by zeroizing the normal component of the velocity (the pressure gradient is calculated with taking into account the local distribution of the velocity). The normal component of the velocity causes change of the pressure on the surface (the Bernoulli's relation), that causes the pressure gradient in the normal direction to the surface. This is the most advanced and accurate method of taking the pressure gradient into account, but this method might be unstable in transient processes.We recommend to use this method when advanced accuracy is required.
Total pressure	This parameter defines how the program calculates the total pressure. Possible options are: • **Simple**: The total pressure is calculated using known analytical formulae. Generally, this method is less accurate, but spends less computational resources. • **Exact**: The total pressure is calculated using integrating. Generally, this method is more accurate, but spends more computational resources. See details in the subsection "*Calculation of total pressure and temperature for gas*" in the section Theory > Substance properties > Gas.
Off speed	This parameter allows you to turn off computation of velocities (and calculate the pressure only). Turning the computation of velocities is useful when you don't need to calculate velocities and the pressure only is interesting to you, for example, for solving steady-state FSI problems when load is to be transmitted

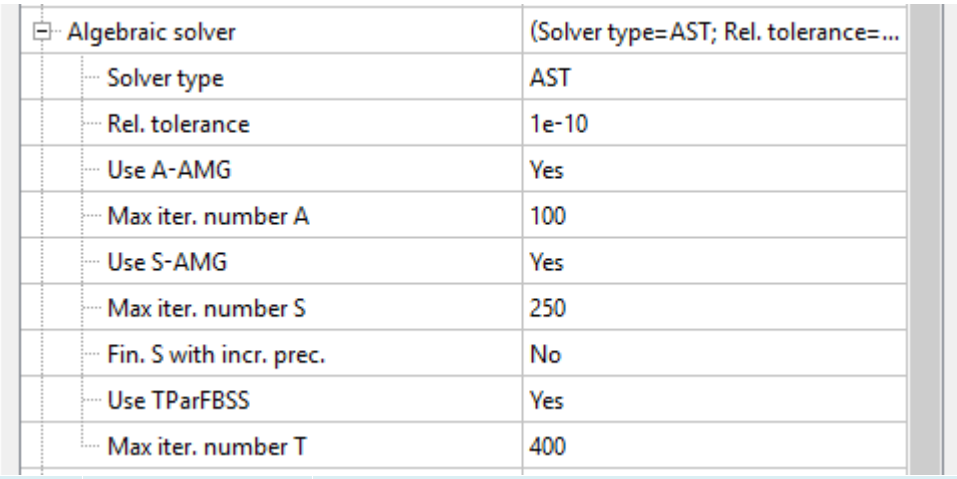
Parameter	Description
	on a complex and changing surface. Also this can be useful for simulating processes with constant volumes of fluid (isochoric process). Possible options: **Yes** | **No**.
Strong density dependence from concentration	Turn this setting on, if density in you project strongly depends on concentration (when density changes more then 20% depending on concentration). Possible options: **Yes** | **No**.
Add. smoothing	Applying additional smoothing. Possible options: **Yes** | **No**.

*) The **Implicit** scheme allows computation of supersonic flows (M>1) with large (CFL>10) time step. Hypersonic (M>10) flows calculated using the **2nd order scheme** in some situations can cause errors; in such cases it is recommended to use **1st order scheme**.

See also: section Parameters of the numerical method.

Parameters for solving the system of algebraic equations

Parameters for solving the system of algebraic equations are set and displayed in the **Properties** window, in the **Algebraic solver** group of parameters.



Element **Advanced settings**, parameters for solving the system of algebraic equations

Parameters for solving the system of algebraic equations:

Parameter	Description
Solver type	Type of the algebraic solver. Possible options are: • **TParFBSS** is the *FlowVision*'s algebraic solver, which has been developed based on incomplete LU factorization and iteration scheme of Krylov subspace. • **AST** is the new *FlowVision*'s technology that is a combination of the known method *Aggregation AMG* (**A**), the selective multi-grid method *Selective AMG* (**S**) and the *TParFBSS* (**T**) algebraic solver. Performance of these computation methods is generally graded in the descending order as **A-S-T**, while their robustness is graded as **T-S-A**. At the beginning the **AST** algebraic solver tries to ensure solution with the specified **Rel. tolerance** (see below) using **A**, and if difficulties encounter, it switches to **S**, and then, if it is required, the solver switches to the most robust **T**. Setting **Solver type = AST** generally speeds up the simulation.
Rel. tolerance	Accuracy of the equations's convergence, at which the process of solving the equation is stopped at the current iteration. The current value of the algebraic residual is displayed in the **Monitor** window in the **Status** tab in the **Algebraic residual** column.
Use A-AMG*)	Use the *Aggregation AMG* method (might be along with others) when **Solver type = AST**. Possible options are: **Yes** | **No**.

Parameter	Description
Max iter. number A[*]	The maximum number of iterations per one time step for the *Aggregation AMG* method. The default value is **100**.
Use S-AMG[*]	Use the *Selective AMG* method (might be along with others) when **Solver type = AST**. Possible options are: **Yes** | **No**.
Max iter. number S[*]	The maximum number of iterations per one time step for the *Selective AMG* method. The default value is **250**.
Fin. S with incr. prec.[*]	This setting (*"Finalize S-AMG with increased precision"*), which is applied when required, provides convergence of the method *Selective AMG* at the expense of time consumption. When **Fin. S with incr. prec. = Yes**, then, if the applied standard *Selective AMG* method does not converge, the program will apply modified *Selective AMG* with *BiCGSTAB* that uses 128-bit floating-point numbers. If this method also does not converge, the program will apply *Selective AMG* with *SOFGMRES* that uses 128-bit floating-point numbers. Possible options are: **Yes** | **No**.
Use TParFBSS[*]	Apply the *TParFBSS* method (might be along with others) when **Solver type = AST**. Possible options are: **Yes** | **No**. Even when **Use TParFBSS = No** is set, the program might use *TParFBSS* in specific cases to provide convergence of the equations.
Max iter. number T	The maximum number of iterations per one time step for the *TParFBSS* method. The default value is **400**.

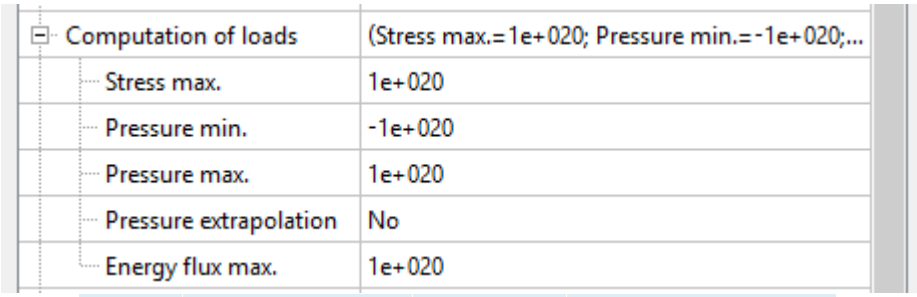
[*] These parameters are available when **Solver type = AST**.

The current number of iterations of the applied method (*Aggregation AMG*, *Selective AMG* or *TParFBSS*) is displayed in the **Monitor** window in the **Status** tab in the **Iteration** column.

Settings for calculation of pressure on a surface(group of parameters "Computation of loads")

Parameters in the group **Computation of loads** display and specify settings, which influence on calculation of hydrodynamic forces acting on a surface. These parameters influence on, for example, calculation of:

- loads that are transferred into finite-element analysis (FEA) software
- motion of **Moving bodies**
- forces in **Characteristics**

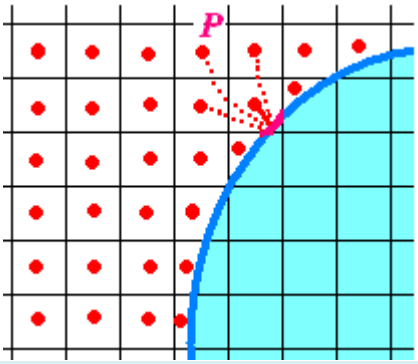


Element **Advanced settings**, parameters **Computation of loads**

Parameters "Computation of loads"	
Parameter	**Description**
Stress max.	Maximal shear stress, [Pa]. This parameter allows the program to limit action of friction forces when the program is computing loads on bodies.
Pressure min.	Limiters on values of relative[*] **Pressure** (limits of a range), which are used to calculate hydrodynamic forces acting on a surface. When the **Pressure** goes out of the range's limits, the calculation uses the value of the appropriate parameter (either **Pressure min.** or **Pressure max.**).

Parameters "Computation of loads"	
Parameter	**Description**
Pressure max.	These limiters are specified in *relative* values (see section Reference parameters, absolute and relative variables for details).
Pressure extrapolation	This parameter toggles use of extrapolation of **Pressure** values from nearby cells. We recommend to enable this setting when you use a coarse computational grid to receive more accurate values for forces on the surface. **Pressure extrapolation** is used for calculating the forces on the current step only and has no influence on calculation of the next step. You can enable **Pressure extrapolation** at any step of the computation. To recalculate the forces, the program has to do the next iteration. A substantial differ between computational results when you use or do not use the extrapolation means that the grid is too coarse near the surface. Disabling the **Pressure extrapolation** does not give an advantage in performance. Possible options: **Yes** | **No**.
Energy flux max.	Maximum flux of energy that can be transferred into a FEA software, [W/m^2]

*) See section Reference parameters, absolute and relative values.



When **Pressure extrapolation** is enabled, the calculation of the pressure acting on the surface the program takes data not only from the cell, which contacts the surface, but also from nearby cells.

Parameters for solving Maxwell equations ("MHD parameters")

MHD parameters	(The number of iter...
The number of iterations	1
Exclude from the calculation the equation for the electric potential	No
Correction factor	0

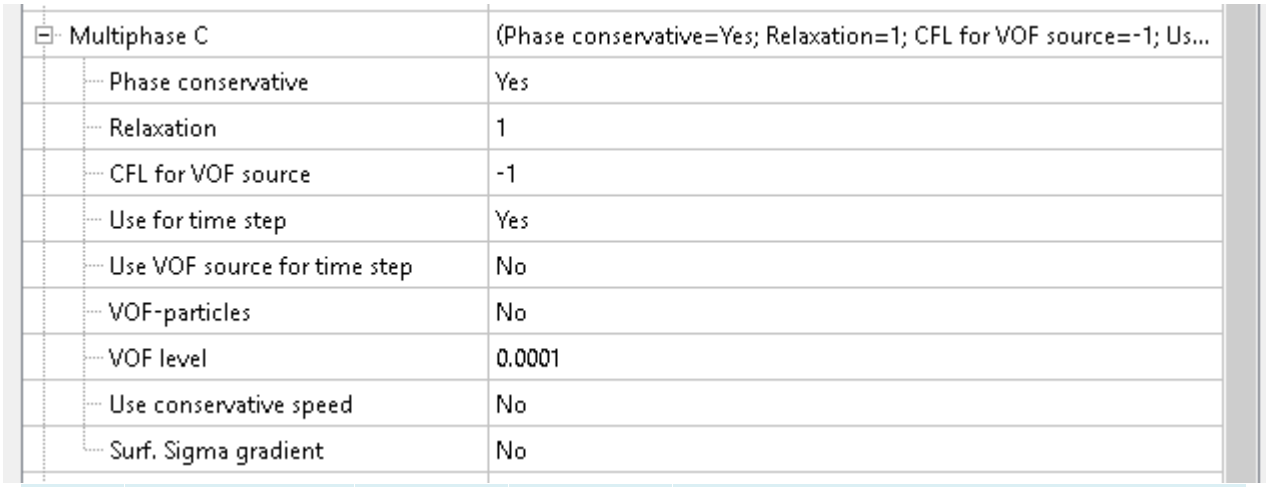
Element **Advanced settings**, parameters for solving Maxwell equations

Parameters for solving Maxwell equations	
Parameter	**Description**
The number of iterations	The number of iterations for simulating the electromagnetic process in one time step
Exclude from the calculation the equation for the electric potential	When this parameter is **Yes**, the equation for electric potential $\boldsymbol{E} = -\nabla\varphi$ (EMHD.3, EMHD.10) is not calculated and existing values of electric potential are not changing. Possible options are: **Yes** | **No**.
Correction factor	The $deffactor$ value in the *deferred correction method* for the electrical potential equation.

Parameters for solving Maxwell equations	
Parameter	**Description**
	Only values ≥ **0** are permissible. When **Correction factor > 0**, the electrical potential equation is calculated with correction. See details in the section Theory > Physical processes > Electromagnetohydrodynamics > Equations, subsection *"Deferred correction method"*.

"Multiphase C", parameters of the VOF solver, which simulates transfer of continuous phases

Parameters of the VOF solver, which simulates transfer of continuous phases, are set in properties of the **Advanced settings** element in the **Multiphase C** group of parameters.

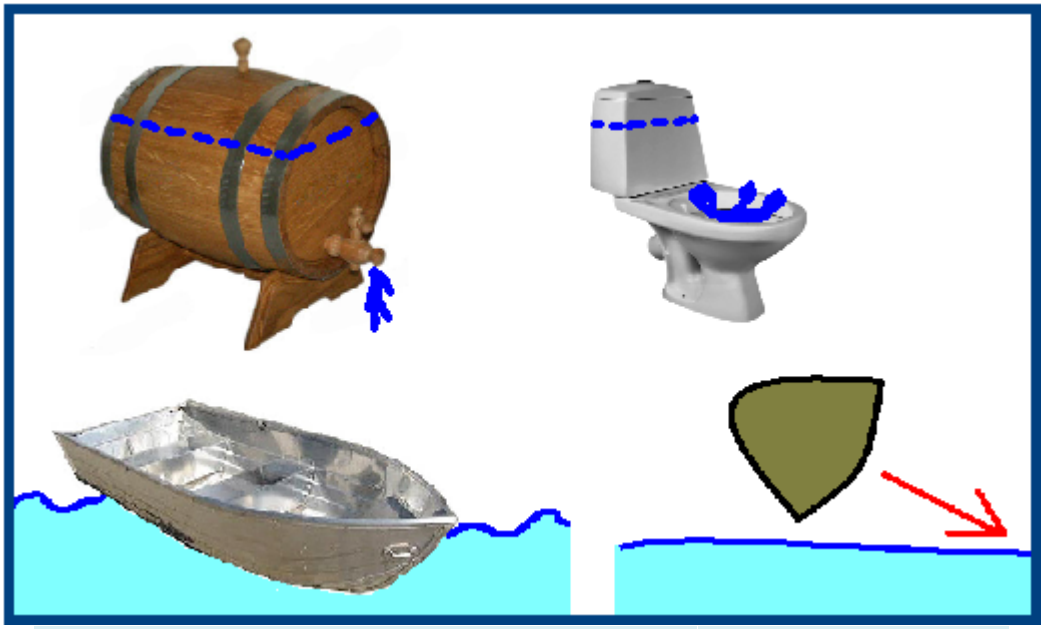


Element **Advanced settings**, parameters **Multiphase C** for configuring the simulation of phase transfer for continuous phases

Phase transfer for continuous phases	
Parameter	**Description**
Phase conservative	This parameter provides calculation of the law of conservation of mass for VOF. Possible options: **Yes** | **No**. When the conservation is enabled, the following numerical effect occurs: if change of the relative cell volume in one time step is less than about 1/100 of the cell volume, then change of the cell's volume will be canceled (the change will be considered infinitely small and insignificant). This can stop movement of the free surface in areas with coarse computational grid. We recommend: • turn this parameter on (set it to **Yes**) in simulations where it is important to preserve the exact mass of the liquid (for example, in closed system or in tasks where the liquid's level is monitored). • turn it off (set it to **No**) in situations when the program calculates the phase transfer through small cells and there are large cells elsewhere on the interphase surface (for example, leaking of liquid through a small hole in a large tank, where cells near the surface of the liquid in the tank are large, and near the hole the cells are small). Also it makes no sense to turn this parameter on when the program simulates external flows (for example, flow of water around vessels, or drop of a physical body into the water (drop-tests)).
Relaxation	The time step, which is used to calculate the phase transfer, is multiplied on this parameter. If **Relaxation=0**, then the transfer of the phase boundary doesn't happens and also a condition of impermeability of the liquid through the inter-phase surface is set.

Phase transfer for continuous phases	
Parameter	**Description**
CFL for VOF source	The CFL (Courant-Friedrichs-Lewy) number for the phase transfer's program block. It is calculated based on the source of the solid phase formed due to the crystallization (from the substance of the dispersed phase). This CFL value determines the step of motion of the inter-phase surface due to action of the source Q_{VOF} in the equation (PhTr.1) for the **VOF** variable. When this parameter is zero or negative (**CFL for VOF source** ≤ **0**), the time step for the phase transfer will be calculated based on the value of the **Relaxation** parameter. Positive value of this parameter (**CFL for VOF source** > **0**) determines the fraction, at which volume of the solid phase can increase/decrease in any cell per one (common for the whole simulation)time step. Thus, at one iteration, volume of the solid phase in any cell cannot increase at one common step more then (**CFL for VOF source**)×(volume of the cell). When icing of an aircraft is simulated, it is recommended to set the **CFL for VOF source** in the range from **0.1** to **0.4**.
Use for time step	This parameter specifies that the time step is calculated via surface CFL. Possible options: **Yes** | **No**. If **Use for time step = No**, then the time step is calculated without taking into account the speed of the phase boundary.
Use VOF source for time step	This parameter specifies if the inter-phase surface's motion depends on the source Q_{VOF} in the equation (PhTr.1) for the variable **VOF**. Possible options: • **Yes** – the problem's time step is calculated *without* taking into account the motion of the inter-phase surface, caused by Q_{VOF}. In this case the value of the **CFL for VOF source** parameter is ignored and motion of the free surface is done with resulting time step with value of the **Relaxation** parameter taken into account. • **No** – the problem's time step is calculated *with* taking into account the motion of the inter-phase surface, caused by Q_{VOF}. In this case motion of the free surface, caused due to action of the source, is simulated with taking into account the **CFL for VOF source** parameter. See details and formulae in the section Time step. When icing of aircraft is simulated, you are to keep the default value of this parameter, **Use VOF source for time step = No**.
VOF-particles	This parameter allows you to turn on or off simulation of VOF-particles. Possible options: **Yes** | **No**.
VOF level	The cut level for the value of **VOF**. When **VOF** in a cell differs from **0** or **1** less then the value of this parameter, then the value **0** or **1** respectively is compulsory assigned to **VOF**.
Use conservative speed	The program will use **Conservative velocity**, which is based on speed of fluid on cell faces, in transport equation for **VOF**. Possible options: **Yes** | **No**.
Surf. Sigma gradient	This parameter specifies the method of calculating the gradient of the surface tension when thermocapillary effects or similar effects, caused by non-homogeneity of distribution of the gradient of the surface tension along the inter-phase surface, are simulated. Possible options: • **Yes** – the gradient of the surface tension is calculated with use of derivatives strictly along the inter-phase surface. Complex high-curvature shape of the

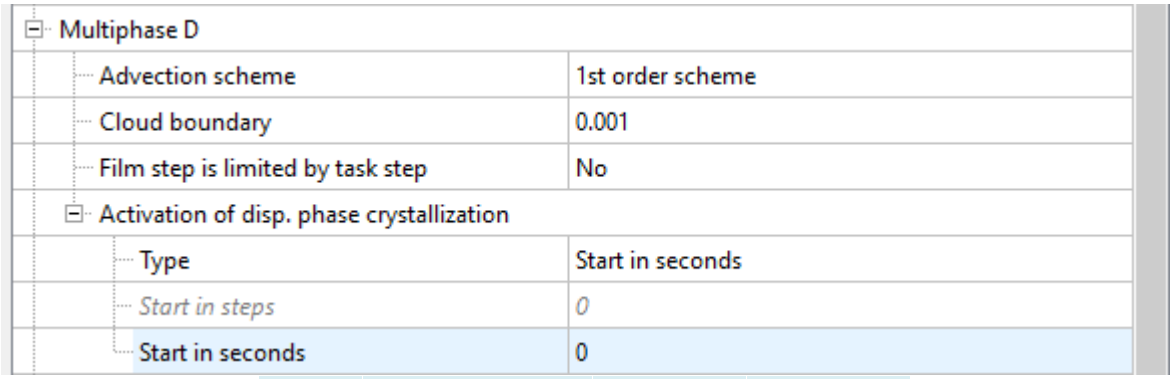
Phase transfer for continuous phases	
Parameter	**Description**
	inter-phase surface can cause substantial oscillations of the solution on the surface. • **No** – the gradient of the surface tension is calculated with use of derivatives in the volume of those phase, from which side the appropriate effect is simulated. In this case the solution along the inter-phase surface will be smoother but less accurate comparing the solution in the case when you set **Surf. Sigma gradient = Yes**. Your choice here should depend on a priori knowledge about resulting shape of the inter-phase surface and requirements to accuracy of the solution. When **Surf. Sigma gradient = No**, the spatial derivatives of the computational variables, which have an influence on the surface tension coefficient, are projected on the inter-phase surface.



Examples of problems in which it is recommended to set **Phase conservative = No**

"Multiphase D", parameters of dispersed solver (simulating the transfer of dispersed phases)

Parameters for multiphase interaction for dispersion medium are set in properties of the **Advanced settings** element in the **Multiphase D** group of parameters.



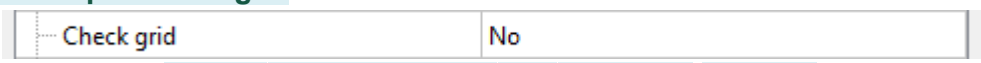
Element **Advanced settings**, parameters **Multiphase D**

Parameters for multiphase dispersion medium	
Parameter	**Description**
Advection scheme	Parameter of the **Dispersed solver**, which simulates phase transfer of dispersed **Phases**. Possible options are: **1st order scheme** | **2nd order scheme**.
Cloud boundary	The minimum value of the **Dispersed Phase**'s relative volume, at which the program do calculations of sources of mass and energy transfer (exchange). The default value is 10^{-3}. In cells where the volume of the **Dispersed Phase** is less then the value of this parameter, the dispersed variables are not taken into account: • in visualization layers • when calculating **Characteristics** • when calculating **User variables**
Film step is limited by task step	This parameter specifies how the time step for simulating the film's motion (the film's spreading) will be selected. Possible options are: • **Yes** – motion of the film will be simulated with the task's time step τ. This option allows the program to obtain a non-steady solution for the film's motion that is synchronized with the problem's time. • **No** – motion of the film will be simulated with its own explicit time step, $\tau_{expl,film}$, even when $\tau_{expl,film}>\tau$. Within a task's time step τ, computation of the film's motion will be done in several iterations, the number of which is set by the **Film CFL** parameter. This option is used to find the steady-state solution. The film's motion shown in this case can differ from the motion in the real non-steady process. See also sections Time step and Element «Time step».
Activation of disp. phase crystallization	The group of parameters **Activation of disp. phase crystallization** allows you to specify the moment of activating the simulation of crystallization the dispersed phase (or refuse starting this simulation). This group of parameters is only available when **Crystallization≠(none)** is set in properties of **Physical processes** of the dispersed phase of the **Particles** type. Some crystallization models include simulating of spreading the liquid film over the surface. When crystallization is not being simulated, the film's spreading is also not being simulated.
Activation of disp. phase crystallization > Type	The method how crystallization of the dispersed phase will be activated (if yes). Possible options are: • **Disabled** – crystallization of the dispersed phase will not be activated until you change the value of the **Type** parameter to either **Start in seconds** or **Start in steps**. • **Start in seconds** – crystallization of the dispersed phase will be activated at the specified time moment. • **Start in steps** – crystallization of the dispersed phase will be activated at the specified step
Activation of disp. phase crystallization > Start in steps	The step, at which crystallization of the dispersed phase will be activated. The default value is **0**. Parameters **Film CFL** (in properties of the **Time step** element) and **CFL for VOF source** (in the **Multiphase C** group of parameters, see subsection *"Multiphase C", parameters of the VOF solver, which simulates transfer of continuous phases* above) can be set arbitrary at starting the project; their values will be ignored until simulation of the icing is activated. This parameter is available when **Activation of disp. phase crystallization > Type = Start in steps**.
Activation of disp. phase crystallization > Start in seconds	The time moment, [s], at which crystallization of the dispersed phase will be activated. The default value is **0**. Parameters **Film CFL** (in properties of the **Time step** element) and **CFL for VOF source** (in the **Multiphase C** group of parameters, see subsection *"Multiphase C", parameters of the VOF solver, which*

Parameters for multiphase dispersion medium	
Parameter	**Description**
	simulates transfer of continuous phases above) can be set arbitrary at starting the project; their values will be ignored until simulation of the icing is activated. This parameter is available when **Activation of disp. phase crystallization > Type = Start in seconds**.

See also: Parameters used in the model of the disperses phase.

Checking the computational grid

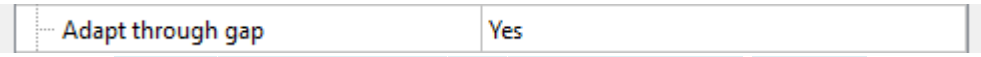


Element **Advanced settings**, the **Check grid** parameter

The **Check grid** parameter specifies necessity to verify the computational grid after each its change. Enabling this parameter can significantly slow down the computation.

Parameter	Description
Check grid	Verifying the correctness of building the computational grid after each its change. Possible options are: • **No**: verifying correctness of building the computational grid only after the initial grid building • **Yes**: verifying correctness of building the computational grid after each change of the grid See details in the section Validating the computational grid structure.

Adaptation of gap cells



Element **Advanced settings**, the **Adapt through gap** parameter

The **Adapt through gap** parameter specifies necessity of the same adaptation level for all cells in a gap.

Parameter	Description
Adapt through gap	Maintain or don't maintain the same level of cells within a gap. Possible options are: **Yes** | **No**. When **Adapt through gap = Yes**, then, if the gap contains cells with different adaptation levels, automatic adaptation of all cells in the the gap is applied up to the same (the maximal) adaptation level of all cells in the gap. Setting **Adapt through gap = Yes** provides correct operation of the gap model, which requires the same adaptation level in all cells. When solution in the gap don't influence substantially on the whole solution, then, to save the number of computational cells, you can disable the automatic adaptation in gaps (set **Adapt through gap = No**). Also it can be recommended to set **Adapt through gap = No** when the problem setting includes closely-spaced gaps of different widths so adaptation of a gap can cause refining in another gap so its cell will become non-gape ones.

Configuring the Dynamic Balancing

Parameters of the dynamic balancing are set and displayed in the **Properties** window, in the **Dynamic balance** group of settings.

Balancing of computational resources can significantly speed up multiprocessor computations due to reducing downtime of some processors.

Parallelizing of the calculation (geometric decomposition of the computational grid) using *MPI* assumes distribution the computational grid, on which systems of linear equations (SLE) are composed, among processors.

This process is automated and implemented in such a way that allows simultaneously minimize losses on the data exchange between processors and improve uniformity of processors' load.

Computing unbalance can be caused by various reasons, for example:

- it is not always possible to determine computational and non-computational cells before start of the computation
- different cells may require different amounts of computational load
- some cells might be active for not all combinations of simulated values

In all these cases, some of these processors complete their computations faster than others and then they are idle. This has a negative effect on the overall duration of the computation. Imbalance of loading of the processors is especially typical for simulations with local high-level adaptations or with moving bodies or with a free surface.

To improve balancing of the processor loads, you can enable the dynamic balancing. This feature allows the program to dynamically evaluate the load on different processors, and, based on this indicator, automatically reallocate the computational grid between the processors.

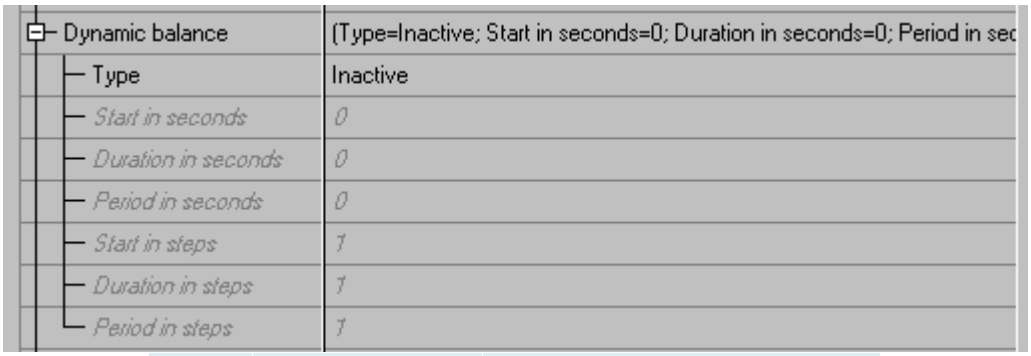
The process of reallocation of the computational grid requires significant computing resources, while the criterion for termination of the balancing is not clear. Therefore, when the balancing is enabled, it is necessary to specify the time moments when this process is turned on and off.

It should be noted that in problems where the computational grid changes significantly during the computation (sequentially changing adaptations, moving bodies, a free surface) it is sometimes better to enable the dynamic balancing periodically only.

In the case of gradual changing the computational grid using a adaptation it is reasonable to carry out the dynamic balancing only after the final computational grid is built.

Typically, successful balancing requires just a few (3-5) time steps. To accurately assess the time moment of achieving the most efficient load balancing among processors, you might examine the **sta**-file in the server part of the project (see section Format of sta files). The closer are the values for different processors in the column **CPU_Time**, the more uniform is the load on the processors.

For a single-processor mode, this group of settings is not used (values of these settings don't affect operation of the program).



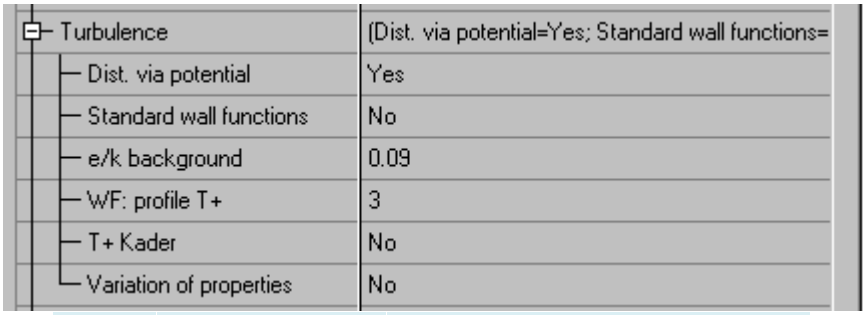
Element **Advanced settings**, parameters of dynamic balancing

Parameter	Description
Type	This option allows you to enable or disable the dynamic balancing of computational grid over processors. Dynamic balance allows the program to distribute the load among processors during the computation and speed up the project's computation. Possible options are: • **Inactive** means that the dynamic balancing is disabled. • **Only once by time** means that the dynamic balancing is active from the moment of **Start in seconds** time moment during the **Duration in seconds** period. • **Only once by step** means that the dynamic balancing is active from the moment of **Start in steps** during the **Duration in steps** period. • **Permanent** means that the dynamic balancing is constantly active. • **Repetitive by time** means that the dynamic balancing is activated periodically. The first activation of the dynamic balancing occurs at the moment of **Start in seconds**, dynamic balancing remains active during the **Duration in seconds**

Parameter	Description
	period. Activation of the balancing occurs with a periodicity defined by parameter **Period in seconds**, counted from the moment of **Start in seconds**. • **Repetitive by step** means that the dynamic balancing is active from the moment of **Start in steps** during the period of **Duration in steps**. This is repeated periodically each integer **Period in steps** measured starting from **Start in steps**. Regardless to the value of this parameter, the dynamic balancing is not turned at small disbalance of processes' load. It is done so, because the balancing is resource consuming and, at small disbalance, costs of the balancing exceed losses caused by the disbalance. The threshold of disbalance, below of which the dynamic balancing does not start, can vary in various releases of *FlowVision*.
Start in seconds	Moment of time, [s], when dynamic balancing is turned on: • for the first time when **Type = Repetitive by time** • only once when **Type = Only once by time**
Duration in seconds	Duration of the dynamic balancing when it starts once or periodically, [s]. (it is available when **Type** = **Only once by time | Repetitive by time**).
Period in seconds	Duration of the period, in which the dynamic balancing is on and off, [s]. This setting is accessible when **Type = Repetitive by time**.
Start in steps	Step(s) on which the dynamic balancing becomes active for the first time (when **Type = Repetitive by step**) or only once (when **Type = Only once by step**)
Duration in steps	The number of steps, during which the dynamic balancing is active. This parameter is used for single or periodic start(s), when **Type= Only once by step | Repetitive by step**.
Period in steps	Period of turning the dynamic balancing on/off, measured in steps. This setting is accessible when **Type = Repetitive by step**.

Parameters of the turbulence model

Parameters of the turbulence model set and displayed in the **Turbulence** group of parameters in the **Properties** window.



Element **Advanced settings**, parameters of the turbulence model

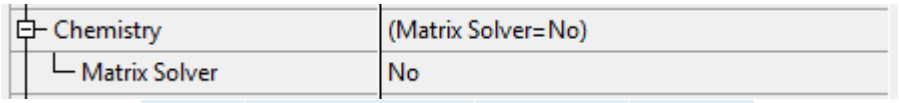
Parameters of the model of turbulence	
Parameter	**Description**
Dist via potential	Specifies that the distance from the wall in the wall cell is calculated using the potential of distance. Possible options are: • **No** - the distance from the wall to the center of the cell is equal to the geometric distance from the wall to the center. • **Yes** - the distance from the wall to the center of the cell is calculated using the potential
Standard wall functions	Use of standard near-wall functions. Possible options are: • **No** – use of the modified near-wall functions in the calculation (the **WFFV** model) • **Yes** – use of the standard near-wall functions in the calculation (the **WFS** model)

Parameters of the model of turbulence	
Parameter	**Description**
e/k background[*)]	Background frequency of turbulent fluctuations [s^{-1}]
WF: profile T+[*)]	One of the three implemented temperature profiles.
T+ Kader	Weighting laminar and turbulent T+ profiles according to Kader. Possible options are: **Yes** | **No**.
Variation of properties	Taking into account variation of fluid properties (density, viscosity) between the wall and the first cell center in wall functions. Possible options are: • **No** - is recommended to specify if fluid properties (density, viscosity) do not depend on the temperature or in simulations without heat transfer. • **Yes** - is recommended to specify for simulations with heat transfer when fluid properties (density, viscosity) depend on the temperature. Do not forget to specify the molar mass of the **Substance** (or of each of the used **Substances**).

[*)] Possible values, default values, recommendations and other details about these parameters see in section Theory> Physical processes> Turbulence> Parameters.

Group of parameters "Chemistry"

The **Chemistry** group of parameters contain settings for convection, diffusion and chemical reactions.



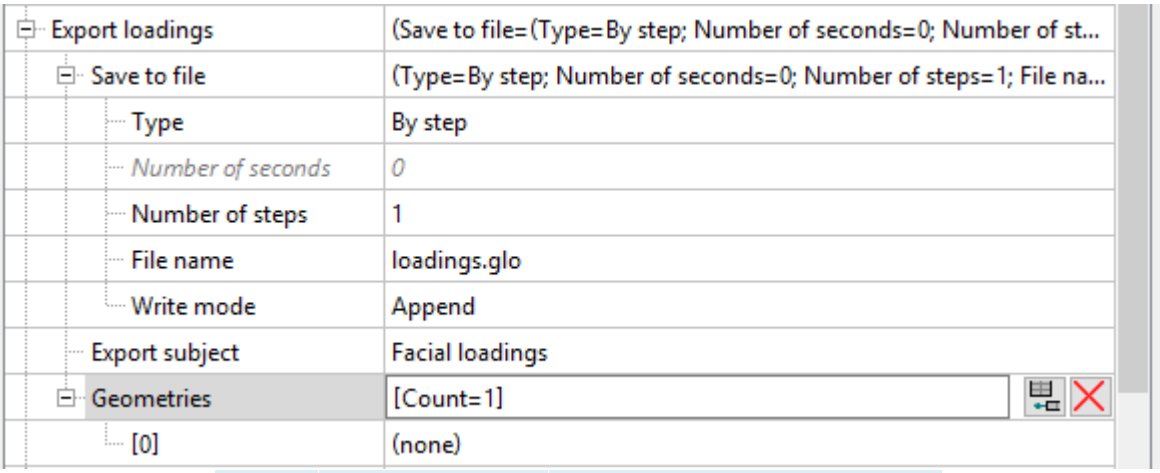
Element **Advanced settings**, parameters **Chemistry**

Group of parameters "Chemistry"	
Parameter	**Description**
Matrix Solver	Use matrix solver for computations in the **Chemistry** mass transfer model. Possible options are: **Yes** | **No**. The matrix solver allows joint solution of uniform convection-diffusion equations. Compared to the segregated solver, the matrix solver provides more stable development of solution at CFL>1 in simulations with large number of components (**Substances**) and large number of chemical reactions. Use of the matrix solver in the **Chemistry** makes sense only when a multicomponent problem is solved without specifying **Elements**.

Group of parameters "Export loadings"

Parameters from the **Export loadings** specify export of loadings into a finite element software.

Loads, which are stored in the file, can be later download in a finite element software package, in which geometry of the corresponding **Imported object** has been created, and use the loads there as a boundary condition.



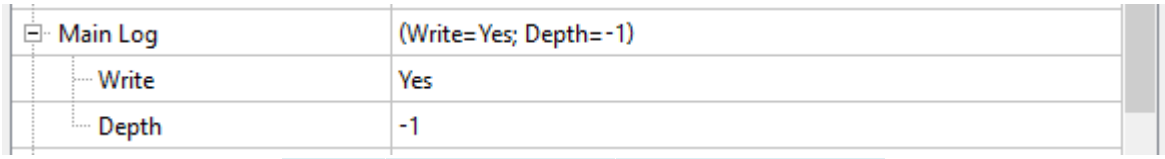
Element **Advanced settings**, parameters for export of loadings

Group of parameters "Export loadings"	
Parameter	**Description**
Save to File > Type	Type of saving the data into a file: • **Disabled**: no data are recorded • **Automatic**: the data are recorded on each time step • **By time**: the data are recorded after a specified time interval • **By step**: the data are recorded after a specified number of steps Regardless of the value of this parameter, the program will write into the file at stopping the computation, if **Save to file > Write mode = Overwrite** and **Save to file > Write on stopping = Yes** are selected (see below).
Save to File > Number of seconds	The time interval (for **Type = By time**)
Save to File > Number of steps	The number of steps (for **Type = By step**)
Save to File > File name	Name a file in the server part of the project, into which data are written. The default name is `loadings.glo`.
Save to File > Write mode	Possible options are: • **Overwrite**- overwrite the file each time you save the data • **Append**- append the data to the end of the file
Save to file> Write on stopping	This parameter instruct the program to make an additional writing into the file after triggering the specified **Stopping conditions** or manual stopping the computation. The program makes writing into the file at each stopping of the project regardless of the value of the **Save to file > Type** parameter. Possible options: **Yes** | **No**. This parameter is available when **Save to file > Type mode = Overwrite**.
Export subject	Format of transferred data for export of loadings. Possible options are: • **Nodal loadings** - in this mode the program exports the loadings on nodes of the geometry model. This setting is used by default. • **Facial loadings** - the program exports the loadings on faces (triangles) of the geometry model. See details in the section Parameters of loadings export.
Geometries	An array of **Subregions** and/or **Imported objects** with 3D geometric grids, loads from which are saved in the file. You can select any **Subregions** and **Imported objects** for export of loadings. You must not duplicate geometries in this list.

When **Solver** is connected to an external program (for example, *Abaqus*), it does not write records into the loadings export file (**loadings.glo**).

See description and examples of export loadings files in section Parameters of loadings export.

Settings of the logging (registration in the log file)



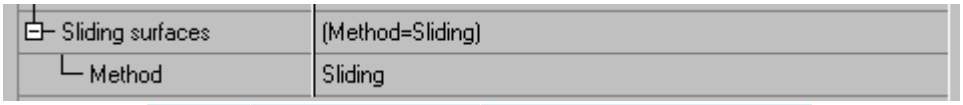
Element **Advanced settings**, settings of the logging

Parameter	Description
Main Log	Settings, which configure the logging of the project (the data registration in the **log**-file of the project)
Main Log > Write	Possible options: • **Yes** - enables the logging • **No** - disables the logging
Main Log > Depth	The logging level, which determines how detailed is the information that is recorded into the log. The log is a text file consisting of nested blocks, which are delimited by the "**Start:**" and "**End:**" keywords. Records of the same levels of logging, have the same indentation from beginnings of their lines. Possible options: • **0** - log the minimal information • *positive integer number* (for example, **1**, **2**, **3**, etc.) - log the information for this (first, second, third, etc.) level of nesting • **-1** - log the maximal detailed information (for all levels of nesting) To obtain the maximal informativeness, specify this parameter as "**-1**". For the minimal logging, specify "**0**", this can reduce the computational time because of minimal access to a disk and to reduce consumption of the disk space, but informativeness of the log will be minimal.

Example of a log's fragment:

```
   Start: Solve equation of continuity
     Start: Building matrix
       Reusing solver data
     End:   Building matrix WallTime= 2.031 2.03431
     Start: Preparing matrix for solution
     End:   Preparing matrix for solution WallTime= 0.781 0.772624
     Start: Building preconditioning
       Preconditioner params: N: 186508, Dens: 496.093585
     End:   Building preconditioning WallTime= 6.75 6.74612
     Start: Solving
       Solver statistics: Niter: 58.000000, BNorm: 22625.4, ResIni: 191.868, ResiFin:
1.40308e-006, ResIni2: 1.47924e-010, ResiFin2: 3.52444e-019, NVect: 2.000000, Ncols_Svd:
6.000000, SvMin_Svd: 0.000504876, SvMax_Svd: 0.0613805
     End:   Solving WallTime= 7.125 7.13536
   End:   Solve equation of continuity WallTime= 18.328 18.3168
```

Sliding surfaces



Element **Advanced settings**, settings for sliding surfaces

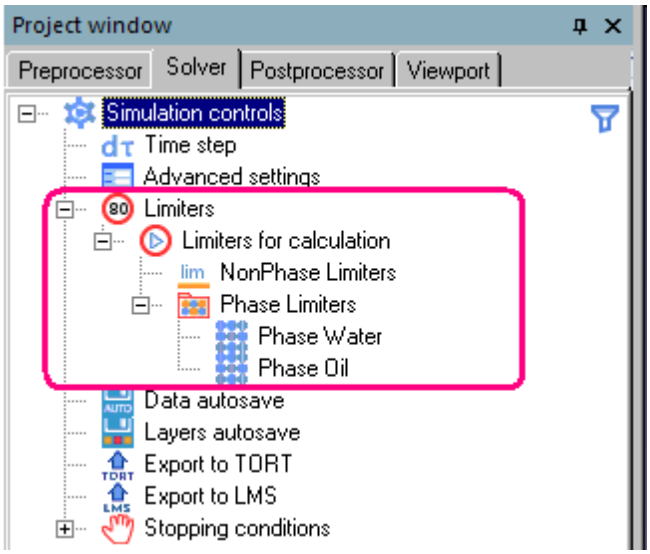
Parameter	Description
Sliding surfaces	Settings for simulation of sliding surfaces.
Sliding surfaces > Method	Possible options are: • **Sliding** for the main method of simulating a sliding surface • **Frozen Rotor** for the "frozen rotor" method is used, which assumes no sliding in cells on the boundary between subregions (rotor and stator). The "frozen rotor" option is recommended for receiving an initial approximation only when solution in the rotating subregion is close to an axially symmetric flow. You can use the **Frozen Rotor** method as the main method for simulation of flows in the sector problem formulation when there are several tens of sectors of the rotor and stator. When the number of sectors is less then 10, it is recommended to use the **Sliding** method.

Smoothing of diffusion fluxes on solid surfaces

Element **Advanced settings**, the setting for smoothing **Shear stress** and **Heat flux** on solid surfaces

Parameter	Description
Smooth diff. fluxes	This setting defines if the program smooths values of **Shear stress** and **Heat flux** variables on solid surfaces. Possible options are: • **No** – In this case, the standard algorithm is applied to compute the diffusion fluxes. When the values of **Sear stress** and **Heat flux** on a solid surface are calculated, the program uses the distance from the boundary to the center of the adjacent cell (the cells are arbitrarily truncated by the curvilinear surface). • **Yes** – In this case, for each cell truncated by the geometry the program calculates coordinates of the point, located at a fixed distance from the wall, are computed for each cell truncated by the geometry. The distance is determined by the dimensions of the original (non-truncated) cell and by the local normal to the wall. In computing diffusion fluxes (shear stress, heat flux, diffusion fluxes of the Substances), the values of the variables, stored in the cell centers, are interpolated into this point. In certain problems such introduction of an equidistant surface allows reducing oscillations of the calculated diffusion flux along a curvilinear wall.

8.1.8.4.3 Folder «Limiters»

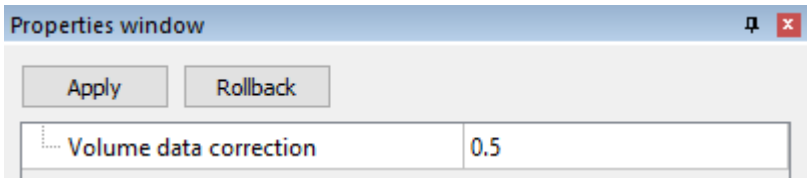


The folder **Limiters** contains the **Limiters for calculation** subfolder, which, in its turn, contains:

- the **NonPhase Limiters** element, which contains common limiters common *for all* **Phases** in the project

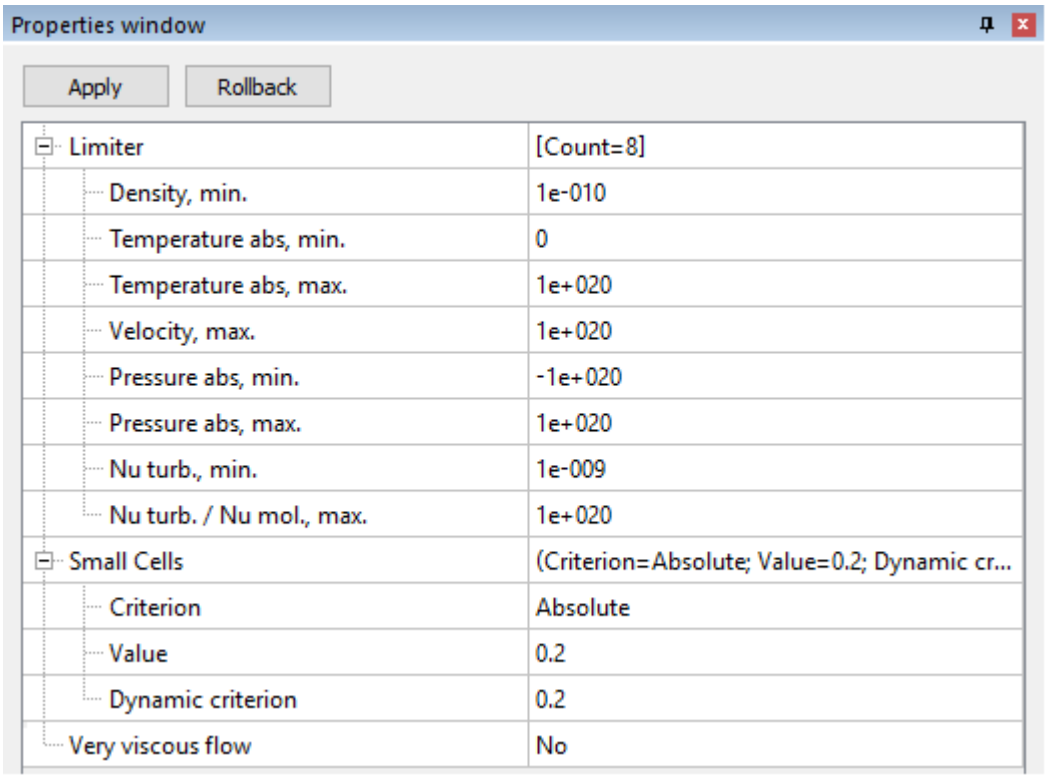
- the **Phase Limiters** folder with subfolders for each of the **Phases** in the project containing limiters for specific **Phases**

Limiters for calculation



The **Properties** window of the element **NonPhase Limiters**

Non-phase limiters for calculation	
Parameter	**Description**
Volume data correction	Volume data correction coefficient (developer's setting for simulation of movement of **Moving bodies**). This setting changes the volume of a cell (from previous time step) used for solving the equations. It allows to change the volume linearly depending on the value, corresponding from the strict satisfaction of the Gauss theorem for moving bodies (this matches to the zero value of the setting) to the value resulting from a real move (this matches to the value **1e+20** of the setting). The default value is **0.5**, its change is not recommended.



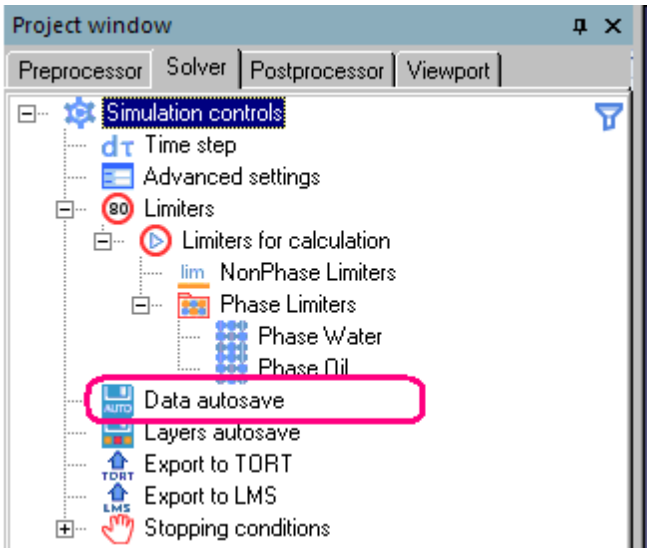
The **Properties** window of the element **Phase Limiters > Phase #N**

Phase limiters for calculation	
Parameter	**Description**
Density, min.	Limiters on values of density, temperature, velocity, and pressure, which can be achieved during the computation.
Temperature abs, min.	

Phase limiters for calculation	
Parameter	**Description**
Temperature abs, max. **Temperature (disp.) [N], min.** **Temperature (disp.) [N], max.** **Velocity, max.** **Pressure abs, min.** **Pressure abs, max.**	Temperature and pressure values in these limiters are specified in *absolute* values (see section Reference parameters, absolute and relative variables for details). You don't have to set the minimal pressure in a liquid, and in a gas it is recommended to set the minimal pressure as zero to avoid a negative density. Limiting values for temperature of dispersed particles are set individually for each size group (a size group is presented in the parameter's name as "**[N]**", where N=0, 1, ...).
Nu turb., min.	The minimal allowed turbulent kinematic viscosity [m^2 s^{-1}]. The default value is 10^{-8}.
Nu turb. / Nu mol., max.	The maximal allowable ratio of the turbulent kinematic viscosity to the local molecular viscosity. The default value is 10^{20}. In some problems, limitation of the maximum turbulent viscosity allows a user to avoid non-physical results at the initial stage of the development of solution.
Small Cells > ...	These are parameters for revealing small cells. Small cells are to be merged with their adjacent cells.
Small Cells > Criterion	This is criterion for revealing small cells. Possible options are: • **Absolute**: the cell is considered as small when, after its clipping by the geometry, its volume becomes less than the specified fraction of its initial volume. • **Relative**: the cell is considered as small when its volume is less than the specified portion of the volume of the largest adjacent cell. You should use the **Absolute** criterion for revealing small cells in all cases except simulating the icing and some other cases when there are explicit recommendations to specify the **Relative** criterion for revealing small cells.
Small Cells > Value	The threshold value of the cell's volume relating to: • volume of the initial cell (when **Criterion = Absolute**) • volume of the largest adjacent cell (when **Criterion = Relative**)
Small Cells > Dynamic criterion	The **Dynamic criterion** of cell smallness is used simultaneously with the specified **Absolute** or **Relative** criterion. This parameter specifies the threshold value of the dynamic volume fraction (it is the criterion of cell smallness relating to the previous volume of the cell). The cell is considered as small if ratio of its volume at the current time step to the volume at the previous time step is less than the specified fraction.
Very viscous flow	When the Reynolds number is too small (less than 1) you can set **Very viscous flow = Yes** to improve stability and accuracy of simulating high viscous liquids (for example, in simulations of forming rubber parts). Possible options: **Yes** | **No**.

See also: Parameters of small cells.

8.1.8.4.4 Element «Data autosave»

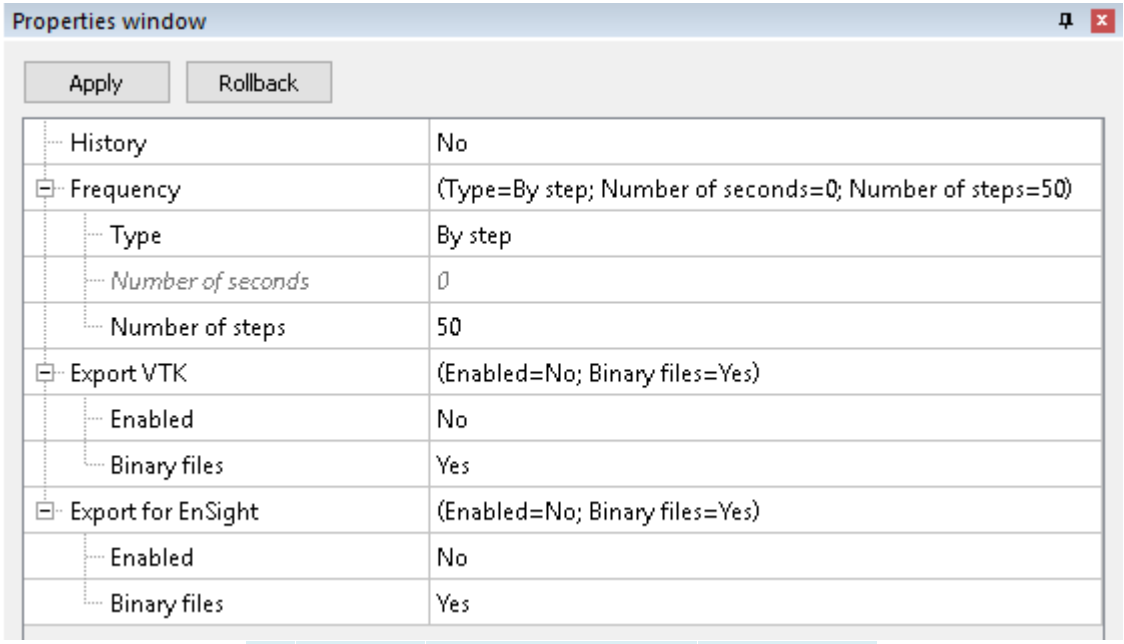


The **Data autosave** element contains parameters of automatic saving the results of the computation.

Recording the history of computation allows you to work with not only the latest results of the computation, but also with intermediate results:

1. view intermediate results using the toolbar **Navigation**
2. create animations

If necessary, you can delete some of these results using the **Non-steady-state steps decimation** dialog box.



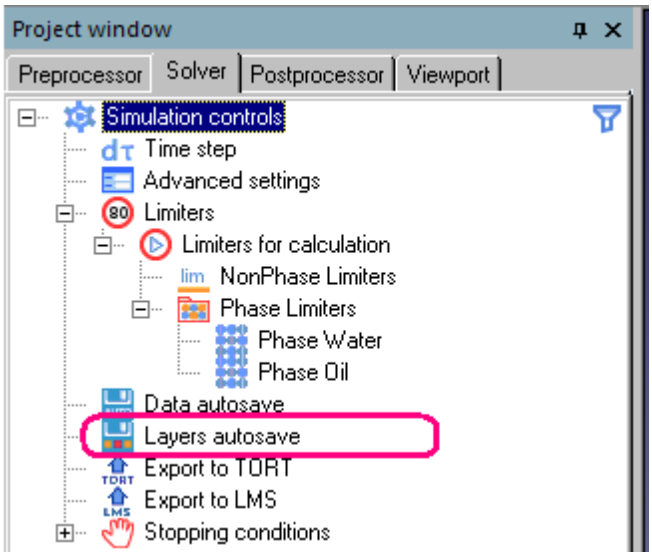
The **Properties** window of the element **Data autosave**

Parameters of the **Data autosave** element:

Parameter	Description
History	Start recording of the computation's history: • **No** – the calculation results are overwritten (recording the results obtained at this time step, over the previously saved) • **Yes** – the results of the calculation are stored in append mode
Frequency > Type	Possible options are: • **Disabled** – recording results of the computation is turned off • **Automatic** – recording results of the computation is done at the end of each step of the calculation

Parameter	Description
	• **By time** – recording results of the computation is done performed periodically with a period specified by the **Number of seconds** parameter • **By step** – recording results of the computation is done periodically with a period specified by the **Number of steps** parameter
Frequency > Number of seconds	The interval (in seconds) between two records of calculation results. This parameter is available when **Type = By time**.
Frequency > Number of steps	The interval (in steps) between two records of calculation results. This parameter is available when **Type = By step**.
Export VTK	Group of parameters that specify data export in the *VTK* format. *VTK* format assumes exporting of the computational grid and values of variables in each cell of the grid. If the computation was done on multiple processors, then data from each processor are saved in separate files. Also, if computational cells belong to multiple **Subregions**, then data from each **Subregion** are saved in separate files. The file names are formed in the `grid_nRank_nSubreg` format.
Export VTK > Enabled	Enable periodic data export in the *VTK* format. Possible options: **Yes** | **No**.
Export VTK > Binary files	Use binary files for data export in the *VTK* format. Possible options: • **Yes** - use binary files • **No**- use text files
Export for EnSight	Group of parameters of data export for visualization in *EnSight*. See section Data export for visualization in EnSight for details.
Export for EnSight > Enabled	Enable periodic data export for *EnSight*. Possible options: **Yes** | **No**.
Export for EnSight > Binary files	Selection of format of files for export to *EnSight* (binary or text files). Possible options: • **Yes** - use binary files • **No**- use text files

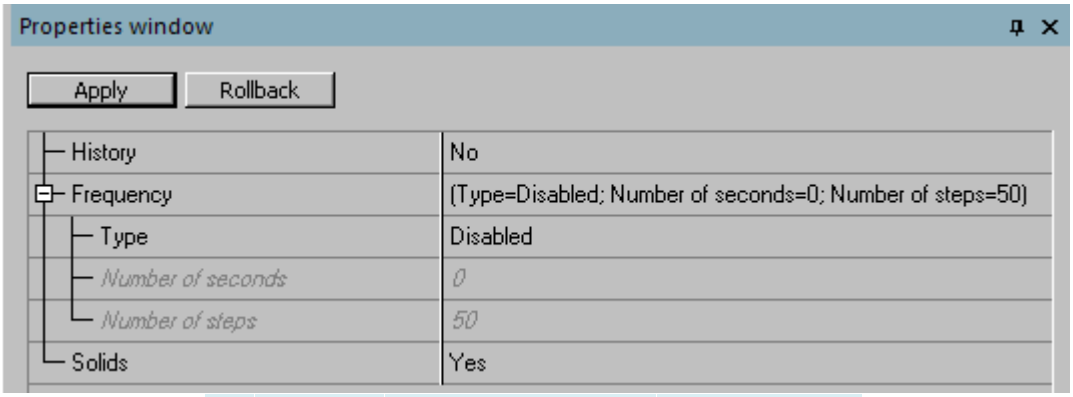
8.1.8.4.5 Element «Layers autosave»



Layers autosave element is designed to display and set the control parameters automatically save to a file `*.fvvis` data visualization layers.

The `fvvis`-files are stored in the server part of the project. If necessary, they can be downloaded to the client of the project using the **File > Download additional files > Visualization files** menu item.

The stored visualization data may then be opened in **Pre-Postprocessor** or **Viewer**.

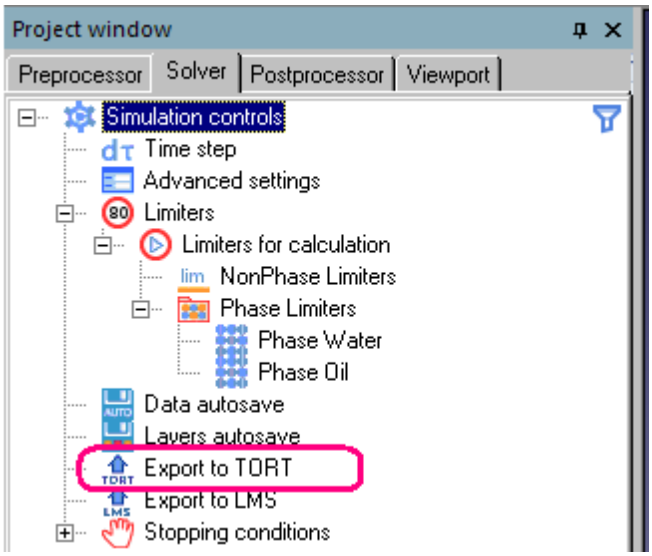


The **Properties** window of the element **Layers autosave**

Parameters of the element **Layers autosave**:

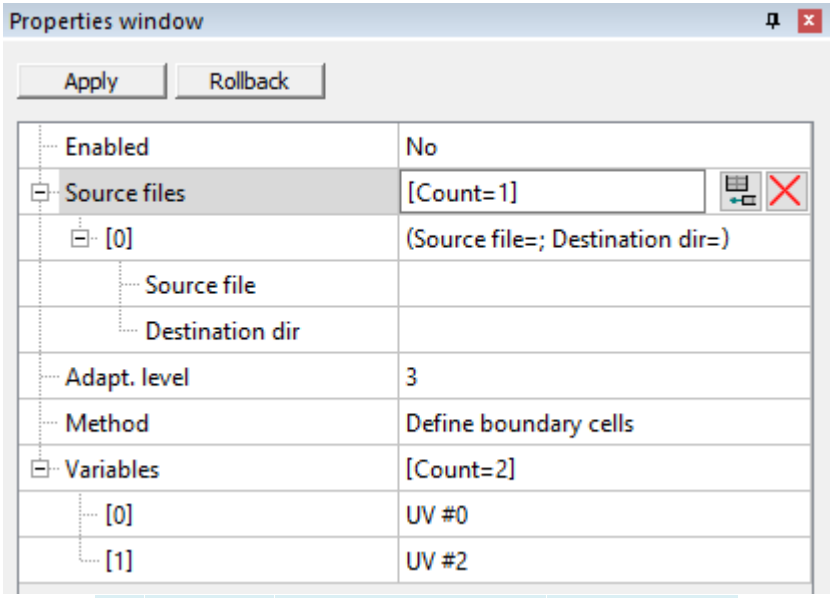
Parameter	Description
History	Start recording the history of visualization. Possible options are: • **None**- the data to visualize the layers are saved in overwrite mode (recording the results obtained at this time step, over the previously saved); • **Yes** - the data to visualize the layers are saved in append mode
Frequency > Type	Possible options are: • **Disabled** - record data to visualize the layers off • **Automatic** - record data to visualize the layers is performed at the end of each time step • **By time** - write data to visualize the layers is performed periodically with a period specified by the **Number of seconds** parameter • **By step** - write data to visualize the layers is performed periodically with a period specified by the **Number of steps** parameter
Frequency > Number of seconds	The interval between two data records to render layers. Option is available if **Type = By time**.
Frequency > Number of steps	The interval between two data records to render layers. Possible options are: Option is available if **Type = By step**.
Solids	Whether to record the **Solids** layer into the visualization data. Possible options are: • **No** - do not include the **Solids** layer into the data for visualization of layers • **Yes** - include the **Solids** layer into the data for visualization of layers

8.1.8.4.6 Element «Export to TORT»



The **Export to TORT** element is used to display and specify the control parameters of interaction FlowVision and TORT.

The weights equal volume fractions of the carrier fluid in the respective zones of the reactor, are automatically recorded in the output files. An array of weighting factors is given in TORT-input file.



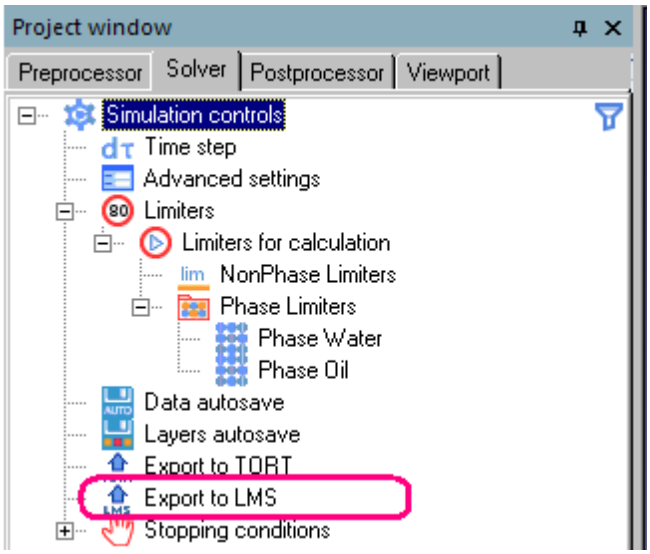
The **Properties** window of the element **Export to TORT**

Parameters of the element **Export to TORT**:

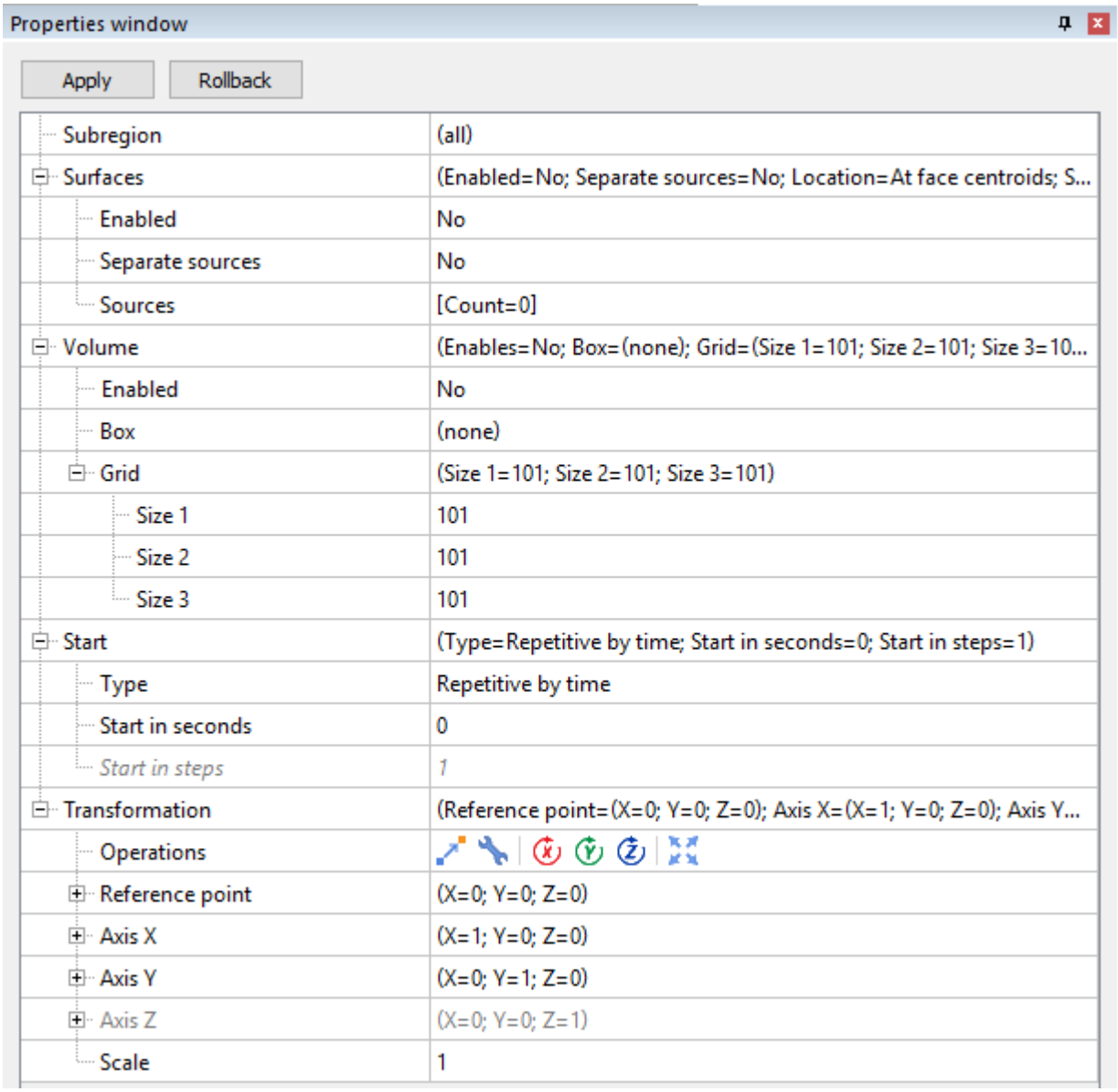
Parameter	Description
Enabled	Defines if export to *TORT* is enabled: • **No** - there is no export data to *TORT* • **Yes**- export data in *TORT* is made at each automatic or manual (by pressing the **Save solution Solver** inthe toolbar **Standard**) save the data on the server side project
Source files	An array of input files.
Source files > [N] > Source file	Absolute or relative (relative to the directory server side project) path to input file to export data to TORT. A relative path is specified from the server directory of the project.

Parameter	Description
Source files > [N] > Destination dir	Absolute or relative (relative to the directory server side project) directory path to the output file. Each input file must match its output directory. If you specify empty or duplicate names directory FlowVision corrects them automatically.
Adapt. level	The level of fragmentation of cells (if **Method = Define boundary cells**).
Method	The processing method of boundary cells used to calculate the concentration of isotopes: • **Define boundary cells** - the use of the method with the definition of the boundary cells • **Don't define boundary cells** - the use of the method without specifying the boundary cells See section Method of calculation of isotope concentrations.
Variables	An array for specifying the list of variables.
Variables > [N]	Selecting a user variable. *FlowVision* will automatically remove any empty and/or repetitive user variables.

8.1.8.4.7 Element «Export to LMS»



In the **Properties** window of the element **Export to LMS** we define parameters of exporting pressure to the LMS software.



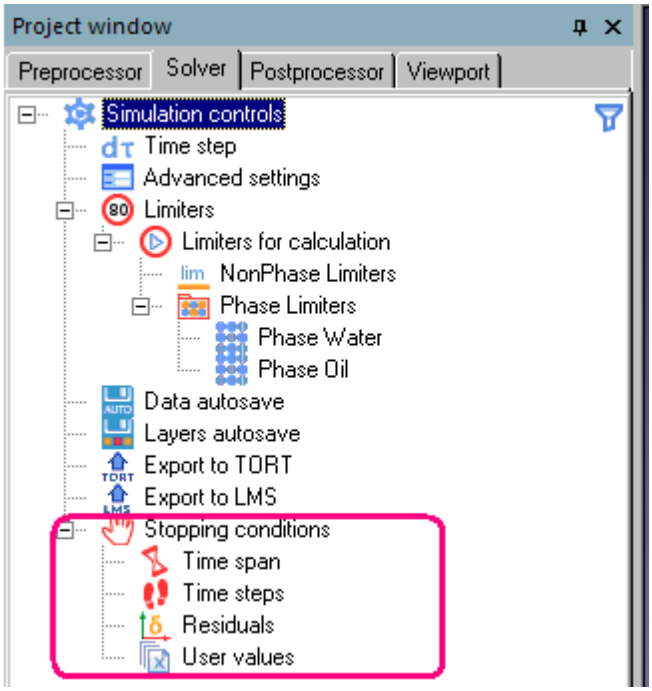
The **Properties** window of the element **Export to LMS**

Parameter	Description
Subregion	The **Subregion**, from which values of the pressure are passed. When this parameter is set as **(all)**, the program exports the data from all **Subregions**.
Surfaces	This group of parameters provides settings for data export from surfaces.
Surfaces > Enabled	This parameter enables export to *LMS* from surfaces of **Boundary conditions** that were selected by the **Surfaces > Boundaries** array (see below). Possible values: **Yes** | **No**.
Surfaces > Separate sources	If data sources are separated, then data from each surface are recorded in separate files. Possible values: **Yes** | **No**.
Surfaces > Boundaries	This is an array of **Boundary conditions**, from which pressure is transferred to the *LMS*.
Volume	This group of parameters provides settings for data export from specified nodes in volume of the selected **Box**.

Parameter	Description
Volume > Enabled	This parameter enables export to *LMS* from the selected volume. Possible values: **Yes** | **No**.
Volume > Box	A **Box** (created in **Preprocessor**), from volume of which the pressure is exported. Data from volume of the **Box** are transferred from nodes of a grid that is oriented along axes of the local coordinate system of the **Box** (see below).
Volume > Grid > Size 1 **Volume > Grid > Size 2** **Volume > Grid > Size 3**	Number of nodes, from which the data are transferred, in the grid along axes X, Y, Z of the local coordinate system of the **Box** (see above).
Start > Type	This is method of specifying the periodic data export. Possible values are: • **Repetitive by time** – start data export to *LMS* at the specified time and continue periodically over this time • **Repetitive by step** – start data export to *LMS* at the specified step number and continue periodically over this number of steps
Start > Start in seconds	The time point when data export to *LMS* begins and period between subsequent data transfers (this parameter is available when **Start > Type = Repetitive by time**)
Start > Start in steps	The step, at which data export to *LMS* begins, and period in steps between subsequent data transfers (this parameter is available when **Start > Type = Repetitive by step**)
Transformation	These are settings of geometry transforming of surfaces of boundary conditions at their export to *LMS* (the additional transformation of coordinates is done at recording to CGNS-files). Buttons for operations and parameters here are similar to appropriate buttons and parameters in properties of geometry **Objects** (see details in the section Objects in the project tree).
Transformation > Operations	Buttons for geometry transformation at export to *LMS*: (**Relative translation**) (**Coordinate system adjustment**) (**Relative rotation around local axis X**) (**Relative rotation around local axis Y**) (**Relative rotation around local axis Z**) (**Relative scaling**) Use of these buttons is generally more convenient then setting numerical values of parameters **Transformation > ...**
Transformation > Reference point > X **Transformation > Reference point > Y** **Transformation > Reference point > Z**	Coordinated of the reference point of the geometry for data export to *LMS* in the absolute coordinate system.
Transformation > Axis X > X **Transformation > Axis X > Y** **Transformation > Axis X > Z**	Direction of the axis X of the local coordinate system of the geometry for data export to *LMS* in the absolute coordinate system. After clicking the **Apply** button the program automatically normalize unit vectors of the exported geometry's local coordinate system.
Transformation > Axis Y > ...	Direction of the axis Y of the local coordinate system of the geometry for data export to *LMS* in the absolute coordinate system. After clicking the **Apply** button the program automatically normalize unit vectors of the exported geometry's local coordinate system.

Parameter	Description
Transformation > Axis Z > ...	Direction of the axis Z of the local coordinate system of the geometry for data export to *LMS* in the absolute coordinate system. These parameters are calculated automatically and they are read-only.
Transformation > Scale	Scaling factor of the geometry for data export to *LMS*

8.1.8.4.8 Folder «Stopping conditions»



The **Stopping conditions** folder in the project tree

In the folder **Stopping conditions** you can specify the criteria under which the computation of the project is stopped automatically (see also the section Stopping conditions for the calculation).

There are following stopping conditions:

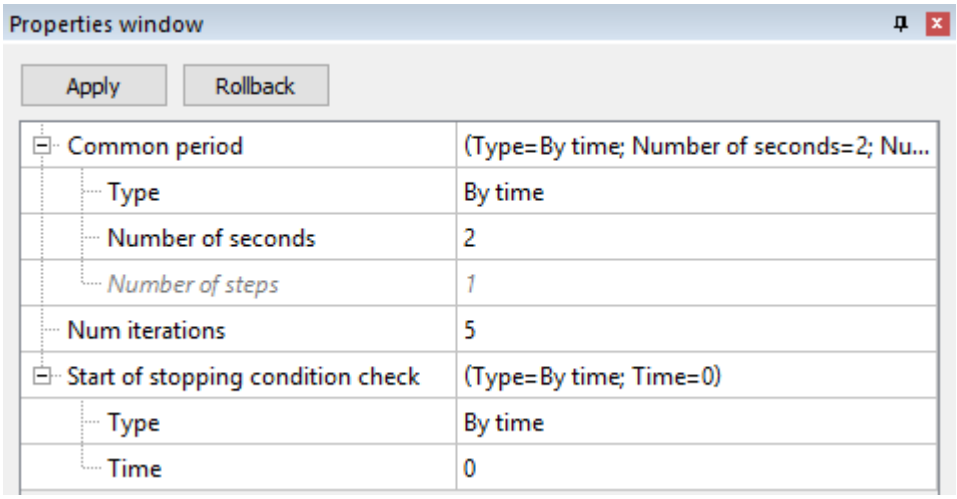
- by time
- by the number of steps
- by value of residuals
- by user variables

The **Stopping conditions** folder contains:

- element **Time span**
- element **Time steps**
- subfolder **Residuals** with child elements ***(Variable)***
- subfolder **User values** with child elements **Stop criterion #N**

(see appropriate subsections below)

The Properties window of the folder "Stopping conditions"



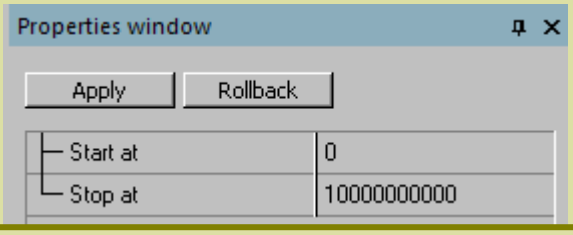
The **Properties** window of the folder **Stopping conditions**

Parameters of the folder "Stopping conditions"		
Parameter	**Description**	**Dimension**
Common period > ...	This group of parameters specifies averaging of the function of residuals and user stopping conditions. The **Common period** allows you to specify T, the common averaging period for residuals and those **User values**, for which *individual averaging periods are not specified*.	
Common period > Type	This parameter defines the method of specifying duration of the **Common period** T (see above). Possible options are: **By time | By step**.	
Common period > Number of seconds	Duration of the **Common period** in seconds. The value **0** means that only those residuals and target functions will be averaged, for which individual averaging periods are set.	[s]
Common period > Number of steps	Duration of the **Common period** in steps. The value **0** means that only those residuals and target functions will be averaged, for which individual averaging periods are set.	
Num iterations	n, the number of time steps on which functional residuals and/or stopping criteria defined by user values or characteristics are calculated (see subsections *"Subfolder «Residuals» and elements in it"* and "*Subfolder "User values" and elements "Stop criterion #N"*" below).	
Start of stopping condition check > ...	This group of parameters allows you to start check of the stopping conditions not from beginning of the computation but from a specified time moment or from a specified step.	
Start of stopping condition check > Type	Possible options are: **By time | By step**.	
Start of stopping condition check > Time	The time, starting from which the stopping conditions will be checked.	[s]
Start of stopping condition check > Step number	The step, starting from which the stopping conditions will be checked.	

Element "Time span"

Element **Time span** in the project tree

Properties of the element "Time span"

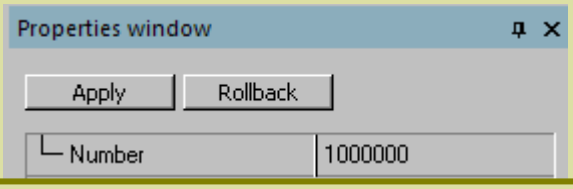


Parameter	Description	Dimension
Start at	Start time of the computation	[s]
Stop at	Stop time (the time when the computation will be stopped)	[s]

Element "Time steps"

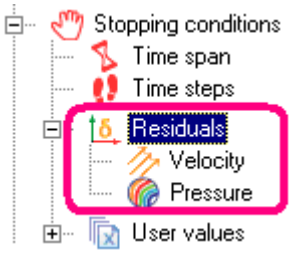
Element **Time steps** in the project tree

Properties of the element "Time steps"



Parameter	Description
Number	The number of time steps before stop of the computation.

Subfolder «Residuals» and its child elements

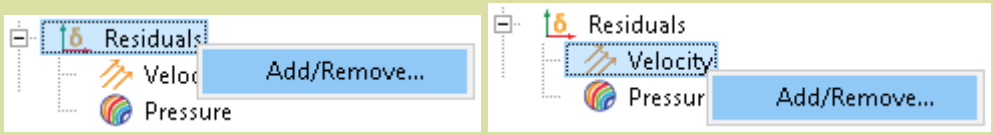


Folder **Residuals** in the project tree

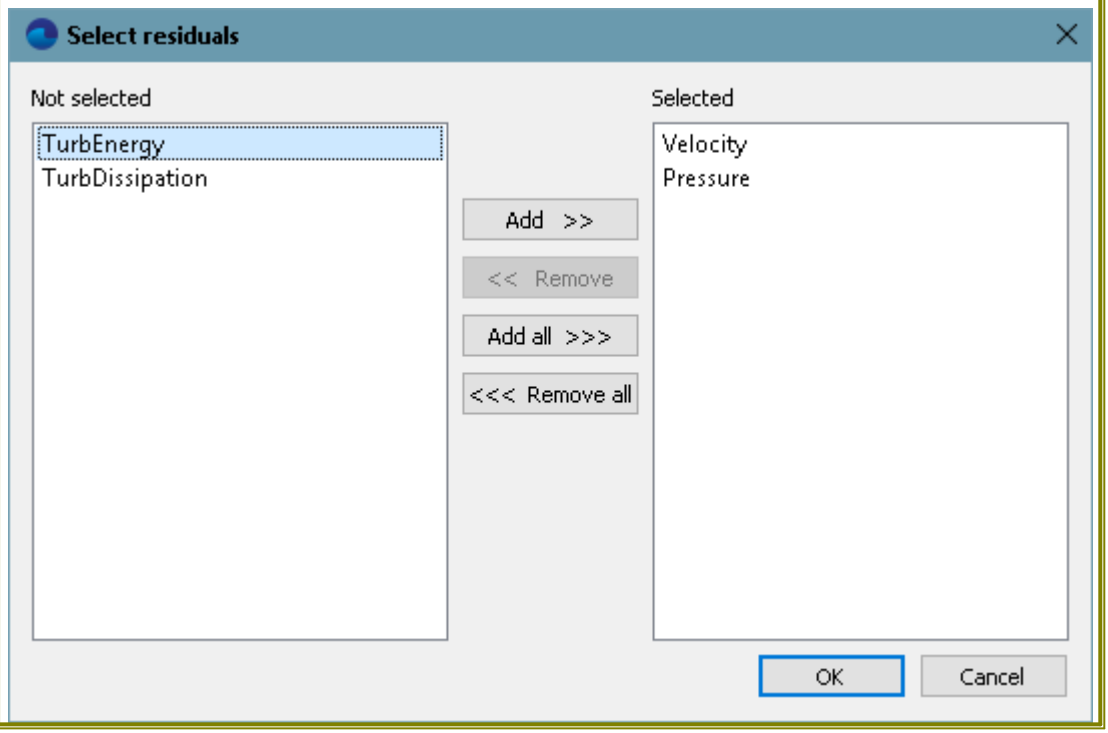
Folder **Residuals** is intended for:

- specifying stopping conditions by functional residuals of calculated variables
- displaying a plot of the dependance the functional residuals of calculated variables by time

Context menu of the folder **Stopping conditions > Residuals** and context menus of its child elements ***(Variable)*** contain the **Add/Remove** command for adding and/or removing the child elements **... > *(Variable)***.

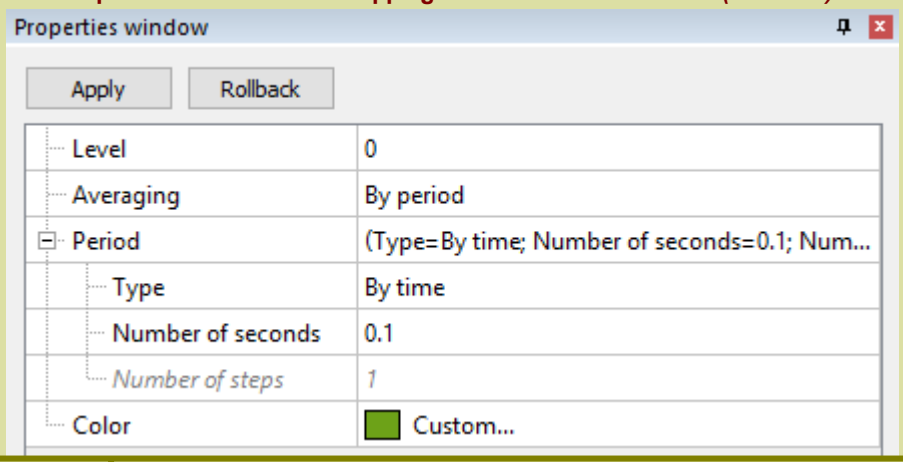


Context menus of the folder "Residuals" and its child elements

Menu item	Description
Add/Remove	Opens the **Select residuals** dialog window, which allows you to change the list of child items in the **Residuals** folder: 

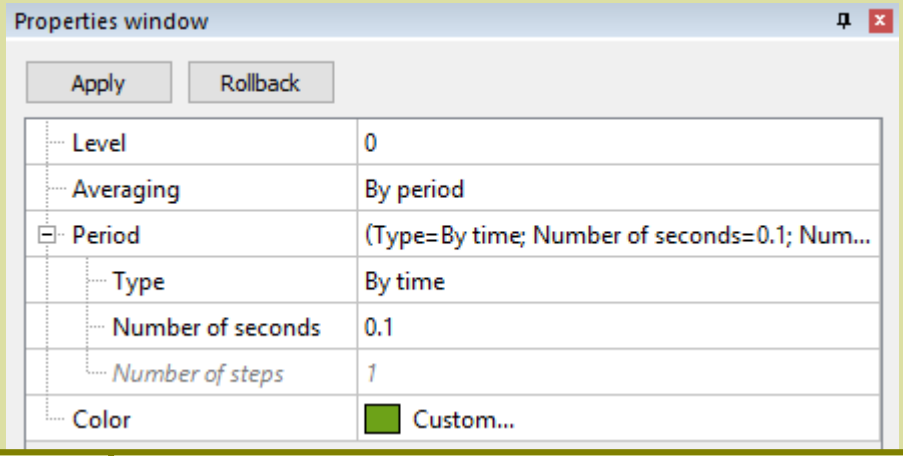
The **Residuals** folder has no parameters in its **Properties** window.

Properties of elements "Stopping conditions > Residuals > *(Variable)*"



Parameter	Description	Dimension
Level	ε, the critical value of the functional residual of the calculated variable. If at last n time steps (the n value is specified by the **Num iterations** parameter in properties of the **Stopping conditions** folder) the difference between maximum and minimum of the residual of the calculated variable is less than ε, then the computation stops.	

Properties of elements "Stopping conditions > Residuals > *(Variable)*"



Parameter	Description	Dimension
	Test of the stopping conditions can be also done for the *averaged* variable's residual variation.	
Averaging	Averaging allows the program to accelerate convergence of the solution. Averaging forms an averaged residual variation function. Testing the stopping conditions is done for the averaged function according to the formula: $$\overline{f}(t) = \frac{1}{T}\int_{t-T}^{t} f(\tau)d\tau$$ where T is the averaging period. The **Averaging** parameter specifies the method of setting the averaging period (will the averaging period be the standard, i.e. applied by default, or will it be set individually for this variable). The possible values are: • **By period** - declares an individual averaging period for this calculated variable, and the duration of this averaging period is specified by parameters **Period > ...**, see below. • **By common period** - the averaging of the calculated variable is done by the period, which is specified by parameters **Common period > ...** in properties of the **Stopping conditions** folder. If you wish not to apply the averaging, set either **Period > Number of seconds = 0** or **Period > Number of steps = 0** (see below).	
Period	This group of parameters allows you to specify the individual averaging period T of the calculated variable (these parameters can only be set if **Averaging = By period**, see above).	
Period > Type	This parameter defines the method of specifying duration of the individual averaging period. Possible options are: **By time | By step**.	
Period > Number of seconds	Duration of the individual averaging period in seconds. The value **0** means that the averaging is not applied.	[s]
Period > Number of steps	Duration of the individual averaging period in steps. The value **0** means that the averaging is not applied.	
Color	Color of the plot of the residual of the calculated variable against the time, which is displayed in the **Monitor** window.	

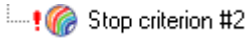
Subfolder "User values" and elements "Stop criterion #N"

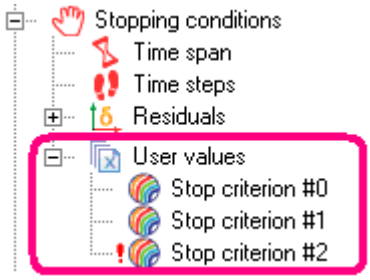
Subfolder **Stopping conditions > User values** contains elements **Stop criterion #N**.

Element **Stop criterion #N** is intended for the following:

- stopping the calculation when the specified minimal variation of the target function on the last n time steps
- displaying the plot of the dependance the target function by time

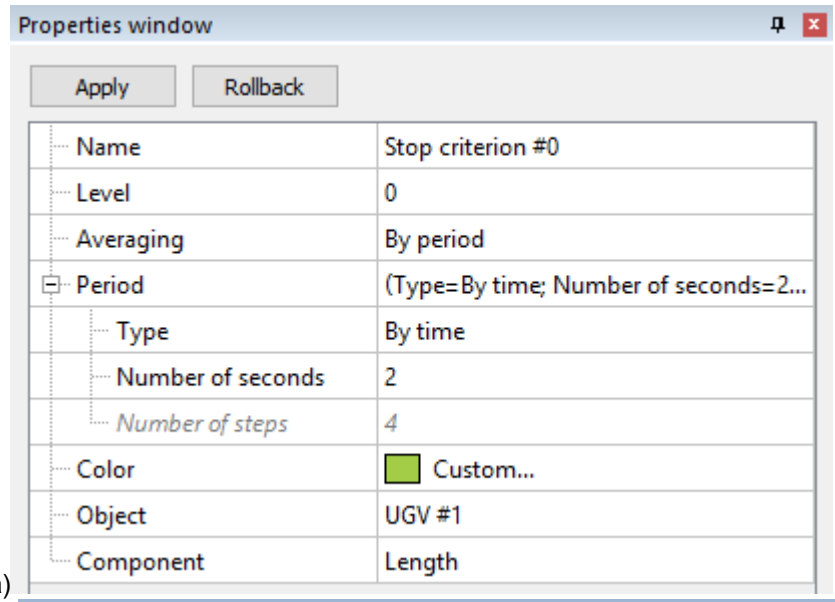
If the parameter **Object** in the **Properties** window is not specified, then this **Stop criterion #N** element is marked with the "**!**" sign in the project tree:



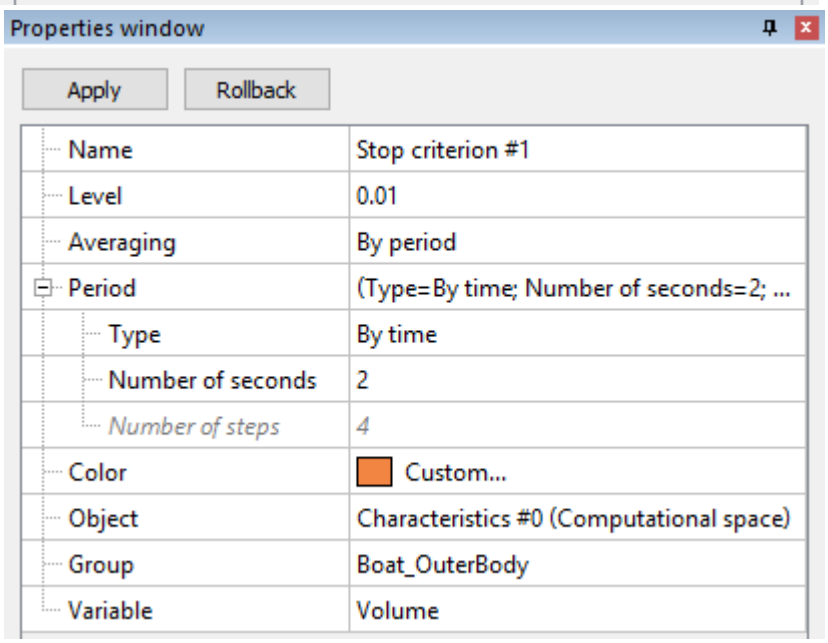


Elements **User values > Stop criterion #N**

a)



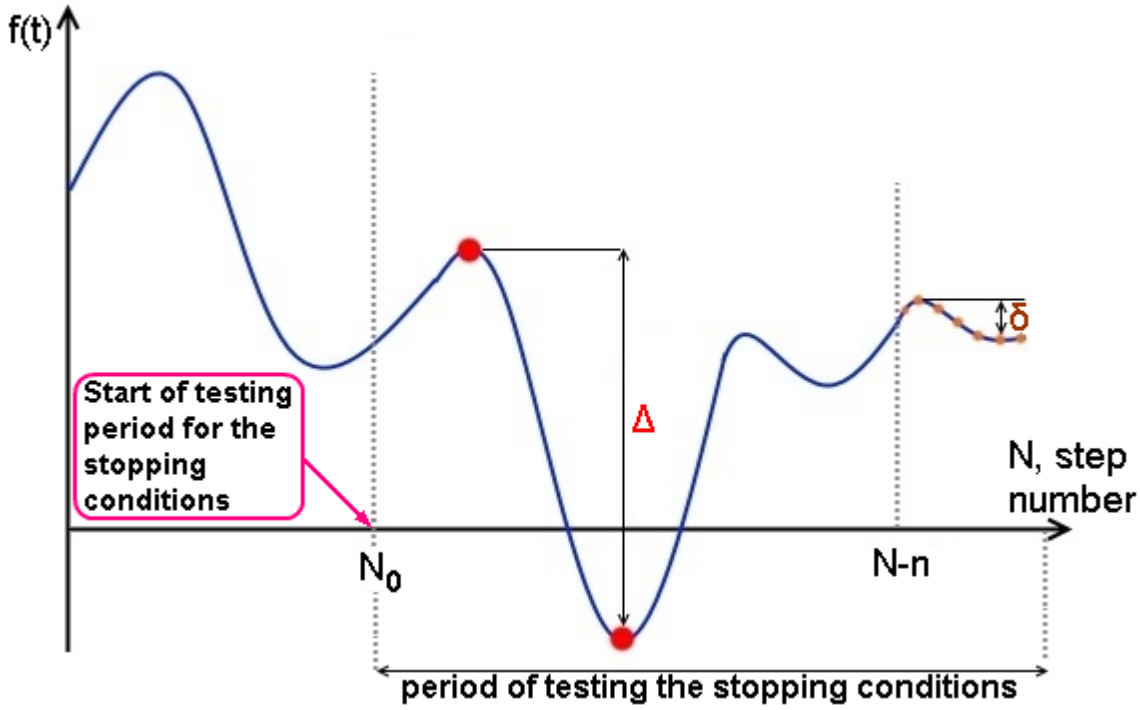
b)



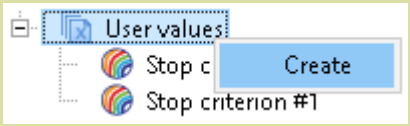
The **Properties** window of the element **Stop criterion #N**: a) for a global user variable (vector); b) for a variable from **Characteristics**

Properties of elements "User values > Stop criterion #N"

Parameter	Description
Name	Name of a stopping criterion of the calculation. If necessary, you can change the default name **Stop criterion #N**.
Level	The **Level** parameter specifies the value ε. The computation stops when $\delta/\Delta<\varepsilon$, where: δ – is variation of the target function $f(t)$ at the last n time steps (the n number is set by the **Num iterations** parameter in properties of the **Stopping conditions** folder) Δ – is variation of the target function $f(t)$ at the period of testing the stopping conditions (start of this period is set by parameters **Start of stopping condition check > ...** in properties of the **Stopping conditions** folder).

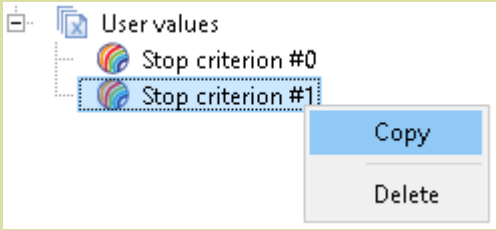
Parameter	Description
	 Testing the **Stop criterion** can also be done for an averaged function $\overline{f}(t)$ that is calculated by the formula: $$\overline{f}(t)=\frac{1}{T}\int_{t-T}^{t} f(\tau)d\tau$$ where T is the averaging period specified by the n the the parameter **Period** (see below) or by parameters **Common period > ...** in properties of the **Stopping conditions** folder.
Averaging	Averaging allows the program to accelerate convergence of the solution. Possible options are: • **By period**: the averaging of the target function $f(t)$ is done at the period specified by the **Period** parameter. • **By common period**: the averaging of the target function $f(t)$ is done with period specified by parameters **Common period > ...** in properties of the **Stopping conditions** folder.
Period > ...	When averaging is used, the stopping criterion uses an averaged function $\overline{f}(t)$ (averaged by the period T) *instead of* the target function $f(t)$ according to the following formula: $$\overline{f}(t)=\frac{1}{T}\int_{t-T}^{t} f(\tau)d\tau$$ The **Period** group of parameters allows you to specify individual (for this stop criterion) averaging period T of the target function $f(t)$.
Period > Type	This parameter defines the method of specifying duration of the individual averaging period. Possible options are: **By time** | **By step**.
Period > Number of seconds	Duration of the individual averaging period, [s]. The value **0** means that averaging is not applied.

Parameter	Description
Period > Number of steps	Duration of the individual averaging period in steps. The value **0** means that averaging is not applied.
Color	Color of the plot of the target function against the time, which is displayed in the **Monitor** window.
Object	This parameter specifies the target function $f(t)$ for this **Stop criterion**. You can specify: • global user variable or constant • *or* a target value calculated by some **Characteristics** (you specify the target value by parameters **Group**, **Variable**, and optionally by **Component**, see below).
Group	A group that includes a variable that is calculated in the **Characteristics** (this parameter is available when **Object = Internal characteristics | Characteristics #N**)
Variable	A variable that is calculated by the **Characteristics** (this parameter is available when **Object = Internal characteristics | Characteristics #N**) For example, for **Characteristics** specified in the **Computational space** by **Pressure**, the following options are available: **Volume	Mass	<f vol.>	<f mass>	<f vol.> * Volume	Maximum	Point max.	Minimum	Point min.** For **Internal characteristics** the following options are available: **Current time	Current step number	Time step	Explicit time step	Reference temperature	Reference pressure	Gravity vector**.
Component	If **Object = *a global vector user variable or constant***: The length or a component of the vector. If **Object = Internal characteristics	Characteristics #N**: The length or a component of the vector (or distance from the coordinate system's origin to a point or a point's coordinate), in the case when the **Variable** parameter (see above) specifies a vector value (for example, **Gravity vector**) or a point in the space (for example, **Point max.** or **Point min.**). Possible options: **Length	X	Y	Z**.



Context menu of the "User values" folder

Menu item	Description
Create	Creates an element **Stop criterion #N** in the folder **User values**

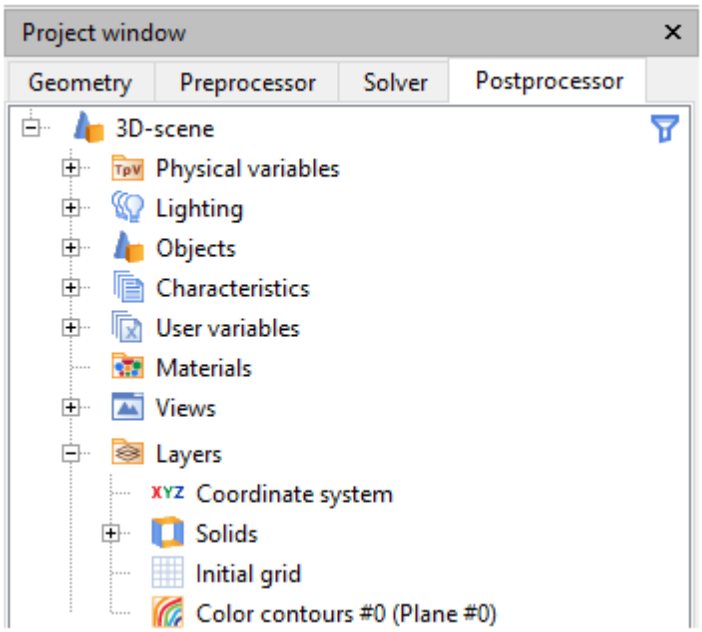


Context menu of the "Stop criterion #N" elements

Menu item	Description
Copy	Creating an element, which is a copy of the selected element
Delete	Deleting the selected element from the project tree

8.1.8.5 The Project window, tab «Postprocessor»

The **Postprocessor** tab displays the tree of the project, which contains elements relating to the analysis of the results of the computation.



The **Project** window, the **Postprocessor** tab

The root folder of the **Postprocessor** tab has the name **3D-scene**, which contains the following subfolders:

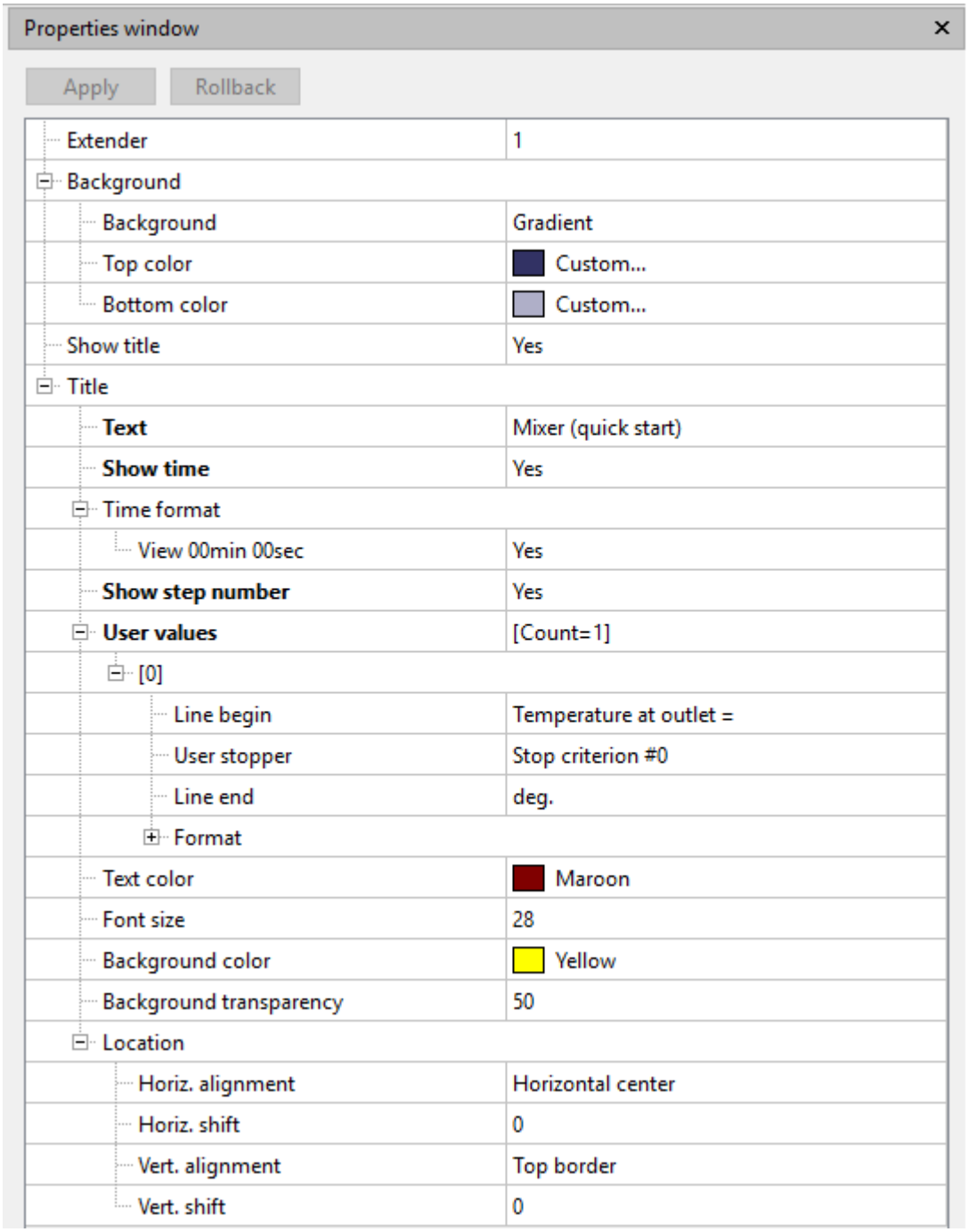
- **Physical variables**
- **Lighting**
- **Objects**
- **Characteristics**
- **User variables**
- **Materials**
- **Views**
- **Layers**

Parameters of the «3D-scene» root folder

Parameters of the **3D-scene** root folder are stored in its **Properties** window.

These parameters allow you to specify for the **View** window:

- scaling of the visible area
- settings of the background
- text and settings of the **Title** (the **Title** might display the current time, values of variables and the step number of the computation)

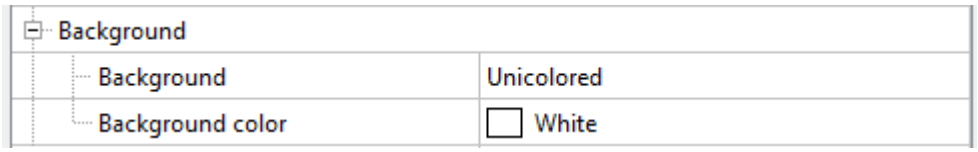


The **Properties** window of the **3D-scene** root folder

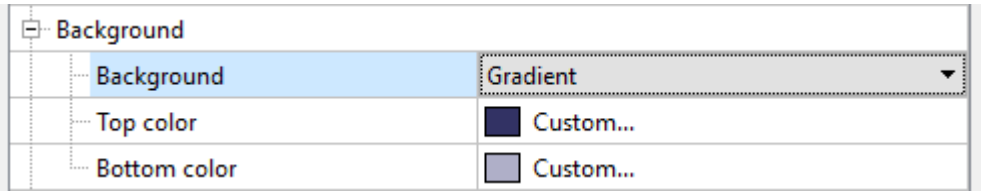
Parameters of the «3D-scene» root folder	
Parameter	**Description**
Extender	Expanding the visible area according to the specified scaling (the image of the object will be reduced according to this scaling)
Background	Possible options are: • **Unicolored** - plain background color specified by the **Background color** parameter • **Gradient** - the background color is transition from the color, which is specified by the **Top color**, to the color, which is specified by the **Bottom color** • **Palette specified** - the background color is specified by the **Palette**, which defines transition between several specified colors

Parameters of the «3D-scene» root folder	
Parameter	**Description**
Background color	This parameter specifies the background color when **Background=Unicolored** is set.
Top color **Bottom color**	These parameters specify colors at the top and the bottom edges of the background when **Background=Gradient** is set.
Palette > ...	Operations with palettes and palette settings when **Background=Palette specified**. See details about **Palettes** and their use in the section Parameters for defining a palette.
Show title	This parameter defines if the title will be displayed in the **View** window. Possible options are: **Yes | No**.
Title > ...	Parameters of the title.
Title > Text	Text of the title. This text will be displayed in bold type on the top.
Title > Show time	This parameter defines displaying the simulated **Time**. Possible options are: **Yes | No**.
Title > Time format > ...	Settings of the format for displaying the time. The standard (non-split) format displays the time in seconds as a decimal fraction.
Title > Time format > View 00min 00sec	Displaying the time in the split format (in hours, minutes, seconds, milliseconds and microseconds; values of three time units of measure are displayed, for example "**2hs 36min 46sec**"). Possible options are: **Yes | No**.
Title > Time format > Scientific	Displaying the time in the exponential (scientific) format ("mantissa **e** exponent") in seconds. Possible options are: **Yes | No**.
Title > Time format > Num.digits	Number of digits in the fraction part of the time displaying format (in non-split and in exponential formats). You can set this value in the range from **1** to **16**.
Title > Show step number	Displaying the computational step number. Possible options are: **Yes | No**.
Title > User values > [N] >...	An array of groups of parameters that allow displaying selected user **Stop criteria** accompanied on the left and on the right with specified text lines. **N** = **0**, **1**, **2**, ...
Title > User values > [N] > Line begin	The text line, which will be displayed on the left from the value of the **Stop criterion**.
Title > User values > [N] > User stopper	The user **Stop criterion**, value of which will be displayed in the title. If it contains no data, the "**?**" symbol will be displayed.
Title > User values > [N] > Line end	The text line, which will be displayed on the right from the value of the **Stop criterion**.
Title > Text color	Font color of the title
Title > Font size	Font size of the title
Title > Background color	Background color of the title
Title > Background transparency	Transparency of the title's background
Title > Location > ...	These parameters allows you to define location of the title within the **View** window.
Title > Location > Horiz. alignment	Horizontal alignment of the title in the **View** window (without shifting). Possible options are: **Left border** | **Horizontal center** | **Right border**.
Title > Location > Horiz. shift	Shifting the title in the right direction relating to the position that is defined by the **Horiz. alignment** parameter. It is defined by a numerical value, which defines a fraction of the horizontal size of the **View** window. To define a shift in the left direction, use negative values of this parameter.

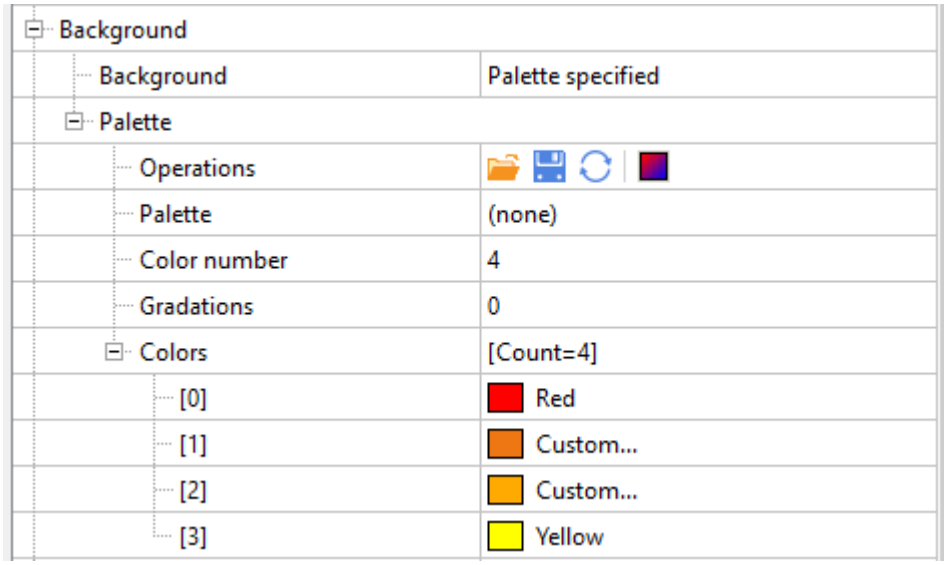
Parameters of the «3D-scene» root folder	
Parameter	**Description**
Title > Location > Vert. alignment	Vertical alignment of the title in the **View** window (without shifting). Possible options are: **Top border** | **Vertical center** | **Right border**.
Title > Location > Vert. shift	Shifting the title downwards relating to the position that is defined by the **Vert. alignment**. It is defined by a numerical value, which defines a fraction of the vertical size of the **View** window. To define an upward shift, use negative values of this parameter.



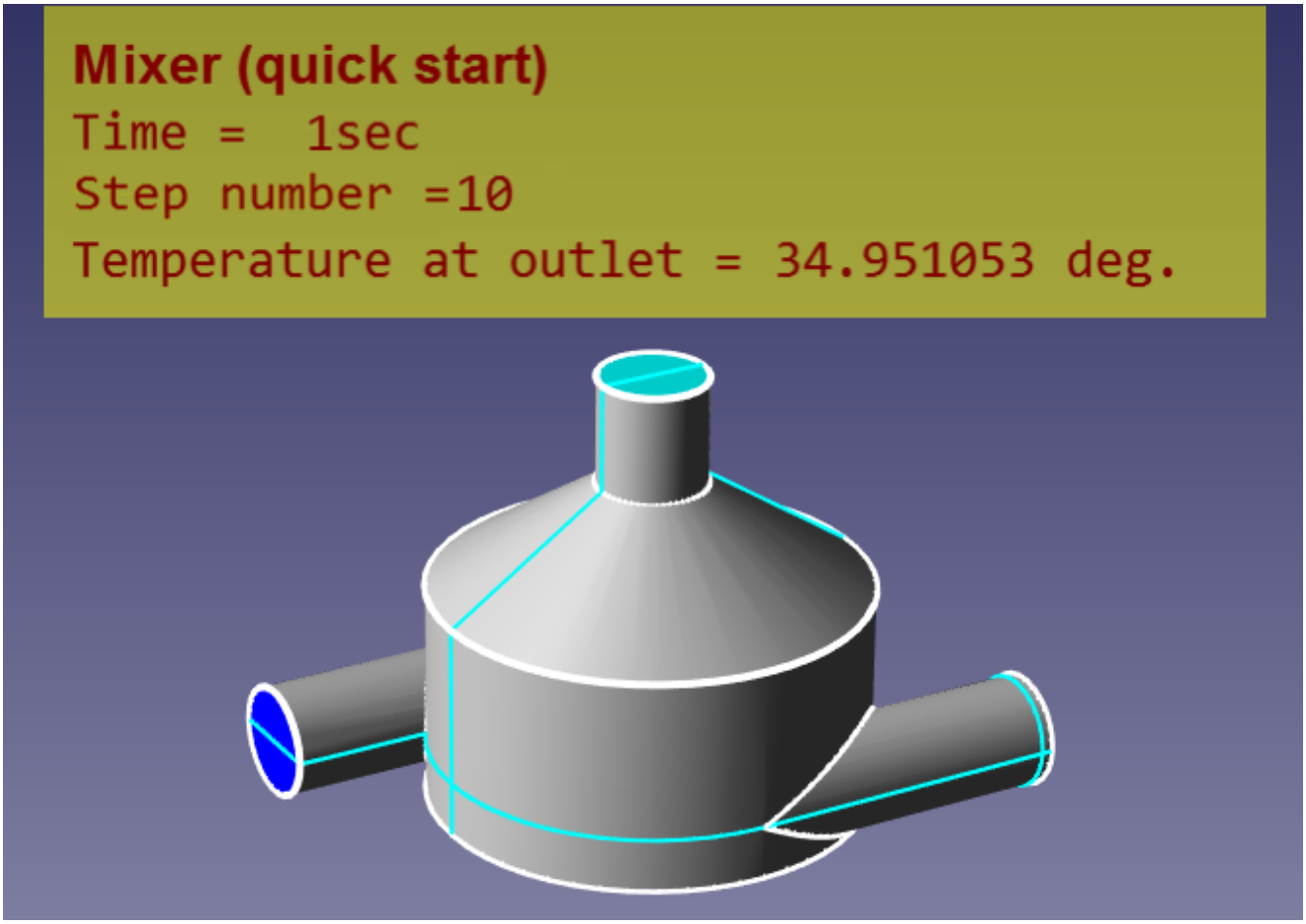
Specifying background of the **View** window by a plain color



Specifying background of the **View** window by a color gradient (the gradient is a smooth transition between two colors)

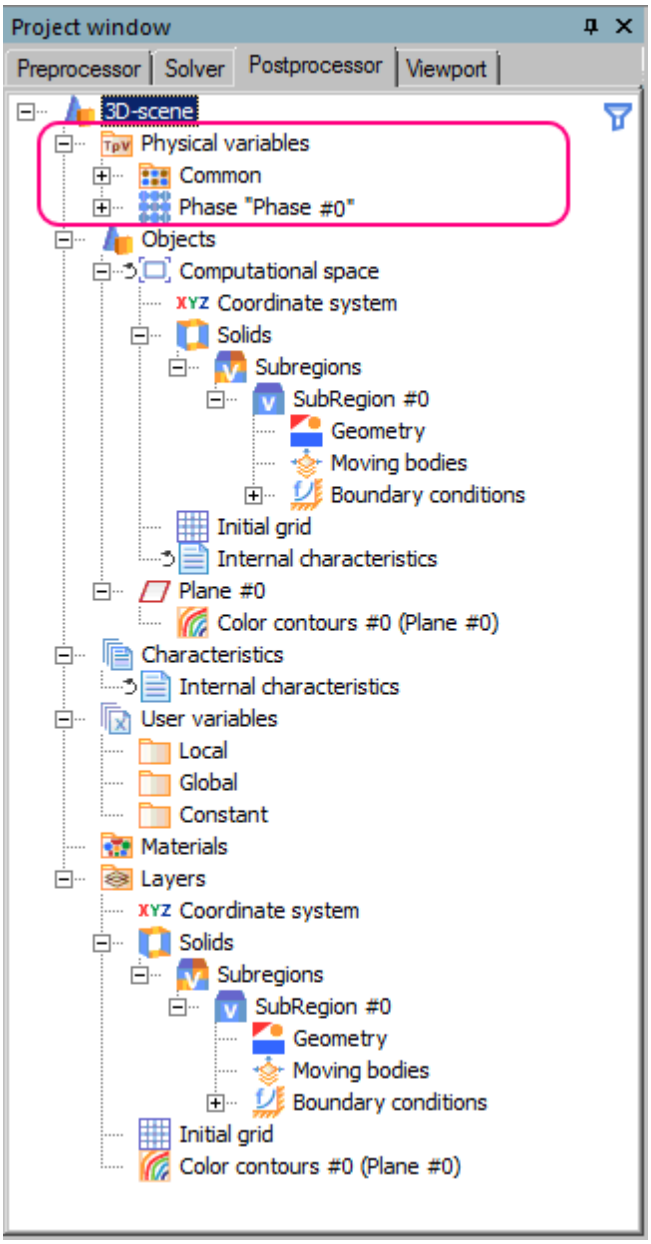


Specifying background of the **View** window by a **Palette**



Example of a title in the **View** window. The title's format is defined by parameters **Title > ...** in properties of the **3D-scene** root folder.

8.1.8.5.1 Folder «Physical variables»



The **Physical variables** folder contains the following subfolders:

- **Common**
- **Phase "Phase #N"**

Folder Physical variables> Common

The folder **Physical variables> Common** is used to display a list *of all the physical variables* that are included in any model specified in any of the **Phases**.

Thus, if in several phases identified the same variables in a folder named **Shared variables** will match one variable. When rendering the value of such a variable in several phases, each phase will be held on the visualization of the variable corresponding to this phase.

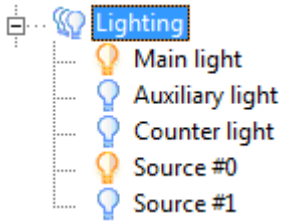
Variables are added to this folder automatically after specifying appropriate physical processes in **Phases**. The variables are also automatically removed from the folder after excluding the appropriate physical processes from all **Phases**.

Folders Physical variables > Phase "Phase #N"

Folders **Physical variables> Phase "Phase #N"** are used to display a list of physical variables that are included in any model of a physical process defined *in this* **Phase**.

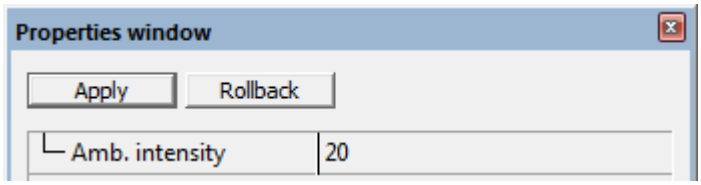
Variables are added to these folders automatically after specifying appropriate physical processes in the **Phase**. The variables are also automatically removed from the folders after excluding the appropriate physical processes from the **Phase**.

8.1.8.5.2 Folder «Lighting»

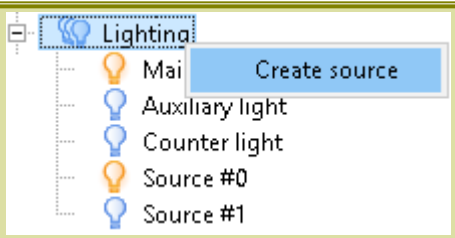


Folder "Lighting" and settings of the background lighting intensity

The **Lighting** folder contains elements that correspond to the light sources. Also, in properties of the **Lighting** folder, the background lighting intensity is specified.



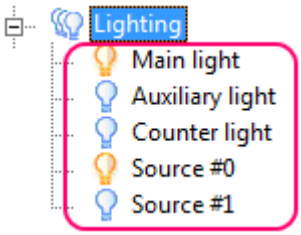
Properties of the folder "Lighting"	
Parameter	**Description**
Amb. intensity	Level of background light intensity



Context menu of the folder "Lighting"

Menu item	Description
Create source	Creating a new light source

Elements for light sources



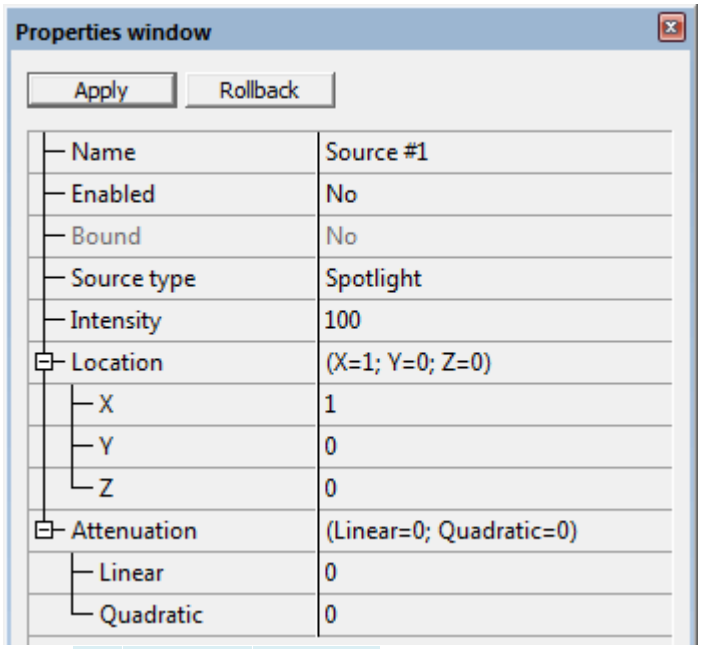
Lighting folder contains three elements for light sources, three of which - the standard and are created automatically with the project, and the rest can be created by the user.

Standard light sources are:

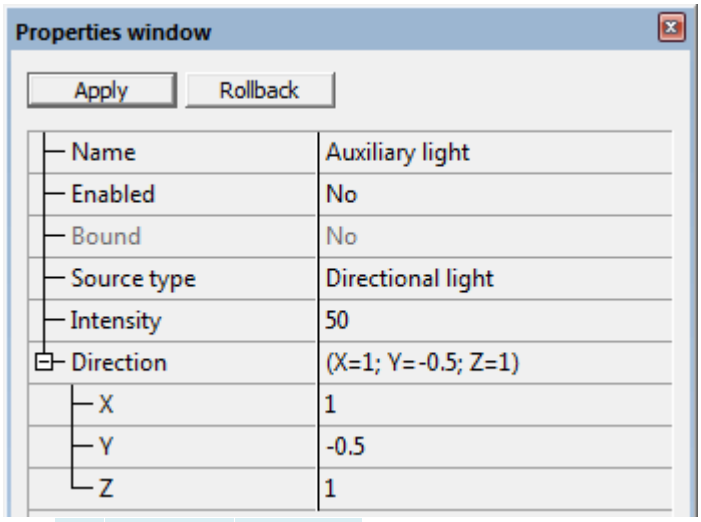
- **Main light** - a standard main light source in the scene
- **Auxiliary light** -standard additional light source on the scene
- **Counter light** - oncoming light source located behind the object (the light is directed towards the observer)

Light sources that are created by the user to give standard names **Source #N**, N = 0, 1, ..., which you can change.

Light parameters are displayed in the windows of their properties.



The **Properties** window of a point source of light

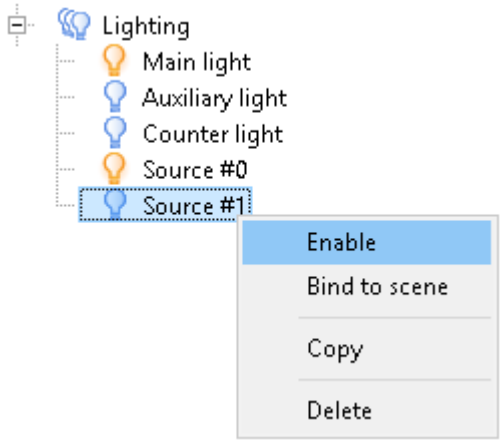


The **Properties** window of a directional light source

Parameters of element **Light Source**:

Parameter	Description
Name	The name of the light source. This option allows you to change the default name of the **main** light, **extra** light, **Backlight** or **Source #N** (N = 0, 1, ...).
Enabled	Whether the light source. Possible options are: • **No**- the source is turned off (in the project tree, its icon looks like this:) • **Yes**- the source is included (in the project tree, its icon looks like this:)
Bound	Possible options are: • **Yes**- a source attached to the stage, i.e. stationary with respect to the computational domain

Parameter	Description
	• **No**- source tied to the observer, i.e. stationary relative to the observer This parameter can not be changed in the **Properties** window to change it used command from the context menu **Link to the scene**.
Source type	Type of light source. Possible options are: • **Spotlight** • **Directional light**
Intensity	The intensity of the light source in %
Location > X **Location > Y** **Location > Z**	The position of a point light source. Is given in absolute coordinate system (if **Bound = Yes**) or in the coordinate system associated with the observer (if **Bound = No**). (Options are available only if **Source type = Spotlight**)
Attenuation > Linear	Coefficient that determines the linear attenuation of light depending on the distance from the source. (Option is available only if **Source type = Spotlight**)
Attenuation > Quadratic	Coefficient determining the quadratic light attenuation depending on the distance from the source. (Option is available only if **Source type = Spotlight**)
Direction > X **Direction > Y** **Direction > Z**	The components of the direction vector along which the light propagates toward the source. (Options are available only if **Source type = Directional light**)



Context menu of element **Light Source**

Context menu of element **Light Source**:

Menu item	Description
Enable	- the light source is disabled - the light source is enabled
Bind to scene	- the light source is bound to the observer - the light source is bound to the scene
Copy	Creating a **Light source**, which is a copy of the selected source
Delete	Deleting the selected **Light source**

Examples of actions

Creating a light source:

Step	Actions
1	Open the context menu of the **Lighting** folder and select **Create source**.
2	The created new **Light source** will be added to the folder **Lighting**.

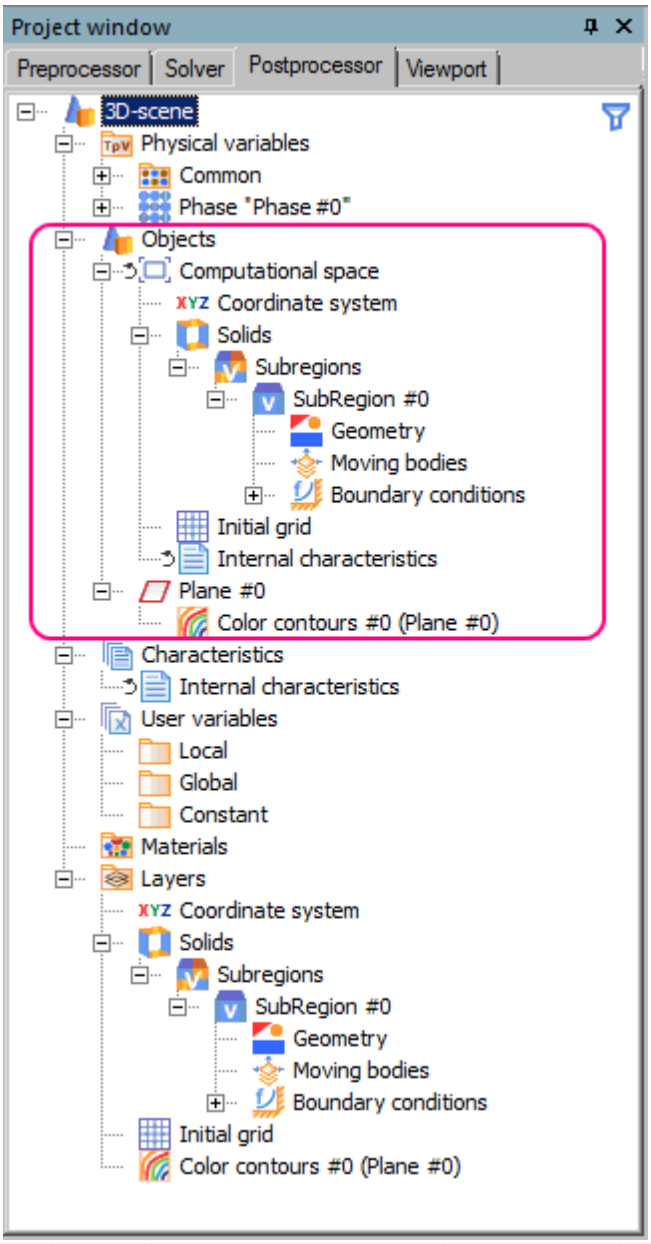
Copying a light source:

Step	Actions
1	Open the context menu of a **Light source** in the **Lighting** folder and select **Copy**.
2	A copy of the **Light source** will be added to the folder **Lighting**.

Deleting a light source:

Step	Actions
1	Open the context menu of the **Lighting > Light source** element, which is to be deleted, and select the **Delete** command.
2	The selected **Light source** will be removed from the project tree.

8.1.8.5.3 Folder «Objects»



Folder **Objects** in the project tree (in the **Postprocessor** tab)

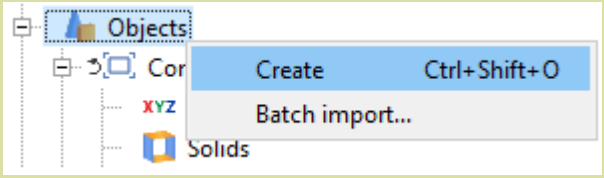
Objects are geometric shapes that are the basis for the creation of other elements (in **Postprocessor** for **Characteristics** and **Layers**). Objects of **Postprocessor** are defined in folder **Objects**.

The **Objects** folder contains:

- the **Computational space** folder
- folders **(Object) #N** that correspond to geometric objects

The **Objects** folder is included into two tabs in the project tree, **Preprocessor** and **Postprocessor**, so its description is given in a separate section, "Objects in the project tree."

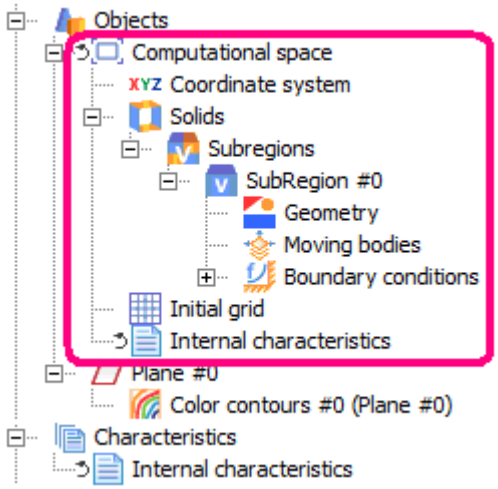
Context menu of the «Objects» folder



Context menu of the «Objects» folder

Command	Description
Create	Creating a new **Object**, or loading one **Imported object**. This command is duplicated by the **Ctrl+Shift+O** hot key.
Batch import	Loading multiple **Imported objects**.

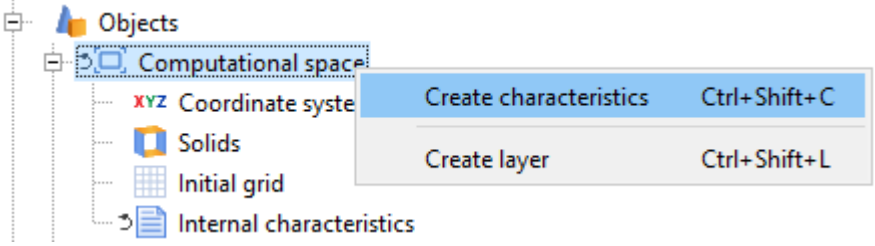
Folder «Objects > Computational space»



Object **Computational space** - a unique object that is created along with the creation of the project, not deleted and can not be edited because it does not have any properties. This object does not impose restrictions on the spatial rendering layer, the layer itself determines where and what to draw him.

Folder **Computational space**in the **Postprocessor** contains:

- element **Coordinate system** - duplicate element that defines the image of the absolute system, see section Layer «Coordinate system», user interface
- **Solids** folder
- **Initial grid** element
- element **Internal characteristics**
- elements **Characteristics #N**
- elements ***(layer)* #N** (element name is formed from the name of the type layer and a number **#N**, N = 0, 1, ..., **Streamlines #N** element has child elements **"Source #N** ...")

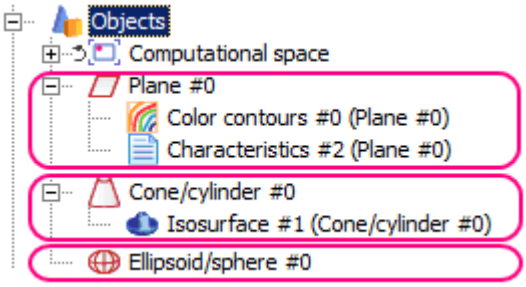


The context menu of the folder "Objects > Computational space":

Command	Description
Create characteristics	Create **Characteristics** related to the **Computational space** (a new element **Characteristics #N** will appear in the folder **Objects > Computational space**). This command is duplicated by the **Ctrl+Shift+C** hot key (by default; you can change this hot key).

Command	Description
Create layer	Create a new item **(Layer) #N** in the folder **Objects > Computational space**. This command is duplicated by the **Ctrl+Shift+L** hot key (by default; you can change this hot key).

Folders «Objects>(Object) #N»



Folders **Object s> (Object) #N** are used to store data about **Objects** of the various types.

There are the following types of objects:

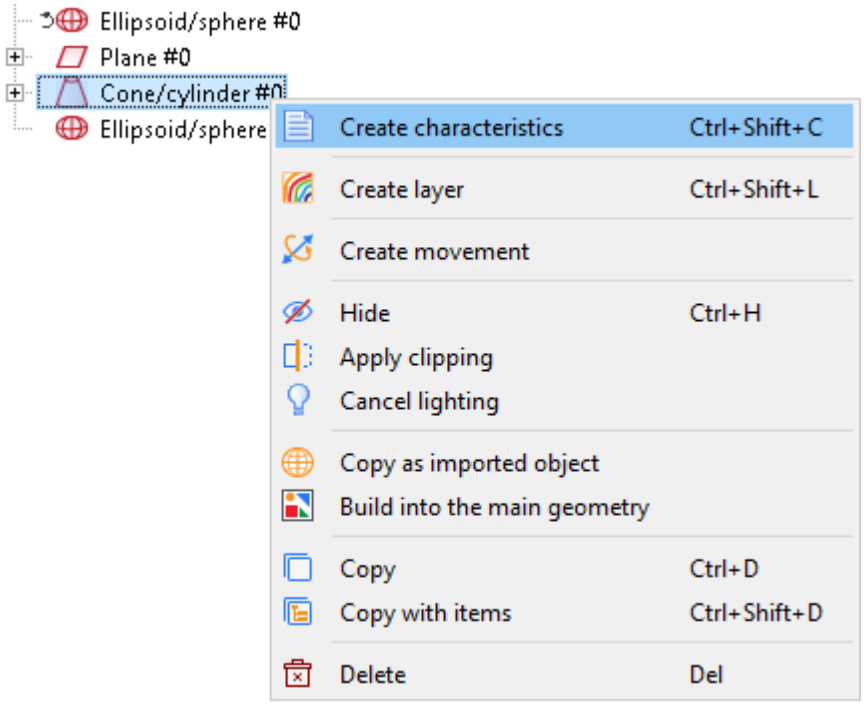
- **Line**
- **Plane**
- **Box**
- **Cone/cylinder**
- **Ellipsoid/sphere**
- **Imported object**
- **Supergroup**

Create and delete **(Object) #N** carried out using the **Create** and **Delete** context menu. Also **Objects** created in the **Preprocessor** tab in the folder **Region> Objects** are displayed; such **Objects** are marked in the project tree with sign "↻" (you can delete such **Objects** from the **Preprocessor** tab only).

Creating a new object consists of two stages:

- in a window, which opens by the context menu command, select a type of the geometric object
- then specify parameters of the created **Object** in its **Properties** window.

Context menu of an Objects in the Postprocessor tab in the project tree



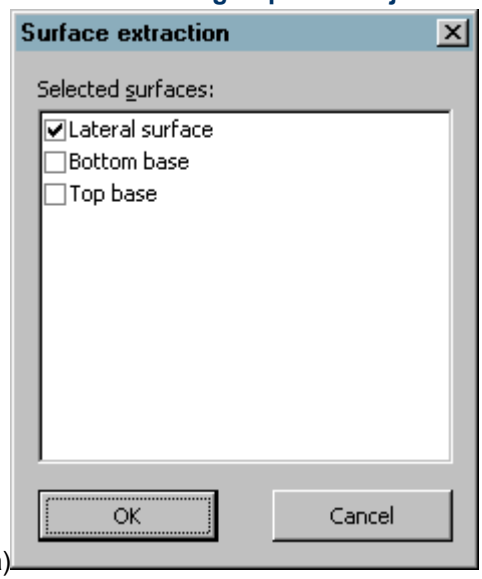
*(list of the context menu commands depends on the current settings and on the type of the **Object**)*

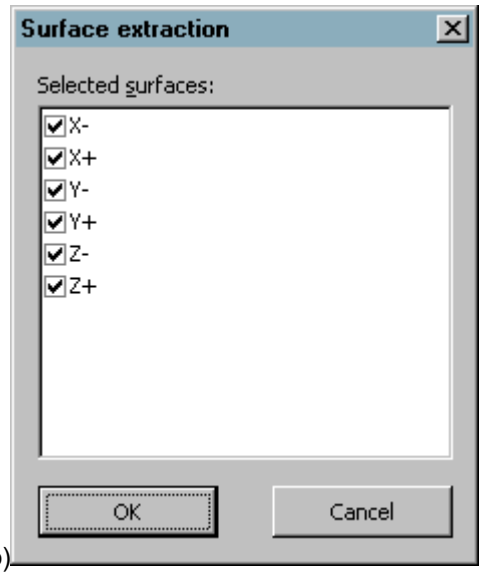
Command	Action
Create layer	Create a **Layer** based on the selected **Object**. This command is duplicated by the **Ctrl+Shift+L** hot key (by default; you can change this hot key).
Create characteristics	Create **Characteristics** based on the selected **Object** (available for all objects except **Lines**). This command is duplicated by the **Ctrl+Shift+C** hot key (by default; you can change this hot key).
Create movement	Create **Movement**. The movement of an **object** created in the **Preprocessor**,you can specify only the **Preprocessor** (so for these **Objects** in the **Postprocessor** tab, this command is absent). The movement can be set on a **Box**, **Cone/cylinder**, **Ellipsoid/sphere** and **Imported objects** (if its base is not created **Moving** body),so for other types **of objects**, this command is not available.
Hide	Makes the **Object** hidden or visible.
Show	A hidden **Object** is always displayed in the **View** window. A visible **Object** is displayed in the **View** window, only when it is selected in the project tree This command is duplicated by the **Ctrl+H** hot key (by default; you can change this hot key).
Apply clipping	Makes the **Object** be able to be clipped.
Cancel clipping	When an **Object** is not able to be clipped, clipping **Planes** do not affect its image. When an **Object** is able to be clipped, clipping **Planes** cut its image.
Apply lighting	Make the **Object** be lit or not be lit by light sources.
Cancel lighting	
Use as clipping object*)	Property of a **Plane** to be a *secant object* means that the this **Plane** will cut off in the **View** window all **Objects** and **Layers** that have the **Clipped**

Command	Action
Do not use as clipping object[*)]	property enabled. Depending on the value of this parameter, the **Plane**'s icon in the project tree in the **Postprocessor** tab is: • - for a **Plane**, which *is not* a clipping object • - for a **Plane**, which *is* a clipping object See illustration in the section Specific settings of the Plane object.
Use as mirror[*)] **Do not use as mirror**[*)]	**Plane** property be *mirrored object* means that the plane will be reflected in its negative half-images, which are located within the computational domain in the positive half (half-subspace, in which the vector normal to this **plane**). See illustration in the section Specific settings of the Plane object.
Copy	Creating an element, which is a copy of the selected element
Copy with items	Copying the selected **Object** with its child elements (available only in **Postprocessor**).
Copy as imported object	Create **Imported object** based on the selected **Object** (only available for objects with finite area). When you create a window **to import objects Copying surfaces** (see description below).
Built into the main geometry	Create a surface area calculated on the basis of the selected object (only available in the **Postprocessor** for objects with finite volume, except **Supergroup** and **Imported object**, on which a **Moving body** is created).
Delete	Deleting the selected element from the project tree If the **Object** has been created in the **Preprocessor** (designated "↻"), Then it can be removed only by the **Preprocessor**, so the **objects** created in the **Preprocessor**, this command is not available.

[*)] The command is present in the menu if the **Object** is a **Plane**.

Copying surfaces when creating imported objects

a) 

b) 

Examples of **Surface extraction** dialog boxes: a) for **Cylinders** and truncated **Cones**, b) for **Boxes**. When copying, you can select any or all of the **Object**'s surface.

The **Surface extraction** dialog box opens when you create an **Imported object** based on another **Object**. The window contains a list of the surfaces of the original object, which will be included in the **Imported object**.

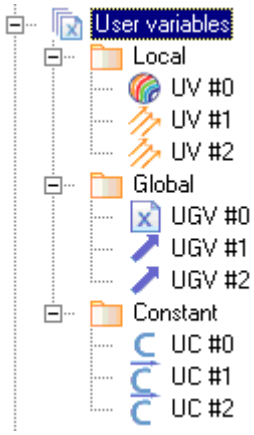
8.1.8.5.4 Folder «Characteristics»

Folder **Characteristics** is presented in two tabs of the **Project** window, **Preprocessor** and **Postprocessor**.

See its description, which has been given below.

8.1.8.5.5 Folder «User variables»

User variables are created by the user based on **Physical variables** and **Constants**. **User variables** can be created in both **Preprocessor** or **Postprocessor**.



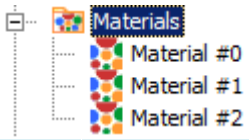
Folder **User variables** is presented in two tabs of the **Project** window, **Preprocessor** and **Postprocessor**. See its description, which has been given below.

8.1.8.5.6 Folder «Materials»

Materials - the elements in the tree **Postprocessor** intended to set parameters for displaying solid objects.

The following describes the following folders and elements of the project tree:

- folder **Materials**
- elements **Material #N**



The **Materials** folder in the project tree

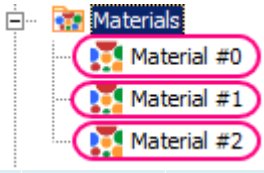
Folder «Materials»

The **Materials** folder contains items **Material #N**.

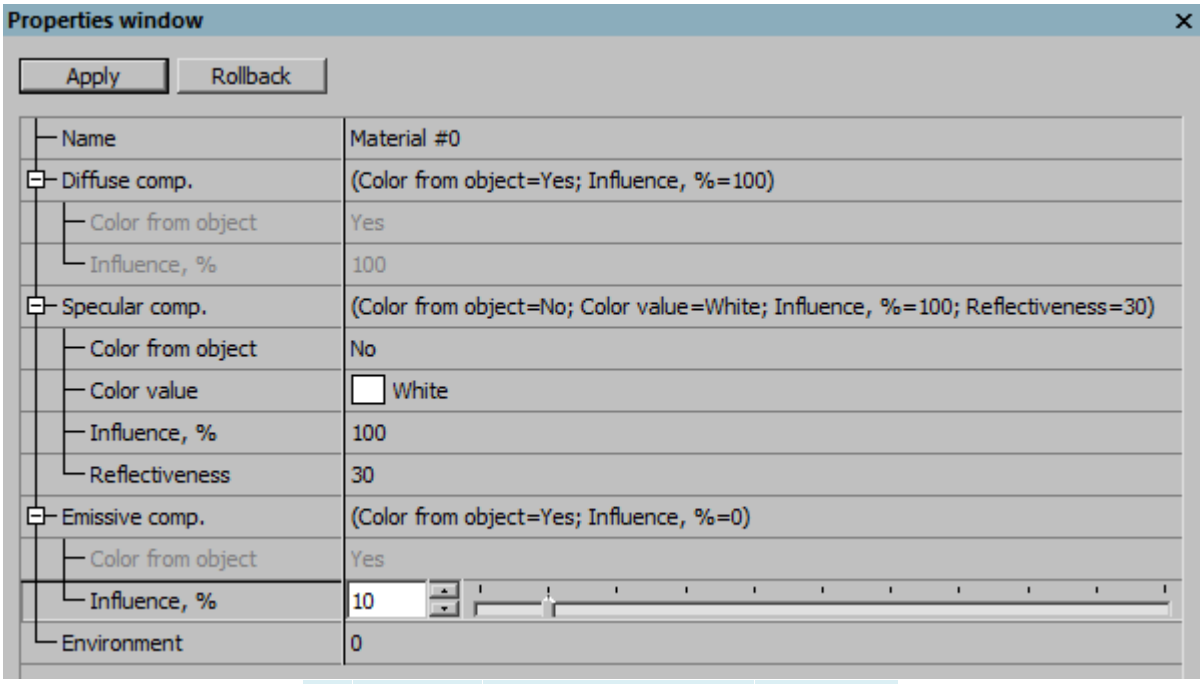
To add a new item **Material #N** to the folder **Materials**, open the context menu of a folder **Materials** and select the **Create** command.

Element «Material #N»

The **Material #N** element is designed to store the display of a "material", which determines the optical properties of the surface.



Elements **Material #N** in the project tree

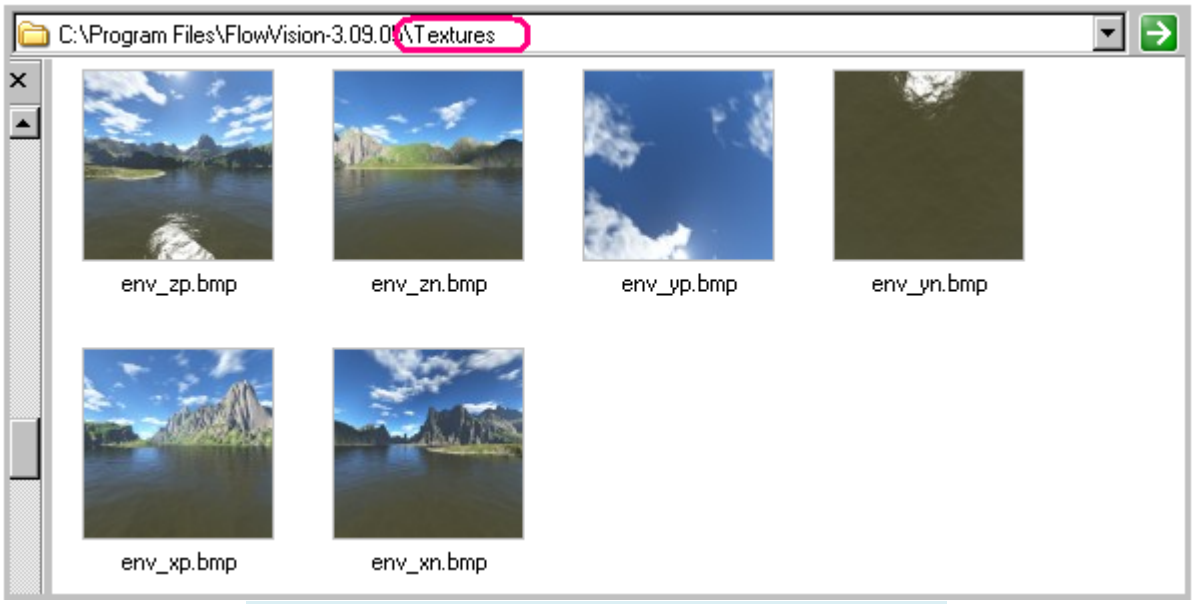


The **Properties** window of the element **Material #N**

Parameters of the element **Material #N**:

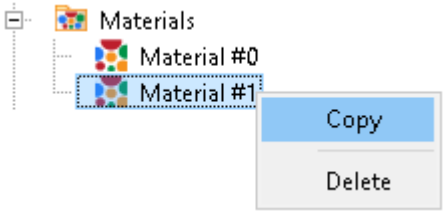
Parameter	Description
Name	Material name (the default name is **"Material #N"**, N = 0, 1, 2 ...)
Diffuse comp.	Parameters of diffuse scattering of incident light on the object surface. Information field, not editable
Diffuse comp. > Color from object	Use the color of the **Object** to diffuse scattering. This field is read-only and can not be changed here.
Diffuse comp. > Influence, %	The contribution of diffuse scattering in the visualization of the surface of the object. The degree of influence of the diffuse component is equal to 100% minus the degree of influence of the radiative component.
Specular comp.	Specular reflection of light on the subject (glares)
Specular comp. > Color from object	Possible options are: • **No** - the color of the specular reflection (glare) is given by the parameter value of the color • **Yes** - the color of the specular reflection (glare) = color of the object
Specular comp. > Color value	The choice of color of specular reflection (glare). Available only if the **Color from object = No**.
Specular comp. > Influence, %	The contribution of the specular reflection (glare) of light from the surface, %
Specular comp. > Reflectiveness	Specular reflection, in %
Emissive comp. > Color from object	When this parameter is **Yes**, it means that the color of the intrinsic emission is the color of the **Object**. This field is read-only and can not be changed here.
Emissive comp. > Influence, %	The contribution of self-radiation in the light from the surface, in %
Environment	Coefficient of reflecting the images of the outer world (the images from the inner surface of the «outer box»), it is specified in percents. Images of the outer world, which can be visible in reflecting surfaces, are stored in files **`env_xp.bmp`**, **`env_xn.bmp`**, **`env_yp.bmp`**, **`env_yn.bmp`**,

Parameter	Description
	env_zp.bmp, **env_zn.bmp** in the subdirectory **Textures**, which locates in the same directory as **Pre-Postprocessor**.



Images of the outer world are stored in the directory Textures

Context menu of the element «Material #N»

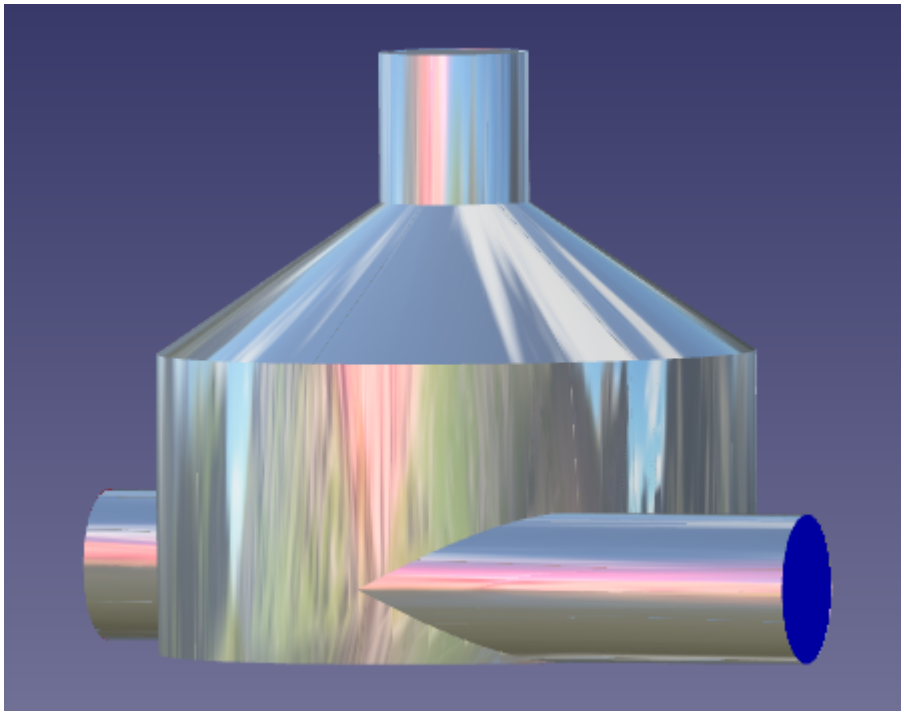


Context menu of the element **Material #N** in the project tree

Context menu of the element **Material #N** in the project tree:

Menu item	Description
Copy	Creating an element, which is a copy of the selected element
Delete	Deleting the selected element from the project tree

Illustration



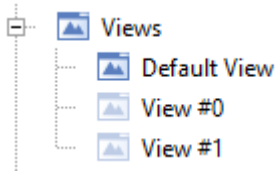
Specifying the properties of the material can be obtained high gloss surfaces and reflections of a given color (in the example in the figure - red)

8.1.8.5.7 Folder «Views»

The **Views** elements allow you to store (within a project) and reuse the following settings of the **View** window:

- location the viewpoint ("camera")
- scale of the scene
- lighting
- settings of the background
- properties of **Layers** that are included in the project (such as ranges, palettes, and parameters **Visible** and **Clipped**)
- properties **Visible**, **Clipped**, and **Clipping object** of the **Objects** in the project
- properties of the displayed image that are specified by buttons in the toolbars **Rendering** and **Solids**, and also states of these buttons in the toolbars
- settings of the **Title**

Settings of the camera can be also saved in a file and downloaded from a file.

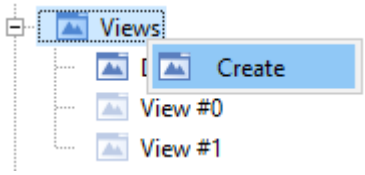


The **Views** folder in the project tree

One **View** is default one, its name is **Default view** and cannot be changed. This **View** is initially and always presented in the project tree and cannot be removed.

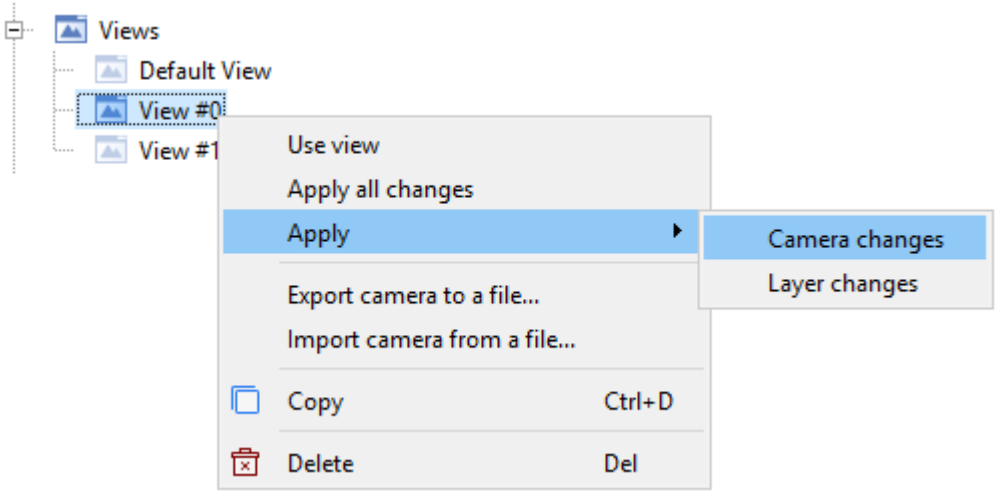
Optionally you can create other elements **View #N** and use them when required.

Context menu of the folder "Views"



The context menu of the folder **Views** contains the **Create** command only, which allows you to create a new **View #N** element.

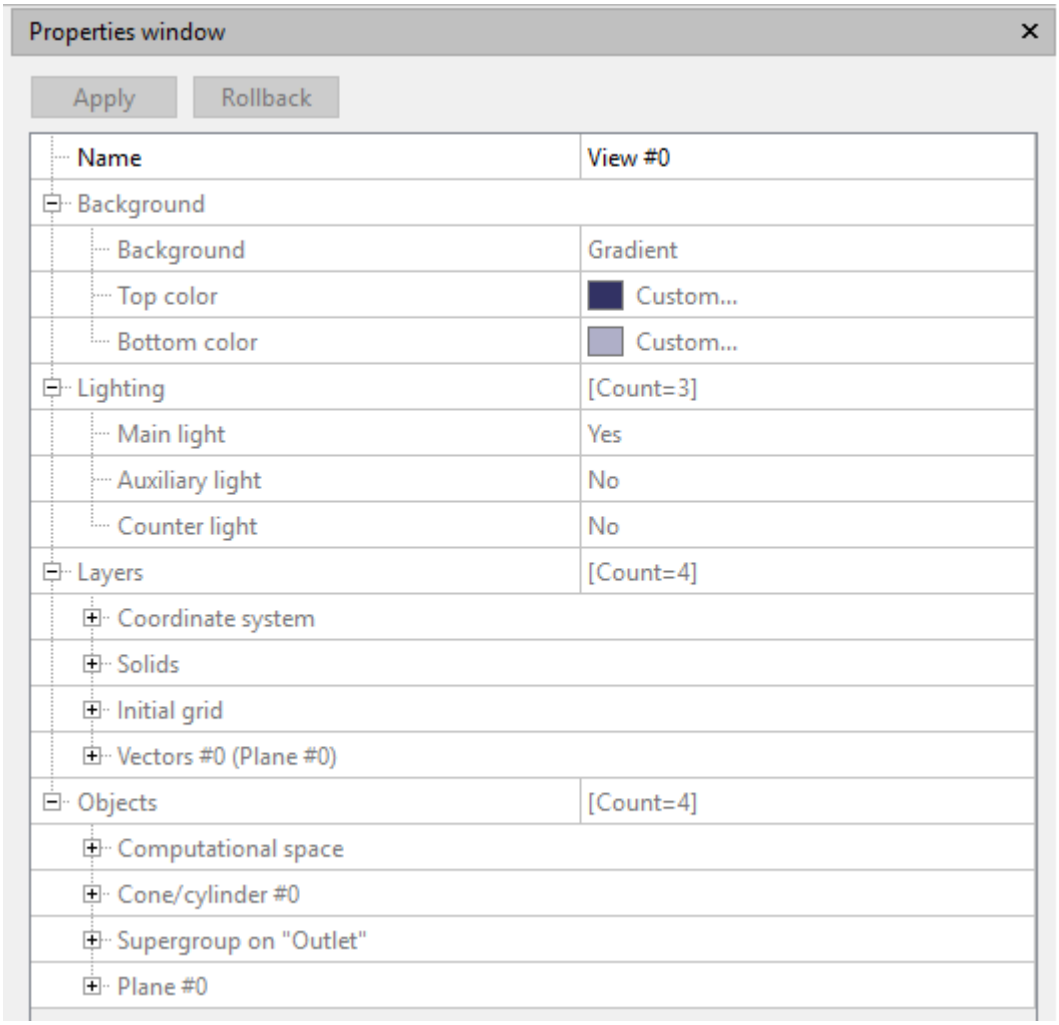
Context menu of a «View» element



Context menu of the **View** element

Menu item	Description
Use view	Use the selected **View** to display the scene in the **View** window. If parameters of this **View** differ from the current settings of the **View** window, then images in the **View** window will change according to the settings that are saved in the selected **View**. The icon of this **View** will be displayed bright in the project tree, while icons of other (unused now) **Views** will be dimmed.
Apply all changes	Saving in the selected **View** all current settings of the **View** window, so all your changes of the settings will be saved in this **View**.
Apply > Camera changes	Saving in the selected **View** those changes only that relate to the camera's location, scale, and settings of the **Rendering** and **Solids** toolbars. This command is not available for the **Default view**.
Apply > Layer changes	Saving in the selected **View** those changes only that relate to **Objects** and **Layers** containing in the project. This command is not available for the **Default view**.
Export camera to a file	Saving settings of the camera in a file
Import camera from a file	Importing (downloading) settings of the camera from a file
Copy	Create a new element **View #N**, which will be a copy of the selected **View**. This command is not available for the **Default view**.
Delete	Delete the selected element **View #N** from the project tree. This command is not available for the **Default view**.

Parameters of the «View» element

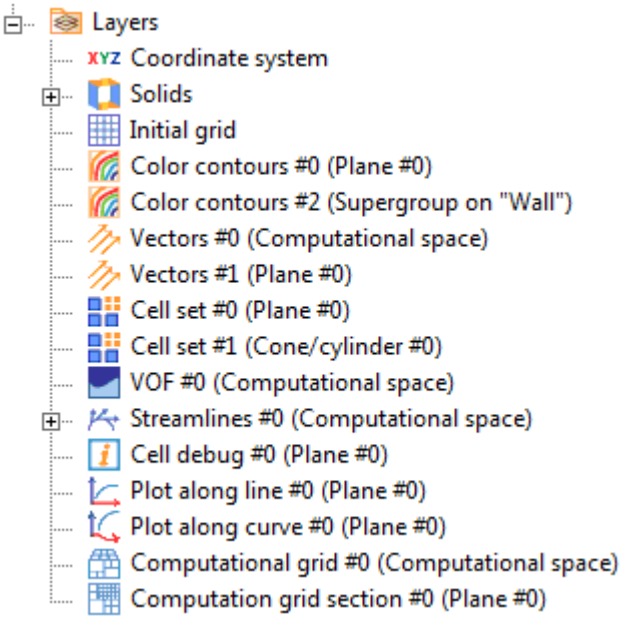


The **Properties** window of the **View** element

All properties of the **View** element, except its **Name**, are read-only. Values of these parameters are displayed for your information, and they can be changed in properties of other elements of the project tree.

Parameters of the "View" element	
Parameter	**Description**
Name	Name of the **View** element (when a new **View** is created, it is named as **View #N**. This name can be changed if required. This parameter is not available for the **Default view**.
Background > ...	Settings of the background of the **View** window
Lighting > ...	Settings of lighting in the **View** window
Layers > ...	List of displayed **Layers**, which are presented in the project, and state of their properties **Visible** and **Clipped**.
Objects > ...	List of displayed **Objects**, which are presented in the project, and state of their properties **Visible**, **Clipped**, and **Clipping object**.

8.1.8.5.8 Folder «Layers»



Folder **Layers** in the project tree

The **Layers** folder in the project tree can contain the following elements:

- layer **Coordinate system** (always presented in the **Layers** folder)
- layers folder **Solids**
- layer **Initial grid**
- layer **Computational grid**
- layer **Computational grid section**
- layer **Color contours**
- layer **Vectors**
- layer **Plot along line**
- layer **Plot along curve**
- layer **Plot along ellipse**
- layer **Distributed characteristics**
- layer **Isosurface**
- layer **Streamlines** and element **Emitter for streamlines**
- layer **Nodal loadings**
- layer **VOF**
- layer **Cell set**
- layer **Cell debug**
- layer **Volume visualization**

Creating a new layer consists of two steps:

- from the context menu invoked from the dialog box, choose the type of the layer (from **Layer**) and the type of geometric object (from **Objects**),
- then in the **Properties** window, select the variable layer being created and edited in the other parameters.

Layers with the calculated data are intended for:

- displayed in the**View**window distribution calculated variable (for a layer with information about the calculation of the variable) or computational grid / cell (for the layer with the data on the computational grid cells);
- displayed in the **Info** window data characterizing the distribution of the variable computed in the calculation of the project (for a layer with information about the calculation of the variable) or data on the computational grid or cell (for the layer with the data on the computational grid cells).

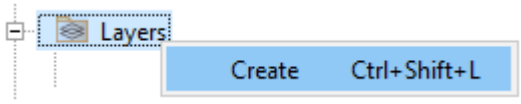
Changing the image of some strata defined by their parameters in the **Properties** window, there is only the next time the project is saved andwhen communication with the**Solver.**
(For the construction of the new image after clicking the **Apply** button in the **Properties** window, click the button  in the **Standard** toolbar or use the **File > Save** command from the Main Menu. Also required due to the **solver** to obtain the data necessary for the construction of the image).

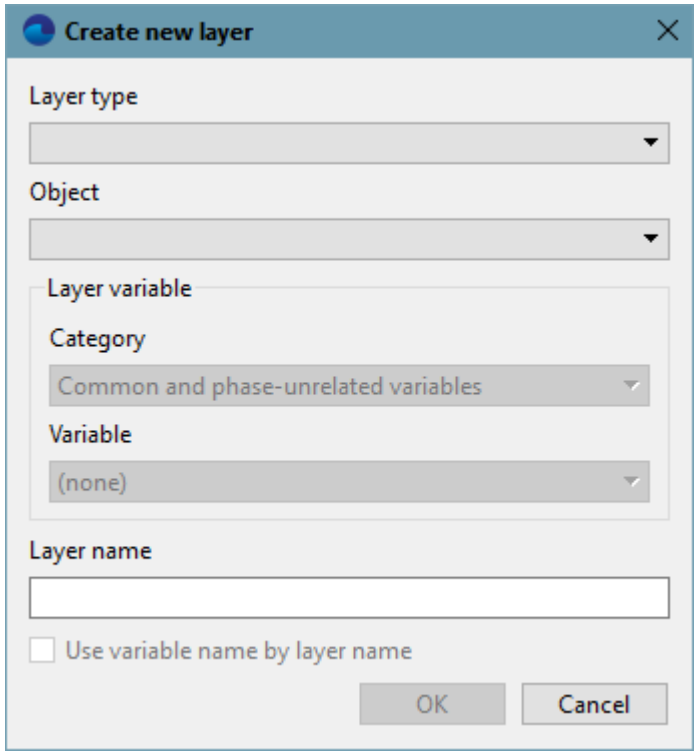
Parameters of Layers, which are same in different types of Layers

The **Properties** window of a **Layer** contains a set of parameters, which depends on the type of the **Layer** and the type of the **Object**, on which the **Layer** is built. Many of these parameters are the same in all or many of types of **Layers**.

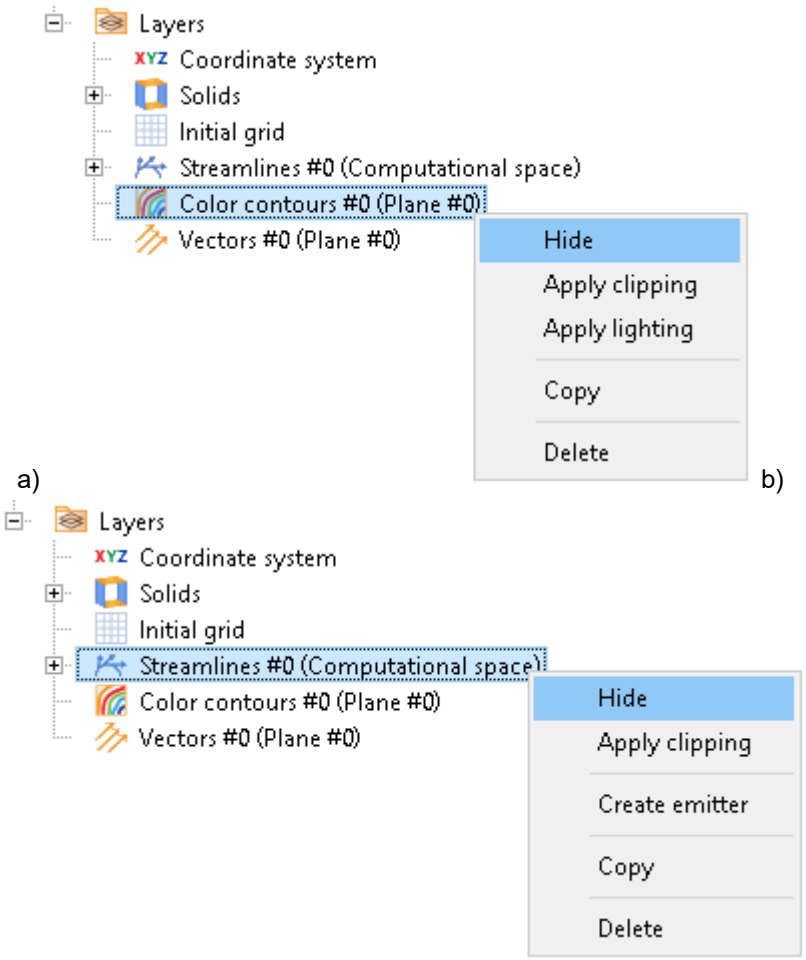
To avoid duplications of the text and for ease of reference, these parameters are described in a separate section, General properties of Layers.

Context menu of the folder "Layers"



Menu item	Description
Create	This command creates a new **Layer** and adds it into the **Layers** folder. The **Create new layer** dialog box will open:  In this dialog box you specify: • Type of the **Layer** • The **Object**, on which the layer will be created • **Category** of the **Variable**, by which the **Layer** will be built, and the **Variable** itself (but this is not applicable for some types of **Layers**). Options for selection the **Category** are: **Common and phase-unrelated variables | Variables of phase "*Phase_name*" | User variables**. For **Vectors** layers the program will automatically prompt you to select the **Velocity** variable (you can select another variable if required). • The **Layer name** (you can specify here a name of the **Layer** instead of the default name). If you select the **Use variable name by layer name** checkbox, the program will prompt the **Layer**'s name based on name of the **Variable**, by which the **Layer** will be built. When an alternative name of the **Layer** is being entered, the "**×**" symbol will appear on the right; clicking it restores the default name. This command is duplicated by the **Ctrl+Shift+L** hot key (by default; you can change this hot key).

Context menu item "Layer" with the calculated data

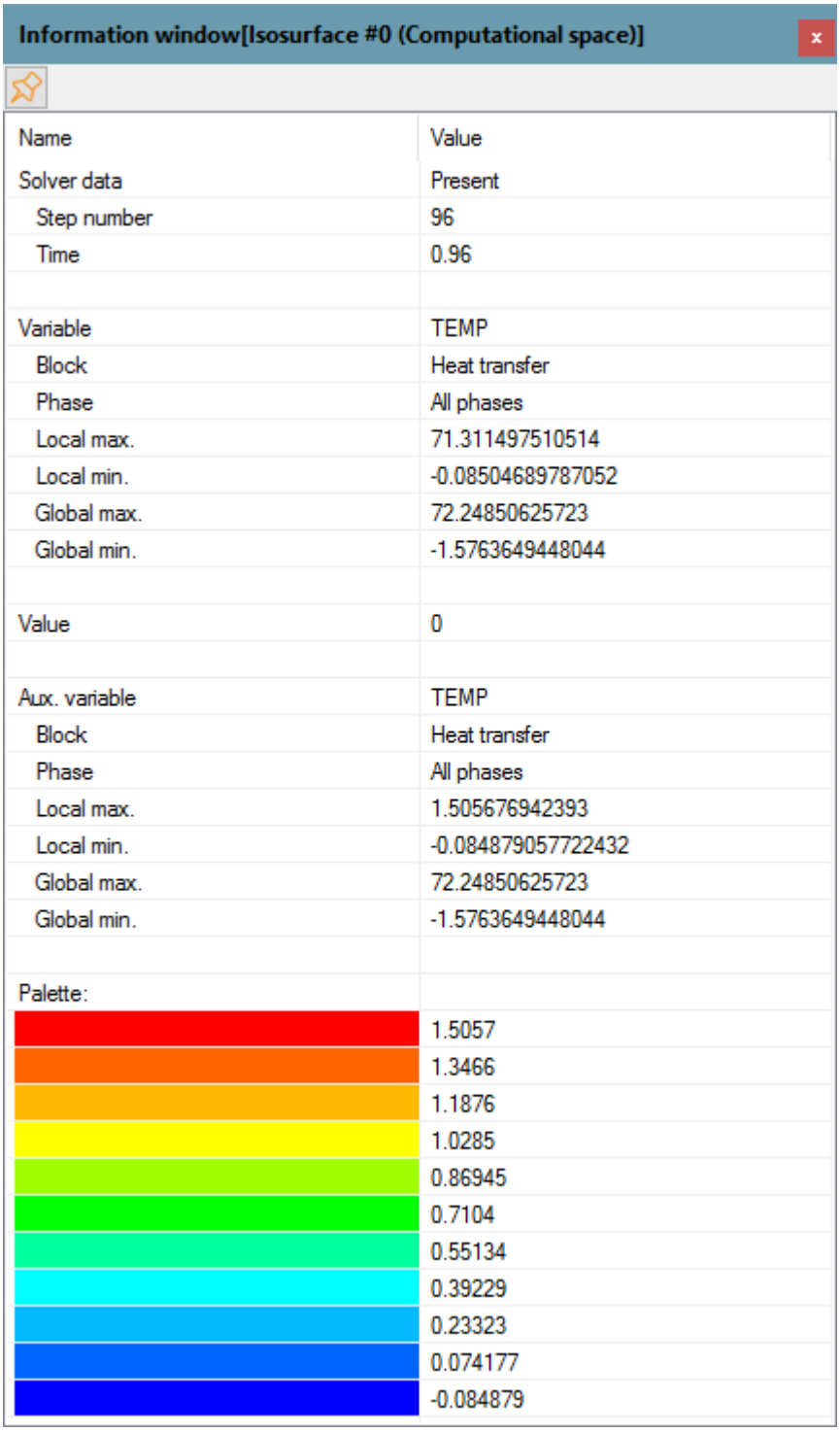


Context menu of the element **Layer** with calculated data in the project tree:
a) for **Layers** except **Streamlines**, b) for **Streamlines**

Context menu of the item **Layer** with calculated data in the project tree:

Menu item	Description
Hide	Make a **layer** invisible (except in cases when it is selected in the project tree). ☐ - the **Layer** is always displayed in the **View** window ☑ - the **Layer** is displayed in the **View** window, only when it is selected in the project tree
Apply clipping	Make the **Layer** be able to be clipped. ☐ - clipping **Planes** do not affect the **Layer** ☑ - clipping **Planes** cut the **Layer**
Apply lighting	Make the **Layer** be lit ☐ - the **Layer** is *not* lit by light sources ☑ - the **Layer** is lit by light sources
Create emitter	Creating a new **Emitter for streamlines** element. (This item is present in the context menu only layer **Streamlines**)
Copy	Create a new layer, which is a copy of the selected layer
Delete	Removing a layer of wood project

Window «Info» for layers



Name	Value
Solver data	Present
Step number	96
Time	0.96
Variable	TEMP
Block	Heat transfer
Phase	All phases
Local max.	71.311497510514
Local min.	-0.08504689787052
Global max.	72.24850625723
Global min.	-1.5763649448044
Value	0
Aux. variable	TEMP
Block	Heat transfer
Phase	All phases
Local max.	1.505676942393
Local min.	-0.084879057722432
Global max.	72.24850625723
Global min.	-1.5763649448044
Palette:	
	1.5057
	1.3466
	1.1876
	1.0285
	0.86945
	0.7104
	0.55134
	0.39229
	0.23323
	0.074177
	-0.084879

In window **Info**, open the layer with the calculated data are displayed

- data describing the distribution of the variable calculated in the calculation of the project (for a layer with information about the design of the variable);
- data characterizing a computational grid or cell (for a layer of data cells of the computational grid).

For layers **VOF**, **Computational grid section** and **Nodal loadings Info** window is not displayed.

If there is a connection with the **Solver** window displays the data layer, updated at the end of each time step. These are grouped in blocks forming a two-tier tree structure.

If an additional variable, which is performed by the values of fill, is given, then it displays the same data block **Variable** and **Palette** for the primary variable.

The data displayed in the window **Info** for most layers:

Parameter	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current time
Variable	Name rendered variable
Variable > Block	Block of equations to which the variable
Variable > Phase	Phase to which the variable
Variable > Lock. maximum	The maximum (minimum) value of a variable volume or on the surface of the **Object**
Variable > Lock. at least	
Variable > Globe. maximum	The maximum (minimum) value of the variable phase
Variable > Globe. minimum	
Mean	The average value of a variable **Layer** (in or on the **Object**'s surface)
Maximum	The maximum value of a variable **Layer** (in or on the **Object**'s surface)
Minimum	The minimum value of a variable **Layer** (in or on the **Object**'s surface)
Maximum > position of the maximum	Value of the abscissa (the layer **Plot**) at which the maximum variable
Maximum > maximum point X	Coordinates (in absolute coordinate system) point at which the maximum variable
Maximum > maximum point Y	
Maximum > maximum point Z	
Minimum > position of the minimum	Value of the abscissa (the layer **Plot**) at which the minimum variable
Minimum > The minimum point X	Coordinates (in absolute SP) points at which the minimum variable
Minimum> The minimum point Y	
Minimum > The minimum point Z	
Extras. variable	• **(No)**- does not apply coloring depending on the values of any additional variable • ***(Name of variable)***- an additional variable, the value of which is performed by coloring
Palette	Correspondence between the values of theadditionalvariable and flowers

General settings of text files

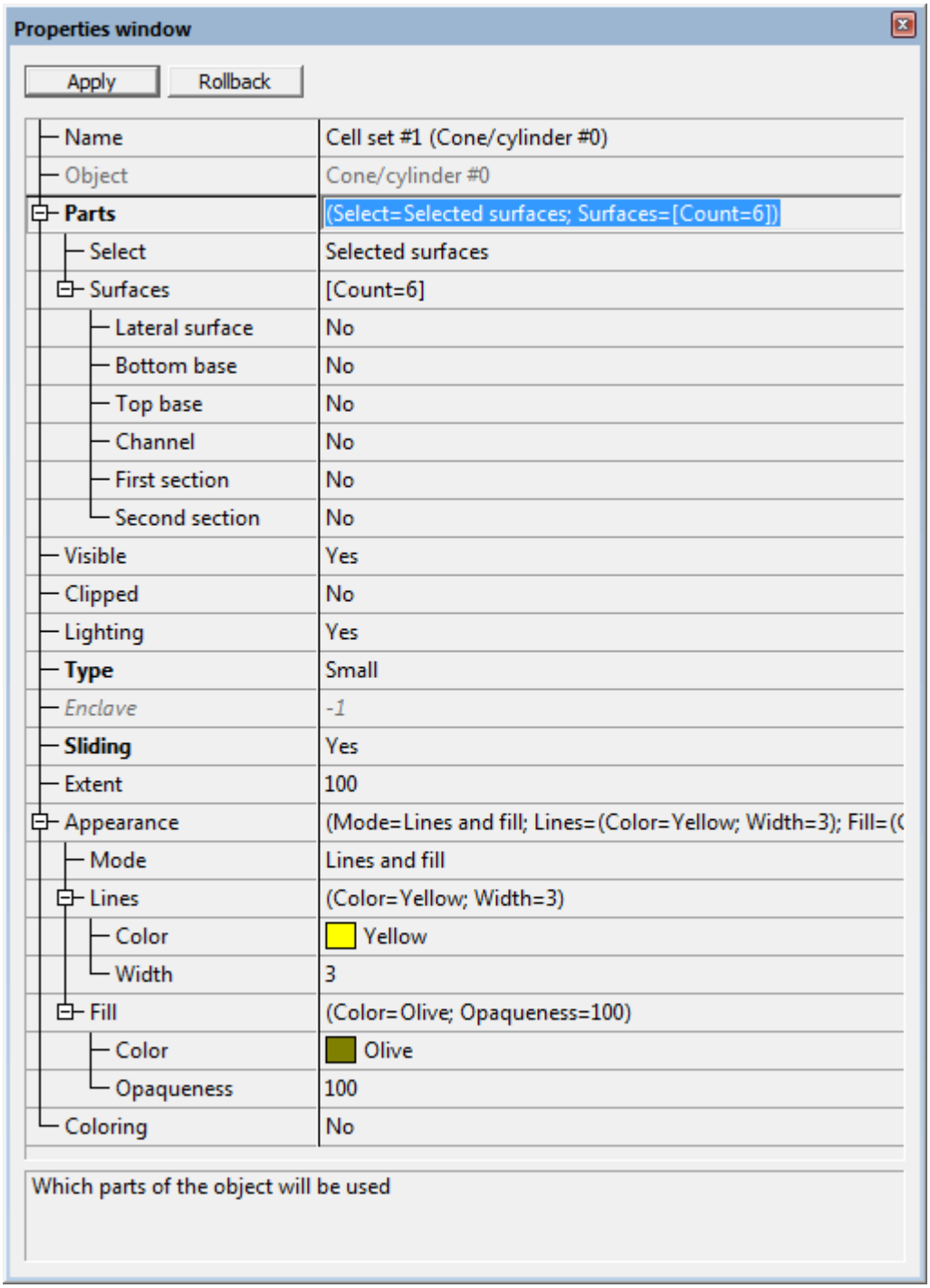
Parameter	Description
Step	Step number
Time	Time
Variable	Variable on which to build a layer
CoverVariable	Variable over which the coloring layer
NumPoints	The number of points on which construction layer
Length	The length of the plot
Avg	The average value of the variable on the plot
Min	The minimum value of the variable on the plot

Parameter	Description
MinArg	Coordinate of the lowest point on the x-axis
MinPt.x	X-coordinate of the minimum
MinPt.y	Y-coordinate of the minimum
MinPt.z	Z-coordinate of the minimum
Max	The maximum value of a variable plot
MaxArg	The coordinate on the abscissa of the maximum
MaxPt.x	X-coordinate of the maximum
MaxPt.y	Y-coordinate of the maximum
MaxPt.z	Z-coordinate of the maximum
CoverAvg	The average value of the variable in the plot coloring
CoverMin	The minimum value of the variable on the plot coloring
CoverMinArg	Coordinate of the minimum variable coloring on the x-axis
CoverMinPt.x	X-coordinate of the minimum variable coloring
CoverMinPt.y	Y-coordinate of the minimum variable coloring
CoverMinPt.z	Z-coordinate of the minimum variable coloring
CoverMax	The maximum value of the variable on the plot coloring
CoverMaxArg	Coordinate of the maximumvariable coloringon the x-axis
CoverMaxPt.x	X-coordinate of the maximum variable coloring
CoverMaxPt.y	Y-coordinate of the maximum variable coloring
CoverMaxPt.z	Z-coordinate of the maximum variable coloring
Number	Point Number
Arg	The coordinate of a point on the axis of abscissas
Value	Value of the variable at
x	X-coordinate of the point
y	Y-coordinate of the point
z	Z-coordinate of the point

8.1.8.5.8.1 General properties of Layers

In this section we describe those parameters, which have the same meaning for **Layers** of various types.

Presence or absence in the **Layer**'s properties some of these parameters depends on on the type of the **Layer** and the type of the **Object**, on which the **Layer** is built.



The **Properties** window of a **Layer** element (an example for a **Cell set** layer)

General properties of Layers[1)]	
Parameter	**Description**
Name	Name of the **Layer** (you can replace the default name, which is formed from the name of the layer's type, a serial number and the name of the object, on which the **Layer** is built).

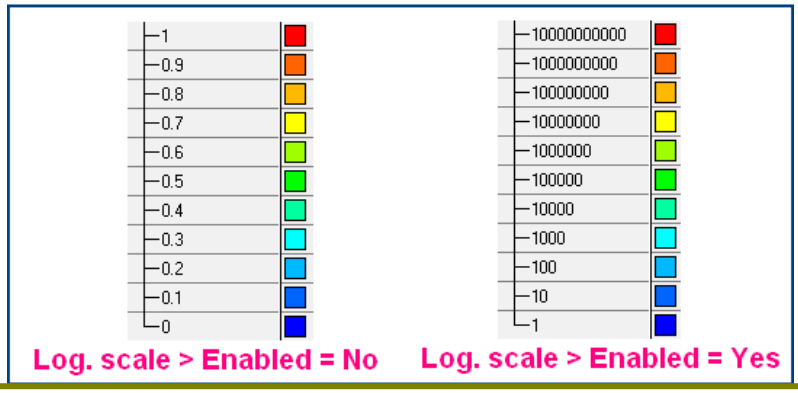
General properties of Layers[1)]	
Parameter	**Description**
Object	Geometry **Object**, on which the **Layer** has been built. This field is read-only and you cannot change it (the **Object** was specified at creation of the **Layer**).
Parts	Specifies the **Object**'s parts, on which the **Layer** is built (volume, the entire surface of any given **Object**'s surface, the whole surface or a fragment containing the reference point). These parameters are available for **Objects**, which have both volume and surface: **Box**, **Cone/cylinder**, **Ellipsoid/sphere**, **Imported object**, and, for some **Layers**, for **Objects**, which are **Planes**. If a **Layer** is built on an **Imported object**, then **Parts > Select = Selected surfaces** is selected automatically and you cannot change this. If the **Object** is a **Plane**, then **Layers** of some types can be built either on the whole cross-section of the computational domain, which is done by the **Plane** or you can select one of fragments.
Parts > Select	Possible options depend on which **Object** the layer is built. The following options are possible: • **Volume** • **Whole surface** • **Selected surfaces** • **Whole plane**: the **Layer** is built on all cross-sections of the computational domain done by the **Plane** • **Selected contour**: the **Layer** is built on those cross-sections of the computational domain done by the **Plane**, where the **Plane**'s **Reference point** locates, see the illustration in the section Layer «Cell set», user interface.
Parts > Surfaces > *(surface)*	These parameters specify, on which specific **Object**'s surfaces the **Layer** is built. The possible options depend on the type of the **Object**. For example, for a truncated cone with a channel and a recess, the following surfaces can be included: **Lateral surface**, **Bottom base**, **Top base**, **Channel**, **First section**, **Second section**. For the cross section of the computational domain **Plane** can be set to build layer **Cell set** for all, and for one of the **Selected circuit**. For each of these parameters it is possible to select either one of the options: • **No** – the layer is not built on this surface • **Yes** – the layer is built on this surface
Visible	If this option is enabled, the layer is displayed in the **View** window. If a tree item, the corresponding layer is selected in the tree **Postprocessor**, the layer is always displayed on the screen, regardless of the value of this parameter. Possible options are: • **No** – the layer is displayed in the **View** window only if it is selected in the project tree (otherwise the layer is not displayed) • **Yes** – the layer is always displayed in the **View** window
Clipped	If this option is enabled, then the layer is cut clipping plane if off, then cutting off the plane it does not apply. Possible options are: • **No** – section planes are not acting on the object • **Yes** – section planes cut to
Lighting	This option allows you to turn on/off lights layer. Possible options are: • **No** – the object is not lit by light sources • **Yes** – the object is lit by light sources
Update	Group of parameters that control updating data for the **Layer**.
Update > Type	The method of updating: • **Disabled**: updating is disabled (turned off) • **Automatic**: automatic update at each time step • **By time**: updates are done over a specified period of time • **By step**: updates are done over a specified number of time steps

General properties of Layers[1)]	
Parameter	**Description**
Update > Number of seconds	Period of time through which the update (this parameter is used when **Update > Type = By time**)
Update > Number of steps	The number of iterations through which the update occurs (this parameter is used when **Update > Type = By step**)
Save to file	Group of parameters that control saving the layer's data in a text file (`glo`-file). Saving in a file is available for not any types of layers.
Save to file > Type	Type of saving the data into a file: • **Disabled**: no data are recorded • **Automatic**: the data are recorded on each time step • **By time**: the data are recorded after a specified time interval • **By step**: the data are recorded after a specified number of steps Regardless of the value of this parameter, the program will write into the file at stopping the computation, if **Save to file > Write mode = Overwrite** and **Save to file > Write on stopping = Yes** are selected (see below).
Save to file > Number of seconds	Period of time, after which recording into the file is done. (this parameter is used when **Save to file > Type = By time**)
Save to file > Number of steps	The number of time steps, after which recording into the file is done. (this parameter is used when **Save to file > Type = By step**)
Save to file > File name	Name of a `glo` file where the data are to be stored. You specify here a a file name (including its extension), but do not specify the path to the file. This file name must not contain some symbols (`: * ? " > < | / \` and first 32 *ASCII* symbols with codes from 0 to 31). Also you can not use file names, which are reserved in the operation system (in *Windows* these are names `AUX`, `COM1` ... `COM9`, `CON`, `LPT1` ... `LPT9`, `NUL` and `PRN`), which are case-insensitive(so you can use no the name `NUL` neither `nul`, `NuL` etc.). If you attempt to specify an unacceptable file name, the program will display an appropriate error message.
Save to file > Write mode	This parameter selects the recording mode: • **Overwrite** – overwrite the file each time you save the data in the file contains only the most recent data and old data are overwritten (it is the default mode). • **Append** – at each record the data are appended to the end of the file, all data are saved.
Save to file > Write on stopping	This parameter instruct the program to make an additional writing into the file after triggering the specified **Stopping conditions** or manual stopping the computation. The program makes writing into the file at each stopping of the project regardless of the value of the **Save to file > Type** parameter. Possible options: **Yes** | **No**. This parameter is available when **Save to file > Type mode = Overwrite**.
Variable	Group of parameters that specify the visualized variable [2) 3)]
Variable > Category	Selection of a category for the variable, which is calculated on the **Object**. Possible options are: • **Common and phase-unrelated variables** • **Variables of phase "Phase #N"** • **User variables** See details in the section Categories of variables.
Variable > Variable	The visualized variable, which is selected from a drop-down list (the list depends on the selected **Category**).
Variable > Component	Visualized component vector variable or the length of the vector (a parameter is present only for vector variables). Possible options are: • **Length** – length of the vector • **X** – the vector's component along the axis X

General properties of Layers[1)]	
Parameter	**Description**
	• **Y** – the vector's component along the axis Y • **Z** – the vector's component along the axis Z
Variable > Interpolation	Interpolation on / off values at the nodes, which is built layer: • **No** - the values of variables in the nodes of the layer are assumed to be the values of the variables in the near center of the cell • **Yes** - the values of variables in the nodes of the layer are determined by interpolation between the values of the variables in the centers of the cells surrounding the node ***Note:*** In both cases, the layer's coloring is smoothed.
Variable > Integrate	Sometimes you can tune the **Layer** so it will visualize not a variable itself but its integral along a specified direction. Integrating of the variable is done over a straight-line segment in a specified direction from the **Object**, on which the **Layer** is built, to the surface of the computational domain or to a **Moving body**.  Periodic surfaces and Sliding surfaces do not interrupt the interval of integration. The integrating can be done for the following **Layers**: • Color contours • Vectors (for the variable, which is used for coloring) • Plot along line (for the main variable and the variable used for coloring) • Plot along curve (for the main variable and the variable used for coloring) • Plot along ellipse (for the main variable and the variable used for coloring) The integrating can be done when the **Object**, on which the **Layer** is built, is a **Plane** or a **Line** only.
Variable > Integrate > Enabled	Enable/disable the integration. • **No** - the integration is not performed • **Yes** - the integration is performed
Variable > Integrate > Integrate along	Selecting the direction, along which the integration will be done. Possible options are: • **Surface normal** (for Plot along line, Plot along curve, Plot along ellipse in the direction, which is normal to the surface of the plot) • **Custom vector** • **X-axis** • **Y-axis** • **Z-axis**
Variable > Integrate > Integration vector > X **Variable > Integrate > Integration vector > Y** **Variable > Integrate > Integration vector > Z**	The components of the direction vector along which the integration is performed (defined in absolute coordinate system). Options are only available when the **Integrate along = Custom vector**.
Variable > Integrate > Invert	Invert the normal vector of integration
Subregion	Limitation layer of a given sub-area. If the layer is built on a calculated boundary (on a **Supergroup** or on the surface of an **Imported object** with a **Moving body**), the value of **(auto)** is applied to automatically select the calculated subregion. In other cases, the list may be a value **(all),**the choice of which is processed all subregions. Automatic selection of the estimated subregion means the following: layer is based on a kind of the estimated border, on both sides of which - two subregions. If one of them is not computational, then automatically selects the calculated subregion. If

General properties of Layers[1)]	
Parameter	**Description**
	both the calculated, then choose that subregion, which is higher in the list of subregions.
Shift	Shift the location of the layer in the direction of or against the normal to the surface on which it is built (option is available in the construction of a layer on the surface that is specified in the parameter group **Parts**), for example, the shift vectors or layer **Color contours** (see illustration). The normal direction can be different for different **Objects**. If necessary, change the direction of the shift, change the sign of numerical values of the parameter to the opposite.
	Layer Distributed characteristics The **Shift** parameter is not specified as a numeric value, but as either **Yes** or **No**, and has the following meaning: • if **Shift=No**, then the values of the integrable variable taken from the surface of the **Object** • if **Shift=Yes**, then the values are taken from the integrable variable volume near the surface of the **object**, while the option becomes available **Variable > Interpolation** **Shift** may be useful, for example, by integrating the flow velocity in the flow near the wall with adhesion, as on the wall itself the speed is zero everywhere and the distribution of the characteristics will be uninformative.
Constraints	Limitations on the area where the **Layer** is built (on a **Plane**), allowing you to build the **Layer** not on the whole **Plane** but only on its some square or rectangular fragment near the **Reference point** of the **Plane**.
Constraints > Enabled	• **No** – layer is based on the entire **Plane** • **Yes** – layer is built on a square or rectangular fragment near the **reference point** of the **Plane**
Constraints > Type	Form fragment **planes** on which to build a layer (if the option is available **Constraints > Enabled = Yes**). Possible options are: • **Square** – limit the construction of a square with its center coincident with the reference point of the **Plane** • **Rectangle** – limit the construction of a rectangle with a center coincident with the reference point of the **Plane**
Constraints > Size	Size of the **Square**'s side, which limits the area of building the **Layer** on the **Plane** (this parameter is available if **Constraints > Type = Square**)
Constraints > Size 1	Sizes of the **Rectangle**'s sides, which limits the area of building the **Layer** on the **Plane** (this parameter is available if **Constraints >Type = Rectangle**)
Constraints > Size 2	
On regular grid	Whether to use for building the **Layer** the computational grid or an auxiliary uniform (regular) grid that is defined by parameters **Grid > ...** (see below). Possible options are: • **No** – use the computational grid • **Yes** – use an auxiliary (regular) grid
Grid > Local	Possible options are: • **Yes** – to build a regular grid along the axes of the local coordinate system **Object** • **No** – specify the number of grid points along the axis of the absolute coordinate system Option is available when **On regular grid = Yes** and built the layer to an **Object** of finite volume (eg, **Box**, **Ellipsoid/sphere**, **Cone/cylinder**, **Imported object**, **Supergroup**).
Grid > Size 1	The number of nodes on a regular grid in a given direction within the **Object** on which the **Layer** is built. Options are available when **On regular grid = Yes**.

General properties of Layers[1)]	
Parameter	**Description**
Grid > Size 2	The **Plane** is built for two-dimensional grid, and therefore the parameter **Size 3** is missing in this case.
Grid > Size 3	
Appearance	Group of parameters that control the visualization of the **Layer**.
Appearance > Mode	The method of display surfaces and contours **Layer**: • **Lines** – displaying as lines only • **Fill** – displaying as filled surface only • **Lines and fill** – displaying as filled surface with lines
Appearance > Lines> Color	Color of the lines
Appearance > Line> Thickness	Thickness of the lines
Appearance > Fill > Color	Color for filling the surfaces
Appearance > Fill > Opacity	Opaqueness of the surfaces
Appearance > Show Grid	• **No** – do not display the grid on the plot • **Yes** – display the grid on the plot
Appearance > Draw over	• **No** – do not draw the plot on the top over other layers • **Yes** – draw the plot on top over other layers
Appearance > Plot > Color	Color of the plot's lines
Appearance > Plot > Width	Thickness of the plot's lines
Appearance > Axes > Color	Color of the plot's axes
Appearance > Axes > Width	Thickness of the plot's axes
Appearance > Grid> Mode	• **Lines** – display the grid lines only • **Fill** – display the shaded surface only • **Lines and fill** – display the grid lines and the shaded surface
Appearance > Grid > Lines > Color	Color of the grid lines
Appearance > Grid > Lines > Width	Thickness of the grid lines
Appearance > Arrows	Color of arrows
Value range[2)]	Group of parameters that specify the range for displaying the variable[4)]
Value range > Mode	Selecting the determination of the maximum and minimum values of the variable for the scale: • **Local** – is automatically determined by the maximum and minimum value of the estimated variable on the object • **Global** – is automatically determined by the values of the estimated variable in the whole phase (available only for the physical variables) • **Manual** – you can adjust the minimum (**Min**) and maximum (**Max**) of the range for displaying the variable
Value range > Max	The visualized variable's value, which corresponds to the upper color of the scale (this parameter can be set if **Mode = Manual** is selected).
Value range > Min	The visualized variable's value, which corresponds to the lower color of the scale (this parameter can be set if **Mode = Manual** is selected).

General properties of Layers[1]	
Parameter	**Description**
Log. scale	Settings for logarithmic scale. When a logarithmic scale is used, the transitions between colors of the **Palette** depend logarithmically from values of the **Variable**. Otherwise the transitions depend linearly. 
Log. scale > Enabled	Defines if the layer be displayed using the logarithmic scale or no. Possible options are: **No** | **Yes**.
Log. scale > Minimum	Value of the **Variable**, that corresponds to the transition between the first and the second colors in the **Palette** in the logarithmic scale.
Axis X	Group of parameters, which configure the x-axis of a plot
Axis X > Num. spans	Number of segments on the axis
Axis X > Invert	Inverts the direction of the axis (oppositely to the standard direction)
Axis X > Length	Length of the axis
Axis X > Length > Mode	Selecting the determination of the length of the axis: • **Automatic** – the length of the axis is the distance from the **Reference point** (the origin of LCS) to the boundary of the computational domain • **Manual** – the length of the axis is set manually
Axis X > Length > Value	The length of the axis defined manually (option is available when the **Length > Mode = Manual**)
Axis Y	Parameter group for setting the y-axis plot.
Axis Y > Num. spans	Parameters that are similar to the corresponding parameters for **Axis X**.
Axis Y > Invert	
Axis Y > Length	
Axis Y > Length > Mode	
Axis Y > Length > Value	
Palette	Settings of the **Palette**, which is applied to visualize a variable[4] using colors [2] [3]. See details about **Palettes** and their use in the section Parameters for defining a palette.

Notes:

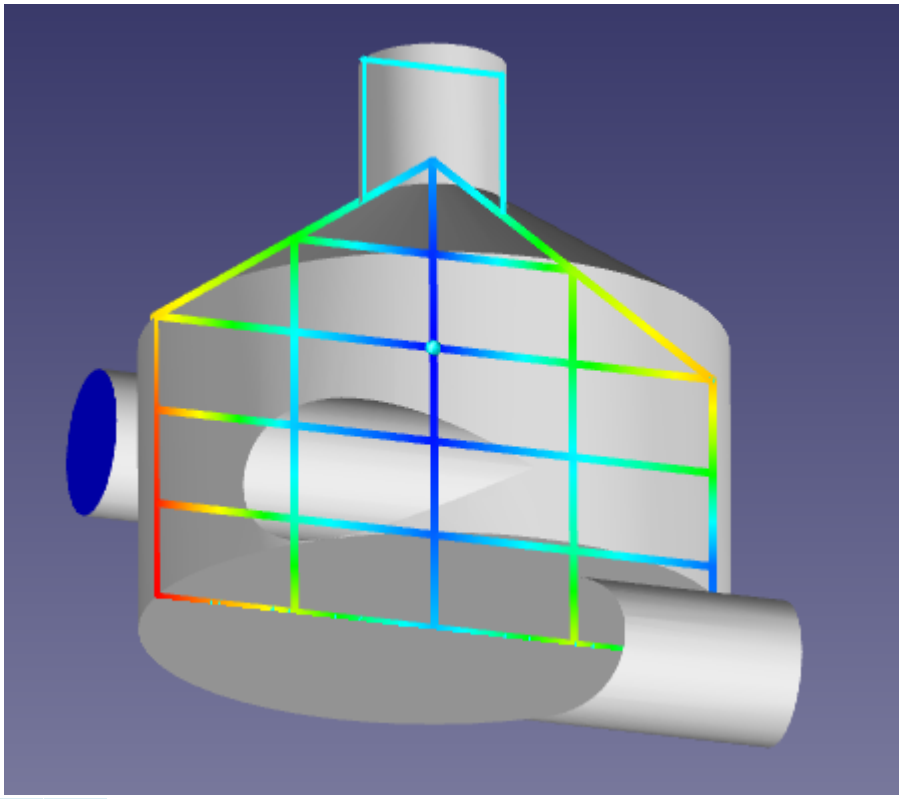
[1] Presence of specific properties of a **Layer** depends on type of the **Layer**, type of the **Variable**, and type of the **Object**, on which the **Layer** is built.

[2] Layer **Vectors** can visualize two variables: one variable (vector) by the vector's direction and length, and the other by the vector's color (the second variable is specified in the **Coloring** group of parameters).

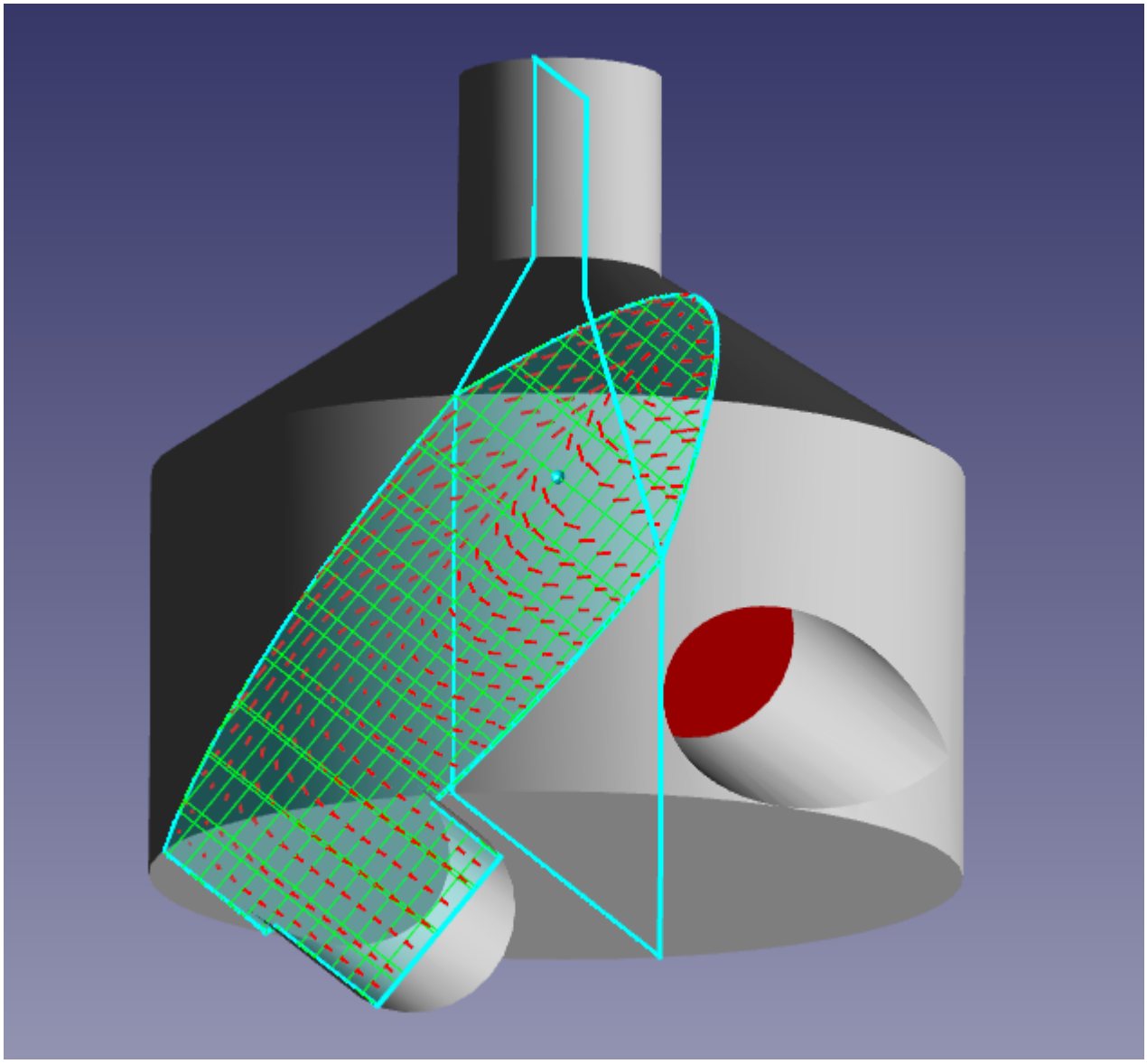
[3] Layer **Streamlines** also can visualize two variables: one variable (vector) forms trajectories of the streamlines, and the other defines coloring of the streamlines (the second variable is specified in the **Coloring** group of parameters)

[4] or volumes of small or gap cells in the layer **Cell set**.

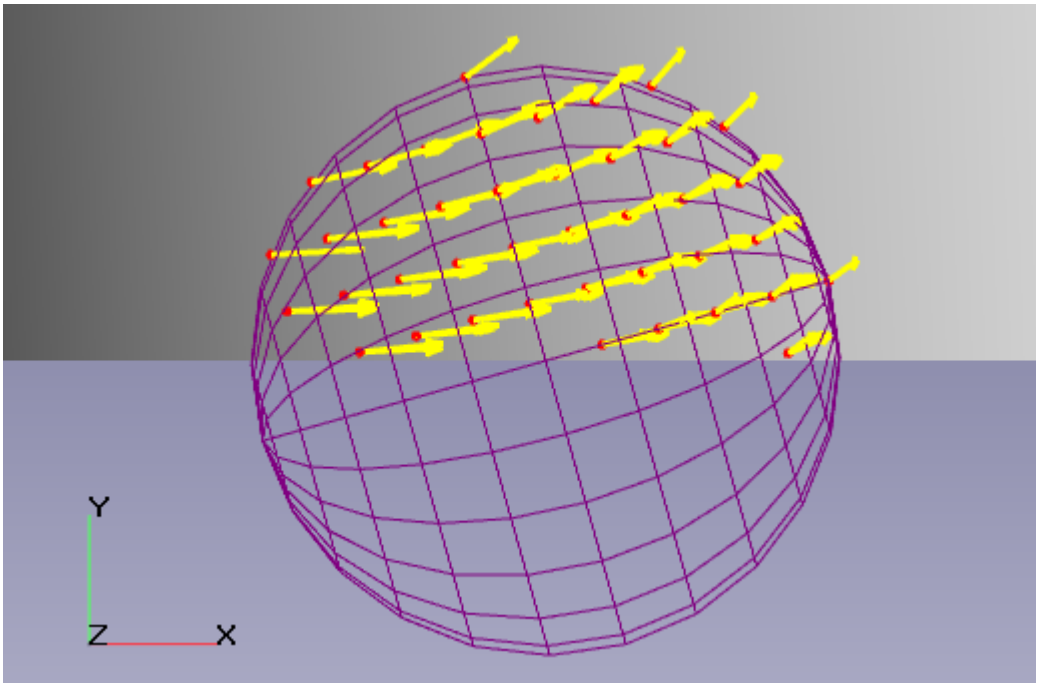
Illustrations



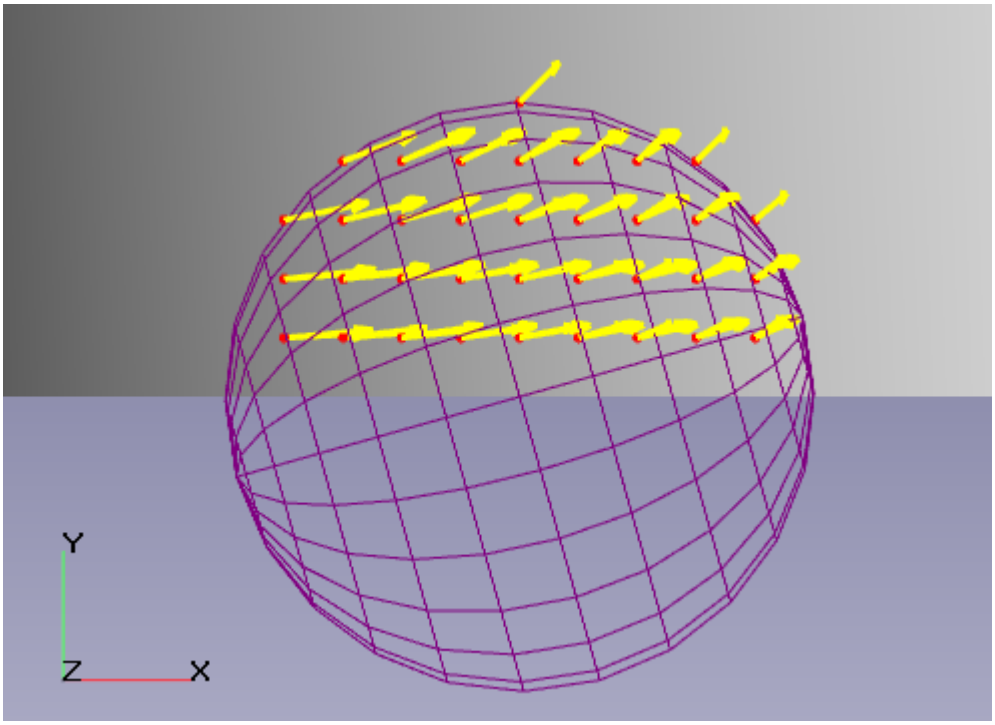
The **Color contours** layer (with the method of displaying **Color frame**) is built on a regular grid, **Grid > Size 1 = 7**, **Grid > Size 2 = 7**



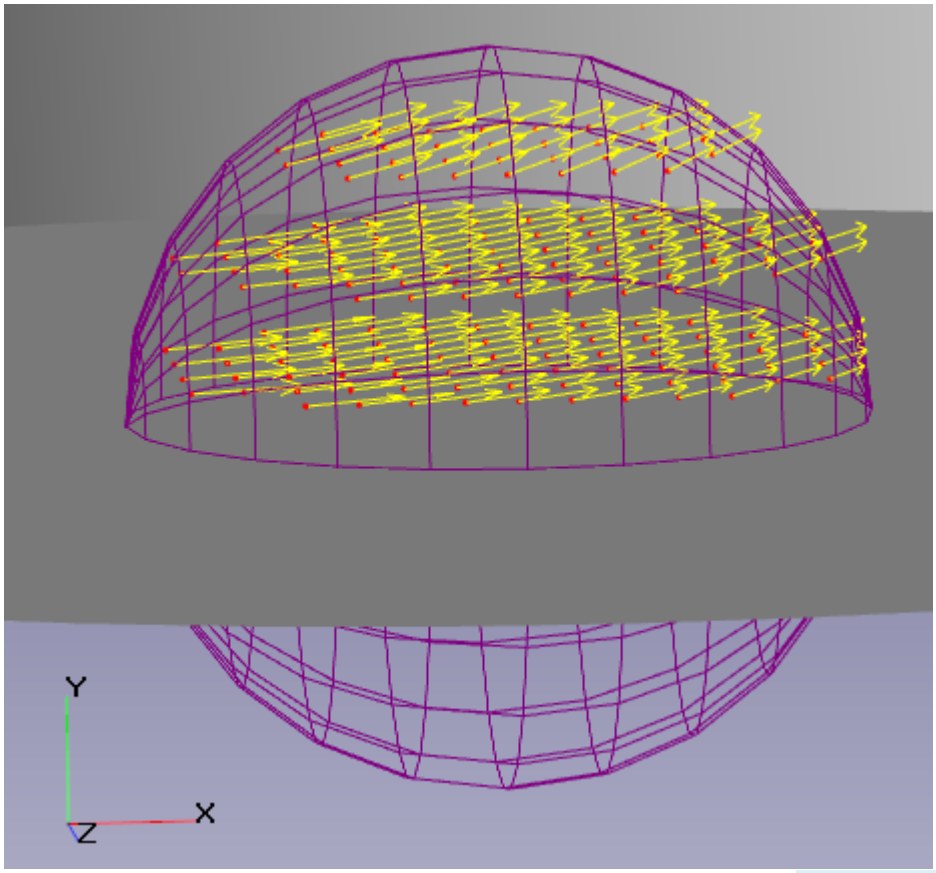
If you set **On a regular grid = No**, then the **Layer** will be built at the nodes of the computational grid (The picture shows two **Layers**, built on the same **Plane** -**Computational grid section** and **Vectors**)



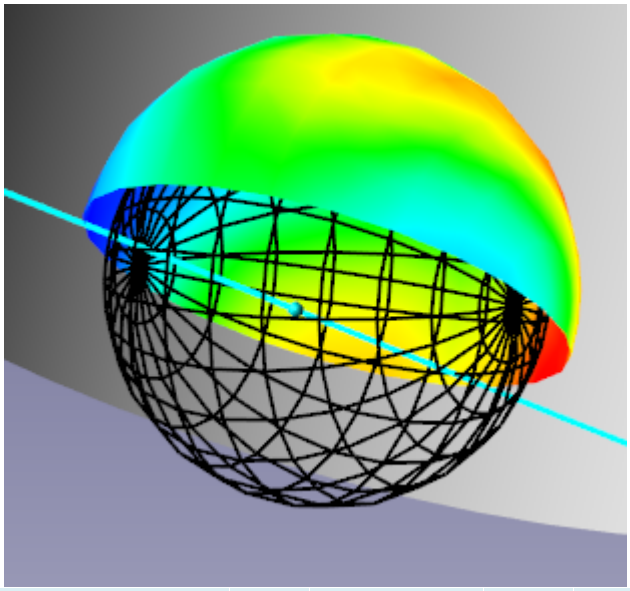
Regular grid is built along the axes of the local coordinate systemof the **Object** (specified **Grid > Local = Yes**)



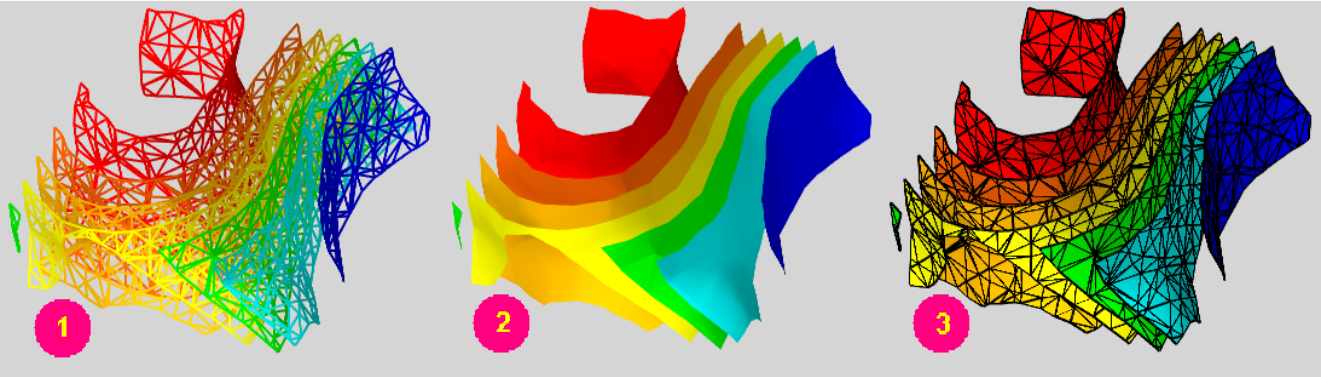
Regular grid is built along the axes of the absolute coordinate system (specified **Grid > Local = No**)



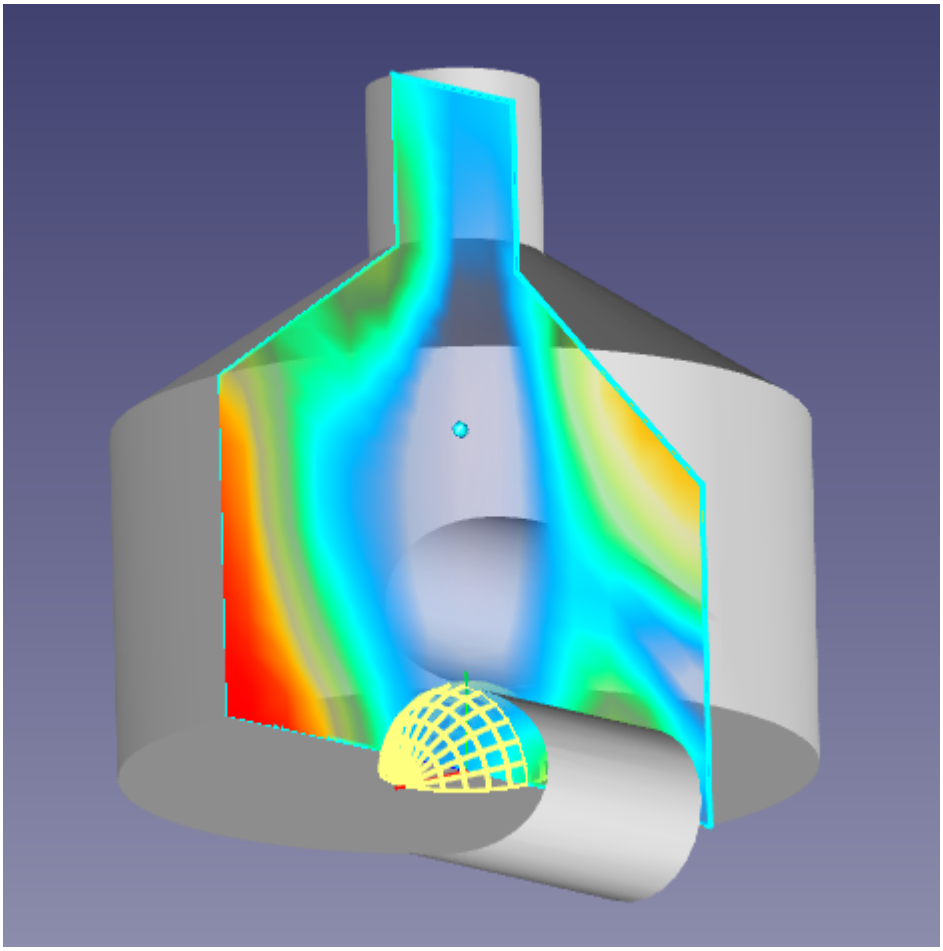
When using a regular grid number of nodes within its **Object** of the given parameters **Grid > Size 1**, **Grid > Size 2** and **Grid > Size 3**.



The **Shift** parameter allows you to shift the **Layer** relatively to the **Object** on which the **Layer** is built.

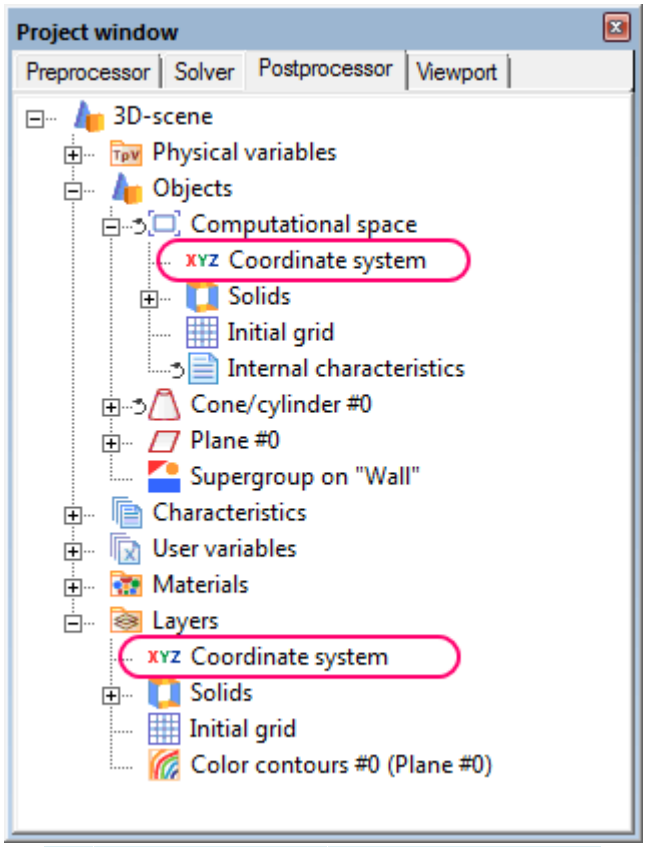


Use of the **Appearance > Mode** parameter for **Isosurfaces** (1 - **Lines**, 2 - **Fill**, 3 - **Lines and fill**)



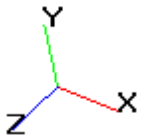
Some colors in **Color contours** and **Isosurfaces** layers can be defined as partially or fully transparent using the **Palette > Transparency** group of properties

8.1.8.5.8.2 Layer «Coordinate system», user interface



The **Coordinate system** layer in the project tree

The **Coordinate system** layer shows the direction of three coordinate axes X, Y and Z. The coordinate axes are rendered as solid lines of different colors (X - red line, Y - green and Z - blue) having a starting point. In addition, about all axes are written their names.

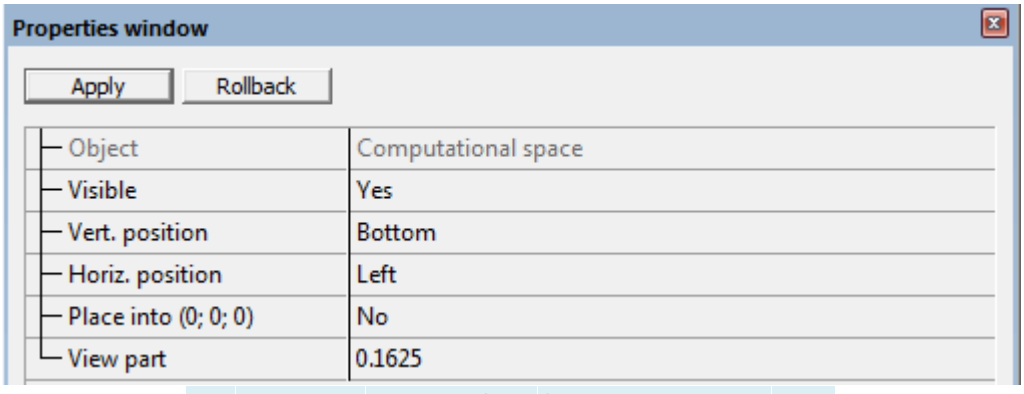


Depending on the layer of origin can be linked to the point of the graphics window or to the point of three-dimensional space (the center of the computational domain). The **Coordinate system** layer is always displayed over other layers, and thus it can not be closed by them.

You can not create more than one layer of this type.

The **Coordinate system** layer is automatically created with the project and receives with the default settings.

Transactions layer **Coordinate system** performed using context menu items to be opened in the project tree.



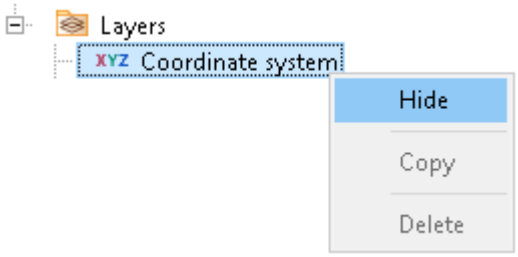
The **Properties** window of the **Coordinate system** layer

Layer **coordinate system** also goes into the folder **Objects > Computational space**.

Parameters of the «Coordinate system» layer

Parameter	Description
Object	Name of the object is associated with the element **(Computational space)**. Field information is not editable
Visible	• **No** - the layer is not visible in the **View** window (except when the layer is selected in the project tree) • **Yes** - the layer is always visible in the **View** window
Vert. position	• **Top** - display the element in the top half of the **View** window • **Bottom** - display the element in the bottom half of the **View** window
Horiz. position	• **Left** - display the element in the left half of the **View** window • **Right** - display the element in the right half of the **View** window
Place into (0; 0; 0)	• **No** - the coordinate system icon is displayed in accordance with the parameters of **Vert. position** and **Horiz. position** (so the icon is displayed in one of the corners of the **View** window). • **Yes** - the coordinate system icon is displayed at the beginning of the absolute coordinate system (ACS).
View part	Part of the **View** window occupied by the coordinate system's icon.

Context menu of the layer «Coordinate system»



Context menu of the **Coordinate system** layer in the project tree

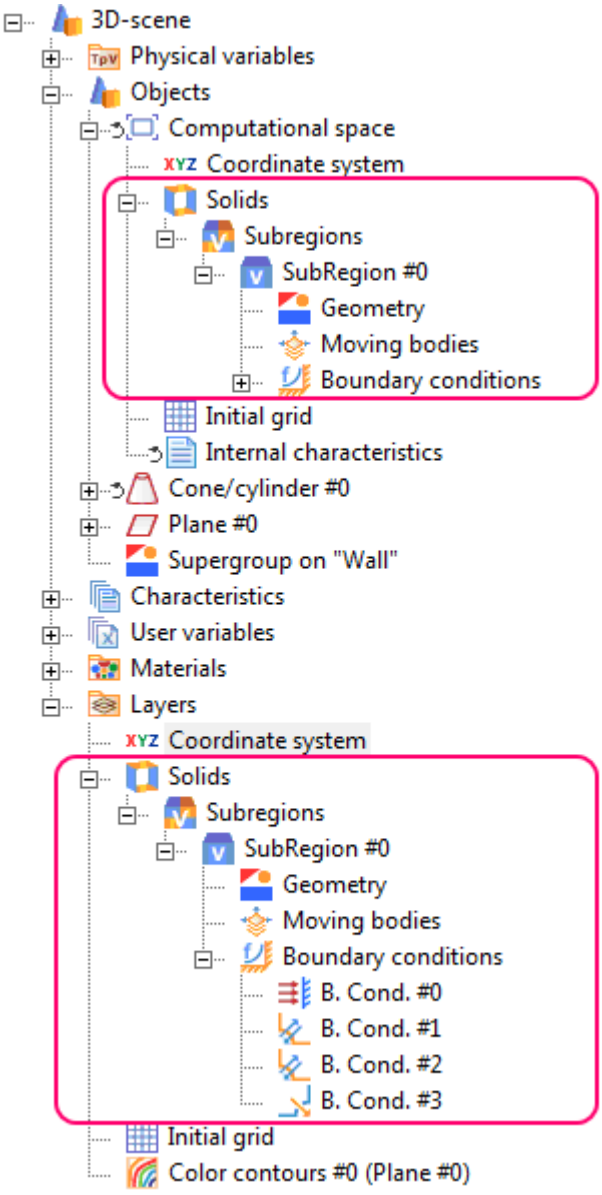
Context menu of the **Coordinate system** layer:

Menu item	Description
Hide	- the **Layer** is always displayed in the **View** window - the **Layer** is displayed in the **View** window, only when it is selected in the project tree
Copy	*These menu items are not used for this layer and are faded.*
Delete	

See also: Layer «Coordinate system» in *Principal concepts of FlowVision.*

8.1.8.5.8.3 Layers in the «Solids» folder, user interface

Folder **Layers > Solids** stores the layer **Solids** and folders **Layers > Solids > Subregions**.



The **Solids** folders in the project tree

Layers **Solids** folder designed to display surfaces making up the computational domain and the moving body.

Folder **Solids** displayed in two places:

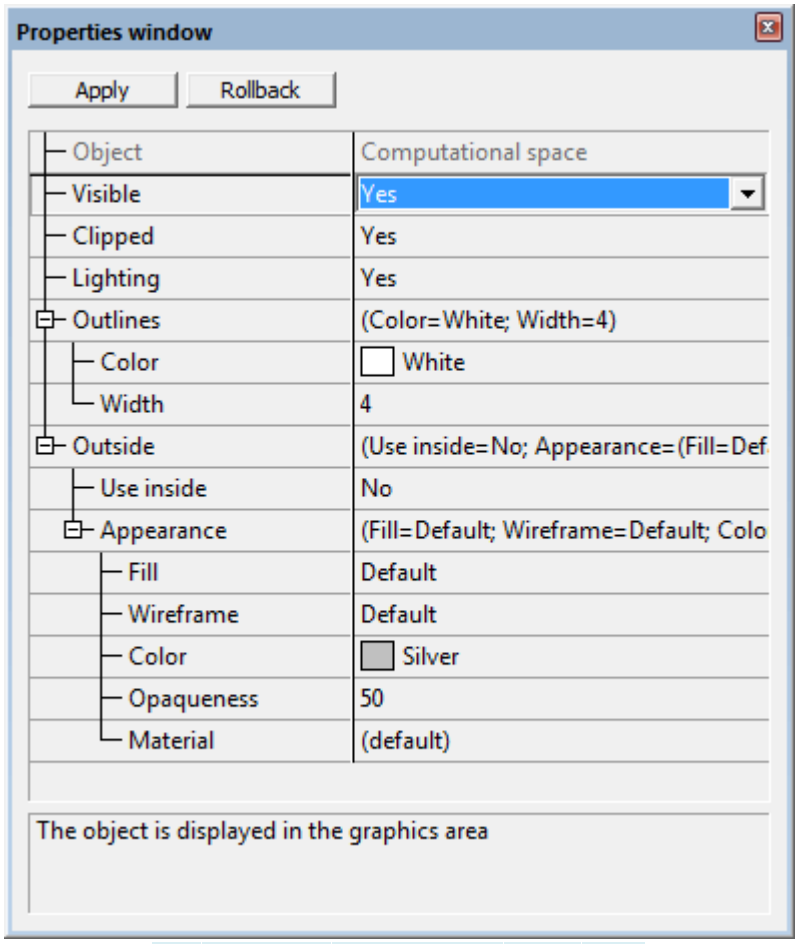
- in the folder **Objects > Computational space**
- in the folder **Layers**

Layers folder **Solids** represent the geometry model of the computational domain.

The items in the folder **Solids** are intended to customize the display of the corresponding geometric elements. The lower layer is a component **Solids** in the project tree, the higher the priority given by them to display. Lowest priority have the display settings set in properties of the folder **Solids**.

Solids layer can be removed and re-established in a single copy.

Parameters of the «Solids» layer

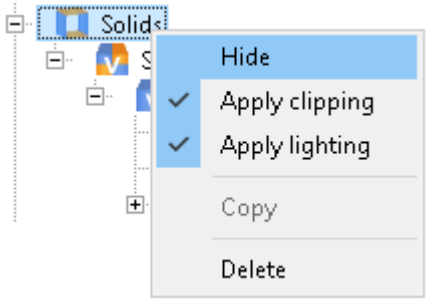


The **Properties** window of the **Solids** layer

Parameters of the **Solids** layer:

Parameter	Description
Object	Name of the object is associated with the element. Field information is not editable.
Visible	Possible options are: • **No** - layer is not displayed in the **View** (if the layer is selected in the project tree) • **Yes** - layer is always displayed in the **View** window
Clipped	Possible options are: • **No** - section planes are not acting on the object • **Yes** - section planes cut to
Lighting	Possible options are: • **No** – the object is not lit by light sources • **Yes** – the object is lit by light sources
Outlines > Color **Outlines > Width**	The choice of color and line thickness for outlines of surfaces of the geometry model
Outside > Use inside	Possible options are: • **No** - Display the outside surfaces of the geometry model with the parameters set in the block **Appearance** • **Yes** - Display the outer side surfaces of the geometry model as well as the inner side
Outside > Appearance > Fill	Possible options are: • **No** - faces of facets are not displayed

Parameter	Description
	• **Yes** - to show faces of facets • **Default** - shows the faces of facets to the settings in the default
Outside > Appearance > Wireframe	Possible options are: • **No** - edges of facets are not displayed • **Yes** - edges of facets are displayed • **Default** - displays the edges of facets with the parameters set by default
Outside > Appearance > Color	Choice of exterior colors surfaces geometry model
Outside > Appearance > Opaqueness	Opacity geometric surface model in %
Outside > Appearance > Material	Possible options are: • **(default)** - do not apply reflecting and emitting properties of any **Material** • **Material #N** - displaying with reflecting and emitting properties of the specified **Material**



Context menu of the **Solids** layer in the project tree

Context menu of the **Solids** layer in the project tree:

Menu item	Description
Hide	- the **Layer** is always displayed in the **View** window - the **Layer** is displayed in the **View** window, only when it is selected in the project tree
Apply clipping	- clipping **Planes** do not affect the **Layer** - clipping **Planes** cut the **Layer**
Apply lighting	- the **Layer** is *not* lit by light sources - the **Layer** is lit by light sources
Copy	*This menu option is not active, because the **Solids** layer can not be copied*
Delete	Removing the **Solids** layer from the project tree

Child Elements folder Layers> Solids

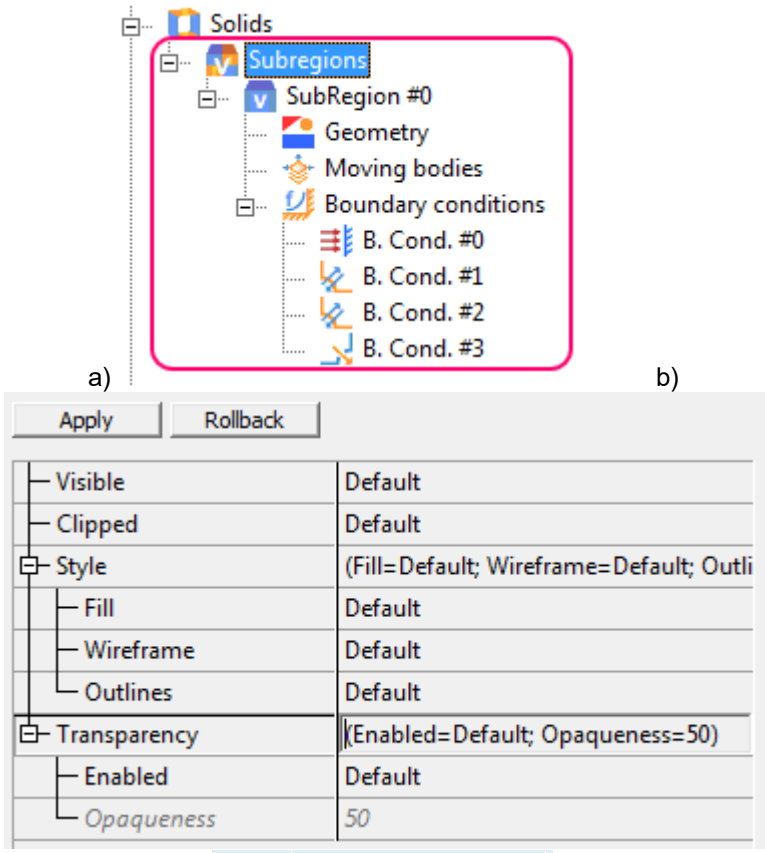
The folder **Layers> Solids** contains the folder **Layers > Solids > Subregions**, which contains a folder separate subregions (**Layers > Solids > Subregions > SubRegion #N**), containing elements and subfolders:

- **Layers > Solids > Subregions > SubRegion #N > Geometry**
- **Layers > Solids > Subregions > SubRegion #N > Moving bodies**
- **Layers > Solids > Subregions > SubRegion #N > Boundary conditions**

Folder "Layers > Solids > Subregions"

Subregions component is displayed in the folder **Layers > Solids > Subregions** and contains display parameters geometry model of the computational domain in the **View**.

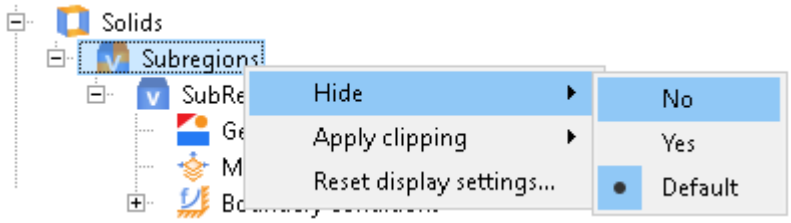
All active parameters may be external parameters.



Folder **Solids > Subregions**:
a) in the project tree; b) its **Properties** window

The Properties window of the folder **Solids > Subregions**:

Parameter	Description
Visible	• **No** – the component layer does not appear in **View** (if the component layer is selected in the project tree) • **Yes** – the component layer is always displayed in the **View** window
Clipped	• **No** – section planes are not acting on the components of the **Layer** • **Yes** – section planes cut component layer
Style > Fill	• **No** – faces of facets are not displayed • **Yes** – to show faces of facets • **Default** – the program displays faces of facets according to parameters that are set for the layer **Solids**
Style > Wireframe	• **No** – edges of facets are not displayed • **Yes** – edges of facets are displayed • **Default** – the program displays edges of facets according to parameters that are set for the layer **Solids**
Style > Outlines	• **No** – outlines of groups of facets are not displayed • **Yes** – outlines of groups of facets are displayed • **Default** – the program displays outlines of groups of facets according to parameters that are set for the layer **Solids**
Transparency > Enabled	• **Yes** – surfaces are translucent • **No** – surfaces are opaque
Transparency > Opaqueness	Opacity geometric surface model in %



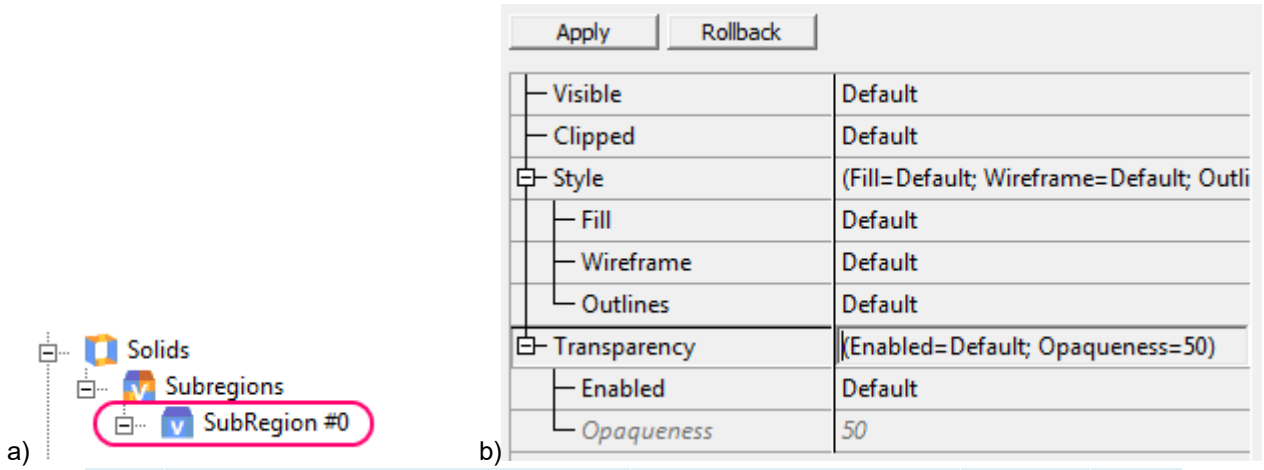
Context menu of the folder **Solids > Subregions** in the project tree

Context menu of the folder **Solids > Subregions** in the project tree:

Menu item	Description
Hide	• **No** – the component layer is always displayed in the **View** window • **Yes** – the component layer does not appear in **View** (if the component layer is selected in the project tree) • **Default** – displays the component layer, if the map is set in a layer of **Solids**
Apply clipping	• **No** – section planes are not acting on the components of the **Layer** • **Yes** – section planes cut components of the **Layer** • **Default** – section planes act as defined in the layer **Solids**
Reset display settings	Set the displaying parameters for this and child elements as they are set for the layer **Solids**.

Folder "Layers > Solids > Subregions >SubRegion #N»

Element **SubRegion #N** is displayed in the folder **Layers > Solids > Subregions > SubRegion #N** and contains display parameters geometry model computational subregion and being in her moving bodies in the **View**.

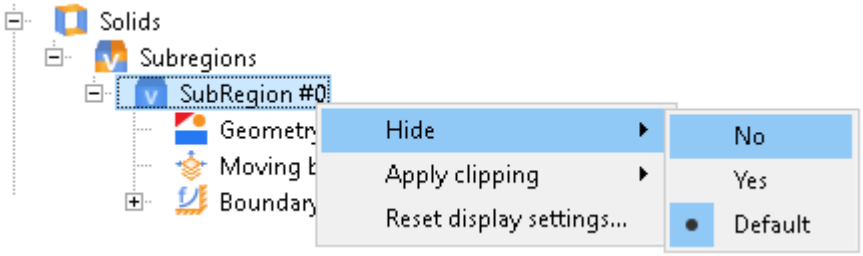


Folder **Solids > Subregions > SubRegion #N**: a) in the project tree; b) its **Properties** window

Parameters of the folder **Solids > Subregions > SubRegion #N**:

Parameter	Description
Visible	• **No** - the component layer does not appear in **View** (if the component layer is selected in the project tree) • **Yes** - the component layer is always displayedin the **View** window
Clipped	• **No** - section planes are not acting on the components of the **Layer** • **Yes** - section planes cut component layer
Style > Fill	• **No** - faces of facets are not visible • **Yes** - to show faces of facets • **Default** - shows the faces of facets with the parameters that are set for the **Subregions** parent element
Style> Wireframe	• **No**: edges of facets are not displayed

Parameter	Description
	• **Yes**: edges of facets are displayed • **Default**: the program displays edges of facets according to parameters that are set for the **Subregions** parent element
Style > Outlines	• **No**: outlines of groups of facets are not displayed • **Yes**: outlines of groups of facets are displayed • **Default**: the program displays outlines of groups of facets according to parameters that are set for the **Subregions** parent element
Transparency > Enabled	• **Yes** - surfaces are translucent • **No** - surfaces are opaque
Transparency > Opaqueness	Opacity geometric surface model in %



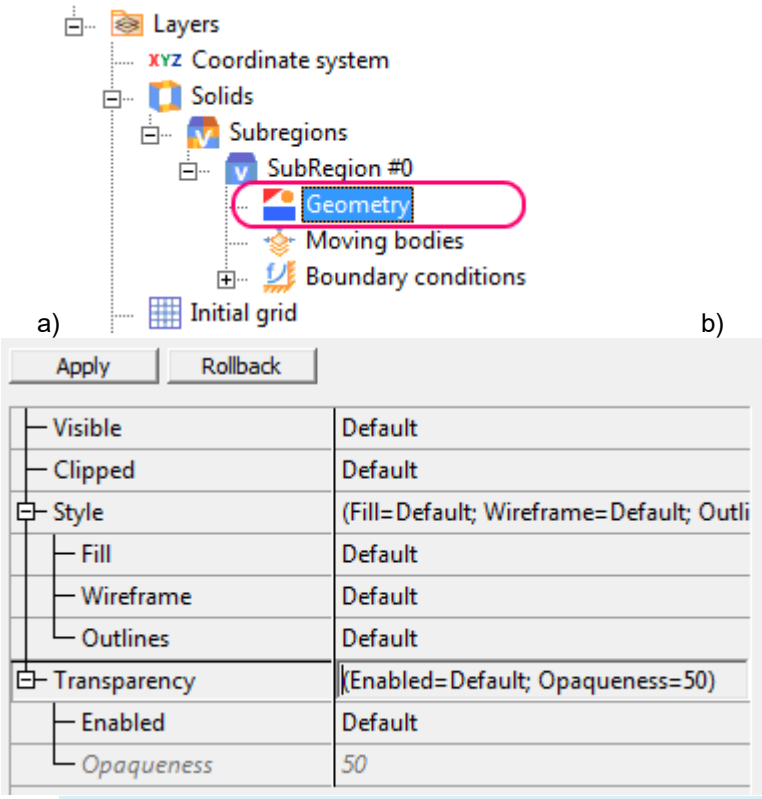
Context menu of the folder **Layers > Solids > Subregions > SubRegion #N** in the project tree

Context menu of the folder **Layers > Solids > Subregions > SubRegion #N** in the project tree:

Menu item	Description
Hide	• **No** – the component layer is always displayed in the **View** window • **Yes** – the component layer does not appear in the **View** window (if the component layer is selected in the project tree) • **Default** – displays the component layer, if the display is defined in the component of the **Subregion**
Apply clipping	• **No** – section planes are not acting on the components • **Yes** – section planes cut components • **Default** – section planes act as it isdefined in the component of the **Subregion**
Reset display settings	Set the displaying parameters for this and child elements as they are set for the parent element in the project tree.

The element "Layers > Solids > Subregions > SubRegion #N > Geometry"

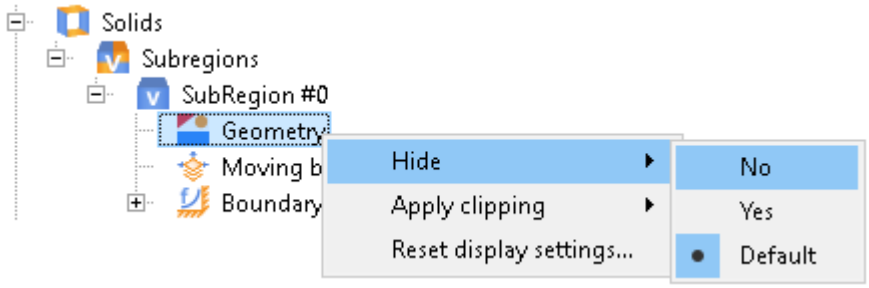
The **Geometry** element is displayed in the folder **Layers > Solids > Subregions > SubRegion #N** (see the subsection above) and contains the display options in the **View** window geometry model computational subregion.



Element **Layers > Solids > Subregions > SubRegion #N > Geometry**:
a) in the project tree, b) its **Properties** window

Parameters of the element **Layers > Solids > Subregions > SubRegion #N > Geometry**:

Parameter	Description
Visible	• **No**: the component layer does not appear in the **View** window (if the component layer is selected in the project tree) • **Yes**: the component layer is always displayed in the **View** window.
Clipped	• **No**: clipping planes do not affect this components of the **Layer** • **Yes**: clipping planes cut this components of the **Layer**
Style > Fill	• **No**: faces of facets are not displayed • **Yes**: faces of facets are displayed • **Default** - the settings are applied, which are set for the folder **Layers > Solids > Subregions > SubRegion #N**
Style > Wireframe	• **No**: edges of facets are not displayed • **Yes**: edges of facets are displayed • **Default** - the settings are applied, which are set for the folder **Layers > Solids > Subregions > SubRegion #N**
Style > Outlines	• **No**: outlines of groups of facets are not displayed • **Yes**: outlines of groups of facets are displayed • **Default** - the settings are applied, which are set for the folder **Layers > Solids > Subregions > SubRegion #N**
Transparency > Enabled	• **Yes** - surfaces are translucent • **No** - surfaces are opaque
Transparency > Opaqueness	Opacity geometric surface model in %

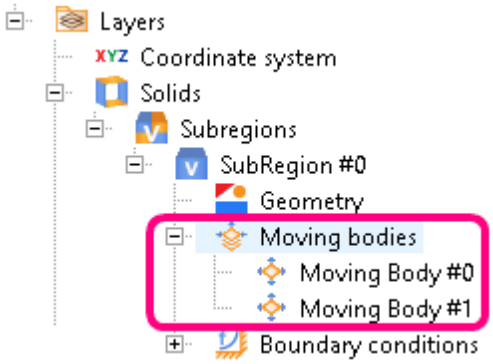


Context menu of the element **Layers > Solids > Subregions > SubRegion #N > Geometry** in the project tree

Context menu of the element **Layers > Solids > Subregions > SubRegion #N > Geometry** in the project tree:

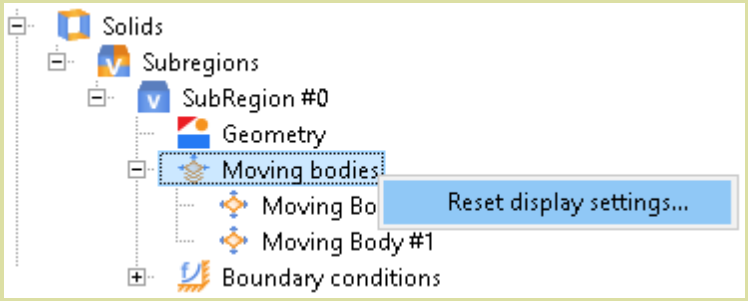
Menu item	Description
Hide	• **No** – the component layer is always displayed in the **View** window • **Yes** – component layer does not appear in **View** (if the component layer is selected in the project tree) • **Default** - the settings are applied, which are set for the folder **Layers > Solids > Subregions > SubRegion #N**
Apply clipping	• **No** – section planes are not acting on the object • **Yes** – section planes cut object • **Default** – the settings are applied, which are set for the folder **Layers > Solids > Subregions > SubRegion #N**
Reset display settings	Set the displaying parameters for this and child elements as they are set for the parent element in the project tree.

Folder «Layers > Solids > Subregions > SubRegion #N > Moving bodies»



Folder **Solids > Subregions > SubRegion #N > Moving bodies** in the project tree

The **Properties** window of the folder **Solids > Subregions > SubRegion #N > Moving bodies** has no parameters.



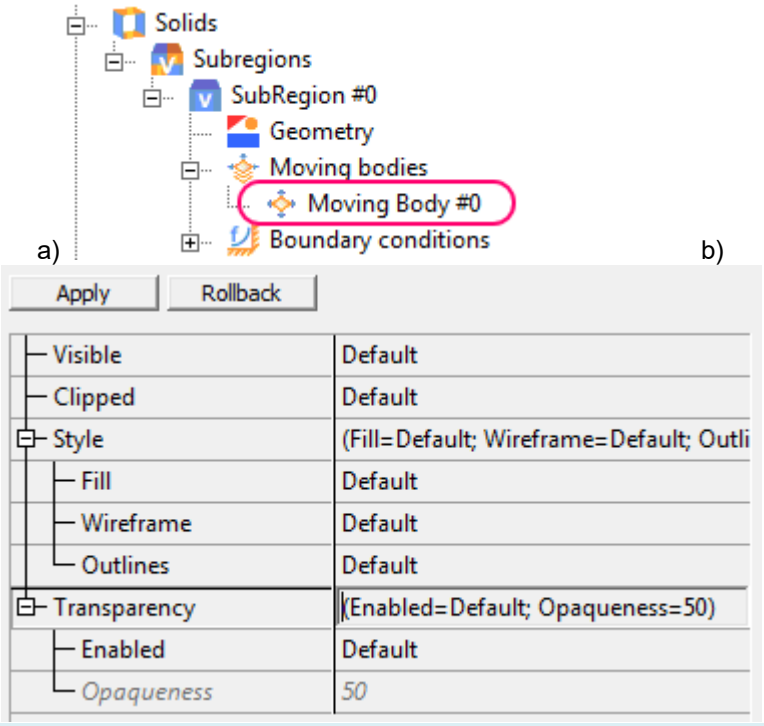
Context menu of the folder «Solids > Subregions > SubRegion #N > Moving bodies»

Menu item	Description
Reset display settings	Set the displaying parameters for this and child elements as they are set for the parent element in the project tree.

Element "Layers >Solids >Subregions >SubRegion #N > Moving bodies > Moving Body #N"

Elements **Layers > Solids > Subregions >SubRegion #N > Moving bodies > Moving Body #N** specify parameters that are used to display appropriate **Moving bodies** in the **View** window.

All active parameters may be included in the number of external parameters.

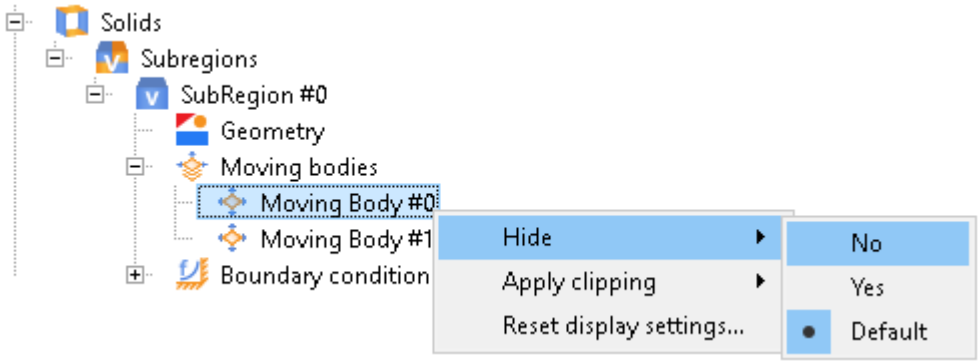


Element **Solids > Subregions > SubRegion #N > Moving bodies > Moving Body #N**:
a - in the project tree, b - its **Properties** window

Parameters of the element **Solids > Subregions > SubRegion #N > Moving bodies > Moving Body #N**:

Parameter	Description
Visible	• **No** – the component layer does not appear in **View** (if the component layer is selected in the project tree) • **Yes** – the component layer is always displayedin the **View** window
Clipped	• **No** – section planes are not acting on the component of the **Layer** • **Yes** – section planes cut component of the **Layer**
Style> Fill	• **No** – faces of facets are not displayed • **Yes** – faces of facets are displayed • **Default** – the program displays faces of facets according to parameters that are set for the **Subregion #N** parent element

Parameter	Description
Style> Wireframe	• **No** – edges of facets are not displayed • **Yes** – edges of facets are displayed • **Default** – the program displays edges of facets according to parameters that are set for the **Subregion #N** parent element
Style> Outlines	• **No** – outlines of groups of facets are not displayed • **Yes** – outlines of groups of facets are displayed • **Default** – the program displays outlines of groups of facets according to parameters that are set for the **Subregion #N** parent element
Transparency > Enabled	• **Yes** – surfaces are translucent • **No** – surfaces are opaque
Transparency > Opaqueness	Opacity geometric surface model in %

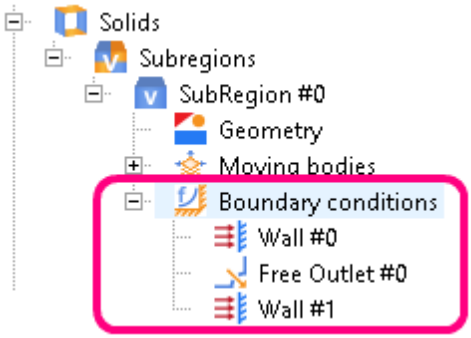


Context menu of the element **Solids > Subregions > SubRegion #N > Moving bodies > Moving Body #N** in the project tree

Context menu of the element **Solids > Subregions > SubRegion #N > Moving bodies > Moving Body #N** in the project tree:

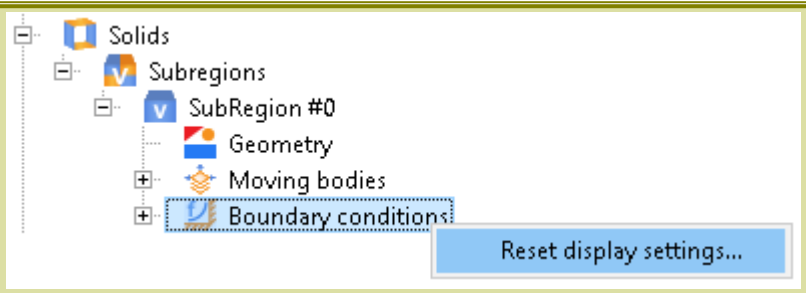
Menu item	Description
Hide	• **No** – the component layer is always displayedin the **View** window • **Yes** – the component layer does not appearin **View** (if the component layer is selected in the project tree) • **Default** – the settings are applied, which are set for the folder **Layers > Solids > Subregions > SubRegion #N**
Apply clipping	• **No** – section planes are not acting on the component of the **Layer** • **Yes** – section planes cut the component of the **Layer** • **Default** – the settings are applied, which are set for the folder **Layers > Solids > Subregions > SubRegion #N**
Reset display settings	Set the displaying parameters for this and child elements as they are set for the parent element in the project tree.

