



3DEXPERIENCE®

XFlow 2022 USER GUIDE

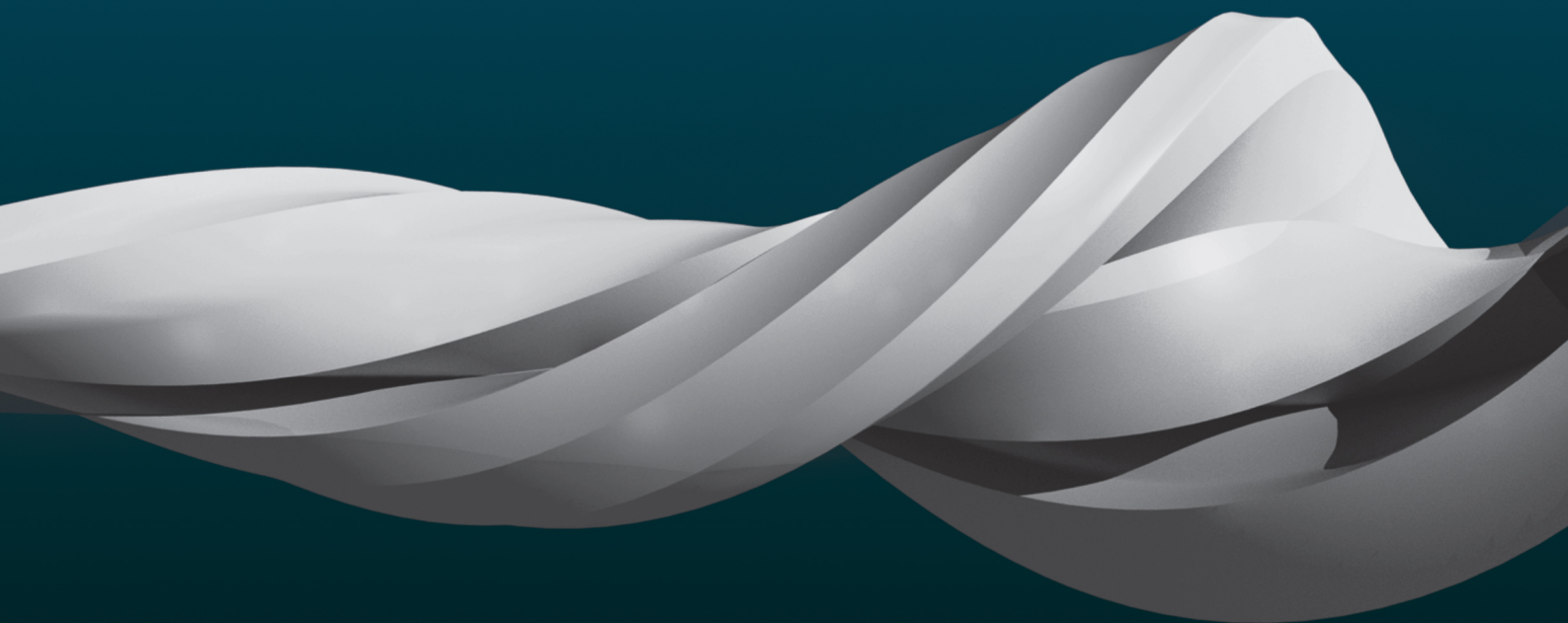


Table of Contents

1	Legal Notices	9
2	Introduction	11
2.1	Using this guide	11
2.2	Conventions	12
2.2.1	Style	12
2.2.2	Units	13
2.2.3	Coordinate system	13
2.2.4	Constants	14
3	Getting Started	15
3.1	Graphical User Interface	15
3.1.1	Windows management	16
3.1.1.1	Position and size	16
3.1.1.2	Show/Hide	16
3.1.1.3	Shortcuts	17
3.1.2	Project Manager	18
3.1.3	Main menu	19
3.1.4	Toolbar	21
3.1.5	Project Tree	24
3.1.5.1	Editing	25
3.1.5.2	Functions	26
3.1.5.2.1	User-defined variables	30
3.1.5.2.2	Tabular data	31
3.1.5.2.3	Unstructured Mesh	33
3.1.5.3	Trees management	34
3.1.5.4	Search in Project Tree	34
3.1.6	Message View	35
3.1.6.1	Simulation progress	36
3.1.6.2	Warnings	39
3.1.6.3	Errors	40
3.1.7	Graphic View	41
3.1.7.1	Legend	44
3.1.7.2	Gizmos	45
3.1.8	Function Viewer	46
3.1.9	Transform Tool	48
3.1.10	Time controls	48
3.2	Executing XFlow	49
3.3	Project modes	50
3.4	XFlow files	53
3.5	Preferences	56
3.5.1	Engine	57
3.5.2	Geometry	58
3.5.3	Graphic View	59

3.5.3.1 Environment	60
3.5.3.2 Lights	61
3.5.3.3 Performance-Quality	62
3.5.4 Project Tree	65
3.5.5 Application	66

4 Geometry 69

4.1 Create entity	69
4.2 Import and export geometry	73
4.3 Select geometry	76
4.3.1 Selection in Graphic View	76
4.3.2 Selection in Project Tree	77
4.4 Visualisation	78
4.4.1 Visualisation material	79
4.4.2 Visualisation mode	81
4.4.3 Back-face culling	82
4.4.3.1 Surface normals	82
4.4.3.1.1 Reorientate normals	83
4.4.3.1.2 Reverse orientation	83
4.5 Geometry Information	83
4.5.1 Normals	84
4.5.2 Local axes	85
4.5.3 Geometrical properties	85
4.5.4 Parametric hierarchy	86
4.5.5 Dimensions	87
4.5.6 Check holes	88
4.6 Modify geometry	89
4.6.1 Translation	89
4.6.2 Rotation	91
4.6.3 Scale	93
4.6.4 Symmetry	94
4.6.5 Duplicate	95
4.6.6 Replace	96
4.6.7 Split	97
4.6.8 Merge	97
4.6.9 Repair geometry	98
4.6.9.1 Hole filling tool	98
4.7 Delete geometry	102

5 Simulation Setup 103

5.1 Engine	103
5.1.1 Flow models	105
5.1.2 Multiphase models	107
5.1.3 Analysis types	109
5.1.4 Thermal and radiation models	110
5.1.5 Turbulence models	112
5.1.6 Turbulence generation	114
5.1.7 Acoustics analysis	114
5.1.8 Scalar transport	118
5.1.9 Advanced options	119

5.2	Environment	121
5.2.1	Domain type	125
5.2.2	Gravitational potential	127
5.2.3	External acceleration laws	127
5.2.4	Initial conditions	128
5.2.5	Volumetric Heat Source	130
5.2.6	Reference length	130
5.2.7	Reference area	131
5.2.8	Reference velocity	132
5.2.9	Water channel	133
5.2.9.1	Waves	134
5.2.10	Liquid regions	136
5.3	Materials	138
5.3.1	Name	139
5.3.2	Type	139
5.3.3	Molecular weight	139
5.3.4	Speed of sound	139
5.3.5	Reference density	139
5.3.6	Operating temperature	139
5.3.7	State equation	140
5.3.8	Viscosity models	140
5.3.8.1	Newtonian fluid	141
5.3.8.2	Non-Newtonian fluid	142
5.3.9	HTC Estimation	144
5.3.10	Thermal conductivity	145
5.3.11	Specific heat capacity	145
5.3.12	Adiabatic index	146
5.3.13	Reference pressure	146
5.3.14	Interactions	146
5.4	Geometry	147
5.4.1	Entities	148
5.4.1.1	Behaviour	150
5.4.1.2	Boundary conditions	152
5.4.1.2.1	Wall boundary condition	153
5.4.1.2.1.1	5.4.1.2.1.1	
5.4.1.2.1.2	5.4.1.2.1.2	
5.4.1.2.1.3	5.4.1.2.1.3	
5.4.1.2.1.4	5.4.1.2.1.4	
5.4.1.2.1.5	5.4.1.2.1.5	
5.4.1.2.1.6	5.4.1.2.1.6	
5.4.1.2.1.7	5.4.1.2.1.7	
5.4.1.2.2	Inlet boundary conditions	156
5.4.1.2.3	Outlet boundary conditions	157
5.4.1.2.4	Other boundary conditions	158
5.4.1.2.5	LODI	159
5.4.1.3	Surfaces	161
5.4.1.4	Children	162
5.4.1.5	Thermal boundary conditions	162
5.4.1.6	Conjugate heat transfer	163
5.4.1.7	Structural coupling	167
5.4.2	Arbitrary reference frame	168
5.4.3	Cables	170
5.4.4	Joints	173
5.5	Simulation	174
5.5.1	Time	175
5.5.2	Lattice discretization	183

5.5.3 Store data	190
6 Computation	199
6.1 Run computation	199
6.1.1 Generate launch scripts	200
6.1.2 Generate co-sim configuration file	200
6.1.3 Advanced computation	201
6.1.3.1 Serial computation	201
6.1.3.2 Distributed computation	204
6.2 Computation progress	208
6.3 Stop computation	209
6.4 Process manager	210
6.5 File-based computation control	211
7 Post-Processing	213
7.1 Load/unload simulation data	213
7.2 Post-Processing tree	213
7.2.1 General	216
7.2.1.1 Data	216
7.2.1.2 Interpolation	217
7.2.1.3 Show	218
7.2.2 Cutting planes	222
7.2.2.1 3d field	224
7.2.2.2 Vectors	225
7.2.2.3 Surface Field distribution	225
7.2.2.4 Domain structure	227
7.2.2.5 Markers	229
7.2.3 Isosurfaces	230
7.2.4 Stream tracers	231
7.2.4.1 Passive	233
7.2.4.2 DPM	236
7.2.4.2.1 Modeling discrete phase	239
7.2.5 Plot lines	239
7.2.6 Sensors	240
7.2.7 Surface integrals	242
7.2.8 Volume integrals	245
7.2.9 Custom fields	247
7.2.10 Entities	247
7.2.11 Cameras	248
7.2.12 Views	249
7.3 Visualisation fields	250
7.4 Import/Export post-processing setup	252
7.5 Animation	252
7.5.1 Basic animation	252
7.5.2 Advanced animation	254
7.5.3 Keyframes animation	257
7.6 Function Viewer	259
7.7 Export data	264

8	Co-Simulation	271
8.1	Import ADAMS simulation	271
8.2	FMI Standard	272
8.3	CSS Co-simulation	276
8.3.1	Abaqus	278
8.3.2	Simpack	284
8.4	Nastran	292
8.4.1	2-way OpenFSI	292
8.4.2	1-way Thermal	300
9	Application modes	305
9.1	Expert mode	305
9.2	Labs mode	306
9.2.1	Execution mode	307
9.2.2	Supersonic flow	308
9.2.3	Coupled energy flow	309
9.2.4	Spalart-Allmaras	309
9.2.5	Time integration scheme	310
9.2.6	MLS interpolation	310
9.2.7	Reference pressure point	311
9.2.8	Output format	311
9.2.9	Highest available frequency	312
9.2.10	Animated geometry behaviour	312
10	Command lines	315
10.1	Advanced command lines	315
10.2	Domain partition optimization	323
	Index	325

1 Legal Notices

SIMULIA XFlow is © 2011 - 2021 Dassault Systèmes España, SLU.

Trademarks

XFlow, 3DEXPERIENCE, the Compass logo and the 3DS logo, CATIA, SOLIDWORKS, ENOVIA, DELMIA, SIMULIA, GEOVIA, EXALEAD, 3D VIA, BIOVIA, NETVIBES, and 3DEXCITE are commercial trademarks or registered trademarks of Dassault Systèmes, a French “société européenne” (Versailles Commercial Register # B 322 306 440), or its subsidiaries in the U. S. and/or other countries. All other trademarks are owned by their respective owners. Use of any Dassault Systèmes or its subsidiaries trademarks is subject to their express written approval.

DS Offerings and services names may be trademarks or service marks of Dassault Systèmes or its subsidiaries.

Legal Notices

XFlow and this documentation may be used or reproduced only in accordance with the terms of the software license agreement signed by the customer, or, absent such an agreement, the then current software license agreement to which the documentation relates.

This documentation and the software described in this documentation are subject to change without prior notice.

Dassault Systèmes and its subsidiaries shall not be responsible for the consequences of any errors or omissions that may appear in this documentation.

SIMULIA XFlow is © 2011 - 2021 Dassault Systèmes España, SLU.

For additional information concerning trademarks, copyrights, and licenses, see the Legal Notices in the XFlow Installation Guide.

2 Introduction

XFlow™ is a powerful Computational Fluid Dynamics (CFD) software designed for engineering analysis. It uses a proprietary, particle-based, fully Lagrangian approach which can easily handle traditionally complex problems such as aerodynamics, aero-acoustics, moving parts, free surface flows and fluid-structure interaction.

Key features and technologies

Meshless approach to CFD: The meshless approach within XFlow is particle-based and fully Lagrangian, which means that classic fluid domain meshing is not required. Also surface complexity is not a limiting factor. XFlow can handle moving bodies and deformable parts, and is tolerant to the quality of the input geometry.

Particle-based kinetic solver: XFlow features a novel particle-based kinetic algorithm that resolves the Boltzmann and the compressible Navier-Stokes equations. The solver features state-of-the-art Large Eddy Simulation (LES) modelling, and advanced non-equilibrium wall models.

Single consistent wall model: XFlow uses a unified non-equilibrium wall function in order to model the boundary layer. This wall model works in all cases, meaning it is not necessary to select between different algorithms and deal with the different limitations of each scheme.

Adaptive wake refinement: The XFlow engine automatically adapts the resolved scales to the user's requirements, refining the quality of the solution near the walls and dynamically adapting to the wake while the flow develops.

Advanced modelling capabilities: XFlow is capable of handling large and complex models, and greatly simplifies the setup of analysis with moving parts, hierarchical structures, enforced or rigid body motion, and contact modelling.

Advanced analysis capabilities: The XFlow solver also features thermal analysis, conjugated heat transfer, transonic and supersonic flows, flow through porous media, non-Newtonian flows, and complex boundary conditions including the porous jump and fan models.

Near-linear scalable performance: XFlow is fast and efficient, even on a standard desktop PC. It is fully parallelized for multi-core technology with near-linear scalability.

Easy-to-use interface: XFlow provides a unique and novel interface and working environment for the user, including pre-processor, solver and post-processor fully integrated in the same environment, state-of-the-art visualization, and configurable layout.

2.1 Using this guide

This User Guide describes all the features and options available in XFlow. It is structured following the typical workflow: [Geometry](#), [Simulation Setup](#), [Computation](#) and [Post-Processing](#).

This guide should be used in conjunction with the **Tutorial Guide** (recommended for new users) and the **Validation Guide**.

2.2 Conventions

In this guide the following conventions are used:

- [Style](#)
- [Units](#)
- [Coordinate system](#)
- [Constants](#)

2.2.1 Style

Typographical conventions are used to facilitate the reading:

- Menu options are indicated in **bold black**.
- Names of windows are in *italics*.
- Items and options in the *Project Tree* are indicated in **Verdana font**.
- Child items in the *Project Tree* are indicated with an arrow bullet, as:
↳ **Child item**
- [Links](#) are underlined in blue.
- Keys are indicated in **bold blue**.
- Cascading menus are represented as: **Menu1 > Menu2 > Menu3**
- Examples are enclosed in tables:

 **Example:**

- Additional explanations and recommendations are enclosed in a message box, as for example:

 **Tip:** Explains an easy way to do a task or just to improve the work flow.

 **Please note:** Contains a brief explanation on what must be taken into account for an specific task.

2.2.2 Units

Units should be given in accordance with the International System of Units (SI), as shown in the table below:

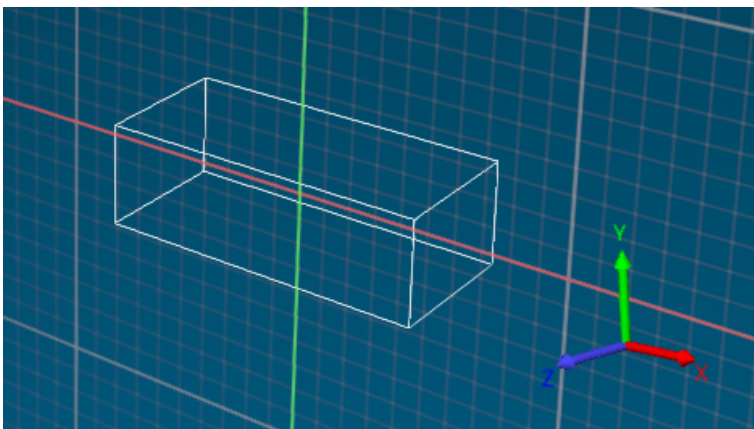
Variable	Symbol	Units
Mass		kg
Length		m
Time	t	s
Velocity	v_x, v_y, v_z	m s^{-1}
Pressure	p	Pa
Temperature	θ	K
Acceleration	a	m s^{-2}
Density	ρ	Kg m^{-3}
Viscosity	μ	Pa s
Thermal Conductivity	k	$\text{W m}^{-1} \text{K}^{-1}$
Specific Heat	C_p	$\text{J kg}^{-1} \text{K}^{-1}$
Angle		degrees
Angular velocity		rad s^{-1}
Force		N

⚠ Please note: Angles are given in degrees, while angular velocities are given in radians per second.

2.2.3 Coordinate system

By default, the X-axis represents the horizontal direction (or length), Y-axis stands for the vertical one (or height); while the Z-axis represents the third dimension in 3D flows (or depth).

Users may have to rotate the geometry when importing it from the CAD software to make it consistent with the above axis convention; this rotation can be made using the XFlow [Transform Tool](#).



Coordinate system shown in the Graphic View window.

2.2.4 Constants

Standard constants used by XFlow are listed in the table below:

Constant	Symbol	Value	Units
Stefan-Boltzmann	σ	5.6703×10^{-8}	$\text{W m}^{-2} \text{K}^{-4}$
Number pi	pi	3.14159	-

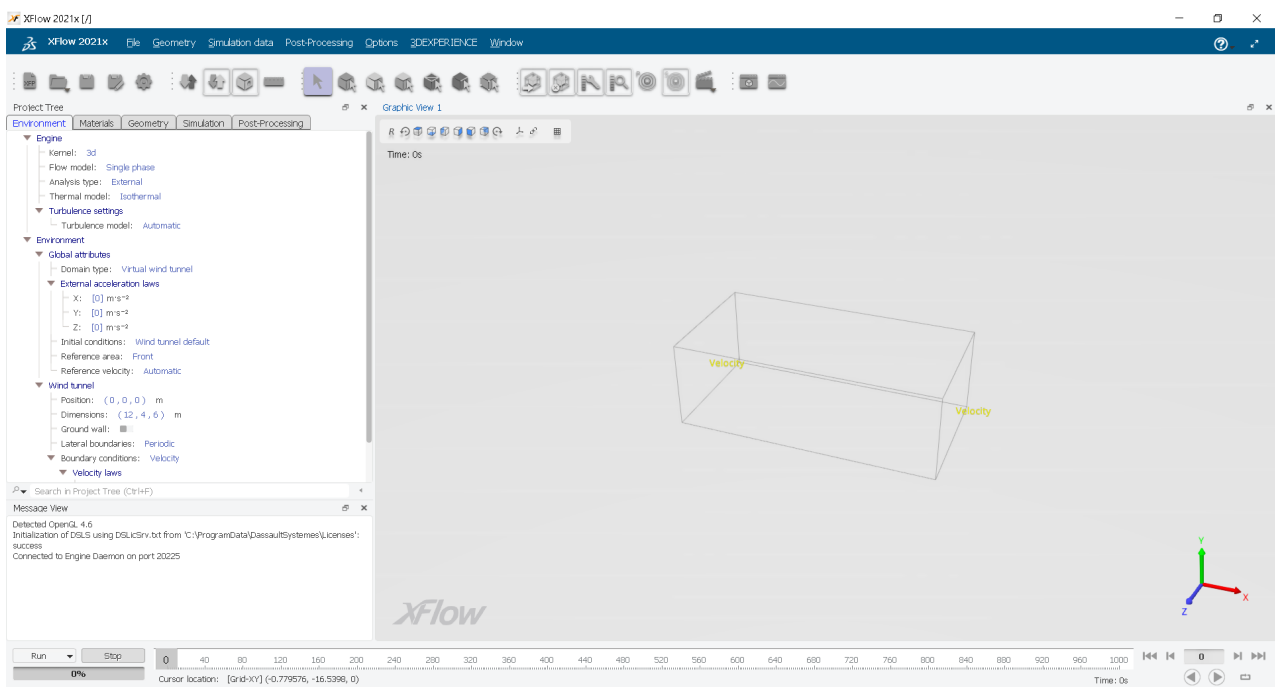
3 Getting Started

In this chapter a short Getting Started guide is presented, covering the following topics:

- How to open XFlow: See [Executing XFlow](#)
- How to use XFlow: See [Project modes](#).
- Introduction to the common XFlow file extensions: See [XFlow files](#)
- How to customize XFlow: see [Preferences](#)

3.1 Graphical User Interface

In this chapter, the Graphical User Interface (GUI) and its [management](#) are described.



XFlow graphical user interface appearance

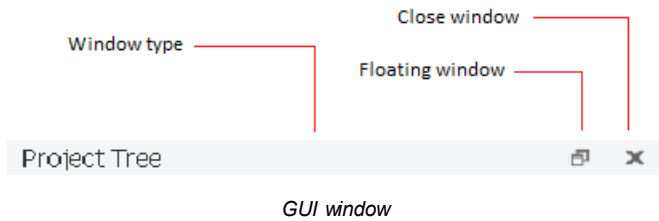
As it is shown in the figure above, the GUI is composed of the following elements:

1. [Main menu](#)
2. [Toolbar](#)
3. [Project tree](#)
4. [Message view](#)
5. [Graphic view](#)
6. [Function viewer](#)
7. [Transform tool](#)
8. [Time controls](#)

3.1 Graphical User Interface

3.1.1 Windows management

The Graphical User Interface has five types of windows ([Project Tree](#), [Message View](#), [Graphic View](#), [Function Viewer](#) and [Transform Tool](#)).



GUI windows can be [moved](#), [resized](#) and [shown or hidden](#); some of them can even be managed using [shortcuts](#).

3.1.1.1 Position and size

The windows can be resized by dragging their borders while clicking the **left mouse button**.

The windows can be moved to a different location just by a drag-and-drop, clicking (**left mouse button**) on the title bar. XFlow will highlight the positions where the window can fit, it can be either over an existing window, at a new location or detached.

When a window is moved on another existing window, XFlow will merge them and tabulations will appear at the bottom of the window to switch from one to another:

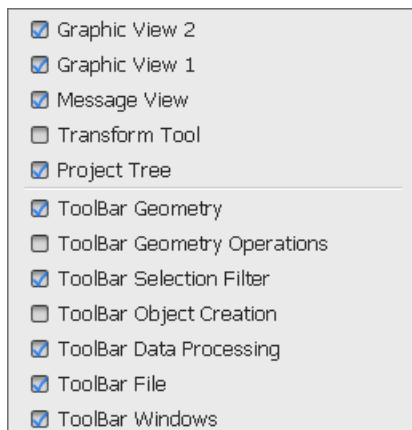


Once the user has customized the layout, it can be saved in a *.lay file:

Main menu > Views > Layout > Save layout

3.1.1.2 Show/Hide

A list of the different available windows will appear if you click the **right mouse button** on the title bar or the toolbar of the interface as shown above. This list allows the user to show/hide windows by ticking/unticking the corresponding checkbox ☒.



 **Please note:** The GUI layout allows only one instance of the *Project Tree*, *Message View* and *Transform tool* windows, while the *Graphic View* and *Function Viewer* windows can be present more than once. Consequently, closing the *Project Tree*, the *Message View* or the *Transform tool* is equivalent to hide them; they can be later recovered as explained above.

3.1.1.3 Shortcuts

Shortcuts by function

File

Ctrl + n	New project
Ctrl + l	Load project
Ctrl + s	Save project
Ctrl + i	Import geometry
Ctrl + e	Export (selected) geometry
Ctrl + k	Create/Delete keyframe (In keyframe mode)
Alt + k	Edit keyframe mode
Alt + r	Remove all keyframe (In keyframe mode)

Analysis information

Alt + U	Hide time and number of elements info from the Graphic View
F	Fit to screen. If a geometry is selected (either from the Graphic View or the Project Tree) it will fit to the geometry.

Mouse shortcuts

Alt + left mouse button drag	Rotate camera in <i>Graphic View</i> Pan in <i>Function Viewer</i>
Alt + right mouse button drag	Zoom in <i>Graphic View</i> Adjust the scale in <i>Function Viewer</i>
Alt + middle mouse button drag	Pan in <i>Graphic View</i> Zoom in <i>Function Viewer</i>
Right mouse button	Contextual menu in Graphic View Contextual menu in Function Viewer
Left mouse button	Applies the selection mode chosen in the Toolbar selection filter, in <i>Graphic View</i> Pan in <i>Function Viewer</i>
Wheel mouse	Zoom in <i>Graphic View</i> Zoom in <i>Function Viewer</i>
Middle mouse button	Scroll up in the <i>Project Tree</i> Create/Delete keyframe (In keyframe mode)

3.1 Graphical User Interface

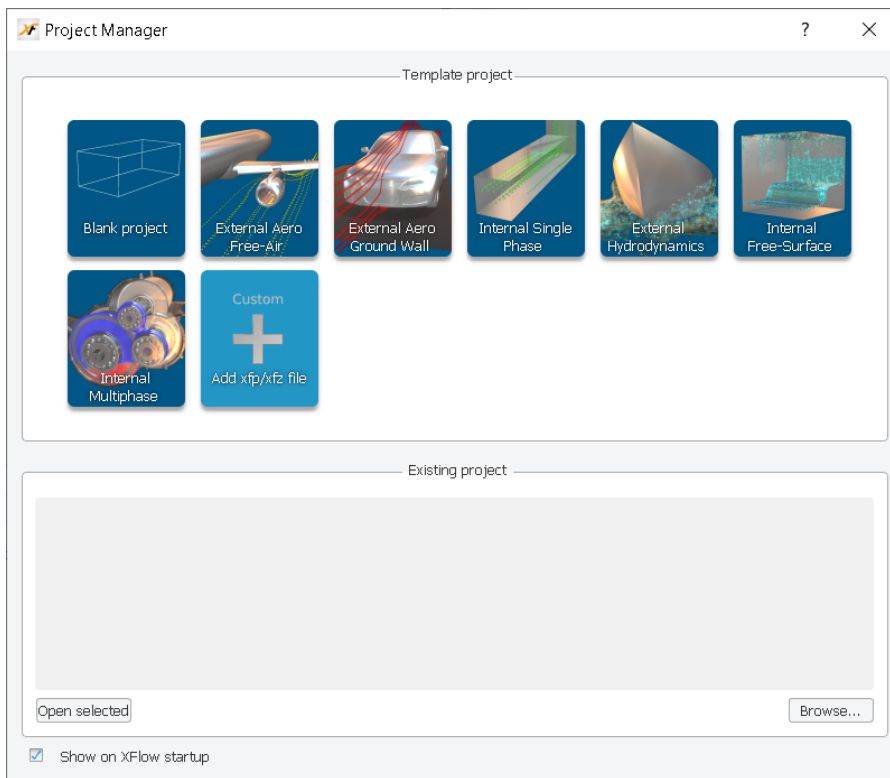
⚠ Please note: The above shortcuts corresponds to the standard keyboard configuration of Microsoft Windows. Linux users might need to press **Ctrl+Shift** instead of **Alt**. If this is the case, the user can change the "Movement key" to **Alt** in the "Window Preferences" of the Linux distribution.

⚠ Please note: New mouse options are available in XFlow:

- Middle mouse button: Rotate camera in *Graphic View* around the point clicked by the user.
- Middle mouse button + **Shift**: Zoom in the *Graphic View*.
- Middle mouse button + **Ctrl**: Pan in the *Graphic View*.

3.1.2 Project Manager

The Project Manager is a window that automatically opens when XFlow starts up in order to get started with a project.



Default view of the Project Manager

💡 Tip: You can hide this window at start-up by switching off the "Show on XFlow startup" button.

Template project

The template project allows to load either a blank project with all default options, some predefined set of project setup for some typical workflows or a custom project setup. The predefined templates are intended to guide the user in setting up a project faster, however inputs should be reviewed and modified in agreement to the intended simulation.

The templates proposed are:

- Blank project: project with all default properties.
- External Aero Free-Air: to simulate aerodynamics of airborne in free air conditions. This predefines a single phase analysis with external wind tunnel and no ground wall, and will require to import the geometry model.
- External Aero Ground Wall: to simulate aerodynamics of ground vehicle. This predefines a single phase analysis with external wind tunnel and ground wall, and will require to import the geometry model.
- Internal Single Phase: to simulate internal single phase flows. This predefines an internal single phase analysis and will require to import the internal geometry to define the internal simulation domain.
- External Hydrodynamics: to simulate ships hydrodynamics. This predefines a free surface analysis with water channel and will require to import the geometry model.
- Internal Free-Surface: to simulate internal free surface flows. This predefines an internal free surface analysis and will require to import the internal geometry to define the internal simulation domain.
- Internal Multiphase: to simulate internal multiphase flows. This predefines an internal multiphase particle-based tracking analysis and will require to import the internal geometry to define the internal simulation domain.
- Custom: to add an existing project file in either *xfp* or *xfz* format as a template. Loading a custom template will include all the geometries, boundary conditions, post-processing setup and layout defined in the project file.

Existing project

This shows a list of recently opened project and is a shortcut to quickly reload a project on which you worked recently.

3.1.3 Main menu

The main menu contains the following items:



File >

- > **New project:** To create a new project.
- > **Load project:** To open an existing project file.
- > **Recent projects:** To open a project that has recently been opened.
- > **Project manager:** To open the project manager
- > **Save project:** To save the current project.
- > **Save project as...:** To save the current project at the specified path.
- > **Exit:** To quit and close XFlow.

Geometry >

- > **Create object:** To create a cylinder, sphere, cone, torus, box, prism or NACA 4digits.
- > **Import a new geometry:** To import a geometry from a file.
- > **Export geometry:** To export a geometry of the current project to an external file.
- > **Selection filter:** To change the selection mode: view only, object, shell, face, mesh, edge and vertex.
- > **Show/hide holes:** To show/hide the holes (open edges) present in the geometry.
- > **Dimensions:** To show/hide the dimensions of the selected geometry.
- > **Symmetry:** To create a symmetric object from the one selected.
- > **Duplicate:** To duplicate the selected object.
- > **Split in connected shapes:** To split an object in its parts.
- > **Split in shapes according to orientation:** To split an object in its parts.

3.1 Graphical User Interface

- > **Create mesh shape:** To merge several objects into one.
- > **Reorientate normals:** To unify the orientation of the surface normals.
- > **Repair geometry:** To repair a geometry with holes.

Simulation Data >

- > **Generate launch scripts:** To generate the Windows (.bat) and Linux (.sh) [launch scripts](#) to launch the simulation.
- > **Generate co-sim configuration file:** To generate the co-simulation configuration file for [CSS co-simulation](#) setups.
- > **Load simulation data:** To load data from a previous run.
- > **Unload simulation data:** To unload data from a previous run.
- > **Export data:** To export the simulation results to other post-processing formats (Paraview, Enight Gold, CGNS, surface format, or XFlow surface data format).
- > **Export surface data:** To export results at the geometry surfaces.
- > **Export cutting plane data to raw format:** To export the data of the cutting planes during post-processing.
- > **Export data of cutting plane distribution:** To export the data of a cutting plane field distribution when created.
- > **Export selected isosurfaces:** To export data from isosurfaces.
- > **Export numerical data:** To export numerical data available in the [Function Viewer](#).
- > **Analysis settings:** To set a user-defined legend range.

Post-Processing >

- > **Create cutting plane:** To create a new cutting plane.
- > **Create isosurface:** To create a new isosurface.
- > **Create stream tracer:** To create a new stream tracer.
- > **Create custom field:** To create a user-defined field for the post-process.
- > **Numerical analysis:** To create tools to measure numerical data.
 - > **Create sensor:** To create a new sensor.
 - > **Create plot line:** To create a new plot line.
 - > **Create surface integral:** To create a new surface integral.
 - > **Create volume integral:** To create a new volume integral.
 - > **Set probe by mouse:** To create a new probe in a location set using the mouse selection.
 - > **Set sensor by mouse:** To create a new sensor in a location set using the mouse selection.
 - > **Import from file:** To import sensors and probes location from file.
 - > **Sensors:** To import sensors location from file.
 - > **Probes:** To import probes location.
 - > **Individually:** To import probes location from file individually.
 - > **Grouped:** To import probes location from file grouped.
- > **Create camera:** To create a new camera.
- > **Create clipping plane:** To create a clipping plane.
- > **Animation:** To create a sequence of images with the evolution of the results.
- > **Look up value:** To get the local value of the visualization field at the point given by the mouse position.
- > **Import post-processing setup:** To import a predefined post-processing layout and setup from an .xfpp file.
- > **Export post-processing setup:** To export the current post-processing layout and post-processing setup to an .xfpp file.

Options >

- > **Setup progressive wave boundary conditions:** To set a progressive wave in the inlet boundary condition and the corresponding consistent initial free surface field, in free surface external flows.
- > **Import ADAMS simulation:** To import a kinematic from MSC.ADAMS. Only available in [Labs mode](#).
- > **User-defined variables:** To create [user-defined variables](#) that can be used in the laws definitions.
- > **Preferences:** To set up the engine settings and the appearance variables of the GUI.

3DEXPERIENCE >

- > **Load 3DEXPERIENCE Model:** To load a model from the 3DEXPERIENCE platform.
- > **Save Model to 3DEXPERIENCE:** To save a model into the 3DEXPERIENCE platform.
- > **Create 3DEXPERIENCE Model Revision:** To create a new revision of a model from the 3DEXPERIENCE platform.
- > **Save results to 3DEXPERIENCE:** To upload simulation results on the 3DEXPERIENCE platform.
- > **Import product/part from 3DEXPERIENCE:** To load a geometry from the 3DEXPERIENCE platform.
- > **Reset the 3DEXPERIENCE Connection:** To reset 3DEXPERIENCE platform credentials and collaborative space.

 **Please note:** The options involving the 3DEXPERIENCE will only be available if the 3DEXPERIENCE and XFlow role (XXF) are installed.

Window >

- > **Graphic View >**
 - > **Fast view:** To allow a quick navigation through the *Graphic View* when moving the camera's point of view (orbit, pan or zoom) through a bounding box representation of the geometry objects.
 - > **Zoom window mode:** To zoom the area selected by the mouse.
- > **Layout >**
 - > **Save layout:** To save the current layout of the GUI into a .lay file.
 - > **Load layout:** To load a .lay file to recover a saved layout of the GUI.
 - > **Default layout:** To load the XFlow default GUI layout.
- > **New Graphic View:** To create a new floating Graphic View.
- > **New Function Viewer:** To create a new floating Function Viewer.



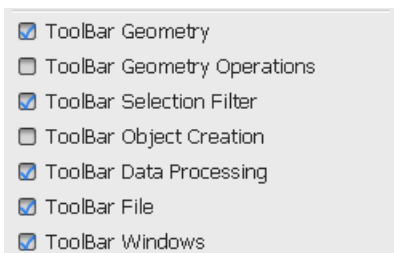
- > **Help:** To open the user guide.
- > **About:** Shows license information and the XFlow version and build number.

3.1.4 Toolbar

Toolbar icons provide shortcuts to the most useful utilities of XFlow, otherwise accessible from the [Main menu](#).

Several toolbars are available in XFlow, each of them can be managed independently by dragging it while clicking the **left mouse button**; as the toolbar is dragged over the GUI the positions where it could be placed are automatically highlighted.

The list of the toolbars will appear if you click the **right mouse button** on any title bar or toolbar of the interface. This list allows the user to show/hide groups of toolbar by ticking/unticking the corresponding checkbox .



List of available Toolbars in XFlow

These toolbars are classified as follows:

Toolbar File

	New project	Main menu > File > New project
	Load project	Main menu > File > Load project
	Load recent project (long press)	Main menu > File > Recent projects
	Save project	Main menu > File > Save project
	Save project As	Main menu > File > Save project As
	Preferences	Main menu > Options > Preferences

Toolbar Geometry

	Import a new geometry	Main menu > Geometry > Import a new geometry
	Export geometry	Main menu > Geometry > Export geometry
	Show/hide holes	Main menu > Geometry > Show/hide holes
	Dimensions	Main menu > Geometry > Dimensions

Toolbar Geometry Operations

	Symmetry	Main menu > Geometry > Symmetry
	Duplicate	Main menu > Geometry > Duplicate
	Split in connected shapes	Main menu > Geometry > Split in connected shapes
	Split in shapes according to orientation	Main menu > Geometry > Split in shapes according to orientation
	Create mesh shape	Main menu > Geometry > Create mesh shape
	Reorientate normals	Main menu > Geometry > Reorientate normals
	Healing	Main menu > Geometry > Healing

Toolbar Object Creation

 Create cylinder	Main menu > Geometry > Create object > Create cylinder
 Create sphere	Main menu > Geometry > Create object > Create sphere
 Create cone	Main menu > Geometry > Create object > Create cone
 Create torus	Main menu > Geometry > Create object > Create torus
 Create box	Main menu > Geometry > Create object > Create box
 Create prism	Main menu > Geometry > Create object > Create prism
 Create NACA (4digit)	Main menu > Geometry > Create object > Create NACA (4digit)
 Create vertex	Main menu > Geometry > Create object > Create vertex
 Create reference frame	Main menu > Geometry > Create object > Create reference frame
 Create line	Main menu > Geometry > Create object > Create line
 Create curve	Main menu > Geometry > Create object > Create curve
 Create surface	Main menu > Geometry > Create object > Create surface

Toolbar Selection Filter

 View only	Main menu > Geometry > Selection filter > View only
 Object filter	Main menu > Geometry > Selection filter > Object filter
 Shell filter	Main menu > Geometry > Selection filter > Shell filter
 Face filter	Main menu > Geometry > Selection filter > Face filter
 Wire filter	Main menu > Geometry > Selection filter > Wire filter
 Edge filter	Main menu > Geometry > Selection filter > Edge filter
 Vertex filter	Main menu > Geometry > Selection filter > Vertex filter

Toolbar Data Processing

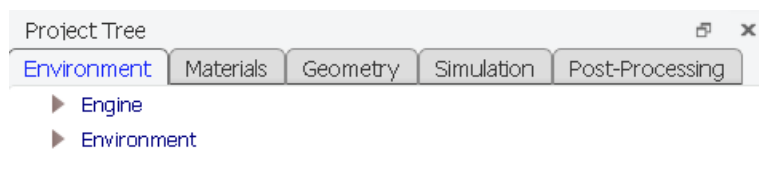
	Load simulation data	Main menu > Simulation data > Load simulation data
	Unload simulation data	Main menu > Simulation data > Unload simulation data
	Analysis settings	Main menu > Simulation data > Analysis settings
	Look up value	Main menu > Post-Processing > Look up value
	Set probe by mouse	Main menu > Post-Processing > Numerical analysis > Set probe by mouse
	Set sensor by mouse	Main menu > Post-Processing > Numerical analysis > Set sensor by mouse
	Animation	Main menu > Post-Processing > Animation
	Keyframe mode	Main menu > Post-Processing > Animation > Edit keyframe mode

Toolbar Windows

	New Graphic View	Main menu > Views > New Graphic View
	New Function Viewer	Main menu > Views > New Function Viewer

3.1.5 Project Tree

The *Project Tree* consists in a hierarchical structure of folders and nodes, where the user can easily set up the simulation parameters. It is divided into five categories that follow the logical sequence required to perform a CFD analysis.



Project Tree window

Project Tree categories:

- **Environment:** This category contains two main folders: (a) **Engine** - to select the XFlow engine (); and (b) **Environment** - to define the problem boundary conditions, initial solution, and external forces ([+Info](#)).
- **Materials:** In this category the thermophysical properties of the fluid have to be defined ([+Info](#)).
- **Geometry:** Geometry objects are shown in this data structure. Here, the objects behavior and boundary conditions are specified ([+Info](#)).
- **Simulation:** Control parameters of the numerical simulation are set up in this branch of the *Project Tree* ([+Info](#)).

- **Post-processing:** Once the simulation is run, the visualization and post-processing options are shown here ([+Info](#)).

Tip: To hide/show all the folders and nodes of the Project Tree structure, **right click** on the *Project Tree* window and a drop-down menu will appear with the following options: **Collapse all**; **Expand all**.

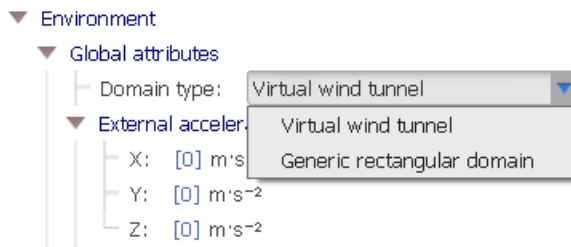
The *Project Tree* window can be customized in:

Main menu > Options > Preferences > Project Tree ([+Info](#)), or 

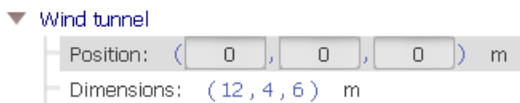
3.1.5.1 Editing

Editable fields in the *Project Tree* are of the following types:

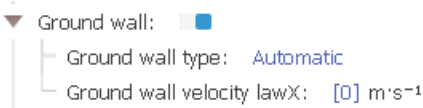
1. **Drop-down list:** Click on the option name or line to display all the options.



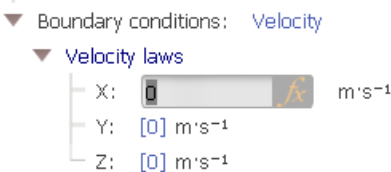
2. **Number:** Click on the number to change it.



3. **On/Off:** Click on the switch icon to switch it on  or off .



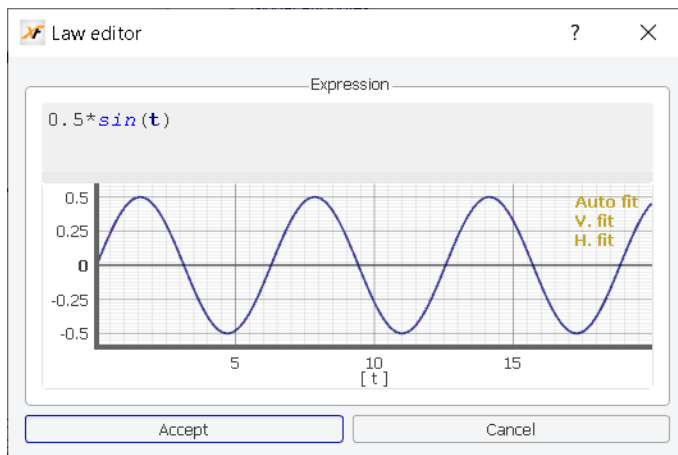
4. **Square brackets []:** The user can define this sort of field with a number, a [function](#), [tabular data](#) from a file, or a string (e.g. the name of a geometry object). Click between the brackets to make the field editable.



Law editor:

A law editor can be opened if you click on the following icon . The law editor allows you to use [functions and variables](#) to write laws and to preview them. Available functions and variables will be suggested automatically as you write them. Valid functions will be highlighted in yellow color, and valid variables will be highlighted in blue.

3.1 Graphical User Interface



Law editor and preview. Functions highlighted in yellow and variables in blue.

⚠ Please note: Input fields accept scientific notation (e.g. $0.00001 = 1e-05$) and decimals must be given with dots and not with commas (e.g. 2.5, not 2,5).

💡 Tip: Standard shortcuts for copy (**Ctrl** + **C**), cut (**Ctrl** + **X**) and paste (**Ctrl** + **V**) can be used to edit the *Project Tree*.

3.1.5.2 Functions

In the *Project Tree*, functions may be defined in the square-brackets editable-fields. The syntax of these functions should be as follows:

⚠ Please note: The laws accept scientific notation (e.g. $0.00001 = 1e-05$) and decimals must be given with dots and not with commas (e.g. 2.5, not 2,5).

💡 Tip: [User-defined variables](#) can be created from all the variables and functions presented below.

💡 Tip: Use the [Law editor](#) in order to preview your functions. Recognized functions will be highlighted in yellow, and variables in blue.

System variables

In general, a square-bracket field may be defined as a function of the following variables of the system:

t	Time
x	Spatial coordinate
y	Spatial coordinate
z	Spatial coordinate
pi	Number pi
$vx(x, y, z)$	x-component of velocity at a discrete point of the domain, given by coordinates (x,y,z)
$vy(x, y, z)$	y-component of velocity at a discrete point of the domain, given by coordinates (x,y,z)
$vz(x, y, z)$	z-component of velocity at a discrete point of the domain, given by coordinates (x,y,z)
$vmod(x, y, z)$	velocity magnitude at a discrete point of the domain, given by coordinates (x,y,z)
$sp(x, y, z)$	static pressure at a discrete point of the domain, given by coordinates (x,y,z)
$u(x, y, z)$	temperature at a discrete point of the domain, given by coordinates (x,y,z)

⚠ Please note: The coordinates of the discrete point, where the velocity or static pressure is evaluated, can be also defined as functions, e.g: $vx(\sin(t), 0, 0)$.

Additionally, for [post-processing](#) use, [custom fields](#) can be defined as a function of the following variables:

sp	Static pressure
rho	Reference density
vmod	Module of the velocity
vrt	Vorticity
qcrit	Q-criterion
cf	Skin friction coefficient
ti	Turbulence intensity
tp	Total pressure
u	Temperature
viscosity	Effective viscosity
nx	x-component of the surface normal
ny	y-component of the surface normal
nz	z-component of the surface normal

⚠ Please note: "rho" states for the reference density defined in **Project Tree > Materials > Fluid > Density** and it is thus a constant value.

In case of a free surface flow with two phases (**Project Tree > Materials > Fluid > Two phase model: On**), "rho" returns the reference density of the first phase (**Project Tree > Materials > Fluid > Density**).

For [Rigid body dynamics](#) solids, the following variables are further at the user disposal:

px, py, pz	Position in X, Y and Z directions
vx, vy, vz	Velocity in X, Y and Z directions
ax, ay, az	Acceleration in X, Y and Z directions
eux, euy, euz	Rotation angles in X,Y and Z directions
wx, wy, wz	Angular velocity in X, Y and Z directions

For [non-isothermal](#) cases and [non-Newtonian user defined](#) fluid, the user can also gain the access to the temperature variable:

theta	Temperature
-------	-------------

For [non-Newtonian user defined](#) fluids, the viscosity of the fluid may depend on the shear rate:

gamma	Shear rate
-------	------------

In [multiphase](#) cases:

vof	→ Volume of liquid phase
-----	--------------------------

When using the Discrete-Phase Model ([DPM](#)), it can be distinguished between fluid and particle system variables as follows:

3.1 Graphical User Interface

particle_x	
particle_y	Particle position in X, Y and Z directions
particle_z	
particle_vx	
particle_vy	Particle velocity in X, Y and Z directions
particle_vz	
particle_t	Particle time
particle_vN	Particle normal velocity during the collision with solids
particle_vT	Particle tangential velocity during the collision with solids
particle_vNdrift	Particle drift normal velocity during the collision with solids
particle_vTdrift	Particle drift tangential velocity during the collision with solids
particle_diam	Particle diameter
fluid_px	
fluid_py	Fluid position in X, Y and Z directions
fluid_pz	
fluid_vx	
fluid_vy	Fluid velocity in X, Y and Z directions
fluid_vz	
fluid_sp	Fluid static pressure
fluid_vrt	Fluid vorticity

In [FMI Standard](#):

fmu () Input variable for FMU computation

Arithmetic operators:

Functions may involve arithmetic operations such as:

- + Sum
- Difference
- * Multiplication
- / Division
- ^ Raise to power

Scalar functions:

Standard scalar functions are also available:

<code>sqrt(x)</code>	Square root	
<code>exp(x)</code>	Exponential	
<code>log(x)</code>	Logarithm	(Base e)
<code>log10(x)</code>	Logarithm	(Base 10)
<code>abs(x)</code>	Absolute value	
<code>sin(x)</code>	Sine	(x in radians)
<code>cos(x)</code>	Cosine	(x in radians)
<code>tan(x)</code>	Tangent	(x in radians)
<code>asin(x)</code>	Arcsine	(x in radians)
<code>acos(x)</code>	Arccosine	(x in radians)
<code>atan(x)</code>	Arctangent	(x in radians)
<code>rnd(x)</code>	Random	(Random number between 0 and x)
<code>floor(x)</code>	Floor	(Largest integer smaller or equal than x)
<code>ceil(x)</code>	Ceil	(Smallest integer larger or equal than x)
<code>j0(x)</code>	Bessel function j0	
<code>j1(x)</code>	Bessel function j1	

Interpolation functions:

A continuous function can be defined by discrete points interpolation, as follows:

<code>linearinterpolation(t; {t_0,value_0}; {t_1,value_1}; ...; {t_n,value_n})</code>	Linear interpolation of discrete points
<code>cubicinterpolation(t; {t_0,value_0}; {t_1,value_1}; ...; {t_n,value_n})</code>	Cubic interpolation of discrete points

where t stands for any of the independent variables (x,y,z,t).

Conditional statements:

`if (condition, statement1, statement2)` If condition is "true": it returns statement1, else: it returns statement2.

Valid relational operators: >, <

To express logical relationships in the conditional statement, it can be done using arithmetic operations of booleans:

`if(a AND b)` `if (a*b)`
`if(a OR b)` `if (a+b)`

Examples:		
<code>if((x>0), 1, 0)</code>	→	It returns a step function with step width 1 at x = 0
<code>if((x>0)(x<1), 1, 0)</code>	→	It returns a boxcar function, with value 1 for x between 0 and 1
<code>if(((x>1)+(x<0)), 1,0)</code>	→	It is 1 for x>1 and for x<0, it is 0 for x between 0 and 1

⚠ Please note: If the arithmetic operation is equal to zero then the statement is FALSE, otherwise it is TRUE.

3.1 Graphical User Interface

Initialization function in arbitrary geometries:

The following law allow the user to initialize a field inside a given fixed geometry (e.g VOF, pressure, temperature, velocity ...):

`insidegeometry("[geometry"])`, where [geometry] must be substituted by the shape name.

 **Tip:** It can be used with operators.

TIP It can be used with operators

Examples:	
<code>insidegeometry("geometry")</code>	It returns 1 inside "geometry"
<code>Initial Velocity > 0.5*y*insidegeometry("geometry")</code>	It returns 0.5*y m/s inside "geometry"
<code>Initial Velocity > if(y>0,insidegeometry("geometry"),0)</code>	It returns 1 m/s inside "geometry" for y>0
<code>Initial Velocity > insidegeometry("geometry-1")+insidegeometry("geometry-2")</code>	It returns 1 m/s inside "geometry" and "geometry-2"

 **Tip:** To use this law, the option "enable boundary condition" of the "geometry" should be disabled ([+Info](#)).

 **Please note:** The geometry must be watertight and the normal pointing inside.

 **Please note:** [Hierarchies](#) are not supported yet.

 **Please note:** This can only be used with fixed geometries, not [rigid body dynamics](#) nor [enforced geometries](#).

3.1.5.2.1 User-defined variables

User-defined variable can be created to use for the simulation setup as well as the post-processing. They are extremely helpful to parameterize the project, and for instance modify at only one place the value of a variable used in several inputs.

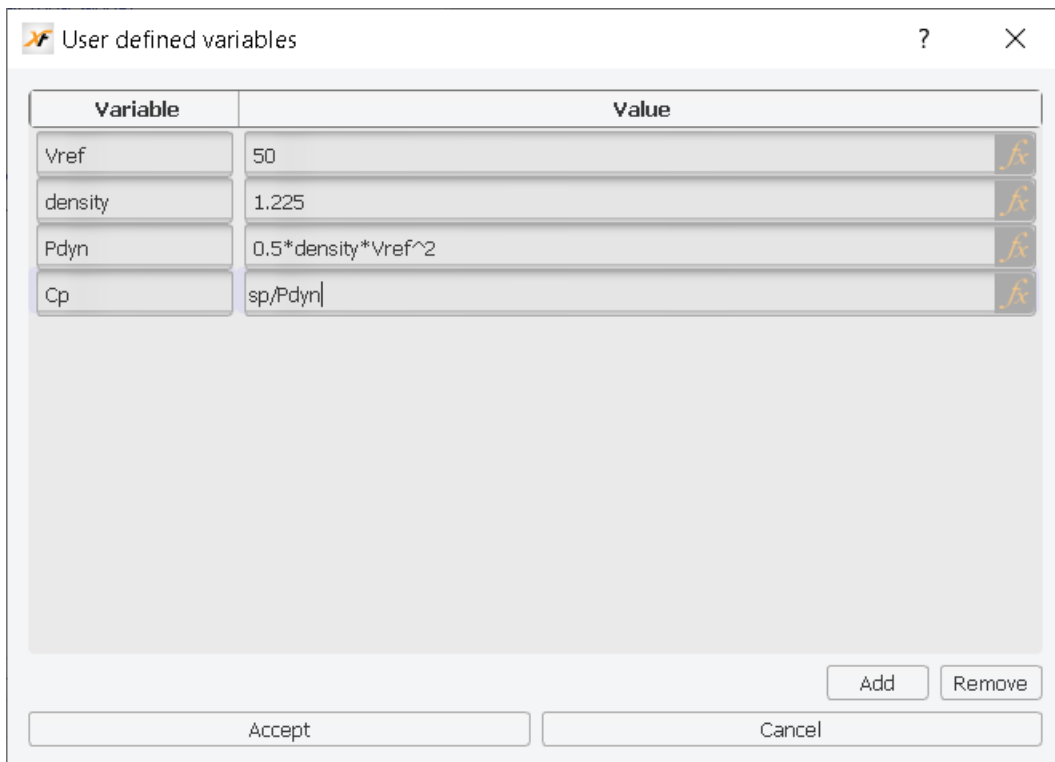
Main menu > Options > User-defined variables

The window is divided into two columns: the variable name on the left side, and the variable value on the right side. To add a new user-defined variable click on "Add" and a new empty line will show. You can remove any user-defined variable by clicking on "Remove". User-variables are defined for each projects and are saved in the XFlow project file. All the functions and variables defined in the [Functions](#) chapter can be used in the variable laws.

 **Please note:** A few important details must be considered:

1. The product "*" should be explicitly written in laws.
2. A user-defined variable can be defined with another user-defined variable, however the definitions are ordered and one must first define the variables called by the other variables.
3. The [Functions](#) defined in XFlow cannot be used as variable name.
4. The variables names are not case sensitive. For example, Cp = CP = cp = cP.

An example where the pressure coefficient Cp is defined is depicted on the following picture:



3.1.5.2.2 Tabular data

XFlow allows the user to fill an editable field with brackets [] by reading tabular data from a text file (.txt, .dat). There are three options available:

[No interpolation](#) `tabulardata (indep_variable_1, ..., indep_variable_n, "file.dat")`

[Linear interpolation](#) `tabulardatalinearinterpolated (indep_variable_1, ..., indep_variable_n, "file.txt")`

[Cubic Interpolation](#) `tabulardatacubicinterpolated (indep_variable_1, ..., indep_variable_n, "file.txt")`

[Sinc Interpolation](#) `tabulardatasincinterpolated (indep_variable_1, ..., indep_variable_n, "file.txt", order)`

where "indep_variable_1" or "indep_variable_n" stand for any of the following variables:

t	→	time
x	→	spatial coordinate
y	→	spatial coordinate
z	→	spatial coordinate
px, py, pz	→	position in X, Y and Z directions
vx, vy, vz	→	velocity in X, Y and Z directions

The table below shows an example of syntax of a text file:

3.1 Graphical User Interface

Example: file.txt or file.dat

```
// Comments
// x      y      value
0.1      1.4      0
2.3      -1.7     1
-5.5     3.8      0
```

To read this file, the syntax would be: `tabulardata(x,y,"file.txt")` or `tabulardata(x,y,"file.dat")`.

⚠ Please note: Although tabulardata files can be set using relative/absolute paths, it is best practice to store these in the same folder where the .xpf file is located and set only the filename in the tabulardata function. This is a requirement when it comes to XFlow automatically copying the files to a different saving destination folder. Moreover, those are included within the .XFZ compressed XFlow project file format.

⚠ Please note: Last column in the table is always the output value. The independent variables of the function do not need to be ordered by value in the text file. File names have to be text enclosed in quotes.

💡 Tip: Independent variables of the tabulardata can be a function of the system variables.
Examples: `tabulardata(2*x,"file.txt")`

💡 Tip: In fields with brackets, it can be operated over a function given as tabular-data. This allows the user to easily rescale the data. Examples: `[10*tabulardata(2*x,"file.txt")]`

No interpolation

The syntax is the following:

```
tabulardata(indep_variable_1, indep_variable_2, ..., indep_variable_n,
"file.txt")
```

It assigns to each computational element the closest value in the table.

Linear interpolation

The syntax is the following:

```
tabulardatalinearinterpolated(indep_variable_1, ..., indep_variable_n,
"file.txt")
```

XFlow can interpolate linearly the results between each points of the tabular data file.

Cubic interpolation

The syntax is the following:

```
tabulardatacubicinterpolated(indep_variable_1, ..., indep_variable_n, "file.
txt")
```

XFlow can interpolate the results between each points of the tabular data file using a cubic polynomial function.

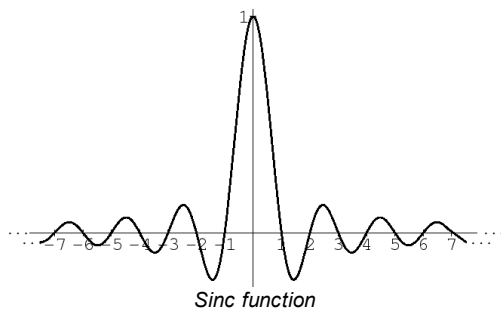
Sinc interpolation

The syntax is the following:


```
tabulardatasincinterpolated(indep_variable_1, indep_variable_2, ...,
indep_variable_n, "file.txt", order)
```

XFlow can also interpolate a tabular data from a file with a "sinc" function ($\text{sinc}(X) = \sin(X) / X$), where the "order" parameter is the number of the sampling points used in the interpolation kernel of the sinc function. This "order" parameter can take values between 3 and the maximum number of values available in the tabular data file.

⚠ Please note: Sinc interpolation requires a constant sampling spacing between points of the tabular data file, i.e. sampling rate must be constant.



⚠ Please note: The higher the "order" parameter is, the more accurate the interpolation will be. However, a computational cost may be observed if this parameter is high.

3.1.5.2.3 Unstructured Mesh

XFlow allows the user to fill an editable field with brackets [] by reading an unstructured mesh from a .vtk or .vtu file.

To read this file, the syntax would be: `unstructuredmesh((x,y[,z]), <file name>.vtu/.vtk, <field name>, [component])`

The meaning of each argument is as follows:

- (x, y) or (x, y, z): tuple with the coordinates where interpolate. Mesh coordinate system.
- <file name>.vtk/.vtu: file name and location.
- <field name>: name of the scalar or vector in the vtk/vtu file to be read by XFlow
- [component]: component of the vector. 0 by default.

Example:

```
0.001*unstructuredmesh((0.1 * x, 0.1 * y), "/home/XFlow/CFDrans/results/CFD_run_export.vtu",
```

3.1 Graphical User Interface

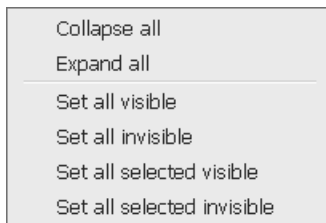
3.1.5.3 Trees management

A drop-down list will appear if right-clicking (**right mouse button**) in an empty space of any tree, it shows the tree management options:

- **Collapse all**: collapses all the elements in the tree
- **Expand all**: expands all the elements in the tree

If any object available in the tree has a visibility icon, the drop-down list will also show:

- **Set all selected visible**: this shows all the object selected
- **Set all selected invisible**: this hides all the object selected
- **Set all visible**: this shows all the objects in the present tree
- **Set all invisible**: this hides all the objects in the present tree

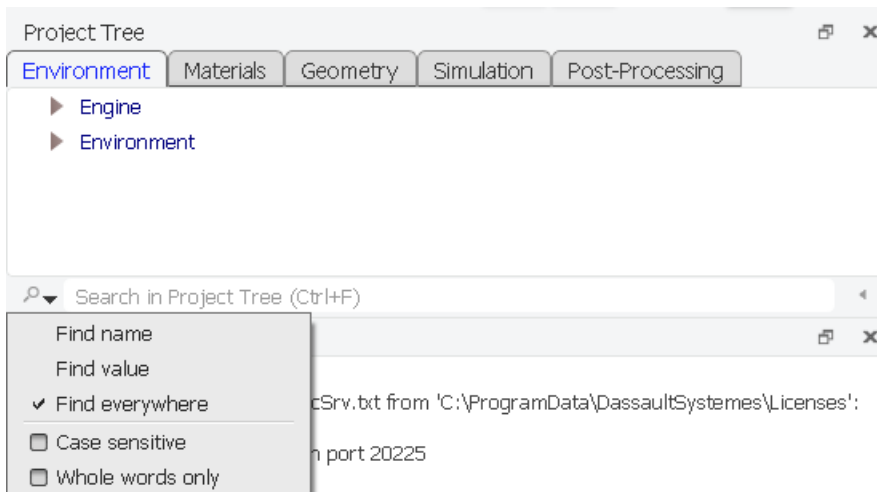


Tree-management drop-down list

Tip: You must right-click on one of the selected shape line in the tree to access this dialog.

3.1.5.4 Search in Project Tree

The search bar at the bottom of the *Project Tree* allows to quickly access an element from any tab based on its name and / or value:



Search bar in project tree with available options displayed

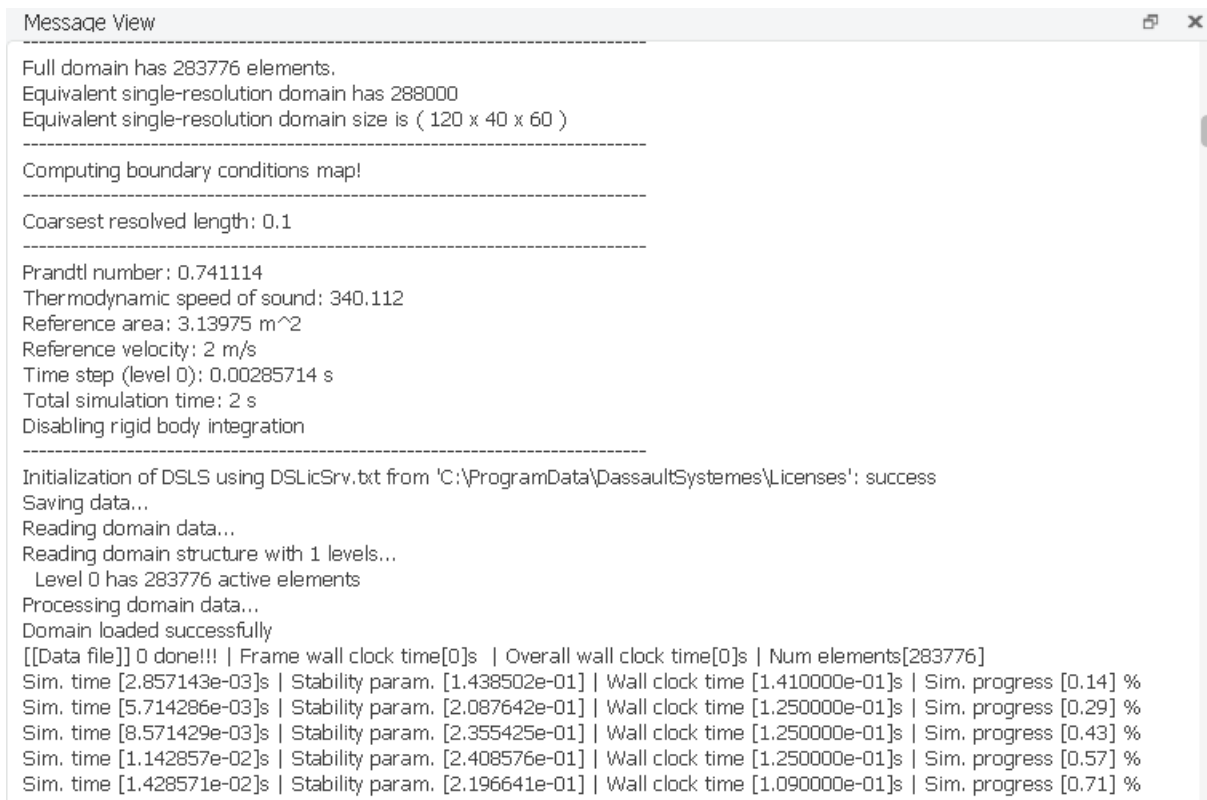
It is possible to filter the search to the name or value of the elements within the Project Tree if desired. *Find everywhere* will apply the search to all fields in the *Project Tree*.

Besides, the options of *Case sensitive* and *Whole words only* search are also available.



Example of search in project tree (Find everywhere, Case sensitive off)

3.1.6 Message View



GUI: Message View window

The log messages are classified in three categories:

- [INFO]: normal information message.
- [WARNING]: warning messages.
- [ERROR]: error messages describing a failure.

These messages are saved in the log files saved in the simulation folder.

This section will explain the different messages types:

- [Simulation progress](#)
- [Warnings](#)
- [Errors](#)

3.1.6.1 Simulation progress

The *Message View* is particularly useful to follow the computation progress as it displays the following information, for each solver iteration:

- [Sim. Time](#) (Simulation time)
- [Stability param.](#) (Stability parameter)
- [Wall clock time](#)
- [Num elems](#) (Number of elements)
- [Simulation progress](#) (percentage)

Simulation time

It is the total amount of physical time simulated already. In other words, it is the addition of the time steps computed so far.

Stability parameter

The stability parameter allows the user to check the solution's stability. The stability parameter is defined as:

S , Stability parameter

ν , Kinematic viscosity

$C_s = \frac{dx}{\sqrt{3} dt}$, Numerical speed of sound

$S_\nu = \frac{\nu}{C_s dx}$, Numerical kinematic viscosity

$S_v = \frac{V_{max} dt}{dx}$

$S_\rho = \frac{2 (\rho_{max} - \rho_{min})}{(\rho_{max} + \rho_{min})}$

$$S = \frac{\max(S_v, S_\nu, S_\rho)}{0.7}$$

The stability parameter must satisfy the Courant-Friedrichs-Lewy (CFL, S_ν) condition as well as a compressibility (S_ρ) and numerical viscosity (S_{mu}) condition, and thus its value must be less than 1. If it reaches 1.0, it means that somewhere in the domain one of these conditions is not being satisfied, the stability of the simulation is thus not ensured and XFlow displays the following message in the *Message View*:

“[WARNING] Stability parameter too high, check [pressure/velocity/viscosity] at (X,Y,Z) m or reduce the time step.”.

The values of the stability parameter can be graphically displayed in the [Function Viewer](#).

⚠ Please note: The Stability parameter is only a warning. The value output in the log is the maximum value within the entire simulation domain. This means a high value of the stability parameter could be caused by one single element of your fluid domain (element highly stressed between narrow regions, moving geometries, etc.), and may not be representative of the global stability of the simulation. If that is the case, the simulation could still be physically valid. The user should open the case and check the field reported in the warning message at the coordinate indicated in order to check any non-physical values.

⚠ Please note: The Stability parameter is not available when the Supersonic thermal model is enabled.

💡 Tip: It is recommended to keep the stability parameter around 0.1-0.3

Wall clock time

It is the amount of real time required by the engine to compute the time step. This time can be affected by external application other than XFlow, and does not only represent the CPU time but only the real wall-clock time.

Number of elements

If the topology of the lattice is not fixed, i.e. the refinement algorithm [Adaptive refinement](#) is enabled, the number of elements in the domain will vary in time. Only in this case, the number of elements at every solver iteration is displayed in the *Message View*.

Simulation progress

It indicates the percentage completion of the simulation, i.e. the amount of simulation time already computed over the total simulation time set.

Example: Message View output

The domain generator (generateDomain3d.exe) starts by creating the fluid domain. These messages are saved in the pre_processor.log file (see [XFlow files](#)) :

```
## DOMAIN GENERATION ##
XFlow Build 106.05
Execution line: C:\Program Files\Dassault Systemes\XFlow\generateDomain3d.exe Project.xfp -
log=2 /maxcpu=4
Computation limited to: 4 cores.
RLM license validation OK
Num cpus detected: 4
Generating Octree with 3 levels.
Level 0
Level 1
Level 2
Determining lattice/geometry intersection...

Determining regions with fluid
Automatically determining regions with fluid

1 discarded regions
1 identified regions
Generating node map.

Num active elements at level 0: 119168 / 125000
Num active elements at level 1: 24704 / 46656
Num active elements at level 2: 111616 / 175616
Total number of elements: 255488
Writing domain
Num boundary elements: 10088

Overall broken links: 87848
Domain successfully generated.
```

Example: Message View output

Exit code: [0::0]

The solver (engine-3d*.exe) now starts and proceed in the pre-processing by loading the fluid domain generated. All the following messages are saved in the project_name.log file (see [XFlow files](#)):

```
## SIMULATION START ##
XFlow Build 106.05
Execution line: C:\Program Files\Dassault Systemes\XFlow\engine-3d.exe Project.xfp /maxcpu=4 -
log=2
Computation limited to: 4 cores.
RLM license validation OK
Num cpus detected: 4
Allocating memory...
Processing geometry...
Done!
-----
Num levels: 3
-----

Level: 0
Num active fluid elements: 119168
-----

Level: 1
Num active fluid elements: 24704
-----

Level: 2
Num active fluid elements: 111616
-----

Processing geometry...
Done!
-----

Full domain has 255488 elements.
Equivalent single-resolution domain has 8000000
Equivalent single-resolution domain size is ( 200 x 200 x 200 )
-----
```

The solver sets the boundary conditions, estimates the main reference values and the time step:

```
Computing boundary conditions map!
-----
Coarsest resolved length: 0.2
-----

Prandtl number: 0.180259
Reference area: 3.99219 m^2
Reference velocity: 5 m/s
Time step (level 0): 0.001 s
Total simulation time: 0.4 s
-----
```

The solver initializes the fluid domain and saves the initial simulation frame (frame 0):

```
Saving data...
[[Data file]] 0 done!!! | Frame wall clock time[0]s | Overall wall clock time[0]s | Num
elements[255488]
```

For every solver iteration the Message View displays the following message:

```
Sim. time [1.000000e-003]s | Stability param. [1.080849e-001] | Wall clock time [2.543000e+000]
s | Sim. progress [1.00] %
Sim. time [2.000000e-003]s | Stability param. [1.097306e-001] | Wall clock time [2.621000e+000]
s | Sim. progress [2.00] %
...
```

Example: Message View output**Every time XFlow saves data, it is displayed:**

```
Saving data...
[[Data file]] 1 done!!! | Frame wall clock time[5.257]s | Overall wall clock time[5.257]s |
Num elements[255488]
...
```

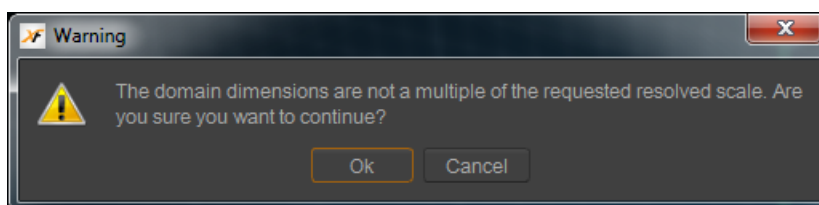
At the end of the computation, it is displayed:

```
Sim. time [5.000000e-002]s | Stability param. [1.212173e-001] | Wall clock time [3.916000e+000]
s
Saving data...
[[Data file]] 25 done!!! | Frame wall clock time[8.05]s | Overall wall clock time[167.311]s |
Num elements[255488]
Total execution time : 3 seconds 327 milliseconds .
## SIMULATION END ##
```

3.1.6.2 Warnings

The warning messages are intended to warn and anticipate about any possible issue. It should be considered but could be omitted if the warning is irrelevant.

- Stability parameter too high, check [pressure/velocity/viscosity] at (X,Y,Z) m or reduce the time step.
 - ☒ The [stability parameter](#) is equal or higher than 1 and the stability of the simulation may be compromised. Check the pressure, velocity or viscosity at the (X,Y,Z) m coordinate which points to the area where the stability parameter has the highest value. You may change your boundary conditions, lattice resolution, or simply reduce the [time step](#).
- No reference pressure defined in the simulation setup.
 - ☒ This warning appears if the pressure is not defined at any of the problem boundaries or no Reference pressure point is specified.
- No reference velocity defined in the simulation setup. The reference velocity is set to 1 m/s by default, please set a custom reference velocity if required.
 - ☒ This warning appears if the inlet velocity is 0 m/s or if there is no inlet velocity boundary specified.
- Your reference velocity cannot be estimated due to the inlet law. Please set a custom reference velocity.
 - ☒ This warning appears if the inlet velocity is a non-constant law depending on the variables t, x, y, z, sp, etc...
- The following popup window shows a warning that appears if any of the domain dimensions is not a multiple of the [Resolved scale](#), leading to a mismatch between the [Resolved scale](#) setup by the user in the *Project Tree* and the real one used in the simulation.

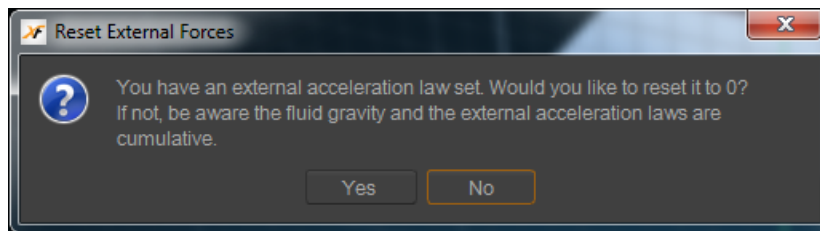


For instance, these may be of utmost importance in *Single Phase* external with symmetric lateral

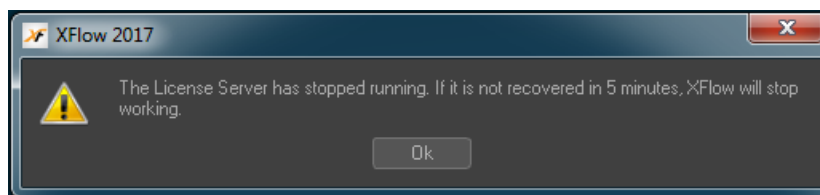
3.1 Graphical User Interface

boundaries: To ensure symmetry in Z-axis, the resolved scale must be exactly a multiple of the Z-dimension of the domain.

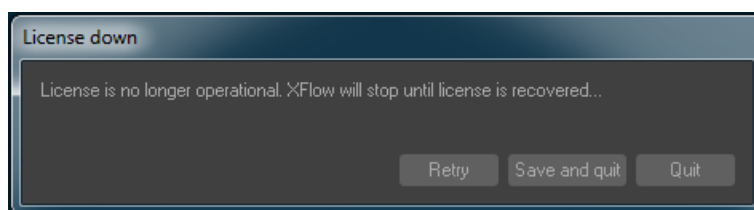
- The following popup window warns the user that the case setup contains a nonzero [Fluid gravity](#) and also a nonzero [external acceleration](#) that are cumulative. For instance, if by mistake the user defines the gravity as an [External acceleration law](#) when it is already taken into account in the field [Fluid gravity](#), he can reset the first one to zero; otherwise the user can neglect the warning.



- If connection with the license server is lost or the license server has stopped suddenly, the following message will show. XFlow will keep running for 5 additional minutes and will then stop if connection to the license server is not recovered or the license server restarted.



- After 5 minutes from the above message, the following message will pop-up and the interface will be disabled and simulation paused (process is not terminated yet). If the license server is recovered, XFlow will detect it automatically and the computation will resume or click on "Retry" to reconnect to the license server. Otherwise the user can save the simulation before quitting.



3.1.6.3 Errors

Error messages are critical and will prevent the simulation from running. They should be considered seriously in order to carry out the simulation successfully.

- Non-physical values (infinite, NaN) detected in the fluid domain [`coord( -0.986233, 0.510175, -0.063925 )`], the simulation has stopped. Please check if inconsistent high pressure or velocity regions are found in the simulation or try to reduce the timestep.
 - ☒ Fluid variables (pressure, velocity, viscosity, etc.) diverged locally in the domain at the coordinates

indicated by [coord(X,Y,Z)] m. The simulation is therefore stopped and the data are saved at the time step where the incidence happens. You may try to check boundary conditions, lattice resolution, or simply reduce the time step to avoid it.

- Non-physical values detected (infinite, NaN) in the numerical data, the simulation has stopped. Please check if inconsistent high pressure or velocity regions are found in the simulation or try to reduce the timestep.
 - ☒ Some numerical data in the Function Viewer diverged. The simulation is therefore stopped and the data are saved at the time step where the incidence happens. You may try to check boundary conditions, lattice resolution, or simply reduce the time step to avoid it.
- No lattice elements in the fluid domain. Please check your spatial discretization parameters and geometries and generate the domain again.
 - ☒ The fluid domain did not generate correctly as the lattice has 0 elements. The most common reasons for this message are: the geometry is not watertight, the surface orientations are incorrect and do not point toward the bulk fluid volume, the resolution too large for the size of the domain. Please check your geometries, in internal simulation you may use the [seed point](#) in order to help the domain generation.
- Impossible to interpolate node data at (-4.20e+00,4.80e+00,1.08e-06) m. Please check the lattice structure to avoid lattice transition near boundaries or modify the buffer zone length.
 - ☒ If the [LODI](#) boundary conditions are used with a [buffer zone](#), there may be issues if the lattice transition happens near the boundaries where the LODI condition is applied. Try to change the domain structure to avoid transitions near the LODI boundaries to avoid this issue. The calculation will carry on, however the interpolation for the LODI boundary cannot be achieved properly.
- project.xfd not found. Please run the domain generator before running the engine.
 - ☒ Indicates the engine is running without the fluid domain generated (project.xfd). Run the executable generateDomain3d to generate the fluid domain.

3.1.7 Graphic View

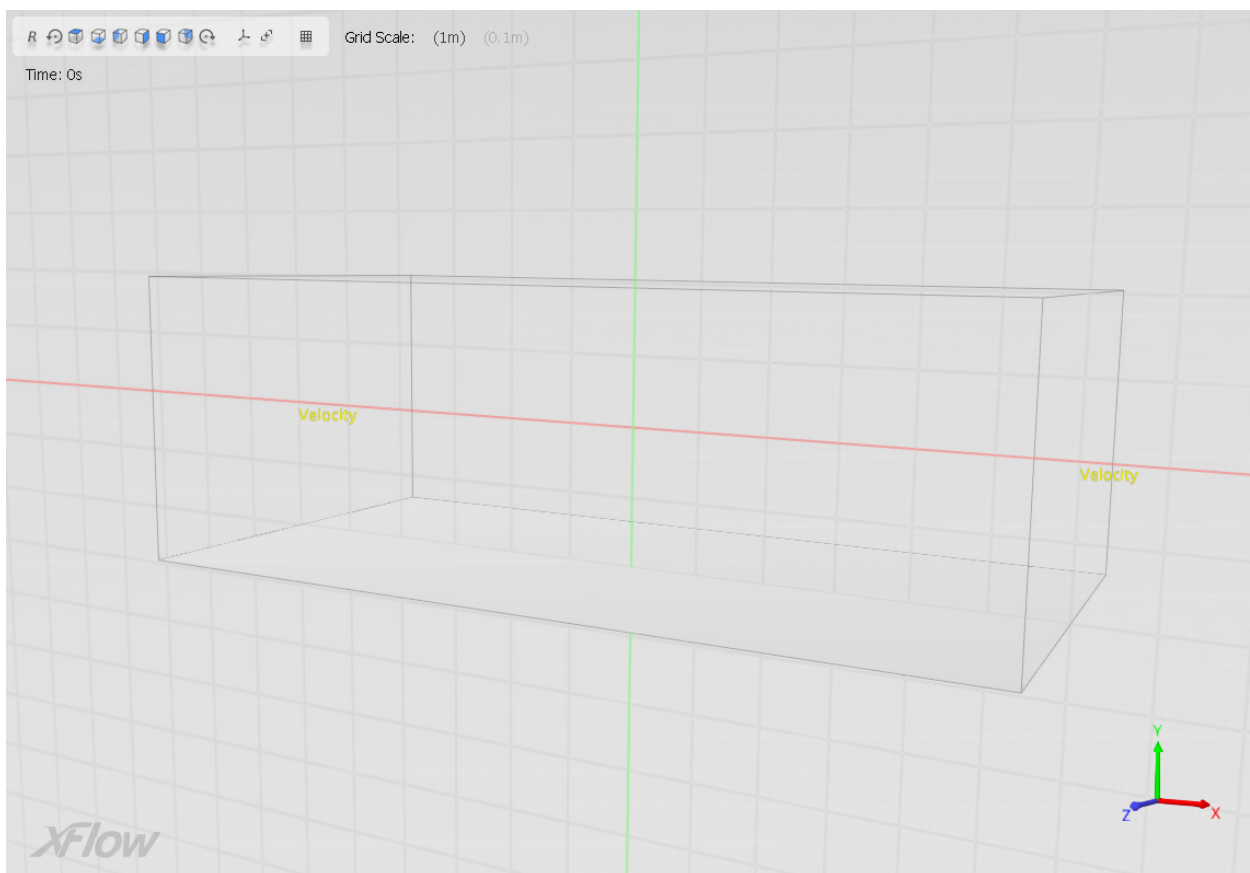
The *Graphic View* window displays the graphical output. Several *Graphic View* windows can be created by:

Main menu > Views > New Graphic View, or 

The options of the *Graphic View* can be set in:

Main menu > Options > Preferences > Graphic View, or 

3.1 Graphical User Interface



Graphic View window

Mouse navigation

The navigation in this window allows the user to move the camera to change the geometry view.

Zoom step by step
(centered on cursor location)
Continuous zoom
Rotation
Translation
Move view to the sides
Move view forwards and
backwards

Wheel

Alt + right mouse button drag
Alt + left mouse button drag
Alt + middle mouse button drag
Alt + Shift + left mouse button drag
Alt + Shift + right mouse button drag

Predefined views

There are predefined views that correspond to the orthogonal and perspective projections of the model:

	Reset
	Rotate view left
	Top view
	Bottom view
	Left-hand side view
	Right-hand side view
	Front view
	Back view
	Rotate view right
	Perspective view
	Fit all
	Show grid

Graphic View Menu

Click on the *Graphic View* window with the **right mouse button** to open the *Graphic-View* drop-down menu.

If no geometry is selected, the menu will show the following options:

Select all geometries	➤ To select all the simulation geometries of the project simultaneously
Select all post-processing geometries	➤ To select all the post-processing geometries of the project simultaneously
Deselect all	➤ To deselect all the geometries of the project simultaneously
Show colour legend	➤ Only available for external analysis

The available graphic view settings depend on the domain type ([Virtual wind tunnel](#) / [Generic rectangular domain](#)):

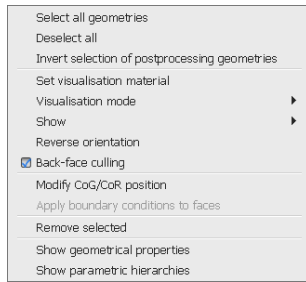
☒ Show domain box	➤ Show/Hide the box that indicates the limits of the domain
☒ Show domain floor/walls	➤ Show/Hide the ground wall that indicates the wall boundary condition on the floor
☐ Show domain floor/walls - shadow only	➤ Show/Hide the projection of the shadows onto the ground wall
☒ Show domain boundary conditions	➤ Show/Hide domain boundary conditions

If no geometry is selected but there are data loaded being visualized, an additional option is available:

Select all geometries	➤ To select all the geometries of the project simultaneously
Deselect all	➤ To deselect all the geometries of the project simultaneously
Domain visualization	➤ Only available for external analysis
Show colour legend	➤ To show/hide the color legend

If a geometry object is selected, then the *Graphic-View* menu shows the following options:

3.1 Graphical User Interface



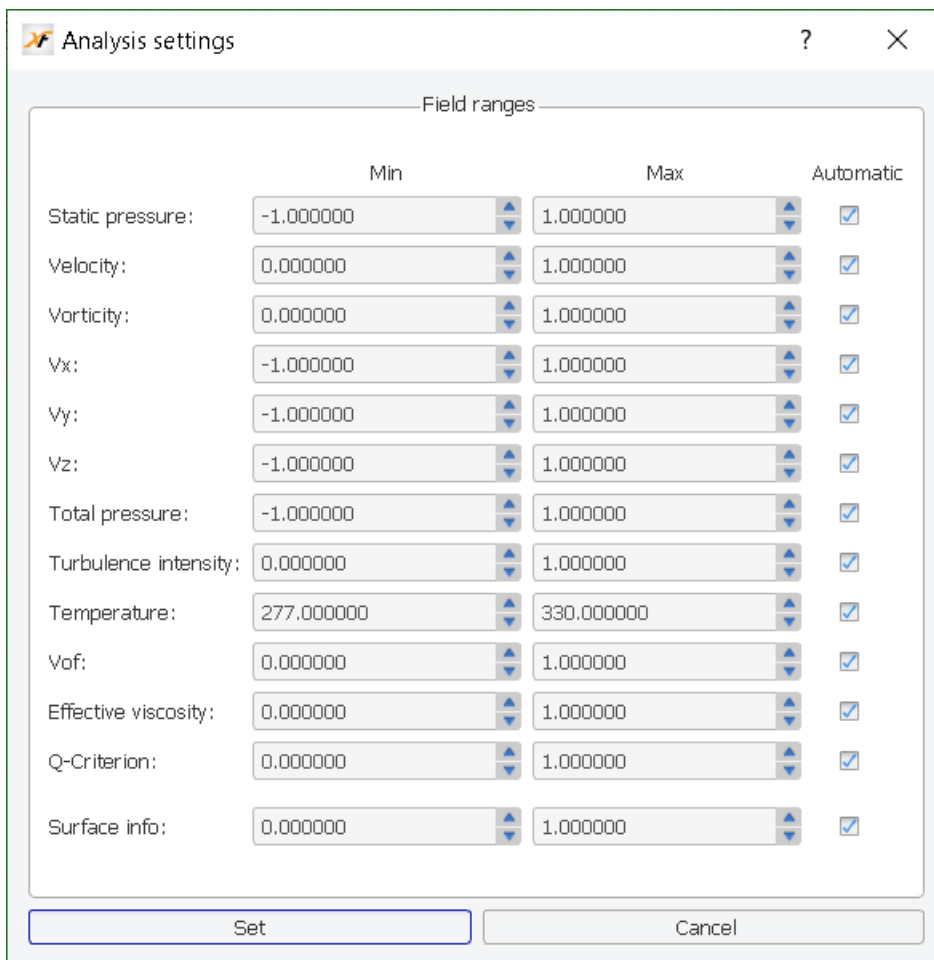
Graphic View Menu

- To select all the geometries of the project simultaneously
- To deselect all the geometries of the project simultaneously
- To select the whole complementary set of simulation geometries, post-processing geometries or isosurfaces
- [To set the visualization material and color](#): rubber, plastic, metal, glass, ...
- [To set the visualization mode](#): shading, wireframe, bounding box, mesh.
- [To show local axes](#) and [normals](#)
- To invert the direction of the [surface normals](#)
- To activate/deactivate the [back-face culling](#) visualization option
- To [modify the coordinates of the centre of gravity and rotation](#)
- To apply a [boundary condition](#) to a geometry-object surface
- To [remove selected object](#)
- To see a report of the relevant [geometrical properties](#)
- To see a report of the relevant [parametric hierarchies](#)

3.1.7.1 Legend

In the Graphic View, the legend range is by default determined by the maximum and minimum values of the numerical data; to manually modify this range go to:

Main menu > Simulation Data > Analysis settings or .



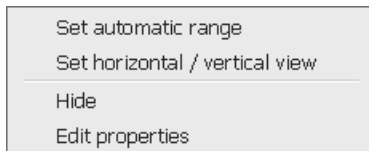
Analysis setting window

To change the visualization range (maximum and minimum) uncheck the "automatic" box, and insert manually the maximum and minimum values.

Another possible way is to edit the Color Bar directly in the GUI, for this select the color bar by clicking it. If you click on the maximum and minimum values, you can edit them to limit the range.

Additionally, further options are available to customize the color bar by right clicking on it, once selected.

This will display various options:



Color bar options dialog

Set automatic range: If the maximum and minimum values have been edited, this will restore to the automatic values which will correspond to the minimum and maximum in the fluid domain at that timestep.

Set horizontal / vertical view: Changes the color bar to be either horizontal or vertical.

 **Tip:** The color bar can be resized by clicking the bottom corner of the bar, and dragging either horizontally, vertically and diagonally.

It can also be resized if you click on it, and scroll with the central mouse button.

Hide: This hides the color bar.

Edit properties:

- Count: Number of value labels that are displayed next to the color bar
- Precision: Decimal precision of the values on the color bar
- Color: Sets the color of the color bar text
- Gradient: Changes the color maps for the color bar

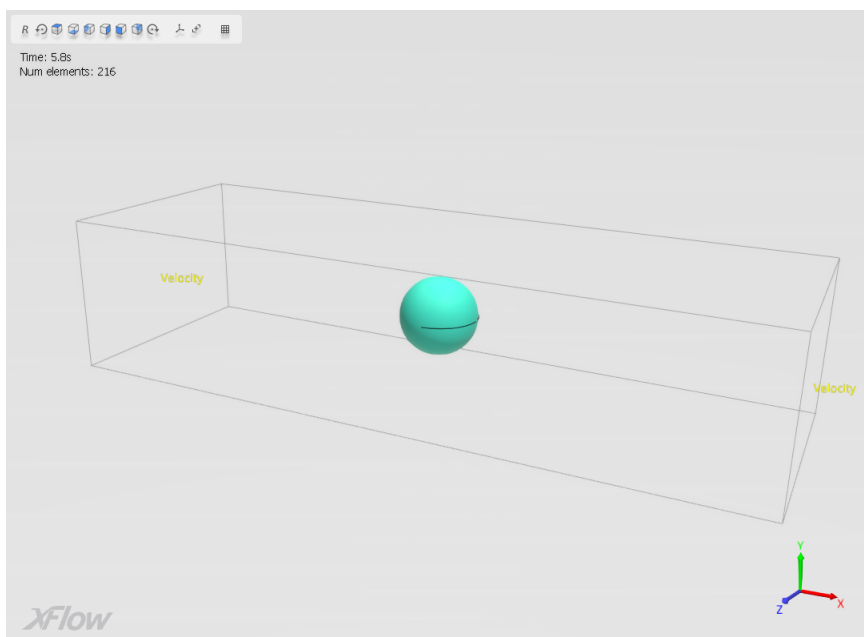
3.1.7.2 Gizmos

Main menu > Geometry > Selection filter > Object filter

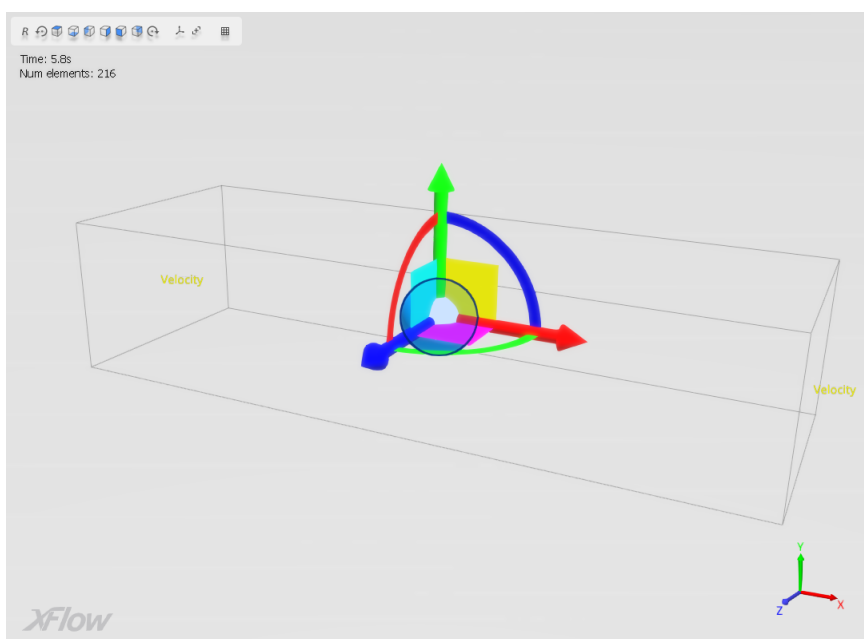
Gizmos are shown when a geometry is selected with the object filter, . Gizmos let the user easily do the following operations:

- [translate](#) a geometry object,
- [rotate](#) a geometry object,
- [scale](#) a geometry object, and
- [move cutting planes](#).

