



3DEXPERIENCE®

XFlow 2022 TUTORIAL GUIDE

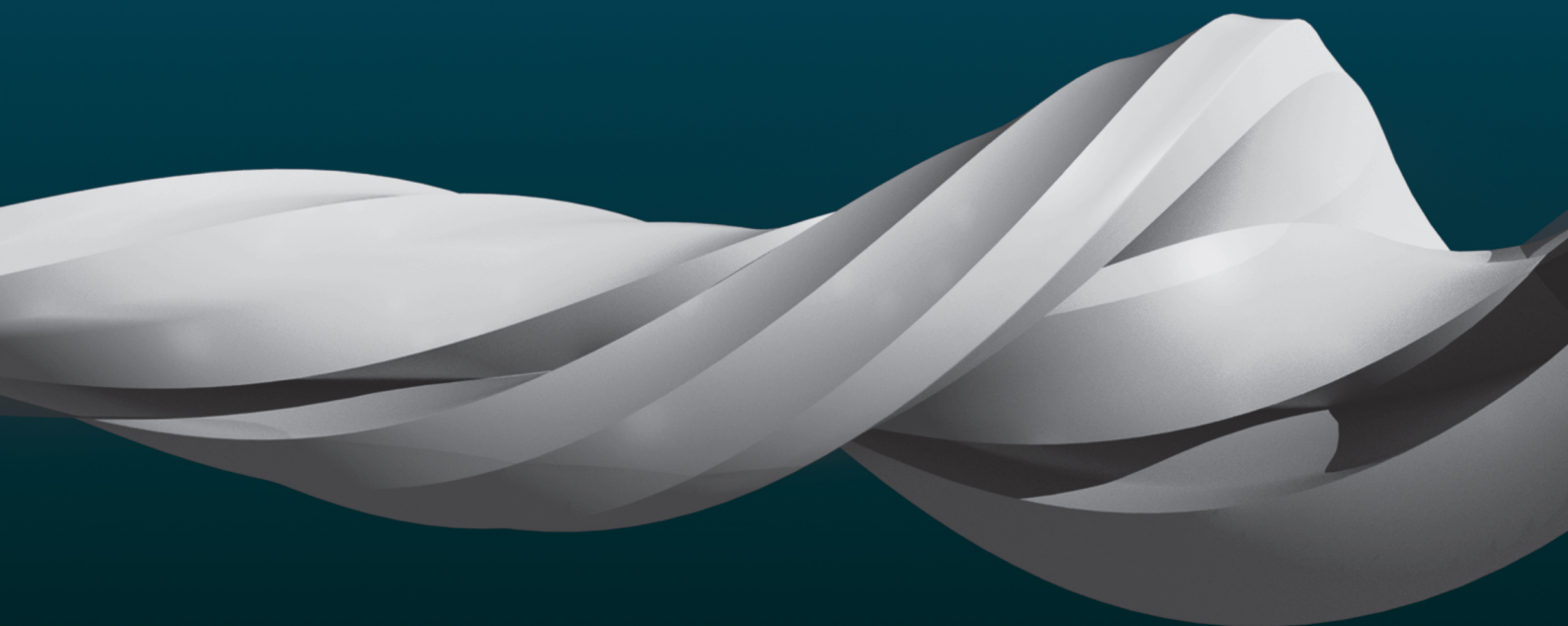


Table of Contents

Legal Notices	5
Using this guide	6
Conventions	6
Tutorial 01 - Flow around a NACA 0012 airfoil	9
Step 0: Execute XFlow	10
Step 1: Interface and geometry	11
Step 2: Problem setup	17
Step 3: Run and workflow	23
Step 4: Post-processing	31
Step 5: Refine the resolution	41
Step 6: Moving NACA - Enforced motion	50
Step 7: Moving NACA - Rigid body dynamics	56
Tutorial 02 - Vehicle aerodynamics	60
Step 1: Import geometry	61
Step 2: Problem setup	64
Step 3: Run	70
Step 4: Post-processing	72
Tutorial 03 - Advanced post processing	87
Advanced post-processing	88
Tutorial 04 - Dam break	97
Step 1: Problem setup	98
Step 2: Post-processing	100
Tutorial 05 - Breaking waves	104
Step 1: Problem setup	105
Step 2: Post-processing	109
Step 3: Gravitational potential	110
Step 4: Porous volume	113
Tutorial 06 - Ball check valve	116

Step 1: No damping case	117
Step 2: Under-damping case	126
Step 3: Critical-damping case	127
Step 4: Over-damping case	129
Tutorial 07 - Wind turbine	131
Step 1: Geometry healing	132
Step 2: Enforced behaviour - Case setup	135
Step 3: Enforced behaviour - Post-processing	140
Step 4: Rigid body dynamics behaviour	143
Tutorial 08 - Heat transfer	147
Step 1: Problem setup	148
Step 2: Post-processing	154
Tutorial 09 - Radiation	160
Step 1: Problem setup	160
Step 2: Post-processing	162
Tutorial 10 - Cyclone flow	167
Step 1: Problem setup	168
Step 2: Post-processing	172
Step 3: Stream tracers	176
Tutorial 11 - FMI standard co-simulation: Pendulum	185
Step 1: Problem setup - XFlow	186
Step 2: Set FMI standard in XFlow	192
Step 3: Problem setup - Dymola/OpenModelica	193
Step 3.1: OpenModelica	193
Step 3.2: Dymola	198
Step 4: Execution of Co-simulation	203
Step 5: Post-processing	207
Tutorial 12 - Helmholtz resonator	209
Step 1: Problem setup	210
Step 2: Post-processing	213
Tutorial 13 - Rising bubble	219
Step 1: Problem setup	220
Step 2: Post-processing	224

Tutorial 14 - Crown splashing	229
Step1: Problem setup	230
Step2: Post-processing	234
Tutorial 15 - Abaqus co-simulation: Flexible bar Dam-break	239
Step 1: Problem setup - Abaqus/Cae	239
Step 2: Problem setup - XFlow	252
Step 3: Execution of Co-simulation	256
Step 4: Post processing	258
Tutorial 16 - Simpack co-simulation: Powertrain Lubrication	263
Step 1: Problem setup - Simpack	264
Step 2: Problem setup - XFlow	265
Step 3: Execution of Co-simulation	271
Step 4: Post processing	272

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.

Using this guide

This guide contains several tutorials that illustrate how to use XFlow in different types of problems:

- [Tutorial 01](#) - Flow around a NACA 0012 airfoil
- [Tutorial 02](#) - Vehicle aerodynamics
- [Tutorial 03](#) - Advanced post processing
- [Tutorial 04](#) - Dam break
- [Tutorial 05](#) - Breaking waves
- [Tutorial 06](#) - Ball check valve
- [Tutorial 07](#) - Wind turbine
- [Tutorial 08](#) - Heat transfer
- [Tutorial 09](#) - Radiation
- [Tutorial 10](#) - Cyclone flow
- [Tutorial 11](#) - FMI standard co-simulation: OpenModelica Pendulum
- [Tutorial 12](#) - Helmholtz resonator
- [Tutorial 13](#) - Rising bubble
- [Tutorial 14](#) - Crown splashing

The user can either create a new project and follow the steps described in the tutorials (recommended), or load the project file `.xfp` provided in the project data files from the `XFlow[version]_TutorialsPack.zip` folder.

Please take into account that the tutorials presented in this guide have been set to have a short computation time. Usually, a finer lattice structure is needed to obtain more accurate results comparable to available literature data for validation purposes.

Conventions

Several typographical conventions are used in this guide:

- Menu options are indicated in **orange**.
- Names of windows are in *italics*.
- Items and options in the project tree are indicated in **Verdana font**.
- [Links](#) are underlined in blue colour.
- Routes to files are indicated in `courier new` font.
- Keys are indicated in **bold blue**.

- Cascading menus are represented as: **Menu1 > Menu2 > Menu3**. This means that in Menu1, click on Menu2. Then, in the Menu2 that comes up, click on Menu3 and so on.
- Additional explanations and recommendations are enclosed in a message box.

 **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 when doing an specific task.

Units

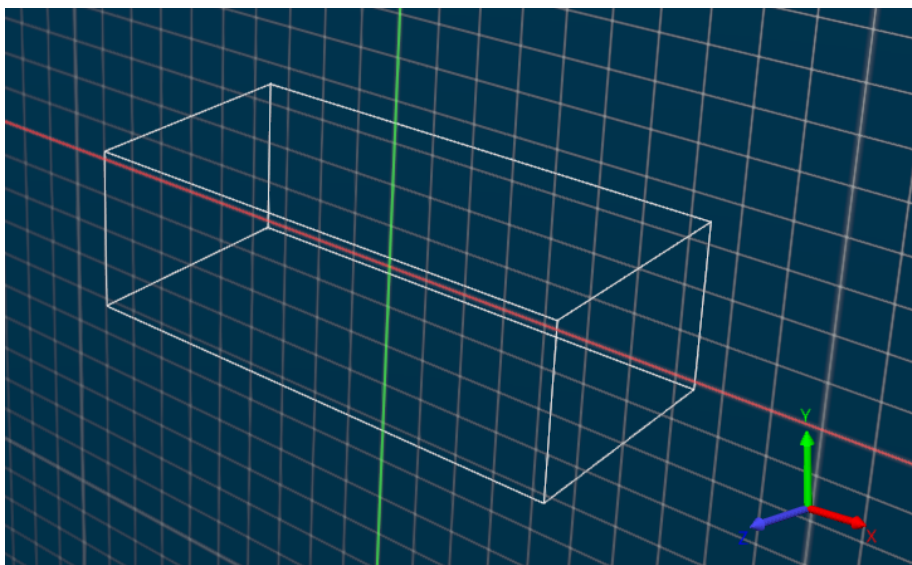
All units are in the international system (SI).

Measure	Symbol	Units
Mass		kg
Length		m
Time	t	s
Velocity	v_x, v_y, v_z	m/s
Pressure	p	Pa
Temperature	θ	Kelvin
Acceleration		m/s ²
Density	ρ	Kg/m ³
Viscosity	μ	Pa-s
Thermal Conductivity	k	W/m-K
Specific Heat	C_p	J/kg-K
Angle		degrees
Angular velocity		rad/s

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

Coordinate system

Special attention needs to be paid to the coordinate system. XFlow's convention is that the Y-axis is assumed to be vertical. Thus, the user may have to rotate the geometry when importing it from a CAD software. This can be done either in the CAD prior to the import operation or in XFlow.

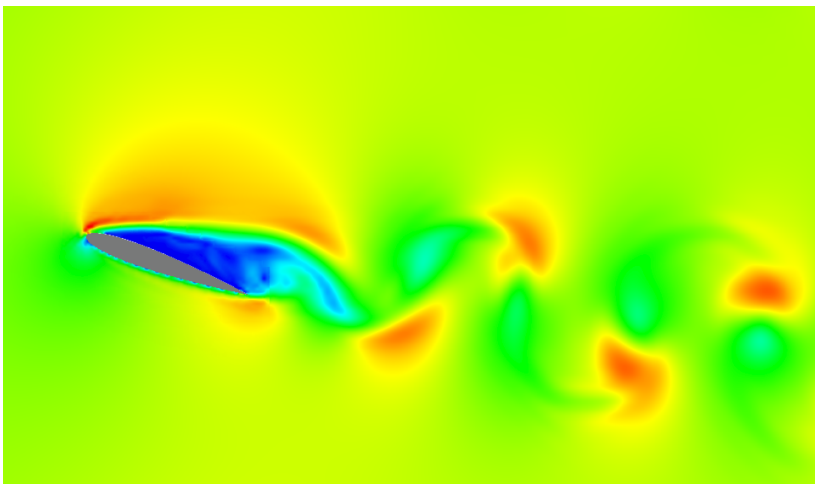


Tutorial 01 - Flow around a NACA 0012 airfoil

This first tutorial illustrates the simulation setup, execution, and post-processing of the two-dimensional flow around a NACA 0012 airfoil at Reynolds number $Re = 10^4$ based on the chord length of the airfoil. The angle-of-attack of the NACA will be 20 degrees.

It is assumed that the user has no experience using XFlow, so every step will be described in detail.

⚠ Please note: The aim of this first tutorial is to introduce and illustrate the basic XFlow features to the user with a NACA 0012 profile. The setup used in this tutorial cannot be considered as a best practice or a guide to perform accurate validations of the NACA profiles. For any validation purpose, please refer to the Validation Guide of XFlow where some airfoil profiles are validated and their setup are explained.



This tutorial shows how to:

- Create a project
- Create a simple geometry
- Use XFlow interface
- Setup a virtual wind tunnel
- Set the fluid properties and boundary conditions
- Launch a simulation
- Load simulation data
- Do basic post-processing of the results
- Refine the resolution

- Set a simple moving geometry
- Set a simple fluid-structure interaction

Contents

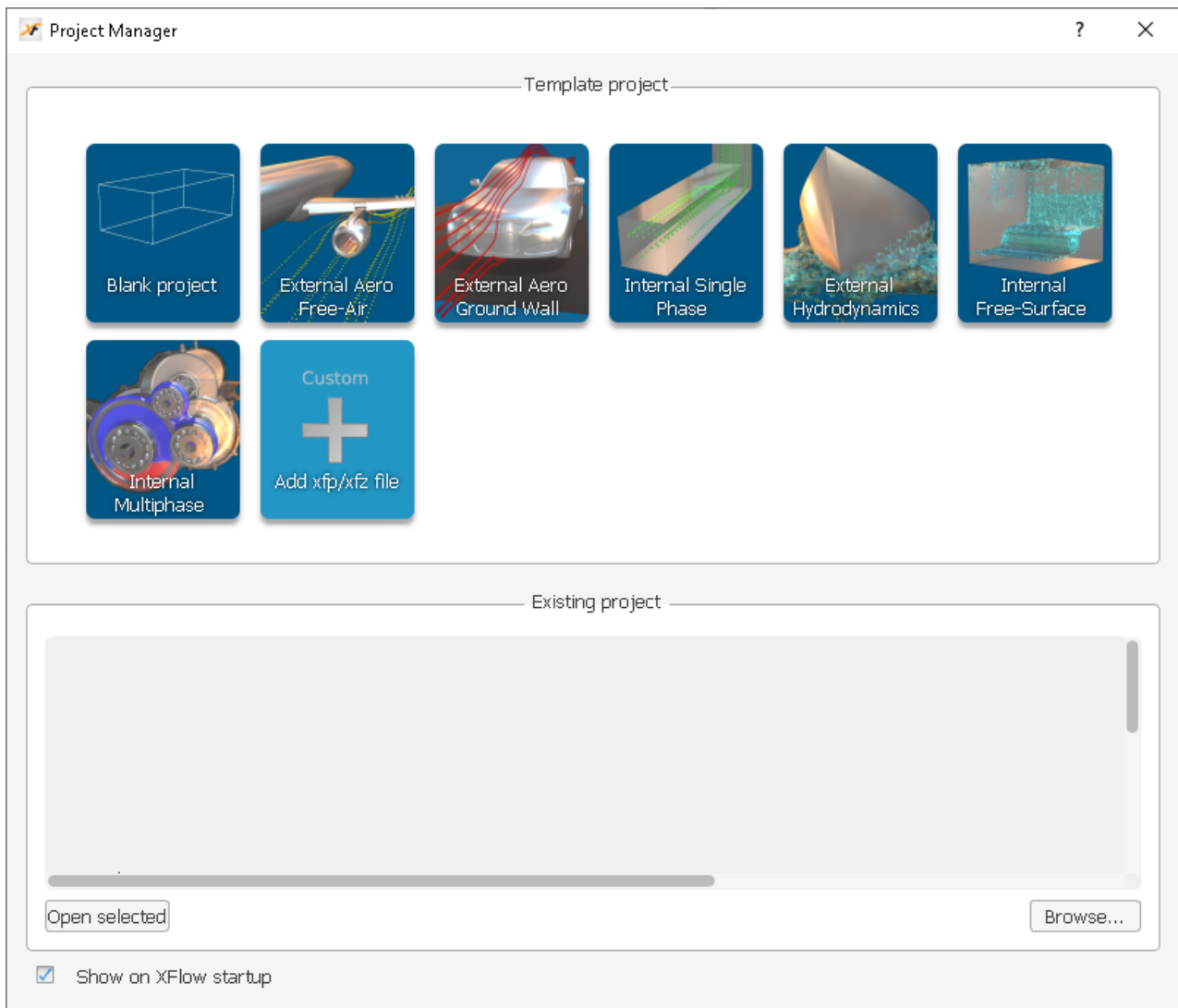
- [Step 0: Execute XFlow](#)
- [Step 1: Create geometry](#)
- [Step 2: Problem setup](#)
- [Step 3: Run](#)
- [Step 4: Post-processing](#)
- [Step 5: Refine the resolution](#)
- [Step 6: Moving NACA - Enforced motion](#)
- [Step 7: Moving NACA - Rigid body dynamics](#)

Step 0: Execute XFlow

Execute XFlow through the direct link in your desktop or by double-clicking the `XFlow.exe` file located in the installation folder. The application displays the *Graphical User Interface* with the default layout and the *Project Manager* window.

 **Please note:** Linux users should execute `xflow-gui`

In the *Project Manager*, select Blank project to create a new case. As this tutorial aims to describe in detail all the steps throughout the setup, the External_Aero Free-Air template will not be selected.



Project Manager window

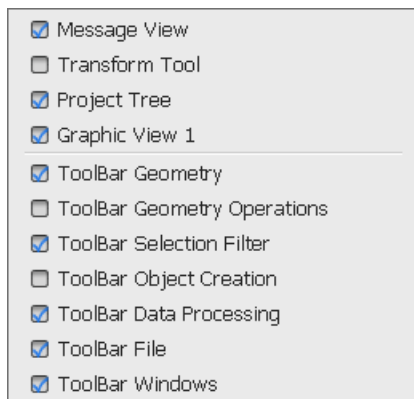
⚠ Please note: The options displayed in this Tutorial assume you are executing XFlow in **Normal** mode. Please check this by going to Preferences < Application < Application mode

Step 1: Interface and geometry

1.1 Create a NACA 0012 profile

Create a NACA 0012 airfoil with chord length of 1 m and a width of 1 m, by means of: **Main menu > Geometry > Create object > NACA (4 digit)**. Alternatively you can access this menu via the *Toolbar Object Creation* that can be shown doing a right-click on the existing toolbars:

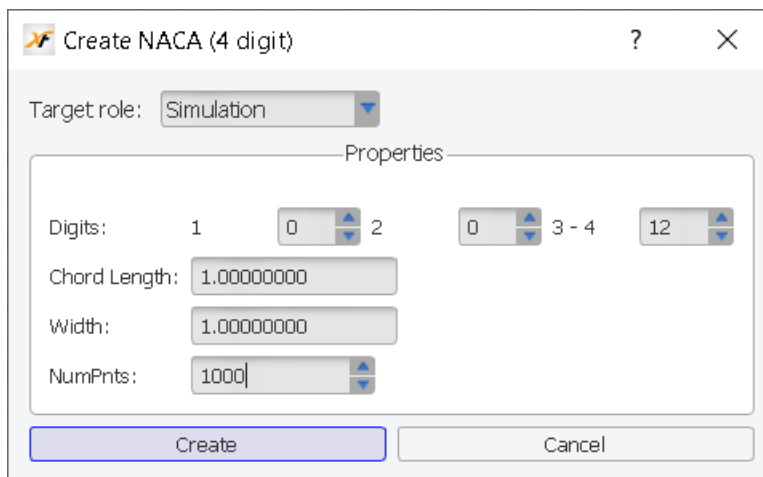
Step 1: Interface and geometry



Toolbars management dialog

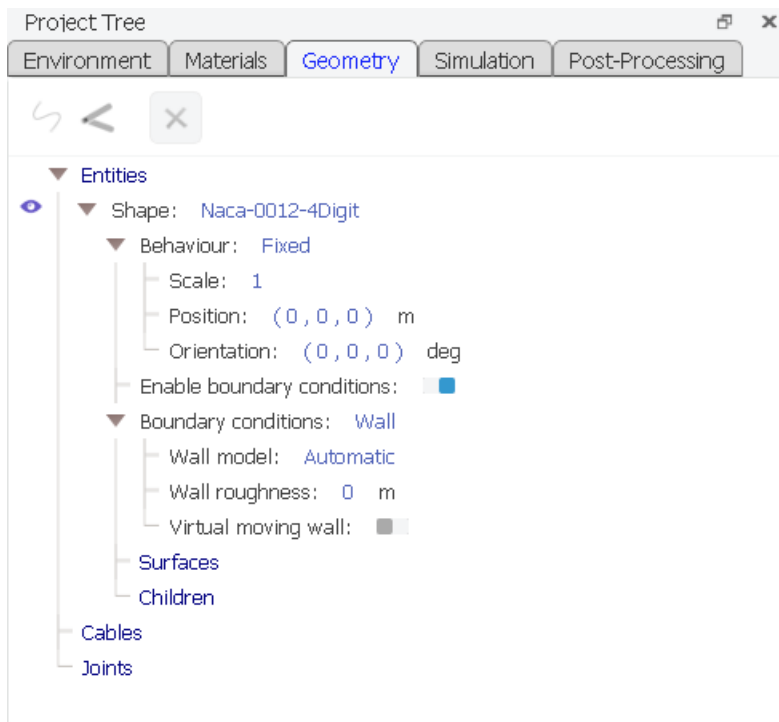
The object creation toolbar will be shown and you can click on the NACA creation icon  .

Introduce the geometrical data of the NACA airfoil as in the following dialogue box ([Units](#) in SI):



NACA creation dialog

The NACA 0012 airfoil appears as a **Shape** in **Project Tree > Geometry > Entities**:

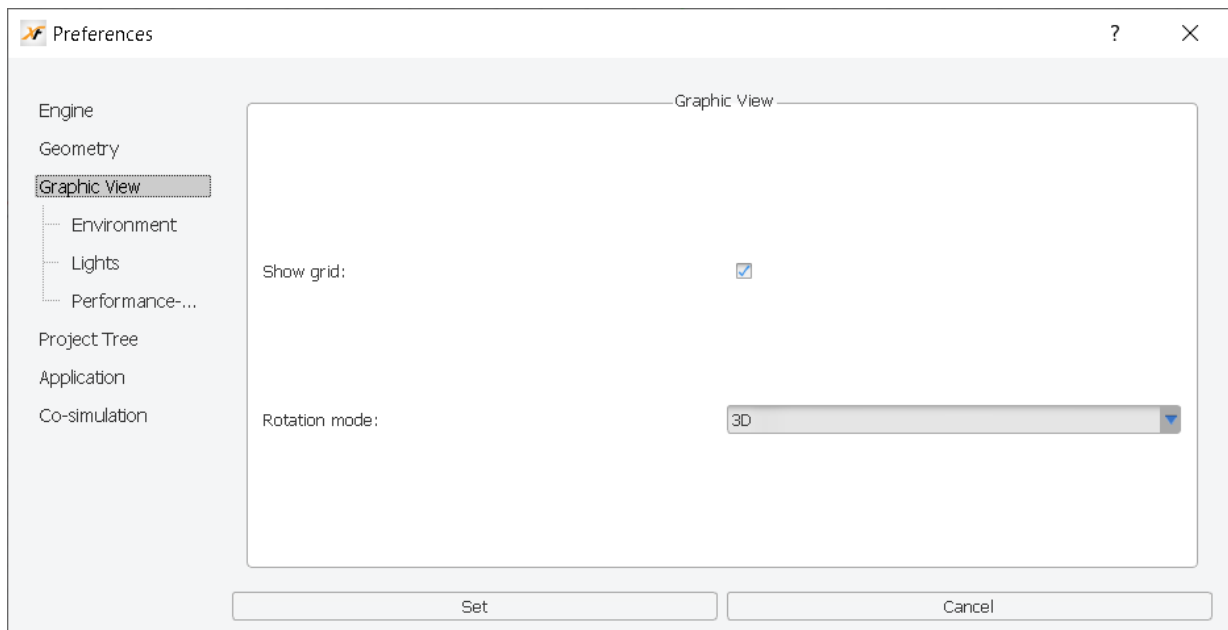


Project tree with NACA

1.2 Check the position of the airfoil with the help of the grid

Go to **Main menu > Options > Preferences**, or press .

Enable the grid in: **Graphic View > Show grid**



Preferences > Graphic View > Show grid

The same option can be accessed with the following highlighted icon on the Graphic View:

Step 1: Interface and geometry



The red and green lines indicate the X and Y axis, respectively.

The scale of the grid is displayed at the top left-hand corner of the *Graphic View* window.



1.3 Navigate in the Graphic View

Change the view by clicking the following toolbar icons:



Navigate in the *Graphic View* window through the following actions:

- Zoom: Move the **mouse wheel** to zoom step by step, or drag the mouse while pressing **Alt** and the **right mouse button** for a continuous zoom.
- Translate view: Drag the mouse while pressing **Alt** and the **middle mouse button** to pan the view.
- Rotate view: Drag the mouse while pressing **Alt** and the **left mouse button** to rotate the view.
- Change center of rotation: Press **middle mouse button** to change the center of rotation to a new

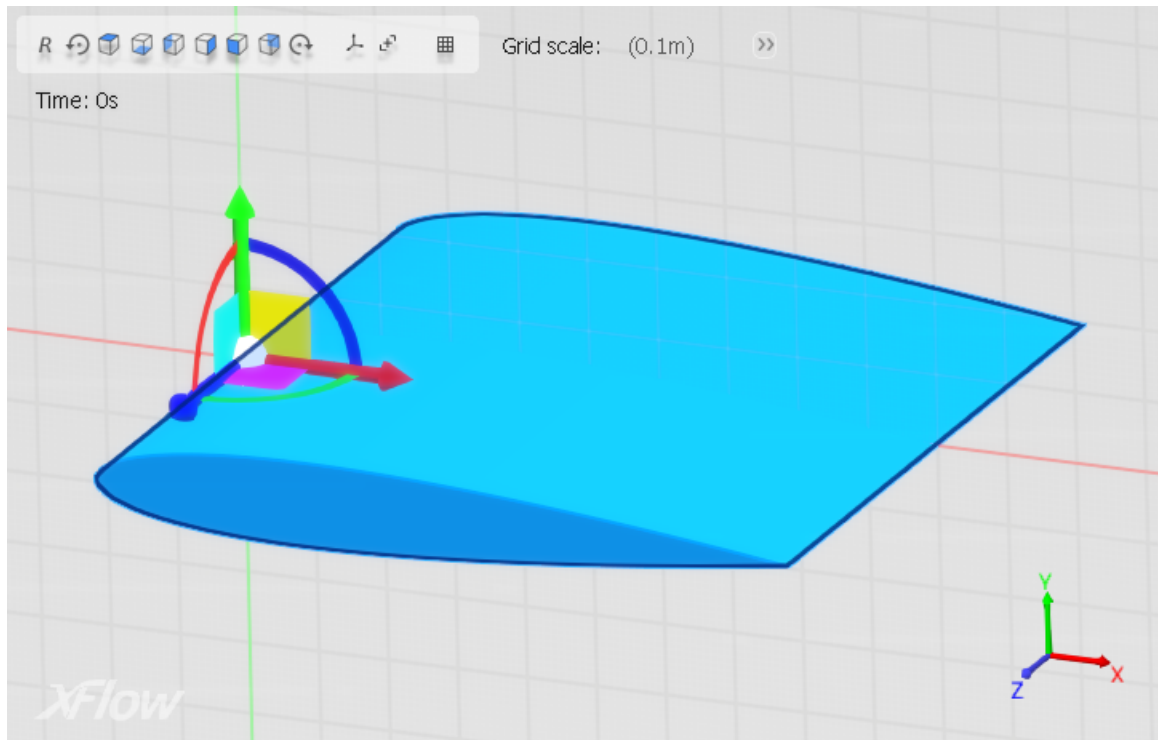
position. New center of rotation is marked with the rotation sphere 

⚠ Please note: 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.

1.4 Select the airfoil

Select the NACA geometry either by:

- selecting the *View only* mode  in the *Toolbar Selection Filter*, and **clicking** either on the NACA in the *Graphic View* or on the word **Shape** in **Project Tree > Geometry > Entities**. Once the object is selected with the View only mode, it is highlighted in blue in the *Graphic View*.
- selecting the *Object filter* mode  in the *Toolbar Selection Filter*, and **clicking** either on the NACA in the *Graphic View* or on the word **Shape** in **Project Tree > Geometry > Entities**. Once the object is selected with the Object filter mode, it is highlighted in blue and the object Gizmos is automatically shown in the *Graphic View*.



Selected NACA profile - Gizmos shown

Gizmos allow the user to translate, rotate and scale the geometry. To familiarize yourself with the Gizmos set the Object filter mode and select the geometry, and perform the following actions:

- **click** on any of the Gizmos axes - the translation dialogue box appears - and enter $X = 3 \text{ m}$, $Y = 2 \text{ m}$, $Z = 1 \text{ m}$. Observe the new position of the NACA.
- **click** on any of the Gizmos arcs - the rotation dialogue box appears - and enter $X = 30^\circ$, $Y = 0^\circ$, $Z = 0^\circ$. Observe the new orientation of the NACA.
- **click** on any the Gizmos axes center - the scale dialogue box appears - and enter 2. Observe the new size of the NACA (twice the original).

Select the NACA and press **Delete** in order to delete the modified geometry, or **right mouse button** in the *Graphic View* > **Graphic View menu > Remove selected**. Create the NACA again as indicated in [Section 1.1](#)

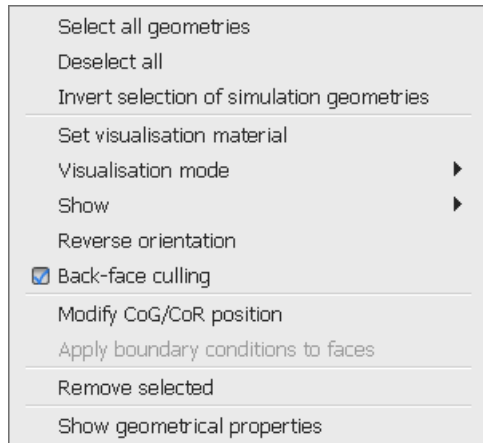
1.5 Check the geometrical properties of the airfoil

Select the NACA and measure its dimensions: **Main menu > Geometry > Dimensions**, or  (*Toolbar geometry*).

Step 1: Interface and geometry

Please note, that the user may need to zoom, translate or rotate the view to see the numbers clearly (see [navigation](#)).

With the airfoil still selected, click on  again to hide the dimensions and click the **right mouse button** in the *Graphic View* window to pop-up the contextual menu:



Geometry contextual menu

In this menu, select **Set visualisation material** to change the colour of the airfoil surface:

Select geometry > **right mouse button** in the *Graphic View* > **Graphic View menu** > **Set visualisation material** > **Colour**

In the same menu, select the option **Visualisation mode** and visualise the NACA as shading, wireframe, bounding box and mesh:

Select geometry > **right mouse button** in the *Graphic View* > **Graphic View menu** > **Visualisation mode** > **Shading**

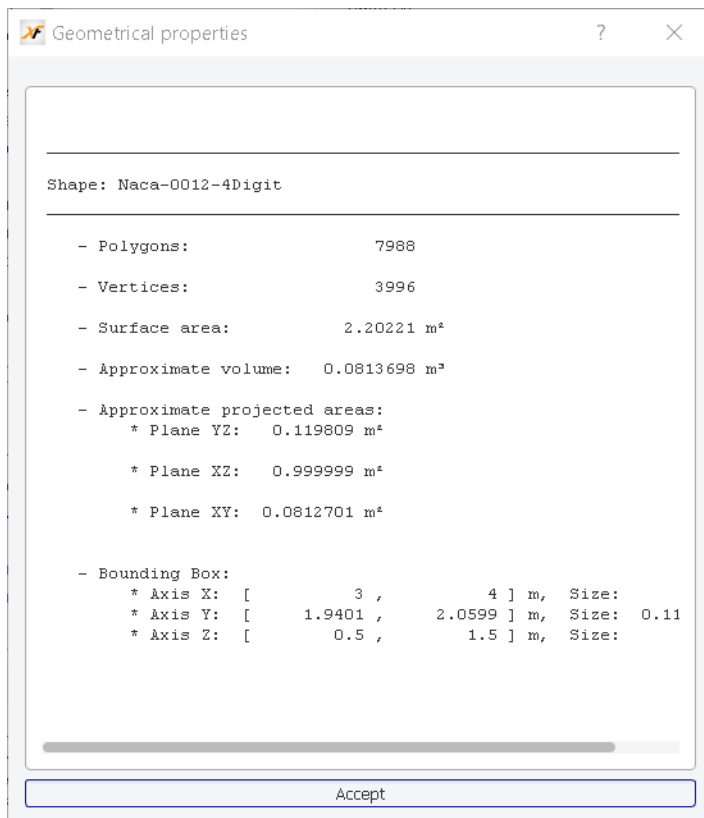
Select geometry > **right mouse button** in the *Graphic View* > **Graphic View menu** > **Visualisation mode** > **Mesh**

Select geometry > **right mouse button** in the *Graphic View* > **Graphic View menu** > **Visualisation mode** > **Wireframe**

Select geometry > **right mouse button** in the *Graphic View* > **Graphic View menu** > **Visualisation mode** > **Bounding box**

Now, select the option **Show geometrical properties**, to get a report of the main geometrical properties of the NACA:

Select geometry > **right mouse button** in the *Graphic View* > **Graphic View menu** > **Show geometrical properties**. Press **Accept** to close this window.



Geometrical properties report of the NACA profile

The surface of the NACA is automatically tessellated by XFlow when it is created. The surface mesh properties (Polygons and Vertices) are displayed as well as other geometrical properties (area, volume, etc.). The NACA geometry is generated as a native XFlow mesh format (.nfb) by default.

For parametric geometries (STEP, IGES, etc.), the geometry tessellation is generated by XFlow automatically and depends on the parameter:

Main menu > Options > Preferences > Geometry: Mesh deflection

which varies between 0 and 2. The larger the mesh deflection, the coarser the tessellation will be.

⚠ Please note: There is no Undo option. Please, save the project frequently: **Main menu > File > Save project** or click the toolbar icon 

Step 2: Problem setup

For external aerodynamic applications, such as the object of this tutorial, XFlow features a Virtual Wind Tunnel module that allows to accelerate the setup process.

The setup of the problem is done in the following sections of the *Project Tree*:

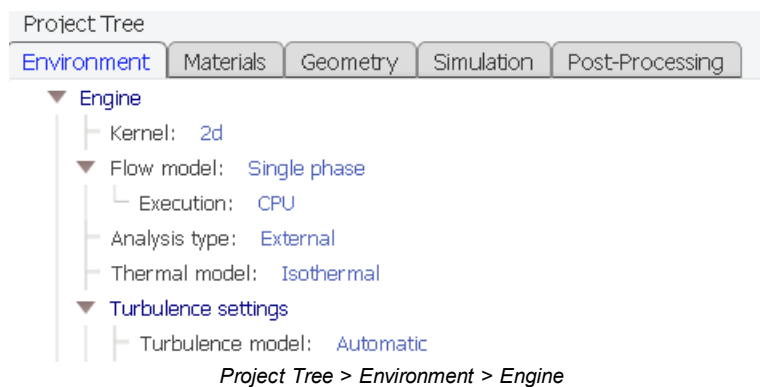
Step 2: Problem setup

- Environment > [Engine](#)
- Environment > [Environment](#)
- [Materials](#)
- [Geometry](#)
- [Simulation](#)

2.1 Engine settings

Configure the section **Project Tree > Environment > Engine >** as follows:

- (a) **Kernel** : **2d** (click on the line to unfold options)
- (b) **Flow model**: **Single phase**
- (c) **Analysis type**: **External**
- (d) **Thermal model**: **Isothermal**
- (e) **Turbulence settings**:
 - **Turbulence model**: **Automatic**

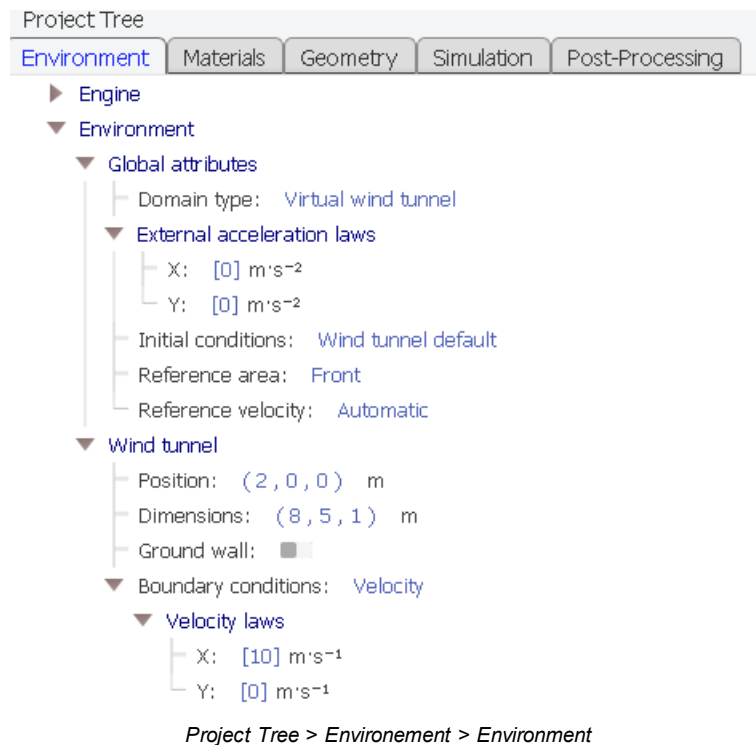


2.2 Environment settings

Configure the section **Project Tree > Environment > Environment >** as follows:

- (a) **Global attributes > Domain type**: **Virtual wind tunnel**
- (b) **Global attributes > Ext. acceleration laws**: leave it to zero
- (c) **Global attributes > Initial conditions**: **Wind tunnel default**
- (d) **Global attributes > Reference area**: **Front**
- (e) **Global attributes > Reference velocity**: **Automatic**
- (f) **Wind tunnel > Position**: move the wind tunnel 2 meters in the +X direction
- (g) **Wind tunnel > Dimensions**: 8 meters length (X), 5 meters height (Y) and 1 meter width (Z)
- (h) **Wind tunnel > Ground wall**: **Off**. The switches **Off / On** are represented respectively by icons  / 
- (i) **Wind tunnel > Boundary conditions**: **Velocity**

- (j) Unfold **Wind tunnel** > **Boundary conditions** > **Velocity law** and set $10 \text{ m}\cdot\text{s}^{-1}$ in +X direction. This boundary condition will be applied on the -X boundary of the wind tunnel (inlet).



Please note: The flow in the wind tunnel is by default assumed to move from -X (inlet) to +X (outlet).

Please note: The initial condition **Wind tunnel default** allows to initialise the flow using the same wind tunnel conditions.

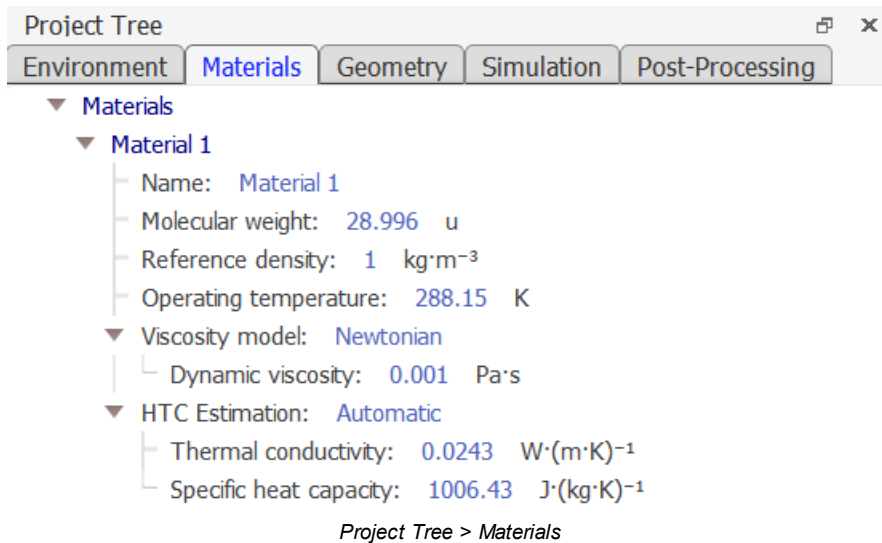
2.3 Materials settings

By default in single phase the fluid material is called *Material 1*, which is initialized with the air thermophysical properties. To get a $Re = 10^4$, given a fluid velocity of $10 \text{ m}\cdot\text{s}^{-1}$ and an airfoil chord length of 1 m, the *Material 1* properties have to be modified as:

- (a) **Project Tree** > **Materials** > **Fluid** > **Density** : $1 \text{ kg}\cdot\text{m}^{-3}$
- (b) **Project Tree** > **Materials** > **Fluid** > **Viscosity model**: **Newtonian**
- (c) **Project Tree** > **Materials** > **Fluid** > **Viscosity model** > **Dynamic viscosity**: **0.001 Pa·s**
- (d) Leave default values for **Project Tree** > **Materials** > **Fluid** > **HTC Estimation** > **Thermal conductivity** and **Specific heat capacity**. The thermal model is set to **Isothermal**, hence the specific heat capacity and thermal conductivity are only used to output the heat transfer coefficient

Step 2: Problem setup

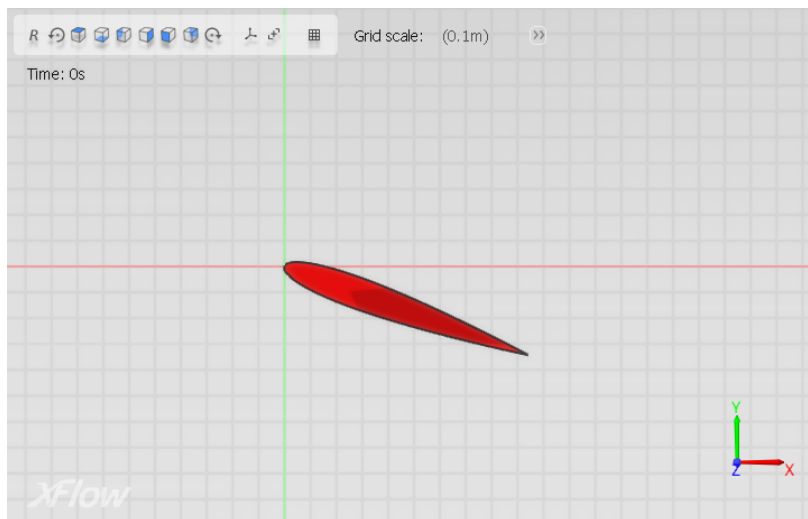
based on the skin friction. These two inputs will not affect the flow field.



2.4 Geometry settings

The geometry (Naca-0012-4Digit) has been previously [created](#). In **Project Tree > Geometry**, its behaviour and boundary conditions can be defined as follows:

- (a) **Project Tree > Geometry > Entities > Shape:** Naca-0012-4Digit > **Behaviour:** Fixed
- (b) Leave the Scale (1) and Position (0,0,0) by default
- (c) Change the Orientation of the NACA profile to -20 deg around the Z-axis and check its orientation in the *Graphic View 1*. The airfoil will be simulated with an angle of attack of 20 degrees and should now be inclined as per the following picture.



NACA 0012 at 20° angle-of-attack in the XFlow Virtual Wind Tunnel

- (d) **Project Tree > Geometry > Entities > Enable boundary conditions:** Leave the default value (activated).

- (e) **Project Tree > Geometry > Entities > Shape: Naca-0012-4Digit > Boundary conditions: Wall**, with Automatic wall model and zero roughness



Project Tree > Geometry > Entities

2.5 Simulation settings

Configure the section **Project Tree > Simulation >** as follows:

- (a) **Time > Simulation time: 1 s**
- (b) **Time > Time step mode: Fixed automatic** (i.e. constant time step automatically calculated by XFlow)
- (c) **Time > Time step mode > Courant: 1**
- (d) **Lattice discretization > Resolved scale: 0.04 m**
- (e) **Lattice discretization > Refinement algorithm: Disabled** (i.e. uniform resolution)
- (f) **Store data > Frames frequency: 100 Hz** which means that a total of 100 frames will be saved for 1 second of simulation time. Note the timeline now adapted to the new output frequency.
- (g) **Store data > Save frames from: 0 s** which means that the frames will be stored from the start of the simulation.
- (h) **Store data > Numerical data frequency: Solver time step**, which means any curve plotted in the *Function Viewer* will be updated at the frequency of the solver steps.
- (i) **Store data > Save averaged fields: Off**

Step 2: Problem setup

- (j) **Store data > Save resume file: Off.** In case you need to stop and resume your simulation you can switch it to On, however this consumes more hard disk space.
- (k) **Store data > Compute makers: Off**
- (l) **Store data > Output format: VTK Off**
- (m) **Store data > Fields to save:** Leave the default fields selected.



At this point the setup has been finished and the computation can be launched.

 **Tip:** You can directly load the setup of this problem from the project file `naca.xfp` in **Main menu > File > Load project** or **Open an existing project** in the *Project Manager* window that appears when executing XFlow.

Step 3: Run and workflow

3.1 Save project

Save the project before running the computation: **Main menu > File > Save project**, or  in *Toolbar File*.

XFlow project files have two extension types: `.xfp` and `.xfz`. Default project file will be saved in `.xfp` which is an uncompressed format, while it can be changed to `.xfz` in order to save a compressed archive of the project (project file + geometries). In this example the project file is `naca.xfp` and will come together with a `naca.lay` file (graphical user interface layout) and a `Naca-0012-4Digits.nfb` file which is the NACA geometry file in XFlow native format.



Uncompressed file format `.xfp` (left) and compressed file format `.xfz` (right)

 **Tip:** `.xfp` format is more efficient for performance and avoids to duplicate geometry files, however `.xfz` format is convenient to send the project to a collaborator or to archive the project since it includes all files in one archive.

The default saving format can be changed in **Main menu > Options > Preferences > Application > Default project format**, also the project can be saved in a different from the current format used using the **Main menu > File > Save project as...** option or  in the *Toolbar File* where Save as type can be selected either as `.xfp` or `.xfz`.

 **Please note:** project loaded into XFlow are saved with the same format as the loaded project, even if it is different from the default project format set in the preferences.

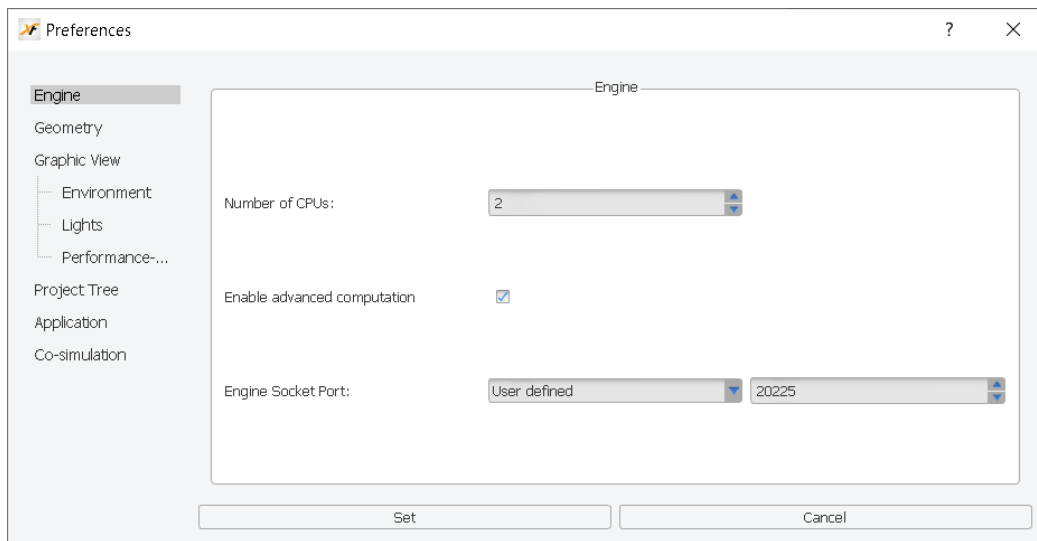
3.2 Set number of CPUs

Set the number of CPUs, to be used for the calculation, in the preferences of the engine:

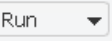
Main menu > Options > Preferences > Engine, or  in *Toolbar File*.

1 or 2 processors are enough for this tutorial.

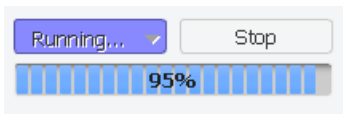
Step 3: Run and workflow



3.3 Start the computation

Click the **Run** button  > **Start computation**

The progress bar moves between 0 and 100%. A new data file (frame) is written when the progress bar gets to 100%.



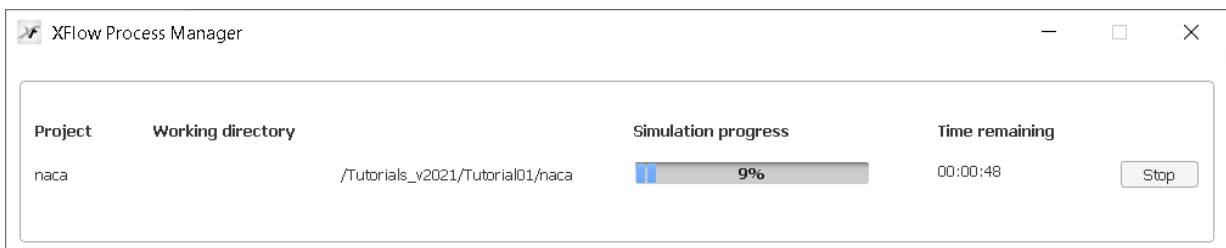
The global progress of the simulation can be observed in the timeline at the bottom of the interface, and shows a growing orange bar that indicates the amount of physical time computed in the simulation:



As XFlow is calculating, engine messages are shown in the *Message View* window.

3.4 Process Manager

When launching a simulation, the XFlow *Process Manager* appears in the Windows Taskbar. It allows the user to monitor and stop the simulations running in a machine.



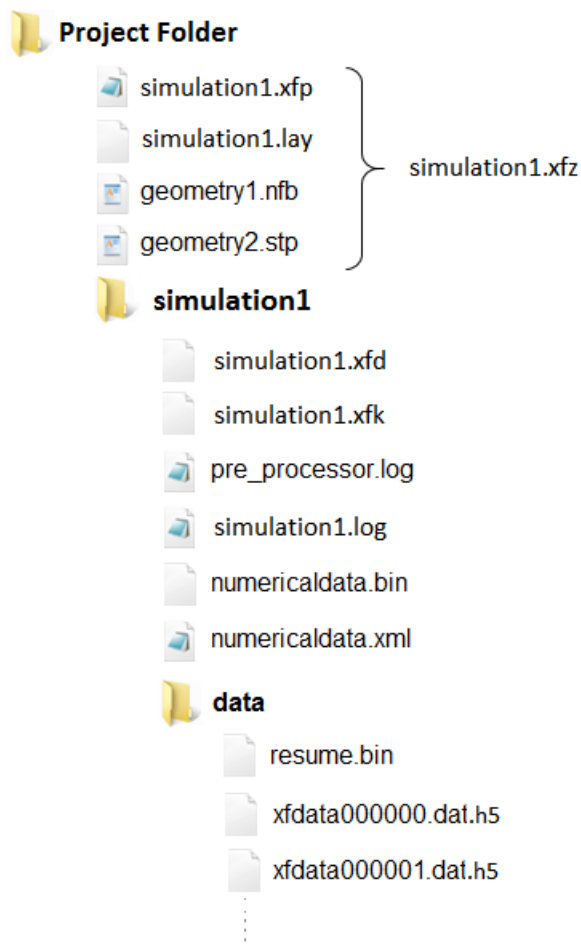
Note that the *Message View* window shows the message "Connected to Engine Daemon on port 20225".
The port can be changed in

Main menu > Options > Preferences > Engine: Engine Socket Port

⚠ Please note: The *Graphical User Interface (GUI)* can be closed while the simulation is running. The *Process Manager* is the minimum interface with your computation. XFlow will reconnect the simulation to the interface by means of the *Process Manager* when reopening the project again.

3.5 XFlow files structure

XFlow creates a folder called with the same name as the project file, in the same location as the project file. For instance if the project is named `naca.xfp`, a simulation folder named `naca` will be created in the same folder. The results data and log files are saved in this folder with the structure depicted as follows:



The content of the simulation folder is the following:

Step 3: Run and workflow

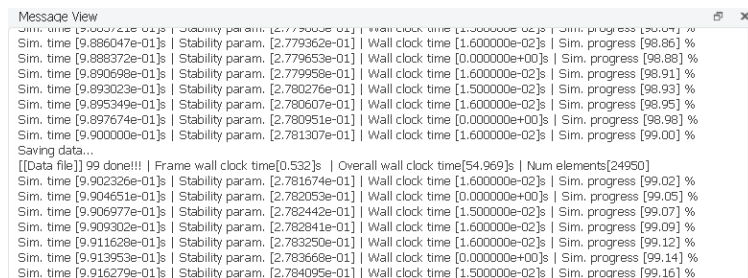
- **pre_processor.LOG** ASCII text file containing the domain generator output shown in the *Message View* window. It describes the octree structure of the lattice.
- **project_name.LOG** ASCII text file containing the solver output shown in the *Message View* window. Time steps iterations and information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

The simulation folder also includes a `project/data` folder where data frames are saved in h5 format (`xfdata000000.dat.h5`).

3.6 Message View

Check the *Message View* window. It is showing all the information XFlow needs to communicate, including the computation logs.

During the computation, the following messages will appear:

The screenshot shows a window titled "Message View" with a scrollable text area containing simulation logs. The logs consist of multiple lines of text, each representing a time step. Each line includes fields for simulation time, stability parameter, wall clock time, and simulation progress percentage. For example, one line reads: "Sim. time [9.886047e-01]s | Stability param. [2.779362e-01] | Wall clock time [1.600000e-02]s | Sim. progress [98.86] %". The logs also include a section for saving data, indicating the completion of a frame.

Message View window

Each line corresponds to one time step (solver frequency). For each time step, XFlow outputs:

- Sim. time: the total simulation time reached at the current time step
- Stability param.: the value of the stability parameter at the current time step, see [Step 3.7: Stability parameter](#)
- Wall clock time: the total time which has been required to compute the time step
- Sim. progress: ratio as a % of the Sim. time reached at each time step and the total simulation time specified in the Simulation setup.

When enough time steps are computed to create a new frame, XFlow will show the message "Saving data..." and then "[[Data file]] 1 Done!!!" (for the 1st frame). Every time a frame is

computed, the following information is also shown:

- **Frame wall clock time:** total time which has been required to compute the current frame
- **Overall wall clock time:** total time which has been required to compute from frame 0 to the current frame
- **Num elements:** number of elements in the fluid domain at the current frame

3.7 Stability parameter

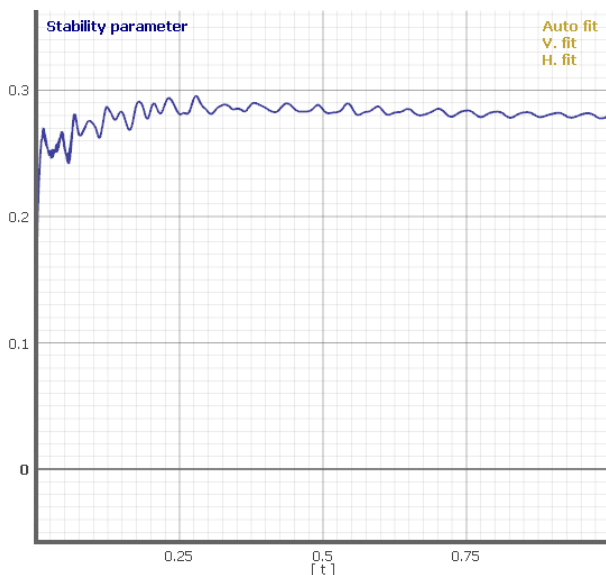
When your simulation is running, the user has the possibility to monitor the evolution of the **Stability parameter** against the time. This Stability parameter has values between 0 and 1 and provides a feedback regarding the stability of the numerical scheme, respect of the Courant–Friedrichs–Lewy (CFL) condition.

A low stability parameter (< 1) means the stability of the numerical scheme is ensured and the solution should therefore be consistent. If it is very close to 0, you may increase your time step to save computation time.

A stability parameter of 1 means the stability of the numerical scheme is not ensured and the simulation may diverge. You must therefore decrease your time step to ensure the convergence.

Ideal value would be around 0.1 - 0.4.

The Stability parameter can be monitored in the *Function Viewer* window. To create a Function Viewer, click on the *New function viewer* icon . In order to plot the Stability Parameter, **right click** on the *Function Viewer* window and select Stability parameter. In this case we can observe the stability is stable and within the ideal range (0.1-0.4).



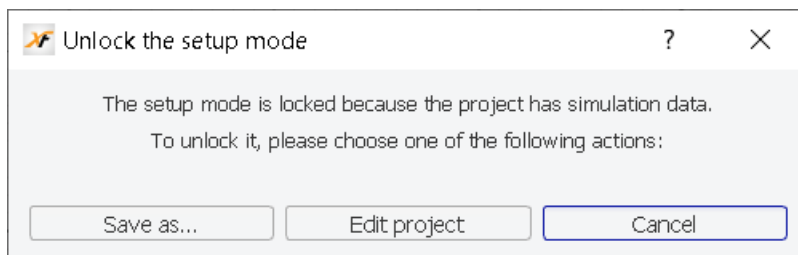
Stability parameter evolution, monitored in the Function Viewer

3.8 Project modes

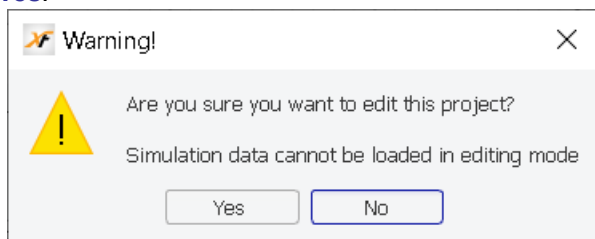
XFlow includes several project modes. Note the project mode indicated is **Post-processing mode** in the top of the project tree since the simulation has been run:



The post-processing mode is enabled automatically when running a simulation. It prevents the user from changing the simulation parameters in the project tree, in order to maintain synchronized input and output data. There are, however, two ways to change the simulation parameters of a project: save as a new project, or edit the current project. Pressing the **Post-processing mode** button, the following dialog will open:

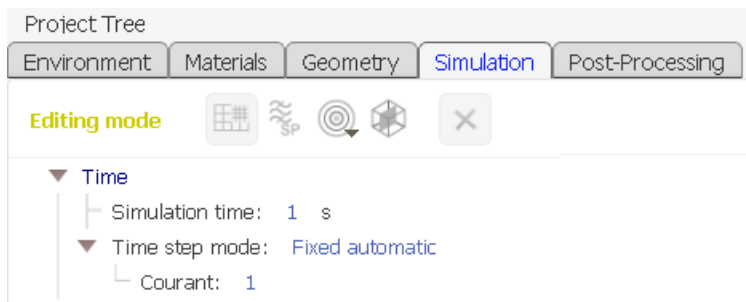


XFlow will ask for confirmation and warn you that simulation data cannot be edited in editing mode. Press **Yes**:



⚠ Please note: If you try to click on a simulation parameter to change it, the same dialog will automatically pop-up.