Folder «Layers > Solids > Subregions > SubRegion #N > Boundary conditions»



Folder **Solids > Subregions > SubRegion #N > Boundary conditions** in the project tree

The **Properties** window of the folder **Solids > Subregions > SubRegion #N > Boundary conditions** has no parameters.

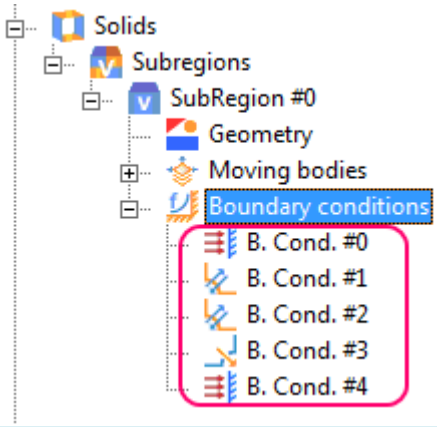


Context menu of the folder «Solids > Subregions > SubRegion #N > Boundary conditions»

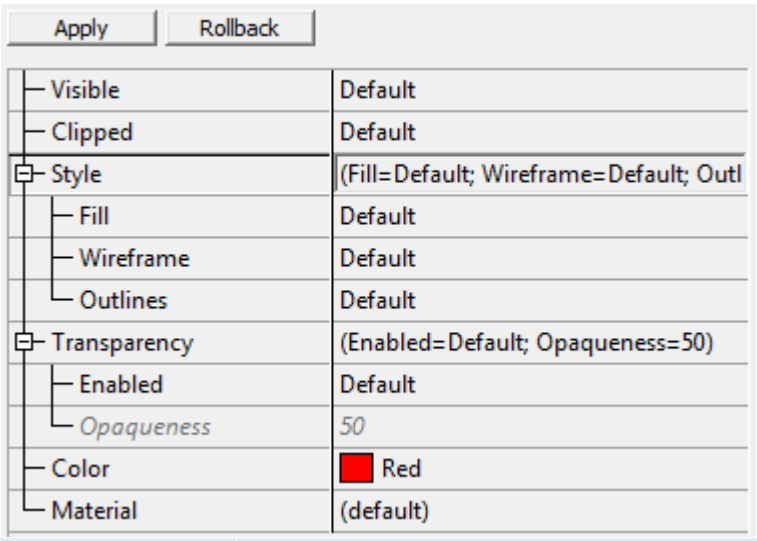
Menu item	Menu item
Reset display settings	Set the displaying parameters for this and child elements as they are set for the parent element in the project tree.

Elements "Layers > Solids >Subregions > SubRegion #N > Boundary conditions >*Boundary condition*"

Elements **Layers > Solids > Subregions > SubRegion #N > Boundary conditions > *Boundary condition*** specify parameters that are used to display appropriate **Boundary conditions** in the **View** window.



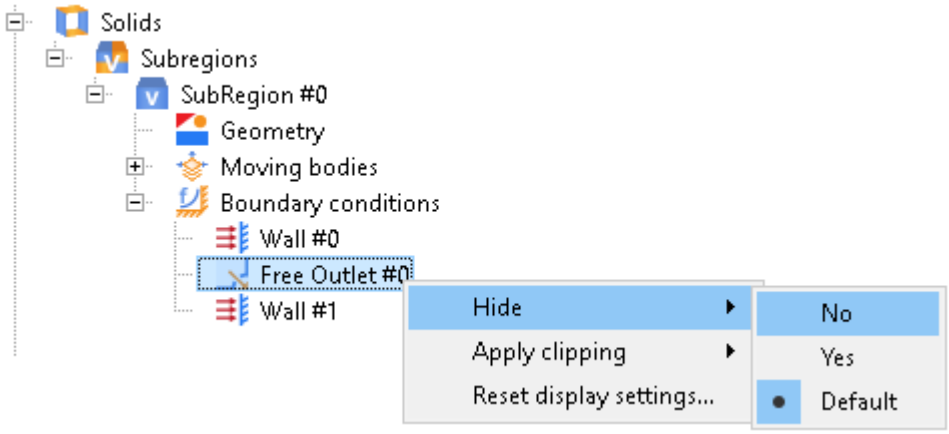
Elements **Layers > Solids > Subregions > SubRegion #N > Boundary conditions > *Boundary condition*** in the project tree



The **Properties** window of the element **Layers > Solids > Subregions > SubRegion #N > Boundary conditions > *Boundary condition***

Parameters of the element **Layers > Solids > Subregions > SubRegion #N > Boundary conditions > *Boundary condition***:

Parameter	Description
Visible	• **No** – the component layer does not appear in **View** (if the component layer is selected in the project tree) • **Yes** – the component layer is always displayedin the **View** window
Clipped	• **No** – section planes are not acting on the components of the **Layer** • **Yes** – section planes cut component of the **Layer**
Style> Fill	• **No** – faces of facets are not displayed • **Yes** – faces of facets are displayed • **Default** – the program displays faces of facets according to parameters that are set for the **Subregion #N** parent element
Style> Wireframe	• **No** – edges of facets are not displayed • **Yes** – edges of facets are displayed • **Default** – the program displays edges of facets according to parameters that are set for the **Subregion #N** parent element
Style> Outlines	• **No** – outlines of groups of facets are not displayed • **Yes** – outlines of groups of facets are displayed • **Default** – the program displays outlines of groups of facets according to parameters that are set for the **Subregion #N** parent element
Transparency > Enabled	• **Yes** – surfaces are translucent • **No** – surfaces are opaque
Transparency > Opaqueness	Opacity geometric surface model in %
Color	Choice of color shading facets with the boundary condition
Material	Visualization taking in account a material of the surface. Possible options: • **(default)** - use of standard display settings • **Material #N** - apply the display settings appropriate any material specified in Materials folder

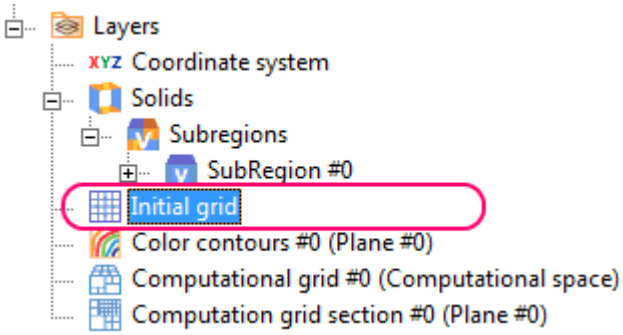


Context menu of the element **Layers > Solids > Subregions > SubRegion #N > Boundary conditions > *Boundary condition*** in the project tree

Context menu of the **B.Cond #N** element in the project tree:

Menu item	Description
Hide	• **No** – the component layer is always displayed in the **View** window • **Yes** – the component layer does not appear in **View** (if the component layer is selected in the project tree) • **Default** – the settings are applied, which are set for the folder **Layers > Solids > Subregions > SubRegion #N**
Apply clipping	• **No** – section planes are not acting on the object • **Yes** – section planes cut the **Layer** • **Default** – the settings are applied, which are set for the folder **Layers > Solids > Subregions > SubRegion #N**
Reset display settings	Set the displaying parameters for this and child elements as they are set for the parent element in the project tree.

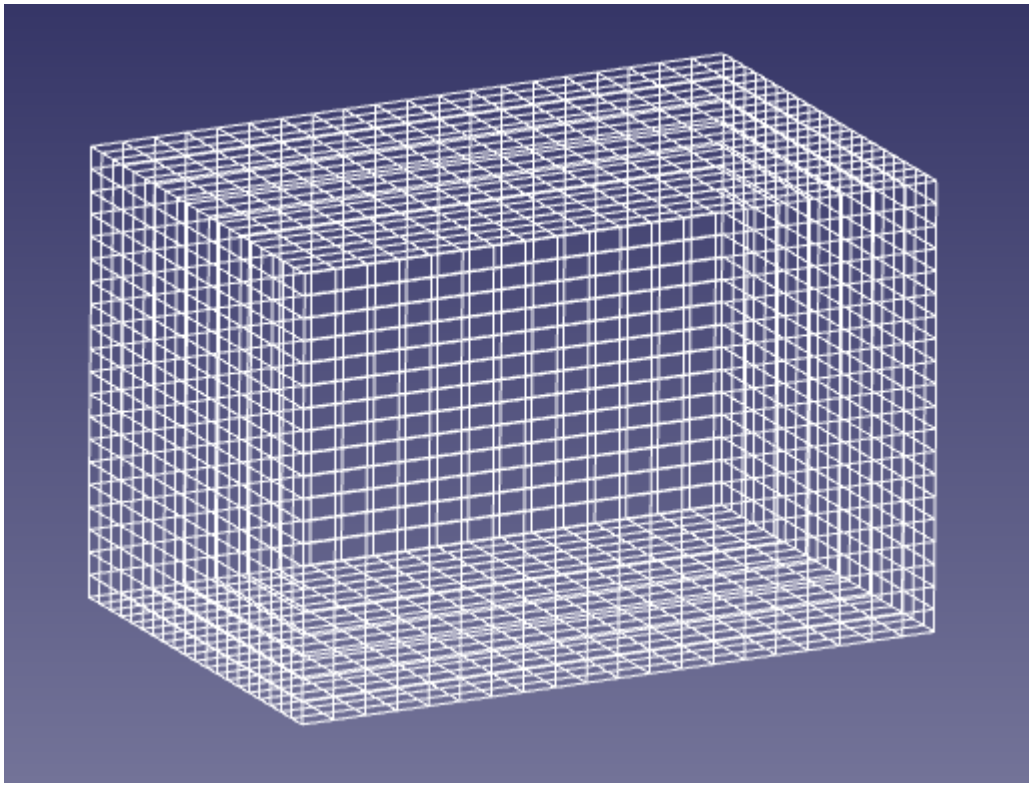
8.1.8.5.8.4 Layer «Initial grid», user interface



The **Initial grid** layer in the project tree

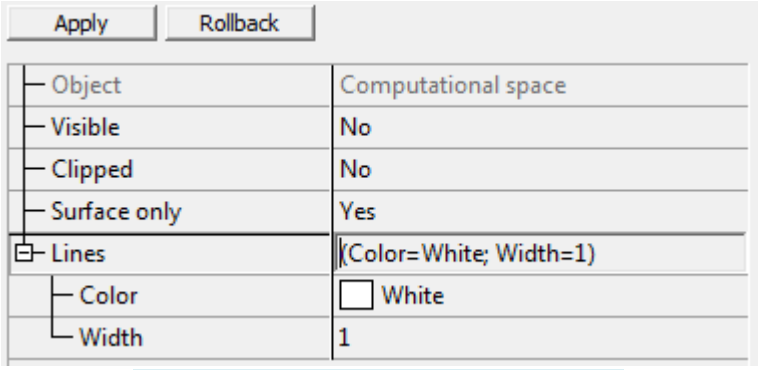
Initial grid layer is designed to display the initial grid. He makes visible cell of the computational grid zero. Furthermore, this layer can draw a line of intersection of faces of cells of the computational grid zero with solids.

Data for the construction of this layer is taken from the client part of the project, and the bed can be built to run on the computation.



Layer **Initial grid**

Parameters of the «Initial grid» layer



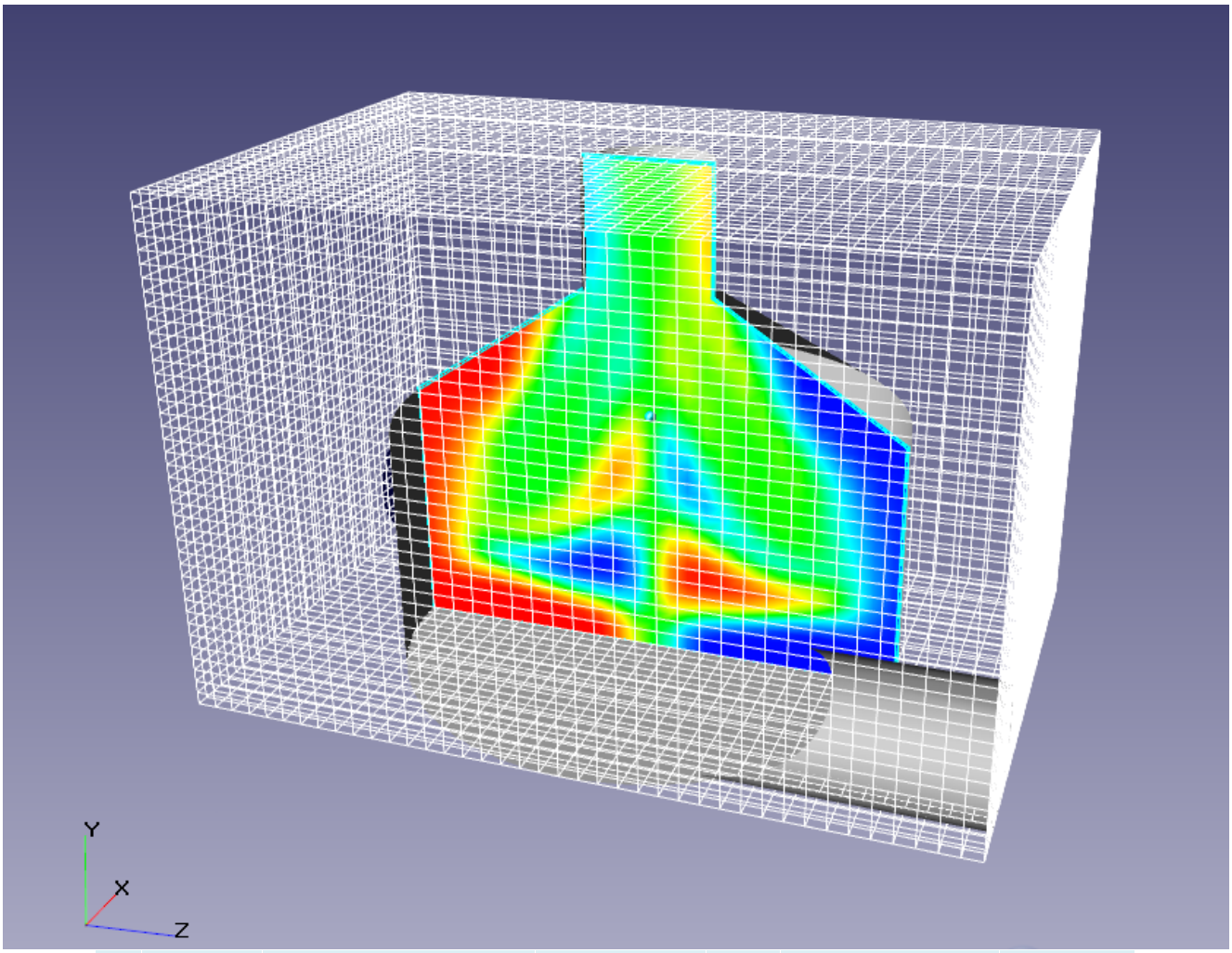
Properties window of the «Initial grid» layer

Layer parameters **Initial grid**:

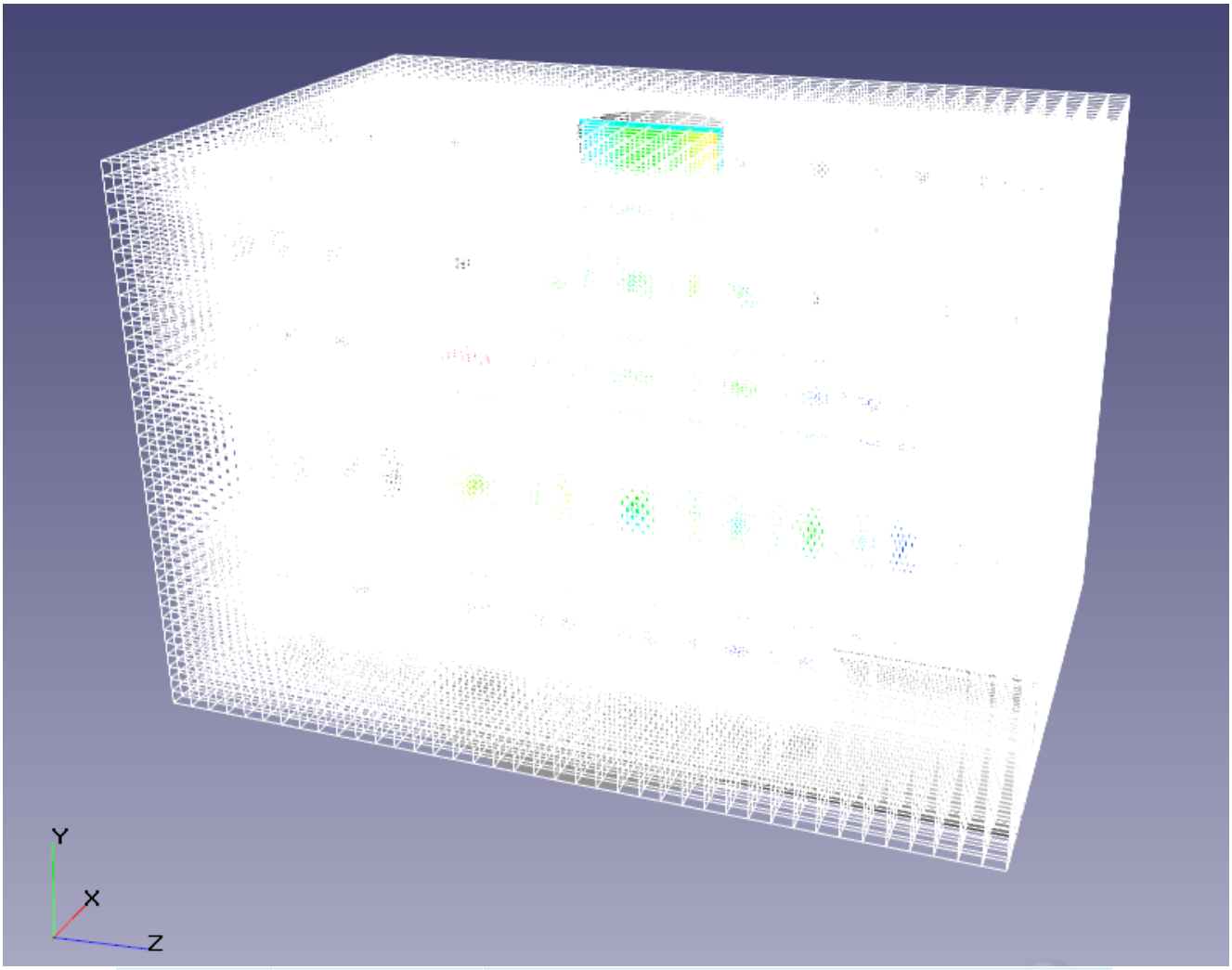
Parameter	Description
Object	Name of the object is associated with the element. Field information is not editable
Visible	• **No** – layer is not displayedin the **View** (if the layer is selected in the project tree) • **Yes** – layer is always displayedin the **View** window
Clipped	• **No** – section planes are not acting on the object • **Yes** – section planes cut to
Surface only	• **No** – on the grid in the entire volume of the computational domain (see Illustration) • **Yes** – show only the cells located on the surface of the parallelepiped is the boundary of the initial grid[1)]
Lines > Color	The choice of color of the grid lines (see illustration)
Line > Width	Select the thickness of the grid lines (see illustration)

[1)] If you are importing the geometry of the computational domain around it automatically builds a box that limits the geometry of the directions X, Y, Z. This parallelepiped is the basis for the construction of the initial grid. Grid

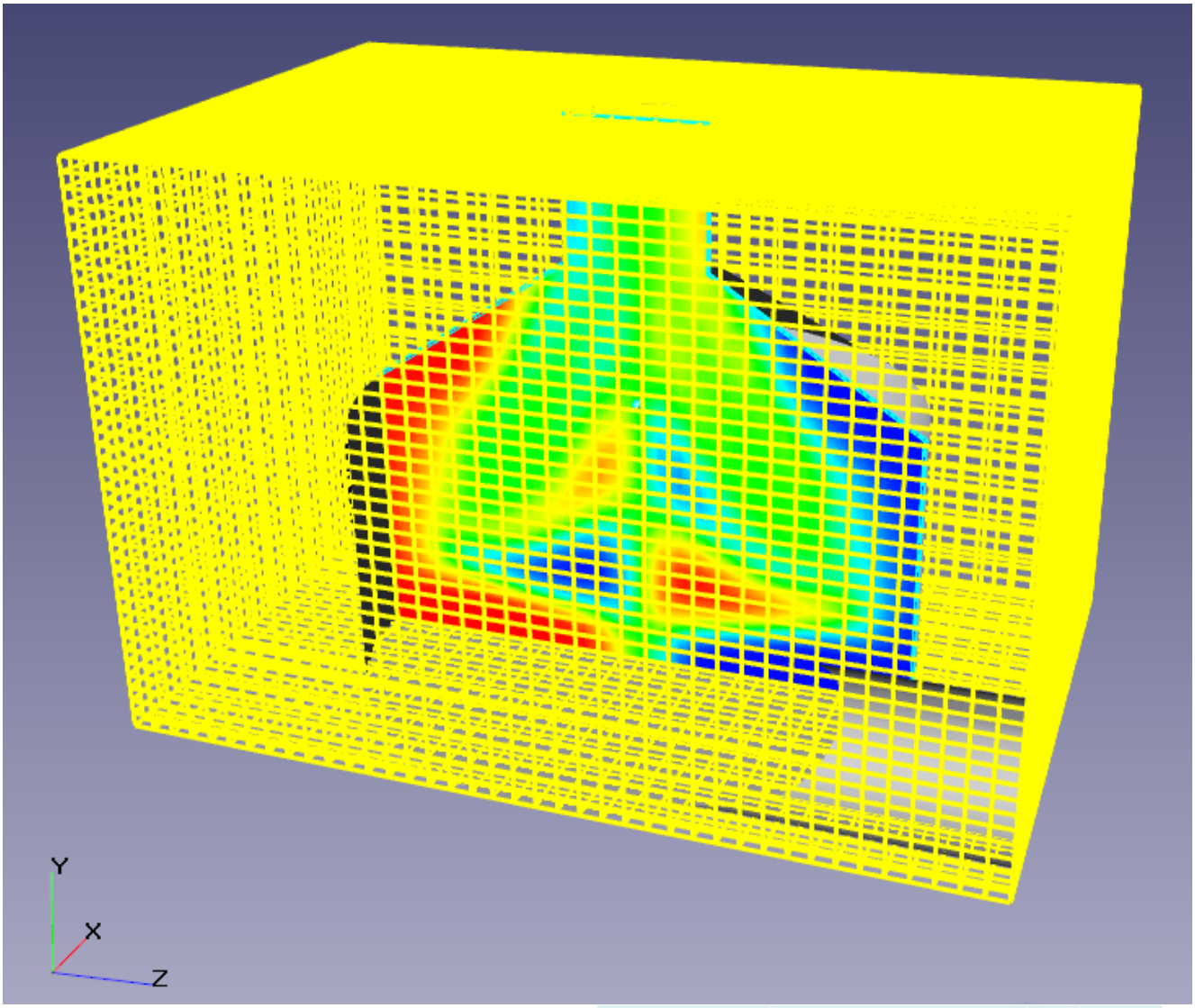
cell, trapped inside the computational domain, are design. Cells cut by the boundary of the computational domain are non-computational.



The **Initial grid** layer is displayed over a **Color contours** layer (**Surface only = Yes** is specified)

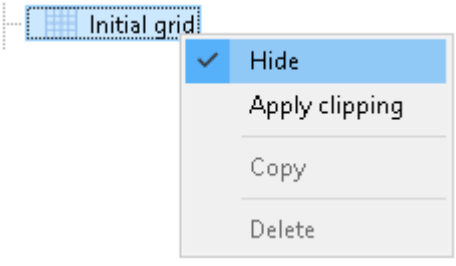


When you set **Surface only = No**, lines of the initial grid can interfere displaying other layers



The color and thickness of the lines can be configured (in this example **Lines > Color = Yellow** and **Lines > Width = 3** are specified)

Context menu of the «Initial grid» layer



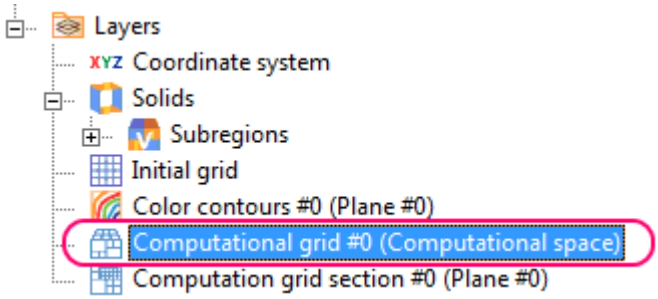
Context menu of the **Initial grid** layer

Context menu of the **Initial grid** layer:

Menu item	Description
Hide	☐ – the **Layer** is always displayed in the **View** window ☑ – the **Layer** is displayed in the **View** window, only when it is selected in the project tree
Apply clipping	☐ – clipping **Planes** do not affect the **Layer**

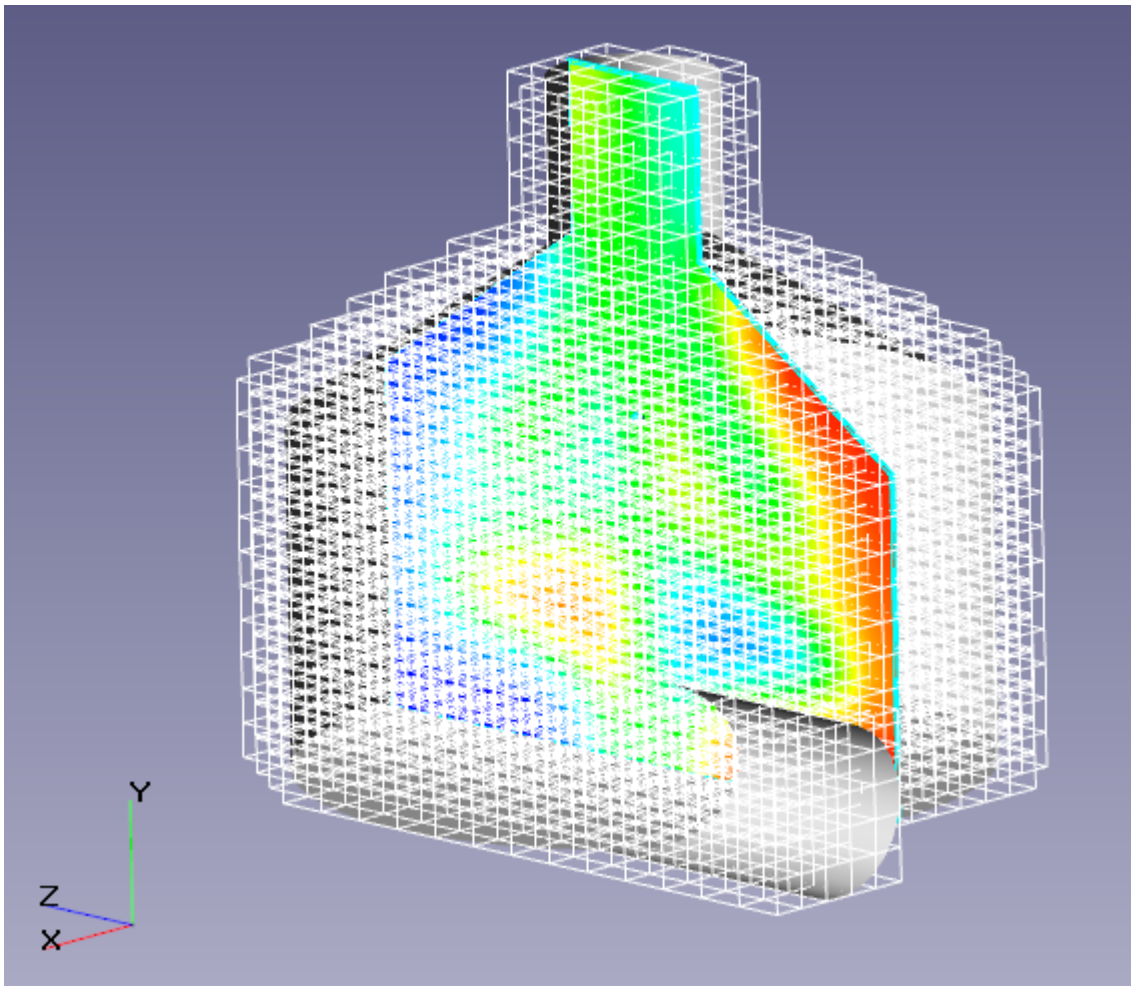
Menu item	Description
	– clipping **Planes** cut the **Layer**
Copy **Delete**	*These commands are not active because the **Initial grid** layer can not be copied or deleted*

8.1.8.5.8.5 Layer «Computational grid», user interface



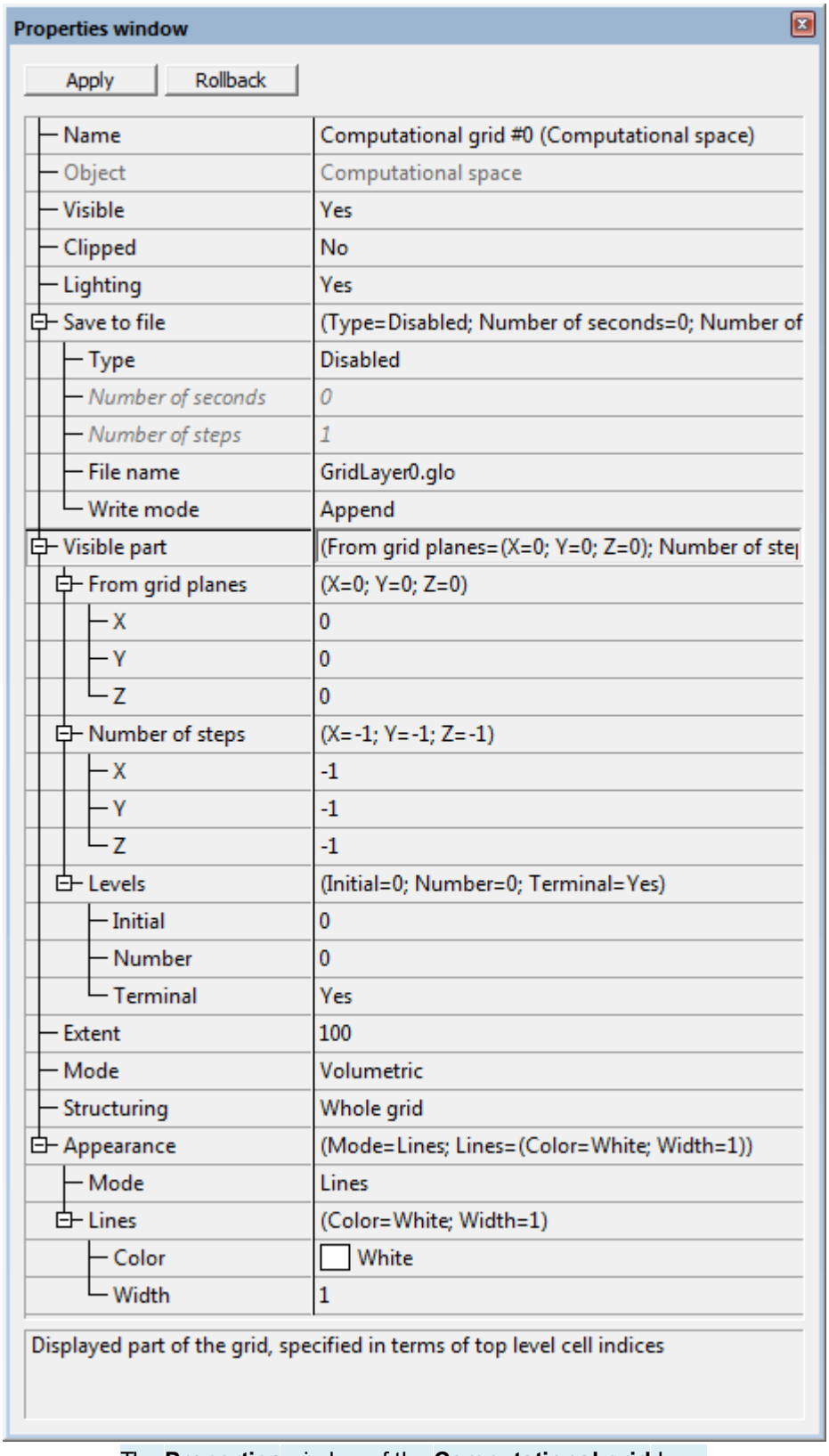
The **Computational grid** layer in the project tree

Layer with the calculated data **Computational grid** is designed to display the computational grid.



Layer **Computational grid**

Parameters of the "Computational grid" layer



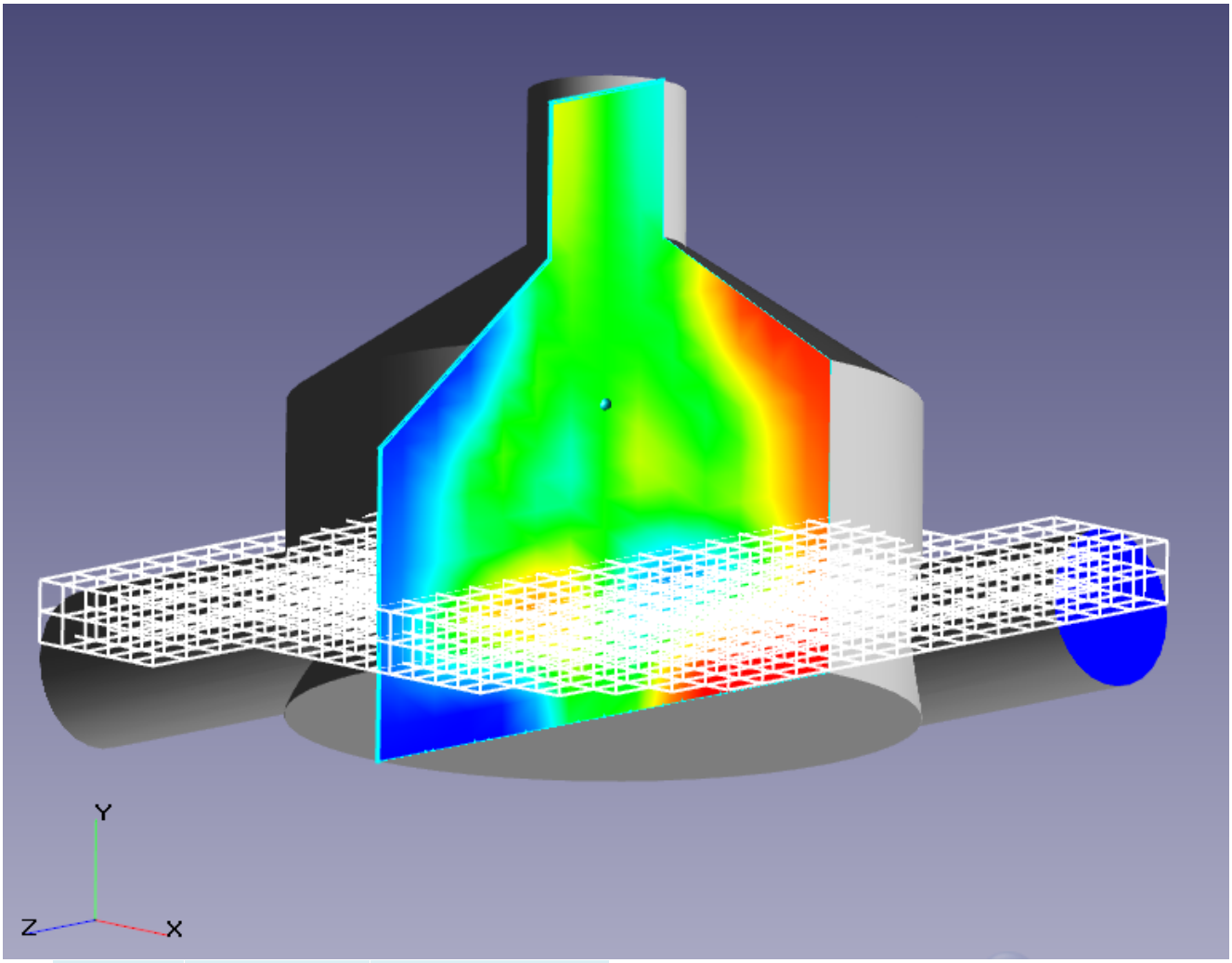
The **Properties** window of the **Computational grid** layer

Properties of the **Computational grid** layer:

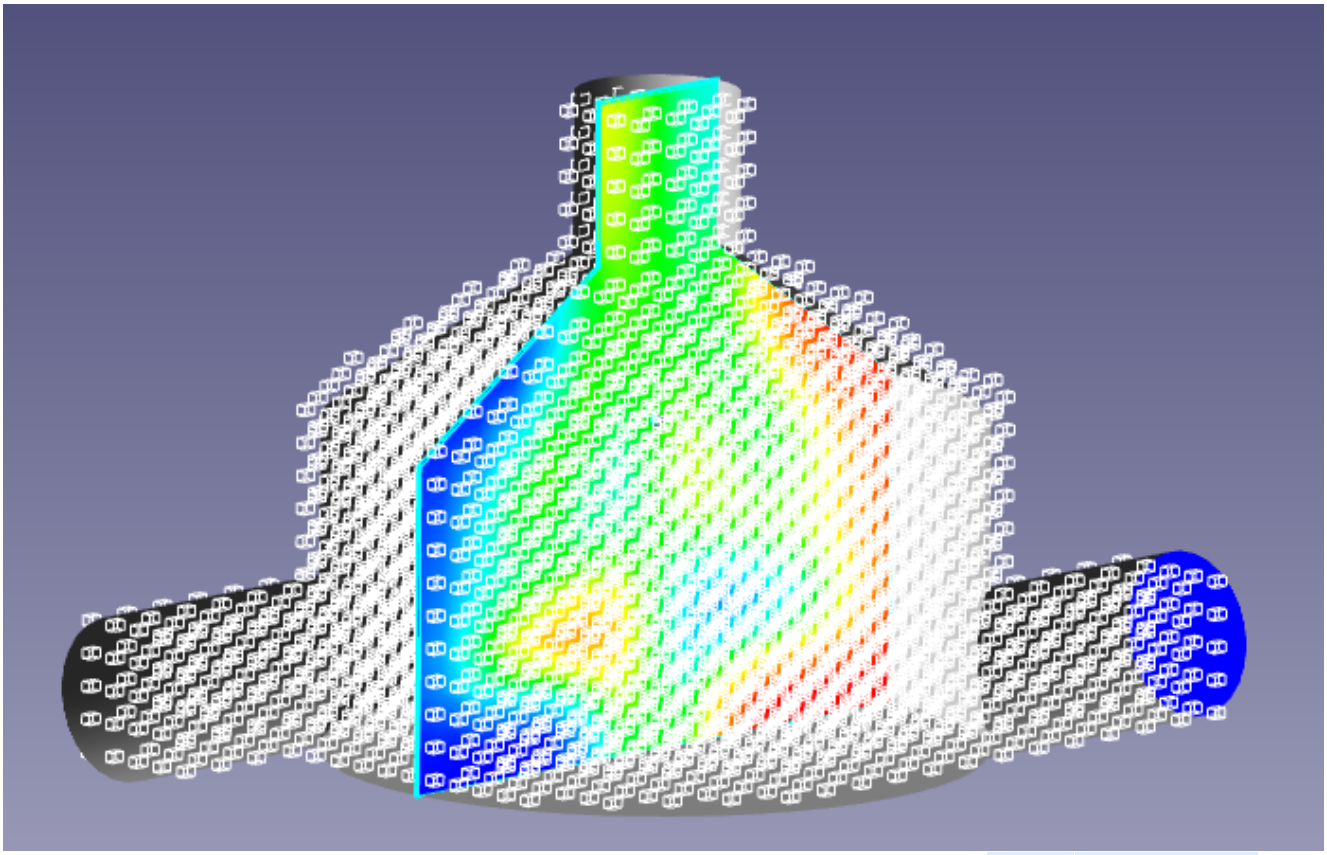
Parameter	Description
Name	Layer name (this option allows you to change the default title "**Computational grid #N (Computational space)**", formed from

Parameter	Description
	the name of the layer type, a number and the **Object**, on which the layer is built).
Object	See General properties of Layers.
Visible	
Clipped	
Lighting	
Save to File > ...	
Visible part	Group of parameters allows you to display only a certain limited range of grid cells.
Visible part > From grid planes > X	Number of the first display of the grid plane along the X, Y, and Z absolute coordinate system (counted from 0), see illustration.
Visible part > From grid planes > Y	
Visible part > From grid planes > Z	
Visible part > Number of steps > X	Display the grid step along the X, Y, and Z absolute coordinate system: • If **-1**, then displays all the cells in that direction • if **0**, the grid displays only one line of direction See illustration.
Visible part > Number of steps > Y	
Visible part > Number of steps > Z	
Visible part > Levels > Initial	Adaptation layer from which the computational grid is displayed
Visible part > Levels> Number	The number of display levels of adaptation (measuredfrom the start)
Visible part > Levels> Terminal	Displaying only real (terminal) cells. Set of real cells - the final result of the application of all the criteria of adaptation. Nonterminal - the cell has undergone fragmentation. Fragmented cell is not considered to be real. Possible options are: • **No**- display of all cells up to a specified level of adaptation, including cells, fragmented as a result of adaptation of the following levels or as a result of the intersection with the x and boundaries; • **Yes**- to show all whole cells up to a specified level of adaptation. Not displayed cells that have undergone further fragmentation as a result of adaptation of the following levels or as a result of the intersection with the boundaries
Extent	Cell size when displayed as a fraction of the original size, specified as a percentage (see illustration). A value of 100% corresponds to the displaying of the grid, which fills the entire space.
Mode	Possible options are: • **Volumetric** - displays all the cells in the volume; • **Surface** - display only the boundaries between processors or hypercells. Selecting this option makes sense only when **Structuring=Processors** or **Structuring=Hypercells** is defined. See illustration.
Only inner	If **Mode=Surface**,variants are possible: • **No**- displaying the boundaries of the outer surface of the grid blocks belonging to different processors or hypercell m • **Yes**- display only the inner surfaces of the boundaries of the grid blocks belonging to different processors or hypercell m See illustration.
Structuring	Possible options are:

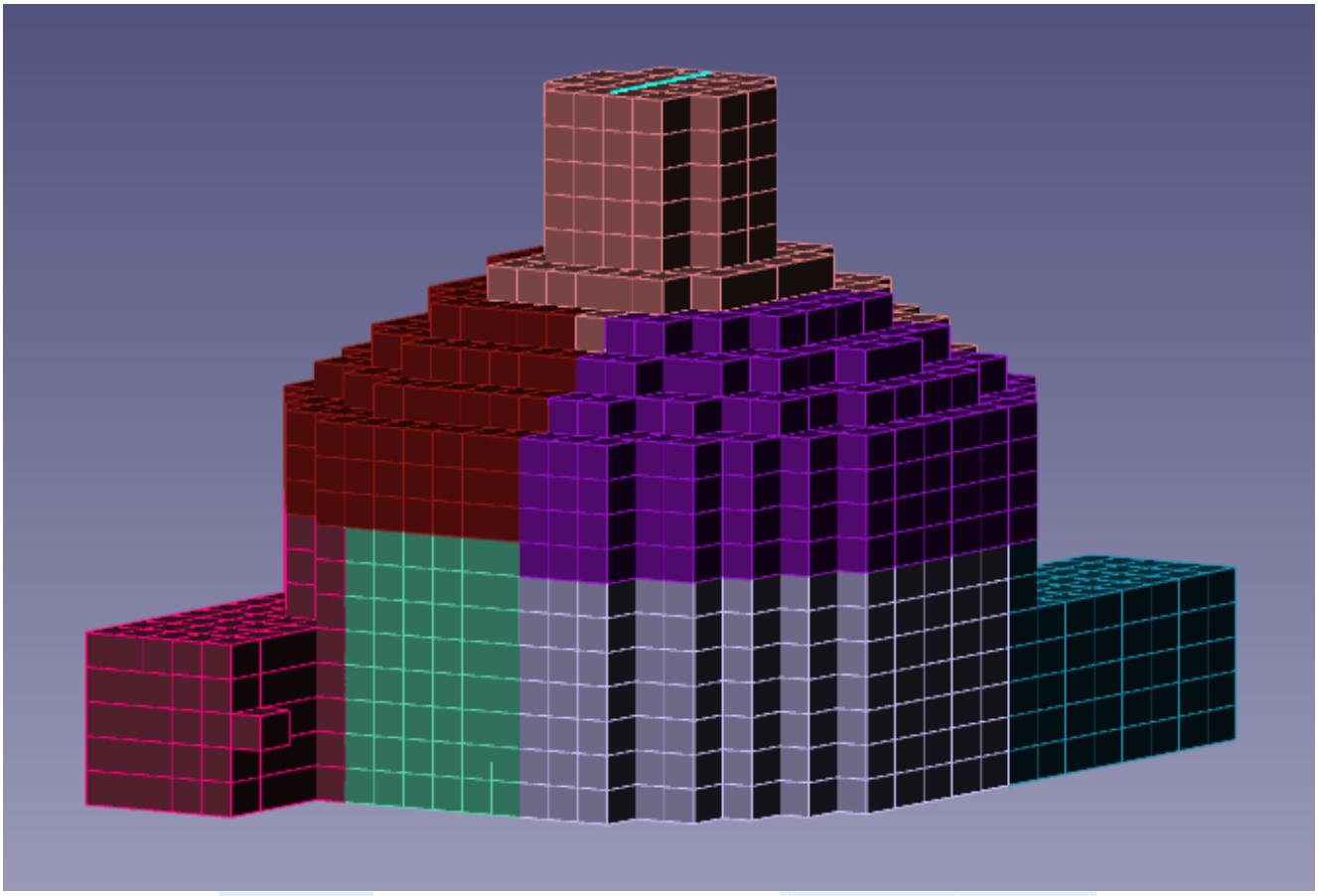
Parameter	Description
	• **Whole grid** - shows the entire grid • **Processors** - displays the distribution of grid processors • **Hypercells** - mapping the distribution of grid hypercells See illustration.
Appearance	General settings for displaying the layer. These parameters are available when **Structuring** = **Whole grid**.
Appearance> Method	See General properties of Layers.
Appearance> Lines> Color	
Appearance> Line> Thickness	
Appearance> Shading> Color	
Appearance> Shading> Opacity	
Hypercells	Display settings for hypercells. These parameters are available when **Structuring** = **Hypercells**.
Hypercells > Visible	Numbers that appear hypercells. These numbers are given as a list whose elements are separated by commas (list item can be a number or a cell range of cells). The remaining hypercells will not be displayed. For example, the list of **"0-1,6-8"** hypercells displays 0, 1, 6, 7 and 8. See illustration.
Hypercells > Mode	Drawing method for hypercells. Possible options are: • **Lines** • **Fill** • **Lines and fill**
Hypercells > Lines > Color> [N]	The color and line thickness for displaying hypercells the N-th (N = 0, 1, 2, ...)
Hypercells > Line > Width > [N]	
Hypercells > Fill > Color > [N]	Color shading for display hypercells N-th (N = 0, 1, 2, ...)
Hypercells > Fill > Opaqueness > [N]	Opacity maps the N-th hypercells (N = 0, 1, 2, ...) is given as a percentage. See illustration.
Hypercells > Processors > [N]	Allocation the hypercells over processors. These parameters are read-only.
Processors	Display settings for allocation the grid over processors. These parameters are available when **Structuring** = **Processors**.
Processors > Visible	Parameters, similar to the corresponding parameters in the group **Hypercells**
Processors > Mode	
Processors > Lines > Color > [N]	
Processors > Line > Width > [N]	
Processors > Fill > Color > [N]	
Processors > Fill > Opaqueness > [N]	



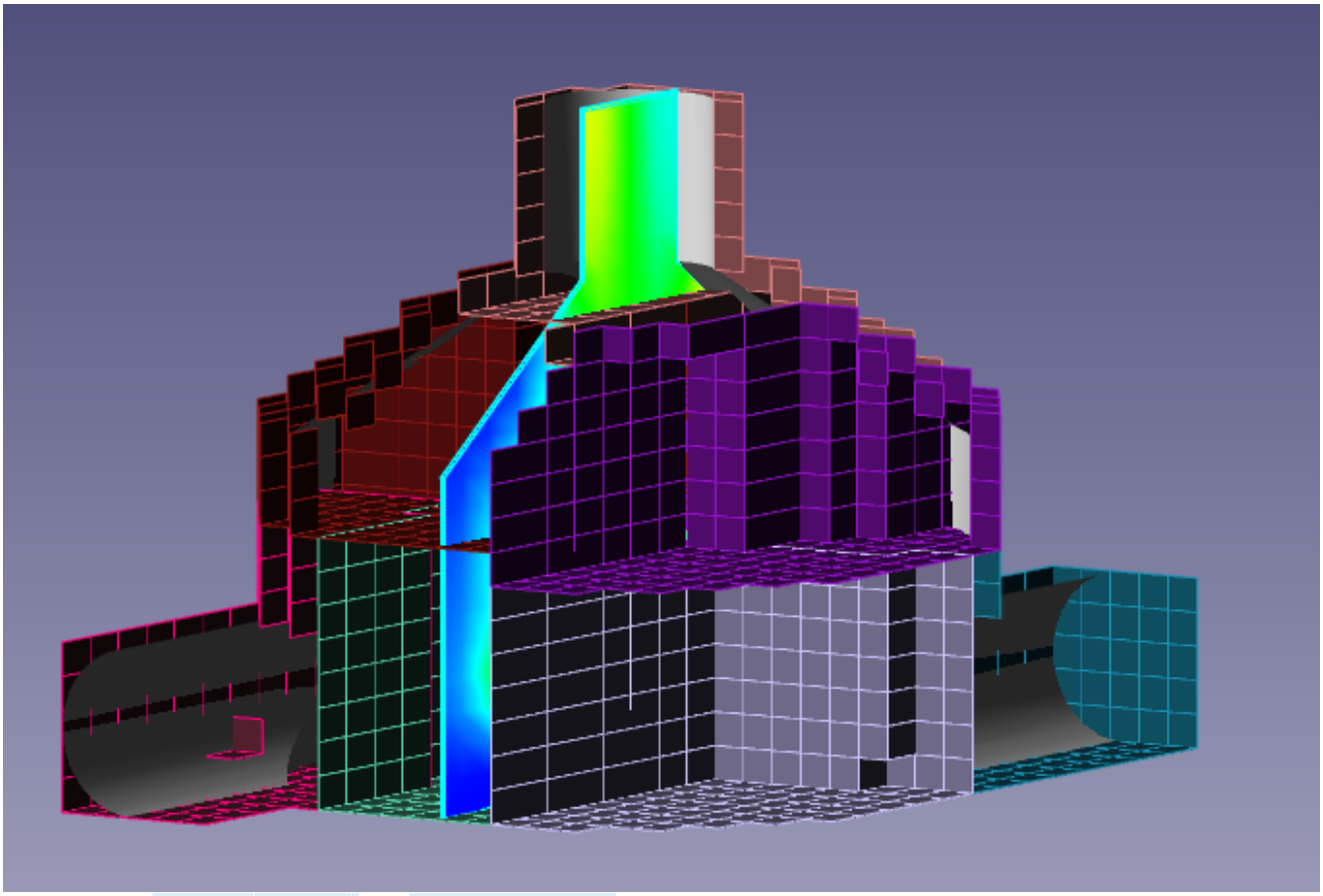
Parameters **Visible part > ...** allows you to display only a certain range of cells of the computational grid (In this example the parameters are set as **From grid planes > X=0**, **Y=3**, **Z=0**, **Number of steps > X =-1**,**Y=2**, **Z=-1**)



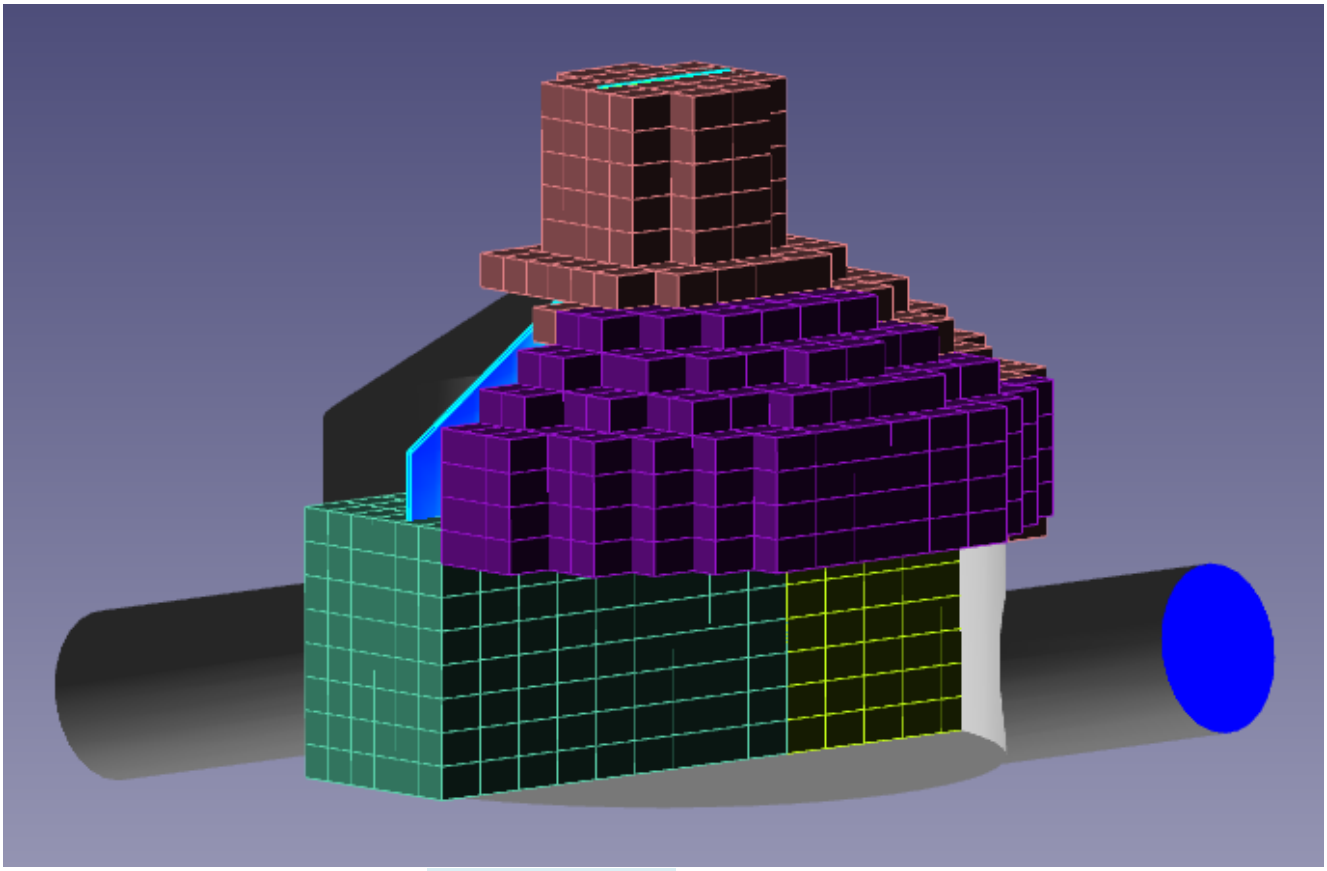
Images of the computational grid cells can be reduced by setting the parameter **Extent** (in this example **Extent=30**)



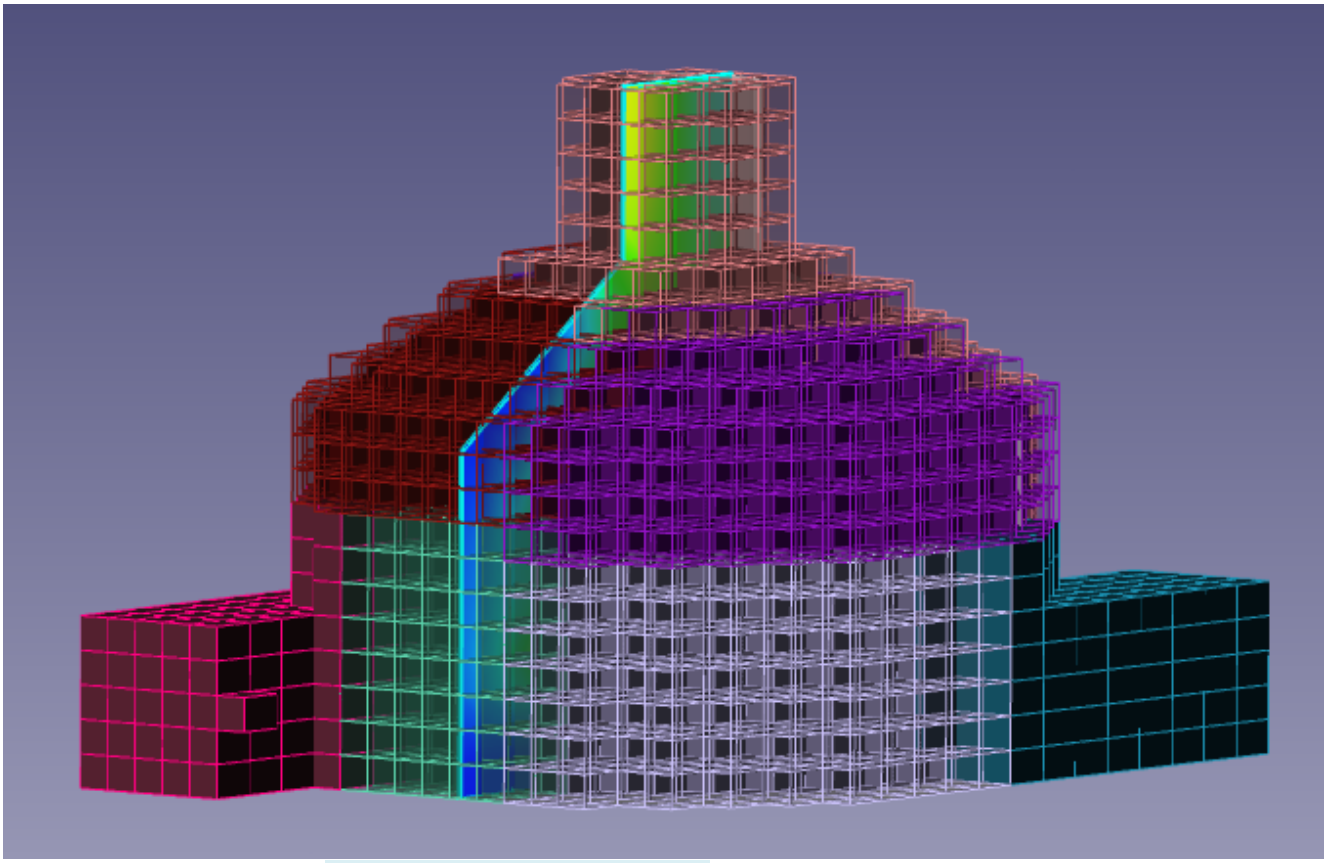
Displaying of the distribution grid over hypercells (**Structuring = Hypercells**)



Specifying **Mode = Surface** and **Only inner = Yes**, you can display only the borders of adjacent blocks of the computational grid belonging to different hypercell

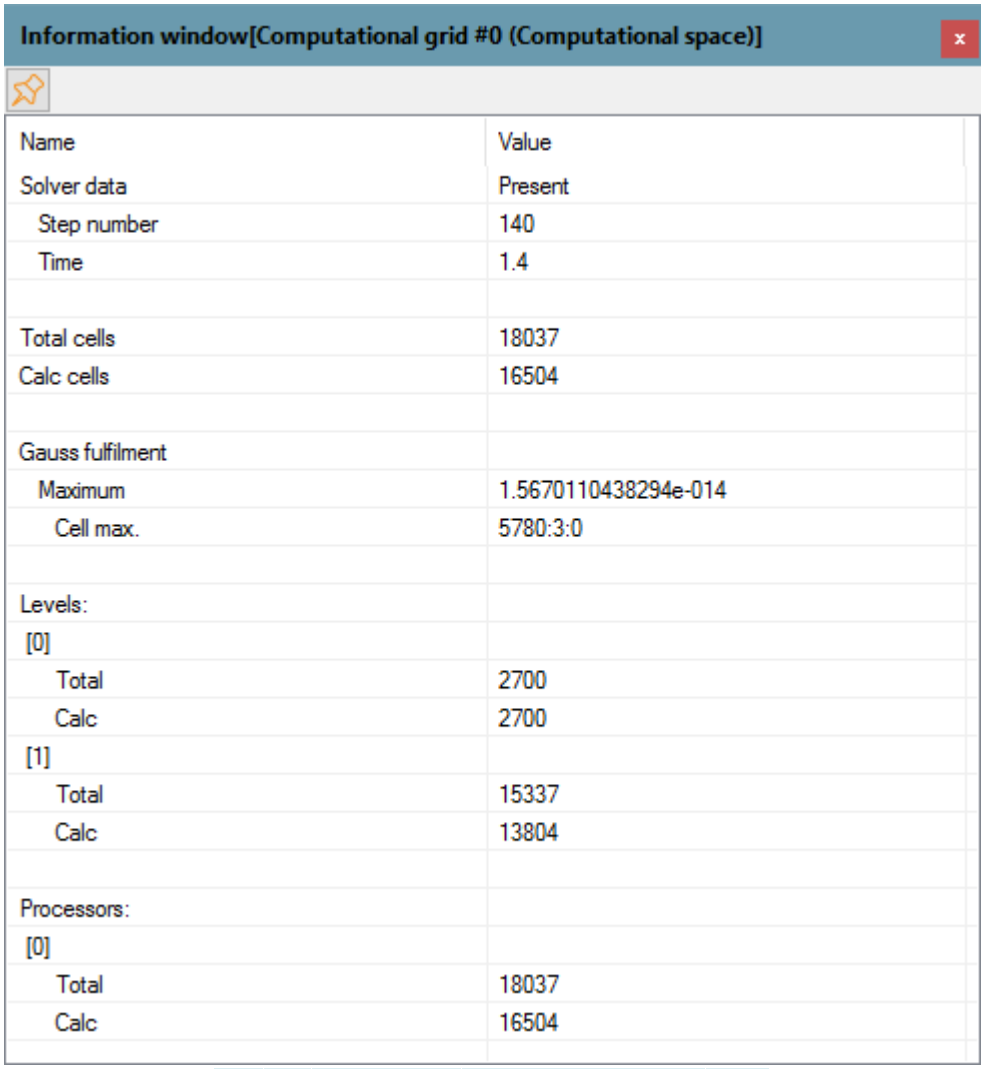


Customize the list **Hypercells > Visible**, you can display only some hypercell



Configuring the settings **Hypercells > Opaqueness > [N]**, you can display some hypercells as translucent

Window «Info» of the «Computational grid» layer



The Info window of a **Computational grid** layer

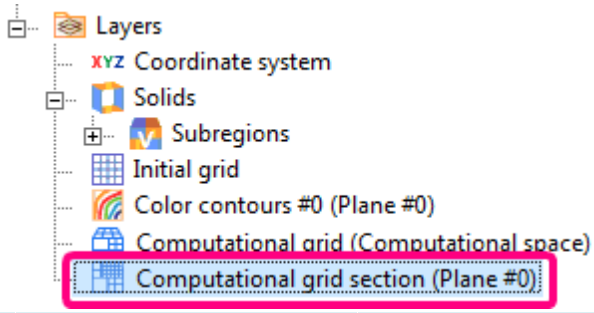
Data item	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current simulated time
Total cells	The total number of cells in the computational grid
Calc cells	Calculated number of grid cells
Gauss fulfillment	According to the Gauss' formula (the divergence theorem, Gauss-Ostrogradsky theorem), the sum of products of areas of closed polyhedron's faces times normal vectors to the faces must be zero. This sum of the products is calculated for the cells of the computational grid to test that cells are closed, and the sum is normalized by dividing by the surface area of the cell. If this value is greater than 10^{-7}, then this cell has been probably built with an error and you need to fix the cause of the error.

Data item	Description
	When Sliding surfaces are used, is is allowed that this parameter increases up to no more then 0.01, which is not an error but the effect of not precise mapping of facets of the unturned and turned positions of the sliding surface. You can check the correctness of the generated grid by this parameter calculated before inserting the Sliding surfaces.
Gauss fulfillment > Maximum **Gauss fulfillment > Cell max.**	The maximal (among all cells in the computational grid) normalized value of sum of products of the areas of the cell's faces times their normal vectors (see above) and the index of the cell where this maximal value is reached.
Levels: [N]> Total	The total number of cells in the level of splitting N
Levels: [N]> Calc	Calculated number of grid cells the level of splitting N
Processors: > [M] > Total	The total number of cells per processor M
Processors: > [M] > Calc	Number of the computational grid's cells on the processor M

Components of the text file for the layer "computational grid"

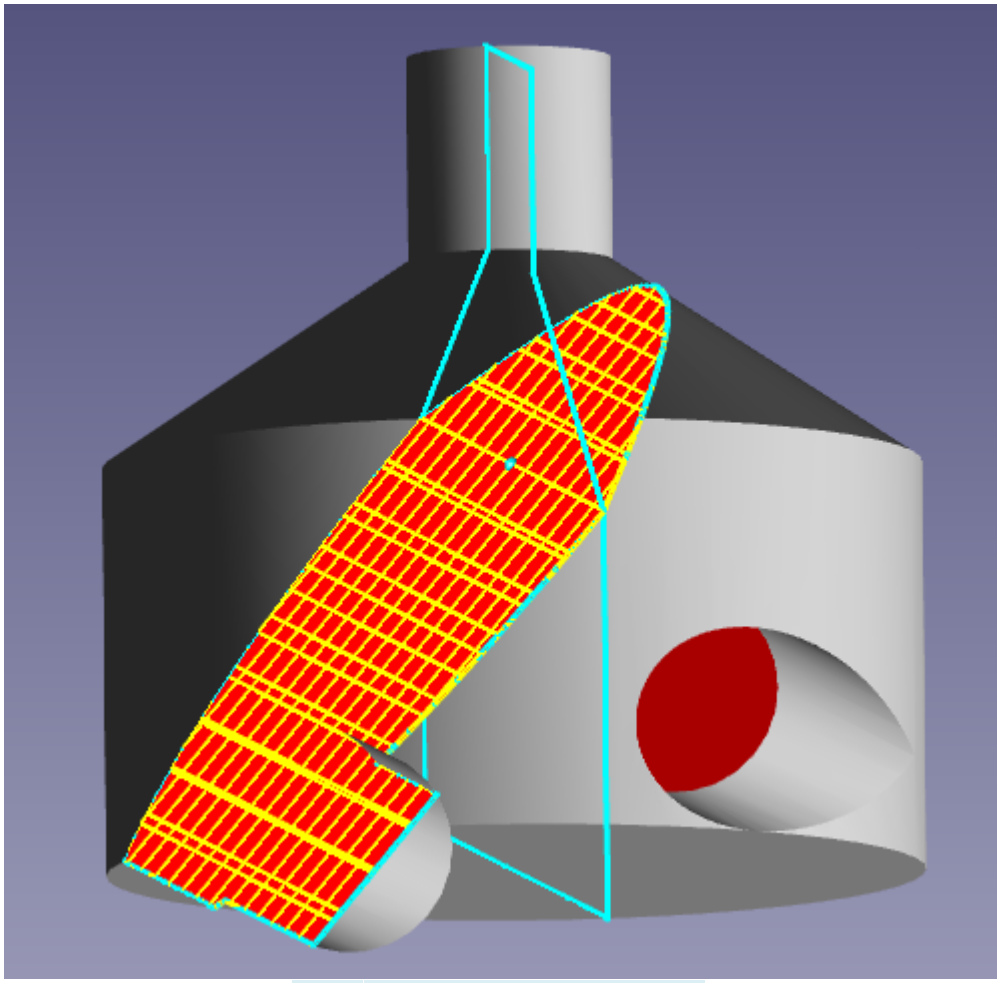
Component	Description
Step	Step number
Time	Time
TotalCells	Total number of cells
CalcCells	Number of computational cells
Levels	The number of levels of adaptation
Total i	The total number of cells of i-th level of adaptation
Calc i	Number of computational cells i-th level of adaptation
...	
Procs	
MaxGauss **MaxGaussCell**	Values of the parameters **Gauss fulfillment > Maximum** and **Gauss fulfillment > Cell max.** from the **Info** window (see above).

8.1.8.5.8.6 Layer «Computational grid section», user interface



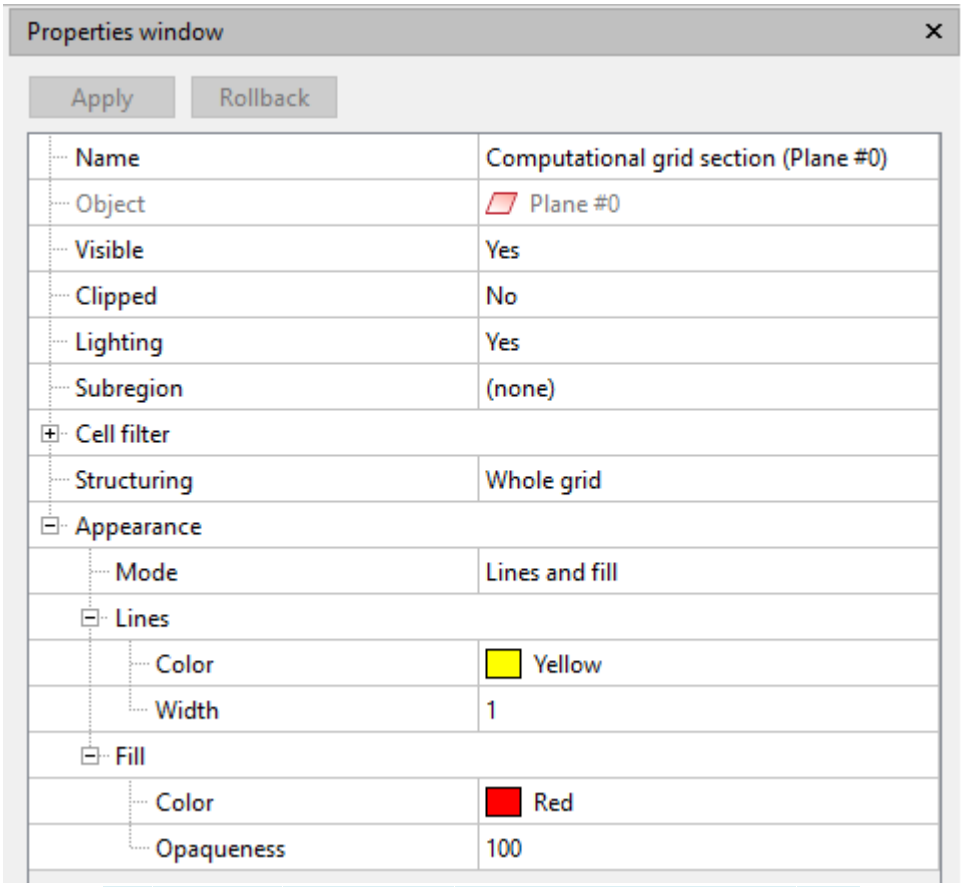
The **Computational grid section** layer in the project tree

Layer **Computational grid section** is intended to show the computational grid of cells located in a some **Plane** or on the surface of some **Object** on which this layer is built, i.e. this layer shows the cross section of the computational grid plane or surface.



Layer **Computational grid section**

Parameters of the «Computational grid section» layer

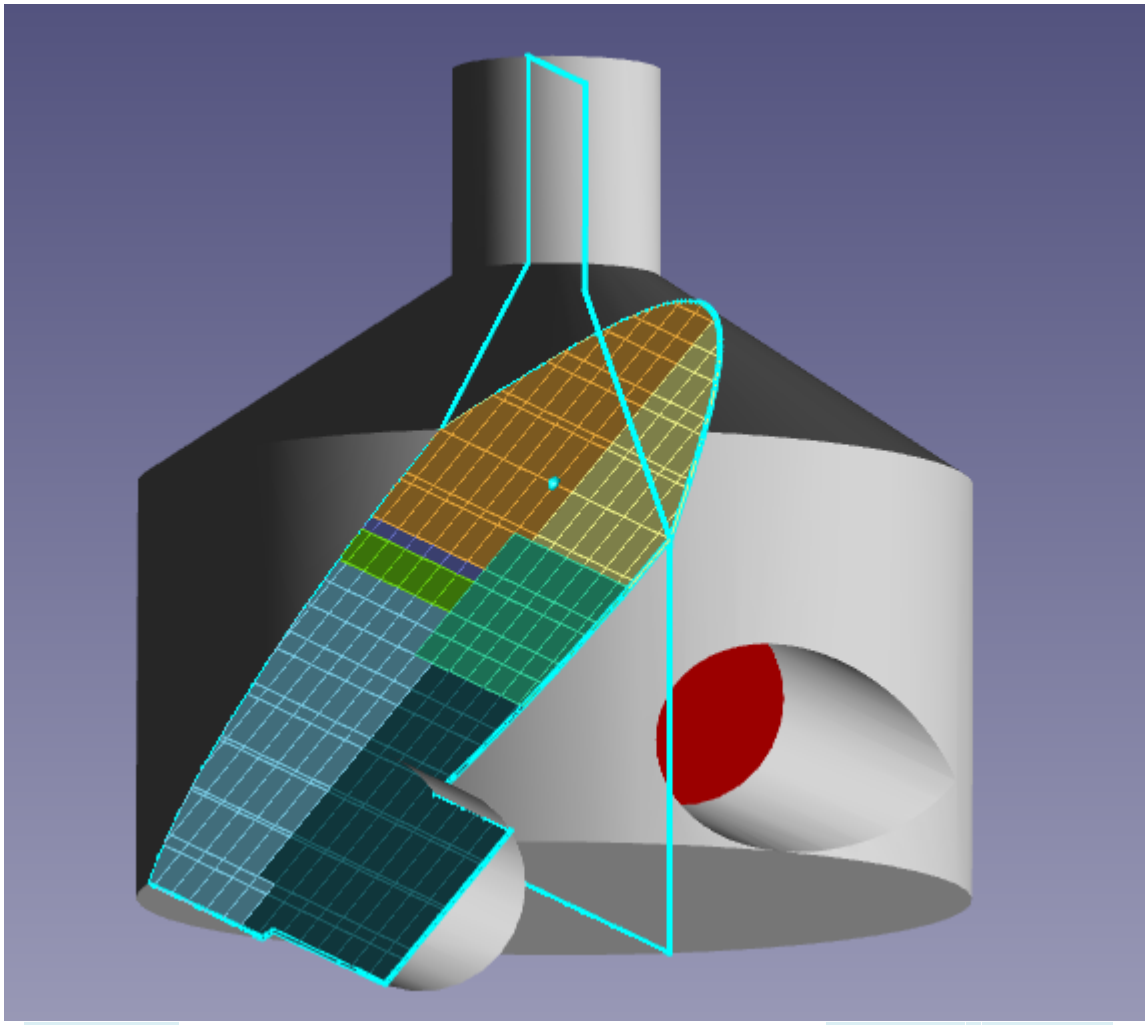


The **Properties** window of the **Computational grid section** layer

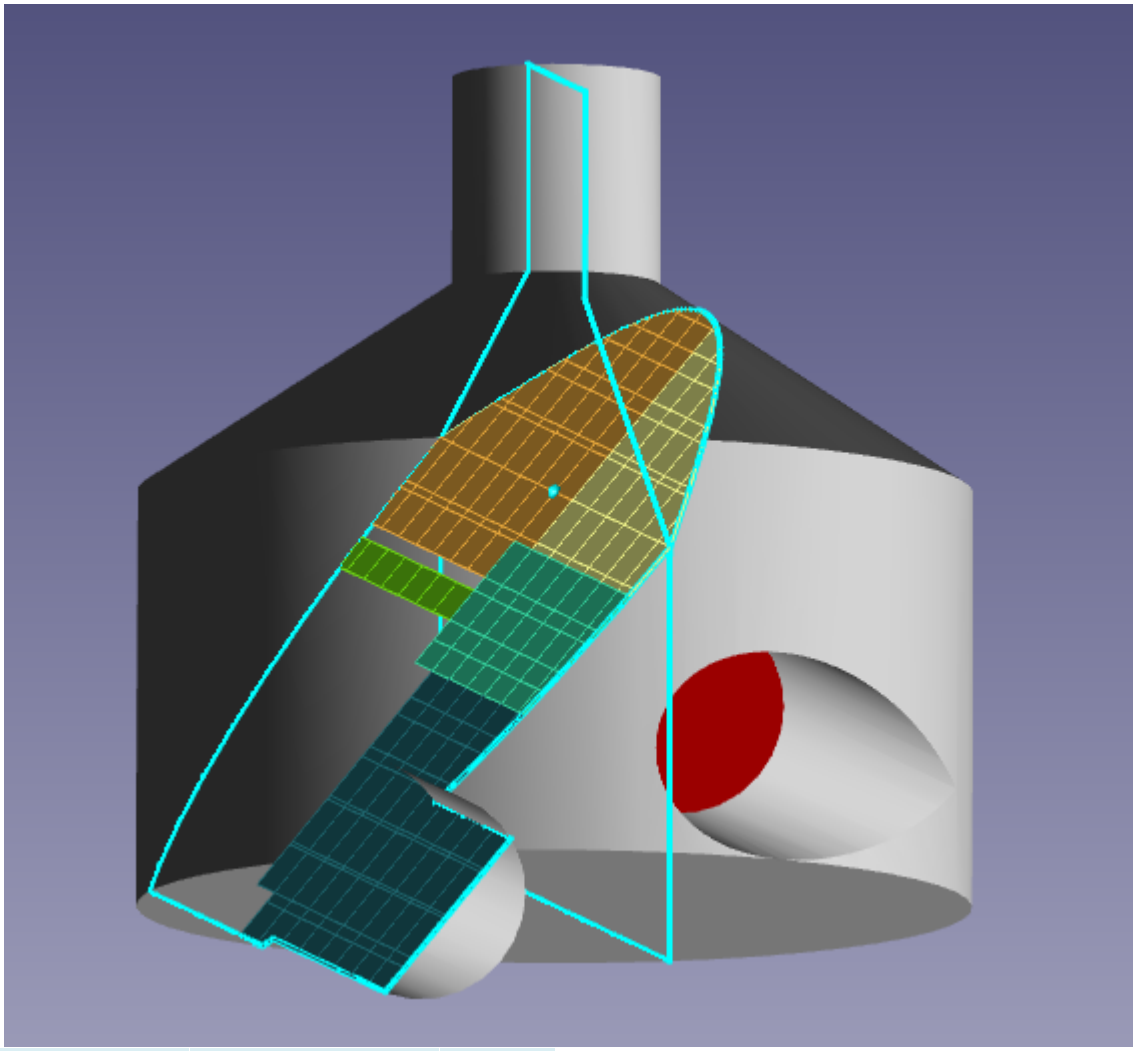
Parameters of the **Computational grid section** layer:

Parameter	Description
Name	Name of the layer (this parameter allows you to change the default name, which is formed based on the layer's type, a number, and the **Object**, on which the layer is built).
Object	See General properties of Layers.
Visible	
Clipped	
Lighting	
Subregion	Limitation layer of a given subregion
Cell filter > Small	Defines whether to display separate small cells or display together with their their acceptor cells.
Cell filter > Vacuum	Defines whether to display vacuum cells (see illustration)
Structuring	Possible options are: • **Whole grid** – shows the entire grid • **Processors** – displaying the distribution grid processors • **Hypercells** – displaying the distribution grid hypercell See illustration.
Appearance	Default settings for the display layer. This group is available when you set **Structuring=Whole grid**.
Appearance> Method	See General properties of Layers.
Appearance> Lines > Color	

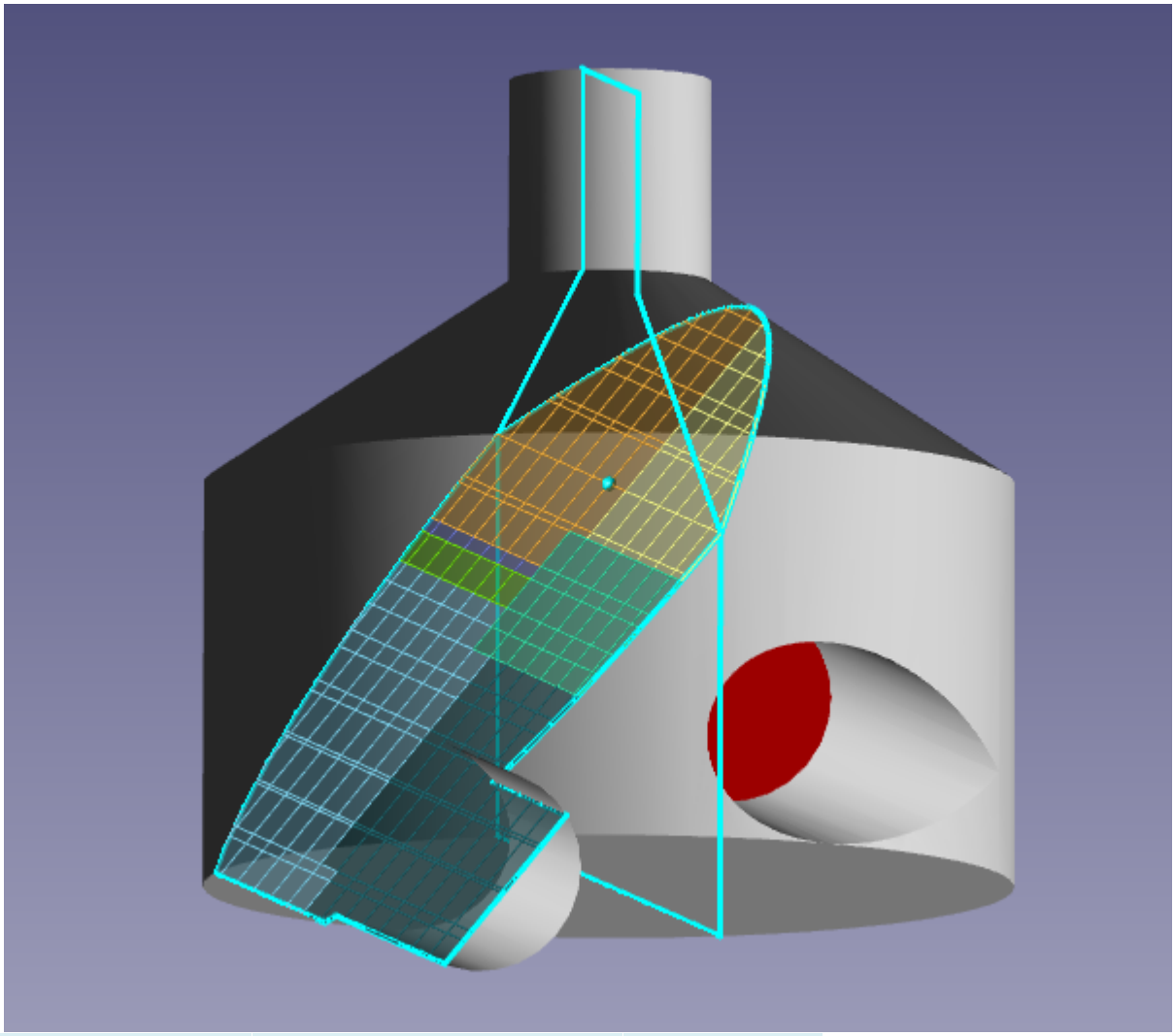
Parameter	Description
Appearance> Lines > Width	
Appearance> Fill > Color	
Appearance> Fill > Opaqueness	
Hypercells	Customize the display of the cross sections hypercell **plane** or surface of **the** object. This group is available when you set **Structuring**=**Hypercells**.
Hypercells > Visible	Numbers that appear hypercell cross sections. These numbers are given as a list whose elements are separated by commas (list item can be a number or a cell range of cells). The remaining section hypercells will not be displayed. For example, the list of **"0-1,6-8"** displays cross-sections of hypercells 0, 1, 6, 7 and 8. See illustration.
Hypercells > Mode	Drawing method of sections hypercells. Possible options are: • **Line** • **Fill** • **Lines and fill**
Hypercells > Lines > Color > [N]	The color and thickness of lines to display the N-th section hypercell (N = 0, 1, 2, ...)
Hypercells > Lines > Width > [N]	
Hypercells > Fill > Color> [N]	Fill color for displaying the N-th section hypercell (N = 0, 1, 2, ...)
Hypercells > Fill > Opaqueness > [N]	Opacity maps the N-th section hypercells (N = 0, 1, 2, ...) is given as a percentage. See illustration.
Processors	Display Settings distribution grid across processors. This group is available when you set **Structuring**=**Processors**.
Processors > Visible	Parameters, similar to the corresponding parameters in the group **Hypercells**
Processors > Mode	
Processors > Lines > Color > [N]	
Processors > Lines > Width > [N]	
Processors > Fill > Color > [N]	
Processors > Fill > Opaqueness > [N]	



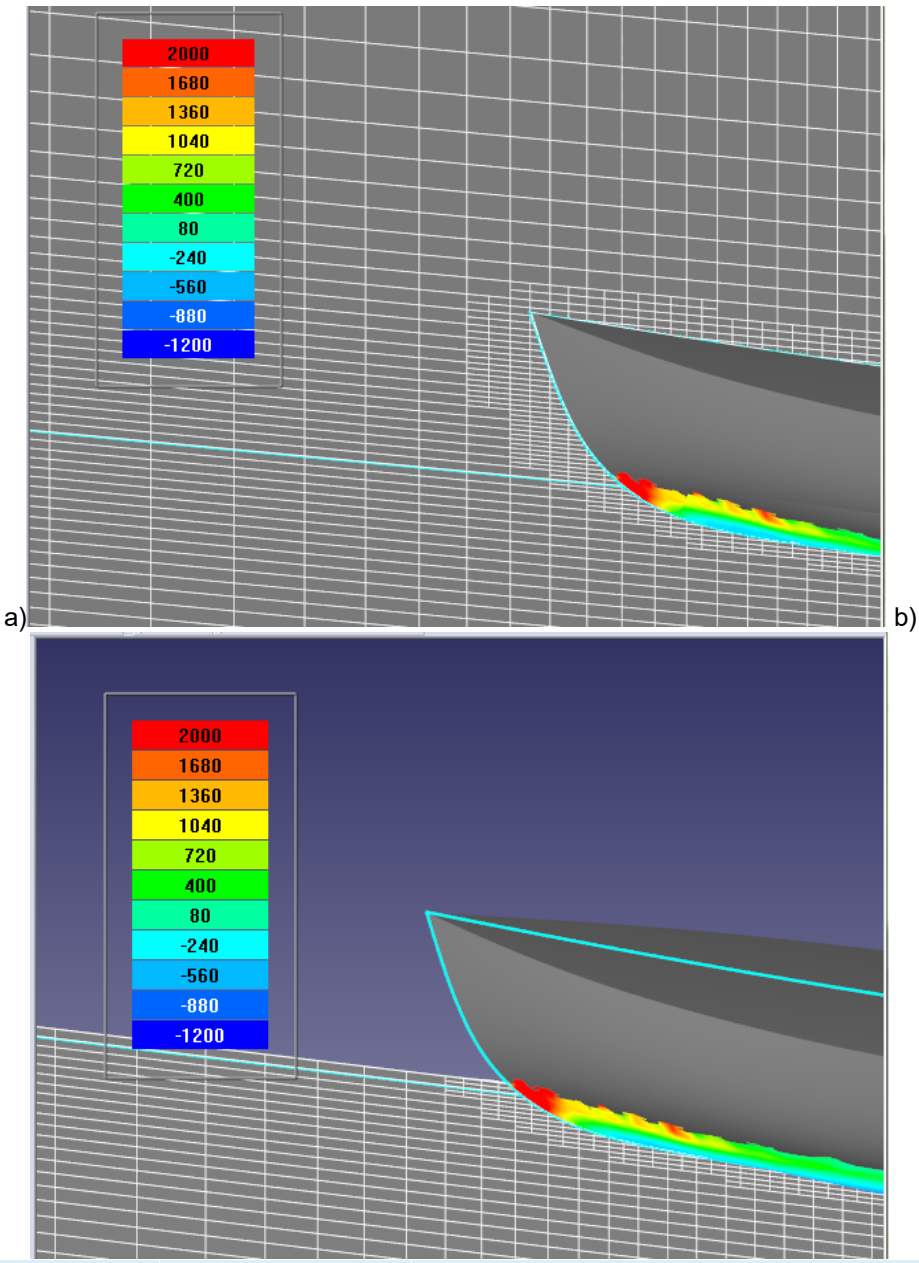
Displaying of the cross sections hypercells computational grid plane (**Structuring**=**Hypercells**)



Configuring the setting **Hypercells > Visible**, you can display only some of the sections of the plane hypercells



Configuring the settings **Hypercells > ... > Opacity > [N]**, you can display all or some section hypercells as translucent

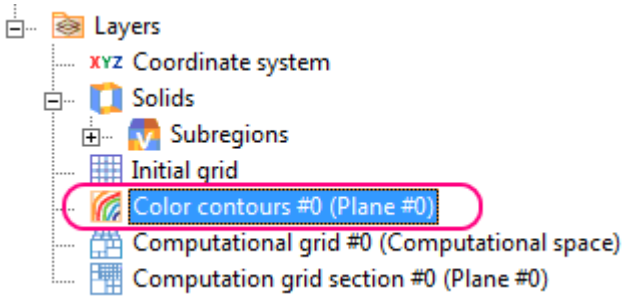


Use of the setting **Cell filter > Vacuum**: a) vacuum cells are displayed; b) not displayed

Window «Info» of the «Computational grid section» layer

The Info window of the layer **Computational grid section** contains no data.

8.1.8.5.8.7 Layer «Color contours», user interface

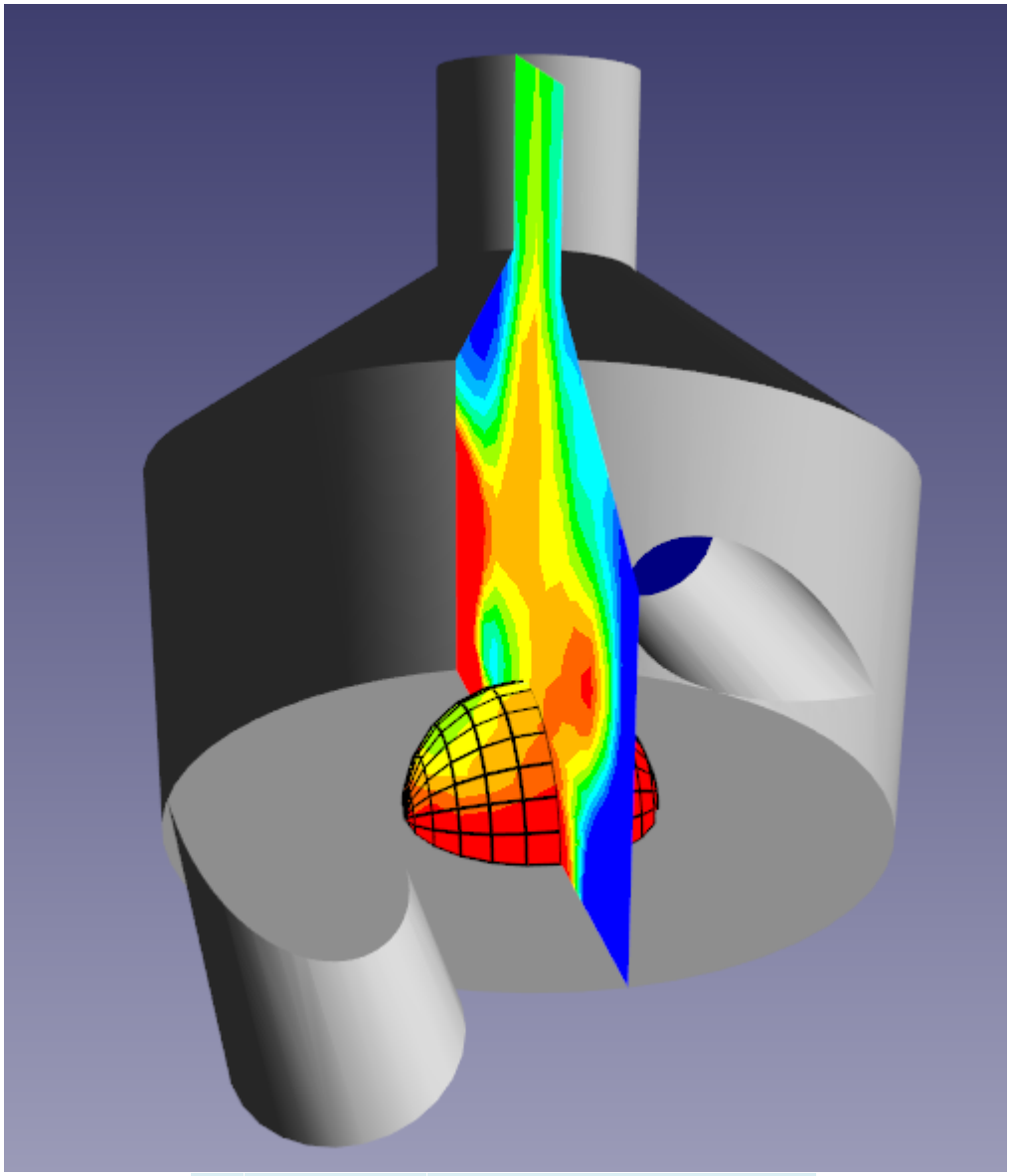


The **Color contours** layer in the project tree

The **Color contours** layer is designed to display the distribution of scalar*) variable using the color transitions.

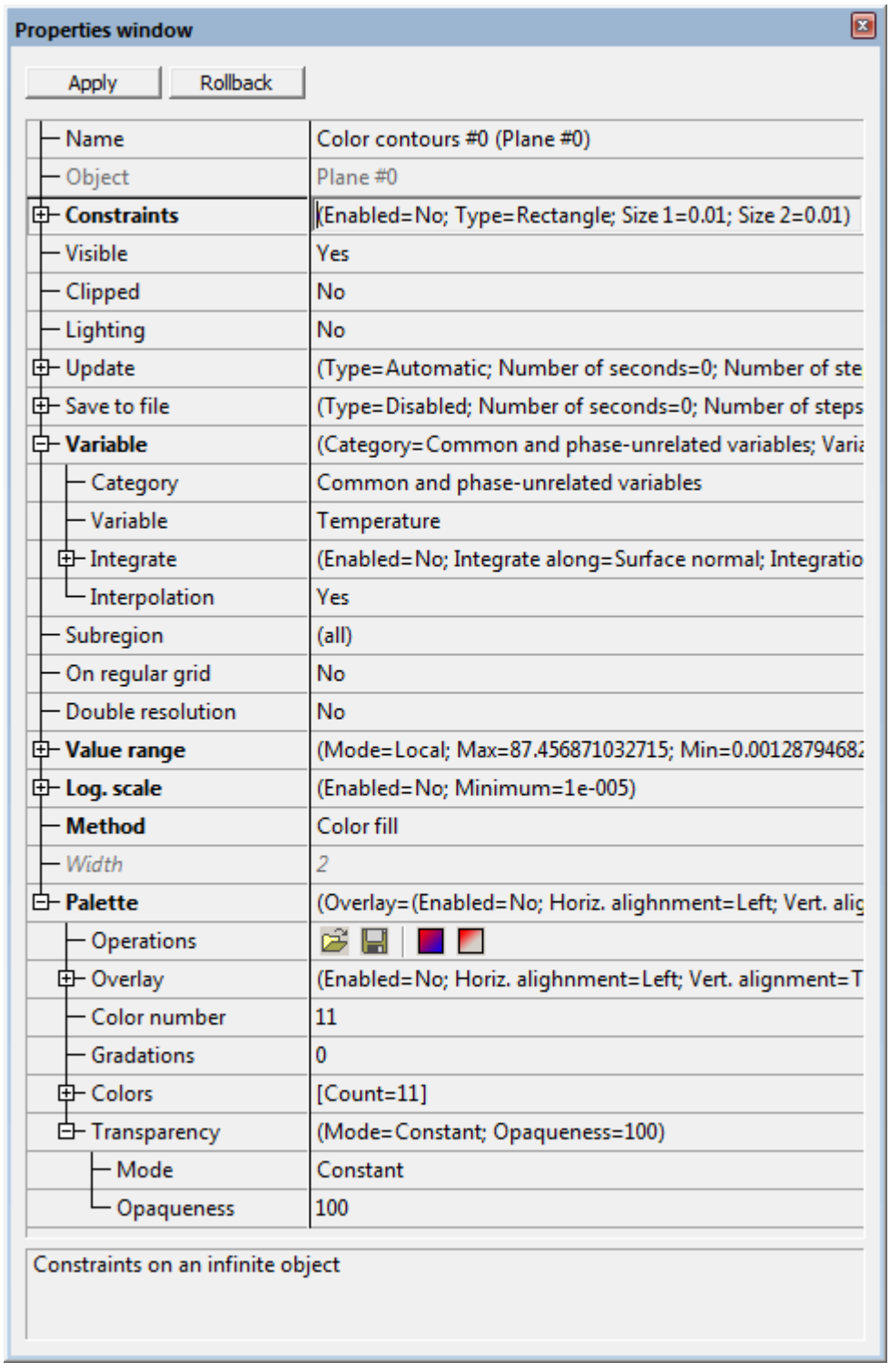
Note:

*) For the vector variable used her unit.



Two **Color contours** layers displaying the temperature
(one of them is built on a **Plane**, another - on a **Sphere**; both the layers use the same **Value range** specified manually)

Parameters of the «Color contours» layer

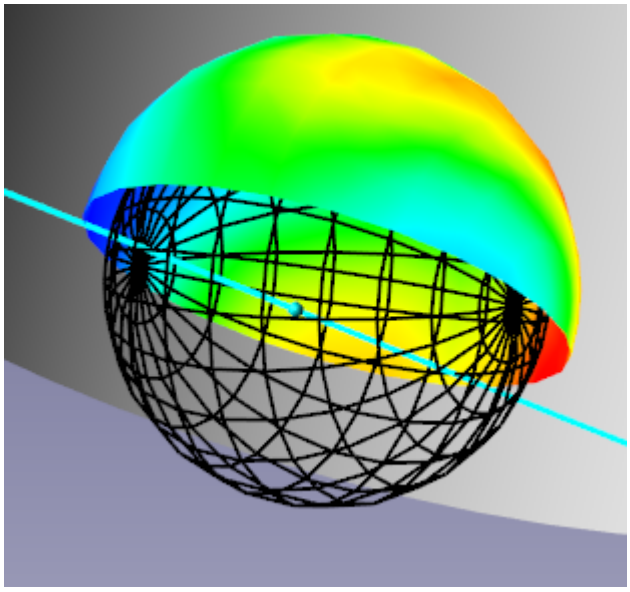


The **Properties** window of the **Color contours** layer

Parameters of the layer **Color contours**:

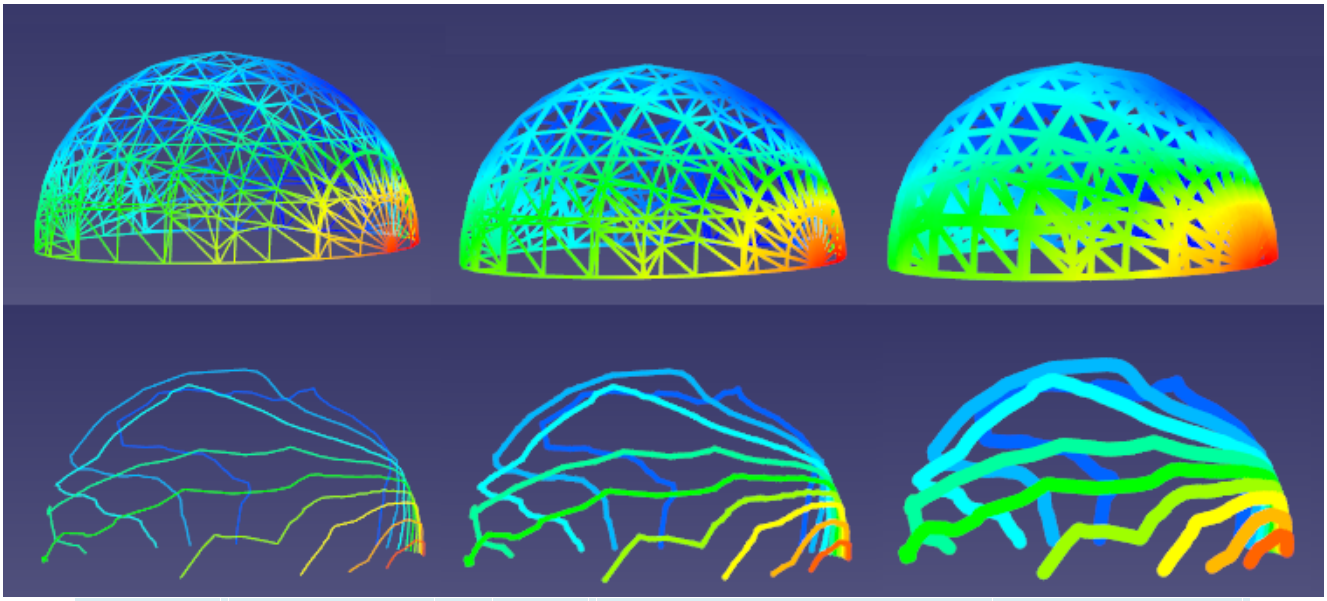
Parameter	Description
Name	Layer name (this option allows you to change the default name "**Color contours #N (Object)**", formed from the name of the layer type, numbers and **Objects** on which the layer is built).
Object	See General properties of Layers.

Parameter	Description
Parts & Accessories> ...	
Restrictions> ...	
Visible	
Clipped	
Lighting	
Update> ...	
Save to file > ...	
Variable> ...	
Subregion	
Shift	**Layer**shift in the direction of or against the normal to the surface on which it is built. The normal direction can be different for different **Objects**. If necessary, change the direction of the shift, change the sign of numerical values of the parameter to the opposite. (See illustration)  When using **Shift** layer based on the data is not an object, but on the surface, the resulting shift of the **Object**'s surface.
On regular grid	Whether to use for the construction of a computational grid layer or auxiliary uniform (regular) grid defined by parameter group **grid** (see below). Possible options are: • **No**- use the computational grid • **Yes**- use a helper (regular) grid ***Note:*** On some **Objects**, this parameter is always **No** and can not be changed. See General properties of Layers.
Grid > ...	See General properties of Layers. These parameters are only available when **On regular grid = Yes**.
Double resolution	Use double the resolution in the construction of the color contours on computational grid*). (These parameters are only available when **On regular grid = No**) • **Yes**- used to construct the fill value of the variable in the cells of the computational grid and the interpolated value of the variable on the edges of the cells • **No**- used to construct the fill only the interpolated value of the variable on the edges of the computational grid cells ***Note:*** *) Double the resolution is more resource intensive, but allows you to more accurately display the results of calculation.
Value range	See General properties of Layers.
Method	The choice of the method of displaying **Color contours**. Possible options are: • **Isolines** - color contour map corresponding to the colors of the **Palette** • **Color fill** - displays a solid color field • **Color wireframe** - painted frame grid display (See illustration)
Method > Width	Thickness of the lines of **Isolines** or the **Color wireframe**. (This parameter is only available when **Method = Isolines** or **Method = Color wireframe**.) You can specify a value from **1**.
Palette > ...	See Parameters for defining a palette.



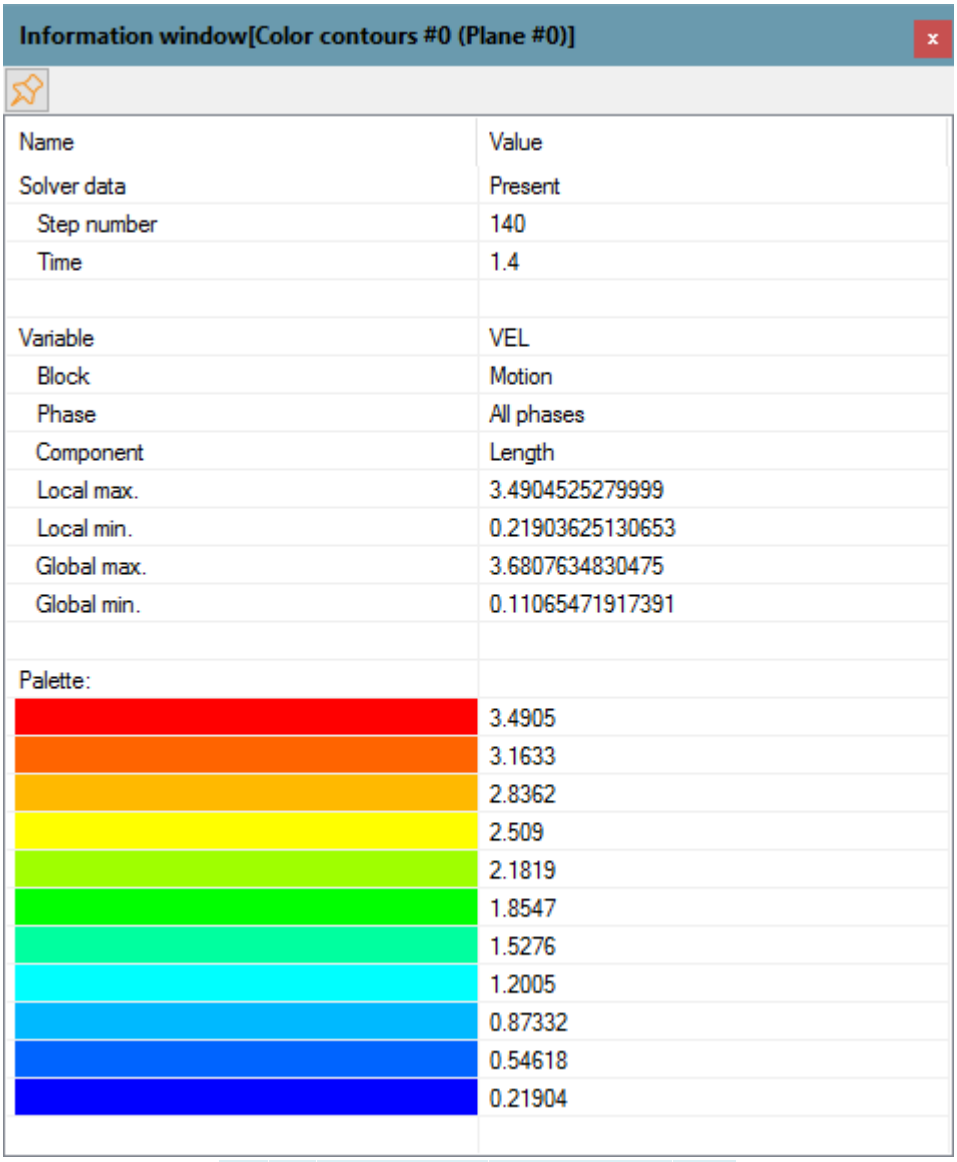
Shift parameter allows to shift the **Layer** relative to the **Object** on which it is built (in the example in the figure there is a shift going on the outside of the sphere)

Layer **Color contours**, built on a hemisphere, for different values of the parameter **Method** (1 -**Color Fill**; 2 - **Isolines**; 3 - **Color wireframe**)



Thickness of **Color wireframe** and **Isolines** can be adjusted by the parameter **Method > Thickness**.

Window «Info» of the «Color contours» layer



The Info window of the **Color contours** layer

Parameter	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current simulated time
Variable	Information about the variable, which is used to build **Color contours**
Palette	Palette, which is used for these **Color contours**

Details about the settings window **Info** for layers see in section Folder "Layers".

Components of the text file for the layer «Color contours»

The file's header describes the format of the data presentation.

At each record to the file, the program writes *n*×*m* sequences of numbers (each sequence is written in a separate line).

Each of these lines represents one node of the color contours.

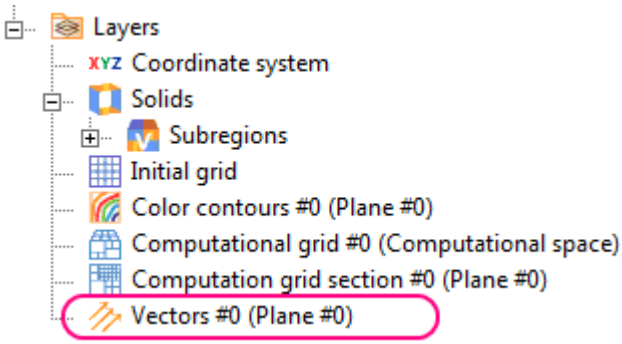
At its beginning, each line contains the following columns:

Step	Step number
Time	Time
Variable	Variable on which to build a layer
NumPoints	The number of points on which construction layer

An i^{th} line contains the following information:

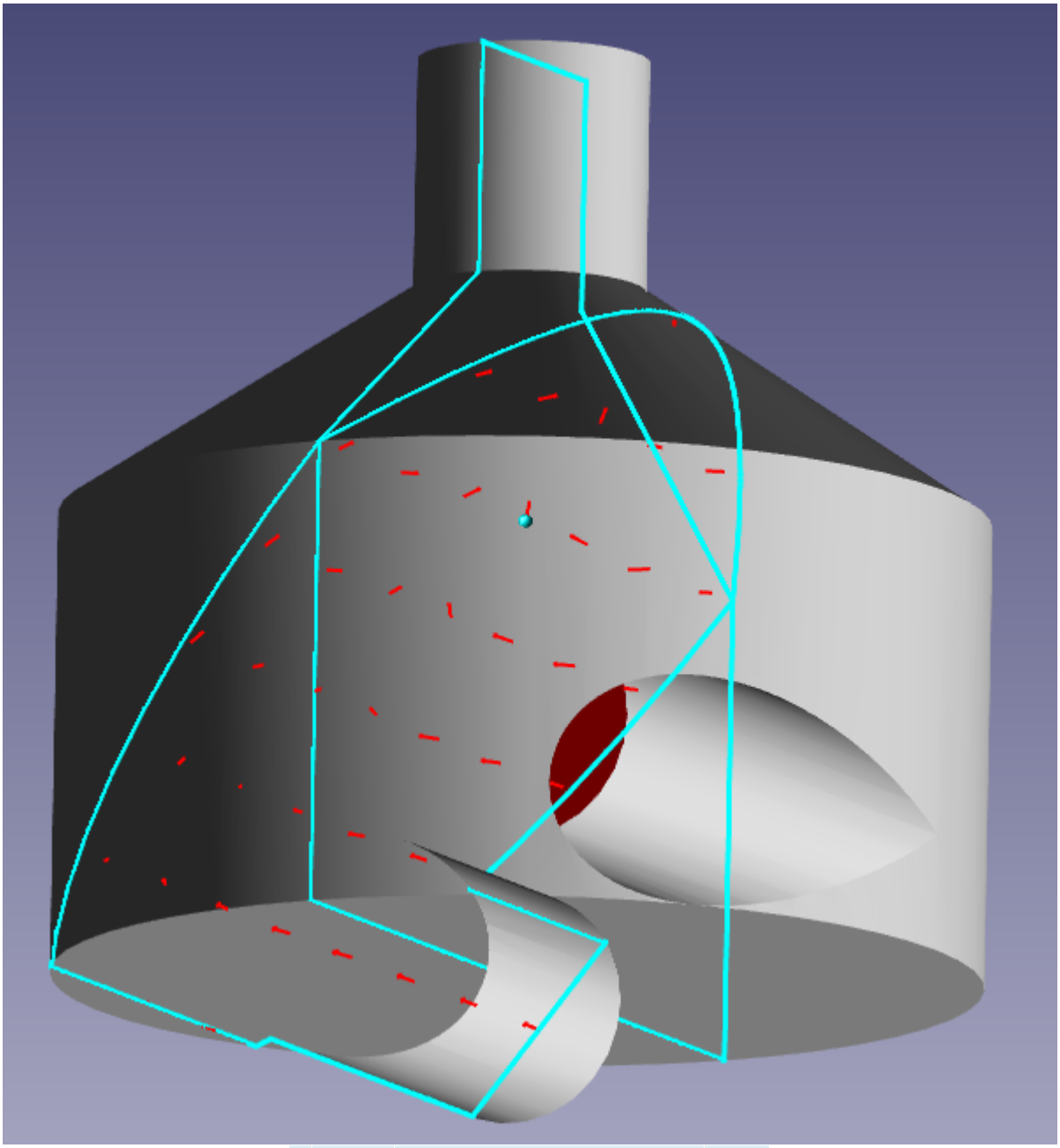
point i.x	X-coordinate of the node
point i.y	Y-coordinate of the node
point i.z	Z-coordinate of the node
value	the variable's value

8.1.8.5.8.8 Layer «Vectors», user interface

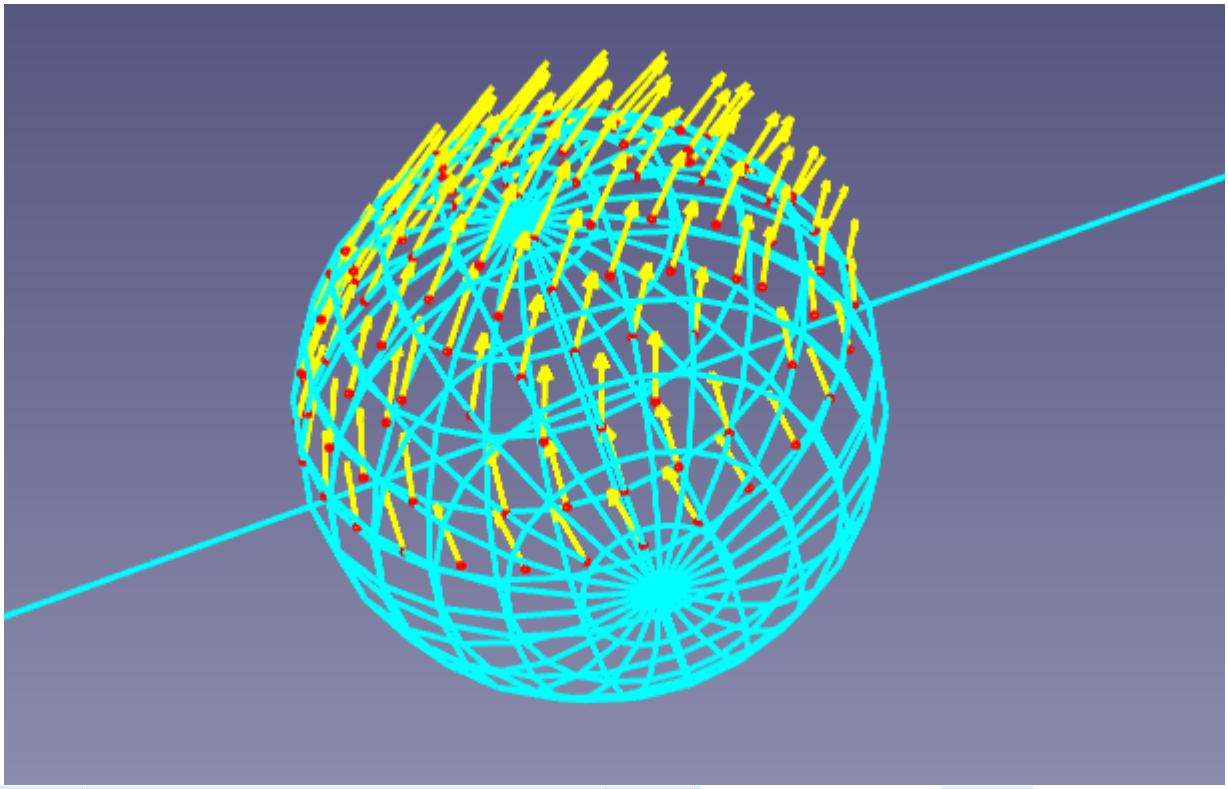


The **Vectors** layer in the project tree

The **Vectors** layer visualize the field of a vector variable in some volume or on a surface.



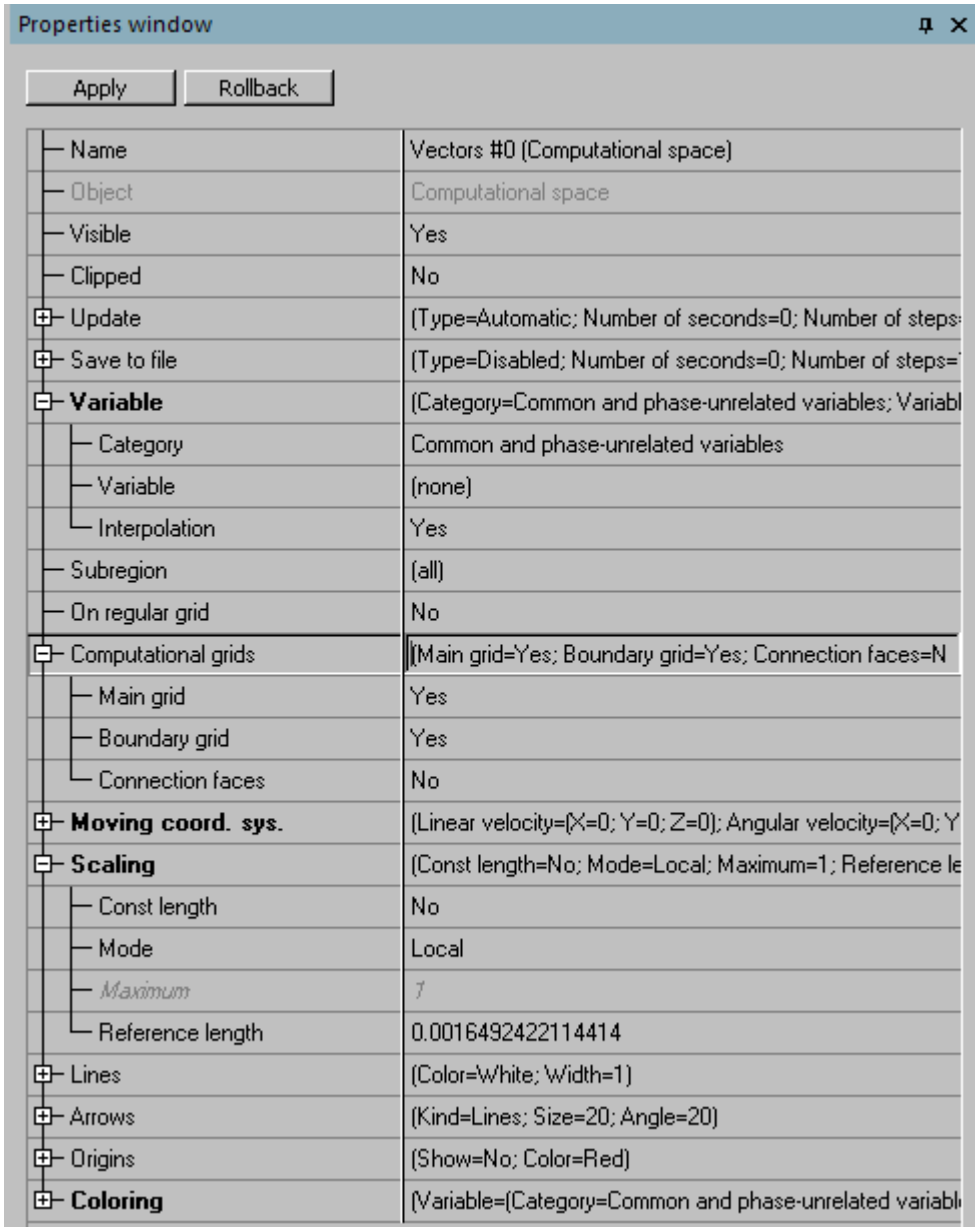
A **Vectors** layer that has been built on a **Plane**



A **Vectors** layer that has been built on the surface of a **Sphere** (the lower half of the **Sphere** locates outside the computational domain)

Vectors can be *colored* according to values of another variable, not the one from the values which they are based (see **Coloring** group of parameters).

Parameters of the «Vector» layer



The **Properties** window of the **Vectors** layer

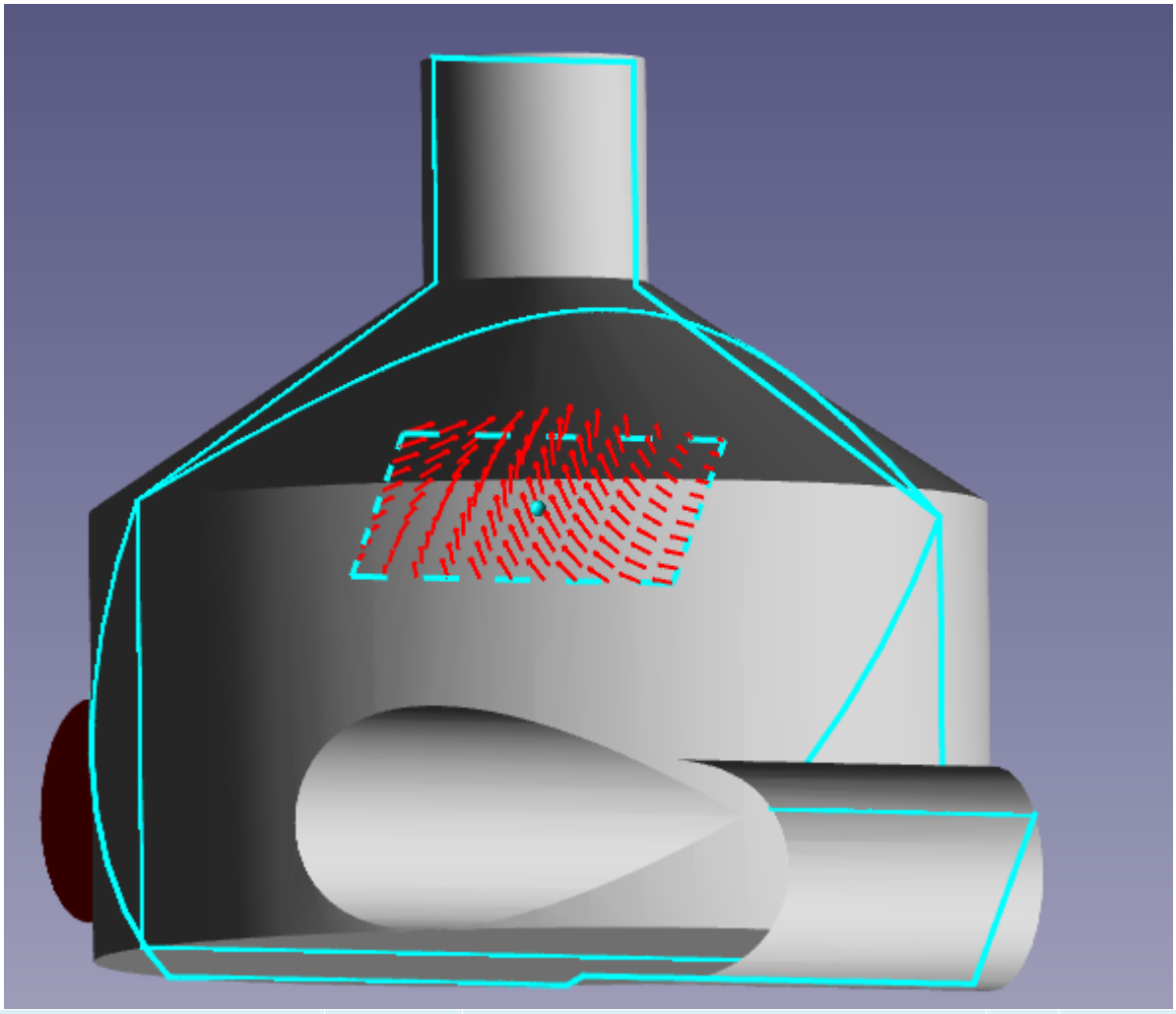
The layer's parameters with the calculated data **Vectors**:

Parameter	Description
Name	Layer name (this option allows you to change the default name **"Vectors #N (Object)"**,formed from the name of the layer type, numbers **and objects** on which a layer).
Object	See General properties of Layers.
Parts > ...	
Constraints > ...	
Visible	
Clipped	
Update > ...	
Save to file > ...	
Variable > ... *)	

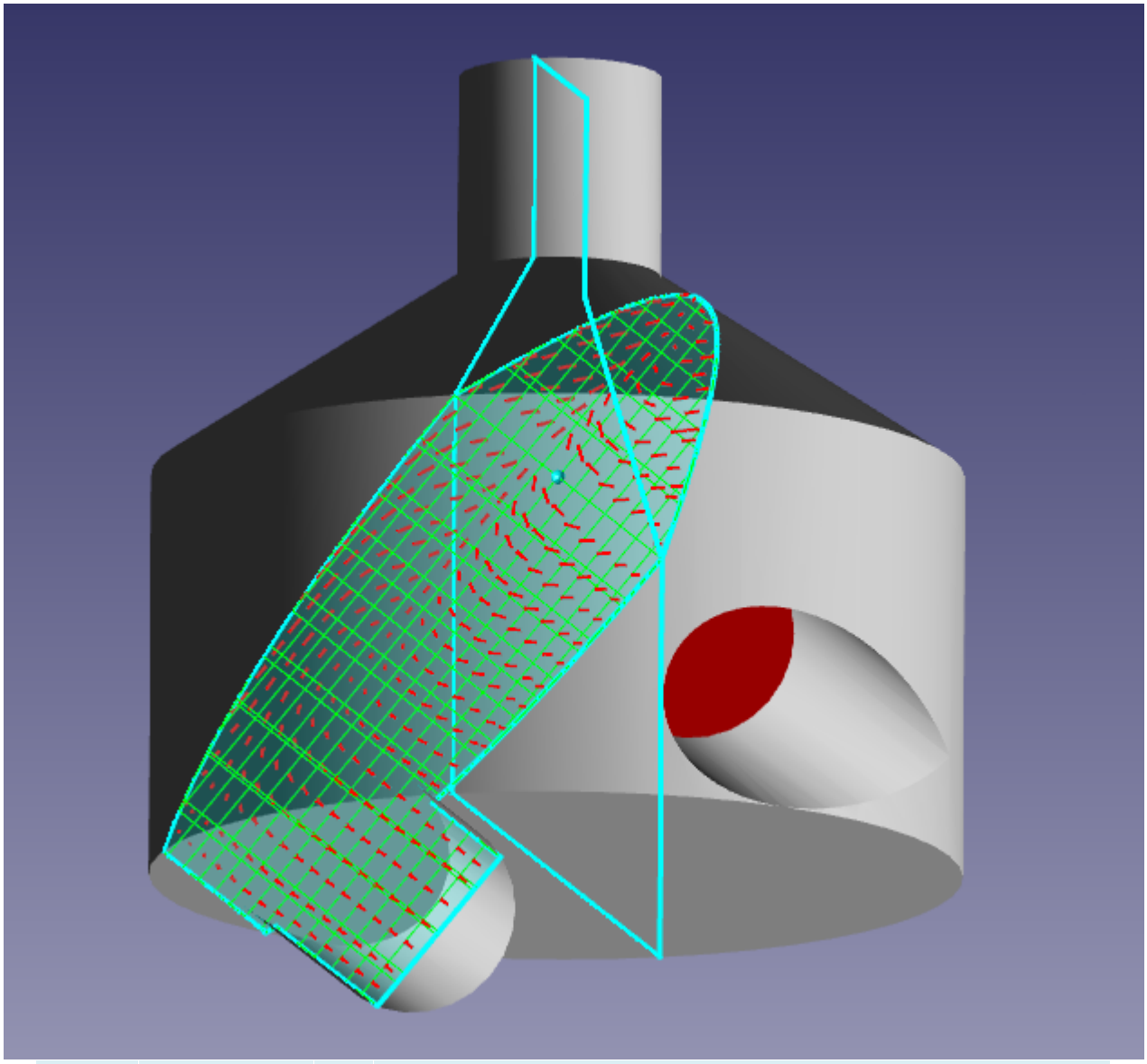
Parameter	Description
Subregion	
Shift	Image shift vectors in the direction of or against the normal to the surface on which they are built (option is available in the construction of a layer on the surface that is defined by the parameters in the group of parts). The normal direction can be different for different **Objects**. If necessary, change the direction of the shift, change the sign of numerical values of the parameter to the opposite. (See figure)
On regular grid	See General properties of Layers.
Computational grids	Selection of computational grids for visualizing data from them. This group of parameters allows you to include or exclude visualization of vectors in the boundary layer grid and in the main grid. This group of parameters is only available when parameter **On regular grid = No**.
Computational grids > Main grid	Visualize data from the main computational grid. Possible options: **Yes** | **No**.
Computational grids > Boundary grid	Visualize data from the boundary layer grid. Possible options: **Yes** | **No**.
Computational grids > Connection faces	Visualize data from faces connecting boundary layer grid with main computational grid. Possible options: **Yes** | **No**.
Grid > ...	See General properties of Layers.
Moving coord. sys. > ...	Displaying vectors in a moving coordinate system*)
Moving coord. sys. >Linear velocity> X	Translational velocity of the moving coordinate system, the components of X, Y, Z.
Moving coord. sys. >Linear velocity> Y	
Moving coord. sys. >Linear velocity> Z	
Moving coord. sys. > The angular velocity>X	The rotational speed of a moving coordinate system components X, Y, Z.
Moving coord. sys. > The angular velocity>Y	
Moving coord. sys. > The angular velocity> Z	
Moving coord. sys. > Center of Rotation>X	The center of rotation of a moving coordinate system, the coordinates X, Y, Z.
Moving coord. sys. > Center of Rotation>Y	
Moving coord. sys. > Center of Rotation> Z	
Scaling	Parameter group, runs the length of the vectors.
Scaling > Const length	You can only show the direction vectors. Possible options are: • **Yes**, the length of the vectors is constant and is calculated by the formula $L = L_{\text{ref}}$ • **No** length of the vectors is proportional to the value of the variable and is calculated by the formula $L = \frac{\lvert V_{\text{cur}} \rvert}{\lvert V_{\max} \rvert} \cdot L_{\text{ref}}$, where:

Parameter	Description
	$\|V_{cur}\|$ is current value of the modulus of the vector quantity $\|V_{max}\|$ is the maximal value of the modulus of the vector quantity, determined in accordance with the **Scaling > Mode** parameter (see below) L_{ref} is the reference length
Scaling > Mode	The choice of method for determining $\|V_{max}\|$. Possible options are: • **Local** - is automatically determined by the values of the variable at the **Object** • **Global** - is automatically determined by the values of the variable in the whole phase • **Manual** - set by parameter **Maximum**
Scaling > Maximum	The value of $\|V_{max}\|$ that is calculated when **Mode = Local** or **Mode = Global** or specified when **Mode = Manual**.
Scaling > Reference length	The value of L_{ref}
Method	The choice of the method of displaying vector variables (available in the construction of vectors on the surfaces): • **Vectors** - display full vectors • **Normal components** - displays only the normal components of the vectors • **Tangential components** - displays only the tangential components of the vectors (See illustration)
Lines > Color **Line > Width**	Line color and width image vectors.
Arrows	Parameters for setting the image of arrows at the ends of the image vectors
Arrows > Kind	Select the type of pointers display: • **No arrows** - the arrows are not displayed • **Lines** - the arrows are displayed as lines • **Triangles** - the arrows are displayed as triangles • **Pyramid** - the arrow is displayed in the form of pyramids (see an illustration below)
Arrows > Size	Size of the arrows (set as a percentage of the length of the vector)
Arrows > Angle	The angle between the lines and the direction of the vector arrows (given in degrees)
Origins > Show	Displaying starting points of vectors. Possible options: • **No** • **Yes**
Origins > Color	Color dots that indicate the beginning of vectors.
Coloring > Variable> ... **Coloring > Value range > ...** **Coloring > Log. scale > ...** **Coloring > Palette > ...**	Coloring the images of vectors in colors, depending on the value of a variable at a given point of the **Object**. Variable range and palette settings are specified similarly to the standard parameter groups **Value range** and **Palette**. See General properties of Layers and Parameters for defining a palette.

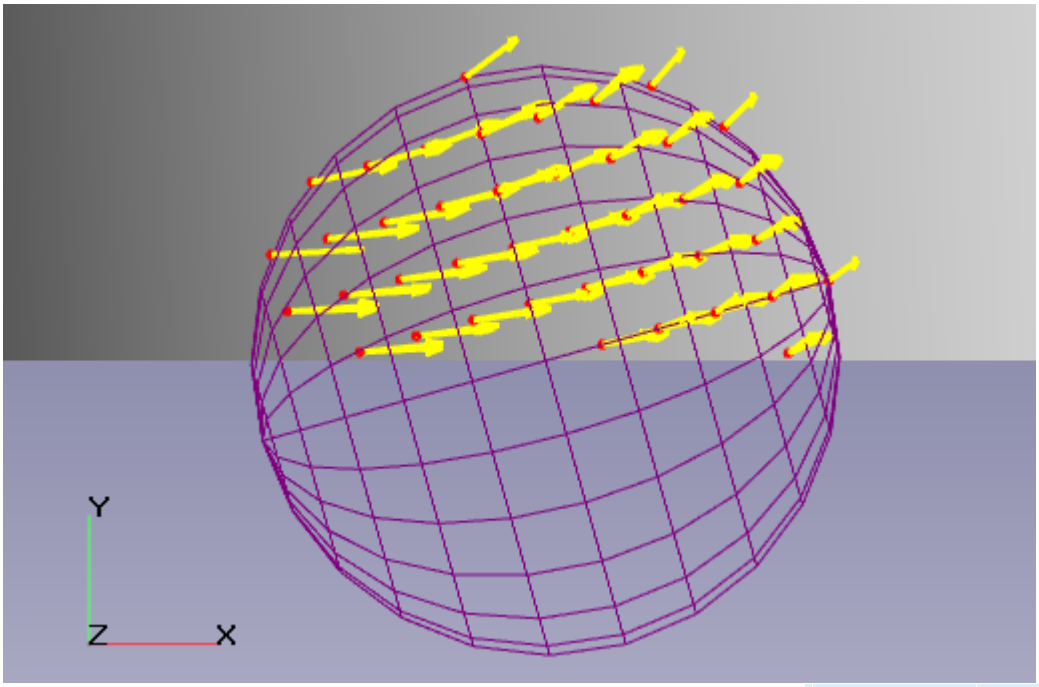
*) If the **Variable**, which is used to vectors, and the **Variable**, which is used for coloring, are the same, then the coloring visualize values of the variable in the relative coordinate system. If motion is calculated in the computational model, then **Variable=Velocity** is specified by default.



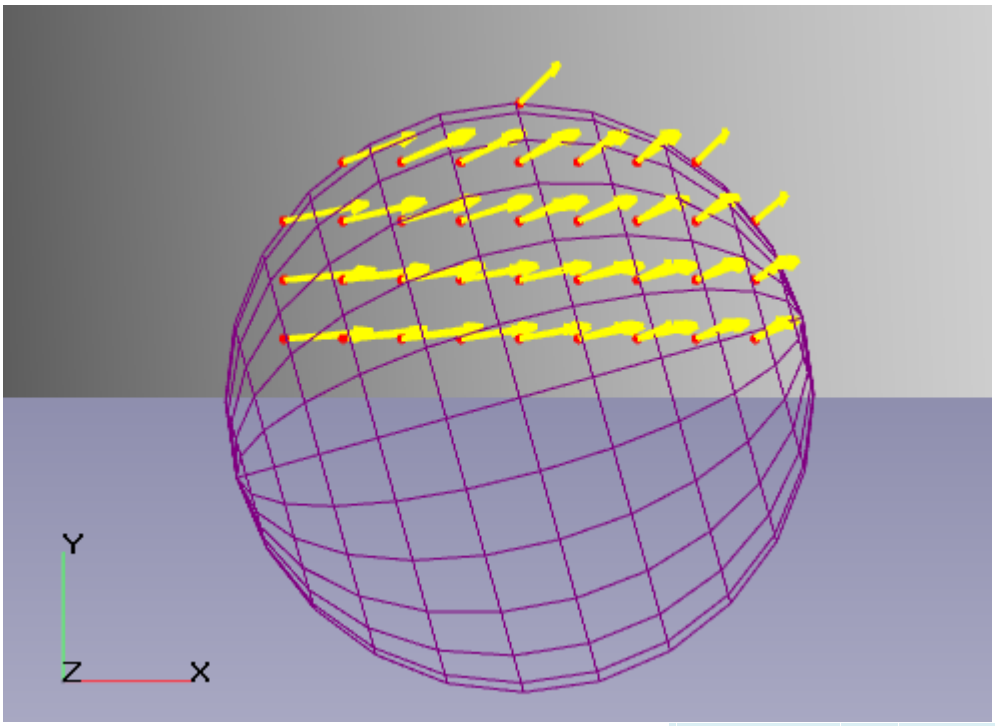
If you specify parameters in the **Constraints** group, you can build a layer not on the entire **Plane**, but only in a rectangle around the **Plane**'s **Reference point**.
(parameters are set: **Enabled = Yes**, **Type = Rectangle**, **Size 1 = 0.004**, **Size 2 = 0.008**)



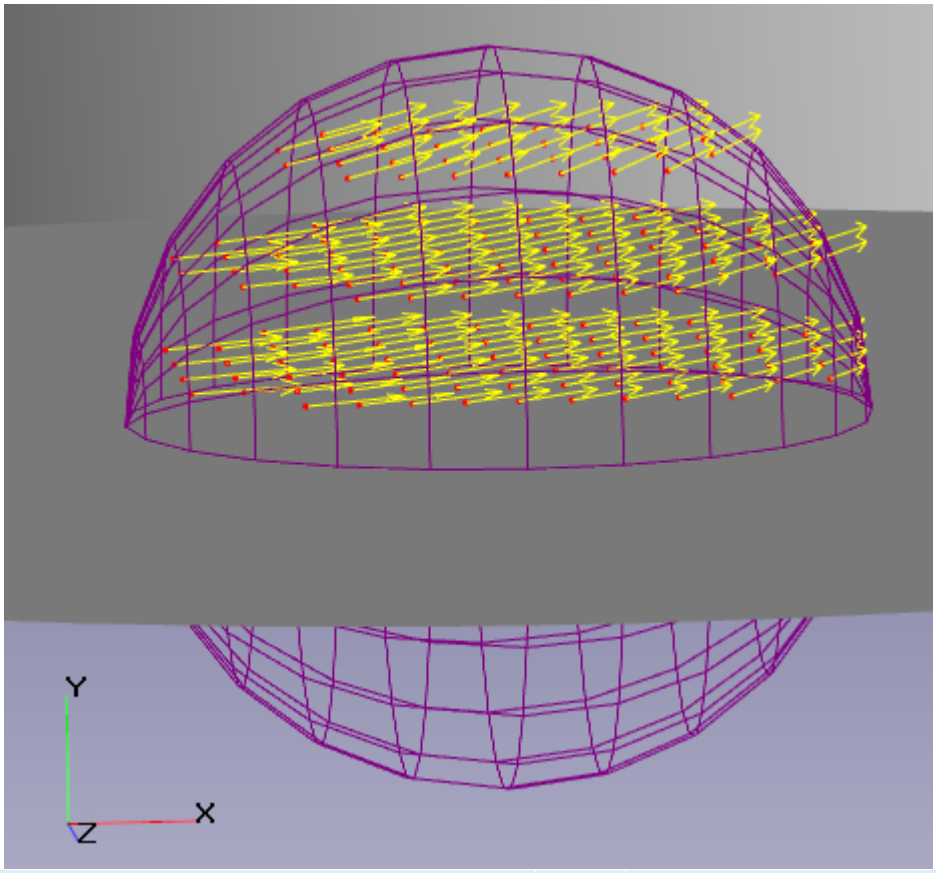
If you set **On regular grid = No**, then the vectors will be formed in the nodes of the computational grid.
(The picture shows two **Layers**: **Computational grid section** and **Vectors**, which are built on the same **Plane**)



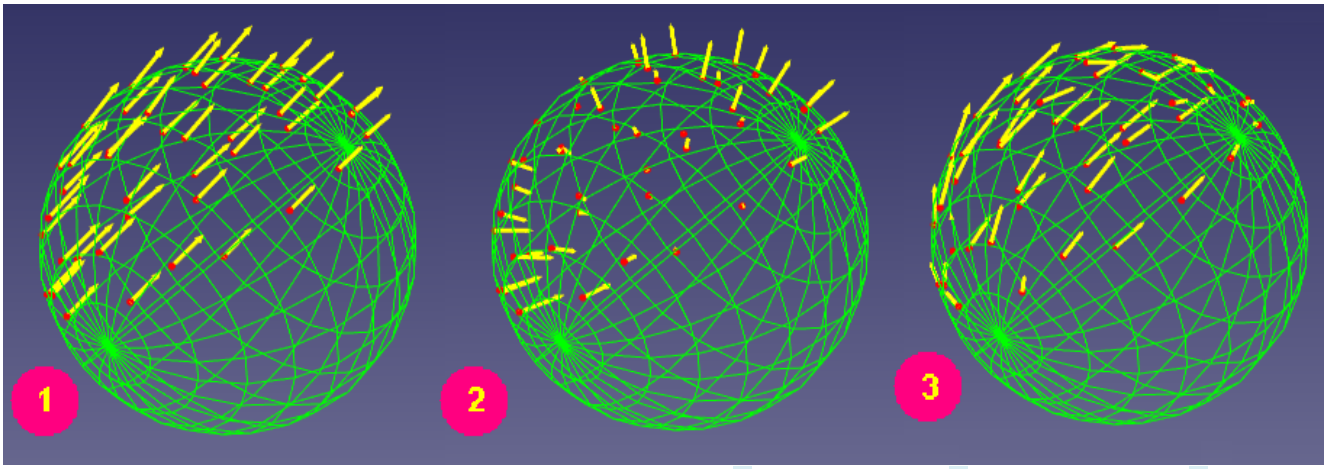
Regular grid is built along the axes of the local coordinate system of the **Object** (**Grid > Local = Yes** has been specified)



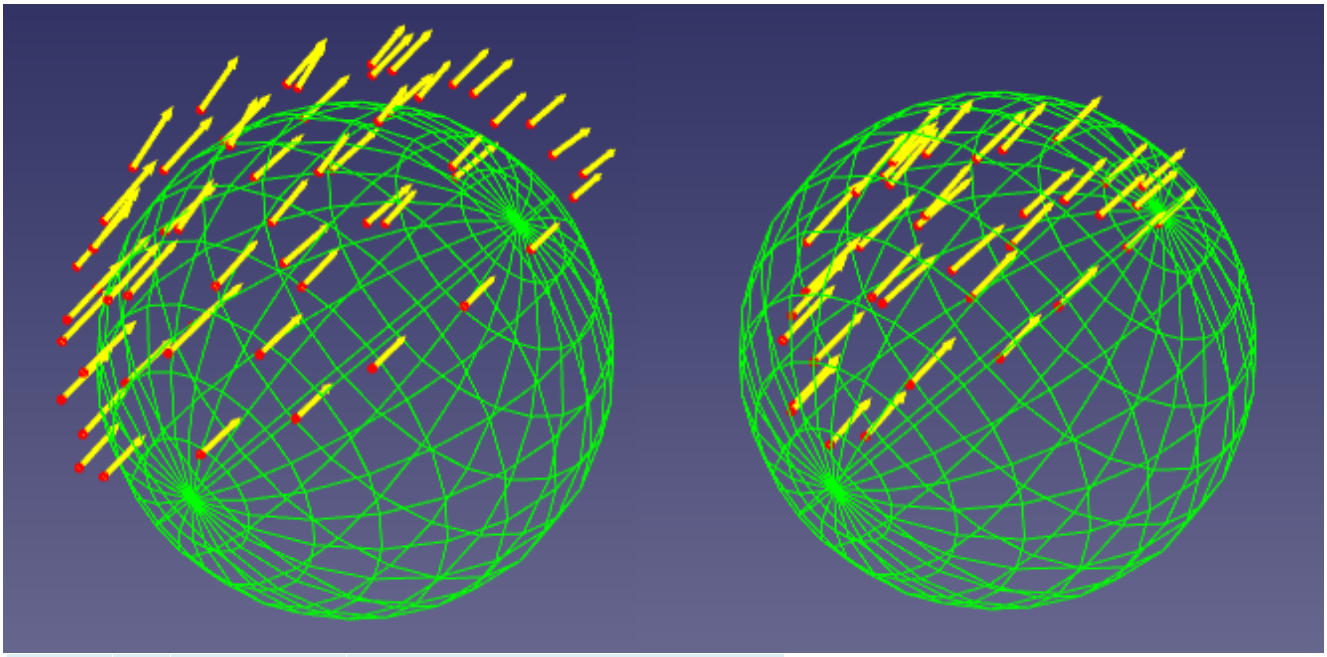
Regular grid is built along the axes of the absolute coordinate system (**Grid > Local = No** has been specified)



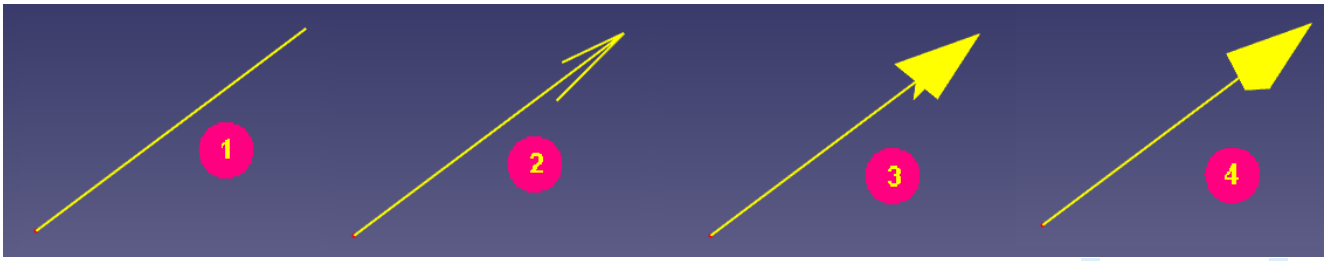
When using a regular grid, the number of its nodes within the **Object** is specified by parameters **Grid > Size 1**, **Grid > Size 2** and **Grid > Size 3**.



With the **Method** parameter the vectors can be displayed as (1) whole vectors; (2) or their normals; (3) or their tangential components

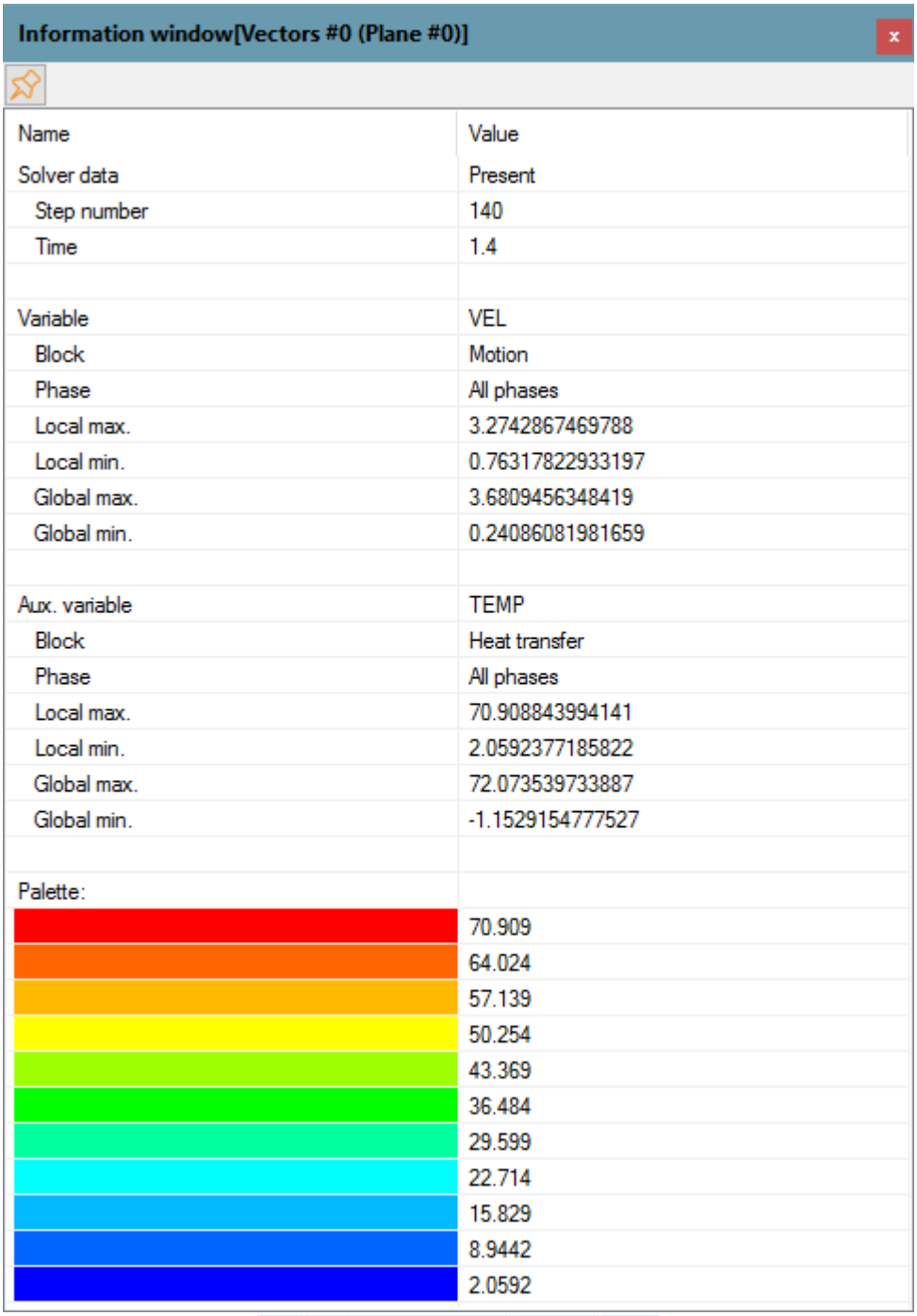


With the **Shift** parameter the vectors can be shifted in the direction of normal to the surface on which the **Layer** is built (or against the normal)



The appearance of the arrows in the images set with the parameter vectors **Arrows > Kind** (1 - **No arrows**; 2 - **Lines**; 3 - **Triangles**; 4 - **Pyramid**)

Window «Info» of the «Vectors» layer



The **Info** window of the **Vectors** layer

Parameter	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current simulated time
Variable	Information about the variable which is used to build the vectors
Aux. variable	Information about the variable, which is used for *coloring* the vectors
Palette	Palette, used for coloring vectors (from the values of the additional variable)

Details about parameters in the **Info** window for layers see in the section Folder "Layers".

Components of the text file for the layer "Vector"

The file's header describes the format of the data presentation.

At each record to the file, the program writes $n \times m$ sequences of numbers (each sequence is written in a separate line).

Each of these lines represents one vector.

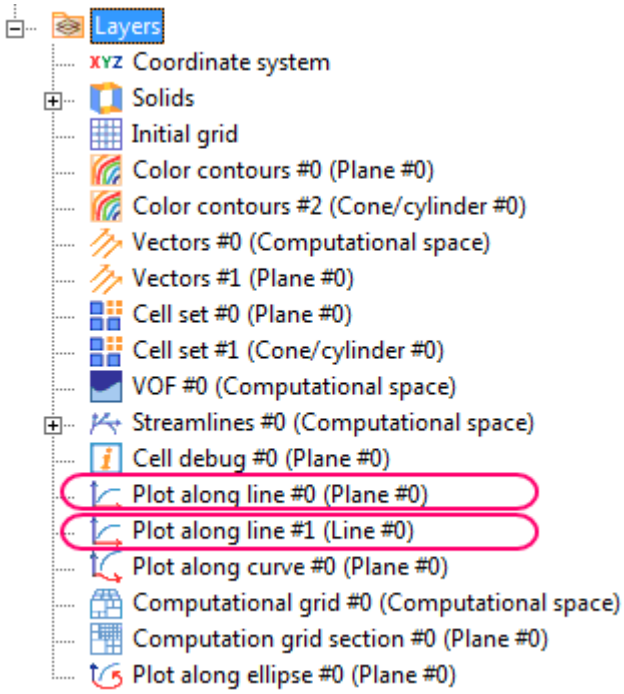
At its beginning, each line contains the following columns:

Step	Step number
Time	Time
Variable	Variable on which to build a layer
NumPoints	The number of points on which construction layer

An i^{th} line contains the following information:

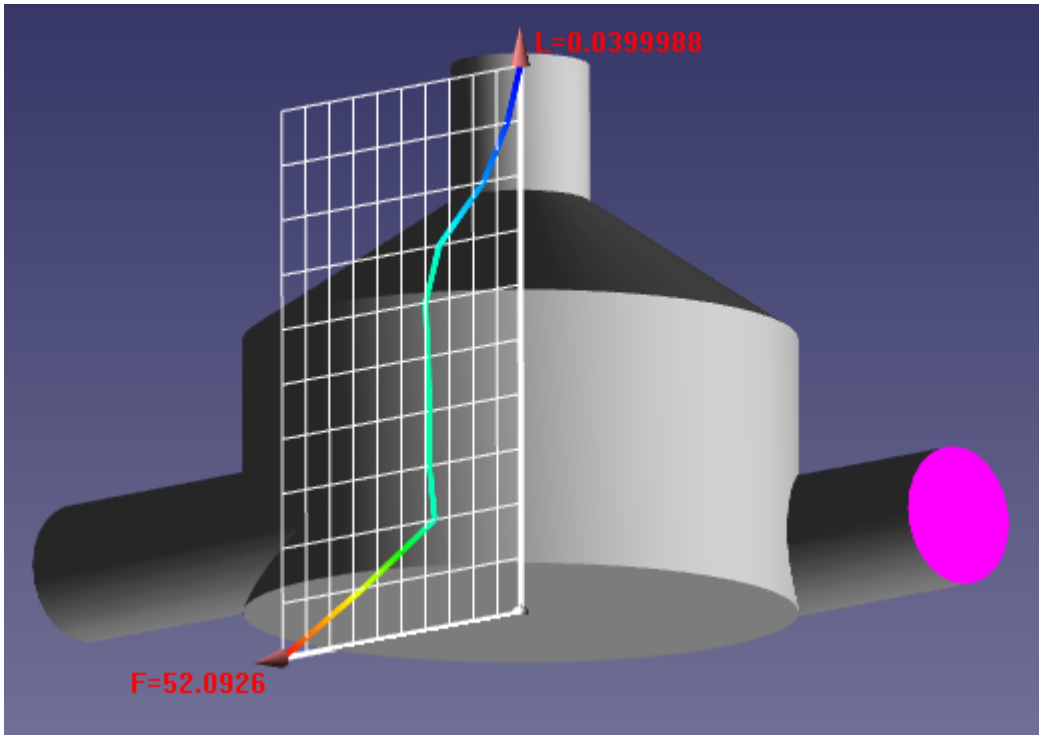
point i.x	X-coordinate of the starting point of the vector
point i.y	Y-coordinate of the starting point of the vector
point i.z	Z-coordinate of the starting point of the vector
value i.x	X-component of the vector
value i.y	Y-component of the vector
value i.z	Z-component of the vector
value i.length	Length of the vector
normal i.x	Direction of the normal to the surface at the point x, y, z
normal i.y	
normal i.z	
cover i	Value of the variable, which is used to color the vector

8.1.8.5.8.9 Layer «Plot along line», user interface



The **Plot along line** layer in the project tree

Layer **Plot along line** is designed to display a scalar variable, depending on the points along the line (for a vector variable its absolute value is used).



The **Plot along line** layer can be built on the following **Objects**:

- **Plane**
- **Line**

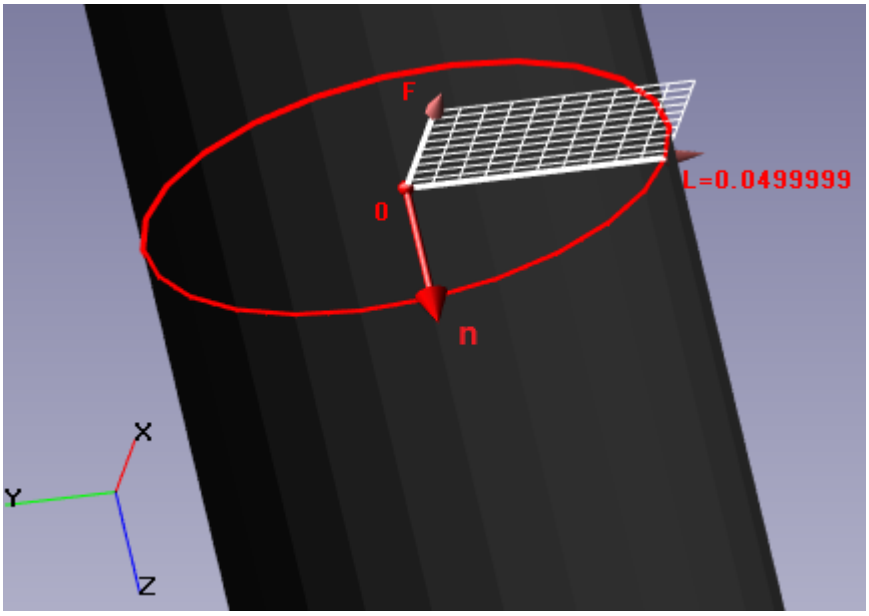
The origin of the ray, on which the **Plot** is built, locates in the center of the base object (**Plane** or **Line**).

Axes of the **Plot along line** are defined as follows:

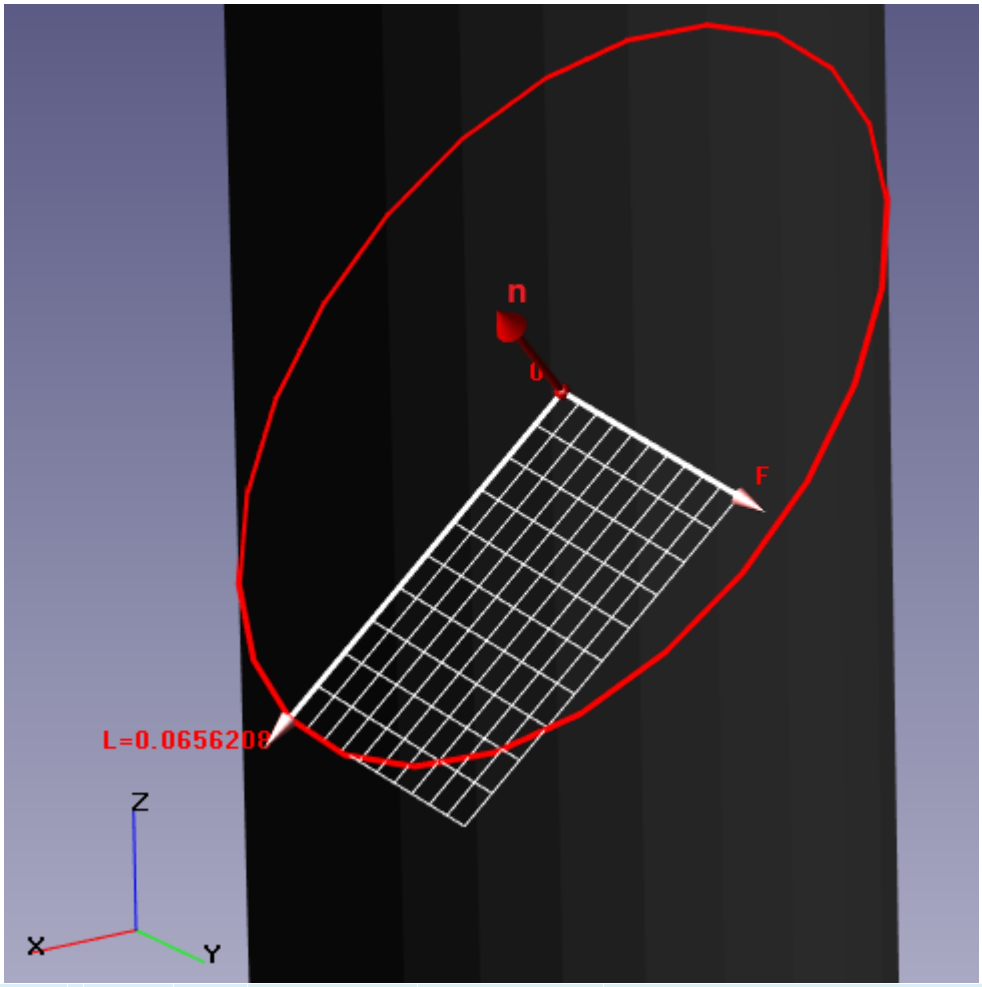
Axes of the Plot	Base object is a Plane*)	Base object is a Line
Axis of abscissas	When the normal vector **n** to the **Plane** is *collinear* to one of axes of the absolute	This axis goes along the **Line**.

Axes of the Plot	Base object is a Plane*)	Base object is a Line
It is marked on the plot as **L**.	coordinate system (ACS): • if **n** is directed along the axis **X**, then **L** (the axis of abscissas of the **Plot along line**) is directed against the axis **Z**. • if **n** is directed against the axis **X**, then **L** is directed along the axis **Z**. • if **n** is directed along the axis **Y**, then **L** is directed along the axis **Z**. • if **n** is directed against the axis **Y**, then **L** is directed against the axis **Z**. • if **n** is directed along the axis **Z**, then **L** is directed against the axis **Y**. • if **n** is directed against the axis **Z**, then **L** is directed along the axis **Y**. When the normal vector **n** to the **Plane** is *not collinear* to any axis of the ACS, then the axis of abscissas of the **Plot along line** (**L**) is directed along the vector product [**t**×**n**], where **t** is the unit vector of those ACS axis that has the minimal absolute value of the vector **n**'s projection on it.	
Axis of ordinates It is marked on the plot as **F**.	The axis of ordinates (**F**) of the **Plot along line** is directed along the vector product [**n**×$\mathbf{L}_e$], where $\mathbf{L}_e$ is the unit vector along the the axis of abscissas (**L**) of the **Plot along line**.	This axis is orthogonal to the **Line**.

*) **X**, **Y**, **Z** are axes of the absolute coordinate system (ACS). See also illustrations below.



Example 1: Normal (**n**) to the **Plane**, on which the **Plot along line** is built, is collinear to the axis **Z**.
The **Plot**'s axis of abscissas (**L**) is directed against the axis **Y**.
The **Plot**'s axis of ordinates (**F**) is directed along the axis **X**, and also along the vector product [**n**×$\mathbf{L}_e$], where $\mathbf{L}_e$ is the unit vector along **L**.



Example 2: Normal (**n**) to the **Plane**, on which the **Plot along line** is built, is not collinear to any axis of the ACS. The absolute value of the vector **n**'s projection on a coordinate axis is minimal on the axis **Y** so the **Plot**'s axis of abscissas (**L**)
is directed along the vector product [$\mathbf{Y}_e \times \mathbf{n}$], where $\mathbf{Y}_e$ is the unit vector along **Y**.
The **Plot**'s axis of ordinates (**F**) is directed along the vector product [$\mathbf{n} \times \mathbf{L}_e$], where $\mathbf{L}_e$ is the unit vector along **L**.

A line that is displayed on the **Plot** can be colored according to the value of another variable (this is specified in the **Coloring > ...** group of parameters).

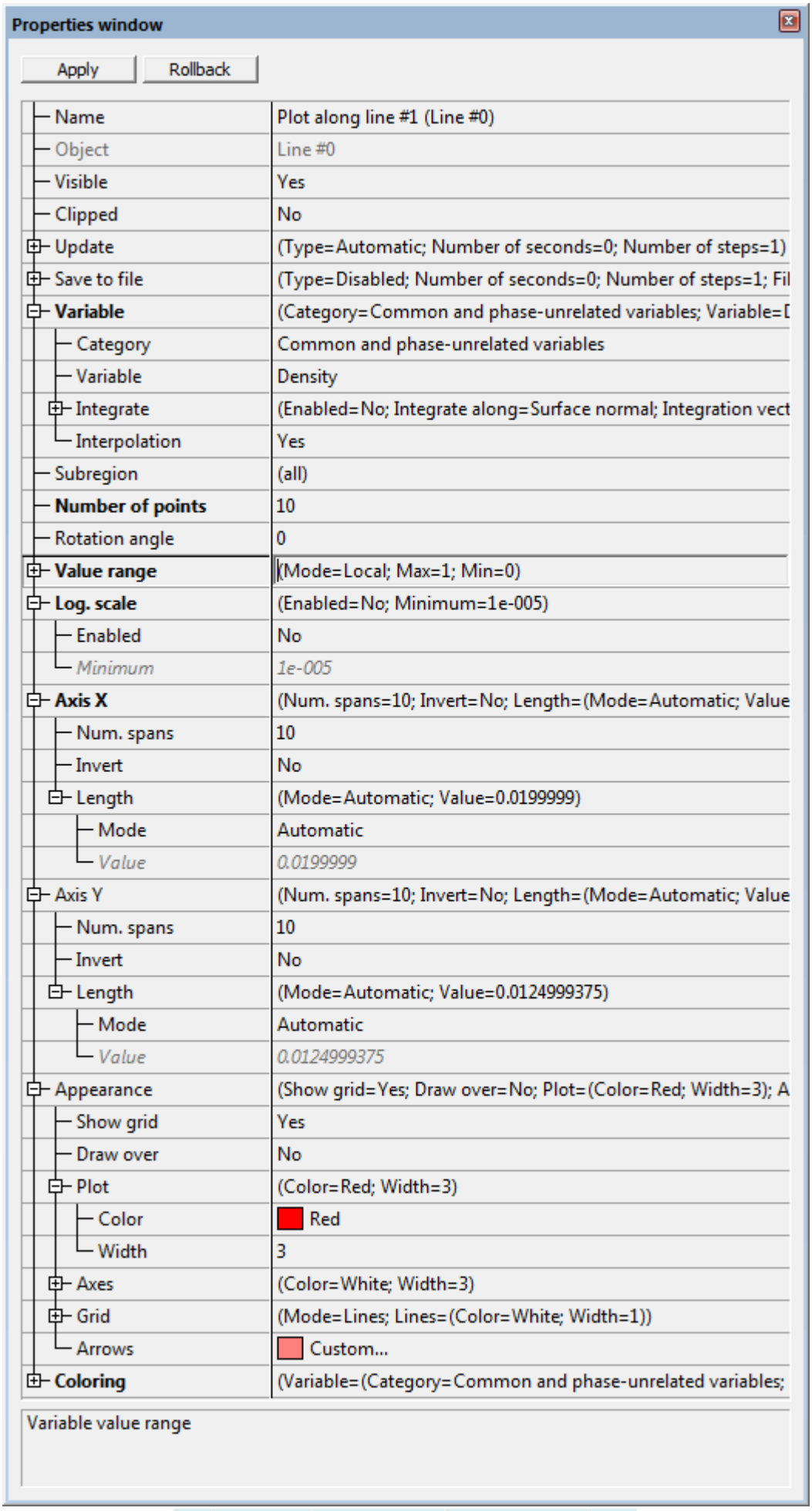
The **Plot** is displayed as an unlit polygonal line with a fixed number of segments, located in the first quarter of the coordinate axes of the **Plot**. At positive ends of the axes small lighted cones locate.

In the plane of the **Plot** displays the following comments:

- The length of the x-axis at the positive end of the x-axis.
- The maximum value of a variable near the positive end of the axis of ordinates.
- The minimum value of the variable near the negative end of the axis of ordinates.

Furthermore, depending on the properties set in the plane coordinate plot can be drawn lines in parallel with each of the axes.

Parameters of the «Plot along line» layer

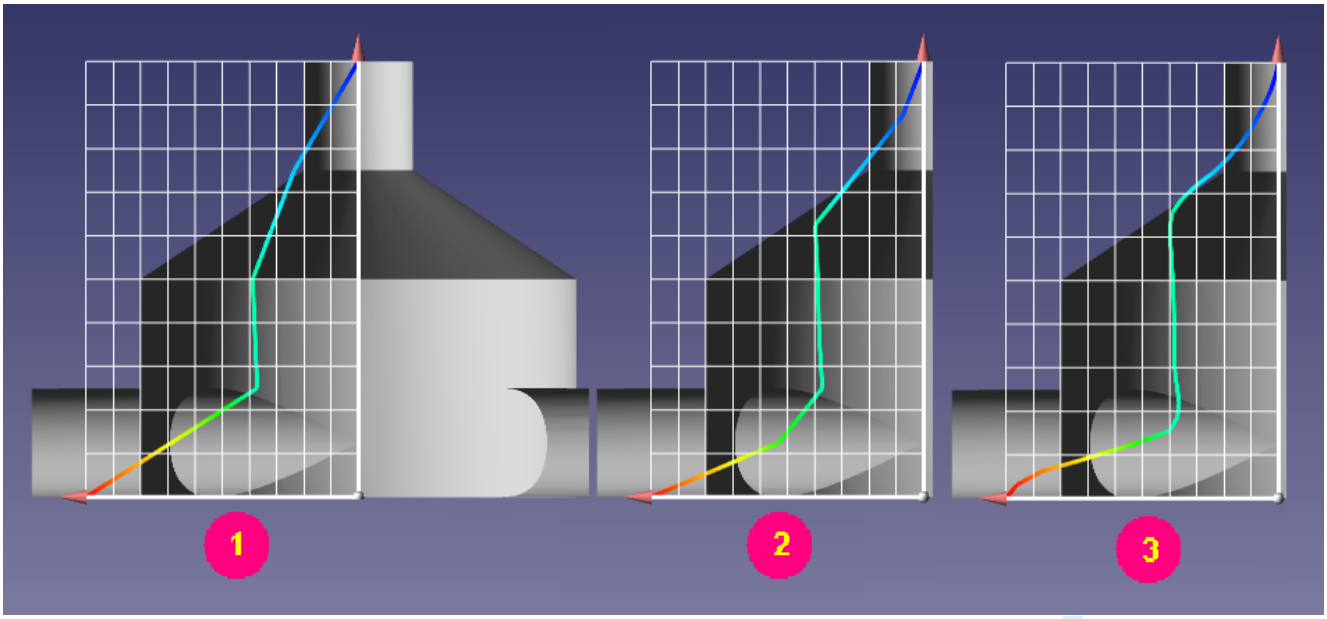


The **Properties** window of the **Plot along line** layer

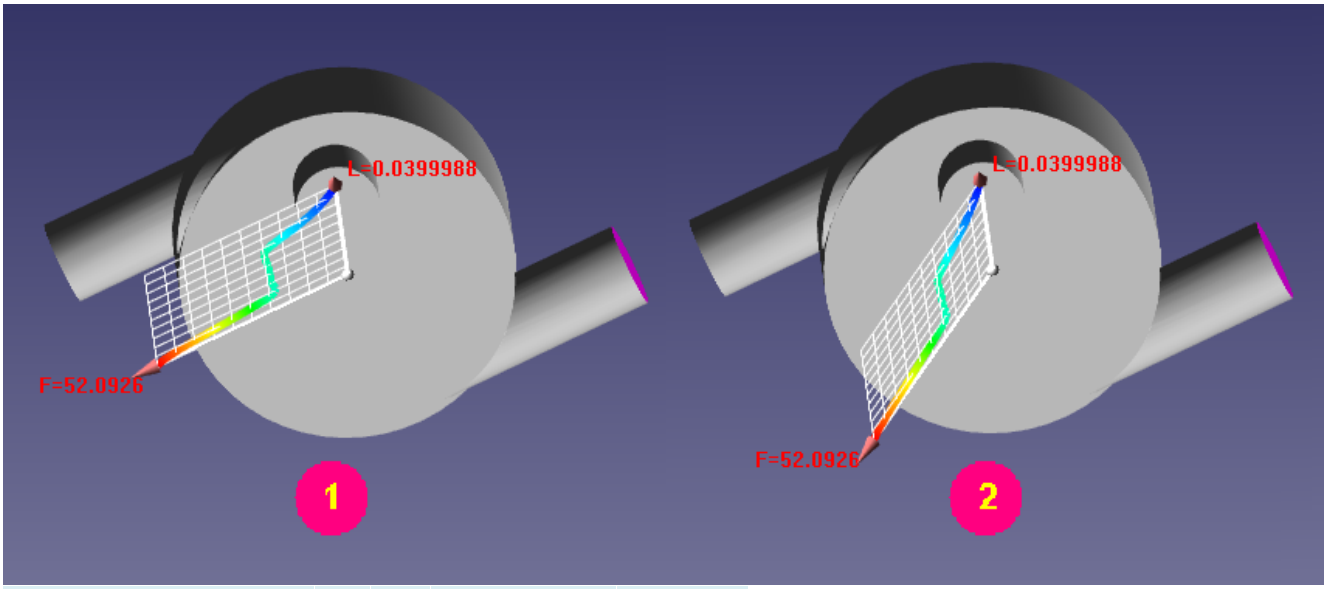
Parameters of the layer **Plot along line**:

Parameter	Description	The dimension of
Name	Layer name (this option allows you to change the default title **"Plotalong line#N** (Object)",formed from the name of the layer type, numbers and **objects** on which the layer is built).	
Object	See General properties of Layers.	
Visible		
Clipped		
Update > ...		
Save to file > ...		
Variable> ...	Variable that determines the trajectory of the plot line. See General properties of Layers.	
Subregion	Limitation layer of a given sub-area. See General properties of Layers.	
Number of points	The number of points on the plot the abscissa. (See illustration)	
Rotation angle	Angle of rotation of the line along which the plot of (x-axis plot): • if the plot is built on the **lines**, the rotation is around the **line** • if the plot is built on the **plane**, the rotation is around the normal to the **plane** passing through the origin of the plot (See illustration)	[Degree]
Value range > ...	See General properties of Layers.	
Log scale > ...		
Axis X > ...		
Axis Y > ...		
Appearance > Show grid	Displaying a grid of the plot	
Appearance > Draw over	Drawing graphics on top of the other elements of the **View** window	
Appearance > Plot > ...	Group of parameters that control the display of the plot line and axis labels graphics	
Appearance > Plot> Color	Line color graphics and axis labels **Plot**. (The line plot can be colored according to the values of a variable, then it this parameter does not apply)	
Appearance > Plot > Width	The thickness of the plot line	
Appearance > Axes > Color	The color and thickness of the chart axes	
Appearance > Axes > Width		
Appearance > Grid > Mode	See General properties of Layers for parameters **Appearance > Grid> ...**	
Appearance > Grid > Lines > Color		
Appearance > Grid > Lines > Width		
Appearance > Grid > Fill > Color		

Parameter	Description	The dimension of
Appearance > Grid > Fill > Opaqueness		
Appearance > Arrows	Color arrows at the ends of the plot axes	
Coloring > Variable> ...	Staining line graphics in color, depending on the values of a variable on the x-axis of the plot (this variable can be the same or not the same as the variable that defines the path of the line graphics).	
Coloring > Value range> ...		
Coloring > Log. scale > ...		
Coloring > Palette > ...	Variable range and palette settings are specified similarly to the standard parameter groups **Variable Range** and **Palette**, see General properties of Layers.	

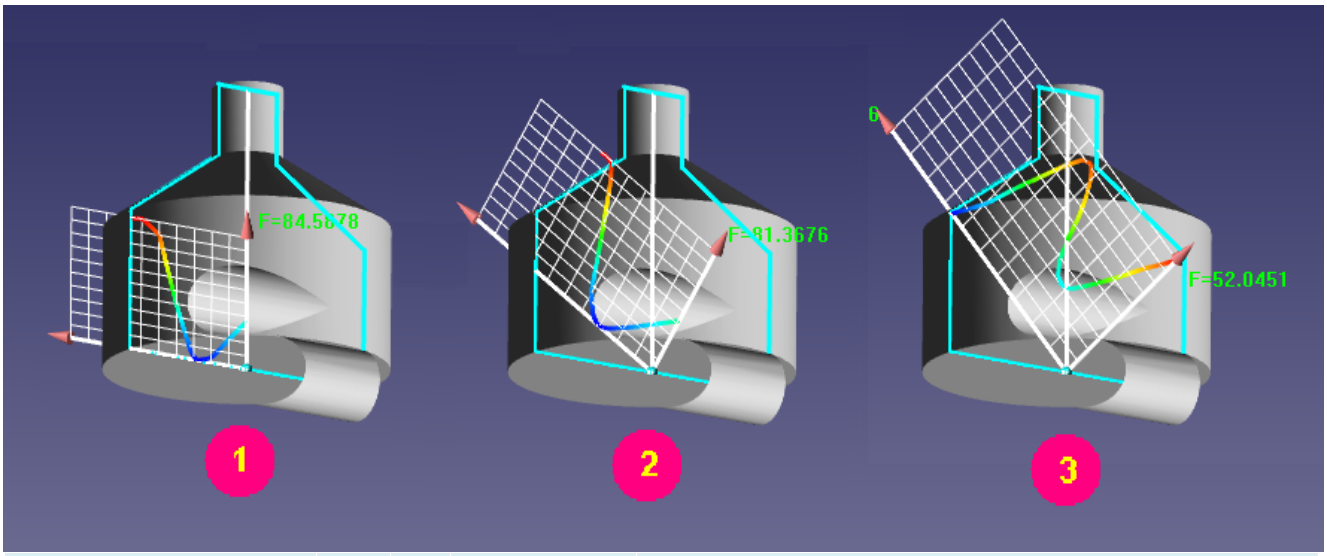


The **Number of points** parameter affects on how many breaks will be in the line of the plot (1 - **Number of points = 5**, 2 - **Number of points = 9**, 3 - **Number of points = 33**)



When the plot is built on a **Line**, the **Rotation angle** parameter (specified in degrees) allows you to rotate the plot along the x-axis

(1 - **Rotation angle = 0**, 2 - **Rotation angle = 30**)



When the plot is built on a **Plane**, the **Rotation angle** parameter allows you to rotate the plot along the normal to the **Plane** (this normal passes through the origin graphics)

(1 - **Rotation angle = 0**, 2 - **Rotation angle = 30**, 3 - **Rotation angle = 45**)

Window «Info» of the «Plot along line» layer

The data displayed in the Info window of the **Plot along line** layer:

Parameter	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current simulated time
Significant points	The number of points on which the plotted

Variable > ...	Information about the variable, which is used to build the the plot
Average value	The average value of the variable, which is used to build the the plot
Maximum	The maximal value of the variable, which is used to build the the plot
Maximum > Maximum position	Coordinates of the maximal point of the variable, which is used to build the the plot, along the abscissa axis of the plot
Maximum > Maximum point > ...	Coordinates of the maximal point of the variable, which is used to build the the plot, along axes X, Y, Z
Minimum	The minimal value of the variable, which is used to build the the plot
Minimum > Minimum position	Coordinates of the minimal point of the variable, which is used to build the the plot, along the abscissa axis of the plot
Minimum > Minimum point > ...	Coordinates of the minimal point of the variable, which is used to build the the plot, along axes X, Y, Z
Aux. variable > ...	Information about the variable, which is used to *color* the plot
Average value	The average value of the variable, which is used to *color* the plot
Maximum	The maximal value of the variable, which is used to *color* the plot
Maximum > Maximum position	Coordinates of the maximal point of the variable, which is used to *color* the plot, along the abscissa axis of the plot
Maximum > Maximum point > ...	Coordinates of the maximal point of the variable, which is used to *color* the plot, along axes X, Y, Z
Minimum	The minimal value of the variable, which is used to *color* the plot
Minimum > Minimum position	Coordinates of the minimal point of the variable, which is used to *color* the plot, along the abscissa axis of the plot
Minimum > Minimum point > ...	Coordinates of the minimal point of the variable, which is used to *color* the plot, along axes X, Y, Z
Palette	The palette, which is used for coloring the plot line (depending on values of the auxiliary variable)

Details about parameters of the **Info** window for **Layers** see in section Folder "Layers".

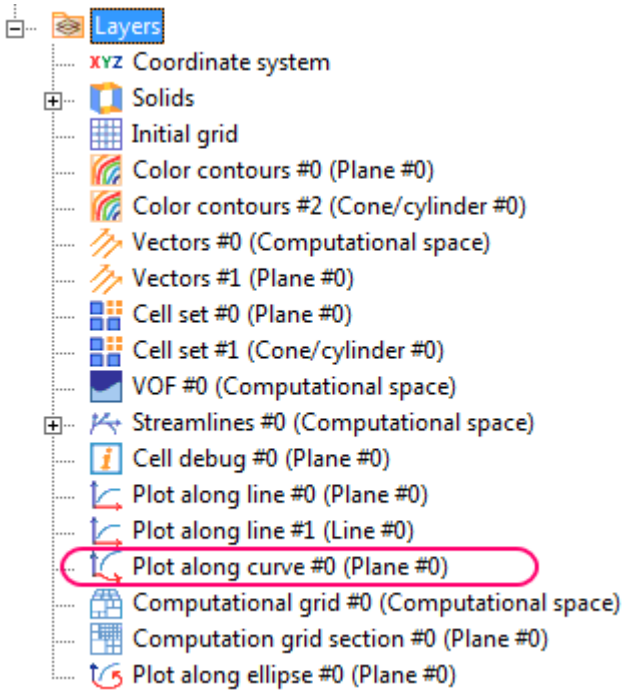
Components of the text file for the layer «Plot along line»

With each entry in the file shows the following information:

Step	Step number
Time	Time
Variable	Variable that determines the trajectory of the plot line
CoverVariable	Variable over which the coloring of the plot line
NumPoints	The number of points on which construction layer
Length	The length of the Plot
Avg	The average value of the variable on the plot
Min	The minimum value of the variable on the plot
MinArg	Coordinate of the lowest point on the x-axis
MinPt.x	X-coordinate of the minimum
MinPt.y	Y-coordinate of the minimum
MinPt.z	Z-coordinate of the minimum
Max	The maximum value of a variable plot

MaxArg	The coordinate on the abscissa of the maximum
MaxPt.x	X-coordinate of the maximum
MaxPt.y	Y-coordinate of the maximum
MaxPt.z	Z-coordinate of the maximum
CoverAvg	The average value of the variable in the plot coloring
CoverMin	The minimum value of the variable on the plot coloring
CoverMinArg	Coordinate of the minimum variable coloring on the x-axis
CoverMinPt.x	X-coordinate of the minimum variable coloring
CoverMinPt.y	Y-coordinate of the minimum variable coloring
CoverMinPt.z	Z-coordinate of the minimum variable coloring
CoverMax	The maximum value of the variable on the plot coloring
CoverMaxArg	Coordinate of the maximum variable coloringonthe x-axis
CoverMaxPt.x	X-coordinate of the maximum variable coloring
CoverMaxPt.y	Y-coordinate of the maximum variable coloring
CoverMaxPt.z	Z-coordinate of the maximum variable coloring
Number	Point Number
Arg	The coordinate of a point on the axis of abscissas
Value	Value of the variable at
x	X-coordinate of the point
y	Y-coordinate of the point
z	Z-coordinate of the point

8.1.8.5.8.10 Layer «Plot along curve», user interface

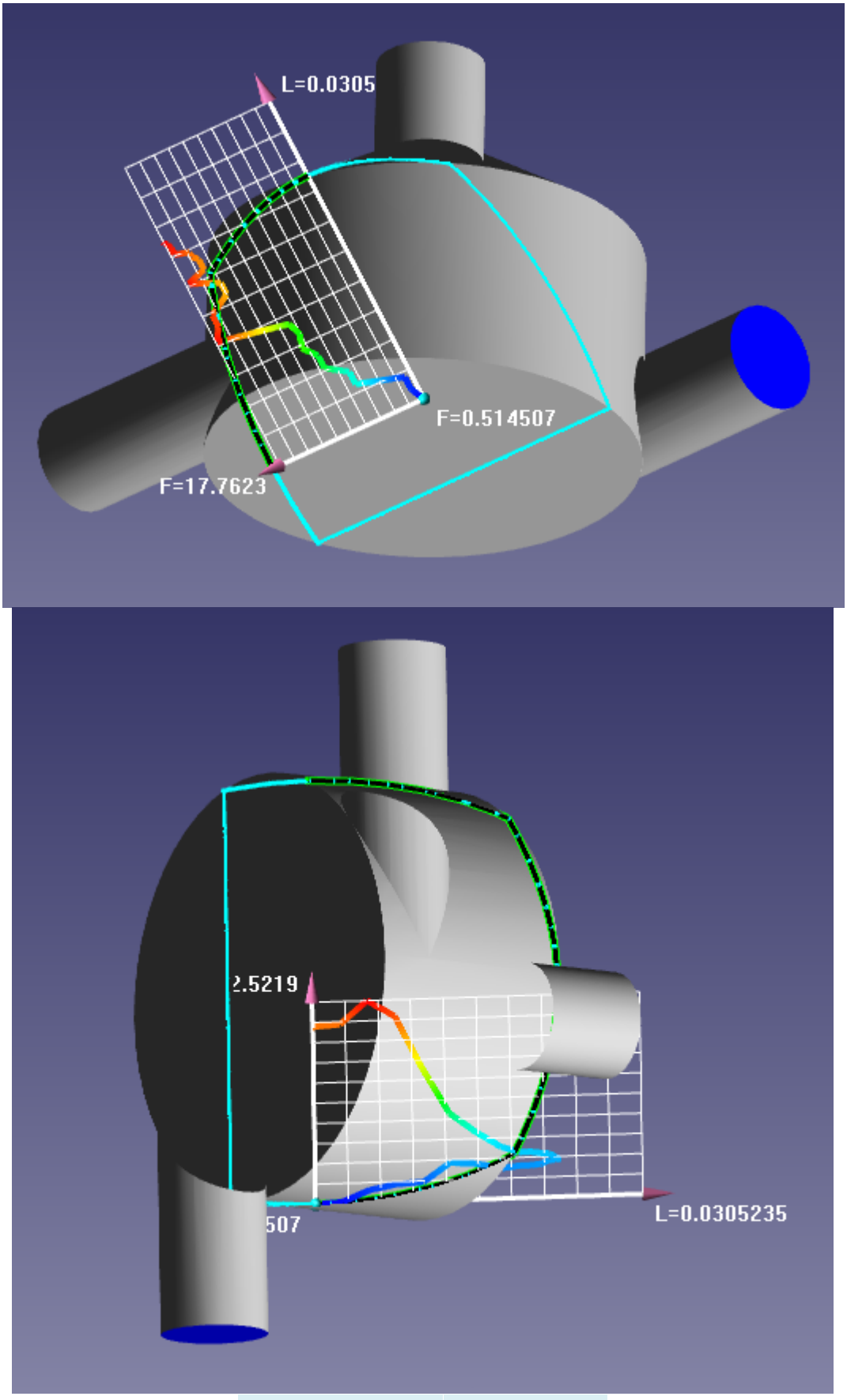


The **Plot along curve** layer in the project tree

Layer **Plot along curve** is designed to display the distribution of scalar[1)] variable along a curve on a plot. The curve formed by the intersection plane, which created a plot with the boundary of the computational domain. Curves can be several.

Note:

[1)] Absolute values are used for vector variables.



Examples of the layer **Plot along curve**

Starting point of the plot coincides with the center of the **Plane**.

Axis of the plot are defined as follows:

Axis of the plot	Base object is a Plane
X, the axis of abscissas Marked on the chart with the letter **L**.	The projection of the curve on one of the coordinate lines **Plane**
Y, the y-axis Marked on the chart with the letter **F**.	Line in the **Plane** is perpendicular to the **X**-axis of the plot.

The plot is displayed as an unlit polygon with a fixed number of units, located in the first quarter of the coordinate axes of the plot. At the positive end of the axes are small lighted cones.

In the plane of the plot displays the following comments:

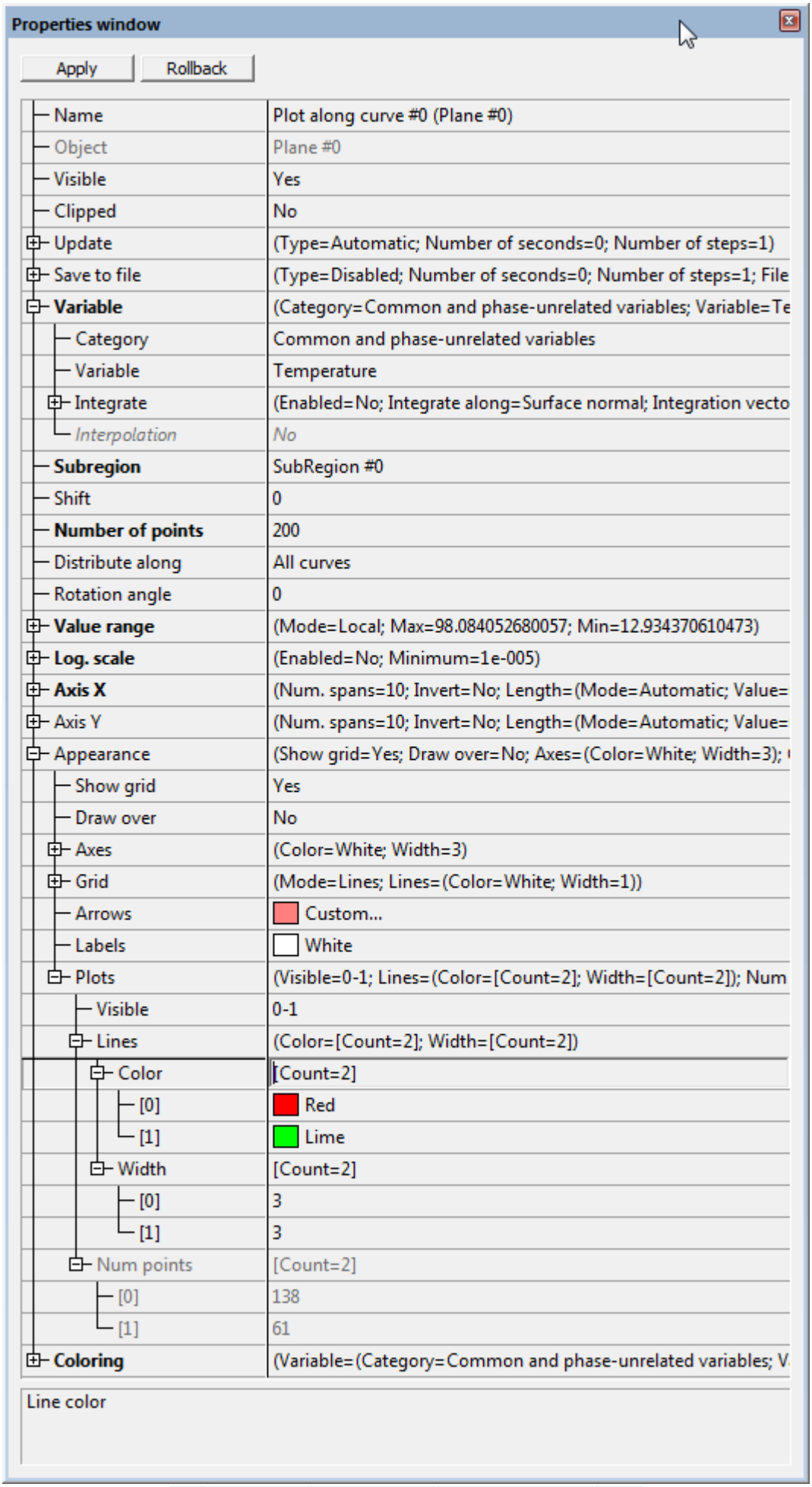
- The length of the x-axis at the positive end of the x-axis.
- The maximum value of a variable near the positive end of the axis of ordinates.
- The minimum value of the variable near the negative end of the axis of ordinates.

Furthermore, depending on the properties set in the plane coordinate plot can be drawn lines in parallel with each of the axes.

The **Plot along curve** layer is built on only one of its quadrants of the coordinate plane.

Line shown in the chart can be colored according to the value of another variable (parameter block **coloring**).

Parameters of the «Plot along curve» layer

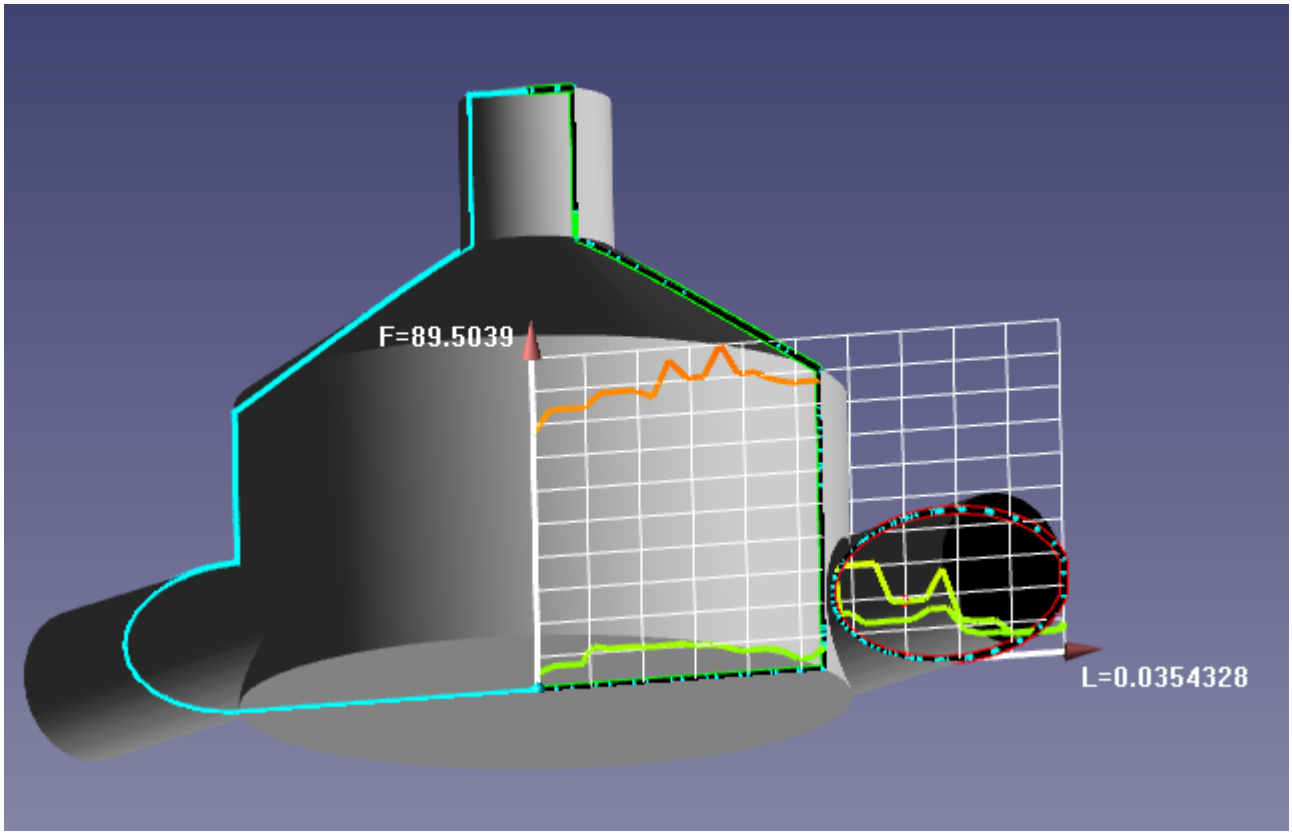


The **Properties** window of the **Plot along curve** layer

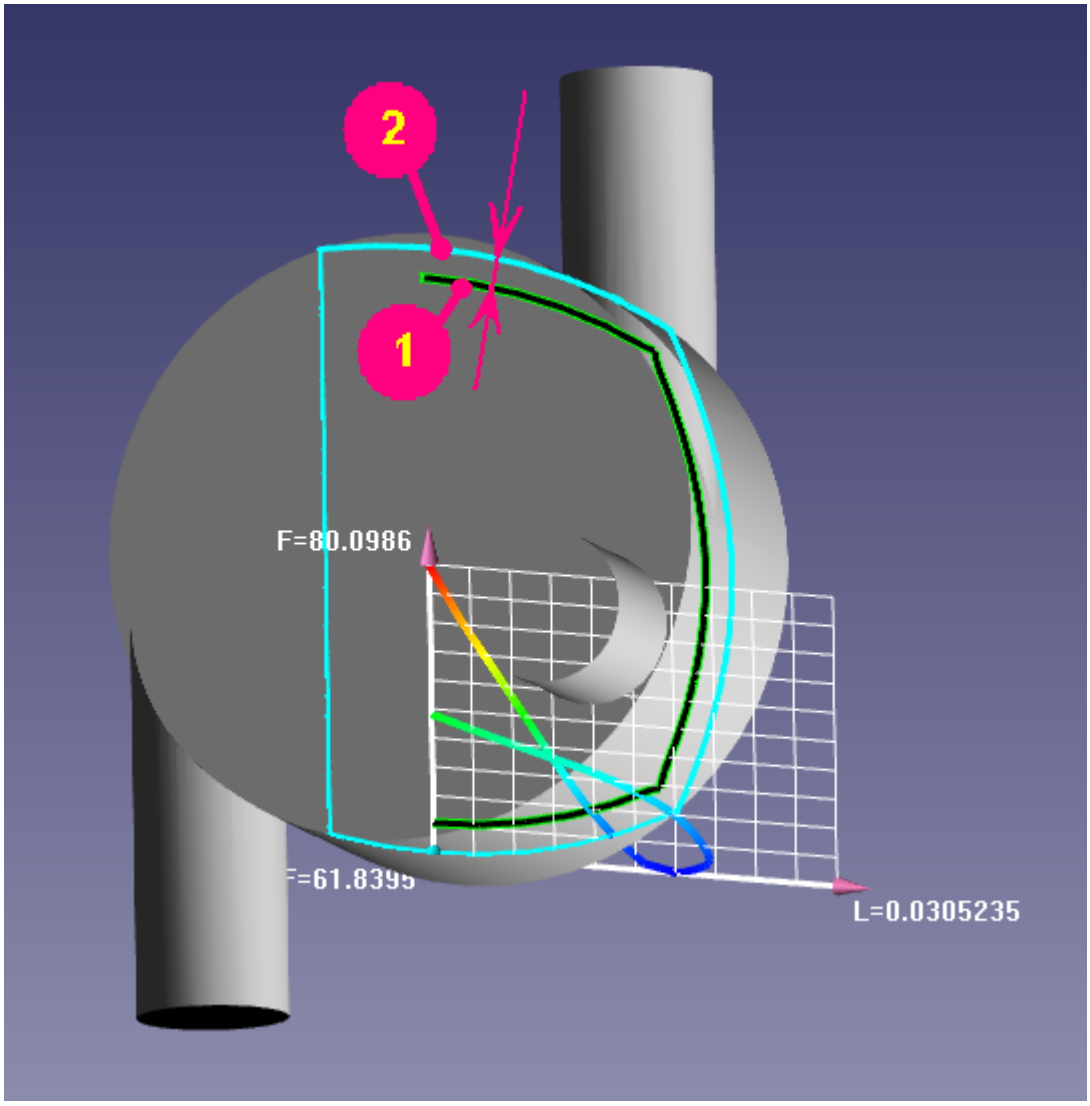
The layer's parameters with the calculated data **Plot along curve**:

Parameter	Description	The dimension of
Name	Layer name (this option allows you to change the default title **"Plot along curve #N (Object)"** formed from the name of the layer type, numbers and **objects** on which a layer is built).	
Object	See General properties of Layers.	
Visible		
Clipped		
Update > ...		
Save to file > ...		
Variable > ...	Variable values on which the plot. See General properties of Layers.	
Shift	Distance, which should be equidistant move the curve in the plane along which a plot, inside the **Subregion**, on which the plot is drown. (See illustration)	[m]
Number of points	The number of points on which the plot.	
Distribute along	• **All curves** - the number of points spread evenly over all curves • **Each curve** - the number of points distributed uniformly in each curve	
Rotation angle	The angle of rotation axes plot around the normal to the **Plane** passing through the **Reference point** (which is also the beginning of the coordinate axes of the plot) (See illustration)	[Degree]
Value range > ...	See General properties of Layers.	
Subregion	Selection of the **Subregion**, which data is used to make the plot	
Subregions		
Subregions > [0]	Selection a **Subregion** for zero plot, for the 1st plot and so on	
Subregions > [1], etc		
Axis X > ...	See General properties of Layers.	
Axis Y > ...		
Appearance > Show Grid	Displaying a grid of the plot	
Appearance > Draw over	Drawing graphics on top of the other elements of the **View** window	
Appearance > Axes > Color	The color and thickness of the chart axes	
Appearance > Axes > Width		
Appearance > Grid > Mode	See General properties of Layers parameters **Appearance > Grid> ...**	
Appearance > Grid > Lines > Color		
Appearance > Grid > Lines > Width		
Appearance > Grid > Fill > Color		
Appearance > Grid > Fill > Opaqueness		
Appearance > Arrows	Color arrows at the ends of the plot axes	

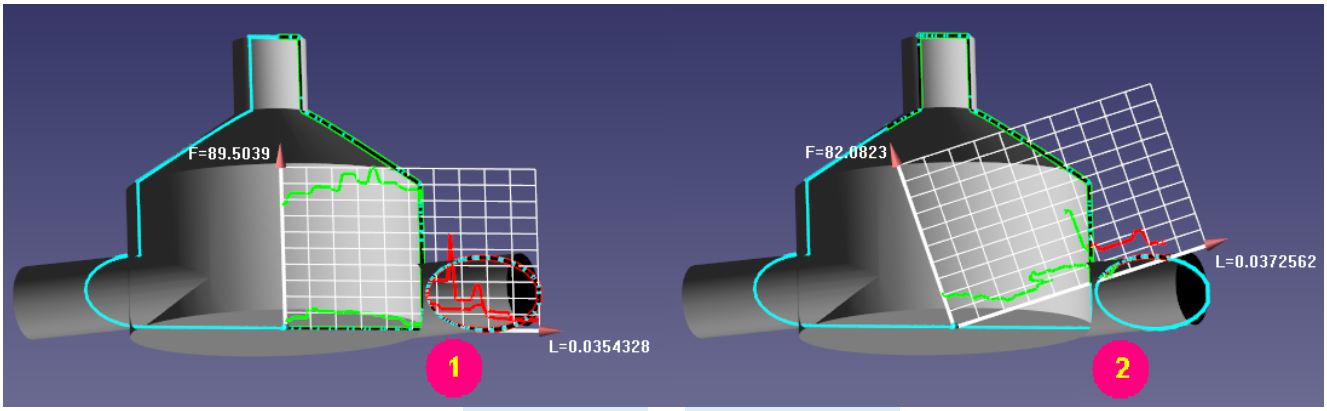
Parameter	Description	The dimension of
Appearance > Labels	Color of labels	
Appearance > Plots > ...	Group of parameters that control the display of lines of graphs	
Appearance > Plots > Visible	Non-displayed graphs corresponding disconnected contours obtained in the cross section of the computational domain **Plane** and enters the operating quadrant **Layer**. These numbers are given as a list whose elements are separated by commas (item can be number graphics, or range of cells). The rest of the graphics will not be displayed. For example, the list of **"0-2.4"**display graphics only for circuits 0, 1, 2, 4. (See illustration)	
Appearance > Plots > Lines > Color > [N] N = 0, 1, ...	Color line N-th plot (Color does not apply if you set the coloring on the additional variable)	
Appearance > Plots > Line > Width > [N] N = 0, 1, ...	Color line N-th plot	
Appearance > Plots > Num points > [N] N = 0, 1, ...	The number of points used to build the N-th plot. These parameters are read-only.	
Coloring > Variable > ...	Staining line graphs in color, depending on the values of a variable on the contours of the cross section of the computational domain and the **Plane** (this variable can be the same or not the same as the variable that defines the path lines). Variable range and palette settings are specified similarly to the standard parameter groups **Variable Range** and **Palette**, see General properties of Layers.	
Coloring > Value range > ...		
Coloring > Log. scale > ...		
Coloring > Palette > ...		



In this example section of the computational domain by a **Plane** consists of two contours



With the parameter **Shift** can move the line (1), which takes the value of the **Variable** from the line (2) The intersection of the **Plane** and the computational domain



This parameter allows you to turn the **Rotation angle** of **Plot along curve** rotating it around the normal to the **Plane** passing through the **Reference point**:

(1) - **Rotation angle = 0**, (2) - **Rotation angle = 18** [degrees]

Window «Info» layer «Plot along curve»

The data displayed in the **Info** window of the **Plot along curve** layer:

Parameter	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current simulated time
Variable	Global information about the primary variable (on which to build a line plot), i.e. common to all fragments of the curve, which is plotted.
[N], N= 0, 1, 2, ...	Blocks of parameters relating to individual fragments of the curve on which a plot containing information about the underlying variable (on which to build a line chart). Displays standard information as well as: • **Significant points** - the number of points on which the plotted • **Length** - length of the curve
Extras. variable	Information about the additional variable (which is held by coloring the plot line)
[N], N= 0, 1, 2, ...	Blocks of parameters relating to individual fragments of the curve on which a plot containing the information about the additional variable.
Palette	Palette, used for coloring the plot line (from the values of the additional variable)

Details about the settings window **Info** for layers see in section Folder «Layers».

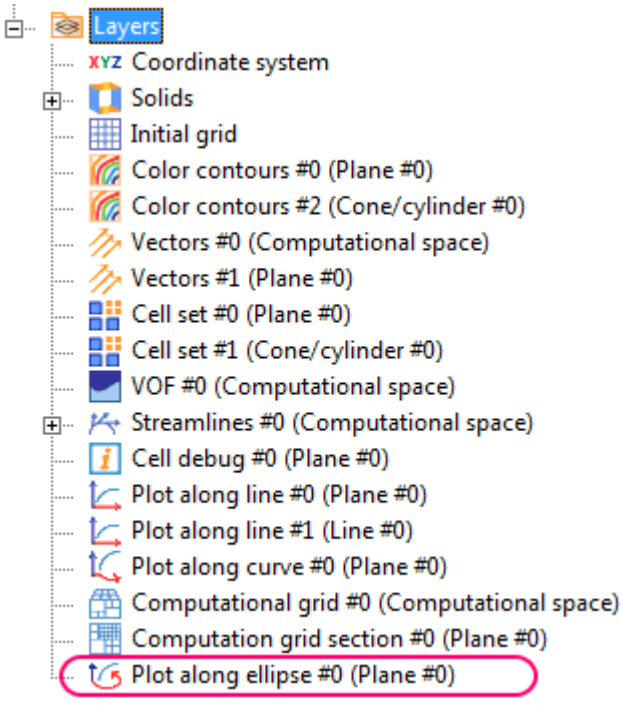
Components of the text file for the layer «Plot along curve»

With each entry in the file shows the following information:

Step	Step number
Time	Time
Variable	Variable that determines the trajectory of the plot
CoverVariable	Variable over which the coloring of the plot line
Plot i	plot along the i-th curve
NumPoints	The number of points on which construction layer
Length	The length of the plot
Avg	The average value of the variable on the plot
Min	The minimum value of the variable on the plot
MinArg	Coordinate of the lowest point on the x-axis
MinPt.x	X-coordinate of the minimum
MinPt.y	Y-coordinate of the minimum
MinPt.z	Z-coordinate of the minimum
Max	The maximum value of a variable plot
MaxArg	The coordinate on the abscissa of the maximum
MaxPt.x	X-coordinate of the maximum
MaxPt.y	Y-coordinate of the maximum

MaxPt.z	Z-coordinate of the maximum
CoverAvg	The average value of the variable in the plot coloring
CoverMin	The minimum value of the variable on the plot coloring
CoverMinArg	Coordinate of the minimum variable coloring on the x-axis
CoverMinPt.x	X-coordinate of the minimum variable coloring
CoverMinPt.y	Y-coordinate of the minimum variable coloring
CoverMinPt.z	Z-coordinate of the minimum variable coloring
CoverMax	The maximum value of the variable on the plot coloring
CoverMaxArg	Coordinate of the maximum variable coloringonthe x-axis
CoverMaxPt.x	X-coordinate of the maximum variable coloring
CoverMaxPt.y	Y-coordinate of the maximum variable coloring
CoverMaxPt.z	Z-coordinate of the maximum variable coloring
Number	Point Number
Arg	The coordinate of a point on the axis of abscissas
Value	Value of the variable at
x	X-coordinate of the point
y	Y-coordinate of the point
z	Z-coordinate of the point

8.1.8.5.8.11 Layer «Plot along ellipse», user interface



The **Plot along ellipse** layer in the project tree

The **Plot along ellipse** layer is designed to display a plot of the scalar[1)] variable along ellipse or a circle or an arc of an ellipse or a *circle.*

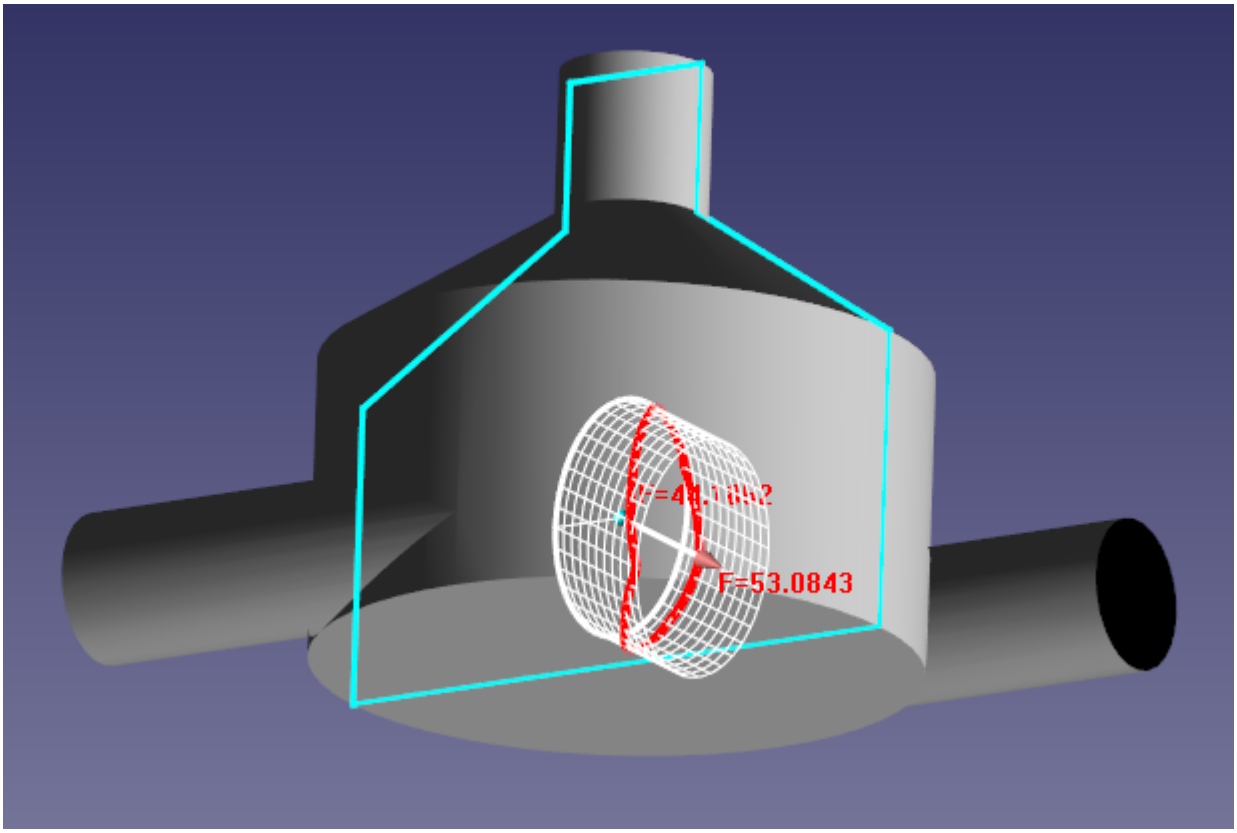
The arc of the ellipse based on the object **Plane** or in a plane perpendicular to the object **Line**.

Center of the ellipse is located in the reference point of the **Plane** or the **Line**.

Notes:

[1] Absolute values are used for vector variables.

[2] For the sake of brevity, the figure on which the plot is typically referred to simply as "an ellipse".



Axes of the plot are defined as follows:

Axis of the plot	Base object is a Plane	Base object is a Line
X, the axis of abscissas (This axis of the plot looped along the contour of the ellipse, is not indicated by the arrow and the words)	Arc of the ellipse in the **Plane** with its center coincident with the center of the **Plane**	Arc of the ellipse in the plane passing through the **Lines** and perpendicular **Lines**.
Y, y-axis Marked on the chart with the letter **F**.	The line passing through the center of the **Plane** and perpendicular to the **Plane**.	**Line**

The plot is displayed as an unlit line, consisting of a fixed number of units, which is located on the lateral surface of the cylinder whose base is an ellipse, and height - Y-axis plot. On the positive end of the Y axis is the arrow - a small lighted taper, showing the direction of the axis Y.

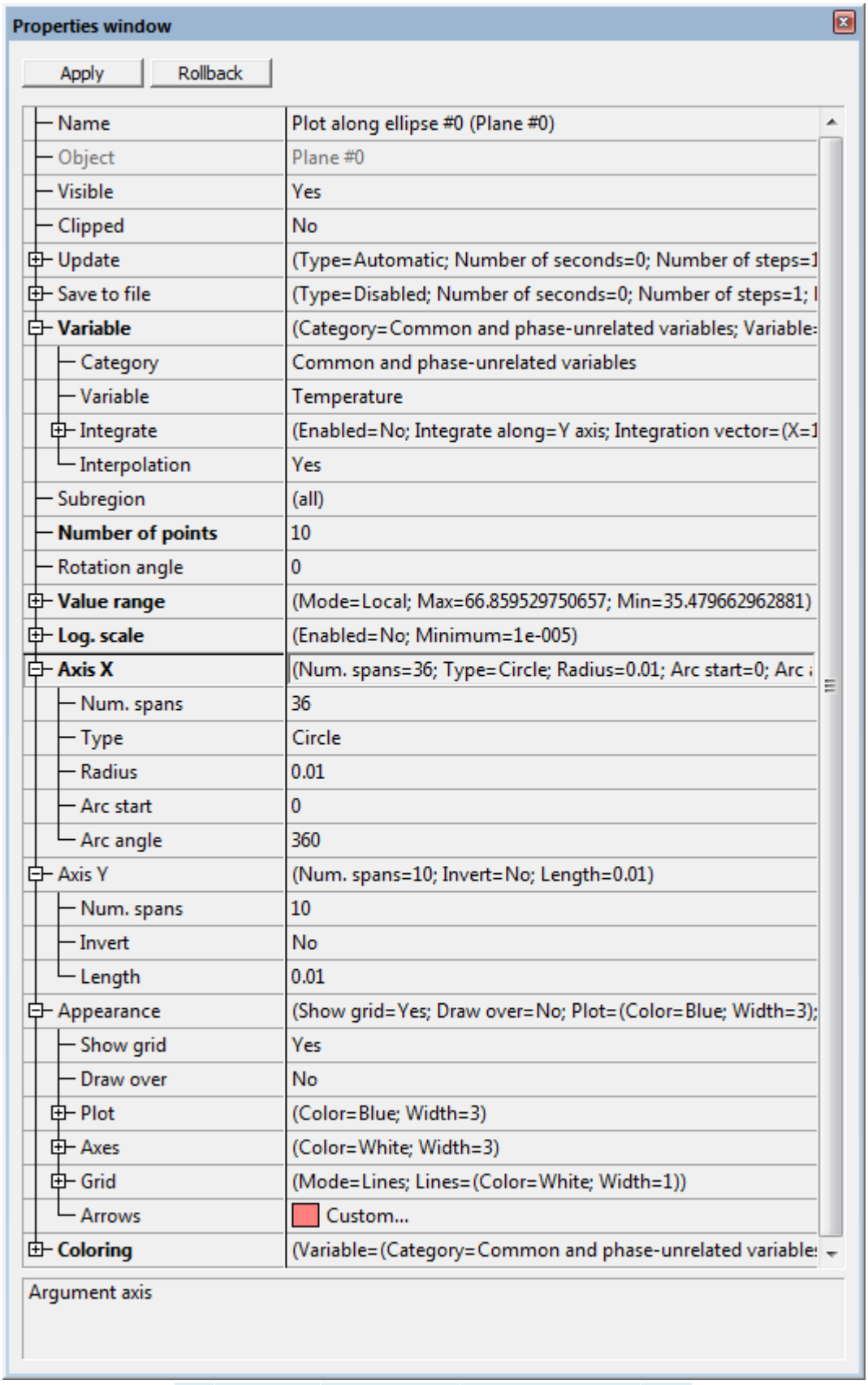
In the plane of the plot displays the following comments:

- The maximum value of a variable near the positive end of the axis of ordinates.
- The minimum value of the variable near the negative end of the axis of ordinates.

Furthermore, depending on the properties set in the plane coordinate plot can be drawn lines in parallel with each of the axes.

Line shown in the chart can be colored according to the value of another variable (parameter block **coloring**).

Parameters of the «Plot along ellipse» layer

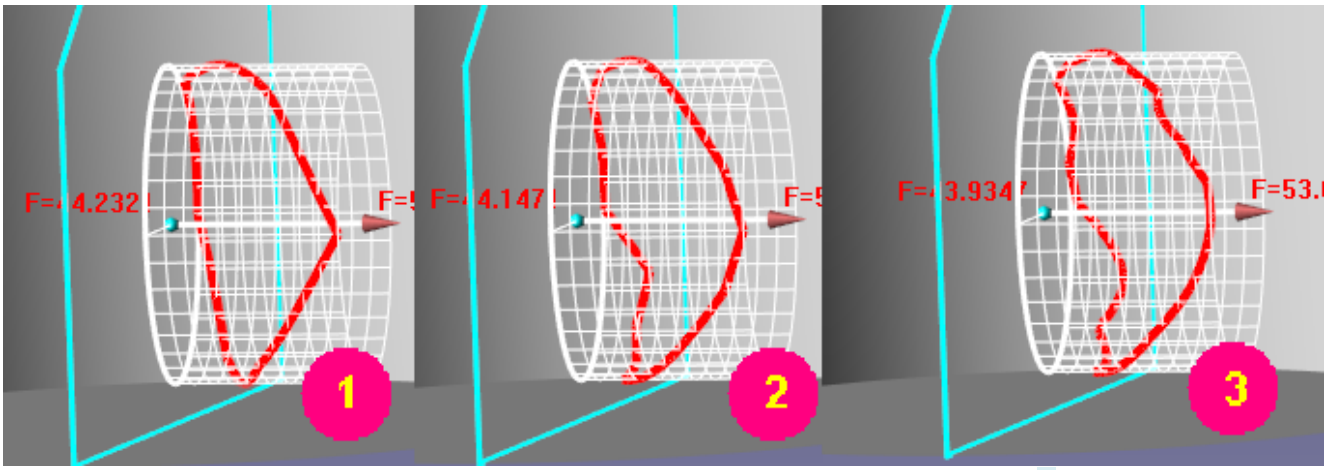


The **Properties** window of the **Plot along ellipse** layer

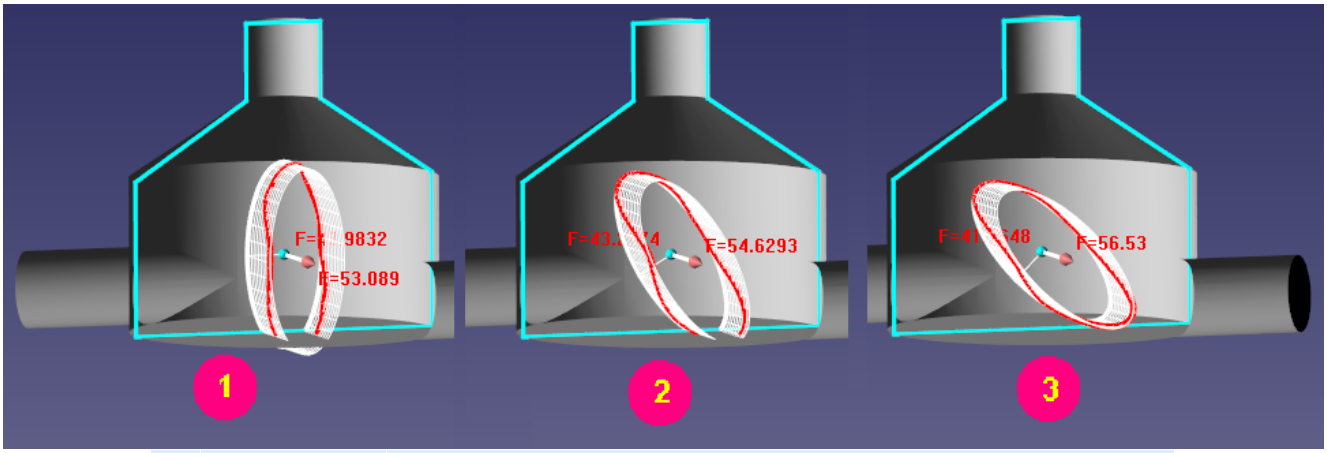
Layer parameters of **Plot along ellipse**:

Parameter	Description	The dimension of
Name	Layer name (this option allows you to change the default title **"Plot along ellipse #N (Object)"**, formed from the name of the layer type, numbers and **Objects** on which the layer is built).	
Object **Visible** **Clipped** **Update> ...** **Save to file > ...**	See General properties of Layers.	
Variable > ...	Variable that determines the trajectory of the plot line. See General properties of Layers.	
Subregion	Limitation layer of a given sub-area. See General properties of Layers.	
Number of points	The number of points on which the plot. (See illustration)	
Rotation angle	The rotation angle of the ellipse on which is plotted in a plane on which it is built with respect to the rotation axis Y plot. (I.e., angle of rotation of the ellipse around the axis of the plane in which it lies) (See illustration)	[Degree]
Value range > ... **Log. scale > ...**	See General properties of Layers.	
Axis X > ...	Parameter group, which manages the ellipse, which is plotted (this ellipse is the x-axis plot).	
Axis X > Num. spans	The number of divisions of the grid, built on the ellipse	
Axis X > Type	Possible options are: • **Circle** - plot is based on a circle or circular arc • **Ellipse** - plot is based on an ellipse, or an arc of an ellipse	
Axis X > Radius	The radius of the circle on which plotted. (Option is available only if the **Axis X > Type = Circle**)	[m]
Axis X > Radius 1 **Axis X > Radius 2**	Dimensions axes of the ellipse, which plotted. (Options are available only if the **Axis X > Type = Ellipse**)	[m] [m]
Axis X > Arc start	Angular coordinate of the start of the arc of the ellipse from which begins to build a plot.	[Degree]
Axis X > Arc angle	The length of the arc of an ellipse, which is plotted; it can be set as a floating-point number. If this parameter is set to **360**, then the plot will be built throughout the ellipse. The value of this parameter should be in the interval (**0**; **360**].	[Degree]
Axis Y > ...	See General properties of Layers.	
Appearance > Show Grid	Displaying a grid of the plot	

Parameter	Description	The dimension of
Appearance > Draw over	Drawing graphics on top of the other elements of the **View** window	
Appearance > Plot > ...	Group of parameters that control the display of the plot line and axis labels graphics	
Appearance > Plot > Color	Line color graphics and signatures. (The line plot can be colored according to the values of a variable, then it this parameter does not apply)	
Appearance > Plot > Width	The thickness of the plot line	
Appearance > Axes > Color	The color and thickness axes of the plot (i.e. the ellipse on which is plotted, and the Y-axis plot, perpendicular to the plane in which lies the ellipse)	
Appearance > Axes > Width		
Appearance > Grid > Mode	See General properties of Layers parameters **Appearance > Grid > ...**	
Appearance > Grid > Lines > Color		
Appearance > Grid > Lines> Width		
Appearance > Grid > Fill > Color		
Appearance > Grid > Fill > Opaqueness		
Appearance > Arrows	The arrow at the end of the Y-axis of the plot (the line coming out of the center of the ellipse and perpendicular to the plane of the ellipse)	
Coloring > Variable > ...	Staining line graphics in color, depending on the values of a variable on the ellipse (this variable can be the same or not the same as the variable that defines the path of the line graphics).	
Coloring > Value range > ...		
Coloring > Log. scale > ...		
Coloring > Palette > ...	Variable range and palette settings are specified similarly to the standard parameter groups **Variable Range** and **Palette**, see General properties of Layers.	



The **Number of points** parameter affects how many breaks will be in the line of the plot (1 -**Number of points = 7**, 2 - **Number of points = 15**, 3 - **Number of points = 200**)



The **Rotation angle** parameter allows you to rotate the ellipse, on which the plot is built:
(1 - **Rotation angle = 0**, 2 - **Rotation angle = 30**, 3 - **Rotation angle = 45**)

The «Info» window of the «Plot along ellipse» layer

The data displayed in the **Info** window of the **Plot along ellipse** layer:

Parameter	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current simulated time
Significant points	The number of points on the ellipse, which plotted
Variable	Information about the variable on which the plot of (path line plot)
Aux. variable	Information about the variable over which the *coloring of* the plot line
Palette	Palette, used for coloring the plot line (from the values of the additional variable)

Details about the settings window **Info** for layers see in section Folder "Layers".

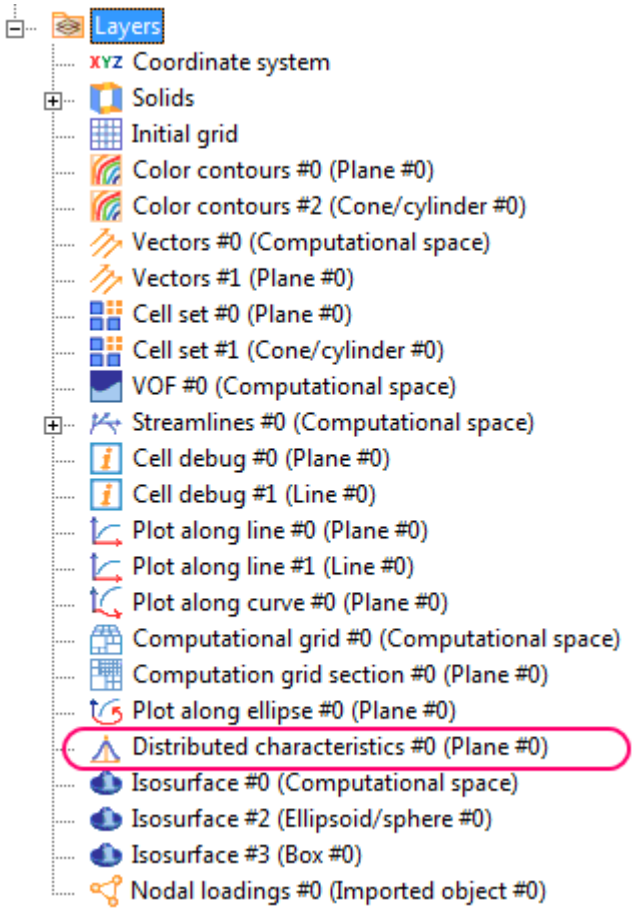
Components of the text file for the layer «Plot along ellipse»

With each entry in the file shows the following information:

Step	Step number
Time	Time
Variable	Variable that determines the trajectory of the plot line
CoverVariable	Variable over which the coloring of the plot line
NumPoints	The number of points on which construction layer
Length	The length of the plot
Avg	The average value of the variable on the plot
Min	The minimum value of the variable on the plot
MinArg	Coordinate of the lowest point on the x-axis
MinPt.x	X-coordinate of the minimum

MinPt.y	Y-coordinate of the minimum
MinPt.z	Z-coordinate of the minimum
Max	The maximum value of a variable plot
MaxArg	The coordinate on the abscissa of the maximum
MaxPt.x	X-coordinate of the maximum
MaxPt.y	Y-coordinate of the maximum
MaxPt.z	Z-coordinate of the maximum
CoverAvg	The average value of the variable in the plot coloring
CoverMin	The minimum value of the variable on the plot coloring
CoverMinArg	Coordinate of the minimum variable coloring on the x-axis
CoverMinPt.x	X-coordinate of the minimum variable coloring
CoverMinPt.y	Y-coordinate of the minimum variable coloring
CoverMinPt.z	Z-coordinate of the minimum variable coloring
CoverMax	The maximum value of the variable on the plot coloring
CoverMaxArg	Coordinate of the maximum variable coloringonthe x-axis
CoverMaxPt.x	X-coordinate of the maximum variable coloring
CoverMaxPt.y	Y-coordinate of the maximum variable coloring
CoverMaxPt.z	Z-coordinate of the maximum variable coloring
Number	Point Number
Arg	The coordinate of a point on the axis of abscissas
Value	Value of the variable at
x	X-coordinate of the point
y	Y-coordinate of the point
z	Z-coordinate of the point

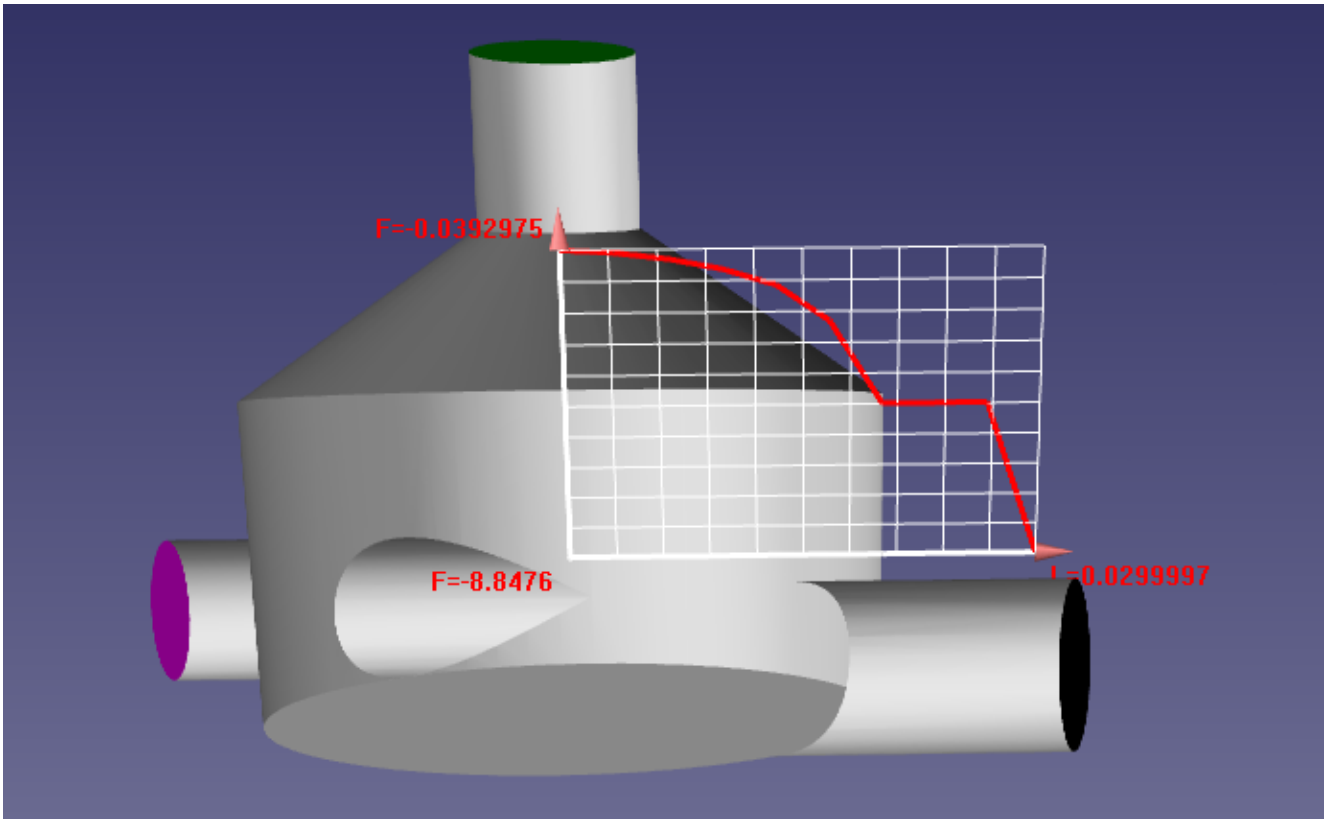
8.1.8.5.8.12 Layer «Distributed characteristics», user interface



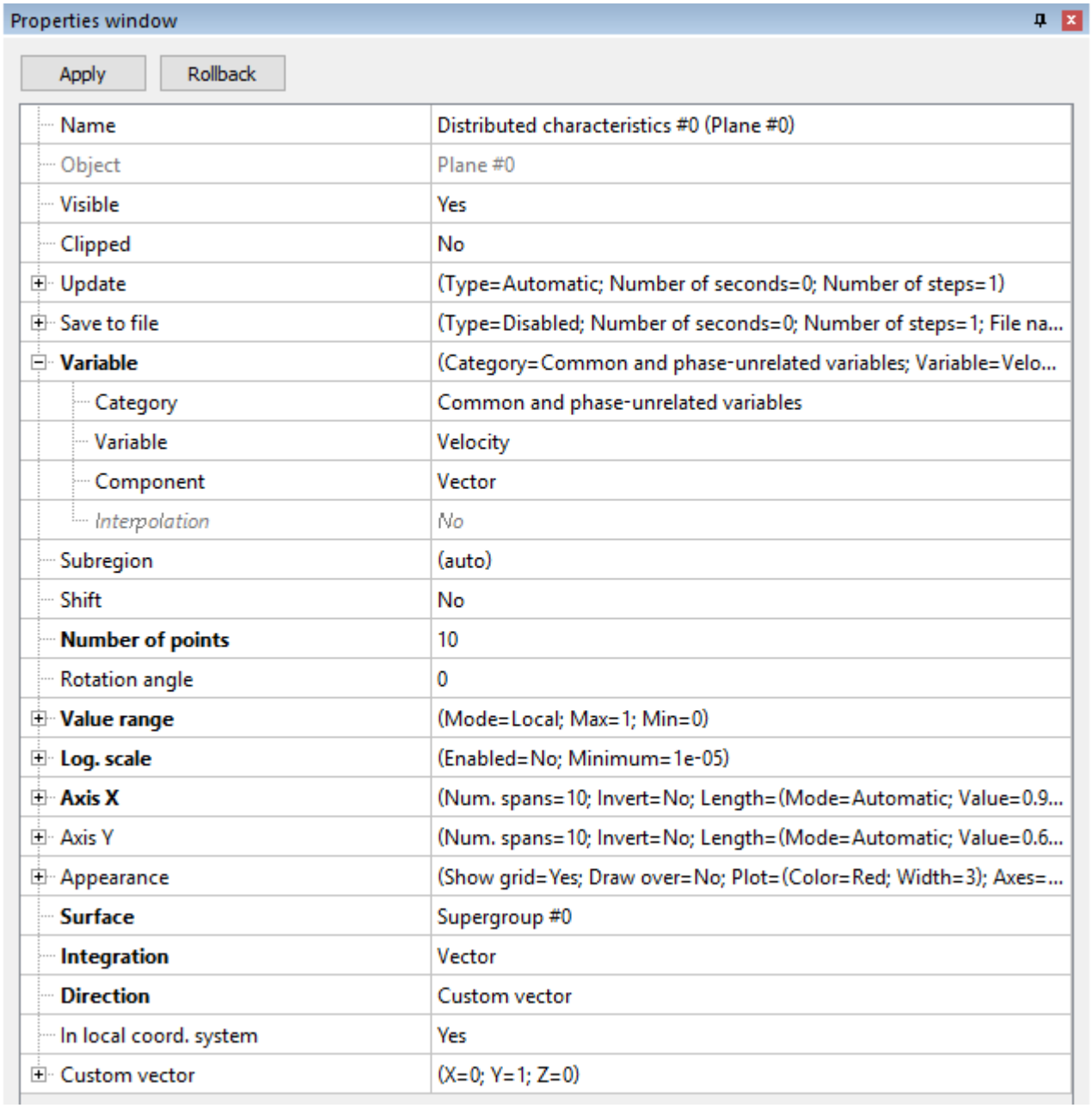
The **Distributed characteristics** layer in the project tree

The **Distributed characteristics** layer displays function $g(x)$, as a plot along a user-defined abscissa axis x of the plot, which visualize distribution of the values of a contour (or a volume) integral of the selected **Variable** over the selected **Surface(s)** (or within volume of **Surfaces**). When the integral is a vector, the program calculates its projection along a user-specified **Direction** (see details in section Layer «Distributed characteristics»).

Axis of the plot	Base object = Plane	Base object =Line
X-axis of the plot (the axis of abscissas) Marked on the plot with the letter **L**.	One of coordinate lines of the **Plane**.	The axis is oriented in the same direction as the **Line**.
Y-axis of the plot (the axis of ordinates) Marked on the plot with the letter **F**.	Line in the **Plane** that is orthogonal to the axis x of the plot.	The axis is oriented orthogonally to the **Line**.



Parameters of the «Distributed characteristics» layer



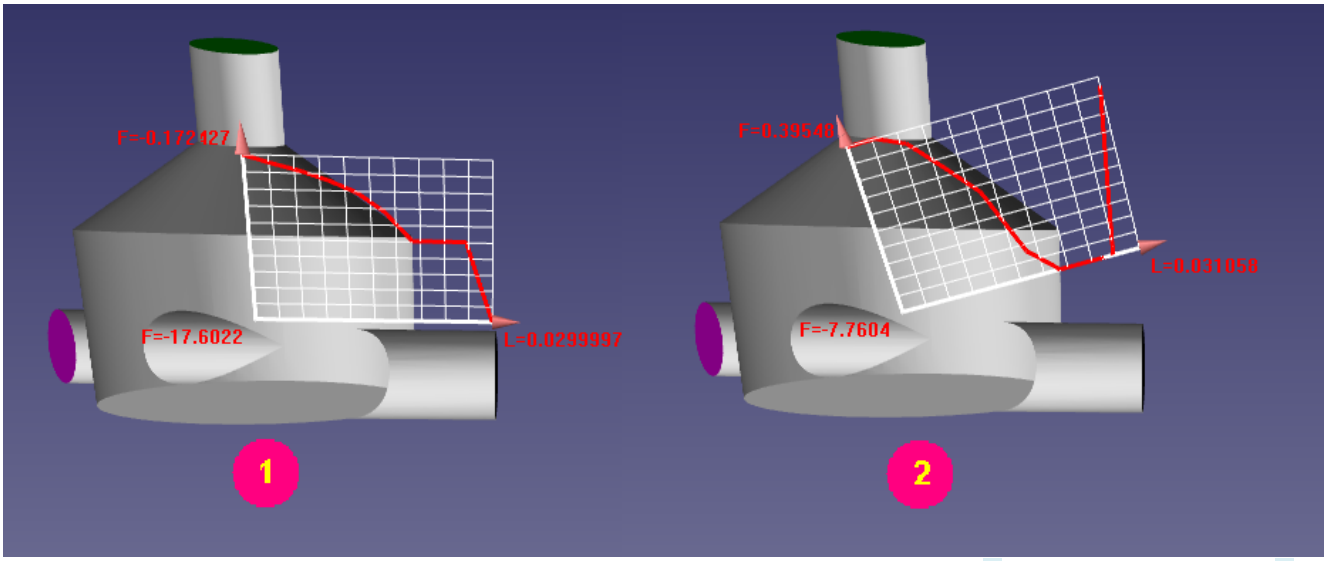
The **Properties** window of the **Distributed characteristics** layer

Parameters of the **Distributed characteristics** layer:

Parameter	Description
Name	Layer name (this option allows you to change the default name **"Distributed characteristics #N (Object)"**, formed from the name of the layer type, numbers and the **Object**, on which the layer is built).
Object	See General properties of Layers.
Visible	
Clipped	
Update> ...	
Save to file > ...	
Variable> ...	Integrable variable. See General properties of Layers.
Subregion	Selection a **Subregion**, where the layer is built. See General properties of Layers.

Parameter	Description
Shift	For the **Distributed characteristics** layer, the **Shift** parameter specified not as a numeric value, but as selection a choice of options (**No** | **Yes**), and has the following meaning: This parameter specifies from where the program takes values of the integrated variable, either from the surface of the **Object** or from the volume near the surface. Possible options are: • **No** - values of the integrable variable are taken from the surface of the **Object** • **Yes** - values of the integrable variable are taken from the volume near the surface of the **Object**, and the **Variable > Interpolation** parameter becomes available. The **Shift** may be useful, for example, by integrating the flow velocity in the flow near the wall with adhesion, as on the wall itself the speed is zero everywhere and the distribution of the characteristics will be uninformative.
Number of points	The number of points on which the plot is built (i.e., the number of points on the x-axis of the plot)
Rotation angle	Angle of rotation of the plot's x-axis, [degrees]. If the plot is built on a **Line**, then the rotation is done around this **Line**. If the plot is built on a **Plane**, then the rotation is done around the **Plane**'s normal that passes through the origin of the plot. (See the illustration after the table)
Value range> ...	See General properties of Layers.
Log scale > ...	
Axis X > ...	
Axis Y > ...	
Appearance > Show grid	Displaying a grid of the plot. Possible options are: **Yes** | **No**.
Appearance > Draw over	The plot will be drawn over other elements in the **View** window. Possible options are: **Yes** | **No**.
Appearance > Plot > ...	Group of parameters that control displaying the plot's line and labels on the plot's axes
Appearance > Plot > Color	Color of the plot's line and labels on axes
Appearance > Plot > Width	Thickness of the plot's line
Appearance > Axes > Color	Color and thickness of the plot's axes
Appearance > Axes > Width	
Appearance > Grid > Mode	See General properties of Layers parameters **Appearance > Grid > ...**
Appearance > Grid > Lines > Color	
Appearance > Grid > Lines > Width	
Appearance > Arrows	Color of arrows at ends of the plot's axes
Surface	Selection of a surface, over which the integrating is done. Possible options are: • **(all)** – integrating will be done over all the surfaces, i.e. over all **Supergroups** and **Imported objects**, on which **Moving bodies** are built and also on the boundary of the computational domain.

Parameter	Description
	• **Computational space** – the distributed characteristics will be calculated not by one or several surfaces but by the volume formed by all the surfaces (**Supergroups**, **Imported objects**, boundaries of the computational domain) and adjacent cutting planes that are orthogonal to the x-axis of the plot. In this case scalar integrating is done, so parameters **Integration**, **Direction**, **In local coord. system**, **Custom vector** (see below) are not available. See details in the section Layer «Distributed characteristics». • individual **Supergroups** and **Imported objects**, on which **Moving bodies** are built
Integration	Contour integration type. Possible options are: **Scalar | Vector**. When you specify **Surface = Computational space** (see above), this parameter is not available and the program will do *scalar* integrating over the volume.
Direction	Selection of the direction, to which the calculated vector contour integral will be projected. Possible options are: • **Custom vector** - use a user-defined arbitrary direction vector. This vector is specified by its components (see parameters **In local coord. system** and **Custom vector > ...** below). • **X axis** - the direction vector is the unit vector of the x-axis • **Y axis** - the direction vector is the unit vector of the y-axis • **Z axis** - the direction vector is the unit vector of the z-axis This parameter is only available when **Integration = Vector** is set (see above).
In local coord. system	The coordinate system (either the absolute coordinate system or the plot's coordinate system), in which **Direction** (see above) for projecting the contour vector integral is specified. Possible options are: • **Yes** - direction of the projecting is specified in the coordinate system of the plot • **No** - direction of the projecting is specified in the coordinate system of the **Region** This parameter is only available when **Integration = Vector** is set (see above).
Custom vector > X	Components of the directing vector, to which the calculated vector contour integral will be projected. These parameters are only available when **Integration = Vector** and **Direction = Custom vector** (see above).
Custom vector > Y	
Custom vector > Z	



Using the **Rotation angle** parameter, you can change direction of the plot's x-axis (1 - **Rotation angle = 0**, 2 - **Rotation angle = 15**)

The Info window of the «Distributed characteristics» layer

The data displayed in the Info window of the **Distributed characteristics** layer:

Parameter	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current simulated time
Significant points	Number of points, on which the plot is built
Variable	Information about the variable on which the layer is built
Discretization step	Distance between points, on which the plot is built
Full integral	Value of the full integral i.e. integral value of the selected variable on the surface multiplied by area of the surface

Details about the parameters in the **Info** window for layers see in the section Folder «Layers».

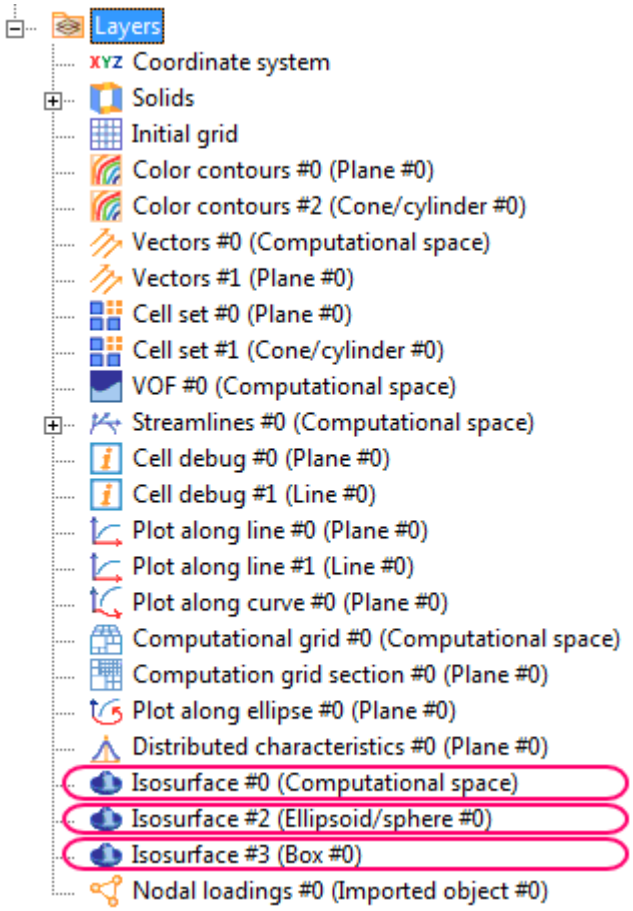
Components of the text file for the layer «Distributed characteristics»

With each entry in the file shows the following information:

Step	Step number
Time	Time
Variable	Variable on which to build a layer
NumPoints	The number of points on which construction layer
Length	The length of the plot
Avg	The average value of the variable on the plot
Min	The minimum value of the variable on the plot

MinArg	Coordinate of the lowest point on the x-axis
MinPt.x	X-coordinate of the minimum
MinPt.y	Y-coordinate of the minimum
MinPt.z	Z-coordinate of the minimum
Max	The maximum value of a variable plot
MaxArg	The coordinate on the abscissa of the maximum
MaxPt.x	X-coordinate of the maximum
MaxPt.y	Y-coordinate of the maximum
MaxPt.z	Z-coordinate of the maximum
Dir.x	The projection of the unit vector direction of the axis of abscissas in the plot axes X, Y and Z coordinate system **Region**
Dir.y	
Dir.z	
Step	The distance between the points at which the plot of
Full Integral	The integral value of the selected variable on a surface, multiplied by the surface area.
Number	Point Number
Arg	The coordinate of a point on the axis of abscissas
Value	Value of the variable at
x	X-coordinate of the point
y	Y-coordinate of the point
z	Z-coordinate of the point

8.1.8.5.8.13 Layer «Isosurface», user interface



The **Isosurface** layer in the project tree

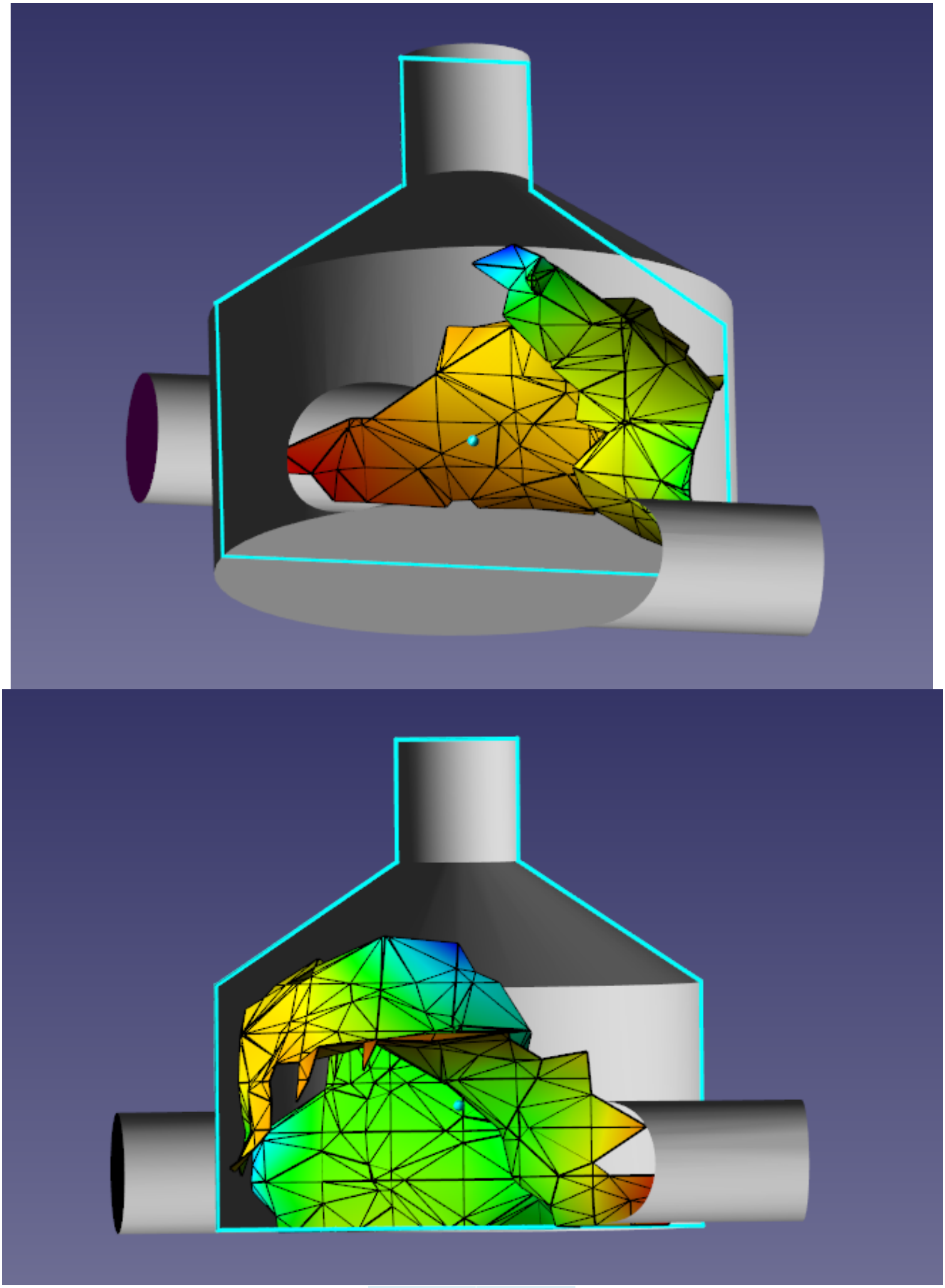
The **Isosurface** layer is designed to display three-dimensional surfaces on which variable[1)] takes the same set point.

You can also customize the display of multiple isosurfaces, corresponding to different values of the variable.

In constructing the isosurface for a single value, it can be colored in accordance with the values of the other ("additional") variable.