3.1 Graphical User Interface



Graphic View - View only mode



Graphic View - Sphere selected with object filter - Guizmos

3.1.8 Function Viewer

The *Function Viewer* is an interactive window where the user can either:

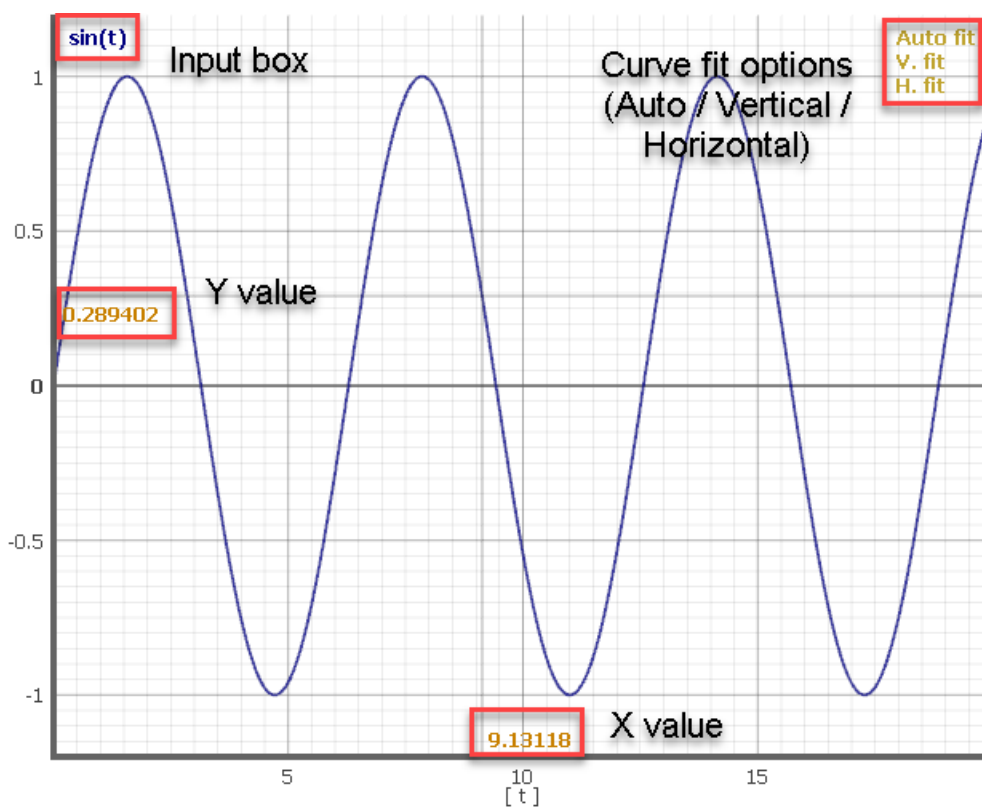
- [Visualize a function](#)
- Analyze and export the results of the simulation (see [Post-Processing](#) chapter).

Different instances of the *Function Viewer* can coexist to allow the visualization of different functions simultaneously. A new *Function Viewer* can be created either by:

- clicking on  at the [Toolbar Windows](#), or
- **Main menu > Views > New Function Viewer**

The function viewer appearance is shown in the figure below, where the following features are indicated:

- **Auto fit function:** Resize the X and Y axis to fit the function.
- **Vertical fit function:** Resize the Y axis to fit the function.
- **Horizontal fit function:** Resize the X axis to fit the function.
- **Input box:** To write [functions](#) and plot them on the *Function Viewer*.
- **X value:** X value of the point under the mouse.
- **Y value:** Y value of the point under the mouse.



The window view can be manipulated as follows:

Left mouse button + drag	→	Pan
Middle mouse button + drag / Roll the mouse wheel	→	Zoom
Alt + right mouse button + drag rightward	→	Horizontal zoom in
Alt + right mouse button + drag leftward	→	Horizontal zoom out
Alt + right mouse button + drag upward	→	Vertical zoom in
Alt + right mouse button + drag downward	→	Vertical zoom out

Function visualization

The user can directly visualize a function by selecting the  icon, which will open a law editor (only when in [Setup](#) or [Editing](#) mode). Additionally, when the software is in [Post-Processing](#) mode, a function

3.1 Graphical User Interface

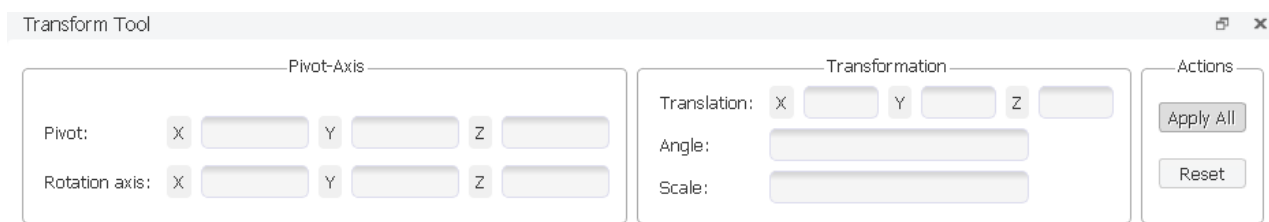
that has been set in any of the square brackets [] fields of the *Project Tree* can be plotted in a law editor by **middle clicking** on it.

3.1.9 Transform Tool

The *Transform Tool* (available clicking the **right mouse button** on the toolbar) can be used to move, rotate and scale a geometry object.

More information regarding its use is described in the section: [Modify Geometry](#).

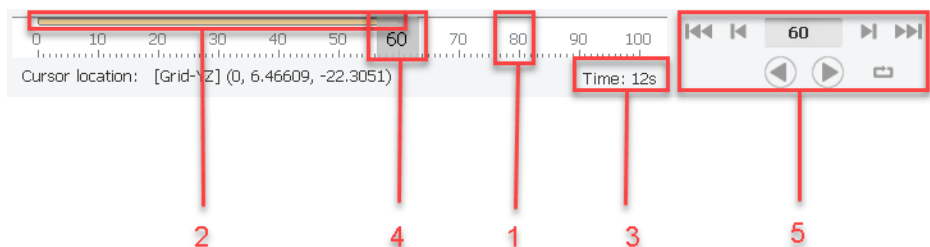
Tip: All the transformations applied are updated in the [Geometry tab](#) of the Project Tree.



Transform Tool window

3.1.10 Time controls

The time evolution of the simulation is monitored and controlled by means of the timeline and the playback panel, shown in the figure below:



Time controls: timeline & playback panel

The timeline reflects the progress of the calculation and allows the navigation through the results, that have already been calculated. It has four main components:

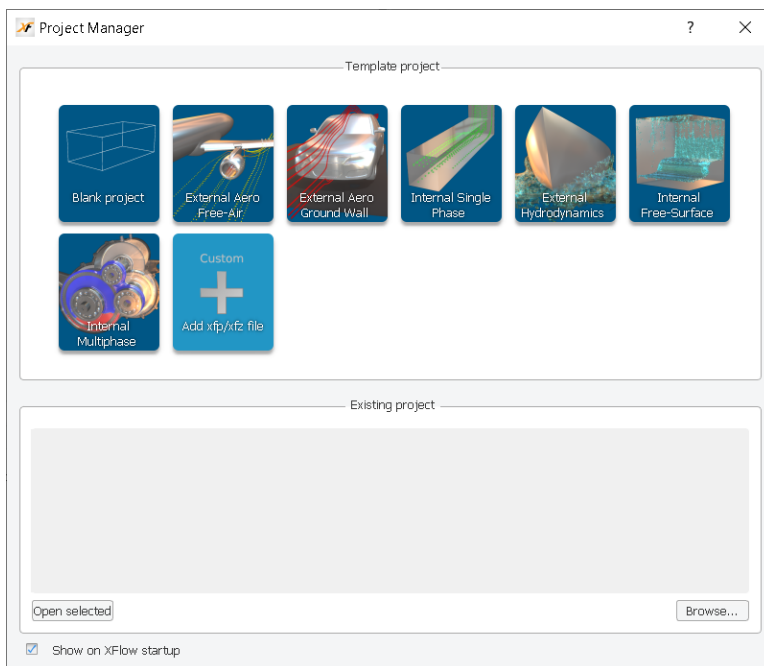
1. Frame number
2. Time bar, that indicates the progress of the simulation, i.e. the frames that have been calculated
3. Time of the frame being currently visualized.
4. Frame number, indicating the frame that is currently visualized/loaded in the *Graphic View*. To visualize another frame, just drag the frame-number button along the timeline, click on the frame number or use the playback controls.
5. Playback controls:

⏮	Go to the last frame
⏪	Go to the first frame
▶	Play forward / Stop
◀	Play backward / Stop
⏭	Move to the next frame
⏩	Move to the previous frame
60	Current frame
🔄	Loop the playback

Tip: Since XFlow takes computer resources to display the results in the *Graphic View*, it is recommended to locate the frame cursor after the last result saved (i.e. in the blank space on the right hand side of the timeline) during the computation.

3.2 Executing XFlow

To start XFlow, execute `XFlow.bat` (Windows) or `xflow.sh` (Linux) located in the installation folder, or double-click on the XFlow icon shortcut . The application will then display the default layout of the [Graphical User Interface](#) and the *Project Manager* window will be automatically opened.



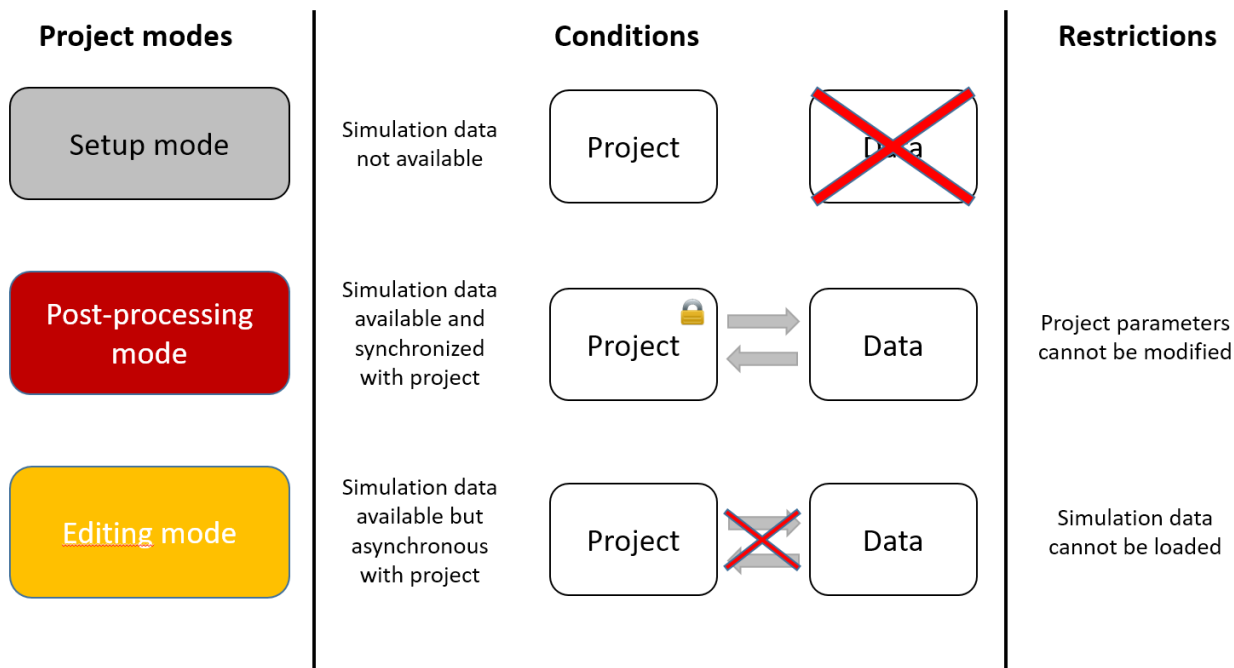
Project Manager window

As shown in the figure above, the *Project Manager* window allows the user to:

- **Create a new project:**
 - Select **Blank project** to start a new setup.
 - Select one of the **Templates** displayed to load some pre-defined settings for different cases.
- **Open an existing project:**
 - Choose a recent project among those shown in the "Existing project" list, press the button **Open selected**
 - Press the button **Browse...** and browse the project through the file system

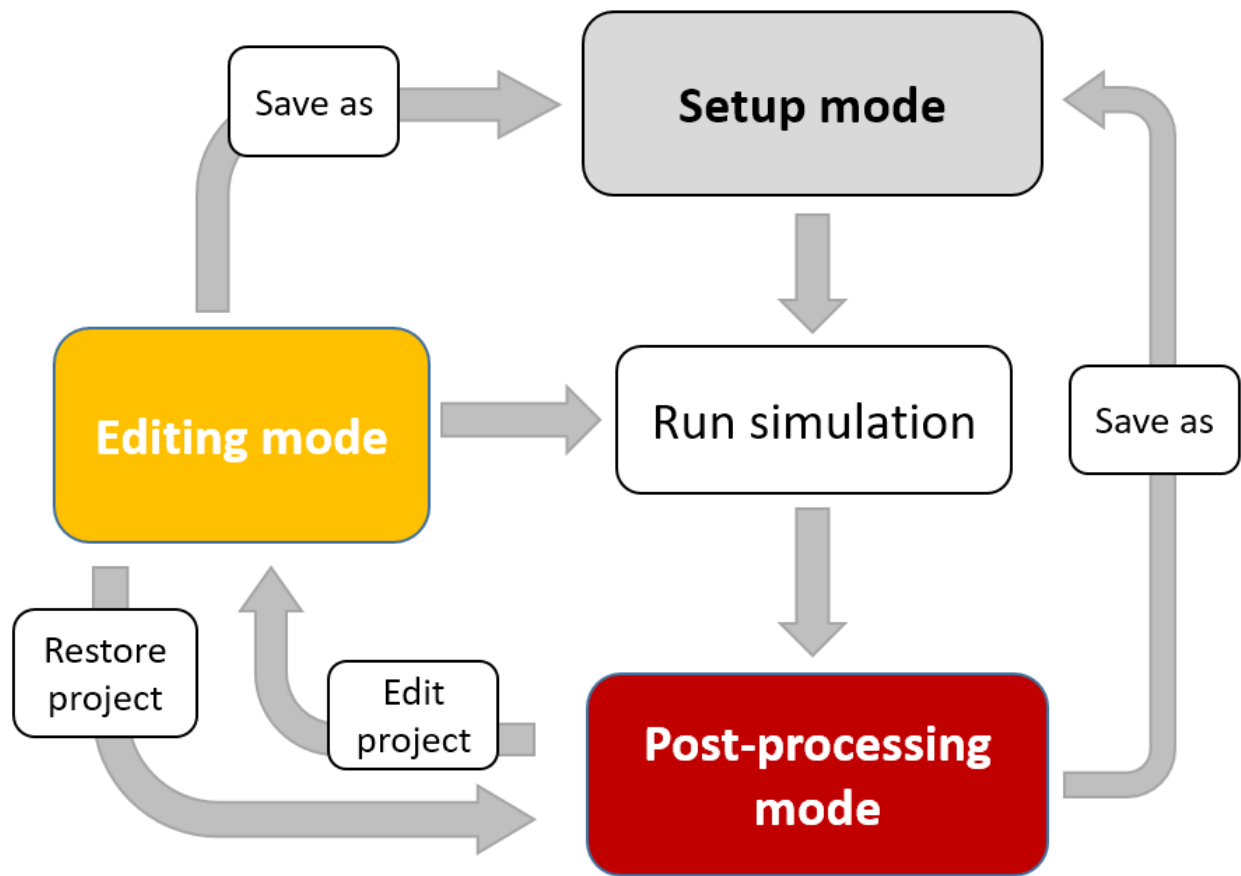
3.3 Project modes

XFlow has several interface modes depending on the current project state. A summary of the different project modes is depicted on the following figure:



Project modes description

A summary of the project modes workflow is depicted on the following figure:



Project modes workflow

Setup mode

The setup mode is the regular project mode and corresponds to projects which have not been simulated yet. Projects in setup mode include new empty projects and projects saved as a new one.

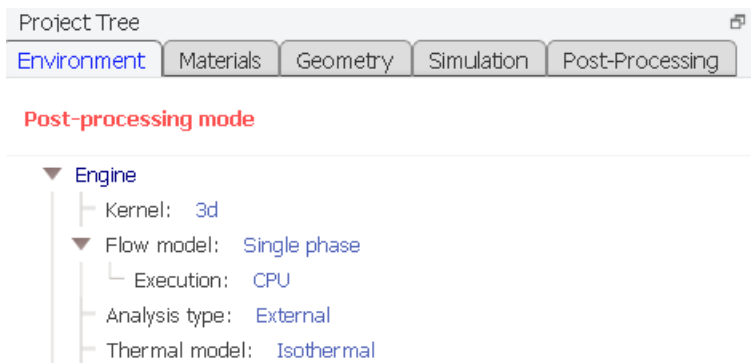
⚠ Please note: If a project is saved as a new one and a project with existing data is selected, [Editing mode](#) will be enabled instead of Setup mode in order to avoid conflictive state.

Post-processing mode

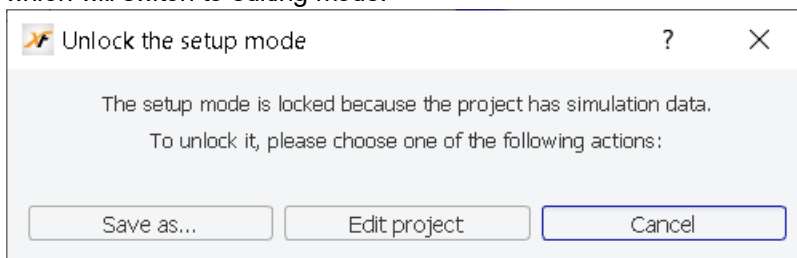
The post-processing mode is set for projects that contain simulation data. In this mode, the simulation data are synchronized with the simulation parameters which means the project is exactly as it was executed. Simulation parameters cannot be edited in order to maintain synchronization between project and simulation data. Post-processing mode is enabled automatically when running a simulation, or when loading a project with simulation data which has not been edited.

Post-processing mode is announced at the top of all the [Project Tree](#) tabs except the post-processing tab as follows:

3.3 Project modes



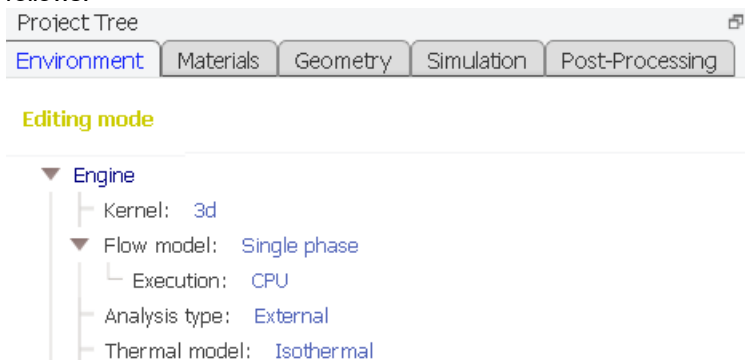
A menu opens if you click on Post-processing mode in the top of the project tree, and proposes two options in order to unlock the project tree: save as a new project which will switch to setup mode, or edit the project which will switch to editing mode.



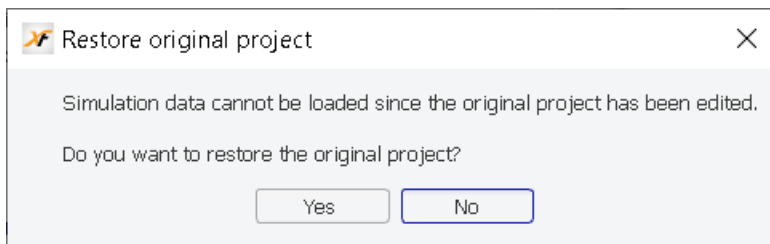
Editing mode

XFlow enters in Editing mode in order to edit projects which have been simulated already. This mode allows to modify simulation parameters and run the simulation again hence overwriting previous simulation data. Note that when the editing mode is enabled, the simulation data cannot be loaded anymore to prevent from visualizing asynchronous set of data and simulation parameters and hence be in an inconsistent state.

Editing mode is announced at the top of the all the [Project Tree](#) tabs except the post-processing tab as follows:



A menu opens if you click on Editing mode in the top of the project tree, and proposes to restore the original project file. If accepted, the original project file will be restored and the project reloaded again hence switching back to post-processing mode and allowing to load the original simulation data.



3.4 XFlow files

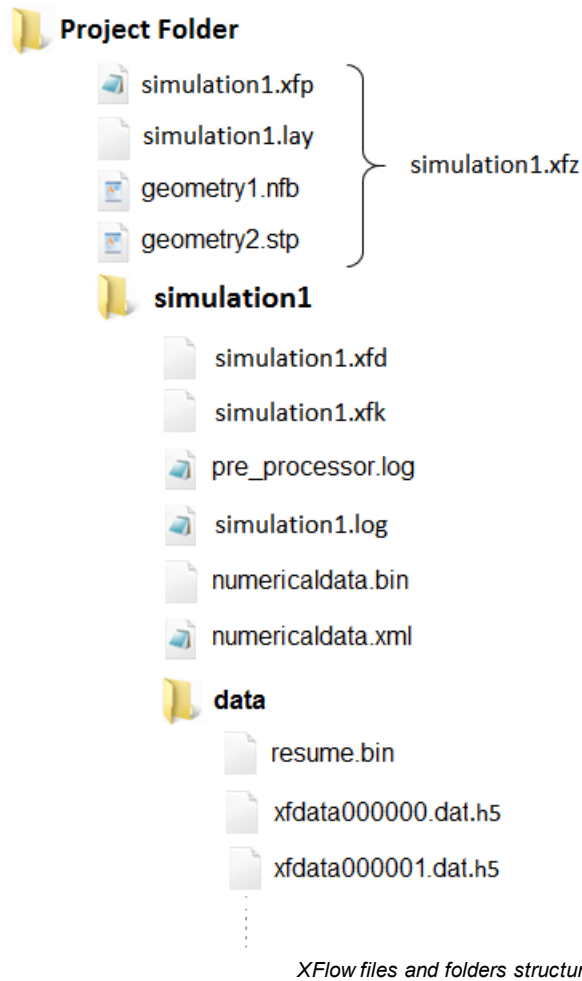
XFlow saves the working project into an **.XFP** file or **.XFZ** file. The **.XFP** file is in ASCII format and can be easily edited with a text editor.

The content of the **.XFP** file is in XML syntax and the structure is similar to the *Project Tree*.

The **.XFZ** file is an archive that gathers all files required to run the simulation, geometries included. It is useful to send your project to someone or archive it, however it may duplicate geometries if several projects are created with the same geometry files.

 **Tip:** It is recommended to use the XFP format for better performance in saving/loading project, and to use the .XFZ format in order to send the project to someone or archive it.

Simulation data are saved in a sub-folder where the project file is located, named as the project, and containing the log files, fluid domain, numerical data. The volumetric data are saved in a sub-folder named "data" ; as shown in the figure below:



Project folder

- **.XFP** Project file in ASCII format that describes the project in XML format. Contains all the parameters, boundary conditions and links to the geometry files used in the simulation.
- **.LAY** File that stores the layout of the GUI windows.
- **.NFB** XFlow native binary file format for geometries. The STL geometries imported in XFlow are saved automatically in this format.
- **.XFZ** Archive gathering all files required to run the project: .xfp, .lay, and geometries.

Simulation folder

- **pre_processor.LOG** ASCII text file containing the domain generator output shown in the *Message View* window. It describes the octree structure of the lattice.
- **project_name.LOG** ASCII text file containing the solver output shown in the *Message View* window. Time steps iterations and information about the simulation are shown, see [Simulation](#)

[progress](#) chapter.

- **.XFD** Fluid domain file. It describes the octree structure of the lattice and is required by the solver to run a simulation.
- **.XFK** Backup file in ASCII format used to recover the project settings of the computation.
- **numericaldata.BIN** Numerical data file used for the graphs are saved in this file. This is a binary file format.
- **numericaldata.XML** Numerical data structure used for the graphs are saved in this file. This XML file allows to read the numericaldata.bin.
- **axisforcesdistributionX/Y/Z.txt** Forces distribution/cumulation over all the geometries in X/Y/Z direction at each time step. These files are generated only when the [Save axis force distribution](#) option is enabled.

Data folder

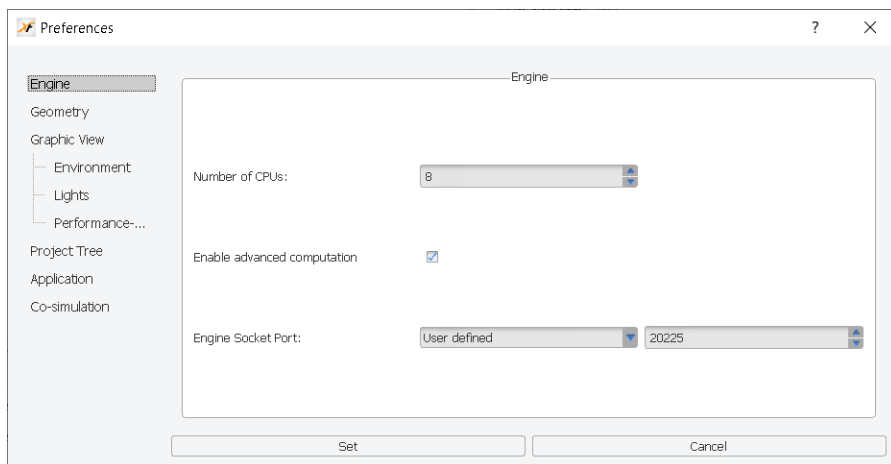
- **resume.BIN** Resume file used to resume a simulation. It is saved only if the [Save resume file](#) option is enabled.
- **xfdata00000N.dat.h5** Data file of frame number N containing all the data calculated by XFlow:
 - Volumetric data: Static pressure (p_s), Velocity modulus (v_{mod}), Vorticity (v_{rt}), Q-criterion (q_{crit}), X/Y/Z components of velocity ($v_x/v_y/v_z$), Turbulence intensity (t_i), Temperature (u), Effective viscosity (vis), Volume of liquid phase (vof)
 - Surface data: Skin friction (C_f), Pressure Coefficient (C_p), Y^+ , P^+
 - Animated objects: Cables, FSI-NASTRAN, Geometries
 - Markers
 - Domain structure
 - Averaged and Standard deviation data

 **Tip:** The structure and the data of the **xfdata00000N.dat.h5** files can be read and extracted by an HDF reader program such as the open source program [HDFView](#)

 **Tip:** Modified (and unsaved) project files (.XFP) are marked with a star * at the top right-hand side of the project route in the GUI.

3.5 Preferences

Main menu > Options > Preferences, or toolbar icon 



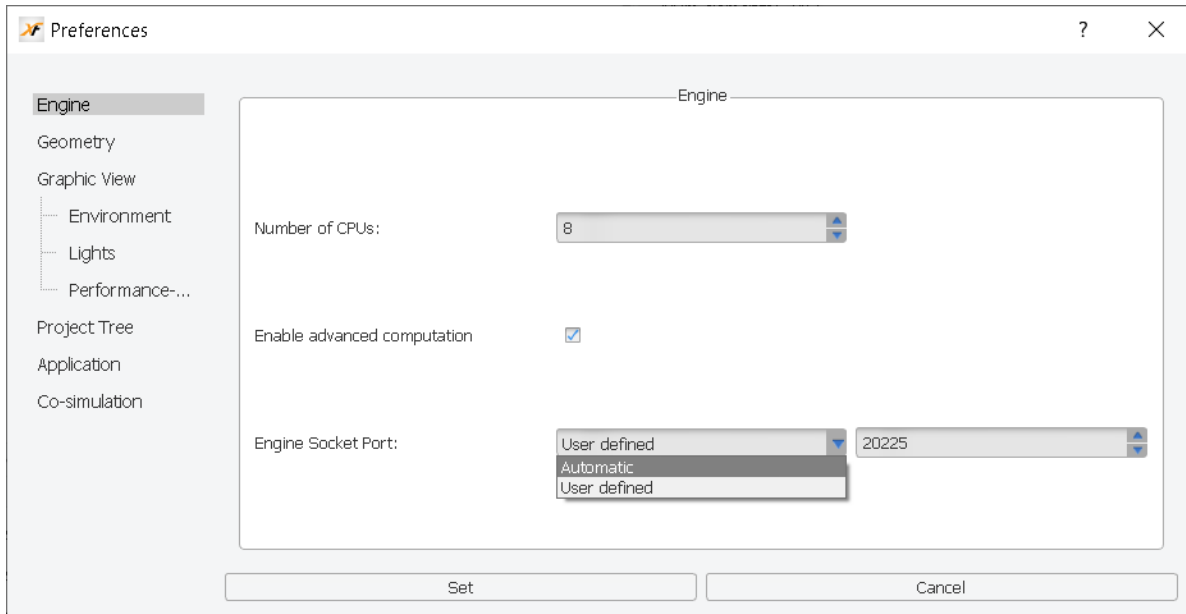
Preferences window

The *Preferences* window allows the user to customize the appearance of XFlow and the engine settings. It is divided into four sections:

- [Engine](#)
- [Geometry](#)
- [Graphic View](#)
 - [Environment](#)
 - [Lights](#)
 - [Performance-Quality](#)
- [Project Tree](#)
- [Application](#)

3.5.1 Engine

Main menu > Options > Preferences > Engine



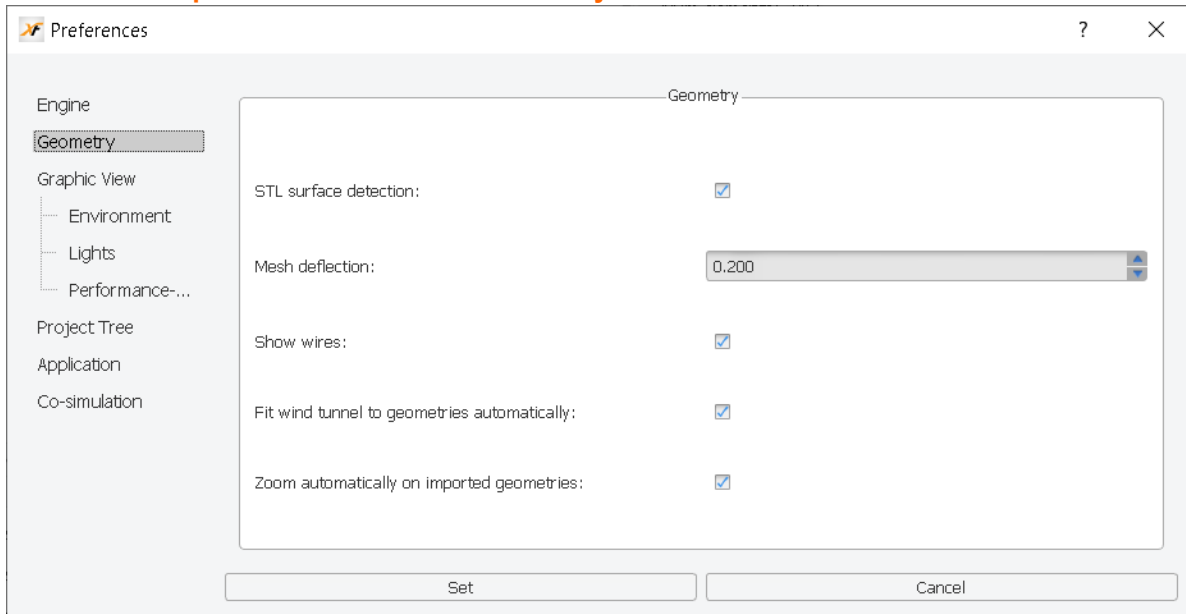
Preferences window: Engine section

In the *Engine* section of the *Preferences* window, the following options can be activated/deactivated:

- **Number of CPUs:** Specifies the number of CPU that will be used in the computation. The maximum number of CPUs is defined in the license agreement.
- **Enable advanced computation:** *Only available in Expert mode.*
- **Engine Socket Port:** Port that will be used by the XFlow Process Manager. This can be set to Automatic (the software will automatically detect and choose an available port) or to User defined.

3.5.2 Geometry

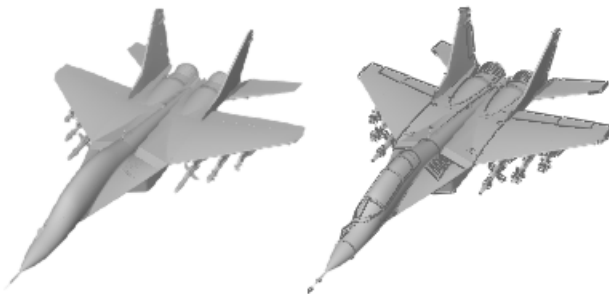
Main menu > Options > Preferences > Geometry



Preferences window: Geometry section

The Geometry section of the *Preferences* window shows options related to geometries:

- **STL surface detection:** This allows the user to select surfaces of the tessellated geometries such as . STL and .NFB formats.
- **Mesh deflection:** Parameter (between 0 and 2) that determines the size of the tessellation for parametric geometries (STEP, IGES). The lower is the parameter the higher is the number of polygons generated to tessellate the geometry. The higher is the number, the lower is the number of polygons.
- **Show wires:** This option enables/disables the visualization of the wires of the geometries.

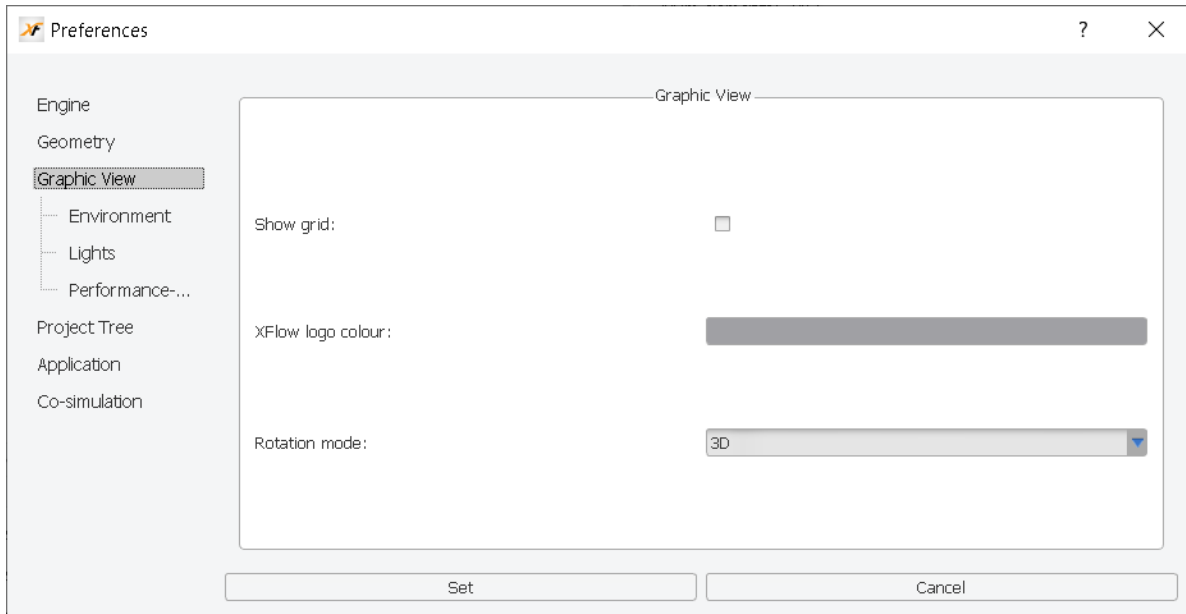


Example of visualisation without wires (left) and with wires (right).

- **Fit wind tunnel to geometries automatically:** If enabled, the [Virtual Wind Tunnel](#) will automatically resize if a larger geometry is imported in order to fit the geometry bounding box.
- **Zoom automatically on imported geometries:** If enabled, the [Graphic View](#) will automatically zoom on the imported geometries.

3.5.3 Graphic View

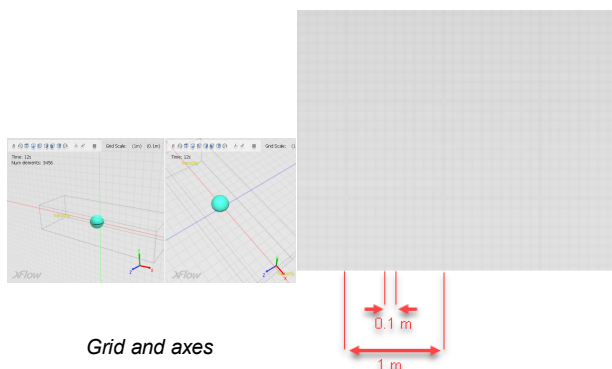
Main menu > Options > Preferences > Graphic View



Preferences window: Graphic View section

The Graphic View section of the *Preferences* window shows the following options to improve the visualization in the [Graphic View](#) window:

- **Rotation mode:** Y-axis mode keeps this axis fixed when rotating the camera, while 3D mode does not have a predefined rotation axis.
- **XFlow logo colour:** It is possible to set XFlow log with any different colour so that it has the desired contrast with the graphic view background.
- **Show grid:** The grid serves as a visual reference, it may be helpful for doing measurements on the screen. The three axes are painted in the [Graphic View](#) by using lines colored by the corresponding axis color. The grid size is displayed on the left-hand top corner of the *Graphic View*. The first number refers to the size of the big scale, while the second number refers to the dimension of the small scale, as pointed out in the figure below:



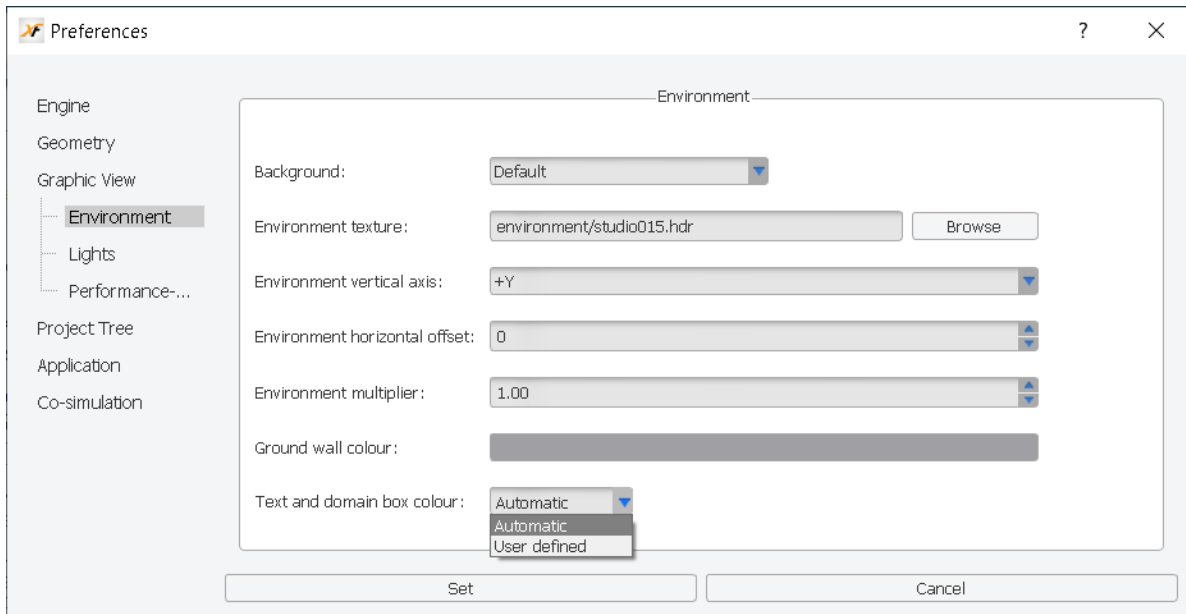
Grid and axes

Grid size

3.5 Preferences

3.5.3.1 Environment

Main menu > Options > Preferences > Graphic View > Environment

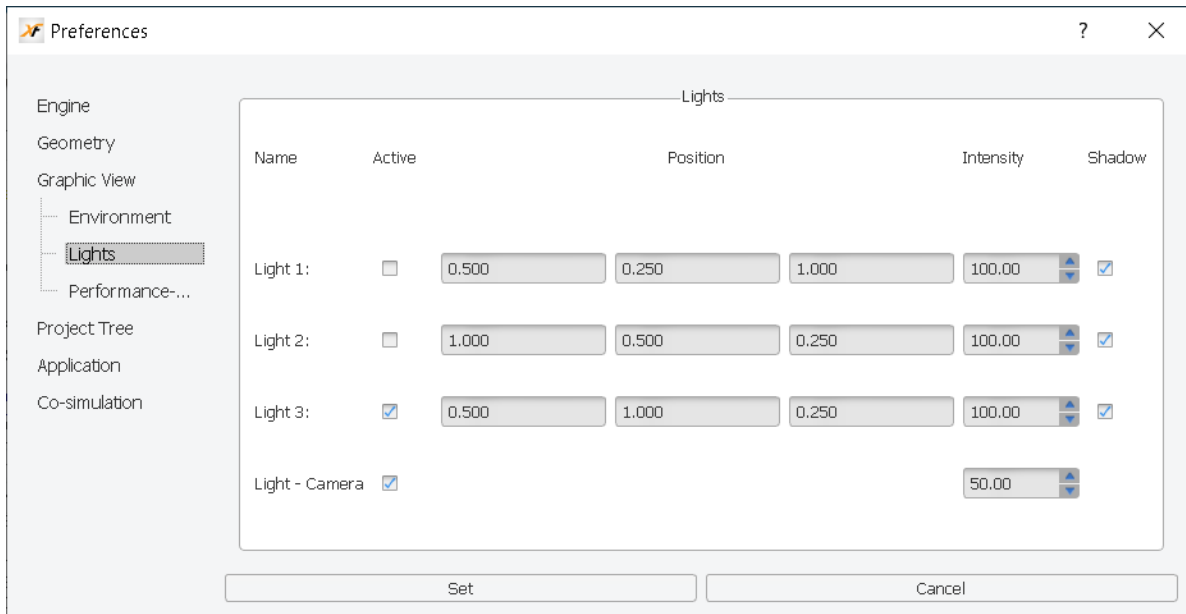


Preferences window: Environment (Graphic View) section

- **Background:** It is possible to set the background to one color, two colors or a texture
- **Environment texture:** To browse for a file containing the texture to be shown in the *Graphic View* background
- **Environment vertical axis:** To allow for the selection of which axis (and sign) to use as the upward direction for the imported HDR environments.
- **Environment horizontal offset:** To rotate the background texture in the horizontal plane. Value range: [0,180] degrees.
- **Environment multiplier:** To multiply the brightness of the environment texture.
- **Ground wall colour:** To customize the ground wall color.
- **Text and domain box colour:** If Automatic is set, the text on the graphic view (Simulation Time, color bar, etc) is adjusted for easier visualization based on the background color. Alternatively, User defined option allows to set a specific color.

3.5.3.2 Lights

Main menu > Options > Preferences > Graphic View > Environment



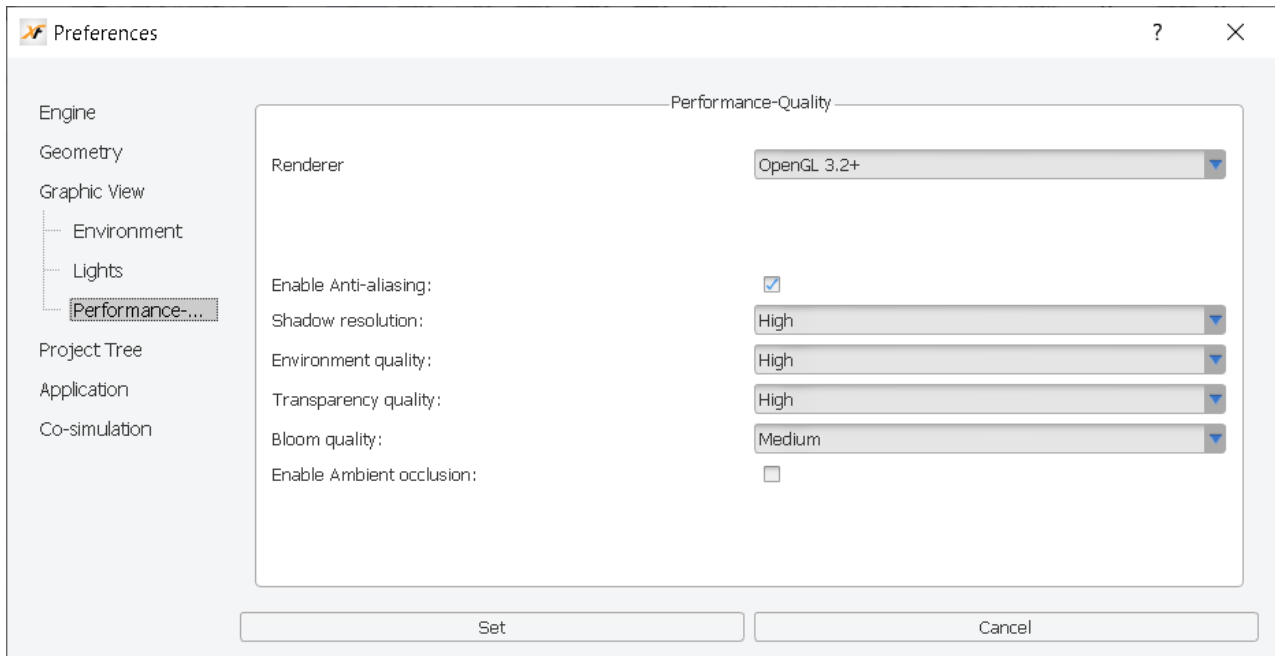
Preferences window: Lights (Graphic View) section

This controls the lights that illuminate the geometry in the *Graphic View*. Each light is defined by a point in space and is directed towards the origin (0,0,0). On this panel one can decide to turn them on/off or change their position.

3.5 Preferences

3.5.3.3 Performance-Quality

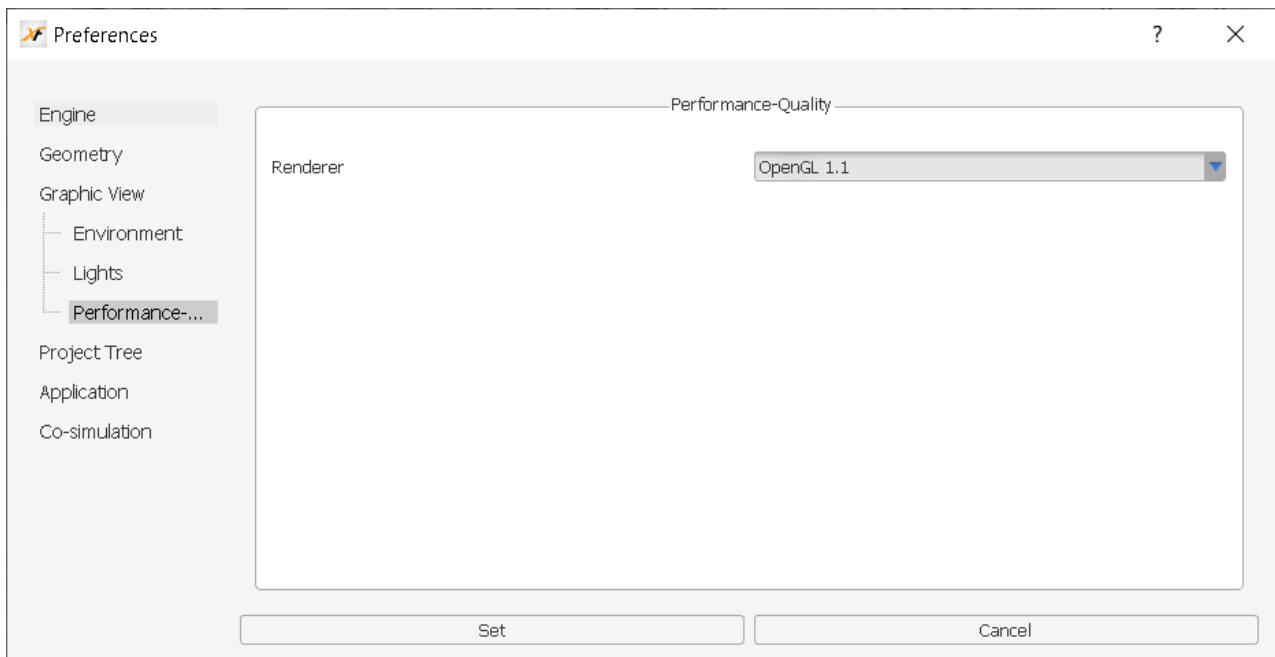
Main menu > Options > Preferences > Graphic View > Environment



Preferences window: Performance Quality (Graphic View) section

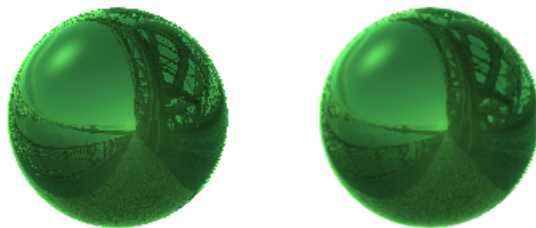
This section affects how the *Graphic View* will behave for real-time visualization and balance performance with quality. The new visualization techniques of XFlow allow for greater realism and sense of depth, so it is easier to inspect and see errors in the geometry. However, these options have a definite impact on the performance and responsiveness of the *Graphic View*, so the user might want to consider turning some of them down (or even off).

- **Renderer:** This option allows the user to switch from the current renderer that runs under OpenGL 3.2+ to a Legacy renderer that runs under OpenGL 1.1. The latter is advised for user that have performance issues using XFlow 2017 new renderer. Although the performance will be improved, allowing XFlow GUI to run with most of graphics cards, it will also decrease the quality of the visualizations in XFlow GUI.



Preferences window: Graphic View section with OpenGL 1.1

- **Anti-aliasing quality:** This determines how much post-processing will be applied to remove jagged edges from the *Graphic View*. The higher the setting, the smoother the geometry contours will be. The cutting-edge algorithm used in XFlow has a very little impact on performance, so it can be set to high even with mid-range graphic cards.

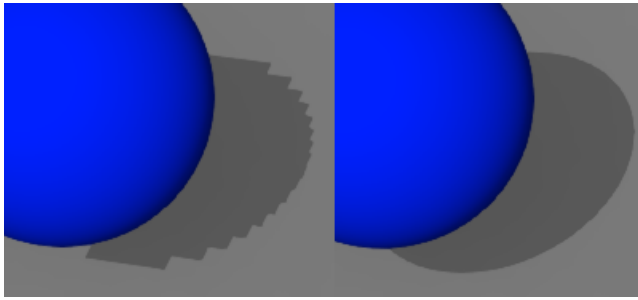


Example of visualisation without anti-aliasing (left) and with high anti-aliasing (right).

- **Shadow resolution:** Shadows can greatly enhance the perception of depth and the spatial relationship between shapes. If this option is set to **Low**, it requires less graphics processing units (GPU) memory and the shadows are computed faster, but depending on the scene it can lead to shadows with stair-case border artifacts. On the other hand, the highest setting produces more defined shadows, but they take longer to calculate and they require more GPU memory.

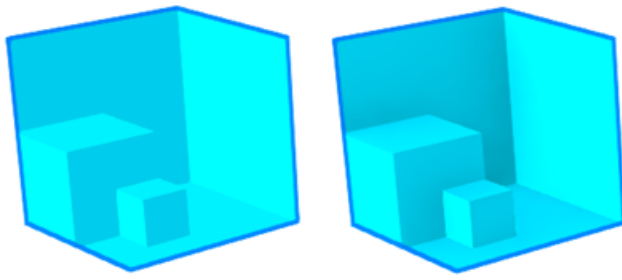
Note that shadow generated by every [light](#) will be recomputed either when the lights or the geometries in the scene are moved, or when a new frame is loaded.

Finally, it is worth pointing out though that shadow computations can have a great impact in performance during playback. So it might be interesting to turn them off if the fastest playback possible is desired. It might also be a good idea to turn the off if your graphics card has little memory.



Example of a shadow with low resolution (left) and with high resolution (right).

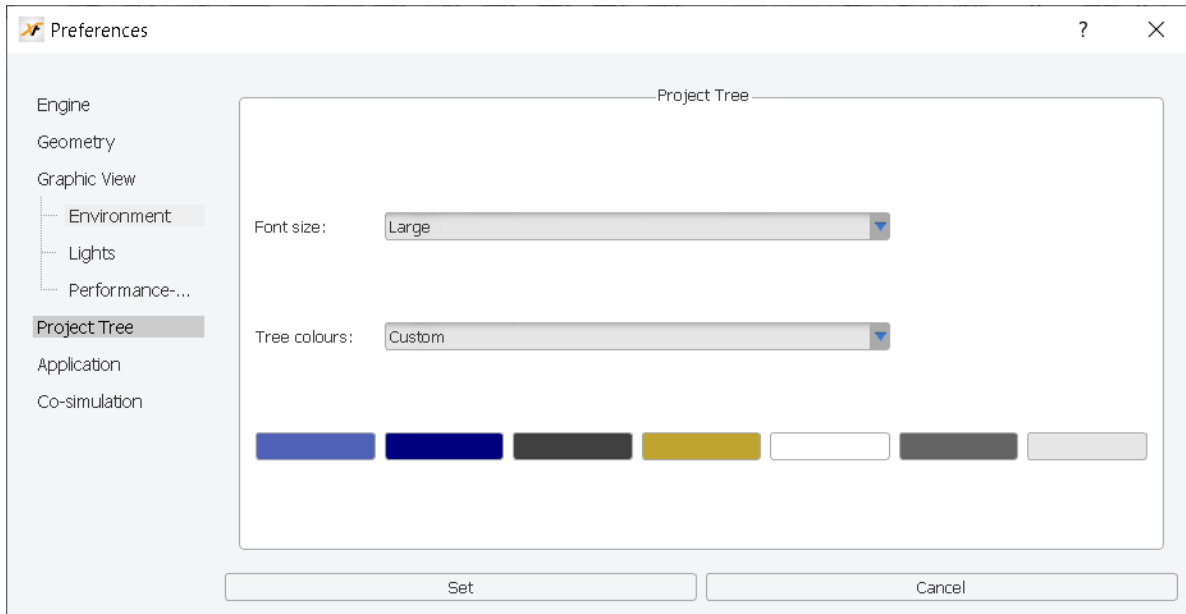
- **Environment quality:** Controls how sharp reflections can look, and how well visualization materials transition from roughness 0 to 1. Higher requires more memory and higher viewport loading times.
- **Transparency quality:** High quality makes the transition from opaque to transparent smoother. It also makes easier to differentiate the front and the back of transparent geometries. It requires more GPU computation, however.
- **Bloom quality:** Bloom is the effect by which bright points appear to have a halo. XFlow simulates this effect to increase perceive brightness, but it comes with a performance cost, which can be regulated from [here](#).
- **Enable Ambient occlusion:** Simulates the effect by which corners are darkened (light comes from fewer directions). It requires more memory and computation time.



Example of visualisation without ambient occlusion (left) and with ambient occlusion (right).

3.5.4 Project Tree

Main menu > Options > Preferences > Project Tree



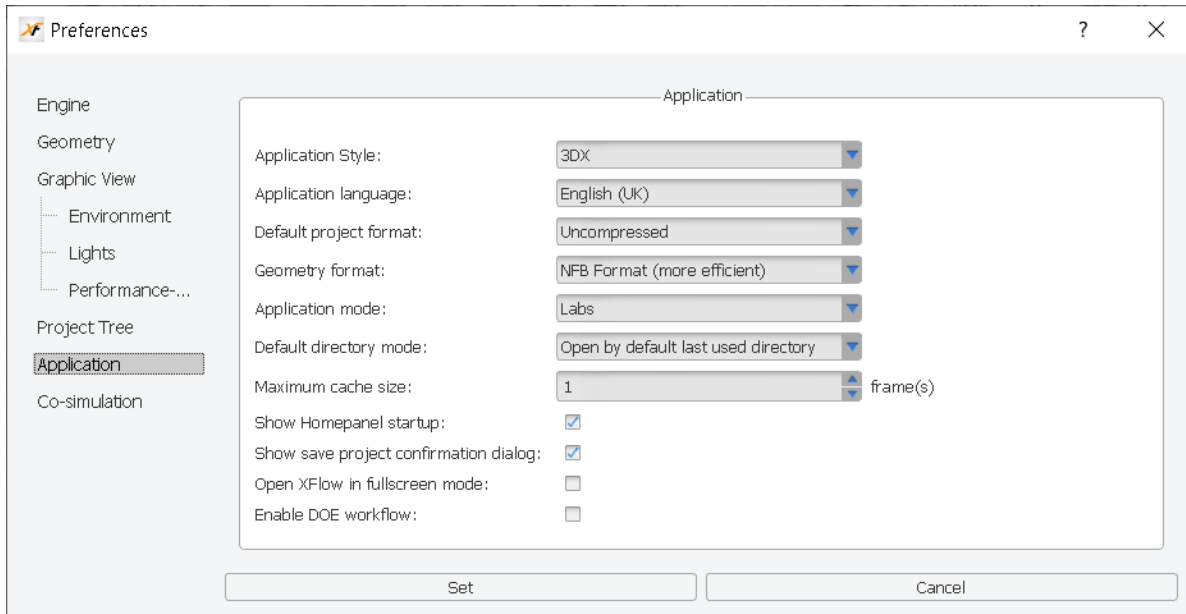
Preferences window: Project Tree section

In the *Project Tree* section of the *Preferences* window, the appearance of the [Project Tree](#) can be customized:

- **Font Size:** Small, Normal, Large and XLarge.
- **Tree colors:** There are six options to customize the colors of the *Project Tree*.

3.5.5 Application

Main menu > Options > Preferences > Application



Preferences window: Application section

In the Application section of the *Preferences* window, the general appearance of XFlow can be customized:

- **Application Style:** XFlow GUI can be executed in two styles (3DX or dark). The default style is 3DX (background and menus in white color).
- **Application language:** defines the language of application. Language can be set in: English, French, Spanish, German, and Chinese.
- **Default project format:** format can be selected between uncompressed (.XFP) or compressed (.XFZ). See [XFlow files formats](#) for more details.
- **Geometry format:** default format used for mesh formats of geometries. Either original format (STL) or native XFlow format (NFB) can be used. NFB format is recommended for better performance.
- **Application mode:** XFlow can be run in *Normal* mode, [Expert mode](#), or [Labs mode](#).
 The *Normal* mode is the basic mode for which the user has the most basic options and parameters available.
 The *Expert* mode is similar to *Normal* mode but additionally include a few more advanced parameters and features, it is recommended for advanced user.
 The *Labs* mode has the same options available than the *Expert* mode but includes additional features that are under development (see [Labs mode](#)).
- **Default directory mode:** To set up the default mode to browse files. There are two possibilities: browse files from the project's directory or from the last directory used.
- **Check for XFlow updates:** XFlow will notify if a latest version is available to download. This option can be disabled in case you do not want XFlow to check for updates.
- **Maximum cache size:** XFlow can store an arbitrary number of simulation frames in memory. Default and recommended value is 1 frame, however several frames can be cached in memory in order to speed-up repetitive frame loading (for instance, for [loop in tracers](#), etc.).
- **Show Homepanel startup:** When deactivated, the [Project Manager](#) window will not appear when starting XFlow.
- **Show save project confirmation dialog:** A window will pop-up and request for confirmation to save the project if a simulation is run and the project has not been saved. The project will automatically be saved in case this option is disabled.

- **Open XFlow in fullscreen mode:** opens the interface in full screen mode automatically if enabled.
- **Enable DOE workflow:** all the inputs and output fields show a DOE icon  in order to make them available for Design Of Experiment in the Process Composer of the 3DEXperience platform. For more information, refer to the 3DEXperience platform documentation.

4 Geometry

This chapter covers how to manage the geometry objects needed for the simulation, covering the following topics:

- How to [create](#) a geometry
- How to [import/export](#) a geometry
- How to [select](#) a geometry
- [visualization](#) modes
- How to get [information](#) of the geometry
- How to [modify](#) the geometry
- How to split a surface into several [boundary patches](#).
- How to [delete](#) a geometry

Geometry objects are listed in **Project Tree > Geometry (+Info)**, where their thermophysical properties, behaviour (e.g. motion) and boundary conditions can be defined.

4.1 Create entity

XFlow provides the utilities to create basic geometries, as detailed below. Once a geometry is created it is shown in the *Graphic View* and appears as a **Shape** in the *Project Tree*.

Create vertex

Main menu > Geometry > Create Object > Create vertex, 

A point in space is created by defining its spatial coordinates (x,y,z).

Create reference frame

Main menu > Geometry > Create Object > Create vertex, 

A [reference frame](#) is created by defining the spatial coordinates of its center (x,y,z) and its orientation in the global axes.

Create line

Main menu > Geometry > Create Object > Create line, 

A line can be created from two points (their coordinates have to be entered in a dialog box) or from existing points (which have to be previously selected, see [Geometry selection](#)).

Create curve

Main menu > Geometry > Create Object > Create curve, 

A curved line can be created from three or more points previously defined. Hence:

- create the points that define the curve, 
- select the points, 
- click the "Create curve" icon, 

Create surface

Main menu > Geometry > Create Object > Create surface, 

4.1 Create entity

Planes can be created from:

1. Its vertices:

- (a) define the vertices of the surface, ,
- (b) select the vertices sequentially (in clockwise or counterclockwise order), ,
- (c) click the "Create plane" icon, .

2. A closed set of lines:

- (a) create a set of lines limiting a closed area, ,
- (b) select these lines, ,
- (c) click the "Create surface" icon, .

3. A closed set of wires:

- (a) select a closed set of wires, ,
- (b) click the "Create surface" icon, .

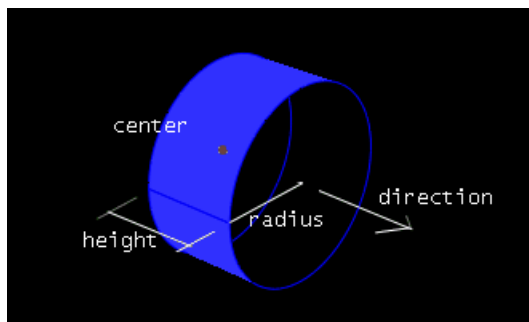
 **Please note:** Depending on the orientation of the surface normal, one may need to move the camera in order to see the surface (see [Back-face culling](#)).

Create Cylinder

Main menu > Geometry > Create object > Create cylinder, 

A cylinder is created by specifying the following parameters:

- **Center** of the circle that describes the base
- **Direction** of extrusion
- **Radius** of the circular section (base or top)
- **Height** of the cylinder
- **Angle** of the cylindrical sector (between 0 and 360)
- **Num azimuthal segments** which is the number of elements in the azimuthal direction used to build the cylinder
- **Num radial segments** which is the number of elements in the radial direction used to build the cylinder
- **Num longit. segments** which is the number of elements in the longitudinal direction used to build the cylinder

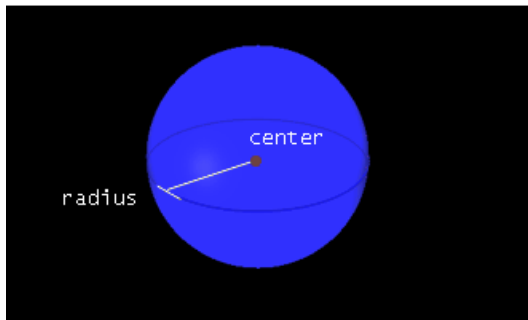


Cylinder parameters

Create Sphere

Main menu > Geometry > Create object > Create sphere, 

A sphere is created by specifying its **center** and **radius**.



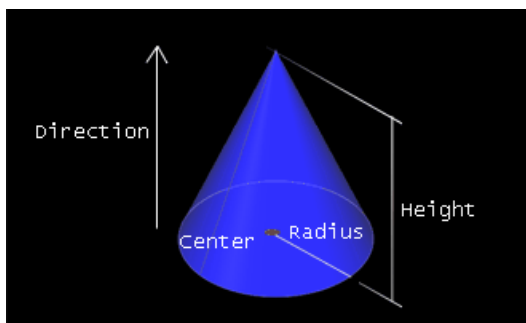
Sphere parameters

Create Cone

Main menu > Geometry > Create object > Create cone, 

A cone is created by specifying four parameters:

- **Center** of the circle that describes the base
- **Direction** of cone height
- **Radius** of the circular base
- **Height** of the cone



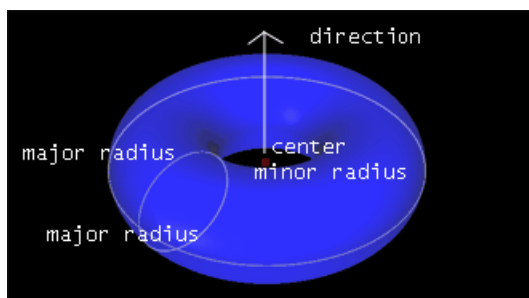
Cone parameters

Create Torus

Main menu > Geometry > Create object > Create torus, 

A torus is created by specifying four parameters:

- **Center** of the radius
- **Direction** normal to the torus plane
- **Major radius**, external radius
- **Minor radius**, internal radius



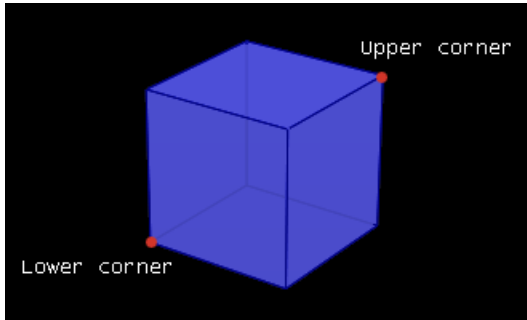
Torus parameters

4.1 Create entity

Create Box

Main menu > Geometry > Create object > Create box, 

A box is created by specifying the coordinates of the upper and lower corners.



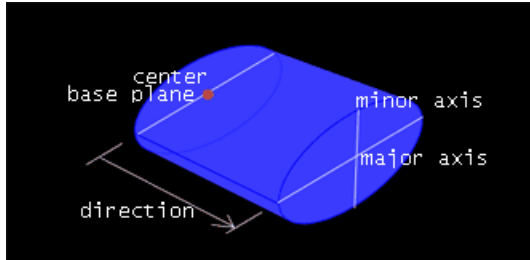
Box definition sketch

Create Prism

Main menu > Geometry > Create object > Create prism, 

To create a prism with elliptical basis, according to the following parameters:

- **Center** of the radius
- **Base plane** normal
- **Direction** of the prism height
- **Minor axis** of the ellipse
- **Major axis** of the ellipse



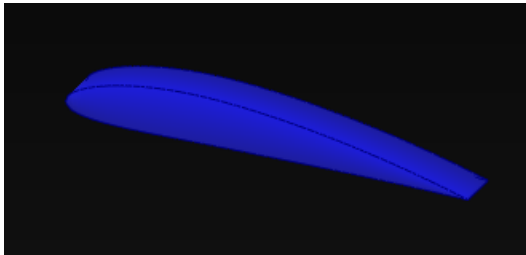
Prism parameters

Create NACA (4 digit)

Main menu > Geometry > Create object > Create naca (4 digit), 

This option allows the user to create an airfoil defined by:

- **Digits** first, second and third and fourth digits of the NACA airfoil
- **Chord** length
- **Width** extrusion size in Z-direction
- **NumPnts** number of points to represent the profile (bunched at leading and trailing edge)



NACA (4 digits)

Info: All surfaces are automatically tessellated when created.

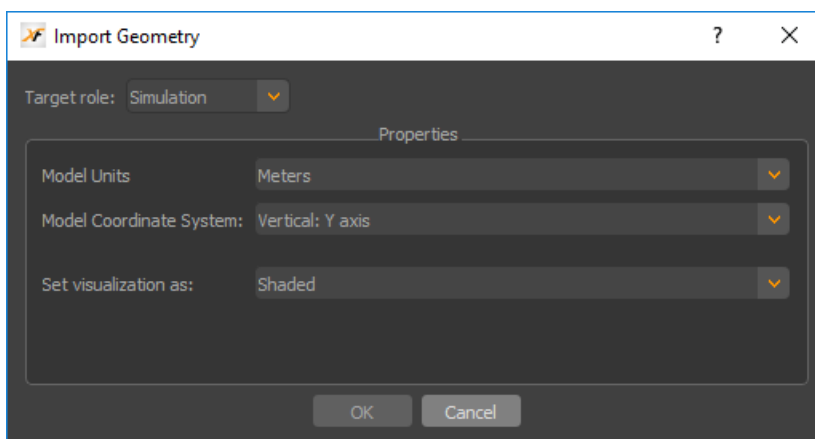
4.2 Import and export geometry

Import geometry

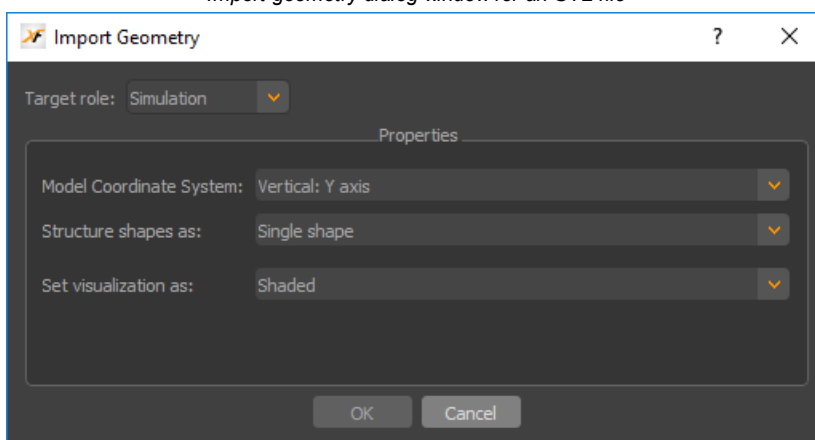
Main menu > Geometry > Import a new geometry,  or **<Ctrl + i>**

XFlow can import a geometry from CAD, 3D mesh models and NURBS models.

This dialog window is shown when importing new geometries:

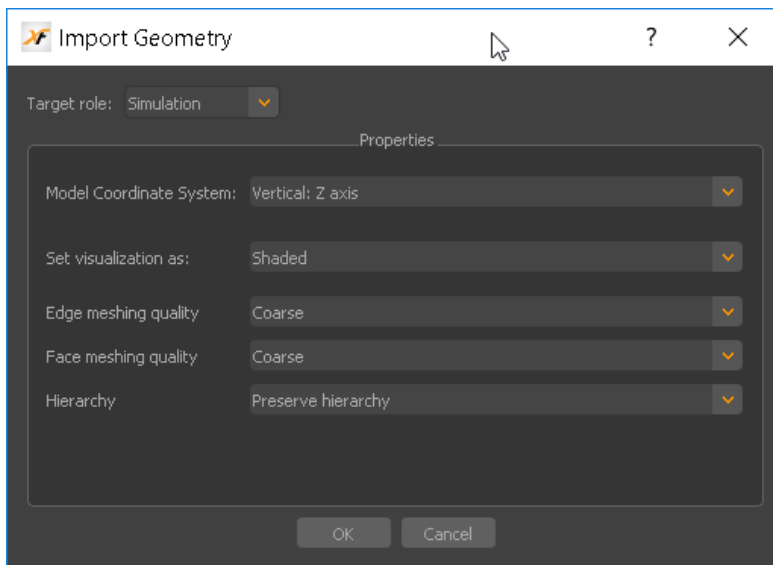


Import geometry dialog window for an STL file



Import geometry dialog window for a parametric geometry file

4.2 Import and export geometry



Import geometry dialog window for CATIA geometry file

- **Target role:** geometries can be imported in two categories: Simulation or Post-processing. Simulation geometries will be imported in the [Geometry](#) tab of the project tree and will participate to the simulation, while the [Post-processing](#) geometries will be imported in the Post-processing tab of the project tree and will be used for post-processing purpose only.
- **Model units:** the user can select the Units system for the imported geometries. This option is only asked for the STL geometries. The other formats(STP, IGES and NFB) already contain the unit in the geometry file.
- **Structure shapes as:** the geometry file can be imported in the [Geometry tab](#) as only one shape or several shapes. This is available for parametric geometries only (STEP, IGES) and depends on the [parametric hierarchy](#).
- **Model coordinate system:** the user can define which is the vertical [axis](#) of the imported geometries.
- **Import as child of:** the imported geometry can defined as a [child](#) of one of the existing geometries.
- **Set visualization as:** to specify directly the [visualization mode](#) of the imported geometry. This is particularly useful to disable the visualization of the heavy geometries when importing them.

The compatible geometries formats in XFlow are:

.STL	STL files only describe the surface mesh of a three dimensional object, without any representation of color, texture or other common CAD model attributes. The STL format specifies both ASCII and binary representations. Binary files are more common, since they are more compact.
.STEP / .STP	STandard for the Exchange of Product model data defines data models for boundary representations.
.IGES / .IGS	Initial Graphics Exchange Specification .
.BREP / .RLE	In Boundary REPresentation, a solid is represented as a collection of connected surface elements, the boundary between solid and non-solid.
.NFB	The native binary file format of geometry in XFlow. This is a mesh format, as for STL.
.CSFDB	Open CASCADE 3D Model File
.OFF	Object File Format.
.BDF	Nastran Model (Only available in Expert mode)
.CATPart / .CATProduct	CATIA part and CATIA product formats (Only available in Expert mode)
.sldptr / .sldasm / .Isddrw	SolidWorks part, SolidWorks assembly and SolidWorks vector formats (Only available in Expert mode)

 **Please note:** When saving an XFlow project file containing an STL geometry, XFlow will automatically convert it to the NFB format which is the XFlow native mesh format. This can be changed in the Preferences > Application > Geometries format, switching to Original format instead of NFB. NFB format is recommended for performance.

 **Tip:** Recommended formats are STEP and STL. STEP is parametric and therefore flexible for surfaces definition, but must be tessellated by XFlow when importing. STL is a mesh format and therefore imported exactly as per the STL file, but surface detection will be based on surface angles and may be altered compared to STEP geometries.

 **Tip:** Check the orientation of the geometry. Remember that in XFlow Y-direction represents the height and Z-direction the width.

 **Please note:** When importing a geometry into XFlow external simulation, if the geometry bounding box is not contained in the fluid domain, this will adapt its dimensions in order to contain the geometry.

 **Please note:** When importing a .CATProduct the user can chose between keep the hierarchy or import each geometry separately.

After specifying the model units, the imported geometry is shown in the *Graphic View* and added as a **Shape** into the **Geometry** section of the *Project Tree*.

Export geometry

Main menu > Geometry > Export Geometry,  or <Ctrl + E>

4.2 Import and export geometry

The geometry can be exported in the same file formats as when importing: STL, STEP, IGES, BREP and NFF.

The geometry object to export has to be selected prior to executing the command.

4.3 Select geometry

Different selection filters exist to select the geometries in XFlow. They can be accessible via the [main menu](#) and [toolbar](#):

- **View only:** selects an object, but do not highlight them when passing the cursor over the geometry. Furthermore, this selection filter does not allow the use of [Gizmos](#). An object is defined as a part of 3D space bounded by shells.
- **Object filter:** selects an object, highlight them when passing the cursor over the geometry. It allows to use [Gizmos](#). An object is defined as a part of 3D space bounded by shells.
- **Shell filter:** selects the shell of an object. A shell is set of faces connected by some of the edges of their wire boundaries. A shell can be open or closed. Shells are only available for parametric geometries (STEP, IGES).
- **Face filter:** selects the face of an object. A face is defined as part of a plane (in 2D geometry) or a surface (in 3D geometry) bounded by a closed wire. Its geometry is constrained (trimmed) by contours.
- **Wire filter:** selects the wire of an object. A wire is a sequence of edges connected by their vertices. It can be open or closed depending on whether the edges are linked or not.
- **Edge filter:** selects the edge of an object. An edge is a single dimensional shape corresponding to a curve, and bound by a vertex at each extremity.
- **Vertex filter:** selects the vertex of an object. A vertex is a zero-dimensional shape corresponding to a point in geometry.

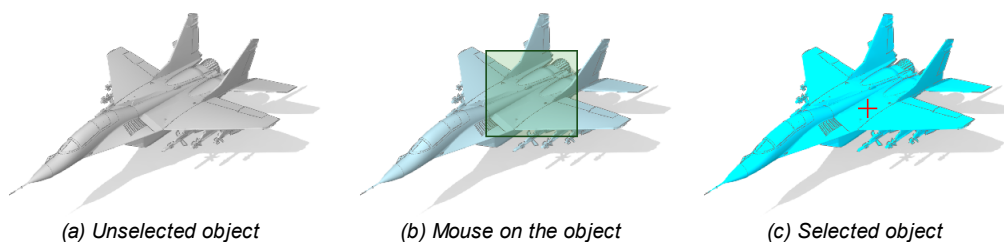
The objects, shells and faces can be selected in both Graphic View and Project Tree:

- [Selection in the Graphic View](#)
- [Selection in the Project Tree](#)

4.3.1 Selection in Graphic View

In the *Graphic View*, geometry objects, shells and faces can be selected either by direct clicking or by rectangular selection (click and drag).

When the mouse is passed over a geometry, it is highlighted in a soft blue to indicate the object that can be selected. The object is eventually highlighted in cyan when it is selected.

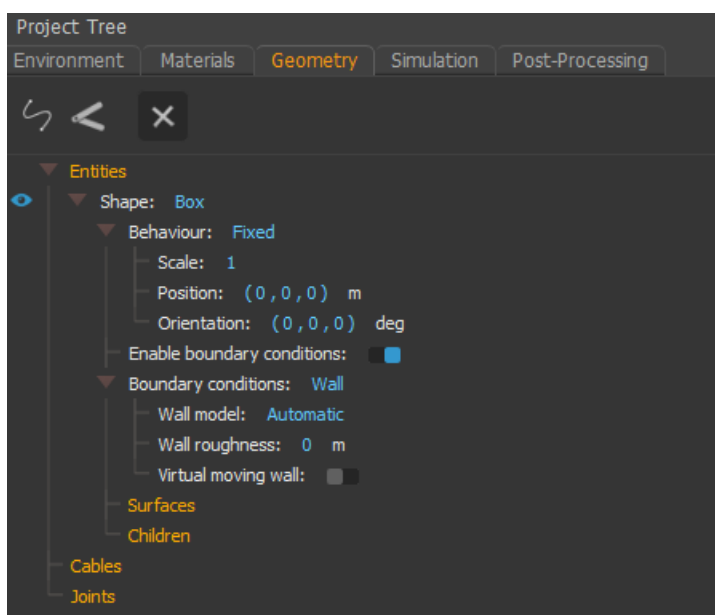


To select additional objects, hold the **Ctrl** key while selecting the objects.
 To remove objects from the collection of selected objects, hold the **Ctrl** key and select the objects again.
 To cancel the selection, just click on any point of the graphic viewer which is not an object or press the **Esc** key.

4.3.2 Selection in Project Tree

Project tree > Geometry > Geometries > Shape

An object can be selected in the *Project Tree* by clicking on its corresponding **Shape**. If selected, it is highlighted.

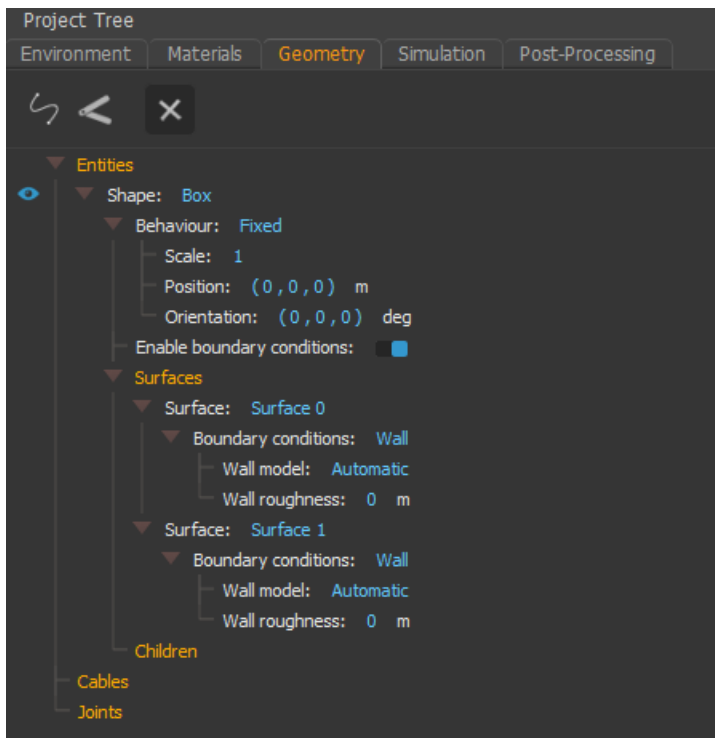


Object selection: *Project Tree > Geometry > Geometries > Shape*

Project tree > Geometry > Geometries > Surface

Shells and faces can also be selected in the *Project Tree* when [boundary conditions are applied](#), by clicking on the corresponding **Surface**. If selected, it is highlighted.

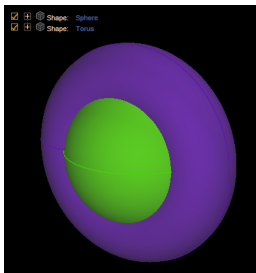
4.3 Select geometry



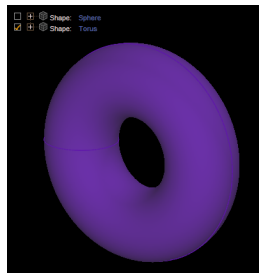
Face selection: Project Tree > Geometry > Geometries > Shape

4.4 Visualisation

A geometry can be shown/hidden activating/deactivating the combo box shown on the left side of the Geometry tree, as shown in the figure below:



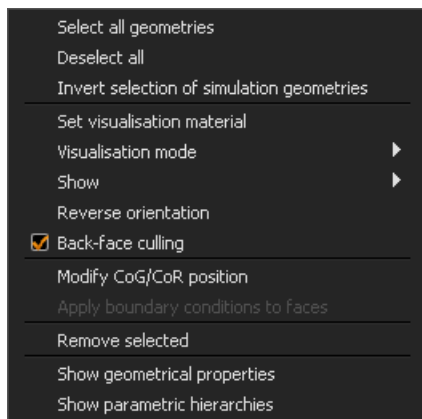
Both sphere and torus are shown



Sphere is hidden and Torus is shown

Once a geometry is shown, XFlow provides different modes and materials to visualize it. To see these options:

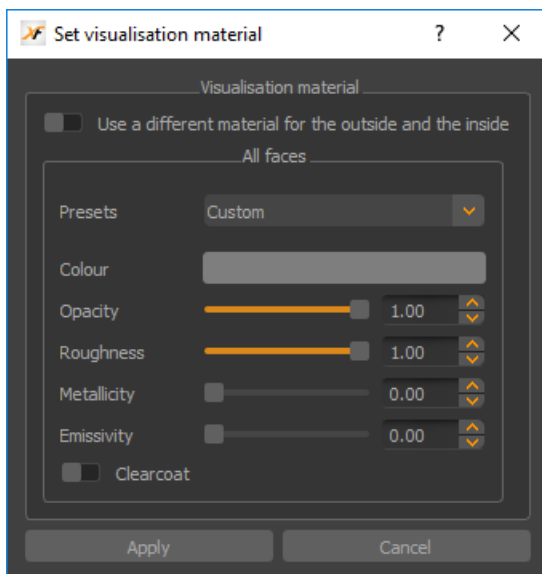
- Select the geometry
 - Click on the *Graphic View* window with the **right button**
 - The [Graphic View Menu](#) appears containing (among others) the following options (see picture below):
1. [Set visualization material](#)
 2. [Visualization mode](#)
 3. [Back-face culling](#)



Graphic View Menu

4.4.1 Visualisation material

This option allows the user to set a material type and a color to the selected geometry. The available parameters are illustrated in the snapshot below.



Set visualisation material parameters.

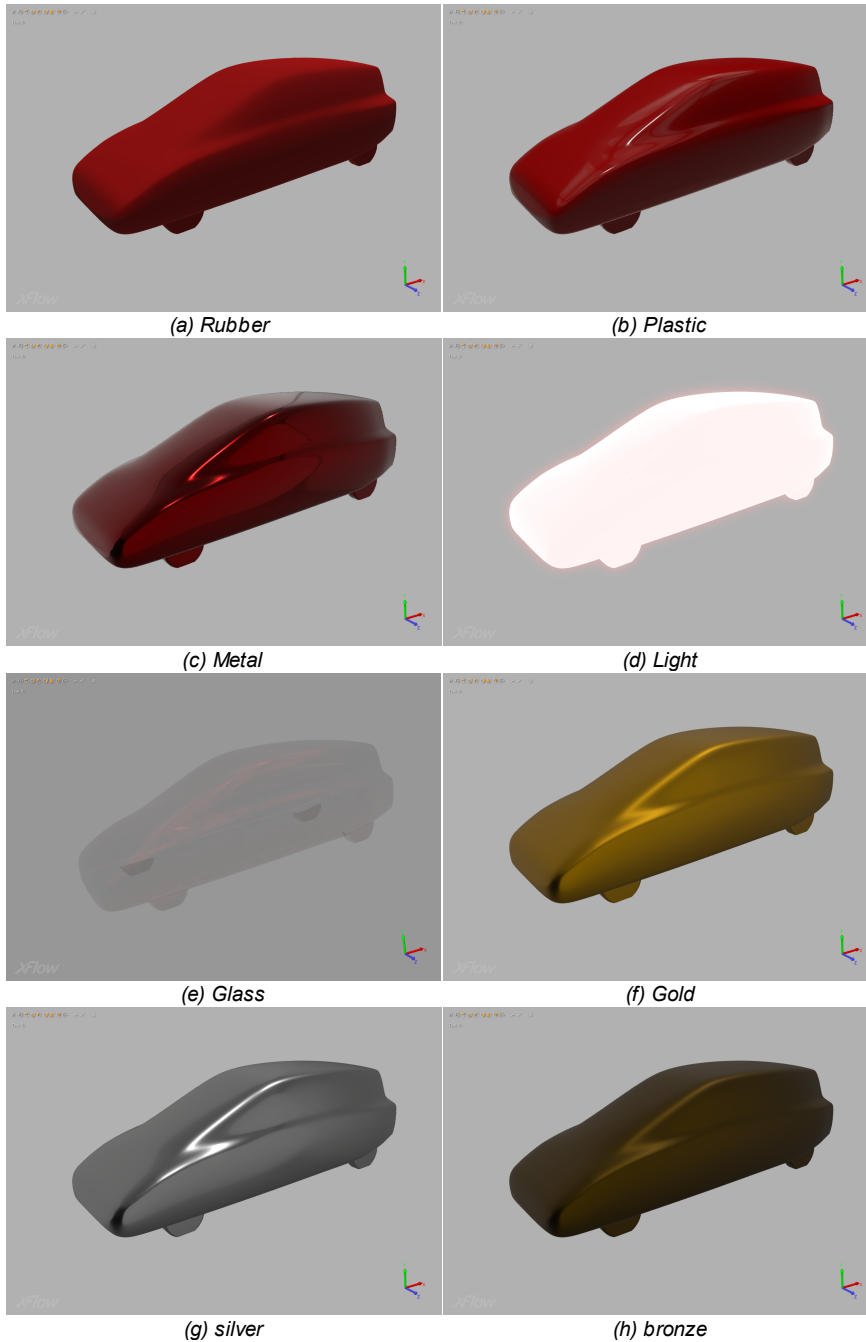
The material types, shown in the figure below, are:

- **Custom:** The user can control all the available parameters.
- **Rubber:** Preset settings to visualize rubber-like materials (maximum value of **Roughness**).
- **Plastic:** Preset settings to visualize plastic-like materials (low value of **Roughness**).
- **Metal:** Preset settings to visualize metallic materials. Additional **Metallicity** parameter available to the user.
- **Light:** Material appropriate for faking light sources since it is not affected by shadows (the same way a lightbulb is not darker when in shadows). It does not, however, illuminate the rest of the elements of the scene. Additional **Emissivity** parameter available to the user.
- **Glass:** Preset settings to visualize glass materials.
- **Gold:** Preset settings to visualize gold.
- **Silver:** Preset settings to visualize silver.

4.4 Visualisation

- **Bronze:** Preset settings to visualize bronze.
- **Car paint metallic blue:** Preset settings to visualize blue car paint.
- **Car paint red:** Preset settings to visualize red car paint.

Colors can be selected either by choosing from a spectrum or by defining the color numerically, based on the HSB (hue, saturation, brightness) or RGB (red, green, blue) color models.



visualization materials

Additionally, the user can manually modify parameters related to the geometry material visualization, which

can allow a customized high quality render.:

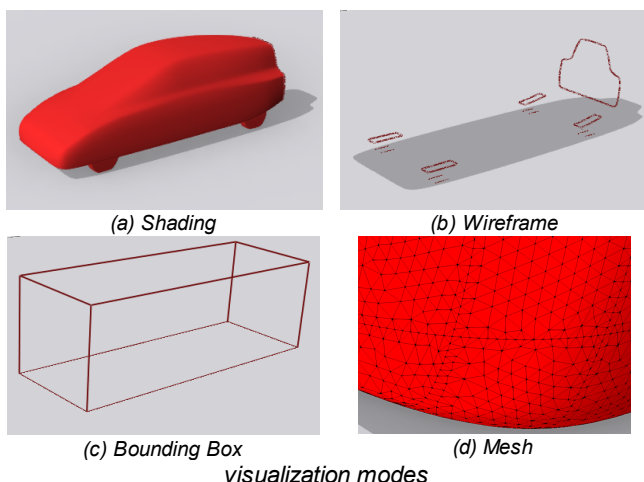
These parameters are:

- **Opacity:** The amount of light transmitted through the material. The higher the opacity the less transparent the object is.
- **Roughness:** In real life, the roughness defines how irregular the microscopic structure of a material is. Therefore, the higher the roughness, the more blurred the reflections are.
- **Metallicity:** Increasing the metallicity of the material reduces the diffusion of the lighting and results in a more tinted reflection.
- **Emissivity:** Establishes the amount of light emitted by the material. The higher the emissivity, the more light is emitted. Please notice that the emitted light does not affect other materials (it does not create shadows).
- **Clearcoat:** Adds a reflective layer on top of the material mimicking the coating effect of painted surfaces. This allows to have sharper reflections..
 - If the opacity is smaller than 1.00, no clearcoat can be applied.
- **Use a different material for the outside and inside:** This toggle allows for using a different material for the front faces and the back faces. It is useful for isosurfaces in mixing cases and might be useful for internal analysis (instead of using backface culling we can apply a glass material to the backfaces).

4.4.2 Visualisation mode

XFlow offers four modes of geometry visualization, as shown in the figures below:

- **Shading:** Surfaces and solids are shown with shading.
- **Wireframe:** Only the edges of the surfaces and solids are shown.
- **Bounding box:** This option shows the edges (wireframe) of the smallest box within which all the geometry points lie. The Bounding Box representation hides the model and so uses fewer computer resources.
- **Mesh:** This option shows the tessellation of the geometry. Surfaces are automatically meshed when a geometry model is created or imported in parametric format; tessellation size can be set up in [Preferences](#). Triangulated formats, such as .stl, keep their original tessellation.



To select/change the visualization mode:

- Select the geometry
- Click on the *Graphic View* window with the **right button** to show a **drop-down menu**, containing a **visualization Mode** option

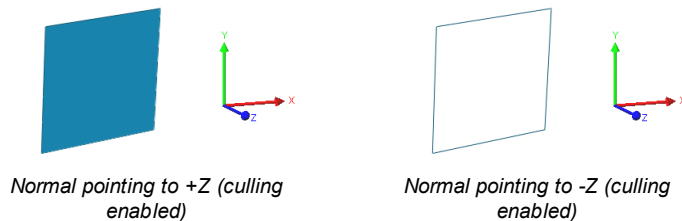
4.4 Visualisation

- Choose among:
 - > visualization Mode > Shading
 - > visualization Mode > Wireframe
 - > visualization Mode > Bounding Box
 - > visualization Mode > Mesh

4.4.3 Back-face culling

The back-face culling determines whether a geometry object in the *Graphic View* is visible or not from the camera position.

If back-face culling is enabled, surfaces which [normal](#) points to the camera (user) are shown as opaque bodies, while those which normal points away from the camera are shown as transparent bodies (only the wireframe is shown).



When back-face culling is disabled, both sides of the surface will be opaque.

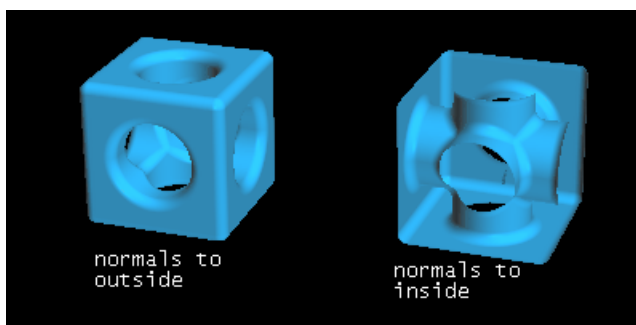
Tip: It is better to enable the culling since this gives useful information about the normal to the surface.

Please note: When culling is enabled only surfaces oriented towards the camera are shown.

4.4.3.1 Surface normals

In XFlow, it is essential that all the geometry normals point to the fluid, because the fluid domain is automatically built based on the surfaces normals orientation. The user can see whether the surface normals point outside or inside the object with the help of the [back-face culling](#). Alternatively, the **show > normals** option can be used ([+info](#)) to plot vectors representing the normal direction.

In the figure below it is shown the difference between the normals pointing outwards (fluid outside the object, i.e. external flow) or inwards (fluid inside the object, i.e. internal flow).



Back-face culling: visualisation according to the normals direction

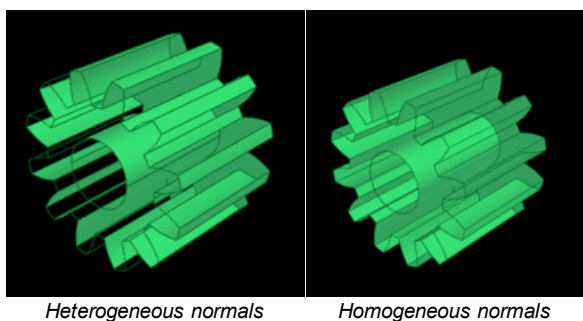
XFlow provides two options to modify the orientation of the normals:

- [Reorientate normals](#)
- [Reverse orientation](#)

4.4.3.1.1 Reorientate normals

(Select the geometry +) **Main menu > Geometry > Reorientate normals,** 

This command allows the user to homogenize the normals of a geometrical object.