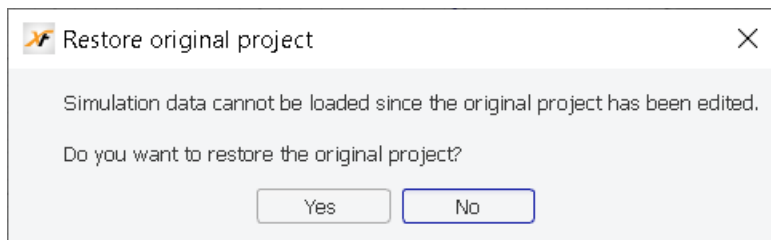
Press **Edit project** and confirm the edition of the project. Observe the project mode is now Editing mode:



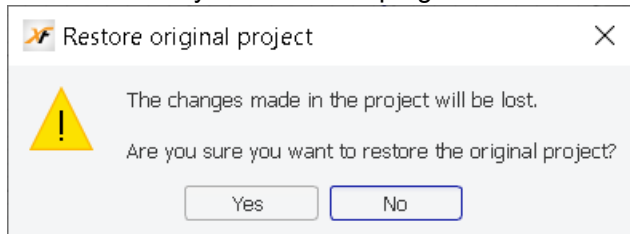
If your simulation is still running XFlow will ask if you want to stop it. Press **Yes**, it will also unload the current simulation data. Note you can now change the simulation parameters and therefore edit your project.

⚠ Please note: simulation data cannot be loaded in Editing mode, as project inputs are asynchronous with simulation outputs.

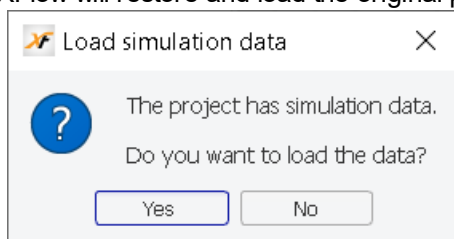
Recover the original project that was executed in step 3.3 by pressing **Editing mode**. XFlow will ask if you want to recover your original project and will ask you to confirm, press **Yes**:



XFlow will warn you the current progress of edition will be lost, press **Yes** again:



XFlow will restore and load the original project and will ask if you want to load the data, press **Yes**:

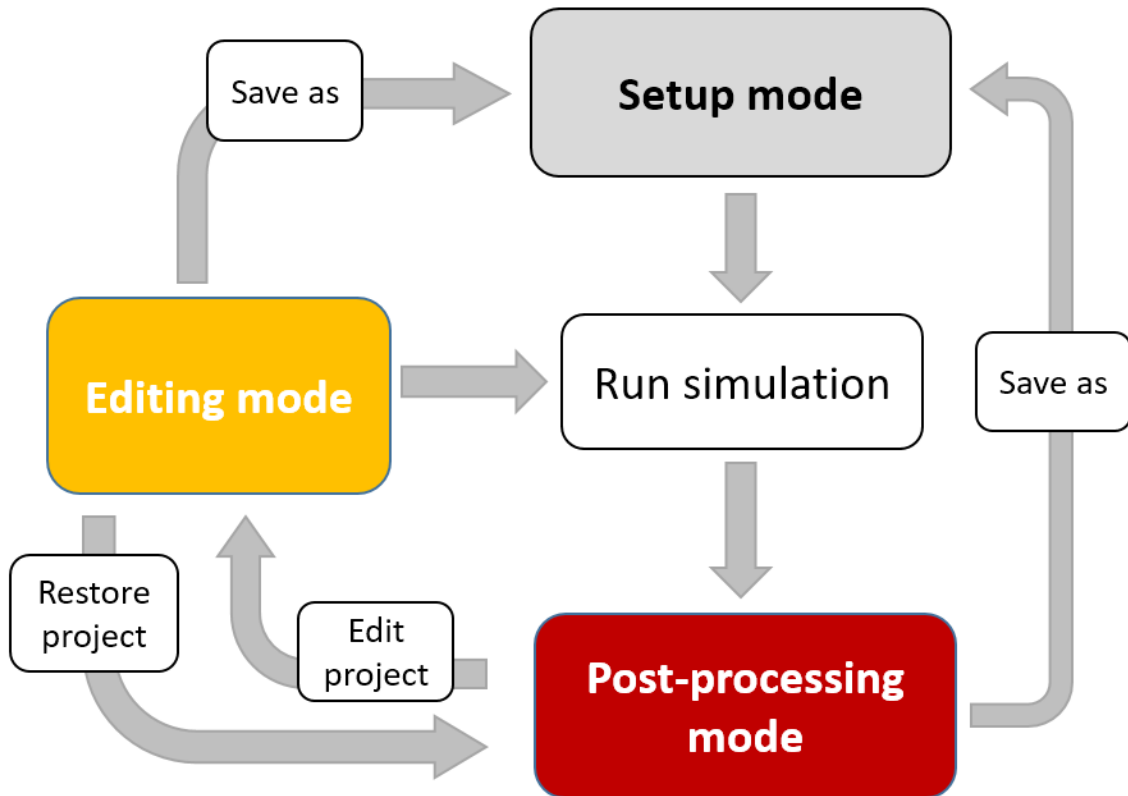


The simulation data are loaded again, observe the orange timeline is shown which indicates data frames are available to load:

Step 3: Run and workflow

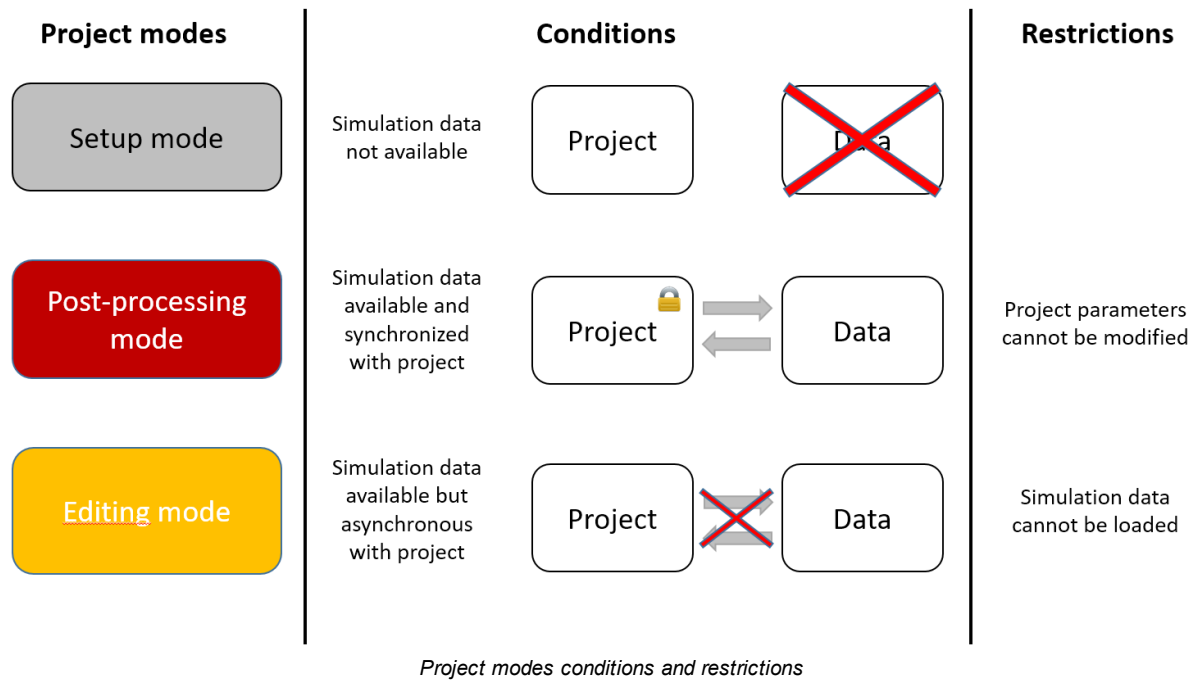


The simulation is now recovered. A summary of the project modes workflow is depicted as follows:



Project modes workflow

The different project modes, conditions and restrictions are summarized in the following Figure:



Step 4: Post-processing

The post-processing is completely managed from the **Post-Processing** section of the *Project Tree*.

If the GUI is left opened during the computation, the resulting numerical data can be post-processed immediately after their generation (data automatically loaded to the GUI). Otherwise, data has to be loaded into the GUI from the folder where it is stored:

Main menu > Simulation data > Load simulation data or in Toolbar Data Processing.

Please note: The project has to be saved before you are able to load the data.

4.1 Visualize the velocity field

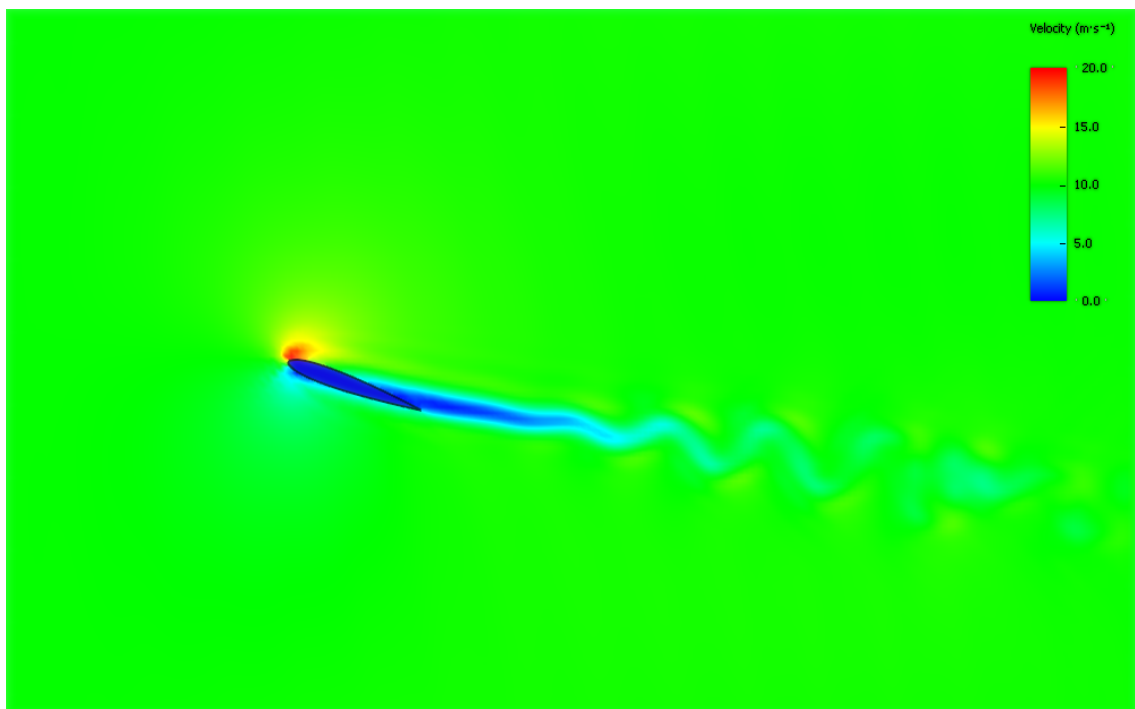
Create a cutting plane to visualize the velocity field. To proceed, go to **Project Tree > Post-processing** and:

- (a) **Right-click** on **Cutting planes** and select **Add cutting plane**, or press in the Post-Processing Toolbar, or go to **Main menu > Post-Processing > Create cutting plane**

Step 4: Post-processing



- (b) **Cutting plane > Axis:** Select **Z** (The position of the plane cannot be modified, as it is a 2D simulation)
- (c) **Cutting plane > Visualisation mode:** **3d field**
- (d) **Cutting plane > Visualisation mode > Field:** **Velocity**, to visualise the velocity field on the cutting plane, at the time (frame) indicated on the timeline.
- (e) Switch on the interpolation in **General > Interpolation mode:** **Convolution** to interpolate data and have smoother contours
- (f) In the *Graphic View* window, select view from the right hand side 
- (g) In the timeline, select the last frame corresponding and observe the velocity field:



Interpolated velocity field at last frame, $t = 1$ s

 **Please note:** Even if the flow has already stabilised at $t = 1$ s, the fluctuations of the wake and the separation of the boundary layer are not captured properly.

- (h) Select other visualisation fields, e.g. Static pressure or Vorticity.

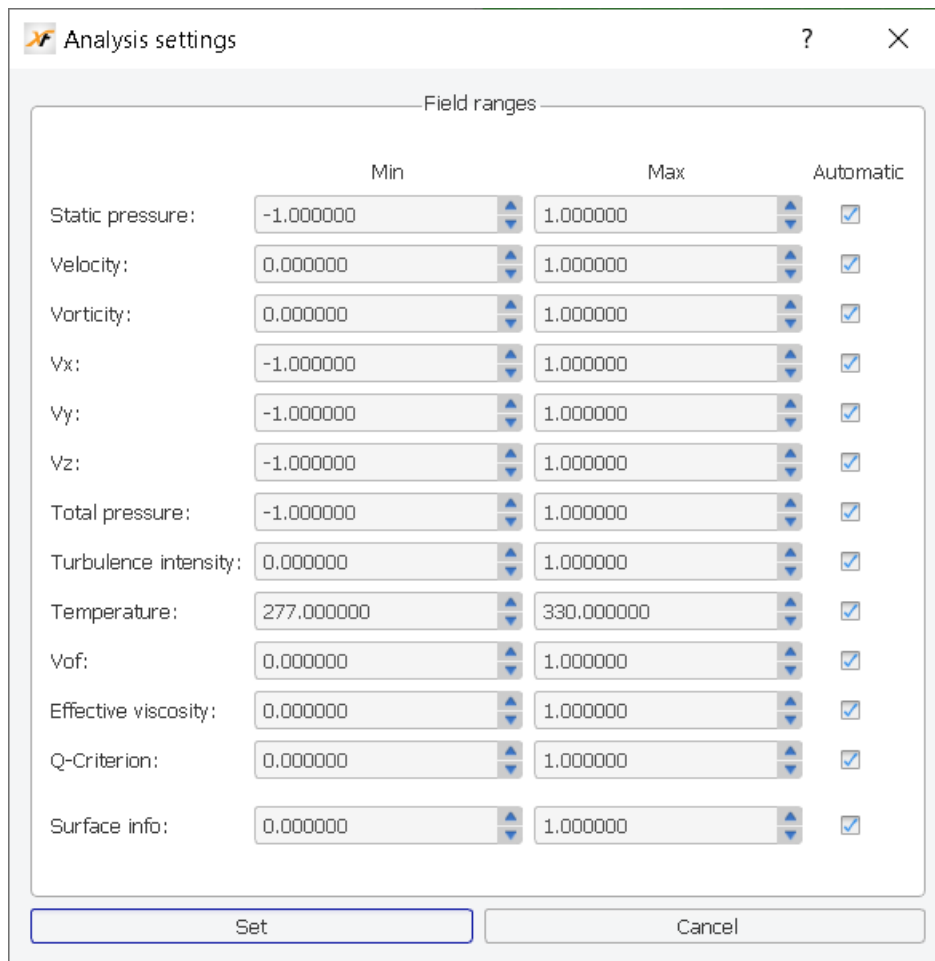
4.2 Use the playback controls to navigate through the transient results

- (a) Go to the last calculated frame by clicking on 
- (b) Go to the first frame (frame 0) by clicking on 
- (c) Play forward the frame sequence by clicking on 
- (d) Play backward the frame sequence by clicking on 
- (e) Move frame by frame by clicking on  and 
- (f) Go to a specific time frame by clicking on its number in the timeline

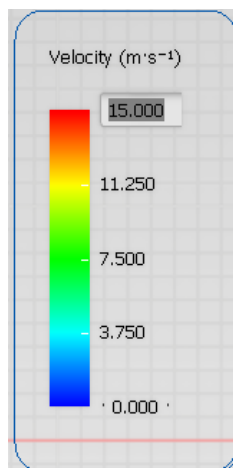
4.3 Customise the visualisation

- (a) In **Main menu > Simulation data > Analysis settings** or using the toolbar icon  prescribe the velocity legend range to **[0,15] m·s⁻¹** by disabling the automatic range and inputting the minimum and maximum values.

Step 4: Post-processing



An alternative to change the range is to click on the maximum and minimum values of the color bars to edit them directly:

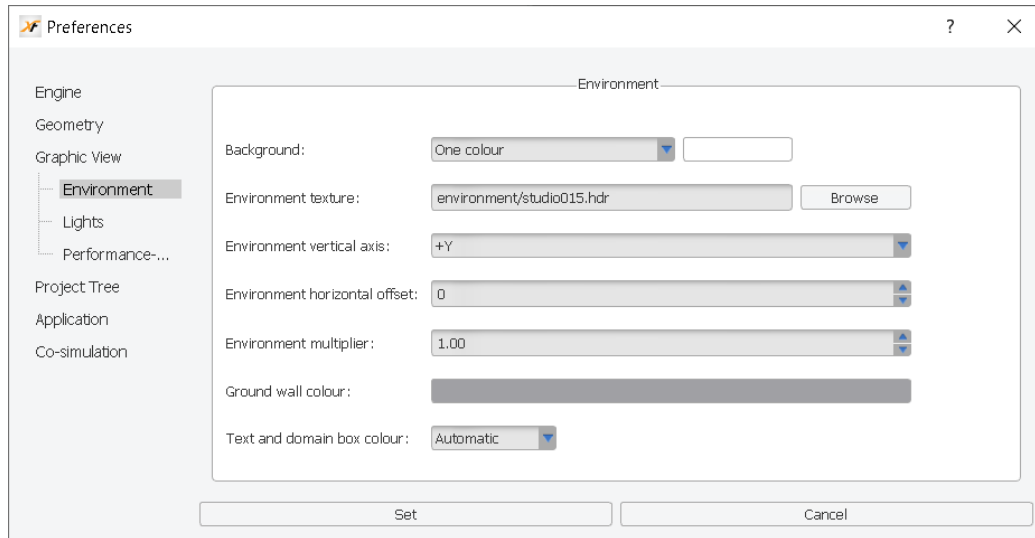


Play the frame sequence again.

(b) Change the background colour to white in **Main menu > Options > Preferences** or the toolbar icon



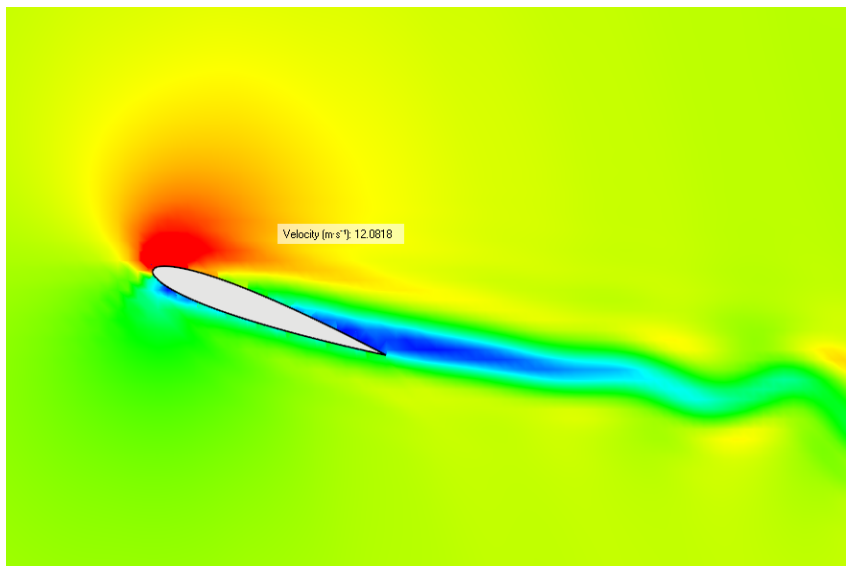
In the **Preferences > Environment** select **One colour** background, click on the coloured rectangle at the right hand side and pick the white colour in the palette.



4.4 Explore details of the results

The look up tool  displays numerically the value of the visualisation field at the position pointed by the mouse.

To deactivate this mode, press again the same toolbar icon.



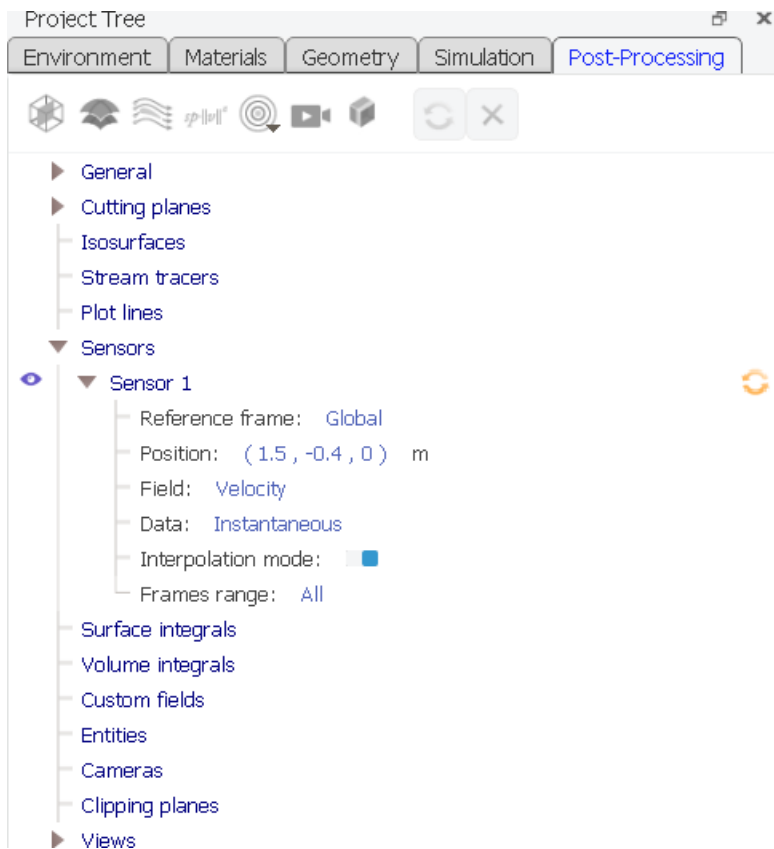
4.5 Create a sensor

Post-Processing > Sensors

(a) Create a sensor: Post-Processing > **Right clicking** on **Sensors** > **Add sensor**

Step 4: Post-processing

The new sensor is automatically called **Sensor 1**. Change its position to **(1.5, -0.4, 0)** to be in the wake region of the airfoil and set the **Field** to **Velocity**.



(b) Hide **Cutting planes > Cutting plane 1** by clicking on visibility icon in front of it  and

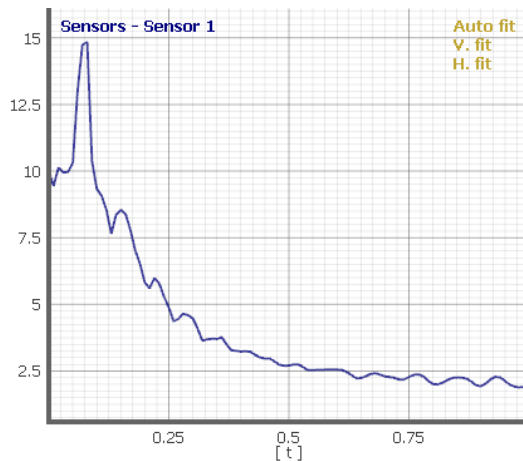
make sure the visibility icon of **Sensors > Sensor 1** is active  in order to visualize the location of the sensor in the *Graphic View*.

(c) Refresh the sensor: **Right click** on **Post-Processing > Sensors > Sensor 1 > Refresh** or click the icon 

(d) By default no function viewer is set in the XFlow GUI, so the user has to create a new one **Main menu > Window > New Function Viewer** or by selecting the icon .

(e) Plot the chosen field at sensor 1: **Right click** on **Function Viewer > Sensors > Sensor 1**

(f) You may want to resize the *Function Viewer* window by dragging its borders, and fit the plot in the window by clicking on **Auto fit** to refit in X and Y axis, **V. fit** to refit in Y-axis only, and **H. fit** to refit on X-axis only.



Drag the mouse to left and right while pressing **Middle mouse button** or **Scroll in/out** to zoom in and out the graph, and drag the mouse while pressing **Left mouse button** to pan the graph.

To change the scale of the graph in only in X or Y separately, drag the mouse horizontally or vertically while pressing **Alt + Right mouse button** to rescale respectively the X-axis or Y-axis.

(g) Save the sensor data to a text file **Right click** on *Function Viewer* > **Export current data** and give it a name. The first column in the file represents time (in seconds) and the second column the values of the field measured by the Sensor 1, i.e. the velocity module (in $\text{m}\cdot\text{s}^{-1}$).

4.6 Calculate drag and lift

The drag and lift coefficients are dimensionless parameters that describe the forces acting on a body in a fluid flow. The drag force is parallel to the airflow, whereas the lift force is perpendicular.

These coefficients and the forces can be plotted in the *Function Viewer*:

Right click on *Function Viewer* > **Shapes > Naca-0012-4Digit > Cx**

Right click on *Function Viewer* > **Shapes > Naca-0012-4Digit > Cy**

The aerodynamic coefficients are calculated based on the reference velocity and area defined in **Environment > Global attributes > Reference area/velocity**. If correct values were not set before the simulation was launched, the coefficients can always be computed from the forces in Newton:

Right click on *Function Viewer* > **Shapes > Naca-0012-4Digit > Fx**

Right click on *Function Viewer* > **Shapes > Naca-0012-4Digit > Fy**

⚠ Please note: This tutorial is executed with a coarse resolution as a demonstration, hence the correct

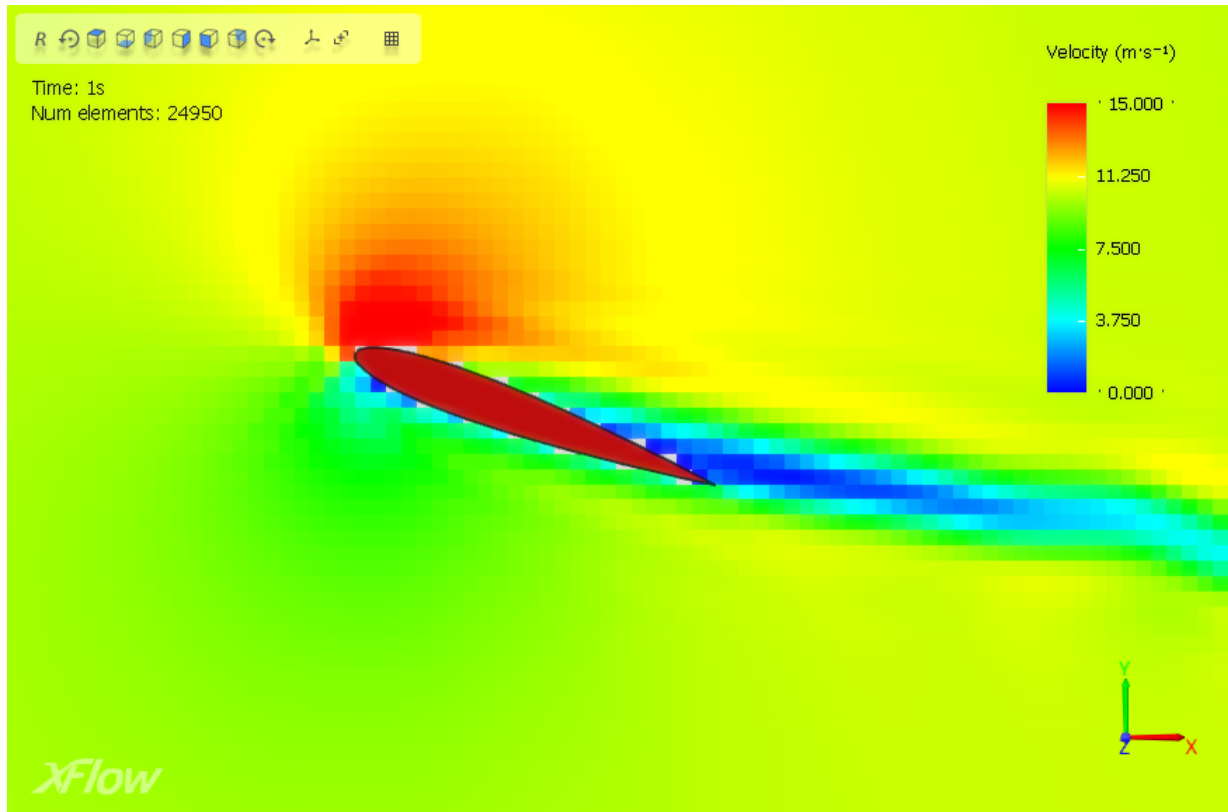
physics may not be captured. For validations purpose, please check the Validation Guide.

4.7 Observe the resolution

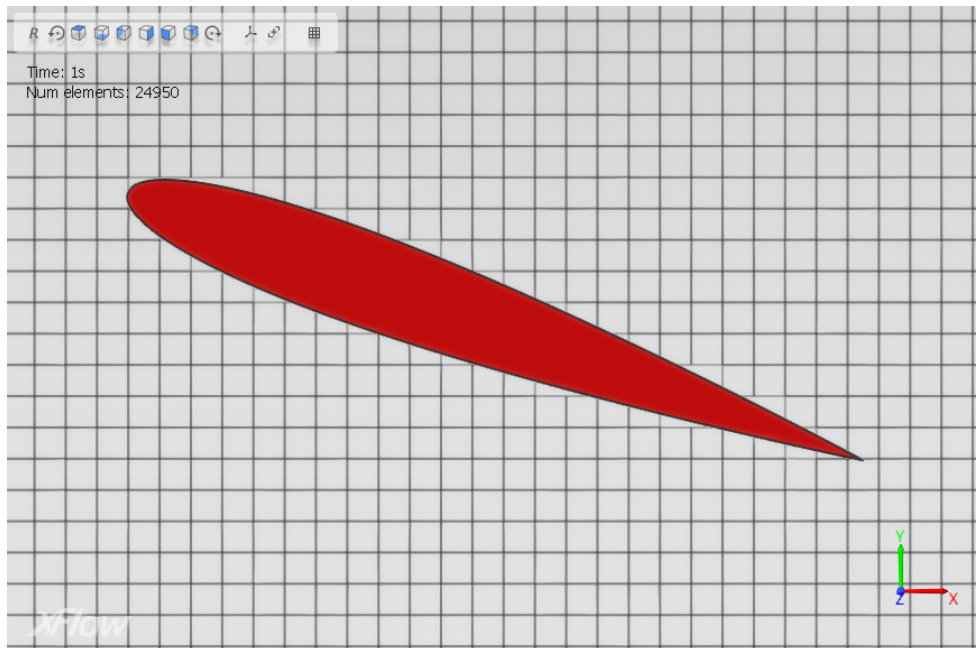
(a) Show the **Cutting plane 1** by checking its visibility icon 

(b) In **Post Processing > General > Show** set the **Interpolation mode** to **Off**.

Now the velocity field is displayed as a voxelization, being each voxel a lattice node. In this way, you can distinguish the size of the cells.



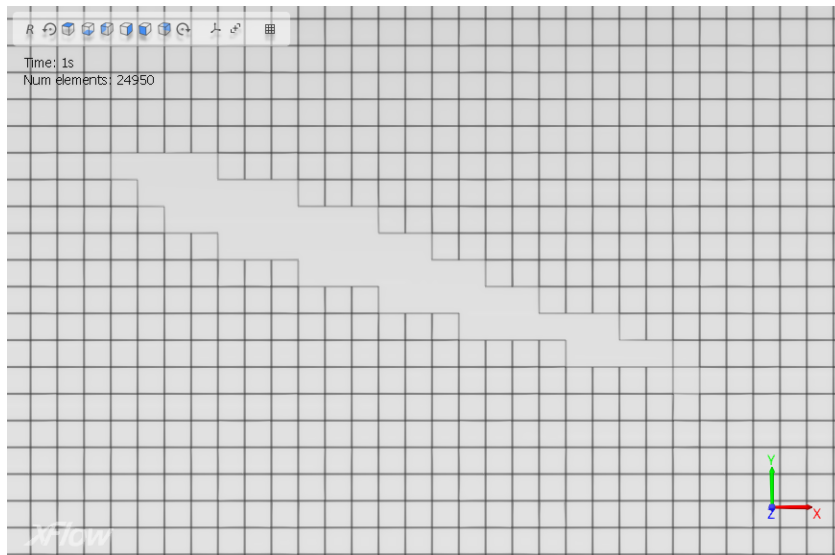
(c) In the **Cutting plane 1** switch the visualization mode to **Domain structure**



Domain structure around the NACA airfoil (lattice)

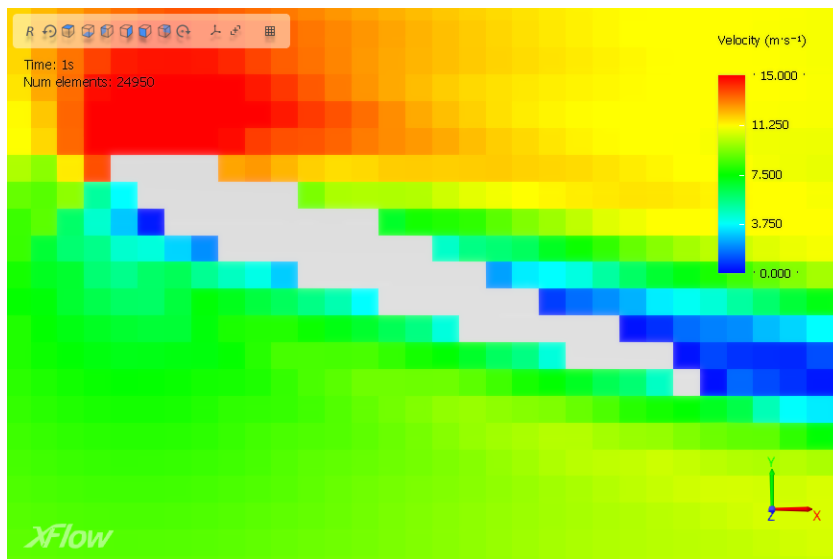
The grid shown is the lattice structure. The lattice nodes are defined at the center of every voxel where all fluid variables are defined (velocity, pressure, etc.). The lattice structure gives a visual feedback of the spatial discretization used.

Now, hide the geometry in **Geometry tree > Entities** by disabling the visibility icon  and observe the **Domain Structure** and the 3d field: **Velocity**:



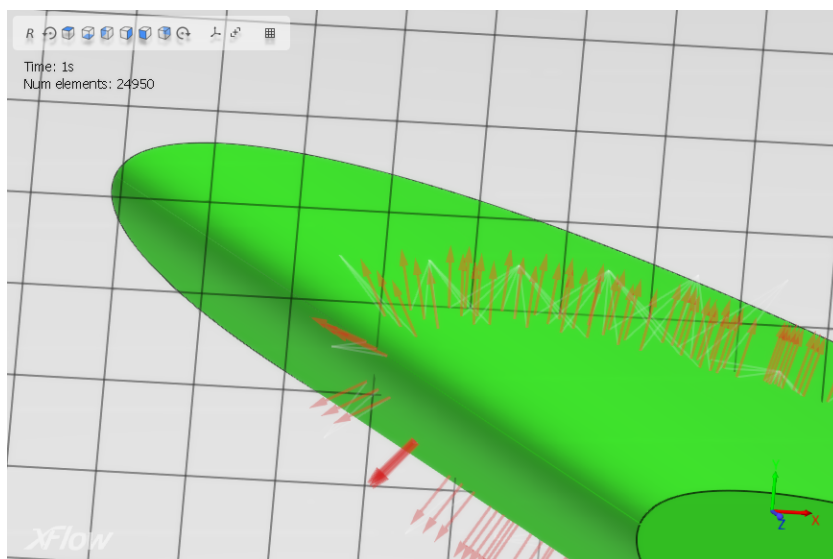
Domain structure around the NACA airfoil hiding geometry

Step 4: Post-processing



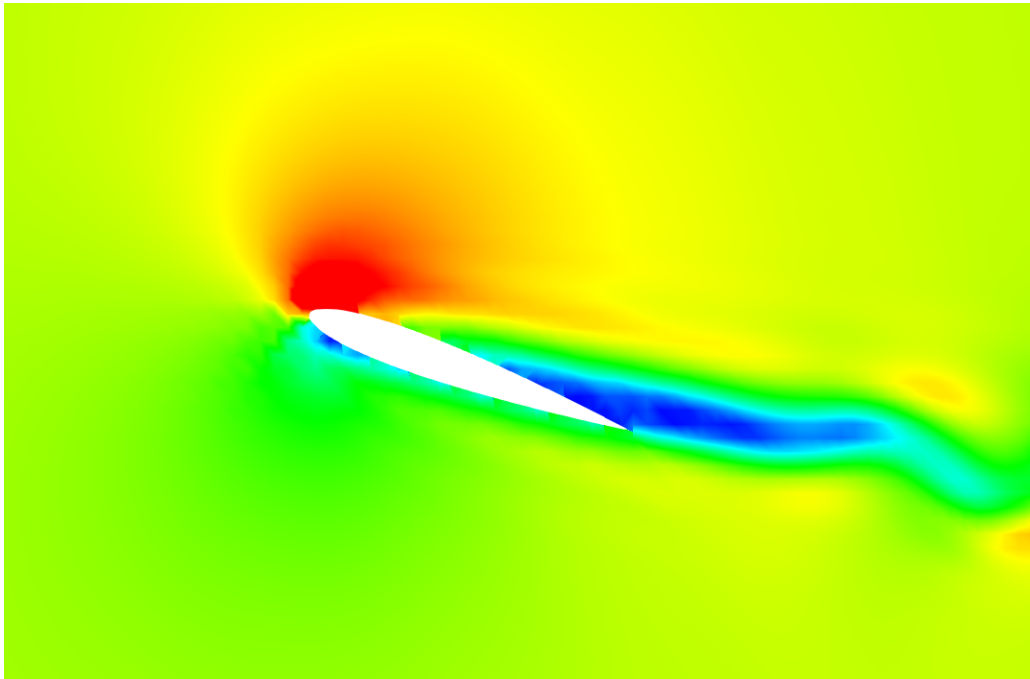
Velocity field around the NACA airfoil hiding geometry

Despite the domain presents a step-like pattern, each near-wall voxel includes up to 27 velocity directions to intersect with surrounding geometries due to the XFlow lattice scheme. Hence the actual geometry discretization is much finer than what is observed on the domain near-wall domain structure. These intersecting velocities are also named **broken links** in XFlow and can be visualized enabling an advanced parameter that will be explained in [Tutorial 06 - Step 01](#):



Geometry discretization with near-wall lattice velocities ("broken links")

For visualization purpose, the 3d fields can be observed with interpolation in order to have a smooth cutting plane across complex geometries. To enable the interpolation, switch to **General > Interpolation mode: Convolution**:



Interpolated velocity field around the NACA airfoil

In this case one can realize the resolution used is quite coarse. For this reason the expected behaviour of the flow is not achieved: the transient behaviour of a flow around a NACA airfoil with an angle of attack of 20 degrees must show a periodic flow motion developing in the wake region as well as a large separation of the boundary layer.

The [Step 5: Refine the resolution](#) will teach how to refine the solution in order to observe the expected behaviour in the wake and boundary layer regions.

Step 5: Refine the resolution

We will now run the same simulation using multi-resolution, i.e. a finer resolution close to the NACA airfoil walls to better resolve the flow velocity gradients. As you have seen in the previous simulation, the flow has stabilized but the wake and the boundary layer did not develop as expected. Therefore, the goal of this section is to capture the wake fluctuations and the separation of the boundary layer.

5.1 Save as a new project file

Main menu > File > Save project as or  in Toolbar File with a different name (for instance: `naca_fine.xfp`).

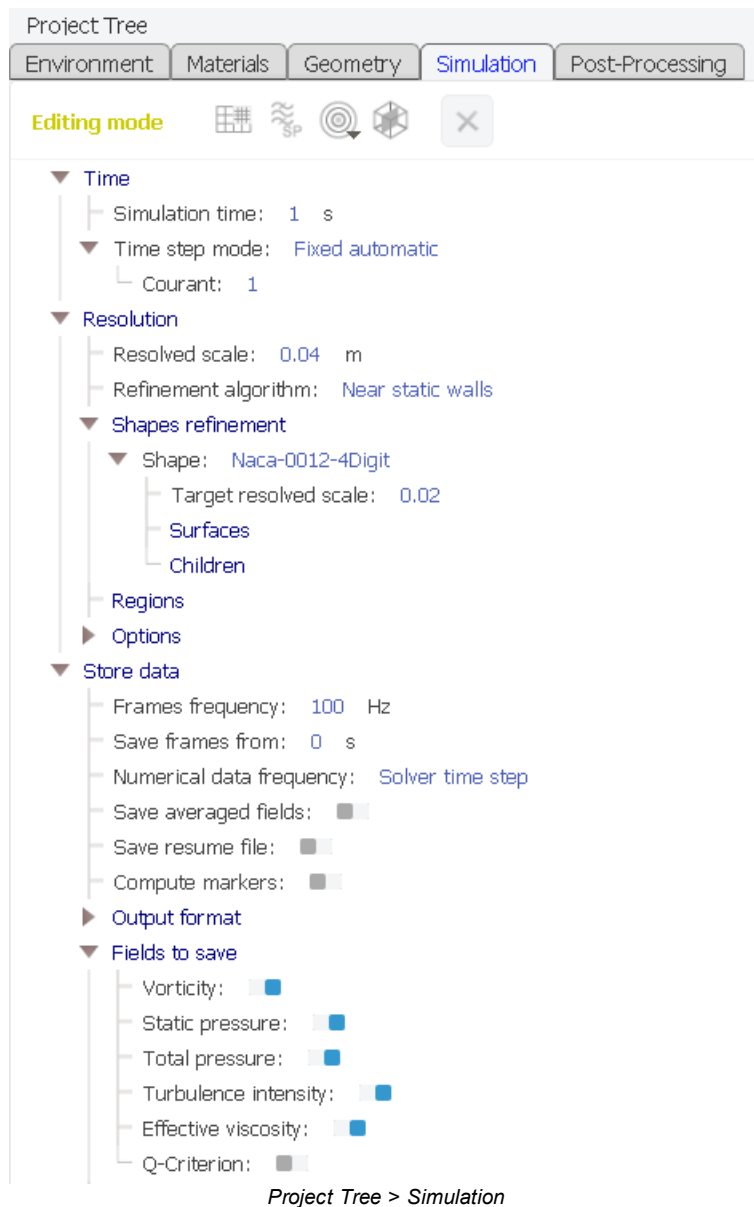


Tip: You can directly load the setup of this problem from the project file `naca_fine.xfp` in **Main menu > File > Load project** or **Open an existing project** in the *Project Manager* window that appears when executing XFlow.

5.2 Refinement algorithm

Change the following parameters in **Project Tree > Simulation**:

- (a) **Time > Simulation time: 1 s** (as in the previous case)
- (b) **Lattice discretization > Resolved scale: 0.04 m** (as in the previous case)
- (c) **Lattice discretization > Refinement algorithm: Near static walls**
- (d) **Lattice discretization > Shapes refinement > Shape: Naca-0012-4Digit > Target resolved scale: 0.02 m**
- (e) **Store data > Frames frequency: 100 Hz** (as in the previous case)
- (f) **Store data > Fields to save: On**, for all the fields



5.3 Run the new refined case

Press the **Run button** > **Start computation**

5.4 Post-process the results of the refined case

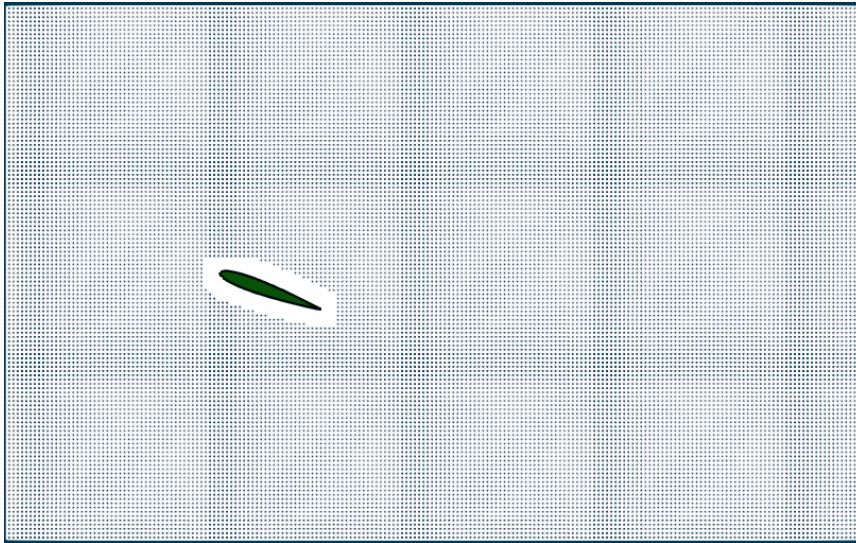
Display the structure of the multi-resolution:

- Locate the time bar on a calculated frame, e.g. frame 100
- Create a cutting plane with the icon  or make visible the **Cutting Plane 1** previously created in Step 4 if not deleted with the visibility icon 
- Set **Cutting Plane 1** > **Visualization mode: Domain structure** in order to observe the

Step 5: Refine the resolution

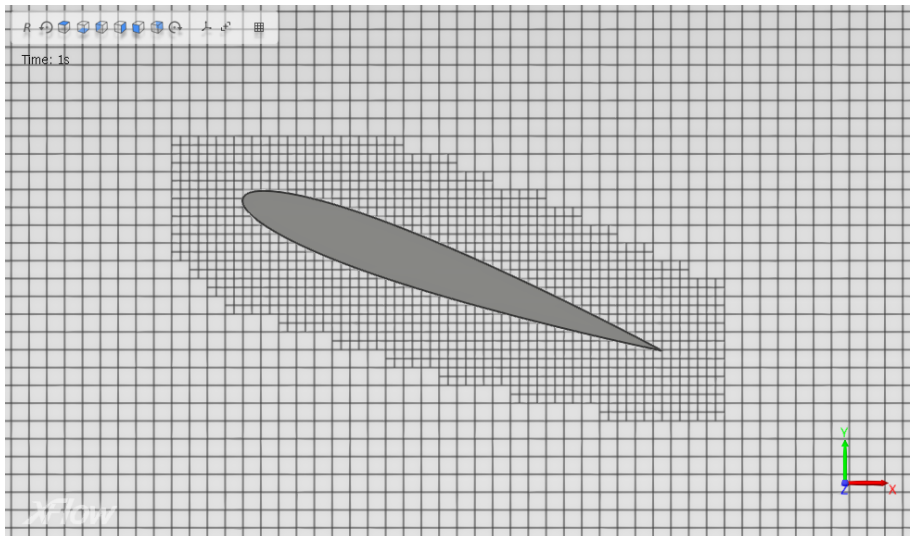
domain.

This will be similar to that shown in the figure below:



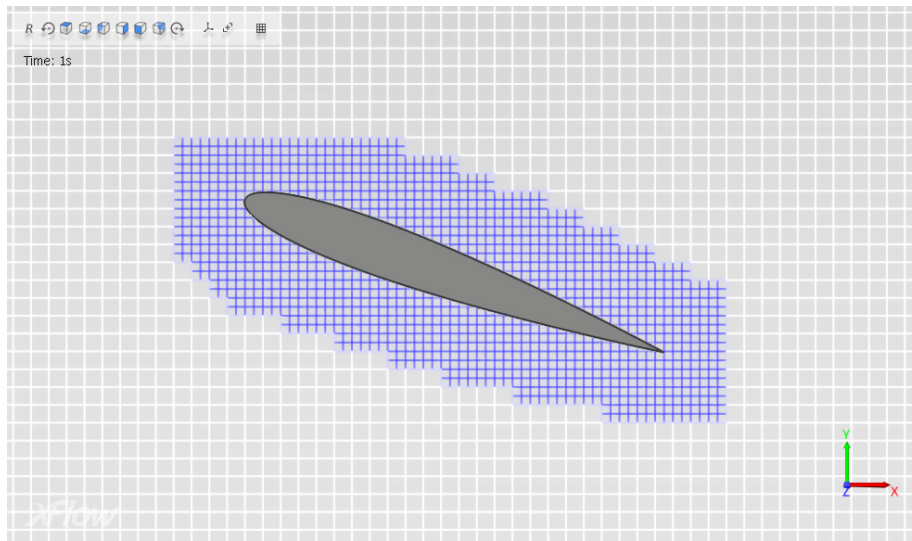
Domain structure at $t = 1$ s, using the Near static walls refinement

Note that the regions around the airfoil walls have been refined. The domain has two levels of refinement now: $h = 0.04$ m and $h = 0.02$ m.



Lattice refinement near the NACA

(d) Set **Color by level: On**. This highlights the different lattice resolution with a distinct color. The following lattice structure should be observed:



Lattice refinement near the NACA

⚠ IMPORTANT: Due to the Cartesian structure of the lattice, the choice of the different resolutions must be related by a **factor 2^n** , n being an integer. Example here with $h = 0.04/2^1 = 0.02$ m near the wall.

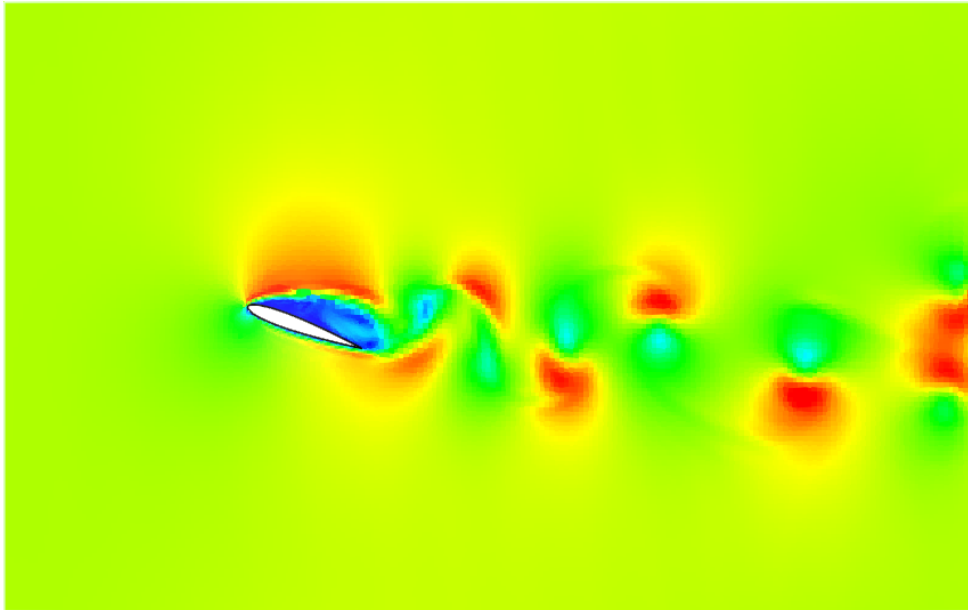
5.5 Observe the flow separation

(a) Set the **Cutting Plane 1** with the following parameters:

- **Axis:** Z
- **Visualisation mode:** 3d field
- **Visualisation mode > Field:** Velocity

(b) Load the frame 100 to see the velocity field at time $t = 1$ s, shown in the figure below:

Step 5: Refine the resolution

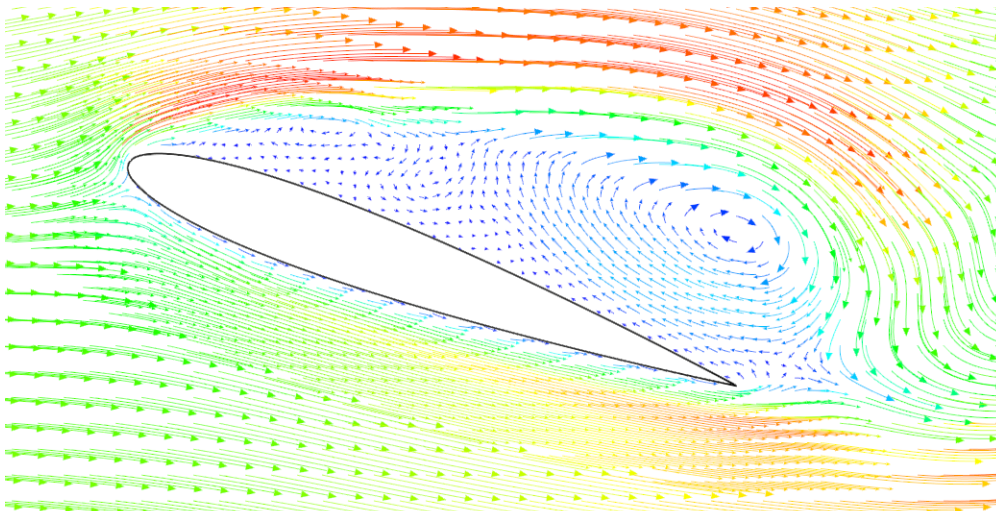


Velocity field showing the fluctuations of the wake and the separation of the boundary layer around the airfoil at frame 100

The proper separated flow patterns are now observed, the recirculation area is much larger on the upper side of the NACA.

(c) Switch now the **Visualisation mode**: Vectors, and set the **Arrow density** to 0.5 and the **Arrow length** to 1. Click the refresh icon .

Zoom in the recirculation area to observe separation of the boundary layer:



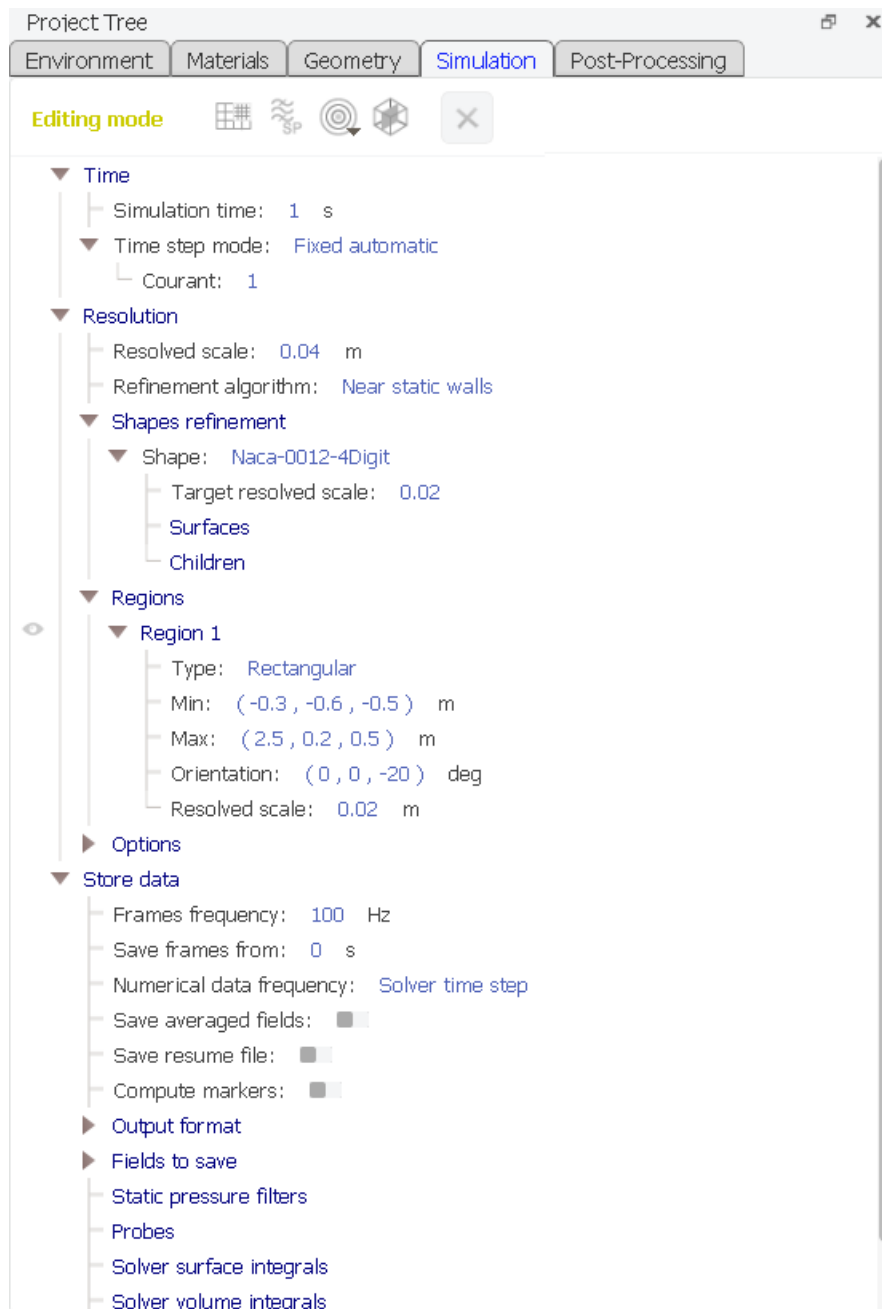
Velocity vectors around the NACA profile at $t = 1$ s

5.6 Create a region of refinement

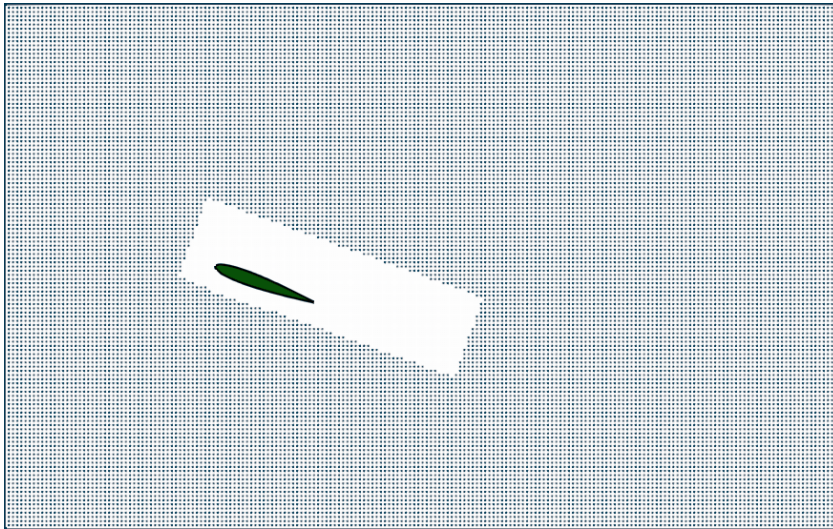
We are now going to edit the current project and define a fixed region of refinement to better refine the flow separation around the NACA airfoil.