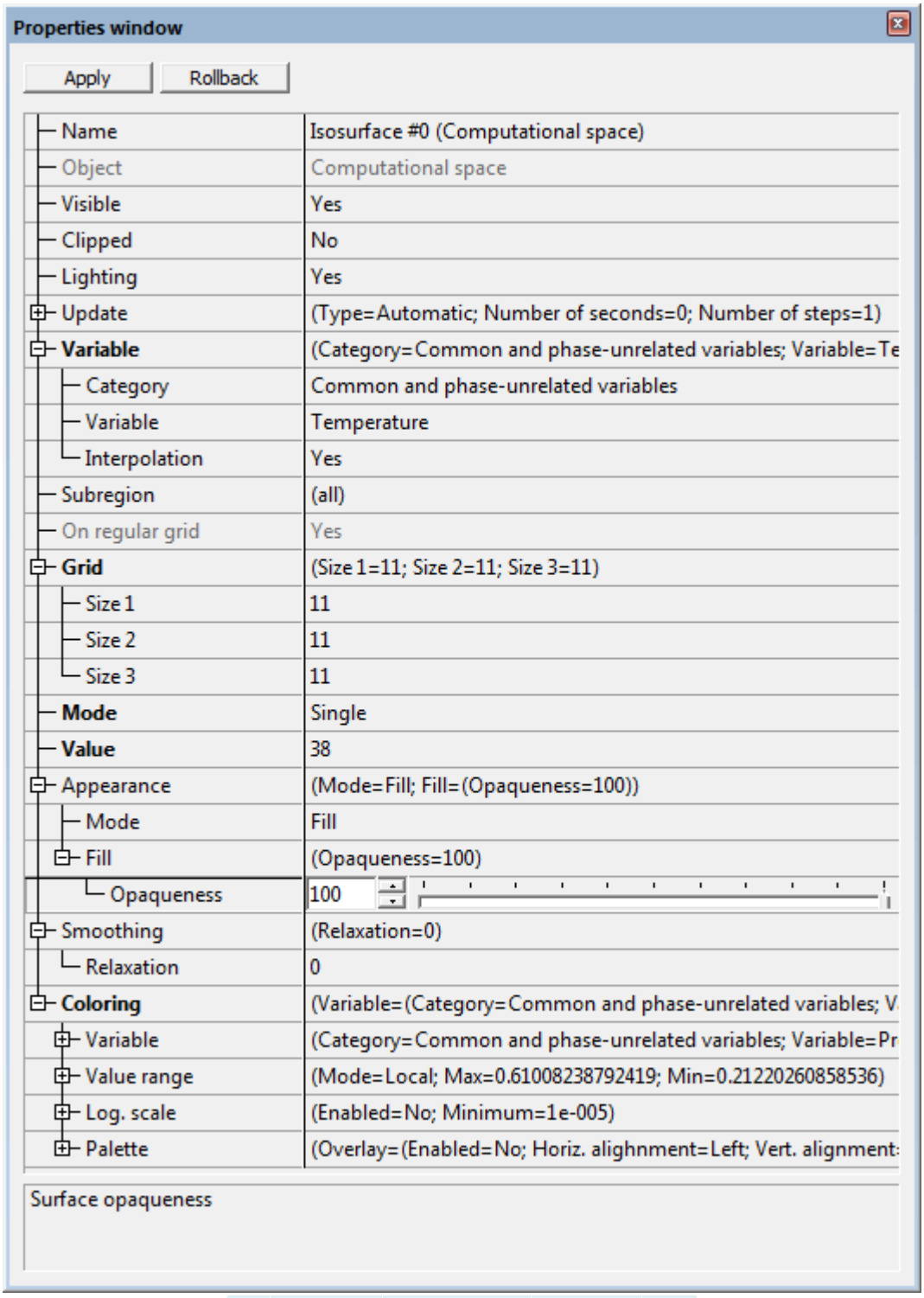
Notes:

[1)] for vector variables their absolute values are used.



Examples of **Isosurfaces**

Parameters of the «Isosurface» layer



The **Properties** window of the **Isosurface** layer

Layer parameters **Isosurface**:

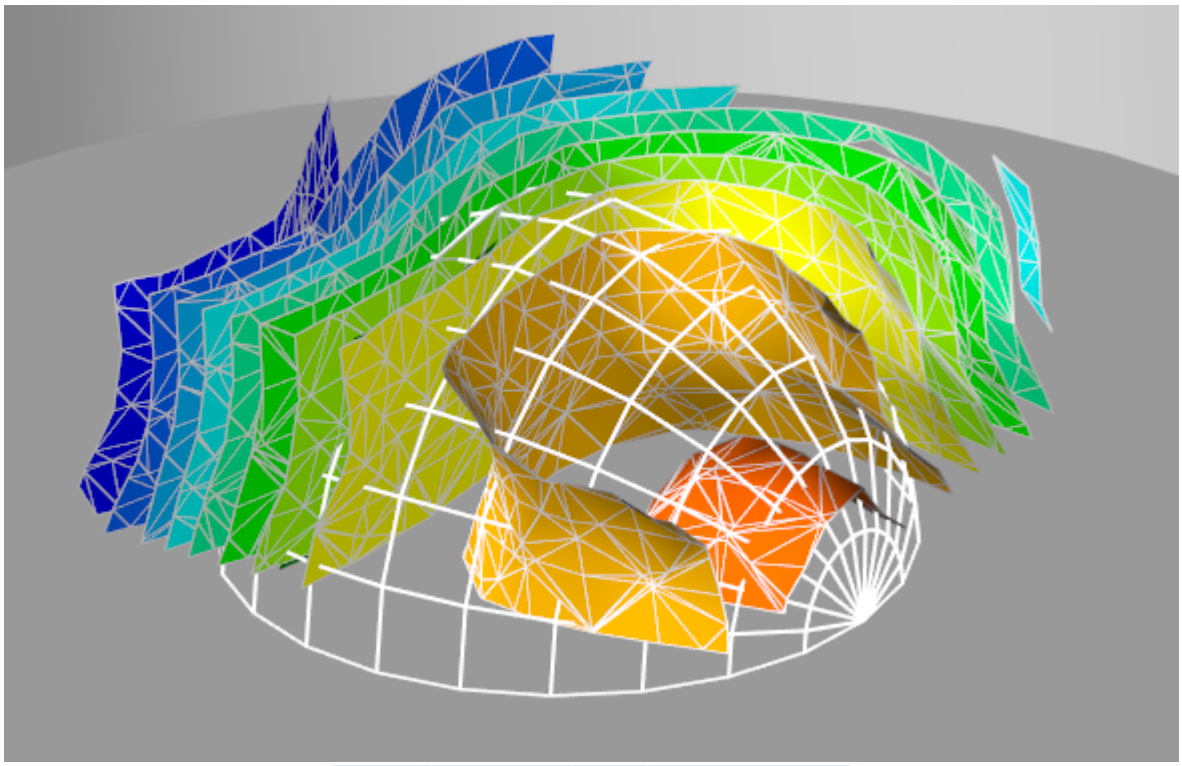
Parameter	Description
Name	Layer name (this option allows you to change the default name **"Isosurface #N (Object)"**, formed from the name of the layer type, number, and **Object**, on which a layer is built).
Object	See General properties of Layers.
Visible	
Clipped	

Parameter	Description
Lighting	
Update> ...	
Variable> ...	Variable for a given value of which is based **Isosurface**. See General properties of Layers. Features of the use of the parameter **Variable > Interpolation** variable **VOF** described in a separate section below.
Subregion	See General properties of Layers.
On regular grid	**Isosurface** construction always formed for her*regular grid*(not on the computational grid, see General properties of Layers), so this parameter is always set to **Yes**, and can not be changed in the current version of the program.
Grid > Local	Possible options are: • **Yes**- to build a regular grid along the axes of the local coordinate system **Object** • **No**- specify the number of grid points along the axis of the absolute coordinate system Option is available when the **Isosurface** is based on an **Object** of finite volume (is not based on the **Computational space**).
Grid > Size 1	The number of nodes on a regular grid in a given direction within the **Object** on which to build **Isosurface**.
Grid > Size 2	
Grid > Size 3	
Mode	Mode of construction **Isosurfaces**: • **Single** - building one isosurface at the specified value of the variable • **Multiple** *)- Construction of a set of isosurfaces from the set of variables defined in the palette
Value	Value of the variable on which to build **Isosurface** (Available if **Mode = Single**)
Value range > ... *)	Parameter group, the control dial to display the variable, see General properties of Layers (You can only select options **Global** and **Manual**) (Available if **Mode = Multiple**)
Palette > ... *)	See Parameters for defining a palette. (Available if **Mode = Multiple**)
Appearance > ...	See General properties of Layers.
Smoothing	Adjusts smoothing of the **Isosurfaces**.
Smoothing > Relaxation	Sets the level of smoothing **Isosurfaces** using relaxation. (See illustration)
Coloring > Variable> ...	Staining line graphics in color, depending on the values of a variable on the x-axis of the plot (this variable can be the same or not the same as the variable that defines the path of the line graphics). Variable range and palette settings are specified similarly to the standard parameter groups **VariableRange** and **Palette**, see General properties of Layers. These options are available only when the **Isosurface** built for one value of the variable (i.e., when the **Mode = Single**).
Coloring > Log scale > ...	
Coloring > Value range > ...	
Coloring > Palette > ...	

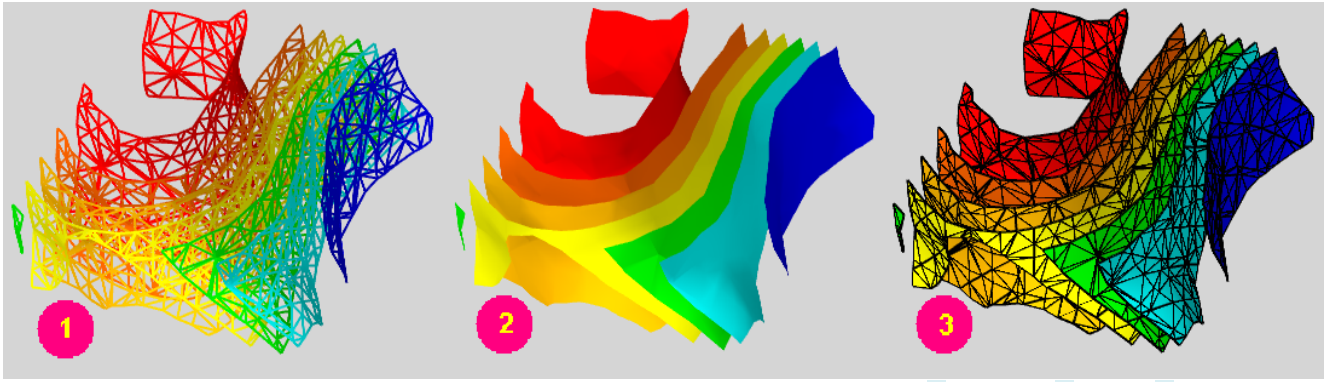
*)

For the construction of multiple isosurfaces might want to store the project *in the presence of communication with solver*
(After clicking on the **Apply** button in the **Properties** window, click the button in the toolbar **Standard** or use the Main Menu command **File > Save**. Also required due to the solver)

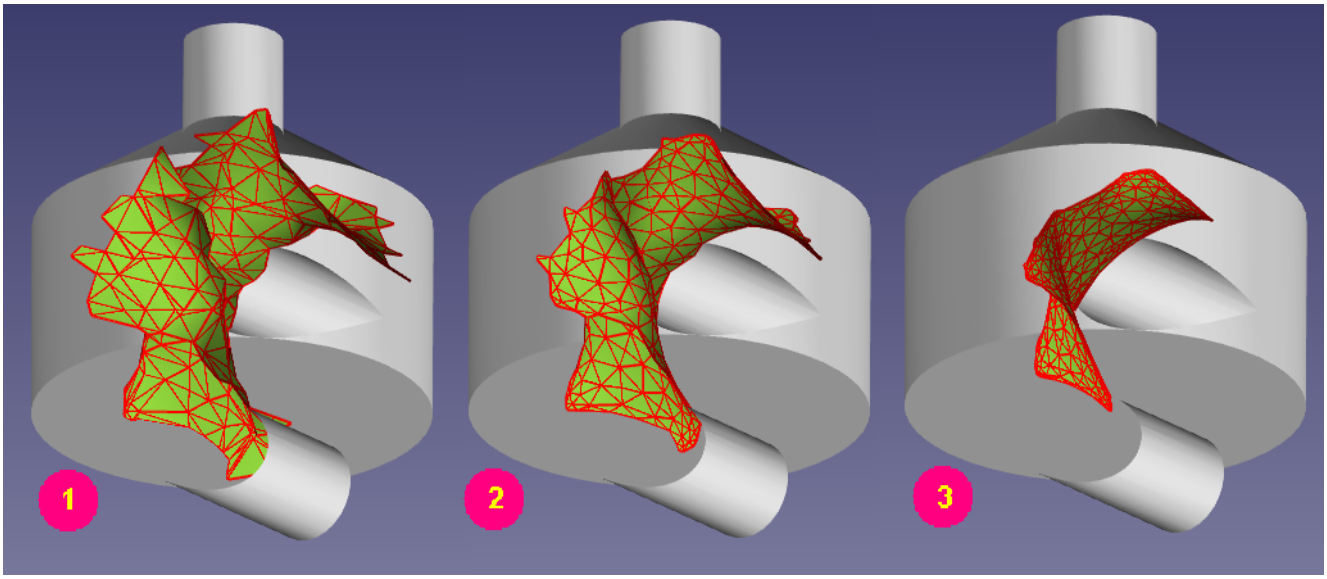
Illustrations



Building **Isosurfaces** when **Mode = Multiple**



Appearance of **Isosurfaces** depending on the **Appearance > Mode** parameter (**1 - Lines**; **2 - Fill**, **3 - Lines and fill**)



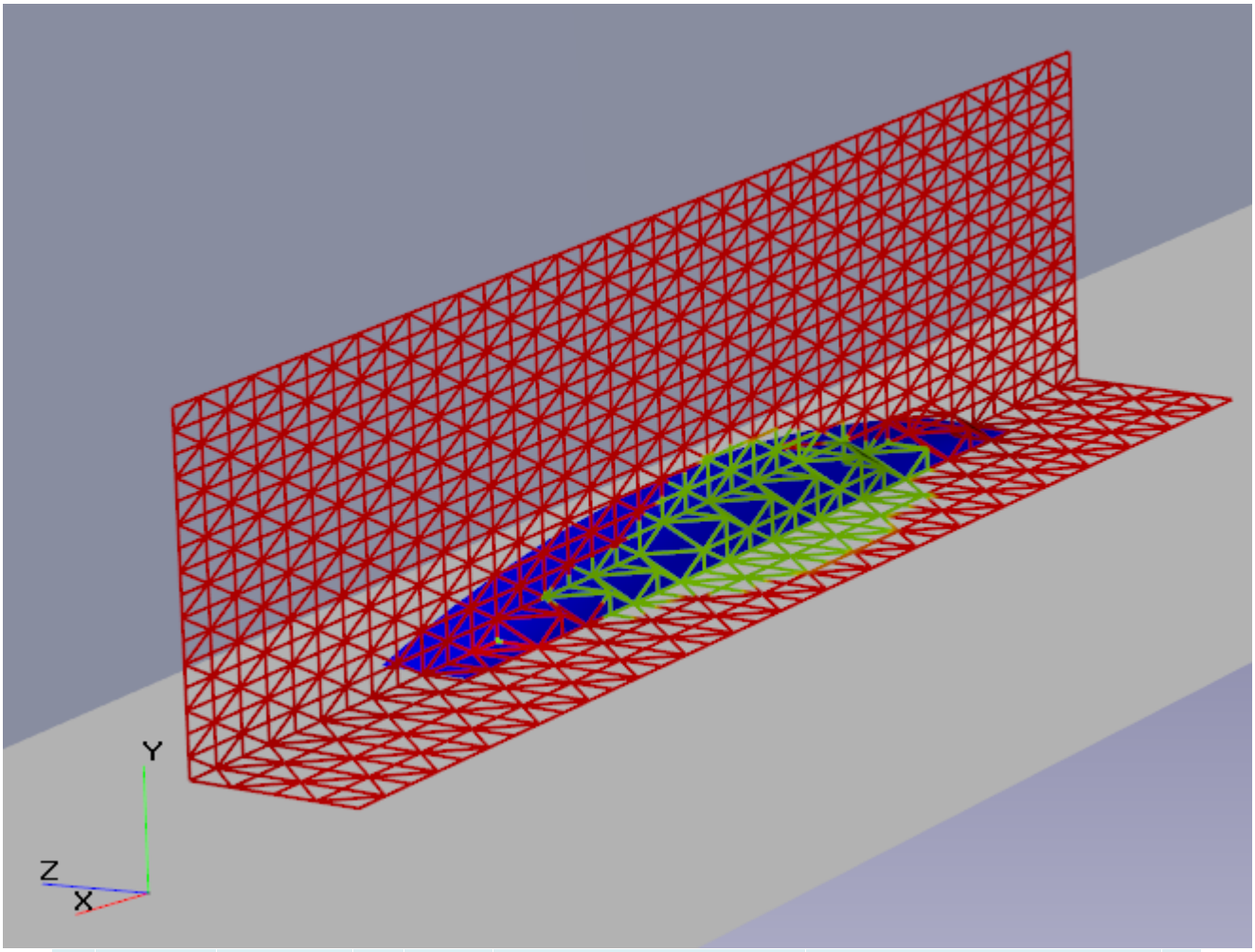
Smoothing **Isosurfaces** (1 -**Relaxation = 0**; 2 - **Relaxation = 2**, 3 - **Relaxation = 5**)

Features using "Variable> Interpolation" variable VOF

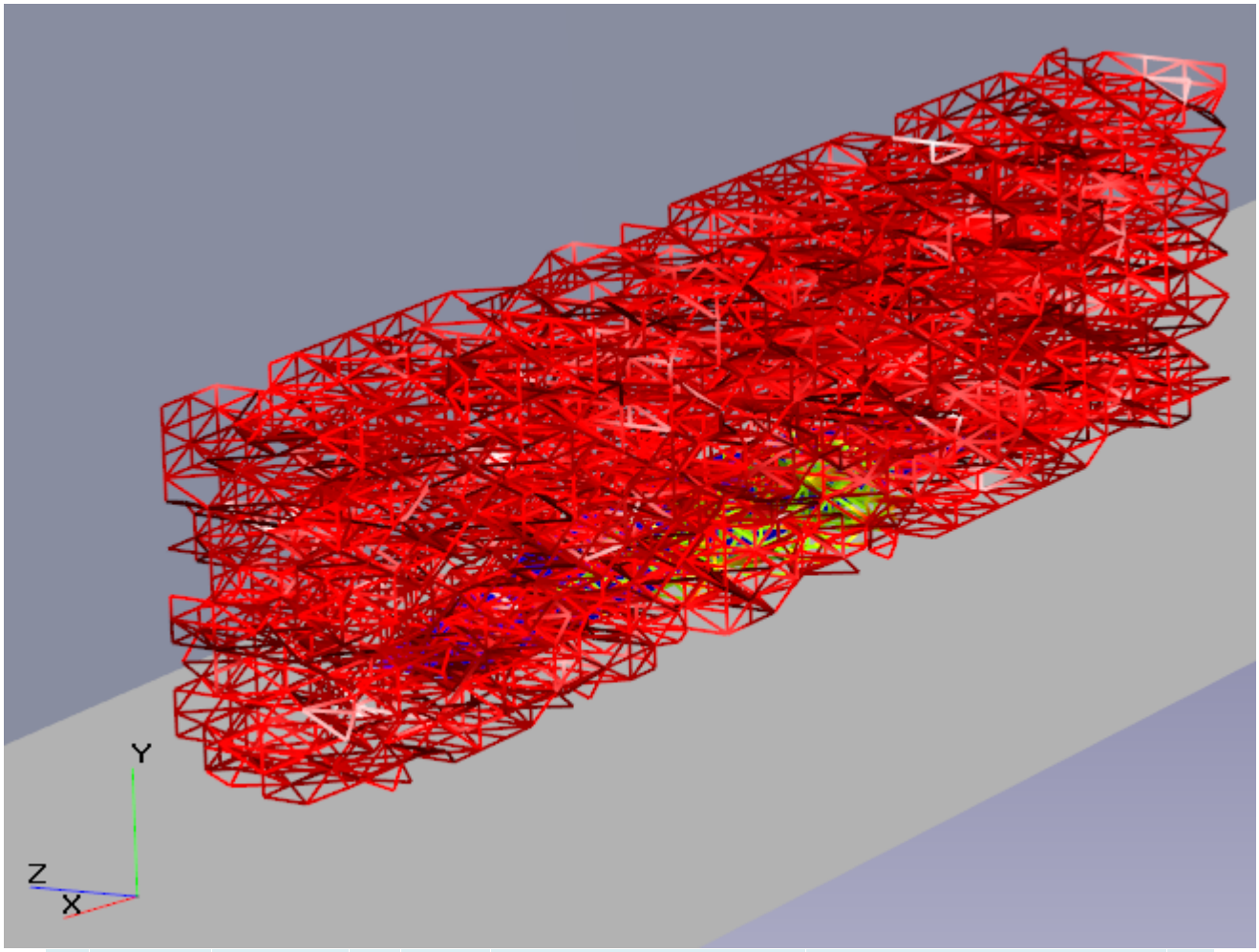
VOF variable anywhere within the liquid volume is set to 1, while on the boundary of the variable definition in the case of equality of values of the variable on the boundary value of the iso-surface, this boundary becomes part of the isosurface and constructed correctly when set to **Variable > Interpolation = No**.

If a given **Variable > Interpolation = Yes**, then there will fluctuate slightly variable values, on which construction isosurface (in the last two significant figures). Because of this variable field becomes uneven, hesitant about unity, and isosurface turns into a mash-up of plane figures that outline the place where the value of the unit remains strictly with holes in places where the value falls below one.

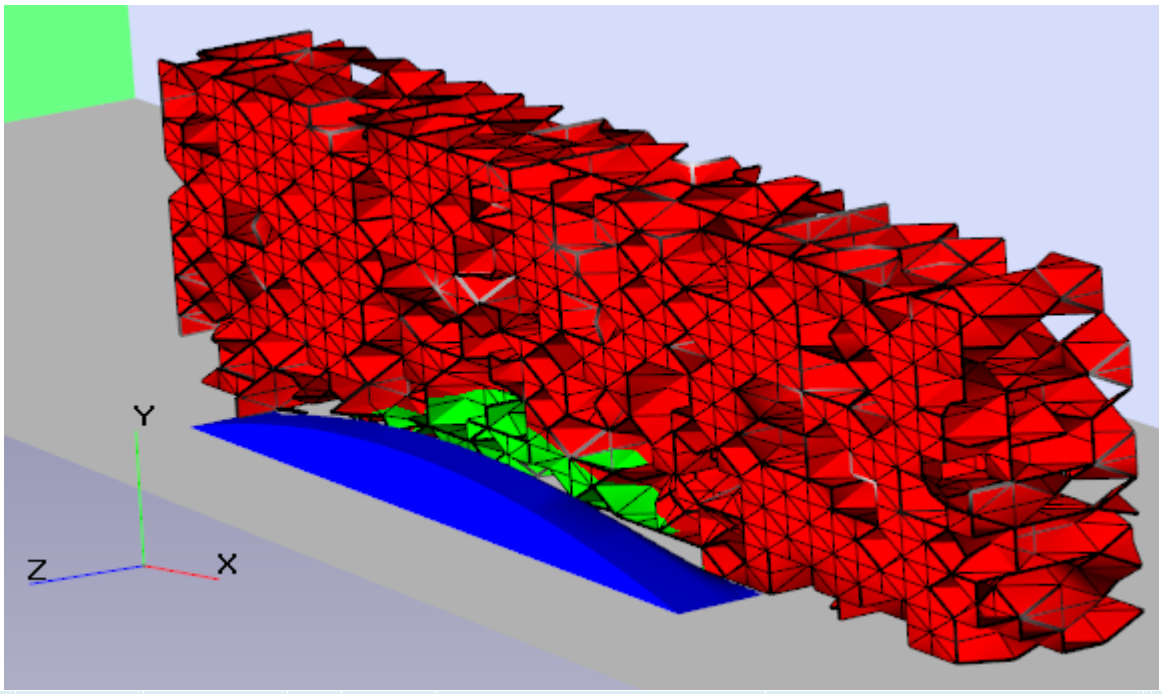
If you still specify **Variable > Interpolation = Yes**,it is recommended to set the **Appearance> Method = lines** so that you can at least partially see inside the isosurface through the gaps between the segments forming the isosurface.



For **Isosurface** built by the **VOF** variable, parameters have been set: **Variable > Interpolation = No** and **Appearance> Mode = Lines**



For **Isosurface** built by the **VOF** variable, parameters have been set: **Variable > Interpolation = Yes** and **Appearance> Mode = Lines**



For **Isosurface** built by the **VOF** variable, parameters have been set: **Variable > Interpolation = Yes** and **Appearance > Mode = Lines and fill**

Window «Info» layer «Isosurface»

Information window[Isosurface #0 (Computational space)]

Name	Value
Solver data	Present
Step number	96
Time	0.96
Variable	TEMP
Block	Heat transfer
Phase	All phases
Local max.	71.311497510514
Local min.	-0.08504689787052
Global max.	72.24850625723
Global min.	-1.5763649448044
Value	0
Aux. variable	TEMP
Block	Heat transfer
Phase	All phases
Local max.	1.505676942393
Local min.	-0.084879057722432
Global max.	72.24850625723
Global min.	-1.5763649448044
Palette:	
	1.5057
	1.3466
	1.1876
	1.0285
	0.86945
	0.7104
	0.55134
	0.39229
	0.23323
	0.074177
	-0.084879

The **Info** window of the **Isosurface** layer

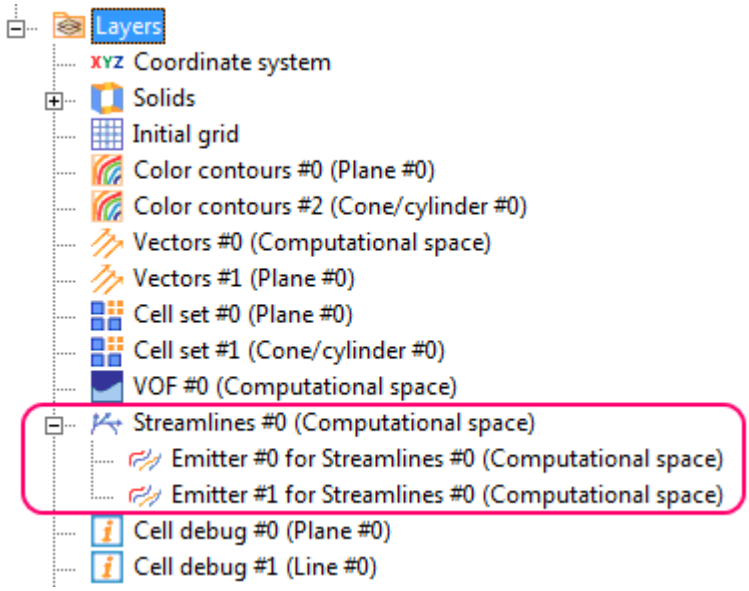
The data displayed in the **Info** window of the **Isosurface line** layer:

Parameter	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current simulated time
Variable	Information about the variable on which to build a layer **Isosurface**

Parameter	Description
Value	Value of the variable on which to build the isosurface (available when **Mode = Single**)
Aux. variable	Information about the variable, which is held by coloring the isosurface (available when **Mode = Single**)
Palette	Palette, used for coloring the isosurface (available when **Mode = Single**)

Details about the settings window **Info** for layers see in section Folder "Layers".

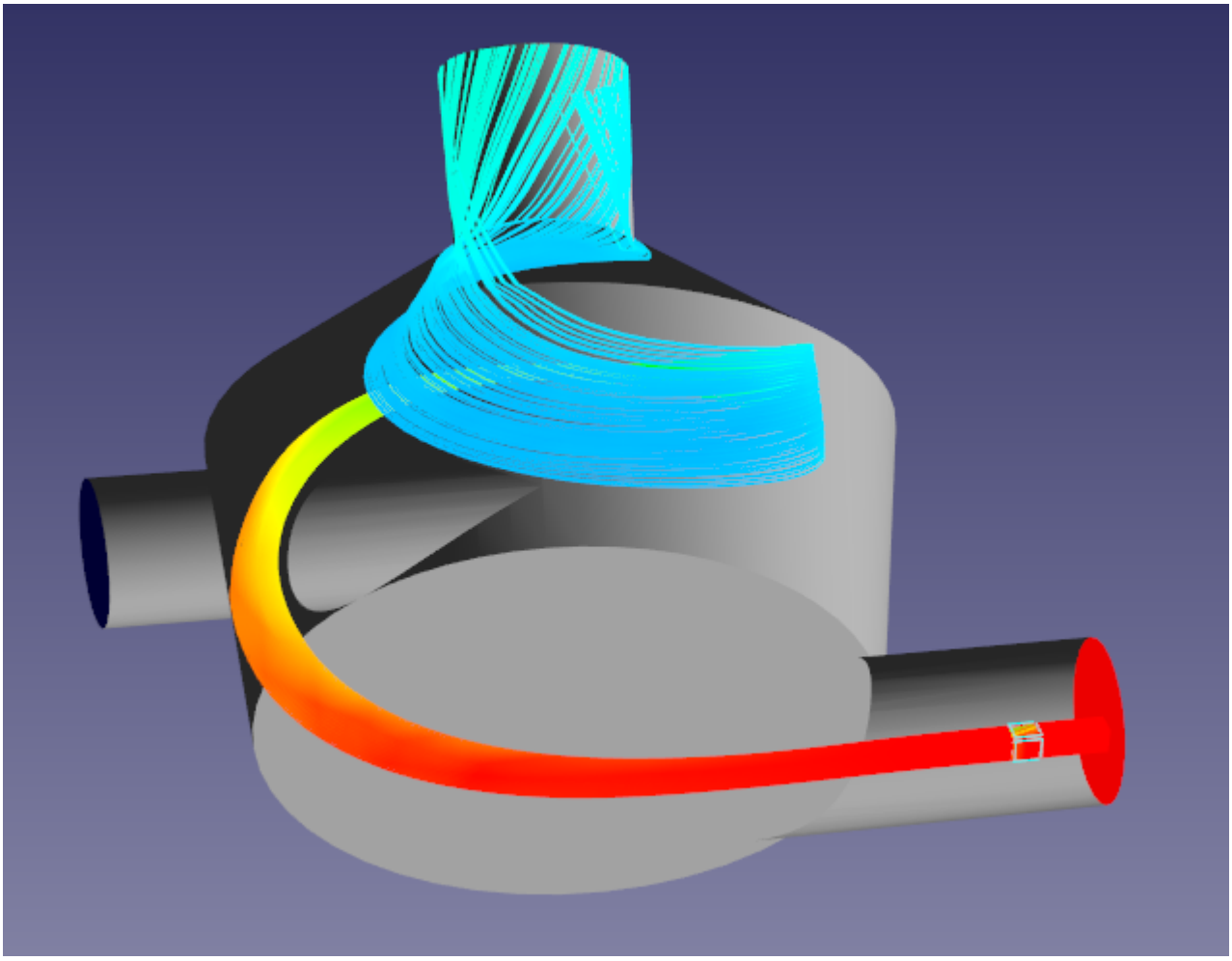
8.1.8.5.8.14 Layer «Streamlines» and element «Emitter for streamlines», user interface



The **Streamlines** layer in the project tree

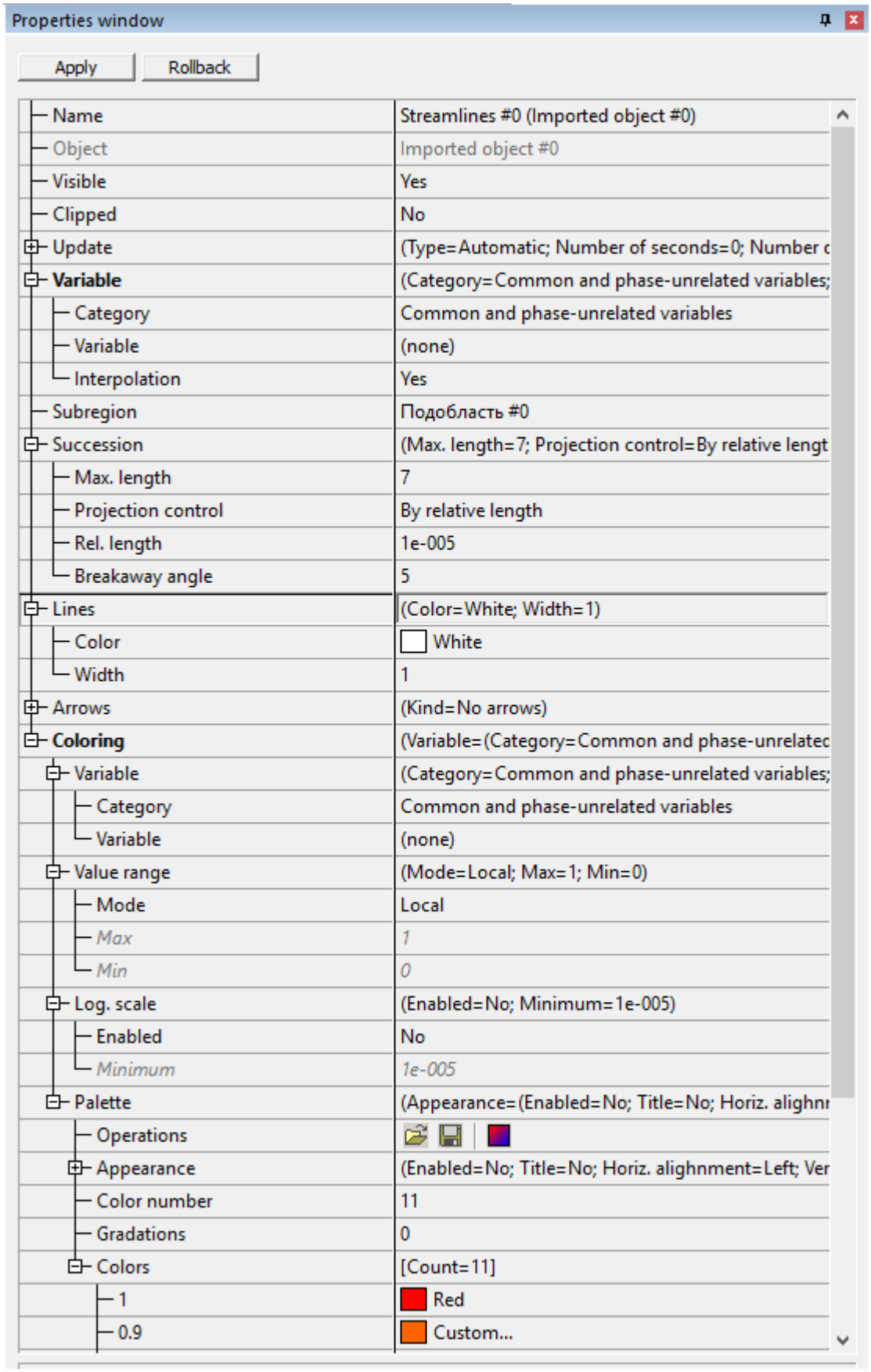
The **Streamlines** layer visualizes streamlines of a vector variable (for example, streamlines of the variable **Velocity** visualize motion of the fluid's and distribution of values of some scalar value along the streamlines).

See also section Layer «Streamlines».



Example of the **Streamlines** layer
Here **Streamlines** display motion of the fluid (emitter of the streamlines is a small **Box** located on the right).
Color of the **Streamlines** visualizes the **Temperature** scalar variable.

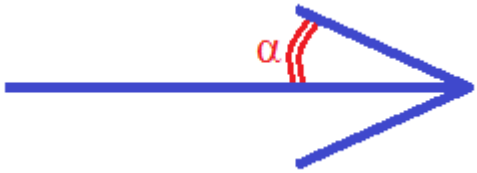
Parameters of the «Streamlines» layer



The **Properties** window of the **Streamlines** layer

Parameters of the **Streamlines** layer:

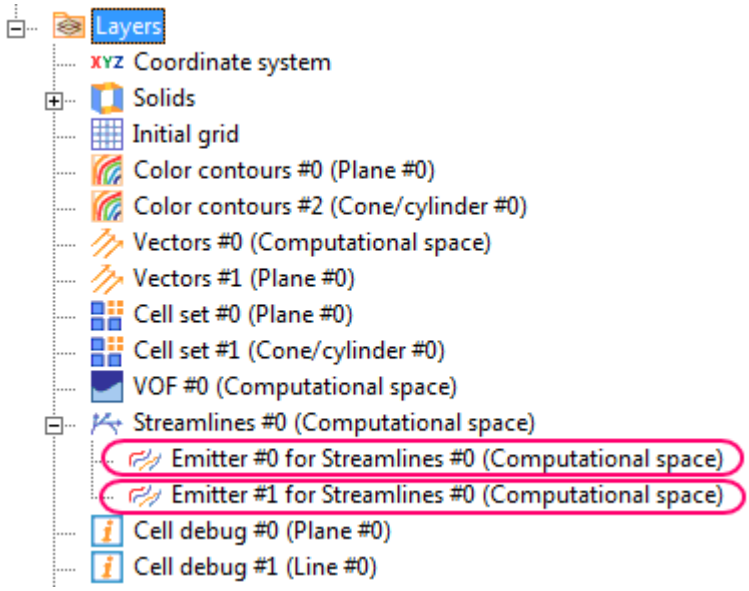
Parameter	Description
Name	Name of the **Layer** (this parameter allows you to change the standard name "**Streamlines #N (*Object*)**" that is formed based on the layers type, a number and the **Objects** , on which the **Layer** is built).
Object	The geometry **Object**, on which the **Layer** is built. This field is read-only and you cannot edit it. **Streamlines** can be built on the following **Objects**: • in the **Computational space** • on a **Plane** • on the surface of a **Supergroup** • on the surface of an **Imported object**, on which a **Moving body** is set See subsection *"Specifics of streamlines on a surface"* below for details about building **Streamlines** on surfaces.
Visible **Clipped** **Update> ...** **Variable > ...** **Subregion**	See General properties of Layers.
Succession > Cell fraction*)	The distance between two nodes is the streamline; it is set as a fraction of the diagonal of the cell. This parameter specifies a constant ratio, on which the local cell size is multiplied to obtain the length of the next straight fragment of the streamline.
Succession > Max. length	The maximum length of a streamline, [m]. The zero value is used to build unlimited streamlines. (See the illustration)
Succession > Cell boundaries*)	This parameter specifies whether the program will place additional nodes of the streamline at cell boundaries. Possible options are: • **No**: no additional nodes will be placed at cell borders • **Yes**: the additional nodes will be placed
Succession > Reflect from walls*)	Behavior of a streamline when it hits a **Wall**. This parameter specifies behavior of a streamline when it hits a boundary condition **Wall**. Possible options are: • **No**: the streamline terminates when it hits a **Wall** • **Yes**: the streamline doesn't terminate and continues its trajectory on the surface of the **Wall** according to the current computation results; the streamline can eventually go back to the **Computational space** after breakaway from the surface. This setting is recommended to view how the forward flow moves over the surface of a body. (See the illustration)
Succession > Projection control**)	This parameter defines how a **Streamline**, which is built on a surface, will terminate depending on the value of the tangential projection of the visualized vector variable. Possible options are: • **By absolute length** - a streamline will terminate if the absolute value of the projection of the visualized vector variable on the surface (the tangential component) is less then the specified value. • **By relative length** - a streamline will terminate if ratio of the tangential component to the maximal (over the whole vector field) length of the visualized vector variable is less then the specified value.
Succession > Abs. length**)	The threshold value of the tangential component of the visualized vector variable that causes termination of a streamline. This parameter is available if **Projection control = By absolute length**.

Parameter	Description
Succession > Rel. length[**)]	The threshold ratio of the tangential component of the visualized vector variable to the maximum value of this variable over the whole vector field. This parameter is available if **Projection control = By relative length**.
Succession > Breakaway angle[**)]	The threshold value of the angle between direction of the visualized vector variable and the normal to the surface, on which the **Streamlines** are built, [Degree]. A streamline will terminate if the angle between the variable's vector and the normal to the surface becomes less then the **Breakaway angle**. The value of the **Breakaway angle** parameter can be set in the range from **0** to **90** degrees.
Lines > Color	Color of the streamlines (if you don't specify coloring depending on values of some variable). This parameter is only available if **Coloring > Variable > Variable = (none)**. (See the illustration)
Lines > Width	Thickness of **Streamlines**
Arrows	Arrows indicate direction of motion along the **Streamlines**
Arrows > Kind	Type of displaying arrows on the **Streamlines**. Possible options are: • **No arrows**: arrows are not displayed • **Lines**: arrows are displayed (with heads displayed as lines) • **Triangles**: arrows are displayed (with heads displayed as triangles) • **Pyramid**: arrows are displayed (with heads displayed as pyramids) (See the illustration)
Arrows > Size	Size of the arrows
Arrows > Angle	The angle between lines of the arrow's head and the central lines of the arrows, [Degree]: 
Arrows > Density	Number of arrows per a streamline of maximal length (if the arrows are placed evenly along the length, **Arrows > By time = No**), or of maximal time duration (if the arrows are placed evenly by time, **Arrows > By time = Yes**). This parameter can be set in the range from **1** to **100**.
Arrows > By time	Possible options are: • **Yes**: the arrows are placed evenly according to the time required to pass along a streamline • **No**: the arrows are placed evenly according to the length of a streamline
Coloring > Variable > ... **Coloring > Value range > ...** **Coloring > Log. scale > ...** **Coloring > Palette > ...**	Coloring the streamlines depending on the value of a variable in a given point of the **Object**. The variable, range and palette are specified similarly to standard parameter groups **Variable**, **Range**, and **Palette**, see General properties of Layers and Parameters for defining a palette.

[*)] This parameter is available only for **Streamlines** that are built in the **Computational space**.
[**)] This parameter is available only for **Streamlines** that are built on a surface.

Element "Emitter for streamlines"

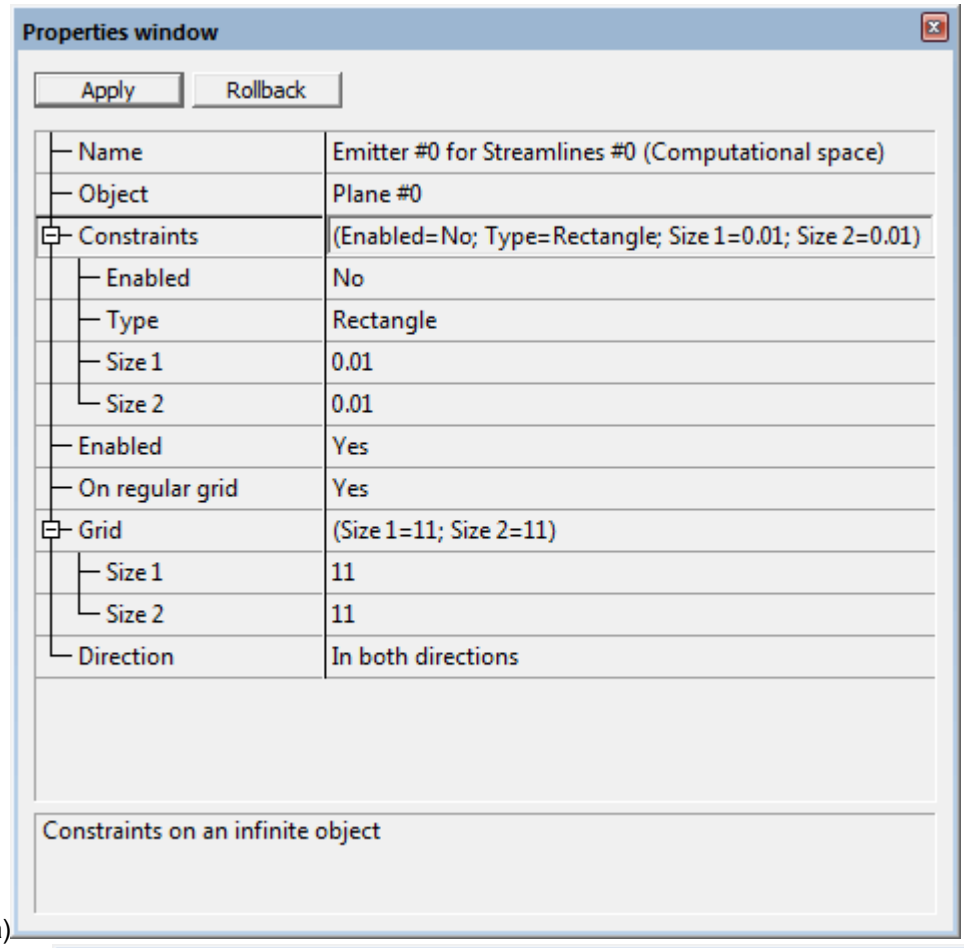
The folder of a **Streamlines** layer has a child element **Emitter**.



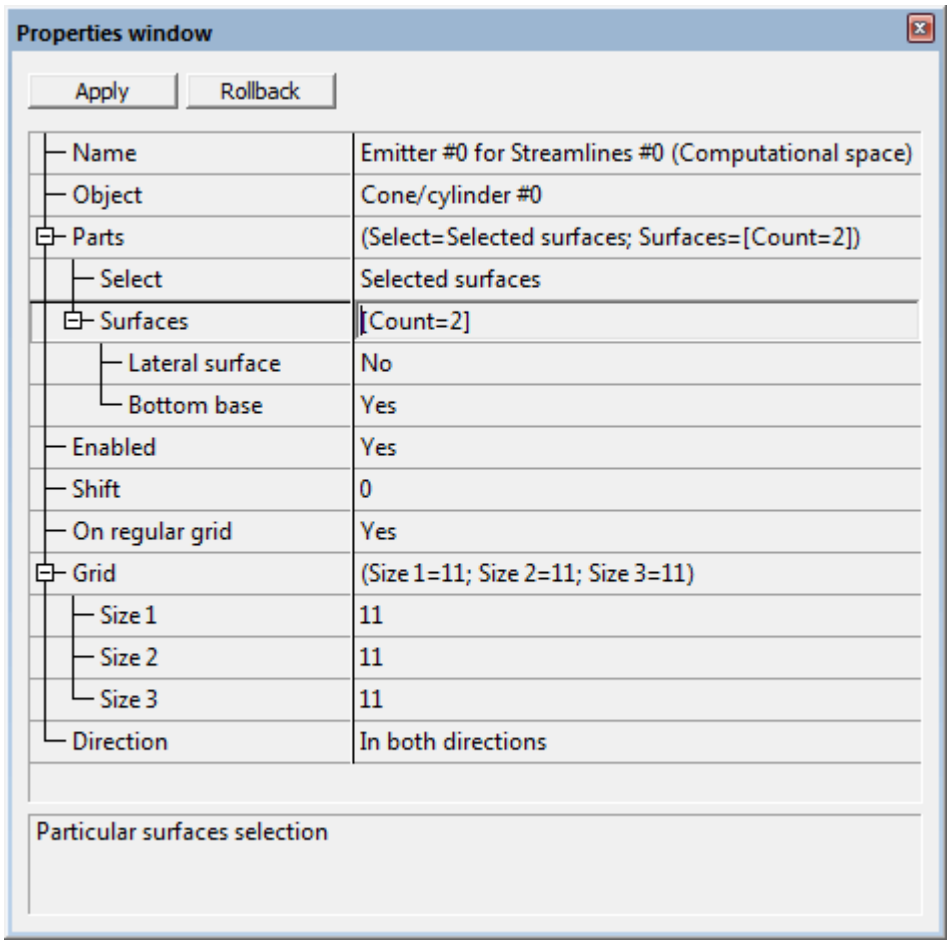
Elements **Emitter for streamlines** in the project tree

Streamlines begin from **Emitters**. An **Emitter** for streamlines locates in/on an **Object** (or part(s) of an **Object**). List of parameters of an **Emitter** depends on the **Object**, in/on which the **Emitter** is set.

a)


Properties window
Apply
Rollback
Name
Emitter #0 for Streamlines #0 (Computational space)
Object
Plane #0
Constraints
(Enabled=No; Type=Rectangle; Size 1=0.01; Size 2=0.01)
Enabled
No
Type
Rectangle
Size 1
0.01
Size 2
0.01
Enabled
Yes
On regular grid
Yes
Grid
(Size 1=11; Size 2=11)
Size 1
11
Size 2
11
Direction
In both directions
Constraints on an infinite object

b)


Properties window
Apply
Rollback
Name
Emitter #0 for Streamlines #0 (Computational space)
Object
Cone/cylinder #0
Parts
(Select=Selected surfaces; Surfaces=[Count=2])
Select
Selected surfaces
Surfaces
[Count=2]
Lateral surface
No
Bottom base
Yes
Enabled
Yes
Shift
0
On regular grid
Yes
Grid
(Size 1=11; Size 2=11; Size 3=11)
Size 1
11
Size 2
11
Size 3
11
Direction
In both directions
Particular surfaces selection

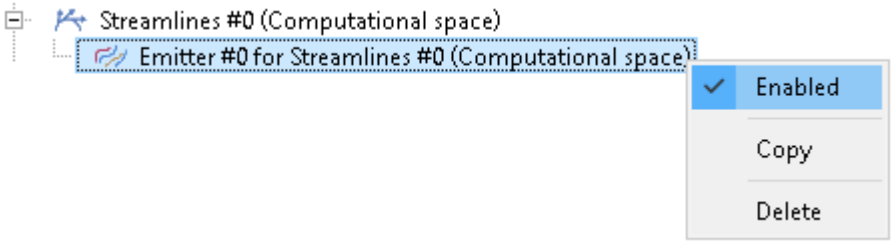
The **Properties** window of the element **Emitter for streamlines** (a: streamlines start from a rectangle in the plane; b: from the base of a cone)

Parameters of the element **Emitter for streamlines**:

Parameter	Description
Name	Name of the **Emitter** of the streamlines (this parameter allows you to change the standard name **Emitter #N for (*name of the layer*)** to another name).
Object	The object, on which **Emitter for Streamlines** is placed. For Streamlines that are set on a surface of a **Supergroup** you can select an **Object**, on which **Emitters** are placed, as: • the **Supergroup** itself, on which the **Layer** is built. • some **Plane**. In this case the **Emitters** will locate on the contour, which is formed as a section of the **Supergroup** by the specified **Plane** (see illustration *"Emitters of streamlines locate on the section of a conic Supergroup by a Plane"* below). In this case number of points in the **Emitter** is set by the **Grid > Size 1** parameter in properties of the **Emitter**. For Streamlines that are set on a **Plane** you can select an **Object**, on which **Emitters** are placed, as: • the same **Plane**, on which the **Layer** is built. • another **Plane**. In this case the streamlines will start from points on the intersection line of these **Planes** and their amount is set by the **Grid > Size 1** parameter in properties of the **Emitter**.
Constraints > ...	These parameters allow you to select an Emitter for Streamlines on a square or rectangular fragment of a **Plane**. (See the illustration) See General properties of Layers.
Parts > ...*)	These parameters allow you to specify an Emitter for Streamlines on individual parts (volumes, surfaces, edges) of geometric objects that have finite volume (see the illustration). See General properties of Layers.
Enabled	Possible options are: • **No**: the Emitter is disabled (it is set off) • **Yes**: the Emitter is enabled (it is set on)
Shift*)	The shift of the emitters of the streamlines in the direction towards or against the normal to the surface of the **Object**, on which the emitters are placed. This is the distance to which the emitters of the streamlines are shifted from the **Object**'s surface in the direction of or against the normal. The normal's direction can be different on different **Objects**. If you need to change the direction of the shift, reverse the sign of the numerical value of this parameter. This parameter is only available when Emitter for Streamlines is set on the surface of the **Object** (the surface is specified in the **Part** group). (See the illustration)
On regular grid*) **Grid > ...**	See General properties of Layers.
Direction	Possible options are: • **Forward** - streamlines are built along the stream • **Backward** - streamlines are built against the stream • **In both directions** - the streamlines are built forward and back in both directions (See the illustration)

*) This parameter is available only for **Streamlines** that are built in the **Computational space**.

Context menu of the emitters for streamlines in the project tree



Context menu of the **Emitter for streamlines** element in the project tree

Context menu of the **Emitter for streamlines** item in the project tree:

Menu item	Description
Enabled	- the **Emitter** is enabled - the **Emitter** is disabled
Copy	Creating an element, which is a copy of the selected element
Delete	Deleting the selected element from the project tree

Window «Info» of the «Streamlines» layer

The data displayed in the **Info** window of the **Streamlines** layer:

Parameter	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current simulated time
Max. distance	Maximal length of the streamline
Max. duration	Maximal time required for a particle to pass a whole streamline
Variable	Information about the vector variable, on which the streamlines are built
Aux. variable	Information about the variable, which is used for coloring the streamlines
Palette	The palette, which is used for coloring the streamlines (according to values of the additional variable)

See section Folder «Layers» for details about parameters of the **Info** window for **Layers**.

Specifics of streamlines on a surface

When **Streamlines** are built on a surface of a **Supergroup** or an **Imported object** (on which a **Moving body** is set), only tangential (relating this surface) component of the visualized vector variable is used.

Streamlines that are built on a surface have the following specifics:

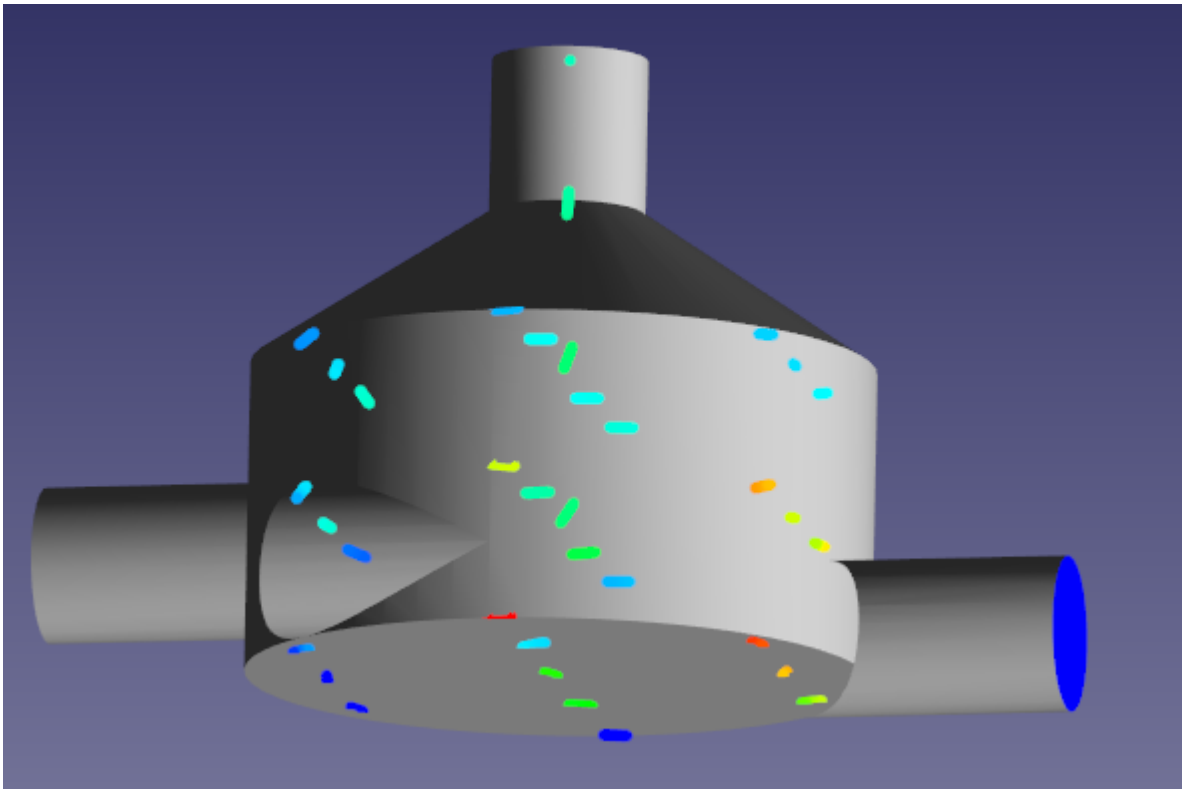
- such streamlines terminate when they go away or substantially deviate form the surface
- the **Succession > Breakaway angle** parameter sets the threshold minimal angle between the **Streamline** and the normal to the surface, on which the **Streamlines** are built, [Degree]. If this angle becomes below this threshold value, the streamline terminates.
- a streamline also terminates when the tangential component of the vector exceeds the value defined by either the **Succession > Abs. length** or the **Succession >Rel. length** parameter. Selection of the method of such limitation is set by the value of the **Succession > Projection control** parameter.
- a streamline also terminates if it comes on a facet that doesn't belong to the surface, on which the **Streamlines** have been built.
- in some situations streamlines could terminate because of inaccuracies of computations when the program works with a difficult geometry

- when **Emitters** are placed on a section of the surface by some **Plane**, the **On regular grid** parameter is absent while the **Emitters** are placed uniformly along the contour of the section in the amount, which is set by the **Grid > Size 1** parameter, while parameters **Grid > Size 2** and **Grid > Size 3** are absent.

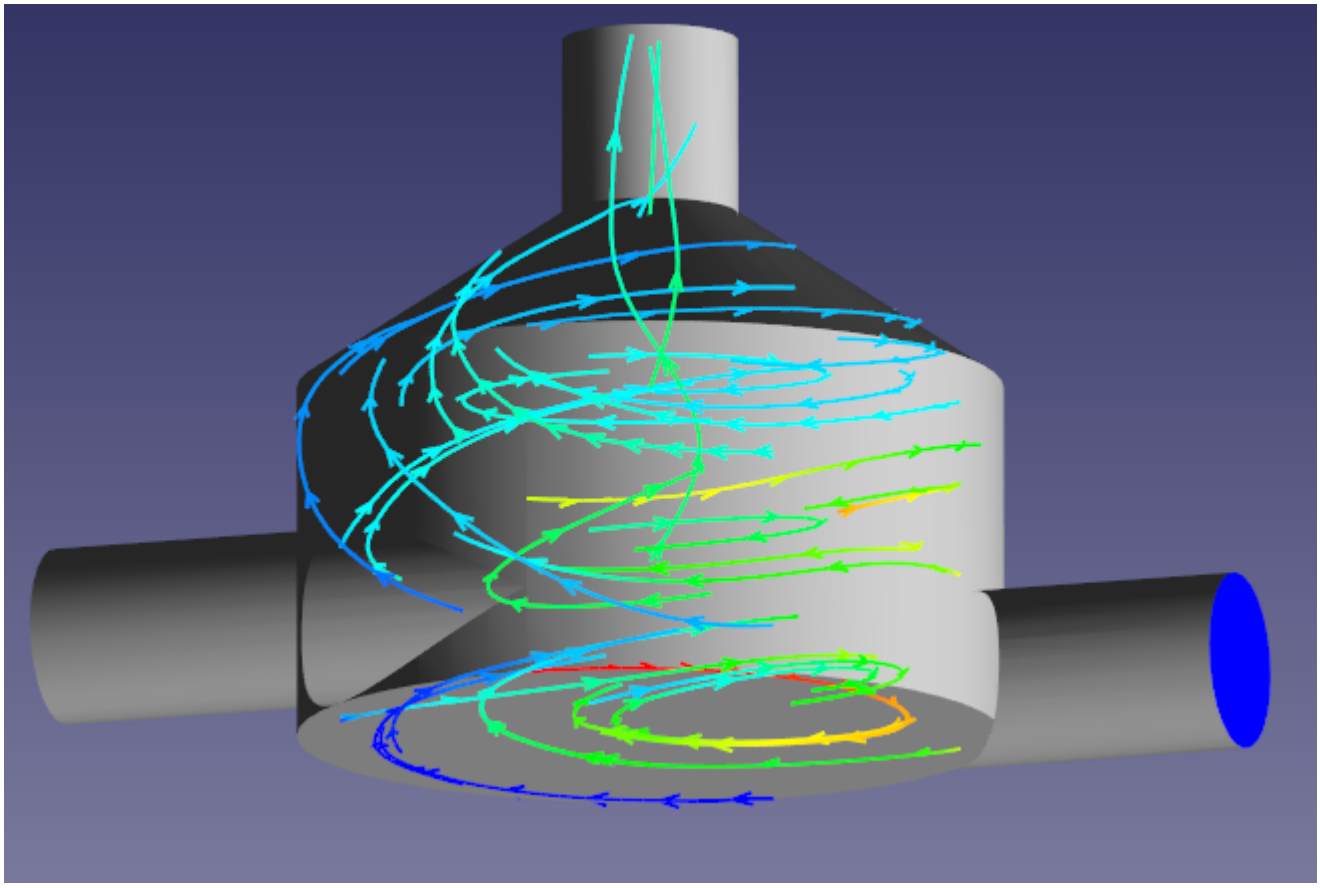
When **Streamlines** are built on a surface that separates **Subregions**, it is strongly recommended to specify the required **Subregion** explicitly, because otherwise **Emitters** could be placed into a wrong **Subregion**.

If some emitters, when they are allocated over the section's contour, are placed into a **Subregion**, in which the visualized vector variable is absent (and so building **Streamlines** over this variable is impossible), then **Streamlines** are not build from such emitters and nevertheless the program doesn't reallocate such emitters.

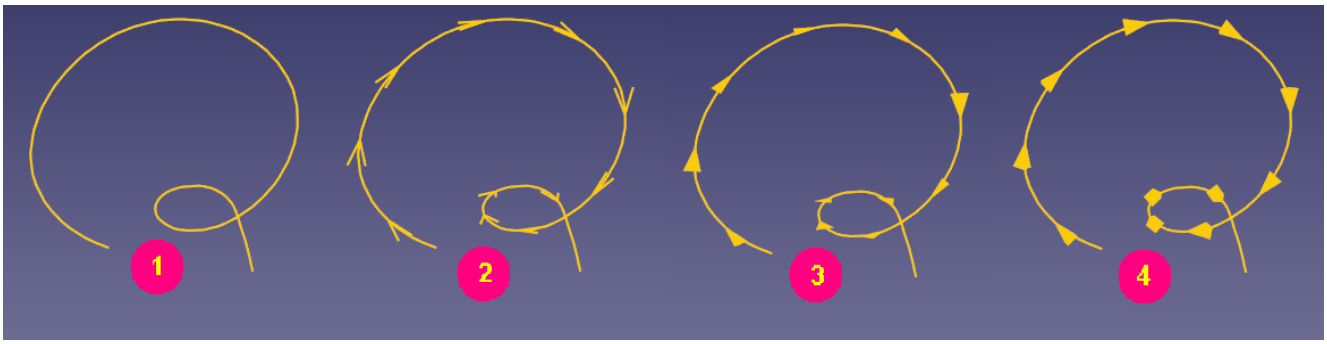
Illustrations



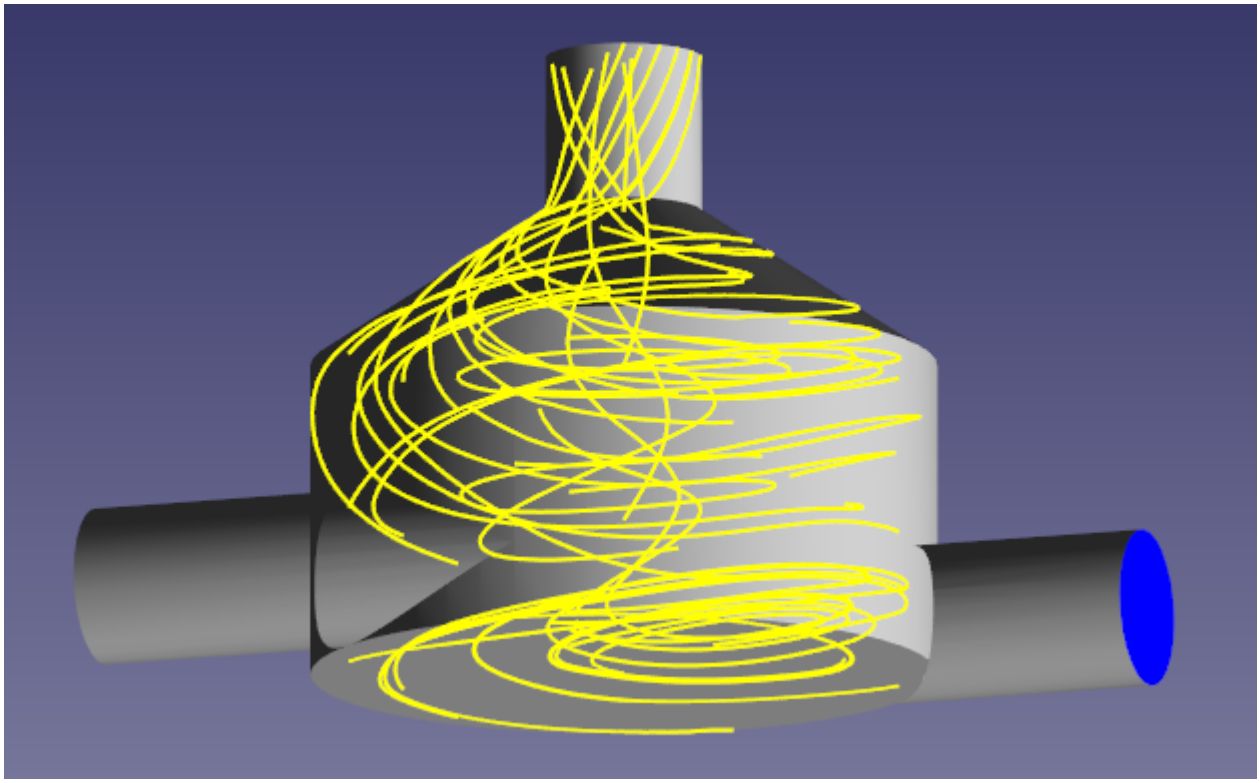
If you specify very short length of the streamlines (by the **Succession > Abs. length** parameter), you will see starting points of the streamlines



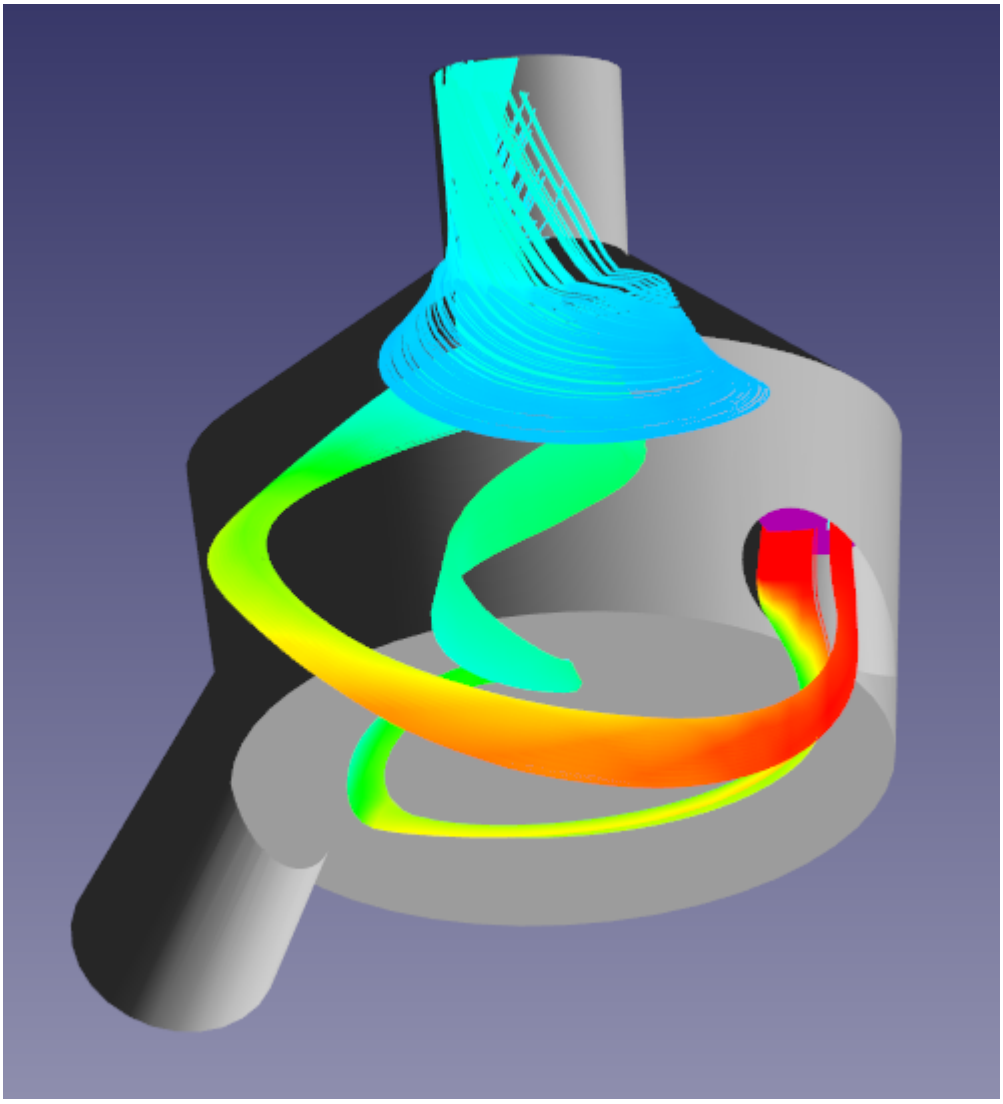
Images arrows can be added to the streamlines. These arrows show direction of the movement (shape and size of the arrows are set in the **Arrows** group of parameters)



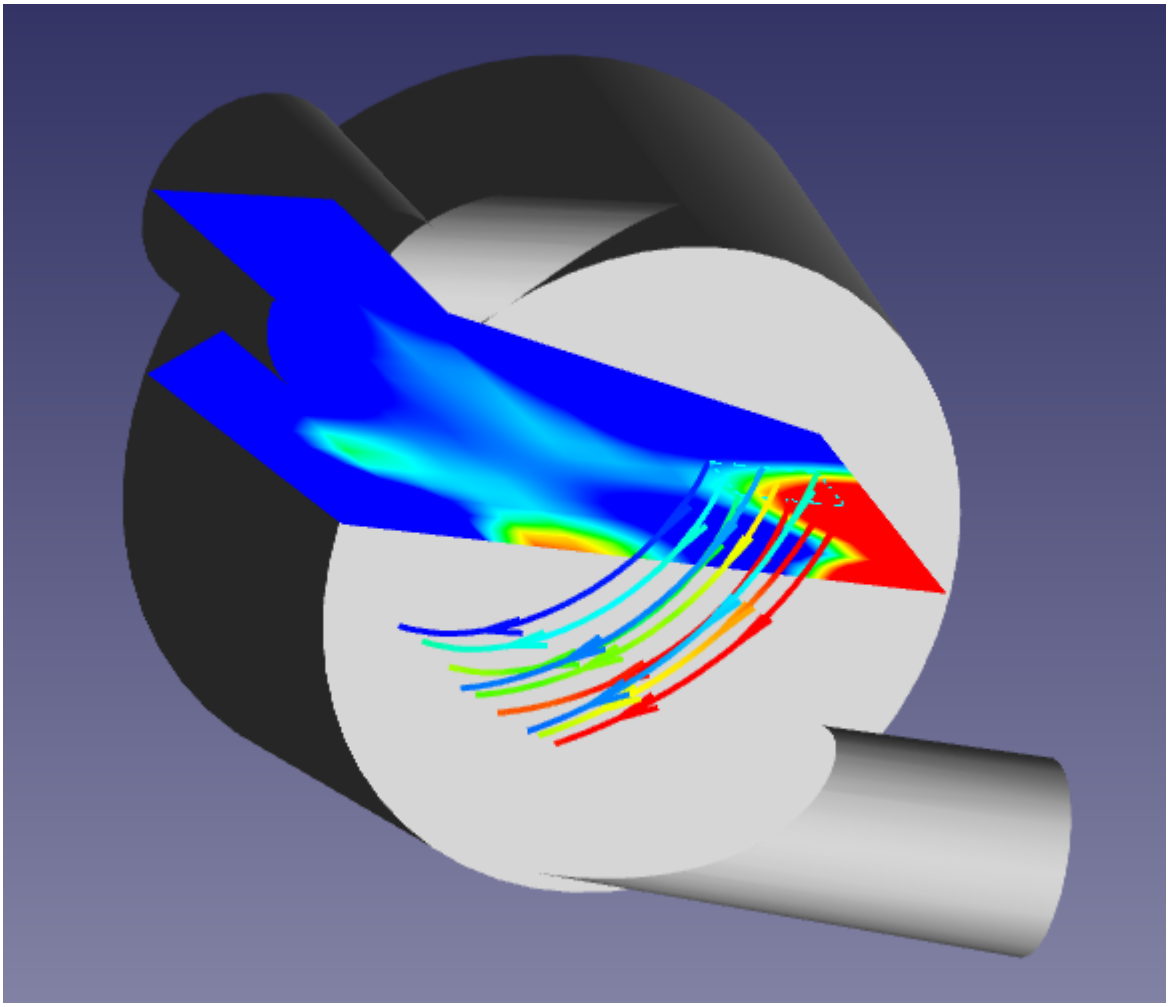
Appearance of arrows on **Streamlines** is set by the **Arrows > Kind** parameter (1: **No arrows**; 2: **Lines**; 3: **Triangles**; 4: **Pyramid**)



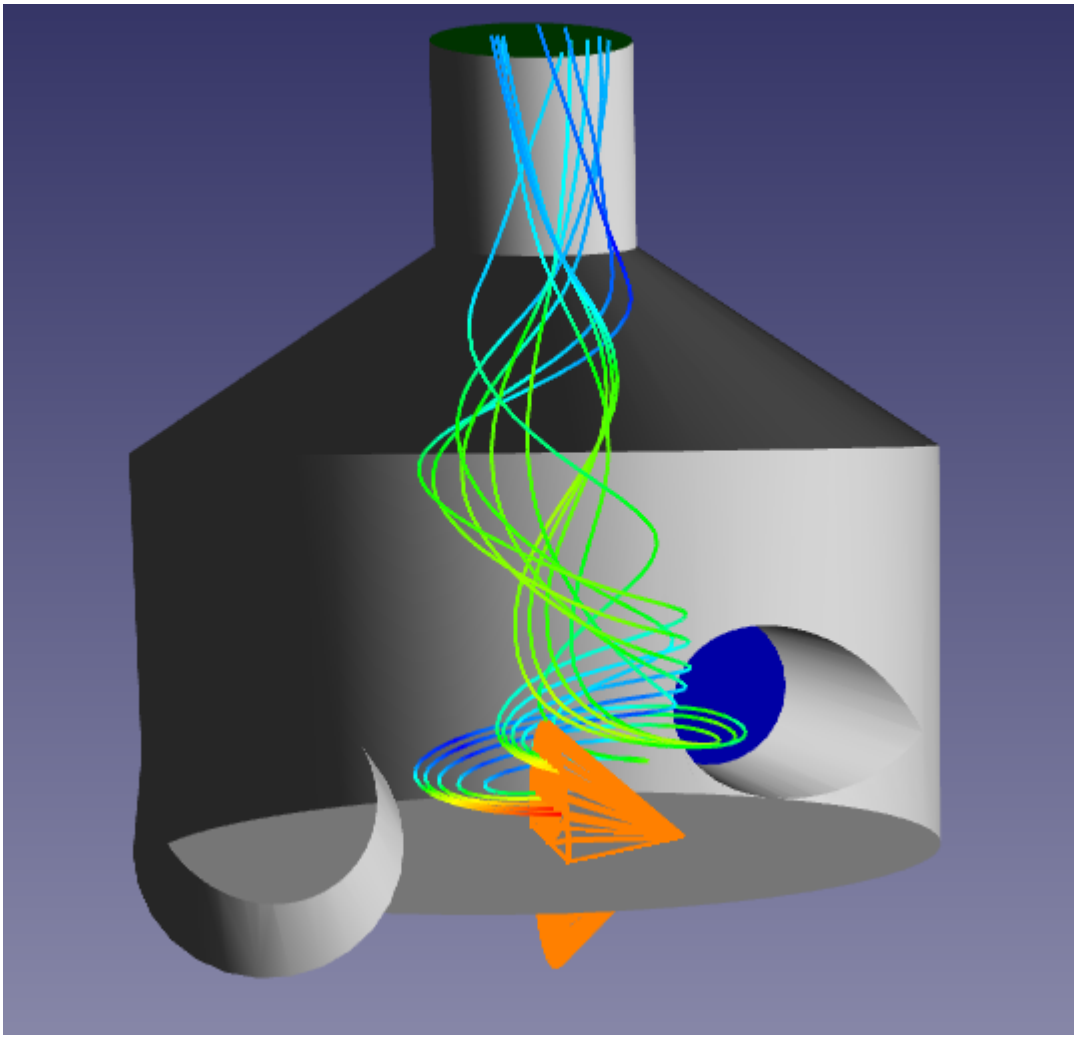
If a coloring variable is not specified for the streamlines, the color of the streamlines is set by the **Lines > Color** parameter



Here **Streamlines** flow away from *two* **Emitters** (each of the emitters generates a single beam of **Streamlines**)



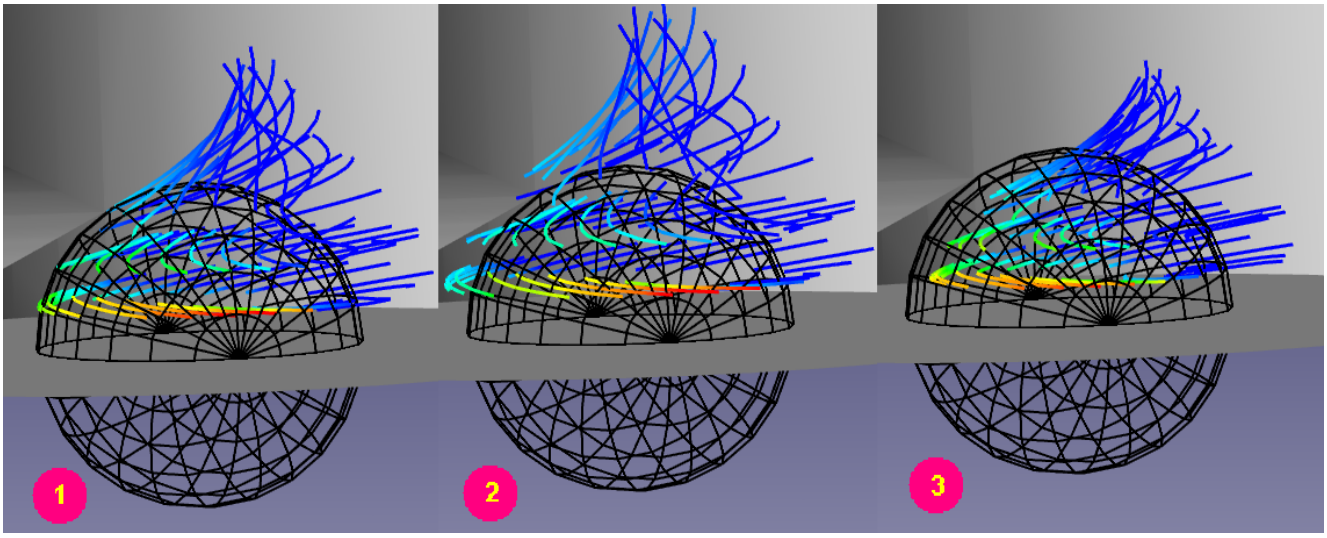
You can limit the emitting the **Streamlines** by a given rectangle or square in a **Plane** (this is set by parameters in the **Constraints** group)



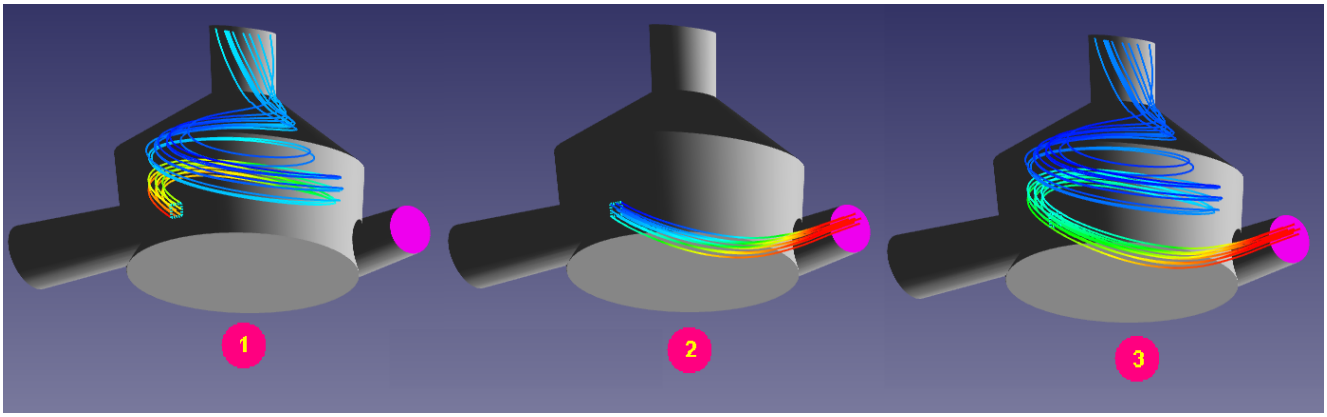
You can also specify emitting **Streamlines** only from certain surfaces of an **Object** (in this example, from the base of a cone).

In the example on the illustration above the following parameters are set:

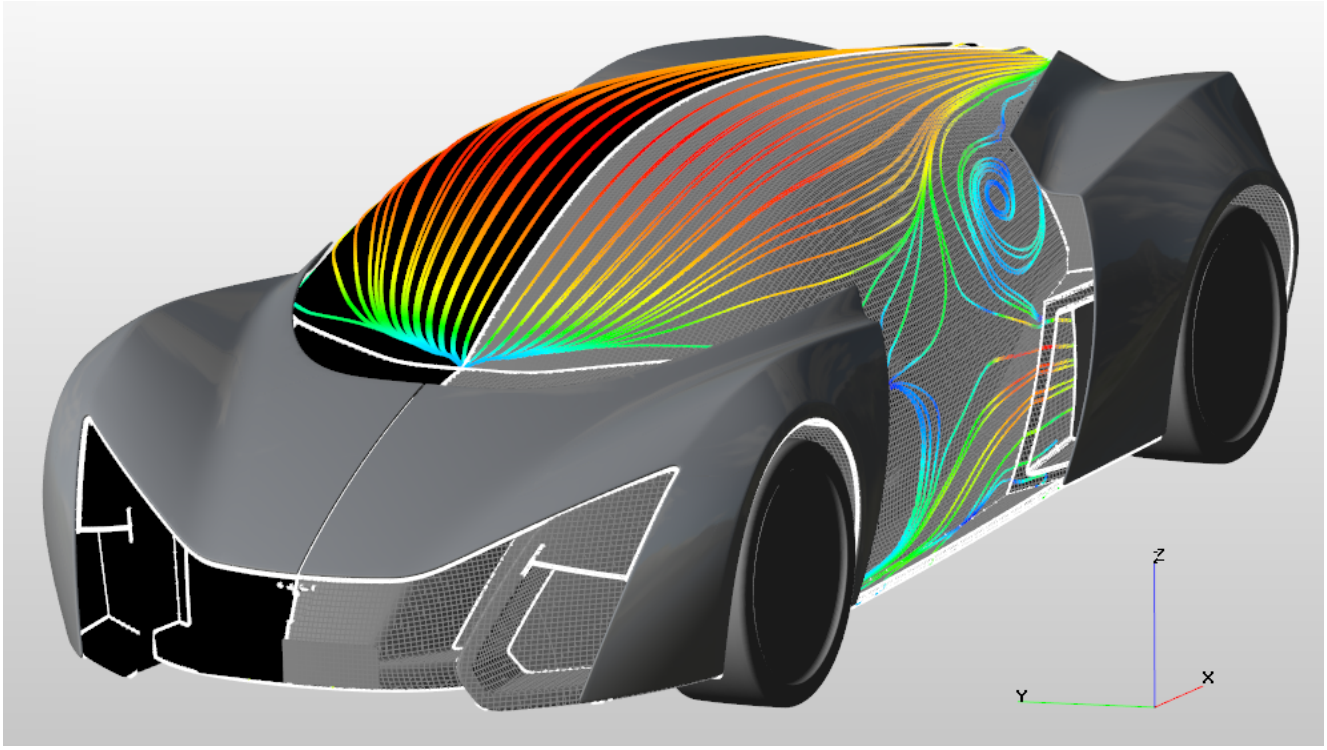
- **Object = Cone/cylinder #N**
- **Parts > Select = Selected surfaces**
- **Parts > Select > Surfaces > Lateral surface = No**
- **Parts > Select > Surfaces > Bottom base = Yes**



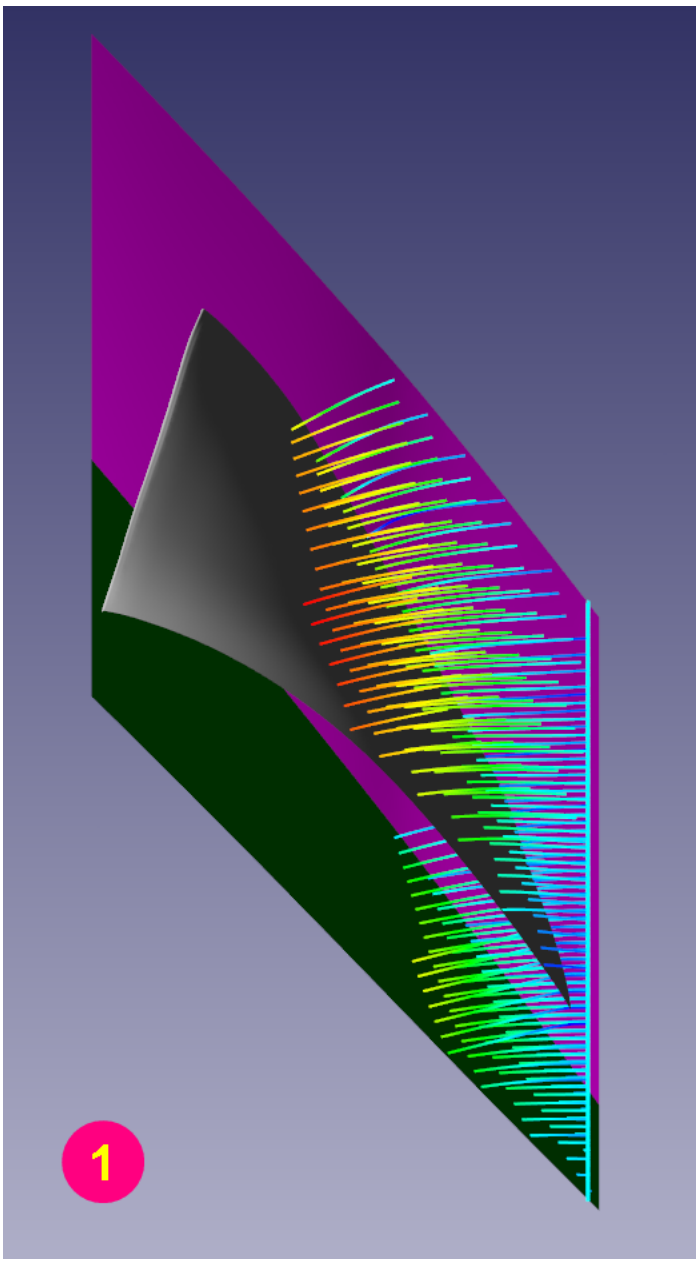
Shift of initial points of the streamlines relatively to the surface, on which the **Emitter** is built
(1: no shift is applied; 2: and 3: shifts is used in different directions relatively to the surface of the **Object**)

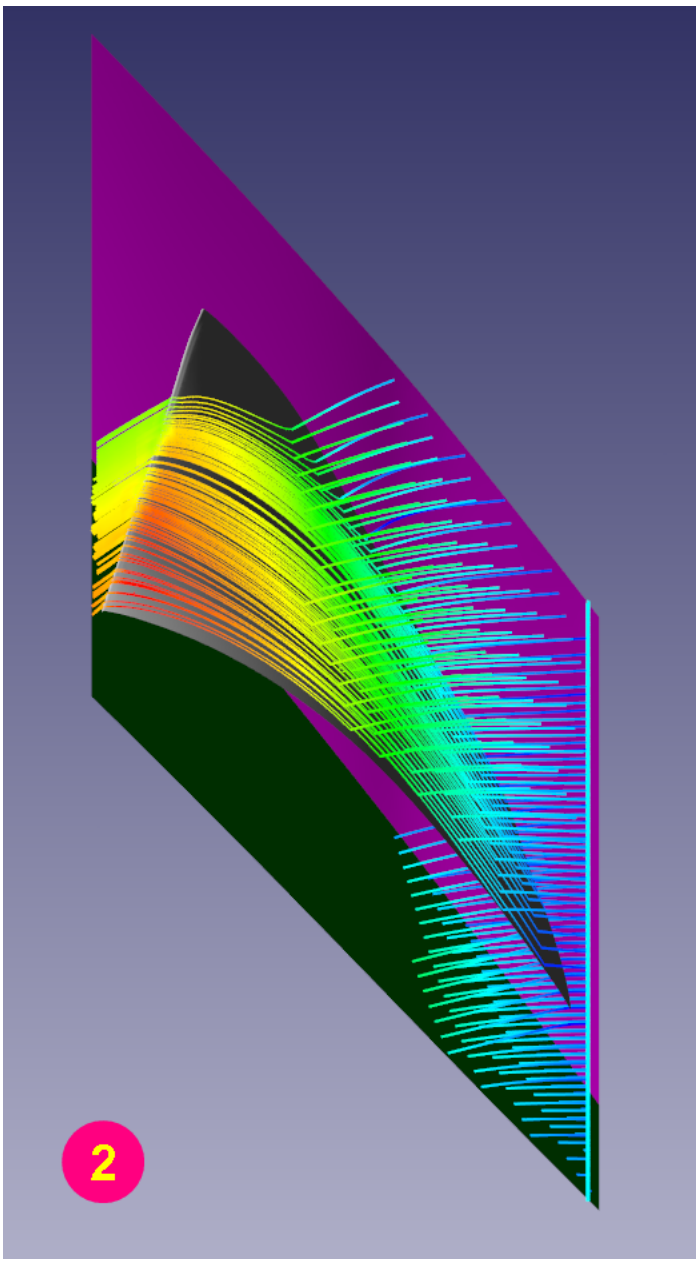


The **Direction** parameter specifies the direction in which the **Streamlines** go from the **Emitter**
(1: **Forward**, 2: **Backward**, 3: **In both directions**)

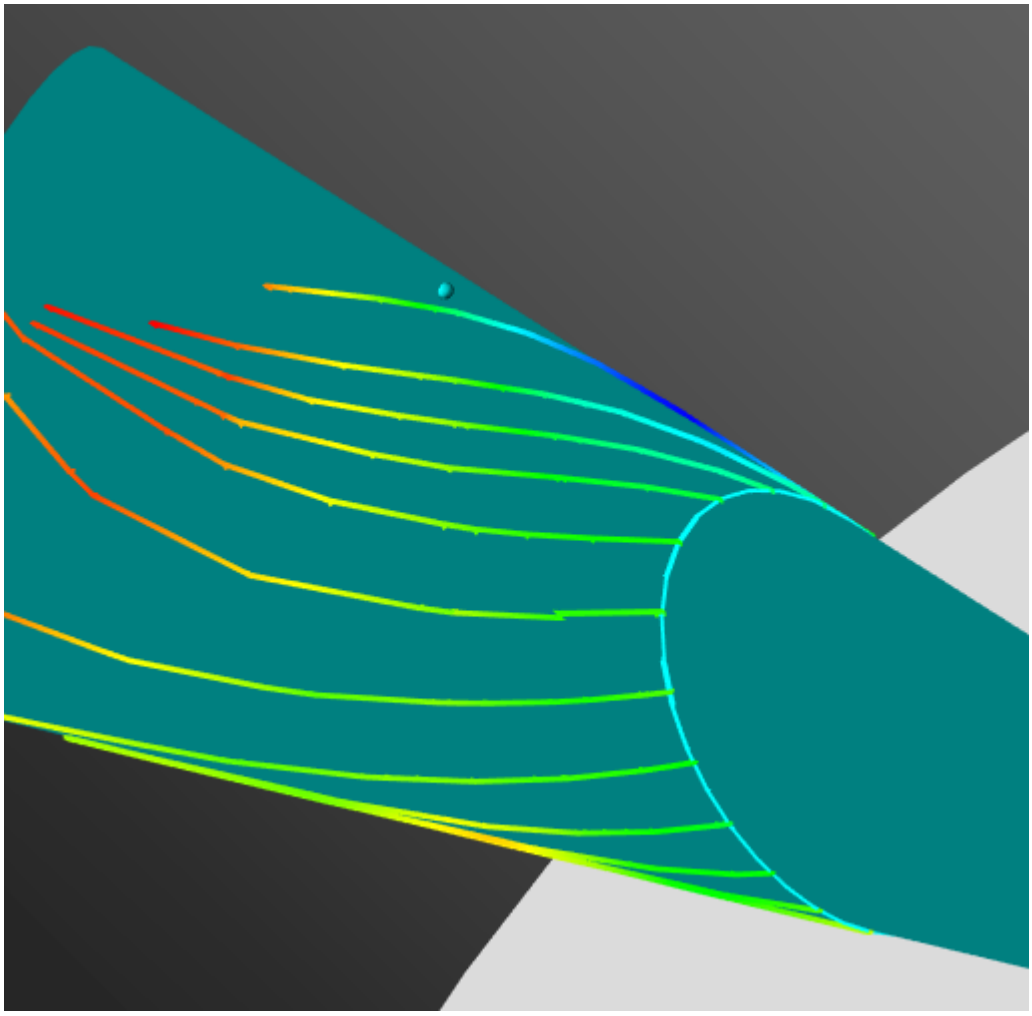


Streamlines that are set on a surface go along the surface and don't go away into the space

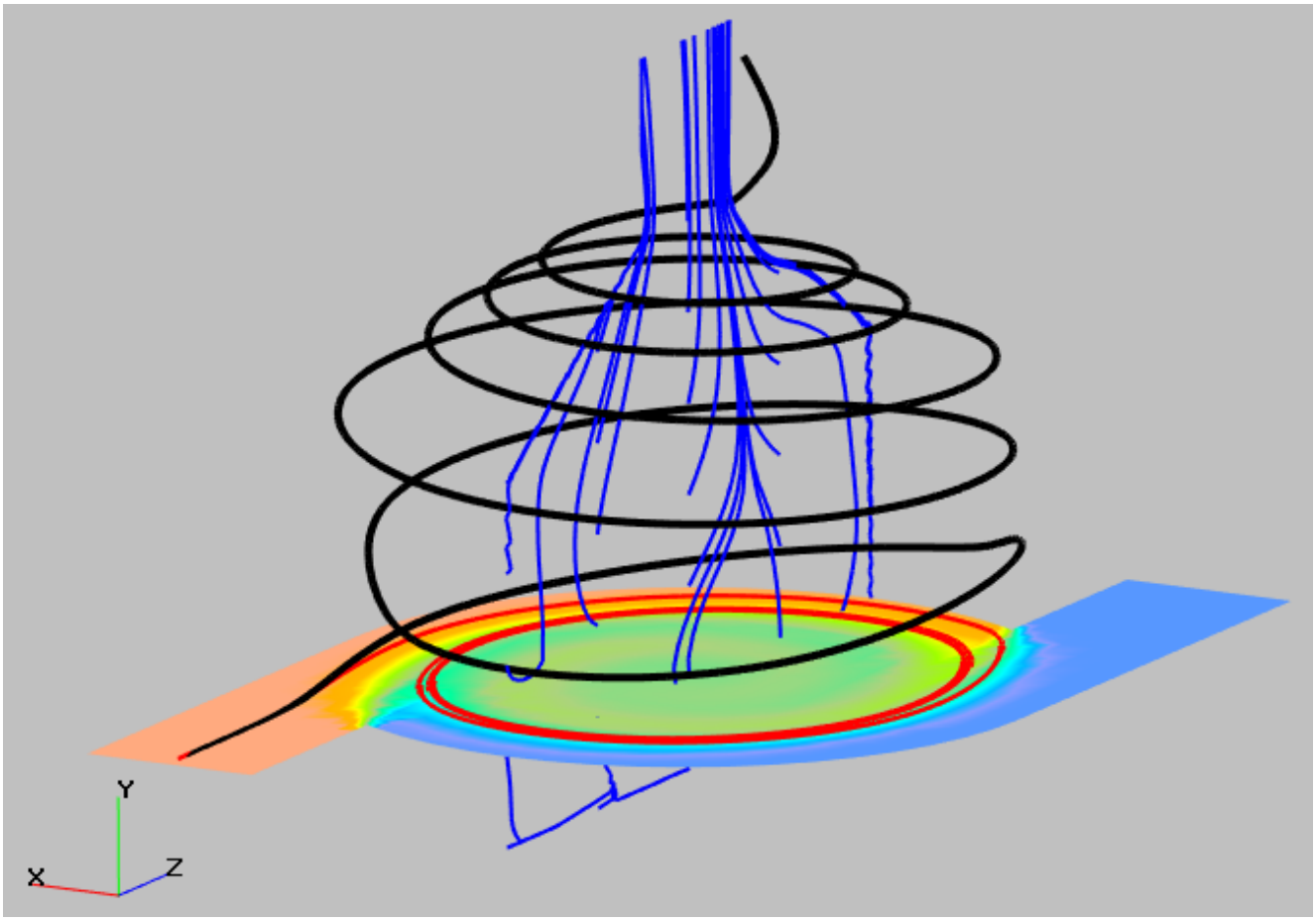

1



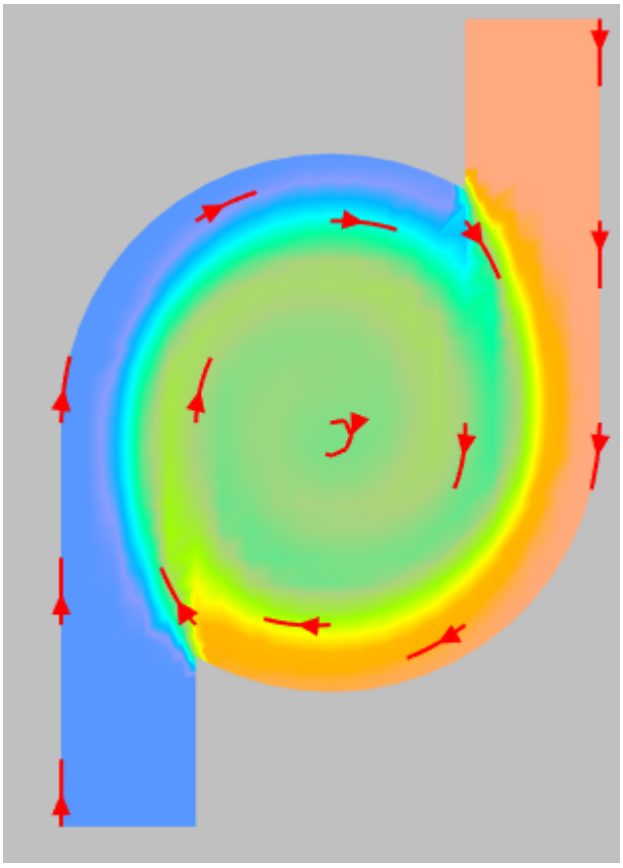
Work of the parameter **Succession > Reflect from walls**. Here the program simulates (in a sector setting) flow of the air through a compressor's rotor. Streamlines, which are built in the **Computational space** by velocity of the air flow, hit a compressor's blade. **1: Reflect from walls = No** (the streamlines terminate when they hit the compressor's blade), **2: Reflect from walls = Yes** (the streamlines continue their trajectories over the blade's surface and then go to the **Computational space** after breakaway from the blade).



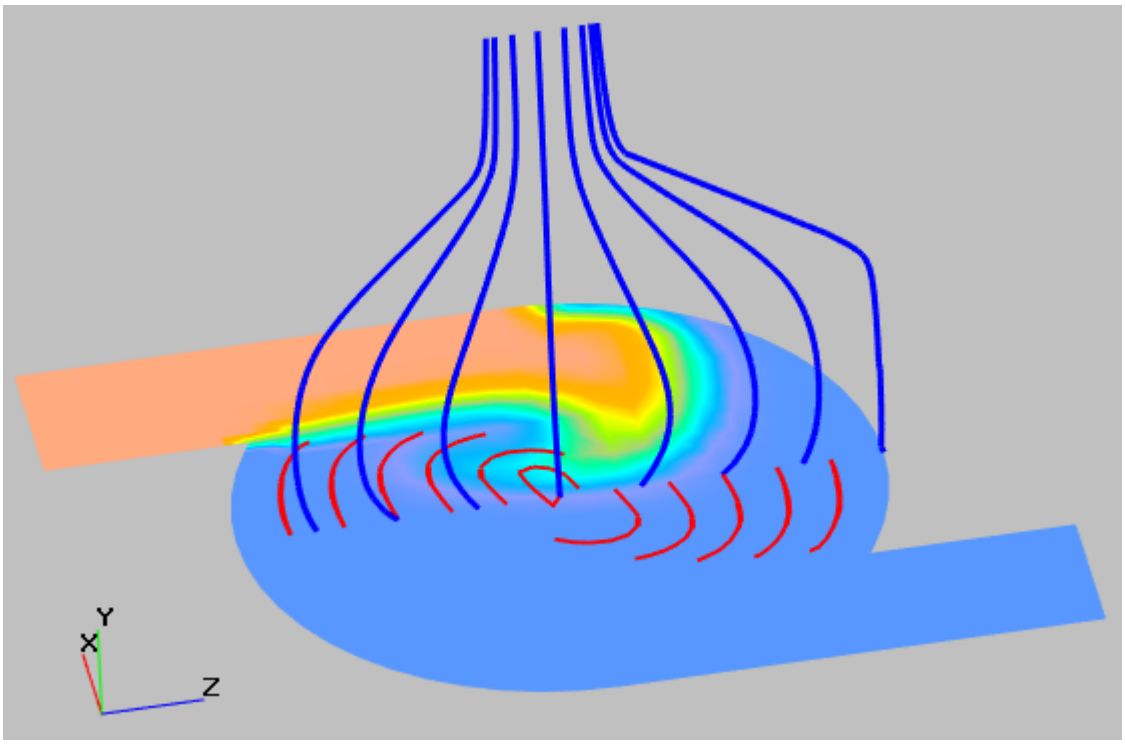
Emitters of **Streamlines** that have been built on a conic surface of an **Imported object** starting from a section of this surface by a **Plane**.
In the parameters of the **Streamlines** it is set **Object = Imported object #0** and in the parameters of the **Emitter** it is set **Object = Plane #0**.



Example of **Streamlines** on two **Planes** (red and blue lines) and in the **Computational space** (black lines) within the mixer



In this case the emitter of **Streamlines**, which are set on a **Plane**, locates on the same **Plane**



In the case the emitter of **Streamlines**, which are set on a **Plane**, locates on another **Plane**. The **Streamlines** start from points on the intersection line of these **Planes**.

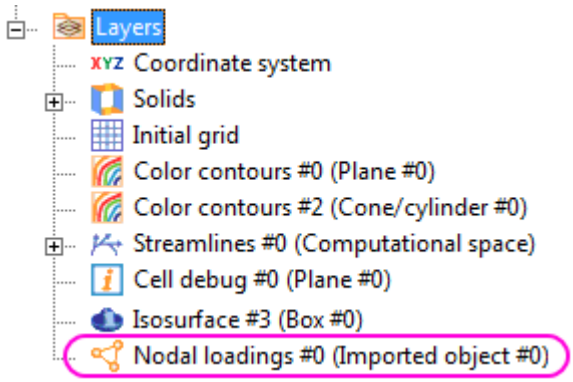
On this **illustration** there are two **Streamlines** layers (red and blue lines) built on intersecting **Planes**, and their emitters locate on the other **Plane** (not the one, on which the layer is built).

Also a **Color contours** layer is built on the horizontal **Plane** displaying temperature of water in the mixer.

8.1.8.5.8.15 Layer «Nodal loadings», user interface

The **Nodal loadings** layer is used to display loads in the nodes of the computational grid (which has been built in *FlowVision*) on an **Imported object**, on the surface on which the connection with *Abaqus* is done.

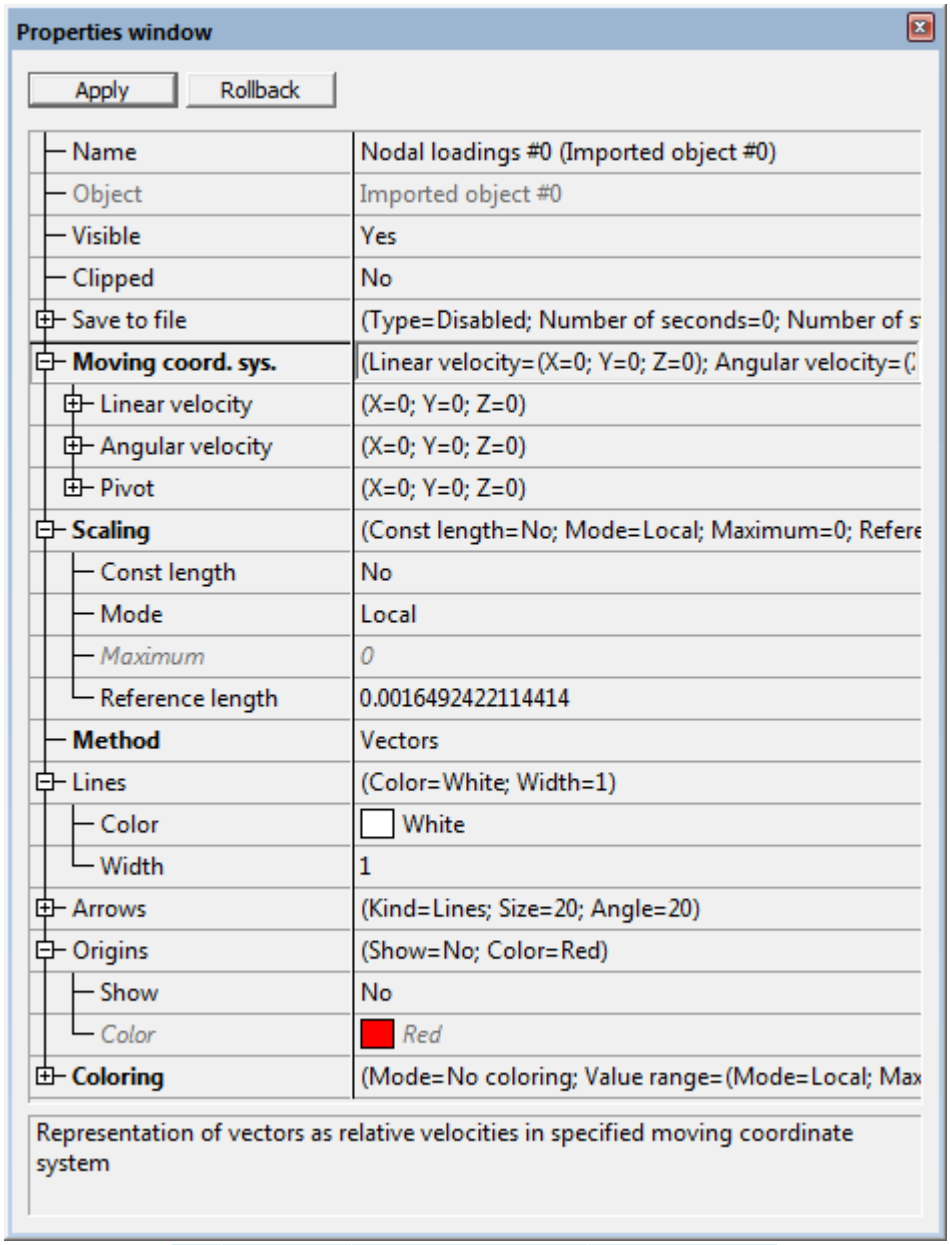
The **Nodal loadings** layer is used to display vectors of loads at the nodes of a geometric grid on the surface of the imported object. These layers are computed only at a joint computation of the flow (*FlowVision*) and stress-strain analysis of the construction (*Abaqus*).



The **Nodal loadings** layer in the project tree

See also: section Layer «Nodal loadings».

Parameters of the «Nodal loadings» layer



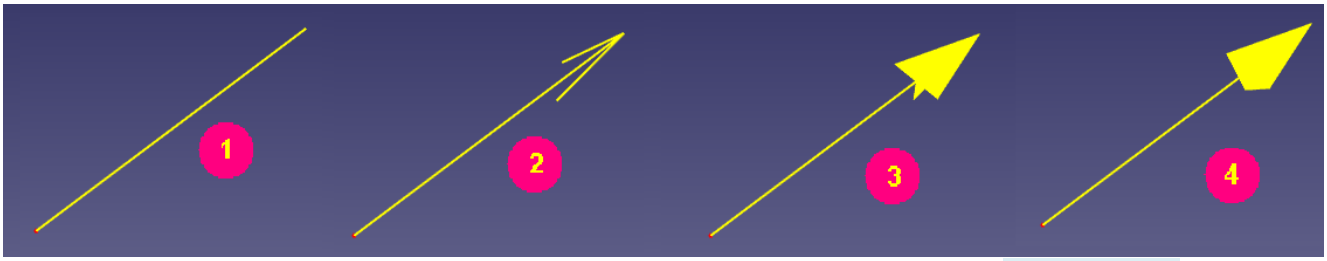
The **Properties** window of the layer **Nodal loadings**

Layer parameters **Nodal loadings**:

Parameter	Description
Name	Layer name (this option allows you to change the default name "**Nodal loadings #N (Object)**", formed from the name of the layer type, numbers and the **Object**, on which the layer is built).
Object	See General properties of Layers.
Visible	
Clipped	
Save to file > ...	
Moving coord. sys. > ...	Parameters of the representation of vectors as relative velocities in a specified moving coordinate system
Moving coord. sys. > Linear velocity > X	Components of the vector of linear velocity of the moving coordinate system.

Parameter	Description
Moving coord. sys. > Linear velocity > Y	
Moving coord. sys. > Linear velocity > Z	
Moving coord. sys. > Angular velocity > X	Components of the vector of angular velocity of the moving coordinate system.
Moving coord. sys. > Angular velocity > Y	
Moving coord. sys. > Angular velocity > Z	
Moving coord. sys. > Pivot > X	Coordinates of the pivot point in the moving coordinate system.
Moving coord. sys. > Pivot > Y	
Moving coord. sys. > Pivot > Z	
Scaling > ...	Group of parameter, which specify the length of the vectors.
Scaling > Const length	The program will show only direction of the vectors. Possible options are: • **Yes**: length of the vectors is constant and is calculated by the formula $L = L_{ref}$ • **No**: length of the vectors is proportional to the value of the variable and is calculated by the formula $L = \frac{\lvert V_{cur} \rvert}{\lvert V_{max} \rvert} \cdot L_{ref}$, where: $\lvert V_{cur} \rvert$ is current value of the modulus of the vector quantity $\lvert V_{max} \rvert$ is the maximal value of the modulus of the vector quantity, determined in accordance with the **Scaling > Mode** parameter (see below) L_{ref} is the reference length
Scaling > Mode	The choice of method for determining $\lvert V_{max} \rvert$. Possible options are: • **Local**- is automatically determined by the values of the variable at the **Object** • **Global**- is automatically determined by the values of the variable in the whole phase • **Manual** - is set by parameter **Maximum**
Scaling > Maximum	The value of $\lvert V_{max} \rvert$ calculated for **Mode=Local** or **Mode=Global** or set when **Mode=Manual**.
Scaling > Reference length	The value of L_{ref}
Lines > Color	Line color and width image vectors.
Line > Width	
Arrows > ...	Parameters for setting the image of arrows at the ends of the image vectors. (see illustration)
Arrows > Kind	Select the type of pointers display: • **No arrows** - arrows are not displayed • **Lines** - the arrows are displayed as lines • **Triangles** - arrows are displayed as triangles • **Pyramid** - the arrow is displayed in the form of pyramids
Arrows > Size	Size of arrows (set as a percentage of the length of the vector)

Parameter	Description
Arrows > Angle	The angle between the lines and the direction of the vector arrows (given in degrees)
Origins > Show	Displaying points points, indicating the beginning of the vectors. Possible options are: **No | Yes**.
Origins > Color	Color dots that indicate the beginning of vectors.
Coloring > ...	This group of parameters specifies coloring of images of vectors depending on values of temperatures or heat fluxes.
Coloring > Mode	The mode of coloring the vectors. Possible options are: • **No coloring** • **Length** • **Temperatures** • **Heat fluxes**
Coloring > Value range> ... **Coloring > Palette> ...**	The variable's range and palette settings are specified similarly to the standard parameter groups **Value range** and **Palette**. See General properties of Layers and Parameters for defining a palette.



The appearance of the arrows in the images set with the parameter vectors **Arrows > Kind**
(1 - **No arrows**, 2 - **Lines**, 3 -**Triangles**, 4 - **Pyramid**)

Window «Info» of the «Nodal loadings» layer

The data displayed in the **Info** window of the **Nodal loadings** layer:

Parameter	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current simulated time
Load	
The local maximum	The maximum load value on the object
The local minimum	The minimum value of the load at the facility
Extras. variable	Information about the variable, which is held on the coloring of the vectors
Palette	Palette, used for coloring vectors

Details about the parameters of the **Info** window for **Layers** see in the section Folder «Layers».

Components of the text file for the layer «Nodal loadings»

The header file describes the format of data presentation.

For each output file is written in general information on time step and N lines, the number of which corresponds to the number of nodes of the imported object.

Each line represents a single node.

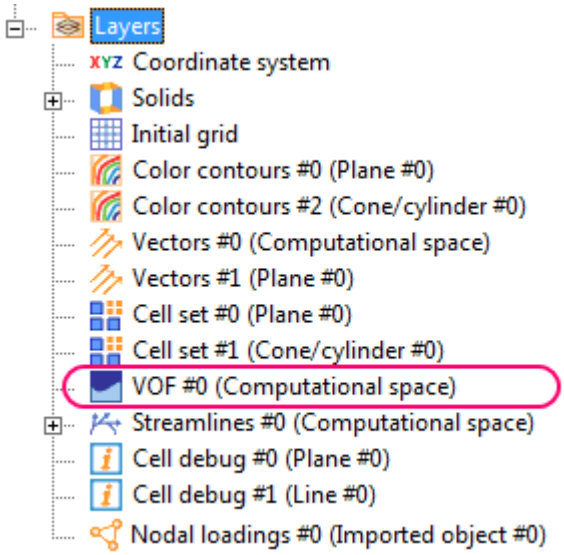
General information:

Step	Step number
Time	Time
NumNodes	Number of nodes

Line number i is as follows:

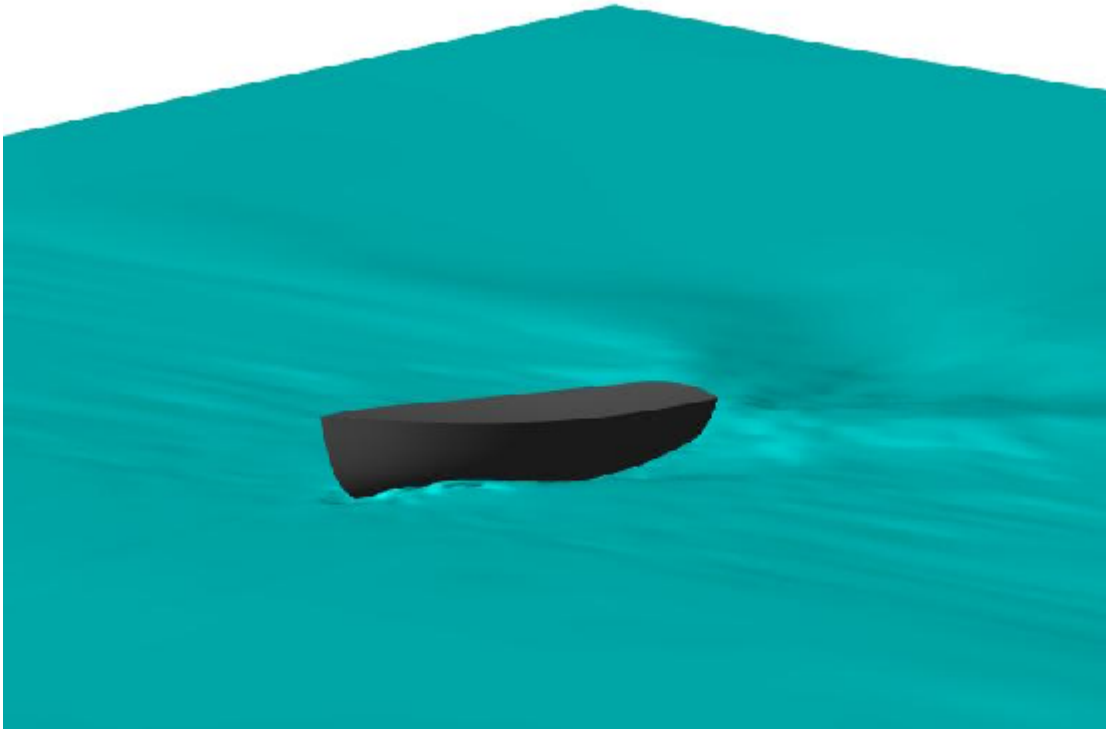
point ix	X-coordinate of the i-th node
point iy	Y-coordinate of the i-th node
point iz	Z-coordinate of the i-th node
stress ix	X-component of load on the i-th node
stress iy	Y-component of load on the i-th node
stress iz	Z-component of load on the i-th node
stress i.modules	Module load
temperature i	The temperature on the i-th node
heatflux i	Heat flux in the i-node

8.1.8.5.8.16 Layer «VOF», user interface



The **VOF** layer in the project tree

The **VOF** layer to be displayed in the space of the computational domain of the interface, calculated by VoF.



The *Volume of Fluid (VoF)* method can *not* be applied in the following cases:

- When subregions are conjugated by velocities, if the inter-phase surface intersects the boundary of the conjugation
- To simulate **Dispersed Phases**
- When using sliding surfaces, if the inter-phase surface intersects the sliding surface
- To simulate interaction of two **Continuous Phases** if no **Motion** physical process is specified in both these **Phases**.

Parameters of the «VOF» layer

Properties window

Apply | Rollback

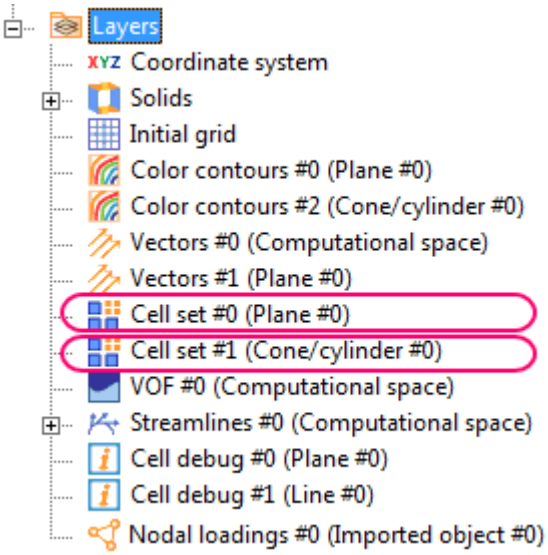
Name	VOF #0 (Computational space)
Object	Computational space
Visible	Yes
Clipped	No
Lighting	Yes
Update	(Type=Automatic; Number of seconds=0; Number of steps=
Value	0.5
Smoothing	Yes
VOF-particles	Yes
Phase	Phase #0
Volume	Yes
Appearance	(Mode=Fill; Fill=(Color=7F3F00; Opaqueness=100))
Mode	Fill
Fill	(Color=7F3F00; Opaqueness=100)
Color	Custom...
Opaqueness	100

The **Properties** window of the **VOF** layer

Layer parameters **VOF**:

Parameter	Description
Name	Name of the layer name (this option allows you to change the layer's default name "**VOF #N (Computational space)**", formed from the name of the layer type, number, and the **Object** on which the layer is built).
Object **Visible** **Clipped** **Lighting** **Update> ...**	See General properties of Layers.
Value	The concentration value is a fraction which is constructed by the interface.
Smoothing	Turn on/off smoothing the surface. Possible options are: • **No** - smoothing the surface of the disabled • **Yes** - included smoothing the surface
VOF-particles	Enables/disables displaying VOF-particles (droplets and bubbles) of one phase in another phase. Possible options are: **No | Yes**.
Phase	The phase boundary is displayed. Possible options are: • **(Vacuum)** - automatic selection of the vacuum phase • **Phase #N** - building a border of said phase
Volume	Possible options are: • **No** - displaying boundary phase with the other phases, with the estimated boundaries subregion (including the movable bodies),and boundary conditions. • **Yes** - displaying only the border of a **Phase** and other **Phase(s)** Available only if **Smoothing = Yes**.
Appearance> ...	See General properties of Layers.

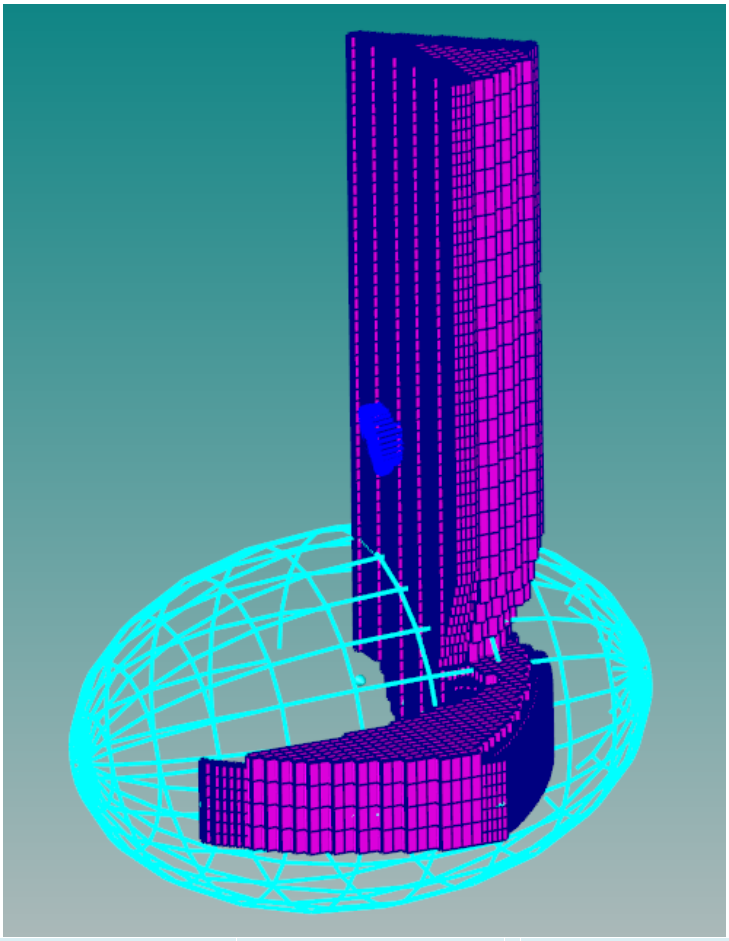
8.1.8.5.8.17 Layer «Cell set», user interface



The «Cell set» layer in the project tree

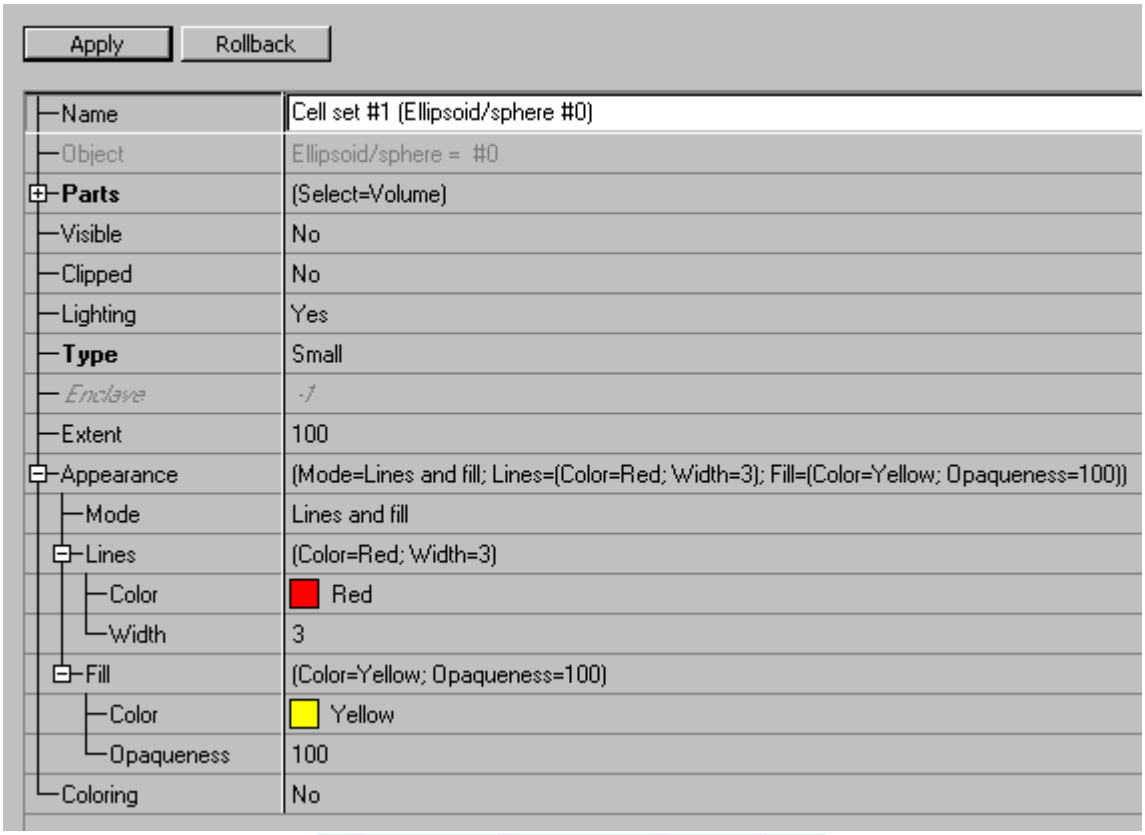
Cell set layer displays cells the user-specified type, falling in a geometric object or contacting the geometric object.

Data for the construction of this layer are taken from the **Solver**, so the layer can not be built to run on the computation.



A cell set built on the **Computational space** (**Type = Boundary**)

Parameters of the "Cell set" layer



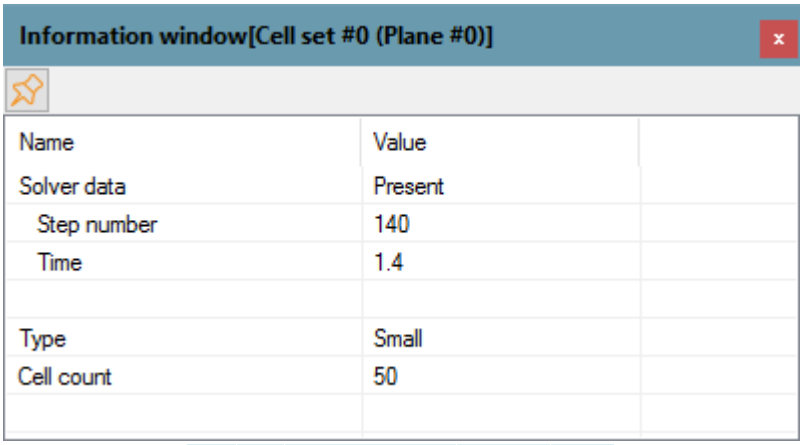
The **Properties** window of the **Cell set** layer

Layer parameters **Cell set**:

Parameter	Description
Name	The name of the layer; you can change the default name of the **Cell set #N *(object's name)***, N = 0, 1, ...
Object	The name of the **Object** on which the **Cell set** is being built. This is a read-only parameter.
Parts	Determines the parts of the **Object**, on which the **Cell set** is built (volume, the entire surface of any given **Object**'s surface,the whole surface or a fragment containing the reference point). This parameter is available when the **Cell set** is built on an object **Plane**, **Box**, **Cone/cylinder**, or **Ellipsoid/sphere**.
Parts > Select	Possible options are: • **Volume** • **Whole surface** • **Selected surfaces** • **Whole plane** (this option is available for a **Plane** object) • **Selected contour** (this option is available for a **Plane** object, it allows you to select one of several fragments of the cross section of the computational domain made by the **Plane**)
Parts > Surfaces > *(surface)*	These parameters are dependent on the type of **Object**, determine which specific surface of the **Object** to be built **Layer**. For example, a truncated cone with a channel and the recess, has the following parameters: **Lateral surface**, **Bottom base**, **Top base**, **Channel**, **First section**, **Second section**. For each of these parameters are possible options: • **No** - a layer at this surface is not constructed • **Yes** - a layer at this surface is constructed

Parameter	Description
	(See the illustration at the end of this section)
Visible	The possible values are: • **No** - layer is not displayed in the **View** (if he is not selected in the project tree) • **Yes** - layer is always displayed in the **View** window
Clipped	The possible values are: • **No** - section planes are not acting on the object • **Yes** - section planes cut to
Lighting	The possible values are: • **No** – the **Layer** is not lit by light sources • **Yes** – the **Layer** is lit by light sources
Type	Selecting the type of cells displayed the computational grid: • **All** - the cells, which fall into the **Object** or into selected parts of the **Object** (for example, cells on the base of the cylinder, cells on some faces of a box, etc.) • **Small** - cells that after clipping of the geometry became less than the allowable size. Small cells are automatically merged with their adjacent large cells. • **Bad small** - cells that after cutting facets geometry models have become small, but they were not connected to the larger cells. Such cells are removed from the calculation grid. • **Gap** -cells where the Gap model is set. • **Free surface** - the cell, through which the free surface (simulation of phase transfer is to be enabled in the project). • **Boundary** - the cell, through which the boundary of the geometry model or a moving body goes.
Enclave	Number of the displayed enclave, possible values are: **-1**, **0**, **1**, **2**, …. Numeration of enclaves starts from **0**. The value **-1** means displaying cells from all the enclaves. An *enclave* is a connected set of cells within the computational volume of the **Object** separated from other such sets, for example, by a **Moving body**. This parameter is only available when **Parts > Select = Volume**, **Type = All**, and **Object = Box | Cone/cylinder | Ellipsoid/sphere**.
Extent	Visible size of the cells (it is specified as a percentage of the original size of cells).
Appearance	Group of standard settings for displaying the layer, see description in section Folder «Layers»
Coloring **Value range > ...** **Palette > ...**	Coloring of small or gap cells depending on their *volumes* (these parameters are available only when **Type=Small** or **Type=Gap**). To apply the coloring, specify **Coloring=Yes**. The **Value range** and **Palette** groups of parameters will open; specify there settings of the coloring (see details in section General properties of Layers).

Window "Info" of the "Cell set" layer



The **Info** window of the **Cell set** layer

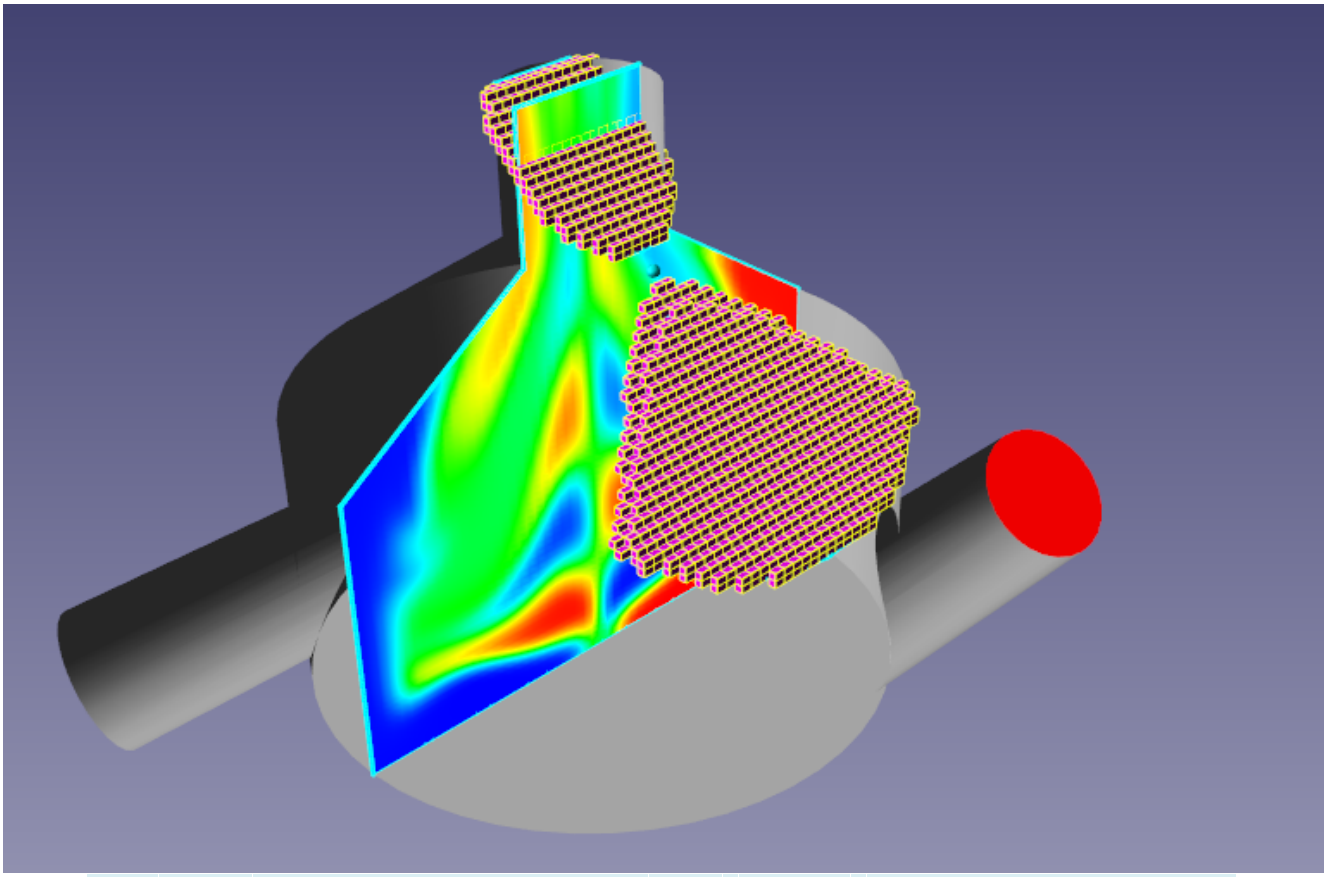
The data displayed in the window **Info** layer **Cell set**:

Data item	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current simulated time
Type	Type of cells. The possible types are: • **All** • **Small** • **Bad small** • **Gap** • **Free surface** • **Boundary**
Cell count	The number of cells of the given type

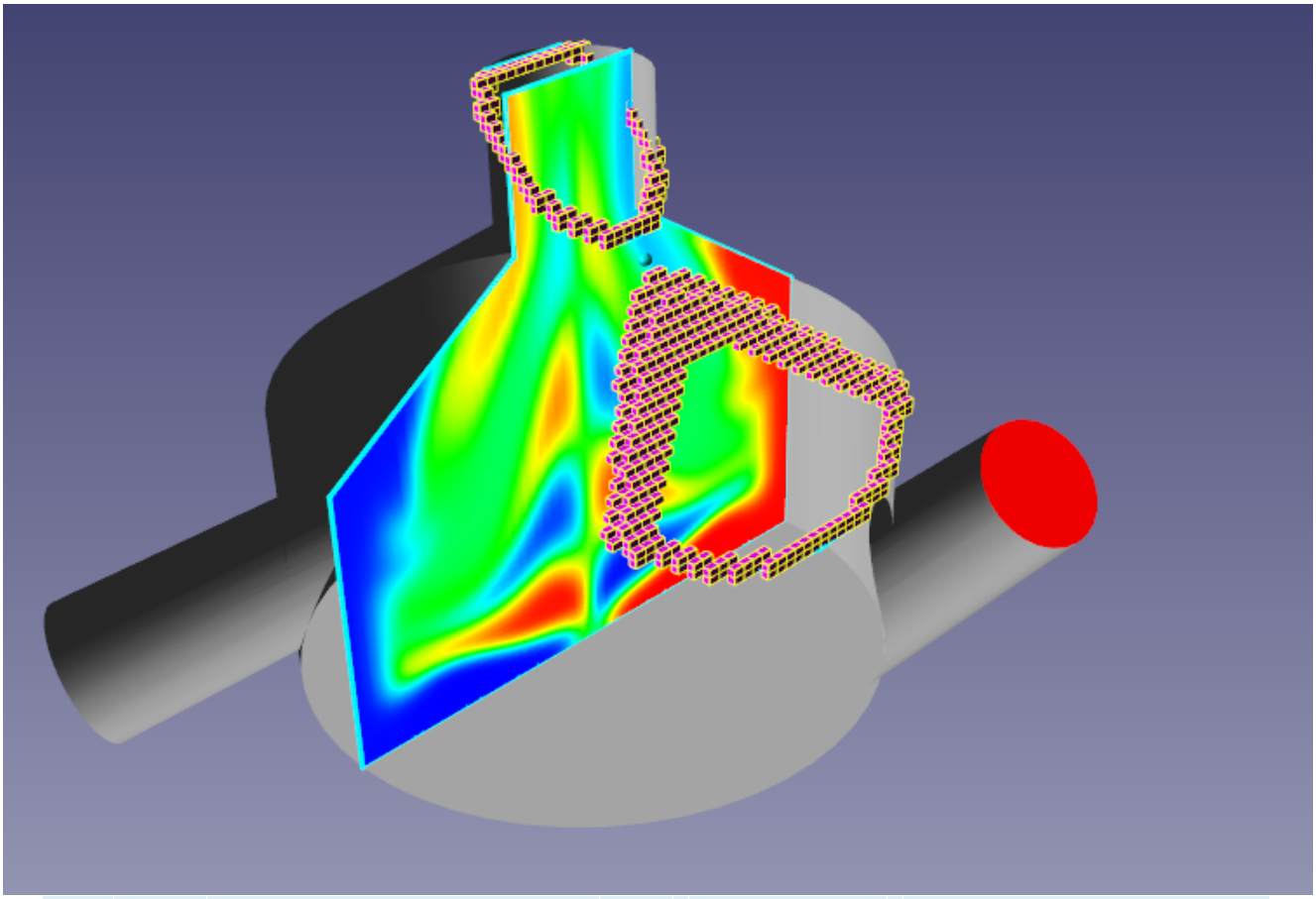
Details about the data displayed in the **Info** window for layers see in section Folder "Layers".

Examples in illustrations

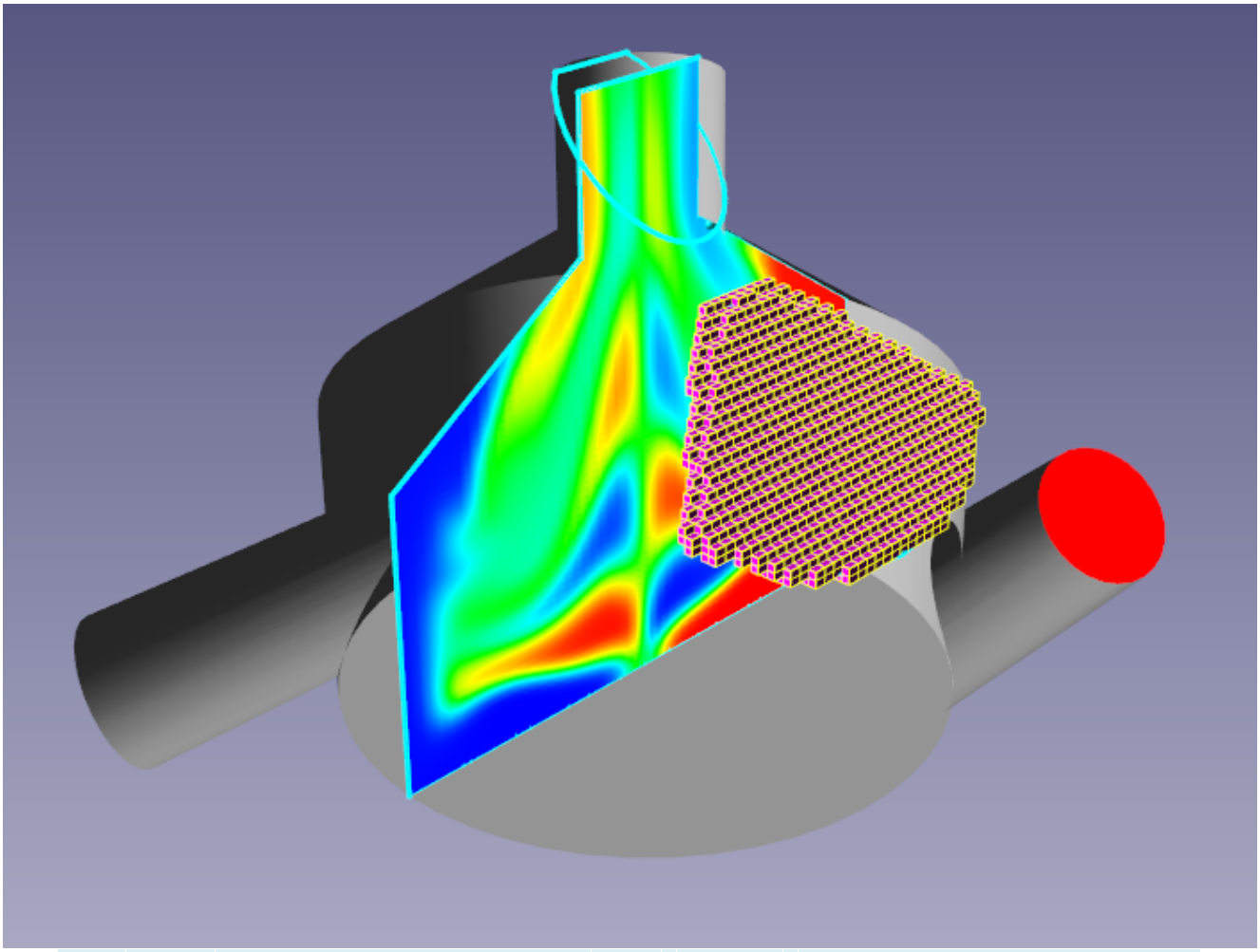
In illustrations with a set of cells that is built at a **Plane** object, the cross section of the computational domain by the **Plane** forms two contours. If **Parts > Select = Selected contour**, the set of cells is built only on the circuit, inside which is a reference point in the plane.



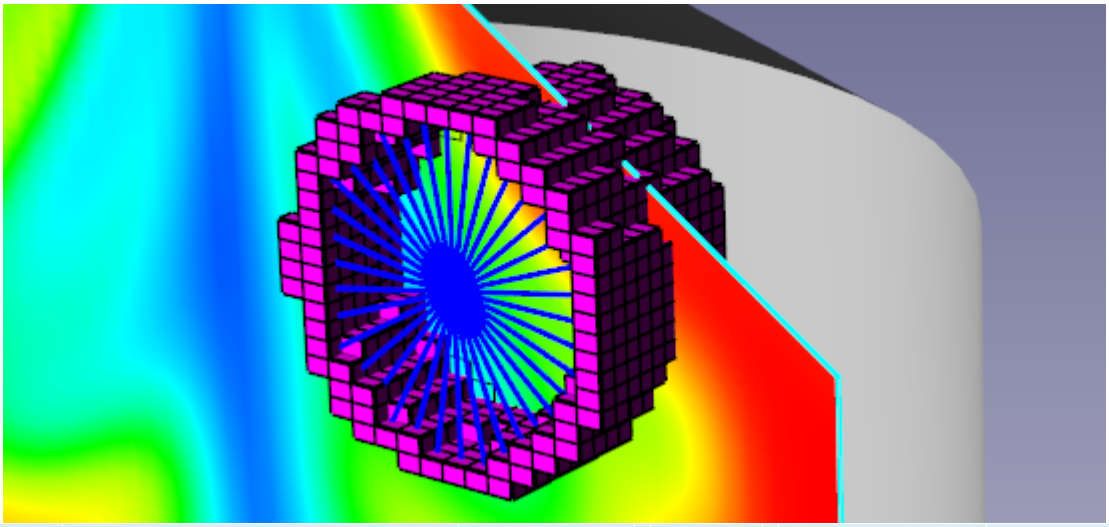
Layer **Cell set**, which has been built on a object **Plane**, **Type=All**, **Parts > Select = Whole plane**



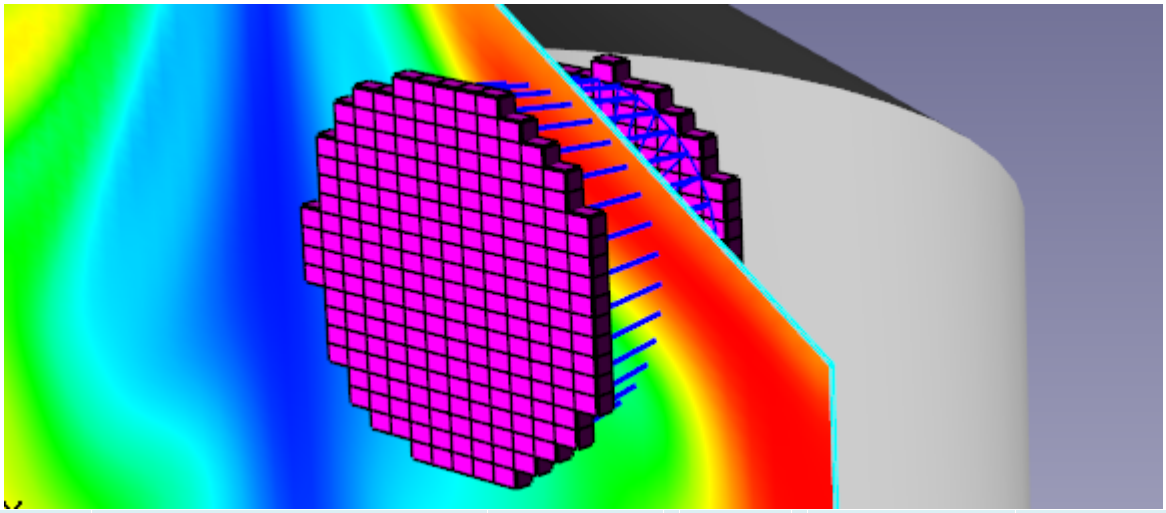
Layer **Cell set**, which has been built on a object **Plane**, **Type=Boundary**, **Parts > Select = Whole plane**



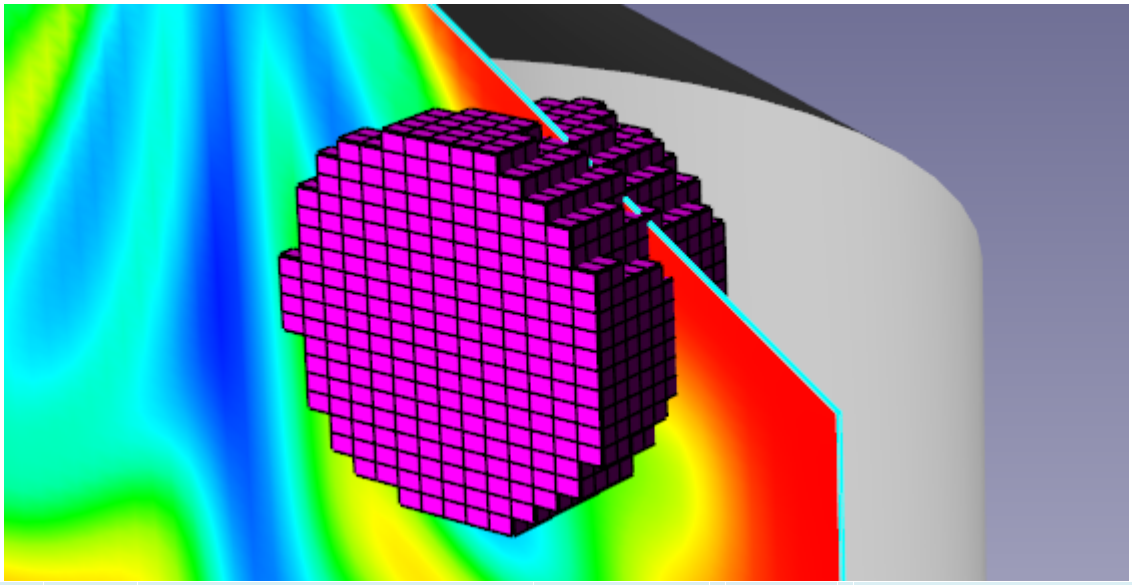
Layer **Cell set**, which has been built on a object **Plane**, **Type=All**, **Parts > Select = Selected contour**



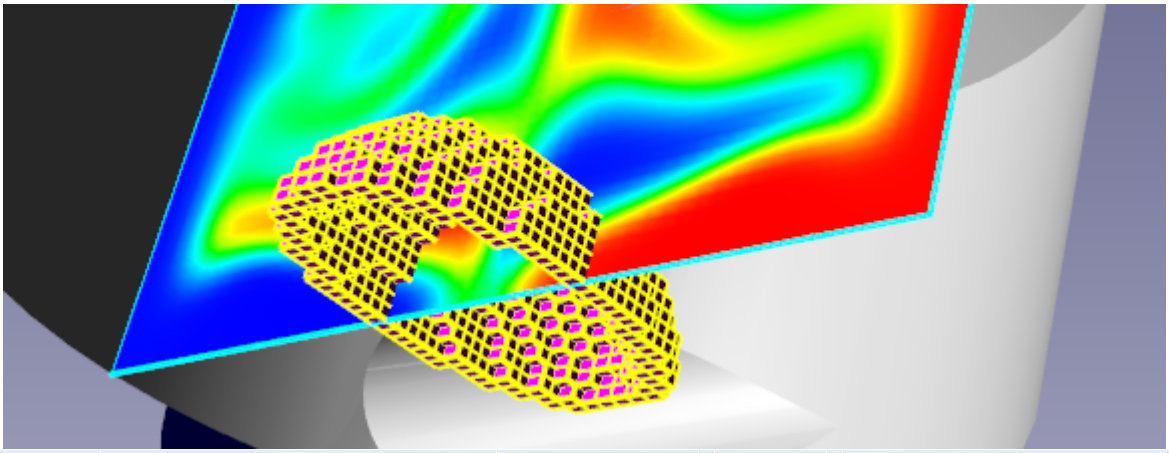
Layer **Cell set**, which has been built on a object **Cone/cylinder**, **Type=All**, **Parts > Select = Selected surfaces** (lateral surface of the cone/cylinder has been selected)



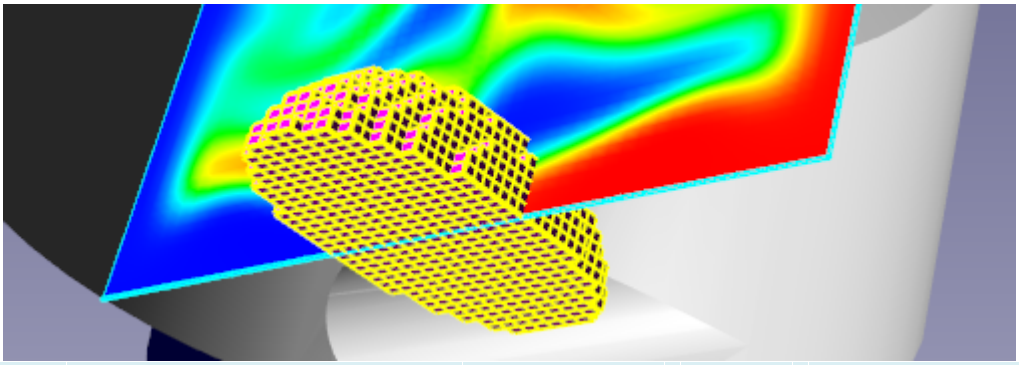
Layer **Cell set**, which has been built on a object **Cone/cylinder**, **Type=All**, **Parts > Select = Selected surfaces** (both bases of the cone/cylinder have been selected)



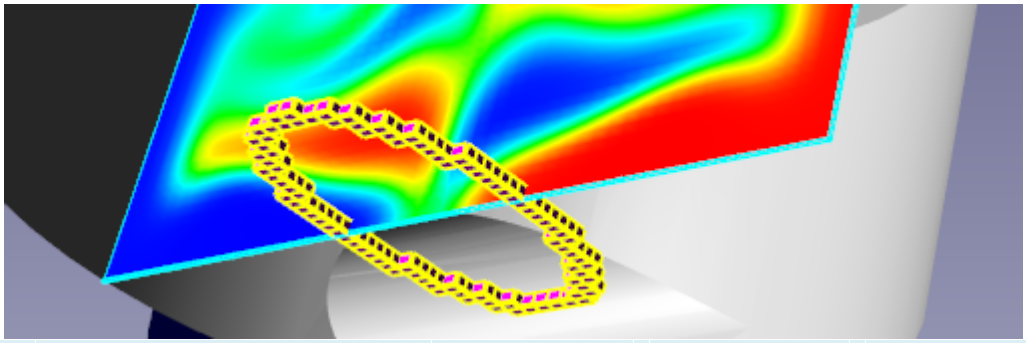
Layer **Cell set**, which has been built on a object **Cone/cylinder**, **Type=All**, **Parts > Select = Volume**



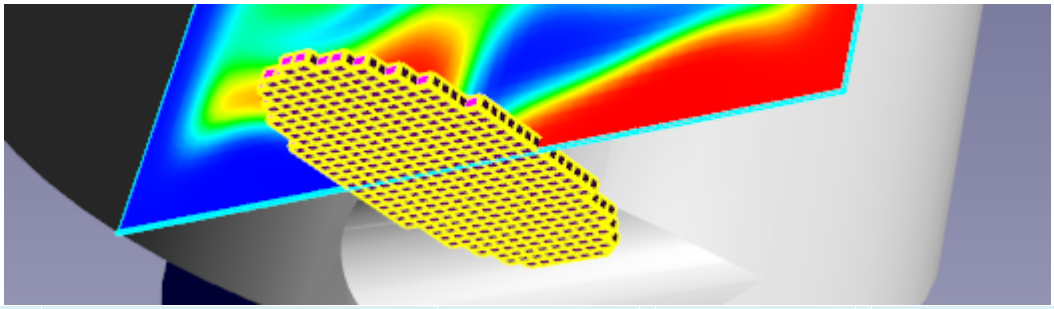
Layer **Cell set**, which has been built on a object **Ellipsoid/sphere**, **Type=All**, **Parts > Select = Whole surface**



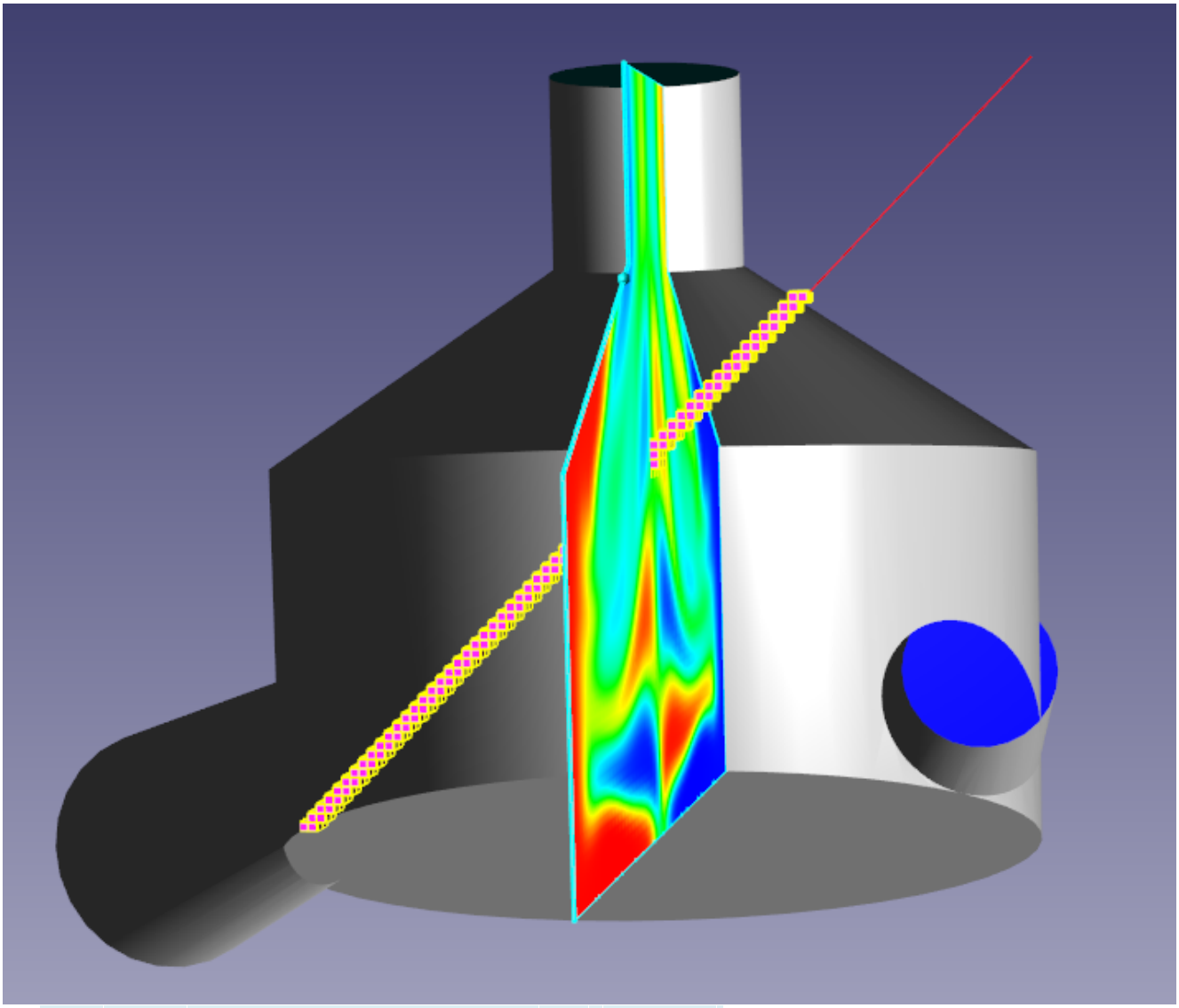
Layer **Cell set**, which has been built on a object **Ellipsoid/sphere**, **Type=All**, **Parts > Select = Volume**



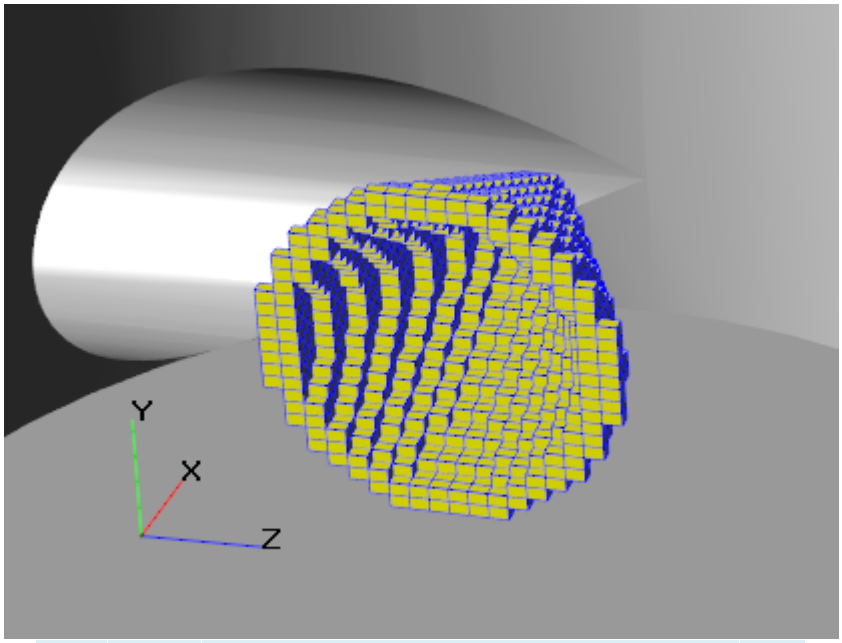
Layer **Cell set**, which has been built on a object **Ellipsoid/sphere**, **Type=Boundary**, **Parts > Select = Whole surface**



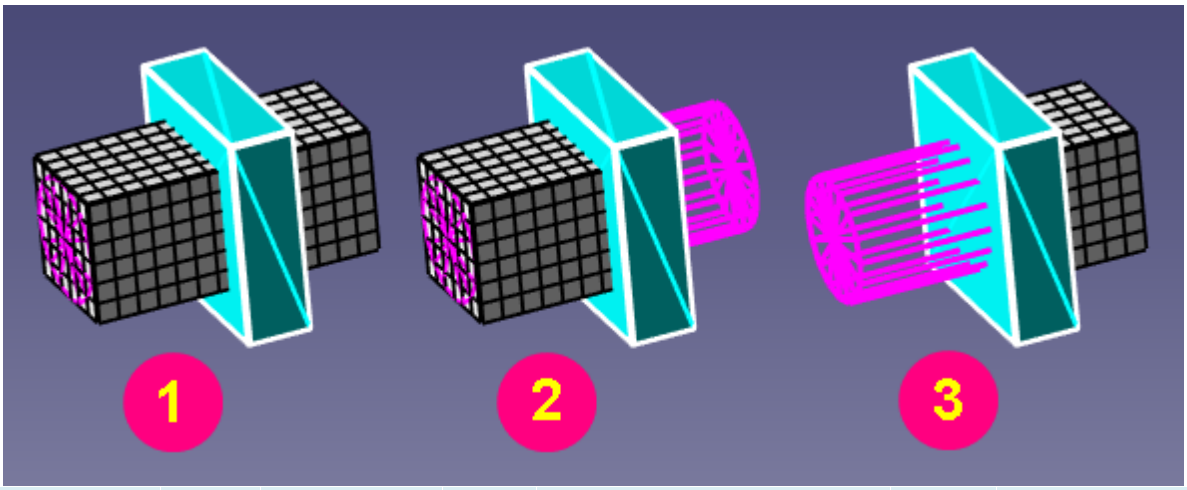
Layer **Cell set**, which has been built on a object **Ellipsoid/sphere**, **Type=Boundary**, **Parts > Select = Volume**



Layer **Cell set**, which has been built on a object **Line**, **Type = All** (pink layer shows the cell along the line)



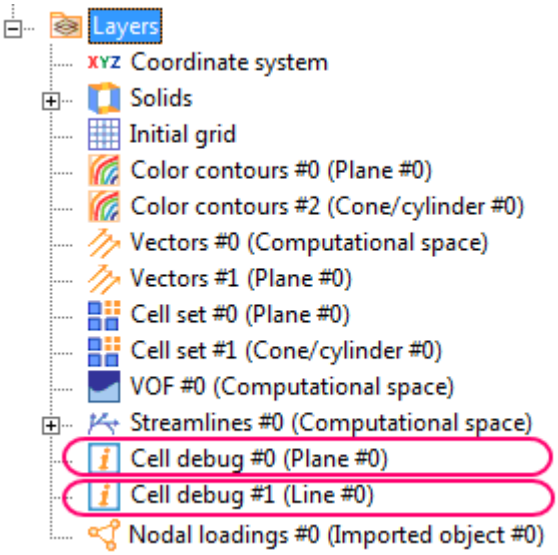
Layer **Cell set**, which has been built on a lateral surface of a **Cone**



Operation of the **Enclave** parameter of a **Cell set** that is built in the volume of a **Cylinder** separated into two enclaves by a **Moving body**.

1: Enclave=-1, **2: Enclave=0**, **3: Enclave=1**

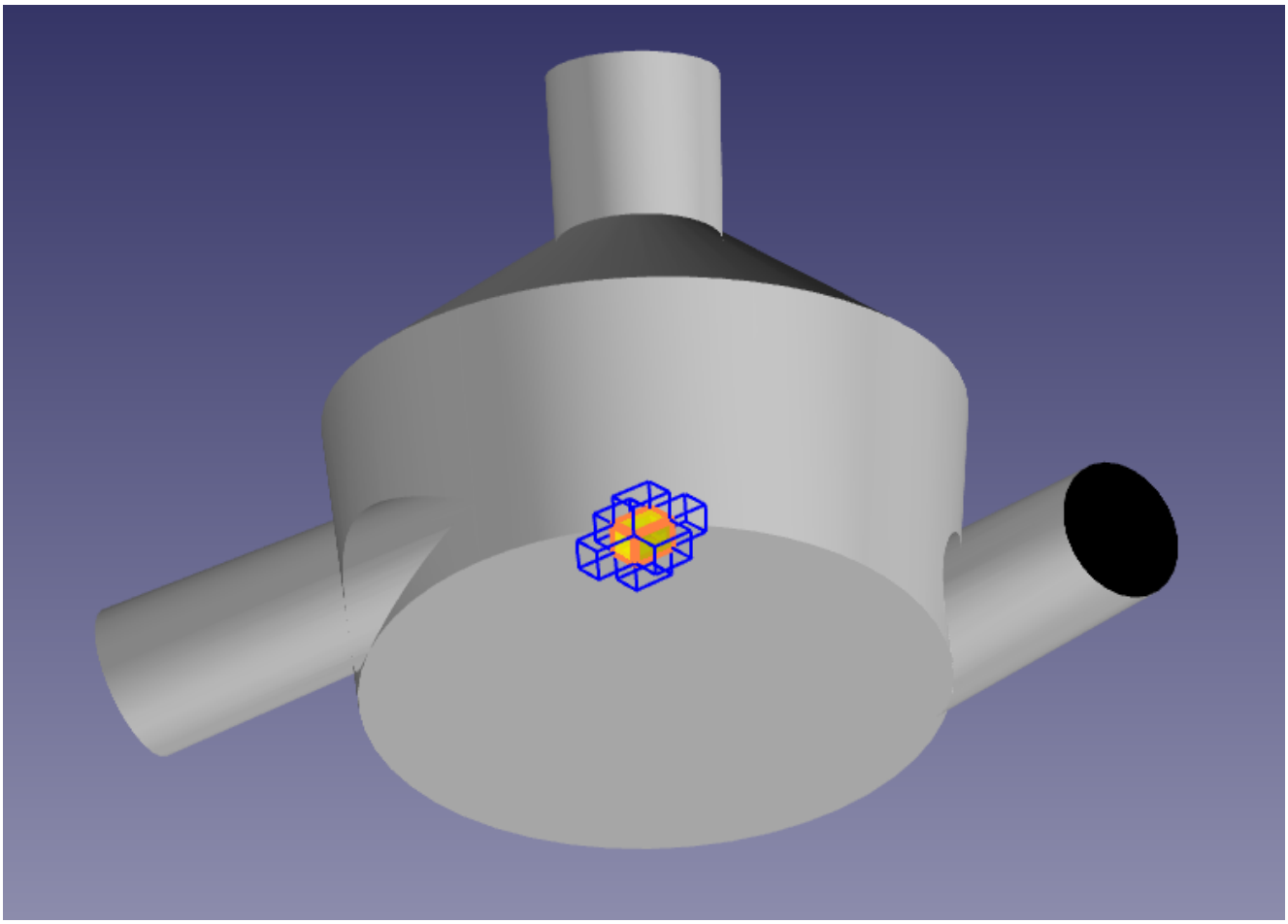
8.1.8.5.8.18 Layer «Cell debug», user interface



The **Cell debug** layer in the project tree

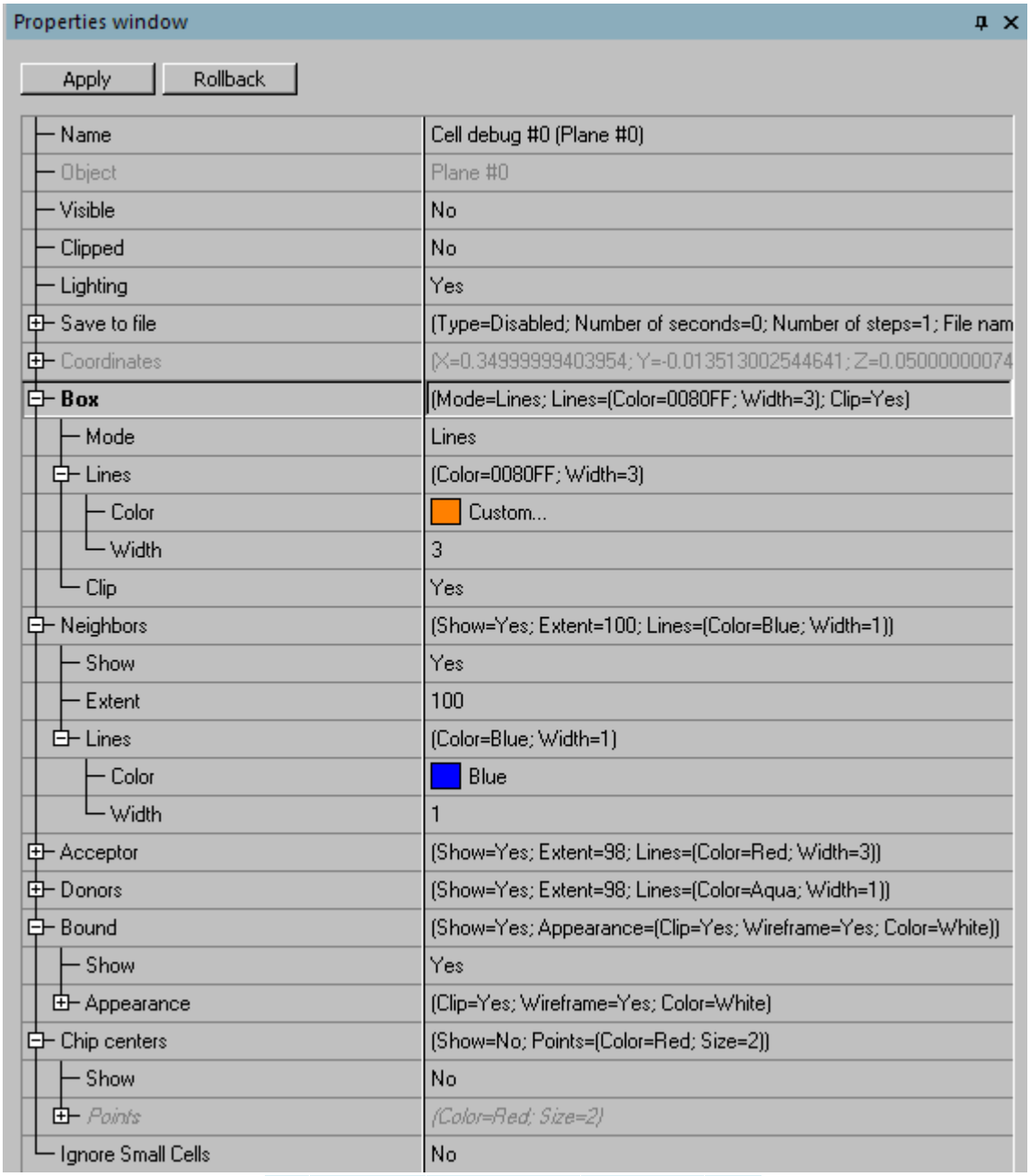
Layer with the calculated data **Cell debug** is intended to indicate the position of the cell of the computational grid in the subregion and the values of a variable in a given cell.

A *chip* is the face of the cell surface, a cut after applying a geometry model of the surface or the moving body, and after the construction of the free surface.



Images of the debugging cell (orange with yellow edges) and its neighbors (blue)

Parameters of the «Cell debug» layer



The **Properties** window of the **Cell debug** layer

Parameters of the **Cell debug** layer:

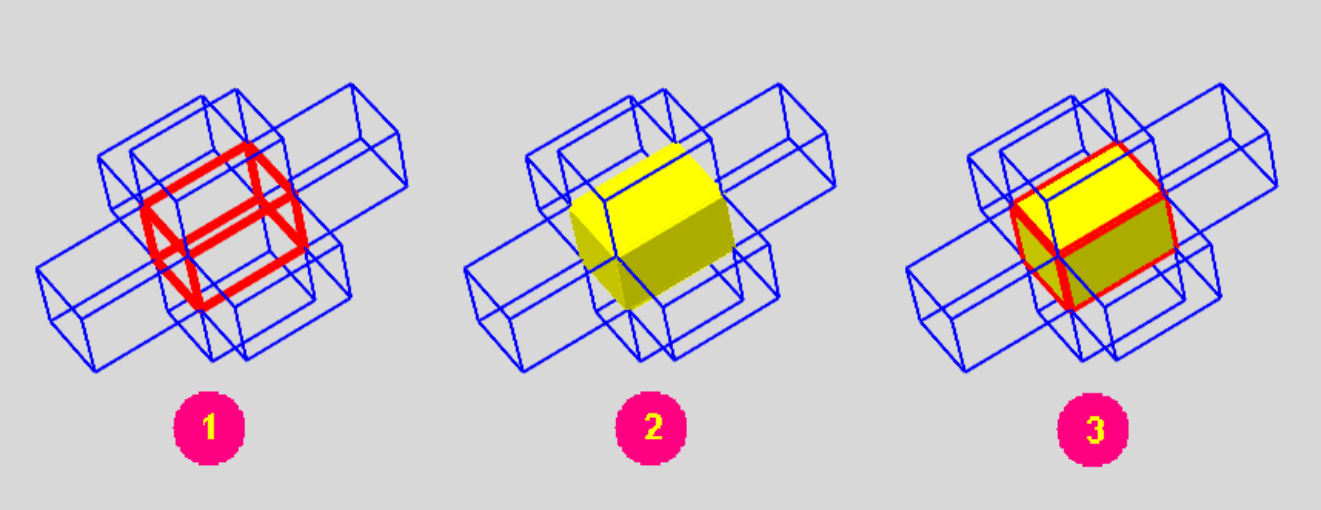
Parameter	Description
Name	The layer's name (this parameter allows you to change the default layer's name **"Cell debug #N (Object)"**, formed from the layer's type, a number, and the name of the **Object**, on which the layer has been built).
Object	See General properties of Layers. The **Cell debug** layer can be built on the following objects: • **Computational space** • **Line** • **Plane** When a **Cell debug** layer is specified in the **Computational space**, in simulations defined in a sector statement with periodic surfaces, the debug cells, which have been specified **By unique index** or **By double-index**, are displayed *in the initial sector* of the geometric model. The cells, which have

Parameter	Description
	been specified **By coordinates**, are displayed *according to the specified* ***Coordinates*** (so they might locate *not necessarily in the in the initial sector*). When a **Cell debug** layer is specified on a **Line** or on a **Plane**, the debug cell will be those cell, into which the reference point of the **Line** or the **Plane** falls (in simulations defined in a sector statement with periodic surfaces this reference *can fall into any sector*). When the reference point is moved, the debug cell also moves along the **Line** or in the **Plane**.
Visible	See General properties of Layers.
Clipped	
Lighting	
Save to file > ...	
Identification > ...	This group of parameters specify the position of the cell (these parameters are only available when **Object = Computational space**).
Identification > Method	This defines the method for specifying the debug cell: • **By unique index** specifies the debug cell by the path in the tree of cells of different levels ("multilevel format"), see section Cell's number (index). • **By double-index** specifies the debug cell by pair of "hypercell-cell" indices ("two-level format"), see section Cell's number (index). • **By coordinates** specifies the debug cell by its coordinates along axes **X**, **Y**, **Z**. In simulations defined in a sector statement with periodic surfaces, the debug cells, which have been specified **By unique index** or **By double-index**, are displayed *in the initial sector* of the geometric model. The cells, which have been specified **By coordinates**, are displayed *according to the specified* ***Coordinates*** (so they might locate *out of the in the initial sector*).
Identification > Unique index*)	A unique string index, which identifies the debug cell, see section Cell's number (index).
Identification > Double index > ...*)	A pair of indices - index of a hypercell and index of a cells within the hypercell, see section Cell's number (index).
Identification > Double index > Hypercell	
Identification > Double index > Cell	
Identification > Coordinates > X*)	Coordinates of the cell along axes X, Y, Z
Identification > Coordinates > Y*)	
Identification > Coordinates > Z*)	
Box > ...	Specifying the displaying of the cell.
Box > Mode	The method of displaying for surfaces and contours of the debug cell. The possible options are: • **Lines** - displays only the edges the cell • **Fill** - shaded surface display cell • **Lines and fill** - display edges and shaded surface of the cell (See illustration)
Box > Lines > Color	Line color and width of the cell edges debug
Box > Line > Width	
Box > Fill > Color	Color facesdebug cell

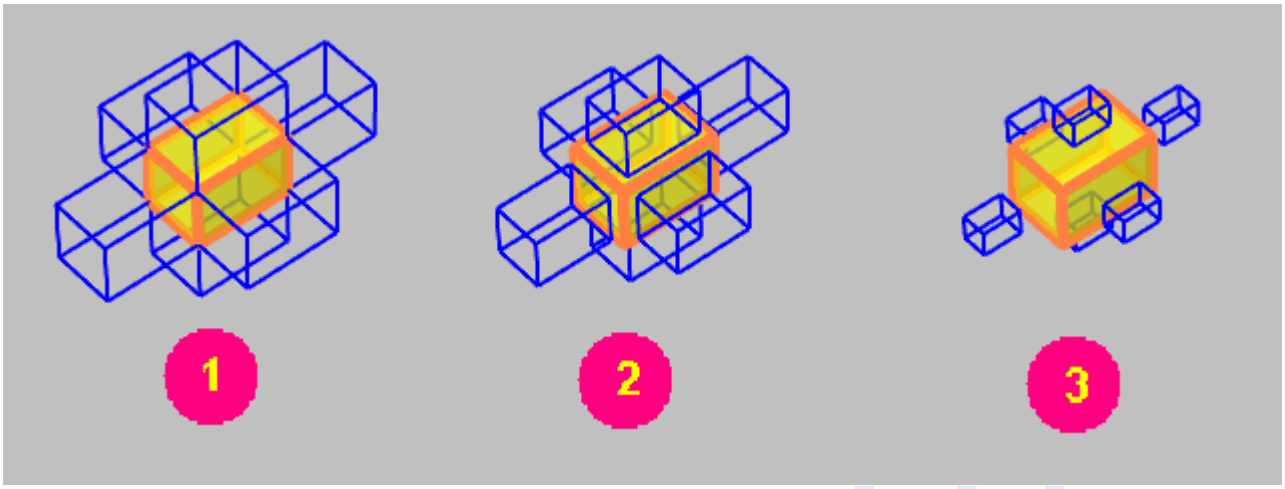
Parameter	Description
Box > Fill > Opaqueness	Opacity shading for faces of debug cells (defined as a percentage from 0 to 100)
Box > Clip	Cut the cell's box by the geometry. Possible options are: • **No** - displaying of the whole cell (without truncation) • **Yes** - displaying the surface of the cell minus the portion severed facets geometry model
Information window > ...	These settings allow you to regulate the volume of data that are transferred into the **Info** window.
Information window > Cell variables	Displaying in the **Info** window values of variables in the debug cell. Possible options: **Yes | No**.
Information window > Chip data	Displaying in the **Info** window chip data of the debug cell. Possible options: **Yes | No**.
Information window > Chip variables	Displaying in the **Info** window values of variables on chips of the debug cell. Possible options: **Yes | No**.
Neighbors > ...	Customize the display of cells adjacent to the imaging unit debugging
Neighbors > Show	This defines whether to display the neighboring cells. Possible options are: • **No**- the adjacent cells are not displayed • **Yes**- adjacent cells are displayed The neighbors of the debug cell in a simulation defined in a sector statement with periodic surfaces are displayed in the same subregion as those, where the debug cell locates, in the same sector. So, if the debug cell locates near the periodic surface, then the neighboring cells will be displayed *in the same sector, not in the adjacent sector*. In a simulation defined in a sector statement with periodic and sliding surfaces, the neighbors of the debug cell, which locates near the sliding surface, *will be displayed in the initial sector*.
Neighbors > Extent	The apparent size of the neighboring cells, as a percentage of the original size. (See illustration)
Neighbors > Lines > Color **Neighbors > Line > Width**	Line color and width for the image edgesof adjacent cells.
Acceptor > ... **Donors > ...**	(Parameters for developers)
Bound	Customize the display faces debug cell formed surfaces geometry of the computational domain
Bound > Show	Whether to show facets of geometry models that fall into the visualized cell. Possible options are: • **No** - facets of a geometry model, trapped in the cell, are not displayed • **Yes** -to show facets of the geometry model, caught in a cell
Bound > Appearance > Clip	Show only facets of the geometry model, entering the debug cell that are inside it. Possible options are: • **No** - display facets geometry model, caught in a cell completely • **Yes** - to show only fragments of the facets of the geometry lie inside the cell
Bound > Appearance > Wireframe	Display frame facets geometry model entering the debug cell. Possible options are: • **No** - display surfaces facets geometry model, which fell into the cell • **Yes** -show the carcass facets geometry model, which fell into the cell
Bound > Appearance > Color	Choice of color lines chassis facets geometry model entering the debug cell.

Parameter	Description
Chip centers > ...	Settings of displaying (as colored dots) the centers of the faces of the debug cell and the center of the debug cell.
Chip centers > Show	Display centers of the faces of the debug cell and the center of the debug cell (see illustration). Possible options: **Yes | No**.
Chip centers > Points > Color **Chip centers > Points > Size**	The color and size of the dots indicates the center of the faces of the cell and the center of debugging the debug cell. (These parameters are available when **Chip centers > Show = Yes**)
Ignore Small Cells	Ignore small cells. Possible options: **Yes | No**.

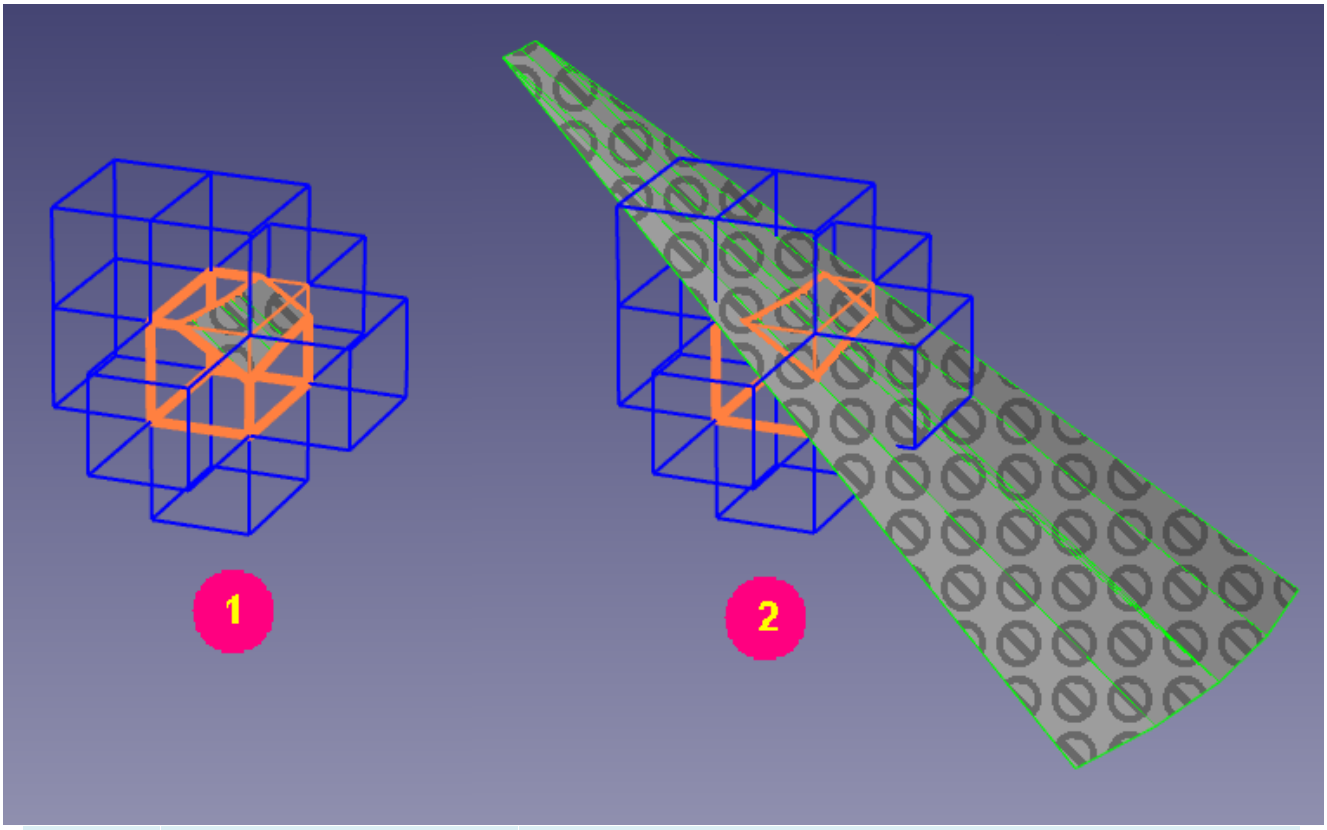
*) The parameters are available for appropriate values of the parameter **Identification > Method**.



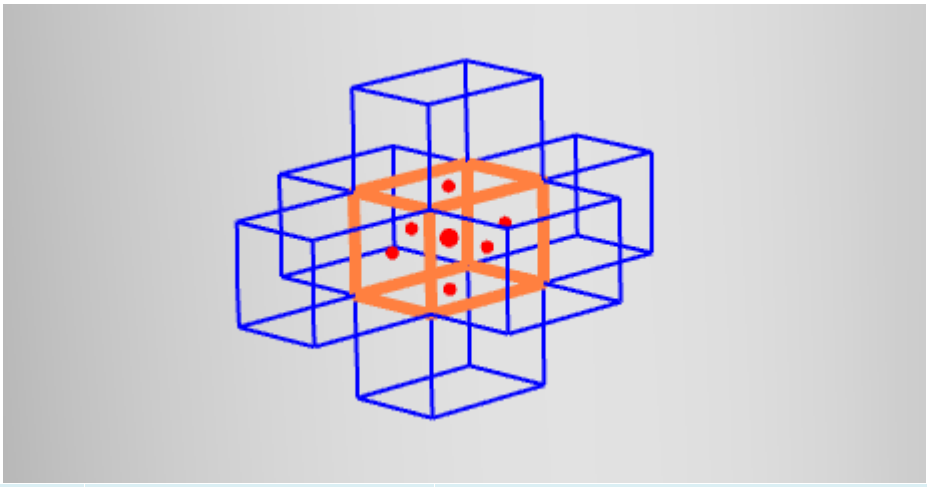
Influence of the parameter **Box > Mode** to display the debug cell (**1** -**Lines**; **2** - **Fill**; **3** - **Lines and fill**)



Influence of the parameter **Neighbors > Extent** to display adjacent cells (**1** - **100**; **2** - **80**; **3** - **40**), defined as a percentage



Influence of **Bound > Appearance > Clip** on displaying of facets of geometry model, falling into the debug cell (1 - **Clip = Yes**, 2 - **Clip = No**)



Use the **Chip centers** parameters to tune displaying centers of the faces of the debug cell and the center of the debug cell

Window «Info» of the «Cell debug» layer

The data displayed in the window Info layer Cell debug:

Parameter	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step

Parameter	Description
Solver data > Time	Current simulated time
Cell class	Class of the cell (information for developers)
Cell type	Type of the cell (information for developers)
Unique index	Cell number in the multilevel format
Double index	Cell number in the two-level format
Level	Level of splitting of the cell (information for developers)
Creators	(parameters for developers)
Volume	Volume of the debug cell
Center > X	Coordinates of the center of the debug cell
Center > Y	
Center > Z	
Box center > X	Coordinates of the center of the cell with a cut geometry
Box center > Y	
Box center > Z	
Half size > X	Half the size of the cell along the coordinate axes
Half size > Y	
Half size > Z	
Total number of child cells	Number of child cells
	Type subsidiary cell N
Subregion	Name of the subregion in which the cell is located
Model	The model name specified in the cell
Gauss fulfillment	Displayed value is a measure of the quality of the computational cell. Value is obtained by integrating the area-oriented facets that make up the part of the computational cell. For a well-formed cell displays the value should be zero, but due to rounding errors of machine it is not 0 *(Information for developers)*
Level	Level of splitting of the cell
No phase > *(name of variable)*	Value of a variable that does not belong to any phase in the cell
Phase #N > *(name of variable)*	Value of the variable belonging to phase N, in the cell
(Name of the physical process)* > *(name of variable)	The value of the variable characterizing the specific physical process in the cell

Details about the settings window **Info** for layers see in section Folder "Layers".

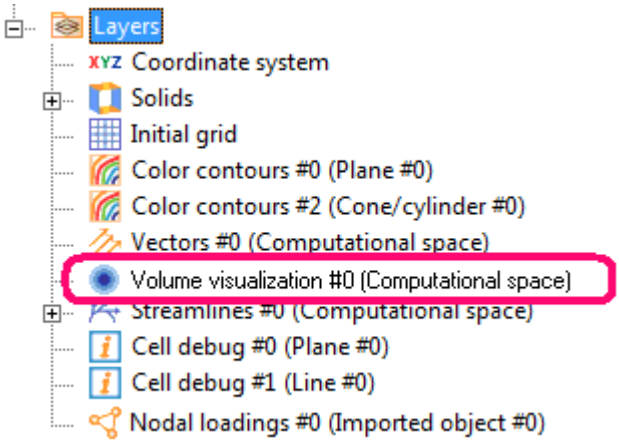
Components of the text file for the layer «Cell debug»

With each entry in the file shows the following information:

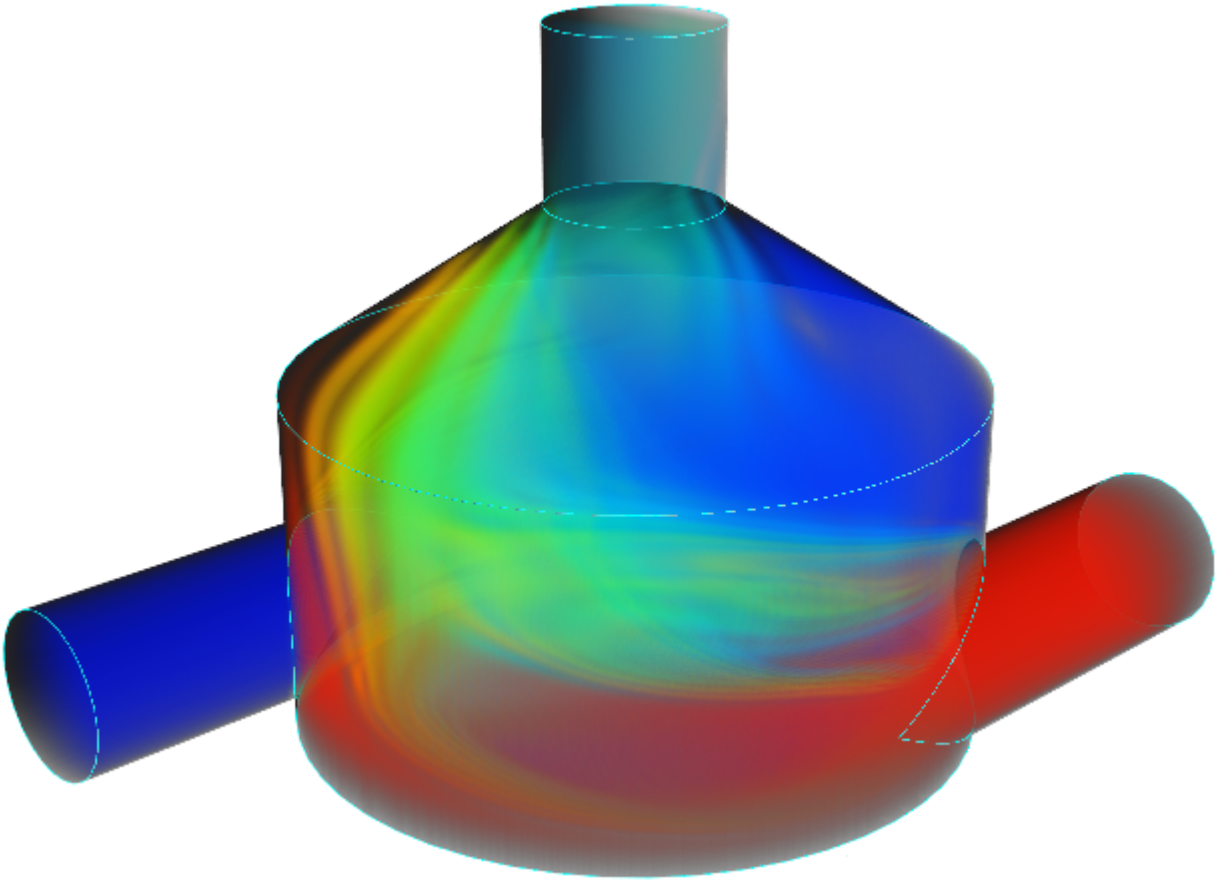
Step	Step number
Time	Time
Class	(Information for developers)
Type	(Information for developers)
Variable	Variable that determines the trajectory of the plot line
ID	Cell number in the multilevel format

Hypercell	Cell number in the two-level format
Cell	
Level	Level of splitting of the cell (information for developers)
Center.x	Coordinates of the center of the cell to x, y, z
Center.y	
Center.z	
HalfSize.x	Half the size of the cell to the x, y, z
HalfSize.y	
HalfSize.z	
Model	The model given in the cell
NoPhase	Group of variables that do not belong phases
NoPhase>*(variable names)*	Values of the variables in the cell
Phase #i	Group variables defined in Phase i
Phase #i>*(variable names that do not belong to physical processes)*	Values of the variables in the cell
(The name of the physical process)	Type of physical process
(The name of the physical* process) >*(variable names belonging to the physical process)	Values of the variables in the cell

8.1.8.5.8.19 Layer «Volume visualization», user interface



The **Volume visualization** layer in the project tree

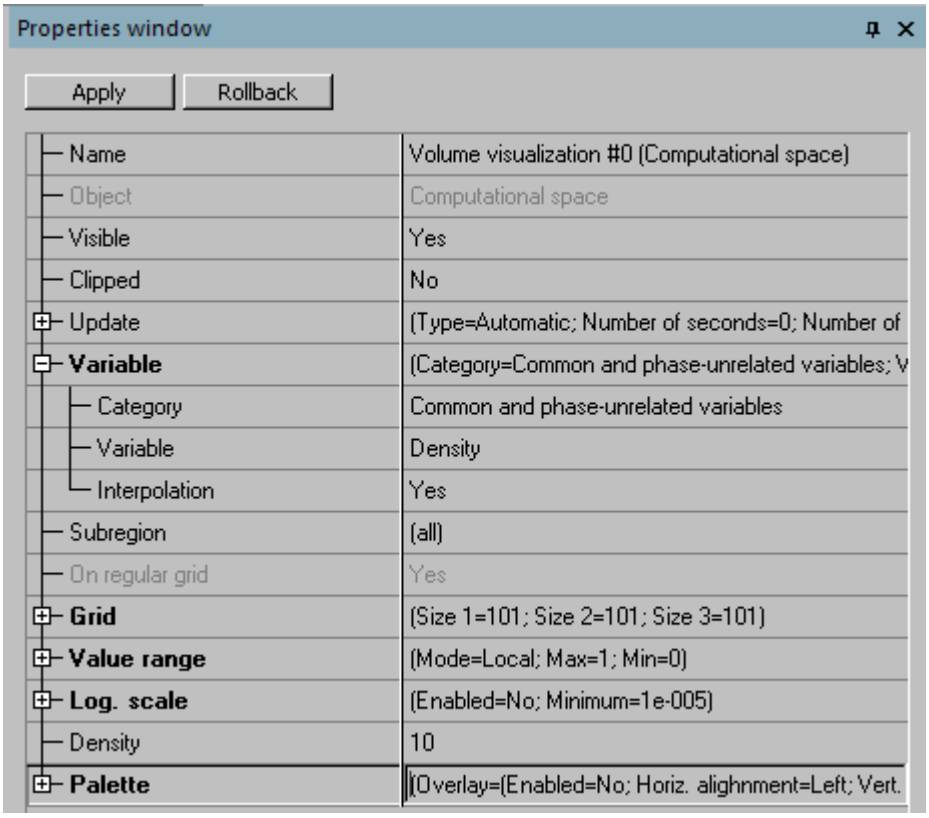


Example of the **Volume visualization** layer

The **Volume visualization** layer allows you to visualize changing variables in the whole space (for example, view zones with high substance concentration or density).

Before use of the **Volume visualization** layer ensure that you have installed the last versions of video card drivers.

Parameters of the «Volume visualization» layer



The **Properties** window of the **Volume visualization** layer

Parameters of the **Volume visualization** layer:

Parameter	Description
Name	Layer name (this option allows you to change the default name "**Volume visualization #N (Object)**", formed from the names of the layer type, number and object on which the layer is built).
Object	See General properties of Layers.
Visible	
Clipped	
Update > ...	
Variable > ...	
Subregion	
On regular grid	
Grid > ...	
Value range > ...	
Log. scale > ...	
Density	This parameter allows you to increase or decrease the layer's visual density
Palette > ...	See Parameters for defining a palette. Transparency of each color in the palette used for the visualized variable can be defined in parameters **Palette > Transparency > Opaqueness** (defined in percents in range from 0 to 100).

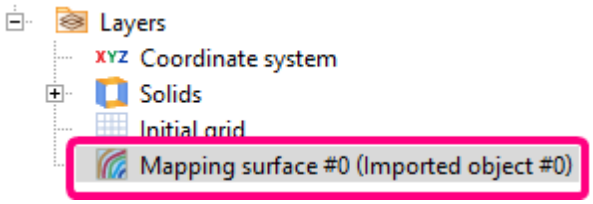
Window «Info» of the «Volume visualization» layer

Window **Info** of the **Volume visualization** layer

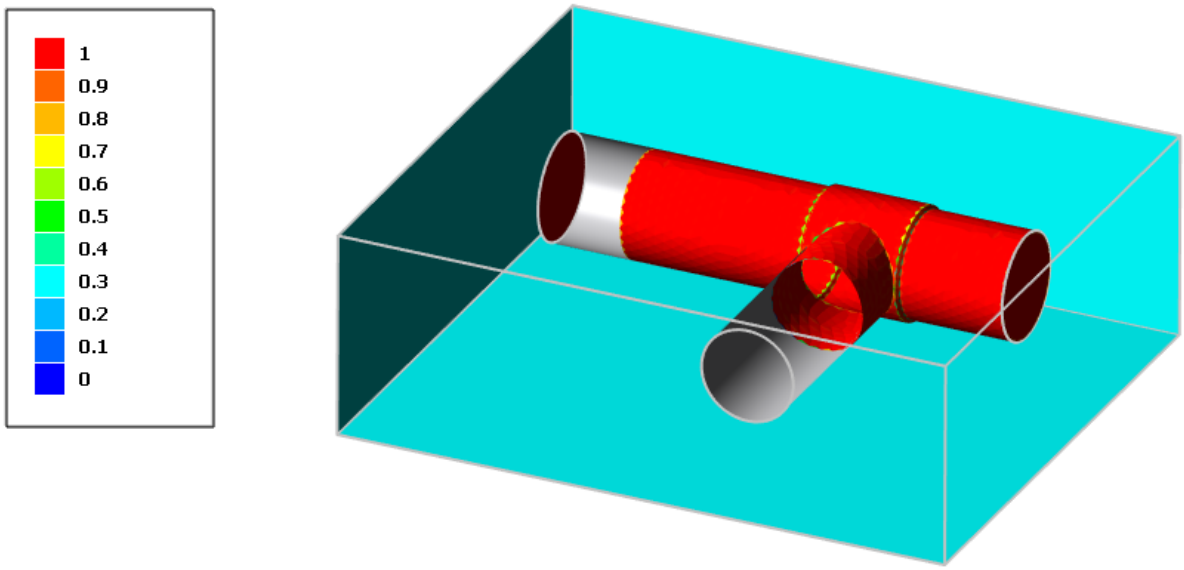
Parameter	Description
Solver data	Availability of data from **Solver** for the **Layer** • **Absent**: the calculated data are not available • **Present**: the calculated data are available
Solver data > Step number	Number of the current time step
Solver data > Time	Current simulated time
Variable	Information about the variable used to build the volume visualization
Palette	Palette, used for the volume visualization

Details about the **Info** window for layers see in section Folder "Layers".

8.1.8.5.8.20 Layer «Mapping surface», user interface



The **Mapping surface** layer in the project tree



Example of the **Mapping surface** layer

In simulations with **External Connections** when mapping is enabled, it is possible to visualize the mapping surface.

The **Mapping surface** layer displays distribution of the value $f_i^{(1)}$ over the source surface.

The source surface is a set of triangles, each of them is mapped to the other (target) surface.

Let us denote the area of projection of the i-th triangle on the second surface as $S_i^{(2)}$, and area of the source surface denote as $s_i^{(1)}$.

Let's consider their relation, the value $f_i^{(1)}$:

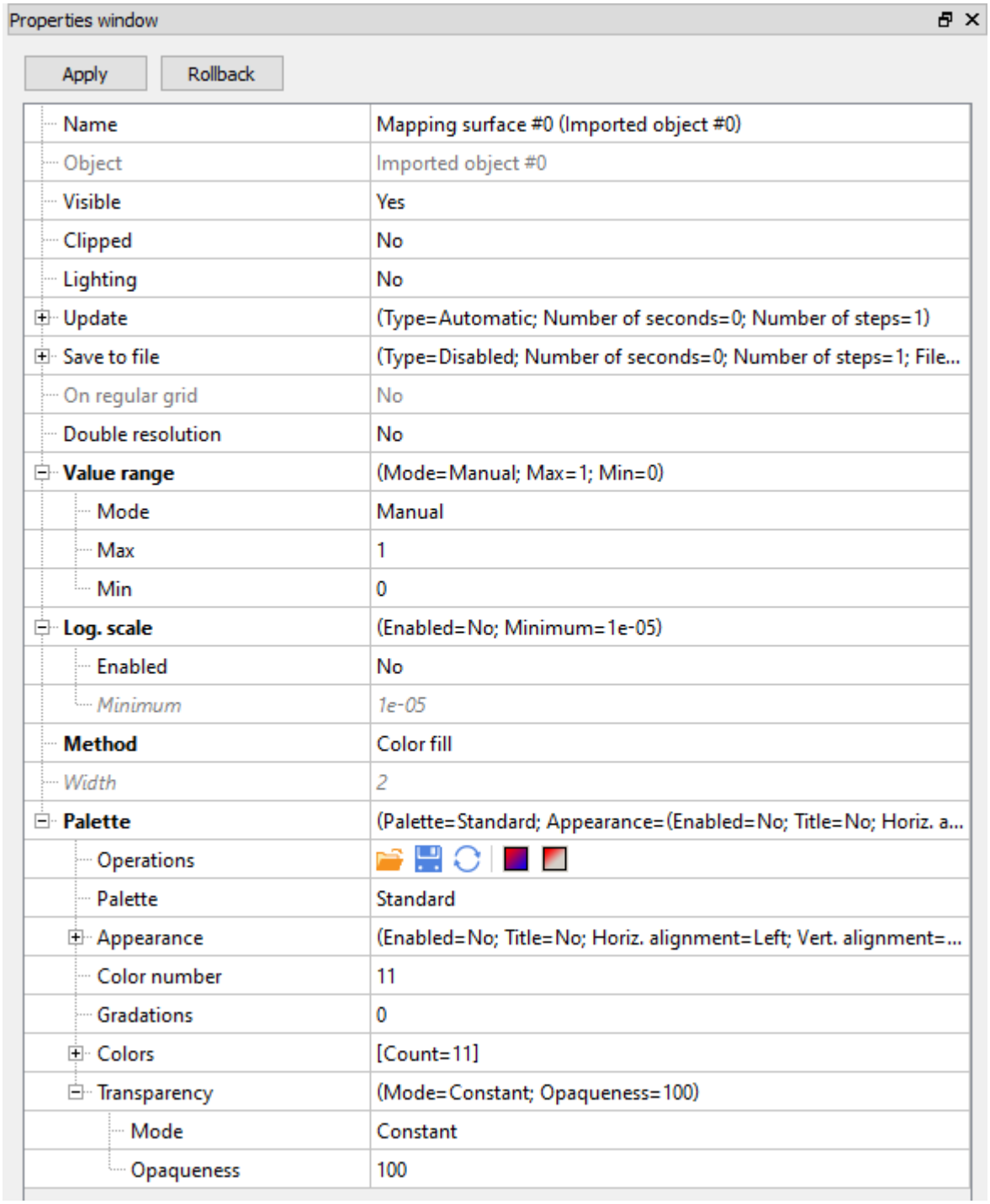
$$f_i^{(1)} = S_i^{(2)} / s_i^{(1)}$$

In the layer **Mapping surface**, each triangle of the source surface is colored according to its value of $f_i^{(1)}$, if it is not zero.

When $f_i^{(1)} = 0$, appropriate triangles of the surface will become invisible.

See details in the section Mapping.

Parameters of the «Mapping surface» layer



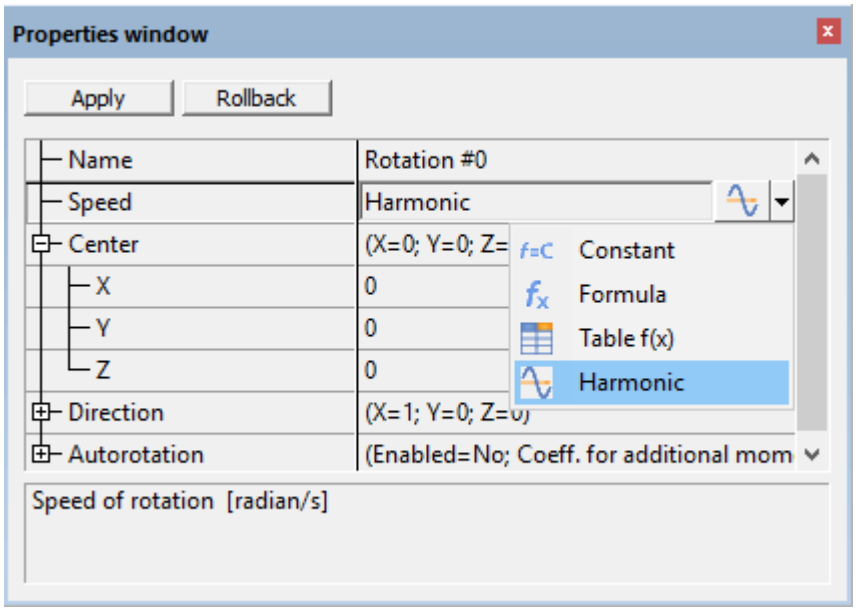
The **Properties** window of the **Mapping surface** layer

Parameters of the **Mapping surface** layer:

Parameter	Description
Name	Name of the layer (you can specify another name instead of the standard name **Mapping surface**).
Object	See General properties of Layers.
Visible	
Clipped	
Lighting	
Update > ...	

Parameter	Description
Save to file	
On regular grid	
Double resolution	
Value range > ...	
Log. scale > ...	
Method	
Palette > ...	

8.1.9 Window «Properties»



Example of the **Properties** window

The **Properties** window is used to view or change parameters of elements in the project tree.

When the **Properties** window is open, it displays the information for that project tree's element, which is currently selected.

To open the **Properties** window, check the item **View > Properties window** in the Main Menu.

To close the **Properties** window, do one of the following:

- *either* uncheck the line **View > Properties window** in the Main Menu
- *or* click the "**×**" icon in the upper right corner of the **Properties** window

The **Properties** window contains:

- list of parameters in the form of a table with two columns containing:
  - names of the parameters
  - values of the parameters
- the **Apply** and **Rollback** buttons located on the top (these buttons become active only after some changes are made in values of fields)
- the comment field, which can display a brief description of the selected parameter

Some rows in the table can be folders that contain child elements. To expand or collapse the folders, click symbols ⊞ and ⊟ in the tree of parameters or press the **Enter** key on your computer's keyboard.

A table's row can be:

- simple input field with a numerical or text value (the value is entered directly into the field)
- selection field is a drop-down list, a numerical value or text can be selected from it
- expanded data input field, the value of which can be:
  - a number or a vector that are set directly in the field
  - a function that is set as a user-defined formula
  - a function that is set in a tabular form
  - a predetermined formula (for example, the formula of the harmonic law)

- an array definition field

See section Parameters for defining a palette for details about settings of color **Palettes** that are used in properties of many **Layers** and in properties of background of the **View** window.

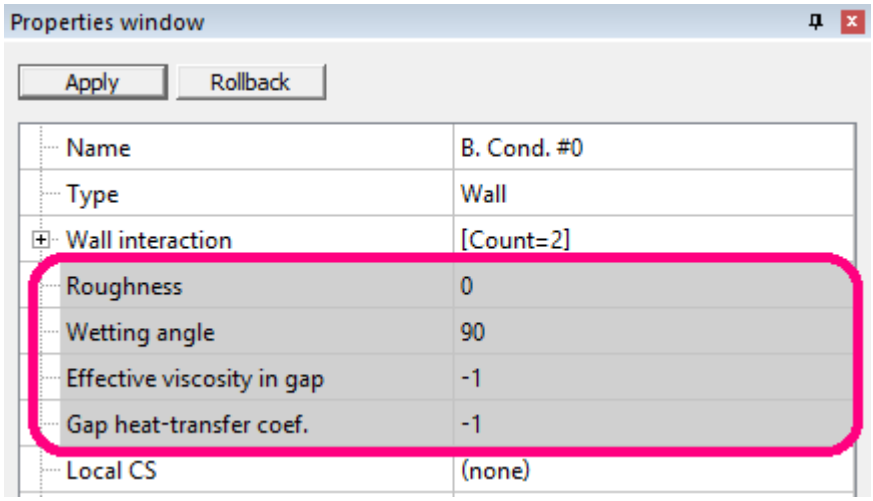
Availability of some parameters to be changed

Changing some parameters might be unavailable (this might depend on values of other parameters). Parameters, which can not be changed, are displayed with gray font color (they are faded).

Adding parameters to the list of external parameters

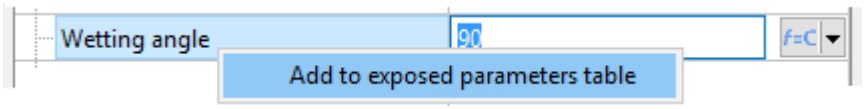
It is possible to change some parameters using third-party's software. A list of these parameters is displayed in the **Exposed parameters window**.

External parameters are highlighted in the **Properties** window with a color:



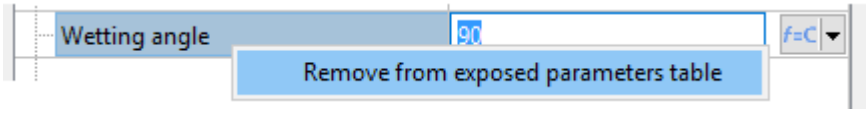
External parameters are highlighted in the **Properties** window using a background color

To add a parameter into the list of external parameters, select **Add to exposed parameters table** from the context menu of the parameter (after this a window will open, which allows you to change the parameter's name in the **Exposed parameters window**).



Adding a parameter to a table of external parameters

To remove a parameter from the list of external parameters, select **Remove from exposed parameters table** from the context menu of the parameter (after this a window will open, which requests you to confirm your decision).



Deleting a parameter from the table of external parameters

Commands **Add to exposed parameters table** and **Remove from exposed parameters table** are not available for parameters, which cannot be edited at all or at the current selection of other parameters. If necessary, click the **Apply** button in the **Properties** window to make these commands available.

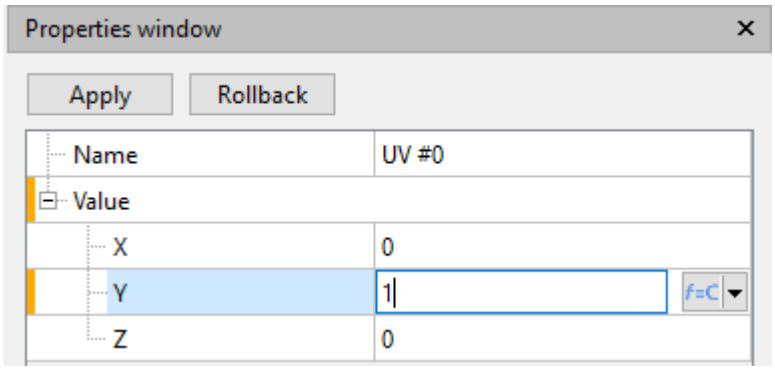
Hot keys used in the Properties window

For your convenience and speeding up your work, you can use the following hot keys in the **Properties** window:

Key	Description
↑ (Arrow Up)	Moves the cursor in the list of parameters up and down.
↓ (Arrow Down)	
Ins	Adds a new element into an array.
Del	Removes an array element or cleaning the whole array.
Enter	*In the column of names:* Pressing **Enter** expands or collapses a subtree of parameters; this displays or hides child elements. This key acts as clicking on the element ⊞ or ⊟ in the tree of parameters in the **Properties** window. *In the column of values:* Pressing **Enter** finalize editing of the selected parameter (this is *not* the action of the **Apply** screen button). When you press **Enter**, the new value of the parameter is checked for correctness and, depending on the result, either the mark of changing the parameter will change or the program will not accept the entered value (sometimes the program might output an error message with explanations).
Esc	Pressing **Esc** cancels the current editing of the selected parameter. This can be done only until either pressing **Enter** or navigating to another parameter.
+ *(in the number pad)*	Expands a subtree of parameters; this displays child elements. This key acts as clicking on the element ⊞.
-	Collapses a subtree of parameters; this hides child elements. This key acts as clicking on the element ⊟.
→ (Arrow Right)	Moves the selection of a screen button right or left.
← (Arrow Left)	
* *(in the number pad)*	Expands a subtree of parameters including all its nested subtrees.
F4	Expands the its list for selection from a combobox.
Alt+↓	
space bar	If a combobox is selected, pressing the space key expands the its list for selection from the combobox (it works same as **F4** or **Alt+↓**). If a screen button is selected, pressing the space key works as clicking this button.
the first letter of an option that can be selected from a combobox	Selects in a combobox the first option, which starts with the specified letter

Highlighting the changes

Parameters, which have been changed but not saved yet, and also their parent groups are highlighted with a color strip on the left.

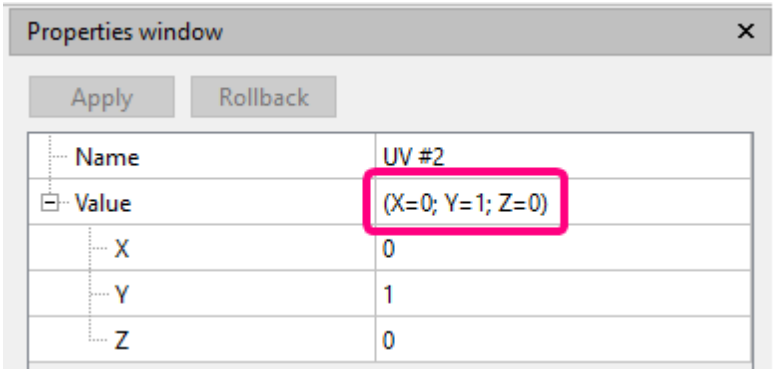


The property **Value > Y** has been changed but not saved yet, so fields **Value** and **Y** are highlighted with color strips on the left.

The color, which is used for highlighting the changed parameters, is specified in the **Property editor > Changes** setting in the basic settings of **Pre-Postprocessor**.

Note that this highlighting is also applied when method of specifying the parameter changes (for example, when you specify parameter's value as a constant instead of previous specifying it a formula).

Displaying structure of groups of properties



You can enable displaying, in fields that correspond to groups of properties, the contents of child elements as lines of the group structure in the "**Property=Value; ...**" format.

To do so, specify **Property editor > Inline structure fields = Yes** in the basic settings of **Pre-Postprocessor**.

You can copy the displayed group structure line into the clipboard. If a long line is not visible in full in the field, it will be displayed in its pop-up tip.

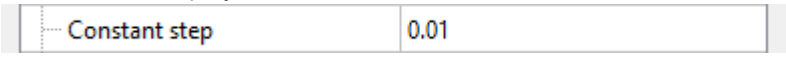
Applying the changes

Any your changes will take effect when you either click the **Apply** button or, if you don't click the button, after when you navigate to another element of the project tree.

8.1.9.1 Field for data input or selection

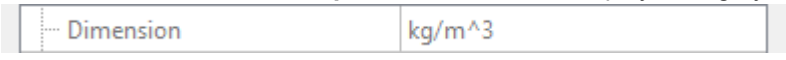
The input or selection field can be displayed with the following appearances:

- fields, which can be edited, are displayed in black Roman font:

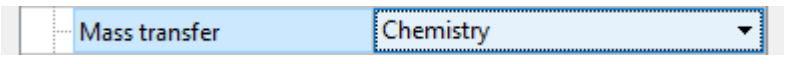


- fields, which can *not* be edited *at the current selection of other parameters*, are displayed in gray italics:

- fields, which can *not* be edited at all in this **Properties** window, are displayed in gray Roman font:



A selected field, which contains a value selectable from a of the drop-down list, is accompanied with an arrow symbol  :

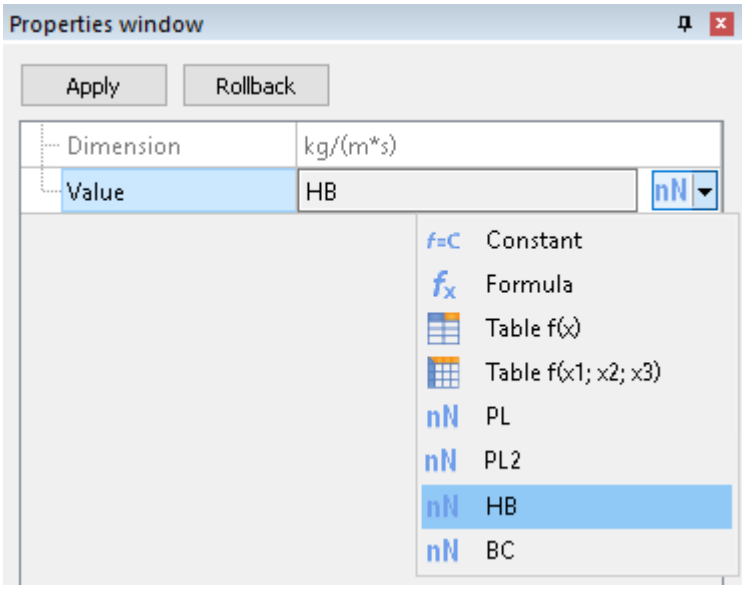


8.1.9.2 Expanded data input field

At the right side of an expanded data input field an arrow icon locates, which expands the list of possible value types.

There are the following value types:

- : a **Constant** is a numerical value, which is set directly in the data entry field
- : a vector **Constant**, which is set in the data entry field as three numbers separated by the ";" symbols
- or : a **Formula** for a scalar or vector value. The **Formula** is specified in the **Formula editor**.
- : a **Constant** from the **Substance Database**.
- : a **Table f(x)**, which is a function of one variable specified using the **Table editor**.
- : a **Table f(x1;x2;x3)**, which is a function of several variables specified using the **Table editor**.
- : an **SDTable f(T)**, which is a function of **Temperature** specified as a table in the **Substance Database**.
- : an **SDTable f(P,T)**, which is a function of **Pressure** and **Temperature** specified as a table in the **Substance Database**.
- , , , , or : use of a predefined formula (**Standard dependency**):
  - : formula of ideal gas (**Ideal gas law**)
  - : harmonic law
  - : specifying the **Viscosity** of a liquid by one of the models of non-Newtonian fluid: **PL** and **PL2** (*Power Law* and *Power Law 2*), **HB** (the *Herschel-Bulkley model*), **BC** (the *Bird-Carreau model*), see details in the section Theory > Substance properties > Liquid.
  - : specifying a value by a polynome or a linear function, see Theory > Substance properties > Gas
  - : specifying the **Viscosity** of a gas by the Blottner formula, see Theory > Substance properties > Gas
  - : specifying the **Thermal conductivity** of a gas by the $f(\mu, C_p)$ formula, see Theory > Substance properties > Gas
- : receive a value from a **User module** of the **Evaluator** type.



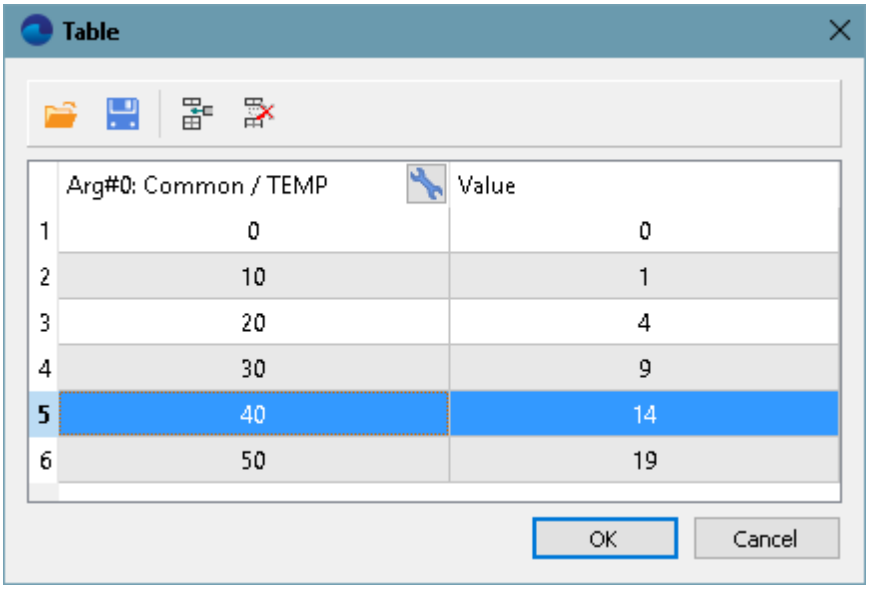
Specifying **Viscosity** of a non-Newtonian liquid by a formula from the Herschel-Bulkley model

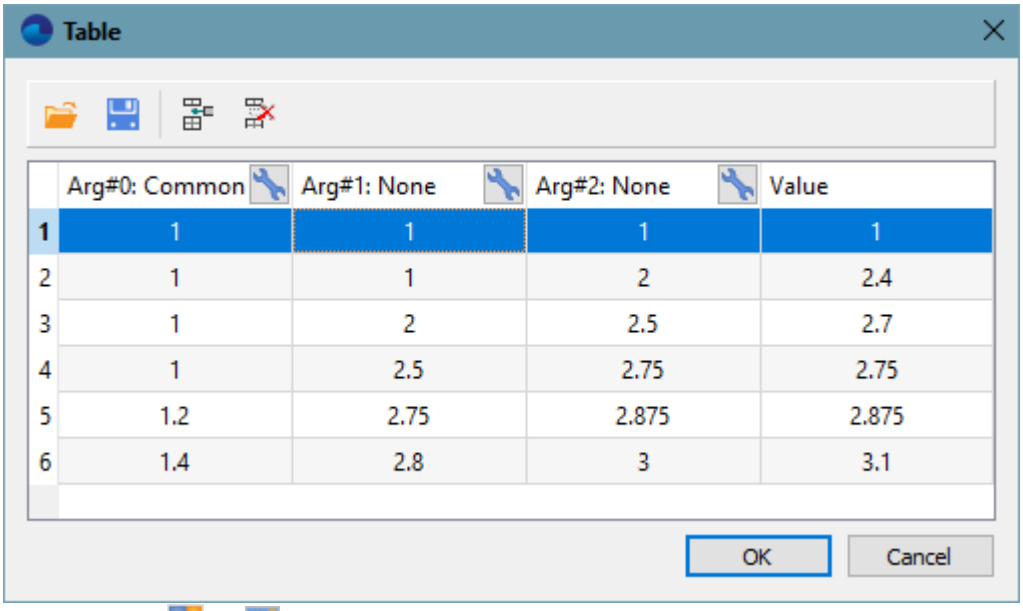
To enter the data, you have at first select the value type from the list that opens by clicking on , and then click on the value type's icon, which will appear to the right of the field (for constants you don't need to click on the icon or and can immediately enter the constant's value into the field).

After you clicking on the value type's icon, the program's reaction will be:

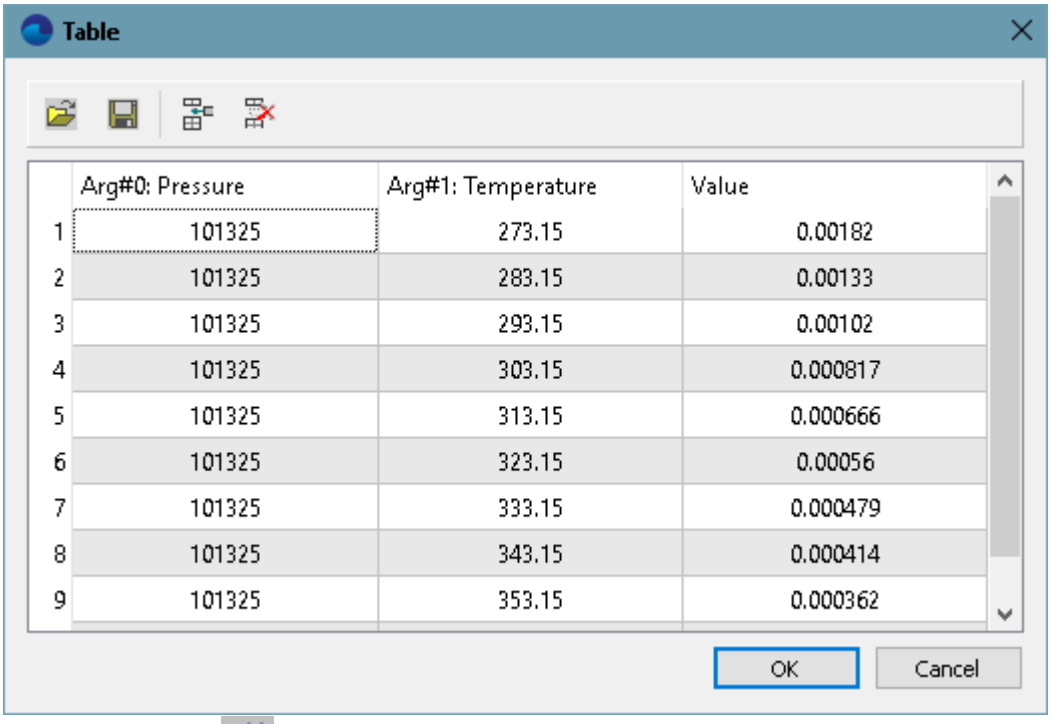
- after clicking the icon the cursor will move into the field for entering the constant's value, no dialog box will open.

- after clicking the icon  or  the **Formula editor** will open where you can enter or edit a formula.
- after clicking the icon  or  the **Table Editor** will open where you can specify the function f of one or several variables:

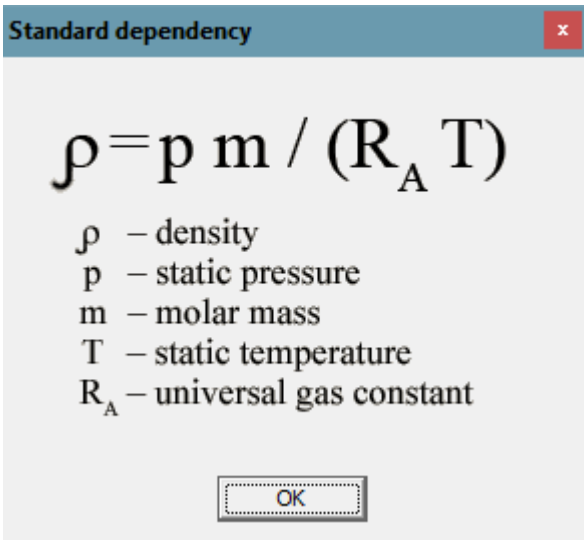




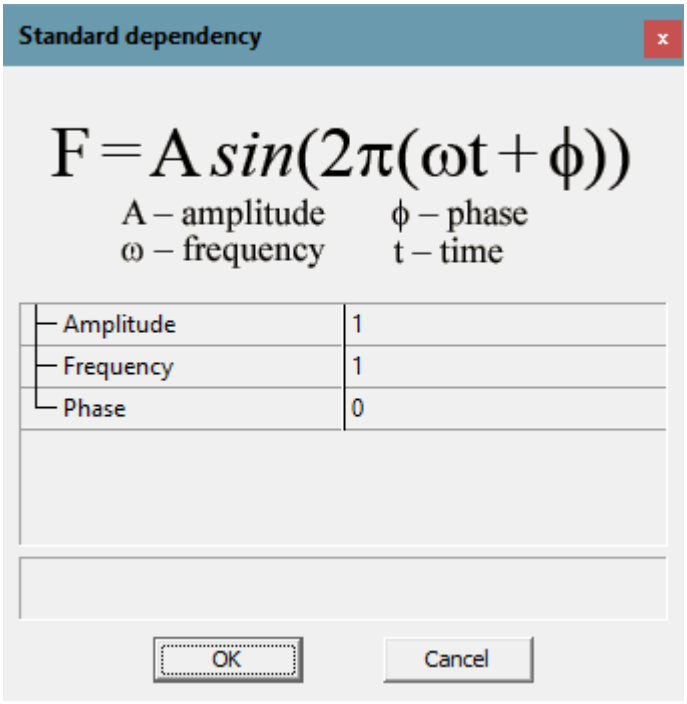
- after clicking the icon  or  the **Table** dialog box will open where you can *view* the values of a function, which has been defined by a table in the **Substance Database** (see also the section Defining a Substance, its Phases and Properties):



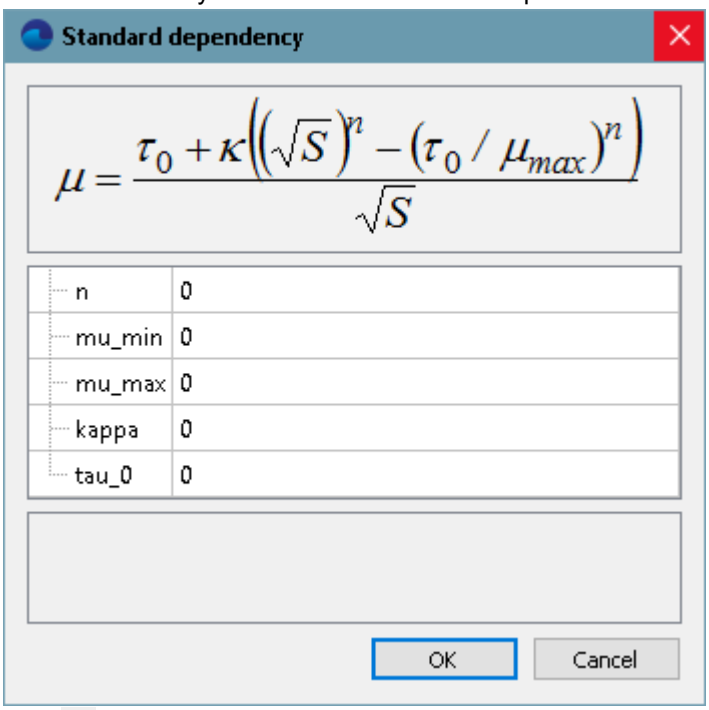
- after clicking on the icon pV the **Standard dependency** dialog box will open, informing you that the ideal gas formula is used:



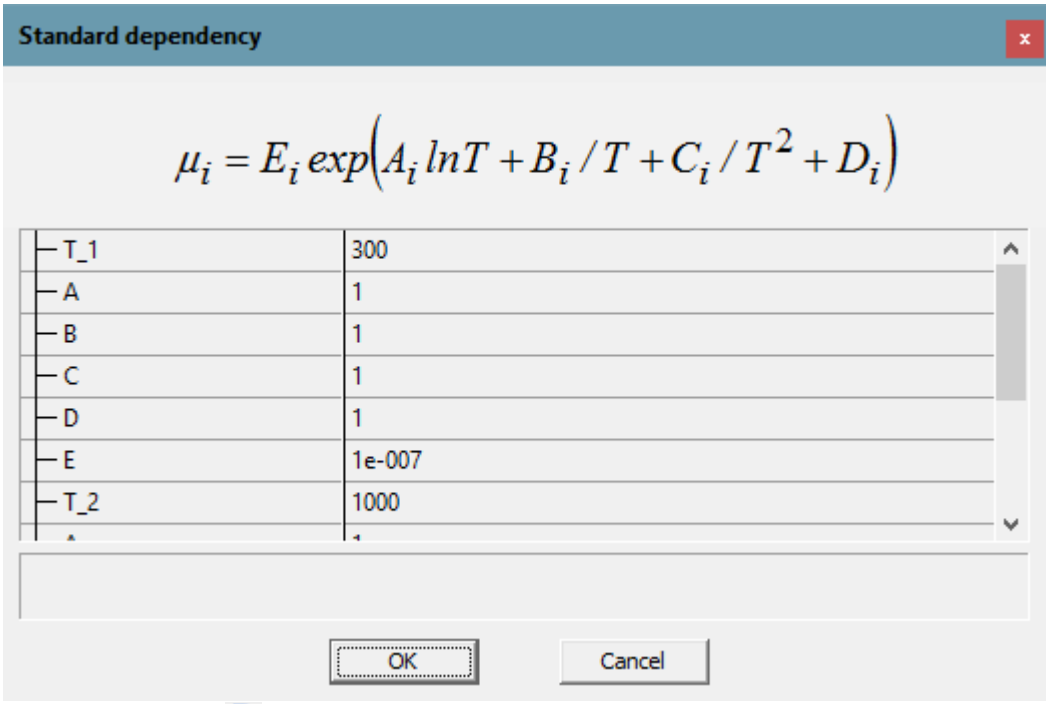
- after clicking on the icon the **Standard dependency** dialog box will open for specifying parameters of the Harmonic law:



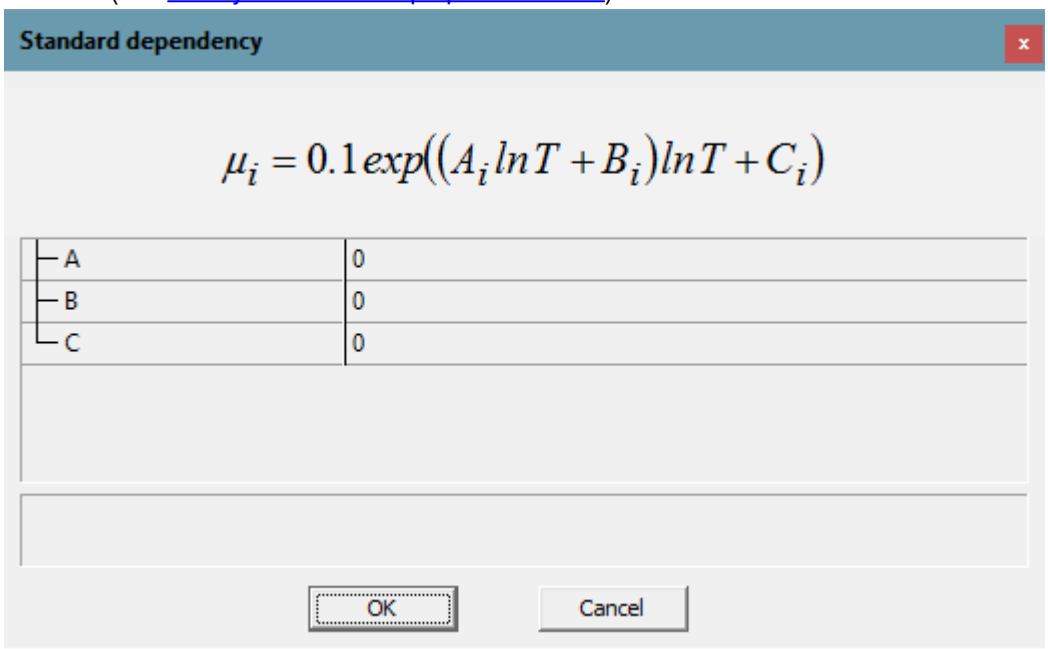
- after clicking on the icon nN the **Standard dependency** dialog box will open for specifying parameters of the selected model of non-Newtonian liquid (see details in the section Theory > Substance properties > Liquid), see an example of the Herschel-Bulkley model of non-Newtonian liquid below:



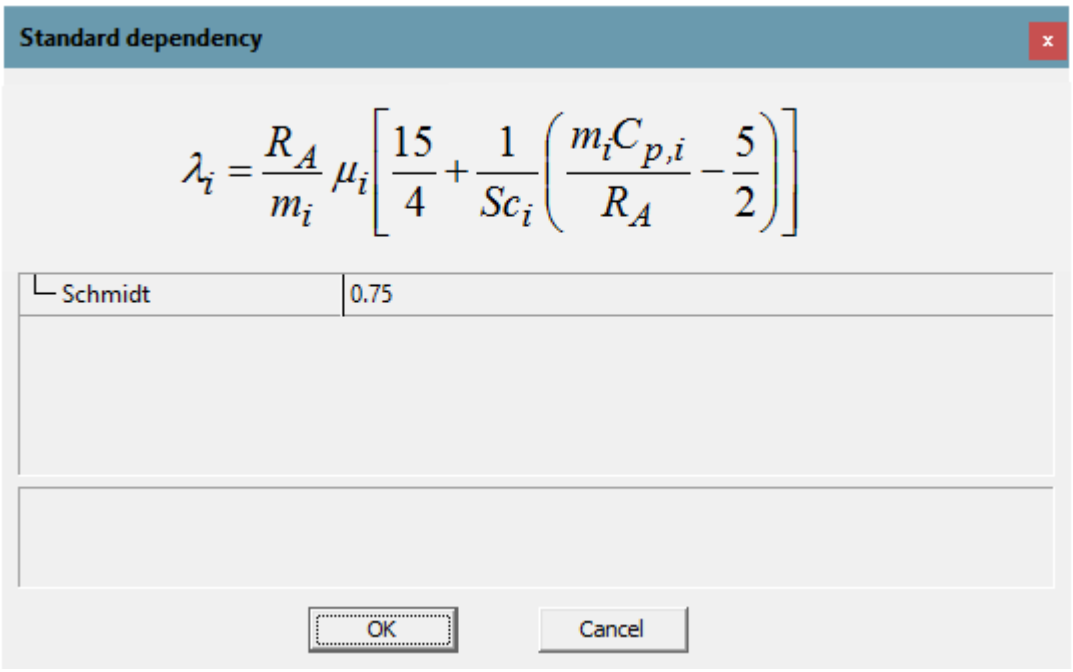
- after clicking on the icon P the **Standard dependency** dialog box will open for specifying a value by a polynome (see Theory > Substance properties > Gas):



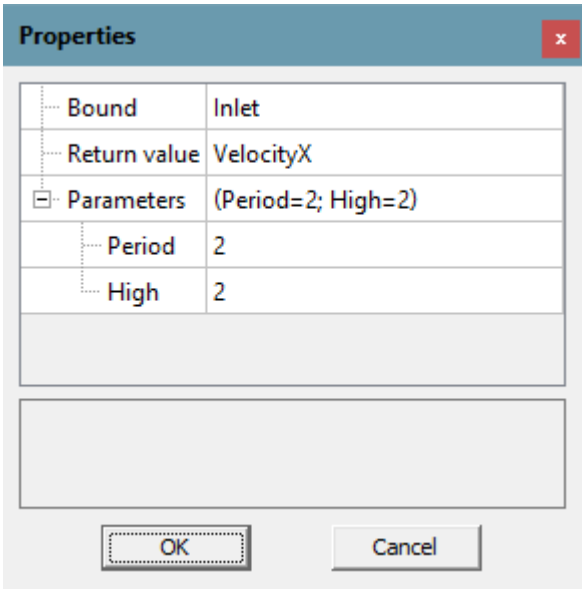
- after clicking on the icon B the **Standard dependency** dialog box will open for specifying a value by the Blottner formula (see Theory > Substance properties > Gas):



- after clicking on the icon C,μ the **Standard dependency** dialog box will open for specifying a value by the the $f(\mu, C_p)$ formula (see Theory > Substance properties > Gas):



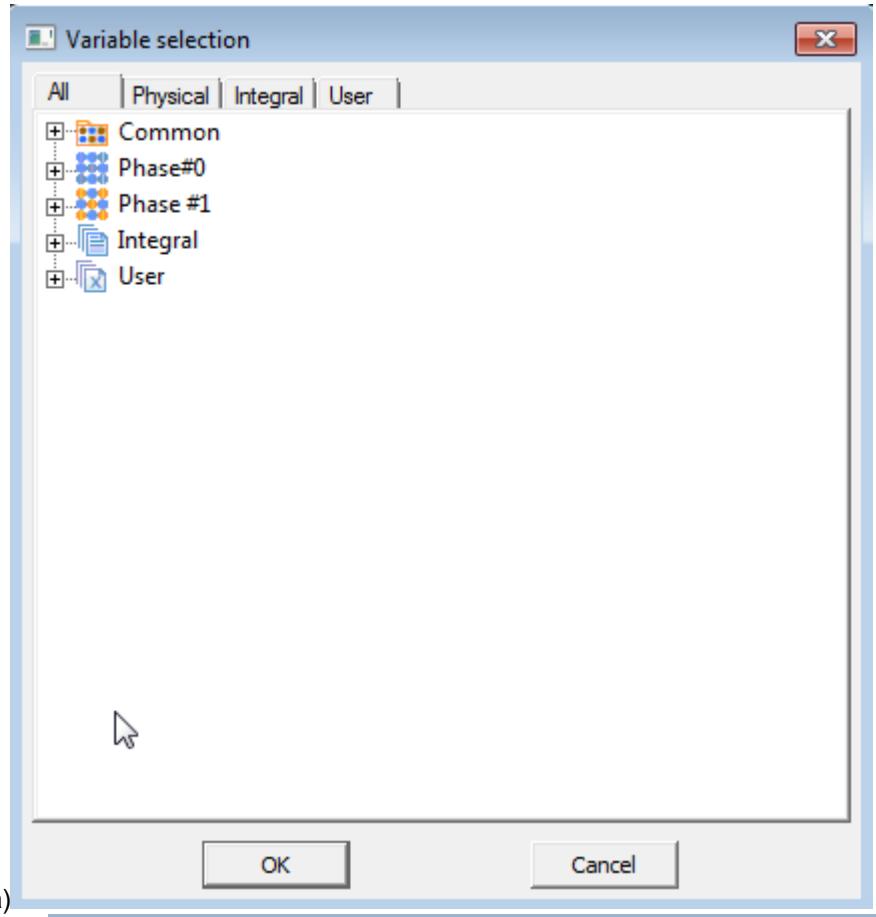
- after clicking on the icon  a dialog box of an appropriate **User module** of the **Evaluator** type will open:



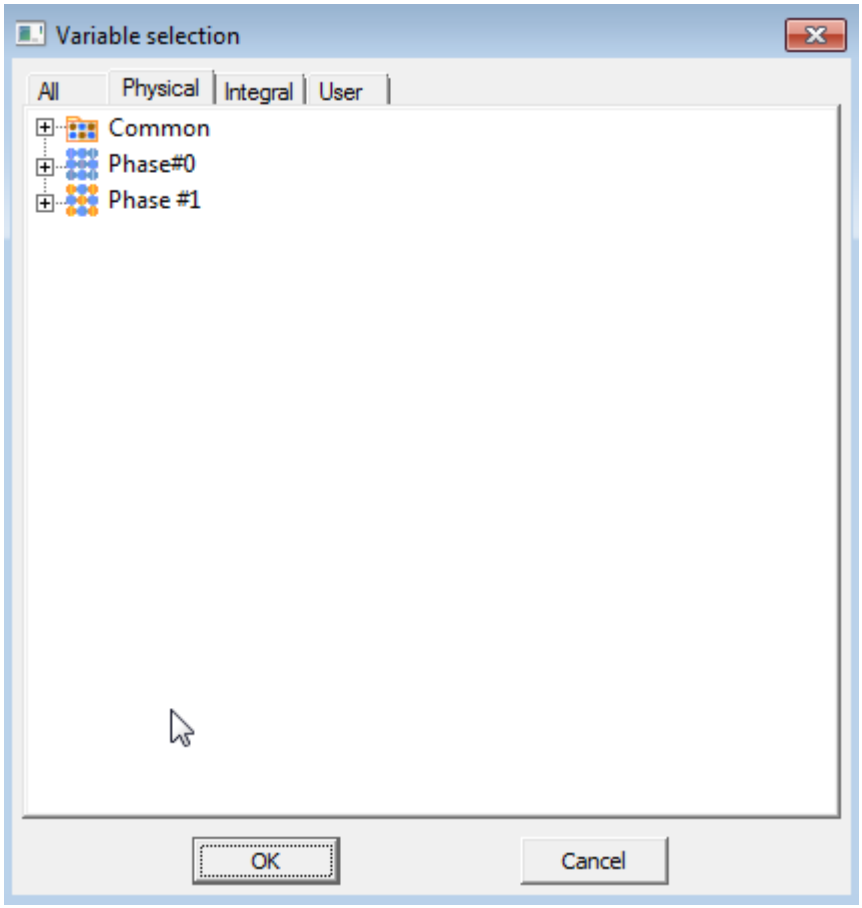
8.1.9.3 Defining a dependanse from a variable

The **Variable selection** dialog box is used to specify a variable, a function which is defined in the table setting.

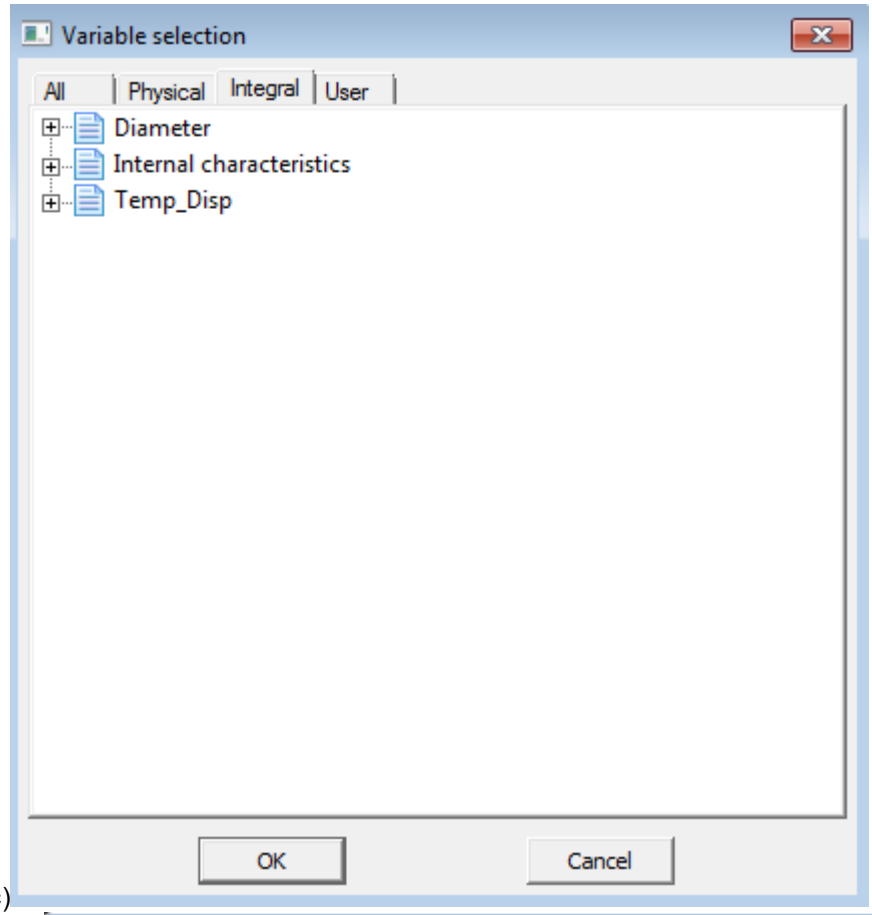
a)


Variable selection
All
Physical
Integral
User
Common
Phase#0
Phase #1
Integral
User
OK
Cancel

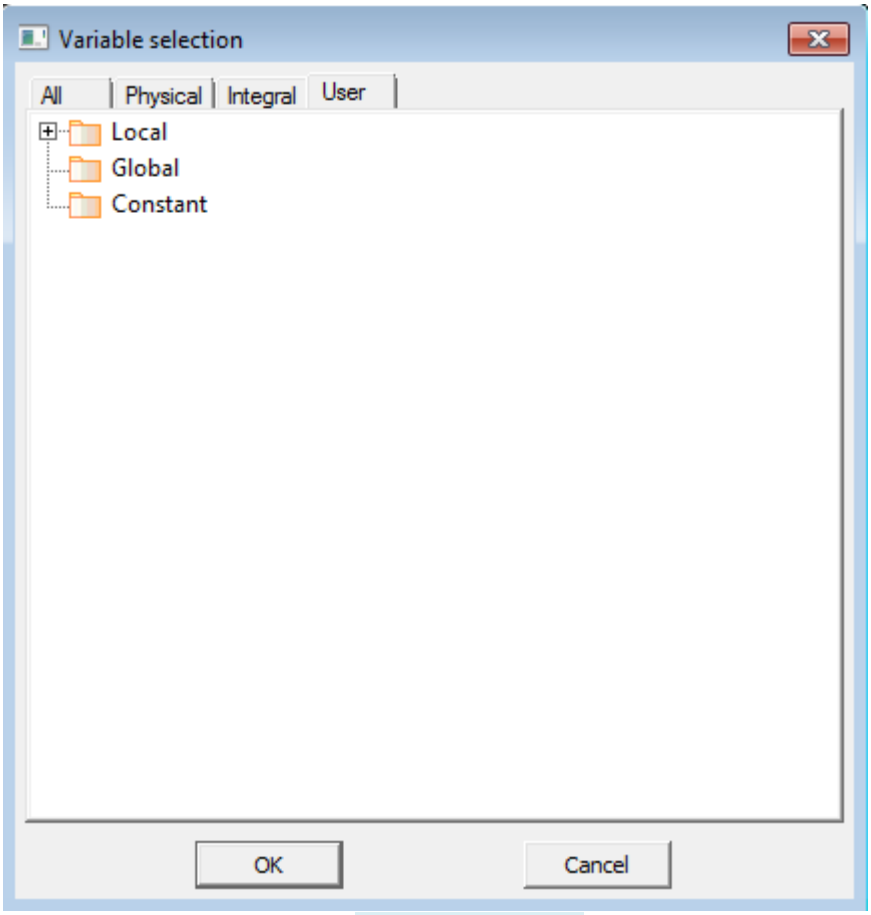
b)


Variable selection
All
Physical
Integral
User
Common
Phase#0
Phase #1
OK
Cancel

c)



d)



Dialog box **Variable selection**, tab:
a) **All**; b) **Physical**; c) **Integral**; d) **User**

The dialog box consists of four tabs:

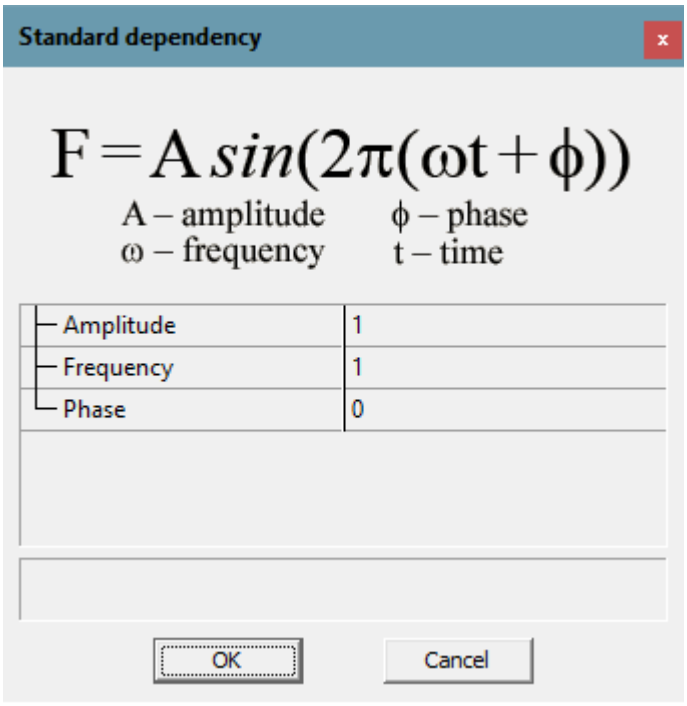
- **All**, which displays all the variables
- **Physical**, which displays the variables that are available in phases in accordance with the set of physical processes
- **Integral**, which displays the integral variables
- **User**, which displays the user-defined variables

To set a variable, select it on one of the tabs, and then click **OK**.