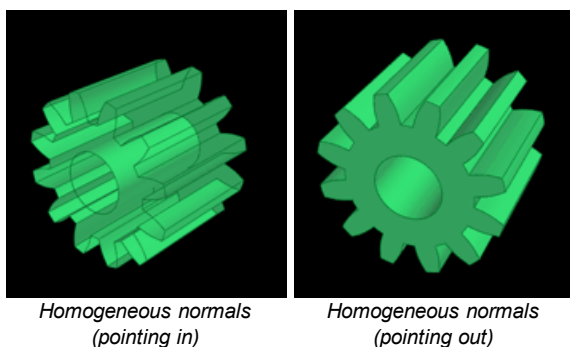


 **Please note:** This operation is included in [Complete healing](#) (preferred).

4.4.3.1.2 Reverse orientation

(Select the geometry +) **Graphic View Menu > Reverse orientation**

This command changes the direction of the normals of a surface, so that they point in the opposite direction. If [back-face culling](#) is enabled, the opaque side will change to the other side.



4.5 Geometry Information

In this section, the XFlow options regarding the geometry properties are described. These are:

- [Show Local axes](#)
- [Geometrical properties](#)
- [Geometry dimensions](#)
- [Check holes](#)

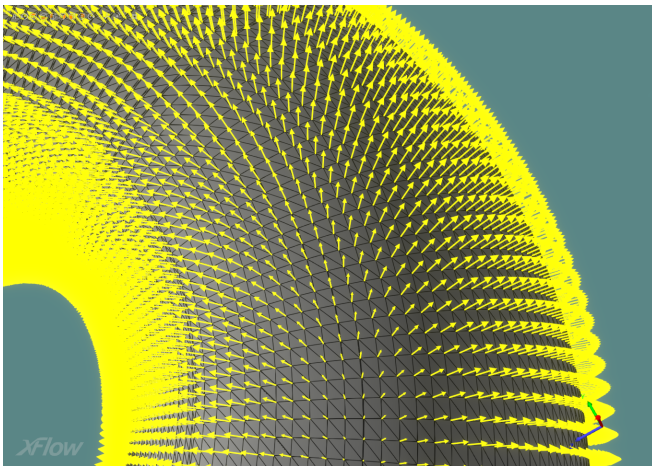
4.5 Geometry Information

These options are available either from the **Main menu** or from the [Graphic View menu](#).

4.5.1 Normals

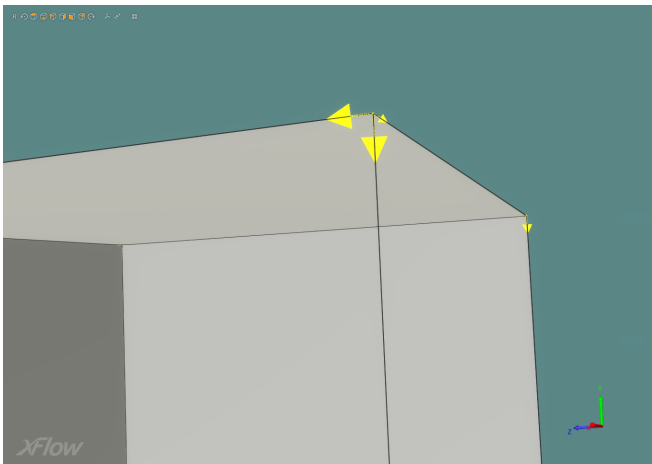
(Select the geometry +) [Graphic View Menu](#) > **Show** > **Normals**

The show normals option allows to visualize the orientation of the surface triangle tessellation by means of vectors. This representation helps in detecting the orientation of surfaces in a shape object and, if required, inconsistent orientation can be automatically fixed using the [Reorientate normals](#) option.



Example of normal vectors showing triangles pointing outwards.

A vector is plot per each geometry vertex, which can clutter the graphic view.

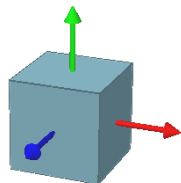


Example of normal vectors showing triangles pointing inwards. The back-face culling option is activated as well.

4.5.2 Local axes

(Select the geometry +) [Graphic View Menu](#) > **Show > Local axes**

In moving geometry objects, prescribed position laws, angular laws and external forces and moments are assumed to be defined with respect to the local axes, whose origin is by default the **centre of gravity (CoG)**, which is automatically calculated assuming that the object density is constant in space.



Local axes

By default the **centre of rotation (CoR)** is the local-axes origin. These can be translated away from the CoG in:

(Select the geometry +) [Graphic View Menu](#) > **Modify CoG/CoR position**

The local-axes origin can later be restored to the center of mass by clicking on  CoM.

 **Tip:** It is recommended to modify the CoG/CoR position with the object shape fixed, not with enforced or rigid body behaviour (see [geometry behaviour](#))

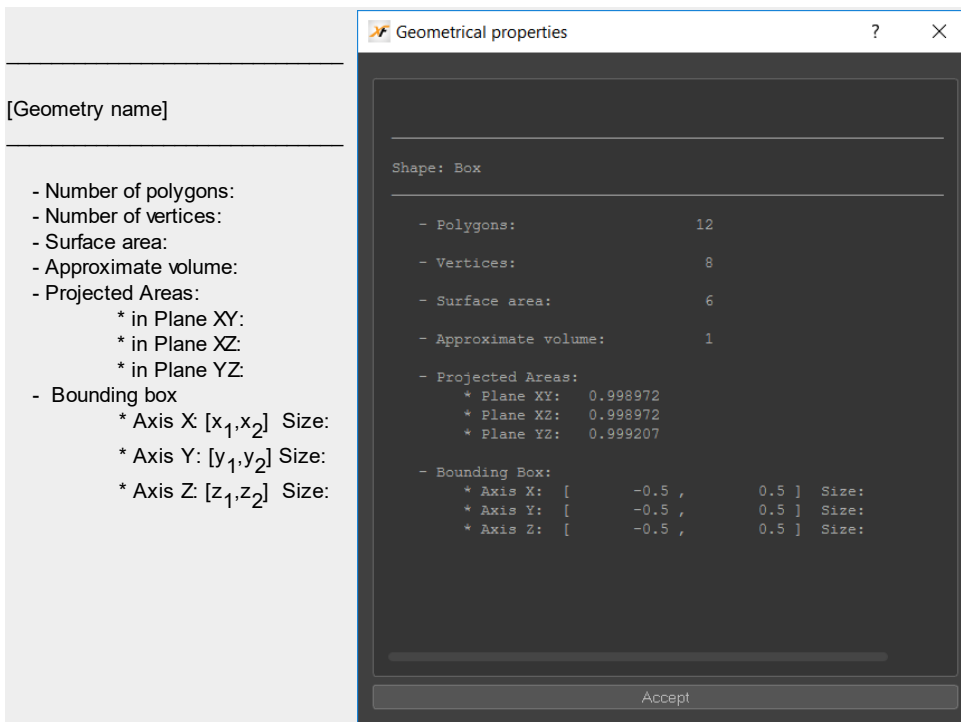
 **Please note:** the CoG/CoR cannot be modified in [post-processing mode](#) for simulation geometries as it would modify the simulated project.

4.5.3 Geometrical properties

(Select the geometry +) [Graphic View Menu](#) > **Show geometrical properties**

This option shows the following information about the geometry objects:

4.5 Geometry Information

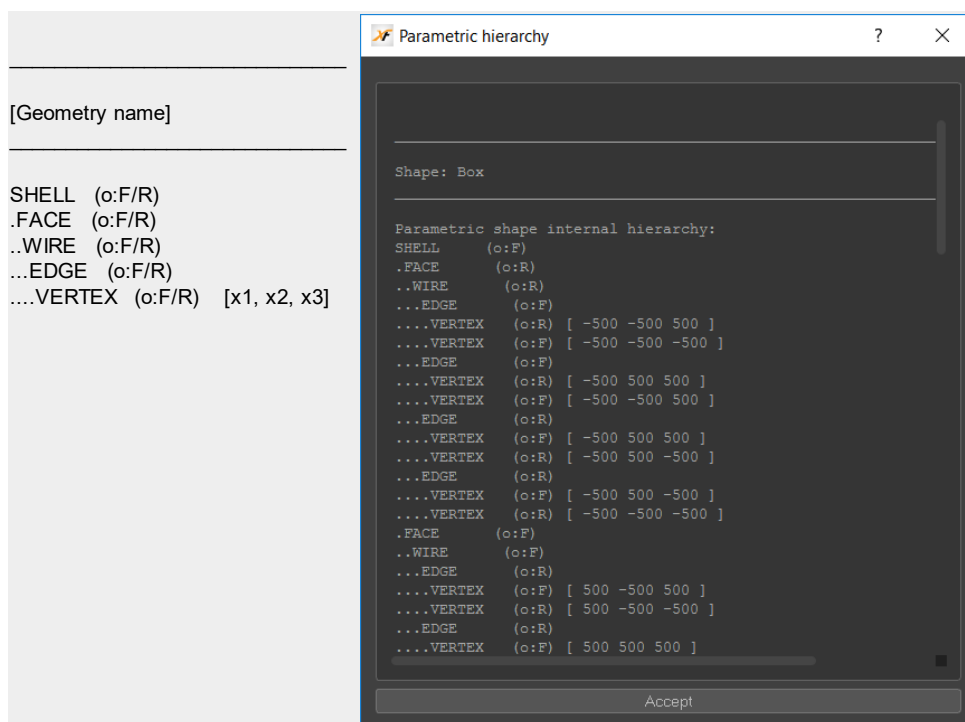


(x_1, y_1, z_1) being the coordinates of the lower corner and (x_2, y_2, z_2) those of the upper corner. Hence, it allows the user to know the dimensions of the selected object (see also [Dimensions](#)).

4.5.4 Parametric hierarchy

(Select the geometry +) [Graphic View Menu](#) > **Show parametric hierarchy**

This option is available only for parametric geometries (STEP, IGES) and shows information about the parametric hierarchy of the geometry selected:



The glossary is described at the bottom of the window and is the following:

- COMPOUND: A group of any of the shapes below.
- COMPSOLID: A set of solids connected by their faces. This expands the notions of WIRE and SHELL to solids.
- SOLID: A part of 3D space bounded by shells.
- SHELL: A set of faces connected by some of the edges of their wire boundaries. A shell can be open or closed.
- FACE: Part of a surface bounded by a closed wire. Its geometry is constrained (trimmed) by contours.
- WIRE: A sequence of edges connected by their vertices. It can be open or closed depending on whether the edges are linked or not.
- EDGE: A single dimensional shape corresponding to a curve, and bound by a vertex at each extremity.
- VERTEX: A zero-dimensional shape corresponding to a point in geometry.

The (o:F/R) stands for:

- o (Orientation): The orientation of the considered element. It describes where the material lies with respect to the parameterization of the element.
- F (Forward): The element orientation is forward.
- R (Reversed): The element orientation is reversed.

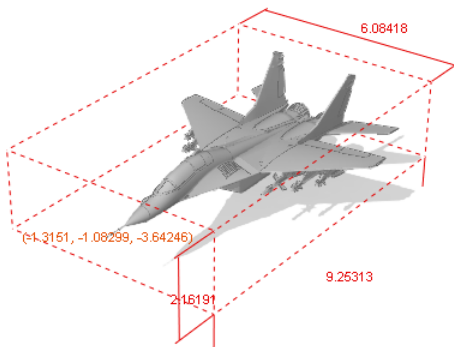
[x1, x2, x3] are the three coordinates of vertex.

4.5.5 Dimensions

(Select the geometry +) **Main menu > Geometry > Dimensions** or

This command shows/hides the bounding box dimensions and lower corner coordinates of the selected geometry object.

4.5 Geometry Information



Geometry dimensions

4.5.6 Check holes

Regarding the handling of complex geometries in XFlow, intersecting surfaces are allowed but holes in the geometry may cause the fluid to leak inside it.

The best practice is to create a watertight geometry using a CAD software. To do so, it is recommended to model from solids instead of patches in order to minimize as much as possible the presence of matching patches, where holes may appear if not perfectly matched.

Two patches are perfectly matched (i.e. no holes) if their tessellations coincide at every common joint, i.e. share nodes (see [Show mesh](#)).

Patch-to-patch matching tolerance can be controlled in:

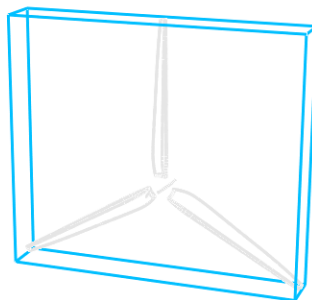
Main menu > Options > Preferences > Graphic View > OpenGL Mesh deflection,

To check the geometry to see if it contains holes due to non-matching patches, set the display in [Bounding Box](#) visualization mode and:

Select the geometry > Main menu > Geometry > Show/Hide holes,



Geometry with intersecting surfaces
(shading visualisation mode)



Show the holes of the geometry
(bounding box visualisation mode)

The Bounding box [visualization mode](#) makes the holes visible, highlighted by white lines, as shown in the figure above.



Tip: In models composed of several parts with the same boundary conditions it is recommended to first merge all parts in a single object.

4.6 Modify geometry

The following operations can be performed on a geometry object to modify its position, orientation and size:

- Translation: either from the [Project Tree](#), the [Transform Tool](#) or the [Graphic View](#)
- Rotation: either from the [Project Tree](#), the [Transform Tool](#) or the [Graphic View](#)
- Scale: in the [Transform Tool](#) or the [Graphic View](#)

Additionally, the [Toolbar Geometry Operations](#) provides the tools to modify the object shape, according to the following operations:

- [Symmetry](#)
- [Duplicate](#)
- [Split](#)
- [Merge](#)
- [Heal](#)

 **Please note:** The changes in the object are shown in wireframe mode while manipulating the *Transform Tool*. To apply the changes, press "**Apply all**" button.

 **Please note:** The changes done using the *Transform Tool* are not saved with the project. To keep the changes, it is recommended to create a new [merged shape](#) of this object: Select it, press **Main menu > Geometry > Create mesh shape** and delete the original object.

4.6.1 Translation

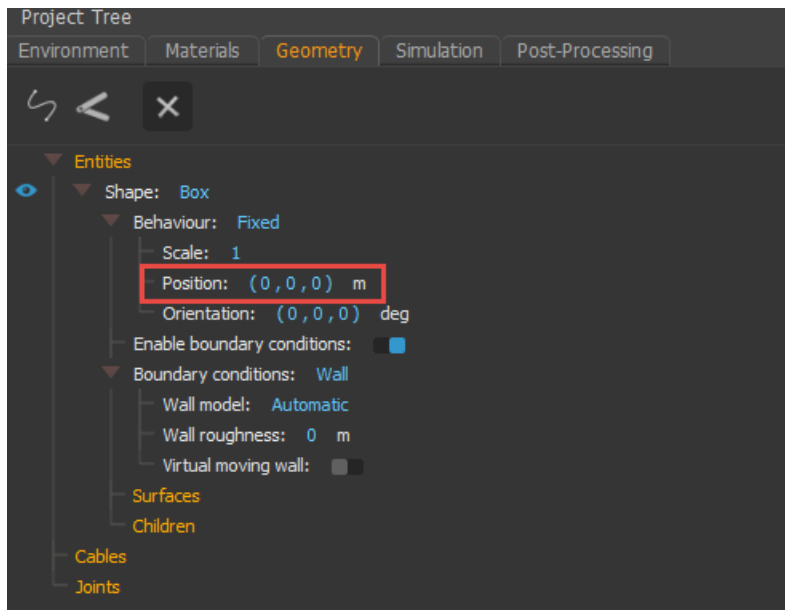
Translation operation can be performed either from the [Project Tree](#), the [Transform Tool](#) or the [Graphic View](#).

Project Tree

Project Tree > Geometry > Shape > Behaviour

In the data structure of a given geometry (**Shape**), its position can be modified as shown in the figure below.

4.6 Modify geometry

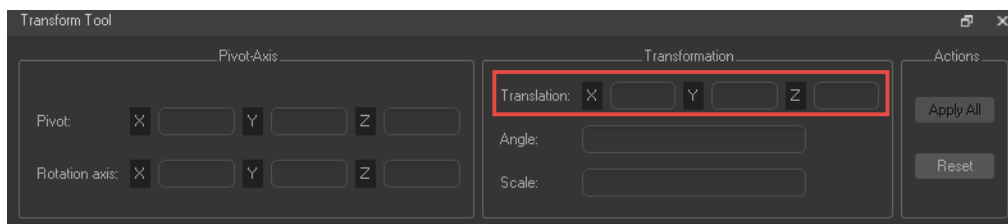


Object modification in Project Tree.

Transform Tool

To modify the position of the geometry object:

- Select the object (**Shape**)
- **Transform Tool > Transformation > Translation:** Set the displacement length by coordinates X, Y, Z (global axes).

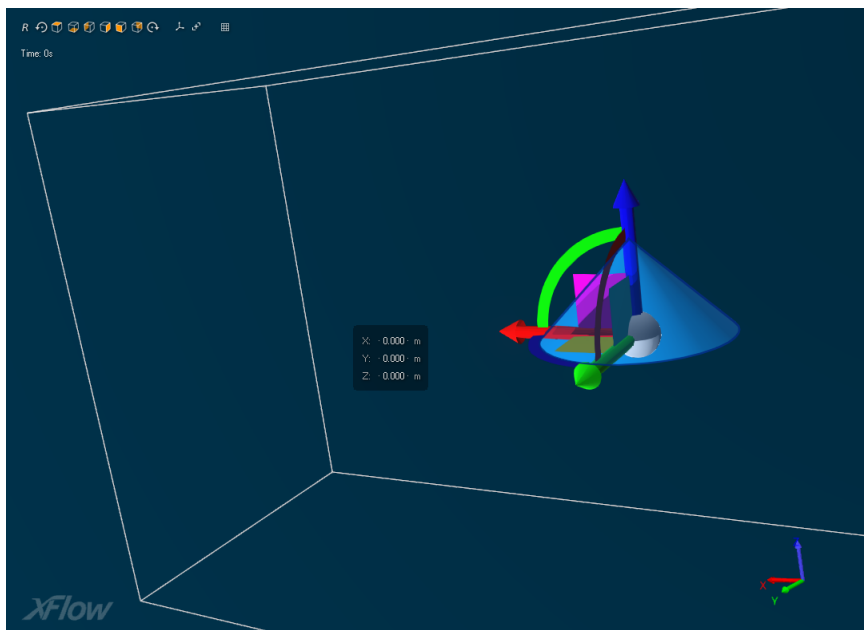


Transform Tool > Transformation > Translation

Graphic View

A geometry can be translated in the Graphic View using the translation Gizmos shown when selecting the geometry object, . The following two translation operations are allowed:

1. Translation in one direction (X, Y, Z) by:
 - **dragging** the corresponding Gizmo axis, or
 - **clicking** on the corresponding Gizmo axis and setting the translation distance (in meter) in the translation dialogue-box, e.g X = 1m, Y = 0m, Z=0m).
2. Translation in a plane (X-Y, X-Z, Y-Z) by:
 - **dragging** the corresponding Gizmos plane, or
 - **clicking** on a Gizmo axis and setting two translation distances (in meter) in the dialogue box, e.g: X = 1m, Y = 1m, Z=0m.
3. Translation in any direction by:
 - **clicking** on a Gizmo axis and setting three translation distances (in meter) in the dialogue box, e.g: X = 1m, Y = 1m, Z=1m.



Graphic View - Selected geometry showing its Gizmos and the translation dialogue-box

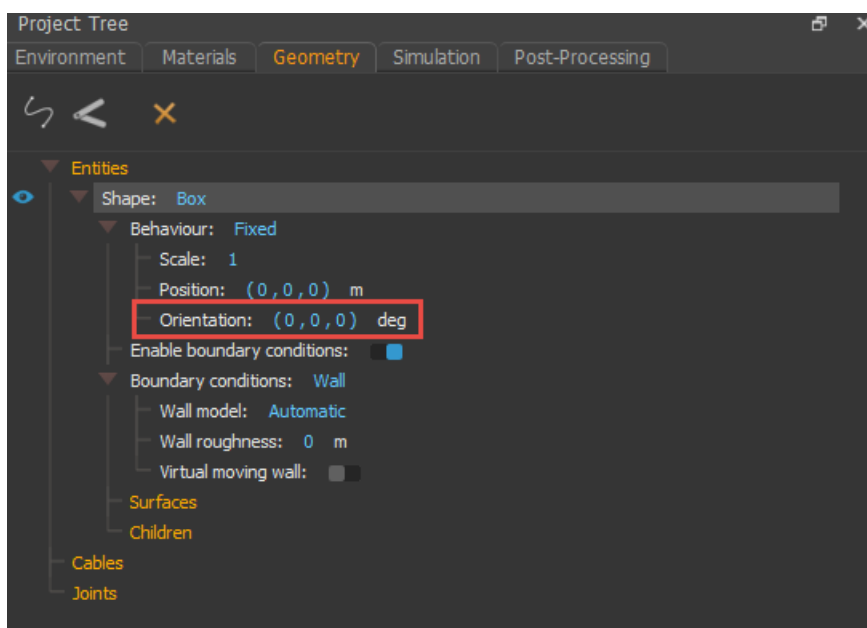
4.6.2 Rotation

An object can be rotated either from the [Project Tree](#), the [Transform Tool](#) or the [Graphic View](#).

Project Tree

Project Tree > Geometry > Shape > Behaviour

In the data structure of a given geometry (**Shape**), its orientation can be modified as shown in the figure below.



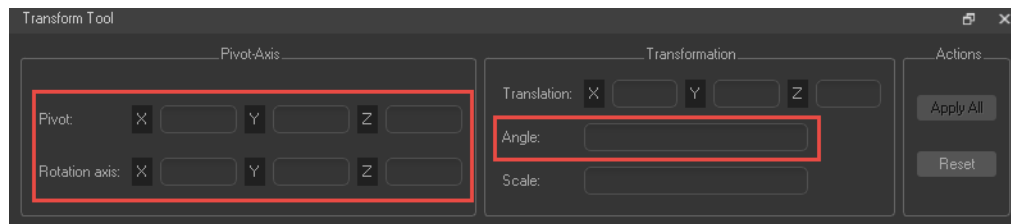
Object modification in Project Tree.

Transform Tool

To rotate a geometry object, follow these steps:

- Select the object (**Shape**)
- **Transform Tool > Pivot-Axis > Pivot:** Coordinates of the origin of the rotation axis (relative to object local axes)
- **Transform Tool > Pivot-Axis > Rotation axis:** Direction of the rotation axis
- **Transform Tool > Transformation > Angle:** Angle of rotation around a fixed axis, defined by **Pivot** and **Rotation axis**.

Notice that when selecting a geometry object, its geometrical center of gravity (assuming mass uniformly distributed) is automatically displayed in the **Pivot-Rotation > Pivot** field.

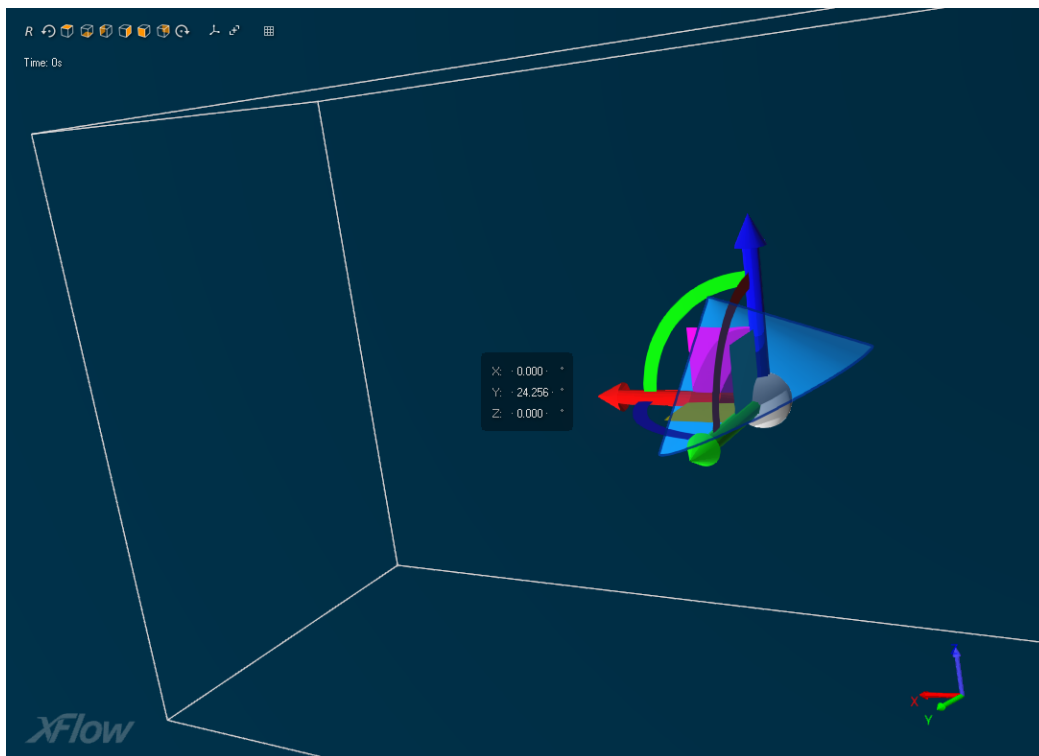


Transform Tool > Transformation > Rotation

Graphic View

A geometry can be rotated in the Graphic View using the rotation Gizmos shown when selecting the geometry object,  :

1. Rotation around an axis (X, Y, Z) by doing the following:
 - **dragging** the corresponding Gizmo arc, or
 - **clicking** on a Gizmo arc and setting the rotation angle (in degrees) in the rotation dialogue-box, e.g X = 90°, Y = 0°, Z=0°.



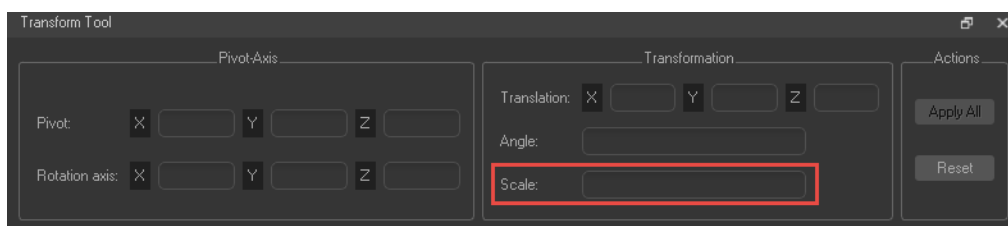
Graphic View - Selected geometry showing its Gizmos and the rotation dialogue-box

4.6.3 Scale

The size of an geometry can be scaled in the [Transform Tool](#) or the [Graphic View](#).

Transform tool

- Select the object (**Shape**)
- **Transform Tool > Transformation > Scale**: To scale the object size

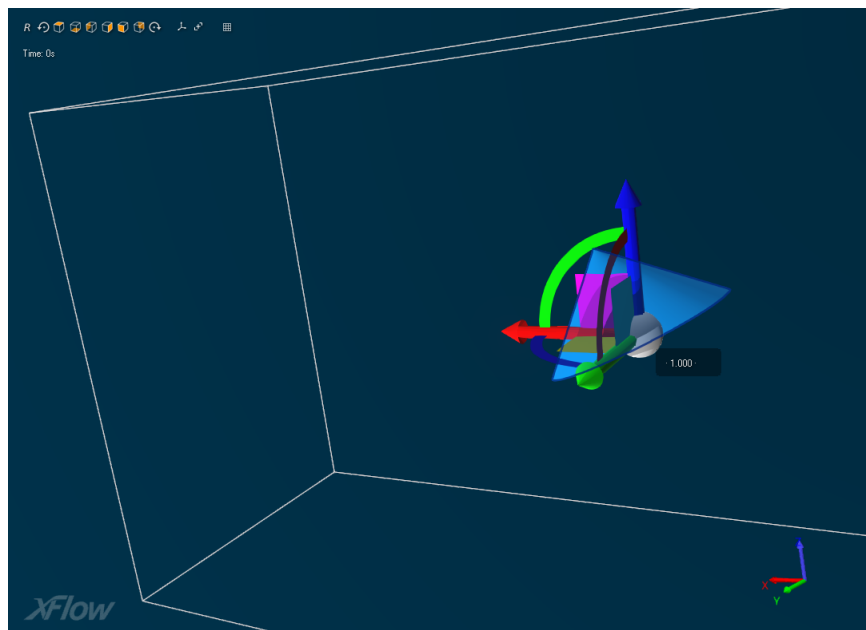


Transform Tool > Transformation > Scale

Graphic View

A selected geometry, , can be scaled in the Graphic View by **clicking** on the center of the Gizmo axes and setting a relative scaling value in the scaling dialogue-box.

4.6 Modify geometry

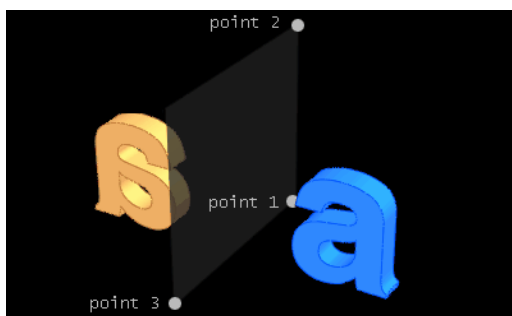


Graphic View - Selected geometry showing its Gizmos and the scaling dialogue-box

4.6.4 Symmetry

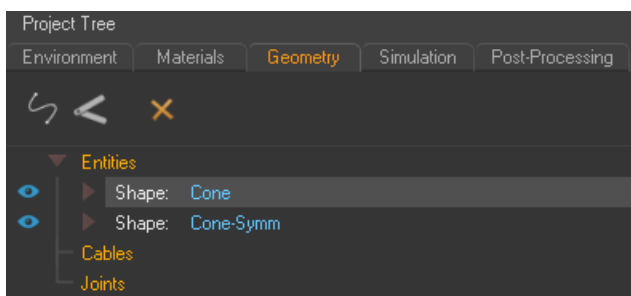
(Select the geometry) **Main menu > Geometry > Symmetry**, 

This command allows the user to make a mirrored copy of the selected geometry with respect to a plane defined by three points (vertex).



Plane of symmetry to duplicate a geometry

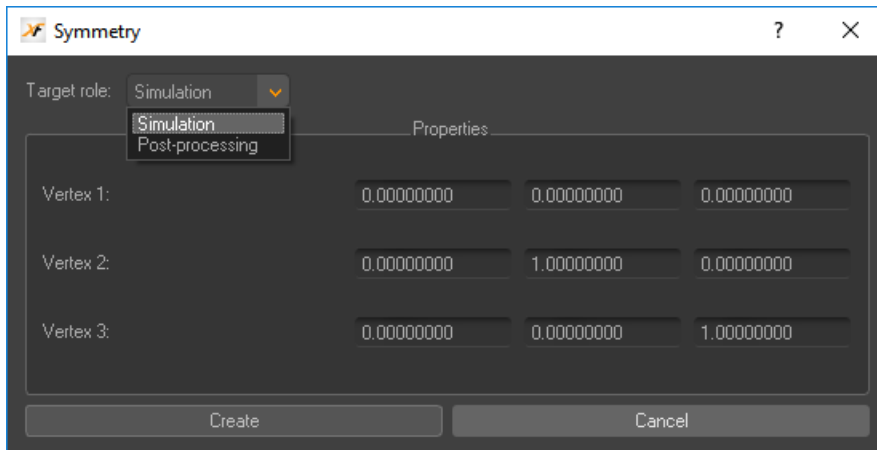
The name of the new **Shape** is the name of the original object plus the suffix "-Symm".



Original object: Sphere; New (symmetric) object: Sphere-Symm

When creating a Symmetric geometry the user will be prompted to choose the target of the newly created geometry, which can either be:

- Simulation: the geometry can be used for the simulation setup (only available when not in [Post-processing mode](#)).
- Post-processing: the geometry can be used only for post-processing tasks.
-

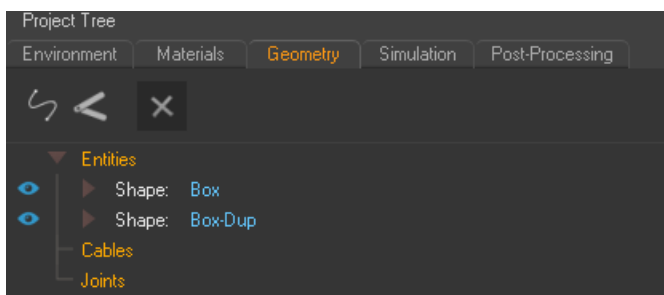


4.6.5 Duplicate

(Select geometry/face/group of faces) **Main menu > Geometry > Duplicate**, 

This command creates a copy of the selected geometry/face/group of faces. The new **Shape** will have the same position and orientation as the original one.

The name of the new **Shape** is the name of the original object plus the suffix "-Dup". If the original geometry is duplicated several times, then the resulting **Shapes** will be numbered according to: "original-Dup-000001, original-Dup-000002, etc".



Original object: Sphere; New (duplicated) object: Sphere-Dup

4.6 Modify geometry

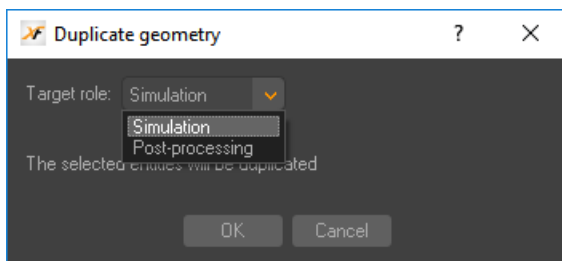
Example:

- Original geometry - **Shape1**
- Select several faces of **Shape1** using 
- Duplicate the selected faces using . It generates a new geometry object **Shape1-Duplicate**, that has the same shape as the selected faces.

Please note: If the user now presses **Del** key, the selected faces will be extracted from the original geometry (see [delete geometry](#)) and the user will have the original geometry split up into two **Shapes**.

When duplicating a geometry the user will be prompted to choose the target of the newly created geometry, which can either be:

- Simulation: the geometry can be used for the simulation setup (only available when not in [Post-processing mode](#)).
- Post-processing: the geometry can be used only for post-processing tasks.



4.6.6 Replace

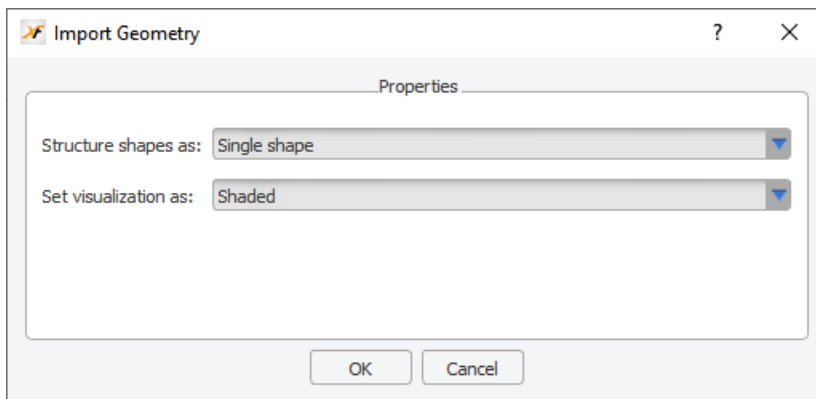
(Select geometry/face/group of faces) **Main menu > Geometry > Replace selected geometry**,

This command replaces the selected geometry with a new one imported by the user from a new file. The **Shape** will keep the same position and orientation details as the original one.

Example:

- Original geometry - **Shape1**
- Select the **Shape1** geometry in the tree or graphic view
- Replace the selected geometry with a new file imported by the user

When replacing a geometry the user will be prompted to choose the import geometry settings ([+info](#)).

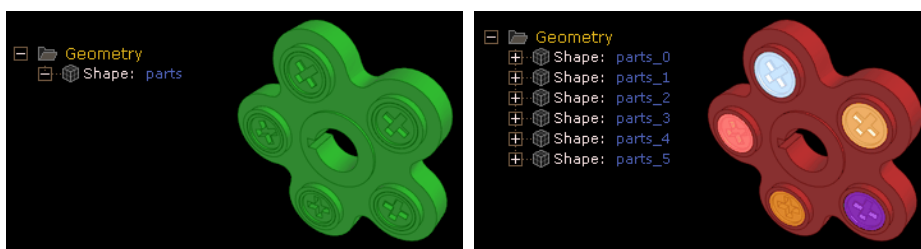


4.6.7 Split

These operation allow the user to split the selected object into its connected parts or into parts with the same surface orientation:

(Select the geometry) **Main menu > Geometry > Split in connected shapes,** 

(Select the geometry) **Main menu > Geometry > Split in shapes according to orientation,** 



Original complex geometry

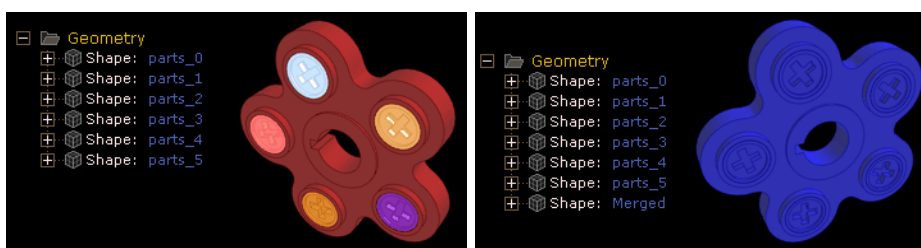
After splitting: several smaller geometries

Please note: Splitting only works for geometries imported as STL files (mesh of triangles).

4.6.8 Merge

(Select the geometries) **Main menu > Geometry > Create mesh shape,** 

This command creates a new shape by merging the selected geometry objects. The new **Shape**, called "Mesh", includes all the original parts.



Original geometry objects

Geometry after merging

4.6 Modify geometry

 **Tip:** Once the merged shape has been created, it is recommended to delete the original parts (see [Delete geometry](#)).

4.6.9 Repair geometry

If a geometry contains [holes](#), some sort of it can be directly repaired in XFlow by the following operations:

(Select the geometry +) **Main menu > Geometry > Healing** , or  :

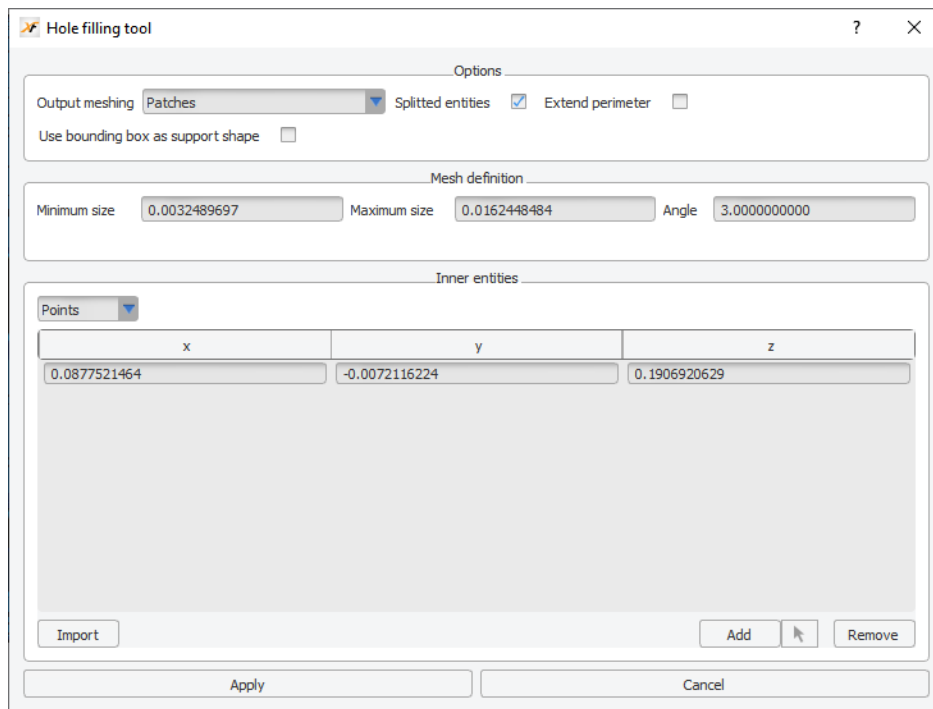
- Basic healing: To repair minor problems with the parametric representation of the geometry.
- Fix small lines: It removes lines that are exceedingly small
- Fix small faces: It removes faces that are exceedingly small
- Remove isolated edges: To remove edges that are not adjacent to faces
- Sew faces: It tries to create watertight connections between faces
- Complete healing (includes all the previous operations plus [reorientation of normals](#))
- Hole filling tool: Tool to create a lid geometry to fill the holes of a geometry ([+ info](#))

 **Please note:** Healing is available only for parametric surfaces (.STEP, .IGS). When the geometry is imported as tessellated surfaces (.STL), the healing functionality is not available.

4.6.9.1 Hole filling tool

The hole filling tool allows to create a lid geometry that will fill the holes of geometry. The user can select one or more geometries (a geometry set) for the tool to calculate the patches to close their holes and as a result new geometries will be obtained containing the resulting patches. Once generated, the geometries are automatically imported into the project.

In order to use it, select the geometry set and use the menu **Geometry > Repair geometry > Hole filling**. The Hole filling tool window will then pop-up.



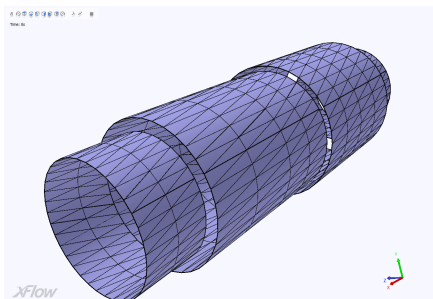
Hole filling tool window

Options

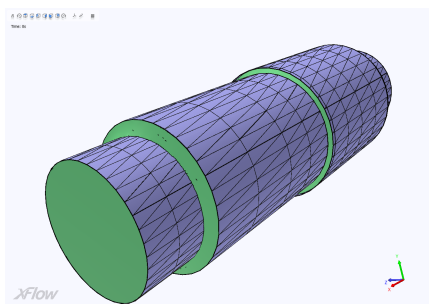
This section includes the general options of the hole filling tool.

- ↪ **Output meshing**: Selects whether only the lid geometries (**Patches**) or an wrapped type geometry (**Wrap**) are imported in the XFlow setup after the hole filling process.
- ↪ **Split entities**: (Only available for **Patches**) Imports each lid geometry as a separate entity in the XFlow setup.
- ↪ **Extend perimeter**: (Only available for **Patches**) The edges of each lid geometry created is extended to intersect the original geometry used in the hole filling process.
- ↪ **Scale**: (Only available for **Wrap**) Scales the created wrap geometry by a factor.
- ↪ **Use bounding box as support shape**: An exterior bounding box is automatically created considering the selected geometry set and used as an auxiliary geometry to improve the stability of the hole filling tool

Example of hole filling process:

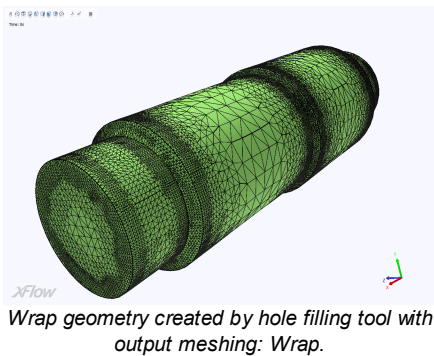


Geometry prior to hole filling tool process showing missing surfaces and gaps.



Initial geometry (purple) plus patches (green) created by hole filling tool with output meshing: Patches.

4.6 Modify geometry

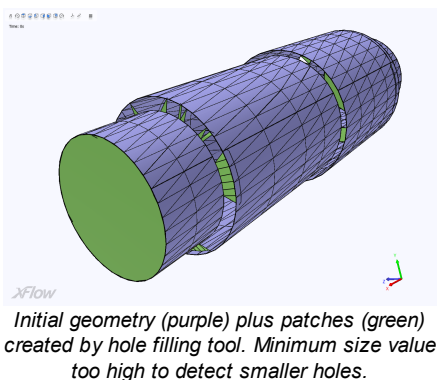


Mesh definition

Parameters of the mesh generated for the lid geometries.

- ↳ **Minimum size:** Minimum size of the mesh triangles (in m) for the patches to close the holes. It must be smaller than the smallest of the holes to close.
- ↳ **Maximum size:** Maximum size of the mesh triangles (in m) for the patches to close the holes. It must be smaller than the largest of the holes to close.
- ↳ **Angle:** Sharp edge angle (in degs)

The the minimum size and maximum size parameters are initialized with values automatically calculated based on the bounding box of the selected geometry set. Although these values lead to a successful geometry healing in most cases, adjustments might be required according to the topology of the geometry considered, the granularity and sizes of the holes.



Inner entities

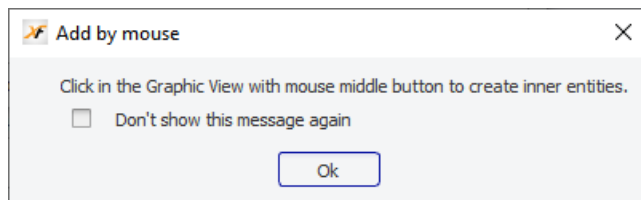
Two types of inner entities can be created in order to help the generation of the lid geometry: inner points and inner spheres.

- ↳ **Inner point:** Seed point for the meshing to define the domain to be meshed. It must be located inside the volume to close.
- ↳ **Inner sphere:** Places a hot-spot as input information for the hole filling process. The algorithm will calculate a seed point automatically in the largest open area inside the geometry.

The list of inner points (default option) is initialized with the point corresponding to the CoG of the selected geometry set. This corresponds to the CoG of the selected geometry, when a single entity is selected, and to an average CoG when several entities are selected.

⚠ Please note: When selecting the Inner spheres option a seed point is automatically created in the largest open area inside the geometry.

Inner entities can be specified by adding new lines to the inner entities table or by mouse selection (arrow icon next to the Add button). In the latter case, selecting the middle mouse button in a location near in the graphic viewer will create a new inner entity. By default inner spheres are created with a radius of 0.1 m.



Add inner entities by mouse.

Moreover, data can be imported from text files using the **Import** option in the Hole Filling Tool window. The syntax required in the text file is:

↳ **Inner point:** x y z

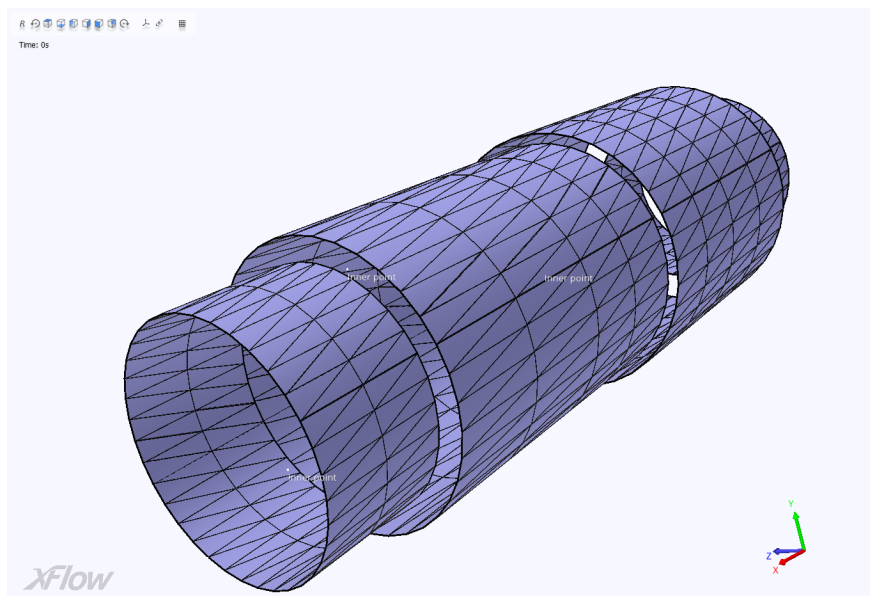
↳ **Inner sphere:** x y z radius

Values should be specified in meters (m) and both space and tab separators are supported. Example files:

Example: Points.txt		
0.1	1.4	0
2.3	-1.7	1
-5.5	3.8	0

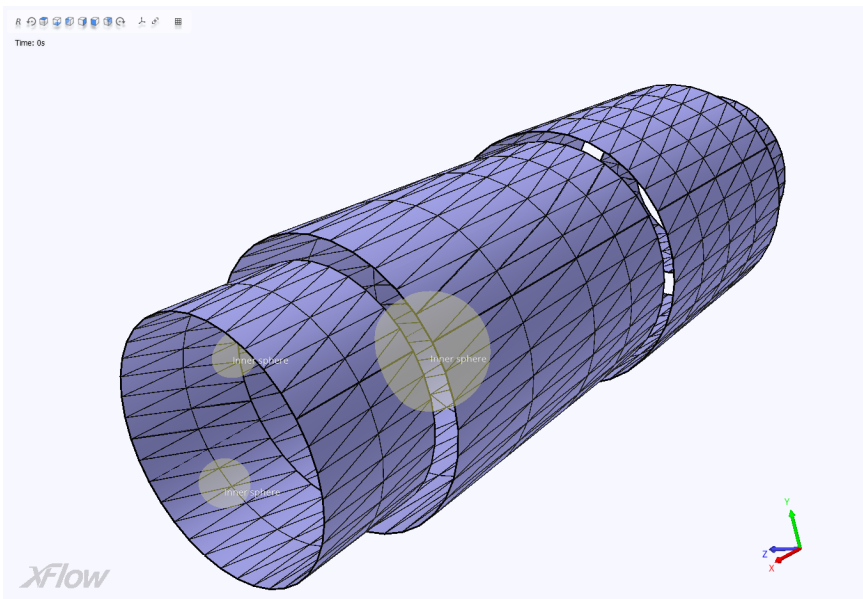
Example: Spheres.txt			
0.1	1.4	0	0.01
2.3	-1.7	1	0.05
-5.5	3.8	0	0.03

Inner entities are visualized in the Graphic Viewer as long as the Hole filling tool window is open.



Inner points visualization in Graphic Viewer.

4.6 Modify geometry



Inner spheres visualization in Graphic Viewer.

Tip: Better results are obtained using the inner points option, placing several points within the geometry as close as possible to the holes to close. The inner entities are optional, if the user does not set one an automatic inner sphere will be used, however this makes the hole filling process more prone to failures.

4.7 Delete geometry

There are two ways to remove a geometry object:

1. (Select the geometry +) press **Del** key
2. (Select the geometry +) **Right click** on the *Graphic View* window > **Remove selected**

Once deleted, the shape disappears from the *Project Tree* (**Geometry>Geometries**) and the *Graphic View* window.

Please note: There is no Undo option; therefore the geometry will be deleted permanently.

Please note: Geometries cannot be removed in post-processing mode.

5 Simulation Setup

This chapter explains how to setup a simulation, following from left to right the workflow of the *Project Tree*:



XFlow workflow: Project Tree categories, those corresponding to the Simulation Setup are highlighted in orange.

The simulation setup consists of the first four steps:

- **Environment:** including [Engine](#) and [Environment](#)
- [Materials](#)
- [Geometry](#)
- [Simulation](#)

Depending on the options selected in the project, some toolbar icons may show. The simulation project toolbar icons available are:

Toolbar Simulation Project

	Add scalar	Environment > Engine > Scalar transport
	Add joint	Geometry > Joints
	Add cable	Geometry > Cables
	Create refinement region	Simulation > Resolution > Regions
	Create static pressure filter	Simulation > Store data > Static pressure filters
	Create probe	Simulation > Store data > Probes

5.1 Engine

Project Tree > Environment > Engine

The **Engine** section allows the user to specify the solver features according to the **physics** of the problem to be solved; available features are:

5.1 Engine

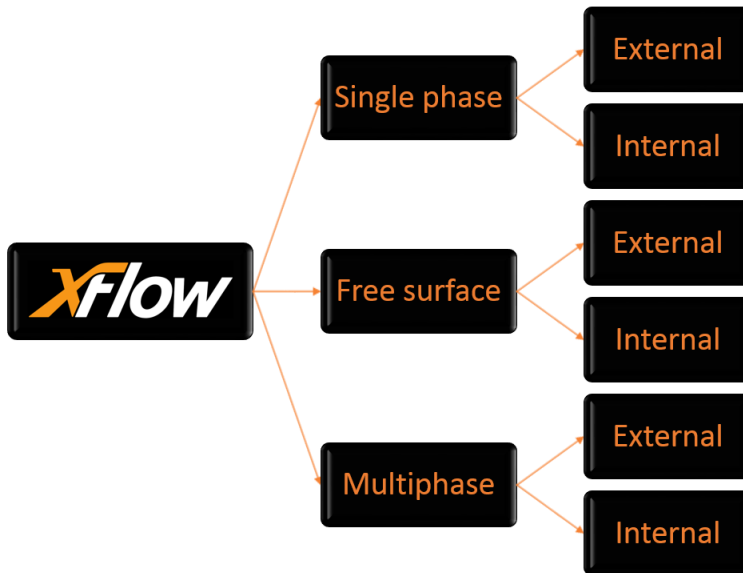
Kernel	3d 2d	
Flow model (+Info)	Single phase ↳ Execution	CPU GPU
	Free surface Multiphase	
Multiphase model (+Info)	Particle-based tracking Phase field VoF	
Analysis type (+Info)	Internal External	
Thermal model (+Info)	Isothermal Segregated energy ↳ Radiation	Disabled Montecarlo
	Coupled energy Supersonic Sequential solid	
Turbulence settings (+Info)	Turbulence model (+Info)	Automatic Off (Resolved) Smagorinsky Dynamic Smagorinsky Wall-Adapting Local Eddy Spalart-Allmaras ⁽⁷⁾
	Turbulence generation (+Info)	Off Automatic Custom
Acoustic analysis (+Info)	Refractive index	
Scalar transport (+Info)		
Advanced Options (+Info)	High order boundary conditions Force evaluation scheme Structural analysis Time integration scheme Wall function time filter Enforced incompressible model Moving parts surface normal detection Volume correction Damping outlet region Enable viscous term in energy equation Refractive index	

Engine **technical** settings, such as number of CPUs, are specified in:

Main Menu > Options > Preferences > Engine (see [Preferences > Engine](#))

5.1.1 Flow models

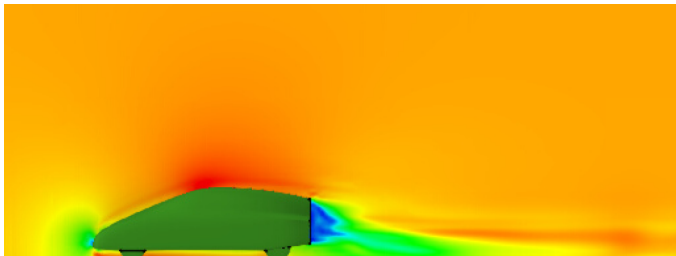
Project Tree > Environment > Engine > Flow Model



Flow models schematic

Single phase:

Single phase flow model only involves one continuous fluid phase within the whole fluid domain. Any point of the fluid domain is the same fluid material (e.g. air, water, etc.). It is typically used for aerodynamics, acoustics, thermal management, etc.

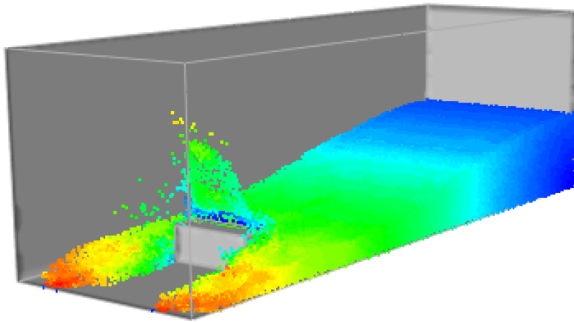


Single phase analysis example: air aerodynamics around a vehicle

Free surface:

Free surface flow model involves one fluid phase with a free surface interface. The presence of fluid is defined by a volume of fluid field which is between 0 and 1. Presence of fluid is not mandatory in the domain, and areas for which volume of fluid is equal to 0 is vacuum. This is typically used for marine, hydrodynamics, water management, and any simulation involving liquid free surface.

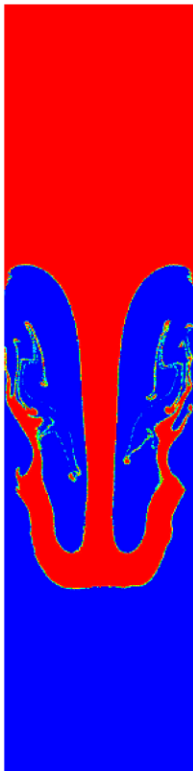
5.1 Engine



Free surface analysis example: water dam break over an obstacle

Multiphase:

Multiphase flow model involves two phases. The main phase is defined by a volume of fluid field such as in Free Surface, but the areas for which the volume of fluid field is equal to 0 is another fluid in this case (second phase). The main and second phase can both be either gas or liquid. This is typically used for two-phase flows, or hydrodynamics and water management where the air phase is not negligible.



*Multiphase analysis
example:
Rayleigh-Taylor
instabilities (heavy
fluid over lighter fluid)*

Project Tree > Environment > Engine > Flow model > Execution: CPU/GPU.

This option is only available in [Labs mode](#) and allows to select if the simulation will run on CPU or GPU.

5.1.2 Multiphase models

Project Tree > Environment > Engine > Multiphase model

There are three different approaches to model multiphase flows in XFlow. The **Particle-based tracking** model, the **Phase field** model, and the **VoF**.

Particle-based tracking:

Starting with an initial distribution of each phase, the markers are advected with the flow, and additional forcing terms are introduced at the interface between phases to recover the surface tension specified by the user. The **Particle-based tracking** method advects discrete markers at the bulk of the flow and detects an interface whenever particles of different type are neighbouring each other.

In the **Particle-based tracking** method, the particles are moving independently of the lattice, the Lagrangian trajectories of the particles are not guaranteed to obey the incompressibility condition. There are therefore volume changes for local portions of a phase embedded in other phase (such as bubbles or drops), and this is globally corrected to ensure global volume conservation of both phases. If your simulation has distinctly separated volumes (such as a positive displacement pump), this correction will effectively transfer mass from one volume to another. The global correction can be disabled with the [Environment > Engine > Advanced Options > Free surface volume correction](#) option, both for free surface and particle-based multiphase simulations.

If the **Particle-based tracking** multiphase model is enabled, the following options appear:

[Engine > Advanced options > Volume correction](#): **On** by default. This option is only available in [Expert mode](#).

[Environment > Environment > Water channel > Fluid 1 initial surface](#)
[Environment > Environment > Water channel > Fluid 1 inlet wave function](#)

[Materials > Materials > Material 1](#)
[Materials > Materials > Material 2](#)
[Materials > Interactions](#)

[Geometry > Geometries > Shape > Boundary Conditions > Apply to phase:](#)
[Material 1/Material 2](#)

Phase field:

This Multiphase model is only available in [Labs mode](#).

Starting with an initial distribution of each phase, the phase field is advected with the flow and additional forcing terms are introduced at the interface between phases to recover the surface tension specified by the user. Instead of discrete markers, the **Phase field** method advects a scalar field that, besides the terms due to pure advection, has additional forcing terms originating from a potential with attractors at concentration $\text{vof} = 0$ and $\text{vof} = 1$, enforcing separation of phases. Similarly, large changes in the **Phase**

field are interpreted as an interface.

The **Phase field** has the advantage of a one-to-one match between the lattice discretization for the momentum equations and for the **Phase field** equations. Conservation of mass does not preclude the dissolution of one phase into the other when the shear forces are larger than the potential terms that enforce phase separation.

Environment > Environment > Water channel > Fluid 1 initial concentration
Environment > Environment > Water channel > Fluid 1 inlet concentration

Materials > Materials > Material 1
Materials > Materials > Material 2
Materials > Interactions

Geometry > Geometries > Shape > Boundary Conditions > Fluid 1 concentration

VoF:

This Multiphase model is only available in [Labs mode](#).

The **Volume of Fluid** (VoF) approach is in between the phase field and the **particle-based tracking** method. Unlike the **particle-based tracking** approach which uses markers, or the **phase field** approach which uses a diffusive interface, the **VoF** interface is tracked using a fluid volume at interface lattice nodes and therefore the interface can only be one lattice unit thick.

On the contrary to the **particle-based tracking**, the pressure field is solved continuously for the whole domain (i.e. there is no boundary condition between the gas and liquid) in the **VoF** approach. This results in a smoother pressure field at the interface compared to the **particle-based tracking** and a more accurate modeling of the surface tension. Similarly to the **particle-based tracking**, the **VoF** is more adapted to large scale problems than the **phase field** model, however the latter should be the most accurate approach to model surface tension.

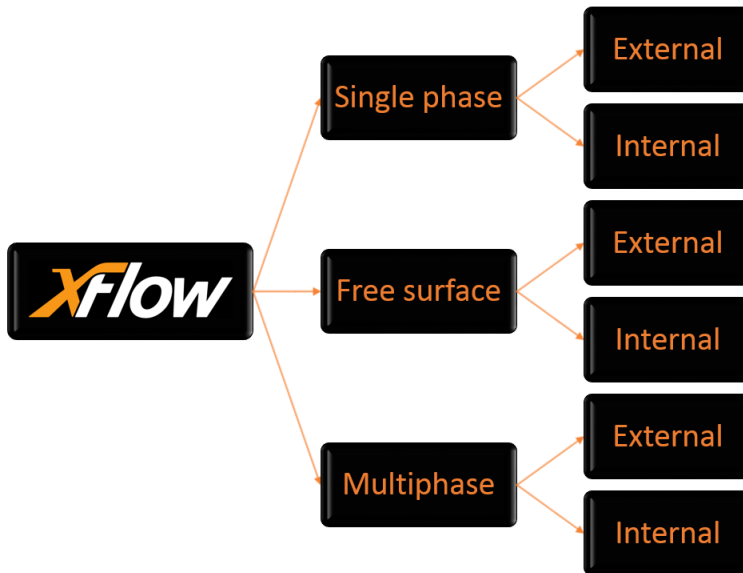
Environment > Environment > Water channel > Fluid 1 initial concentration
Environment > Environment > Water channel > Fluid 1 inlet concentration

Materials > Materials > Material 1
Materials > Materials > Material 2
Materials > Interactions

Geometry > Geometries > Shape > Boundary Conditions > Fluid 1 concentration

5.1.3 Analysis types

Project Tree > Environment > Engine > Analysis type

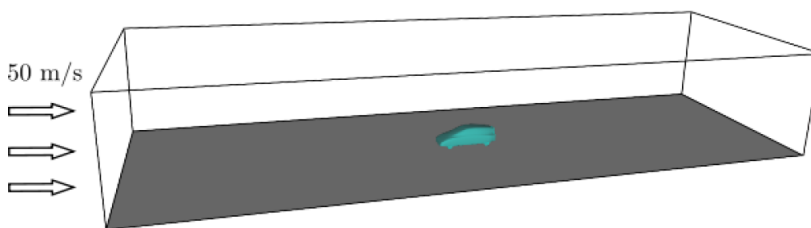


Flow models schematic

External:

External simulations do not require geometry to define the boundaries of the fluid domain. Fluid domain is defined by an XFlow pre-defined domain:

- Single phase: [virtual wind-tunnel](#) or [generic rectangular domain](#)
- Free surface: [virtual water channel](#)
- Multiphase: [virtual water channel](#)

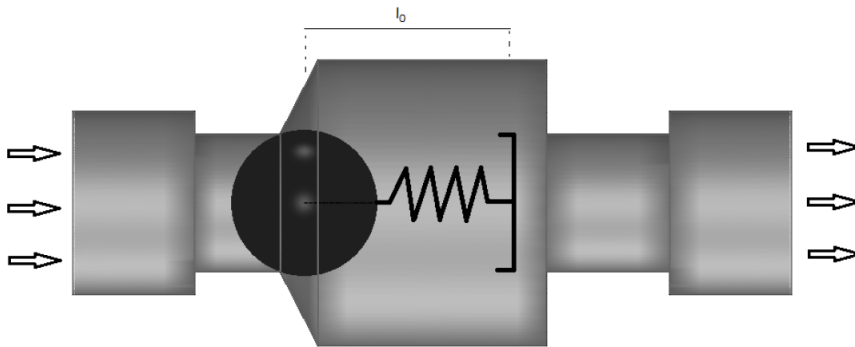


Example of external simulation: car aerodynamic in wind tunnel

Internal:

Internal simulation have no default fluid domain boundaries. It requires a geometry input from the user to define the boundaries of the fluid domain.

Tip: Make sure the geometry defining the domain is perfectly watertight so that XFlow can generate the fluid domain correctly.



Example of internal analysis: pipe flow with ball check valve

5.1.4 Thermal and radiation models

Project Tree > Environment > Engine > Thermal Model

Energy equation:

The energy transport in XFlow is modelled according to the following sensible-enthalpy conservation-equation:

$$C_p \frac{D(\rho T)}{Dt} = \underbrace{\nabla \cdot (k \nabla T)}_{\text{Heat flux}} + \underbrace{\nabla \cdot (\tau \cdot \vec{v})}_{\text{Viscous term}} - \underbrace{p \nabla \cdot \vec{v}}_{\text{Compressibility effects}}$$

where T is the temperature, ρ is the density, C_p is the specific heat capacity, k is the thermal conductivity, τ is the viscous stress tensor, and $\vec{v}$ the velocity vector.



Please note: The viscous term can be disabled in:

Project Tree > Engine > Advanced Options > Enable viscous term in energy equation

Thermal models

The thermal models available in XFlow are:

- **Isothermal:** The [energy equation](#) is not solved. The fluid temperature remains constant in space and time.

Examples of applications: subsonic aerodynamics, non-thermal applications, low compressibility effects, etc.

- **Segregated energy:** The [energy equation](#) is solved without the compressibility term. Buoyancy effects are included in the flow-motion equations using Boussinesq approximation:

$$\rho = \rho_0 - \alpha \rho_0 \Delta T$$

where α is the thermal expansion coefficient and ρ_0 is the reference density. This approximation is only valid for small density and temperature variation only.

Examples of applications: HVAC, room ventilation, natural and forced convection, electronics cooling, etc.

- **Coupled energy:** Only available in [Labs mode](#). The [energy equation](#) is solved and takes into account for the compressibility term. This solver is useful in order to account for the pressure/temperature variations when the gas is highly compressed/expanded, and is valid only for isentropic processes.

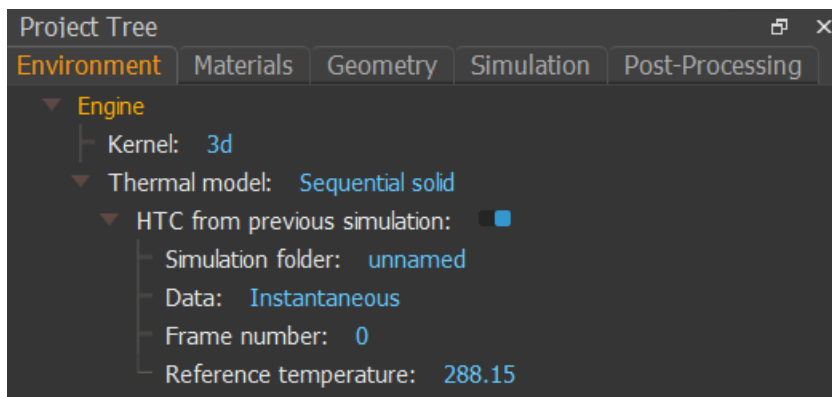
Examples of applications: adiabatic isentropic compression, expansions, etc.

- **Supersonic:** Only available in [Labs mode](#). Allows to solve flows with speeds higher or close to the speed of sound.

Examples of applications: aerodynamics for supersonic, transonic, and hypersonic flows, flows involving shockwaves, etc.

Please note: The coupled energy and supersonic solvers are only a prototypes and therefore the output should be considered with precaution.

- **Sequential solid:** Allows to solve the temperature distribution inside a solid body without solving the flow field around it. This can be used sequentially, computing a flow field from a first simulation that provides the [heat transfer coefficient](#) (HTC) distribution on a geometry, and compute only the solid conduction inside this geometry in a second simulation. This is especially useful to decouple the flow field (short time scales) and thermal (long time scales) convergence.



HTC from previous simulation

Please note: When the Sequential solid model is used, the simulation will only occur in geometries that are set as [conjugate heat transfer](#) (CHT).

In case the **HTC from previous simulation** is enabled, the boundary conditions will be read from the specified simulation results. The **Simulation folder**, **Data** type (instantaneous or averaged) and the **Frame** of interest must be specified. The **Reference temperature** is required in case the HTC boundary condition is applied from an isothermal simulation.

The default boundary condition applied to the CHT body when using a HTC distribution is [Free Convection](#). Any other boundary condition applied by the user will overwrite the HTC distribution on the surface.

In case the **HTC from previous simulation** is disabled, the default boundary condition applied to the CHT body will be adiabatic. Any other boundary condition applied by the user will overwrite this.

Montecarlo radiation model

Project Tree > Engine > Thermal model > Radiation model > Montecarlo

In XFlow, fluids are considered non-participating media by default. Therefore, radiation source terms are not accounted for in the [energy equation](#). However, XFlow do consider surface-to-surface radiative heat transfer mode, which is modelled according to Montecarlo model.

In the Montecarlo model the radiation leaving a surface element in a certain solid angle is approximated by

5.1 Engine

a single ray and therefore involves the tracing of rays from one surface to another through the domain. This technique provides a prediction of radiative heat transfer between surfaces without calculation of visibility coefficients. The accuracy of the model is determined by two parameters:

- ↳ **Ray density:** number of rays per element
- ↳ **Number of iterations:** number of iterations to perform in the iterative radiation process

Radiative surfaces are assumed gray bodies by default. Hence, the net radiative heat flux from a surface (radiation source) is given by:

$$q_{out} = (1-\varepsilon) q_{in} + \varepsilon \sigma \theta^4$$

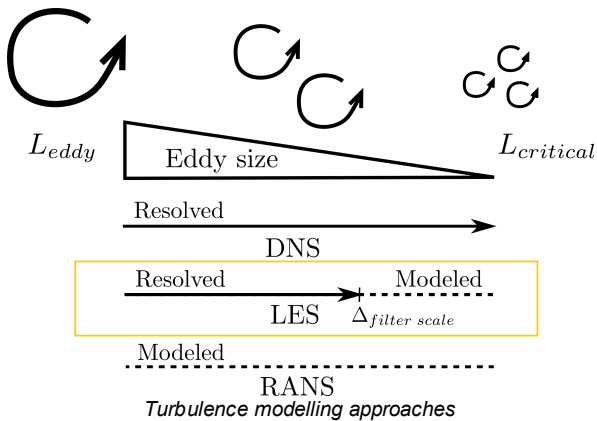
with q_{in} incident radiative heat flux, ε the emissivity coefficient of the surface ($0 \leq \varepsilon \leq 1$), σ the [Stefan-Boltzmann constant](#) and θ temperature.

Radiation sources are assumed to be at constant conditions over time; therefore irradiance is computed only once at the beginning of the simulation.

5.1.5 Turbulence models

Project Tree > Environment > Engine > Turbulence Model

Turbulence modelling is approached in XFlow using Large Eddy Simulation (LES). As sketched in the figure below, LES solves the turbulence scales that are larger than a given filter, scales below this filter are modelled.



XFlow provides the following survey of models to represent the smallest scales of turbulence (i.e. the turbulent viscosity):

- [Automatic](#)
- [Off \(Resolved\)](#)
- [Smagorinsky](#)
- [Dynamic Smagorinsky](#)
- [Wall-Adapting Local Eddy](#)
- [Spalart-Allmaras](#)

Automatic

By default, XFlow selects automatically the [Wall-Adapting Local Eddy](#) model.

Off (Resolved)

This option disables the modeling of subgrid-scales.

If this option is activated and the wall-type boundaries are set to `Resolved`, then the turbulence-modeling approach becomes a Direct Numerical Simulation (DNS). For the sake of accuracy, please make sure that dissipative scales of your problem are larger than the lattice resolution (smallest resolved scale).

Smagorinsky model

In the Smagorinsky model, the eddy viscosity is modeled by

$$\begin{aligned}\nu_{turbulent} &= \Delta^2 |S| \\ \Delta &= C_s \text{Vol}^{1/3} \\ S &= \sqrt{2S_{\alpha\beta}S_{\alpha\beta}}\end{aligned}$$

where Δ is the filter scale, S is the strain rate tensor of the resolved scale and the Smagorinsky constant (C_s) usually has a value between 0.1 and 0.2; by default $C_s = 0.12$. The user can modify this value in the

Project Tree:

Project Tree > Environment > Engine > Turbulence settings > Turbulence model > C_s

Dynamic Smagorinsky model

The dynamic Smagorinsky model is a modification of the previous model where C_s is dynamically computed based on the information provided by the resolved scales of motion, and thus may vary in space and time. C_s is initialized by default to 0.12; it can be modified in:

Project Tree > Environment > Engine > Turbulence settings > Turbulence model > C_s

This model uses two filter scales to avoid non-zero turbulent viscosity in the laminar case.

Wall-Adapting Local Eddy (WALE) model

The Wall-Adapting Local Eddy-viscosity (WALE) model has good properties both near to and far from the wall and both for laminar and turbulent flows. This model recovers the asymptotic behavior of the turbulent boundary layer when this layer can be directly solved and it does not add artificial turbulent viscosity in the shear regions out of the wake. The WALE model is formulated as follows:

$$\begin{aligned}\nu_{turbulent} &= \Delta^2 \frac{(G_{\alpha\beta}^d G_{\alpha\beta}^d)^{3/2}}{(S_{\alpha\beta} S_{\alpha\beta})^{5/2} + (G_{\alpha\beta}^d G_{\alpha\beta}^d)^{5/4}} \\ S_{\alpha\beta} &= \frac{1}{2} \left(\frac{\partial v_\alpha}{\partial r_\beta} + \frac{\partial v_\beta}{\partial r_\alpha} \right) \\ G_{\alpha\beta}^d &= \frac{1}{2} (g_{\alpha\beta}^2 + g_{\beta\alpha}^2) - \frac{1}{3} \delta_{\alpha\beta} g_{\gamma\gamma}^2 \\ g_{\alpha\beta} &= \frac{\partial v_\alpha}{\partial r_\beta} \\ \Delta &= C_w \text{Vol}^{1/3}\end{aligned}$$

where the WALE constant (C_w) is typically 0.2. The user though can modify this value in the *Project Tree*:

Project Tree > Environment > Engine > Turbulence model > C_w

Spalart-Allmaras

This model is only available in [Labs mode](#).

5.1.6 Turbulence generation

To define a given rate of turbulence generation, the following options are available:

- **Off:** Turbulence generation is not defined in the initial field or inlet boundary.
- **Automatic:** A turbulence intensity can be defined at the [inlet boundary](#) and to [initialize the internal field](#).
- **Custom:** This option allows the user to set the turbulence intensity, but also to customise the following parameters:
 - ↳ **Initial turbulence scale:** By default 1 m. Turbulence length scale should be a positive double.
 - ↳ **Number of harmonics:** By default 100. Harmonics should be a positive integer.

5.1.7 Acoustics analysis

Project Tree > Environment > Engine > Acoustics analysis (On/Off)

⚠ Please note: The **Acoustics analysis** option is only available in the [Expert mode](#) and is not compatible with the following options:

- Project Tree > Environment > Engine > Multiphase model > Phase field
- Project Tree > Environment > Engine > Thermal model > Supersonic

Acoustics analysis

The fluid solver of XFlow is compressible and it thus physically deals with sound pressure waves. Furthermore, the turbulence modeling approach used in XFlow, Large Eddy Simulation (LES), computes the flow fluctuating structures down to the resolution specified by the user. Hence, in XFlow, the acoustics approach is the one called: **Direct Computation of Sound** (DCS). The numerical algorithm does not introduce any artificial shear viscosity or stabilization.

Since the solver is transient and compressible and due to the nature of the lattice Boltzmann scheme employed in XFlow, the information (pressure waves) naturally travels at the numerical speed of sound $c_{numerical}$:

$$c_{numerical} = \frac{dx}{\sqrt{3} dt}$$

where dx is the resolution at a given lattice level, and dt the associated time step for the same lattice level.

Whereas $c_{thermodynamic}$ is the thermodynamic speed of sound calculated estimated by XFlow using:

- For single phase/multiphase (gas) flows: the ideal gas law for the single phase flows, with gamma the [adiabatic index](#), the [molecular weight](#) M , the perfect gas constant $R=8314$ and the [operating temperature](#) T :

$$c_{thermodynamic} = \sqrt{\frac{\gamma RT}{M}}$$

- For free surface/multiphase (liquids) flows: the input [speed of sound](#),

Therefore, when the Acoustics analysis is enabled, XFlow automatically sets the time step dt which satisfies the following relationship:

$$c_{numerical} = c_{thermodynamic}$$

Hence:

$$dt = \frac{dx}{\sqrt{3} c_{thermodynamic}}$$

⚠ Please note: Since the time step is unique with acoustics analysis, the frames frequency is adjusted automatically in order to get an integer number of the acoustics time step for each saved frame. A cap is also imposed since the data cannot be saved at a frequency higher than the one used by the solver (given by $1/\text{dt}$).

💡 Tip: The *Message View* will show the values of Numerical speed of sound and the Thermodynamic speed of sound when the acoustics analysis is enabled, make sure they are equal and equal to the expected fluid speed of sound.

Refractive index

By default it is set to 1. The refractive index is a dimensionless parameter n that allows to set a specific speed of sound in the fluid medium v_{medium} defined as following:

$$c_{\text{thermodynamics}} = n \cdot v_{\text{medium}}$$

⚠ Please note: The refractive index should always be equal or greater than 1 to ensure numerical stability, as the propagation of information is can only be decelerated and cannot be transported beyond the time step limit.

💡 Tip: The refractive index can be used to model two different speed of sound in two-phases simulations for instance, using a law as a function of the VoF field. It could also be used to model the change of speed of sound in hot and cold gases with the ideal gas law through the temperature field. In case such laws are implemented, make sure to use the highest speed of sound as your numerical speed of sound and use laws to define deceleration of the speed of sound as function of other fields (VoF, temperature, etc.).

Bulk viscosity

The acoustics analysis also enables the user to set up a constant value for the volume bulk viscosity (also called second viscosity), μ_v , to model the dissipation of pressure waves forced to travel at the real speed of sound. The value of the volume bulk viscosity can be defined in:

Project Tree > Materials > Fluid > Volume bulk viscosity

The bulk viscosity contributes to the viscous stress tensor, and thus to waves dissipation, as indicated in the following equation:

$$\tau_{ik}(\mu_v) = \mu_v \delta_{ik} \frac{\partial v_i}{\partial x_i}$$

Sound Pressure Level (SPL)

A useful output for the acoustics analysis is the **Sound Pressure Level**. By definition, the sound pressure is the deviation from the average pressure caused by a sound wave. The Sound Pressure Level (**SPL**) is a logarithmic measure of the effective sound pressure relative to a reference value. It is calculated in XFlow as:

$$SPL = 10 \log_{10} \left(\frac{p^2}{p_{ref}^2} \right)$$

with p being the fluctuating gauge static pressure measured in a probe or sensor, and p_{ref} the reference pressure equal to $2 \cdot 10^{-5}$ Pa (i.e. threshold of human hearing). It is measured in decibels (dB) above the reference level. The SPL can be plotted in the Function Viewer:

Function Viewer > (right click) > Data management > Set graph to [SPL vs freq] mode

The SPL is plotted in the frequency space obtained by Fourier transform.

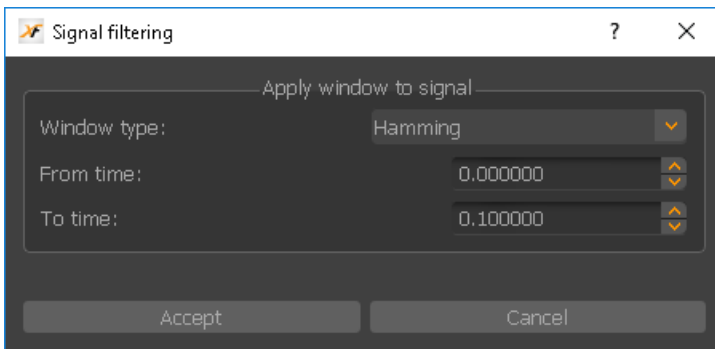
Power Spectral Density (PSD)

The Power Spectral Density describes the power of a signal as a function of frequency. It shows at which frequencies the signal is strong and weak. The Power Spectral Density E_x of a signal x is the square root of the module of its Fourier transform X divided by the total integration time T :

$$E_x = |X|^2/T$$

Apply window to signal

In spectral analysis, the Fourier transform can be multiplied by a window function which is a mathematical function that is zero-valued outside of some chosen interval. It is useful to filter out spurious frequencies due to applying the Fourier transform to a non-periodic signal. The following window will open:



↳ **Window type:** this is the window type to apply to the Fourier transform. The window functions available are: None, Hamming, Hann, Barlett, Blackman, flat top, Gaussian.

↳ **From time:** the start time of the window interval.

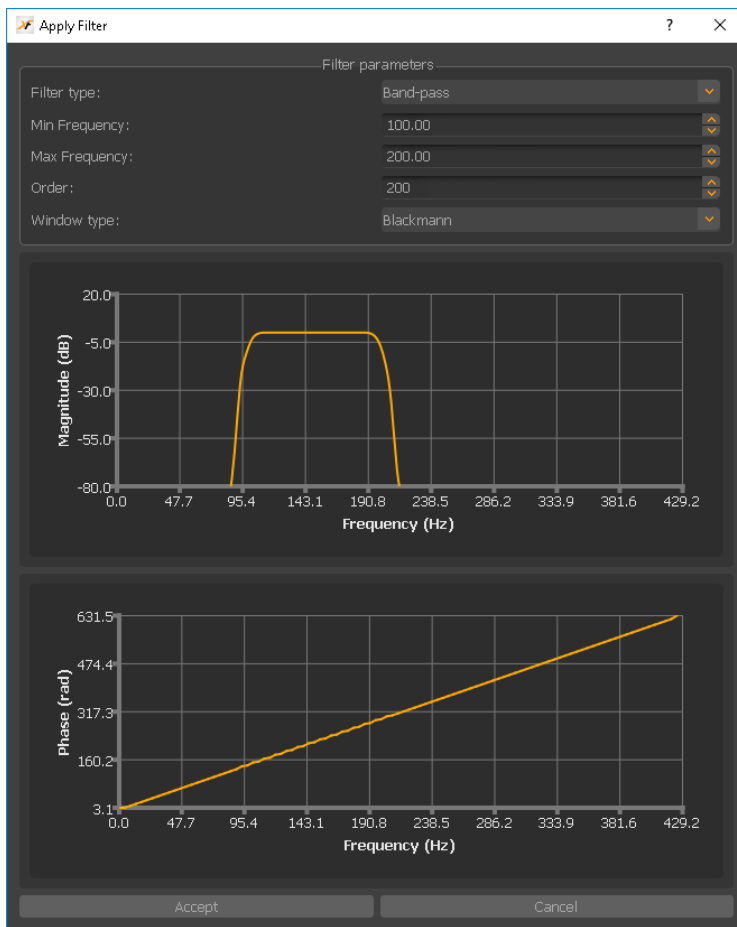
↳ **To time:** the end time of the window interval.

Tip: For a good signal processing, it is recommended to skip the transient period in the windows interval, thus use a "From time" value that is after the transient period.

Tip: The maximum frequency of Fourier transform is determined by the numerical output time step and is $f_{max} = 1 / dt$. You can adjust the [numerical data frequency](#) in order to have a larger spectrum. Furthermore, the larger the time range used for the Fourier transform and the smaller the frequency steps, therefore your spectrum becomes better detailed if using a larger time range.

Apply filter to signal

The Finite Impulse Response (FIR) filters of XFlow are very useful to remove, partially or entirely, the noise and undesirable oscillations from the signals displayed by the Function Viewer. They can be applied to any signal displayed in the Function Viewer. The following window will open when applying a filter to signal:



- ↪ **Filter type:** the type of the FIR filter to apply on the signal displayed in the Function Viewer. The filters available: Low pass, Band-pass, Band-stop and High pass.
- ↪ **Min Frequency:** minimum frequency of the filter in Hz.
- ↪ **Max Frequency:** maximum frequency of the filter in Hz.
- ↪ **Order:** this is the order of the filter transfer function polynomial used in the convolution operation between the filter and the time signal. The higher the order is, the more accurate would be the approximation of the transfer function polynomial of the filter. However, a higher order will introduce more delay to the filter response and the calculation of the filtered signal will take a longer time by the Function Viewer.
The delay τ introduced by the Order can be estimated (in seconds) with this formula: $(\text{Order} \times \text{Simulation time-step}) / 2$.
- ↪ **Window type:** this is the window type applied to the filtered signal. The window functions available are: Barlett, Blackman, Hamming, Hann.

The representations of the response in Magnitude (dB) and Phase (rad) of the selected filter are provided in the graphs of the Apply filter window in function of frequency.

⚠ Please note: The filter is always applied to the signal displayed in the function Viewer. Therefore, if the displayed signal has already been filtered, the filter will be applied twice.
In order to change the filter, it is always recommended to display the original signal again in the Function Viewer before changing the filter.

5.1.8 Scalar transport

Project Tree > Environment > Engine > Scalar transport

The scalar transport feature allows the user to solve an additional advection-diffusion equation for an arbitrary scalar Φ :

$$\frac{D\Phi}{Dt} = D\nabla^2\Phi$$

where D [$\text{m}^2 \text{s}^{-1}$] is the scalar diffusion coefficient (or **Diffusivity**) in the fluid.

The scalar is transported by the fluid phase and it can either actively affect the flow, or be set as a passive scalar. In the first case, the scalar could represent mist dispersed in air, while the passive case could stand, for instance, for the concentration of a dye or an indicator in the flow.

There are two ways to consider the scalar field:

1. Mass concentration of the specie in [kg m^{-3}]. The user input the actual mass concentration value and read the scalar as [kg m^{-3}] in XFlow.
2. Mass fraction of the specie (between 0 and 1). The user input and read fraction of mass from the scalar in XFlow, and must multiply it by its density to get the actual mass concentration in [kg m^{-3}].

To enable the scalar transport please switch **On** this option from the [Environment](#) tab; then add a scalar ([+info](#)) and the following data structure will then be added to the project tree:

Environment > Engine > Scalar transport - **On**:

- ↳ **Scalar - Diffusivity** ([+Info](#))
- ↳ **Scalar - Buoyancy**
 - **Off** The scalar is passive.
 - **On** The scalar is actively affect the flow.
 - ↳ **Density**: the density of the scalar [kg m^{-3}].

Environment > Environment > Global a. > Initial Conditions > Initial scalar field:

- ↳ **Scalar - Concentration** ([+Info](#))

Environment > Environment > Domain type > Boundary Conditions > Scalar concentrations:

- ↳ **Scalar -** ([+Info](#))

Geometry > Geometries > Shape > Boundary Conditions > Scalar concentrations:

- ↳ **Scalar -** ([+Info](#))

Post-Processing > General > Show > Volumetric field > visualization field

- ↳ **Scalar** ([+Info](#))

Add/Remove a scalar

To add a scalar, **right click** on **Scalars** and select the option **Add scalar**. A scalar defined by a diffusion coefficient (**Diffusivity**) will automatically appear below; by default the **Buoyancy** option is set to Off.

To remove a scalar, e.g. **Scalar 1**, **right click** on **Scalars > Scalar 1** and select the option **Remove scalar**. The scalar will be removed from the list.

Initialize a scalar

A [function](#) of space and time can be defined as the initial passive scalar law.

 **Please note:** The value of the scalar is in the range [0;1].

External-domain inlet boundary-condition for the scalar concentration

If the flow model is external, a [function](#) of space and time can be defined as boundary condition for the scalar at the external domain inlet.

 **Please note:** The value of the scalar is in the range [0;1].

Shapes boundary-conditions for the scalar concentration

If the Shape boundary is either Inlet or Outlet, a [function](#) of space and time can be defined as boundary condition for the scalar.

 **Please note:** The value of the scalar is in the range [0;1].

Visualize a scalar

Every scalar included in the simulation will appear in the list of [visualization fields](#). Thus, its post-processing can be done as that of any other field. It can be visualized through cutting planes, iso-surfaces, etc.

5.1.9 Advanced options

The Advanced options are only available in [Expert mode](#).

Project Tree > Environment > Engine > Advanced options

Several advanced options are accessible depending on the flow model selected:

High order boundary conditions

Allows the user to use second order boundary conditions instead of first order when imposing the velocity at a surface. Second order is more accurate but less robust. This option is disabled by default.

Force evaluation scheme

There are different schemes available to compute the overall forces exerted on bodies. The default scheme follows the momentum exchange approach which is more accurate for the fixed geometries. Alternatively, a direct surface stress integration is also available and is suited for moving geometries. The automatic mode applies the momentum exchange scheme on the fixed geometries and surface stress integration on the moving geometries, and is the recommended mode for most cases.

Structural analysis

Only available in [Labs mode](#).

For Fluid-Structure Interaction (FSI) simulations, co-simulation between XFlow and [Abaqus](#) or [MSC.Nastran](#) can be performed. Refer to the [Co-Simulation](#) chapter for more information.

Time integration scheme

5.1 Engine

Only available in [Labs mode](#).

Wall function time filter

Only available in [Expert mode](#) for: *Single phase*.

The **Wall function time filter** is effective if the **Automatic** wall function (**Enhanced wall function**) and the **Non-equilibrium wall function** are enabled for static walls.

This option allows to filter potential noises near walls by applying a time filter on the near walls velocity used in the wall functions. This time filter can be set as:

- **Off**: The time filter is disabled. XFlow uses the instantaneous velocities for the wall functions.
- **Automatic**: A temporal filter is applied on the instantaneous velocities. The automatic characteristic time used by XFlow is calculated with the Reference length and the Reference velocity:
Characteristic time = L_{ref} / V_{ref}
- **Custom**:
 - ↳ **Characteristic time**: By defaults set as 1 s.
 - In order to set a custom value of the **Characteristic time**, this is equation used to compute the **Wall function time filter**:

$$V_{wall} = W_1 \cdot v + W_2 \cdot V_{mean}$$

Where:

V_{wall} is the velocity near walls used for the wall functions,

v the instantaneous velocity near walls,

V_{mean} the time averaged velocity near walls,

And W_1 and W_2 are weights defined as following:

- $W_1 = \text{Local time-step} / \text{Characteristic time}$ if: Characteristic time > Local time-step
= 1 if: Characteristic time = Local time-step
- $W_2 = 1 - W_1$

The local time-step corresponds to the time step at the lattice level where the wall function is applied. For more information about the time-step structure please refer to the section [Time](#).

 **Please note:** The setting $W_1 = 1$ (Characteristic time = Local time-step) is the same than setting **Wall function time filter: Off**.

Enforce incompressible model

Allows the user to enable or disable an approach of the incompressible Navier-Stokes equations. It is recommended to activated in applications where the compressibility effects are undesirable (i.e. liquid as material).

Moving parts surface normal detection

This option allows the solver to use the moving geometries normals in order to reduce the computational effort when simulating enforced behaviours. It should be used when the enforced geometry is properly defined as a watertight object.

Volume correction

When this parameter is enabled the software enforces the conservation of volume for each phase present in the simulation. Note that the default particle-based tracking scheme used by the free surface and [particle-based tracking](#) multiphase solvers does not exactly preserve volume. The corrections are performed by adding additional forces/displacements on the surface trackers, which achieves a global conservation but may create artifacts locally. The [phase field](#) multiphase solver does enforce volume conservation both locally and globally and does not require such corrections. Only available for: *Free surface and Multiphase*.

 **Please note:** This might slightly affect the stability and require a lower timestep.

Damping outlet region

When enabled, this option introduces a porous media at the outlet (a sort of artificial beach) to reduce the kinetic energy of the fluid so that the pressure imposed at the outlet is consistent with the progressive waves. Only available for: *Free surface external*.

Enable viscous term in Energy equation

Allows the user to enable or disable the viscous term in the energy equation. Only available for: *Segregated energy, Coupled energy and Supersonic*.

5.2 Environment

Project Tree > Environment > Environment

The **Environment** section consist of one or several of the following folders, depending on the **Flow model** chosen in the [Engine](#) section:

Global attributes	Domain type (+Info)	Virtual wind tunnel Generic rectangular domain
	Gravitational potential (+Info)	
	Ext. acceleration laws (+Info)	
	Volumetric heat source (+Info)	
	Initial conditions (+Info)	Wind tunnel default Water channel default User defined Simulation data Automatic
	Reference length (+Info)	X bounding box Y bounding box Z bounding box Custom
	Reference area (+Info)	Front Top Side

		Custom
	Reference velocity (+Info)	Automatic Local Custom
Wind tunnel (+Info)	Position	
	Dimensions	
	Ground wall	
	Lateral boundaries	Periodic Symmetric
	Boundary conditions	Velocity Inlet turbulence intensity (+Info)
	Thermal boundary conditions	Adiabatic Temperature
Rectangular domain (+Info)	Position	
	Dimensions	
	X periodic	
	Y periodic	
	Z periodic	
Water channel (+Info)	Position	
	Dimensions	
	Velocity laws	
	Water initial surface	
	Water inlet wave function	
	Thermal boundary conditions ⁽⁶⁾	Adiabatic Temperature Heat flux
	Channel walls	
Liquid regions (+Info)	Initial liquid function	

5.2.1 Domain type

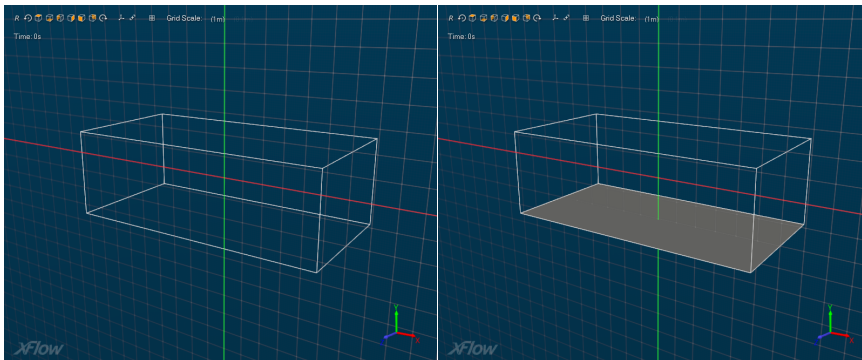
Project Tree > Environment > Environment > Global attributes > Domain type

To study the fluid dynamics of a monophasic fluid around a body (Single Phase External), the user has to define the body geometry and the external limits of the fluid domain. To this end, the user has the following options:

- [Virtual wind tunnel](#)
- [Generic rectangular domain](#)

Please note: When importing a geometry into XFlow external simulations, by default, if the geometry bounding box is not contained in the fluid domain, this will adapt its dimensions in order to contain the geometry. In order to modify this behaviour, please change it in [Preferences > Geometry](#).