- (a) Click on **Post-processing mode** in the top of the project tree (in any tab except Post-processing tab). A window asks which action do you want to perform in order to unlock the project tree.
- (b) Press **Edit project**, and confirm pressing **Yes**. Editing mode is now enabled and you can edit your project. Set the project as described in the following steps.
- (c) **Time > Simulation time: 1 s**
- (d) **Project Tree > Simulation > Lattice discretization > Refinement algorithm: Near static walls**
- (e) **Project Tree > Simulation > Lattice discretization > Shapes refinement: Naca-0012-4Digit > Target resolved Scale: 0.02 m**
- (f) **Project Tree > Simulation > Lattice discretization > Regions: Right click** and select to **Add region** or click on the icon 
- (g) **Project Tree > Simulation > Lattice discretization > Regions > Region 1: Rectangular**
- (d) **Project Tree > Simulation > Lattice discretization > Regions > Region 1 > Min: (-0.3, -0.6, -0.5) m**
- (e) **Project Tree > Simulation > Lattice discretization > Regions > Region 1 > Max: (2.5, 0.2, 0.5) m**
- (f) **Project Tree > Simulation > Lattice discretization > Regions > Region 1 > Orientation: (0, 0, -20) deg.** This way the refinement region will be aligned with the NACA airfoil that has an angle of attack of 20 degrees.
- (g) **Project Tree > Simulation > Lattice discretization > Regions > Region 1 > Resolved scale: 0.02 m**
- (h) **Project Tree > Simulation > Store data > Frames frequency: 100 Hz**

Step 5: Refine the resolution



- (i) Hide the refinement box disabling the visibility icon 
- (j) Save the project.
- (k) Run the simulation. XFlow will ask you if you want to overwrite the existing simulation data computed previously Press **Yes**: you previous simulation data will be overwritten by a new one.
- (l) Show the domain structure by checking the box: **Post-Processing > Cutting planes > Cutting plane 1 > Visualization mode: Domain structure**

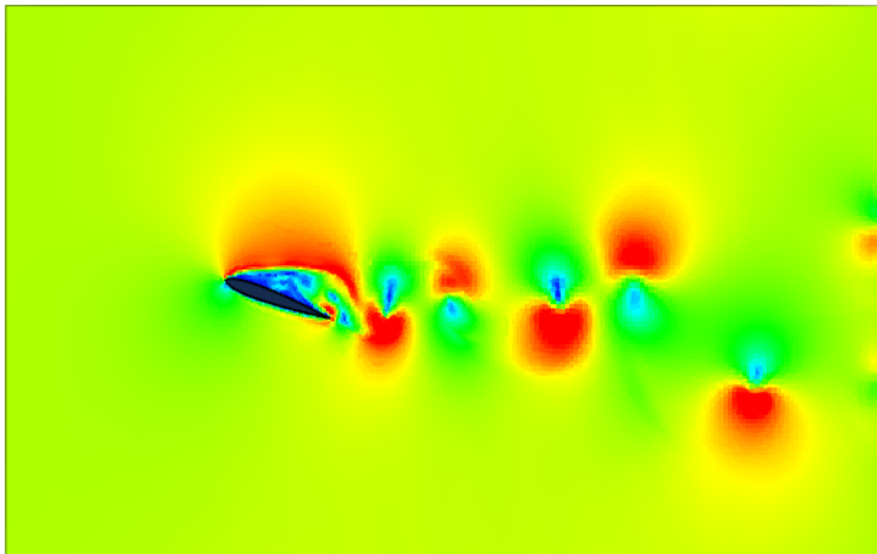


Domain structure with refinement region

(m) Show the cutting plane of velocity setting the following parameters

- **Axis: Z**
- **Visualisation mode: 3d field**
- **Visualisation mode > Field: Velocity**

(n) Load the frame 100 to see the velocity field at time $t = 1$ s, shown in the figure below:



Flow with a fixed region of refinement at $t = 1$ s

The flow in all the region around the airfoil is solved more accurately with the refinement region. This makes the flow separation fluctuations larger in this case compared to the case without the refinement region.

⚠ Please note: The gradients in the solution may become smoothed across the border of the refinement region. Therefore the region should be large enough to contain the relevant flow features. The next section will present the *Adaptive refinement* algorithm which allows the dynamic refinement of the

Step 5: Refine the resolution

wake.

Step 6: Moving NACA - Enforced motion

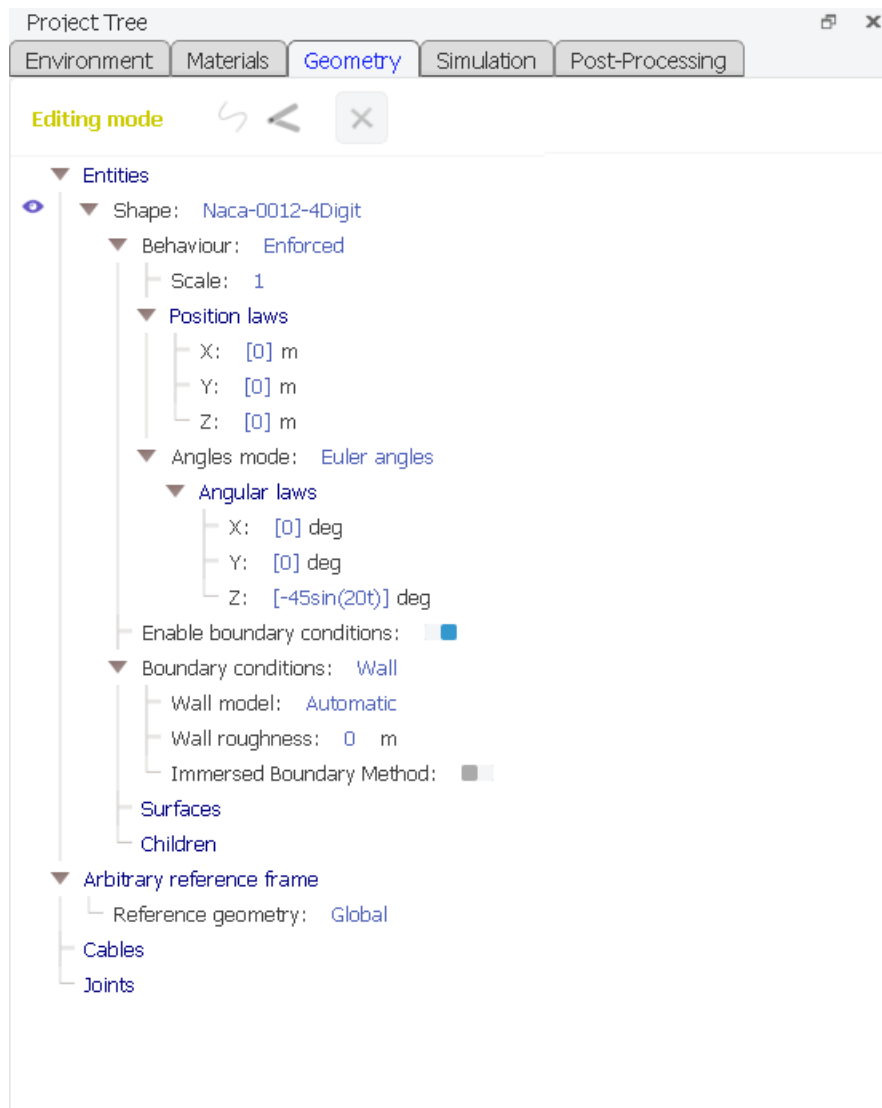
Now we are going to prescribe an enforced oscillating motion to the NACA through an analytic function and see its interaction with the flow.

6.1 Save as a new project file

Main menu > File > Save project as or  in Toolbar File and save the project with a different name (for instance: `naca-enforced.xfp`).

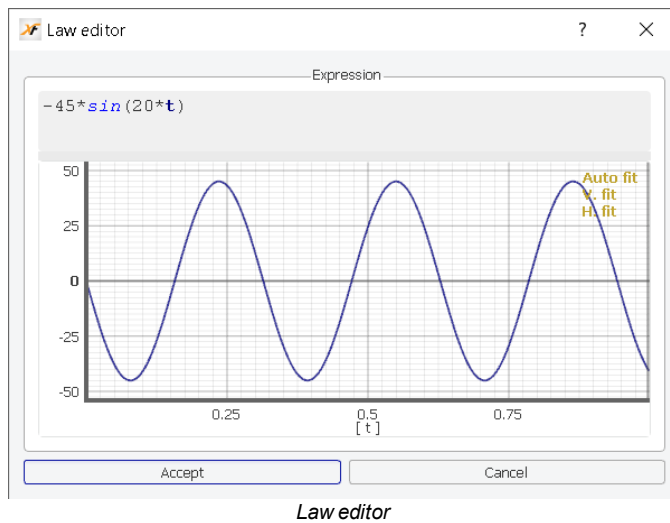
6.2 Set the shape motion

- (a) **Project Tree > Geometry > Entities > Shape: Naca-0012-4Digit > Behaviour: Enforced**
- (b) Let the **Position laws** with default values since the motion will only be a rotation (no translation).
- (c) **Project Tree > Geometry > Entities > Shape: Naca-0012-4Digit > Behaviour > Angular laws Z: $-45 \cdot \sin(20 \cdot t)$ deg**



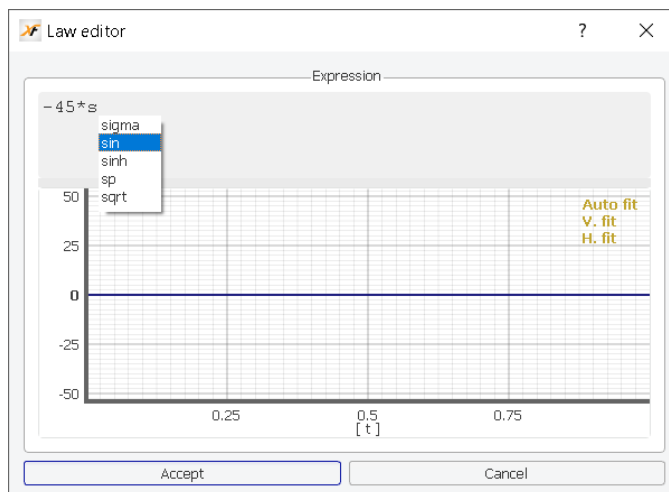
- (d) In XFlow, input laws are fields presented with brackets []. When you click on an input law, a law editor icon will show . Click on the icon to open the law editor and check the law is correctly interpreted:

Step 6: Moving NACA - Enforced motion



The law editor is a convenient way to check if your law is correctly interpreted by XFlow. Functions and constants that are recognized will be highlighted in yellow, while variables will be highlighted in blue. Use the **Auto fit** function to resize the function in the viewer below. This window can be resized to your convenience.

Check that if you start to write a variable or function name, a list of suggestion is shown:



(e) Check the motion of the airfoil by playing the frames timeline. You will see the NACA rotate with a periodic function around its leading edge.

6.3 Set the refinement algorithm

To enable a refinement algorithm go to **Project Tree > Simulation >** and do as follows:

- (a) **Time > Simulation time: 1 s**
- (b) **Lattice discretization > Resolved scale: 0.04 m** (as in the previous case)

- (c) **Lattice discretization > Refinement algorithm: Adaptive refinement.** Adaptive refinement allows to dynamically refine the moving geometry, as well as the wake area as it develops. Therefore it is well suited for this case.
- (d) **Lattice discretization > Shapes refinement > Shape: Naca-0012-4Digit > Target resolved scale: 0.02 m**
- (e) **Lattice discretization > Adaptive refinement > Wake resolution: 0.02 m**
- (f) **Lattice discretization > Regions:** No refinement regions. If the refinement region from Step 5 is still present in your simulation setup, remove it by pressing the **right mouse button** on **Region 1 > Remove region**.
- (g) **Store data > Frames frequency: 100 Hz**

6.4 Run the new refined case

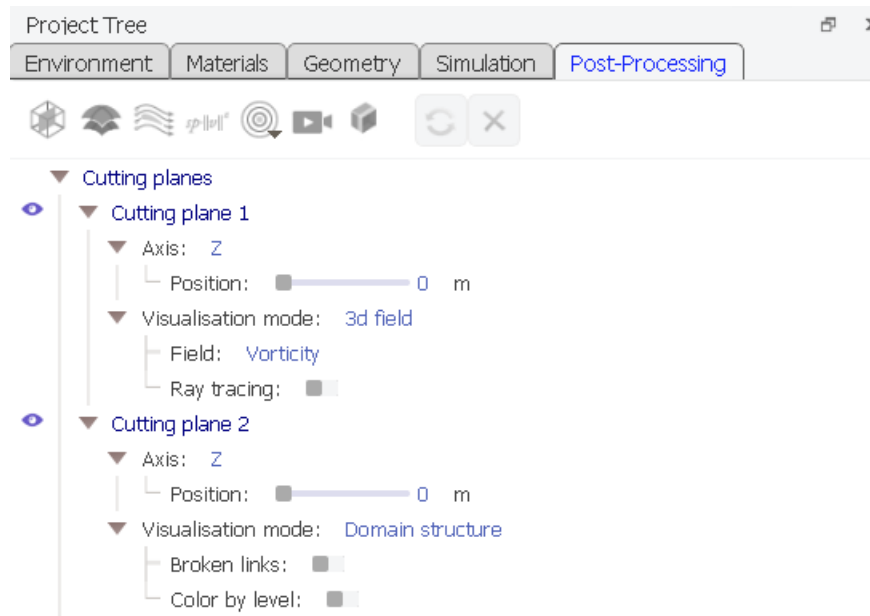
Press the **Run button > Start computation**. Post-processing mode will be enabled.

6.5 Post-process the results of the enforced-motion case

Visualize the time evolution of the adaptive domain structure:

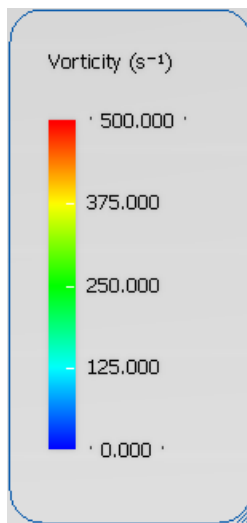
- (a) Create a first cutting plane and set **Visualization mode: 3d field**, with **Field: Vorticity** in order to visualize the vorticity field.
- (b) Create a second cutting plane and set **Visualization mode: Domain structure** in order to visualize the lattice structure during the simulation.

The two cutting planes parameters should look as follows:

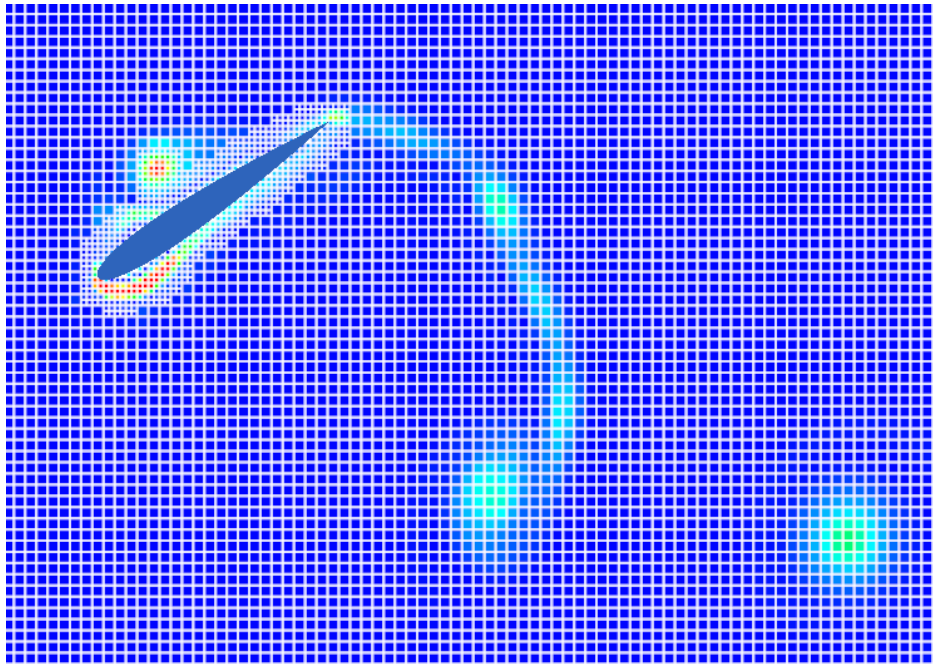


- (c) Set the vorticity range between 0 and 500:

Step 6: Moving NACA - Enforced motion



- (d) Go to the first frame  and press play . See that the refinement is dynamically adapted as a function of time around the airfoil walls and refined some eddies in the highest vorticity regions:



Domain structure over vorticity field at frame 20 ($t = 0.2$ s)

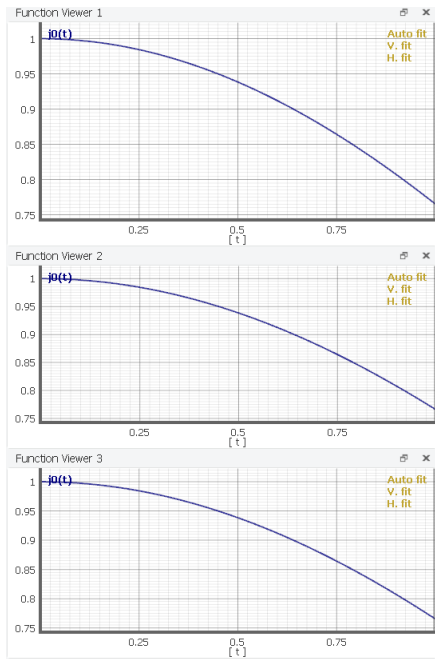
Visualize the temporal evolution of time dependent variables:

- (e) Create a *Function Viewer* window by clicking on the icon . A *Function Viewer* will pop-up, drag and drop the window in the desired location.
- (f) In the *Function Viewer* window, check that the rotation around Z of the NACA airfoil is the prescribed one: **Right click** on the *Function Viewer* > **Shapes** > **Naca-0012-4Digit** > **Euz**.

Any moving geometry has a new set of numerical data available in the Function Viewer: the position (P_x , P_y , P_z), the angular position (E_{ux} , E_{uy} , E_{uz}), the velocity (V_x , V_y , V_z), and the angular velocity

(W_x , W_y , W_z).

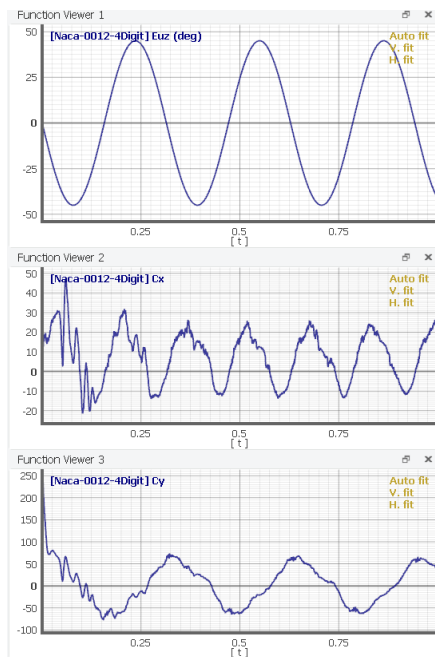
- (g) Create two additional Function Viewer windows clicking the icon . Click on each window and drag & drop them below the current Graphic View: the interface automatically adapts the positions and sizes of each window to fit the Function Viewer in the new location. Align the three Function Viewers vertically to have their X-axis aligned:



Now plot the three following quantities in the three Function Viewers: E_{uz} , C_x , and C_y of the NACA geometry:

- **Right click** on the *Function Viewer* > **Shapes** > **Naca-0012-4Digit** > **E_{uz}**
- **Right click** on the *Function Viewer* > **Shapes** > **Naca-0012-4Digit** > **C_x**
- **Right click** on the *Function Viewer* > **Shapes** > **Naca-0012-4Digit** > **C_y**

Step 6: Moving NACA - Enforced motion



Observe the drag and lift are oscillating as expected: the drag has higher values for +45 and -45 degrees while the lift has higher values for -45 degrees and lower values for +45 degrees.

Step 7: Moving NACA - Rigid body dynamics

We will now allow the NACA airfoil to move vertically under the effect of the vertical force (lift) applied on it by the fluid. The airfoil movement will simultaneously affect the flow, and this is therefore a Fluid-Structure Interaction (FSI) analysis.

7.1 Save as a new project file

Save the project with a different name (for instance `naca_rbd.xfp`): **Main menu > File > Save project** as or  in Toolbar File.

7.2 Model the airfoil with one degree of freedom in the vertical direction:

Go to **Project Tree > Geometry > Entities > Shape**: Naca-0012-4Digit, and set up the tree as follows:

- (a) **Behaviour: Rigid body dynamics**
- (b) **Behaviour > Initial conditions > Orientation: (0,0,-20)**, in order to set an initial angle of incidence of 20 degrees.
- (c) **Behaviour > Mechanical properties > Inertia tensor: Automatic**
- (d) **Behaviour > Mechanical properties > Inertia tensor > Density: 500 kg·m⁻³**

- (e) **Behaviour > Interaction properties:** Leave the default values
- (f) **Behaviour > Constraints > Translation (Global):** **Axis y**, the translation in the vertical Y direction will be allowed under the effect of the lift force.
- (g) **Behaviour > Constraints > Rotation (Local):** **Fixed**, the rotations are not allowed.
- (h) **Behaviour > Constraints > External force global >** Leave the default values
- (i) **Behaviour > Constraints > External moment global >** Leave the default values

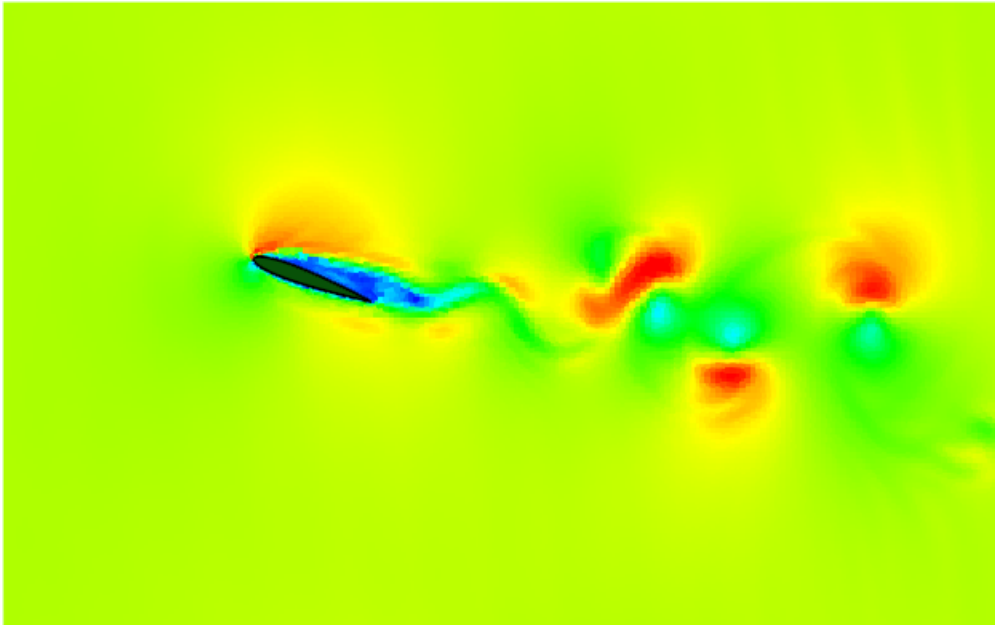


7.3 Run the case

Press the **Run button** > **Start computation**

7.4. Post-process the rigid body motion

Once the calculation has finished, visualize the time evolution of the velocity field on a cutting plane:

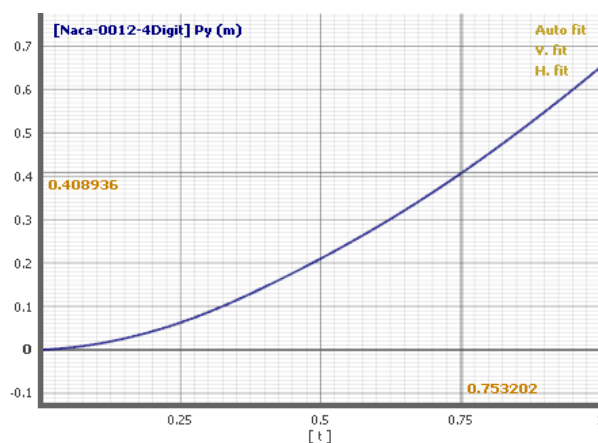


Velocity field at frame 80 ($t = 0.8$ s)

Observe the NACA moves up due to the pressure exerted by the fluid on it, since Y translation was allowed.

Display the vertical displacement of the airfoil:

- (a) **Right-click** on the *Function Viewer* window. Chose from the menu: **Shapes > Naca-0012-4Digit > Py** ; go to first frame and play forward to refresh the data in the *Function Viewer*



Evolution of the airfoil elevation in Y direction (P_y)

In the P_y curve plotted in the *Function Viewer*, it can be checked that the elevation of the Naca airfoil

under the lift force is around 0.65 m after 1 second of simulation time.

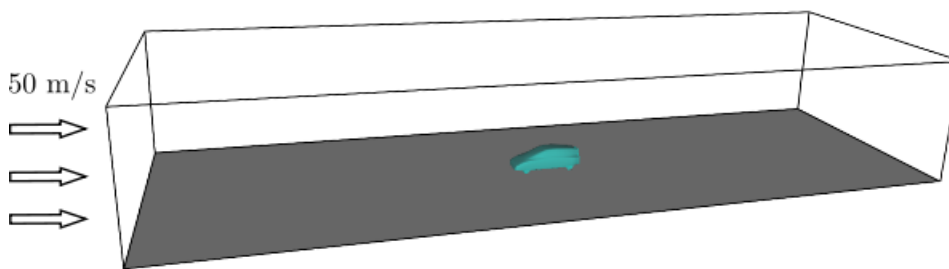
(b) Save the vertical displacement in a file, by **Right clicking** on the *Function Viewer* window and choosing **Export current data**

(a) Check that the X and Z displacements, and all the rotational Euler angles are constant for the Naca-0012-4Digit:

- **Right click** on *Function Viewer* window > **Shape > Naca-0012-4Digit > Px**
- **Right click** on *Function Viewer* window > **Shape > Naca-0012-4Digit > Pz**
- **Right click** on *Function Viewer* window > **Shape > Naca-0012-4Digit > Eux**
- **Right click** on *Function Viewer* window > **Shape > Naca-0012-4Digit > Euy**
- **Right click** on *Function Viewer* window > **Shape > Naca-0012-4Digit > Euz**

Tutorial 02 - Vehicle aerodynamics

This is the second tutorial of XFlow. It illustrates the setup and solution of the three-dimensional air flow around a reference vehicle (the ASMO model) and the study of the forces generated by the flow on the vehicle. This is a typical external aerodynamic case using the virtual wind tunnel.



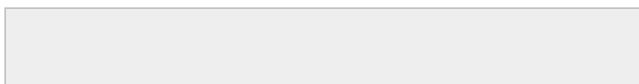
This tutorial shows how to:

- Import a geometry from a CAD file
- Check the imported model
- Work with the virtual wind tunnel
- Explore the log file
- Visualise flow variables on the vehicle surface
- Visualise isosurfaces and stream lines
- Visualise a custom field
- Create an animation
- Analyse aerodynamic forces on the geometry model

At this point, it is assumed that the reader has completed Tutorial 01. Thus, some steps in the setup and post-process will not be described in detail.

Some of the steps described in this tutorial requires the user to switch to **Expert** mode in **Main menu > Options > Preferences > Application > Application mode: Expert**, or  in Toolbar menu.

Before starting the tutorial, please download the project data files from the XFlow[version]_TutorialsPack.zip folder.



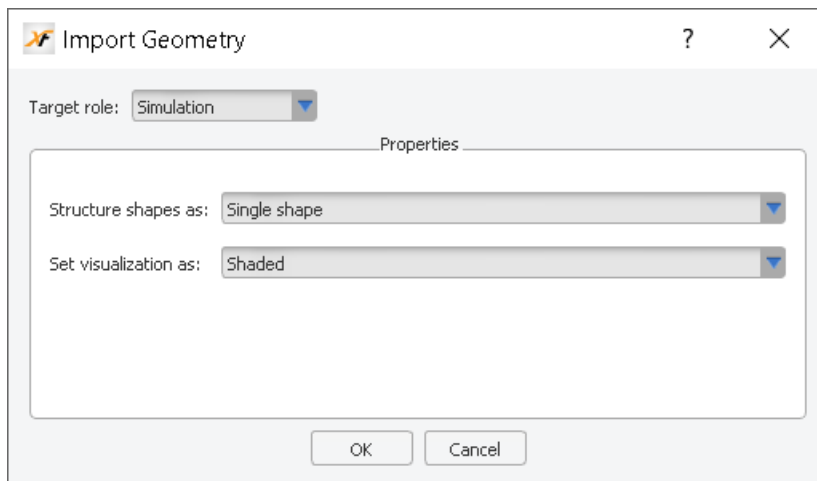
Contents

- [Step 1: Import geometry](#)
- [Step 2: Set up the problem](#)
- [Step 3: Run](#)
- [Step 4: Post-processing](#)

Step 1: Import geometry

1.1 Import the geometry

Main menu > Geometry > Import a new geometry (or use the toolbar icon ) and select the file `asmo.nfb`. The Model Units window, shown in the figure below, will appear automatically. Select Target role: "Simulation", "Single shape" structure, visualization "Shaded" and press **OK**.



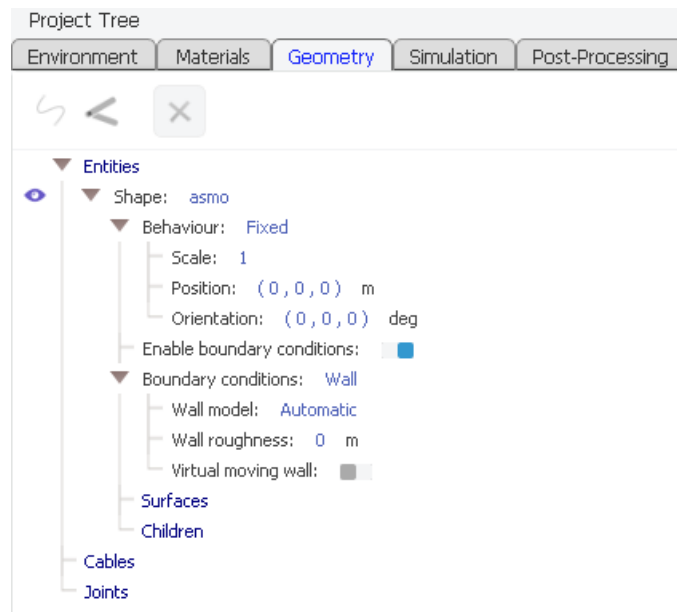
Import geometry dialog window

The imported geometry will be shown in the Graphic View, and it will appear as a **Shape** in the **Geometry** section of the *Project Tree*.

Step 1: Import geometry



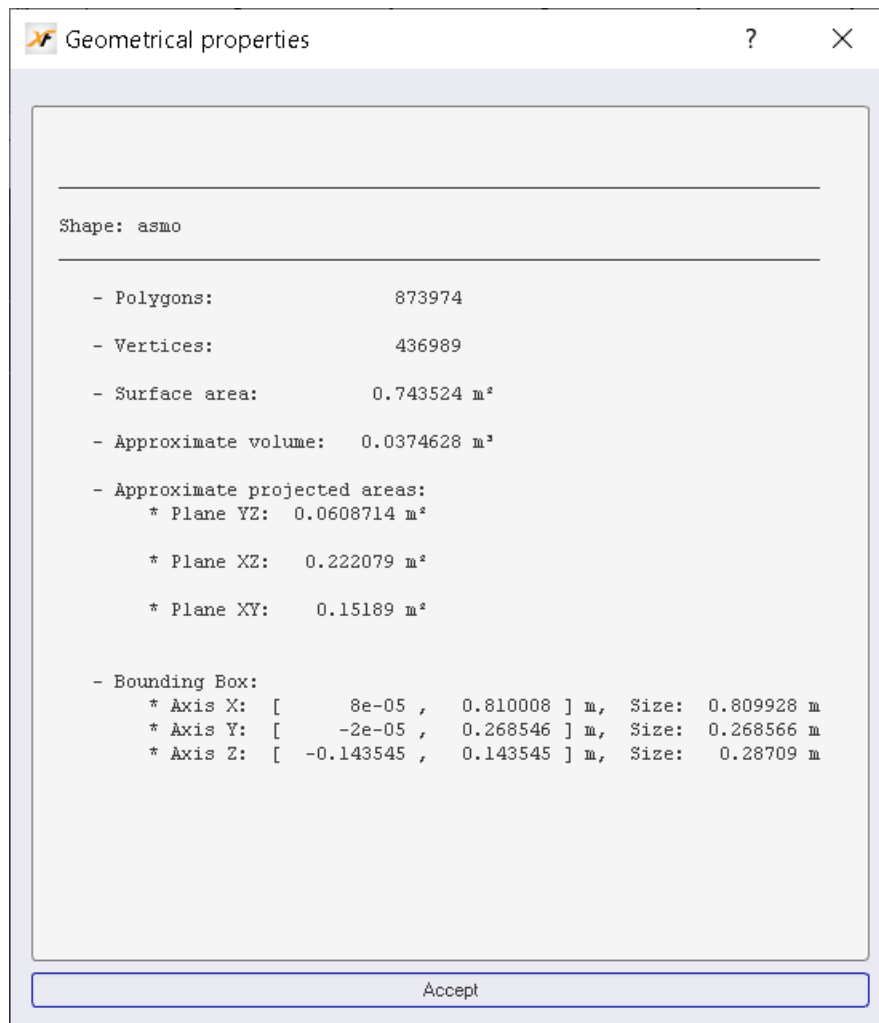
Asmo geometry shown in the Graphic View



Asmo geometry shown in the Geometry tree

1.2 Check the model

- Check the orientation of the model in the *Graphic View* window. Remember that the Y direction is the height and that the flow in the wind tunnel goes from -X to +X.
- Show geometrical information of the model by selecting the geometry, **right clicking** on the Graphic View window and choosing in the **Graphic View Menu > Show geometrical properties**



ASMO: Geometrical properties

⚠ Check: Make sure that **Mesh deflection** = 0.1. If that is not the case, please change the parameter to 0.1, delete the geometry: Select geometry > **right mouse button** in the *Graphic View* > **Graphic View menu** > **Remove selected**, and import it again (see [above](#)). The geometry has to be created again in order to take into account the change of the **Mesh deflection** value.

(c) Measure the dimensions of the model. Select the geometry object and press **Main Menu** > **Geometry** > **Dimensions** or

- Length (X direction): 0.809928 m
- Height (Y direction): 0.268566 m
- Width (Z direction): 0.28709 m

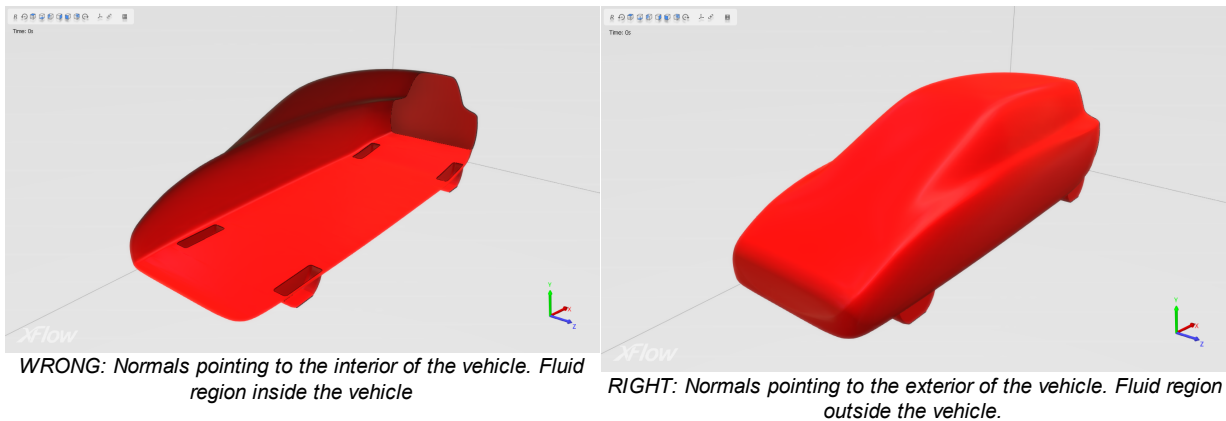
Keep the geometry selected and click again to hide the dimensions.

(d) Check the quality of the surface tessellation. To this end: select the geometry object, **right click** on

Step 1: Import geometry

the *Graphic View* window and choose **Graphic View Menu > Visualisation mode > Mesh**

(e) Check the surface normals enabling the culling (the option is On by default); thus select the geometry object, **right click** on the *Graphic View* window and choose **Back-face culling**. Now reverse the orientation of the normals to learn the different representations of the vehicle depending on the normals orientation: **Reverse orientation**. Eventually, leave the normals pointing to the exterior of the vehicle as shown in the figure below on the right.



⚠ Please note: It is essential that the surface normals point to the fluid region. You can see where the normals are pointing by means of the Back-face culling option.

(f) Check that the model is watertight: select the geometry and select **Main menu > Geometry > Show/hide holes**. No warnings should be printed in the Message view and no white edges should be shown when displaying the geometry in the bounding box mode: **Graphic View Menu > Visualisation mode > Bounding Box**

⚠ Please note: If the geometry is not watertight, fluid will leak inside it which will be initialised with the inlet velocity condition, as the rest of the fluid. However, since this in this region the air is confined, it will generate pressure waves inside the geometry which will lead to wrong values of the aerodynamic forces. Furthermore, closed volumes with a small opening will equilibrate the interior pressure to the local static pressure at the hole, leading to wrong overall forces.

Some holes can be directly repaired in XFlow: **Main menu > Geometry > Healing**. See Tutorial 07 for more information about the healing feature.

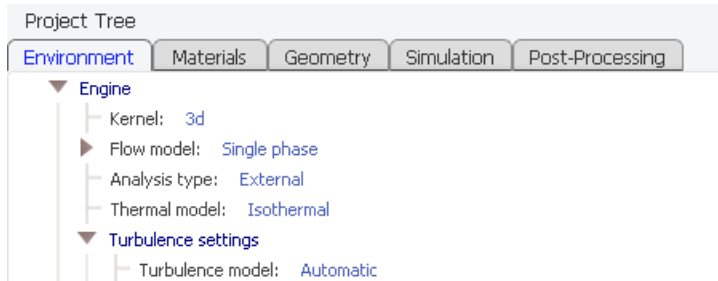
Step 2: Problem setup

2.1 Engine settings

Project Tree > Environment > Engine:

(a) **Kernel:** 3d

- (b) Flow model: **Single phase**
- (c) Analysis type: **External**
- (d) Thermal model: **Isothermal**
- (e) Turbulence settings > Turbulence model: **Automatic**



2.2 Environment settings

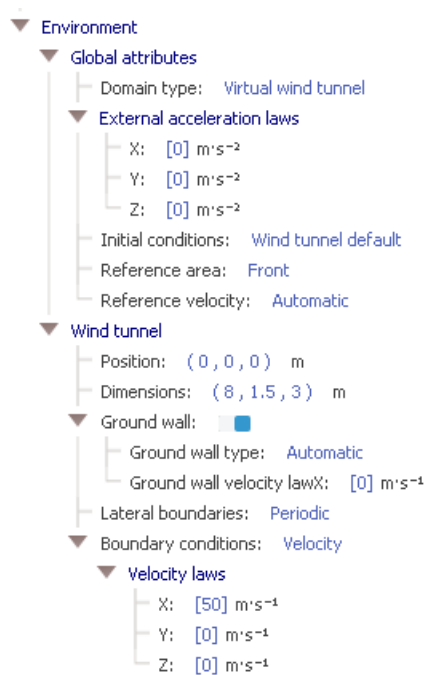
Project Tree > Environment > Environment:

- (a) Global attributes > Domain type: **Virtual wind tunnel**
- (b) Global attributes > Ext. acceleration laws: leave it to zero
- (c) Global attributes > Initial conditions: **Wind tunnel default**

 **Tip:** When using the **Wind tunnel default** initial condition, XFlow initialises the velocity field according to the inlet **Velocity laws**. This is a good practice for external aerodynamic simulations.

- (d) Keep **Reference area** as **Front** and **Reference velocity** as **Automatic**
- (e) **Wind tunnel > Dimensions:** **(8, 1.5, 3) m**. This leads to a blockage ratio - vehicle frontal area (YZ) divided by the wind tunnel cross-section - of 1.38%
- (f) **Wind tunnel > Ground wall:** **On**. **Ground wall type:** **Automatic**. It is possible to specify a velocity for a moving ground, but in this example the ground will be fixed (zero velocity law X)
- (g) **Wind tunnel > Lateral boundaries:** **Periodic**. Symmetric lateral boundaries could be used as well
- (h) **Wind tunnel > Boundary conditions:** **Velocity**. Set the **Velocity laws** at inlet equal to **(50, 0, 0) m·s⁻¹**

Step 2: Problem setup



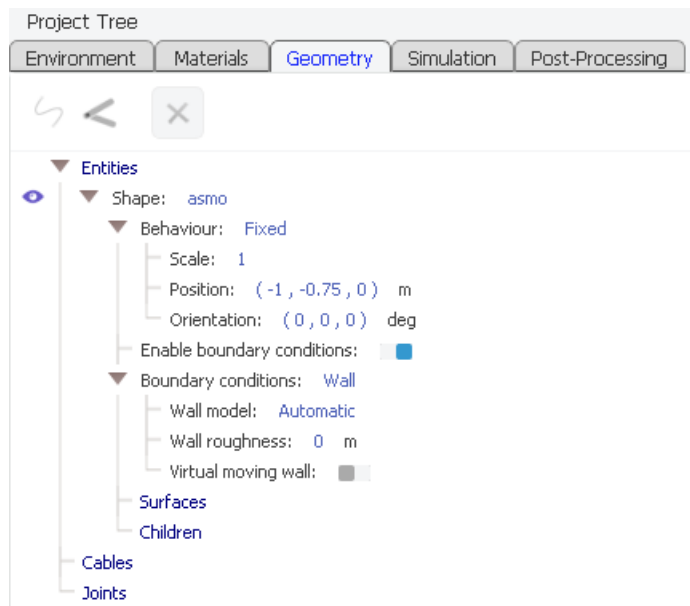
2.3 Material settings

Keep the default parameters of **Material 1** (standard air at 15 degrees Celsius) in **Project Tree > Materials**.

2.4 Geometry settings

Project Tree > Geometry > Entities > Shape: asmo

- The asmo geometry should be located just over the ground, leaving room behind the vehicle for the wake development. Hence, the object position should be changed to (**Behaviour > Position**) **(-1, -0.75, 0) m** and make sure that the wheels are touching the ground,
- Set the **Boundary Conditions** to **Wall**, with **Wall model: Automatic** and **Wall roughness: 0 m**.



2.5 Simulation settings

Project Tree > Simulation:

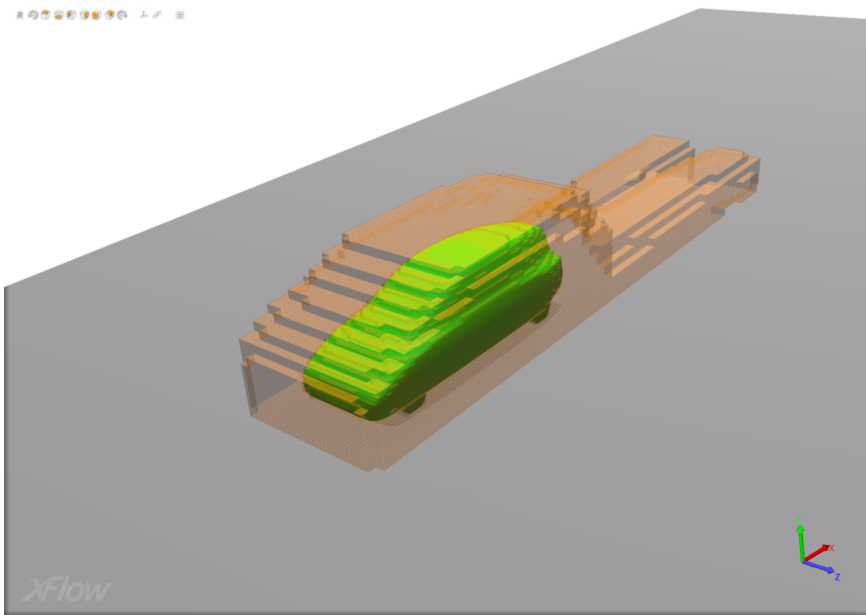
- (a) Time > Simulation time: **0.15 s**
- (b) Time > Time step mode: **Fixed automatic**
- (c) Time > Time step mode > Courant : **1**
- (d) Lattice discretization > Resolved scale: **0.2 m** (resolution at the far field)
- (e) Lattice discretization > Refinement algorithm: **Adaptive refinement**
- (f) Lattice discretization > Shapes refinement > Shape: **asmo** > Target resolved scale: **0.0125 m** (this represents the near wall resolution).
- (g) Lattice discretization > Adaptive refinement > Wake resolution: **0.0125 m**, equal to the asmo resolution.

Set **Wake distance control: On**.

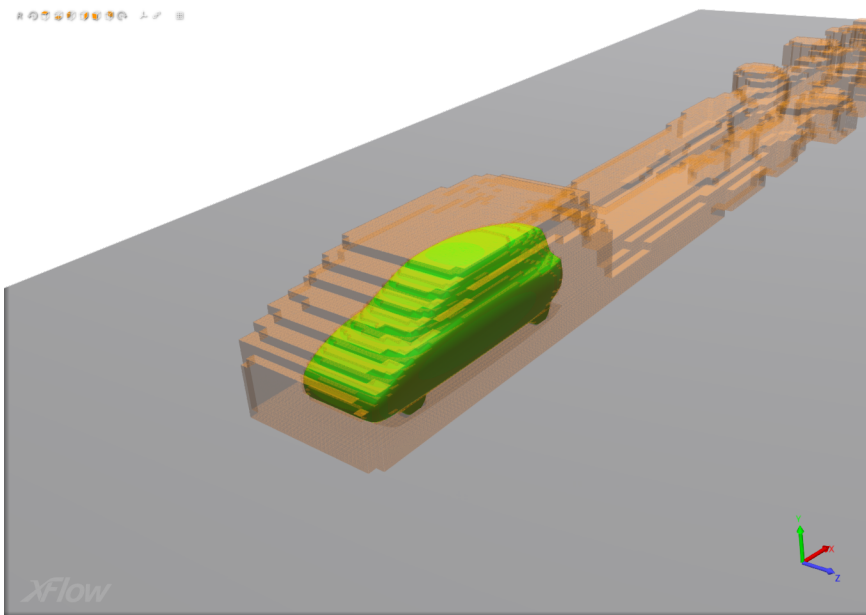
Distance : 1 m. Approximately 1 time the characteristic length.

⚠ Please note: The wake distance control allows to limit the length of the refined wake in the flow normal direction starting from the geometry end. It allows to limit the increase of the number of elements in the wake.

Step 2: Problem setup



Wake refinement control enabled.



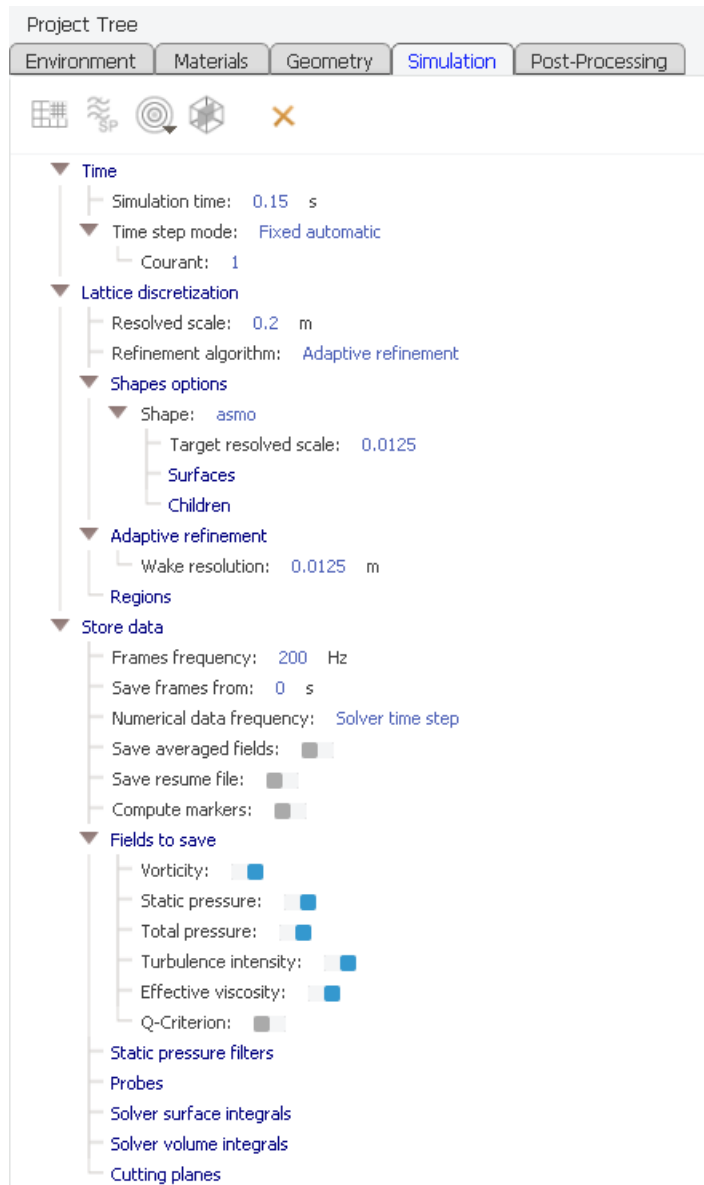
Wake refinement control disabled.

 **Tip:** To visualize just one lattice level at the *Graphic View*: plot the domain structure in **Post-Processing > General > Show > Domain structure**, **left clicking** on the Graphic View and press +/- button to change the Min lattice level, and SHIFT +/- to change the Max lattice level.

 **IMPORTANT:** Due to the Cartesian structure of the lattice, the choice of the different resolutions must be related by a **factor 2^n** , with n an integer. In this tutorial, $n = 4$: $h = 0.2 / 2^4 = 0.0125$ m near the wall. The wake is developing only because of the flow surrounding a body, therefore the wake resolution must

be equal or greater than the near wall resolution.

- (i) Set a **Frames frequency** of **200 Hz** and **Numerical data frequency** to **Frames frequency**; this will save hard disk space although the curves in the *Function Viewer* will have less points than if Solver time step frequency had been chosen.
- (g) Leave disabled the computation of averaged fields and markers, as well as resume file. Leave all the **Fields to save**: **On**, except for the Q-Criterion.



⚠ Please note: In aerodynamic applications it is essential to properly resolve the turbulent wake. The *Adaptive refinement* algorithm is well suited for such applications and one must specify enough simulation time to let the wake develop: typically 6 or 7 times the characteristic time (length of the vehicle

Step 2: Problem setup

divided by the inlet velocity).

⚠ Please note: The adaptive wake refinement is not yet supported for DMP (MPI) simulations.

Step 3: Run

3.1 Save project

Save the project before running the computation:

Main menu > File > Save project, or  in Toolbar File.

XFlow project files have the extension `.xfp` (uncompressed format) or `.xfz` (compressed format which includes setup files and geometries).

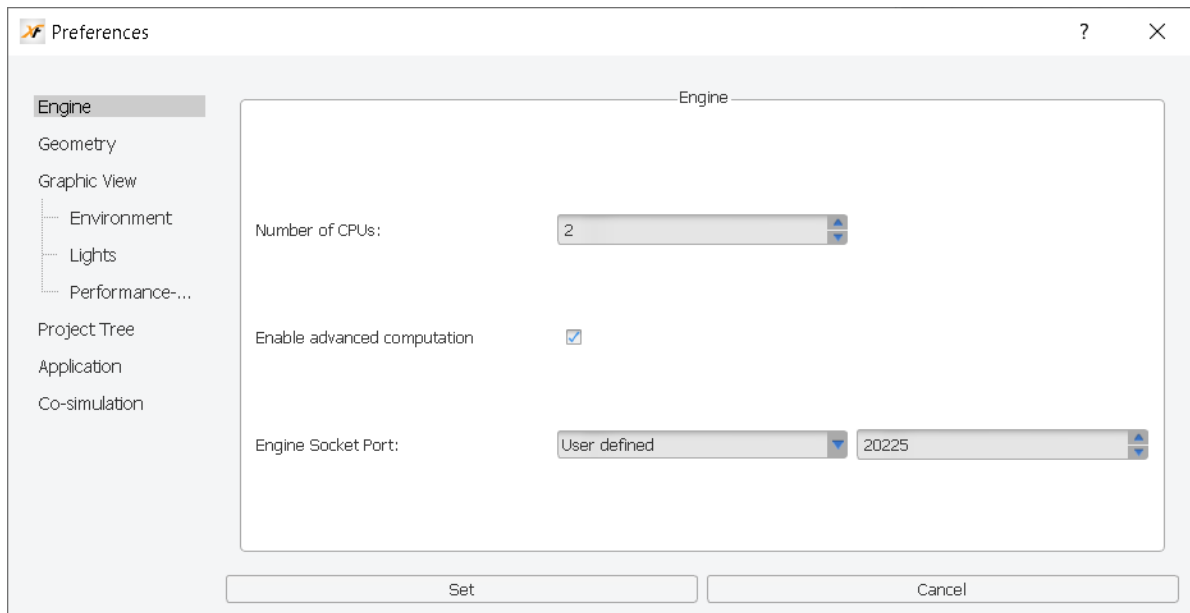
💡 Tip: You can directly load the setup of this problem from the project file `asmo.xfp`

3.2 Set number of CPUs

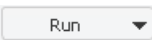
Set the number of CPUs, to be used for the calculation, in the preferences of the engine:

Main menu > Options > Preferences > Engine, or  in Toolbar File.

One or two processors are enough for this tutorial.



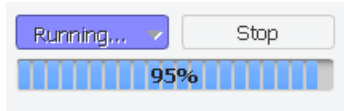
3.3 Start the computation

Click the **Run** button  > **Start computation**

XFlow creates a folder called `asmo` in the same location as the project file. The resulting data and log files

are saved in this folder.

As XFlow is calculating, engine messages are shown in the *Message View* window and the progress bar moves between 0 and 100%. A new data file (frame) is written when the progress bar gets to 100%.



The XFlow GUI will switch to Post-processing mode and all the project tree's tabs but the Post-Processing one will be disabled.

3.4 Explore the .log file

The output displayed in the *Message View* window is also written in a text file called `project_name.log` in the simulation folder `asmo`.

The log file contains important information, such as:

- Number of CPUs detected
- Levels of refinement
- Number of active fluid elements in each refinement level
- Total number of elements
- Reference values (used to calculate the aerodynamic coefficients)
- Time step

After a first analysis of the simulation domain, the data file 0 is written and the computation of the flow starts.

For each time step, XFlow displays the total accumulated simulation time, the Stability parameter, the wall clock time (in seconds) to compute the current time step, and the number of elements at each time step. The latter varies as the wake develops since the **Refinement algorithm** is set to *Adaptive refinement*. Each line corresponds to a new solver time step.

Example: .log file output.

```
-----
[ INFO ] Saving data...
[ INFO ] [[Data file]] 0 done!!! | Frame wall clock time[0]s | Overall wall clock time[0]s | Num elements
[72605]
[ INFO ] Initializing distance field...
[ INFO ] done!
[ INFO ] Sim. time [3.333333e-04]s | Stability param. [2.968628e-01] | Wall clock time [9.609799e-01]s
[ INFO ] Sim. time [6.666667e-04]s | Stability param. [3.437246e-01] | Wall clock time [6.756830e-01]s
...
[ INFO ] Sim. time [5.000000e-03]s | Stability param. [3.055790e-01] | Wall clock time [6.620450e-01]s
```

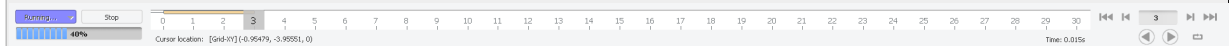

Step 3: Run

Example: .log file output.

```
[ INFO ] Saving data...  
[ INFO ] [[Data file]] 1 done!!! | Frame wall clock time[11.1652]s | Overall wall clock time[11.1652]s |  
Num elements[74733]  
[ INFO ] Sim. time [5.333333e-03]s | Stability param. [2.995797e-01] | Wall clock time [6.761091e-01]s  
...
```

Every time a new frame is calculated the following message is printed: *[[Data file]] #frame done!!!*; the written frame can then be selected in the timeline to load its data for post-processing.