8.1.9.4 Harmonic function

The **Harmonic low** (harmonic function) is a standard dependency for a variable. The harmonic function in defined by the following formula:

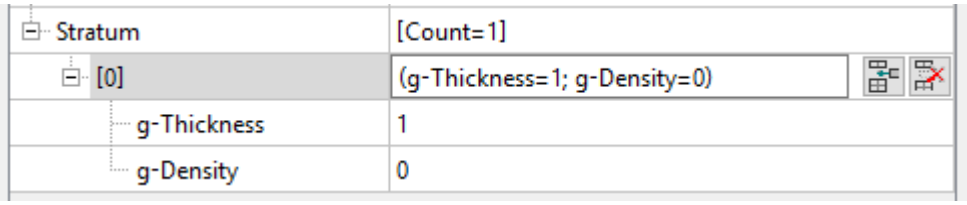
$$F = \mathrm{A}\,\sin(2\pi(\omega t + \phi))$$



The specified parameters of the harmonic low:

Component	Letter in the formula
Amplitude	A
Frequency	ω
Phase	ϕ

8.1.9.5 Field for defining an array

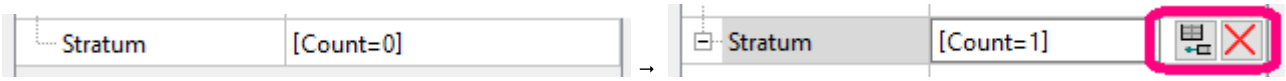


An *array definition field* is intended to define the list of similar numbered elements with:

- the same sets of properties
- similar child elements if any

Elements of an array are displayed in the **Properties** window as a number in square brackets (**[0]**, **[1]**, **[2]**, ...).

When a list definition field is selected (using either ↑ (Arrow Up) or ↓ (Arrow Down) key on the keyboard or by clicking it by your mouse), on the field's right side screen buttons and appear that allow you to do some actions with the list of elements:



Screen button	Description
Append item to the array (Ins)	Adds a new element to the end of the array. The new element is visible when you expand the tree structure by clicking the sign. You can also use the **Ins** key.
Clear the array (Del)	Clears the array (the program will prompt you to confirm this action). You can also use the **Del** key.
Insert item before current (Ins)	Adds a new element into the array, above the selected element. You can also use the **Ins** key.
Delete this item (Del)	Deletes the selected element (the program will prompt you to confirm this action). You can also use the **Del** key.

8.1.9.6 Parameters for defining a palette

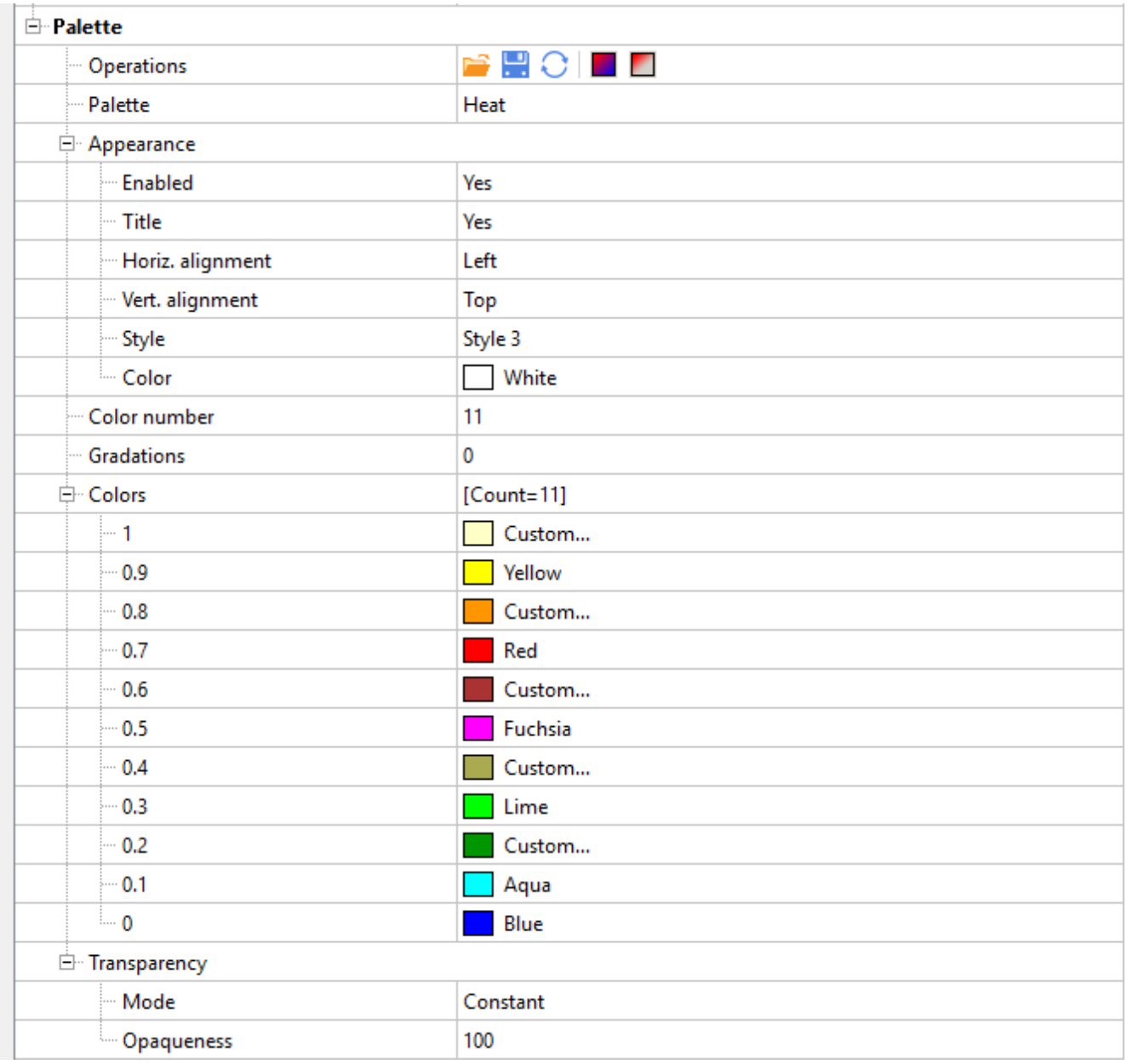
In some cases the **Properties** window contains parameters for defining a **Palette**.

Particularly, a palette can be specified in properties of:

- background of the **View** window
- visualization **Layers**

Palettes can be saved in text files in XML format. The program's distribution package includes several files with preset palettes (these files are the standard library of palettes) that locate in the subdirectory `Palette` in the installation directory of *FlowVision*. You can also save in files your own **Palettes** for future use.

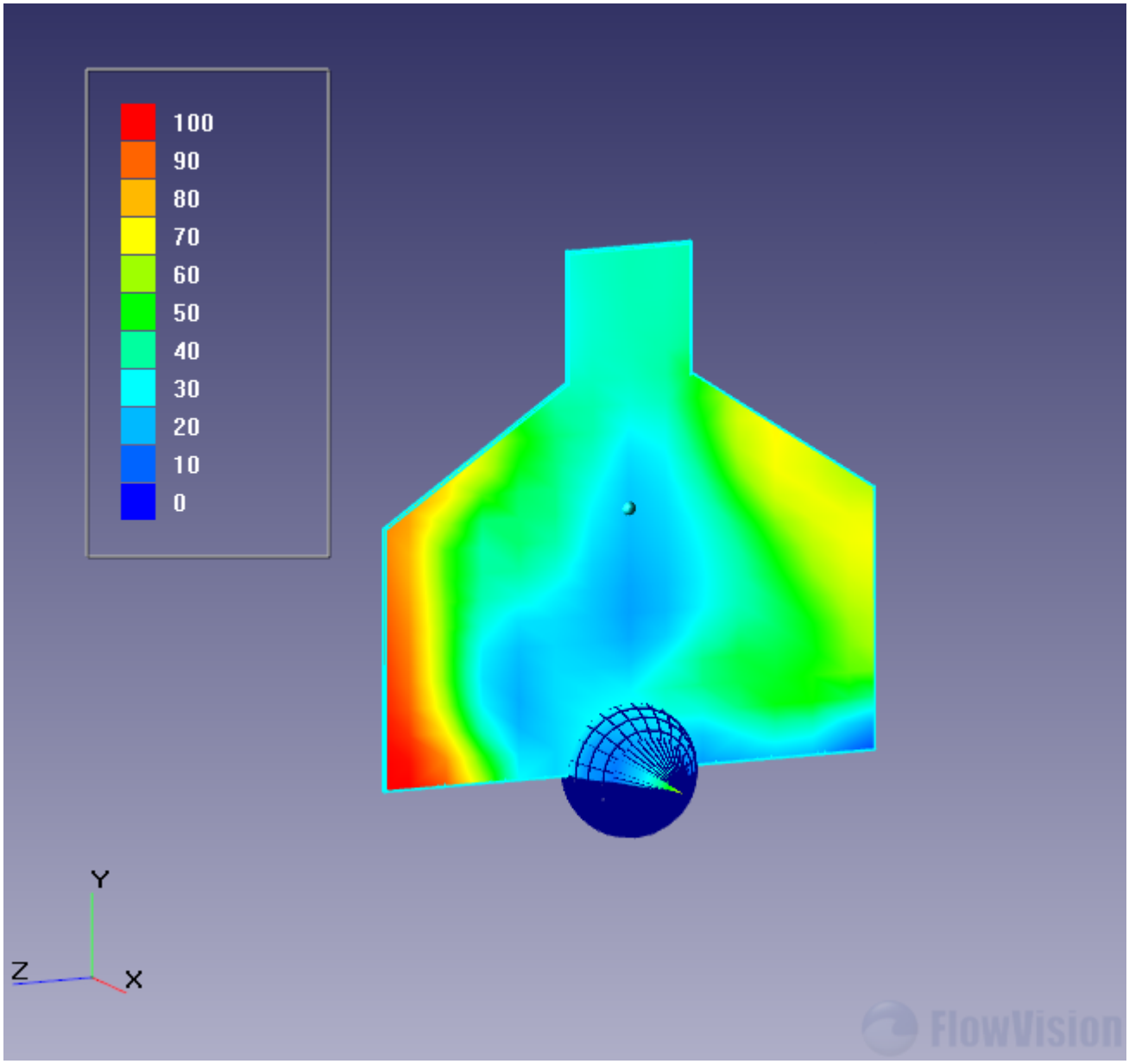
Parameters and operations with **Palettes** are specified in the **Properties** window in the **Palette** group of parameters:



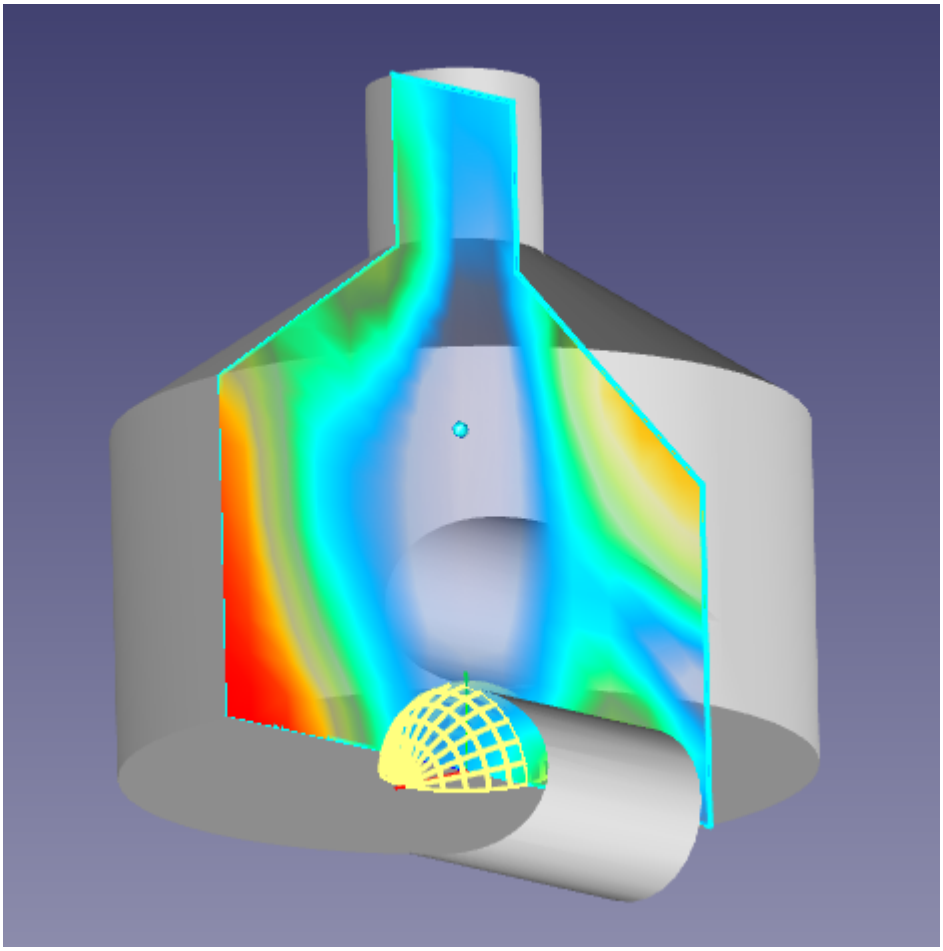
Parameter	Description
Palette	Group of parameters that specify settings of the **Palette**, which is used, and buttons for operations withe **Palettes**
Palette > Operations > (Load palette from file)	Loading a palette, which will be used, from a `.fvpal` file. In the case if the `fvpal` file is damaged or contains incorrect data, loading of the **Palette** will fail and an error message will be displayed in the **Log** window ("**Error read file**").
Palette > Operations > (Save palette to file)	Saving the current palette to a `.fvpal` file. A dialog box for access to files will open where you have to specify location and name of the file, in which the palette will be saved. If you specify the location as a directory, which isn't included in the list of user libraries of palettes, the program prompt you to add the new location in this list ("**Do you want to add this folder in palette library?**"). In the case of your positive answer, the program will automatically add the new directory to the value of the **User libraries > Palettes** parameter in the basic settings of **Pre-Postprocessor**. The program doesn't allow saving palettes in system directories and in the in the installation directory of *FlowVision* including subdirectories.

Parameter	Description
Palette > Operations > (Reload palette list)	Reloading the list of palettes that are available for selection from the drop-down list of values of the **Palette > Palette** parameter (see below).
Palette > Operations > (Generate gradient palette)	Generating colors of the new palette as gradient from the first to the last color of the current palette. This operation changes all middle colors of the current palette.
Palette > Operations > (Generate transparency gradient)	Generating values of opacity of colors as gradient from the first color to the last color. Generated percentages of opacity fill list **Palette > Transparency > Opacity > (number)** (see below).
Palette > Palette	Selecting a **Palette** from the standard library of palettes or from a user library of palettes. To obtain access to all palettes stored there you might have to click the button **Palette > Operations > (Reload palette list)** before (see above). The standard library of palettes is supplied complete with the *FlowVision* program. Files `.fvpal` of the standard library of palettes locate in the subdirectory `Palette` in the installation directory of *FlowVision*. Location of user libraries of palettes is set by the **User libraries > Palettes** parameter in the basic settings of **Pre-Postprocessor**.
Palette > Appearance > Enabled	This parameters specifies if the image formed in the **View** window will contain the legend of the palette. Possible options: **No** | **Yes**. (see the illustration after this table)
Palette > Appearance > Title	Use the layer's **Name** as a title for the palette's legend (see an illustration). Possible options: **No** | **Yes**.
Palette > Appearance > Vert. alignment	• **Top** – display panel in the upper half of the **View** window • **Bottom** - display palette on the bottom half of the **View** window
Palette > Appearance > Horiz. alignment	• **Left** – display palette on the left half of the **View** window • **Right** – display palette on the right half of the **View** window
Palette > Appearance > Style	• **Style 1** – discrete palette, inscriptions on the background color • **Style 2** – continuous palette, inscriptions on the background color • **Style 3** – discrete palette, inscription on the side • **Style 4** – continuous palette, inscription on the side
Palette > Appearance > Color	The choice of color lettering for styles 3 and 4
Palette > Appearance > Number of labels	The number of color gradations in palette for styles 2 and 4
Palette > Color number	Number of reference colors in the palette
Palette > Gradations	Number of gradations between the reference color in the palette. When **Gradations=0**, transitions between colors are most smooth.
Palette > Colors > *(number)*	List of reference colors in the palette
Palette > Transparency > Mode	• **Constant** – all the colors of the color contours and isosurfaces equally opaque • **Variable** – some color color contours and iso-surfaces can be defined as partially or completely transparent (see illustration)
Palette > Transparency > Opaqueness	Opacity (in %) of all colors and color contours isosurface (if **Transparency > Mode = Constant**)
Palette > Transparency > Opaqueness > *(number)*	List of opacity (in %) of individual colors in **Layers**, for which this functionality is implemented. This parameters are available when **Transparency > Mode = Variable**. This list can be filled manually or generated automatically by clicking on the **Palette > Operations > (Generate transparency gradient)** icon, see above.

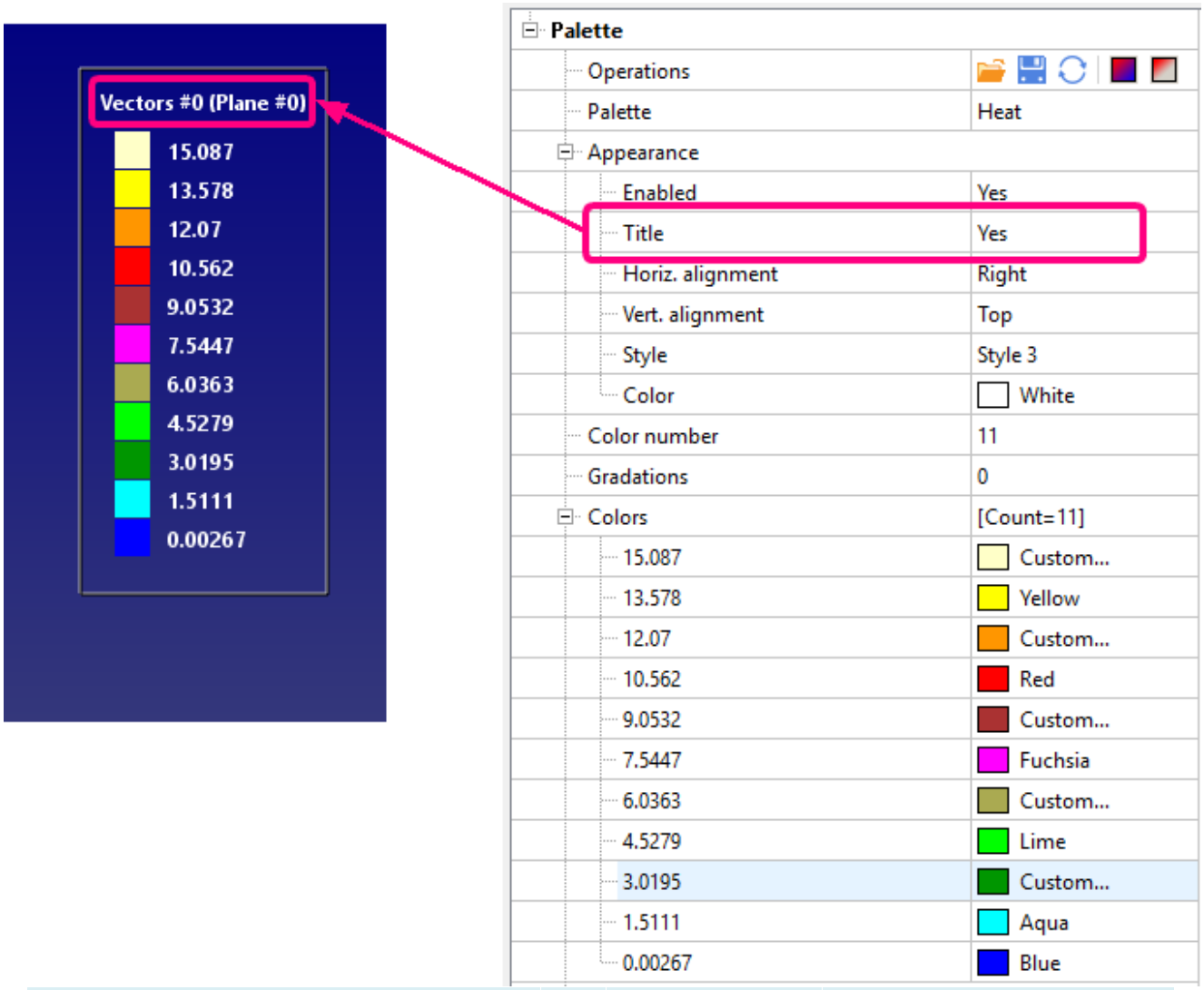
Illustrations



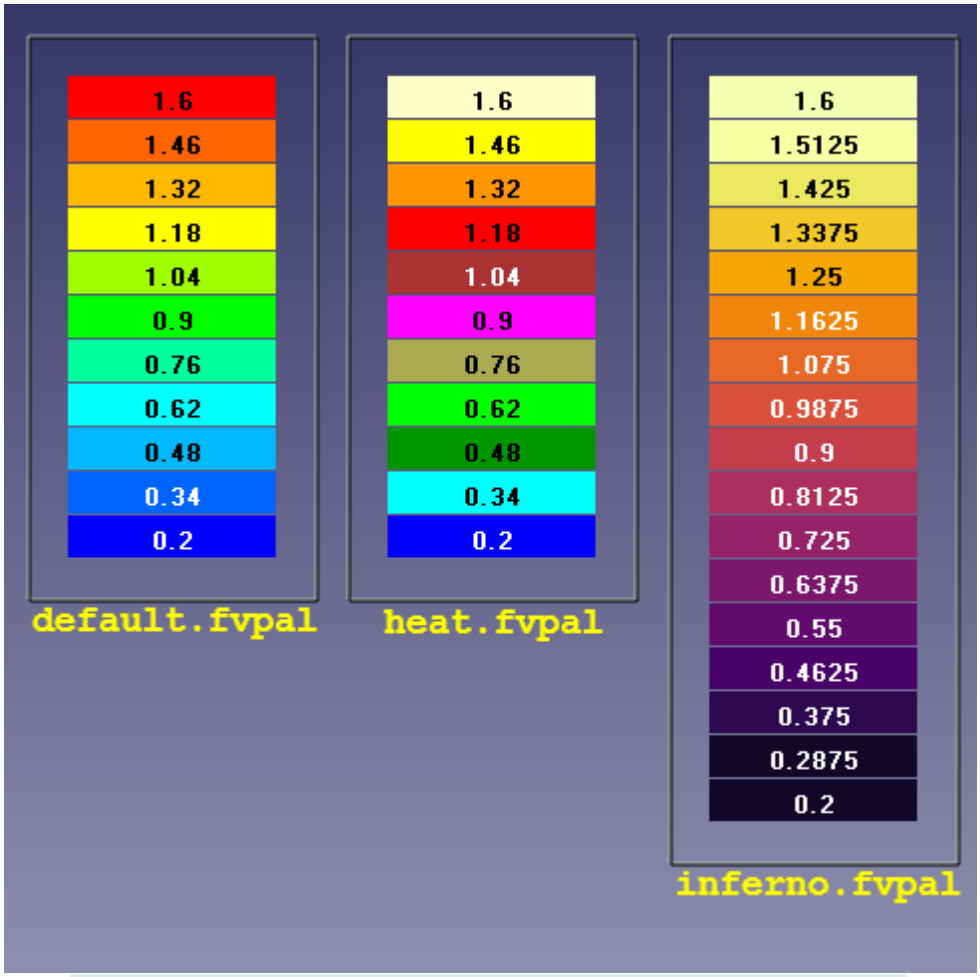
In group settings **Palette > Appearance**, you can specify the location and appearance of the palette's legend, which is placed over the image in the **View** window



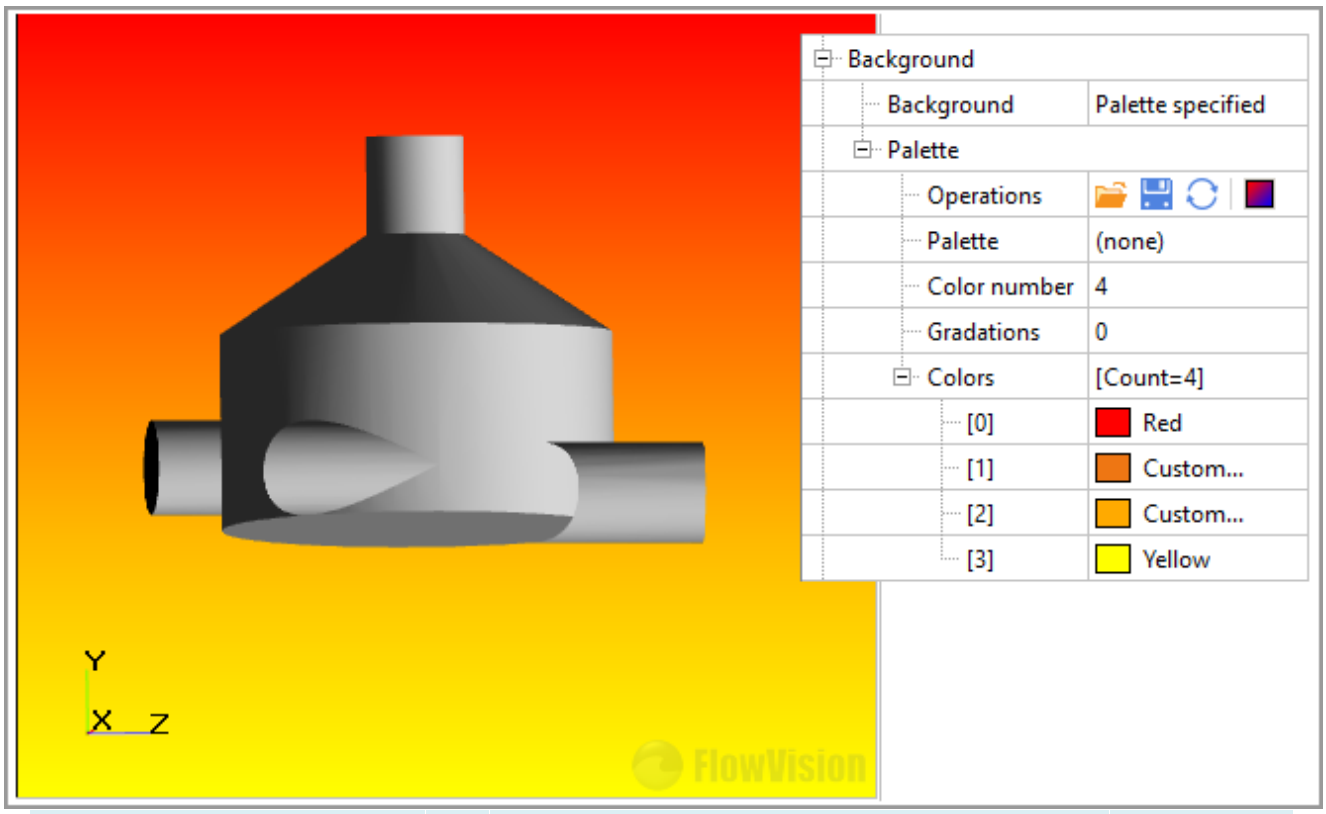
Some colors of color contours and isosurfaces can be defined as partially or fully transparent with the parameters of the group **Palette > Transparency**



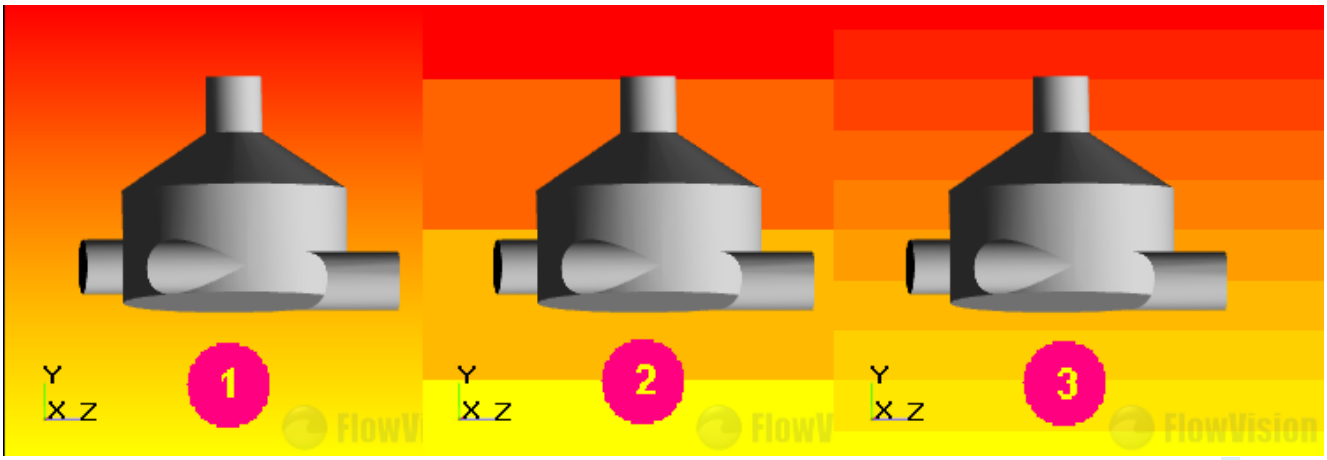
The palette's legend can include the name of the **Layer**. To enable this, set **Palette > Appearance > Title = Yes**.



Examples of preset palettes that are supplied complete with the program



Example of the background of the **View** window specified by a palette that contains 4 colors; **Gradations = 0**



The **Palette > Gradations** parameter specifies smoothness of color transitions on the background (1 - **Gradations = 0**; 2 - **Gradations = 1**; 3 - **Gradations = 3**)

8.1.10 Window «Monitor»

The **Monitor** window is used to display general parameters of the project's calculation.

To display the **Monitor** window, check **View > Monitor window** in the main menu.

To hide the **Monitor** window, uncheck **View > Monitor window** in the main menu or click "**×**" in the upper right corner of the window.

Notations

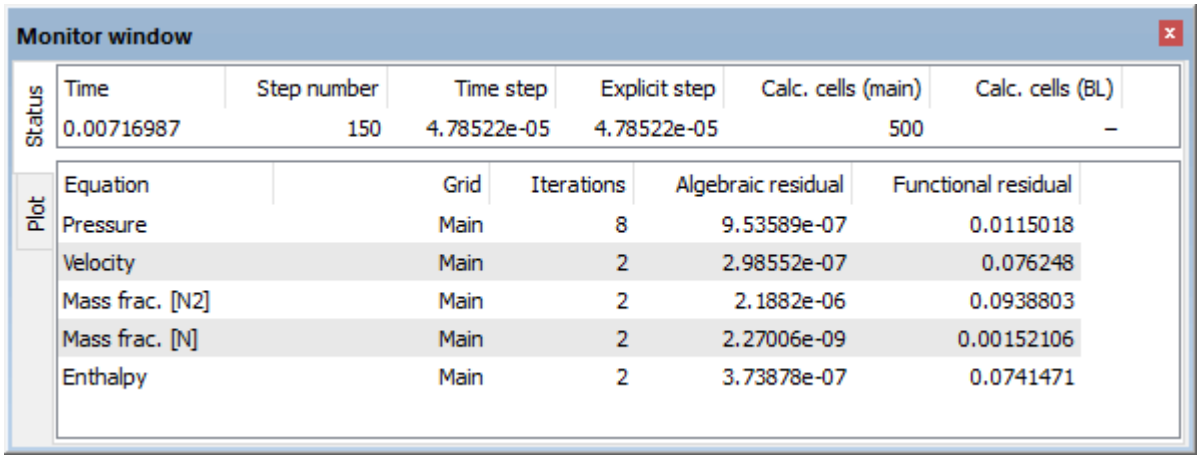
Designation	Value of	Name in *FlowVision*	Dimension
n	The time step's number	**Step number**	
τ	The time step	**Time step**	[s]
T	Time of the calculation	**Time**	[s]
f^n	Value of the variable at the n^{th} step		[f]
V	The calculated volume		[m^3]
$R_A(t^n)=\frac{\left\|b-Af^{n+1}\right\|_2}{\left\|b\right\|_2}$ where $Af^{n+1}=b$ is the system of algebraic equations that are solved iteratively for the variable f at the current time step	Algebraic residual	**Algebraic residual**	
$R_{norm}(T)=\frac{R(T)}{\max\limits_{0<t<T} R(t)}$ where $R(T)=\sqrt{\frac{1}{V}\int_V\left(\frac{f^{n+1}-f^n}{\tau}\right)^2 dV}$	Functional residual	**Functional residual**	[f]

User interface

The **Monitor** window contains two tabs located on its left edge:

- **Status**
- **Plot**

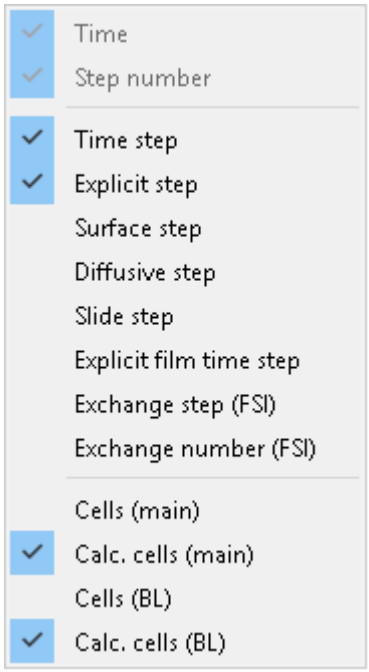
The "Status" tab



Window **Monitor**, tab **Status**

The **Status** tab contains two tables.

The top table displays the current time, the number of the current time step, and other data, list of which is tuned from the context menu opened when you right-click the table's header:



The lower table displays properties of equations, *which are solved using an implicit method*. For each main calculated variable (when computing a motion they are **Speed** and **Pressure**, when calculating the heat transfer this is **Temperature**) characteristics of the last time step is displayed (these are characteristics of systems of linear equations to be solved at the current time step). Each calculated variable has its own system of linear equations and a line in this table. Also number of iterations taken for convergence of the solution of algebraic equations, maximum residual in the computational domain, and functional residual are displayed.

You can copy data from the table's cell using hot keys **Ctrl+Ins** and **Ctrl+C**.

The data displayed in the *upper* table in the **Status** tab of the **Monitor** window:

Column's name (displaying the columns is tuned in the context menu, which opens when you right-click on the table's header)	**Description**
Time	The current time, T
Step number	The number of the current time step, n
Time step	The time step τ [s], see section Time step
Explicit step	The explicit convective time step, $\tau_{expl,\ conv}$ *)  At the beginning of the iteration to calculate all time steps (explicit, surface, diffusion), and on the basis of user-defined properties CFL calculated time step τ. Next there is a movement of the moving bodies and the free surface. After this explicit convective time step $\tau_{expl,\ conv}$ is *calculated again* with the changed position of the moving bodies and the free surface, while τ is not calculated again. The program displays this re-calculated value $\tau_{expl,conv}$.
Surface step	The surface of the time step, τ_{surf} *)
Diffusive step	The diffusion time step, τ_{diff} *)
Slide step	The time step for a sliding surface, τ_{slide} *)

Column's name (displaying the columns is tuned in the context menu, which opens when you right-click on the table's header)	Description
Explicit film time step	The explicit time step for spreading the liquid film in simulations of the dispersed phase crystallization.*) If a quasi-stationary process is implemented in simulating of crystallization of the dispersed phase (when **Film CFL** > **1** is specified), this column will display the explicit time step of the film's motion from the last iteration of the film's spreading-icing cycle.
Exchange step (FSI) **Exchange number (FSI)**	The exchange step and number for co-simulations that are solved jointly with the *Abaqus* software
Cells (main)	The total number of cells in the main computational grid
Calc. cells (main)	Number of computation cells in the main computational grid
Cells (BL)	The total number of cells in the boundary layer grid
Calc. cells (BL)	Number of computation cells in the boundary layer grid (BL)

*) see section Time step.

The data displayed in the *lower* table in the **Status** tab of the **Monitor** window:

Column's name	Description
Equation	List of main calculated variables
Grid	The grid (either the main grid or the BL grid), on which the variable is calculated
Iterations	The number of iterations that were done to obtain convergence of the solution of algebraic equations. The number of iterations in the calculations of steady flow, which are done using the relaxation method, should decrease. If the number of iterations is stable large, this means that the calculated process is non-stationary. The number of iterations is limited depending on the applied algebraic solver (*Aggregation AMG*, *Selective AMG*, *TParFBSS*). This limitations are set by parameters **Max iter. number A**, **Max iter. number S**, and **Max iter. number T** in properties of the element **Advanced Settings** in the **Solver** tab of the project tree).
Algebraic residual	$R_A(t^n)$, which is ratio of residual norm of the algebraic equation's solution to the norm of the the right-hand side of the equation, achieved during solving the algebraic equations. This value is to be less than the value of the **Rel. tolerance** parameter in properties of the element **Advanced Settings** in the **Solver** tab of the project tree. Monitoring of values displayed in the **Algebraic residual** column is restricted by confirmation only that all systems of algebraic equations are being solved consistently with the given accuracy.
Functional residual	$R_{\text{norm}}(t^n)$, which is the maximum at an estimated rate of change in the field of basic calculated variable

The "Plot" tab

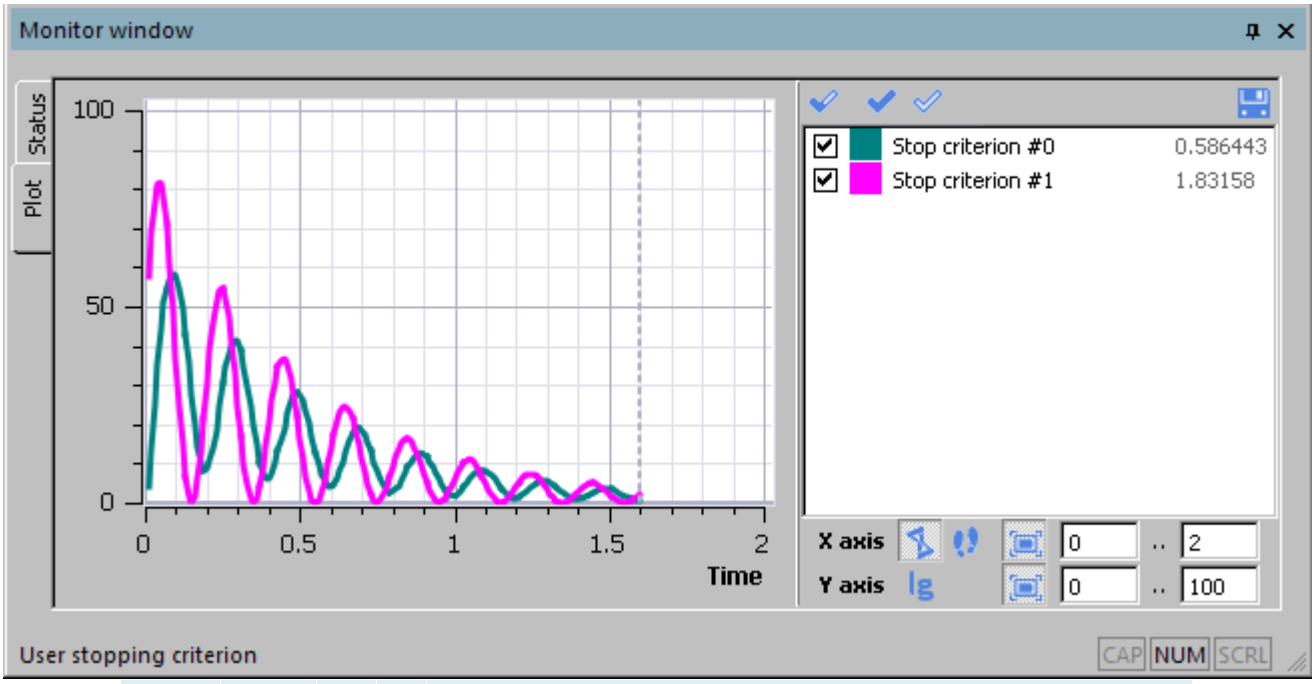
The **Plot** tab displays plots of functional residuals of calculated variables and values of user variables (characteristics), for which **Stop criteria** are defined.

To display a plot of a variable of interest in **Pre-Postprocessor** or **Viewer**, you have to, before starting the computation, create:

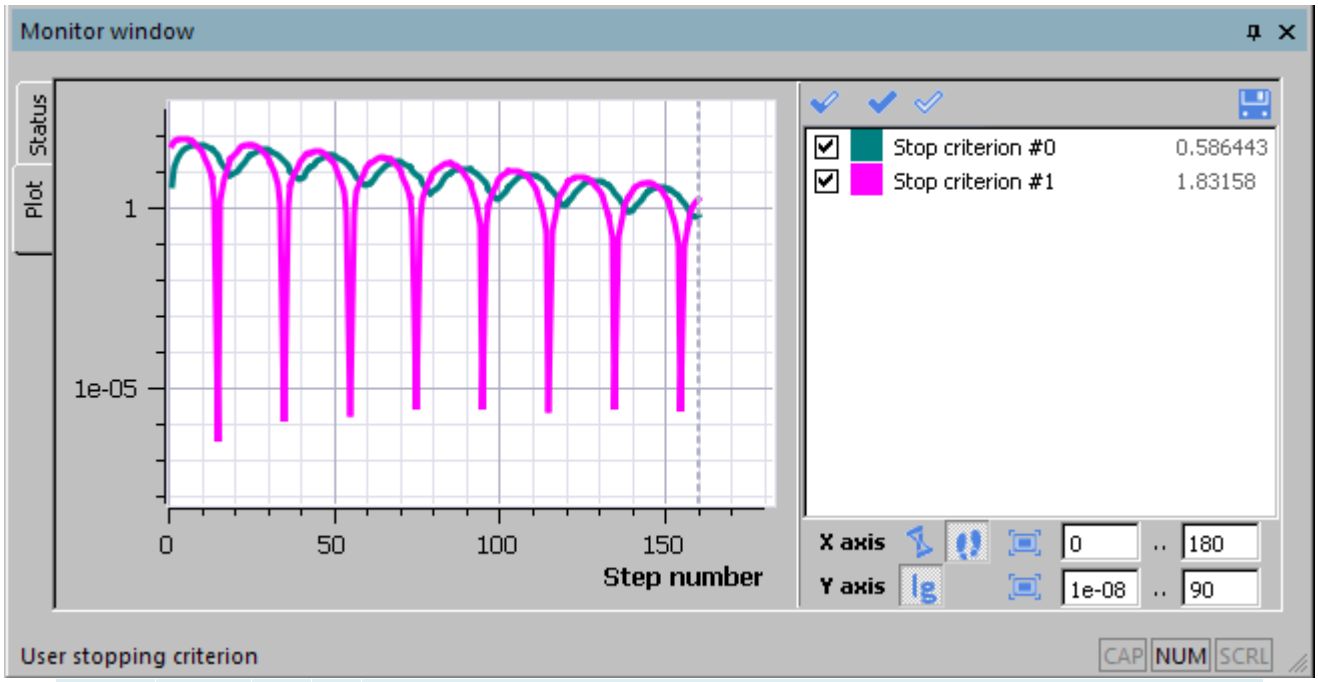
- in the **Preprocessor** tab: a **User variable** (**Global** or **Constant**) or **Characteristics** containing the appropriate value
- in the **Solver** tab: a **Stop criterion** based on the appropriate user value

Along the axis of abscissas, either time or number of steps is marked off. Along the axis of ordinates, value of the functional residual or the value of a user variable is used.

The current time step is displayed on the plot as a vertical dashed line.

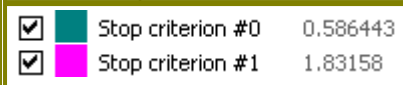


Window **Monitor**, tab **Plot** (dependence on the time, a uniform scale at the axis of ordinates)



Window **Monitor**, tab **Plot** (dependence on the step number, a logarithmic scale at the axis of ordinates)

The list of the plot's lines (with their colors and values on the last step) and other interface elements for tuning the plot are displayed to the right of the plot:

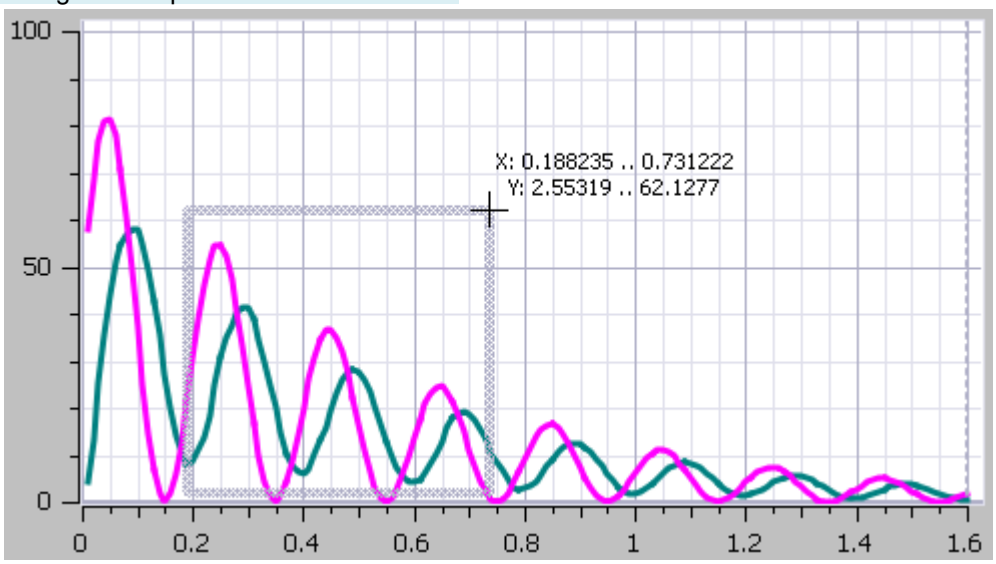
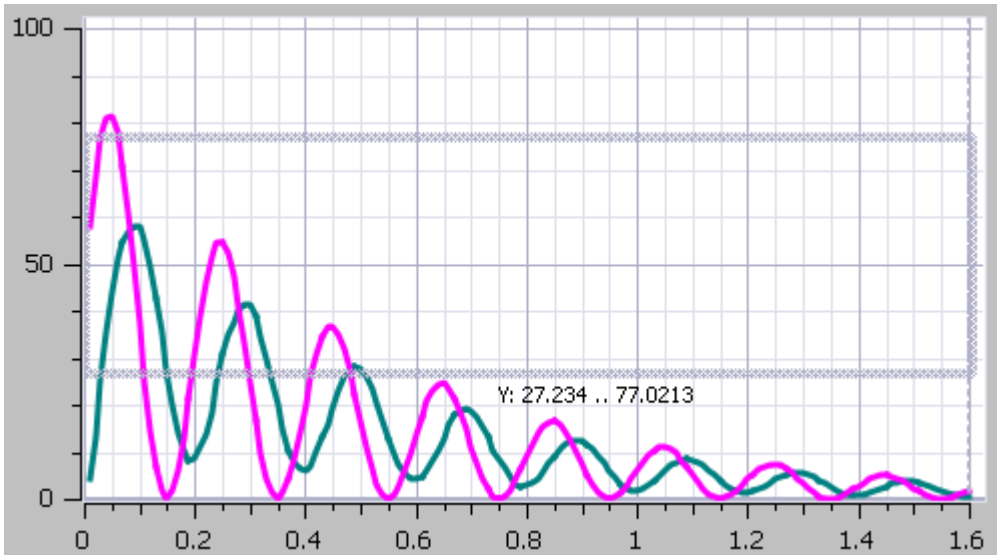
Interface elements	Description
List of the plot's lines	
	Use the checkboxes to select the lines that will be displayed on the plot. The lines, which are not displayed, are shown in the list with a faded font color. The colored box on the left indicates the color of the line. On the right the values on the last time step are displayed. When the computation is stopped and name of the plot line or value is selected, pressing **Ctrl-C** keys cause copying the name or value into the *Windows* clipboard.
Bulk change of displaying or hiding the plots	
	**Toggle selection** of displayed and hidden plots
	**Select all** plots (all plots will be displayed)
	**Clear selection** of plots (all plots will be hidden)
*Settings for the axis of abscissas ("**X axis**")*	
	Selection a variable for the axis of abscissas: • time • step number
	Automatic tuning the scale along the axis of abscissas (so the whole plot will fit in the horizontally)
	These fields are use to enter the plot's range along the axis of abscissas manually. After entering the data, click the **Enter** key on your keyboard.
*Settings for the axis of ordinates ("**Y axis**")*	
	Toggling the logarithmic scale at the axis of ordinates. When this button is released, a uniform scale is used.
	Automatic tuning the scale along the axis of ordinates (so the whole plot will fit vertically)
	These fields are use to enter the plot's range along the axis of ordinates manually. After entering the data, click the **Enter** key on your keyboard.

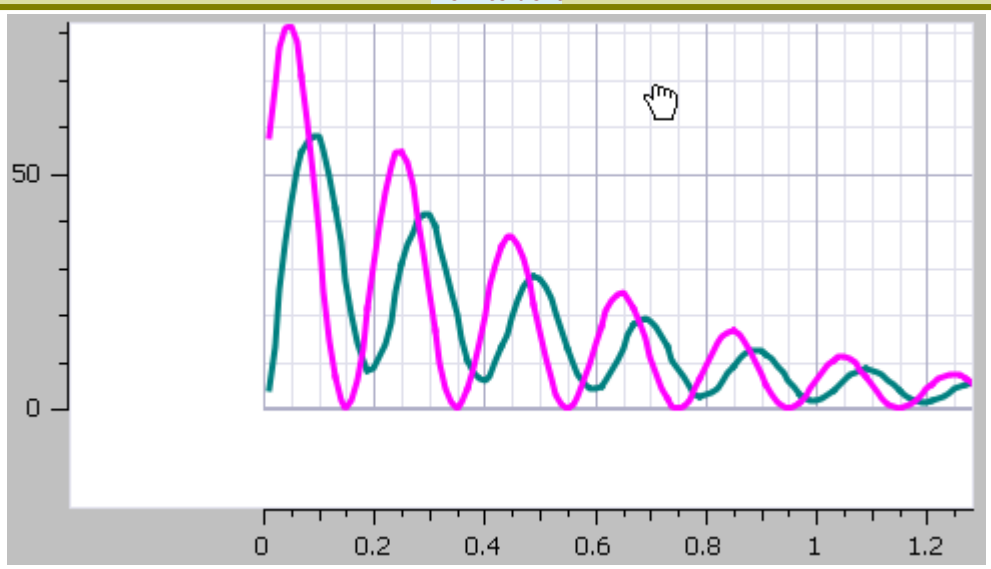
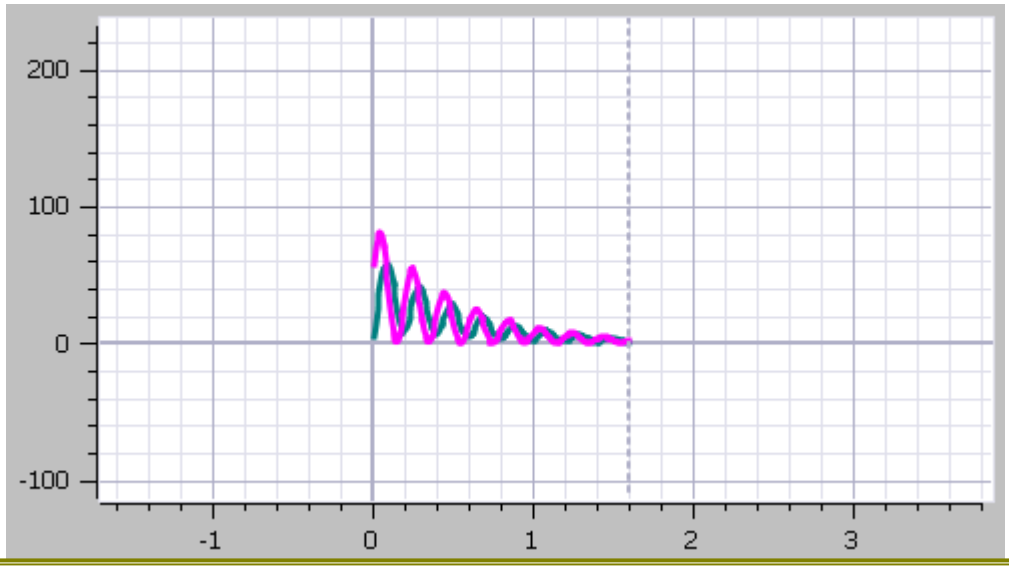
Saving the plot into a text file	
	You can save the plot's data in a text file where the data columns are separated by tabulations. After clicking this button, a standard operation system's window for access to files will open, where you select a file, into which the data are to be saved. Here is an example of the text file:

```
Step    Time    Pressure        Velocity
1       100     1
2       200     1       1
3       300     1       0.904514
4       400     0.594614        0.766632
5       500     0.0251451       1
6       600     0.132585        0.779342
7       700     0.239844        0.555458
8       800     0.351604        0.422173
```

Manual scaling and/or shifting the plot

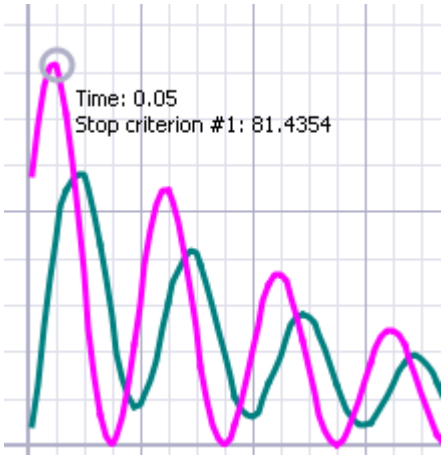
It is convenient to use the mouse for manual scaling and/or shifting the plot.

Desired action	How to do it
Expanding a rectangular fragment of the plot over the whole field	Click in the plot's field the left mouse button and then, keeping the button pressed, move the mouse pointer to another position and release the button there. The selected rectangle will expand over the whole field.  The mouse pointer is displayed as a cross, near which the values of the new plot's ranges are displayed.
Expanding a range along one axis	Click the left mouse button slightly below the axis of abscissas or to the left of the axis of ordinates. Then, holding the button, move the pointer along the axis to another position and release the button there. The selected interval will expand over the whole field.  Near the mouse pointer the new ranges of the plot along the selected axis are displayed.
Shifting the plot's field	Click the right mouse button within the plot's field and, keeping it pressed, shift the plot.

Desired action	How to do it
	 Until the shift is not finished, it can be canceled by pressing on the **Esc** key.
Scaling the whole field of the plot	Click a point within the plot's field and rotate the mouse wheel. The plot's display will zoom in or zoom out depending on direction, in which you rotate the wheel. 

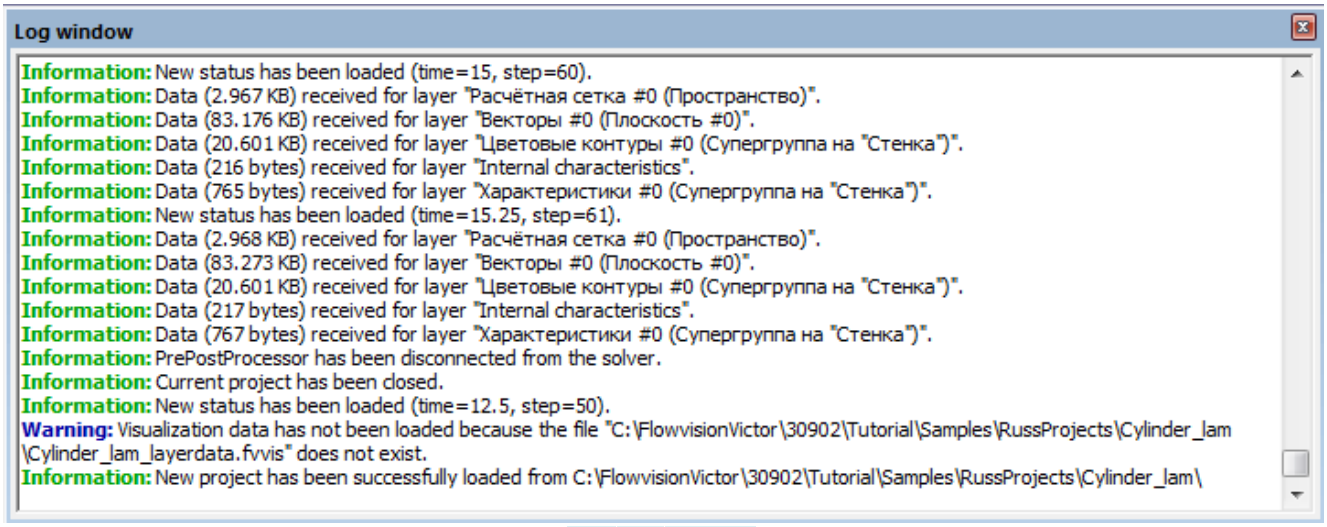
Viewing values displayed by the plot

When you move the mouse pointer close to a plot's line, the program automatically selects a point on the nearest plot's line (this point is marked by a small gray circle) and displays nearby its coordinates:



See also: an example of displaying a plot in the **Monitor** window in the section Viewing data in the Monitor window.

8.1.11 Window «Log»

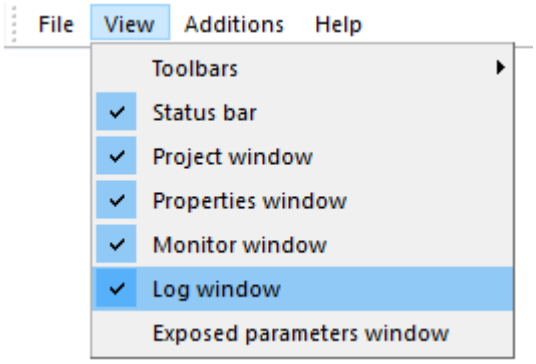


The **Log** window

The **Log** window is used to display the following types of messages:

- news reports on the implementation of the data exchange between the **Pre-Postprocessor** and **Solvers**
- warning and error messages in the **Pre-Postprocessor** when downloading files from the geometry model of the computational domain and imported objects or open the project files.

Displaying of the **Log** window is set by the **Main menu**'s command **View> Log** window:



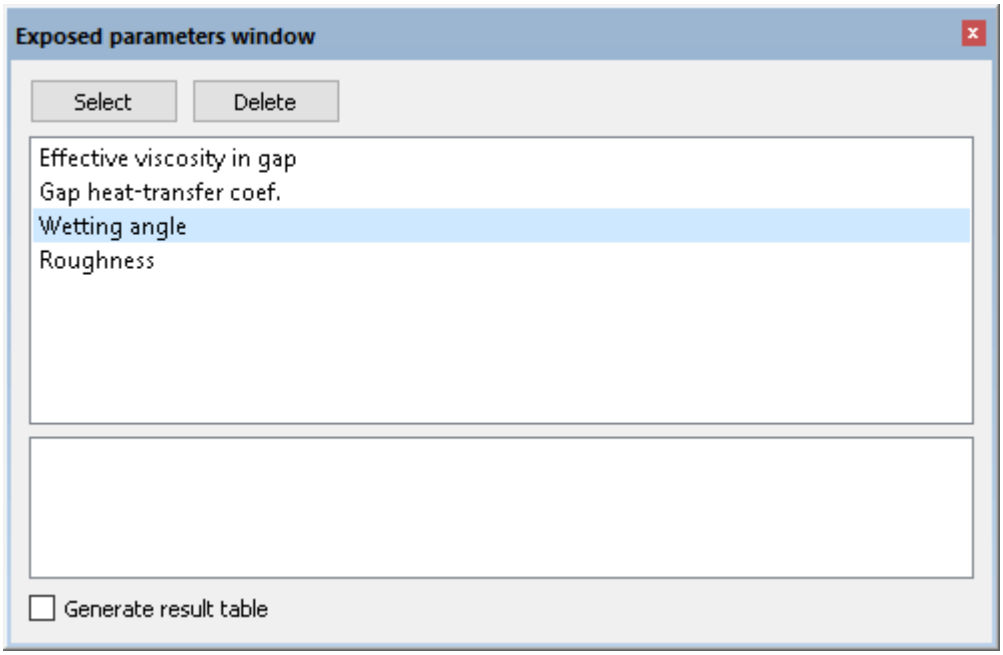
If an error occurs, try to fix it yourself or contact technical support FlowVision.

See also section Pre-Postprocessor's error messages and warnings.

8.1.12 Exposed parameters window

The **Exposed parameters window** contains a table of external parameters, the parameters that can be changed by a third-party program. Adding a parameter into this list is done from the **Properties** window, which includes the appropriate parameter. Additionally, in the **Exposed parameters window** you can generate output data, which will be read by a third-party program. Contents of the **Exposed parameters window** is determined by data in file of the external parameters (this file has name ***project name*.fvdtbl**).

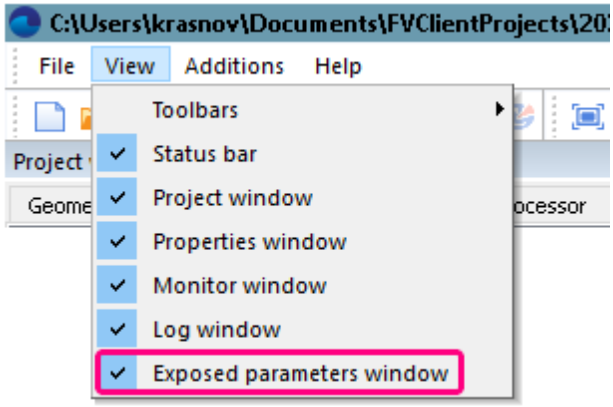
The bottom part of the **Exposed parameters window** contains the comment field where you can enter a brief description of the external parameter, which is selected in the list.



The **Exposed parameters window**

To display the **Exposed parameters window**, set the mark in the line **View > Exposed parameters window** of the Main Menu.

To hide the **Exposed parameters window**, unset the line **View > Exposed parameters window** of the Main Menu or click the **"x"** symbol in the upper right corner of the window.



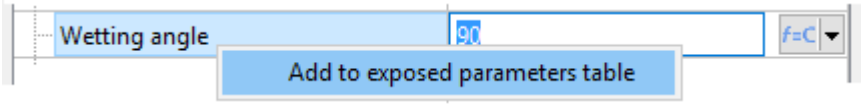
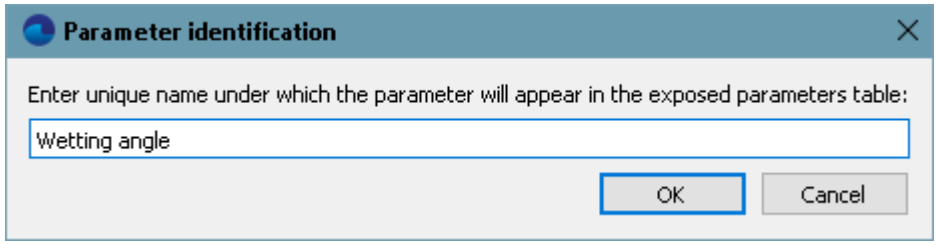
Command of the main menu for displaying/hiding the **Exposed parameters window**

The **Exposed parameters window** has the following elements:

Interface element	Description
Select (button)	Navigate to the selected parameter in the **Properties** window. Clicking the **Select** moves the focus in the Project window on the element in the project tree, which includes the parameter selected in the **Exposed parameters window**. Then you will be able to view and/or change the value of the selected external parameter in the **Properties** window.
Delete (button)	Delete the selected parameter from the table of the external parameters
List of the external parameters (this large field occupies the most area of the window)	Contains the list of external parameters

Interface element	Description
Comment to the selected external parameter (the text box at the bottom of the window)	Contains a commentary on the parameter, which is selected in the list of external parameters
Generate results table (checkbox)	Creating a file containing the latest values of residuals and user variables used in the stopping conditions. This file will be read by an external software. Checking **Generate results table** checkbox causes exporting into the files in client and server parts of the project the data presented in the following folders of the project tree: • Stopping conditions > **Residuals** • Stopping conditions > **User values**

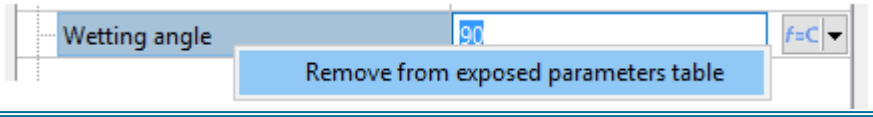
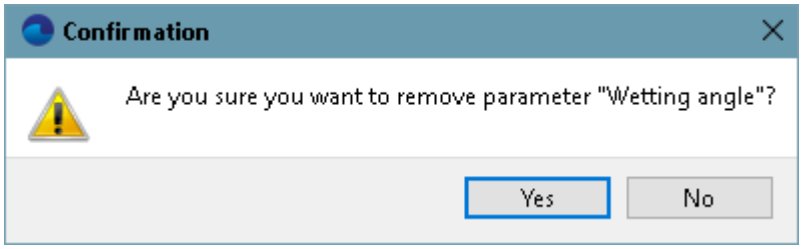
Adding a parameter to the list of external parameters

Step	Actions
1	Select the appropriate element in the project tree, then select the parameter in the Properties window (the parameter must be active, i.e. editable; active settings are displayed using a dark font, inactive ones with gray font).
2	Right-click the *name of* the parameter. A context menu will open with the **Add to exposed parameters table** command:  Select this command from the context menu.
3	The **Parameter identification** dialog box will open requesting you to specify a unique name, which will be displayed in the **Exposed parameters window**:  The program will prompt the parameter's name the same as is used in the **Properties** window (if this name is already included in the **Exposed parameters window**, then the program will prompt the same name with a digit appended). Enter a name for the new external parameter (or leave the default name), then click **OK**.
4	The parameter will be added into the **Exposed parameters window**.

Commands **Add to exposed parameters table** and **Remove from exposed parameters table** are not available for parameters, which cannot be edited at all or at the current selection of other parameters. If necessary, click the **Apply** button in the **Properties** window to make these commands available.

Removing a parameter from the list of external parameters

Step	Actions
1a	In the **Exposed parameters window** window select a parameter from the list and click the **Delete** button.
or 1b	*Or* in the **Properties** window of the corresponding element of the project tree right-click the parameter, which you wish to delete from the list of external parameters. Then select the **Remove from exposed parameters table** command:

Step	Actions
	
2	A dialog box will open requesting you to confirm your decision to remove the external parameter ("**Are you sure you want to remove parameter ... ?**"):  Click **Yes**. The parameter will be removed from the list of external parameters.

8.1.13 Window «Info»

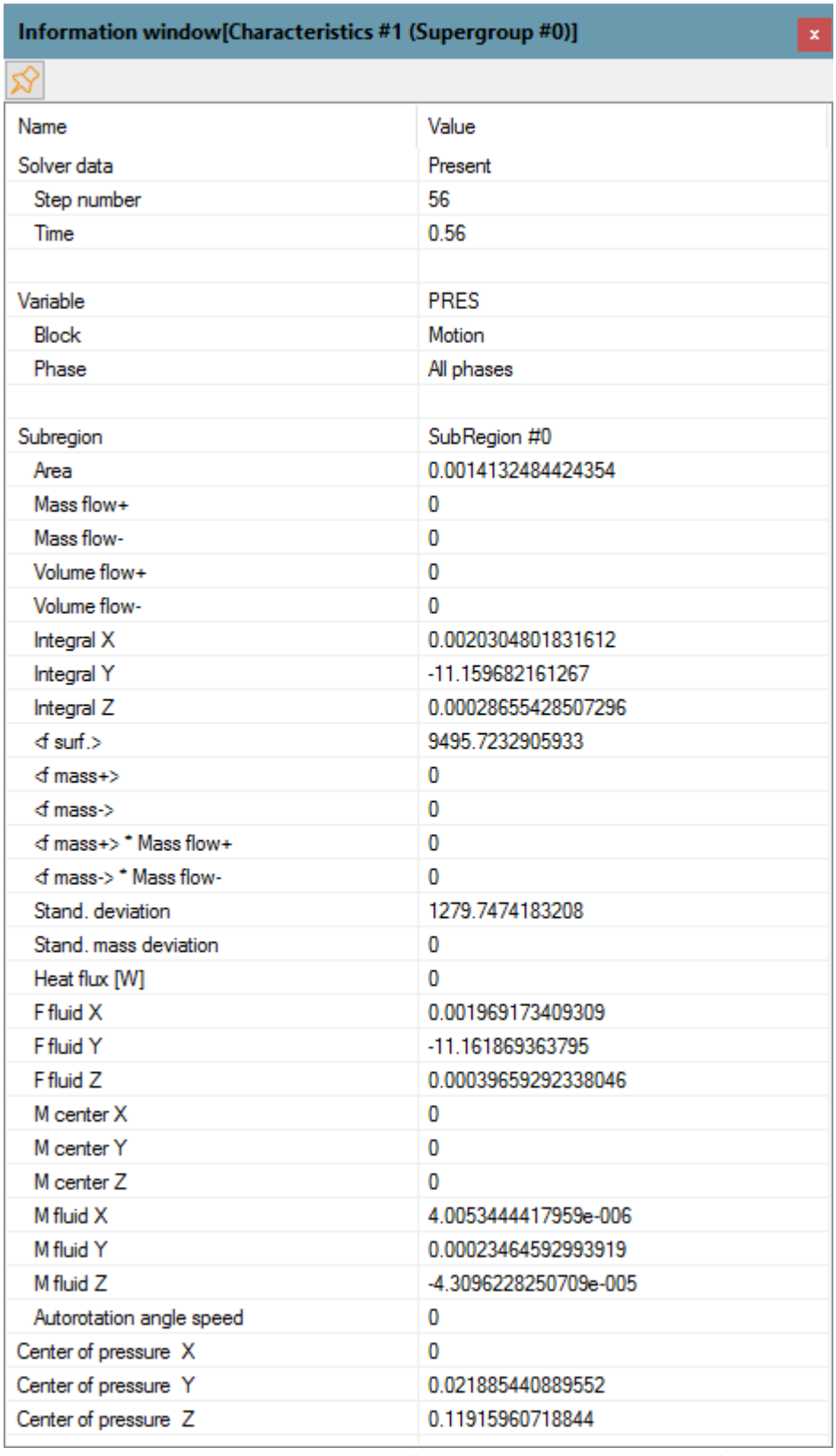
The **Info** window is used to display calculated at the current time step:

- components of the **Characteristics** element
- data that show distribution of a variable in a **Layer** element.

If you try to open the **Info** window for the other elements in the project tree, the window will display no data.

The **Info** window can be displayed in either on of the two modes:

- when the pushpin button  is released, the window displays the data of the currently selected project tree element; selection of another element of the project tree causes displaying other date that correspond to the selected element;
- when the pushpin button  pressed, the window displays parameters of only those element, which was selected in the project tree at the initial opening of this Info window.



Information window[Characteristics #1 (Supergroup #0)]

Name	Value
Solver data	Present
Step number	56
Time	0.56
Variable	PRES
Block	Motion
Phase	All phases
Subregion	SubRegion #0
Area	0.0014132484424354
Mass flow+	0
Mass flow-	0
Volume flow+	0
Volume flow-	0
Integral X	0.0020304801831612
Integral Y	-11.159682161267
Integral Z	0.00028655428507296
<f surf.>	9495.7232905933
<f mass+>	0
<f mass->	0
<f mass+> * Mass flow+	0
<f mass-> * Mass flow-	0
Stand. deviation	1279.7474183208
Stand. mass deviation	0
Heat flux [W]	0
F fluid X	0.001969173409309
F fluid Y	-11.161869363795
F fluid Z	0.00039659292338046
M center X	0
M center Y	0
M center Z	0
M fluid X	4.0053444417959e-006
M fluid Y	0.00023464592993919
M fluid Z	-4.3096228250709e-005
Autorotation angle speed	0
Center of pressure X	0
Center of pressure Y	0.021885440889552
Center of pressure Z	0.11915960718844

Example of the **Info** window (for a **Characteristics** element)

The **Info** window opens after selecting an element in the project tree and clicking in the toolbar **Work modes**.

The **Info** window closes by:

- clicking the icon "**X**" in the upper right corner,
- *or* clicking in the toolbar **Work modes** (in the latter case, *all the* **Info** windows, which are opened for different elements, close).

The data displayed in the **Info** window can be copied into the *Windows* clipboard by the **Ctrl+C** hot keys.

8.1.14 Formula editor

Formula editor is a component of **Pro-Postprocessor**, designed to set a variable in a formula.

Formula editor dialog box is intended to specify a set of expressions (formulas), determine the value of a parameter.

Defined set of expressions (formulas) has the form:
`<basic expression>; <auxiliary formula 1>; <auxiliary formula 2>; ... <auxiliary formula N>;`

where

`<basic expression>` is an expression that defines value of the parameter. This expression contains no "=" symbol.

`<auxiliary formula>` is an assignment (it contains the "=" symbol) that defines a variable, which is used in the `<basic expression>` or in another `<auxiliary formula>`.

These expressions (formulae) can be presented in the set in random order.

Digital separator in numerical values is dot (".").

All the variables and constants in the set of expressions (formulas), the user is assigned to a specific *local name (link)*, which operates within a given set of expressions. The name referred to a local to emphasize its uniqueness only within the edited set of expressions. In another set of expressions that name can be assigned to the same or another variable or constant.

The *local name (link)* is set when you first turn on a variable or constant in the set of expressions (formulas). It consists of numbers, letters and an underscore, and the first character must be a letter. Register dialed is not case sensitive.

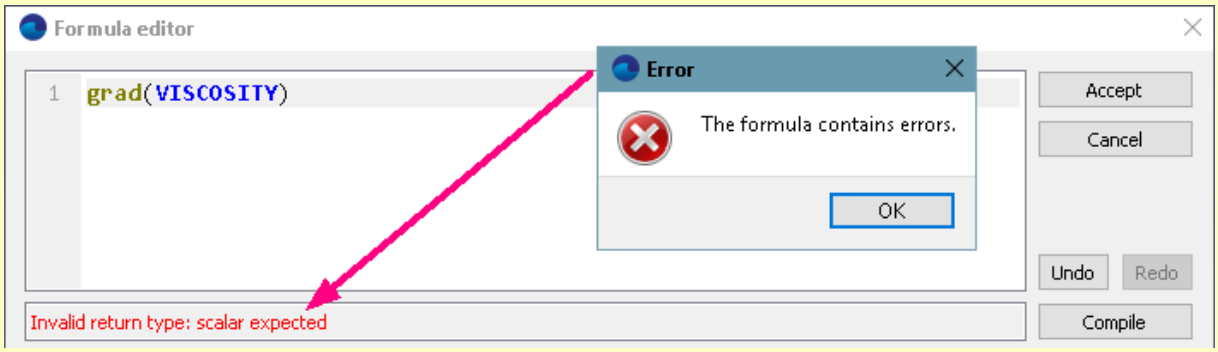
The local name cannot coincide with the names of functions (operations) and constants.

When **Formula editor** is used to specify properties of **Substances**, *absolute* **Temperature** and **Pressure** are taken (see section Reference parameters, absolute and relative variables for details).

In other cases of use of formulae or tables, program takes the *relative* **Temperature** and **Pressure**.

Type of the value (scalar or vector) specified in the **Formula editor** has to correspond to aim of the formula.

So, at an attempt to determine a scalar user variable by a formula, which returns a vector, an error message "**The formula contains errors**" will be displayed and the "**Invalid return type: scalar expected**" message will appear in the informational line:



Similarly, if a formula is intended to determine a vector but returns a scalar, an error message will be displayed with the "**Invalid return type: vector expected**" message in the informational line.

Multiple branching

The structure of formulas and expressions can include multiple branching structure:

Syntax	Description

{
(<COND1>): <S1>;
(<COND2>): <S2>;
...
<DEFAULT_COND>
}
	If condition <COND1> is fulfilled, the value of the expression will be <S1>. If not, condition <COND2> will be tested.
	If all conditions are not fulfilled, then the default value, <DEFAULT_COND>, will be assigned to the expression.

Syntax	Description
{(<COND1>): <S1>; (<COND2>): <S2>; ...<DEFAULT_COND>}	

When building a structure, the following symbols:

- braces ("{" and "}") are used to indicate the beginning and end of the construction of multiple branching
- colon (":") is used as a separator *conditions* and *values* in the design of multiple branching
- semicolon (";") is used as a separator conditions in the construction of multiple branching

Syntax checking formulas

When you set the formula system performs partial analysis of the syntax of typed text for errors. Those elements of the text, which are treated by the system as erroneous, are displayed in red (false color and elements can be changed, it is configured in the menu **File > Preferences** in group settings **Formula editor**).

Full scan syntactically correct set of expressions and formulas are done by pressing **Compile**.

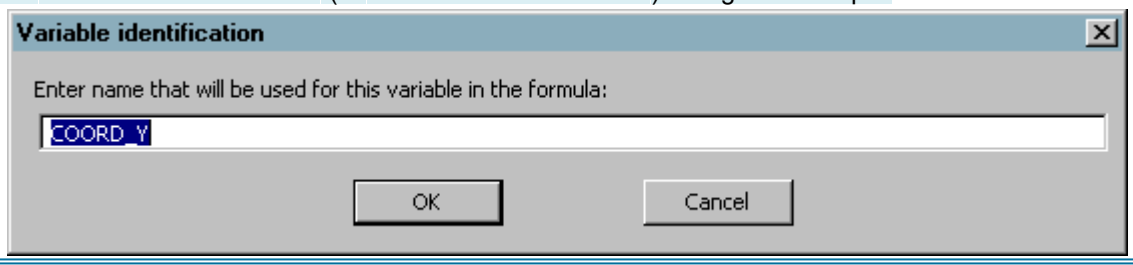
Without syntax errors can not exit the dialog **Formula editor** with saving changes.

8.1.14.1 Editing a set of expressions (formulae)

Editing a set of expressions (formulas) consists of the following operations are described below:

- insert a variable (constant) in the formula (performing the first and subsequent inserts differ)
- removing the reference (local variable names (constants))
- insertion operation

First insert a variable or a constant

Step	Actions
1	Open the tab, which displays the variable or constant (the **Constants** tab for a constant) and then: • *either* double-click the row variable (constant) • *or* right-click the row variable (constant) and select **Add reference & insert into formula**. The **Variable identification** (or **Constant identification**) dialog box will open: 
2	Enter the local name of the variable (constant) and then click **OK**. If you added a variable, then tab to the name of the variable is added to the name of a local variable, enclosed in square brackets. If you added a constant, the **Constant** tab in the **Name** column in the formula is displayed local name constants. In the list on the **References** tab added to the string with the added reference.

Inserting a variable (constant) for which there is a link

Do one of the following:

	Actions
•	Click the tab, which displays the variable or constant, and do either of the following: • double-click the row variable (constant) • *or* right-click the row variable (constant) and select the **Insert into formula** command
•	*or* go *to* the **References** tab and do one of the following: • Double-click the reference string • *or* right-click the link line and select the **Insert into formula** command

Removing links (a local name of a variable/constant)

Do one of the following:

Actions	
•	On the tab, which displays the variable or constant (the **Constants** tab for constants) to open the context menu on the row of the variable (constant) and select **Remove reference**.
•	On the **References** tab right-click the link line and select **Remove reference**.

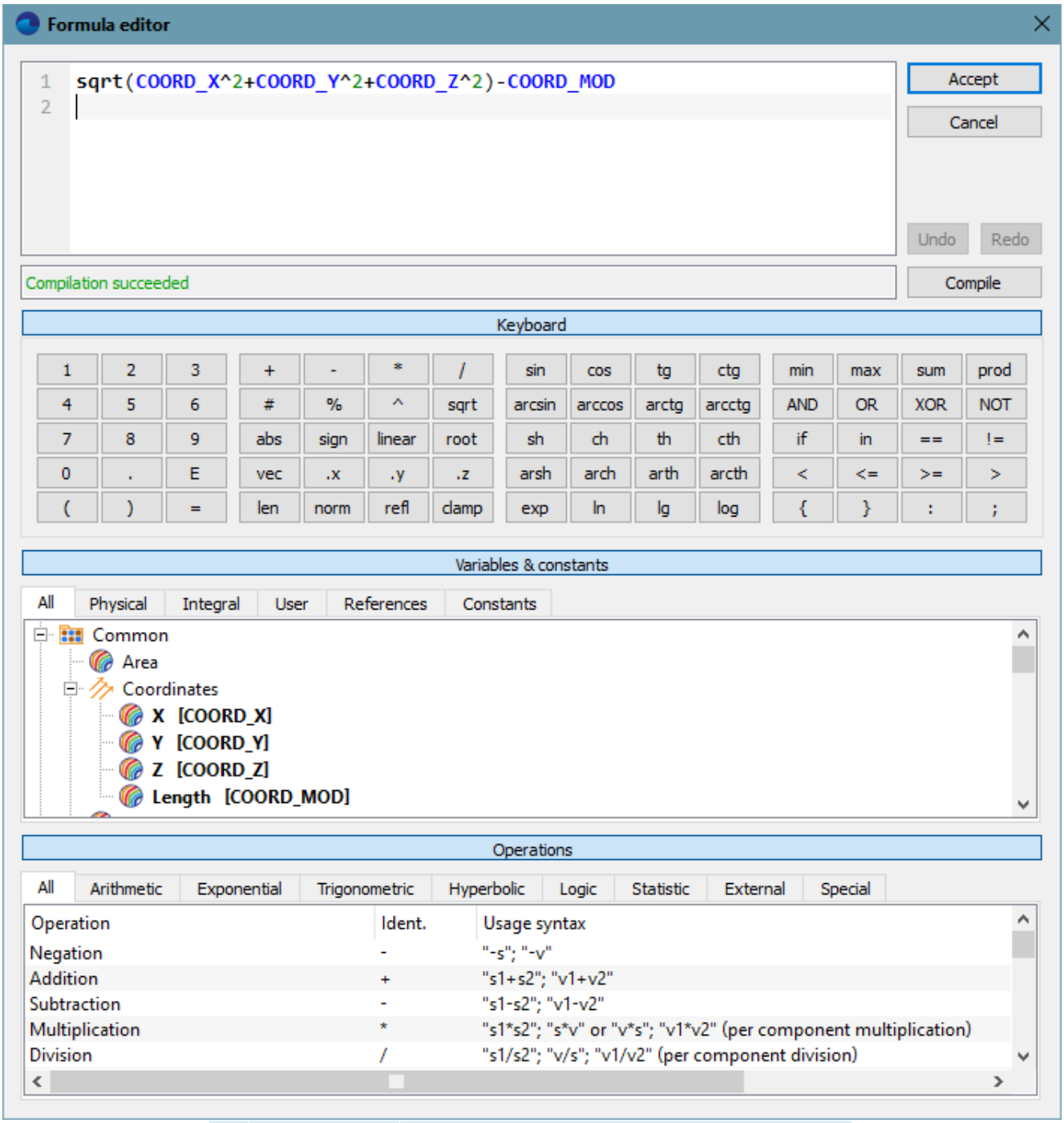
Insert operations

On the **Operations** pane, which displays the operation, and then do one of the following:

Actions	
•	Double-click the row of the operation.
•	Right-click the line operation and select **Insert into formula**.

8.1.14.2 User interface of Formula editor

The **Formula editor** dialog box is opened by clicking the f_x or f_x icon in the expanded input field in the **Properties** window.



The **Formula editor** dialog box (an example of setting a scalar value)

Elements of Formula editor

Formula editor contains the following interface elements:

- **Formula pane**, which displays the formula
- **Informational line**, where messages about successful compilations or error messages.
- **Keyboard**, which is a set of buttons that are used to enter a formula. The keyboard includes digits and the most used operations and functions. See details in the section"Keyboard of Formula editor".
- the **Variables & constants** panel, which contains variables and constants that can be used for specifying formulae

- the **Operations** panel, which contains operators and functions available to specify formulae.
- buttons:

Button	Description
Accept	Compile the formula and close the **Formula editor**. The same effect is caused also by pressing either **Ctrl+Enter** or **Alt+Enter** or **Shift+Enter** when the cursor locates in the **Formula pane**.
Cancel	Close the editor without saving the changes
Undo	Undo the last change in the formula
Redo	Redo the last undone change
Compile	Compile the formula. To the left of this button, there is a line informing you about results of the compilation.

Rules for defining a formula

1. Formula is displayed in the **Formula pane**. The main formula, which is used to calculate the target variable, is defined first in the **Formula pane** and does not contain the '=' symbol.
2. To define a formula, you can use numbers, variables, constants, mathematical operators and functions. Each class of these elements is displayed using its specified color. The colors are configured in the **File > Preferences** menu in the **Formula editor** group of settings (see Basic settings of Pre-Postprocessor).
3. Variables and constants are used in a formula under only specially created name (link). Links to existing variables are created when you first insert the variable or constant in the formula. To insert a variable or constant in the formula, you need to double-click and, in the case of the first run, in the window that appears, set the variable name (which is the link).
4. Constants and variables can also be set directly in the **Formula pane**. Syntax of variable assignment is **x=f(a,b,c)**, where **x** is the name of the variable or constant, and **f(a,b,c)** is a formula. Variables and constants are given after the main expression. The **";"** symbol is used as a separator.
5. Syntax of **Formula editor** only is only used to set the operators or functions.
6. After setting the formula is recommended to compile it in order to prevent possible errors.

You can not use same names for different variables or constants.

Pane "Keyboard"

The composition of the panel included the number buttons, and arithmetic operations listed in the **Main** toolbar,as well as signs of multiple branching structures.

Action to be taken is indicated in the tooltip that appears when the mouse pointer to the presentation to the button panel **Keyboard**.

Pane "Variables & constants"

The **Variables & constants** panel is designed to provide the insert variables and constants under the local names in a set of expressions (formulas).

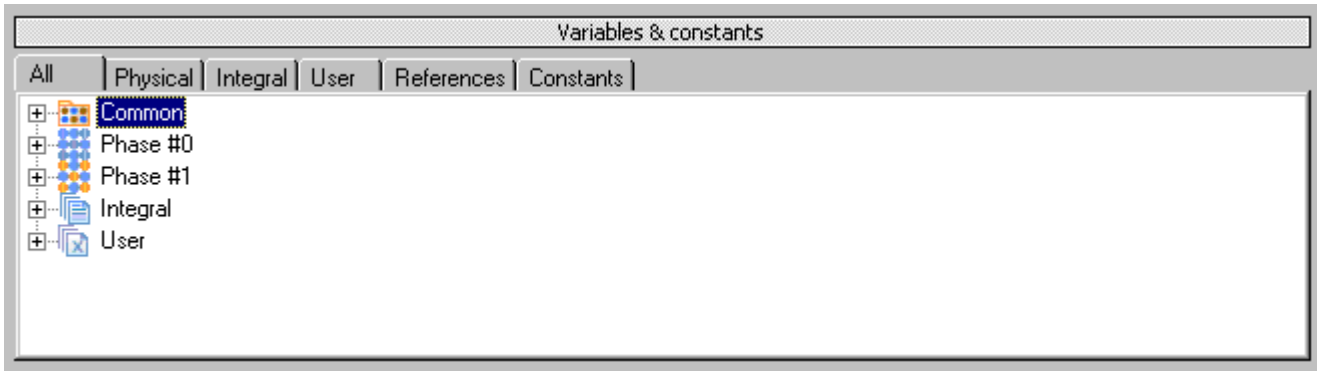
The **Variables & constants** panel includes several tabs (set of tabs depends on the parameter for which is given by the formula). Variables tabs organized in a tree structure. Variable name consists of two parts

- global name under which the variable appears in the system formed by the position of the variable in the tree (**Group > Variable > Component**)
- in square brackets - the local name of a user-defined

On the **Variables & constants** tabs are listed below.

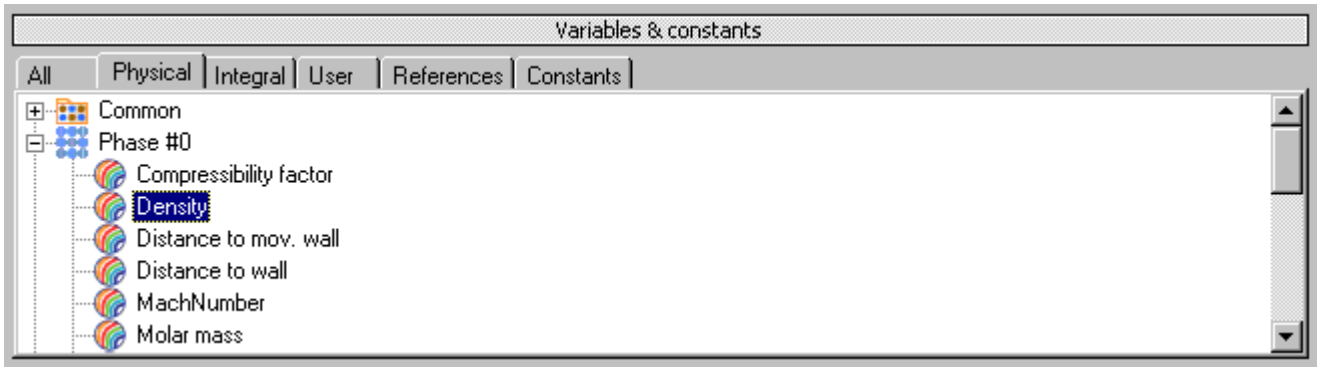
Tab "All"

The **All** tab contains all variables.

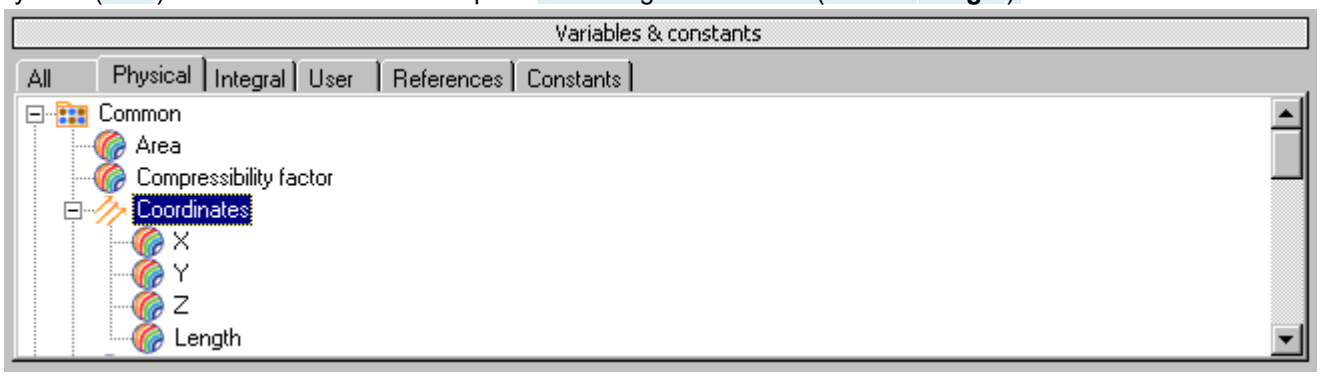


Tab "Physical"

Local variables are generated by physical processes (available only if you set the **local variables**).

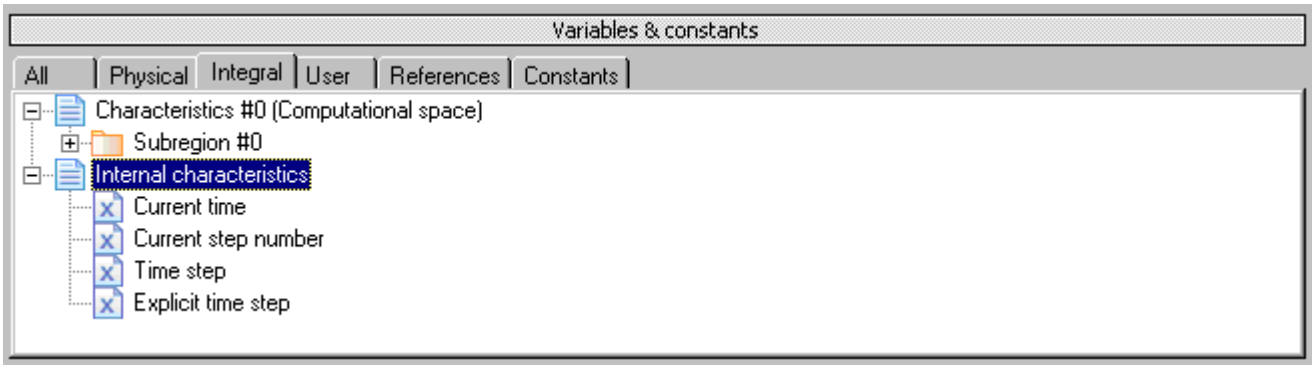


In the folder **Common > Coordinates** can indicate the coordinates **X**, **Y**, **Z** point in the absolute coordinate system (ACS) and the distance from the point to the origin of the ACS (variable **Length**).



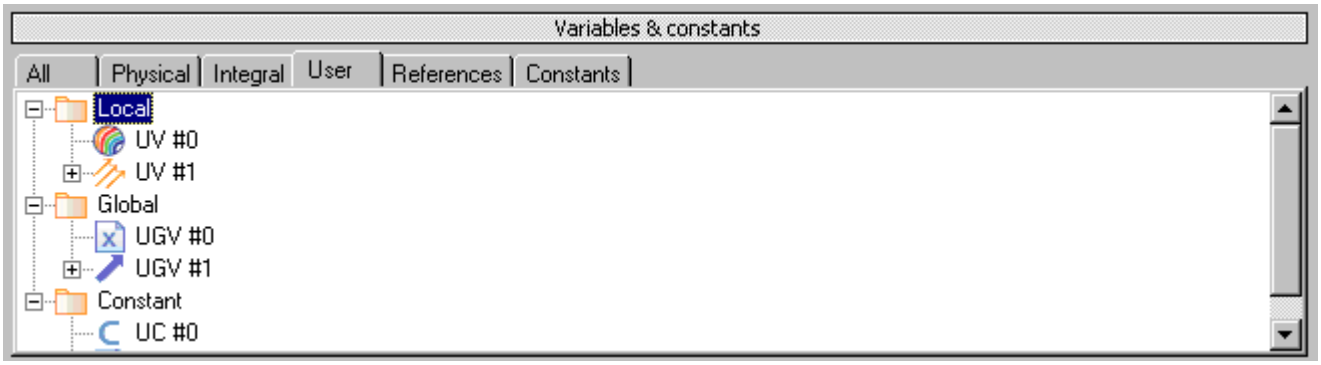
Tab "Integral"

The **Integral** tab contains integral variables.



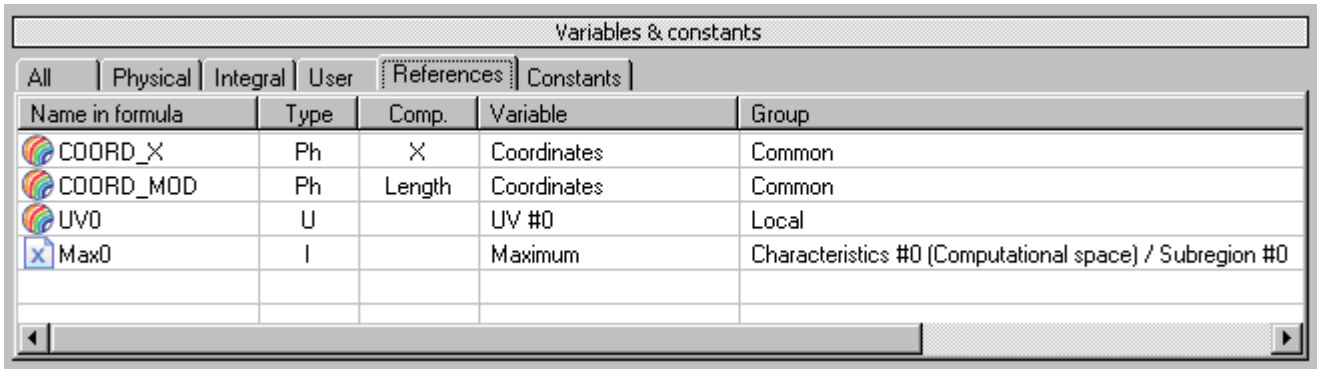
Tab "User"

The **User** tab contains user-defined variables.



Tab "References"

The **References** tab contains a list of all the local names of the variables used in the set of expressions (formulas) i.e. names of **Physical**, **Integral**, and **User** variables, used in formulae. Reference to the variable is created through the context menu of the variable, as well as automatically when you add a variable in the formula.

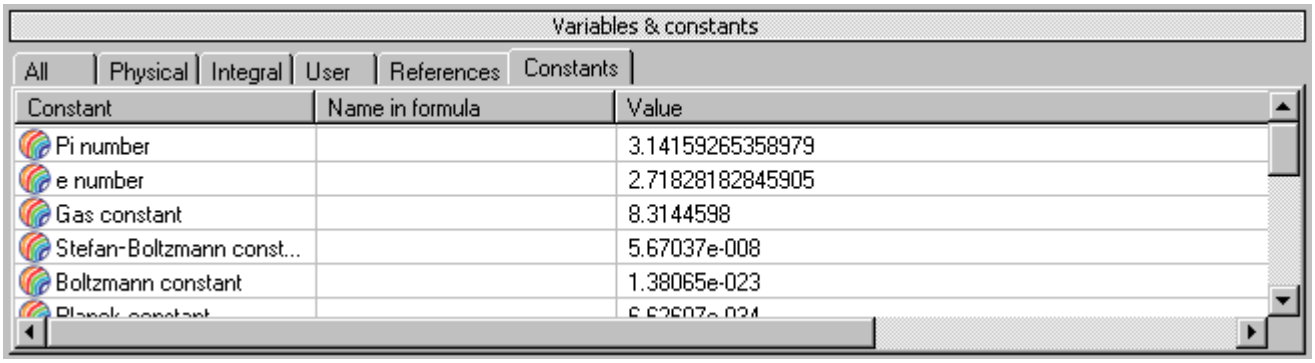


Columns in the **References** tab:

Column's name	Description
Name in formula	The local name of the variable
Type	The variable type (tab, where it is displayed): • **Ph** means **Physical** • **I** means **Integral** • **U** means **User**
Comp.	Component of the vector variable
Variable	Name of variable
Group	The group to which the variable

Tab "Constants"

The **Constants** tab contains all constants:



Columns in the **Constants** tab:

Column's name	Description
Constant	The global name of the constant

Column's name	Description
Name in formula	The local name of the constant
Value	The numerical value of the constant

The following constants are provided:

Constant	The default name to be inserted into the formula	Value
Pi number	`PI`	**3.14159265358979**
e number	`E`	**2.71828182845905**
Gas constant	`R`	**8.3144598**
Stefan–Boltzmann constant	`SIGMA`	**5.67037e-008**
Boltzmann constant	`K`	**1.38065e-023**
Planck constant	`H`	**6.62607004e-034**
Vacuum permittivity	`EPS_ZERO`	**8.85419E-012**
Vacuum permeability	`MU_ZERO`	**1.25664E-006**
Gravitational constant	`G`	**6.67408E-011**
Speed of light in vacuum	`C`	**299792458**
Ort X	`ORT_X`	**1; 0; 0**
Ort Y	`ORT_Y`	**0; 1; 0**
Ort Z	`ORT_Z`	**0; 0; 1**
Invalid value	`NONE`	**(none)**
Invalid vector	`NO_VECTOR`	**(none); (none); (none)**

Context menu for a variable

Menu item	Description
Add a link to insert into the formula	Add a link to the selected variable and insert the variable in the formula
Add a link	Add a link to the selected variable

Pane "Operations"

Pane **Operations** includes 9 tabs that display available in the **Formula Editor** operations. Part of the process corresponds to the buttons of the **Keyboard** pane.

Each tab displays a list of transactions in a table with columns:

- **Operation** – name of the operation
- **Ident.** – symbol or code of the operation. This symbol or code is displayed in the formula.
- **Usage syntax** – examples of use of the operation in the formula, as well as comments

In examples of the syntax the following notation is used:

- `s`, `s1`, `s2`, `s3` – scalar values
- `v`, `v1`, `v2`, `v3` – vector values
- `sk` – the coefficient of linear interpolation (this is a scalar value)
- `smin`, `smax` – scalar limiters for scalar and vector values

- **vmin**, **vmax** – vector limiters for vector values
- **spower** – index of power or index of radical
- **sbase** – logarithm base
- **scond**, **strue**, **sfalse**, **vcond**, **vtrue**, **vfalse** – conditions and results of a conditional operator
- **a**, **b** – limits of the range for the range check operation
- **sVar** – an external (relating to the formula) scalar local user variable or physical variable*)
- **sPhysVar** – an external (relating to the formula) scalar physical variable*)
- **vVar** – an external (relating to the formula) vector local user variable or physical variable*)
- **vPhysVar** – an external (relating to the formula) vector physical variable*)
- **vVarComp** – component (x, y, z) of an external (relating to the formula) vector local user variable or physical variable*)

*) See details in the subsection "*Tab «External»*" below.

Local user variables generally depend on spatial coordinates.

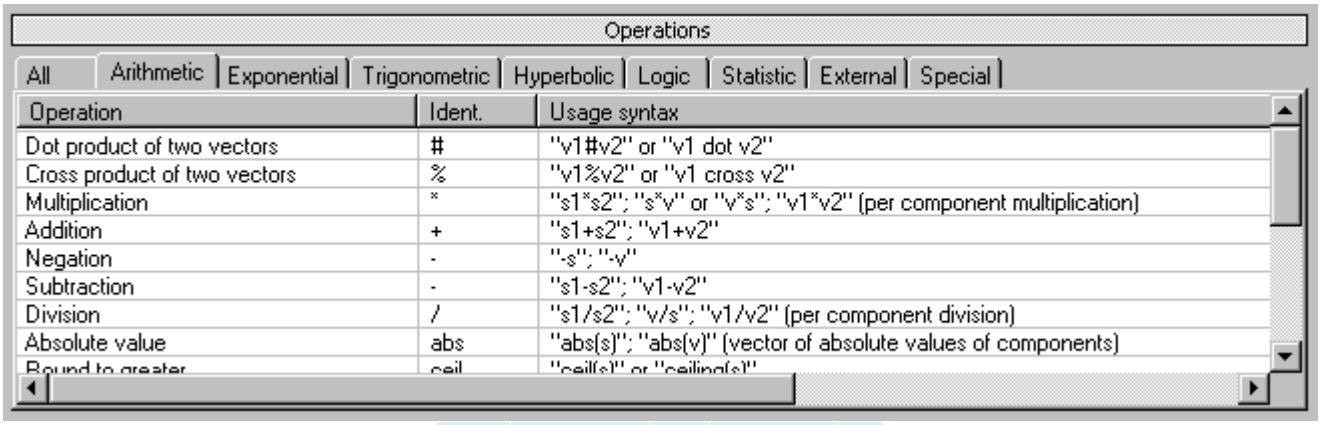
Operations are grouped in tabs by type in the title of the tab. On the **All** tab displays all the operations included in the lists of "sectoral" tabs.

To insert an operation into the formula, do one of the following:

- *either* double-click the line of the operation
- *or* open the context menu on the line and select **Insert into formula**

Tab «Arithmetic»

The **Arithmetic** tab contains arithmetic operations.



Panel **Operations**, the **Arithmetic** tab

Operation	Identifier	Syntax	Description
Negation	-	-s -v	Change the sign of the scalar or vector components of the opposite
Addition	+	s1+s2 v1+v2	Addition of scalars and vectors
Subtraction	-	s1-s2 v1-v2	Subtraction of scalars and vectors
Multiplication	*	s1*s2 s*v v1*v2	Multiplication (for vectors - component wise multiplication)
Division	/	s1/s2	Division (for vectors - exploded division)
Dot product of two vectors	#	v1#v2 v1 dot v2	Scalar product of vectors
Cross product of two vectors	%	v1% v2 v1 cross v2	Vector multiplication of vectors **v1** and **v2**

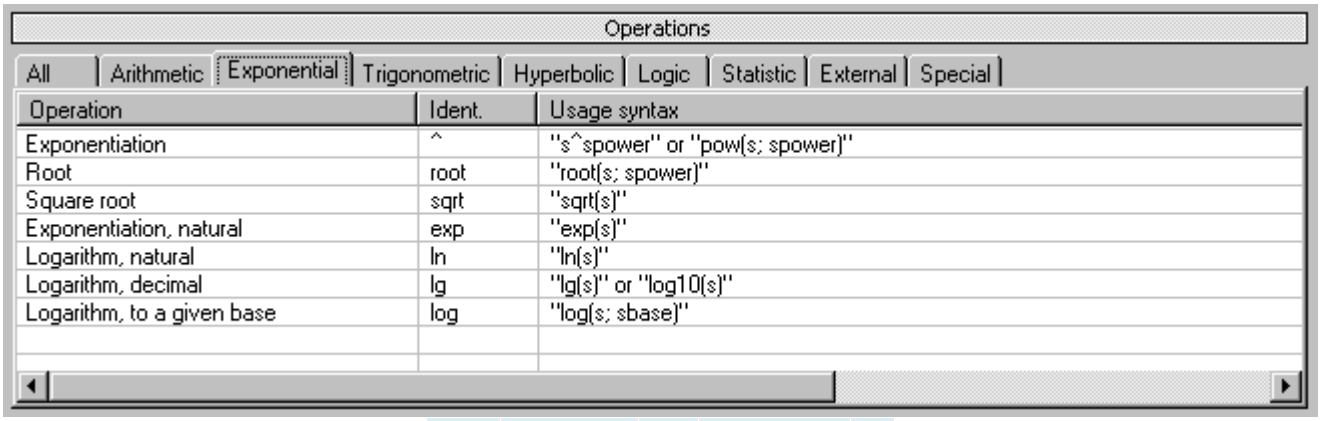
Operation	Identifier	Syntax	Description
Absolute value	`abs`	`abs(s)` `abs(v)`	Module s Vector component modules `v`
Sign	`sign`	`sign(s)` `sign(v)`	Sign `s(-1;0;1)` Vector signs of the components of the vector `v`
Floating-point division remainder	`mod`	`s1 mod s2`	Fractional remainder of the division of `s1` to `s2`
Integer part	`trunc`	`trunc(s)` `truncate (s)`	The integer part of `s`
Fractional part	`frac`	`frac(s)` `fraction (s)`	The fractional part of `s`
Rounding to smaller	`floor`	`floor(s)`	`s` is rounded to the less
Round to greater	`ceil`	`ceil(s)` `ceiling (s)`	`s` is rounded to the greater
Round to closest	`round`	`round(s)`	`s` is rounded to the nearest
Linear interpolation	`linear`	`linear(s1; s2; sk)` `linear(v1; v2; sk)`	Linear interpolation $s_1+k(s_2-s_1)$ $v_1+k(v_2-v_1)$ The result of the linear interpolation `linear(s1;s2;sk)` is the expression `s1+sk*(s2-s1)` where `sk` is the coefficient (for vector quantities it is similarly)

Clamping	`clamp`	`clamp(s; smin; smax)`
		`clamp(v; smin; smax)`
		`clamp(v; vmin; vmax)`
			Restriction of x within the ranges from xmin to xmax:
			• values x<xmin are replaced by xmin
			• values x>xmax are replaced by xmax
			The result of the operation of clamping `clamp(s;smin;smax)` is the following expression:
			• `s`, if `smin<s<smax`
			• *or* `smin`, if `s<smin`
			• *or* `smax`, if `s>smax`
			Components of vector quantities are limited either scalar quantities (restrictions are the same for all components) or vector (in this case, set the borders for each component of the vector).

Random value in range [0;s]	`random`	`random(s)`	A random value in the range [0;`s`]

Tab «Exponential»

The **Exponential** tab contains exponential functions. In the description of the syntax `spower` denotes the degree, `sbase`- log base.

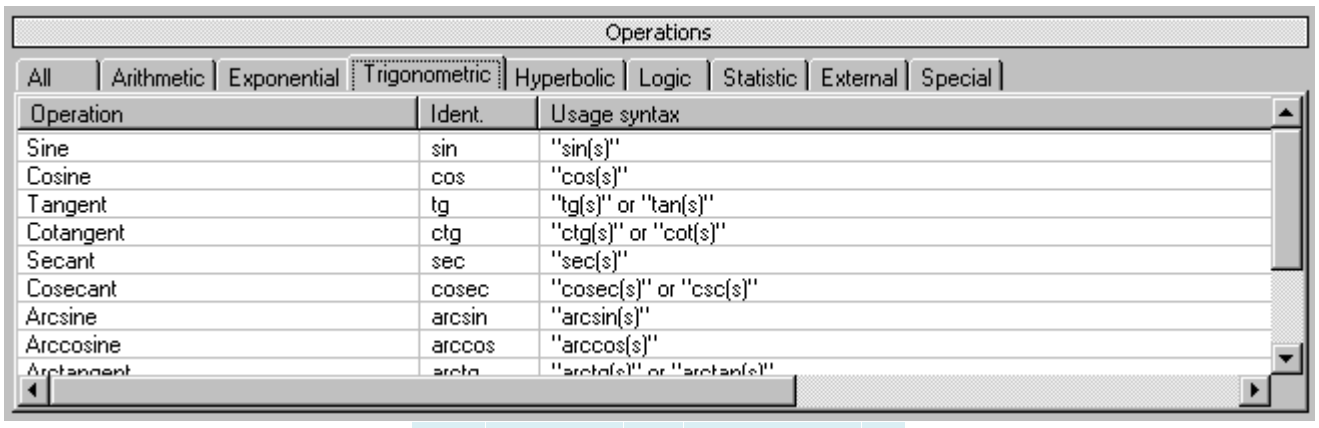


Panel **Operations**, the **Exponential** tab

Operation	Identifier	Syntax	Description
Exponentiation	`^`	`s^spower`	Raising `s` to `spower`th power
Root	`root`	`root(s; spower)`	The `spower`th root of `s`
Square root	`sqrt`	`sqrt(s)`	Square root of `s`
Exponentiation, natural	`exp`	`exp(s)`	Exhibitor e^s
Logarithm, natural	`ln`	`ln(s)`	The natural logarithm of `s`
Logarithm, decimal	`lg`	`lg(s)` *or* `log10(s)`	Decimal logarithm of `s`
Logarithm, to a given base	`log`	`log(s;sbase)`	Logarithm of `s` to the base `sbase`

Tab «Trigonometric»

The **Trigonometric** tab contains trigonometric functions.



Panel **Operations**, the **Trigonometric** tab

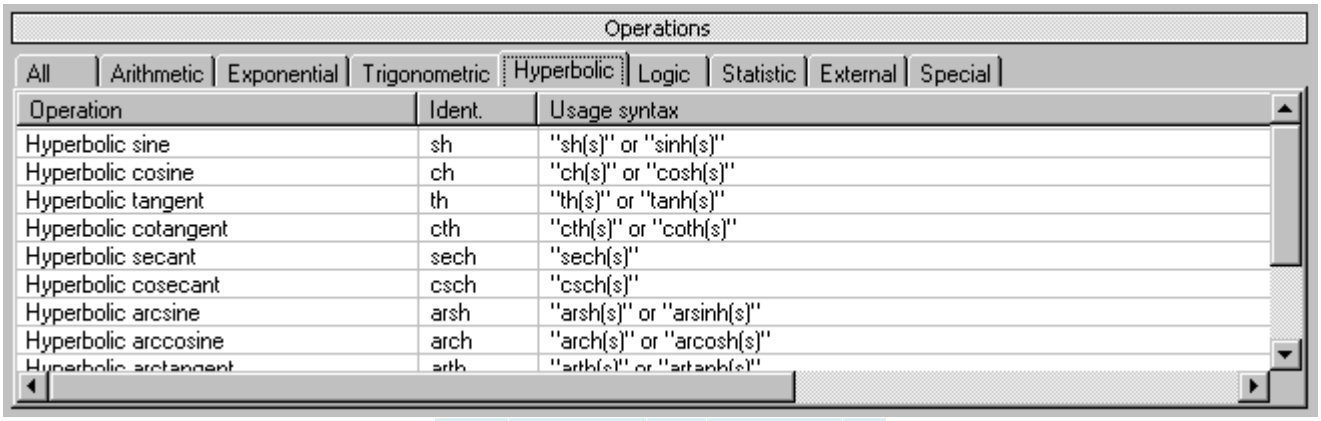
Operation	Identifier	Syntax	Description
Sine[*)]	`sin`	`sin(s)`	Sine s
Cosine[*)]	`cos`	`cos(s)`	Cosine s
Tangent[*)]	`tg`	`tg(s)`	Tangent s
Cotangent[*)]	`ctg`	`ctg(s)`	Cotangent s

Operation	Identifier	Syntax	Description
Secant[*)]	sec	sec(s)	Secant s
Cosecant[*)]	cosec	cosec(s)	Cosecant s
Arcsine	arcsin	arcsin(s)	Arcsine s
Arccosine	arccos	arccos(s)	Arccosine s
Arctangent	arctg	arctg(s)	Arctangent s
Arccotangent	arcctg	arcctg(s)	Arc cotangent s
Arcsecant	arcsec	arcsec(s)	Arc secant s
Arccosecant	arccsc	arccosec(s) arccsc(s)	Arc cosecant s

[*)] Arguments of these functions are set in radians.

Tab «Hyperbolic»

The **Hyperbolic** tab contains hyperbolic functions.



Panel **Operations**, the **Hyperbolic** tab

Operation	Identifier	Syntax	Description
Hyperbolic sine	sh	sh(s)	Hyperbolic sine s
Hyperbolic cosine	ch	ch(s)	Hyperbolic cosine s
Hyperbolic tangent	th	th(s)	Hyperbolic tangent s
Hyperbolic cotangent	cth	cth(s)	Hyperbolic cotangent s
Hyperbolic secant	sech	sech(s)	Hyperbolic secant s
Hyperbolic cosecant	csch	csch(s)	Hyperbolic cosecant s
Hyperbolic arcsine	arsh	arsh(s)	Hyperbolic sine s
Hyperbolic arccosine	arch	arch(s)	Hyperbolic cosine s
Hyperbolic arctangent	arth	arth(s)	Hyperbolic tangent s
Hyperbolic arccotangent	arcth	arcth(s)	The inverse hyperbolic cotangent s
Hyperbolic arcsecant	arsech	arsech(s)	Hyperbolic areasekans s
Hyperbolic arccosecant	arcsch	arcsch(s)	Hyperbolic areakosekans s

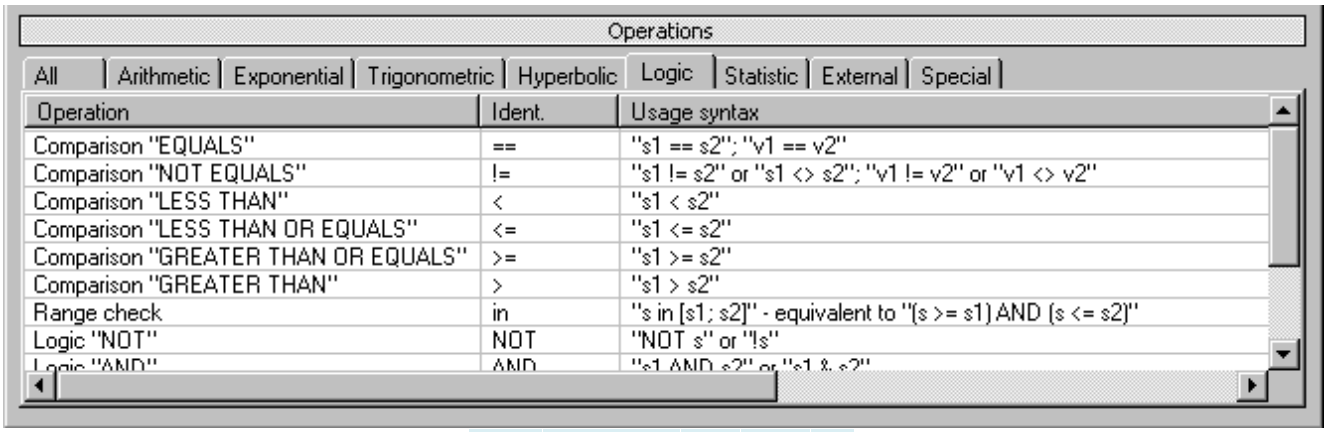
Tab «Logic»

The **Logic** tab contains logical operations.

The result of the logic operation is a scalar quantity equal to `1` if the result is true, and `0` if the result is false. For example, the result of the check of being be equal operation (`s1==s2`) is `1` when `s1=s2`, and `0` when `s1≠s2`.

It is recommended to enclosed the logical operations in parentheses, for example, `((s1<s2)AND(s1>s3))`.

In several operations - **`NOT`**, **`AND`**, **`OR`**, **`XOR`**, **`if`** - an implicit conversion of real variables in logic: `s = 0` if `s = 0.0`, and `s=1` if `s≠0.0`.

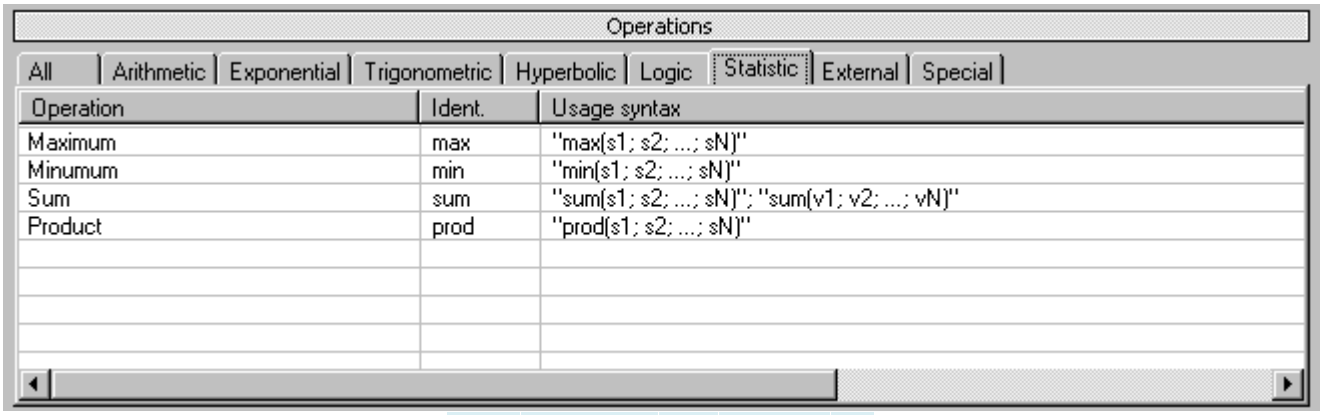


Panel **Operations**, the **Logic** tab

Operation	Identifier	Syntax	Description
Comparison "EQUALS"	`==`	`s1==s2` `v1=v2`	Equality
Comparison "NOT EQUALS"	`!=`	`s1!=s2; s1<>s2` `v1!=v2; v1<>v2`	Inequality
Comparison "LESS THEN"	`<`	`s1<s2`	Less
Comparison "LESS THEN OR EQUALS"	`<=`	`s1<=s2`	Less than or equal
Comparison "GREATER THEN OR EQUALS"	`>=`	`s1>=s2`	Greater than or equal
Comparison "GREATER THEN"	`>`	`s1>s2`	Greater than
Logic "NOT"	`NOT`	`NOT s` `!s`	Not `s`
Logic "AND"	`AND`	`s1 AND s2`	Operator AND (conjunction)
Logic "OR"	`OR`	`s1 OR s2`	Operator OR (disjunction)
Logical "EXCLUSIVE OR"	`XOR`	`s1 XOR s2`	Operator XOR
Conditional branching	`if`	`if (scond; strue; sfalse)` `if (vcond; vtrue; vfalse)`	The "if" operator. If the condition `scond` (`vcond`) is fulfilled, then the value of the expression will be `strue` (`vtrue`). Else the value will be `sfalse` (`vfalse`).
Range check	`in`	`s in [a;b]`	If `a≤s≤b` then **1**. If `s<a` or `s>b`, then **0**.

Tab «Statistic»

The **Statistic** tab contains statistical operations.

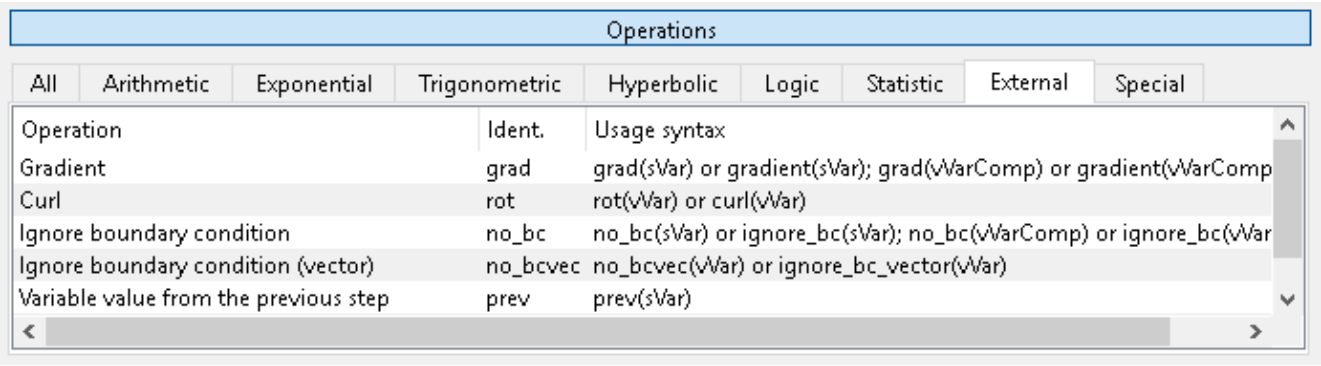


Panel **Operations**, the **Statistics** tab

Operation	Identifier	Syntax	Description
Minimum	`min`	`min(s1;s2;...;sN)`	Minimum of `s1`, `s2`, ..., `sN`*)
Maximum	`max`	`max(s1;s2;...;sN)`	Maximum of `s1`, `s2`, ..., `sN`*)
Sum	`sum`	`sum(s1;s2;...;sN)`	Sum `s1+ s2+...+sN`
Product	`prod`	`prod(s1;s2;...;sN)`	Product `s1*s2*...*sN`

*) These operations must have 3 or more arguments.

Tab «External»



Panel **Operations**, the **External** tab

The **External** tab contains external functions, which are calculated by **Solver** out of the context of the current formula.

Arguments of these functions (they are noted as `sVar`, `sPhysVar`, `vVar`, `vPhysVar`, `vVarComp`) are external variables relating to the current formula. They cannot be values or expressions specified in the current formula, data from **Characteristics** or constants.

Examples of acceptable use:

- `vec(grad(VEL_X).x; grad(VEL_Y).y; grad(VEL_Z).z)` is vector of gradient of the velocity. Arguments of the function `grad` are components of the local vector physical variable **Velocity**. The resulting vector is formed by the function `vec`, which is available in the tab **Special**.
- `grad(sVar).y` is the component along the axis y of the gradient vector of the scalar local variable `sVar`.
- `grad(VEL_Z).z` is the component along the axis z of the gradient vector of the vector physical variable **Velocity**.
- `rot(vVar).x` is the component along the axis x of the curl vector of the vector local variable `vVar`.

Examples of invalid use:

- `grad(sVar1+sVar2)`, where `sVar1` and `sVar2` reference to two local user variables. Argument of the function `grad` has to correspond to an existing user or physical variable but not to an expression calculated in the

formula. *How to fix:* create a user variable that will correspond to the sum of `sVar1+sVar2` and use it as the function's argument.

- `A=PRES+1000; grad(A).x` – also the argument of the function `grad` is calculated within the formula itself in the assignment "`A=PRES+1000;`". *How to fix:* create a user variable that will correspond to **Pressure+1000** and use it as the function's argument and remove "`A=PRES+1000;`" from the formula.

Operation	Identifier	Syntax	Description

Gradient	`grad`	`grad(sVar)` or `gradient(sVar)`; `grad(vVarComp)` or `gradient(vVarComp)`	Gradient of a scalar value or of a component of a vector value.
			This function returns a vector value. To obtain components (x, y, z) of the gradient vector, use the following syntax:
			`grad(sVar).x`
			`grad(sVar).y`
			`grad(sVar).z`
			For variables that are defined on surfaces only, the `grad` function returns zero values.

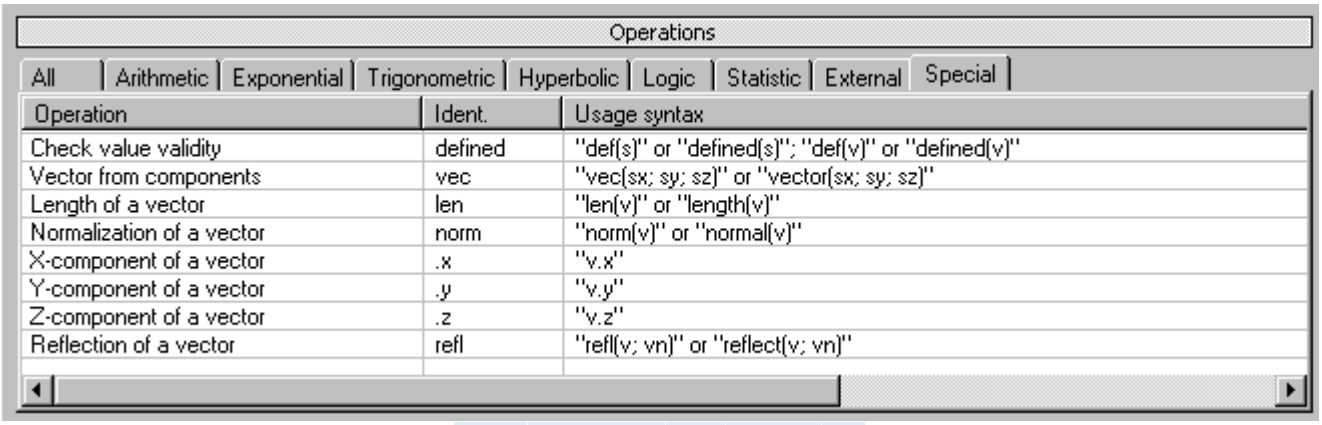
Curl	`rot`	`rot(vVar)` or `curl(vVar)`	Curl (rotor) of a vector value.
			This function returns a vector value. To obtain components (x, y, z) of the curl vector, use the following syntax:
			`rot(sVar).x` or `curl(sVar).x`
			`rot(sVar).y` or `curl(sVar).y`
			`rot(sVar).z` or `curl(sVar).z`

Ignore boundary condition	`no_bc`	`no_bc(sVar)` or `ignore_bc(sVar)`; `no_bc(vVarComp)` or `ignore_bc(vVarComp)`	These operations prevent taking a variable's value from a wall. So when the formula is calculated in a volume, they do nothing, but when the formula is calculated on a wall (when a boundary condition is set, when visualization on a **Supergroup**, etc.) these operations force taking the value from the volume instead of the wall. For example, `TEMP` means the temperature taken from the wall (defined by a boundary condition), while `ignore_bc(TEMP)` means the temperature taken from the center of adjacent cell. These operations allow defining a local variable on a surface of a boundary condition by the value from the center of the cell, which is adjacent to the surface, and not immediately from the surface of the boundary condition as it is doing by default. If on a boundary condition it is required to calculate a local variable's value, which depends on the value from the previous time step, then use of these operations is the only correct approach, because the value from the previous time step is not stored in the memory but calculated at each time step and so must not depend on itself. Also these operations allow use of values of velocity near a wall in characteristics, because velocity's values is zero immediately on the wall.
Ignore boundary condition (vector)	`no_bcvec`	`no_bcvec(vVar)` or `ignore_bc_vector(vVar)`	
Variable value from the previous step	`prev`	`prev(sVar)`	This is the value of a scalar or vector global user variable from the previous time step. This operation allows you, for any scalar or vector global user variable, to calculate: • time derivative

Operation	Identifier	Syntax	Description
Variable value from the previous step (vector)	`prevvec`	`prevvec(vVar)`	• integral over time At the first time of the computation the function returns the value, which was set by the "Default" parameter in properties of the user variable, which is argument of the function.

Tab «Special»

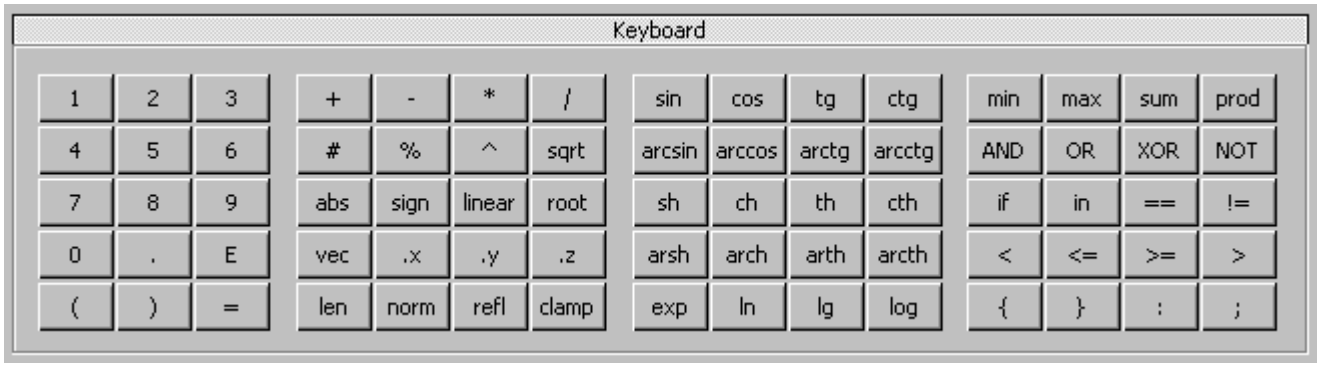
The **Special** tab contains special operations with vectors and their components used in *FlowVision*.



Panel **Operations**, the **Special** tab

Operation	Identifier	Syntax	Description
Check value validity	`defined`	`def(s)` or `defined(s)` or `def(v)` or `defined(v)`	A Boolean function, which checks if its argument is defined in a cell
Vector from components	`vec`	`vec(sx; sy; sz)` or `vector(sx; sy; sz)`	Specifying a vector with components x, y, z.
Length of a vector	`len`	`len(v)` or `length(v)`	Calculation of the length of the vector V
Normalization of a vector	`norm`	`norm(v)` or `normal(v)`	Normalization of the vector V
X-component of a vector	`.x`	`v.x`	x-component of the vector V
Y-component of a vector	`.y`	`v.y`	y-component of the vector V
Z-component of a vector	`.z`	`v.z`	z-component of the vector V
Reflection of a vector	`refl`	`refl(v;vn)` or `reflect(v;vn)`	Reflection vector V of the plane with the normal n. The operation resulted in the reflection vector refl (v1; vn) is the vector reflected from the plane with the normal vn, i.e. v1 2 * norm (vn) * (v1 # norm (vn)).

8.1.14.3 Keyboard of Formula editor



Keyboard of the **Formula editor** is a set of screen buttons that are used to specify a formula. The **Keyboard** includes:

- digits
- signs of mathematical operations
- most commonly used functions and operators

Button	Description
+	Addition
-	Subtraction
*****	Multiplication
/	Division
#	The scalar product
%	Vector product
^	Degree
sqrt	Square root
abs	The absolute value
sign	Sign
linear	Linear interpolation
root	Square specified degree. The degree of the root is specified as the second argument to the function, for example, the cube root of eight is denoted as `root(8,3)`.
vec	Vector component
.x	X-component of the vector
.y	Y-component of the vector
.z	Z-component of the vector
len	Length of the vector
norm	Normalization of the vector
refl	Reflection vector
clamp	Restriction
sin	The sine
cos	Cosine
tg	The tangent
ctg	Cotangent
arcsin	Arcsine
arccos	Arccosine

Button	Description
arctg	Arctangent
arcctg	Arccotangent
sh	Hyperbolic sine
ch	Hyperbolic cosine
th	Hyperbolic tangent
cth	Hyperbolic cotangent
arsh	Hyperbolic areasinus
arch	Hyperbolic areakosinus
arth	Hyperbolic areatangens
arcth	Hyperbolic areakotangens
exp	Exhibitor original
ln	Natural logarithm
lg	Logarithm of the decimal
log	Logarithm to the base
min	At least
max	Maximum
sum	Sum
prod	Artwork
AND	Logical "AND"
OR	Logical "OR"
XOR	Logical "exclusive OR"
NOT	Boolean "NOT"
if	Conditional branching
in	Joining the band
==	Comparison of "equal"
, =	Comparison of "not equal"
<	Comparison of the "lesser"
<=	Comparison of "less than or equal"
> =	Comparison of "greater than or equal"
>	Comparison of the "big"
{ }	Multiple branching
:	Separating "colon"
;	Separating "semicolon"

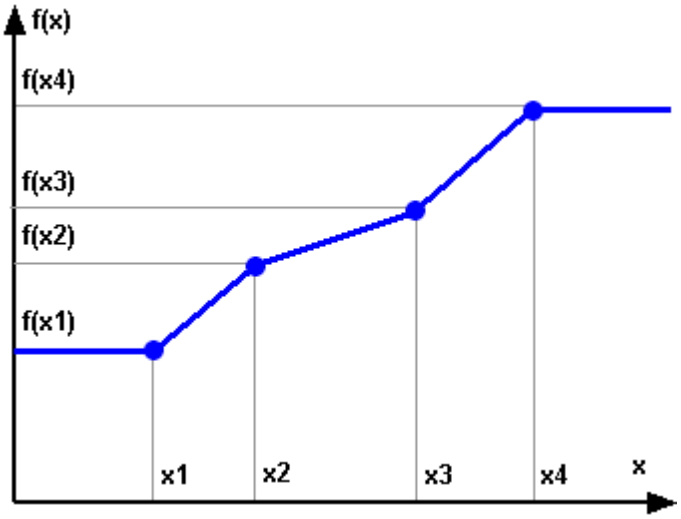
8.1.15 Table editor

Table editor is the component of **Pre-Postprocessor**, which is used to set values as table functions $f(x)$ or $f(x1, [x2, x3])$.

One version of **Table editor** allows you to specify a table function of only one argument; other version of **Table editor** allows specifying a function of several (from one to three) arguments.

The user specifies several groups of values $x1$, $[x2, x3]$, $f(x)$.

Values of $f(x1,[x2, x3])$ at intermediate points are linearly interpolated between values of $f(x1,[x2, x3])$ that are set in the table. If an argument locates beyond the table range, it will be replaced with the nearest table value.



Sample of constructing a table function of one argument

When **Table editor** is used to specify properties of **Substances**, *absolute* **Temperature** and **Pressure** are taken (see section Reference parameters, absolute and relative variables for details).

In other cases of use of formulae or tables, program takes the *relative* **Temperature** and **Pressure**.

The **Table editor** is implemented in the following versions:

- Editor of tables of only one argument, which opens for the expanded data input field with icon **Table f(x)**
- Editor of tables of several arguments, which opens for the expanded data input field with icon **Table f(x1;x2;x3)**

8.1.15.1 Editor of tables of only one argument

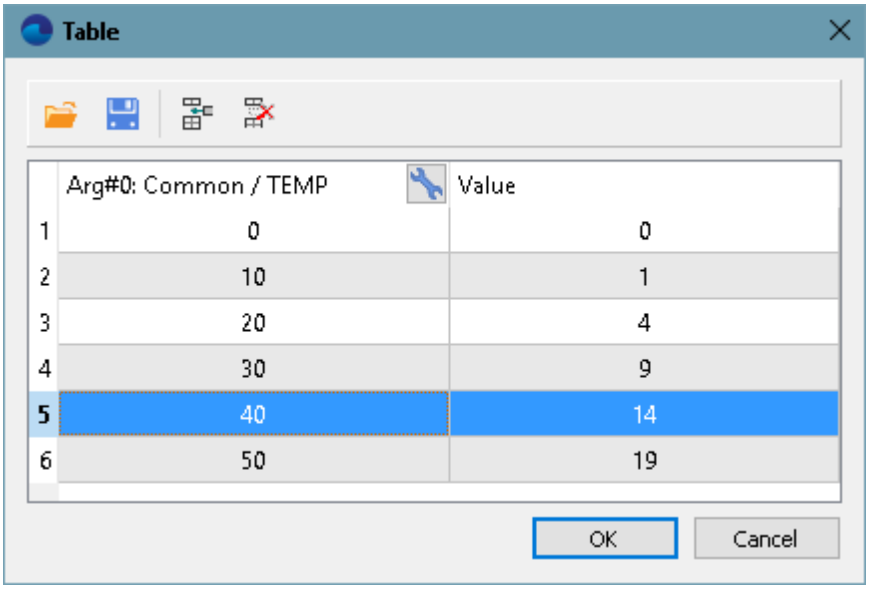


Table editor for a function of only one argument

Interface elements of the **Table editor** for tables of one argument:

Element	Description
(**Load Table**)	Loading the table from a text file[1)]

Element	Description
(**Save Table**)	Saving the table in a text file[1)]
(**Insert line before first of selected line**)	Inserting a line before the first selected line. If the table has no selected line(s), then the new line will be added to the end of the table.
(**Delete selected lines**)	Deleting the selected line(s). This button can be used when a line is selected.
The **OK** button	Accepting all the changes and closing the **Table editor**'s dialog box
The **Cancel** button	Canceling all the changes and closing the **Table editor**'s dialog box
	Opening the menu for specifying the argument of the table function. This menu contains the following command: • **Bind to variable**: select a variable that will be the argument of the table function (the Variable selection dialog box will open).

Note:

[1)] Such a file contains numerical data only but doesn't contain information about which variable is bound to the argument.

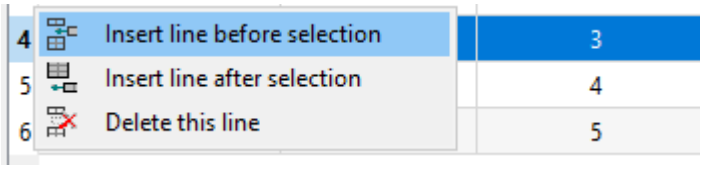
An inserted line is filled with interpolated or extrapolated data.

A table, which specifies the function of only one argument, contains the following columns:

Column*)	Description
(unnamed column)	Line numbers
Arg#0: ***variable_code***	Values of x, the argument of the function. When you place the mouse pointer over the column header, the pop-up tip duplicates the header's text; this in useful when the text is too wide and doesn't fit in the header.
Value	Values of the function $f(x1)$

*) Clicking on the header of a column with an argument or function values causes sorting the whole table in ascending or descending order by values in this column. Repeated clicks toggle the order of sorting. When some values in the column are equal, the sorting is carried out by values in the other column according to its previously specified order.

Context menu of a line in a table of only one argument



The context menu of a line in a table of several arguments opens by right-clicking on the selected line.

Command in the context menu	Description
Insert line before selection*)	Insert a line into the table before the selected line
Insert line after selection*)	Insert a line into the table after the selected line
Delete this line	Delete the selected line

*) The inserted line is filled with interpolated or extrapolated data.

Step-by-step procedure for specifying a table of only one argument

To specify a table function $f(x)$, follow these steps:

Step	Actions
1	Open the table by clicking the **Table f(x)** button in the Expanded data input field.
2	In the argument column click the button and from the menu, which opens, select **Bind to variable** and then in the Variable selection dialog box, which opens, select a variable (x).
3	Using the button at the window's top or either **Insert line before selection** or **Insert line after selection** command from the line's context menu, add the desired number of lines into the table.
4	Fill the table with data.
5	Click the **OK** button.
6	Click the **Apply** button in the **Properties** window.

8.1.15.2 Editor of tables of several arguments

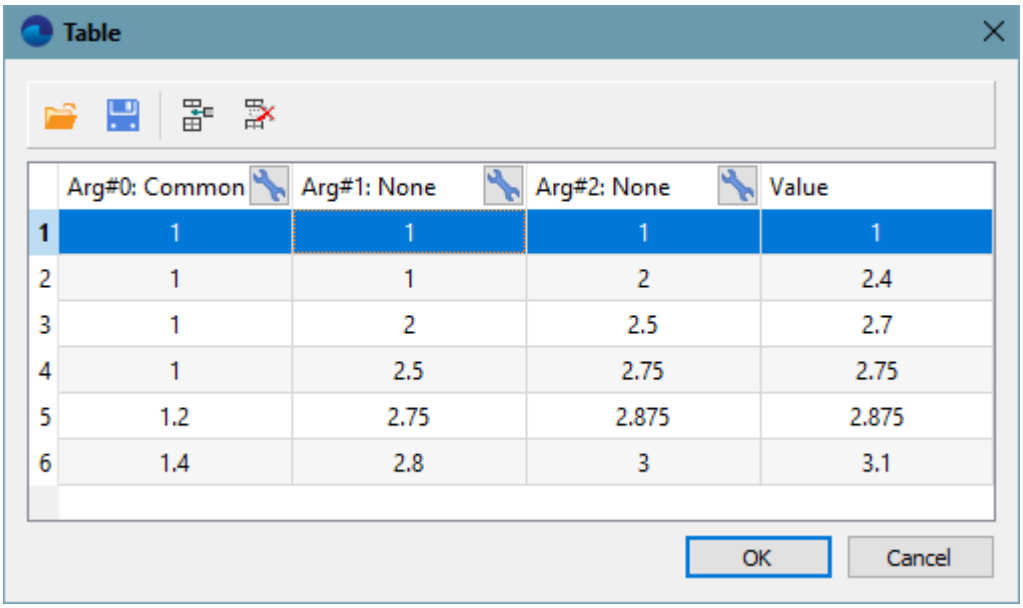


Table editor for a function of several (from 1 to 3) arguments

Interface elements of the **Table editor** for tables of several arguments:

Element	Description
(**Load Table**)	Loading the table from a text file[1)]
(**Save Table**)	Saving the table in a text file[1)]
(**Insert line before first of selected line**)	Inserting a line before the first selected line. If the table has no selected line(s), then the new line will be added to the end of the table.
(**Delete selected lines**)	Deleting the selected line(s). This button can be used when some line(s) are selected.
The **OK** button	Accepting all the changes and closing the **Table editor**'s dialog box
The **Cancel** button	Canceling all the changes and closing the **Table editor**'s dialog box

Element	Description
	Opening the menu for specifying the list of arguments of the table function of several arguments. This menu contains the following commands: • **Insert variable left of selection**: create an argument column in the table before the current column (on the left of it). • **Insert variable right of selection**: create an argument column in the table after the current column (on the right of it). • **Delete variable**: delete the current argument column from the table. This command is not available when the table has a single argument table. • **Bind to variable**: select a variable for the current argument column (the Variable selection dialog box will open).

Note:

[1] Such a file contains numerical data only but doesn't contain information about which variables are bound to the arguments.

An inserted line is filled with interpolated or extrapolated data.

A table, which specifies the function, contains the following columns:

Column*)	Description
(unnamed column)	Line numbers
Arg#0: ***variable_code***	One, two, or three columns that are filled with values of arguments $x1$ and optionally $x2$, $x3$. When you place the mouse pointer over the column header, the pop-up tip duplicates the header's text; this in useful when the text is too wide and doesn't fit in the header.
Arg#1: ***variable_code***	
Arg#2: ***variable_code***	
Value	Values of the function $f(x1,[x2,x3])$

*) Clicking on the header of a column with an argument or function values causes sorting the whole table in ascending or descending order by values in this column. Repeated clicks toggle the order of sorting. When some values in the column are equal, the sorting is carried out by values in the other columns with priority of the columns from left to right.

Context menu of a line in a table of several arguments

The context menu of a line in a table of several arguments opens by right-clicking on the selected line.

Command in the context menu	Description
Insert line before selection*)	Insert a line into the table before the selected line
Insert line after selection*)	Insert a line into the table after the selected line
Delete this line	Delete the selected line

*) The inserted line is filled with interpolated or extrapolated data.

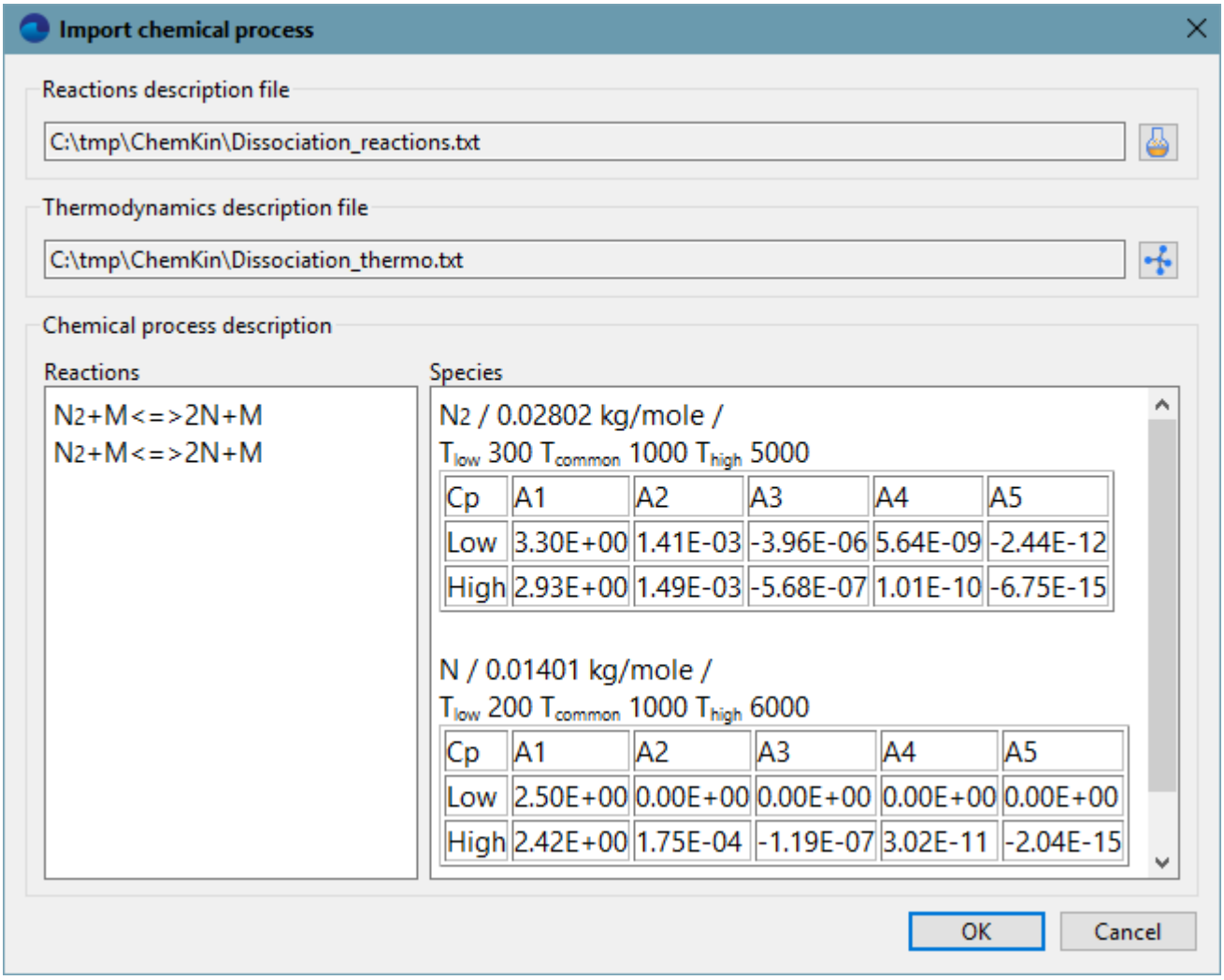
Step-by-step procedure for specifying a table of several arguments

To specify a table function $f(x1,[x2,x3])$, follow these steps:

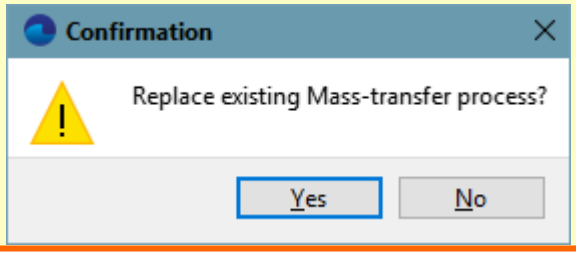
Step	Actions
1	Open the table by clicking the **Table f(x1;x2;x3)** button in the Expanded data input field. By default this table initially includes two argument columns and one function's values column.
2	In an argument column click the button and from the menu, which opens, select **Insert variable left of selection** or **Insert variable right of selection**. A column for the second argument will appear in the table. A column for the third argument will appear in the table. To remove an argument column, apply the **Delete variable** command.
3	In each argument column click and from the menu, which opens, select **Bind to variable** and then in the Variable selection dialog box, which opens, select a variable for $x1$, $x2$ or $x3$.
4	Using the button near the table's header add the desired number of table lines.
5	Fill the table with data.
6	Click the **OK** button.
7	Click the **Apply** button in the **Properties** window.

8.1.16 Importing a chemical processes

The form for importing chemical processes in the *CHEMKIN* format is opened by the **Import chemical process** command from the context menu of the folder **Phase #N**.



If the **Mass transfer** physical process is already set for the **Phase**, then attempt of importing chemical processes cause that the program requests you to confirm the import ("**Replace existing Mass-transfer process?**"):



The **Import chemical process** dialog box has the following fields and interface elements:

Element	Description
Reactions description file	This field displays the file (including the path to it), from which descriptions of chemical reactions will be imported. To select this file, click . If the file format is incorrect, an error message will be displayed.
Thermodynamics description file	This field displays the file (including the path to it), from which thermodynamic descriptions will be imported.

Element	Description
	To select this file, click  . If the file format is incorrect, an error message will be displayed. If the thermodynamic properties are not imported from a file, then properties of **Substances** that are already contained in the project will be applied.
Chemical process description > Reactions	List of chemical reactions that were imported from the **Reactions description file**
Chemical process description > Species	List of chemical substances that were imported from the **Reactions description file** and **Thermodynamics description file**
OK *(screen button)*	The program will import the data and then close the **Import chemical process** dialog box. After successful import from the files, the imported **Substances** will appear in the project tree (their physical properties are also imported). If the project tree contains **Substances** with same names, a dialog box opens, which request your confirmation to replace physical properties of these **Substances**.
Cancel *(screen button)*	The program will *not* import the data and then close the **Import chemical process** dialog box

Example of *CHEMKIN* files

Here are examples of files, from which it is possible to import data for simulating of nitrogen dissociation.

File with reactions, **Dissociation_reactions.txt**:

```
SPECIES
    N2 N
END
REACTIONS  SI
    N2+M<=>2N+M            192000000000 -0.5 113100
      N2/2.5/
      REV /                    10900 -0.5 0/

    N2+M<=>2N+M            4.15e+016  -1.5 113100
      N/1/
      REV /                    2320000000  -1.5 0/
END
```

File with thermodynamic properties, **Dissociation_thermo.txt**:

```
THERMO
N                 L 6/88N   1               G   200.000  6000.000  1000.000    1
 0.24159429E+01 0.17489065E-03-0.11902369E-06 0.30226245E-10-0.20360982E-14    2
 0.56133773E+05 0.46496096E+01 0.25000000E+01 0.00000000E+00 0.00000000E+00    3
 0.00000000E+00 0.00000000E+00 0.56104637E+05 0.41939087E+01                   4
N2                121286N   2               G   300.000  5000.000  1000.000    1
 0.02926640E+02 0.14879768E-02-0.05684760E-05 0.10097038E-09-0.06753351E-13    2
-0.09227977E+04 0.05980528E+02 0.03298677E+02 0.14082404E-02-0.03963222E-04    3
 0.05641515E-07-0.02444854E-10-0.10208999E+04 0.03950372E+02                   4
END
```

8.2 Operations in Pre-Postprocessor

This section describes operations that are carried out **Pre-Postprocessor**.

See sections:

- Forming a project and operations with a project
- Operations with the geometry model of the computational domain
- Operations in the View window
- Operations with elements of the project tree
- Controlling the project's calculation
- Analysis of project calculation's results
- Saving and loading the settings
- Operations with external parameters and exported results
- Operations with backup
- Work of Pre-Postprocessor in the read-only mode

8.2.1 Forming a project and operations with a project

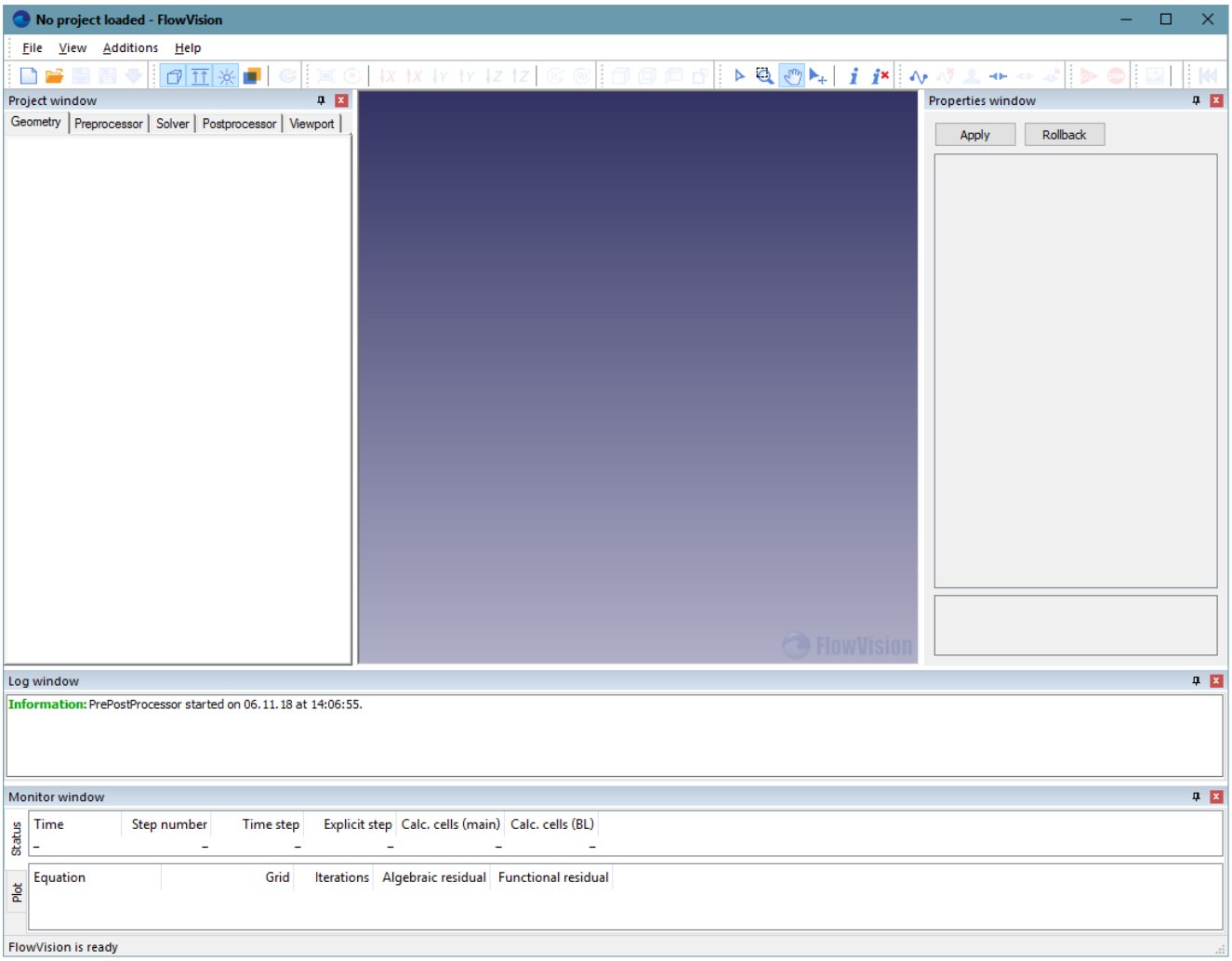
See sections below:

- Creation or loading a project
- Saving a project
- Opening a previously saved project
- Renaming a project
- Starting solve, stop and resuming the project's computation
- Closing a project

8.2.1.1 Creation or loading a project

Loading geometry model of the computational domain is required when a new project is created.

Immediately after its start, **Pre-Postprocessor** doesn't have any project loaded. **Pre-Postprocessor**'s windows are empty, only the **Log** window displays the **Pre-Postprocessor**'s start time.



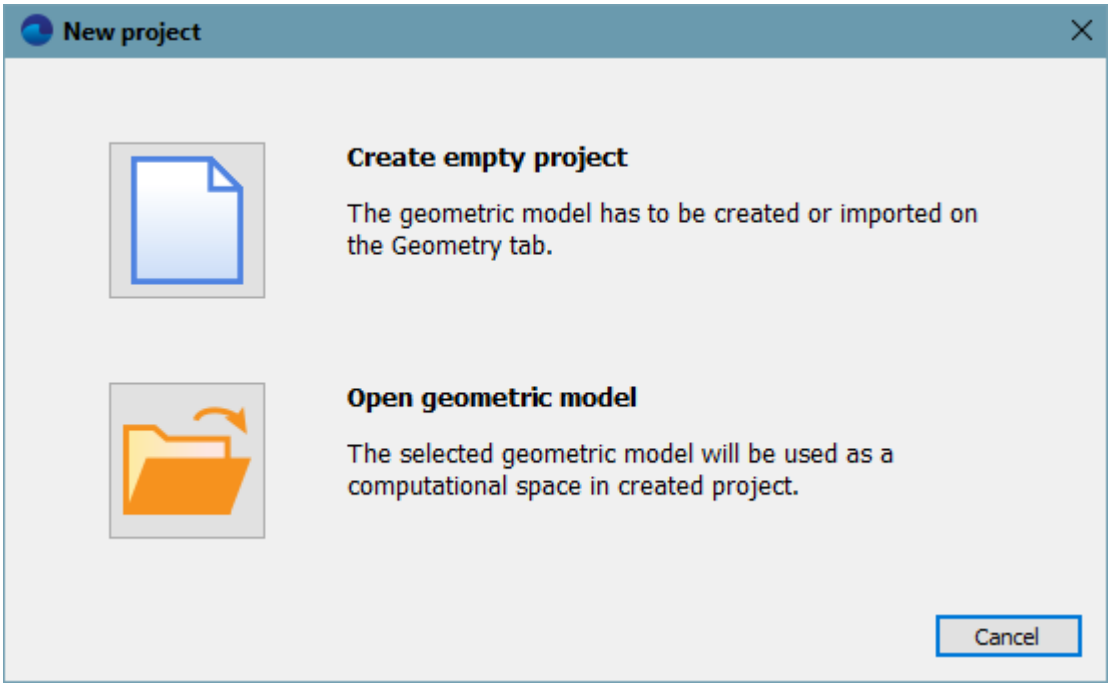
Pre-Postprocessor's window immediately after startup (before loading the geometry or loading a previously saved project)

After the **Pre-Postprocessor**'s start, you can:

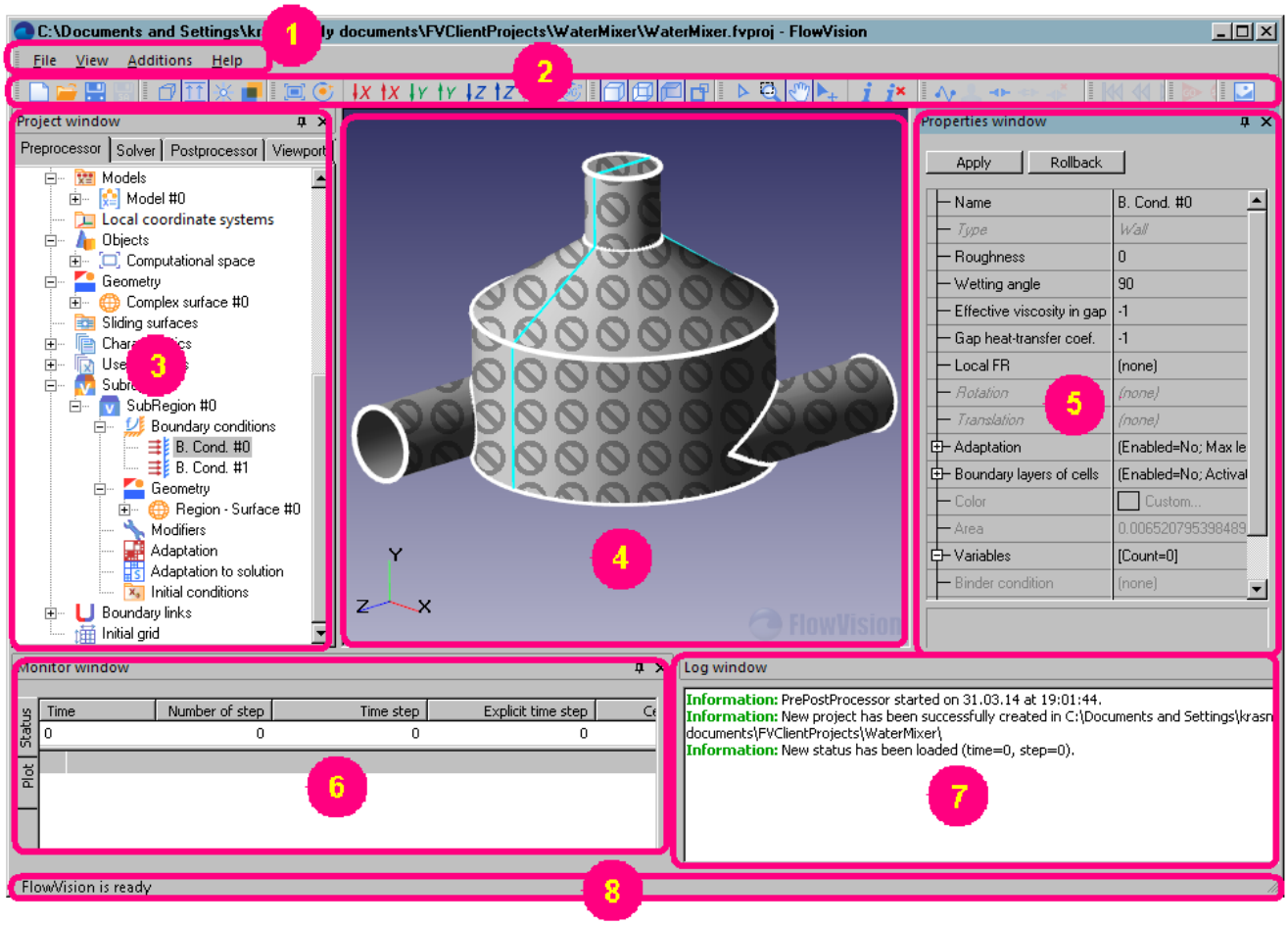
- create a new project and either don't download any geometry model or download a geometry model from a file.
- *or* open an existing project that was saved before.

When a new project is created, the **New project** dialog box opens where you select one of the options:

- **Create empty project**. An empty project will be created. The geometry model will be formed in the **Geometry** tab after beginning your work with the project.
- **Open geometric model**. The geometry model will be loaded from a a file (see section Loading a geometry model of the computational domain into a project).



After loading the geometry or previously saved project, **Pre-Postprocessor**'s windows will be filled with data.



Pre-Postprocessor's window (after loading the geometry or a previously saved project)

1 - the main menu; 2 - toolbars; 3 - the Project window; 4 - the View window; 5 - the Properties window; 6 - the Monitor window; 7 - the Log window; 8 - the status bar

After creation or loading a project, the program will create in the client part of the project a *subdirectory of the project* that contains files of the project's client part.

Some **Pre-Postprocessor**'s windows will be filled with data:

- the project tree will appear in the **Project** window (3)
- visualization of the geometry models will appear in the **View** window (4)
- a message about creation of the new project will appear in the **Log** window (7)

See also:

- Saving a project
- Opening a previously saved project

8.2.1.2 Saving a project

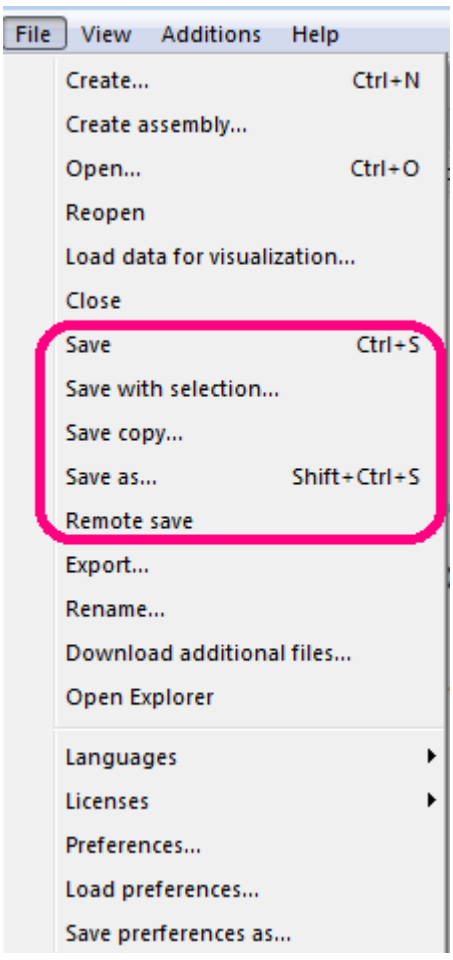
When working on a project is recommended that you periodically save it to be able to recover, for example, after a break in the work or after a random computer crashes.

Saving project data can be performed as in the directory of the client project and the server in the directory of the project. The client part of the project is available for **Pre-Postprocessor** directly, the server part of the project - just across the **Solver**. Therefore, preserving the client part of the project is always possible, and the server side of the project - only if the project is downloaded to the **Solver** (when **Solver** is connected to the project and **Pre-Postprocessor**).

To save the project, the following commands are available from the **File** menu:

- **File> Save** (it is duplicated by the button **Save changes to the client side of the project** in the toolbar **Standard** and by the hot keys **Ctrl+S**)
- **File>Save with selection**
- **File> Save copy**
- **File> Save as** (it is duplicated by the hot keys **Shift+Ctrl+S**)
- **File> Remote save** (it is duplicated by the button **Save solution on the solver** in the toolbar **Standard**)

Ability of performing these commands depends on the state of the project and **Solver**.



Command to save the existing project

Saving a project (command *File > Save*)

When you use the **File > Save** data client side of the project are stored in the client part of the project, and save the data in a directory server part of the project is carried out depending on the status of the project and the **Solver**:

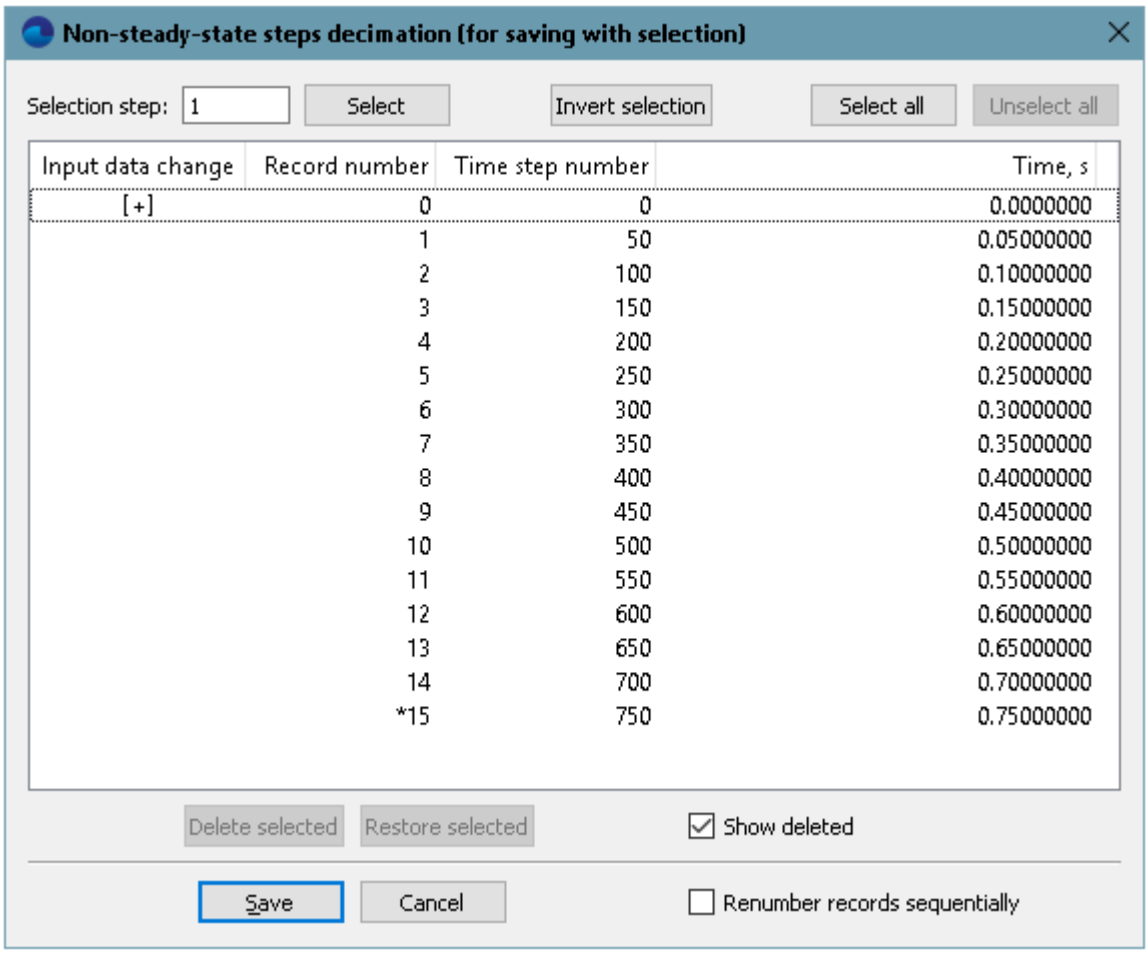
Saving the *server part* of the project by the command "File > Save"	
Status of the project and Solver	**What data are stored**
The project is not loaded on the **Solver**	Data saving is not done
Project has been loaded on the **Solver**, the computation of the project is not being performed	Saves all project data
Project has been loaded on the **Solver**, the computation is being performed	Only the following project data are saved: changes **Postprocessor**, project status, algebraic and functional discrepancies, display and configuration in the **View** window

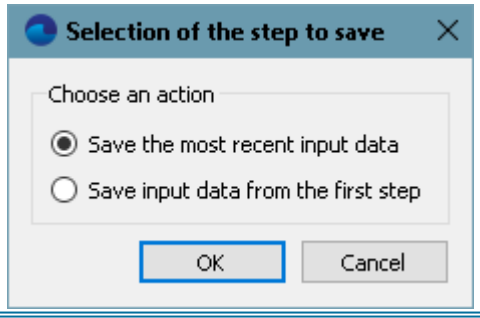
Selective saving a project (command *File> Save with selection*)

Selective saving a project, which is loaded on **Solver**, is, in fact, decimation of stored records in the event that it is necessary to reduce the number of stored records of the results on many time steps.

The command of selective saving a project, which is not loaded on **Solver**, is available with a client in the directory of the project saved more than one input file of the project that have been previously saved in the calculation of the project. Selective saving a project, which is not loaded on **Solver**, is, in fact, removing the directory client-side project of all stored records of the input data, except one. When you send the project on computation, these input data will be its starting data.

Selective saving a project *loaded* on Solver	
Step	**Description**
1	Use the command **File > Save with selection**. The **Non-steady-state steps decimation (for saving with selection)** dialog box will open:

Selective saving a project *loaded* on Solver	
	
2	In the **Non-steady-state steps decimation** window mark the records that you want to delete (select them in the list and click **Delete selected**), and then click **Save**. Marked entries will be deleted from the list. Abbreviated list of entries calculation results will be recorded instead of the old.

Selective saving a project, *not loaded* on Solver	
Step	**Description**
1	Use the command **File> Save with selection**. The dialog box **Selection of the step to save** will open: 
2	In the **Selection of the step to save** window, select: • **Save the most recent input data**, if all of the options previously stored input data necessary to leave the last option • **Save input data from the first step**, if all of the options previously stored input data necessary to leave the first option Then click **OK**.

Copy Project (command *File > Save copy* or *File > Save as*)

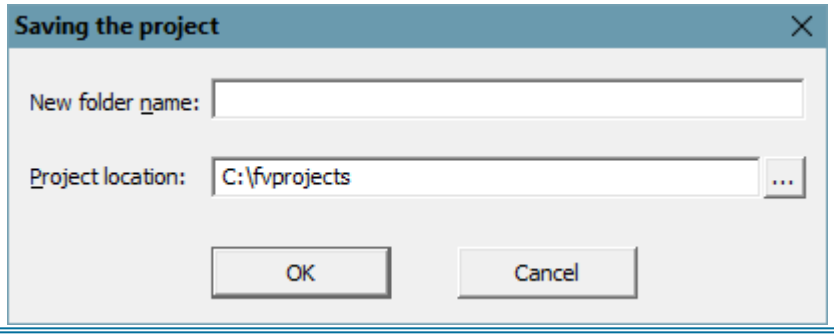
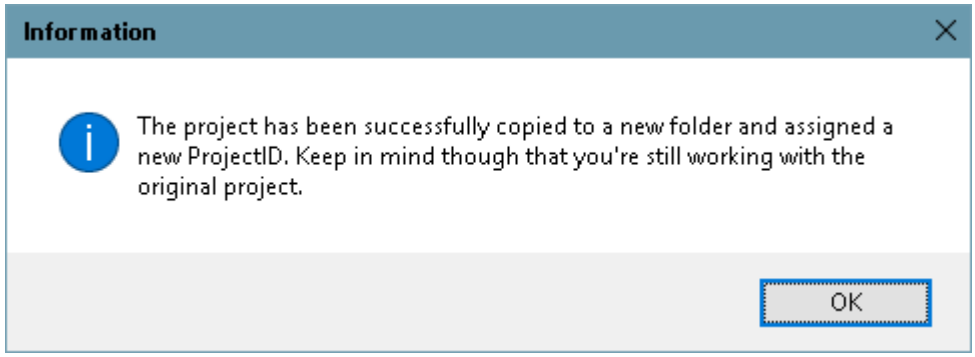
Commands **File > Save copy** and **File > Save as** copy a project under a new name.

These commands differ in the following:

- after copying using the **File> Save copy** command, the project remains loaded with its old name. The created project exists as a copy on a disk.
- after copying using the **File> Save as** command, loaded gets created project-copy (with a new name).

Made up:

- if the project is not loaded on the**Solver** - the client part of the project
- if the project is downloaded to the**Solver** -
  - both (client and server) parts of the project. This option is not available if there is any unsaved changes on the client side of the project and/or going calculation.
  - client-only project

Copy a project, which is *not loaded* on Solver	
Step	**Description**
1	Use a command **File > Save copy** or **File > Save as**. The **Saving the project** dialog box will open: 
2	In the **Saving the project** dialog box, specify: • the project's name in the **New folder name** field • if necessary, the project's location in the **Project location** field Then click **OK**. In this directory, a subdirectory will be created with a copy of the client part of the project. **Pre-Postprocessor** or continue to work with the project saved under a new name using **File > Save as**, or, in the case of **File > Save copy** a window opens with the message operation (**The project has been successfully copied to a new folder and assigned a new ProjectID. Keep in mind though that you're still working with the original project.**):  If the copy is command **File > Save copy**, then in **Pre-Postprocessor** the old project remains loaded. If the copy is command **File > Save as**, in the **Pre-Postprocessor** be loaded created project-copy.

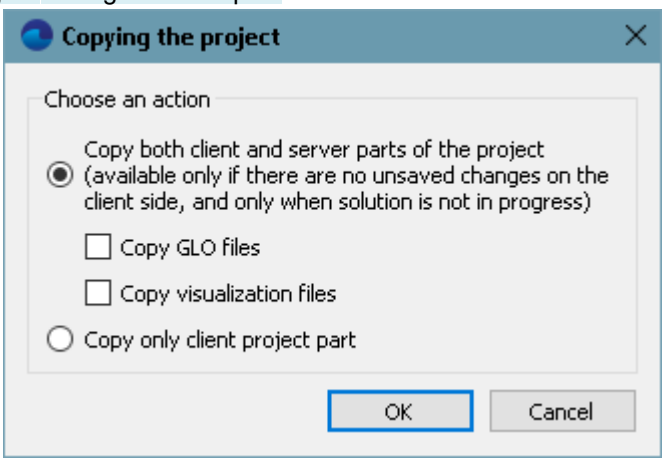
Copy *the client and server* parts of a project, which is loaded on Solver (Not available if there is any unsaved changes on the client side of the project and/or calculation is going on)	
Step	**Description**

Copy *the client and server* parts of a project, which is loaded on Solver
(Not available if there is any unsaved changes on the client side of the project and/or calculation is going on)

1

Apply the **File > Save copy** or **File > Save as** command.

The **Copying the project** dialog box will open:



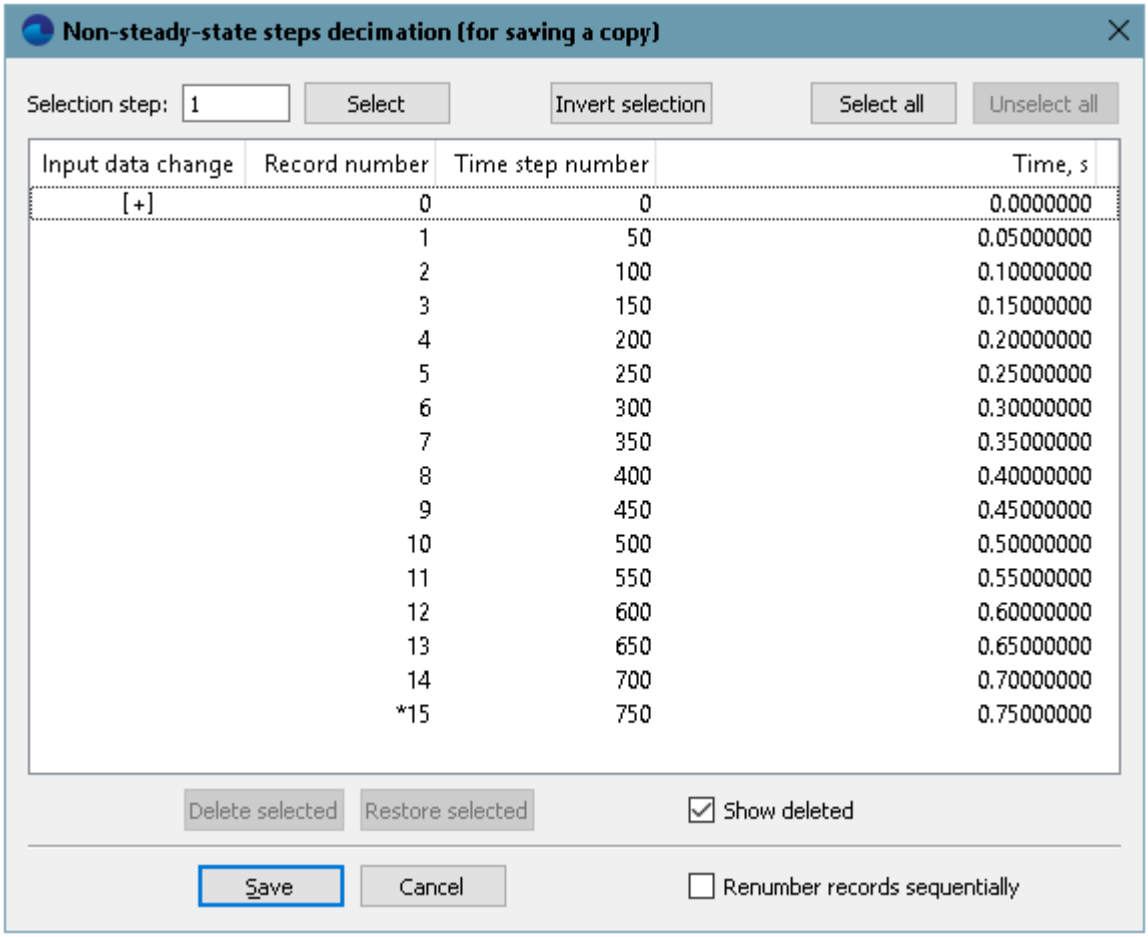
Select **Copy both client and server parts of the project**.

When you need to copy `glo` and/or `fvvis` files of the project, check the **Copy GLO files** and/or **Copy visualization files** checkboxes.

Then click **OK**.

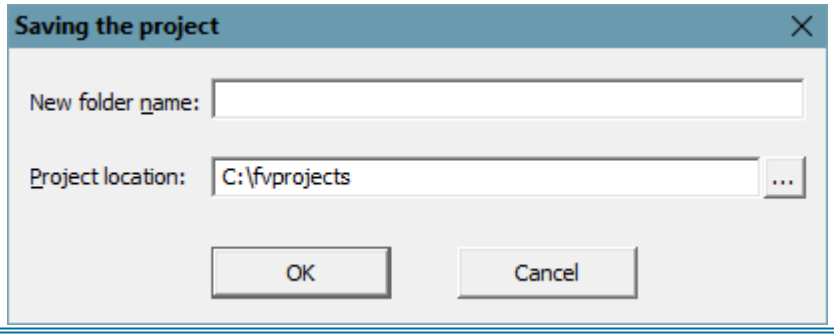
Copy *the client and server* parts of a project, which is loaded on Solver
(Not available if there is any unsaved changes on the client side of the project and/or calculation is going on)

2 *If* the project logged the history, the **Non-steady-state steps decimation (for saving a copy)** dialog box will open where you can select the records, which will not be included into the copy of the project (i.e., the records will be deleted from copy of the project).

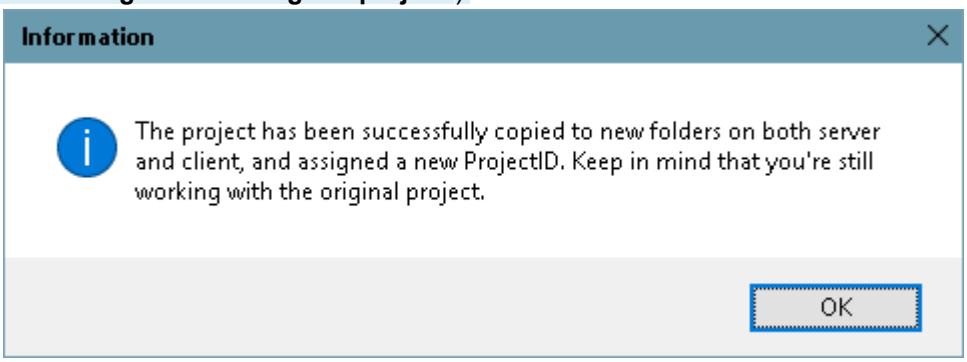


Mark the records that you wish to delete (select them in the list and click **Delete selected**), and then click **Save**.

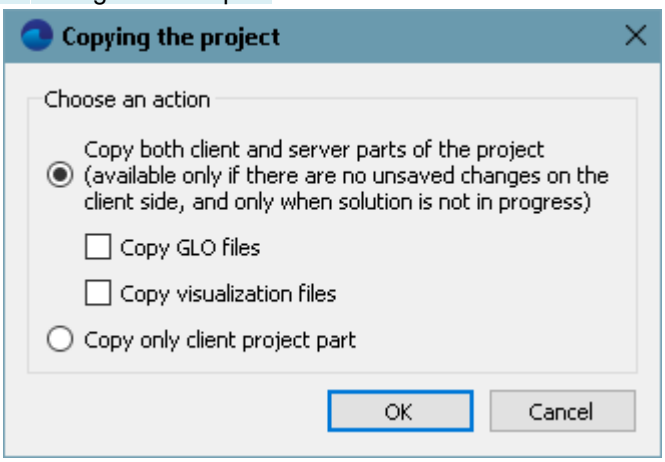
The **Saving the project** dialog box will open:

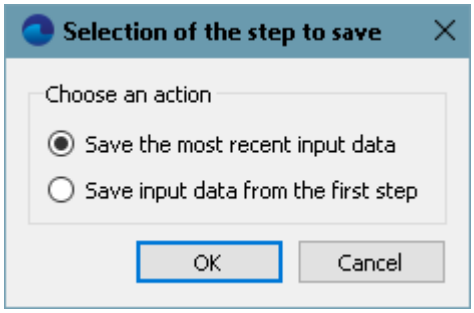
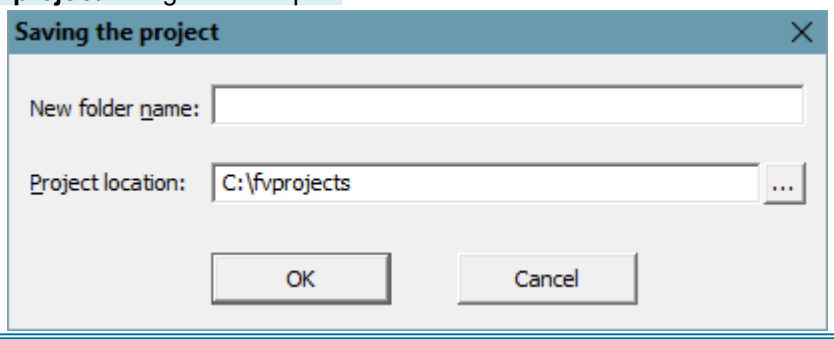
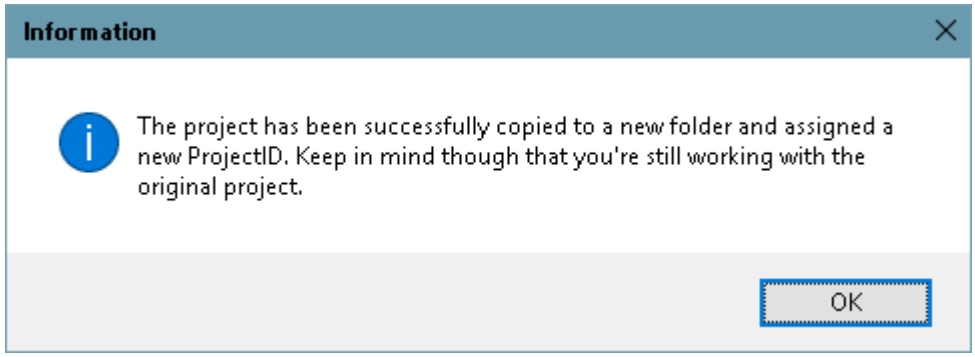


Copy *the client and server* parts of a project, which is loaded on Solver
(Not available if there is any unsaved changes on the client side of the project and/or calculation is going on)

3	In the **Saving the project** window, specify: • the project's name in the **New folder name** field • if necessary, the project's location in the **Project location** field Then click **OK**. In the specified directory, a subdirectory will be created containing a copy of the client part of the project. A copy of the server part of the project is written into a subdirectory with the same name in the directory of the server part of the project. **Pre-Postprocessor** will either continue its work with the project, which has been saved under a new name by the **File > Save as** command, or, if the **File > Save copy** command has been used, display a message informing completion of the operation (**The project has been successfully copied to new folders on both server and client, and assigned a new ProjectID. Keep in mind though that you're still working with the original project.**):  If the copy is command **File > Save copy**, then the old project remains loaded in **Pre-Postprocessor**. If the copy is command **File > Save as**, then the new project (the copy) will be loaded in **Pre-Postprocessor**.

Copy *only the client part* of the project, which is loaded on the Solver

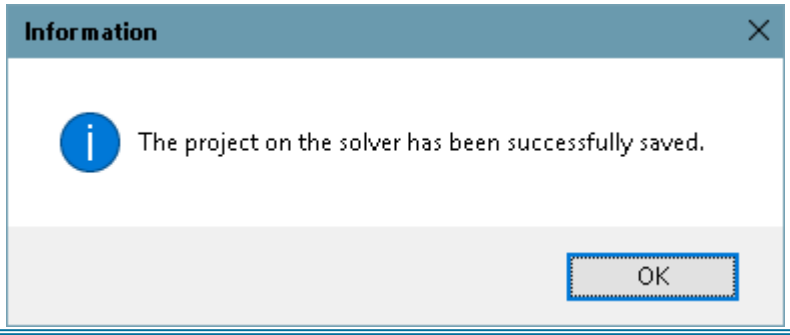
Step	Description
1	Apply the **File > Save copy** or **File > Save as** command. The **Copying the project** dialog box will open:  Select **Copy only client project part** and then click **OK**.
2	*If* a directory *is* the client part of the project has saved more than one input file of the project, previously stored in the calculation of the project, the **Selection of the step to save** dialog box will open:

	Copy *only the client part* of the project, which is loaded on the Solver
	 In the **Selection of the step to save** dialog box, select: • **Save the most recent input data**, if all of the options previously stored input data necessary to leave the last option • **Save input data from the first step**, if all of the options previously stored input data necessary to leave the first option Then click **OK**. The **Saving the project** dialog box will open: 
3	In the **Saving the project** dialog box specify: • the project's name in the **New folder name** field • if necessary, the project's location in the **Project location** field Then click **OK**. In this directory, create a subdirectory with a copy of the client part of the project. **Pre-Postprocessor** or continue to work with the project saved under a new name using **File >Save as**, or, in the case of **File > Save copy**, a message about completing the operation will be displayed (**The project has been successfully copied to a new folder and assigned a new ProjectID. Keep in mind though that you're still working with the original project.**):  If the copy is command **File > Save copy**, then the old project remains loaded in **Pre-Postprocessor**. If the copy is command **File > Save as**, then the new project (the copy) will be loaded in **Pre-Postprocessor**.

Saving project data on the Solver (command *File > Remote save*)

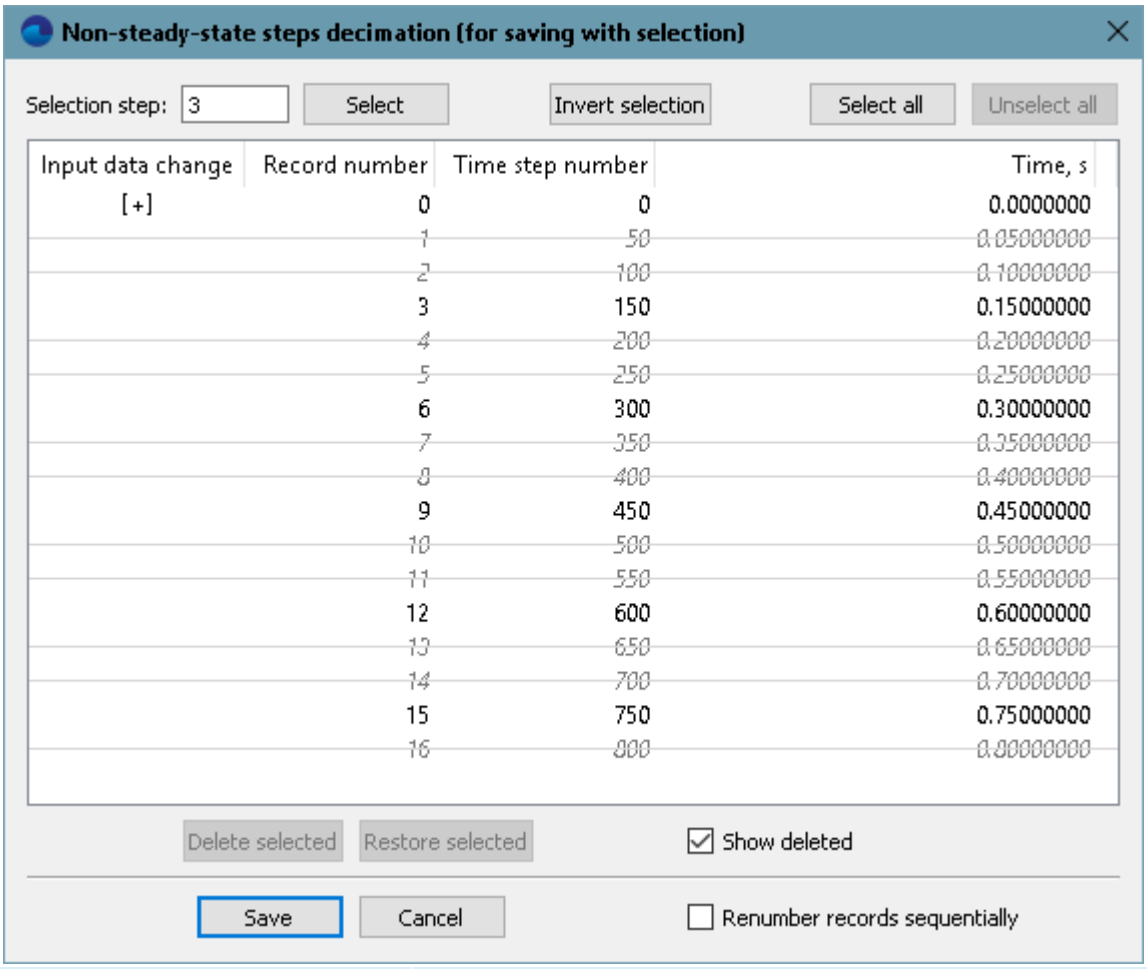
Saving the project data on **Solver** using the **File > Remote save** command, which is available when the project is loaded to **Solver**.

The results of the calculation are stored in the server part of the project files **`fvgrid`**, **`fvdata`**, **`fvstat`**.

Saving project data on the Solver	
Step	**Description**
1	Use **File > Remote save** or click in the **Standard** toolbar. Project data will be stored in the Solver and will see a message that this (**The project on the solver has been successfully saved**): 

See also: Opening a previously saved project.

Dialog box «Non-steady-state steps decimation»



The **Non-steady-state steps decimation** dialog box. Records that are planned to be deleted are indicated with gray italics and strikethrough.

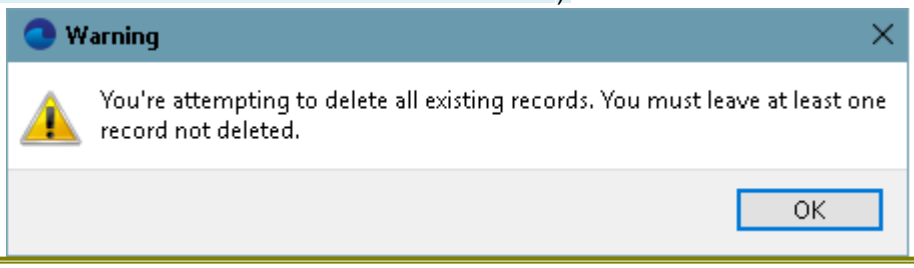
The **Non-steady-state steps decimation** dialog box allows deleting from both server and client parts of the project some records that were made when the computation's history was enabled:

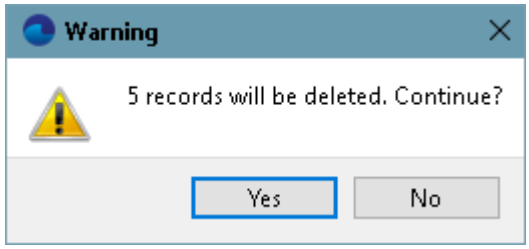
- when results of the computation are saved with decimation (this is done by the **File > Save with selection** command from the Main Menu).
- when a project is copied (this is done by the **File > Save copy** or **File > Save as** command from the Main Menu with further selection **Copy both client and server parts of the project** in the **Copying the project** dialog box).

The title of the **Non-steady-state steps decimation** dialog box is appended with either "**(for saving with selection)**" or "**(for saving a copy)**" text accordingly.

The **Non-steady-state steps decimation** dialog box can only open when the project *is connected* to **Solver**.

The **Non-steady-state steps decimation** dialog box has the following elements:

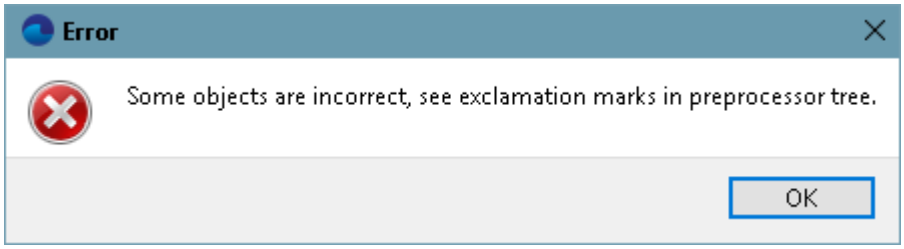
Element	Description
Table with records	This table contains the list of existing records that make up the history of the computation. This table contains 4 columns, see their descriptions below. Selecting records can be made using the mouse and the computer's keyboard keys. Selection using the mouse: • **Shift+(mouse click)** – selecting a range of records • **Ctrl+(mouse click)** – selecting some records in addition to already selected ones Selection without use of the mouse: • **space** – selection • **Shift+↑** (arrow up) or **Shift+↓** (arrow down) – moving with selection • **Ctrl+↑** (arrow up) or **Ctrl+↓** (arrow down) – moving without selection
Input data change *(column in the table with records)*	The "**+**" mark in this column means that input data have been changed in the marked record. If such record is to be deleted, its input data are assigned to the next record, which is not planned to be deleted.
Record number *(column in the table with records)*	The record's number
Time step number *(column in the table with records)*	The record's time step. When records are deleted, values of the time steps of the remaining records do not change.
Time, s *(column in the table with records)*	Simulated time of the record, [s]
Selection step *(input field)* and **Select** *(button)*	Selecting each n-th record. The n number is specified in the **Selection step** field.
Invert selection *(button)*	Inverting the selection. This is duplicated by the "*****" key on the keyboard.
Select all *(button)*	Selecting all records. This is duplicated by the **"Grey+"** key on the keyboard.
Unselect all *(button)*	Unselecting all record selections. This is duplicated by the **"Grey-"** key on the keyboard.
Delete selected *(button)*	This button *prepares deleting* of the all selected records: the selected records will be marked as not active (planned to to be deleted) using gray italics and strikethrough. Also, if required, the "**(+)**" mark in the **Input data change** column will be moved to the next record, which is not planned to be deleted. This button is duplicated by the **Delete** key on the keyboard. When you try to delete all records a warning message opens informing about impossibility of such operation (**You're attempting to delete all existing records. You must leave at least one record not deleted.**): 

Element	Description
Restore selected *(button)*	This button cancels *preparation for deleting* for the selected records. This is duplicated by the **"Insert"** key on the keyboard. The records, which are planned to be deleted, are displayed (with gray italics and strikethrough) only when the **Show deleted** checkbox (see below) is selected. When the **Show deleted** checkbox is unselected, the records, which are planned to be deleted, are not displayed in the list at all.
Show deleted *(checkbox)*	This checkbox specifies whether to show or hide the records, which are planned to be deleted.
Save *(button)*	This button deletes all the records, which are planned to be deleted, and closes the dialog box. After clicking the button a warning opens with information about the number of the records to be deleted and a request to confirm the operation (**"N records will be deleted. Continue?"**):  If all records are marked as planned to be deleted, this button is not active (because deleting all records is not allowed).
Cancel *(button)*	This button closes the dialog box without applying the changes.
Renumber records sequentially *(checkbox)*	When this checkbox is selected, the records that remain in the computation's history will be renumbered after clicking the **Save** button.

Specifics of saving a project with incorrect objects: it is blocked until Pre-Postprocessor is connected to a Solver

When some objects in the project are in-completed and/or incorrect (they are marked by "**!**" symbols in the project tree), you can not save the project until **Pre-Postprocessor** is connected to a **Solver**.

When **Pre-Postprocessor** is connected to a **Solver**, an attempt of saving such a project will cause an error message **"Some objects are incorrect, see exclamation marks in preprocessor tree"**:

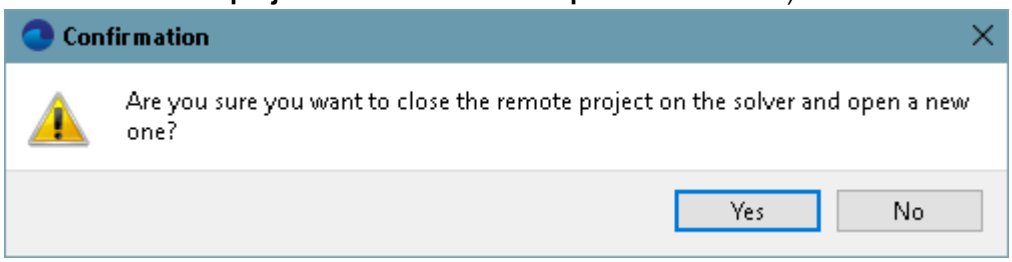


8.2.1.3 Opening a previously saved project

If a project has been saved, you can opened it again and continue to work with it.

To do this, apply the command **File> Open** (duplicated by the button in the **Standard** toolbar and by the **Ctrl+O** keyboard combination).

Opening a saved project	
Step	**Description**
1	Apply the command **File > Open** or click in the **Standard** toolbar.

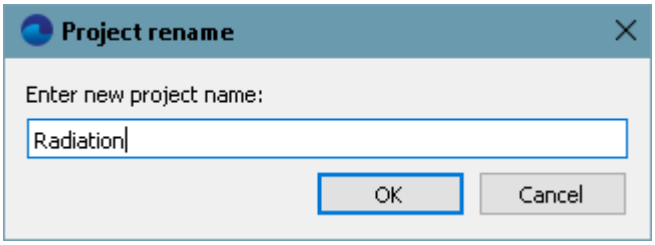
Opening a saved project	
	If the current project is loaded on **Solver**, the program will open a confirmation request ("**Are you sure you want to close the remote project on the solver and open a new one?**") asking you to confirm your decision to close the remote project on **Solver** and to load a new project ("**Are you sure you want to close the remote project on the solver and open a new one?**"):  To confirm your decision, click **Yes**.
2	A standard operating system's dialog box for selecting files will open. Select the file, in which the project is saved, and click **Open**. The previously saved project will be loaded.

Also, a project can be opened by dragging its folder or file (`*.fvproj`) by mouse from *Windows Explorer* to the window of **Pre-Postprocessor**.

8.2.1.4 Renaming a project

To rename a project, use the command **File > Rename**.

Pre-Postprocessor will request you to specify a new name for the project:



Enter a new name and click **OK**.

8.2.1.5 Starting solve, stop and resuming the project's computation

Running on the computation made after the project Define the problem, calculation parameters, and preferably, the parameters display the results.

Before the project's computation, the following preliminaries are to be done:

- registering a user on **Solver agent**
- running **Solver**
- loading the project on **Solver**

Running the computation can be started:

- from **Pre-Postprocessor**
- from **Terminal**
- in **Batch mode**

Way to start the calculation depends on the system configuration. In order to run the project on the computation:

- log on **Solver-Agent**
- download the project to the **Solver**
- run **Solver** on computation

If you have any problems downloading the project **Solver**:

- read the description of the problems encountered during loading and displayed in the Solver's console window and the error file (`err`), and try to solve their own
- If the problem is not solved, send the project, including `log` and `err` files, and file with diagnostic information to the technical support of FlowVision with a description of the problems

If you encounter problems when running the project on the computation:

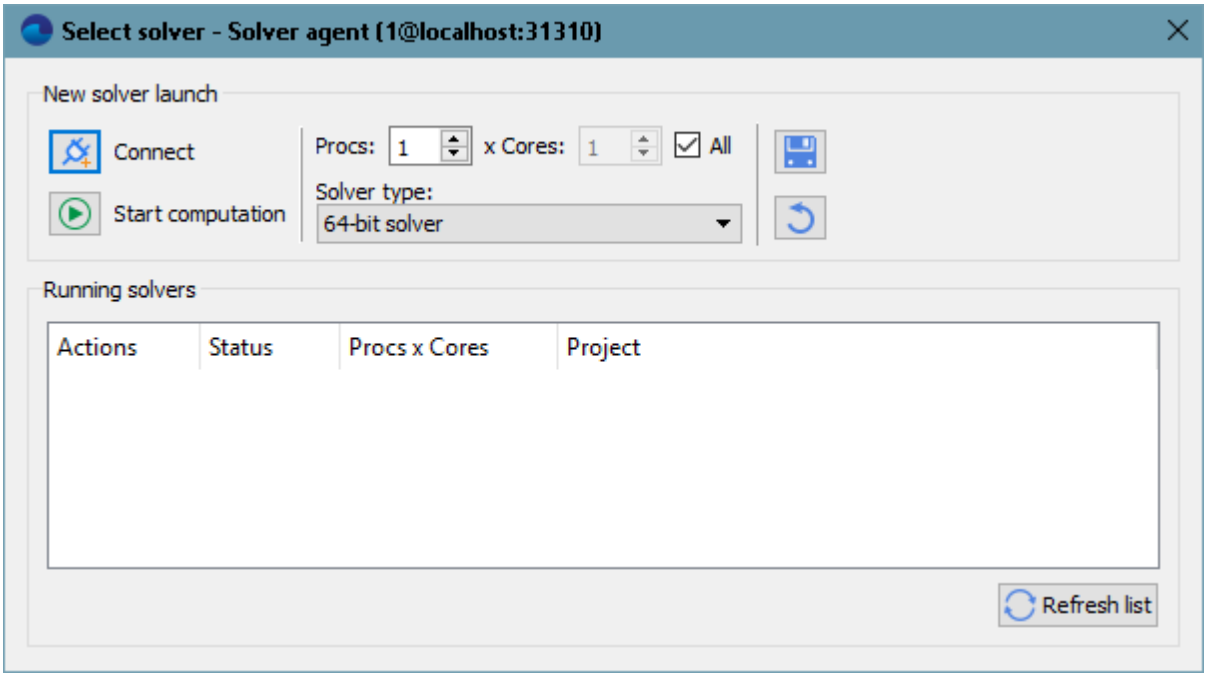
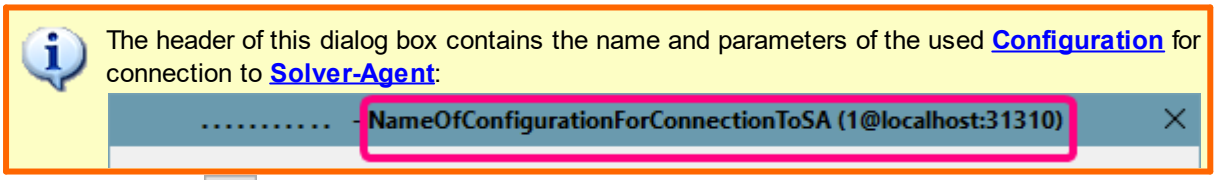
- make sure that the project is set correctly

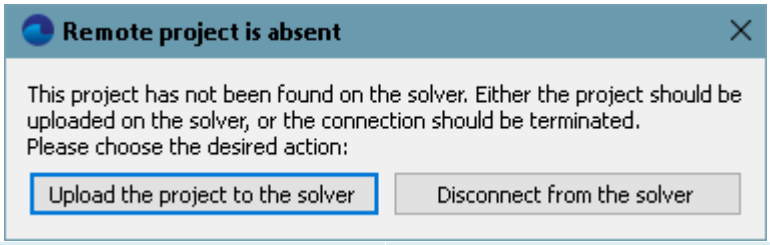
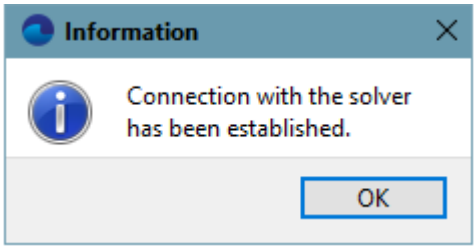
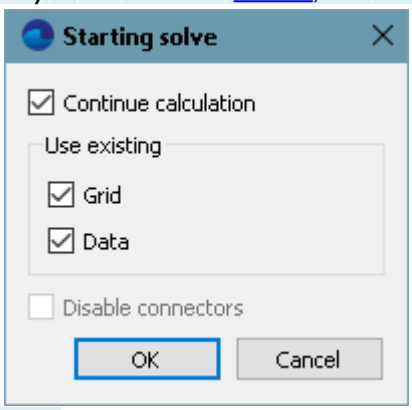
- make sure that interaction between **Solver** and the **License Manager** is configured correctly
- If the problem is not solved, send the project, including **log** and **err** files, and file with diagnostic information in technical support of FlowVision with a description of the problems encountered

If you encounter problems during the calculation:

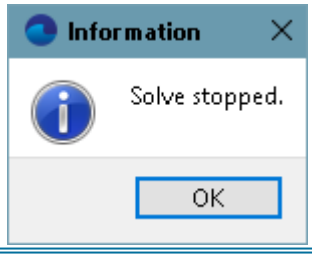
- make sure that the project is set correctly
- read describing the problems encountered in the calculation and display of the Solver's console window and the error file (**err**)
- send the project, including **log** and **err** files, and file with diagnostic information in the technical support of FlowVision with a description of the problems encountered

First computation of a project

Step	Actions
1	Before starting the computation, save the project by clicking the (**Save changes to the client side of the project**) button in the toolbar.
2	If automatic user authentication on **Solver-Agent** has not been done by the program, log in **Solver-Agent** manually. See details in the section Connection to Solver-Agent and user authentication on Solver-Agent.
3	Load the project on a **Solver**: a) click the button (**Open solver selection window**) in the **Network** toolbar; the **Select solver** dialog box will open:   b) click the **Connect** button; a new line corresponding to the new **Solver** will appear in the **Running solvers** list; c) highlight it in the list of **Solver** (click it in the column **Solver**), and then click **Connect**; dialog box **Remote project is absent** is displayed, there is no directory server project created in the client directory:

Step	Actions
	 d) Click the **Upload the project to the solver** button; a message box will open informing you about successful uploading the project on **Solver** ("**Connection with the solver has been established**"):  e) click **OK**.
4	Start computation of the project: a) click **(Start computation)** in the **Network** toolbar; the **Starting solve** dialog box will open:  b) uncheck **Continue calculation** (as made the first start of calculation of the project), and then click **OK**.

Stopping the project's computation

Step	Actions
1	Click **(Stop computation)** in the **Network** toolbar. A window opens with the message **"Solve stopped"**, click there on **OK**: 

Resuming the project's computation

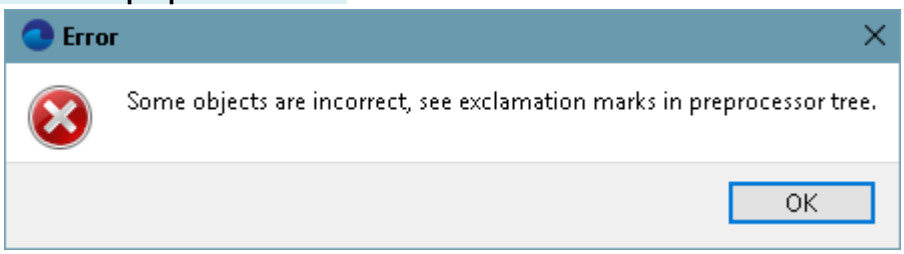
If the project is not disconnected from a **Solver**, then proceed to step **4** of the procedure *First computation of a project* (see above) and do not unselect **Continue calculation** in the **Starting solve** dialog box. Otherwise, repeat the *First computation of a project* procedure.

If you want to continue the computation (when there are no dramatic changes in the project), keep selected all checkboxes in the **Starting solve** dialog box.

If you want to start the computation from its beginning, unselect all checkboxes in the **Starting solve** dialog box.

Blocking of starting the computation with incomplete or incorrect data

In case of incomplete or incorrect data on the project start-up computation is not made, but at the respective nodes of the tree of the project will be the "**!**" and will display the error message **"Some objects are incorrect, see exclamation marks in preprocessor tree**":

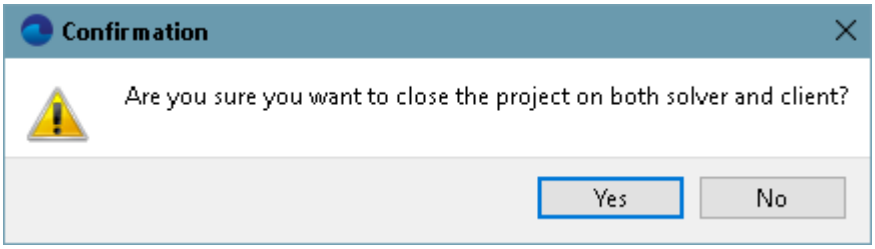


For example, if no **Model** is defined in any **Subregion**, then the "**!**" symbol will appear in the project tree near the **Region** root folder.

8.2.1.6 Closing a project

When closing a project **Pre-Postprocessor** returns to the state without the downloaded data, such as immediately after its launch.

To close the project, use the command **File > Close**.

Closes the current project	
Step	**Description**
1	Apply the **File > Close** command. *If* the current project is loaded on **Solver**, the program request you to confirm your decision to close the project on both client and **Solver** ("**Are you sure you want to close the project on both solver and client?**"):  To confirm your decision, click **Yes**. The project will be closed on the client and on **Solver**, and windows of **Pre-Postprocessor** (except the **Log** window) will be empty.

8.2.2 Operations with the geometry model of the computational domain

Surface geometry of the computational domain and the moving bodies is loaded from any geometry modeling systems are in *FlowVision* based **Objects**.

More information about geometry models, consisting of embedded surfaces and surfaces without or with *multiconnection* see in the section Multiconnection.

Among the operations geometry model of the computational domain include:

- Loading a geometry model of computational domain into a project
- Replacing a geometry model of computational domain in a project
- Adding a surface into a geometry model of computational domain
- Deleting a surface from a geometry model of computational domain
- Checking a geometry model of computational domain and moving bodies for self-intersections
- Fixing self-intersections of a surface in a geometry model

- Removal of too-small facets of geometry model of computational domain
- Exporting a geometric model into a file (step-by-step procedure)
- Procedure of regrouping a geometric model of computational domain (and moving bodies)
- Transformation of geometry model of the computational domain (and moving bodies)

In the future, a geometry model, loaded in the project, you can:

- regroup
- replace
- transform
- add to it other geometry models

8.2.2.1 Loading a geometry model of the computational domain into a project

The geometry model of the computational domain can be loaded into the project from:

- one file
- *or* multiple files (an assembly)

The loading can be done using the commands:

- **File > Create** (it is duplicated by the button in the toolbar **Standard** and, by default, the **Ctrl+N** hot key).
- **File > Create assembly**

The geometry it transferred into *FlowVision* using one of the following standard formats:

1. surface mesh: *VRML*, *STL*, *MESH* (internal format);
2. volume mesh on which to construct the corresponding *FlowVision* surface mesh: *ANSYS*, *NASTRAN*, *ABAQUS*, *NGEOM*.

Surface mesh is a triangulated (consisting of triangles) surface that approximates the original parametric surface. In *FlowVision* precision of a triangulated surface can be changed only in the direction of coarsening by using the **Geometry import > Tolerance** to the **The basic settings** (This option is valid only on the surface, shown in *VRML* format and *STL*). When loading geometry is automatically split into groups.

Group- a set of facets. The facets are combined in the **Group** in order to make it convenientto arrange the boundary conditions on the surface geometry.

When downloading information about the units in which the geometry was created, is ignored. The units of measurement are selected meters.

To avoid confusion, it is recommended to use the geometry given in meters.

Splitting into groups depending on the file format

Grouping depends on the file format:

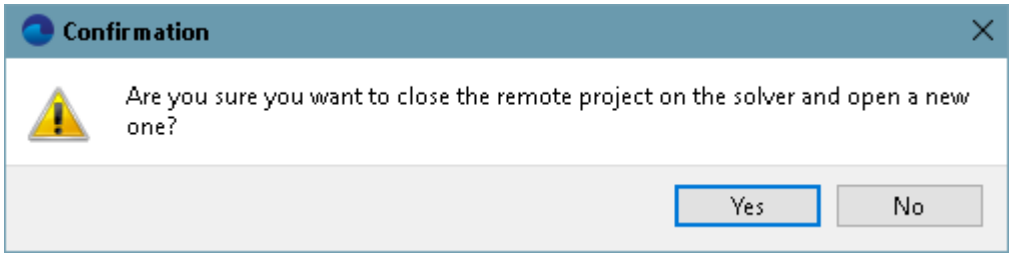
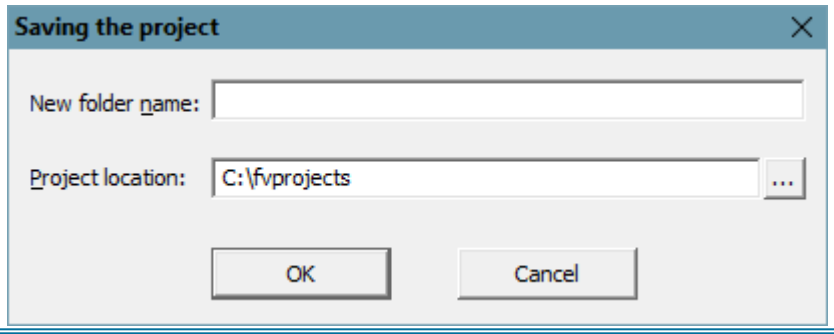
1. *VRML*: In **Geometry import** of the **basic settings**, **Tolerance** is only applied. Grouping is made on the surface color. For each color creates its boundary condition. Boundary conditions are placed automatically.
2. *STL*: The accuracy with which controlled geometric characteristics of geometry, and the criteria for **grouping** are defined in the **Geometry import**. Surface color information is not supported, respectively, automatic alignment of the **boundary conditions** on the **Group** does not.
3. *MESH*: **Geometry import** settings do not work. Grouping is contained in the source file. For each color creates its **boundary condition**. **Boundary conditions** are placed automatically.

If automatic **grouping** does not allow to place **Boundary conditions** in the desired manner, it is possible to carry out the **Regrouping**.

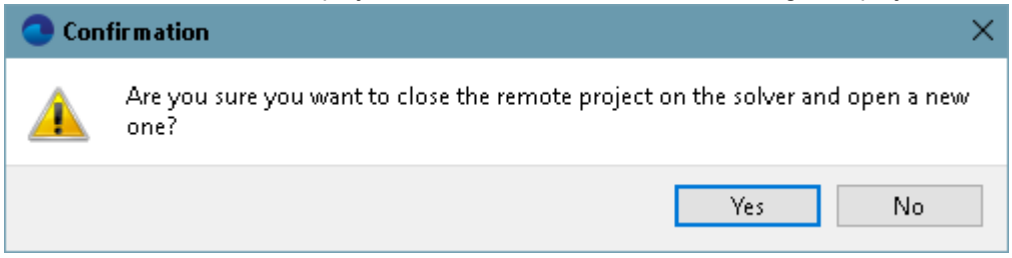
See also: Requirements to contents and formats of geometry files.

Loading geometry model of a single file

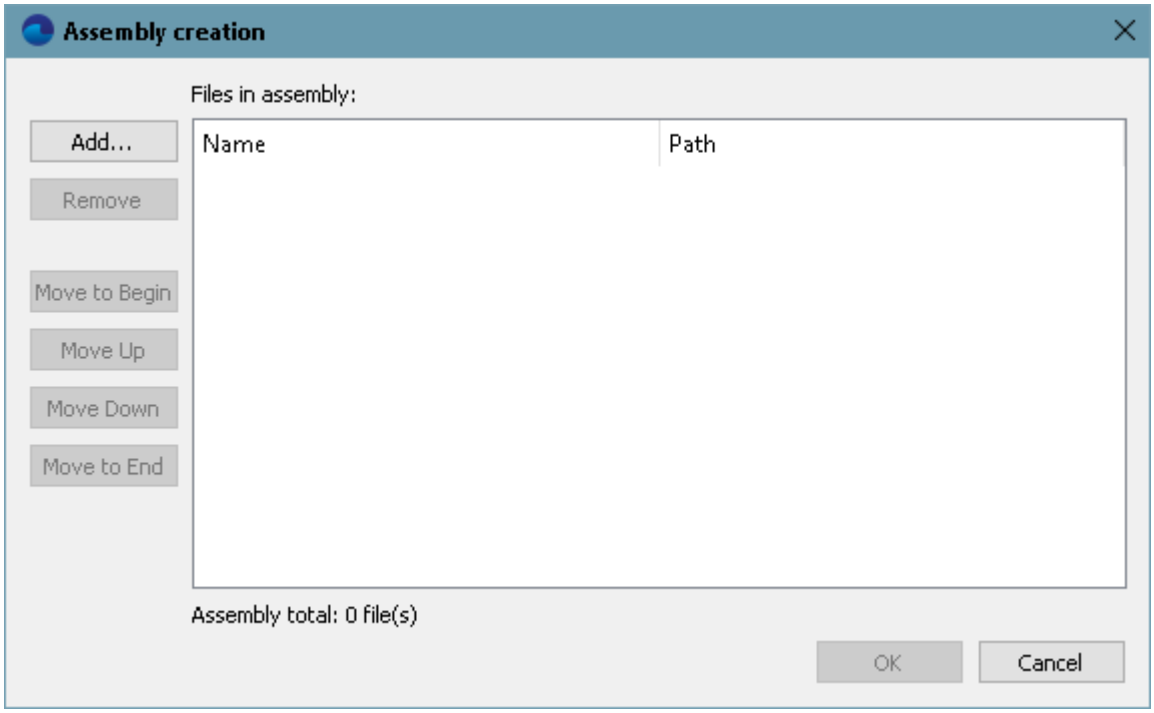
Loading geometry model of a single file	
Step	**Description**
1	Apply the **File > Create** command or click the button in the **Standard** toolbar. *If the* current project has been loaded on **Solver**, the program will open a confirmation request ("**Are you sure you want to close the remote project on the solver and open a new one?**") asking you to confirm your decision to close the project on the remote **Solver** and load the new project:

Loading geometry model of a single file	
Step	**Description**
	 To confirm your decision, click **Yes**. A standard operating system's dialog box for file selection will open.
2	In this dialog box select the required file of the geometry model and click the **Open** button. The **Saving the project** dialog box will open requesting you to specify the name and location of the project: 
3	In the **Saving the project** dialog box specify: • the project's name in the **New folder name** field • if necessary, specify, in the **Project location** field, the project's location Then click **OK**. In the client part of the project will create a subdirectory of the project that contains the client files of the project. Windows of **Pre-Postprocessor** will be filled with data matching the loaded geometry model.

Loading geometry model (assembly) from multiple files

Loading geometry model of multiple files (assembly)	
Step	**Description**
1	*Assembly* is the geometry model developed by combining the geometry models of the various files. Use the **File > Create assembly** command. *If* the current project has been loaded on **Solver**, the program will open a confirmation request ("**Are you sure you want to close the remote project on the solver and open a new one?**") asking you to confirm the decision to close the project on the remote **Solver** and loading new project:  To confirm your decision, click **Yes**. The **Assembly creation** dialog box will open, which is used to compose the list of the assembly's files:

Loading geometry model of multiple files (assembly)	
Step	**Description**

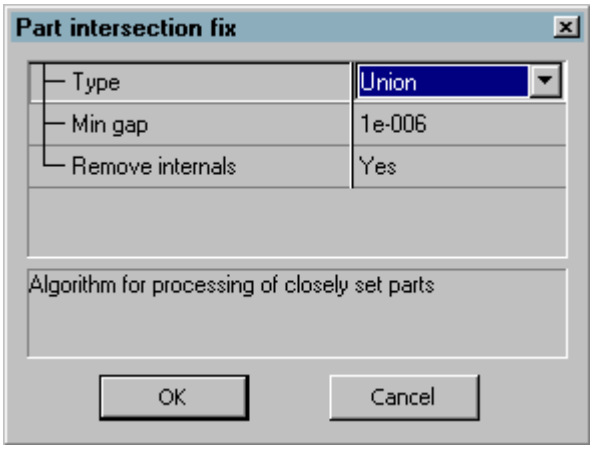
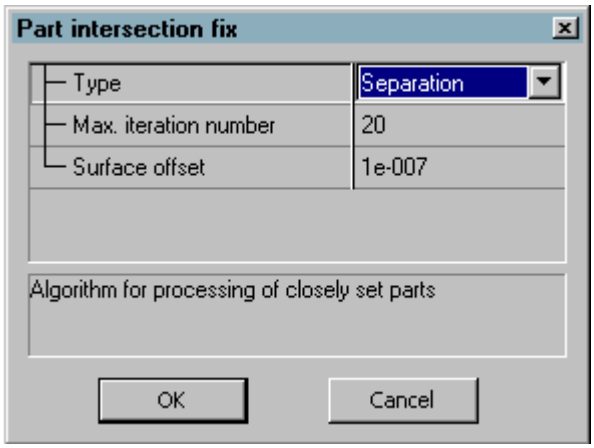


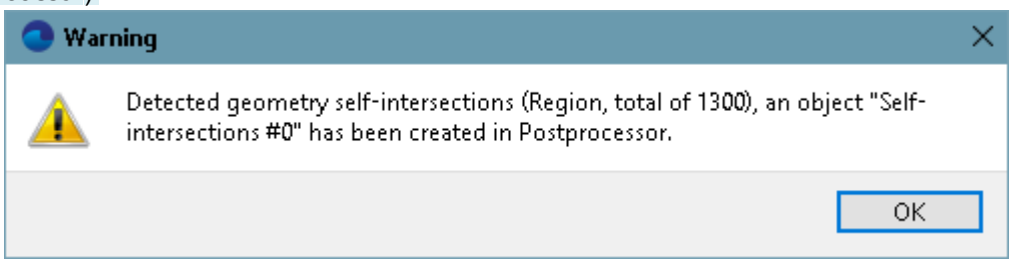
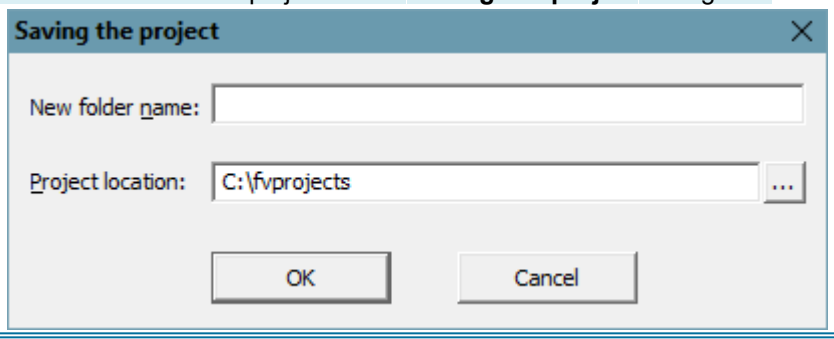
This window contains the following elements:

pane **Files in assembly**	This is the list of geometry files that will be included to the assembly. This pane contains columns: • **Name**: name of the file • **Path**: path to the file in the file system
button **Add**	Adding file(s) to the assembly. When you click this button, a standard operating system's dialog box for file selection will open where you can select one or more files that will be added to the assembly. Names of the selected files and paths to them will be added into the **Files in assembly** pane.
button **Remove**	Remove the records selected in the **Files in assembly** pane. This button is duplicated by the **Delete** key on the keyboard.
button **Move to Begin**	Move the selected records to the beginning of the list. This button is duplicated by the **Alt+Home** hot key.
button **Move Up**	Move the selected records one position up. This button is duplicated by the **Alt+↑** hot key.
button **Move Down**	Move the selected records one position down. This button is duplicated by the **Alt+↓** hot key.
button **Move to End**	Move the selected records to the end of the list. This button is duplicated by the **Alt+End** hot key.
Assembly total: ... file(s)	Number of records in the **Files in assembly** pane.
button **OK**	Loading the assembly
button **Cancel**	Canceling the creation of the assembly

An assembly always has to include an the outer surface of the computational domain and this surface has to be the first in the list of geometries. Otherwise, another geometry, which is specified as the first one, will be considered as the outer geometry and the result might be incorrect.

Loading geometry model of multiple files (assembly)	
Step	**Description**
	Create a list of the assembly's files, by clicking the **Add** button and then selecting the desired files in the operating system's dialog box, which opens. When necessary, place the file with outer surface of the computational domain on the first position in the list. When you form the list of files for the assembly, click **OK**.

2	The **Part intersection fix** window will open:
	 
	
	or
	 
	
	This window contains settings for handling the intersections:
	• **Type**- This parameter defines, which algorithm will be applied to the contacting parts. Possible options are:
	  ○ **Union** - make union of closely spaced surfaces with the eliminating the gap and forming a contact face
	  ○ **Separation** - splitting the contacting surfaces and forming a gap between them
	• **Min gap** is the minimal gap, which is between the parts. This parameter is only available when when **Type=Union**. In the locations, where the surfaces approach each other to the distances less then **Min gap**, shift of the surfaces and, depending on the **Remove internals** setting, removing of some fragments of the surfaces will be done (see details in sections Assembling with union and Algorithm of assembling with union). The **Min gap** parameter is the criterion, which determines overlap of two surfaces. If the distance between the surfaces is less then **Min gap**, they will be joined using shift (offset) even when they intersect each other, because this is necessary for making a correct Boolean sum of sets.
	• **Remove internals** - automatically remove internal volumes and excess surfaces. This parameter is only available when when **Type=Union**. If you select **Remove internals = No**, then in future you will have manually to remove the appearing internal volumes and excess surfaces.
	• **Max. iteration number** - the maximum number of iterations in the procedure, which searches intersections. This parameter is only available when when **Type=Separation**.
	• **Surface offset** - the maximum displacement of nodes of facets during per one iteration. If a positive value is specified, the nodes of facets are displaced inside the object; if the value is negative, the nodes are displaced out. This parameter is only available when when **Type=Separation**.

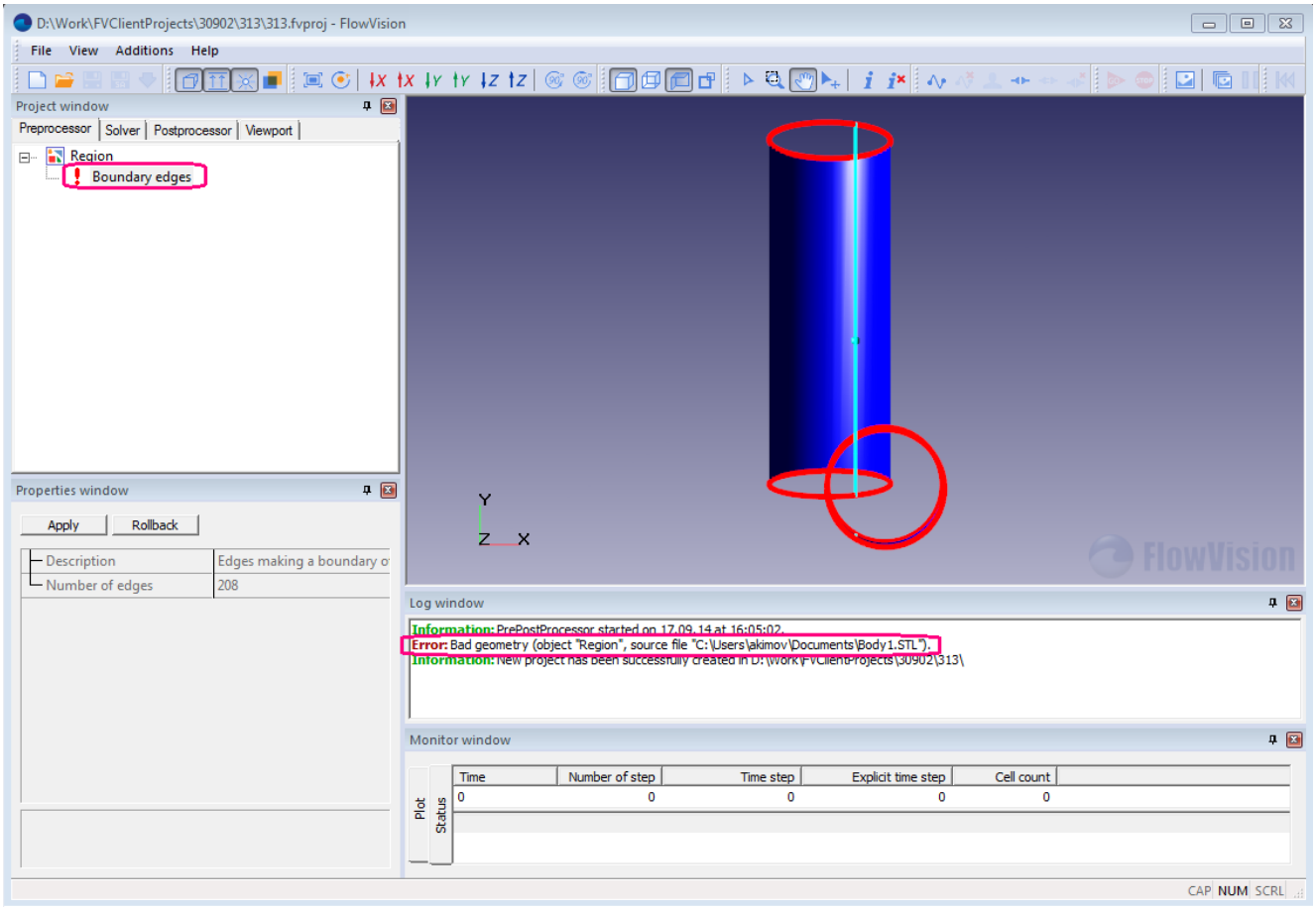
Loading geometry model of multiple files (assembly)	
Step	**Description**
	Select the method of eliminating the intersections and, if necessary, change the default settings. Then click **OK**.
3	*If* some self-intersections occur, **Pre-Postprocessor** will create an element representing them in the project tree and will display a warning about the self-intersections (**Detected geometry self-intersections (..., total of ...), and object "Self-intersections #N" has been created in Postprocessor**):  Click **OK**.
4	Specify the name and location of the project in the **Saving the project** dialog box: 
5	In the **Saving the project** dialog box specify: • project name in the **New folder name** field • if necessary, the project location in the **Project location** Then click **OK**. In the client part of the project will create a subdirectory of the project that contains the client files of the project. Windows Pre-Postprocessor filled data to match the loaded geometry model.

Displaying of errors of the loaded geometry model

If loaded geometry model of the computational domain is unsuccessful, in the **Pre-Postprocessor** displays diagnostics loaded model.

In the **Log** window a message will be displayed: **Error:Bad geometry**, and then, after saving the project, in the project tree, in the **Preprocessor** tab, a list of detected errors will be displayed.

When you select in the project tree of any type of errors in the **View** window with all the elements of such errors will be shown in red (see illustration).



Types of errors of the geometry model:

Type of error	Description
Boundary edges	In the geometry model contains the edges forming the boundary of an open surface
Overlap	In the geometry model contains edges belonging to overlapping facets
Hanging ribs	In the geometry model contains an edge not belonging to any facet
Multiconnection	In the geometry model contains edges belonging to more than two facets. Using multiconnection can allow or block basic settings of FlowVision. See also subsection "*Multiconnection*" in section geometry model of the computational domain: surfaces and subregions.

When errors are detected in a busy geometry model should be corrected by means of their CAD system and upload the corrected geometry model.

8.2.2.2 Replacing a geometric model of computational domain in a project

The *geometry replacement* occurs when the geometry of the selected element is replaced by another (new) geometry. Geometry replacement is available for objects **Region**, **Imported object** and **Moving body**.

In an existing project it is possible to replace a part of the geometry model by other surfaces (by other geometry model).

To replace the geometry model of the project, follow these steps:

Step	Actions
1	Open the context menu of the element (**Region**, **Imported object** or **Moving body**) and select the **Replace geometry** command.
2	In the standard window of the operating system select the file with the geometry model of the computational domain, and then click **Open**. The **View** window will display the loaded geometry model.

Please note:

Replacement of geometry of **Region** makes the whole project be calculated from the scratch (even if continuation of the computation has been selected).

When the geometry is replaced for an **Imported object**, on which a **Moving body** is defined, or just for a **Moving body**, the computation can be run for continuation, but, in this case, it is necessary to enable updating in the properties of the **Moving body** (it is set by the **Update > ...** parameters), or else the new geometry would not replace the old geometry on **Solver**. If **Update > Type > Disabled** is set in the properties of the **Moving body**, then the geometry will not be replaced on **Solver**.

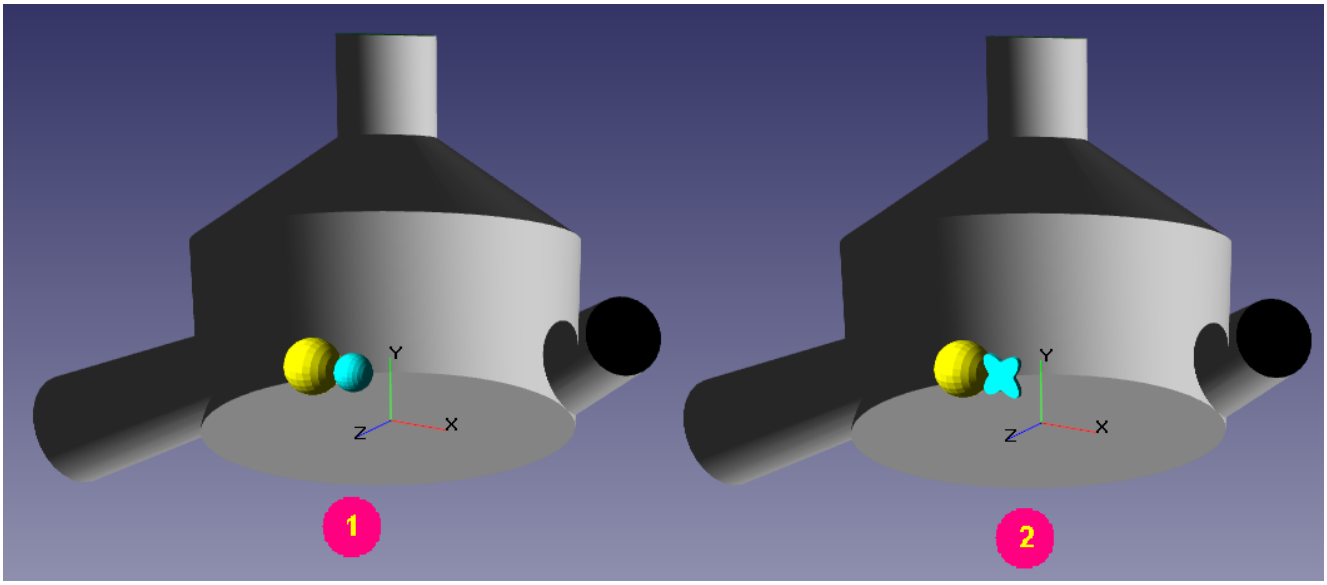
Therefore, immediately after changing the geometry of a **Moving body**, which update is disabled, you have to make at least a single update, that is, to make at least one iteration of the computation with enabled update of the **Moving body**. If the update of the **Moving body** has been disabled, enable it before the geometry replacement (by specifying, for example, **Update > Type = Auto**) and then, after connection to **Solver** and making the geometry replacement, do the computation until the computational grid is rebuilt. Then you can again disable the update of the just modified **Moving body** by setting **Update > Type > Disabled**.

When the computation is started from scratch, you do not have to enable updates of the **Moving body** after the geometry replacement.

When FSI computations are carried out with geometry replacement, updates of **Moving bodies** must be enabled.

Why sometimes it is desirable to disable updates of a Moving Body:

The computational grid is rebuilt at each update of a **Moving body**. This process takes a certain CPU time. Therefore, in simulations where the **Moving body** does not move, it is recommended to disable the updates. If the project has several **Moving bodies**, then update of one of them will cause update of all other **Moving bodies**.



Geometry of a **Moving body** has been replaced:

(1) the **Moving body** (blue ball) was originally created on an **Imported object** obtained from a standard geometric object **Ellipsoid/sphere**;

(2) - the geometry of the **Moving body** was replaced by a geometry from a file (the **Imported object**, on which the **Moving body** has been built, was also replaced by this operation)

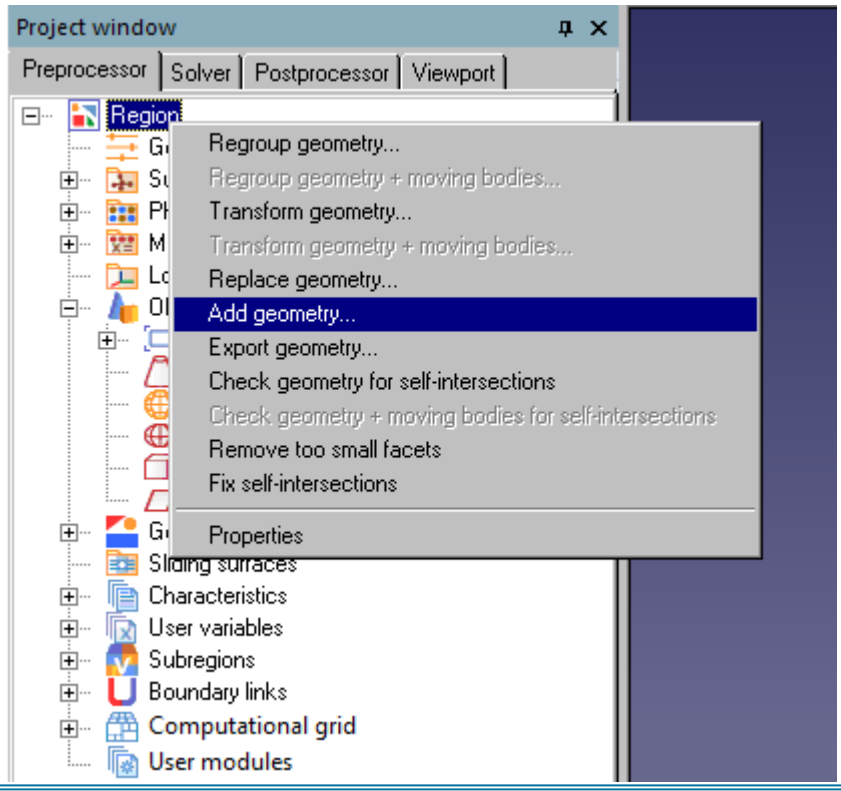
See also: section Geometry replacement.

8.2.2.3 Adding a surface into a geometric model of computational domain

Adding to the surface geometry model of the computational domain can be accomplished in two ways:

- the addition of the surface by importing*a file with a geometric surface model*
- inserting the object into a geometry model of the computational domain.

Adding surface by importing a file with a geometric surface model

Adding a surface in a geometry model of the computational domain	
Step	**Description**
1	Open the context menu of the root folder **Region** or context menu of a surfaces and select there the **Add geometry** command: 
2	The standard operating system window for file selection. Select a file, which is imported from the surface, and then click **Open**.
3	There will be the addition of the surface of the computational domain with automatic repartitioning the computational domain into subregions according to the existing algorithm.
4	When you add in the estimated surface area of the program will prompt you to tested for self-intersection. It is recommended to perform this test.

Embedding an Object into the geometry model of the computational domain

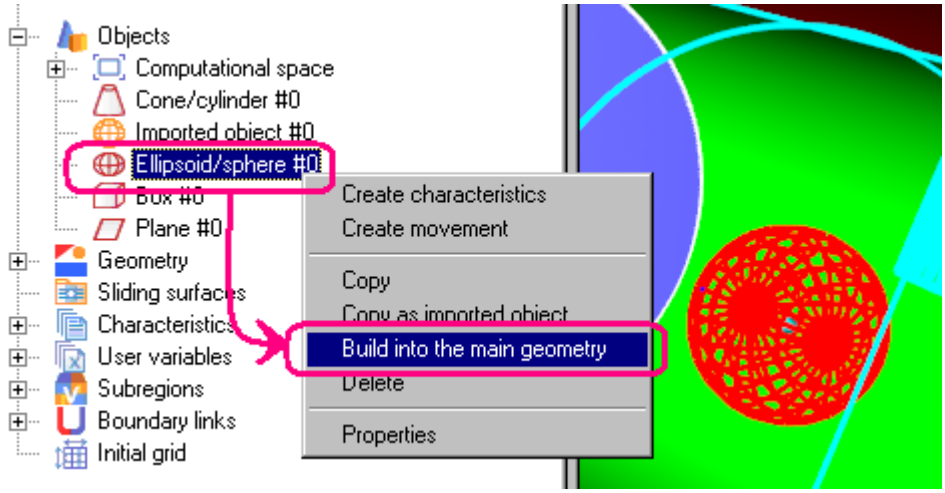
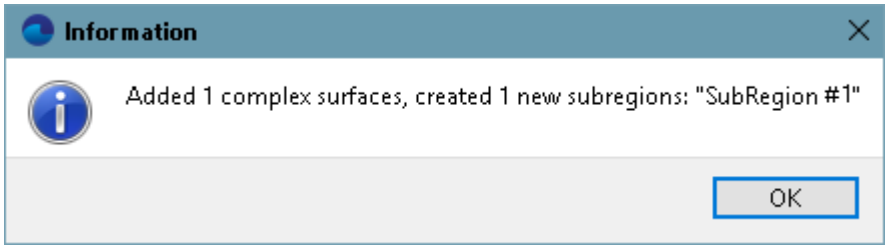
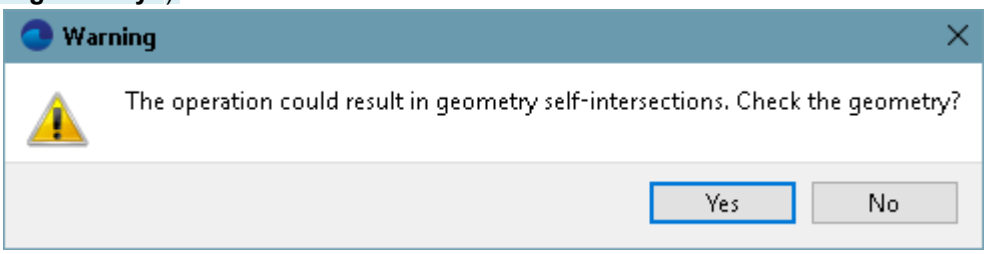
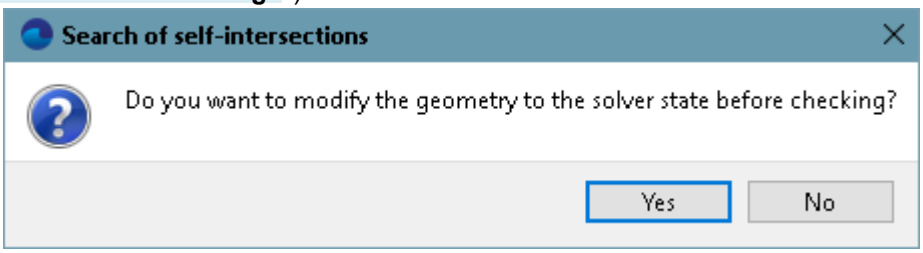
In the geometry model of the computational domain can be embedded **Objects** of the following type:

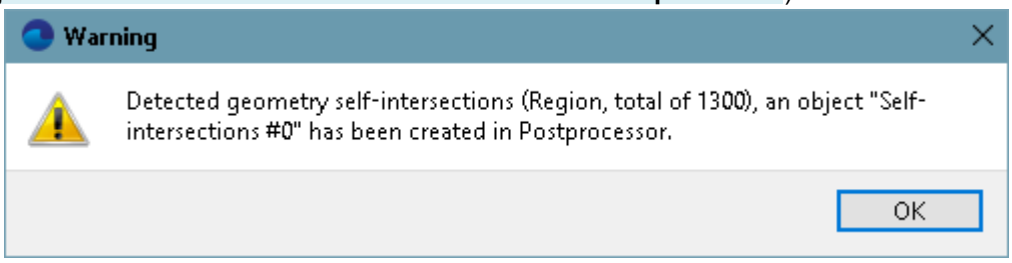
- to **Imported object**
- standard finite volume geometric objects: **Box**, **Cone/cylinder**, **Ellipsoid/sphere**.

Embedding **Object** in the computational domain as follows:

- creates a copy of an object's surface facets, which limits individual subregion of computational domain
- check whether the self-intersections of facets resulting geometry model of the computational domain
- detection of self-intersections, you can return to the original state by removing the added surface of the computational domain (see section Removal of the surface of a geometry model of the computational domain)

Object to embed in a geometry model of the computational domain, follow these steps:

Embed an object in a geometry model of the computational domain	
Step	**Description**
1	Right-click the row of the embedded **object** in the **Objects** folder and select the command **Built into the main geometry**:  Added subregion appears in the project tree as an element **Subregions > SubRegion #N >Geometry > Surface #M**, thus there will be the creation of a new surface and automatic repartition the computational domain. An informational dialog box (**"Added ... complex surfaces, created ... new subregions: "SubRegion #..."**) will open with a message about results of the embedding:  Click **OK**.
2	A dialog box will open with a warning (**The operation could result in geometry self-intersections. Check the geometry?**):  Click **Yes**.
3	Window opens to request (**"Search of self-intersections"** / **"Do you want to modify the geometry to the solver state before checking?**"):  When you embed the geometry produced a small correction of geometry, which avoids some of the problems with the initial dissection of the cells of the grid. Sense to specify the question is to choose: • check on the intersection geometry uncorrected

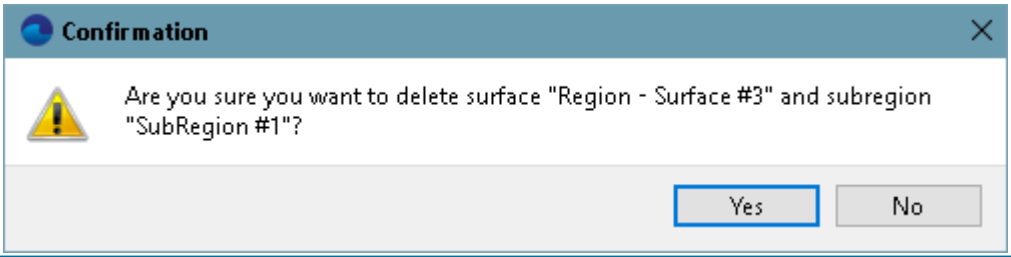
Embed an object in a geometry model of the computational domain	
	• *or*check on the intersection geometry after applying the correction We recommend you select as follows: • if the initial grid has already been built, click **Yes** • if the initial grid is not built yet, click **No**
4	The program will search for self-intersections. Upon detection of self-intersections of the program will perform the following steps: • folder **3D-scene > Objects > Computational space** created imported to **Self-intersections of #0**; this property consists of intersecting facets and created as an auxiliary to analyze the reasons that led to an error in setting the computational domain; after analysis of the object to be removed • in the Project tab displays **Postprocessor** • a warning dialog box is displayed (**Detected geometry self-intersections (..., total of ...), an object "Self-intersections #N" has been created in Postprocessor.**):  Click in the window **OK**.
5	To correct for the error should be removed from the surface of the added subregion of the computational domain. This is done by removing the added subregion from the project tree displayed on the tab **Preprocessor** (see section Deleting a surface from a geometry model of computational domain).

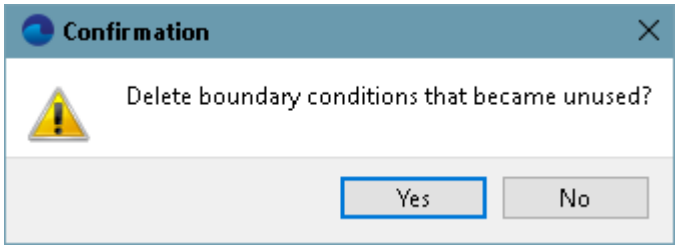
8.2.2.4 Deleting a surface from a geometric model of computational domain

When you delete from the geometry model of the computational domain surface that delimits two subregions, is removed as the surface and one of the subregions.