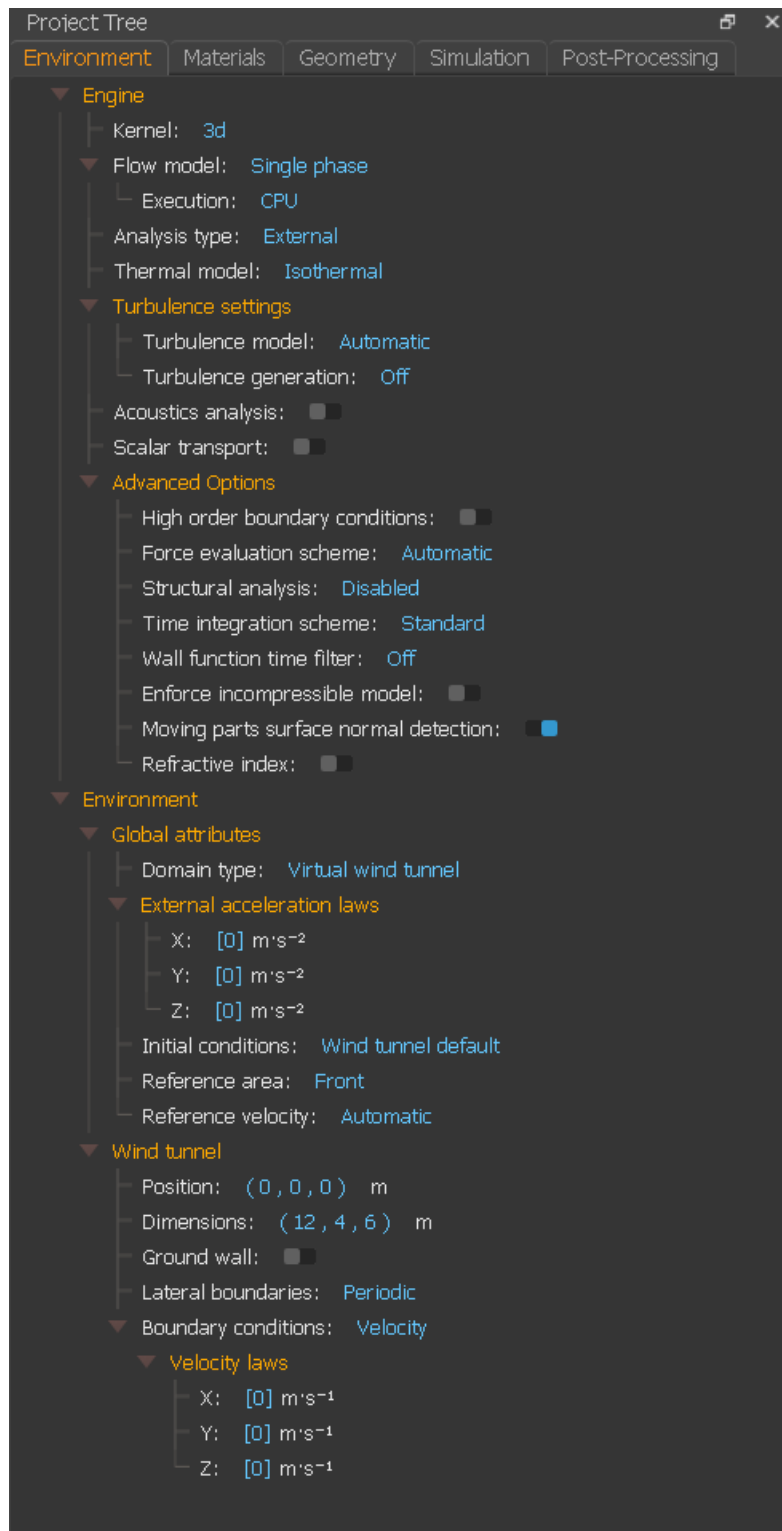
The domain is identified, in either case, as a wire box in the *Graphic View*.



Wind tunnel with ground wall disabled (left) and enabled (right)

Virtual wind tunnel

The virtual wind tunnel facilitates the setup by predefining automatically some of the boundary conditions to perform a wind-tunnel-like external aerodynamics analysis. By default, the wind profile is assumed to be aligned with the X-axis (from -X to +X), while the Y-axis is assumed to be vertical. The wind tunnel options are:



Project Tree > Environment > Environment > Wind tunnel

👉 **Position:** Position coordinates (x,y,z) are those of the wind-tunnel centre.

👉 **Dimensions:** The dimensions of the domain are given by the length (x), height (y) and width (z).

↳ **Ground wall:**

- **Off:** -Y and +Y boundaries are set to `Periodic`.
- **On:** +Y boundary is set as the inlet boundary condition (velocity profile or pressure); -Y boundary is the so called Ground wall and is colored in gray in the *Graphic View*; -Y boundary options are:
 - ↳ **Ground wall type:** Wall-boundary type, see [Wall types](#).
 - ↳ **Ground wall velocity law X:** If moving wall, the velocity in X-direction can be defined.

↳ **Lateral boundaries:** by default, Z+ and Z- are `Periodic` boundaries; `Symmetric` type is also available.

↳ **Boundary conditions:** The -X +X boundary condition:

- ↳ **Velocity > Velocity laws:** In -X boundary, the wind velocity profile is defined and the reference pressure is set to +X boundary

 **Please remember:** Squared-brackets fields can be defined using a [function](#). It is thus possible to setup varying conditions.

In either cases, the turbulence intensity can be defined in the -X boundary condition:

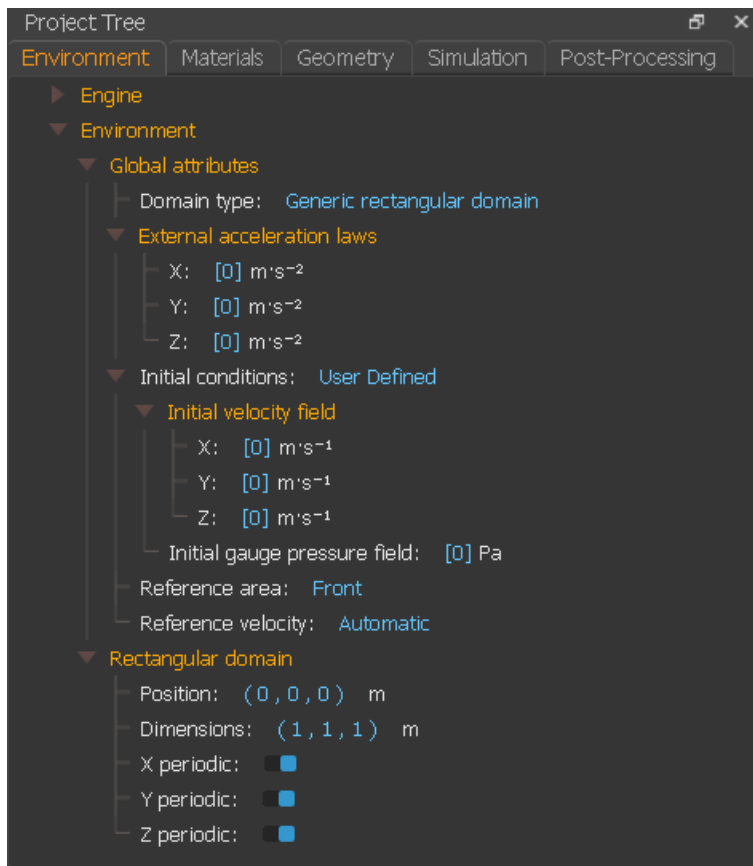
- ↳ **Inlet turbulence intensity:** To set the inlet (-X) turbulence intensity. It must be given as a percentage (%).

↳ **Thermal boundary conditions:** For non isothermal cases, the wind-tunnel thermal boundary-conditions are automatically defined as follows:

- **Adiabatic:** If **Ground wall** is **Off**, +Y, -Y thermal boundaries are `Periodic` and +X,-X boundaries are `Adiabatic`. If **Ground wall** is **On**, +Y, -Y, +X,-X boundaries are set to `Adiabatic`. Thermal boundary condition at +Z and -Z is also given by **Lateral boundaries** (`Periodic` or `Symmetric`)
- **Temperature** If **Ground wall** is **Off**, +Y, -Y thermal boundaries are `Periodic` and the temperature at +X,-X boundaries is given by the user-defined **Temperature law**. If **Ground wall** is **On**, -Y boundary is set to `Adiabatic` while the temperature at +X,-X,+Y is given by the user-defined **Temperature law**. Thermal boundary condition at +Z and -Z is given by **Lateral boundaries** (`Periodic` or `Symmetric`)

Generic rectangular domain

The Generic rectangular domain stands for a right tetragonal prism defined by its position and dimensions; every boundary of which can be defined by the user.



Project tree > Environment > Environment > Generic rectangular domain

✚ **Position:** Position coordinates (x,y,z) are those of the prism centre.

✚ **Dimensions:** Dimensions given by the length (x), height (y) and width (z).

✚ **X periodic:**

- **On:** Both -X and +X boundaries are set to *Periodic*
- **Off:** -X and +X can be any of the available boundary conditions (see [Boundary Conditions](#)). For non isothermal cases, thermal boundary conditions must also be defined ([Thermal Boundary Conditions](#)).

✚ **Y periodic:**

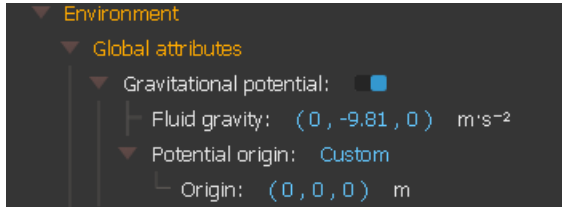
- **On:** Both -Y and +Y boundaries are set to *Periodic*
- **Off:** -Y and +Y can be any of the available boundary conditions (see [Boundary Conditions](#)). For non isothermal cases, thermal boundary conditions must also be defined ([Thermal Boundary Conditions](#)).

✚ **Z periodic:**

- **On:** Both -Z and +Z boundaries are set to *Periodic*
- **Off:** -Z and +Z can be any of the available boundary conditions (see [Boundary Conditions](#)). For non isothermal cases, thermal boundary conditions must also be defined ([Thermal Boundary Conditions](#)).

5.2.2 Gravitational potential

Project Tree > Environment > Environment > Global attributes > Gravitational potential



This option is only available for Free surface flows, either external or internal.

The **Gravitational potential** allows the user to take into account the effect on the fluid of a body force; this being expressed as a force derived from a potential function (ϕ):

$$\rho \vec{f}_m = -\rho \nabla \phi$$

If the density variation produced by the body force is negligible (incompressible), the Navier-Stokes equation incorporating the body force can be expressed in the same form as in the absence of the body force but considering an altered pressure, p^* , given by:

$$p^* = p + \rho \phi$$

The user may activate/deactivate this option in the *Project Tree*:

- **Off:** Pressure field = p . The hydrostatic pressure is not considered in the pressure field.
- **On:** Pressure field = p^* . The pressure variable includes the hydrostatic pressure, thus initial pressure field is automatically initialized according to the hydrostatic pressure.
 - ↳ **Fluid gravity:** By default, the value of the gravity of earth (0, -9.81, 0). The user can though specify another acceleration.
 - ↳ **Potential origin:** To setup the reference point where the potential is zero, $\phi = 0 \Rightarrow p^* = p$; there are two options:
 - **Automatic:** Potential origin is set at the free surface.
 - **Custom:** The user can specify the coordinates of the potential origin.

⚠ Please note: If **Gravity potential** = **On**, the initial and boundary conditions for pressure are applied to p^* , but the resulting pressure field from the simulation (numerical data), which is used for the post-processing, corresponds to p (gauge pressure).

By default, the **Gravitational potential** is switched **Off** for Internal Free Surface flows and it is switched **On** for External Free Surface flows.

Example:

For the gravity of earth: $\vec{f}_m = \vec{g} = -g\vec{z}$; $\vec{f}_m = -\nabla\phi \Rightarrow \nabla\phi = g\vec{z} \Rightarrow \phi = gz$

5.2.3 External acceleration laws

Project Tree > Environment > Environment > Global attributes > Ext. acceleration laws

External accelerations applied to the fluid can be here defined by its components: X, Y, Z (m s⁻²); as for example: the gravity.

 **Please note:** The external acceleration is applied to the fluid only, not the solid dynamics. For example, for the [rigid body dynamics](#), the gravity should be defined in **Behavior: Rigid body dynamics > Constraints > External forces**.

5.2.4 Initial conditions

Project Tree > Environment > Environment > Global attributes > Initial conditions

The initial field can be defined as follows:

- **Wind tunnel default:** Only available if Single Phase External flow and Wind tunnel (domain type). This option sets the internal field of the velocity and turbulence intensity equal to their corresponding values at the (inlet) boundary condition of the [wind tunnel](#). The gauge pressure is automatically initialized as a constant field of value zero for isothermal and segregated energy meanwhile it is set to the absolute reference pressure value in coupled energy and supersonic [thermal models](#).
- **Water channel default:** Only available if Free Surface External flow. This option sets the velocity internal field according to the velocity law given in the [water channel](#) section. The pressure, though, is initialized to zero by default.
- **User defined:** the computation is initialized by the user.
 - ↳ **Initial velocity field :** Every component of the initial velocity field is to be defined.
 - ↳ **Initial gauge pressure field:** The gauge pressure at the initial time is to be defined.
 - ↳ **Initial temperature field:** If non-isothermal case, the initial temperature has to be defined.
 - ↳ **Initial turbulence intensity:** To define the initial turbulence intensity of the initial internal field. This option is available if the *Flow Model* is *Single Phase*.

 **Please remember:** Squared-brackets fields can be defined using a [function](#).

- **Simulation data:** the computation is initialized from results of a previous XFlow computation:
 - ↳ **Folder:** Path of the folder containing the results as it is seen by the machine where the simulation will be run.
 - ↳ **Initial data frame:** Frame number of the simulation defined in **Folder** to use to initialize the case.

 **Tip:** Multiphase simulations can use Free Surface simulations to initialize the liquid region, and conversely.

- **Automatic:** Only available in [Expert mode](#).
The automatic initialisation option performs a number of steps previous to the actual simulation using a coarser grid. In each step, the domain is initialized using the data obtained in the previous step, emulating a multigrid algorithm.

The parameters the user can set for the automatic initialisation are:

↳ **Initialisation steps:** Number of steps to be performed before running the simulation.

↳ **Steps to skip:** Resolution change between steps. If it is 0, the resolution for the next step is that of the current step divided by 2. If it is 1, the resolution is divided by 4. In general:

$\text{Resolution_@Step}(n+1) = \text{Resolution_@Step}(n) / (2^{(\text{steps_to_skip} + 1)})$

The change in resolution between steps affects every resolution parameter: resolved scale, wake resolution, shapes target resolved scales and regions target resolved scales.

↪ **Total time fraction:** Fraction of the [simulation time](#) used as simulation time for the initialisation steps. This is useful because the time to converge in the initialisation steps is expected to be smaller than in the actual simulation.

Find below an example illustrating how the automatic initialisation works:

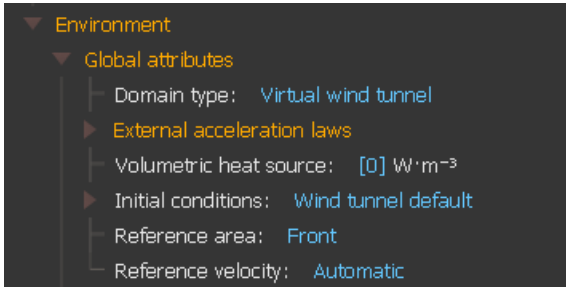
Example:	
Simulation Setup:	Initialisation steps:
Project Tree > Environment > Environment > Global attributes > Initial conditions > Automatic > Initialisation steps: 4 > Steps to skip: 0 > Total time fraction: 0.5 Project Tree > Simulation > Simulation time: 1s Project Tree > Simulation > Resolution > > Resolved Scale: 0.2 > Wake Resolution: 0.1 > Shape > Target resolved scale: 0.025	Step 1: Simulation time: 0.99s Resolved Scale: 0.4 Wake Resolution: 0.4 Shape > Target resolved scale: 0.4 Step 2: Simulation time: 0.5s Resolved Scale: 0.2 Wake Resolution: 0.2 Shape > Target resolved scale: 0.2 Step 3: Simulation time: 0.5s Resolved Scale: 0.2 Wake Resolution: 0.1 Shape > Target resolved scale: 0.1 Step 4: Simulation time: 0.5s Resolved Scale: 0.2 Wake Resolution: 0.1 Shape > Target resolved scale: 0.05 Final step (actual simulation, parameters match the GUI setup): Simulation time: 1s Resolved Scale: 0.2 Wake Resolution: 0.1 Shape > Target resolved scale: 0.025

 **Please note:** In the first initialisation step, the simulation time is 0.99s instead of being 0.5s as it should correspond after applying the total time fraction parameter. This is because, in wind tunnel simulations, the simulation time for the first step is set to $1.1 \times (\text{time it takes for the fluid to cross the tunnel from } -X \text{ to } +X)$, which in the example turns out to be 0.99s for the given tunnel length and inlet fluid velocity.

 **Please note:** If this option is combined with [initialize domain with vorticity data](#), at each initialisation step the vorticity of the previous one is taken as refinement criterion for the domain generation.

5.2.5 Volumetric Heat Source

Project Tree > Environment > Environment > Global attributes > Volumetric heat source



This option is only available in [Labs mode](#) for Single Phase flows with the [Segregated Energy](#) thermal model activated.

The **Volumetric Heat Source** allows the user to define the amount of energy that enters or leaves the system per volumetric unit at any point:

$$\dot{q}_v = \rho c_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T)$$

For this, the user can specify a constant Volumetric Heat Source in Watts per cubic meter throughout the entire domain, or limit it to a region by defining a function.

5.2.6 Reference length

The Reference length feature is only available when the follow options are selected ([Expert mode](#)):

- [Turbulence generation](#): Automatic
- [Wall function time filter](#): Automatic
- [LODI](#): On

To define a given rate of turbulence generation, the following options are available:

- **X/Y/Z bounding box**: XFlow takes the X/Y/Z dimension of global bounding box of the geometries.
- **Custom**: This option allows the user to set a custom reference length in meters. by default, the reference length is 1 m.

⚠ Please note: If there are no geometries in the domain and the Reference length is set as X/Y/Z bounding box, XFlow takes the X/Y/Z dimension of the domain.

 **Tip:** The Reference length value is displayed in the *Message View*:

```
Computing boundary conditions map!
-----
Coarsest resolved length: 0.1
-----
Reading domain data...
Reading domain structure with 1 levels...
  Level 0 has 4800 active elements
Processing domain data...
Domain loaded successfully
Thermodynamic speed of sound: 340.112
Reference area: 480 m^2
Reference length: 1.2 m
Reference velocity: 10 m/s
Time step (level 0): 0.00057231 s
-----
Saving data...
[[Data file]] 0 done!!! | Frame wall clock time[0]s | Overall wall clock time[0]s | Num elements[4800]
```

5.2.7 Reference area

Project Tree > Environment > Environment > Global attributes > Reference area

Reference area is used to compute the [aerodynamic coefficients](#).

f

Reference area can be set to:

- **Front:** Projected area of the geometry in plane YZ (see [Geometrical properties](#) > projected areas),
- **Top:** Projected area of the geometry in plane XZ
- **Side:** Projected area of the geometry inplane XY
- **Custom:** Defined by the user in:
 - ↳ **Reference area value:** in m²

 **Please note:** If there are no geometries in the domain and the Reference area is set as Front/Top/Side, XFlow will take the Front/Top/side area of the domain.

 **Please note:** XFlow cannot estimate the area correctly, if the geometry object is bigger than the computational domain. In this case it is recommended to set a custom reference area.

 **Tip:** The Reference area value is displayed in the *Message View*:

```
Computing boundary conditions map!
-----
Coarsest resolved length: 0.1
-----
Reading domain data...
Reading domain structure with 1 levels...
  Level 0 has 4800 active elements
Processing domain data...
Domain loaded successfully
Thermodynamic speed of sound: 340.112
Reference area: 480 m^2
Reference length: 1.2 m
Reference velocity: 10 m/s
Time step (level 0): 0.00057231 s
-----
Saving data...
[[Data file]] 0 done!!! | Frame wall clock time[0]s | Overall wall clock time[0]s | Num elements[4800]
```

5.2.8 Reference velocity

Project Tree > Environment > Environment > Global attributes > Reference velocity

Reference velocity is used to compute the [aerodynamic coefficients](#). It can be set to:

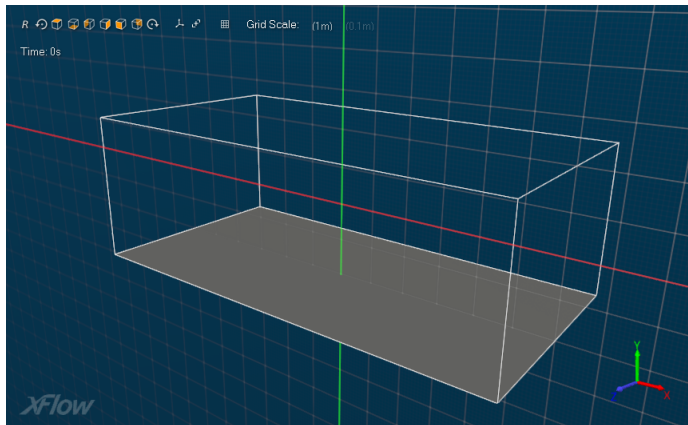
- **Automatic:** the reference velocity is automatically estimated by XFlow as the inlet velocity.
- **Local:** It corresponds to the velocity at the wall.
- **Custom:** It is defined by the user in:
 - ↳ **Reference velocity value:** in m s^{-1}

 **Tip:** The Reference velocity value is displayed in the *Message View*:

```
Computing boundary conditions map!
-----
Coarsest resolved length: 0.1
-----
Reading domain data...
Reading domain structure with 1 levels...
  Level 0 has 4800 active elements
Processing domain data...
Domain loaded successfully
Thermodynamic speed of sound: 340.112
Reference area: 480 m^2
Reference length: 1.2 m
Reference velocity: 10 m/s
Time step (level 0): 0.00057231 s
-----
Saving data...
[[Data file]] 0 done!!! | Frame wall clock time[0]s | Overall wall clock time[0]s | Num elements[4800]
```

5.2.9 Water channel

Project Tree > Environment > Environment > Water channel



Free Surface External domain - Water channel

The water channel represents the external domain of the fluid for **Free surface external** cases. The water channel assumes that the flow is aligned with the X-axis (from -X to +X), while the Y-axis is the vertical direction; it further predefines some boundary conditions by default:

- - X: Liquid inlet. Velocity can be defined by the user.
- +X: Liquid outlet, where the reference pressure is set.
- - Y: Ground wall. Wall-type boundary can be defined by the user.
- +Y: Top. Liquid-inlet velocity-law is applied to this plane.
- - Z, +Z : Periodic boundary, to lower the perturbations due to the finite dimensions of the water channel.

The water-channel customizable options are:

- ↳ **Position:** Position coordinates (x,y,z) are those of the water-channel centre.
- ↳ **Dimensions:** The dimensions of the domain are given by the length (x), height (y) and width (z).
- ↳ **Velocity laws:** The liquid inlet velocity is here defined by its components (x,y,z).
- ↳ **Water initial surface:** The shape of the initial surface is here defined by the user. Please see [Waves](#) or [Liquid regions](#).
- ↳ **Water inlet wave function:** To describes the free surface level at the inlet (-X boundary). It can be constant or vary over time (see [Waves](#)).
- ↳ **Fluid 1 concentration:** To set up the initial concentration of the fluid 1 at the initial region. The concentration C_1 of is a number between 0 and 1.
 - If $C_1 = 0$, there is no fluid 1 initially.
 - If $C_1 = 1$, all the domain is filled by the fluid 1.
 - The concentration of the fluid 2 corresponds to $C_2 = 1 - C_1$

The initial concentration of fluid 1 can be associated with spatial and time variable to set it at the a specific region of the domain.

↳ **Fluid 1 inlet concentration:** To set up the fluid 1 concentration of the inlet flow. If C_1 is the part of fluid 1 from the inlet flow, $C_2 = 1 - C_1$ will be the part of the fluid 2 from the inlet flow.

↳ **Thermal boundary conditions:** For non isothermal cases, the water channel thermal boundary conditions are here determined; possible options are:

- **Adiabatic:** +Y, -Y, +X, -X boundaries are set to `Adiabatic`; +Z, -Z boundaries are always `Periodic`.
- **Temperature** -Y boundary is set to `Adiabatic`; the temperature at +X, -X, +Y is given by the user-defined **Temperature law**. The thermal boundary condition at +Z and -Z is `Periodic`.

↳ **Channel walls:** the water channel walls boundaries are defined by the following options:

- ↳ **Lateral walls:** If this option is Off, only ground wall (-Y boundary) is enabled. If this option is On, walls are applied on the ground boundary (-Y) as well as on the lateral boundaries (-Z and +Z).
- ↳ **Channel wall type:** Wall-boundary type applied to all channel walls, see [Wall types](#).
- ↳ **Channel wall velocity law X:** If moving wall, the velocity in X-direction can be defined. This velocity is applied to all walls (ground wall and laterals walls if switched on).

It is possible to set progressive waves boundary conditions with the help of a wizard.

5.2.9.1 Waves

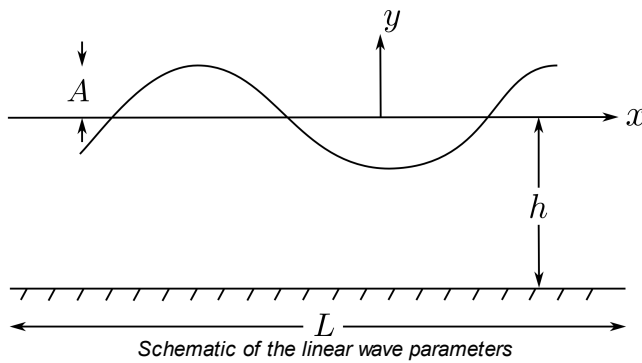
Main menu > Options > Setup progressive wave boundary conditions

XFlow contains a wizard to set progressive waves in the free-surface cases. The wizard parameters are:

- Type of wave:
 - [Linear](#)
 - [Fifth order Stokes](#)
- Wave definition
 - Water channel dimensions: Length (L), height, width;
 - Liquid depth (h)
 - Current velocity (v_w)
 - Progressive wave: amplitude (A) and frequency (f)

The wizard automatically calculates the wave velocity (**Water channel > Velocity laws**) and consistent laws to describe the initial free surface (**Water channel > Water initial surface**) and the inlet wave function (**Water channel > Water inlet wave function**)

Linear wave



From the linear wave theory, the **Velocity law** reads:

$$v_x = v_w + \frac{A\omega}{\sinh kh} \cosh k(h+y) \sin(\omega t - kx)$$

$$v_y = \frac{A\omega}{\sinh kh} \sinh k(h+y) \cos(\omega t - kx)$$

where x is the horizontal position and y the vertical one, t is the time, g is the gravity, ω is the angular velocity ($\omega=2\pi f$) and k is the wave number, given by the following equation:

$$\omega^2 = g k \tanh(kh)$$

Consistently, the **Water initial surface** is given by:

$$\text{Initial water surface} = -A \sin(kx)$$

$$\text{Water inlet wave function} = A \sin(\omega t + k\frac{L}{2})$$

The position of the channel changes automatically so that the mean free surface level is located at $y=0$. Therefore, $\text{position}Y = \text{height}/2\text{-depth}$.

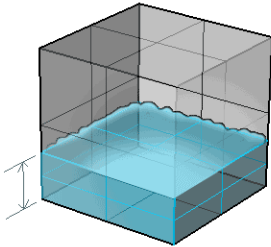
Fifth order Stokes

For details on the theory of the fifth order Stokes expansion of small-amplitude waves ($A \ll h$), the reader is referred to:

"Stokes' expansion of internal deep water waves to the fifth order", Y. Tsuji and Y. Nagata, Journal of the Oceanographical Society of Japan, vol 29, pp 61-69, 1973.

5.2.10 Liquid regions

Project Tree > Environment > Environment > Liquid regions



Liquid regions

When performing an **Internal** analysis with the **Multiphase** or **Free Surface** solvers, the user is required to define the initial liquid region by means of a [function](#). Some examples are given below:

Examples:	
	Initial liquid function: $y < 0.5$
	Initial liquid function: $(y < 0.5)(z < 0.5)$
	Initial liquid function: $\text{if } ((x < 0.5)(y < 0.5)(z < 0.5), 1, 0)$

Coordinates (x,y,z) refer to the global coordinate system, not relative to the object.

⚠ Please note: For **Free Surface External**, the initial liquid region has also to be defined, but in:
[Project Tree > Environment > Environment > Water channel > Water initial surface](#)

⚠ Please note: For Multiphase flows, the Fluid 1 regions are defined instead and the complementary volume is filled with Fluid 2.

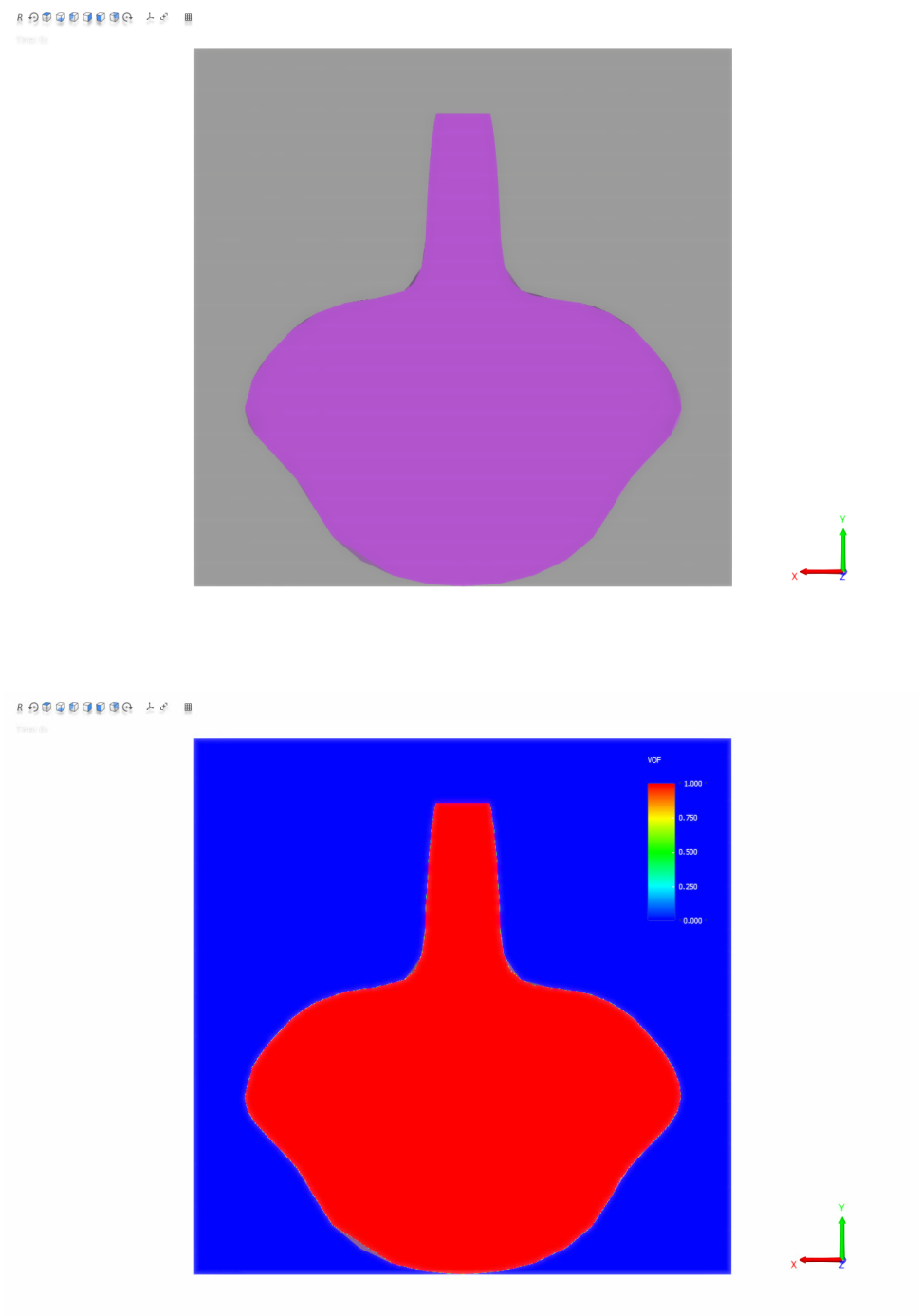
An additional option is to use the **Initialization function in arbitrary geometries** functionality. This feature is handy when the user would like to initialize the Initial Liquid Region within a specific geometry that is to be provided.

For this, you can define the following function:

```
insidegeometry("geometry")
```

Where, "geometry" corresponds to the specific geometry you would like to initialize from.

In the example below, where the purple geometry is the one in which we would like to initialize the liquid region, this function



5.3 Materials

Project Tree > Materials

The thermo-physical properties of the fluid can be specified in this tab of the *Project Tree*, according to the following hierarchy:

Fluid	Name (+Info)	
	Type (+Info)	Gas Liquid
	Molecular weight (+Info)	
	Speed of sound (+Info)	
	Reference density (+Info)	
	Operating temperature (+Info)	
	State Equation (+Info)	Reference density Reference static pressure Boussinesq Incompressible
	Viscosity model (+Info)	Newtonian Newtonian powerlaw Newtonian Sutherland Non-Newtonian Cross Non-Newtonian Herschel-Bulkley Non-Newtonian powerlaw Non-Newtonian Carreau Non-Newtonian user defined
	HTC Estimation (+Info)	
	Specific heat capacity (+Info)	
	Thermal conductivity (+Info)	
	Adiabatic index (+Info)	
	Reference pressure (+Info)	
Interactions	Surface tension model (+Info)	
	Interface thickness (+Info)	

5.3.1 Name

Project Tree > Materials > Fluid > Name

Fluid name contains a string with the name of the fluid. By default, it is set to **Material 1**.

5.3.2 Type

Project Tree > Materials > Fluid > Type

Fluid type defines if the fluid is a **Gas** or a **Liquid**. This option appears only for multiphase flows, since the single phase considers the fluid as a gas and the free surface as a liquid by default. According to the fluid type, the material properties required are different.

5.3.3 Molecular weight

Project Tree > Materials > Fluid > Molecular weight

This field is active only in the case of Single phase flows. It is required to specify the molecular weight of the fluid (i.e. gas), used to compute the speed of sound through the ideal gas law:

$$C_{thermodynamic} = \sqrt{\frac{\gamma RT}{M}}$$

The value of the air molecular weight (28.996 g/mole) is set by default; it can be modified by the user.

5.3.4 Speed of sound

Project Tree > Materials > Fluid > Speed of sound

This field is active only in the case of free surface or multiphase flows involving liquids, when acoustics mode is enabled. It is required to define the speed of sound of the liquid phase since the ideal gas law does not apply anymore. The default value is set to 1484 m/s which corresponds to the water speed of sound.

5.3.5 Reference density

Project Tree > Materials > Fluid > Reference density

This is the reference density of the fluid used in the simulation.

The user can modify the value, which by default is:

- For gases (i.e. Single Phase): 1.225 kg/m³ (air density)
- For liquids (i.e. Free Surface): 998.3 kg/m³ (water density)

5.3.6 Operating temperature

Project Tree > Materials > Fluid > Operating temperature

The operating temperature is useful to compute the speed of sound through the ideal gas law for single phase analysis:

$$C_{thermodynamic} = \sqrt{\frac{\gamma RT}{M}}$$

Since XFlow solver works internally with non-dimensional values, therefore this value is also used to make temperature units non-dimensional.

The user can modify the value, which by default is 288.15 K (15 °C).

5.3.7 State equation

Project Tree > Materials > Fluid > State Equation

This option is only available for non-isothermal cases. According to the **Flow model**, the following state equations can be chosen:

- Single Phase/Multiphase (gas)
 - [Reference density](#)
 - [Reference static pressure](#)
 - [Boussinesq](#)
- Free Surface/Multiphase (liquid)
 - [Incompressible](#)
 - [Boussinesq](#)

Reference density

The ideal gas equation is used, taking as a reference density the one specified by the user in:

Project Tree > Materials > Fluid > State Equation > **Reference density** > Density

Reference static pressure

The ideal gas equation is used, taking as reference the operating pressure given by the user in:

Project Tree > Materials > Fluid > State Equation > **Reference static pressure** > Operating pressure

Boussinesq

Density fluctuations are modelled according to Boussinesq approximation:

$$\rho = \rho_0 - \alpha \rho_0 \Delta T$$

where α is the thermal expansion coefficient and ρ_0 is the reference density.

Project Tree > Materials > Fluid > State Equation > **Boussinesq** > Density

Project Tree > Materials > Fluid > State Equation > **Boussinesq** > Thermal expansion coefficient

Incompressible

The specified density is defined and considered constant, using a stiff state equation:

Project Tree > Materials > Fluid > State Equation > **Incompressible** > Density

5.3.8 Viscosity models

Project Tree > Materials > Fluid > Viscosity model

XFlow provides several options to model the dynamic viscosity of the fluid. The main classification of these models considers the fluid behaviour:

- [Newtonian fluid](#)
- [Non-Newtonian fluid](#)

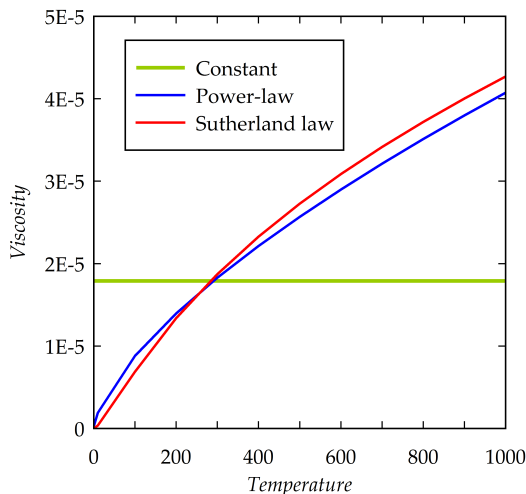
5.3.8.1 Newtonian fluid

In a Newtonian fluid, the relationship between the shear stress τ and the strain rate D is linear:

$$\tau = \mu D, \quad D = \nabla \mathbf{u} + \nabla^T \mathbf{u}$$

the coefficient of proportionality being the dynamic viscosity (μ).

As shown in the figure below, the dynamic viscosity of a Newtonian fluid may be constant or vary with the fluid temperature, θ ([theta](#)).



Newtonian viscosity models, temperature dependency.

XFlow provides the following viscosity models for Newtonian fluids:

- ↳ Temperature independent
 - [Newtonian](#)
- ↳ Temperature dependent
 - [Newtonian powerlaw](#)
 - [Newtonian Sutherland](#)

Newtonian model:

The dynamic viscosity is assumed constant; its value being defined by the user:

Project Tree > Materials > Fluid > Viscosity model > Dynamic viscosity

For *Single Phase* flows of Newtonian fluids, if [Acoustics analysis](#) is activated, the bulk viscosity can be further specified by switching on the following feature:

Project Tree > Materials > Fluid > Viscosity model > Enable bulk viscosity

Newtonian power-law model

The power law models the dynamic viscosity according to the following expression:

$$\mu = \mu_0 \left(\frac{\theta}{\theta_0} \right)^n$$

where n is the power-law index, θ_0 is the reference temperature (in K) and μ_0 is the reference viscosity (in Pa·s); the values of which have to be specified in:

Project Tree > Materials > Fluid > Viscosity model > Powerlaw index (n)

Project Tree > Materials > Fluid > Viscosity model > Reference temperature (θ_0)

Project Tree > Materials > Fluid > Viscosity model > Reference viscosity (μ_0)

Newtonian Sutherland model

Sutherland's law relates the dynamic viscosity and the absolute temperature of an ideal gas. The formula is based on kinetic theory of ideal gases and an idealized intermolecular-force potential:

$$\mu = \mu_0 \left(\frac{\theta}{\theta_0} \right)^{3/2} \frac{\theta_0 + C}{\theta + C}$$

where μ_0 is the reference viscosity (in Pa·s), θ_0 is the reference temperature (in K) and C is the Sutherland constant (in K); the values of which have to be specified in:

Project Tree > Materials > Fluid > Viscosity model > Reference viscosity (μ_0)

Project Tree > Materials > Fluid > Viscosity model > Reference temperature (θ_0)

Project Tree > Materials > Fluid > Viscosity model > Sutherland constant (C)

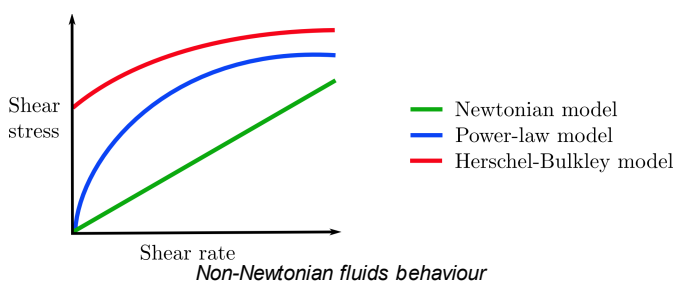
5.3.8.2 Non-Newtonian fluid

In a non-Newtonian fluid, the relation between the shear stress and the strain rate is nonlinear, and can even be time-dependent. Viscosity can be considered to depend on the temperature, θ ([theta](#)), and the shear rate, $\dot{\gamma}$ ([gamma](#)):

$$\mu = f(\dot{\gamma}, \theta)$$

For non-Newtonian fluids, XFlow provides the following viscosity models:

- Temperature independent
 - [Non-Newtonian Herschel-Bulkley](#)
- [Non-Newtonian Cross](#)
 - [Non-Newtonian user defined](#)
- Temperature dependent
 - [Non-Newtonian powerlaw](#)
 - [Non-Newtonian Carreau](#)
 - [Non-Newtonian user defined](#)



Non-Newtonian Herschel-Bulkley

The Herschel-Bulkley model combines the power-law model with a yield stress variable.

$$\mu = \frac{\tau_o + k[\dot{\gamma}^n - (\tau_o/\mu_o)^n]}{\dot{\gamma}}, \quad \tau > \tau_o$$

where k is the consistency index (in Pa·sⁿ), n is the power-law index, τ_o is the yields stress threshold (in Pa) and μ_o is the yielding viscosity (in Pa·s). The value of these parameters is to be given in:

Project Tree > Materials > Fluid > Viscosity model > Consistency index (k)

Project Tree > Materials > Fluid > Viscosity model > Powerlaw index (n)

Project Tree > Materials > Fluid > Viscosity model > Yields stress threshold (τ_0)

Project Tree > Materials > Fluid > Viscosity model > Yielding viscosity (μ_0)

Non-Newtonian Cross model

Cross model reads:

$$\mu = \frac{\mu_0}{1 + (\lambda \dot{\gamma})^{1-n}}$$

where μ_0 is the zero shear viscosity (in Pa·s), n is the power-law index and λ is the time constant (in s).

The value of the model parameters has to be given in:

Project Tree > Materials > Fluid > Viscosity model > Zero shear viscosity (μ_0)

Project Tree > Materials > Fluid > Viscosity model > Powerlaw index (n)

Project Tree > Materials > Fluid > Viscosity model > Time constant (λ)

Non-Newtonian power-law

The non-Newtonian power-law is given by:

$$\mu = k \dot{\gamma}^{n-1} e^{\theta_0/\theta}$$

where k is the consistency index (in Pa·s ^{n}), n is the power-law index, θ_0 is the reference temperature (in K); the value of which has to be given in:

Project Tree > Materials > Fluid > Viscosity model > Consistency index (k)

Project Tree > Materials > Fluid > Viscosity model > Powerlaw index (n)

Project Tree > Materials > Fluid > Viscosity model > Reference temperature (θ_0)

Additionally, it is possible to set minimum ($\mu_{\min}$) and maximum ($\mu_{\max}$) values for the power-law function:

Project Tree > Materials > Fluid > Viscosity model > Minimum viscosity ($\mu_{\min}$)

Project Tree > Materials > Fluid > Viscosity model > Maximum viscosity ($\mu_{\max}$)

Power-law fluids can be subdivided into three different types based on the value of the index:

- $n > 1$ pseudoplastic or shear-thinning fluid
- $n < 1$ dilatant or shear-thickening fluid
- $n = 1$ is equivalent to Newtonian fluid

Non-Newtonian Carreau

At low shear rate the Carreau model behaves as Newtonian and at high shear rate as power-law:

$$\mu = \mu_\infty + (\mu_0 - \mu_\infty) \left[1 + (H(\theta)(\dot{\gamma})\lambda)^2 \right]^{(n-1)/2}$$

where

$$H(\theta) = \exp \left[\alpha \left(\frac{1}{\theta} - \frac{1}{\theta_\alpha} \right) \right]$$

and λ is the time constant (in s), n is the power-law index, θ_α is the reference temperature at which $H(\theta) = 1$ (in K), μ_0 is the zero shear viscosity (in Pa·s), μ_∞ is the infinite shear viscosity (in Pa·s) and α is the temperature sensitivity (in K):

Project Tree > Materials > Fluid > Viscosity model > Time constant (λ)

Project Tree > Materials > Fluid > Viscosity model > Powerlaw index (n)

Project Tree > Materials > Fluid > Viscosity model > Reference temperature (θ_0)

Project Tree > Materials > Fluid > Viscosity model > Zero shear viscosity (μ_0)

Project Tree > Materials > Fluid > Viscosity model > Infinite shear viscosity (μ_∞)

Project Tree > Materials > Fluid > Viscosity model > Temperature sensitivity (α)

Non-Newtonian user defined

User can [edit](#) the dynamic viscosity field using the variables: temperature (**theta**) and shear rate (**gamma**).

5.3.9 HTC Estimation

Project Tree > Materials > Fluid > HTC Estimation

This option is available only for isothermal flows, where an estimation of HTC can be done based on the skin friction at the wall according to the mode selected:

- **Automatic:** The required input are the **Specific heat capacity** and the **Thermal diffusion**.

The heat transfer coefficient is estimated using the following equation:

$$\frac{C_p \rho u_\tau}{\sigma_{\phi,t}(u^+ + P)}$$

where P is the sublayer resistance factor given by Jayatilleke^[1]:

$$P \equiv 9.24 \left[\left(\frac{\sigma_\phi}{\sigma_{\phi,t}} - 1 \right)^{3/4} \right] \left[1 + 0.28 \exp \left(\frac{-0.007 \sigma_\phi}{\sigma_{\phi,t}} \right) \right]$$

and where C_p is the specific heat capacity, ρ the density, u_τ is the friction velocity, $\sigma_{\phi,t}$ the turbulent Prandtl number, σ_ϕ the molecular Prandtl number, and u^+ is the dimensionless velocity (u/u_τ).

[1] C. Jayatilleke, *The Influence of Prandtl Number and Surface Roughness on the Resistance of the Laminar Sublayer to Momentum and Heat Transfer*, Prog. Heat Mass Transfer, 1:193-321, 1969.

- **Custom:** The required input are the **Specific heat capacity**, the **Thermal diffusion**, and the **Reference Y+**.

The heat transfer coefficient is estimated using the following equation:

$$htc = \rho C_p \frac{u_{ref}}{\left(\frac{T^+}{u^+} \right)} \frac{C_f}{2} = C_p \frac{\tau}{\left(\frac{T^+}{u^+} \right) 2 u_{ref}}$$

where u^+ and T^+ are dimensionless velocity and temperature evaluated from the [law of the wall](#), and are such as:

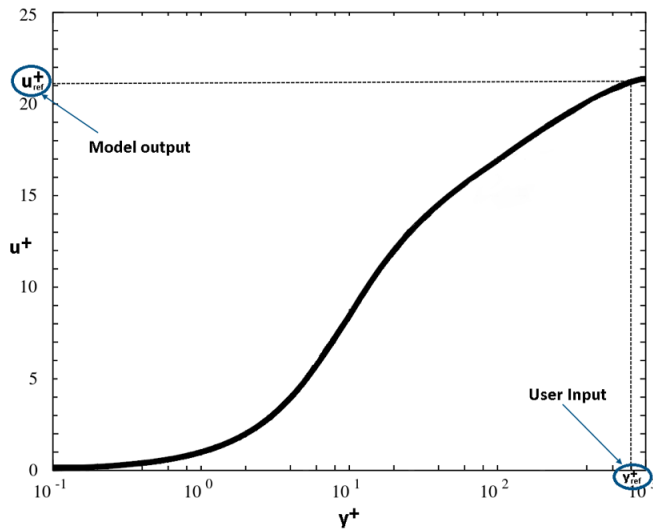
$$u^+ = \frac{u}{u_\tau}$$

and

$$T^+ = \frac{(T_w - T_f) \rho C_p u_\tau}{q}$$

τ is the shear stress, u_τ the friction velocity, C_f the skin friction, C_p the specific heat capacity, T_w the wall temperature, T_f the near-wall fluid temperature, ρ the density, and q the heat flux.

When using the **Custom** mode, the user is requested to provide the **Reference Y+** (Y^+_{ref}) in order to determine the dimensionless reference velocity u^+_{ref} via the [law of the wall](#):



The dimensionless reference velocity is then used to determine the reference velocity u_{ref} used to compute the HTC value, through the relation:

$$u_{ref} = u_{\tau} \cdot u_{ref}^{+}$$

5.3.10 Thermal conductivity

Project Tree > Materials > Fluid > Thermal conductivity

The thermal conductivity is used to compute the [heat transfer coefficient](#) for both non-isothermal and isothermal simulations. For non-isothermal simulations (segregated energy, coupled energy, and supersonic), it will also be taken into account in the energy equation to model the fluid behavior.

The default value for gases is that of air at 15°C (0.0243 W m⁻¹ K⁻¹). For liquids the default value is that of the liquid water (0.58 W m⁻¹ K⁻¹).

5.3.11 Specific heat capacity

Project Tree > Materials > Fluid > Specific heat capacity

The Specific heat capacity parameter is used to compute the [heat transfer coefficient](#) for both non-isothermal and isothermal simulations. For non-isothermal simulations (segregated energy, coupled energy, and supersonic), it will also be taken into account in the energy equation to model the fluid behavior.

The default value for gases is that of air at 15°C (1006.43 J kg⁻¹ K⁻¹). For liquids the default value is that of the liquid water (4182 J kg⁻¹ K⁻¹).

⚠ Please note: Specific heat capacity is a [numerical field](#). Hence, the user must specify a constant value. Specific heat capacity dependence on temperature cannot be considered yet.

5.3.12 Adiabatic index

Project Tree > Materials > Fluid > Adiabatic index

The adiabatic index is required to compute the speed of sound of an ideal gas. This parameter is therefore only available for [Acoustics analysis](#) and [Supersonic](#) flow model. For an ideal gas, the speed of sound $C_{thermodynamic}$ is related by the following equation:

$$C_{thermodynamic} = \sqrt{\frac{\gamma RT}{M}}$$

where the M is the [molecular weight](#), $R=8314$ is the the perfect gas constant, and T is the [operating temperature](#).

The default value of the adiabatic index for air at atmospheric conditions is 1.4.

 **Please note:** The speed of sound used in the simulation is specified in the [simulation log file](#).

5.3.13 Reference pressure

Project Tree > Materials > Fluid > Reference pressure/density

The reference pressure/density is shown for [Coupled Energy](#) and [Supersonic](#) thermal models since these two models work with absolute pressure. If the state equation is set to impose the reference pressure then the reference density is shown, and if the reference density is imposed the reference pressure is shown. The reference pressure and density are computed based on the ideal gas law:

$$P = \rho \frac{R}{M} T$$

where P is the pressure, ρ the density, $R=8314$ is the the perfect gas constant, M is the [molecular weight](#), and T is the [operating temperature](#).

5.3.14 Interactions

Surface tension model

The surface tension is available for Free Surface and Multiphase flow models.

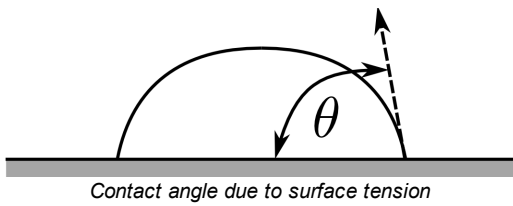
Project Tree > Materials > Interactions > Surface tension model

Surface tension is a property of the fluid whose origins lie in the different inter-molecular forces that act on both sides of the interphase between two fluid phases, e.g. on a liquid free surface. This results is an energy per unit area that acts on the interphase that can be specified by the user in:

Project Tree > Materials > Interactions > Surface tension model > Surface tension

The default value is 0.072 N.m^{-1} that corresponds to the Surface tension of water in contact with air at 25°C .

Surface tension is also responsible for the contact angle θ at which a free surface meets a solid wall, as shown in the figure below. The contact angle is specific for any given system, it is determined by the interactions across the fluid-fluid and fluid-solid interfaces.



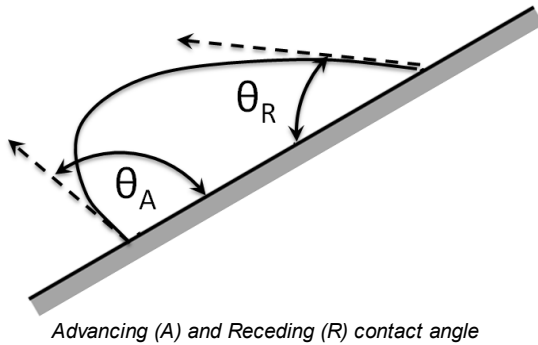
The user can specify the value of the contact angle in:

Project Tree > Materials > Interactions > Surface tension model > Contact angle (by default 90 degrees)

Also, contact angle hysteresis can be simulated (only available for [Multiphase Phase Field](#)) by setting the following option to **On**:

Project Tree > Materials > Interactions > Surface tension model > Contact angle hysteresis (by **Off**)

In this case the contact angle set by the solver will differ in case the surface has already been in contact with the fluid. Hence, an advancing and a receding contact angle must be specified (by default 90 degrees).



Phase field parameters

These parameters are available only when the [Multiphase Phase Field](#) model is enabled.

- **Project Tree > Materials > Interactions > Interface thickness** Number of lattice elements used to model the interface thickness (default value 2.5).
- **Project Tree > Geometry > Shape: Name of shape > Shape contact angle** Setting this option to **On** allows the user to specify different contact angle and a contact angle hysteresis for each shape. These values will overwrite the ones specified in the Materials tab.

5.4 Geometry

Project Tree > Geometry

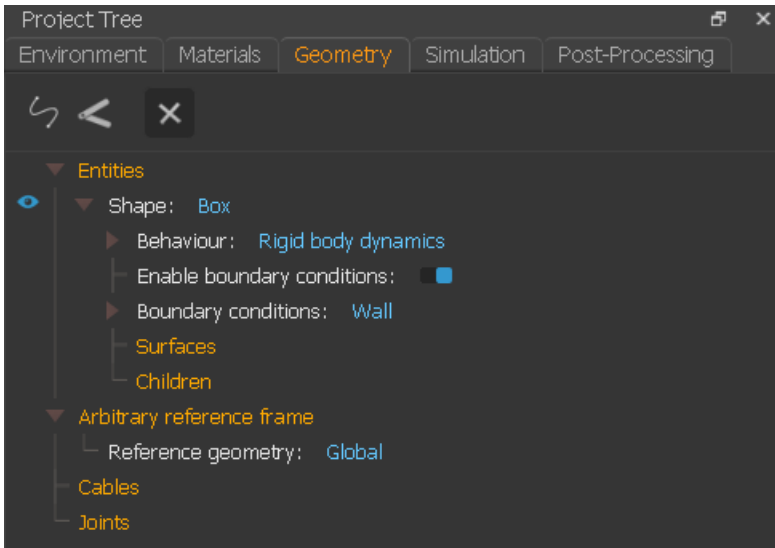
The **Geometry** section of the *Project Tree* contains the information about the geometry objects; these can be of the following types:

- [Entities](#)
- [Cables](#)

- [Joints](#)

5.4.1 Entities

Project Tree > Geometry > Entities



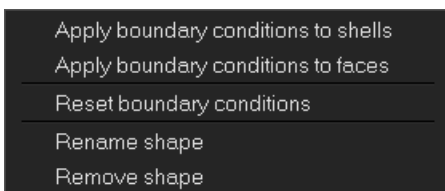
Entities are divided in two types:

Shapes

If the problem under study requires a geometry (e.g. internal analysis, or external analysis with objects within the external domain), the user can [create](#) / [import](#) it (see [Chapter Geometry](#) for more information about geometry handling).

Every geometry object is listed in this tab of the *Project Tree* as a **Shape**. Here, the behaviour (motion) and boundary conditions of any **Shape** can be defined.

Several actions can be applied to a shape by doing a right-click on the shape line:



- **Apply boundary conditions to shells**: applies boundary conditions on every shells of the geometry.
- **Apply boundary conditions to faces**: applies boundary conditions on every surface of the geometry.
- **Reset boundary conditions**: resets the surfaces boundary conditions as one single condition applied over the entire shape.
- **Rename shape**: renames the shape in the geometry tree. Renaming can also be performed selecting the geometry and pressing the F2 key.
- **Remove shape**: removes the shape from the project.

Please note: It is recommended to make a copy of the geometry before changing names or doing operations as all changes will be applied and saved from the original geometry file imported

Reference frame

Geometry > Create object > Reference frame

A reference frame is an arbitrary axis system defined by a point and orientation. It allows to create alternative reference frames than the XFlow global frame (point of coordinates (0,0,0) m and global X, Y and Z axis). It is especially useful to compute forces and moments according to any reference frame in the [Function Viewer](#).

Shape	Name	
	Behaviour (+Info)	Fixed Enforced Rigid body dynamics Animated geometry Control geometry
	Enable boundary conditions	Boundary conditions ⁽¹⁾ (+Info) Wall Inlet Outlet Others Thermal boundary conditions ⁽¹⁾ ⁽²⁾ (+Info) Adiabatic Temperature Temperature jump Heat flux Free convection Volume heat source Convection radiation
		Conjugate heat transfer ⁽²⁾ ⁽³⁾ (+Info) Structural Coupling ⁽⁴⁾ (+Info)
Reference frame	Name	
	Behaviour (+Info)	Fixed Enforced

⁽¹⁾ Only if Enable boundary conditions enabled

⁽²⁾ Only if (Thermal model= Isothermal)

⁽³⁾ Only if (Behaviour = Fixed)

⁽⁴⁾ Only if (Structural analysis = Abaqus)

5.4 Geometry

5.4.1.1 Behaviour

Project Tree > Geometry > Entity > Behaviour

Entities can have the following behaviours in the simulation:

1. [Fixed](#)
2. [Enforced](#)
3. [Rigid body dynamics](#)⁽¹⁾
4. Others
 - a) [Animated geometry](#)⁽¹⁾

⁽¹⁾ Only if (Entity = Shape)

Fixed

The entity does not move. Its behaviour is defined by its **Position** and **Orientation**, which remain constant in time. A scaling factor can also be provided with the **Scale** field.

 **Please note:** The scaling factor is available for all the behaviors and is also updated in the [Transform Tool](#).

Enforced

The entity has a prescribed movement given by the **Position law** (translation) and **Angle mode** (rotation), [edited](#) by the user.

The **Angle mode** has two options:

- **Euler angles:** for prescribed rotation around global axes. **Angular laws** around X, Y and Z-axis need to be entered.
- **Axis angles:** for prescribed rotation around an arbitrary axis. **Axis direction law** and **Angular law** need to be entered.

 **Please note:** The analytic expression for these laws can only depend on time **t**.

The **Immersed Boundary Method** option (only available in [Expert mode](#)) allows the user to select a different technique to simulate moving geometries in XFlow. Whenever this option is enabled no [broken links](#) will be visualized in domain cutting planes, since the solver does not use this information to track the motion of the moving objects. Instead, an extension of the fluid region is created inside the geometry in which the velocity and pressure fields are solved to provide the correct wall boundary conditions. The computational cells are filled with fluid using a solid fraction field and the collision step of the LBM is modified to include a term that depends on that solid fraction

Considerable runtime and MPI parallel scalability improvements as well as smoother forces time-history plots can be expected when selecting this option.

 **Please note:** The immersed-boundary method requires a sufficient number of lattice elements across the thickness of the object in order to resolve the solid fraction field properly, and therefore it is usually recommended to use the IB method for geometries that are well resolved. Also, the Immersed Boundary Method option is not suitable for zero-thickness geometries. Therefore, thin-shell moving objects should be simulated using the standard option.

Rigid body dynamics

The **Shape** behaves as a rigid solid. Displacement (in X, Y and Z direction) and rotation (around X, Y and Z axis) degrees of freedom can be activated. Each degree of freedom can be subject to external forces and external moments.

The settings for this behaviour are:

- **Initial conditions:**

- ↳ **Position:** initial position
- ↳ **Velocity:** initial velocity
- ↳ **Orientation:** initial orientation with respect to local axes (in degrees)
- ↳ **Angular velocity:** initial angular velocity in global axis (in radians/second)

- **Mechanical properties:**

- ↳ **Inertia tensor:**

- **Automatic:** XFlow automatically computes the mass and the inertia tensor assuming a uniform mass distribution, based on:
 - ↳ **Density:** The object density.
- **User defined:** The tensor is assumed to be diagonal. The user has to specify:
 - ↳ **Mass:** Total object mass.
 - ↳ **Ixx:** first diagonal component of the inertia tensor
 - ↳ **Iyy:** second diagonal component of the inertia tensor
 - ↳ **Izz:** third diagonal component of the inertia tensor
- **User defined full tensor:** The tensor is not assumed to be diagonal and thus the user has to specify all the tensor components:
 - ↳ **Mass:** Total object mass.
 - ↳ **Ixx:** diagonal component of the inertia tensor
 - ↳ **Iyy:** diagonal component of the inertia tensor
 - ↳ **Izz:** diagonal component of the inertia tensor
 - ↳ **Ixy:** non-diagonal component of the inertia tensor
 - ↳ **Ixz:** non-diagonal component of the inertia tensor
 - ↳ **Iyz:** non-diagonal component of the inertia tensor

- **Interaction properties:**

- ↳ **Restitution coefficient:** solid/solid contact coefficient (dimensionless). It expresses how one solid colliding another will bounce. If this coefficient is equal to 1, the object will bounce as a perfectly elastic solid since all the energy will be returned to the bouncing solid. If it is equal to 0, it will not bounce and stick to the other object which will absorb all its energy.
- ↳ **Static friction coefficient:** the static friction coefficient between two surfaces of solids is defined as the ratio of the tangential force (F) required to make an object sliding over the other divided by the normal force between the surfaces (N): $sfc = F/N$. Basically, this means a static friction coefficient of 1 will require strong forces to move two static objects in contact, and a coefficient of 0 will let them move immediately when a force is applied, as a slippery surface (dimensionless).
- ↳ **Dynamic friction coefficient:** The dynamic friction coefficient is the same than the static friction coefficient but for two objects in motion. Even though they are in motion, the force of friction cannot be eliminated completely and it continues to be a resistive force to motion. (dimensionless)

 **Please note:** Although friction forces act between pairs of bodies, in XFlow friction coefficients are assigned to the shapes. The effective friction coefficient used in the computation is the average of those of the two shapes interacting. For example, if solid1 has a friction coefficient of 0.5 and solid2 of 0.1, the effective friction coefficient between both solids will be 0.3.

- **Constraints**

↳ **Translation** (Global): sets the geometry degrees of freedom (DOF) in translation. The axis X, Y and Z are expressed in the global axis. Available options are self descriptive:

1. **Free:** all DOF in translation allowed;
2. **Fixed:** no DOF in translation allowed;
3. 1D translation: **Axis x**, **Axis y**, **Axis z**, **Arbitrary axis** (**Direction** user defined);
4. 2D translation: **Plane x-y**, **Plane x-z**, **Plane y-z**, **Arbitrary plane** (**Normal** user defined).

↳ **Rotation** (Local): sets the geometry degrees of freedom (DOF) in rotation. The axis X, Y and Z are expressed in the local axes. Available options are self descriptive:

1. **Free:** all DOF in rotation allowed;
2. **Fixed:** no DOF in rotation allowed;
3. 1D rotation: **Axis x**, **Axis y**, **Axis z**, **Arbitrary axis** (**Direction** user defined);
4. 2D rotation: **Axis x-y**, **Axis x-z**, **Axis y-z**.

↳ **External forces:** allows the user to define an external force applied on the geometry (e.g. the gravity force, spring force, damping force, etc.). This force can be expressed in the global and local axes.

↳ **External moments:** allows the user to define an external moment applied on the geometry (e.g. resistive torque, frictions losses, etc.). This moment can be expressed in the global and local axes.

External forces and external moments can be expressed as [function](#) of the following variables:

px, py, pz	position of CoG in global X, Y and Z directions
vx, vy, vz	velocity of CoG in global X, Y and Z directions
eux, euy, euz	angle of the CoG in the local X, Y and Z directions
wx, wy, wz	angular velocity in the local X, Y and Z directions

 **Please note:** External forces and moments are both applied in the center of gravity (CoG)

 **Please note:** Rigid body dynamics behaviour is not available for these entities: Vertex, Reference frame, Line and Curve.

Animated geometry

Only available in [Labs mode](#).

5.4.1.2 Boundary conditions

Project Tree > Geometry > Shape > Enable boundary conditions (On/Off)

This option enables/disables a shape boundary conditions (i.e.: it enables interaction of the shape with fluid during the simulation). If disabled, the geometry can still be used as an [arbitrary refinement region](#) or [initialize a field](#) inside the geometry, but it will not interact with the fluid.

Project Tree > Geometry > Shape > Boundary conditions

In XFlow, boundary conditions are classified as follows:

- [Wall](#)
- [Inlet:](#)
 - Velocity
 - Mass flow
 - Pressure inlet
 - Total pressure inlet
- [Outlet:](#)
 - Pressure outlet
 - Convective outlet
 - Velocity
 - Mass flow
- [Others:](#)
 - Fan model
 - Porous jump
 - Porous volume
 - Periodicity
 - Porous wall inlet

5.4.1.2.1 Wall boundary condition

In order to reduce computing needs near walls, XFlow uses a wall function to estimate the velocity and the temperature at the closest node to the wall.

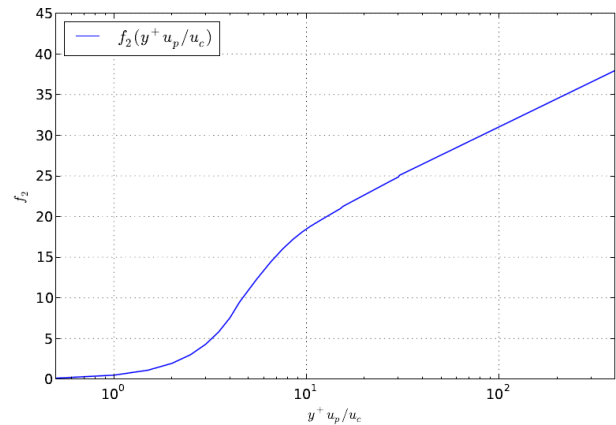
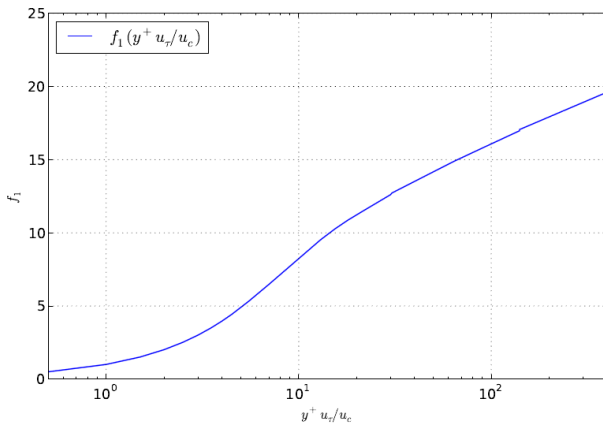
XFlow features a generalized law of the wall that takes into account the effect of adverse and favorable pressure gradients to model the boundary layer. This wall function is valid for all y^+ and accurately resolves the turbulent boundary layer taking into account also for the influence of curvature and pressure gradient.

The formulation of this wall function to estimate the velocity is as follows based on Shih et al. [1]:

$$\begin{aligned}
 \frac{U}{u_c} &= \frac{U_1 + U_2}{u_c} = \frac{u_\tau}{u_c} \frac{U_1}{u_\tau} + \frac{u_p}{u_c} \frac{U_2}{u_p} \\
 &= \frac{\tau_w}{\rho u_\tau^2} \frac{u_\tau}{u_c} f_1 \left(y^+ \frac{u_\tau}{u_c} \right) + \frac{dp_w/dx}{|dp_w/dx|} \frac{u_p}{u_c} f_2 \left(y^+ \frac{u_p}{u_c} \right) \\
 y^+ &= \frac{u_c y}{\nu} \\
 u_c &= u_\tau + u_p \\
 u_\tau &= \sqrt{|\tau_w| / \rho} \\
 u_p &= \left(\frac{\nu}{\rho} \left| \frac{dp_w}{dx} \right| \right)^{1/3}
 \end{aligned}$$

Here, y^+ is the normal distance from the wall, u_τ is the skin friction velocity, τ_w is the turbulent wall shear stress, dp_w/dx is the wall pressure gradient, u_p is a characteristic velocity of the adverse wall pressure gradient and U is the mean velocity at a given distance from the wall. The interpolating functions f_1 and f_2 are depicted as follows:

5.4 Geometry



This single consistent law of the wall is based on a unified non-equilibrium wall function that accounts for continuous blending between viscous sub-layer and logarithmic layer, adverse and favorable pressure gradients and surface curvature in a completely automatic way.



Tip: See the following reference:

[1] T.-H. Shih, L. Povinelli, N.-S. Liu, M. Potapczuk and J. Lumley, "A generalized wall function", National Aeronautics and Space Administration (NASA), 1999.

For thermal analysis, the temperature is estimated with the following logarithmic profile broadly used in the literature [2,3,4]:

$$T^* = \begin{cases} Pr_t y^* & y^* < y_T^* \\ Pr_t \left\{ \frac{1}{k} \ln(9.8y^*) + P \right\} & y^* > y_T^* \end{cases}$$

And T^* , y^* are given by:

$$T^* = \frac{(T_w - T_P) \rho c_p u_\tau}{q} \quad y^* = \frac{y u_\tau}{\nu}$$

Where:

T_w is the temperature at the wall
 T_P is the temperature at the lattice node
 ρ is the fluid density
 c_p is the specific heat
 u_τ is the friction velocity
 q is the heat flux
 ν is the kinematic viscosity

The constant P is calculated following the Jayatilleke proposal in [3], its expression is as follows:

$$P = 9.24 \left\{ \left(\frac{Pr}{Pr_t} \right)^{3/4} - 1 \right\} \left\{ 1 + 0.28 e^{-0.007 Pr / Pr_t} \right\}$$

The use of the linear or logarithmic expression in the law of the wall depends on the point where both parts join each other (y_T^*). If the node is in the linear part, there is enough spatial resolution for resolving the boundary layer and in such case the thermal wall function is disabled.



Tip: See the following references:

- [2] Patankar, S. V. and Spalding, D. B. 'Heat and mass transfer in boundary layers', 2nd ed., Inter Text Books, London, 1970.
- [3] Launder, B. E. and Spalding, D. B. 'The numerical computation of turbulent flow', Computer Methods Appl. Mech. Eng., 1974, 3, 269-289.
- [4] von Karman, T. 'The analogy between fluid friction and heat transfer', Trans. ASME, 1939, 61, 705-710.

Several options are available in XFlow for the wall treatment:

Automatic

It sets the enhanced wall-function, explained below. This formulation is the most generalist and robust one, hence is the automatic selection.

↳ **Wall roughness [m]:** Wall characteristic roughness, by default is zero. Roughness size should be smaller than the distance between the wall and the first lattice node in order to be consistent with the wall function.

Off (resolved)

It sets the velocity of the fluid to zero at the wall boundary. No wall function is used and thus the boundary layer must be resolved by XFlow which requires a very low Y^+ , and therefore may lead to a very high number of elements.

Enhanced wall-function

This sets the generalized wall function described above, but it disables the pressure gradients term.

↳ **Wall roughness [m]:** Wall characteristic roughness, by default is zero. Roughness size should be smaller than the distance between the wall and the first lattice node in order to be consistent with the wall function.

Non-equilibrium enhanced wall-function

It sets the generalized wall function described above and takes into account for the pressure gradients to predict separation.

↳ **Wall roughness [m]:** Wall characteristic roughness, by default is zero. Roughness size should be smaller than the distance between the wall and the first lattice node in order to be consistent with the wall function.



Tip: It is strongly recommended to use the non-equilibrium enhanced wall-function for aerodynamic applications, since the prediction of the boundary layer separation based on pressure gradients is a fundamental feature. However, it would be recommended to keep the enhanced wall-function for general applications not related to aerodynamics (multiphase flows, free surface flows, etc.).

Free-slip

This sets the normal component of the fluid velocity at the wall as well as the wall shear stress to zero. The fluid is reflected symmetrically by the wall.

For any of the above wall models, an additional feature to set a [virtual moving wall](#) boundary condition is available *only in Expert mode*.



Please note: The maximum value of wall roughness is not limited by XFlow. It is the user's responsibility to set the cell size near the wall larger than the wall roughness.

5.4 Geometry

5.4.1.2.1.1 Virtual moving wall boundary

This option is only available in [Expert mode](#).

Project Tree > Geometry > Shape > Boundary conditions > Wall

- ↳ **Virtual moving wall:** (On/Off) If enabled, this feature imposes velocity on the boundary but geometry shape remains fixed. The user has to ensure that the boundary conditions are consistent, i. e. velocity component normal to the wall is zero.

The **Shape** has a prescribed movement given by the **Position laws** of translation and rotation, [edited](#) by the user.

- **Axis ref point:** coordinates of the virtual center of rotation in global axis.
- **Axis direction:** coordinates of the virtual axis of rotation with respect to global axis.
- **Angular law:** angular rotation in degrees.

 **Tip:** The virtual moving walls are useful to model rotating that are axisymmetric, e.g. rotating wheels for automotive simulations. This avoids to set an [enforced](#) rotating motion and makes the computation faster.

5.4.1.2.2 Inlet boundary conditions

Total pressure inlet

This boundary condition is used to define a total **pressure law** at inlet boundaries. The total pressure is defined by the following equation:

$$P_{\text{total}} = P_{\text{static}} + P_{\text{dynamic}} = P_{\text{static}} + \frac{1}{2}\rho V^2$$

- ↳ **LODI: On/Off.** Enables local one-dimensional inviscid (LODI) condition at the boundary (non-reflecting boundary condition). ([+Info](#))

 **Please note:** the pressure boundary conditions are **gauge pressure values** for isothermal and segregated energy [thermal models](#), whereas they are **absolute pressure values** for [coupled energy](#) and [supersonic](#) thermal models.

Velocity

This boundary condition is used to impose a **Velocity law** and an [Inlet turbulence intensity](#) at a surface.

The [advanced option](#) in the Engine section named High order boundary conditions activates 2nd order BC instead of 1st order.

- ↳ **LODI: On/Off.** Enables local one-dimensional inviscid (LODI) condition at the boundary (absorbing boundary condition). ([+Info](#))

Mass flow

This boundary condition is used to prescribe a mass flow rate at an inlet, given by the **Mass flow law** (in kg/s). The mass flow is defined by the following equation:

$$\dot{m} = \rho V A$$

where the density ρ and the velocity V both adapt to the section area A of the boundary.

- ↳ **LODI: On/Off.** Enables local one-dimensional inviscid (LODI) condition at the boundary (non-reflecting boundary condition). ([+Info](#))

Pressure inlet

This boundary condition is used to define the static pressure at flow inlets. All other flow quantities are extrapolated from the internal domain. This boundary type has the following items:

- ↳ **Gauge/absolute pressure law:** to define the value of the gauge or absolute pressure at the outlet. The pressure boundary conditions are **gauge pressure values** for isothermal and segregated energy [thermal models](#), whereas they are **absolute pressure values** for [coupled energy](#) and [supersonic](#) thermal models.

5.4.1.2.3 Outlet boundary conditions

Pressure outlet

This boundary condition is used to define the static pressure at flow outlets. All other flow quantities are extrapolated from the internal domain. This boundary type has the following items:

- ↳ **Gauge/absolute pressure law:** to define the value of the gauge or absolute pressure at the outlet. The pressure boundary conditions are **gauge pressure values** for isothermal and segregated energy [thermal models](#), whereas they are **absolute pressure values** for [coupled energy](#) and [supersonic](#) thermal models.
- ↳ **Allow backflow: On/Off.** When set to on, the fluid can flow back through the outlet boundary into the domain. When set to off, the backflow is not allowed and it forces the fluid to exit the domain.
- ↳ **LODI: On/Off.** Enables local one-dimensional inviscid (LODI) condition at the boundary (non-reflecting boundary condition). ([+Info](#))

Convective outlet

In this boundary condition both static pressure and velocity at the outlet are extrapolated from the internal domain.

- ↳ **Allow backflow** can be set enabled. When set to on, the backflow is allowed which means the fluid can flow back in the domain from the outlet boundary. When set to off, the backflow is not allowed and it forces the fluid to go out of the domain.

 **Tip:** The use of the pressure outlet boundary condition often results in a better rate of convergence than convective outlet when backflow occurs. On the other hand, a convective outlet absorbs better the reflection waves caused by inconsistent initial conditions.

Velocity

This boundary condition is used to impose a **Velocity law** at an outlet.

- ↳ **LODI: On/Off.** Enables local one-dimensional inviscid (LODI) condition at the boundary (non-reflecting boundary condition). ([+Info](#))

Mass flow

This boundary condition is used to prescribe a mass flow rate at an outlet, given by the **Mass flow law** (in kg/s). The mass flow is defined by the following equation:

$$\dot{m} = \rho V A$$

where the density ρ and the velocity V both adapt to the section area A of the boundary.

It should be noticed that the sign applied to the mass flow rate will be coherent with the normals orientation of the outlet surface.

- ↳ **LODI: On/Off.** Enables local one-dimensional inviscid (LODI) condition at the boundary (non-reflecting boundary condition). ([+Info](#))

5.4.1.2.4 Other boundary conditions

Fan model

This boundary condition allows the user to prescribe any law that defines a (static) pressure jump in the direction of the normal to the surface as a function of spatial coordinates and time.

↳ **Fan law:** $p_{out} - p_{in} = f(px, py, pz, vx, vy, vz, t)$

Porous jump

The porous jump boundary condition is a simplification of the porous volume and can be used to model thin "membranes" with known pressure-drop properties.

The pressure jump through the porous surface is modeled by the Darcy-Ergun formulation:

$$p_{out} - p_{in} = -th \left(\frac{\mu}{k} v + \rho \frac{c_E}{\sqrt{k}} v^2 \right)$$

where k is the permeability coefficient (in m^2), v is the velocity normal to the porous surface (in m/s), c_E is the Ergun coefficient (dimensionless) and th is the surface thickness (in m). This expression takes into account inertial and viscous losses in the porous medium due to high flow velocities.

The parameters defining this boundary condition are:

↳ **Porosity type:** Porosity can be defined as anisotropic or isotropic:

- **Anisotropic**

- ↳ **Permeability coefficient** (Vector relative to the porosity principal directions) [m^2]

- ↳ **Ergun coefficient** (Vector relative to the porosity principal directions) [-]

- ↳ **Direction 1:** First porosity principal direction

- ↳ **Direction 2:** Second porosity principal direction

The third direction of the base for the porosity principal directions, is taken normal to the plane defined by **Direction 1** and **Direction 2**.

 **Please note:** For anisotropic porous media (jump or volume) it is advisable to set a permeability coefficient and an Ergun coefficient vector with a ratio between the components not higher than 1e4 and 1e2 respectively.

- **Isotropic**

- ↳ **Permeability coefficient** (Scalar law) [m^2]

- ↳ **Ergun coefficient** (Scalar law) [-]

- ↳ **Surface thickness** [m]

 **Please note:** For Isotropic porous media (jump or volume), the Permeability and Ergun coefficients could also be laws and not only constants.

Porous volume

The porous volume can be used for modeling flow in porous media, such as flows through sand, packed beds, filters, perforated plates, or foams. As in the [porous jump](#) condition, it assumes that the flow through the porous medium obeys the Darcy-Ergun formulation, and porosity can be isotropic or anisotropic. The settings are the same as in the porous jump condition.

 **Please note:** For porous volumes, the normals should be oriented outside the body.

Thermal boundary conditions can be specified for porous volume, once the [Segregated Energy](#) thermal model is selected in the [Environment](#) tab.

↳ **Thermal boundary conditions:**

- **Non-adiabatic** Heat is exchanged within the porous volume
- **Volume heat source** Heat is generated within the porous volume
 - ↳ **Volume heat:** Volumetric heat generated within the porous volume [W m^{-3}]

If any of the above thermal condition is selected, additional thermal properties must be specified for the porous volume:

↳ **Thermal properties:**

- ↳ **Porosity:** Fraction of the volume of voids over the total volume
- ↳ **Specific heat capacity:** Specific heat capacity of the solid matrix within the porous volume [J (kg K)^{-1}]
- ↳ **Solid matrix density:** Density of the solid matrix within the porous volume [kg m^{-3}]

This information is then used to calculate the porous volume properties using the formula:

$$\gamma_p = p \gamma_f + (1 - p) \gamma_s$$

where γ_p is the property of the porous volume, γ_f is the property of the fluid, γ_s is the property of the solid and p is the porosity (as above defined).

Periodicity

Periodic boundary conditions are imposed on two associated surfaces or groups of surfaces, defining a periodicity between them. This boundary condition may be useful to model an infinite system, or if periodicities occur in the system.

⚠ Please note: Periodicity boundary condition must be used on corresponding surfaces that match (same size and shape).

⚠ Please note: The Periodicity boundary condition is only available in for single phase flows.

Porous wall inlet

This boundary condition allows the user to generate mass from a non-slip wall in the direction of the normal to the surface as a function of time. The use of this BC is only available with the [Phase field](#) solver.

- ↳ **Porosity distribution:** Percentage of voxels that are generating mass. The value must be in the range [0, 1].
- ↳ **Mass flow law:** The mass flow (in kg/s) is defined by the following equation:

$$\dot{m} = \rho V A$$

where the velocity v adapt to the area A of the boundary. The considered density ρ is always from Fluid 1.

5.4.1.2.5 LODI

The characteristic boundary condition are non-reflective boundary conditions designed to absorb undesired pressure waves and reflections that may occur in compressible fluid flows. The local one-dimensional inviscid (LODI) equations are solved for this purpose. The goal is to derive non-reflective Dirichlet boundary conditions for both the velocity and the pressure.

Some applications are highly sensible to reflections of pressure waves that may occur at the inlet and outlet boundary conditions in compressible unsteady simulations. To overcome this issue, non-reflective boundary conditions are designed to absorb undesired pressure waves and reflections that may occur in compressible fluid flows. The Local One-Dimensional Inviscid (LODI) equations are solved for this purpose. The goal is to derive non-reflective Dirichlet boundary conditions for both the velocity and the pressure.

5.4 Geometry

It is especially useful for acoustic applications, where the acoustic pressure waves can reflect with the inlet and outlet boundary conditions and interfere the signals. Internal simulations where the flow is not unidirectional would also benefit from this approach, to shorten the transient phase of the flow field initialization.

⚠ Please note: The LODI boundary conditions are available only for single phase flows and in [Expert mode](#).

↪ **Relaxation parameter:** adjusts the stiffness of the LODI equations. The higher and the quicker the system will stabilize to equilibrium and absorb reflections, but it may compromise numerical stability. The lower the softer the absorption will occur but the simulation may be more stable. This can analogically be compared to a spring-mass system where the relaxation parameter would represent the spring stiffness.

⚠ Please note: Recommended values for relaxation parameter should stay within the range of 0.5 - 1.5. The default value of 0.5 is the recommended value. A relaxation parameter of 0 is equivalent to the convective outlet boundary condition.

⚠ Please note: When the LODI option is turned on, the [reference length](#) parameter is available in the Environment tab. The reference length must be set to the characteristic distance traveled by the pressure waves to be damped.

The LODI equations expressed in macroscopic variables are solved at the boundaries. The Euler equations without the transverse terms for a boundary normal to the X direction in 2D are:

$$\begin{aligned}\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + \rho \frac{\partial u}{\partial x} &= 0 \\ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{1}{\rho} \frac{\partial p}{\partial x} &= 0 \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} &= 0 \\ \frac{\partial \rho E}{\partial t} + \frac{\partial [u(\rho E + p)]}{\partial x} &= 0\end{aligned}$$

The LODI variables can be written in the form of a vector $\mathbf{U} = (\rho; u; v; E)$. The LODI equations can be written in the vectorial form:

$$\frac{\partial}{\partial t} \mathbf{U} + \Gamma_x \frac{\partial}{\partial x} \mathbf{U} = 0$$

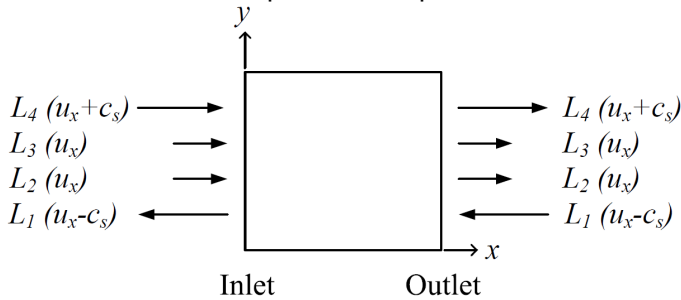
which can be rewritten in the following form:

$$\frac{\partial}{\partial t} \mathbf{U} + \mathbf{S}^{-1} \frac{\partial}{\partial x} \mathbf{L} = 0$$

where $\mathbf{L} = \mathbf{\Lambda} \mathbf{S} \frac{\partial}{\partial t} \mathbf{U}$ with $\mathbf{L}$ the amplitude variations, $\mathbf{S}$ a matrix of left eigenvectors of Γ_x , $\mathbf{\Lambda}$ the diagonal matrix of the eigenvalues of $\mathbf{\Lambda}$. The amplitudes $\mathbf{L} = (L_1; L_2; L_3; L_4)$ are therefore:

$$\mathbf{L} = \begin{pmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{pmatrix} = \begin{pmatrix} (u_x - c_s) \left(\frac{\partial p}{\partial x} - \rho c_s \frac{\partial u_x}{\partial x} \right) \\ u_x \frac{\partial u_y}{\partial x} \\ u_x \left(c_s^2 \frac{\partial \rho}{\partial x} - \frac{\partial p}{\partial x} \right) \\ (u_x + c_s) \left(\frac{\partial p}{\partial x} + \rho c_s \frac{\partial u_x}{\partial x} \right) \end{pmatrix}$$

An illustration of the amplitudes is depicted as follows:



Considering the test cases are isothermal and by means of the above equations, the LODI equations can be reduced to the following relations:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{1}{2c_s^2}(L_4 + L_1) + \frac{1}{c_s^2}L_3 &= 0 \\ \frac{\partial u_x}{\partial t} + \frac{1}{2\rho c_s^2}(L_4 - L_1) &= 0 \\ \frac{\partial u_y}{\partial t} + L_2 &= 0 \end{aligned}$$

5.4.1.3 Surfaces

Apply boundary condition on selected faces/shells

By default, imported shapes are considered as one single group of surfaces on which a single boundary condition is applied. It is possible to apply different boundary conditions to separate faces or shells of a geometrical object.

Select the face  or shell  on which you want to apply a specific boundary condition:

[Graphic View Menu](#) > **Apply boundary conditions to faces**

The surfaces appear then in the tree of the parent shape (**Project Tree** > **Geometry** > **Shape** > **Surfaces**), where the corresponding boundary condition can be specified.

It is possible to apply boundary condition on all shell or surfaces and/or rename shapes and surfaces doing a right-click on the considered shape and as explained in the [Shapes](#) section.

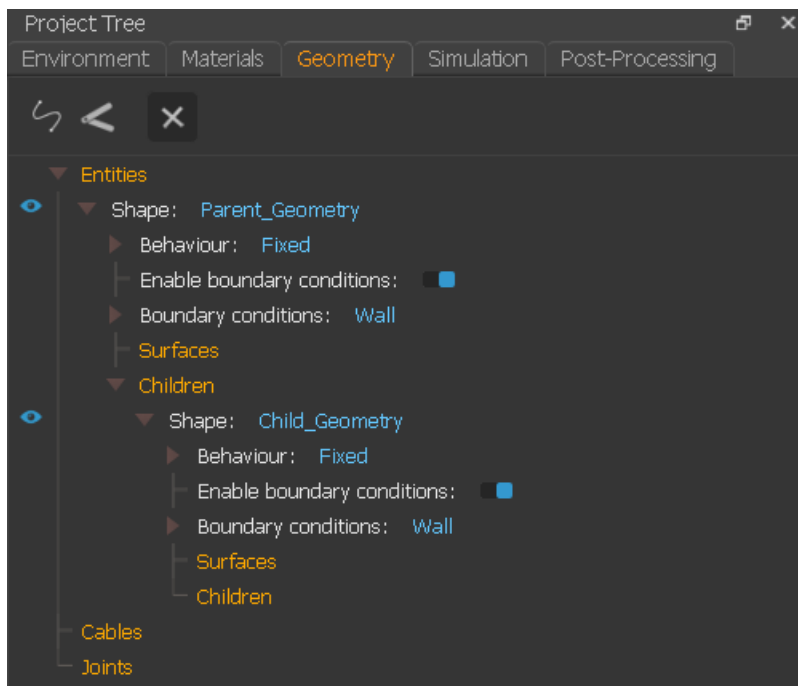
5.4 Geometry

5.4.1.4 Children

If the project deals with several entities, the user can establish a hierarchic relationship among them by dragging one entity (child) onto **Entities > Shape > Children** of another (parent) entity, as shown in the figure below. The reference frame of the child entity will then be defined with respect to the parent entity. To reset the hierarchy level of a geometry right click on it in the project tree and select **Send to 0-hierarchy level**.

⚠ Please note: Rigid body dynamics shapes cannot have enforced or rigid body dynamics parent.

💡 Tip: Hierarchies can be applied between shapes entities as well as reference frame entities.



Entities hierarchy

5.4.1.5 Thermal boundary conditions

When the energy equation is solved, it is needed to define thermal boundary conditions at the surfaces. Four types of thermal boundary conditions are available in XFlow:

Adiabatic

Imposes zero heat flux through the boundary.

Temperature

↳ **Temperature law:** This imposes a fixed temperature field at the boundary, according to the given **Temperature law** (in K), which may depend on the space coordinates and time.

Heat flux

↳ **Heat flux** q (in W/m²): The user can impose a heat flux defined by:

$$q = k A \frac{\partial \theta}{\partial n}$$

where k is the thermal conductivity (in W/K·m), A is the area through which the heat is being transferred

(in m^2) and n is the normal direction to the surface.

Tip: The Heat Transfer Coefficient (HTC) thermal boundary condition can be imposed using the heat flux boundary condition and the [temperature variable](#) $u(x, y, z)$. Indeed, the HTC is related to the heat flux through the law: $h = \frac{q}{\Delta T}$, where q is the heat flux and ΔT the difference of temperature between the fluid in contact to the wall and the outer temperature.

Therefore, to impose an HTC boundary condition one can impose the following heat flux law: $[\text{HTC} * (\text{Temperature_Out} - u(x, y, z))]$, where HTC and Temperature_Out must both be substituted their values.

Free convection

Lets the energy go through the boundary. This is useful for instance to allow free heat transfer through shell components, or for geometries set as [conjugate heat transfer](#) with a heat source (temperature or heat flux) set on one of the faces so that the conduction between fluid and solid happens.

Please note: It is inconsistent to apply free convection on outer boundaries of an internal simulation as no temperature information is available out of the fluid domain. The boundary would act as an adiabatic one.

Volume heat source

This option is available only for geometries set as [porous volume](#) and allows to use the porous volume as a volume heat source. This is useful to model for instance radiators or heat exchangers.

↳ **Volume heat:** Volumetric heat generated within the porous volume [W m^{-3}]

Temperature jump

↳ **Temperature law:** This boundary condition imposes a fixed temperature jump at both sides of the boundary, according to the given **Temperature law** (in K), which may depend on the space coordinates and time. Using the temperature field as the reference, half the temperature jump imposed by the law is added on the boundary side where the surface normals are pointing to whilst the inner side of the surface removes from the temperature field the other half of the jump. This boundary condition conserves the energy of the system.

Convection Radiation

This boundary condition imposes a heat flux resultant from a parallel thermal circuit on a surface. The thermal circuit parameters are the following:

↳ **Surrounding temperature / Temperature at infinity:** Reference temperature of the thermal circuit. Although both terms are named after the convective and radiative common notation, the effect on the fluid is the same. [K]

↳ **Heat transfer coefficient:** The heat transfer coefficient as defined in the previous tip under the Heat Flux boundary condition [$\text{W m}^{-2}\text{K}^{-1}$]

5.4.1.6 Conjugate heat transfer

This option is used to solve Conjugate Heat Transfer (CHT) analysis, where both solid conduction and fluid convection are solved simultaneously. The solid geometry set as conjugate heat transfer must be a watertight volume and the normals must point outwards. Also it is possible to impose CHT in geometry with thickness to model solid conduction even if the geometry is bounding the internal simulation.