 **Tip:** As the calculation progresses, it is possible to post-process the calculated frames even when the simulation has not reached the final time yet. However, this uses computer resources and therefore slows down the calculation. It is recommended to set the timeline to an empty frame when not post-processing, as shown below:



 **Tip:** Run the case setting with a different number of CPUs and compare the computation time. The optimal number may depend on the size of the problem and hardware configuration.

Step 4: Post-processing

Main menu > Simulation data > Load simulation data or  in Toolbar Data Processing.

4.1 Visualise the solution

4.1.1 Visualise a flow variable on a cutting plane

Create a cutting plane. Go to the **Project Tree > Post Processing** tab and:

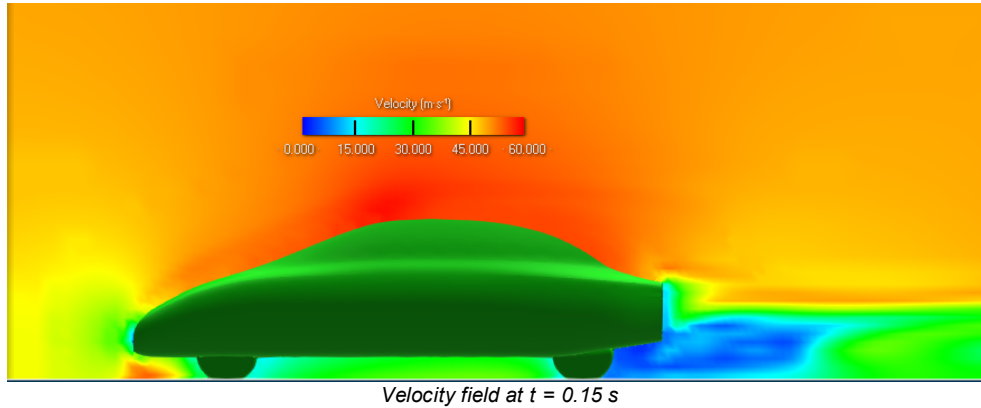
- Right-click** on **Cutting planes > Add cutting plane** or press  in the Post-Processing toolbar
- Plot the Cutting plane with a Z-axis normal, and set its position 0.1 m in the z-axis: **Cutting plane 1 > Position: 0.1 [m]**
- Visualise the velocity in the cutting plane by setting: **Cutting plane 1 > Visualisation mode: 3d field**, and choose the **Field: Velocity**
- Activate the interpolation mode: **General > Interpolation mode: Convolution**
- Set the velocity range to [0,60]: select the top and bottom numbers of the colour bar range or **Main**

menu > Simulation data > Analysis settings or press  in Toolbar Data Processing.

(f) Select the geometry and set the right view 

(g) Press Play 

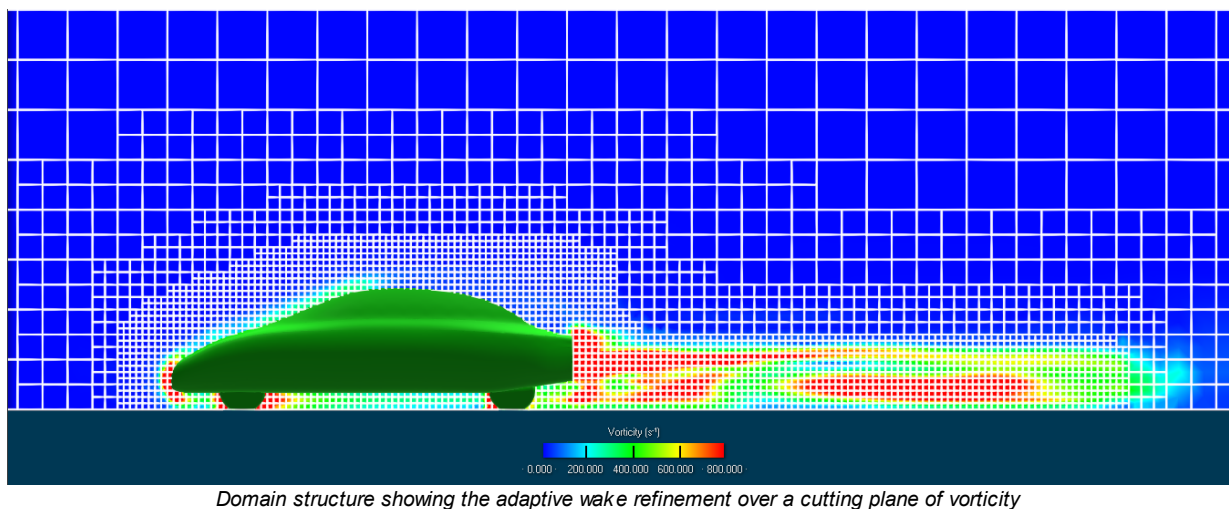
Note that the internal domain is initialised (frame 0) with the inlet velocity of the virtual wind tunnel.



Go back to point (c) and visualise other quantities such as Vorticity, Static pressure or Turbulence intensity, which provide meaningful insight about the flow.

4.1.2 Visualize the adaptive wake refinement

Set the previous generated cutting plane to show Vorticity field and set a range of [0,800]. Add an additional cutting plane at the same location ($z=0.1 \text{ m}$) but now set the **Visualisation mode: > Domain structure** to observe the lattice domain evolution against the time. The wake is now refined dynamically behind the car based on the vorticity gradients up to 1 m of distance from the car geometry.



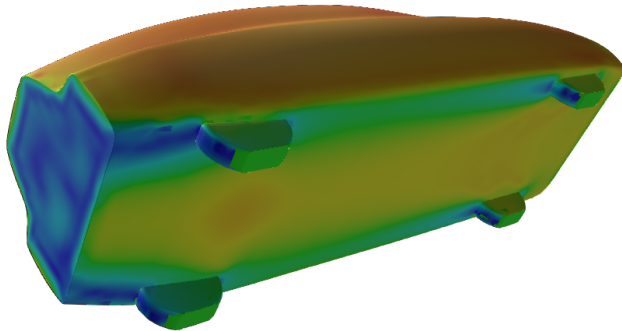
4.1.3 Visualise a flow variable on the vehicle surface

Step 4: Post-processing

Hide the cutting plane by clicking on the icon .

Go to **Project Tree > Post-Processing > General > Show** and:

- (a) **Surface info: 3d field.** A 3d field will be represented on the vehicle surface.
- (b) Choose the field, e.g. velocity: **Surface info > Field: Velocity**
- (c) Enable the **Surface info** by clicking on the icon , to see the projection of the velocity over the vehicle in the *Graphic View*:

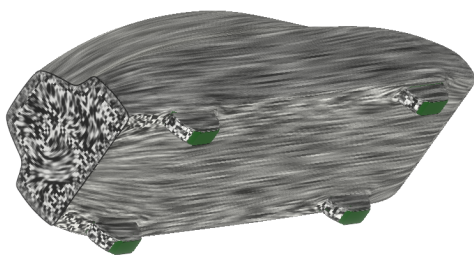


Velocity field projected on the vehicle surface at $t = 0.15$ s

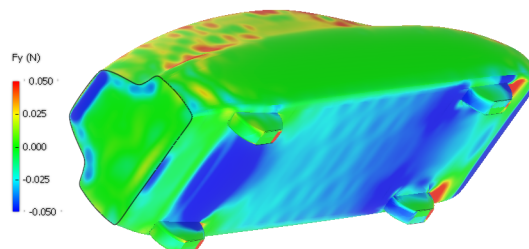
- (d) Hide the Surface info by clicking on the icon .

Note, that the following surface quantities can be further visualised in **Surface info**:

- **LIC** (Line Integral Convolution) of the velocity field
- **Cp** pressure coefficient
- **Cf** skin friction coefficient
- **Y+** dimensionless distance from the wall
- **P+** wall pressure
- **Heat transfer coefficient**
- **Fx** distribution of force contribution on X-axis
- **Fy** distribution of force contribution on Y-axis
- **Fz** distribution of force contribution on Z-axis



LIC at $t = 0.15$ s, Noise size: 20, Motion blur amount: 1



Fy (downforce) distribution at 0.15 s.

4.1.4 Create a custom field

The user can create a custom field for post-processing using the following system variables:

t	→ Time
x	→ Spatial coordinate
y	→ Spatial coordinate
z	→ Spatial coordinate
pi	→ Number π
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)
sp	→ Static pressure
rho	→ Reference density defined in the <i>Materials</i> tab
vmod	→ Module of the velocity
vrt	→ Vorticity
cf	→ Skin friction coefficient
ti	→ Turbulence intensity
tp	→ Total pressure
u	→ Temperature
nx	→ X-component of the surface normal
ny	→ Y-component of the surface normal
nz	→ Z-component of the surface normal

(a) To create a custom field go to **Project Tree > Post-Processing**, do **right-click** on **Custom fields** and choose **Add custom field** or press  in the Post-Processing Toolbar

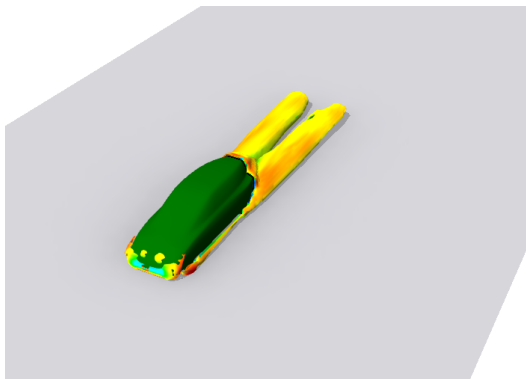
(b) The user can define the custom field in **Custom Fields > Custom field 1 > Expression**. For instance, to define the pressure coefficient C_p : **Expression: $[sp/(0.5 \cdot \rho \cdot 50^2)]$** ($V_{ref} = V_{inlet} = 50 \text{ m} \cdot \text{s}^{-1}$)

(c) Switch the visualisation field of the Surface info: **Custom field 1** to observe the pressure coefficient on the surface.

4.1.5 Visualise isosurfaces

Go to **Project Tree > Post-Processing > Isosurfaces** and:

- (a) Create an isosurface by **right-clicking** on **Isosurfaces > Add isosurface** or press  in the Post-Processing Toolbar
- (b) Choose a field to be visualised in the isosurface, e.g. the vorticity: **Isosurfaces > Isosurface 1 > Field: Total pressure**
- (c) Set the **Value** to 500 Pa. **Isosurfaces > Isosurface 1 > Value: 500**. XFlow will plot an isosurface representing all the fluid elements that have a total pressure value of 500 Pa.
- (d) The user can colour the isosurface by other fields (e.g. vorticity) rather than the isosurface field in: **Isosurfaces > Isosurface 1 > Coloured by field: Vorticity**
- (e) Refresh the isosurface.

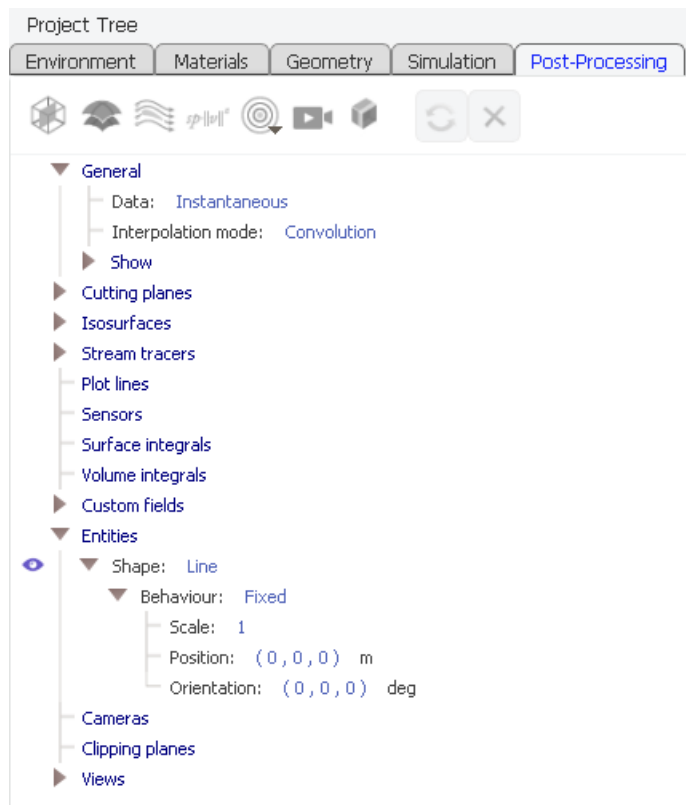


Isosurface of vorticity at $t = 0.15$ s coloured by Velocity field

- (f) Go to point (c) and change the **Value** (between the min and the max values of total pressure colour bar). Recompute the isosurface by **right-clicking** on **Isosurface 1** and select **> Recompute isosurface** or press , which appears next to **Isosurface 1**. It may take a few seconds to calculate the isosurface.
- (g) Hide the isosurface by clicking on the icon .

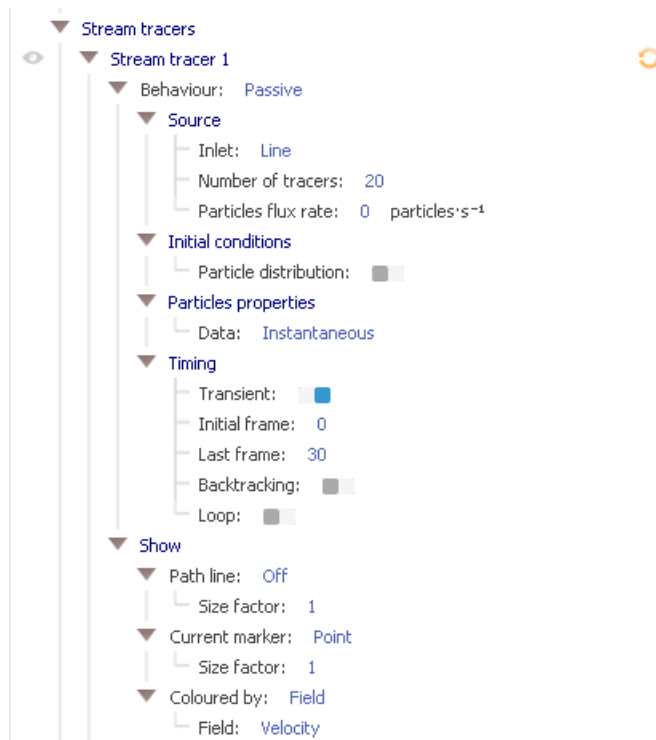
4.1.6 Visualise stream tracers

- (a) Create the line, where the stream tracers are released from. **Main menu > Geometry > Create object > Create line** or . Choose **From two vertices**, where first vertex is **(-1.5, -0.65, -0.25)** and the second is given by **(-1.5, -0.65, 0.25)**. The line will appear in the **Post-Processing > Entities** section of the **Project Tree**. Geometries created once simulation data is available can only be used for post-processing purposes.
- (b) Let the line **Behaviour** as **Fixed**.



- (c) Create a stream tracer in **Project Tree > Post-Processing**. **Right-click** on **Stream tracers** and choose **Add stream tracer** or press  in the Post-Processing Toolbar
- (d) Set the behaviour to **Passive** and select the geometry shape that is going to be the source of tracers, **Source > Inlet: Line**.
- (e) Set the other parameters as follows:
- Number of tracers: 20**
 - Particles flux rate: 0**
 - Data: Instantaneous**
 - Transient: On**
 - Initial frame: 0**
 - Last frame: 30**
 - Backtracking: Off**
 - Loop: Off**

Step 4: Post-processing



(e) **Right click** on **Tracer 1** and select **Recompute**. Save the project. It will take a few minutes.

(f) Note that a second timeline appears in pink colour (over the yellow timeline), this is the timeline of the stream tracers. Press play  and observe the particles trajectory.

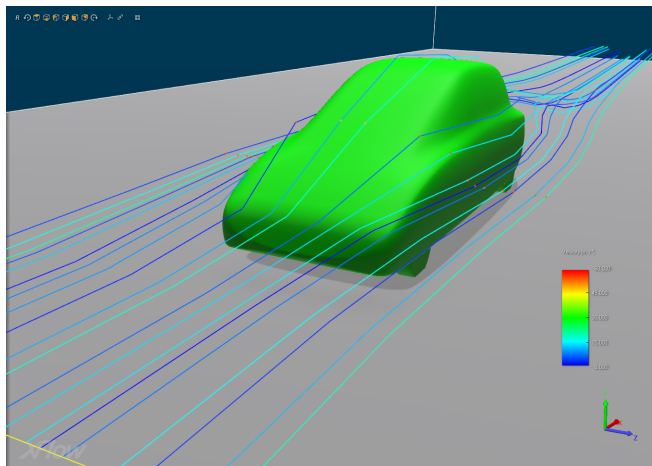
(g) Show the streamlines in **Post-Processing > Stream tracers: Stream tracer 1 > Path lines**.

Set **From frame: 0** and **To frame: 30** to indicate the stream tracer frames to use in order to plot the path lines of the stream tracers. In our case we will draw path lines from the beginning (frame 0) until the end (frame 30)

(h) Show the streamlines in **Post-Processing > Stream tracers: Stream tracer 1 > Show > Path line: Line**

(e) Set the field **Coloured by: Field** and select **Velocity**

(f) **Right click** on **Tracer 1** and select **Generate path lines** to create the streamlines or just press  next to **Path lines**. The path lines are now visible



Stream tracers and pathlines at $t = 0.015\text{ s}$

(i) Hide the stream tracer by clicking on the icon 

 **Please note:** See [Tutorial 10](#) for more details on the computation of stream tracers.

4.1.7 Visualise vector field

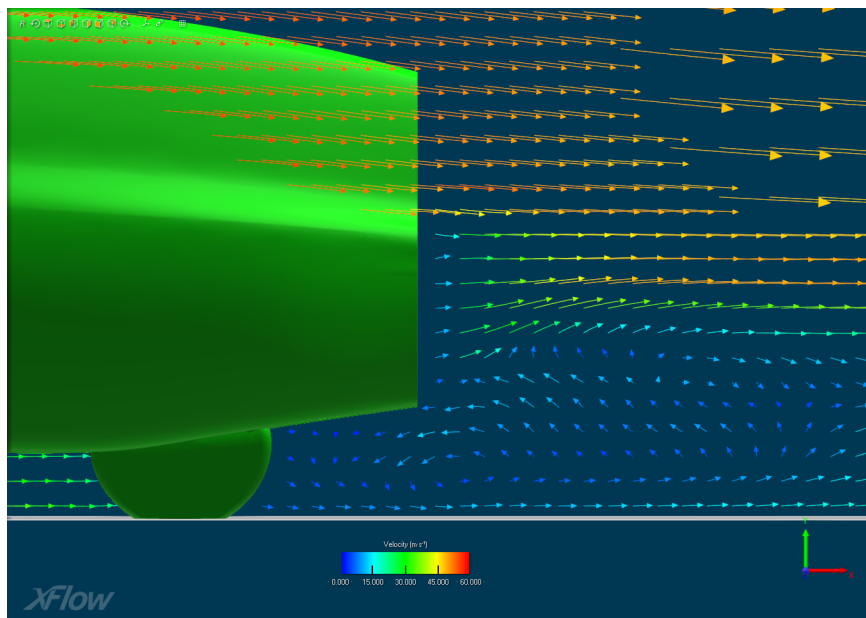
(a) Activate the **Cutting plane 1** again by clicking on the icon 

(b) Select **Visualisation mode: Vectors**

(c) Select **Field: Velocity**

(h) Adjust the **Arrows Density** and **Arrows length** with the slider

(i) To refresh press the  icon



Vectors plot coloured by Velocity.

Step 4: Post-processing

4.1.8 Visualise surface field distribution

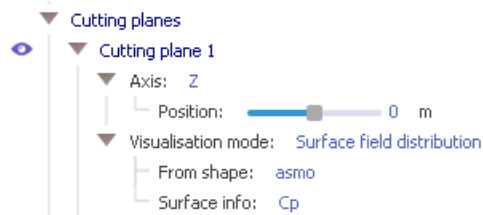
It is possible to project on the vehicle surface the distribution of the pressure coefficient C_p :

(a) Set the cutting plane position to 0 m.

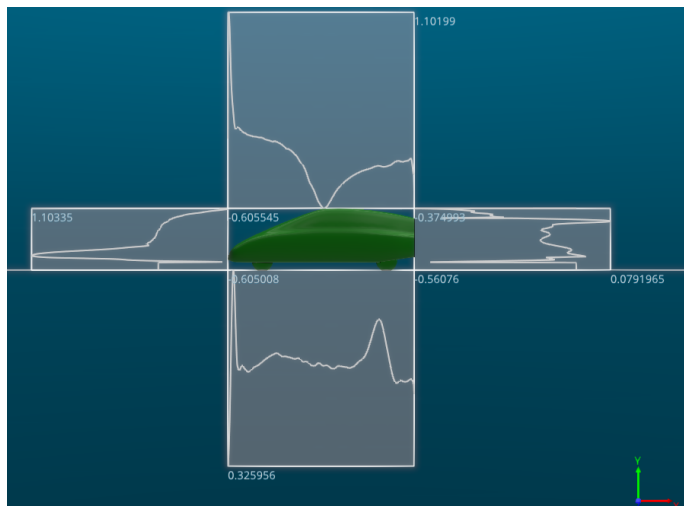
(b) Switch to the **Interpolation mode: Convolution**

(c) Switch the cutting plane visualisation mode to **Surface field distribution** and select **Surface**

Info: C_p



(d) Note that you can specify on which geometry you will project the field by the option **From shape: asmo**.



Surface field distribution of C_p at $t = 0.15$ s

The *Graphic View* window will now show the surface distribution as illustrated in the figure above. The four graphs show the projection of the pressure coefficient C_p on the +Y, -Y, -X and +X side of the geometry. They can be saved in text files by **Main menu > Simulation data > Export data of cutting plane distribution**.

The default names for the files are `field_distribution_1_minusY.txt` for distribution of C_p at the bottom of the vehicle (underbody), `field_distribution_1_minusX.txt` at the front of the vehicle, `field_distribution_1_plusX.txt` at the back (base) and `field_distribution_1_plusY.txt` at the top (roof).

(e) Hide the cutting plane by clicking on the icon 

4.1.9 Work with averaged data

XFlow can save three types of data: Instantaneous, Averaged and Standard deviation. By default only Instantaneous is saved. Averaged data can be saved if the following option is enabled: **Project Tree > Simulation > Store data > Save averaged fields**

If **Save averaged fields** is enabled, it is also possible to save standard deviation and RMS (calculated from averaged and Standard deviation) data.

To save average or standard deviation and RMS data, the user needs to switch to **On** this option **before** running the calculation. Then, in the post-process, the user will be able to choose the type of data to be visualised: **Post Processing > General > Data: Averaged**.

4.2. Analyse aerodynamic forces

The evolution in time of the aerodynamic forces can be observed directly in the *Function Viewer*. By default no function viewer is set in the XFlow GUI, so the user has to create a new one **Main menu >**

Window > New Function Viewer or by selecting the icon .

(a) Do **right click** on the *Function Viewer* window, the *Function Viewer* drop-down menu will appear

(b) Select **> Overall forces**. The user can choose here among the different aerodynamic coefficients and forces:

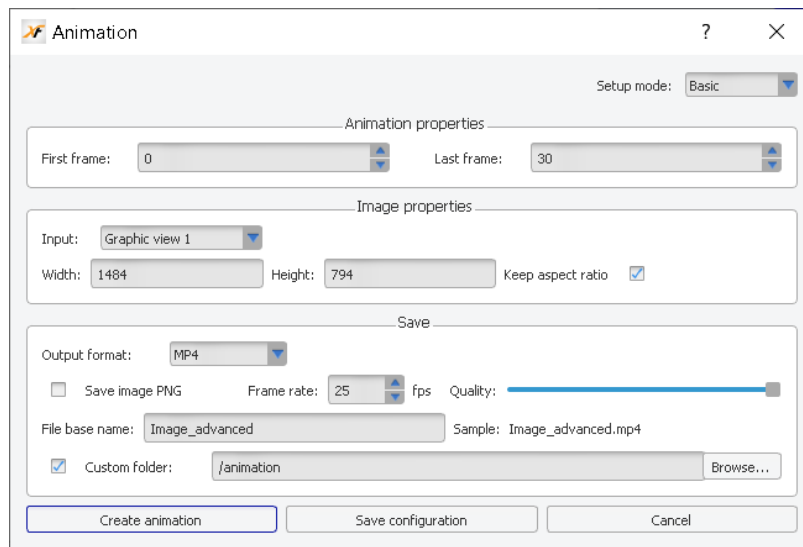
Cx, Cy, Cz	→ Force coefficients in each direction
Fx, Fy, Fz	→ Force in each direction

 **Please note:** The drag coefficient obtained with this coarse resolution is far from the experimental values 0.153-0.158. You should refine the resolution near the walls and in the wake (e.g. target scale around 0.0025 m) to properly resolve the turbulent wake and obtain accurate drag values. The validation and verification of XFlow results for this problem can be found in the Validation Guide.

4.3. Create an animation

The user can create an animation by saving a sequence of images and using an external software to assemble the images sequence to a video file. The XFlow animation wizard is available in **Main menu > Post-Processing > Animation**, or  in the Data Processing toolbar.

Step 4: Post-processing



Basic animation wizard

4.3.1 Basic animation

Set a cutting plane as per point 4.1.

- Main menu > Post-Processing > Animation**
- Select **Setup mode: Basic**
- Select **First frame** and **Last frame** for the sequence, e.g. 0 and 30
- Select the **Input**: "Graphic View 1" for creating images just of the Graphic View, or "GUI view" for creating video of the entire graphical interface
- Enter an **File base name** for video name, the path of the **Folder** where to save it and press **Create Animation**. It will create a video file in the directory `$Working-directory/asmo/animation`.

4.3.2 Advanced animation

- Main menu > Post-Processing > Animation**
- Select **Setup mode: Advanced**
- Duration: 1 s**, this represents the duration (real time) of the animation
- Image frequency: 100 Hz**, frames per second. Since the duration is 1 second, the animation will consist of a sequence of 100 images.
- Animation time: 0.15 s**. This implies, that time will be frozen at $t = 0.15$ s for the animation, thus only the results of the last frame of the *simulation* will be shown in the image sequence.
- Go to **Animated items > Cutting planes** and press **Add**. Select **Cutting plane 1**.
- Leave blank the **Visibility law** of the Cutting plane 1, this means that the cutting plane will be visible during the whole animation.
- Cutting plane 1 > Position law: $t_{\text{animation}}$** . This variable ranges from 0 to the animation

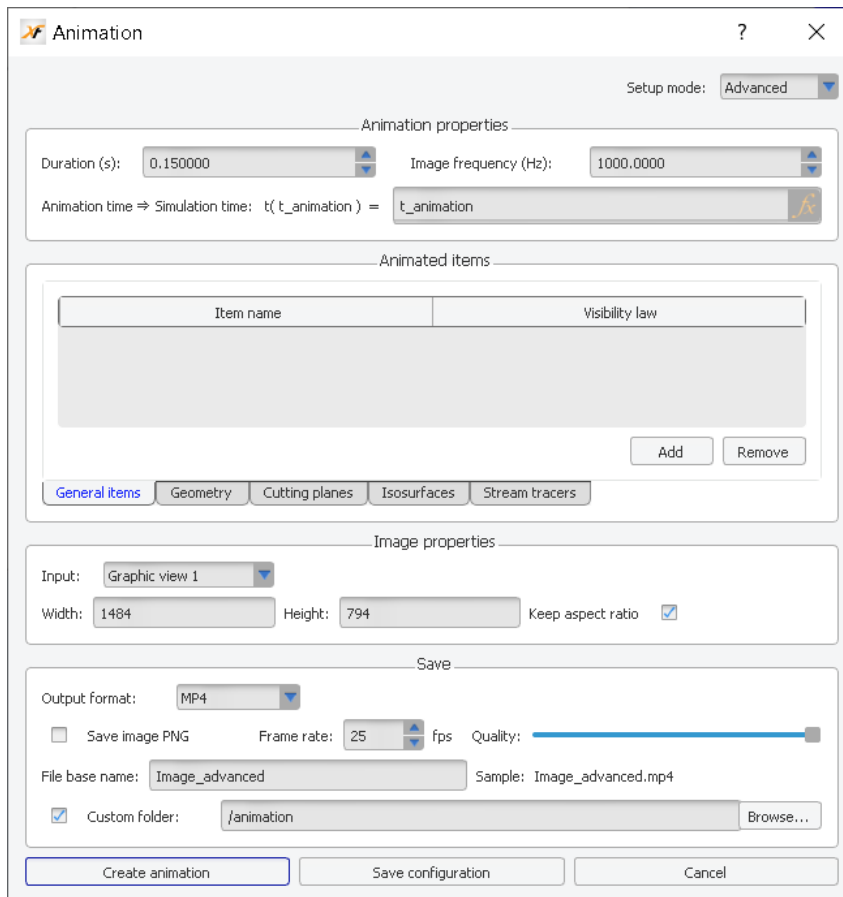
duration, in this case duration is 1 s. Hence, 100 cutting planes will be created sweeping the entire domain. In this wizard the cutting planes positioning are defined by an adimensional number ranging between 0 and 1.

(i) **Image properties**, leave the default values.

(j) **Save > Image base name: Image_advanced**

(k) Change the animation folder name to `$Working-directory/asm0/img_advanced` for example.

(l) Press button **Create animation**



Advanced animation wizard

(m) Check the new images stored in folder `$Working-directory/asm0/img_advanced/`. The images sequence shows the velocity contours at the Cutting plane 1 at a given time (0.15 s). The cutting plane moves from $Z = 0$ to $Z = 1$, showing the velocity contours in the whole domain.

(n) Create a text file named "*position.txt*" in the same folder where you have saved the project file (.xpf) with the following data:

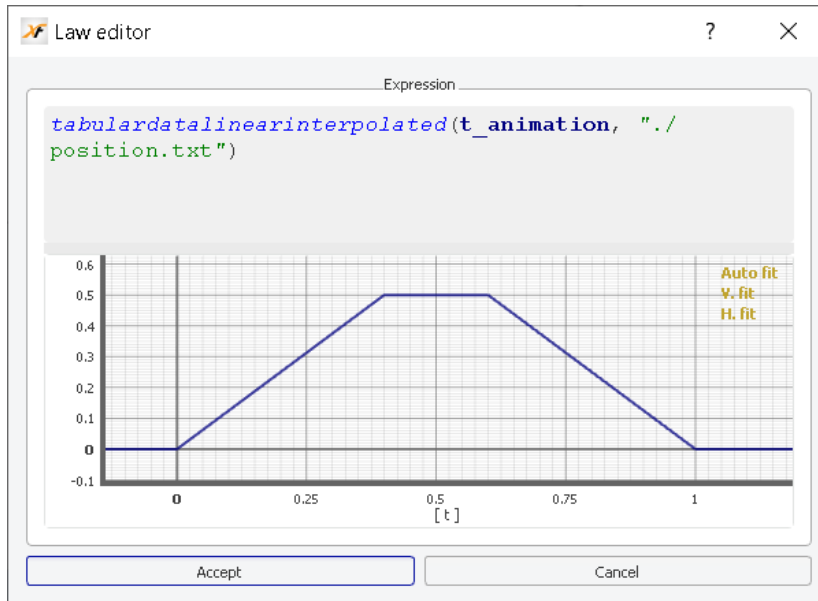
```
position.txt
0 0
0.4 0.5
0.6 0.5
1 0
```

First column indicates **t_animation**, which ranges from zero to duration, whereas the second column

Step 4: Post-processing

stands for the normalised position of the Z-plane ($0 = -Z$; $1 = +Z$).

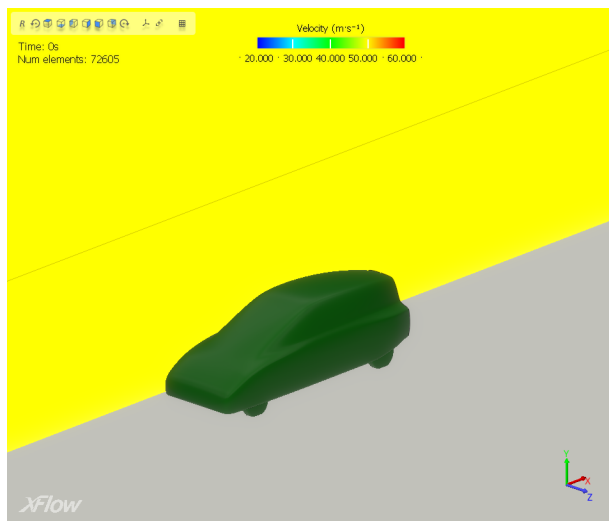
- (o) Go back to point (a) and create a new animation where: **Cutting plane 1 > Position law:** `tabulardatalinearinterpolated(t_animation, "./position.txt")` and click on  to check if the data is read correctly by XFlow:



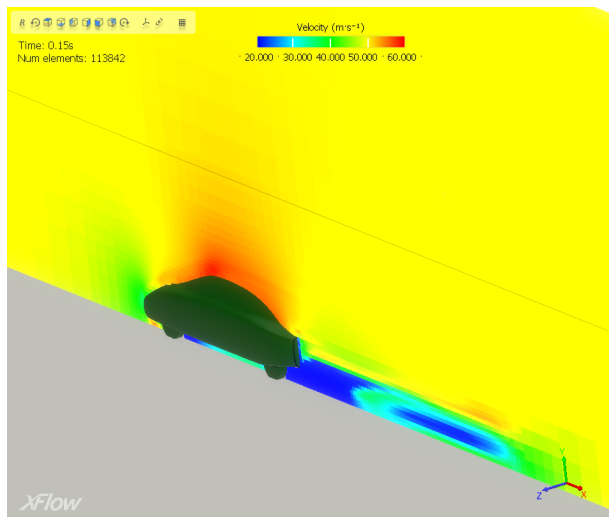
Note that the cutting plane now moves from $Z = 0$ to $Z = 0.5$, it stays still during 0.2 s (real time) and then goes back to $Z = 0$.

4.3.3 Keyframes animation

- (a) **Main menu > Post-Processing > Animation > Edit keyframe mode**. It is also possible to enter keyframe mode by long pressing  in the Data Processing toolbar or with **A/t + k** shortcut.
- (b) Select first frame ($t=0s$)
- (c) Following point 4.1 steps, create a **z-axis normal cutting plane at -0.2 m** location.
- (d) Move the camera to get a front perspective view of the geometry
- (e) **Main menu > Post-Processing > Animation > Create/Delete keyframe**. It is also possible to create a keyframe mode by using  in the Data Processing toolbar.
- (f) Select last frame ($t=0.15s$)
- (g) Following point 4.1 steps, create a **z-axis normal cutting plane at 0.1 m** location.
- (h) Move the camera to get a back perspective view of the geometry
- (i) **Main menu > Post-Processing > Animation > Create/Delete keyframe**. It is also possible to create a keyframe mode by using  in the Data Processing toolbar.



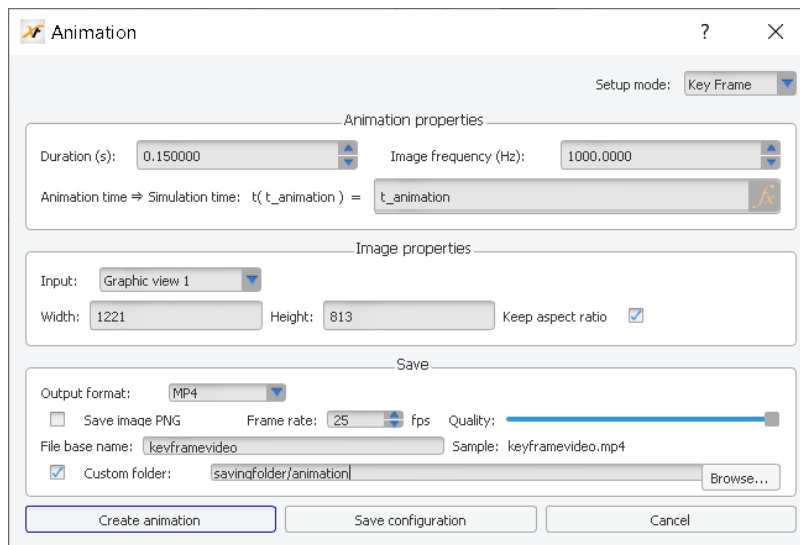
Front perspective view (t=0s)



Back perspective view (t=0.15s)

- (j) **Main menu > Post-Processing > Animation > Create animation**, or click on  in the Data Processing toolbar.

Step 4: Post-processing

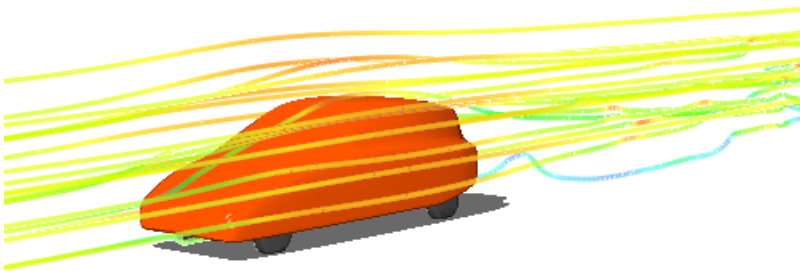


Keyframe animation wizard

- (k) **Duration: 0.15 s**, this represents the duration (real time) of the animation
- (l) **Image frequency: 1000 Hz**, frames per second. Since the duration is 0.15 seconds, the animation will consist of a sequence of 149 images.
- (m) **Animation time: t_animation**. This implies, the results will be interpolated in both space (created keyframes) and time.
- (n) **Image properties**, leave the default values.
- (p) **Save > File base name: keyframevideo** for example.
- (q) Change the animation folder name to `$Working-directory/asmo/animation` for example.
- (r) Press button **Create animation**

Tutorial 03 - Advanced post processing

This tutorial illustrates some of the advanced post processing capabilities of XFlow using the results computed in Tutorial 02. You will explore the powerful post-processing features of XFlow and learn more about the rendering in XFlow.



This tutorial shows how to:

- Change colour of the GUI
- Set colours and materials to the geometries
- Use the ray-tracing render mode
- Set lights and ground wall colour
- Set environment textures
- Hide the domain box
- Visualise the volumetric field
- Visualise stream tracers
- Use the Discrete Phase Model
- Create a camera
- Create additional *Graphic View* windows

It is assumed that the reader has completed Tutorial 01 and 02. Hence, some steps in the setup and post-process will not be described in detail.

Before starting the tutorial, please download the project data files from the XFlow[version]_TutorialsPack.zip folder.

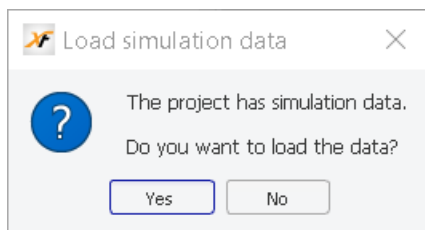
Contents

[Step 1: Advanced post-processing](#)

Advanced post-processing

1.1 Getting started

(a) Execute XFlow. In the *Project Manager* window click **Browse for existing project** and search for the path of the Tutorial-02 project file (e.g. asmo.xfp). XFlow will open and will ask the user whether to load the data, select **Yes**.



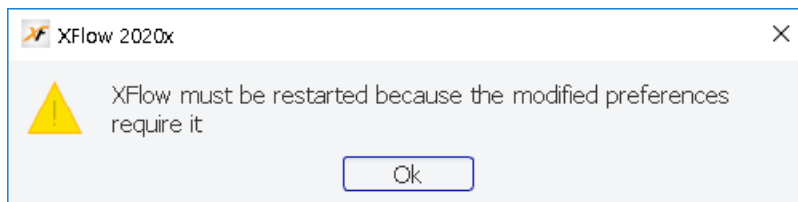
1.2 Change colour of the GUI

XFlow GUI colours can be customised in two styles:

- Dark: with dark colours
- 3DX: with light colours (the default configuration)

Keep the default style in: **Main menu > Options > Preferences, ⚙ > Application > Application style: 3DX**

In case of any change, the user will need to restart XFlow. This will be indicated by a warning like the one shown in the figure below:



1.3 Import geometry parts and set colours

Once a simulation has been performed, the user can import or create additional geometries for visualisation purposes. These geometries will be listed in the Post-Processing tab of the Project tree in

order to distinguish them from simulation geometries.

Here, the user is going to import a geometry of the ASMO where the wheels are separated from the main body:

- (a) Import car.nfb, wheel1.nfb, wheel2.nfb, wheel3.nfb, wheel4.nfb (**Main menu > Geometry > Import a new geometry** or use the toolbar icon ). In the pop-up window, leave the default options and press **OK**
- (b) Move every new **Shape** to position (-0.57, -0.63, 0) to make it to coincide with the original geometry
- (c) Hide the original geometry "asmo" by clicking on the icon .
- (d) Remove the original geometry "Line" by selecting the **Shape: Line** in the **Project Tree > Post-processing > Entities** and pressing key **Del**. To permanently remove this geometry, press **Ok** in the pop-up window.
- (e) Collapse the subfields of all Shapes in the **Project Tree > Post-Processing** by **right clicking** on the background and select the **Collapse all** option in the drop-down menu that appears
- (f) The user can assign different visualisation properties (e.g. colours) to the different parts of the asmo geometry. To do so, select a **Shape**, **right click** in the **Graphic View** and select **> Set visualisation material** in the **Graphic View** menu.
- (g) Click the coloured rectangle, select the desired colour from the palette and press **Apply**. Repeat the process to change the colour of every Shape (car and four wheels).

1.4 Set materials and real-time properties

In addition you can set a surface material to the shapes. Each material is displayed depending on its lighting properties.

- (a) Set **Plastic** material to the wheels (hold the **Ctrl** button to select all the wheels geometries) and **Metal** to the car. Select a **Shape**, **right click** in the **Graphic View** and select **> Set visualisation material** in the **Graphic View** menu.
- (b) Try different materials and different values of the parameters (between 0 and 1) such as the **Opacity** index in **Glass** or the **Metallicity** index in **Metal**.

Observe that the metal and glass materials reflect the colour of the ground wall.

It is possible to change the real-time visualisation properties in: **Main menu > Options > Preferences**,  **> Graphic View > Performance-Quality**. Lowering those parameters can speed up the visualisation of the **Graphic View**, but high parameter values improve the quality of the rendering. If the graphics card of the used hardware has limited resources, it is recommended to decrease these parameters lowering in such a way the graphical resources required by XFlow.

1.5 Ground wall colour and Show wires

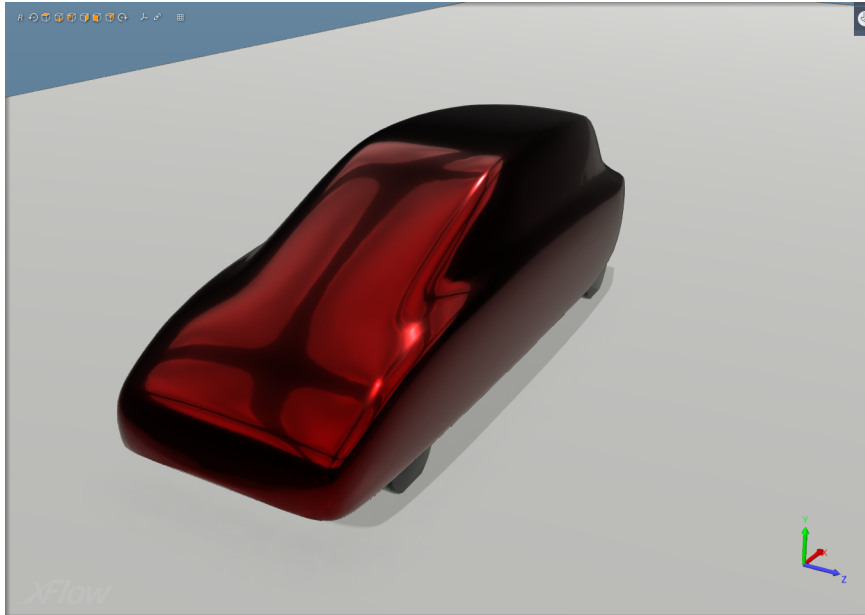
Change the colour of the ground wall: **Main menu > Options > Preferences**,  **> Graphic View >**

Environment > Ground wall colour

Click on the coloured rectangle and select for instance the gray colour.

Disable now the visualisation of the geometries wires: **Main menu > Options > Preferences,  >**

Geometry > Show wires:

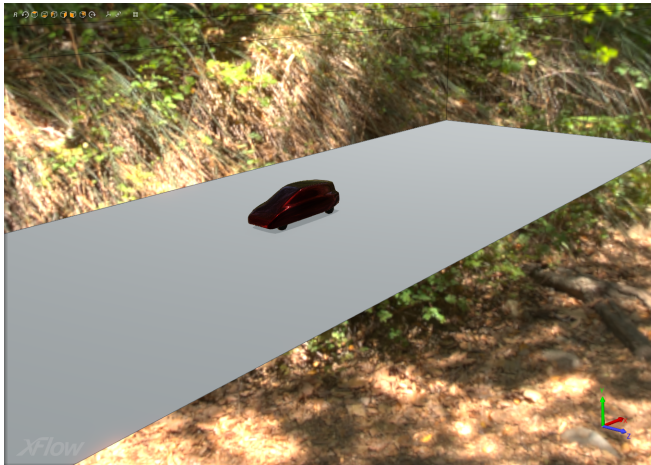


Ground wall shown in gray and geometries shown without wires

1.6 Set environment textures

In Tutorial 01 the user has already changed the colour of the background (see step 4). It is also possible to have a background as a two-colour gradient or even have a texture as environment map.

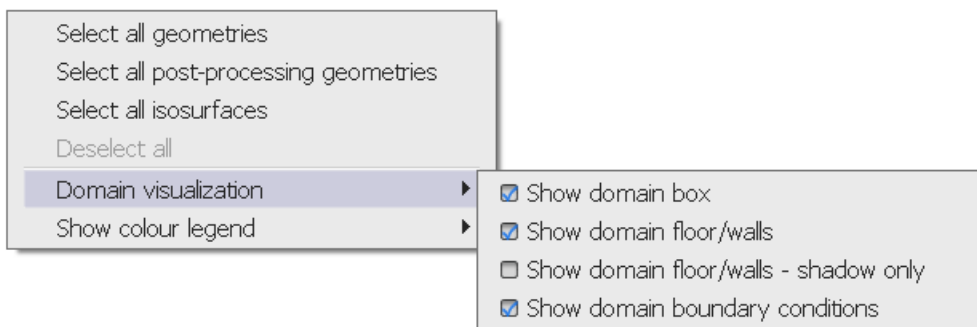
- (a) **Main menu > Options > Preferences,  > Graphic View > Environment: Background** select **Two colours** and choose them by clicking on the coloured rectangles
- (b) Now, **Main menu > Options > Preferences,  > Graphic View > Environment: Background** select **Texture**, and browse for a texture image (e.g Topanga_Forest_B_3k.hdr)
- (c) Set the view to as perspective and rotate the camera around to visualise the background texture



1.7 Hide the domain box and the ground

To hide the domain box:

- Deselect any geometry shape in the *Project Tree*
- Right click** in an empty area of the *Graphic View* window
- In the pop-up menu, deactivate **Show domain box**.



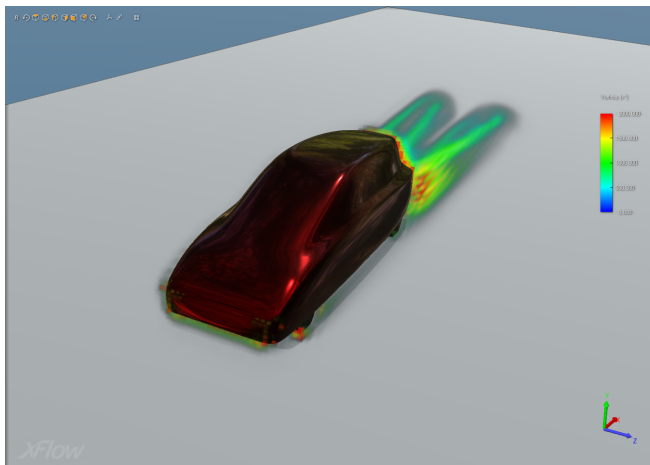
This option will hide the wireframe of the domain box as well as the ground wall.

1.8 Visualise the volumetric field

Volumetric field shows the chosen visualisation field in the entire domain with an opacity given by a specified transference law. This law is a function of a (alpha value). $a=0$ corresponds to the legend minimum and $a=1$ to the maximum.

- If the Simulation data are not already loaded, browse for the `asmo` simulation folder of the Tutorial 02 in **Project Tree > Simulation > Store data > Folder**, save the project file and click on  to load the simulation data.
- In **Post Processing > General > Show > Volumetric field** select the vorticity as visualisation field

- (c) In **Main menu > Simulation data > Analysis settings** or using the toolbar icon  prescribe the vorticity legend range to [0, 2000] by disabling the automatic range and inputting the minimum and maximum values.
- (d) Switch off the interpolation **Post-Processing > General > Interpolation mode: Off**. Otherwise the render of the volumetric field will be too slow.
- (e) Show the volumetric field by clicking on the **Volumetric field** icon . By default, the transference law is set to $a \cdot a$, which assigns a higher opacity to the higher vorticity values. Try different transference laws, e.g. a and a^3 .



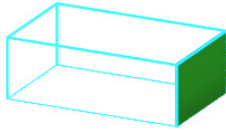
Vorticity volumetric field visualisation with transference law $[a \cdot a]$

1.9 Visualise stream tracers

Stream tracers refer to streamlines of the velocity vector field generated from a collection of source points. XFlow distinguishes between two cases: (i) when the tracer is massless and therefore has a passive behaviour, and (ii) when the tracer is a particle with mass, drag and inertia (Discrete Phase Model, DPM). In this tutorial we are going to calculate a **steady passive tracer**.

To visualise stream tracers a geometry shape is required that will be the source of stream tracers. Instead of a line like in [tutorial 02](#), a surface will be created.

- (a) first create a box (**Main menu > Geometry > Create object > Create box** or just clicking on  in **Object Creation** toolbar) with lower corner (-3,-0.75,-0.3) and upper corner (-2,-0.4,0.3). Then eliminate all the box surfaces except the one at $x = -2$ by selecting the face filter  in the toolbar, select each surface indicated in red wireframe in the image below and remove them by pressing **Del**

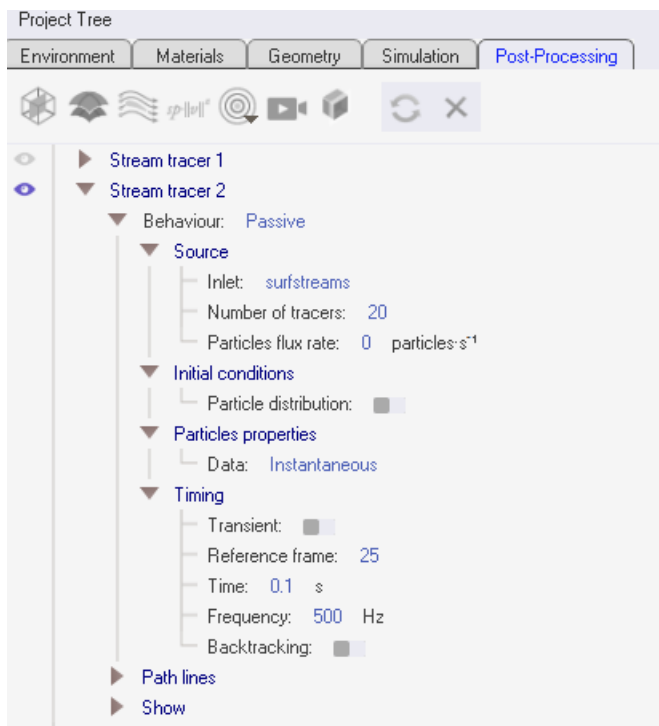


Face (not to be deleted) highlighted in green

In this way only the surface remaining is the one pointing to the vehicle. In **Project Tree > Post-Processing > Entities**, rename (right click > Rename) this **Shape** as "surfstreams" and set its behaviour as **Disabled**.

(b) Go to **Project Tree > Post-Processing > Stream tracers**. If **Tracer 1** already exists enable it, otherwise create it by **right clicking** on **Stream tracers** and select **Add stream tracer**. This creates **Stream tracer 2**.

(c) Define **Stream tracer 2** according to the following setup: passive steady tracer consisting of 20 tracers released from a random but constant position at the shape "surfstreams" and calculated from the results at frame 25, as indicated in the figure below



(d) Select  next to **Stream tracer 2**

(e) To visualise the path lines, set the **To frame:** value to 50 **right click** again on **Stream tracer 2** and select **Generate path lines** or hit  next to **Path lines**

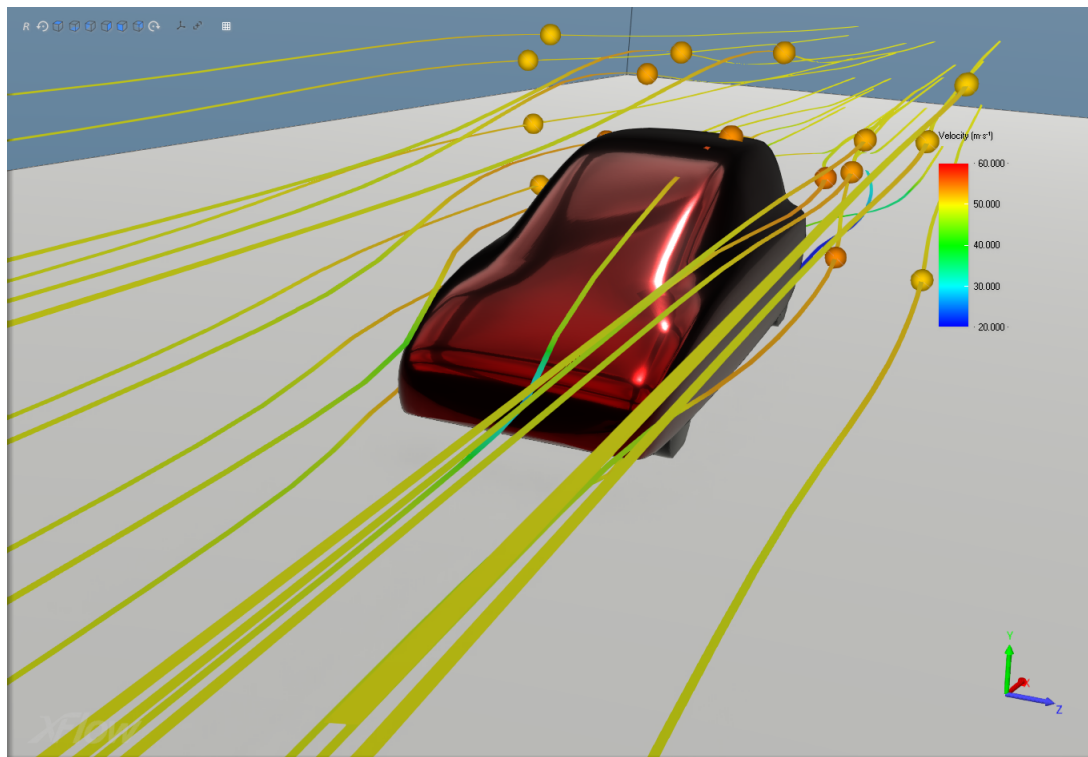
(f) Choose **Stream tracer 2 > Show > Path line: Ribbon** and **Size factor: 0.1**.

(g) Choose **Stream tracer 2 > Show > Current marker: Sphere** and **Size factor: 0.3**.

(h) By default the tracers are coloured by the particle Id. Set the **Coloured by:** field to **Field** and select **Velocity**

Set the velocity legend range to [20, 60] and recompute the path lines (change the **To frame:** 49 and back to 50) hit  to take effect.

Change **Path lines** style to Line, Spheres chain and Tube. You may need to adjust the size factor to visualise these path lines correctly.



Stream tracers with path line representation

(g) Hide the **Stream tracer 2** by clicking on the icon . The timeline will now be unlocked.

1.10 Create a camera

When you have played the solution in the previous step, the camera is fixed and defined by the parameters in **Post-Processing > Views > Graphic View 1 > Camera settings > Location:**

- **From** describes the position of the camera or eye point.
- **To** describes the position of the target, where the camera is looking at. It is the center of the view.
- **Up** describes the direction of the vertical axis.

Notice how these parameters change when you navigate in the *Graphic View* (pan, zoom, rotation).

You can create your own cameras in **Post-Processing > Cameras**. Right click on **Cameras** string to add a new camera or click on  in the *Post-Processing* toolbar.