If the surface of the **Region- Surface #N** delineates two subregions **SubRegion #K** and **SubRegion #M**, then the surface is displayed in two folders of the project tree:

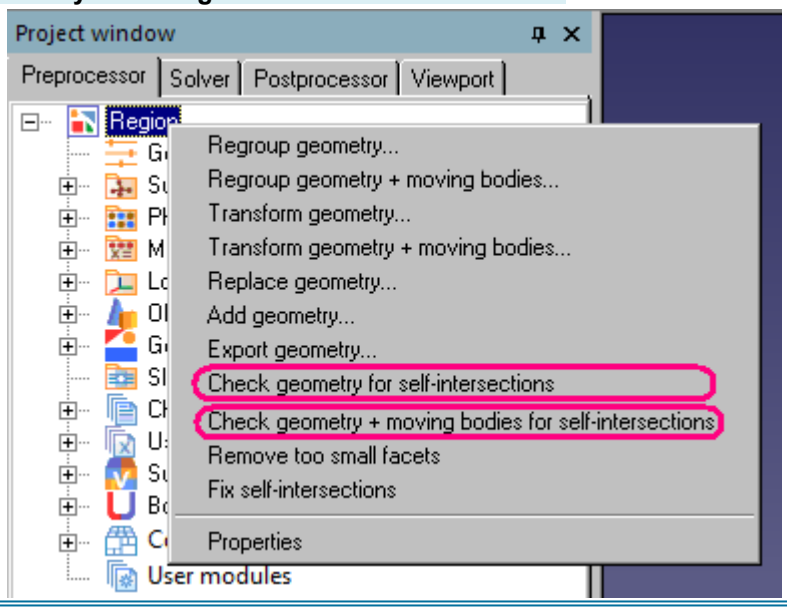
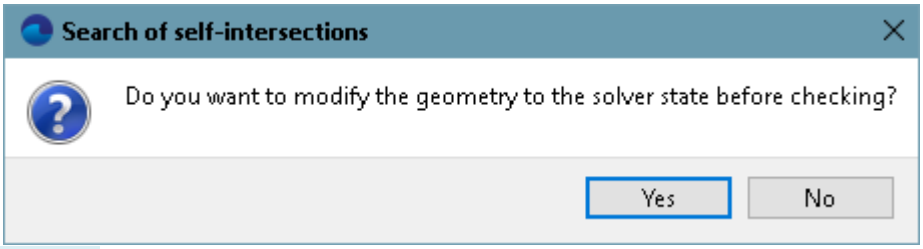
- folder **Subregions > SubRegion #K > Geometry > Region- Surface #N**
- folder **Subregions > SubRegion #M > Geometry > Region- Surface #N**

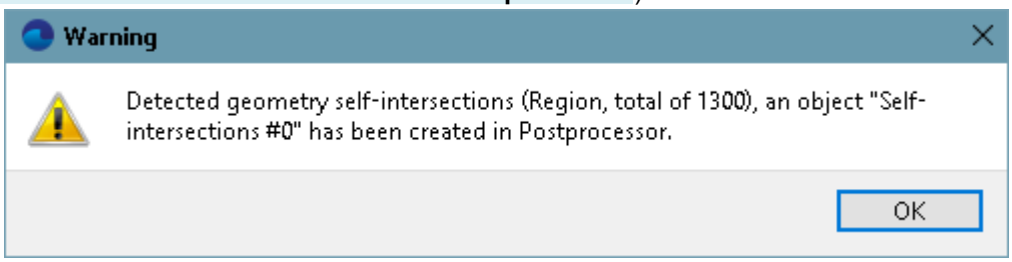
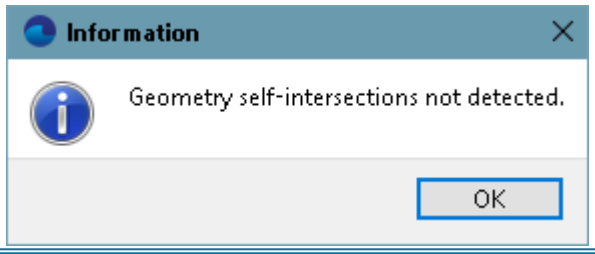
To remove the surface of a geometry model of the computational domain:	
Step	**Description**
1	Right-click the item **Subregions > SubRegion #N > Geometry > Surface #N** (surface assembly in the folder of the subregion, which must be removed together with the surface). Select in the context menu, click **Delete, surface and subregion**.
2	A **Confirmation** dialog box will open with information about what would be removed ("**Are you sure you want to delete surface ... and subregion ...?**"): 
3	A **Confirmation** dialog box will open ("**Delete boundary conditions that became unused?**"):

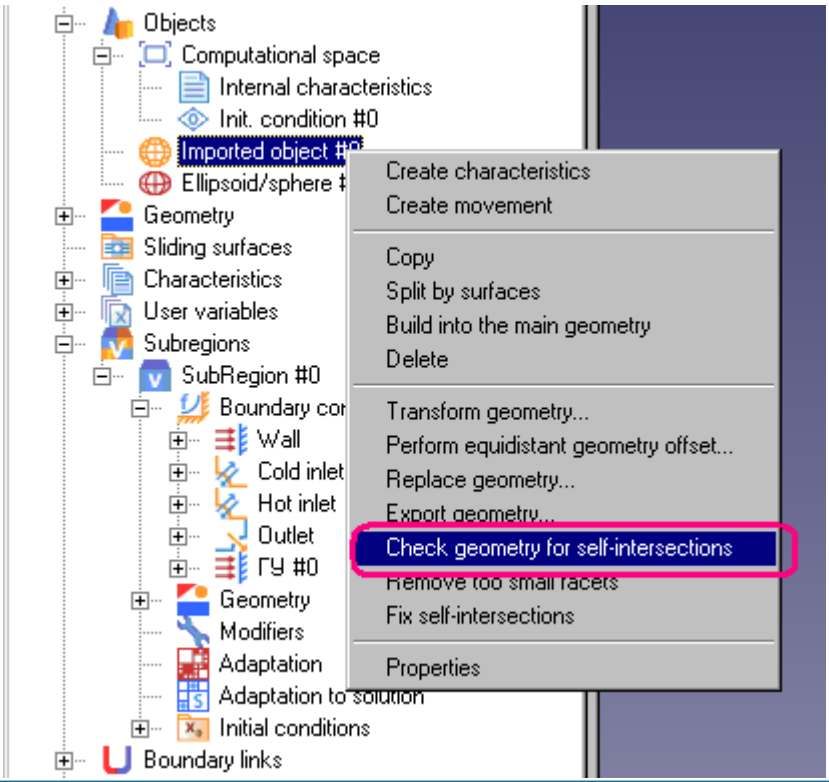
To remove the surface of a geometry model of the computational domain:	
Step	**Description**
	 Click **Yes** or **No**, depending on whether or not to save the unused boundary conditions. The recommended answer is **Yes**.

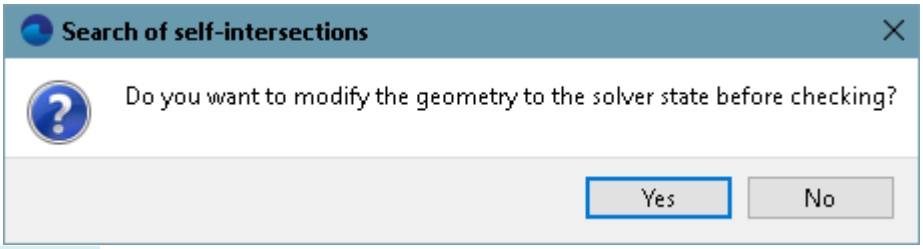
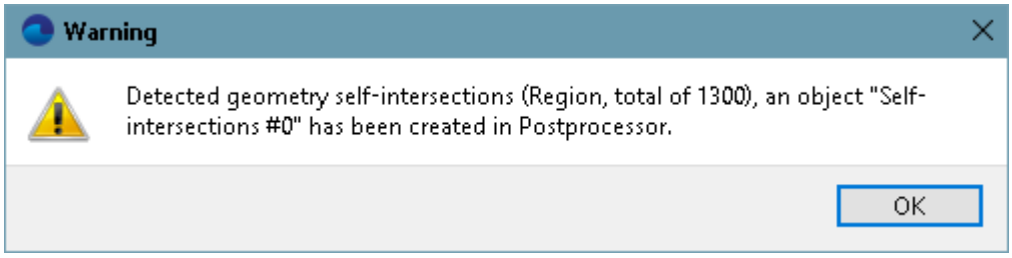
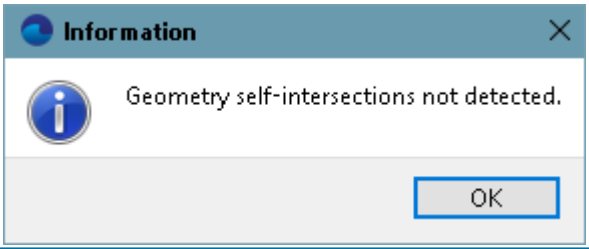
8.2.2.5 Checking a geometric model of computational domain and moving bodies for self-intersections

Checking the geometry of self-intersection searches for self-intersections throughout the geometry model of the computational domain, or a selected surface of a geometric object.

Check for self-intersection of the surface of a geometry model of the computational domain (and moving bodies)	
Step	**Description**
1	Open the context menu of the root folder **Region** and select the command: • **Check geometry for self-intersections** • *or* **Check geometry + moving bodies for self-intersections** 
2	A dialog box with a request ("**Search of self-intersections**" / "**Do you want to modify the geometry to the solver state before checking?**") will open:  • If the initial grid has already been built, click **Yes** • If the initial grid is not yet built, click **No**
3	The system searches for self-intersections. Upon detection of self-intersections of the system performs the following actions: • in the **Project** tab displays **Postprocessor**

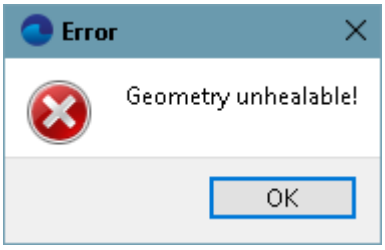
Check for self-intersection of the surface of a geometry model of the computational domain (and moving bodies)	
Step	**Description**
	• folder **3D-scene > Objects > Computational space**created imported to **Self-intersections #0**;this property consists of intersecting facets and created as an auxiliary to analyze the reasons that led to an error in setting the computational domain; after analysis of the object should be removed; • A warning dialog box is displayed (**Detected geometry self-intersections (...), an object "Self-intersections #..." has been created in Postprocessor.**):  In the absence of self-intersections is also a message box opens (**Geometry self-intersections not detected**) of this: 
4	Click **OK**.

Check for self-intersection of the surface of the selected geometric object	
Step	**Description**
1	Open the context menu of the selected geometrical object and select **Check geometry for self-intersection**. 
2	A window for the query (**"Search of self-intersections" / "Do you want to modify the geometry to the solver state before checking?"**):

Check for self-intersection of the surface of the selected geometric object	
Step	**Description**
	 • If the initial grid has already been built, click **Yes** • If the initial grid is not yet built, click the **No** button.
3	The system searches for self-intersections. Upon detection of self-intersections of the system performs the following actions: • in the **Project** tab displays **Postprocessor** • folder **3D-scene > Objects > Computational space** created imported to **Self-intersections #0** This property consists of intersecting facets and created as an auxiliary to analyze the reasons that led to an error in setting the computational domain; after analysis of the object should be removed; • A warning dialog box is displayed (**Detected geometry self-intersections (...), an object "Self-intersections #..." has been created in Postprocessor.**):  In the absence of self-intersections is also a message box opens (**Geometry self-intersections not detected**) of this: 
4	Click **OK**.

8.2.2.6 Fixing self-intersections of a surface in a geometric model

In order to eliminate the self-intersection surface of the object:	
Step	**Description**
1	Open the context menu of the root folder **Region** and select **Fix self-intersections**. Displays information about the results of correcting self-intersections: • if the operation is to eliminate the self-intersection is unsuccessful, the message **«Geometry unhealable»**. In this case it is necessary to correct the geometry model of the surface by means of CAD-systems. It is recommended to change the export geometry model in the CAD-system, or fix bugs grid in a special program (eg, *3DTransVidia*);

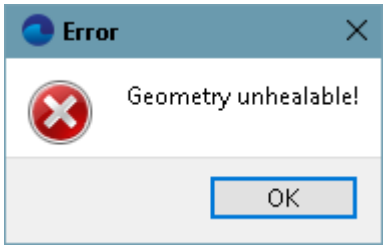
In order to eliminate the self-intersection surface of the object:	
Step	**Description**
	 • if the operation of removing the self-intersections is successful, then a dialog box with information about results of the correction will open (**Self-intersections have been fixed, ...% of surface area has been deleted. Save changes and continue with new geometry?**).
2	If area of the deleted surface is 0.00%, this means absence of self-intersections. Click **No**. If area of the deleted surface is sufficiently small, then click **Yes**. If area of the deleted surface is large, click the **No** button and correct the geometry model of the surface by means of CAD-systems.

The functionality of automatic fix of self-intersections does not guarantee positive results, because its algorithm is not applicable for all situations.

Fixing self-intersections of elements **Self-intersections #N** is done in the similar way:

- In the context menu, select **Fix self-intersections**.
- In the dialog box, which opens, if you have not built the computational grid yet, select **No**. Otherwise, select **Yes**.

If automatic correction of self-intersections failed, a message is displayed:

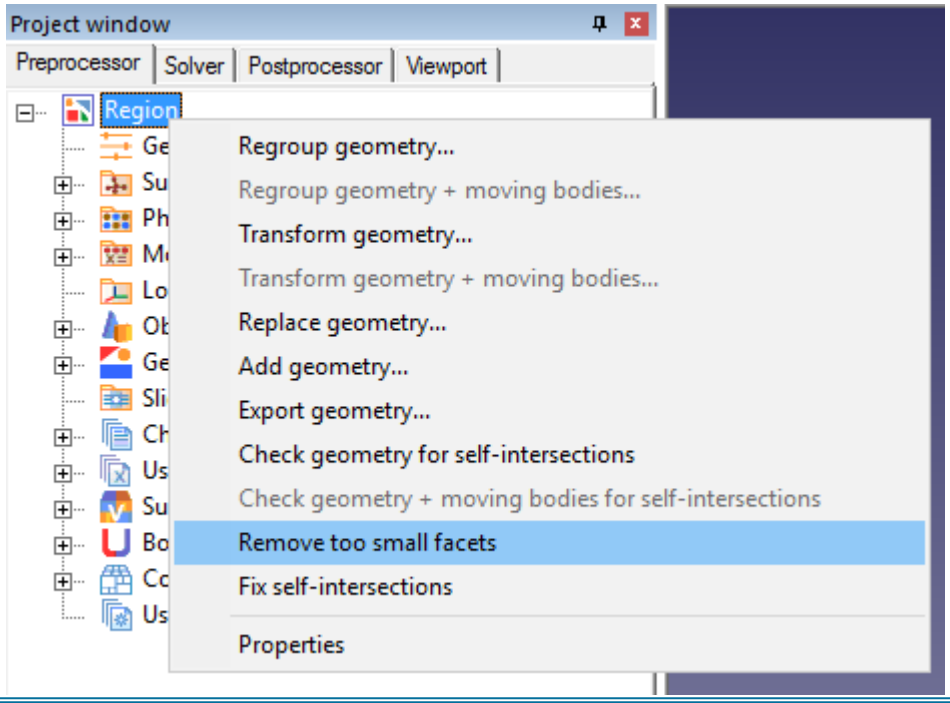
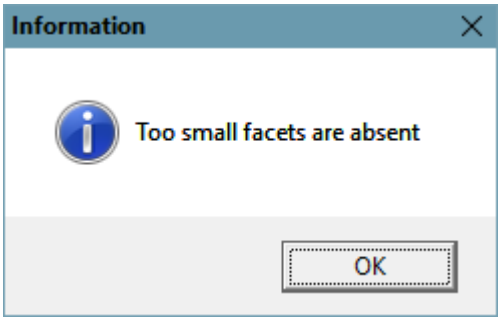


In this case it is recommended to change the export geometry model in the CAD-system, or fix bugs grid in a special program (eg, *3DTransVidia*).

8.2.2.7 Removal of too-small facets of geometric model of computational domain

Removing too small facet surface of the object:

Step	Actions
1	Open the context menu of the element (**Region, Imported object #N**) and then click **Remove too small facets**.

Step	Actions
	
2	A window appears with information about the results of the correction.  (Message "**Too small facets are absent**" is displayed in the absence of very small facets)
3	If small facets removed to increase the value of **the smallest edge** element **Region** (displayed in the **Properties** window).

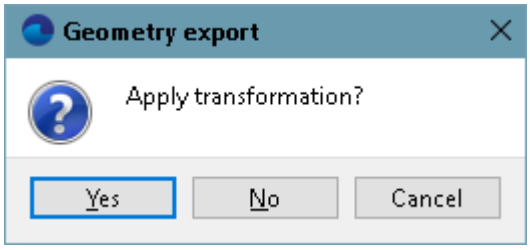
See also: section Removal of too-small facets.

8.2.2.8 Exporting a geometric model into a file (step-by-step procedure)

Export of the geometry model of the whole computational domain or some **Imported object** is performed to a file in the format of *WRML* (`*.wrl`), *VTK* (`*.vtk`), or *3DVision* (`*.mesh`).

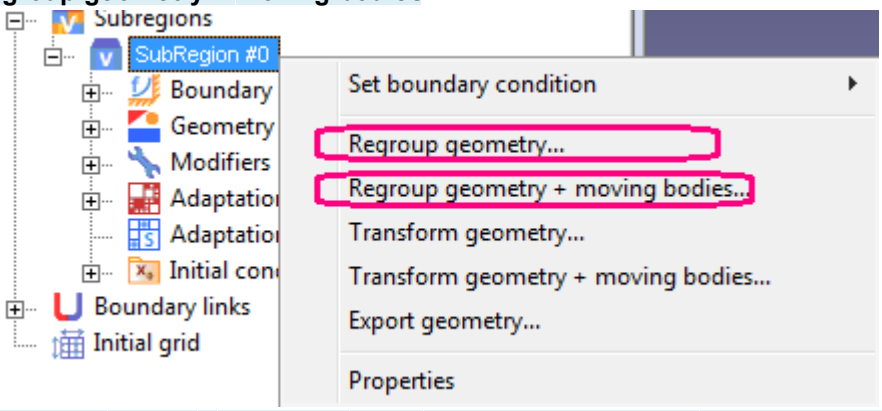
To export the geometry, follow these steps:

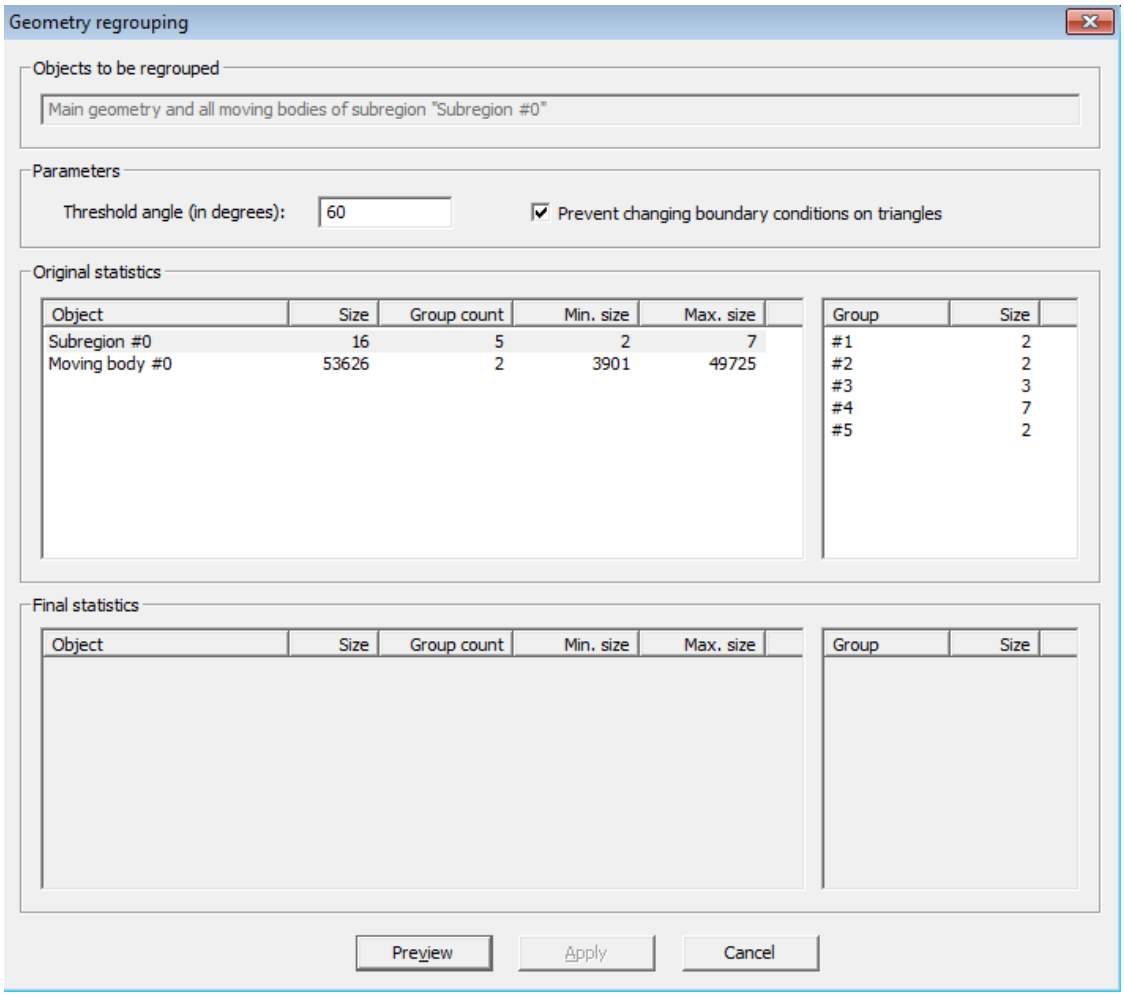
Step	Actions
1	In the context menu of the **Object**, the geometry of which is saved in a file (for example, the context menu of the root folder **Region**), select **Export geometry**.
2	When geometry of an **Imported object** is exported, a **Geometry export** dialog box will open where the program requests you whether is it necessary to save the geometry of the **Imported object** with transformations, which were applied to it ("**Apply transformation?**"):

Step	Actions
	 When you answer **Yes**, then geometry of the **Imported object** will be saved in a file along with transformations, which were applied to it. When you answer **No**, the geometry will be saved without the transformations. Clicking **Cancel** cancels export of the geometry model.
3	In the **Geometry export** dialog box, which opens, select the type and name of the file, into which the geometry will be saved.
4	Click **Save**. The data will be saved in the file.

8.2.2.9 Procedure of regrouping a geometric model of computational domain (and moving bodies)

Regrouping the geometry allows you to change splitting the geometric surfaces into **Groups** of facets.

To regroup the geometry model of the computational domain (and moving bodies):	
Step	**Description**
1	The dialog box for geometry regrouping can be opened from the context menus of the following objects in the project tree (in the **Preprocessor** tab): 1. **Region** - regrouping all geometry 2. **Subregion** - regrouping within the selected subregionsfrom one surface subregion 3. **Group** - regrouping within the selected group; 4. **Boundary conditions** - regrouping within the selected boundary condition; 5. **Moving body** - regrouping within the selected moving body. On all elements except **Moving body** and **Group**, it is possible to carry out the regrouping, including or not including **Moving bodies**. Open the context menu item of the project treeand select a command from it: • **Regroup geometry** • *or* **Regroup geometry + moving bodies**[*)]  [*)] Regrouping of **Region**, **Subregion**, and **Boundary conditions** can be carried out, either including or not including the surfaces of moving bodies.
2	The **Geometry regrouping** window will open:

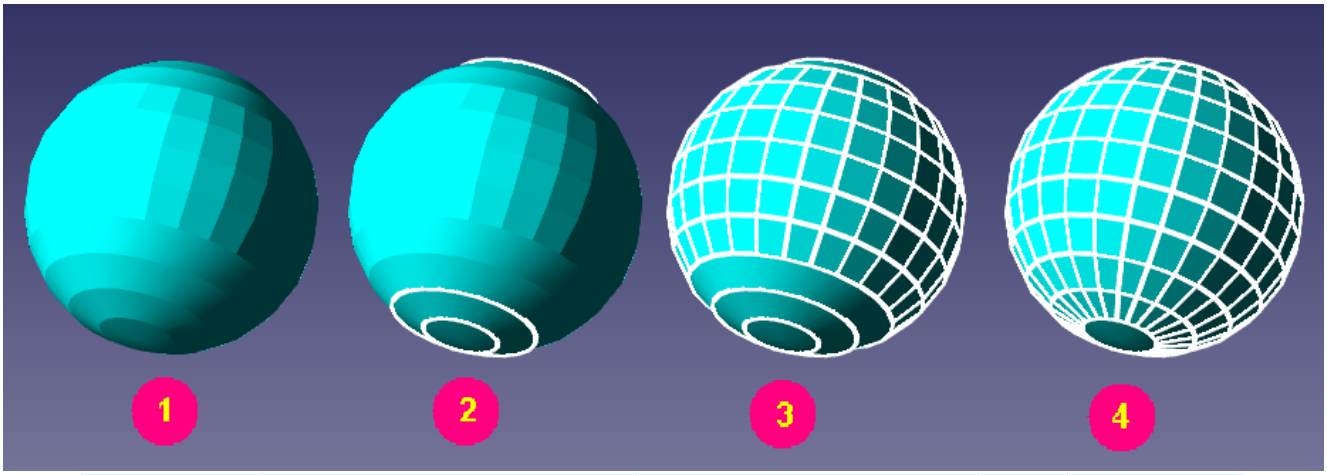
To regroup the geometry model of the computational domain (and moving bodies):	
Step	**Description**
	

The **Geometry regrouping** window contains the following elements:

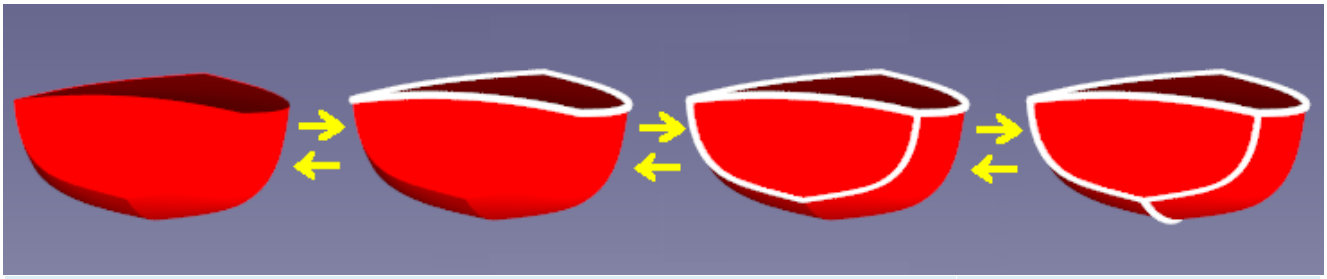
- The field **Objects to be regrouped**, which displays the name of regrouped object (reference information can not be changed by the user)
- The **Parameters** group of settings, which includes:
  - the **Threshold angle (in degrees)** field, which gives a critical spatial angle between the facets. If the angle between the facets is smaller than this angle, such veneers are considered belonging to the same group. With a decrease in this parameter, the number of groups into which the surface is growing, with an increase in falls.
  - the **Prevent changing boundary conditions on triangles** checkbox: do not merge into one group the facets, which have different boundary conditions. If this option is enabled, the facet with different boundary conditions are not included in the same group. If disabled, then the partition is carried out in groups regardless of the boundary conditions; after regrouping all facets of the group is assigned to one of the boundary conditions of those that were to regroup.
- **Original statistics** and **Final statistics** panels, which display statistics on the geometry before and after the regrouping:
  - in the left table:
    - ✓ **Object**- name of the regrouped surface
    - ✓ **Size**- the number of facets on the surface
    - ✓ **Group count**- the number of groups on the surface
    - ✓ **Min. size**- the minimum number of facets in the group
    - ✓ **Max. size**- the maximum number of facets in the group
  - in the right table:
    - ✓ **Group**- the group number
    - ✓ **Size**- the number of facets in the group
- The **Preview** button refreshes the **Final statistics** panel, i.e. look at the statistics on the geometry, which is obtained after regrouping

To regroup the geometry model of the computational domain (and moving bodies):	
Step	**Description**
	• The **Apply** button, click Run to regroup with the given parameters (the button becomes active after clicking on the **Preview** button) • The **Cancel** button- close the window and do not carry the regrouping. The same thing happens if you close the window marked **"x"** in the upper right corner. The main criterion for the regrouping of geometry - is *the deflection angle* - the angle between the normals of adjacent triangles belonging to different groups. For example, in a case where the loaded geometry consists of a cube, the deviation angle of 90°, means that one group corresponds to one of the triangles of the cube. Thus, the deflection angle is smaller the smaller the grouping, and conversely, the more this angle, the larger partition. Specify the **Threshold angle (in degrees)** and uncheck/check the **Prevent changing boundary conditions on triangles** checkbox.
3	Click the **Preview** button. The evaluated results of the grouping will be displayed in the tables of the **Final statistics** panel.
4	After reaching the desired result lock it by clicking the **Apply** button.

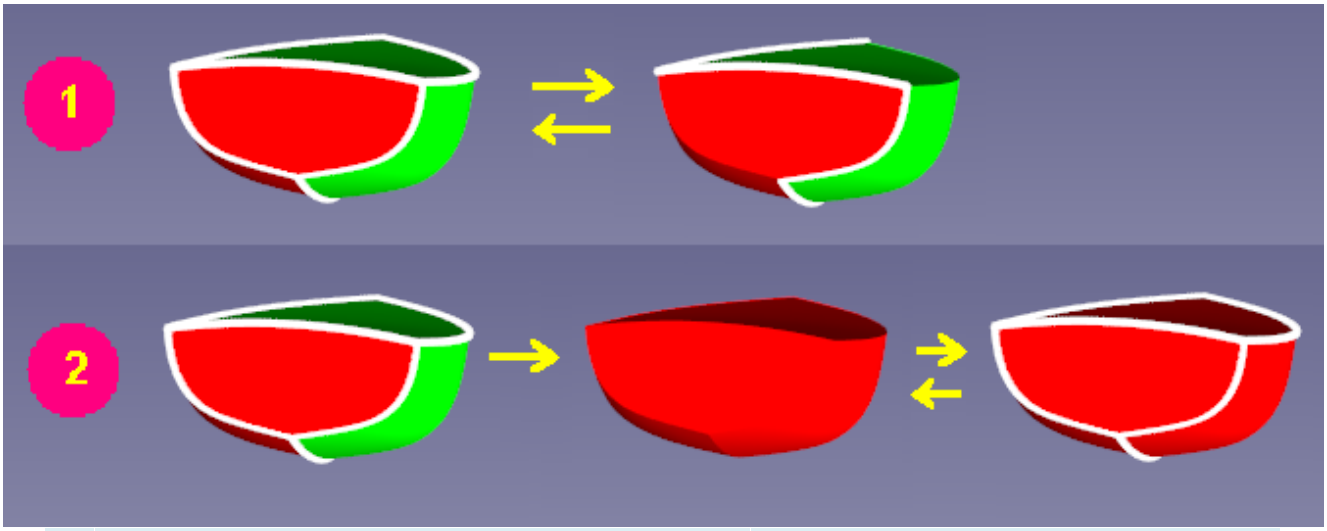
Illustrations



Influence of the **Threshold angle (in degrees)** parameter on how facets of a **Sphere** (with geodesic approximation) are grouped: (1) **30** degrees; (2) **15** degrees; (3) **= 10** degrees; (4) **= 5** degrees



Grouping the facets of the geometry model of a boat depending on the value of the **Threshold angle (in degrees)** parameter



The **Prevent changing boundary conditions on triangles** checkbox allows you to tune merging into one **Group** the facets, on which different boundary conditions are set.
(1) **Prevent changing boundary conditions on triangles** is checked
(2) **Prevent changing boundary conditions on triangles** is unchecked

---

See also:

- Grouping the facets
- Groups of facets as geometric objects
- Regrouping the facets
- Folder «SubRegion #N > Geometry»

8.2.2.10 Transformation of geometric model of the computational domain (and moving bodies)

Transformation geometry- shift, rotation, scaling geometry (see section The transformation of a geometry model of the computational domain and the imported object).

Transformation geometry is made using the context menu:

- **Transform geometry** or **Transform**
- *or* **Transform geometry + moving bodies** (this command may not be available in the context menu) - in this case,the transformation is also applied to the **Moving bodies** available in the **Object**

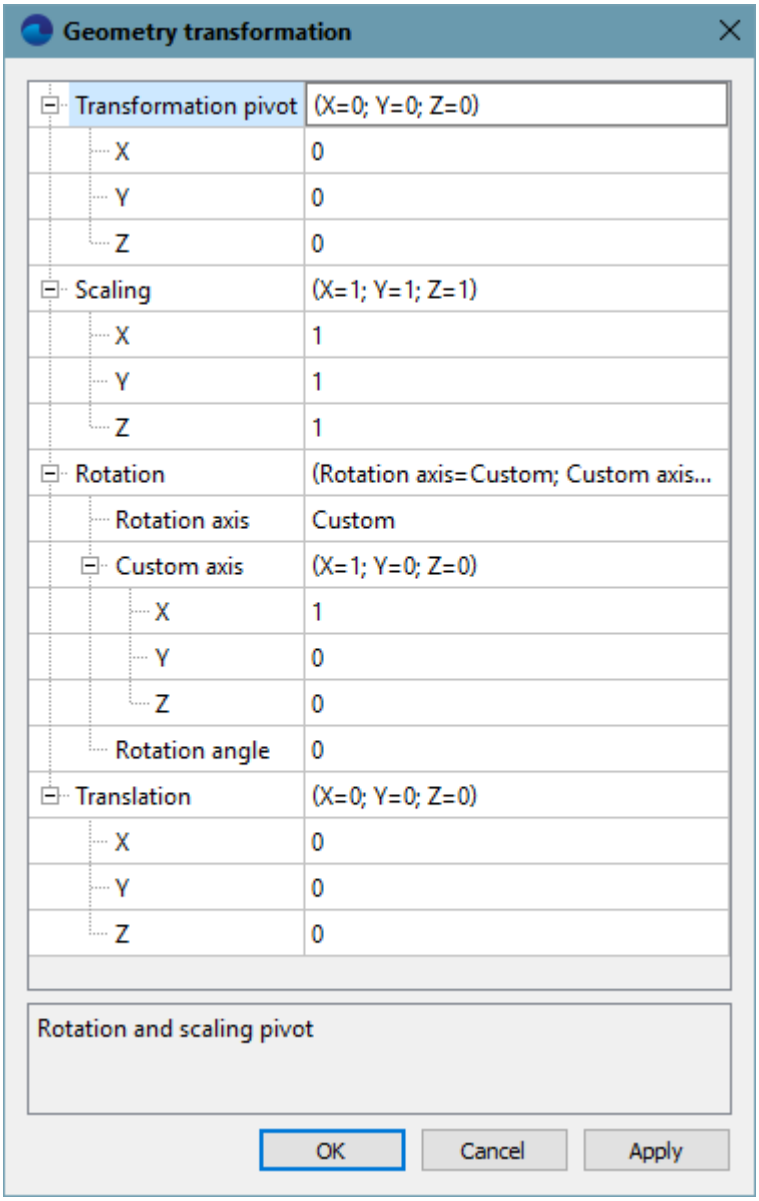
Converting elements **Region**, **SubRegion #N** and **Region- Surface #N** performed in absolute coordinate system.

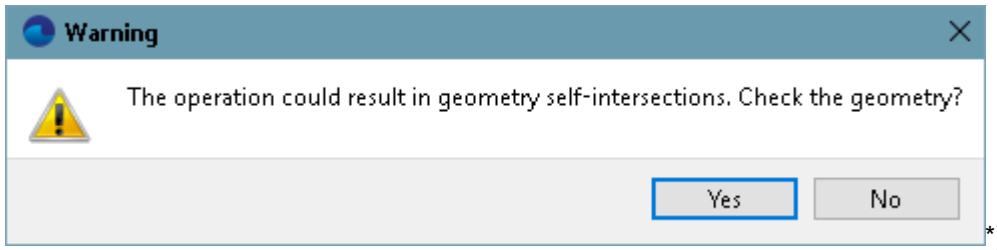
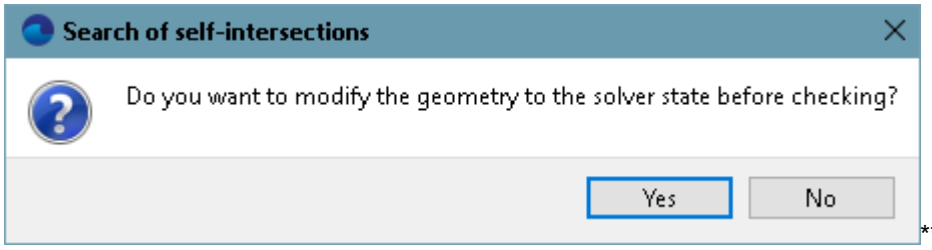
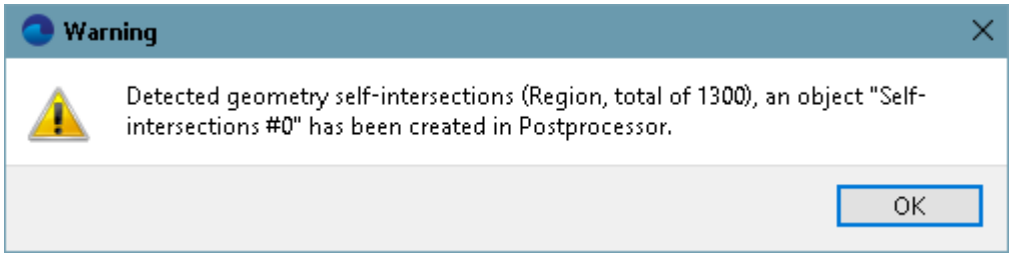
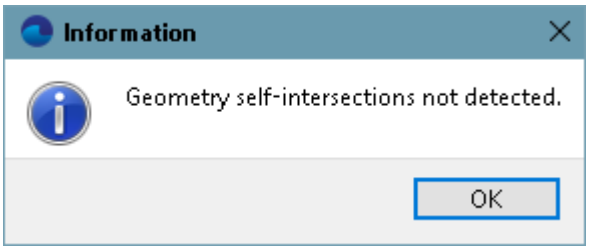
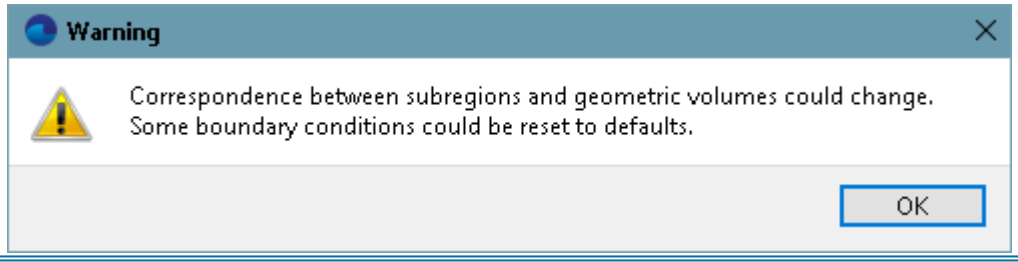
Converting elements **Imported object #N** and **Moving body #N** performed in the local coordinate system.

Transformation of geometry, called for an element **Region** (geometry model of the computational domain) is carried out in the coordinate system and allows absolute change the relative position of the geometry model of the computational domain and absolute coordinate system.

How to open the "Geometry transformation" dialog box, specify parameters of the transformation and start the transformation

The transformation of a geometry model of the computational domain (and moving bodies)	
Step	**Description**
1	Open the context menu root folder **Region** and select it: • **Transform geometry** • *or* **Transform geometry + moving body** Once clicked, a dialog box **Geometry transformation** opens:

The transformation of a geometry model of the computational domain (and moving bodies)	
Step	**Description**
	 Transformation geometry is made by pressing the **OK** button in this window (the **Geometry transformation** dialog box closes). Clicking the **Cancel** button or icon (**"x"** in the upper right corner to close the Geometry transformation dialog box without performing transformation geometry. Click the **Apply** button to see the result of transformation to the actual implementation of the transformation (i.e., the **Apply** button performs the preview). Individual actions to be taken in the course of the transformation, made in a manner consistent with the location of items in the **Transformation of** geometry: 1. Determined by the operations center 2. scaling is done 3. rotation is done 4. The shift is done Details and examples are described separately below.
2	Specify conversion options and click **Apply** to preview the transformation in the **View** window. If necessary, repeat this step by including other data, until you get satisfied with the results.
3	Click **OK** to actually carry out the transformation geometry in accordance with the entered data. The **Geometry transformation** dialog box closes at the same time. After transformation can be displayed warning and informational messages (see sections Self-intersections of surfaces and their correction, Adding or removing Subregions in geometry models).

The transformation of a geometry model of the computational domain (and moving bodies)	
Step	**Description**
	 *)  **)   
4	If necessary, do *several rotations* or other transformations of the geometry (scaling, translation), repeat steps 1-3 several times (you have to open the **Geometry transformation** dialog box several times and close it with clicking **OK**).

*) It is recommended to answer **Yes**.

) When you embed the geometry produced a small correction of geometry, which avoids some of the problems with the initial dissection of the cells of the grid. This window allows you to choose whether to be tested for self-intersection geometry before or after correction. We recommend you select as follows: if the initial grid has already been built, click **Yes; if the initial grid is not yet built, click **No**).

Please note:

If the properties of the **Moving body** established **Updates > Type = Disabled**, the transformation geometry on **Solver** will not happen.

Therefore, immediately after the transformation geometry **Moving body** (with disabled updates) you need to organize at least a single update, that is to produce at least one iteration of the calculation included updating **Moving body**.

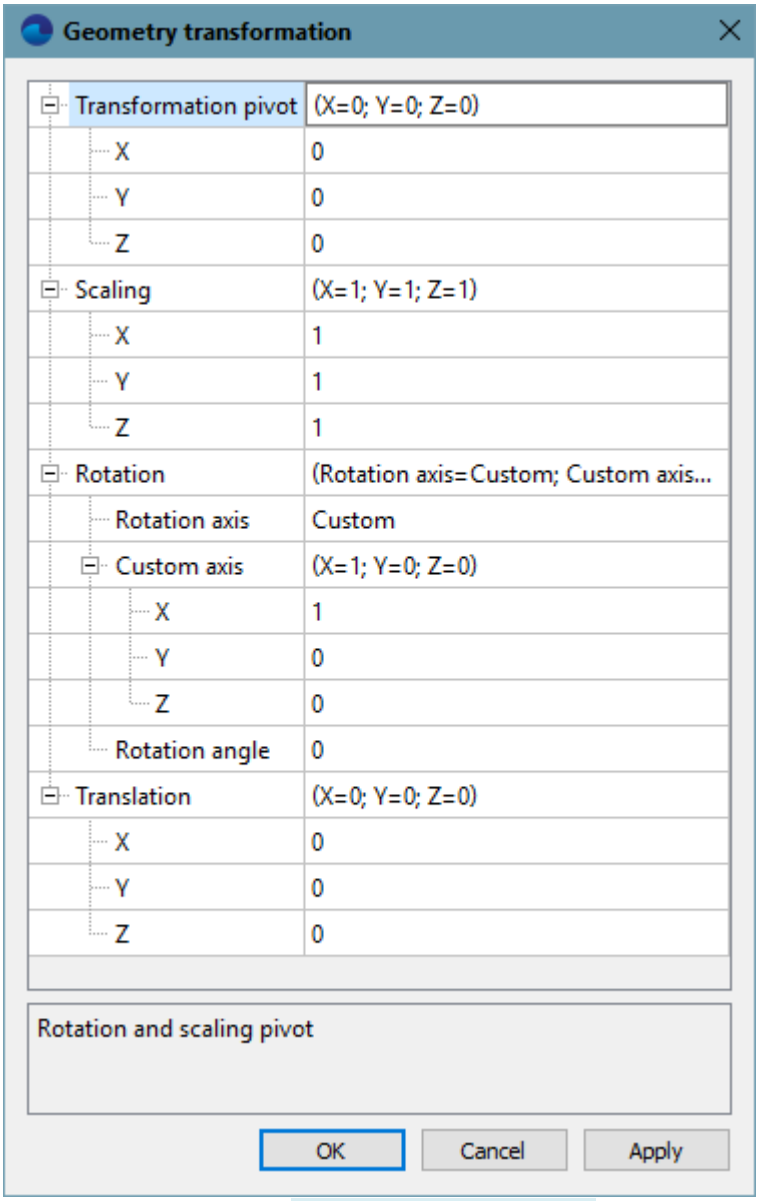
If the update **Moving body** was turned off, turn it on (by specifying, for example, **Update > Type = Auto**) and, connecting to **Solver**, do the calculation, until a rebuild of the computational grid. Then update the modified **Moving body** can be disabled again by setting **Update > Type = Disabled**.

Why update Moving body can be turned off:

With each refresh **Moving body** computational grid is rebuilt. This process takes a certain CPU time. Therefore, problems in which the **Moving body** is not moving, it is recommended to disable the update. If the project several **Moving bodies**, update at least one of them will lead to the renewal of all the rest.

Transformation parameters, defined in the "Geometry transformation" dialog box

The transformation parameters are specified in the **Geometry transformation** dialog box itself is performed after clicking in the window with **OK**.



Dialog box "Geometry transformation"

Transformation geometry will be made by clicking on **OK** (after this the Geometry transformation closes).

Clicking the **Cancel** button or **"x"** icon in the upper right corner closes the **Geometry transformation** window without performing transformation of geometry.

Click the button **Apply** to see the result of transformation to the actual implementation of the transformation (i.e., the **Apply** button performs the preview).

Transformation parameters, defined in the dialog box "Converting geometry"	
Parameter	**Description**
Transformation pivot > X	Coordinates of the center of operations of transformation geometry.
Transformation pivot > Y	
Transformation pivot > Z	
Scaling > X	The scaling factors along the axes X, Y, Z relative to the operations center.*)

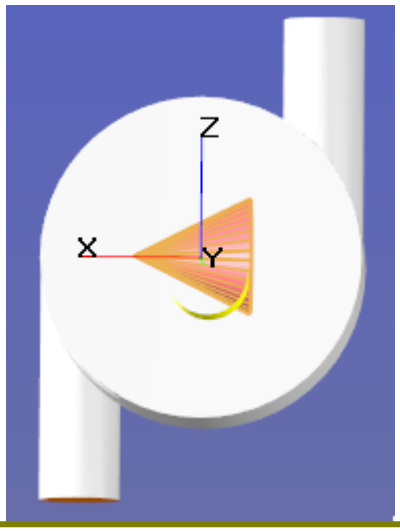
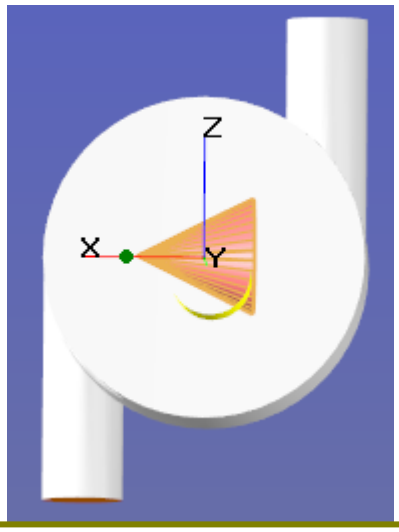
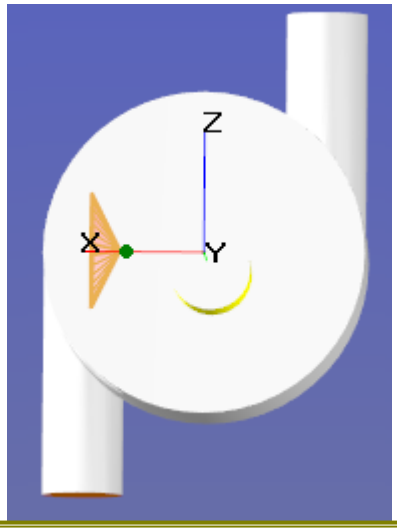
Transformation parameters, defined in the dialog box "Converting geometry"	
Parameter	**Description**
Scaling > Y **Scaling > Z**	 Scaling can also be used to mirror symmetry. For example, you can perform the following symmetric transformation: • If scaling factors on two axes are equal to **1**, and on the third is equal to **-1**, it will be made a mirror image relative to the plane in which lie the first two axes • If scaling factors on two axes are equal to **-1**, and on the third axis is equal to **1**, then the transformation will be mirrored axial symmetry about a third axis • if we take the scaling factors in all axes**-1**, the transformation will be a central symmetry with respect to the center of operations (see illustrations)
Rotation > Rotation axis[*)]	The direction of the rotation axis. Possible options are: • **X** - axis of rotation parallel to the X axis and passes through the center of operations. • **Y** - axis of rotation parallel to the Y axis and passes through the center of operations. • **Z** - axis of rotation parallel to the Z axis and passes through the center operations. • **Custom** - the direction vector of the axis of rotation is set in fields **Rotation > Custom axis > ...**,the axis of rotation passes through the center of operations
Rotation > Custom axis > X **Rotation > Custom axis > Y** **Rotation > Custom axis > Z**	The projection of the direction vector of the **Custom axis** on the axis X, Y, Z.[*)] Clicking the **Apply** button is held automatic normalization of the direction vector axis. These parameters are available when **Rotation > Rotation axis = Custom axis**.
Rotation > Rotation angle	The angle of rotation around the said axis (in degrees). Rotate performed after scaling.
Translation > X **Translation > Y** **Translation > Z**	The final shift after all other transformations. Specified components of the shift vector of the object along the axes X, Y, Z.[*)] shift is performed after scaling and rotation

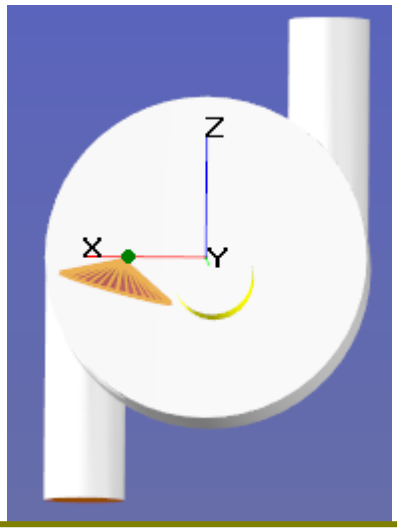
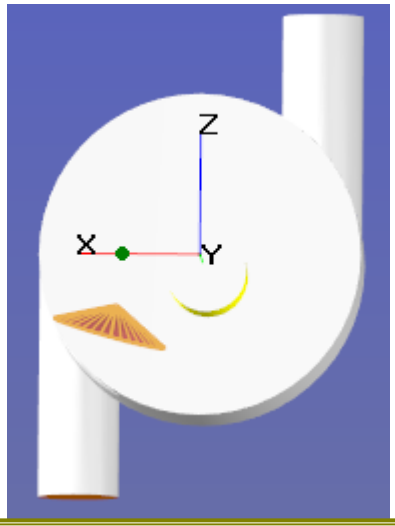
[*)] Conversion elements **Region**, **SubRegion #N** and **Region - Surface #N** performed in absolute coordinate system. Converting elements **Imported object #N** and **Moving body #N** performed in the local coordinate system.

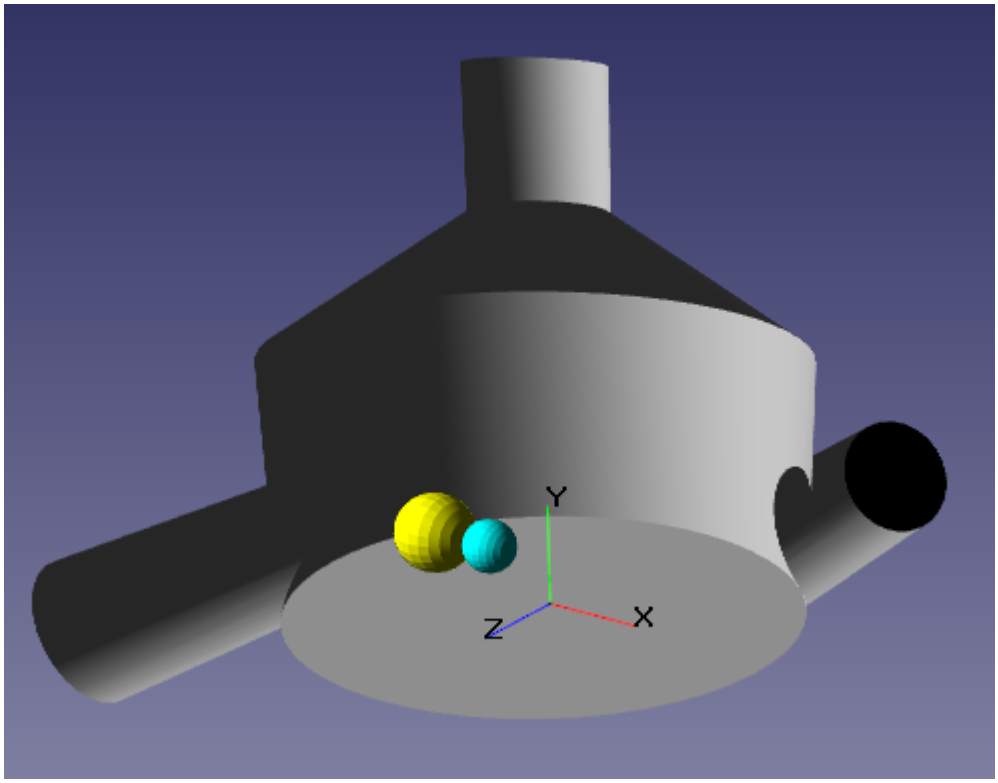
Individual actions to be taken in the course of the transformation, in the following order (corresponding to the location of items in the **Geometry transformation** dialog box)

1. Determined by the operations center
2. Is scaled
3. Produced rotation
4. The shift is

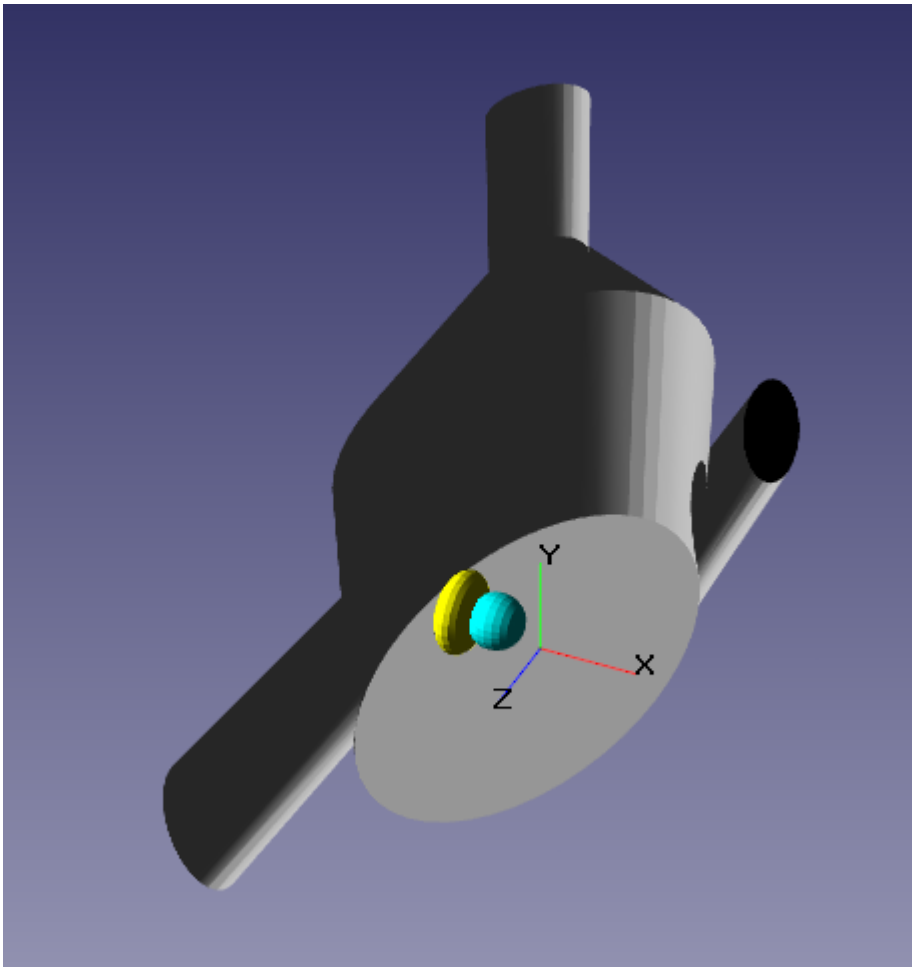
Examples and illustrations

Example of the transformation geometry		
Ref	**Action**	**Illustration**
0	The illustration shows the original shape and the location of the imported object (orange cone), the view against the axis Y.	
1	Define the center of operations in accordance with the values of the parameters in the **Transformation pivot** is shown in the illustration a green dot. In our example, set the following parameters: **Transformation pivot > X = 0.01** **Transformation pivot > Y = 0** **Transformation pivot > Z = 0**	
2	Is scaled along the coordinate axes, the center of operations performed. In this example, the compression is performed along the X axis and the mirror image relative to a plane perpendicular to the axis X and passing through the center of operations, defined parameters: **Scaling > X =-0.25** **Scaling > Y = 1** **Scaling > Z = 1**	

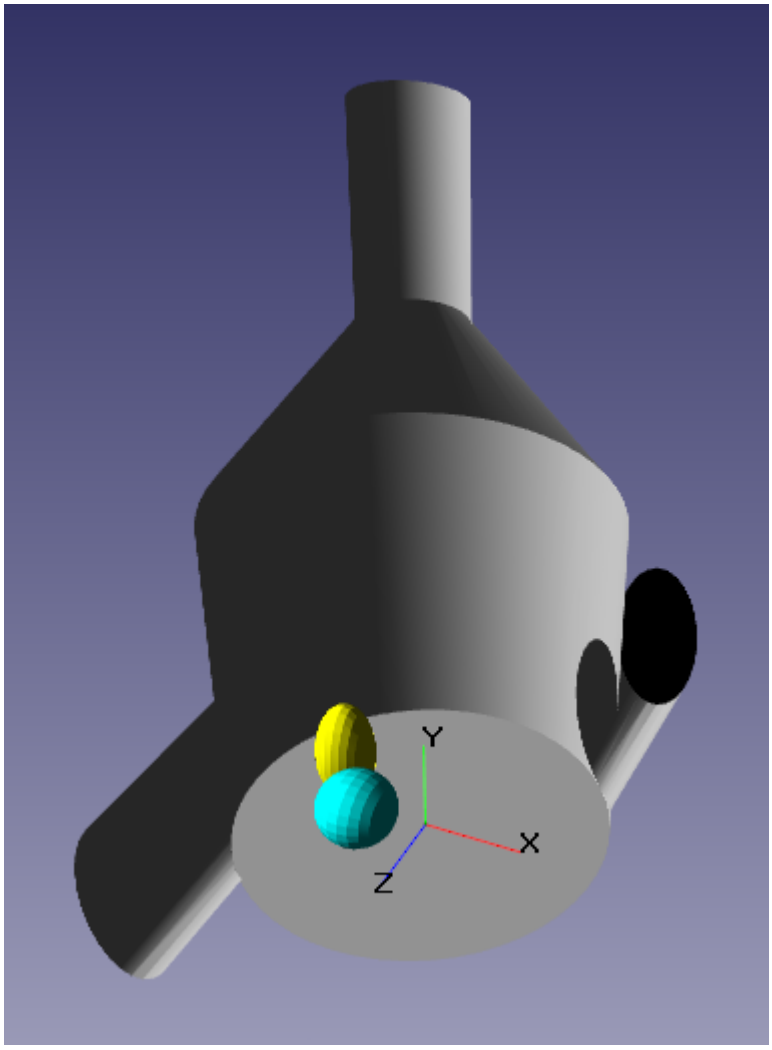
Example of the transformation geometry		
Ref	**Action**	**Illustration**
3	Made a rotation around the rotation axis passing through the center operations. In this example, rotation is performed around an axis parallel to axis Y, by 75 degrees, defined by the following parameters: **Rotation > = Y axis of rotation** **Rotation > Angle = 75**	
4	The **shift** is made. In this example, the shift is performed against the axis Z, defined parameters: **Shift > X = 0** **Shift > Y = 0** **Shift > Z =-0.006**	



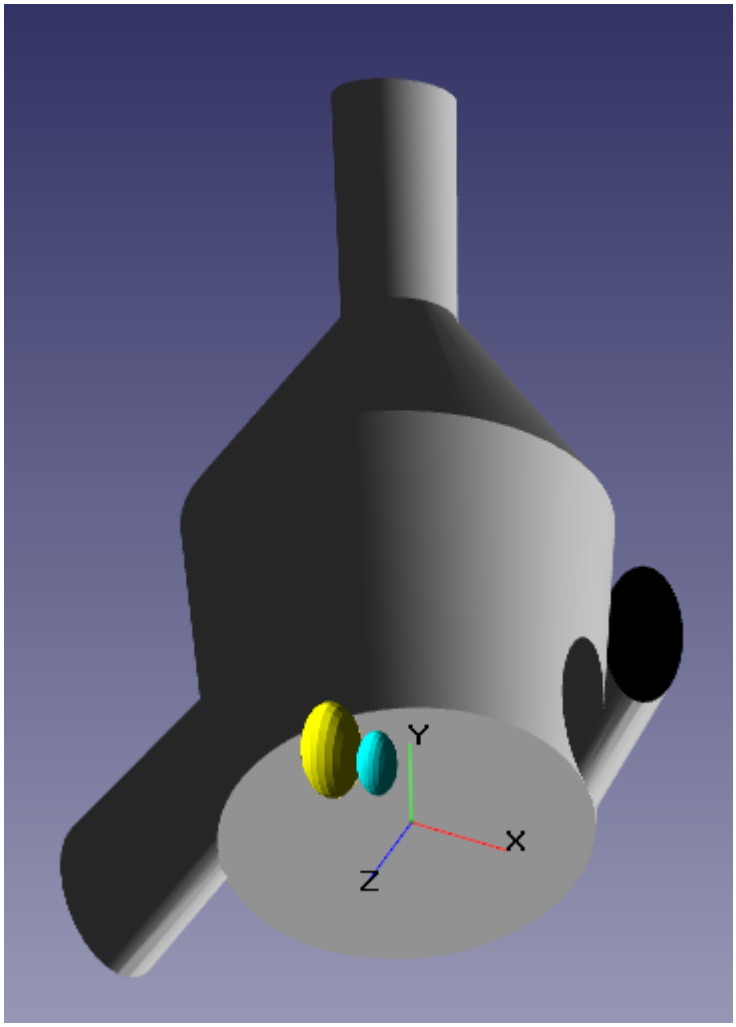
An example of a geometry model *to the* transformation. The geometry model consists of two **Subregions**: **SubRegion #0**- tap water model (gray) and **SubRegion #1** (yellow sphere). There is also a **Moving body** (light blue sphere).



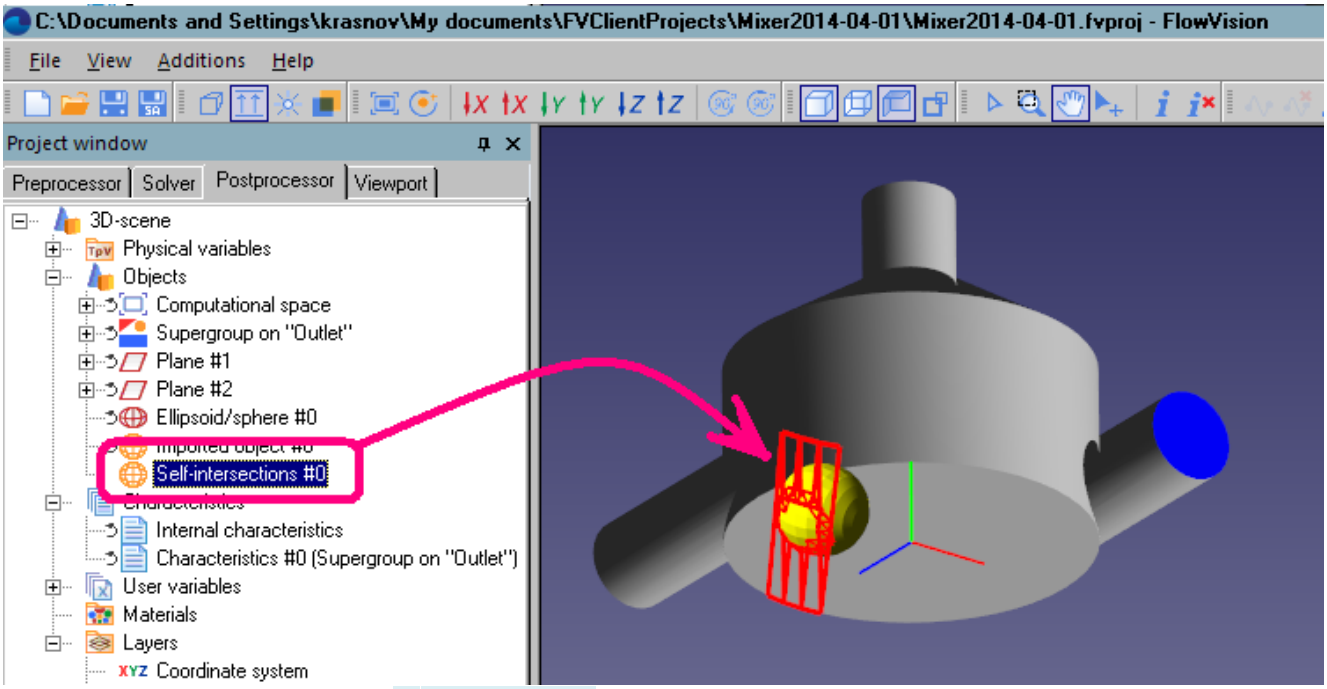
The result of the transformation of the **Region** - scaling on the X axis by a factor of 0.5.
Unlike the rest of the tank and the **SubRegion #1 Moving body** itself has not undergone transformation **(Transform command appliedgeometry)**.



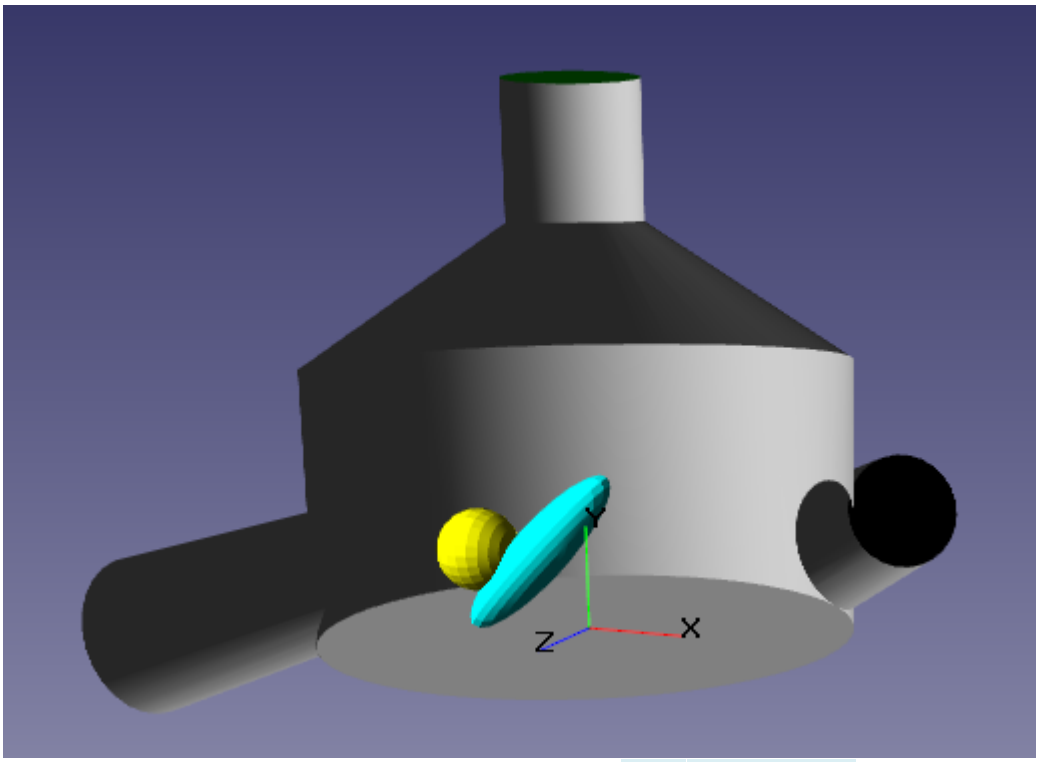
The result of the transformation of the **Region**- scaling in the X and Z by a factor of 0.5 for each axis. As in the previous example, **Moving body** has not undergone transformation (the **Transform geometry** command has been applied).



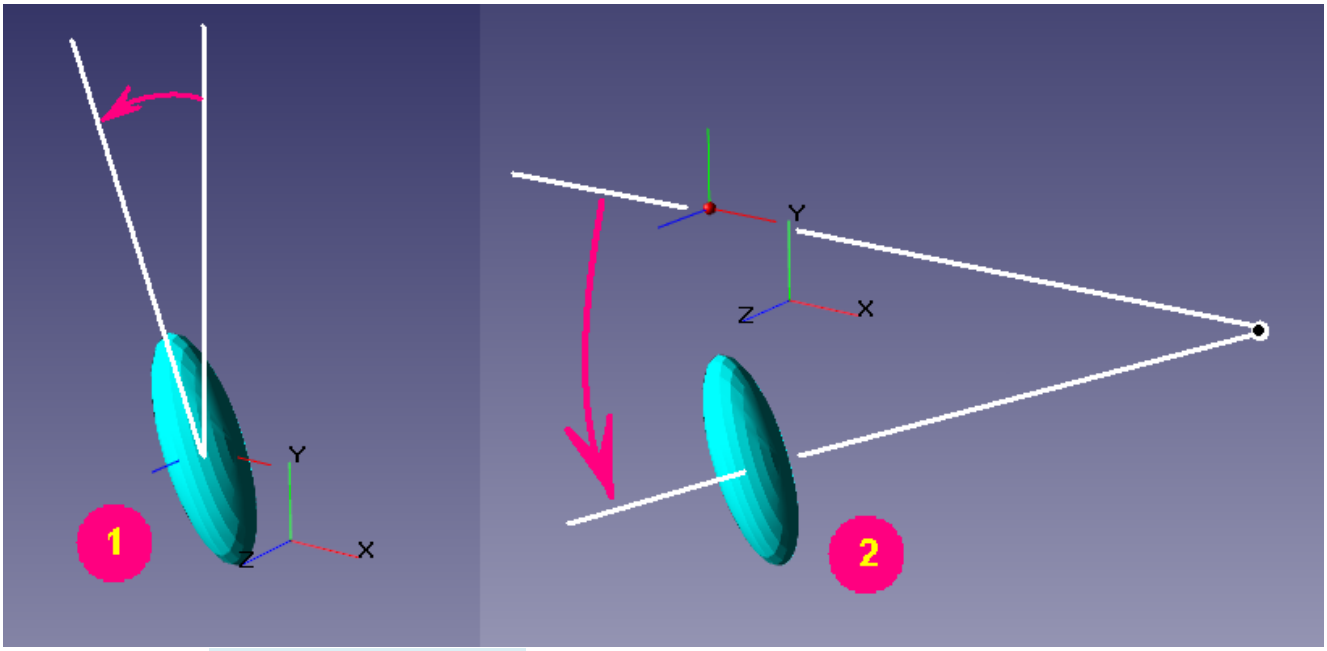
The result of the transformation of the **Region**- scaling in the X and Z by a factor of 0.5 for each axis. In this example, **Moving body**,too, has undergone transformation (the **Transform geometry + moving bodies** command has been applied).



As a result of the transformationof **Subregion #1** (scaling in the X and Z by a factor of 2 for each axis) any self-intersection of the computational domain
(Shown as item **Self-intersections #0** in the project tree and displayed in the **View**)



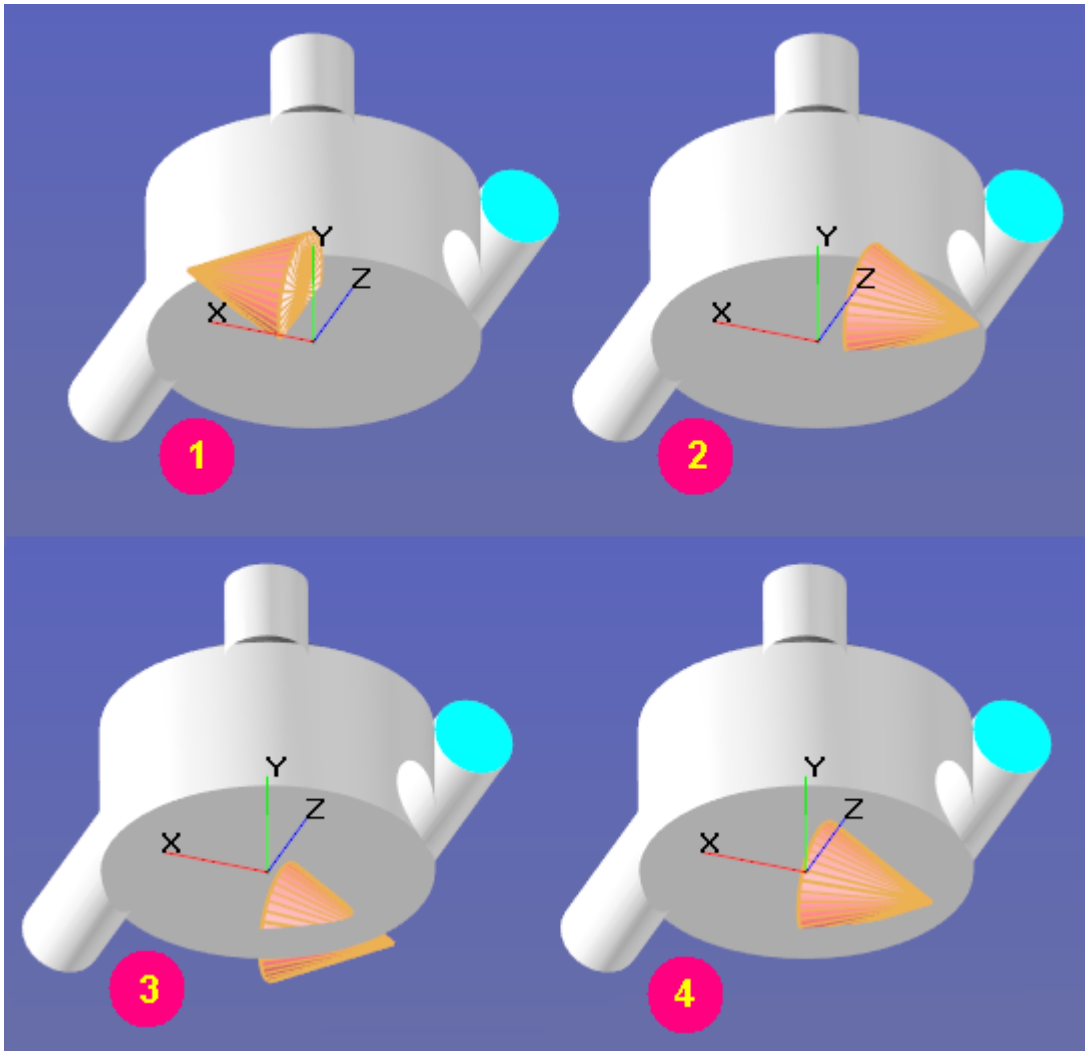
The result of applying transformations to the **Moving body**
(First - scaling the Y-axis by a factor 4, then rotation around the Z axis by 45 degrees)



Parameters **Transformation pivot > ...** specify the position of *the center of operations* of the geometry transformation.

In the examples in the figure shows the transformation **Moving body** counterclockwise rotation;

(1) - the center of operations locates in the center of the **Moving body**; (2) - the center of operations is shifted along the X axis



Specifying negative values of scaling factors can perform transformation of mirror symmetry.
The following figure shows: (1) - the original position of **Imported objects** (orange cone) prior to transformation;
(2) - a mirror image with respect to the plane YZ, specify the zoom ratio (-1,1,1) axes XYZ;
(3) - the central symmetry with respect to the origin, is given by the scaling factors of (-1, -1, -1);
(4) - axial mirror symmetry relative to the axis Y, specify the zoom ratio (-1.1, -1).
Center operations in these examples is the origin.

See also: section The transformation of a geometry model of the computational domain and the imported object

8.2.2.11 Operations in the View window

Operations performed in the **View** window include:

- operation display the entire computational domain When it becomes visible to the entire computational domain (operation is particularly useful if the computational domain "disappears" from the window of **View**);
- operation with the center of rotation - Rotate the scene in the **View** window is made around the center of rotation, which by default is located in one of two points:
  - in the geometric center of the model of the computational domain (if the project tree is not allocated to any object);
  - the geometric center of the selected object in the project tree.
- scaling operation
- operation orienteering scene turning the scene along one of the axes of the absolute coordinate system
- operation of rotating the image scenePerformed around the normal to the plane of the screen;
- operations performed in the mode transformation of the form
- turn the scene around the axes of the screen (By means of keyboard keys)
- change operation objects, performed with the mouse (During these operations the viewing angle of the scene does not change)

- identifying the group of facets

8.2.2.11.1 Displaying the whole computational domain

To display in the **View** window the entire computational domain, click the (**Fit calculation region to window**) button in the **Sights** toolbar.

8.2.2.11.2 Operations with the pivot

Shifting of the center of rotation in the reference point of the object

In the project tree, select the object, which will be the reference point is the center of rotation, and then click (**Set a new rotation center for the scene**) in the toolbar **Sights**.

Shifting of the center of rotation in the geometric center of the computational domain

Highlight the project tree all the computational domain (folder **Subregions** tab **Preprocessor** or folder **Objects > Computational space > Solids > Subregions** tab **Postprocessor**) and click (**Set a new rotation center for the scene**) in the toolbar **Sights**.

8.2.2.11.3 Operations of scaling

Image scaling in the **View** window can be performed using the mouse:

- in all display modes
- in the mode of transformation of the form
- in zoom mode

When scaling is fixed point in the center of the **View** window.

Zooming in all display modes

Place the mouse pointer on the panel review with the scroll wheel of the mouse. Rotation in one direction reduces the image to another - increases it.

Scaling mode transformation of the form

View transformation mode is enabled by clicking (**View transformation mode**) in the **Work modes** toolbar.

Place the mouse pointer in the window **View**, press the left and right mouse buttons, hold both of them, and then move the mouse cursor up and down in the **View** window.

Zooming in the zoom mode

Zoom mode is enabled by clicking (**Enable window zoom mode**) in the **Work modes** toolbar.

Increasing the portion of the image

Step	Actions
1	Place the mouse pointer to one corner of the frame and press the left mouse button.
2	Without releasing the left mouse button and drag the mouse pointer to the opposite corner of the frame and release the left mouse button. Image of the objects covered by the frame will be expanded to the entire **View** window.

Reducing the image in the View window

Place the mouse pointer at a certain point the **View** window and click the right mouse button. Preview in the **View** window is halved. The point at which clicked, will become the center of a new kind.

8.2.2.11.4 Scene orientation along axes of ACS

Orientation of the scene (rotation scene so that it is visible along one of the axes of the absolute coordinate system) is performed by pressing in the **Sights** toolbar.

Orienting the scene in the direction of the axis X (Y, Z) of the absolute coordinate system

To rotate the scene so that it was visible in the direction of the axis **X** (**Y**, **Z**) the absolute coordinate system click  ( , ).

To rotate the scene so that it was visible against the direction of the axis **X** (**Y**, **Z**) the absolute coordinate system click  ( , ).

8.2.2.11.5 Rotation of the scene

Rotate an image of the scene is carried around the normal to the plane of the screen and is performed using the buttons in the **Sights** toolbar.

Rotating the image in the View window 90 degrees (counter) clockwise

Click  () in the **Sights** toolbar.

8.2.2.11.6 Operations in the view transformation mode

View transformation modeis set by clicking  (**View transformation mode**) in the **Work modes** toolbar.

Among the operations performed in the mode transformation of the form include:

- operation with the image of the scene
- rotation operation around the rotation center scene

Operation with the image of the scene

As the number of operations depicting a scene performed by a mode transformation of the form include the following:

- image rotation around the center of rotation of the scene; image scene is rotated around the normal to the plane of the screen, passing through the center of rotation;
- displacement of the image of the scene;
- scaling of the image scene
- image rotation around the center of rotation of the scene with scaling; This operation is the sum of the rotation and scaling operations performed by moving the mouse pointer in the **View** window.

Rotating the image of the scene around the center of rotation

Place the mouse pointer on the panel review, press the left and right mouse button and hold both keys, move the mouse pointer to the left to the right in the **View** window.

Shifting the image of the scene in the View window

Place the mouse pointer on the panel review, click the right mouse button and keep it pressed, move the mouse pointer over the **View** window.

Rotating the image of the scene around the center of rotation with scaling

Place the mouse pointer on the panel review, press the left and right mouse button and hold both keys, move the mouse pointer over the **View** window.

Rotating the scene around the rotation center of rotation

Rotate the scene around the center of rotation is performed in the conversion type. Position of the axis of rotation depends on the movement of the mouse pointer on the screen (the scene is rotated about an axis passing through the center of rotation, the direction of the axis perpendicular to the direction of movement of the mouse pointer).

Rotating the image of the scene around the center of rotation

Place the mouse pointer on the panel review, click the left mouse button and keep it pressed, move the mouse pointer over the **View** window.

8.2.2.11.7 Rotation of the scene around axes of the display's screen

Rotating the scene around the axes of the screen (using the keyboard)

Action to be taken	Change Image
Pressing ← (→) key on your keyboard	Rotation around the vertical axis of the scene screen through the center of rotation
Pressing the ↑ (↓) key on your keyboard	Rotate the scene around a horizontal axis of the screen passing through the center of rotation

8.2.2.11.8 Changing objects using the mouse

In **Pre-Postprocessor** the valid operations on objects, you can do with the mouse in the**View**:

- changing parameters of a **Line** object
- changing parameters of a **Plane** object
- changing parameters of a finite volume object (it can be a **Box**, **Cone/cylinder**, **Ellipsoid/sphere**, **Imported object**)

Changing object parameters «Line»

Changing object parameters **Line** performed in **Object editing** mode,which is set by clicking the  (**Enable edit mode for selected object**) button in the **Work modes** toolbar.

This mode can be performed the following operations:

- rotate the line around the reference point
- move the reference point along the line
- move the reference point

Rotating the line around the reference point

Step	Actions
1	Select tab **Postprocessor** to **Line**.
2	Click the left mouse button and keep it pressed, move the mouse pointer.

Moving the reference point along the line

Step	Actions
1	Select tab **Postprocessor** to **Line**.
2	Press the left and right mouse button and hold it, move the mouse pointer.

Moving the reference point

Step	Actions
1	Select tab **Postprocessor** to **Line**.
2	Click the right mouse button and keep it pressed, move the mouse pointer.

Changing object parameters "Plane"

Editing the object **plane** is created on the tab **Postprocessor**, may be performed:

- in the mode of transformation of the form
- by editing the selected object

In the view transformation mode

View transformation mode is set by pressing  in the toolbar **Work modes**.

This mode can be performed the following operations:

- rotation of the plane around the reference point
- move the reference point along the plane
- move the reference point along the normal to the plane

Rotating of the plane around the reference point	
Step	**Actions**
1	Select tab **Postprocessor** to **plane**.
2	Follow the steps: a) press the **Shift** key on the keyboard and hold it; b) click the left mouse button and keep it pressed, move the mouse pointer.

Moving the reference point along the plane	
Step	**Actions**
1	Select tab **Postprocessor** to **Plane**.
2	Follow the steps: a) press *the* **Shift** key on the keyboard and hold it; b) click the right mouse button and keep it pressed, move the mouse pointer.

Moving the reference point along the normal to the plane	
Step	**Actions**
1	Select tab **Postprocessor** to **plane**.
2	Follow the steps: a) press the **Shift** key on the keyboard and hold it; b) press the left and right mouse button and hold it, move the mouse pointer.

In edit mode, the object parameters

Object editing mode is set by clicking the  (**Enable edit mode for selected object**) button in the **Work modes** toolbar.

This mode can be performed the following operations:

- rotation of the plane around the reference point;
- move the reference point along the plane;
- move the reference point along the normal to the plane (in this case moves along the normal vector).

Rotating the plane around the reference point	
Step	**Actions**
1	Select tab **Postprocessor** to **Plane**.
2	Click the left mouse button and keep it pressed, move the mouse pointer.

Moving the reference point along a plane	
Step	**Actions**
1	Select tab **Postprocessor** to **Plane**.
2	Click the right mouse button and keep it pressed, move the mouse pointer.

Moving the reference point along the normal to the plane	
Step	**Actions**
1	Select tab **Postprocessor** to **Plane**.
2	Press the left and right mouse button and hold it, move the mouse pointer.

Changing parameters of finite volume Objects (Boxes, Cones/cylinders, Ellipsoids/spheres, Imported objects)

Editing an object of finite volume (such as **Box**, **Cone/cylinder**, **Ellipsoid/sphere**, **Imported object**) that were created in the tab **Postprocessor**, may be performed:

- in the mode of transformation of the form
- by editing the selected object

In the view transformation mode

Object editing mode is set by clicking the  (**Enable edit mode for selected object**) button in the **Work modes** toolbar.

This mode can be performed the following operations:

- rotate the object around the reference point;
- move an object in a plane perpendicular to the line of sight;
- scaling object.

Rotating an object around the reference point	
Step	**Actions**
1	Select a finite volume **Object** in the **Postprocessor** tab.
2	Follow the steps: a) press the **Shift** key on the keyboard and hold it b) click the left mouse button and keep it pressed, move the mouse pointer.

Moving the reference point in a plane perpendicular to the line of sight	
Step	**Actions**
1	Select a finite volume **Object** in the **Postprocessor** tab.
2	Follow the steps: a) press the **Shift** key on the keyboard and hold it b) click the right mouse button and keep it pressed, move the mouse pointer

Scaling an object	
Step	**Actions**
1	Select a finite volume **Object** in the **Postprocessor** tab.
2	Follow the steps: a) press the **Shift** key on the keyboard and hold it b) press the left and right mouse button and hold it, move the mouse pointer

In the edit mode, the object parameters

Object editing mode is set by clicking the  (**Enable edit mode for selected object**) button in the **Work modes** toolbar.

This mode can be performed the following operations:

- rotate the object around the reference point;
- move an object in a plane perpendicular to the line of sight;
- scaling object.

Rotating an object around the reference point	
Step	**Actions**
1	Select a finite volume **Object** in the **Postprocessor** tab.
2	Click the left mouse button and keep it pressed, move the mouse pointer.

Moving an object in a plane perpendicular to the line of sight	
Step	**Actions**
1	Select a finite volume **Object** in the **Postprocessor** tab.
2	Click the right mouse button and keep it pressed, move the mouse pointer.

Scaling an object	
Step	**Actions**
1	Select a finite volume object in the **Postprocessor** tab.
2	Press both the left and right mouse buttons and keeping them pressed, move the mouse pointer up or down.

8.2.2.11.9 Selection a group of facets

Selecting a group of facets in the View window

Step	Actions
1	To enable the selection mode for groups of facets, click (**Enable facet group selection mode**) in the toolbar **Work modes** or press the **Ctrl** key and hold it.
2	Place the mouse pointer on the desired group of facets and left click the mouse. In the View window is the closest group of facets. One side of the group is a front surface and painted in the color specified for the given boundary conditions on it. On the reverse side of the surface pattern applied to the group of fragments . In the project tree, the focus is transferred to the panel **Preprocessor** and is allocated in the project tree line group facets.
3	To isolate the back side surface of the selected group, open the context menu and choose **Select another direction**.
4	Selecting a Group of facets, located in the back of the scene, carried out by one of the following ways: **Method 1:** a) move the mouse pointer to the point, which is located under the desired group of facets and open the context menu (right-click) b) select the**Select the next group**as long as you want is highlighted group **Method 2:** a) move the mouse pointer to the point, which is located under the desired group of facets; b) press the left mouse button as long as you want is highlighted group.

8.2.3 Operations with elements of the project tree

Operations with elements of the project tree is handled through the project tree (**Geometry**, **Preprocessor**, **Solver**, **Postprocessor**).

Operations with elements of the project tree on the tab «Preprocessor»

- operations with Substances
- operations with Phases
- operation to the physical processes
- operation with the model and its elements
- operations with local coordinate system (LCS), rotations and translation movements
- operations on objects
- operation with the characteristics
- operations with user variables
- operations with computational subregions and elements in them

- operations with boundary links
- operations with an initial grid

Operations with elements of the project tree on the tab «Solver»

The **Solver** tab of the **Project** window contains control parameters for the computation:

- time step
- choice of the numerical method and parameters of solving the system of algebraic equations
- limiters
- multiphase
- parameters of the turbulence model
- the export loads
- configure automatic saving results calculations
- configuration data automatically to render layers
- setting conditions for stopping the calculation

Operations with elements of the project tree on the tab «Postprocessor»

- operations on objects
- operation with the characteristics
- operations with user variables
- operations with Materials
- operation with Layers

Note: Elements **Objects**, **Characteristics** and **User variables** are presented in both tabs **Preprocessor** and **Postprocessor**.

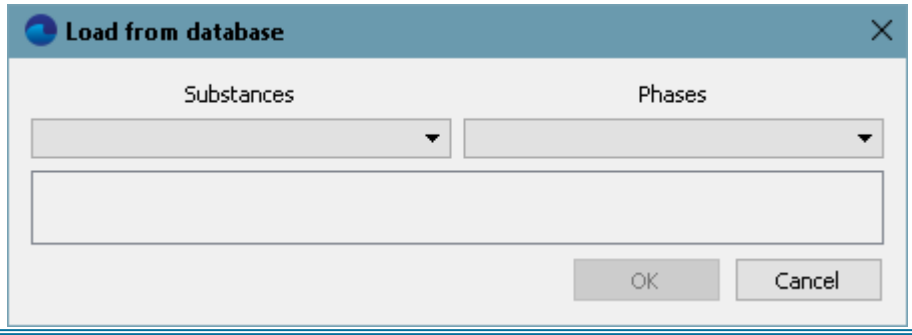
8.2.3.1 Operations with Substances

Creation of a substance with the properties of the task manually

Step	Actions
1	Right-click the folder row **Substances Preprocessor** tab and choose **Create**. In the folder **Substances** a new subfolder **Substance #N** will appear.
2	Select the folder created element **Substance #N**.
3	In the **Properties** window, select the drop-down list, **Appearance**, and then click **Apply**. In the folder list **Substance #N** material properties change so as to correspond to the selected state of aggregation.
4	Consistently set the values of the properties of the **Substance**: • selects an item in the folder properties of a substance**Substance #N** • in the **Properties** window to set the necessary value of the selected property in the**Value** field and click **Apply**.

Loading of the substance from the database

Step	Actions
1	Right-click the folder row **Substances Preprocessor** tab and choose **Create**. In the folder **Substances** folder is displayed new element **Substance #N**.
2	Right-click the folder, create a line item and choose **Load from SD > (name of the substance database**). The **Load from database** dialog box opens:

Step	Actions
	
3	Select a substance from the **Substances** drop-down list.
4	Select the phase of the substance from the **Phases** drop-down list.
5	Click **OK**.

See also: section Substance databases.

Changing properties of a Substance

Step	Actions
1	Select the folder material property (properties) that you want to change.
2	If necessary, change the aggregative state of the **Substance** - in the **Properties** window, select a value from the **Aggregative state** drop-down list and then click **Apply**. In the folder list **Substance #N** material properties change so as to correspond to the selected state of aggregation.
3	Consistently set the values of the properties of the substance: • selects an item in the folder properties of a substance **Substance #N** • in the **Properties** window to set the necessary value of the selected property in the **Value** field and click **Apply**.

When you change the physical properties of a **Substance** loaded from the **Base of substances** varies with the **type of element** from the **Base of substances** on the **Standard**.

Copying a Substance

Step	Actions
1	Right-click the object **Substance**, which is to be copied, and select **Copy**.
2	In the folder **Substances** folder is displayed new object **Substance #N**, whose parameters match those of the copy objects.

Removing a Substance

Step	Actions
1	If you delete a substance is included in phase (in the folder **Phases > Phase #N >Substances**), first remove it from **Phase**.
2	Right-click the object **Substance #N**, to delete, and select the **Delete** command. **Substances** removed from the folder directory object **Substance #N**.

8.2.3.2 Operations with Phases

Among the operations phase and the items within are the following:

- establishment phase
- up phase
- removal phase
- specify the substances in phase (add and delete)

Creation a Phase

Step	Actions
1	Right-click the folder **Phases** line and choose **Create continuous** or **Create dispersed**.
2	In your folder is displayed **Phases** new element **Phase #0**.

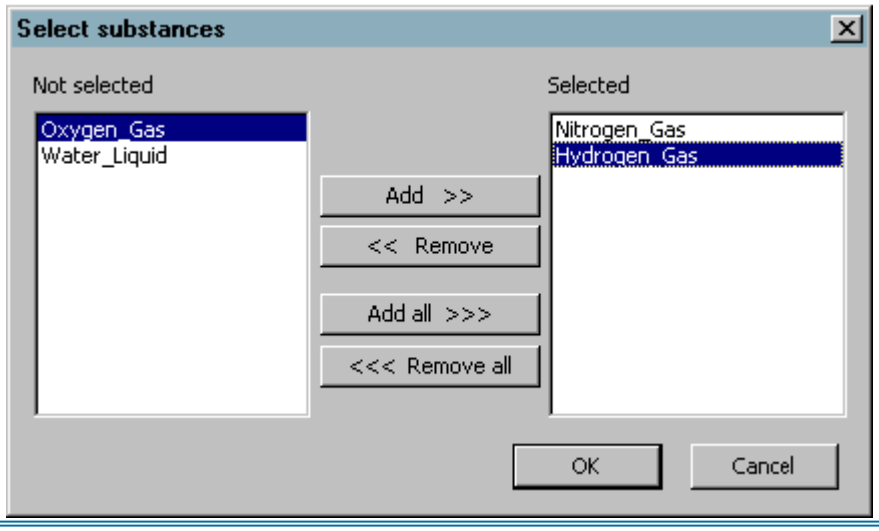
Copying a Phase

Step	Actions
1	Right-click the folder line **Phases > Phase #N**, be copied, and select **Copy**.
2	In the folder is added to the folder **Phases** of a new phase, the parameters of which coincide with the parameters of the selected phase.

Removal a Phase

Step	Actions
1	If the **Phase**, which is to be deleted, is included into a **Model** (in a folder **Models > Model > Phases**), first remove it from the model.
2	Right-click the folder line **Phases > Phase #N**, to be removed, and select **Delete**. In **Phases** folder deletes the folder selected phase.

Specifying Substances in a Phase

Step	Actions
1	Right-click the folder line **Phases > Phase #N >Substances Substance** or line in this folder and select **Add/remove**. In the **Select substances** dialog box, click the desired substances: 
2	Create a list of substances phase in the **Selected**:

Step	Actions
	• highlight the added substances from the **Not selected** list, and then click **Add**; highlight the substances, which are to be removed, in the **Selected** list and then click **Remove**. • to remove all substances, click **Remove all**
3	Click **OK**. Selected material displayed in the folder **Phases > Phase #N > Substances**. If you are in the phase of two or more substances in the Object **Properties Physical processes** should choose one of the models of mass transfer. If the model of heat transfer is not selected, line phase  displays a warning symbol "**!**".

If a **Phase** contains several **Substances** and the **Mixing** model of mass transfer is used, it is strongly recommended to enter the **Substances** in the **Select substances** dialog box so that the **Substance** with the maximal amount in the simulated process would be placed on the last position in the **Selected** list list.

(when the **Chemistry** or **Combustion** model of mass transfer is used, it doesn't matter in which order the **Substances** are listed)

Setting parameters of a Phase

Step	Actions
1	Specify substances contained in the folder **Phases > Phase #N >Substances** (see subsection *"Specifying Substances in a Phase"* above).
2	Highlight the folder **Phases > Phase > Physical processes** in the **Properties** window, create a list of physical processes and adjust their parameters.

8.2.3.3 Operations with Physical processes

As the number of operations with physical processes include the following:

- assignment of physical processes
- removal of a physical process
- editing parameters of a physical process

When defining the physical processes are the following features:

- physical process can only be set for the phase in which the substance is
- model of heat transfer can only be set for the phase, which indicates at least two substances
- turbulence model can be defined only for the phase with a predetermined motion pattern
- radiation model can only be specified for a given phase heat transfer model
- phase transfer model (**VOF model**) is set between two continuous media or between a continuous medium and vacuum

Specifying the physical processes

Step	Actions
1	Select the folder **Phases > Phase #N > Physical processes** in the project tree.
2	In the properties of the element **Physical processes**, select the appropriate physical processes in the drop-down lists of parameters and click **Apply**. Defined physical processes are displayed as rows of the **Physical processes** in the folder **Phases > Phase #N> Physical processes**.

Removing the physical process

Step	Actions
1	Select the folder **Phases > Phase #N > Physical processes** in the project tree and in the **Properties** window, select **None** in the exhaust line of the physical process.
2	In the folder **Phases > Phase #N > Physical processes** this element **Physical processes** will be removed.

Changing parameters of a physical process

Step	Actions
1	Open the folder **Phases > Phase #N > Physical processes** in the project tree and select a physical process, which is to be modified.
2	Adjust the values of the parameters in the **Properties** window, and then click **Apply**.

8.2.3.4 Operations with Models and their elements

Creating a Model

Open the context menu of the folder **Models** and select there the **Create** command.

In the folder **Models** folder is displayed a new element **Model #N**.

Copying a Model

Open the context menu of the folder **Models > Model #N**, of the **Model** which is to be copied, and select there the **Copy** command.

In the folder **Models** a new model will be added with parameters that are the same as parameters of the selected **Model**.

Deleting a Model

Step	Actions
1	If the deleted model is included in the calculation subregion (Parameter **Model Properties** window subregion), first remove it from the computational subregion.
2	Right-click the folder row **Models > Model #N**, to erase, and select **Delete**. The **Models** folder deletes the folder of the selected model.

Specifying a Model Phases and phase parameters of the interaction

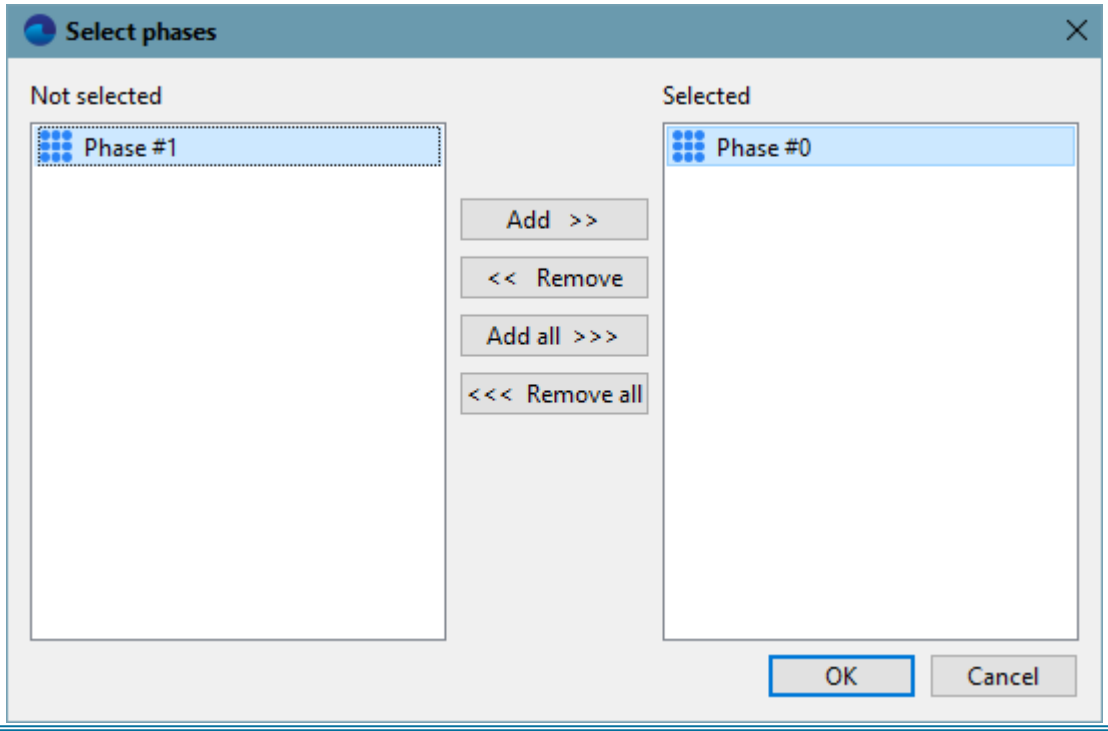
As the number of operations with the model include the following:

- designation of phases in the model (add and delete);
- Edit the interaction parameters of the phases.

Specifying the Phases, which are presented in a Model (adding and removing Phases)

If there is a two-phase medium for the calculation of the phase boundary, then the phases listed in the folder **Model #N > Phases** in the first place must be the phase with higher density, and the second - to a smaller or phase-vacuum (i.e., no matter whether or not the physical processes).

Step	Actions
1	Right-click the folder row **Models > Model #N > Phases** or line **Phase** in this folder and select **Add/remove**. The **Select phases** dialog box will open:

Step	Actions
	
2	Create a list of phases in the **Selected** list: • highlight the added phase in the list **Not selected**, and click **Add**; to add all the phases, click **Add all** • phase highlight excluded in the **Selected** and click **Remove**; to the exclusion of all phases click **Remove all**
3	Click **OK**.

To improve the stability of the solution must be in the**Select phase**arrange selected phase descending density.

For example, if one of the phases - liquid and the other - a gas or a vacuum, in the first place should be liquid phase and in the second - stage with gas or vacuum.

If the density of the phases of the same order, the phases can be specified in any order.

Selected phases appear in the folder **Models > Model #N > Phases**.

If the model has two phases, then in the folder **Models > Model #N > Phase interaction** displayed element **Phase interaction**.

To edit the parameters of the interaction of phases

Step	Actions
1	Select the folder **Models > Model #N > Phase interaction** element **Phase interaction**.
2	Adjust the values of the parameters in the **Properties** window, and then click **Apply**.

Operation with the initial data model

As the number of operations with initial data include the following:

- creation of the **Initial data**
- Removal of the initial data; folder **Models > Model #N > Init. data** you can remove all elements of the **Initial data**, except one
- Copy the **Initial data**
- editing of the **Initial data**

To create Initial data:

Step	Actions
1	Right-click the folder row **Models > Model #N > Init. data**, and then click **Create**.
2	In the folder **Models > Model #N > Init. data** folder will appear new element **Init. data #N**.

To delete Initial data:

Step	Actions
1	If you removed the initial conditions are included in the calculation subregion (in the folder **Subregions > subregion > Initial conditions**),first remove them from the computational subregion.
2	Right-click the folder row **Models > Model #N > Init. data** to be deleted, and select **Delete**. From the **Models > Model #N > Init. data** can remove all the **initial data** other than one. **Models** folder deletes the folder of the selected model.

To copy the initial data:

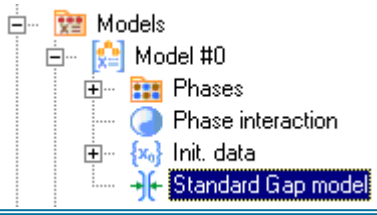
Step	Actions
1	Right-click the line of initial data in the folder **Models > Model #N > Init. data** to be backed up, and select **Copy**.
2	In the folder **Models > Model #N > Init. data** row is added to the new initial data, the parameters of which coincide with the parameters of the selected initial data.

To change the parameters of the initial data:

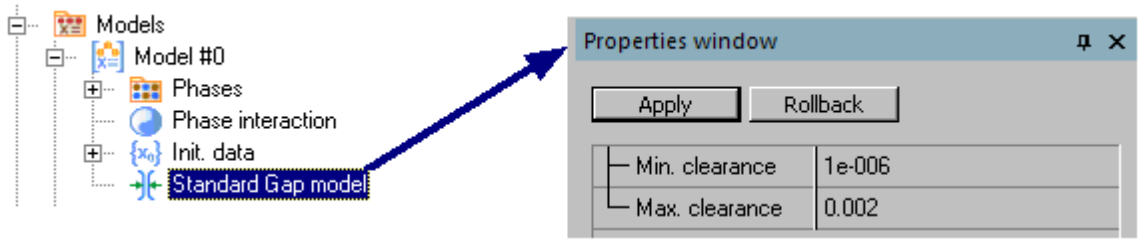
Step	Actions
1	Select a folder **Models > Model #N > Init. data > Init. data #N** and in the **Properties** window, change the name, if necessary, the initial data.
2	Expand **Models > Model #N > Init. data > Init. data #N**,sequentially select the items **Initial value of** variable, included in the folder, and set the initial values in the extended entry field.

Specifying use of a gap model

Toggling use of a gap model

Step	Actions
1	In the project tree select the line of the folder **Models > Model #N** and then in its **Properties** window select the desired value of the **Use Gap model** parameter.
2	Click **Apply**. In the **Models > Model #N** folder an appropriate gap model element will display or hide: 

Specifying parameters of a gap model

Step	Actions
1	In the project tree select the element, which corresponds to the used gap model (**Models > Model #N > Standard Gap model**):
2	Specify the desired values of the parameters in the **Properties** window of this element and then click **Apply**: 

8.2.3.5 Operations with local coordinate system (LCS), rotations and translation movements

Operation with the local coordinate system

- establishment of a local coordinate system
- removal of the local coordinate system
- editing parameters of local coordinate system

Creating a local coordinate system	
Step	**Actions**
1	Open the context menu of the folder **Local coordinate systems** and select the **Create** command. In the folder **Local coordinate systems** of the new folder is displayed element **Local CS #N**.

Deleting a local coordinate system	
Step	**Actions**
1	If you delete local CS included in other elements previously been excluded from all the elements.
2	Right-click the folder row **Local coordinate systems > Local CS #N** to be deleted and select **Delete**. In the folder **Local coordinate systems** deletes the folder selected local CS.

Change the settings of the local coordinate system	
Step	**Actions**
1	Select the folder **Local coordinate systems** to be changed element **Local CS #N**.
2	Adjust the values of the parameters in the **Properties** window, and then click **Apply**.

Operations with rotations

Some of the operations include the following rotations:

- establishment of rotation
- copying of rotation
- removal of rotation
- change in the parameters of rotation

Create rotation	
Step	**Actions**
1	In the context menu of the folder **Local coordinate systems > Local CS #N > Rotation** pane, click **Create**. In the folder **Local coordinate systems > Local CS #N > Rotation** will be a new element **Rotation #N**.

Copying rotation	
Step	**Actions**
1	Right-click the row of the **Local coordinate systems > Local CS #N > Rotation > Rotation #N**, be copied, and select **Copy**. In the folder **Local coordinate systems > Local CS #N > Rotation** will be a new element **Rotation #N**, whose parameters match those of the **Copy** item.

Removal of rotation	
Step	**Actions**
1	If the deleted rotation is included in other elements, you have previously exclude it from all of these elements.
2	Right-click the row of the **Local coordinate systems > Local CS #N > Rotation > Rotation #N**, to delete, and select **Delete**. In the folder **Local coordinate systems > Local CS #N > Rotation** removes the selected rotation.

Changing the rotation parameters	
Step	**Actions**
1	Select the folder **Local coordinate systems > Local CS #N > Rotation > Rotation #N** element to be changed.
2	Adjust the values of the parameters in the **Properties** window, and then click **Apply**.

Operations forth motions

Creating a translational motion	
Step	**Actions**
1	In the context menu of the folder **Local coordinate systems > Local CS #N > Translation**, click **Create**.

Creating a translational motion	
Step	**Actions**
	In the folder **Local coordinate systems > Local CS #N > Translation** will be a new element of **Translation #N**.

Copy the translational motion	
Step	**Actions**
1	Right-click the row of the **Local coordinate systems > Local CS #N > Translation > Translation #N**, be copied, and select **Copy**. In the folder **Local coordinate systems > Local CS #N > Translation** will be a new element of **Translation #N**,whose parameters match those of the **Copy** item.

Removal of the translational motion	
Step	**Actions**
1	If you remove a translational motion which is included with other elements, exclude it from all these elements.
2	Right-click the row of the **Local coordinate systems > Local CS #N > Translation > Translation #N**, to delete, and select **Delete**. In the folder **Local coordinate systems > Local CS #N > Translation** removes the selected translational motion.

Change the settings for the translational motion	
Step	**Actions**
1	Select the folder **Local coordinate systems > Local CS #N > Translation** be changed element **Translation #N**.
2	Adjust the values of the parameters in the **Properties** window, and then click **Apply**.

8.2.3.6 Operations with Objects

At the beginning of the project folders **Objects** in **Preprocessor** tab or **Postprocessor** tab contains the **Computational space** folder only.

Other objects as necessary are added to the folder **Objects**, edited, and deleted.

Operations with objects, user variables, characteristics are done:

- in the project tree through the context menu that opens in a rows of elements in the tabs **Preprocessor** and **Postprocessor**
- in the **Properties** window

Operation with the images of objects in the **View** window with your mouse.

Setting the display of objects

Setting the display of objects is handled through the **Postprocessor**.

Setting the display of objects include:

- operations that are common to all objects
- operation set the properties **Plane**

There is also an editing operation objects created tab **Postprocessor**, and the operation of selecting facets **Group** items, which is performed in the **View** window with the mouse.

8.2.3.6.1 Common operations for any Objects

Operations that are common to all **Objects**:

- hiding / showing the object; setting display object means that the object will always be displayedin the**View**; the establishment of the hidden object means that the object will be displayed in the **View** window only in the event it in the Project Tree tab **Preprocessor** or **Postprocessor**
- setting properties to be clipping object or cancellation of this property; establishment of property means that the object will be truncated in the **View** window to those objects **plane**, for which the property is set to be the object of a secant, in the **View** window shows the part of the object, which falls into the positive half-plane (half-subspace, in which the vector normal to the plane)
- setting properties of the object to be lit or cancellation of the properties (except the object **Line**); light sources are set in the **Lighting** folder in the **Postprocessor** tab of the **Project** window.
- delete the object

Setting hidden (display) of the Object

Do one of the following:

- Open the context menu of the **Object** in the **Postprocessor** tab and install (remove) mark in the **Hide** option
- *or* select the line items in the **Postprocessor** and in the **Properties** window, select **No** (**Yes**) for the parameter **is** displayed.

Setting the property to be a clipping Object

Do one of the following:

- Right-click the line items in the **Postprocessor** and install (remove) mark in the **Apply clipping** option
- select the line items in the **Postprocessor** and in the **Properties** window, select **Yes** (**No**) for the parameter **are** truncated

Setting the property to be lit Object

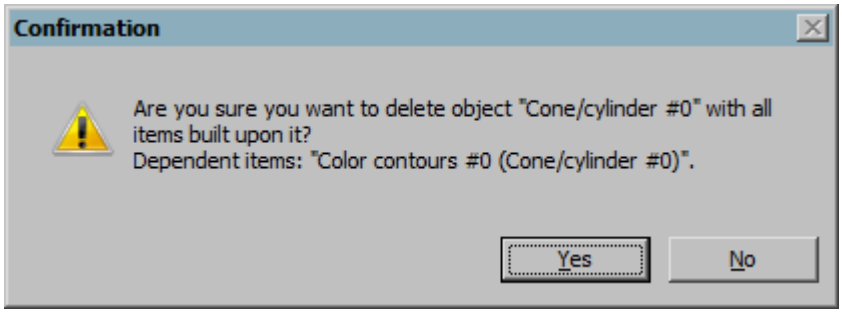
Do one of the following:

- Right-click the line items in the **Postprocessor** and install (remove) mark in the **Apply lighting** option
- select the line items in the **Postprocessor** and in the **Properties** window, select **Yes** (**No**) for the parameter **is** lighted.

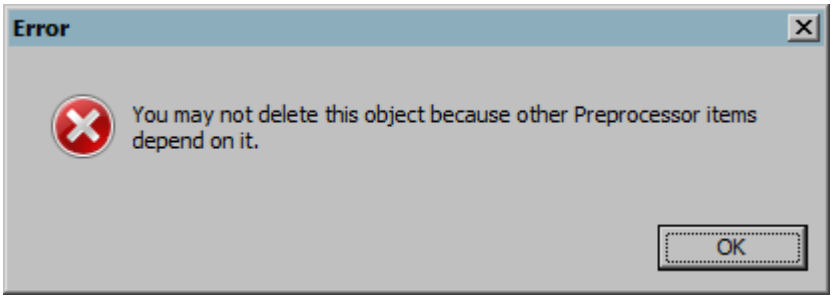
Deleting an object

Deleting objects is performed by selecting the **Delete** context menu opens on the line object in the project tree.

If the deleted **Object** contains elements built on it, a window asking you to confirm deletion of **Objects**, together with all the elements built on it:



The operation is complicated in the case of the removal of the associated entity that is the object used in the project tree as a component of any of its elements. If these links are not broken, then the program when you click **Delete** displays a message about the impossibility of deleting the **Object** (**You may not delete this object because other Preprocessor items depend on it**)



To delete a linked **Object**, you must first break the bond between the **Object** and other elements.

8.2.3.6.2 *Specific settings of the Plane object*

Setting the properties of the object to be secant

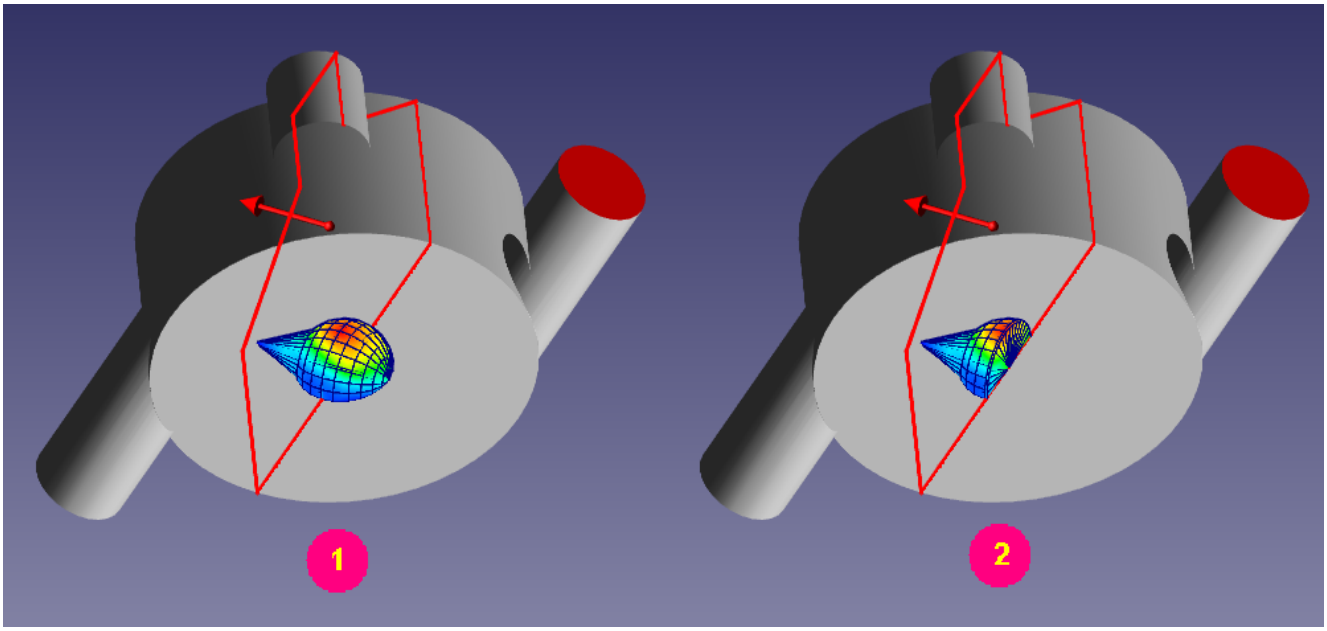
Property *clipping* **Planes** be *subject to* means that the plane will be cut off in the **View** all **Objects** and **Layers** for which the property is set in **"negative."**

Depending on this property, a **Plane**'s icon in the project tree in the **Postprocessor** tab is:

- - for a **Plane**, which *is not* a clipping object
- - for a **Plane**, which *is* a clipping object

Do one of the following:

- *either* open the context menu of the **Plane** in the **Postprocessor** tab and set (or remove) a mark in the **Clipping object** option
- *or* in the **Properties** window of the **Plane**, select the desired value for the **Clipping object** parameter



Property of a **Plane** to be a secant object:

1 - the **Plane** is not secant object;

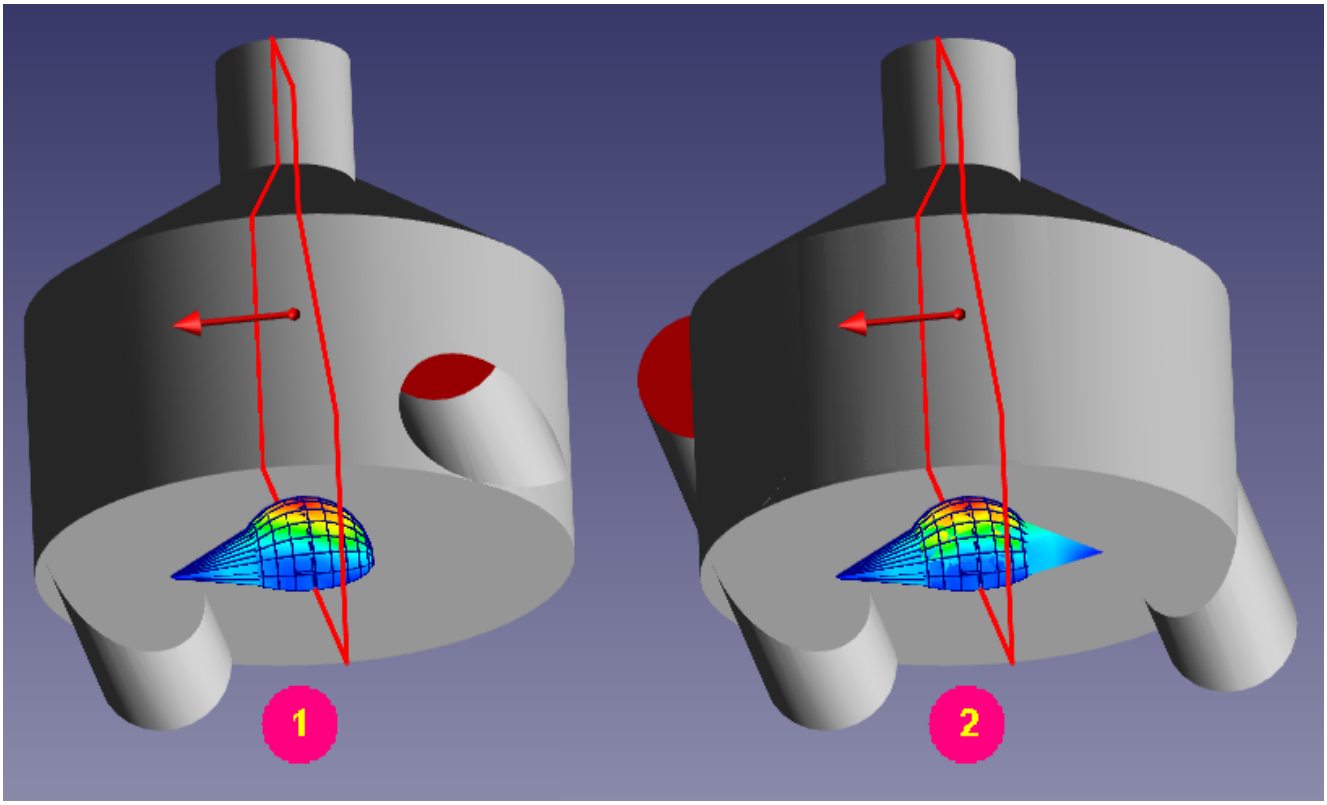
2 - the **Plane** is a secant object and cuts **Objects** and **Layers**, in which properties **Clipped=Yes** is set

Setting the properties of the object to be mirrored

Plane property be *mirrored object* means that the plane will be reflected in its negative half-images, which are located within the computational domain in the positive half (half-subspace, in which the vector normal to this **Plane**).[1.2)]

Do one of the following:

- Right-click the line items in the **Postprocessor** and install (or remove) a mark in parameter **Mirror**
- select the line items in the **Postprocessor** and in the **Properties** window, select **Yes** (or **No)** parameter **Mirror**



Plane property be mirrored object: 1 - not a mirror plane; 2 - a plane of mirror (reflecting its positive half-subspace in the negative half-subspace)

Notes:

[1] Mirror Images layers work in the same sequence in which were set

[2] In one project may not work more than three mirror images

8.2.3.6.3 Operations with standard geomentric objects

The following operations are available with standard geometric objects:

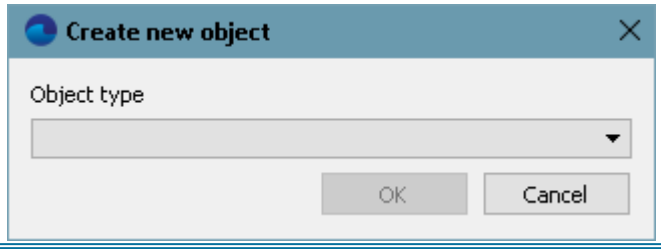
- the addition of a standard object in the project (creating, copying)
- removal of the standard object of the project
- change in the parameters of the object (item);
- incorporation into the computational domain of the standard object of finite volume:
  - **Box**
  - **Cone/cylinder**
  - **Ellipsoid/sphere**
- conversion of standard object **Imported object**

Adding a standard object in the project (creating, copying)

Adding a standard object in the project is carried out by:

- create a new object;
- copying an existing standard object;
- copying an existing standard object with a child element.

Creating a standard object	
Step	**Actions**
1	Use the **Create** command from the context menu on the line folder **Objects**. Once clicked, the dialog box **Create new object**:

Creating a standard object	
Step	**Actions**
	
2	Select the drop-down list the type of object, and then click **OK**. Created object is added to the folder **Objects**.
3	Select the added object and set the required parameters in the **Properties** window.

Creating a new object by copying an existing object	
Step	**Actions**
1	Select the line object in the folder **Objects**, open the context menu and select **Copy**. Created copy of the object is added to the folder **Objects**.
2	Select the added object and set the required parameters in the **Properties** window.

Creating a new object by copying an existing object with a child element	
Step	**Actions**
1	Select the line object in the folder **Objects**, open the context menu and select **Copy** from the elements. Created copy of the object is added to the folder **Objects**.
2	Select the added object and set the required parameters in the **Properties** window.

Changing object parameters	
Step	**Actions**
1	Select the line object in the **Objects** folder and edit the necessary settings in the **Properties** window.
2	Click the **Apply** button in the **Properties** window.

8.2.3.6.4 Operations with an Object's Movement

Creation an Object's Movement

Step	Actions
1	Open the context menu of a standard geometric object with finite volume (**Box**, **Cone/cylinder**, **Ellipsoid/sphere**) or of an **Imported object** (if it is not specified modifier **Moving body**), and select there the **Create movement** command.
2	In the project tree a **Movement** element will be created as a child element of the **Object**.

Removing an Object's Movement

Step	Actions
1	Open the context menu of the **Movement**, which is to be deleted, and select the **Delete** command.
2	The **Movement** will be removed from the project tree.

Changing parameters of an Object's Movement

Step	Actions
1	Select a **Movement** in the project tree and set its parameters in the **Properties** window.
2	Click the **Apply** button in the **Properties** window.

If a **Movement** is set on an **Object**, then the Object's location parameters are not available, but you can use the similar parameters of the **Movement** that specify location of the **Object** at the initial time moment of the **Movement** (these parameters locate in the **Movement**'s **Properties** window in the **Initial position** group of parameters).

See also: Element «Movement».

8.2.3.6.5 Operations with a Supergroup

Below is a description of the operations with **Supergroups**:

- creation a **Supergroup**
- copying a **Supergroup**
- copying a **Supergroup** with its child elements
- copying and conversion a **Supergroup** to an **Imported object**
- deleting a **Supergroup**

Create a Supergroup

Step	Actions
1	In the subregions of the computational domain, specify a single boundary condition for all groups of the surface facets of the computational domain, which should form a supergroup. This boundary condition should not be specified for groups of facets that should not enter into a supergroup. Assigning a single boundary condition may be temporary, only for the operation of creating a supergroup. You can then change the boundary conditions on the groups that make up the supergroup, but the object of the supergroup will not change.
2	Right-click the line **Subregions > SubRegion #N > Boundary conditions > B. Cond #N**, and then click: • **Create supergroup > preprocessor** • *or* **Create supergroup > postprocessing**

Step	Actions
	The created object **Supergroup on B. Cond #N** is added into the folder **Objects** in the **Preprocessor** or in the **Postprocessor** tab.

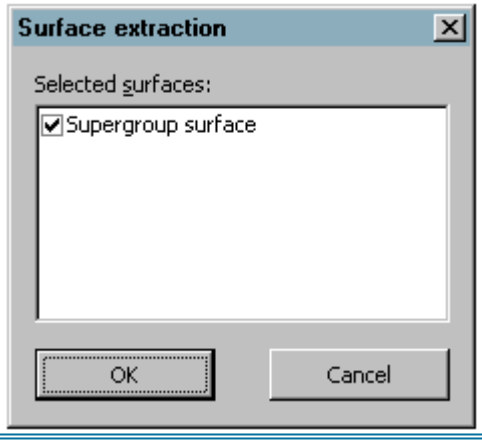
Create a new Supergroup copying existing Supergroup

Step	Actions
1	Select the row of a supergroup in the folder **Objects**, open the context menu and select **Copy**. Created copy of the object is added to the folder **Objects**.
2	Select the added object and set the required parameters in the **Properties** window.

Create a new Supergroup copying an existing Supergroup with child elements

Step	Actions
1	Select the row of a supergroup in the folder **Objects**, open the context menu and select **Copy** from the elements. Created copy of the object is added to the folder **Objects**.
2	Select the added object and set the required parameters in the **Properties** window.

Copying transformation a Supergroup into an Imported object

Step	Actions
1	Select the row of a supergroup in the folder **Objects**, open the context menu and select **Copy as imported object**. Once clicked, a dialog box **Copy** surfaces: 
2	Select/deselect a list of surfaces and then click **OK**. Created imported object is added to the folder **Objects**.

Removal a Supergroup

Step	Actions
1	If the deleted **Supergroup** is included in other elements, previously exclude it from all the elements.
2	Right-click the term of a supergroup **Objects** folder and select **Delete**. **Supergroup** is removed from the project tree.

8.2.3.6.6 Operations with Imported objects

Operation with **Imported objects** include:

- Transformation of an Imported objectadd the imported object to the project (creating, copying)
- Transformation of an Imported object
- Replacement of an Imported object
- Exporting the surface of an Imported object into a file
- Splitting an Imported object into separate Imported objects
- Small offset of the surface of an Imported object (equidistant offset)
- Surface scan of an Imported object to the self-intersection
- Fixing self-intersections of the surface of an Imported object
- Removing too small facets of the surface of an Imported object
- Embedding an **Imported object** into the computational domain.

and editing operations are objects created tab **Postprocessor**, perform in a window **View** using a mouse.

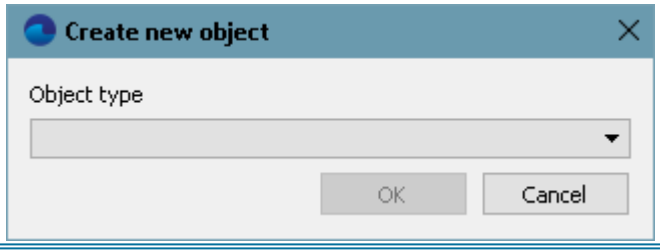
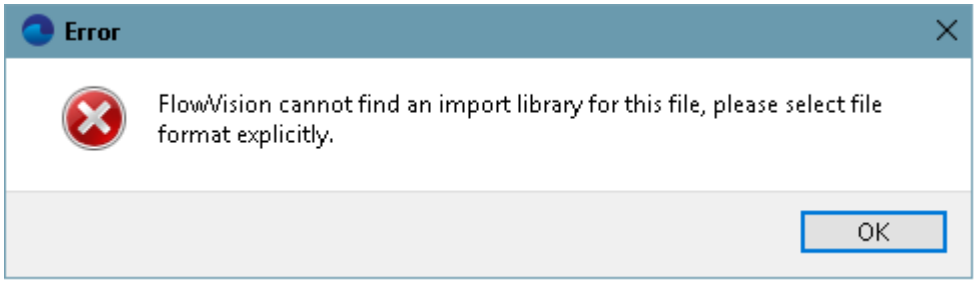
8.2.3.6.6.1 Adding an Imported object to the project

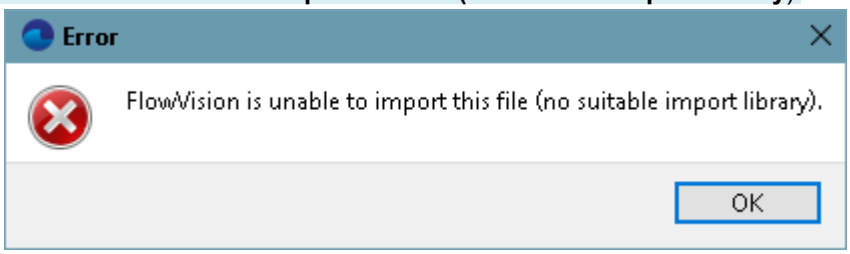
Adding an **Imported object** to a project is carried out by one of the following ways:

- Importing files with the object specified in the CAD-system
- Copying of the imported object
- Copying of the imported object with child elements (tab only **Postprocessor**)
- copy of the standard object with conversion to type **Imported object**

In *FlowVision* it is also possible to import a group of external objects.

Import a single object

Step	Actions
1	Open the context menu of a folder **Objects** and then click **Create**. Once clicked, the dialog box **Create new object**: 
2	From the drop-down list **Object type**, select **Imported object** and click **OK**.
3	From a standard operating system's dialog box, select a file, from which is the geometry of the created **Import object** is loaded, and then click **Open**. The **Imported object** will be added into the folder **Objects**. If the reading of file's contents failed (for example, due to an incorrect format or if the file is damaged), the program will display an error message (**FlowVision cannot find an import library for this file, please select file format explicitly.**):  This error can be caused by the following reasons: • *FlowVision* failed to find an appropriate library to import this file. • The file is damaged or contains incomplete geometry data. • The geometry doesn't meet *FlowVision*'s requirements to geometric objects.

Step	Actions
	In the standard operating system's dialog box, specify the required file format (file type), then select the required file and click **Open**. If the contents of the file is still unacceptable and/or damaged, the program displays an error message (**FlowVision is unable to import the file (no suitable import library)**:  Repeat steps 1-3 with another file that is not damaged and has the correct format.

Import a group of objects

Step	Actions
1	Open the context menu of a folder **Objects** and select **Batch import**.
2	Select from the drop down standard **File Open** dialog box,from which objects are imported, and then click **Open**. Imported objects are added to the folder **Objects**.

Creating a new object by copying an existing object

Step	Actions
1	Right-click the line imported object in the folder **Objects** and select **Copy**. Created copy of the object is added to the folder **Objects**.
2	Select the added object and set the required parameters in the **Properties** window.

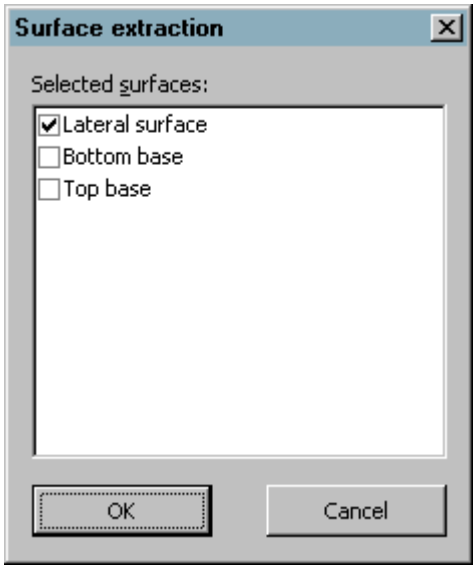
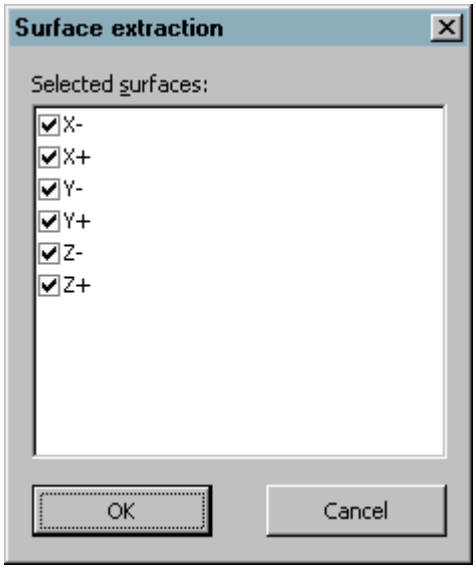
Creating a new object by copying an existing object with a child element

Step	Actions
1	Right-click the line imported object in the folder **Objects** and choose **Copy from the elements**. Created copy of the object is added to the folder **Objects**.
2	Select the added object and set the required parameters in the **Properties** window.

Copying-conversion of a standard object to an "Imported objects"

Converting to **Imported object** is only allowed for objects of finite volume: for **Box**, **Cone/cylinder**, **Ellipsoid/sphere**, and **Supergroup**.

Step	Actions
1	Right-click the line of standard object in the folder **Objects** and choose **Copy as imported object**. Once clicked, a dialog box **Surface extraction**, which lists the existing standard object surface (the list depends on the type of the object):

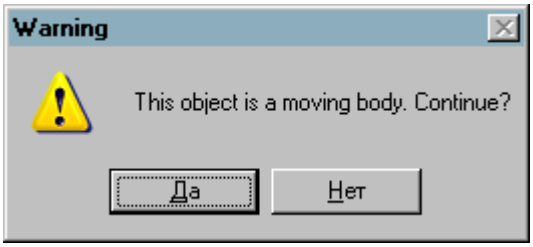
Step	Actions
	 
2	Select / unselect a list of surfaces and then click **OK**. Created imported object is added to the folder **Objects**.

8.2.3.6.6.2 Transformation of an Imported object

Geometry **Imported objects** can be transformed (i.e., perform compression / expansion / reflection, the spatial turn and shift).

Transformations of **Imported objects** are performed in the local coordinate system, that is transformed **Imported object** can change its orientation and position in the local coordinate system. Also transformed **Imported object** can change its size and aspect ratio (and also converted using the mirror, the mirror axis and the center of symmetry).

Transforming an Imported object

Step	Actions
1	Right-click the **Imported object** in the project tree (folder **Region > Objects > Imported object #N**) and select **Transform geometry**. Once clicked, the **Geometry transformation** dialog box will open (see section Transformation of geometry model of the computational domain and imported object).
2	If the **imported objects** defined **Moving body**, then the program will ask for confirmation before executing the transformation (**This object is a moving body. Continue?**):  In this case, the transformation of the geometry will be applied not only to the **Imported objects**, but also to the specified **Moving body** on it.
3	Set the parameters of the transformation of the object and click the **Apply** button to preview in the **View** window,the proposed transformation. If the results do not satisfy the transformation, change the transformation parameters and click **Apply** again. Repeat this step until you get the desired results.
4	Click **OK**, to actually carry out the transformation geometry in accordance with the entered data. The **Geometry transformation** dialog box will be closed and made the transformation geometry

Step	Actions
	Imported objects that properly displayed in the **View**.
5	If you need to perform some transformations (eg, several consecutive turns), repeat the previous steps as many times.

See also:

Details and illustrations in the section Transformation of geometry model of the computational domain and imported object.

8.2.3.6.6.3 Replacement of an Imported object

Replacing an Imported object

Step	Actions
1	Open the context menu of the **Imported object** in the folder **Objects** and select **Replace geometry**.
2	Opens a standard **Open** dialog box, select the file from which the object is imported, and then click **Open**.

8.2.3.6.6.4 Exporting the surface of an Imported object into a file

Exporting the surface of an Imported object into a file

Step	Actions
1	Right-click the line imported object in the folder **Objects** and select **Export** geometry. Once clicked, a standard dialog box **Export** geometry.
2	Select the file type, select the file name and click **Save**.

8.2.3.6.6.5 Splitting an Imported object into separate Imported objects

Splitting an Imported object, consisting of several unrelated surfaces on individual Imported objects

Step	Actions
1	Right-click the line imported object in **Objects** folder and select **Split by** surfaces.
2	Displays a message box with the results of the partition.

8.2.3.6.6.6 Small offset of the surface of an Imported object (equidistant offset)

Small changes in the surface of the object can be imported, in particular, is performed to achieve the following objectives:

- receipt of the computational domain with non-overlapping boundaries, constructed on the basis of the assembly drawing and detail drawings (border crossing is generated by the fact that the assembly drawing may be a small overlap of parts, which is physically removed during assembly, but requires specific actions during preparing a project in *FlowVision*);
- Specifying a small gap between adjacent parts, with subsequent determination it special conditions (for example, thermal resistance).

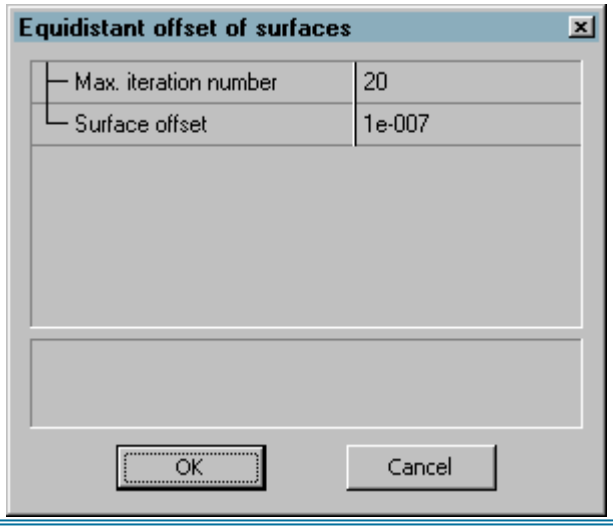
Equidistant shear geometry- creating a new surface components obtained by shifting the starting point inside or outside surface of the items a selected amount. It is designed to address the intersection of the elements of geometry. The shift can be carried out several times until the intersection will not be eliminated.

The program when the shear surface equidistant

A small change in the size of the object is made of nodes offset facets inside or outside the body. The user specifies the maximum amount of shift in the parameter **Offset**. Since the displacement of nodes facets can lead to self-intersection surface, the procedure for changing the surface of the object is performed iteratively.

Step	Description
A	The amount of shift for all facets of nodes is assumed to be the maximum.
B	Calculate the position of the displaced nodes facets after the shift.
C	Self-intersection of the surface being tested, and if there is no self-intersection, the procedure is completed.
D	If you have any self-intersection of the surface, defined by crossing the facets. For nodes these cells the shift amount is reduced by 20% for the other node offset unchanged.
•	Further, the procedure is repeated from step **B** (the number of repetitions is limited by the user in the parameter **Max. Number of iterations**).

Implementation of the equidistant surface shear

Step	Description
1	Open the context menu of the **Imported object** in the **Objects** folder, and then select the **Perform equidistant geometry offset** command. The **Equidistant offset of surfaces** dialog box will open: 
2	Set the parameters of the equidistant shift and then click **OK**.
3	Check for self-intersection of the surface of the object:

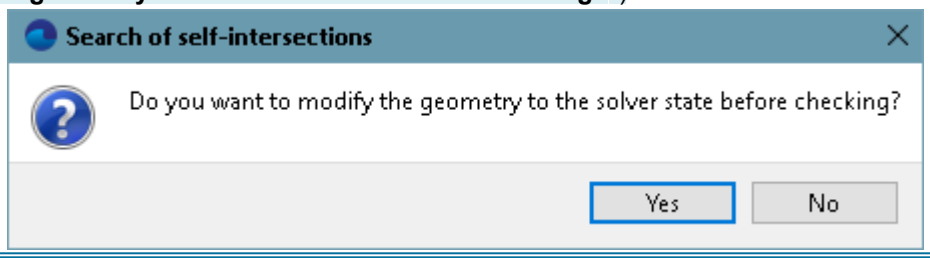
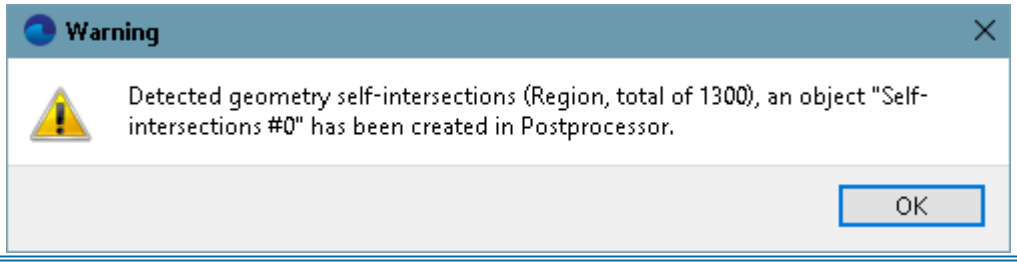
Parameters of equidistant offset

Parameter	Description
Max. iteration number	The maximum number of iterations equidistant offset
Surface offset	The maximum amount of shift of the new surface with respect to the initial (maximum displacement nodes facets per iteration). It is recommended to be set in the range of 3 to 7 parameter values **Geometry import > Tolerance** in the basic settings of **Pre-Postprocessor** (they are opened by the **File > Preferences** menu command). If you specify a positive value, the nodes are displaced inside the facets of the object, if negative - out.

Implementation of the equidistant surface shear	
Step	**Description**
	• open the context menu of **Imported objects** in the folder,and then click **Objects Check geometry for self-intersections** • if there is no self-intersections, the system will notify the information window • if the self-intersection is, then remove them (see section Fixing self-intersections of a surface in a geometry model)

8.2.3.6.6.7 Checking the surface of an Imported object for self-intersections

Checking the surface of an Imported object to the self-intersection

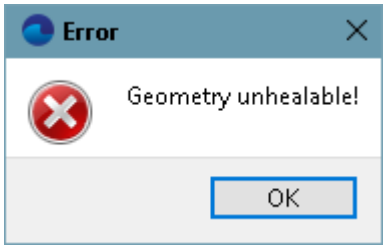
Step	Actions
1	Right-click the line of an **Imported object** in the folder **Objects** and select **Check geometry for self-intersections**. The program will display a query window (**"Search of self-intersections" / "Do you want to modify the geometry to the solver state before checking?"**): 
2	If the initial grid has already been built, click **Yes**. If the initial grid is not yet built, click the **No** button.
3	The system searches for self-intersections. Upon detection of self-intersections of the system performs the following actions: • folder **3D-scene > Objects > Computational space** created imported to **Self-intersections #0** this property consists of intersecting facets and created as an auxiliary to analyze the reasons that led to an error in setting the computational domain; after analysis of the object should be removed; • in the **Project** tab displays **Postprocessor** and you will see a warning (**Detected geometry self-intersections (... total ...), an object "Self-intersections #0", has been created in Postprocessor**): 
4	Click **OK**.

See also: Fixing self-intersections of the surface of an Imported object.

8.2.3.6.6.8 Fixing self-intersections of the surface of an Imported object

Eliminating self-intersections of surface of an Imported object

Step	Actions
1	Open the context menu of the **Imported object** in the folder **Objects** and select **Fix self-intersections**.

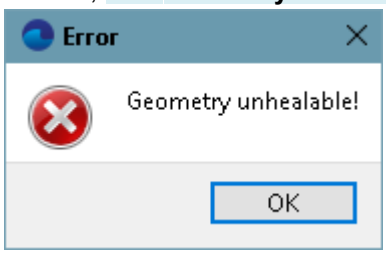
Step	Actions
	Displays information about the results of correcting self-intersections: • if the operation is to eliminate the self-intersection is unsuccessful, the message **«Geometry unhealable!»**:  In this case it is necessary to correct the geometry model of the surface by means of CAD-systems. It is recommended to change the export geometry model in the CAD-system, or fix bugs grid in a special program (eg, *3DTransVidia*); • if the operation of removing the self-intersections is successful, then a dialog box with information about results of the correction will open (**Self-intersections have been fixed, ...% of surface area has been deleted. Save changes and continue with new geometry?**).
2	If area of the deleted surface is 0.00%, this means absence of self-intersections. Click **No**. If area of the deleted surface is sufficiently small, then click **Yes**. If area of the deleted surface is large, click the **No** button and correct the imported surface by means of CAD-systems.

The functionality of automatic fix of self-intersections does not guarantee positive results, because its algorithm is not applicable for all situations.

Fixing self-intersections of elements **Self-intersections #N** is done in the similar way:

- In the context menu, select **Fix self-intersections**.
- In the dialog box, which opens, if you have not built the computational grid yet, select **No**. Otherwise, select **Yes**.

If automatic correction of self-intersections failed, the «**Geometry unhealable!**» error message will be displayed:



In this case it is recommended to change the export geometry model in the CAD-system, or fix bugs grid in a special program (eg, *3DTransVidia*).

8.2.3.6.6.9 Removing too small facets of the surface of an Imported object

To remove too small facets of surface of an Imported object:

Step	Actions
1	Right-click the line imported object in the folder **Objects**, and then click **Delete too small triangles**. Displays a window with information about the results of correction.
2	If small facets removed, it will increase the value of the**smallest edge**in the**Properties** window of the imported object.

8.2.3.7 Operations with Characteristics

Creating Characteristics

Step	Actions
1	Open the context menu of the Characteristics folder and select **Create**.
2	New element **Characteristics #N (<object>)** is displayed in the project tree in two places: **Characteristics** in the folder and the folder of the **Object** on which characteristics is built.

Removing Characteristics

Step	Actions
1	If the deleted characteristic is included in the other elements previously been excluded from all the elements.
2	Open the context menu of **Characteristics** to be deleted and select **Delete**. Item **Characteristics #N (<Object>)** is removed from the project tree in two places: **Characteristics** of the folder and the folder **Object** on which was built characteristics.

Copying Characteristics

Step	Actions
1	Open the context menu of the element **Characteristics** to be copied, and select **Copy**.
2	New element **Characteristics #N (<object>)** is displayed in the project tree in two places: **Characteristics** in the folder and the folder **Object** on which is built parameters of Characteristics Parameters element coincide with the parameters copied characteristics.

Displaying parameters specified by Characteristics

Step	Actions
1	Select the project tree line, and then the desired characteristics  in the toolbar **Work modes**. The **Info** window will open.
2	If you want to display in the **Info** window meaning only that feature that is currently selected in the project tree, click  in the **Info**. Button takes the form  .

Close all windows «Info»,which displays the parameters of performance

Step	Actions
1	Click  in the toolbar **Work modes**.

Setting conservation characteristic component in the file during the calculation of the project

Step	Actions
1	Scroll to the tab row **Postprocessor** desired characteristics and in the **Properties** window, specify the save settings to a file.

Step	Actions
2	Click the **Apply** button.

8.2.3.8 Operations with user variables

As the number of operations with user variables are:

- the creation of a custom variable
- Copy the custom variable
- Remove custom variable
- Editing custom variable

Creating a user variable

Step	Actions
1	Right-click the folder line **User variables > Local (Global, Constant)** and select **Create**.
2	New elements of the **User variable**is displayed in the project tree in the folder **User variables > Local (Global, Constant**).

Copying a user variable

Step	Actions
1	Open the context menu of the element of the **User variable** to be backed up, and select **Copy**.
2	New elements of the user variable is displayed in the project tree in the folder **User variables > Local (Global, Constant**). Element parameters match those of the copied custom variable.

Deleting a user variable

Step	Actions
1	If the deleted user variable is included in the other elements previously been excluded from all the elements.
2	Open the context menu of the element of the **user variable** to be deleted and select **Delete.** Elements of the **user variable** is removed from the project tree of folders **User variables > Local (Global, Constant**).

Change the settings for a user variable

Step	Actions
1	Select the item in the Project Tree **Custom Variable** to be edited.
2	Adjust the values of the parameters in the**Properties**window, and then click **Apply**. Parameter **Value** and **Value > X (Y, Z)** are shown in the extended entry fields. Their values can be specified as constants, functions in the form of tables or formulas using natural, integrated and user-defined variables.

8.2.3.9 Operations with computational Subregions and elements in them

See below step-by-step descriptions of operations with the computational **Subregions** and items in them:

- Operations with geometry model of computational Subregion
- Operations with surfaces
- Operations with boundary conditions
- Operations with groups of facets
- Operations with Modifiers (including operations with **Moving bodies**)

- Operations with initial conditions

See also: the section Operations with the geometry model of the computational domain.

8.2.3.9.1 Operations with geometric model of computational Subregion

As the number of transactions, with a computational subregion includes the following:

- specify the models in computational subregion
- regrouping of the geometry model of computational subregion (and moving bodies)
- transformation of a geometry model of a subregion (and moving bodies); conversion element **subregion** is performed in absolute coordinate system. Converting a **subregion** can be carried out, with or without including the surfaces of the moving bodies into regrouped surfaces. Regrouping mode determined by the choice context menu item **Transform geometry** or **geometry + moving body**
- setting boundary conditions on the boundary of the computational subregion
- Export border computational subregion file format *WRML* (`*.wrl`), *VTK* (`*.vtk`), *3DVision* (`*.mesh`).

Specifying models in computational subregion:

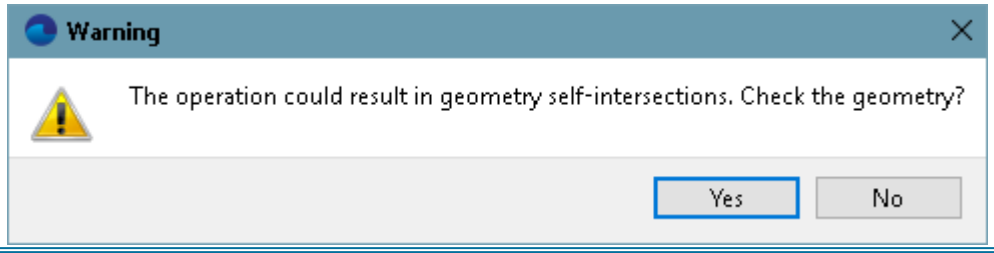
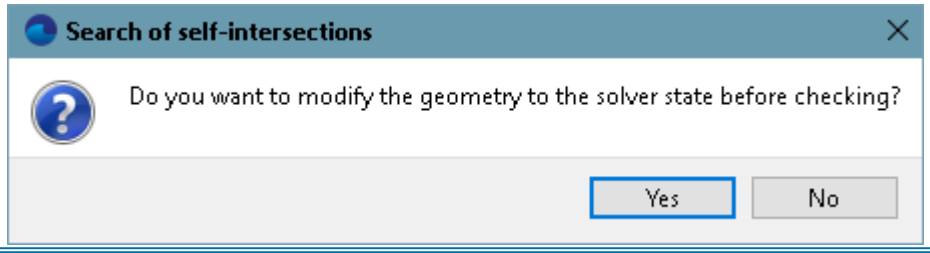
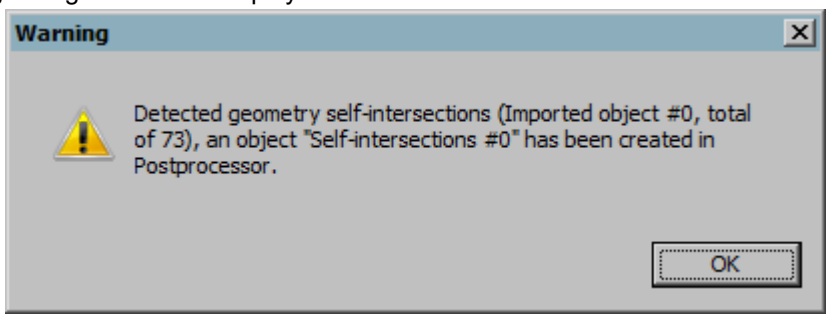
Step	Actions
1	Select the item in the project tree **subregion**.
2	In the **Properties** window, select **Model #N** for the **model** parameters, and then click **Apply**.

Regrouping of the geometry model computational subregion (and moving bodies):

Step	Actions
1	Open the context menu of the element **subregion** and select: • **Regroup geometry** • *or* **Regroup geometry + moving body** A dialog box **Geometry regrouping** will open.
2	Specify a criterion for grouping - parameter **deflection angle (in degrees)** and remove / mark set in the input field **not break the balance of GIs at the** triangles.
3	Click the **Browse** button. The results of the group are displayed in the table on the panel **statistics after regrouping**.
4	After reaching the desired result lock it by pressing the **Apply** button.

The transformation of a geometry model subregion (and moving bodies):

Step	Actions
1	Right-click the row of the **Sub-region** and select: • **Transform geometry** • *or* **Transform geometry + moving body** Once clicked, the **Geometry transformation** dialog box will open.
2	Set the parameters of transformation subregion and then click **Apply** to preview in the **View** window alleged transformation. If the results do not satisfy the transformation, change the transformation parameters and click **Apply** again. Repeat this step until you get the desired results.
3	Press **OK**, to actually carry out the transformation geometry in accordance with the entered data. Window opens to request checks on self-intersection (**"The operation could result in geometry self-intersections. Check the geometry?"**):

Step	Actions
	
4	Click **Yes**. This opens the query window (**"Search of self-intersections"** / **"Do you want to modify the geometry to the solver state before checking?"**): 
5	If the initial grid has already been built, click **Yes**. If the initial grid has not built yet, click the **No**.
6	The program will search for self-intersections. Upon detection of self-intersections of the system performs the following actions: • folder **3D-scene > Objects > Computational space** created imported to **Self-intersections of #0**; this property consists of intersecting facets and created as an auxiliary to analyze the reasons that led to an error in setting the computational domain; after analysis of the object should be removed; • in the project window tab opens **Postprocessor** • a warning dialog box will be displayed:  Click **OK**.
7	To remedy the resulting self-intersections should change the conversion boundary computational subregion (and moving bodies) and delete the imported object **Self-intersections #0**, which is displayed in the folder **3D-scene > Objects > Computational space**.
8	If you need to perform several transformations geometry (for example, several consecutive rotations), repeat the previous steps as many times.

Setting the boundary conditions on the boundary of the computational subregions:

Step	Actions
1	Open the context menu of the element SubRegion #N and select the **Set boundary condition > B. Cond. #N** command.
2	Selected boundary condition is set on the entire boundary of **SubRegion #N**.

Export to file border computational subregion:

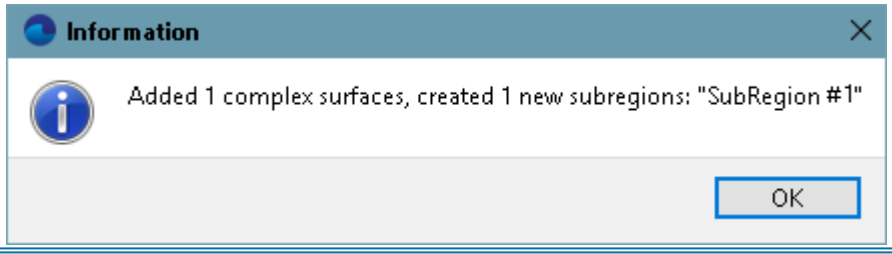
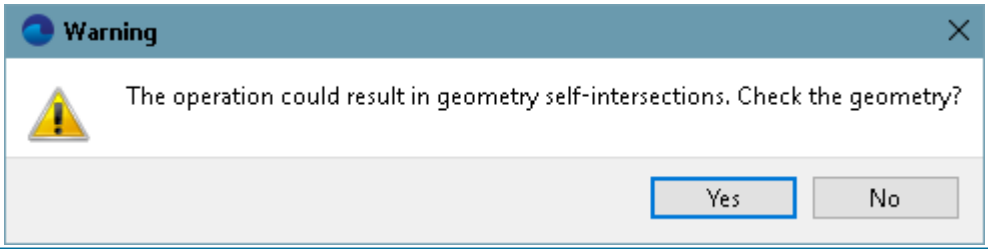
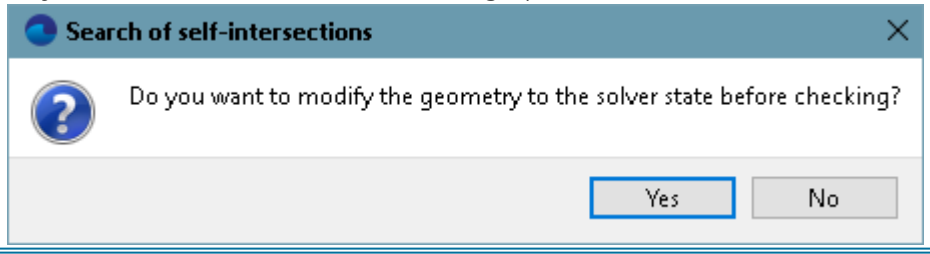
Step	Actions
1	Open the context menu of the element SubRegion #N and select the **Export geometry** command. Once clicked, a standard dialog box **Export geometry** opens.
2	Select the file type, then select the file name and click **Save**.

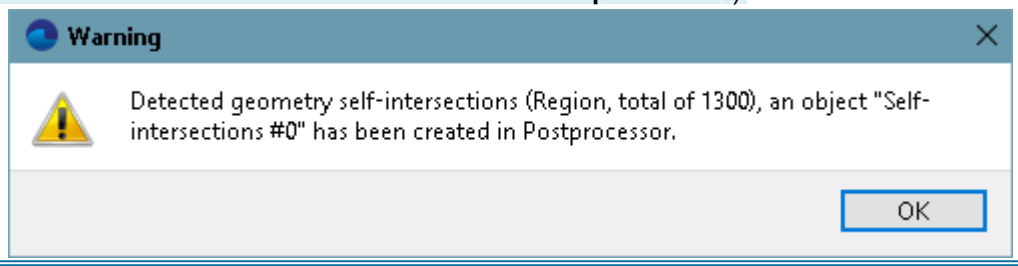
8.2.3.9.2 Operations with surfaces

Some of the operations comprises the following surfaces:

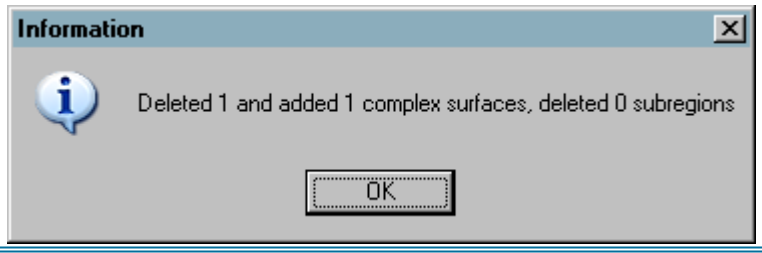
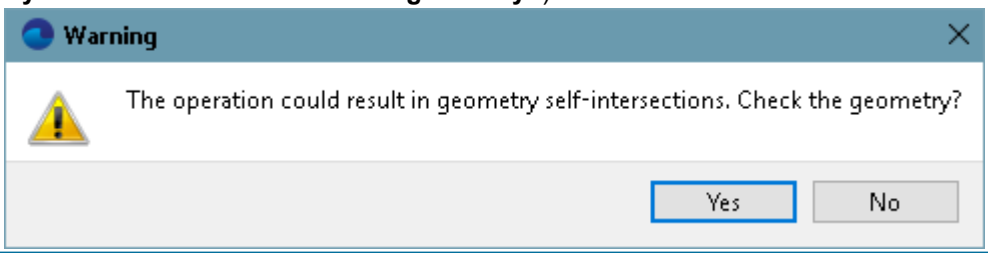
- adding a surface
- exchange surface
- removal of the surface of a geometry model of the computational domain
- regrouping a surface
- transformation (conversion) of the surface
- export surface; export geometry model of the computational domain is performed to the file format *WRML* (`*.wrl`), *VTK* (`*.vtk`), *3DVision* (`*.mesh`)
- setting boundary conditions on the surface

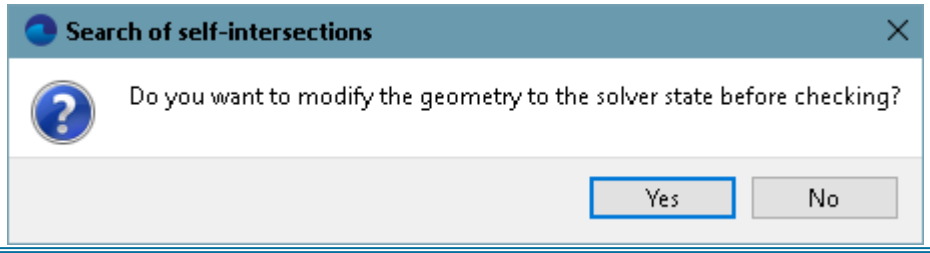
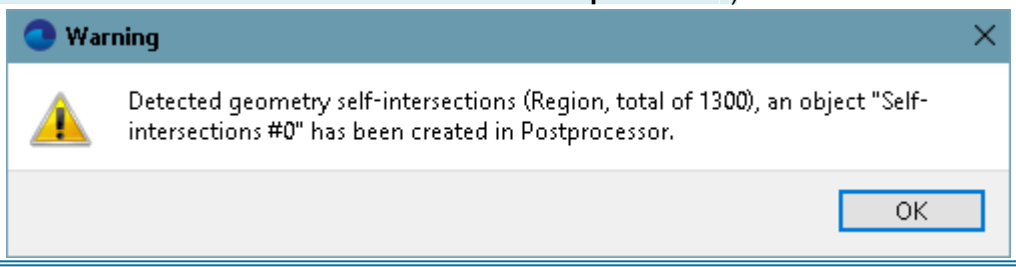
To add a surface:

Step	Actions
1	Open the context menu of the root folder Region and select **Add geometry**.
2	From a standard drop down dialog box select a file, from which the surfaces will be imported, and click **Open**. Displays a message box with the message (**"Added ... complex surfaces, created ... new subregions: "SubRegion #..."**): 
3	Click **OK**. Displaying the query window check the self-intersection (**"The operation could result in geometry self-intersections. Check the geometry?"**): 
4	Click **Yes**. Displaying the query window (**"Search of self-intersections" / "Do you want to modify the geometry to the solver state before checking?"**): 

Step	Actions
5	If the initial grid has already been built, click **Yes**. If the initial grid has not built yet, click **No**.
6	The system searches for self-intersections. Upon detection of self-intersections of the system performs the following actions: • folder **3D-scene> Objects> Computational space** created imported to **Self-intersections of # 0**;this property consists of intersecting facets and created as an auxiliary to analyze the reasons that led to an error in setting the computational domain; after analysis of the object to be removed • in the **Project** tab displays **Postprocessor** • A warning dialog box is displayed (**"Detected geometry self-intersections (...), an object "Self-intersections #N" has been created in Postprocessor."**): 
7	Click **OK**.
8	To correct for the error should be added to remove the surface of the computational domain and imported to **Self-intersections #0** is displayed in the folder **3D-scene> Objects> Computational space**.

To replace a surface:

Step	Actions
1	Right-click the line item **Region- Surface #N** and select **Replace surface**.
2	In the **Open** dialog box,select the file with the geometry model of the computational domain, and then click **Open**. In the **View** window displays the loaded geometry model of the computational domain. An informational dialog box appears (**"Deleted ... and added ... complex surfaces, deleted ... subregions"**): 
3	Click **OK**. Displaying the query window check the self-intersection (**The operation could result in geometry self-intersections. Check the geometry?**): 
4	Click **Yes**. Displaying the query window (**"Search of self-intersections" / "Do you want to modify the geometry to the solver state before checking?"**):

Step	Actions
	
5	If the initial grid has already been built, click **Yes**. If the initial grid has not built yet, click **No**.
6	The system searches for self-intersections. Upon detection of self-intersections of the system performs the following actions: • folder **3D-scene> Objects> Computational space** created imported to **Self-intersections #0**;this property consists of intersecting facets and created as an auxiliary to analyze the reasons that led to an error in setting the computational domain; after analysis of the object to be removed • in the Project tab displays **Postprocessor** • A warning dialog box is displayed (**"Detected geometry self-intersections (...), an object "Self-intersections #N" has been created in Postprocessor."**): 
7	Click **OK**.
8	To correct for the error should be added to remove the surface of the computational domain and imported to **Self-intersections #0** is displayed in the folder **3D-scene> Objects > Computational space**.

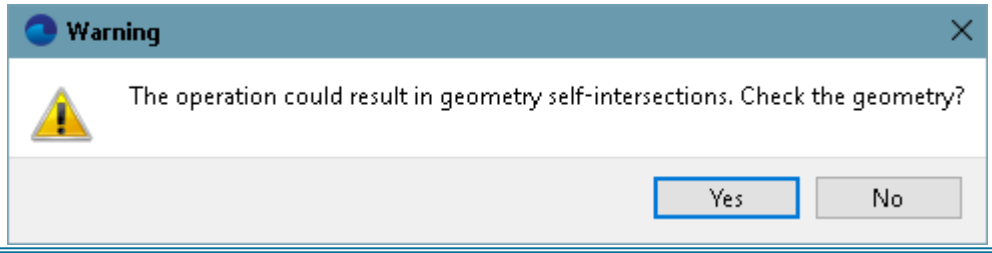
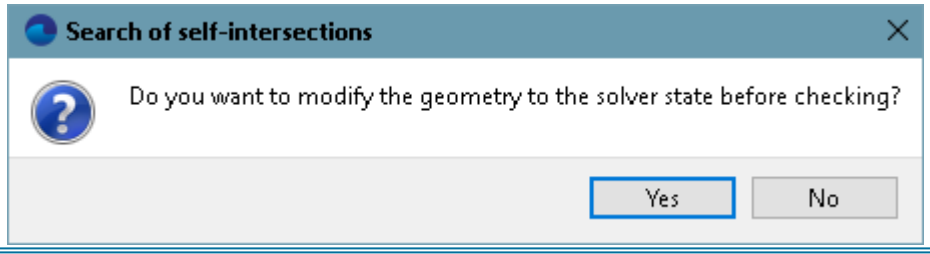
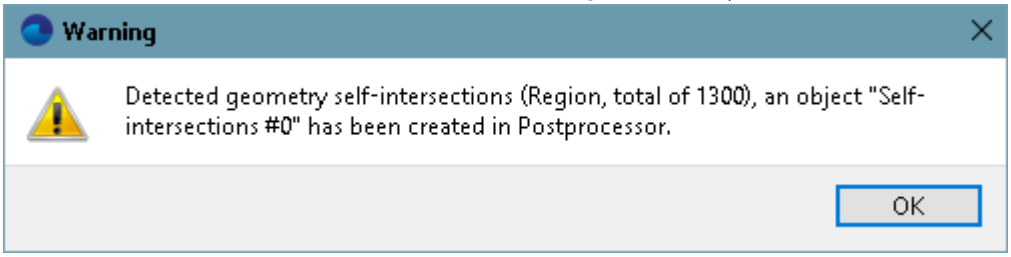
To regroup a surface:

Step	Actions
1	Right-click the row of the **Region- Surface #N** and select the **Regroup surface**. A dialog box **Geometry regrouping** will open.
2	Specify a criterion for grouping - parameter **Threshold angle (in degrees)** and unselect or keep selected the checkbox **Prevent changing boundary conditions on triangles**.
3	Click the **Preview** button. The results of the group are shown in the tables in the **Final statistics** window.
4	After reaching the desired result lock it by pressing the **Apply** button.

To transform a surface:

Converting a **Region- Surface #N** performed in absolute coordinate system.

Step	Actions
1	Right-click the row of the **Region- Surface #N** and select **Transform surface**. Once clicked, the Geometry transformation dialog box will open.

Step	Actions
	Transformation geometry is made by pressing the**OK** button in this window (the **Geometry transformation** dialog box will close). Clicking the **Cancel** button or icon **"x"** in the upper right corner to close the **Geometry transformation** dialog box without performing transformation of the geometry. Clicking on the button **Apply** to see the result of transformation to the actual implementation of the transformation (i.e.,the **Apply** button performs the preview). Individual actions to be taken in the course of the transformation, made in a manner consistent with the location of parameters in the **Geometry transformation** dialog box: 1. Determined by the operations center 2. Is scaled 3. Produced rotation 4. The shift is
2	Specify conversion options and click **Apply** to preview the transformation in the **View**. If necessary, repeat this step by including other data, until you get satisfied with the results.
3	Click **OK**, to actually carry out the transformation geometry in accordance with the entered data. The **Geometry transformation** dialog box will close at the same time. This opens a window in which the program will prompt you to check for self-intersections: 
4	Click **Yes**. Displaying the query window: 
5	If the initial grid has already been built, click **Yes**. If the initial grid has not built yet, click **No**.
6	The system searches for self-intersections. Upon detection of self-intersections of the system performs the following actions: • folder **3D-scene> Objects> Computational space** created imported to **Self-intersections #0**; this property consists of intersecting facets and created as an auxiliary to analyze the reasons that led to an error in setting the computational domain; after analysis of the object should be removed; • in the Project tab displays **Postprocessor**; • A warning dialog box is displayed (**"Detected geometry self-intersections (...), an object "Self-intersections #N" has been created in Postprocessor."**):  Click **OK**.

Step	Actions
7	To remedy the resulting self-intersections should change the conversion surface and remove the imported object **Self-intersections #0** is displayed in the folder **3D-scene> Objects> Computational space**.
8	If you need to perform several transformations geometry (eg, several consecutive turns), repeat the previous steps as many times.

To export a surface to a file:

Step	Actions
1	Open the context menu of the element **Region- Surface #N** and choose **Export surface**. Once clicked, a standard dialog box **Export geometry**.
2	Select the file type, select the file name and click **Save**.

To set the boundary conditions on a surface:

Step	Actions
1	In the context menu element **Region- Surface #N** and use the command **Set boundary condition > B. Cond. #N**.
2	Selected boundary condition **B. Cond #N** set on the entire surface **Region- Surface #N**.

8.2.3.9.3 Operations with boundary conditions

As the number of operations with the boundary conditions are the following:

- establishment of boundary conditions;
- up the boundary conditions;
- removal of boundary conditions;
- editing parameters of the boundary conditions;
- task group color facets with the given boundary conditions;
- setting boundary conditions on the boundary of the computational subregion on the surface or a group of facets
- regrouping of facets, which put certain boundary conditions (and moving bodies)
- assignment of material surfaces of facets, which put certain boundary conditions

Create boundary conditions:

Step	Actions
1	Right-click the folder line **Subregions > SubRegion #N > Boundary conditions**. From the menu screen, click **Create**.
2	New item **B. Cond #N** will appear in the project tree in the folder **Subregions > SubRegion #N > Boundary conditions**.

Copying boundary conditions:

Step	Actions
1	Open the context menu of the element **Boundary** conditions, be backed up.
2	Click **Copy**. New item **B. Cond #N** displayed in the project tree in the folder **Subregions > SubRegion #N > Boundary conditions**. Element parameters match those of the copied item **Boundary conditions**.

Removal of boundary conditions:

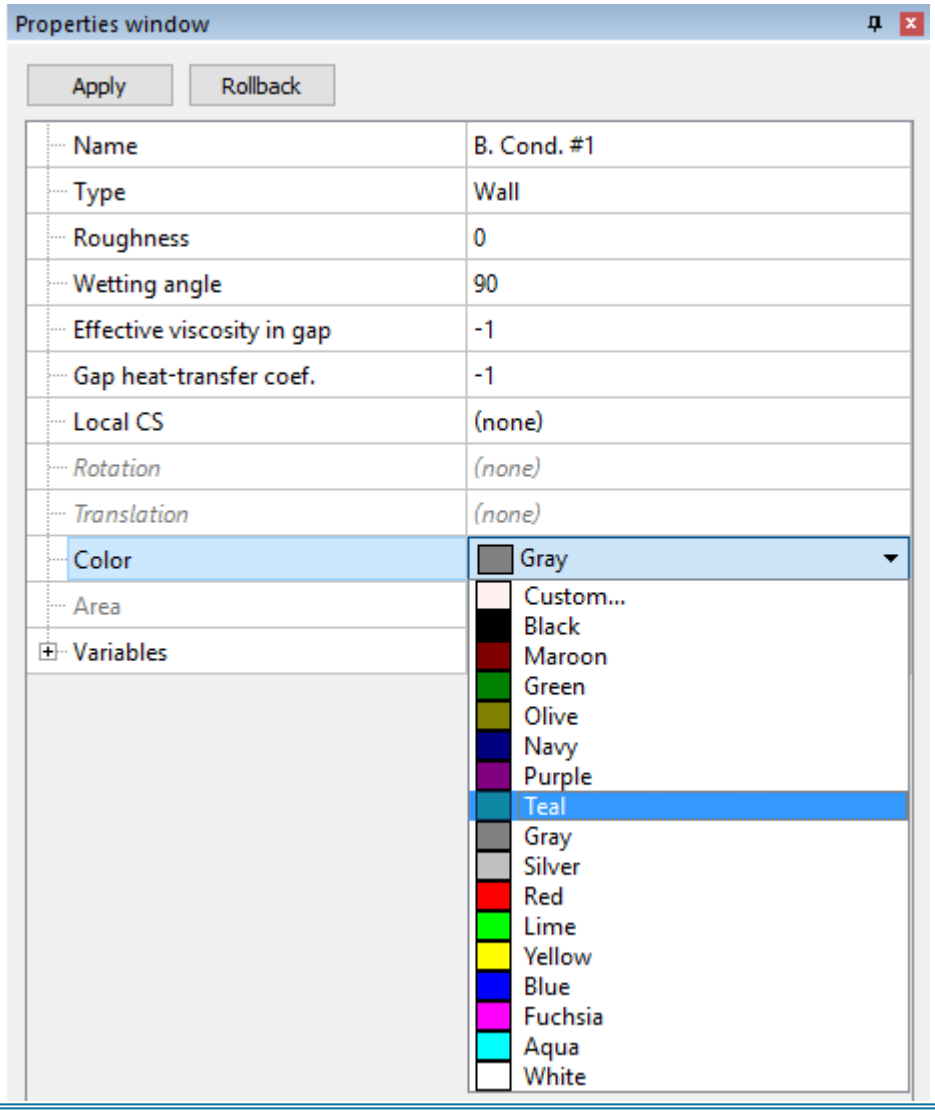
Step	Actions
1	If the deleted item **B. Cond #N** included in other elements of the pre-exclude it from all the elements.
2	Open the context menu of the element **Boundary conditions**, to erase, and select **Delete**. Item **B. Cond #N** removed from the project tree.

Changing the parameters of the boundary conditions:

Step	Actions
1	Select the item in the project tree **Boundary conditions**, to edit.
2	Adjust the values of the parameters in the **Properties** window, and then click **Apply**.

Specifies the color of a group of facets with the given boundary conditions:

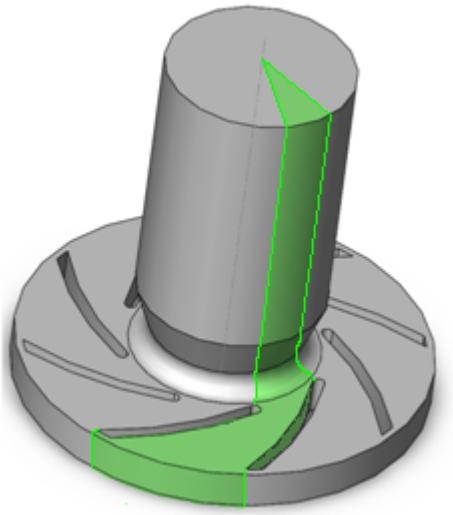
Step	Actions
1	Open the context menu of the **Boundary conditions** element and then open the **Set color** submenu. In the color menu you can select one of the basic colors or any color. To specify an arbitrary color: a) select **Custom**, then a standard dialog box of the operating system for selecting a color will open. b) select a color and click **OK**.
2	Color facets with the selected boundary condition is changed to the specified.
•	*Alternative method:* set a color as value of the **Color** parameter in properties of the **Boundary condition**:

Step	Actions
	

Regrouping facets, which put certain boundary conditions (basic geometry and moving bodies):

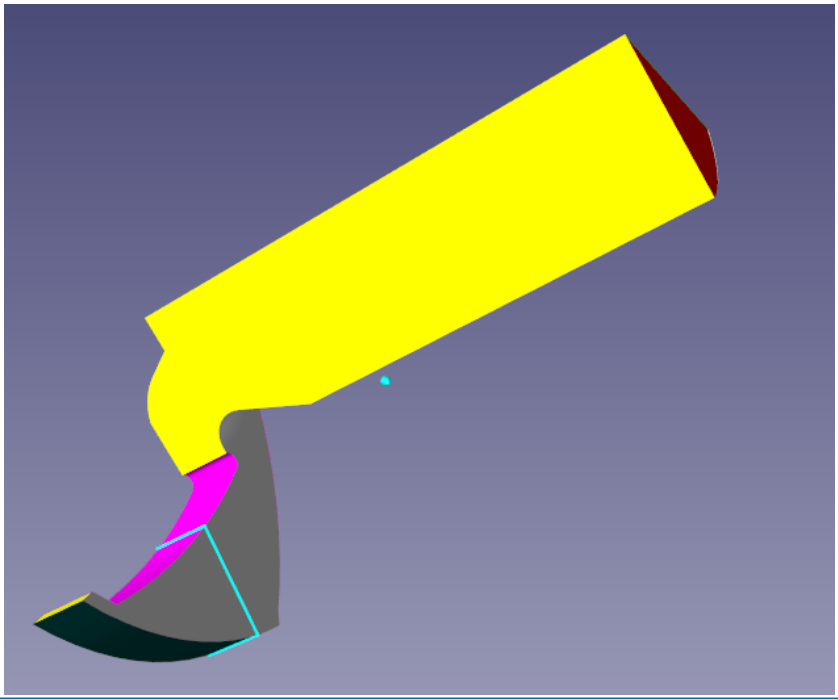
Step	Actions
1	Open the context menu of the element**Boundary conditions** and select the command: • **Regroup geometry** • *or* **Regroup geometry + moving body** The **Geometry regrouping** window will open.
2	Specify a criterion for grouping - parameter **Threshold angle (in degrees)** and unselect or keep selected the **Prevent changing boundary conditions on triangles** checkbox.
3	Click the **Preview** button. The results of the group are displayed in the table on the panel **Final statistics**.
4	After reaching the desired result lock it by pressing the **Apply** button.

Specifying and using periodic surfaces

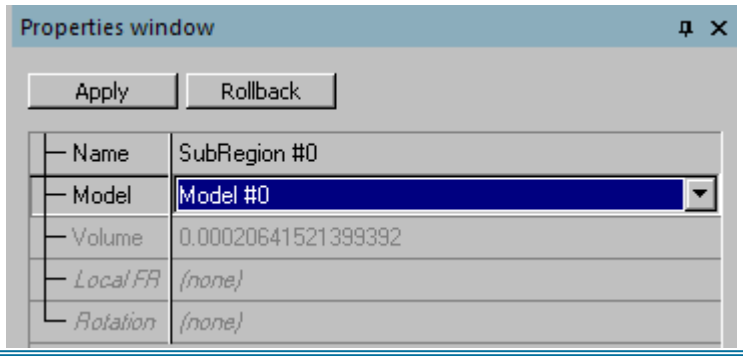
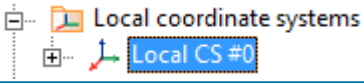
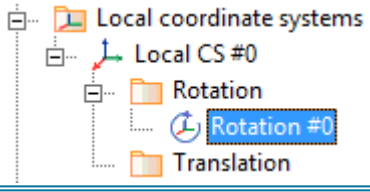


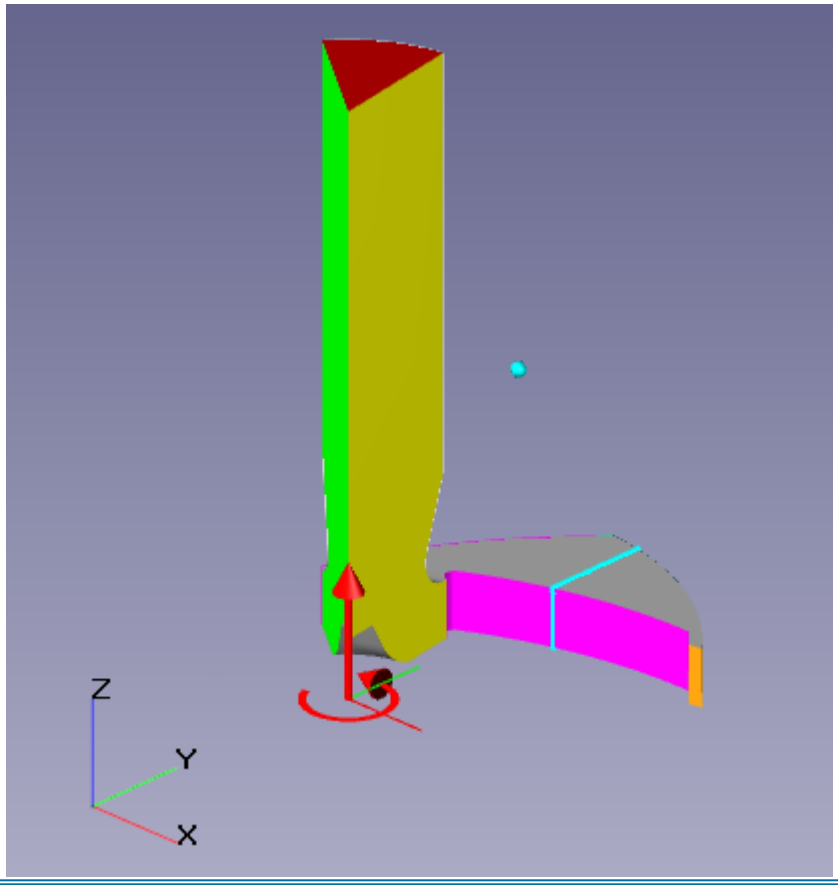
Periodic surfaces, see the section Periodic surface.

Below is an example of setting a **Periodic surface**. Follow the steps listed in the table below:

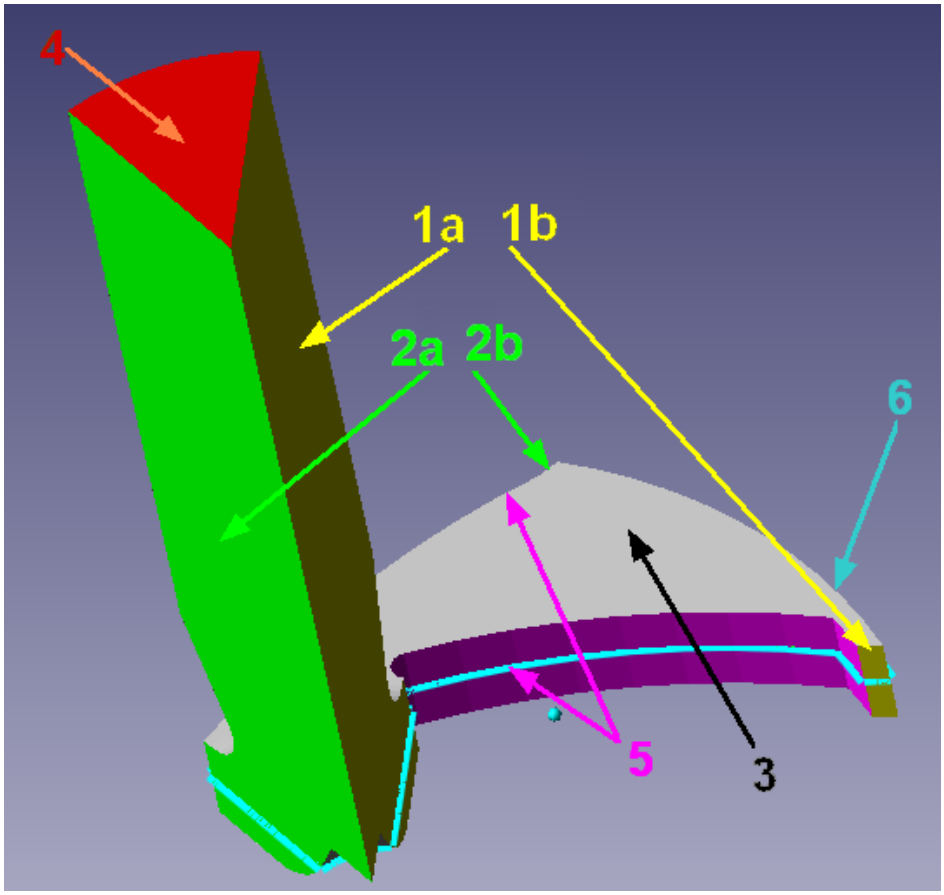
Step	Actions
1	Download the geometry model of the pump from the file `RotorSector.wrl`, save the project in a folder. The geometry model will be displayed in the **View** window: 

2	Enter the properties of the physical model:
	• in folder **Substances** create **Substance #0** and load it from the standard base of substances the substance **Air** in the phase **Gas (equilibrium)**
	• in folder **Phases** create a continuous phase **Phase #0** and add to it the substance **Air_Gas (equilibrium)**
	• specify parameters in properties of the folder **Phases > Phase #0 > Physical processes**:
	  ○ **Motion = Navier-Stokes model**
	  ○ **Turbulence = KES**
	• in folder **Models** create **Model #0** and add to it **Phase #0**
	• in properties of child elements of the subfolder **Models > Model #0 > Init. data > Init. data #0** specify:

Step	Actions
	○ for **Pulsations (Phase #0)** specify **Value=0.01** ○ for **Turbulent scale (Phase #0)** specify **Value=0.01**
3	For **Subregion #0** specify in its **Properties** window: • **Model = Model #0** 
4	Create a new local coordinate system for movements. To do this, in the project tree, in **Preprocessor**, from the context menu of the folder **Local coordinate systems** select **Create**. In the project tree will be a new local coordinate system for movements will appear named as **Local CS #0**: 
5	From the context menu of the just created LCS **Local CS #0** select the command **Add rotation**. As a result, in the subfolder **Rotation** the **Rotation #0** element will be created: 
6	Change some parameters of **Rotation #0** in its **Properties** window: • **Speed = 50** (indicated in [rad/s]) • **Direction > X = 0** • **Direction > Y = 0** • **Direction > Z = 1** Click the **Apply** button. The axis and the direction of **Rotation #0** will be displayed in the **View** window as two arrows:

Step	Actions
	
7	Set the boundary conditions on the surfaces of the rotor fragment (in the project tree, they correspond to a sub-directory **Subregions > SubRegion #0** in **Preprocessor**). Some of them will be *periodic* - they will connect edges fragment corresponding edges of adjacent identical fragments.

Step	Actions



Boundary conditions for a portion of the **Rotor**:
1a and 1b - back surface periodic (consists of two fragments),
2a and 2b - a periodic front surface (also consists of two fragments),
3 - the surface of the blades and for the blades,
4 - the air inlet, 5 - blade, 6 - outlet for air

If periodic surface of each of the binding pair consists of several unrelated fragments, it is necessary to specify certain boundary conditions in each of these fragments.

In the example in Figure 4 above should create the boundary conditions for surfaces 1a, 1b, 2a and 2b. Then, the boundary conditions are matched in pairs:

- 1a and 2a
- 1b and 2b

If you set the boundary conditions for uniform surfaces (1a and 1b) and (2a and 2b), and then link them, the calculation of the periodic surface will be incorrect.

To *the rear* of the two *fragments of the periodic surface* (1a and 1b), specify two boundary conditions, each of which:

- **Type = Connected**

To *the front* of the two *fragments of the periodic surface* (2a and 2b), specify two boundary conditions, each of which:

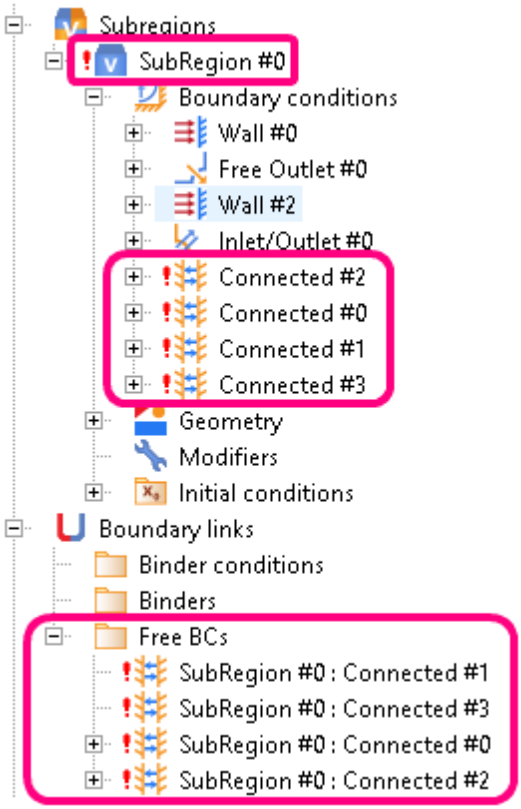
- **Type = Connected**

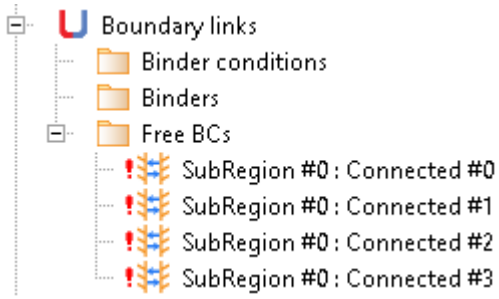
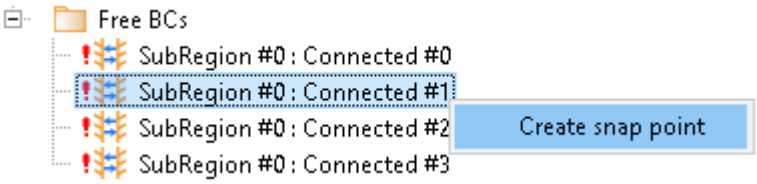
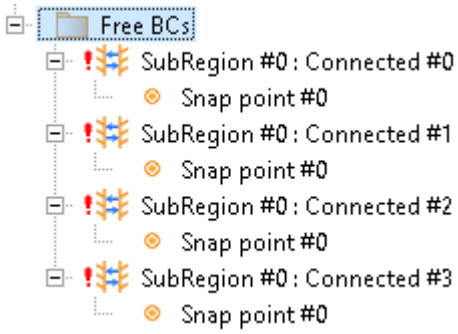
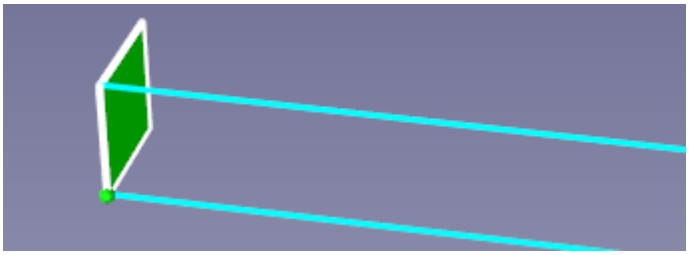
For *surfaces below and above the blades* (3) specify:

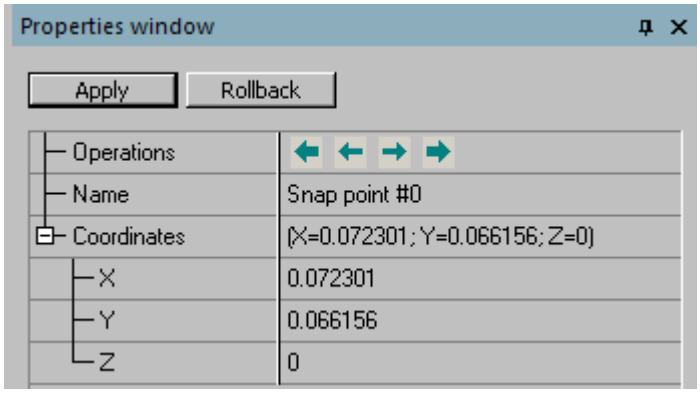
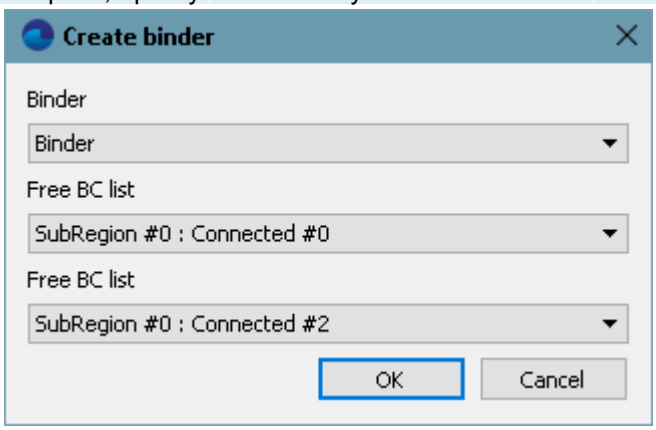
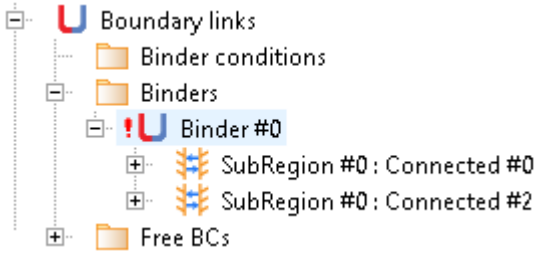
- **Type =Wall**
- **Variables > Velocity (Phase #0) = Logarithm law**
- **Variables > TurbEnergy (Phase #0) = Value in cell near wall**
- **Variables > TurbDissipation (Phase #0) = Value in cell near wall**

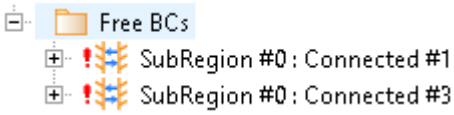
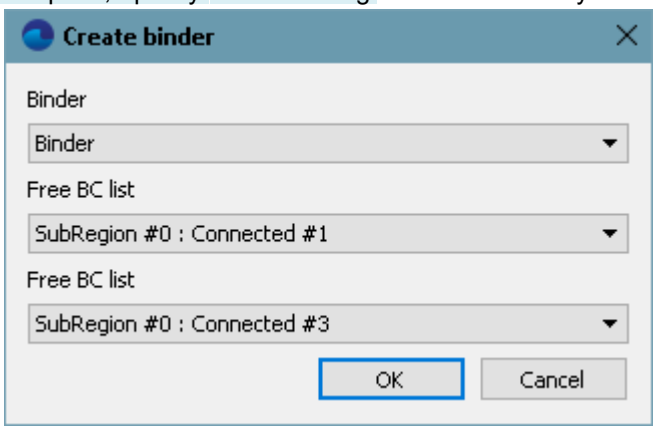
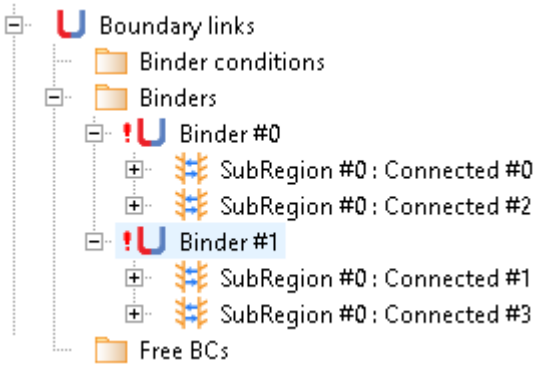
Air inlet(4) specify:

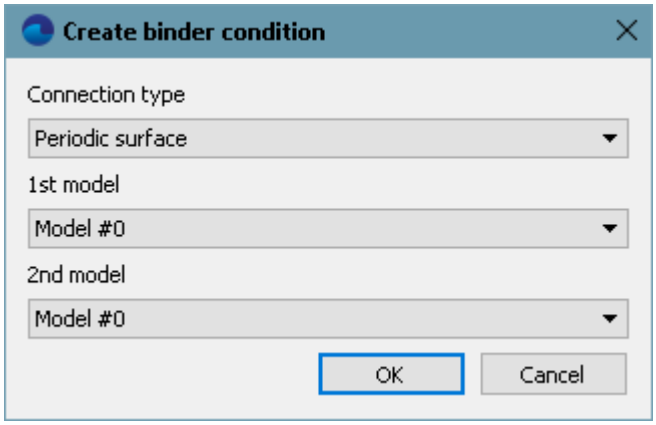
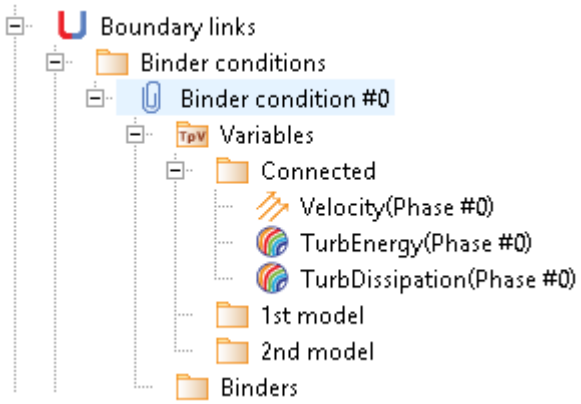
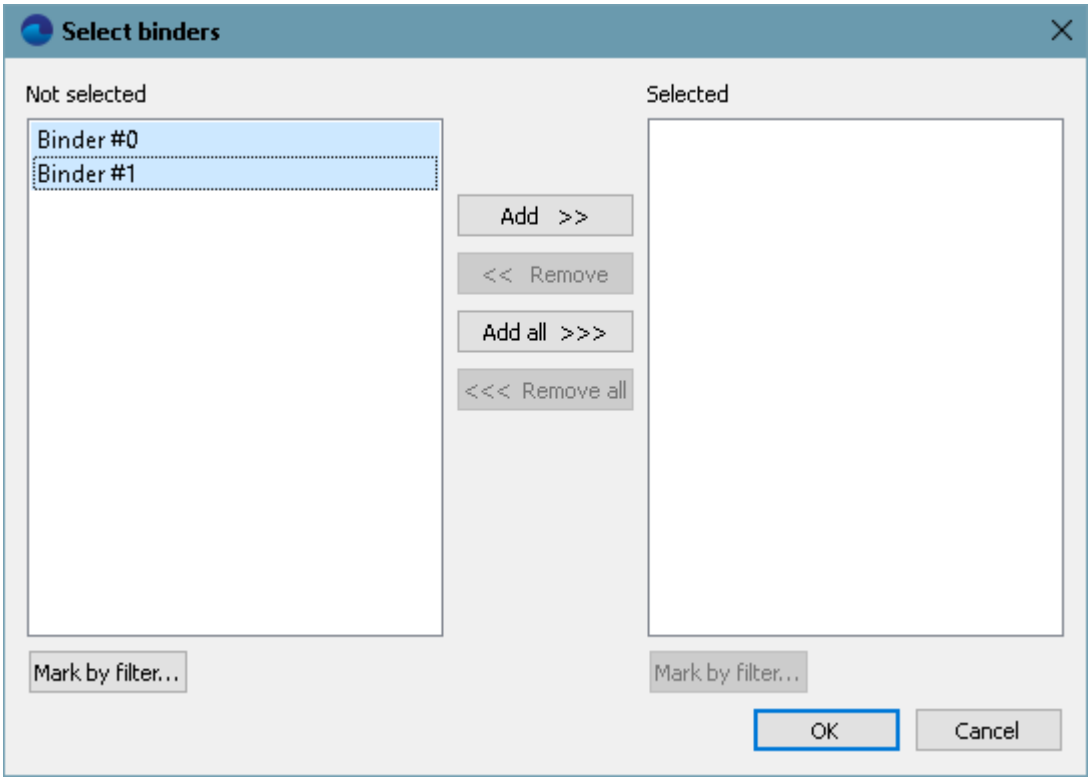
- **Type =Inlet / Outlet**
- **Variables > Velocity (Phase #0) = Normal mass velocity** and set **Mass velocity = 20** in the properties of the child element **Velocity (Phase #0)**

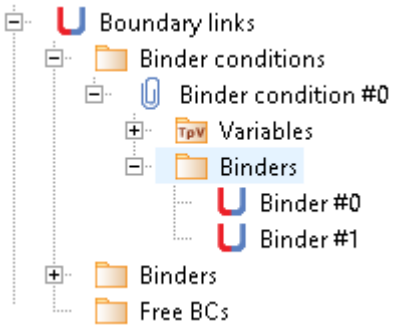
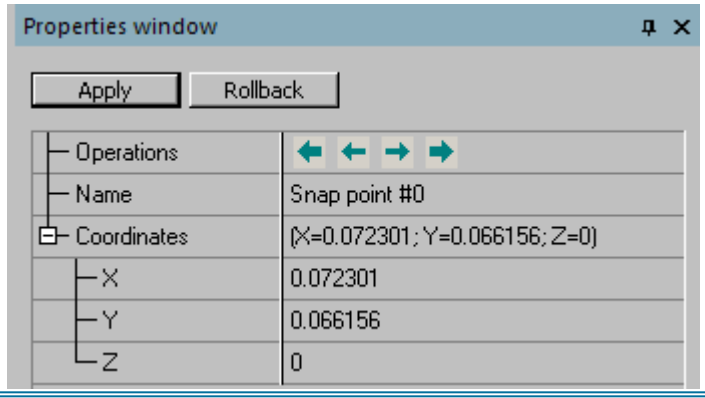
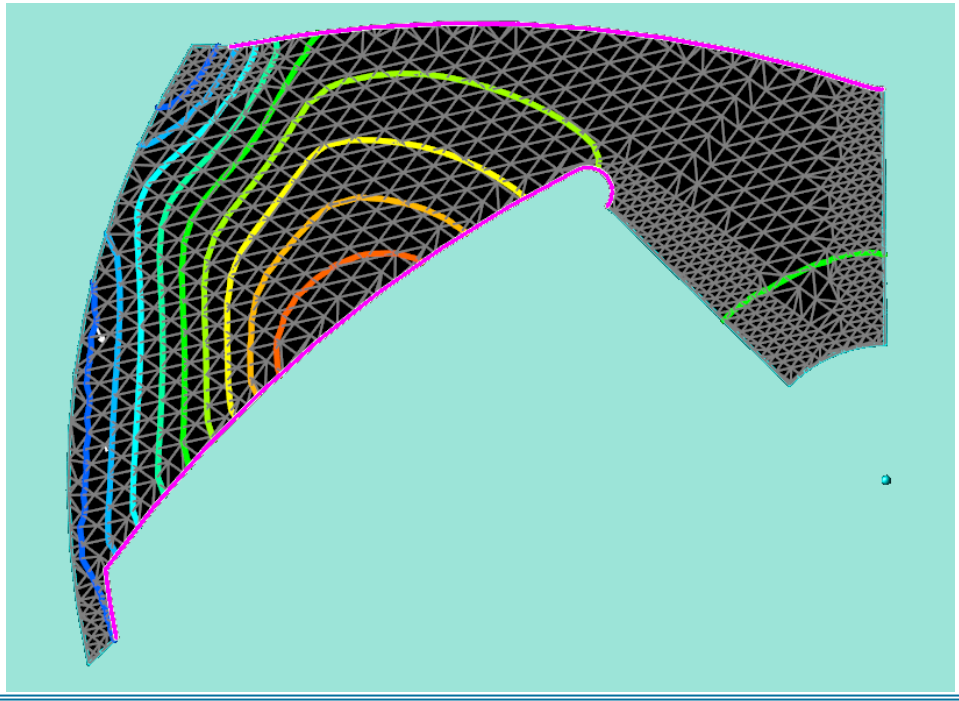
Step	Actions
	• **Variables > TurbEnergy (Phase #0) = Pulsations** and set **Value = 0.01** in the properties of the child element **TurbEnergy (Phase #0)** • **Variables > TurbDissipation (Phase #0) = scale turbulence** and set **Value = 0.01** in the properties of the child element **TurbDissipation (Phase #0)** For *the rotor blades* (5) specify: • **Type =Wall** • **Local CS = Local CS #0** • **Rotation = Rotation #0** • **Variables > Velocity (Phase #0) = Logarithm law** • **Variables > TurbEnergy (Phase #0) = Value in cell near wall** • **Variables > TurbDissipation (Phase #0)= Value in cell near wall** To *vent* (6) specify: • **Type = Free Outlet** • **Variables > Velocity (Phase #0) = Pressure** and set **Value = 0** in the properties of the child element **Velocity (Phase #0)** • **Variables > TurbEnergy (Phase #0) = Pulsations** and set **Value = 0.01** in the properties of the child element **TurbEnergy (Phase #0)** • **Variables > TurbDissipation (Phase #0) = scale turbulence** and set **Value = 0.01** in the properties of the child element **TurbDissipation (Phase #0)** Boundary conditions, with the type of **Bound** and **SubRegion #0** itself will be marked in the project tree with an exclamation point symbol ("**!**"), since these boundary conditions have not yet been compared with each other. Also, these boundary conditions will fall as the elements in the project tree in the folder **Boundary links > Free BCs**:  *(Note: the names of the boundary conditions in the project tree does not refer to the illustration in Step 7)*
8	In this and the next few steps described pairwise binding periodic surfaces. Until now, they have not yet linked and displayed in the project tree in a subfolder **Boundary links > Free BCs**:

Step	Actions
	 Should mutually bind the surface, indicated in our example as: • 1a and 2a • 1b and 2b This will require to place on these pairs of surfaces *control points* snap points.
9	Set the snap points on the bonding pairs of periodic surfaces contained in a subfolder **Boundary links > Free BCs**. To do this, apply on the relevant elements **Boundary links > Free BCs > SubRegion #N: *BC #M*** context menu command **Create snap point**:  Create a manner desired the required number of snap points that correspond to the project tree child elements **Boundary links > Free BCs > SubRegion #N: *BC #M* > Snap point #0**:  Snap points are displayed in the **View** window in the form of small balls whose color matches the color of periodic surfaces, for which they are given:  In the windows of the **Properties** of the elements of **Snap point #0**, if necessary, change the location of the snap points on the boundary condition by using the arrow keys (you can also specify the coordinates of the snap points in the fields **Coordinates > X**, **Coordinates > Y**, **Coordinates > Z**):

Step	Actions
	 Buttons and (thick arrows) moving the snap point along the contour of the periodic surface to the next point with a sharp break circuit. It is recommended to use these buttons when it allowed the contour shape of the periodic surface (if the circuit has a pronounced kink). Buttons and (thin arrows) moving the snap point along the contour of a periodic surface, placing it in the top of each polyhedron approximating curved contour (so these buttons move the snap point is relatively slow and does not identify the characteristic points of the contour, which increases the risk of error when setting the snap point). The choice of snap point must provide a valid comparison of periodic surfaces, corresponding to the physical meaning of the problem being solved (see subsection *"Snap points of periodic surfaces"* in the section Periodic surface). Also, if desired, in the **Properties** window, the snap point in the **Name** field, you can specify a meaningful name of the snap point instead of the default name **Snap point #0**. After setting the position of each snap point not forget to click in its **Properties** window on the **Apply** button.
10	From the context menu of the subfolder **Boundary links > Binders** select **New**, and, in the **Create binder** dialog box, which opens, specify two boundary conditions in fields **Free BC list**:  (If you choose any of the boundary conditions in one of the fields in the selection of another field is confined to the options in this example, the option - only). Click **OK**. In the project tree will be a bunch of **Binder #0** in a subfolder **Boundary links > Binders**:  The just created **Binder #0** is indicated in the project tree with a red exclamation point symbol ("**!**") because for the bonding boundary conditions **Binder condition** has not been set yet.

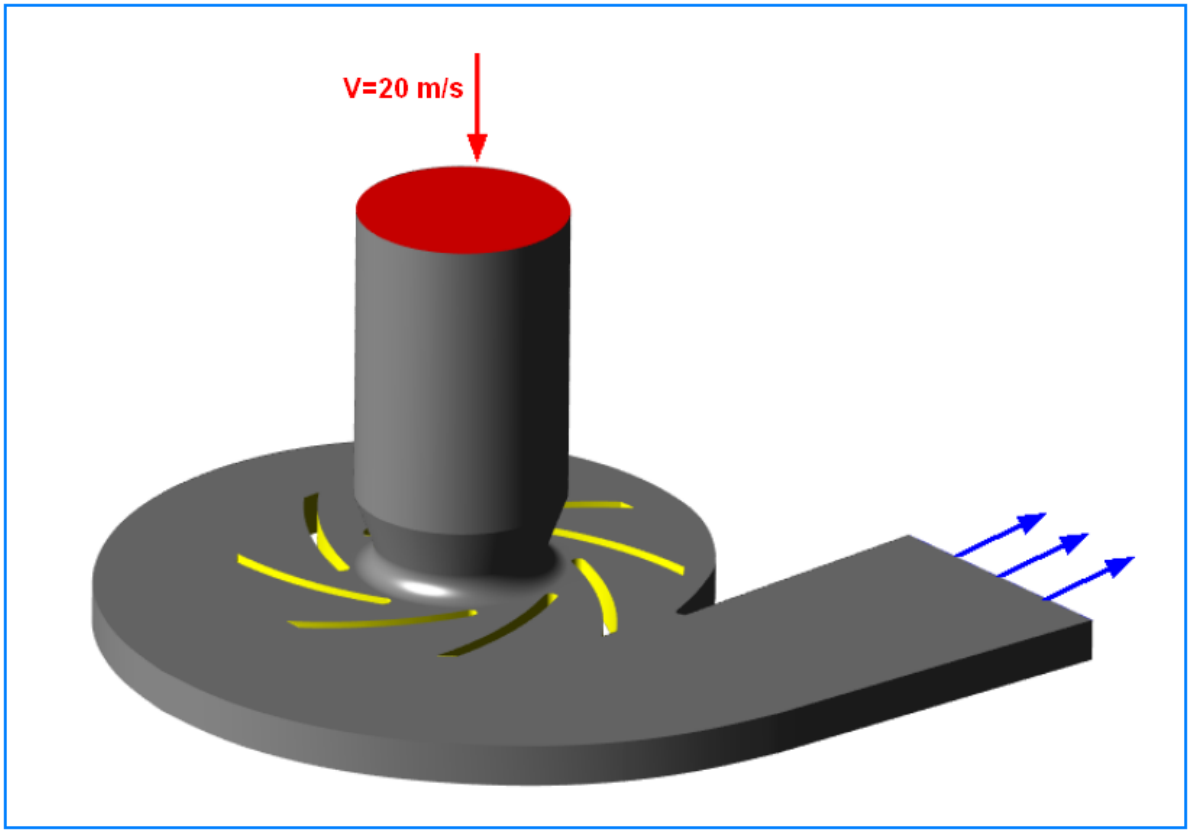
Step	Actions
	In the subfolder **Boundary links > Free BCs** remain two of the remaining unbound boundary conditions, which are also to be related:   *FlowVision* understands what surfaces can be related to each other, which prevents the wrong choice of the bonding surfaces and enables the automatic linking. Instead of pair bonding boundary conditions can perform automatic linking all combined pairs of boundary conditions are satisfied, select **Create** from the context menu of **all** subfolders **Boundary links > Binders** and skipping the next few steps. Binding of the boundary conditions in a manual mode for novice users to better understand the process of formation of pairs of periodic surfaces.
11	Bind the remaining two unbound boundary conditions. From the context menu of the subfolder **Boundary links > Binders** select **New**, and, in the **Create binder** dialog box, which opens, specify two remaining unbound boundary conditions:  Click **OK**. In the project tree will be a bunch of **Binder #1** in a subfolder **Boundary links > Binders**:  The just created **Binder #1** is also indicated in the project tree with a red exclamation point symbol ("**!**") because for the bonding boundary conditions have not yet set **Binder condition**. Subfolder **Boundary links > Free BCs** is empty.
12	Create a new **Binder condition**. To do this, from the context menu subfolders **Boundary links > Binder conditions** select **Create** and, in the form, which opens (**Create binder condition**), specify **Connection type = Periodic surface**. The fields **1st model** and **2nd model** (both their values are **Model #0**) are not to be changed.

Step	Actions
	 Click **OK**. In the project tree **Binder condition #0** will appear in the subfolder **Boundary links > Binder conditions**: 
13	Match **Binder condition #0** with bundles **Binder #0** and **Binder #1**. To do this in the context menu subfolders **Boundary links > Binder conditions > Binder condition #0 > Binders**, click **Add/remove**, and then, in the form that opens, select **Binder #0** and **Binder #1** in the panel **Not selected**, move them to the **Selected** pane, and click **OK**.  In the **Boundary links > Binder conditions > Binder condition #0 > Binders** subfolder, elements **Binder #0** and **Binder #1** will appear:

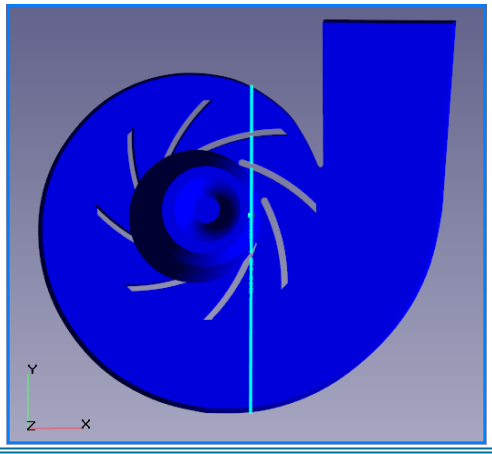
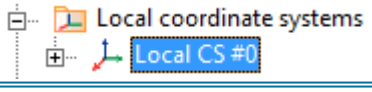
Step	Actions
	
14	You can change the location of the snap points on the pairs of periodic surfaces included in binders **Binder #0** and **Binder #1**. For this purpose, the window **Properties** of the corresponding snap points, as well as one to set the location of the snap points when they are created (see description in Step 9): 
15	Specify other project settings and can run the project on computation. *FlowVision* will calculate the result in the same calculated variables mutually corresponding points of periodic surfaces: 

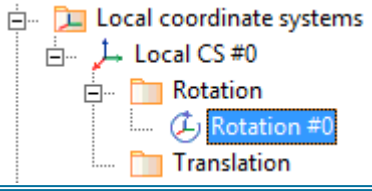
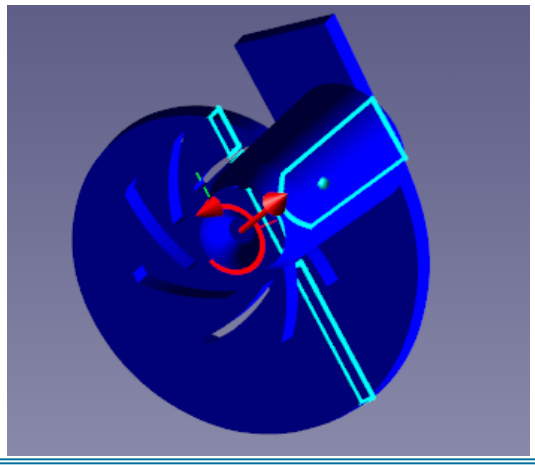
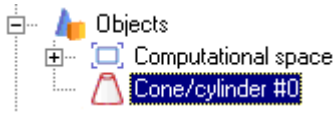
Setting and use sliding surfaces

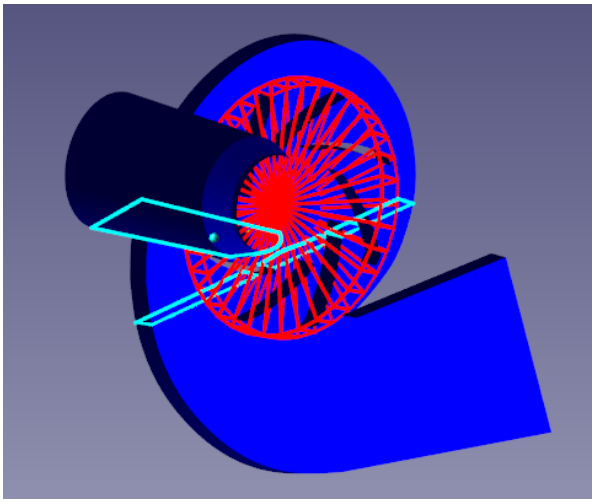
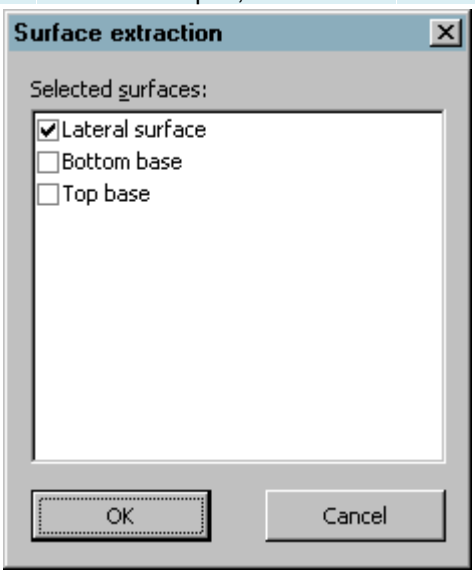
On the sliding surfaces, see the section Sliding surface.

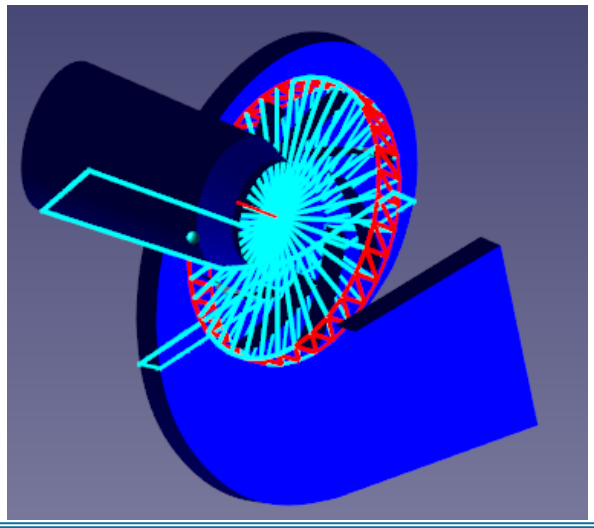
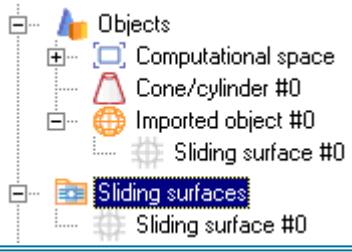
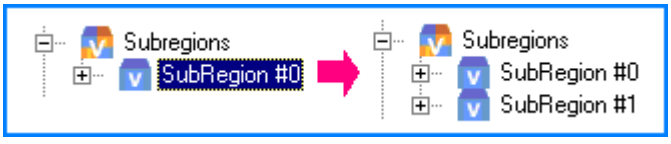
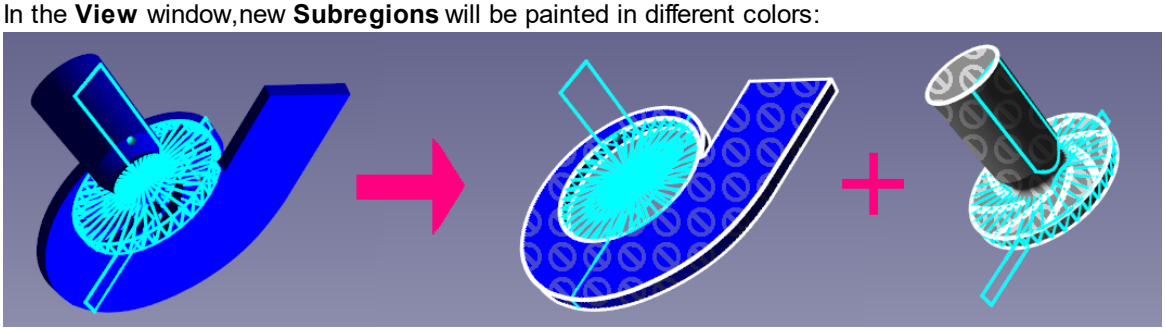


For training the practical skills of specifying a **Sliding** surface, which separates rotor's and stator's subregions a centrifugal pump, follow the steps listed in the table below:

Step	Actions
1	Download the geometry model of the pump from the file `RotorStator.STL`, supplied *FlowVision*, save the project in a folder. Geometry model is displayed in the **View** window: 
2	Create a new local coordinate system for the movement. To do this in the project tree in the **Preprocessor** in the context menu Folder **Local coordinate systems** use the command **New**. In the project tree will be a new local coordinate system for the movement called **Local CS #0**: 
3	In the context menu created LCS **Local CS #0** apply, click **Add rotation**. As a result, the element **Rotation #0** will be created in the **Rotation** subfolder of the LCS:

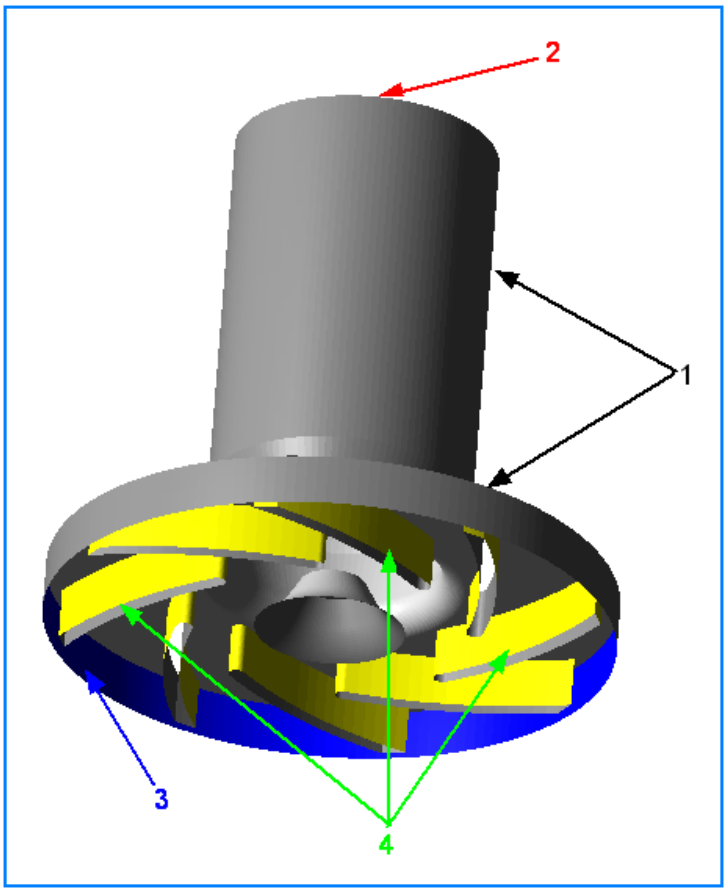
Step	Actions
	
4	Change some parameters **Rotation #0** in its **Properties** window: • **Speed = 50** (indicated in [rad/s]) • **Destination> X = 0** • **Destination> Y = 0** • **Destination> Z = 1** Click the button **Apply**. Axis and the direction of **Rotation #0** will be displayed in the **View** window as arrows: 
5	Create an axisymmetric geometric object from a side surface which is obtained **Sliding** surface. It will be a cylinder, a standard geometric object. To do this, the **Preprocessor** in the folder **Objects** apply in the context menu, click **New** (the type of the object, select a **Cone/cylinder**). A new standard geometric object **Cone/cylinder #0** will appear in **Objects**:  Change some parameters **Cone/cylinder #0** in its **Properties** window: • **Location > Axis X > X = 0** • **Location > Axis X > Y = 0** • **Location > Axis X > Z = 1** • **Location > Axis Y > X = 0** • **Location > Axis Y > Y = -1** • **Location > Axis Y > Z = 0**[*)] • **Parameters > Height = 0.02** • **Parameters > Radius 1 = 0.1025** • **Parameters > Radius 2 = 0.1025** • **Parameters > Base ratio = 1** Click the button **Apply**. Created a cylinder **Cone/cylinder #0** will be located around the rotor blades, which can be seen in the **View** window:

Step	Actions
	 *) Z axis direction will be automatically calculated and significance of X = 1, Y = 0, Z = 0
6	Create from the side surface of the **Cone/cylinder #0** an imported object to form a **Sliding surface**. To do this, from the context menu of **Cone/cylinder #0** select the command **Copy as imported object**. The **Surface extraction** window will open; select there **Lateral surface**:  As a result of the project tree in the **Objects** folder will be a new element called **Imported object #0**:

Step	Actions
	 
7	In the context menu of a folder **Sliding surfaces** click **New**, and in the form that appears, select the object **Imported object #0**. As a result, the folder **Sliding surfaces** and in the folder **Imported object #0** will elements **Sliding surface #0**. 
8	Divide the computational domain into **Subregions** by the dividing surface. To do this, you have first to specify a local coordinate system and a rotation in the properties of **Sliding surface #0** (if you do not specify them, then inserting of the **Sliding surface** into the **Subregion** is not allowed and the appropriate command is not active). • **Local CS = Local CS #0** • **Rotation = Rotation #0** Then, in the context menu **Sliding surface #0** apply **Insert**. As a result, **SubRegion #0** will be split into two sub-areas - **SubRegion #0** and **SubRegion #1**:  In the **View** window,new **Subregions** will be painted in different colors:  Icon **Sliding surface #0** in the project tree will change color from gray to blue.

Step	Actions
	Note 1: There is an opportunity to reconnect separated subregions, removing the sliding surface. To do this, use the **Revert** command from the context menu **Sliding surface**. ***Note 2:*** After insertion of the sliding surface direction of the axis of rotation is unavailable for editing. Because it is impossible to change the orientation of the axes of the **Local CS #0** and the **Rotation #0** axis direction. In this case, you can change the speed of rotation.
9	Enter the properties of the physical model: • in folder **Substances** create **Substance #0** and load it from the standard base material substances **Air_Gas (equilibrium)** phase • in folder **Phases** create a continuous phase **Phase #0**and add to it the substance **Air_Gas (equilibrium)** • specify the settings in the folder properties **Physical processes** created **Phase #0**: ○ **Movement = Navier-Stokes model** ○ **Turbulence = KES** • in folder **Models** Create **Model #0** and add to it the **Phase #0** • in the child element of the **Setup. data #0**,located in a subfolder of the **Setup. data** in the **Model #0**, specify: ○ **Pulsations (Phase #0) = 0.01** ○ **Turbulence scale (Phase #0) = 0.01**
10	For each subregion **Subregion #0** and **Subregion #1** in the **Properties** window to change some of the parameters: • subregions set meaningful names - **Stator** and **Rotor** of our example • set **Model = Model #0**  ***Please note:*** While subregion will not contain boundary condition **Sliding surface**, the properties of this subregion can not specify a local coordinate system and rotation (**local coordinate system** parameters and **Rotation**).
11	Set the boundary conditions for the subregions of **Rotor**and **Stator**.

Step	Actions



Boundary conditions for **Rotor**: 1 - rotor housing; 2 - inlet air; 3 - sliding surface; 4 - rotor blades

Impeller housing (1), specify:

- **Type =Wall**
- **Variable> Speed (Phase #0) = Logarithm law**
- **Variables > TurbEnergy (Phase #0) = value of the cell away from the wall**
- **Variables > TurbDissipation (Phase #0) = value of the cell away from the wall**

For *Air inlet* (2), specify:

- **Type = Inlet / Outlet**
- **Variable > Speed (Phase #0) = Normal mass velocity** and set **mass speed = 20** in the properties of the child element **Velocity (Phase #0)**
- **Variables > TurbEnergy (Phase #0) = Pulsations** and set the **value = 0.01** in the properties of the child element **TurbEnergy (Phase #0)**
- **Variables > TurbDissipation (Phase #0) = scale turbulence** and set the **value = 0.01** in the properties of the child element **TurbDissipation (Phase #0)**

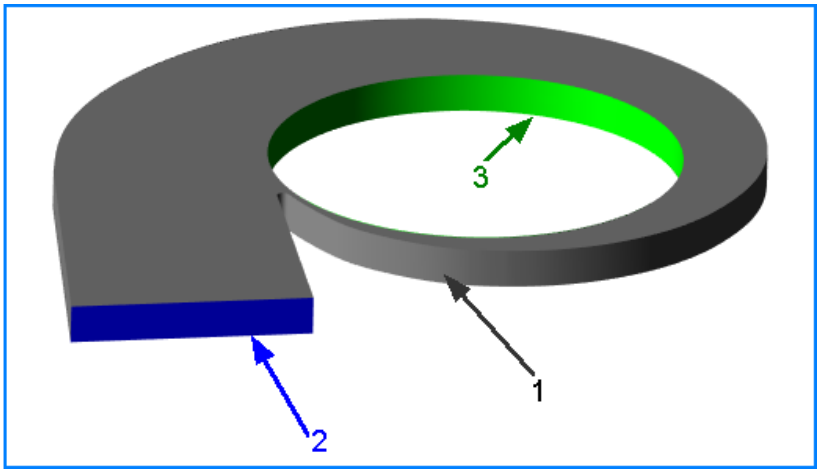
For the *sliding surface of the rotor* (3) specify:

- **Type = Linked**

For the *rotor blades* (4) specify:

- **Type =Wall**
- **Local CS = Local CS #0**
- **Rotation = Rotation #0**
- **Variable > Speed (Phase #0) = Logarithm law**
- **Variables > TurbEnergy (Phase #0) = value of the cell away from the wall**
- **Variables > TurbDissipation (Phase #0) = value of the cell away from the wall**

Step	Actions



Boundary conditions for **Stator**: 1 - stator housing; 2 - inlet; 3 - sliding surface

For the *stator housing* (1) specify:

- **Type =Wall**
- **Variable> Speed (Phase #0) = Logarithm law**
- **Variables > TurbEnergy (Phase #0) = value of the cell away from the wall**
- **Variables> TurbDissipation (Phase #0) = value of the cell away from the wall**

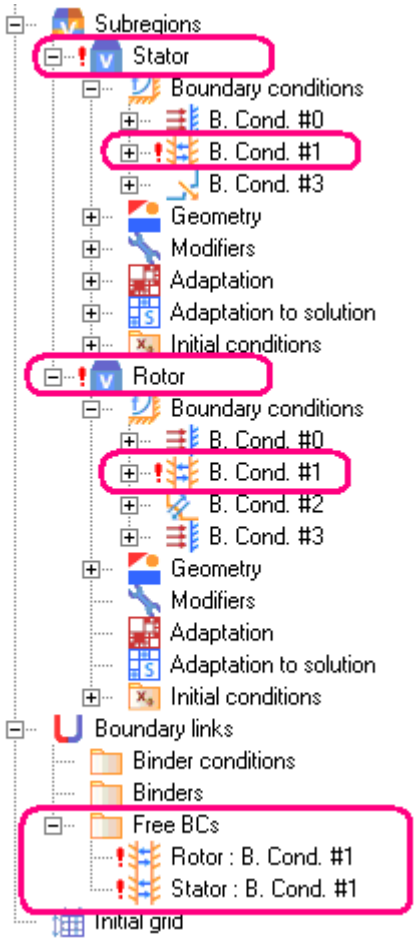
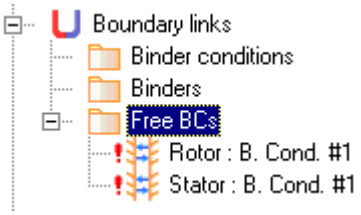
For *vent* (2) specify:

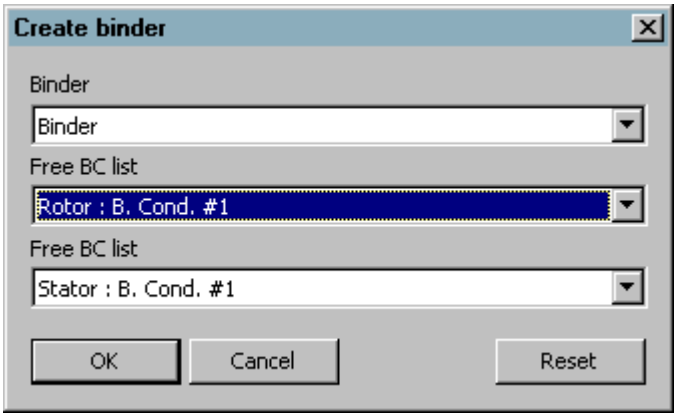
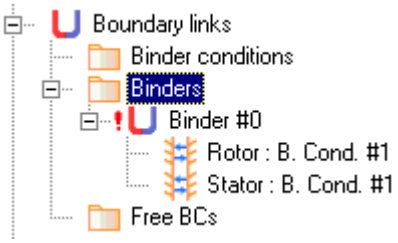
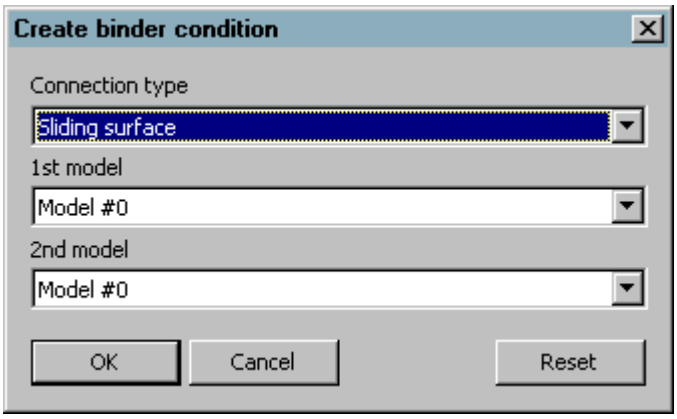
- **Type = Free outlet**
- **Variable> Speed (Phase #0) = Pressure** and set **Value = 0** in the properties of the child element **Velocity (Phase #0)**
- **Variables > TurbEnergy (Phase #0) = Pulsations** and set **Value = 0.01** in the properties of the child element **TurbEnergy (Phase #0)**
- **Variables > TurbDissipation (Phase #0) = scale turbulence** and set **Value = 0.01** in the properties of the child element **TurbDissipation (Phase #0)**

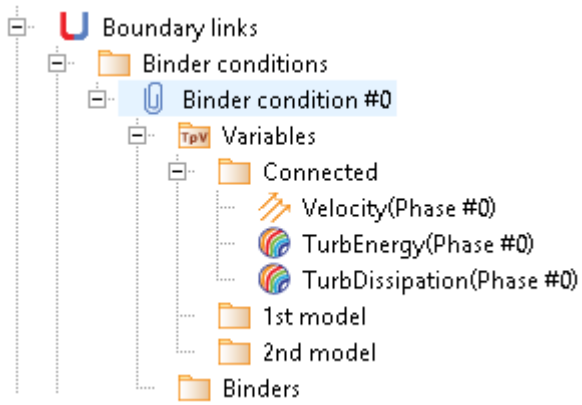
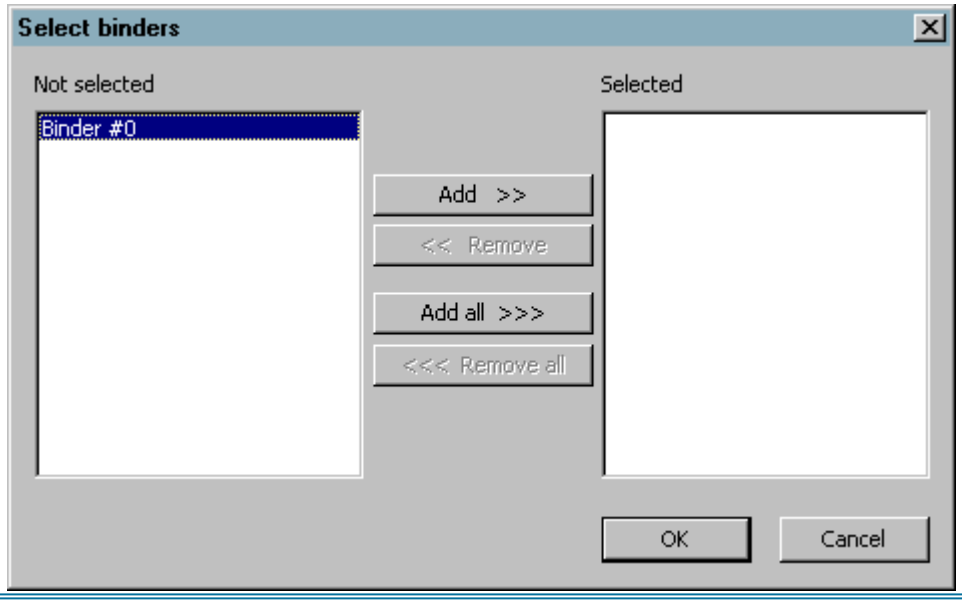
For the *sliding surface of the stator* (3) specify:

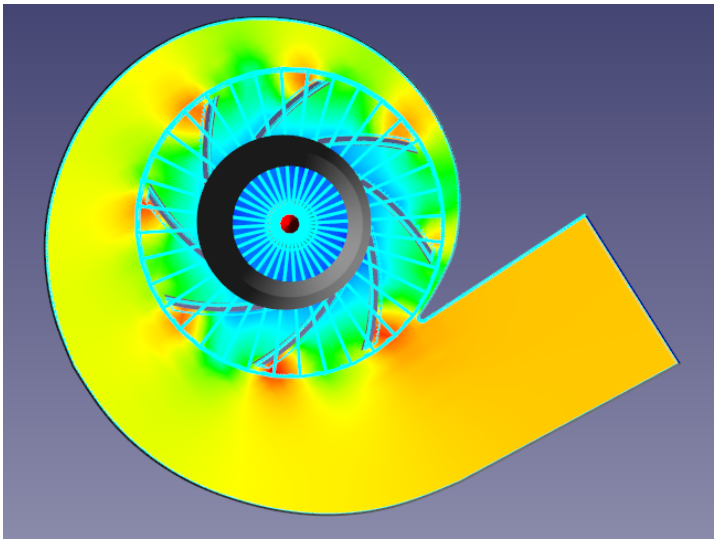
- **Type = Connected**

Boundary conditions, with the type of **Connected** and themselves subregion **Rotor** and **Stator** will be identified in the project tree, an exclamation point symbol ("**!**"). Since these boundary conditions have not yet been compared with each other. Also, these boundary conditions will fall as the elements in the project tree in the folder **Boundary links > Free BCs**:

Step	Actions
	 *(The names of the boundary conditions in the project tree can not comply with the symbols in the figures)*
12	Bind the sliding surfaces - the moving surface of the **Rotor** with the sliding surface of the **Stator**. At this stage they are not linked and are shown in the project tree in a subfolder **Boundary links > Free BCs**:  To do this, in the context menu of the subfolder **Boundary links > Binders** select **Create** and in the **Create binder** form, which opens, select boundary conditions corresponding to the sliding surfaces of the **Rotor** and **Stator** from the **Free BC list** fields:

Step	Actions
	 Click **OK**. In the project tree will be a binder of **Binder #0** in the subfolder **Boundary links > Binders**:  The just created **Binder #0** is marked in the project tree with a red exclamation point symbol ("**!**") because no **Binder condition** is specified for the boundary conditions, which are being bonded, yet. In our example, the same effect the **Create all** command would do selected from the context menu of the **Boundary links > Binders** subfolder. The **Create all** command automatically searches connected boundary conditions and forms **Binders** based on them.
13	First, create a new **Binder condition**. To do this in the context menu subfolders **Boundary links > Binder conditions**, click **Create** and in the **Create binder condition** form, which opens, specify **Connection type = Sliding surface**. Do not change values in the fields **1st model** and **2nd model**.  Click **OK**. **Binder condition #0** will appear in the project tree in the **Boundary links > Binder conditions** subfolder:

Step	Actions
	
14	Match **Binder condition #0** with **Binder #0**. To do this in the context menu subfolders **Boundary links > Binder conditions > Binder condition #0 > Binders**, click **Add/remove**, and then in the form that appears, select **Binder #0** in the panel **Not selected**, move it to the pane **Selected**, and then click **OK**. 
15	Now, when the sliding surfaces of rotor and stator are connected to a sliding surface **Binder condition #0**, you can set the rotation of the **rotor** subregion (**Stator rotation**set is not necessary, because it is stationary). In the **Properties** window, set the **Rotor** subregion's settings: • **Local CS = Local CS #0** • **Rotation = Rotation #0**
16	Specify other project settings (initial grid settings **Solver**, display settings) and can run the project on the computation. *FlowVision* will make the computation taking into account connection of mobile subregion **Rotor** and the stationary subregion **Stator** by the sliding surface:

Step	Actions
	

Please note: There is a feature of the solution of problems with the sliding surface. Namely, the time step must be such that the cell mobile subregion (in our example -**Rotor**), adjacent to the sliding surface, *not jumped* cells fixed subregion (in our example -**Stator**), adjacent to the sliding surface. This requirement can be automatically satisfied when **Method = Via CFL number** is set in parameters of the time step. By default, **Slide CFL = 1**. It is the parameter, which guarantees fulfillment of the above mentioned requirement to the time step.

See also: section Folder «Sliding surfaces».

8.2.3.9.4 Operations with groups of facets

As the number of operations with groups composed of the following facets:

- display (hide) the list of groups of facets in the project tree;
- regrouping of facets, facets belonging to a certain group of facets;
- setting boundary conditions on a group of facets;
- selection of a group of other facets;
- creation a supergroup
- identifying the group of facets in the **View** window

Displaying (hiding) a list of facet groups in the project tree

Step	Actions
1	Select **File> Preferences** from the main menu. **Settings** dialog box appears.
2	Select **Yes** (**No**) parameter **Display> Show all groups**.
3	Click **OK**.

Regrouping facets belonging to a certain group of facets

Step	Actions
1	Open the context menu of the element **Region- Group #N** and select **Regroup**. The **Geometry regrouping** window will open.
2	Specify a criterion for grouping - parameter **Threshold angle (in degrees)** and unselect or keep selected the **Prevent changing boundary conditions on triangles** checkbox.
3	Click the **Preview** button. The results of the group are displayed in the table on the **Final statistics** panel.
4	After reaching the desired result lock it by pressing the **Apply** button.

Setting the boundary conditions on the group of facets

Step	Actions
1	Open the context menu of the element **Region- Group #N** in the folder **Subregions > SubRegion #N> Geometry > Region- Surface #N**.
2	Select **Boundary condition > B. Cond #N** from the context menu.

Select a group of other facets

Step	Actions
1	Open the context menu of the element **Region- Group #N** in the folder **Subregions > SubRegion #N> Geometry>Region- Surface #N**.
2	Select Select the other side of the context menu. The direction normal to the surface of the object is reversed.

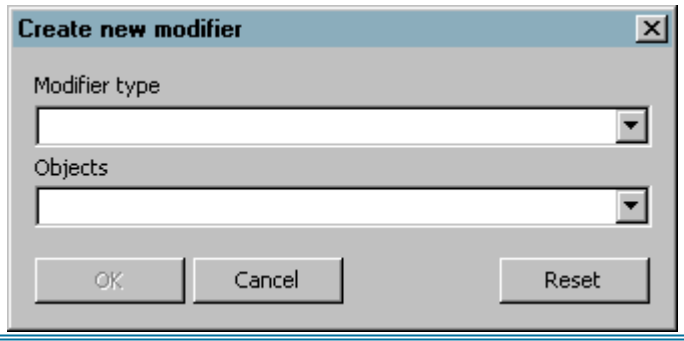
8.2.3.9.5 Operations with Modifiers

Among the operations performed by all modifiers (including mobile body), is composed as follows:

- the creation of the **Modifier**
- copy of the **Modifier**
- removal of the **Modifier**
- change in the parameters of the **Modifier**

Create a modifier (except modifier «Moving body»)

Before creating the **Modifier** create the object to which this **Modifier** should be attached. **Modifiers** for non-type **Moving body**, the **Object** maybe standard object or **Imported objects**.

Step	Actions
1	Right-click the folder line **Subregions > SubRegion #N > Modifiers** and select the **Create** command. A dialog box **Create new modifier**: 
2	Select the type of modifier and an object on which to create a modifier.
3	Click **OK**. Created **modifier**i s added to the folder **Subregions > SubRegion #N > Modifiers** and folder **Objects>(object)**.

Copy modifier

Step	Actions
1	Open the context menu of the element **Modifier**, to be copied in a folder **Subregions > SubRegion #N> Modifiers** or in a folder **Objects > (object)**.

Step	Actions
2	Select **Copy** from the context menu. Created modifier is added to the folder **Subregions > SubRegion #N> Modifiers** and folder **Objects > (Object)**. Modifier parameters match those of the copied modifier.

Removal of the modifier

Step	Actions
1	Open the context menu of the element **Modifier**, to be disposed of in a folder **Subregions > SubRegion #N > Modifiers** or in a folder **Objects > (Object)**.
2	Select **Delete** from the context menu. **Modifier** is removed from the folder **Subregions > SubRegion #N> Modifiers** and folder **Objects > (Object)**.

Change settings modifier

Step	Actions
1	Select from the project tree to edit the element **Modifier** in the folder **Subregions > SubRegion #N > Modifiers** or in a folder **Objects > (Object)**.
2	Adjust the values of the parameters in the **Properties** window, and then click **Apply**.

Operations with modifiers «Moving body»

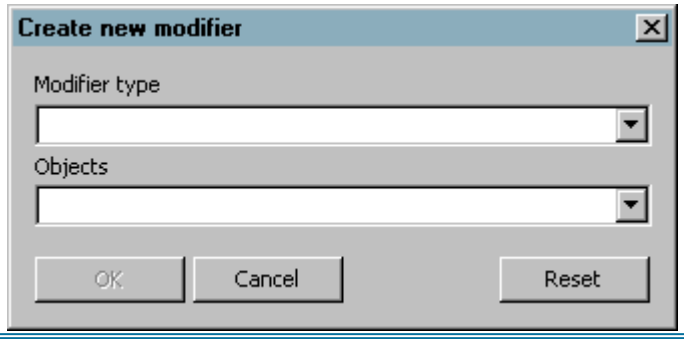
As the number of operations with modifiers "**Moving body**"includes some specificoperations:

- import group **Imported objects** and the creation of these modifiers **Moving body**
- assignment of boundary conditions on**Moving** body;the boundary conditions on the moving body can be specified in two ways:
  - by setting a uniform boundary conditions on the surface of all groups of facets of the moving body
  - by setting different boundary conditions on the surface of the groups of facets of the moving body
- **Moving body** contortion inside
- replacement of imported object with which the **Moving body**
- transformation (conversion) of the imported object, which is associated with **Moving body**
- regrouping of the imported object, which is associated with **Moving body**
- export to a file imported object with which the **Moving body**

Creation a «Moving body» modifier on existing Imported object

Before you create a modifier to create, to which should be attached modifier. **Moving body** modifier object can only be **imported objects**. **Moving body** modifier can not be created on the**imported objects**in the case at this site are given an element **movement**.

Step	Actions
1	Open the context menu of the folder **Subregions > SubRegion #N > Modifiers** and select command **Create**. Dialog box **Create new modifier** will open:

Step	Actions
	
2	Select **Moving Body** as the **Modifier type** and choose one of existing **Imported objects** from the **Object** drop-down list.
3	Click **OK**. As a result, the following actions: • **Moving body** created is added to the folder **Subregions > SubRegion #N > Modifiers** and folder **Objects> (object)** • folder in **Subregions > SubRegion #N > Geometry** added element **Moving body #N**, which is associated with elements of **Moving body #N - Group #M** • folder in **Subregions > SubRegion #N > Boundary conditions** added element**Boundary conditions** • to add items **Moving body #N - Group #M** put added boundary conditions**Boundary** conditions.
4	Set the necessary properties in the **Properties** window of the **Moving** body,which you have just created. ***Notes:*** 1) Every **Moving body**'s update leads to computational grid rebuilding. This process can consume a certain amount of processor time. Therefore, disabling updates is recommended if your case involves a static **Moving** body. If your project includes several **Moving bodies**, then updating one of them will cause others to update as well. 2) If a constraining **Object** has a **Movement** defined, then the limiting point T of the **Moving body** can be transferred outside the **Object**'s boundary if the **Object** moves. In this case, this constraint does not work.

A **Moving body** can only be created on an **Imported object** with no **Movement** defined.

After creating a **Moving body** on an **Imported** object, properties of the **Imported object** become read-only.

Creation a «Moving body» modifier using geometric data from a file

When creating**Moving body**in the geometric data of the file, you first create these data**Imported** object. Then, on the basis of**the imported objects**can be created**Moving body**as previously described in the previous section.

To create an **Imported object** based on geometric data from a file and also create a **Moving body** on this **Imported** object,do the following:

Step	Actions
1	In the**Preprocessor**tab of the project tree in the context menu of the **Objects** folder select: • command **Create** and create an **Imported object** • *or* command **Batch import** (which allows you to create several **Imported objects** at once)
2	A window opens, where you have to select a file with geometric data. If you used the**Batch import**command on the previous step, you can select several files.
3	A new **Imported object** (or several **Imported objects**) will appear in the project tree.

Step	Actions
	Create a **Moving body** (or several **Moving bodies** for batch import) *on the newly imported object*(s) as was described in subsection*"Creation a «Moving body» modifier on existing Imported object"* above.

When you create an **Imported object**based on a geometry file, only a single geometry file can be used for this. The imported object geometry must conform to the geometric data requirements. Inner surfaces of an imported object will be automatically deleted after a **Moving body** is created on it.

After creating a **Moving body** on an **Imported object**,a test for self-intersections is recommended.

Note: To create multiple **Moving bodies** based on geometric data from files, you can use the batch import (see section below), which will speed up the user experience.

Group creation modifiers «Moving body»

When using the **batch import**, you can select multiple files at once geometry. This will create a few **Imported objects** or **Imported objects** and **Moving bodies**. In this case, **Imported objects** and **moving body** they will be created automatically.

Step	Actions
1	Right-click the folder line **Subregions > SubRegion #N> Modifiers** and choose **Batch import**.
2	In the window that opens,select **Open files**, which are created andloaded into a tree **Imported objects**, which will also set the **moving body**.
3	Click **OK**. As a result, the following actions: • by **Imported objects** are added to the folder **Objects**. By modifiers are added to the folder **Subregions > SubRegion #N > Modifiers** and folder **Objects > Imported object #N**. • folder in **Subregions > SubRegion #N > Geometry** adds items **Moving body #N**, which is associated with elements of the **Moving body #N - Group #M** • folder in **Subregions > SubRegion #N>Boundary conditions** added element**Boundary conditions** • to add items **Moving body #N - Group #M** put added **Boundary conditions**.

Specifying the boundary conditions on the moving body by setting uniform boundary conditions on all groups of facets

Step	Actions
1	Right-click the modifier**Moving body** folder **Subregions > SubRegion #N > Modifiers** or in a folder **Objects > Imported object #N**.
2	Select **Set boundary condition > B. Cond. #N** in the context menu.

Specifying the boundary conditions on the moving body by setting different boundary conditions on all groups of facets

Step	Actions
1	Open the context menu of the element **Moving body #N - Group #M** in the folder **Subregions > SubRegion #N > Geometry > Moving body #N**.
2	Select **Set boundary condition > B. Cond. #N** in the context menu.
3	Repeat steps 1 and 2 for all groups of facets of the moving body **Moving body #N - Group #M**.

Turn a moving body inside out

Step	Actions
1	Right-click the modifier **Moving body** folder **Subregions > SubRegion #N > Modifiers** or in a folder **Objects > Imported object #N**.
2	Click on **Turn inside out**. The direction normal to the surface of the object is reversed.

Replacing imported object with which the moving body

Step	Actions
1	Right-click the modifier **Moving body** folder **Subregions > SubRegion #N> Modifiers** or in a folder **Objects > Imported object #N**.
2	Click **Replace geometry**.
3	Opens a standard **Open** dialog box,select the file from which the object is imported, and then click **Open**.

The transformation of the imported object with which the moving body

Step	Actions
1	Open the context menu modifier **Moving body** folder **Subregions > SubRegion #N > Modifiers** or in a folder **Objects > Imported object #N**.
2	Click **Transform geometry**. Once clicked, the Geometry transformation dialog box will open. Transformation geometry is made by pressing the**OK** button in this window (the window closes **Convert geometry**). Clicking the **Cancel** button or icon **"x"** in the upper right corner to close the **Geometry transformation** dialog box window transforming the geometry. Clicking on the button **Apply** to see the result of transformation to the actual implementation of the transformation (i.e., the **Apply** button performs the preview). Individual actions to be taken in the course of the transformation, made in a manner consistent with the location of items in the **Geometry transformation** dialog box: 1. Determined by the operations center 2. Is scaled 3. Produced rotation 4. The shift
3	Specify conversion options and click **Apply** to preview the transformation in the **View**. If necessary, repeat this step by including other data, until you get satisfied with the results.
4	Click **OK**, to actually carry out the transformation geometry in accordance with the entered data. The **Geometry transformation** dialog box will close at the same time.
5	If you need to perform several transformations geometry (eg, several consecutive turns), repeat the previous steps as many times.

Regrouping of the imported object with which the moving body

Step	Actions
1	Open the context menu modifier **Moving body** folder **Subregions > SubRegion #N> Modifiers** or in a folder **Objects > Imported object #N**.

Step	Actions
2	Click **Regroup geometry**. The **Geometry regrouping** window will open.
3	Specify a criterion for grouping - parameter **Threshold angle (in degrees)** and unselect or keep selected the **Prevent changing boundary conditions on triangles** checkbox.
4	Click the **Preview** button. The results of the group are displayed in the table on the **Final statistics** panel.
5	After reaching the desired result lock it by clicking the **Apply** button.

Export to file imported object with which the moving body

Step	Actions
1	Right-click the modifier **Moving body** folder **Subregions > SubRegion #N > Modifiers** or in a folder **Objects > Imported object #N**.
2	Click **Export geometry**. Once clicked, a standard dialog box**Export** geometry.
3	Select the file type, select the file name and click **Save**.

8.2.3.9.6 Operations with initial conditions

As the number of operations with the initial conditions are the following:

- creation of the initial conditions;
- copy of the initial conditions;
- removal of the initial conditions;
- editing parameters of the initial conditions;
- change the order of the initial conditions in the element **Sub-region**, which is necessary for the regulation of the initial conditions before the start of the calculation.

Creating the initial conditions:

Step	Actions
1	Right-click the folder line **Subregions > SubRegion #N > Initial conditions**. Click **Create**.
2	New element **Init. condition #N** is displayed in the project tree in the folder **Subregions > SubRegion #N > Initial conditions** and folder **Objects**.

Copy the initial conditions:

Step	Actions
1	Open the context menu of the element **Initial condition**, which is to be copied, and select **Copy**.
2	New element **Init. condition #N** is displayed in the project tree in the folder **Subregions > SubRegion #N > Initial conditions** and folder **Objects**. Element parameters match those of the copied item **Initial condition**.

Removal of the initial conditions:

Step	Actions
1	Open the context menu of the element **Initial condition**, to delete, and select **Delete**.
2	Element **Init. condition #N** is removed from the project tree.