⚠ Please note: the thermal boundary conditions disappear when the option is enabled as this is a volumetric condition, and by default all boundaries of the geometry are set as [Free convection](#). If one face of the solid has a specific input temperature or heat flux, one can **select the surface > right-click > Apply boundary condition to faces**. The different surface groups will appear and thermal boundary condition can then be set. Use then [free convection](#) on the faces that should have a conjugate heat transfer.

The solid follows the thermo-physical parameters inputs:

- ↳ **Thermal conductivity** (in W/K·m) of the solid material
- ↳ **Initial temperature law** (in K) of the solid material
- ↳ **Specific heat capacity** (in J/kg·K) of the solid material
- ↳ **Density** (in kg/m³) of the solid material
- ↳ **Volume heat** (in W/m³) to use the geometry as a heat source
- ↳ Thermal coating: models a thermal coating surrounding any CHT geometry when enabled. Considering an homogeneous material, its associated thermal resistance is modeled as:

$$R = \frac{L}{k}$$

Where R is the thermal resistance (in K·m²/W), L (in m) is the coating material thickness and k (in W/(K·m)) is the thermal conductivity of the coating layer. Heat flux through the material q is modeled using the Fourier's law:

$$q = kA \frac{\partial \theta}{\partial n} = \frac{kA}{L}(\theta_1 - \theta_2) = \frac{A}{R}(\theta_1 - \theta_2)$$

Where A (in m²) is the area through which the heat is being transferred. Therefore, the required input parameters are the following:

- ↳ **Coating conductivity** (in W/K·m) of the coating layer
- ↳ **Coating thickness** (in m) of the coating layer

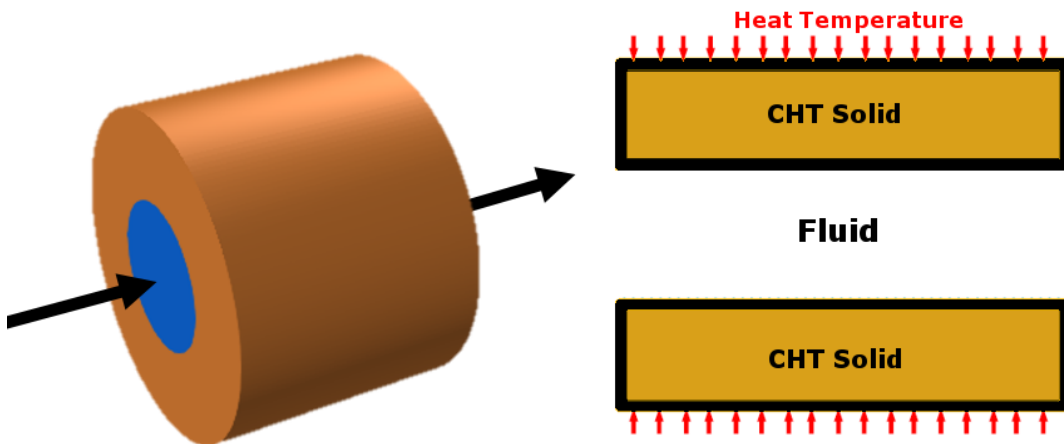
⚠ Please note: In case of bodies in contact (conduction), the heat transfer between them will take into account the series of coating resistances.

⚠ Please note: Conjugate heat transfer is only available for fixed and enforced geometries.

💡 Tip: Conduction between solids can be modeled introducing a small gap (less than one lattice) between the solids in contact.

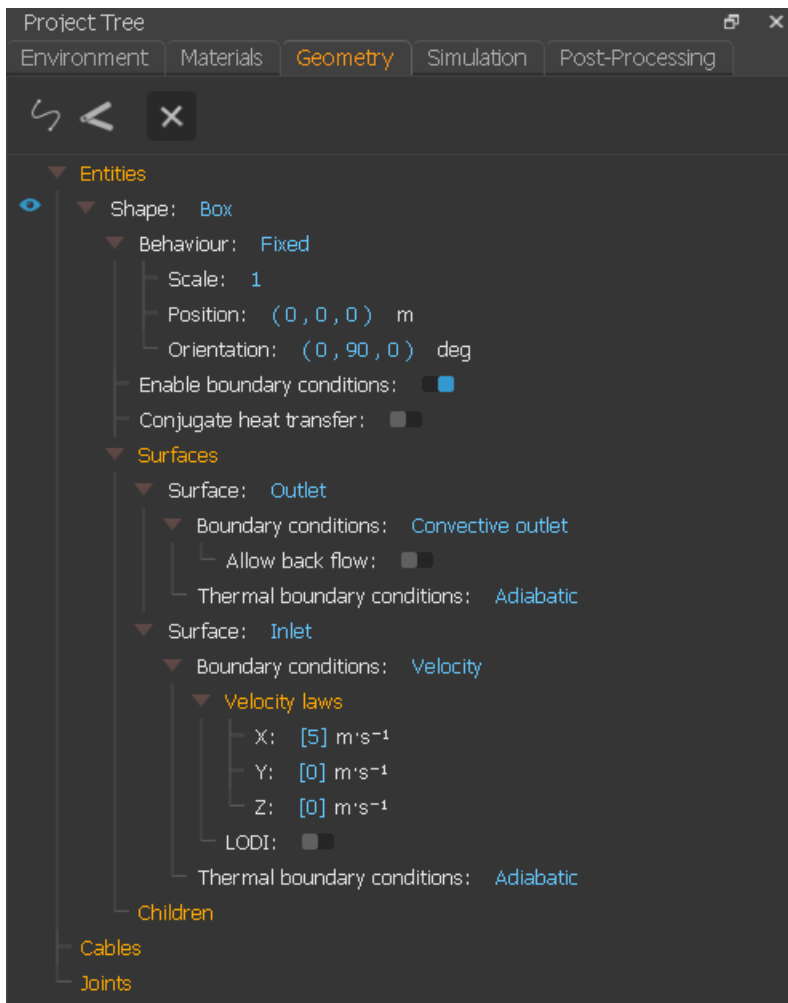
Example of application

A simple illustration of the CHT application is an internal pipe flow with conduction through pipe thickness.

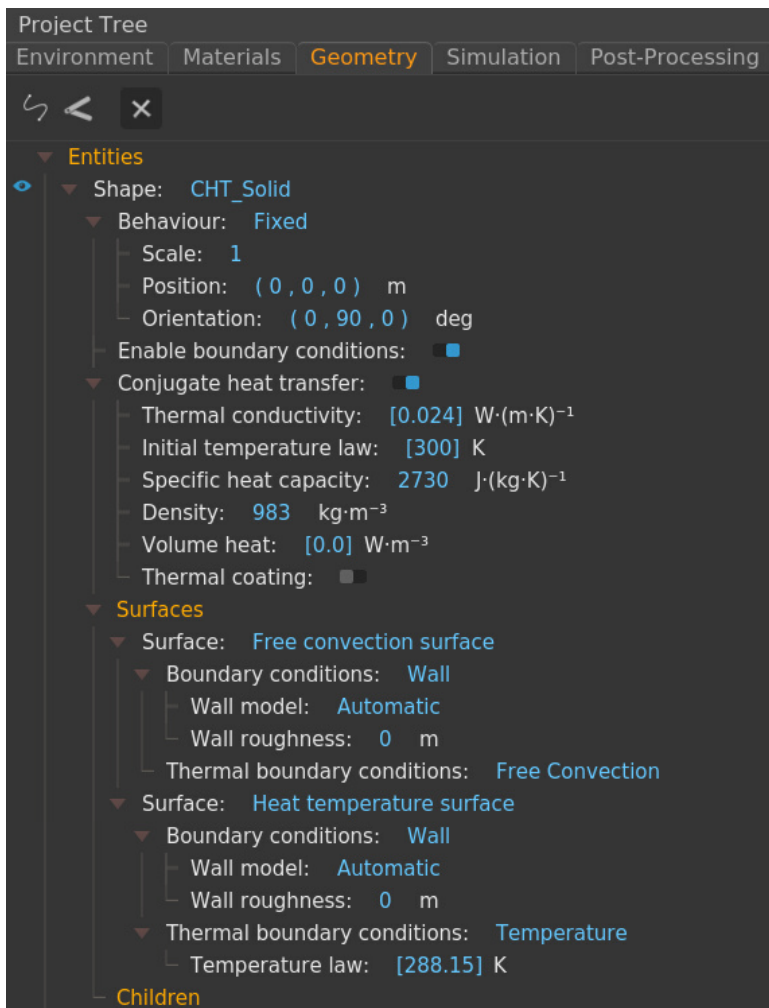


Setup of internal pipe flow heat exchanger.

The pipe geometry is considered as a solid and the normals have to be orientated to point outwards. The fluid domain is bounded by the CHT solid pipe and the ends must be closed with surfaces tessellated correctly (solid/fluid mesh must connect) with the normals orientated to inwards. The covers have been defined as Velocity inlet (5m/s) and Convective outlet boundary condition respectively. The CHT solid is defined as Conjugate Heat Transfer, and the outer surface has a Temperature (300K) boundary condition applied:



XFlow boundary condition setup.



CHT settings and Surface BC.

5.4.1.7 Structural coupling

This option is available when the Structural analysis is activated by selecting the **Abaqus** option from the dropdown list in **Project Tree > Environment > Engine > Advanced Options > Structural analysis**. ([+info](#))

↳ **None:** The shape will not participate in the co-simulation analysis

↳ **One way:** The loads on the shape will be sent to Abaqus, but the deformed geometry will not be retrieved by XFlow. The initial undeformed geometry is maintained throughout the simulation.

⚠ Please note: The one way co-simulation is only available for Enforced behavior.

↳ **Two way:** The loads will be sent to Abaqus, and the shape will be updated with the deformed geometry sent by Abaqus.

 **Please note:** The two way co-simulation is only available for Fixed behavior. The local axis remains fixed during the simulation, however the geometry may displace in Abaqus with respect to its local axis. This displacement will be updated in the XFlow GUI. Although initial position and orientation are available in the GUI, the current implementation is limited to geometry with (0,0,0) initial Position and Orientation.

5.4.2 Arbitrary reference frame

Project Tree > Geometry > Arbitrary reference frame

Simulations involving enforced geometries can be computed in arbitrary reference frame including non-inertial reference frame. The option is shown when all the simulation geometries in the project are set to [enforced](#), or there is a geometry with [rigid body dynamics](#) behavior, either for internal or external simulations.

The geometry selected as reference frame can be designated via the option:

Project Tree > Geometry > Arbitrary reference frame > Reference geometry

 **Please note:** For internal simulations including a single enforced geometry, the arbitrary reference frame will automatically be set to that geometry. However, if a rigid body dynamics geometry is added, the arbitrary reference frame will be set to that geometry.

 **Please note:** It is not possible to define an arbitrary reference frame if two shapes are set to rigid body dynamics behaviour. In this case, the arbitrary reference frame dialog will disappear.

The selected geometry becomes the reference of the simulation, hence the geometry will become static in the Graphic View and instead body forces will be applied to model the geometry motion.

 **Tip:** The advantage of using arbitrary reference frame instead of actual moving geometries is to save computational time avoiding the geometry displacement, and allows to visualize the output data as if located on the selected arbitrary reference frame.

The possible applications that can be modeled using an arbitrary reference frame include:

- Sloshing
- Rotating systems
- Accelerating bodies
- Oscillating bodies
- Complex relative motions between bodies

 **Please note:** All the post-processing output will be provided in the arbitrary reference frame, including position and orientations in the Function Viewer.

 **Please note:** For external simulations, geometry used as an arbitrary reference should move towards inlet boundary conditions only in order to have consistent boundary condition treatment. It is recommended to use a generic rectangular domain and apply inlet boundaries in the directions of motion. Example: body translating towards -X should have -X set as velocity boundary condition.

 **Please note:** Rotations are supported only in one direction hence change in rotation direction must be avoided.

 **Please note:** Geometries moving towards other boundary conditions (walls, periodicity, etc.) should only have very small displacement magnitude to remain consistent and domain size should be large enough. Geometries moving towards outlet boundary conditions should be avoided. Example: body with sinus rotation or translation in a wind tunnel flow.

 **Tip:** when using tabulardata input in the motion laws please follow these guidelines:

- use interpolated data (tatabulardatacubicinterpolated is suggested)
- include an extrapolation of the law for $t < 0$ for at least 1 timestep
- data should be sampled with more significant digits than the timestep (e.g. 0.002916 and not 0.0029 for a $dt=0.0025$)

5.4.3 Cables

Project Tree > Geometry > Cables



For shapes with [rigid body dynamic](#) simulation, the cable feature allows the user to include cables (with physical properties) interacting with solids and joints.

To add a cable: **Right click** on **Project Tree > Geometry > Cables**; it appears a menu with the option: **Add cable**, click on it and a new cable will appear on the list.

To remove a cable: **Right click** on the cable to be removed **Project Tree > Geometry > Cables**; it appears a menu with the option: **Remove cable**, click on it and the cable will be removed from the list.

Cables are modeled by a junction of short solid segments which number is automatically calculated by XFlow. Each segment follows the damped harmonic oscillator equation:

$$F = -kx - c \frac{dx}{dt} = m \frac{d^2x}{dt^2}$$

Damped harmonic oscillator equation

 **Remember:** Cables only interact with solids and joints. They do not interact yet with the fluid.

The data structure of a cable is as follows:

- Cable**
 - Name
 - Geometry ([+Info](#))
 - Material ([+Info](#))
 - Interaction properties ([+Info](#))
 - External force global ([+Info](#))

Geometry

- ↳ **Point1/Point2:** Coordinates defining both ends of a straight cable. The cable must be straight initially but will be then able to move according to the parameters.
- ↳ **Radius:** The cable section is circular. The thickness is therefore set as 2*radius.
- ↳ **Number of segments:** number of azimuthal segments used to discretize the circular cross section of the cable.

 **Tip:** When using [joints](#), it is recommended to locate them inside the cable to make sure they are effective.

 **Tip:** It is recommended to leave space between geometries and cables that are joined in order to prevent from undesired collisions. Contacts between geometries and cables are always enabled.

Material

- ↳ **Density:** Density of the cable material.
- ↳ **Young modulus:** Young modulus of the cable material. It characterizes the stiffness of an elastic material.
- ↳ **Second moment of area:** the second moment of area J_l of the cross-section of a cable along the

$$J_l = \int_A n^2 dA$$

line l is defined as:

where A is the cross section area of the cable and n the perpendicular distance of the element dA from the line l . Typically, for a cable of circular section the second moment of area is such as:

$$J = \frac{\pi r^4}{2}$$

- ↳ **Torsional rigidity:** it is the product $J_T * G$ where J_T is the torsion constant for the section and G the shear modulus.
- ↳ **Damping:** the damping (in N.s/m) allows to damp the oscillations produced by elasticity of the cable. The damping is the constant c in the above [damped harmonic oscillator equation](#) and it is related to the cable material properties through the following expression:

$$c = 2\xi\sqrt{mk} = 2\xi\sqrt{\frac{mEA}{L}}$$

where ξ is the damping ratio, m the mass of the cable, E the Young modulus, A the cross section and L the length of the cable. If $\xi = 1$, the cable will be in critical damping regime, if $\xi < 1$ it will be under damped, and $\xi > 1$ will be over damped.

 **Please note:** The maximum damping allowed is $c = (0.5 \cdot L) / (dt \cdot m)$ where dt is the smaller time step, L the cable length, and m the cable mass.

Interaction properties

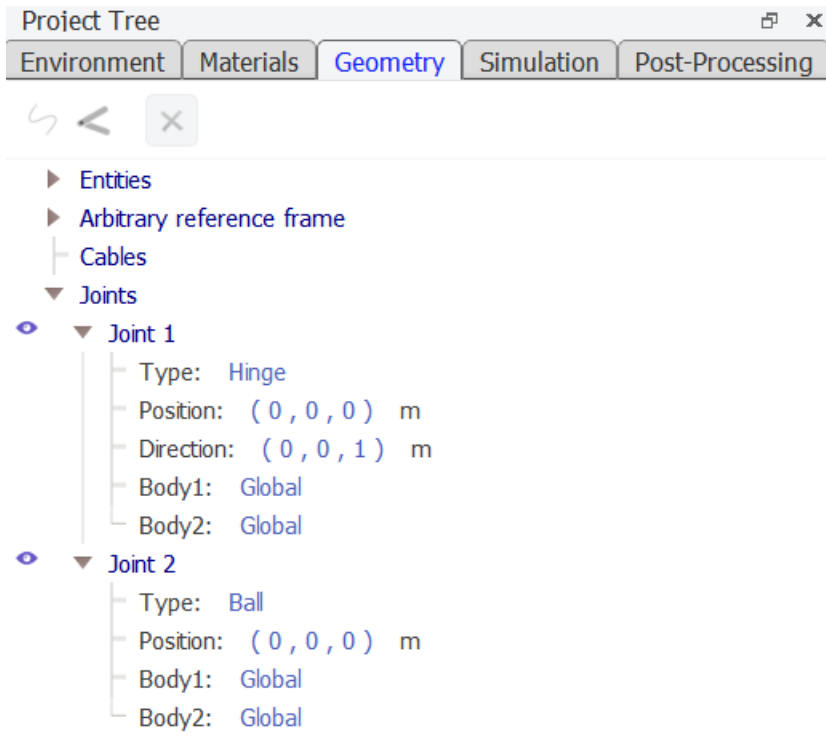
- ↳ **Static friction coefficient:** the static friction coefficient between two surfaces of solids is defined as the ratio of the tangential force (F) required to make an object sliding over the other divided by the normal force between the surfaces (N): $sfc = F/N$. Basically, this means a static friction coefficient of 1 will require strong forces to move two static objects in contact, and a coefficient of 0 will let them move immediately when a force is applied, as a slippery surface. (dimensionless)
- ↳ **Dynamic friction coefficient:** The dynamic friction coefficient is the same than the static friction coefficient but for two objects in motion. Even though they are in motion, the force of friction cannot be eliminated completely and it continues to be a resistive force to motion. (dimensionless)
- ↳ **Coefficient of restitution:** solid/solid contact coefficient (dimensionless). It expresses how one solid colliding another will “bounce” on it due more or less restitution of its energy. If this coefficient is equal to 1, the object will bounce such as a perfectly elastic solid since all the energy will be returned to the “bouncing” solid. If it is equal to 0, it will not bounce and stick to the other object which will absorb all its energy.

External force global

- ↳ **X,Y,Z:** Coordinates of an external force with respect to the global axes. It can be useful to include for example the gravity force.

5.4.4 Joints

Project Tree > Geometry > Joints



Joints are designed to constrain two geometries together, or a geometry to a given fixed point.

⚠ Please note: Joints are only available when at least one **Shape** has been set to [rigid body dynamics behavior](#) or one [cable](#) has been created.

To create a joint: **Right click** on **Joints** and select **Add joint**. A new joint will appear on the list.

To remove a joint: **Right click** on the joint to be removed and select **Remove joint**. The joint will be removed from the list.

Joints can be of different types:

- **Hinge:** this requires a point (**Pos**) and a direction (**Dir**) in order to define an axis. The two objects are then free to move around the axis but keep their distance to it. As an illustration, the constraint is similar the operation of a door where an axis constraint two objects.

⚠ Please note: The hinge joints currently cannot be fixed to cables. To join a cable, please use a ball joint.

- **Ball:** this requires only a point (**Pos**). The two objects are free to move around the point, but keep a constant distance to the point.

For any kind of joint, two bodies must be constrained together: **Body1** and **Body2**. One can choose any geometry set in rigid body dynamics or enforced behavior, as well as **World**. World defines the background

and let fix the object to a fixed joint, whereas selecting a geometry may induce a displacement of the joint.

 **Tip:** When using [cables](#), it is recommended to locate the joints inside the cable to make sure they are effective.

5.5 Simulation

Project Tree > Simulation

The **Simulation** tab of the Project Tree allows the user to set up both the temporal and spatial discretisation resolution; and to specify how (where & when) to save the simulation results.

The tree structure of this tab is as follows:

Time (+Info)	Simulation Time (+Info)	
	Time step mode (+Info)	Fixed automatic Fixed custom
Lattice discretization (+Info)	Resolved scale (+Info)	
	Refinement algorithm (+Info)	Disabled Near static walls Adaptive refinement
	Shapes options (+Info)	Target resolved scale Seal small gaps
	Adaptive refinement (+Info)	Wake resolution Wake distance control Wake refinement threshold
	Regions (+Info)	Seed point (+Info) Seal small gaps (+Info)
	Options (+Info)	Refinement transition length (+Info) Buffer zone length (+Info) Refinement based on simulation data (+Info)
Store data (+Info)	Frames frequency (+Info)	
	Numerical data frequency (+Info)	Solver time step Frames frequency Highest available frequency
	Save averaged fields (+Info)	Initial time for averaging Save standard deviation and RMS fields Save axis force distribution
	Save resume file (+Info)	
	Compute markers (+Info)	Native XFlow CGNS
	Output format (+Info)	VTK

Fields to save

Vorticity
 Static pressure
 Total pressure
 Turbulence intensity
 Effective viscosity
 Temperature
 Q-criterion

Static pressure filters
 Probes ([+Info](#))
 Solver surface integrals ([+Info](#))
)
 Solver volume integrals ([+Info](#))
)
 Cutting planes

5.5.1 Time

XFlow is intrinsically transient, therefore time discretization must be defined.



Simulation time

Project Tree > Simulation > Time > Simulation Time

This parameter sets the temporal domain of the simulation. The physical time to be simulated.

Time step mode

Project Tree > Simulation > Time > Time step mode

The time discretisation resolution is specified by selecting one of the following time-step modes:

- **Fixed automatic:** XFlow automatically estimates the time step that will remain constant during the simulation. This is calculated taking into account the initial maximum velocity and pressure gradient in the domain, the minimum cell size and the value of the Courant number given by the user in:
 - ↳ **Courant:** By default it is set to 1 because it is the stability limit. For analysis where other quantities are more relevant for the time step estimation (e.g. surface tension or solids movement) it may be necessary to decrease the Courant number. In this case the [Message View](#) will show the following message:

Warning! Time step too big. Please, try to run again the computation with a lower Courant number!!!

The Courant Number is the main control over the time step scheme. Larger time steps lead to faster computation, so it is advantageous to set the Courant Number as high as possible. On other hand smaller numbers mean a more stable solution but it will be slower since it is doing more steps.

 **Tip:** For internal simulations where pressure boundary conditions are imposed at both inlet and outlet, the estimated time step may be too low. In this case, it is recommended to adjust a fixed custom time step or increase the Courant number to reach a [Stability parameter](#) around 0.1 - 0.3.

- **Fixed custom:** The user defines a constant time step for the whole simulation:
 **Time step:** By default it is set to 0.001 s.

 **Please note:** The time step, either estimated by XFlow (Fixed automatic) or given by the user (Fixed custom) corresponds to the biggest resolution of the lattice. Other resolution levels are automatically created using spatial and temporal resolutions twice smaller than the previous level, forming the octree structure.

 **Please note:** If the **Acoustics analysis** option is enabled, the time step can only be set as Fixed automatic. The time step is forced in order to allow the pressure waves to travel with the right Thermodynamic speed of sound (see [Acoustics analysis](#)).
 The time step can be changed if the [Time integration scheme](#) is Fractional propagation (**Environnement > Engine > Advanced options > Time integration scheme: Fractional propagation**) available in [Labs mode](#).

Example 1:		
Time step mode	→	Fixed custom
Time step	→	0.1 s
Frame frequency (+Info)	→	5 Hz; fulfilling $f = 1 / (dt * m)$, m being a natural number: $f = 5 = 1 / (0.1 * 2)$
Temporal discretisation octree structure:		
Level 0 (n= 0)	→	Coarsest level
dt = 0.1 s	→	Temporal resolution = Time step
Level 1 (n= 1)	→	Intermediate level:
dt = 0.05 s	→	Temporal resolution = (Time step / 2 ¹)
Level 2 (n= 2)	→	Finest level:
dt = 0.025 s	→	Temporal resolution = (Time step / 2 ²)

Example 2:

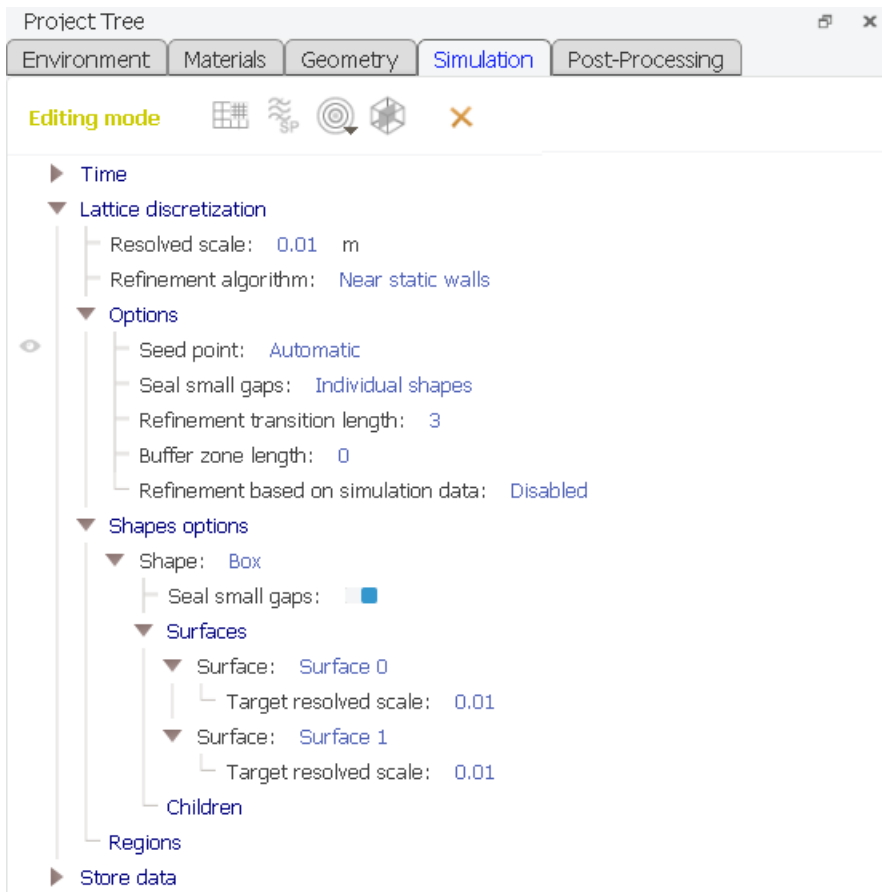
Time step mode	→	Fixed custom
Time step	→	0.1 s
Frame frequency (+Info)	→	3 Hz; not fulfilling $f = 1 / (dt * m)$, m being a natural number; $m = 3.333$

Temporal discretisation octree structure:

Level 0 (n= 0)	→	Coarsest level
dt = 0.083333 s	→	Temporal resolution = $1 / (f * m)$ where $m = \text{ceil}[1 / (f * \text{Time_step})]$? Time step (+Info) ↳ $m = \text{ceil}[1 / (3 * 0.1)] = 4$ ↳ $dt = 1 / (3 * 4) = 0.083333$
Level 1 (n= 1)	→	Intermediate level:
dt = 0.041666 s	→	Temporal resolution = ("Temporal resolution at level zero" / 2^1)
Level 2 (n= 2)	→	Finest level:
dt = 0.020833 s	→	Temporal resolution = ("Temporal resolution at level zero" / 2^2)

5.5.2 Lattice discretization

In XFlow, as it is based on the Lattice-Boltzmann method, the discretisation of the spatial domain results in the lattice (or domain structure). This is automatically generated according to the following parameters:

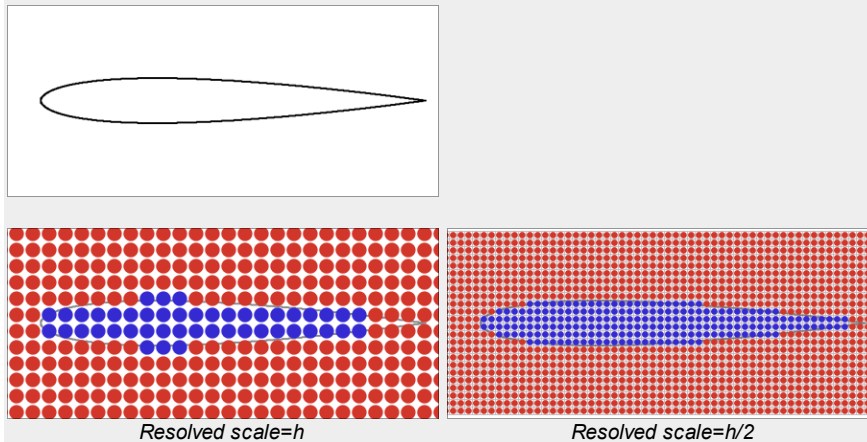


Resolved scale

Project Tree > Simulation > Lattice discretization > Resolved scale

Spatial resolution at the far field, i.e. the resolution of the largest level of the lattice. The smaller it is, the finer will be the spatial discretisation of your domain and thus the more accurate. However, the simulation time will be large as more elements must be computed.

Example: NACA airfoil (I)



In the example, every red circle represents a voxel of fluid whereas the blue circles represent the NACA boundaries. The discretisation of the NACA airfoil is very coarse and inaccurate, thus mesh refinement would be recommended to get a definition of the geometry boundaries as the one shown on the right hand side of the figure above.

XFlow provides different [refinement algorithms](#) to refine the solution only at some areas where details are required; these is recommended since it reduces the overall number of elements, the amount of memory and the computation time.

Refinement algorithm:

Project Tree > Simulation > Lattice discretization > Refinement algorithm

To improve the quality of the lattice, it can be locally refined according to the following options:

- **Disabled:** It uses uniform resolution in the entire domain using the **Resolved scale** above.
- **Near static walls:** To use a finer resolution than the **Resolved scale** in the regions close to geometries. Every [Shape](#) resolution can be specified independently by the user.

⚠ Please note: The near static walls is a static domain topology and cannot follow moving geometries. Use the adaptive refinement to refine dynamically moving objects.

- **Adaptive refinement:** This algorithm is a dynamic domain topology. As for the near static walls it refines in regions close to geometries, but besides refinements adapts dynamically to moving geometries following their motion. Furthermore, it can dynamically refine the wake generated.

⚠ Please note: The adaptive refinement is not available yet for distributed computations (MPI). Use the disabled or near static walls algorithms for distributed computation, eventually using [refinement regions](#).

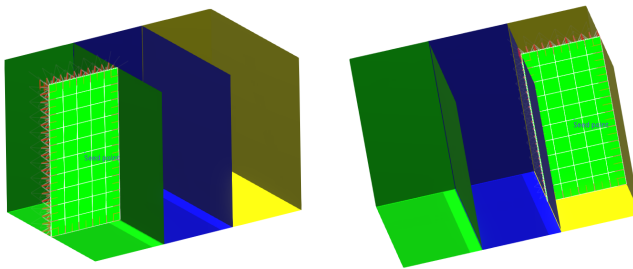
Options

Seed point *(available for [internal simulations](#) only)*

Project Tree > Simulation > Lattice discretization > Options > Seed point

Internal simulations require XFlow to identify the different internal volumes and generate the fluid domain in them. In some case it may be convenient to discard some volumes from the simulations, or to help the domain generation pointing to the fluid region. To point out where the fluid region is placed, there are two options:

- **Automatic:** XFlow will detect automatically the different regions of fluid.
- **Specify:** the user can point out a region of fluid giving the coordinate of a single point:
 ↳ **Position:** (X, Y, Z)



Example of two overlapping boxes with seed point on the left side and seed point on the right side

Seal small gaps

Project Tree > Simulation > Lattice discretization > Options > Seal small gaps

This feature allows to tag geometries in the XFlow setup so that the lattice discretization process can ignore holes and gaps smaller than the specified resolution in their surface tessellation. This way, if the geometry has small holes/gaps which should not be resolved, these will be ignored during the lattice generation process.

It is important to highlight that the original geometry will not be modified (i.e. the holes/gaps will still be visible), but the lattice generation algorithm will not detect those features as holes/gaps.

There are four options available:

- **Automatic:** Only applies to geometries generated by the Hole filling tool.
- **Disabled:** Does not apply.
- **All shapes:** Applies to all geometries.
- **Individual shapes:** Applies only to shapes with the option **Seal small gaps: On** in the [Shapes Options](#).

 **Tip:** If Automatic mode is set, the Seal small gaps feature will be activated only on those geometries containing the string "holeFilling" in their names. This is the case for all the patches geometries output from the [Hole filling tool](#).

Discard narrow isolated regions

Project Tree > Simulation > Lattice discretization > Options > Discard narrow isolated regions

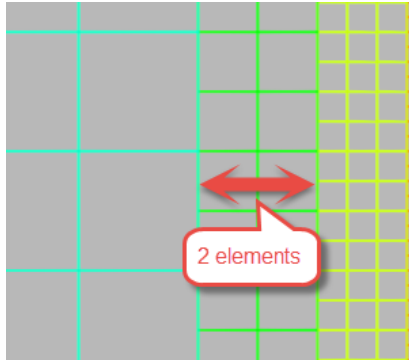
If enabled, the XFlow solver will detect automatically the narrow and isolated regions of fluid to discard them from the simulation since they can be source of instabilities. These regions are usually fluid regions trapped between a moving geometries and another one, and may generate numerical instabilities since highly compressed on the narrow region. Example: small gap between two rotating shafts, etc.

Refinement transition length

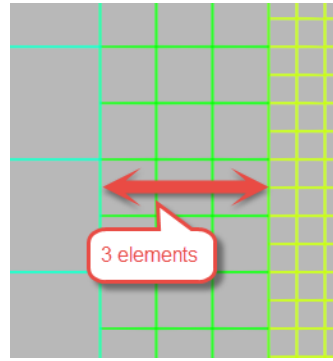
Project Tree > Simulation > Lattice discretization > Options > Refinement transition

length

The refinement transition length (rtl) refers to the number of element layers between two refinement levels, i.e. it represents the gradient in the transition from fine resolution at the walls to coarse resolution in the far field. Small rtl values lead to quickly growing element sizes while large rtl values lead to smooth transition but more elements.



Refinement transition length (green resolution level): $rtl = 2$

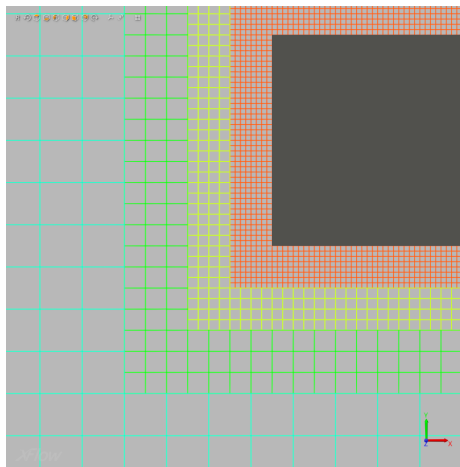


Refinement transition length (green resolution level): $rtl = 3$

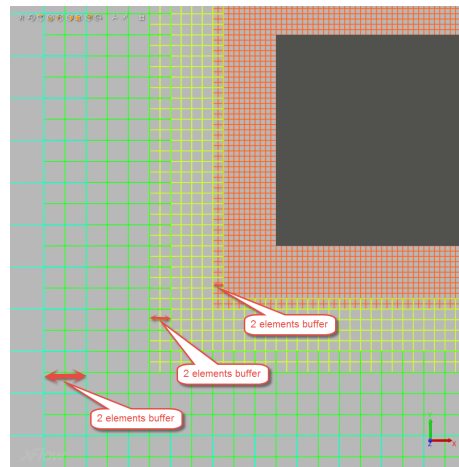
Buffer zone length

Project Tree > Simulation > Lattice discretization > Options > Buffer zone length

This option is only available when the refinement algorithm is set to *Near static walls*. If enabled, every refinement level will overlap the finer adjacent one by a distance equal to the buffer-zone-length times the finer-level-resolution. The introduction of this simple modification is expected to alleviate the discontinuities appearing between refinement levels.



No overlapping



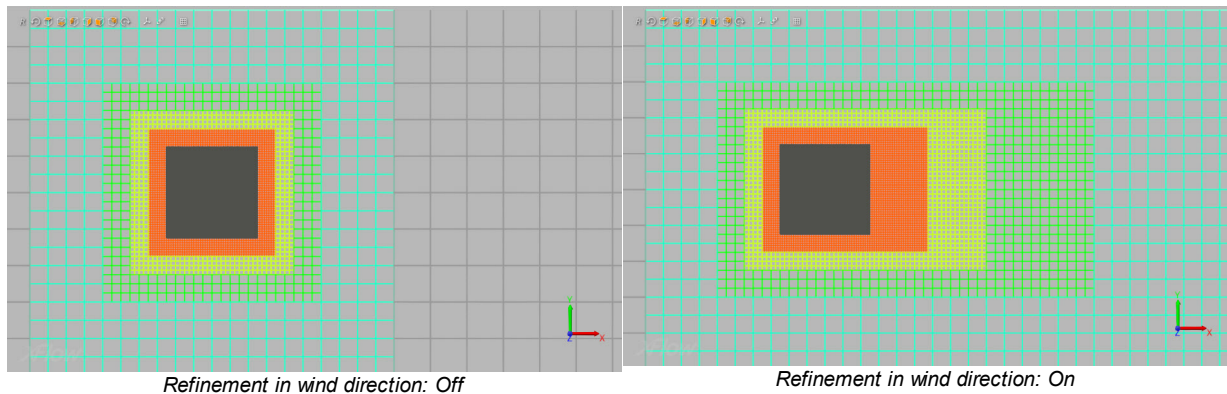
Buffer zone length = 2

⚠ Please note: The buffer zone length can only be an even number of lattice nodes due to the octree structure of the lattice.

Refinement in wind direction

Project Tree > Simulation > Lattice discretization > Options > Refinement in wind direction

This option is only available when using the [Wind Tunnel](#) and **Near static walls** as refinement algorithm. It refines with a few additional cells in the direction of the wind (from -X to +X)



Percentage of the total domain volume that will be refined, i.e. if *Threshold* = 50 %, half of the initial domain will be refined. Refinement is performed from high vorticity to low vorticity zones up to reach these threshold.

Refinement based on simulation data

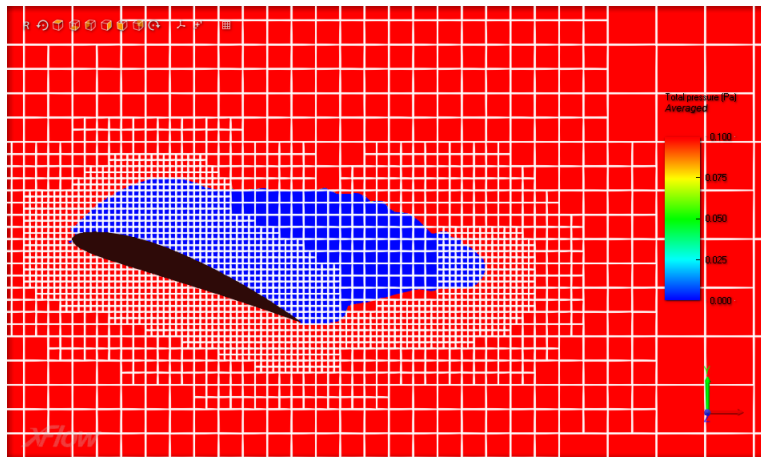
Project Tree > Simulation > Lattice discretization > Options > Refinement based on simulation data

This option allows the user to refine the initial domain based on values of the Average Total Pressure field calculated in a former simulation. It does not imply the initialization of the simulation from a previous solution ([+info](#)), which has to be setup in the **Environment** tab, under **Initial condition** ([+info](#)).

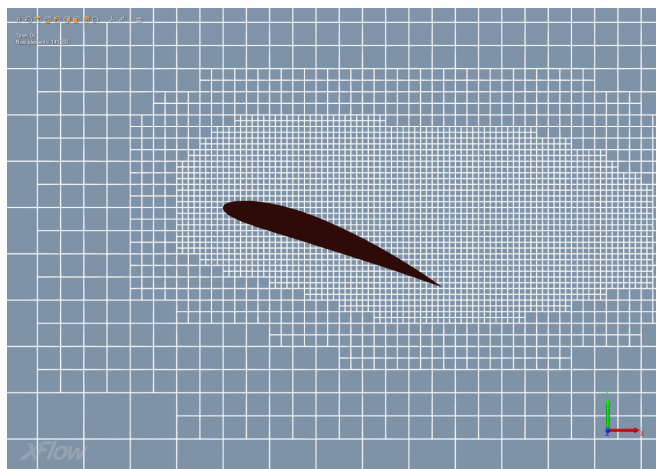
The user can set the following parameters:

- ↳ **Simulation folder:** Path of the folder containing the results as it is seen by the machine where the simulation will be run.
- ↳ **Simulation frame:** Frame number of the simulation defined in **Folder**.
- ↳ **Resolver scale:** Wake refinement scale in [m].
- ↳ **Threshold:** Criteria used to perform the lattice refinement. Two values can be chosen:
 - **Pressure coefficient:** the total pressure coefficient will be used as criteria.
 - **Absolute value:** the total pressure value will be used as criteria:
- ↳ **Threshold value:** value of the Threshold Criteria.
- ↳ **Refinement symmetrization:** Force the generated refinement region to be symmetric. Possible choices are:
 - **+XY:** XFlow will use the domain in the +XY region and duplicate it in the -XY.
 - **+XZ:** XFlow will use the domain in the +XZ region and duplicate it in the -XZ.
 - **-XY:** XFlow will use the domain in the -XY region and duplicate it in the +XY.
 - **-XZ:** XFlow will use the domain in the -XZ region and duplicate it in the +XZ.

This option is available for *Single Phase* flows when the [Refinement algorithm](#) is set to **Near static walls**.



Simulation 1: Adaptive Refinement - Average Pressure Plot (highlight on TP=0)



Simulation 2: Refinement based on Average Total Pressure = 0 of Simulation 1

Shapes options

Project Tree > Simulation > Lattice discretization > Shapes options

This section allows to apply options to specific shapes (local refinement, etc.).

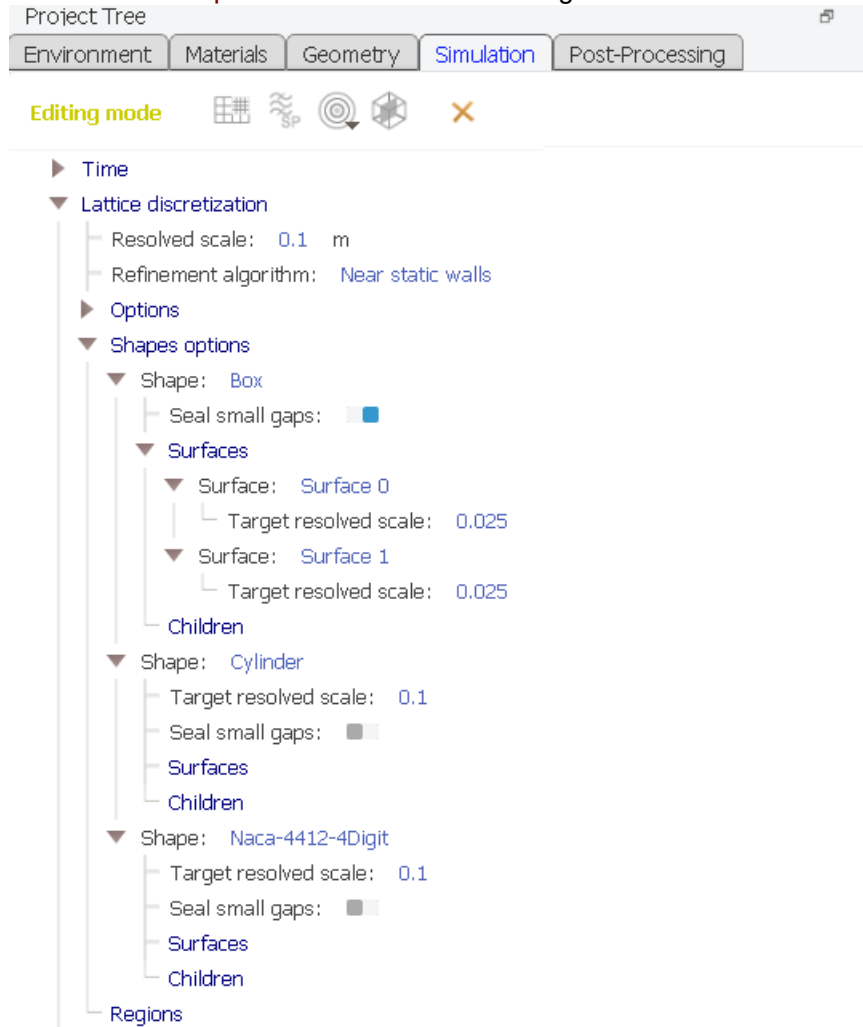
Refinement near geometry wall is activated either in *Near static walls* or *Adaptive refinement*. A list of the geometry objects involved in the simulation will appear to allow the user to define the resolution for every **Shape** according to the following parameters:

↳ **Target resolved scale:** Starting at the far field scale (**Resolved scale**), XFlow progressively reduces the resolution size by a factor of 2 to approach the closer resolution to the **Target resolved scale**.

Due to the lattice structure, each level of refinement is half size than the upper one. Hence, the **Target resolved scale** should be: $x/(2^n)$, x being the Resolved scale and n a positive integer. If the user-defined **Target resolved scale** do not satisfy this rule, XFlow will automatically replace it for the closest superior number that fulfills it.

Refine a specific surface of a geometry

To refine a specific surface or group of surfaces, use the [Apply boundary condition to faces](#) option on the surface of interest. For each shape, the structure of the boundary conditions groups is shown in the **Shapes refinement** tree allowing to refine surfaces with different resolutions.

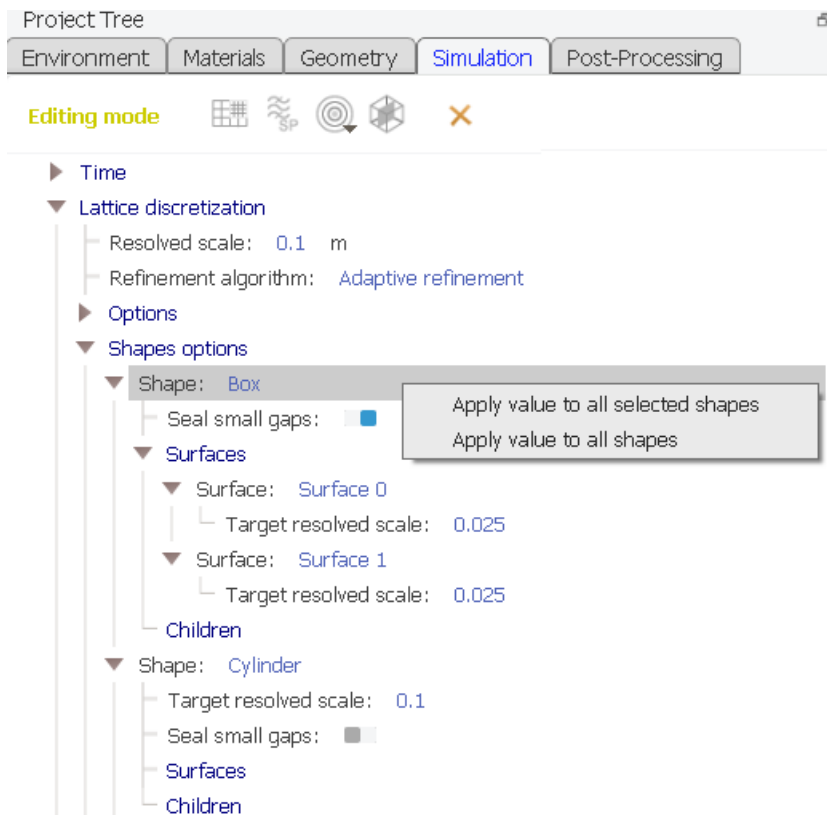


Refinement structure for a shape with different boundary conditions

Apply resolved scale on several shapes

If the same **Target resolved scale** is to be applied on several **Shapes** and/or **Surfaces**, the user can do it quickly by right clicking on the considered Shapes and/or Surfaces and selecting any of the options shown in the drop-down menu:

- **Apply value to selected shapes**
- **Apply value to all shapes**



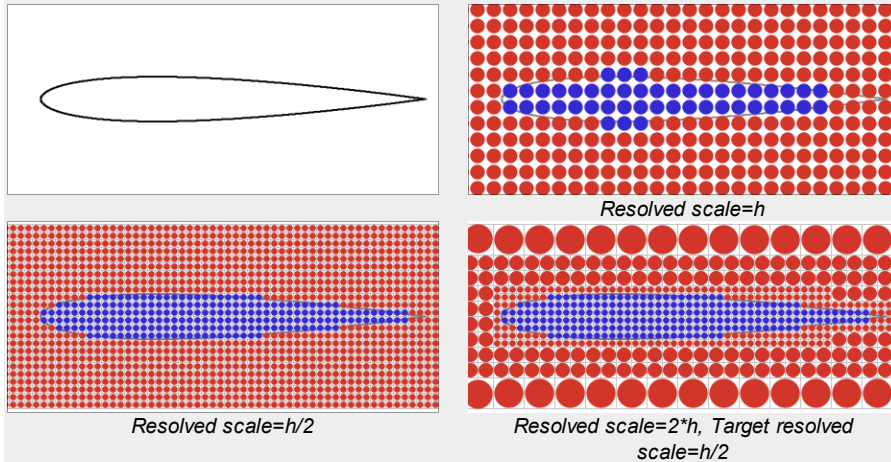
Target resolved scale: options

The resolved scale from the Shape or Surface on which the right-click is done will be applied to the others.

Rule: Resolved scale $\geq$ Wake resolution $\geq$ Target resolved scale (Shape).

Please note: Due to the lattice structure, each level of refinement is half size than the upper one. Hence, if the **Resolved scale** is set to x , for finer refinement levels (**Wake resolution**, **Target resolved scale**, **Minimum scale**) the resolution should be: $x/(2^n)$, n being is a positive integer. If any of the user-defined resolution-values do not satisfy this rule, XFlow automatically replace it for the closest superior number that fulfills it.

Example: NACA airfoil (II)



Example 1:

Resolved scale	→	1 m
Refinement algorithm	→	Near static walls
Target resolved scale	→	0.25 m; fulfilling $x/(2^n)$, x being the Resolved scale and n a positive integer: $n=2$.

Spatial discretisation octree structure:

Level 0 (n= 0) dx = 1	→	Coarsest level
	→	Spatial resolution = Resolved scale (+ Info)
Level 1 (n= 1) dx = 0.5 m	→	Intermediate level:
	→	Spatial resolution = (Resolved scale / 2^1) (+ Info)
Level 2 (n= 2) dx = 0.25 m	→	Finest level:
	→	Spatial resolution = (Resolved scale / 2^2) = Target resolved scale (+ Info)

Example 2:		
Resolved scale	→	1 m
Refinement algorithm	→	Near static walls
Target resolved scale	→	0.06 m; not fulfilling $x/(2^n)$, x being the Resolved scale and n a positive integer: ↳ $n = 1$ Target resolved scale = 0.5 ↳ $n = 2$ Target resolved scale = 0.25 ↳ $n = 3$ Target resolved scale = 0.125 ↳ $n = 4$ Target resolved scale = 0.0625 (Larger and closest number) ↳ $n = 5$ Target resolved scale = 0.03125
Spatial discretisation octree structure:		
Level 0 ($n = 0$)	→	Coarsest level
$dx = 1$	→	Spatial resolution = Resolved scale
Level 1 ($n = 1$)	→	Intermediate level:
$dx = 0.5$ m	→	Spatial resolution = (Resolved scale / 2^1)
Level 2 ($n = 2$)	→	Intermediate level:
$dx = 0.25$ m	→	Spatial resolution = (Resolved scale / 2^2)
Level 3 ($n = 3$)	→	Intermediate level:
$dx = 0.125$ m	→	Spatial resolution = (Resolved scale / 2^3)
Level 4 ($n = 4$)	→	Finest level:
$dx = 0.0625$ m	→	Spatial resolution = (Resolved scale / 2^4) ≠ Target resolved scale (+ Info)

Adaptive refinement:

Wake resolution

Project Tree > Simulation > Lattice discretization > Adaptive refinement > Wake resolution

If the refinement algorithm is set to *Adaptive refinement*, the wake is dynamically refined at the given wake resolution. The wake refinement criteria is based on the level of vorticity: XFlow dynamically refines the regions of high vorticity which are characteristics for wake regions.

⚠ Please note: In free surface flows, the Wake resolution is called **Interface/wake resolution** because the free surface is refined dynamically according to the wake resolution.

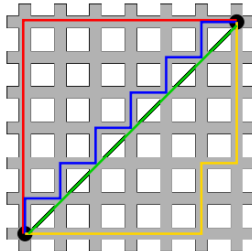
Wake distance control

Project Tree > Simulation > Lattice discretization > Adaptive refinement > Wake distance control

If the refinement algorithm is set to *Adaptive refinement*, the wake distance control can be enabled to set a maximum distance from the object (specified in **Distance**: [m]) up to which the wake refinement will take place. The Manhattan distance (the distance between two points in a grid based on a strictly

5.5 Simulation

horizontal and/or vertical path) is used to calculate the distance from the object to any lattice node in order to impose such condition.



Manhattan distances (red, blue, and yellow) in a grid compared to the diagonal distance (green)

Wake refinement threshold

Project Tree > Simulation > Lattice discretization > Adaptive refinement > Wake refinement threshold

If the refinement algorithm is set to *Adaptive refinement*, the wake refinement is activated when the local dimensionless vorticity is larger than a threshold value.

ω , Vorticity [s^{-1}]

ω^* , Refinement threshold [m^{-2}]

$$\omega^* = \left(\frac{\omega}{dx/dt} \right)^2$$

Threshold can be set to (only in [expert mode](#)):

☐ **Automatic:** By default is 0.1. Most case will work for this recommended value.

☐ **Custom:** The user can set up the value of the threshold manually in:

↳ Threshold

Tip: If the wake refinement is not sensitive enough, decrease the threshold value (e.g. 0.01). If the wake refinement is too sensitive, increase the threshold value (e.g. 0.5).

Please note: In free surface/multiphase flows, the wake refinement threshold is called **Interface/wake refinement threshold** since it refines dynamically the free surface.

Regions

Project Tree > Simulation > Lattice discretization > Regions

This option is only visible if the refinement algorithm is either *Near static walls* or *Adaptive refinement*. It allows the user to create regions of uniform refinement.

A new region is added by **right clicking** the **Regions** string, or by clicking the icon

A **Region** is defined by the following parameters depending on its **Type**:

- **Rectangular:**

- ☐ **Min:** Coordinates of the lower corner

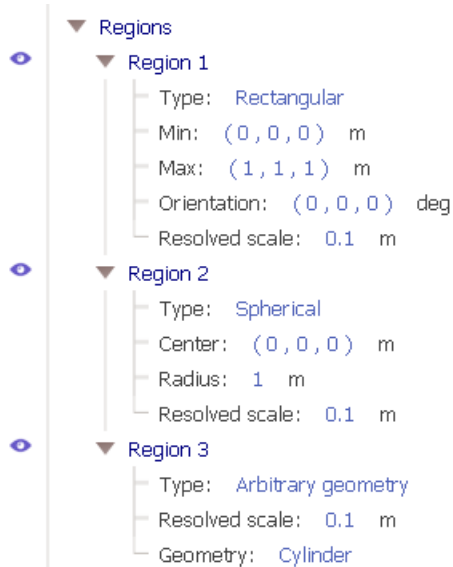
- ☐ **Max:** Coordinates of the upper corner

- ☐ **Orientation:** Orientation of the geometry in global axis
- ☐ **Resolved scale:** Lattice resolution within the region
- **Spherical:**
 - ☐ **Centre:** Coordinates of the sphere center
 - ☐ **Radius:** Length of the sphere radius
 - ☐ **Resolved scale:** Lattice resolution within the region
- **Cylindrical:**
 - ☐ **Center:** Coordinates of the cylinder center
 - ☐ **Direction:** Direction vector of the cylinder
 - ☐ **Height:** Cylinder height (length) in meter
 - ☐ **Radius:** Cylinder radius in meter
 - ☐ **Resolved scale:** Lattice resolution within the region
- **Tubular:**
 - ☐ **Center:** Coordinates of the tube center
 - ☐ **Direction:** Direction vector of the tube
 - ☐ **Height:** Tube height (length) in meter
 - ☐ **Outer radius:** Tube outer (maximum) radius in meter
 - ☐ **Inner radius:** Tube inner (minimum) radius in meter
 - ☐ **Resolved scale:** Lattice resolution within the region
- **Arbitrary geometry:**

The user can specify a geometry as an arbitrary region of uniform refinement.

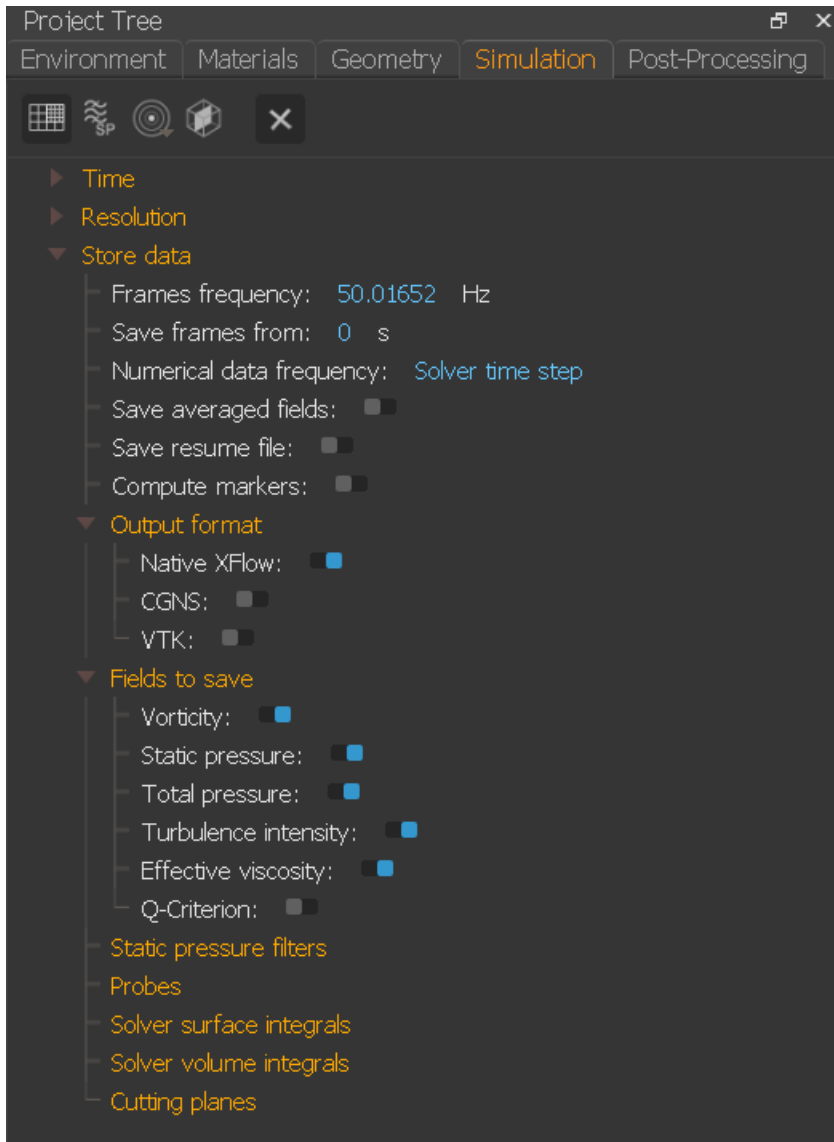
 - ☐ **Resolved scale:** Lattice resolution within the region
 - ☐ **Geometry:** Name of the geometry used as an arbitrary region

⚠ Please note: The option [Enable boundary conditions](#) must be deactivated on a geometry that wants to be used as an arbitrary region of refinement.



5.5.3 Store data

Simulation data is saved according to the parameters specified in the **Simulation** tab of the Project Tree.



Folder

XFlow automatically creates a folder using the basename of the saved project file. This is where the computational data will be saved

For more information about the files saved in the computation folder see: [XFlow files](#).

Frames frequency

Project Tree > Simulation > Store data > Frames frequency

Number of frames per second (of simulated time) that have to be saved. This is the frequency at which volumetric and surface data are output.

⚠ Please note: Time step (dt) and Frame frequency (f) are related as follows: $f = 1 / (dt * m)$; m being a natural number. If Time step mode = Fixed custom and the above expression is not satisfied, XFlow will automatically adjust the Time step value as: $dt_{new} = 1 / (f * m)$ where $m = \text{ceil}[1 / (f * dt_{custom})]$. See Example 2 in [Time](#).

⚠ Please note: Frames frequency is automatically adjusted when [Acoustics analysis](#) is enabled. Since the time step is selected automatically by XFlow and is unique, the frames frequency is adjusted automatically to be the closest multiple. See the [Acoustics analysis](#) chapter for more information.

Save frames from

Project Tree > Simulation > Store data > Save frames from

Allows the user to select a specific time in the simulation on which to start saving frames

Numerical data frequency

Project Tree > Simulation > Store data > Numerical data frequency

Frequency at which the numerical data plotted in the Function Viewer (stability parameter, forces, probes, etc) are saved in the file `numericaldata.xml` ([XFlow files](#)).

Since lattice-Boltzmann uses sub-time-steps, the data is available at different frequency depending on the local time steps. For this reason different options are available:

- **Solver time step:** saves the numerical data at the coarser lattice level time step ([Resolution > Resolved scale](#)). Every time step of level 0 will add a new plot to the numerical data graphs. This is the recommended option in most cases since it is a good compromise between a too high and too low frequency.
- **Frames frequency:** saves the numerical data at the frame frequency defined ([Store data > Frames frequency](#)). This option is useful only if you are not interested in the numerical post-processing, you numerical data graphs may have a too low number of points for a good curve description.
- **Highest available frequency:** saves the numerical data at the finest lattice level set in the domain. Indeed, since in the lattice-Boltzmann method each finer lattice level has a sub-time-step twice smaller, the finest lattice level has the smallest time step and therefore the highest resolution frequency. It may be useful to save numerical data at such high frequency in cases where you need good sampling of your signals, such as for acoustics for instance. This option is available in [Expert mode](#).
- **Arbitrary lattice level:** saves the numerical data at the time step of a given lattice level. Indeed, since in the lattice-Boltzmann method each finer lattice level has a sub-time-step twice smaller, the finest lattice level has the smallest time step and therefore the highest resolution frequency. It may be useful to save numerical data at intermediate lattice levels in case [Probes](#) are located in intermediate lattice levels for instance. This option is available in [Expert mode](#).

⚠ Please note: Curves may show in staircase when options [Highest available frequency](#) and [Arbitrary lattice level](#) are used. This happens when the local data is available at a lower frequency than the one define: since all numerical data are output with the same length some values are repeated. For instance, if a probe is located out of the finest level level most of the signal will show staircase. To avoid this, you must either make sure your probe lying within the finest lattice level, either switch to "Arbitrary lattice level".

Tip: Solver time step is recommended for most cases, but cases it may lead to a large numerical data file (`numericaldata.xml`) when the solver frequency is high. Highest available frequency or arbitrary lattice level may be useful for acoustics simulations where the signal sampling frequency (e.g. probe static pressure) required must be high, and when the numerical data is located in the correct refinement region (e.g. probe in a refined region), but it may conduct to a very large numerical data file (`numericaldata.xml`).

Save averaged fields

Project Tree > Simulation > Store data > Save averaged fields

This option allows the user to save the averaged data in time. The averaged data is useful to study the steady state of the solution. It is recommended in many cases to start the averaging after a small amount of time to cut-off the initial transient peaks due to initialization of the solution. When enabled, the user can specify:

↳ **Initial time for averaging** (zero by default) This is the simulation time after which the computation of the averaged (and standard deviation/RMS if enabled) will start. Before that time the averaged, standard deviation and RMS fields will have no values.

Tip: It is useful to start the averaging after the transient period of the simulation in order to get a better steady-state solution.

↳ **Save standard deviation and RMS fields** (On/Off). To save the standard deviation and RMS volumetric fields in addition to the averaged fields. All fields (velocity, pressure, etc.) will be available as instantaneous, averaged, standard deviation, and RMS at the post-processing when this option is enabled.

The standard deviation σ represents the variation and dispersion around an averaged value. It is defined as:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

where x_i is the instantaneous field value and μ the averaged field value and N the number of time step computed since the starting of the averaging.

The Root Mean Square (RMS) is the statistical measure of the magnitude of a varying field. It is directly related with the averaged (`avg`) and the standard deviation (`std`) fields through the relation:

$$x_{\text{rms}}^2 = x_{\text{avg}}^2 + x_{\text{std}}^2$$

Please note: Only standard deviation is stored in the hard disk, the RMS field is computed based on the averaged and standard deviation fields.

↳ **Save axis force distribution** (On/Off). This allows the computation by XFlow of two additional averaged force values:

- **Force cumulation** (in X, Y or Z direction): this is force cumulation over all geometries along one direction, divided by the overall force in the considered direction. It is a non-dimensional number which starts from 0 value and ends to 1 at the maximum length point in the considered direction. It is useful to study the contribution of force increase/decrease along a geometry, e.g. drag cumulation along a body in aerodynamics. The force cumulation can be visualized in the [Function Viewer](#) and always shows the last time-

averaged (not instantaneous) force cumulation:

Right click on the *Function Viewer* window > **Axis force cumulation** > **Axis X/Y/Z** > **Fx/Fy/Fz**

- **Force distribution** (in X, Y or Z direction): this is the force distribution over all geometries along one direction. It shows the integrated force in two directions and how it distributed along the third direction, e.g. the force in the plane YZ along the direction X. The force distribution can be visualized in the [Function Viewer](#) and always shows the last time-averaged (not instantaneous) force distribution:

Right click on the *Function Viewer* window > **Axis force distribution** > **Axis X/Y/Z** > **Fx/Fy/Fz**

 **Please note:** in multi-resolution this feature is supported only when all geometries are refined with the finest resolution.

 **Please note:** XFlow saves the force cumulation/distribution data of the last time step in the files `axisforcesdistributionX/Y/Z.txt` available in the [simulation folder](#). These files are raw data, it is therefore recommended to get the force cumulation/distribution through the Function Viewer.

 **Please note:** Saving averaged and standard deviation fields will require three times more hard drive space than saving only instantaneous data.

Save resume file

Project Tree > Simulation > Store data > Save resume file

If enabled, a file named `resume.bin` is saved in the data [folder](#). This file is required to resume a computation that has been stopped before reaching the [Simulation Time](#) (end of the simulation). To resume the simulation from the last time calculated (time at which the simulation was stopped) press: Run>[Resume Computation](#).

Compute markers

Markers are mass-less particles advected by the flow field. When enabled, the user will be able to visualize the flow with [markers](#) in the post-process.

 **Please note:** The markers must be enabled before running the simulation to be available in the post-processing. Computation is quicker if they are disabled.

Output format

Only available in [Labs mode](#).

Fields to save

The volumetric fields (vorticity, static pressure, total pressure, turbulence intensity, effective viscosity, Q-criterion, etc.) storage into the data folder can be enabled/disabled. By default they are all enabled. To save hard disk space, it is recommended to enable only the fields of interest.

Static pressure filters

The static pressure filters are additional static pressure fields computed during the simulation applying a pre-defined filter. They can be plotted at the post-processing stage in addition to the original static pressure field. It is useful to get rid of undesired source of noise, or identify the noise generation

mechanism for instance.

Several filters can be computed for the same simulation. In order to create a filter: **Right click** on **Project Tree > Simulation > Store Data > Static pressure filters** and select **Add filter**.

Each filter requires a few input definition:

- ↳ **Filter type**: the type of the Finite Impulse Response (FIR) filter to apply on the signal displayed in the Function Viewer. The filters available: Low pass, Band-pass, Band-stop and High pass.
- ↳ **Min frequency**: minimum frequency of the filter in Hz.
- ↳ **Max frequency**: maximum frequency of the filter in Hz.
- ↳ **Filter order**: this is the order of the filter transfer function polynomial used in the convolution operation between the filter and the time signal. The higher the order is, the more accurate would be the approximation of the transfer function polynomial of the filter. However, a higher order will introduce more delay to the filter response and the calculation of the filtered signal will take a longer time by the Function Viewer.
The delay τ introduced by the Order can be estimated (in seconds) with this formula: $(\text{Order} \times \text{Simulation time-step}) / 2$.
- ↳ **Window type**: this is the window type to apply to the Fourier transform. The window functions available are: None, Hamming, Hann, Barlett, Blackman, flat top, Gaussian.

⚠ Please note: The higher the number of filter and the higher the computation time and hard disk usage. Furthermore, the higher the filter order and the longer the computation.

⚠ Please note: The maximum frequency is half of the solver frequency due to Shannon-Nyquist sampling theorem.

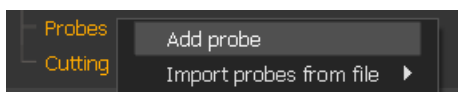
Probes

Probe is a predefined point where data are measured and saved during computation. The value of the flow variables at a probe can be plotted in the [Function Viewer](#).

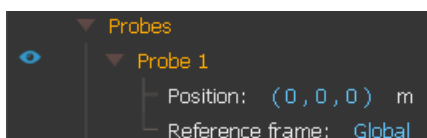
Single Probe

A single probe can be created by:

- **Main menu > Post-Processing > Import from file > Probes > Grouped**
- **Right click** on **Project Tree > Simulation > Store Data > Probes** and select **Add probe**.



then input the probe coordinates in [m], and select the reference frame with respect to which the probes are defined:



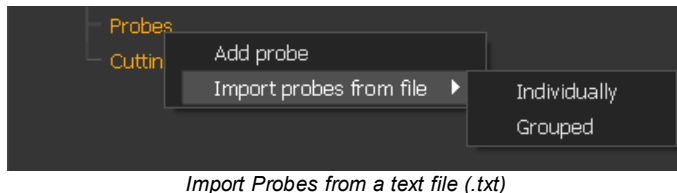
Import probes from file

A list of probes can be imported from a text file:

- **Main menu > Post-Processing > Import from file > Probes > Grouped**
- **Right click** on **Project Tree > Simulation > Store Data > Probes** and select **Import probe**

from file, then choose between **Individually** or **Grouped**.

⚠ Please note: Importing probes from file **individually** XFlow will prompt to select the desired reference frame to use.



The filename extension should be: **.txt**. The text file should contain the probes coordinates given as follows:

```
X-Coordinate-Probe-1 Y-Coordinate-Probe-1 Z-Coordinate-Probe-1
...
X-Coordinate-Probe-N Y-Coordinate-Probe-N Z-Coordinate-Probe-N
```

Example:

1	0	0
2	3	0
3	0	2

⚠ Please note: The file containing the probes location will be copied in the current working directory at saving. The XFlow GUI will remove the actual path from the Project Tree.
If the probes file cannot be read by the solver at runtimes the simulation will exit with an error:
[ERROR] Probes file is missing: [filename]. Aborting simulation.

Set probe by mouse

A single probe can be created by mouse control with a simple click:

Main menu > Post-Processing > Set probe by mouse, or 

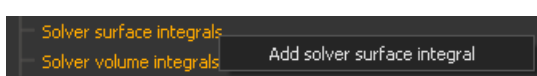
Once selected, click in the Graphic View with the mouse **middle button** to create the probe.

Solver surface integrals

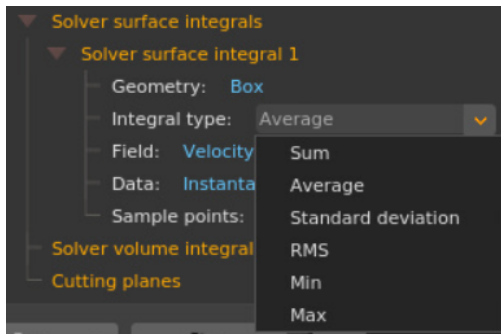
This feature allows to automatically compute and store surface integral values in the selected surfaces during the computation at the [numerical data frequency](#). The integral type, data field, data type (instantaneous or averaged) and number of sample points on the surface must be specified. For more information, please check the [Surface integrals](#) chapter.

A Solver surface integrals can be created by:

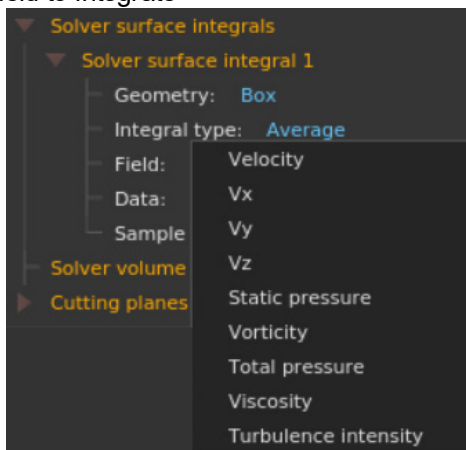
- **Right click** on **Project Tree > Simulation > Store Data > Solver surface integrals** and select **Add solver surface integral**.



Then the user can define the Geometry, the Integral type



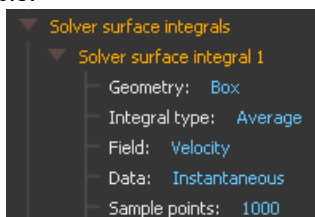
and Field to integrate



The other inputs required to define the visualization of the cutting plane are:

- ↳ **Data:** Chose Instantaneous, Averaged or Standard deviation.
- ↳ **Sample points:** Number of points on the surface to compute the integral.

Example:



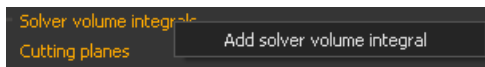
⚠ Please note: Solver surface integrals currently supports all geometries (fixed, enforced with and without [immersed-boundary](#) and rigid body dynamics) for **Enable boundary conditions: On** and **Off**. No [post-processing geometries](#) can be used.

Solver volume integrals

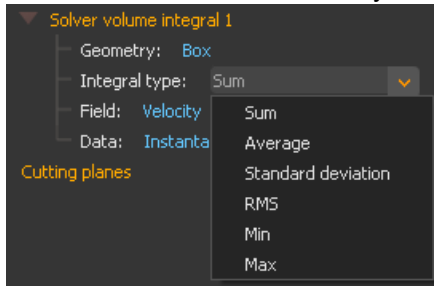
This feature allows to automatically compute and store volume integral values in the selected bodies during the computation at the [numerical data frequency](#). The integral type, data field and data type (instantaneous or averaged) must be specified. For more information, please check the [Volume integrals](#) chapter.

A single Solver volume integrals can be created by:

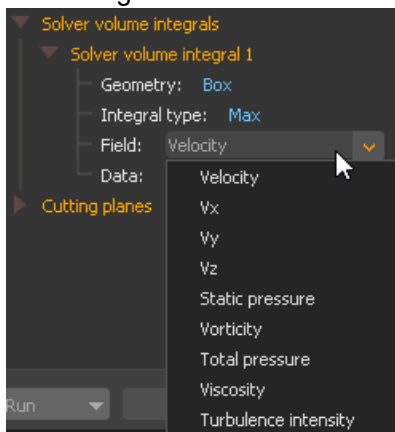
- **Right click** on **Project Tree > Simulation > Store Data > Solver volume integrals** and select **Add solver volume integral**.



Then the user can define the Geometry, the Integral type



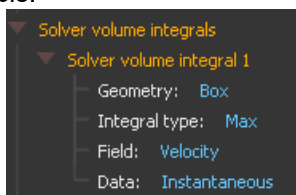
and Field to integrate



The other inputs required to define the visualization of the cutting plane are:

↳ **Data:** Chose Instantaneous, Averaged or Standard deviation.

Example:



⚠ Please note: Solver volume integrals currently supports all geometries (fixed, enforced with and without [immersed-boundary](#) and rigid body dynamics) for **Enable boundary conditions: On** and **Off**. No [post-processing geometries](#) can be used.

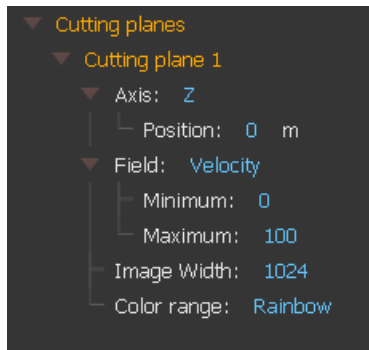
Cutting planes

This feature allows to automatically store images of cutting planes of the XFlow simulation and store them in the `cutting_planes` folder located in the simulation folder. The **Axis**, **Position** and **Field** options are defined as in the [Cutting Planes](#) of the pos-processing section.

⚠ Please note: Automatic storage of cutting planes only allows to defined the axis in the global axes directions (X,Y,Z) and to save only [3D fields](#).

The other inputs required to define the visualization of the cutting plane are:

- ↳ **Minimum:** Minimum value of the plot field
- ↳ **Maximum:** Maximum value of the plot field
- ↳ **Image width:** width of the image in pixels
- ↳ **Color range:** Color gradient to plot the cutting planes.



Simulation cutting planes

6 Computation

This chapter explains how to run a simulation, once it has been [set up](#).



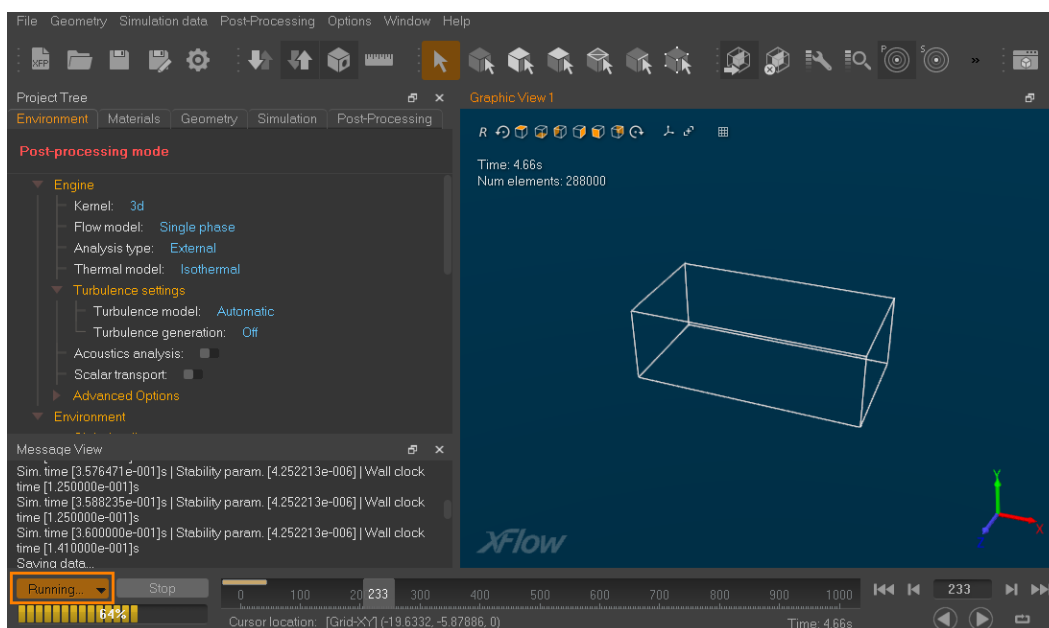
XFlow workflow: Project Tree categories in black. Computation is highlighted in orange.

In the following sections, it is described:

- How to [run a computation](#)
- How to [stop a computation](#)
- How to [monitor the progress](#) of the computation

6.1 Run computation

A computation can be launched by clicking the **Run** button at the bottom left corner of the screen, shown in the figure below.



Graphical User Interface: Run button highlighted by an orange frame

Run button displays a drop-down menu with the following options:

- [Start computation](#)
- [Start advanced computation](#) (only available in [Labs](#) mode)
- [Start FMU computation](#) (only available in [Labs](#) mode)
- [Resume computation](#)

6.1 Run computation

Start computation

This option starts the computation (initial time = 0) and launches the XFlow [Process Manager](#).

Numerical results are written in the data [folder](#) ([Project Tree](#) > [Simulation](#) > [Store data](#) > [Folder](#)). If the folder contains data of a previous calculation, the user will be asked whether he wants to overwrite them or rename the folder where the new data is going to be saved.

Start FMU computation

This option starts the computation (initial time = 0) and wait for the start of the coordinator software to couple with XFlow. See more in [FMI Standard](#).

Numerical results are written in the data [folder](#) ([Project Tree](#) > [Simulation](#) > [Store data](#) > [Folder](#)). If the folder contains data of a previous calculation, the user will be asked whether he wants to overwrite them or rename the folder where the new data is going to be saved.

Resume computation

Continues the simulation from the last saved result (initial time $\neq$ 0) launching the XFlow [Process Manager](#).

Simulation results are written in the data [folder](#) ([Project Tree](#) > [Simulation](#) > [Store data](#) > [Folder](#)), without overwriting previous simulation results.

To be able to use this feature, the user has to switch on the option:

[Project Tree](#) > [Simulation](#) > [Store data](#) > [Save resume file](#) ([+Info](#)).

The user will not be able to resume the computation without having enabled this option before launching the computation.

 **Please note:** The project has to be saved before being able to start or resume a computation.

6.1.1 Generate launch scripts

This option is only available in [Expert mode](#).

To generate the Windows (.bat) and the Linux (.sh) scripts to launch the simulations:

[Main menu](#) > [Simulation Data](#) > [Generate launch scripts](#)

For more information about the simulation execution by command lines, see [Advanced command lines](#).

 **Please note:** The XFlow installation path and the Simulation directory path have to be checked and changed before executing these scripts.

 **Tip:** The Launch scripts are particularly usefully to launch simulations with [Automatic initialization](#) in command lines.

6.1.2 Generate co-sim configuration file

This option is only available in [Expert mode](#).

To generate the co-simulation configuration file for [CSS co-simulation](#) setups:

[Main menu](#) > [Simulation Data](#) > [Generate co-sim configuration file](#)

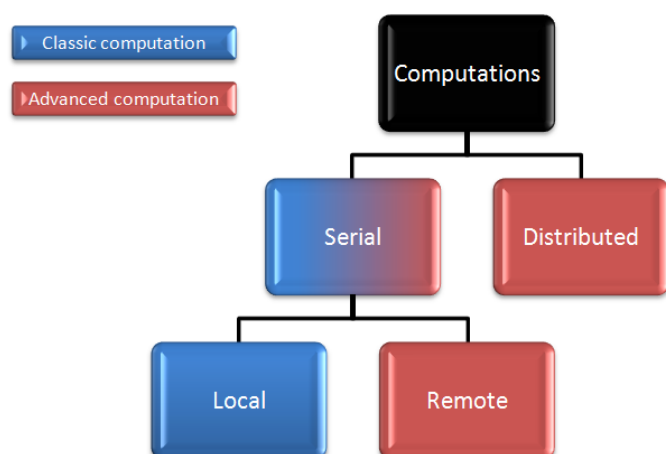
For more information about the co-simulation workflow see [CSS Co-simulation](#).

Tip: The co-sim launch script is automatically generated when running the XFlow simulation from the GUI.

Tip: The co-sim launch script should only be used for Simpack cosimulations. For Abaqus cosimulation, it is recommended to use the FSI_II_std_css.xml file distributed with the XFlow installation package.

6.1.3 Advanced computation

Main menu > Options > Preferences > Engine > Enable advanced computations
Run> Start Advanced Computation > Start advanced computation



Schematic of the parallel computing options

The advanced computation differs from the classic-serial and local computation in that it can run serial remote as well as distributed computations. These are especially useful if a network of computational workstations or even a High Performance Computing (HPC) cluster is available.

To run a simulation using the advanced computation features of XFlow:

1. Activate the option **Enable advanced computations** in **Main menu > Options > Preferences > Engine >** ,
2. Launch the simulation using the option **Start advanced computation** in **Run**, and select the Run type between:
 - a) Advanced serial, i.e. remote ([+Info](#))
 - b) Distributed ([+Info](#))

⚠ Please note: The advanced computation is a user-friendly interface, but any of these simulations can similarly be executed through command lines. See the [Advanced command lines](#) chapter.

6.1.3.1 Serial computation

⚠ Please note: Local serial computations are those executed through the simple [Run computation](#) button and are out of the scope of this section.

Run> Start Advanced Computation > Run type > Serial

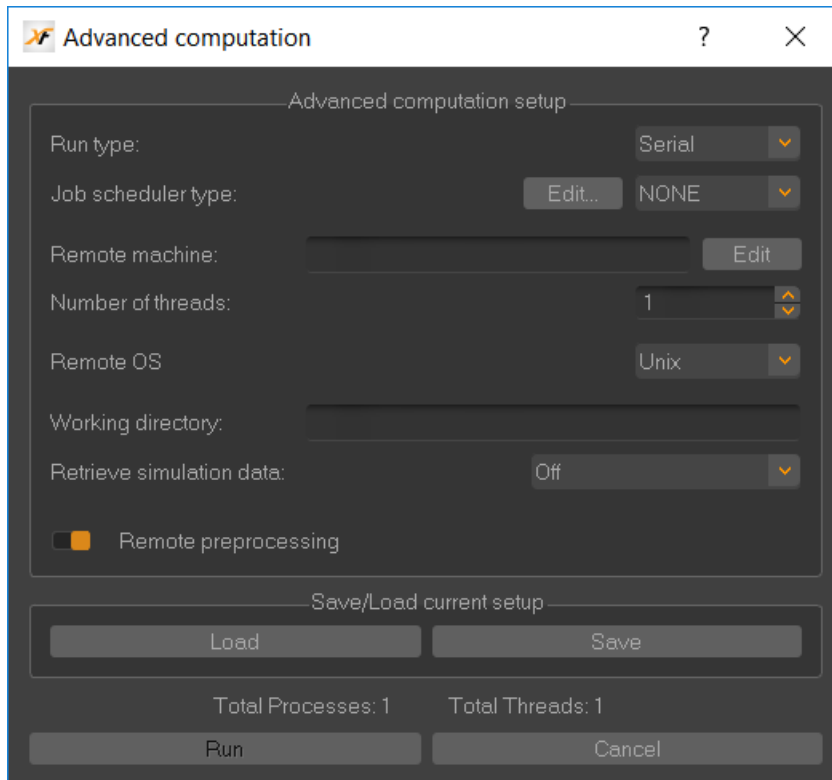
The serial computations allows the user to execute a computation on another machine remotely. This may be

6.1 Run computation

useful in different situations, e.g. if your local workstation has not enough computational resources but another machine on your network has.

Please note: The computations launched remotely are connected to the local process manager. This enables to monitor the simulation progress. If the simulation is stopped, XFlow creates an [exit-xflow](#) file on the remote working directory to cleanly exit the engine execution.

XFlow provides a wizard to setup advanced computations in serial. This is shown in the figure and its options are commented below:



Advanced Computation wizard: Serial

Job scheduler type

Job schedulers are used on HPC clusters to distribute and schedule the jobs on the available computational nodes; in case there is no computational resources available, the job scheduler automatically queues the jobs. XFlow is compatible with the following job schedulers:

- **None:** This is the most classic option and it should be used to run a simulation on a remote workstation where no job scheduler is installed and setup.
- **LSF:** Load Sharing Facility. This is a commercial job scheduler.
- **SLURM:** Simple Linux Utility for Resource Management. This is an open-source resource manager designed for Linux clusters.
- **PBS-Torque:** Portable Batch System. This is an open-source job scheduler.
- **Custom:** allows the user to execute a custom remote command.

Edit button: the command line as well as the execution script can be edited:

- The command line is the command executed on the execution machine.
- The script is a sequence of instructions for the job scheduler. It can include different variables with the syntax `%variable`:

`%engine`: returns the installation path and corresponding engine. Example: `/opt/DassaultSystemes/XFlow/engine-3d (Linux)`.

`%workingdirectory`: returns the [working directory](#) defined in the Advanced computation wizard.

`%runfile`: returns the name of the simulation file. Example: `projectname.xfp`.

`%args`: returns the [engine options](#) according to the project file.

`%threadsperjob`: returns the [number of threads](#) defined in the Advanced computation wizard.

`%numjobs`: returns the [number of processes](#) defined in the Advanced computation wizard. Only available for [distributed computations](#).

`%outputfile`: returns the name of the log file preceded by the symbol ">". Example: `> projectname.out`.

`%machines`: returns the list of the nodes participating in the computation. Only available if job [scheduler type = None](#). Example: `hostname1,hostname2,hostname3`.

`%pathengine`: returns the list of the installation path for each node of the table. Must be used only when at least two nodes have a different installation path (otherwise, nothing is returned). Only available if job [scheduler type = None](#). Example: `/path1,/path2,/path3`.

`%script`: returns the name of the script file defined in the script text area. It corresponds to `projectname.sh`.

`%jobname`: returns `XFlow-[projectName]`. Can be used as an identifier for job schedulers.

 **Please note:** The default command line usually does not need modifications when no job scheduler is used. You can recover the default command line clicking on the "Reset to default" button.

 **Tip:** Please reset the Advanced Computation execution line when installing a new XFlow version or when changing the MPI implementation. To proceed, please use the "Reset to default" button.

Remote machine

Host name or IP of the remote machine where the computation will be run.

 **Please note:** You have to click on the Edit button and set the installation folder as seen by the remote computing machine.

Number of threads

Number of threads used on the remote machine for the computation.

Remote OS

Operating System (OS) of the remote machine.

 **Please note:** For serial computation the remote machine OS must be Unix since Windows machines do not include SSH server.

Working directory

Directory where the computation will be saved on the remote machine.

Retrieve simulation data

- **Off:** The simulation will be executed remotely and simulation data will remain on the remote machine. XFlow will not copy the simulation data back to your local machine.
- **Locally mounted as:** If the working directory is mounted locally on your machine, you can specify the local directory mounted indicating the local path. The data will not be copied and therefore will be quicker to monitor the remote simulation computation.
- **Copy to simulation folder:** The simulation will be executed remotely and simulation data will be copied

6.1 Run computation

from the remote machine to your local machine in the local [simulation folder](#) defined in the Simulation tab.

Remote post-processing

Executes the domain generator on the remote machine. This is recommended especially if the local machine has low memory resources.

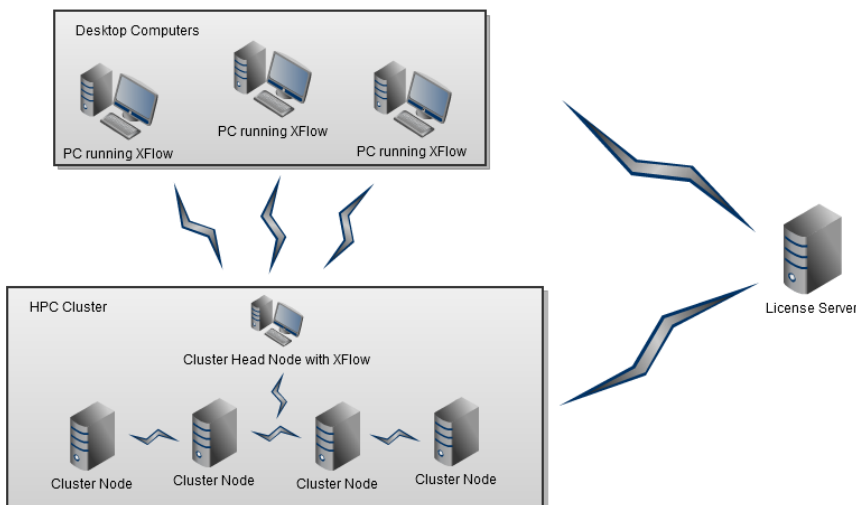
6.1.3.2 Distributed computation

Run> Start Advanced Computation > Run type > Distributed

The distributed computations are running on more than one machine simultaneously, using an MPI (Message Passing Interface) system. MPI splits the domain in several partitions, each of which will be computed on a different machine. This option is usually recommended for HPC cluster, as illustrated in the figure below, because the network must be optimum to efficiently exchange information among the computing nodes. However, it can also be used on a single machine where different MPI instances of the engine can be run simultaneously.

⚠ Please note: Distributed computation is currently available for any case-setup except those involving:
(1) **Refinement algorithm = Adaptive refinement**

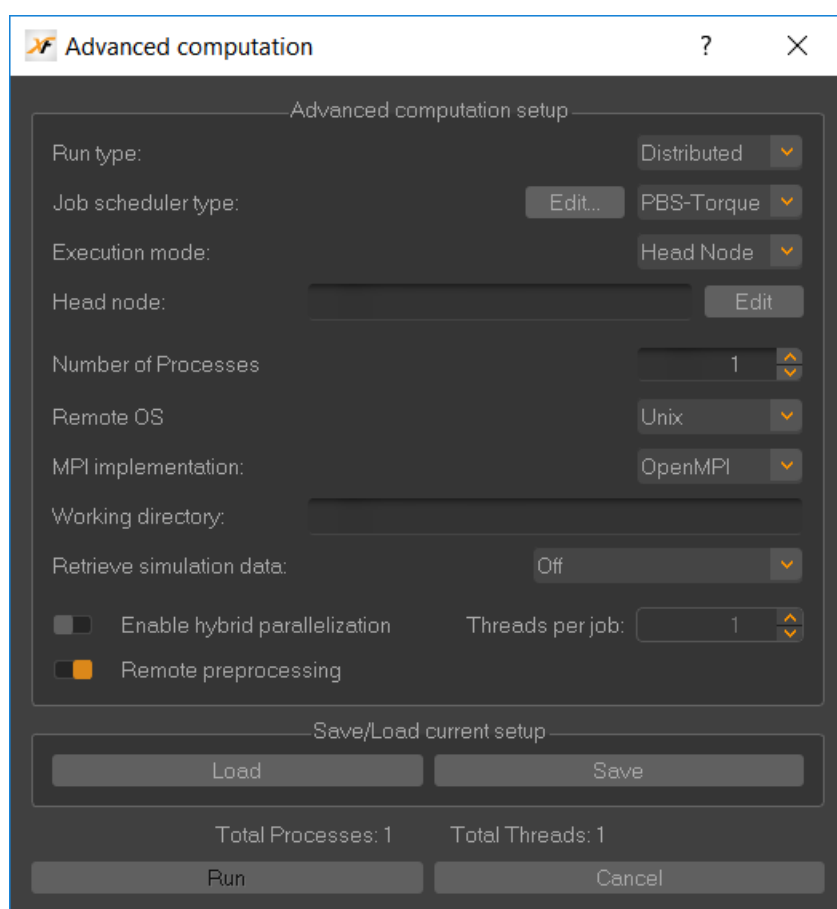
⚠ Please note: The computations are connected to the local process manager. This enables to monitor the simulation progress even if executed remotely from the cluster head node. If the simulation is stopped, XFlow creates an [exit-xflow](#) file on the remote working directory to cleanly exit the engine execution.