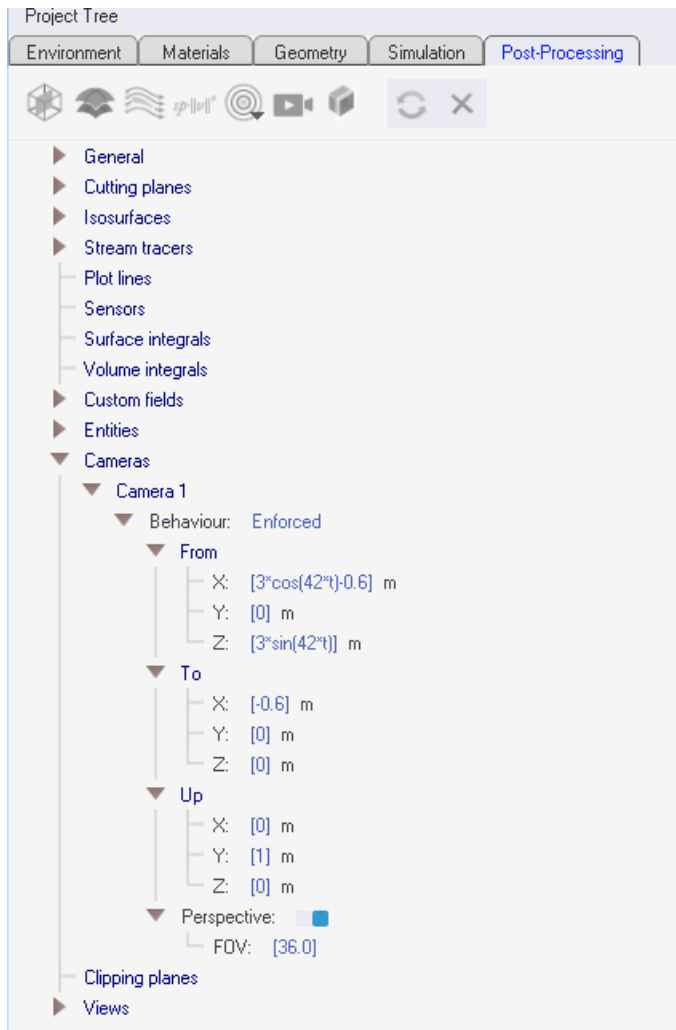


For example, define a camera from the top by defining **From** = (0, 10, 0), **To** = (0, 0, 0), **Up** = (1, 0, 0), behaviour **Fixed** and perspective **On**. To activate this camera, go to **Post-Processing > Views > Graphic View 1 > Camera settings > Link to camera** and select **Camera 1**.

To return to the interactive camera, select **Link to camera: Graphic View 1**.

It is also possible to define a moving camera. Select **Enforced** behavior. Now the entries for all the parameters allow the use of functions (it is indicated by the brackets).

(a) Enter the following expressions to describe the camera motion:

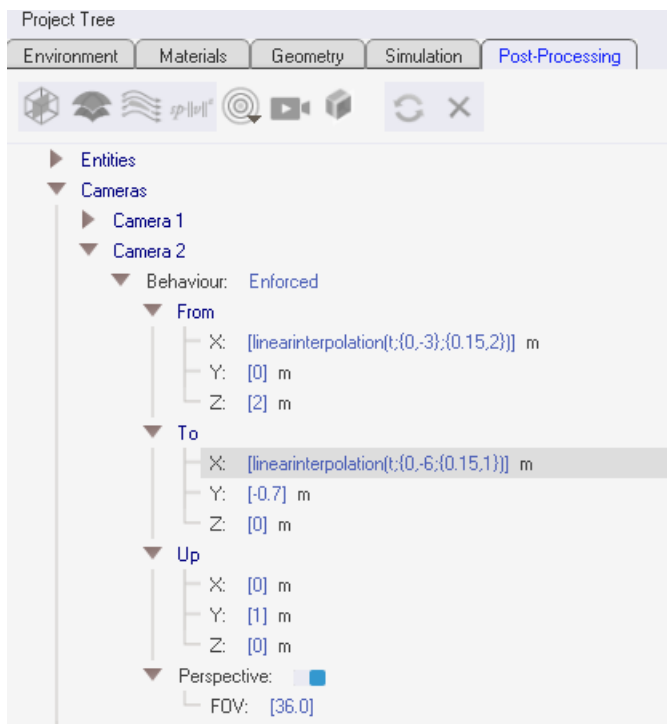


- (b) Display the graph of the From X-coordinate in the *Function Viewer* (you may need to create one) by selecting the icon . Check that it is periodic with a period of 0.15 s.
- (c) Link the Graphic View to Camera 1: **Post-processing > Views > Graphic View 1 > Camera settings > Link to camera: Camera 1.**
- (d) Play forward the results. The camera follows a loop around the vehicle.

1.11 Create additional Graphic View windows

Each *Graphic View* can display different fields and can have different cameras.

- (a) Create a new *Graphic View* window by hitting . This creates a new floating window corresponding to *Graphic View 2*. This new window can be located anywhere on the GUI by snapping it.
- (b) Create a second camera with the following parameters:

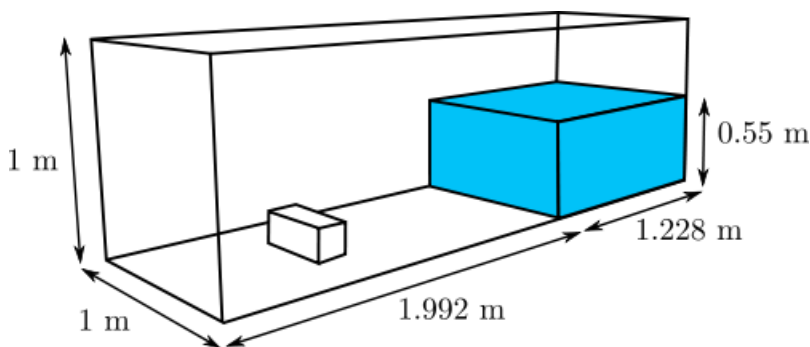


- (c) Link the Graphic View 2 to Camera 2. Select Show volumetric field as indicated in step 8 and play forward the results.

Tutorial 04 - Dam break

XFlow can also be applied to problems that involve a free surface between a liquid and a gas. Examples of this type of problem include simulating the waves produced by a ship or the forces exerted by waves that break against an oil platform.

This tutorial illustrates the setup and solution process of a dam break flow. The problem consists of a rectangular tank with an obstacle. In the right part of the tank there is a water column that will flow through the tank due to the effect of gravity and impact against the obstacle.



This tutorial shows how to:

- Set a free surface problem
- Define liquid regions
- Visualise the fluid particles
- Create arbitrary cutting planes
- Create clipping planes

It is assumed that the reader has completed Tutorials 01 and 02. Some steps in the setup and post-process will not be described in detail.

Before starting the tutorial, please download the project data files from the XFlow[version]_TutorialsPack.zip folder.

Contents

[Step 1: Problem setup](#)

Step 1: Problem setup

1.1 Engine settings

In **Project Tree > Environment > Engine** set the following parameters:

- (a) **Kernel**: **3d**
- (b) **Flow model**: **Free surface**
- (c) **Analysis type**: **Internal**. Now the external domain box will disappear from the Graphic View and you will have to provide the geometry of the tank that will contain the fluid
- (d) **Turbulence settings > Turbulent model**: **Automatic**

1.2 Environment settings

In **Project Tree > Environment > Environment** set the following parameters:

- (a) **Global attributes > Ext. acceleration laws**: **(0, -9.81, 0) m.s⁻²**, to take into account the gravity.
- (b) Leave the **Initial conditions** as **User defined**, and keep the **Initial velocity field** as **(0, 0, 0) m.s⁻¹**
- (c) **Liquid regions > Initial liquid function**: **if (x<0,y<0.55)**, to define the initial water column. This means that if $x < 0$ then the liquid region has height 0 m, else ($x > 0$) the liquid region has height 0.55 m.

 **Tip:** The Initial liquid function can also be written $(x<0)*(y<0.55)$, since multiplication operator acts as a AND Boolean operator (OR Boolean operator is the addition operator). It is sometimes simpler to compose several conditions.

1.3 Material settings

In **Project Tree > Materials** leave the default **Material 1** properties. In free-surface simulations, the default fluid is water.

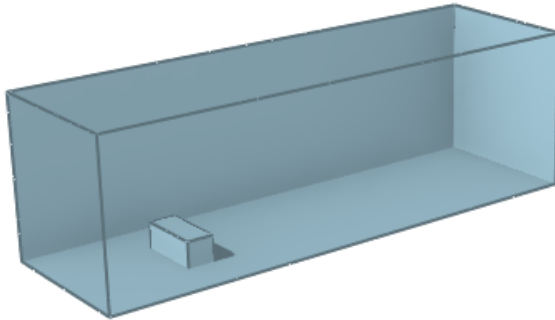
1.4 Import the geometry

Import the geometry of the tank from the file `tank.nfb`: **Main menu > Geometry > Import a new geometry** or press  in toolbar Geometry. In the Import Geometry pop-up window, leave the the default

options and press **OK**. The geometry is then displayed in the *Graphic View 1*.

To observe that the coordinate origin is at the location where the water column begins, show the grid

selecting the button  or **Main menu > Options > Preferences > Graphic View: Show grid**



Correct tank orientation (Otherwise, reverse normals)

1.5 Geometry settings

In **Project Tree > Geometry > Entities > Shape: tank** set the following parameters:

- (a) **Behaviour: Fixed**
- (b) **Boundary conditions: Wall**, with **Wall model: Free-slip**

1.6 Simulation settings

In **Project Tree > Simulation** set the following parameters:

- (a) **Time > Simulation time: 3 s**
- (b) **Time > Time step mode: Fixed automatic**, with **Courant: 1**
- (c) **Lattice discretization > Resolved scale: 0.03 m**, with the **Refinement algorithm: Disabled**
- (d) **Store data > Frames frequency: 50 frames per second (Hz)**
- (e) Enable **Compute markers**

1.7 Launch the calculation

- (a) Save the project
- (b) Set the number of CPUs in **Main menu > Options > Preferences > Engine**
- (c) Press **Run** button > **Start computation**

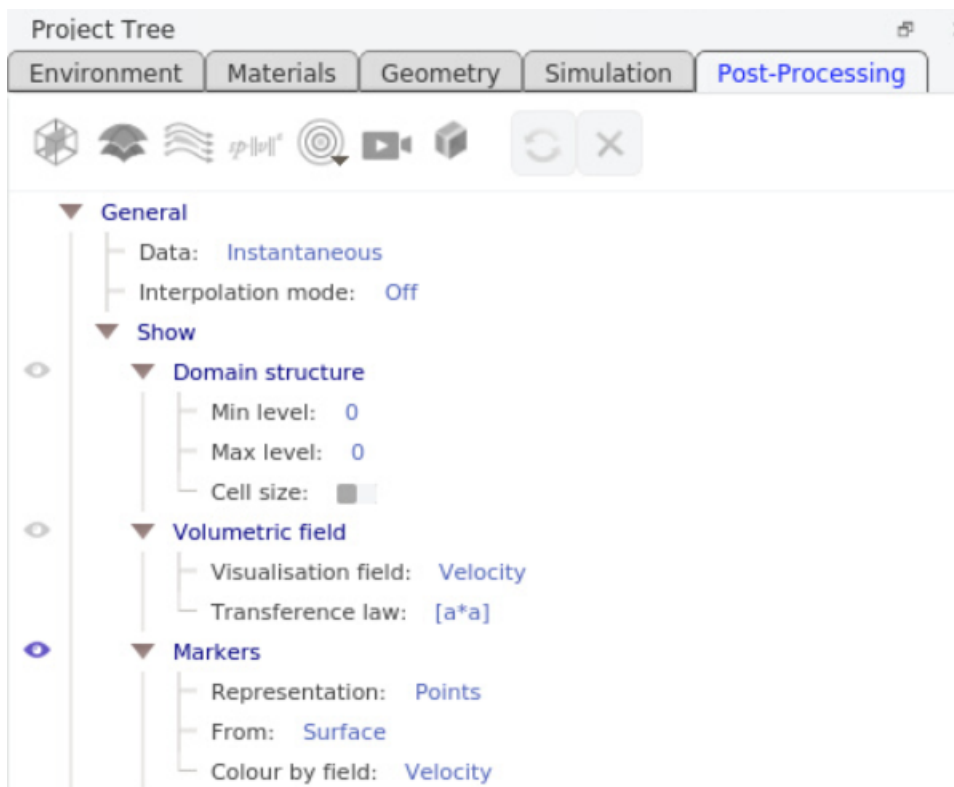
Step 2: Post-processing

2.1 Load data

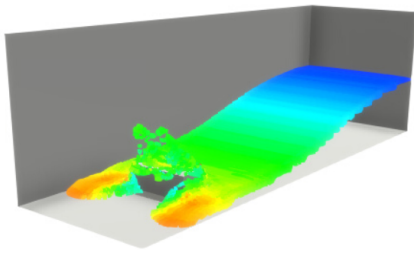
Load data: **Simulation data > Load simulation data** or 

2.2 Show particles

Tick the visibility icon  at **Post processing > General > Show > Markers**



- (a) To show only the particles at the free surface, select **Markers > From: Surface**; otherwise, select **All** to show the whole domain of fluid
- (b) To represent the particles with different shapes, go to the last frame, and select **Arrows** and **Sphere** in **Markers > Representation**.
- (c) Play the simulation by pressing 



2.3 Create a clipping plane

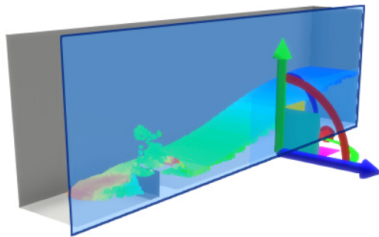
Post-Processing > Views > Graphic View 1 > Camera settings > Clipping planes

(a) **Right click** on **Clipping planes** and choose **Add clipping plane**. **Plane 1**, defined by a **Origin** point (0, 0, 0) and a **Normal** vector (1, 0, 0), will then be created.

(b) Click the visibility icon  in front of **Clipping plane 1** to activate the clipping plane and change the **Normal** vector to: **(0, 0, -1)**. Zoom-in to see the details of the flow close to the obstacle.

(c) Click on **Clipping plane 1** to select the cutting plane and note the axis plot in the Graphic View.

Also note the selection mode has been changed to **Object filter** . Click on the blue axis (Z-axis) in order to move the clipping plane along the Z-axis.



(d) Switch back to **View only** filter by clicking the icon 

(e) Disable the clipping plane clicking again on the visibility icon 

2.4 Create data sensors

Post-Processing > Sensors

(a) To create a sensor, **right click** on **Sensors** and choose **Add sensor**. Create two sensors, one on the front of the obstacle at position **(-1.1675, 0.110, 0) m** and other on the top of the obstacle at position **(-1.262, 0.165, 0) m**

(b) Select: **Field: Static pressure** for both sensors

(c) Refresh the sensors with 

 **Tip:** You can select one sensor, press the **Shift key** and click on the second sensor in order to select both of them. You can then click on the refresh icon at the top of the Post-Processing tab in order to

Step 2: Post-processing

refresh the two sensors at the same time.

It is also recommended to always disable all post-processing items before refreshing sensors, to avoid loading them every frame refreshed.

(d) Display the *Function Viewer* window and measure the pressure in both sensors: **right click** on *Function Viewer 1* to show the Function Viewer menu, then choose **Sensors > Sensor 1** and **Sensors > Sensor 2**

(e) Save the measurements of both sensors to a text file. To do so, **right click** on *Function Viewer 1* to show the Function Viewer menu, then **Sensors > Export all**

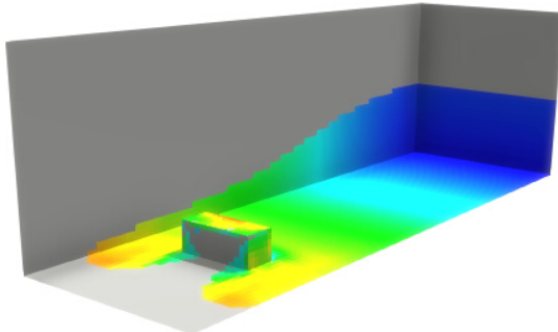
⚠ Please note: To avoid to have to refresh the sensors, a smart alternative is to use the Probes instead, **Simulation > Store data > Probes**. However, the Probes need to be defined before running your computation.

2.5 Show surface info

Post-Processing > General > Show > Surface info

(a) Select **3d field** and a **Field: Velocity**

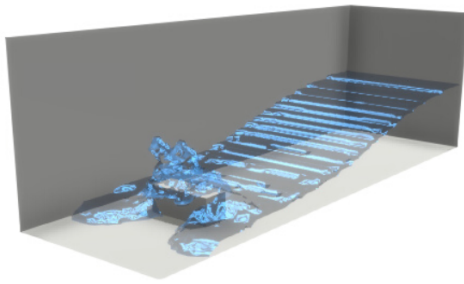
(b) Click the visibility icon 



2.6 Create an isosurface

(a) Create an isosurface **Right clicking** on **Post-Processing > Isosurfaces** and **Add isosurface** with **Field: VOF** and **Value: 0.1**. This way, the isosurface will represent the water free surface.

(b) Change the visualisation material of the Isosurface to get visual effect similar to water: Select the Isosurface in the **Graphic View 1 > Right click > Set visualisation material**. Select the **Glass** material type, blue colour and **Opacity** to **0.30**:



Isosurface of Volume of liquid phase

Tutorial 05 - Breaking waves

This second tutorial on free surface flows consists of traveling water waves approaching a beach.

The simulation consists of a channel flow where waves will be generated following the linear wave theory. The beach will be modeled first as impermeable and later as a porous medium. A buoy will be included to track the water movement.



This tutorial shows how to:

- Set an external free surface problem
- Set progressive waves with the help of the wizard
- Set center of gravity for moving objects
- Set the gravity as an external acceleration $1a$ or as an external body force derived from a potential function
- Set porous regions

It is assumed that the reader has completed Tutorials 01, 02 and 04. Some steps in the setup and post-process will not be described in detail.

Before starting the tutorial, please download the project data files from the XFlow[version]_TutorialsPack.zip folder.

Contents

[Step 1: Problem setup](#)

[Step 2: Post-processing](#)

[Step 3: Gravitational potential](#)

[Step 4: Porous volume](#)

Step 1: Problem setup

1.1 Engine settings

In **Project Tree > Environment > Engine** set the following parameters:

- (a) **Kernel:** 2d
- (b) **Flow model:** Free surface
- (c) **Analysis type:** External. The water channel domain appears.
- (d) **Turbulence settings > Turbulent model:** Automatic

1.2 Import the geometry

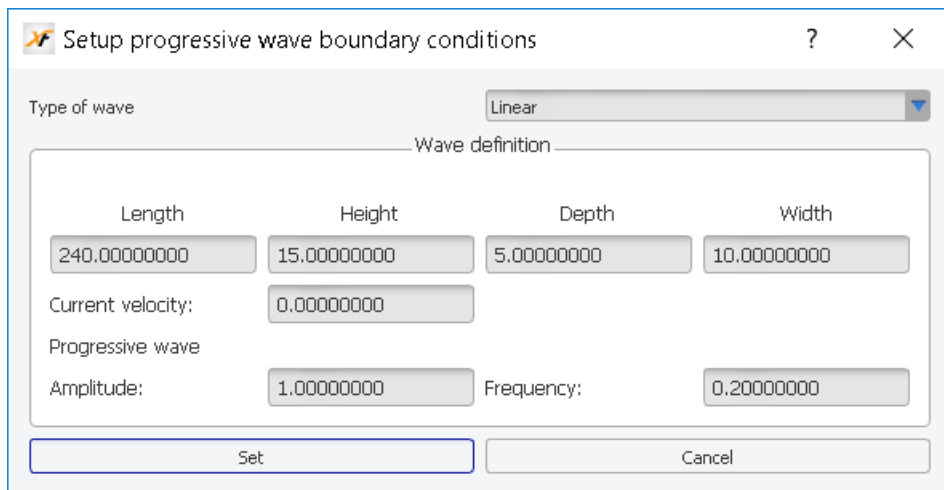
- (a) Import the geometry of the beach from the file `beach.nfb`: **Main menu > Geometry > Import a new geometry** or . Leave the the default options and press **OK**
- (b) Check the correct orientation of the wedge surface, i.e. make sure the normals are pointing to the outside. Otherwise, reverse orientation (Select the geometry, **right click** on the *Graphic View* window to show the **Graphic View Menu** and choose **Reverse orientation**).

1.3 Environment settings

In **Project Tree > Environment > Environment** set the following parameters:

- (a) Leave the **Gravitational potential** Off and introduce the external acceleration law **(0, -10) m·s⁻²** to include the gravity
- (b) **Initial conditions:** Water channel default
- (c) With the help of the wizard (**Main menu > Options > Setup progressive wave boundary conditions**), define the velocity laws, the water initial surface and the inlet wave function according to a linear wave given by the following parameters:
 - dimensions: **Length = 240 m**, **Height = 15 m** and **Width = 10 m**
 - settings of the waves: **Type = Linear**, **Depth = 5 m**, **Current velocity = 0 m·s⁻¹**, **Amplitude = 1 m** and **Frequency = 0.2 Hz**.

Step 1: Problem setup



Wizard for the setup of progressive wave boundary conditions

Notice that the position of the channel changes automatically to **(0, 2.5, 0) m** so that the mean free surface level is located at $y = 0$ m.

- (d) Set the **Channel Wall types > Automatic** with zero velocity in X-direction

1.4 Material settings

In **Project Tree > Materials** leave the default **Material 1** properties, since these are the water properties.

1.5 Geometry settings

In **Project Tree > Geometry > Entities > Shape: beach** set the following parameters:

- (a) **Behaviour: Fixed**, located at **Position: (-3, -8, -6)**
- (b) **Boundary conditions: Wall** (solid impermeable beach), with **Wall model: Automatic** and **Roughness: 0.5 m**

1.6 Simulation settings

In **Project Tree > Simulation:**

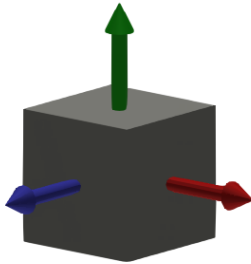
- (a) Set the **Time > Simulation time: 20 s** and the **Time > Time step mode** to **Fixed automatic** with **Courant: 1**
- (b) Set the **Lattice discretization > Resolved scale: 0.2 m**, with the **Refinement algorithm: Disabled**
- (c) **Store data** at a **Frame frequency** of **5 Hz** and leave the numerical data frequency by default
- (d) Leave off **Save averaged fields**
- (e) Enable **Compute markers**

1.7 Create a cubic buoy

- (a) Create a box (**Main Menu > Geometry > Create Object > Create box**) with lower corner (-1, -1, -1) and upper corner (1, 1, 1)
- (c) Move the box to position (-68, 0, 0) m: **Geometry > Entities > Shape: Box > Behaviour > Position**

⚠ Please note: You could also create the cube directly in the right position, i.e. lower corner (-69, -1, -1) and upper corner (-67, 1, 1); but observe that then the center of gravity would be in the global origin (0, 0, 0), which would make the simulation to fail. In this case you should move the center CoG to the cube center: Select the geometry > **Right click** in *Graphic View* > **Modify CoG/CoR position:** X = -68, Y = 0, Z = 0

- (b) To check that the center of gravity is located in the center, select the box and show its local axes: **right click** on the *Graphic View* window to show the **Graphic View Menu > Show > Local axes**. The origin of the local axes is the center of gravity and the center of rotation of the object.



Local axes

- (d) Hide the local axes: Select the geometry and **right click** on the *Graphic View* window and choose **> Show > Local axes**
- (e) Set the **Geometry > Shape: Box > Behaviour** to **Rigid body dynamics**, with a **Density** of **250 kg·m⁻³** (i.e. mass = 2000 kg) and leave free the three degrees-of-freedom: displacements in X and Y and rotation around Z in 2D. Thus set both **Constraints > Translation** and **Constraints > Rotation** to Free, with **External force global Y: -20000 N** (gravity*mass). Keep the default wall boundary conditions.

Step 1: Problem setup



Buoy properties

1.8 Launch the calculation

(a) Save the project

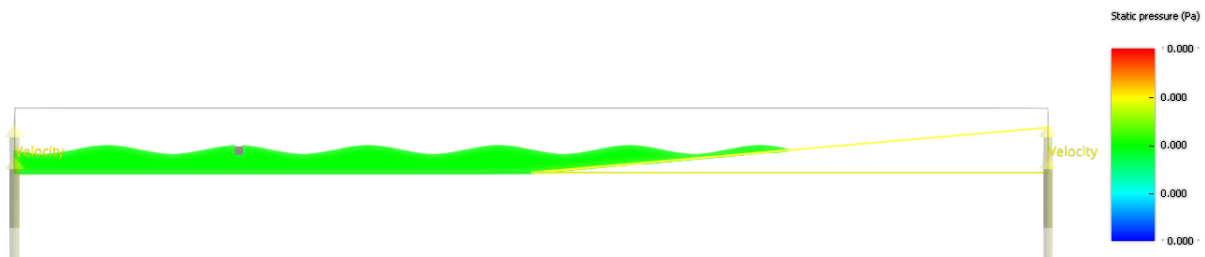
 **Tip:** You can directly load the setup of this problem from the project file `BreakingWaves.xfp`

- (b) Set the number of CPUs in **Main menu > Options > Preferences > Engine**
- (c) Press **Run** button > **Start computation**

Step 2: Post-processing

2.1 Check initial pressure field

- (a) Select frame zero, time zero.
- (b) Activate the markers visualisation in **Post-Processing > General > Show > Markers**.
- (c) Set **Markers > From**: All, to see all the markers in the domain.
- (d) Select **Markers > Color by field**: Static pressure
- (e) Select the beach and visualise it in wireframe mode (**Graphic View Menu > Visualisation Mode > Wireframe**). Observe that there is no fluid inside

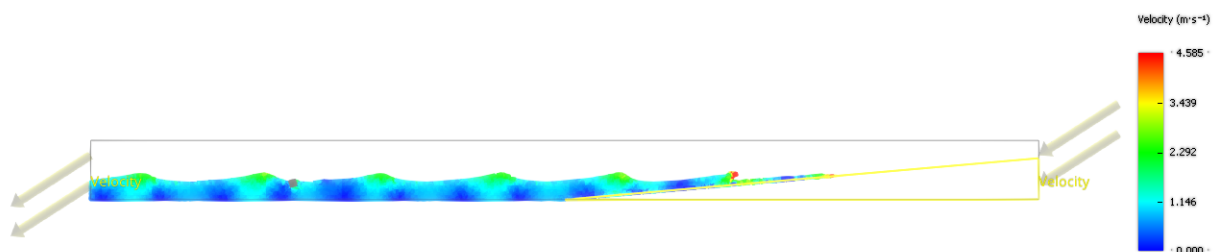


Markers coloured by the initial static pressure field (time = 0).

⚠ Please note: the **Initial condition: Water channel default** sets to zero the initial static pressure when **Gravitational potential** is **Off**.

2.2 Show particles

- (a) Show all particles from the fluid region by ticking the checkbox  in **Post-Processing > Show > Markers**. By default particles are represented by points and are colored by the velocity field.



Breaking waves markers

- (b) Play forward the results. Observe how the waves accelerate when they arrive at the beach and end up breaking due to shoaling.

Step 2: Post-processing

- (c) Visualise vorticity and static pressure fields (**Post-Processing > Show > Markers > Coloured by field**).

2.3 Track the movement of the buoy

- (a) Select the box and play forward the results. Initial position is marked in wireframe.
- (b) Create a *Function Viewer* window and display the evolution of the box position and rotation:
- **Right click** on the *Function Viewer* window, a drop-down menu appears then select **Shapes > Box > Px**
 - **Right click** on the *Function Viewer* window, a drop-down menu appears then select **Shapes > Box > Py**
 - **Right click** on the *Function Viewer* window, a drop-down menu appears then select **Shapes > Box > Euz**

Step 3: Gravitational potential

For *Free surface flows*, the feature **Gravitational potential** allows the user to easily consider external body forces derived from a potential function. This is the case of gravity.

In this section, the feature **Gravitational potential** is used to compute the [Breaking waves](#) case.

3.1 Gravitational potential setup

Load the project generated in [Step 1](#),  `BreakingWaves.xfp`, and save it as a new project with another name, , for example: `BreakingWaves_GravitationalPotential.xfp`.

Modify the *Project Tree* setup as follows:

- (a) Switch **On** the **Gravitational potential** in **Environment > Environment > Global attributes**.

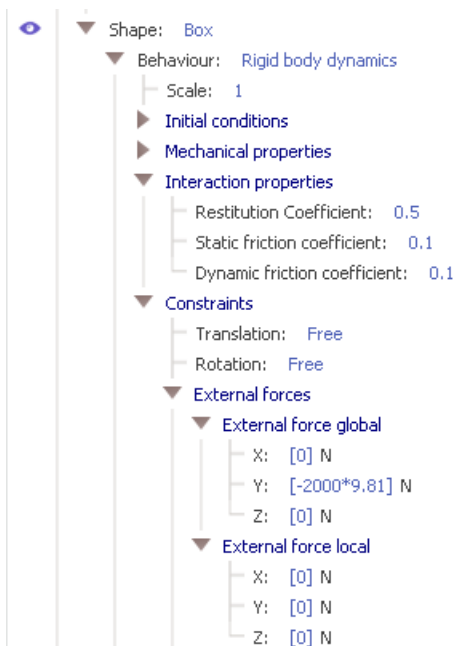
 **Please note:** the user must check the **External acceleration laws**. By default it is set the previous one **(0, -10)** and the user have to change it to **(0, 0)**.

 **Please note:** the **Gravitational potential: On** initializes the static pressure according to the hydrostatic pressure.

- (b) Leave the earth gravity as the **Fluid gravity: (0, -9.81, 0) m·s⁻²**
- (c) Leave the **Gravitational potential > Potential origin: Automatic**, which by default sets the potential origin at the free surface.



(d) Change the External force global applied to the Box geometry since this is affected by the gravitational potential by clicking In **Project Tree > Geometry > Entities > Shape: Box**.



3.2 Launch the calculation

(a) Save changes, 

 **Tip:** You can directly load the setup of this problem from the project file

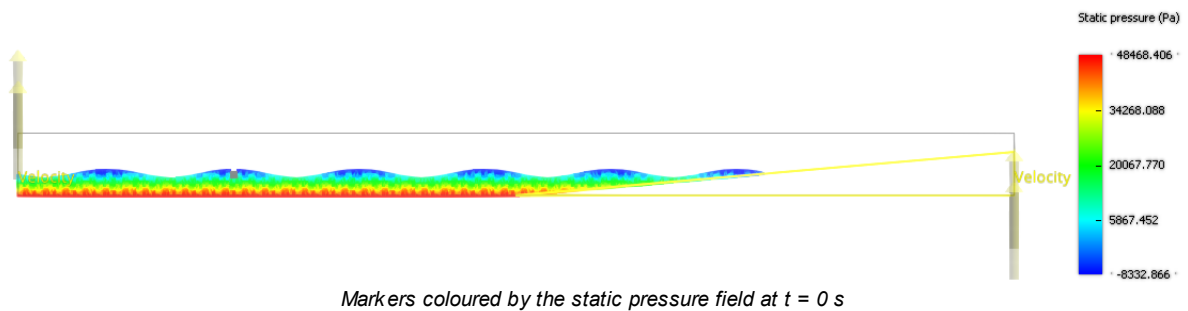
Step 3: Gravitational potential

BreakingWaves_GravitationalPotential.xfp

(b) Press **Run** button > **Start computation**

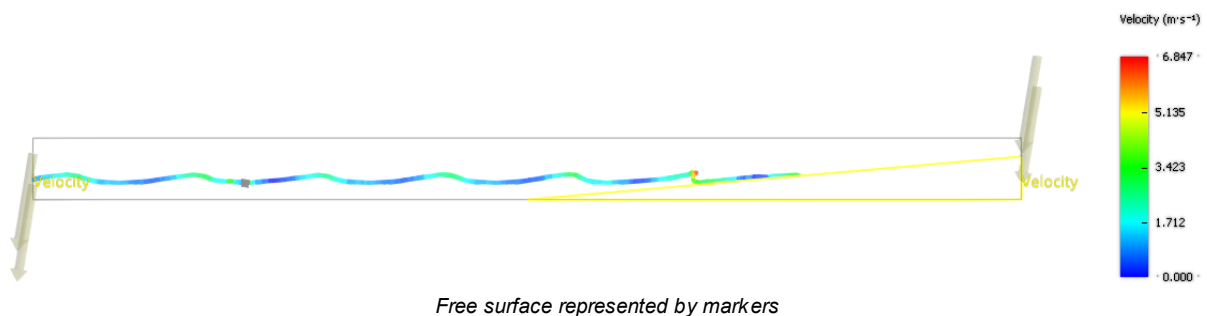
3.3 Post-processing

- (a) Visualize the initial static pressure field (frame 0). Activate the markers visualization in **Post-processing > General > Show > Markers** by ticking the box . Set **Markers > From: All**, to see all the markers in the domain, and select the **Static pressure** in **Markers > Color by field**.



 **Please note:** the **Gravitational potential: On** initialises the static pressure according to the hydrostatic pressure.

- (b) Visualise the free surface by activating the markers visualisation as follows:
- Enable **Post-processing > General > Show > Markers** by ticking the box .
 - Set **Markers > From: Surface**
 - Set **Markers > Color by field: Velocity**
 - Press Play 
 - Disable **Post-processing > General > Show > Markers**.

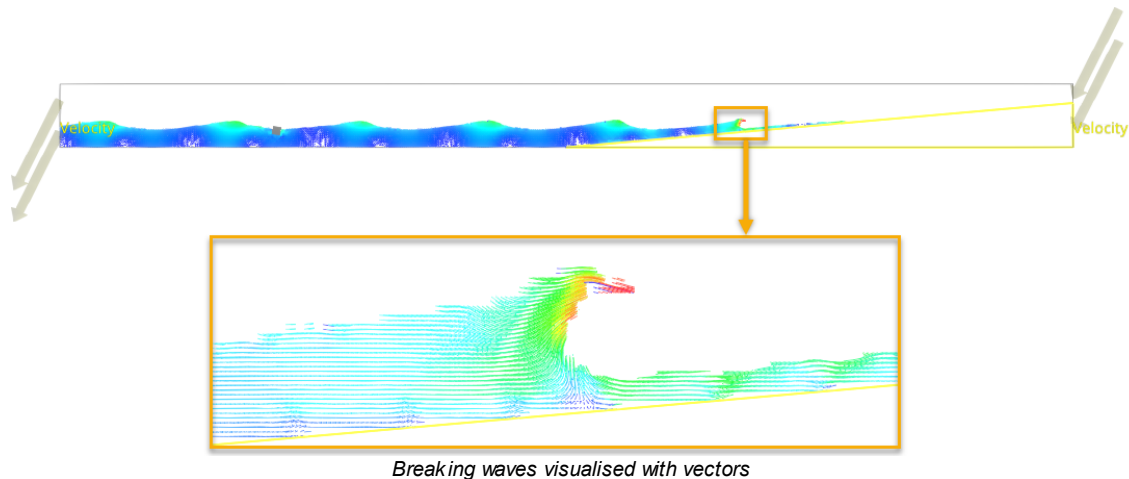


- (c) Visualise the flow pattern in the breaking waves:
- **Right click** on **Post-processing > Cutting planes** and select **Add cutting plane** to create the **Cutting plane 1**.
 - Set **Cutting plane 1 > Axis: Z** (Position = 0, as it is a 2d case)

- Cutting plane 1 > Visualisation mode: **3d field**
- Cutting plane 1 > Visualisation mode > Field: **Velocity**

Turn the cutting plane of 3d field to Vectors:

- Cutting plane 1 > Visualisation mode: **Vectors**
- Cutting plane 1 > Visualisation mode > Uniform distribution: **Off**
- Cutting plane 1 > Visualisation mode > Arrows density: **0.8**
- Cutting plane 1 > Visualisation mode > Arrow length: **1**
- Cutting plane 1 > Visualisation mode > Field: **Velocity**



Step 4: Porous volume

The beach will be now modelled as a porous volume, with a permeability of 10^{-6} m^2 .

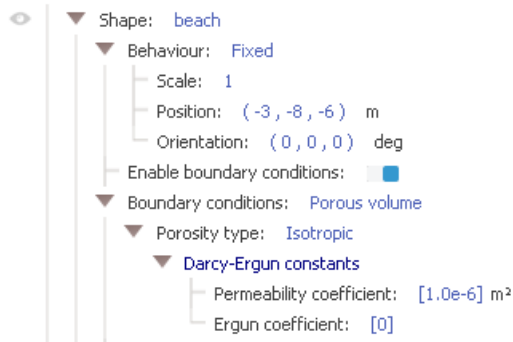
4.1 Porous region setup

Load the project generated in [Step 3](#),  BreakingWaves_GravitationalPotential.xfp, and save it as a new project with another name, , for example: BreakingWaves_Porous.xfp.

To define the beach as a porous volume please proceed as follows:

- Set the boundary conditions of the beach, **Geometry > Entities > Shape: beach > Boundary conditions > Others > Porous volume**, with Porosity type: **Isotropic**.
- Specify a **Permeability coefficient** of **1e-6** and **0 Ergun coefficient**

Step 4: Porous volume



- (c) Modify the initial water surface to leave void part of the beach interior, in **Environment > Water Channel > Water initial surface**: $\text{if}(x < 60, y < -\sin(0.207303 \cdot x), 0)$

⚠ Please note: The porous volume should be oriented with its normals pointing outside the body

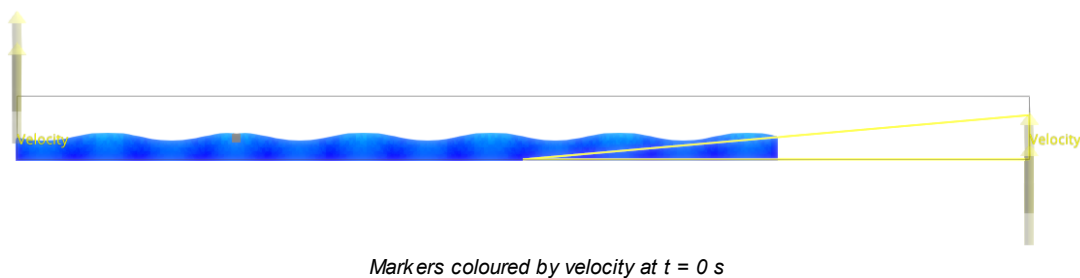
4.2 Launch the calculation

- (a) Save the project
- (b) Set the number of CPUs in **Main menu > Options > Preferences > Engine**
- (c) Press **Run** button > **Start computation**

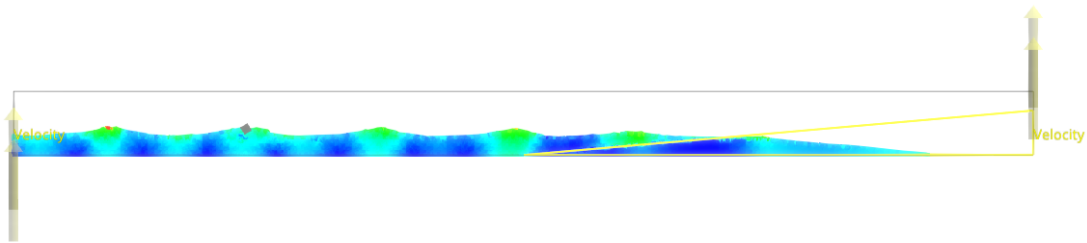
💡 Tip: You can directly load the setup of this problem from the project file `BreakingWaves_Porous.xfp`

4.3 Post-processing

- (a) Select the beach and display it in wireframe visualisation mode. Observe the different initial configurations in both cases (1) in previous beach, and (2) porous beach.



- (b) Play forward the results and observe how the flow evolves inside the porous region.

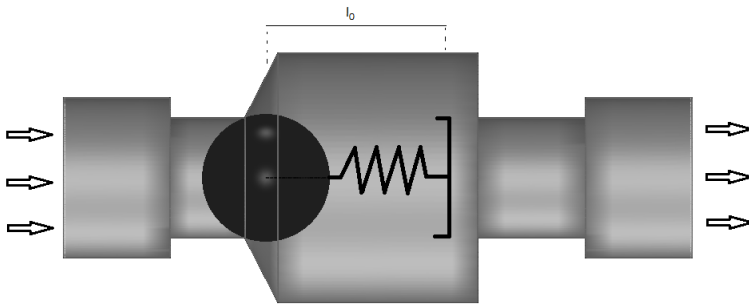


Markers coloured by velocity at $t = 15$ s

Tutorial 06 - Ball check valve

This tutorial illustrates how to set up a fluid-structure interaction problem, namely the case of a ball check valve working with a spring. It consists of a valve with inflow and outflow ducts. The inflow duct is blocked by a sphere mounted on a spring at rest with an extension of l_0 . When the fluid flows in it pushes the ball and exerts a force which opposes the spring reaction. If the fluid flow is ceased on the inlet duct the ball goes back to its original location and seals the passage.

The spring will be modeled by an external force on the sphere acting in the X-direction. The system will be analysed first without and then with a damping force, in order to simulate different regimes of the spring.



The differential equation that describe the movement of the ball due to the spring is:

$$\ddot{x} + 2\xi\dot{x} + \omega_0^2 x = 0, \quad \text{where} \quad \omega_0 = \sqrt{\frac{k}{m}}$$

The parameter ξ is called the damping ratio and determines the behavior of the system, while k is the spring constant. This leads to the following expression of the force exerted by the spring on the sphere:

$$F = -k(x - x_0) - 2\xi\sqrt{km}\dot{x}$$

This tutorial shows how to:

- Create closing boundary surfaces for single phase internal flow
- Impose inlet and outlet boundary conditions on closing surfaces
- Model a 1D-freedom constrained system such as a spring-mass system
- Model damping effects
- Visualise the forces exerted by the fluid on a moving solid
- Visualise the different damping regimes

It is assumed that the reader has completed Tutorial 01, 02, 03, 04 and 05. Some steps in the setup and post-process will not be described in detail.

Before starting the tutorial, please download the project data files from the XFlow[version]_TutorialsPack.zip folder.

Contents

- [Step 1: No damping case](#)
- [Step 2: Under-damping case](#)
- [Step 3: Critical-damping case](#)
- [Step 4: Over-damping case](#)

Step 1: No damping case

In this case no damping will be modeled for the spring. This means ξ will be considered equal to zero.

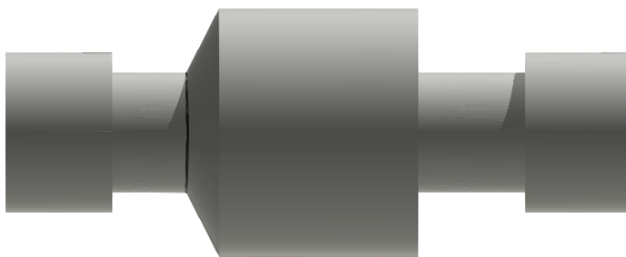
1.1 Import/Create the geometry

Load and setup the valve geometry as indicated below:

- (a) Import the geometry file named `Valve.stp`  In the Import Geometry pop-up window, leave the default options and press **OK**.

As the geometry is small, zoom in to be able to visualize it

- (b) In order to orientate the flow from -X to +X, make a rotation of 180 degrees around Y-axis by changing to **(0, 180, 0)** the geometry orientation: **Geometry > Entities > Shape: Valve > Behavior > Orientation**. The valve is now correctly oriented as shown in the picture below.



Valve geometry

- (c) Check the orientation of the geometry. Remember that the back-face culling helps to distinguish the orientation of the normals of a surface (see Tutorial 02, Step 1, 1.2). The normals always have to point toward the fluid region. To reverse the normals orientation, select the geometry and **right click** in the

Step 1: No damping case

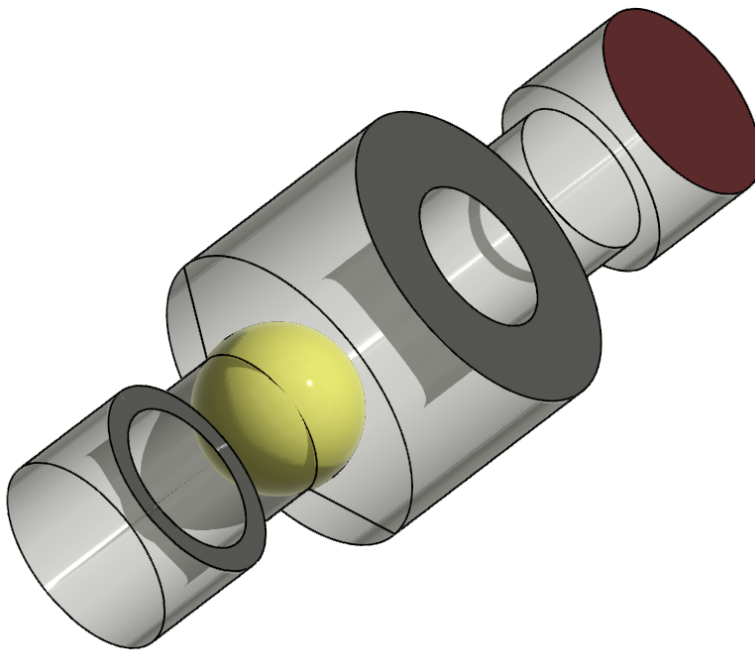
Graphic View window and in the **Graphic View Menu** choose **Reverse orientation**

- (d) Import geometries named `Inflow.stp` and `Outflow.stp` and check they are correctly located, sealing the inlet and outlet.

Now we will create the ball which acts as the valve:

- (a) Create a sphere of radius **0.008 m** at the location **(0, 0, 0)**. To do so, **Main Menu > Geometry > Create Object > Create Sphere** or press  in *Toolbar Object Creation*.
- (b) In **Geometry > Entities > Shape: Sphere > Behaviour > Position**, move it to the initial position at rest: **(-0.014, 0, 0)**. Check its orientation to ensure the domain of fluid to be outside the sphere.
- (c) Check if the orientation of the normals of the sphere is pointing outside. This means that the fluid will be outside of the sphere, and not inside.

The final geometry should consist of a valve with Inflow and Outflow surfaces and a ball obstructing the passage at the inlet, as it is shown in the figure below.



Full geometry at initial conditions

1.2 Engine settings

- (a) Select the **Kernel: 3d**
- (b) Select the **Flow model: Single phase**
- (c) Select the **Analysis type: Internal**. In the *Graphic View* window the wind tunnel disappears and the user will have to provide the geometry that contains the fluid, the valve.

- (d) Leave the **Thermal model: Isothermal**, since this is an incompressible simulation.
- (e) Leave the default **Turbulence settings**

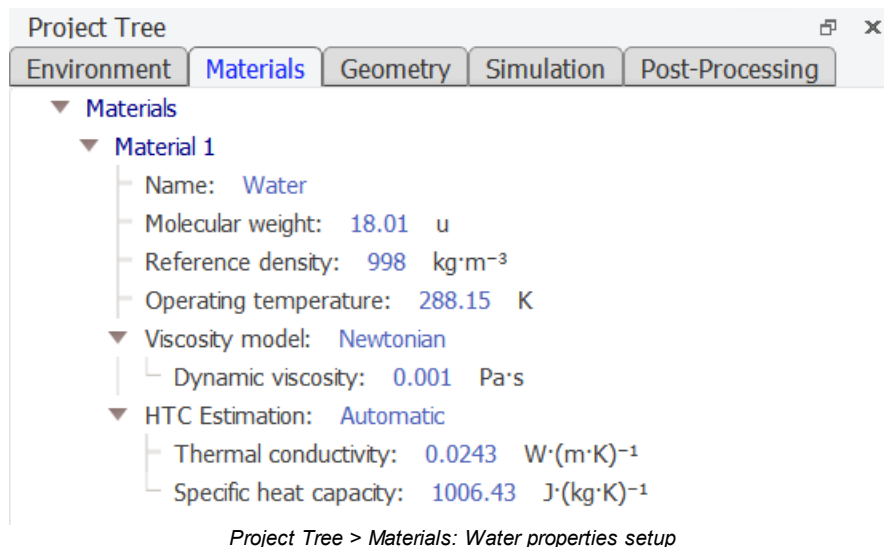
1.3 Environment settings

Leave the default configuration, this means: (i) no **External acceleration laws (0, 0, 0)**; and (ii) **user defined Initial conditions** with zero initial values. The system should be at rest at the first frame.

1.4 Material settings

By default, the **Material 1** used for an internal single phase analysis is air. However, in this tutorial the operating fluid is water. So, change the fluid properties to make the simulation with water:

- (a) **Name: Water**
- (b) **Reference density: 998 kg·m⁻³**
- (c) **Viscosity model: Newtonian**, with **Dynamic viscosity: 0.001 Pa·s**



1.5 Geometry settings

The geometries will now be set up to define boundary conditions and constraints.

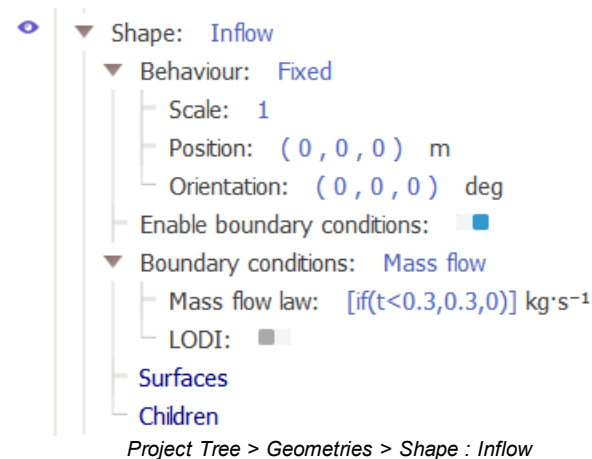
- (a) Valve geometry - **Shape: Valve**:
 - **Behaviour: Fixed**, since this will be the fixed domain outline
 - The valve will remain fixed at the **Position (0, 0, 0)** with **Orientation (0, 180, 0)** (This has been set up previously).
 - **Boundary conditions: Wall** with **Automatic Wall model** and **0 Wall roughness**.

Step 1: No damping case



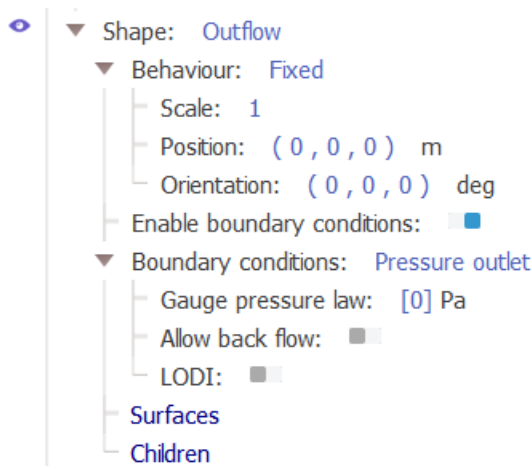
(b) Inflow surface - **Shape: Inflow**:

- **Behaviour: Fixed**
- **Position (0, 0, 0)** and **Orientation (0, 0, 0)** should already be set up correctly.
- Define the Inflow surface as a mass flow boundary condition of **0.3 kg·s⁻¹** for **t < 0.3 s** and **0** after. **Boundary conditions: Inlet > Mass flow** with **Mass flow law: if(t<0.3,0.3,0) kg·s⁻¹**. Leave the **LODI** option disabled.



(c) Outflow surface - **Shape: Outflow**:

- **Behaviour: Fixed**
- **Position (0, 0, 0)** and **Orientation (0, 0, 0)** should already be set up correctly.
- Set up the Outflow surface to be a pressure outlet condition at atmospheric pressure. **Boundary conditions: Outlet > Gauge pressure outlet** with **Gauge pressure law: 0 Pa**. Leave the **LODI** and the **Allow backflow** option disabled.



Project Tree > Geometries > Shape : Outflow

(d) Ball - **Shape: Sphere**: set up the Sphere to simulate the spring. In this section the spring is modeled with no damping:

- **Behaviour: Rigid body dynamics** since we want to apply spring forces on it which is a 1D constrained movement
- Make sure that the **Position** is still **(-0.014, 0, 0)** as defined previously. Leave velocity, orientation, and angular velocity as they are by default **(0, 0, 0)**.
- **Inertia tensor** will be left **Automatic**, assuming a uniform mass distribution, and **Density: 11350 kg·m⁻³**, which correspond to the lead density.
- Keep the default **Interaction properties**
- Let the sphere translate in X direction by setting the **Translation (Global)** option to **Axis x**. Don't let it rotate by setting **Rotation (Local): Fixed**.
- **External force global > X: -150*(px-(-0.014)) N**, to simulate a spring with no damping and with a spring constant of **150 N·m⁻¹**
- **Boundary conditions: Wall** with **Automatic Wall model** and **0 Wall roughness**.

Step 1: No damping case



1.6 Simulation settings

- (a) Time > Simulation time: **0.5 s**, with a **Fixed automatic Time step mode** and a **Courant** number of **1**
- (b) Lattice discretization > Resolved scale: **0.001 m** with no refinement (Refinement algorithm: **Disabled**)
- (c) Store data > Frames frequency: **300 Hz**
- (d) Leave the Numerical data frequency to **Solver time step**
- (e) Leave the computation of the averaged fields and markers disabled

Please note: The Resolved scale is in fact quite coarse for such a case because the dimensions of

the valve are very small. This choice is made on purpose for this tutorial in order to reduce the calculation time since it is enough to simulate and show the interaction between the fluid and the sphere for different damping. Such a coarse resolution should not be used for an accurate analysis of the solution.

1.7 Launch the calculation

- (a) Save the project



Tip: You can directly load the setup of this problem from the project file `BallCheckValve_NoDamping.xfp`

- (b) Set the number of CPUs in **Main menu > Options > Preferences > Engine**

- (c) Press **Run** button > **Start computation**



Tip: To accelerate the calculation time you can select an empty frame (the last one for instance). This way XFlow will not post-process data in real-time which saves the computer resources for the calculation.

1.8 Load data

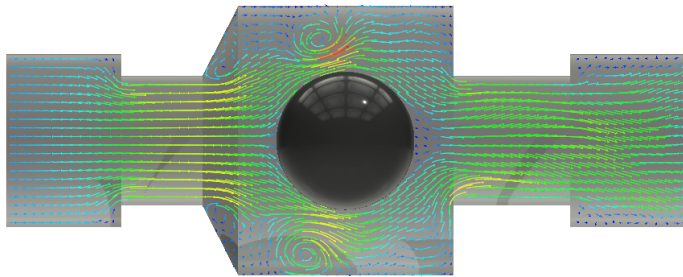
If the user has not closed the GUI, the data will be automatically loaded. Otherwise, load the simulation

data 

1.9 Post-processing: Vectors

- Create a cutting plane by clicking on the icon  on the Post-processing toolbar. A **Cutting plane 1** will appear in the *Post-processing* tab. Change its position to 0 m to visualise the flow at the mid section of the Valve geometry.
- Select **Vectors** as **Visualisation mode** of the **Cutting plane 1**. Adjust **Arrows density** to 0.5 and **Arrows length** to 1.0.
- Change the velocity range to [0; 4.5] by hitting .
- Set the view to Right view by pressing .
- Play forward the simulation by pressing .

Step 1: No damping case



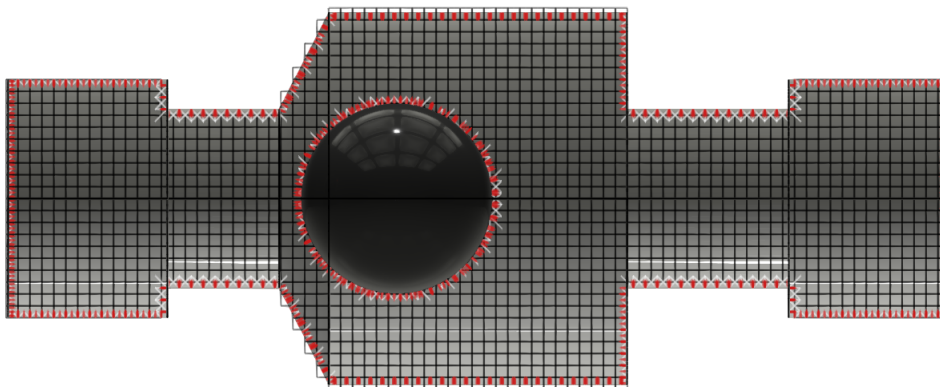
Velocity vectors cutting plane at 0.28s

Observe that the fluid is pushing the sphere and that the sphere has a periodic movement as expected since, in this case, there is no damping.

1.10 Post-processing: Broken links

In order to observe the treatment of the moving geometry, the near-wall treatment can be observed through the broken links (see [Tutorial 01 - Step 4](#)). The application mode must be set to **Expert** in order to access the broken link option. The Expert mode makes additional advanced options accessible.

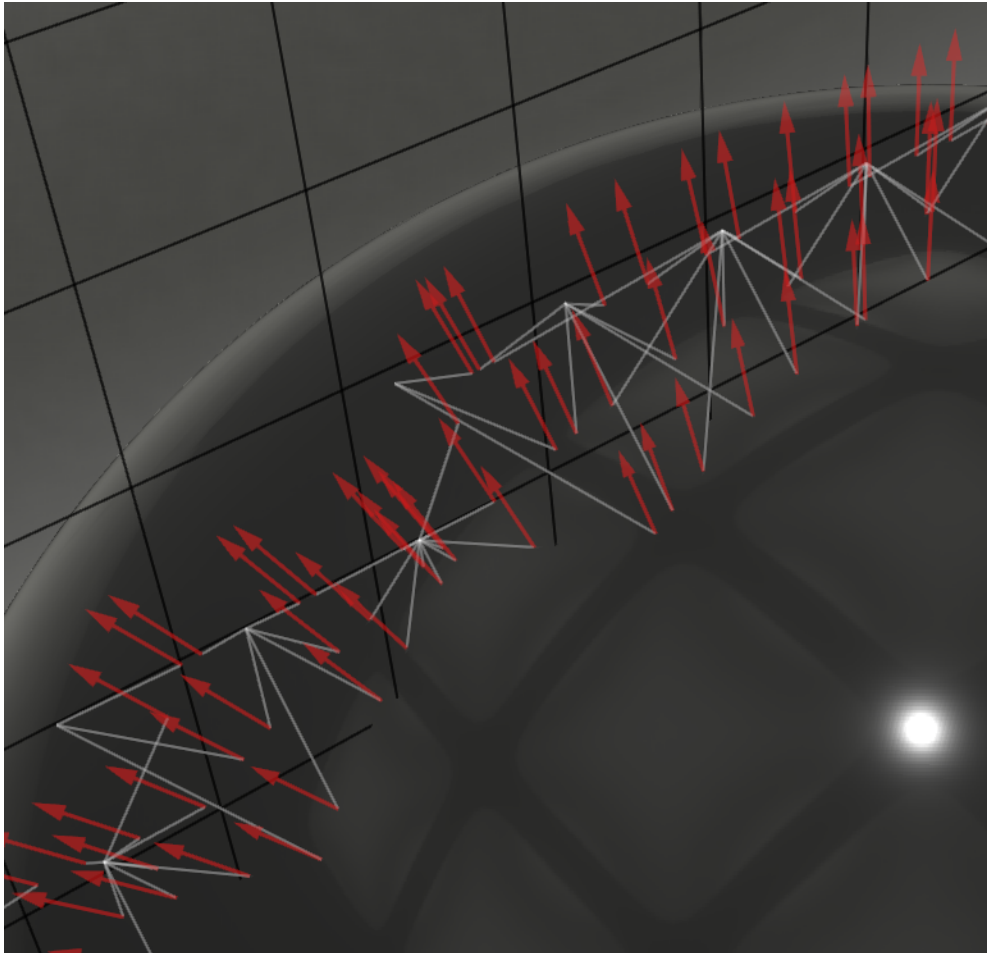
- (a) Save the current project.
- (b) Set the Expert mode through the preferences options  : **Options > Preferences > Application > Application mode: Expert.**
- (c) Restart XFlow so the new application mode becomes effective.
- (d) Load the ball check valve project again. XFlow will ask if you want to load the simulation data, press **Yes**.
- (e) Switch to **Cutting Plane 1 > Visualization mode > Domain structure**. Observe a new **Broken links** option is available: switch it **On**. Play forward the frames with the icon  in order to observe how the broken links are recalculated and updated every new position of the moving sphere.