Change the settings for the initial conditions:

Step	Actions
1	Select from the project tree to edit elements of the **Initial conditions**.
2	Adjust the values of the parameters in the **Properties** window, and then click **Apply**.

Changing the order of the initial conditions:

Step	Actions
1	Open the context menu of the element **Initial conditions** in the folder **Subregions > SubRegion #N > Initial conditions**.
2	Click **Move Up** (**MoveDown**).

8.2.3.10 Operations with boundary links

The sequence of specifying connected boundary conditions on a surface that separates two subregions, is the following (all operations are performed in the folder **Boundary links** in the project tree, in the **Preprocessor** tab):

a) in each subregion are given model and the boundary conditions are (elements **Boundary conditions**) Bound

b) all groups of the facets of the surface on which to put the connected boundary conditions are given by related boundary conditions; these boundary conditions are displayed in a folder **Boundary links > Free BCs**

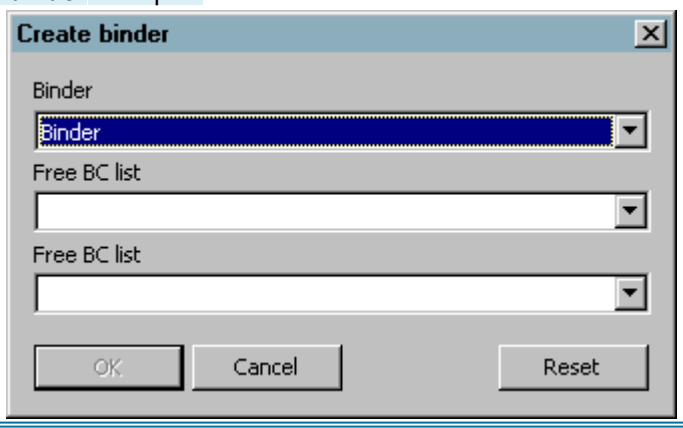
c) folder **Boundary links > Binders** created ligament (**Bundle** element) of a pair of boundary conditions on both sides of the surface (from the folder **Boundary links > Free BCs**);

d) folder **Boundary links > Binder conditions** created by the constraint (element **Binder** condition), which specifies the linkable model and type of connection (**Conjugate temperature** or **Conjugate all variables**).

As the number of operations with the boundary constraints include the following:

- the creation of a bunch of pairs of boundary conditions
- creating bundles of all the boundary conditions in the folder **Free BCs**
- removal of binders
- creation of binder conditions
- removal of the binder conditions
- editing binder conditions
- edit the binder parameters of **conjugate heat transfer**

Creating a bunch of pairs of boundary conditions:

Step	Actions
1	Right-click the folder row **Boundary links** >**Binders** and select the **Create** command. A dialog box **Create binder** will open: 
2	Select the boundary conditions, which are to be bound, from both **Free BC list** fields.
3	Click **OK**.

Step	Actions
	In the folder **Boundary links > Binders** displays created element **Bundle**.

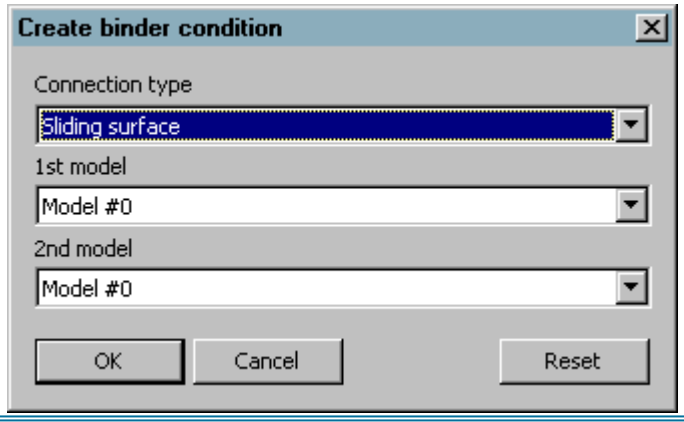
Create bundles of all boundary conditions, in the folder «FreeBCs»:

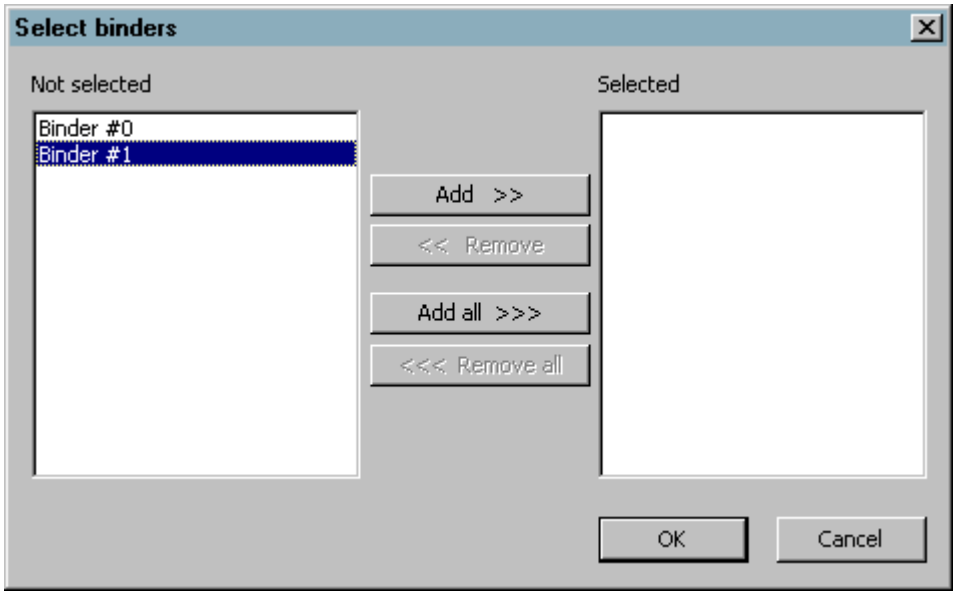
Step	Actions
1	Right-click the folder row **Boundary links > Binders** and select **Assign all**.
2	In the folder **Boundary links > Binders** displayed by elements **Binder #N**.

Deleting a bunch:

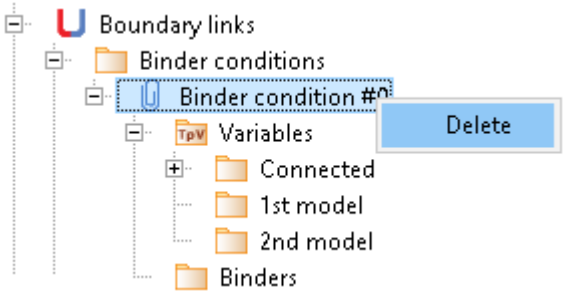
Step	Actions
1	If the item to be deleted **Binder #N** is included in a **Binder** condition, pre exclude it from all the elements.
2	Right-click the folder row **Boundary links > Binders** and select **Delete**.

Creating conditions for connection:

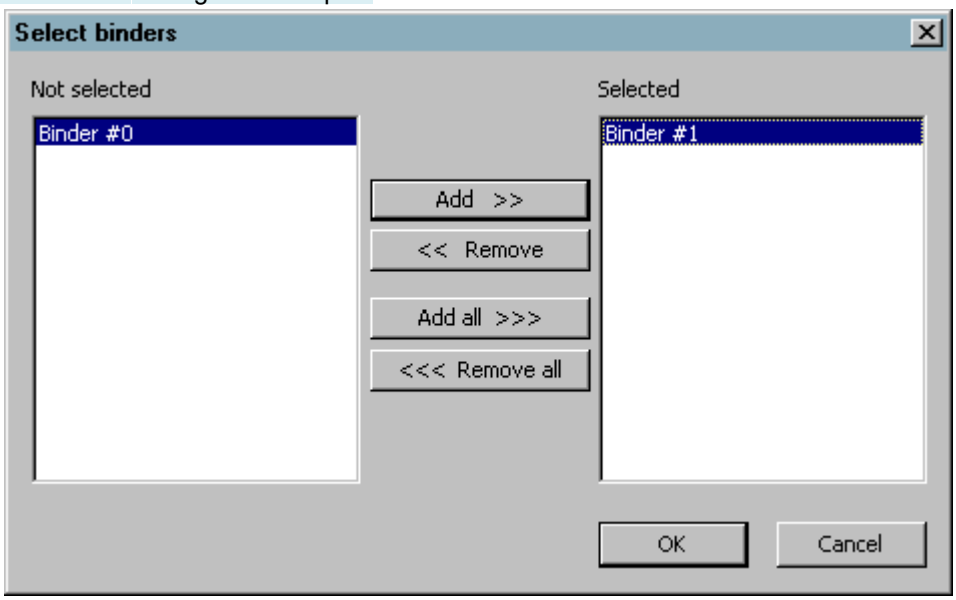
Step	Actions
1	Right-click the folder row **Boundary links > Binder conditions** and select the **Create** command. Dialog box appears **Create binder condition**: 
2	Select models in the bonding boundary condition subregions in the drop-down lists **1st model** and **2nd model**.
3	Select the type of connection in the drop-down list, type connection **Connection** type.
4	Click **OK**. In the folder **Boundary links >Terms of communications** will be a new (newly created) element **Binder condition #N**, which itself is a folder containing multiple elements.
5	Expand the newly created folder **Binder condition #N** and open the context menu is in its element **Binders**. Select in the context menu, click **Add/remove**. The **Select binders** window will open:

Step	Actions
	 In some cases, the conditions for connection type **Sliding surface** when comparing the **conditions due** to **a bunch of** program can offer in the **Select a bunch** of acceptable and unacceptable **Binders** options when the boundary conditions do not correspond to some Sliding surface #N in the project tree. Selecting this option will make the project unacceptable incorrect and will lead to an inability to start the project on the computation. Do not make mistakes when choosing a ligament in the **Select** ligament.
6	Create a list of the binders in the**Selected**: • highlight the added bundles in the list **are not** selected,and click **Add**; to add all the binders, click **Add All** • highlight excluded ligament in the **Selected** and click **Remove**; to the exclusion of all binders, click **Remove all**
7	Click **OK**. Selected binders appear as child nodes of the tree in the project folder **Boundary links > Binder conditions > Binders**.

Removal of the communication conditions:

Step	Actions
1	Open the context menu of the element Boundary links **> Binder conditions > Binder condition #N** and select there **Delete**: 
2	**Binder condition #N** will be deleted.

Change the settings for the communication conditions:

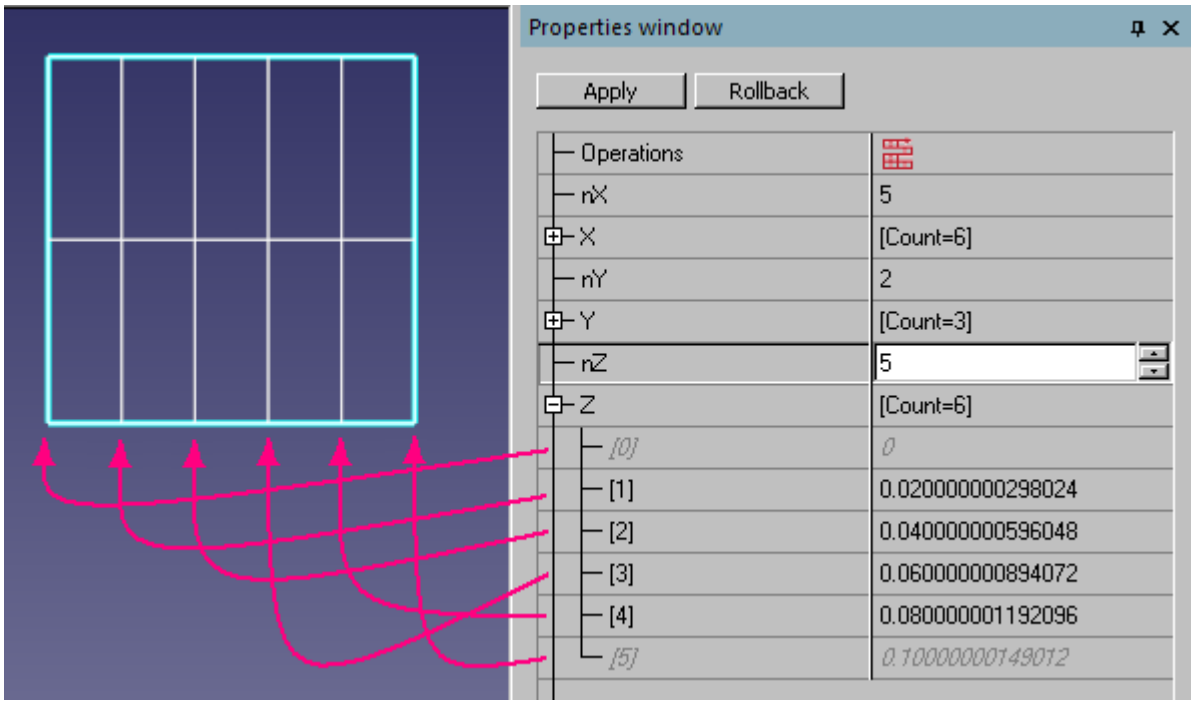
Step	Actions
1	Open the context menu of a folder **Boundary links > Binder conditions > Binder condition #N >Binders** and select **Add/Remove**. The **Select binders** dialog box will open:  In some cases, the conditions for connection type **Sliding surface** when comparing the **conditions due** to **a bunch of** program can offer in the **Select binders** of acceptable and unacceptable **Binders** options when the boundary conditions do not correspond to some Sliding surface #N in the project tree. Selecting this option will make the project unacceptable incorrect and will lead to an inability to start the project on the computation. Do not make mistakes when choosing a ligament in the**Select** ligament.
2	Create a list of the binders in the **Selected**: • highlight the added bundles in the list **are not** selected, and click **Add**; to add all the binders, click **Add All** • highlight excluded ligament in the **Selected** and click **Remove**; to the exclusion of all binders, click **Remove all**
3	Click **OK**. Selected bundles displayed in the folder **Boundary links > Binder conditions > Binders**.

8.2.3.11 Operations with initial grid

Defining a uniform initial grid along an axis

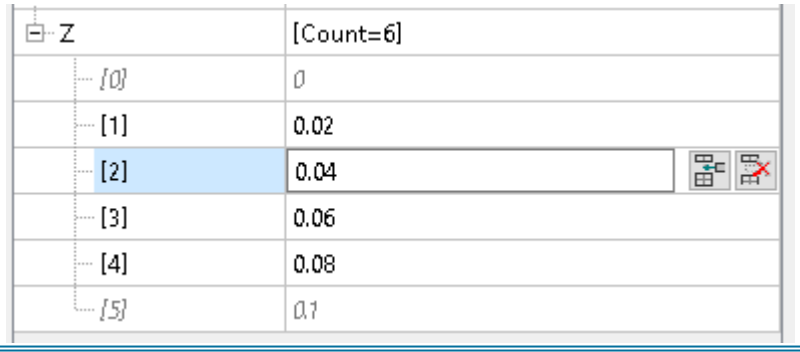
Step	Actions
1	Specify the number of cells of the initial grid along the axes (X, Y, or Z) in the **nX** (**nY**, **nZ**) field in the **Properties** window of the element Initial grid. ***Note:*** If the required number of steps is the same as the existing number of steps, then you have previously to clear the array of grid lines' data by clicking on the button in the field **X** (**Y**, **Z**) and confirming your decision in the pop-up window box that will be displayed.
2	Click the **Apply** button. In the **Properties** window of the element **X** (**Y**, **Z**) child elements **[0]**, **[1]**, ... , **[nX]** (**[nY]**, **[nZ]**) in amount of **nX** (**nY**, **nZ**) will appear with values, which provide uniform positioning of grid lines.

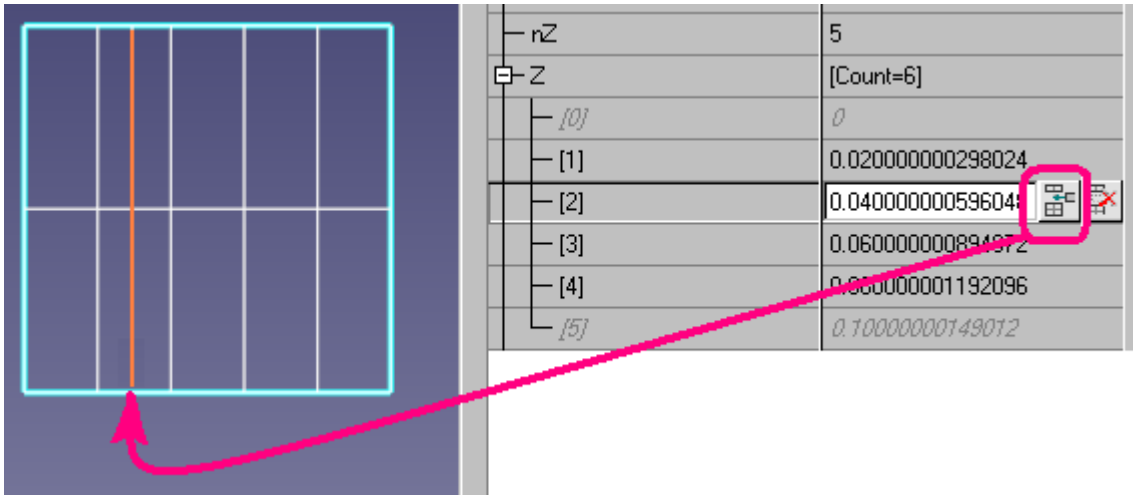
Step	Actions
	When required, you can change values of non-outermost elements. You can also insert new grid lines (see appropriate subsections below).



Example of a uniform initial grid with 5 steps along the axis **Z**

Inserting a new grid line between the selected line (non-initial) and the previous line

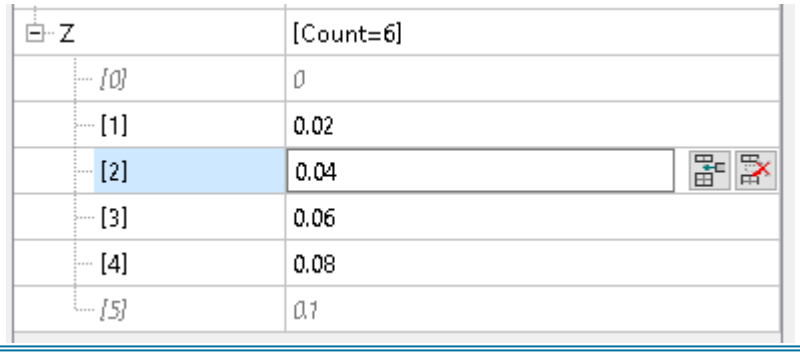
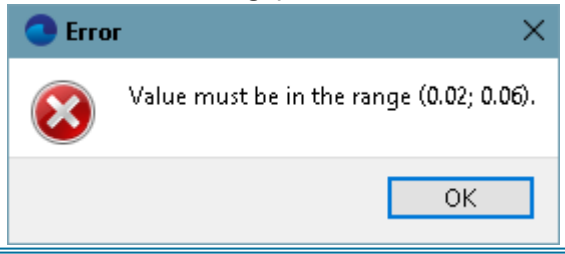
Step	Actions
1	Select in the **Properties** window of the element Initial grid an element **X > [N]** (**Y > [N]**, **Z > [N]**) corresponding to the selected grid line (which is *not* the initial grid line with index **0**). On the right of the selected grid line you will find screen buttons  **(Insert item before current)** and  **(Delete this item)**: 
2	Click the  **(Insert item before current)** button or press the **Ins** key on you computer's keyboard. The program will insert into the initial grid a line located between the selected line and its previous line.
3	Click the **Apply** button to accept your changes.



Inserting a new grid line (but not before the initial line "0")

Changing position of a non-outermost grid line

You can change positions of grid lines (except outermost grid lines, their positions are defined by boundaries of the computational domain). Changing of a grid line position is limited by positions of adjacent lines.

Step	Actions
1	Select in the **Properties** window of the element **Initial grid** an element **X > [N]** (**Y > [N]**, **Z > [N]**) corresponding to the selected grid line (which is not an outermost grid line): 
2	Press the **F2** key on you computer's keyboard or click by mouse in the data entry field of the element **[N]**. The field will be available for data entry.
3	Enter into the field a value of the new required position of the grid line. Press the **Enter** key on you computer's keyboard or locate the mouse cursor into another field. Position of the grid line will change. ***Note:*** The entered value must fall in the range between existing positions of the adjacent grid lines. If you attempt to enter an inappropriate value, the program will display an error message **"Value must be in the range (*boundaries of the range*)"**: 
4	Click the **Apply** button to accept your changes.

Deleting a non-outermost grid line

You can delete grid lines (except outermost grid lines that are always present at boundaries of the computational domain).

Step	Actions
1	Select in the **Properties** window of the element Initial grid an element **X > [N]** (**Y > [N]**, **Z > [N]**) corresponding to the selected grid line (which is not an outermost grid line).
2	Click the  **(Delete this item)** button on the right of the field or press the **Del** key on you computer's keyboard. The grid line will be deleted.
3	Click the **Apply** button to accept your changes.

Deleting all non-outermost grid lines of an axis

You can delete all non-outermost grid lines of an axis X, Y or Z using clearing the whole array of the lines.

Step	Actions
1	Select in the **Properties** window of the element Initial grid an element **X** (**Y**, **Z**), corresponding to an array of the grid lines on the appropriate coordinate axis.
2	Click the  **(Clear the array)** button on the right of the array's field or press the **Del** key on you computer's keyboard. A message box will open prompting you to confirm the deleting (**Are you sure you want to clear the array?**).
3	Click **Yes** to confirm deleting the grid lines.
4	Click the **Apply** button to accept your changes.

Exporting the Initial grid to a text file

Step	Actions
1	Right-click the element Initial grid in the project tree and select from the context menu the command **Export to a text file**. This opens a standard operating system's window **Save As**.
2	Specify the file name (`*.txt`), select the directory of its location, and then click **Save**.

Importing the Initial grid from a text file

Step	Actions
1	Right-click the element Initial grid in the project tree and select the context menu the command **Import from a text file**. This opens a standard operating system's window **Open**.
2	Select the file to be imported (`*.txt`) and then click **Open**.
3	Click the **Apply** button to accept your changes.

Defining the Initial grid in the Initial grid editor

Step	Actions
1	Click **Operations >**  **(Initial grid editor)** in the **Properties** window of the element Initial grid. This opens the Initial grid editor.

Step	Actions
2	Define a desired initial grid and then click **OK**.
3	Click the **Apply** button to accept your changes.

See also: sections Initial grid and Element «Initial grid».

8.2.3.11.1 Initial grid editor

The **Initial grid editor** allows you to build automatically a grid with specified step and ratios between adjacent steps on specified grid lines.

Terminology

(the formulae are provided relating to coordinates along axis X; similar formulae are used for other axes)

The *Grid step* is the distance $h_i = x_{i+1} - x_i$ between adjacent grid lines with coordinates x_{i+1} and x_i. Actually this is the cell size along the axis.

The *ratio between steps* is the ratio between the sizes of the adjacent cells $kh = \frac{h_{i+1}}{h_i}$.

The *reference line* is a grid line, on which the grid step and the ratio between adjacent steps on the left and on the right from the grid line (for the pair of adjacent cells on the left from the line and the pair of adjacent cells on the right from the line) are specified.

To generate an initial grid, you have to specify the following parameters:

- values of general control parameters
- positions of required reference lines
- values of control parameters on the reference lines

Parameters specified in the Initial grid editor and their limitations

The **Initial grid editor** dialog box is used to specify parameters for generation the initial computational grid.

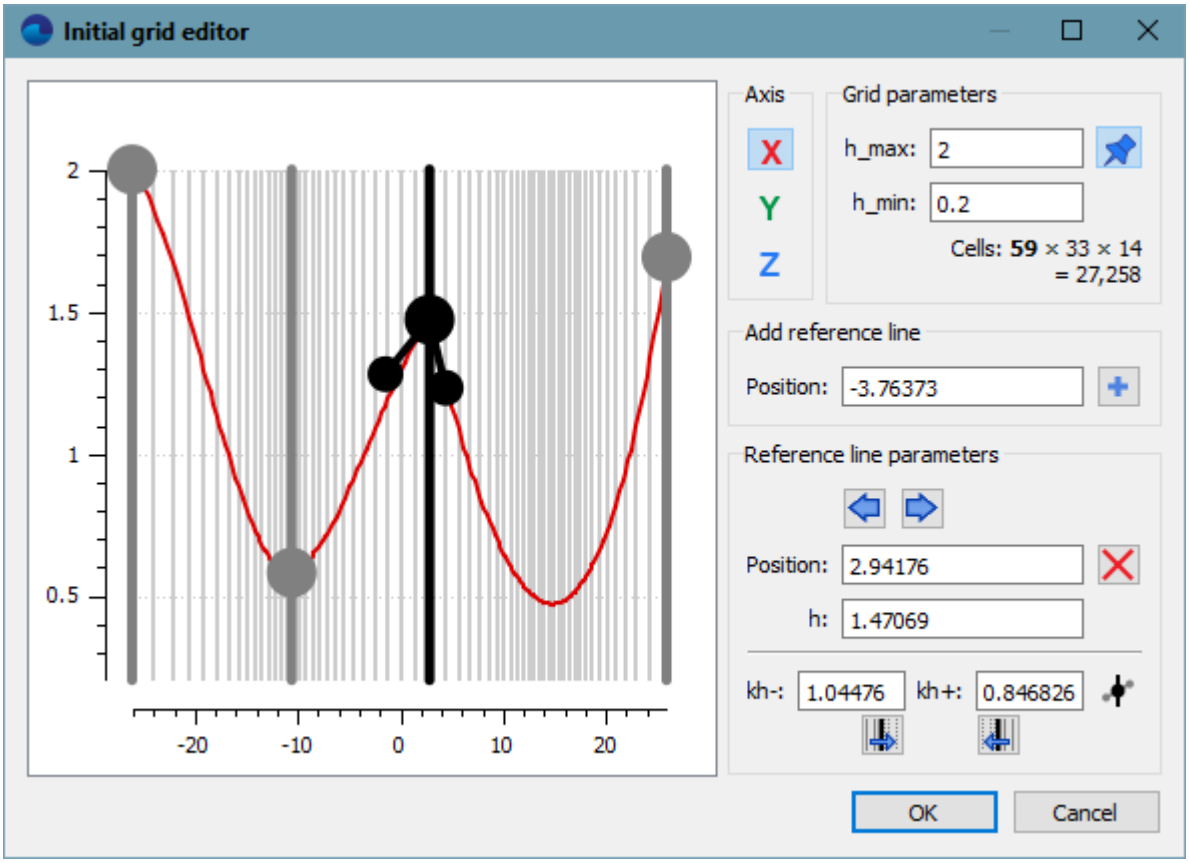
The grid for a three-dimensional domain is generated as three one-dimensional grids, x_i, y_j, z_k with variable step along each of the coordinates.

The **Initial grid editor** allows generation a grid with specified steps and ratios between adjacent steps on user-specified reference lines.

When an initial grid is being generated, the following constraints are applied:

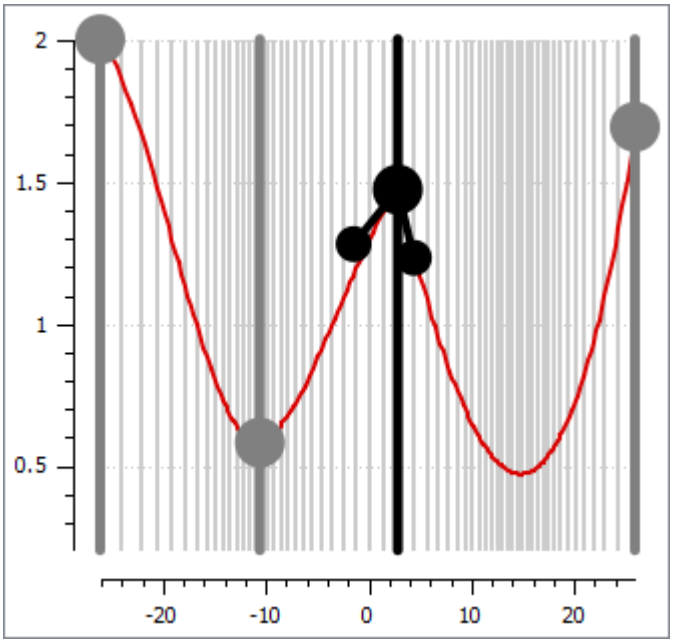
- the ratio between adjacent steps cannot exceed the value $k_{h\max}$=2.
- the minimal distance $d_{min,real}$ between reference lines, [m], cannot be less then $d_{min} \times L$, where d_{min}=0.025 and L is length of the computational domain along the selected axis.
- the maximal grid step h_{max}, [m], is specified by the user but can be automatically changed by the program
- the minimal grid step h_{min}, [m], is specified by the user (by default, $h_{min}=h_{max}/10$)
- $h_{max} \le d_{min,real}/4$ (this constraint shows itself in the program interface either as impossibility of adding or moving reference lines at some positions or as changing the h_{max}, which has been specified by the user).
- $1.001 \le h_{max}/h_{min} \le 1000$

User interface of the Initial grid editor



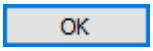
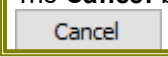
Dialog box of the **Initial grid editor**

The dialog box of the **Initial grid editor** has the following interface elements:

Element	Description
Graphical pane 	The graphical pane of the **Initial grid editor** displays a plot of how the grid step depends on the coordinate along the selected coordinate axis (X, Y, Z), and also the pane displays grid lines (reference lines and grid lines that will be created). The plot is colored according to the selected coordinate axis. Here you can select a reference ling by clicking it with the mouse. Any reference lines, except outermost ones, can be moved by the mouse right/left. Possible motion of the lines is limited to meet the specified value of h_{min} and mandatory relations between h_{max} and h_{min}. When the  screen button is pressed, motion of the lines is additionally limited by the specified value of h_{max}. The grid step, which is specified for a reference line, is marked with a round slider:  On the selected reference line this slider can be moved up/down with the mouse. Also the slider on the selected reference line includes handles that can be used to tune direction of the plot line and,

Element		Description
		respectively, gradient of the grid steps near the reference line (the handles are also can be moved by the mouse):  Tilts of the handles correspond to values of parameters **kh-** and **kh+** (see below). Movements of objects in the graphical pane can be interrupted by pressing the **Esc** key.
Axis		Group of buttons for selection an axis, along which the grid is specified.
	The **X** button 	Selection of the coordinate axis **X**
	The **Y** button `	Selection of the coordinate axis **Y**
	The **Z** button 	Selection of the coordinate axis **Z**
Grid parameters		
	h_max	The maximal grid step h_{max}, [m]. It is specified by the user.
	h_min	The minimal grid step h_{min}, [m]. It is specified by the user.
	 **(Disallow h_max to be changed by reference line additions and movements)**	When this button is pressed (), adding of reference lines or changing positions of existing reference line are not allowed if they require the specified h_{max}. Such attempt will cause displaying a message: • **Can't add a reference line. Change h_max or allow it to be changed by reference line additions and movements.** • **Can't add a reference line with the specified position. Change h_max or allow it to be changed by reference line additions and movements.** • **Can't move this reference line to the specified position. Change h_max or allow it to be changed by reference line additions and movements.** When this button is released (), changing of h_{max} is not prohibited, but positions of the lines are subject to limitations to relations between h_{min}, h_{max} and $d_{min,real}$. In this case operations of adding or moving reference lines might cause reducing h_{max}.
	Cells: ...	Here the program informs you about number of cells along axes X, Y, Z and the total number of cells.

Element		Description
		Number of cells along the selected axis is shown in bold.
Add reference line		
	Position:	This is the coordinate of a new reference line that will be inserted, [m]. The default position, which is prompted to be entered, is the center of the interval on the left from the selected reference line (for the leftmost reference line it is the center of the interval on the right of the line).
	 **(Add a reference line with the selected position)**	When you click this screen button, a new reference line will be added at the specified position. Also you can a new reference line by double-click in the graphical pane. The specified position of the reference line might be corrected according to existing limitations. If the program had to correct the line's position, it displays the message: "**Can't add a reference line with the specified position.**" If adding the reference line is impossible even with correction of its position, the program displays the message: "**Can't add a reference line.**"
Reference line parameters		
	 **(Switch to the previous reference line)**	These screen buttons allow you to switch from the current selected reference line to an adjacent reference line (to the previous one, on the left, or to the next one, on the right). Also you can press keys → (right arrow) or ← (left arrow) on you computer's keyboard.
	 **(Switch to the next reference line)**	
	Position:	Coordinate of the selected reference line. At attempt to enter a new position that doesn't meet the existing limitations, the program will try to correct the position and will display the message: "**Can't move this reference line to the specified position.**"
	 **(Remove this reference line)**	Clicking this button removes the selected reference line. You cannot remove an outermost reference line.
	h	Step of the grid at the selected reference line. This parameter can be also changed by moving the slider on the selected reference line up/down.
	kh-	The ratio between the adjacent steps on the left of the selected reference line. This parameter can be also changed by moving the left handle of the slider on the selected reference line.
	kh+	The ratio between the adjacent steps on the right of the selected reference line. This parameter can be also changed by moving the right handle of the slider on the selected reference line.
	 **(Automatic kh values)**	When this button is active, parameters **kh-** and **kh+** are specified by the program automatically. In this situation handles of the slider on the selected reference line become inactive.

Element		Description
		After manual entering values of **kh-** and **kh+** the button  becomes inactive.
	 **(Choose kh+ to produce a symmetric grid near this reference line)**	Setting the **kh+** parameter so that the grid on the right near the selected reference line be symmetric to the grid on the left near this reference line.
	 **(Choose kh- to produce a symmetric grid near this reference line)**	Setting the **kh-** parameter so that the grid on the left near the selected reference line be symmetric to the grid on the right near this reference line.
The **OK** button 		Clicking this **OK** button closes the **Initial grid editor** and transfers the generated grid into the **Properties** window.  Don't forget to click the **Apply** button in the **Properties** window before selecting another element in the project tree. Otherwise your new **Initial grid** will be lost.
The **Cancel** button 		Exit the **Initial grid editor** without saving changes and without generating the new grid.

Notes:

- Entering a value into a numerical field occurs when you press the **Enter** key on the keyboard, when the cursor locates in the field, or also when you relocate the cursor into another input field. The program prevents entering inappropriate values and tries to corrects the entered values.
- You can simultaneously be working in the **Initial grid editor** and in the View window of **Pre-Postprocessor**. So, when you are working in the **Initial grid editor**, you can in the same time view forming the **Initial grid** in the **View** window, selecting the convenient view points and locating the grid refinements in desired places of the computational domain.
- For the outermost reference lines, which correspond to the appropriate outermost positions in the computational domain, only one of the **kh** coefficients is defined, because the grid is not defined on the left from the leftmost line and on the right from the rightmost line.
- During plotting the plot line (spline), the obtained step **h** on the reference line sometimes might be a bit less than the specified step. If necessary, enter the value of **h** a bit larger than the required value.
- When in properties of the Initial grid non-computational directions are specified along some axes (**X | Y | Z**) , the appropriate buttons for selecting these axes ( ,  , ) will be inactive and the grid will contain only one cell along such axes.

8.2.3.12 Defining the time step

Defining the time step	
Step	**Description**
1	In the **Properties** window of the Time step element, select the desired value of the **Method** parameter.
2	If **Method = In seconds** is selected, specify the time step in seconds as value of the **Constant step** parameter. Click **Apply**.
3	If **Method = Via CFL number** is selected, specify the following parameters: a) if flow of liquid or gas is calculated, set the value of the **Convective CFL** parameter b) if motion of two-phase medium with a phase interface surface or motion of moving bodies is calculated, set the value of the **Surface CFL** parameter c) if conditions are calculated where diffusion processes are important (flows of liquids or gases when viscosity is large), set the value of the **Diffusive CFL** parameter d) set values of the parameters **Max step** and **Min step** Click **Apply**.

8.2.3.13 Selection of numerical method and defining parameters of algebraic solver

See section Parameters of the numerical method for details of user interface of selecting the numerical method and parameters of solving the system of algebraic equations.

For a numerical method:

Step	Actions
1	Select the **Advanced settings** element in the project tree and in its properties expand the **Numerical method** group of parameters.
2	Set the options: a) Select the method of integration over time as the value of the **Time integration > Method**. b) select the order of approximation of the convective terms of the transfer equation as a parameter value **Advection scheme**. c) specify the use of high-order approximation (WPA), the pressure gradient and diffusion terms of the equation as a parameter value **Use SGA**.
3	Click **Apply**.

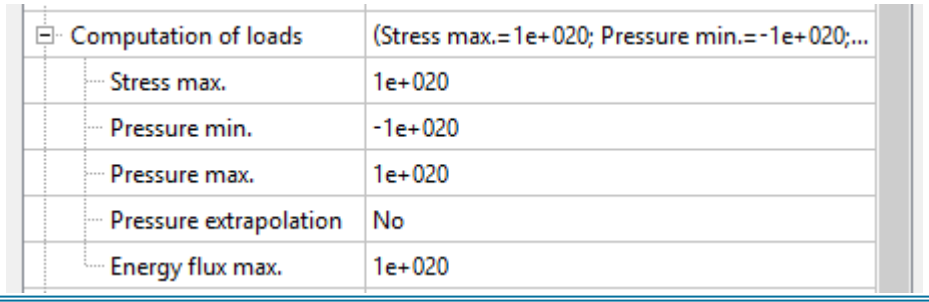
To set parameters for solving the system of algebraic equations:

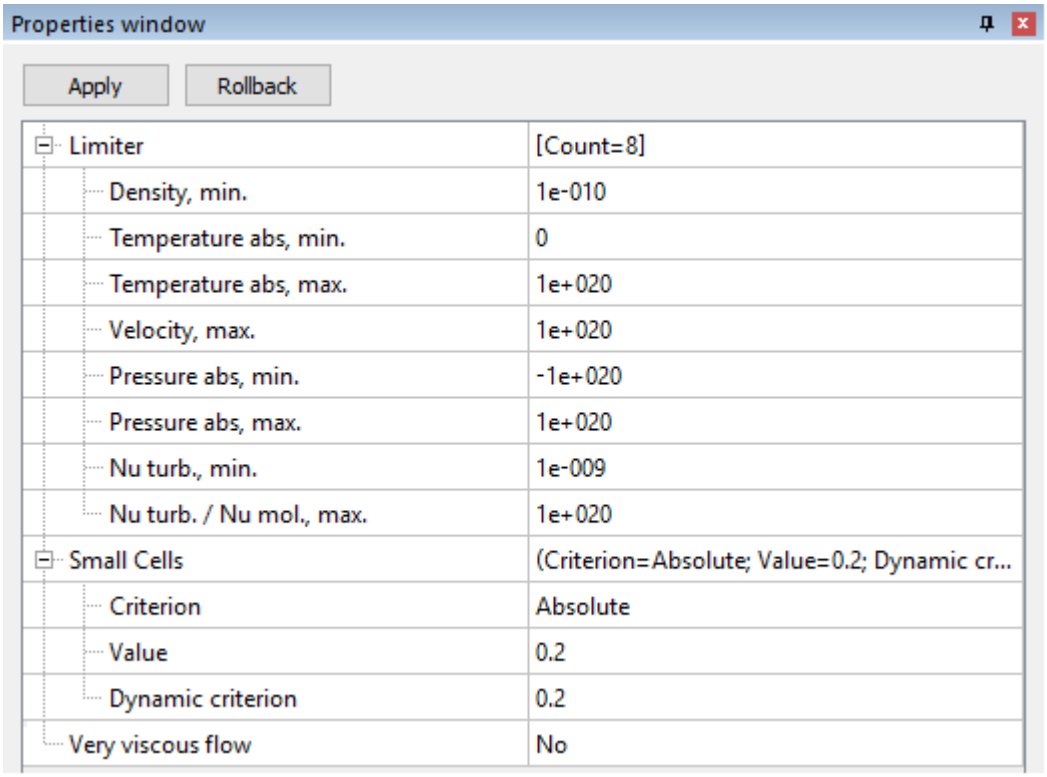
Step	Actions
1	Select the **Advanced settings** element in the project tree and in its properties expand the **Algebraic solver** group of parameters.
2	Set the parameters: a) Set the type of the algebraic solver in the parameter **Solver type**. Possible options are: • **TParFBSS** specifies use of the old algebraic solver of *FlowVision*. • **AST** specifies use of a new type of the algebraic solver, which is a combination of *Aggregation AMG*, *Selective AMG* and *TParFBSS* algebraic solvers. Use of the **AST** solver generally speeds up the simulation. b) Specify accuracy of the equations's convergence in the **Rel. tolerance** parameter.
3	Click **Apply**.

8.2.3.14 Specifying Limiters

See details about the limiters in section Constraining parameters.

To specify limiters:

Step	Actions
1	Select from the project tree in the **Solver** tab the element **Advanced settings** and in its **Properties** window, expand the **Computation of loads** group of parameters and specify the desired parameters limiting transferred loads: 
2	Click **Apply**.

Step	Actions
3	Select from the project tree in the **Solver** tab the element **Limiters** > **Limiters for calculation** > **Phase Limiters** > **Phase #N** for the appropriate **Phase** and in its **Properties** window specify the desired limiters:   In a liquid the minimal pressure is not be limited, but in a gas the minimal pressure should be limited by zero to avoid a negative density.
4	Click **Apply**.
5	Repeat steps 3 and 4 for all **Phases** for which you wish to specify limiters for calculation.

Notes:

1. Parameters of the element **Limiters** > **Limiters for calculation** > **NonPhase Limiters** contain service settings, which are not recommended to be changed.
2. Limiters for loads (group of parameters **Computation of loads** in the **Advanced settings of Solver**) limit the range of relative pressures used for calculation hydrodynamic forces acting of surfaces.

8.2.3.15 Defining multiphase parameters

To set parameters for multiphase:

Step	Actions
1	Select the project tree parameter string **Advanced settings** in the **Properties** window, expand the group of parameters **Multiphase C** (for continuous phase) or **Multiphase D** (for dispersed phases).
2	Set parameters in the **Properties** window, and then click **Apply**.

8.2.3.16 Defining checking of the calculation grid

Setting for checking the generating a computational grid

Step	Actions
1	Select in the project tree **Solver** tab the **Advanced settings** element.

Step	Actions
2	In the **Properties** window, set the value of this element **Check grid** parameter (select from the drop-down list).
3	In the **Properties** window, click **Apply**.

See also: section Validating the computational grid structure.

8.2.3.17 Defining parameters of turbulence model

See the section Turbulence model parameters for details about the turbulence model.

Setting the turbulence model	
Step	**Actions**
1	In the **Solver** tab, in properties of the element **Advanced settings**, expand the **Turbulence** group of parameters.
2	Select the method of calculating the distance from the center of the cell to the nearest wall in the drop-down list **Dist. via potential**.
3	Select the type of wall functions in the drop-down list **Standard wall functions**.
4	Specify the parameters limiters.
5	Click the **Apply** button.

8.2.3.18 Defining parameters of loading export

On the parameters of export loads see section Parameters of loadings export.

To set the export options of loads:

Step	Actions
1	Select the project tree parameter string **Advanced settings** in the **Properties** window, expand **exports loads**.
2	Select: • type of saving to a file from the **Type** list • recording mode from the **Write mode** list
3	Select any other options in the **Properties** window.
4	Click the **Apply** button.
5	Specify **Imported objects**, for which the surfaces are calculated load.

8.2.3.19 Defining parameters of data autosave

About automatic saving results calculations see section Settings for automatic saving of calculation results and visualization data.

To set the parameters automatically save results calculations:

Step	Actions
1	Select the project tree line parameter **Data autosave** and make the desired settings in the **Properties** window.
2	To save the results of a sequence of calculations, select **Yes** for the **story**. Select a value from the drop-down list and specify the type, if necessary, the frequency of saving the results calculations.

Step	Actions
3	Click the **Apply** button.

8.2.3.20 Defining of layers visualization data autosave

About auto save the data to visualize the layers see in section Analysis of the data saved at several time steps.

To set the parameters automatically save data to render layers:

Step	Actions
1	Select the project tree line **Layers autosave** option and specify the required settings in the **Properties** window.
2	To save the data sequence to visualize the layers, select **Yes** for the **History**.
3	Select a value from the drop-down list and specify the type, if necessary, the frequency of storing data for visualization layers.
4	If necessary, write data of a **solid** in the data for the visualization of the layers, select **Yes** for **Solids**.
5	Click the **Apply** button.

8.2.3.21 Defining the stopping conditions

Some of operations with characteristics include the following:

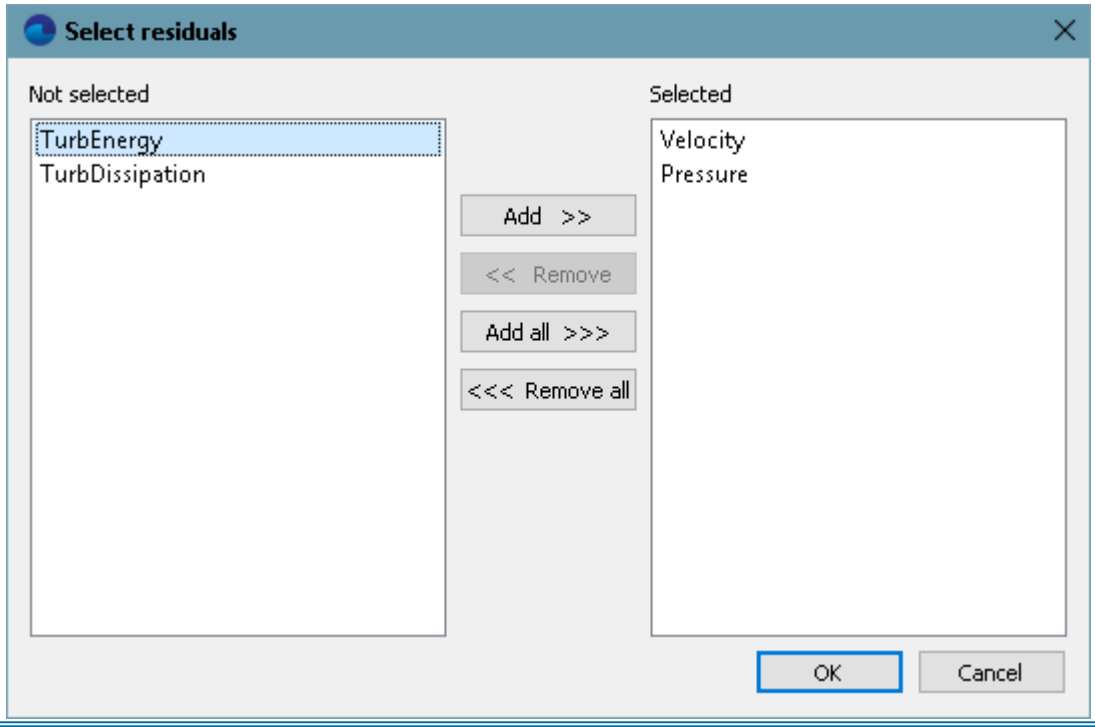
- setting conditions for stopping the calculation
- edit the list of residuals
- the creation of a stopping criterion
- up stopping criterion
- removal of a stopping criterion
- editing a stopping criterion

Setting conditions for stopping the calculation:

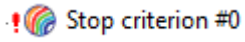
Step	Actions
1	Select **Length of time** in the project tree, and in its **Properties** window, set the start and end times of calculation. Click the **Apply** button.
2	Select **Number of steps** in the project tree and set in its **Properties** window, the number of time steps, which must be done before the end of the calculation. Click the **Apply** button.
3	Edit the list of residuals in the folder **Stopping conditions > Residuals** (see below).
4	In the folder **Stopping conditions > User values**, create the required criteria for a stop and set their parameters (see below).

Change the list of residuals (add and remove residuals):

Step	Actions
1	Open the context menu of the folder **Stopping conditions > Residuals** or of a **Residual** item, and select **Add/remove**. The **Select residuals** dialog box will open:

Step	Actions
	
2	Create a list of residuals in the **Selected** pane: • select the residuals, which should be added into the list, in the **Not selected** pane, and then click **Add** • select the residuals, which should be removed from the list, in the **Selected** pane, and then click **Remove**
3	Click **OK**. The selected residuals will appear in the folder **Stopping conditions > Residuals**.

Create a stopping criterion:

Step	Actions
1	Open the context menu of the element **Stopping conditions > User values**. Select **Create** there.
2	New element **Stop criterion #N** appears in the project tree in the folder **Stopping conditions > User values**. If the **Object** parameter in the **Properties** window is not specified, in the line of the element a warning symbol ("**!**") will appear: 

Copying a stopping criterion:

Step	Actions
1	Open the context menu of the element **Stop criterion #N** to be copied, and select **Copy**.
2	A new element **Stop criterion** will appear in the project tree in the folder **Stopping conditions > User values**. Element parameters will be same as those of the copied item **Stop criterion**.

Deleting a stopping criterion:

Step	Actions
1	Open the context menu of the element **Stop criterion** to be deleted and select **Delete**.
2	**Stop criterion #N** element is removed from the project tree.

Change the settings for a stopping criterion:

Step	Actions
1	Select from the project tree to edit element **Stop criterion**.
2	Adjust the values of the parameters in the **Properties** window, and then click **Apply**.

8.2.3.22 Operations with Materials

Some of the operations include the following materials:

- creation of material
- copying material
- removing material
- editing material
- assignment of material surfaces of facets, which put certain boundary conditions

Creating a Material

Step	Actions
1	Open the context menu on the folder line **Materials** and select **Create**.
2	Created material is added to the folder **Materials**.

Copying a Material

Step	Actions
1	Select the row of the material in the folder **Materials**, open the context menu and select **Copy**. Created a copy of the material is added to the folder **Materials**.
2	Select the added material, and specify the desired options in the **Properties** window.

Removing a Material

Step	Actions
1	If the removed material included in other elements, pre exclude it from all the elements.
2	Open the context menu of the element **Material #N**, to delete, and select **Delete**. **Material #N** element is removed from the project tree of folders **Materials**.

Changing properties of a Material

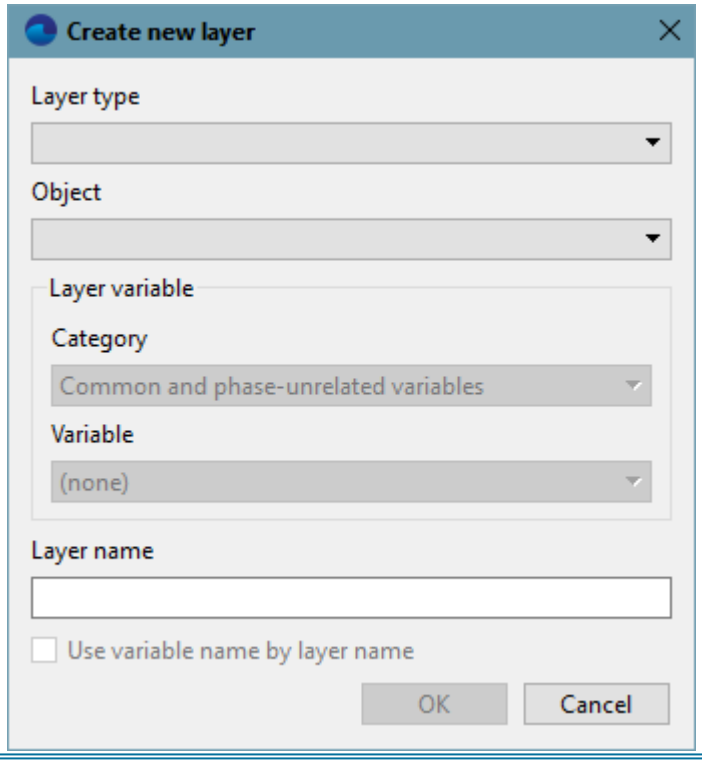
Step	Actions
1	Select from the project tree to edit element **Material #N**.
2	Adjust the values of the parameters in the **Properties** window, and then click **Apply**.

Appointment a Material to surfaces of facets, which put certain boundary conditions

Step	Actions
1	Select in the Project Tree tab **Postprocessor** element **Boundary conditions**.
2	Select the desired parameters of the **material** in the **Properties** window, and then click **Apply**.

8.2.3.23 Operations with Layers

Creating a Layer

Step	Actions
1	Open the context menu of the **Layers** folder, and select there the **Create** command. The **Create new layer** dialog box will open: 
2	In the **Layer type** field select type of the **Layer** (for example, **Vectors**, **Color contours**, **Streamlines**, etc.).
3	In the **Object** field select the **Object**, on which the new **Layer** will be built.
4	If selection of a **Variable** is applicable for the new **Layer**, do selections in fields **Category** and **Variable**. Options for selection the **Category** are: **Common and phase-unrelated variables | Variables of phase "*Phase_name*" | User variables**. For **Vectors** layers the program will automatically prompt you to select the **Velocity** variable (you can select another variable if required). You can also specify the **Variable** later, after the new **Layer** is created.
5	In the **Layer name** you can specify a name of the **Layer** instead of the default name. When an alternative name of the **Layer** is being entered, the "**×**" symbol will appear on the right; clicking it restores the default name. If you select the **Use variable name by layer name** checkbox, the program will prompt the **Layer**'s name based on name of the **Variable**, by which the **Layer** will be built.
6	Click the **OK** screen button. The new **Layer** will be created and added into the **Layers** folder. If no **Variable** is selected in the **Layer**, then the the warning symbol "**!**" will be displayed in the project tree near presentation of the **Layer**:

Step	Actions
	 Plot along curve #1 (Plane #0)
7	In the project tree select the new **Layer** and specify its other parameters in the **Properties** window.

Copying a Layer

Step	Actions
1	In the project tree open the context menu of the **Layer**, which is to be copied, and select the **Copy** command. The just created copy of the **Layer** will appear in the **Layers** folder.
2	Select the added layer and set the required parameters in the **Properties** window.

Removing a Layer

Step	Actions
1	Open the context menu of the element **Layer** to be deleted and select **Delete**.
2	Element **Layer** is removed from the project tree folder layers.

Changing parameters of a Layer

Step	Actions
1	Select from the project tree to edit the element **Layer**.
2	Adjust the values of the parameters in the **Properties** window, and then click **Apply**.

Displaying layer parameters in the «Info» window

Step	Actions
1	Select the item in the project tree **Layer**, and then click  in the toolbar **Work modes**. The **Info** informational window will open.
2	To close the **Info** window, click the icon in the upper right corner of the window.

Hiding or displaying layers

Step	Actions
1	Do one of the following: • open the context menu on the element **Layer** to be hiding, and install (or remove) a mark in the **Hide** option • *or* select the item in the project tree **Layer**, in the **Properties** window, select **No** (**Yes**) for the parameter **is displayed**, and then click **Apply**.
2	If the layer is configured as hidden, it will be displayed only when selected in the project tree.

Clipping or cancel clipping layer

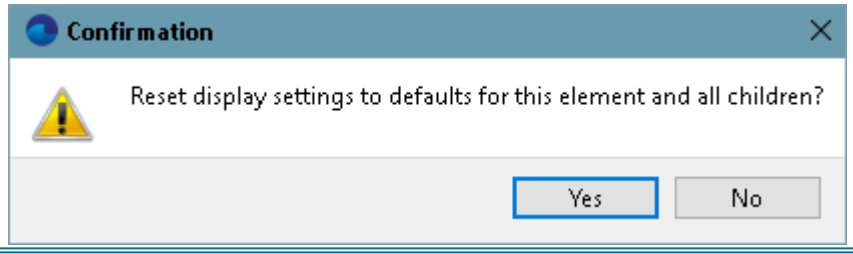
Step	Actions
1	Do one of the following:

Step	Actions
	• *either* open the context menu on the element **Layer** for which is given the ability to be cut off and the install (or remove) a mark in the Apply clipping option • *or* select the item in the project tree **Layer**, in the **Properties** window, select **Yes** (**No**) for the parameter **is cut off**, and then click **Apply**.
2	Layer will acquire the relevant property.

Lighting or cancel coverage layer

Step	Actions
1	Do one of the following: • open the context menu on the element **Layer** for which the adjusted lighting, and install (or remove) a mark in the **Apply lighting** option • *or* select the item in the project tree **Layer**, in the **Properties** window, select **Yes** (**No**) for the parameter **is lighted**,and then click **Apply**.
2	Layer will acquire the relevant property.

Assigning a default value for a component's layer «Solids»

Step	Actions
1	Right-click the line component of the layer **Solids** and select **Reset display settings**. The confirmation dialog is displayed (**Reset display settings to defaults for this element and all children?**): 
2	Click **Yes**. Visualization parameters of the **Layer**'s component and its child elements will be the same as visualization parameters specified for the parent component.

Create an Emitter for streamlines

Step	Actions
1	Right-click the layer **Streamlines** and select **Create emitter**.
2	New element **Emitter #N for Streamlines** is added to the folder layer **Streamlines**.

Copy an Emitter for the streamlines

Step	Actions
1	Select the line item **Emitter #N for Streamlines** in the folder layer **Streamlines**, open the context menu and select **Copy**. Created copy is added to the folder layer **Streamlines**.
2	Select the added element and specify the necessary settings in the **Properties** window.

Removal of an Emitter for the streamlines

Step	Actions
1	If the deleted element is included in other elements, pre exclude it from all the elements.
2	Open the context menu of the element **Emitter #N for Streamlines** to be deleted and select **Delete**. Element **Emitter #N for Streamlines** remove from layer **Streamlines**.

8.2.4 Analysis of project calculation's results

Operation analysis of calculation results of the project include:

- Operation analysis of the results of calculation of the project received the final step in time
- Operation analysis of calculation results stored in a number of time steps
- sequence analysis of the stored data to render layers (without regard to the solver)

See also: section Management of capturing images.

8.2.4.1 Analysis of calculation's results received on the last step

Analyzing the results of the calculation operations, obtaining in the last time step, include:

- displaying the calculated results as a distribution of calculated variables in the calculation region (to the extent on the surface or along the line)
- quantitative displaying of the integral characteristics
- display, depending on the calculation's time of the variables or characteristics, which are included as criteria for stopping the calculation
- display the values of variables in the center of the computational cell

Displaying calculation results in a distribution of calculated variables in the computational domain:

Step	Actions
1	Connect the project to **Solver**.
2	Create the required **Layer** elements with calculated data and specify their visualization settings.

Displaying quantitative integral characteristics:

Step	Actions
1	Connect the project to **Solver**.
2	Create in tab **Postprocessor** the required elements **Characteristics** and set their parameters.
3	Select **Characteristics** elements in the project tree and click *i* in the toolbar **Work modes**. Components of the **Characteristics** are displayed in the **Info** window.

Displaying depending on the calculation's time of the variables or characteristics, which are included as criteria for stopping the calculation:

Step	Actions
1	Connect the project to **Solver**.
2	Click the **Chart** tab window **Monitor** and adjust the display time dependencies of the relevant variables or characteristics.

Displaying values of variables in the center of the computational cell:

Step	Actions
1	Connect the project to **Solver**.
2	Create a layer **Cell debug** and set the number of display cells.
3	Select the project tree line created layer and click in the toolbar **Work modes**. Values of the variables are displayed in the **Info**.

8.2.4.2 Analysis of project calculation's results received on many steps

Operation analysis of calculation results stored in a number of time steps, include the following:

- mode setting to automatically save the data in the calculation
- loading into **Pre-Postprocessor** stored intermediate result of the calculation
- automatic display of a sequence of layers with the calculation results stored in a number of time steps

Loading in Pre-Postprocessorproject saved intermediate calculation result

Loading in **Pre-Postprocessor** saved intermediate results of the calculation is possible, if the project is downloaded to **Solver**.

Loading the first stored result of the calculation

Click (**Load first step**) in the **Navigation** toolbar.

Loading the last saved result of the calculation

Click (**Load last step**) in the **Navigation** toolbar.

Loading next stored result of the calculation

Click (**Load next step**) in the **Navigation** toolbar.

Loading the previously stored result of the calculation

Click (**Load previous step**) in the **Navigation** toolbar.

Download result of the calculation stored at a certain time step

Click (**Selecting the calculation history step to download**) in the **Navigation** toolbar. In the Step selection dialog box set the desired time step.

Automatic display of a sequence of layers with the calculation results stored in a number of time steps

Step	Actions
1	Make sure that the **Pre-Postprocessor** is connected to **Solver**; if it is not connected, load the project on **Solver**.
2	Create required **Layer** elements with the calculated data and set the display of calculation results.
3	Specify the start watching one of the following ways: • to go to the first preset the recording, click in the **Navigation** toolbar • to go to the saved record at a certain time step: a) click in the **Navigation** toolbar. b) in the dialog box **Step selection**, which opens, select the starting time step.

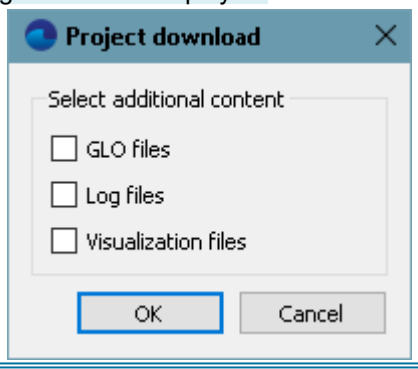
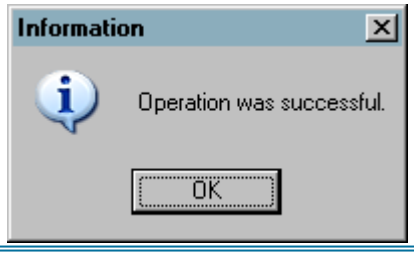
Step	Actions
4	To save images from the **View** window for each time step in a sequence of image files, click the (**Start capturing an image sequence**) button in the **Capture** toolbar and specify the file format in which the image will be saved.
5	Run the sequential scan by clicking (**Start automatic sequential transition through the steps of the calculation history**) in the **Navigation** toolbar.
6	If it is necessary to interrupt the playback, click (**Stop automatic sequential transition through the steps of the calculation history**) in the **Navigation** toolbar.
7	Upon completion of viewing the whole sequence of the saved data to be stored images **View** window by clicking (**Complete capture sequence**) in the **Capture** toolbar.

8.2.4.3 Analysis of saved layer visualization data (with no connection to Solver)

Operation sequence analysis of stored data to visualize the layers include the following:

- auto-save mode setting data for imaging layers during calculation
- copying the data from the server to render to the client part of the project
- loading in **Pre-Postprocessor** for visualizing stored in the calculation
- automatic display of the sequence data for visualization, stored in a number of time steps.

Copying data from the server to render to the client part of the project

Copying data from the server to render to the client part of the project	
Step	**Actions**
1	Select from the main menu **File > Download additional files**. The **Project download** dialog box will be displayed: 
2	Select the **Visualization files** checkbox, and then click **OK**. A message appears **Operation was successful**: 
3	Click **OK**.

Loading in Pre-Postprocessor for visualizing stored in the calculation

Download to **Pre-Postprocessor** can only data for the visualization layer files **(*.fvvis)**, stored in the course of computation and copied from the server to the client part of the project.

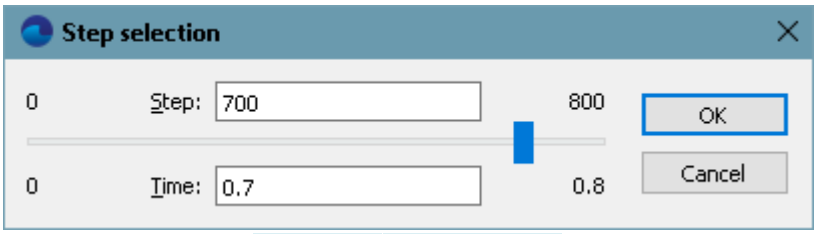
Loading in Pre-Postprocessor for visualizing stored in the calculation:	
Step	**Actions**
1	Select from the main menu **File > Load data for visualization**. The standard **Open** dialog box of the operating system to access the files (**Open**).
2	Select a file `*.fvvis` and then click **Open**. In the **View** window displays the saved layer.

Automatic display of sequence data for visualization, stored in a number of time steps

Conditions under which it is possible to automatically display sequence data for visualization, stored in a number of time steps.

Automatic display of sequence data for visualization, stored in a number of time steps:	
Step	**Actions**
1	Click the (**Autoload visualization files**) button in the **Navigation** toolbar.
2	Specify the start watching one of the following ways: • to go to the first preset the recording, click the button in the **Navigation** toolbar. • to go to the saved record at a certain time step: a) click the button in the **Navigation** toolbar. b) in the dialog box **Step selection**, which opens, select the starting time step.
3	To save images from the **View** window for each time step in a sequence of image files, click the (**Start capturing an image sequence**) button in the **Capture** toolbar and specify the file format in which the image will be saved.
4	Run the playback by clicking the (**Start automatic sequential transition through the steps of the calculation history**) button in the **Navigation** toolbar.
5	If it is necessary to stop the playback, click the (**Stop automatic sequential transition through the steps of the calculation history**) button in the **Navigation** toolbar.
6	Upon completion of viewing the whole sequence of the saved data to be stored images from the **View** window by clicking on the (**Complete capture sequence**) button in the **Capture** toolbar.

8.2.4.4 Dialog window «Step selection»



Dialog box **Step selection**

Dialog box **Step selection** is designed to perform the following actions:
- indication of the time step in the sequence of data stored in the calculation;
- Upload a **Pre-Postprocessor** data stored in the selected step.

The **Step selection** dialog box is opened by pressing the button in the **Navigation** toolbar.

Selecting a desired record is done by one of the following ways:
- moving the slider on the scale
- specifying the number for the time step in the **Step** field
- specifying the calculation's time of the **Time** field

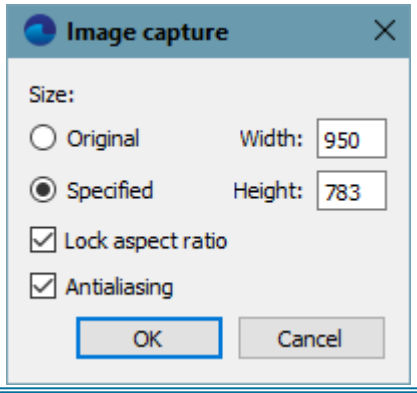
Then click **OK** to go to the required step.

8.2.4.5 Controlling visual capture

In the control operation by the image capture include:

- capture a single image
- capture a series of images
- suspension or resumption of image capture
- completion of image acquisition

Capture a single image

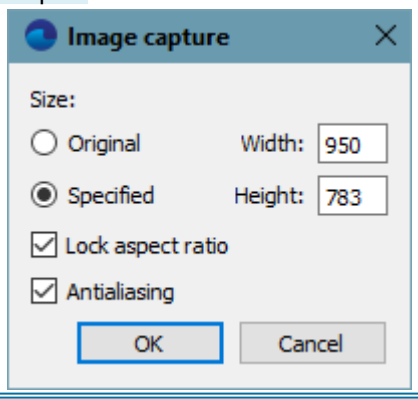
Step	Actions
1	Click the button (**Save the contents of the graphics window to an image file**) in the **Capture** toolbar. The **Image capture** dialog box will open: 
2	Specify in the **Image Capture** settings (size, elimination of aliasing). Click **OK**. (Description of the elements of this window, see in a separate section below)
3	Click **OK**. Displays the standard Windows operating system to access the files (**SaveAs**). Point it to the location and type a name for the file, which will be recorded. Then click the **Save** button.

Starting capture a series of images

A series of images captured in files in the following cases:

- in the calculation - at every time step
- sequential viewing of previously saved calculation data - after the display the next time step;
- sequential viewing of stored data for visualization - after the display the next time step.

Perform the following steps:

Step	Actions
1	Click the (**Start capturing an image sequence**) button in the **Capture** toolbar. The **Image capture** dialog will open: 

Step	Actions
2	Specify in the **Image capture** settings (size, elimination of aliasing). Click **OK** (description of the elements of this window, see in a separate section below).
3	This opens the standard Windows operating system to access the files **(Save Image)**. Point it to the location, type, and name prefix for the files that will be written to the image. Then click the **Save** button. Images will be written to the file at each update window **View**. The file names are formed as follows: `(Prefix) _00000.`(filetype) `(Prefix) _00001.`(filetype) `(Prefix) _00002.`(filetype) etc.

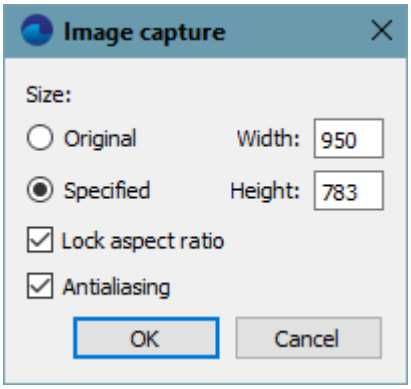
Pause or resume the image acquisition

Step	Actions
1	Click the button (**Pause capturing an image sequence**) in the **Capture** toolbar.

Completion of the image acquisition

Step	Actions
1	Click the button (**Finish capturing an image sequence**) in the **Capture** toolbar.

Dialog box "Image capture"



The **Image capture** dialog box sets the options for saving a single image or series of images (after clicking on the button or in the **Capture** toolbar.). It contains the following elements:

Element	Description
Size:	
Original	The saved image size is the size of the **View** window.
Specified	The saved image size is set by the user.
Width	Image width
Height	Image height
Lock aspect ratio	Keep aspect ratio of the original image with respect to the window View. When this option is enabled, the user can specify only one image size (either width or height), and the second size is inserted automatically by the program in accordance with the original proportion.
Antialiasing	Eliminate jagged lines in the stores images.

File formats

Images may be saved in the following file formats:

- **`bmp`** (bitmap storage format developed by *Microsoft*, see http://en.wikipedia.org/wiki/BMP_file_format)
- **`png`** (image compression format portable network graphics)

8.2.5 Saving and loading the settings

Saving the configuration to a file and load the configuration from a file

Your settings can be saved to the file format **`*.cfg`** and loaded from the file format **`*.cfg`**.

Saving settings in a file

Step	Actions
1	Select from the main menu the **File > Save preferences as** command.
2	In the dialog box that opens, specify the folder and the file name where the configuration is to be saved.

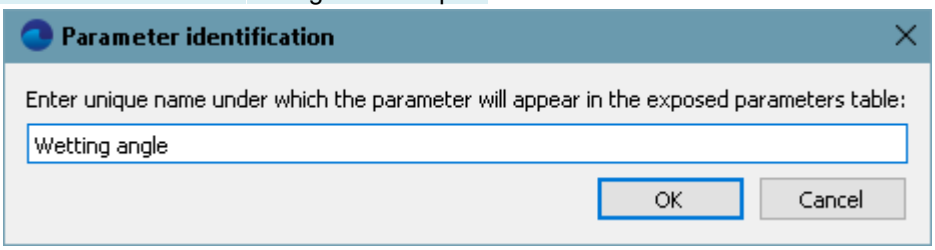
Loading settings from a file

Step	Actions
1	Select from the main menu the **File> Load preferences** command.
2	In the **Open** dialog box that opens select a directory and name of the file from which the settings will be loaded.

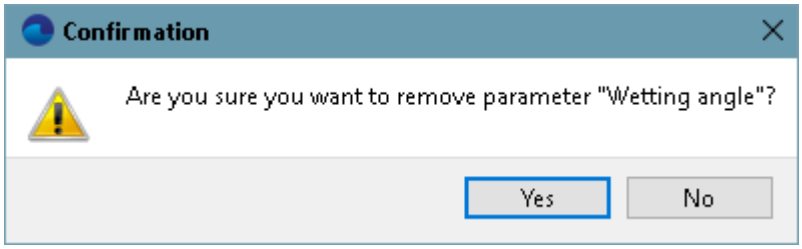
8.2.6 Operations with external parameters and exported results

Commands **Add to exposed parameters table** and **Remove from exposed parameters table** are not available for parameters, which cannot be edited at all or at the current selection of other parameters. If necessary, click the **Apply** button in the **Properties** window to make these commands available.

Declaring an external parameter

Step	Actions
1	In the **Properties** window, open context menu from the the line of the parameter and select the **Add to exposed parameters table** command. The **Parameter identification** dialog box will open:  This window is defined by the parameter name, which will be used to identify it in the table of the external parameters. The default name of the parameter of the **Properties** window.
2	Enter the name under which the parameter will appear in the list of external parameters, and then click **OK**. The parameter will be: • displayed on a green background in the **Properties** window • added to the list in the **Exposed parameters window**.

Cancel parameter declaration outside

Step	Actions
1	Do one of the following: • in the **Exposed parameters window** select the option from the list and click the **Delete** button; • in the **Properties** window right-click the parameter and select the **Remove from exposed parameters table** command.
2	A dialog box will open requesting you to confirm your decision to remove the external parameter ("**Are you sure you want to remove parameter ... ?**"):  Click **Yes** to confirm your decision.
3	The parameter will be removed from the list in the **Exposed parameters window**.

Viewing the declared external:

Step	Actions
1	In the **Exposed parameters window** select the parameter in the list and click the **Select** button.

Step	Actions
2	Focus moves to the **Preprocessor** tab to an item in the project tree, the parameter which is an external item that is highlighted in the list. In the **Properties** window shows the value of the selected external parameter.

Assignment (cancel) mode export results to a file:

Step	Actions
1	In the Exposed parameters window set (or clear) the **Generate results table** checkbox.

8.2.7 Operations with backup

In the event of a computer failure during writing the project's files (for example, during saving the project), and loss the data from the last time step, you can try to recover the project manually from one of the previous time steps, at which a backup was made.

Project's data recovery from backup files	
Step	**Actions**
1	Make copies of folders with the server part and the client part of the project, including all files in them. You will require these copies in the case if the manual backup fails and you would have to contact to the technical support.
2	In the client and server parts of the project, remove the files with extensions **`.fvmind`**, **`.fvgrid`***), .fvdata, .fvgobj, **`.fvstat`**, which were corrupted during the saving.
3	In the client and server parts of the project get the files with the extension **`*.backup`** and copy the same files, but without the extension**`*`** **`.backup`**(ie delete characters in the file extension **`.backup`**). Examine, which files with extension **`.backup`** locate in the client and in the server parts of the project, find a pair of **`backup`**-files with the latest same dates. If you find such files, rename them by removing symbols **`.backup`** from the extensions, after previous removing the existing files with target names.
4	Try to load the project on **Solver** or open the project in **Pre-Postprocessor**. If you fail this, contact the technical support service and provide there the folders and project files that were saved on Step 1.

*) Before version *8.3.00*, files **`fvmesh`** were used instead of files **`fvgrid`**.

8.2.8 Work of Pre-Postprocessor in the read-only mode

When there are no access to a license or the client part of the project is write-protected, **Pre-Postprocessor** can operate in the read-only mode.

Operation in the read-only mode has the following specifics:

- Commands **Save**, **Save with selection**, **Save copy** and **Save as** in the Main menu are not available.
- **Pre-Postprocessor** can connect to **Solver**. If **Solver** has access to a license, **Pre-Postprocessor** can run the project and visualize the simulation on the condition that the client and the server parts of the project match together.
- Changes made in the project are not saved on exit from **Pre-Postprocessor**, which operates in the read-only mode.

Pre-Postprocessor will not escape the read-only mode even after fixing the cause of entering this mode. All the project's not saved data will be lost on exiting the project.

Take this into consideration and make a responsible choice in the **License loss** dialog box (**Try to recover** the connection or **Edit preferences** might be better alternatives).

8.2.9 Controlling the project's calculation

See sections:

- Operations with Solver and Solver-Agent
- Starting and stopping project's computation
- Project computation progress control (operations)

8.2.9.1 Operations with Solver and Solver-Agent

Inter-operation between **Pre-Processor** and **Solver** and **Solver-Agent** allows you to:

1. Run the project for computation
2. Obtain data for visualization results of the computation

Operations with **Solver** include:

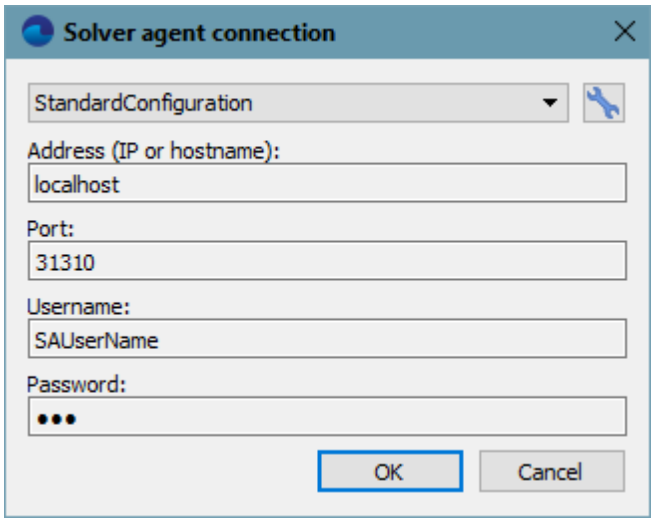
- user authentication on Solver-Agent
- starting a new Solver
- connecting a project to the selected Solver (see also section Dialog box "Select solver")
- disconnecting Pre-Postprocessor from Solver – this is useful when carrying out a computation that does not require constant monitoring, as in this case:
  - **Solver** does not calculate parameters of specified layers and does not send them to **Pre-Postprocessor**
  - **Pre-Postprocessor** does not display the new state of the layers
- terminating operation of the Solver

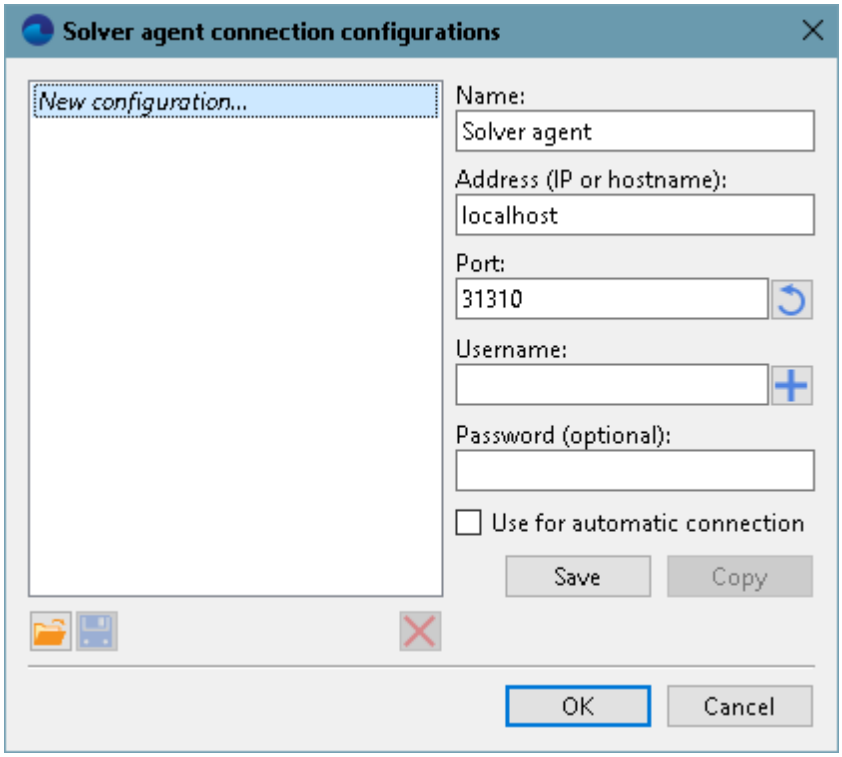
See also sections:

- Dialog box "Select solver"
- Client and server parts don't match, so their synchronization is required

8.2.9.1.1 Authentication on Solver-Agent from Pre-Postprocessor

Authentication on **Solver-Agent** from **Pre-Postprocessor** might be done automatically at start of **Pre-Postprocessor**, when the default **Configuration** for connection to **Solver-Agent** is is tuned correctly. In this case the steps listed below are done automatically and you don't have to follow them.

Step	Actions
1	Click the button (**Solver agent log in**) in the **Network** toolbar. The **Solver agent connection** dialog box will open (if at least one connection configuration has been created):  Select a configuration and click **OK**.
2 *(optional)*	If no configuration has been created yet, an empty **Solver agent connection configurations** dialog box will open where you have to create at least one configuration:

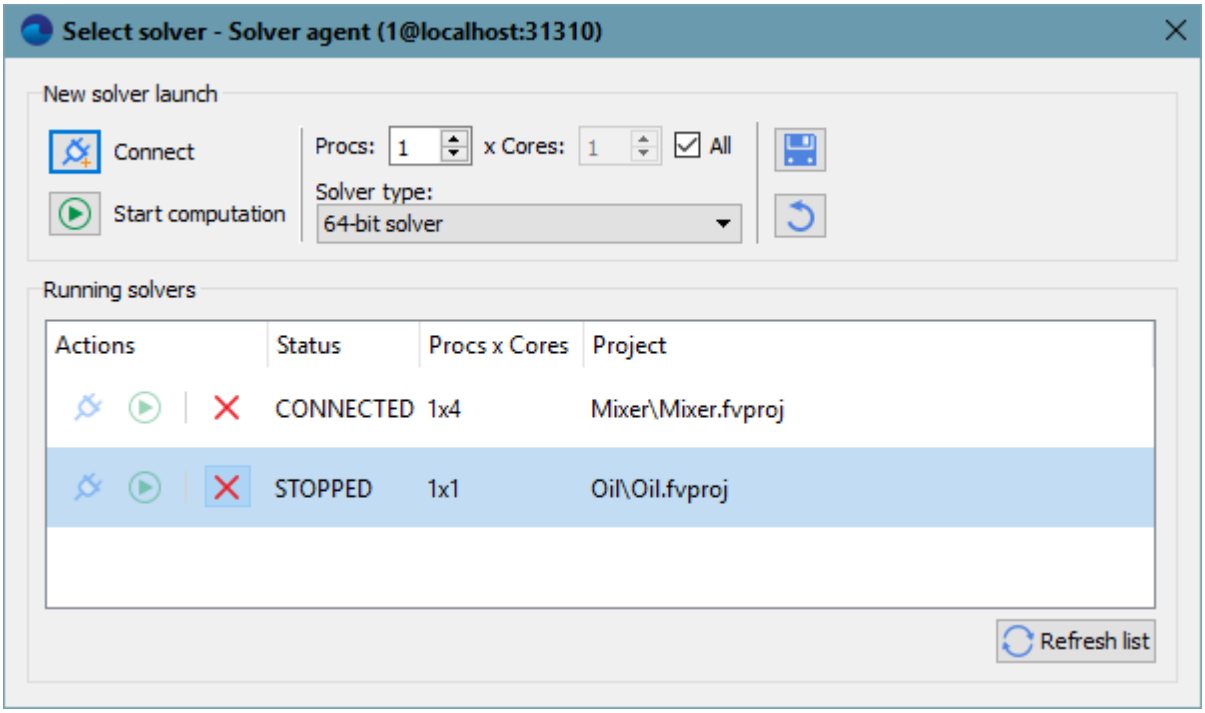
Step	Actions
	 If you have to create a new **Solver-Agent** user, click the  **(Register a new user)** screen button to open the **User registration** dialog box. After filling the **Solver agent connection configurations** dialog box with data, click **OK** and, in the **Solver agent connection** dialog box, which again opens, select a configuration and click **OK**.

See details in the section Connection to Solver-Agent and user authentication on Solver-Agent.

8.2.9.1.2 Dialog box "Select solver"

The **Select solver** dialog box is used to control the interaction between **Pre-Postprocessor** and **Solver agent**. It allows you to perform the following steps:

- Starting a **Solver**
- Connecting to a **Solver**, which has been started and is running
- Terminating a **Solver**



The **Select solver** dialog box

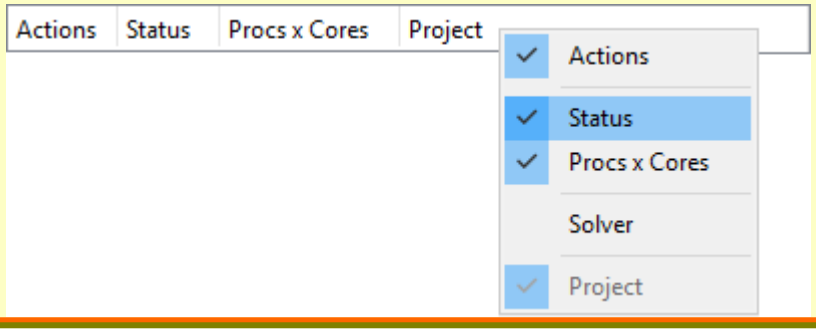
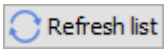
The header of this dialog box contains the name and parameters of the used **Configuration** for connection to **Solver-Agent**:

The **Select solver** dialog box is opened when you click the button (**Open solver selection window**) in the **Network** toolbar.

This dialog box contains the list of running solvers and user interface elements listed below:

Interface elements of the dialog box "Select solver"	
Element	**Description**
New solver launch *(group of interface elements)*	This group of interface elements is used to start a new **Solver** with specified parameters, and, optionally, to run the current project on this **Solver**. See descriptions below.
Connect *(screen button)*	Starting a new **Solver** with specified parameters and connecting the current project to it. After this the **Select solver** dialog box will automatically close.
Start computation *(screen button)*	Starting a new **Solver** with specified parameters, connecting the current project to it, and starting computation of the current project on this **Solver**. After this the **Select solver** dialog box will automatically close.
Procs *(number)*	The number of processors on which the new **Solver** will run in multiprocessor mode
Cores *(number)*	Number of cores on each processor used by the new **Solver** at the calculation. To run a **Solver** in multiprocessor mode, you have to install and configure MPI, as well as to configure Solver-Agent.
All *(checkbox)*	☐ – the computation will be performed using the specified number of cores per processor ☑ – the computation will be performed using all cores on each processor

Interface elements of the dialog box "Select solver"	
Element	**Description**
Solver type *(drop-down list)*	This is selection of a command line, this specifies the type of solver. This field is available for selection when the configuration file of Solver-Agent (FvSolverAgent.cfg) contains more than one command line of the appropriate type (for a single-processor or multiprocessor mode). Each option in this list consist text of the comment (the text from symbol ";" to the end of the line) from an appropriate command line in the configuration file of Solver-Agent (FvSolverAgent.cfg). Examples of options selectable from the **Solver type** list: • **64-bit solver** • **64-bit Microsoft MPI solver**
 **Save as default parameters for creating solver**	For the future use, save as default ones, the parameters, which are specified in fields **Procs**, **Cores**, **All**, and **Solverf type**. To load these saved parameters, apply the button  (**Load default parameters**), see below.
 **Load default parameters**	Load the default parameters for the **Solver**, which will be started. These parameters were saved before by clicking the button  (**Save as default parameters for creating solver**), see above.
Running solvers *(table)*	This part of the dialog box allows you to do actions with existing running **Solvers**. The table of running **Solvers** has the following columns: • **Actions** – this column contains buttons each of them can be either active (available for clicking, displayed in full color) or inactive (unavailable for clicking, faded): ○  (**Connect**) connects **Pre-Postprocessor** to the selected **Solver**. ○  (**Start computation**) runs the current project on the selected **Solver**. ○  (**Terminate**) terminates (shuts down) the selected **Solver**. These screen buttons are also available from the context menu that opens at right-clicking on the line. Buttons  and  might be inactive because of a project cannot be loaded on more then one **Solver**. • **Status** displays the status of the **Solver**: ○ **EMPTY** – no project is loaded on the **Solver** ○ **STOPPED** – a project is loaded on the **Solver** and the calculation is not running and no controlling client (**Pre-Postprocessor**) is connected to the **Solver**. ○ **SOLVING** – a project is loaded on the **Solver** and the calculation is running ○ **OCCUPIED** – a project is loaded on the **Solver** and some other controlling client (**Pre-Postprocessor**) is connected to the **Solver**. ○ **CONNECTED** – a project is loaded on the **Solver** and this **Pre-Postprocessor** is connected to the **Solver**. ○ **INACCESSIBLE** – the **Solver** is not available for connection by **Pre-Postprocessor** [1)] ○ **WAITS FOR DECIMATION** – decimation a project from **Terminal** from the moment of clicking the **Save** button in the **Non-steady-state steps decimation** dialog box until finishing the decimation. See Terminal's menu Projects. ○ **DECIMATION ERROR** – unsuccessful attempt of decimation a project from **Terminal** in situation when the project is connecting to a **Solver** that was opened from **Pre-Postprocessor**. • **Procs x Cores** displays the number of processors and cores per processor used by the **Solver** • **Solver** displays the **Solver**'s identifier. By default this column is not displayed. • **Project** displays the name of the project, which is loaded on the **Solver**.

Interface elements of the dialog box "Select solver"	
Element	**Description**
	You can select only one **Solver** from this list of **Solvers**. To discard your selection, either apply hot keys **Ctrl+Space**, or click by the mouse when the **Ctrl** key is pressed, or click on an empty area of the **Running solvers** list. If the current project, which is opened in this **Pre-Postprocessor**, is loaded on some of the running **Solvers**, then connection to another **Solver** is impossible. Displaying of any column, except the **Project** column, can be either enabled or disabled. To tune displaying the columns, right-click on the header of the list and select or unselect checkboxes in the context menu, which opens: 
 **Refresh list** *(screen button)*	This button updates information in the **Running solvers** table. You can also press the **F5** key on the keyboard for the same effect.

[1] Possible causes of the **INACCESSIBLE** status:
- A firewall blocks the **Solver**, **Pre-Postprocessor**, or **Retranslator**.
- The configuration file contains settings for **Retranslator** but **Retranslator** is not running.
- Versions of **Pre-Postprocessor** and **Solver** are different.
- There was an incorrect shut down of **Retranslator** or **Solver**.

How to run a new Solver

In the **New solver launch** group of interface elements:
- Specify parameters of the new **Solver** that will be started:
  - set the number of processors in the **Procs** field.
  - enable use of all cores (select the **All** checkbox), or specify the number of cores per processor (unselect the **All** checkbox and enter the number into the **Cores** field).
  - from the **Solver type** drop-down list, select the command line to run the **Solver**.
- Click the  **Connect** screen button.

How to run a new Solver and run computation of the current project on it:

In the **New solver launch** group of interface elements:
- Specify parameters of the new **Solver** that will be started:
  - set the number of processors in the **Procs** field.
  - enable use of all cores (select the **All** checkbox), or specify the number of cores per processor (unselect the **All** checkbox and enter the number into the **Cores** field).
  - from the **Solver type** drop-down list, select the command line to run the **Solver**.
- Click the  **Start computation** screen button.

How to connect to a running Solver:

In the **Running solvers** table:
- Select a line with the required **Solver**.

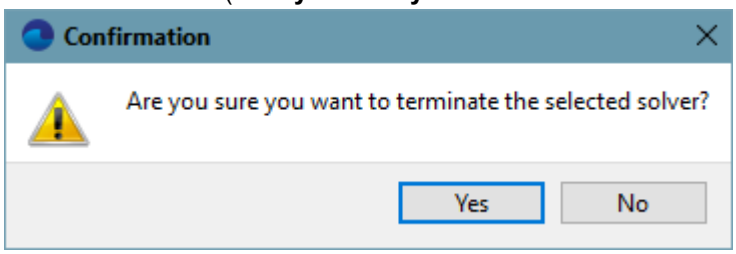
- In the **Actions** column click the (**Connect**) screen button. This button is also duplicated by the command **Connect** from the context menu of the line with the selected **Solver**.

If the project, which is to be loaded on a **Solver**, already exists in the Server directory, then synchronization of server and client parts of the project will be done at connecting to the **Solver**.

How to terminate operation of a running Solver

To terminate operation of a running **Solver**, follow these steps in the **Running solvers** table:

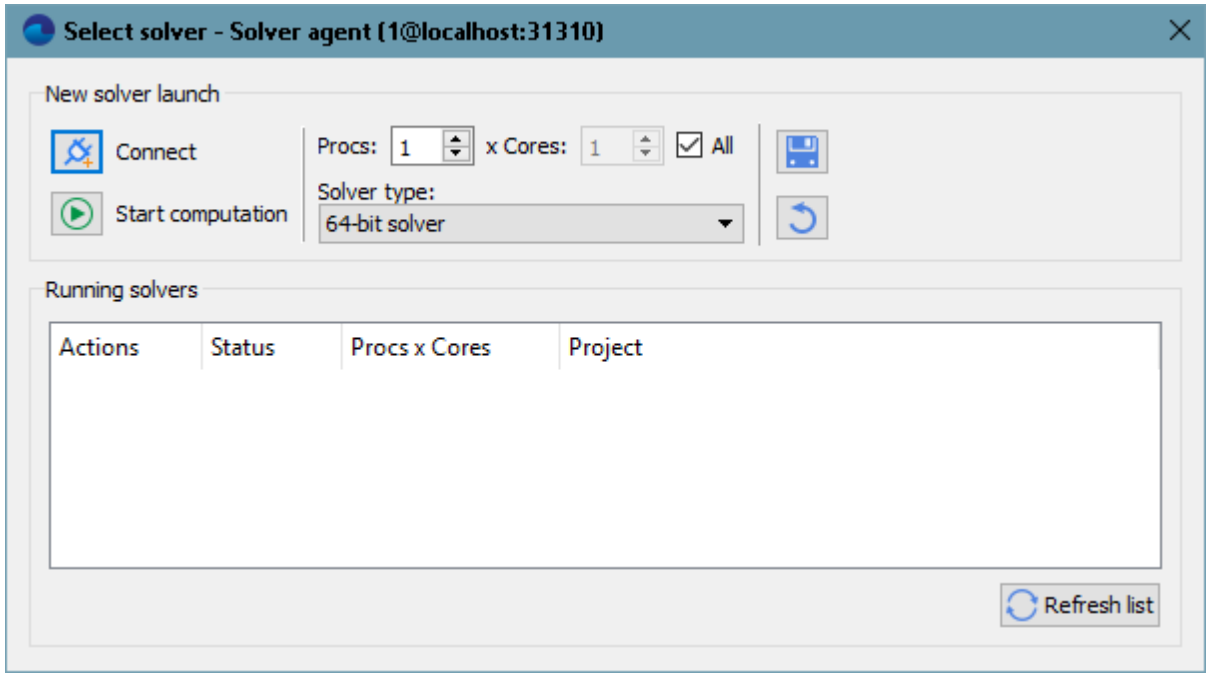
- Select a **Solver** in the **Running solvers** list.
- In the **Actions** column click the (**Terminate**) screen button. This button is also duplicated by the command **Terminate** from the context menu of the line with the selected **Solver**.
- The program will request a confirmation ("**Are you sure you want to terminate the selected solver?**"):



- Click **Yes**. The selected **Solver** will disappear from the **Running solvers** table.

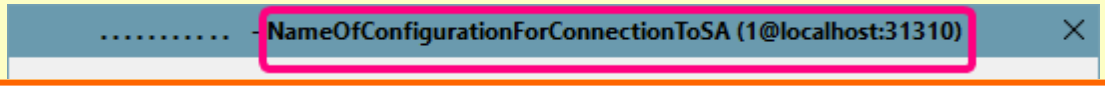
8.2.9.1.3 Starting a new Solver

Starting a new **Solver** is done from the **Select solver** dialog box.



The **Select solver** dialog box

The header of this dialog box contains the name and parameters of the used Configuration for connection to Solver-Agent:



To start a new **Solver**, follow the steps:

Step	Actions
1	Click the button (**Open solver selection window**) in the **Network** toolbar to open the **Select solver** dialog box.
2	In the **New solver launch** group of interface elements specify parameters of the new **Solver** that will be started: • set the number of processors in the **Procs** field. • enable use of all cores (select the **All** checkbox), or specify the number of cores per processor (unselect the **All** checkbox and enter the number into the **Cores** field). • from the **Solver type** drop-down list, select the command line to run the **Solver**. • click the **Connect** screen button. The new **Solver** will be displayed in the **Running solvers** list.

8.2.9.1.4 Connecting Pre-Postprocessor to Solver. Work with client and server parts of the project.

Pre-Postprocessor allows you to:
- prepare a project for computation
- start the computation
- analyze the obtained results

For both starting of the computation and analysis of the results, **Pre-Postprocessor** has to be connected to a **Solver**.

When connected to **Solver**, the program checks whether the server part of the current project exists. Search for the project's server part is done in **Server directory** (both server and client parts of the same project must have the same parameter `ProjID` in the file `.fvproj`).

In the sections below the following cases of connecting **Pre-Postprocessor** to **Solver** are considered:
- Client part of the project is opened in Pre-Postprocessor, while server part is absent (for example, if the project has been just created and has never been started for computation)
- Client part of the project is opened in Pre-Postprocessor, and server part is loaded on Solver
- Client part of the project is opened in Pre-Postprocessor. Server part exists but is not loaded on Solver.
- Client part of the project is lost or damaged, but server part of the project exists (obtaining the client part from the server part)
- Client and server parts don't match, so their synchronization is required
- Client and server contain parts of different projects

Two methods of work with client parts of projects

There can be recommended two methods of work with client parts of projects:

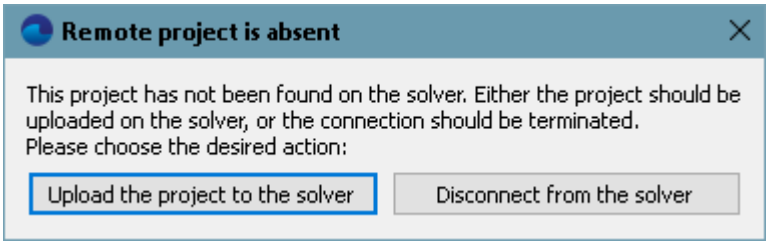
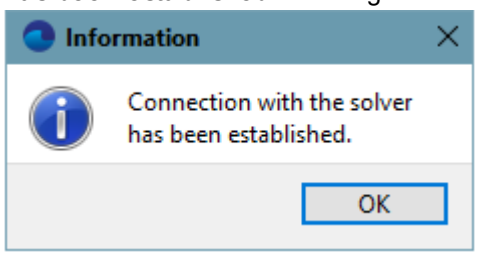
1. The user keeps all client parts of projects according to some system, which he or she applied. When the user wishes to analyze the results of the computation, he or she opens the client part of the project and connects to **Solver**. After this, **Solver** automatically loads the server part of the project. If the server part of the project was changed during the computation, the user updates the client part of the project.

or

2. The user does not keep client parts of projects but always download them from server parts of projects into a temporary directory, which is cleaned regularly. On the one side, this method require additional operations of loading the server projects to **Solver** (as it was described in the previous subsection in the paragraph *"The client part of the project is lost or damaged, but the server part of the project exists (obtaining the client part from the server part)"*). But on the other side, when you use this method you don't have to bother about keeping the system for storing your projects.

8.2.9.1.4.1 Client part of the project is opened in Pre-Postprocessor, while server part is absent (for example, if the project has been just created and has never been started for computation)

Step	Actions
1	In **Pre-Postprocessor**, open the client part of the project.
2	Connect to **Solver-Agent**: click the button (**Solver agent log in**) in the **Network** toolbar, and then select a connection **Configuration** in the **Solver agent connection** dialog box (it opens if at least one connection has been created).

Step	Actions
	If no **Configuration** has been created yet, create a **Configuration** as described in the section Connection to Solver-Agent and user authentication on Solver-Agent. If some **Configuration** is set as default one and contains correct settings, then connection to **Solver-Agent** will be done using this **Configuration** automatically at each start of **Pre-Postprocessor**.
3	Run the **Solver**: click the button (**Open solver selection window**) in the **Network** toolbar to open the **Select solver** dialog box; then run a new **Solver** or select an existing running **Solver** and then click the (**Connect**) button.
4	The **Remote project is absent** dialog box with the message "**This project has not been found on the solver. Either the project should be uploaded on the solver, or the connection should be terminated. Please chose the desired action.**" will open:  This message is displayed in the situation when in the server directory the program cannot find the server part, which matches to the client part that is opened in **Pre-Postprocessor**.
5	Click the button **Upload the project to the solver** to copy the client part of the project into the server directory. The program will create the server part of the project based on the client part, and then automatically upload the server part to **Solver**. Clicking **Disconnect from the solver** will cancel the attempt of connecting **Pre-Postprocessor** to the selected **Solver**.
6	After successful loading the project to **Solver** and connecting **Pre-Postprocessor** to **Solver**, the "**Connection with the solver has been established**" message will be displayed:  Click **OK**.
7	Make changes in the project, if required. Then start the computation of the project clicking on the button (**Start computation**) in the **Network** toolbar.

8.2.9.1.4.2 Client part of the project is opened in Pre-Postprocessor, and server part is loaded on Solver

This case is typical for situations when computation has been started on **Solver** while **Pre-Postprocessor** is disconnected from **Solver** to save computational resources (do not waste the resources for building visualization layers).

Solver with an uploaded project can be in either of one of two states: **STOPPED** or **SOLVING**.

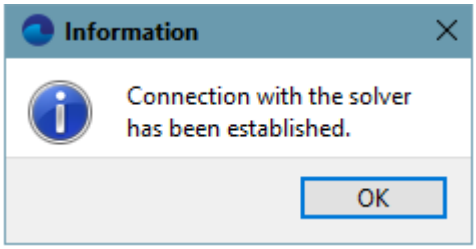
If the **Solver**'s status is **SOLVING**, then some difficulties can arise at connection to a working **Solver** (appropriate messages will be displayed, for example, informing that long time is required for the connection (it is necessary to wait until **Solver** finishes calculation of the current step), and multiple **Resolving a version conflict** messages might be displayed. To avoid or fix this situation, it is recommended to stop **Solver** before connection to it, to change its status to **STOPPED** (it should be done before the connection using **Terminal**).

Step	Actions
1	In **Pre-Postprocessor**, open the client part of the project.
2	Connect to **Solver-Agent**: click the button (**Solver agent log in**) in the **Network** toolbar, and then select a connection **Configuration** in the **Solver agent connection** dialog box (it opens if at least one connection has been created). If no **Configuration** has been created yet, create a **Configuration** as described in the section Connection to Solver-Agent and user authentication on Solver-Agent. If some **Configuration** is set as default one and contains correct settings, then connection to **Solver-Agent** will be done using this **Configuration** automatically at each start of **Pre-Postprocessor**.
3	There can be two cases when **Pre-Postprocessor** with the client part is trying to connect to **Solver**: 1) If versions of the server and client parts are the same (do not differ), the connection will be established immediately. A message about successful connection between **Pre-Postprocessor** and **Solver** will be displayed. 2) If versions of the server and client parts differ, then, depending on what is differ in the project's parts, a window will open: **Resolving a version conflict**, **Resolving simulation control files conflict**, or **Resolving a scene conflict**. Details of this situation are described below in the section Client and server parts don't match, so their synchronization is required.

8.2.9.1.4.3 Client part of the project is opened in Pre-Postprocessor. Server part exists but is not loaded on Solver.

When **Pre-Postprocessor**, with an opened client part of the project, connects to a **Solver**, to which no server part of any project is loaded, the following will take place automatically:

- **Solver** will find the server part of the project
- If the client and server parts of the project are the same (do not differ), **Solver** will automatically load the server part of the project and a message about successful connection between **Pre-Postprocessor** and **Solver** will be displayed ("**Connection with the solver has been established**"):



- If the client and server parts of the project differ, a message about conflict of versions will be displayed. See details in the section Client and server parts don't match, so their synchronization is required.

8.2.9.1.4.4 Client part of the project is lost or damaged, but server part of the project exists (obtaining the client part from the server part)

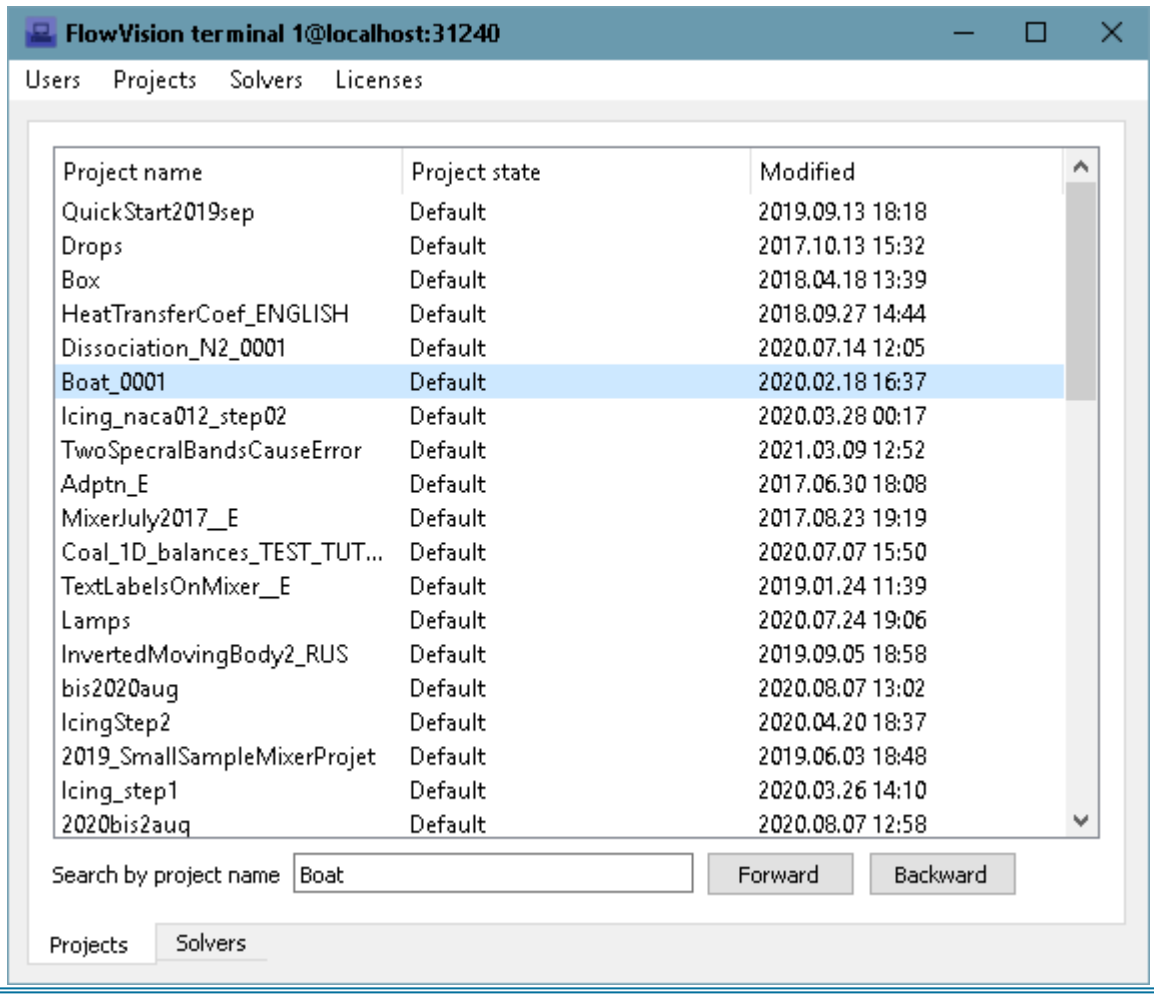
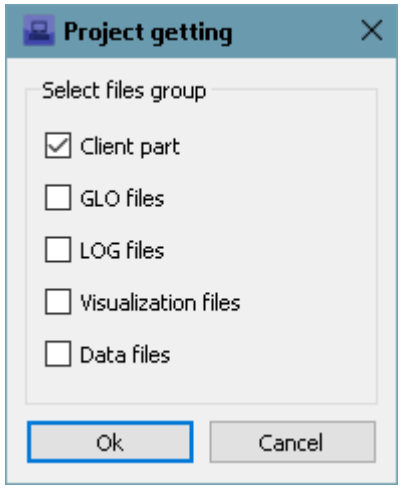
Sometimes it necessary to restore the client part of the project, for example, if:

- An undesirable change of the client part of the project has been done, so it is necessary to return to the state, in which the previous computation was done.
- Colleagues gave you the server part of the project only, without the client part.
- The client part of the project has been lost or damaged.

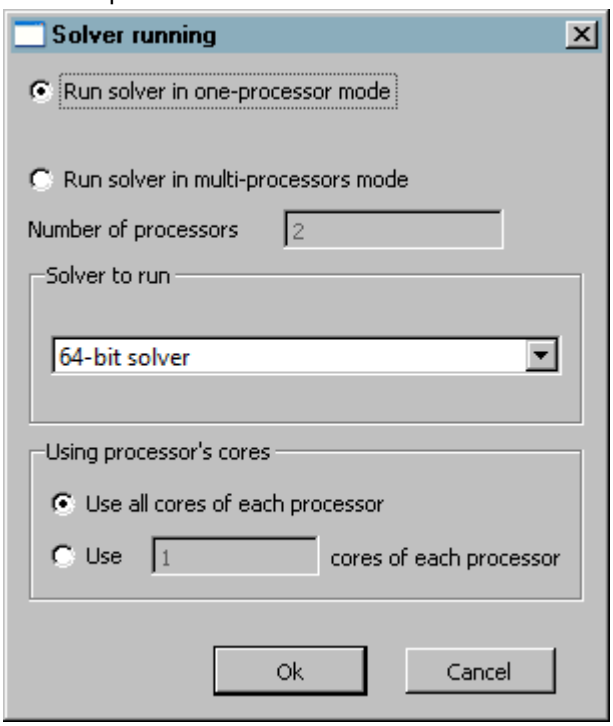
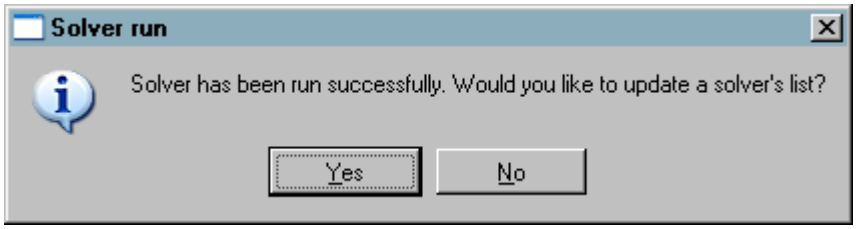
The client part of the project is always can be obtained from the saved server part of the project. We can recommend two methods:

- using **Terminal** without uploading the project to **Solver**
- using **Terminal** and **Pre-Postprocessor** without uploading the project to **Solver**

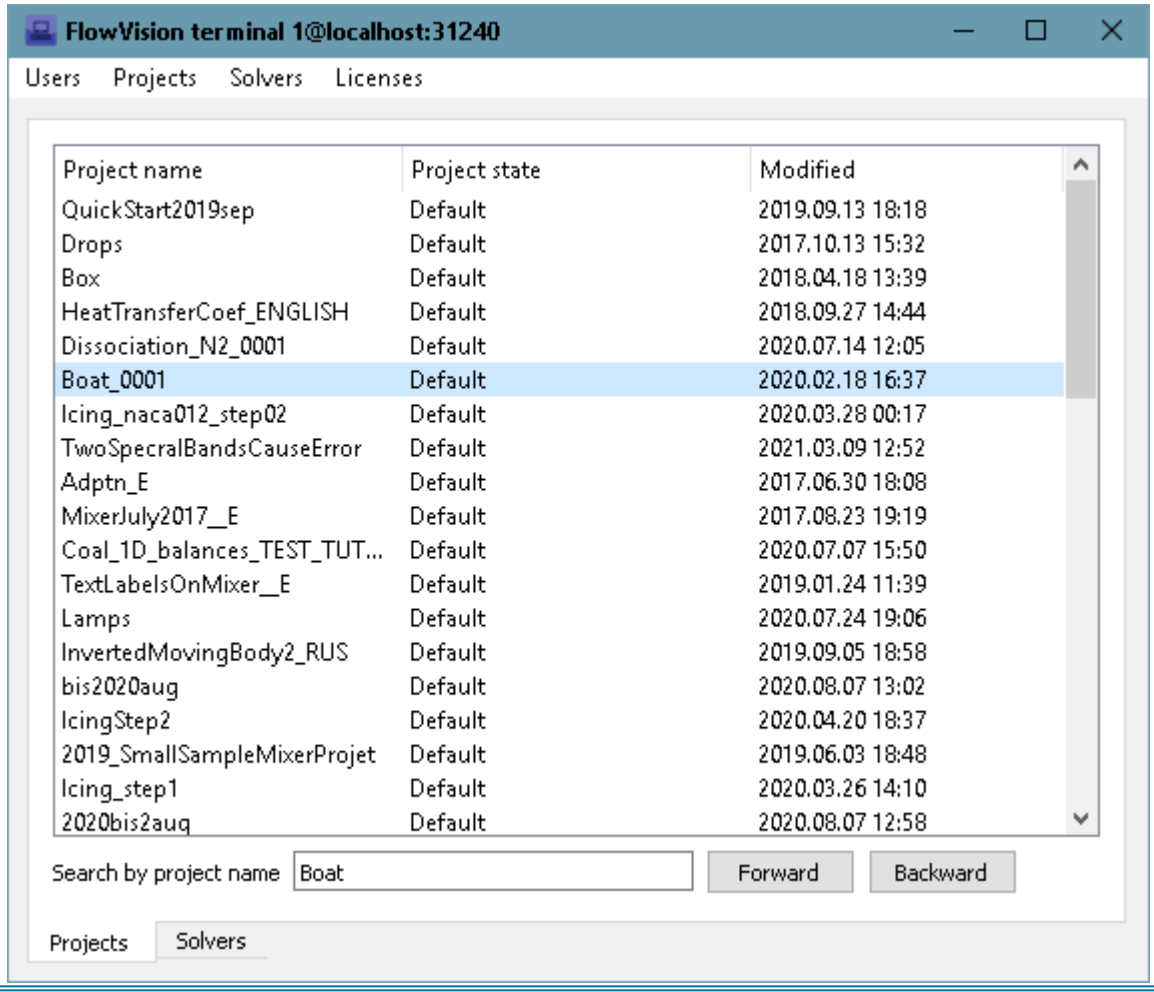
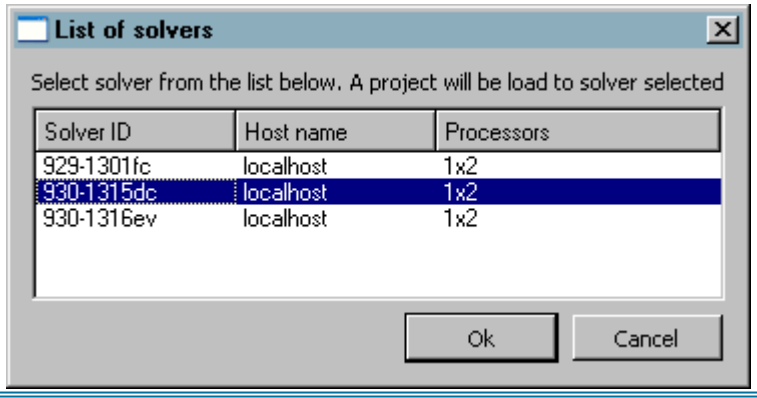
Obtaining the client part of the project from the server part without uploading the server part to Solver	
Step	**Actions**
1	Start **Terminal**.

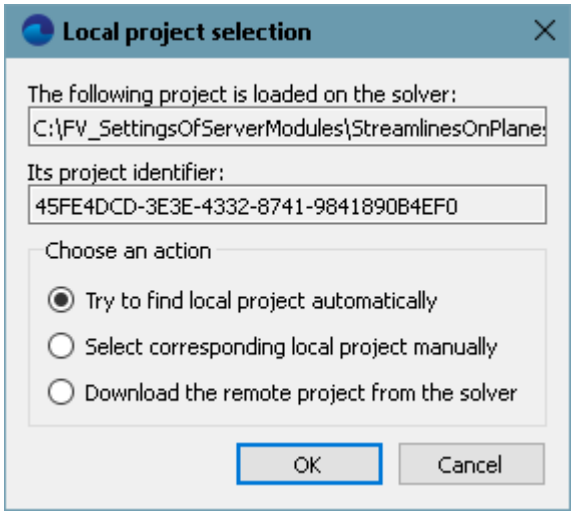
Obtaining the client part of the project from the server part without uploading the server part to Solver	
Step	**Actions**
2	Open the list of projects (it is empty just after the start of **Terminal**) and update it using **Terminal**'s menu command **Projects** > **Refresh list of projects** or pressing on the **F5** key): 
3	Select the required project and from its context menu select **Get project** or use the menu command **Projects** > **Get project**. The **Project getting** window will open:  Select there the **Client part** checkbox and then click **OK** and specify the location for the project's files, which will be saved. After this you will obtain the client part of the project, which will be copied from the server part of the project.

Obtaining the client part of the project from the server part without uploading the server part to Solver	
Step	**Actions**
4	In **Pre-Postprocessor**, open the just created client part of the project.
5	Now you can connect to **Solver-Agent**, start **Solver** and connect to **Solver**, in order to start processing of the computation's results or change the initial data before resuming the existing computation.

Obtaining the client part of the project from the server part with uploading the server part to Solver	
Step	**Actions**
1	Start **Terminal** and connect to **Solver-Agent**.
2	From **Terminal**, start **Solver** (use **Terminal**'s menu command **Solvers** > **Run new solver** or press **Ctrl+R** keys). The **Solver running** window will open:  In this window, specify the necessary parameters of the **Solver**, which is to be started, and then click **Ok**. The **Solver run** window will open with the message "**Solver has been run successfully. Would you like to update a solver's list?**":  Click **Yes**. The list of **Solvers** will open, into which the new **Solver** will be included. Remember the identifier of this **Solver**.
3	Open the list of projects and update it (using **Terminal**'s menu command **Projects** > **Refresh list of projects** or pressing on the **F5** key):

Obtaining the client part of the project from the server part with uploading the server part to Solver

Step	Actions
	
4	Select the required project and upload it to **Solver** (using the command **Load project to solver** from its context menu or using **Terminal**'s menu command **Projects** > **Load project to solver**). The **List of solvers** window will open; select there the **Solver**, which was started at the step 2, and then click **Ok**: 
5	Start **Pre-Postprocessor**.
6	Don't open the client part of the project. Connect to the **Solver**, which has been started.
7	**Pre-Postprocessor**, in the **Local project selection** dialog box, will ask where to find the client part of the project:

Obtaining the client part of the project from the server part with uploading the server part to Solver	
Step	**Actions**
	 Select **Download the remote project from the solver** and click **OK**.
8	Specify, where the program should save the client part of the project.
9	**Pre-Postprocessor** will copy the server part of the project into the client part, which will be created, and then will work with this copy (the client part) being connected with **Solver**.

8.2.9.1.4.5 Client and server parts don't match, so their synchronization is required

In this section we consider situation of inconsistency the client and the server parts of the *same project*.

An attempt to connect to a **Solver**, on which *another project* is loaded, is considered in the section Client and server contain parts of different projects.

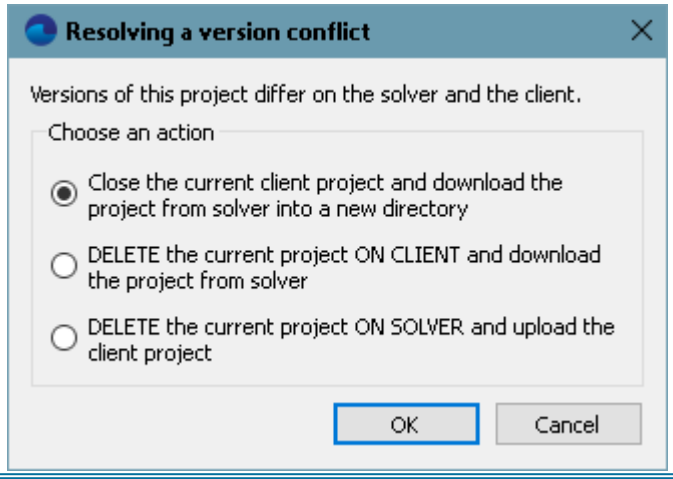
The server and client parts of the project, during some independent operation of **Pre-Postprocessor** and **Solver**, might be changed in such a way, that the projects become incompatible.

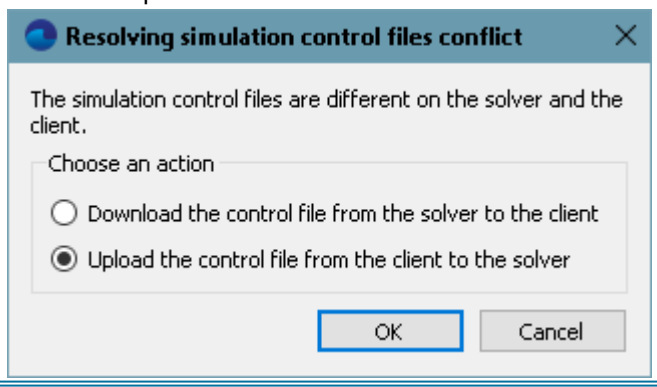
For example, if you, when there are no connection to **Solver**, edit the input data, the projects will become incompatible. Or, if **Solver**, with no connection to **Pre-Postprocessor**, saves the data or changes the shape of the geometry model (during FSI calculation), then the server part of the project also will cardinally differ from the client part. In this situation, the **Resolving a version conflict** window will open requesting to delete one of project's parts replacing it by the other part, or copy the server part into the client part in other directory.

The changes, which have been done in **Pre-Postprocessor** in the tabs **Solver** and **Postprocessor**, are not cardinal and do not cause incompatibility between server and client parts of the project. After such changes, either the **Resolving simulation control files conflict** or the **Resolving a scene conflict** window will open requesting which part of the project is to be changed.

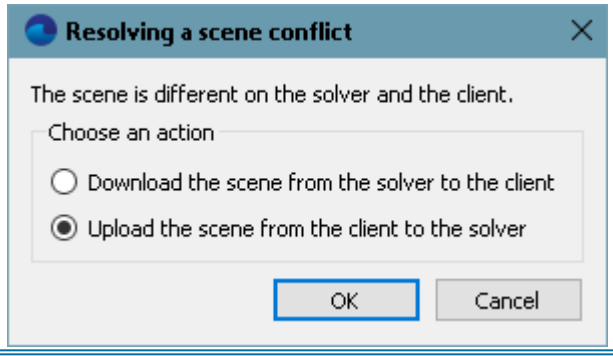
These three cases are examined in the tables below.

Cardinal difference between client and server parts of the project	
Step	**Actions**
1	When changes are made in the project's input data (in the **Preprocessor** tab) with no connection to **Solver**, or if the geometry is changed by **Solver** with no connection to **Pre-Postprocessor**, the client part of the project will cardinally differ from the server part. In this situation, in the client and the server parts of the project, the `VersionID` parameter in the `.fvproj` file differs. In this situation, at the attempt to connect **Pre-Postprocessor** to **Solver**, the **Resolving a version conflict** window will open:

Cardinal difference between client and server parts of the project	
Step	**Actions**
	
2	If you wish to save both client and server parts of the project, you have to copy the server part into a client part in other directory. To do so, select "**Close the current client project and download the project from solver into a new directory**". If you *need* results of the computation, you can abandon changes that have been done in the client part, by deleting the client part and copying the server part into the client part. To do so, select "**DELETE the current project ON CLIENT and download the project from solver**". If you *don't need* the results of the computation, which are contained in the server part of the project, you can delete them and upload to **Solver** the changed copy of the client part (after this you can start the computation only from scratch). To do so, select "**DELETE the current project on ON SOLVER and upload the client project**".
3	Click **OK**.

Calculation parameters, specified in the client and in the server parts of the project, differs	
Step	**Actions**
1	When changes are made in the **Solver** tab with no connection to **Solver**, the calculation parameters, specified in the client and in the server parts of the project, will differ.  In this situation, in the client and the server parts of the project, the `CtrlID` parameter in the `.fvproj` file differs. In this situation, at the attempt to connect **Pre-Postprocessor** to **Solver**, the **Resolving simulation control files conflict** window will open: 
2	If you need to copy the calculation parameters from the server part of the project to the client part, select "**Download the control file from the solver to the client**".

Calculation parameters, specified in the client and in the server parts of the project, differs	
Step	**Actions**
	If you need to copy the calculation parameters from the client part of the project to the server part, select "**Upload control file from the client to the solver**".
3	Click **OK**.

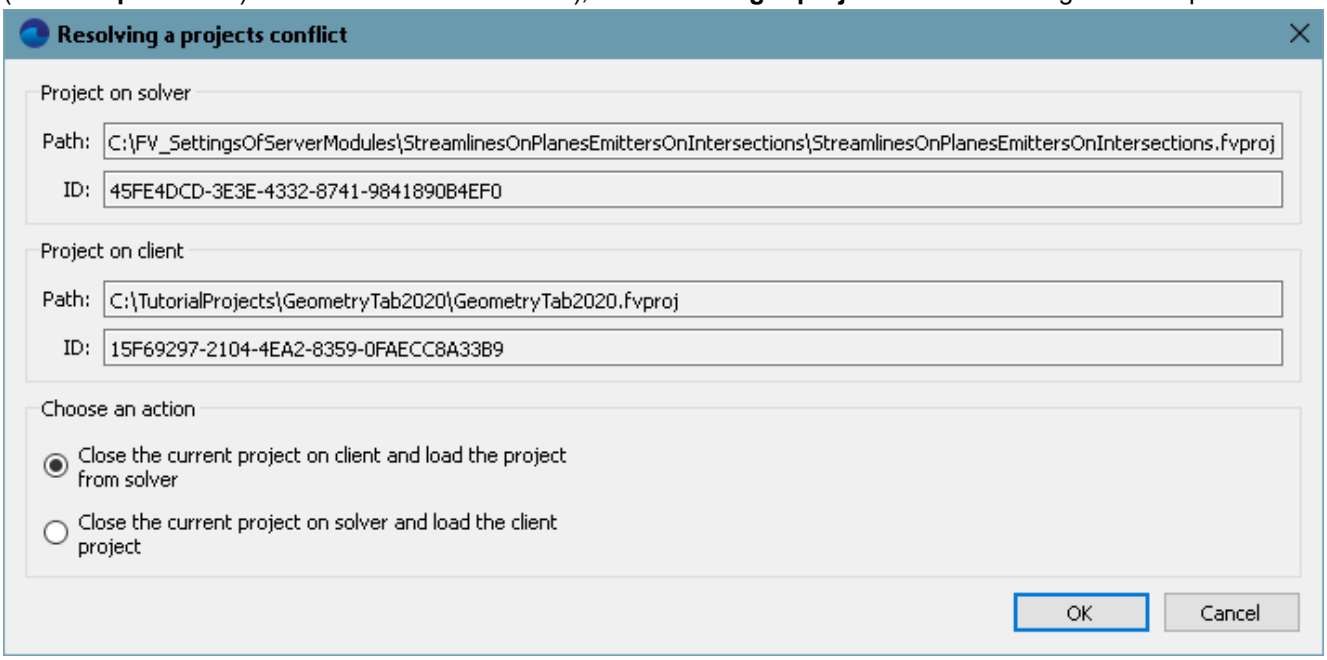
Parameters of the scene, specified in the client and in the server parts of the project, differs	
Step	**Actions**
1	When changes are made in the **Postprocessor** tab with no connection to **Solver**, the parameters of the scene, specified in the client and in the server parts of the project, will differ.  In this situation, in the client and the server parts of the project, the `SceneID` parameter in the `.fvproj` file differs. In this situation, at the attempt to connect **Pre-Postprocessor** to **Solver**, the **Resolving a scene conflict** dialog box with message "**The scene is different on the solver and the client**" opens: 
2	If you need to copy the parameters of the scene from the server part of the project to the client part, select "**Download the scene from the solver to the client**". If you need to copy the parameters of the scene from the server part of the project to the client part, select "**Upload the scene from the client to the solver**".
3	Click **OK**.

 If you need to make changes in the input data and *continue* the computation, but *not start the computation from scratch*, you should do changes in the project's client part when **Solver** is connected; this provides automatic and synchronous change of the server part of the project.

 Also the following scenario of the work is possible: during the computation, **Solver** saves the previously adjusted scene into `fvvis` files. The user connects to **Solver** to analyze the current results during the computation. The user can create new layers or move existing layers, but cannot save the changes in the server part (because such saving would change the stored scene). At the next connection the user can reject all these changes in the project's client part and load the scene parameters from the project's server part.

8.2.9.1.4.6 Client and server contain parts of different projects

At attempt of connecting the project to a **Solver**, on which another project is loaded (and no controlling client (**Pre-Postprocessor**) is connected to this **Solver**), the **Resolving a projects conflict** dialog box will open:

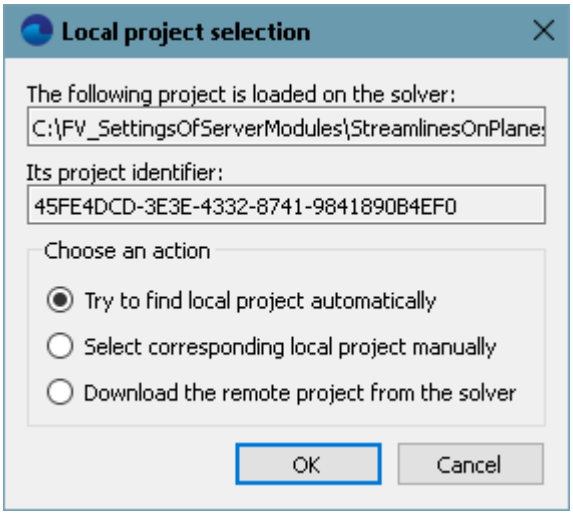


This dialog box displays the following information:

- network path and file name of the project loaded on the **Solver**
- identifier of the project loaded on the **Solver**
- network path and file name of the project on the client (the project opened in **Pre-Postprocessor**)
- identifier of the project on the client (the project opened in **Pre-Postprocessor**)

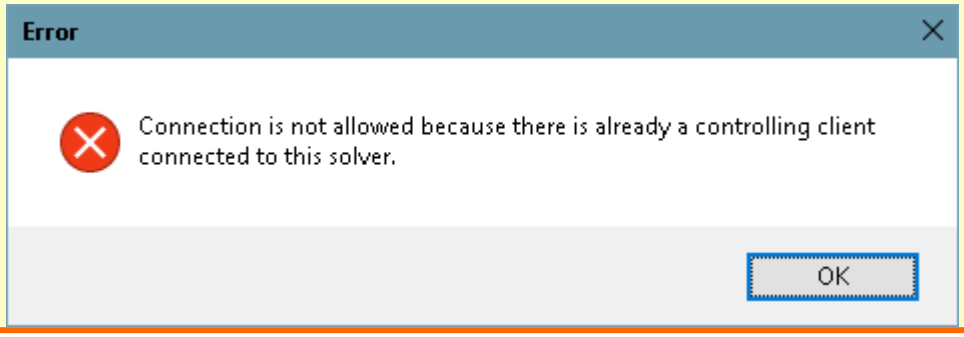
The user has the following choices:

- **Close the current project on client and load the project from solver** – project from the server part's directory will be loaded into the client part's directory.
  The **Local project selection** dialog box will open:



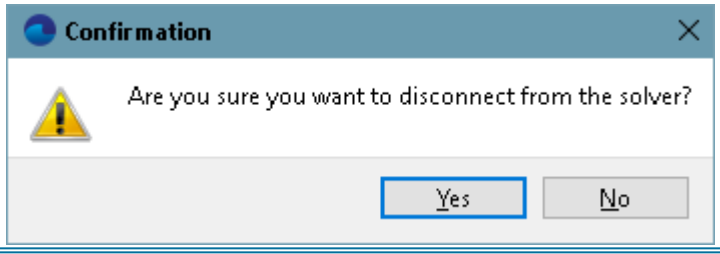
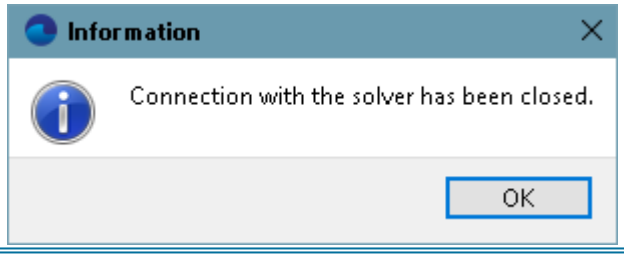
  In the **Local project selection** dialog box you can select the following choices:
  - **Try to find local project automatically** – the program will attempt to automatically find the client part of the project, which is loaded on the **Solver**.
  - **Select corresponding local project manually** – on **Pre-Postprocessor** the project will be loaded, which you selected manually as a **`fvproj`**-file on a disk. If this project is not the same one, which is loaded on **Solver**, a conflict of projects will appear again and the **Resolving a projects conflict** dialog box will open.
  - **Download the remote project from the solver** – on **Pre-Postprocessor** the project will be loaded, which is loaded on **Solver**. The network path, file name, and identifier of this project are listed in the upper part of the dialog box.
- *or* **Close the current project on solver and load the client project** – the project from the client part will be loaded into the server part's directory.

At attempt of connecting to a **Solver**, to which another controlling client (**Pre-Postprocessor**) is connected, the program outputs an error message (**Connection is now allowed because there is already a controlling client connected to this solver**):



8.2.9.1.5 Disconnecting Pre-Postprocessor from Solver

To disconnect Pre-Postprocessor from Solver:

Step	Actions
1	Click the (**Disconnect from the solver**) button in the toolbar **Network**. The system will prompt you to confirm disconnection of **Pre-Postprocessor** from **Solver** ("**Are you sure you want to disconnect from the solver?**"): 
2	Click **Yes**. An information window will display a message about disconnection of **Pre-Postprocessor** from **Solver** ("**Connection with the solver has been closed.**"): 
3	Click **OK**.

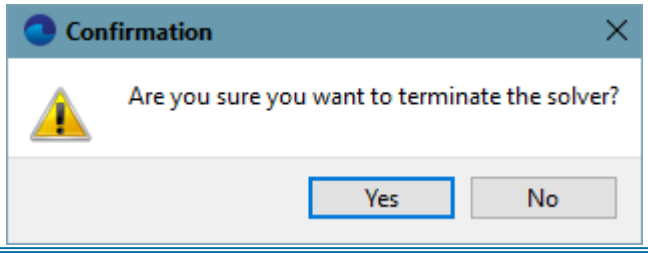
If you open the Select solver dialog box, then, in the **Running solvers** list, the **Solver**, which has been disconnected from **Pre-Postprocessor**, can have the following statuses:

- **STOPPED** (if the **Solver** doesn't calculate a project)
- *or* **SOLVING** (if the **Solver** calculates a project).

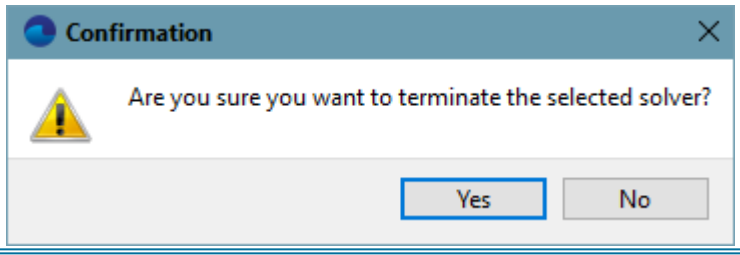
8.2.9.1.6 Terminating a Solver

Terminating the currently connected Solver

Step	Actions
1	Click the **(Close the solver and disconnect from it)** button in the **Network** toolbar. The system will request a confirmation (**Are you sure you want to terminate the solver?**):

Step	Actions
	
2	Click **Yes**. The line of the **Solver**, to which the project was attached, will disappear from the list of running solvers in the **Select solver** dialog box.

Terminating a Solver selected from the "Select solver" dialog box

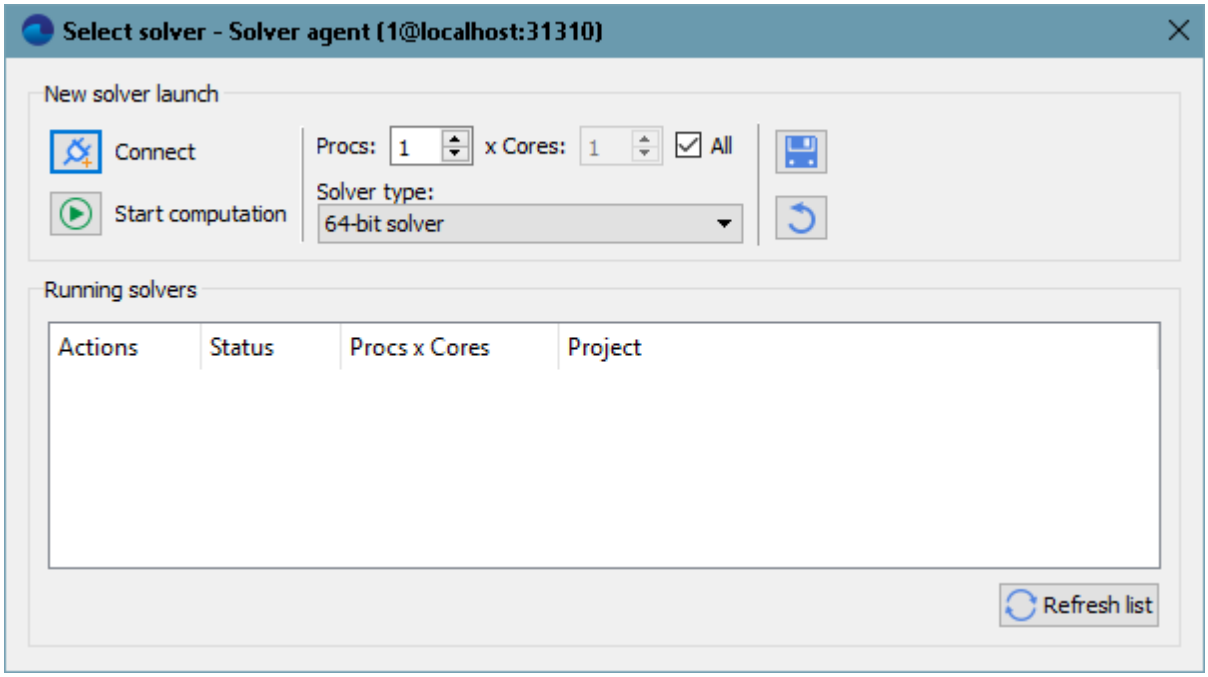
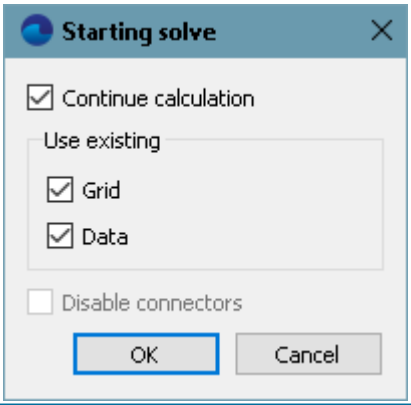
Step	Actions
1	Click the button (**Open solver selection window**) in the **Network** toolbar to open the **Select solver** dialog box.
2	Select the desired **Solver** from the **Running solvers** list, and then, in the **Actions** column, click the (**Terminate**) button. The system will request a confirmation (**Are you sure you want to terminate the selected solver?**): 
3	Click **Yes**. The line of the selected **Solver** will be deleted from the **Running solvers** list.

8.2.9.2 Starting and stopping project's computation

Procedures of initial starting and of restarting project's computation are somewhat different.

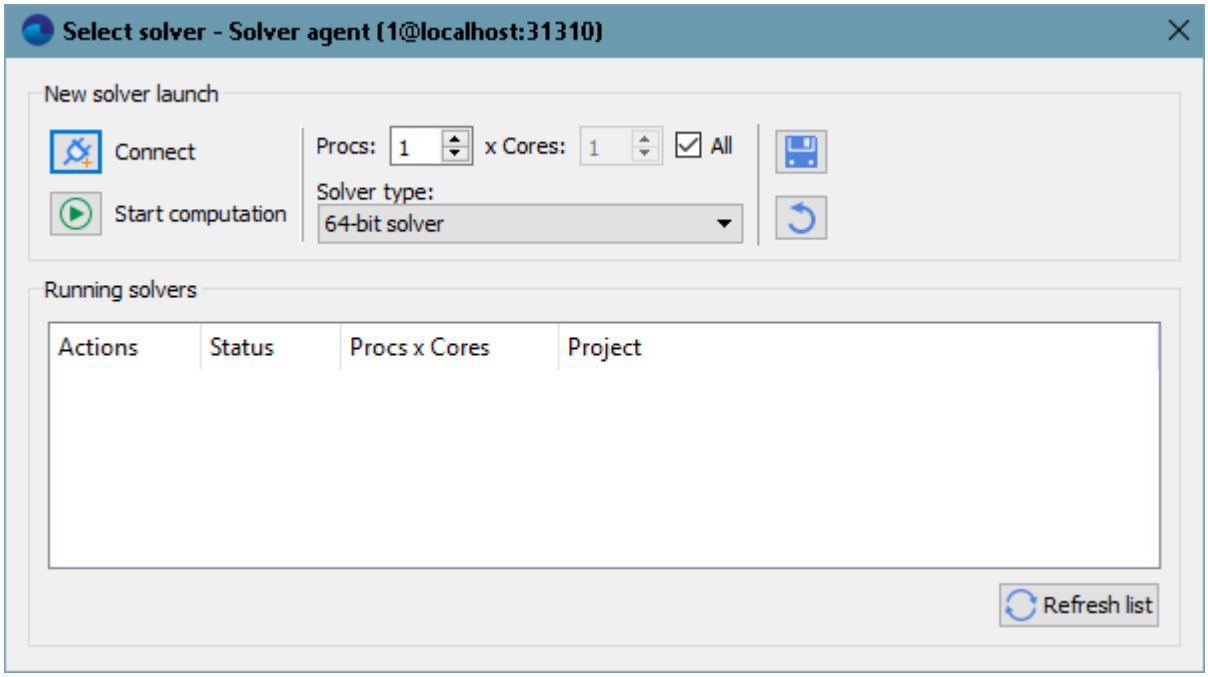
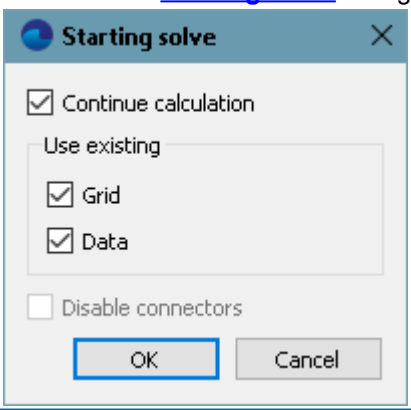
Initial starting of the project's computation

Step	Actions
1	Before starting computation of the project, save the project by clicking the (**Save changes to the client side of the project**) button in the toolbar **Standard**.
2	Authenticate on **Solver-agent**.
3	Click the button (**Open solver selection window**) in the **Network** toolbar. The **Select solver** dialog box will open:

Step	Actions
	 If necessary, change parameters of the **Solver**, which will be started, in fields **Procs**, **Cores**, **All**, and **Solverf type** (see details in the section Dialog box "Select solver"). If you wish, you can save the specified parameters of **Solver** for future use as default settings. To do so, apply the button (**Save as default parameters for creating solver**). To load the default parameters of **Solver**, apply the button (**Load default parameters**).
4	Start a new **Solver** with connecting the current project to it, and starting computation of the current project on this **Solver**. To do so, in the opened **Select solver** dialog box, in the **New solver launch** group of interface elements, specify parameters of the new **Solver** and then click the **Start computation** screen button. The **Select solver** dialog box will close and the **Starting solve** dialog box will open: 
5	In the **Starting solve** dialog box click **OK**.

Restarting the project's computation (either resuming the computation or starting the computation from scratch)

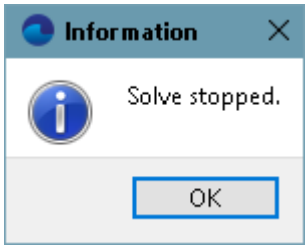
Step	Actions
1	Before restarting computation of the project, save the project with its last changes by clicking the (**Save changes to the client side of the project**) button in the toolbar **Standard**.

Step	Actions
2	If necessary, authorize in **Solver-agent**.
3	Click the button (**Open solver selection window**) in the **Network** toolbar. The **Select solver** dialog box will open:  If the **Solver**, on which you wish to run the project's computation, is absent in the **Running solvers** table, specify parameters of a new **Solver** and click **Solver** and then click the (**Start computation**) screen button to run the new **Solver** and run computation of the current project on it. If the **Solver**, on which you wish to run the project's computation, is present in the **Running solvers** table, then click in its line on the (**Start computation**) screen button to run computation of the current project on this **Solver**. The **Select solver** dialog box will close and the **Starting solve** dialog box will open: 
4	In the **Starting solve** dialog box either unselect or keep selected checkboxes according to your requirements and click **OK**.

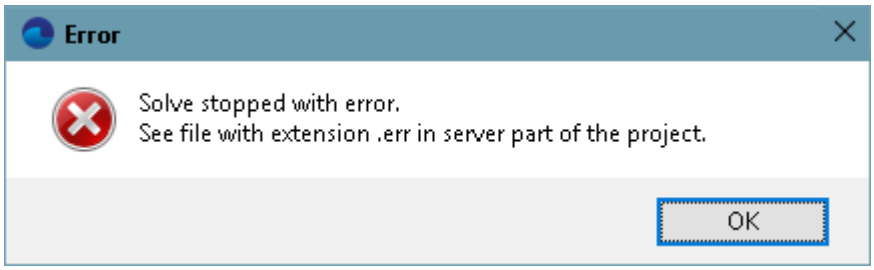
Stopping the project's computation

When a project is being calculated, you can stop it by clicking the **(Stop computation)** button in the toolbar **Network**.

An informational dialog box will open where you have to click **OK**:

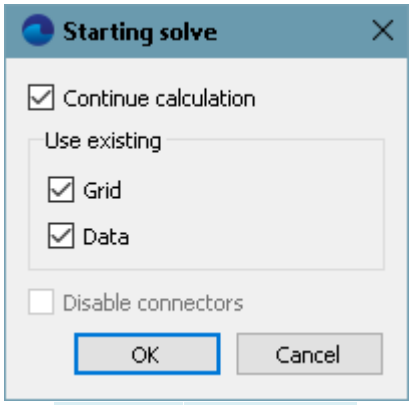


In the case of abnormal termination of the computation, which occurs without a user's command, an error message will be displayed ("**Solve stopped with error. See file with extension .err in server part of the project.**"):



8.2.9.2.1 Dialog box «Starting solve»

The **Starting solve** dialog box is used to specify what the results of the previous calculation - the grid and fields of the calculated variables - should be used in the continuation or update the calculation of the project.



Dialog box **Starting solve**

The **Starting solve** dialog box is opened when you click the screen button in the **Network** toolbar or in the **Select solver** dialog box.

The dialog box closes when you click **OK** (this starts the computation with selected settings) or click **Cancel** (this cancels starting of the computation).

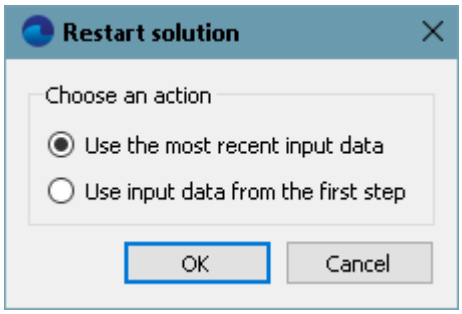
Actions performed by the program when selecting/unselecting the checkboxes in the dialog box «Starting solve»		
Checkbox	**Actions when the checkbox is selected**	**Actions when the checkbox is unselected**
Continue calculation	Resuming the computation with the following parameters:	Resuming the computation
***(Use existing)* Grid**	The computation proceeds on the same grid as the previous run, i.e. without adjustment of the computational grid.	The computation begins with the construction of a new grid. The calculated data obtained at the end of the previous run are deleted. The computation starts anew, the initial data for the calculated variables are set equal to the specified in the project tree

Actions performed by the program when selecting/unselecting the checkboxes in the dialog box «Starting solve»		
Checkbox	**Actions when the checkbox is selected**	**Actions when the checkbox is unselected**
***(Use existing)* Data**	Computation continues with the data that were obtained at the end of the previous run	The calculated data obtained at the end of the previous run are deleted. The computation starts anew, the initial data for the calculated variables are set equal to the specified in the project tree.
Disable connectors (this checkbox is available only for joint simulations, see Connectors)	Start the computation without connection to an external software	Start the computation when an external software connects

Depending on the selected/unselected checkboxes, *FlowVision*'s actions will be the following:

1. The **Continue calculation** checkbox is unselected. This means that the computation runs again, starting from the initial conditions that are specified in the **Pre-Postprocessor**.
   - If the back end of the project there is automatically stored data, the calculation starts project **Solvers**.
   - If the server part of the project has more than one portion is automatically stored data, then the data will not be used for calculating in the next calculation.

   However, the calculation might use various the project's input data; the user can select the data from the **Restart solution** dialog box that opens:



   Possible action by the user and the system response:
   - if the switch is set to **Use the most recent input data**, the calculation will be carried out, starting from the latter source project data specified in **Pre-Postprocessor**.
   - if the switch is set to **Use input data from the first step**, then the calculation will be carried out, starting from the initial project data stored at the first start of calculation of the project.

   In both cases of the client and server will delete all files are automatically saved with the previous launches of the calculation of the project. History of data autosave begins from scratch after the project's restart.
   - The existing `glo` and `fvvis` files of the project will be removed after the project's restart.

2. The **Continue calculation** checkbox is selected. Actions performed by the system after selecting/unselecting checkboxes **Grid** and **Data** (if **Continue calculation** is selected) are listed above in the table *"Actions performed by the system when installing/removing the marks in the dialog box «Starting solve»"*.

8.2.9.3 Project computation progress control (operations)

Inspection of the calculation of the project is carried out, if in the course of calculating the project **Pre-Postprocessor** is connected to **Solver**. Control on the last calculated time step in the following ways:
- control of the main parameters of the calculation in the **Monitor** window
- displaying component characteristics in the **Info** window
- visualization of the layersin **View** window and display the data on the variable in the **Info** window

Also possible to view the intermediate results in the dynamics calculations.

Control of the main parameters of the calculation in the «Monitor»

Select the display mode of the window **Monitor** and follow in the calculation of the parameter is displayed:

a) **Status** tab to track the changes of the following parameters:
   - number of iterations, which should not significantly increase (strong growth means instability of the computation).

- Algebraic residuals that must be within a predetermined accuracy of calculation;
- functional residuals, which should eventually decrease to a low value corresponding to steady flow, or oscillate with a limited amplitude.

b) **Plot** tab to track the changes of functional residual variables and user specified as a criterion for stopping the calculation.

Displaying components of characteristics in the «Info» window

Step	Actions
1	Select the project tree line item **features** to be displayed.
2	Press  in the toolbar **Work modes**. **Info** window will open with the characteristics of components.

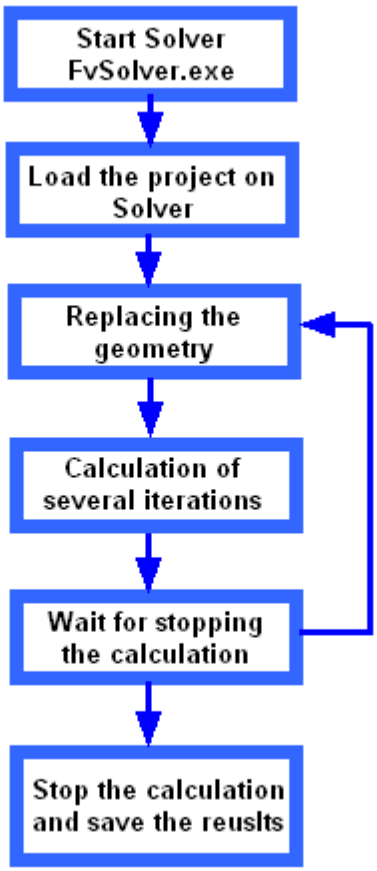
Visualization of Layers on the last calculated time step

Step	Actions
1	Select the project tree line element **Layer**, to be displayed.
2	Settings to see if the layer in the **View**.
3	Press  in the toolbar **Work modes**. **Info** window will open with the data on the variable.

8.2.9.4 Automatic replacement of the Moving body's geometry during the computation

In this section we show how to do automatic replacement of the moving body's geometry during the computation.

FlowVision allows you to start **Solver** in the batch mode. In this mode you can instruct **Solver** to replace the geometry of a **Moving body**. The computation can be done by the following scheme:



This algorithm requires the following files:

1. An *executable file* containing a command string to start **Solver**.
2. A *command file* containing a set of commands for the running **Solver**.

Example:

The command line in the executable file:

```
"C:\FlowVision\FvSolver.exe" threads=4 sa_user=1 sa_ID=111-111111 sa_license=fvlic cmdfile="D:\FV_Batch_Run\cmd.txt"
```

where:

- `"C:\FlowVision\FvSolver.exe"` defines the path to **Solver**
- `threads=4` defines the run mode (run on 4 processor cores)
- `sa_user=1` defines the username on **Solver-Agent**
- `sa_ID=111-111111` defines the **Solver**'s identifier
- `sa_license=fvlic` defines the license name
- `cmdfile="D:\FV_Batch_Run\cmd.txt"` defines the path to the command file

Set of commands in the command file:

```
SS_PROJECTLOAD<D:\solverside\FV_Batch\fvbatch.fvproj><><><>
SS_APPLYNEWGEOMETRY<OBJECT[10]><D:\FV_Batch_Run\Geom\0.wrl>
SS_SOLVESTART_EX<0><RI><2>
SS_DO_NOT_DISTURB
SS_APPLYNEWGEOMETRY<OBJECT[10]><D:\FV_Batch_Run\Geom\1.wrl>
SS_SOLVESTART_EX<7><RI><2>
SS_DO_NOT_DISTURB
SS_APPLYNEWGEOMETRY<OBJECT[10]><D:\FV_Batch_Run\Geom\2.wrl>
SS_SOLVESTART_EX<7><RI><2>
SS_DO_NOT_DISTURB
```

...
...
...
SS_PRJDATASAVE
SS_PROJECTUNLOAD
SS_SHUTDOWNSOLVER

where:

- **SS_PROJECTLOAD<D:\solverside\FV_Batch\fvbatch.fvproj>** loads the specified project on **Solver**
- **SS_APPLYNEWGEOMETRY<OBJECT[10]><D:\FOR FV\FV_Batch_Run\Geom*n*.wrl>** replaces the geometry of the **Imported object**, which has identifier **10**, by the geometry located in the specified directory. The object's identifier is specified in the file **fvinp** in the string with the appropriate value of the **myid** parameter (here is an example of such string: **UIName="Imported object #0" class="CMeshEntity" myid="10"**).
- **SS_SOLVESTART_EX<0** *or* **7><RI><2>** runs **Solver** on calculation. Parameters in these commands have the following meanings: **0** means starting the calculation from scratch (in the first command) or **0**, which specifies continuation of the calculation (in further commands), **RI** specifies stopping the calculation after the specified number of steps, which is set by the next parameter, **2** is the number of steps after which the calculation stops.
- **SS_DO_NOT_DISTURB** means "do not process other requests until computation is not finished"
- **SS_SOLVESTOP** stops the **Solver**'s computation
- **SS_PRJDATASAVE** saves results of the computation
- **SS_PROJECTUNLOAD** unloads the project from **Solver**
- **SS_SHUTDOWNSOLVER** unloads **Solver**

(see descriptions of parameters of the command file in the section Command file)

Note, that repeating groups of commands **SS_APPLYNEWGEOMETRY**, **SS_SOLVESTART_EX** and **SS_DO_NOT_DISTURB** form the cycle of calculations with replacements of geometry.

Notes:

1) Frequency of the **Moving body**'s geometry replacements does not have be the same as the computation's time step.
2) Updates of the **Moving body** must be enabled.

9 Third-party software interaction

FlowVision's operation can be conjugated with operation of third-party (external) programs; this can significantly improve efficiency of the work.

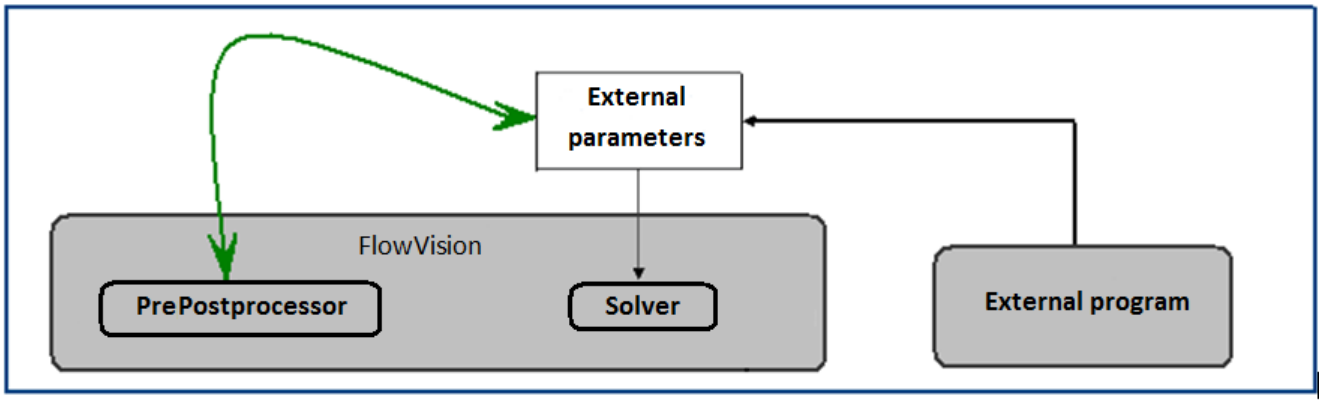
Problems, which are solved simultaneously by *FlowVision* and two or more other programs, are called *conjugate* ones.

See sections:

9.1 External input parameters

Some parameters of the project can be declared external parameters with values set in the **Properties** window is not in the **Pre-Postprocessor**, as taken from the *file of the external parameters*.

The file is located in the external parameters of both parts of the project (client and server), and has a name ***(the name of the project)*.fvdtbl**.



Getting input parameters from an external program

External parameters file is a text file containing data in XML.

External input parameters are used in repeated automatic one by one runs of *FlowVision* and another program. This allows effectively solve optimization problems (see Optimization).

FlowVision action for non-compliance of the external parameters file list of external parameters

Parameters specified in the fileof the external parametersare displayed in the list of external parameters in the **Exposed parameters window**. If for some reason the fileexternal parameters to specify a parameter that does not exist in the project, it will be ignored in *FlowVision*.

File format of the external parameters

All data in the file of the external parameters are stored within the root tag **<PARAMETERTABLE>**. Information about individual settings are stored in the tags **<PARAMETER>**, with the following attributes:

Attribute	Description
ID	Internal identifier parameter in the project tree *FlowVision*.
Name	The unique name of the parameter defined in the **Exposed parameters window**
Comment	Comment that is displayed at the bottom of the **Exposed parameters window**
Type	Data type of the parameter
Value	Value of the parameter. This value will replace the value specified for this parameter in the **Properties** window.

Sample file of the external parameters

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<PARAMETERTABLE>
  <PARAMETER Comment="" ID="OBJECT[2].OBJECT[268].OBJECT[26].NormMassVel"
Name="Mass_velocity_inlet2" Type="double" Value="3"/>
  <PARAMETER Comment="" ID="OBJECT[2].OBJECT[92].OBJECT[26].NormMassVel"
Name="Mass_velocity_inlet1" Type="double" Value="3"/>
```

```
  <PARAMETER Comment="" ID="OBJECT[2].OBJECT[87].OBJECT[23].FluxValue" Name="Value"
Type="double" Value="10000"/>
  <PARAMETER Comment="" ID="OBJECT[2].OBJECT[92].OBJECT[13].WallValue"
Name="Temperature_inlet1 " Type="double" Value="15"/>
  <PARAMETER Comment="" ID="OBJECT[2].OBJECT[268].OBJECT[13].WallValue"
Name="Temperature_inlet2" Type="double" Value="15"/>
</PARAMETERTABLE>
```

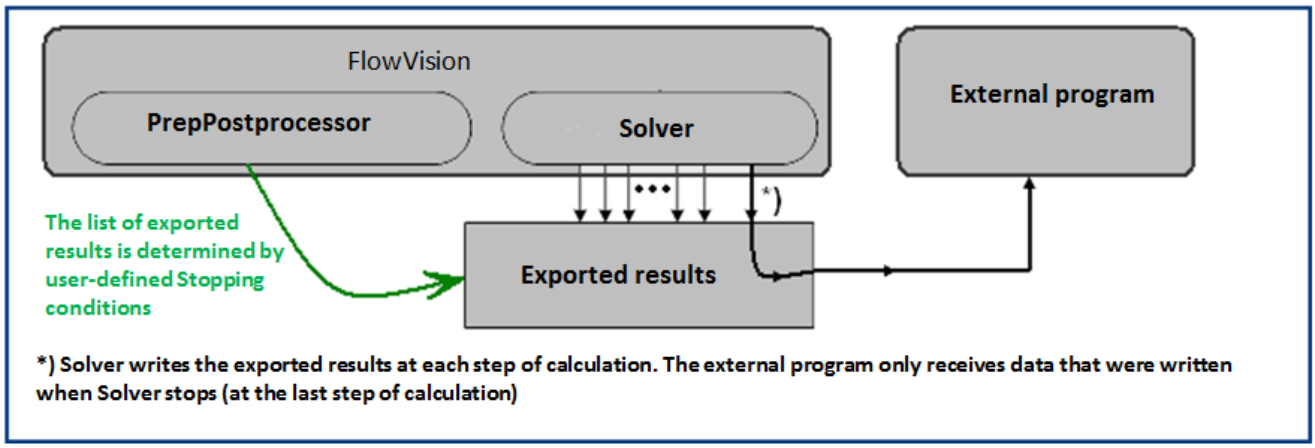
9.2 Exported results

Part of the data obtained bycalculating the project may be stored in a *file of exported results*. This file is located in both parts of the project (client and server), and has the name `(the project's name).fvrtbl`.

Composition of data stored in the file is determined by the contents of the elements **Stopping conditions > User values > Stop criterion #N** in the project tree (in the **Solver** tab).

The data stored at each step of the calculation. It is assumed that the data will be retrieved from the file any external program after stopping **Solver**, so the external program receives data stored on the last step of the calculation. External program, these data will be imported.

Transfer the exported resultsused in multiple automatic start *FlowVision* alternately with an external program (see Optimization).



Transfer the exported results to an external program

File format of exported results

All data in the file exported results are stored within the root tag `PARAMETERTABLE`.

Information about individual settings are stored in the tags `PARAMETER`, with the following attributes:

Attribute	Description
`ID`	An internal identifier of the parameter in the project tree.
`Name`	A unique name of the parameter. This name corresponds to the name of a stopping criterion specified in the **Properties** window.
`Comment`	Comment
`Type`	The parameter's data type
`Value`	The parameter's value, it is filled by **Solver**.

Example file exported results

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<PARAMETERTABLE>
  <PARAMETER Comment="Integral characteristic &quot;Temperature_on_wall&quot;-&quot;Surface of a supergroup&quot;-&quot;&lt;f surf.>&quot;"
ID="{CONTROLS}.Stoppers.StopperUser[0].CurrentValue" Name="Temperature_of_wall" Type="double"
Value="0"/>
  <PARAMETER Comment="Integral characteristic &quot;Temperature on wall&quot;-&quot;Surface of a supergroup&quot;-&quot;Deviation on surface&quot;"
ID="{CONTROLS}.Stoppers.StopperUser[1].CurrentValue" Name="Deviation_of_temperature"
Type="double" Value="0"/>
</PARAMETERTABLE>
```

9.3 Data export after computation

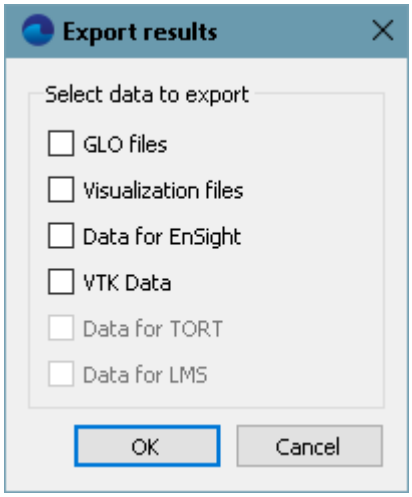
During computation the following results can be exported:

- files **glo** (including the loadings export file, which default name is **loadings.glo**)
- visualizations data (files **fvvis**)
- visualization data for the *EnSight* package
- data in the *VTK* visualization format
- data for *TORT*
- data for *LMS* (in the *CGNS* format)

For this data export use:

- the **File > Export** command from the Main menu
- or the (**Export data on solver**) button on the **Standard** toolbar
- or the **Export results** command from Terminal (menu Projects)

The **Export results** dialog box will open:



Export data for *TORT* and *LMS* requires prior specify additional parameters of their exports to the **Pre-Postprocessor** in the project tree in the **Solver** tab.

To automatically export at "playing" steps in **Pre-Postprocessor** has a button turns on / off data export during the "play". It is available only in connection with the solver, and only when the data is stored on the steps and playing is "stopped".

If, at the beginning of the "play" button automatic export is enabled, the data will be exported, the export of which is set on the tabs **Solver** and **Postprocessor**. Exports will be made with the frequency with which it is given, starting from the step, which started with "play." It should be borne in mind that when "playing" periodicity export data will be limited to periodic data storage. For example, if the frequency of data storage was 5 iterations, and the frequency of data export - 3 iterations, during the calculation, we obtain the export data at 3, 6, 9, 12,15, 18, etc. iteration, and during a "playback" - 5, 10, 15, 20 etc. saved iterations.

When settings of data saving for *TORT* and/or for *LMS* is changed in **Properties** windows of element **Export to TORT** or **Export to LMS**, then checkboxes **Data for TORT** and **Data for LMS** in the **Export results** might be unavailable until the project is saved by:

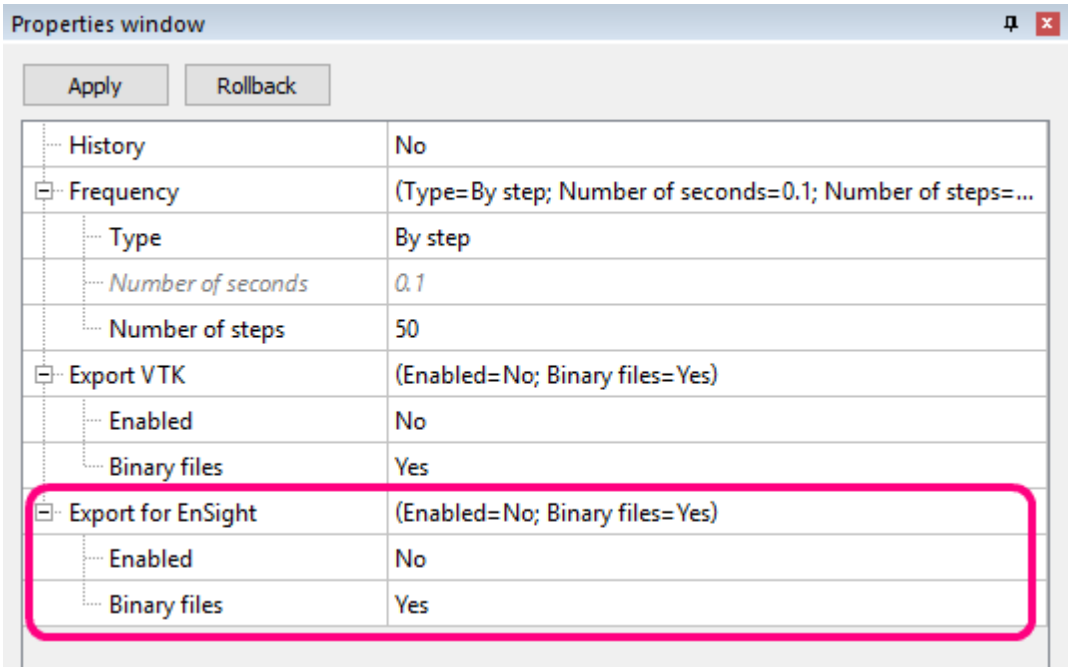
- the command **File > Save** of the Main menu
- *or* button (**Save changes to the client side of the project**) in the **Standard** toolbar
- *or* hot key **Ctrl-S**

9.4 Data export for visualization in EnSight

FlowVision can save computational grid and the calculation results recorded at one time step or steps at several time steps in the format of *EnSight* for subsequent visualization, building layers and analysis in *EnSight* (https://www.ansys.com/products/fluids/ansys-ensight).

As computational grid of *FlowVision* might substantially differ from traditional grids used in *Paraview* (https://www.paraview.org/), reading *EnSight* files in *Paraview* could be incorrect.

Parameters of export to *EnSight* are defined in the **Properties** window of the element **Data autosave** in the **Solver** tab in the project tree, in the group of parameters **Export for EnSight**:



Parameters of data export for visualization in *EnSight*:

Parameter	Description
Export for EnSight > Enabled	Enable periodic export for *EnSight*. Possible options: **Yes** | **No**.
Export for EnSight > Binary files	Selection of format of files for export to *EnSight* (binary or text files). Possible options: • **Yes** - use binary files • **No** - use text files

(Turning on the data saving and its frequency are defined by parameters **History** and **Frequency > ...** in the **Properties** window of the element **Data autosave**)

When you request data export for *EnSight* using button (**Export data on solver**) on the toolbar **Standard** (see section Data export after computation) data from one (the current) time step are exported.

If data export to *EnSight* is going during playback of the history, then settings defined by the **Data autosave** element are applied.

Files for data export to *EnSight*

File	Format	Description
`case`	XML	This file is a directory, which lists all calculated variables
`geom`	binary	Geometry model of the computational domain («main geometry»)
data files	XML or binary	Data files for each calculated variable

Impossibility of displaying all data if number of processors was reduced

If a computation is resumed (started for continuation) with smaller number of processors, then after export to *EnSight*, the *EnSight* visualization does not contain all partitions until the moment when number of processors was changed.

For example, if the program does computation of N steps on 4 processors, and continue on 2 processors, then the visualization in *EnSight* at N beginning steps will contain only a half of partitions.

Recording without saving the history

The data exported without saving the history go into the **`EnSight`** subdirectory in the server part of the project.

When a file is overwritten, its old contents is saved into the file **`.backup`**.

Recording without the history saving can be done:

- as automatic periodical saving the data during the computation without history saving
- by request if the history is disabled
- by request while data export at the current step

Recording with saving the history

The data exported with saving the history go into the **`EnSightTransient`** subdirectory in the server part of the project.

When a file is overwritten, its old contents is saved into the file **`.backup`**.

Recording with the history saving can be done:

- as automatic periodical saving the data during the computation with history saving
- by request if the history is enabled
- while the data export goes during a playback of the saved history

9.5 Use of connectors

Interaction between *FlowVision* and external programs is implemented using convenient interface of *connectors* that are specified in the folder **External Connections**.

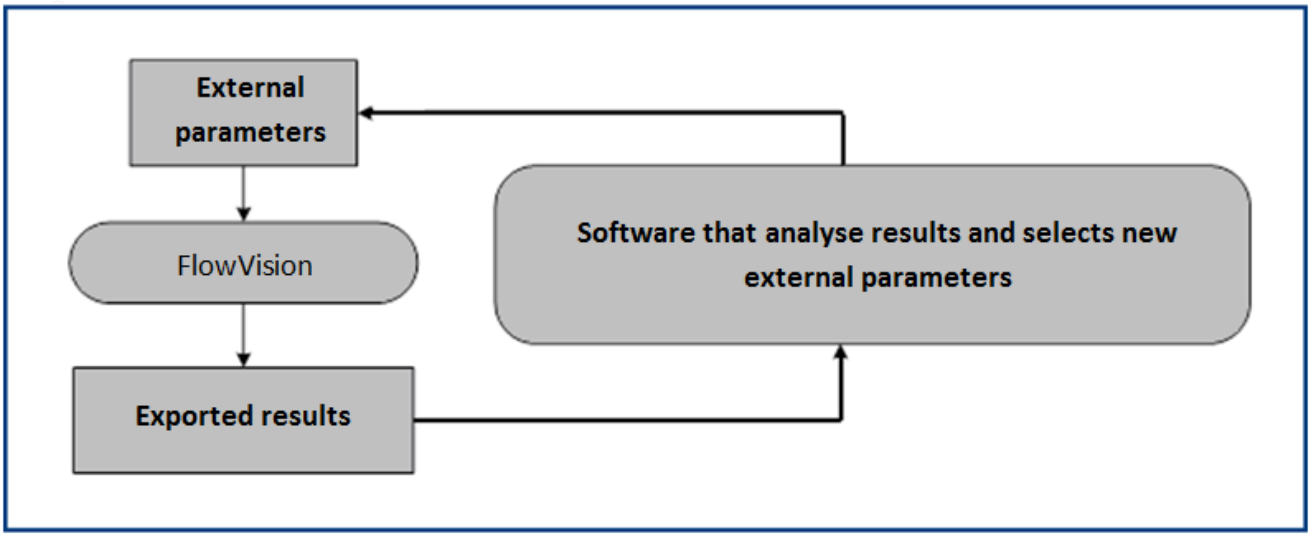
9.6 Optimization

Optimization technology

Often the search for the optimal design of a technical device or optimal operation of the device can be reduced to sorting projects whose parameters vary within certain limits.

Such a brute force does not have to be the piece, and can be arranged so that the parameters of the project is determined by the following calculation results of the previous project.

In these cases, you can run *FlowVision* automatically and repeatedly in order to find the optimal design:



Scheme for finding the optimal project

External parameters and exported results are stored in files in the project folder:

- external parameters- in the *file of the external parameters* **(Project name).fvdtbl**
- exported results *-results* in a *file exported* **(Project name).fvrtbl**

These files are text files that store data in the format of XML, see sections External inputs and Exported results.

If any of the parameters included in the list of external parameters, when you start **Solver** their values are not taken from the **Properties** window of **Pre-Postprocessor** and file of the external parameters **(fvdtbl)**.

Also, at each step of the calculation (and therefore also at the end of the calculation) element content **Stopping conditions > User values > Stop criterion #N** in the project tree (in the **Solver** tab) if **Generate results table** checkbox is selected in the **Exposed parameters window** recorded in the file of exported results (**fvrtbl**), from which they will be able to pick up an external program.

After each stop **Solver** (in accordance with the conditions specified break point) external program reads data from a file exported by the results and the results of their analysis generates new data placed in the file of the external parameters, and starts calculating a new project. Thus *FlowVision* and external program running in a loop that allows you to search the optimal solution is not rude brute force, and adjusting to find the optimal solution on the basis of the previous steps.

See also:

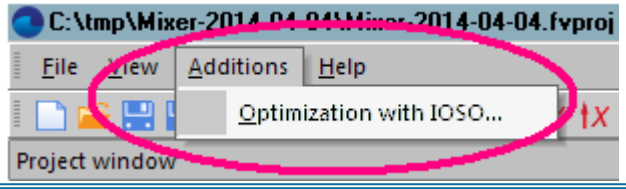
- External input parameters
- Exported results
- Connection of FlowVision and IOSO

9.6.1 Connection of FlowVision and IOSO

Communication of *FlowVision* with the *IOSO* software package allows solving the problem of parameter optimization for the flow.

Interaction between *FlowVision* and *IOSO* is a special case of optimization with data transfer via 2 files *(*the *file of external parameters* and the *file of export results*).

In order to create a joint project *FlowVision+IOSO*, follow these steps:

Step	Actions
1	Open a project in *FlowVision*.
2	Make optimization parameters in *FlowVision* available to change the software package *IOSO*, adding them into the **Exposed parameters window**. In the **Exposed parameters window** dialog box, check the **Generate result table** checkbox.
3	In the project tree, in the tab **Preprocessor**, create an element **Characteristics #N (object name)**, containing values which (or which combination) can be used as criteria for optimization.
4	Create **Stopping conditions** by **User values**, which are equivalent to the optimization criteria.
5	Run *IOSO* using the command **Additions > Optimization with IOSO** from the Main Menu (this command is available when *IOSO* is installed on the local computer and **Pre-Postprocessor** is not connected to **Solver**): 

6	Define an IOSO project.When defining a project the following is done:
	• Dialog boxes with various questions open.
	• According to your answers on these questions, a `IOSOAdapter.cfg` file is created, which is placed along with *IOSO*-adapter and several libraries into the folder with the client part of the project.
	• *IOSO* runs and automatically creates a model for *FlowVision* (with a specified path to the project).
	• Then this project is used by *IOSO* as a template for all computational projects. This template will be copied on the server part with a new name and a new file with parameters.
	• When a *FlowVision* project is started via *IOSOAdapter*, the computation is controlled by the `command.txt` file. If there is no `command.txt` file in the project's directory, then it will be created automatically. Also you can create the `command.txt` file manually and specify the required commands in it (see section Command file). Here is an example of an automatically created file `command.txt`:
	`SS_PROJECTLOAD<./xxxxxxxx.fvproj><><><>`
	`SS_SOLVESTART<0>`
	`SS_DO_NOT_DISTURB`
	`SS_WRITE_FINAL_RTBL`
	`SS_PROJECTUNLOAD`
	`SS_SHUTDOWNSOLVER`

Description of the file `IOSOAdapter.cfg`

Parameter with an example of its value	Description

`CmdLineIndex=0`	The number (starting from **0**) of a line for staring **Solver** as it is specified in settings of **Solver Agent** (`FvSolverAgent.cfg`).
	Single-processor and multiprocessor lines are numbered independently. Use of a single-processor or a multiprocessor line is defined by the parameter `ProcNum`.
	Example:
	Assume that the file `FvSolverAgent.cfg` contains the following lines:
	`SAClientsPort=30950`
	`SASolversPort=30951`
	`SolverCmdLine=FvSolver64.exe;64-bit solver`
	`SolverCmdLineMPI="C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n %1 "C:\Program Files\FlowVision-3.13.01\FvSolver64.exe"`
	`mpi=libFvMPI_MSMPI_v7_x64.dll;64-bit Microsoft MPI solver`

Parameter with an example of its value	Description

	`SolverCmdLineMPI=job submit /numsockets:%1 /JobName:FlowVisionHPCSolver mpiexec.exe FvSolver64.exe mpi=libFvMPI_MSMPI_v7_x64.dll;64-bit Microsoft Compute Cluster solver`
	Then, for a single-processor run, the value **0** of this parameter corresponds to the following line for starting **Solver**:
	`SolverCmdLine=FvSolver64.exe;64-bit solver`
	and, for a multiprocessor run, the values **0** and **2** correspond to the following lines:
	`SolverCmdLineMPI="C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n %1 "C:\Program Files\FlowVision-3.13.01\FvSolver64.exe" mpi=libFvMPI_MSMPI_v7_x64.dll;64-bit Microsoft MPI solver`
	`SolverCmdLineMPI=job submit /numsockets:%1 /JobName:FlowVisionHPCSolver mpiexec.exe FvSolver64.exe mpi=libFvMPI_MSMPI_v7_x64.dll;64-bit Microsoft Compute Cluster solver`

`ProcNum=0`	Number of processors. The value **0** is applied for a single-processor run (a **Solver Agent**'s line for a single-processor run of **Solver** is used). When its value is more then zero, a line for multiprocessor run will be used.
`ProjectTimeout=180`	Time after which, if the computation is not finished, *IOSO* decides that the result from this computation has not been received.
`SAHost=fvserver2`	Host of **Solver Agent**
`SAPassword=1`	User's password of **Solver Agent**
`SAPort=30910`	Port of **Solver Agent**
`SAUser=1`	Username for **Solver Agent**
`ThreadsNum=0`	Number of threads. **0** means that number of cores is defined automatically; values **1** or more define a specific number of cores.
`GLOforDownload=`*files*	List of files that will be copied from the server part of the project to the client part at the termination of the computation. The list consists of names of the files separated by symbols "`;`".

Checking from *IOSO* how Solver terminates (normally or abnormally)

When **FinalRTBLAutoSave=Yes** is set in the configuration file of Solver (FvSolver.cfg), then, if the table of external parameters is used, *FlowVision* creates an `fvrtbl_f`-file.

Creation the `fvrtbl_f`-file means that *FlowVision*'s **Solver** terminated due to fulfillment its stopping condition and unloaded correctly.

If the **Solver** terminated abnormally, then the `fvrtbl_f`-file would not be created and the controlling external software would understand this situation.

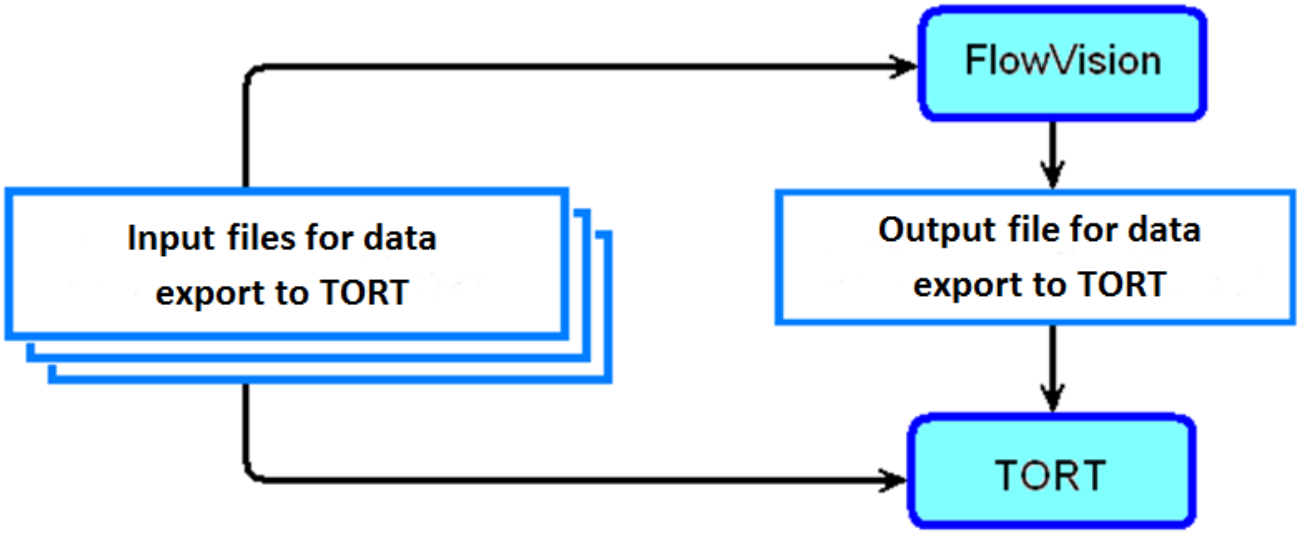
9.7 Neutron transfer (TORT)

A technology using a software complex composed of *FlowVision* and *TORT* has been developed to allow the calculation of neutron and proton radiation or gas emission in a liquid flow.

The *TORT* software can calculate the space-energy density distribution in particle flux (and their functionals) from various neutron and photon radiation emitters in the materials of calculated construction. The calculation is performed in 3D geometry models.

FlowVision is used for carrier liquid flow and radioactive isotope distribution calculation. The results of these calculations are transferred into TORT and serve as input data for radioactive emission calculation.

The data transfer from *FlowVision* to *TORT* uses an output file for data export to TORT, But a correct connection between *FlowVision* and *TORT* requires that the both programs receive data from one or more input files for data export to TORT.



FlowVision and *TORT* interaction

The *input files* for export to *TORT* are used *for* synchronization of coordinate systems and geometry models in *FlowVision* and *TORT*.

FlowVision can export the concentration of the delayed neutron precursors into a text file for *TORT* import at user-defined moments of time during coolant flow simulation. The original concentration in *FlowVision* is a mass property, i.e., nondimensional value in a range between 0 and 1. The *TORT* software requires the importation of the neutron quantity in cm^{3}. *FlowVision* requires appropriate user variables which values could be exported to *TORT*.

Launching a solver in single- or multiprocessor mode is performed via a batch file (with the `* .bat` extension). The following command line is used for export values of user variables from *FlowVision*:

- `TortExport.bat` - for single-processor version
- `TortExportMPI.bat` - for multiprocessor version

The structure of the batch file (`TortExport.bat`) is: **RUNMPI RUNSOLVER SALICENSE CMDFILE**, see descriptions of these parts in the table below:

Part of the `TortExport.bat` file	Description
RUNMPI	The command to run *MPI* (when **Solver** is used in a multiprocessor mode). ***Example:*** `"C:\MSMPI\Bin\mpiexec" -n 2` the last number corresponds to the number of processors on which **Solver** is to be run.
RUNSOLVER	The command to run **Solver**. It specifies the path to the **Solver**'s file `FvSolver.exe`.
SALICENSE	License name, `sa_license=licensename`.

Part of the `TortExport.bat` file	Description
CMDFILE	Name of a batch file with its path. `cmdfile = path /(name of the batch file).cmd`

Batch file **`(TortExport.cmd)`** contains the data to run. Its structure is as follows (see description in the table below):

```
SS_PROJECTLOAD <./ Part2ig4 / Part2ig4.fvproj> <> <> <>
SS_CUSTOM_COMMAND <TORT_EXPORT_1.0><tortindex.txt><tort.txt> <0>
SS_SHUTDOWNSOLVER
```

Part of the `TortExport.cmd` file	Description
SS_PROJECTLOAD <relative path to the project file on the solver> <> <> <>	load the specified project solver
SS_CUSTOM_COMMAND <TORT_EXPORT_1.0> <Input file name TORT> <output file name TORT> <0> *) Input file located in the server part of the project, the output file is produced in the same	
SS_SHUTDOWNSOLVER	unload solver

To create a joint *FlowVision*+*TORT* project, do the following:

Step	Actions
1	Create input files for data export to TORT [1)]
2	Create a *FlowVision* project that compute isotope distribution in the carrier liquid/gas - see Mass transfer, Equations.
3	Create user variables corresponding to concentrations the specific isotope types (quantities per 1 cubic centimeter).
4	Set the parameters of export to *TORT* in the **Export to TORT** element in the project tree or in batch mode [2)].
5	Perform the project computation in *FlowVision* and save the results [3)]
6	Perform the computation in *TORT* using data about the distribution of radioactive isotopes received from FlowVision output files.

Notes:

1) There can be several input files if it is required to carry out calculation for several sets of *TORT* input files.

2) Export of data to*TORT* is also possible in batch mode.

3) The output files for exporting data to *TORT* are autosaved at each scheduled or manual saving of *FlowVision* data.

9.7.1 Input file of FlowVision for data export to TORT

The input file must be located in the server-side and have the following form:

- **`F`** - service field
- **`model-1.out`** - name of the input *TORT*-file;
- **`<name_1> ... <name_N>`** - the names of user-defined variables.

Spaces (or other characters) between adjacent brackets should be omitted, that is, must be `<...><...>` and so on.

TORT-input file **`(filename.out)`** should be supplemented by communication between the coordinate systems *FlowVision* and *TORT*.

Communication systems and coordinate *FlowVision* and *TORT* is set as follows:

- `1000 **`= flag beginning of the array coordinate model *TORT*
- `0. 0. 0.` = Coordinates of the origin of the coordinate system in the model *TORT*, given in centimeters
- `1. 0. 0.` = X-axis of the coordinate system in the model *TORT*
- `0. 1. 0.` = Y-axis of the coordinate system in the model *TORT*

- `0. 0. 1.` = Z-axis of the coordinate system in the model *TORT*

9.7.2 Output file of FlowVision for data export to TORT

When you export data from *FlowVision* in *TORT* values of the selected user-defined variables, multiplied by the weighting factors, are automatically recorded in the output file[1)].

An array of weighting factors is given in TORT-input file.

The data in the output file are represented in the following form:

```
/ Name of a custom variable, step = number of time step, time = time
98 **
output array
t
```

Here

`98 **` = mark of the output array

`t` = mark of end of file.

Output array is written in text format and has the following structure:

S (I, J, K), I = 1, IM, J = 1, JM, K = 1, KM

Here IM, JM, KM = number of X, Y, Z coordinate planes model TORT.

To reduce the output file null values in the output arrays are counted and recorded as `"nr0."`,Where `n` - number of zero values, `r` - a sign of recurrence.

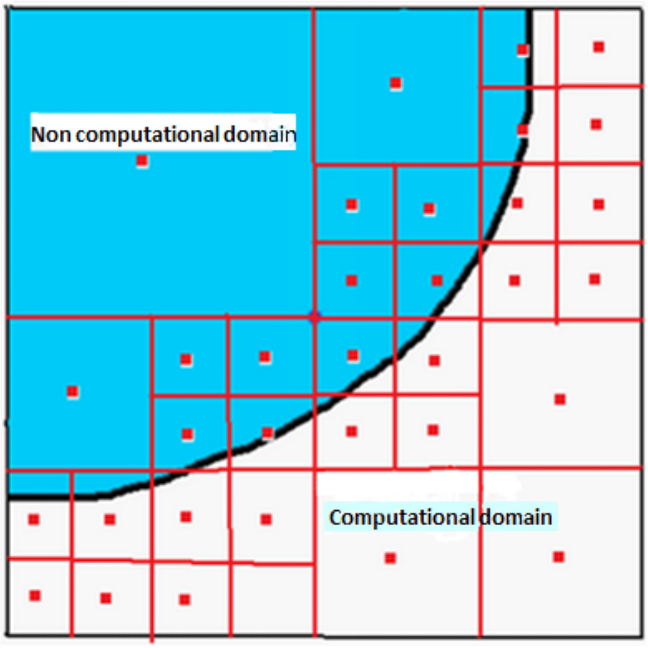
Note:

[1)] For each selected custom variable recorded its output file with a name that matches the name of a user variable. In the case of the history of conservation (see the sections Settings for automatic saving of calculation results and visualization data and Element «**Data autosave**"), The file name is attributed to more number conservation.

9.7.3 Method of calculation of isotope concentrations

In *FlowVision* there are two methods of calculating the concentration of isotopes in the transmission of software *TORT*.

When using the method without specifying the boundary cells in *TORT* transferred isotope concentrations calculated in the center of the cell.



When using the method with the definition of the boundary cells of the cell on the boundary of sodium and sodium-free zones are broken down to a specified level of fragmentation algorithm similar adaptation.

If the volume fraction of sodium in the cell entered in the input file,ε= 1, the concentration of isotopes calculated by the formula:

$$C_{out} = \frac{\sum_i C_i V_i}{V_{tort}}$$

If the volume fraction of sodium $0 < \varepsilon < 1$, the concentration of isotopes is calculated by the formula:

$$C_{out} = \frac{\sum_i C_i V_i}{\sum_i V_i} \varepsilon$$

where

C_i - The concentration of sodium in the i-th cell,

V_i - The volume of the i-th cell, the resulting cleavage

V_{tort} - Volume of the cell in *TORT*.

Summation is carried out only in cells whose centers were in the calculation (sodium) region *FlowVision*.

See also section: Element «Export to TORT».

9.8 Acoustics (LMS)

A *FlowVision* connection to the *LMS* software serves for calculating acoustic noise caused by pressure fluctuations in moving fluid.

Pressure fluctuations in fluid flow are simulated by *FlowVision*. Acoustic noise in simulated in *LMS*.

The connection between *FlowVision* and *LMS* software packages is carried out according to the surface of the boundary conditions selected. The surface of boundary conditions is saved to the **`ProjectName_mesh.CGNS`** file and pressure on it is saved to the **`ProjectName_nnnn.CGNS`** file starting from specified time at each time step number **nnnn**. Then, after the end of calculations, files are uploaded to an *LMS* project and acoustic noise is simulated using data obtained from *FlowVision*.

If necessary, the surfaces of boundary conditions can be transformed when uploading to *LMS*. To do so, transformation parameters should be set in the export parameters before calculating.

Follow the steps to create a joint *FlowVision+LMS* project:

Step	Actions
1	Open a *FlowVision* project.
2	Define parameters for export to LMS in the **Solver** tab of the project tree.
3	Perform the calculation in *FlowVision*.
4	Upload the received data into *LMS* (files **`ProjectName_mesh.CGNS`** and **`ProjectName_nnnn.CGNS`**).
5	Open the project in *LMS* and perform the computation.

9.9 Output of additional information during a joint computation

During a joint computation that is done by *FlowVision* and third party (external) software, *FlowVision* can output additional information into files that are stored in the server part of the project.

The additional information will not be output during a single (*FlowVision* only) computation.

To enable the output of the additional information, in the configuration file of Solver (FvSolver.cfg) specify:

```
DeformLog=Yes
LoadingsLog=Yes
```

The **DeformLog** enables output of files that *FlowVision* receives from the external program. These are files (***NNNN*** is a number):

- **mpm_exchange_*NNNN*.log**
- **mpm_iter_*NNNN*.log**

The **LoadingsLog** parameter enables output of files with information that *FlowVision* transfers to the external software. These are files **mpm_loadings_*NNNN*.log** (***NNNN*** is a number).

Files mpm_exchange_*NNNN*.log

Files **mpm_exchange_*NNNN*.log** are generated at each data exchange. The file names are formed according to exchange step numbers, so the name **mpm_exchange_0007.log** means that this file was received from the 7th exchange.

```
1  ================================================================================
2  EXCHANGE STEP 4
3  ================================================================================
4  Status.current: 0.0299999999999999989
5  Status.timeIter: 25
6  Status.timeStep: 0.00096864094837209511
7  MPM.m_LoadTime: 0.0299999999999999989
8
9  MPM.m_NextExchangeTime: 0.0400000000000000008
10
11 MPM.ExchangeTimeInterval0.0100000000000000002
12
13 --------------------------------------------------------------------------------
14 ASSEMBLY_DC-SURF1
15 --------------------------------------------------------------------------------
16 0.0108118377760930626   -0.00101003474890079808 -0.0370393569392422212  0
17 0.0108118604715777503   0.0190100176077545521   -0.0370393504605225285  0
18 0.00200000009000000012  0.0189999994731194777   0.00100000003376102785  0
19 0.00200000009000000012  -0.00100000012688051588 0.00099999979999996563 0
20 0.00606442239497175482  0.0189887634742225732   -0.0386077018088071752  0
21 -0.00300000002999999994 0.0189999994731194777   0.00100000003376102785  0
22 -0.00300000002999999994 -0.00100000012688051588 0.00099999979999996563 0
23 0.00606439861746548015  -0.000988769397863288341    -0.0386077048083420804  0
24 0.0120736100565578246   0.0189975974400958258   -0.047155083231815921   0
25 0.013970528315806325    0.0190037914558009724   -0.0465218233097990444  0
26 0.0139704942709292686   -0.00100381211723995405 -0.0465218355588979948  0
27 0.0120735752053643709   -0.000997612096701047711    -0.0471550925538102156  0
```

Example of a file **mpm_exchange_*NNNN*.log**

Each of these files contains the following information:

- **EXCHANGE STEP** - the current exchange step number.
- **Status.current** - the current time.
- **Status.timeIter** - count of the current time step.
- **Status.timeStep** - duration of the current time step.
- **MPM.m_LoadTime** - time when *FlowVision* sent the loadings.
- **MPM.m_NextExchangeTime** - time when the next exchange will occur.
- **MPM.ExchangeTimeInterval** - the current exchange step between the programs.

- **ASSEMBLY_DC-SURF1** - name of the exchange surface, for which coordinates of nodes are listed.
- Coordinates of nodes for the exchange surface

When several bodies are simulated, the further information in the file relates to the exchange surface and coordinates of this exchange surface.

Files mpm_iter_*NNNN*.log

Files **mpm_iter_*NNNN*.log** are also generated at each data exchange. The file names are formed according to current step numbers of *FlowVision*, so the name **mpm_iter_0052.log** that this file was received from the 52nd step of *FlowVision*.

As one exchange step might meet to several steps of *FlowVision*, the number of files **mpm_iter_*NNNN*.log** might be greater then number of files **mpm_exchange_*NNNN*.log**.

Each of the **mpm_iter_*NNNN*.log** files contains information about the time step, to which coordinates correspond, and at which exchange step it was.

```
1  ================================================================================
2  DISPLACEMENTS APPLIED AT ITERATION 11
3  ================================================================================
4  Exchange step: 11
5  Iteration per exchange step: 1
6  Current time: 1.1000000000000014e-05
7  Deformations load time: 1.0000000000000008e-05
8  Delta time: 1.0000000000000059e-06
9  Status.current: 1.0500000000000011e-05
10 Status.timeIter: 11
11 Status.timeStep: 5.0000000000000295e-07
12 Status.expTimeStep: 9.9999999999999955e-07
13 --------------------------------------------------------------------------------
14 Exchange
15 --------------------------------------------------------------------------------
16 TRANSFORMATION:
17 0.0010000000000000002  0   0   0
18 0   0.0010000000000000002  0   0
19 0   0   0.0010000000000000002  0
20 0   0   0   1
21
22 CURRENT COORDINATES_TR, COORDINATES_FE, VELOCITIES AND TEMPERATURES:
23 0.151423489444433235    0.258159550477292277    1.79137135358771941e-41 151.423489444433244 258.159550477292271
24 0.152016536353054482    0.258213313648707643    -2.964041540266475 82e-41    152.016536353054477 258.21331364870
25 0.151897188538685557    0.259164997871404512    -1.10539517612995774e-40    151.897188538685555 259.16499787140
26 0.151795435871936779    0.26012682585480873 -1.89259571366768828e-40    151.795435871936775 260.126825854808715
27 0.151692460795544631    0.261097826163865288    -2.57731609782213507e-40    151.692460795544633 261.09782616386
28 0.151582331539905107    0.262073769590792427    -3.25737282106024288e-40    151.582331539905113 262.07376959079
29 0.151461305941410052    0.263051754092055057    -3.88825961391191998e-40    151.461305941410046 263.05175409205
30 0.151320403039876344    0.264028999471057457    -4.6217879699072715e-40 151.320403039876339 264.028999471057432
31 0.15117283592498218 0.265007138382932494    -5.45973442666745659e-40    151.172835924982166 265.00713838293251
```

Example of a file **mpm_iter_*NNNN*.log**

Each of these files contains the following information:

- **DISPLACEMENTS APPLIED AT ITERATION** - iteration, on which the displacements were applied.
- **Exchange step** - count of the current exchange step.
- **Iteration per exchange step** - count of the step within the current exchange step (when more then one step occurs within the exchange).
- **Current time** - the time moment when the full values of deformations are achieved (**Solver** might make several steps during one exchange; also it can apply deformations smoothly).
- **Deformations load time** - the time moment when the loadings were transferred.
- **Delta time** - duration of the exchange step.
- **Status.current** - the current time in *FlowVision*.
- **Status.timeIter** - count of the current time step.
- **Status.timeStep** - duration of the current step until the **Current time** time moment.
- **Status.expTimeStep** - the explicit time step in *FlowVision*.

Then information about exchange surfaces goes:

- Name of the exchange surface, for example, **Exchange** or **DC-SURF**
- Transformation matrix that are applied to the body: **TRANSFORMATION**

- Coordinates, to which the matrix is applied (3 components): **`CURRENT COORDINATES_TR`**
- Coordinates from the FE package (3 components): **`COORDINATES_FE`**
- Velocities (3 components): **`VELOCITIES`**
- Temperature: **`TEMPERATURES`**
- After the coordinates the integral velocity for the body (3 components) follows: **`AVERAGE VELOCITY`**

Files mpm_loadings_*NNNN*.log

Files **`mpm_loadings_`*`NNNN`*`.log`** are generated at each exchange step of *FlowVision*. The file names are formed according to the current step of *FlowVision*, on which the exchange occurred, so the name **`mpm_loadings_0034.log`** means that this file was received from the 34th step of *FlowVision* and that data exchange between *FlowVision* and the external software took place on this step.

```
================================================================================
EXCHANGE STEP 4
================================================================================
Deformations load time: 0.0299999999999999989
Status.current: 0.0400000000000000008
Status.timeIter: 34
Status.timeStep: 0.000933155143314622071
Status.expTimeStep: 0.0100000000000000002

Pressure relaxation: 1
Temperature relaxation: 1

For "ASSEMBLY_DC-SURF1":
Integral force FV: 547.18080090302  -0.0026138590680478 174.57029848784
Integral force predicted: 547.18080090302   -0.0026138590680478 174.57029848784

POINT   POINT.x POINT.y POINT.z FORCE.x FORCE.y FORCE.z
0   0.010811832776093   -0.0010100397489008 -0.037039361939242  -0.095010034558987  0.00042822202770122 -0.031632676542356
1   0.010811855971578   0.019010013107755   -0.037039354960523  -0.095014576279218  -0.00040353617966633    -0.031633894958922
2   0.00199999609   0.018999995473119   0.00099999603376103 0   0   0
3   0.00199999659   -0.0010000036268805 0.00099999628   -0.017017673575804  0.00010701500204337 -0.00029356804692
4   0.0060644193949718  0.018988760474223   -0.038607704808807  0.2810177733722 0.0013208689945151  0.13785295197854
5   -0.00300000253  0.018999996973119   0.00099999753376103 0   0   0
6   -0.00300000203  -0.0010000021268805 0.00099999778   0.077715100114466   -0.00062435077250178    0.0014756315069652
7   0.0060643971174655  -0.00098877089786329    -0.038607706308342  0.28099943361231    -0.0013716869428972 0.13783998490268
8   0.012073609056558   0.018997596440096   -0.047155084231816  0.043832267166973   0.00021599515280613 0.067412234262379
9   0.013970527815806   0.019003790955801   -0.046521823809799  -0.059818814563008  -0.00015242613525538    0.016242147528669
10  0.013970494270929   -0.00100381211724   -0.046521835558898  -0.059806199706582  0.00015275251993482 0.016238899256088
11  0.012073575705364   -0.00099761159670105    -0.04715509205381   0.043803097746859   -0.00021654090775166    0.067375214748467
12  0.01081764941278    0.00052921347345672 -0.037036738830491  -0.37149619620463   0.0015735548152276  -0.12361409588183
13  0.010823461981738   0.0020685308541566  -0.037035031604176  -0.55302001794556   0.0021384213967529  -0.18388621201076
```

Example of a file **`mpm_loadings_`*`NNNN`*`.log`**

Each of these files contains the following information:

- **`DISPLACEMENTS APPLIED AT ITERATION 11`** - iteration at which displacements were applied.
- **`Exchange step`** - count of the current exchange step.
- **`Deformations load time`** - time when the loadings were transferred.
- **`Status.current`** -the current time.
- **`Status.timeIter`** - count of the current time step.
- **`Status.timeStep`** - duration of the current time step.
- **`Status.expTimeStep`** - explicit time step in *FlowVision*.
- **`Pressure relaxation`** - loads relaxation coefficient.
- **`Temperature relaxation`** - temperature relaxation coefficient.

Then information about exchange surfaces goes:

- **`For "`*`Exchange_surface_name`*`"`** - name of the exchange surface.
- **`Integral force FV`** - the integral force acting on the body from *FlowVision*.
- **`Integral force predicted`** - the integral force acting on the body from *FlowVision*, that *FlowVision* predicts to the next step.
- **`POINT`** - count of the node.
- **`POINT.x POINT.y POINT.z`** - coordinates of the node (3 components).
- **`FORCE.x FORCE.y FORCE.z`** - forces acting on the node (3 components).

9.10 MBC (Moving Body Controller)

MBC (Moving Body Controller) is an application (an FEA code) that is used for interaction with *FlowVision* in FSI computations.

MBC includes two blocks:

1. Communication block, which provides data exchange with *FlowVision* through a network channel.
2. Calculation block, which provides calculations for output values transferred to *FlowVision* (for example, coordinates of nodes) with taking into account the input values received from *FlowVision* (for example, loadings in the nodes).

Application *MBC* is written on the programming language *C++* with use of multiplatform library *Qt* (version *4*). *MBC* has no user interface and works automatically according to its algorithm. *MBC* is supplied as a kit of source codes completely ready to compilation.

Also you have to install the *Qt* library (version *4.7.3* or more recent) on the computer. You have to create the environment variable `QTDIR`, which will store the path to the installed *Qt* library.

10 Theory

The following topics are discussed in the current chapter:

- simulation of flows in the *FlowVision* suite of software (basic principles are described)
- **Substance** properties
- **Phase** properties
- purpose of **Modifiers**
- **Physical processes** implemented in *FlowVision*
- *FlowVision* capability to simulate **Physical processes** in narrow channels without resolving these channels by computational grid
- *FlowVision* capability to simulate **Physical processes** when a **Dispersed Phase** is present in the computational domain
- *FlowVision* capability to improve solution accuracy near solid surface by means of a boundary level grid
- numerical methods implemented in *FlowVision*

10.1 Basic notations

Below the notations frequently met in the given chapter are listed:

Notation	Physical quantity	Name in *FlowVision*	Dimension
c	sonic speed	**Sonic speed**	m s^{-1}
C_p	specific heat at constant pressure	**Specific heat**	J kg^{-1} K^{-1} = = m^2 s^{-2} K^{-1}
D	diffusion coefficient		m^2 s^{-1}
$\hat{\boldsymbol{S}}$	strain rate tensor		s^{-1}
F	variable VOF	**VOF**	
$\boldsymbol{g}$	gravity acceleration (vector)	**Gravity vector**	m s^{-2}
h	thermodynamic enthalpy [1)]	**Enthalpy**	m^2 s^{-2}
h_0	enthalpy of **Substance** formation at 298.15 K	**Formation enthalpy**	m^2 s^{-2}
$H = h + \frac{\boldsymbol{V}_{abs}^2}{2}$	total enthalpy	**EnthalpyTotal**	m^2 s^{-2}
h_g	hydrostatic level [2)]		m
$\boldsymbol{J}_q$	specific heat flux (vector)	**HeatFlux**	W m^{-2} = kg s^{-3}
k	turbulent energy	**TurbulentEnergy**	m^2 s^{-2}
$k = 1.3806504 \cdot 10^{-23}$	Boltzmann constant		J K^{-1} =kg m^2 s^{-2} K^{-1}
L	characteristic linear size		m
m	molar mass	**Molar mass**	kg $mole^{-1}$
$M = \frac{\lvert\boldsymbol{V}\rvert}{c}$	Mach number	**MachNumber**	
$\boldsymbol{n}$	internal normal to cell face (vector)		
P	relative static pressure	**Pressure**	Pa
P_{ref}	reference pressure [2)]		Pa
P_{hst}	hydrostatic pressure		Pa
P_{+hst}	relative pressure + hydrostatic pressure	**Pressure (+hydrostatic)**	Pa
$P_{abs} = P + P_{ref} + P_{hst}$	absolute static pressure		Pa
P_{tot}	total relative pressure [3)]	**PressureTotal**	Pa
$P_{tot,abs} = P_{tot} + P_{ref}$	total absolute pressure		Pa
P_s	saturation pressure	**Saturation pressure**	Pa
$Pr = \frac{\mu C_p}{\lambda}$	molecular Prandtl number	**Prandtl**	

Notation	Physical quantity	Name in *FlowVision*	Dimension
$Pr_t = \frac{\mu_t C_p}{\lambda_t}$	turbulent Prandtl number	**PrandtlTurb**	
$R_A = 8.31441$	universal gas constant		J mole^{-1} K^{-1}
$Re = \frac{\rho \lvert \boldsymbol{V} \rvert L}{\mu}$	Reynolds number		
$\boldsymbol{r}$	radius-vector whose origin coincides with the rotation center of **Region** or **Boundary condition**		m
$Sc = \frac{\mu}{\rho D}$	molecular Schmidt number	**Schmidt**	
$Sc_t = \frac{\mu_t}{\rho D_t}$	turbulent Schmidt number	**SchmidtTurb**	
t	time		s
T	relative temperature	**Temperature**	K
T_{ref}	reference temperature[2] (absolute)	**Temperature**	K
$T_{abs} = T + T_{ref}$	absolute temperature		K
T_{tot}	total temperature [3] [4] (relative)	**TemperatureTotal**	K
$T_{tot,abs} = T_{tot} + T_{ref}$	total absolute temperature		K
T_B	boiling temperature (absolute)	**Boiling temperature**	K
T_M	melting temperature (absolute)	**Melting temperature**	K
u_τ	dynamic velocity		m s^{-1}
$\boldsymbol{V}$	relative velocity in moving coordinate system (vector)	**Velocity**	m s^{-1}
$\boldsymbol{V}_{abs}$	absolute velocity (vector)	**Velocity**	m s^{-1}
$\lvert \boldsymbol{V} \rvert$	modulus of relative velocity		m s^{-1}
$\lvert \boldsymbol{V}_{abs} \rvert$	modulus of absolute velocity		m s^{-1}
W_f	rate of combustion reaction	**Reaction rate**	kg m^{-3} s^{-1}
X_i	molar fraction of **Substance** i (= volume fraction in the case of gas mixture)	**Mol. fraction_Substance #i**	
Y_i	mass fraction of **Substance** i	**Mass. fraction_Substance #i**	
y	distance to nearest wall	**Distance to wall**	m
$y^+ = \frac{u_\tau y}{\nu}$	dimensionless distance to the nearest wall	**Y_plus**	

Notation	Physical quantity	Name in *FlowVision*	Dimension
k_d	permeability		m^{-2}
$\gamma = \frac{C_P}{C_P - R_A/m}$	specific rate ratio for gas mixture		
λ	coefficient of thermal conductivity (molecular)	**Thermal conductivity**	kg m s^{-3} K^{-1}
μ	dynamic coefficient of viscosity (molecular)	**Viscosity**	Pa s = kg m^{-1} s^{-1}
μ_t	dynamic coefficient of viscosity (turbulent)	**ViscosityTurb**	Pa s = kg m^{-1} s^{-1}
$\nu = \mu / \rho$	kinematic coefficient of viscosity (molecular)		m^2 s^{-1}
$\nu_t = \mu_t / \rho$	kinematic coefficient of viscosity (turbulent)		m^2 s^{-1}
ρ	density	**Density**	kg m^{-3}
ρ_g	hydrostatic density [2)]	**g-Density**	kg m^{-3}
σ	surface tension coefficient	**Surface tension** (in properties of a **Substance**) **SurfTension, value** (in properties of an element **Model #i > Phase interaction > Continuum-Continuum** or **> Continuum-Vacuum**)	N m^{-1}
σ_{LJ}	1st parameter of Lennard-Jones potential	**LJ6-12 Sigma**	Å = 10^{-10}m (Ångström)
ε_{LJ}/k	2nd parameter of Lennard-Jones potential	**LJ6-12 Epsilon/k**	K
ε	rate of dissipation of turbulent energy	**TurbDissipation**	m^2 s^{-3}
$\boldsymbol{\omega}$	velocity of rotation of **Region** or **Boundary condition** (vector)		rad s^{-1}
$\hat{\boldsymbol{\tau}}$	shear stress tensor		Pa
$\boldsymbol{\tau}_w$	Viscous force acting on a solid surface specific per unit of area (viscous stress). This value can be negative.	**Shear stress**	Pa

Indices:

b - boundary value
cell - value in cell center
G - Gas (aggregative state)
ini - initial value
inl - inlet value
L - Liquid (aggregative state)
n - value at time layer n (in numerical method)
n+1 - value at time layer n+1 (in numerical method)
x - tangential (to boundary) component of vector
y - normal (to boundary) component of vector
w - value at wall
∞ - free stream value

Comments:

[1)] Property **Enthalpy** is absent in folder **Preprocessor > Substances > Substance #i**. It is computed as follows:

$$h = h_0(298.15) + \int_{298.15}^{T_{abs}} C_p dT$$

[2] Reference quantities are specified in **General settings**.

[3] See details: for gas in the section **Substance properties, Gas**, for liquid in the section **Substance properties, liquid**.

[4] In the case of constant specific heat

$$T_{tot} = T + \frac{V_{abs}^2}{2C_p}$$

In the case of variable specific heat the total temperature is found from:

$$h(T_{tot}) = H$$

10.2 Properties of Substances

A **Substance** is determined by its aggregative state and physical properties. The aggregative state determines:

1. set of physical properties
2. method of computing the sonic speed
3. method of computing total pressure
4. state equation

Three aggregative states of a **Substance** are defined in the *FlowVision* software suit:

- Solid
- Liquid
- Gas

Enthalpy

Enthalpy is defined by the formula:

$$h(t) = h_0 + \int_{T_0}^{T} C_p(s)ds$$

where

h_0 is the enthalpy of formation at temperature T_0

C_p is the specific heat capacity at constant pressure

If C_p is a constant, then the formula, which defines the enthalpy, will be:

$$h(t) = h_0 + C_p(T - T_0)$$

In this case the enthalpy is a linear function.

In the general case, the enthalpy of formation h_0 might depend on the recovered oxidizer excess factor (the variable **Oxid. excess factor rec.**, α).

The table below lists possible combinations of expressions for the enthalpy of formation h_0, specific heat capacity C_p, and resulting expression for the enthalpy h.

h_0	C_p	h
constant	constant	linear function of T
	table of dependency on T	table of dependency on T
	table of dependency on P and T	table of dependency on P and T
	polynome of dependency on T	polynome depending on T
table of dependency on α	constant	linear function of T
	table of dependency on T	table of dependency on α and T
	table of dependency on P and T	table of dependency on α, P and T
	polynome of dependency on T	polynome of dependency on T

This section has individual numeration of equations.

10.2.1 Solid

Physical properties of Solid body

Notation	Physical quantity	Name in *FlowVision*	Dimension
m	molar mass	**Molar mass**	kg mole^{-1}
ρ	density	**Density**	kg m^{-3}
λ	coefficient of thermal conductivity	**Thermal conductivity**	kg m s^{-3} K^{-1}
C_p	specific heat capacity	**Specific heat**	J kg^{-1} K^{-1} = m^2 s^{-2} K^{-1}
h_0	enthalpy of formation at 298.15 K	**Enthalpy of formation**	m^2 s^{-2}
T_M	melting temperature	**Melting temperature**	K
S_s	temperature spectrum of solidification	**Solidification spectrum**	K
σ_e	specific electrical conductivity	**Conductivity**	S m^{-1} = Ω^{-1} m^{-1}
ε	relative dielectric permittivity	**Permittivity**	
μ	relative magnetic permeability	**Permeability**	

10.2.2 Liquid

Physical properties of Liquid

Notation	Physical quantity	Name in *FlowVision*	Dimension
m	molar mass	**Molar mass**	kg mole^{-1}
ρ	density	**Density**	kg m^{-3}
μ	dynamic coefficient of viscosity	**Viscosity**	Pa s = kg m^{-1} s^{-1}
λ	coefficient of thermal conductivity	**Thermal conductivity**	kg m s^{-3} K^{-1}
C_p	specific heat capacity	**Specific heat**	J kg^{-1} K^{-1} = m^2 s^{-2} K^{-1}
h_0	enthalpy of formation at 298.15 K	**Enthalpy of formation**$^{*)}$	m^2 s^{-2}
T_B	boiling temperature	**Boiling temperature**	K
σ	surface tension	**Surface tension**	N m^{-1}
σ_e	specific electrical conductivity	**Conductivity**	S m^{-1} = Ω^{-1} m^{-1}
ε	relative dielectric permittivity	**Permittivity**	
μ	relative magnetic permeability	**Permeability**	

*)

If chemical reactions, combustion or phase transitions are not taken into account in the computation, then it is recommended for the used **Substance**, which has been loaded from the **Substance Database**, to set zero **Enthalpy of formation** (h_0). This will improve accuracy of the simulation.

Sonic speed in liquid:

$$c = \sqrt{\frac{1}{\chi\rho}} \qquad \text{(Liquid.1)}$$

Total (relative) pressure in liquid:

$$P_{tot} = P + \frac{1}{2}\rho|V_{abs}|^2 \qquad \text{(Liquid.2)}$$

All these properties are specified in the *FlowVision* interface by constant, formula or two-parametric table. Besides that, for some properties other frequently used methods of their computation are implemented. These properties and methods are listed in the table below.

Property	Permissible values	Comment
Viscosity	• **PL** • **PL2** • **HB** • **BC**	Methods **PL**, **PL2**, **HB**, **BC** are used in simulation of flows of non-Newtonian fluids (liquid rubber, plastic materials, mud, snow, etc.).

The following expressions use quantity S, which can be interpreted as the modulus of generalized velocity gradient:

$$S = 2\sum_{i,j} S_{ij}S_{ij} \qquad \text{(Liquid.3)}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right) \qquad \text{(Liquid.4)}$$

Preprocessor > Substances > Substance #i > Viscosity > PL

Non-Newtonian fluid model *Power Law* assumes the following format for the dynamic coefficient of viscosity:

$$\mu_{NN} = \kappa \cdot S^{\frac{n-1}{2}} \cdot e^{T_0/T_{abs}} \qquad \text{(Liquid.5)}$$

$$\mu_{min} < \mu_{NN} < \mu_{max} \qquad \text{(Liquid.6)}$$

The user specifies parameters:

Parameter in the interface	Value in the formula	Description
n	n	Exponent
mu_min	$\mu_{\min}$	Minimal viscosity
mu_max	$\mu_{\max}$	Maximal viscosity
kappa	κ	Constant κ
T_0	T_0	Temperature of activation

Preprocessor > Substances > Substance #i > Viscosity > PL2

Non-Newtonian fluid model *Power Law 2* assumes the following format for the dynamic coefficient of viscosity:

$$\mu_{NN} = \frac{\mu_0}{1 + \left(\frac{\mu_0\sqrt{S}}{\tau_0}\right)^{1-n}} \qquad \text{(Liquid.5bis)}$$

$$\mu_0 = \kappa \cdot e^{T_0/T_{abs}} \qquad \text{(Liquid.5bis2)}$$

$$\mu_{min} < \mu_{NN} < \mu_{max} \qquad \text{(Liquid.6bis)}$$

The user specifies parameters:

Parameter in the interface	Value in the formula	Description
n	n	Exponent
mu_min	$\mu_{\min}$	Minimal viscosity
mu_max	$\mu_{\max}$	Maximal viscosity
kappa	κ	Constant κ
T_0	T_0	Temperature of activation
tau0	τ_0	Reference strain

Preprocessor > Substances > Substance #i > Viscosity > HB

Non-Newtonian fluid model *Herschel-Bulkley* assumes the following format for the dynamic coefficient of viscosity:

$$\mu_{NN} = \mu_{max} \quad \text{if} \quad \sqrt{S} \le \tau_0 / \mu_{max}$$

$$\mu_{NN} = \frac{\tau_0 + \kappa\left(\left(\sqrt{S}\right)^n - \left(\tau_0 / \mu_{max}\right)^n\right)}{\sqrt{S}} \quad \text{if} \quad \sqrt{S} > \tau_0 / \mu_{max} \qquad \text{(Liquid.7)}$$

$$\mu_{min} < \mu_{NN} < \mu_{max} \qquad \text{(Liquid.8)}$$

The user specifies parameters:

Parameter in the interface	Value in the formula	Description
n	n	Exponent
mu_min	$\mu_{\min}$	Minimal viscosity
mu_max	$\mu_{\max}$	Maximal viscosity
kappa	κ	Constant κ
tau0	τ_0	Strain yield

Preprocessor > Substances > Substance #i > Viscosity > BC

Non-Newtonian fluid model *Bird-Carreau* assumes the following format for the dynamic coefficient of viscosity:

$$\mu = e^{-\alpha\left(T_{abs} - T_{ref}\right)}\left[\mu_{min} + \left(\mu_{max} - \mu_{min}\right)\left(1 + \lambda^2 S\right)^{\frac{n-1}{2}}\right] \qquad \text{(Liquid.9)}$$

$$\mu_{min} < \mu_{NN} < \mu_{max} \qquad \text{(Liquid.10)}$$

The user specifies parameters:

Parameter in the interface	Value in the formula	Description
n	n	Exponent
mu_min	$\mu_{\min}$	Minimal viscosity

Parameter in the interface	Value in the formula	Description
mu_max	μ_{max}	Maximal viscosity
lambda	λ	Relaxation time
alpha	α	T coefficient
T_ref	T_{ref}	Reference temperature of the non-Newtonian fluid model*)

*) Here the reference temperature T_{ref} of the non-Newtonian fluid model is specified. Generally it differs from the **Reference temperature**, which is specified by the parameter **Preprocessor > General settings > Reference values > Temperature**.

10.2.3 Gas

Physical properties of Gas:

Notation	Physical quantity	Name in FlowVision	Dimension
m	molar mass	**Molar mass**	kg mole^{-1}
ρ	density	**Density**	kg m^{-3}
μ	dynamic coefficient of viscosity	**Viscosity**	Pa s = kg m^{-1} s^{-1}
λ	coefficient of thermal conductivity	**Thermal conductivity**	kg m s^{-3} K^{-1}
C_p	specific heat capacity	**Specific heat**	J kg^{-1} K^{-1} = m^2 s^{-2} K^{-1}
h_0	enthalpy of formation at 298.15 K	**Enthalpy of formation***)	m^2 s^{-2}
P_s	saturation pressure	**Saturation pressure**	Pa
σ_{LJ}	parameter of the Lennard-Jonse 6-12 potential for molecular interaction	**LG6-12 Sigma**	Å = 10^{-10} m (Ångström)
ε_{LJ}/k	parameter of the Lennard-Jonse 6-12 potential for molecular interaction	**LG6-12 Epsilon/k**	K
σ_e	specific electrical conductivity	**Conductivity**	S m^{-1} = Ω^{-1} m^{-1}
ε	relative dielectric permittivity	**Permittivity**	
μ	relative magnetic permeability	**Permeability**	

*)
If chemical reactions, combustion or phase transitions are not taken into account in the computation, then it is recommended for the used **Substance**, which has been loaded from the **Substance Database**, to set zero **Enthalpy of formation** (h_0). This will improve accuracy of the simulation.

All these properties are specified in the *FlowVision* interface by constant, formula or two-parametric table. Besides that, for separate properties other frequently used methods of their computation are implemented. These properties and methods are listed in the table below.

Property	Possible alternative methods	Comment
Density	• **Ideal gas law**	**Density** is determined by the ideal gas law.

Property	Possible alternative methods	Comment
Viscosity	• **Polynom** • **Blottner**	Computation of dynamic coefficient of viscosity using methods **Polynom** and **Blottner** is recommended for simulation of hot gas flows accompanied by nonequilibrium chemical reactions, in particular, it is recommended for hypersonic problems.
Thermal conductivity	• **Polynom** • **f(mu, Cp)**	Computation of thermal conductivity using methods **Polynom** and **f(mu, Cp)** is recommended for simulation of hot gas flows accompanied by nonequilibrium chemical reactions, in particular, it is recommended for hypersonic problems.
Specific heat	• **Polynom**	Computation of specific heat capacity using method **Polynom** is recommended for simulation of hot gas flows accompanied by nonequilibrium chemical reactions, in particular, it is recommended for hypersonic problems.
Enthalpy	• **SDTable f(P,T)** • **Linear function**	For **Substances** downloaded from a **Substance Database**, **Enthalpy** is specified by a table of dependency on **Pressure** and **Temperature** from the **Substance Database**. This table is read-only, you can view it but not edit. For **Substances** created in **Preprocessor**, **Enthalpy** is calculated automatically as a linear function of **Temperature**. Parameters of this function are enthalpy of formation h_0 and the specific heat capacity C_p at constant pressure (see details in the section Properties of Substances). So you cannot specify **Enthalpy** in the user interface of **Pre-Postprocessor**, but you can specify it at creating the **Substance** in the **Substance Database Editor**.

Preprocessor > Substances > Substance #i > Density > Ideal gas law

$$\rho = \frac{P_{abs} m}{R_A T_{abs}} \qquad \text{(Gas.1)}$$

User specifies nothing.

Preprocessor > Substances > Substance #i > Viscosity > Polynom

The polynom has the following format:

$$\mu = E\, exp\left(A \cdot ln\, T + B / T + C / T^2 + D\right) \qquad \text{(Gas.2)}$$

Five numbers E, A, B, C, D are specified for two temperature intervals [T_1,T_2] and [T_2,T_3].

By default, T_1=300K, T_2=1000K, T_3=5000K:

$$300\ K \le T \le 1000\ K$$
$$1000\ K \le T \le 5000\ K \qquad \text{(Gas.3)}$$

Preprocessor > Substances > Substance #i > Viscosity > Blottner

The Blottner formula has the following format:

$$\mu = 0.1 exp\left(\left(A \cdot ln\, T + B\right) ln\, T + C\right) \quad \text{(Gas.4)}$$

The user specifies three numbers: A, B, C.

Preprocessor > Substances > Substance #i > Thermal conductivity > Polynom.

The polynom has the following format:

$$\lambda = E\, exp\left(A \cdot ln\, T + B / T + C / T^2 + D\right) \quad \text{(Gas.5)}$$

Five numbers E, A, B, C, D are specified for two temperature intervals:

$$\begin{aligned} &300\ K \le T \le 1000\ K \\ &1000\ K \le T \le 5000\ K \end{aligned} \quad \text{(Gas.6)}$$

Preprocessor > Substances > Substance #i > Thermal conductivity > f(mu, Cp).

The formula has the following format:

$$\lambda = \frac{R_A}{m} \mu \left[\frac{15}{4} + \frac{1}{Sc} \left(\frac{m C_p}{R_A} - \frac{5}{2} \right) \right] \quad \text{(Gas.7)}$$

User specifies individual Schmidt number Sc for the given **Substance**.

Preprocessor > Substances > Substance #i > Specific heat > Polynom.

The polynom has the following format:

$$C_p = \frac{R_A}{m} \left(A_1 + A_2 T + A_3 T^2 + A_4 T^3 + A_5 T^4 \right) \quad \text{(Gas.8)}$$

Here T is the absolute temperature (index *abs* is omitted). Five numbers $A_1 \div A_5$ are specified for five temperature intervals:

$$\begin{aligned} &300\ K \le T \le 1000\ K \\ &1000\ K \le T \le 6000\ K \\ &6000\ K \le T \le 15000\ K \\ &15000\ K \le T \le 25000\ K \\ &25000\ K \le T \le 30000\ K \end{aligned} \quad \text{(Gas.9)}$$

In the range T < 300 [K] the specific heat capacity is constant computed by formula

$$C_{p,min} = \frac{R_A}{m} \left(A_1 + A_2 T_1 + A_3 T_1^2 + A_4 T_1^3 + A_5 T_1^4 \right) \quad \text{(Gas.10)}$$

where $T_1 = 300\ K$.

In the range $T >$ 30000 [K] the specific heat capacity is constant computed by formula

$$C_{p,max} = \frac{R_A}{m} \left(A_1 + A_2 T_5 + A_3 T_5^2 + A_4 T_5^3 + A_5 T_5^4 \right) \quad \text{(Gas.11)}$$

where T_5=30000 [K]. The numbers specified in the interface as well as the enthalpy of formation determine the specific enthalpy of the given **Substance** as function of the absolute temperature.

Calculation of total pressure and temperature for gas

Suppose that in some point in the space the following parameters of the gas flow are known: pressure (P), temperature (T), velocity ($\mathbf{V}$), and total enthalpy (H).

As it is known,

$$H = h(P,T) + \frac{\mathbf{V}^2}{2}$$

Here h is thermodynamic enthalpy.

The total temperature T_{tot} and the total pressure P_{tot} are found using the following iteration method:

$$H = h(P^{(i-1)}, T^{(i)}) + \frac{(N-i)}{N}\frac{\mathbf{V}^2}{2}$$

$$P^{(i)} = P^{(i-1)}\left(\frac{T^{(i)}}{T^{(i-1)}}\right)^{\alpha}$$

(i= 1...N)

Here:

$P^{(0)}$=P

P_{tot}=$P^{(N)}$

$T^{(0)}$=T

T_{tot}=$T^{(N)}$

α=$C_p(P^{(i\text{-}1)}, T^{(i\text{-}1/2)})\cdot m(P^{(i\text{-}1)}, T^{(i\text{-}1/2)})/R$

$T^{(i\text{-}1/2)}$=$(T^{(i)}+T^{(i\text{-}1)})/2$

C_p is the specific heat

m is the molar mass

R is the universal gas constant

For the *simple* method of calculation, the number N is 1.

For the *exact* method of calculation, the number N is selected in the way so the values P_{tot} and T_{tot} be found with accuracy of 0.01%.

Use of either simple or exact method of calculation the total pressure and total temperature is set by the parameter **Numerical method > Total pressure** in the Advanced settings of Solver.

10.3 Phase propertries

This section has individual numeration of equations.

In general case, a **Phase** is a mixture of **Substances**. The properties and fractions of the **Substances** determine the **Phase** properties.

Notation	Physical quantity	Name in *FlowVision*	Dimension
μ	dynamic coefficient of viscosity	**Viscosity**	Pa s = kg m^{-1} s^{-1}
λ	coefficient of thermal conductivity	**Thermal conductivity**	kg m s^{-3} K^{-1}
C_p	specific heat capacity	**Specific heat**	J kg^{-1} K^{-1} = m^2 s^{-2} K^{-1}
ρ	density	**Density**	kg m^{-3}
m	molar mass	**Molar mass**	kg $mole^{-1}$
h	thermodynamic enthalpy	**EnthalpyTotal** [1)]	$m^2 \cdot s^{-2}$
H	total enthalpy	**EnthalpyTotal**[1)]	
c	sonic speed	**Sonic speed**	$m \cdot s^{-1}$
M	Mach number	**MachNumber**	
	total temperature	**TemperatureTotal**	K
P_{tot}	total pressure	**PressureTotal**	Pa

[1)] When **Preprocessor > Phases > Phase #i > Physical processes > Heat transfer > Heat transfer via h** is selected, the name **EnthalpyTotal** is used for visualization of thermodynamic enthalpy. When **Preprocessor > Phases > Phase #i > Physical processes > Heat transfer > Heat transfer via H** is selected, it is used for visualization of total enthalpy.

Computing methods for **Phase** properties **Specific heat**, **Density**, **Molar mass**, **EnthalpyTotal, MachNumber, TemperatureTotal, PressureTotal** are uniquely determined:

$$C_p = \sum_{i=1}^{N} Y_i C_{p,i} \qquad \text{(Phase.1)}$$

$$\rho = \left(\sum_{i=1}^{N} \frac{Y_i}{\rho_i} \right)^{-1} = \sum_{i=1}^{N} X_i \rho_i \qquad \text{(Phase.2)}$$

$$m = \sum_{i=1}^{N} m_i X_i = \left(\sum_{i=1}^{N} \frac{Y_i}{m_i} \right)^{-1} \qquad \text{(Phase.3)}$$

$$h = \sum_{i=1}^{N} Y_i h_i \qquad \text{(Phase.4)}$$

$$H = \sum_{i=1}^{N} Y_i h_i + \frac{1}{2} \mathbf{V}^2 \qquad \text{(Phase.5)}$$

$$c = \sqrt{\gamma \frac{R_A T_{abs}}{m}} \quad \text{for Gas} \qquad \text{(Phase.6)}$$

$$M = c / |\mathbf{V}| \quad \text{for Gas} \qquad \text{(Phase.7)}$$

$$P_{tot} = P_{abs}\left(\frac{T_{tot}+T_{ref}}{T_{abs}}\right)^{\frac{C_P}{R_A/m}} - P_{ref}$$ for Gas (Phase.8)

$$P_{tot} = P + \frac{1}{2}\rho \boldsymbol{V}^2$$ for Liquid (Phase.9)

$$H = h(T_{tot})$$ (Phase.10)

There is choice for Phase properties **Viscosity** and **Thermal conductivity** in window **Preprocessor > Phases > Phase #i**:

Property	Permissible values	Comment
Viscosity of mixture	• **Mass weighting** • **Molar weighting** • **Wilke**	Computation of dynamic coefficient of viscosity by the **Wilke** formula is recommended for simulation of hot gas flows accompanied by nonequilibrium chemical reactions, in particular, it is recommended for hypersonic problems.
Thermal conductivity of mixture	• **Mass weighting** • **Molar weighting** • **Mason-Saxena**	Computation of thermal conductivity by the **Mason-Saxena** formula is recommended for simulation of hot gas flows accompanied by nonequilibrium chemical reactions, in particular, it is recommended for hypersonic problems.

Preprocessor > Phases > Phase #i > Viscosity of mixture > Mass weighting

$$\mu = \sum_{i=1}^{N} Y_i \mu_i$$ (Phase.11)

Preprocessor > Phases > Phase #i > Viscosity of mixture > Molar weighting

$$\mu = \sum_{i=1}^{N} X_i \mu_i$$ (Phase.12)

Preprocessor > Phases > Phase #i > Viscosity of mixture > Wilke

$$\mu_{mix} = \sum_{i=1}^{N} \frac{\mu_i X_i}{X_i + \sum\limits_{\substack{j=1 \\ j\neq i}}^{N} X_j \left(1. + \sqrt{\frac{\sqrt{m_j}\,\mu_i}{\sqrt{m_i}\,\mu_j}}\right)^2 \sqrt{\frac{m_j}{8(m_j + m_i)}}}$$ (Phase.13)

Preprocessor > Phases > Phase #i > Thermal conductivity of mixture > Mass weighting

$$\lambda = \sum_{i=1}^{N} Y_i \lambda_i$$ (Phase.14)

Preprocessor > Phases > Phase #i > Thermal conductivity of mixture > Molar weighting

$$\lambda = \sum_{i=1}^{N} X_i \lambda_i$$ (Phase.15)

Preprocessor > Phases > Phase #i > Thermal conductivity of mixture > Mason-Saxena

$$\lambda_{mix} = \sum_{i=1}^{N} \frac{\lambda_i X_i}{X_i + 1.065 \sum_{\substack{j=1 \\ j \neq i}}^{N} X_j \left(1. + \sqrt{\frac{\sqrt{m_i}\lambda_i}{\sqrt{m_j}\lambda_j}} \right)^2 \sqrt{\frac{m_j}{8(m_j + m_i)}}} \qquad \text{(Phase.16)}$$

Notations in formulae (Phase.1) - (Phase.16)

In expressions (Phase.1) - (Phase.16)

N - the number of **Substances** in the **Phase** (the number of mixture components),

X_i - molar fraction of **Substance i** (= volume fraction in the case of gases),

Y_i - mass fraction of **Substance i**.

Relation between molar and mass fractions

The relationship between the molar and mass fractions of **Substance i** is as follows:

$$Y_i = X_i \frac{m_i}{m} \qquad \text{(Phase.17)}$$

10.4 Modifiers

Modifiers are the elements which allow changing the solution and the computational domain. A *Modifier* is created on an **Object** and has effect in all the cells entirely or partly entering the **Object**. This rule does not work only for *Modifier* **Moving body**.

This section describes the *Modifiers* common for all the **Physical processes**, viz:

- **Setting variable**
- **Moving body**

Modifiers, specific for various **Physical processes**, are:

- **Volume force**
- **Resistance**
- **Volume heat source**
- **Ignition/Extinction**

are described in the corresponding sections below.

10.4.1 Setting variable

Modifier **Setting variable** sets the specified value for a pointed out variable in all the cells entirely or partly entering the **Object** on which the **Modifier** was created.

10.4.2 Moving body

Modifier **Moving body** transforms the **Object** on which it was created into an element of the geometry model and allows specifying boundary conditions at its surface. Physical processes can be simulated both outside and inside a **Moving body**.

In *FlowVision*, a **Moving body** is a part of the geometry which moves inside the computational domains relative to its boundaries. In the interface,it is represented by two elements: **Imported object** and Modifier **Moving body**. An **Imported object** is the geometry model of an object imported from an external CAD system. The modifier supplies the **Imported object** with additional parameters:

- boundary conditions on the surface of the object
- controls of the object translation and rotation
- forces and moments exerted onto the object

The motion of the **Moving body** relative to the computational grid causes multiple rebuilding the grid. Sub-grid resolution of the body surface takes into account the peculiarities of the body shape. The parameters controlling the body translation and rotation may depend on time and other variables. The variables are specified as constants, tabulated functions or formulas. Computing the current values of these parameters is performed after updating Modifier **Moving body.** This updating includes:

- calculation of the new position of the **Moving body**
- calculation of new forces and moments
- rebuilding the computational grid around the body

If a project contains several **Moving bodies** and updating is toggled on only for one of them, all the bodies are updated. Certain problems are solved with immobile **Moving bodies**. It is recommended to toggle off updating the **Moving bodies** in such a problem, since grid rebuilding takes essential time.

In *FlowVision*, the forces and the moments of the forces exerted onto bodies and surfaces are computed in the absolute coordinate system (CS). The moment of inertia and the rotation center of a **Moving body** is specified in the local CS of the **Object** on which this **Moving body** was created (i.e. in the local CS of the given **Imported object**). Conversion of these parameters from the local CS to the absolute one is performed automatically. The law of motion of the **Moving body** along each coordinate axis (**X**, **Y**, **Z**) is specified separately. It is defined for three time intervals:

- at the first interval, the body stays in rest,
- at the second interval, the body moves with the linear or angular velocity specified in the interface,
- at the third interval, the body motion is determined by the action of forces $\boldsymbol{F}_{\text{ext}} + \boldsymbol{F}_{\text{hydr}}$ (for translation) and moments of forces $\boldsymbol{T}_{\text{ext}} + \boldsymbol{T}_{\text{hydr}}$ (for rotation) as well as by the damping coefficient (O_V for translation, O_ω for rotation).

See details in section Modifier «Moving body».

10.5 Physical processes

The mathematical models of physical processes implemented in the FlowVision software suite are submitted in the given section. Hereafter they are referred to as *mathematical models*. The models are specified and tuned in the properties windows of different elements of the **Preprocessor** and **Solver** trees. Hereafter the properties window of a tree element is referred as *window*, for instance, window **Preprocessor > Phases > Phase #i > Physical processes > Heat transfer** or window **Solver > Advanced settings**.

The physical processes proceeding in a continuous **Phase** are activated in properties of the element **Preprocessor > Phases > Phase #i** (continuous) **> Physical processes**. These are:

- **Motion**
- **Heat transfer**
- **Mass transfer**
- **Turbulence**
- **Phase transfer**
- **Radiation**

The physical processes proceeding in a dispersed **Phase** of the **Particles** type are activated in properties of the element **Preprocessor > Phases > Phase #i** (dispersed of the **Particles** type) **> Physical processes**. These are:

- **Heat transfer**
- **Phase transfer**
- **Motion**
- **Mass transfer**

The physical processes proceeding in a dispersed **Phase** of the **Carcass** type are activated in properties of the element **Preprocessor > Phases > Phase #i** (dispersed of the **Carcass** type) **> Physical processes**. These are:

- **Heat transfer**
- **Mass transfer**

Selection **(none)** means that the given process is disabled.

The user can specify the **Use Gap model** parameter in the properties of **Preprocessor > Models > Model #i** as follows:

- **(none)**
- **Standard gap model**

Selection **(none)** implies that the governing equations are solved in thin clearances (if any) similarly to the other regions of the computational domain. The gap models modify the mathematical models of the **Physical processes** in thin clearances in such a way that the viscous forces and diffusive fluxes are computed from analytical expressions. This technique allows user to avoid resolving thin clearances between solid surfaces by computational grid.

If a continuous and a dispersed **Phases** are presented in the folder **Preprocessor > Phases**, it is necessary to specify parameters of interaction between the two **Phases** in properties of the element **Models > Model #i > Phase interaction > Continuous-particles | Continuous-carcass**. These models take into account the exchanges by mass (ablation/condensation), momentum (force interaction), energy (heat exchange) between the given dispersed and the continuous (carrier) media.

10.5.1 Motion

The following selection is present in position **Motion** in properties of the element **Preprocessor > Phases > Phase #i > Physical processes**:

- **(none)**
- **Navier-Stokes model**
- **Darcy model**

Below the flow models based on the Navier-Stokes equations and the Darcy equations are described.

This section has individual numeration of equations, references, and illustrations.

10.5.1.1 Notations

Notation	Physical quantity	Name in *FlowVision*	Dimension
$B = \frac{A_w}{m}$	ratio of transferring the momentum from the fluid to a deformable or mobile body	**Mobility**	m kg^{-1}

Notation	Physical quantity	Name in *FlowVision*	Dimension
$C = \frac{dl}{dP}$	material load response	**Flexibility**	m Pa^{-1}
C	inertia coefficient		m^{-1}
D	isotropic resistance coefficient	**Resistance coef.**	kg m^{-3} s^{-1}
$D = \frac{\delta}{\alpha}$		**Coef. D**	m^{-1}
$E = \delta \cdot C$		**Coef. E**	
$\boldsymbol{F}_{user}$	user-defined volume force	**Volume force**	N m^{-3}
L_g	thickness of hydrostatic layer	**g-Thickness**	m
$M = \frac{\lvert \boldsymbol{V} \rvert}{c}$	Mach number	**MachNumber**	
n	exponent in the power law (for non-Newtonian fluid)	**Power**	
P_{hst}	hydrostatic pressure		Pa
P_{+hst}	relative pressure + hydrostatic pressure	**Pressure (+hydrostatic)**	Pa
R_1, R_2	Riemann invariants		m s^{-1}
$\hat{\boldsymbol{S}}$	strain rate tensor		s^{-1}
T_0	absolute activation temperature (for non-Newtonian fluid)	**T activation**	K
u_τ	dynamic velocity		m s^{-1}
$\boldsymbol{V}$	relative velocity (vector)	**Velocity**	m s^{-1}
$\boldsymbol{V}_{abs}$	absolute velocity (vector)	**Velocity**	m s^{-1}
$\lvert \boldsymbol{V} \rvert$	modulus of relative velocity		m s^{-1}
$\lvert \boldsymbol{V}_{abs} \rvert$	modulus of absolute velocity		m s^{-1}
y	distance to nearest wall	**Distance to wall**	m
k_d	permeability		m^2
δ	thickness of porous boundary		m
$\gamma = \frac{C_p}{C_p - R_A/m}$	specific rate ratio for gas mixture		
κ	pre-exponential factor (for non-Newtonian fluid)	**Constant**	depends on n
λ	coefficient of thermal conductivity (molecular)	**Thermal conductivity**	kg m s^{-3} K^{-1}
μ	dynamic coefficient of viscosity (molecular)	**Viscosity**	Pa s = kg m^{-1} s^{-1}

Notation	Physical quantity	Name in *FlowVision*	Dimension
μ_{NN}	dynamic coefficient of viscosity (for non-Newtonian fluid)	**Viscosity**	Pa s = kg m^{-1} s^{-1}
μ_t	dynamic coefficient of viscosity (turbulent)	**TurbViscosity**	Pa s = kg m^{-1} s^{-1}
$\nu = \frac{\mu}{\rho}$	kinematic coefficient of viscosity (molecular)		m^2 s^{-1}
$\nu_t = \frac{\mu_t}{\rho}$	kinematic coefficient of viscosity (turbulent)		m^2 s^{-1}
ρ	density	**Density**	kg m^{-3}
ρ_g	hydrostatic density	**g-Density**	
σ	surface tension coefficient	**Surface tension**	N m^{-1}
$\hat{\tau}$	shear stress tensor		Pa
τ_w	specific viscous force exerted by fluid onto solid surface	**Modulus of shear stress**	Pa

Indices:

- b - boundary value
- G - Gas (aggregative state)
- L - Liquid (aggregative state)
- w - value at wall
- n - normal (to cell face) component of vector
- τ - tangential (to cell face) component of vector
- ∞ - free stream value

The other notations see in section **Basic notations**.

10.5.1.2 Parameters

Parameters in window **Preprocessor > Phases > Phase #i > Physical processes > Motion**:

Parameter	Permissible values	Description
Math. model	• **Navier-Stokes model** • **Darcy model**	A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.)
Time step coefficient	An arbitrary numerical value determined by the problem (the default value is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different time steps for different processes sometimes accelerates convergence to the steady-state solution. Meaning of different values:	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the whole project (τ). *Thus:* Own time step of a physical process = τ × **Time step coefficient** Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time.

Parameter	Permissible values	Description
	1 = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the problem as a whole). **> 1** = The calculations of the given process are accelerated. **(0, 1)** = The calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations / initialization).	
Visc. force supplement	**Yes** | **No**	When **Visc. force supplement = Yes**, the full (Motion.2) is solved: the part of viscous force resulting from variation of viscosity and density in space is toggled on in a cell.

10.5.1.3 Equations

The continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \boldsymbol{V}) = 0 \qquad \text{(Motion.1)}$$

Navier-Stokes model

The momentum equation:

$$\frac{\partial \rho \boldsymbol{V}}{\partial t} + \nabla(\rho \boldsymbol{V} \otimes \boldsymbol{V}) = -\nabla p + \nabla \cdot \hat{\boldsymbol{\tau}}_{eff} - D\boldsymbol{V} - \hat{\boldsymbol{D}}\boldsymbol{V} + \boldsymbol{F} + \boldsymbol{F}_L + \boldsymbol{F}_{user} \qquad \text{(Motion.2)}$$

$$\hat{\boldsymbol{\tau}}_{eff} = (\mu + \mu_t)\left(2\hat{\boldsymbol{S}} - \frac{2}{3}(\nabla \cdot \boldsymbol{V})\hat{\boldsymbol{I}}\right) \qquad \text{(Motion.3)}$$

$$\hat{\boldsymbol{S}}_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right) \qquad \text{(Motion.4)}$$

$$D = \left(\frac{\mu}{k_d} + C\frac{1}{2}\rho|\boldsymbol{V}|\right)\varphi_c \qquad \text{(Motion.5)}$$

$$\boldsymbol{F} = (\rho - \rho_g)\boldsymbol{g} - \rho(2\boldsymbol{\omega} \times \boldsymbol{V} + \boldsymbol{\omega} \times \boldsymbol{\omega} \times \boldsymbol{r}) \qquad \text{(Motion.6)}$$

Here

D is the coefficient of isotropic resistance

$\hat{\boldsymbol{D}}$ is the matrix coefficient of anisotropic resistance (see section Modifier "Anisotropic resistance")

k_d is permeability of carcass (solid immovable matrix)

φ_c is relative volume of cell occupied by fluid (void volume)

$\boldsymbol{F}_L$ is the Lorentz force exerted by electromagnetic field onto a unit volume of the medium,

$\boldsymbol{F}_{user}$ is the user volume force,

$\hat{\boldsymbol{I}}$ is the unit tensor,

$\boldsymbol{\omega}$ is the angular velocity of the rotating **Subregion**,

$\boldsymbol{r}$ is the radius-vector with origin at the rotation axis.

The system of Eqs. (Motion.1), (Motion.2) is usually called the Navier-Stokes equations. If non-zero coefficient D is specified in the interface, Eq. (Motion.2) simulates the flow of liquid or gas in isotropic porous medium. The implemented numerical algorithm does not depend on the local Mach number. It allows calculations of sub-, trans, super-, and hypersonic flows.

Model for interacting fluid and a deformable/moving body

When interaction between fluid and a deformable or moving body is simulated, the acceleration of the moving body's surface due to change of the pressure at a given time step is taken into account:

$$\hat{V}_w^n = V_w^n + \frac{\partial P}{\partial t} K \qquad \text{(Motion.7)}$$

where

V_w^n is the local velocity of the body's surface on the current time step calculated without taking the deformation into account (as if the body be solid)

$\hat{V}_w^n$ is the local velocity of the body's surface calculated with taking the deformation into account in the process of integrating the momentum equation. In *FlowVision* the momentum equation is solved after the equation for pressure (the continuity equations), so the pressure P_{n+1} is known and the pressure's time derivative is calculated explicitly:

$$\frac{\partial P}{\partial t} = \frac{P^{n+1} - P^n}{\tau}$$

To simulate interacting a solid structure and fluid, a separated two-part method has been developed (the problem is solved by two solvers, the *FlowVision*'s solver and an external FEA software's solver).

The current *FlowVision* version uses an explicit splitting scheme (only one iteration of *FlowVision* and one iteration of external code are done at each time step). This approach can cause non-conformity of forces acting on the body and the body's displacement at the end of the time step. For example, *FlowVision* initially calculates the time step and transfers the loads to the body, and only after this the FEA software calculates displacements of the body. This can cause unsynchronization of forces and displacements. This effect is especially noticeable when density of the body is much less then density of the fluid acting on the body (the so called decoupling of FSI equations occurs in this situation).

To improve stability of the computation, the K factor is used, the **Artificial compressibility**:

$$K = (C + \Delta t^2 B) \qquad \text{(Motion.8)}$$

Values C and B here determine **Flexibility** and **Mobility** of the body respectively, they are set in the program's user interface by parameters **FSI > Flexibility** and **FSI > Mobility** in properties of the **Moving body**.

The **Artificial compressibility** provides smoothing of the pressure on the wall and reduces the effect of unsynchronization. But use of **Artificial compressibility** causes «leakage» of the fluid through the surface, as it is seen from formula (Motion.7) that velocity of the body's surface doesn't correspond to the real velocity at the current iteration when the K is large. This, in its turn, can violate the mass conservation. So, you should take into

account defects in calculating the mass change when running conjugated simulations with large **Artificial compressibility**.

Here

$C = \frac{dl}{dP}$ is **Flexibility** (deformability) of the body under applied loads. This parameter is recommended for use when the time step changes slightly during the computation. In this situation the initially set value of this parameter will work well during the whole computation.

dl is displacement of the body's surface.

dP is the pressure increment at the given time step

$B = \frac{A_w}{m}$ is **Mobility**, which is characterizes the momentum transfer from the fluid to the **Moving body**. **Mobility** characterizes acceleration of the wall under the applied pressure and depends heavily on changing the time step during the computation. This parameter is recommended for use when the time step changes significantly (tenfold or more).

A_w is area of the body's surface that interact with the fluid.

m is the body's mass that is put in motion by the force $dP \cdot A_w$ during the time step Δt.

Darcy model

In the Darcy model, differential momentum equation (Motion.2) is not solved. Instead, an algebraic relationship between the fluid velocity and pressure gradient is assumed. Below examples of such a relationship are given:

$$\varphi_c \boldsymbol{V} = -\frac{k_d}{\mu_c}\nabla p + \frac{k_d}{\mu_c}\left(\rho_c - \rho_g\right)\boldsymbol{g} \quad \text{(Motion.9)}$$

$$-\nabla p + \left(\rho_c - \rho_g\right)\boldsymbol{g} = D\boldsymbol{V} \quad \text{(Motion.10)}$$

$$\varphi_c \boldsymbol{V} = -\frac{\hat{\boldsymbol{K}}_d}{\mu_c}\nabla p + \frac{\hat{\boldsymbol{K}}_d}{\mu_c}\left(\rho_c - \rho_g\right)\boldsymbol{g} \quad \text{(Motion.11)}$$

$$-\nabla p + \left(\rho_c - \rho_g\right)\boldsymbol{g} = \hat{\boldsymbol{D}}\boldsymbol{V} \quad \text{(Motion.12)}$$

Here

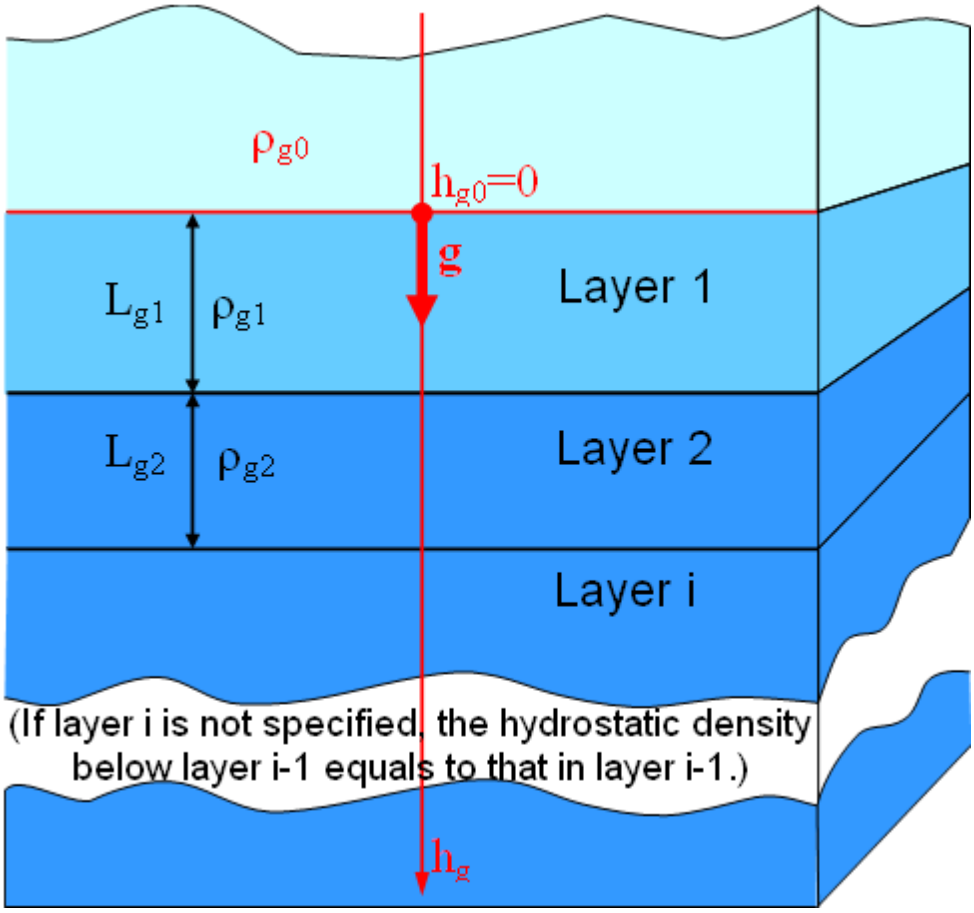
φ_c is relative volume of cell occupied by fluid (void volume)

k_d is isotropic permeability of carcass (of solid immovable porous body)

$\hat{\boldsymbol{K}}_d$ is anisotropic permeability of carcass

10.5.1.4 Hydrostatics

Hydrostatic pressure P_{hst} is the pressure produced by a quiescent layer L_g of incompressible fluid with density ρ_g. Hydrostatic density ρ_g is the density which is used in calculation of hydrostatic pressure. Zero hydrostatic level is a level h_{g0}=0 at which hydrostatic pressure is assumed equal to zero. Hydrostatic layer is a layer L_g of fluid with constant hydrostatic density. Hydrostatic layers reside below zero level.