Schematic of distributed computation on a HPC cluster

⚠ Please note: In order to run distributed simulations in Windows OS the user must have a valid Intel MPI installation or, at least, the Intel® MPI Library Runtime Environment for Windows.

XFlow provides a wizard to setup advanced computations in distributed mode. This is shown in the figure and its options are commented below:



Advanced computation wizard: Distributed

Job scheduler type

Job schedulers are used on HPC clusters to distribute and schedule the jobs on the available computational nodes; in case there is no computational resources available, the job scheduler automatically queues the jobs. XFlow is compatible with the following job schedulers:

- **None:** This is the most classic option and it should be used to run a simulation on a remote workstation where no job scheduler is installed and setup.
- **LSF:** Load Sharing Facility. This is a commercial job scheduler.
- **SLURM:** Simple Linux Utility for Resource Management. This is an open-source resource manager designed for Linux clusters.
- **PBS-Torque:** Portable Batch System. This is an open-source job scheduler.
- **Custom:** allows the user to execute a custom remote command.

Edit button: the command line as well as the execution script can be edited:

- The command line is the command executed on the execution machine.
- The script is a sequence of instructions for the job scheduler. It can include different variables with the syntax `%variable:`

`%engine:` returns the installation path and corresponding engine. Example: `/opt/DassaultSystemes/XFlow/engine-3d (Linux)`.

`%workingdirectory:` returns the [working directory](#) defined in the Advanced computation wizard.

`%runfile:` returns the name of the simulation file. Example: `projectname.xfp`.

`%args:` returns the [engine options](#) according to the project file.

6.1 Run computation

`%threadsperjob`: returns the [number of threads](#) defined in the Advanced computation wizard.
`%numjobs`: returns the [number of processes](#) defined in the Advanced computation wizard. Only available for [distributed computations](#).
`%outputfile`: returns the name of the log file preceded by the symbol ">". Example: > projectname.out.
`%machines`: returns the list of the nodes participating in the computation. Only available if job [scheduler type = None](#). Example: hostname1,hostname2,hostname3.
`%pathengine`: returns the list of the installation path for each node of the table. Must be used only when at least two nodes have a different installation path (otherwise, nothing is returned). Only available if job [scheduler type = None](#). Example: /path1,/path2,/path3.
`%script`: returns the name of the script file defined in the script text area. It corresponds to projectname.sh.
`%jobname`: returns XFlow-[projectName]. Can be used as an identifier for job schedulers.

⚠ Please note: The default command line usually does not need modifications when no job scheduler is used. You can recover the default command line clicking on the "Reset to default" button.

💡 Tip: Please reset the Advanced Computation execution line when installing a new XFlow version or when changing the MPI implementation. To proceed, please use the "Reset to default" button.

Execution mode

The execution mode allows the user to choose either if the computation will be managed through a head node or not:

- **Local:** This mode does not manage the simulation through a head node. The jobs will be sent from the local host and distributed to the nodes.
- **Head node:** A head node will automatically appear at first position of the nodes table. The head node is the node responsible to manage the simulation, send the jobs to the computational nodes and retrieve the simulation results. The edit button allows the user to set the installation path of XFlow on the head node.

⚠ Please note: One must select the "Head node" execution mode when using HPC clusters unless XFlow is executed directly on a cluster node.

Nodes table [Job scheduler type = NONE]

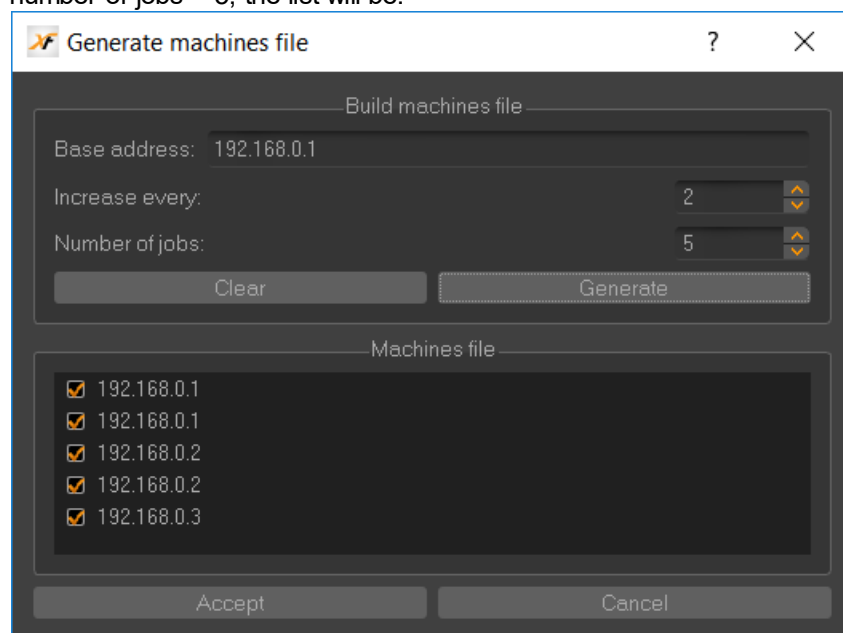
If no job scheduler is used, a table appears in order to define the head and computational nodes: a list can be added manually with the **Add/Remove** buttons, it can be imported from a text file with the **Browse** button, or it can also be generated by means of the **Autogen** button. The table can be saved with the **Save** button.

Node Type	Host Name	Processes	
Head Node	127.0.0.1	-	Add
Computation Node	127.0.0.1	1	Remove
Computation Node	127.0.0.1	1	Edit
			Browse
			Save
			Autogen

Example of distributed computation with no job scheduler, with Execution mode=Head node, on two machine using 4 and 8 processes respectively

The number of nodes stands for the number of physical machines that will be running the computation in the HPC cluster. Those nodes are automatically selected and managed by the job scheduler (if used). The number of processes is the number of MPI engines run on each node. The total number of cores used in the simulation will therefore be: $\text{number of nodes} \times \text{number of processes}$.

The Autogen button is a tool that automatically generates a list of nodes. Once the **Base address** is entered, the **Number of jobs** defines the number of total address generated, repeating each address by the number set in **Increase every**. Example: if base address is 192.168.0.1, increase every = 2 and number of jobs = 5, the list will be:



Nodes [Job scheduler type ? NONE]

If a job scheduler is used, it will organize the job submission on the HPC environment. Therefore, there is no need to specify the nodes hostname, and only the number of processes is required as a user input. The job scheduler will automatically select the nodes available to run the simulation, or queue it if nodes are busy.

Number of Processes

This is the number of processes running on each node.

Remote OS

Operating System (OS) used on the computation environment (head node, nodes, remote machines, etc.) and is independent of the local OS.

⚠ Please note: Remote OS must be Unix when a job scheduler is used, and when execution mode is head node.

MPI implementation

Different MPI implementation are available in XFlow. One must select an implementation installed and configured on the computation environment to be able to run a distributed computation. Compatible MPI implementations are:

- Intel MPI 5
- OpenMPI 1.4

6.1 Run computation

- OpenMPI 1.6
- OpenMPI 1.10

 **Please note:** When using the Intel MPI 5.1.2 implementation the user should check that the SMPD service is running on each calculation machine by executing the command: `C:\Program Files (x86)\IntelSWTools\mpi\5.1.2.146\intel64\bin\smpd.exe -status`.

Working directory

Directory where the computation will be saved on the head node or locally depending on the execution mode.

Retrieve simulation data

- **Off:** The simulation will be executed remotely and simulation data will remain on the remote machine. XFlow will not copy the simulation data back to your local machine.
- **Locally mounted as:** If the working directory is mounted locally on your machine, you can specify the local directory mounted indicating the local path. The data will not be copied and therefore will be quicker to monitor the remote simulation computation.
- **Copy to simulation folder:** The simulation will be executed remotely and simulation data will be copied from the remote machine to your local machine in the local [simulation folder](#) defined in the Simulation tab.

Enable hybrid parallelization

This enables multi-thread parallelization for the different processes of MPI running.

 **Tip:** For the best scalability performance it is recommended to use one partition ([process](#)) per node and enable the hybrid parallelization in order for each process to use all the node threads available on the node.

Example: a simulation is run on 3 nodes of 12 cores, then use Number of processes = 3 and enable hybrid parallelization = 12 threads. The simulation will therefore run with $12 \times 3 = 36$ cores in total.

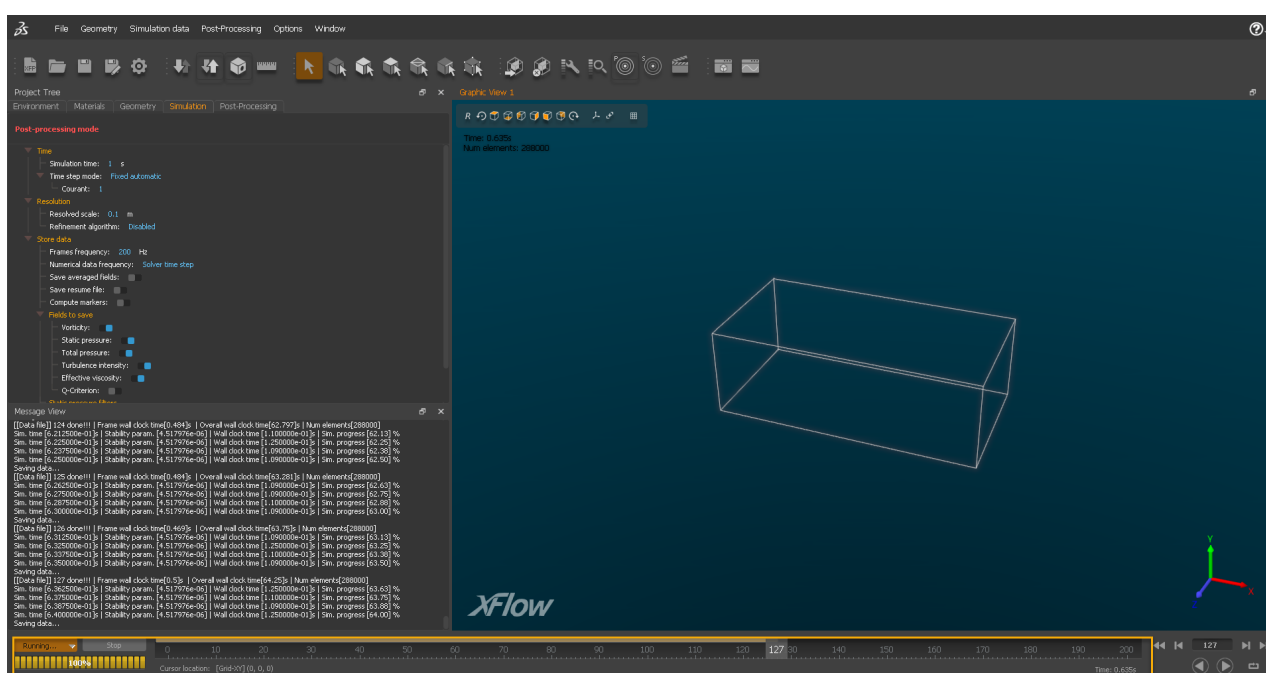
Remote pre-processing

Executes the domain generator on the remote machine. This is recommended especially if the local machine has low memory resources.

6.2 Computation progress

The simulation progress is shown in the GUI as follows:

- The progress bar moves from 0 and 100% indicating the progress to complete the next saved frame of data;
- The time bar of the time controls indicates the global progress of the simulation, i.e. the frames that have been calculated.



GUI: Monitoring the simulation progress, highlighted in orange

If the user closes the GUI but decides to keep the simulation running, the global progress of the simulation can be followed in the [Process Manager](#) window.

The **Message View** reports additional information at every solver step, such as the stability parameter, warnings or errors. For more information please see [Message View](#).



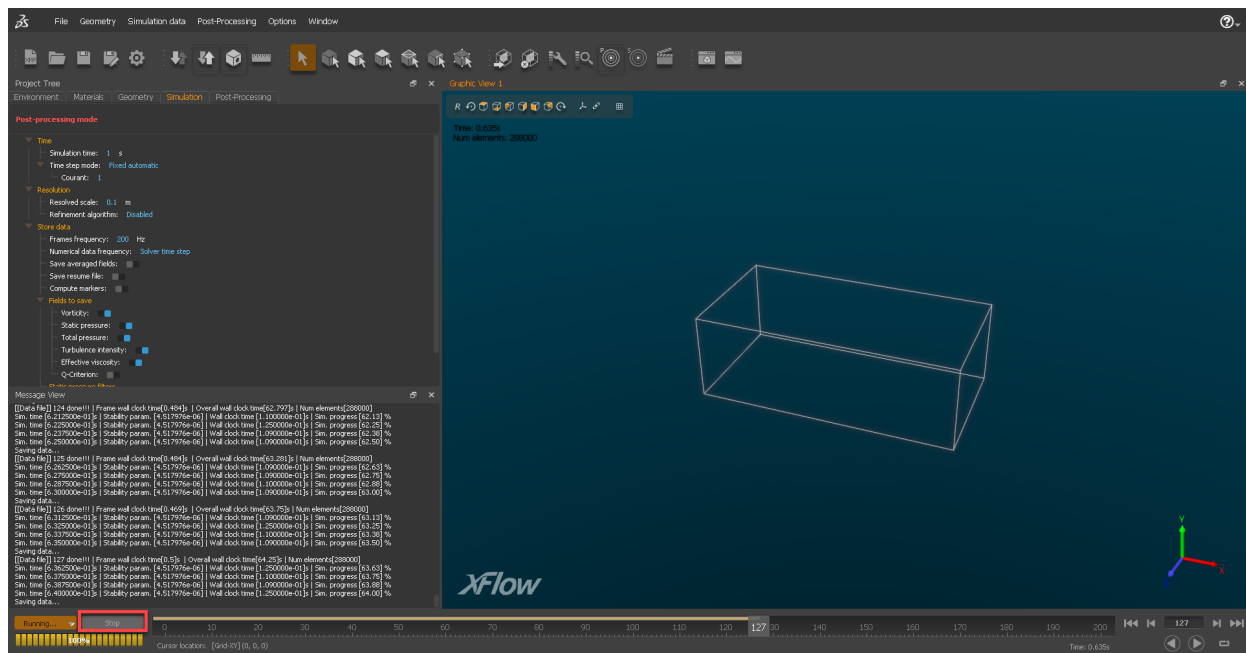
Tip: If the computation time between two frames is long, it is possible to save an intermediate frame by creating an empty text file named "**savedata-xflow**" or "**savedata-xflow.txt**" in the [Simulation folder](#). The engine will automatically detect this file and will proceed to save a frame data and continue the simulation. Once the simulation finally arrives at the actual next frame, it will automatically replace the intermediate saved frame by the actual one.

6.3 Stop computation

There are three ways to stop a computation:

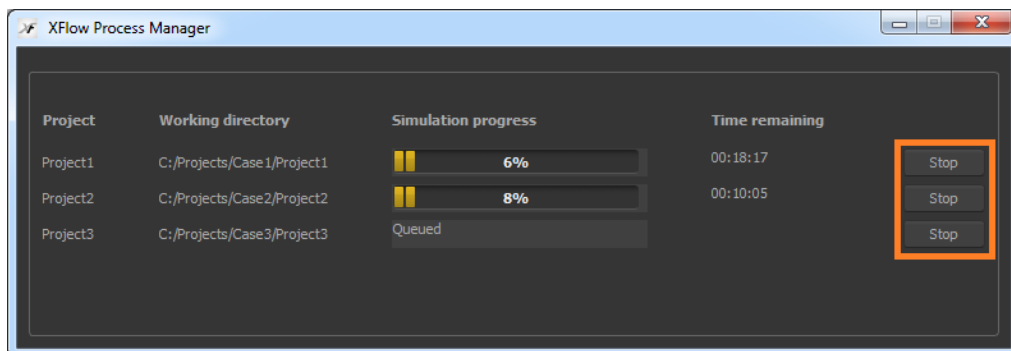
1. Click the **Stop** button at the bottom left corner of the [GUI](#)

6.3 Stop computation



GUI, Stop button highlighted by an orange frame

2. Click the **Stop** button shown in the [Process Manager](#) window

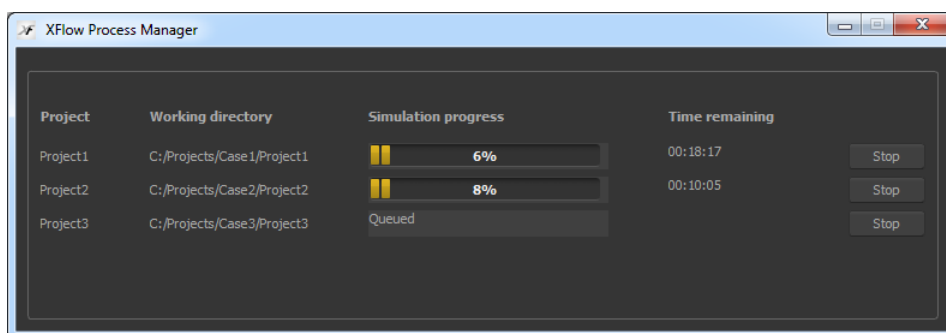


Process Manager stop-button highlighted by an orange frame.

3. Create an empty text file named "**exit-xflow**" or "**exit-xflow.txt**" in the [Simulation folder](#). The engine will automatically detect this file and will proceed to save the frame data and to exit.

6.4 Process manager

XFlow Process Manager, , is a daemon process running in background that manages all simulations running in a machine. The Process Manager window (shown in the image below) pops up everytime a simulation is started.



XFlow Process Manager lists progress of the simulations (if running) and those that are queued.

The user can minimize/maximize and even close/open this window. Once it is closed it can be reopened by **clicking** on the icon appearing in the system tray (Windows) or in the main panel (Linux), .



Process Manager icon on system tray

Through the *Process Manager* the user can.

- [Monitor](#) the progress of the simulation on progress (Simulation progress in the figure above)
- Manage jobs [queues](#) (List of projects in the figure above)
- Stop the simulations running, by clicking the corresponding Stop button (see figure above).

Monitoring the computation progress

The Graphical User Interface (GUI) can be closed while the simulation is kept running; if the user does so, XFlow will inform:

"There is a simulation running. Would you like to keep the simulation running?".

→ **Yes**: XFlow will close the GUI and keep the process manager opened, allowing to monitor the simulation without the GUI.

→ **No**: XFlow will close both GUI and process manager, stopping the simulation.

→ **Cancel**: XFlow will keep both interface and process manager opened.

Queue jobs

XFlow allows the user to run several jobs simultaneously as long as enough cores are available (see figure above: Test1 and Test2 running simultaneously). If a job submitted to the *Process Manager* and the number of available cores is not sufficient, XFlow automatically queues that process (see figure above: Test3 queued).

Once a computation finishes leaving enough cores to run a queued project, this will started automatically.

6.5 File-based computation control

It is possible to control some parameters of the computation by creating files in the [Simulation folder](#). Specifically:

- **"savedata-xflow"** or **"savedata-xflow.txt"**. The engine will automatically detect this file and will proceed to save a frame data and continue the simulation. Once the simulation reaches the actual next

frame, it will automatically replace the intermediate saved frame by the actual one.

- **"exit-xflow"** or **"exit-xflow.txt"**. The engine will automatically detect this file and will proceed to save a last frame data after completion of the current time-step (even if the simulation time does not match the user setup value) and stop the computation.
- **"saveresume-xflow"** or **"saveresume-xflow.txt"**. The engine will automatically detect this file and will create the resume files once the simulation reaches the following frame. This behaviour is not equivalent to enabling the save resume file option in the GUI since it will create a resume point only for that specific frame (the standard save resume file option overwrites the resume point at each frame). The computation can, once the resume file saving operation has completed, be resumed from that specific frame only via command line. No changes will be reflected in the GUI setup. A log entry will be created: [INFO] Saveresume-xflow file found: saving a resume point.
- **"simulationtime-xflow"** or **"simulationtime-xflow.txt"**. The engine will automatically detect this file and will change the simulation time initially set by the user with the number specified in the file. Various formats can be used to specify the time in seconds (e.g. 10.0 s, 10, 1e10 s). The change will only be effective if the specified time is greater than the current computation time. This parameter change will be detected by the XFlow GUI and the following message will be shown to the user: *"Simulation time changed via the simulationtime-xflow.txt file. The project will be modified accordingly."* A log entry will be created: [INFO] Simulationtime-xflow file found: changing simulation time to 10.0 s.

7 Post-Processing

This chapter explains how to post-process the simulation data, covering the following topics:

- [Load and unload simulation data](#)
- [Post-processing tree](#): XFlow has a wide range of post-processing features that can be controlled from the Post-Processing branch of the *Project Tree*.
- [Visualization fields](#), these are the XFlow output variables that can be post-processed.
- [Import/Export post-processing setup](#) to repeat the same post-processing on several simulations.
- [Create animation](#) from results
- Data visualization modes: [Graphich View](#) & [Function Viewer](#)
- [Export](#) XFlow results into another file extension.

7.1 Load/unload simulation data

If the Graphical User Interface window (from where the simulation has been launched) is kept opened during the simulation, the numerical data are automatically loaded as they are generated. Otherwise, the user has to load the numerical data to be post-processed by:

Main menu > Simulation data > Load simulation data, or 

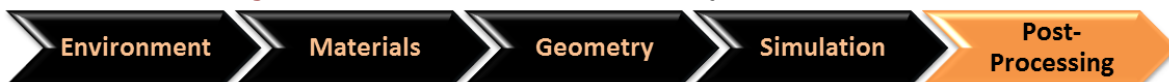
The numerical data can also be unloaded by selecting:

Main menu > Simulation data > Unload simulation data, or 

 **Please note:** Simulation data cannot be loaded in [Editing mode](#) since the project data might be different from your simulation data. In order to be able to load the simulation data again, you may recover the original project clicking on Editing mode in the top of the project tree.

7.2 Post-Processing tree

The **Post-Processing** tree is shown in the last tab of the *Project Tree*.

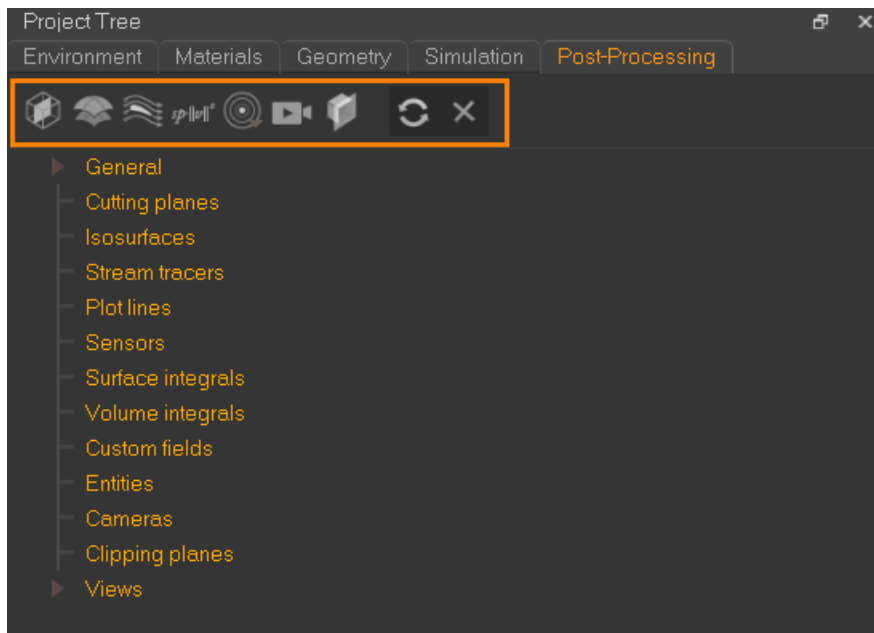


XFlow workflow: Project Tree categories, the one corresponding to the Post-Processing is highlighted in orange.

Post-Processing toolbar

Unlike other tabs of the Project Tree, this one has its own toolbar as it is shown in the figure below:

7.2 Post-Processing tree



Post-Processing toolbar in an orange frame.

Post-processing toolbar icons are:

	Create cutting plane	Main menu > Post-Processing > Create cutting plane
	Create isosurface	Main menu > Post-Processing > Create isosurface
	Create stream tracer	Main menu > Post-Processing > Create stream tracer
	Create sensor	Main menu > Post-Processing > Create sensor
	Create plot line	Main menu > Post-Processing > Create plot line
	Create surface integrals	Main menu > Post-Processing > Create surface integral
	Create custom field	Main menu > Post-Processing > Create custom field
	Create camera	Main menu > Post-Processing > Create camera
	Refresh selected	Main menu > Post-Processing > Refresh selected
	Delete selected	Main menu > Post-Processing > Delete selected

Create a post-processing object

The post-processing objects have to be created, except for:

- Post-Processing > General
- Post-Processing > Views

To create any other type of post-processing object, the user have three possibilities:

1. **Right-click** on it and select the *Add* button
2. Click on the corresponding icon in the [post-processing toolbar](#), or go to **Main menu > Post-Processing** and choose the object to be created

Each object will appear in the **Post-Processing** tab of the *Project Tree*, in the corresponding section to

its type.

Refresh a post-processing object

Refreshing a post-processing object will regenerate it from the current parameters. The objects which can be refreshed are:

- Isosurfaces
- Stream tracers
- Data plot lines
- Data sensors
- Data plot surfaces

To select an object proceed as follows:

1. Select the object
2. Click on the refresh button of the [post-processing toolbar](#), , or go to **Main menu > Post-Processing > Refresh selected**

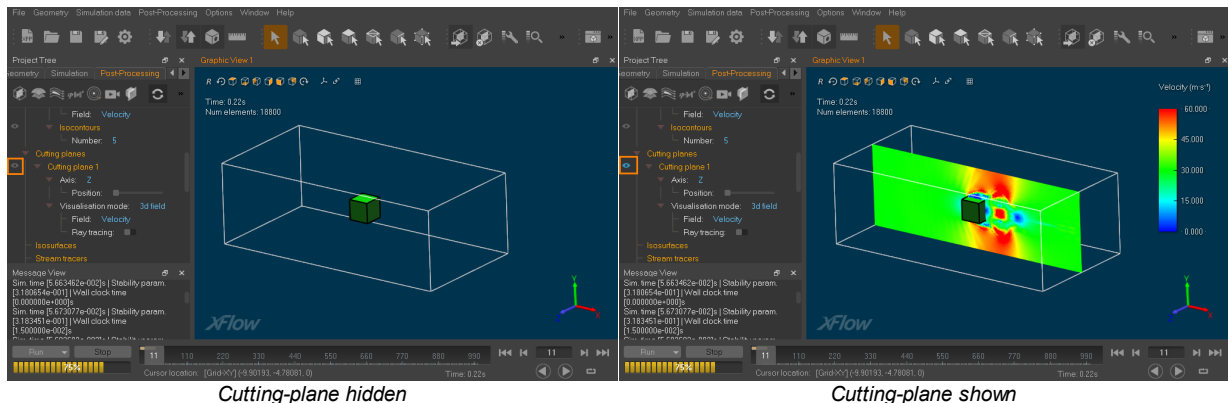
Delete a post-processing object

Any post-processing object can be deleted by:

1. Selecting the object in the **Post-Processing** tree
2. Clicking on the delete button of the [post-processing toolbar](#), , or going to **Main menu > Post-Processing > Delete selected**

Show/hide post-processing object

All the post-processing objects can be shown/hidden in the *Graphic View* window with the help of the eye visibility check . The visualization will affect the active *Graphic view* only. A Graphic view becomes active when you click on it.



Cutting-plane hidden

Cutting-plane shown

7.2 Post-Processing tree

The object-structure of the **Post-Processing** tree is as follows:

- General ([+Info](#))
- Cutting planes ([+Info](#))
- Isosurfaces ([+Info](#))
- Stream tracers ([+Info](#))
- Plot lines ([+Info](#))
- Sensors ([+Info](#))
- Surface integrals ([+Info](#))
- Volume integrals ([+Info](#))
- Custom fields ([+Info](#))
- Cameras ([+Info](#))
- Views ([+Info](#))

7.2.1 General

Project Tree > Post-Processing > General

The **General** section of the **Post-Processing** tree gather global parameters of visualization as well as 3D visualization features which are global to the domain.

The tree structure of this section is:

General Data (+Info)	Instantaneous Averaged Standard deviation RMS
Interpolation mode (+Info)	Off Convolution MLS
Show (+Info)	Domain structure Volumetric field Surface info Isocontours

7.2.1.1 Data

Project Tree > Post-Processing > General > Data

The option **Data** allows the user to select the type of data to be post-processed among:

- **Instantaneous**
- **Averaged**
- **Standard deviation**
- **RMS**

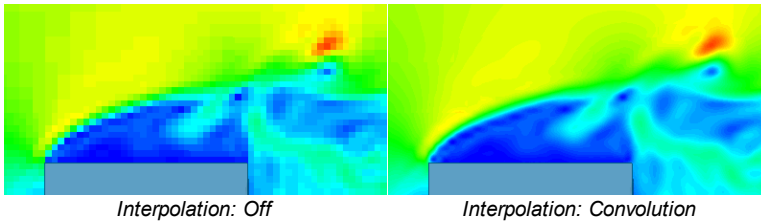
Please note that average, standard deviation, and RMS data are only available if the user has chosen to save the average and standard deviation and RMS values in the simulation setup ([+Info](#) in **Project Tree** > **Simulation** > **Store data** > **Save averaged fields**).

7.2.1.2 Interpolation

Project Tree > **Post-Processing** > **General** > **Interpolation**

The interpolation options are:

- **Off:** If interpolation is not enabled, voxels centered at the lattice points are uniformly colored according to the value at this lattice node of the variable being visualized.
- **Convolution:** Voxels centered at the lattice points are non-uniformly colored according to the values obtained from a continuum (third order accurate) tricubic interpolation of the neighbouring lattice nodes.
- **MLS:** Only available in [Labs mode](#).



7.2.1.3 Show

The data structure of the **Show** option is:

Show	Domain structure (+Info)	
	Volumetric field (+Info)	visualization field Transference law
	Markers ⁽¹⁾ (+Info)	Representation From color by field
	Surface info (+Info)	3d field LIC Cp Cf Y+ P+ Heat flux Heat transfer coefficient Fx Fy Fz
	Isocontours (+Info)	Number

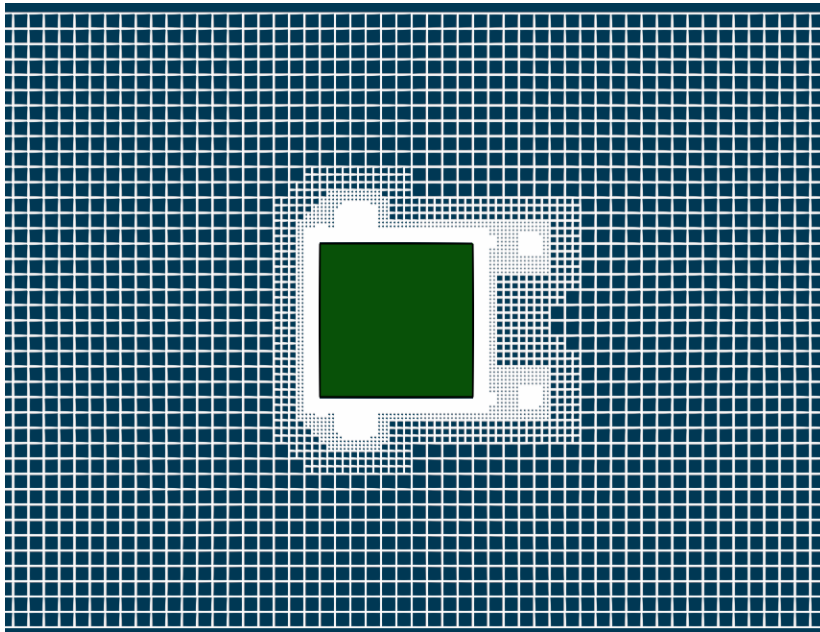
⁽¹⁾ Only if Flow model = Free surface and Compute markers = on.

Domain structure

Project Tree > Post-Processing > General > Show > Domain structure

If enabled with the view icon , it displays the borders between the different lattice levels in three dimensions. Each lattice level is shown as an isosurface with transparency and with a different color. The different visualization options are:

- ↳ **Min level:** minimum lattice level number to be plot. Zero corresponds to the coarser lattice level, defined by the project [resolved scale](#).
- ↳ **Max level:** maximum lattice level number to be plot.
- ↳ **Cell size:** displays the voxelization of the lattice elements on the border surfaces.



Show domain structure

Tip: To navigate through the different refinement levels: Select the *Graphic View* window and press **+** to increase the level or **-** to decrease the level

Please note: By default the domain structure is visualized on the entire 3D domain and it can require considerable memory usage. It is advisable to set a cutting plane visibility on before activating the Domain structure. This will limit the domain structure to be visualized only in the cutting plane region and, hence, ease the memory requirements.

Volumetric field

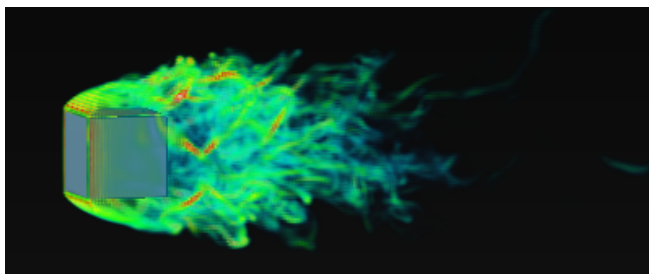
Project Tree > Post-Processing > General > Show > Volumetric field

Volumetric field shows (in the entire domain) the chosen visualization field with an opacity given by a specified transference law:

↳ **visualization field:** see [visualization field](#)

↳ **Transference law:** This law is a function of a (alpha value), where $a=0$ corresponds to the legend minimum and $a=1$ to the maximum. Examples:

- Transference law = $a*a$ assigns more opacity to the largest values of the visualization field
- Transference law = $1-a$ assigns more opacity to the smallest values of the visualization field

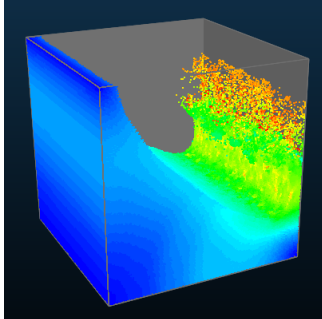


Show - Volumetric field of vorticity

Tip: The volumetric field using [interpolated](#) data is extremely slow. It is recommended to have the interpolation switched off.

Markers

Project Tree > Post-Processing > General > Show > Markers



Markers shown in a free surface internal flow

This option is only available for [Free surface](#) simulations and if the **Compute markers** option is enabled prior to launch the simulation ([+Info Project Tree > Simulation > Store data > Compute Markers](#)). If so, markers represent mass-less particles advected by the flow field and represent the fluid in the domain.

↳ **Representation:** Select the shape of the marker: **Points, Arrows, Spheres, Sharks, Snowflakes.**

↳ **From:** Select where to show the markers:

- **Liquid:** Markers are only shown for the liquid phase [*available if [Multiphase particle-based tracking](#) model enabled*]
- **Surface:** Markers are only shown at the free surface
- **All:** Markers are shown within the whole domain.

↳ **color by field:** Select the field according to which the markers are colored (see [visualization fields](#)).

⚠ Please note: The markers require to be enabled before running the simulation to be available in the post-processing. Computation is quicker if they are disabled.

Surface info

Project Tree > Post-Processing > General > Show > Surface info

This option allows the user to observe the projection of fields over geometries. One can choose to project one field from the following list:

- **3d field:** It displays the field value of the fluid touching the surface (first element next to the surface), not the value of the solid surfaces.
 - ↳ **Field:** The user has to select the field to be displayed on the surface.
- **LIC:** Line Integral Convolution of the velocity field. It provides a representation of the flow analogous to the resulting pattern of a tract of wind-blown sand, and involves selectively blurring a noise image as a function of the velocity vector field.
 - ↳ **Noise size:** To set up the initial Perlin noise pixel size.
 - ↳ **Motion blur amount:** To set up the blurring level (length).
- **Cp:** Pressure coefficient, defined as

$$C_p = \frac{2 P_{static}}{\rho V_{ref}^2}$$

where P_{static} is the local static pressure, ρ the reference density and V_{ref} the specified reference velocity.

- **Cf:** Skin friction coefficient, defined as

$$C_f = \frac{2 \tau_w}{\rho V_{ref}^2}, \quad \tau_w = \mu \left(\frac{\partial v}{\partial y} \right)_{y=0}$$

where τ_w is the wall shear stress.

- **Y⁺:** Dimensionless distance from the wall, defined as

$$y^+ = \frac{\rho v_\tau y}{\mu}, \quad v_\tau = \sqrt{\frac{\tau_w}{\rho}}$$

where v_τ is the friction velocity at the wall and y is the distance to the wall.

- **P⁺:** Dimensionless pressure at the wall (pressure-gradient parameter), defined as

$$p^+ = \frac{\nu}{\rho v_\tau^3} \left(\frac{\partial P}{\partial x} \right)$$

where ν is the kinematic viscosity and P is the local static pressure.

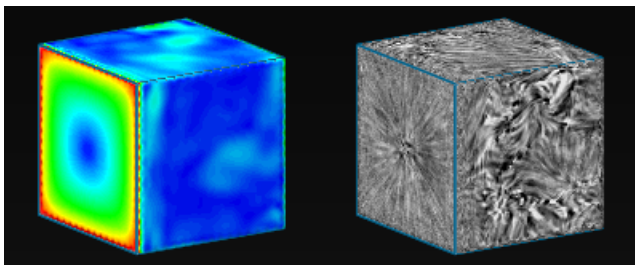
- **Heat flux:** Heat flux at the surface. This option is only enabled if *Thermal model* is non isothermal.
- **Heat transfer coefficient:** The heat transfer coefficient (HTC) is a constant of proportionality between the heat flux and the temperature gradient at wall boundaries. It is used to compute the heat transfer such as the convection between fluid and solid. The heat transfer coefficient is available for non-isothermal and isothermal simulations:

➤ The isothermal definition is estimated based on the skin friction, according to the [HTC Estimation](#) mode selected.

➤ The non-isothermal (segregated energy, coupled energy, supersonic) is defined with the classic definition:

$$h = \frac{q}{\Delta T}$$

For non-isothermal simulations, the HTC is only defined on wall boundaries with a temperature [thermal boundary condition](#), and free convection in case of [solid conduction](#). The HTC is equal to zero on adiabatic boundaries and heat flux boundaries.



Surface info: 3d field (left) and LIC (right)

- **Fx, Fy and Fz:** Force contribution per unit of surface in N/m². This can be used in order to see

7.2 Post-Processing tree

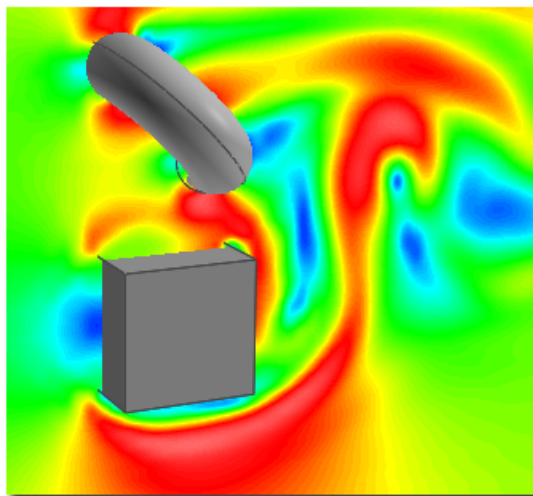
areas of drag and lift contributions in aerodynamics.

Isocontours

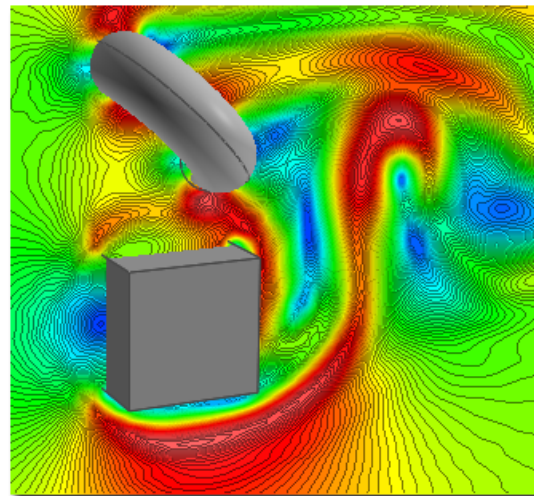
Project Tree > Post-Processing > General > Show > Isocontours

Isocontours plot lines that joint the points in the domain with the same value (magnitude). The number of contours is set by the variable **Number**.

Following images show the graphic effect of plotting isocontours:



3d field



3d field and isocontours

7.2.2 Cutting planes

Project Tree > Post-Processing > Cutting planes



Project Tree > Post-Processing > General > Cutting planes

Cutting planes are useful to visualize the numerical data on a plane of interest; the dimensions of which are limited by the domain size. To create a cutting plane do **right click** on the **Cutting planes** branch of the

Project Tree and choose **Add Cutting plane**, or press button  on the [Post-Processing toolbar](#).

The data structure of a **Cutting plane** object is:

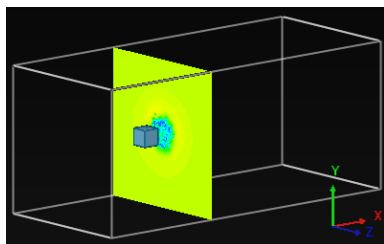
Cutting plane	Axis (+Info)	X Y Z Arbitrary
	Visualization mode (+Info)	3d fields Vectors Surface fields distribution Domain structure Markers

Axis

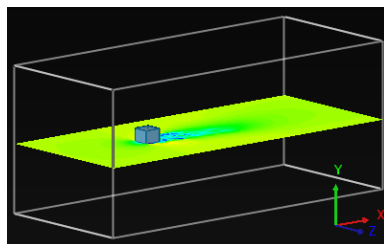
The axis defines the normal of the cutting plane. It can be set to:

- **X:** Global X axis
- **Y:** Global Y axis
- **Z:** Global Z axis
- **Arbitrary axis:** To set a user defined plane defined by:
 - ↳ **Origin:** Coordinates of the origin of the normal.
 - ↳ **Normal:** Components of the vector normal to the plane.

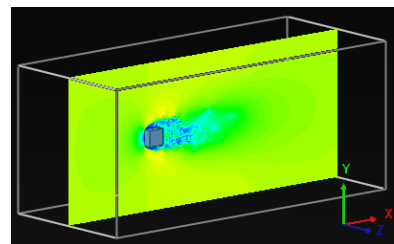
↳ **Position** represents the location of the cutting plane in the direction of the axis, according to the domain size (between 0 and 1).



X - Cutting Plane



Y - Cutting Plane



Z - Cutting Plane

Tip: Cutting planes can be translated using the translation Gizmos. To do so, select the cutting plane, , and do the following:

- **drag** the Gizmo axis (corresponding with the cutting plane axis), or
- **click** on the Gizmo axis and set the translation distance (in meter) in the translation dialogue-box, e.g X = 1m, Y = 0m, Z=0m) .

Visualization mode

The visualization mode of the cutting plane can be set up to:

- [3d fields](#)
- [Vectors](#)
- [Surface fields distribution](#)
- [Domain structure](#)

The inputs required to define the visualization of the cutting plane are:

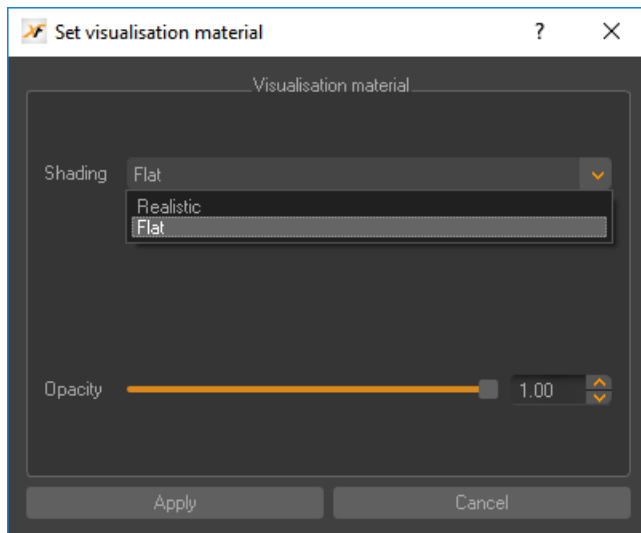
- ↳ **Field:** Field to plot on the cutting plane
- ↳ **Ray-tracing:** Ray-tracing is a technology to display the field values on the plane which pixelizes

7.2 Post-Processing tree

the cutting plane progressively through rays. This requires to recompute the cutting plane display every time the camera position is changed and may be slower. It is recommended to leave this option disabled unless you encounter specific issue to plot the cutting plane.

Visualization material

It is possible to set a material setting to the cutting plane by selecting the Object Filter and **right click** on the cutting plane.

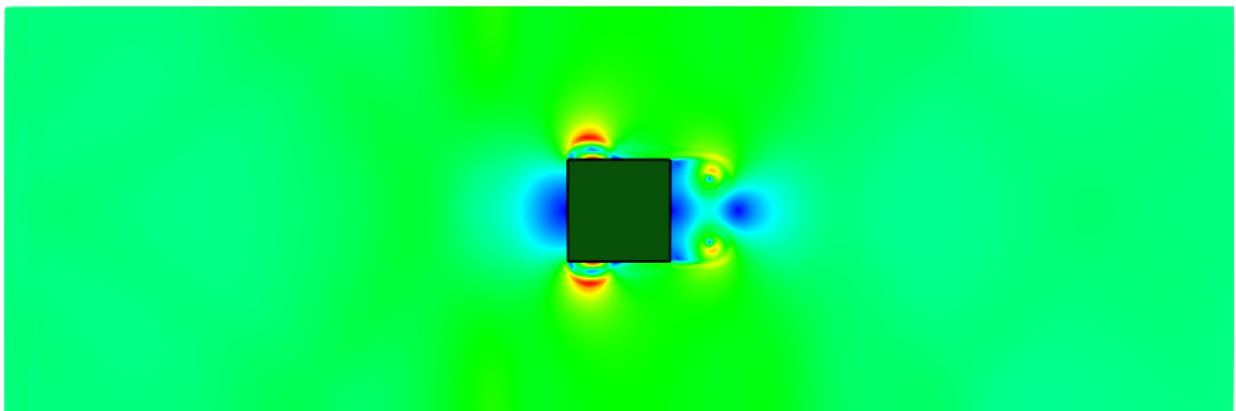


Set material in cutting plane.

The default choice is **Flat**, which allows the user to control only the **Opacity** level (transparency), while the **Realistic** option gives the user access to the same material presets used for geometries ([+info](#)).

7.2.2.1 3d field

This visualization mode allows the user to select a **Field** and to visualize its contours on the selected cutting plane, as shown in the figure below.



Example of 3d field in a cutting plane Z with interpolation enabled

The list of fields that can be visualized are listed in [visualization fields](#). The **3d field** visualization mode is

affected by some of the options available in the [General](#), namely: **Data**, **Interpolation mode**, **Domain structure** and **Isocontours**.

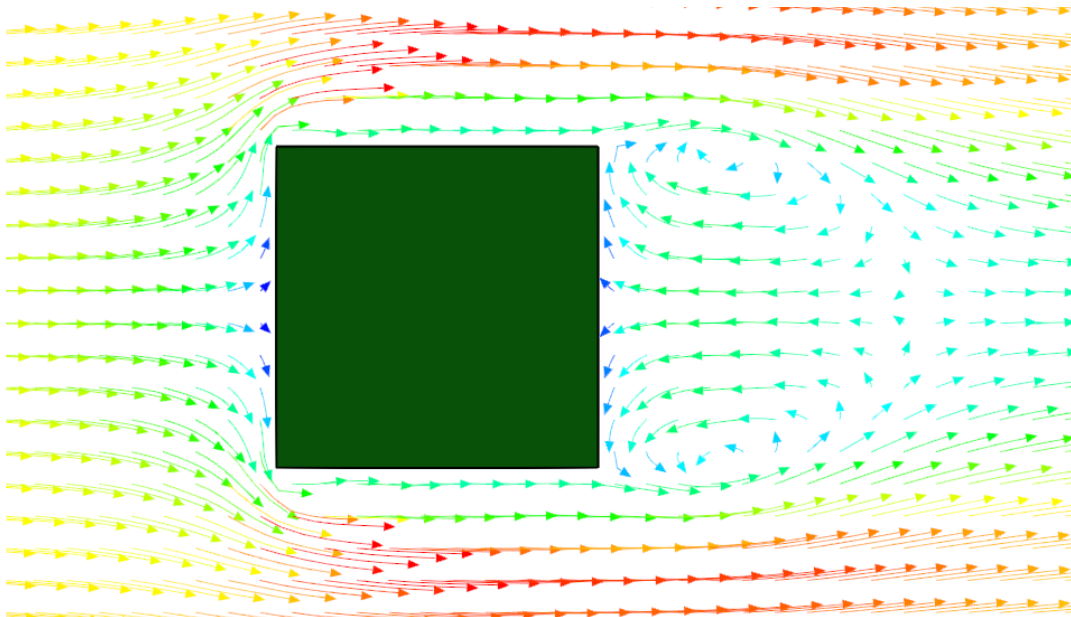
The cutting plane data can be exported as raw data (.tex file) as explained in the [Export data](#) section:

Main menu > Simulation Data > Export cutting plane data to raw format

Local values of the field being visualized on a **Cutting Plane** are accessible via the option *Look up value* of the [Toolbar Data Processing](#), . This displays the value of the visualized field at the cursor location.

7.2.2.2 Vectors

The velocity vectors can be visualized on a **Cutting Plane** by selecting the **Vectors** visualization mode.



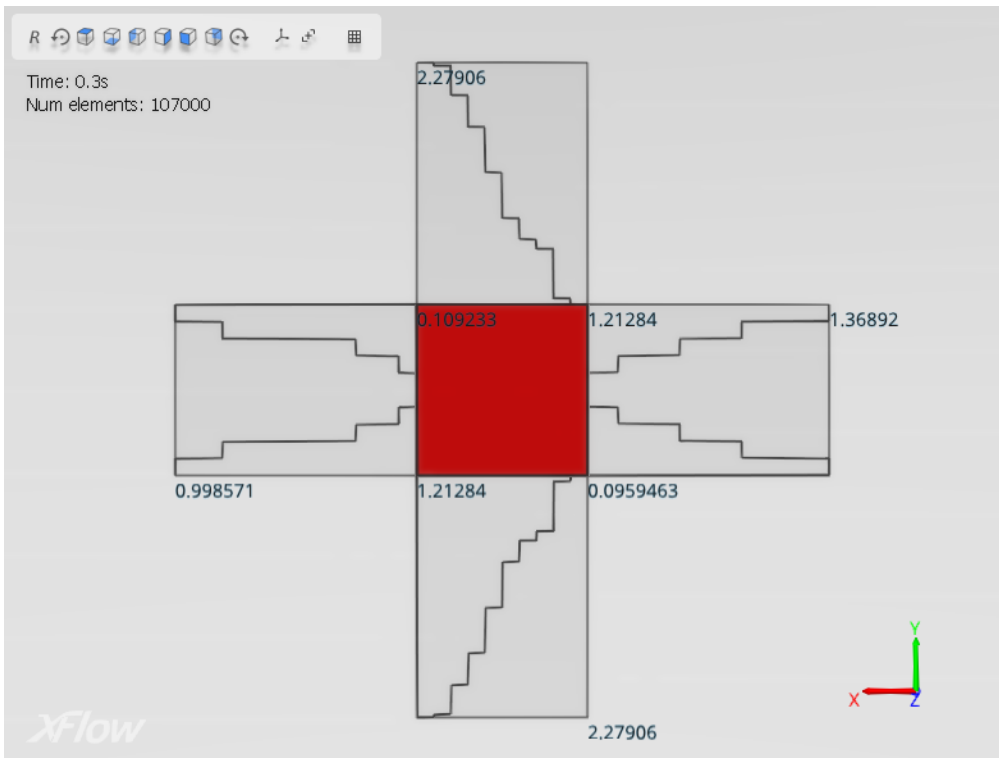
Detail of plane vectors close to the object

The velocity field is represented by arrows that can be customized with the following parameters:

- ↳ **Uniform distribution (On/Off)**; If enabled, it allows the user to see a uniform distribution of the plane vectors, in case a [refinement algorithm](#) has been used in the simulation.
- ↳ **Arrows density**: To change the density of the vectors. The arrow density ranges between 1, that corresponds to one arrow per lattice point ([domain structure](#)), and 0 corresponding to a default minimum number of vectors.
- ↳ **Arrow length**: To change the length of the arrow shaft.
- ↳ **Field**: Arrows are colored according to this field magnitude on the cutting plane.

7.2.2.3 Surface Field distribution

At least one geometry is required to see the **Surface field distribution**. XFlow will show 4 graphs around the object: top, bottom, left and right; showing the projection of a given field on the intersection between the geometry and the plane. For each of them the maximum and minimum value is indicated, as shown in the figure below:



Cutting plane of the pressure field distribution over a car

The **Surface field distribution** visualization mode is affected by some of the options available in the [General](#), namely: **Data**, **Interpolation mode**.

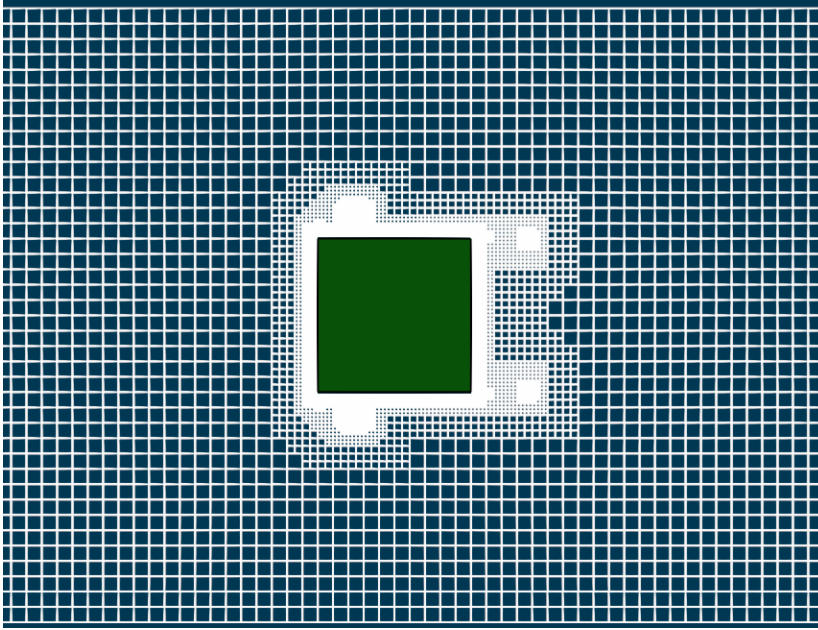
The four graphs can be exported numerically in separated files as explained in the [Export data](#) section:
Main menu > Simulation Data > Export data of cutting plane field distribution

The visualization of the surface field distribution can be customized as follows:

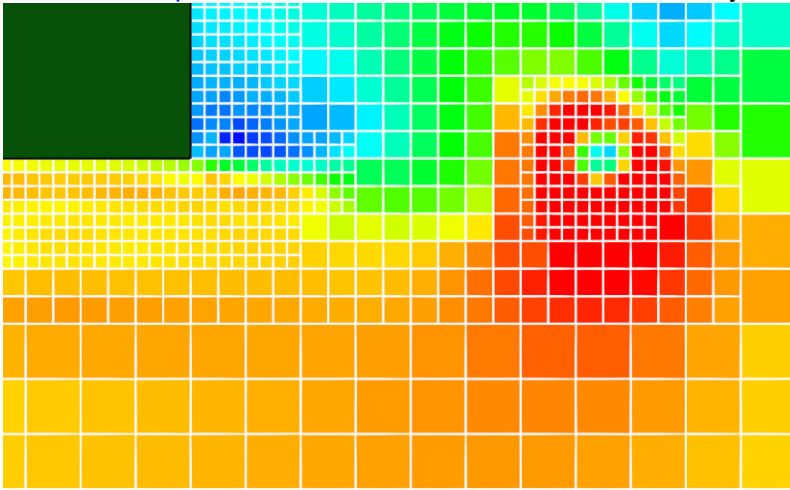
- ↳ **From shape:** Drop-down menu that allows the user to select the **Shape(s)** to which the surface field distribution is applied. By default **All** shapes are selected.
- ↳ **Surface info:** This is the field to be projected on the geometry. These are: 3d field, Cp, Cf, Y+, P+ (see [Show surface info](#)).

7.2.2.4 Domain structure

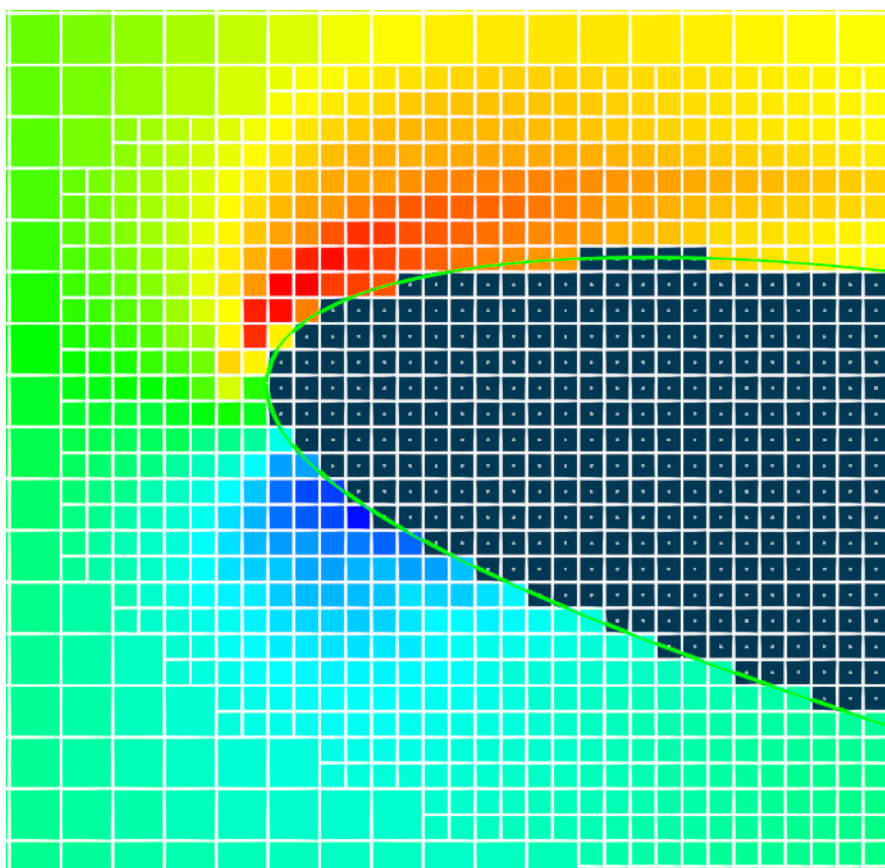
The fluid domain structure can be observed selecting the domain structure as visualization mode. This shows the lattice structure as a voxelization where lattice nodes are located in the center of every square.



When the [interpolation](#) is disabled, the entire voxel is colored by the value at the lattice node:

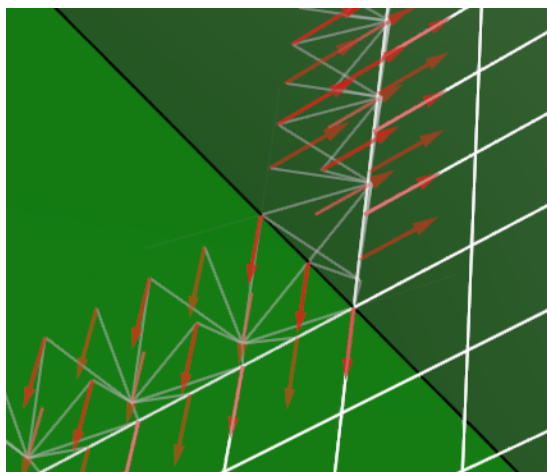


Lattice nodes inside moving geometries will be marked with a dot inside in order to differentiate fluid and solid regions:

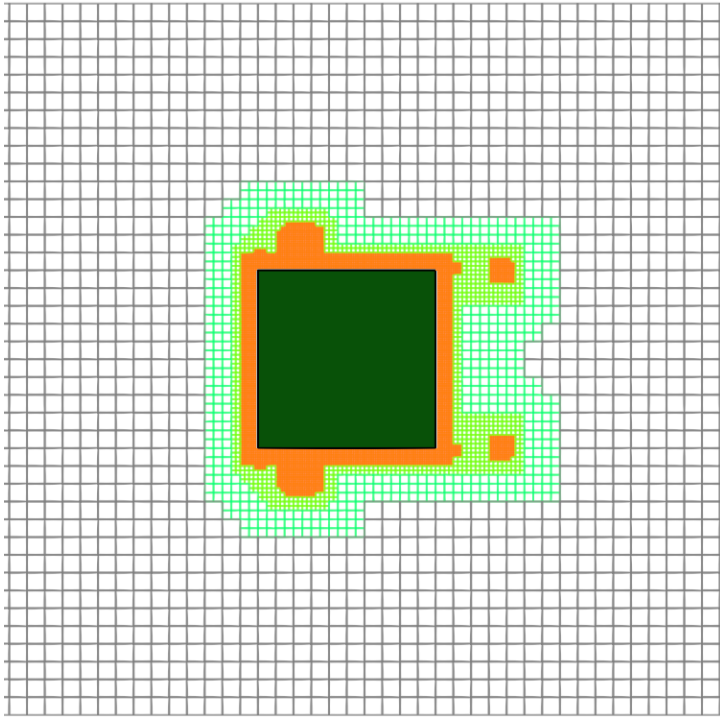


Additionally, it is possible to visualize the following parameters in a cutting plane domain:

- ↪ **Broken links (On/Off)**: shows the broken links and surface normals (only available in [Expert mode](#)). Broken links are the lattice velocity directions intersecting the geometry, hence defining the geometry discretization.



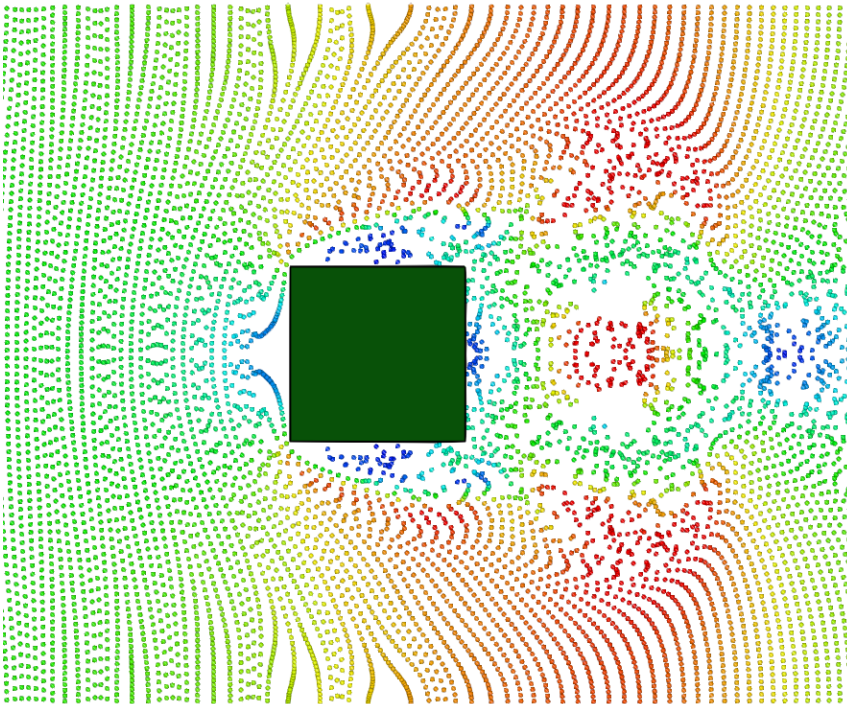
- ↪ **Color by level (On/Off)**: colors each lattice level with a different color.



 **Tip:** If [buffer zone length](#) greater than zero is used, it is useful to color the domain structure by lattice levels in order to observe the buffer zone.

7.2.2.5 Markers

Markers visualization mode is only available for [Single phase](#) simulations and when the markers computation has been activated before running the computation ([+ Info](#) [Project Tree](#) > [Simulation](#) > [Store data](#) > [Compute markers](#)).



Markers with spheres representation

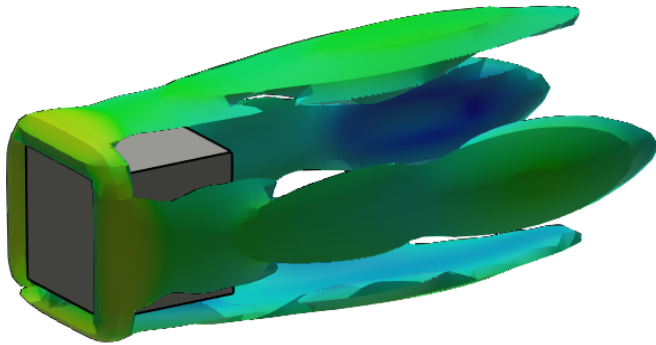
Representation of markers can be customized according to the following options:

- **Points:** Markers are represented by points, colored according to the field selected in:
 - ↳ **Fields** (see [visualization fields](#))
- **Arrows:** Markers are represented by arrows describing the velocity field:
 - ↳ **Uniform length On/Off.** If disabled, the arrow length is proportional to the velocity magnitude at the point where the marker is placed. If enabled, the user can select the arrow length with:
 - ↳ **Arrow length**
 - ↳ **Fields:** To choose the field according to the magnitude of which the arrows are colored (see [visualization fields](#))
- **Spheres:** Markers are represented by spheres, colored according to the field selected in:
 - ↳ **Fields** (see [visualization fields](#))

7.2.3 Isosurfaces

Project Tree > Post-Processing > Isosurfaces

An isosurface is a surface defined by a constant value of a field. Since the field is constant on the surface, it can be colored according to the value of other variables at the surface points. This is shown in the figure below:



Isosurface of vorticity colored by velocity

To create an isosurface do **right click** on the **Isosurfaces** branch of the *Project Tree* and choose **Add Isosurface**, or press the button  on the [Post-Processing toolbar](#).

The user can define and customize the isosurface using the **Isosurface** branch options:

- ↳ **Field**: A drop-down menu which shows the available [fields](#) for the generation of isosurfaces. A constant value of this field will define the isosurface topology (see **Value** below).
- ↳ **Value**: This option allow the user to define the value (of the **Field** chosen above) used to build the isosurface.
- ↳ **Coloured by field**: An isosurface represents a constant field, thus another field can be used to color the surface, for better representation. The user can choose one of the [fields](#) available in XFlow, or just set it to **Off** to visualize it with a constant color (see [visualization material](#)).
- ↳ **Joint Levels**: This option smoothes out the transition between levels on a multi-resolution case, at the cost of taking longer to compute.

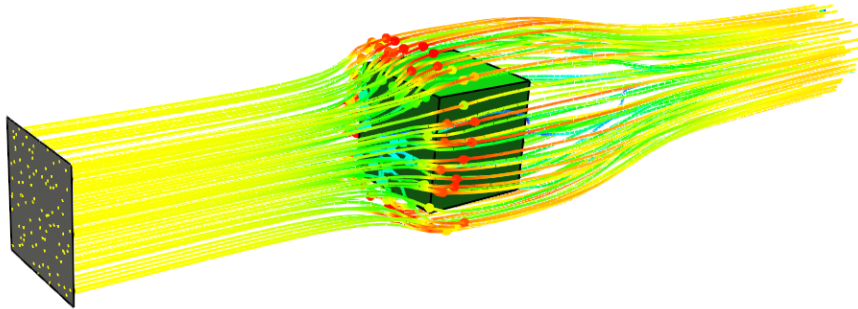
Once an isosurface is created, if either the **Field** or **Value** settings are modified the isosurface has to be refreshed to regenerate it according to the new setup. To do so, **right click** on **Isosurface** and select the option **Recompute isosurface** or just click on the refresh button .

7.2.4 Stream tracers

Project Tree > Post-Processing > Stream tracers

Stream tracers integrate the streamlines in the velocity field and so allow the tracking of the Lagrangian particles along the flow.

The particles can be mass-less particles (passive) advected by the fluid, or solid spherical particles with physical properties ([Discrete Phase Model](#)).



Passive stream tracers showing the flow path around a cube

To create an stream tracer do **right click** on the **Stream tracer** branch of the *Project Tree* and choose

Add stream tracer, or press the button  on the [Post-Processing toolbar](#).

The **Stream tracer** feature has a complex data structure that if well understood makes this feature very versatile. This data structure is as follows:

Stream tracer	Behaviour (+Info)	Passive DPM
	Path lines (+Info)	From frame To frame
	Show (+Info)	Path line Current marker Size factor colored by field

Behaviour

The particle behaviour defines the physics represented by the stream tracers:

- **Passive:** The particles are mass-less particles (passive) advected by the fluid. If the user sets up this feature properly, he will be able to visualize stream lines or flow trajectories. See more details in [Passive](#).
- **DPM:** This option define the particles (tracers) as solid spherical particles with physical properties. DPM is a simplified but useful method to simulate a solid dispersed phase or bubbles in the matrix of another fluid (either liquid or gas). See more details in [DPM](#).

Path lines

Stream-tracer path-lines are generated using the information of the numerical results in the frames range limited by:

- ↳ From frame
- ↳ To frame

Show

The visualization of the stream tracers can be customized here with the following options:

- ↳ **Path line:** Off / Line / Ribbon / Spheres chain / Tube
- ↳ **Current maker:** Off / Point / Sphere
- ↳ **Size factor:**
- ↳ **colored by:**

- **Particle attribute:**

Id: each particle is colored by a different color according to its **ID**.

Velocity: the particles are colored by their own velocities that could differ from the fluid velocity.

Size: the particles are colored according to their diameters.

- **3d field:** See [Fields](#)

⚠ Please note: Tracers will not collide with simulation entities with **Enabled boundary condition: Off**, and with post-processing entities. They will only collide against active simulation geometries.

Compute stream tracers

After setting/modifying the parameters, the stream tracers have to be computed

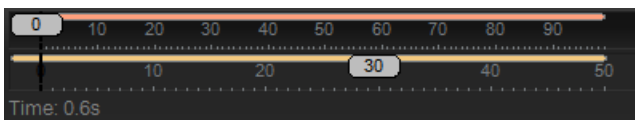
Stream Tracers > Right click **Tracer #** > **Recompute** or



Timeline

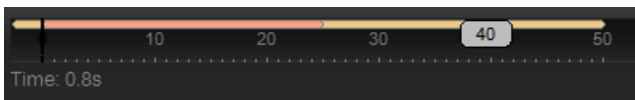
Visualization of stream tracers enables a second timeline.

- For steady tracers, this second timeline over the standard one refers to the integration time set in **Stream Tracers > Tracer # > Timing**.



Timeline for a steady tracer with reference frame = 30

- For transient tracers, this second timeline is superposed to the standard one and therefore it refers to the physical time.



Timeline for a transient tracer between frames 0 and 25

Function Viewer

In the *Function Viewer* it is possible to monitor some of the particles properties, such as the number of active particles. For more information go to [Post-Processing-Function Viewer](#)

7.2.4.1 Passive

Passive stream-tracers allow the user to track mass-less Lagrangian particles along the flow. It is thus useful to observe the classical field lines, such as streamlines, streaklines or trajectories; the visualization of which is controlled by the following data structure:

Stream tracer > Passive	Source (+Info)	Inlet Number of tracers Particles flux rate
	Particle properties (+Info)	Data
	Timing (+Info)	Transient Backtracking

Source

The source of stream tracers is a geometry **Shape**, usually a line or plane. If the source is not one of the geometry surfaces, the user can create a post-processing entity (see [Geometry creation](#)).

- ↳ **Inlet:** Shape acting as the tracers inlet. If "None" is selected, no tracer inlet will be used and an [initial volume distribution](#) could be used instead.
- ↳ **Number of tracers:** Number of particle emitters (seed points), the distribution of which over the source Shape is randomly generated but it is kept constant in time. If it is set to zero, only one emitter is created but its position is not fixed in time, but it is randomly modified from frame to frame.
- ↳ **Particles flux rate:** Total number of particles (tracers) emitted per second. Note, that if it is set to zero, it will generate just one sequence of particles.

Example: Passive stream tracers I

Number of tracers: 2 Particles flux rate: 10 particles/second	→	Two emitters are randomly created on the source Shape. The location of these emitters is fixed. From each emitter 5 particles are released per second.
--	---	--

Example: Passive stream tracers II

Number of tracers: 0 Particles flux rate: 10 particles/second	→	One emitter is randomly created on the source Shape. The location of this emitter is random. From the emitter 10 particles are released per second.
--	---	---

Example: Passive stream tracers III

Number of tracers: 2 Particles flux rate: 0 particles/second	→	Two emitters are randomly created on the source Shape. The location of this emitter is fixed. From each emitter only 1 particles is released.
---	---	---

Example: Passive stream tracers IV

Number of tracers: 1 Particles flux rate: 1 particles/second	→	One emitter is randomly created on the source Shape. The location of this emitter is fixed. From the emitter 1 particles is released per second.
---	---	--

Initial conditions

Initial conditions for tracers can be set in this section.

↳ Particles distribution **On / Off**

- **Source volume** The volumetric geometry where the initial particles will be created
- **Number of particles** The number of particles that will be created initially inside the source volume

Particle properties

Passive tracers are mass-less and physical properties of the particle are thus not required.

In this section, only the velocity field used to calculate the particles-flowlines can be chosen between:

↳ **Data:** Instantaneous / Averaged

Timing

The transient behaviour of the pathlines can be switched On/Off.

- **On:** the tracers are computed taking into account the variations of the field in time (e.g. trajectories) and therefore require the initial and last frames.

↳ **Initial frame**

↳ **Last frame**

↳ **Loop: On/Off.** When enabled, allows to loop the tracer calculation from between initial and last frame. The number of loop iterations must be provided. This option is only available for transient tracers.

Example: a pump revolution is done from frame 5 to frame 15 and one wants to compute the tracers for 10 revolutions. The initial frame will be set to 5 and the last frame to 15, and the number of iterations will be set to 10.

- **Off:** then the tracers are steady and require the reference frame at which the tracers will be computed (e.g. stream line). XFlow will take the reference frame and compute the tracers according to the flow field at the corresponding simulation time.

↳ **Reference frame**

↳ **Time**

↳ **Frequency**

 **Please note:** The **trajectory** of a single particle is given by the [pathline](#) resulting from the following setup:
Number of tracers = 1
Particles flux rate = 0
Timing = On

 **Please note:** The **streaklines** of a flow are represented by the tracers position at given frame, resulting from the following setup:
Number of tracers = 1
Particles flux rate = 20
Timing = On

 **Please note:** The **stream lines** of the flow at a given instant of time are given by the [pathlines](#) resulting from the following setup:
Number of tracers = 10
Particles flux rate = 1
Timing = Off

- ↳ **Backtracking: On/Off.** Switch to enable/disable the tracers back-track. If enabled the passive particles placed at the **Source> Inlet** are traced back in time, so the tracer path leads back to its source in the flow inlet or far field.

Example: Transient passive tracers backtracking

Source:
Inlet: Surface1
Number of tracers: 500
Particles flux rate: 0 particles/second
Timing:
Transient: On
Initial frame: 0
Last frame: 100
Backtracking: On

→ 500 particles are randomly created on Surface1 (Source>Inlet) at the instant of time defined by Last frame (100). The position of these particles is calculated from a given initial time (defined by Initial frame, 0) to their final position on Surface1 at the time given by Last frame (100).

Example: Non-transient passive tracers backtracking

Source:
Inlet: Surface1
Number of tracers: 500
Particles flux rate: 0 particles/second
Timing:
Transient: Off
Reference frame: 100
Time: 1 s
Frequency: 100 Hz
Backtracking: On

→ 500 particles are randomly created on Surface1 (Source>Inlet) at the instant of time defined by Reference frame (100). The up-stream lines of these particles are calculated with a resolution of 100 points (Time*Frequency).

7.2.4.2 DPM

Discrete Phase Model (DPM) allows the user to simulate particles with mass. For more information about this model, see [Modeling Discrete Phase](#).

The data structure for the stream tracers behaving as DPM is similar to that of the [passive](#) tracers. As shown below, the main difference between DPM and Passive tracers is found in the Particle properties definition, where physical properties are required for DPM particles but not for Passive ones.

Stream tracer > Passive Source ([+Info](#))Inlet
Number of tracers
Particles flux rate

Initial conditions

Particles distribution
Velocity lawsParticle properties ([+Info](#))Density
Diameter
Standard deviation
Restitution coefficient
Turbulent dispersion
Acceleration laws
DataTiming ([+Info](#))

Transient

Analysis ([+Info](#))

Number of bins

Source

The definition of the source of this type of stream tracers is analogous to that of Passive tracers. Please, go to [Source](#).

Initial conditions

↳ **Particles distribution** On / Off

- **Source volume** The volumetric geometry where the initial particles will be created
- **Number of particles** The number of particles that will be created initially inside the source volume

↳ **Velocity laws** Initial velocity $u(t=0)$. It can be defined as a function of properties of both phases. See [Functions](#).

Particle Properties

↳ **Density** Particles density in kg/m^3 .

↳ **Diameter** Characteristic particles diameter, required to calculate the drag coefficient. The larger the heavier, and the higher the drag.

↳ **Standard deviation**: Standard deviation in particle diameter in meter. This allows to use a variable diameter distribution.

↳ **Normal velocity restitution**: Restitution coefficient of the normal velocity component after colliding a wall. If it is equal to 1, the particles will bounce with the same incident normal velocity. This factor can also be defined as a law in function of the particles or fluid variables. It can be a function of the DPM variables, see [Functions](#).



Normal velocity restitution = 0

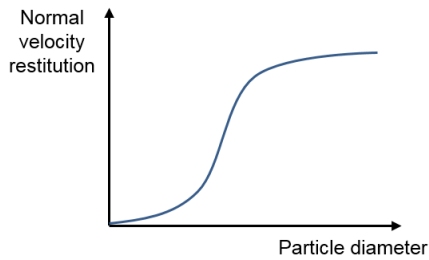


Normal velocity restitution = 0.5



Normal velocity restitution = 1

7.2 Post-Processing tree



Example of normal velocity restitution as a function of the particle diameter

↪ **Tangential velocity restitution:** Restitution coefficient of the tangential velocity after colliding a wall. It basically represents the amount of tangential velocity lost by friction to the walls. If the value is set to 1 then the particle slides perfectly as on a free-slip wall, if the value is set to 0.5 the particle will have a tangential velocity decelerating by half. This factor can also be defined as a law in function of the particles or fluid variables. See [Functions](#).

↪ **Wall adherence model: On / Off**

- **Normal restitution threshold** Threshold value of the normal velocity restitution defined above below which the particle will adhere to the wall.

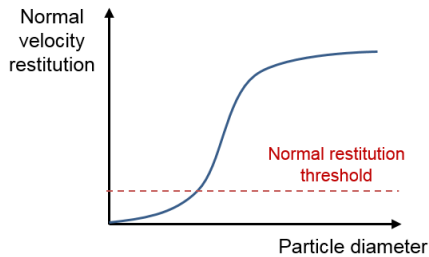


Illustration of the normal restitution threshold application: all particles below the red dashed line will adhere to the wall.

↪ **Turbulent dispersion** Turbulent dispersion can be enabled only when Averaged results have been saved and **Data = Averaged**.

↪ **Acceleration laws** ($a_{Ext(p)}$) External acceleration that only affects the disperse phase (not the matrix fluid). It can be defined as a function of properties of both phases. See [Functions](#).

↪ **Data** Instantaneous or Averaged data.

Timing

The definition of the timing of DPM stream-tracers is analogous to that of Passive stream-tracers. Please, go to [Timing](#).

Histogram

↪ **Number of bins:** The particles distribution histograms is available in the [Function Viewer](#) when the diameter Standard deviation value is not 0.

XFlow uses a distribution function to generate particles with diameters between the maximum and the minimum diameter value. This distribution function is divided by the Number of bins provided by this parameter in order to calculate the distribution of the particles along the diameter range. Therefore the higher the number of bins, the more accurate the discretization of the distribution function.

7.2.4.2.1 Modeling discrete phase

Besides solving the transport equations for the continuous phase, XFlow can also calculate the transport of a discrete phase consisting of spherical particles (representing: droplets, dust, bubbles, etc) dispersed in the continuous phase. To this end, XFlow solves the following equation of motion:

$$\frac{d\vec{u}_p}{dt} = \vec{a}_{D(f-p)} + \vec{a}_{Ext_{f,p}} + \vec{a}_{Ext_p}$$

where:

- $\vec{u}_p$: Particle velocity
- $\vec{a}_{D(f-p)}$: Acceleration of the particle due to the drag force exerted by the fluid on the particle.
- $\vec{a}_{Ext_{f,p}}$: External acceleration affecting both phases (f,p), e.g. gravity.
- $\vec{a}_{Ext_p}$: External acceleration affecting only the disperse phase (p).

The discrete phase model (DPM) characteristics are thus the following:

1. It calculates the particles trajectory using a Lagrangian formulation that includes the discrete phase inertia, hydrodynamic drag, and the effect of external forces, for both steady and transient flows
2. It predicts the particles dispersion due to turbulence
3. One-way coupling, i.e. fluid flow influences particles via drag and turbulence, but particles have no influence on the fluid flow
4. It accounts for particle-wall collisions, but particle-particle ones are not considered.

Limitations

The discrete phase formulation used by XFlow contains the assumption that the second phase is sufficiently diluted that particle-particle interactions and the effects of the particle volume fraction on the continuous phase are negligible. In practice, these issues imply that the discrete phase must be present at a fairly low volume fraction, usually less than 10-12%. Nevertheless, it is possible to solve problems in which the mass loading of the discrete phase exceeds this proportion.

The discrete phase model is available once the matrix flow has been computed.

The setup is done in **Post-Processing > Stream Tracers > Tracer #**, while the visualization is controlled in **Post Processing > Stream tracers > Tracer # > Show** (see [Stream tracers](#)).

7.2.5 Plot lines

Project Tree > Post-Processing > Plot lines

A data plot line allows the user to monitor in the [Function Viewer](#) the evolution of a field along a line.

To create a **Plot line** do **right click** on the **Plot line** branch of the *Project Tree* and choose **Add data plot line**, or press button  on the [Post-Processing toolbar](#).

The data structure to customize the **Plot lines** is very simple:

- ✎ **Vertex 1**: Point defining where the line starts
- ✎ **Vertex 2**: Point defining where the line ends
- ✎ **Number of samples**: Number of points (along the line) must be used to plot the data
- ✎ **Field**: [Field](#) to be plotted along the given line (**Vertex 1** to **Vertex 2**)
- ✎ **Data**: Type of data to be plotted: instantaneous, averaged or standard deviation (The last two are available only if [calculated in the simulation](#)).
- ✎ **Interpolation mode**: **On/Off**. If enabled the results plotted on the line are [interpolated](#) values, else

7.2 Post-Processing tree

raw data is used.

↳ **Auto-refresh: On/Off.** Lines are computed automatically when a new frame is loaded if this option is enabled.

⚠ Please note: If **Interpolation mode: On**, the interpolation scheme would be either **Convolution** or **MLS** according to what is selected in **Project Tree > Post-Processing > General > Interpolation**

If the user changes the setup of the plot lines, it will be necessary to refresh the data by **right clicking** on **Line #** in the *Project Tree* and then selecting **Refresh**, or just pressing the refreshment button .

To display the results along the line, open a *Function Viewer* window and do **right click**, a drop-down menu will appear showing the option: **Plot lines > Line #**. This displays the evaluated field (Y-axis) against the line length (X-axis). For transient cases, playing forward  the results for each frame will be shown, updating the curve on the function viewer.

7.2.6 Sensors

Project Tree > Post-Processing > Sensors

Sensors allow the user to monitor the evolution with time of a field at a point.

Single sensor

To add a sensor do **right click** on the **Sensors** branch of the *Project Tree* and select **Add sensor**, or press button  on the [Post-Processing toolbar](#). Once a sensor is created, the user should define the following parameters:

↳ **Position:** Coordinates defining the location of the sensor.

↳ **Field:** Choose the [field](#) to be monitored.

↳ **Data:** It can be chosen among: instantaneous, averaged or standard deviation (The last two are available only if [calculated in the simulation](#)).

↳ **Interpolation mode: On/Off.** Uses interpolated data.

⚠ Please note: If **Interpolation mode: On**, the interpolation scheme would be either **Convolution** or **MLS** according to what is selected in **Project Tree > Post-Processing > General > Interpolation**

↳ **Frames range:**

- **All:** Reads the sensor values for all the frames available in the simulation.
- **Only current:** Reads the sensor value only for the current selected frame.
- **Custom:** Reads the sensor value for a custom range of frames, between **From frame** until **To frame**.

Import sensors from file

To import a sensor or a group of sensors from a text file, go to **Main Menu > Post-Processing > Numerical**

analysis > Import from file > Sensors > and choose the text file containing the sensors data. The sensors must be defined in the text file according to the following syntax:

```
coordinate-X coordinate-Y coordinate-Z
Name_of_the_field_to_be_monitored
Data_type
Interpolation_switch
```

Example: groupOfSensors.txt

```
1 1 1
Velocity
Instantaneous
On

3 0.5 0.2
Vorticity
Instantaneous
On

2.0 2.0 1.0
Vx
Instantaneous
Off
```

Set sensor by mouse

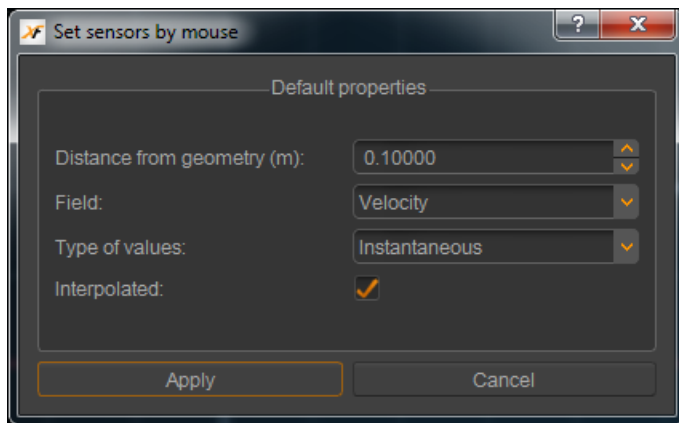
A single sensor can be created:

- **Main menu > Post-Processing > Numerical analysis > Set sensor by mouse**, or 

XFlow provides a window to define:

- ↳ **Distance from geometry(m)** : Choose the distance in the normal direction between the geometry and the sensor.
- ↳ **Field**: Choose the [field](#) to be monitored
- ↳ **Type of values**: It can be chosen among: instantaneous, averaged or standard deviation (The last two are available only if [calculated in the simulation](#)).
- ↳ **Interpolation mode**: Enables or disables interpolated data.

7.2 Post-Processing tree



Finally, click in the Graphic View with mouse **middle button** to create the sensor.

To display the results at the sensor, open a *Function Viewer* window and do **right click on it**, a drop-down menu will show the following option **> Sensors > Sensor #**. This displays the evaluated field (Y-axis) against the time (X-axis).

Please note: The *Function Viewer* plots the value of the field at the sensor position versus time. It is thus relevant to see the transient evolution of the field at the location of the sensor, and not only an instantaneous value.

When changing any of the sensor parameters (**Position, Field, Data, Interpolation mode**) it is necessary to refresh the data sensor again by **right clicking** on **Sensor #** in the *Project Tree* and then selecting **Refresh**, or just pressing the refreshment button

Tip: To avoid refreshing the sensors it is good practice to use the **Probes**. **Probes** are available in **Project Tree > Simulation > Store data > Probes** and are predefined points where data are measured and saved during computation. The graph of the flow variables at the probes are directly available in the *Function Viewer*.

7.2.7 Surface integrals

Project Tree > Post-Processing > Surface integrals

A surface integral allows the user to integrate a scalar over a surface and obtain quantities such as the average, root mean square, or standard deviation values of a scalar field on a surface.

In this section the variables are denoted as follows:

I , Surface integral

A , Surface [m^2]

Φ , Scalar field

Create the surface integral

To create a surface integral do **right click** on the **Surface integrals** branch of the *Project Tree* and choose **Add surface integral**, or press button on the [Post-Processing toolbar](#).

The data structure to customize the **Surface integrals** is as follows:

↳ **Geometry**: Surface where the field is integrated.

↳ **Integral type**:

- **Sum**: integrates the scalar Φ over the **Geometry** surface A. This returns the sum of the discrete values expressed in the units of the field multiplied by meter square [**Field** units $\times \text{m}^2$]. It is defined by the following relation:

$$I = \int_A \Phi \, dA$$

This integral type can be used to compute for instance a fluid force exerted on a body, or a mass flow through a section.

 **Tip**: To compute the mass flow passing through an arbitrary section, create a custom field such as: `[rho*(vx*nx+vy*ny+vz*nz)]`. Compute the sum-type surface integral of this custom field to get the mass flow in kg/s.

- **Average**: integrates the scalar Φ over the **Geometry** surface A, normalized with the **Geometry** surface A. This returns the area-average value of the field, expressed in the units of the field [**Field** units]. It is defined by the following relation:

$$I = \frac{1}{A} \int_A \Phi \, dA$$

This integral type can be used to compute for instance the area-averaged pressure on a geometry, or submersed fraction of an object integrating the VOF field.

- **Standard deviation**: integrates over the **Geometry** surface A the deviation of the scalar Φ from the average value Φ_{avg} . This returns the standard deviation of the scalar field over the surface, expressed in the units of the field multiplied by meter square [**Field** units $\times \text{m}^2$]. It is defined by the following relation:

$$I = \sqrt{\int_A (\Phi - \Phi_{\text{avg}})^2 \, dA}$$

- **RMS**: computes the square root of the integrated quadratic terms over the **Geometry** surface A, normalized with the **Geometry** surface A. This returns the root mean square of the scalar field over the surface, expressed in the units of the field multiplied by meter square [**Field** units]. It is defined by the following relation:

$$I = \frac{1}{A} \sqrt{\int_A \Phi^2 \, dA}$$

- **Max**: returns the maximum value of the selected field on the surface.

- **Min**: returns the minimum value of the selected field on the surface.

↳ **Field**: **Field** to be averaged on the surface, **custom fields** are also available, which can be defined as a function of the **Geometry** normal components (**nx**, **ny**, **nz**).

 **Tip:** To compute the mass flow passing through an arbitrary section, create a custom field such as: $[\text{rho} * (\text{vx} * \text{nx} + \text{vy} * \text{ny} + \text{vz} * \text{nz})]$. Compute the surface integral of this custom field with **Normalization = Off**.

↳ **Data:** Type of [data](#) to be used for the integral: instantaneous, averaged or standard deviation. The last two are available only if [calculated in the simulation](#).

↳ **Interpolation mode:** **On/Off**. If enabled the integral results on the surface are [interpolated](#), else non-interpolated data is used.

 **Please note:** If **Interpolation mode: On**, the interpolation scheme would be either **Convolution** or **MLS** according to what is selected in **Project Tree > Post-Processing > General > Interpolation**

↳ **Sample points:** Number of points to sample the surface of the object. The more points, the more accurate the integral will be however it may increase the computation time of the integrals.

↳ **Frames range:**

- **All:** Computes the integral for all the frames available in the simulation.
- **Only current:** Computes the integral only for the current selected frame.
- **Custom:** Computes the integral for a custom range of frames, between **From frame** until **To frame**.

↳ **Compute both sides:** **On/Off**. If enabled, for each sampled point two field evaluations are made: one on the surface side where the normals are pointing to and another one on the reverse side; otherwise the surface integral is only calculated over the surface side to which the normals are pointing.

Compute the surface integral

To compute a surface integral, **right click** on **Surface integrals** in the *Project Tree* and select **Refresh**, or just press the refreshment button .

↳ If the option **Only current frame** is set to **Off**, XFlow will load the simulation frames one after one automatically and the surface integral computed for each of them to get the time history of the integral.

↳ If the option **Only current frame** is set to **On**, XFlow will compute the surface integral on the current selected frame.

 **Tip:** You can compute all surface integrals at once by selecting all surface integrals (select the first surface integral and press SHIFT + select the last of the list) and then using the refresh icon  in the post-processing toolbar.

Visualize the surface integral

To observe the surface integral history, open a *Function Viewer* window and do **right click on it**, a drop-down menu will show, among others, the following option **> Surface integrals > Surface #**. This displays the

integrated value of the field (Y-axis) versus the time (X-axis). If the option **Only current frame** is set to **On**, a single value will be shown as a straight horizontal line.

⚠ Please note: The *Function Viewer* plots the value of the visualization field versus time. It is therefore relevant to see the transient evolution of the integral, and not only an instantaneous value as fluctuations can be expected.

7.2.8 Volume integrals

Project Tree > Post-Processing > Volume integrals

A volume integral allows the user to integrate a scalar over a volume and obtain quantities such as the average, root mean square, or standard deviation values of a scalar field on in the volume of integration.

In this section the variables are denoted as follows:

I , Volume integral

V , Volume [m^3]

Φ , Scalar field

Create the volume integral

To create a volume integral do **right click** on the **Volume integrals** branch of the *Project Tree* and choose **Add volume integral**.

The data structure of the **Volume integrals** is as follows:

↳ **Geometry:** To set the geometry within the **Field** is to be integrated.