Domain structure, broken links, and surface normals

The broken links are the lattice velocity directions that intersect with geometry. These intersection are where boundary conditions are applied, showing that the spatial discretization is at a greater order

than the voxelization presented in the cutting planes. Every red arrow represents a discretization point on the geometry surface, and the white lines show the broken links from the near-wall lattice nodes until they intersect the geometry surface:

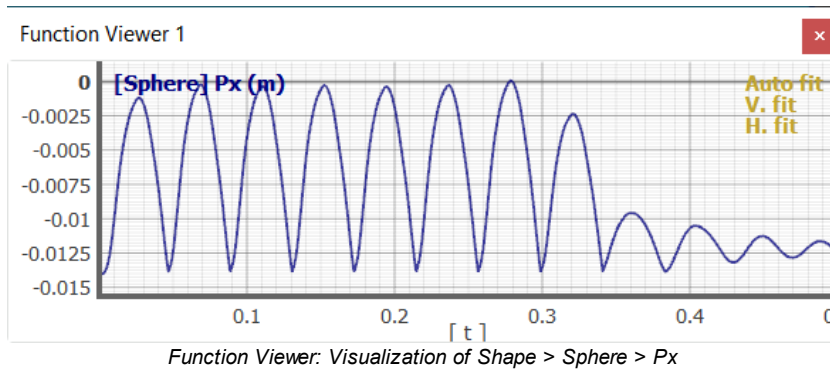


Zoom on broken links (white lines) and surface normals (red arrows)

1.11 Post-processing: Sphere position

- (f) Reset to the first frame by pressing  and switch off the **Cutting plane 1** with help of its checkbox 
- (g) Plot the X-position of the Shape: Sphere by **right-click** on the *Function Viewer*, in the drop-down menu select **Shapes > Sphere > Px**
- (h) Refit the range of view by clicking on **Auto fit** at the right-top of the *Function Viewer*
- (i) Play the simulation again and observe the evolution of the sphere position in real time

Step 1: No damping case



Note that the evolution is periodic until 0.35 seconds, approximately, when the mass flow goes down to 0 $\text{kg}\cdot\text{s}^{-1}$. The simulation of the non-damping spring is therefore successful.

Step 2: Under-damping case

In this case the spring will be modeled with under-damping. The value of the spring constant will be taken as $\xi = 0.5$.

2.1 Save as a new project

Main menu > File > Save project as or  in **Toolbar File** and save the project with a different name (for instance: BallCheckValve_UnderDamping.xfp).

2.2 Run the under-damping case

Taking the ["No damping" case](#) as reference, modify the following settings:

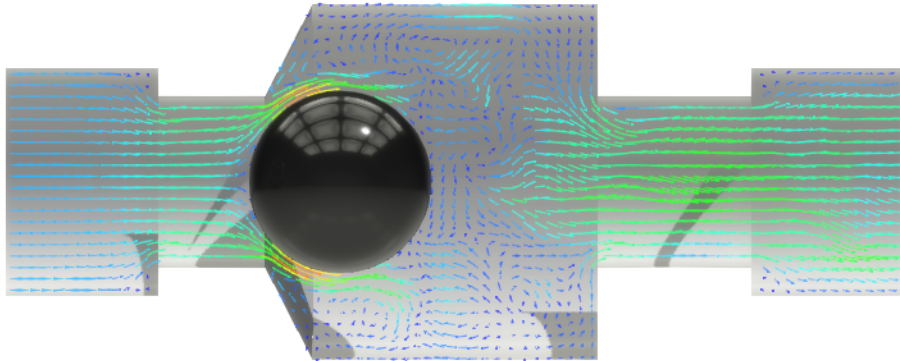
- In **Geometry > Entities > Shape: Sphere > Behavior: Rigid body dynamics > Constraints > External forces** replace **External force global > X: $[-150*(px-(-0.014))]$** by **$[-150*(px-(-0.014))-1.91*vx]$ N**. The new term corresponds to the damping term and depends on the velocity. According to the [equations](#) and taking $\xi = 0.5$, the calculation leads to $2\xi(km)^{1/2} = 1.91$.
- Save the case, check the number of CPUs

 **Tip:** You can directly load the setup of this problem from the project file BallCheckValve_UnderDamping.xfp

- Press **Run** button > **Start computation**

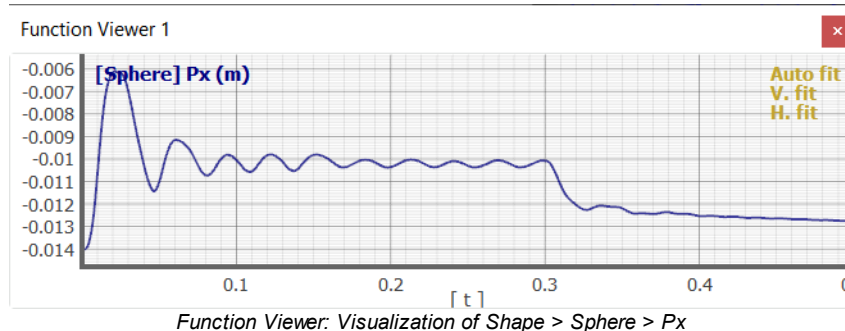
2.3 Post-process the results

- (a) Observe the velocity vectors in the **Cutting plane 1** as done in [section 1.9](#). Note that the sphere is reaching an equilibrium position after few oscillation, due to the damping. The flow is then smoother until the mass flow stops and the sphere is then going back to its original location.



Velocity vectors cutting plane at 0.3s

- (b) Switch the cutting plane off and rewind
- (c) Display the X-position of the sphere as done in [section 1.10](#). Note that the damping of the system is effective. Observe that the second oscillation is reduced by approximately half of the first one, which is the behavior expected from a spring that has a spring constant of 0.5.



Step 3: Critical-damping case

In this case the spring will be modeled with critical-damping. The value of the spring constant will be taken as $\xi = 1$.

3.1 Save as a new project

Main menu > File > Save project as or  in **Toolbar File** and save the project with a different name (for instance: BallCheckValve_CriticalDamping.xfp).

3.2 Run the under-damping case

Step 3: Critical-damping case

Taking the ["Under damping" case](#) as reference, modify the following settings:

- (a) In **Geometry > Entities > Shape: Sphere > Behavior: Rigid body dynamics > Constraints > External forces** replace **External force global > X: $[-150*(px-(-0.014))-1.91*vx]$** by **$[-150*(px-(-0.014))-3.82*vx]$ N**. According to the [equations](#) and taking $\xi = 1$, the calculation leads to $2\xi(km)^{1/2} = 3.82$.

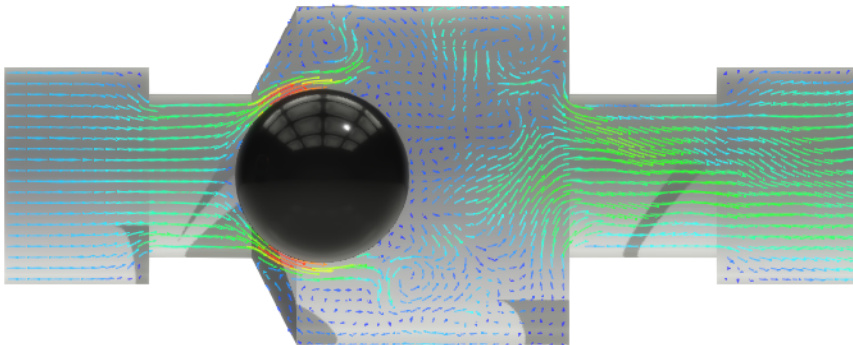
- (b) Save the case, check the number of CPUs

 **Tip:** You can directly load the setup of this problem from the project file `BallCheckValve_CriticalDamping.xfp`

- (c) Press **Run** button > **Start computation**

3.3 Post-process the results

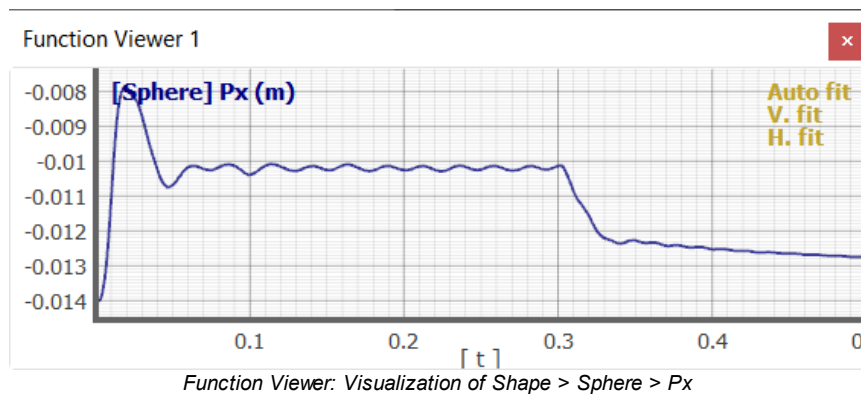
- (a) Observe the velocity vectors in the **Cutting plane 1** as done in [section 1.9](#). Note that the sphere is reaching an equilibrium position directly without completing a full oscillation. The flow is smooth until the mass flow stops and the sphere is then going back to its original location slightly slower than on the previous cases



Velocity vectors cutting plane at 0.3s

- (b) Switch the cutting plane off and rewind

- (c) Display the X-position of the sphere as done in [section 1.10](#). Note that the damping of the system is effective. Observe that the first oscillation is not fully completed and the system stabilise directly: this is the critical damping.



Function Viewer: Visualization of Shape > Sphere > Px

Step 4: Over-damping case

In this case the spring will be modeled with over-damping. The value of the spring constant will be taken as $\xi = 6$ to show a case of very high damping.

4.1 Save as a new project

Main menu > File > Save project as or  in **Toolbar File** and save the project with a different name (for instance: BallCheckValve_OverDamping.xfp).

4.2 Run the over-damping case

Taking the ["Critical-damping" case](#) as reference, modify the following settings:

- (a) In **Geometry > Entities > Shape: Sphere > Behaviour: Rigid body dynamics > Constraints > External forces** replace **External force global > X: [-150*(px-(-0.014))-3.82*vx]** by **[-150*(px-(-0.014))-22.92*vx] N**. According to the [equations](#) and taking $\xi = 6$, the calculation leads to $2\xi(km)^{1/2} = 22.92$.

- (b) Save the case, check the number of CPUs

 **Tip:** You can directly load the setup of this problem from the project file BallCheckValve_OverDamping.xfp

- (c) Press **Run** button > **Start computation**

4.3 Check the numerical stability

- (a) Go to the *Function Viewer* and check the stability parameter by **right clicking** on the *Function Viewer* and selecting **Stability Parameter**

- (b) Observe that the courant number is going over a value of 1 and that the message "[WARNING] Stability parameter too high, check pressure at (-1.90-002,-6.50e-003,-3.50e-003) m or reduce the time step." is output in the Message View between 0.007 and 0.025 s, showing you what field and at what location needs to be inspected.

In this specific case, it seems the pressure has too high values at the coordinate (-0.0190, -0.0065, -0.0035) m and means the numerical stability is not ensured.

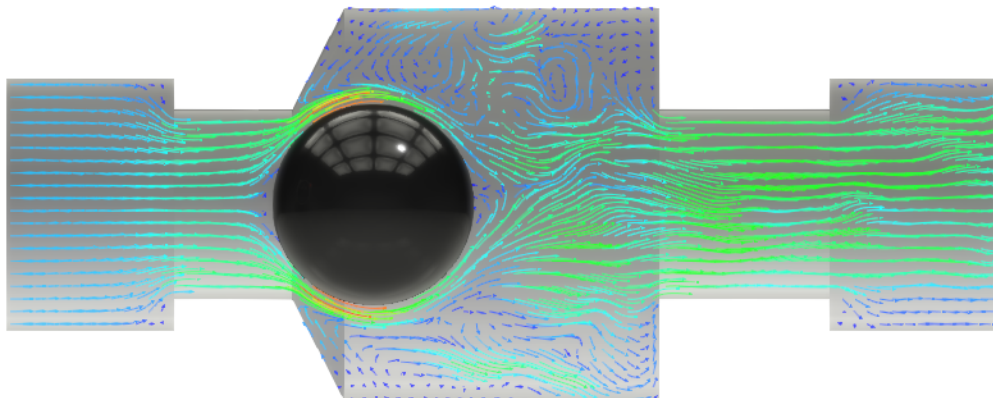
- (c) Adjust the **Courant** number to 0.7 in the **Simulation** tree. This will multiply the time step automatically estimated by XFlow by a factor of 0.7, which will lower the time step.

- (d) Press **Run** button > **Start computation**

Step 4: Over-damping case

4.4 Post-process the results

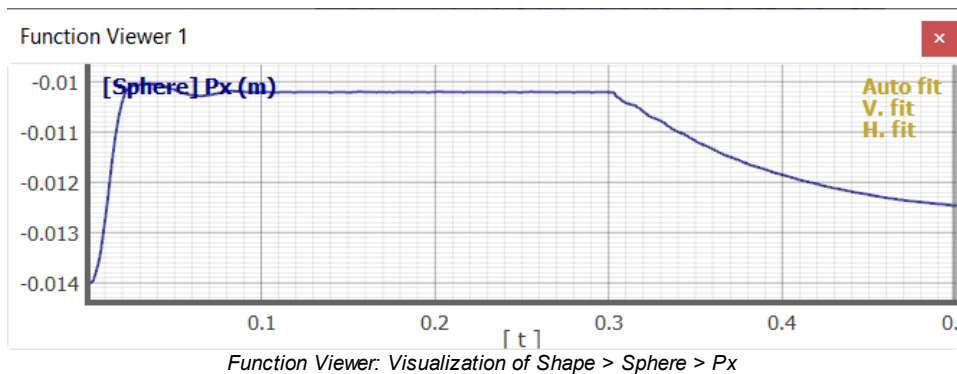
(a) Observe the velocity vectors in the **Cutting plane 1** as done in [section 1.9](#). Note that the sphere is reaching an equilibrium position slowly and does not oscillate at all. The flow tends to be smooth later than in the previous cases since the sphere slowly goes to its maximum X location. When the mass flow stops the sphere is then going back to its original location much slower than previous cases



Velocity vectors cutting plane at 0.3s

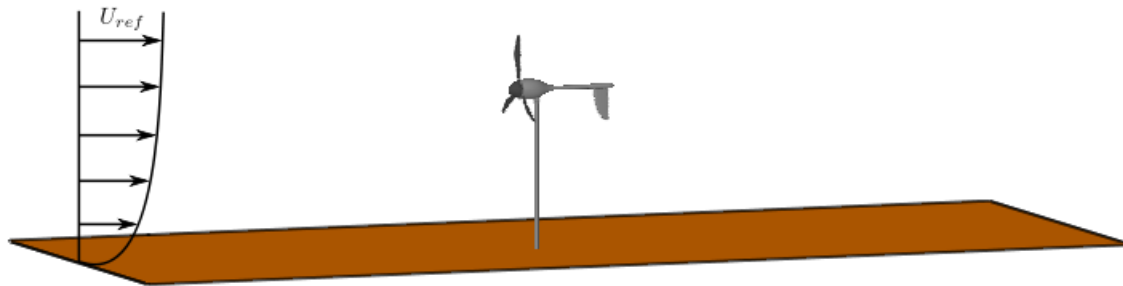
(b) Switch the cutting plane off and rewind

(c) Display the X-position of the sphere as done in [section 1.10](#). Note that the damping of the system is effective. Observe that no oscillation occurs: this is the over-damping. The drawback of such an over-damped system is that the sphere struggles to go back to its original location, and therefore does not seal the inlet passage properly.



Tutorial 07 - Wind turbine

This tutorial illustrates the setup and solution of the flow around a wind turbine and its aerodynamics analysis. The rotor will rotate first with a prescribed angular velocity and later due to the forces exerted by the flow. The wind profile is assumed to follow a power-law.



This tutorial shows how to:

- Heal a geometry with holes
- Set appropriate parameters for importing IGES files
- Set enforced and rigid body motions for the rotor
- Define a power-law for the inlet velocity
- Perform measurements with probes

It is assumed that the reader has completed Tutorial 01, 02 and 06. Some steps in the setup and post-process will not be described in detail.

Before starting the tutorial, please download the project data files from the XFlow[version]_TutorialsPack.zip folder.

⚠ Please note: This tutorial requires several hours of computation and large hard disk resources.

Contents

- [Step 1: Geometry healing](#)
- [Step 2: Enforced behaviour - Setup](#)
- [Step 3: Enforced behaviour - Post-processing](#)
- [Step 4: Rigid body dynamics behaviour](#)

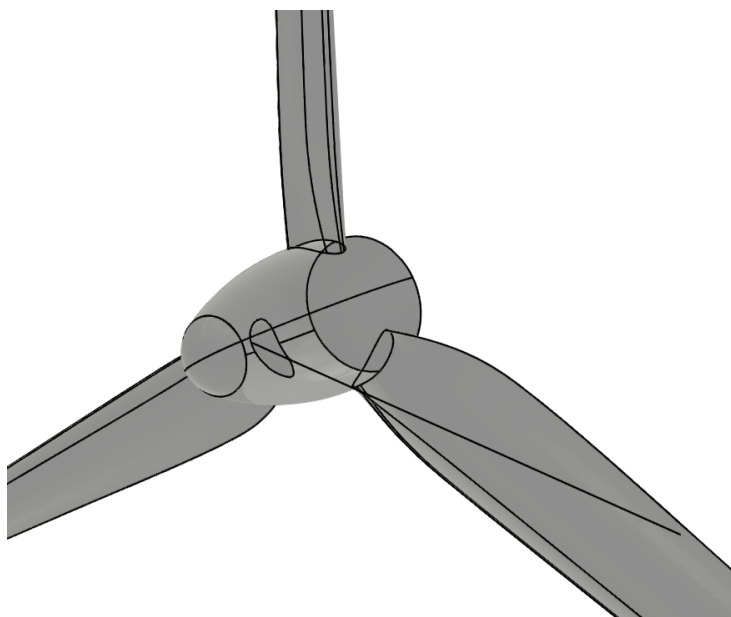
Step 1: Geometry healing

1.1 Import the CAD file blades-to-heal.igs

- (a) Set the Mesh deflection to 1 in **Main menu > Options > Preferences > Geometry > Mesh Deflection**. This parameter (between 0 and 2) determines the size of the automatic tessellation for parametric geometries (the lower the value the higher the number of polygons created)
- (b) Importing the CAD file blades-to-heal.igs, which is in parametric IGES format
- (c) Select the blades and show the tessellation by **right clicking** in the *Graphic View* window to show the Graphic View Menu. Then, choose: **Visualisation mode > Mesh**
- (d) Show the geometrical properties (select the blades, **right click** in the *Graphic View* window to show the Graphic View Menu and choose: **Show geometrical properties**). Observe that it has around 6,000 polygons
- (e) Delete this shape, change the OpenGL mesh deflection to 0.1 and import the geometry again. Observe that now the tessellation is finer, it has around 50,000 polygons. Keep this geometry.

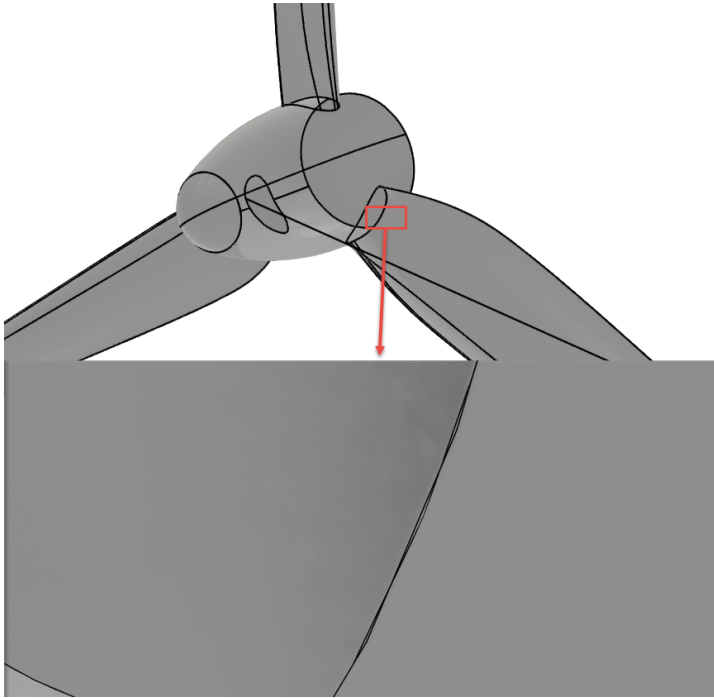
1.2 Check the surfaces orientation

Select the blades, **right click** in the *Graphic View* window to show the Graphic View Menu and make sure that **Back-face culling** is enabled. It can be observed that some surfaces of the geometry have the normals pointing outwards and others pointing inwards.



1.3 Show holes

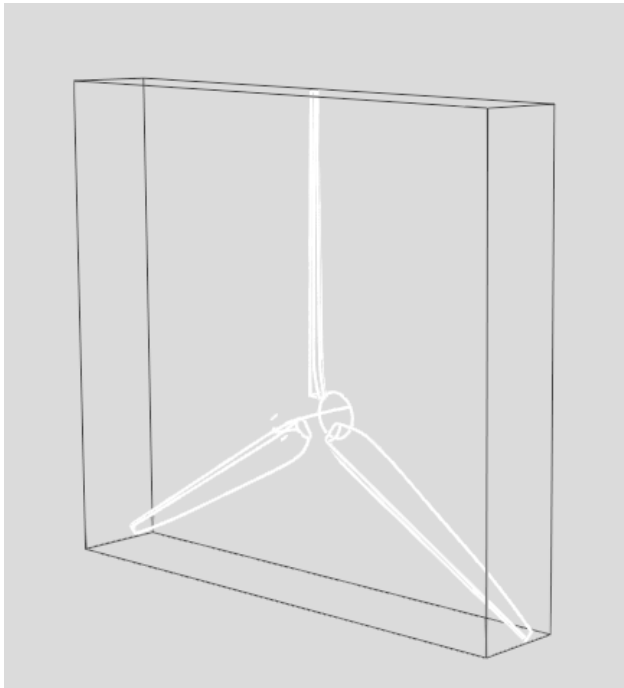
Additionally, the surfaces (patches) defining the geometry are not sewed, there are gaps between them. For instance, the blades are not properly joined to the hub. Zoom in to observe that the joint edge is a double curve.



You can view all the holes in the model by selecting the object, clicking **Main menu > Geometry >**

Show/hide holes or  and visualizing the object in bounding box mode (**right click** in the *Graphic View window > Visualisation mode > Bounding Box*).

Step 1: Geometry healing



Geometry blades-to-heal (Bounding Box) + Show holes

XFlow allows intersecting surfaces but holes in the model may cause the fluid to enter inside it. The tolerance is approximately 1% of the finest resolution used in the computation.

Tip: A good practice is to make sure your model is exported as a watertight solid: all the surfaces must be joined with no naked edges to avoid gaps. XFlow is tolerant to crossing surfaces, but the computation may encounter problems with surface patches separated by gaps.

To clean the information about the holes, press again **Main menu > Geometry > Show/hide holes** or

 . Return to the Shading visualisation mode.

1.4 Healing

XFlow has some basic healing operations such as fix small lines, fix small faces, remove isolated lines, or sew faces.

(a) Select the geometry shape and perform the following healing operation (please make sure that the geometry is selected when performing such operation):

Main menu > Geometry > Healing > Sew faces

(b) Check that most of the holes have disappeared after healing.

⚠ Please note: If the geometry is not watertight, fluid will leak inside it which will be initialised with the inlet velocity condition, as the rest of the fluid. However, since this in this region the air is confined, it will generate pressure waves inside the geometry which will lead to wrong values of the aerodynamic forces. Furthermore, closed volumes with a small opening will equilibrate the interior pressure to the local static

pressure at the hole, leading to wrong overall forces.

- (c) Remove the geometry: blades-to-heal.

Step 2: Enforced behaviour - Case setup

2.1 Import the geometry

The geometry used in [Step 1](#) will not be used anymore in this tutorial. Now import the geometry files: `tower.nfb` and `blades.nfb`.

The wind turbine consists of a 3-blade rotor with a diameter of 52 meters and a tower of 82 meters height.

2.2 Engine settings

- (a) Select the **3d Kernel**, the **Single phase Flow model**, the **External Analysis type** and **Isothermal** for the **Thermal model**.
- (b) Leave the default **Turbulence settings**

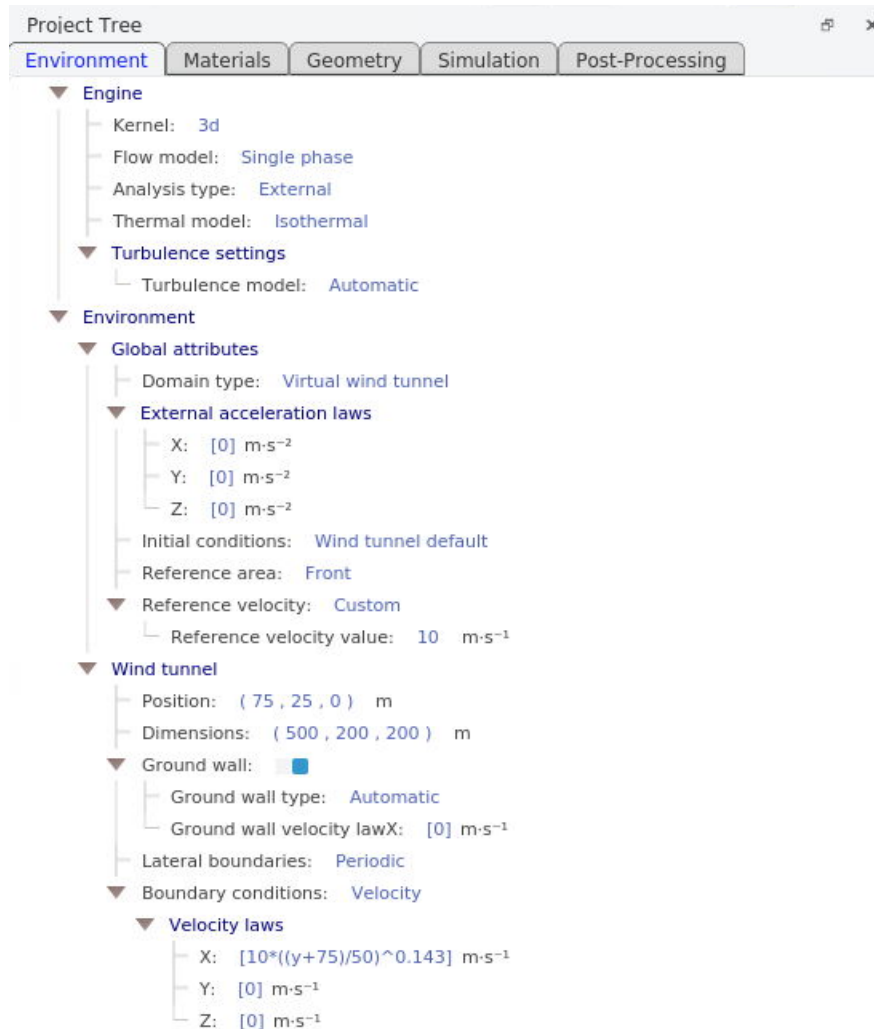
2.3 Environment settings

- (a) Set the **Domain type** to **Virtual wind tunnel**
- (b) Leave to zero the **Ext. acceleration laws**
- (c) **Initial conditions: Wind tunnel default**
- (d) Keep the **Front Reference area** and the set **Reference velocity** to **custom** and a value of **10 m·s⁻¹**
- (e) Introduce the wind tunnel **Dimensions: (500, 200, 200) m**. By default, the wind tunnel is centred at **(0, 0, 0) m**.
- (f) Change the wind tunnel **Position** to **(75, 25, 0) m**. This sets the ground at **Y = -75 m**. You may check it with the help of the grid (**Main Menu > Options > Preferences > Graphic View > Show grid: On**).
- (g) Enable the **Ground wall**, with **Automatic Ground wall type** and **Ground wall velocity law X: 0 m·s⁻¹**
- (h) Set the **Lateral boundaries** to **Periodic** and the inlet **Boundary conditions** as **Velocity** defined by the following power-law profile

$$u = u_{ref} \left(\frac{y - y_0}{y_{ref}} \right)^\alpha$$

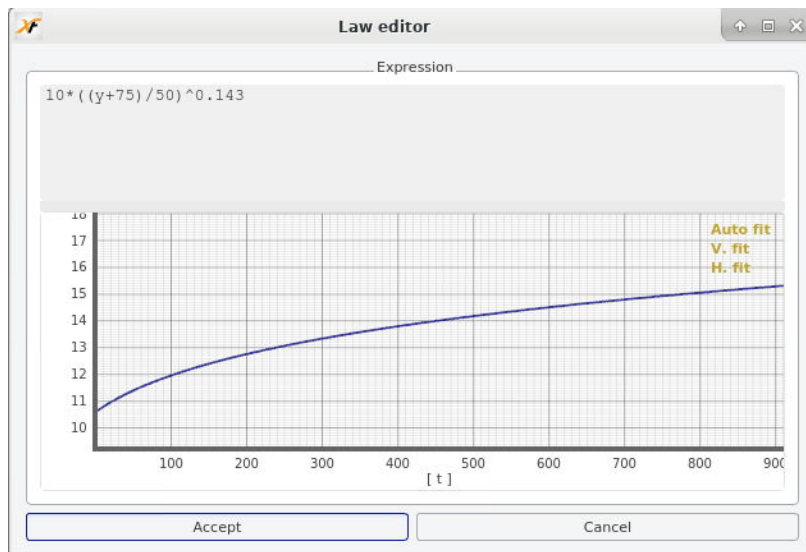
Step 2: Enforced behaviour - Case setup

with reference height $y_{ref} = 50 \text{ m}$, velocity at the reference height $u_{ref} = 10 \text{ m}\cdot\text{s}^{-1}$, ground height $y_0 = -75 \text{ m}$ and shear exponent $\alpha = 0.143$ corresponding to neutral atmospheric conditions. The XFlow input is $10*((y+75)/50)^{0.143} \text{ m}\cdot\text{s}^{-1}$



(i) Display the graph of the X-velocity law in the *Function Viewer* by clicking the expression followed by clicking , highlighted in the snapshot below. Check the exponential shape of the wind profile. At height 50 m ($y = -25$) the wind velocity is $10 \text{ m}\cdot\text{s}^{-1}$. To adjust the range of the *Function Viewer*:

- Zoom: **middle (wheel) mouse button + drag** or **roll wheel**
- Pan: **left mouse button + drag**



Wind velocity profile at inlet

Tip: This function is not defined for $y < -75$, therefore the function is not shown if the X-axis range includes values lower than -75. You might pan the graph to center your X range in $[0;20000]$ to check the law appropriately and use the **V. fit** function.

2.4 Material settings

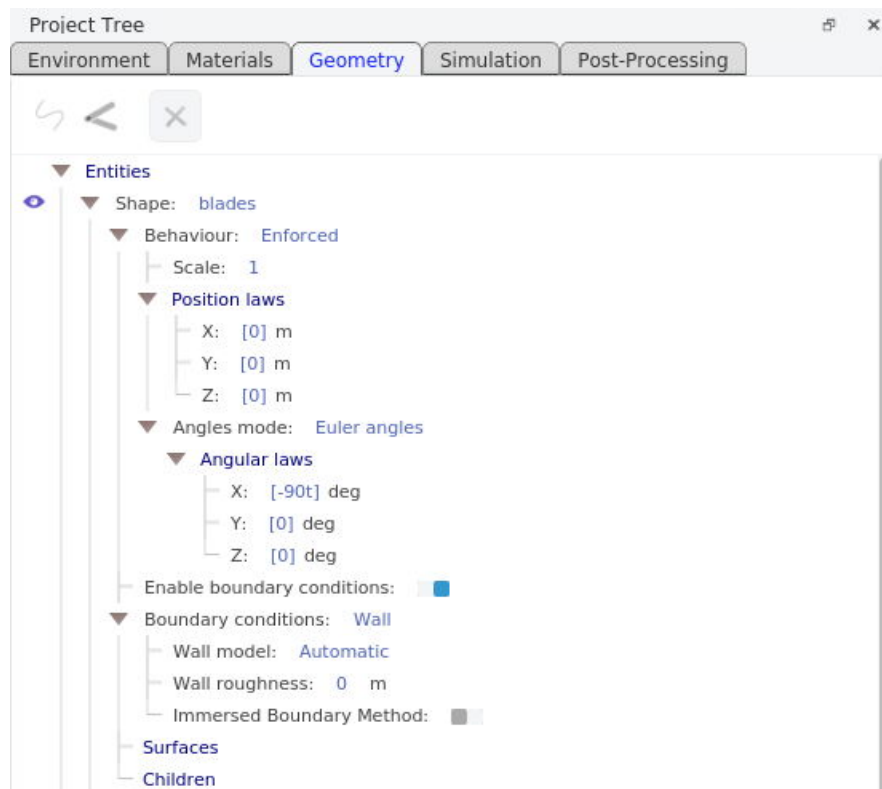
Keep the default parameters of **Material 1**, which by default correspond to those of air.

2.5 Geometry settings

In Step 2, the blades are modeled so that they rotate at a constant speed of 15 rpm. To this end, proceed with the following setup:

- Geometry > Entities > Shape: tower > Behaviour > Fixed** and **Boundary conditions: Wall** with **Automatic Wall model** and **0 Wall roughness**.
- Geometry > Entities > Shape: blades: Behaviour: Enforced**
- Check the local axes of the **blades** by selecting the geometry, **right clicking** in the *Graphic View* window and enabling **Show > Local axes**. Observe that the local X-axis corresponds to the rotation axis.
- Geometry > Entities > Shape: blades:** set the **Angular law** for the X-axis to **[-90t] deg** ($15 \text{ rpm} = 90 \text{ deg} \cdot \text{s}^{-1}$)
- Check the enforced motion of the blades by pressing  or dragging the current frame along the timeline

Step 2: Enforced behaviour - Case setup



2.6 Simulation settings

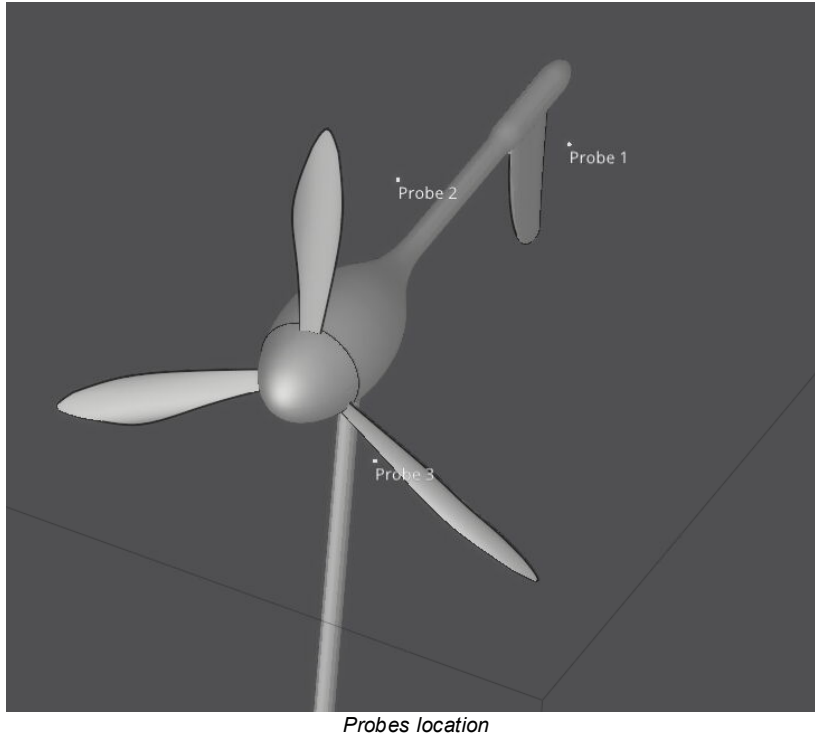
- (a) Set the **Simulation time** to **12 s**
- (b) **Time step mode** will be **Fixed automatic** with **Courant: 1**
- (c) Set **8 m** for the **Resolved scale** (far field resolution)
- (d) Select **Refinement algorithm: Adaptive refinement**
- (e) Set **0.5 m** as target resolution for the wake, tower and blades (**Wake resolution; Shape: blades > Target resolved scale; Shape: tower > Target resolved scale**)
- (f) **Store data** with a **Frames frequency** of **25 Hz** and set the **Numerical data frequency** to **Solver time step**.
- (g) Leave **Save averaged fields** and **Compute markers** disabled
- (h) Switch on the **Q-Criterion** fields to save

2.7 Probes

Probes are predefined points where data (velocity, pressure, etc.) are measured and saved during computation. They allow to save this data at the solver time step frequency instead of the frames frequency as with the sensors, and therefore provide better measurements.

- (a) Create a probe by **right clicking** the string **Probes** and choose **Add probe**

- (b) Set **Probe 1 > Position** to **(50, -10, 0) m**
- (c) Create two additional probes at the following locations:
Probe 2 > Position: (15, 10, 0) m
Probe 3 > Position: (15, -20, 0) m
- (d) Show the probes position by switching on the visibility icon in the *Project Tree*



2.8 Launch the calculation

- (a) Save the project

 **Tip:** You can directly load the setup of this problem from the project file `WindTurbine.xfp`

- (b) Set the number of CPUs in **Main menu > Options > Preferences > Engine**
- (c) Press **Run** button > **Start computation**

 **Tip:** During the calculation it is recommended that you set the timeline to an empty frame not to slow down the XFlow performance.

Step 3: Enforced behaviour - Post-processing

3.1 Monitor the evolution of the number of active particles

The refinement adapted to the wake causes the number of active particles to change during the computation.

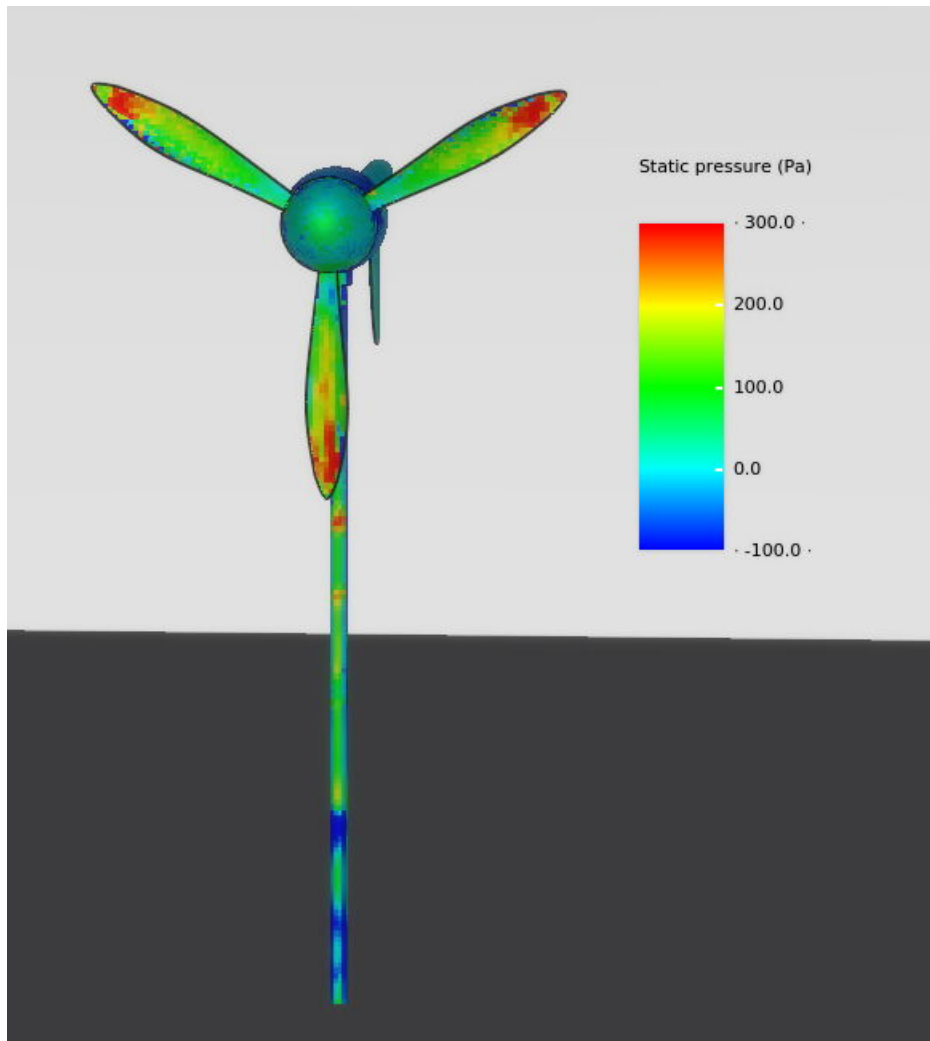
Load the frames 10, 100 and 200 and check the number of particles in the *Message View* is approximately as in the Table below.

	Frame 10	Frame 100	Frame 200
Level 0	38,000	37,000	37,000
Level 1	8,000	9,000	12,000
Level 2	17,000	24,000	35,000
Level 3	41,000	92,000	162,000
Level 4	171,000	669,000	1,315,000
Total	275,000	832,000	1,600,000

In the *Message View*, the number of particles in the domain can also be checked at each iteration by looking at "Num elems [xxx]" during the computation.

3.2 Visualise surface info

- Add a cutting plane in the Z axis **Post-processing > Cutting planes > Add cutting plane: 3d field** and choose **Velocity** as **Field**
- The wind velocity at hub is $u(y=0) = 10.6 \text{ m.s}^{-1}$. Adjust the velocity range (**Main menu > Simulation data > Analysis settings**) to [0, 20] and navigate through different frames.
- Enable **Post-processing > Show > Surface info: 3d field** and choose **Pressure** as **Field**. Adjust the range to [-100, 300] Pa in **Main menu > Simulation data > Analysis settings > Surface info**.

Surface Pressure at $t = 10\text{ s}$

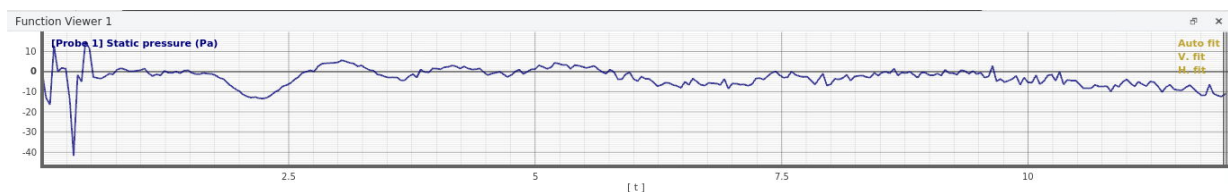
3.3 Probes measurements

(a) Monitor the evolution of the static pressure at the probes:

right click in *Function Viewer* > **Probes** > **Probe 1** > **Static pressure**

right click in *Function Viewer* > **Probes** > **Probe 2** > **Static pressure**

right click in *Function Viewer* > **Probes** > **Probe 3** > **Static pressure**



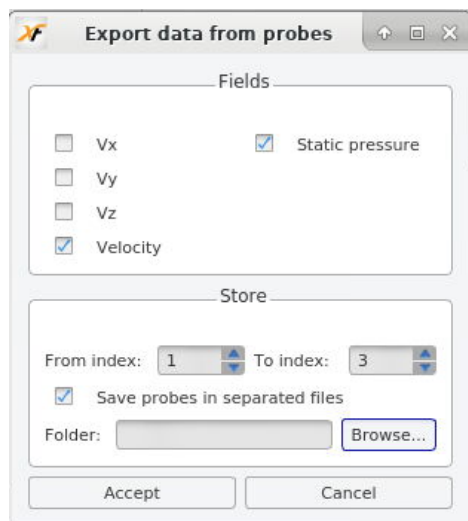
Static pressure at probe 1

(b) You can export the data of the probe **right clicking** in *Function Viewer* > **Export all**. A pop-up

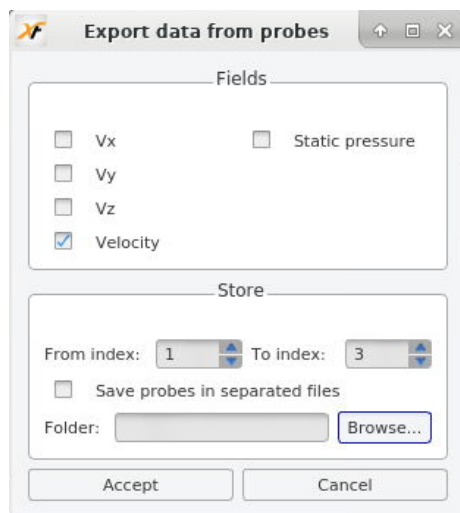
Step 3: Enforced behaviour - Post-processing

window appears and asks you to select which fields and which probes would you like to export. Select the following options:

- **Fields:** **Velocity** and **Static pressure**
- **Store:** **From index:1, To index: 3**. This means that you will export the data of the Probe1, Probe 2 and Probe 3.
- Enable the **"Save probes in separated files"** option. This means that the data of each probe will be exported in separated files called "probe_1.txt", "probe_2.txt" and "probe_3.txt" containing the velocity and static pressure measurements.
- Select the folder where you want to save the files



(c) Export only the **Velocity** field measured by the probes 1 to 3 and leaving the **"Save probes in separated files"** option unchecked:



XFlow will export the Velocity measurements of the three probes to one file called "probe_1_3.txt".

 **Please note:** The Probes are working in a similar way as the Sensors (used in the Tutorial 04 for example). The only difference is that Probes can measure the information at the **Numerical data frequency** instead of the **Frames frequency** as for the Sensors, and therefore save data at the solver frequency which can be much higher. The Probes need to be defined before running the calculation to be able to save data at solver frequency, as they will save data continuously during the calculation. Furthermore, the Probes do not need to refresh at every frame in order to load the data as with Sensors, reducing time for post-processing. Hence, Probes are usually recommended over Sensors.

Step 4: Rigid body dynamics behaviour

4.1 Case setup

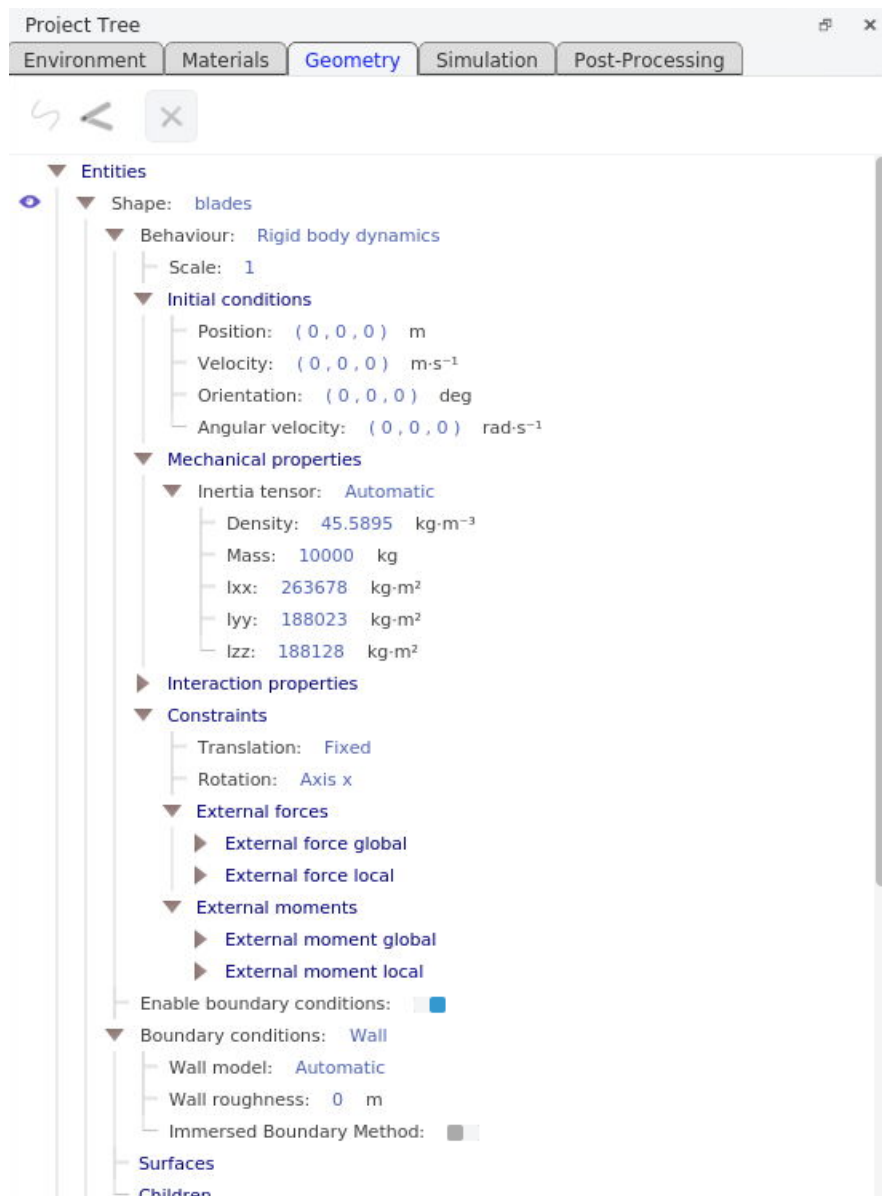
Start a new XFlow GUI, load the [Enforced case](#) and save as a new project (for instance `WindTurbine-RBD.xfp`), and proceed in the following modifications.

Now the rotor will be free to rotate due to the forces exerted by the wind.

(a) In **Geometry > Shape: blades** set the **Behaviour** to **Rigid body dynamics**, with:

- **Density:** **45.5894 kg·m⁻³** which corresponds to a mass of **10000 kg**
- **Translation (global):** **Fixed**, **Rotation (local):** **Axis x**

Step 4: Rigid body dynamics behaviour



⚠ Please note: The automatic inertia tensor calculation is done assuming a uniform mass distribution.

- (b) In **Simulation**, leave the **Simulation time** to **12 s**
- (c) In this case, a frequency of **10 Hz** will be sufficient
- (d) Remove all the probes by **right clicking** the **Probe #** and selecting **Remove probe**

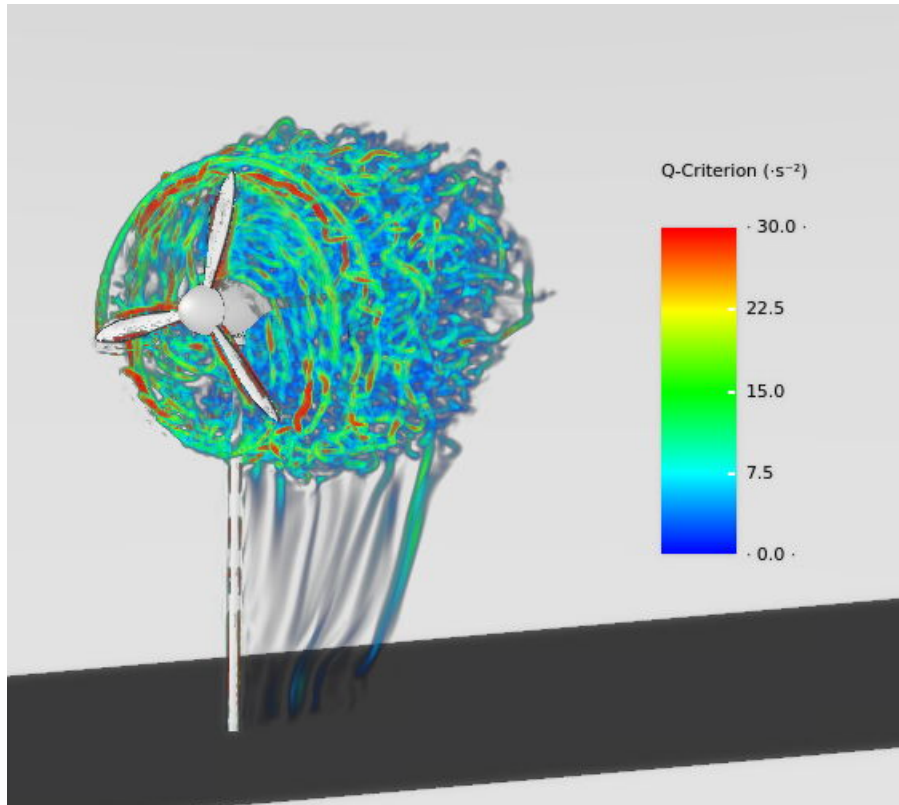
💡 Tip: You can directly load the setup of this problem from the project file `WindTurbine_RBD.xfp`

4.2 Post Processing

- (a) Play the simulation from the first frame to see how the blades start to rotate

(b) Choose the **Q-criterion** in **General > Show > Volumetric field > Visualisation field**, adjust its range to **[0, 30]** and show the Q-criterion field with **Transference law: a**

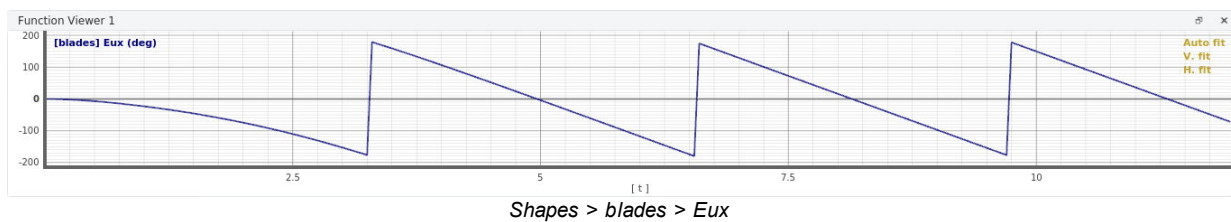
(c) Optionally, switch the **Interpolation mode: Convolution**. This will use an interpolation function to smooth the volumetric rendering.



3D Q-criterion field at $t = 8$ s

(c) Display in the *Function Viewer* the X-angular displacement of the blades

right click in the *Function Viewer* > **Shapes > blades > Eux**

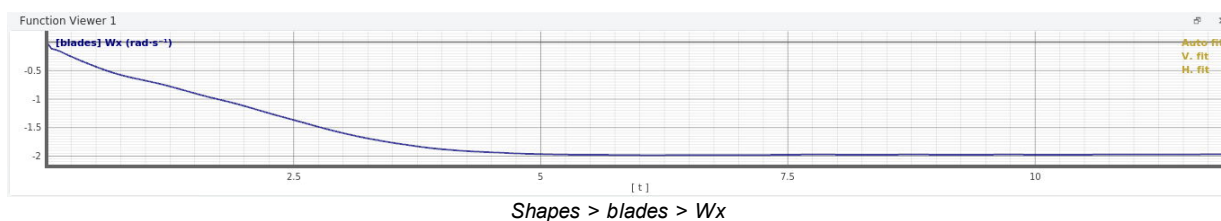


Shapes > blades > Eux

(d) Display in the *Function Viewer* the X-angular velocity of the blades

right click in the *Function Viewer* > **Shapes > blades > Wx**

Step 4: Rigid body dynamics behaviour



Observe that the blades achieve a uniform rotational speed of approximately $2 \text{ rad/s} = 19 \text{ rpm}$

Tutorial 08 - Heat transfer

This tutorial illustrates the setup and solution of a conjugate heat transfer problem, where both solid conduction and fluid convection are solved simultaneously.

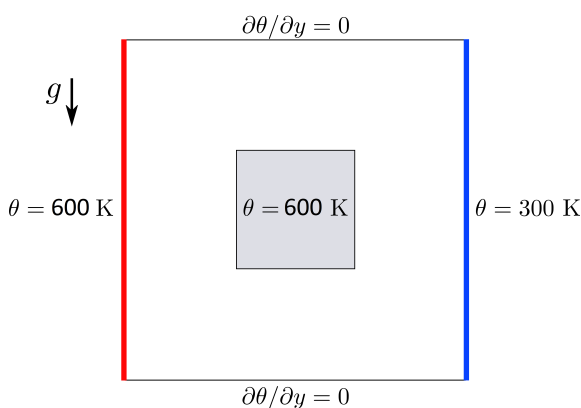
It consists of a square cavity with differentially heated vertical walls and adiabatic horizontal walls. The cavity is filled with air and a solid box is placed in its center. Fluid convection is driven by buoyancy forces, which are modeled using the Boussinesq approximation:

$$\rho = \rho_0 [1 - \alpha(\theta - \theta_0)]$$

where α is the thermal expansion coefficient, θ_0 a reference temperature and $\rho_0 = \rho(\theta_0)$.

The Rayleigh number of the flow is $Ra \sim 10^6$, leading to strong convection. At the initial time instant, the solid is at a temperature of 400 K and it will be cooled by the surrounding fluid.

For radiative heat transfer, see Tutorial 09.