The location of zero hydrostatic level is defined by the location of **g-Point** (in the coordinate systems of the computational domain). Direction of the gravity force determines counting of hydrostatic layers. Hydrostatic pressure is computed as follows:

$$P_{hst}=\begin{cases} -\rho_{g0} g h_g & h_g<0 \\ 0 & h_g=0 \\ \rho_{g1} g h_g & 0<h_g<L_{g1} \\ \rho_{g1} g L_{g1}+\rho_{g2} g\left(h_g-L_{g1}\right) & L_{g1}<h_g<L_{g1}+L_{g2} \\ \ldots & \end{cases}$$

Here ρ_{g0} is the fluid density above zero hydrostatic level (h_g<0). If hydrostatic layers are not specified, this density determines hydrostatic pressure below zero level (h_g>0). If hydrostatic layers are specified, their densities ρ_{g1}, ρ_{g2}, ... must be also specified. These densities determine the hydrostatic pressure below zero level.

Coordinates of **g-Point**, the fluid density above zero hydrostatic level and hydrostatic layers are specified in window **Preprocessor >** General settings.

10.5.1.5 Boundary conditions

The current section describes boundary conditions for variables **Velocity** and **Pressure**, corresponding to different boundary templates. Conditions for these variables are selected in window **Subregions > Subregion #i > Boundary conditions > Boundary condition** (template) **> Variables > Velocity**.

A boundary can move. A boundary is regarded as moving one if

- this is the surface of **Moving body**,
- **Rotation** is specified in folder **Boundary condition** (template) **> Local CS**,
- **Translation** is specified in folder **Boundary condition** (template) **> Local CS**.

In general case, the boundary velocity in the absolute coordinate system is computed as follows:

$$\boldsymbol{V}_B=\boldsymbol{V}_{MB}+\boldsymbol{\omega}\times\boldsymbol{r}+\boldsymbol{V}_{tr}-\boldsymbol{n}\left(\boldsymbol{V}_{tr}\cdot\boldsymbol{n}\right) \qquad \text{(Motion-BC.1)}$$

Here

$\boldsymbol{V}_{MB}$ - velocity of **Moving body**,

$\boldsymbol{\omega}$ - angular velocity of **Rotation** specified in **Local CS**,

$\boldsymbol{r}$ - radius-vector of the center of a given cell face,

$\boldsymbol{V}_{tr}$ - velocity of **Translation** specified in **Local CS**.

The origin of radius-vector $\boldsymbol{r}$ is located at the axis of rotation in the point specified in the corresponding **Local CS**.

If **Translation** is specified, only the tangential component of vector $\boldsymbol{V}_{tr}$ enters the expression for the absolute boundary velocity. In all the boundary templates, velocity and its direction are specified in the coordinate system related to the moving boundary.

Therefore,

$$\boldsymbol{V}_{b,abs} = \boldsymbol{V}_b + \boldsymbol{V}_B$$

$$\boldsymbol{V}_b = \boldsymbol{V}_{user}$$

Here

$\boldsymbol{V}_{b,abs}$ - local velocity of the boundary in the absolute coordinate system,

$\boldsymbol{V}_b$ - local velocity of the boundary in the coordinate system, related to the boundary,

$\boldsymbol{V}_{user}$ - fluid velocity specified by user.

This section has individual numeration of equations.

10.5.1.5.1 Template 'Wall'

No slip

This boundary condition is defined by selection

- **Phase #i > Physical processes > Turbulence** = **(none)**

or

- **Wall > Wall interaction > Phase #i** = **No wall functions**.

User specifies nothing. The following condition is automatically set on the given boundary:

$$\boldsymbol{V}_b = 0 \qquad \text{(Motion-BC.2)}$$

Slip

In the template discussed, selection of the this boundary condition is possible only when **Wall > Wall interaction > Phase #i** = **No wall functions**.

User specifies nothing. The following condition is automatically set on the given boundary:

$$V_{b,n} = 0 \qquad \text{(Motion-BC.3)}$$

$$V_{b,\tau} = \left|\boldsymbol{V}_{c,abs} - \boldsymbol{V}_B\right| \qquad \text{(Motion-BC.4)}$$

Here

$\boldsymbol{V}_{c,abs}$ - absolute velocity at the center of a cell adjacent to the boundary.

Thus, the fluid velocity is tangent to the boundary (the normal component is set zero). The modulus of the relative velocity at the boundary equals to that in the cell center.

Logarithmic law

The boundary condition **Logarithmic law** is defined by selection in the properties of the boundary condition **Wall**:

- **Wall > Wall interaction > Phase #i** = **Wall functions, equilibrium** or **= Wall functions, nonequilibrium**

- *and* **Variables > Velocity = Logarithm law**.

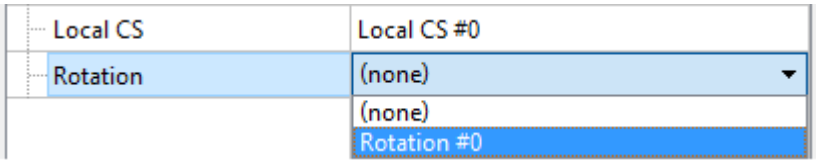
User specifies nothing. The condition assumes that a pre-defined profile of relative velocity develops near the given wall - see section **Models of wall functions**.

Track

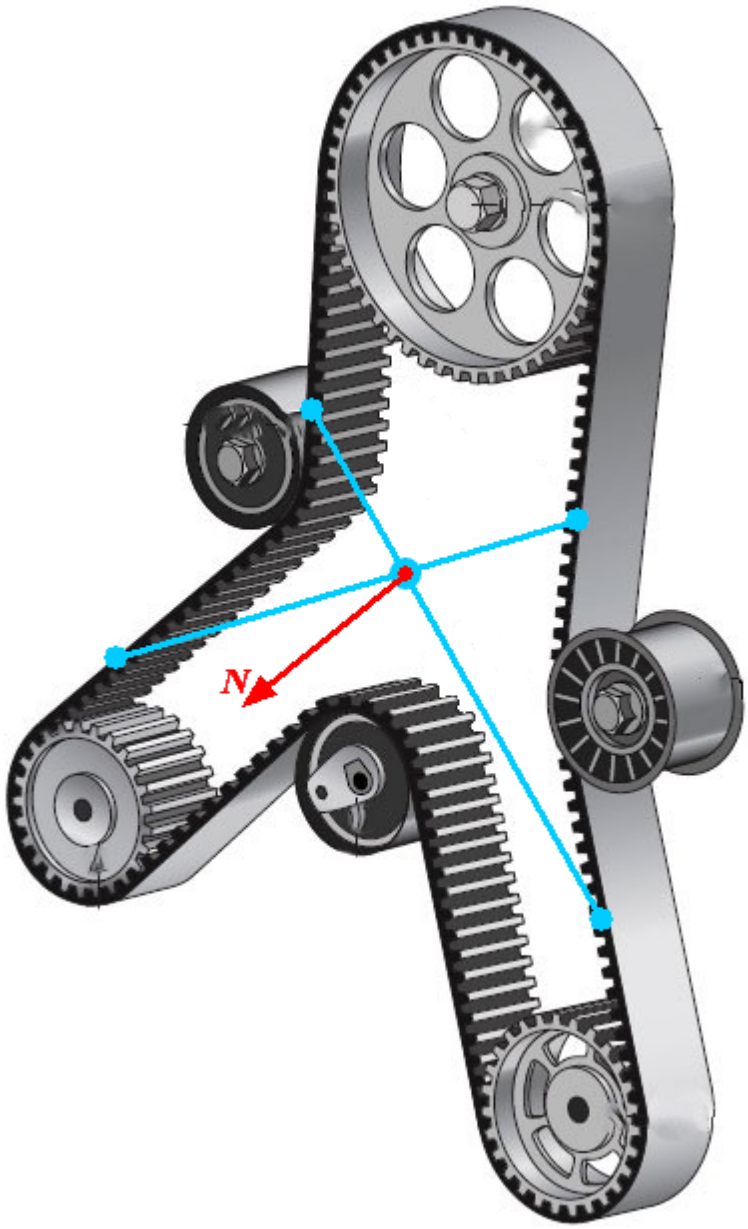
The boundary condition **Track** allows you to simulate a moving band like a track or closed-loop band belt on pulleys. It is assumed that trajectory of any point on this surface lie in a plane. To use this boundary condition, you have to specify in the properties of the boundary condition **Wall**: **Variables > Velocity = Track**.

Direction of the **Track**'s motion is set by the **Rotation** vector, which is to be orthogonal to the plane, in which any selected point of the **Track** moves. You have start with specifying direction of this vector by creating a **Local coordinate systems > Local CS #N > Rotation > Rotation #N** element in the project tree and then specifying its parameters **Direction > ...**. Origin of the **Rotation** vector (it is specified by parameters **Center > ...**) is to be located in the interior of the **Track**'s contour.

Then in properties of the boundary condition **Wall** you have to specify **Local CS = Local CS #N** and **Rotation = Rotation #N**:



Absolute value of the **Rotation** vector (which is set in the properties of **Rotation #N** by the parameter **Speed**) is not used and can be even zero.



The boundary condition **Track** is used to simulate a notched belt.
Vector of **Rotation** N sets direction of the **Track** and it is to be orthogonal to the plane, in which the edge of the belt lies.

The linear velocity, [m/s], of the **Track**'s surface is set by the **Track speed** parameter in the properties of the child element **B. Cond. #N > Velocity**.

The **Friction coefficient** in the properties of the child element **B. Cond. #N > Velocity** specifies the non-dimensional friction coefficient C_f, which is used for calculation of the friction force acting on an elementary area of the **Track**'s surface.

When C_f >= 0, the friction force F acting on an elementary area of the **Track**'s surface is calculated by the formula:

$$F = C_f s \rho (V\text{-}V_t)^2/2$$

where:

ρ is density of the fluid,

V is velocity of the fluid near the **Track**,

V_t is **Track speed**, which is velocity of the **Track**,

s is size of the elementary area, on which the friction force acts.

When C_f < 0, the friction force is calculated in the same way as the friction force for a moving wall.

10.5.1.5.2 Template 'Symmetry'

Slip

User specifies nothing, see previous section Template 'Wall'.

10.5.1.5.3 Template 'Inlet/Outlet'

Normal mass velocity

This boundary condition is specified at subsonic inlet and subsonic outlet. The user specifies a value for the normal component of mass velocity:

$$\rho V_n\big|_b = \rho V_n\big|_{user} \qquad \text{(Motion-BC.5)}$$

Positive value means inlet. Negative value means outlet (forced suction). In the case of moving boundary, the fluid velocity, specified in such a way, is regarded as relative one.

The absolute velocity $\boldsymbol{V}_{b,abs}$ is calculated in the program as the sum of the relative fluid velocity $\boldsymbol{V}_b$ and the local boundary velocity $\boldsymbol{V}_B$ – see Eq. (Motion-BC.1).

It should be noted that in the case of compressible flow (density depends on pressure) the inlet / outlet velocity will not be fixed. In will depend on the static pressure near the surface to which the given boundary condition is assigned. The static pressure in the center of a cell adjacent to this surface is obtained in the course of solving the subsonic problem.

Normal velocity with pressure

This boundary condition is specified at inlet. User specifies the velocity modulus and static pressure in the free stream (at infinity):

$$V_{n,b} = V_{user} > 0 \qquad \text{(Motion-BC.6)}$$

$$P_b = P_{user} \qquad \text{(Motion-BC.7)}$$

The absolute velocity $\boldsymbol{V}_{b,abs}$ is calculated in the program as the sum of the relative fluid velocity $\boldsymbol{V}_b$ and the local boundary velocity $\boldsymbol{V}_B$ – see Eq. (Motion-BC.1).

Negative velocity value (forced suction) is forbidden. However, this condition allows outflow of the fluid through the inlet if an appropriate situation appears during simulating the flow. In this case, it works like condition **Total pressure**. The following cases are implemented in the program:

Inlet, supersonic flow:

The boundary values of the velocity modulus and static pressure are equal to the specified free stream values. The velocity is directed inside the computational domain along the internal normal to the given boundary. Static temperature is specified for process **Heat transfer**.

Inlet, subsonic flow:

The velocity modulus at the given boundary equals to that in the center of the corresponding near-boundary cell. The velocity vector is directed inside the computational domain along the internal normal to the given boundary. The velocity in the cell center results from solving the momentum equation in the given cell. The static pressure is computed from the total pressure determined by the specified values of static pressure, velocity and static temperature – see Eqs. (Motion-BC.10), (Motion-BC.11) below.

Outlet, supersonic flow:

The velocity vector at the given boundary equals to that in the center of the corresponding near-boundary cell. The static pressure at the boundary equals to the static pressure in the cell's center.

Outlet, subsonic flow:

The velocity vector at the given boundary equals to that in the center of the corresponding near-boundary cell. The static pressure at the boundary equals to the total pressure defined by values P_{user}, V_{user}, T_{user}.

Velocity with pressure

This boundary condition is used to specify inlet of the fluid. If during the simulation a vortex comes to the BC and circumstances for outflow through the BC will appear, then this BC will also work correctly (see descriptions of situations *"Outlet, supersonic flow"* and *"Outlet, subsonic flow"* below).

This boundary condition is specified at inlet. User specifies the velocity vector:

$$\boldsymbol{V}_b = \boldsymbol{V}_{user}, \quad V_n\big|_b = \boldsymbol{V}_b \cdot \boldsymbol{n} > 0 \qquad \text{(Motion-BC.8)}$$

and static pressure (Motion-BC.7) in the free stream (at infinity).

The absolute velocity $\boldsymbol{V}_{b,abs}$ is calculated in the program as the sum of the relative fluid velocity $\boldsymbol{V}_b$ and the local boundary velocity $\boldsymbol{V}_B$ – see Eq. (Motion-BC.1).

Negative value of the velocity projection on the internal normal to the given boundary (forced suction) is forbidden. However, the condition allows outflow of the fluid through the inlet if an appropriate situation appears during simulating the flow. In this case, it works like condition **Total pressure**.

The following cases are implemented in the program:

Inlet, supersonic flow:

The boundary values of the velocity components and static pressure are equal to the free stream values. Static temperature is specified for process **Heat transfer**.

Inlet, subsonic flow:

The velocity modulus at the given boundary equals to that in the center of the corresponding near-boundary cell. The direction of the velocity vector is determined by vector specified in the interface. The velocity in the cell center results from solving the momentum equation in the given cell. The static pressure is computed from the total pressure determined by the specified values of static pressure, velocity and static temperature – see Eqs. (Motion-BC.10), (Motion-BC.11) below.

Outlet, supersonic flow:

The velocity vector at the given boundary equals to that in the center of the corresponding near-boundary cell. The static pressure at the boundary equals to the static pressure in the cell center.

Outlet, subsonic flow:

The velocity vector at the given boundary equals to that in the center of the corresponding near-boundary cell. The static pressure at the boundary equals to the total pressure defined by values P_{user}, V_{user}, T_{user}.

Inlet pressure

The user specifies the static pressure (Motion-BC.7). The velocity vector at the given boundary equals to that in the center of the corresponding near-boundary cell. The cell-center velocity results from integration of the momentum equation. The given boundary condition (BC) can be specified at subsonic inlet and subsonic outlet.

The following cases are implemented in the program:

- ***Outlet, supersonic flow*:** The static pressure at the boundary equals to that in the cell center. The pressure value, specified in the interface, is ignored.
- ***Outlet, supersonic flow*:** The static pressure at the boundary equals to the value specified in the interface.
- ***Inlet, subsonic flow:*** The static pressure at the boundary equals to the value specified in the interface.

If only one inlet and one outlet are presented in a *FlowVision* project, then specifying BC **Inlet pressure** both at the inlet and at the outlet is incorrect from mathematical point of view. In this case it will be also incorrect to specify BC **Inlet pressure** at the inlet and pressure at the outlet.

An example of correct combination is **Inlet pressure** (at inlet) + **Normal mass velocity** (at outlet). In this problem setting the normal mass velocity at outlet is to be negative (outflow).

Total pressure

This boundary condition is specified at:

- subsonic inlet
- supersonic outlet
- subsonic outlet

The user specifies a value for the total pressure:

$$P_{tot,b} = P_{user} \qquad \text{(Motion-BC.9)}$$

Inlet, subsonic flow:

The modulus of the absolute velocity at the boundary equals to the modulus of the absolute velocity in the center of a cell adjacent to the boundary. The fluid velocity in the cell center results from solving the momentum equation in the given cell. The vector of the relative velocity is directed along the internal normal to the boundary. The static pressure at the boundary is computed from the following relationships:

$$P_{tot} = P + \rho \frac{\boldsymbol{V}_{b,abs}^2}{2} \quad \text{if } \textbf{Substance > Aggregative state = Liquid} \qquad \text{(Motion-BC.10)}$$

$$P_{tot} = P\left(\frac{T_{tot}}{T}\right)^{\frac{\gamma}{\gamma-1}} \quad \text{if } \textbf{Substance > Aggregative state = Gas} \qquad \text{(Motion-BC.11)}$$

The total and static temperature are related by:

$$H(T_{tot}) = h(T) + \frac{\boldsymbol{V}_{b,abs}^2}{2} \qquad \text{(Motion-BC.12)}$$

Outlet, supersonic flow:

The vector of the absolute velocity at the boundary equals to the vector of the absolute velocity in the center of a cell adjacent to the boundary. The fluid velocity in the cell center results from solving the momentum equation in the given cell. The static pressure at the boundary equals to the pressure in the cell center. The interface value is ignored.

Outlet, subsonic flow:

The vector of the absolute velocity at the boundary equals to the vector of the absolute velocity in the center of a cell adjacent to the boundary. The fluid velocity in the cell center results from solving the momentum equation in the given cell. The static pressure at the boundary equals to the total pressure specified in the interface.

Total pressure and velocity direction

This boundary condition is similar to the condition **Total pressure**. As **Total pressure**, it limits from above the modulus of the boundary velocity by the local sonic speed. The only difference is that the velocity vector of the inflowing fluid is directed along the vector specified in the interface (not along the local normal to the boundary). The specified vector **Velocity direction** is not necessarily a unit one. If it is zero, the velocity direction is determined by the internal normal to the boundary (exactly like in boundary condition **Total pressure**). Otherwise, the vector is automatically normalized in the program.

Fixed velocity

This boundary condition can be specified at subsonic inlet and subsonic outlet. The user specifies the velocity vector, which doesn't change during the computation.

$$\boldsymbol{V}_b = \boldsymbol{V}_{user}$$

When $\boldsymbol{V}_b \cdot \boldsymbol{n} > 0$, this means that the fluid inflows through the given boundary.

When $V_b \cdot n < 0$, this means that the fluid outflows through the boundary (the normal is directed inside the computational domain).

The **Fixed velocity** condition is recommended for solving problems with known distribution of velocity (usually at the inlet). Notice that in simulating compressible flows (when density depends on pressure) with use of the given condition, the mass flow rate will not be constant. It will depend on the static pressure obtained near the surface in the course of solving a subsonic problem. In the case of moving boundary, the fluid velocity, specified in such a way, is regarded as relative one.

The absolute velocity $V_{b,abs}$ is calculated in the program as the sum of the relative fluid velocity V_b and the local boundary velocity V_B – see Eq. (Motion-BC.1).

Supersonic inlet

You can specify **Supersonic inlet** with the following combinations of physical values:

Temperature*)	Static		Total	
Pressure**)	Static		Total	
Velocity**)	Absolute value (modulus)	Mach number	Vector	Mach number and direction of the velocity

*) is set in properties of the element **B.Cond. #N > Temperature**.

) is set in properties of the element **B.Cond. #N > Velocity, see the table below.

Parameter	Description
P (BC) type	Method of specifying the pressure. Possible options are: **Static pressure | Total pressure**.
P static or total	Value of either static or total pressure.
V (BC) type	Method of specifying the velocity. Possible options are: • **Velocity modulus** – specifying the absolute value of the velocity. The velocity vector is directed along the normal to the surface of the boundary condition. • **Mach number** – specifying the velocity by the Mach number. The velocity vector is directed along the normal to the surface of the boundary condition. • **Velocity vector** – specifying the velocity vector by its components • **Mach number + direction** – specifying the velocity by the Mach number and direction of the velocity vector
|V| or M	Here you specify either absolute value (modulus) of the velocity or Mach number. This parameters are available if **V (BC) type = Velocity modulus | Mach number | Mach number + direction**.
V or direction > X	Components of the velocity's vector or directing vector of the velocity. This parameters are available if **V (BC) type = Velocity vector | Mach number + direction**.
V or direction > Y	
V or direction > Z	

Some parameters of physical processes are inappropriate if:

- The **Heat transfer** process is disabled (**Heat transfer = (none)** is set in the properties of the **Phase #N > Physical processes** element); an error message will be displayed: "**Heat transfer is not activated**".
- **Heat transfer = Heat transfer via h** is set in the properties of the **Phase #N > Physical processes** element; an error message will be displayed: "**Supersonic problems must be solved with 'Heat transfer via H'"**.
- **Variables > Temperature = Total temperature** is set in the properties of the boundary condition and during calculating of the static temperature the absolute static temperature becomes equal or less 0; an error message will be displayed: "**Absolute static temperature is negative. Probably, the specified total temperature is too low**".
- absolute value (modulus) of **Velocity**, which has been specified by any method, becomes less then the sonic speed; an error message will be displayed: "**Velocity modulus is less than sonic speed**".

Calculations are done in an iteration process using the following equations:

$$P_{tot} = P\left(\frac{T_{tot}}{T}\right)^{\frac{\gamma}{\gamma-1}} = P\left(\frac{T_{tot}}{T}\right)^{\frac{C_p}{R_A/m}}$$

$$H = h + \left|\frac{\mathbf{V}}{2}\right|^2$$

$$C_p = \sum_1^N C_{p,i}\, Y_i$$

$$h = \sum_1^N h_i\, Y_i$$

$$\frac{1}{m} = \sum_1^N C_{p,i}\, Y_i$$

$$c = \sqrt{\frac{C_p}{C_p - \frac{R_A}{m}}\frac{R_A}{m}T}$$

$$\rho = \frac{P_m}{R_A T}$$

The **Supersonic inlet** condition cannot be used with the Darcy model.

10.5.1.5.4 Template 'Free outlet'

Static pressure

This is set in the program's user interface as **Variables > Velocity = Pressure** in properties of the boundary condition.

The user specifies a value for the static pressure:

$$P_b = P_{user} \qquad \text{(Motion-BC.13)}$$

The velocity vector at the given boundary equals to that in the center of the corresponding near-boundary cell. The cell-center velocity results from integration of the momentum equation. The given boundary condition (BC) can be specified at subsonic inlet and subsonic outlet. The following situations are implemented in the program:

- ***Outlet, supersonic flow***: The static pressure at the boundary equals to that in the cell center. The pressure value, specified in the interface, is ignored.
- ***Outlet, subsonic flow***: The static pressure at the boundary equals to the value specified in the interface.
- ***Inlet, subsonic flow***: The static pressure at the boundary equals to the value specified in the interface.

If only one inlet and one outlet are presented in a *FlowVision* project, then specifying BC **Static pressure** both at the inlet and outlet is incorrect from mathematical point of view. In this case it will be also incorrect to specify **Inlet pressure** at the inlet and **Static pressure** at the outlet. The correct combinations are, for example, the following:

- **Normal mass velocity** (positive) **+ Static pressure**
- **Total pressure + Static pressure**
- **Total pressure and velocity direction + Static pressure**
- **Normal velocity with pressure + Static pressure**
- **Velocity with pressure + Static pressure**

Supersonic outlet

This boundary condition is specified at a supersonic outlet. User specifies nothing. The velocity vector at the given boundary equals to that in the center of the corresponding near-boundary cell. The static pressure at the boundary equals to the value in the cell center.

Pressure at porous surface

This boundary condition, contrary to **Static pressure**, allows specifying hydraulic resistance to the flow through the given boundary. The condition assumes the following relationship between the pressure drop across the boundary and the normal component of velocity:

$$\Delta P\big|_b = -D\mu V_n - E\frac{1}{2}\rho V_n^2 \qquad \text{(Motion-BC.14)}$$

$$D = \delta / \alpha \qquad \text{(Motion-BC.15)}$$

$$E = \delta \cdot C \qquad \text{(Motion-BC.16)}$$

Here

ΔP - pressure drop across the boundary

V_n - projection of velocity onto internal normal to the outlet surface

α - permeability

C - inertia coefficient

δ - virtual thickness of the porous boundary

User specifies:

- static pressure (Motion-BC.13)
- coefficient D
- coefficient E

The **Pressure at porous surface** condition cannot be used with the **Darcy model**.

Fixed velocity

The user specifies the velocity vector, which doesn't change during the computation.

Total pressure

This boundary condition is similar to the boundary condition **Total pressure**, which is used for the template 'Inlet/Outlet'.

The user specifies a value for the total pressure:

$$P_{tot,b} = P_{user}$$

Normal mass velocity

This boundary condition is similar to the boundary condition **Normal mass velocity**, which is used for the template 'Inlet/Outlet'.

The user specifies a value for the normal component of mass velocity:

$$\rho V_n\big|_b = \rho V_n\big|_{user}$$

10.5.1.5.5 Template 'Connected'

After selecting command **Create** in the contextual menu of element **Preprocessor > Boundary links > Binder conditions**, user must select **Connection type** in window **Create binder condition**.

When **Connection type** = **Conjugate temperature** is selected, template **Wall** is automatically set for variable **Velocity** on the given boundary.

Selection **Connection type** = **Conjugate all variables | Periodic surface | Sliding surface**, causes automatic setting of the corresponding condition of matching all the variables on the two surfaces (user specifies nothing).

10.5.1.5.6 Template 'Nonreflecting'

The **Nonreflecting** template has no physical meaning when the **Darcy model** is used.

Riemann

This condition can be set both on inlet and outlet. User specifies x, y, z components of velocity at infinity and static pressure at infinity:

$$\boldsymbol{V}_{\infty} = \boldsymbol{V}_{user} \qquad \text{(Motion-BC.17)}$$

$$P_{\infty} = P_{user} \qquad \text{(Motion-BC.18)}$$

The velocity vector is specified in the absolute coordinate system. It is not recommended to use this condition when the Mach number is getting low (M < 0.1) at the external boundaries.

10.5.1.6 Modifyers

This section has individual numeration of equations.

Volume force

The **Modifier** sets the vector of a volume force in all the cells entirely or partly entering the **Object** on which the **Modifier** was created.

This vector ($\boldsymbol{F}_{user}$) is set by the user and it is included in Eq. (Motion-BC.2).

Resistance

The **Modifier** sets an isotropic resistance in all the cells entirely or partly entering the **Object** on which the **Modifier** was created. User specifies a value for isotropic resistance coefficient:

$$D = D_{user} \qquad \text{(Motion-Mod.1)}$$

This value is used in Eq. (Motion-BC.2). The resistance coefficient can be specified by formula, e. g.,

$$D_{user} = \frac{\mu}{\alpha} + C\frac{1}{2}\rho|\boldsymbol{V}| \qquad \text{(Motion-Mod.2)}$$

Here

μ - molecular viscosity,

α - permeability,

C - inertia coefficient.

10.5.1.7 References

1. Zhluktov S.V., Aksenov A.A., Kharchenko S.A.., Moskalev I.V., Sushko G.B., Shishaeva A.S. (2010) "Simulation of separated flows in software FlowVision-HPC" Computational methods and programming, *Vol. 11, No.2, pp*. 76-87.

10.5.2 Heat transfer

The following selection is present in position **Heat transfer** in window **Preprocessor > Phases > Phase #i > Physical processes**:

- **(none)**
- **Heat transfer via h**
- **Heat transfer via H**

Below the implemented model of convective-diffusive heat transfer is described.

This section has individual numeration of equations.

10.5.2.1 Notations

Notation	Physical quantity	Name in *FlowVision*	Dimension
$\hat{\mathbf{D}}$	matrix, which determines anisotropy of heat transfer	**Rel.Therm.Conductivity coef.**	
C_P	specific heat at constant pressure	**Specific heat**	J kg^{-1} K^{-1} = = m^2 s^{-2} K^{-1}
E_r	density of radiation energy	**Radiation density**	W·m^{-2} = kg s^{-3}
$h(T) = h_0 + \int_{298.15}^{T} Cp(T)dT$	thermodynamic enthalpy	**Enthalpy**	m^2 s^{-2}
h_0	enthalpy of **Substance** formation at 298.15 K	**Formation enthalpy***)	m^2 s^{-2}
$H = h + \frac{\boldsymbol{V}_{abs}^2}{2}$	total enthalpy	**EnthalpyTotal**	m^2 s^{-2}
$\boldsymbol{J}_q$	specific heat flux (vector)	**HeatFlux**	W·m^{-2} = kg s^{-3}
$J_{q,b}$	modulus of specific heat (diffusion) flux from boundary		W·m^{-2} = kg s^{-3}
$J_{rad,b}$	modulus of specific radiation heat flux from boundary		W·m^{-2} = kg s^{-3}
$Pr = \frac{\mu C_p}{\lambda}$	molecular Prandtl number	**Prandtl**	
Pr_t	turbulent Prandtl number	**PrandtlTurb**	
Q_b	volume source of energy in a boundary cell, represented as heat flux from the given boundary	**Flux**	W·m^{-2} = kg s^{-3}
Q_{vis}	volume source of energy due to viscous dissipation		W·m^{-3} = kg m^{-1} s^{-3}
Q_{dif}	volume source of energy due to diffusion of components		W·m^{-3} = kg m^{-1} s^{-3}
Q_{rad}	volume source of energy due to radiation		W·m^{-3} = kg m^{-1} s^{-3}
Q_{user}	user volume source of energy		W·m^{-3} = kg m^{-1} s^{-3}

Notation	Physical quantity	Name in *FlowVision*	Dimension
R	rothalpy	**EnthalpyTotal**	$m^2\ s^{-2}$
T	relative temperature	**Temperature**	K
T_{ext}	relative temperature of the external medium	**T of external medium**	K
$\boldsymbol{V}$	relative velocity in moving coordinate system (vector)	**Velocity**	$m\ s^{-1}$
$\boldsymbol{V}_{abs}$	absolute velocity (vector)	**Velocity**	$m\ s^{-1}$
Y_i	mass fraction of component i	**Mass. fraction_Substance #i**	
y	distance to wall	**Distance to wall**	m
α_{ext}	coefficient of external heat transfer	**Heat-transfer coef.**	$W{\cdot}m^{-2}\ K^{-1}$= $= kg\ s^{-3}\ K^{-1}$
α_w	heat transfer coefficient of a virtual wall in the case when the effective heat transfer connection for connected boundary conditions is used	**Wall heat-transfer coef.**	$W{\cdot}m^{-2}\ K^{-1}$= $= kg\ s^{-3}\ K^{-1}$
λ	coefficient of thermal conductivity (molecular)	**Thermal conductivity**	$kg\ m\ s^{-3}\ K^{-1}$
$\sigma_{rad} = 5.67 \cdot 10^{-8}$	Stefan-Boltzmann constant	**Stefan–Boltzmann constant**	$W{\cdot}m^{-2}{\cdot}K^{-4}$
ε_w	surface emissivity (blackness)	**Blackness**	

*)
If chemical reactions, combustion or phase transitions are not taken into account in the computation, then it is recommended for the used **Substance**, which has been loaded from the **Substance Database**, to set zero **Enthalpy of formation** (h_0). This will improve accuracy of the simulation.

Indices:
b - value on boundary,
cell - value in cell center,
ext - value in the ambient medium,
G - gas,
L - liquid,
user - value specified by user,
w - value at wall.

The other notations see in section **Basic notations**.

10.5.2.2 Parameters

Parameters in properties of the element **Preprocessor > Phases > Phase #i > Physical processes > Heat transfer** (for continuous **Phases**):

Parameter	Possible values	Description
Math. model	• **Heat transfer via h** • **Heat transfer via H**	A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.)
Time step coefficient	An arbitrary numerical value determined by the problem (the default value is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different time steps for different processes sometimes accelerates convergence to the	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the whole project (τ). *Thus:* Own time step of a physical process = τ × **Time step coefficient**

Parameter	Possible values	Description
	steady-state solution. Meaning of different values: **1** = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the problem as a whole). **> 1** = The calculations of the given process are accelerated. **(0, 1)** = The calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations / initialization).	Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time.
All terms	**No** | **Yes**	Account of heat generation by means of viscous dissipation (term Q_{vis})
Use pressure term	**No** | **Yes** The default value is: • **No**: if, at creation of the **Phase**, the **Heat transfer** physical process was specified as **Heat transfer via h** • **Yes**: if, at creation of the **Phase**, the **Heat transfer** physical process was specified as **Heat transfer via H**	Taking into account the pressure term in the energy equation. Possible options are: **No** | **Yes**. When **Use pressure term = No**, the following terms are removed from the the energy equation: • If the **Heat transfer via h** mathematical model is used, terms $\partial P/\partial t$ and $\boldsymbol{V}\cdot\nabla P$ are removed from the equation (HeatTransfer.4). • If the **Heat transfer via H** mathematical model is used, term $\partial P/\partial t$ is removed from the equation (HeatTransfer.9). In most of the cases it is recommended to specify **Use pressure term = Yes** after several time steps after beginning the computation.
D	Determined by the problem. The default value is **0**.	User-defined coefficient in Eq. (HeatTransfer.13).
F	Determined by the problem. The default value is **0**.	User-defined free term in Eq. (HeatTransfer.13).

Parameters of the «Anisotropic thermal conductivity» modifier (if it is applied) that specify anisotropy of molecular heat transfer by a matrix.

Parameter	Description
Rel.Therm.Conductivity coef. 1-1	Element Dxx in the matrix of the anisotropic thermal conductivity.The default value is **1**.
Rel.Therm.Conductivity coef. 2-2	Element Dyy in the matrix of the anisotropic thermal conductivity. The default value is **1**.
Rel.Therm.Conductivity coef. 3-3	Element Dzz in the matrix of the anisotropic thermal conductivity. The default value is **1**.
Rel.Therm.Conductivity coef. 1-2	Element Dxy in the matrix of the anisotropic thermal conductivity. The default value is **0**.

Parameter	Description
Rel.Therm.Conductivity coef. 1-3	Element Dxz in the matrix of the anisotropic thermal conductivity. The default value is **0**.
Rel.Therm.Conductivity coef. 2-3	Element Dyz in the matrix of the anisotropic thermal conductivity. The default value is **0**.

Parameters in properties of the element **Boundary conditions > B. Cond. #N > Temperature (Phase #N)** (presence of these parameters depends on the selected boundary condition template and method of specifying **Temperature**).

Parameter	Possible values	Description
T of external medium	Numerical value determined by the problem. The default value is **0**.	Relative static temperature of the external medium, T_{ext}, [K]
Heat-transfer coef.	Numerical value determined by the problem. The default value is **0**.	Coefficient of external heat transfer of the surface, α_{ext}
Blackness	Numerical value determined by the problem. The default value depends on the selected BC template.	Emissivity (blackness) of the surface, ε_w
T_inf	Numerical value determined by the problem. The default value is **0**.	Absolute temperature at infinity, $T_{\infty,abs}$, [K]

10.5.2.3 Equations

Three energy equations are implemented in the *FlowVision* software:

- equation for total enthalpy (H)
- equation for thermodynamic enthalpy (h)
- simplified equation for thermodynamic enthalpy, in which the convective term is absent.

If the **Motion** process *is not* activated in the given **Phase**, the simplified equation is integrated.

If the **Motion** process *is* activated, a user can select either equation for H or equation for h.

When flows of incompressible liquids and slow flows of gases (for example, in slow gas flows with $M \le 0.1$ with no shock waves) are simulated, it is recommended to use the simpler energy equation, which is formulated via the thermodynamic enthalpy (h).

When flows of compressible fluids are simulated, particularly when solving problems with shock waves, it is recommended to use the energy equation formulated via the total enthalpy (H).

Please note, that shock waves can appear and propagate in liquids after high local pressure surges. To simulate shock waves in liquid, you have to correctly specify the liquid's compressibility.

The simplified energy equation (for a solid body):

$$\frac{\partial(\rho h)}{\partial t} = -\nabla \cdot \boldsymbol{J}_q + Q_{rad} + Q_{user} \qquad \text{(HeatTransfer.1)}$$

$$\boldsymbol{J}_q = -\lambda \nabla T + \sum_{i=species} h_i \boldsymbol{J}_i = -\frac{\lambda}{C_p} \nabla h - \rho D \sum_{i=species} h_i \nabla Y_i \qquad \text{(HeatTransfer.2)}$$

$$Q_{rad} = 4\alpha n^2 \left(E_r - \sigma_{rad} T_{abs}^4 \right) \qquad \text{(HeatTransfer.3)}$$

The energy equation for thermodynamic enthalpy:

$$\frac{\partial(\rho h)}{\partial t} + \nabla(\rho \boldsymbol{V} h) = \frac{\partial P}{\partial t} + \boldsymbol{V} \cdot \nabla P - \nabla \cdot \boldsymbol{J}_q + Q_{vis,L} + Q_{rad} + Q_{user} \qquad \text{(HeatTransfer.4)}$$

$$\boldsymbol{J}_q = -\left(\lambda + \frac{\mu_t C_p}{Pr_t}\right)\nabla T + \sum_{i=species} h_i \boldsymbol{J}_i = -\left(\frac{\lambda}{C_p} + \frac{\mu_t}{Pr_t}\right)\nabla h + \left(\frac{\lambda}{C_p}(1-Le) + \frac{\mu_t}{Pr_t}(1-Le_t)\right)\sum_{i=species} h_i \nabla Y_i \quad \text{(HeatTransfer.5)}$$

$$Q_{vis,L} = \sum_{i,j=1}^{3} \tau_{ij} S_{ij} + \rho\varepsilon \quad \text{(HeatTransfer.6)}$$

$$\tau_{ij} = \mu\left(2S_{ij} - \frac{2}{3}(\nabla \cdot V)\delta_{ij}\right) \quad \text{(HeatTransfer.7)}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right) \quad \text{(HeatTransfer.8)}$$

The energy equation for total enthalpy:

$$\frac{\partial(\rho H)}{\partial t} + \nabla(\rho \boldsymbol{V} H) = \frac{\partial P}{\partial t} + \rho \boldsymbol{V} \cdot \boldsymbol{F} - \nabla \cdot \boldsymbol{J}_q + Q_{vis,G} + Q_{rad} + Q_{user} \quad \text{(HeatTransfer.9)}$$

$$\boldsymbol{J}_q = -\left(\lambda + \frac{\mu_t C_p}{Pr_t}\right)\nabla T + \sum_{i=species} h_i \boldsymbol{J}_i = \\ = -\left(\frac{\lambda}{C_p} + \frac{\mu_t}{Pr_t}\right)(\nabla H - \boldsymbol{V} \cdot (\nabla \cdot \boldsymbol{V})) + \left(\frac{\lambda}{C_p}(1-Le) + \frac{\mu_t}{Pr_t}(1-Le_t)\right)\sum_{i=species} h_i \nabla Y_i \quad \text{(HeatTransfer.10)}$$

$$Q_{vis,G} = \nabla \cdot \left[(\mu + \mu_t)\left(2\hat{\boldsymbol{S}} - \frac{2}{3}(\nabla \cdot \boldsymbol{V})\hat{\boldsymbol{I}}\right) \cdot \boldsymbol{V}\right] \quad \text{(HeatTransfer.11)}$$

In rotating coordinate system, total enthalpy H is replaced by rhotalpy R in Eq. (HeatTransfer.9):

$$R = h + \frac{\boldsymbol{V}^2}{2} - \frac{(\boldsymbol{\omega} \times \boldsymbol{r})^2}{2} \quad \text{(HeatTransfer.12)}$$

Here

$\boldsymbol{V}$ - relative velocity (velocity in rotating coordinate system).

Quantity Q_{user} present in Eqs. (HeatTransfer.1), (HeatTransfer.4), (HeatTransfer.9) is defined by the following relationship

$$Q_{user} = D \cdot T + F + Q_{mod} \quad \text{(HeatTransfer.13)}$$

where

D - coefficient specified by user in field **Preprocessor > Phases > Phase #i > Physical processes > Heat transfer > D**

F - free term, specified by user in field **Preprocessor > Phases > Phase #i > Physical processes > Heat transfer > F**

Q_{mod} - heat production rate specified by user in **Modifier Heat source**

Calculation of radiation energy density E_r see in **Equations for method P1**.

The enthalpy of a mixture is defined by the following relationship:

$$h(T)=\sum_{i=1}^{N} h_i(T)Y_i \qquad \text{(HeatTransfer.14)}$$

$$h_i(T_{abs})=h_0(298.15)+\int_{298.15}^{Tabs} C_p(T)dT \qquad \text{(HeatTransfer.15)}$$

Eq. (HeatTransfer.14) is used to calculate the temperature.

If, in some part of the computational domain, anisotropic heat transfer is specified, the heat flow is calculated as follows:

$$\mathbf{J}_q=-\lambda\hat{\mathbf{D}}\cdot\nabla T+\sum_{i=species} h_i\mathbf{J}_i \qquad \text{(HeatTransfer.16)}$$

Here

$\hat{\mathbf{D}}$ is a 3x3 matrix with coefficients.

10.5.2.4 Boundary conditions

The current section describes boundary conditions for variable **Temperature**, corresponding to different boundary templates.

When a flow of a fluid bounded by a free surface (the second continuous Phase is "vacuum") is simulated, it is possible to take account of the heat exchange between the fluid and the ambient medium (which is not modeled). For this purpose open window of tree element **Preprocessor > Models > Model #i > Phase interaction > Continuous-vacuum**, select **Ext. heat exchange = Yes**, and specify values of parameters **Heat exchange coef**. and **T of external medium**. After that boundary condition **External heat exchange** will automatically set up on the free surface (see section Template 'Wall').

This section has individual numeration of equations.

10.5.2.4.1 Template 'Wall'

Temperature

User specifies the value of relative static temperature at the given boundary:

$$T_b=T_{user} \qquad \text{(HeatTransfer-BC.1)}$$

Zero gradient

User specifies nothing. The condition assumes:

$$\left.\frac{\partial T}{\partial y}\right|_b=0 \qquad \text{(HeatTransfer-BC.2)}$$

$$T_b=T_{cell} \qquad \text{(HeatTransfer-BC.3)}$$

Note, that the heat flux might be non-zero when the diffusion fluxes of the mixture components are non-zero at the given boundary.

The convective flux of enthalpy is zero in the absence of the mass blow-in from the wall:

$$\rho V_n h\big|_b=0 \quad \text{or} \quad \rho V_n H\big|_b=0 \qquad \text{(HeatTransfer-BC.4)}$$

Flux

User specifies the value of heat flux at the given boundary:

$$J_{q,b}=J_{user} \qquad \text{(HeatTransfer-BC.5)}$$

External heat exchange

User specifies the value of relative static temperature of the external medium (parameter **T of external medium**):

$$T_{ext} = T_{user} \qquad \text{(HeatTransfer-BC.6)}$$

the heat-transfer coefficient (parameter **Heat-transfer coef.**) and wall emissivity (parameter **Blackness**):

$$\alpha_{ext} = \alpha_{user}\ ,\quad \varepsilon_w = \varepsilon_{user} \qquad \text{(HeatTransfer-BC.7)}$$

The wall temperature and the heat flux are computed using these quantities. The condition assumes the following balance relationship:

$$J_{q,b} = \alpha_{ext}\left(T_{ext} - T_w\right) + \sigma_{rad}\varepsilon_w\left(\left(T_{ext} + T_{ref}\right)^4 - \left(T_w + T_{ref}\right)^4\right) \qquad \text{(HeatTransfer-BC.8)}$$

Here

σ_{rad} is the Stefan-Boltzmann constant, $5.67\cdot10^{-8}$ [W m^{-2} K^{-4}],

T_{ref} is the reference temperature.

This relationship allows obtaining distributions of temperature and heat flux over the given surface with account of convective and radiative heat exchange with external medium.

Parameters **T of external medium**, **Heat-transfer coef.**, and **Blackness** are set in properties of the element **Boundary conditions > B. Cond. #N > Temperature (Phase #N)**.

Recommendations when you use the boundary condition **External heat exchange** in the template **Wall**:

- for the radiation model **P1** use the boundary condition **Calculating of radiation flux density**.
- for the radiation model **Discrete-ordinates method** use the boundary condition **Opaque wall**.

External conjugate

User specifies nothing. This condition is used for linking the temperature fields obtained in the *FlowVision* and *Abaqus* software suits. The temperature distribution over the common boundary is imported from *Abaqus* to *FlowVision*. It is used as boundary condition for the energy equation in *FlowVision*. The heat flux distribution over the common boundary is exported from *FlowVision* to *Abaqus*. If connection with *Abaqus* is absent, boundary condition **Zero gradient** is automatically set at the given surface.

Radiation equilibrium

If several **Substances** are present in the **Phase** (the **Mass transfer** process is activated), this boundary condition assumes that the solid surface (wall), to which the condition is assigned, is chemically neutral. Correspondingly, the diffusion fluxes of **Substances** onto the wall are zero. The energy balance at the surface is expressed as follows:

$$J_{q,diff,n} + J_{rad,n} = J_{q,n}\ ,$$

$$J_{q,diff,n} = -\left(\lambda + C_p\frac{\mu_t}{Pr_t}\right)\nabla_n T\ ,$$

$$J_{rad,n} = \sigma_{rad}\,\varepsilon_w\left(T^4_{w,abs} - T^4_{\infty,abs}\right)$$

Here

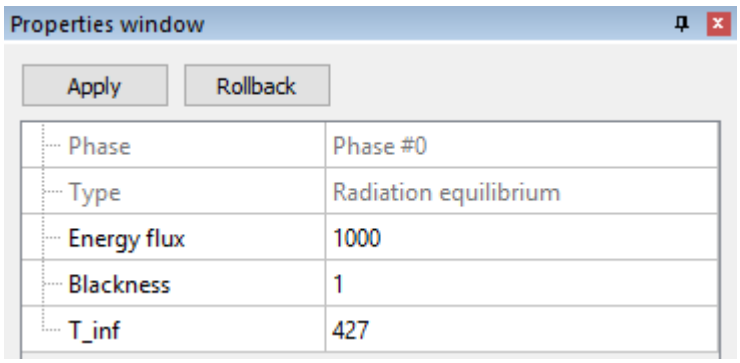
σ_{rad} is the Stefan-Boltzmann constant, $5.67\cdot10^{-8}$ [W m^{-2} K^{-4}].

The following values are specified in the program's interface, in the **Properties** window of the element **Boundary conditions > B.Cond. #N > Temperature (Phase #N)**:

$J_{q,n}$ is the total specific flux of energy from the wall to gas/liquid, it is specified by the parameter **Energy flux**.

ε_w is the wall emissivity, it is specified by the parameter **Blackness**.

$T_{\infty,abs}$ is absolute temperature, which determines the radiation energy flux falling onto the wall; it is equal to sum of the **Reference temperature** and the relative temperature at infinity (which is specified by the parameter **T_inf**).



Parameters of the radiation equilibrium; they are specified in properties of the element **Boundary conditions > B.Cond. #N > Temperature (Phase #N)**

It is assumed that the diffusion and radiation energy fluxes are normal to the surface. The surface temperature is found iteratively from the following equation:

$$J_q = \left(\lambda + C_p \frac{\mu_t}{Pr_t}\right)\frac{T_{w,abs} - T_{c,abs}}{y_c} + \sigma_{rad}\varepsilon_w(T_{w,abs}^4 - T_{\infty,abs}^4)$$

Here y_c is the distance from the surface to the center of a cell adjacent to the surface.

10.5.2.4.2 Template 'Inlet/Outlet'

Temperature

See Eq. (HeatTransfer-BC.1).

Total temperature

User specifies the value of relative total temperature:

$$T_{tot,b} = T_{user} \qquad \text{(HeatTransfer-BC.9)}$$

Static temperature is found from the following relationship:

$$H(T_{tot}) = h(T) + \frac{\boldsymbol{V}^2}{2} \qquad \text{(HeatTransfer-BC.10)}$$

In the case of constant heat capacity

$$T_{tot} = T + \frac{\boldsymbol{V}^2}{2C_P} \qquad \text{(HeatTransfer-BC.11)}$$

10.5.2.4.3 Template 'Free outlet'

Zero gradient

See Eqs. (HeatTransfer-BC.2), (HeatTransfer-BC.3).

Temperature

See Eq. (HeatTransfer-BC.1).

10.5.2.4.4 Template 'Connected'

Conjugate

If user selects **Connection type** = **Conjugate temperature** in window **Create binder condition**, condition **Conjugate** is automatically set on the given boundary. The user specifies the intensity of the volume energy source in the cells adjacent to the surface in terms of specific heat flux (i. e. the required energy source is multiplied by the volume of the given cell and divided by the surface area present in the cell):

$$J_{q,b} = J_{q,user} \qquad \text{(HeatTransfer-BC.12)}$$

This is parameter **Flux** in window **Boundary condition > Temperature** (it is equal to 0 by default).

Sign of the parameter **Wall heat-transfer coef.** in properties of the **Boundary condition** defines if the program will do simple or effective simulation of the heat-conducting surface (see descriptions below). By default **Wall heat-transfer coef. = -1** and the program makes simple simulation of the heat transfer connection.

When the simple heat transfer connection is used (**Wall heat-transfer coef. < 0**), the condition of continuity of the total energy flux has the following form:

$$J_{q,b,1} + J_{rad,b,1} = J_{q,b,2} + J_{rad,b,2} \qquad \text{(HeatTransfer-BC.13)}$$

This relationship defines the temperature of the common surface.

You can either specify effective heat transfer connection, which simulates a virtual wall of finite thickness that is not resolved by the geometry. To do so, the program uses the heat transfer coefficient α_w, which is specified by a positive value of the **Wall heat-transfer coef.** parameter (**Wall heat-transfer coef. =** α_w **> 0**). In this case the condition of continuity of the total energy flux has the following form:

$$J_{q,b,1} + J_{rad,b,1} = \alpha_w(T_{w2} - T_{w1}) = J_{q,b,2} + J_{rad,b,2} \qquad \text{(HeatTransfer-BC.14)}$$

Condition (14) defines temperatures of the two surfaces of the virtual wall.

The heat transfer coefficient α_w [W·m^{-2} K^{-1}] is defined by the formula:

$$\alpha_w = \frac{\lambda}{h}$$

where λ is thermal conductivity of the virtual wall's material and h is the virtual wall's thickness.

Specifying the **Wall heat-transfer coef.** parameter on either one of the connected **Boundary conditions** automatically causes its specifying on the other connected **Boundary conditions** of the same connected surface.

If **Phase #i > Physical processes > Radiation** = **(none) | Optically thin layer**, then

$$J_{rad,b,1} = J_{rad,b,2} = 0 \qquad \text{(HeatTransfer-BC.15)}$$

The temperature values, diffusion heat fluxes and radiation heat fluxes on surfaces of the virtual wall are computed automatically.

Selection **Connection type** = **Conjugate all variables** | **Periodic surface** | **Sliding surface** causes automatic setting of the corresponding condition of matching all the variables on the two surfaces (the user specifies nothing).

10.5.2.4.5 Template 'Nonreflecting'

Riemann

This boundary condition is automatically set for **Temperature** when template **Nonreflecting** is selected. User specifies the value of relative static temperature at infinity:

$$T_\infty = T_{user} \qquad \text{(HeatTransfer-BC.16)}$$

10.5.2.5 Modifiers

Volume heat source

The given modifier specifies a heat source / sink in all the computational cells completely or partly entering the **Object**, on which the **Modifier** is created. User specifies a value of the heat production rate Q_{user} – see equations Eqs. (HeatTransfer.1), (HeatTransfer.4).

10.5.2.6 References

1. Исаченко В. П., Осипова В. А., Сукомел А. С. Теплопередача // Учебник для вузов, Изд. 3-е, Москва, Издательство “Энергия”, 1975, 488 с.

10.5.3 Radiation

The following selection is present in position **Radiation** in window **Preprocessor > Phases > Phase #i > Physical processes**:

- **(none)**
- **P1**
- **Optically thin layer**
- **Discrete-ordinates method**

Below the two implemented models are described.

See details about these models of radiation transfer in sections below.

Recommendations when you use the boundary condition **External heat exchange** in the template **Wall**:

- for the radiation model **P1** use the boundary condition **Calculating of radiation flux density**.
- for the radiation model **Discrete-ordinates method** use the boundary condition **Opaque wall**.

This section has individual numeration of equations for each model of radiation transfer. List of references is common for all the radiation transfer models.

10.5.3.1 Notations

Notation	Physical quantity	Name in *FlowVision*	Dimension
C_P	specific heat capacity	**Specific heat**	J kg^{-1} K^{-1} = = m^2 s^{-2} K^{-1}
E_r	radiation energy density	**Radiation density**	W m^{-2} = kg s^{-3}
E_{eq}	equilibrium radiation energy density		W m^{-2} = kg s^{-3}
n	refraction coefficient of the medium	**Refraction index**	
$J_{q,b}$	modulus (the absolute value) of diffusive heat flux per unit area density from the boundary		W m^{-2} = kg s^{-3}
$J_{rad,b}$	modulus (the absolute value) of radiation heat flux per unit area density from the boundary		
Q_b	volume source of energy in a near-boundary cell submitted as heat flux (recalculated using the cell volume and the area of the conjugating surface present in the cell)		
Q_{rad}	source term due to radiation in the energy equation		W m^{-3} = kg m^{-1} s^{-3}
P_{ext}	power of the source of radiation (it is used in the **Point Source** boundary condition)	**Source power**	W

Notation	Physical quantity	Name in *FlowVision*	Dimension
φ	half of the apex angle of the cone, in which radiation from a point source transfers (it is used in the **Point Source** boundary condition)	**Angle**	degree
T_{abs}	absolute temperature		K
y	distance to nearest wall (along the normal)	**Distance to wall**	m
α	integral (over spectrum) absorption coefficient for medium	**Absorption coefficient**	m^{-1}
α_{λ_n}	spectral absorption coefficient		
α_w	coefficient of heat transfer of the virtual wall in the case when the effective heat transfer connection for connected boundary conditions is used	**Wall heat transfer coef.**	$W\ m^{-2}\ K^{-1}$ = $= kg\ s^{-3}\ K^{-1}$
β	scattering coefficient for medium	**Dispersion coefficient**	m^{-1}
ε_b	surface emissivity		
ε_{λ_n}	spectral surface emissivity		
$\sigma_{rad} = 5.67 \cdot 10^{-8}$	Stefan-Boltzmann constant	**Stefan–Boltzmann constant**	$W\ m^{-2}\ K^{-4}$
$h = 6.62607004 \times 10^{-34}$	Planck constant	**Planck constant**	J·s
λ	wavelength	**Radiation wavelength**	m
I	radiation intensity		W/m^2
I_{λ_n}	spectral radiation intensity		
f_d	index of diffusion reflection on a wall $0 \le f_d \le 1$	**Diffusive fraction**	
$\boldsymbol{n}$	normal to the surface (directed into the computational domain)		

Indices:
1, 2 - values on the sides of conjugating surface,
b - value on boundary,
cell - value in cell center,
user - value specified by user.

The other notations see in the section **Basic notations**.

10.5.3.2 Parameters

Parameters of the element **Preprocessor > Phases > Phase #i > Physical processes > Radiation**:

Parameter	Permissible values	Default value	Description
Math. model	• **Discrete-ordinates method** • **P1** • **Optically thin layer**		A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.)
Time step coefficient	An arbitrary numerical value determined by the problem (the default value	**1**	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the

Parameter	Permissible values	Default value	Description
	is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different time steps for different processes sometimes accelerates convergence to the steady-state solution. Meaning of different values: **1** = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the problem as a whole). **> 1** = The calculations of the given process are accelerated. **(0, 1)** = The calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations / initialization).		whole project (τ). *Thus:* Own time step of a physical process = τ × **Time step coefficient** Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time.
Refraction index	Determined by the problem	**1**	The refraction coefficient n of the medium
Absorption coefficient*)	Determined by the problem	**1**	The absorption coefficient of the medium, α, [m^{-1}]
Relaxation coefficient	Determined by the problem	**0.2**	The weighting factor, which is used to damper the term $divQ$, given by the radiation model before substituting into the heat transfer equation at the current time step
Dispersion coefficient**)	Determined by the problem	**0**	Dispersion coefficient, [m^{-1}]
Anisotropy dispersion coefficient**)	-1≤a≤1, determined by the problem	**0**	Dimensionless anisotropy dispersion coefficient a (the scattering coefficient)
Number Polar Angle**)	Determined by the problem, it is to be even	**2**	Number of polar angles N_θ
Number Azimuth Angle**)	2×**Number Polar Angle**. This parameter is set automatically based on	**4**	Number of azimuth angles N_φ

Parameter	Permissible values	Default value	Description
	the value of the **Number Polar Angle** parameter.		
Axis for polar angle**)	**Axis X** | **Axis Y** | **Axis Z**	**Axis X**	Axis for the polar angle

*) This parameter is not available in the **Discrete-ordinates method** model (in this case the absorption coefficient of the medium is set individually for each **Spectral band** $\Delta\lambda_n$).

) These parameters are available in the **Discrete-ordinates method model only.

When the **Discrete-ordinates method** is used, the following parameters of spectral bands are set in properties of elements **Radiation > Spectrum > Spectral band #N**:

Parameter	Permissible values	Default value	Description
Absorption	Determined by the problem	**1**	Spectral absorption coefficient α_{λ_n}, [m^{-1}]
Begin wavelength	Determined by the problem	10^{-20}	Boundaries of the spectral bands, λ_{n-1} and λ_n, [m]
End wavelength	Determined by the problem	10^{20}	

Some parameters are specified in boundary conditions, in properties of the variable **Radiation Intensity**.

10.5.3.3 P1

The given model represents the radiation transfer of energy as diffusion process. It assumes that:

- local thermal equilibrium occurs in the medium
- radiation is isotropic
- scattering is isotropic

The model assumes also that the radiation energy flux is proportional to the gradient of the radiation energy density: photons diffuse similarly to molecules in gas. Usually, this assumption is valid when the anisotropy of the radiation field is weak. The condition of weak anisotropy is satisfied when the photon free path is much less than the characteristic length of the problem. However, the practice of solving different problems of radiation gas dynamics (RGD) shows that diffusion approximation often yields good results also in the situations when this condition is not strictly satisfied [8]. Hence, the accuracy of diffusion approximation is sufficient for many RGD problems.

The description of the diffusion radiation model (model **P1**) see in [8].

This section has individual numeration of equations.

10.5.3.3.1 Equations for method P1

The radiative transfer equation is:

$$\nabla\left(-\frac{4}{3}\frac{1}{\alpha+\beta}\nabla E_r\right)=4\alpha\left(E_{eq}-E_r\right) \qquad \text{(Rad-P1.1)}$$

$$E_{eq}=\sigma_{rad}T_{abs}^4 \qquad \text{(Rad-P1.2)}$$

Expression for the normal component of the radiation flux density from the boundary:

$$J_{rad,b}=-\frac{4}{3(\alpha+\beta)}\frac{\partial E_r}{\partial y}=\frac{2\varepsilon_b}{2-\varepsilon_b}\left(E_{eq,b}-E_{r,b}\right) \qquad \text{(Rad-P1.3)}$$

Simple approximation of the derivative in Eq. (Rad-P1.3)

$$\frac{\partial E_r}{\partial y} = \frac{E_{r,cell} - E_{r,b}}{y_{cell}} \qquad \text{(Rad-P1.4)}$$

yields

$$J_{rad,b} = \frac{2\varepsilon_b}{2-\varepsilon_b} \frac{E_{eq,b} - E_{r,c}}{1 + \frac{2\varepsilon_b}{2-\varepsilon_b} \frac{3(\alpha+\beta)y}{4}} \qquad \text{(Rad-P1.5)}$$

This equation is used in the program for forming the matrix.

The expression for the source term in the energy equation (HeatTransfer.1), (HeatTransfer.4) or (HeatTransfer.9) reads

$$Q_{rad} = 4\alpha n^2 \left(E_r - E_{eq}\right) \qquad \text{(Rad-P1.6)}$$

The integral absorption coefficient is specified by constant or formula in the interface.

In the current implementation of the model, it is assumed that β=0.

10.5.3.3.2 Boundary conditions

The current section describes boundary conditions for variable **Radiation density**, corresponding to different boundary templates.

This section has individual numeration of equations and formulas.

10.5.3.3.2.1 Template 'Wall'

Radiation flux density

This boundary condition is defined by selection:

- **Radiation = P1** in properties of the element **Phase #N > Physical processes**
- *and* **Variables > Radiation density = Radiation flux density** in properties of the boundary condition **Wall**.

The user specifies a value for the radiation energy flux from the given boundary:

$$J_{rad,b} = J_{user} \qquad \text{(Rad-P1-BC.1)}$$

Calculating of radiation flux density

This boundary condition is defined by selection:

- **Radiation = P1** in properties of the element **Phase #N > Physical processes**
- *and* **Variables > Radiation density = Calculating of radiation flux density** in properties of the boundary condition **Wall**.

The user specifies a value for the surface emissivity:

$$\varepsilon_b = \varepsilon_{b,user} \qquad \text{(Rad-P1-BC.2)}$$

Density of the radiation flux from the boundary is calculated automatically:

$$J_{rad,b} = \frac{2\varepsilon_b}{2-\varepsilon_b} \frac{E_{eq,b} - E_{r,cell}}{1 + \frac{2\varepsilon_b}{2-\varepsilon_b} \frac{3(\alpha+\beta)y}{4}} \qquad \text{(Rad-P1-BC.3)}$$

Recommendations when you use the boundary condition **External heat exchange** in the template **Wall**:

- for the radiation model **P1** use the boundary condition **Calculating of radiation flux density**.
- for the radiation model **Discrete-ordinates method** use the boundary condition **Opaque wall**.

10.5.3.3.2.2 Template 'Symmetry'

Symmetry

User specifies nothing. The condition assumes that

$$J_{rad,b} = 0 \qquad \text{(Rad-P1-BC.4)}$$

From Eq. (Rad-P1-BC.4) it follows that the value of density of radiation energy at the boundary equals to the value at the center of the adjacent cell:

$$E_{r,b} = E_{r,cell} \qquad \text{(Rad-P1-BC.5)}$$

10.5.3.3.2.3 Templates 'Inlet/Outlet', 'Free outlet', 'Nonreflecting'

Radiation flux density

User specifies the value of the radiation energy flux density from the boundary:

$$J_{rad,b} = J_{user} \qquad \text{(Rad-P1-BC.6)}$$

10.5.3.3.2.4 Template 'Connected'

Calculating of radiation flux density

If user selects **Connection type** = **Conjugate temperature** in the **Create binder condition** dialog box, condition **Conjugate** is automatically set on the boundary. User specifies a value for the volume source of energy in a near-boundary cell submitted as heat flux (recalculated using the cell volume and the area of the conjugating surface present in the cell) in position **Flux** in window **Subregions > Subregion #i > Boundary conditions > Connected** (template) **> Variables > Temperature**:

$$Q_b = Q_{user} \qquad \text{(Rad-P1-BC.7)}$$

The default value of this parameter is **0**.

Sign of the parameter **Wall heat-transfer coef.** in properties of the **Boundary condition** defines if the program will do simple or effective simulation of the heat-conducting surface (see descriptions below). By default **Wall heat-transfer coef. = -1** and the program makes simple simulation of the heat transfer connection.

When the simple heat transfer connection is used (**Wall heat-transfer coef. < 0**), the condition of continuity of the total energy flux has the following form:

$$J_{q,b,1} + J_{rad,b,1} = J_{q,b,2} + J_{rad,b,2} \qquad \text{(Rad-P1-BC.13)}$$

This relationship defines the temperature of the common surface.

You can either specify effective heat transfer connection, which simulates a virtual wall of finite thickness that is not resolved by the geometry. To do so, the program uses the heat transfer coefficient α_w, which is specified by a positive value of the **Wall heat-transfer coef.** parameter (**Wall heat-transfer coef. =** α_w **> 0**). In this case the condition of continuity of the total energy flux has the following form:

$$J_{q,b,1} + J_{rad,b,1} = \alpha_w(T_{w2} - T_{w1}) = J_{q,b,2} + J_{rad,b,2} \qquad \text{(Rad-P1-BC.14)}$$

Condition (Rad-P1-BC.14) defines temperatures of the two surfaces of the virtual wall.

The heat transfer coefficient α_w [W·m^{-2} K^{-1}] is defined by the formula:

$$\alpha_w = \frac{\lambda}{h}$$

where λ is thermal conductivity of the virtual wall's material and h is the virtual wall's thickness.

Specifying the **Wall heat-transfer coef.** parameter on either one of the connected **Boundary conditions** automatically causes its specifying on the other connected **Boundary conditions** of the same connected surface.

Selection **Connection type** = **Conjugate all variables | Periodic surface | Sliding surface** causes automatic setting of the corresponding condition of matching all the variables on the two surfaces (the user specifies nothing).

10.5.3.4 Optically thin layer

The given model does not require solving the equation for the radiation energy density and specifying boundary conditions. Within this model, the radiation source term in the energy equation does not take into account the radiation flux entering a computational cell:

$$Q_{rad} = -4\alpha n^2 E_{eq}$$

The **Optically thin layer** model is often used for simulation of combustion.

See also: Notations, Parameters.

10.5.3.5 Discrete-ordinates method

The *discrete-ordinates method* is the most comprehensive model of radiation energy transfer. This model is applicable for the whole range of optical depth and allows to solve such problems as:

- surface to surface radiation energy transfer
- taking radiation into account in simulations of combustion
- radiation transfer in semitransparent media

The discrete-ordinates method solves equations of radiation energy transfer for N solid angles Ω_i, each of them matches to a direction $\boldsymbol{s}_i$ that is determined in the absolute Cartesian coordinate system.

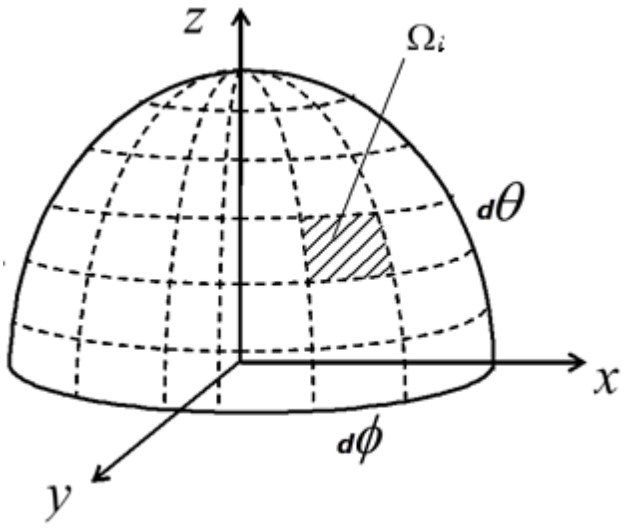
Sum of all these solid angles is 4π:

$$\sum_N \Omega_i = 4\pi$$

Accuracy of the solution and number of the solved equations depend on the number N of the solid angles Ω_i, to which the space is split. The discrete-ordinates method has moderate requirements to computational resources when discretization is typical (no more than 32 solid angles are used) but when number of solid angles is large, substantial computational resources will be required.

Angular discretization

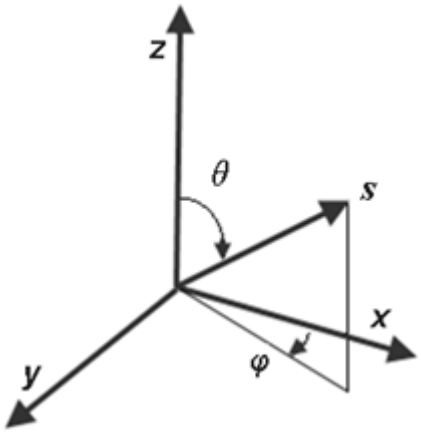
The whole angular space 4π is split into $N_\theta \times N_\varphi$ solid angles. The polar angle θ and the azimuth angle φ are defined in the Cartesian coordinate system as shown on the illustration below:



The polar angle lies in the range $0 \le \theta \le \pi$, the azimuth angle lies in the range $0 \le \varphi \le 2\pi$.

The discretization is done in a manner that angles $d\theta$ and $d\varphi$ are equal, so $N_\theta=2N_\varphi$.

Direction $\boldsymbol{s}$ for a solid angle $d\Omega$ is determined by directional cosines:



10.5.3.5.1 Equations for the discrete-ordinates method

"Gray" approach

The radiation transfer equation describes balance of the radiation energy receiving along direction $\boldsymbol{s}$ into a small element of absorbing, emitting and diffusing medium [1-6]:

$$\frac{dI(\boldsymbol{r},\boldsymbol{s})}{ds}+(\alpha(\boldsymbol{r})+\beta(\boldsymbol{r}))\cdot I(\boldsymbol{r},\boldsymbol{s})=\alpha(\boldsymbol{r})\cdot n^2 I_b+\frac{\beta(\boldsymbol{r})}{4\pi}\int_{4\pi} I(\boldsymbol{r},\boldsymbol{s}')\Phi(\boldsymbol{s},\boldsymbol{s}')\,d\Omega' \qquad \text{(Rad-DO.1)}$$

Here:

$\boldsymbol{r}$ is the radius vector

$\boldsymbol{s}$, $\boldsymbol{s}'$ are unit directing vectors of all possible directions of the radiation transfer

$\Phi(\boldsymbol{s}, \boldsymbol{s}')$ is the phase function of diffusion

$I_b = \sigma T^4/\pi$ is radiation intensity of the absolutely black body

The second term in the left-hand side of the equation (Rad-DO.1) represents intensity attenuation in direction $\boldsymbol{s}$ due to absorption and scattering of the radiation energy. The first term on the right side of the equation is responsible for the increasing of radiation intensity due to emitting the medium. The second term in the right-hand side represents contribution of radiant energy from other directions $\boldsymbol{s}$ due to scattering of the radiation.

Spectral approach

The spectral approach implements simulation of L spectral bands. Each spectral band $\Delta\lambda_n$ is specified by a range of wavelengths $[\lambda_{n-1}, \lambda_n]$.

Within each spectral band the spectral radiation intensity is assumed as not depending on the wavelength, and the spectral absorption coefficient is averaged. Within each spectral band the radiation is simulated using the "gray" approach.

In the spectral approach, radiation intensity of a spectral band $[\lambda_{n-1}, \lambda_n]$ is described by the equation:

$$I_{\lambda_n}(\boldsymbol{r},\boldsymbol{s})=\int_{\lambda_{n-1}}^{\lambda_n} I(\lambda,\boldsymbol{r},\boldsymbol{s})\,d\lambda \qquad \text{(Rad-DO.2)}$$

For each band $\Delta\lambda_n$ the transfer equation for the spectral radiation intensity will be:

$$\frac{\partial I_{\lambda_n}}{\partial t}+\frac{dI_{\lambda_n}(\boldsymbol{r},\boldsymbol{s})}{ds}+(\alpha_{\lambda_n}(\boldsymbol{r})+\beta(\boldsymbol{r}))\cdot I_{\lambda_n}(\boldsymbol{r},\boldsymbol{s})=\alpha_{\lambda_n}(\boldsymbol{r})\cdot n^2 I_{b\lambda_n}+\frac{\beta(\boldsymbol{r})}{4\pi}\int_{4\pi} I'_{\lambda_n}(\boldsymbol{r},\boldsymbol{s}')\,\Phi(\boldsymbol{s},\boldsymbol{s}')d\Omega' \quad \text{(Rad-DO.3)}$$

$$I_{b\lambda_n}=\frac{u_{\lambda_n}\cdot\Delta\lambda_n}{\pi} \quad \text{(Rad-DO.4)}$$

Here u_λ is the spectral power density of the radiation determined by the Planck formula:

$$u(\lambda,T)=\frac{2\pi\hbar c^2}{\lambda^5\left(\exp\left(\frac{\hbar c}{\lambda kT}\right)-1\right)} \quad \text{(Rad-DO.5)}$$

Source term in the energy equation

Formula for the source term in the energy equation for the "gray" approach:

$$Q_{rad}=\alpha\cdot n^2(ER-ER_{eq}) \quad \text{(Rad-DO.6)}$$

Here:

$$ER=\int_{4\pi} I(\boldsymbol{r},\boldsymbol{s})d\Omega \quad \text{(Rad-DO.7)}$$

$$ER_{eq}=4\sigma T^4 \quad \text{(Rad-DO.8)}$$

Formula for the source term in the energy equation for the spectral approach:

$$Q_{rad}=n^2\sum_{\lambda_n}\alpha_{\lambda_n}(ER_{\lambda_{ni}}-ER_{eq\lambda_n}) \quad \text{(Rad-DO.9)}$$

Here:

$$ER_{\lambda_n}=\int_{4\pi} I^l_{\lambda_n}(\boldsymbol{r},\boldsymbol{s})\cdot d\Omega \quad \text{(Rad-DO.10)}$$

$$ER_{eq\lambda_n}=u(\lambda_n,T)\cdot\Delta\lambda_n \quad \text{(Rad-DO.11)}$$

Radiation flux from the wall, the "gray" approach:

$$J_w=\int_{\boldsymbol{s}\cdot\boldsymbol{n}>0} I(\boldsymbol{r},\boldsymbol{s})d\Omega$$

Spectral approach:

$$J_w=\int_{\lambda}\int_{\boldsymbol{s}\cdot\boldsymbol{n}>0} I(\boldsymbol{r},\boldsymbol{s})d\Omega d\lambda$$

Anisotropy dispersion

FlowVision supports simulation of linear anisotropic function of dispersion:

$$\Phi(\boldsymbol{s},\boldsymbol{s}')=1+a\cdot\boldsymbol{s}'\boldsymbol{s} \quad \text{(Rad-DO.12)}$$

The a coefficient can be set within the range [–1, 1].

A positive value of a means that more of the radiation energy is emitted forward than back. A negative value of a means that more of the radiation energy is emitted back then forward. Zero value of a determines isotropic dispersion (scattering) and the radiation is uniformly intensive in all directions.

10.5.3.5.2 Boundary conditions

In this section we consider boundary conditions for the variable **Radiation Intensity (Phase #N)** that corresponds to various boundary templates.

The boundary conditions can represent opaque, semitransparent and transparent (with free outlet of radiation) walls.

Recommendations when you use the boundary condition **External heat exchange** in the template **Wall**:
- for the radiation model **P1** use the boundary condition **Calculating of radiation flux density**.
- for the radiation model **Discrete-ordinates method** use the boundary condition **Opaque wall**.

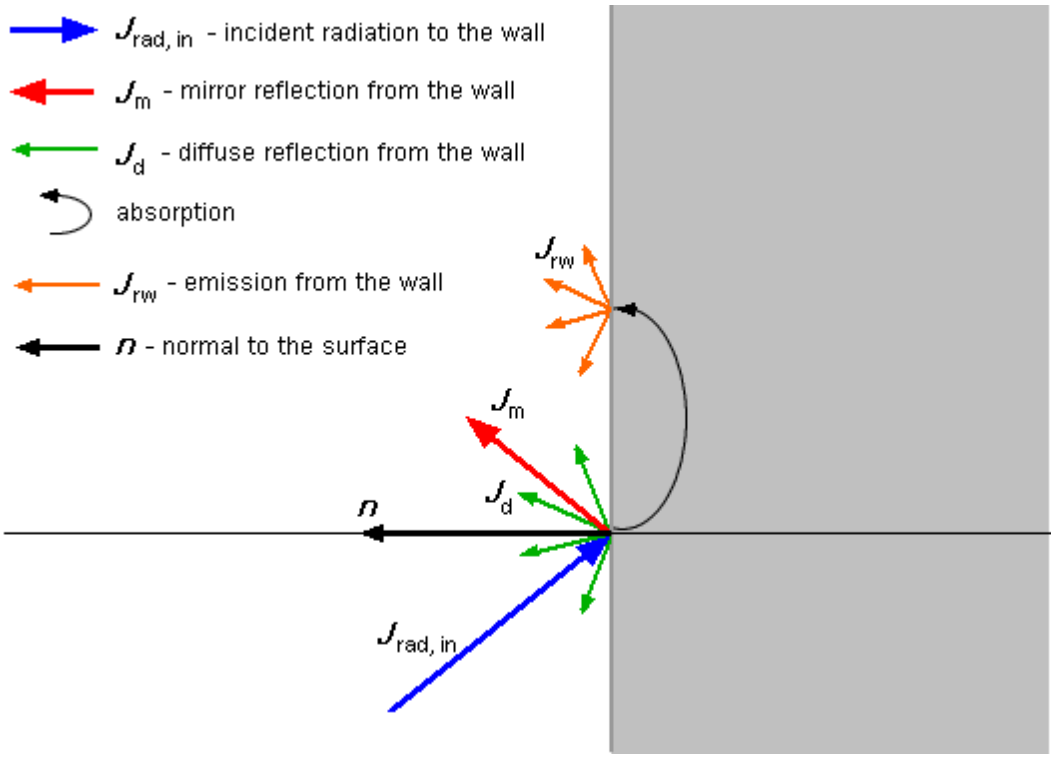
Opaque wall

The following parameters are set in properties of the element ***(boundary condition)*** **> Radiation Intensity (Phase #N)**:

Parameter	Description
Blackness	$\varepsilon_w(\lambda)$, blackness of the wall
Diffusive fraction	$f_d(\lambda)$, the fraction of diffusion reflection from the wall (index of diffusion reflection)

These parameters generally depend on the wavelength, so they can be different in different spectral bands. Their dependency on the wavelength λ is specified by a formula or a table, which includes the variable **Radiation wavelength** as an argument.

When the spectral approach is used, the conditions are fulfilled for each spectral band.



Interaction between radiation and an opaque wall

Radiation flux density from the wall is determined by the formula:

$$J_{rad,b} = J_{rw} + J_d + J_m \qquad \text{(Rad-DO.13)}$$

Here:

$J_{rw} = n^2 \cdot \varepsilon_w \cdot \sigma \cdot T^4$ – emission from the surface of the wall

$J_d = f_d \cdot (1-\varepsilon_w) J_{rad,in}$ – diffuse reflection from the wall

$J_m = (1-f_d) \cdot (1-\varepsilon_w) J_{rad,in}$ – mirror (specular) reflection from the wall

f_d – fraction of diffusion reflection from the wall

$J_{rad,in} = \int\limits_{\boldsymbol{s}\cdot\boldsymbol{n}<0} I_{cell}\, \boldsymbol{s}\cdot\boldsymbol{n}\, d\Omega$ – incident radiation flux to the wall

For each direction $\boldsymbol{s}$ from the wall the radiant energy intensity is sum of the following values:

- intensity of emission from the surface of the wall $J_{rw}/\pi = n^2 \cdot \varepsilon_w \cdot \sigma \cdot T^4/\pi$
- diffuse reflection $J_d/\pi = f_d \cdot (1-\varepsilon_w) J_{rad,in}/\pi$
- mirror (specular) reflection $(1-f_d) \cdot (1-\varepsilon_w) I_{cell}(\boldsymbol{s}_m)$

Here $\boldsymbol{s}_m$ is the direction, from which the mirror reflection in the direction $\boldsymbol{s}$ occurs. The direction $\boldsymbol{s}_m$ is determined based on that the angle of incidence is equal to the angle of reflection:

$\boldsymbol{s}_m = \boldsymbol{s} + 2(\boldsymbol{n}-\boldsymbol{s})$

So radiation intensity from the wall in direction $\boldsymbol{s}$ is:

$$I_w(\boldsymbol{s}) = n^2 \cdot \varepsilon_w \cdot \sigma \cdot T_w^4 / \pi + f_d \cdot (1-\varepsilon_w) J_{rad,in} / \pi + (1-f_d) \cdot (1-\varepsilon_w) \cdot I_{cell}(\boldsymbol{s}_m) \qquad \text{(Rad-DO.14)}$$

Conjugation at the boundary of two phases

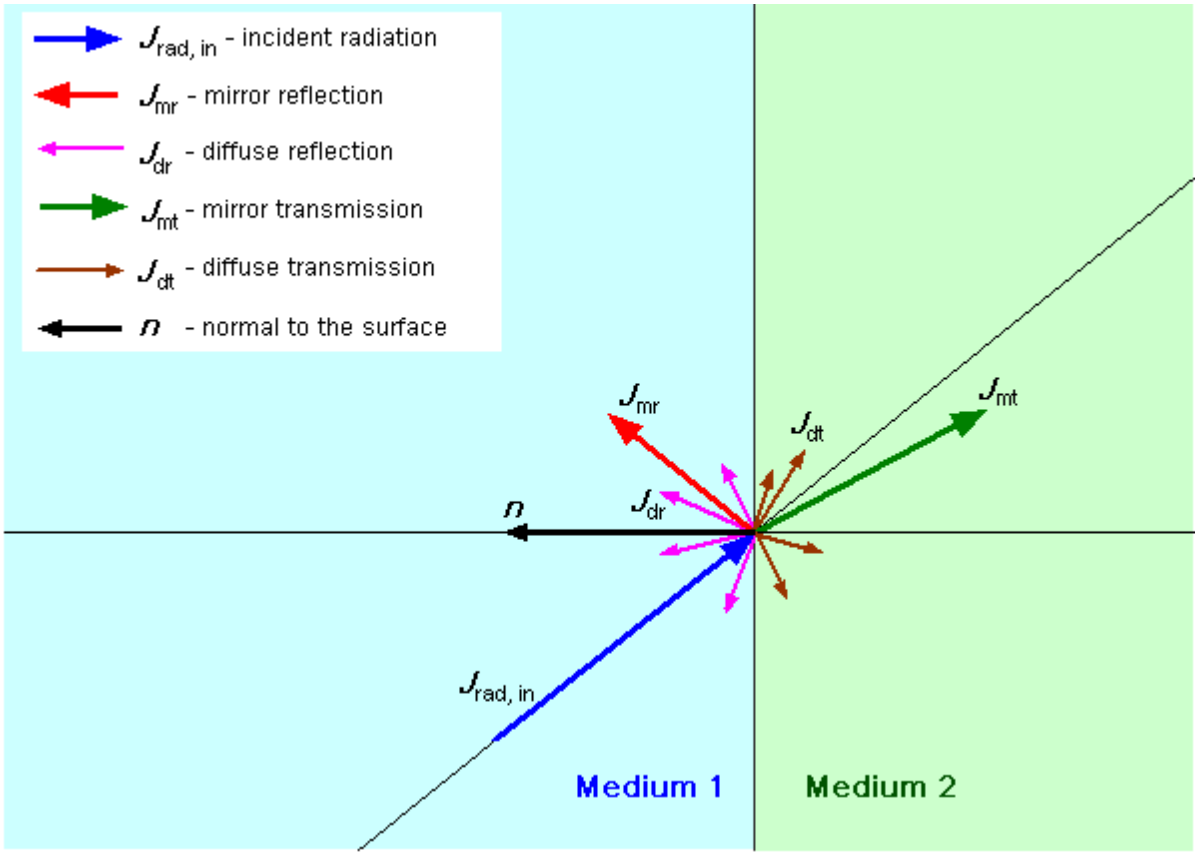
The **Conjugation at the boundary of two phases** boundary condition is used to simulate surfaces between two **Phases** (liquid/solid body, liquid/gas).

The following parameter is set in properties of the element ***(boundary condition)*** **> Radiation Intensity (Phase #N)**:

Parameter	Description
Diffusive fraction	$f_d(\lambda)$, the fraction of diffusion reflection from the semitransparent wall

When the spectral approach is used, the conditions for the internal semitransparent wall are fulfilled for each spectral band.

On the surface between two substances with different optical properties the incident radiation from medium 1 can transfer to medium 2 (diffusively and specularly) and also reflect from the contact surface back to medium 1 (diffusively and specularly), see illustration below.



Interaction between radiation and a semitransparent wall

The substances can be liquids or solids.

Ratios of mirror (specular) and diffusion reflection and transmission through the wall are determined by the f_d parameter.

When f_d=0, reflection and transmission are totally mirror-like (specular). When f_d=1, reflection and transmission are totally diffusive.

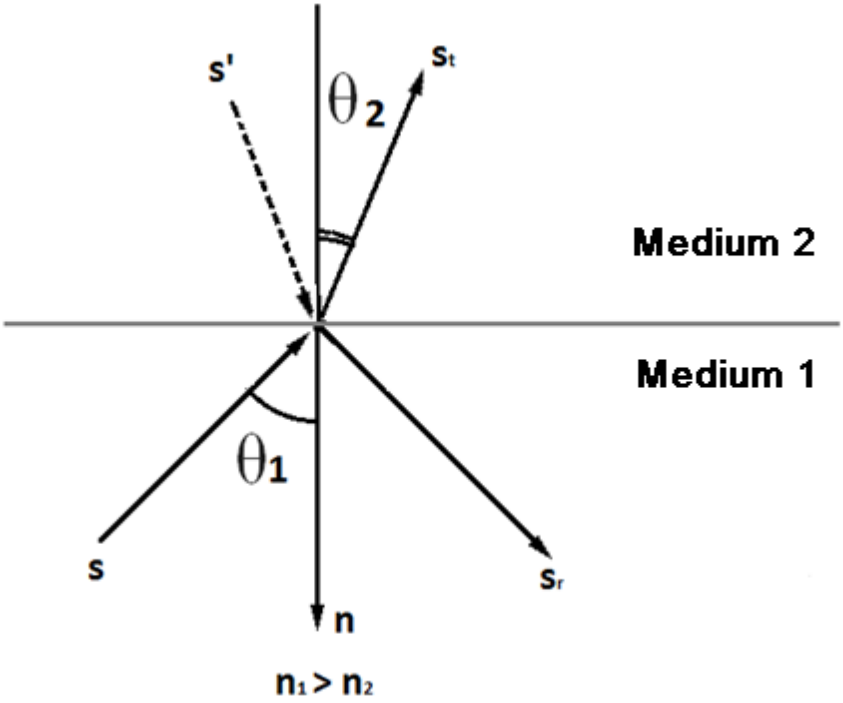
In the general case (when 0<f_d<1) the following formula is applied:

$$J_{w,1} = f_d \cdot \left(J_{dr,1} + J_{dt,2}\right) + (1 - f_d)\left(J_{mr,1} + J_{mt,2}\right) \qquad \text{(Rad-DO.15)}$$

Some particular cases of mirror (specular) and diffusive semitransparent wall are considered below.

Specular semitransparent wall and $n_1 > n_2$

For the case of a specular (f_d=0) semitransparent wall, when the refraction coefficient of the first medium is greater then the refraction coefficient of the second medium ($n_1 > n_2$), part of the incident radiation coming in the direction $\boldsymbol{s}$, reflects from the inter-phase surface to the first medium in the direction $\boldsymbol{s}_r$, as shown on the illustration below.



Specular reflection and refraction on an inter-phase surface

Radiation from medium 1 to medium 2 will be transferred in the direction corresponding to the angle θ_2:

$$\sin\theta_2 = \frac{n_1}{n_2}\sin\theta_1 \qquad \text{(Rad-DO.16)}$$

Here θ_1 is the angle of incidence and θ_2 is the angle of reflection.

According to [5], fraction of the energy, which is transferred from direction $\boldsymbol{s}$ to direction $\boldsymbol{s}_r$ into medium 1, is determined by the formula:

$$r_1 = \frac{1}{2}\left(\frac{n_1\cos\theta_2 - n_2\cos\theta_1}{n_1\cos\theta_2 + n_2\cos\theta_1}\right)^2 + \frac{1}{2}\left(\frac{n_1\cos\theta_1 - n_2\cos\theta_2}{n_1\cos\theta_1 + n_2\cos\theta_{21}}\right)^2 \qquad \text{(Rad-DO.17)}$$

Intensity $I_{m,w,1}(\boldsymbol{s}_r)$ of radiation from the boundary surface into medium 1 in direction $\boldsymbol{s}_r$ is the sum of radiation reflected from the contact surface to medium 1 and reflection transferred from medium 2:

$$I_{m,w,1}(\boldsymbol{s}_r) = r_1 \cdot I_{w,1}(\boldsymbol{s}) + (1 - r_2)\cdot I_{w,2}(\boldsymbol{s}') \qquad \text{(Rad-DO.18)}$$

Respectively, radiation intensity from the boundary surface into medium 2 in direction $\boldsymbol{s}_t$ will be:

$$I_{m,w,2}(\boldsymbol{s}_t) = r_2 \cdot I_{w,2}(\boldsymbol{s}') + (1 - r_1)\cdot I_{w,1}(\boldsymbol{s}) \qquad \text{(Rad-DO.19)}$$

Here $r_2 = r_2(\boldsymbol{s}')$ is the transmissivity coefficient.

Specular semitransparent wall and $n_1 < n_2$

In this case (f_d=0 and $n_1 < n_2$), energy of radiation coming from medium 1 in the solid angle 2π will be transferred to medium 2 in a cone with apex angle θ_c:

$$\theta_c = \arcsin\left(\frac{n_1}{n_2}\right) \quad \text{(Rad-DO.20)}$$

Respectively, radiation coming from medium 2 in such cone will be transferred into medium 1 in the solid angle 2π. Radiation that is outside of this cone will be totally reflected into medium 2.

Diffusive semitransparent wall

In the other particular case, when f_d=1, both reflection and refraction are totally diffusive. The contact surface is a diffusive source.

Intensity of reflected and transmitted radiation doesn't depend on direction $\boldsymbol{s}$ and is equal to the average value by a hemisphere.

When n_1>n_2 radiation intensity from the contact surface to direction into medium 1 and medium 2 will be:

$$I_{d,w,1} = \frac{r_{d,1} J_{rad,in,1} + (1 - r_{d,2}) J_{rad,in,2}}{\pi} \quad \text{(Rad-DO.21)}$$

$$I_{d,w,2} = \frac{r_{d,2} J_{rad,in,2} + (1 - r_{d,1}) J_{rad,in,1}}{\pi} \quad \text{(Rad-DO.22)}$$

Here:

$$J_{rad,in,1} = \int_{\boldsymbol{s}\cdot\boldsymbol{n}<0} I_{w,1}\, \boldsymbol{s}\cdot\boldsymbol{n}\; d\Omega \quad \text{(Rad-DO.23)}$$

$$J_{rad,in,2} = \int_{\boldsymbol{s}\cdot\boldsymbol{n}\geq 0} I_{w,2}\, \boldsymbol{s}\cdot\boldsymbol{n}\, d\Omega \quad \text{(Rad-DO.24)}$$

According to [6], coefficients $r_{d,1}$ and $r_{d,2}$ are determined by formulae:

(Rad-DO.25)

$$r_{d,2} = \frac{1}{2} + \frac{(3n+1)(n-1)}{6(n+1)^2} + \frac{n^2(n^2-1)^2}{(n^2+1)^3}\ln\left(\frac{n-1}{n+1}\right) - \frac{2n^3(n^2+2n-1)}{(n^2+1)(n^4-1)} + \frac{8n^4(n^4+1)}{(n^2+1)(n^4-1)^2}\ln(n) \quad \text{(Rad-DO.26)}$$

Here $n = n_1 / n_2$.

When n_1<n_2 the same algorithm is used except for substituting n_1 for n_2 and vice versa.

When n_1=n_2:

$$r_{d,1} = r_{d,2} = 0.5 \quad \text{(Rad-DO.27)}$$

$$I_{d,w,1} = I_{d,w,1} = \frac{J_{rad,in,1} + J_{rad,in,2}}{2\pi} \quad \text{(Rad-DO.28)}$$

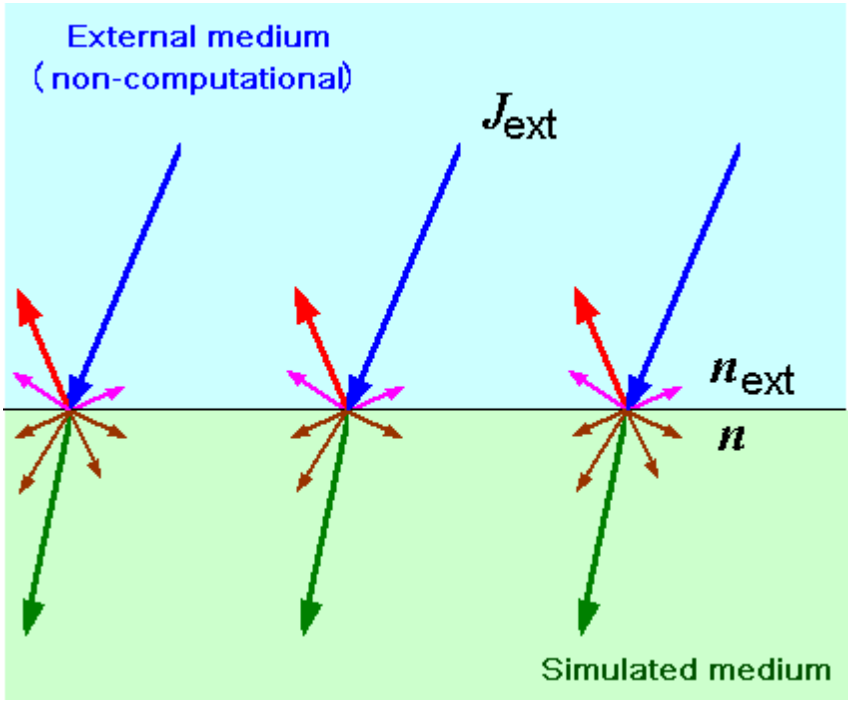
Transparent Wall

The boundary condition **Transparent Wall** assumes a free outlet of radiation:

$$\nabla \boldsymbol{I}\cdot\boldsymbol{n} = 0 \quad \text{(Rad-DO.29)}$$

Radiation flux density

The **Radiation flux density** boundary condition is used to simulate plane-parallel radiation flux with specified intensity J_{ext} that is incident at an angle on the simulated medium from external medium with refraction coefficient n_{ext}:



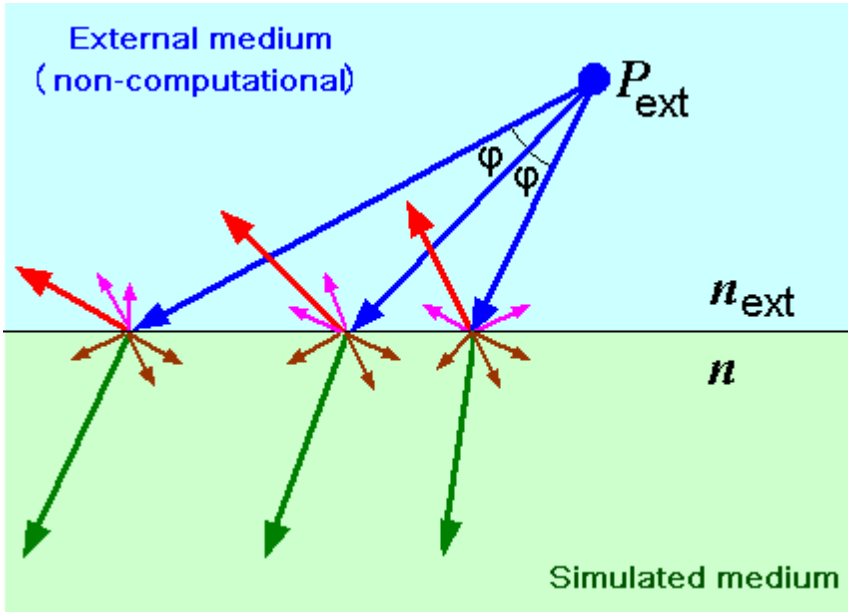
The following parameters are set in properties of the element ***(boundary condition)*** **> Radiation Intensity (Phase #N)**:

Parameter	Description
Diffusive fraction	$f_d(\lambda)$, the fraction of diffusion reflection from the surface. This parameter generally depends on the wavelength. Their dependency on the wavelength λ is specified by a formula or a table, which includes the variable **Radiation wavelength** as an argument.
Refraction index	n_{ext}, the refraction coefficient of the external medium
Radiation flux	J_{ext}, modulus (absolute value) of flux of the plane-parallel radiation from the external medium, [W/m^2]
Direction of radiation flux > X	Direction of plane-parallel radiation from the external medium specified componentwise
Direction of radiation flux > Y	
Direction of radiation flux > Z	

Radiation transport to the simulated medium is described by formulae (Rad-DO.15) — (Rad-DO.28).

Point Source

The **Point Source** boundary condition is used to simulate radiation flux from a point source with specified power P_{ext}, that locates in external medium with refraction coefficient n_{ext}:



The radiation transfers within a cone with apex angle 2φ.

The following parameters are set in properties of the element ***(boundary condition)*** **> Radiation Intensity (Phase #N)**:

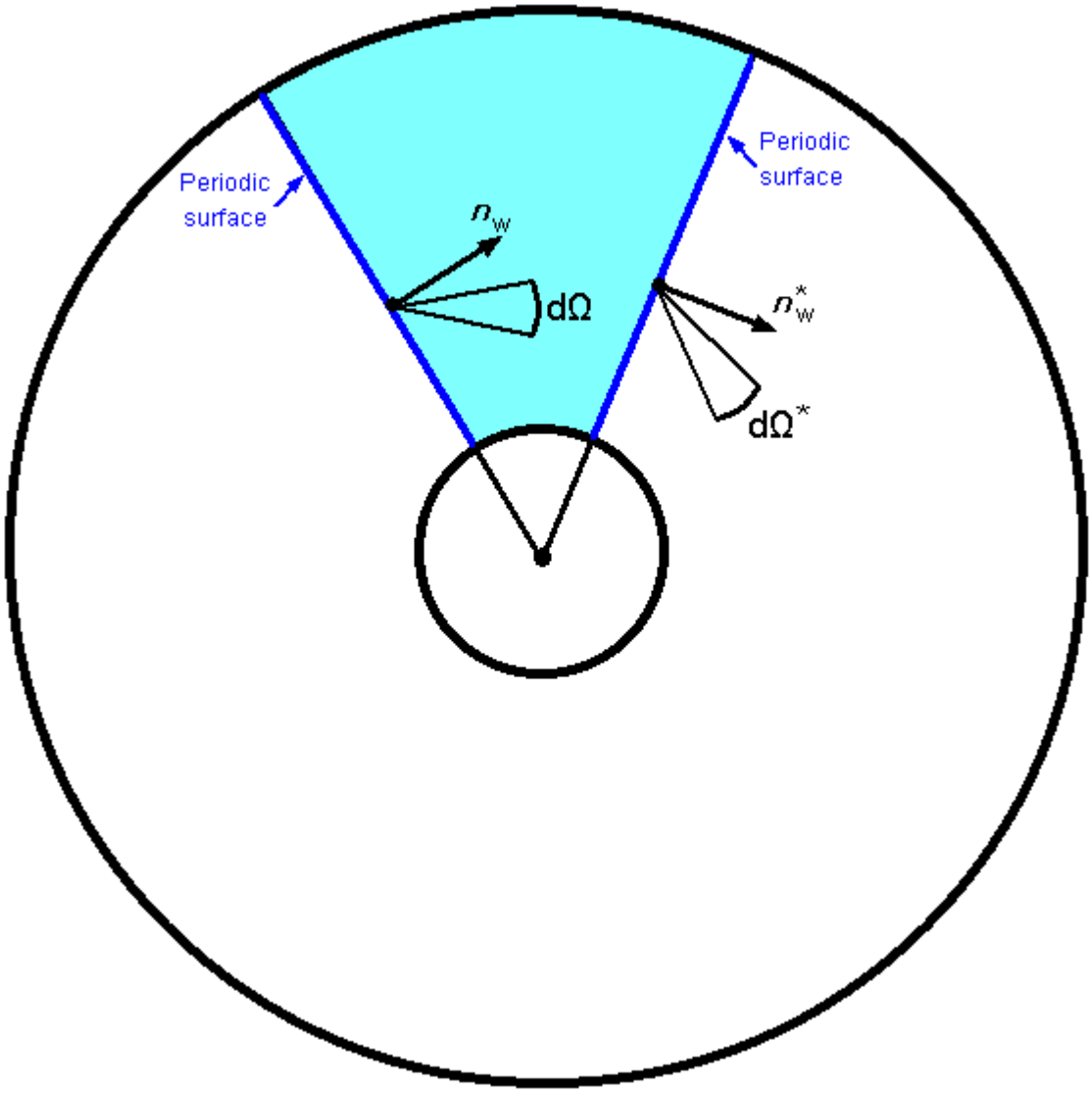
Parameter	Description
Diffusive fraction	$f_d(\lambda)$, the fraction of diffusion reflection from the surface. This parameter generally depends on the wavelength. Their dependency on the wavelength λ is specified by a formula or a table, which includes the variable **Radiation wavelength** as an argument.
Refraction index	n_{ext}, the refraction coefficient of the external medium
Source power	P_{ext}, power of the point source of radiation, [W]
Source coordinates > X	Coordinates of the point source of radiation, [m]
Source coordinates > Y	
Source coordinates > Z	
Angle	φ, half of the apex angle of the cone, in which radiation from the point source transfers, [degree]. Possible range: $0^o<\varphi\leq 180^o$.
Direction of radiation flux > X	Direction of the axis of symmetry of the cone, in which radiation from the point source transfers. This direction is specified componentwise.
Direction of radiation flux > Y	
Direction of radiation flux > Z	

Radiation transport to the simulated medium is described by formulae (Rad-DO.15) — (Rad-DO.28).

Periodic surfaces

On **Periodic surfaces** the program set equal distribution of radiation intensities for radiation incoming on one side of a surface and emitting from the other side, see [7].

On periodic surfaces, local coordinate systems are constructed, which are obtained by applying the transition matrices to the absolute coordinate system. For a local coordinate system, angular discretization is constructed the same as for an absolute coordinate system.



Forming the angular discretization on periodic surfaces

Relations between solid angles on periodic boundary conditions are determined by:

$$\boldsymbol{I}^{l'} = \sum_i f^i_{l'} \boldsymbol{I}^i \qquad \sum_i f^i_{l'} = 1 \tag{Rad-DO.30}$$

Here $f^i_{l'}$ is the part of the solid angle $d\Omega_l$ that is inside the solid angle $d\Omega^*_{l'}$.

Sliding surfaces

For **Sliding surface** the same technique is used as for **Periodic surfaces**.

10.5.3.5.2.1 Template 'Wall'

Opaque wall

See description above.

10.5.3.5.2.2 Template 'Symmetry'

Symmetry

User specifies nothing. The condition assumes that

$$J_{rad,b} = 0$$

This template corresponds to the boundary condition **Opaque wall** with parameters ε_w=0 and f_d=0 that correspond to total mirror (specular) reflection from the wall.

10.5.3.5.2.3 Templates 'Inlet/Outlet', 'Free outlet', 'Non-reflecting'

Transparent wall

These boundary condition templates assumes that radiation leaves the computational domain and comes into the computational domain freely.

The **Transparent Wall** boundary condition is applied.

10.5.3.5.2.4 Template 'Connected'

If user selects **Connection type** = **Conjugate temperature** | **Conjugate all variables** in the **Create binder condition** dialog box, condition **Conjugate** is automatically set on the boundary.

The following options are available for these connections types:

- When the **Discrete-ordinates method** model is set in *only one* of the **Subregions**, then the **Opaque wall** boundary condition is applied.
  The user specifies parameters $\varepsilon_w(\lambda)$ and $f_d(\lambda)$.
- When the **Discrete-ordinates method** model is set in *both* **Subregions**, then the **Internal semitransparent wall** boundary condition is applied.
  The user specifies $f_d(\lambda)$.

Selection **Connection type** = **Periodic surface | Sliding surface** causes automatic setting of the corresponding condition of matching all the variables on the two surfaces (the user specifies nothing).

10.5.3.6 References

1. Белл Д., Глессон С. Теория ядерных ректоров. *Атомиздат, 1974.*

2. Chandrasekhar S. Radiative Transfer, *Oxford University Press, 1950, Chap 11.*

3. H. Chui and G. D. Raithby. "Computation of Radiant Heat Transfer on a Non-Orthogonal Mesh Using the Finite-Volume Method". *Numerical Heat Transfer, Part B. 23. 269–288. 1993.*

4. Chae J.C., Lee H. S., Patankar S.V. Finite Volume Method for Radiation Heat Transfer. *Jornal of Thermophysics and Heat Transfer, Vol. 8, No. 3, 1994.*

5. M. F. Modest. Radiative Heat Transfer. *Series in Mechanical EngineeringMcGraw-Hill. 1993.*

6. R. Siegel and C. M. Spuckler. "Effect of Refractive Index and Diffuse or Specular Boundaries on a RadiatingIsothermal Layer". *J. Heat Transfer. 116. 787–790. 1994.*

7. Литвинцев К.Ю. Использование метода конечных объемов для решения уравнения переноса теплового излучения в трехмерных задачах. *Вычислительные технологии. Т. 19, № 5, 2014.*

8. Четверушкин Б.Н., Математическое моделирование задач динамики излучающего газа, *Москва, Наука: Главная редакция физико-математической литературы, 1985, 304 с.* (Chetwerushkin B.N., Mathematical simulation of the problems of radiation gas dynamics, *Moscow, Nauka: The main publishing house of physical and mathematical literature, 1985, 304 p.*)

10.5.4 Electromagnetohydrodynamics

The following options can be selected for the **EMHD** element in properties of **Preprocessor > Phases > Phase #i > Physical processes**:

- **(none)**
- **Electrodynamics**, which allows the program to simulate interaction of liquid and electrostatic fields. This model doesn't take magnetic fields into account.

- **MHD Potential model**, which allows the program to simulate interaction of liquid and electromagnetic fields.

Below the implemented electromagnetohydrodynamic models are described.

This section has individual numeration of equations and references.

10.5.4.1 Notations

Notation	Description	Name in *FlowVision*	Dimension
$\boldsymbol{D}$	Electric displacement density (vector)		C m^{-2}
$\boldsymbol{E}$	Electric field intensity (vector)	**Electric field intensity**	V m^{-1}
$\boldsymbol{b}$	Magnetic induction (vector) generated by flow of the fluid		T
$\boldsymbol{B}$	Magnetic induction (vector)	**Induction**	
$\boldsymbol{B}_{\text{ext}}$	External magnetic induction (vector) specified by the user		
$\boldsymbol{H}$	Magnetic field strength (vector)		A m^{-1}
$\boldsymbol{F}_L$	The Lorentz force, which is the volume force due to electromagnetic field	**Lorentz force**	N m^{-3}
$\boldsymbol{j}$	Conduction current density (vector)		A m^{-2}
$\boldsymbol{j}_{\text{ext}}$	External conduction current density (vector) specified by the user		
Q_{Joule}	Source term in the energy equation due to Joule heat	**Joule heat**	W m^{-3} = kg m^{-1} s^{-3}
q	Volume density of electric charge		C m^{-3}
q_{ext}	Volume density of the external electric charge specified by the user		
y	Distance to nearest wall	**Distance to wall**	m
σ	Specific conductivity	**Conductivity**	S m^{-1} = Ω^{-1} m^{-1}
φ	Potential of the electric field	**Electrical potential**	V
$\boldsymbol{A}$	Potential of the magnetic field (vector)	**Magnet Potential**	T·m = Wb/m = = kg·m·s^{-2}·A^{-1}
$\varepsilon_0 = 8.842 \cdot 10^{-12}$	Electric constant	**Vacuum permittivity**	F m^{-1}
$\varepsilon_a = \varepsilon_0 \varepsilon$	Absolute dielectric permittivity		
ε	Relative dielectric permittivity	**Permittivity**	
μ_0=1.25663706·10^{-6}	Magnetic constant (permeability of vacuum)	**Vacuum permeability**	N·A^{-2}
$\mu_a=\mu_0 \cdot \mu$	Absolute magnetic permeability		
μ	Relative magnetic permeability	**Permeability**	
$Re_m=\mu\mu_a \cdot \sigma \cdot V \cdot L$	Magnetic Reynolds number		
$deffactor$	Correction factor for the deferred correction method	**Correction factor**	

Indices:

b - value on boundary
cell - value in cell center
user - value specified by user
n - normal projection of a vector to a surface
τ - tangential projection of a vector to a surface

The other notations see in section **Basic notations**.

10.5.4.2 Parameters

Properties of the element **Preprocessor > Phases > Phase #N > Physical processes > EMHD**:

Parameter	Permissible values	Description
Math. model	• **Electrodynamics** • **MHD Potential model**	A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.)
Time step coefficient	An arbitrary numerical value determined by the problem (the default value is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different time steps for different processes sometimes accelerates convergence to the steady-state solution. Meaning of different values: **1** = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the problem as a whole). **> 1** = The calculations of the given process are accelerated. **(0, 1)** = The calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations / initialization).	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the whole project (τ). *Thus:* Own time step of a physical process = τ × **Time step coefficient** Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time.

MHD parameters in the Advanced settings of Solver:

Parameter	Permissible values	Description
The number of iterations	An integer numerical value. The default value is **1**.	The number of iterations for the electromagnetic process in one time step
Exclude from the calculation the equation for the electric potential	**Yes** | **No**	When this parameter is **Yes**, the equation for electric potential is not considered.
Correction factor	A numerical value ≥ **0**. The default value is **0**.	If **Correction factor** > 0, then the electrical potential equation is calculated with correction. This parameter specifies the $deffactor$ value for the *deferred correction method*.

10.5.4.3 Equations

Model "Electrodynamics"

The implemented model is described by the following equations:

$$\nabla \cdot \boldsymbol{j} = 0 \qquad \text{(EMHD.1)}$$

$$\boldsymbol{j} = \sigma \boldsymbol{E} \qquad \text{(EMHD.2)}$$

$$\boldsymbol{E} = -\nabla \varphi \qquad \text{(EMHD.3)}$$

The expression for the Joule source term in the energy equation is as follows

$$Q_{Joule} = \boldsymbol{j} \cdot \boldsymbol{E} = \sigma \boldsymbol{E}^2 \qquad \text{(EMHD.4)}$$

According to ref. [1], the volume force exerted onto a dielectric placed in an electromagnetic field is expressed as:

$$\boldsymbol{F}_L = q\boldsymbol{E} - \frac{1}{8\pi}\boldsymbol{E}^2\nabla\varepsilon + \frac{1}{8\pi}\left(\boldsymbol{E}^2\rho\frac{\partial\varepsilon}{\partial\rho}\right) \qquad \text{(EMHD.5)}$$

Here

q is volume density of electric charge.

In the current implementation of process **EHD**, it is assumed that dielectric permeability does not depend on the density of the medium:

$$\frac{\partial\varepsilon}{\partial\rho} = 0 \qquad \text{(EMHD.6)}$$

In this case

$$\boldsymbol{F}_L = q\boldsymbol{E} - \frac{1}{8\pi}\boldsymbol{E}^2\nabla\varepsilon \qquad \text{(EMHD.7)}$$

Remind that the Lorentz force $\boldsymbol{F}_L$ enters the momentum equation (Motion.2).

The volume density of electric charge in the expression (EMHD.7), is calculated from the following equation

$$\nabla \cdot \boldsymbol{D} = q \qquad \text{(EMHD.8)}$$

$$\boldsymbol{D} = \varepsilon_a \boldsymbol{E} = \varepsilon_0 \varepsilon \boldsymbol{E} \qquad \text{(EMHD.9)}$$

Model "MHD Potential model"

When the magnetic Reynolds number is much less than 1 (Re_m << 1), the system of steady state Maxwell's equations is:

$$\boldsymbol{E} = -\nabla\varphi \qquad \text{(EMHD.10)}$$

$$\boldsymbol{B} = -\nabla \times \boldsymbol{A} \qquad \text{(EMHD.11)}$$

$$\boldsymbol{j} = \sigma\left(-\nabla\varphi + \boldsymbol{V} \times \left(\boldsymbol{B} + \boldsymbol{B}_{ext}\right)\right) \qquad \text{(EMHD.12)}$$

$$\nabla \boldsymbol{j} = 0 \qquad \text{(EMHD.13)}$$

$$q + \nabla \cdot \varepsilon_a \nabla\varphi + q_{external} = 0 \qquad \text{(EMHD.14)}$$

$$\nabla \times \left(\boldsymbol{H}\right) = \boldsymbol{j} + \boldsymbol{j}_{ext} \qquad \text{(EMHD.15)}$$

The following relations take place in isotropic medium:

$$\boldsymbol{B} = \mu_a \cdot \boldsymbol{H} \quad \text{(EMHD.16)}$$

$$\boldsymbol{D} = \varepsilon_a \cdot \boldsymbol{E} \quad \text{(EMHD.17)}$$

The source term in the energy equation (the Joule heat) is:

$$Q_{Joule} = (\boldsymbol{j} + \boldsymbol{j}_{ext})^2 / \sigma \quad \text{(EMHD.18)}$$

The density of the force acting on the liquid from the electromagnetic field in the momentum equations is:

$$\boldsymbol{F}_L = \boldsymbol{j} \times (\boldsymbol{B} + \boldsymbol{B}_{ext}) + q\boldsymbol{E} - \frac{1}{8\pi} E^2 \cdot \nabla \varepsilon \quad \text{(EMHD.19)}$$

Deferred correction method

In the case of poor convergence of the equation for electric potential it is recommended to apply the *deferred correction method*.

When the method is applied, the program, instead of equation

$$(\nabla \sigma \nabla \varphi)^{n+1} = 0 \quad \text{(EMHD.20)}$$

solves the following equation:

$$(\nabla (\sigma + \omega) \nabla \varphi)^{n+1} = (\nabla \omega \nabla \varphi)^n \quad \text{(EMHD.21)}$$

where ω is additional electrical conductivity, which is defined as $\omega = \omega' \cdot deffactor$.

Here:

$deffactor$	is the correction factor (a positive value). The correction factor $deffactor$ is specified in the program's user interface by **MHD parameters > Correction factor** parameter in the Advanced settings of Solver. If you set **Correction factor = 0**, the deferred correction method is not applied.
$\omega' = e^{0.5(\lg \sigma_{max} + \lg \sigma_{min})}$	where σ_{max} and σ_{min} are maximal and minimal conductivities in the problem setting at the initial value of temperature.

The deferred correction method is available for both **Electrodynamics** and **MHD Potential model** models.

10.5.4.4 Boundary conditions

The current section describes boundary conditions for **Electrical potential** and **Magnet Potential** corresponding to different boundary templates.

This section has individual numeration of equations.

10.5.4.4.1 Template "Wall"

For the variable "Electrical potential"

Methods of specifying the scalar **Electrical potential** (φ) of the electric field are described below.

Method "Value of el.potential"

The electrical potential on the boundary is set by the user:

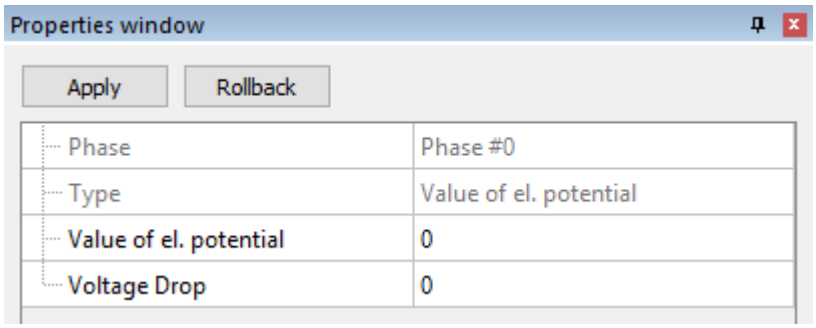
$$\varphi_b = \varphi_{ext} \quad \text{(EMHD-BC.1)}$$

Also it is possible to set voltage drop on the boundary:

$$\varphi_b = \varphi_{ext} - U \quad \text{(EMHD-BC.1bis)}$$

Here U is the voltage drop that can be set by a value, a formula or a table as a function of **Normal current at the boundary** (**CURRENT_NORM**).

Values φ_{ext} and U are specified by parameters **Value of el.potential** and **Voltage Drop** in properties of the **Electrical potential** variable:



Method "Zero gradient"

User specifies nothing. The **Zero gradient** method assumes:

$$\left.\frac{\partial\varphi}{\partial y}\right|_b = 0 \qquad \text{(EMHD-BC.2)}$$

It follows from Eq. (EMHD-BC.2) that the value of electrostatic potential at the boundary equals to its value at the center of the adjacent cell:

$$\varphi_b = \varphi_{cell} \qquad \text{(EMHD-BC.3)}$$

Method "Current density"

The **Current density** method of specifying the scalar potential of the electric field means that surface electric current density j is specified on the boundary.

$$\left(\frac{\partial\varphi}{\partial n}\right)_b = \frac{j}{\sigma} \qquad \text{(EMHD-BC.4)}$$

For the variable "Magnet Potential"

Methods of specifying the vector potential of the magnetic field (**Magnet Potential** $\boldsymbol{A}$) are described below.

Method "Value"

Potential of the magnetic field on the boundary is set by the user:

$$\boldsymbol{A}_b = \boldsymbol{A}_{ext} \qquad \text{(EMHD-BC.5)}$$

Method "Zero gradient"

User specifies nothing. The **Zero gradient** method assumes:

$$\frac{\partial\boldsymbol{A}}{\partial n} = (\nabla\cdot\boldsymbol{A})_b\cdot\boldsymbol{n} = 0 \qquad \text{(EMHD-BC.6)}$$

10.5.4.4.2 Template "Symmetry"

For the variable "Electrical potential"

Method "Zero gradient"

See Eqs. (EMHD-BC.2), (EMHD-BC.3).

For the variable "Magnet Potential"

Method "Symmetry"

User specifies nothing. The **Symmetry** method assumes:

$$\boldsymbol{A}_b = \boldsymbol{A}_{cell} - \left(\boldsymbol{A}_{cell} \cdot \boldsymbol{n}\right) \cdot \boldsymbol{n} \qquad \text{(EMHD-BC.7)}$$

The **Magnet Potential** vector is tangent to the surface of the boundary condition.

10.5.4.4.3 Template "Inlet/Outlet"

For variables "Electrical potential" and "Magnet Potential"

Method "Value"

See Eqs. (EMHD-BC.1), (EMHD-BC.1bis), (EMHD-BC.5).

Method "Zero gradient"

See Eqs. (EMHD-BC.3), (EMHD-BC.6).

10.5.4.4.4 Template "Free outlet"

For variables "Electrical potential" and "Magnet Potential"

Method "Value"

See Eqs. (EMHD-BC.1), (EMHD-BC.1bis), (EMHD-BC.5).

Method "Zero gradient"

See Eqs. (EMHD-BC.3), (EMHD-BC.6).

10.5.4.4.5 Template "Connected"

For the variable "Electrical potential"

Method "Conjugation for potential"

This boundary condition provides continuity of the electric current on the boundary of the subregion:

$$\sigma_2 \frac{\partial \varphi_2}{\partial n_2} = \sigma_1 \frac{\partial \varphi_1}{\partial n_1} \qquad \text{(EMHD-BC.8)}$$

$$\boldsymbol{E}_{\tau 2} = \boldsymbol{E}_{\tau 1} \qquad \text{(EMHD-BC.9)}$$

$$\boldsymbol{B}_{n2} = \boldsymbol{B}_{n1} \qquad \text{(EMHD-BC.10)}$$

$$\boldsymbol{H}_{\tau 2} = \boldsymbol{H}_{\tau 1} \qquad \text{(EMHD-BC.11)}$$

The **Voltage Drop** model can be applied for the **Electrical potential** variable.

The Voltage Drop model

In properties of the element **Boundary conditions > Connected #0 > Electrical potential(*Phase name*)** you can specify the **Voltage Drop** on the inter-phase boundary. The **Voltage Drop** is specified in [V] as a constant, as a formula, or as a table value that depends on **Normal current at the boundary** (**CURRENT_NORM**).

If the voltage drop is simulated from both sides of the inter-phase boundary, specify it in both connected boundary conditions that are set in appropriate **Subregions**.

On inter-phase surfaces in plasma, in the narrow area near electrodes, the voltage drop occurs when the electric current is flowing. The value of the voltage drop depends on materials of the electrodes and on density of the current that is flowing through the boundary. The **Voltage Drop** is specified on the boundary condition by dependency on the normal component of the current density on the boundary.

From the continuity of the normal component of the electric current on the connected boundary follows:

$$j_n = \sigma_1 \frac{\varphi_w - U_1 - \varphi_1}{y_1} = \sigma_2 \frac{\varphi_2 - \varphi_w - U_2}{y_2} \quad \text{(EMHD-BC.12)}$$

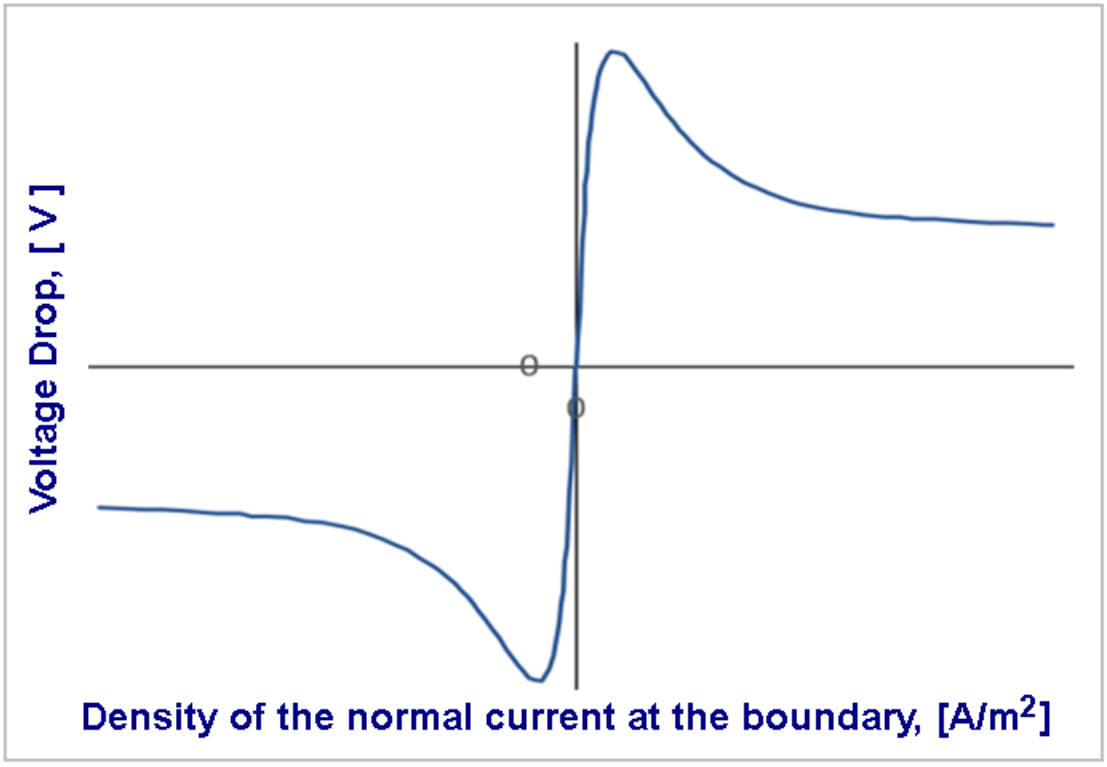
U_1 and U_2 are voltage drop on appropriate sides of the boundary.

The resistance R_i (i=1,2) is determined so that the interfacial voltage drop at the boundary is as follows:

$$U_i = R_i \cdot j_n \quad \text{(EMHD-BC.13)}$$

Then

$$\frac{\sigma_1}{1+\frac{\sigma_1 \cdot R_1}{y_1}} \frac{\varphi_w - \varphi_1}{y_1} = \frac{\sigma_2}{1+\frac{\sigma_2 \cdot R_2}{y_2}} \frac{\varphi_2 - \varphi_w}{y_2} \quad \text{(EMHD-BC.14)}$$



Example of specifying the voltage drop as a function of the normal current the at the boundary

For the variable "Magnet Potential"

Method "Conjugate for Maxwell"

According to ref. [1], the following relations take place on the **Conjugate for Maxwell** boundary condition:

$$\begin{cases} B_{n1} = B_{n2} \\ \frac{B_{2\tau}}{\mu_2 \cdot \mu_0} - \frac{B_{1\tau}}{\mu_1 \cdot \mu_0} = 0 \end{cases} \quad \text{(EMHD-BC.15)}$$

$$\boldsymbol{B} = \nabla \times \boldsymbol{A} \qquad \text{(EMHD-BC.16)}$$

10.5.4.4.6 Template "Nonreflecting"

For variables "Electrical potential" and "Magnet Potential"

Method "Value"

See Eqs. (EMHD-BC.1), (EMHD-BC.1bis), (EMHD-BC.5).

Method "Zero gradient"

See Eqs. (EMHD-BC.3), (EMHD-BC.6).

10.5.4.5 References

1. Landau L.D., Lifshits E.M. (1982) "Theoretical physics. Volume VIII. Electro-dynamics of continuous media", *Moscow, Nauka: The main publishing house of physical and mathematical literature*, 620 p.

10.5.5 Mass transfer

The following selection is present in position **Mass transfer** in window **Preprocessor > Phases > Phase #i > Physical processes**:

- **(none)**
- **Mixing**
- **Chemistry**
- **Combustion**
- **Coal**

The selection becomes accessible after a second **Substance** is collected to the **Phase**.

Below the model describing the flow of a multi-component liquid or gas accompanied by chemical reactions between the components and the implemented gas combustion models are described.

This section has individual numeration of references.

10.5.5.1 Mixing

To activate model **Mixing+Chemistry**, select **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer** = **Mixing**. The **Phase** must contain at least two **Substances**.

User can simulate both simple mixing of the components (**Substances**) and mixing accompanied by chemical reactions between the components. The number of **Substances** is not limited from above. Reactions can be irreversible and reversible.

The given model is compatible with all the turbulence models.

This section has individual numeration of equations.

10.5.5.1.1 Notations

Notation	Physical quantity	Name in *FlowVision*	Dimension
$N_A = 6.022 \cdot 10^{23}$	Avogadro number		$mole^{-1}$
$C_p = \sum_{i=1}^{N} C_{pi} Y_i$	specific heat capacity of mixture	**Specific heat**	$J\ kg^{-1}\ K^{-1} = m^2\ s^{-2}\ K^{-1}$
C_{pi}	specific heat capacity of component i		$J\ kg^{-1}\ K^{-1} = m^2\ s^{-2}\ K^{-1}$
D_i	effective molecular diffusion coefficient for component i		$m^2\ s^{-1}$
$D_{W,i}$	coefficient at Y_i in the source term of Eq. (1)	**D**	$kg\ m^{-3}\ s^{-1}$

Notation	Physical quantity	Name in *FlowVision*	Dimension
$\hat{\boldsymbol{e}}$	strain rate tensor		s^{-1}
$\boldsymbol{F}$	acceleration which determines external volume force		$m\ s^{-2}$
$F_{W,i}$	free term in Eq. (1)	**F**	$kg\ m^{-3}\ s^{-1}$
$\boldsymbol{J}_i$	diffusive flux of component i (vector)		$kg\ m^{-2}\ s^{-1}$
$\boldsymbol{J}_q$	specific heat flux (vector)	**HeatFlux**	$W{\cdot}m^{-2} = kg\ s^{-3}$
$H = h + \frac{1}{2}\boldsymbol{V}^2$	total enthalpy of mixture	**EnthalpyTotal**	$m^2\ s^{-2}$
$h = \sum_{i=1}^{N} h_i Y_i$	thermodynamic enthalpy of mixture	**Enthalpy**	$m^2\ s^{-2}$
h_i	thermodynamic enthalpy of component i		$m^2\ s^{-2}$
$Le_i = \frac{Pr}{Sc_i}$	molecular Lewis number for component i		
$Le_t = \frac{Pr_t}{Sc_t}$	turbulent Lewis number		
m	molar mass of mixture	**Molar mass**	$kg\ mole^{-1}$
m_i	molar mass of component i		$kg\ mole^{-1}$
P	pressure	**Pressure**	$Pa = kg\ m^{-1}\ s^{-2}$
$Pr = \frac{\mu C_p}{\lambda}$	molecular Prandtl number	**Prandtl**	
Pr_t	turbulent Prandtl number	**PrandtlTurb**	
$Sc = \frac{\mu}{\rho D}$	molecular Schmidt number	**Schmidt**	
$Sc_i = \frac{\mu}{\rho D_i}$	molecular Schmidt number for component i		
Sc_t	turbulent Schmidt number	**SchmidtTurb**	
Q_{rad}	volume source of energy due to radiation		$W{\cdot}m^{-3} = kg\ m^{-1}\ s^{-3}$
T_{abs}	absolute temperature		K
$T = T_{abs} - T_{ref}$	relative temperature	**Temperature**	K
T_{ref}	reference temperature	**Temperature**	K
t	time		s
$\boldsymbol{V}$	velocity	**Velocity**	$m\ s^{-1}$
$X_i = m\frac{Y_i}{m_i}$	molar fraction of **Substance** i (volume fraction in the case of gas mixture)	**Mol. fraction_Substance #i**	
Y_i	mass fraction of **Substance** i	**Mass. fraction_Substance #i**	
λ	coefficient of thermal conductivity (molecular)	**Thermal conductivity**	$kg\ m\ s^{-3}\ K^{-1}$

Notation	Physical quantity	Name in *FlowVision*	Dimension
μ	dynamic coefficient of viscosity (molecular)	**Viscosity**	Pa s = kg m^{-1} s^{-1}
μ_t	dynamic coefficient of viscosity (turbulent)	**ViscosityTurb**	Pa s = kg m^{-1} s^{-1}
ρ	density	**Density**	ku m^{-3}
$\hat{\tau}$	shear stress tensor		Pa = kg m^{-1} s^{-2}

Indices:

b - value at boundary
cell - value in cell center
eff - effective value (molecular + turbulent)
n - value at time layer n (in numerical method)
n+1 - value at time layer n+1 (in numerical method)
x - tangent (to boundary) component of vector
y - normal (to boundary) component of vector
w - value at wall

The other notations see in section **Basic notations**.

10.5.5.1.2 Parameters

Parameters in window **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer**:

Parameter	Permissible values	Description
Math. model	• **Mixing** • **Chemistry** • **Combustion** • **Coal**	A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.)
Time step coefficient	An arbitrary numerical value determined by the problem (the default value is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different time steps for different processes sometimes accelerates convergence to the steady-state solution. Meaning of different values: **1** = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the problem as a whole). **> 1** = The calculations of the given process are accelerated. **(0, 1)** = The	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the whole project (τ). *Thus:* Own time step of a physical process = τ × **Time step coefficient** Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time.

Parameter	Permissible values	Description
	calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations / initialization).	
Schmidt	Determined by the problem. (Default value is **1**)	Molecular Schmidt number [1)]
Explicit scheme	• **Yes** • **No**	**Yes** = The mass transfer equations are solved using **Explicit** computational scheme. It is of interest for simulation of spreading radioactive isotopes.

[1)] In the given implementation, user specifies one molecular Schmidt number for all the components. This exactly corresponds to the assumption about Fick's law (2) for diffusion of all the components.

Folder **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer** contains N-1 components (**Substances**), for which convection-diffusion equations are solved. The equation is not solved for the last **Substance**. Its mass fraction is found from Eq. (MT-Mixing.5). In collecting **Substances** into the **Phase**, it is recommended to put the most abundant **Substance** after the others.

Parameters in properties of **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Substances > Substance #i**:

Parameter	Permissible values	Default value	Description
D	Determined by the problem.	**0**	Coefficient $D_{W,i}$ in Eq. (MT-Mixing.3) (is specified by formula in general case).
F	Determined by the problem.	**0**	Free term $F_{W,i}$ in Eq. (MT-Mixing.3) (is specified by formula in general case).

10.5.5.1.3 Equations

Mass transfer equation for component i:

$$\frac{\partial(\rho Y_i)}{\partial t}+\nabla\cdot(\rho Y_i\,\mathbf{V})+\nabla\cdot J_{i,eff}=Q_i \qquad \text{(MT-Mixing.1)}$$

$$\boldsymbol{J}_i=-\left(\rho D_i+\frac{\mu_t}{Sc_{i,t}}\right)\nabla Y_i=-\left(\frac{\mu}{Sc_i}+\frac{\mu_t}{Sc_{i,t}}\right)\nabla Y_i \qquad \text{(MT-Mixing.2)}$$

$$Q_i=D_i(T,P,Y_1,...,Y_N)\cdot Y_i+F_i(T,P,Y_1,...,Y_N) \qquad \text{(MT-Mixing.3)}$$

Expression for heat flux:

$$\boldsymbol{J}_q = -\left(\lambda + \frac{\mu_t C_p}{\mathrm{Pr}_t}\right)\nabla T + \sum_{i=species} h_i \boldsymbol{J}_i = \quad \text{(MT-Mixing.4)}$$

$$= -\left(\frac{\lambda}{C_p} + \frac{\mu_t}{\mathrm{Pr}_t}\right)\nabla h + \sum_{i=species} h_i \nabla Y_i \left(\frac{\lambda}{C_p}\left(1 - Le_{i,mol}\right) + \frac{\mu_t}{\mathrm{Pr}_t}\left(1 - Le_{i,t}\right)\right)$$

$$= -\left(\frac{\lambda}{C_p} + \frac{\mu_t}{\mathrm{Pr}_t}\right)\left(\nabla H - \boldsymbol{V}\cdot\left(\nabla\cdot\boldsymbol{V}\right)\right) + \sum_{i=species} h_i \nabla Y_i \left(\frac{\lambda}{C_p}\left(1 - Le_{i,mol}\right) + \frac{\mu_t}{\mathrm{Pr}_t}\left(1 - Le_{i,t}\right)\right)$$

General relationships:

$$\sum_{i=1}^{N} Y_i = 1 \quad \text{(MT-Mixing.5)}$$

$$\sum_{i=1}^{N} X_i = 1 \quad \text{(MT-Mixing.6)}$$

$$\frac{1}{m} = \sum_{i=1}^{N} \frac{Y_i}{m_i} \quad \text{(MT-Mixing.7)}$$

$$m = \sum_{i=1}^{N} m_i X_i \quad \text{(MT-Mixing.8)}$$

$$h(T) = \sum_{i=1}^{N} h_i(T) Y_i \quad \text{(MT-Mixing.9)}$$

Convection-diffusion equations (MT-Mixing.1) are solved for the first n-1 **Substances** of the given **Phase**. Eq. (MT-Mixing.1) is not solved for the last **Substance**. Its mass fraction is found from Eq. (MT-Mixing.5). In collecting **Substances** into the **Phase**, make the last the most abundant **Substance**. Eq. (MT-Mixing.9) is used for calculation of temperature.

Example 1: Radioactive decay of isotopes

The radioactive decay of an isotope is described by the following equation

$$N(t) = N_0 e^{-t/T} \quad \text{(MT-Mixing.10)}$$

Here

N_0 - initial number of the particles representing the isotope in a close volume.

T - life time of the isotope

As it follows from Eq. (MT-Mixing.10), the molar fraction of the isotope decreases according the following law:

$$X(t) = X_0 e^{-t/T} \quad \text{(MT-Mixing.11)}$$

Here

X_0 - initial molar fraction of the isotope in the volume.

Differentiation of this equation yields the following expression for the rate of changing the isotope molar fraction:

$$\frac{dX}{dt} = -\frac{1}{T} X(t) \quad \text{(MT-Mixing.12)}$$

$$T = \frac{T_{1/2}}{ln\,2} \quad \text{(MT-Mixing.13)}$$

Here

$T_{1/2}$ - half-life of the isotope.

Normally, the concentration of isotopes in the flow of the carrier **Substance** is small. Therefore it is permissible to assume that their properties coincide with the carrier **Substance**. It follows thence that the mass fractions of the isotopes equal to their molar fractions:

$$Y_i = X_i \qquad \text{(MT-Mixing.14)}$$

In such a statement of the problem, "colored" molecules of the carrier **Substance** play the role of the isotopes. Consider a mixture, which consists of liquid sodium (the carrier **Substance**) and for groups of isotopes:

$$Y_{Na} + Y_1 + Y_2 + Y_3 + Y_4 = 1 \qquad \text{(MT-Mixing.15)}$$

The sodium is the last **Substance** in the **Phase**. For the mass (= molar) fractions of the isotopes, the following convection-diffusion equations are solved:

$$\frac{\partial(\rho Y_i)}{\partial t} + \nabla\cdot(\rho Y_i V) - \nabla\cdot\left[\left(\frac{\mu}{Sc} + \frac{\mu_t}{Sc_t}\right)\nabla Y_i\right] = -\frac{1}{T}\rho Y_i, \qquad i = 1 \div 4 \qquad \text{(MT-Mixing.16)}$$

Two constants are specified in window **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Isotope #i**:

$$D_i = -\frac{1}{T_i}\rho \qquad \text{(MT-Mixing.17)}$$

$$F_i = 0 \qquad \text{(MT-Mixing.18)}$$

The sodium mass fraction is found from Eq. (MT-Mixing.15). If distributions of the concentrations (C_i) of the isotopes are required, create user variables

$$C_i = Y_i C_{Na} \quad [\text{cm}^{-3}] \qquad \text{(MT-Mixing.19)}$$

Here

C_{Na} - concentration of the sodium atoms:

$$C_{NA} = \frac{\rho_{NA}\left[kg\,m^{-3}\right]}{m_{NA}\left[kg\,mole^{-1}\right]} A_0\left[mole^{-1}\right] 10^{-6} \quad [\text{cm}^{-3}] \qquad \text{(MT-Mixing.20)}$$

Example 2: Dissociation of oxygen

The **Phase** contains two **Substances**: atomic oxygen O and molecular oxygen O_2. One reversible reaction proceeds between them:

$$O_2 + M \leftrightarrow 2O + M \qquad \text{(MT-Mixing.21)}$$

Here M is a "third body": O_2 or O. The molecular oxygen is the second (last) **Substance** in the **Phase**. Its mass fraction is found from Eq. (MT-Mixing.5).The following convection-diffusion equation is solved for the atomic oxygen:

$$\frac{\partial(\rho Y_O)}{\partial t} + \nabla\cdot(\rho Y_O V) - \nabla\cdot\left[\left(\frac{\mu}{Sc} + \frac{\mu_t}{Sc_t}\right)\nabla Y_O\right] = Q_O \qquad \text{(MT-Mixing.22)}$$

$$Q_O = 2m_O\rho^2 A\left[k_f\frac{Y_{O2}}{m_{O2}} - \rho k_b\left(\frac{Y_O}{m_O}\right)^2\right] \quad \text{(MT-Mixing.23)}$$

Quantity A takes into account the effect of the "third bodies" in dissociation-recombination reaction (MT-Mixing.21):

$$A = 25\frac{Y_O}{m_O} + 9\frac{Y_{O2}}{m_{O2}} \quad \text{(MT-Mixing.24)}$$

The constants of the forward and backward reactions, respectively, are:

$$k_f = 3.6\cdot 10^{12} T_{abs}^{-1} exp\left(-\frac{59500}{T_{abs}}\right) \quad [\text{m}^3\ \text{mole}^{-1}\ \text{s}^{-1}] \quad \text{(MT-Mixing.25)}$$

$$k_b = 3\cdot 10^3 T_{abs}^{-0.5} \quad [\text{m}^6\ \text{mole}^{-2}\ \text{s}^{-1}] \quad \text{(MT-Mixing.26)}$$

Linearization of source term (MT-Mixing.23) yields:

$$\begin{aligned} Q_O &= \rho^2 A\left[k_f(1-Y_O) - \frac{2}{m_O}\rho k_b (Y_O)^2\right] = \\ &= \rho^2 A\left[k_f(1-Y_O^{n+1}) - \frac{2}{m_O}\rho k_b \left(2Y_O^n Y_O^{n+1} - (Y_O^n)^2\right)\right] = \\ &= -Y_O^{n+1}\rho^2 A\left[k_f + \frac{2}{m_O}\rho k_b 2Y_O^n\right] + \rho^2 A\left[k_f + \frac{2}{m_O}\rho k_b (Y_O^n)^2\right] = D_O Y_O + F_O \end{aligned} \quad \text{(MT-Mixing.27)}$$

Finally we have:

$$\begin{aligned} D_O &= -\rho^2(1562.5\cdot Y_O^n + 281.25\cdot(1-Y_O^n))\times \\ &\times\left[3.6\cdot 10^{12} T_{abs}^{-1} exp\left(-\frac{59500}{T_{abs}}\right) + \rho\cdot 7.5\cdot 10^5 T_{abs}^{-0.5} Y_O^n\right] \end{aligned} \quad \text{(MT-Mixing.28)}$$

$$\begin{aligned} F_O &= \rho^2(1562.5\cdot Y_O^n + 281.25\cdot(1-Y_O^n))\times \\ &\times\left[3.6\cdot 10^{12} T^{-1} exp\left(-\frac{59500}{T_{abs}}\right) + \rho\cdot 3.75\cdot 10^5 T^{-0.5}(Y_O^n)^2\right] \end{aligned} \quad \text{(MT-Mixing.29)}$$

These formulas are specified in window **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > O**:

D:

```
-DENSITY * DENSITY * (1562.5 * MASSFRACTIONS00 + 281.25 * MASSFRACTIONS01) * (3.6e+12 * exp(-
59500/(TEMP+273)) / (TEMP+273) + DENSITY*7.5e+5 * (TEMP+273)^(-0.5) * MASSFRACTIONS00)
```

F:

```
DENSITY * DENSITY * (1562.5*MASSFRACTIONS00 + 281.25*MASSFRACTIONS01) * (3.6e+12*exp(-59500/
(TEMP+273)) / (TEMP+273) + DENSITY*3.75e+5 * (TEMP+273)^(-0.5) *
MASSFRACTIONS00*MASSFRACTIONS00)
```

10.5.5.1.4 Boundary conditions

The current section describes boundary conditions for variables **Mass fraction_Substance #i**, corresponding to different boundary templates.

This section has individual numeration of equations.

10.5.5.1.4.1 Template 'Wall'

Value

User specifies the value of the mass fraction of the i-th component:

$$Y_{i,b} = Y_{user} \qquad \text{(MT-Mixing-BC.1)}$$

Symmetry

User specifies nothing. This boundary condition assumes zero gradient of the mass fraction of the i-th component at the given boundary:

$$\left.\frac{\partial Y_i}{\partial y}\right|_b = 0 \qquad \text{(MT-Mixing-BC.2)}$$

From Eq. (MT-Mixing-BC.2) it follows that the value of the mass fraction at the boundary equals to its value at the center of the adjacent cell:

$$Y_{i,b} = Y_{i,cell} \qquad \text{(MT-Mixing-BC.3)}$$

The convective flux of this component also equals to zero:

$$\rho V_n Y_i|_b = 0 \qquad \text{(MT-Mixing-BC.4)}$$

10.5.5.1.4.2 Template 'Symmetry'

Symmetry

User specifies nothing - see Eqs. (MT-Mixing-BC.2) - (MT-Mixing-BC.4).

10.5.5.1.4.3 Template 'Inlet/Outlet'

Value at the inlet

User specifies the value of the mass fraction of the i-th component - see Eq. (MT-Mixing-BC.1).

Zero gradient

User specifies nothing. This boundary condition assumes zero gradient of the mass fraction of the i-th component at the given boundary - see Eqs. (MT-Mixing-BC.2) - (MT-Mixing-BC.3). The convective flux of this component is not zero in general case. It determines by the fluid velocity at this boundary.

10.5.5.1.4.4 Template 'Free outlet'

Value

User specifies the value of the mass fraction of the i-th component - see Eq. (MT-Mixing-BC.1).

Zero gradient

User specifies nothing. This boundary condition assumes zero gradient of the mass fraction of the i-th component at the given boundary - see Eqs. (MT-Mixing-BC.2) - (MT-Mixing-BC.3). The convective flux of this component is not zero in general case. It is determined by the fluid velocity at this boundary.

10.5.5.1.4.5 Template 'Connected'

If user selects **Connection type** = **Conjugate temperature** in window **Create binder condition**, template **Wall** is automatically set on the given boundary.

If user selects **Connection type** = **Conjugate all variables** in window **Create binder condition**, the condition conjugating all the variables is automatically set on the given boundary (user specifies nothing).

Selection **Connection type** = **Periodic surface | Sliding surface** causes automatic setting of the corresponding condition of matching all the variables on the two surfaces (user specifies nothing).

10.5.5.1.4.6 Template 'Nonreflecting'

Value

User specifies the value of the mass fraction of the i-th component outside the computational domain - see Eq. (MT-Mixing-BC.1).

10.5.5.2 Combustion

To activate model **Combustion**, select **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer** = **Combustion**. The **Phase** must contain at least three **Substances**.

The model assumes that one irreversible brutto reaction proceeds in the **Phase**. It is possible to specify a reaction which yields one product (gas mixture in general case):

$$f + i_1 o \rightarrow i_2 p \qquad \text{(MT-Combustion.1)}$$

or a reaction which yields two products:

$$f + i_1 o \rightarrow i_2 p_1 + i_3 p_2 \qquad \text{(MT-Combustion.2)}$$

User computes stoichiometric coefficients $\mathbf{i_1}$, $\mathbf{i_2}$, $\mathbf{i_3}$ and specify them in window **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer**. Value $\mathbf{i_3}$ = 0 means that reaction (MT-Combustion.1) is specified. In this case

$$i_2 = i_1 + 1 \qquad \text{(MT-Combustion.3)}$$

A non-zero value of parameter $\mathbf{i_3}$ (> 0) means that reaction (MT-Combustion.2) is specified. Equations below assume more general reaction (MT-Combustion.2).

In the field **Combustion model** in the properties of the element **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Combustion model** you select the required combustion model:

- **Zeldovich**
- **Arrhenius**
- **Magnussen***)
- **Arrhenius-Magnussen***)
- **EDC***)

*) Models **Magnussen**, **Arrhenius-Magnussen** and **EDC** are compatible only with k-ε turbulence models (KES, KEAKN, KEFV, KENL) and the SST turbulence model (in the **SST** model ε is calculated from ω).

Combustion models **Magnussen**, **Arrhenius-Magnussen** and **EDC** are to be applied only with an enabled turbulence model (or else the computation will be done based on the **Arrhenius** model only).

Combustion models **Magnussen**, **Arrhenius-Magnussen** and **EDC** are compatible only with k-ε turbulence models (**KES**, **KEAKN**, **KEFV**, **KENL**) and the **SST** turbulence model (in the **SST** model ε is calculated from ω). All other turbulence models cannot be used with these combustion models, or else the results would be substantially incorrect.

This section has individual numeration of equations.

10.5.5.2.1 Notations

Notation	Physical quantity	Name in *FlowVision*	Dimension
f	**Substance** = **Fuel**		
g_f	variance of recovered mass fraction of **Fuel**	**Variance of fuel**	
i_1	stoichiometric coefficient at **Oxidizer**		
i_2	stoichiometric coefficient at **Product 1**		
i_3	stoichiometric coefficient at **Product 2**		
k	turbulent energy	**TurbulentEnergy**	$m^2\ s^{-2}$
o	**Substance** = **Oxidizer**		
p	**Substance** = **Products** (in reaction (1))		
p_1	**Substance** = **Product 1** (in reaction (2))		
p_2	**Substance** = **Product 2** (in reaction (2))		
$T_{Z,abs}$	absolute temperature, corresponding to "Zeldovich composition" for given enthalpy		K
T_{ign}	ignition temperature (absolute)	**T ignition**	K
T_{lean}	lower combustion limit (in terms of the absolute temperature obtained assuming complete consumption of **Fuel**)	**T lean**	K
T_{reach}	upper combustion limit (in terms of the absolute temperature obtained assuming complete consumption of **Oxidizer**)	**T rich**	K
Y_f^*	recovered mass fraction of **Fuel**	**Fuel mass frac.**	
Y_f	true mass fraction of **Fuel**	**Fuel mass frac. true**	
Y_o^*	recovered mass fraction of **Oxidizer**		
Y_o	true mass fraction of **Oxidizer**		
Y_{p1}^*	recovered mass fraction of **Product 1**	**Mass frac. [Product 1]**	
Y_{p1}	true mass fraction of **Product 1**		
Y_{p2}^*	recovered mass fraction of **Product 2**	**Mass frac. [Product 2]**	
Y_{p2}	true mass fraction of **Product 2**		
$\alpha^* = \frac{Y_o^*}{i_1 Y_f^*}$	recovered **Oxidizer** excess factor	**Oxid. excess factor rec.**	
$\alpha = \frac{Y_o}{i_1 Y_f}$	true **Oxidizer** excess factor	**Oxid. excess factor true**	
α_{min}^*	lower combustion limit (in terms of recovered **Oxidizer** excess factor)	**Alpha min.**	
α_{max}^*	upper combustion limit	**Alpha max.**	

Notation	Physical quantity	Name in *FlowVision*	Dimension
	(in terms of recovered **Oxidizer** excess factor)		
γ	parameter which determines the weight of turbulent reaction of combustion in model **Arrhenius-Magnussen**		
ε	rate of dissipation of turbulent energy	**TurbDissipation**	$m^2 s^{-3}$

The other notations see in sections **Mixing > Notations** and **Basic notations**.

10.5.5.2.2 Parameters

Parameters of the element **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer**:

Parameter	Permissible values	Description
Math. model	• **Mixing** • **Chemistry** • **Combustion** • **Coal**	A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.)
Time step coefficient	An arbitrary numerical value determined by the problem (the default value is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different time steps for different processes sometimes accelerates convergence to the steady-state solution. Meaning of different values: **1** = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the problem as a whole). **> 1** = The calculations of the given process are accelerated. **(0, 1)** = The calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations / initialization).	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the whole project (τ). *Thus:* Own time step of a physical process = τ × **Time step coefficient** Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time.
Schmidt	Determined by the problem. (Default value: **1**)	Molecular Schmidt number [1)]
Explicit scheme	• **Yes** • **No** (Default value: **No**)	**Yes** = The mass transfer equations are solved using the **Explicit** computational scheme. Recommended setting: **No**.
Combustion reaction	• **Zeldovich** • **Arrhenius** • **Magnussen** • **Arrhenius-Magnussen**	See **Equations**.

Parameter	Permissible values	Description
	• **EDC**	
i_1	Determined by the problem. (Default value: **17.167**)	See Eqs. (MT-Combustion.1), (MT-Combustion.2).
i_2	Determined by the problem. (Default value: **18.167**)	
i_3	Determined by the problem. (Default value: **0**)	
T ignition	Determined by the problem. (Default value: **0**)	See Eqs. (MT-Combustion.6), (MT-Combustion.7).
Alpha min.	Determined by the problem. (Default value: **0**)	Minimum admissible value of oxidizer-fuel equivalence ratio, see Eq. (MT-Combustion.6).
Alpha max.	Determined by the problem. (Default value: **1.8e+308**)	Maximum admissible value of oxidizer-fuel equivalence ratio, see Eq. (MT-Combustion.6).
New combustion limits	• **No** • **Yes** (Default value: **No**)	See **Equations**.
T lean	Determined by the problem. (Default value: **0**)	Temperature determining the lower combustion limit - see Eq. (MT-Combustion.7). (The position is accessible if **New combustion limits = Yes**)
T rich	Determined by the problem. (Default value: **0**)	Temperature determining the upper combustion limit - see Eq. (MT-Combustion.7). (The position is accessible if **New combustion limits = Yes**)
A	Determined by the problem. (Default value [2]: 10^{10})	Parameter of the kinetic reaction rate constant - see Eq. (MT-Combustion.8).
n	Determined by the problem. (Default value [2]: **0**)	Parameter of the kinetic reaction rate constant - see (MT-Combustion.8).
B	Determined by the problem. (Default value [2]: **18 400**)	Parameter of the kinetic reaction rate constant - see (MT-Combustion.8).
n_f	Determined by the problem. (Default value [2]: **1**)	Power of the fuel molar concentration in kinetic rate of the combustion reaction – see (MT-Combustion.8).
n_o	Determined by the problem. (Default value [2]: **1**)	Power of the oxidizer molar concentration in kinetic rate of the combustion reaction – see (MT-Combustion.8).
C	Determined by the problem. (Default value [3]: **23.6**)	Parameter of the turbulent reaction rate constant - see (MT-Combustion.9).
Fuel	Selected by the user from a drop-down list.	The **Substance** is selected, which is the fuel.
Oxidizer		The **Substance** is selected, which is the oxidizer.
Product-1		The **Substances** are selected, which are combustion products.
Product-2		

[1] In the given implementation, user specifies one molecular Schmidt number for all the components. This exactly corresponds to the assumption about Fick's law (MT-Mixing.2) for diffusion of all the components (see the section Mixing > Equations).

[2] Taken from [1].

[3] Taken from [2].

Folder **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer** contains all N components (**Substances**) of the **Phase**. The first component must be **Fuel**, the second one - **Oxidizer**, the third one - **Products** (if $i_3 = 0$) or **Product 1** (if $i_3 > 0$), the fourth one - **Product 2** (if $i_3 > 0$). Other components (if any) follow **Products** or **Product 2**. Convection-diffusion equations are solved for all the components excluding the second one. The mass fraction of the second component (**Oxidizer**) is found from Eqs. (MT-Combustion.13), (MT-Combustion.15) or (MT-Combustion.13), (MT-Combustion.17).

Parameters of elements **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Substances > Substance #i**:

Parameter	Permissible values	Description
D	Determined by the problem. (Default value: **0**)	Coefficient $D_{W,i}$ in Eq. (MT-Combustion.22) (specified by a formula in general)
F	Determined by the problem. (Default value: **0**)	Free term $F_{W,i}$ in Eq. (MT-Combustion.22) (specified by a formula in general)

10.5.5.2.3 Equations

Convection-diffusion equations, describing mass transfer of components, are solved for all the **Substances** of the **Phase** excluding the second one (**Oxidizer**). The **Oxidizer** mass fraction is found from algebraic relationships given below. Two equations are solved for **Fuel** (the first **Substances** of the **Phase**): homogeneous equation for the recovered mass fraction of **Fuel** and inhomogeneous equation for the true mass fraction of **Fuel**.

$$\frac{\partial(\rho Y_f^*)}{\partial t} + \vec{\nabla}\cdot(\rho Y_f^* V) + \vec{\nabla}\cdot J_{f,eff}^* = 0 \qquad \text{(MT-Combustion. 4)}$$

$$\frac{\partial(\rho Y_f)}{\partial t} + \vec{\nabla}\cdot(\rho Y_f V) + \vec{\nabla}\cdot J_{f,eff} = -Q_f \qquad \text{(MT-Combustion. 5)}$$

If **New combustion limits** = **No**,

$$Q_f = 0 \quad \text{where} \quad T_{abs} < T_{ign} \quad \text{or} \quad \alpha < \alpha_{\min} \quad \text{or} \quad \alpha > \alpha_{\max} \qquad \text{(MT-Combustion. 6)}$$

If **New combustion limits** = **Yes**,

$$Q_f = 0 \quad \text{where} \quad T_{abs} < T_{ign} \quad \text{or} \quad T_{Z,abs} < T_{lean} \quad \text{or} \quad T_{Z,abs} > T_{reach} \qquad \text{(MT-Combustion. 7)}$$

Here

$T_{Z,abs}$ - absolute temperature corresponding to "Zeldovich composition" at the local enthalpy

The method for calculation of the true mass fractions of components assuming the Zeldovich combustion mechanism is described below. The expression for the source term of Eq. (MT-Combustion.5) is determined by the selected combustion model.

Arrhenius:

$$Q_f = Q_{kin} = m_f \cdot k \cdot \left(\rho \frac{Y_f}{m_f}\right)^{n\_f} \left(\rho \frac{Y_o}{m_o}\right)^{n\_o}$$

$$k = A T_{abs}^n \exp\left(-\frac{B}{T_{abs}}\right) \qquad \text{(MT-Combustion. 8)}$$

In many works the reaction rate constant is defined as:

$$k = AT_{abs}^{n} \exp\left(-\frac{E_A}{R_A T_{abs}}\right)$$

So, to specify the reaction rate constant in the user interface of *FlowVision*, it is necessary to calculate the value:

$$B = \frac{E_A}{R_A} \quad [K]$$

In the *SI* system (*International System of Units*) dimension of energy of activation E_A is [J/mole].

The universal gas constant in *SI* is:

$$R_A = 8.31441 \quad \left[\frac{J}{mole \cdot K}\right]$$

In the *CGS* system (the *"centimeter-gram-second" system of units*) dimension of energy of activation E_A is [cal/mole].

The universal gas constant in *CGS* is:

$$R_A = 1.9872 \quad \left[\frac{cal}{mole \cdot K}\right]$$

Please note dimension of the complex:

$$\left[\rho \frac{Y_f}{m_f}\right] = \frac{mole}{m^3}$$

In the literature this complex is referred as *"molar concentration"* and shown in formulae as:

$[CH_4]$, $[O_2]$, $[N_2]$, and so on.

The reaction in the 0-dimensional setting is often written as:

$$\frac{d[CH_4]}{dt} = -k[CH_4]^{n\_f}[O_2]^{n\_o}$$

which means:

$$\frac{d\left(\rho \frac{Y_{CH4}}{m_{CH4}}\right)}{dt} = -k\left(\rho \frac{Y_{CH4}}{m_{CH4}}\right)^{n\_f}\left(\rho \frac{Y_{O2}}{m_{O2}}\right)^{n\_o}$$

On the assumption of that $m_{CH4}=const$, we obtain:

$$\frac{d(\rho Y_{CH4})}{dt} = -m_{CH4}k\left(\rho \frac{Y_{CH4}}{m_{CH4}}\right)^{n\_f}\left(\rho \frac{Y_{O2}}{m_{O2}}\right)^{n\_o}$$

This is format of the mass transfer equation used in *FlowVision*, see equation (MT-Combustion.5). This format also remains in the case when $m_{CH4}\neq const$ (the *"individual equilibrium"* approach). Dimension of the pre-exponential factor A depends on values of parameters n_f and n_o:

$$[A] = [k] = \frac{1}{c}\left(\frac{mole}{m^3}\right)^{1-n\_f-n\_0}$$

When units of the *CGS* system are used in an academic paper, conversion to *SI* is elementary:

$$A_{SI} = A_{CGS}\left(10^6\right)^{-n\_f-n\_o}$$

Magnussen:

$$Q_f = Q_{turb} = C\left(\frac{\mu\varepsilon}{\rho k^2}\right)^{0.25}\rho\frac{\varepsilon}{k}\min\left(Y_f, \frac{Y_o}{i_1}\right)$$ (MT-Combustion. 9)

Arrhenius-Magnussen:

$$\frac{1}{Q_f} = \frac{1-\gamma}{Q_{kin}} + \frac{\gamma}{Q_{turb}}$$ (MT-Combustion. 10)

$$\gamma = min\left(1, \frac{\sqrt{g_f}}{Y_f^*}\right)$$ (MT-Combustion. 11)

The following convection-diffusion equation is solved for dispersion of the recovered mass fraction of **Fuel** g_f within the given model [4]:

$$\frac{\partial g_f}{\partial t} + \nabla\left(\rho V g_f\right) = \frac{1}{\rho}\nabla\left(\mu_t \nabla g_f\right) + 2.8\,\mu_t\left(\nabla g_f\right)^2 - 2\,\rho\frac{\varepsilon}{k}g_f$$ (MT-Combustion. 12)

EDC:

$$Q_f = \rho\frac{2.43\left(\frac{\varepsilon}{\nu}\right)^{0.5}}{1-\left[2.13\left(\frac{\nu\varepsilon}{k^2}\right)^{0.25}\right]^2}\left(Y_f - Y_f^0\right)$$ (MT-Combustion. 13)

This model (Eddy Dissipation Concept) [5] assumes that combustion proceeds in the narrow laminar zones ("fine structures") between turbulent eddies. These structures partly fill a computational cell. The mass fractions of **Fuel** Y_f^0 and **Oxidizer** Y_o^0 in the "fine structures" differ from their volume-average values. They are determined from condition of rates of turbulent diffusion and laminar combustion.

$$\frac{\rho}{m_f}\frac{2.43\left(\frac{\varepsilon}{\nu}\right)^{0.5}}{1-\left[2.13\left(\frac{\nu\varepsilon}{k^2}\right)^{0.25}\right]^2}\left(Y_f - Y_f^0\right) = \frac{\rho}{\rho^0}k\left(\rho\frac{Y_f^0}{m_f}\right)^{n\_f}\left(\rho\frac{Y_o^0}{m_o}\right)^{n\_o}$$

and, in their turn, determine temperature and density of the "fine structures" in the cell.

Zeldovich:

The given model assumes infinitely fast combustion. Eq. (MT-Combustion.5) for the true mass fraction of **Fuel** is not solved within this model. The true mass fractions of **Fuel** and **Oxidizer** are computed as follows:

$$Y_f = \begin{cases} 0 & \text{if } Y_O^* > Y_f^* \cdot i_1 \\ Y_f^* - \frac{Y_O^*}{i_1} & \text{otherwise} \end{cases}$$ (MT-Combustion. 14)

$$Y_O = \begin{cases} Y_O^* - Y_f^* \cdot i_1 & \text{if } Y_O^* > Y_f^* \cdot i_1 \\ 0 & \text{otherwise} \end{cases}$$ (MT-Combustion. 15)

$$Y_o^* = 1 - Y_f^* - Y_{p1}^* - Y_{p2}^* - \sum_{i=p2+1}^{N} Y_i$$ (MT-Combustion. 16)

In models **Arrhenius**, **Magnussen**, **Arrhenius-Magnussen**, **EDC**, the true mass fraction of **Oxidizer** is recovered from:

$$Y_O = \begin{cases} Y_O^* - \Delta f \cdot i_1 & \text{if } Y_O^* > \Delta f \cdot i_1 \\ 0 & \text{otherwise} \end{cases}$$ (MT-Combustion. 17)

In all the models, the true mass fractions of the **Products** are computed as follows:

$$Y_{p1} = Y_{p1}^* + \Delta f \cdot i_2$$
$$Y_{p2} = Y_{p2}^* + \Delta f \cdot i_3$$ (MT-Combustion. 18)

$$\Delta f = Y_f^* - Y_f$$ (MT-Combustion. 19)

Here

Y_{p1}^* and Y_{p2}^* are the recovered mass fractions of **Products**, which are obtained as a result of solving the corresponding homogeneous convection-diffusion equations (it is assumed that **Products** can be supplied to the inlets).

Temperature is found iteratively from the following equation

$$h(T) = \sum_{i=1}^{N} h_i(T) Y_i$$ (MT-Combustion. 20)

$$h_i(T_{abs}) = h_0(298.15) + \int_{298.15}^{Tabs} C_p(T) dT$$ (MT-Combustion. 21)

Heat release in combustion results from decrease of the mass fractions of the reactants and increase of the mass fractions of the products of reaction (MT-Combustion.1) or (MT-Combustion.2).

User can specify an additional source term in the mass transfer equations discussed

$$Q_{i,user} = D_i(T, P, Y_1, ..., Y_N) \cdot Y_i + F_i(T, P, Y_1, ..., Y_N)$$ (MT-Combustion. 22)

similarly to model **Mixing**.

10.5.5.2.4 Boundary conditions

If combustion model **Arrhenius-Magnussen** is selected, boundary conditions for variable **Variance of fuel** must be specified similarly to the boundary conditions for components - see **Mixing > Boundary conditions**. Boundary conditions for the true mass fraction of **Fuel** (variable **Fuel mass frac. true**) are set automatically. They are identical to the conditions for the recovered mass fraction of **Fuel** (variable **Mass fraction_Fuel**).

10.5.5.2.5 Modifyers

Ignition/Extinction

This **Modifier** specifies ignition (very fast combustion) or extinction (infinitely slow combustion) of the given gas mixture in all the cells entirely or partly entering the **Object** on which the **Modifier** was created.

If **Subregions > Subregion #i > Modifiers > Ignition / extinction zone #i > Type = Ignition**, the selected **Combustion model** is replaced by model **Zeldovich** in the region of the **Modifier** activity. In doing so, all the combustion limits are canceled. If **Subregions > Subregion #i > Modifiers > Ignition / extinction zone #i > Type = Extinction**, the reaction rate is put zero in the region of the **Modifier** activity.

10.5.5.3 Chemistry

To activate model **Chemistry**, select **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer** = **Chemistry**. The **Phase** must contain at least two **Substances**.

It is wise to use this model when many Arrhenius reactions are to be taken into account. Each reaction is specified in a separate element of the **Preprocessor** tree. The *FlowVision*'s solver automatically computes the source terms for the mass transfer equations. To augment stability of calculations, a user may define **Elements** (conservative scalars).

This section has individual numeration of equations.

10.5.5.3.1 Notations

Notation	Physical quantity	Name in *FlowVision*	Dimension
D_i	effective molecular diffusion coefficient for **Substance** i		$m^2\ s^{-1}$
e_i	charge of ion i		
$\boldsymbol{J}_i$	diffusive flux of **Substance** i (vector)		$kg\ m^{-2}\ s^{-1}$
$k_{f,i}$	rate constant of forward reaction i		determined by reaction
$k_{r,i}$	rate constant of reverse reaction i		determined by reaction
m	molar mass of mixture	**Molar mass**	$kg\ mole^{-1}$
m_i	molar mass of Substance i		$kg\ mole^{-1}$
$Sc_i = \frac{\mu}{\rho D_i}$	molecular Schmidt number for **Substance** i	**Schmidt**	
Sc_t	turbulent Schmidt number	**SchmidtTurb**	
T_{abs}	absolute temperature		K
$T = T_{abs} - T_{ref}$	relative temperature	**Temperature**	o
T_{ref}	reference temperature	**Temperature**	K
t	time		s
$\boldsymbol{V}$	velocity (vector)	**Velocity**	$m\ s^{-1}$
W_i	rate of reaction i		$mole\ m^{-3}\ s^{-1}$
$X_i = m\frac{Y_i}{m_i}$	molar fraction of **Substance** i	**Mol. fraction [Substance #i]**	
Y_i	mass fraction of **Substance** i	**Mass. fraction [Substance #i]**	
Y_i^*	mass fraction of **Element** i	**Mass. fraction [Element #i]**	
μ	dynamic coefficient of molecular viscosity of the mixture	**Viscosity**	$Pa\ s = kg\ m^{-1}\ s^{-1}$
μ_t	dynamic coefficient of turbulent viscosity of the mixture	**ViscosityTurb**	$Pa\ s = kg\ m^{-1}\ s^{-1}$
ρ	mixture density	**Density**	$kg\ m^{-3}$

Indices:
f - parameter of forward reaction
r - parameter of reverse reaction
eff - effective value (molecular + turbulent)

The other notations see in section **Basic notations**.

10.5.5.3.2 Parameters

Parameters in window **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer**:

Parameter	Permissible values	Description
Math. model	• **Mixing** • **Combustion** • **Chemistry** • **Coal**	A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.)
Time step coefficient	An arbitrary numerical value determined by the problem (the default value is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different time steps for different processes sometimes accelerates convergence to the steady-state solution. Meaning of different values: **1** = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the problem as a whole). **> 1** = The calculations of the given process are accelerated. **(0, 1)** = The calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations /	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the whole project (τ). *Thus:* Own time step of a physical process = τ × **Time step coefficient** Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time.

Parameter	Permissible values	Description
	initialization).	
Explicit scheme	• **Yes** • **No**	**Yes** = The mass transfer equations are solved using the **Explicit** computational scheme. It is of interest for simulation of spreading radioactive isotopes.
Ablation	• **(none)** • **Carcass** • **Chemistry** • **Boiling** • **Sublimation**	Use of the specified ablation model.
Schmidt	N numbers determined by the problem (Default value: **0**)	The molecular Schmidt numbers. In model **Chemistry**, a user specifies N molecular Schmidt numbers for all N **Substances** of the **Phase**.

Folder **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Reactions** contains the Preprocessor tree elements **Reaction #0**, **Reaction #1**, **Reaction #2**, ... The number of these elements is not limited. One chemical reaction is specified in on such element. A new reaction is created by command **Create**. An existing reaction is deleted by command **Delete**.

Parameters in window **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Reactions > Reaction #i**:

Parameter	Permissible values	Default value	Description
Name	Numerical value determined by the problem	**Reaction #i**	Name of reaction
Af	Numerical value determined by the problem	**0**	Parameter of the forward reaction rate [1)]
nf	Numerical value determined by the problem	**0**	Parameter of the forward reaction rate [1)]
Tf	Numerical value determined by the problem	**0**	Parameter of the forward reaction rate [1)]
Ar	Numerical value determined by the problem	**0**	Parameter of the reverse reaction rate [2)]
nr	Numerical value determined by the problem	**0**	Parameter of the reverse reaction rate [2)]
Tr	Numerical value determined by the problem	**0**	Parameter of the reverse reaction rate [2)]
Stoichiometric coeffs. > *(Substance)* > Real	Numerical value determined by the problem	**0**	Real (as in the molar equation of the reaction) and effective stoichiometric coefficients of **Substances**. These values determine the molar formula of the **Reaction**. Real stoichiometric coefficients in an elementary reaction determine balance of atoms. These coefficients are used for assembling source terms in equations for mass fractions of **Substances** in the **Phase**. Also these values determine

Parameter	Permissible values	Default value	Description
Stoichiometric coeffs. > *(Substance)* > Effective	Numerical value determined by the problem	**0**	powers of mass fractions of **Substances** in expressions for reaction rates. When elementary chemical reactions are simulated, you can specify only real stoichiometric coefficients (the effective stoichiometric coefficients coefficients are to be set the same). But often, to save computational resources, artificial brutto reactions are simulated with fractional powers of mass fractions of reacting **Substances** in expressions for reaction rates. In such cases you have to specify *real* stoichiometric coefficients for correct automatic assembling source terms the source terms and *effective* stoichiometric coefficients (that are be different from the real ones) according to empiric expressions for reaction rates. When real and effective stoichiometric coefficients of a reaction specified in the *FlowVision*'s user interface are the same, this reaction is elementary.
Efficiencies	N numbers determined by the problem	**0**	Efficiencies of "third bodies". These are coefficients, which determine efficiencies in the dissociation-recombination reactions of those **Substances** of the **Phase** that are "third bodies". Reaction of dissociation (dissipation of a molecule to atoms or smaller molecules), to run, requires that the initial molecule has to be impacted by another particle (a molecule or an atom). During the impact the impacting particle remain unaffected, it is only transfer its kinetic energy to the initial molecule (this energy is required to break interatomic bindings). Recombination also requires some "third" particle, which takes the energy that releases during the recombination. In this case the "third" particle also doesn't change, only its kinetic energy changes (increases). Rate of the dissociation/recombination reaction might substantially depend on which particle is the "third body". Efficiencies of "third bodies" can be found in academic literature, in reference tables that relate to the kinetic gas theory.

1) $k_{f,i} = A_{f,i} T_{abs}^{nf,i} \exp\left[-T_{f,i} / T_{abs}\right]$

2) $k_{r,i} = A_{r,i} T_{abs}^{nr,i} \exp\left[-T_{r,i} / T_{abs}\right]$

Folder **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Elements** contains the **Preprocessor** tree elements **Element #0**, **Element #1**, **Element #2**, ... The number of these elements is strictly

less than the number of the **Phase Substances**. It is determined by a user. The elements are created and deleted by command **Add/Remove**. Such an element is designed for specifying the coefficients which determine the weight of every **Substance** in the given conservative scalar.

Elements are not mandatory attributed of the **Chemistry** model. It is recommended to specify them to improve stability of integrating the mass transfer equations by time. In the section "Equations" we provide an examples of equations for mass fractions of **Elements** (conservative scalars) and calculating the stoichiometric coefficients, which are required to specify **Elements**.

Parameters in window **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Elements > Element #i**:

Parameter	Permissible values	Default value	Description
Substance	Name		Line with the name of the **Substance** selected from the **Substance** list. (The position is inaccessible for editing.)
Coefficients	N numbers determined by the problem	**0**	Stoichiometric coefficients that specify molar formulae of reactions. For **Substances** from left-hand sides of the chemical equations, stoichiometric coefficients are set with the minus sign (i.e. multiplied by **-1**). For **Substances** from right-hand sides of the chemical equations, positive stoichiometric coefficients are set.

10.5.5.3.3 Equations

Mass transfer model **Chemistry** assumes that a **Phase** contains several **Substances**. Several chemical reactions proceed between these **Substances**. The number of **Substances** is not limited. The number of reactions does not depend on the number of **Substances**. Reversible and irreversible reactions can be specified in the interface. Part of Substances may not participate in the reactions. Consider a **Phase** containing seven gaseous **Substances**: O2, N2, NO, O, N, NO+, E. Here E is an electron gas Six reversible reactions proceed between these **Substances**:

$$O2 + M \leftrightarrow 2O + M \quad \text{(MT-Chemistry.1)}$$

$$N2 + M \leftrightarrow 2N + M \quad \text{(MT-Chemistry.2)}$$

$$NO + M \leftrightarrow N + O + M \quad \text{(MT-Chemistry.3)}$$

$$O + N2 \leftrightarrow NO + N \quad \text{(MT-Chemistry.4)}$$

$$NO + O \leftrightarrow N + O2 \quad \text{(MT-Chemistry.5)}$$

$$O + N \leftrightarrow NO^{+} + E \quad \text{(MT-Chemistry.6)}$$

The following equations are integrated:

$$\frac{\partial(\rho Y_{O2})}{\partial t} + \nabla\cdot(\rho Y_{O2}\boldsymbol{V}) + \nabla\cdot J_{O2,eff} = m_{O2}(-W_1 + W_5) \quad \text{(MT-Chemistry.7)}$$

$$\frac{\partial(\rho Y_{N2})}{\partial t} + \nabla\cdot(\rho Y_{N2}\boldsymbol{V}) + \nabla\cdot J_{N2,eff} = m_{N2}(-W_2 - W_4) \quad \text{(MT-Chemistry.8)}$$

$$\frac{\partial(\rho Y_{NO})}{\partial t} + \nabla\cdot(\rho Y_{NO}\boldsymbol{V}) + \nabla\cdot J_{NO,eff} = m_{NO}(-W_3 + W_4 - W_5) \quad \text{(MT-Chemistry.9)}$$

$$\frac{\partial(\rho Y_{O})}{\partial t} + \nabla\cdot(\rho Y_{O}\boldsymbol{V}) + \nabla\cdot J_{O,eff} = m_{O}(2W_1 + W_3 - W_4 - W_5 - W_6) \quad \text{(MT-Chemistry.10)}$$

$$\frac{\partial(\rho Y_{N})}{\partial t} + \nabla\cdot(\rho Y_{N}\boldsymbol{V}) + \nabla\cdot J_{N,eff} = m_{N}(2W_2 + W_3 + W_4 + W_5 - W_6) \quad \text{(MT-Chemistry.11)}$$

$$\frac{\partial(\rho Y_{NO+})}{\partial t}+\nabla\cdot(\rho Y_{NO+}\boldsymbol{V})+\nabla\cdot J_{NO+,eff}=m_{NO+}W_6 \quad \text{(MT-Chemistry.12)}$$

$$\frac{\partial(\rho Y_E)}{\partial t}+\nabla\cdot(\rho Y_E\boldsymbol{V})+\nabla\cdot J_{E,eff}=m_E W_6 \quad \text{(MT-Chemistry.13)}$$

The reaction rates are determined by the following expressions:

$$W_1=M_1\left(k_{f,1}\rho\frac{Y_{O2}}{m_{O2}}-k_{r,1}\left(\rho\frac{Y_O}{m_O}\right)^2\right) \quad \text{(MT-Chemistry.14)}$$

$$W_2=M_2\left(k_{f,2}\rho\frac{Y_{N2}}{m_{N2}}-k_{r,2}\left(\rho\frac{Y_N}{m_N}\right)^2\right) \quad \text{(MT-Chemistry.15)}$$

$$W_3=M_3\left(k_{f,3}\rho\frac{Y_{NO}}{m_{NO}}-k_{r,3}\rho\frac{Y_O}{m_O}\rho\frac{Y_N}{m_N}\right) \quad \text{(MT-Chemistry.16)}$$

$$W_4=\left(k_{f,4}\rho\frac{Y_O}{m_O}\rho\frac{Y_{N2}}{m_{N2}}-k_{r,4}\rho\frac{Y_{NO}}{m_{NO}}\rho\frac{Y_N}{m_N}\right) \quad \text{(MT-Chemistry.17)}$$

$$W_5=\left(k_{f,5}\rho\frac{Y_{NO}}{m_{NO}}\rho\frac{Y_O}{m_O}-k_{r,5}\rho\frac{Y_N}{m_N}\rho\frac{Y_{O2}}{m_{O2}}\right) \quad \text{(MT-Chemistry.18)}$$

$$W_6=\left(k_{f,6}\rho\frac{Y_O}{m_O}\rho\frac{Y_N}{m_N}-k_{r,6}\rho\frac{Y_{NO+}}{m_{NO+}}\rho\frac{Y_E}{m_E}\right) \quad \text{(MT-Chemistry.19)}$$

Here

$k_{f,i}$ - is the constant of the rate of forward reaction i,

$k_{r,i}$ - is the constant of the rate of reverse reaction i.

The Arrhenius law is assumed for the reaction rates. Therefore

$$k_{f,i}=A_{f,i}T_{abs}^{nf,i}\,exp\left[-\frac{T_{f,i}}{T_{abs}}\right] \quad \text{(MT-Chemistry.20)}$$

$$k_{r,i}=A_{r,i}T_{abs}^{nr,i}\,exp\left[-\frac{T_{r,i}}{T_{abs}}\right] \quad \text{(MT-Chemistry.21)}$$

Quantities $A_{f,i}$, $n_{f,i}$, $T_{f,i}$, $A_{r,i}$, $n_{r,i}$, $T_{r,i}$ are specified in the interface. Coefficient M_i, present in the expression for dissociation-recombination reaction i ($i=1,2,3$), accounts for the dependency of the reaction rate on the "third" particle M:

$$M_i=w_{O2,i}\rho\frac{Y_{O2}}{m_{O2}}+w_{N2,i}\rho\frac{Y_{N2}}{m_{N2}}+w_{NO,i}\rho\frac{Y_{NO}}{m_{NO}}+w_{O,i}\rho\frac{Y_O}{m_O}+w_{N,i}\rho\frac{Y_N}{m_N}+w_{NO+,i}\rho\frac{Y_{NO+}}{m_{NO+}}+w_{E,i}\rho\frac{Y_E}{m_E} \quad \text{(MT-Chemistry.22)}$$

Here $w_{O2,i}$, $w_{N2,i}$, $w_{NO,i}$, $w_{O,i}$, $w_{N,i}$, $w_{NO+,i}$, $w_{E,i}$ are dimensionless coefficients which determine the efficiencies of the "third bodies". These coefficients are specified in the interface.

Implemented model **Chemistry** assumes that the relationship between diffusion flux i and the gradient of mass fraction i is determined by the "individual" Fick law:

$$\boldsymbol{J_i} = -\left(\rho D_i + \frac{\mu_t}{Sc_{i,t}}\right)\nabla Y_i = -\left(\frac{\mu}{Sc_i} + \frac{\mu_t}{Sc_t}\right)\nabla Y_i \qquad \text{(MT-Chemistry.23)}$$

The developed algorithm affords a user an opportunity to select elements (conservative scalars) from the list of Substances. The mass fraction of element *j* is defined by

$$Y_j^* = Y_j + \sum_{k=1}^{Np} \nu_{kj} \frac{m_j}{m_k} Y_k \qquad \text{(MT-Chemistry.24)}$$

Here

N_p - number of Substances regarded as products of chemical reactions,

ν_{kj} - stoichiometric coefficients.

The developed computational algorithm allows the user to select **Substances-Elements** (conservative scalars). This improves stability of time integrating the mass transfer equations.

Select the minimum essential number of **Substances** that provide ability to formally derive from them all remaining **Substances** taking part in reactions (as a general rule, it is recommended to select the most stable substance existing in nature. The selected **Substances** are **Elements** (conservative scalars). **Elements** are selected from the list of **Substances** that take part in reactions of the concerned kinetic scheme. Number of **Elements** is to be strictly less then number of **Substances** taking part in reactions.

For **Elements** (conservative scalars) the following condition is fulfilled: (MT-Chemistry.25)

$$\sum_{j=Np+1}^{N} Y_j^* = 1$$

In the considered example, the **Elements** are **Substances O2**, **N2**, and **E**. The other **Substances** are regarded as products. The products are unambiguously expressed via **Elements**. We present below formal equations (*not* chemical reactions!), in which products are derived from **Elements** using stoichiometric coefficients:

$$O = \frac{1}{2} O_2 \qquad \text{(MT-Chemistry.26)}$$

$$N = \frac{1}{2} N_2 \qquad \text{(MT-Chemistry.27)}$$

$$NO = \frac{1}{2} N_2 + \frac{1}{2} O_2 \qquad \text{(MT-Chemistry.28)}$$

$$NO^+ = \frac{1}{2} N_2 + \frac{1}{2} O_2 - E \qquad \text{(MT-Chemistry.29)}$$

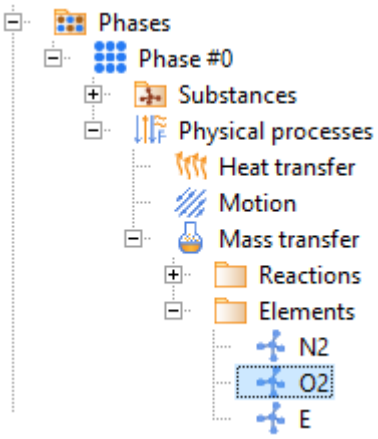
The coefficients at the elements in the right-hand sides of gross-reactions (MT-Chemistry.26) - (MT-Chemistry.29) are stoichiometric coefficients ν_{kj}.

Let's use formula (MT-Chemistry.24) to formulate expressions that determine mass fractions of **Elements**:

$$Y_{O2}^* = Y_{O2} + \frac{1}{2}\frac{32}{16} Y_O + \frac{1}{2}\frac{32}{30} Y_{NO} + \frac{1}{2}\frac{32}{30} Y_{NO+} = Y_{O2} + Y_O + \frac{8}{15} Y_{NO} + \frac{8}{15} Y_{NO+} \qquad \text{(MT-Chemistry.30)}$$

$$Y_{N2}^* = Y_{N2} + \frac{1}{2}\frac{28}{14} Y_N + \frac{1}{2}\frac{28}{30} Y_{NO} + \frac{1}{2}\frac{28}{30} Y_{NO+} = Y_{N2} + Y_N + \frac{7}{15} Y_{NO} + \frac{7}{15} Y_{NO+} \qquad \text{(MT-Chemistry.31)}$$

$$Y_E^* = Y_E - \frac{m_E}{m_{NO+}} Y_{NO+} \qquad \text{(MT-Chemistry.32)}$$



These three equations, for three **Elements** created in the program's interface, specify coefficients for each **Substance**. For example, in propertied of the **Element O2** the set of coefficients will be:

Coefficients > N2 = 0
Coefficients > NO = 0.53333333
Coefficients > O2 = 1
Coefficients > O = 1
Coefficients > N = 0
Coefficients > NO+ = 0.53333333
Coefficients > E = 0

The value **0.53333333** here is approximation of the precise fraction 8/15.

If we sum the equations for **Substances** present in the right-hand sides of Eqs. (MT-Chemistry.30) - (MT-Chemistry.32) with these coefficients, we obtain homogeneous equations (the source terms cancel):

$$\frac{\partial(\rho Y_{O2}^*)}{\partial t} + \nabla \cdot (\rho Y_{O2}^* \boldsymbol{V}) + \nabla \cdot J_{O2,eff}^* = 0 \qquad \text{(MT-Chemistry.33)}$$

$$\frac{\partial(\rho Y_{N2}^*)}{\partial t} + \nabla \cdot (\rho Y_{N2}^* \boldsymbol{V}) + \nabla \cdot J_{N2,eff}^* = 0 \qquad \text{(MT-Chemistry.34)}$$

$$\frac{\partial(\rho Y_E)}{\partial t} + \nabla \cdot (\rho Y_E \boldsymbol{V}) + \nabla \cdot J_{E,eff}^* = 0 \qquad \text{(MT-Chemistry.35)}$$

The electron gas is a special component. In most practical problems, in which electro-magnetic field is absent, plasma is quasi-neutral. In this case

$$Y_E^* = 0 \qquad \text{(MT-Chemistry.36)}$$

For molar fraction of electrons we have:

$$X_E = \sum_{i \neq E} e_i X_i = X_{NO+} \qquad \text{(MT-Chemistry.37}$$

Here

e_i - is the charge of ion i.

Algebraic relationship (MT-Chemistry.36) (or (MT-Chemistry.37)) allows excluding solution of the convection-diffusion equation for electrons from the algorithm. Condition (MT-Chemistry.25) allows excluding solution of one of the equations for elements (in addition to the equation for electrons). When the mass transfer equations are integrated sequentially, calculations with elements are more stable. Five equations are integrated in the considered example: Eqs. (MT-Chemistry.9) - (MT-Chemistry.12) and one of Eqs. (MT-Chemistry.33), (MT-Chemistry.34). True mass fractions of O2 and N2 are found from algebraic relationships (MT-Chemistry.30), (MT-Chemistry.31).

10.5.5.3.4 Boundary conditions

See section **Mass transfer > Mixing > Boundary conditions**.

10.5.5.4 Ablation

Ablation of a body surface is a thermo-chemical process. Therefore, process **Heat transfer** must be active.

Select a required ablation model in box **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Ablation**:

- **(none)**
- **Carcass**
- **Chemistry**
- **Boiling**
- **Sublimation**

Each of these ablation model is compatible with all the mass transfer models (**Mixing**, **Combustion**, **Chemistry**).

This section has individual numeration of equations.

10.5.5.4.1 Notations

Notation	Physical quantity	Name in *FlowVision*	Dimension
$C_p = \sum_{i=1}^{N} C_{pi} Y_i$	specific heat capacity of mixture	**Specific heat**	J kg^{-1} K^{-1} = = m^2 s^{-2} K^{-1}
C_{pi}	specific heat capacity of **Substance** i		J kg^{-1} K^{-1} = = m^2 s^{-2} K^{-1}
D_i	effective molecular diffusion coefficient for **Substance** i		m^2 s^{-1}
$\boldsymbol{J}_i$	diffusive flux of **Substance** i (vector)		kg m^{-2} s^{-1}
$\boldsymbol{J}_q$	specific heat flux (vector)		W·m^{-2} = kg s^{-3}
h_i	thermodynamic enthalpy of **Substance** i		m^2 s^{-2}
$\Delta h(T_w) = h_{vapor}(T_w) - h_{melt}(T_w)$	latent heat of phase transfer in model **Boiling**		m^2 s^{-2}
$h_{vapor}(T_w)$	thermodynamic enthalpy of vapor [1)]		m^2 s^{-2}
$h_{melt}(T_w)$	thermodynamic enthalpy of melt in model **Boiling'**	**h_melt**	m^2 s^{-2}
$k_{f,i}$	rate constant of irreversible reaction i		determined by reaction
M_i	molecular mass of **Substance** i		kg
m	molar mass of mixture	**Molar mass**	kg mole^{-1}
m_i	molar mass of **Substance** i		kg mole^{-1}
$\dot{m}_i$	specific mass flow rate of **Substance** i from the body surface		kg s^{-1} m^{-2}
$\dot{m}$	total specific mass flow rate of **Substance** i from the body's surface	**Mass blow-in from boundary**	kg s^{-1} m^{-2}
N	number of **Substances** in the **Phase**		

Notation	Physical quantity	Name in *FlowVision*	Dimension
n_i	concentration of **Substance** i		m^{-3}
$P_{i,sat}$	saturation pressure of the ablation product (partial pressure of gaseous **Substance** i)	**Saturation pressure**	Pa
P_{abs}	absolute pressure		Pa
$Sc_i = \frac{\mu}{\rho D_i}$	molecular Schmidt number for **Substance** i	**Schmidt**	
Sc_t	turbulent Schmidt number	**SchmidtTurb**	
T_w	absolute temperature of a solid surface (wall)	**Wall temperature**	$m^2\ s^{-2}$
W_i	rate of irreversible reaction i		mole $m^{-2}\ s^{-1}$
$X_{i,w} = m_w \frac{Y_{i,w}}{m_i}$	molar fraction of **Substance** i at the wall		
$Y_{i,c}$	mass fraction of **Substance** i in the cell center	**Mass. fraction [Substance #i]**	
$Y_{i,w}$	mass fraction of **Substance** i at the wall	**Mass. fraction at wall [Substance #i]**	
ε_w	surface emissivity (blackness)	**Blackness**	
λ	coefficient of molecular thermal conductivity of the mixture	**Thermal conductivity**	kg m s^{-3} K^{-1}
μ	dynamic coefficient of molecular viscosity of the mixture	**Viscosity**	Pa s = kg m^{-1} s^{-1}
μ_t	dynamic coefficient of turbulent viscosity of the mixture	**ViscosityTurb**	Pa s = kg m^{-1} s^{-1}
ρ	mixture density	**Density**	kg m^{-3}
$\rho_i = M_i n_i$	density of **Substance** i		kg m^{-3}
$\sigma_{rad} = 5.67 \cdot 10^{-8}$	Stefan-Boltzmann constant	**Stefan–Boltzmann constant**	W m^{-2} K^{-4}

[1)] Ablation product is one of the **Substances** of the gas **Phase**.

Indices:

g - gas **Phase**,
s - solid **Phase**,
n - normal to wall direction,
w - wall.

The other notations see in section **Basic notations**.

10.5.5.4.2 Parameters

After selection of one of the three implemented ablation models (in the **Properties** window of the element **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer**), element **Ablation** appears in the **Preprocessor** tree. Its name depends on the selected ablation model.

Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Ablation: Carcass

Parameter	Permissible values	Default value	Description
blow_s_TPS	Determined by the problem	**0**	Dimensionless blow-in rate of the ablation products of the thermal-protective coating's matrix into boundary layer
h_w_TPS	Determined by the problem	**0**	Thermodynamic enthalpy of the ablation products of the thermal-protective coating's matrix, [m^2/s^2]
Phase	Determined by the problem	**(none)**	Select here the **Phase**, which corresponds to the ablation products of the thermal-protective coating.

Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Ablation: Chemistry

The window parameters are absent. This tree element contains folder **Reactions. Folder Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Ablation: Chemistry > Reactions** contains tree elements **Reaction #0**, **Reaction #1**, **Reaction #2**, ... The number of such elements is not limited. One reaction is specified in one element. A new reaction is created by command **Create**. An existing reaction is deleted by command **Delete**.

Parameters of the element **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Ablation: Chemistry > Reactions > Reaction #i**:

Parameter	Permissible values	Default value	Description
Name	Determined by a user	**Reaction #i**	Name of reaction.
Af	Determined by the problem	**0**	Parameters of the forward reaction rate.[1)]
nf	Determined by the problem	**0**	
Tf	Determined by the problem	**0**	
Reactant	One of the **Substances**	**(none)**	Selection from the list of the **Phase**'s **Substances**.
Reactant stoichiometric coeff.	Determined by the problem	**0**	Stoichiometric coefficient which determine the molar formula of the reaction.
Product	One of the **Substances**	**(none)**	Selection from the list of the **Phase**'s **Substances**.[2)]
Product stoichiometric coeff.	Determined by the problem	**0**	Stoichiometric coefficient which determine the molar formula of the reaction.

1) $k_{f,i} = A_{f,i} T_{abs}^{nf,i} \exp\left[-T_{f,i} / T_{abs}\right]$

2) **Reactant** and **Product** must be different **Substances**.

Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Ablation: Boiling

Parameters of the element **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Ablation: Boiling**:

Parameter	Permissible values	Default value	Description
Ablation product	One of the **Substances**	**(none)**	Selection from the list of the **Phase**'s **Substances**

This model assumes that the temperature of the surface, at which ablation occurs, is specified in the boundary condition for temperature.

Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Ablation: Sublimation

Parameters of the element **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer > Ablation: Sublimation:**

Parameter	Permissible values	Default value	Description
Ablation product	One of the **Substances**	**(none)**	Selection from the list of the **Phase**'s **Substances** [1)]

[1)] Substance property **Saturation pressure** must be specified for the given **Substance**.

10.5.5.4.3 Equations

Now the following ablation models are implemented in *FlowVision*:

- **Carcass**
- **Chemistry**
- **Boiling**
- **Sublimation**

Below we shall assume that thermo-chemical processes proceed at the contact surface between a solid body and a gas **Phase**. The processes result in mass loss. The gaseous ablation products are blown in the gas **Phase**.

Balance of mass

In general case, the mass balance at the contact surface (at wall), at which ablation occurs, can be written as follows:

$$\dot{m} = \sum_{i=1}^{Ng} \dot{m}_{i,g} = \sum_{i=1}^{Ns} \dot{m}_{i,s} \qquad \text{(MT-Ablation.1)}$$

Here

$\dot{m}$ - total specific mass flow rate from the body surface [kg s^{-1} m^{-2}],

$\dot{m}_{i,g}$ - specific mass flow rate of gaseous Substance i from the body surface [kg s^{-1} m^{-2}],

$\dot{m}_{i,s}$ - specific mass flow rate of solid Substance i from the body surface [kg s^{-1} m^{-2}],

Ng - number of **Substances** in the gas **Phase**,

Ns - number of **Substances** in the gas **Phase**.

By definition, the mass flow rate of gaseous **Substance i** from the body surface is

$$\dot{m}_{i,g} = \rho_{i,w} V_{i,w} \qquad \text{(MT-Ablation.2)}$$

Here

$\rho_{i,w} = M_i n_{i,w}$ - density of Substance i at the surface,

M_i - molecular mass of **Substance i** [kg],

$n_{i,w}$ - concentration of **Substance i** at the surface [m^{-3}],

$V_{i,w}$ - normal velocity of **Substance i** at the surface.

The total specific mass flow rate (the mass loss rate) can be expressed as follows:

$$\dot{m} = \sum_{i=1}^{N} \rho_{i,w} V_{i,w} = \rho_w V_w \qquad \text{(MT-Ablation.3)}$$

By definition, the diffusive flux of gaseous **Substance i** from the body surface is

$$J_{i,w} = \rho_{i,w}\left(V_{i,w} - V_w\right) = \rho_{i,w}\left(V_{i,w} - \frac{\dot{m}}{\rho_w}\right) = \dot{m}_{i,g} - \dot{m}\frac{\rho_{i,w}}{\rho_w} \qquad \text{(MT-Ablation.4)}$$

By definition,

$$\frac{\rho_i}{\rho} = Y_i \qquad \text{(MT-Ablation.5)}$$

$$\rho = \sum_{i=1}^{N} \rho_i \qquad \text{(MT-Ablation.6)}$$

Thus, the total specific mass flow rate of gaseous **Substance i** can be divided into convective and diffusive parts:

$$\dot{m}_i = \dot{m} Y_{i,w} + J_{i,w} \qquad \text{(MT-Ablation.7)}$$

Assuming the Fick law for the diffusive flux, we have:

$$J_{i,w} = -\rho D_i \nabla Y_i = \frac{\mu}{Sc_i}\frac{Y_{i,w} - Y_{i,c}}{y} \qquad \text{(MT-Ablation.8)}$$

Here

$Y_{i,w}$ - mass fraction of Substance i at the wall,

$Y_{i,c}$ - mass fraction of Substance i in the center of a cell adjacent to the wall,

y - distance from the wall to the cell center.

In turbulent flow

$$J_{i,w} = \left(\frac{\mu}{Sc_i} + \frac{\mu_t}{Sc_t}\right)\frac{Y_{i,w} - Y_{i,c}}{y} \qquad \text{(MT-Ablation.9)}$$

Balance of energy

In general case, the energy balance at the contact surface, at which ablation occurs, can be written as follows:

$$-\left(\lambda_g + \lambda_{g,t}\right)\nabla_n T + \sum_{i=1}^{Ng} h_{i,g}\left(T_w\right)\dot{m}_i + \sigma_{rad}\varepsilon_w T_w^4 = -\lambda_s \nabla_n T + \sum_{i=1}^{Ns} h_{i,s}\left(T_w\right)\dot{m}_i \qquad \text{(MT-Ablation.10)}$$

Here

λ_g - coefficient of molecular thermal conductivity of gas **Phase**,

$\lambda_{g,t}$ - coefficient of turbulent thermal conductivity of gas **Phase**,

λ_s - coefficient of thermal conductivity of solid **Phase**,

$\nabla_n T$ - normal (to the contact surface) component of the temperature gradient (the normal is directed inside the gas **Phase**),

$h_{i,g}(T_w)$ - thermodynamic enthalpy of gaseous **Substance** i,

$h_{i,s}(T_w)$ - thermodynamic enthalpy of solid **Substance** i,

T_w - absolute temperature of the surface.

Model 'Carcass'

This is a complex model of destruction of porous thermal-protective coating (TPC) and mass loss from its surface. Gases, which are generated during the pyrolysis process, move through pores and are blown into the boundary layer.

You can find details in works [6-13].

Model 'Chemistry'

This model assumes very simple pre-defined format for irreversible reactions proceeding at the contact surface (at a wall):

$$n_{Ai}A_i \rightarrow n_{Bi}B_i \qquad \text{(MT-Ablation.11)}$$

Here

A_i - reactant of reaction i,

$n_{A,i}$ - stoichiometric coefficient of reactant,

B_i - product of reaction i,

$n_{B,i}$ - stoichiometric coefficient of product.

A_i and B_i are different **Substances**. They are selected in the interface from the list of **Phase Substances**. The rate of heterogeneous reaction (MT-Ablation.11) is determined by

$$W_i = k_{f,i}\left(\rho\frac{Y_A}{m_A}\right)^{nAi} \qquad \text{(MT-Ablation.12)}$$

$$k_{f,i} = A_{f,i}T_w^{nf,i}\,exp\left[-\frac{T_{f,i}}{T_w}\right] \qquad \text{(MT-Ablation.13)}$$

Quantities $n_{A,i}$, $n_{B,i}$, $A_{f,i}$, $n_{f,i}$, $T_{f,i}$ are specified in the interface. In the case of one heterogeneous reaction we have:

$$\dot{m}_A = -m_A n_A W \qquad \text{(MT-Ablation.14)}$$

$$\dot{m}_B = m_B n_B W \qquad \text{(MT-Ablation.15)}$$

$$\dot{m} = \dot{m}_A + \dot{m}_B \qquad \text{(MT-Ablation.16)}$$

Model 'Boiling'

This model assumes that the temperature of the contact surface (at a wall) is known. When a non-conjugate problem is solved, the expression for the mass flow rate reads:

$$\dot{m}(T_w) = max\left\{0, \frac{1}{\Delta h(T_w)}\left[(\lambda_g + \lambda_{g,t})\frac{T_c - T_w}{y} - \sigma_{rad}\varepsilon_w T_w^4\right]\right\} \qquad \text{(MT-Ablation.17)}$$

Here

$\Delta h(T_w)$ - latent heat of phase transfer,

T_c - temperature at the center of a cell adjacent to the wall.

When a conjugate problem is solved, the expression for the mass flow rate reads:

$$\dot{m}(T_w) = max\left\{0, \frac{1}{\Delta h(T_w)}\left[(\lambda_g + \lambda_{g,t})\frac{T_{c,g} - T_w}{y_g} + \lambda_s \frac{T_{c,s} - T_w}{y_s} - \sigma_{rad}\varepsilon_w T_w^4\right]\right\} \quad \text{(MT-Ablation.18)}$$

Here

$T_{c,g}$ - temperature at the center of a cell adjacent to the wall at the side of the gas **Phase**,

$T_{c,s}$ - temperature at the center of a cell adjacent to the wall at the side of the solid **Phase**,

y_g - distance from the wall to the cell center at the side of the gas **Phase**,

y_s - distance from the wall to the cell center at the side of the solid **Phase**.

Model 'Sublimation'

The molar fraction of gaseous ablation product A at the contact surface (at a wall) is determined by the partial pressure of saturated vapor A which is a function of the wall temperature:

$$X_{A,w} = \frac{P_{A,sat}(T_w)}{P_{abs}} \quad \text{(MT-Ablation.19)}$$

By definition, the mass fraction of A at the wall is

$$Y_{A,w} = X_{A,w}\frac{m_w}{m_A} \quad \text{(MT-Ablation.20)}$$

Here

m_A - molar mass of A,

m_w - molar mass of the gas mixture at the wall.

The mass flow rate is found from relationship

$$\frac{\mu}{Sc}\frac{Y_{A,w} - Y_{A,c}}{y} = \dot{m}(1 - Y_{A,w}) \quad \text{(MT-Ablation.21)}$$

with additional condition

$$0 \le \dot{m} \le \frac{P_{A,sat}(T_w)}{\sqrt{2\pi\frac{R_A T_w}{m_A}}} \quad \text{(MT-Ablation.22)}$$

10.5.5.4.4 Boundary conditions

In order to simulate ablation at a surface, select template **Wall, ablation** or template **Connected**, **Connection type** = **Ablation**.

10.5.5.4.4.1 Template 'Wall, ablation'

In this template, boundary conditions **Ablation** are automatically set up for temperature, velocity, and mass fractions of Substances. A user specifies only the following parameters in the boundary condition for temperature:

Parameter	Description
Blackness	Emissivity factor (the surface blackness coefficient)
h_solid	Enthalpy of solid body or melt (in a non-conjugate problem)

All the physical quantities which determine the mass loss process at the surface are computed automatically.

10.5.5.4.4.2 Template 'Connected'

If **Connection type** = **Ablation** is selected in creation of a Binder condition, boundary conditions **Ablation** are automatically set up for temperature, velocity, and mass fractions of Substances at the given surface. A user specifies only **Blackness** in the boundary condition for temperature. All the physical quantities which determine the mass loss process at the surface are computed automatically with conjugate heat transfer taken into account.

10.5.5.5 References

1. Volkov E.P., Kudryavtsev N.Yu, (1989) "Simulation of formation of nitric oxides in turbulent diffision flame", *Engineering-physical journal, Vol. 56, No. 6,* pp. 885-894.

2. Magnussen B.F., and Hjertager B.H. (1976) "On Mathematical Modelling of Turbulent Combustion with Special Emphasis on Soot Formation and Combustion," *Sixteenth Symposium (International) on Combustion,* pp. 719-729.

3. Aksenov A.A., Pohilko V.I., Tishin A.P. (1998) "Investigation of two-step methane combustion in a swirl burner", *Proceedings of the 2-nd Russian national conference on heat transfer, Moscow, 26-30 October 1998, Vol.3,* pp.161-164.

4. Gupta A.K., Lilley D.G., and Syred N. (1984) "Swirl Flows," *Abacus Press*.

5. Magel, H.C., Schnell, U. and Hein, K.R.G. (1996) "Simulation of detailed chemistry in a turbulent combustion flow," *Twenty-Sixth Symposium (International) on Combustion / The Combustion Institute,* pp. 67-74.

6. Lachaud J., Martin A., van Eekelen T., Cozmuta I., Ablation test-case series #2 // version 2.8., February 6 2012., 9 p.

7. van Eekelen T., Martin A., Lachaud J., Bianchi D., Test Case Series 3 // Ablation Workshop: Code Comparison. Paper 2., 2014, 20 p.

8. Lachaud J., Eekelen T., Scoggins J.B., Maginc T.E., Mansour N.N. Detailed chemical equilibrium model for porous ablative materials // International Journal of Heat and Mass Transfer, 2015, Vol. 90, pp. 1034-1045.

9. Theoretical Ablative Composite for Open Testing (TACOT), Version 2.2 // 5th Ablation Workshop, Lexington, KY, March 2012.

10. Mansour N.N., Lachaud, J.R., Magin T.E, Muelenaer J.de High-fidelity charring ablator thermal response model // AIAA 2011-3124 paper, 42nd AIAA Thermophysics Conference, Honolulu, HI, June 2011.

11. M. E. Ewing, D. E. Richardson Phenomena and material property requirements for a combined structural and thermal ablation model // 4th AF/SNL/NASA Ablation Workshop, Albuquerque, NM, 2011.

12. Martin A., Boyd D. Chemistry model for ablating carbon-phenolic material during atmospheric re-entry // 48th AIAA Aerospace Sciences Meeting, Orlando, FL, January 2010.

13. Lachaud J., Scoggins J.B., Maginc T.E., Mayer M.G., Mansour N.N. A generic local thermal equilibrium model for porous reactive materials submitted to high temperatures // International Journal of Heat and Mass Transfer, 2017, Vol. 108, Part B, pp. 1406-1417.

10.5.6 Turbulence

The following selection is present in position **Turbulence** in window **Preprocessor > Phases > Phase #i > Physical processes**:

- **(none)**
- **KES** (Standard k-ε model)
- **KEAKN** (Low-Reynolds k-ε model of Abe, Kondoh, Nagano)
- **KEFV** (k-ε model *FlowVision*)
- **KENL** (nonlinear k-ε model based on works of E.Baglietto)

- **SST** (Shear Stress Transport k-ω model)
- **SA** (Spalart-Allmaras model)
- **Sm** (algebraic Smagorinsky model)

Selection of turbulence model becomes accessible after activation of process **Motion**.

After selecting one of the k-ε models (**KES**, **KEAKN**, **KEFV**, **KENL**), a heat turbulence model can be selected in position **KEteta model** in window **Preprocessor > Phases > Phase #i > Physical processes > Turbulence**:

- **(none)**
- **KEtetaAKN** (k_θ-ε_θ model of Abe, Kondoh, Nagano)
- **KEtetaS&S** (k_θ-ε_θ model of Sommer and So)

These models enable calculation of variable turbulent Prandtl number.

Two models of wall functions are implemented in *FlowVision*. Selection is performed in position **Solver > Advanced settings > Turbulence > Standard wall functions**:

- **No** (model WFFV)
- **Yes** (model WFS)

Using wall functions enables calculation of flows over a solid surface on a relatively coarse grid without resolution of the part of the boundary layer adjacent to the surface.

In the *FlowVision* software, simulation of turbulent flows reposes on using turbulent viscosity in the equations describing physical processes.

Turbulent viscosity μ_t enters the diffusion coefficients of the Navier-Stokes and other convection-diffusion equations. The way of calculation of μ_t is determined by the selected turbulence model.

This section has individual numeration of references.

10.5.6.1 Notations

Notation	Physical quantity	Name in *FlowVision*	Dimension
A_1	parameter of wall functions model **WFFV**	**WF: A1**	
A_2	parameter of wall functions model **WFFV**	**WF: A2**	
A_ε	parameter of wall functions model **WFFV** (of profile ε)	**WF: A_eps**	
a_{vis}	parameter of "viscous" profile ω	**a_vis**	
B_{w1}	parameter of wall functions model **WFFV** (of temperature profile)	**WF: Bw1**	
B_{w2}	parameter of wall functions model **WFFV** (of temperature profile)	**WF: Bw2**	
C_1	parameter of model **KEFV**	**C1**	
C_2	parameter of model **KEFV**	**C2**	
C_3	parameter of model **KEFV**	**C3**	
C_4	parameter of model **KEFV**	**C4**	
C_{PP}	parameter of model **KEFV**	**C_pp**	
C_5	parameter of model **KEFV**	**C5**	
$C_{\varepsilon 1}$	parameter of model **KEFV**	**C_e1**	
$C_{\varepsilon 2}$	parameter of model **KEFV**	**C_e2**	
$C_{\varepsilon 2}$	parameter of model **KEFV**	**C_e3**	
d_{Pr}	parameter in the expression for the turbulent Prandtl number	**dPr_t(Pe_t)**	
GD_{max}	parameter of model KEFV	**GD_max**	

Notation	Physical quantity	Name in *FlowVision*	Dimension
I_t	Intensity of turbulent pulsations. The following cases can be distinguished: • low level of turbulization (I_t < 0.03) • medium level of turbulization (0.03 < I_t < 0.05) • high level of turbulization (0.05 < I_t < 0.1)	**Pulsations**	
$C_\mu = \beta^* = 0.09$	frequently used coefficient		
E_1	parameter of wall functions model **WFS**	**WF: E1**	
E_2	parameter of wall functions model **WFS**	**WF: E2**	
h_s	equivalent "sand" roughness of wall	**Roughness**	m
$h_s^+ = \frac{\rho u_\tau h_s}{\mu}$	dimensionless equivalent "sand" roughness of wall		
$\boldsymbol{J}_q$	specific heat flux (vector)	**HeatFlux**	$W \cdot m^{-2} = kg\ s^{-3}$
$J_{q,y}$	normal (to wall) component of specific heat flux (scalar)	**HeatFlux**	$W \cdot m^{-2} = kg\ s^{-3}$
k	turbulent energy	**TurbulentEnergy**	$m^2\ s^{-2}$
$k_\theta = \langle T'^2 \rangle / 2$	half of temperature variance	**Kteta**	K^2
L_t	Linear scale of turbulent eddies. The following cases can be distinguished: • low level of turbulization (L_t < 0.01 L) • medium level of turbulization (0.01 L < L_t < 0.1 L) • high level of turbulization (0.1 L < L_t < L) where L is the characteristic dimension.	**Turbulent scale**	m
Pr_t	turbulent Prandtl number	**PrandtlTurb**	
$R = \frac{k_\theta/\varepsilon_\theta}{k/\varepsilon}$	ratio of the heat time scale to the dynamic one		
$Re_t = \frac{k^2}{\nu\varepsilon}$	turbulent Reynolds number		
$Re_{t,min}$	parameter of model **KEFV**	**Re_t_min**	
$\tilde{R}e_{\theta t}$	the Reynolds number computed via the momentum loss thickness on a solid surface	**TR_RETETA**	
Sc_t	turbulent Schmidt number	**SchmidtTurb**	
$S_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right)$	component of strain rate tensor		s^{-1}
T	relative temperature	**Temperature**	K

Notation	Physical quantity	Name in *FlowVision*	Dimension
T'	temperature pulsation (deviation from the mean value at given instant)		K
$T^+ = \frac{\rho C_p u_\tau (T_w - T_c)}{J_{q,y}}$	dimensionless temperature wall functions model		
$Tu = 100 \cdot I_t$	turbulence intensity expressed in %		
$u_k = C_\mu^{1/4} k^{1/2}$	characteristic velocity based on turbulent energy		m s^{-1}
u_τ	dynamic velocity		m s^{-1}
$V_{x,1}(y)$	profile of tangential component of velocity in the absence of pressure gradient		m s^{-1}
$V^+_{x,1}(y^+_\tau) = V_{x,1} / u_\tau$	profile of dimensionless tangential velocity in the absence of pressure gradient		
$V_{x,2}(y)$	profile of tangential component of velocity in separation point		m s^{-1}
$V^+_{x,2}(y^+_p) = V_{x,2} / u_p$	profile of dimensionless tangential velocity in separation point		
$W_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} - \frac{\partial V_j}{\partial x_i}\right)$	component of vorticity tensor		s^{-1}
y	distance to nearest wall	**Distance to wall**	m
$y^+_\tau = \frac{u_\tau y}{\nu}$	dimensionless distance to nearest wall	**Y_plus**	
$y^+_p = \frac{u_p y}{\nu}$	dimensionless distance to nearest wall		
$y^+_{Kolm} = \frac{(\varepsilon\nu)^{1/4} y}{\nu}$	dimensionless distance to nearest wall		
β	coefficient of thermal expansion	**Therm. expansion**	K^{-1}
γ	intermittency	**TR_GAMMA**	
ε	rate of dissipation of turbulent energy	**TurbDissipation**	m^2 s^{-3}
ε_θ	rate of dissipation of temperature dispersion	**Eteta**	K^2 s^{-1}
κ_1	parameter of wall functions models **WFFV** and **WFS**	**WF: kappa1**	
κ_2	parameter of wall functions models **WFFV** and **WFS**	**WF: kappa2**	
μ	dynamic coefficient of viscosity (molecular)	**Viscosity**	Pa s = kg m^{-1} s^{-1}
μ_t	dynamic coefficient of viscosity (turbulent)	**TurbViscosity**	Pa s = kg m^{-1} s^{-1}
ν	kinematic coefficient of viscosity (molecular)		m^2 s^{-1}
ν_t	kinematic coefficient of viscosity (turbulent)		m^2 s^{-1}

Notation	Physical quantity	Name in *FlowVision*	Dimension
σ_k	parameter of models **KES**, **KEAKN**, **KEFV, KENL**	**Sigma_k**	
σ_ε	parameter of models **KES**, **KEAKN**, **KEFV, KENL**	**Sigma_e**	
ω	specific rate of dissipation of turbulent energy	**TurbDissipation specific**	s^{-1}
$\tau = \rho u_\tau^2$	specific viscous force exerted by fluid onto solid surface (this value can be negative)	**Shear stress**	Pa = kg m^{-1} s^{-2}
Φ	distance potential		m^2

Indices:
ini - initial value,
inl - inlet value,
n - value at time layer n (in numerical method)
n+1 - value at time layer n+1 (in numerical method)
x - tangent (to boundary) component of vector
y - normal (to boundary) component of vector
w - value at wall
cell - value in cell center

See other notations in the section Basic notations.

10.5.6.2 Parameters

Basic parameters of the element Preprocessor > Phases > Phase #N > Physical processes > Turbulence

Parameter	Possible values	Default value	Description
Math. model	• SST • Sm • SA • KES • KEAKN • KEFV • KENL		A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.)
Time step coefficient	An arbitrary numerical value determined by the problem (the default value is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different time steps for different processes sometimes accelerates convergence to the steady-state solution. Meaning of different values: **1** = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the problem as a whole). **> 1** = The calculations of	**1**	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the whole project (τ). *Thus:* Own time step of a physical process = τ × **Time step coefficient** Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time.

Parameter	Possible values	Default value	Description
	the given process are accelerated. **(0, 1)** = The calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations / initialization).		
PrandtlTurb	numerical value	**1**	Turbulent Prandtl number
dPr_t(Pe_t)	numerical value	**0**	Parameter at the additional term in the expression for the turbulent Prandtl number. This term allows for dependency of the turbulent Prandtl number on the turbulent Peclet number.
SchmidtTurb	numerical value	**1**	Turbulent Schmidt number
WF: kappa1	numerical value	**0.41**	Parameter of models 'Standard wall functions' and 'Wall functions FV'
WF: E1	numerical value	**9**	Parameter of model 'Standard wall functions"
WF: A1	numerical value	**26**	Parameter of model 'Wall functions FV"
WF: kappa2	numerical value	**0.2**	
WF: E2	numerical value	**4.955**	
WF: A2	numerical value	**13.6**	
WF: P gradient [1)]	• **Yes** • **No**	**No**	This parameter enables/disables account of the pressure gradient: • **No**: wall functions are used without taking the downstream pressure gradient into account • **Yes**: wall functions are used with taking the downstream pressure gradient into account
Roughness constant	**0.1 - 0.5**	**0.276**	Value of the the roughness constant
Therm. expansion	**0 - 0.1**	**0**	Coefficient β in generation terms G_k (taking into account the effect of the buoyancy force)
Ksi[2)]	**0 - 1.5**	**1.5**	Constant ξ in the equation for turbulent energy (in the coefficient which allows for the effect of compressibility on dissipation)
Mt0[2)]	**0 - 0.25**	**0.25**	Constant M_{t0} in the equation for turbulent energy (in the coefficient which allows for the effect of compressibility on dissipation)

[1)] Currently, the account of pressure gradient is appropriate in both wall function's models (**WFFV** and **WFS**).

[2)] The default combination of values of **Ksi** and **Mt0** (that is **Ksi=1.5**, **Mt0=0.25**), corresponds to the Wilcox' model. The combination of **Ksi=1** and **Mt0=0** corresponds to the Sarkar's model. See *Wilcox D. C. (1994) "Turbulence modeling for CFD", DCW Industries, Inc., 460 p.*

When a model **KES**, **KEAKN**, **KEFV**, or **KENL** is selected, the following parameters become accessible:

Parameter	Possible values	Default value	Description
WF: A_eps	numerical value	**15**	Parameter of model 'Wall functions FV"
KEteta model	• **(none)** • **AKN** • **S&S** • **LMS**	**(none)**	Model of turbulent heat transfer
Sigma_k	numerical value	**1**	Model constants
Sigma_e	numerical value	**1.3**	
C_e1	numerical value	**1.44**	
C_e2	numerical value	**1.92**	
Bradshaw	• **No** • **Yes**	**No**	**No** = Turbulent viscosity is computed by a standard method. **Yes** = Turbulent viscosity is computed by the Bradshaw formula.
a_1	numerical value	**0.31**	The a_1 parameter in the Bradshaw formula

When the **KEFV** model is selected, the following parameters become also accessible:

Parameter	Possible values	Default value	Description
C1	numerical value	**5**	Model constants
C2	numerical value	**50**	
C_e3	numerical value	**0.05**	
C3	numerical value	**0.335**	
C4	numerical value	**3.4**	
C5	numerical value	**5.8**	
C6	numerical value	**10000**	
Nu_t_gen	numerical value	**0**	
C_pp	numerical value	**0.003**	
Dist_pp	numerical value	**10^{20}**	
C_Rich	numerical value	**0**	Model constant at the Richardson number (in the coefficient which allows for streamline's curvature).

When the **SST** model is selected, the following parameters become also accessible:

Parameter	Possible values	Default value	Description
W_turb	**1 | 2 | 3**	**1**	One of the three implemented ways to specify the profile of variable ω in high-Reynolds calculations [1)]
W_vis	**1 | 2**	**1**	One of the three implemented viscous profiles of variable ω.
a_vis	numerical value	**0.6**	Parameter of viscous profiles of variable ω.
C_Rich	numerical value	**0**	Model constant at the Richardson number (in the coefficient which takes account of the curvature of streamlines).
Transition	• **Yes** • **No**	**No**	**Yes** = Transition model $\gamma - \tilde{R}e_{\theta t}$ is activated **No** = Transition model $\gamma - \tilde{R}e_{\theta t}$ is not activated

Parameter	Possible values	Default value	Description
sigma_f	numerical value	**1**	Constant σ_f of transition model $\gamma - \tilde{R}e_{\theta t}$.
sigma_t	numerical value	**2**	Constant $\sigma_{\theta t}$ of transition model $\gamma - \tilde{R}e_{\theta t}$.
c_a1	numerical value	**2**	Constant c_{a1} of transition model $\gamma - \tilde{R}e_{\theta t}$.
c_a2	numerical value	**0.06**	Constant c_{a2} of transition model $\gamma - \tilde{R}e_{\theta t}$.
c_e1	numerical value	**1**	Constant c_{e1} of transition model $\gamma - \tilde{R}e_{\theta t}$.
c_e2	numerical value	**50**	Constant c_{e2} of transition model $\gamma - \tilde{R}e_{\theta t}$.
c_t	numerical value	**0.03**	Constant $c_{\theta t}$ of transition model $\gamma - \tilde{R}e_{\theta t}$.
s_1	numerical value	**2**	Constant s_1 of transition model $\gamma - \tilde{R}e_{\theta t}$.
Re_v - Re_tc	numerical value	**3.235**	Constant controlling ratio $Re_v / Re_{\theta c}$ in transition model $\gamma - \tilde{R}e_{\theta t}$.
gamma_sep_lim	numerical value	**2**	Constant limiting quantity γ_{sep} from above in transition model $\gamma - \tilde{R}e_{\theta t}$.
F_reattach	numerical value	**20**	Parameter of function $F_{reattach}$ in transition model $\gamma - \tilde{R}e_{\theta t}$.
Tu_min	numerical value	**0.027**	Constant limiting turbulence intensity Tu from below in transition model $\gamma - \tilde{R}e_{\theta t}$.

[1] Selection **W_turb = 1** or **W_turb = 2** assumes that two profiles are combined: viscous (which occurs in the viscous sub-layer) and logarithmic ones. One of the two implemented viscous profiles, specified by parameter **W_vis**, is used in these calculations. There are two ways to specify the value of variable ω in the center of a near-wall cell for low-Reynolds calculations. They are determined by values **W_vis = 1, 2**. There are five ways to specify the value of variable ω in the center of a near-wall cell for in high-Reynolds calculations. They are determined by the following combinations (**W_turb**, **W_vis**) **= (1, 1), (1, 2), (2, 1), (2, 2), (3, 1** or **2)**. In the last case parameter **W_vis** does not affect result. If equilibrium wall functions are specified on the given wall, a proper formula $k_c(y)$ is used for each value of parameter **W_turb**.

When the **KENL** model is selected, the following numerical parameters become also accessible:

Parameter	Default value	Description
C_a0	**0.66667**	Model constant C_{a0}
C_a1	**5.5**	Model constant C_{a1}
C_a2	**0.7**	Model constant C_{a2}
C_NL1	**0.8**	Model constant C_{NL1}
C_NL2	**11**	Model constant C_{NL2}
C_NL3	**4.5**	Model constant C_{NL3}

Parameter	Default value	Description
C_NL4	-5	Model constant C_{NL4}
C_NL5	-4.5	Model constant C_{NL5}
C_NL6	1000	Model constant C_{NL6}
C_NL7	1	Model constant C_{NL7}

Parameters of the turbulence in the advanced settings of Solver

These parameters are set in the advanced settings of Solver (the **Turbulence** group of parameters).

Parameter	Possible values	Description
Dist. via potential	• **Yes** • **No**	• **Yes** = In all the cells, the distance to the nearest wall is calculated using the method described in section **Distance to wall** • **No** = In a near-wall cell, the distance to the nearest wall is calculated as the minimum over the distances from the center of this cell (in general case, cut by geometry) to the geometry triangles found in this cell. [1]
Standard wall functions	• **Yes** • **No**	• **Yes** = the program uses the **WFS** model for calculations. • **No** = the program uses the **WFFV** model for calculations.'
e/k background	numerical value The default value is **0.09**	Background frequency of turbulent pulsations [s^{-1}]
WF: profile T+ [2]	• **1** • **2** • **3**	One of the three implemented profiles of temperature.
T+ Kader [2]	• **Yes** • **No**	• **Yes** = Viscous and turbulent profiles $T^+(y^+)$ are combined using the Kader formula [11]. • **No** = The minimum value is taken: $T^+(y^+) = min(T^+_{vis}, T^+_{turb})$.
Variation of properties	• **Yes** • **No**	• **Yes** = Variation of the fluid properties between a wall and the center of the adjacent cell is taken into account. • **No** = Variation of the fluid properties is no taken into account in the wall functions.

[1] In all other cells, the distance to the nearest wall is calculated using the method described in section **Distance to wall**.

[2] When turbulent heat transfer in circular pipes is simulated, it is recommended to set **WF: profile T+ = 1** and **T+ Kader = No**.

Phase limiters for kinematic coefficient of turbulent viscosity

These are parameters **Nu turb. min.** and **Nu turb. max. / Nu mol.**, which are set in the folder **Limiters > Limiters for calculation > Phase Limiters > (Phase)** in the **Solver** tab of the project tree. In some problems, limitation of the maximum turbulent viscosity prevents non-physical results at the initial stage of formation the solution.

The default values of these parameters are 10^{-8} and 10^{20}.

10.5.6.3 Equations

This section describes the implemented turbulence models and wall functions.

Every following section has individual numeration of equations.

10.5.6.3.1 Model KES

k-ε model 'Standard':

$$\frac{\partial(\rho k)}{\partial t}+\nabla(\rho \boldsymbol{V} k)=\nabla\left(\left(\mu+\frac{\mu_t}{\sigma_k}\right)\nabla k\right)+\rho(P_k+G_k)-\rho\varepsilon\left(1+\xi\left(max\left(M_t^2,M_{t0}^2\right)-M_{t0}^2\right)\right) \quad \text{(Turb-KES.1)}$$

$$\frac{\partial(\rho \varepsilon)}{\partial t}+\nabla(\rho \boldsymbol{V} \varepsilon)=\nabla\left(\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\nabla \varepsilon\right)+C_{\varepsilon 1}\frac{\varepsilon}{k}\rho(P_k+G_k)-C_{\varepsilon 2}\rho\frac{\varepsilon^2}{k} \quad \text{(Turb-KES.2)}$$

$$\mu_t=C_\mu\rho\frac{k^2}{\varepsilon} \quad \text{(Turb-KES.3)}$$

$$P_k=\nu_t\left(S-\frac{2}{3}(\nabla\cdot\mathbf{V})^2\right)-\frac{2}{3}(\nabla\cdot\mathbf{V})k \quad \text{(Turb-KES.4)}$$

$$G_k=\nu_t\frac{\beta}{Pr_t}\boldsymbol{g}\nabla T \quad \text{(Turb-KES.5)}$$

$$S=2\sum_{i,j}S_{ij}S_{ij}=\sum_{i,j}\left(\frac{\partial V_i}{\partial x_j}+\frac{\partial V_j}{\partial x_i}\right)\frac{\partial V_j}{\partial x_i} \quad \text{(Turb-KES.6)}$$

$$S_{ij}=\frac{1}{2}\left(\frac{\partial V_i}{\partial x_j}+\frac{\partial V_j}{\partial x_i}\right) \quad \text{(Turb-KES.7)}$$

$$M_t^2=\frac{2k}{c^2} \quad \text{(Turb-KES.8)}$$

$$\sigma_k=1,\quad \sigma_\varepsilon=1.3,\quad C_1=1.44,\quad C_2=1.92,\quad \xi=1.5,\quad M_{t0}=0.25 \quad \text{(Turb-KES.9)}$$

Here

c is the sonic speed

$C_\mu=0.09$ is a fixed model constant

User can also select calculation of turbulent viscosity by the Bradshaw formula:

$$\mu_t=\frac{a_1 k}{max\left[a_1\frac{\varepsilon}{C_\mu k},\sqrt{S}\cdot tanh\left(arg^2\right)\right]} \quad \text{(Turb-KES.3_a)}$$

$$arg=\frac{k}{\varepsilon}\cdot max\left[\frac{2\sqrt{k}}{y},\frac{45\nu}{y^2}\right] \quad \text{(Turb-KES.3_b)}$$

Here

y is distance to nearest wall,

a_1 is a model constant equal 0.31 by default.

If **Phase #i > Physical processes > Turbulence >** Therm. expansion = 0 (value β = 0 is specified in the interface), turbulence generation due to the buoyancy force in Eqs. (Turb-KES.1) and (Turb-KES.2) is not allowed for: $G_k = 0$.

If β > 0 and **Substance #i > Aggregative state** = **Gas**, the thermal expansion coefficient is computed automatically:

$$\beta = \frac{1}{T_{abs}} \qquad \text{(Turb-KES.10)}$$

If β > 0 and **Substance #i > Aggregative state** = **Liquid**, the interface value of β is used in Eqs. (Turb-KES.2) and (Turb-KES.3).

Description of the model see in [1]. The default values of the model constants are given in (Turb-KES.9). They can be changed in the interface. Constants ξ and M_{t0} define a model intended for account of the fluid viscosity. The aforementioned values define the Wilcox model. Values $\xi = 1$, $M_{t0} = 0$ define the Sarkar model. Both models are discussed in [1]. Specifying $\xi = 0$ toggles off the compressibility effect in the k equation.

This model can be used only in high-Reynolds calculations (on a relatively coarse grid, using wall functions).

10.5.6.3.2 Model KEAKN

k-ε model 'Abe, Kondoh, Nagano':'

$$\frac{\partial(\rho k)}{\partial t} + \nabla(\rho \boldsymbol{V} k) = \nabla\left(\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right) + \rho(P_k + G_k) - \rho\varepsilon\left(1 + \xi\left(max\left(M_t^2, M_{t0}^2\right) - M_{t0}^2\right)\right) \qquad \text{(Turb-KEAKN.1)}$$

$$\frac{\partial(\rho \varepsilon)}{\partial t} + \nabla(\rho \boldsymbol{V} \varepsilon) = \nabla\left(\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\nabla \varepsilon\right) + C_{\varepsilon 1}\frac{\varepsilon}{k}\rho(P_k + G_k) - C_{\varepsilon 2} f_\varepsilon \rho \frac{\varepsilon^2}{k} \qquad \text{(Turb-KEAKN.2)}$$

$$\mu_t = C_\mu f_\mu \rho \frac{k^2}{\varepsilon} \qquad \text{(Turb-KEAKN.3)}$$

$$P_k = \nu_t\left(S - \frac{2}{3}(\nabla \cdot \mathbf{V})^2\right) - \frac{2}{3}(\nabla \cdot \mathbf{V})k \qquad \text{(Turb-KEAKN.4)}$$

$$G_k = \nu_t \frac{\beta}{Pr_t}\boldsymbol{g}\nabla T \qquad \text{(Turb-KEAKN.5)}$$

$$S = 2\sum_{i,j} S_{ij}S_{ij} = \sum_{i,j}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right)\frac{\partial V_j}{\partial x_i} \qquad \text{(Turb-KEAKN.6)}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right) \qquad \text{(Turb-KEAKN.7)}$$

$$f_\varepsilon = \left[1 - exp\left(-\frac{y^+_{Kolm}}{3.1}\right)\right]^2\left[1 - 0.3\,exp\left\{-\left(\frac{Re_t}{6.5}\right)^2\right\}\right] \qquad \text{(Turb-KEAKN.8)}$$

$$f_\mu = \left[1 - exp\left(-\frac{y^+_{Kolm}}{14}\right)\right]^2 \left[1 + \frac{5}{Re_t^{3/4}} exp\left\{-\left(\frac{Re_t}{200}\right)^2\right\}\right] \quad \text{(Turb-KEAKN.9)}$$

$$y^+_{Kolm} = \frac{(\varepsilon\nu)^{1/4} y}{\nu} \quad \text{(Turb-KEAKN.10)}$$

$$Re_t = \frac{k^2}{\varepsilon\nu} \quad \text{(Turb-KEAKN.11)}$$

$$M_t^2 = \frac{2k}{c^2} \quad \text{(Turb-KEAKN.12)}$$

$$\sigma_k = 1.4, \quad \sigma_\varepsilon = 1.4, \quad C_1 = 1.5, \quad C_2 = 1.9, \quad \xi = 1.5, \quad M_{t0} = 0.25 \quad \text{(Turb-KEAKN.13)}$$

Here

c is the sonic speed

$C_\mu = 0.09$ is a fixed model constant

User can also select calculation of turbulent viscosity by the Bradshaw formula:

$$\mu_t = f_\mu \frac{a_1 k}{\max\left[a_1 \frac{\varepsilon}{C_\mu k}, \sqrt{S} \cdot \tanh\left(\arg^2\right)\right]} \quad \text{(Turb-KEAKN.3_a)}$$

$$arg = \frac{k}{\varepsilon} \cdot max\left[\frac{2\sqrt{k}}{y}, \frac{45\nu}{y^2}\right] \quad \text{(Turb-KEAKN.3_b)}$$

Here

y is distance to nearest wall,

a_1 is a model constant equal 0.31 by default.

Description of the model see in [2]. Model KEAKN differs from model KES by presence of damping functions f_ε and f_μ which depend on dimensionless distance to the nearest wall y^+_{Kolm} and turbulent Reynolds number Re_t. The default values of the model constants are given in (Turb-KEAKN.13). They can be changed in the interface. Constants ξ and M_{t0} define a model intended for account of the fluid viscosity. The aforementioned values define the Wilcox model. Values $\xi = 1$, $M_{t0} = 0$ define the Sarkar model. Both models are discussed in [1]. Specifying $\xi = 0$ toggles off the compressibility effect in the k equation.

This model is recommended only for low-Reynolds calculations (on a grid resolving viscous sub-layers near walls, without using wall functions).

10.5.6.3.3 Model KEFV

k-ε model *'FlowVision'*:

$$\frac{\partial(\rho k)}{\partial t} + \nabla(\rho V k) = \nabla\left(\left(\mu + f_t \frac{\mu_t}{\sigma_k}\right)\nabla k\right) + \rho\left(P_k + P_{k,gen} + G_k\right) - \rho\varepsilon\left(1 + \xi\left(max\left(M_t^2, M_{t0}^2\right) - M_{t0}^2\right)\right) - D_{pp} \quad \text{(Turb-KEFV.1)}$$

$$\frac{\partial(\rho\varepsilon)}{\partial t}+\nabla(\rho\mathbf{V}\varepsilon)=\nabla\left(\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\nabla\varepsilon\right)+\frac{1}{\sqrt{1+\frac{2}{Re_t}}}\frac{\varepsilon}{k}\rho(C_{\varepsilon 1}f_1(P_k+G_k)-C_{\varepsilon 2}f_2\varepsilon)$$ (Turb-KEFV.2)

$$\mu_t=C_\mu\rho\frac{k^2}{\varepsilon}$$ (Turb-KEFV.3)

$$P_k=\nu_t\left(S-\frac{2}{3}(\nabla\cdot\mathbf{V})^2\right)-\frac{2}{3}(\nabla\cdot\mathbf{V})k$$ (Turb-KEFV.4)

$$P_{k,gen}=\nu_{t,gen}S\,exp\left\{-\left(\frac{Re_t}{C_6}\right)^2\right\}$$ (Turb-KEFV.5)

$$G_k=\nu_t\frac{\beta}{Pr_t}\boldsymbol{g}\cdot\nabla T$$ (Turb-KEFV.6)

$$D_{pp}=\begin{cases}0 & y>Dist_{pp}\\ C_{pp}\rho\{\boldsymbol{n}\cdot\nabla(k|\boldsymbol{V}|)\} & y\le Dist_{pp}\end{cases}$$ (Turb-KEFV.7)

$$f_t=1+C_1\,exp\left\{-\left(\frac{Re_t}{C_2}\right)^2\right\}$$ (Turb-KEFV.8)

$$f_1=1-C_{\varepsilon 3}+C_{\varepsilon 3}\frac{P_k}{\varepsilon}$$ (Turb-KEFV.9)

$$f_2=\frac{1}{1+C_{Rich}\sqrt{\frac{W}{S}}\frac{\sqrt{W}-\sqrt{S}}{\sqrt{W}+\sqrt{S}}}\left\{1-exp\left(-\frac{y^+_{Kolm}}{C_4}\right)\right\}^2\left[1-C_3\,exp\left\{-\left(\frac{Re_t}{C_6}\right)^2\right\}\right]$$ (Turb-KEFV.10)

$$y^+_{Kolm}=\frac{(\nu\cdot\varepsilon)^{1/4}y}{\nu}$$ (Turb-KEFV.11)

$$Re_t=\frac{k^2}{\nu\varepsilon}$$ (Turb-KEFV.12)

$$M_t^2=\frac{2k}{c^2}$$ (Turb-KEFV.13)

$$S=2\sum_{i,j}S_{ij}S_{ij}$$ (Turb-KEFV.14)

$$S_{ij}=\frac{1}{2}\left(\frac{\partial V_i}{\partial x_j}+\frac{\partial V_j}{\partial x_i}\right)$$ (Turb-KEFV.15)

$$W = 2\sum_{i,j} W_{ij}W_{ij} \qquad \text{(Turb-KEFV.16)}$$

$$W_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} - \frac{\partial V_j}{\partial x_i}\right) \qquad \text{(Turb-KEFV.17)}$$

$$\begin{aligned} &\sigma_k = 1, \quad \sigma_\varepsilon = 1.3, \\ &C_{\varepsilon 1} = 1.44, \quad C_{\varepsilon 2} = 1.92, \quad C_{\varepsilon 3} = 0.05, \\ &C_1 = 5, \quad C_2 = 50, \quad C_3 = 0.335, \quad C_4 = 3.4, \quad C_5 = 5.8, \\ &C_6 = 10000, \quad \nu_{t,gen} = 0, \quad C_{pp} = 0.003, \quad C_{Rich} = 0, \quad \xi = 1.5, \quad M_{t0} = 0.25 \end{aligned} \qquad \text{(Turb-KEFV.18)}$$

Here

$Dist_{pp}$	is the farthest distance at which term D_{pp} is allowed for
$\boldsymbol{n}$	is the normal to wall
c	is the sonic speed
$C_\mu = 0.09$	is a fixed model constant

User can also select calculation of turbulent viscosity by the Bradshaw formula:

$$\mu_t = \frac{a_1 k}{max\left[a_1 \frac{\varepsilon}{C_\mu k}, \sqrt{S}\cdot tanh\left(arg^2\right)\right]} \qquad \text{(Turb-KEFV.3_a)}$$

$$arg = \frac{k}{\varepsilon}\cdot max\left[\frac{2\sqrt{k}}{y}, \frac{45\nu}{y^2}\right] \qquad \text{(Turb-KEFV.3_b)}$$

Here

y	is distance to nearest wall,
a_1	is a model constant that equals to 0.31 by default.

The default values of the model constants are given in (Turb-KEFV.18). They can be changed in the interface. Constants ξ and M_{t0} define a model intended for account of the fluid viscosity. The aforementioned values define the Wilcox model. Values $\xi = 1$, $M_{t0} = 0$ define the Sarkar model. Both models are discussed in [1]. Specifying $\xi = 0$ toggles off the compressibility effect in the k equation.

This model can be used both in low-Reynolds and high-Reynolds calculations. In the first case, the laminar sub-layer is resolved by the computational grid (wall functions are not used). In the second case, the laminar sub-layer is not resolved by the computational grid (wall functions are used). The model predicts the position of bypass transition on a solid surface. In the low-Reynolds calculations, at is assumed that the free stream is turbulent.

10.5.6.3.4 Model KENL

This is a nonlinear k-ε model based on works of E.Baglietto, [4a] - [4c].

KENL is a high-Reynolds model (as well as the **KES** model), because it includes no damping functions. It is intended that they are used together with wall functions.

The **KENL** model belongs to the class of anisotropy algebraic turbulence models. The **KENL** model is tuned by 10 constants that are available from the *FlowVision*'s user interface. Both cubic and quadratic parts of the model can be either enabled or disabled. The quadratic part improves simulating of secondary flows; the cubic part improves simulating of swirling flows.

$$\frac{\partial(\rho k)}{\partial t}+\nabla(\rho \boldsymbol{V} k)=\nabla\left(\left(\mu+\frac{\mu_t}{\sigma_k}\right)\nabla k\right)+P_k+G_k-\rho\varepsilon\left(1+\xi\left(\max\left(M_t^2,M_{t0}^2\right)-M_{t0}^2\right)\right) \quad \text{(Turb-KENL.1)}$$

$$\frac{\partial(\rho \varepsilon)}{\partial t}+\nabla(\rho \boldsymbol{V} \varepsilon)=\nabla\left(\left(\mu+\frac{\mu_t}{\sigma_\varepsilon}\right)\nabla\varepsilon\right)+\frac{\varepsilon}{k}\left(C_{\varepsilon 1}\left(P_k+G_k\right)-C_{\varepsilon 2}\,\rho\varepsilon\right) \quad \text{(Turb-KENL.2)}$$

$$\mu_t=\rho C_\mu k^2/\varepsilon \quad \text{(Turb-KENL.3)}$$

$$P_k=\mu_t\left(S-\frac{2}{3}(\nabla\cdot\boldsymbol{V})^2\right)-\frac{2}{3}\rho(\nabla\cdot\boldsymbol{V})k \quad \text{(Turb-KENL.4)}$$

$$G_k=\mu_t\frac{\beta}{\mathbf{Pr}_t}\boldsymbol{g}\cdot\nabla T \quad \text{(Turb-KENL.5)}$$

Thus the **KENL** model assumes integrating the same equations as the **KES** model. The models differ in the generating complex:

$$S=\sum_{i,j=1}^{3}\frac{\tau_{ij,turb1}+\tau_{ij,turb2}+\tau_{ij,turb3}}{\mu_t}\frac{\partial V_i}{\partial x_j} \quad \text{(Turb-KENL.6)}$$

In the **KENL** model, the shear stress tensor is determined by the formulae:

$$\tau_{ij}=\tau_{ij,mol}+\tau_{ij,turb1}+\tau_{ij,turb2}+\tau_{ij,turb3} \quad \text{(Turb-KENL.7)}$$

$$\tau_{ij,mol}=2\mu\left(S_{ij}-\frac{1}{3}S_{kk}\right) \quad \text{(Turb-KENL.8)}$$

$$\tau_{ij,turb1}=2\mu_t\left(S_{ij}-\frac{1}{3}S_{kk}\right)-\frac{2}{3}k\delta_{ij} \quad \text{(Turb-KENL.9)}$$

$$\tau_{ij,turb2}=-4C_1\mu_t\frac{k}{\varepsilon}\left(SS_{ij}-\frac{1}{3}S^2\delta_{ij}\right)-$$
$$-4C_2\mu_t\frac{k}{\varepsilon}\left(WS_{ij}+WS_{ji}\right)-4C_3\mu_t\frac{k}{\varepsilon}\left(WW_{ij}-\frac{1}{3}W^2\delta_{ij}\right) \quad \text{(Turb-KENL.10)}$$

$$\tau_{ij,turb3}=-8C_4\mu_t\frac{k^2}{\varepsilon^2}\left(SSW_{ij}+SSW_{ji}\right)-8C_5\mu_t\frac{k^2}{\varepsilon^2}\left(S^2-W^2\right)\delta_{ij} \quad \text{(Turb-KENL.11)}$$

Here:

$$S^2=\sum_{k,l=1}^{3}S_{kl}S_{lk}$$

$$W^2 = \sum_{k,l=1}^{3} \Omega_{kl}\Omega_{kl} =$$
$$= \Omega_{11}\Omega_{11} + \Omega_{12}\Omega_{12} + \Omega_{13}\Omega_{13} +$$
$$+ \Omega_{21}\Omega_{21} + \Omega_{22}\Omega_{22} + \Omega_{23}\Omega_{23} +$$
$$+ \Omega_{31}\Omega_{31} + \Omega_{32}\Omega_{32} + \Omega_{33}\Omega_{33} =$$
$$= 2(\Omega_{12})^2 + 2(\Omega_{13})^2 + 2(\Omega_{23})^2$$

$$SS_{ij} = \sum_{k=1}^{3} S_{ik}S_{kj}$$

$$WW_{ij} = \sum_{k=1}^{3} \Omega_{ik}\Omega_{kj}$$

$$WS_{ij} = \sum_{k=1}^{3} \Omega_{ik}S_{kj}$$

$$SSW_{ij} = \sum_{l=1}^{3} SS_{il}\Omega_{lj}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right)$$

$$W_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} - \frac{\partial V_j}{\partial x_i}\right)$$

$$C_\mu = \frac{C_{a0}}{C_{a1} + C_{a2}\,\mathbf{max}\left(S^*, \Omega^*\right)}$$

$$C_1 = \frac{C_{NL1}}{\left(C_{NL6} + C_{NL7}S^{*3}\right)C_\mu}$$

$$C_2 = \frac{C_{NL2}}{\left(C_{NL6} + C_{NL7}S^{*3}\right)C_\mu}$$

$$C_3 = \frac{C_{NL3}}{\left(C_{NL6} + C_{NL7}S^{*3}\right)C_\mu}$$

$$C_4 = C_{NL4}C_\mu^2$$

$$C_5 = C_{NL5}C_\mu^2$$

$$S^* = \frac{k}{\varepsilon}\sqrt{2S^2}$$

$$\Omega^* = \frac{k}{\varepsilon}\sqrt{2\Omega^2}$$

The model constants are available from the FlowVision's user interface and have the following default values:

Constant	Name in *FlowVision*	Default value
C_{a0}	**C_a0**	**0.66667**
C_{a1}	**C_a1**	**5.5**
C_{a2}	**C_a2**	**0.7**
C_{NL1}	**C_NL1**	**0.8**
C_{NL2}	**C_NL2**	**11**
C_{NL3}	**C_NL3**	**4.5**
C_{NL4}	**C_NL4**	**-5**
C_{NL5}	**C_NL5**	**-4.5**
C_{NL6}	**C_NL6**	**1000**
C_{NL7}	**C_NL7**	**1**

Formula (Turb-KENL.8) determines the molecular part of the shear stress tensor.

Formula (Turb-KENL.9) determines the linear turbulent part of the shear stress tensor.

Formula (Turb-KENL.10) determines the quadratic turbulent part of the shear stress tensor.

Formula (Turb-KENL.11) determines the cubic turbulent part of the shear stress tensor.

Anisotropic viscous force supplement for the KENL turbulence model

The **KENL** model assumes use of an anisotropic viscous force supplement (specified by the **Visc. force supplement** parameter) on edges of cells with the external normal $\boldsymbol{n}$:

$$d\boldsymbol{\tau}_n = d\boldsymbol{\tau}_{n,1} + d\boldsymbol{\tau}_{n,2} + d\boldsymbol{\tau}_{n,3} \qquad \text{(Turb-KENL.12)}$$

$$d\boldsymbol{\tau}_{n,1} = \begin{Bmatrix} (\mu+\mu_t)\left\{\left[\frac{\partial V_x}{\partial x} - \frac{2}{3}\left(\frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z}\right)\right]n_x + \frac{\partial V_y}{\partial x}n_y + \frac{\partial V_z}{\partial x}n_z\right\} \\ (\mu+\mu_t)\left\{\frac{\partial V_x}{\partial y}n_x + \left[\frac{\partial V_y}{\partial y} - \frac{2}{3}\left(\frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z}\right)\right]n_y + \frac{\partial V_z}{\partial y}n_z\right\} \\ (\mu+\mu_t)\left\{\frac{\partial V_x}{\partial z}n_x + \frac{\partial V_y}{\partial z}n_y + \left[\frac{\partial V_z}{\partial z} - \frac{2}{3}\left(\frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z}\right)\right]n_z\right\} \end{Bmatrix} \qquad \text{(Turb-KENL.13)}$$

$$d\boldsymbol{\tau}_{n,2} = \boldsymbol{\tau}_{turb\,2} \cdot \boldsymbol{n} \qquad \text{(Turb-KENL.14)}$$

$$d\boldsymbol{\tau}_{n,3} = \boldsymbol{\tau}_{turb\,3} \cdot \boldsymbol{n} \qquad \text{(Turb-KENL.15)}$$

Calculating the turbulent viscosity by the Bradshaw formula

When **Bradshaw = Yes** is set in properties of the **Physical processes > Turbulence** element, the turbulent viscosity is calculated by the Bradshaw formula:

$$\mu_t = f_\mu \frac{a_1 k}{\max\left[a_1 \frac{\varepsilon}{C_\mu k}, \sqrt{S}\cdot\tanh\left(\arg^2\right)\right]} \quad \text{(Turb-KENL.3_a)}$$

$$arg = \frac{k}{\varepsilon}\cdot max\left[\frac{2\sqrt{k}}{y}, \frac{45\nu}{y^2}\right] \quad \text{(Turb-KENL.3_b)}$$

Here

y is distance to the nearest wall,

a_1 is a model constant with default value **0.31** (this value is set in the *FlowVision*'s user interface by the parameter **a_1**).

10.5.6.3.5 Model SST

k-ω model 'Shear Stress Transport':

$$\frac{\partial(\rho k)}{\partial t} + \nabla(\rho \boldsymbol{V} k) = \nabla\left(\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right) + \rho P_k - \rho\beta^* k\omega\left(1 + \xi\left(max\left(M_t^2, M_{t0}^2\right) - M_{t0}^2\right)\right) \quad \text{(Turb-SST.1)}$$

$$\frac{\partial(\rho\omega)}{\partial t} + \nabla(\rho \boldsymbol{V}\omega) = \nabla\left(\left(\mu + \frac{\mu_t}{\sigma_\omega}\right)\nabla\omega\right) + \rho\alpha S - F_4\rho\beta\omega^2 + D_\omega \quad \text{(Turb-SST.2)}$$

$$\mu_t = \rho\frac{0.31k}{max\left[0.31\omega, F_2F_3S\right]} \quad \text{(Turb-SST.3)}$$

$$\sigma_k = \frac{1}{F_1\sigma_{k,1} + (1-F_1)\sigma_{k,2}} \quad \text{(Turb-SST.4)}$$

$$\sigma_\omega = \frac{1}{F_1\sigma_{\omega,1} + (1-F_1)\sigma_{\omega,2}} \quad \text{(Turb-SST.5)}$$

$$P_k = min\left(\nu_t S, 10\beta^* k\omega\right) \quad \text{(Turb-SST.6)}$$

$$D_\omega = 2(1-F_1)\rho\sigma_{\omega,2}\frac{1}{\omega}\nabla k\cdot\nabla\omega \quad \text{(Turb-SST.7)}$$

$$\alpha = F_1\cdot 5/9 + (1-F_1)\cdot 0.44 \quad \text{(Turb-SST.8)}$$

$$\beta = F_1\cdot 0.075 + (1-F_1)\cdot 0.0828 - \beta^*\varsigma^* F(M_t) \quad \text{(Turb-SST.9)}$$

$$F_1 = tanh\left(\Phi_1^4\right) \quad \text{(Turb-SST.10)}$$

$$F_2 = tanh\left(\Phi_2^2\right) \quad \text{(Turb-SST.11)}$$

$$F_3 = 1 - tanh\left[\left(\frac{150\mu}{\rho y^2\omega}\right)^4\right] \quad \text{(Turb-SST.12)}$$

$$F_4 = \frac{1}{1 + C_{Rich}\sqrt{\frac{W}{S}}\left(\sqrt{\frac{W}{S}} - 1\right)} \quad \text{(Turb-SST.13)}$$

$$\Phi_1 = min\left[max\left(\frac{\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right), \frac{4\sigma_{\omega,2} \rho k}{CD_{k\omega} y^2} \right]$$ (Turb-SST.14)

$$\Phi_2 = max\left(2\frac{\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right)$$ (Turb-SST.15)

$$CD_{k\omega} = max\left[2\rho\sigma_{\omega,2} \frac{1}{\omega} \nabla k \cdot \nabla \omega, 10^{-10} \right]$$ (Turb-SST.16)

$$S = 2\sum_{i,j} S_{ij} S_{ij}$$ (Turb-SST.17)

$$S_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i} \right)$$ (Turb-SST.18)

$$W = 2\sum_{i,j} W_{ij} W_{ij}$$ (Turb-SST.19)

$$W_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} - \frac{\partial V_j}{\partial x_i} \right)$$ (Turb-SST.20)

$$M_t^2 = \frac{2k}{c^2}$$ (Turb-SST.21)

$$\sigma_{k,1} = 0.85, \quad \sigma_{\omega,1} = 0.5, \quad \sigma_{k,2} = 1, \quad \sigma_{\omega,2} = 0.856, \quad \beta^* = 0.09, \quad C_{Rich} = 0, \quad \xi = 1.5, \quad M_{t0} = 0.25$$ (Turb-SST.22)

Here

c is the sonic speed.

Description of the model see in [5]. The method of account of compressibility in the k equation (the dependency of the dissipation term on M_t) see in [1]. The account of streamline curvature (coefficient F_4) is taken from [6]. This model can be used both in low-Reynolds and high-Reynolds calculations.