↳ **Integral type:**

- **Sum:** integrates the scalar Φ over the **Geometry** volume V . This returns the sum of the discrete values expressed in the units of the field multiplied by cubic meter [**Field** units $\times m^3$]. It is defined by the following relation:

$$I = \int_V \Phi dV$$

This integral type can be used to compute for instance a fluid force exerted on a body, or a mass flow through a section.

💡 Tip: To compute the mass flow passing through an arbitrary section, create a custom field such as: `[rho*(vx*nx+vy*ny+vz*nz)]`. Compute the sum-type surface integral of this custom field to get the mass flow in kg/s.

- **Average:** integrates the scalar Φ over the **Geometry** volume V , normalized with the **Geometry** volume V . This returns the average value of the field in the volume, expressed in the units of the field [**Field** units]. It is defined by the following relation:

$$I = \frac{1}{V} \int_V \Phi dV$$

This integral type can be used to compute for instance the area-averaged pressure on a geometry, or submersed fraction of an object integrating the VOF field.

- **Standard deviation:** integrates over the **Geometry** volume V the deviation of the scalar Φ from the average value Φ_{avg} . This returns the standard deviation of the scalar field in the volume of integration, expressed in the units of the field multiplied by meter square [**Field units** × m²]. It is defined by the following relation:

$$I = \sqrt{\int_V (\Phi - \Phi_{avg})^2 dV}$$

- **RMS:** computes the square root of the integrated quadratic terms over the **Geometry** volume V , normalized with the **Geometry** volume V . This returns the root mean square of the scalar field in the volume of integration, expressed in the units of the field multiplied by meter square [**Field units**]. It is defined by the following relation:

$$I = \frac{1}{V} \sqrt{\int_V \Phi^2 dV}$$

- **Max:** returns the maximum value of the selected field on the surface.
- **Min:** returns the minimum value of the selected field on the surface.

↳ **Field:** The field to be integrated. It can be a [custom field](#).

↳ **Data:** Type of [data](#) to be used for the integral: instantaneous, averaged or standard deviation (the two latest options are available only if [calculated in the simulation](#)).

↳ **Interpolation mode: On/Off.** If enabled the integral results are [interpolated](#), else raw data is used.

⚠ **Please note:** If **Interpolation mode: On**, the interpolation scheme would be either **Convolution** or **MLS** according to what is selected in **Project Tree > Post-Processing > General > Interpolation**

↳ **Frames range:**

- **All:** Computes the integral for all the frames available in the simulation.
- **Only current:** Computes the integral only for the current selected frame.
- **Custom:** Computes the integral for a custom range of frames, between **From frame** until **To frame**.

Compute the volume integral

To compute a volume integral, **right click** on **Volume integrals** in the *Project Tree* and select **Refresh**, or just press the refreshment button .

↳ If the option **Only current frame** is set to **Off**, XFlow will load the simulation frames one after one automatically and the volume integral computed for each of them to get the time history of the integral.

- ✎ If the option **Only current frame** is set to **On**, XFlow will compute the volume integral on the current selected frame.

 **Tip:** You can compute all integrals at once by selecting several volume integrals (select the first volume integral and press SHIFT + select the last of the list) and then using the refresh icon  in the post-processing toolbar.

Visualize the volume integral

To observe the volume integral history, open a *Function Viewer* window and do **right click on it**, a drop-down menu will show, among others, the following option **> Volume integrals > Volume #**. This displays the integrated value of the field (Y-axis) versus the time (X-axis). If the option **Only current frame** is set to **On**, a single value will be shown as a straight horizontal line.

 **Please note:** The *Function Viewer* plots the value of the visualization field versus time. It is therefore relevant to see the transient evolution of the integral, and not only an instantaneous value as fluctuations can be expected.

7.2.9 Custom fields

Project Tree > Post-Processing > Custom fields

The custom fields section allows the user to create new [visualization fields](#) as a [function](#) of the default ones. To create a custom field for post-processing, do **right click on Custom fields** in the *Project Tree* and then select **Add custom field** or press  on the [Post-Processing toolbar](#).

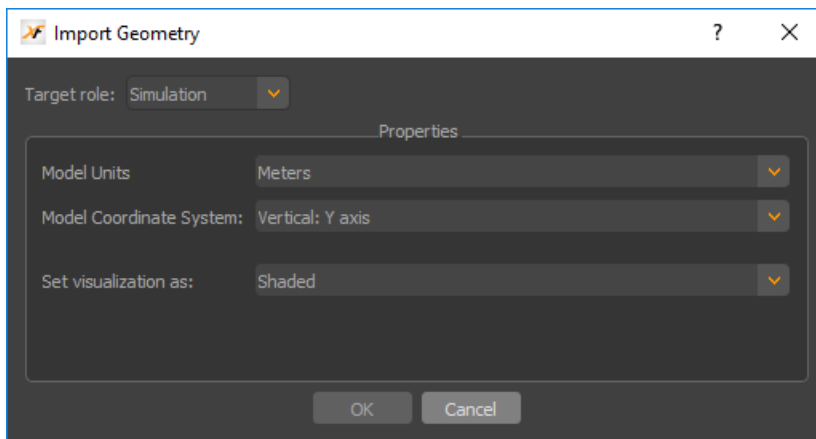
Custom fields appear in the visualization fields list and can thus be visualized in the Graphic View as any other standard field.

7.2.10 Entities

Project Tree > Post-Processing > Entities

During the importation of a geometry, a target role is specified as explained in the [import and export geometry chapter](#). The target role can be either Simulation or Post-processing, in case Post-processing is selected the geometry will be available in **Project Tree > Post-Processing > Entities**

7.2 Post-Processing tree



Target role specified during the geometry importation

Post-processing entities can be used for post-processing purpose and do not participate in the simulation process. These geometries can be used for the following purpose:

- Surface integrals
- Volume integrals
- Stream tracers inlet
- Stream tracers volume source

⚠ Please note: Boundary conditions cannot be applied to faces of post-processing entities.

The behavior of post-processing entities includes [Fixed](#) and [Enforced](#) behaviors similarly to the Simulation entities. An additional Control geometry behavior is however available.

Control geometry

The **Shape** does not take part in the simulation (as if it were [Disabled](#)), but it defines a geometry where it is possible to do post-processing operations like the counting of [DPM particles](#) flowing through, the user can then plot the evolution of the DPM particles within the control geometry in the [Function Viewer](#):

Right click on the *Function Viewer* window > **Stream tracers > Tracer 0 > Control geometries >**

⚠ Please note: This behaviour is, at the moment, only useful when DPM stream tracers are used.

7.2.11 Cameras

Project Tree > Post-Processing > Cameras

To create a custom field for visualization, do **right click** on **Cameras** in the *Project Tree* and then select

Add camera or press  on the [Post-Processing toolbar](#).

A camera can have two **Behaviours**:

- ↳ **Fixed**: equivalent to the [camera of the Graphic View](#). This is to define a fixed point of view.
- ↳ **Forced**: allows the user to enter laws to describe the camera location. In this way it is possible to prescribe a motion for the camera. Here it is very helpful to use the [linearinterpolation](#) function.

A perspective different from the orthographic view can be enabled in:

- ↳ **FOV** (Field Of View): angular extent of scene that is seen from the camera.

 **Tip:** The camera position and direction is set to the one of the active Graphic View by default. You may set your Graphic View to the desired position and create a new camera in order to save the view.

7.2.12 Views

Project Tree > Post-Processing > Views

A camera is the representation of the view displayed in the *Graphic View* window. Each *Graphic View* has its own camera.

Camera settings

- [link to camera](#)
- [location](#)
- [perspective](#)
- [clipping planes](#)

Link to camera

Allows the user to choose between the camera of the *Graphic View* and a camera defined by the user (see [Cameras](#)).

Location

The camera location (in Cartesian coordinates) is described by the following parameters:

- ↳ **From**: Position of the camera or eye point.
- ↳ **To**: Position of the target, where the camera is looking at. It is the center of the view.
- ↳ **Up**: Direction of the vertical axis.

Perspective

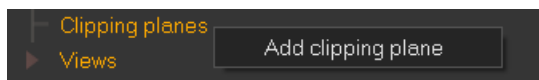
Enables the perspective view. Otherwise, the orthographic view is used.

- ↳ **FOV** (Field Of View): angular extent of scene that is seen from the camera.

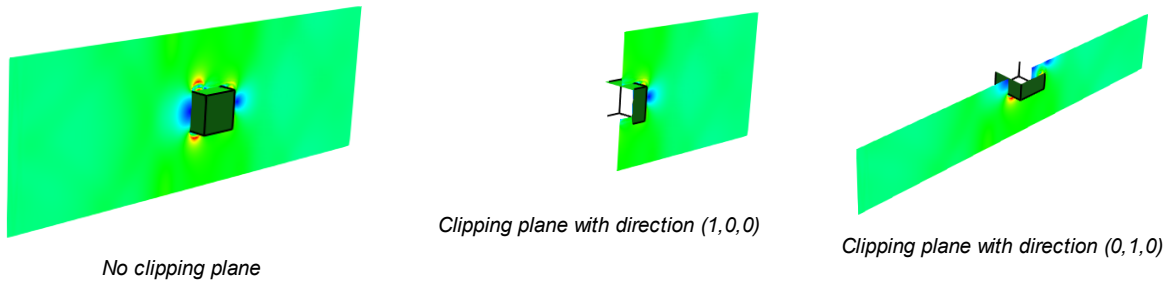
Clipping planes

Clipping planes cut away the portion of the geometry model on one side of the plane and therefore are useful to see inside the model. A plane needs to be defined by the **Origin** (a point the plane passes by) and its **Normal** direction.

Right click the **Clipping planes** string to add a clipping plane.



The visibility depends on the camera position and everything on the other side of the defined clipping plane with respect to the camera position will be shown, while everything on the side of the plane pointing towards the camera position will be clipped.



Tip: Clipping planes can be translated or rotated using the Gizmos. To do so, select the clipping plane in the Project Tree using the Object Filter icon , and do the following:

- **drag** the Gizmo axis normal to the clipping plane to translate it, or alternatively
- **click** on the Gizmo axis and set the translation distance (in meter) in the translation dialogue-box
- **drag** the corresponding Gizmo arc, or alternatively
- **click** on a Gizmo arc and setting the rotation angle (in degrees) in the rotation dialogue-box

Manual clipping planes

This feature allows the user to manually adjust camera's clipping planes.

In Automatic Mode, XFlow will try and adjust the near and far rendering range to be as small as possible. However, if the Manual mode is activated the user can define a value for the near plane and for the far one.

7.3 Visualisation fields

XFlow dispose of a set of data which can be visualized after a computation has run. They can be used for all the post-processing features of XFlow: cutting planes, isosurfaces, stream tracers, sensors, plot lines and plot surfaces.

Here is the list of the available fields in XFlow

Velocity

The modulus of the velocity V (in m/s) is defined as:

$$V = \sqrt{V_x^2 + V_y^2 + V_z^2}$$

Static pressure

The static pressure P_{static} (in Pa) is defined as:

$$P = \frac{\rho C_s^2}{\gamma}$$

where ρ stands for the fluid density, C_s is the speed of sound and γ is the adiabatic index.

XFlow uses the ideal gas state equation for gases and a stiff state equation for liquids. The pressure shown at the Post-Processing is a **gauge** static pressure.

Vorticity

The vorticity ω (in 1/s) is the modulus of the velocity curl:

$$\omega = \left| \vec{\nabla} \times \vec{V} \right|$$

Q-criterion

The Q-criterion defines a vortex as a connected fluid region with a positive second invariant of the velocity gradient.

Vx , Vy, Vz

The different components of the velocity vector (in m/s) are V_x , V_y and V_z , respectively to the X, Y and Z direction.

Total pressure

The total pressure P_{total} (in Pa) is the addition of the dynamic pressure and the static pressure:

$$P_{total} = \frac{1}{2}\rho V^2 + P_{static}$$

The first term is the dynamic pressure. The pressure shown at the Post-Processing is a **gauge** total pressure.

Turbulence intensity

The turbulence intensity I (in %) is defined as:

$$I = \frac{u'}{U}$$

where U is the mean velocity, and u' the root-mean-square of the turbulent velocity fluctuations which can be described as:

$$u' = \sqrt{\frac{1}{3}(u_x'^2 + u_y'^2 + u_z'^2)} = \sqrt{\frac{2}{3}k}$$

with k the turbulent kinetic energy.

The mean velocity U is computed from the three mean velocity components:

$$U = \sqrt{U_x^2 + U_y^2 + U_z^2}$$

Volume of liquid phase

This is a function equal to 1 for the liquid phase, 0 for the gas phase. It is only available in [Free surface](#) or [Multiphase](#) simulations and allows the user to see the presence of fluid in the domain.

Temperature

The temperature T (in K) describes the temperature of the fluid in the domain. This field is only available if the [energy equation](#) is solved.

Effective viscosity

The effective viscosity $\mu_{\text{effective}}$ (in Pa·s) is defined as:

7.3 Visualisation fields

$$\mu_{\text{effective}} = \mu_{\text{molecular}} + \mu_{\text{turbulent}}$$

where the $\mu_{\text{molecular}}$ is the dynamic viscosity specified in the Materials tab and $\mu_{\text{turbulent}}$ is the turbulent viscosity.

Custom field

The user can also visualize a [custom field](#) if any has been previously created.

Scalar

Shows the scalar concentration. Only available if the [scalar transport equation](#) has been enabled.

Filter

Shows the filtered static pressure. Only available if the [static pressure filters](#) have been defined prior run.

7.4 Import/Export post-processing setup

It is sometimes convenient to prepare a post-processing setup and layout once, and apply it on several simulations to compare them in a consistent way. XFlow allows to save a given post-processing setup and layout into an .xfpp file (XFlow Post-Processing file), and to load this file from any project.

Import post-processing setup

Main menu > Post-processing > Import post-processing setup

Imports an existing post-processing setup and layout from an .xfpp file and its associated .lay file. A window will open to browse the .xfpp file of your choice and will apply the same layout and post-processing setup to the current project. The post-processing tree, cameras, fields ranges, and visualization checkbox will be loaded from the .xfpp file.

Export post-processing setup

Main menu > Post-processing > Export post-processing setup

Exports the current post-processing setup to an .xfpp file and the current layout to a .lay file. The current post-processing tree, cameras, fields ranges, and visualization checkbox is saved into the .xfpp file, and the GUI layout is saved in the .lay file.

7.5 Animation

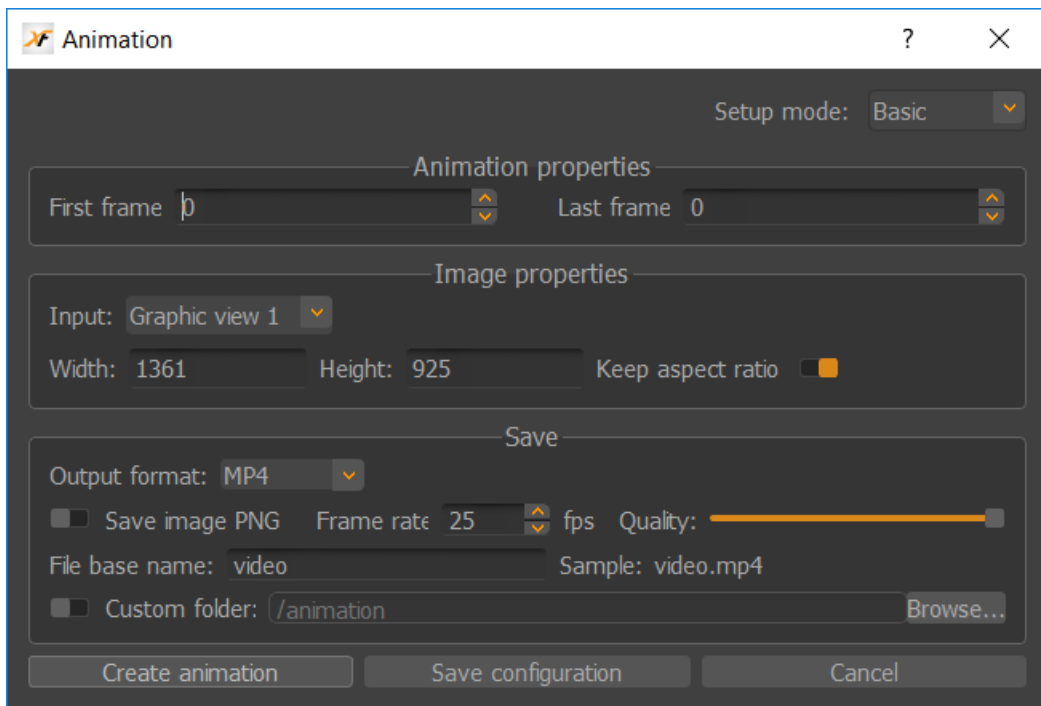
XFlow provides a wizard that allow the user to save a sequence of images, that can be later assembled in a video file using an external software. This wizard is accessible from **Main menu > Post-Processing >**

Animation, or from the button  in the [Toolbar Data Processing](#).

Animation can be [Basic](#), [Advanced](#) (Setup mode) or using [Key frame mode](#)

7.5.1 Basic animation

The Basic animation wizard is shown in the figure below; it consists of a simple interface that allows the user to easily setup animations in time. XFlow generates sequence of images that can be used to create a video or animation.



Basic Animation window

The sections of this wizard are explained below.

Animation properties

- **First frame:** first frame in the sequence
- **Last frame:** last frame in the sequence

Image properties

- **Input:** select [GUI view](#) to save the entire screen, or [Graphic view #](#) to save only the *Graphic View #* window.
- **Width:** Image width
- **Height:** Image height
- **Hold aspect ratio**

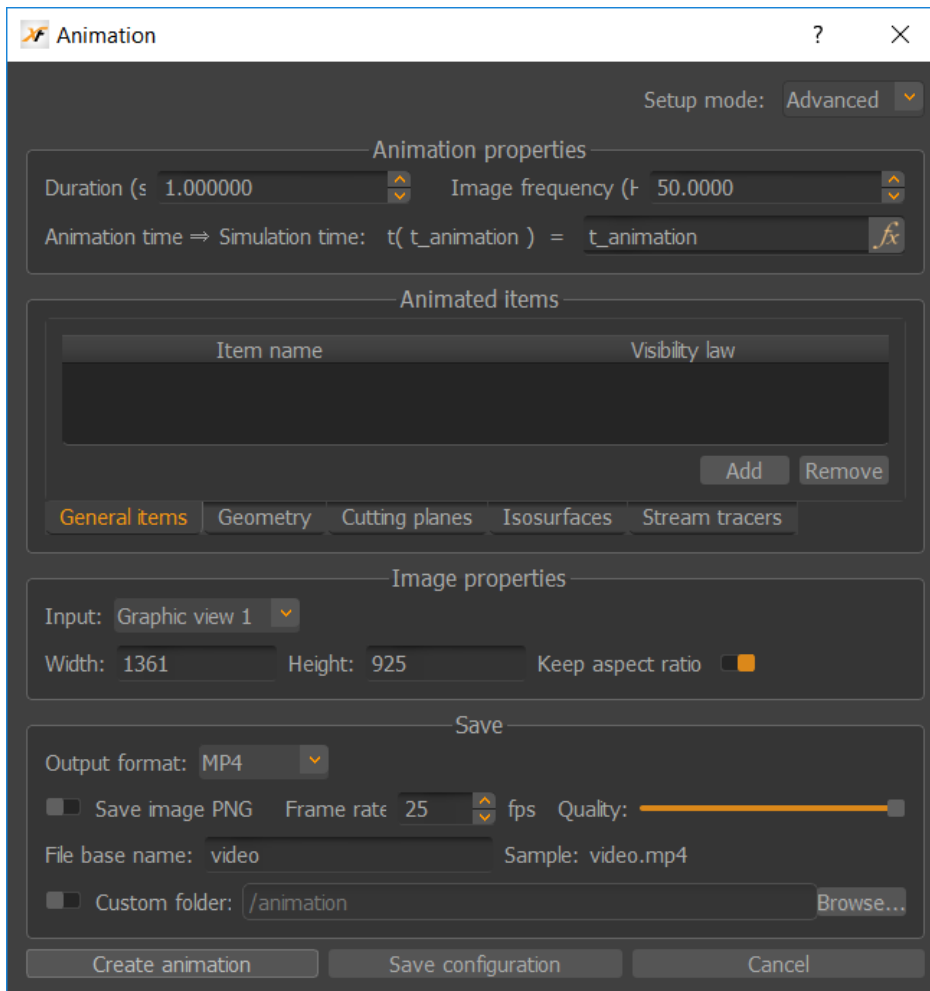
Save

- **Output format:** Output format can be either video (.ovg, .avi, .mp4, .png) or a sequence of pictures (.png). If a video format is selected, an additional option **Save image PNG** shows in order to save the sequence of images in .png format in addition to the video.
- **Frame rate:** Defines the frame rate of the output video. Only available for .ovg, .avi, and .mp4.
- **Quality:** Defines quality of the output video. The higher the quality, the higher the video file size. The lower the quality, the lower the video file size. Only available for .ovg, .avi, and .mp4.
- **File base name:** Images will be named after this name followed by the frame number.
- **Custom Folder:** If disabled, the animation file(s) (images and/or video) will be created in the default folder: [simulation_folder]/animation. The simulation folder is named after the project. If the option is enabled, a custom folder can be defined.

Please note: In Basic mode, the filename of the images sequence will be numbered by frame number.

7.5.2 Advanced animation

The Advanced animation wizard is shown in the figure below. This is a versatile tool that allows the user to setup animations not only in time but also (or just) in space, showing and hiding geometry objects, etc.



Advanced Animation window

The customisation parameters of this wizard are explained below.

Animation properties

- **Duration:** Duration of the video (real time).
- **Image frequency:** Images per second to build the video.
- **Simulation time:** This parameter indicates the interval of the [simulation time](#) that the user wants to show in the animation. It can be:
 - A number indicating a fixed simulation time (simulation frame). The animation data will be frozen in time.
 - A function depending on the animation time ($t_{\text{animation}} = \text{No.ImagesSaved} / \text{frequency}$).

Example: Advanced animation I

Simulation > Simulation time: 20 s
 Animation (Advanced) > Duration: 1 s
 Animation (Advanced) > Images Frequency: 50 Hz
 Animation (Advanced) > t(t_animation): 20



The animation will generate 50 images, all of them showing the data of the the last frame of the simulation.

Example: Advanced animation II

Simulation > Simulation time: 10 s
 Animation (Advanced) > Duration: 1 s
 Animation (Advanced) > Images Frequency: 50 Hz
 Animation (Advanced) > t(t_animation): t_animation



The animation will generate 50 images that show the simulation data every 0,02 seconds up to 1 second. Simulation data from time 1 to 10 are not represented in the animation.

Example: Advanced animation III

Simulation > Simulation time: 10 s
 Animation (Advanced) > Duration: 1 s
 Animation (Advanced) > Images Frequency: 50 Hz
 Animation (Advanced) > t(t_animation): 5+t_animation



The animation will generate 50 images that show the simulation data every 0,02 seconds in the interval [5,6] seconds of the simulation time. Simulation data corresponding to times [0-5) and (6,10] are not shown in the animation

Animated items

In this section, the behaviour of the objects can be defined. These can be of the following type:

- **General items:** The general post-processing options, namely: [domain structure](#), [volumetric field](#), [surface info](#), [isocontours](#), can be shown/hidden during the animation according to the given Visibility law, which is a boolean switch. Hence, the general item will be shown if the visibility law is one, and it will be hidden when it is zero.
- **Geometry:** [Geometry objects](#) can also be shown and hidden according to a user-defined Visibility law. If the visibility law equals 1 the object is shown, if it equals 0 it is not shown.
- **Cutting planes:** [Cutting planes](#) (previously created) can be moved within the spatial domain (Position law) besides being shown/hidden (Visibility law). The Position law must take values ranging from 0 to 1, e.g. for a cutting plane normal to the Z-axis, Position law = 0 stands for the -Z boundary and Position law = 1 represents the +Z boundary. Visibility law is of boolean type, so it should take value 0 or 1. By default, it is set to 1 (visible).
- **Isosurfaces:** [Isosurfaces](#) (if previously created) can be visualized in the simulation according to the given boolean Visibility law. Moreover, the level of the isosurface being represented can be defined as a function in Level law.
- **Stream tracers:** [Stream tracers](#) (if previously created) can be visualized in the simulation according to the given boolean Visibility law.

Example:**Position law:** $t_animation/Duration$

→

By definition, $t_animation$ ranges from 0 to duration (See animation properties). The cutting plane will thus move from 0 to 1, scanning the whole spatial domain.

Position law:
 $tabulardatalinearinterpolated(t_animation, \rightarrow$
 "CuttingPlaneAnimationTest.txt")

The cutting plane will move according to the tabular data given in the file "CuttingPlaneAnimationTest.txt", where the first column contains the $t_animation$ value and the second column the corresponding position of the cutting plane.

where, CuttingPlaneAnimationTest.txt":

0	0.5
2	0.5
6	0.0
10	0.5
14	1.0
18	0.5
20	0.5

The cutting plane movement (e.g. normal to Z axis) is thus:

- $t_animation$ in $[0,2]$, it remains at the middle of the Z-domain.
- $t_animation$ in $(2,6)$, it moves towards the -Z boundary.
- $t_animation = 6s$ it is in the -Z boundary.
- $t_animation$ in $(6, 14)$, it keeps moving to +Z-boundary, passing through the middle of the domain at $t_animation = 10s$.
- $t_animation = 14s$, it is placed at +Z-boundary.
- $t_animation$ in $(14, 18)$, it moves back to the initial position, where it remains from $t_animation = 18$ to $t_animation = 20s$.

Example:

Animation (Advanced) > Duration: 1 s

Visibility law:

→

By default the visibility law is set to 1. If the user leaves it blank, the animated item will be shown during the whole animation.

Visibility law: $t_animation < 0.5$

→

The animated item will be shown during the first half of the animation, then it will be hidden.

Example:**Level law:** 0.1

→

The isosurface will represent the points of the domain which value is 10% of the maximum value of the variable (in the domain).

Image properties

- **Input:** select [GUI view](#) to save the entire screen, or [Graphic view #](#) to save only the Graphic View # window.
- **Width:** Image width
- **Height:** Image height
- **Hold aspect ratio**

Save

- **Output format:** Output format can be either video (.ovg, .avi, .mp4, .png) or a sequence of pictures (.png). If a video format is selected, an additional option **Save image PNG** shows in order to save the sequence of images in .png format in addition to the video.
- **Frame rate:** Defines the frame rate of the output video. Only available for .ovg, .avi, and .mp4.
- **Quality:** Defines quality of the output video. The higher the quality, the higher the video file size. The

lower the quality, the lower the video file size. Only available for .ovg, .avi, and .mp4.

- **File base name:** Images will be named after this name followed by the frame number.
- **Custom Folder:** If disabled, the animation file(s) (images and/or video) will be created in the default folder: [simulation_folder]/animation. The simulation folder is named after the project. If the option is enabled, a custom folder can be defined.

Please note: In Advanced mode, the filename of the images sequence will be numbered by image number.

7.5.3 Keyframes animation

This is a user-friendly tool that allows the user to setup animations interpolate positions and switch visibility:

- **Position interpolation:** the camera position can be interpolated linearly between different frames, as well as the cutting plane position.
- **Visibility:** the visibility of any item in the [Post-Processing](#) tree can be switched on or off at the desired frame.

The workflow to use keyframes is as follows:

1. **Enter key frame mode:** Select **Main menu > Post-Processing > Animation > Edit keyframe mode**, or press a few second the icon , or press **Alt+k shortcut**.

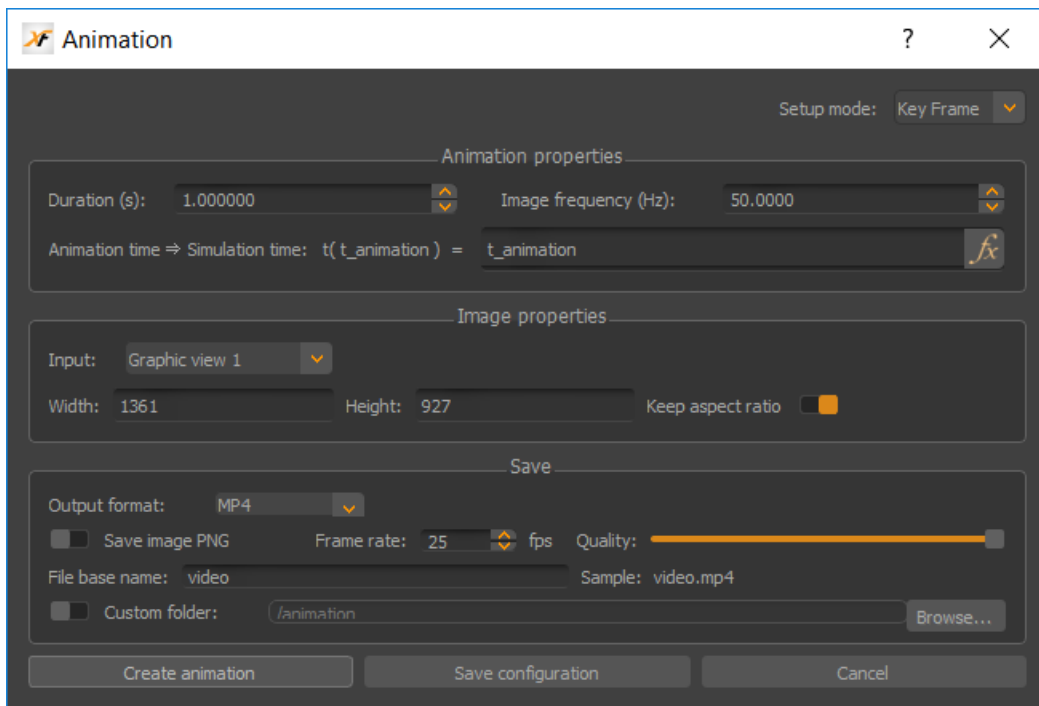
Please note: The project cannot be saved in keyframes edit mode. In order to save the project, please leave the keyframes edit mode. The keyframes are saved automatically.

2. **Create a key frame:** Set the entities visibility and positions at the desired frame and select **Main menu > Post-Processing > Animation > Create/Delete keyframe**, or use  toolbar icon. This will create a key frame at the current frame. Alternatively, you can click the **middle mouse button** in the [timeline](#) on the frame where the key frame must be created. This will create a keyframe which will be marked by a vertical yellow line.



This process can be repeated for several entities visibility and positions at different frames. Once generated, the animation will interpolate linearly the camera and/or cutting planes positions and switch on or off the visibility of the different items.

3. **Create animation:** Open the [Animation](#) dialog, and select the **Setup mode: Key Frame** on the top-right corner of the window:



Key Frame Animation window

The parameters of this dialog are explained below.

Animation properties

- **Duration:** Duration of the video (real time).
- **Image frequency:** Images per second to build the video.
- **Simulation time:** This parameter indicates the interval of the [simulation time](#) that the user wants to show in the animation. It can be:
 - A number indicating a fixed simulation time (simulation frame). The animation data will be frozen in time.
 - A function depending on the animation time ($t_{\text{animation}} = \text{No.ImagesSaved} / \text{frequency}$).

Image properties

- **Input:** select [GUI view](#) to save the entire screen, or [Graphic view #](#) to save only the *Graphic View #* window.
- **Width:** Image width
- **Height:** Image height
- **Keep aspect ratio**

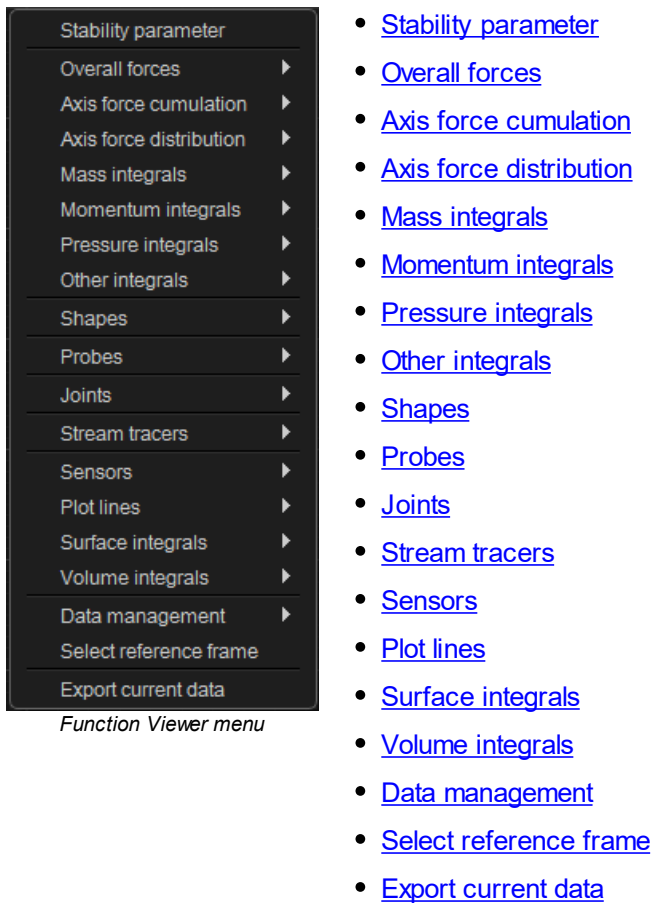
Save

- **Output format:** Output format can be either video (.ovg, .avi, .mp4, .png) or a sequence of pictures (.png). If a video format is selected, an additional option **Save image PNG** shows in order to save the sequence of images in .png format in addition to the video.
- **Frame rate:** Defines the frame rate of the output video. Only available for .ovg, .avi, and .mp4.
- **Quality:** Defines quality of the output video. The higher the quality, the higher the video file size. The lower the quality, the lower the video file size. Only available for .ovg, .avi, and .mp4.
- **File base name:** Images will be named after this name followed by the frame number.
- **Custom Folder:** If disabled, the animation file(s) (images and/or video) will be created in the default folder: [simulation_folder]/animation. The simulation folder is named after the project. If the option is enabled, a custom folder can be defined.

Please note: It is only possible to use geometries, cutting planes and camera movements in this mode.

7.6 Function Viewer

The numerical post-processing mostly happens via the *Function Viewer*. If simulation data are loaded and the user **right clicks** on the *Function Viewer* window, then a contextual menu, like the one shown in the figure below, will pop up. This menu shows the survey of results that can be visualized in the *Function Viewer*:



Function Viewer menu

Stability parameter

It displays the values of the stability parameter of the computation (see [Message View](#)).

Overall forces

It displays the overall forces (F_x , F_y , F_z), in global axes, exerted by the fluid on the entire geometry model and the corresponding force coefficients (C_x , C_y , C_z):

$$C_x = \frac{2 F_x}{\rho V_{ref}^2 A_{ref}}, \quad C_y = \frac{2 F_y}{\rho V_{ref}^2 A_{ref}}, \quad C_z = \frac{2 F_z}{\rho V_{ref}^2 A_{ref}}$$

V_{ref} and A_{ref} being the reference velocity and area specified in the [Environment](#) settings.

⚠ Please note: The user has to be cautious with the values of the reference velocity and area used in order to get correct aerodynamic coefficients. Check the [Reference area and velocity](#) settings.

Axis force distribution

The force in the X direction on the lattice slice Δx located at x can be expressed as:

f_X , Volumetric force in X direction [$N.m^{-3}$]

$$\Delta F_X(x, t) = \int_x^{x+\Delta x} \int_{y_{\min}}^{y_{\max}} \int_{z_{\min}}^{z_{\max}} f_X(x, y, z, t) dx dy dz$$

This instantaneous force is time averaged starting from initial averaging time t_{init} :

$$\Delta \bar{F}_X(x, t) = \frac{1}{t - t_{\text{init}}} \int_{t_{\text{init}}}^t \Delta F_X(x, \tau) d\tau$$

The axis force distribution graph is therefore:

$$\Delta \bar{F}_X(x, T)$$

where T corresponds to the total computed simulation time.

The force distributions are available for the three force components F_X , F_Y , and F_Z as well as for the three directions X , Y , and Z .

⚠ Please note: the X -axis range is between 0 and the maximum length of the considered direction.

Axis force cumulation

According to the above axis force distribution explanations, the axis force cumulation graph is:

$$\Delta \hat{F}_X(x, t) = \frac{\sum_x \Delta \bar{F}_X(x, T)}{\sum_{x_{\min}}^{x_{\max}} \Delta \bar{F}_X(x, T)}$$

The force cumulations are available for the three force components F_X , F_Y , and F_Z as well as for the three directions X , Y , and Z .

Mass integrals

Inlet mass flow $\int_{A_{\text{inlet}}} \rho V_n dA$

Outlet mass flow $\int_{A_{\text{outlet}}} \rho V_n dA$

Overall mass $\int_{\Omega} \rho d\Omega$

where V_n is the normal velocity to the surfaces A_{inlet} and A_{outlet}

Momentum integrals

$$\text{Inlet X-momentum flux} \quad \int_{A_{inlet}} \rho V_x V_n \, dA$$

$$\text{Inlet Y-momentum flux} \quad \int_{A_{inlet}} \rho V_y V_n \, dA$$

$$\text{Inlet Z-momentum flux} \quad \int_{A_{inlet}} \rho V_z V_n \, dA$$

$$\text{Outlet X-momentum flux} \quad \int_{A_{outlet}} \rho V_x V_n \, dA$$

$$\text{Outlet Y-momentum flux} \quad \int_{A_{outlet}} \rho V_y V_n \, dA$$

$$\text{Outlet Z-momentum flux} \quad \int_{A_{outlet}} \rho V_z V_n \, dA$$

 **Please note:** Here, (X, Y, Z) stand for the global coordinates

Pressure integrals

$$\text{Inlet total pressure flux} \quad \int_{A_{inlet}} (p_{static} + p_{dyn}) V_n \, dA$$

$$\text{Inlet static pressure flux} \quad \int_{A_{inlet}} p_{static} V_n \, dA$$

$$\text{Inlet dynamic pressure flux} \quad \int_{A_{inlet}} \frac{1}{2} \rho V^2 V_n \, dA$$

$$\text{Outlet total pressure flux} \quad \int_{A_{outlet}} (p_{static} + p_{dyn}) V_n \, dA$$

$$\text{Outlet static pressure flux} \quad \int_{A_{outlet}} p_{static} V_n \, dA$$

$$\text{Outlet dynamic pressure flux} \quad \int_{A_{outlet}} \frac{1}{2} \rho V^2 V_n \, dA$$

Other integrals

$$\text{Inlet enstrophy flux} \quad \int_{A_{inlet}} \frac{1}{2} \rho \omega^2 V_n dA$$

$$\text{Outlet enstrophy flux} \quad \int_{A_{outlet}} \frac{1}{2} \rho \omega^2 V_n dA$$

$$\text{Overall enstrophy} \quad \int_{\Omega} \frac{1}{2} \rho \omega^2 d\Omega$$

$$\text{Overall kinetic energy} \quad \int_{\Omega} \frac{1}{2} \rho V^2 d\Omega$$

where ω is the vorticity of the flow.

Shapes

For every [Shape](#), the user can plot in the *Function Viewer*:

- Aerodynamic coefficients (Cx, Cy, Cz) in global axes.
- Forces (Fx, Fy, Fz) in global axes.
- Moments around the [centre of rotation](#) (Mx, My, Mz) in global axes.

If the **Flow model** is set to **Segregated Energy**, the user can further plot the heat transfer from/to the shape:

- Heat: this is the heat transferred from/to the shape where [conjugate heat transfer](#) or [wall](#) boundary condition has been specified.

If the **Flow model** is set to **Multiphase**, the user can further plot the different forces and moments contributions:

- Overall: this is the total force/moment of Fluid 1 + Fluid 2.
- Fluid 1: this is the force/moment contribution of Fluid 1 only.
- Fluid 2: this is the force/moment contribution of Fluid 2 only.

If the **Shape** behaviour is either **Enforced** or **Rigid body dynamics**, the user can further plot:

- Position (Px, Py, Pz) of the centre of gravity in global X, Y and Z directions.
- Euler angles (Eux, Euy, Euz) around the centre of gravity.
- Linear velocity (Vx, Vy, Vz) of the centre of gravity in global X, Y and Z directions.
- Angular velocity (Wx, Wy, Wz) around the centre of gravity in global X, Y and Z directions.

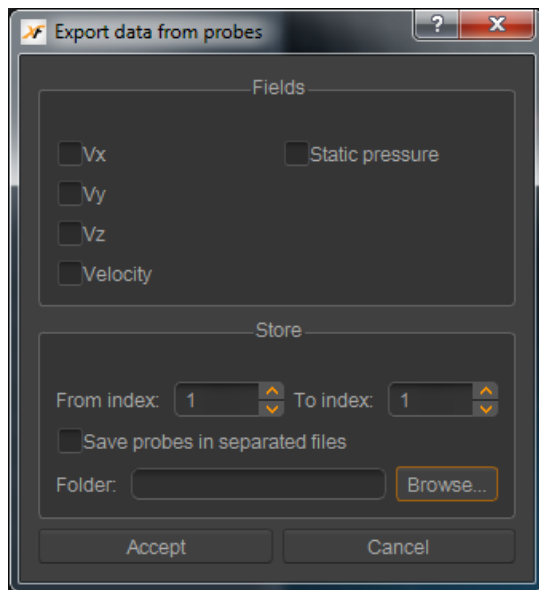
Probes

Displays a list of all [Probes](#) in the simulation and the flow variables monitored in each probe:

- Static pressure.
- Velocity module.
- X, Y and Z components of the velocity (Vx, Vy, Vz).
- Volume of liquid phase (for Free-Surface/multi-phase simulations)
- Temperature (for thermal applications)

The "Export all" function allows to customize the probe data exportation. The following parameters of exportation are available:

- Fields: select the fields to export.
- From index: select the range of probes to export. To export a single probe N, select from index N to index N.
- Save probes in separated files: If checked, each probe will be saved in a separated file named "probe-N.txt". If not, all probes will be included in the same data file in a column format: Time Probe1_Field1 Probe1_Field2 Probe2_Field1 Probe2_Field2 ...
- Folder: indicate the folder where to export the file(s).



Joints

Displays a list of all [Joints](#) in the simulation and the associated efforts and positions:

- Position coordinates of the joint (Px, Py, Pz) in global axes.
- Forces applied on the joint by other geometries/cables (Fx, Fy, Fz) in global axes.
- Moments applied around the joint position by other geometries/cables (Mx, My, Mz) in global axes.

Stream tracers

Displays a list of all calculated [Stream tracers](#). It shows:

- **Inlet particles distribution** It shows a histogram of the particles generated from the [source shape](#) in function of their diameters. It is only calculated if the diameter Standard deviation is not 0.
- **Outlet particles flux** The total number of particles that have exited through the outlets until the time of measurement.
- **Outlet particles distribution** It plots a histogram of the particles leaving the domain from the outlet in function of their diameters. It is only available if the diameter Standard deviation is not 0.
- **Active particles** The evolution of number of the particles inside the fluid domain.
- **Control geometries:**
 - **<Shape name>** It shows the evolution of the active particles inside the [Control geometry](#).
 - **<Shape name> distribution** It shows the diameter distribution of the active particles inside the [Control geometry](#). It is only calculated if the diameter Standard deviation is not 0.

Sensors

Displays a list of all [Sensors](#). It shows the value of the chosen **field** at the sensor.

Plot lines

Displays a list of all [Plot lines](#). It shows the value of the chosen **field** at the line.

Surface integrals

Displays a list of all [Surface integrals](#). It shows the integrated value of a **field** over a surface.

Volume integrals

Displays a list of [Volume integrals](#). It shows the integrated value of a **field** within a volume.

Data management

The user can manage the graph representation:

- **Set graph to [Time space] mode:** sets graph as a function of time.
- **Set graph to [Frequency space] mode:** sets graph as a function of frequency (Fourier Transform) with the possibility to apply a [window function](#) on the signal.
- **Set graph to [SPL vs freq] mode:** plots the [Sound Pressure Level](#) (SPL) in the frequency space with the possibility to apply a [window function](#) on the signal.
- **Set graph to [PSD vs freq] mode:** plots the [Power Spectral Density](#) (PSD) in the frequency space with the possibility to apply a [window function](#) on the signal.
- **Apply filter to signal:** applies a [Filter to the signal](#) displayed in the Function Viewer with the possibility to apply a [window function](#) on the signal.

Select reference frame

The user can display the numerical values in one of these reference frames:

- Global reference frame: to plot the graphics in the XFlow global axes.
- Shape reference frame: to plot the graphics in the [local axes](#) of one of the [shapes](#).
- Custom reference frame: to plot the graphics in custom [reference frames](#) entities. See this [section](#) to learn how to create a custom reference frame.

By default the graphics displayed in the *Function Viewer* are represented in the XFlow global reference.

⚠ Please note: The "[Export current data](#)" option must be used if the data wants to be exported in a different reference frame than the global one. The "[Export all](#)" function will always export the data referred to the global reference frame.

⚠ Please note: During post-processing, the change of reference frame does not admit changes of entity names.

Export numerical data

Right click in Function Viewer > Export current data exports the data currently displayed in this window.

Right click in Function Viewer > ... > Export all exports all the data available in the selected section (Overall forces, Mass integrals, Momentum integrals...). It has the following format:

Time	Field_1	Field_2	...	Field_M
...				
Value_Time	Value_Field_1	Value_Field_2	...	Value_Field_M

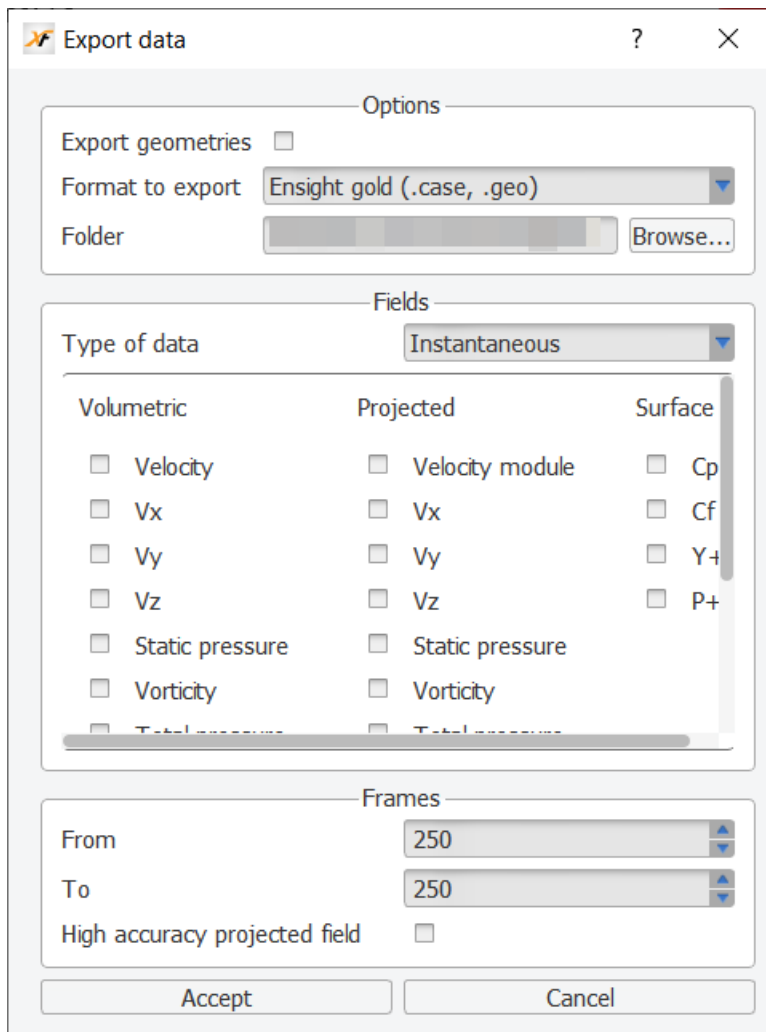
⚠ Please note: To learn how to create and manage a *Function Viewer*, see [GUI-Function Viewer](#).

7.7 Export data

Export simulation data

Main menu > Simulation Data > Export data

Allows the user to export simulation data to an external format.



Export Data Window

Options

Export geometries: exports the geometries participating to the simulation.

Format to export:

- ✎ **Paraview:** exports volumetric, projected and surface data to Paraview format. ParaView is an open source multiple-platform application for interactive, scientific visualization.
- ✎ **Enight Gold:** exports volumetric, projected and surface data to Enight Gold format. Enight Gold is a post-processing software developed by CEI Software.
- ✎ **CGNS:** exports volumetric, projected and surface data to CGNS format. CGNS stands for CFD General Notation System. It is a general, portable, and extensible standard for the storage and retrieval of CFD analysis data. It consists of a collection of conventions, and free and open software implementing those conventions.
- ✎ **ABAQUS surface format:** exports the projected static pressure on all geometries to ABAQUS format. It is useful to proceed in Finite Element Analysis (FEA) analysis from the pressure map obtained. (*Only available in Labs mode.*)
- ✎ **XFlow ASCII format:** exports the volumetric data and/or surface data from all geometries at vertices position providing the connectivity matrix. The surface data exported will be consistent to the **Interpolation mode** selected in the GUI ([+info](#)), and the volumetric data are exported at

every lattice position. The generated files have the following format:

```
LATTICE
-----

FIELDS=Field_1, ..., Field_M

X1 Y1 Z1 Value_Field_1 ... Value_Field_M
...
XN YN ZN Value_Field_1 ... Value_Field_M

GEOMETRIES
-----

SHAPE_NAME=ShapeName
NUM_VERTICES=N (Number of geometry vertices)
NUM_TRIANGLES=T (Number of geometry triangles)
FIELDS=Field_1, ..., Field_M

X1 Y1 Z1 Value_Field_1 ... Value_Field_M
...
XN YN ZN Value_Field_1 ... Value_Field_M
INDEX_I1 INDEX_J1 INDEX_K1
...
INDEX_IT INDEX_JT INDEX_KT
```

where INDEX_I/J/K_i are the first/second/third vertices of ith triangle. INDEX are integers between [0; N-1], referring the vertex line number.

Example:

```
GEOMETRIES
-----

SHAPE_NAME=Box
NUM_VERTICES=24
NUM_TRIANGLES=12
FIELDS=VEL,SP

0 0 0 5.48172 -8.50565
0 0 1 5.05112 0.434451
...
1 1 1 5.26072 -4.59222
1 1 1 5.1933 -4.91023
2 0 1
3 2 1
...
23 15 11
23 11 19
```

Folder: folder where data files are saved.

Type of data

Data can be exported as instantaneous, averaged, RMS and standard deviation type. The averaged and standard deviation data type must be enabled before the computation in the [Simulation tab](#). If the scalar transport is enabled (see [scalar](#)) it is possible to export the scalars concentration using the CGNS export data format.

Volumetric / Projected / Surface

These are the fields that can be exported. They are divided in three columns: volumetric fields, projected fields and surface fields. The volumetric fields are defined within the whole region of fluid, the project fields are the lattice nodes neighbors to the walls and projected over the geometries, and the surface fields are the fields only defined at geometry surfaces.

Frames

The simulation data can be exported for each [frame](#) saved by XFlow. This field specifies the first and last frame to be exported , e.g. from frame 0 to frame 10.

High accuracy projected field

Only available for Ensignt gold format, this exporter searches with a higher precision the neighbor cells intersecting with the geometry at the expense of higher export time.

 **Tip:** This data export can be executed in command lines as well ([+Info](#)).

Main menu > Simulation Data > Export plot line data

Allows the user to export the **Plot line** data automatically for different frames in .txt file format. Each plot line will be exported into a different file (plot_line_#_history.txt where # is the plot line number), containing the field values, the coordinates of the discretization points along the line, and the simulation time corresponding to each frame.

 **Please note:** The Auto-refresh option should be enabled when exporting **Plot lines** for different frames ([+Info](#)).

Main menu > Simulation Data > Export surface data

Allows the user to export the **Surface info** field applied on the selected geometries and at the selected frame, saved into a file named by default `shape[name] frame[number].srfi`.

The file structure contains four columns with the following information on every computational element at the surface: position X, position Y, position Z and value of the selected **Surface info**. An example of file structure, exporting the temperature field on the geometry named "Box" from frame 48 is the following:

Example:

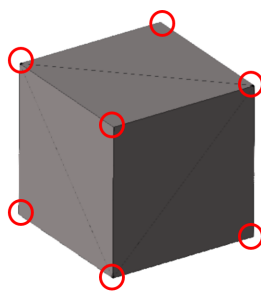
```
Box
48

x y z Temperature
-0.975000 -0.975000 -0.975000 458.015385
-0.975000 -0.875000 -0.975000 418.679619
-0.975000 -0.925000 -0.975000 433.235283
-0.975000 -0.925000 -0.925000 407.786074
...
-0.925000 -0.925000 -0.975000 294.127870
-0.925000 -0.875000 -0.975000 291.732359
-0.875000 -0.875000 -0.975000 307.031493
-0.875000 -0.975000 -0.975000 312.930360
```

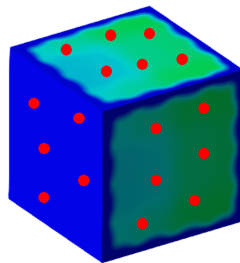
The number of exported points depends on the option **Export data on** selected:

- ↳ **Geometry vertices**: exports data on the geometry vertices, i.e. the mesh nodes of the geometry tessellation. Therefore, one must be careful of the geometry tessellation if using this option to export surface data, as a uniform mesh distribution would be recommended in order to export the data adequately.
- ↳ **Point distribution**: exports data on a random point distribution generated over the geometry. The number of points is defined by the "Sample points" option set, the higher the number of sample points the more accurate will be the distribution but the more points will be saved in the `shape [name] frame [number].srifi` file.
- ↳ **Near-wall lattice nodes**: exports data from the near-wall lattice nodes, i.e. the first fluid layer near the walls. This means the data exported is not defined rigorously on the geometry surface and will depend on your lattice size and distribution.

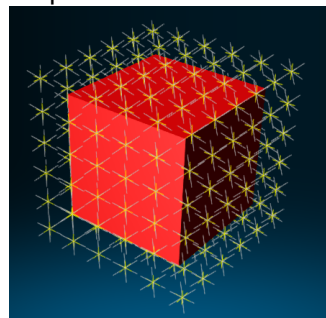
The following pictures illustrate the location of the data exported for each of the above options.



Geometry vertices



Point distribution



Near-wall lattice nodes

⚠ Please note: No connectivity matrix is provided using the **Export surface data** option. In order to export data with connectivity matrix, please use the **Export data > XFlow surface data** option describe in this section above.

Main menu > Simulation Data > Export cutting plane data to raw format

Exports the data of all the active cutting planes of the active *Graphic View* (without interpolation) to a different file for each cutting plane exported, by default named `currentCuttingPlane#.txt`. It contains four columns with the following information on every computational element at the cutting plane: position X, position Y, position Z and field value.

Main menu > Simulation Data > Export data of cutting plane field distribution

If one or many cutting planes with a [field distribution](#) are active, this saves the field distribution (coordinates and field value) in four files (one corresponding to each face) for each cutting plane field distribution: `field_distribution_#_minusX.txt` for the face oriented towards -X, `field_distribution_#_plusX.txt` for the face oriented towards +X, `field_distribution_#_minusY.txt` for the face oriented towards -Y, and `field_distribution_#_plusY.txt` for the face oriented towards +Y. If the cutting plane is not axis aligned, the filename becomes `field_distribution_#_+i_+j_+k.txt` where (i,j,k) is the normalized vector direction.

Main menu > Simulation Data > Export selected isosurfaces

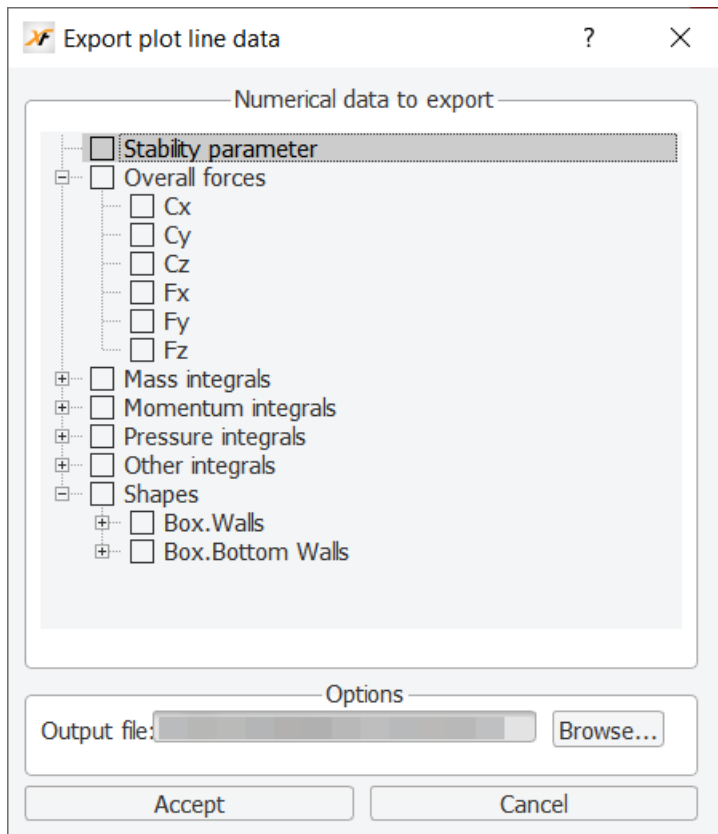
To export the selected isosurfaces to a geometry file format. Available formats: .stl, .nff, .nfb

Main menu > Simulation Data > Export numerical data

To export the numerical data computed by the engine during the simulation. This data can be plotted in the Function Viewer, and can be exported into an ASCII text file via this menu. Check the curves to be exported within the list and a file named `outputFile.txt` will be generated in the selected folder.

The output format is such as:

```
Time Sim1_Field1 Sim2_Field1
...
Time Sim1_Field2 Sim2_Field2
...
```



Export Numerical Data Window



Tip: The numerical data export can be executed in command lines as well ([+Info](#)).

8 Co-Simulation

XFlow is capable to perform coupled simulations with:

- **ADAMS simulation:** Import multibody dynamics computed in ADAMS to the geometry shapes in XFlow ([+info](#)).
- **FMI Standard:** Export inputs/outputs variables from XFlow in FMI Standard format ([+info](#)).
- **Abaqus:** Co-simulation between XFlow and Abaqus is implemented through the Co-Simulation Service (CSS) ([+info](#)).
- **Simpack:** Co-simulation between XFlow and Simpack is implemented through the Co-Simulation Service (CSS) ([+info](#)).
- **MSC Nastran:** Two-way coupling is implemented through the OpenFSI standard, which involves the nonlinear SOL400 solution of MSC Nastran ([+info](#)).
- **MSC Nastran:** One-way thermal coupling is implemented through the modification of an imported .bdf file ([+info](#)).

8.1 Import ADAMS simulation

In XFlow it is possible to import MSC/ADAMS® result files and apply the multibody dynamics computed in ADAMS to the geometry shapes in XFlow.

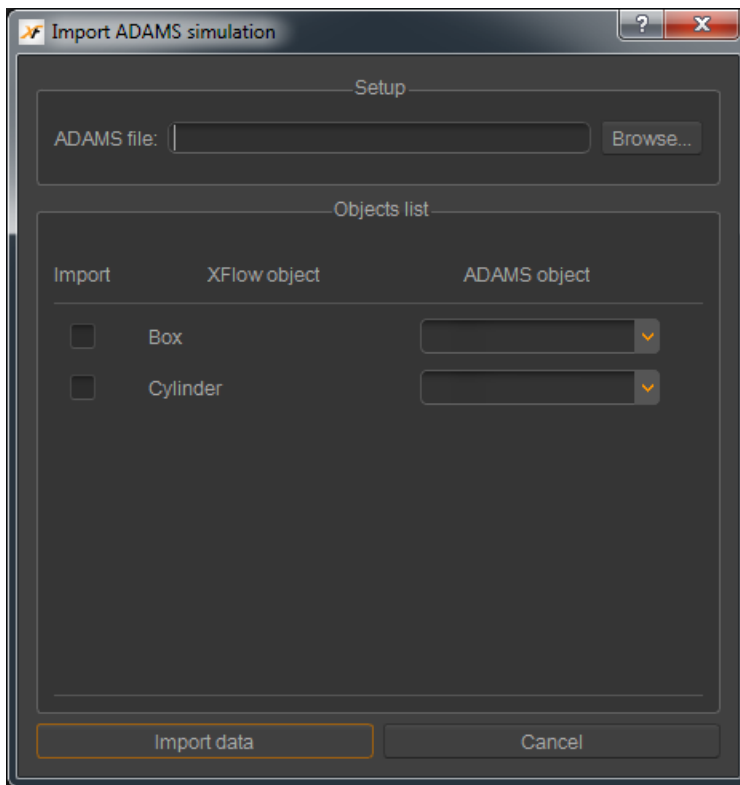
The first stage is to export the geometries and results from Adams:

- a) select the geometry one by one and export it: **File > Export > File type: STEP**
- b) export the results file: **File > Export > File type: Adams/solver results file .RES**

The second stage is in XFlow:

- c) import the STEP geometries from Adams
- d) import the Adams results: **Main menu > Options > Import ADAMS simulation**
- e) select the Adams result file and associate the XFlow shapes to Adams parts

8.1 Import ADAMS simulation



Import ADAMS simulation window

This applies the motion computed in Adams to the specified shapes.

 **Tip:** Adams motion can be applied on any arbitrary geometry shape in XFlow, it does not need to be the one modeled in Adams.

8.2 FMI Standard

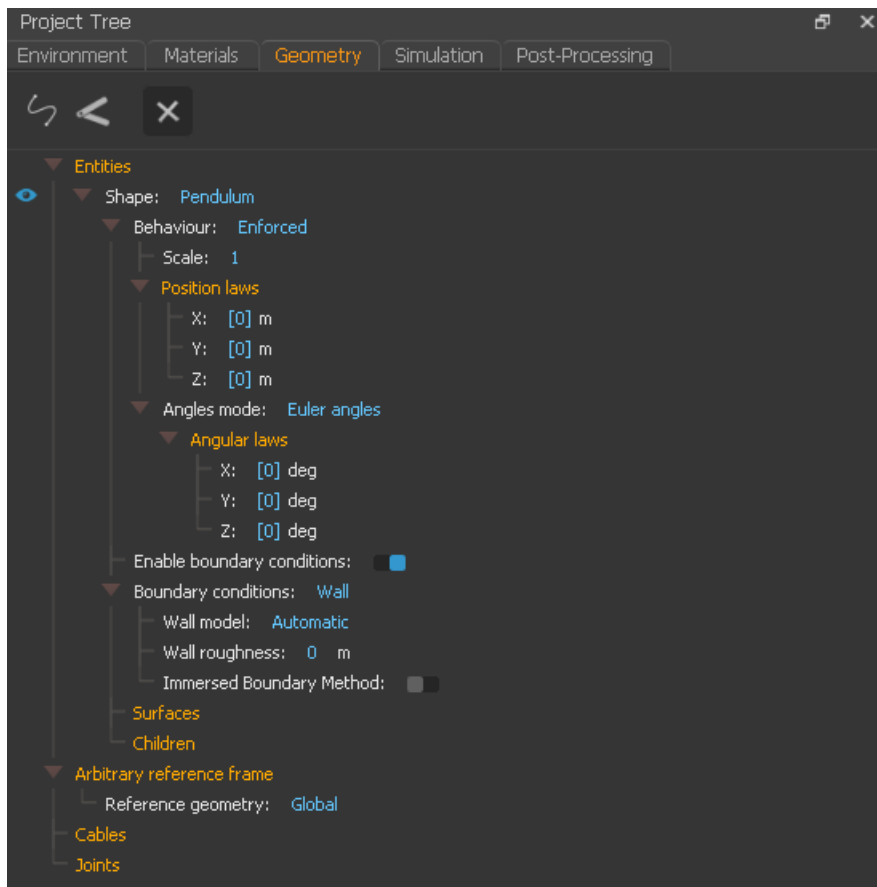
XFlow integrates Functional Mock-up Interface (FMI), a tool independent standard which supports both model exchange and co-simulation of dynamic models using a combination of xml-files and compiled C-code.

FMI Standard works with a coordinator-worker concept, the **workers** simulating sub-problems while the **coordinator** is responsible for both coordinating the overall simulation and handling the data transfer. **XFlow** can only be set as a **worker**, hence it can be connected with external software which act as a **coordinator**. For more information regarding the FMI standard please check <https://www.fmi-standard.org/tools>

 **Please note:** FMU computations are available in [Labs mode](#) only.

XFlow steps to couple with an external software through the FMI Standard:

- a) Create a geometry and set it as enforced behavior. Currently, only object's position and angular laws are available as shared variables for FMI coupled simulations.



Enforced geometry

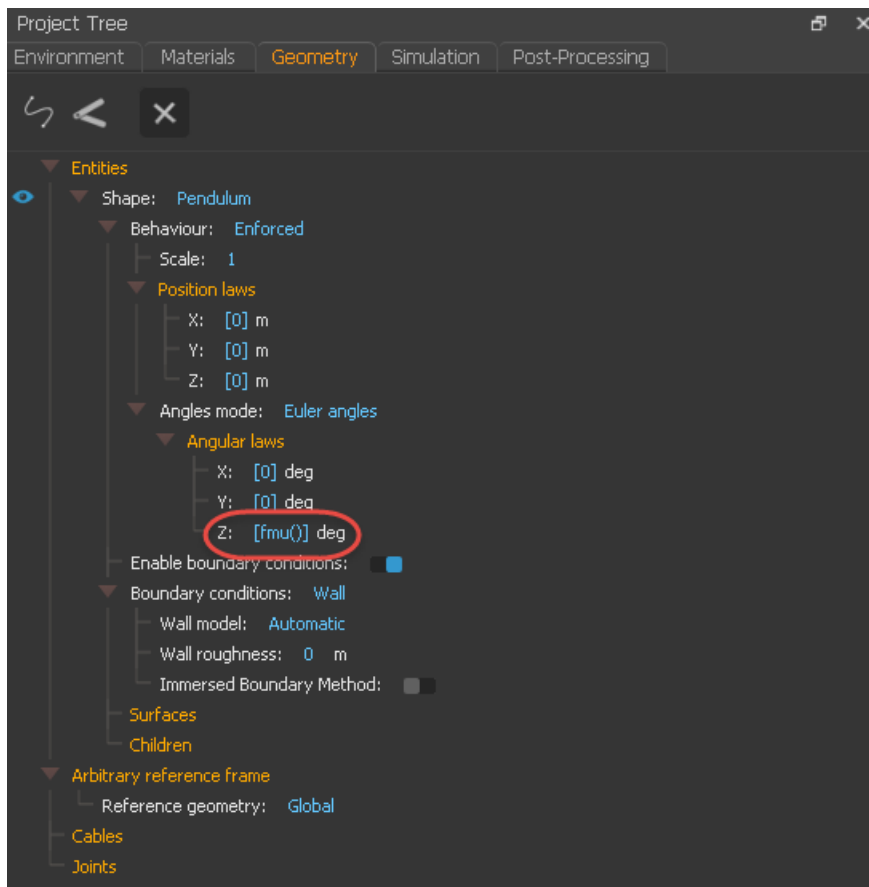
- b) Select available inputs and outputs in **Main menu > Options > Export to FMI Standard**. In the left column the available inputs and outputs for the previously created geometry are listed. The right column, instead, contains the selected FMI inputs and outputs. Any XFlow numerical data value (e.g. Time, Cx, Cy ...) can be selected as FMI outputs.

⚠ Please note: the FMU variables setup will be stored in the project file. This will ease the process of changing the setup or re-using an existing one for a different simulation.



Export to FMI Standard window

- c) When an available input has been set as FMI input variable its state changes in the Project tree, as shown in the picture below. Whenever a variable is set to `fmu()` it will be used as input for the FMI standard file.



Y Angular law as a fmu input in the enforced body

d) The option **Main menu > Options data > Export to FMI standard** will create an **xflowFMI.fmu** file in the [Simulation folder](#).

e) Configure the coordinator software using the **xflowFMI.fmu** file created by XFlow.

⚠ Please note: XFlow supports only one simulation at a time. That means, only one XFlow slave can be used in an FMU computation.

f) Run the simulation by selecting **Run > Start FMU computation**. XFlow will wait for the synchronization point of the coordinator simulation and will start to calculate.

⚠ Please note: The FMU computation is not available yet for distributed MPI computations.

8.3 CSS Co-simulation

Project Tree > Environment > Engine > Advanced Options > Co-Simulation

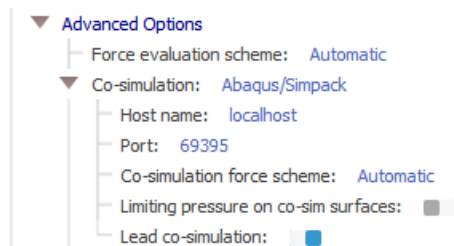
XFlow provides the possibility to perform coupled fluid-structure simulations using either an external MBS (Multi-Body Simulation) software or an external FEA (Finite Element Analysis) structural software. In particular, it is possible to setup co-simulations XFlow / Simpack or XFlow / Abaqus exploiting the CSS (Co-Simulation Service). This service enables a two-way coupling of the software through the exchange of loads from XFlow to Abaqus / Simpack, and the resultant displacements from Abaqus / Simpack to XFlow.

Please note: Co-simulation coupling is available in [Labs mode](#) only.

Please note: It is preferred to use .xfp file format over .xfz for co-simulation workflows.

Co-Simulation Options

- **Co-simulation:** Abaqus/Simpack
 - **Host name:** Host name where the co-simulation engine is run.
 - **Port:** Port used to connect to the co-simulation engine.
 - **Co-simulation force scheme:** determines the scheme to compute the forces sent to Abaqus
 - **Automatic:** Applies stress tensor for two way co-simulations, and nodal force for one way co-simulation.
 - **Stress tensor:** Computes the local fluid force based on the stress tensor to apply on the structural nodes. More robust scheme, recommended for two way co-simulations.
 - **Nodal force:** Applies local fluid force on structural nodes. More accurate scheme but less stable, recommended for one way co-simulations.
 - **Limiting pressure on co-sim surfaces:** pressure limiter for force evaluation in co-sims
 - **Lead co-sim:** specifies XFlow as the first actor in the co-sim interchange



Setting tab for Co-simulation parameters.

Co-Simulation Mode

Whenever the **Environment > Engine > Advanced Options > Co-Simulation** option is enabled (set different to **None**) the **Geometry > Shape > Co-simulation mode** drop-down menu will be enabled under each shape included in the simulation setup.

Several options are available which reflects a specific co-simulation workflow.

- **Nastran 2-way:** It allows exchange of forces from XFlow to Nastran and the exchange of displacements from Nastran to XFlow using the OpenFSI method.
- **Abaqus 2-way:** It allows exchange of forces from XFlow to Abaqus and the exchange of displacements from Abaqus to XFlow.
- **Abaqus 1-way deformations:** It allows exchange of displacements from Abaqus to XFlow but no forces are sent from XFlow to Abaqus. This option is relevant when forces from XFlow are negligible

for the Abaqus setup.

- **Abaqus 1-way forces:** It allows exchange of forces from XFlow to Abaqus but no displacements are sent from Abaqus to XFlow. This option is relevant when displacements from Abaqus are negligible for the XFlow setup.
- **Simpack Flexi 2-way:** The default value for flexible bodies. It allows exchange of forces from XFlow to Simpack and the exchange of displacements from Simpack to XFlow.
- **Simpack Flexi 1-way deformation:** It allows exchange of displacements from Simpack to XFlow but no forces are sent from XFlow to Simpack. This option is relevant when forces from XFlow are negligible for the Simpack setup.
- **Simpack Flexi 1-way forces:** It allows exchange of forces from XFlow to Simpack but no displacements are sent from Simpack to XFlow. This option is relevant when displacements from Simpack are negligible for the XFlow setup.
- **Simpack Rigid 2-way:** The default value for rigid bodies. It allows exchange of forces from XFlow to Simpack and the exchange of displacements from Simpack to XFlow.
- **Simpack Rigid kinematics:** It allows exchange of displacements from Simpack to XFlow but no forces are sent from XFlow to Simpack. This option is relevant when forces from XFlow are negligible for the Simpack setup.

The options available for selection depend on the **Co-simulation** option, shape **Behaviour** and on the steps followed to import the specific setup into XFlow (e.g. Simpack import [+info](#)). Specifically:

- **Co-simulation:** Nastran
 - **Rigid Bodies**
 - Nastran 2-way
- **Co-simulation:** Abaqus/Simpack (no Simpack setup imported [+info](#))
 - **Rigid Bodies**
 - Abaqus 2-way
 - Abaqus 1-way deformations
 - **Flexible Bodies**
 - Abaqus 1-way forces:
 - Simpack Flexi 1-way forces
- **Co-simulation:** Abaqus/Simpack (Simpack setup imported [+info](#))
 - **Rigid Bodies**
 - Simpack Rigid 2-way
 - Simpack Rigid kinematics
 - **Flexible Bodies**
 - Simpack Flexi 1-way deformation
 - Simpack Flexi 2-way

Co-simulation Orchestrator

The `cosim_orchestrator` is an utility tool provided with the XFlow installation to streamline the execution of co-simulation workflows. The only required argument is the XFlow project file (either `.xpf` or `xfz`). If no other input parameter is used it creates the co-simulation setup file `$project-cosim.xml` in the XFlow setup simulation folder.

It takes the following parameters as input:

- `--cse` , launches the CSE director distributed with the XFlow installation ([+info](#))
- `--cseconfig` , does not use the generated co-simulation setup file but an user specified one
- `--simpack` , starts the Simpack simulation (using the system defined command `simpack-slv`)
- `--simpackdir` , specify the path to the Simpack installation folder to select a specific version of Simpack (or in case the `simpack` command `simpack-slv` is not pre-defined)
- `--listenerport` , uses the specified port for the CSE execution (it overrides the port specified in the `.xpf` file)
- `--help` , displays the help

 **Tip:** The generation of the co-simulation setup file `$project-cosim.xml` can also be performed from the XFlow GUI, **Simulation Data > Generate co-simulation file** ([+info](#)).

Co-Simulation Service

The CSS (Co-Simulation Service) libraries and CSE (Co-Simulation Engine) executable are distributed with the XFlow installation, located under `$XFlow_INSTALL/modules/cse`. This version of the CSS and CSE are the one invoked by the [cosim_orchestrator](#) and guarantee compatibility with the used XFlow version.

 **Please note:** Under Linux OS, when launching the XFlow simulation in command line the path to the CSS lib must be explicitly set: `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$XFlow_INSTALL/modules/cse/lib`

8.3.1 Abaqus

In order to setup a coupled XFlow / Abaqus simulation follow the steps:

- a) Setup the structural model in Abaqus including a definition of the coupling surfaces (which should always be tagged as Surf-1)
- b) Include a dummy Interactions setting
- c) Export the structural mesh created in Abaqus in STL format (Abaqus Plug-ins)
- d) Export an Abaqus `.inp` file
- e) Manually modify the `.inp` file to include the following text under the interaction settings:

```
** Interaction: Int-1
*Co-simulation, name=Int-1, program=MULTIPHYSICS
*Co-simulation Region, import, type=SURFACE
  Surf-1, CF
*Co-simulation Region, export, type=SURFACE
  Surf-1, COORD
  Surf-1, U
  Surf-1, V
**
```

- f) Enable the Structural analysis in XFlow by selecting the **Abaqus/Simpack** option from the dropdown list in **Project Tree > Environment > Engine > Advanced Options > Co-simulation**;

- g) Set the co-simulation parameters in **Project Tree > Environment > Engine > Advanced Options > Co-simulation** ([+info](#))
- h) Import the Abaqus mesh in .stl format in XFlow ([+info](#));
- i) Select the Co-simulation mode for the imported Abaqus meshes ([+info](#));
- j) Setup the fluid simulation in XFlow according to the specific case;
- k) Modify the FSI_II_std_css.xml file if required ([+info](#));
- l) Execute the XFlow simulation, the Abaqus one and the Co-Simulation Engine ([+info](#)).

 **Please note:** It is highly recommended to use SI units when setting up the Abaqus simulation.

 **Please note:** Abaqus coupling is available in [Labs mode](#) only.

Limitations of Current Implementation

- The current XFlow / Abaqus implementation supports all XFlow solvers.
- Thermal simulations will not exchange heat/temperature loads to Abaqus. Only displacements, velocities and forces will be exchanged.
- Every geometry that is set as **Co-simulation mode: Abaqus 2-way | Abaqus 1-way deformations** must have a single boundary condition applied. This means that the [apply boundary condition to faces](#) option is not valid for such geometries.
- It is necessary to specify refinement regions in such a way that the shapes involved in the -simulation do not intersect different lattice levels during the whole duration of the simulation and it is contained in the finest refinement level.
- XFlow supports the entire set of 2D and 3D Abaqus structural element types for a fluid-structure simulation.
- XFlow 2D engine is not supported in XFlow-Abaqus co-simulation.
- Only one XFlow shape can be specified as co-simulation shape. If two or more XFlow shapes are defined as co-simulation shapes, the co-simulation will fail.

 **Please note:** Non-linear elements can also be used, however, XFlow will re-generate the geometry tessellation and split the non-linear elements into a combination of linear ones. This approach gives rise to approximation in the representation of non-linear deformation (within a single element) and can also give rise to small holes and gaps in the mesh.

Co-Simulation with XFlow and Abaqus					
		Qualified Abaqus Version			
		2018x	2019x	2020x	2021
XFlow version	2017x FP1810	✓	✓*		
	2018	✓	✓*		
	2019x	✓	✓*		
	2020	✓	✓*		
	2020x	✓	✓*		
	2021			✓	✓

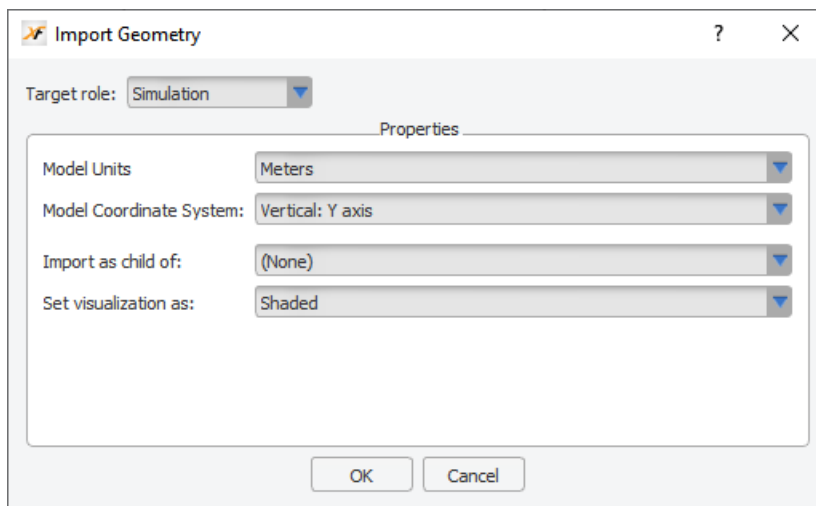
⚠ Please note: In case of running XFlow 2021 in a Windows machine, please use Abaqus 2020 HF1 and later.

⚠ Please note: In case of running Abaqus 2021, it is necessary to use CONSTANTDT as negotiation method in the FSI_II_std_css.xml file ([+info](#)).

Import Abaqus mesh

Main menu > Geometry > Import a new geometry,  or <Ctrl + i>

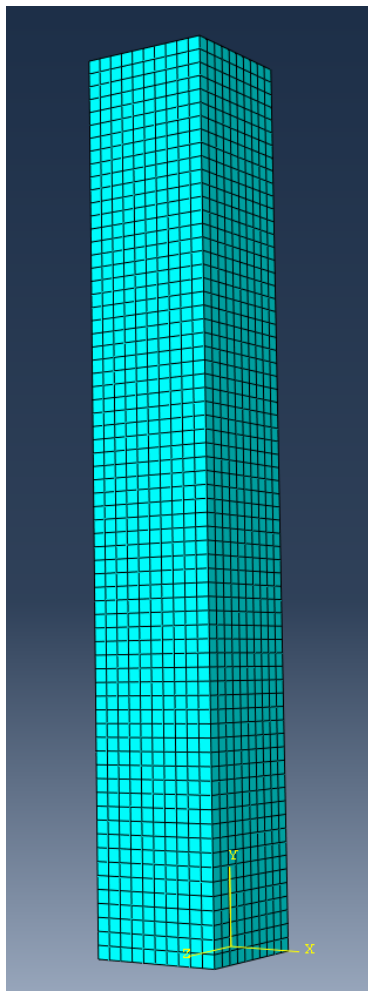
XFlow can import Abaqus mesh as .STL files:



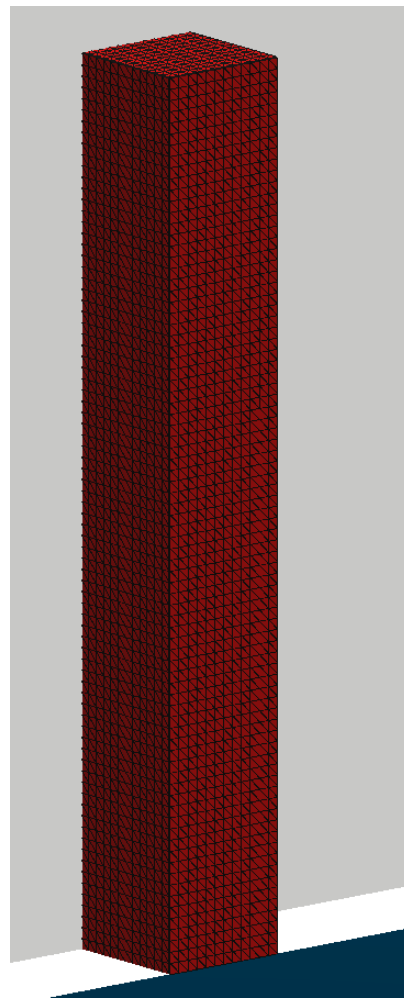
Geometry import options for .stl files.

Specify the Model Units used in Abaqus to setup the structural simulation in the correspondent drop-down list box.

⚠ Please note: The ensemble of the coupling surfaces must create a watertight geometry in order to be correctly simulated in XFlow.



Abaqus Structural Mesh



XFlow imported and re-tessellated mesh

Structural Co-simulation Mode

A complete list of Co-simulation mode options is provided in the [CSS Co-simulation](#) section.

When setting up an XFlow - Abaqus co-simulation the following options are available for the **Co-simulation mode** (options visible only when Abaqus/Simpack [Co-simulation](#) is selected):

- **Abaqus 2-way:** It allows exchange of forces from XFlow to Abaqus and the exchange of displacements from Abaqus to XFlow. Only available with **Fixed** behaviour shapes.
- **Abaqus 1-way deformations:** It allows exchange of displacements from Abaqus to XFlow but no forces are sent from XFlow to Abaqus. This option is relevant when forces from XFlow are negligible for the Abaqus setup. Only available with **Fixed** behaviour shapes.
- **Abaqus 1-way forces:** It allows exchange of forces from XFlow to Abaqus but no displacements are sent from Abaqus to XFlow. This option is relevant when displacements from Abaqus are negligible for the XFlow setup. Only available with **Enforced** behaviour shapes.

8.3 CSS Co-simulation



Structural Behaviour specification

Please note: The geometries which are selected as **Abaqus 2-way | Abaqus 1-way deformations | Abaqus 1-way forces** must be set to the finest lattice resolution of the simulation.

XFlow / Abaqus execution

To start the coupled simulation follow the steps here specified:

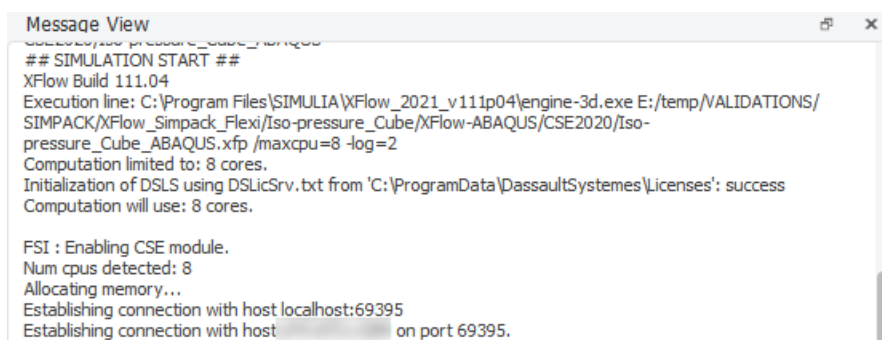
Please note: At this stage the co-simulation is only supported via command line for the Abaqus and the CSE execution. XFlow can be executed either from the GUI or in command lines ([+info](#)).

1. execute the XFlow simulation (example below in command lines)

- `$XFlow_INSTALL/generateDomain3d $project.xfp`
- `$XFlow_INSTALL/engine-3d $project.xfp`

Optionally, it is possible to override the Co-Simulation host and port details saved in the `$project.xfp` file with the command line option `-csedirector=$CSE_HOST:$CSE_PORT`.

Example: `$XFlow_INSTALL/engine-3d $project.xfp -csedirector=localhost:1025`



XFlow log file with indication of FSI: Enabling CSE module, CSE host and port.

XFlow simulation will start and pend (Establishing connection with host `$CSE_HOST:$CSE_PORT`) till both Abaqus and the CSE are executed.

2. Copy the provided FSI_II_std_css.xml file (located in \$XFlow-INSTALL/cse/FSI_II_std_css.xml) to the location of the Abaqus input file and modify accordingly to the case setup, specifying:

- the duration of the simulation (should be the same in both XFlow and Abaqus) by modifying the line
`<duration>1</duration>`
- the Co-Simulation negotiation method, which will define the way information will be transferred, by modifying the line: `<negotiationMethod>MASTER</negotiationMethod>`. Available options are MIN, MAX, MASTER, CONSTANTDT. For more information please consult the Abaqus Co-Simulation Engine manual.
- the name of the interaction geometry in XFlow by changing the lines:

```

                                <connector name="Connector-CFD">
                                    <componentInstance>Fluid</componentInstance>
                                    <variables>
                                        <input>
                                            <variable>coordinates.Bar</
variable>
                                            <variable>displacement.Bar</
variable>
                                            <variable>velocity.Bar</variable>
                                        </input>
                                        <output>
                                            <variable>force.Bar</variable>
                                        </output>
                                    </variables>
                                </connector>

```

 **Please note:** Alternatively, you can name the interaction geometry "MovingWall" in the XFlow setup, so that there is no need to modify the FSI_II_std_css.xml file.

3. Start the CSE in command line using the [cosim_orchestrator](#) provided with the XFlow installation specifying:

- the flag to start the CSE (`--cse`)
- the file containing the co-simulation settings (`--cseconfig FSI_II_std_css.xml`)

Optionally, it is possible to override the Co-Simulation host and port details saved in the \$project.xfp file with the command line option `--listenerport $CSE_PORT`.

cosim_orchestrator example run command: `$XFlow_INSTALL/cosim_orchestrator.bat $project.xfp --cse --cseconfig FSI_II_std_css.xml --listenerport 1025 &`

4. Execute Abaqus solver specifying:

- the `-job` option, which points to the co-simulation step specified in the provided FSI_II_std_css.xml setting file (`-job FSI_II_std`)
- the required input file (`-input $Abaqus_input`)
- the address and port of the Co-Simulation Engine (`-csedirector $CSE_HOST:$CSE_PORT`). The \$CSE_HOST can be set as localhost if the Co-simulation engine is run on the same machine where Abaqus and XFlow are executed.

- set the simulation to double precision (`-double`)
- optionally the Abaqus case can be run in interactive mode (`-int`)

Abaqus example run command: `abaqus -job FSI_II_std -input example -double -csedirector localhost:1025 -int`

The status of the coupled XFlow / Abaqus simulation can be checked examining the log files created in the execution folder.

The termination of any of the two processes (either XFlow or Abaqus) will terminate the entire co-simulation.

Both Abaqus and XFlow will write intermediate step solutions at user's specified frequency.

8.3.2 Simpack

In order to setup a coupled XFlow / Simpack simulation follow the steps:

- Setup the Simpack model including a definition of the **113: Fluid Interface** force element in Simpack (check the Simpack doc for more info)
- Specify the Simpack installation folder in XFlow under **Preferences > Co-simulation** (enable the tick boxes to launch both Simpack and CSE on the local host [+info](#))
- Import the Simpack model in XFlow, **Options > Import Simpack model** ([+info](#))
 - The Co-simulation analysis will automatically be enabled under **Project Tree > Environment > Engine > Advanced Options** ([+info](#))
 - The Co-simulation mode will automatically be enabled for each geometry imported under **Project Tree > Geometry** ([+info](#));
- Setup the fluid simulation in XFlow according to the specific case
- Execute the XFlow simulation, the Simpack one and the Co-Simulation Engine ([+info](#)).

Limitations of Current Implementation

- The current XFlow / Simpack implementation supports all XFlow solvers.
- Every geometry that is set as **Co-Simulation mode: Simpack flexi 2-way | Simpack flexi 1-way deformations** (and hence will exchange surface mesh deformations) must have a single Wall boundary condition applied. This means that the [apply boundary condition to faces](#) option is not valid for such geometries.

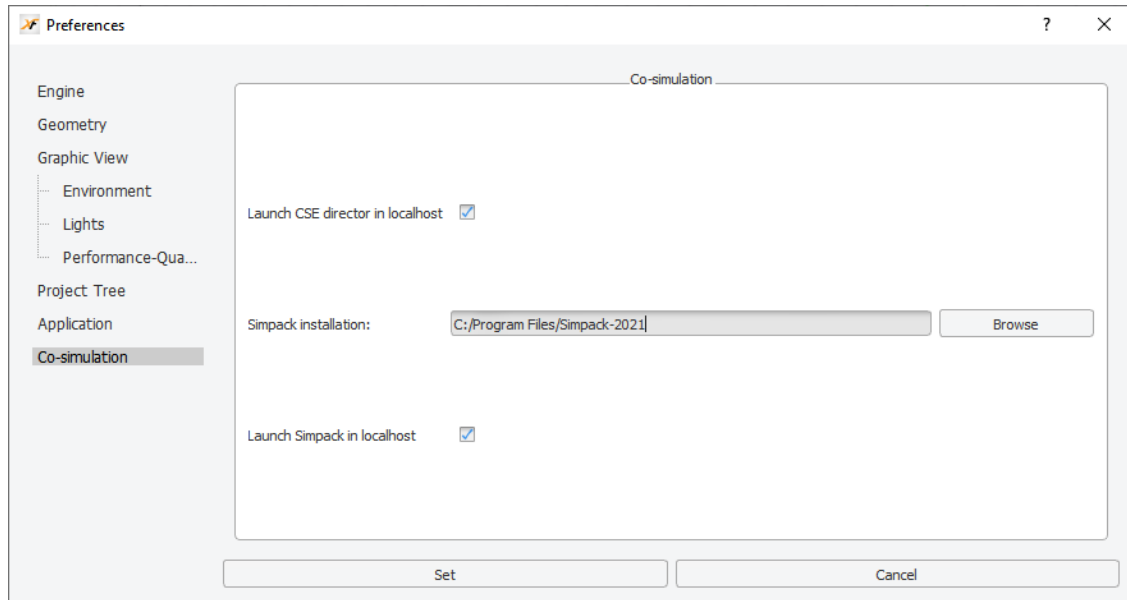
Simpack Installation Settings

In order to import Simpack models XFlow requires access to the Simpack installation folder. This setting can be specified in the **Preferences > Co-simulation** menu.

Additionally, both the CSE (Co-Simulation Engine) and the Simpack execution can be launched

automatically from the XFlow GUI by enabling the options:

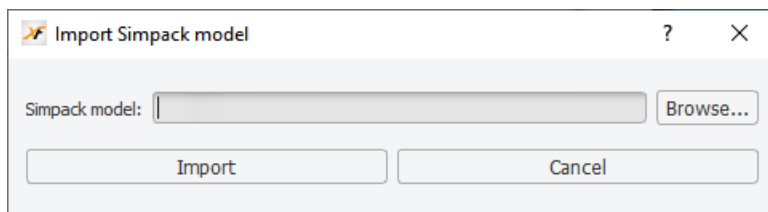
- Launch CSE director in localhost
- Launch Simpack in localhost



Settings for Simpack co-simulation

Import Simpack model

A Simpack model that has defined **113: Fluid Interface** force element can be imported in XFlow GUI using **Options > Import Simpack model**. The model can be selected using the Browse... button to specify the full Simpack file path. Alternatively, the name of the Simpack setup file can be typed into the window and a relative path to the location of the XFlow .xfl file will be used. The latest option is the preferred one to guarantee that the Simpack setup is not lost when copying the XFlow setup to different folders (e.g. HPC execution).



Import Simpack model window.

Multiple bodies can be imported from a single Simpack setup, as long as each body is included in the definition of the **113: Fluid Interface** force element (multiple bodies can be included in one force element and/or multiple force elements can be used).

The XFlow-Simpack co-simulation can be performed considering both rigid bodies and/or flexible bodies. The following considerations apply for the two cases.

Rigid Bodies

XFlow-Simpack co-simulation with rigid bodies will exchange forces/torque values at the CoG of each body and displacements/rotations values of the CoG. Therefore, up to 6 DOFs (Degrees of Freedom) can be specified in the Simpact setup for each body and 6 Forces/Moments will be exchanged by XFlow.

When importing bodies which are defined as Rigid in Simpact the following parameters will be automatically set under **Project Tree > Geometry**

- **Behaviour:** Enforced
- **Co-simulation mode:** Simpact rigid 2-way
- **Co-simulation region name:** name of the co-simulation region read from the Simpact setup

In addition, the shape position and orientation values will be limited to constant values (no time-dependent laws or formulas allowed). The value of position, rotation and scale will be added to the Simpact provided ones during the co-simulation.