This tutorial shows how to:

- Create simple geometries, such as boxes
- Apply boundary conditions to faces
- Use the single phase internal flow model
- Use the segregated energy thermal model
- Enable the viscous term in the energy equation
- Use the Boussinesq state equation
- Set conjugate heat transfer problems

- Compute particles in single phase flows
- Visualise heat flux and results on a line
- Export cutting plane data to raw format

It is assumed that the reader has completed Tutorial 01 and 02. Some steps in the setup and post-process will not be described in detail.

Contents

[Step 1: Problem setup](#)

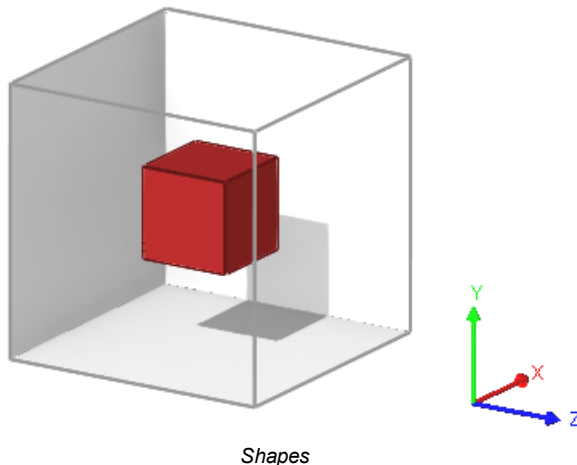
[Step 2: Post-processing](#)

Step 1: Problem setup

1.1 Create the geometry

- Create the cavity as a box () with lower corner **(-1.5, -1.5, -1.5)** and upper corner **(1.5, 1.5, 1.5)**.
Check with the help of the **Back-face culling** that the box normals are orientated towards the interior and rename this **Shape** as **Cavity**.
- Create the inner solid as a box with lower corner **(-0.5, -0.5, -0.5)** and upper corner **(0.5, 0.5, 0.5)**.
Check that the box normals are orientated to the exterior.

 **Remember:** Back-face culling helps to distinguish the orientation of the normals of a surface (see [Tutorial 02, step 1](#)). To reverse the orientation: Select the geometry > **Right click** in *Graphic View* > **Reverse orientation**



1.2 Engine settings

- (a) Switch to the **Expert** mode in **Main menu > Options > Preferences > Application > Application mode: Expert**, or  in Toolbar menu, and restart XFlow.
- (b) Select the **Kernel: 2d**
- (c) Select the **Flow model: Single phase**
- (d) Select the **Analysis type: Internal**
- (e) Select the **Thermal model: Segregated energy**. Leave **Disabled** the **Radiation model**, as radiation is not of significant in this example.
- (f) Leave the default **Turbulence settings**, **Acoustics analysis**, **Scalar transport** and **Advanced options**. Note that **Advanced options > Enable viscous term in energy equation** is enabled by default.

The effect of viscous dissipation in natural convection is appreciable only when the induced kinetic energy becomes comparable to the amount of heat transferred, and this occurs when either the buoyancy force is large or when the convection region is extensive. Viscous heat dissipation will not be relevant in this example. You may test both cases, with the viscous term enabled and disabled.

1.3 Environment settings

Keep the default settings except:

- (a) To model the buoyancy effects, set the gravity ($-9.81 \text{ m}\cdot\text{s}^{-2}$) as the **External acceleration laws** in **Y** direction.
- (b) Leave the **Initial velocity field** and **Volumetric heat source** in their default values.
- (c) Set the **Initial temperature field** to **350 K**

Step 1: Problem setup



Project Tree > Environment

1.4 Material settings

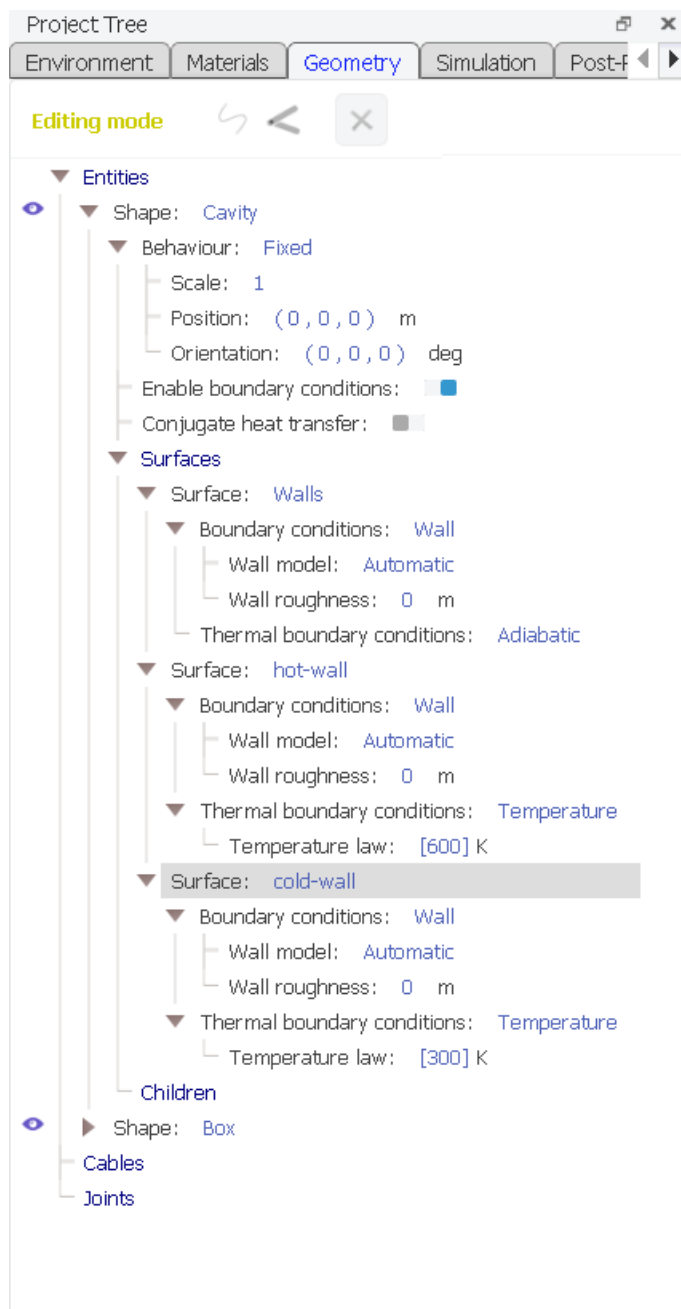
- Set the Material 1 **State equation** to **Boussinesq**, with **Density: $1 \text{ kg}\cdot\text{m}^{-3}$** and **Thermal expansion coefficient: 0.1 K^{-1}**
- Set the **Newtonian Dynamic viscosity** to **$1\text{e-5 Pa}\cdot\text{s}$** , the **Thermal conductivity** to **$10 \text{ W}\cdot(\text{m}\cdot\text{K})^{-1}$** and the **Specific heat capacity** to **$200 \text{ J}\cdot(\text{kg}\cdot\text{K})^{-1}$**

1.5 Geometry settings

It is required to apply different conditions to each face of the cavity. This is done in the following way:

- (a) Leave the **Enable boundary conditions** option switched **On** and **Conjugate heat transfer** switched **Off**
- (b) Choose the selection mode "Face filter"  in the toolbar . Select the -x vertical face of the **Shape: Cavity** (it highlights), **right click** on the *Graphic View* and choose **Apply boundary conditions to faces** from the contextual menu
- (c) In the **Geometry > Entities** section of the *Project Tree*, check that two surfaces appear in the **Cavity Shape**: "Surface 1" is the one you have selected and "Surface 0" contains the rest of the cavity surfaces
- (d) Select the opposite face, **right click** in the *Graphic View*, and choose **Apply boundary conditions to faces**
- (e) Check that a new surface called "Surface 2" appears in the *Project Tree*
- (f) Rename "Surface 0" to "walls", "Surface 1" to "hot-wall" and "surface 2" to "cold-wall"
- (g) Return to the View only selection mode by choosing  in the toolbar
- (h) Leave the **Conjugate heat transfer** option switched **Off**
- (i) Set all surfaces to **Wall Boundary conditions**, with **Automatic Wall model** and zero **Wall roughness**
- (j) Set the **Thermal boundary conditions** of the cavity's **hot-wall** to a prescribed **Temperature** of **600 K**, the **cold-wall** to a prescribed **Temperature** of **300 K** and the rest ("walls") to **Adiabatic**

Step 1: Problem setup

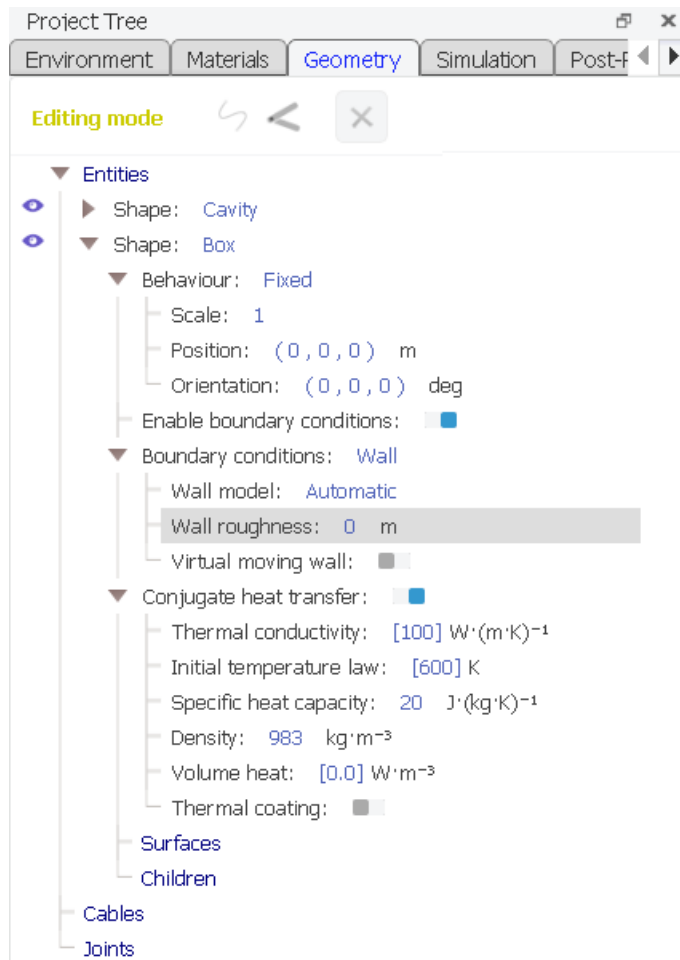


Project Tree > Geometry > Cavity

About the boundary conditions for the **Shape: Box**:

- (a) Leave the **Enable boundary conditions** option switched **On** and **Conjugate heat transfer** switched **Off**
- (b) Set the whole **Shape** to **Wall Boundary conditions**, with an **Automatic Wall model**, zero **Wall roughness** and leave the **Virtual moving wall** option switched **Off**
- (c) Switch **On** the **Conjugate heat transfer** option, with a **Thermal conductivity** of **100 W·(m·K)⁻¹**

¹, Initial temperature law: **600 K**, Specific heat capacity: **20 J·(kg·K)⁻¹**, Density: **983 kg·m⁻³** and Volume heat: **0 W·m⁻³**.



Project Tree > Geometry > Box

1.6 Simulation settings

- Set the **Simulation time** to **3 s** and keep **Courant** to **1**
- Set the **Resolved scale** to **0.05 m**
- Enable the **Refinement algorithm: Near static walls** with a **Target resolved scale** of **0.025 m** for all the shapes.
- Set the **Frame frequency** to **50 Hz** and leave the **Numerical data frequency** by default
- Enable **Compute markers** (passive particles that advect with the flow). They will be used just for post-processing purposes.

1.7 Launch the calculation

- Save the project

Step 1: Problem setup

Tip: You can directly load the setup of this problem from the project file `HeatTransfer.xfp`

- (b) Set the number of CPUs in **Main menu > Options > Preferences > Engine**
- (c) Press **Run** button > **Start computation**

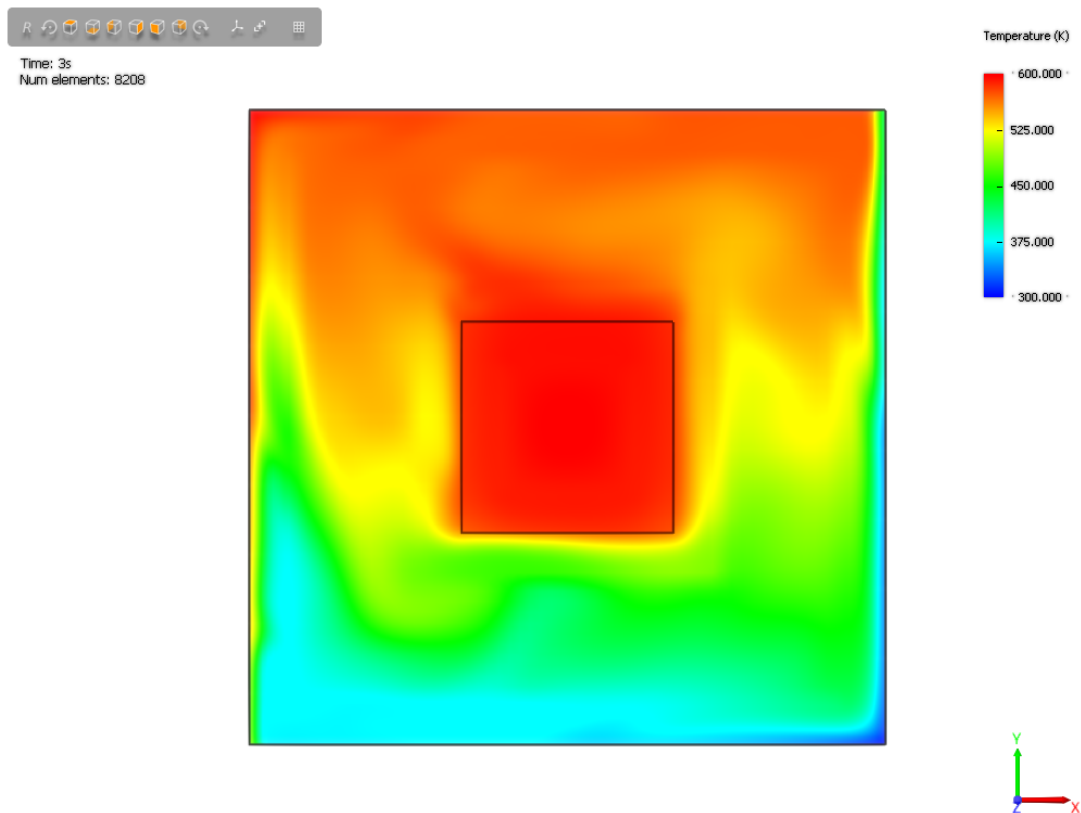
Step 2: Post-processing

Load the results data by: **Main menu > Simulation data > Load simulation data** or 

Set the **Post-Processing > Interpolation mode**: to Convolution.

2.1 Temperature field analysis

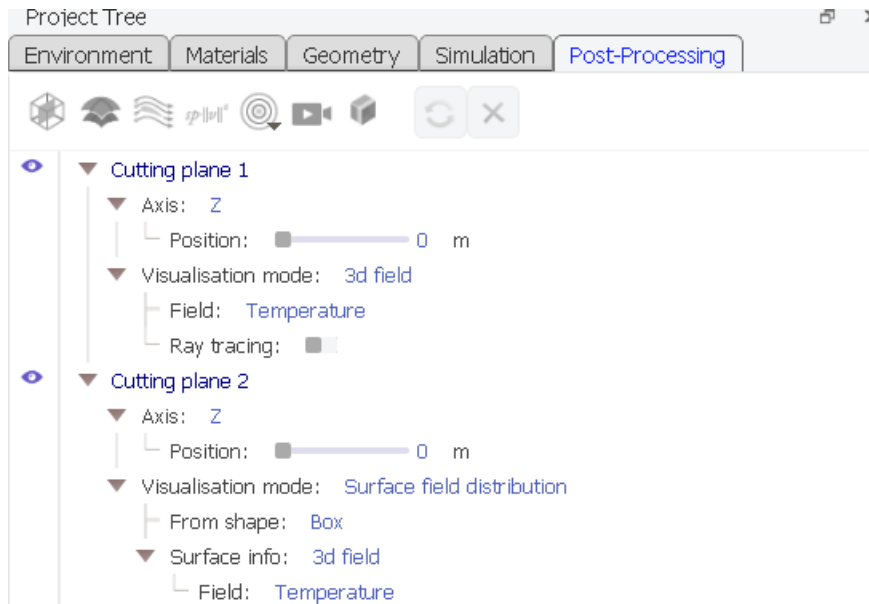
- (a) **Post-Processing > Cutting planes**. Create a cutting plane on the Z axis and visualise the **Temperature**
- (b) Adjust the the temperature range  to be **[300, 600] K**
- (c) Select the solid box and change its visualisation mode to Wireframe to be able to visualise the results inside it: Select the **Box** in **Graphic View > Right click > Visualisation mode > Wireframe**. Play forward the results and observe the influence of the solid's temperature in the overall solution.



Temperature field inside the Cavity and the Box at $t = 3$ s

- (d) Create a second cutting plane and display the temperature distribution along the Box walls by selecting **Post-Processing > Cutting planes > Cutting plane 2 > Visualisation mode:**

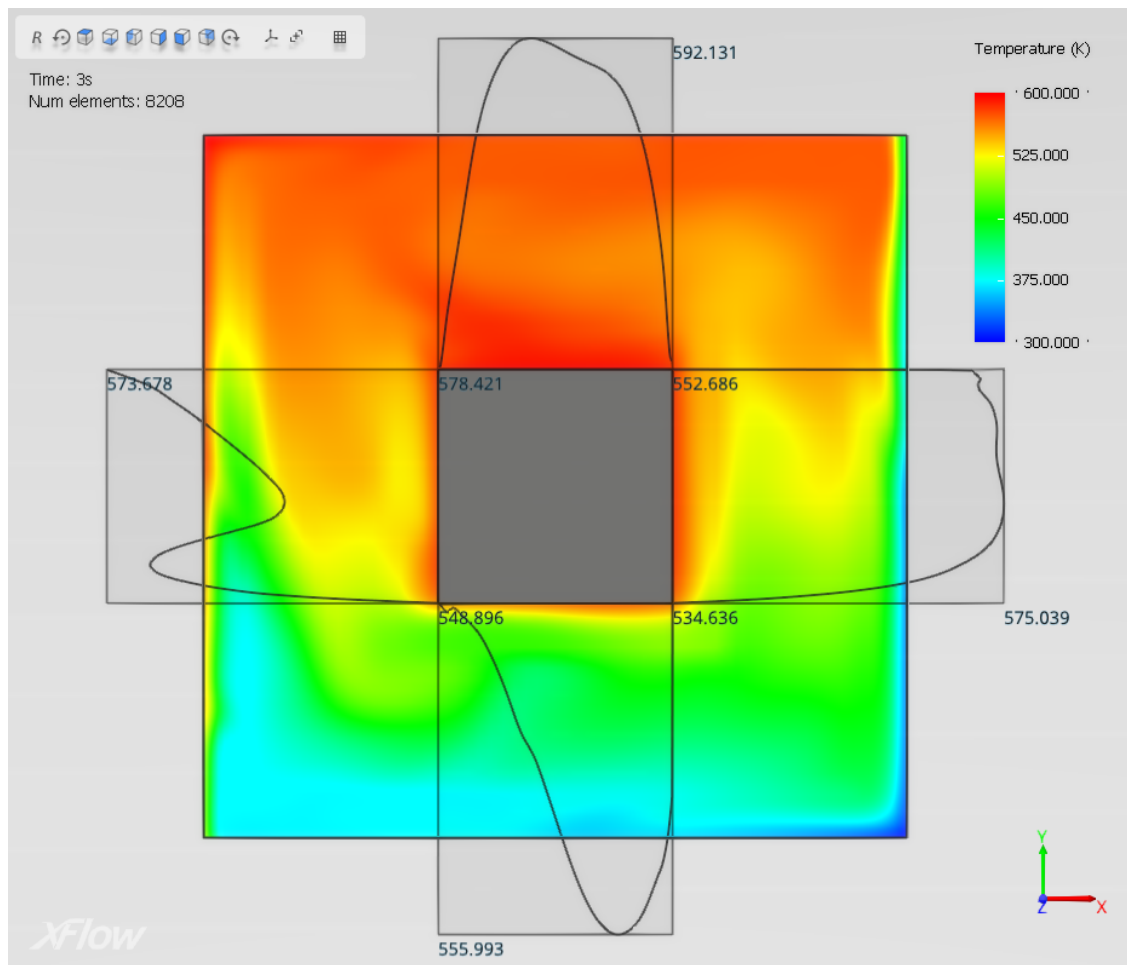
Surface field distribution



⚠ Please note: This is the temperature of the fluid in contact with the box walls, not the walls temperature.

You should see the superposition of the two cutting planes, showing the graphs of the temperature fields along the Box as well as the temperature contours inside the Cavity and the Box.

Step 2: Post-processing

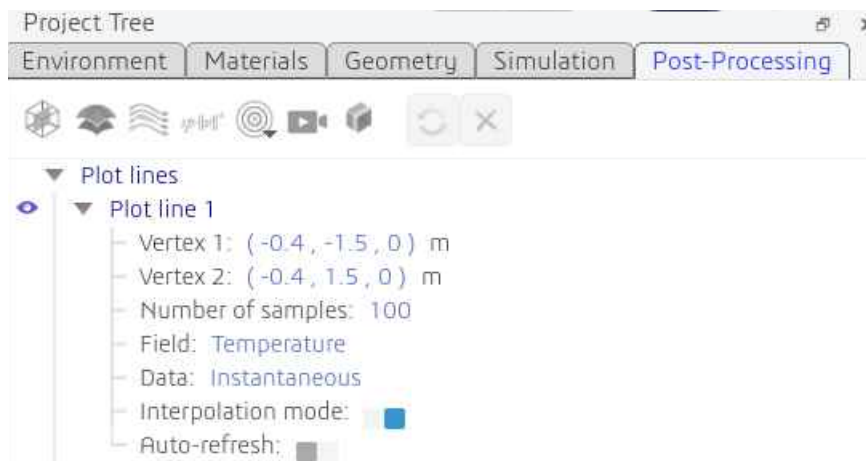


Temperature distribution on the Box geometry at frame $t = 3$ s

The figure above shows the temperature field and distribution along the Box surfaces at frame 150 ($t = 3$ s).

You can now switch off the temperature distribution by unticking the **Cutting plane 2** checkbox.

- (e) Go to frame 150 ($t = 3$ s) and create a line graph by **right clicking** on **Plot lines** and selecting **Add plot line**. Set **Vertex 1**: **(-0.4, -1.5, 0)** and **Vertex 2**: **(-0.4, 1.5, 0)**; observe its location on the *Graphic View 1*. Choose the **Temperature** as the **Field** to be visualised over the line.



- (f) To refresh the **Plot lines** just select the refresh icon  next to the plot line 1 entry or enable the **Auto-refresh** option available for each plot line. Then, in the *Function Viewer* **right click** and choose **Plot lines > Line 1** to display the temperature along the line. In the *Function Viewer* graph: X-coordinate is length from **Vertex 1** to **Vertex 2** and Y-coordinate is temperature. Click **Auto fit** and adjust the scale of the graph to [0, 3] in the X-coordinate and [300, 650] in the Y-coordinate. To do so, remember that:

To zoom: **Middle (wheel) mouse button** + drag or **scroll up/down**

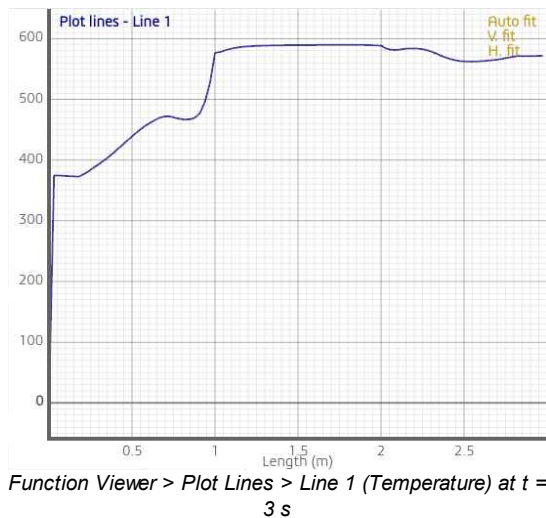
To pan: **Left mouse button** + drag

To adjust the X and Y axis:

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

You should see something as shown below:

Step 2: Post-processing



- (f) Export the data from the plot line to a file: **Right click** in *Function Viewer* > **Export current data**
- (e) Enable only the **Cutting plane 1** with the temperature field selected. Then export the temperature results at the cutting plane by **Main menu** > **Simulation data** > **Export cutting plane data to raw format**. This creates the file `currentCuttingPlane0.txt` in the project directory with the (non-interpolated) temperature value at the center of every cell. It contains four columns: position X, position Y, position Z and temperature value in K, and can be processed with spreadsheet software. Compare the data plot line values with those of the current cutting plane at $X = 0$.

2.2 Heat flux analysis

- (a) Disable the **Cutting plane 1** and show the heat flux at the walls **Post-Processing** > **General** > **Show** > **Surface info: Heat flux**
- (b) Change the visualisation mode of the Box to Shading: Select the **Box** in *Graphic View* > **Right click** > **Visualisation mode** > **Shading**
- (c) Note that the flux is constant at the top and bottom walls of the **Cavity**. Check that it is zero, as corresponds to an adiabatic wall.
- (d) Adjust the legend range to distinguish the heat flux variation along the box walls **Main menu** > **Simulation data** > **Analysis settings: Surface info** or  for example, min -10000 and max 0.
- (e) In the *Function Viewer*, display the evolution of the overall heat flux at the box. **Right click** in *Function Viewer* > **Shapes** > **Box** > **Heat**

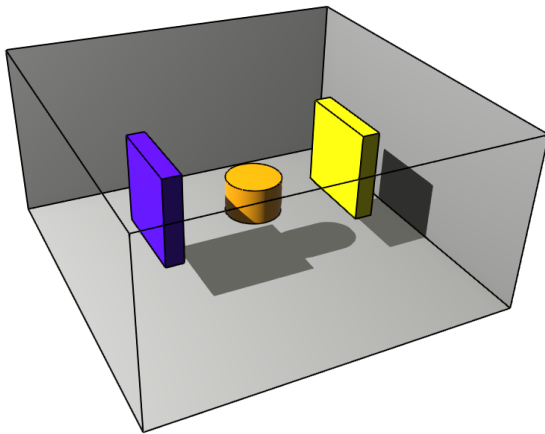
2.3 Velocity field analysis

- (a) Enable again the **Cutting plane 1**, choose the **Velocity** as **Visualisation field** and show it as particles in: **Post-Processing** > **Cutting planes** > **Cutting plane 1** > **Visualisation mode: Markers**

Play the results forward to see the movement of the particles. Observe the forced convection induces a recirculation of the flow next to the walls in the clockwise direction.

Tutorial 09 - Radiation

This tutorial illustrates the setup and solution of a radiation problem. It consists of a heat source at 400 K surrounded by two obstacles at 0 and 200 K and an outer wall at 0 K. The temperature of the wall behind the obstacles will be lower than the other part of the wall since the obstacles block the heat radiation energy. XFlow simulates the reflection and absorption of heat radiation energy at the surfaces.



This tutorial shows how to:

- Create simple geometries, such as a cylinder
- Use the Montecarlo radiation model
- Analyse the heat flux at the surfaces

It is assumed that the reader has completed Tutorial 01, 02 and 08. Some steps in the setup and post-process will not be described in detail.

Contents

[Step 1: Problem setup](#)

[Step 2: Post-processing](#)

Step 1: Problem setup

1.1 Create the geometry

- (a) Create the outer walls as a box (toolbar icon ) with lower corner **(-1, 0, -1)** and upper corner **(1, 1, 1)**. Check that the box is orientated to the interior and rename it as **OuterWall**.
- (b) Create the obstacles also as boxes. **Obstacle1** with lower corner **(-0.6, 0.25, -0.25)** and upper corner **(-0.5, 0.75, 0.25)**, **Obstacle2** defined by lower corner **(0.5, 0.25, -0.25)** and upper corner **(0.6, 0.75, 0.25)**. Check their orientation and rename the shapes.
- (c) Create the heat source as a cylinder (toolbar icon ) with centre at **(0, 0.4, 0)**, direction **(0, 1, 0)**, radius **0.15 m** and height **0.2 m**. The rest of settings can be left as default. Notice that the cylinder is closed, it has lids.
- (d) Check that the orientation of the cylinder is to outside and rename it as **HeatSource**. Remember that the culling helps to distinguish the orientation of the normals of a surface. The normals have to point always to the fluid region, see figure in the [previous page](#). To reverse the orientation (if required): Select the geometry > **Right click** in *Graphic View* > **Reverse orientation**

1.2 Engine settings

- (a) Select the **3d Kernel**
- (b) Select the **Flow model: Single phase**
- (c) Select the **Analysis type: Internal**
- (d) Select the **Thermal model Segregated energy**
- (e) Select the **Radiation: Montecarlo**, with **Ray density: 200 ray·m⁻²** and **Number of iterations: 1**
- (f) Leave the default **Turbulence settings**.

1.3 Environment settings

Keep the default settings except the **Initial temperature field**, which will be set to **250 K**.

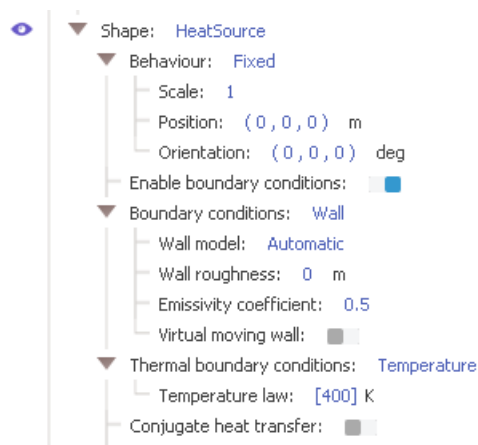
1.4 Material settings

Keep the **Material 1** settings to their default values.

1.5 Geometry settings

- (a) Set **HeatSource** to **Fixed Behaviour**, **Wall Boundary conditions** with **Automatic Wall model**, **Emissivity coefficient: 0.5**, **Thermal boundary conditions** with prescribed **Temperature** at **400 K** and the **Conjugate heat transfer** option to **Off**.

Step 1: Problem setup



- (b) Set **OuterWall** and **Obstacle1** to the same boundary conditions than the **HeatSource** but prescribed temperature to **0 K**
- (c) Set **Obstacle2** to the same boundary conditions than the **HeatSource** but prescribed temperature to **200 K**

1.6 Simulation settings

- (a) Set the **Simulation time** to **0.0002 s** (it is enough to calculate just few time steps) and **Time step mode**: **Fixed automatic** with **Courant** to **1**
- (b) Set the **Resolved scale** to **0.04 m**, with the **Refinement algorithm**: **Near static walls** and **Target resolved scale**: **0.02 m** for all the shapes.
- (c) Set the **Frames frequency** to **10000** frames per second and leave the **Numerical data frequency** by default
- (d) Leave the computation of averaged fields and markers disabled

1.7 Launch the calculation

- (a) Save the project

 **Tip:** You can directly load the setup of this problem from the project file `Radiation.xfp`

- (b) Set the number of CPUs in **Main menu > Options > Preferences > Engine**
- (c) Press **Run** button > **Start computation**

Step 2: Post-processing

In this problem the fluid is static (no forced flow nor natural convection) because the goal is to analyse the capabilities of the radiation solver in XFlow.

2.1 Read the Message View output

When you launch the simulation, XFlow discretises the domain (in this example, the full domain has 345332 elements in two levels of refinement), estimates the time step (= 0.00001 seconds at level 0) and generates the irradiance map to calculate the heat flux at each point. Due to the facts that heat source has constant temperature along time and the gas is non-absorbing, the irradiance map is calculated only once and the radiative heat flux remains constant. Therefore, in this case it is enough to calculate just one time step.

```

Message View:
-----
--
Full domain has 345332 elements.
Equivalent single-resolution domain has 584064
Equivalent single-resolution domain size is ( 104 x 54 x 104 )
-----
--

Computing boundary conditions map!
-----
--

Coarsest resolved length: 0.04
-----
--

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.
No reference pressure defined in the simulation setup.
Prandtl number: 0.741114
Thermodynamic speed of sound: 340.112
Reference area: 1.99707 m^2
Reference velocity: 1 m/s
Time step (level 0): 0.0001 s
-----
--

Generating irradiance map...
Iteration: 1 / 1
Processing Level: 1 / 2
Processing Level: 2 / 2
Progress: 0%
Progress: 3%
Progress: 7%
Progress: 10%
Progress: 14%
Progress: 17%
Progress: 21%
Progress: 28%
Progress: 32%
Progress: 35%
Progress: 39%
Progress: 42%

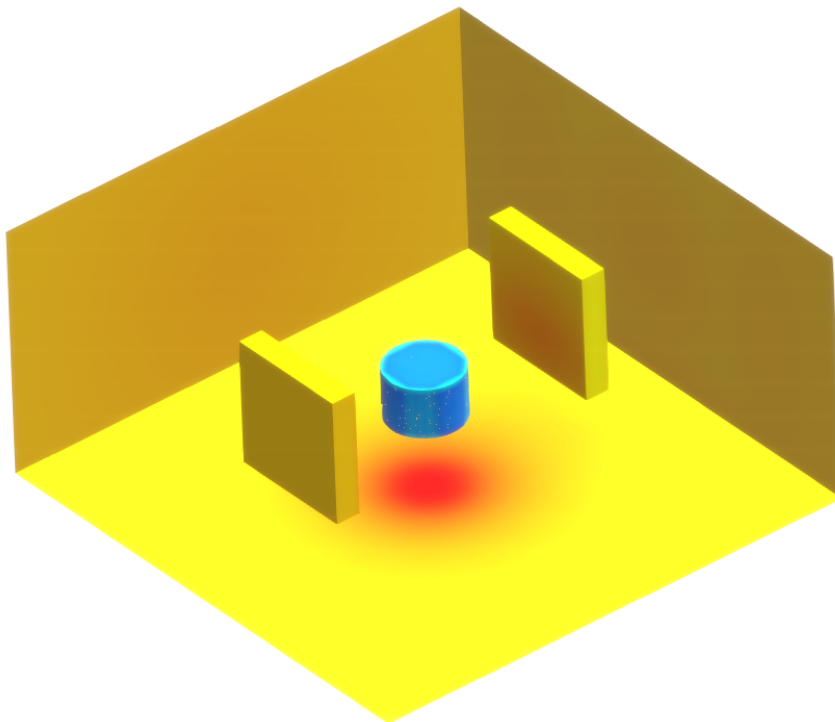
```

Step 2: Post-processing

```
Progress: 46%
Progress: 53%
Progress: 57%
Progress: 60%
Progress: 64%
Progress: 67%
Progress: 71%
Progress: 78%
Progress: 82%
Progress: 85%
Progress: 89%
Progress: 92%
Progress: 96%
Irradiance map generated in 54 minutes 58 seconds 612 milliseconds
Saving data...
```

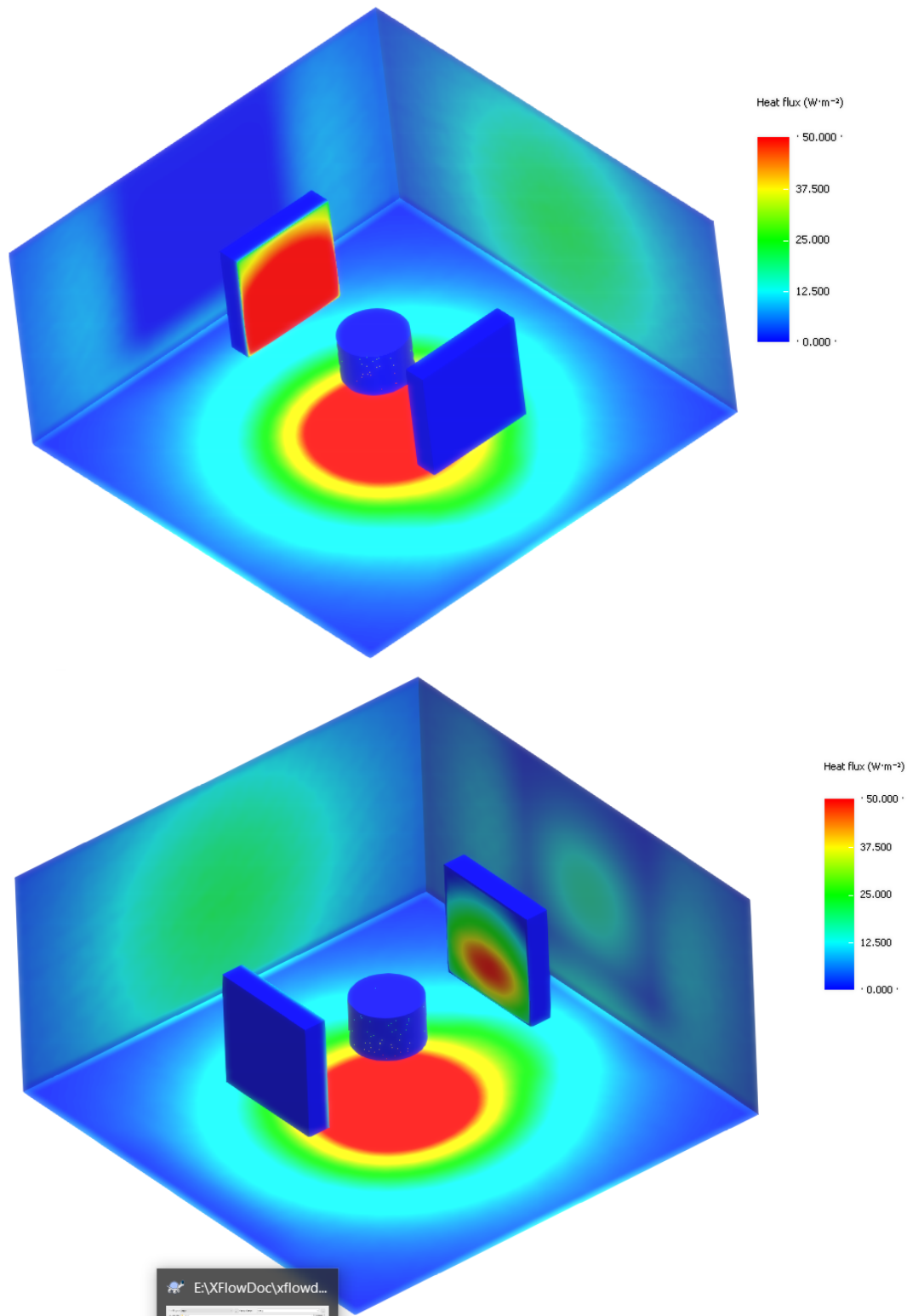
2.2 Analyse the heat flux at the surfaces

- (a) Go to frame 0 and show the total heat flux at the surfaces: **Post-Processing** > **General** > **Show** > **Surface info: Heat flux**



The heat flux with **Obstacle2** should be smaller because it is at a higher temperature (200 K).

- (b) Now set the **Surface info range** to **[0, 50]** and observe the solution at the walls. Notice that on the surface of the **OuterWall** behind **Obstacle1**, the heat flux is smaller than behind **Obstacle2**.



Heat flux at frame 0 showing pure radiative heat transfer

Heat flux at frame 0 is purely radiative, while in frames 1 and 2 the heat flux has also contributions from

Step 2: Post-processing

the energy equation (diffusion + convection).

The farther from the source and with less visibility, the noisier is the solution. This is due to the stochastic nature of the Montecarlo model. Increase the ray density and/or number of iterations and observe the improvement of the solution.

If you analyse the temperature at the surfaces (choose **Show > Surface info: 3d field** with **Field: Temperature**), in frame 0 the temperature is constant to 250 K on all surfaces. Remember that this is the temperature of the fluid touching the walls, not the temperature of the solid objects. Solids must have conjugate heat transfer thermal boundary conditions (see [Tutorial 08](#)) to be able to display their temperature.

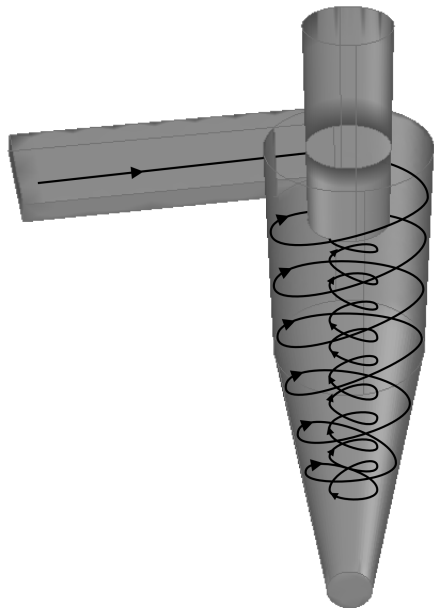
Refine the resolution to achieve better results, remember to adjust the simulation time depending on the time step size.

Tutorial 10 - Cyclone flow

This tutorial illustrates the setup and post-processing of a cyclone flow. Cyclones are used in many industrial applications, specially in separation processes such as dust from a gas stream.

Dust particles can be modeled in XFlow with the Discrete Phase Model (DPM), which represents solid spherical particles with physical properties (inertia, drag and gravity).

The inflowing fluid rotates inside the chamber and is constrained to follow a swirling flow path. In cyclones of this type the larger suspended particles move outwards to the chamber wall where they travel in a downward spiral to the base. The smaller particles move slowly and therefore their distribution across the flow changes little. Those in the center are captured in the upward flow and exit the chamber through the upper outlet. Cyclones are commonly used when the density of the inflowing fluid (the carrier phase) is less than that of the suspended phase.



This tutorial shows how to:

- Calculate passive stream tracers
- Calculate Discrete Phase Model stream tracers
- Show the path lines
- Monitor the number of particles through the outlets

It is assumed that the reader has completed Tutorial 01, 02 and 03. Some steps in the setup will not be

described in detail.

Contents

[Step 1: Problem setup](#)

[Step 2: Post-process](#)

[Step 3: Stream tracers](#)

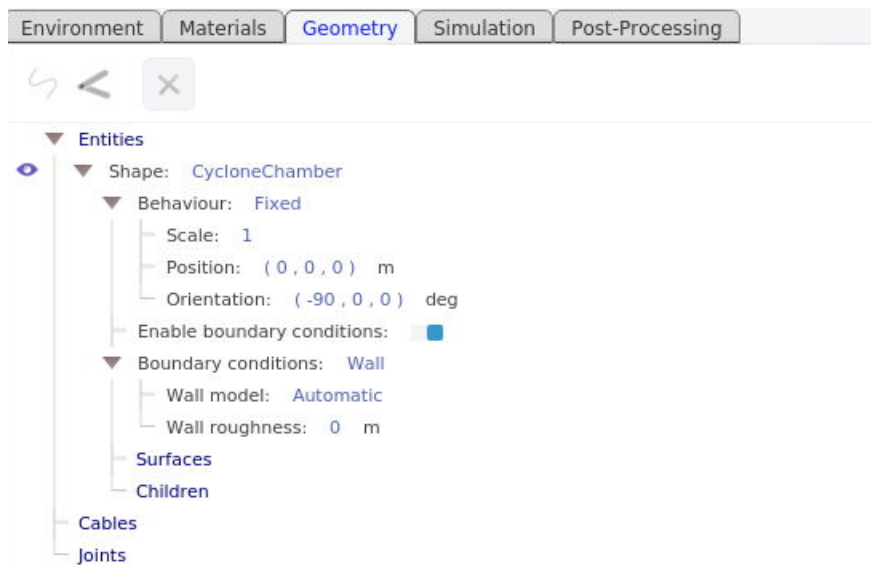
Step 1: Problem setup

1.1 Engine settings

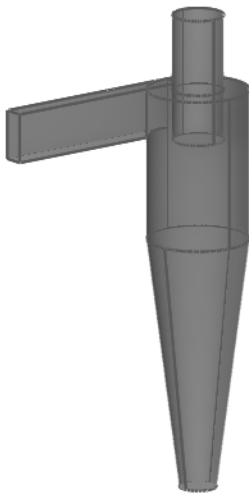
- (a) Select the **3d Kernel**
- (b) Select the **Flow model Single phase**
- (c) Select the **Analysis type: Internal**. Now the wind tunnel will disappear and you will have to provide the geometry of the chamber that will contain the fluid
- (d) Leave the default options for **Thermal model** and **Turbulence settings**

1.2 Import the geometry

- (a) Import the geometry of the chamber from the file `CycloneChamber.stp` (**Main menu > Geometry > Import a new geometry** or ). Note that the vertical direction of the chamber, when it is imported, is in the Z-axis.
- (b) Rotate the chamber so that the vertical direction is parallel to Y-axis: **Geometry > Entities > Shape: CycloneChamber > Orientation: (-90, 0, 0) deg**



- (c) Check the orientation of the normals and make sure they are pointing inside the geometry, i.e. to the region with fluid. If required, select the geometry and do **right click** in *Graphic View* > **Reverse orientation**. The geometry should look as following:



*Shape-CycloneChamber with correct normals
orientation: pointing inside*

1.3 Environment settings

- Keep the **External acceleration laws** and **User defined Initial conditions** (velocity field and gauge pressure field) to zero
- Keep the **Reference area** to **Front** and **Reference velocity** to **Automatic**

1.4 Material settings

Step 1: Problem setup

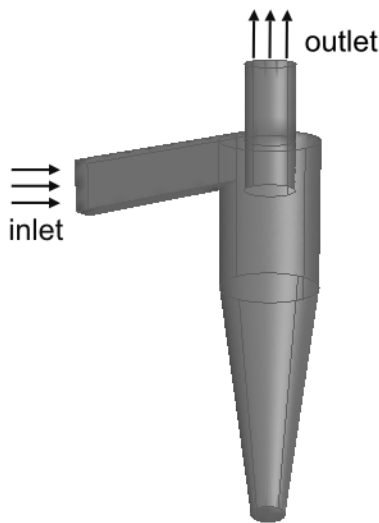
Leave the default **Material 1** properties which is air.

1.5 Geometry settings

(a) Set the **CycloneChamber Behaviour** as **Fixed**

(b) Set the inlet and outlet boundary conditions:

- Choose the "Face filter" mode  in the toolbar.
- Select the inlet surface as indicated the picture below and **right clicking** on the *Graphic View*, choose **Apply boundary conditions to faces** from the contextual menu



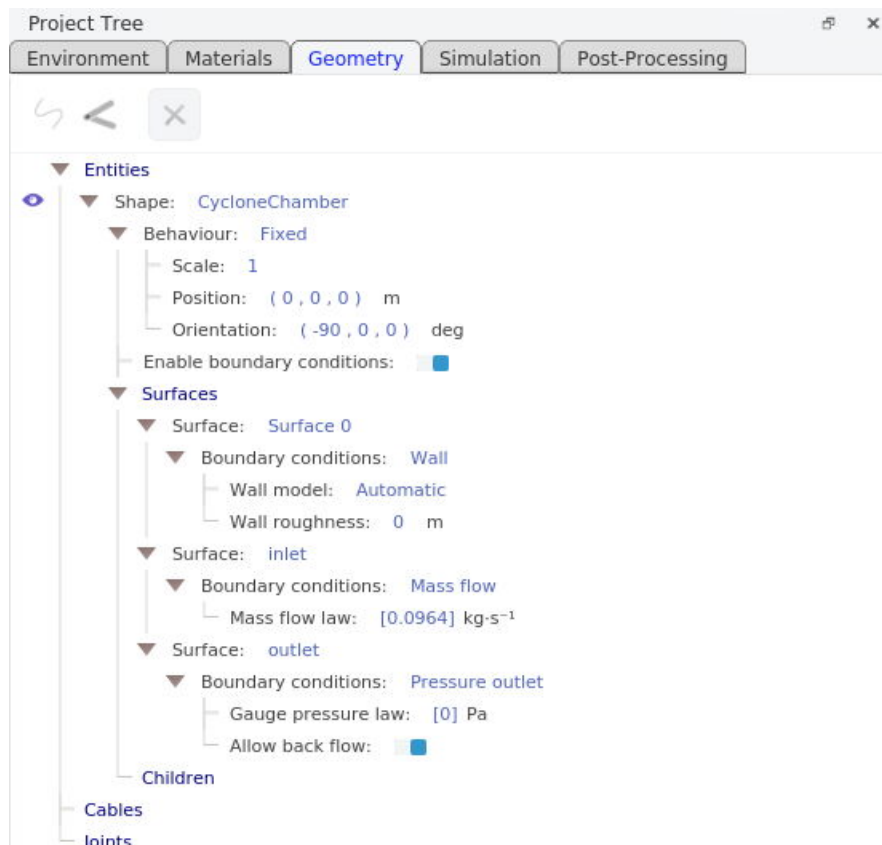
Boundaries schematic

- Check that in the **Geometry** section of the *Project Tree*, two surfaces appear in the **CycloneChamber** shape: "Surface 1" is the one you have selected and "Surface 0" contains the rest of the chamber surfaces
- Select the outlet surface, **right click** in the *Graphic View*, and choose **Apply boundary conditions to faces**
- Check that a new surface called "Surface 2" appears in the *Project Tree*
- Rename "Surface 0" to "**chamber**", "Surface 1" to "**inlet**" and "Surface 2" to "**outlet**"
- Return to the "View only" mode by choosing  in the toolbar

(c) Leave the **Wall Boundary condition** at the **CycloneChamber** with an **Automatic Wall model**

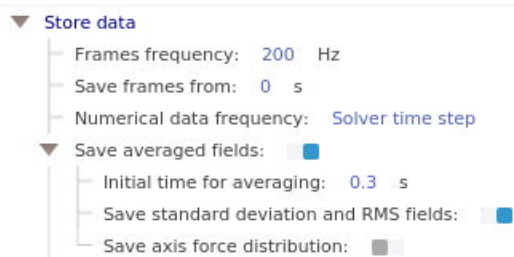
(d) Set the **Boundary condition > Inlet > Mass flow** to the **inlet** surface. It will be assumed that the air volumetric inflow rate is $0.08 \text{ m}^3 \cdot \text{s}^{-1}$. This is equivalent to a **Mass flow law** of (air density)* (volumetric flow rate) = **$0.0964 \text{ kg} \cdot \text{s}^{-1}$**

(e) Set the **Pressure outlet** to the **outlet** surface, with **0 Pa**. Enable the **Allow back flow** option



1.6 Simulation settings

- Set the **Simulation time** to **1 s**, **Time step mode**: **Fixed automatic** and **Courant** to **1**
- Set the **Resolved scale** to **0.005 m**, with the **Refinement algorithm**: **Disabled**
- Set the **Frames frequency** to store data to **200** frames per second
- Switch on **Save averaged fields** with **Initial time for averaging**: **0.3 s**. This will filter out the transient period of the flow before starting the averaging.
- Set **Save standard deviation and RMS fields**: **On**



1.7 Launch the calculation

- Save the project

Step 1: Problem setup

 **Tip:** You can directly load the setup of this problem from the project file `CycloneChamber.xfp`

- (b) Set the number of CPUs in **Main menu > Options > Preferences > Engine**
- (c) Press **Run** button > **Start computation**

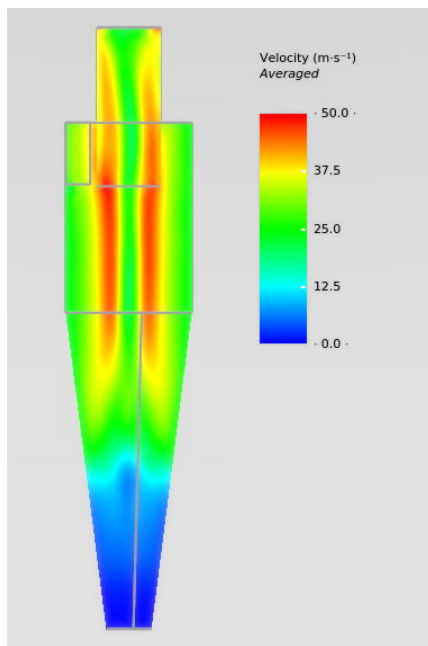
 **Please note:** Depending on your hardware, the computation of this tutorial may take a few hours and around 5 GB of hard disk.

Step 2: Post-processing

2.1 Visualize the velocity field on a cutting plane

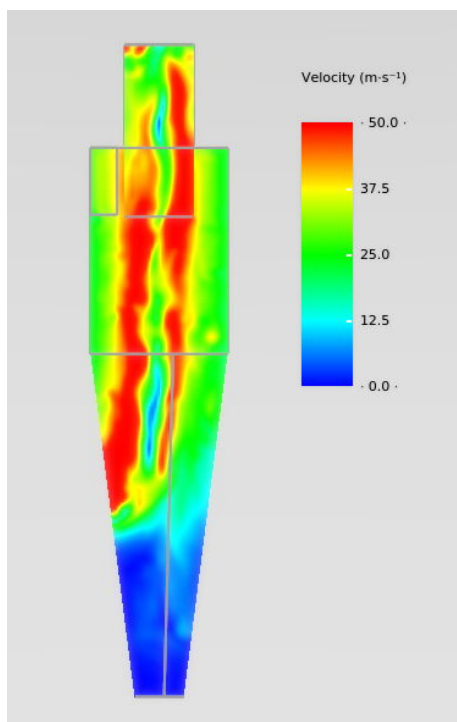
In **Post Processing > Cutting planes**

- (a) Create a **Cutting plane 1** on the Z axis and adjust its **Position** to **0.0037**
- (b) Choose **Visualisation mode: 3d field**
- (c) Choose **Field: Velocity**
- (d) Set the **General > Interpolation mode: Convolution**
- (e) Select view from right 
- (f) Choose **General > Data: Averaged**
- (g) Adjust the legend range of velocity to **[0, 50] m·s⁻¹** (**Main menu > Simulation data > Analysis settings** or )
- (h) Select the geometry and visualise it in wireframe mode (**Graphic View Menu > Visualisation Mode > Wireframe**).
- (i) Press play  and note that there is no data before $t = 0.3$ s as we specified for the averaging initial time



Averaged velocity field on cutting plane 1 at $t = 1 \text{ s}$

- (h) Change to **Instantaneous** data and play forward the results. Observe the differences with the averaged data for the same frame.



Instantaneous velocity field on cutting plane 1 at $t = 1 \text{ s}$

- (i) Show also static pressure, vorticity, V_y , V_z and turbulence intensity fields. For each field, switch

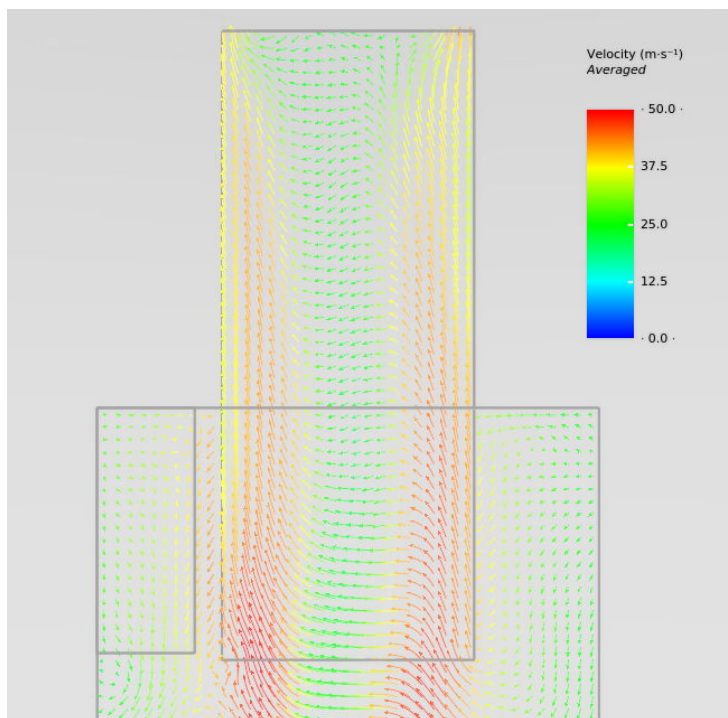
Step 2: Post-processing

between Instantaneous and Averaged, Standard deviation and RMS data in order to check the difference. You may need to adjust the legend ranges.

2.2 Visualize the velocity vector field on a cutting plane

In **Post-Processing > Cutting planes > Cutting plane 1**

- (a) Go to the last frame available ►►
- (c) Choose **Visualisation mode: Vectors** and increase arrows density and arrows length
- (d) Choose **Field: Velocity**
- (e) Select view from right 
- (f) Change between Instantaneous and Averaged and refresh the cutting plane with the icon 



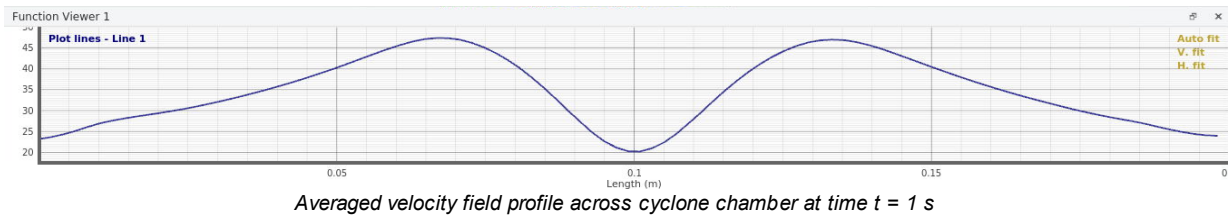
Averaged velocity vectors field

2.3 Measure velocity field along a line

- (a) Create a plot line **right clicking** on **Post-Processing > Plot lines** and selecting **Add plot line**
- (b) Line goes from **Vertex 1: (-0.1, -0.3, 0)** to **Vertex 2: (0.1, -0.3, 0)**
- (c) Set the **Number of samples** to **1000**. This is the number of points used to draw the curve, therefore the higher is this number, the smoother is the curve.
- (d) Select the **Velocity** field and **Averaged** data, with interpolation **On**

(e) Add a function viewer using the icon 

(f) **Right click** in *Function Viewer* > **Plot lines** > **Line 1** to display the evolution of the averaged velocity field along the line:



2.4 Visualize the vorticity volumetric field

(a) Deactivate any cutting plane or plot line

(b) Set the **Interpolation mode** to **Off**

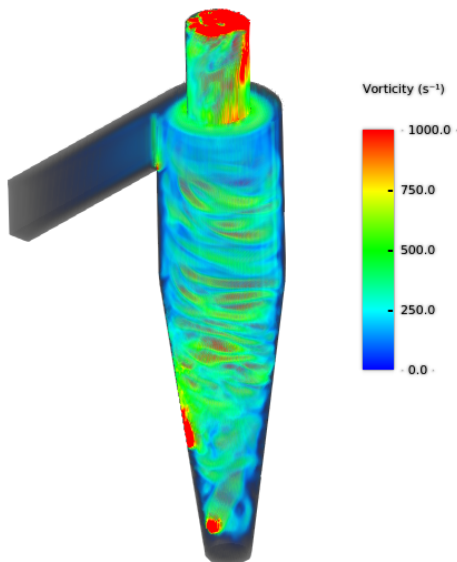
(d) Set the vorticity range to **[0, 1000]** (**Main menu** > **Simulation data** > **Analysis settings** or )

(e) Set **Post-Processing** > **General** > **Data: Instantaneous**

(f) Enable the visualisation of the **Volumetric field** and visualise the **Vorticity** field with a transference law: **a*a**

(g) Press play 

(h) Change the **Transference law** to **a**. You will see that the low vorticity values are more visible now.