Model SST $k-\omega$ is supplemented by transition model $\gamma - \tilde{R}e_{\theta t}$. This model assumes solving two additional convection-diffusion equations:

$$\frac{\partial(\rho\gamma)}{\partial t} + \nabla(\rho \boldsymbol{V} \gamma) = \nabla\left(\left(\mu + \frac{\mu_t}{\sigma_f} \right) \nabla\gamma \right) +$$
$$+ c_{a1}\, \rho \sqrt{S}\, F_{length} \sqrt{F_{onset}} \sqrt{\gamma}(1 - c_{e1}\gamma) + c_{a2}\, \rho \sqrt{W}\, F_{turb}\gamma(1 - c_{e2}\gamma)$$ (Turb-SST.23)

$$\frac{\partial(\rho \tilde{R}e_{\theta t})}{\partial t} + \nabla(\rho \boldsymbol{V} \tilde{R}e_{\theta t}) = \nabla\left(\sigma_{\theta t}(\mu + \mu_t) \nabla \tilde{R}e_{\theta t} \right) +$$
$$+ c_{\theta t} \cdot 2 \cdot 10^{-3} \frac{\rho |\boldsymbol{V}|^2}{\nu}(1 - F_{\theta t})(Re_{\theta t} - \tilde{R}e_{\theta t})$$ (Turb-SST.24)

$$F_{length}=\begin{cases}39.8189-1.1927\cdot10^{-2}\,\tilde{R}e_{\theta t}-1.32567\cdot10^{-4}\,\tilde{R}e_{\theta t}^{2} & \tilde{R}e_{\theta t}<400\\ 263.404-1.23939\,\tilde{R}e_{\theta t}+1.94548\cdot10^{-3}\,\tilde{R}e_{\theta t}^{2}-1.01695\cdot10^{-6}\,\tilde{R}e_{\theta t}^{3} & 400\le\tilde{R}e_{\theta t}<596\\ 0.6788-3\cdot10^{-4}\,\tilde{R}e_{\theta t} & 596\le\tilde{R}e_{\theta t}<1200\\ 0.3188 & \tilde{R}e_{\theta t}\ge1200\end{cases}$$ (Turb-SST.25)

$$F_{length}=F_{length}\left(1-F_{sublayer}\right)+40\cdot F_{sublayer}$$ (Turb-SST.26)

$$F_{sublayer}=exp\left[-\left(2.5\,Re_{\omega}\right)^{2}\right]$$ (Turb-SST.27)

$$F_{onset}=max\left[0,\quad F_{onset2}-F_{onset3}\right]$$ (Turb-SST.28)

$$F_{onset3}=max\left[0,\quad 1-\left(0.4\cdot Re_{t}\right)^{3}\right]$$ (Turb-SST.29)

$$F_{onset2}=min\left[2,\quad max\left(F_{onset1},F_{onset1}^{4}\right)\right]$$ (Turb-SST.30)

$$F_{onset1}=\frac{Re_{v}}{2.193\cdot Re_{\theta c}}$$ (Turb-SST.31)

$$Re_{\theta c}=\begin{cases}-3.96035+1.0120656\,\tilde{R}e_{\theta t}-8.6823\cdot10^{-4}\,\tilde{R}e_{\theta t}^{2}+\\ \quad+6.96506\cdot10^{-7}\,\tilde{R}e_{\theta t}^{3}-1.74105\cdot10^{-10}\,\tilde{R}e_{\theta t}^{4} & \tilde{R}e_{\theta t}\le1870\\ 308.23+0.518\,\tilde{R}e_{\theta t} & \tilde{R}e_{\theta t}\ge1870\end{cases}$$ (Turb-SST.32)

$$F_{turb}=exp\left[-\left(0.25\,Re_{t}\right)^{4}\right]$$ (Turb-SST.34)

$$F_{\theta t}=min\left[1,\quad max\left[F_{wake}\,exp\left(-\left(y/\delta\right)^{4}\right),\quad 1-\left(\frac{\gamma-1/c_{e2}}{1-1/c_{e2}}\right)^{2}\right]\right]$$ (Turb-SST.35)

$$F_{wake}=exp\left[-\left(10^{-5}\,Re_{\omega}\right)^{2}\right]$$ (Turb-SST.36)

$$\delta=375\frac{\sqrt{W}\,y\,\nu}{U^{2}}\tilde{R}e_{\theta t}$$ (Turb-SST.37)

$$Re_{\theta t}=\begin{cases}F\left(\lambda_{\theta}\right)\cdot\left[1173.51-589.428\cdot Tu+0.2196\cdot Tu^{-2}\right] & Tu\le1.3\\ F\left(\lambda_{\theta}\right)\cdot331.5\cdot\left[Tu-0.5658\right]^{-0.671} & Tu>1.3\end{cases}$$ (Turb-SST.38)

$$Tu=100\frac{\sqrt{2k/3}}{|\boldsymbol{V}|}$$ (Turb-SST.39)

$$F\left(\lambda_{\theta}\right)=\begin{cases}1+\left[12.986\cdot\lambda_{\theta}+123.66\cdot\lambda_{\theta}^{2}+405.689\cdot\lambda_{\theta}^{3}\right]exp\left(-\left(Tu/1.5\right)^{1.5}\right) & \lambda_{\theta}\le0\\ 1+0.275\cdot\left[1-exp\left(-35\cdot\lambda_{\theta}\right)\right]\cdot exp\left(-2Tu\right) & \lambda_{\theta}>0\end{cases}$$ (Turb-SST.40)

$$\lambda_\theta = \frac{\theta^2}{\nu}\frac{d|\boldsymbol{V}|}{ds}$$ (Turb-SST.41)

$$\theta = \frac{\tilde{R}e_{\theta t}\cdot\nu}{|\boldsymbol{V}|}$$ (Turb-SST.42)

$$\frac{d|\boldsymbol{V}|}{ds} = i_x\frac{\partial|\boldsymbol{V}|}{\partial x} + i_y\frac{\partial|\boldsymbol{V}|}{\partial y} + i_z\frac{\partial|\boldsymbol{V}|}{\partial z}$$ (Turb-SST.43)

$$\frac{\partial|\boldsymbol{V}|}{\partial x} = i_x\frac{\partial V_x}{\partial x} + i_y\frac{\partial V_y}{\partial x} + i_z\frac{\partial V_z}{\partial x}$$ (Turb-SST.44)

$$\frac{\partial|\boldsymbol{V}|}{\partial y} = i_x\frac{\partial V_x}{\partial y} + i_y\frac{\partial V_y}{\partial y} + i_z\frac{\partial V_z}{\partial y}$$ (Turb-SST.45)

$$\frac{\partial|\boldsymbol{V}|}{\partial z} = i_x\frac{\partial V_x}{\partial z} + i_y\frac{\partial V_y}{\partial z} + i_z\frac{\partial V_z}{\partial z}$$ (Turb-SST.46)

$$\boldsymbol{i} = \frac{\boldsymbol{V}}{|\boldsymbol{V}|}$$ (Turb-SST.47)

Interaction with model SST $k-\omega$ is performed by means of correction of the source term of the equation for k. Introduce notation

$$D_k = \beta^* k\omega\left(1 + \xi\left(max\left(M_t^2, M_{t0}^2\right) - M_{t0}^2\right)\right)$$ (Turb-SST.48)

Then the k - equation in the original SST model takes form:

$$\frac{\partial(\rho k)}{\partial t} + \nabla(\rho\boldsymbol{V}k) = \nabla\left(\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right) + \rho P_k - \rho D_k$$ (Turb-SST.49)

When transition model $\gamma - \tilde{R}e_{\theta t}$ is turned on, the generation and dissipation terms of this equation transform:

$$P_k \to \gamma_{eff}\cdot P_k$$ (Turb-SST.50)

$$D_k \to \mathbf{min}\left[1, \quad \mathbf{max}\left[0.1, \quad \gamma_{eff}\right]\right]D_k$$ (Turb-SST.51)

Quantity γ_{eff} is computed as follows:

$$\gamma_{eff} = max\left[\gamma, \gamma_{sep}\right]$$ (Turb-SST.52)

$$\gamma_{sep} = F_{\theta t}\cdot min\left[2, \quad s_1\cdot F_{reattach}\cdot max\left[0, \quad \frac{Re_v}{3.235\,Re_{\theta c}} - 1\right]\right]$$ (Turb-SST.53)

$$F_{reattach} = exp\left[-\left(Re_t/20\right)^4\right]$$ (Turb-SST.54)

Quantity $F_{\theta t}$ is determined by expression (Turb-SST.35). Quantity $Re_{\theta c}$ is determined by expression (Turb-SST.38). Function F_1 also transforms:

$$F_1 \to max[F_3, \quad F_1] \qquad \text{(Turb-SST.55)}$$

$$F_3 = exp\left[-\left(Re_y / 120\right)^8\right] \qquad \text{(Turb-SST.56)}$$

The following auxiliary Reynolds numbers are used in transition model $\gamma - \tilde{R}e_{\theta t}$:

$$Re_t = \frac{\rho k}{\mu \omega} \qquad \text{(Turb-SST.57)}$$

$$Re_\omega = \frac{\omega y^2}{\nu} \qquad \text{(Turb-SST.58)}$$

$$Re_v = \frac{\sqrt{S}\, y^2}{\nu} \qquad \text{(Turb-SST.59)}$$

$$Re_y = \frac{\sqrt{k}\, y}{\nu} \qquad \text{(Turb-SST.60)}$$

Constants of model $\gamma - \tilde{R}e_{\theta t}$ are:

$$\sigma_f = 1, \quad \sigma_{\theta t} = 2 \qquad \text{(Turb-SST.61)}$$

$$c_{a1} = 2, \quad c_{a2} = 0.06 \qquad \text{(Turb-SST.62)}$$

$$c_{\varepsilon 1} = 1, \quad c_{\varepsilon 2} = 50 \qquad \text{(Turb-SST.63)}$$

$$c_{\theta t} = 0.03, \quad s_1 = 2 \qquad \text{(Turb-SST.64)}$$

These constants present in the interface. User may change them. Parameters λ_θ , Tu , $Re_{\theta t}$ are limited in the code:

$$-0.1 \le \lambda_\theta \le 0.1 \qquad \text{(Turb-SST.65)}$$

$$Tu \ge 0.027 \qquad \text{(Turb-SST.66)}$$

$$Re_{\theta t} \ge 20 \qquad \text{(Turb-SST.67)}$$

Quantity Tu , determined by expression (Turb-SST.39) and entering relationship (Turb-SST.38), is limited from below as follows:

$$Tu \ge 0.027 \qquad \text{(Turb-SST.68)}$$

Model $\gamma - \tilde{R}e_{\theta t}$ is described in Ref. [18]. Another expression for quantity γ_{sep} is used in Ref. [19]:

$$\gamma_{sep} = F_{\theta t} \cdot min\left[5, \quad s_1 \cdot F_{reattach} \cdot max\left[0, \quad \frac{Re_v}{2.193\, Re_{\theta c}} - 1 \right] \right]$$ (Turb-SST.69)

$$F_{reattach} = exp\left[-\left(Re_t / 15 \right)^4 \right]$$ (Turb-SST.70)

The authors of this article claim that these parameters allow predicting correct position of transition initiated by separation of laminar boundary layer. The three parameters which differ in articles [18] and [19] are brought into the FlowVision interface - see Turbulence, Parameters. The minimum value of the turbulence intensity (Turb-SST.66) is also brought into the interface.

Zero diffusion flux is specified as boundary condition for quantities γ and $\tilde{R}e_{\theta t}$ at solid surfaces. Constant values for γ and $\tilde{R}e_{\theta t}$ are assigned to the inlets: $\gamma = 1$, $\tilde{R}e_{\theta t}$ is computed by formulas (Turb-SST.38). Since at an inlet $d|\boldsymbol{V}| / ds = 0$, these expressions are simplified:

$$Re_{\theta t} = \begin{cases} 1173.51 - 589.428 \cdot Tu + 0.2196 \cdot Tu^{-2} & Tu \le 1.3 \\ 331.5 \cdot \left[Tu - 0.5658 \right]^{-0.671} & Tu > 1.3 \end{cases}$$ (Turb-SST.71)

$$Tu = 100 \cdot I_t$$ (Turb-SST.72)

The same values are automatically set as initial conditions.

10.5.6.3.6 Model SA

Model 'Spalart-Allmaras':

$$\frac{\partial(\rho v_t)}{\partial t} + \nabla(\rho \boldsymbol{V} v_t) = \frac{1}{\sigma_v} \left[\nabla\left((\mu + \rho v_t) \nabla v_t \right) + C_{b2} \rho \left| \nabla v_t \right|^2 \right] + C_{b1} \rho \tilde{\Omega} v_t - C_{w1} \rho f_w \left(\frac{v_t}{y} \right)^2$$ (Turb-SA.1)

$$\mu_t = \rho v_t f_{v1}$$ (Turb-SA.2)

$$\tilde{\Omega} = W + \frac{v_t}{\kappa^2 y^2} f_{v2}$$ (Turb-SA.3)

$$W = 2 \sum_{i,j} W_{ij} W_{ij}$$ (Turb-SA.4)

$$W_{ij} = \frac{1}{2} \left(\frac{\partial V_i}{\partial x_j} - \frac{\partial V_j}{\partial x_i} \right)$$ (Turb-SA.5)

$$C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{1 + C_{b2}}{\sigma_v}$$ (Turb-SA.6)

$$f_w = g \left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right]^{1/6}$$ (Turb-SA.7)

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}$$ (Turb-SA.8)

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \quad \text{(Turb-SA.9)}$$

$$\chi = \frac{v_t}{v} \quad \text{(Turb-SA.10)}$$

$$g = r + C_{w2}\left(r^6 - r\right) \quad \text{(Turb-SA.11)}$$

$$r = \frac{v_t}{\tilde{\Omega}\kappa^2 y^2} \quad \text{(Turb-SA.12)}$$

$$\sigma_v = \frac{2}{3}, \quad C_{b1} = 0.1355, \quad C_{b2} = 0.622, \quad C_{v1} = 7.1, \quad C_{w2} = 0.3, \quad C_{w3} = 2, \quad \kappa = 0.4187 \quad \text{(Turb-SA.13)}$$

See [1] for description of the model.

This model can be used both in low-Reynolds and high-Reynolds calculations.

10.5.6.3.7 Model Sm

Implementation of the Smagorinsky model in *FlowVision* software:

$$\mu_t = \rho\left(C_s h\right)^2 \sqrt{S} \quad \text{(Turb-Sm.1)}$$

$$S = 2\sum_{i,j} S_{ij} S_{ij} = \sum_{i,j}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right)\frac{\partial V_j}{\partial x_i} \quad \text{(Turb-Sm.2)}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right) \quad \text{(Turb-Sm.3)}$$

$$h = \left(\Delta x \Delta y \Delta z\right)^{1/3} \quad \text{(Turb-Sm.4)}$$

Here

$\Delta x, \quad \Delta y, \quad \Delta z$ - Cartesian dimensions of a cell,

C_S - model constant (default value: 0.17).

This is an algebraic model. It does not require integration of convection-diffusion equations. It can be used only in low-Reynolds calculations on a fine grid. In the cells adjacent to walls, $\mu_t = 0$.

See [7] for description of the model.

10.5.6.3.8 Heat turbulence models

The implemented models for turbulent heat transfer are discussed in the given section. They allow calculation a variable turbulent Prandtl number. The models below can be used only together with k-ε turbulence models.

Model KEtetaAKN (Abe, Kondoh, Nagano)

$$\frac{\partial\left(\rho k_\theta\right)}{\partial t} + \nabla\left(\rho V k_\theta\right) = \nabla\left(\rho\left(\alpha + \frac{\alpha_t}{\sigma_{k\theta}}\right)\nabla k_\theta\right) + \rho G_{k\theta} - \rho\varepsilon_\theta \quad \text{(Turb-HeatTransfer.1)}$$

$$\frac{\partial(\rho\varepsilon_\theta)}{\partial t}+\nabla(\rho\mathbf{V}\varepsilon_\theta)=\nabla\left(\rho\left(\alpha+\frac{\alpha_t}{\sigma_{\varepsilon\theta}}\right)\nabla\varepsilon_\theta\right)+C_{P1}f_{P1}\rho\frac{\varepsilon_\theta}{2k_\theta}G_{k\theta}+C_{P2}\rho\frac{\varepsilon_\theta}{k}v_tP_k-$$
$$-C_{D1}f_{D1}\rho\frac{\varepsilon_\theta^2}{2k_\theta}-C_{D2}f_{D2}\rho\frac{\varepsilon\varepsilon_\theta}{k}$$
(Turb-HeatTransfer. 2)

$$\alpha_t=C_\lambda f_\lambda\frac{k^2}{\varepsilon}\equiv\frac{\lambda_t}{\rho C_p}\equiv\frac{v_t}{Pr_t}$$
(Turb-HeatTransfer. 3)

$$G_{k\theta}=\alpha_t(\nabla T)^2$$
(Turb-HeatTransfer. 4)

$$P_k=v_tS$$
(Turb-HeatTransfer. 5)

$$S=2\sum_{i,j}S_{ij}S_{ij}=\sum_{i,j}\left(\frac{\partial V_i}{\partial x_j}+\frac{\partial V_j}{\partial x_i}\right)\frac{\partial V_j}{\partial x_i}$$
(Turb-HeatTransfer. 6)

$$S_{ij}=\frac{1}{2}\left(\frac{\partial V_i}{\partial x_j}+\frac{\partial V_j}{\partial x_i}\right)$$
(Turb-HeatTransfer. 7)

$$Pr_t=\frac{C_\mu f_\mu}{C_\lambda f_\lambda}$$
(Turb-HeatTransfer. 8)

$$f_\lambda=\left\{\frac{2R}{C_m+R}+3\frac{(\nu\nu\varepsilon)^{3/4}}{k^{3/2}}\frac{(2R)^{1/2}}{Pr}f_d\right\}\times$$
$$\times\left\{1-exp\left(-\frac{y^+_{Kolm}}{14}\right)\right\}\left[1-exp\left(-\frac{Pr^{1/2}y^+_{Kolm}}{14}\right)\right]$$
(Turb-HeatTransfer. 9)

$$f_{D1}=\left\{1-exp\left(-\frac{y^+_{Kolm}}{A_{D1}}\right)\right\}^2$$
(Turb-HeatTransfer. 10)

$$f_{D2}=\frac{C_{\varepsilon2}f_2-1}{C_{D2}}\left\{1-exp\left(-\frac{y^+_{Kolm}}{A_{D2}}\right)\right\}^2$$
(Turb-HeatTransfer. 11)

$$f_d=exp\left\{-\left(\frac{Re_t}{200}\right)^2\right\}$$
(Turb-HeatTransfer. 12)

$$f_2=1-0.3exp\left\{-\left(\frac{Re_t}{6.5}\right)^2\right\}$$
(Turb-HeatTransfer. 13)

$$y^+_{Kolm}=\frac{(\varepsilon\nu)^{1/4}y}{\nu}$$
(Turb-HeatTransfer. 14)

$$Re_t=\frac{k^2}{\varepsilon\nu}$$
(Turb-HeatTransfer. 15)

$$R = \frac{k_\theta/\varepsilon_\theta}{k/\varepsilon}$$ (Turb-HeatTransfer.16)

$$C_\lambda = 0.1 \qquad \sigma_{k\theta} = 1.6 \qquad \sigma_{\varepsilon\theta} = 1.6 \qquad C_{\varepsilon 2} = 1.9$$

$$C_{P1} = 1.9 \qquad C_{P2} = 0.6 \qquad C_{D1} = 2 \qquad C_{D2} = 0.9$$ (Turb-HeatTransfer.17)

$$C_m = 0.5 \qquad A_{D1} = 1 \qquad A_{D2} = 5.7$$

See [3] for description of the model.

Model KEtetaS&S (Sommer, So)

$$\frac{\partial(\rho k_\theta)}{\partial t} + \nabla(\rho V k_\theta) = \nabla\left(\rho\left(\alpha + \frac{\alpha_t}{\sigma_{k\theta}}\right)\nabla k_\theta\right) + \rho G_{k\theta} - \rho\varepsilon_\theta$$ (Turb-HeatTransfer.18)

$$\frac{\partial(\rho\varepsilon_\theta)}{\partial t} + \nabla(\rho V \varepsilon_\theta) = \nabla\left(\rho\left(\alpha + \frac{\alpha_t}{\sigma_{\varepsilon\theta}}\right)\nabla\varepsilon_\theta\right) + C_{P1} f_{P1} \rho \frac{\varepsilon_\theta}{2k_\theta} G_{k\theta} + C_{P2}\rho\frac{\varepsilon_\theta}{k} v_t P_k - C_{D1}\rho\frac{\varepsilon_\theta^2}{2k_\theta} - C_{D2}\rho\frac{\varepsilon\varepsilon_\theta}{k} + \rho\xi_{\varepsilon\theta}$$ (Turb-HeatTransfer.19)

$$\xi_{\varepsilon\theta} = f_{w,\varepsilon,}\left[(C_{D1} - 4)\frac{\varepsilon_\theta^2}{2k_\theta} + C_{D2}\frac{\varepsilon\varepsilon_\theta}{k} - \frac{(\varepsilon_\theta - \alpha 2k_\theta/y^2)^2}{2k_\theta} + (2 - C_{P1})\frac{\varepsilon_\theta}{2k_\theta}G_{k\theta}\right]$$ (Turb-HeatTransfer.20)

$$\alpha_t = C_\lambda f_\lambda k\sqrt{\frac{k\cdot 2k_\theta}{\varepsilon\cdot\varepsilon_\theta}} \equiv \frac{\lambda_t}{\rho C_p} \equiv \frac{v_t}{Pr_t}$$ (Turb-HeatTransfer.21)

$$Pr_t = \frac{C_\mu f_\mu}{C_\lambda f_\lambda}\sqrt{\frac{1}{2R}}$$ (Turb-HeatTransfer.22)

$$f_\lambda = f_{w,\varepsilon,}\frac{C_{1\lambda}}{Re_t^{1/4}} + \left[1 - exp\left(\frac{y^+}{A^+}\right)\right]^2$$ (Turb-HeatTransfer.23)

$$f_{w,\varepsilon,} = exp\left[-\left(\frac{Re_t}{80}\right)^2\right]$$ (Turb-HeatTransfer.24)

$$y^+ = \frac{C_\mu^{1/4} k^{1/2} y}{v}$$ (Turb-HeatTransfer.25)

$$C_\lambda = 0.11 \qquad \sigma_{k\theta} = 1.227 \qquad \sigma_{\varepsilon\theta} = 1.227$$

$$C_{P1} = 1.8 \qquad C_{P2} = 0.72 \qquad C_{D1} = 2.2 \qquad C_{D2} = 0.8$$ (Turb-HeatTransfer.26)

$$C_{1\lambda} = 0.1 \qquad A^+ = 30$$

Quantities $G_{k\theta}$, P_k, Re_t and R are respectively determined by Eqs. (Turb-HeatTransfer.4), (Turb-HeatTransfer.5), (Turb-HeatTransfer.15), (Turb-HeatTransfer.16). The original model see in [8]. Eqs (Turb-HeatTransfer.18) - (Turb-HeatTransfer.26) describe a simplified version of model [8].

Model LMS

This is a special model, for which a separate license is required.

10.5.6.3.9 Distance to wall

The distance to the nearest wall (y) is required in models **KEAKN**, **KEFV**, **SST**, **SA**, **KEtetaAKN**, **KEtetaS&S**. It is computed as follows:

$$y = -|\nabla \Phi| + \sqrt{|\nabla \Phi|^2 + 2\Phi} \qquad \text{(Turb-WallDistance.1)}$$

Here

Φ is potential of the distance to the nearest wall

The distance potential is found as a result of integration of the Poisson equation:

$$\nabla^2 \Phi = -1 \qquad \text{(Turb-WallDistance.2)}$$

Boundary conditions for this equation are set automatically:

$\Phi = 0$ - at walls (Turb-WallDistance.3)

$\nabla_y \Phi = 0$ - at other boundaries (Turb-WallDistance.4)

10.5.6.3.10 Models of wall functions

Using wall functions allows user not to resolve by grid the part of the turbulent boundary layer adjacent to the considered solid surface. This essentially reduces the required computational resources.

Two models of wall functions are implemented in *FlowVision*:

- **WFFV**
- **WFS**

Selection of the wall function model is determined by the **Standard wall functions** parameter in the **Turbulence** group of parameters in the advanced settings of Solver.

These models allow accounting for the pressure gradient (both positive and negative). By default, the account of the pressure gradient is off. To toggle it on, select **Preprocessor > Phases > Phase #i > Physical processes > Turbulence > WF: P gradient** = **Yes**. It should be mentioned that when a large pressure gradient occurs near a body, the account of the pressure gradient can case failure of the program. For this reason, it is recommended

1) to toggle on the account of the pressure gradient after a smooth pressure distribution is formed,
2) to toggle off the account of the pressure gradient when the grid is rebuilt near the body.

Both models of wall functions can be used in nonequilibrium regime.

This section has individual numeration of equations.

10.5.6.3.10.1 Model WFFV

To make the FV wall function model (**WFFV**) active, select **Solver > Advanced settings > Turbulence > Standard wall functions** = **No** (the default value). The model uses smooth profiles of all the flow characteristics from a wall to the first cell center.

Profile of tangential velocity

The model assumes a combined profile of the tangential component of velocity [9]:

$$V_x = V_{x,1} + V_{x,2} \cdot sign(\nabla_x P) \qquad \text{(Turb-WF.1)}$$

Here

$V_{x,1}(y)$ - profile of tangential velocity developing on a plate in infinite space (the pressure gradient is zero),

$V_{x,2}(y)$ - profile of tangential velocity developing in the separation point (the viscous stress is zero),

$sign(\nabla_x P)$ - sign of the projection of the pressure gradient onto the direction of the tangential velocity vector in the first cell center.

The first profile is determined by the following relationship:

$$\frac{\partial V_{x,1}}{\partial y} = \frac{-\mu + \sqrt{\mu^2 + 4\rho l_1^2 \tau_{wx}}}{2\rho l_1^2} \quad \text{(Turb-WF.2)}$$

$$l_1 = \kappa_1 y \cdot \left(1 - exp\left(-y_\tau^+ \frac{1}{A_1}\right)\right) \quad \text{(Turb-WF.3)}$$

$$y_\tau^+ = \frac{\rho u_\tau y}{\mu} \quad \text{(Turb-WF.4)}$$

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad \text{(Turb-WF.5)}$$

In the limit $y \to 0$

$$l_1 = \frac{\rho u_\tau y^2}{\mu} \frac{\kappa_1}{A_1} \quad \text{(Turb-WF.6)}$$

$$\frac{\partial V_{x,1}}{\partial y} = \frac{\tau_{wx}}{\mu} \quad \text{(Turb-WF.7)}$$

In the limit $y \to \infty$

$$l_1 = \kappa_1 y \quad \text{(Turb-WF.8)}$$

$$V_{x,1} = \frac{u_\tau}{\kappa_1} ln\, E_1 y_\tau^+ \quad \text{(Turb-WF.9)}$$

$$\frac{\partial V_{x,1}}{\partial y} = \frac{u_\tau}{\kappa_1 y} \quad \text{(Turb-WF.10)}$$

The second profile is determined by the following relationship:

$$\frac{\partial V_{x,2}}{\partial y} = \frac{-\mu + \sqrt{\mu^2 + 4\rho l_2^2 |\nabla_x P| y}}{2\rho l_2^2} \quad \text{(Turb-WF.11)}$$

$$l_2 = \kappa_2 y \sqrt{y_p^+} \cdot \left(1 - exp\left(-y_p^+ \frac{1}{A_2}\right)\right) \quad \text{(Turb-WF.12)}$$

$$y_p^+ = \frac{\rho u_p y}{\mu} \quad \text{(Turb-WF.13)}$$

$$u_p = \left(\frac{\mu |\nabla_x P|}{\rho^2}\right)^{1/3} \quad \text{(Turb-WF.14)}$$

Here

$\nabla_x P$ is projection of the pressure gradient onto the direction of the tangential velocity in the first cell center,

In the limit $y \to 0$

$$l_2 = \frac{y_p}{\alpha}\left(y_p^+\right)^{3/2}\frac{1}{A_2} \quad \text{(Turb-WF.15)}$$

$$\frac{\partial V_{x,2}}{\partial y} = \frac{|\nabla_x P| y}{\mu} \quad \text{(Turb-WF.16)}$$

In the limit $y \to \infty$

$$l_2 = \kappa_2 y\sqrt{y_p^+} \quad \text{(Turb-WF.17)}$$

$$V_{x,2} = \frac{u_p}{\kappa_2} ln\, E_2 y_p^+ \quad \text{(Turb-WF.18)}$$

$$\frac{\partial V_{x,2}}{\partial y} = \frac{u_p}{\kappa_2 y} \quad \text{(Turb-WF.19)}$$

It is stated in [9] that the tangential velocity profile developing in the separation point has a logarithmic part:

$$V_{x,2}^+ \equiv \frac{V_{x,2}}{u_p} = \alpha\, ln\, y_p^+ + \beta = \frac{1}{\kappa_2} ln\, E_2 y_p^+ \quad \text{(Turb-WF.20)}$$

Profiles $V_{x,1}^+\left(y_p^+\right)$ and $V_{x,2}^+\left(y_p^+\right)$ are defined by interface parameters **WF: kappa1**, **WF: A1**, **WF: kappa2**, **WF: A2**. They are computed one time and stored in the program in tabulated form. In each near-wall cell, quantity

$$V_{x,2} = u_p V_{x,2}^+\left(y_p^+\right) \quad \text{(Turb-WF.21)}$$

is computed using the known pressure gradient. Then quantity

$$V_{x,1} = V_x - V_{x,2} \cdot sign\left(\nabla_x P\right) \quad \text{(Turb-WF.22)}$$

is calculated. After that the dynamic velocity (u_τ) is found from the following transcendental equation.

$$V_{x,1} = u_\tau V_{x,2}^+\left(y_\tau^+\right) \quad \text{(Turb-WF.23)}$$

It determines the viscous stress, corresponding to the given V_x , at the wall:

$$\tau_w = \rho u_\tau^2 \quad \text{(Turb-WF.24)}$$

Boundary condition for momentum equation

When the matrix is built for implicit integration of the momentum equation (**Solver > Advanced settings > Numerical method > Type of scheme = Implicit**), the viscous force exerted onto the given wall is submitted in the following form:

$$\boldsymbol{\tau}_{w,n} = a_V\left[\left(\boldsymbol{V}_{x,cell}^{n+1} - \boldsymbol{V}_{x,w}^{n+1}\right) - \boldsymbol{b}_V\right] \quad \text{(Turb-WF.25)}$$

$$a_V = \frac{1}{J_{V1}} \quad \text{(Turb-WF.26)}$$

$$\boldsymbol{b}_V = J_{V2} \cdot \nabla_x P^{n+1} \quad \text{(Turb-WF.27)}$$

Integrals J_{V1} and J_{V2} are determined by:

$$J_{V1} = \int_0^y \frac{dy}{\mu^n + \mu_t^n} \quad \text{(Turb-WF.28)}$$

$$J_{V2} = \int_0^y \frac{ydy}{\mu^n + \mu_t^n} \quad \text{(Turb-WF.29)}$$

Profile of turbulent viscosity

The turbulent viscosity profile is determined by

$$\mu_t = \frac{1}{2}\left(-\mu + \sqrt{\mu^2 + 4\rho l^2 (\tau_{wx} + \nabla_x P \cdot y)}\right) \quad \text{(Turb-WF.30)}$$

$$l^2 = l_1^{\,2} + sign(\nabla_x P) \cdot l_2^{\,2} \quad \text{(Turb-WF.31)}$$

In the limit $y \to 0$

$$\mu_t = \frac{\rho l^2}{\mu}(\tau_{wx} + \nabla_x P \cdot y) \approx \frac{\rho l^2}{\mu}\tau_{wx} \quad \text{(Turb-WF.32)}$$

In the limit $y \to \infty$

$$\mu_t = l\sqrt{\rho(\tau_{wx} + \nabla_x P \cdot y)} \quad \text{(Turb-WF.33)}$$

Profile of turbulent energy

In turbulence models **KES**, **KEAKN**, **KENL**, the profile of turbulent energy (k) is determined by the following relationship:

$$k_{cell} = \frac{u_\tau^2 + sign(\nabla_x P) \cdot \frac{u_p^3}{\nu} y}{0.3 + \frac{1}{0.003 \cdot \left(y_\tau^+\right)^{3.5}}} \quad \text{(Turb-WF.34)}$$

In model KEFV, the k profile is determined by

$$k_{cell} = \frac{u_\tau^2 + sign(\nabla_x P) \cdot \frac{u_p^3}{\nu} y}{0.3} \quad \text{(Turb-WF.35)}$$

In turbulence model SST, the profile of turbulent energy is determined as follows:

$$k_{cell} = \frac{u_\tau^2}{0.3 + \frac{1}{0.002 \cdot \left(y_\tau^+\right)^{3.5}}} \quad \text{- WF: profile W+ = 1} \quad \text{(Turb-WF.36)}$$

$$k_{cell} = \frac{\left(min\left(\frac{y_\tau^+}{3}, 1 \right) \cdot u_\tau \right)^2}{0.3}$$ - **WF: profile W+ = 2** (Turb-WF.37)

$$k_{cell} = \frac{u_\tau^2}{0.3}$$ - **WF: profile W+ = 3** (Turb-WF.38)

The k value is automatically set in the first cell center if **Boundary conditions > Wall** (template) **> Wall interaction > Phase #i** = **Wall functions, equilibrium**.

Profile of dissipation rate of turbulent energy

In turbulence models **KES**, **KEAKN**, **KEFV**, **KENL**, the profile of the dissipation rate of turbulent energy (ε) is determined by the following relationships:

$$\varepsilon_{cell} = \frac{u_k^3}{l_\varepsilon}$$ (Turb-WF.39)

$$l_\varepsilon = \kappa y \left(1 - exp\left(-y_k^+ / A_\varepsilon\right)\right)$$ (Turb-WF.40)

$$y_k^+ = \frac{\rho u_k y_{cell}}{\mu}$$ (Turb-WF.41)

$$u_k = C_\mu^{1/4} \left(\frac{u_\tau^2 + sign\left(\nabla_x P\right) \cdot \frac{u_p^3}{\nu} y}{0.3} \right)^{1/2}$$ (Turb-WF.42)

Profile of specific dissipation rate of turbulent energy

In turbulence model **SST**, the profile of the specific dissipation rate of turbulent energy (ω) is determined by the following relationships:

$$\omega_{cell} = \sqrt{\omega_{vis}^2 + \omega_{turb}^2}$$ - **WF: profile W+ = 1** (Turb-WF.43)

$$\omega_{cell} = exp\left(\Gamma\right) \cdot \omega_{vis} + exp\left(1/\Gamma\right) \cdot \omega_{turb}$$ - **WF: profile W+ = 2** (Turb-WF.44)

$$\omega_{cell} = \omega_{turb}$$ - **WF: profile W+ = 3** (Turb-WF.45)

$$\omega_{vis} = \frac{80 \cdot a_{vis} \cdot \nu}{y_{cell}^2}$$ - **W_vis = 1** (Turb-WF.46)

$$\omega_{vis} = \frac{\omega_w}{\left(1 + y_{cell} \sqrt{\frac{\omega_w}{80 \cdot a_{vis} \cdot \nu}}\right)^2}$$ - **W_vis = 2** (Turb-WF.47)

$$\omega_w = \begin{cases} \dfrac{u_\tau^2}{\nu}\dfrac{2500}{\left[max\left(h_s^+, 2.4\left(y_\tau^+\right)^{0.85}\right)\right]^2} & h_s^+ < 25 \\ \dfrac{u_\tau^2}{\nu}\dfrac{100}{h_s^+} & h_s^+ \geq 25 \end{cases} \quad \text{(Turb-WF.48)}$$

$$\omega_{turb} = \frac{u_\tau}{0.3\kappa_1 y_{cell}} \quad \text{(Turb-WF.49)}$$

$$\Gamma = -\frac{a\left(y^+\right)^4}{1+by^+} \quad \text{(Turb-WF.50)}$$

$$a = 0.01, \quad b = 5 \quad \text{(Turb-WF.51)}$$

The method of combining 'viscous' and 'turbulent' profiles of ω determined by Eq. (Turb-WF.43) is taken from [10]. Method (Turb-WF.44) is taken from [11]. 'Viscous' profile of ω determined by Eq. (Turb-WF.46) is taken from [12]. The wall value of ω defined by (Turb-WF.48) is taken from [6].

The first version of model **WFFV** was published in [13].

10.5.6.3.10.2 Model WFS

The 'Standard' model of wall functions (**WFS**), implemented in *FlowVision*, contains different ideas taken from the open literature. To make active this model, select **Solver > Advanced settings > Turbulence > Standard wall functions** = **Yes**. Traditional wall functions do not allow for the pressure gradient. The current implementation is extended by possibility to take into account the pressure gradient as it is done in model **WFFV**.

Profile of tangential velocity

Similarly to model **WFFV**, model **WFS** also assumes a combined profile of the tangential component of velocity:

$$V_x = V_{x,1} + V_{x,2} \cdot sign\left(\nabla_x P\right) \quad \text{(Turb-WF.52)}$$

$$V_{x,1}^+ = \left[\left(y_\tau^+\right)^4 + \left(\frac{1}{\kappa_1} ln\, E_1 y_\tau^+\right)^{-4}\right]^{-1/4} \quad \text{(Turb-WF.53)}$$

$$V_{x,2}^+ = \left[\left(\frac{1}{2} y_p^{+2}\right)^{-4} + \left(\frac{1}{\kappa_2} ln\, E_2 y_p^+\right)^{-4}\right]^{-1/4} \quad \text{(Turb-WF.54)}$$

$$\kappa_1 = 0.41, \quad E_1 = 9, \quad \kappa_2 = 0.2, \quad E_2 = 4.953 \quad \text{(Turb-WF.55)}$$

The method of combining 'viscous' and 'turbulent' profiles of $V_{x,1}^+$ determined by Eq. (Turb-WF.52) is taken from [14].

The algorithm of computing dynamic viscosity and viscous stress is determined by Eqs. (Turb-WF.21) - (Turb-WF.24).

Boundary condition for momentum equation

When the matrix is built for implicit integration of the momentum equation (**Solver > Advanced settings > Numerical method > Type of scheme = Implicit**), the viscous force exerted onto the given wall is submitted in the following form:

$$\boldsymbol{\tau}_{w,n} = a_V \left[\left(\boldsymbol{V}_{x,cell}^{n+1} - \boldsymbol{V}_{x,w}^{n+1} \right) - \boldsymbol{b}_V \right] \quad \text{(Turb-WF.56)}$$

$$a_V = \frac{\rho u_\tau}{V_{x,1}^+} \quad \text{(Turb-WF.57)}$$

$$\boldsymbol{b}_V = V_{x,2}^+ \frac{\nu}{\rho u_p^2} \nabla_x P^{n+1} \quad \text{(Turb-WF.58)}$$

Profile of turbulent viscosity

The turbulent viscosity profile is determined by

$$\mu_t = max\left(\mu_{min}, \mu_{t1} + \mu_{t2} \cdot sign\left(\nabla_x P\right)\right) \quad \text{(Turb-WF.59)}$$

$$\frac{\mu_{t1}}{\mu} = \frac{\nu_{t1}}{\nu} = \begin{cases} 0 & \text{at } y_\tau^+ \le 3 \\ \kappa_1 y_\tau^+ \frac{30}{27}\left(y_\tau^+ - 3\right) & \text{at } 3 < y_\tau^+ \le 30 \\ \kappa_1 y_\tau^+ & \text{at } y_\tau^+ > 30 \end{cases} \quad \text{(Turb-WF.60)}$$

$$\frac{\mu_{t2}}{\mu} = \frac{\nu_{t2}}{\nu} = \begin{cases} 0 & \text{at } y_p^+ \le 1 \\ \kappa_2 y_p^+ \frac{30}{29}\left(y_\tau^+ - 1\right) & \text{at } 1 < y_p^+ \le 30 \\ \kappa_2 y_p^+ & \text{at } y_p^+ > 30 \end{cases} \quad \text{(Turb-WF.61)}$$

Profile of turbulent energy

In turbulence models **KES**, **KEAKN**, **KEFV**, **KENL**, the profile of turbulent energy = parabola + constant:

$$k_{cell} = \frac{\left(min\left(\frac{y_\tau^+}{y_{LinLog}^+}, 1 \right) \cdot u_\tau \right)^2}{0.3} \quad \text{(Turb-WF.62)}$$

Here

$y_{LinLog}^+ \approx 11$ - value of y_τ^+, at which the linear and logarithmic branches of profile $V_{x,1}^+$ meet.

This value is determined by the following equation

$$y_{LinLog}^+ = \frac{1}{\kappa_1} ln\left(E_1 y_{LinLog}^+\right) \quad \text{(Turb-WF.63)}$$

In turbulence model **SST**, the profile of turbulent energy is determined by Eqs. (Turb-WF.36) - (Turb-WF.38). The k value is automatically set in the first cell center if **Boundary conditions > Wall** (template) **> Wall interaction > Phase #i** = **Wall functions, equilibrium**.

Profile of dissipation rate of turbulent energy

In turbulence models **KES**, **KEAKN**, **KEFV**, **KENL**, the profile of the dissipation rate of turbulent energy = constant + hyperbola:

$$\varepsilon_{cell} = \frac{u_\tau^3}{\kappa_1} min\left(\frac{1}{y_{cell}}, \frac{u_\tau}{3\nu} \right) \quad \text{(Turb-WF.64)}$$

Profile of specific dissipation rate of turbulent energy

In turbulence model **SST**, the profile of the specific dissipation rate of turbulent energy (ω) is determined by Eqs. (Turb-WF.43) - (Turb-WF.51).

10.5.6.3.10.3 Temperature profiles

The heat flux from a wall and the wall temperature are related by the following expression:

$$J_{q,y} = a_T \left(T_w - T_c - b_T \right) \quad \text{(Turb-WF.65)}$$

If the wall temperature is specified, this expression is used for calculation of the heat flux. If the heat flux is specified, it is used for calculation of the wall temperature. Three options for computing quantities a_T and b_T are implemented in *FlowVision*. This options are submitted below.

Temperature profiles

Solver > Advanced settings > Turbulence > WF: profile T+ = 1:

$$T_{vis}^+ = Pr\, y_\tau^+ \quad \text{(Turb-WF.66)}$$

$$T_{turb}^+ = Pr_t \left[\frac{1}{\kappa_1} ln(E_1 y_\tau^+) + P_{fit} \right] \quad \text{(Turb-WF.67)}$$

$$P_{fit} = 9.24 \left[\left(\frac{Pr}{Pr_t} \right)^{3/4} - 1 \right] \left[1 + 0.28\, exp\left(-0.007 \frac{Pr}{Pr_t} \right) \right] \quad \text{(Turb-WF.68)}$$

Solver > Advanced settings > Turbulence > T+ Kader = No:

$$T^+ \equiv \frac{\rho C_p u_\tau \left(T_w - T_c \right)}{J_{q,y}} = min\left(T_{vis}^+, T_{turb}^+ \right) \quad \text{(Turb-WF.69)}$$

Solver > Advanced settings > Turbulence > T+ Kader = Yes:

$$T^+ \equiv \frac{\rho C_p u_\tau \left(T_w - T_c \right)}{J_{q,y}} = exp(\Gamma) \cdot T_{vis}^+ + exp\left(\frac{1}{\Gamma} \right) \cdot T_{turb}^+ \quad \text{(Turb-WF.70)}$$

$$\Gamma = -\frac{a \left(Pr \cdot y_\tau^+ \right)^4}{1 + b \cdot Pr^3 y_\tau^+} \quad \text{(Turb-WF.71)}$$

$$a = 0.01, \quad b = 5 \quad \text{(Turb-WF.72)}$$

The method for weighting the 'viscous' and 'turbulent' profiles T^+, determined by Eqs. (Turb-WF.70)-(Turb-WF.72), is taken from [11]. Quantities a_T and b_T are computed as follows:

$$a_T = \frac{\rho C_p u_\tau}{T^+} \quad \text{(Turb-WF.73)}$$

$$b_T = \frac{1}{2}\left(V_{x,c}^2 - V_{x,w}^2\right)\frac{\mu_c + \frac{1}{2}\mu_{t,c}}{\lambda_c + \frac{1}{2}\mu_{t,c}\frac{C_{p,c}}{Pr_{t,c}}} - \frac{\tau_w |V_{x,w}|}{a_T} \quad \text{(Turb-WF.74)}$$

Solver > Advanced settings > Turbulence > WF: profile T+ = 2:

$$T_{turb}^+ = 2.12 \cdot ln\left(Pr\, y_\tau^+\right) + \left[3.85 \cdot Pr^{1/3} - 1.3\right]^2 \quad \text{(Turb-WF.75)}$$

Expression (Turb-WF.75) is taken from [10].

Solver > Advanced settings > Turbulence > T+ Kader = No:

See Eq. (Turb-WF.69).

Solver > Advanced settings > Turbulence > T+ Kader = Yes:

See Eq. (Turb-WF.70).

Quantities a_T and b_T are computed by formulae (Turb-WF.73) and (Turb-WF.74) respectively.

Solver > Advanced settings > Turbulence > WF: profile T+ = 3:

$$a_T = \frac{1}{J_{T1}} \quad \text{(Turb-WF.76)}$$

Liquid:

$$b_T = J_{T2} R_h \quad \text{(Turb-WF.77)}$$

$$R_h = \sum_{i,j=1}^{3} \tau_{ij} e_{ij} + \rho_c \varepsilon_c \quad \text{(Turb-WF.78)}$$

Gas:

$$b_T = \left\langle \frac{\mu + \mu_t}{\lambda + \lambda_t} \right\rangle \frac{1}{2}\left(V_{x,c}^2 - V_{x,w}^2\right) - J_{T2} R_H \quad \text{(Turb-WF.79)}$$

$$R_H = \rho_c V_{x,c} F_{x,c} + \rho_c V_{y,c} F_{y,c} \quad \text{(Turb-WF.80)}$$

In expression (Turb-WF.79),

$\left\langle \frac{\mu + \mu_t}{\lambda + \lambda_t} \right\rangle$ is the integral average value of the ratio of effective viscosity to effective thermal conductivity over interval $0 < y < y_{cell}$.

Integrals J_{T1} and J_{T2} are determined by the following expressions:

$$J_{T1} = \int_0^{yc} \frac{dy}{\lambda + \lambda_t} \quad \text{(Turb-WF.81)}$$

$$J_{T2} = \int_0^{yc} \frac{ydy}{\lambda + \lambda_t} \qquad \text{(Turb-WF.82)}$$

Turbulent thermal conductivity is computed as follows:

$$\lambda_t = C_{p,c}\mu_t / Pr_t \qquad \text{(Turb-WF.83)}$$

$$Pr_t = Pr_{t,user} + \frac{d_{Pr}}{\mu_t^+ Pr} \qquad \text{(Turb-WF.84)}$$

$$\mu_t^+ = \mu_t / \mu \qquad \text{(Turb-WF.85)}$$

Expression for the turbulent Prandtl number (Turb-WF.84) is taken from [15].

Eqs (Turb-WF.76)-(Turb-WF.85) determine the heat model **WFFV**. The model takes into account the influence of the pressure gradient on turbulent thermal conductivity through turbulent viscosity.

Boundary condition for enthalpy

When the matrix is built for implicit integration of the energy equation (**Solver > Advanced settings > Numerical method > Type of scheme = Implicit**), the heat flux from the given wall is submitted in the following form:

$$J_{q,y} = a_T \left[\frac{1}{C_p^n} h_c^{n+1} + T_w^{n+1} - T_c^{n+1} + \frac{1}{C_p^n} h_c^n - b_T \right] \quad \text{- liquid} \qquad \text{(Turb-WF.86)}$$

$$J_{q,y} = a_T \left[\frac{1}{C_p^n} H_c^{n+1} + T_w^{n+1} - T_c^{n+1} + \frac{1}{C_p^n} \left(h_c^n + \frac{1}{2} \left(V_c^{n+1} \right)^2 \right) - b_T \right] \quad \text{- gas} \qquad \text{(Turb-WF.87)}$$

10.5.6.3.10.4 Nom-equilibrium regime

The computational scheme, in which the value of turbulent energy in a near-wall cell (k_{cell}) is found from the convection-diffusion equation, is traditionally called *non-equilibrium regime* of using wall functions or *non-equilibrium wall functions*. In this regime, the following condition is automatically set on the wall:

$$\left. \frac{\partial k}{\partial y} \right|_w = 0 \qquad \text{(Turb-WF.88)}$$

To specify *non-equilibrium regime* of using wall functions, select **Subregions > Subregion #i > Boundary conditions > Wall** (template) **> Wall interaction > Phase #i = Wall functions, nonequilibrium.** Algebraic relationships (Turb-WF.34)-(Turb-WF.38), (Turb-WF.62) for k_{cell} are not used in this scheme. The generation term in a near-wall cell is computed as follows:

$$P_k = \nu_t \left(\frac{\partial V_x}{\partial y} \right)^2 = \nu_t \left(\frac{\tau_{wx} + \nabla_x P \cdot y}{\mu + \mu_t} \right)^2 \qquad \text{(Turb-WF.89)}$$

In turbulence models **KES**, **KEAKN**, **KEFV**, **KENL**, the value of the dissipation rate of turbulent energy in a near-wall cell is determined as follows:

$$\varepsilon_{cell} = \frac{u_k^3}{l_\varepsilon} \quad \text{- WFFV} \qquad \text{(Turb-WF.90)}$$

$$\varepsilon_{cell} = \frac{u_k^3}{\kappa_1} min\left(\frac{1}{y_{cell}}, \frac{u_k}{3\nu}\right)$$ - **WFS** (Turb-WF.91)

In turbulence model **SST**, the value of the specific dissipation rate of turbulent energy in a near-wall cell (ω_{cell}) is determined by Eqs. (Turb-WF.43)-(Turb-WF.45), in which

$$\omega_{turb} = \frac{u_k}{0.3\kappa_1 y}$$ (Turb-WF.92)

In Eqs. (Turb-WF.90)-(Turb-WF.92)

$$u_k = C_\mu^{1/4} k_{cell}^{1/2}$$ (Turb-WF.93)

In turbulence model **SA**, the non-equilibrium and non-equilibrium computational schemes coincide, because this model does not include equation for k .

10.5.6.3.10.5 Account of wall roughness

The resistance exerted by a rough wall onto a flow is higher than the resistance exerted by a smooth wall. Roughness may be uniform and non-uniform. Roughness is also distinguished by the shape of knobs, by their sizes, by the distance between the knobs, etc. It is difficult to use real geometrical characteristics of roughness in mathematical models. Therefore term "equivalent sand roughness" was introduced in 1930-s. This is the size of sand grains yielding the same resistance coefficient as the original roughness. N. Nikuradze used sand roughness in his experiments on definition of the hydraulic resistance of rough pipes.

Three flow regimes are distinguished in rough pipes [16]:

1. Flow without manifestation of the pipe roughness:

   $0 \le h_s^+ \le 5$.

   The resistance depends only on the Reynolds number. All the knobs are submerged in the viscous sub-layer.
2. Transient flow:

   $5 \le h_s^+ \le 70$

   The resistance depends both on the Reynolds number and on the roughness. The knobs partly overhang the viscous sub-layer.
3. Flow with complete manifestation of the pipe roughness:

   $h_s^+ > 70$

The resistance is mostly determined by the roughness. The height of the knobs essentially exceeds the viscous sub-layer thickness.

In the submitted models of wall functions, the effect of wall roughness on a flow is allowed for by introducing an effective viscosity [17] and formal replacing the molecular viscosity by the effective one:

$$\mu \to \mu_{eff} = \mu + a_h \rho u_\tau h_s$$ (Turb-WF.94)

The default value of the roughness constant is $a_h = 0.276$. It provides smooth transition from profile

$$V_{x,1}^+ = \frac{1}{0.41} ln\left(\frac{y_\tau^+}{1 + a_h h_s^+}\right)$$ (Turb-WF.95)

to profile

$$V_{x,1}^+ = \frac{1}{0.41} ln \frac{y_\tau^+}{h_s^+} + 8.5$$ (Turb-WF.96)

Here

$h_s^+ = \frac{\rho u_\tau h_s}{\mu}$ - dimensionless "equivalent sand roughness" of the given wall.

In Eqs. (Turb-WF.50) and (Turb-WF.71), the effect of roughness is taken into account by correction of coefficients a and b:

$$a = 0.01 \cdot c \quad \text{(Turb-WF.97)}$$

$$b = 5/c \quad \text{(Turb-WF.98)}$$

$$c = exp\left(a_h \cdot h_s^+\right) \quad \text{(Turb-WF.99)}$$

10.5.6.4 Boundary conditions

The current section describes boundary conditions for variables $k, \varepsilon, \omega, \nu_t$, corresponding to different boundary templates.

- Models **KES**, **KEAKN**, **KEFV**, **KENL** require conditions for k and ε at a boundary.
- Model **SST** requires conditions for k and ω.
- Model **SA** requires a condition for ν_t.

This section has individual numeration of equations.

10.5.6.4.1 Template 'Wall'

The boundary conditions for turbulent variables are determined by the selected model of flow-wall interaction:

Wall interaction	Variable	Boundary condition
Wall functions, equilibrium	k, k_θ	**Value in cell near wall**
	$\varepsilon, \varepsilon_\theta$	**Value in cell near wall**
	ω	**Value in cell near wall**
	ν_t	**Fixed value**
Wall functions, nonequilibrium	k, k_θ	**Zero gradient**
	$\varepsilon, \varepsilon_\theta$	**Value in cell near wall**
	ω	**Value in cell near wall**
	ν_t	**Fixed value**
No wall functions	k, k_θ	**Fixed value**
	$\varepsilon, \varepsilon_\theta$	**Auto-condition for turb. dissipation**
	ω	**Value in cell near wall**
	ν_t	**Fixed value**

Value in cell near wall

If a k-ε model (**KES**, **KEAKN**, **KEFV**, or **KENL**) is selected, user specifies nothing. In these models, the given boundary condition for variables k and ε is used only in high-Reynolds calculations. In model **SST**, is also set in low-Reynolds calculations for variable ω (**Subregions > Subregion #i > Boundary conditions > Wall** (template) **> Wall interaction > Phase #i = No wall functions**). The implemented options for specifying variables k, ε, ω in the centers of the near-wall cells for high-Reynolds calculations are considered in section **Models of wall functions**. The implemented methods of computing variable ω in the centers of the near-wall cells for low-Reynolds calculations are determined by Eqs. (Turb-WF.41), (Turb-WF.42), **Model WFFV**.

Zero gradient

User specifies nothing. This condition assumes zero diffusion flux of the corresponding scalar quantity:

$$\left.\frac{\partial k}{\partial y}\right|_w = 0$$ - **KES**, **KEAKN**, **KEFV**, **KENL**, **SST** (Turb-BC.1)

$$\left.\frac{\partial k_\theta}{\partial y}\right|_w = 0$$ - **KEtetaAKN**, **KEtetaS&S** (Turb-BC.2)

At a wall, the condition is equivalent to condition **Symmetry**.

Fixed value

User specifies nothing. The following conditions are automatically set at a wall

$k_w = 0$ - **KES**, **KEAKN**, **KEFV**, **KENL**, **SST** (Turb-BC.3)

$k_{\theta,w} = 0$ - **KEtetaAKN**, **KEtetaS&S** (Turb-BC.4)

$v_{t,w} = 0$ - **SA** (Turb-BC.5)

Auto-condition for turb. dissipation

User specifies nothing. The following conditions are automatically set at a wall

$$\varepsilon_w = 2v\left(\frac{\sqrt{k_{cell}}}{y_{cell}}\right)^2$$ - **KES**, **KEAKN**, **KEFV**, **KENL** (Turb-BC.6)

$$\varepsilon_{\theta,w} = \frac{\lambda}{\rho C_p}\left(\frac{\sqrt{2k_{\theta,cell}}}{y_{cell}}\right)^2$$ - **KEtetaAKN**, **KEtetaS&S** (Turb-BC.7)

10.5.6.4.2 Template 'Symmetry'

Symmetry

User specifies nothing. This condition assumes zero diffusion flux of the corresponding scalar quantity:

$$\left.\frac{\partial k}{\partial y}\right|_b = 0$$ - **KES**, **KEAKN**, **KEFV**, **KENL**, **SST** (Turb-BC.8)

$$\left.\frac{\partial \varepsilon}{\partial y}\right|_b = 0$$ - **KES**, **KEAKN**, **KEFV**, **KENL** (Turb-BC.9)

$$\left.\frac{\partial k_\theta}{\partial y}\right|_b = 0$$ - **KEtetaAKN**, **KEtetaS&S** (Turb-BC.10)

$$\left.\frac{\partial \varepsilon_\theta}{\partial y}\right|_b = 0$$ - **KEtetaAKN**, **KEtetaS&S** (Turb-BC.11)

$$\left.\frac{\partial \omega}{\partial y}\right|_b = 0$$ - **SST** (Turb-BC.12)

$$\left.\frac{\partial v_t}{\partial y}\right|_b = 0$$ - **SA** (Turb-BC.13)

The convective flluxes are also zero:

$$\rho V_n f\big|_b = 0$$ (Turb-BC.14)

Here

$$f = k, \varepsilon, k_\theta, \varepsilon_\theta, \omega, \nu_t$$

10.5.6.4.3 Template 'Inlet/Outlet'

Pulsations

This condition is intended for variable k (**KES**, **KEAKN**, **KEFV**, **KENL**, **SST**). User specifies an average intensity of turbulent pulsations I_t at the given surface:

$$I_t = \sqrt{\frac{2}{3}k} / |\boldsymbol{V}_{\text{inl}}|$$ (Turb-BC.15)

Here

$\boldsymbol{V}_{\text{inl}}$ - local velocity of the flow entering the computational domain

The turbulent energy at the inlet is computed by the following formula:

$$k_b = \frac{3}{2}\left(I_t |\boldsymbol{V}_{inl}|\right)^2$$ (Turb-BC.16)

This value is not used in the case of outflow through the given surface. After changing turbulence model **KE** => **SST** and **SST** => **KE** in the interface, the specified value of I_t is kept (all 4 implemented k-ε turbulence models are assumed here).

Turbulent scale

This condition is intended for variables

ε - **KES**, **KEAKN**, **KEFV**, **KENL**

ω - **SST**

User specifies an average linear scale of turbulent eddies L_t. The dissipation rate of turbulent energy at the inlet is computed by

$$\varepsilon_b = C_\mu^{3/4}\frac{k_b^{3/2}}{L_t}$$ (Turb-BC.17)

The specific dissipation rate of turbulent energy at the inlet is computed by

$$\omega_b = \frac{k_b^{1/2}}{C_\mu^{1/4} L_t}$$ (Turb-BC.18)

The computed value ε_b or ω_b is not used in the case of outflow through the given surface. After changing turbulence model **KE** => **SST** and **SST** => **KE** in the interface, the specified value of L_t is not kept: value 0 is automatically set for L_t right after the change of model.

Value

This condition is intended for variables

k - **KES**, **KEAKN**, **KEFV**, **KENL**, **SST**

ε - **KES**, **KEAKN**, **KEFV**, **KENL**

k_θ - **KEtetaAKN**, **KEtetaS&S**

ε_θ - **KEtetaAKN**, **KEtetaS&S**

ω - **SST**

ν_t - **SA**

User specifies a value for the corresponding variable. This value is not used in the case of outflow through the given surface. After changing turbulence model **KE** => **SST** and **SST** => **KE** in the interface, the specified value of k is kept, the specified value of ε or ω is not kept: value 0 is automatically set right after the change of model. After changing turbulence model **KE** => **SA**, **SA** => **KE**, **SST** => **SA**, **SA** => **SST** in the interface, value 0 is automatically set for all the listed variables.

Unit for TR_GAMMA

This condition automatically sets fixed value "1" for variable **TR_GAMMA** at the inlet. Relevant when the **SST** k-ω turbulence model is selected and **Transition** = **Yes**.

Auto-condition for TR_RETETA

This condition automatically computes (via the specified turbulence intensity) the value for variable **TR_RETETA** at the inlet

$$Re_{\theta t} = \begin{cases} 1173.51 - 589.428 \cdot Tu + 0.2196 \cdot Tu^{-2} & Tu \le 1.3 \\ 331.5 \cdot \left[Tu - 0.5658\right]^{-0.671} & Tu > 1.3 \end{cases}$$

$$Tu = 100 \cdot I_t$$

and sets this value at the inlet. Relevant when the **SST** k-ω turbulence model is selected and **Transition** = **Yes**.

10.5.6.4.4 Template 'Free outlet'

Zero gradient

User specifies nothing. This condition assumes zero diffusion flux of the corresponding scalar quantity - see Template 'Symmetry'. The convective flux of this quantity is not zero in general case. It is determined by the fluid velocity at this boundary.

Pulsations, Turbulent scale

See Template 'Inlet/Outlet'.

This conditions have sense when inflow of the fluid is possible through the given outlet. In the case of outflow, the specified values of the turbulent quantities are not used.

Value

See Template 'Inlet/Outlet'.

This condition has sense when inflow of the fluid is possible through the given outlet. In the case of outflow, the specified values of the turbulent quantities are not used.

10.5.6.4.5 Template 'Connected'

If user selects **Connection type** = **Conjugate temperature** in window **Create binder condition**, template **Wall** is automatically set on the given boundary.

Selection **Connection type** =

- **Conjugate all variables**
- **Periodic surface**
- **Sliding surface**

causes automatic setting of the corresponding condition of matching all the variables on the two surfaces (user specifies nothing).

10.5.6.4.6 Template 'Nonreflecting'

Pulsations, Turbulent scale

See Template 'Inlet/Outlet'.

This conditions have sense when inflow of the fluid is possible through the given outlet. In the case of outflow, the specified values of the turbulent quantities are not used.

Value

See Template 'Inlet/Outlet'.

This condition has sense when inflow of the fluid is possible through the given outlet. In the case of outflow, the specified values of the turbulent quantities are not used.

10.5.6.5 Initial conditions

This section has individual numeration of equations.

For all the models user specifies initial **Pulsations** I_t (initial turbulence intensity) and initial **Turbulent scale** L_t. The other turbulent quantities are computed automatically according to the selected turbulence model:

$k = \frac{3}{2}\left(I_t |V_{ini}|\right)^2$ - **KES**, **KEAKN**, **KEFV**, **KENL**, **SST** (Turb-InitCond.1)

$\varepsilon = C_\mu^{3/4} \frac{k^{3/2}}{L_t}$ - **KES**, **KEAKN**, **KEFV**, **KENL** (Turb-InitCond.2)

$\omega = \frac{k^{1/2}}{C_\mu^{1/4} L_t}$ - **SST** (Turb-InitCond.3)

$v_t = C_\mu^{1/4} \sqrt{\frac{3}{2}} |V_{ini}| I_t L_t$ - **SA** (Turb-InitCond.4)

Here

V_{ini} is local initial velocity of the flow

10.5.6.6 References

1. Wilcox D. C. (1994) "Turbulence modeling for CFD", *DCW Industries, Inc.,* 460 p.

2. Abe K., Kondoh T., Nagano Y. (1994) “A New Turbulence Model for Predicting Fluid Flow and Heat Transfer in Separating and Reattaching flows-I. Flow Field Calculation”, *Int. Journal of Heat and Mass Transfer, Vol. 37, No. 1*, pp. 139-151.

3. Abe K., Kondoh T., Nagano Y. (1995) “A New Turbulence Model for Predicting Fluid Flow and Heat Transfer in Separating and Reattaching flows-II. Thermal Field Calculation”, *Int. Journal of Heat and Mass Transfer, Vol. 38, No. 8*, pp. 1467-1481.

4a. Acton, M.J., Lenci, G., Baglietto, E., 2015. Structure-Based Resolution of Turbulence for Sodium Fast Reactor Thermal Striping Applications, in: *Submitted to the 16th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-16)*.

4b. Baglietto, E., Ninokata, H., 2007. Improved Turbulence Modeling for Performance Evaluation of Novel Fuel Designs. *Nucl. Technol. 158, 237–248*.

4c. Baglietto, E., Ninokata, H., 2006. Anisotropic Eddy Viscosity Modeling for Application to Industrial Engineering Internal Flows. *Int. J. Transp. Phenom. 8, 85–101*.

5. Menter F.R., Kuntz M., and Langtry R. (2003) "Ten years of Industrial Experience with the SST Turbulence Model", *Turbulence, Heat and Mass Transfer 4, Begell House, Inc.,* 8 p.

6. Hellsten A. (1998) "Some improvements in Menter's SST k-w turbulence model," *AIAA Paper 98-2554*, 11 p.

7. Garbaruk A.V., Strelets M.H., Shur M.L. (2012) "Turbulence simulation in calculations of complex flows", *Textbook, St. Petersburg, Publishing house of Polytechnical University*, 88 p.

8. Sommer T. P., So R. M. C. and Zhang H. S (1993) “Near-wall variable Prandtl turbulence model for compressible flows,” *AIAA Journal Vol. 31,* pp. 27-35.

9. Tsan-Hsing Shih, Povinelli Louis A., Nan-Suey Liu, Potapczuk Mark G., Lumley J.L. (1999) "A Generalized Wall Function," *NASA/TM - 1999 - 209398*, 16 p.

10. Menter F., Carregal Ferreira J., Esch T., Konno B. (2003) "The SST Turbulence Model with Improved Wall Treatment for Heat Transfer Predictions in Gas Turbines" *Proceedings of the International Gas Turbine Congress 2003 Tokyo November 2-7*, 7 p.

11. Kader B.A. (1981) "Temperature and concentration profiles in fully turbulent boundary layers," *International Journal of Heat and Mass Transfer, Vol. 24, No. 9,* pp.1541-1544.

12. Smirnov E. M., Zaitsev D. K., “Modification of Wall Boundary Conditions for Low-Re k-Turbulence Models Aimed at Grid Sensitivity Reduction”, *European Conference for Aerospace Sciences, July 4-7th 2005, Moscow (EUCASS 2005), CD-ROM proceedings, ID 2.09.06*, 7p.

13. Zhluktov S.V., Aksenov A.A., Kharchenko S.A.., Moskalev I.V., Sushko G.B., Shishaeva A.S. (2010) "Simulation of separated flows in software FlowVision-HPC" Computational methods and programming*, Vol. 11, No.2, pp*. 76-87.

14. Grotjans H., Menter F.R . (1998) "Wall functions for industrial applications" *Proc. of Computational Fluid Dynamics'98, ECCOMAS, ed. by K.D. Papailiou - U.K., Wiley: Chichester, Vol. 1 (2)*, pp 1112-1117

15. Kays, W.M. (1994) “Turbulent Prandtl Number - Where Are We”, *Transactions of the ASME, Vol. 116*, pp. 284-295.

16. Schlichting H. (1979) "Boundary-layer theory", *McGraw-Hill, Inc.,* 817 p.

17. Souders D. T. and Hirt C.W. (2002) "Modeling Roughness Effects in Open Channel Flows," *Flow Science Technical Note (FSI-02-TN60),* 10 p.

18. Langtry R. B., Menter F. R. (2009) "Correlation-Based Transition Modeling for Unstructured Parallelized Computational Fluid Dynamics Codes," AIAA Journal, Vol. 47, No. 12, pp. 2894-2906.

19. Corral R and Gisbert F. "Prediction of Separation-induced Transition using a Correlation-based Transition Model," AIAA Paper 2009-3666, 11 p.

10.5.7 Phase transfer

Properties of the element **Phases > Phase #N > Physical processes** include the **Phase transfer** parameter. This parameter is read-only and by default is set as **(none)**. After loading the second continuous **Phase** into the folder **Models > Model #N > Phases**, the value of this parameter will automatically change to **VOF model**.

Motion of the inter-phase surface between two continuous **Phases** is simulated by the **VOF** method ("Volume Of Fluid"). This method assumes solution of the convective transfer equation of the continuous **Phase**, which locates at the first position in the list of **Phases** in the **Model**. At each time step the program calculates the fraction of the computational cell filled by this **Phase**. Suppose that a problem with two immiscible continuous **Phases** is simulated. The cells, in which the **VOF** variable equals **1**, are filled with the first **Phase**. The cells, in which **VOF=0**, are filled with the second **Phase**.

The inter-phase surface between the **Phases** goes through the cells with intermediate values of the **VOF** variable. Configuration of this surface is updated at each time step.

See also: Two phase media with an inter-phase surface (VOF model).

This section has individual numeration of equations, references, and illustrations.

10.5.7.1 Notations

Notation	Physical quantity	Name in *FlowVision*	Dimension
F	Variable **VOF**	**PhaseVolume**	
$\boldsymbol{n}$	Normal to boundary (always directed inside the computational domain) (vector)		
P	Relative pressure	**Pressure**	Pa
$\boldsymbol{V}$	Velocity (vector)	**Velocity**	m s^{-1}
y	Distance to wall	**Distance to wall**	m
ρ	Density	**Density**	kg m^{-3}
σ	Surface tension coefficient	**Surface tension** (in properties of a **Substance**) **SurfTension, value** (in properties of an element **Model #i > Phase interaction > Continuum-Continuum** or **> Continuum-Vacuum**)	N m^{-1}
σ_{ref}	The reference (at temperature T_{ref}) value of the surface tension coefficient of the inter-phase surface	**SurfTension, value** (in properties of an element **Model #i > Phase interaction > Continuum-Continuum** or **> Continuum-Vacuum**) when this parameter is set by a constant	
σ_T	The temperature coefficient of the surface tension $\sigma_T=-d\sigma/dT$	**dSigma/dT**	$N \cdot m^{-1} \cdot K^{-1}$
Q_{VOF}	Source of the **VOF** variable (caused, for example, by phase transition at crystallization or by cavitation, etc.). See equation (PhTr.1).		s^{-1}

Indices:
1 - the continuous phase, for which Eq. (PhTr.1) is solved,
2 - second continuous phase,
cell - value in cell center,
user - value specified by user.

10.5.7.2 Parameters

Parameters in window **Preprocessor > Phases > Phase #i > Physical processes > Phase transfer**:

Parameter	Description
Math. model	A model for the given physical process [1)] (This field is not accessible for editing by user and has the value **VOF model**)
Time step coefficient	For the physical process **Phase transfer** for *continuous* **Phases** this parameter is not applied and can not be edited by user, and has its default value **1**. Use the **Solver > Advanced settings > Multiphase C > Relaxation** parameter instead. For other physical processes this parameter is used to specify an individual time step (Time step for a specific physical process = common time step τ x value of the parameter **Time step coefficient**). Specifying different time steps for different processes sometimes allows acceleration of convergence to steady-state solution.

[1] The value of this parameter is set by the program automatically when two continuous **Phases** are entered into a **Model**.

Parameters in window **Preprocessor > Models > Model #i > Phase interaction > Continuum-Continuum**:

Parameter	Permissible values	Description
Math. model	**Continuum-Continuum**	A model for the given process [1] (The position is inaccessible for editing.)
Phase0	**Phase #j** [2]	One from the **Continuous Phases**, present in **Model #i** (The position is inaccessible for editing.)
Phase1	**Phase #k** [2]	The second from the given pair of **Continuous Phases**. (The position is inaccessible for editing.)
SurfTension, auto	**Yes** | **No**	**Yes** = the surface tension at the contact surface is computed as the modulus of the difference of the surface tensions of the two contacting **Phases**. The **Phase** surface tension is determined by property **Surface tension** of the first **Substance** in folder **Preprocessor > Phases > Phase #i > Substances**.[3] **No** = surface tension is specified in the following position (**SurfTension, value**).
SurfTension, value	determined by the problem (the default value: **0**)	A value for the surface tension at the contact surface σ, [N m^{-1}]. The position is accessible for editing if **SurfTension, auto = No**. The **SurfTension, value** parameter can be set by a constant, a formula or a table. When **SurfTension, value** is set by a constant, you specify here the reference value σ_{ref}, which is applied at the reference temperature T_{ref}, while the actual value of the surface tension coefficient is calculated with use of the temperature coefficient σ_T, which is set by the **dSigma/dT** parameter (see below). The surface tension coefficient is calculated using the formula: $\sigma = \sigma_{ref} - \sigma_T(T_{\text{abs}} - T_{\text{ref}}) = \sigma_{ref} - \sigma_T T$
dSigma/dT	determined by the problem (the default value: **0**)	$\sigma_T = -d\sigma/dT$, the temperature coefficient of the surface tension, [N·m^{-1}·K^{-1}]. This parameter is available when **SurfTension, value** is set by a constant (see above).
Blackness	determined by the problem (the default value: **1**)	The blackness (emissivity) of the free inter-phase surface
Mass transfer	**Yes** | **No**	The blackness (emissivity) of the free inter-phase surface

[1] A model for **Phase** interaction is automatically set up corresponding to the types of the **Phases** present in the **Model**.

[2] The number of **Continuous Phases** in the given **Model** must be > 1.

[3] Property **Surface tension** is not defined for gaseous **Substances** (**Preprocessor > Phases > Phase #i > Substances > Substance #i > Aggregative state = Gas**). The implemented algorithm sets value 0 for the surface tension of gases.

Parameters of process Phase transfer in the advanced settings of Solver

These parameters are specified in the advanced settings of Solver (group of parameters **Multiphase C**).

Parameter	Permissible values	Description
Phase conservative	**Yes** | **No**	**Yes** – The mass conservation law is strictly fulfilled in phase transfer. **No** – The mass conservation law is not strictly fulfilled in phase transfer.[1]
Relaxation	From **0** to **1**. (the default value is **1**)	This parameter allows reduction of the time step used in the integration of the phase transfer equation. If **Relaxation = 0**, then a) the inter-phase boundary is "frozen", b) the zero-flow and slip conditions are set on it (like in BC **Symmetry**).
CFL for VOF source	Numerical value	The CFL (Courant-Friedrichs-Lewy) number for the phase transfer's program block. It is calculated based on the source of the solid phase formed due to the crystallization (from the substance of the dispersed phase). This CFL value determines the step of motion of the inter-phase surface due to action of the source Q_{VOF} in the equation (PhTr.1) for the **VOF** variable. When this parameter is zero or negative (**CFL for VOF source** ≤ **0**), the time step for the phase transfer will be calculated based on the value of the **Relaxation** parameter. Positive value of this parameter (**CFL for VOF source** > **0**) determines the fraction, at which volume of the solid phase can increase/decrease in any cell per one (common for the whole simulation)time step. Thus, at one iteration, volume of the solid phase in any cell cannot increase at one common step more then (**CFL for VOF source**)×(volume of the cell). When icing of an aircraft is simulated, it is recommended to set the **CFL for VOF source** in the range from **0.1** to **0.4**.
Use for time step	**Yes** | **No**	**Yes** – The velocity of the inter-phase boundary is taken into account in calculation of the time step. **No** – The velocity of the inter-phase boundary is not taken into account in calculation of the time step.
Use VOF source for time step	**Yes** | **No**	**Yes** – the problem's time step is calculated *without* taking into account

Parameter	Permissible values	Description
		the motion of the inter-phase surface, caused by Q_{VOF}. In this case the value of the **CFL for VOF source** parameter is ignored and motion of the free surface is done with resulting time step with value of the **Relaxation** parameter taken into account. **No** – the problem's time step is calculated *with* taking into account the motion of the inter-phase surface, caused by Q_{VOF}. In this case motion of the free surface, caused due to action of the source, is simulated with taking into account the **CFL for VOF source** parameter.
VOF-particles	**Yes** | **No**	**Yes** – *VOF-particles* are identified, their migration is simulated. **No** – *VOF-particles* are not identified, their migration is not simulated.
VOF level	Determined by the problem (the default value is **0.0001**)	If **VOF** < **VOF level,** value 0 is forcibly assigned to variable **VOF**. If **VOF** > 1 - **VOF level,** value 1 is forcibly assigned to variable **VOF**.[2)]
Use conservative speed	**Yes** | **No**	The program will use **Conservative velocity**, which is based on speed of fluid on cell faces, in transport equation for **VOF**.

[1)] Strict conservation of mass is usually required when fluid motion is simulated in an enclosed volume. Otherwise (for instance, in simulation of water flow around a ship), it is recommended to specify **Phase conservative** = **No**.

[2)] The procedure of checking out and rounding-off the values of variable **VOF** is performed in all the computational cells.

10.5.7.3 Equations

Transfer of the **Phase** boundary is described by the equation for the volume fraction of this (continuous) phase in a computational cell ('Volume Of Fluid' = VOF). Consider a flow of two immiscible fluids. The heavy one we shall call Liquid, the light one - Gas. Variable **VOF** takes values from **0** (in Gas) to **1** (in Liquid) - see Figure 1. A cell where 0 < **VOF** < 1 contains a contact surface. In the solver, this surface is represented by a set of polygons.

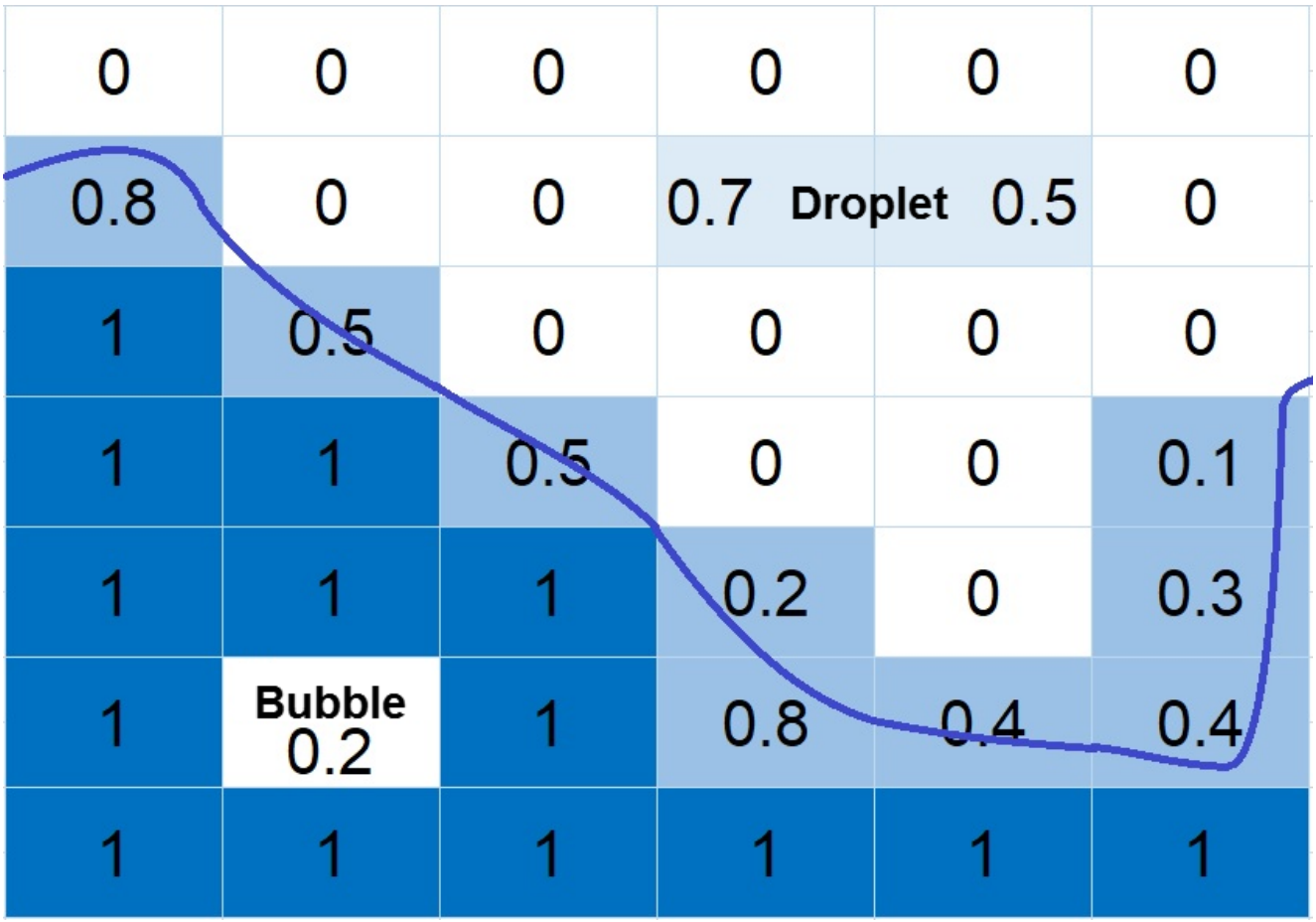


Figure 1. Approximation of contact / free surface.

The convection equation for transfer of the contact surface (equation for variable **VOF**) reads:

$$\frac{\partial F}{\partial t} + \boldsymbol{V} \cdot \nabla F = Q_{VOF} \qquad \text{(PhTr.1)}$$

If a nonzero surface tension is specified in the project, the Laplace pressure is taken into account at the surface:

$$P_1 = P_2 + \sigma \cdot div\ \boldsymbol{n} \qquad \text{(PhTr.2)}$$

Here

σ - surface tension,

$\boldsymbol{n}$ - normal to the contact surface.

Surface tension on the inter-phase surface can be not a constant only but depend on temperature, doping concentrations or other parameters of the two-phase flow, so additional tangential component of the viscous force can appear on the inter-phase surface:

$$\tau_{w,1} = \tau_{w,2} + grad(\sigma) \qquad \text{(PhTr.3)}$$

After integrating Eq. (PhTr.1), the contact surface is reconstructed. Then the equations of continuity, momentum, energy, etc., for the continuous phases are solved. The boundary conditions conjugating all the sough-for variables are automatically set at the contact surface. Hence, the implemented algorithm does not require "weighting" the

properties of the contacting continuous phases (like density, viscosity, thermal conductivity, etc.) in the cells crossed by the surface ($0<F_{cell}<1$).

Simulating VOF-particles

A computational grid cannot always resolve local bends of the inter-phase boundary (contact surface). For this reason, cells with **VOF** > 0 may arise in Gas, cells with **VOF** < 1 may arise in Liquid. In *FlowVision*, such things are called *VOF-particles*. In general case, *VOF-particles* migrate over the computational grid. Their motion is simulated in a special way: the Lagrange equations are solved for each particle. VOF-particles affect the flowfield, however, they are not taken into account in computing **Characteristics**. The *VOF-particles* do not participate in heat exchange.

Simulating VOF-particles is turned on by the **Multiphase C > VOF-particles** parameter in the **Advanced settings** of **Solver**.

Visualization of VOF-particles (droplets and bubbles) is set by the **VOF-particles** parameter in properties of the **VOF** layer.

10.5.7.4 Boundary conditions

The current section describes boundary conditions for variable **PhaseVolume** (VOF), corresponding to different boundary templates.

This section has individual numeration of equations.

10.5.7.4.1 Template 'Wall'

Symmetry

User specifies nothing. This boundary condition assumes:

$$\left.\frac{\partial F}{\partial y}\right|_b = 0 \qquad \text{(PhTr-BC.1)}$$

$$\rho V_n F\big|_b = 0 \qquad \text{(PhTr-BC.2)}$$

From Eq. (PhTr-BC.1) it follows that the value of variable VOF at the boundary equals to that in the center of the cell adjacent to the boundary:

$$F_b = F_{cell} \qquad \text{(PhTr-BC.3)}$$

Value

User specifies value of the variable **VOF**:

$$F_b = F_{user} \qquad \text{(PhTr-BC.4)}$$

10.5.7.4.2 Template 'Symmetry'

Symmetry

User specifies nothing - see Template 'Wall' > Symmetry.

10.5.7.4.3 Template 'Inlet/Outlet'

Value

User specifies value of the variable **VOF** similarly as for the **Wall** template.

See Template 'Inlet/Outlet' > Value.

10.5.7.4.4 Template'Free outlet'

Zero gradient

User specifies nothing. This boundary condition assumes:

$$\left.\frac{\partial F}{\partial y}\right|_b = 0 \qquad \text{(PhTr-BC.5)}$$

$$\rho V_n F\big|_b = \left(\rho V_n\right)_{cell} F_{cell} \qquad \text{(PhTr-BC.6)}$$

From Eq. (PhTr-BC.5) it follows that the value of variable **VOF** at the boundary equals to that in the center of the cell adjacent to the boundary:

$$F_b = F_{cell} \qquad \text{(PhTr-BC.7)}$$

Value

User specifies value of the variable **VOF** similarly as for the **Wall** template.

See Template 'Inlet/Outlet' > Value.

10.5.7.4.5 Template 'Connected'

If user selects **Connection type** = **Conjugate temperature** in window **Create binder condition**, template **Wall** is automatically set on the given boundary.

Selection **Connection type** = **Conjugate all variables | Periodic surface | Sliding surface** causes automatic setting of the corresponding condition of matching all the variables on the two surfaces (user specifies nothing).

10.5.7.4.6 Template 'Nonreflecting'

Value

User specifies value of the variable **VOF** similarly as for the **Wall** template.

See Template 'Inlet/Outlet' > Value.

In the **Nonreflecting** template, user specifies the value of the variable **VOF** (F_{user}) outside the computational domain.

10.5.7.5 References

1. Hirt C. W., Nicholls B. D. Volume of fluid (VOF) method for the dynamics of free boundaries // J. Comput. Phys. – 1981. – Vol. 39. – P. 201.*)

*) this work provides a basic approach only.

10.5.8 Processes in the presence of dispersed medium

The physical processes proceeding in a dispersed phase are activated in window **Preprocessor > Phases > Phase #i** (dispersed **Phase Particles** or dispersed **Phase Carcass**) **> Physical processes**. These are:

- **Heat transfer**
- **Phase transfer** (for **Phase Particles** only)
- **Motion** (for **Phase Particles** only)
- **Mass transfer**
- **Crystallization** (for **Phase Particles** only)

Selection **(none)** means that the given process is disabled.

The implemented model of dispersed medium allows solving the following problems:

- flow of liquid/gas in porous carcass
- motion of solid particles in liquid/gas
- motion of droplets in liquid/gas
- motion of bubbles in liquid

- crystallization of **Phase Particles** on the surface of the geometrical model and/or on the solid phase (crystallized **Phase Particles**)
- motion of the film of Substance of **Phase Particles** over the surface of the geometrical model and over the surface of crystallized **Substance** of **Phase Particles**
- destruction of porous carcass
- combustion of pulverized coal

If a continuous phase and a dispersed phase (**Phase Particles** or **Phase Carcass**) are present in folder **Preprocessor > Phases**, you have to select physical models for interaction between the two phases in the window of element **Models > Model #i > Phase interaction > Continuum-particles** or **... > Phase interaction > Continuum-carcass**. These models take into account the exchanges by mass, momentum and energy between the given dispersed and continuous (carrier) phases.

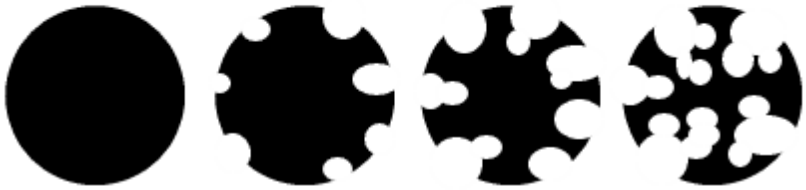
Models for transformation of particles

Two general models for transformation of particles are implemented in *FlowVision*:

- **Variable diameter**, which assumes that the density of a particle remains constant while the diameter changes according to the mass loss of the **Particles Phase**.
- **Constant diameter**, which assumes that the diameter of a particle remains constant while the density changes according to the mass loss of the **Particles Phase**.

The illustration below schematically displays how these models work when sublimation of dispersed particles is simulated:

Model for particles = Variable diameter



Model for particles = Constant diameter

Selection of a model is set by the **Model for particles** parameter in properties of the Mass transfer physical process of the **Particles Phase**.

For Euler (non-coal) particles:

- The **Variable diameter** model simulates natural processes of evaporation/sublimation of liquid or solid substance and vapor condensation on surfaces of particles. Density of the liquid or solid substance is set in properties of the **Substance** (the density can be set either by a constant or a formula or a table).
- The **Constant diameter** model assumes permanency of sizes of particles (so sizes of particles that were specified in initial and on boundary conditions will not change during the simulation). Densities of particles will be calculated by their masses taking into account that diameters are known. Density of the dispersed **Phase** in this model corresponds to a porous particle filled with liquid, which can transfer to the continuous **Phase** (evaporate) or come from continuous **Phase** (condense on surfaces of the particles).

This section has individual numeration of equations, references, and illustrations.

10.5.8.1 Notations

Notation	Physical quantity	Name in *FlowVision*	Dimension
C_p	specific heat at constant pressure	**Specific heat**	J kg^{-1} K^{-1} = = m^2 s^{-2} K^{-1}
C_D	drag coefficient for particles		

Notation	Physical quantity	Name in *FlowVision*	Dimension
d	local diameter of particles	**Diameter**	m
$\boldsymbol{g}$	gravity acceleration (vector)	**Gravity vector**	m s^{-2}
h	thermodynamic enthalpy	**Enthalpy**	m^2 s^{-2}
h_{lat}	latent heat of evaporation / sublimation of **Dispersed Phase**		m^2 s^{-2}
$\hat{\boldsymbol{I}}$	metric (unit) tensor		
$\boldsymbol{J}_q$	specific heat flux (vector)	**HeatFlux**	$W \cdot m^{-2} = kg\ s^{-3}$
k_c	turbulent energy of **Continuous Phase**	**TurbulentEnergy**	m^2 s^{-2}
k_x	coefficient of restitution of tangential component of particles momentum after collision with wall	**Coeff. tang.**	
k_y	coefficient of restitution of tangential component of particles momentum after collision with wall	**Coeff. norm.**	
$\boldsymbol{n}$	internal normal to cell face (vector)		
$P = P_{\text{abs}} - P_{\text{ref}}$	relative static pressure	**Pressure**	Pa = kg m^{-1} s^{-2}
P_{abs}	absolute static pressure		Pa = kg m^{-1} s^{-2}
P_{ref}	reference pressure		Pa = kg m^{-1} s^{-2}
Pr	molecular Prandtl number	**Prandtl**	
Pr_{t}	turbulent Prandtl number	**PrandtlTurb**	
$Sc_{\text{t,d}}$	turbulent Schmidt number, which determines the turbulent diffusion of particles	**SchmidtTurb**	
$\dot{q}_{c\to d}$	specific heat flux from **Continuous Phase** to **Dispersed Phase**		$W \cdot m^{-2} = kg\ s^{-3}$
$Re_d = \frac{\rho_c \lvert \boldsymbol{V}_c - \boldsymbol{V}_d \rvert d}{\mu_c}$	Reynolds number based on particle diameter		
$\hat{\boldsymbol{S}}$	strain rate tensor		s^{-1}
$T = T_{\text{abs}} - T_{\text{ref}}$	relative temperature	**Temperature**	K
T_{abs}	absolute temperature		K
T_{ref}	reference temperature		K
$T_{d,boil}$	absolute boiling temperature of droplets	**Boiling temperature**	K
t	time		s
$\boldsymbol{V}$	velocity (vector)	**Velocity**	m s^{-1}
V_{i}	i-th Cartesian component of velocity		m s^{-1}

Notation	Physical quantity	Name in *FlowVision*	Dimension
$V_{d,x}$	tangential component of relative (to wall) velocity (normal is directed inside computational domain)		m s^{-1}
$V_{d,y}$	normal component of relative (to wall) velocity (normal is directed inside computational domain)		m s^{-1}
β	coefficient of thermal expansion	**Therm. expansion**	K^{-1}
λ	coefficient of molecular thermal conductivity	**Thermal conductivity**	kg m s^{-3} K^{-1}
$\mu_c = \rho\nu_c$	dynamic coefficient of molecular viscosity (**Continuous Phase**)	**Viscosity**	kg m^{-1} s^{-1}
$\mu_{t,d}$	dynamic coefficient of turbulent viscosity (**Dispersed Phase**)		kg m^{-1} s^{-1}
$\mu_{t,c}$	dynamic coefficient of turbulent viscosity (**Continuous Phase**)	**TurbViscosity**	kg m^{-1} s^{-1}
ν_c	kinematic coefficient of molecular viscosity (**Continuous Phase**)		m^2 s^{-1}
$\nu_{t,d}$	kinematic coefficient of turbulent viscosity (**Dispersed Phase**)		m^2 s^{-1}
$\nu_{t,c}$	kinematic coefficient of turbulent viscosity (**Continuous Phase**)		m^2 s^{-1}
ρ	density	**Density**	kg m^{-3}
ε_c	rate of dissipation of turbulent energy	**TurbDissipation**	m^2 s^{-3}
$\hat{\tau}_{eff}$	effective shear stress tensor		Pa = kg m^{-1} s^{-2}
φ_d	volume fraction of cell filed with dispersed phase	**Phase volume**	
φ_c	volume fraction of cell filed with continuous phase	**Phase volume** it is not displayed in the elements of **Initial data** in the project tree and calculated by the formula: $\varphi_c=1-\Sigma\varphi_d$ where $\Sigma\varphi_d$ is the sum of volumes of all dispersed phases presented in the model	
$\Delta h_{evap(subl)}$	latent heat of evaporation of the film or heat of sublimation of the solid phase		m^2 s^{-2}
Δh_{fusion}	latent heat of fusion of the solid phase		
LWC	the liquid water content, which measures the mass of condensed water containing in a unit volume of a cloud	**LWC**	kg/m^3 In the program's user interface this value is specified in g/m^3.

Notation	Physical quantity	Name in *FlowVision*	Dimension
$\dot{m}_d$	specific mass flow of the substance of the dispersed phase on a surface		kg m^{-2} s^{-1}
$\dot{m}_f$	specific mass flow of forming the film from the substance of the dispersed phase		
$\dot{m}_s$	specific mass flow of crystallization of the substance of the dispersed phase or fusion of the solid phase		
w_f	film height (thickness) on the surface	**Film thickness**	m
k_s	equivalent grit roughness of the surface	**Roughness constant**	m

Indices:

c - continuous phase

d- dispersed phase

f - dispersed medium (phase) forming a film on a surface of a geometry or a solid phase

s - solid phase (formed by the substance of a dispersed phase)

n - value at time layer n (in numerical method)

n+1 - value at time layer n+1 (in numerical method)

10.5.8.2 Parameters

To create in the project tree an element **Phase #N**, which will present a dispersed phase of the type **Particles** or **Carcass**, select from the context menu of the folder **Phases** the command **Create particles** or **Create carcass** respectively.

Parameters of the element **Models > Model #i > Phase interaction > Continuum-particles**:

Parameter	Possible values	Description
Math. model	**Continuum-particles**	A model for interaction between the **Phases**, it is set automatically corresponding to types of **Phases**, which are presented in the **Model**. (This field is read-only)
Phase0	**Phase #j**	One of **Continuous Phases**, present in **Model #i** (This field is read-only)
Phase1	**Phase #k**	One of the dispersed **Phases** of the **Particles** type, present in **Model #i** (This field is read-only)
Blackness	Numerical value. The default value is **1**.	The blackness (emissivity) of surface of particles or carcass
P gradient *(this parameter is available only when a model for the drag coefficient of particles* **Cd** *is selected, see below)*	**Yes** | **No**	**Yes** = The force action of the pressure gradient, developing in the carrier (**Continuous**) **Phase**, on particles is taken into account. **No** = The force action of the pressure gradient is not taken into account.
Substance pair	*Folder*	The folder contains the pairs of **Substances** (one from **Phase #j**, another from **Phase #k**), which are different aggregative states of one substance. These pairs are to be set to simulate phase transfers.

Parameter	Possible values	Description
		Specifying a pair of substances is only possible when: • **Motion** in the continuous **Phase** is enabled • **Motion** in the dispersed **Phase** is enabled • **Mass transfer** in the continuous **Phase** is enabled • **Mass transfer** in the dispersed **Phase** is enabled
[n]	*Folder*	The folder contains the **Substances** entering the given **Substance pair**. Pair number **[n]** is generated automatically. The numbering starts from 0.
[n] > Phase0	**Substances** of **Phase0** Default value: **(none)**	**Substance** from **Phase0** entering the given **Substance pair**.
[n] > Phase1	**Substances** of **Phase1** Default value: **(none)**	**Substance** from **Phase1** entering the given **Substance pair**.
SchmidtTurb	Numerical value. Default value: **1**.	Turbulent Schmidt number $Sc_{t,d}$, which determines the turbulent diffusion of particles
C_kPrt	Numerical value. Default value: **0**.	Factor in the generation term of the equation for turbulent energy due to motion of particles
C_ePrt	Numerical value. Default value: **0**.	Factor in the generation term of the equation for the rate of dissipation of turbulent energy due to motion of particles
Repulsion force[*)]	**Yes** | **No**	Account of the force pushing bubbles off a wall. This parameter is available when **Cd ≠ (none)**.
Cw factor	Numerical value. Default value: **0**.	Factor in the model coefficient for the force pushing bubbles off wall. This is coefficient C_W in Eq. (7) in section Equations for particles > Process 'Motion'. This parameter is available when **Repulsion force = Yes**.
Dw	Numerical value. Default value: **10000000000**.	Distance D_W, [m], at which the force pushing bubbles off a wall stops its action. This distance equals to approximately one diameter of the particles. This parameter is available when **Repulsion force = Yes**.
Lift force[*)]	**Yes** | **No**	Account of the lift force due to rotation of particles. This parameter is available when **Cd ≠ (none)**.

Parameter	Possible values	Description
Cl factor	Numerical value. Default value: **0**.	Factor in the coefficient for the lift force due to rotation of particles. This is constant $C_{Lfactor}$ in Eq. (DispParticles.15) in section Equations for particles > Process 'Motion'. This parameter is available when **Lift force= Yes**.
Dl	Numerical value. Default value: **0**.	Distance D_L, [m], from wall at which the lift force due to particles rotation starts to act. This distance equals to approximately one diameter of the particles. This parameter is available when **Lift force= Yes**.
Cd	• **(none)** • **Model 1** • **Model 2** • **Model 3** • **Model 4** • **Model 5**	Drag coefficient of particles. Models No. 1-4 are for solid particles and droplets, model No. 5 is for bubbles. The selection is only possible when: • **Motion** in the continuous **Phase** is enabled • **Motion** in the dispersed **Phase** is enabled
Nu	• **(none)** • **Model 1** • **Model 2**	Selection of model for the **Nusselt** number. The selection is only possible when: • **Motion** in the continuous **Phase** is enabled • **Heat transfer** in the continuous **Phase** is enabled • **Motion** in the dispersed **Phase** is enabled • **Heat transfer** in the dispersed **Phase** is enabled
Heat exchange coef. (D-C)	A numerical value that can be specified by a constant, function or table.	Heat exchange coefficient between a dispersed **Phase** of the **Particles** type and a continuous **Phase**
Evaporation model	• **(none)** • **Model 1** • **Model 2** • **Model 3** • **Model 4** • **Model 5**	Selection of model for droplet evaporation. The selection is only possible when • **Motion** in the continuous **Phase** is enabled • **Motion** in the dispersed **Phase** is enabled • **Mass transfer** in the continuous **Phase** is enabled • **Mass transfer** in the dispersed **Phase** is enabled
Sh	• **Model 1** • **Model 2**	Selection of model for the **Sherwood** number
Is carrier phase	**Yes** | **No**	Ability of dispersed particles to move through the continuous **Phase**.

Parameter	Possible values	Description
		FlowVision doesn't examine which dispersed **Phase** infiltrate into any continuous **Phase**. If you set in a project that water drops infiltrate into continuous water phase, the program will simulate motion of water drops within water. So the user is completely responsible for correct and physically meaningful selection of **Substances** for **Phases**.
		This parameter is used in simulating of icing. **Is carrier phase = No** specified in properties of interaction of ice and waterdrops means that waterdrops cannot move through ice.

*) The **Lift force** has sense for all types of particles. The **Repulsion force** has sense only for bubbles; this is the force that pushes bubbles off a wall according to the lubrication theory.

It is possible to simulate mass flow from a dispersed **Phase** to a continuous **Phase** without use of the crystallization model of the dispersed **Phase**.

For this you have to specify **Is carrier phase = Yes** in properties of the **Phase interaction > Continuum-particles** element for the continuous **Phase**, in which the dispersed **Phase** is presented as a separate phase, and specify **Is carrier phase = No** in properties of the **Phase interaction > Continuum-particles** element for the continuous **Phase**, for which the dispersed **Phase** is source of the mass flow.

Initially the software implementation assumed simulation of transformation the dispersed **Phase** into a continuous **Phase** in the same aggregative state, but *FlowVision* doesn't trace, which dispersed **Phase** transforms into which continuous **Phase**, so if infiltration of particles of sand into continuous water is specified in a project, the program will actually simulate turning the substance of sand into the substance of water with the same mass. Thus the user is also completely responsible for correct and physically meaningful selection of **Substances** for **Phases**.

Parameters of the element **Models > Model #i > Phase interaction > Continuum-carcass**:

Parameter	Possible values	Description
Math. model	• **Continuum-carcass**	A model for interaction between the **Phases**, it is set automatically corresponding to types of **Phases**, which are presented in the **Model**. (This field is read-only)
Phase0	**Phase #j**	One of **Continuous Phases**, present in **Model #i** (This field is read-only)
Phase1	**Phase #k**	One of dispersed **Phases** of the **Carcass** type, present in **Model #i** (This field is read-only)
Blackness	Numerical value. The default value is **1**.	The blackness (emissivity) of surface of particles or carcass

Parameter	Possible values	Description
Heat exchange coef. (D-C)	A numerical value that can be specified by a constant, function or table.	Heat exchange coefficient between a dispersed **Phase** of the **Carcass** type and a continuous **Phase**

Group of parameters **Multiphase D** in the folder **Solver > Advanced settings**:

Parameter	Description
Advection scheme	Parameter of the **Dispersed solver**, which simulates phase transfer of dispersed **Phases**. Possible options are: **1st order scheme** | **2nd order scheme**.
Cloud boundary	Minimum value for the relative volume of **Dispersed Phase**, at which the volume mass and energy sources are calculated. The default value is 10^{-3}.
Film step is limited by task step	This parameter specifies how the time step for simulating the film's motion (the film's spreading) will be selected. Possible options are: • **Yes** – motion of the film will be simulated with the task's time step τ. This option allows the program to obtain a non-steady solution for the film's motion that is synchronized with the problem's time. • **No** – motion of the film will be simulated with its own explicit time step, $\tau_{expl,film}$, even when $\tau_{expl,film}>\tau$. Within a task's time step τ, computation of the film's motion will be done in several iterations, the number of which is set by the **Film CFL** parameter. This option is used to find the steady-state solution. The film's motion shown in this case can differ from the motion in the real non-steady process. See also sections Time step and Element «Time step».
Activation of disp. phase crystallization	The group of parameters **Activation of disp. phase crystallization** allows you to specify the moment of activating the simulation of crystallization the dispersed phase (or refuse starting this simulation). This group of parameters is only available when **Crystallization≠(none)** is set in properties of **Physical processes** of the dispersed phase of the **Particles** type. Some crystallization models include simulating of spreading the liquid film over the surface. When crystallization is not being simulated, the film's spreading is also not being simulated.
Activation of disp. phase crystallization > Type	The method how crystallization of the dispersed phase will be activated (if yes). Possible options are: • **Disabled** – crystallization of the dispersed phase will not be activated until you change the value of the **Type** parameter to either **Start in seconds** or **Start in steps**. • **Start in seconds** – crystallization of the dispersed phase will be activated at the specified time moment. • **Start in steps** – crystallization of the dispersed phase will be activated at the specified step
Activation of disp. phase crystallization > Start in steps	The step, at which crystallization of the dispersed phase will be activated. The default value is **0**. Parameters **Film CFL** (in properties of the **Time step** element) and **CFL for VOF source** (in the **Multiphase C** group of parameters, see subsection *"Multiphase C", parameters of the VOF solver, which simulates transfer of continuous phases*) can be set arbitrary at starting the project; their values will be ignored until simulation of the icing is activated. This parameter is available when **Activation of disp. phase crystallization > Type = Start in steps**.

Parameter	Description
Activation of disp. phase crystallization > Start in seconds	The time moment, [s], at which crystallization of the dispersed phase will be activated. The default value is **0**. Parameters **Film CFL** (in properties of the **Time step** element) and **CFL for VOF source** (in the **Multiphase C** group of parameters, see subsection *"Multiphase C", parameters of the VOF solver, which simulates transfer of continuous phases*) can be set arbitrary at starting the project; their values will be ignored until simulation of the icing is activated. This parameter is available when **Activation of disp. phase crystallization > Type = Start in seconds**.

Parameters of the folder **Phase #N**:

Parameter	Description
Number of size groups	Number of size groups in a size spectrum. See details in sections: • Spectra of particle sizes • Folder «Phases» (subsection *Folder «Phases > Phase #N > Size spectra» and elements «Size spectrum #N»*)

Parameters of elements **Size spectrum #N**:

Parameter	Description
Name	Name of the size spectrum
Size groups > [N] > Diam. particles	Diameter d^i and volume fraction φ^i of particles in the dispersed **Phase** for the size group **N**. The default value of the **Diam. particles** parameter is **1e-09**, which means 10^{-9} [m]. See details in sections: • Spectra of particle sizes • Folder «Phases» (subsection *Folder «Phases > Phase #N > Size spectra» and elements «Size spectrum #N»*)
Size groups > [N] > Volume fraction in the Phase	

Specific parameters of physical processes see below in sections:

- **Process 'Phase transfer'**
- **Process 'Motion'**
- **Process 'Heat transfer'**
- **Process 'Mass transfer'**
- **Process 'Crystallization'**

Parameters of physical processes in a dispersed **Phase** are described in next sections.

10.5.8.2.1 Process 'Phase transfer'

Parameters of the element **Preprocessor > Phases > Phase #i > Physical processes > Phase transfer**:

Parameter	Possible values	Description
Math. model	• **Convection & diffusion**	A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.)
Time step coefficient	An arbitrary numerical value determined by the problem (the default value is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the whole project (τ). *Thus:*

Parameter	Possible values	Description
	time steps for different processes sometimes accelerates convergence to the steady-state solution. Meaning of different values: **1** = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the problem as a whole). **> 1** = The calculations of the given process are accelerated. **(0, 1)** = The calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations / initialization).	Own time step of a physical process = τ × **Time step coefficient** Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time.
Droplets breakup **Min. radius model**	Models of breakup and models of coalescence of *liquid* droplets. Parameters of the selected models are set in properties of the **Droplets breakup: Model of droplets breakup** child element.	See details in section Equations for particles > Process 'Phase transfer', subsection *"Simulation of breakup and coalescence of droplets"*.

It is possible to simulate mass flow from a dispersed **Phase** to a continuous **Phase** without use of the crystallization model of the dispersed **Phase**.

For this you have to specify **Is carrier phase = Yes** in properties of the **Phase interaction > Continuum-particles** element for the continuous **Phase**, in which the dispersed **Phase** is presented as a separate phase, and specify **Is carrier phase = No** in properties of the **Phase interaction > Continuum-particles** element for the continuous **Phase**, for which the dispersed **Phase** is source of the mass flow.

Initially the software implementation assumed simulation of transformation the dispersed **Phase** into a continuous **Phase** in the same aggregative state, but *FlowVision* doesn't trace, which dispersed **Phase** transforms into which continuous **Phase**, so if infiltration of particles of sand into continuous water is specified in a project, the program will actually simulate turning the substance of sand into the substance of water with the same mass. Thus the user is also completely responsible for correct and physically meaningful selection of **Substances** for **Phases**.

10.5.8.2.2 Process 'Motion'

Parameters of the element **Preprocessor > Phases > Phase #i > Physical processes > Motion**:

Parameter	Possible values	Description
Math. model	• **Convection & conduction**	A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.)
Time step coefficient	An arbitrary numerical value determined by the problem (the default value is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different time steps for different processes sometimes accelerates convergence to the steady-	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the whole project (τ). *Thus:* Own time step of a physical process = τ × **Time step coefficient**

Parameter	Possible values	Description
	state solution. Meaning of different values: **1** = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the problem as a whole). **> 1** = The calculations of the given process are accelerated. **(0, 1)** = The calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations / initialization).	Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time.
D > X **D > Y** **D > Z**	Determined by the problem. (Default value: **0**)	These D and F parameters in the source term **D*TEMP_D+F** specify a user-defined force that acts on particles.
F > X **F > Y** **F > Z**	Determined by the problem. (Default value: **0**)	
A_rep	Determined by the problem. (Default value: **0**)	Coefficients in the implemented model of particles repulsion - see Eq. (DispParticles.4).
B_rep	Determined by the problem. (Default value: **600**)	
PhVol_max	Determined by the problem. (Default value: **0.63**)	Maximum permissible fraction of the cell volume occupied by the **Dispersed Phase** - see Eq. (DispParticles.4).

10.5.8.2.3 Process 'Heat transfer'

Parameters of the element **Preprocessor > Phases > Phase #i > Physical processes > Heat transfer**:

Parameter	Possible values	Description
Math. model	• **Convection & conduction**	A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.)
Time step coefficient	An arbitrary numerical value determined by the problem (the default value is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different time steps for different processes sometimes accelerates convergence to the steady-state solution. Meaning of different values: **1** = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the whole project (τ). *Thus:* Own time step of a physical process = τ × **Time step coefficient** Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time.

Parameter	Possible values	Description
	problem as a whole). **> 1** = The calculations of the given process are accelerated. **(0, 1)** = The calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations / initialization).	
D	Determined by the problem. (Default value: **0**)	User defined coefficient D_d in the formula: • (DispParticles.16) for phases **Particles** • (DispPorous.1) for phases **Carcass**
F	Determined by the problem. (Default value: 0)	User defined free term F_d in the formula: • (DispParticles.16) for phases **Particles** • (DispPorous.1) for phases **Carcass**

10.5.8.2.4 Process 'Mass transfer'

Parameters of the element **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer**:

Parameter	Possible values	Description
Math. model	• **Mass transfer** • **Coal**	A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.)
Time step coefficient	An arbitrary numerical value determined by the problem (the default value is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different time steps for different processes sometimes accelerates convergence to the steady-state solution. Meaning of different values: **1** = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the problem as a whole). **> 1** = The calculations of the given process are accelerated. **(0, 1)** = The calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations / initialization).	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the whole project (τ). *Thus:* Own time step of a physical process = τ × **Time step coefficient** Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time.
D	Determined by the problem. (Default value: **0**)	Expression for quantity D in the source term **D * MASS_D + F** (do not multiply by particles concentration)

Parameter	Possible values	Description
F	Determined by the problem. (Default value: **0**)	Expression for quantity F in the in source term **D * MASS_D + F** (do not multiply by particles concentration)
Model for particles	**Variable diameter** | **Constant diameter**	Model of transforming particles. The **Variable diameter** model assumes that the density of a particle remains constant while the diameter changes according to the mass loss of the **Particles Phase**. The **Constant diameter** model assumes that the diameter of a particle remains constant while the density changes according to the mass loss of the **Particles Phase**. See subsection *Models for transformation of particles*.

10.5.8.2.5 Process 'Crystallization'

Parameters of the element **Preprocessor > Phases > Phase #i > Physical processes > Crystallization**:

Parameter	Possible values	Description
Math. model	A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.) Possible values: • **Dry model** • **Film model**	
Time step coefficient	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the whole project (τ). *Thus:* Own time step of a physical process = τ × **Time step coefficient** Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time. An arbitrary numerical value determined by the problem (the default value is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different time steps for different processes sometimes accelerates convergence to the steady-state solution. Meaning of different values: **1** = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the problem as a whole). **> 1** = The calculations of the given process are accelerated. **(0, 1)** = The calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations / initialization).	
Ice roughness model	This is an empiric model of roughness applied to calculate the equivalent grit roughness on a surface of the geometry or on a surface of the solid phase. The surface becomes rough because of sequential growth and crystallization of subcooled drops of the dispersed phase on the surface. Empiric models of roughness are recommended for use only when icing of an aircraft is simulated.	

Parameter	Possible values	Description
	Possible values: • **(none)** • **Shin-Bond** • **Shin-Bond (local)** See subsection *Roughness of the ice surface*.	
LWC	Liquid water content, which is amount of condensed water containing in a unit volume of a cloud, in an undisturbed two-phase flow. The **LWC** parameter specifies the initial liquid water content in cells. This parameter is specified as a constant only. This parameter is available when **Ice roughness model = Shin-Bond**. ⚠ In the program's user interface the **LWC** parameter is set in [g/m^3]. A numerical value specified as a constant only	
Source smoothing	Number of iterations of surface smoothing of the volume source of the dispersed phase's substance that settle on the surface from the two-phase flow. This is an integer number specified as a constant only. An integer number specified as a constant only	
Film substance	This **Substance**, which will be used in calculations of thermal balances and the film's motion. The **Substance** of the dispersed phase might be not liquid only; it can be solid (as snowflakes or ice particles). The value is selected from the list.	
Film shedding model	The model of shedding the film from the surface (loss of the dispersed phase's substance from the film). Forming a new dispersed phase due to the film shedding is not taken into account yet. This parameter is available when **Math. model = Film model**. Possible values: • **(none)** • **Parametrical model** – the film's shedding from the surface occurs when any of the following conditions takes place: ○ ratio of the film's thickness h to the radius R of curvature of the surface in the near-surface cell exceeds the value of the **h/R min. shedding value** parameter (see below) ○ *or* projections of vectors of the film's velocity in adjacent near-surface cells to the normal to the face between these cells are directed towards to each other.	
h/R min. shedding value	The minimal value h/R for the film shedding model. This parameter specifies the threshold ratio of the film's thickness h to the radius R of curvature of the surface. When the h/R ratio exceeds the specified is value, film shedding occurs (film shedding can also occur due to another reason, see subsection *Film shedding model "Parametrical model"*). This is a numerical value that can be specified as a constant only. The default value is **1**.	
Minimum film height	This is an empirical estimate of the minimal height of the film (thickness), [m], below which the film is stable in this specific problem setting. In simulations of aircraft icing, when you have no empirical estimate for this minimal height of the film, it is recommended to keep the default value of this parameter. This is a numerical value that can be specified as a constant only. The default value is **1e-8**.	
Wetting model parameters > Model	The wetting model's coefficient β, see formula Cryst.12. The default value is **0**.	

Parameter	Possible values	Description
coefficient		
Wetting model parameters > Model of wetting angle deviation	Distribution law of the random value θ, which is local wetting angle on the solid surface, see formula Cryst.12. The **Normal distribution** value only is possible now.	
Wetting model parameters > Contact angle with solid phase	The mathematical expectation of the random value θ, which is local wetting (contact) angle on the solid surface, [degree], see formula Cryst.12. The default value is **75**.	
Wetting model parameters > Wetting angle deviation	The mean-square deviation of the random value θ, which is local wetting (contact) angle on the solid surface, [degree], see formula Cryst.12. The default value is **10**.	

Parameters of the element **Solver > Advanced settings > Multiphase C**:

Parameter	Description
CFL for VOF source	The CFL (Courant-Friedrichs-Lewy) number for the phase transfer's program block. It is calculated based on the source of the solid phase formed due to the crystallization (from the substance of the dispersed phase). This CFL value determines the step of motion of the inter-phase surface due to action of the source Q_{VOF} in the equation (PhTr.1) for the **VOF** variable. When this parameter is zero or negative (**CFL for VOF source** ≤ **0**), the time step for the phase transfer will be calculated based on the value of the **Relaxation** parameter. Positive value of this parameter (**CFL for VOF source** > **0**) determines the fraction, at which volume of the solid phase can increase/decrease in any cell per one (common for the whole simulation)time step. Thus, at one iteration, volume of the solid phase in any cell cannot increase at one common step more then (**CFL for VOF source**)×(volume of the cell). When icing of an aircraft is simulated, it is recommended to set the **CFL for VOF source** in the range from **0.1** to **0.4**.
Use VOF source for time step	This parameter specifies if the inter-phase surface's motion depends on the source Q_{VOF} in the equation (PhTr.1) for the variable **VOF**. When icing of aircraft is simulated, you should keep the default value of this parameter, **Use VOF source for time step = No**.

Parameters of the element **Solver > Time step**:

Parameter	Description
Film CFL	This is CFL (Courant-Friedrichs-Lewy) number for the dispersed phase crystallization's program block. This parameter determines the number of cycles of computation that are required to obtain a quasi-stationary solution of forming the film. When this parameter is zero, the phase crystallization's program block is inactive but the program makes calculations of the volume source of the dispersed phase on the solid surface. This is useful at the preliminary computation of the project to make adaptation of the grid in the area where drops fell out. Values above **1** define the number of computational circles of the quasi-stationary process of "substance fell-out - spreading the film - crystallization of the substance" with explicit step of convective mass transfer in the film. Such values are recommended for use only for simulating icing of aircraft or other closed contours. When icing of an aircraft is simulated, **Film CFL** is recommended to be set in the range from **3** to **5**.

It is possible to simulate mass flow from a dispersed **Phase** to a continuous **Phase** without use of the crystallization model of the dispersed **Phase**.

For this you have to specify **Is carrier phase = Yes** in properties of the **Phase interaction > Continuum-particles** element for the continuous **Phase**, in which the dispersed **Phase** is presented as a separate phase, and specify **Is carrier phase = No** in properties of the **Phase interaction > Continuum-particles** element for the continuous **Phase**, for which the dispersed **Phase** is source of the mass flow.

Initially the software implementation assumed simulation of transformation the dispersed **Phase** into a continuous **Phase** in the same aggregative state, but *FlowVision* doesn't trace, which dispersed **Phase** transforms into which continuous **Phase**, so if infiltration of particles of sand into continuous water is specified in a project, the program will actually simulate turning the substance of sand into the substance of water with the same mass. Thus the user is also completely responsible for correct and physically meaningful selection of **Substances** for **Phases**.

10.5.8.3 Equations for particles

The equations describing the processes proceeding in a dispersed **Phase** of the **Particles** type are considered in the given section.

This section has individual numeration of equations, references, and illustrations.

10.5.8.3.1 Process 'Phase transfer'

Simulation of this process assumes integration of inhomogeneous convection-diffusion equation for concentration of particles:

$$\frac{\partial n_d}{\partial t} + \nabla \cdot (\boldsymbol{V}_d n_d) = \nabla \cdot \left(\frac{\nu_{t,d}}{Sc_{t,d}} \nabla n_d \right) + \dot{n}_d \qquad \text{(DispParticles.1)}$$

$$\nu_{t,d} = \nu_{t,c} \qquad \text{(DispParticles.2)}$$

Here

n_d - concentration of particles [m^{-3}]

$\boldsymbol{V}_d$ - velocity of **Dispersed Phase** (i. e. particles) [m s^{-1}]

$\dot{n}_d$ - formation / destruction rate of particles [m^{-3} s^{-1}]

$\nu_{t,d}$ - kinematic coefficient of turbulent viscosity of particles [m^2 s^{-1}]

$\nu_{t,c}$ - kinematic coefficient of turbulent viscosity of **Continuous Phase** [m^2 s^{-1}]

$Sc_{t,d}$ - turbulent Schmidt number (dimensionless parameter characterizing turbulent diffusion of particles)

Simulation of breakup/coalescence of bubbles is not currently supported.

Simulation of breakup and coalescence of droplets

Based on many experimental data show that it is necessary to take into account processes of breakup and coalescence of *liquid particles* to adequately simulate dynamics of the dispersed phase when solving such problems as spraying in the atmosphere, injection of various kinds of fuel in combustion chambers, etc.

FlowVision implements three models of *breakup of liquid particles* and two models of *coalescence of liquid particles* based on the Euler's approach to simulating processes in the dispersed phase.

Breakup of droplets is simulated using the following models:

- **KHRT**
- **RD**
- **WAVE**

Coalescence of droplets is simulated using the following models:

- **MW**
- **Hiroyasu**

Main notations

Notation	Formula	Description
$\mathbf{U}_g$		Vector of velocity of the continuous (gas) phase, [m/s]
$\mathbf{U}_l$		Vector of velocity of the dispersed (liquid) phase, [m/s]
U_{rel}	$U_{rel} = \lvert\mathbf{U}_g - \mathbf{U}_l\rvert$	Absolute value of the relative velocity of the continuous phase (relating to the dispersed phase), [m/s]
ν_g		Kinematic coefficient of viscosity of the continuous phase, [m^2/s]
ν_l		Kinematic coefficient of viscosity of the dispersed phase, [m^2/s]
σ		Surface tension coefficient of the substance of the dispersed phase, [N/m]
r_0		Initial (on the current time step) radius of particles of the dispersed phase, [m]
We_g	$\frac{\rho_g U_{rel}^2 r_0}{\sigma}$	The Weber number of the continuous phase
We_l	$\frac{\rho_l U_{rel}^2 r_0}{\sigma}$	The Weber number of the dispersed phase
Re_g	$\frac{U_{rel} r_0}{\nu_g}$	The Reynolds number of the continuous phase
Oh	$\frac{\sqrt{We_l}}{Re_l} = \nu_l \sqrt{\frac{\rho_l}{r_0 \sigma}}$	The Ohnesorge number
T	$Oh\sqrt{We_g}$	The Taylor number

Models of breakup of droplets

Simulation of breakup of liquid particles includes two stages:

- breakup before the outlet of the nozzle and on the outlet of the nozzle (this is the primary breakup)
- breakup in the rest of the computational domain (this is the secondary breakup)

FlowVision implements the following models of the secondary breakup of droplets:

- **RD** [1]
- **WAVE** [2]
- **KHRT** [6]

These models adequately describe both secondary and primary (on the assumption of that diameter of droplets on exit from the nozzle is equal to the effective diameter of the nozzle) breakups of liquid droplets. The **KHRT** model assumes implementation of a special boundary condition for calculation the length of the initial jet, which is necessary for simulation of the secondary breakup. All these mentioned models supposedly allow correct implementation in the Euler's approach.

The fundamental equation for all aforementioned breakup models is the following equation:

$$\frac{dr}{dt} = -\frac{r_0 - r_{st}}{\tau}$$

where r_0 is the initial radius of liquid particles, r_{st} is the minimal stable radius of liquid particles (it is defined by the selected breakup model; the breakup occurs when $r_0 > r_{st}$), τ is the characteristic time of breakup of liquid particles (it is defined by the selected breakup model). Integration of this equation makes it possible to calculate the resulting average diameter of particles after the breakup:

$$r_{res} = r_0 - \frac{\Delta t}{\tau}(r_0 - r_{st})$$

where Δt is the time step of the computation. The resulting (after the breakup) concentration of the particles is calculated according to the following formula:

$$n_{res} = n_0 \left(\frac{r_0}{r_{res}} \right)^3$$

where n_0 is the initial concentration of the particles.

Specifics of the RD model

The **RD** model adequately describes dynamics of sprays in flows of continuous phase with Weber number in the range of 6 < We_g < 500 (these are most of natural and technical processes that include dynamics of sprays with below-catastrophic breakup mode.

It is assumed that the breakup occurs according either one or other physical mechanisms, the *bag* one or the *stripping* one.

The *bag* mechanism is applied when the relative velocity of the continuous phase relating to the dispersed phase is low, when the Weber number of the continuous phase $We_g < 0.5\sqrt{Re_g}$.

When the relative velocity increases ($We_g > 0.5\sqrt{Re_g}$) the *stripping* mechanism is applied.

Formulae to calculate the stable radius and time of breakup are provided below.

For the *bag* mode:

$$r_{st\_b} = \frac{6\sigma}{\rho_g U_{rel}^2}, \quad \tau_b = \pi \sqrt{\frac{\rho_l r_0^3}{2\sigma}}$$

For the *stripping* mode:

$$r_{st\_s} = \frac{\sigma^2}{4\rho_g^2 U_{rel}^3 \nu_g}, \quad \tau_s = B_2 \frac{r_0}{U_{rel}} \sqrt{\frac{\rho_l}{\rho_g}}$$

where B_2 is the model's constant, which is set in properties of the **Droplets breakup: RD model** element in the project tree by the **B_2 coef.** parameter (the default value is **10**).

Specifics of the WAVE model

The **RD** model soundly describes dynamics of sprays in flows of the continuous phase with large values of the Weber number in the range of 40 < We_g < 500 (this cases are high-speed in combustion chambers and other similar technique processes with below-catastrophic breakup mode).

According to [2], it is assumed that breakup of a particle occurs because of appearing a disturbance on the surface of the particle due to Kelvin-Helmholtz instability, and it is defined by wave parameters:

$$\frac{\Lambda_{KH}}{r_0} = 9.02\frac{\left(1+0.45Oh^{0.5}\right)\left(1+0.4T^{0.7}\right)}{\left(1+0.87\,We_g^{1.67}\right)^{0.6}}$$

$$\Omega_{KH} = \frac{0.34+0.38We_g^{1.5}}{\left(1+Oh\right)\left(1+1.4T^{0.6}\right)}\sqrt{\frac{\sigma}{\rho_l r_0^3}}$$

The stable radius of particles is defined as $r_{st}=B_0\Lambda_{KH}$, where B_0=0.61 is the model's constant.

The time of breakup is defined as:

$$\tau_{KH} = \frac{3.726B_1 r_0}{\Lambda\Omega}$$

where B_1 is the model's constant, which is set in properties of the **Droplets breakup: WAVE model** element in the project tree by the **B_1 coef.** parameter (the default value is **10**).

Specifics of the KHRT model

The **KHRT** model soundly describes dynamics of secondary breakup of sprays with high Weber numbers in the continuous phase, 40 < We_g < 500. In the current implementation this model doesn't take into account existence of the initial jet, which is non-fragmented part of the jet of known length (*Levich core length*) along direction of the injection, so this model is not recommended to simulate nozzles. The **KHRT** model should not be used with values of the Weber number below 40.

According to [6], it is assumed that breakup of a particle occurs because of appearing a disturbance on the surface of the particle due to either one or other competing mechanisms, *Kelvin-Helmholtz (KH)* or *Rayleigh-Taylor (RT)* ones, in the whole area where the dispersed phase is presented (without taking into account the space of the initial jet if the special boundary condition is not set). The scheme of breakup by the *Kelvin-Helmholtz* mechanism is similar to aforementioned one for the **WAVE** model. Breakup by the *Rayleigh-Taylor* mechanism is defined by the following wave parameters:

$$\Omega_{RT} = \sqrt{\frac{2}{3\sqrt{3\sigma}}}\frac{\left[-g_t(\rho_l-\rho_g)\right]^{3/2}}{\rho_l-\rho_g}$$

$$K_{RT} = \sqrt{\frac{-g_t(\rho_l-\rho_g)}{3\sigma}}$$

where g_t is acceleration in the direction of a particle's motion, g_t = **g**·**j**+**a**·**j**, where **a** is acceleration of the particle, **j** is the tangential vector to the trajectory.

The stable radius of particles is defined as:

$$r_{st\_RT} = \frac{\pi C_{RT}}{K_{RT}}$$

where C_{RT} is the model's constant, which is set in properties of the **Droplets breakup: KHRT model** element in the project tree by the **C_rt coef.** parameter (the default value is **0.1**).

The time of breakup is defined as:

$$\tau_{RT} = \frac{C_\tau}{\Omega_{RT}}$$

where C_τ is the model's constant, which is set in properties of the **Droplets breakup: KHRT model** element in the project tree by the **C_t coef.** parameter (the default value is **0.5**).

In the range where mechanisms *KH* and *RT* compete, those mechanism has priority, which requires less time upon condition that stable radius of particles, generated by the breakup, is less then the initial radius. It is supposed that if the *KH* mechanism requires less time and is not applied, the *RT* also will not be applied. So:

- when $\tau_{RT}<\tau_{KH}$: if $r_{st\_RT}<r_0$, then the *RT* breakup mechanism is applied, else ($r_{st\_RT}\geq r_0$) when $r_{st\_KH}<r_0$ the *KH* breakup mechanism is applied, and otherwise ($r_{st\_RT}\geq r_0$, $r_{st\_KH}\geq r_0$) the breakup doesn't occur;
- when $\tau_{RT}>\tau_{KH}$: if $r_{st\_KH}<r_0$, then the *KH* breakup mechanism is applied, else ($r_{st\_KH}\geq r_0$) the breakup doesn't occur.

Recommendation on how to calculate the initial radius of particles when a nozzle is simulated

$$2r_0 = d_{eff} = d_{inj}\sqrt{C_a}$$

where d_{eff} is the effective diameter of the nozzle, d_{inj} is the actual internal diameter of the nozzle's outlet, C_a is the contraction coefficient that depends on injection pressure and design features of the nozzle.

Taking small particles into account

You can enable taking generating small particles into account in any of the implemented breakup models (by specifying **Small drops account = Account (hybrid model)** in properties of the element **Droplets breakup: Model of droplets breakup**). See [2] for description of this approach.

In each of the implemented breakup models it is assumed that during one time step, along with reducing the initial radius of particles from r_0 to r_{res} (due to the breakup), small particles with radius r_{st} are generated (separated).

The small particles are taken into account when, within a time step, their required mass is accumulated defined as specified percentage to the total mass of the particles in a computational cell; more specifically, they are taken into account when the following condition is fulfilled:

$$\frac{(r_0^3 - r_{res}^3)}{r_0^3} > SD^{(\%)}$$

where the $SD^{(\%)}$ parameter is set in the user interface (this is the **Hyb. model %** parameter in properties of the element **Droplets breakup: Model of droplets breakup**).

When generating small particles is taken into account, concentration and radius of the particles, generated due to the breakup, are recalculated using the averaging, which is done using the following formulae:

$$n_{st} = n_0 \frac{(r_0^3 - r_{res}^3)}{r_{st}^3}$$

$$n_{res} = n_0 + n_{st}$$

$$r_{res} = \sqrt[3]{\frac{n_0 r_0^3 + n_{st} r_{st}^3}{n_{res}}}$$

Models of coalescence of droplets

Now the following models of coalescence of droplets (models of minimal radius) are implemented in *FlowVision*:

- **MW**
- **Hiroyasu**

Techniques of use of the coalescence models includes limitation the minimal allowed radius of the particles, which are generated during the coalescence, by the value r_{st}^{min}. Thus, if in the breakup model the stable radius $r_{st} > r_{st}^{min}$, then calculations will use r_{st}, and when $r_{st} \leq r_{st}^{min}$, the calculations will use r_{st}^{min}.

Specifics of the MW model

According to [8], it is assumed that minimal allowed radius of particles, which are generated by the breakup, is defined by the following equation:

$$r_{st}^{\min} = r_0 \times 60 \times Re_l^{-0.22} \times We_l^{-0.31} \times \left(\frac{\rho_l}{\rho_g} \right)^{-0.17}$$

Specifics of the Hiroyasu model

According to [9], it is assumed that minimal allowed radius of particles, which are generated by the breakup, is defined by the following equation:

$$r_{st}^{\min} = \max(r_{LS}, r_{HS})$$

where:

$$r_{LS} = r_0 \times 4.12 \times Re_l^{0.12} \times We_l^{-0.75} \times \left(\frac{\nu_l \rho_l}{\nu_g \rho_g} \right)^{0.54} \times \left(\frac{\rho_l}{\rho_g} \right)^{-0.18}$$

$$r_{HS} = r_0 \times 0.38 \times Re_l^{0.25} \times We_l^{-0.32} \times \left(\frac{\nu_l \rho_l}{\nu_g \rho_g} \right)^{0.37} \times \left(\frac{\rho_l}{\rho_g} \right)^{-0.47}$$

References for this subsection

1. Reitz R.D., Diwakar R. Structure of high-pressure fuel sprays // SAE Technical paper 870598, 1987.
2. Reitz R.D. Modeling Atomization Processes in High-Pressure Vaporizing Sprays // Atomization and Spray Technology, 1987, Vol.3, pp. 309-337.
3. Hsiang L.-P., Faeth G.M. Near-limit drop deformation and secondary breakup // Int. j. Multiphase flow, 1992, Vol. 18, No. 5, pp. 635-652.
4. O'Rourke P.J., Amsden A.A. The TAB method for numerical calculation of spray droplet breakup // SAE Technical Paper 872089, 1987.
5. Tanner F.X. Liquid jet atomization and droplet breakup modeling of non-evaporating diesel fuel sprays // SAE Technical Paper Series 970050, 1997.
6. Beale J.C., Reitz R.D. Modeling spray atomization with the Kelvin-Helmholtz/Rayleigh-Taylor hybrid model // Atomization and sprays, 1999, Vol. 9, pp. 623-650.
7. O'Rourke P.J., Bracco F.V. Modelling of drop interactions in thick sprays and comparison with experiment // Stratified Charge Auto Engines Conf. (IMechE, London, 1980), p. 101.
8. Morel T., Wahiduzzaman S. Modeling of diesel combustion and emissions // XXVI FISITA Congress, 1996.
9. Hiroyasu H., Arai. M., Tabata M. Empirical equations for the Sauter Mean Diameter of diesel spray // SAE paper 890464, 1989.

Initial conditions

User specifies a value for the initial relative fraction in a cell volume occupied by **Phase volume** of the dispersed **Phase Particles**, φ_d.

Boundary conditions

"Symmetry"

User specifies nothing. The condition can be selected in the following boundary templates:

- **Symmetry**
- **Wall**
- **Free outlet**
- **Inlet/Outlet**
- **Free surface**
- **Nonreflecting**

"Value"

User specifies the value for the **Phase volume** of the dispersed **Phase Particles**, φ_d, in the **Value** parameter. The boundary condition can be set on boundaries of the following types:

- **Symmetry**
- **Wall**
- **Free outlet**
- **Inlet/Outlet**
- **Nonreflecting**

"Zero gradient"

User specifies nothing. The condition can be selected in the following boundary templates:

- **Symmetry**
- **Wall**
- **Free outlet**
- **Inlet/Outlet**

"Permeable surface"

User specifies nothing. The condition can be selected in the following boundary templates:

- **Symmetry**
- **Wall**
- **Wall, film**
- **Free outlet**
- **Inlet/Outlet**

10.5.8.3.2 Process 'Motion'

The momentum transfer of particles is described by non-uniform convection-diffusion equation for conservative variable $\boldsymbol{V_d} M_d n_d$. In the course of integration of this equation, it is assumed that concentrations of particles n_d^n , n_d^{n+1} and their masses M_d^n , M_d^{n+1} are known.

$$\frac{\partial\left(V_{d,i} M_d n_d\right)}{\partial t} + \nabla\cdot\left(\boldsymbol{V_d} V_{d,i} M_d n_d\right) = \nabla\cdot\left(\frac{\nu_{t,d}}{Sc_{t,d}}\nabla\left(V_{d,i} M_d n_d\right)\right) - n_d\frac{\pi d^3}{6}\nabla_i p + n_d\frac{\pi d^3}{6}\left(\rho_d - \rho_c\right)g_i + F_{rep,i} + F_{D,i} + F_{L,i} + F_{W,i} \quad \text{(DispParticles.3)}$$

$$\boldsymbol{F}_{rep} = -\frac{\pi d^3}{6} A_{rep}\, exp\left(-B_{rep}\left(\varphi_{d,max} - \varphi_d\right)\right)\nabla n_d \quad \text{(DispParticles.4)}$$

$$\boldsymbol{F}_D = n_d \rho_c C_D \frac{\pi d^2}{8}\left|\boldsymbol{V}_c - \boldsymbol{V}_d\right|\left(\boldsymbol{V}_c - \boldsymbol{V}_d\right) \quad \text{(DispParticles.5)}$$

$$\boldsymbol{F}_L = n_d \rho_c C_L \frac{\pi d^3}{8}\left(\boldsymbol{V}_c - \boldsymbol{V}_d\right)\times\boldsymbol{\omega_c} \quad \text{(DispParticles.6)}$$

$$\boldsymbol{\omega_c} = \nabla\times\boldsymbol{V}_c$$

$$\boldsymbol{F}_W = n_d \rho_c C_W \frac{\pi d^3}{6} max\left\{0, \left(\frac{1}{y} - \frac{1}{D_w}\right)\right\}\left(\boldsymbol{V}_c - \boldsymbol{V}_d\right)^2 \boldsymbol{n} \quad \text{(DispParticles.7)}$$

Here

p - pressure [Pa]

ρ_d - density of **Phase Particles** [kg m^{-3}]

ρ_c - density of **Continuous Phase** [kg m^{-3}]

$\boldsymbol{F}_{rep}$ - force pushing particles apart [N m^{-3}]

A_{rep} - coefficient in the implemented model for the force pushing particles apart [Pa]

B_{rep} - coefficient in the implemented model for the force pushing particles apart

$\varphi_{d,max}$ - maximum permissible relative volume of **Phase Particles** in a cell in the implemented model for the force pushing particles apart

$\boldsymbol{F}_D$ - drag force of the whole of particles of the dispersed phase [N m^{-3}]

C_D - drag coefficient

$\boldsymbol{V}_c$ - velocity of **Continuous Phase** [m s^{-1}]

$\boldsymbol{F}_L$ - lift force for particles [N m^{-3}]

C_L - lift coefficient

$\boldsymbol{F}_W$ - lubrication force for bubbles (the force pushing bubbles off a wall according to the lubrication theory)

C_W - lubrication coefficient

D_W - distance to a wall (approximately equal to the particles' diameter) at which the action of the lubrication force cancels [m]

Five models for the particles drag coefficient are implemented in the *FlowVision* software:

$$C_D = \frac{21.12}{\mathrm{Re}_d} + 6.3\,\mathrm{Re}_d^{-0.5} + 0.25$$ - Model 1 [1] (DispParticles.8)

$$C_D = \frac{24}{\mathrm{Re}_d} + 0.44$$ - Model 2 [2] (DispParticles.9)

$$C_D = \frac{24}{Re_d} + 4\,Re_d^{-1/3} \qquad Re_d < 1000$$
$$C_D = 0.44 \qquad Re_d \geq 1000$$
- Model 3 [3] (DispParticles.10)

$$C_D = \frac{24}{\mathrm{Re}_d} + 5.48\,\mathrm{Re}_d^{-0.573} + 0.36$$ - Model 4 [4] (DispParticles.11)

$$C_D = \frac{24}{Re_d}\left(1 + 0.15\,Re_d^{0.687}\right)\frac{1}{\varphi_c^2}$$ - Model 5 (for bubbles) [5] (DispParticles.12)

The last factor in model (DispParticles.12) takes account of the effect of particles' cloud (swarm) [6], [7].

In expressions (DispParticles.8) - (DispParticles.12)

Re_d - Reynolds number for particles

It is determined by formula

$$Re_d = \frac{\rho_c \left|\boldsymbol{V}_c - \boldsymbol{V}_d\right| d}{\mu_c}$$ (DispParticles.13)

In the Stokes regime (Re_d < 0.1)

$$C_D = \frac{24}{\mathrm{Re}_d} = 24\,\frac{\mu_c}{\rho_c \cdot \left|\boldsymbol{V}_c - \boldsymbol{V}_d\right| \cdot d}$$ (DispParticles.14)

The model for the lift coefficient implemented in the *FlowVision* software is as follows [8, 9]:

$$C_L = min\left(0.5, C_{Lfactor}\frac{1}{d}\sqrt{\frac{\mu_c}{\rho_c\sqrt{S}}}\right)\frac{1}{\varphi_c^2} \qquad \text{(DispParticles.15)}$$

Here

$C_{Lfactor}$ - model constant

$\sqrt{S}$ - scalar quantity characterizing local gradient of fluid velocity [s^{-1}]

The last factor in model (DispParticles.15) takes account of the effect of particles' cloud (swarm) similarly to model (DispParticles.12) for the drag coefficient. The action of the lift force starts at distance D_L from a wall. This distance is approximately equal to the particles' radius.

Quantities $C_{Lfactor}$, D_L, C_W, D_W are specified in the *FlowVision* interface.

Non-sphericity of particles is not taken into account in expressions (DispParticles.8) - (DispParticles.12), (DispParticles.15).

Initial conditions

User specifies values of components x-, y-, z- of the initial particles velocity [m s^{-1}].

Boundary conditions

"Contact with wall"

User specifies the restitution coefficients for the normal and tangential components of the particles momentum (parameters **Coeff. norm.** and **Coeff. tang.**). The physical sense of these coefficients is explained below. It is assumed that particles can bounce from the wall and stick to it.

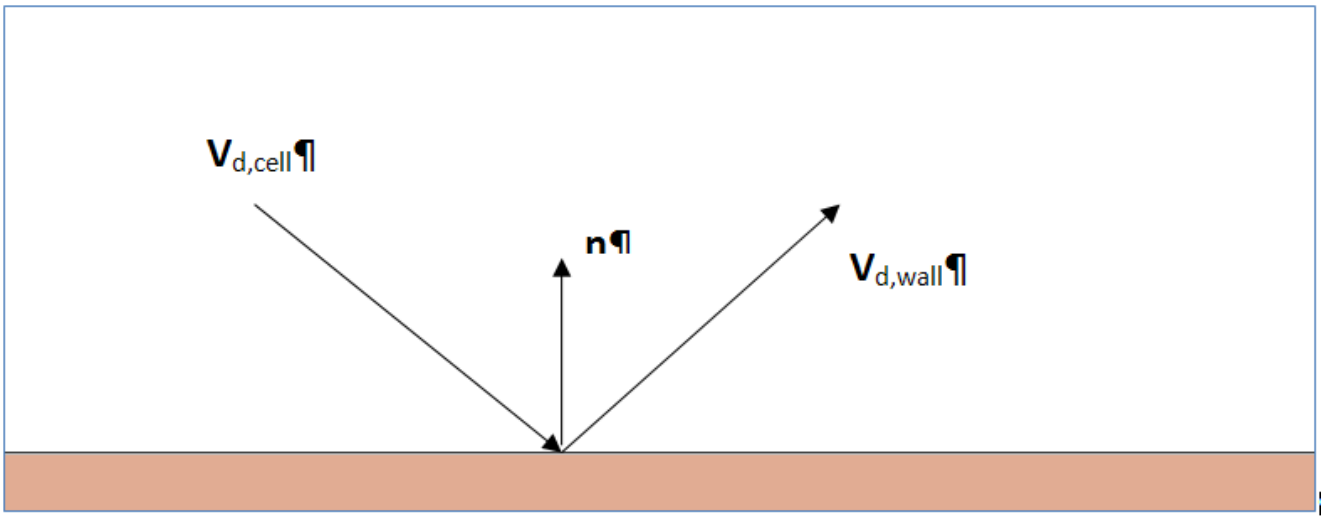


Figure 1. Perfectly elastic rebound of a particle from a wall.

Introduce the following notations:

$V_{d,y} = \boldsymbol{V}_{d,cell}\boldsymbol{n}$ - normal component of the particles velocity (in the local coordinate system, related to the wall)

$V_{d,x} = \left|\boldsymbol{V}_{d,cell} - \boldsymbol{n}V_{d,n}\right| = \boldsymbol{V}_{d,cell}\boldsymbol{\tau}$ - tangential component of the particles velocity (in the local coordinate system, related to the wall)

Here

$\boldsymbol{n}$ - internal normal to the wall (directed inside the adjacent cell)

The convective flux of the particles specific momentum from the wall reads:

$$\boldsymbol{F}_{d,conv} = V_{d,y}\boldsymbol{V}_{d,wall}$$

If the particles move from the wall ($\boldsymbol{V}_{d,cell}\boldsymbol{n} \geq 0$), we shall assume that

$$\boldsymbol{V}_{d,wall} = \boldsymbol{V}_{d,cell}$$

If the particles move towards the wall ($\boldsymbol{V}_{d,cell}\boldsymbol{n} < 0$), we shall assume that after collision with the wall their velocity becomes:

$$\boldsymbol{V}_{d,wall} = \boldsymbol{n}k_y\left(-V_{d,y}\right) + \boldsymbol{\tau}k_x V_{d,x} = \boldsymbol{n}k_y\left(-V_{d,y}\right) + k_x\left(\boldsymbol{V}_{d,cell} - \boldsymbol{n}V_{d,y}\right) = \boldsymbol{V}_{d,cell}k_x - \boldsymbol{n}V_{d,y}\left(k_y + k_x\right)$$

$$0 \leq k_y \leq 1 \quad 0 \leq k_x \leq 1$$

Here

k_x - coefficient of restitution of the tangential component of the particles momentum after collision with the given surface,

k_y - coefficient of restitution of the normal component of the particles momentum after collision with the given surface.

In the case of perfectly elastic rebound preserving the tangential component of the particles velocity

$$k_y = 1 \quad k_x = 1$$

Therefore

$$\boldsymbol{V}_{d,wall} = \boldsymbol{V}_{d,cell} - 2\boldsymbol{n}\left(\boldsymbol{V}_{d,cell}\boldsymbol{n}\right)$$

In the case of perfectly inelastic rebound preserving the tangential component of the particles velocity

$$k_y = 0 \quad k_x = 1$$

Therefore

$$\boldsymbol{V}_{d,wall} = \boldsymbol{V}_{d,cell} - \boldsymbol{n}\left(\boldsymbol{V}_{d,cell}\boldsymbol{n}\right)$$

In the case of sticking:

$$k_y = 0 \quad k_x = 0$$

$$\boldsymbol{V}_{d,wall} = 0$$

In general case, the convective flux of the specific momentum from the wall is computed from:

$$\boldsymbol{J}_{d,conv} = V_{d,y}\left[\boldsymbol{V}_{d,cell}k_x - \boldsymbol{n}V_{d,y}\left(k_y + k_x\right)\right]$$

The condition can be selected in the following boundary templates:

- **Symmetry**
- **Wall**
- **Free surface**

"Particles' velocity"

User specifies values for the x-, y-, z- components of the vector of particles velocity [m s^{-1}] (parameters **X**, **Y**, **Z**). The physical sense of this condition is explained below. If the particles flow obtained in a near-boundary cell at the previous time is directed inside the computational domain (particles inlet), the particles velocity at the given boundary is determined by the specified values of parameters **X**, **Y**, **Z**. If the particles flow obtained in a near-boundary cell at the previous time is directed inside the computational domain (particles outlet), the particles velocity at the given boundary is assumed equal to that in the center of the cell. In this case, the components of the particles velocity specified in the interface are ignored.

The condition can be selected in the following boundary templates:

- **Inlet/Outlet**
- **Symmetry**
- **Wall**
- **Free surface**
- **Free outlet**
- **Nonreflecting**

"Particles' volume velocity"

User specifies a value for the volume particles velocity $V_{d,n}\varphi_d$ (parameter **Value**) [m s^{-1}]. The condition can be selected in the following boundary templates:

- **Inlet/Outlet**
- **Symmetry**
- **Wall**
- **Free surface**

"Particles' mass velocity"

User specifies a value for the mass particles velocity $\rho_d V_{d,n}\varphi_d$ [kg m^{-2} s^{-1}] (parameter **Value**). The condition can be selected in the following boundary templates:

- **Inlet/Outlet**
- **Symmetry**
- **Wall**
- **Free surface**

"Permeable surface"

User specifies nothing. The condition can be selected in the following boundary templates:

- **Symmetry**
- **Wall**
- **Wall, film**
- **Free outlet**
- **Inlet/Outlet**

10.5.8.3.3 Process 'Heat transfer'

The energy transfer of particles is described by non-uniform convection-diffusion equation for conservative variable $h_d M_d n_d$. In the course of integration of this equation, it is assumed that concentrations of particles n_d^n, n_d^{n+1} and their masses M_d^n, M_d^{n+1} are known.

$$\frac{\partial(h_d M_d n_d)}{\partial t} + \nabla(\mathbf{V}_d h_d M_d n_d) = \nabla\cdot\left(\left(\frac{\nu_{t,d}}{Sc_{t,d}}\nabla(h_d M_d n_d)\right)\right) + \\ + Q_d^{enth} + D_d T_d + F_d \qquad \text{(DispParticles.16)}$$

$$Q_d^{enth} = n_d\left[\pi d Nu_d \lambda_c (T_c - T_d) + \pi d^2 \sigma_{rad}\varepsilon_d (T_c^4 - T_d^4) - \pi d^2 \sum_{i=1}^{Nd} \dot{m}_{d,i} h_{d,i}(T_d)\right] \qquad \text{(DispParticles.17)}$$

Here

h_d - thermodynamic enthalpy of **Phase Particles** [J kg^{-1}]

Nu_d - Nusselt number for **Phase Particles**

λ_c - coefficient of thermal conductivity of **Continuous Phase** [kg m s^{-3} K^{-1}]

T_d - temperature of **Phase Particles** (relative to the reference one) [K]

T_c - temperature of **Continuous Phase** (relative to the reference one) [K]

σ_{rad} - Stefan-Boltzmann constant [W m^{-2} K^{-4}]

ε_d - blackness (emissivity) of **Phase Particles**

$\dot{m}_{d,i}$ - specific rate of change of the mass of the i-th Substance in **Phase Particles** [kg m^{-2} s^{-1}]

$h_{d,i}$ - enthalpy of the i-th Substance in **Phase Particles** [J kg^{-1}]

N_d - number of Substances in **Phase Particles**

D_d - the user's coefficient at relative temperature of particles, which is set by parameter **D** in the window of element **Physical processes > Heat transfer** of **Phase Particles**.

F_d - the user's source term, which is set by parameter **F** in the window of element **Physical processes > Heat transfer** of **Phase Particles**.

The Nusselt number Nu_d is determined by:

$$\left|\boldsymbol{J}_{q,c\to d}\right| = Nu_d \frac{\lambda_c}{d}\left(T_c - T_d\right) \qquad \text{(DispParticles.18)}$$

where

$\boldsymbol{J}_{q,c\to d}$ - specific heat flux from **Continuous Phase** to **Phase Particles** [W m^{-2}]

Two models for the particles' Nusselt number are implemented in *FlowVision* (both are given in [10]):

$$Nu_d = 2 + 0.552\,\mathrm{Re}_d^{1/2}\,\mathrm{Pr}_c^{1/3} \quad \text{- Model 1} \qquad \text{(DispParticles.19)}$$

$$Nu_d = 1 + \left(1 + \mathrm{Re}_d\,\mathrm{Pr}_c\right)^{1/3}, \quad \mathrm{Re}_d \le 1$$
$$Nu_d = 1 + \left(1 + \mathrm{Re}_d\,\mathrm{Pr}_c\right)^{1/3}\mathrm{Re}_d^{0.077}, \quad \mathrm{Re}_d > 1 \quad \text{- Model 2} \qquad \text{(DispParticles.20)}$$

In expressions (DispParticles.19) and (DispParticles.20),

$\mathrm{Pr}_c = \dfrac{\mu_c C_{p,c}}{\lambda_c}$ - the Prandtl number for **Continuous Phase**

$C_{p,c}$ - specific heat of **Continuous Phase** [J kg^{-1} K^{-1}]

Initial conditions

User specifies a value for the initial particles temperature T_d [K].

Boundary conditions

"Symmetry"

User specifies nothing. The condition can be selected in the following boundary templates:

- **Symmetry**
- **Wall**

- **Free outlet**
- **Inlet/Outlet**
- **Free surface**
- **Nonreflecting**

"Value"

User specifies a value for the particles temperature T_d [K] (parameter **Value**). The condition can be selected in the following boundary templates:

- **Symmetry**
- **Wall**
- **Free outlet**
- **Inlet/Outlet**
- **Nonreflecting**

"Zero gradient"

User specifies nothing. The condition can be selected in the following boundary templates:

- **Symmetry**
- **Wall**
- **Free outlet**
- **Free surface**
- **Inlet/Outlet**

"Permeable surface"

User specifies nothing. The condition can be selected in the following boundary templates:

- **Symmetry**
- **Wall**
- **Wall, film**
- **Free outlet**
- **Inlet/Outlet**

10.5.8.3.4 Process 'Mass transfer'

The mass transfer of particles is described by non-uniform convection-diffusion equation for conservative variable $M_d n_d$. In the course of integration of this equation, it is assumed that concentrations of particles n_d^n , n_d^{n+1} .

$$\frac{\partial(M_d n_d)}{\partial t} + \nabla \cdot (\boldsymbol{V}_d M_d n_d) = \nabla \cdot \left(\left(\frac{\nu_{t,d}}{Sc_{t,d}} \nabla (M_d n_d) \right) \right) - Q_d^{mass} + M_d \dot{n}_d \qquad \text{(DispParticles.21)}$$

$$Q_d^{mass} = n_d \pi d^2 \dot{m}_d = n_d \pi d^2 \sum_{i=1}^{Nd} \dot{m}_{d,i} \qquad \text{(DispParticles.22)}$$

Here

M_d - local average (over a cell) mass of particles of a given size family [kg]

Q_{d}^{mass} - rate of change of the mass of **Phase Particles** [kg m^{-3} s^{-1}]

$\dot{m}_d$ - specific rate of change of the mass of **Phase Particles** [kg m^{-2} s^{-1}]

$\dot{m}_{d,i}$ - specific rate of change of the mass of the i-th Substance in **Phase Particles** [kg m^{-2} s^{-1}]

N_d - number of Substances in **Phase Particles**

d - local average (over a cell) diameter of particles [m]

Equation (DispParticles.21) allows for simulation of evaporation / condensation proceeding at the particles' surface. In this case

$$\dot{m}_d = \frac{Sh_d}{Sc_c}\frac{\mu_c}{d}\Phi_d$$ (DispParticles.23)

Here

Sh_d - Sherwood number for **Phase Particles**

$Sc_c = \frac{\mu_c}{\rho_c D_c}$ - molecular Schmidt number for **Continuous Phase**

μ_c - dynamic coefficient of viscosity for **Continuous Phase** [kg m^{-1} s^{-1}]

Φ_d - dimensionless quantity characterizing specific rate of evaporation proceeding at the particles surface

Two models for the particles Sherwood number are implemented in the *FlowVision* software (both are given in [10]):

$$Sh_d = 2 + 0.552\, Re_d^{1/2}\, Sc_c^{1/3}$$ - Model 1 (DispParticles.24)

$$Sh_d = 1 + (1 + Re_d\, Sc_c)^{1/3}, \quad Re_d \le 1$$
$$Sh_d = 1 + (1 + Re_d\, Sc_c)^{1/3}\, Re_d^{0.077}, \quad Re_d > 1$$
- Model 2 (DispParticles.25)

Five models for quantity Φ_d, characterizing the rate of particles evaporation, are implemented in the *FlowVision* software:

$$\Phi_d = \ln\left[1 + \frac{Y_{vapor,sat}(T_d) - Y_{vapor}}{1 - Y_{vapor,sat}(T_d)}\right]$$ - Model 1 (DispParticles.26)

$$\Phi_d = \frac{Y_{vapor,sat}(T_d) - Y_{vapor}}{1 - Y_{vapor,sat}(T_d)}$$ - Model 2 (DispParticles.27)

$$\Phi_d = \frac{Y_{vapor,sat}(T_d) - Y_{vapor}}{1 - Y_{vapor,sat}(T_d)} \frac{1}{1 + C_{p,vapor}\frac{T_c - T_d}{h_{lat}(T_d)}}$$ - Model 3 (DispParticles.28)

$$\Phi_d = Y_{vapor,sat}(T_d) - Y_{vapor}$$ - Model 4 (DispParticles.29)

$$\Phi_d = \left(Y_{vapor,sat}(T_d) - Y_{vapor}\right) \frac{1}{1 + C_{p,vapor}\frac{T_c - T_d}{h_{lat}(T_d)}}$$ - Model 5 (DispParticles.30)

Here

Y_{vapor} - average mass fraction of vapor in the cell volume filled by **Continuous Phase** (gas)

$Y_{vapor,sat}(T_d)$ - mass fraction of vapor at the particles surface, defined by the saturated vapor pressure at temperature T_d. The mass fraction of vapor at the particles' surface is calculated from the saturated vapor pressure using equations (MT-Ablation.19) and (MT-Ablation.20) in subsection *Model 'Sublimation'* of section **Mass transfer > Ablation > Equations**.

T_d - average temperature of **Phase Particles** in a given cell (relative to the reference one) [K]

T_c - average temperature of **Continuous Phase** in a given cell (relative to the reference one) [K]

The difference in the results obtained with these models are discussed in [11].

In the *FlowVision* software, it is assumed that at $Y_{vapor,sat}(T_d)>0.99$ and $T_c>T_{d,boil}$ droplets evaporate in the boiling regime:

$$\Phi = Nu\frac{Sc}{Sh}\frac{\lambda_c}{\mu_c}\frac{T_c - T_{d,boil}}{h_{lat}(T_{d,boil})} \quad \text{(DispParticles.31)}$$

Here

$T_{d,boil}$ - boiling temperature of **Phase Particles** [K]

λ_c - thermal conductivity of **Continuous Phase** (gas) [kg m s^{-3} K^{-1}]

h_{lat} - specific latent heat of evaporation of **Phase Particles** [J kg^{-1}]

Initial conditions

User specifies a value for the initial particles diameter d [m].

Boundary conditions

"Symmetry"

User specifies nothing. The condition can be selected in the following boundary templates:

- **Symmetry**
- **Wall**
- **Free outlet**
- **Inlet/Outlet**
- **Free surface**
- **Nonreflecting**

"Value"

User specifies a value for the particles diameter d [m] (parameter **Value**). The condition can be selected in the following boundary templates:

- **Symmetry**
- **Wall**
- **Free outlet**
- **Inlet/Outlet**
- **Nonreflecting**

"Zero gradient"

User specifies nothing. The condition can be selected in the following boundary templates:

- **Symmetry**
- **Wall**
- **Free outlet**
- **Inlet/Outlet**

"Permeable surface"

User specifies nothing. The condition can be selected in the following boundary templates:

- **Symmetry**
- **Wall**
- **Wall, film**
- **Free outlet**
- **Inlet/Outlet**

10.5.8.4 Equations for porous carcass

The equations describing the processes proceeding in **Phase Carcass** are considered in the given section.

This phase is a solid porous body (a body having voids). Examples:

- arbitrary filter,
- heat shield of re-entry vehicle,
- porous burner,
- oil-bearing rock.

It is assumed that the simulated porous body is immovable.

This section has individual numeration of equations.

10.5.8.4.1 Process "Heat transfer"

The energy equation is solved with respect to the thermodynamic enthalpy:

$$\frac{\partial(\varphi_d \rho_d h_d(T_d))}{\partial t} = \nabla(\varphi_d \hat{\lambda}_d \nabla T_d) + \varphi_d k_{HE}(T_c - T_d) + \varphi_d \sigma_{rad} \varepsilon(T_c^4 - T_d^4) + \sum_{i=1}^{N_s} \dot{m}_{d,i} h_{d,i}(T_d) + D_d T_d + F_d \quad \text{(DispPorous.1)}$$

Here

φ_d - relative volume of computational cell, occupied by **Phase Carcass**

ρ_d - density of **Phase Carcass** [kg m^{-3}]

h_d - thermodynamic enthalpy of **Phase Carcass** [J kg^{-1}]

T_d - temperature of **Phase Carcass** (relative to the reference temperature) [K]

T_c - temperature of **Continuous Phase** (relative to the reference temperature) [K]

$\hat{\lambda}_d$ - heat conductivity tensor for **Phase Carcass** [kg m s^{-3} K^{-1}]

k_{HE} - coefficient of heat exchange between **Continuous Phase** and **Phase Carcass**, which is set by parameter **Heat exchange coef. (D-C)** in the **Properties** window of element **Models > Phase interaction > Continuum-dispersed**

σ_{rad} - Stefan-Boltzmann constant [W m^{-2} K^{-4}]

ε - blackness (emissivity) of **Phase Carcass**

$\dot{m}_{d,i}$ - mass rate of destruction of the i-th Substance in **Phase Carcass** [kg m^{-3} s^{-1}]

$h_{d,i}$ - enthalpy of the i-th Substance in **Phase Carcass** [J kg^{-1}]

D_d - user's coefficient at the relative temperature of **Phase Carcass**, which is set by parameter **D** in the **Properties** window of element **Physical processes > Heat transfer** of **Phase Carcass**.

F_d - user's source, which is set by parameter **F** in the **Properties** window of element **Physical processes > Heat transfer** of **Phase Carcass**.

10.5.8.4.2 Process "Mass transfer"

The rate of changing the volume fraction of the i-th **Substance** in **Phase Carcass** is described by an equation of Arrhenius type:

$$\frac{\partial \varphi_{d,i}}{\partial t} = -A \varphi_{d,i}^n \exp\left(-\frac{E}{R_A T_d}\right) \quad \text{(DispPorous.2)}$$

The mass rate of destruction of the i-th **Substance** in **Phase Carcass** is determined by expression

$$\dot{m}_{d,i} = \rho_{d,i} \frac{\partial \varphi_{d,i}}{\partial t} \quad \text{(DispPorous.3)}$$

In Eqs. (DispPorous.2) and (DispPorous.3)

$\varphi_{d,i}$ - relative volume of computational cell, occupied by the i-th **Substance** in **Phase Carcass**

ρ_d - density of **Phase Carcass** [kg m^{-3}]

A, n, E - parameters, which determine the rate of destruction of the i-th **Substance** in **Phase Carcass**

T_d - temperature of **Phase Carcass** (relative to the reference temperature) [K]

R_A - universal gas constant [J mole^{-1}·K^{-1}]

10.5.8.5 Equations for continuous medium

The equations describing the processes proceeding in a **Continuous Phase** are considered in the given section. The equations take into account presence of a dispersed phase (**Phase Particles** or **Phase Carcass**) in the computational domain. Therefore they differ from the equations discussed in sections:

- **Motion**
- **Heat transfer**
- **Mass transfer**
- **Turbulence**

This section has individual numeration of equations.

10.5.8.5.1 Process 'Motion'

The continuity equation:

$$\frac{\partial(\varphi_c\rho_c)}{\partial t}+\nabla\cdot(\varphi_c\rho_c\boldsymbol{V}_c)=Q_d^{mass}-M_d\dot{n}_d \qquad \text{(DispContinuous.1)}$$

Here

φ_c - relative volume of **Continuous Phase**

ρ_c - density of **Continuous Phase** [kg m^{-3}]

$\boldsymbol{V}_c$ - velocity of **Continuous Phase** [m s^{-1}]

Source term Q_d^{mass} is determined by Eq. (22) (**Equations for particles > Process 'Mass transfer'**).

Navier-Stokes model

The momentum equation:

$$\frac{\partial(\varphi_c\rho_c\boldsymbol{V}_c)}{\partial t}+\nabla\cdot(\varphi_c\rho_c\boldsymbol{V}_c\otimes\boldsymbol{V}_c)=-\varphi_c\nabla p+\nabla\cdot(\varphi_c\hat{\boldsymbol{\tau}}_{eff})+ \qquad \text{(DispContinuous.2)}$$
$$+\varphi_c(\rho_c-\rho_{hyd})\boldsymbol{g}-\varphi_c\rho_c(2\boldsymbol{\omega}\times\boldsymbol{V}_c+\boldsymbol{\omega}\times\boldsymbol{\omega}\times\boldsymbol{r})-$$
$$-\varphi_c D\boldsymbol{V}-\varphi_c\hat{\boldsymbol{D}}\boldsymbol{V}-\boldsymbol{F}_D-\boldsymbol{F}_L$$

$$\boldsymbol{V}_c\otimes\boldsymbol{V}_c=\begin{Bmatrix}V_{c,x}V_{c,x} & V_{c,y}V_{c,x} & V_{c,z}V_{c,x}\\ V_{c,x}V_{c,y} & V_{c,y}V_{c,y} & V_{c,z}V_{c,y}\\ V_{c,x}V_{c,z} & V_{c,y}V_{c,z} & V_{c,z}V_{c,z}\end{Bmatrix} \qquad \text{(DispContinuous.3)}$$

$$\hat{\boldsymbol{\tau}}_{eff}=(\mu_c+\mu_{c,t})\left(2\hat{\boldsymbol{S}}-\frac{2}{3}(\nabla\cdot\boldsymbol{V}_c)\hat{\boldsymbol{I}}\right) \qquad \text{(DispContinuous.4)}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right)$$ (DispContinuous.5)

Here

$\hat{\boldsymbol{\tau}}_{eff}$ - effective shear stress tensor [Pa]

$\hat{\boldsymbol{S}}$ - strain rate tensor [s^{-1}]

$\hat{\boldsymbol{I}}$ - unit tensor

$\boldsymbol{\omega}$ - angular velocity of the rotating **Subregion** [s^{-1}]

$\boldsymbol{r}$ - radius-vector with origin at the rotation axis [m]

D - coefficient of isotropic resistance [kg m^{-3} s^{-1}]

$\hat{\boldsymbol{D}}$ - matrix coefficient of anisotropic resistance [kg m^{-3} s^{-1}] (see section **Modifier «Anisotropic resistance»**)

$\boldsymbol{F}_D$ - drag force of the whole of particles of the dispersed phase [N m^{-3}] (see section **Equations for particles > Process 'Motion'**)

$\boldsymbol{F}_L$ - lift force due to particles [N m^{-3}] (see section **Equations for particles > Process 'Motion'**)

Darcy model

In the Darcy model, differential momentum equation (DispContinuous.2) is not solved. Instead, an algebraic relationship between the fluid velocity and pressure gradient is assumed. Below examples of such a relationship are given:

$$\varphi_c \boldsymbol{V}_c = -\frac{k_d}{\mu_c}\nabla p + \frac{k_d}{\mu_c}\left(\rho_c - \rho_{hyd}\right)\boldsymbol{g}$$ (DispContinuous.6)

$$-\nabla p + \left(\rho_c - \rho_{hyd}\right)\boldsymbol{g} = D\boldsymbol{V}_c$$ (DispContinuous.7)

$$\varphi_c \boldsymbol{V}_c = -\frac{\hat{\boldsymbol{K}}_d}{\mu_c}\nabla p + \frac{\hat{\boldsymbol{K}}_d}{\mu_c}\left(\rho_c - \rho_{hyd}\right)\boldsymbol{g}$$ (DispContinuous.8)

$$-\nabla p + \left(\rho_c - \rho_{hyd}\right)\boldsymbol{g} = \hat{\boldsymbol{D}}\boldsymbol{V}_c$$ (DispContinuous.9)

Here

k_d - isotropic permeability of carcass (of solid immovable porous body) [m^2]

$\hat{\boldsymbol{K}}_d$ - anisotropic permeability of carcass [m^2]

Examples of model expressions for the resistance coefficient:

$$D = \frac{\mu_c}{k_d}\varphi_c$$ (DispContinuous.10)

$$\hat{\boldsymbol{D}} = \hat{\boldsymbol{K}}_d^{-1}\mu_c\varphi_c \qquad \text{(DispContinuous.11)}$$

$$D = \left(\frac{\mu_c}{k_d} + C\frac{1}{2}\rho_c|\boldsymbol{V}_c|\right)\varphi_c \qquad \text{(DispContinuous.12)}$$

$$D = 150\frac{\varphi_d^2}{\varphi_c^2}\frac{\mu_c}{d^2} + 1.75\frac{\varphi_d}{\varphi_c}\frac{\rho_c}{d}|\boldsymbol{V}_c| \qquad \text{(DispContinuous.13)}$$

Here

C - model coefficient

φ_d - relative volume of **Phase Carcass** in the Ergun model (DispContinuous.13)

d - diameter of closely packed spherical particles [m] in the Ergun model (DispContinuous.13)

10.5.8.5.2 Process 'Heat transfer'

The energy equation solved for thermodynamic enthalpy:

$$\frac{\partial(\rho_c\varphi_c h_c)}{\partial t} + \nabla(\rho_c\varphi_c\boldsymbol{V}_c h_c) = \frac{\partial(\varphi_c P)}{\partial t} + \varphi_c\boldsymbol{V}\cdot\nabla P - \nabla\cdot(\varphi_c\boldsymbol{J}_q) + \\ + \varphi_c Q_{vis,L} + \varphi_c Q_{rad} + Q_{user} - Q_d^{enth} + \varphi_d k_{HE}(T_d - T_c) + D_c T_c + F_c \qquad \text{(DispContinuous.14)}$$

$$\boldsymbol{J}_q = -\left(\lambda + \frac{\mu_t C_p}{Pr_t}\right)\nabla T + \sum_{i=\text{species}} h_i\boldsymbol{J}_i = -\left(\frac{\lambda}{C_p} + \frac{\mu_t}{Pr_t}\right)\nabla h + \\ + \sum_{i=\text{species}} h_i\nabla Y_i\left(\frac{\lambda}{C_p}\left(1 - Le_{i,mol}\right) + \frac{\mu_t}{Pr_t}\left(1 - Le_{i,t}\right)\right) \qquad \text{(DispContinuous.15)}$$

$$Q_{vis,L} = \sum_{i,j=1}^{3}\tau_{ij}S_{ij} + \rho\varepsilon \qquad \text{(DispContinuous.16)}$$

$$\tau_{ij} = \mu\left(2S_{ij} - \frac{2}{3}(\nabla\cdot V)\delta_{ij}\right) \qquad \text{(DispContinuous.17)}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i}\right) \qquad \text{(DispContinuous.18)}$$

$$Q_d^{enth} = n_d\left[\pi d Nu_d\lambda_c(T_c - T_d) + \pi d^2\sigma_{rad}\varepsilon_d\left(T_c^4 - T_p^4\right) - \pi d^2\sum_{i=1}^{Nd}\dot{m}_{d,i}h_{d,i}(T_d)\right] \qquad \text{(DispContinuous.19)}$$

The energy equation solved for total enthalpy:

$$\frac{\partial(\rho_c\varphi_c H_c)}{\partial t} + \nabla(\rho_c\varphi_c\boldsymbol{V}H_c) = \frac{\partial(\varphi_c P)}{\partial t} + \rho\boldsymbol{V}\cdot\boldsymbol{F} - \nabla\cdot(\varphi_c\boldsymbol{J}_q) + \\ + \varphi_c Q_{vis,G} + \varphi_c Q_{rad} + Q_{user} - Q_d^{enth} + \varphi_d k_{HE}(T_d - T_c) + D_c T_c + F_c \qquad \text{(DispContinuous.20)}$$

$$\boldsymbol{J}_q = -\left(\lambda + \frac{\mu_t C_p}{Pr_t}\right)\nabla T + \sum_{i=species} h_i \boldsymbol{J}_i =$$
$$= -\left(\frac{\lambda}{C_p} + \frac{\mu_t}{Pr_t}\right)\left(\nabla H - \boldsymbol{V}\cdot(\nabla\cdot\boldsymbol{V})\right) + \sum_{i=species} h_i \nabla Y_i \left(\frac{\lambda}{C_p}\left(1 - Le_{i,mol}\right) + \frac{\mu_t}{Pr_t}\left(1 - Le_{i,t}\right)\right) \quad \text{(DispContinuous.21)}$$

$$Q_{vis,G} = \nabla\cdot\left[\left(\mu + \mu_t\right)\left(2\hat{\boldsymbol{S}} - \frac{2}{3}(\nabla\cdot\boldsymbol{V})\hat{\boldsymbol{I}}\right)\cdot\boldsymbol{V}\right] \quad \text{(DispContinuous.22)}$$

$$Q_d^{enth} = n_d\left[\pi d Nu_d \lambda_c (T_c - T_d) + \pi d^2 \sigma_{rad}\varepsilon_d\left(T_c^4 - T_p^4\right) - \pi d^2 \sum_{i=1}^{Nd} \dot{m}_{d,i} h_{d,i}(T_d)\right] \quad \text{(DispContinuous.23)}$$

In equations (DispContinuous.14) and (DispContinuous.20):

k_{HE} - coefficient of heat exchange between **Continuous Phase** and **Phase Carcass**, which is set by parameter **Heat exchange coef. (D-C)** in the window of element **Models > Phase interaction > Continuum-dispersed**.

D_c - user's coefficient at relative temperature of **Continuous Phase**, which is set by parameter **D** in the window of element **Physical processes > Heat transfer** of **Continuous Phase**.

F_c - user's source term, which is set by parameter **F** in the window of element **Physical processes > Heat transfer** of **Continuous Phase**.

Q_{user} - user's source term, which is set by **Modifier** **Volume heat source** .

When evaporation of liquid droplets is simulated, the specific mass loss rate at the particles' surface $\dot{m}_d$ is determined by Eq. (DispParticles.23) (**Equations for particles > Process 'Mass transfer'**).

The particles' Nusselt number Nu_d is determined by Eq. (DispParticles.18) (**Equations for particles > Process 'Heat transfer'**).

10.5.8.5.3 Process 'Mass transfer'

The convection-diffusion equation for the mass fraction of the i-th component:

$$\frac{\partial(\varphi_c \rho Y_i)}{\partial t} + \nabla\cdot(\varphi_c \rho Y_i \boldsymbol{V}) = -\nabla\cdot(\varphi_c \boldsymbol{J}_{i,eff}) + \varphi_c W_i + Q_{d,i}^{mass} \quad \text{(DispContinuous.24)}$$

$$\boldsymbol{J}_{i,eff} = -\left(\rho D_i + \frac{\mu_t}{Sc_t}\right)\nabla Y_i = -\left(\frac{\mu}{Sc_i} + \frac{\mu_t}{Sc_t}\right)\nabla Y_i \quad \text{(DispContinuous.25)}$$

$$W_i = D_{s,i}(T, P, Y_1, ..., Y_N)\cdot Y_i + F_{s,i}(T, P, Y_1, ..., Y_N) \quad \text{(DispContinuous.26)}$$

Here

$Q_{d,i}^{mass}$ - mass source of the *i*-th **Substance** of **Continuous Phase** due to change of mass of **Phase Particles** [kg m^{-3} s^{-1}]

Y_i - mass fraction of the *i*-th **Substance** of **Continuous Phase**

$\boldsymbol{J}_{i,eff}$ - effective diffusive flux of the *i*-th **Substance** of **Continuous Phase** [kg m^{-2} s^{-1}]

$Sc_i = \frac{\mu}{\rho D_i}$ - molecular Schmidt number

$Sc_t = \frac{\mu_t}{\rho D_t}$ - turbulent Schmidt number

D_i - effective diffusion coefficient for the *i*-th **Substance** of **Continuous Phase** [$m^2\ s^{-1}$]

D_t - turbulent diffusion coefficient for **Continuous Phase** [$m^2\ s^{-1}$]

If **Phase Particles** consists of one substance,

$$Q_{d,i}^{mass} = Q_d^{mass} \quad \text{(DispContinuous.27)}$$

Source term in the continuity equation Q_d^{mass} is determined by Eq. (DispParticles.22) (**Equations for particles > Process 'Mass transfer'**).

10.5.8.5.4 Process 'Turbulence'

Below the equations of model KES, modified with account of presence of **Phase Particles** in the computational domain, are submitted. Models **KEAKN**, **KEFV**, **KENL** are modified in a similar way.

$$\frac{\partial(\varphi_c\rho_c k_c)}{\partial t} + \nabla(\varphi_c\rho_c \boldsymbol{V}_c k_c) = \nabla\left(\varphi_c\left(\mu_c + \frac{\mu_{t,c}}{\sigma_k}\right)\nabla k_c\right) + \varphi_c\rho_c\left[P_k + G_k - \varepsilon_c\left(1 + 1.5\left(max\left(M_t^2, M_t^{\cdot}\right.\right.\right.\right. \quad \text{(DispContinuous.28)}$$

$$\frac{\partial(\varphi_c\rho_c \varepsilon_c)}{\partial t} + \nabla(\varphi_c\rho_c \boldsymbol{V}_c\varepsilon_c) = \nabla\left(\varphi_c\left(\mu_c + \frac{\mu_{t,c}}{\sigma_\varepsilon}\right)\nabla\varepsilon_c\right) + \varphi_c\rho_c\frac{\varepsilon_c}{k_c}\left[C_{\varepsilon 1}\left(P_k + G_k\right) - C_{\varepsilon 2}\varepsilon_c\right] + Q_{\varepsilon,d}^{turb} \quad \text{(DispContinuous.29)}$$

The extra terms due to particles are modeled as follows (see [1]):

$$Q_{k,d}^{turb} = C_{kD} n_d \rho_c C_D \frac{\pi d^3}{8}\left|\boldsymbol{V}_c - \boldsymbol{V}_d\right|^3 \quad \text{(DispContinuous.30)}$$

$$Q_{\varepsilon,d}^{turb} = C_{\varepsilon D} n_d \rho_c C_D \frac{\pi d^3}{8}\left|\boldsymbol{V}_c - \boldsymbol{V}_d\right|^3 \frac{\sqrt{k}}{d} \quad \text{(DispContinuous.31)}$$

Model constants C_{kD} and $C_{\varepsilon D}$ are specified in the *FlowVision* interface.

Account of particles in models **SST** and **SA** requires additional study.

10.5.8.6 Crystallization

The crystallization model (see [12]) can be applied to simulate fallout of particles (drops) of the dispersed phase from a two-phase flow on the surface of the geometry model or on the surface of the solid phase with forming, depending on parameters of the flow, either:

- liquid film formed by the substance of the dispersed phase
- *or* solid phase formed by the substance of the dispersed phase.

The following situations are possible:

- the film moves (flows) over the surface of the geometry model
- the film moves (flows) over the surface of the formed solid phase
- the dispersed phase immediately turns into the solid phase without stage of forming the film

The following models of crystallization the dispersed phase are implemented:

- **Dry model**
- **Film model**

The **Dry model** assumes absence of the film on the surface. This model takes into account sublimation of the dispersed phase's substance from the surface of the solid phase (which has been formed by dispersed phase's

substance). The **Dry model** can be recommended for cases when absence of the film on the surface is evident a priori (for example, when clearly low temperature of the forward two-phase flow is specified).

The more broad **Film model** allows you to take into account processes of mass and heat transfer during spreading the liquid film over the surface and also to take into account processes of evaporating the dispersed phase's substance from the surface of the liquid film and sublimating the dispersed phase's substance from the surface of the solid phase (which has been formed by the dispersed phase's substance). Simulating the mass and heat transfer in the liquid film can be done with its own individual time step, which differs from the common time step of the simulation. This step is specified by the **Film CFL** parameter.

The mass source of the solid phase, which is formed by crystallization the dispersed phase's substance, is used as the source of substance in simulations of phase transfer. Simulating the growth of the solid phase with use the mass source of the dispersed phase's substance can have its own individual time step, which differs from the common time step. This step is specified by the **CFL for VOF source** parameter.

When icing of aircraft is simulated, you are to keep the default value of this parameter, **Use VOF source for time step = No**.

When icing is simulated, it is recommended to use the **Relative** criterion for revealing small cells for the solid-state **Phase** (ice body); this is set by the **Small Cells > Criterion** parameter in properties of the element **Limiters > Limiters for calculation > Phase Limiters > Phase #N**.

Thermal balance on the inter-phase surface

In the **Dry model** the value of the specific mass velocity of crystallization of the dispersed phase's substance is determined by the formulae:

$$\dot{m}_s = \dot{m}_d - \dot{m}_{subl} \qquad \text{(Cryst.1)}$$

$$\dot{m}_d = \rho_d \varphi_d \boldsymbol{V}_d \cdot \vec{\boldsymbol{n}} \qquad \text{(Cryst.2)}$$

where

$\vec{\boldsymbol{n}}$ is the local normal to the surface

$\dot{m}_{subl}$ is the specific mass flow of the substance from the surface of the solid phase due to sublimating the solid phase

In the **Film model** the value of the specific mass velocity of either crystallization or melting of the solid phase's substance is determined by solution of the energy balance equation on the inter-phase surface «two-phase flow – film» or «two-phase flow – solid phase»:

$$\left[\left(h_d - h_f\right) + \frac{1}{2}\boldsymbol{V}_d^{\,2}\right]\dot{m}_d + \left(\lambda_c + \lambda_{c,t}\right)\frac{T_{c,cell} - T_f}{y_{c,cell}} - \dot{m}_{evap(subl)}\Delta h_{evap(subl)} = \lambda_s \frac{T_f - T_{c,cell}}{y_{s,cell}} - \dot{m}_s \Delta h_{fusion} \qquad \text{(Cryst.3)}$$

where

$\dot{m}_{evap(subl)}$ is the specific mass flow from the surface of the solid phase due to evaporating the film or sublimating the solid phase

It is assumed that temperature of the film on the surface of the solid phase T_f is equal to the crystallization temperature of the dispersed phase's substance. Equation (Cryst.3) is solved iteratively. As the iterations are carried out, “dry” and “wet” zones on the surface of the solid phase are determined. In absence of the solid phase, temperature of the film is determined from solution of the conjugate problem of heat exchange between the film and the surface, which is being flown around. The specific mass velocity of forming the film from the dispersed phase's substance is determined by the formula:

$$\dot{m}_f = \dot{m}_d - \dot{m}_s - \dot{m}_{evap(subl)} \qquad \text{(Cryst.4)}$$

Roughness of the ice surface

The crystallization model assumes that the solid surface becomes rough because of because of sequential growth and crystallization of subcooled drops of the dispersed phase.

To calculate the equivalent sand roughness k_s of the surface, empiric roughness models are applied, **Shin-Bond** and **Shin-Bond (local)**. These models use correlations with experimental results of aircraft icing.

Shin-Bond model:

$$k_s = k_0 \cdot 0.6839 \cdot \left(0.047 \cdot T_\infty - 11.27\right)\left(0.571 + 0.246 \cdot LWC_\infty + 1.257 \cdot \left(LWC_\infty\right)^2\right) \quad \text{(Cryst.5)}$$

Shin-Bond (local) model:

$$k_s = k_0 \cdot 0.6839 \cdot \left(0.047 \cdot T_{cell} - 11.27\right)\left(0.571 + 0.246 \cdot LWC_{cell} + 1.257 \cdot \left(LWC_{cell}\right)^2\right) \quad \text{(Cryst.6)}$$

where

k_0=0.000682 is the model's constant

T_∞ is temperature of the undisturbed two-phase flow

$LWC_\infty = 1000 \cdot (\varphi_d \rho_d)_\infty$ is liquid water content of the undisturbed two-phase flow, [g/m^3]

$LWC_{cell} = 1000 \cdot (\varphi_d \rho_d)_{cell}$ is local liquid water content, [g/m^3]

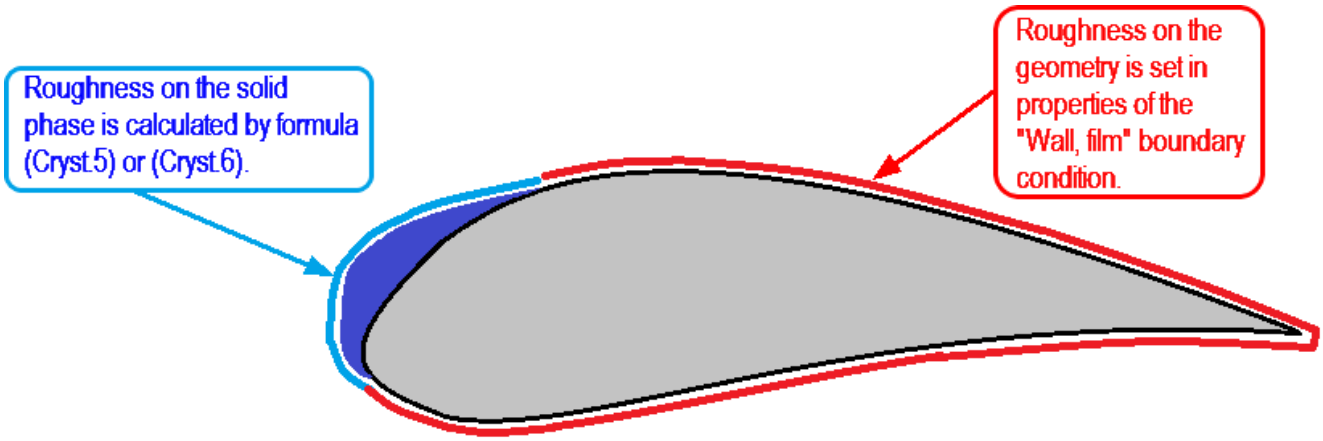
The roughness model is selected by the **Ice roughness model** parameter.

The equivalent sand roughness is applied similarly as the **Roughness**, which is set in properties of the **Wall** boundary condition.

On surface of the solid phase (ice) the program applies the equivalent sand roughness, which is calculated by formula (Cryst.5) or (Cryst.6).

On surface of the geometry (on which the **Wall, film** boundary condition is set) the program applies the equivalent sand roughness, which is set in properties of the boundary condition.

The equivalent sand roughness is taken into account in the semiempirical model of mass flow of the dispersed phase's substance from the surface due to evaporating the film or sublimating the solid phase, see subsection *"Evaporating/sublimating from the surface of the film or solid phase"* below.



Evaporating/sublimating from the surface of the film or solid phase

When evaporation/sublimation of the film's substance is simulated, you have to specify pairs of **Substances** that correspond to the liquid and vapor states of the film's substances.

These pairs of **Substances** are specified as elements of the **Substance pair** array in properties of the **Phase interaction** element for the **Continuum-particles** phase pair, which correspond to the carrier phase (when icing is simulated: the air) and the dispersed phase of particles (when icing is simulated: water droplets).

If the film and the particles contain only one **Substance**, you fill only one element in the **Substance pair** array. If the film and the particles contain multiple **Substances** (for example, they contain mix of water and ethanol), you have to fill in several elements of the array. For example, when the film and particles consist of mix of water and ethanol, you specify water and water vapor in one of the elements of the **Substance pair** array, and specify liquid and gaseous ethanol in another element of the array.

If the **Substance pair** is not set in such situation, then the "**!**" symbol will appear near the appropriate **Phase**

interaction element, and the error message "**In the phase interaction, a pair of substances with a vapor-forming substance for the film is not defined (Continuum-particles).**" will be displayed in the **Log** window.

The specific mass flow of the dispersed phase's substance from the surface, which is caused by evaporating the film or sublimating the solid phase, is calculated using the following algebraic relations:

$$\dot{m}_{evap(subl)} \cdot \left(1 - Y_{vapor,f}\right) = \rho_c D_{vapor} \frac{Y_{vapor,f} - Y_{vapor,cell}}{y_{c,cell}} \quad \text{(Cryst.7)}$$

$$Y_{vapor,f} = X_{vapor,f} \frac{m_{vapor}}{m_{f,c}} \quad \text{(Cryst.8)}$$

$$m_{f,c} = X_{vapor,f} m_{vapor} + \left(1 - X_{vapor,f}\right) m_c \quad \text{(Cryst.9)}$$

$$X_{vapor,f} = \frac{p_{vapor}\left(T_f\right)}{p} \quad \text{(Cryst.10)}$$

where

$Y_{vapor,cell}$ is the mass fraction of the vapor in the center of the cell adjacent to the surface (this value is found in solution of mass transfer equations for the continuous phase),

$Y_{vapor,f}$ is the mass fraction of the vapor on the surface of the film or the solid phase,

$m_{f,c}$ is the molar mass of the vapor,

$X_{vapor,f}$ is the molar fraction of the vapor on the surface of the film or the solid phase,

$p_{vapor}(T_f)$ is the partial pressure of the saturated vapor that depends on temperature of the surface (it is determined by properties of the gaseous state of the dispersed phase's substance),

D_{vapor} is the binary diffusion coefficient of the vapor.

The diffusion coefficient of the vapor on the surface of the film (or the solid phase) is calculated using the diffusive model of evaporation/sublimation, which is implemented in the program.

Convective mass transfer in the film

The film is not resolved by the computational grid and is assumed as incompressible. Mass transfer in the film is simulated by the equation of transfer of the film's height. This equation arises out of the mass conservation law for the film:

$$\frac{\partial w_f}{\partial t} + \nabla \cdot \left(\boldsymbol{V}_f w_f\right) = \frac{1}{\rho_d} \dot{m}_f \quad \text{(Cryst.11)}$$

The mean velocity of the film is assumed as linearly dependent on the film's height:

$$\boldsymbol{V}_f = \frac{w_f}{2\mu_f} \boldsymbol{\tau}_w + \frac{w_f^2}{3\nu_f} \left(\boldsymbol{g}^{\tau} - \boldsymbol{g}^{n} \cdot \nabla w_f + \beta \frac{\sigma\left(\cos\theta - 1\right)}{\rho_f w_f \Delta_{\perp}^{cl}} \boldsymbol{n}^{cl} \right) \quad \text{(Cryst.12)}$$

where

ν_f is kinematic coefficient of viscosity of the film's substance

$\boldsymbol{g}^{\tau}$ is the gravity vector's component, which is tangential to the surface, on which the film moves

$\boldsymbol{g}^{n}$ is the gravity vector's component, which is normal to the surface, on which the film moves

$\boldsymbol{g}^{n} \cdot \nabla w_f$ is the term, which determines how the gradient of hydrostatic pressure in the film takes part in the film's motion

$\beta \frac{\sigma\left(\cos\theta - 1\right)}{\rho_f w_f \Delta_{\perp}^{cl}} \boldsymbol{n}^{cl}$ is the term, which is determined by the wetting model (it specifies how the surface tension force, which acts orthogonally to the contact line between the film and the solid surface, takes part in the film's

motion). This term is applied only in cells that contain edge of the film (i.e. the cells with contact of: the surface, on which the film moves; the film itself; the medium, in which the film moves).

β is the coefficient of the wetting model, which is set by the **Wetting model parameters > Model coefficient** parameter

σ is the surface tension coefficient of the film's substance

θ is the local wetting (contact) angle on the solid phase. This is a random value; is mathematical expectation is set by the **Wetting model parameters > Contact angle with solid phase** parameter and its mean-square deviation is set by the **Wetting model parameters > Wetting angle deviation** parameter. The distribution law of the random value θ is specified by the **Wetting model parameters > Model of wetting angle deviation** parameter (the **Normal distribution** of θ only is implemented now).

$\boldsymbol{n}^{cl}$ is the normal to the contact line in a local cell on the surface, on which the film moves. The normal is oriented into the wet part of the surface.

$\Delta^{cl}{}_{\perp}$ is the size of the section of a cell by local plane of the surface, on which the film moves. This size is measured in direction, which is perpendicular to the contact line.

The film's height transfer equation is solved using an explicit scheme.

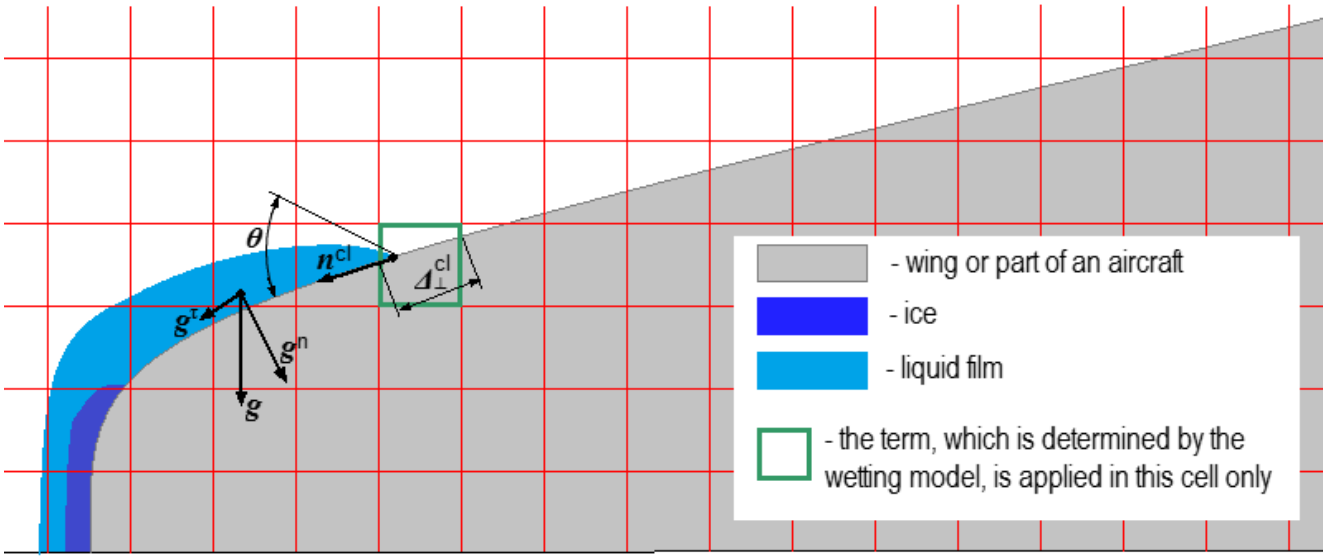


Illustration for formula Cryst.12

Due to use a random data generator, results of individual simulations of the film's flows can be different in various computations (in reality the flows also are formed randomly).

Convective heat transfer in the film

The film is not resolved by the computational grid and is assumed as incompressible. Heat transfer in the film is simulated by the equation of transfer of the film's enthalpy. This equation arises out of the energy conservation law for the film:

$$\frac{\partial h_f}{\partial t} + \nabla \cdot \left(\boldsymbol{V}_f h_f \right) = 0 \qquad \text{(Cryst.13)}$$

The film's enthalpy transfer equation is solved using an explicit scheme.

Initial conditions

User specifies values of the film's height w_f [m] and the film's temperature T_f [K].

Simulating of the film's shedding

The film's shedding model is enabled by the **Film shedding model** parameter in properties of the **Crystallization** physical process.

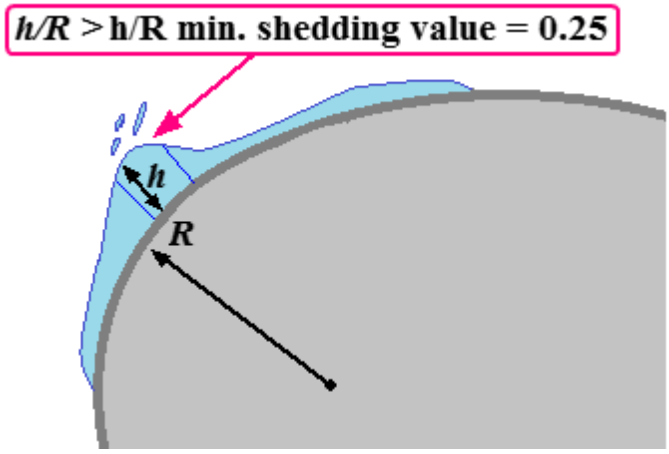
This parameter is available when **Crystallization = Film model** is set in properties of the **Physical processes** folder of the dispersed **Phase** (particles).

Film shedding model "Parametrical model"

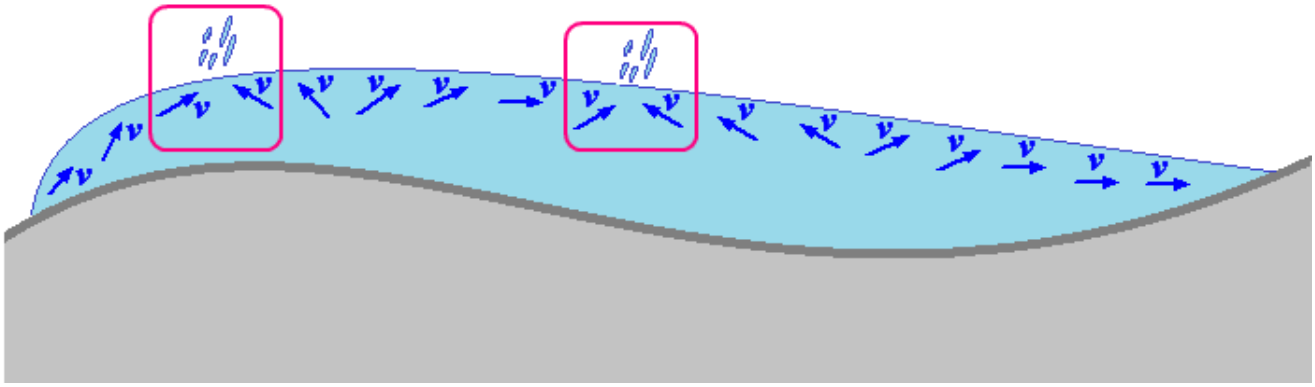
The film's shedding from the surface occurs when any of the following conditions takes place when:

- ratio of the film's thickness h to the radius R of curvature of the surface in the near-surface cell exceeds the value of the **h/R min. shedding value** parameter (see below)
- *or* the film flows in opposite directions in adjacent near-surface cells (*formal description:* when projections of vectors of the film's velocity in adjacent near-surface cells to the normal to the face between these cells are directed towards to each other).

See illustrations below.



The film's shedding due to h/R exceeds the threshold value **h/R min. shedding value** (**h/R min. shedding value = 0.25** in this example)



The film's shedding due to opposite flow of the film in adjacent near-surface cells

10.5.8.7 Coal combustion

The coal combustion model implemented in the *FlowVision* software is based on the following assumptions:

1. A solid particle is a capsule containing predefined mass fractions of the following **Substances**:
   - **WATER**
   - **VOLATILES**
   - **CHAR**
   - **ASH**
   - **NITROGEN**
   - **SULFUR**
2. The particle-related processes cannot proceed simultaneously. The sequence of the processes is as follows:
   1. drying (evaporation of liquid water

2. pyrolysis (devolatization)
3. char burning.

It is assumed that while a particle releases volatiles, it transforms into char. The char is mixed with ash.

3. Volatiles are a gas mixture represented by formula CxHyOz. In coal combustion model of *FlowVision*, this is one **Substance**. Values x, y, z and the properties of this **Substance** are computed outside *FlowVision* using known elemental composition (ultimate analysis). The enthalpy of formation of **Substance** CxHyOz at 298.15 K is computed using known lower heating value (LHV) for a given rank of coal.

Boundary conditions

On boundary conditions **Inlet/Outlet** the program automatically specifies the method of setting for the following variables:

- **Diameter** – either **Size spectrum #N** (when a spectrum consisting of several size groups of coal particles is simulated) or **Coal** (when size groups are not simulated). When size groups are simulated, you have to select here some of previously prepared **Size spectra** of the coal particles. When size groups are not simulated, you don't have to specify anything here.
- **Mass frac. [*Substance of coal particles*]** – **Coal**. You don't have to specify anything here; mass fractions of **Substances** of coal particles are specified by parameters **Composition > ...** in properties of the dispersed **Phase** that corresponds to the coal particles.

This section has individual numeration of equations.

10.5.8.7.1 Notations

Notation	Physical quantity	Name in FlowVision	Dimension
C_D	drag coefficient for particles		
$C_{p,i}$	specific heat of **Substance** *i* at constant pressure	**Specific heat**	$m^2\ s^{-2}\ K^{-1}$
C_p	specific heat of **Phase** (mixture) at constant pressure	**Specific heat**	$m^2\ s^{-2}\ K^{-1}$
d	local diameter of particles	**Diameter**	m
h_i	thermodynamic enthalpy of **Substance** i	**Enthalpy**	$m^2\ s^{-2}$
h_d	thermodynamic enthalpy of particles	**Enthalpy (disp.)**	$m^2\ s^{-2}$
h_i^0	enthalpy of formation of **Substance** i at 298.15 K	**Enthalpy of formation**	$m^2\ s^{-2}$
M_d	local mass of particles	**Mass**	kg
m_i	molar mass of **Substance** *i* or chemical element i	**Molar mass**	$kg\ mole^{-1}$
m	molar mass of **Phase** (mixture)	**Molar mass**	$kg\ mole^{-1}$
$\dot{m}_d$	mass loss rate		$kg\ s^{-1}$
$\dot{m}_{d,1}$	mass loss rate in the course of particles drying		$kg\ s^{-1}$
$\dot{m}_{d,2}$	mass loss rate in the course of pyrolysis		$kg\ s^{-1}$
$\dot{m}_{d,3}$	mass loss rate in the course of char burning		$kg\ s^{-1}$
n_d	concentration of particles	**Concentration**	m^{-3}
Nu_d	Nusselt number for particles		

Notation	Physical quantity	Name in FlowVision	Dimension
$P_{H2O,sat}(T_d)$	partial pressure of saturated water vapor		
Q_i	source term for **Substance** i		kg m^{-3} s^{-1}
Sc	Schmidt number for gas		
$Sc_{t,d}$	turbulent Schmidt number for particles		
Sh	Sherwood number for particles		
V_d	velocity of particles	**Velocity (disp.)**	m s^{-1}
X_i	molar fraction of **Substance** i or chemical element i		
$Y_{WATER,d}$	mass fraction of liquid water in coal particle	**Mass frac. [WATER]**	
$Y_{VOL,d}$	mass fraction of volatiles in coal particle	**Mass frac. [VOLATILES]**	
$Y_{CHAR,d}$	mass fraction of char in coal particle	**Mass frac. [CHAR]**	
$Y_{ASH,d}$	mass fraction of ash in coal particle	**Mass frac. [ASH]**	
$Y_{N,d}$	mass fraction of element *N* in coal particle	**Mass frac. [NITROGEN]**	
$Y_{S,d}$	mass fraction of element *S* in coal particle	**Mass frac. [SULFUR]**	
$Y_{C,VOL}$	mass fraction of element *C* in volatiles		
$Y_{H,VOL}$	mass fraction of element *H* in volatiles		
$Y_{O,VOL}$	mass fraction of element *O* in volatiles		
μ	dynamic coefficient of viscosity of gas	**Viscosity**	Pa s = kg m^{-1} s^{-1}
$0 \le \varepsilon_{rad} \le 1$	emissivity of particles		
$\sigma_{rad} = 5.67 \cdot 10^{-8}$	Stefan-Boltzmann constant		W m^{-2} K^{-4}

Indices:

c - **Continuous Phase** (gas mixture)
d - **Phase Particles** (dispersed phase)
ini - initial (inlet) value
sat - saturation conditions
t - turbulent value

10.5.8.7.2 Parameters

Parameters of gas medium, where combustion of coal particles occurs, are set in properties of the element **Preprocessor > Phases > Phase #N > Physical processes > Mass transfer** for a continuous **Phase #N**, see description in the table *Parameters of the "Physical processes > Mass transfer" element (for continuous phases)* in section Folder «Phases».

Parameters for coal particles are set in properties of the element **Preprocessor > Phases > Phase #N > Physical processes > Mass transfer** for a dispersed **Phase #N** of the **Particles** type:

Parameters of the "Physical process > Mass transfer" element for dispersed Phases of the Particles type and the the Coal mass transfer model		
Parameter	**Possible values**	**Description**
Math. model	• **Mass transfer** • **Coal**	A model of the physical process. (This parameter cannot be edited here, its value is selected in properties of the *folder* **Phase N > Physical processes**.)
Time step coefficient	An arbitrary numerical value determined by the problem (the default value is **1**) Individual time step for the given process = general time step x **Time step coefficient**. Specifying different time steps for different processes sometimes accelerates convergence to the steady-state solution. Meaning of different values: **1** = The general time step is used for the calculations of the given process (the calculations of the given process is synchronized with the calculations of the problem as a whole). **> 1** = The calculations of the given process are accelerated. **(0, 1)** = The calculations of the given process are decelerated. **< -1** = The calculations of the given process are ceased (the distributions of the variables characterizing the process are left untouched after preceding calculations / initialization).	This is a coefficient that is equal to the ratio of the individual time step, which is used to calculate the physical process, to the common time step of the whole project (τ). *Thus:* Own time step of a physical process = τ × **Time step coefficient** Specifying individual time steps for different processes allows you to obtain a steady-state solution in less time.
D	Determined by the problem. (Default value: **0**)	Expression for quantity D in the source term **D * MASS_D + F** (do not multiply by particles concentration)
F	Determined by the problem. (Default value: **0**)	Expression for quantity F in the in source term **D * MASS_D + F** (do not multiply by particles concentration)
Model for particles	• **Variable diameter** • **Constant diameter**	Model of transforming the coal particles. The **Variable diameter** model assumes that the density of a particle remains constant while the

Parameters of the "Physical process > Mass transfer" element for dispersed Phases of the Particles type and the the Coal mass transfer model		
Parameter	**Possible values**	**Description**
		diameter changes according to the mass loss of the **Particles Phase**. The **Constant diameter** model assumes that the diameter of a particle remains constant while the density changes according to the mass loss of the **Particles Phase**. See subsection *Models for transformation of particles*.
dens. initial [1]	near **1000**	Initial density of the coal particles, [kg/m^3]
Composition > ... [1]	$0 \le Y_{i,d} < 1$	Mass fractions $Y_{i,d}$ of the coal components (**Substances**) that are included in the dispersed **Phase**, which corresponds to coal particles. This list has to include substances **WATER**, **VOLATILES**, **CHAR**, and **ASH**. It is not necessary to include substances **NITROGEN** and **SULFUR** into the **Phase**. Sum of all values $Y_{i,d}$ is to be equal to **1** (keeping this rule is the the user's responsibility).
A_pyr [1]	Determined by the problem	Parameters of the pyrolysis model. They correspond to A_{pyr}, n_{pyr}, T_{pyr} in formula (Coal.20).
n_pyr [1]		
T_pyr [1]		
C_diff [1]	Determined by the problem	Parameter of the char burning model. They correspond to C_{dif}, A_{CO2}, n_{CO2}, T_{CO2} in formulae (Coal.23) and (Coal.24).
A_CO2 [1]		
n_CO2 [1]		
T_CO2 [1]		
C_NOx [1]	Determined by the problem	Parameter of the model for NOx formation. It corresponds to C_{NOx} in formula (Coal.25).
C_SOx [1]	Determined by the problem	Parameter of the model for SOx formation. It corresponds to C_{SOx} in formula (Coal.25).
LastWV	[**0**; **1**]	This parameter determines the mass fractions of remaining **WATER** and **VOLATILES** that leave a particle per one time step, so this speeds up transfer to the next process in the multistage model of destruction of coal particles. Value **0.1** is recommended.

[1] Parameter becomes accessible after selecting the **Coal** model.

The other parameters are described in sections:

- Processes in the presence of dispersed medium > Parameters > Process 'Phase transfer'
- Processes in the presence of medium > Parameters > Process 'Motion'
- Processes in the presence of medium > Parameters > Process 'Heat transfer'

FlowVision can simulate multiple size groups of coal particles. In this case the diameter of coal particles in the **Initial data** and on boundary conditions **Inlet/outlet** is to be matched with some of previously prepared **Size spectra** of the coal particles.

10.5.8.7.3 Fuel

A coal particle is a capsule containing pre-defined mass fractions of:

- **WATER**
- **VOLATILES**
- **CHAR**
- **ASH**
- **NITROGEN**
- **SULFUR**

Apparently,

$$Y_{WATER,d} + Y_{VOL,d} + Y_{CHAR,d} + Y_{ASH,d} + Y_{N,d} + Y_{S,d} = 1 \quad \text{(Coal.1)}$$

Generalized formula for volatiles

Volatiles are a gas mixture of H2O, CO2, CO, H2, CH4 and other hydro-carbons. Let the elemental composition (C, H, O) of this mixture is known:

$$Y_{C,VOL} + Y_{H,VOL} + Y_{O,VOL} = 1 \quad \text{(Coal.2)}$$

Submit the C, H, O mixture by generalized formula CxHyOz. For this purpose

1. Compute the molar mass of the mixture:

$$\frac{1}{m} = \frac{Y_C}{m_C} + \frac{Y_H}{m_H} + \frac{Y_O}{m_O} \quad \text{(Coal.3)}$$

2. Compute the molar fractions of elements C, H, O:

$$x = X_C = Y_C \cdot \frac{m}{m_C} \quad \text{(Coal.4)}$$

$$y = X_H = Y_H \cdot \frac{m}{m_H} \quad \text{(Coal.5)}$$

$$z = X_O = Y_O \cdot \frac{m}{m_O} \quad \text{(Coal.6)}$$

Stoichiometric coefficients

Let the following gross-reaction proceeds in the gas phase (molar formula):

$$C_x H_y O_z + (x + 0.25y - 0.5z) O_2 \rightarrow x CO_2 + 0.5y H_2O \quad \text{(Coal.7)}$$

The corresponding mass formula reeds:

$$C_x H_y O_z + i_{O2} O_2 \rightarrow i_{CO2} CO_2 + i_{H2O} H_2O \quad \text{(Coal.8)}$$

Here

$i_{O2} = \frac{32 \cdot (x + 0.25y - 0.5z)}{12x + y + 16z}$ is stoichiometric coefficient at **Oxidizer**,

$i_{CO2} = \frac{44 \cdot x}{12x + y + 16z}$ is stoichiometric coefficient at **Product** CO2,

$i_{H2O} = \frac{18 \cdot 0.5y}{12x + y + 16z}$ is stoichiometric coefficient at **Product** H2O.

Lower heating value of coal

The lower heating value for a given rank of coal “as received” is determined by the following expression:

$$LHV_{coal} = \left(h^0_{WATER} - h^0_{H2O}\right) Y_{WATER,d} + \left(h^0_{VOL} - h^0_{CO2} \frac{44x}{12x+y+16z} - h^0_{H2O} \frac{9y}{12x+y+16z} \right) \cdot Y_{VOL,d} + \left(h^0_{CHAR} - \frac{11}{3} h^0_{CO2} \right) \cdot Y_{CHAR,d} \quad \text{(Coal.9)}$$

Here

h^0_i - enthalpy of formation of **Substance** i at 298.15 K [$m^2\ s^{-2}$]

Values h^0_{WATER}, h^0_{H2O}, h^0_{CO2} are known. It is reasonable to assume that $h^0_{CHAR} = 0$. Then $h^0_{VOL} = 0$ can be computed from Eq. (Coal.9).

Specific heat of coal

The *FlowVision*'s model for coal combustion assumes:

$$C_{p,d}(T_d) = C_{p,WATER}(T_d) Y_{WATER,d} + C_{p,VOL}(T_d) Y_{VOL} + = C_{p,CHAR}(T_d) Y_{CHAR} + C_{p,ASH}(T_d) Y_{ASH} \quad \text{(Coal.10)}$$

$$Y_{WATER,d} + Y_{VOL} + Y_{CHAR} + Y_{ASH} = 1 \quad \text{(Coal.11)}$$

Here the mass fractions of **Substances** N and S are added to the mass fraction of **Substance CHAR**. Normally, $Y_{N,d}$ and $Y_{S,d}$ are small. Therefore, elements N and S do not affect thermodynamics.

10.5.8.7.4 Equations for particles

Process "Phase transfer"

$$\frac{\partial n_d}{\partial t} + \nabla \cdot (\boldsymbol{V}_d n_d) = \nabla \cdot \left(\frac{\nu_{t,d}}{Sc_{t,d}} \nabla n_d \right) \quad \text{(Coal.12)}$$

Process "Mass transfer"

$$\frac{\partial (M_d n_d)}{\partial t} + \vec{\nabla} \cdot (\boldsymbol{V}_d M_d n_d) = \vec{\nabla} \cdot \left(\frac{\nu_{t,d}}{Sc_{t,d}} \nabla (M_d n_d) \right) - n_d \pi d^2 \dot{m}_d \quad \text{(Coal.13a)}$$

$$\frac{\partial (Y_{d,i} M_d n_d)}{\partial t} + \vec{\nabla} \cdot (Y_{d,i} M_d n_d \boldsymbol{V}_d) = \nabla \cdot \left(\left(\frac{\nu_{t,p}}{Sc_{t,p}} \nabla (Y_{d,i} M_d n_d) \right) \right) - n_d \pi d^2 \dot{m}_{d,i} \quad \text{(Coal.13b)}$$

Here:

$\dot{m}_d = \sum_{i=1}^{3} \dot{m}_{d,i}$ is mass loss rate for **Phase Particles** (coal).

$\dot{m}_{d,i}$ is mass loss rate for one of three **Substances**: **WATER**, **VOLATILES**, or **CHAR**.

$Y_{d,i}$ is mass fraction of **Substance WATER**, **VOLATILES**, or **CHAR**.

Boundary conditions **Coal** are automatically set in template **"Inlet/Outlet"** for variables **Diameter**, **Mass frac. [WATER]**, **Mass frac. [VOLATILES]**, **Mass frac. [CHAR]**. These boundary conditions use the values of physical quantities specified in properties of **Preprocessor > Phases > Phase #i** and in properties of **Preprocessor > Phases > Phase #i > Physical processes > Mass transfer**. So, user specifies nothing for the aforementioned 4 variables at two-phase inlets.

Process "Motion"

$$\frac{\partial\left(V_{d,i}M_d n_d\right)}{\partial t}+\nabla\cdot\left(\boldsymbol{V}_d V_{d,i}M_d n_d\right)=\nabla\cdot\left(\frac{\nu_{t,d}}{Sc_{t,d}}\nabla\left(V_{d,i}M_d n_d\right)\right)- \tag{Coal.14}$$

$$-n_d\frac{\pi d^3}{6}\nabla_i P+n_d M_d g_i+n_d\rho_c C_D\frac{\pi d^2}{8}\left|\boldsymbol{V}_c-\boldsymbol{V}_d\right|\left(V_{c,i}-V_{d,i}\right)$$

Here the index i is used for a Cartesian component of the corresponding vector.

Process "Heat transfer"

$$\frac{\partial\left(h_d M_d n_d\right)}{\partial t}+\nabla\left(\boldsymbol{V}_d h_d M_d n_d\right)=\nabla\cdot\left(\left(\frac{\nu_{t,p}}{Sc_{t,p}}\nabla\left(h_d M_d n_d\right)\right)\right)+ \tag{Coal.15}$$

$$+n_d\left[\pi d Nu_d\lambda_c\left(T_c-T_d\right)+\pi d^2\sigma_{rad}\varepsilon_{rad}\left(T_c^4-T_d^4\right)-\pi d^2\sum_{i=1}^{3}\dot{m}_{d,i}h_{d,i}\left(T_d\right)\right]$$

Here:

$0\le\varepsilon_{rad}\le 1$ is emissivity of particles.

Drying

The rate of the particles drying $\dot{m}_{d,1}$ is determined by dependency $P_{H2O,sat}\left(T_d\right)$:

$$\dot{m}_{d,1}=\pi d^2\cdot\frac{Sh}{Sc}\frac{\mu_c}{d}\cdot\frac{Y_{H2O,sat}\left(T_d\right)-Y_{H2O,c}}{1-Y_{H2O,sat}\left(T_d\right)} \tag{Coal.16}$$

$$Y_{H2O,sat}=\frac{X_{H2O,sat}}{X_{H2O,sat}+\left(1-X_{H2O,sat}\right)\dfrac{1-Y_{H2O,c}}{\dfrac{m_{H2O}}{m_c}-Y_{H2O,c}}} \tag{Coal.17}$$

$$X_{H2O,sat}=\frac{P_{H2O,sat}\left(T_d\right)}{P} \tag{Coal.18}$$

Here:

d is local average (over a cell) diameter of particles

μ_c is dynamic coefficient of viscosity for Coal. (gas mixture)

Sh is Sherwood number for particles

$Sc_c=\dfrac{\mu_c}{\rho_c D}$ is Schmidt number for **Continuous Phase** (gas mixture)

$Y_{H2O,sat}(T_d)$ is mass fraction of saturated water vapor at the particle surface

$X_{H2O,sat}(T_d)$ is molar fraction of water vapor at the particle surface under saturation conditions

$Y_{H2O,c}(T_d)$ is mass fraction of water vapor in the cell center

m_c is molar mass of the **Continuous Phase** (gas mixture) in the cell center

$P_{H2O,sat}(T_d)$ is partial pressure of saturated water vapor at the particle surface

Pyrolysis (devolatization)

It is assumed that the mass loss rate due to coal devolatization is governed by the following model relationship [8]:

$$\dot{m}_{d,2} = -k_{pyr} M_{d,ini} Y_{VOL,d} \quad \text{(Coal.19)}$$

$$k_{pyr} = A_{pyr} T_{d,abs}^{n_{pyr}} \exp\left(-\frac{T_{pyr}}{T_{d,abs}}\right) \quad \text{(Coal.20)}$$

Here:

k_{pyr} is rate constant, [s^{-1}]

$M_{d,ini}$ is initial mass of coal particles, [kg]

The following values of the model parameters are recommended in ref. [8]:

A_{pyr} = 2.08·10^4 (it is specified by the **A_pyr** parameter)

n_{pyr} = 0 (it is specified by the **n_pyr** parameter)

T_{pyr} = 11029 (it is specified by the **T_pyr** parameter)

Char burning

It is assumed that the following gross-reaction proceeds at the char surface:

$$O_2 + (C) \rightarrow CO_2 \quad \text{(Coal.21)}$$

The mass loss rate due to the char burning is governed by the following model relationship [8]:

$$\dot{m}_{d,3} = \pi d^2 \cdot X_{O2,c} \cdot P \cdot \frac{k_{kin} k_{dif}}{k_{kin} + k_{dif}} \quad \text{(Coal.22)}$$

$$k_{kin} = A_{CO2} T_{d,abs}^{n_{CO2}} \exp\left(-\frac{T_{CO2}}{T_{d,abs}}\right) \quad \text{(Coal.23)}$$

$$k_{dif} = \frac{C_{dif}}{d}\left(\frac{T_d + T_c}{2}\right)^{0.75} \quad \text{(Coal.24)}$$

Here:

$X_{O2,c}$ is molar fraction of oxygen in the cell center

The following values of the model parameters are recommended in ref. [8]:

C_{dif} = 5.06·10^{-7} (it is specified by the **C_diff** parameter)

A_{CO2} = 1.2 (it is specified by the **A_CO2** parameter)

n_{CO2} = 0 (it is specified by the **n_CO2** parameter)

T_{CO2} = 6013.7 (it is specified by the T_CO2 parameter)

Formation of NOx and SOx

If chemical elements N and S are present in the fuel (coal particles), the rate of formation of gases NOx and SOx in the course of oxidation of these elements is determined by the following expressions:

$$\dot{m}_{NOx} = \dot{m}_{d,3} C_{NOx} Y_{N,d} \quad \text{(Coal.25)}$$

$$\dot{m}_{SOx} = \dot{m}_{d,3} C_{SOx} Y_{S,d} \quad \text{(Coal.26)}$$

Here:

$Y_{N,d}$ is local mass fraction of element N in the fuel

$Y_{S,d}$ is local mass fraction of element S in the fuel

Model constants C_{NOx} and C_{SOx} are set by parameters **C_NOx** and **C_SOx**. The model constants are found in solving benchmark problems.

10.5.8.7.5 Equations for gas

The basic equations for the carrier phase in the presence of particles see in **Processes in the presence of dispersed medium > Equations for continuous medium**. Here the equations related to the coal combustion model are given. The equation for the recovered mass fraction of H2O (one of the two **Products**) reads:

$$\frac{\partial(\rho Y^*_{H2O})}{\partial t} + \vec{\nabla}\cdot(\boldsymbol{V}\rho Y^*_{H2O}) + \vec{\nabla}\cdot\boldsymbol{J}_{H2O} = \pi d^2 \dot{m}_{d,1} \quad \text{(Coal.27)}$$

The equation for the recovered mass fraction of volatiles (**Fuel**) reads:

$$\frac{\partial(\rho Y^*_{VOL})}{\partial t} + \vec{\nabla}\cdot(\boldsymbol{V}\rho Y^*_{VOL}) + \vec{\nabla}\cdot\boldsymbol{J}_{VOL} = \pi d^2 \dot{m}_{d,2} \quad \text{(Coal.28)}$$

The equation for the true mass fraction of volatiles (**Fuel**) reads:

$$\frac{\partial(\rho Y_{VOL})}{\partial t} + \vec{\nabla}\cdot(\boldsymbol{V}\rho Y_{VOL}) + \vec{\nabla}\cdot\boldsymbol{J}_{VOL} = \pi d^2 \dot{m}_{d,2} - Q_{VOL} \quad \text{(Coal.29)}$$

Source term Q_{VOL} is determined by the selected gas-phase combustion model - see **Mass transfer > Combustion > Equations**. The equation for the recovered mass fraction of O2 (**Oxidizer**) reads:

$$\frac{\partial(\rho Y^*_{O2})}{\partial t} + \vec{\nabla}\cdot(\boldsymbol{V}\rho Y^*_{O2}) + \vec{\nabla}\cdot\boldsymbol{J}_{O2} = -\pi d^2 \frac{8}{3}\dot{m}_{d,3} \quad \text{(Coal.30)}$$

The equation for the recovered mass fraction of CO2 (the second from the two **Products**) reads:

$$\frac{\partial(\rho Y^*_{CO2})}{\partial t} + \vec{\nabla}\cdot(\boldsymbol{V}\rho Y^*_{CO2}) + \vec{\nabla}\cdot\boldsymbol{J}_{CO2} = \pi d^2 \frac{11}{3}\dot{m}_{d,3} \quad \text{(Coal.31)}$$

The true mass fractions of O2, H2O, CO2 are computed with use of the difference $Y^*_{VOL} - Y_{VOL}$ - see **Mass transfer > Combustion > Equations, Eqs. (14)-(18)**.

Extra **Substances** can be included in the mathematical model of the **Continuous Phase**. In particular, equations for *NOx* and *SOx* read:

$$\frac{\partial(\rho Y_{NOx})}{\partial t}+\nabla\cdot(\rho Y_{NOx}\boldsymbol{V})+\nabla\cdot J_{NOx}=Q_{NOx} \quad \text{(Coal.32)}$$

$$Q_{NOx}=Q_{NOx,fuel}+Q_{NOx,chem} \quad \text{(Coal.33)}$$

$$\frac{\partial(\rho Y_{SOx})}{\partial t}+\nabla\cdot(\rho Y_{SOx}\boldsymbol{V})+\nabla\cdot J_{SOx}=Q_{SOx} \quad \text{(Coal.34)}$$

$$Q_{SOx}=Q_{SOx,fuel}+Q_{SOx,chem} \quad \text{(Coal.35)}$$

Here

$Q_{i,fuel}$ is source term for **Substance** *i* (*NOx* or *SOx*) due to presence of chemical element *N* or *S* in the solid fuel [kg m^{-3} s^{-1}]

$Q_{i,chem}$ is source term for **Substance** *i* (*NOx* or *SOx*) due to different chemical reactions in the gas (**Continuous**) **Phase** [kg m^{-3} s^{-1}]

The diffusion flux of **Substance** *i* is determined by the following expression:

$$\boldsymbol{J_i}=-\left(\rho_c D_i+\frac{\mu_{t,c}}{Sc_{t,c}}\right)\nabla Y_i=-\left(\frac{\mu_c}{Sc_c}+\frac{\mu_{t,c}}{Sc_{t,c}}\right)\nabla Y_i \quad \text{(Coal.36)}$$

10.5.8.8 References

1. Crowe C., Sommerfeld M., and Tsuji Yu. (1998) “Multiphase Flows with Droplets and Particles", *CRC Press LLC*, 471 p.

2. James P.W., Hewitt G.F. and Whalley P.B., "Droplet Motion in Two Phase Flow", *Proc. of the ANS/ASM/NRC Inst. Topical Meeting on Nuclear Reactor Thermohydraulics, Report NUREG/CP-0014*, pp. 1484-1503.

3. Klyachko L..S. (1934) "Equation for motion of dust particles in dust collecting equipment", *Heating and ventillation, No 4*, pp. 11-17

4. Schmehl R., Rosskamp H., Willmann M., and Witting S. (1999) "CFD Analysis of Spray Propagation and Evaporation Including Wall Film Formation and Spray/Film Interaction", *Int. J. of Heat and Fluid Flow, Vol. 20*, pp. 520-529.

5. Schiller L. and A. Naumann (1933) “A Drag Coefficient Correlation” *VDI Zeits., Vol. 77*, pp. 318-320.

6. Colombet D., Legendre D., Risso F., Cockx A., Guiraud P. (2014) Dynamics and mass transfer of rising bubbles in a homogenous swarm at large gas volume fraction // Journal of Fluid Mechanics, Vol. 763. pp. 254-285. ISSN 0022-1120.

7. Wallis G. B. (1961) Some hydrodynamic aspects of two-phase flow and boiling // Int. Heat Transfer Conference, Boulder, Colorado USA, Vol. 2, pp. 319–325.

8. Rosa L. M., Maurina G. Z., Beal L. L., Baldasso C., Torres A. P., Sousa M. P. (2015) Influence of interfacial forces on the mixture prediction of an anaerobic sequencing batch reactor (ASBR) // Brazilin Journal of Chemical Engineering, Vol. 32, No. 02, pp. 531 - 542, dx.doi.org/10.1590/0104-6632.20150322s00003300.

9. Saffman, P. G. (1965) The lift on a small sphere in a slow shear flow // Journal of Fluid Mechanics, 22, pp. 385-400.

10. Abramzon, B. & Sirignano, W.A. (1989) "Droplet Vaporization Model for Spray Combustion Calculations," *Int. J. of Heat and Mass Transfer, 32(9)*, pp. 1605-1618.

11. Miller, R.S., Harstad, K., and Bellan, J. (1998) "Evaluation of Equilibrium and Non-equilibrium Evaporation Models for Many-droplet Gas-Liquid Flow Simulation," *Int. J. of Multiphase Flow, 24*, pp. 1025-1055.

12. Сорокин К. Э., Бывальцев П. М., Аксенов А. А., Жлуктов С. В., Савицкий Д. В., Бабулин А. А., Шевяков В. И. Численное моделирование обледенения в программном комплексе FlowVision // Компьютерные исследования и моделирование. – Том 12, № 1. – 2020.

10.5.9 Processes in clearance

The following selection is present in position **Use Gap model** in window **Preprocessor > Models > Model #i**:

- **(none)**
- **Standard gap model**

Selection **(none)** implies that the governing equations are solved in thin clearances (if any) similarly to the other regions of the computational domain.

The gap model modifies the mathematical models of the **Physical processes** in thin clearances in such a way that the viscous forces and diffusive fluxes are computed from analytical expressions. This technique allows user to avoid resolving thin clearances between solid surfaces by computational grid. A gap model works only in the *gap cells* (see below).

This section has individual numeration of equations and illustrations.

10.5.9.1 Notations

Notation	Physical quantity	Name in *FlowVision*	Dimension
E_r	radiation energy density	**Radiation density**	W m^{-2} = kg s^{-3}
$\boldsymbol{F}$	friction force exerted onto a unit volume of fluid (vector)		kg m^{-2} s^{-2}
$\boldsymbol{n}$	normal to wall[1)] (vector)		
$\boldsymbol{J}_q$	heat flux from wall (vector)	**HeatFlux**	W m^{-2} = kg s^{-3}
$\langle T \rangle$	average relative temperature[2)]	**Temperature**	K
$\langle V \rangle$	modulus of average velocity[2)] (scalar)	**Velocity**	m s^{-1}
y	distance from the middle of clearance to the given point [3)]		m
δ	width of clearance		m
λ	molecular thermal conductivity	**Thermal conductivity**	kg m s^{-3} K^{-1}
λ/δ	coefficient of heat transfer in clearance	**Gap heat-transfer coef.**	kg s^{-3} K^{-1}
μ	molecular dynamic viscosity	**Viscosity**	Pa s = kg m^{-1} s^{-1}
ε_w	wall emissivity		
$\sigma_{rad} = 5.67 \cdot 10^{-8}$	Stefan-Boltzmann constant	**Stefan–Boltzmann constant**	W m^{-2} K^{-4}

[1)] The normal is directed inside the computational domain (inside a clearance in the given case).
[2)] Averaging is performed across a clearance.
[3)] The distance is measured along the normal to one of the clearance walls.

Indices:
1, 2 - values at the walls bounding a clearance,
eff - effective quantity specified at the clearance walls,
gap - value inside the clearance

The other notations see in section **Basic notations**.

10.5.9.2 Parameters

Parameters in window **Preprocessor > Models > Model #i > Standard gap model**:

Parameter	Permissible values	Description
Min. clearance	Determined by the problem.	Minimum permissible distance between the *gap-forming surfaces*: if the actual distance in certain cells is less

Parameter	Permissible values	Description
	(Default value: **0.000001**)	than the specified value, the value is used in calculations.
Max. clearance	Determined by the problem. (Default value: **1**)	Quantitative criterion used for definition of the *gap cells*: **Standard gap model** works in the cells where the distance between the *gap-forming surfaces* is less than the specified value.

Parameters in window **Preprocessor > Subregions > Subregion #i > Boundary conditions > Wall** or **Connected** (template):

Parameter	Permissible values	Description
Effective viscosity in gap	Determined by the problem. (Default value: **-1**)	Possibility to specify in a clearance coefficient of viscosity which differs from the molecular one.
Gap heat-transfer coef.	Determined by the problem. (Default value: **-1**)	Possibility to specify in a clearance coefficient of thermal conductivity which differs from the molecular one.

10.5.9.3 Gap cells

A computational cell is considered as a *gap cell* if
1) it is bounded by
   a) either two *gap-forming surfaces*,
   b) or one *gap-forming surface* with another bounding a neighbor cell;
2) the distance between the *gap-forming surfaces* is less than the value pointed out in the **Max. gap** parameter.

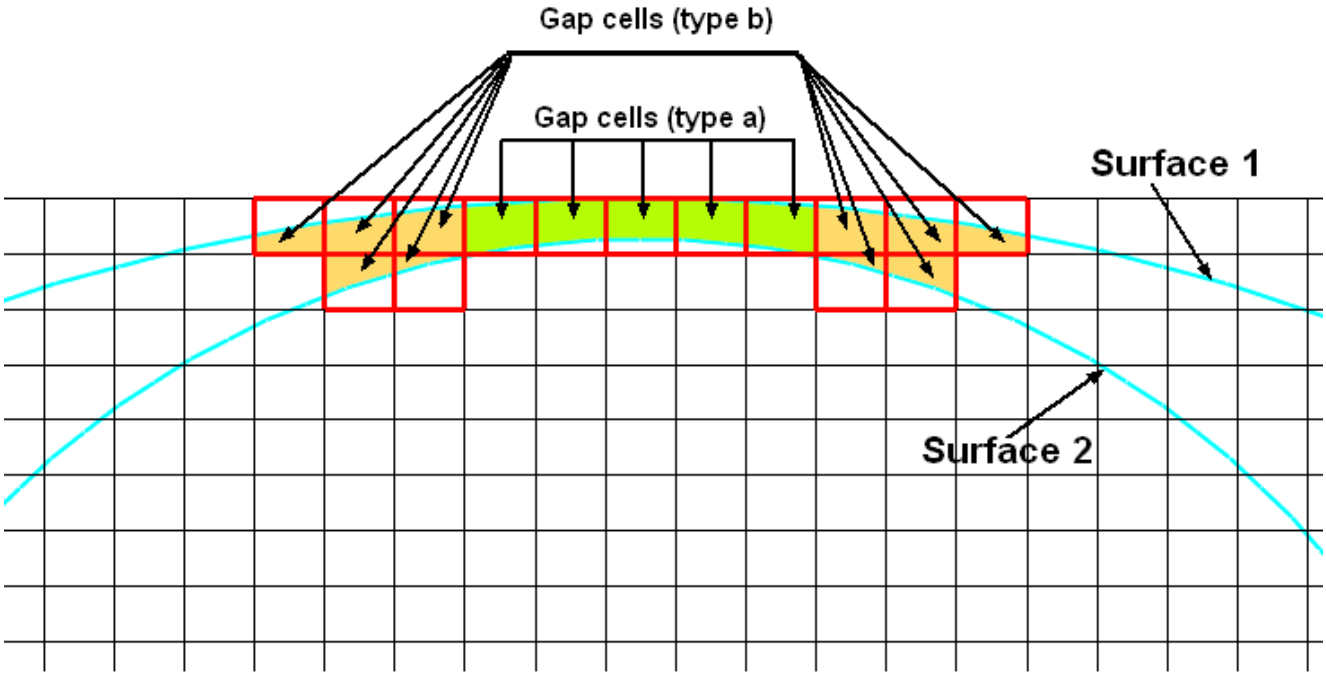


Figure 1. Two types of gap cells

Two surface are considered as *gap-forming* ones if
1) two different **Boundary conditions** are set on them, and the **Boundary conditions** are of types:
   - **Wall**
   - *and/or* **Connected** with **Conjugate temperature** type of binding
2) the angle between their normals lies in the range $120 \le \alpha \le 180$
3) the distance between the **Boundary conditions** is less then the maximal gap value, which is specified in the settings of the gap model (in the **Max. gap** parameter).

4) for **Velocity** on the **Boundary conditions** the **No slip** or **Logarithmic law** condition is specified.

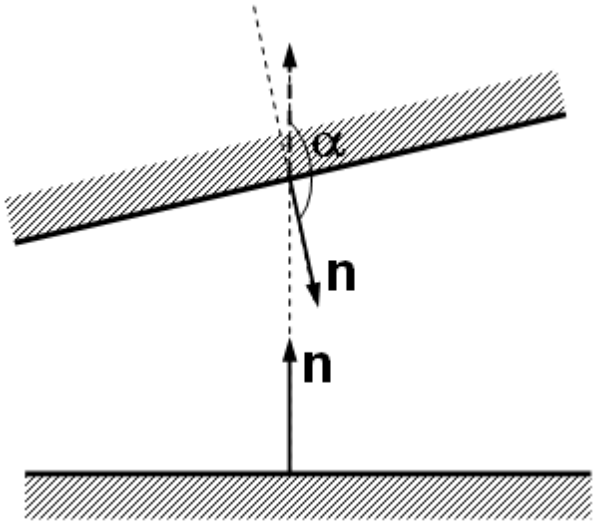


Figure 2. Angle between normals to gap-forming surfaces

FlowVision determines the *gap cells* automatically.

10.5.9.4 Equations

Process 'Motion'

The gap model assumes that a 2D Poiseuillle flow occurs in a clearance (see [1]). This is a laminar flow (the Reynolds number based on the clearance thickness is less than 1000). The velocity profile is parabolic. Consider a *gap cell* of volume Ω. In fact, it represents a thin channel of width δ with area of boundaries $2S$. Let a parabolic velocity profile develops in the given clearance, the average value of velocity being $\langle V \rangle$. Then the total friction force in the cell is

$$F = -\frac{12\mu\langle V\rangle S}{\delta\cdot\Omega} \quad [\mathrm{N\ m^{-3}}] \qquad \text{(Gap.1)}$$

Since $\Omega = S\cdot\delta$, we have

$$F = -\frac{12\mu\langle V\rangle}{\delta^2} \qquad \text{(Gap.2)}$$

Write the friction force in vector form in terms of the Darcy law:

$$\boldsymbol{F} = -\rho D\boldsymbol{V} \qquad \text{(Gap.3)}$$

Here

$$D = \frac{12\mu}{\rho\delta^2} > 0 \qquad \text{(Gap.4)}$$

Volume force (Gap.3) is added to the momentum equation in the *gap cell*. In doing so, the viscous forces exerted by the *gap-forming surfaces* are canceled.

The default values of the effective coefficient of viscosity set on the gap walls (specified in position **Preprocessor > Subregions > Subregion #i > Boundary conditions > Wall** or **Connected** (template) **> Effective viscosity in gap**) are:

$$\mu_{eff,1} = -1$$
$$\mu_{eff,2} = -1 \qquad \text{(Gap.5)}$$

Inside the clearance, the coefficient of viscosity is computed as follows:

$$\mu_{gap} = \frac{1}{2}\left(\mu_{eff,1} + \mu_{eff,2}\right) \quad \text{if} \quad \mu_{eff,1} \geq 0 \text{ and } \mu_{eff,2} \geq 0 \qquad \text{(Gap.6)}$$

Otherwise, the molecular viscosity is used in Eqs. (Gap.1)-(Gap.4), (Gap.11), (Gap.12).

Process 'Heat transfer'

It is assumed that in general case a parabolic profile of temperature develops across the clearance. If a source of energy is absent in the clearance, the steady-state solution provides a linear profile of temperature.

Presence of an energy source yields a non-linear profile. A non-linear profile of temperature may develop in a transient solution without any energy source.

The expression for the heat flux at wall "1" depends on the combination of the boundary conditions specified at the *gap-forming surfaces*. The following situations are considered:

1. Temperature is specified at both walls.
2. Temperature is specified at wall "1", heat flux is specified at wall "2".
3. Conjugate heat transfer is specified at both walls.
4. Conjugate heat transfer is specified at wall "1", temperature is specified at wall "2".
5. Temperature is specified at wall "1", conjugate heat transfer is specified at wall "2".
6. Conjugate heat transfer is specified at wall "1", heat flux is specified at wall "2".

In the first case,

$$\boldsymbol{J}_{q,1} = -\frac{2\lambda}{\delta}\left(3\langle T\rangle - 2T_1 - T_2\right)\boldsymbol{n} \qquad \text{(Gap.7)}$$

In the second case,

$$\boldsymbol{J}_{q,1} = -\frac{1}{2}\boldsymbol{J}_{q,2} + 3\lambda\frac{T_1 - \langle T\rangle}{\delta}\boldsymbol{n} \qquad \text{(Gap.8)}$$

The other cases are reduced to these two.

The default values of the effective heat transfer coefficient set on the gap walls (specified in position **Preprocessor > Subregions > Subregion #i > Boundary conditions > Wall** or **Connected** (template) **> Gap heat-transfer coef.**) are:

$$\begin{aligned}\left(\lambda/\delta\right)_{eff,1} &= -1\\ \left(\lambda/\delta\right)_{eff,2} &= -1\end{aligned} \qquad \text{(Gap.9)}$$

Inside the clearance, the coefficient of thermal conductivity is computed from

$$\lambda_{gap} = \frac{\delta\cdot\left(\lambda/\delta\right)_{eff,1}\cdot\left(\lambda/\delta\right)_{eff,2}}{\left(\lambda/\delta\right)_{eff,1} + \left(\lambda/\delta\right)_{eff,2} + 10^{-100}} \quad \text{if} \quad \left(\lambda/\delta\right)_{eff,1} \geq 0 \text{ and } \left(\lambda/\delta\right)_{eff,2} \geq 0 \qquad \text{(Gap.10)}$$

Otherwise, the molecular thermal conductivity is used in Eqs. (Gap.7), (Gap.8).

The heat generation due to viscous dissipation is taken into account in the energy equation by source terms (HeatTransfer.6), (HeatTransfer.11), which are modified in the clearance as follows:

$$G_{vis,L} = \frac{\mu}{\delta}\int_{-\delta/2}^{\delta/2}(V')^2 dy = \frac{\mu}{\delta^2}\left[12\left(\langle V\rangle - \frac{1}{2}(V_1+V_2)\right)^2 + (V_2-V_1)^2\right] \qquad \text{(Gap.11)}$$

$$\begin{aligned}G_{vis,G} &= \frac{\mu}{\delta}\left(V_1V'(-\delta/2) - V_2V'(\delta\delta 2)\right) =\\ &= \frac{\mu}{\delta^2}\left[6(V_2+V_1)\left(\langle V\rangle - \frac{1}{2}(V_1+V_2)\right) - (V_2-V_1)^2\right]\end{aligned} \qquad \text{(Gap.12)}$$

Process 'Radiation'

It is assumed that a clearance is always thin. As a consequence,

$$E_{r,2} = E_{r,1} = E_{r,gap} \qquad \text{(Gap.13)}$$

The value of the density of radiation energy in the clearance is found from the condition of equal radiation fluxes at the walls (see **Equations for method P1**, (Rad-P1.3)):

$$E_{r,gap} = \frac{\frac{2\varepsilon_{w,1}}{2-\varepsilon_{w,1}}\sigma_{rad}T_1^4 + \frac{2\varepsilon_{w,2}}{2-\varepsilon_{w,2}}\sigma_{rad}T_2^4}{\frac{2\varepsilon_{w,1}}{2-\varepsilon_{w,1}} + \frac{2\varepsilon_{w,2}}{2-\varepsilon_{w,2}}} \qquad \text{(Gap.14)}$$

10.5.9.5 Boundary conditions

Two types (templates) of **Boundary conditions** are permissible on the *gap-forming surfaces*: **Wall** and **Connected** (**Connection type** = **Conjugate temperature**).

The parameters and settings in these templates are standard, excluding parameters

- **Effective viscosity in gap** (see Eqs. (Gap.5), (Gap.6)),
- **Gap heat-transfer coef.** (see Eqs. (Gap.9), (Gap.10)).

10.5.9.6 References

1. Колесниченко В. И., Шарифулин А. Н. Введение в механику несжимаемой жидкости : учеб. пособие // Пермь : Изд-во Перм. нац. исслед. политехн. ун-та, 2019. – 127 с.

10.5.10 Overlapping boundary layer grid

FlowVision can use a special computational scheme, which allows adequate resolution of the fluid boundary layer near a solid curvilinear boundary of the computational domain (a wall). Calculations are performed on two grids:

- locally adapted (main) grid
- boundary layer grid (BL grid)

(see also sections Boundary layer grid (BL grid) and Subfolder «Boundary layer grids»)

The approach to generation of a boundary layer grid (BL grid) and the principle of calculations on two grids are described below.

This section has individual numeration of equations and figures.

10.5.10.1 Notations

Notation	Physical quantity	Name in *FlowVision*	Dimension
h_j	height of layer j of of prismatic cells		m
$\boldsymbol{n}_{eff}$	normal of *effective site*[1)] (vector)		
$\boldsymbol{P}_{eff}$	radius-vector of *effective site* [1)] (vector)		m
S_{eff}	area of *effective site*		m^2
y_j	distance from *effective site* to coordinate plane j of the *boundary layer grid* in the given main grid cell	**[0]** **[1]** **[2]** ...	m

[1)] Normal is directed inside computational domain.

Indices:

i - index of an *effective site*
j - number of a layer of prismatic cells
k - number of a surface polygon in a boundary cell of the main grid

10.5.10.2 Parameters

See section Subfolder «Boundary layer grids».

10.5.10.3 Generation of overlapping boundary layer grid

Assume that a Cartesian locally adapted grid is built in the computational space. Let a solid curvilinear boundary of the computational domain (a wall) is represented by connected triangles. The faces of the main grid cells cut these triangles. As a consequence, each boundary cell contains a group of wall polygons (borders between groups of facets also take part in forming these polygons).

For all N wall polygons in cell i of the main grid one *effective site* is introduced. Area S_{eff}, normal $\boldsymbol{n}_{eff}$ and the central point radius-vector $\boldsymbol{P}_{eff}$ are determined for this site as follows (see Figure 1):

$$S_{eff} = \sum_{k=1}^{N} S_k \qquad \text{(OBL.1)}$$

$$\boldsymbol{n}_{eff} = \frac{\sum_{k=1}^{N} S_k \boldsymbol{n}_k}{\left| \sum_{k=1}^{N} S_k \boldsymbol{n}_k \right|} \qquad \text{(OBL.2)}$$

$$\boldsymbol{P}_{eff} = \frac{\sum_{k=1}^{N} S_k \boldsymbol{P}_k}{S_{eff}} \qquad \text{(OBL.3)}$$

An irregular (in general case) array of prismatic cells is built on each *effective site* (see Figure 2). The number of prismatic cells built on the *effective sites* belonging to different cells of the main grid is the same. The heights of different prismatic cells from layer j are equal.

Note that the implemented algorithm does not subtract the volume of the *boundary layer grid* from that of the main grid.

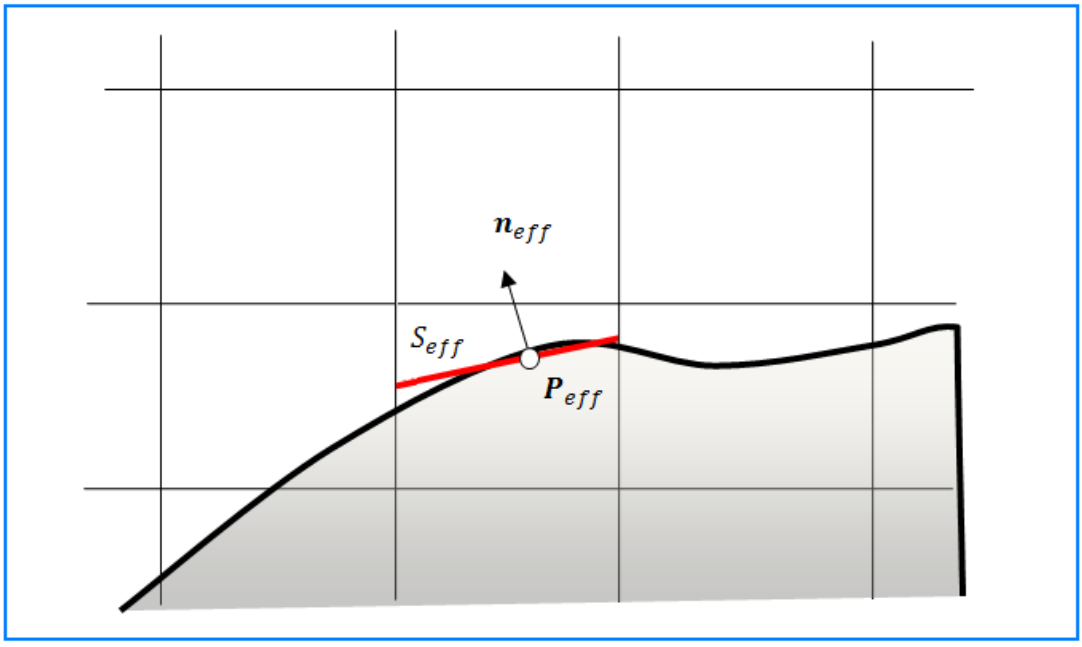


Figure.1. An *effective site* in a main grid cell (shown in red).

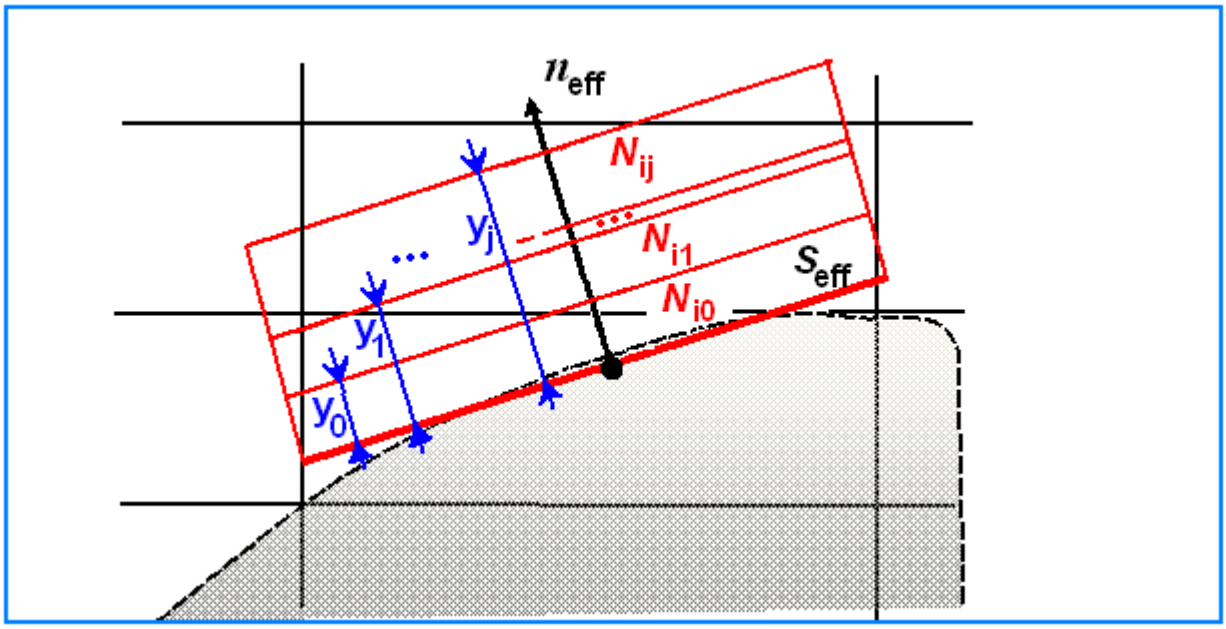


Figure.2. Fragment of the *boundary layer grid* corresponding to one *effective site*.

10.5.10.4 Calculations on two grids

The equations of continuity, momentum, energy, mass transfer, turbulence, etc. are integrated on both grids.

Values of desired variables are transferred from the main grid to the outer border of the *BL grid* (these values are interpolated from a center of the main grid's cell to which the outer border of the BL grid falls).

When the equations are solved on the *BL grid*, on the wall the boundary conditions, specified in the program's interface, are used. When transferring a momentum from cells of one prismatic stack to another (to appropriate cells of the neighboring stacks) the tangential component of the velocity changes its direction (rotates).

Solutions on the BL grid received on the wall (shear stress, etc.) are transferred into the main grid in which they are assumed as fixed values. If turbulence parameters in the boundary condition are specified as **Value in cell near wall**, then turbulence parameters from the BL grid are also transferred into the main grid's cell located near to the wall.

Hence, one can say that boundary conditions are conveyed from the wall to the centers of main grid cells adjacent to the wall.

See also section Boundary layer grid (BL grid).

11 References and bibliography

[1] Gupta A.K., Lilley D.G., and Syred N. (1984) "Swirl Flows," Abacus Press.

[2] Magel, H.C., Schnell, U. and Hein, K.R.G. (1996) "Simulation of detailed chemistry in a turbulent combustion flow," Twenty-Sixth Symposium (International) on Combustion / The Combustion Institute, pp. 67-74.

[3] Abe, K., Kondoh, T., Nagano, Y. (1995) “A New Turbulence Model for Predicting Fluid Flow and Heat Transfer in Separating and Reattaching flows-II. Thermal Field Calculation,” Int. Journal of Heat and Mass Transfer, Vol. 38, No. 8, pp. 1467-1481.

[4] Sommer, T. P., So, R. M. C. and Zhang, H. S (1993) “Near-wall variable Prandtl turbulence model for compressible flows,” AIAA Journal Vol. 31, pp. 27-35.

[5] Влияние распараллеливания вычислений с поверхностными межпроцессорными границами на масштабируемость параллельного итерационного алгоритма решения систем линейных уравнений на примере уравнений вычислительной гидродинамики / Харченко С.А. // Труды Международной научной конференции “Параллельные вычислительные технологии”, Санкт-Петербург, 28 января - 1 февраля 2008 г., Челябинск: Издво ЮУрГУ, 2008, с. 494-499.

[6] Abramzon, B. & Sirignano, W.A. (1989) "Droplet Vaporization Model for Spray Combustion Calculations," Int. J. of Heat and Mass Transfer, 32(9), pp. 1605-1618.

[7] Miller, R.S., Harstad, K., and Bellan, J. (1998) "Evaluation of Equilibrium and Non-equilibrium Evaporation Models for Many-droplet Gas-Liquid Flow Simulation," Int. J. of Multiphase Flow, 24, pp. 1025-1055

[8] Crowe, C., Sommerfeld, M., and Tsuji, Yu. (1998) “Multiphase Flows with Droplets and Particles // CRC Press LLC, 471 p.

[9] James, P.W., Hewitt, G.F. and Whalley, P.B., "Droplet Motion in Two Phase Flow," Proc. of the ANS/ASM/NRC Inst. Topical Meeting on Nuclear Reactor Thermohydraulics, Report NUREG/CP-0014, pp. 1484-1503.

[10] Клячко Л.С. (1934) Уравнение движения пылевых частиц в пылеприёмных устройствах, «Отопление и вентиляция», № 4, с. 11-17

[11] Schmehl, R., Rosskamp, H., Willmann, M., and Witting, S. (1999) "CFD Analysis of Spray Propagation and Evaporation Including Wall Film Formation and Spray/Film Interaction," Int. J. of Heat and Fluid Flow 20, pp. 520-529.

[12] Schiller, L. and A. Naumann (1933) “A Drag Coefficient Correlation” // VDI Zeits., Vol. 77, pp. 318-320.

[13] К.С. Басниев, А.М. Власов, В.М. Максимов. Подземная гидравлика. М.: «Недра», 1993 г.

[14] К.С. Басниев, Н.М. Дмитриев, Г.Д. Розенберг. Подземная гидромеханика. М.: 2005 г.

[15] Жлуктов С.В., Аксенов А.А., Харченко С.А., Москалев И.В., Сушко Г.Б., Шишаева А.С. Моделирование отрывных течений в программном комплексе FlowVision-HPC, Вычислительные методы и программирование, 2010, Т 11, №2, с. 246-249.

[16] Волков Э.П., Кудрявцев Н.Ю. (1989) "Моделирование образования окислов азота в турбулентном диффузионном факеле," Инженерно-физический журнал, Т.56, N 6, с. 885-894.

[17] Magnussen B.F., and Hjertager B.H. (1976) "On Mathematical Modelling of Turbulent Combustion with Special Emphasis on Soot Formation and Combustion," Sixteenth Symposium (International) on Combustion, pp. 719-729.

[18] Аксенов А.А., Похилко В.И., Тишин А.П. (1998) "Исследование двухступенчатого сжигания метана в вихревой горелке," Труды 2-ой Российской национальной конференции по теплообмену, Москва, 26-30 октября 1998 г., т.3, с.161-164.

[19] Wilcox, D. C. (1994) "Turbulence modeling for CFD," DCW Industries, Inc., 460 p.

[20] Jagadeesh, P. and Murali, K. (2005) “Application of Low-Re Turbulence Models for Flow simulations past Underwater Vehicle Hull Forms,” Journal of Naval Architecture and Marine Engineering, June 2005, pp. 41-54.

[21] Lien, F.S., Chen, W.L., and Leschziner, M.A. (1996) "Low-Reynolds-Number Eddy-Viscosity Modelling Based on Non-Linear Stress-Strain/Vorticity Relations," Proc. 3rd Symp. on Engineering Turbulence Modelling and Measurements, Crete, Greece, 1996.

[22] Menter, F.R., Kuntz, M., and Langtry, R. (2003) "Ten years of Industrial Experience with the SST Turbulence Model," Turbulence, Heat and Mass Transfer 4, Begell House, Inc.

[23] Bondar, T. and Prihoda J. (2006) "Numerical Simulation of Turbulent Free-Surface Flow in Curved Channel," Applied Scientific Research (Springer), Vol. 76, No. 4, June 2006, pp. 429-442.

[24] Kader, B.A. (1981) "Temperature and concentration profiles in fully turbulent boundary layers," International Journal of Heat and Mass Transfer, Vol. 24, No. 9, pp.1541-1544.

[25] Жлуктов С.В., Аксенов А.А., Харченко С.А., Москалёв И.В., Сушко Г.Б., Шишаева А.С. (2010) "Моделирование отрывных течений в программном комплексе FlowVision-HPC" Вычислительные методы и программирование, Т. 11, № 2, с. 76-87. ***

[26] Tsan-Hsing Shih, Povinelli, Louis A., Nan-Suey Liu, Potapczuk, Mark G., Lumley, J.L. (1999) "A Generalized Wall Function," NASA/TM - 1999 - 209398, 16 p.

[27] Шлихтинг, Г. (1974) "Теория пограничного слоя," Москва, Наука.

[28] Souders, D. T. and Hirt C.W. (2002) "Modeling Roughness Effects in Open Channel Flows," Flow Science Technical Note (FSI-02-TN60), 10 p.

[29] Четверушкин Б.Н., Математическое моделирование задач динамики излучающего газа, Москва, Наука: Главная редакция физико-математической литературы, 1985, 304 с.

[30] Моделирование отрывных течений в программном комплексе FlowVision-HPC [Текст] / Жлуктов С.В., Аксёнов А.А., Харченко С.А., Москалёв И.В., Сушко Г.Б., Шишаева А.С. // Вычислительные методы и программирование, 2010, Том 11, С. 234-245.

[31] Оптимизация положения элементов механизированного крыла с использованием программных комплексов FlowVision-HPC и IOSO [Текст] / Шишаева А.С., Москалёв И.В., Жлуктов С.В., Аксёнов А.А., Бабий Ю.И., Егоров И.Н, Судаков В.Г. // САПР и графика, сентябрь 2010.

[32] Harten, A; Engquist, Bjorn; Osher, Stanley; Chakravarthy, Sukumar R. (1997), "Uniformly High Order Accurate Essentially Non-oscillatory Schemes, III", J. Comput. Phys. 137: 3-47.

[32] Electrohydrodynamic Instability of Fluid Jet in Microgravity / A.A. Aksenov, A.V. Gudzovsky, A.A. Serebrov [Text] // Proc. of 5th Int. Symposium on Computational Fluid Dynamics (ISCFD), Aug. 31 - Sept. 3, 1993, Sendai, Japan, Japan Society of Computational Fluid Dynamics, 1993, Vol.1, P. 19-24.

12 ServeceInformationPage

2022

<%STYLESHEET%>

Index

T

U

V

W