Rigid body import default values set in XFlow GUI

Flexible Bodies

When importing bodies which are defined as Linear flexible in Simpact the following parameters will be automatically set under **Project Tree > Geometry**

- **Behaviour:** Fixed
- **Co-simulation mode:** Simpact flexi 2-way
- **Co-simulation region name:** name of the co-simulation region read from the Simpact setup



Linear flexible body import default values set in XFlow GUI

Simpack Co-simulation Mode

A complete list of Co-simulation mode options is provided in the [CSS Co-simulation](#) section.

When importing a Simpack model in XFlow, the default settings for the **Co-simulation mode** are:

Simpack flexi 2-way (flexible bodies) and **Simpack rigid 2-way** (rigid bodies). Other options are available and are described below.

Rigid Bodies

All options under rigid bodies exchange information on the CoG of each shape. Therefore, concentrated forces/torques are computed in XFlow and CoG translations/rotations are sent from Simpack to XFlow.

Options are:

- **Simpack Rigid 2-way**: The default value for rigid bodies. It allows exchange of forces from XFlow to Simpack and the exchange of displacements from Simpack to XFlow.
- **Simpack Rigid kinematics**: It allows exchange of displacements from Simpack to XFlow but no forces are sent from XFlow to Simpack. This option is relevant when forces from XFlow are negligible for the Simpack setup.
- **Simpack Flexi 1-way forces**: It allows exchange of forces from XFlow to Simpack but no displacements are sent from Simpack to XFlow. This option is relevant when displacements from Simpack are negligible for the XFlow setup.

Flexible Bodies

All options under flexible bodies exchange information on the surface mesh of each shape. Therefore, distributed forces are computed in XFlow and mesh nodes deformation is sent from Simpack to XFlow.

Options are:

- **Simpack Flexi 2-way**: The default value for flexible bodies. It allows exchange of forces from XFlow to Simpack and the exchange of displacements from Simpack to XFlow.
- **Simpack Flexi 1-way deformation**: It allows exchange of displacements from Simpack to XFlow but no forces are sent from XFlow to Simpack. This option is relevant when forces from XFlow are negligible for the Simpack setup.

XFlow / Simpack Co-simulation Execution

⚠ Please note: The simulation time set in Simpack should be identical to the XFlow one. Also the Global time variable in Simpack must be left to the default value of 0.

The entire co-simulation process (XFlow + CSE + Simpack) can be started easily from the XFlow GUI by selecting **Run > Start Computation** if the required options in [Simpack installation settings](#) have been enabled.

Alternatively, it is possible to execute each component separately:

1. Start XFlow, Run > Start Computation
2. Start the Simpack Solver, Time Integration > Offline (with Measurements). Make sure the CSE server is setup to remote and that the same port number as the XFlow setup is used ([+info](#)).
3. Start the CSE in command line using the [cosim_orchestrator](#) provided with the XFlow installation,
`$XFlow_INSTALL/cosim_orchestrator.bat $project.xfp --cse &`

For batch execution with no graphic interface, the following commands will launch each component:

1. Start the XFlow simulation
 - `$XFlow_INSTALL/generateDomain3d.bat $project.xfp`
 - `$XFlow_INSTALL/xflow-engine.bat $project.xfp` (under Linux OS the path to the CSS libraries must be set prior to the XFlow execution, [+info](#))
2. Start the Simpack simulation, `$Simpack_INSTALL/simpack-slv.exe --no-gui --csedirector localhost:69395 $project.spck`
3. Start the CSE using the [cosim_orchestrator](#) ([+info](#)) provided with the XFlow installation,
`$XFlow_INSTALL/cosim_orchestrator.bat $project.xfp --cse`

Optionally, it is possible to override the Co-Simulation host and port details saved in the `$project.xfp` file with the XFlow command line option `-csedirector=$CSE_HOST:$CSE_PORT` (i.e.
`$XFlow_INSTALL/xflow-engine.bat $project.xfp -csedirector=$CSE_HOST:$CSE_PORT`
)

The status of the coupled XFlow / Simpack simulation can be checked examining the log files created in the execution folder.

The termination of any of the two processes (either XFlow or Simpack) will terminate the entire co-simulation.

XFlow will write intermediate step solutions (frames) at user's specified frequency.


```

Message View
Cube_Rotation
## SIMULATION START ##
XFlow Build 111.04
Execution line: C:\Program Files\SIMULIA\XFlow_2021_v111p04\engine-3d.exe E:/temp/VALIDATIONS/
SIMPACT/XFlow_Simpack/Rigid-Cube_Rotation-Translation/Rigid-Cube_Rotation.xfp /maxcpu=8 -log=2
Computation limited to: 8 cores.
Initialization of DSLS using DSLSrv.txt from 'C:\ProgramData\DassaultSystemes\Licenses': success
Computation will use: 8 cores.

FSI : Enabling CSE module.
Num cpus detected: 8
Allocating memory...
Establishing connection with host localhost:69395
Establishing connection with host [redacted] on port 69395.
Successfully connected to host [redacted] on port 69395.
Successfully connected to host localhost:69395
Establishing connection with host [redacted] on port 50853.
Successfully connected to host [redacted] on port 50853.
Processing geometry...
Done!

-----
Num levels: 1
-----

Level: 0
Num active fluid elements: 288000
-----

Full domain has 288000 elements.
Equivalent single-resolution domain has 288000
Equivalent single-resolution domain size is ( 120 x 40 x 60 )
-----

Computing boundary conditions map!
-----

Coarsest resolved length: 0.1
-----

Prandtl number: 0.741114
Thermodynamic speed of sound: 340.112
Reference area: 1 m^2

```

XFlow log file with indication of FSI: Enabling CSE module and CSE host and port.

Simpack Co-simulation Mapping

When performing XFlow-Simpack co-simulations with **rigid bodies** it is possible to use the [replace selected geometry](#) feature to replace the Simpack provided primitives with a user selected geometry. This feature can be useful to use a more detailed geometry in the XFlow simulation and a coarser representation in the Simpack setup, since concentrated forces/moments and translation/rotation at the CoG will be exchanged.

 **Please note:** The replace geometry cannot be used with **flexible bodies**, since the actual deformed geometry will be updated at each coupling step with the Simpack primitives.

Simpack SIMULIA Co-simulation Settings

When executing each component of the co-sim workflow separately the following settings need to be setup In Simpack: Extras > Options > Solver as per snapshot below (a different port number can be specified)

Simpack settings for XFlow co-simulation

Co-simulation initial time

Within Simpack, it is possible to select an initial time on which to start a simulation that includes time dependant variables (like a signal or a force) within the model.

Therefore, to avoid any de-synchronization issues with the co-simulation with XFlow, this initial time value that has been setup in Simpack, will be automatically read by the XFlow Graphical Interface when importing the Simpack model and preset to match the same value that was established in Simpack.

To make it a user-friendly feature, if the user modifies the initial time value in Simpack, to avoid having to re-import the Simpack model in XFlow, you can simply edit this value within the XFlow GUI, ensuring that it matches the Simpack.

⚠ Please note: To avoid synchronization issues, it is compulsory that the user ensures that the Simpack initial time value always matches the value in this field prior to running.

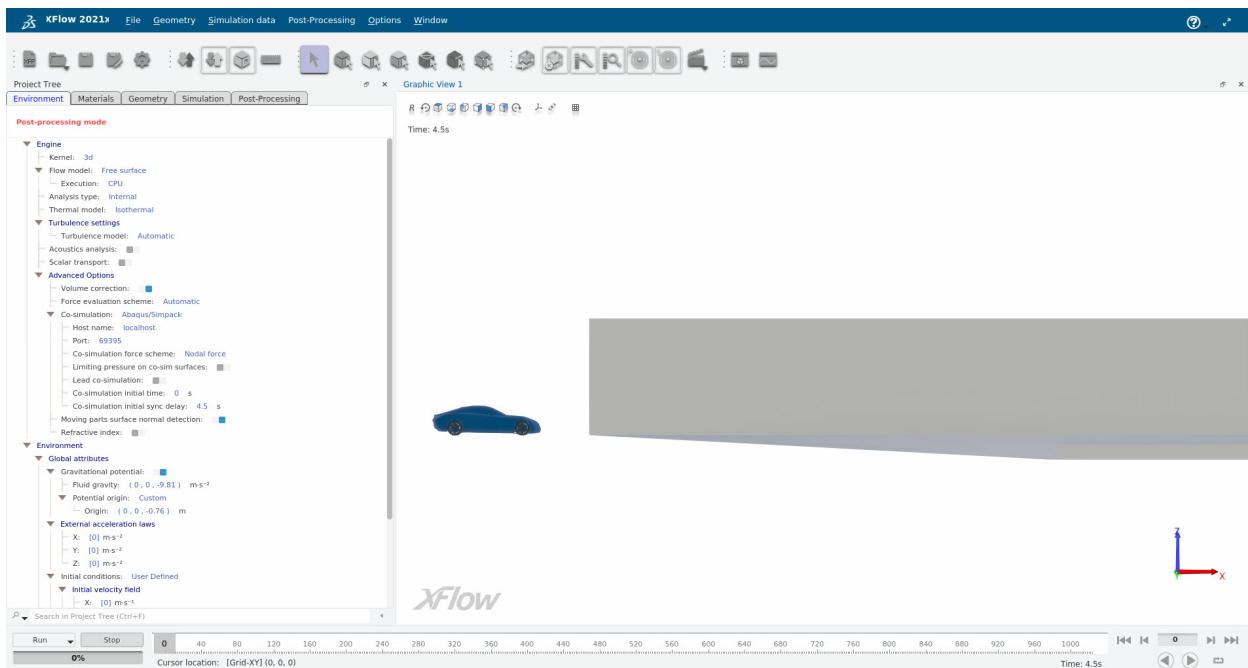
Co-simulation initial sync delay

When performing XFlow-Simpack co-simulations it is possible to select the time at which the XFlow fluid solver will synchronize with the motion received from Simpack, through a time delay. This can improve the overall simulation performance seen as on most occasions, the XFlow simulation will be more computationally expensive than the Simpack one, and any unnecessary fluid simulation time steps can be neglected.

In the example below, an XFlow-Simpack co-simulation has been performed for a Vehicle Wading application, with this feature enabled.

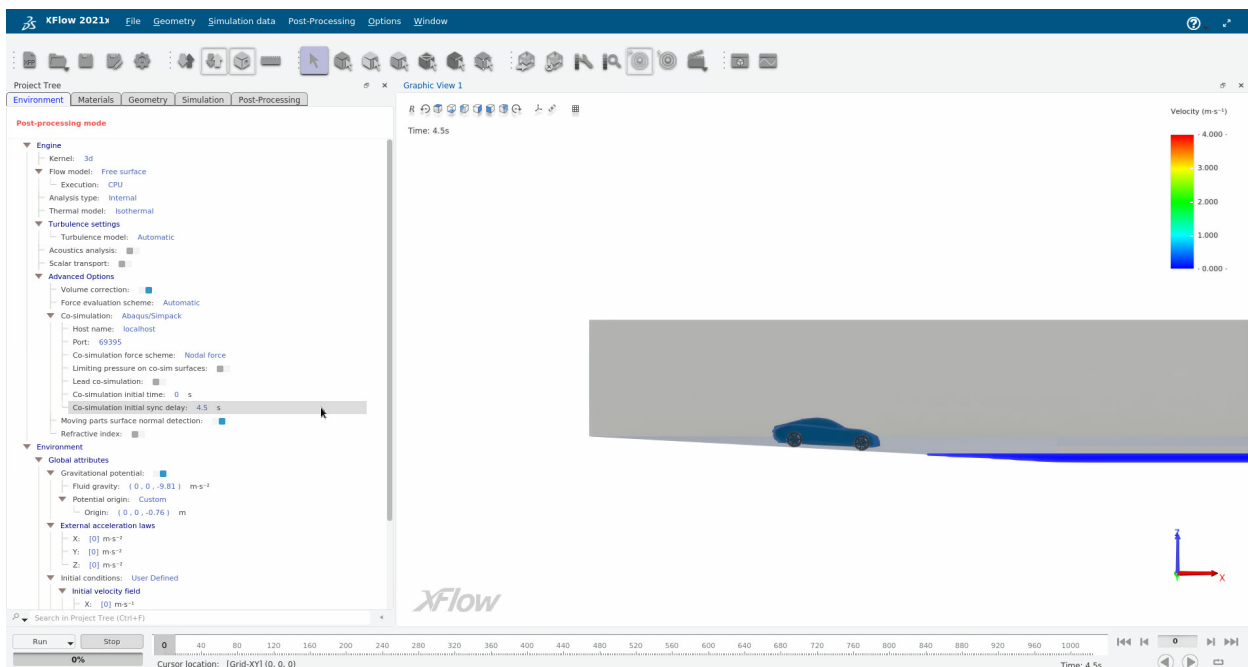
On the Simpack side, it is convenient to initialize the Simpack simulation horizontally, without any inclination and progressively send the car down the tank's ramp.

When loading the Simpack model to XFlow, the geometry can be seen at the same location where it is initialized in Simpack.



Due to the Co-simulation initial sync delay being set a 4.5 seconds, when we load the simulation data we can see that the XFlow simulation has been started 4.5 seconds after the Simpack car model has been displaced in and down the tank ramp.

This optimizes the overall run time seen as the XFlow simulation is now initialized when the car is much closer to the initial water level, reducing the initial waste of moving the car through void when we are interested with the interaction with the water.



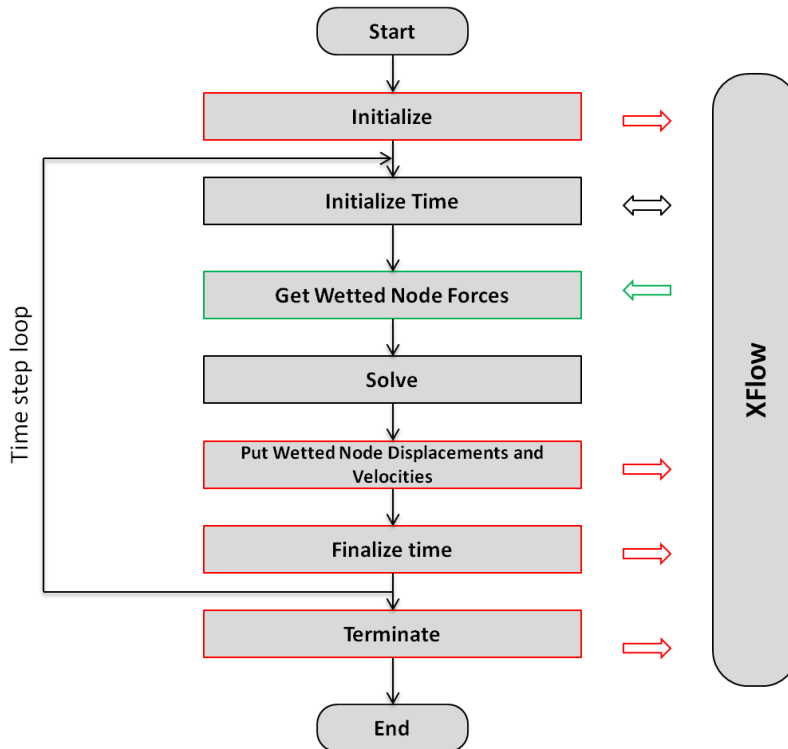
8.4 Nastran

- **MSC Nastran:** Two-way coupling is implemented through the OpenFSI standard, which involves the nonlinear SOL400 solution of MSC Nastran ([+info](#)).
- **MSC Nastran:** One-way thermal coupling is implemented through the modification of an imported .bdf file ([+info](#)).

8.4.1 2-way OpenFSI

Project Tree > Environment > Engine > Advanced Options > Structural analysis

XFlow provides the possibility to perform coupled fluid-structure simulations using an external FEA (Finite Element Analysis) structural solver. In particular, it is possible to setup co-simulations XFlow / MSC Nastran exploiting an ad hoc developed OpenFSI SCA service. This service enables a two-way coupling of the software through the exchange of loads information from XFlow to MSC Nastran, and the resultant structural displacements from MSC Nastran to XFlow. A flow diagram illustrating the connectivity between the two software is presented below. For more information about how to setup a MSC Nastran simulation with OpenFSI services please refer to the MSC Nastran User Defined Services guide.



The data flow for the XFlow / MSC Nastran coupled simulation for an explicit OpenFSI SCA service.

Please note: The OpenFSI service allows data exchanges only to and from the non-linear solution sequence MSC Nastran SOL400.

In order to setup a coupled XFlow / MSC Nastran simulation please install the XFlowOpenFSI Service ([+info](#)) (this only has to be done once) and then follow the steps:

- Setup the structural model in either MSC SimXpert, including the specification of wetted elements and

- the OpenFSI service to use ([+info](#));
- b) Enable the Structural analysis in XFlow by selecting the **Nastran** option from the dropdown list in **Project Tree > Environment > Engine > Advanced Options > Structural analysis**;
 - c) Import the MSC Nastran mesh ([+info](#));
 - d) Select Structural as geometry behaviour for the imported MSC Nastran meshes ([+info](#));
 - e) Setup the fluid simulation in XFlow according to the specific case;
 - f) Execute the XFlow simulation (it will automatically synchronize with MSC Nastran) and the MSC Nastran one ([+info](#)).

When executing coupled XFlow / MSC Nastran simulations it is possible to select a combination of 2D / 3D settings for both software. For example, it is possible to simulate 2D fluid flows and 3D structural deformations or viceversa. In the first case, the 2D loads provided by XFlow are distributed on the correspondent surface of the 3D MSC Nastran model, while the deformations of the center section of the 3D FEA mesh are mapped to the correspondent 2D geometry in XFlow. When considering 3D fluid flows and 2D MSC Nastran model, instead, the fluid loads are concentrated onto the 2D FEA mesh and the correspondent displacements are mapped to the 3D geometry in XFlow.

 **Please note:** Inputs in MSC Nastran must be in SI units.

 **Please note:** MSC Nastran coupling is available in [Labs mode](#) only.

Limitations of Current Implementation

- The current XFlow / MSC Nastran implementation is limited to **Single-Phase, Isothermal** simulations executed in SMP (Shared Memory Processing) environment. In addition, both software installation must be found in the same machine.
- The current implementation of OpenFSI in XFlow only supports MSC Nastran 2013.1.1.
- Currently, only the explicit OpenFSI Service is implemented, which means that there is no exchange of information between XFlow and MSC Nastran during the Newton-Raphson loop in MSC Nastran (for more detailed information please refer to the MSC Nastran User Defined Services guide).
- Every geometry that is set as **Structural** behavior (and hence will exchange information with MSC Nastran) must have a single boundary condition applied. This mean that the [apply boundary condition to faces](#) option is disable for such geometries.
- It is recommended to specify refinement regions in such a way that the **Structural** geometry does not intersect different lattice levels during the whole duration of the simulation.
- XFlow supports the following MSC Nastran element types for a fluid-structure simulation:
 - Points
 - GRID
 - 3D elements
 - CHEXA
 - CPENTA
 - CTETRA

2D elements

- CQUAD {4,8,R}
- CTRIA {3,6,R}
- WETELMG (support for tri3, tri6 and quad4 only)
- WETELME

 **Please note:** Non-linear elements can also be used, however, XFlow will re-generate the geometry tessellation and split the non-linear elements into a combination of linear ones. This approach gives rise to approximation in the representation of non-linear deformation (within a single element) and can also give rise to small holes and gaps in the mesh. Because of this, the usage of non-linear elements is not recommended.

 **Please note:** The current implementation of OpenFSI in XFlow only supports MSC Nastran 2013.1.1.

XFlow OpenFSI Service Installation

The XFlowOpenFSI Service must be installed before it is possible to run a coupled XFlow / MSC Nastran simulation.

Windows

The following steps describe the installation procedure in Windows (64 bit) machines, where two scenarios are contemplated: no other OpenFSI installed in the machine; previous OpenFSI installed.

Scenario 1: no other OpenFSI installed

- Copy the folder <XFlow_install_dir>\nastran\Apps to
<Nastran_install_dir>\MSC_Nastran\<version>\msc<version>\nast\services
- Set the following Windows environment variables:
 - SCA_LIBRARY_PATH=<Nastran_install_dir>\MSC_Nastran\<version>\msc<version>\nast\services\ Apps\WIN8664\lib
 - SCA_SERVICE_CATALOG=<Nastran_install_dir>\MSC_Nastran\<version>\msc<version>\nast\services\ Apps\res\SCAServiceCatalog.xml
 - SCA_RESOURCE_DIR=<Nastran_install_dir>\MSC_Nastran\<version>\msc<version>\nast\services\ Apps\res

Scenario 2: previous OpenFSI installed

- Copy the folder <XFlow_install_dir>\nastran\Apps\WIN8664\lib\xflow to
<Nastran_install_dir>\MSC_Nastran\<version>\msc<version>\nast\services\ Apps\WIN8664\lib
- Copy the folder <XFlow_install_dir>\nastran\Apps\res\types\xflow to
<Nastran_install_dir>\MSC_Nastran\<version>\msc<version>\nast\services\ Apps\res\types

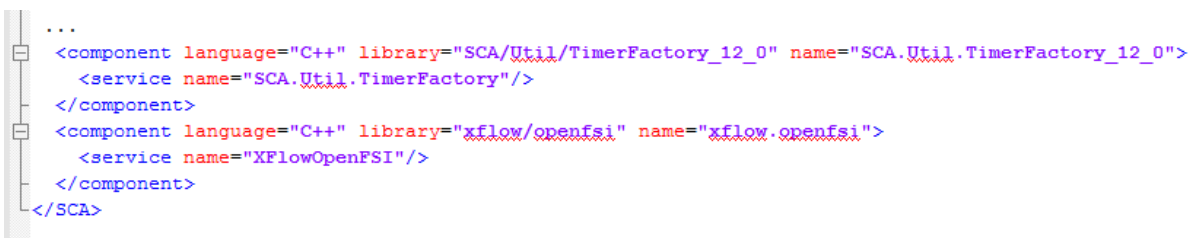
The structure of the folders should reflect the following:

```
|-- <Nastran_install_dir>\MSC_Nastran\<version>\msc<version>\nast\services\  
|   |-- Apps  
|       |-- res  
|           -- SCAServiceCatalog.xml  
|               |-- types
```

```
|           |-- xflow
|               -- openfsiComp.xml
|           |-- WIN8664
|               |-- lib
|                   |-- xflow
|                       -- openfsi.dll
```

- Add the following lines to the file `<Nastran_install_dir>\MSC_Nastran\<version>\msc<version>\nast\services\Apps\res\SCAServiceCatalog.xml`, just before the last line `</SCA>`:


```
<component language="C++" library="xflow/openfsi" name="xflow.openfsi">
  <service name="XFlowOpenFSI"/>
</component>
```



```
...
<component language="C++" library="SCA/Util/TimerFactory_12_0" name="SCA.Util.TimerFactory_12_0">
  <service name="SCA.Util.TimerFactory"/>
</component>
<component language="C++" library="xflow/openfsi" name="xflow.openfsi">
  <service name="XFlowOpenFSI"/>
</component>
</SCA>
```

Example of the modification required on the SCAServiceCatalog.xml file

- Add the following lines to the Windows environment variables:
 - `SCA_LIBRARY_PATH=<Nastran_install_dir>\MSC_Nastran\<version>\msc<version>\nast\services\Apps\WIN8664\lib`
 - `SCA_SERVICE_CATALOG=<Nastran_install_dir>\MSC_Nastran\<version>\msc<version>\nast\services\Apps\res\SCAServiceCatalog.xml`
 - `SCA_RESOURCE_DIR=<Nastran_install_dir>\MSC_Nastran\<version>\msc<version>\nast\services\Apps\res`



Please note: The XFlowOpenFSI Service might not be recognized by MSC Nastran if a different version than "4.8.5" of the Qt library is used by a program included in the %PATH% Windows environment variable. In this case, please do include the `<XFlow_install_dir>` as the first item in the %PATH% Windows environment variable.

Linux

The following steps describe the installation procedure in Linux (64 bit) machines, where two scenarios are contemplated: no other OpenFSI installed in the machine; other OpenFSI Services present.

Scenario 1: no other OpenFSI installed

- Copy the folder `<XFlow_install_dir>/nastran/Apps` to `<Nastran_install_dir>/MSC_Nastran/<version>/msc<version>/nast/services`
- Set the following Linux variables (using the export command in the `.bashrc` file):
 - `SCA_LIBRARY_PATH=<Nastran_install_dir>/MSC_Nastran/<version>/msc<version>/nast/services/Apps/LX8664/lib`
 - `SCA_SERVICE_CATALOG=<Nastran_install_dir>/MSC_Nastran/<version>/msc<version>/nast/services/Apps/res/SCAServiceCatalog.xml`
 - `SCA_RESOURCE_DIR=<Nastran_install_dir>/MSC_Nastran/<version>/msc<version>/nast/services/Apps/res`

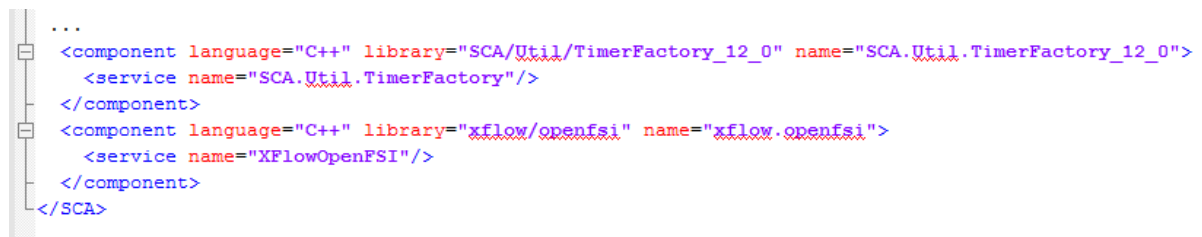
Scenario 2: previous OpenFSI installed

- Copy the folder <XFlow_install_dir>/nastran/Apps/LX8664/lib/xflow to <Nastran_install_dir>/MSC_Nastran/<version>/msc<version>/nast/services/Apps/LX8664/lib
- Copy the folder <XFlow_install_dir>/nastran/Apps/res/types/xflow to <Nastran_install_dir>/MSC_Nastran/<version>/msc<version>/nast/services/Apps/res/types

The structure of the folders should reflect the following:

```
|-- <Nastran_install_dir>/MSC_Nastran/<version>/msc<version>/nast/services/
|   |-- Apps
|       |-- res
|           -- SCAServiceCatalog.xml
|           |-- types
|           |-- xflow
|               -- openfsiComp.xml
|       |-- LX8664
|           |-- lib
|               |-- xflow
|                   -- openfsi.dll
```

- Add the following lines to the file <Nastran_install_dir>/MSC_Nastran/<version>/msc<version>/nast/services/Apps/res/SCAServiceCatalog.xml, just before the last line "</SCA>":
 <component language="C++" library="xflow/openfsi" name="xflow.openfsi">
 <service name="XFlowOpenFSI"/>
 </component>



```
...
<component language="C++" library="SCA/Util/TimerFactory_12_0" name="SCA.Util.TimerFactory_12_0">
  <service name="SCA.Util.TimerFactory" />
</component>
<component language="C++" library="xflow/openfsi" name="xflow.openfsi">
  <service name="XFlowOpenFSI" />
</component>
</SCA>
```

Example of the modification required on the SCAServiceCatalog.xml file

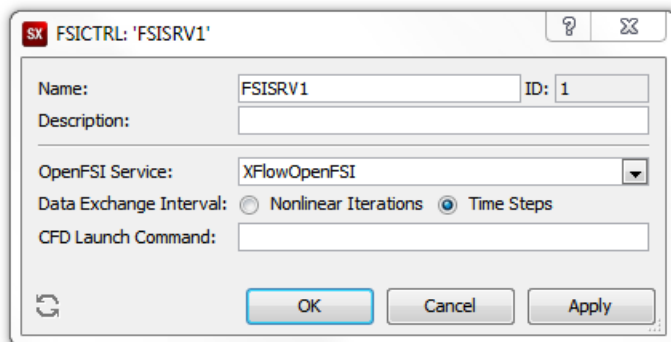
- Add the following lines to the export command in the .bashrc file:
 - SCA_LIBRARY_PATH=<Nastran_install_dir>/MSC_Nastran/<version>/msc<version>/nast/services/Apps/LX8664/lib
 - SCA_SERVICE_CATALOG=<Nastran_install_dir>/MSC_Nastran/<version>/msc<version>/nast/services/Apps/res/SCAServiceCatalog.xml
 - SCA_RESOURCE_DIR=<Nastran_install_dir>/MSC_Nastran/<version>/msc<version>/nast/services/Apps/res

⚠ Please note: The XFlowOpenFSI Service might not be recognized by MSC Nastran if a different version than "4.8.5" of the Qt library is used by a program included in the \$LD_LIBRARY_PATH Linux environment variable. In this case, please do include the <XFlow_install_dir> as the first item in the \$LD_LIBRARY_PATH Linux environment variable.

Setup Usage of XFlow OpenFSI Service

In MSC SimXpert select the XFlowOpenFSI Service under the option **User Services > OpenFSI**. If the specific service is not available from the dropdown list insert the line **XFlowOpenFSI** in the field **OpenFSI**

Service:

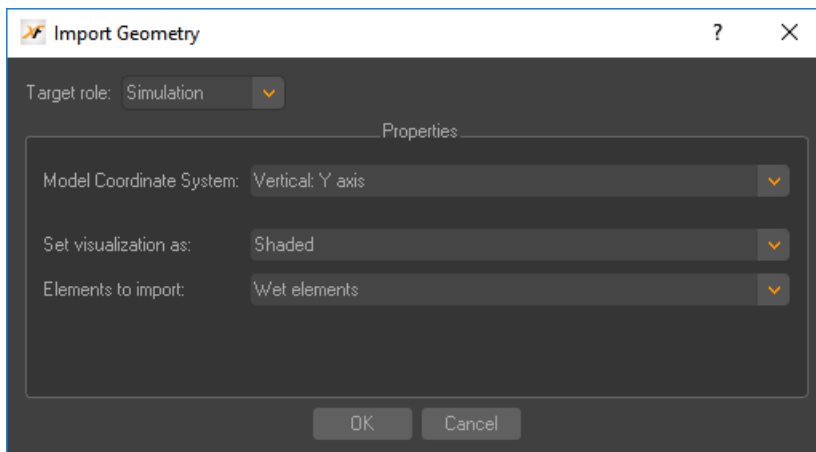


XFlow OpenFSI Service specification in MSC SimXpert

Import MSC Nastran mesh

Main menu > Geometry > Import a new geometry,  or <Ctrl + i>

XFlow can import mesh definitions from the following MSC Nastran file types: .BDF, .BULK, .BLK, .DAT, .DECK, .NAS, .PCH. When reading a mesh from a supported MSC Nastran file XFlow will prompt the user which elements to import, as shown in the picture below:



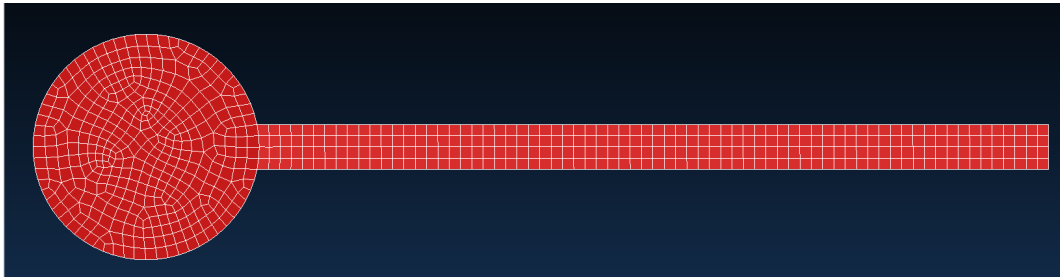
Geometry import options for Nastran .bdf files.

The **Wet elements** option only loads the surfaces of the elements which have previously been marked as wet (i.e. in contact with the fluid). These can either be WETELME or WETELMG, according to the way the mesh has been setup in MSC SimXpert (or any other compatible pre-processing tool).

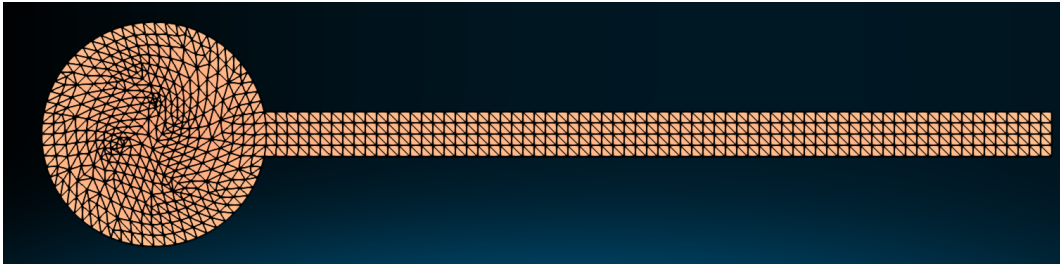
The **All elements** option, instead, imports every grid point defined in the .bdf file independently of their status as "wet elements".

 **Please note:** The ensemble of the wet surfaces of the MSC Nastran mesh must be watertight in order to be correctly handled in XFlow.

 **Please note:** Only the MSC Nastran element types [here](#) specified are supported in XFlow.



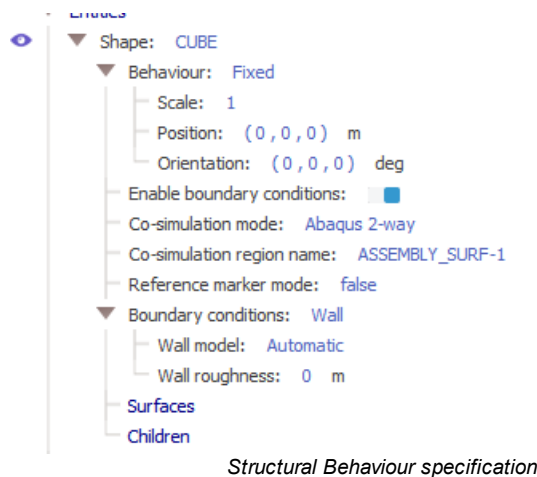
MSC Nastran structural mesh



XFlow imported and re-tessellated mesh

Structural Coupling Option

Within XFlow it is necessary to specify the geometries which will have a flexible structural behaviour and will, hence, exchange information with MSC Nastran. This is done by setting **Structural coupling: Two way** from the **Project Tree > Geometry > Geometries > Shape > Structural coupling**. This option is only enabled when the [structural analysis](#) is switched on. Although initial position and orientation are available in the GUI, the current implementation is limited to geometry with (0,0,0) initial Position and Orientation. The geometry deformation law will be given by the structural solver during the computation.



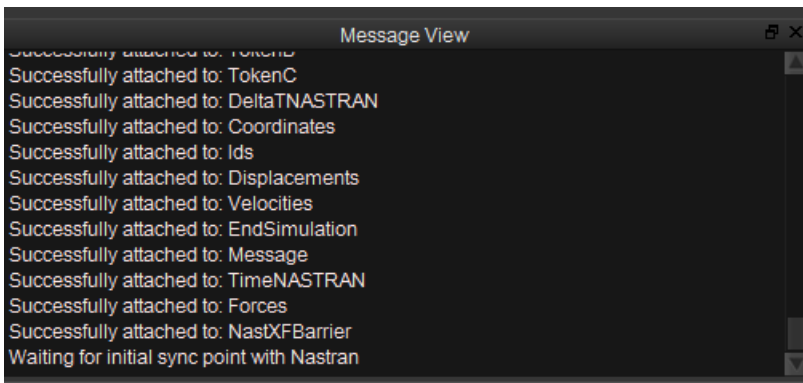
Structural Behaviour specification

⚠ Please note: The geometries which are selected as **Structural coupling: Two way** must be set to the finer lattice resolution of the simulation and should be limited to the case of (0,0,0) initial Position and Orientation.

XFlow / MSC Nastran execution

To start the coupled simulation first execute the XFlow simulation by selecting the **Start Computation** option

from the **Run** button ([+info](#)). The message view will indicate that XFlow is waiting for the initial sync point with MSC Nastran.



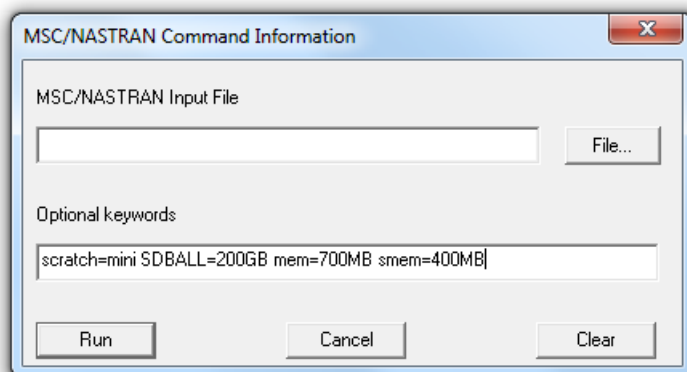
XFlow message view communicating the start of a coupled simulation and the need to run MSC Nastran

Execute MSC Nastran and select the generated .bdf file (identical to the one imported in XFlow to setup the case, see [here](#)).

The status of the coupled XFlow / MSC Nastran can be checked from the log file (<name_of_bdf_file>.log) generated in the execution folder. This is also the main file used for debugging issues during the coupled simulation.

To terminate the simulation stop the XFlow execution selecting the **Stop** button ([+info](#)) and stop the MSC Nastran execution.

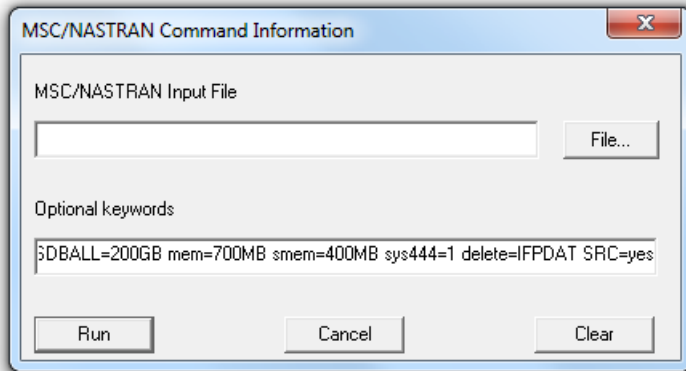
It is highly recommended to specify the following optional keywords for MSC Nastran to optimize the simulation runtime: "scratch=mini SDBALL=200GB mem=700MB smem=400MB".



MSC Nastran optional keywords for runtime optimization

In addition, when using the R4 DEV beta version of MSC Nastran please add the extra parameters: "sys444=1 delete=IFPDAT src=yes".

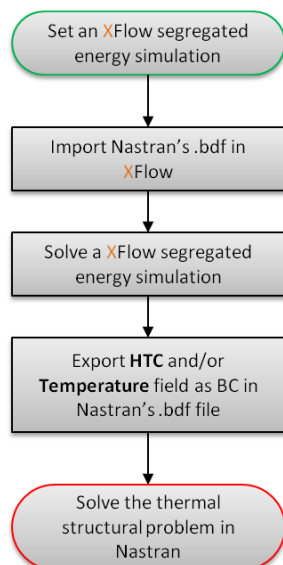
8.4 Nastran



MSC Nastran extra optional keywords for R4 DEV beta version

8.4.2 1-way Thermal

XFlow allows a 1-way coupled thermal simulation with Nastran. The goal in these type of analysis, as shown in the diagram below, is to obtain the Heat Transfer Coefficient (HTC) ([+info](#)) and Temperature surface distributions from XFlow's simulations and set them as boundary condition in the Nastran setup. It is, hence, possible to solve the thermal structural problem on the body in a separate Nastran simulation.



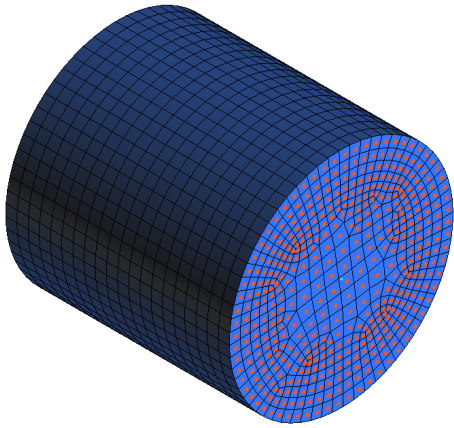
1-way Nastran thermal analysis workflow

⚠ Please note: The Nastran thermal export functionality is only available for **Single phase, Free surface Segregated energy** thermal model.

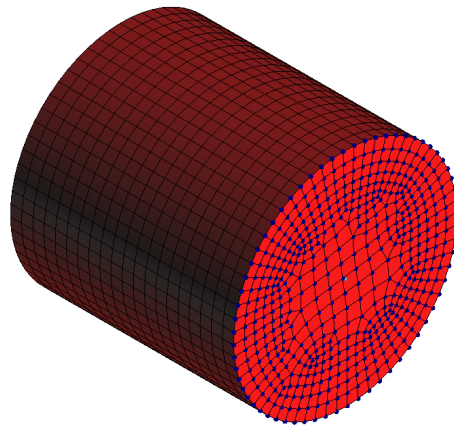
In order to setup a 1-way coupled XFlow / Nastran thermal simulation please follow the steps:

- a) Set a boundary condition (set to 0 value) of Heat Transfer Coefficient (CONV key in Nastran) or Initial

Temperature (TEMP key in Nastran) on the relevant in Nastran's .bdf file

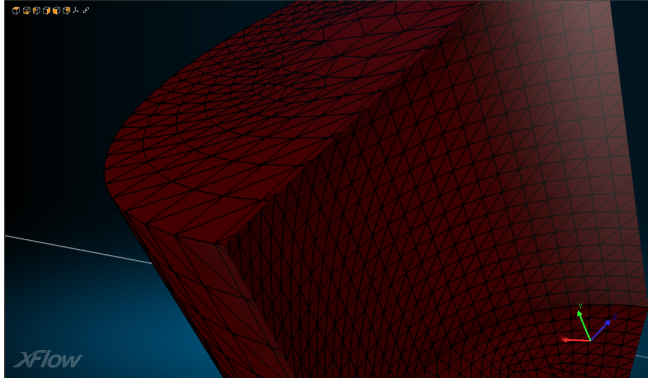


Initial HTC thermal boundary condition set on the surface of solid elements.

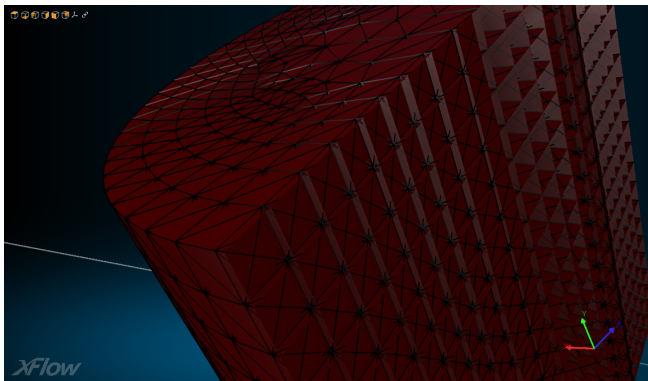


Initial Temperature thermal boundary condition set on the nodes of solid elements.

- b) Setup a Single **Phase/Free Surface** Thermal model: **Segregated energy** simulation in XFlow;
- c) Import the Nastran .bdf file containing the structural analysis setup. It is highly advised to set correctly the surface elements of interest as "wet" and select the **Wet element** option. This represents the most accurate setup and will also reduce the computation time (no elements within the solid will be considered). However, if no wet element definition is present in the .bdf file please do select the **All elements** option;



Nastran .bdf file import, Wet Elements option.



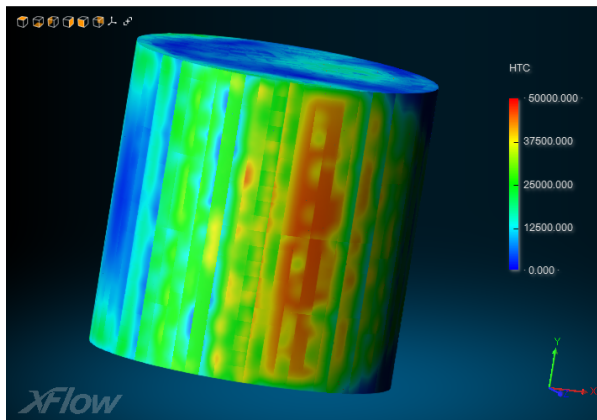
Nastran .bdf file import, All element option.

d) Set a proper set of thermal boundary conditions on the imported geometry. Those could either be (the conditions are not exclusive):

- **Conjugate Heat Transfer** volumetric condition on the entire body;
- **Temperature** or **Free-convection** on at least one of the body's surface.

d) Execute the XFlow's simulation;

e) It is possible to visualize the surface field distribution prior to exporting them using the **Surface info** post-processing ([+Info](#))

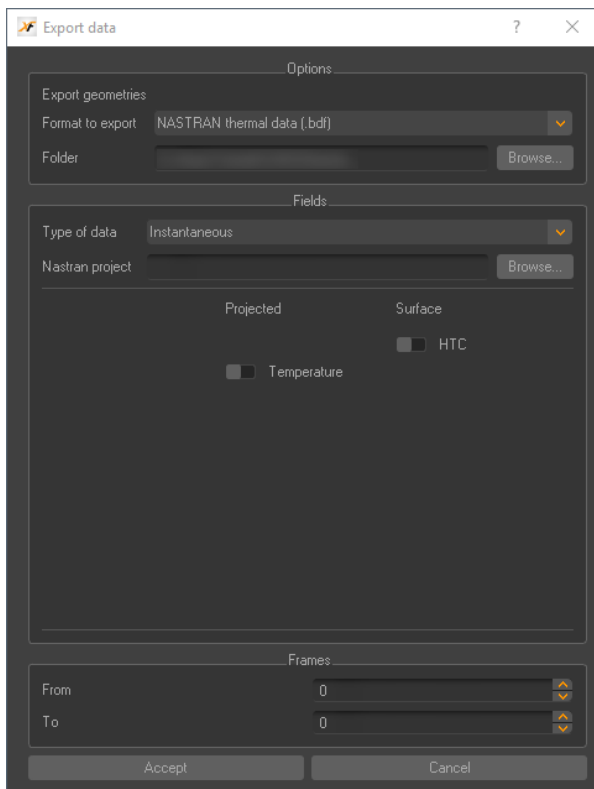


Heat Transfer Coefficient surface info.

⚠ Please note: When performing CHT simulations in XFlow, if the solution is not yet in thermal equilibrium too different values of Temperature will be defined on the body surface. The value shown by XFlow (and exported) by default is the temperature of the fluid in contact with the body. If the body temperature wants to be exported its normals must be reversed first ([+info](#)). This allows XFlow to select the temperature field of the solid region.

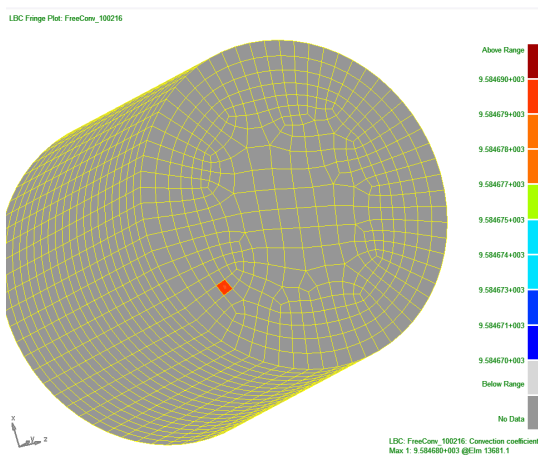
f) Export the surface HTC or Temperature field using the **Simulation data > Export data**. Select **NASTRAN thermal data (.bdf)** in the **Format to export** option, as shown in the image below;

- **Folder** specifies in which folder the modified .bdf will be created.
- **Type of data** specifies if Instantaneous/Averaged data will be used when exporting the **Temperature** field only. HTC values are always instantaneous.
- **Nastran project** set the project which will be modified automatically by XFlow to include the boundary conditions. The same .bdf file used as import in the geometry session must be used.
- Select either **Temperature** (volumetric field projected onto the surface's body) or **HTC** (actual surface values calculated by the solver). The selected field tag will be added to the exported filename.
- **Frames** select the frame of interest or a range of frames to export. In the latter case several files will be created with the frame number included in the filename (e.g. xflowproject_Nastran_0_HTC.bdf, xflowproject_Nastran_1_HTC.bdf, etc...)

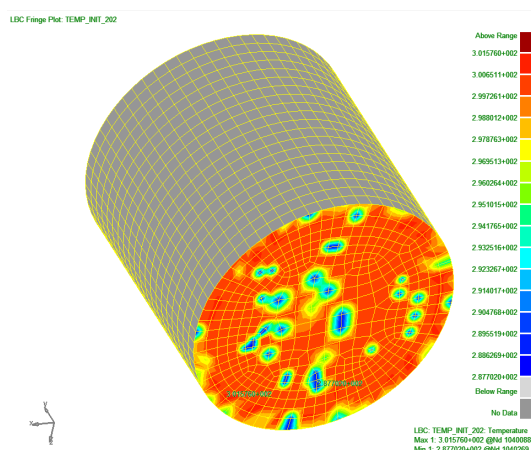


Export data of HTC or Temperature fields in Nastran .bdf file.

- g) One or several (depending on the frame range selected) files will be created which filename will be defined as "xflowproject_\$Nastran-file_\$frame_\$variable.bdf". The file(s) will contain a boundary condition for each of the element faces in contact with the fluid, when the HTC variable is exported. When, instead the Temperature variable is exported an initial temperature distribution is set in the newly created .bdf files, as shown in the snapshot below.



Visualisation of a HTC boundary condition applied on an element's surface set as CONV in the initial .bdf file.



h) Use the created .bdf file(s) in g) as input for Nastran's simulation(s).

Limitations of Current Implementation

- The current 1-way XFlow/Nastran thermal coupling is limited to **Single phase, Free surface Segregated energy** thermal model simulations.
- Only **Fixed** geometry behaviour can be set and no initial Position and Orientation can be specified.

9 Application modes

This chapter explains the different [Application modes](#) available:

- Normal mode: basic features and options.
- [Expert mode](#): basic + advanced features and options.
- [Labs mode](#): basic + advanced + under-development features and options.

9.1 Expert mode

The Expert mode includes the advanced analysis features. This mode can be enabled in:

Main menu > Options > Preferences > Application mode

The user has to restart the XFlow interface to make the application mode change effective.

The features available in the Expert mode are:

- [Scalar transport](#)
- [Acoustic analysis](#)
- [Turbulence generation](#)
- [Advanced options](#)
 - High order boundary conditions
 - Force evaluation scheme
 - Wall function time filter
- [Automatic initial conditions](#)
- [Virtual moving wall boundary](#)
- [Immersed Boundary Method](#)
- [Wake refinement threshold](#)
- [Refinement transition length](#)
- [Buffer zone length](#)
- [Highest available frequency](#) / [Arbitrary lattice level](#)

9.1 Expert mode

- [Generate scripts](#)
- [Import ADAMS simulation](#)
- [Start advanced computation](#)
- [LODI boundary conditions](#)
- [CATPart and CATProduct importation](#)

9.2 Labs mode

The Labs mode includes additional analysis possibilities that are still under development. The use of these features must therefore be done with a particular care.

The Labs mode can be activated in:

Main menu > Options > Preferences > Application mode

The user has to restart the XFlow interface to make the application mode change effective.

The Labs mode additional features are:

- [GPU execution](#)
- [Supersonic flow](#)
- [Coupled energy flow](#)
- [Multiphase - Phase field](#)
- [Spalart-Allmaras](#) (turbulence model)
- [Time integration scheme](#)
- [MLS interpolation](#)
- [Reference pressure point](#)
- [Output format](#) (CGNS)
- [Highest available frequency](#)
- [Animated geometry behavior](#)
- [Import ADAMS simulation](#)
- [FMI Standard](#)
- [Abaqus Co-simulation](#)
- [MSc Nastran Co-simulation](#)
- [Multiphase - VoF](#)

9.2.1 Execution mode

Project Tree > Environment > Engine > Flow model > Execution: CPU/GPU. This option is only available in [Labs mode](#) and allows to select if the simulation will run on CPU or GPU.

Execution: CPU

Normal type of XFlow execution. The solver will be running on CPU.

Execution: GPU

New solver for XFlow GPU simulations. The solver runs using GPU instead of CPU, offering great speed-up compared to CPU execution.

The GPU solver is still an on-going development and therefore it is limited to the following features:

Engine

- 3d kernel
- Flow model: Single Phase (**SP**) | Free surface (**FS**) | Multiphase Particle-based tracking (**PBT**) / Phase Field (**PF**)
- Isothermal solver
- Turbulence model: Automatic | Off
- Turbulence generation: Off
- Acoustic analysis : Off
- Scalar transport : On | Off
- Advanced options : Volume correction

Environment

- Initial conditions : User Defined | Wind Tunnel Default
- Reference velocity : Custom
- Ground wall : Automatic | Resolved | Non-equilibrium Wall function | Free-slip
- Lateral boundaries : Automatic | Non-equilibrium Wall function | Resolved | Free-slip
- Generic rectangular domain -> Wall model : Automatic | Non-equilibrium Wall function | Resolved | Free-slip

Materials

- Viscosity model: Newtonian
- Surface tension (PBT/PF only)
- Contact angle (PBT/PF only)
- Contact angle hysteresis (PF only)
- Interface thickness (PF only)

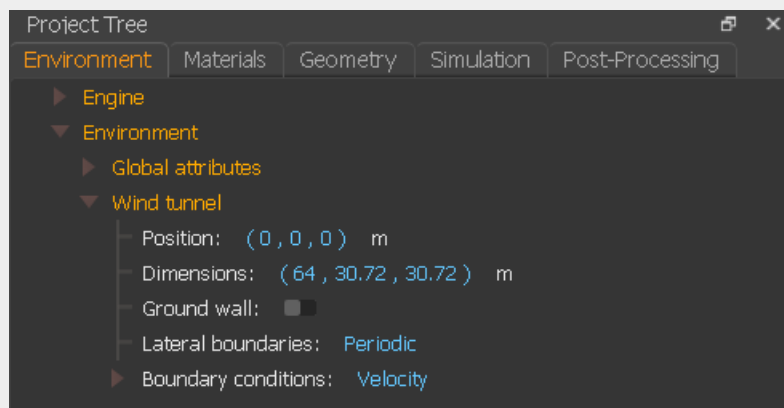
Geometry

- Behavior: Fixed | Enforced (SP/FS/PBT only)
- Boundary conditions : Wall | Velocity/Mass Flow/Pressure/Total Pressure Inlet | Velocity/Mass Flow/Pressure/Total Pressure Outlet | Porous Wall Inlet (PF only) | Porous Volume
- Moving boundaries: Immersed Boundary method: On
- Shape contact angle (PBT/PF only): On | Off
- Wall model : Automatic | Non-equilibrium Wall function | Resolved | Free-slip

Simulation

- Time step mode : Custom | Automatic
- Refinement algorithm: Disabled | Near static walls | Adaptive refinement (SP only)
- Seal small gaps: Automatic | Disabled | All shapes | Individual shapes
- Refinement transition length : 8
- Buffer zone length : Always 0
- Refinement in wind direction : On | Off
- Refinement based on simulation data : Disabled | Averaged total pressure
- Numerical data time-step : Solver time step
- Save averaged : Off
- Save resume : Off
- Compute markers : Off
- Output format : XFlow
- Probes
- Solver surface integrals (only working with VOF and velocity)

⚠ Please note: Due to its lattice generation algorithm which is block-based, the domain restricts the dimensions of the fluid domain to a multiple of 8 times the coarsest resolution as minimum. Blocks are 8x8x8 voxels in the entire domain structure. Also, the wind tunnel dimensions are automatically adapted to be an entire multiple of blocks.



Virtual wind tunnel dimensions for a resolved scale of 0.32 m.

⚠ Please note: When using moving bodies in GPU, the following best-practices should be followed:

1. Force integration scheme: Surface Stress Integration should be enabled in the Environment tab ([+info](#)).
2. The frame frequency should be a multiple of 1/dt.

9.2.2 Supersonic flow

Project Tree > Environment > Engine > Thermal model

Supersonic: Only available in [Labs mode](#). Allows to solve flows with fluid speed close or higher than the speed of sound.

Examples of applications: aerodynamics for supersonic, transonic, and hypersonic flows, flows involving shockwaves, etc.

⚠ Please note: This solver is only a prototype and therefore the output should be considered with precaution.

Please note: The solver can only support supersonic flow velocities, while enforced geometries should have motion laws which result in velocities lower than the speed of sound.

9.2.3 Coupled energy flow

Project Tree > Environment > Engine > Thermal model

$$C_p \frac{D(\rho T)}{Dt} = \underbrace{\nabla \cdot (k \nabla T)}_{\text{Heat flux}} + \underbrace{\nabla \cdot (\tau \cdot \vec{v})}_{\text{Viscous term}} - \underbrace{p \nabla \cdot \vec{v}}_{\text{Compressibility effects}}$$

where T is the temperature, ρ is the density, C_p is the specific heat capacity, k is the thermal conductivity, τ is the viscous stress tensor, and $\vec{v}$ the velocity vector.

Coupled energy: Only available in [Labs mode](#). The [energy equation](#) is solved and takes into account for the compressibility term. This solver is useful in order to account for the pressure/temperature variations when the gas is highly compressed/expanded, and is valid only for isentropic processes.

Examples of applications: adiabatic isentropic compression, expansions, etc.

Please note: This solver is only a prototype and therefore the output should be considered with precaution.

9.2.4 Spalart-Allmaras

Project Tree > Environment > Engine > Turbulence Model

Spalart-Allmaras is the only RANS (in fact, uRANS) model included in XFlow.

This model solves a single transport equation that determines a variable $\tilde{\nu}$ named Spalart-Allmaras variable, and that is identical to viscosity except near the walls. Viscosity relations are as following:

$$\nu_{turbulent} = \tilde{\nu} f_{v1} \quad \text{with} \quad \begin{cases} f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3} \\ \chi \equiv \frac{\tilde{\nu}}{\nu} \end{cases}$$

C_{v1} is a constant of 7.1 by default.

The transport equation of the Spalart-Allmaras variable is:

$$\frac{\partial}{\partial t} (\rho \tilde{\nu}) + \frac{\partial}{\partial x_\alpha} (\rho \tilde{\nu} v_\alpha) = P_\nu + \frac{1}{\sigma_{\nu_t}} \left[\frac{\partial}{\partial r_\beta} \left\{ (\mu + \rho \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial r_\beta} \right\} + C_{b2} \rho \left(\frac{\partial \tilde{\nu}}{\partial r_\beta} \right)^2 \right] - D_\nu$$

where P_ν is the turbulent viscosity production term, D_ν the destruction term, ν is the molecular kinematic viscosity and σ_{ν_t} and C_{b2} are constants respectively equal to 2/3 and 0.622 by default.

The production term is calculated as below:

$$P_\nu = C_{b1} \rho \tilde{S} \tilde{\nu}$$

$$\tilde{S} \equiv S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \quad \text{with} \quad \begin{cases} S \equiv |\Omega_{\alpha\beta}| + 2 \min(0, |S_{\alpha\beta}| - |\Omega_{\alpha\beta}|) \\ f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \end{cases}$$

where

$$|\Omega_{\alpha\beta}| = \sqrt{2\Omega_{\alpha\beta}\Omega_{\alpha\beta}}$$

$$|S_{\alpha\beta}| = \sqrt{2S_{\alpha\beta}S_{\alpha\beta}}$$

$$\Omega_{\alpha\beta} = \frac{1}{2} \left(\frac{\partial v_\alpha}{\partial r_\beta} - \frac{\partial v_\beta}{\partial r_\alpha} \right)$$

$$S_{\alpha\beta} = \frac{1}{2} \left(\frac{\partial v_\alpha}{\partial r_\beta} + \frac{\partial v_\beta}{\partial r_\alpha} \right)$$

And the destruction term is calculated as follows:

$$D_\nu = C_{w1}\rho f_w \left(\frac{\tilde{\nu}}{d} \right)^2 \quad \text{with} \quad \begin{cases} f_w = g \left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right]^{1/6} \\ g = k + C_{w2} (k^6 - k) \\ k \equiv \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2} \\ C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{(1 + C_{b2})}{\sigma_{\nu_t}} \end{cases}$$

where C_{w2} , C_{w3} , K and C_{b1} are constants respectively equal to 0.3, 2, 0.4187 and 0.1355 by default.

9.2.5 Time integration scheme

[Project Tree > Environment > Engine > Advanced options > Time integration scheme](#)

For *Single Phase* cases, XFlow allows the user to choose between the following time integration schemes:

- **Standard:** Second order accuracy
- **Fractional Propagation:** [1]

[1] Fractional Propagation and the Elimination of Staggered Invariants in Lattice-BGK Models, Yue-Hong Qian, Int. J. Mod. Phys. C, 08, 753 (1997).

9.2.6 MLS interpolation

[Project Tree > Post-Processing > General > Interpolation](#)

Moving least squares (MLS) is a method of reconstructing continuous functions from a set of point samples via the calculation of a weighted least squares measure biased towards the region around the point at which the reconstructed value is requested.

This method improves the interpolation between different lattice levels or domain border limiting some discontinuities which can be shown by the convolution method, despite of being slightly slower.

A comparison table between the three interpolation method is presented below:

	Off	Convolution	MLS
Computation time	Fast	Medium	Slow
Approximation order	0	3	2
Preserves value at nodes	YES	YES	NO
Preserves maxima and minima	YES	NO	NO
Discontinuity in-between levels	HIGH	MEDIUM	LOW

9.2.7 Reference pressure point

Project Tree > Environment > Global attributes > Reference pressure point: On/Off

The reference pressure point option appears only when no pressure boundary condition is set in the simulation and allows to fix one point of the domain at the reference pressure. If no pressure boundary condition is set, the gauge pressure may diverge since there is no condition to impose the pressure equilibrium in the fluid domain.

 **Please note:** The reference pressure point is only available for Single Phase flow model.

9.2.8 Output format

XFlow can save volumetric data in different formats directly from the engine:

XFlow format

Project Tree > Simulation > Store data > Output format > Native XFlow: On/Off

The XFlow native format is a binary format and saves one file per frame, per field and per data type. See [XFlow files](#).

 **Please note:** XFlow native format can be read only by XFlow from the XFlow interface.

CGNS format

Project Tree > Simulation > Store data > Output format > CGNS: On/Off

CGNS stands for CFD General Notation System. It is a general, portable, and extensible standard for the storage and retrieval of CFD analysis data. It consists of a collection of conventions, and free and open software implementing those conventions. CGNS is readable by most external post-processing software.

 **Please note:** In the case averaged data is saved, XFlow saves in the last frame of the CGNS file the averaged data simulation. CGNS format is not available for moving geometries and adaptive refinement.

VTK format

Project Tree > Simulation > Store data > Output format > VTK: On/Off

The Visualization Toolkit (VTK) (<http://www.vtk.org/>) is an open-source, freely available software system for 3D computer graphics, image processing, and visualization. VTK is readable by most external post-processing software.

Please note: In the case averaged data is saved, XFlow saves in the last frame of the VTK file the averaged data simulation. VTK format supports moving parts and adaptive refinement.

9.2.9 Highest available frequency

Project Tree > Simulation > Store data > Numerical data frequency

This sampling mode allow the user to save the data with the highest frequency available in the domain, i.e. the frequency in the finer level of the lattice.

Remember: The **Time step**, either estimated by XFlow or given by the user, corresponds to the biggest resolution of the lattice. Other resolution levels are automatically created using spatial and temporal resolutions twice smaller than the previous level, resulting in an octree structure both in space and time.

Please note: This feature is only relevant when running a simulation with different levels of refinement, otherwise it is equivalent to **Solver time step**.

9.2.10 Animated geometry behaviour

Project Tree > Geometry > Geometries > Shape > Behaviour

This feature allows the user to perform fluid-structure-interaction (FSI) simulations where the solid behaves as a non-rigid body, according to a user-defined motion.

This feature is defined with the following parameters:

- ↳ **Source > Frame folder:** Path to the folder where a sequence of .STL geometries defining the **Shape** motion is stored. The STL files are read following their lexicographic order as shown in the examples below.

Example 1:

Frame folder- containing: frame0.stl, frame1.stl, frame2.stl, frame10.stl, frame11.stl

Animated geometry behaviour - given by the following geometry sequence:

1. Frame 0 = frame0.stl
2. Frame 1 = frame1.stl
3. Frame 2 = frame10.stl
4. Frame 3 = frame11.stl
5. Frame 4 = frame2.stl

Example 2:

Frame folder- containing: b.stl, d.stl, a.stl

Animated geometry behaviour - given by the following geometry sequence:

1. Frame 0 = a.stl
2. Frame 1 = b.stl
3. Frame 2 = d.stl

Example 3:

Frame folder- containing: firstFrame.stl, secondFrame.stl, thirdFrame.stl, fourthFrame.stl

Animated geometry behaviour - given by the following geometry sequence:

1. Frame 0 = firstFrame.stl
2. Frame 1 = fourthFrame.stl
3. Frame 2 = secondFrame.stl
4. Frame 3 = thirdFrame.stl

Example 4:

Frame folder- containing: 05_frame.stl, 01_frame.stl, 02_frame.stl, 00_frame.stl, 20_frame.stl

Animated geometry behaviour - given by the following geometry sequence:

1. Frame 0 = 00_frame.stl
2. Frame 1 = 01_frame.stl
3. Frame 2 = 02_frame.stl
4. Frame 3 = 05_frame.stl
5. Frame 4 = 20_frame.stl

⚠ Please note: Every geometry of the sequence must have the same number of triangles. Only vertex displacements are accounted for. Hence, from a .STL geometry to the next of the sequence, vertices can be moved but no topological changes can be made. Therefore, facet should be listed always in the same order and vertices of each facet should be in the same order too.

⚠ Please note: The deformation velocity is applied at the geometrical vertices (not the surface) and, hence, if those vertices are not in the fluid domain no effect is transferred to the surrounding fluid. This also applies to geometry with a sparse vertices count on the surface definition.

↩ **Source > Frames per second:** Number of the .STL geometries stored in **Frame folder** that has to be morphed per second.

↩ **Volume preservation: On/Off.** If this switch is activated XFlow will check whether the volume of the **Shape** is conserved after each morphing. In case it is not, it will enforced the volume conservation by rescaling the mesh.

⚠ Please note: This geometry behaviour is not compatible with [Apply boundary condition to faces](#).

10 Command lines

XFlow simulations can be executed by command lines which enables the automation of the simulations executions. Two main executions methods exist:

- Batch mode: this is the simplest way to automate your simulation execution, but requires OpenGL support.
- Advanced command lines: this is a step-by-step execution but allows more control in the process and can be executed without OpenGL support.

10.1 Advanced command lines

Command lines helps the user to use more advanced batch capabilities. For instance, this can be helpful to manually run simulations on clusters, but requires to execute each step of the simulation separately.

⚠ Please note: All the examples below are given for Linux systems. Please adapt adequately if working on a Windows system. Example: XFlow executable becomes XFlow.bat instead of xflow-gui, engine-3d becomes engine-3d.bat, etc.

General command lines

Open XFlow interface

- To start XFlow from the command line:

```
[xflow-project-folder]> [xflow-installation-path]/xflow-gui [xflow-project-folder]/project.xfp
```

Open XFlow interface + launch computation

- To start XFlow and launch the computation:

```
[xflow-project-folder]> [xflow-installation-path]/xflow-gui [xflow-project-folder]/project.xfp /run
```

⚠ Please note: /run sequentially executes: (i) domain generation; (ii) solver calculation. These three actions can be individually performed as explained below.

Simulation launch command lines

There are three steps to execute a computation in command lines:

1) Domain generation

- Launch the domain generator using the simulation file specified:

```
[simulation-folder]> [xflow-installation-path]/generateDomain3d [simulation-folder]/project.xfp <options>
```

2) Engine execution

The XFlow-engine.bat (xflow-engine in Linux) handles the execution of the solver.

- Launch the engine for serial calculation:

```
[simulation-folder]> [xflow-installation-path]/xflow-engine [simulation-folder]/project.xfp <options>
```

- Launch the solver for MPI calculation: more details in the [Distributed computations](#) section below.

Alternatively, it is possible to specify the actual executable required by each flow model.

- Launch the engine for serial calculation:

```
[simulation-folder]> [xflow-installation-path]/engine-3d[-flow-model]  
[simulation-folder]/project.xfp <options>
```

- Launch the solver for MPI calculation: more details in the [Distributed computations](#) section below.



Please note: In this section the input file used is the **.XFP** file format, however the same procedure applies with the **.XFZ** file. Both formats can be used to run the domain generator as well as the XFlow engines.

Distributed computations (MPI)

In order to run a distributed MPI computation firstly the domain must be decomposed in n partitions. This process is performed at the domain generation stage by running the domain generator with the `-mpi` option:

```
generateDomain3d [project-name].xfp -mpi=n
```

The distributed solver run is executed using the command

```
mpirun -n [number_of_partitions] [other-mpi-parameters] [xflow-installation-  
path]/xflow-engine [project-name].xfp [mpi-version] <options>
```

The `[mpi-version]` flag defines which engine to use in order to guarantee compatibility with the system MPI version. The default options defined for the Linux installation are:

```
-ompi=1.4 (corresponding to OpenMPI 1.4)  
-ompi=1.6 (corresponding to OpenMPI 1.6)  
-ompi=1.10 (corresponding to OpenMPI 1.10)  
-impi (corresponding to IntelMPI)
```

While the only option available for Windows installation is:

```
-impi (corresponding to IntelMPI)
```

The available `[mpi-version]` options are defined in the text file `xflow_solver_mpi_ids.txt` located in the XFlow installation folder. This file should not be deleted nor modified.

The user should specify the `mpi` option that matches the MPI implementation installed on the computing environment.

Alternatively, it is possible to specify the actual executable required by each flow model.

```
mpirun -n [number_of_partitions] [other-mpi-parameters] [xflow-installation-  
path]/engine-3d[-flow-model]-[mpi] [project-name].xfp <options>
```

where the `[mpi]` suffix corresponds to the specific MPI compatible engine:

- `engine-3d[-flow-model]-mpi:` for Intel MPI
- `engine-3d[-flow-model]-mpi-ompi:` for OpenMPI 1.4 (Linux only)
- `engine-3d[-flow-model]-mpi-ompi6:` for OpenMPI 1.6 (Linux only)
- `engine-3d[-flow-model]-mpi-ompi10:` for OpenMPI 1.10 (Linux only)

 **Please note:** The MPI command illustrated above is a generic one assuming OpenMPI is used. This command mostly depends on the MPI implementation (Intel MPI, OpenMPI, etc.) and HPC system. Please contact your system administrator for more details.

GUI options

The options for the XFlow.bat executable are the following:

/nogui	→	Runs a simulation from the XFlow GUI without opening the window.
/maxcpu=xxx	→	This is the number of CPU to use in the computation.
/setgl1	→	Opens the GUI forcing the OpenGL 1.1 renderer.

Engine options

The options for the engine-3d* executables are the following:

/log=mode	→	Different log modes depending on mode value: • /log=0: saves to log file and print to the screen (default), • /log=1: only save the log file, • /log=2: only print to the screen.
/nosavedata	→	It does not save any data
/maxcpu=xxx	→	This is the number of CPU to use in the computation
/	→	Initializes the simulation from a previous computation at the folder given after the "=" symbol
initializewithdatafolder=sim_path		
/initializewithframe=xxx	→	Uses the frame number of the computation defined with /initializewithdatafolder=sim_path
/r	→	Resumes the simulation. This requires the Project Tree > Simulation > Store data > Save resume file option to be enabled before running the simulation to resume.
/sparse	→	To enforce the use of a sparse memory structure for the storage of the static domain. It may incur in a small speed penalty but it will use less memory. It does not change the behaviour of the program if Adaptive refinement is enabled in Project Tree > Simulation > Resolution > Refinement algorithm .
/nosavevelocity	→	Disables the storage of the velocity field data.
/nosavevorticity	→	Disables the storage of the vorticity field data.
/nosavetotalpressure	→	Disables the storage of the total pressure field data.
/nosavestaticpressure	→	Disables the storage of the static pressure field data.
/	→	Disables the storage of the turbulence intensity field data.
nosaveturbulenceintensity		
/nosavevof	→	Disables the storage of the volume of liquid phase field data.
/nosaveviscosity	→	Disables the storage of the effective viscosity field data.
/fmu	→	Starts an FMU computation.
/mpimachlimit=M	→	Must be specified for supersonic engine only. M is the maximum Mach number in the simulation. The lower M and the faster the simulation.

 **Please note:** The options syntax for Windows is "/", whereas it is "-" for Linux. Example: /maxcpu=4 becomes -maxcpu=4 on Linux.

 **Please note:** The option /initializewithdatafolder can be defined in the project file directly (**Project Tree > Environment > Environment > Global attributes > Initial conditions: Simulation data**) but the command line option will have priority over it. Note the path must be defined as seen by the machine executing the calculation.

Domain generator options

The options for the generateDomain3d executable are the following:

/maxcpu=xxx	→	This is the number of CPU to use in the domain generation
/mpi=N	→	Creates N partitions of the fluid domain for MPI computations
/force	→	Forces the domain generation for internal simulations where the fluid domain is not generated properly due to issues in the geometry topology (holes, etc.). The domain generator will force the lattice generation filling the bounding box of the entire simulation, thus including elements out of the geometry within the bounding box.

The domain generator features additional options for [advanced MPI partition optimization](#). Default values are recommended however.

 **Tip:** The -mpi option can be cumulated several times in order to create several partitions at once. This is very useful when the number of nodes/cores that will be used for the MPI simulation is not yet determined since it avoids to generate the domain again for another number of partitions. Example:
`generateDomain3d project.xfp -mpi=4 -mpi=8 -mpi=16.`

Executables

Each flow model requires a different engine executable. This is automatically selected when executing the xflow-engine command.

Nonetheless, the list of the available engine executable is here provided:

engine-3d	single phase analysis
engine-3d-inc	single phase analysis + incompressible model
engine-3d-t	single phase + segregated energy analysis
engine-3d-tc	single phase + coupled energy analysis
engine-3d-fs	free-surface analysis
engine-3d-fs-t	free-surface + segregated energy analysis
engine-3d-mfs	multiphase particle-based tracking
engine-3d-mfs-t	multiphase particle-based tracking + segregated energy analysis
engine-3d-pf	multiphase phase field
engine-3d-vof	multiphase volume of fluid
engine-3d-s	single phase + supersonic analysis
engine-3d-gpu	all flow physics running in GPU

List of the executables and how they are related to the values in the .XFP.

If a combination of values does not appear on this list, it means that it is not allowed:

- **engine-3d**

Used when the xfp contains the values:
 <FlowModel>Single phase</FlowModel>
 <ThermalModel type="Isothermal"/>

- **engine-3d-inc**

Used when the xfp contains the values:
 <FlowModel>Single phase</FlowModel>
 <ThermalModel type="Isothermal"/>
 <EnforceIncompressibleModel>On</EnforceIncompressibleModel>

- **engine-3d-t**

Used when the xfp contains the values:
 <FlowModel>Single phase</FlowModel>
 <ThermalModel type="Segregated energy"/>

- **engine-3d-tc**

Used when the xfp contains the values:
 <FlowModel>Single phase</FlowModel>
 <ThermalModel type="Coupled energy"/>

- **engine-3d-fs**

Used when the xfp contains the values:
 <FlowModel>Free surface</FlowModel>
 <ThermalModel type="Isothermal"/>

- **engine-3d-fs-t**

Used when the xfp contains the values:
 <FlowModel>Free surface</FlowModel>
 <ThermalModel type="Segregated energy"/>

- **engine-3d-mfs**

Used when the xfp contains the values:
 <FlowModel>Multiphase</FlowModel>
 <MultiphaseModel>Particle-based tracking</MultiphaseModel>
 <ThermalModel type="Isothermal"/>

- **engine-3d-mfs-t**

Used when the xfp contains the values:
 <FlowModel>Multiphase</FlowModel>
 <MultiphaseModel>Particle-based tracking</MultiphaseModel>
 <ThermalModel type="Segregated energy"/>

- **engine-3d-pf**

Used when the xfp contains the values:
 <FlowModel>Multiphase</FlowModel>
 <MultiphaseModel>Phase field</MultiphaseModel>
 <ThermalModel type="Isothermal"/>

- **engine-3d-vof**

Used when the xfp contains the values:
 <FlowModel>Multiphase</FlowModel>
 <MultiphaseModel>VoF</MultiphaseModel>
 <ThermalModel type="Isothermal"/>

- **engine-3d-s**

Used when the xfp contains the values:
 <FlowModel>Single phase</FlowModel>
 <ThermalModel type="Supersonic"/>

- **engine-3d-gpu**

Used when the xfp contains the values:
 <Execution>GPU</Execution>

Required files for simulation execution

10.1 Advanced command lines

To execute a simulation by command lines, a few files are necessary for XFlow. The **required files** for the given engine executable are:

- for non-MPI engines:
 - [project-name].xfp (or xfz)
 - [project-name].xfd
 - In addition, if the simulation is run with the GPU solver: [project-name].blocks.xfd
- for MPI engines:
 - [project-name].xfp (or xfz)
 - [project-name].xfd
 - [project-name].xfd.part.[N]
 - In addition, if the simulation is run with the GPU solver: [project-name].blocks.xfd

The [project-name].xfd file is the fluid domain file output by the executable generateDomain3d. The [project-name].xfd.part.[N] is the partition file for distributed calculation on N partitions.

Data export by command lines

XFlow data can be exported in several formats, see [Export data](#). This can be executed with command lines as follows:

XFlow.bat project.xfp	-exportdata=ensight,cgns,paraview	Data format to export: Ensight Gold, CGNS, Paraview. Only one argument allowed.
	-exportfrom=N	Starting frame to export. Example: from frame 0.
	-exporttto=M	Ending frame to export. Example: to frame 20.
	-exportdatatype=inst,avg,std	Data type to export: instantaneous, averaged, or standard deviation. Only one argument allowed.
	-exportfields=vel,sp,vort,tp,ti,temp,vof,vel_proj,sp_proj,vort_proj,tp_proj,ti_proj,temp_proj,vof_proj,cp,cf,yplus,pplus	List of fields to export: • Volumetric fields (ex: vel) • Projected fields (ex: vel_proj) • Surface fields (ex: cp) Can contain more than one field separated by commas.

Data are exported in /simulation_folder/exporteddata.

Numerical data export by command lines

XFlow numerical data can be exported into a tabular data file running the xflow_numericaldata_export executable available in the simulation folder. This can be executed with command lines as follows:


```
xflow_numericaldata_export -s "simulationfolder1,
simulationfolder2"

-d "entity1/component1,
entity2/component2"

-o outputFilename.txt
-h
```

Indicates the simulation folder to extract. **A maximum of 2 simulation folders can be extracted at the same time.** Must indicate the entity and component of the data to extract. The entity and component are described in the numericaldata.xml file located in the simulation folder (and within the subfolder node_N folder for MPI simulations). Name of the file to output. Displays the help.

Example of usage: `xflow_numericaldata_export.exe -s "simulation1,simulation2" -d "Overall forces/Fx" -o overall_force_X.txt`
This would extract the overall force Fx of both simulation1 and simulation2 within a file named overall_force_X.txt.

The output format is such as:

```
Time Sim1_Field1 Time Sim2_Field1 Time Sim1_Field2 ... Time Sim1_FieldN Time
Sim2_FieldN
```

Automatic post-processing options

Some of XFlow post-processing capabilities can be automated through command lines while opening an XFlow project file. The automated tasks are the following:

<pre>XFlow.bat project.xfp - importpostprocessingfile="pathToFile" -refreshpostprocessing - exportpostprocessingresults="pathToDir" - -exportanimation - exportallplotlines="PathToDir"</pre>	Loads an xfp file (post-processing file), defined in "pathToFile", when xflow-gui starts. Equivalent to Import xfp. Refreshes all the post-processing features defined in the xfp file (project file) when xflow-gui starts. Equivalent to Refresh Post-processing object. Exports a tabular file with all the curves corresponding to the post-processing non-default numerical data (stream tracers, volume/surface integrals, sensors, data plot lines) when xflow-gui starts. Equivalent to Export numerical data. Generates an animation when xflow-gui starts. Equivalent to Animation. Export all plot lines in .txt format. The specification of "PathToDir" is optional. Default directory is the simulation directory.
---	---

 **Tip:** In order to specify the properties of an animation through an xfp file, it is possible to combine `-exportanimation` with `-importpostprocessingfile="pathToFile"` command.

Please note: The post-processing and exportation options must be executed with the XFlow.exe instead of the XFlow.bat in Windows.

Co-simulation Orchestrator

The `cosim_orchestrator` is an utility tool provided with the XFlow installation to streamline the execution of co-simulation workflows ([+info](#)).

DsIs_stat

The `dsIs_stat` is an utility tool provided with the XFlow installation to query DSLS license servers and provide license availability and usage information. This utility can be used, for example, to check XFlow license availability before execution in HPC environments.

The default behaviour of the utility is to check any DSLS server that has been configured to be used (e.g. checking the system default location for the `DSLicSrv.txt` license client configuration file). However, the following command line options are available:

- server** → Use this option to specify the name of the server machine and the port number on which the DSLS software is running; for example `dsIs_stat -server jupiter:4085`
- usage** → Use this option to show license usage details for individual users in the output

An example output of the utility is found below:

Server: xxx

```
Name           : xxx
Port           : 4085
Status         : Running
Type           : Standalone
Version        : 6.422.0
OS Name        : Windows 10
OS Version     : 10.0
OS Arch        : amd64
Computer Id    : xxx
```

Licenses:

+ Feature	Version	Model	Type	Number
InUse Expires	Server Name	Customer ID		
XFE	0	ConcurrentUser	STD	1
0 16-Oct-2020 01:59:00	xxxx	xxx		
XFE	0	Token	STD	10
0 11-Dec-2020 00:59:00	xxxx	xxx		
XFG	0	ConcurrentUser	STD	1
1 16-Oct-2020 01:59:00	xxxx	xxx		
XFG	0	ConcurrentUser	STD	10
0 11-Dec-2020 00:59:00	xxxx	xxx		

10.2 Domain partition optimization

The domain generator (generateDomain3d executable) can divide the fluid domain into several partitions for MPI computations. By default, the domain generator only requires the number of partitions and runs with no optimization of the load balance. However, for advanced users, several options are available to allow a fine control of the partition optimization process in order to improve the distributed load balance.

⚠ Please note: The **default** domain partition is **strongly recommended**, and the use of the domain partition optimization is available for advanced users only and may deeply affect the scalability performance if it is not employed correctly.

Stage 1

Options available for this stage with default values:

```
-maxAllowedImbalance=0
-nPruneIterations=100
```

Pruning of the small partitions or segments in the Hilbert curve. All the segments are enumerated and their lengths put in a list which is sorted. The resulted list is divided in `nPruneIterations` parts and we attempt to remove all the segments with sizes smaller than the considered length `Lc`. If this results in an imbalance smaller than `maxAllowedImbalance`, the segments with lengths smaller than `Lc` are discarded from the partition.

⚠ Please note: When one of these options is set to 0 (default value), the Stage 1 will be disabled altogether.

Example:

```
Segments: 100 100 20 19 15 15 5 5 5 5 2 2 2 2 2 1 1 1 1 1 2 5 1 2 5 1
List of lengths: 1 2 5 15 19 20 100
Assuming nPruneIterations=3, lengths to consider: 5, 19, 100
```

```
Assuming maxAllowedImbalance=3:
Attempt at removing smaller than 5, resulting imbalance ->1
Attempt at removing smaller than 19, resulting imbalance ->10. End.
```

```
Remove all segments smaller than 5.
```

Stage 2

Options for this stage with default values:

```
-nSAIterations=0
```

Optimization of the resulting list after pruning, we employ a simulated annealing-like algorithm, with a given number of iterations.

⚠ Please note: When this parameter is set to 0 (default value), the Stage 2 will be disabled altogether.

Additionally

Additionally, weights have been introduced for the different types of cells, which are weighted on a link-by-link basis. In principle these need to be adjusted, by default they are assumed to be 1, so the default partitioning behavior should be recovered with `-nPruneIterations=0` and `-nSAIterations=0`.

10.2 Domain partition optimization

The weights are the following:

- Normal links:
`-weightNormal=1`
- Links that see a wall:
`-weightWall=1`
- Links that see a lower (coarser) level, in principle these require the most expensive interpolation and should have the highest weight:
`-weightLower=1`
- Links that see a higher (finer) level:
`-weightHigher=1`

Index

- 1 -

1-way Thermal 300

- A -

abaqus 200, 278
 Abaqus co-simulation 167
 Absorbing boundary conditions 159
 acoustics analysis 114
 Adams 271
 adaptive 178
 Adiabatic index 146
 advanced computation 57, 201
 distributed (MPI) 204
 serial 201
 advanced options 119
 acoustics analysis 114
 analysis settings 44
 animated geometry 312
 animation 252
 advanced 254
 basic 252
 application
 labs 66
 normal 66
 application mode 305
 arbitrary reference frame 168

- B -

back-face culling 82
 batch 315
 batch mode 315
 bins 238
 boundary condition 156, 157
 boundary conditions 152
 buffer zone length 178
 bulk viscosity 141

- C -

cables 170
 CGNS 311
 check holes 88
 heal 98
 child 162
 children 162
 Co simulation 271
 command lines 315
 computation 199
 progress 208
 resume computation 199
 run 199
 stop 209
 compute
 average fields 190
 markers 190
 rms 190
 standard deviation 190
 static pressure filters 190
 conjugate heat transfer 163
 conventions 12
 co-sim 200
 Cosimulation 271
 co-simulation 200, 271, 278, 284
 Coupled energy 110
 coupling 271, 292, 300
 CSS 200
 cutting plane 227

- D -

discard narrow regions 178
 domain structure 227

- E -

edit project 50
 editing mode 50
 edition mode 52
 engine 103
 advanced options 119
 analysis type 109
 flow models 105
 multiphase model 107

- engine 103
 - particle-based tracking 107
 - turbulence models 112
 - vof 107
 - volume of fluid 107
- engine-3d 315
- Entities 148
- entity 69, 148
 - behaviour 150
- environment 121
 - external acceleration laws 127
 - generic rectangular domain 122
 - gravitational potential 127
 - initial conditions 128
 - liquid regions 136
 - reference area 131
 - reference velocity 132
 - water channel 133
 - waves 134
 - wind tunnel 122
- executing XFlow 49
- expert 305
- export all 264
- export current data 264
- export post-processing 252
- external 109

- F -

- FEA analysis 292
- flow modelling 305
- flow models 105
- FMI standard 272
- format 311
- fourrier transform 114
- free surface 105
- FSI 292
- function viewer
 - functions 26

- G -

- Generate co-sim configuration file
 - scripts 200
- generateDomain3d 315
- Genrate launch scripts
 - scripts 200

- geometry 69, 150
 - apply boundary conditions to faces 161
 - back-face culling 82
 - boundary conditions 152
 - cables 170
 - check holes 88
 - create box 69
 - create cone 69
 - create cylinder 69
 - create line 69
 - create point 69
 - create prism 69
 - create sphere 69
 - create surface 69
 - create torus 69
 - delete 102
 - dimensions 87
 - duplicate 95, 96
 - export 73
 - geometrical properties 85
 - heal 98
 - import 73
 - information 83
 - joints 173
 - local axes 85
 - merge 97
 - mesh deflection 58
 - modification 89
 - normals 82
 - parametric hierarchy 86
 - preferences 58
 - rotation 91
 - scale 93
 - selection 76
 - show wires 58
 - split 97
 - STL 58
 - surface detection 58
 - symmetry 94
 - thermal boundary conditions 162
 - translation 89
 - visualisation mode 78
- geometry hierarchy 162
- graphic view 41
 - Anti-aliasing quality 62
 - Bloom quality 62
 - Enable Ambient occlusion 62
 - environment 59, 60, 61
 - Environment quality 62

- graphic view 41
 - grid 60
 - lights 59, 61
 - performance quality 62
 - rotation mode 60
 - Shadow resolution 62
 - Transparency quality 62
- graphical user interface 15
 - Existing project 18
 - function viewer 46
 - graphic view 41
 - GUI management 16
 - main menu 19
 - message view 35
 - project manager 18
 - project tree 24
 - shortcuts 17
 - Template project 18
 - time controls 48
 - toolbar 21
 - transform tool 48
 - trees management 34

- H -

- heal 98
- heat flux 162
- heat source 130
- Heat Transfer Coefficient 162
- hierarchies 162
- HTC 162

- I -

- import Adams simulation 271
- import post-processing 252
- interface thickness 146
- internal 109
- Isothermal 110

- J -

- joints 173

- L -

- labs mode 271, 306

- advanced computation 201
- coupled energy flow 309
- CPU 307
- execution mode 307
- fully compressible flow 311
- GPU 307
- passive scalar transport 118
- Spalart-Allmaras 309
- supersonic flow 308, 309
- virtual moving wall boundary 156
- lattice levels 227
- lattice structure 227
- layout 252
- legal notices 9
- LODI 159

- M -

- main menu 19
- markers 229
- mass flow inlet 156
- mass flow outlet 157
- material
 - estimation 144
 - heat transfer coefficient 144
 - HTC 144
 - U+ref 144
 - Y+ Reference 144
 - Y+ref 144
- materials 138
 - density 139
 - Material 1 139
 - molecular weight 139
 - name 139
 - specific heat capacity 145
 - speed of sound free surface 139
 - state equation 140
 - surface tension model 146
 - temperature 139
 - thermal conductivity 145
 - viscosity models 140
- mesh 33
- message view 35
- mobility 146
- mpi 315
- mpirun 315
- MSC Nastran 292
- multiphase 105

- N -

Nastran 292, 300
 near walls 178
 non-inertial 168
 Non-reflecting boundary conditions 159
 normals
 reorientate normals 83
 reverse orientation 83
 numerical speed of sound 114

- O -

object 69
 OpenFSI 292
 output 311

- P -

parent 162
 post-pro 252
 post-processing 213
 analysis settings 44
 animation 252
 averaged 216
 camera 248
 custom fields 247
 cutting planes 222
 data 216
 data plot lines 239
 entities 247
 export cutting plane distribution 264
 export data 264
 export data to raw format 264
 export isosurface 264
 export surface data 264
 FFT 259
 function viewer 259
 geometries 247
 geometry 247
 geometry vertices 264
 instantaneous 216
 interpolation 217
 isosurfaces 230
 near-wall lattice nodes 264
 point distribution 264
 post-processing entities 247

post-processing geometries 247
 PSD 259
 RMS 216
 sensors 240
 signal filter 259
 SPL 259
 standard deviation 216
 stream tracers 231
 surface integrals 242
 views 249
 visualisation fields 250
 post-processing automation 252
 post-processing mode 50, 51
 power spectral density (PSD) 114
 preferences
 application mode 56
 engine 56
 environment 56
 graphic view 56
 lights 56
 project tree 56
 pressure coefficient 218
 pressure inlet 156
 pressure outlet 157
 Pressure waves 159
 process manager 210
 project tree 24
 engine 103
 entity 147
 environment 121
 font size and color 65
 geometry 147
 materials 138
 post-processing 213
 simulation 174

- R -

Radiation 110
 Reference density 146
 Reference pressure 146
 refinement 178
 refinement transition level 178
 Reflections 159
 resolution 178
 resolved scale 178
 restore project 50

resume computation 190
roughness 153

- S -

save 252
seed point 178
Segregated energy 110
Setup mode 50, 51
shortcuts 17
signal filters 114
simpack 200, 276, 284
simulation
 resolution 178
simulation setup 103
 engine 103
 entity 147
 environment 121
 geometry 147
 materials 138
 simulation 174
simulation time 175
single phase 105
sloshing 168
solid conduction 162, 163
sound pressure level (SPL) 114
stability parameter 36
store data
 fields to save 190
 probes 190
stream tracers 231
 discrete phase model 236
 passive 233
structural coupling 167
structure 292
Supersonic 110
surface tension model 146

- T -

target resolved scale
 curvature 178
 shape 178
 wake 178
thermal 300
thermal boundary conditions 162
thermodynamic speed of sound 114

threshold 178
time step 175
toolbar 21
total pressure inlet 156
total pressure outlet 157
trademarks 9
transform tool 48
turbulence models
 automatic 112
 dynamic Smagorinsky model 112
 Smagorinsky model 112
 wall-adapting local eddy model 112
two-ways coupling 278, 284

- U -

unstructured mesh 33
unstructuredmesh 33
user defined variables 30
user-defined refinement
 Annular 178
 box 178
 Cylinder 178
 rectangular 178
 sphere 178
 Tubular 178
user-defined variables 30

- V -

velocity inlet 156
velocity outlet 157
viscosity models
 Newtonian 140
 Newtonian powerlaw 141
 Newtonian Sutherland 141
 non-Newtonian 140
 non-Newtonian Carreau 142
 non-Newtonian Cross 142
 non-Newtonian Herschel-Bulkley 142
 non-Newtonian powerlaw 142
 non-Newtonian user defined 142
visualisation fields 250
visualisation material
 glass 79
 metal 79
 plastic 79

- visualisation material
 - rubber 79
 - shadows only 79
- visualisation mode
 - bounding box 81
 - mesh 81
 - shading 81
 - wireframe 81
- volume heat 163
- volumetric heat source 130
- voxels 227
- vtk 33
- vtu 33

- W -

- W/m³ 130
- wall boundary condition 153
- wall function 153
- window functions 114

- X -

- XFlow files 53
- XFlow native 311

Our **3DEXPERIENCE®** platform powers our brand applications, serving 11 industries, and provides a rich portfolio of industry solution experiences.

Dassault Systèmes, the **3DEXPERIENCE** Company, is a catalyst for human progress. We provide business and people with collaborative virtual environments to imagine sustainable innovations. By creating 'virtual experience twins' of the real world with our **3DEXPERIENCE** platform and applications, our customers push the boundaries of innovation, learning and production.

Dassault Systèmes' 20,000 employees are bringing value to more than 270,000 customers of all sizes, in all industries, in more than 140 countries. For more information, visit www.3ds.com.



3DEXPERIENCE®