Volumetric rendering of instantaneous vorticity at $t = 1$ s with a transference law $[a]$

Step 3: Stream tracers

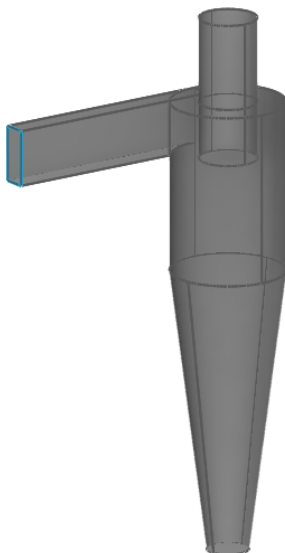
Stream tracers integrate the streamlines in the velocity field and so allow to track the trajectory of Lagrangian particles along the flow.

The particles can be massless (passive) particles advected by the fluid, or solid spherical particles (discrete phase model) with physical properties (inertia, drag and gravity).

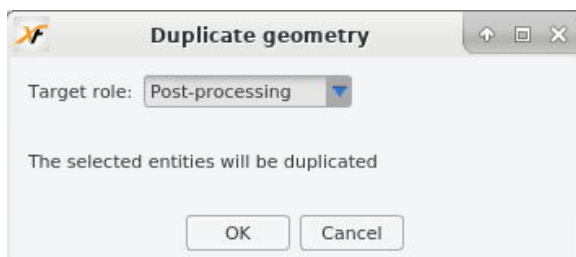
3.1 Create the surface source of the tracers

The source of stream tracers is a geometry **Shape**, usually a line or plane. We will create a surface similar to the inlet for the source.

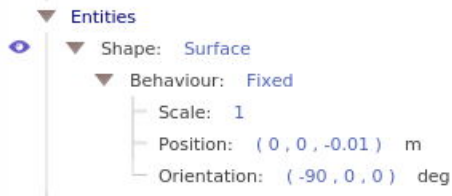
- (a) Choose "Face filter" mode  in the toolbar and select the surface corresponding to the inlet
- (b) Duplicate the inlet surface with **Main menu > Geometry > Duplicate**



- (c) In the pop-up window, leave the default options and press **OK**. This option will generate a new surface (called "CycloneChamber-Duplicated") directly in the post-processing mode that appears as a **Shape** in the **Post-Processing > Entities > Shape** section of the *Project Tree*. Therefore this new entity will not affect the flowfield but will only serve as a post-processing geometry.



- (d) Rename the CycloneChamber-Duplicated to "**Surface**", move it **0.01 m** to the interior (-Z direction) and rotate it in X axis 90 degrees **(-90, 0, 0)**.



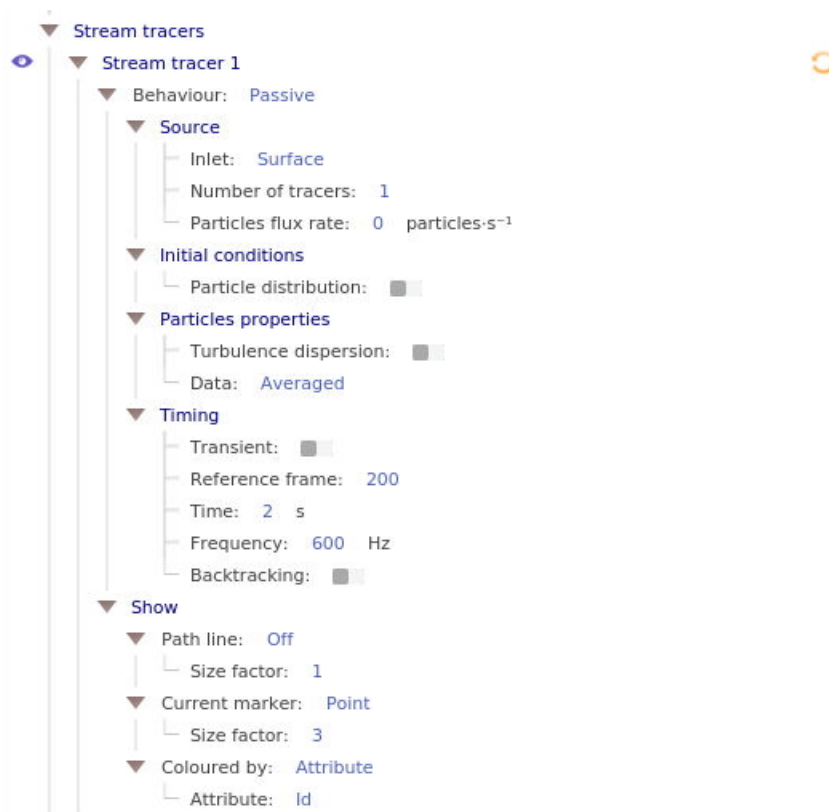
- (f) Return to the "View only" selection mode by choosing  in the toolbar

3.2 Passive stream tracers

3.2.1 Steady tracer

- (a) Create a stream tracer field by right clicking the string **Stream tracers > Add stream tracer**. The new stream tracer is automatically called **Stream tracer 1**.
- (b) Select **Behavior Passive**
- (c) In **Source > Inlet**, select the geometry shape that is going to be the source of tracers, in our case **Inlet: Surface**
- (d) Set the other parameters to:
- Number of tracers: 1** (Only one particle)
 - Particles flux rate: 0 particles·s⁻¹**
 - Particles distribution: Off**
 - Turbulent dispersion: Off**
 - Data: Averaged**
 - Transient: Off**
 - Reference frame: 200**
 - Time: 1.5 s**
 - Frequency: 600 Hz**
 - Backtracking: Off**

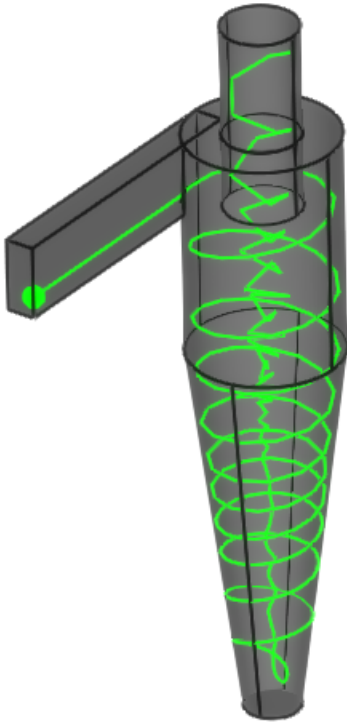
Step 3: Stream tracers



- (d) **Right click** on **Stream tracer 1** and select 

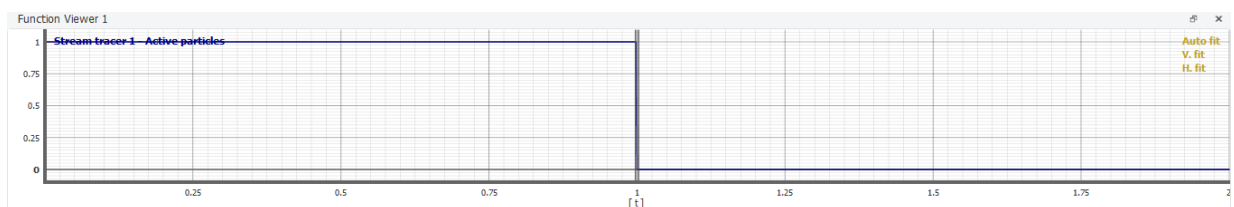
Observe that a second timeline is shown (in red colour) related to the stream tracers, and that the frame 200 is locked. To unlock the visualization of the Stream tracer 1, you must disable the visibility icon.

- (e) Choose **Stream tracer > Show > Current marker: Point** with size factor **3** and press play  to see the motion of the passive particle.
- (f) Once the stream tracers have been computed, it is possible to generate the path lines of the stream tracers to see the associated streamlines. Set **Path lines > From frame: 0** and **Path lines > To frame: 1200** and refresh the **Path lines** by the refresh icon  or by **Right clicking** on **Stream tracer 1** and **Generate path lines**
- (g) When the generation finishes, display the line **Post-Processing > Stream Tracers > Tracer 1 > Show > Path line: Line**. You can also choose Ribbon, Sphere chain or Tube. You may need to adjust the size factor to visualize these path lines correctly.



You can observe the streamline of one particle which is captured by the inner vortex and exits the cyclone.

- (h) **Right click** on the *Function Viewer* and select **Stream tracers > Tracer 1 > Active particles** and refresh the **Tracer 1**. The function viewer will show the number of particles in the domain as a function of time. In this case, there is only one particle in the domain



Function Viewer > Stream tracers > Tracer 1 > Active particles

- (i) Untick the checkbox of **Stream tracer 1** to disable its visualisation and unlock the timeline

3.2.2 Transient tracer

- (a) Create a new stream tracer (right click **Stream tracers > Add stream tracer**). It is called **Stream tracer 2**.

- (b) Set the same parameters than before except

Data: Instantaneous

Transient: On

Initial frame: 0

Step 3: Stream tracers

Last frame: 50

In transient tracers it makes more sense to use instantaneous data.

(c) **Right click** on **Stream tracer 2** and select **Recompute**. Now the computation of the stream tracer integrates the instantaneous velocity vector field from frame 0 to 50.

(d) Again **right click** on **Stream tracer 2** and select **Generate path lines**. From frame: **0** and To frame: **50**

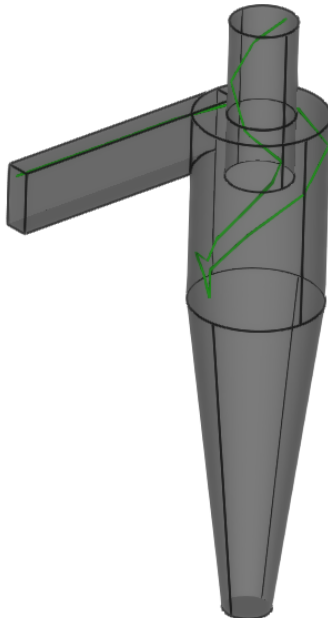
(e) Show the line and the marker

Post-Processing > Stream tracers > Stream tracer 2 > Show > Path line: **Tube**

Post-Processing > Stream tracers > Stream tracer 2 > Show > Size factor: **0.1**

Post-Processing > Stream tracers > Stream tracer 2 > Show > Current marker: **Sphere**

Post-Processing > Stream tracers > Stream tracer 2 > Show > Size factor: **0.1**



Observe that now the second timeline is placed on top of the simulation timeline and that the path line is rougher due to the fact that the computation uses less data than before. The frames frequency is 200 Hz against a frequency of 600 Hz in section 3.2.1.

(f) Press play  to see the motion of the passive particle

3.3 Discrete Phase Model

DPM models particles with physical properties such as density and diameter.

For stream tracers with more than one particle, it is possible to specify a standard deviation in the diameter size.

Normal and Tangential velocity restitution values refer to the collision of particles with the surfaces. Restitutions equal to 1 means perfectly elastic collision, while 0 means perfectly inelastic. There is no particle-particle collision.

Turbulent dispersion can be taken into account only when using averaged data.

In the initial velocity and acceleration, laws it is possible to use the following variables related to the particle and to the fluid fields:

particle_x, particle_y, particle_z	
(particle position)	fluid_px, fluid_py, fluid_pz (fluid position)
particle_vx, particle_vy, particle_vz	
(particle velocity)	fluid_vx, fluid_vy, fluid_vz (fluid velocity)
particle_t (particle time)	fluid_sp (fluid static pressure)
	fluid_vrt (fluid vorticity)

3.3.1 Big particle

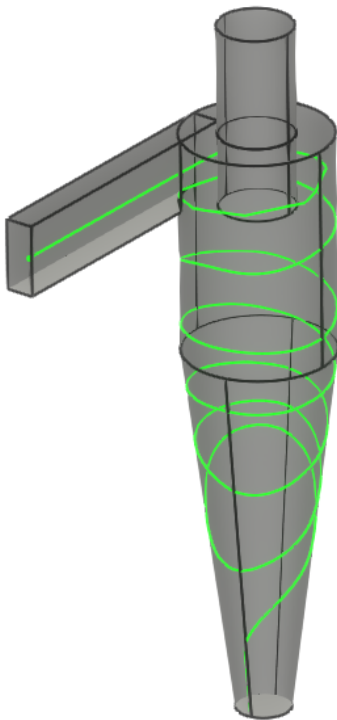
Let us model a big particle of density **1000 kg·m⁻³** and **1 mm** of diameter, initialised with the fluid velocities and subject to gravity forces.

- Create a new stream tracer (right click **Stream tracers** > **Add stream tracer**). It is called **Stream tracer 3**.
- Set the following parameters

Step 3: Stream tracers



- (c) **Right click** on **Stream tracer 3** and select **Recompute**
- (d) Again **right click** on **Stream tracer 3** and select **Generate path lines** from frame 0 to 1600.
- (e) Show the line and the marker
- (f) Press play  to see the motion of the passive particle. Now the DPM particle moves downward to the chamber base.

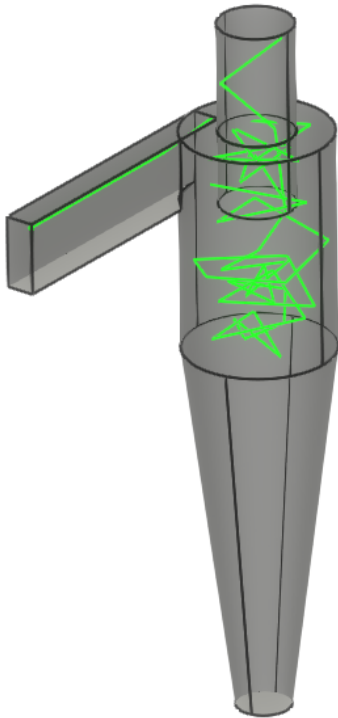


DPM with diameter = $1e-3$ m

3.3.2 Other cases

Taking the previous case as basis:

- (a) Vary the **Diameter** of the particle to **$1e-7$ m** and compute tracer and path line. Observe that now the particle does not reach the bottom.

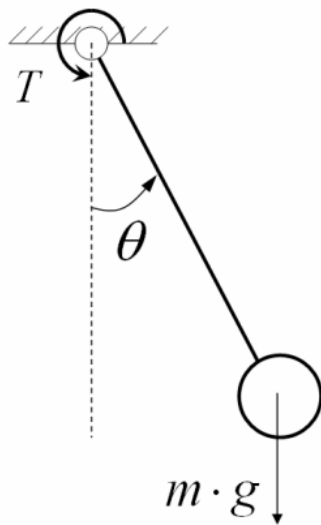


DPM with diameter = $1e-7$ m

- (b) Change the **Normal velocity restitution** : **1**, the **Tangential velocity restitution** : **1** and then recompute. The collision is perfectly elastic. If the particle happens to be caught in the inner cyclone, it will exit the chamber by the outlet.
- (c) Change the **Normal velocity restitution** : **0**, the **Tangential velocity restitution** : **0**, increase the **Frequency** to **500 Hz** and recompute. The collision is perfectly inelastic (the particle does not bounce off the wall and sticks to it)
- (d) Change the **Number of tracers** to **4** and the **Diameter standard deviation** to **0.005**. Now 4 tracers will be created at the source surface and at the initial time. The particle sizes follow a normal distribution with mean diameter **0.001 m** and **Diameter standard deviation**: **0.005**. Therefore each particle has a different diameter. Show the current marker as a point and play to see the motion of the particles.
- (e) Change the **Number of tracers** to **0** and **Particles flux rate** to **4**. Now particles will be created at a rate of 4 particles per second from random positions at the source surface.
- (f) Set **Turbulent dispersion**: **On** to take into account the fluctuations in the velocity field due to turbulence

Tutorial 11 - FMI standard co-simulation: Pendulum

This tutorial addresses a pendulum modeling by coupling Dymola or OpenModelica and XFlow through Standard FMI. Dymola/OpenModelica give the angular position via the pendulum equation, using the forces calculated by XFlow.



FMI Standard works with the coordinator-worker concept: the workers simulate sub-problems whereas the coordinator is responsible for both coordinating the overall simulation as well as transferring data. XFlow works just as a worker, so it can be connected to external software defined as a coordinator. Within the present tutorial, two external coordinators software are used:

- OpenModelica is an open-source Modelica-based modeling and simulation environment intended for industrial and academic usage.
- Dymola, dynamic modeling laboratory, is a complete tool for modeling and simulation of integrated and complex systems.

It is assumed that the reader has completed Tutorials 01, 02 and 06. Some steps in the setup and post-process will not be described in detail.

This tutorial shows how to:

- Export to FMI Standard
- Couple with a coordinator software, i.e. Dymola or OpenModelica

⚠ Please note: This tutorial requires having activated the Labs mode. The Labs mode can be activated in: **Main menu > Options > Preferences > Application mode**

⚠ Please note: The version used of OpenModelica is 1.9.1 (r22929). The version used of Dymola is 2021.

Contents

[Step 1: Problem setup - XFlow](#)

[Step 2: Set FMI standard in XFlow](#)

[Step 3: Problem setup - OpenModelica](#)

[Step 4: Execution of Co-simulation](#)

[Step 5: Post-processing](#)

Step 1: Problem setup - XFlow

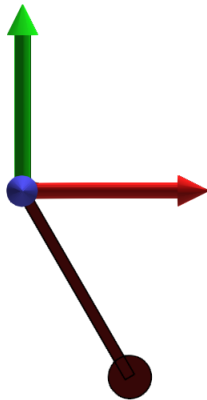
1.1 Import pendulum

Import a pendulum geometry by means of:

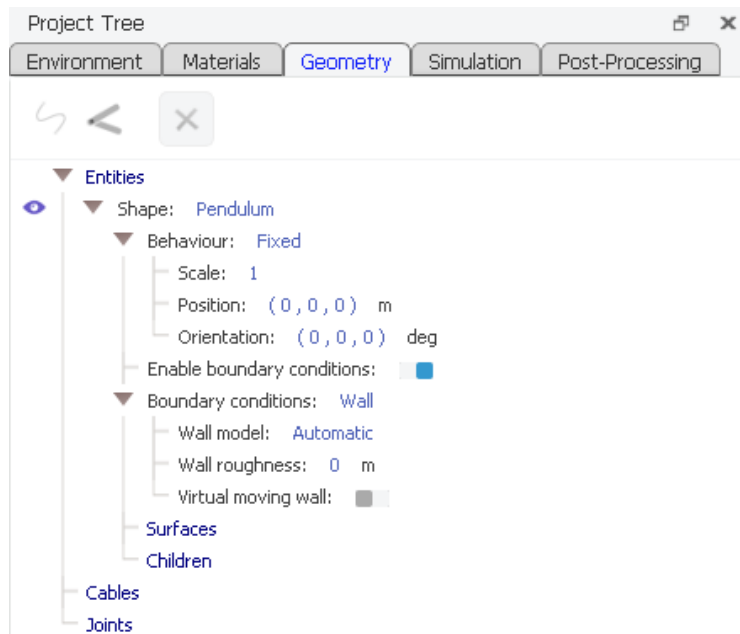
Main menu > Geometry > Import a new geometry (or use the toolbar icon ) and select the file `Pendulum.nfb`. Select "Single shape" structure, visualization "Shaded" and press **Ok**.

The imported geometry will be shown in the Graphic View, and it will appear as a **Shape** in **Project Tree > Geometry > Entities**:

The local axis are centered at the top extremity with the global axis orientation, it belongs with the axis rotation.



Pendulum geometry



Project Tree > Geometry > Entities

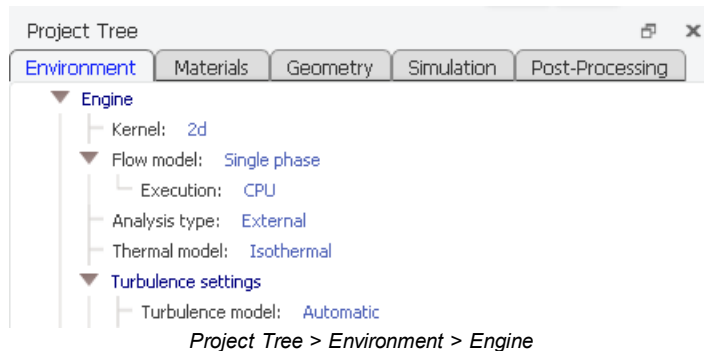
1.2 Engine settings

Configure the section **Project Tree > Environment > Engine >** as follows:

- (a) **Kernel** : 2d
- (b) **Flow model**: Single phase
- (c) **Analysis type**: External
- (d) **Thermal model**: Isothermal

(e) Turbulence settings:

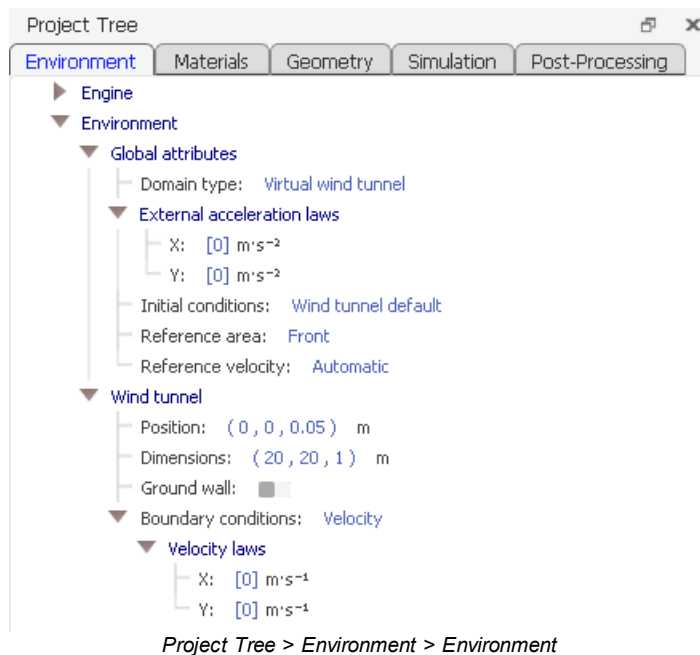
- Turbulence model: **Automatic**



1.3 Environment settings

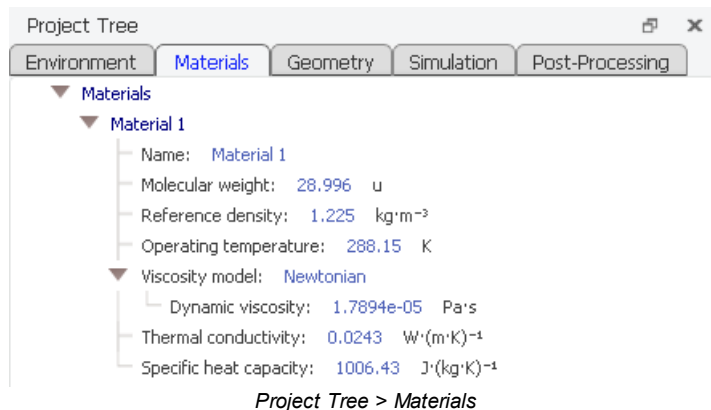
Configure the section **Project Tree > Environment > Environment >** as follows:

- (a) **Global attributes > Domain type: Virtual wind tunnel**
- (b) **Global attributes > Ext. acceleration laws:** leave it to 0 (these accelerations will only act on the fluid. The pendulum's weight is included in the OpenModelica modeling).
- (c) **Global attributes > Initial conditions: Wind tunnel default**
- (d) **Global attributes > Reference area: Front**
- (e) **Global attributes > Reference velocity: Automatic**
- (f) **Wind tunnel > Position:** set it to 0, 0, 0.05.
- (g) **Wind tunnel > Dimensions:** 20 meters length (X), 20 meters height (Y) and 1 meter width (Z).
- (h) **Wind tunnel > Ground wall: Off**
- (i) **Wind tunnel > Boundary conditions: Velocity**
- (j) **Wind tunnel > Boundary conditions > Velocity law:** leave it to 0, 0.



1.4 Materials settings

By default in single phase the fluid material is called *Material 1*, which is by default air standard at 15 degrees.

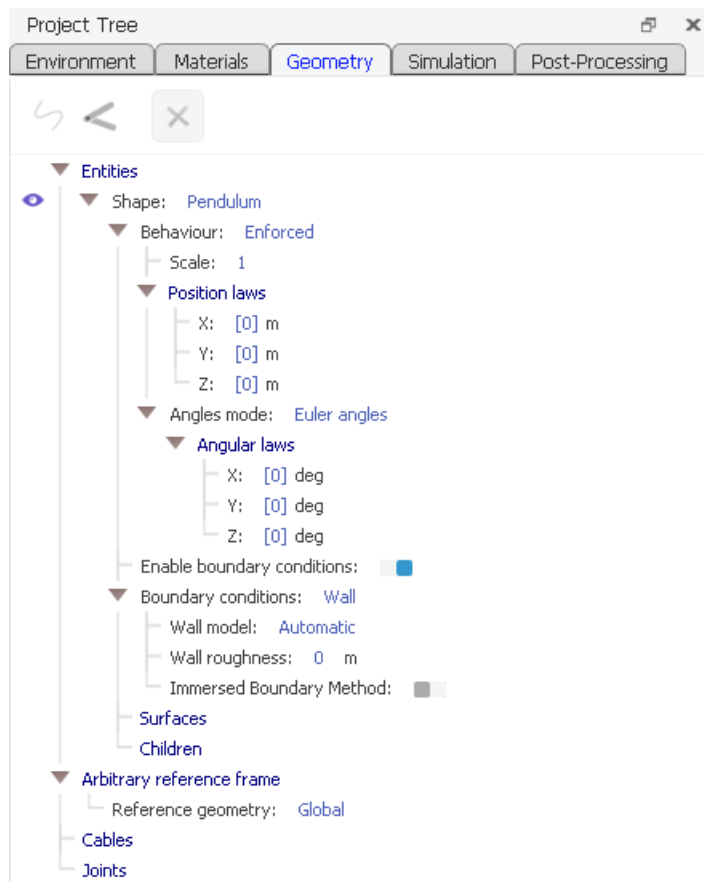


1.5 Geometry settings

The geometry (Pendulum) has been previously imported. In **Project Tree > Geometry**, its behaviour and boundary conditions can be defined as follows:

- Project Tree > Geometry > Entities > Shape: Pendulum > Behaviour: Enforced**, leave blank position and angular laws.
- Project Tree > Geometry > Entities > Shape: Pendulum > Boundary conditions: Wall**, with Automatic wall model and zero roughness

Step 1: Problem setup - XFlow



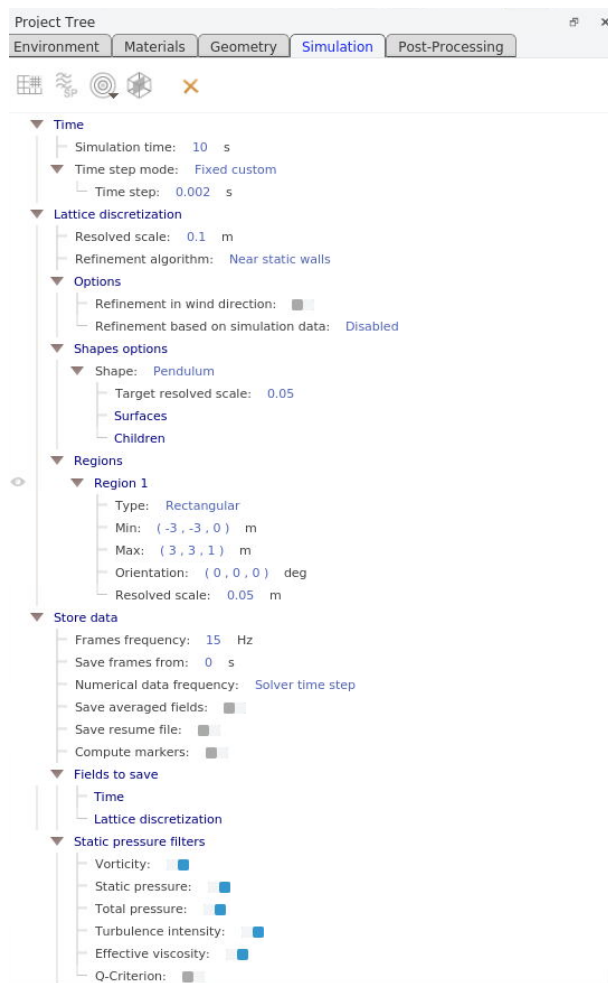
Project Tree > Geometry > Entities

2.5 Simulation settings

Configure the section **Project Tree > Simulation >** as follows:

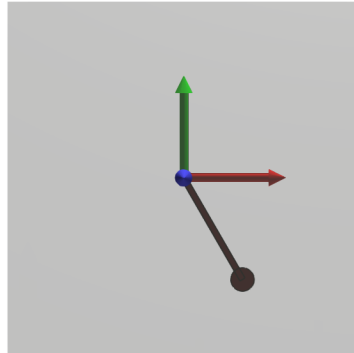
- (a) **Time > Simulation time: 10 s**
- (b) **Time > Time step mode: Fixed Custom**
- (c) **Time > Time step mode > Time step: 0.002 s**
- (d) **Lattice discretization > Resolved scale: 0.1 m**
- (e) **Lattice discretization > Refinement algorithm: Near static walls**
- (f) **Lattice discretization > Shapes refinement > Shape: Pendulum > Target resolved scale: 0.05 m**
- (g) **Lattice discretization > Region: Rectangular > Min:(-3,-3,-1) ; Max: (3,3,1) ; Resolved scale: 0.05 m**
- (h) **Store data > Frames frequency: 15 Hz** (i.e. a total of 150 frames will be saved for the 10 seconds of simulation time)
- (i) **Store data > Save frames from: 0 s** (i.e. a total of 150 frames will be saved for the 10 seconds of simulation time)

- (j) **Store data > Numerical data frequency: Solver time step**, which means any curve plotted in the *Function Viewer* will be updated at the frequency of the solver steps.
- (k) **Store data > Save averaged fields: Off**
- (l) **Store data > Save resume file: Off**. In case you need to stop and resume your simulation you can switch it to On, however this consumes more hard disk space.
- (m) **Store data > Compute markers: Off**
- (n) **Store data > Fields to save:** Leave the default selection in all the fields, which means that all the flow fields will be saved on the hard disc.



Project Tree > Simulation

Step 1: Problem setup - XFlow



XFlow

Region refinement

2.6 Save project

Save the project as Pendulum.xfp (or .xfz if the compressed file format is preferred).

At this point the setup is finished and the [FMI Standard has to be exported](#).

Step 2: Set FMI standard in XFlow

2.1 Export FMI Standard

Export FMI Standard file which contains the input/output variables to communicate with OpenModelica :

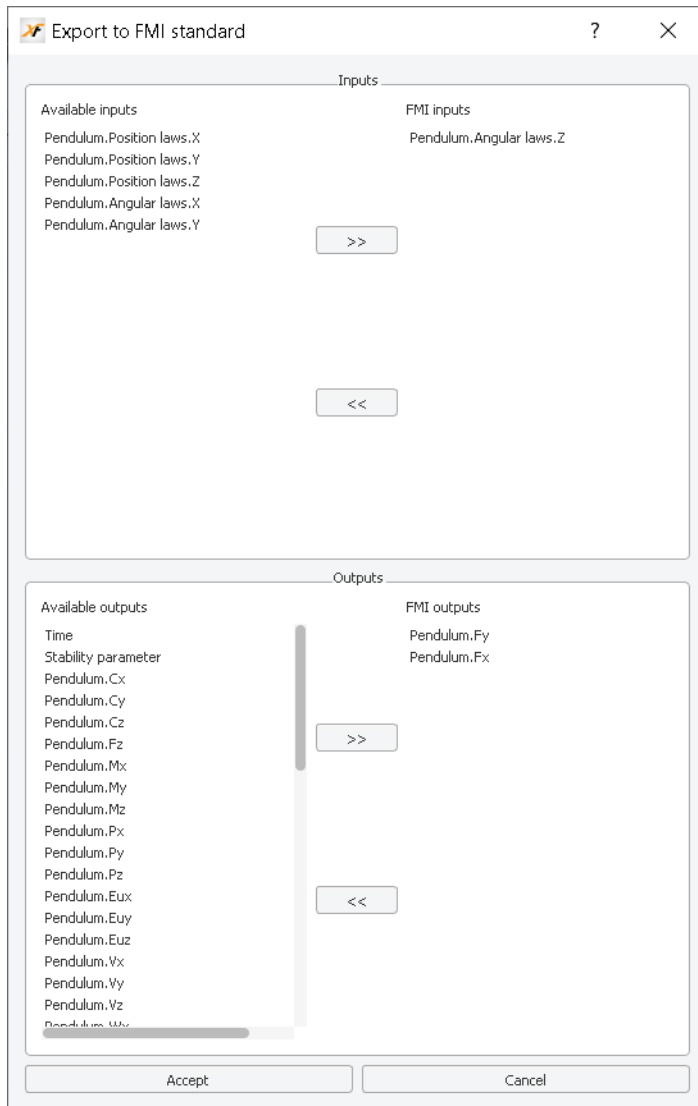
Main menu > Options > Export to FMI standard

The Pendulum modeling needs an angular position as input and the internal forces (Fx and Fy) as output.

Select "Pendulum.Angular laws Z" FMI input and press . Select "Pendulum.Fx" and "Pendulum.Fy"

FMI outputs and press . Finally press **Accept** and the file xflowFMI.fmu is created in the simulation folder. It will be used by OpenModelica for the co-simulation.

⚠ Please note: Currently XFlow is only able to use the geometry translational and angular laws as FMI inputs, and any variable stored in the numerical data as FMI outputs.



FMI outputs and inputs

Step 3: Problem setup - Dymola/OpenModelica

Step 3.1: OpenModelica

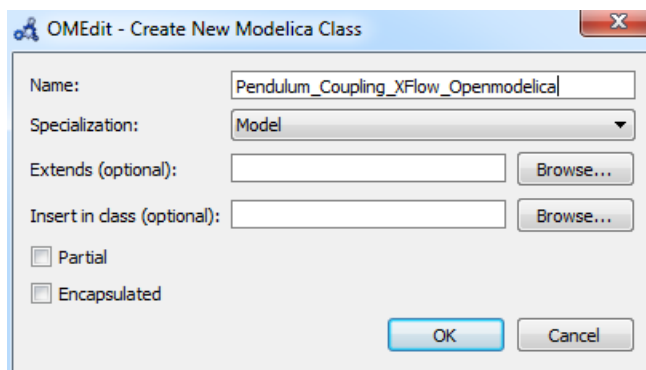
The user needs to install OpenModelica in its computer.

⚠ Please note: The version used of OpenModelica is 1.9.1 (r22929) available online.

3.1.1 Create new Modelica Class

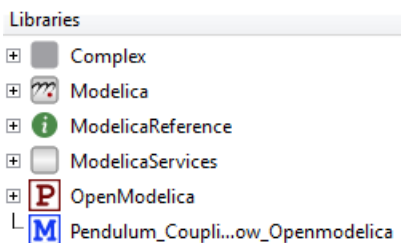
Execute OpenModelica through the direct link in your desktop or by double-clicking the `OMEdit.exe` file located in the installation folder. The application displays the *Graphical User Interface*.

Create a new Modelica Class **Main menu > File > New Modelica Class**, enter the name "Pendulum_Coupling_XFlow_Openmodelica" and press **Ok**.



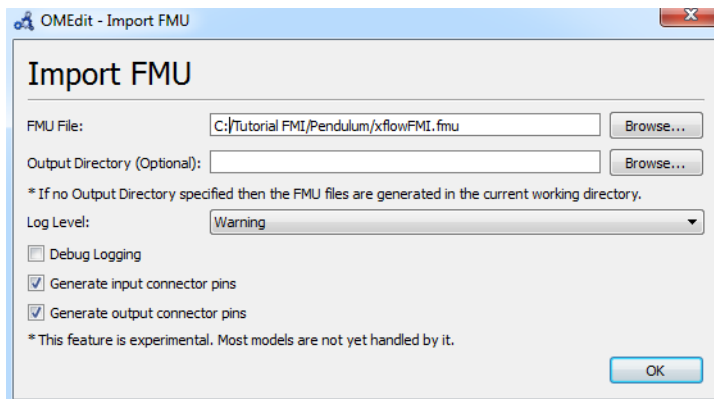
Main menu > File > New Modelica Class

In the Libraries tree (left) a new Modelica Class has appeared, called "Pendulum_Coupling_XFlow_Openmodelica".



3.1.2 Import FMU

Import the file `xflowFMI.fmu` created by XFlow to OpenModelica **Main menu > FMI > Import FMU**.



Main menu > FMI > Import FMI

3.1.3 Set pendulum equation

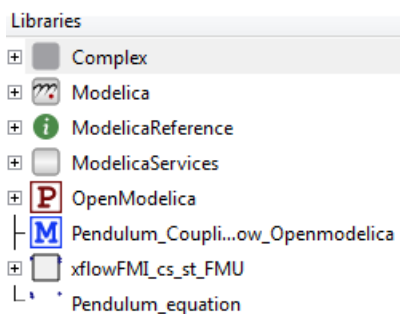
In this simulation the Pendulum equation is going to be introduced as an external library. It contains the following code:

```
1 model Pendulum_equation
2   import Modelica.SIunits.Conversions;
3   parameter Real m = 1, g = 9.81, L = 2.2;
4   constant Real pi = 2 * Modelica.Math.asin(1.0);
5   Modelica.Blocks.Interfaces.RealOutput theta annotation(Placement(transformation(extent = {{100, 60}, {120, 80}})));
6   Modelica.Blocks.Interfaces.RealInput extFx annotation(Placement(transformation(extent = {{-120, 60}, {-100, 80}})));
7   Modelica.Blocks.Interfaces.RealInput extFy annotation(Placement(transformation(extent = {{-100, 35}, {-120, 55}})));
8   output Real thetaRad;
9   //theta initialization of 30 degrees from equilibrium.
10  Real y(start = -1.905255888, fixed = true);
11  Real x(start = 1.1, fixed = true);
12  output Real F;
13  output Real vx, vy;
14  equation
15    //X axis
16    m * der(vx) = (-x) / L * F + extFx;
17    //Y axis
18    m * der(vy) = (-y) / L * F - m * g + extFy;
19    der(x) = vx;
20    der(y) = vy;
21    x^2 + y^2 = L^2;
22    thetaRad = atan2(x, -y);
23    //Angular position in degrees
24    theta = Modelica.SIunits.Conversions.to_deg(thetaRad) - 30;
25    annotation(Placement(visible = true, transformation(origin = {0, 80}, extent = {{-10, -10}, {10, 10}}, rotation = 0)), Icon(coordinateSystem(extent = {{-100, -100}, {100, 100}}, preserveAspectRatio = true, initialScale = 0.1, grid = {2, 2})), Diagram(coordinateSystem(extent = {{-100, -100}, {100, 100}}, preserveAspectRatio = true, initialScale = 0.1, grid = {2, 2})), experiment(StartTime = 0, StopTime = 2, Tolerance = 1e-06, Interval = 0.004));
26 end Pendulum_equation;
```

Pendulum equation code

Open the file Pendulum_equation.mo which contains the equation of the pendulum behavior **Main menu > File > Open Model/Library file(s)**.

It appears in the Libraries tree:



Libraries tree

3.1.4 Modeling

Open in the libraries tree the "Pendulum_Coupling_XFlow_Openmodelica" by double **clicking** the **left mouse button**, then pick and drag the "xflowFMI_cs_st_FMU" and "Pendulum_equation" libraries on the Diagram view. Connect the inputs (blue arrows) and the output (white arrows) between them through wires as following:

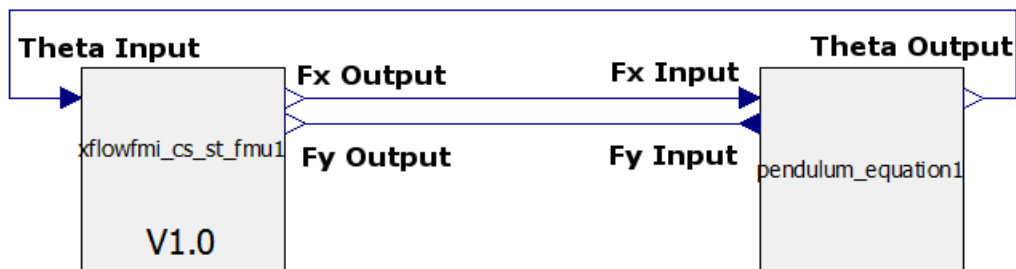


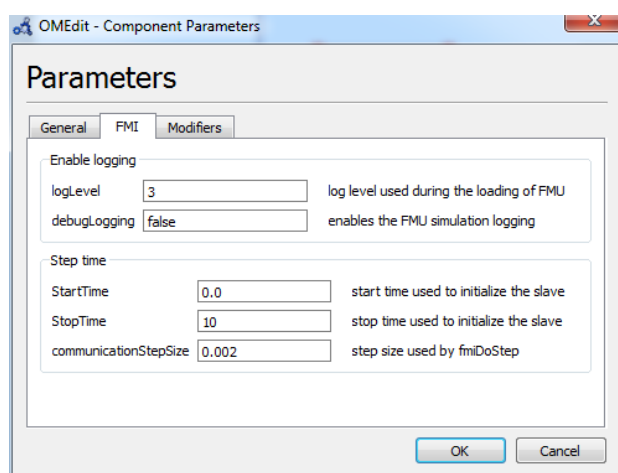
Diagram OpenModelica

3.1.5 Simulation settings

Configure the "xflowFMI_cs_st_FMU" parameter by double **clicking** the **left mouse button** on the "xflowFMI_cs_st_FMU" box in the Diagram view to open the setup. Configure the section **Parameters > FMI** as follows:

- (a) **StartTime:** 0 s
- (b) **StopTime:** 10 s.
- (c) **communicationStepSize:** 0.002 s.

⚠ Please note: These values should match with the XFlow values.



xflowFMI_cs_st_FMU parameters

Configure the "Pendulum_equation" simulation parameter by **selecting** the **Simulation setup** icon



in the main menu bar.

Configure the section **Simulation > General** as follows:

- (d) **StartTime**: 0 s.
- (e) **StopTime**: 10 s.

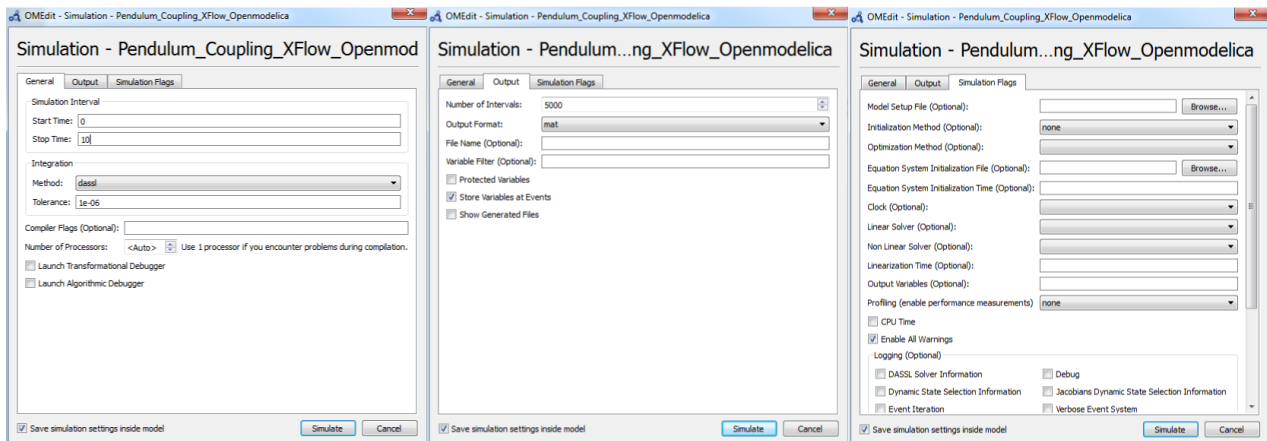
Configure the section **Simulation > Output** as follows:

- (f) **Number of Intervals**: 5000
which is the result of the current operation(**StopTime** - **StartTime**) / **communicationStepSize**.

Configure the section **Simulation > Simulation Flags** as follows:

- (g) **Initialization Method**: none.

Step 3: Problem setup - Dymola/OpenModelica



Pendulum_equation simulation parameters

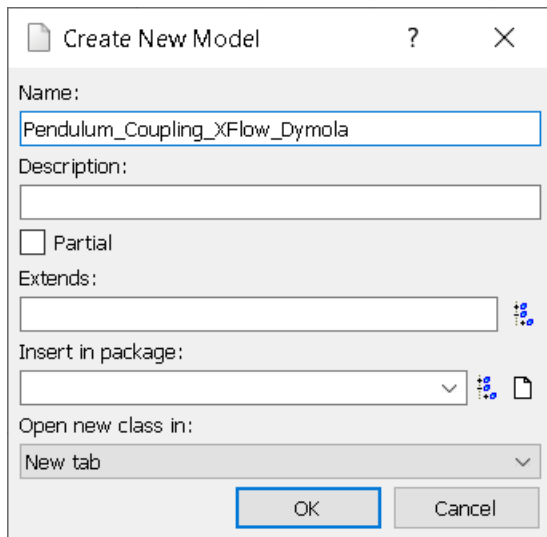
Leave the remaining parameters to the default value.

Step 3.2: Dymola

The user needs to install Dymola 2021 in its computer.

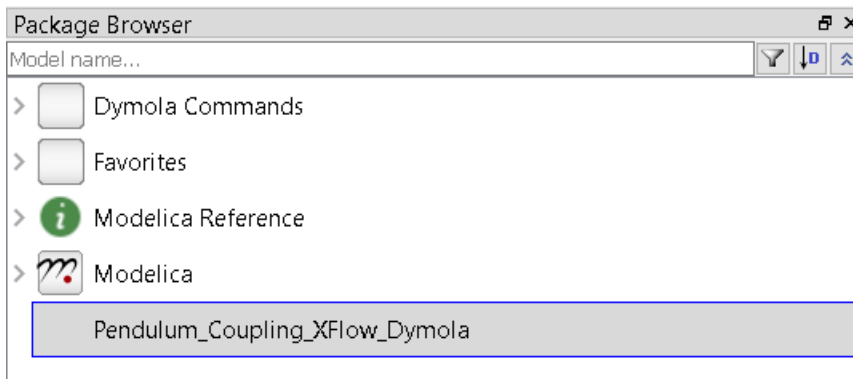
3.2.1 Create new Modelica Class

Create a new Modelica model **Main menu > File > New > Model** , enter the name "Pendulum_Coupling_XFlow_Dymola" and press **Ok**.



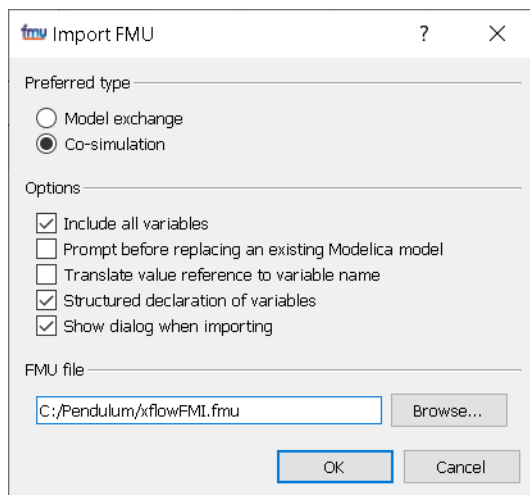
Main menu > File > New > Model

In the Libraries tree (left) a new Modelica Class has appeared, called "Pendulum_Coupling_XFlow_Dymola".



3.2.2 Import FMU

Import the file `xflowFMI.fmu` created by XFlow to Dymola **Main menu > File > Open > Import FMU...**



Main menu > File > Open > Import FMU

3.2.3 Set pendulum equation

In this simulation the Pedulum equation is going to be introduced as an extern library. It contains the following code:

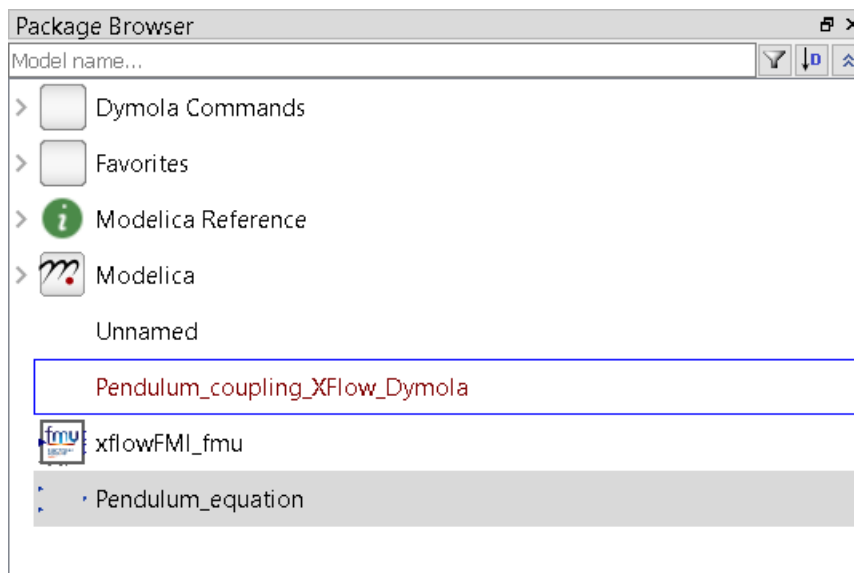
Step 3: Problem setup - Dymola/OpenModelica

```
1 model Pendulum_equation
2 import Modelica.SIunits.Conversions;
3 parameter Real m = 1, g = 9.81, L = 2.2;
4 constant Real pi = 2 * Modelica.Math.asin(1.0);
5 Modelica.Blocks.Interfaces.RealOutput theta annotation(Placement(transformation(extent = {{100, 60}, {120, 80}})));
6 Modelica.Blocks.Interfaces.RealInput extFx annotation(Placement(transformation(extent = {{-120, 60}, {-100, 80}})));
7 Modelica.Blocks.Interfaces.RealInput extFy annotation(Placement(transformation(extent = {{-100, 35}, {-120, 55}})));
8 output Real thetaRad;
9 //theta initialization of 30 degrees from equilibrium.
10 Real y(start = -1.905255888, fixed = true);
11 Real x(start = 1.1, fixed = true);
12 output Real F;
13 output Real vx, vy;
14 equation
15 //X axis
16 m * der(vx) = (-x) / L * F + extFx;
17 //Y axis
18 m * der(vy) = (-y) / L * F - m * g + extFy;
19 der(x) = vx;
20 der(y) = vy;
21 x^2 + y^2 = L^2;
22 thetaRad = atan2(x, -y);
23 //Angular position in degrees
24 theta = Modelica.SIunits.Conversions.to_deg(thetaRad) - 30;
25 annotation(Placement(visible = true, transformation(origin = (0, 80), extent = {{-10, -10}, {10, 10}}, rotation = 0)), Icon(coordinatesSystem(extent = {{-100, -100}, {100, 100}}, preserveAspectRatio = true, initialScale = 0.1, grid = {2, 2})), Diagram(coordinatesSystem(extent = {{-100, -100}, {100, 100}}, preserveAspectRatio = true, initialScale = 0.1, grid = {2, 2})), experiment(StartTime = 0, StopTime = 2, Tolerance = 1e-06, Interval = 0.004));
26 end Pendulum_equation;
```

Pendulum equation code

Open the file Pendulum_equation.mo which contains the equation of the pendulum behavior **Main menu > File > Open > Load.**

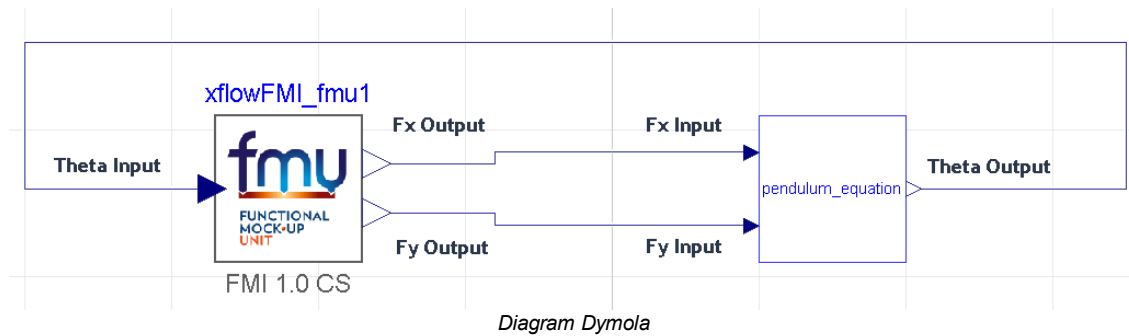
It appears in the Libraries tree:



Libraries tree

3.2.4 Modeling

Open in the libraries tree the "Pendulum_Coupling_XFlow_Dymola" by double **clicking** the **left mouse button**, then pick and drag the "xflowFMI_cs_st_FMU" and "Pendulum_equation" libraries on the Diagram view. Connect the inputs (blue arrows) and the output (white arrows) between them through wires as following:



3.2.5 Simulation settings

Configure the "xflowFMI_cs_st_FMU" parameter by double **clicking** the **left mouse button** on the "xflowFMI_cs_st_FMU" box in the Diagram view to open the setup. Configure the section **FMI** as follows:

- (a) **fmi_StartTime:** 0 s
- (b) **fmi_StopTime:** 10 s
- (c) **fmi_NumberOfSteps:** 5000
- (d) **fmi_communicationStepSize:** 0.002 s.

⚠ Please note: These values should match with the XFlow values.

Step 3: Problem setup - Dymola/OpenModelica

The screenshot shows the 'xflowFMI_fm1 in Pendulum_coupling_XFlow_Dymola' dialog box with the 'FMI' tab selected. The 'General' sub-tab is active, showing various configuration options for the FMI instance. The 'Instance name' is 'xflowFMI_fm1'. 'Enable logging' is set to 'false'. 'Inputs for initialization' has 'fmi_pullInputsForInitialization' set to 'true'. 'Input Handling' has 'fmi_InputTime' set to 'false' and 'fmi_UsePreOnInputSignals' set to 'true'. 'Step time' parameters include 'fmi_StartTime' (0), 'fmi_StopTime' (10), 'fmi_NumberOfSteps' (5000), 'fmi_CommunicationStepSize' (0.002), 'stepSizeScaleFactor' (1), and 'fmi_forceShutDownAtStopTime' (false). 'Instantiation' has 'fmi_fmLocation' set to a file path. The 'OK' button is highlighted.

Parameter	Value	Description
fmi_instanceName	"xflowFMI_fm1"	
fmi_loggingOn	false	
fmi_pullInputsForInitialization	true	Set to false when using the FMU in feedback connections
fmi_InputTime	false	Time point of input used when calling doStep.
fmi_UsePreOnInputSignals	true	
fmi_StartTime	0	
fmi_StopTime	10	
fmi_NumberOfSteps	5000	
fmi_CommunicationStepSize	0.002	
stepSizeScaleFactor	1	Number of doSteps called between two CommunicationStepSize
fmi_forceShutDownAtStopTime	false	
fmi_fmLocation	"file:/// + ModelicaServices.ExternalReferences.loadResource("modelica://xflowFMI_fm1/Resources/Library/FMU/xflowFMI")	

xflowFMI_cs_st_FMU parameters

Configure the "Pendulum equation" parameter by double **clicking** the **left mouse button** on the "Pendulum equation" box in the Diagram view to open the setup. Configure the section **General** as follows

(e) **Initialization x.start**: False: start-value is only a guess-value

(f) **Initialization y.start**: False: start-value is only a guess-value

The screenshot shows the 'pendulum_equation in Pendulum_Coupling_XFlow_Dymola' dialog box with the 'General' tab selected. The 'Component' section shows 'Name' as 'pendulum_equation' and 'Path' as 'Pendulum_equation'. The 'Parameters' section includes 'm' (1), 'g' (9.81), and 'L' (2.2). The 'Initialization' section has 'y.start' (-1.905259888) and 'x.start' (1.1). The 'OK' button is highlighted.

Parameter	Value
m	1
g	9.81
L	2.2
y.start	-1.905259888
x.start	1.1

Pendulum equation parameters

Configure the "Pendulum_equation" simulation parameter by selecting **Main menu > Simulation tab > Setup** icon in the main menu bar.



Configure the section **Simulation > General** as follows:

- (g) **StartTime**: 0 s.
- (h) **StopTime**: 10 s.

Configure the section **Simulation > Output** as follows:

- (i) **Number of Intervals**: 5000
which is the result of the current operation $(\text{StopTime} - \text{StartTime}) / \text{communicationStepSize}$.
- (j) **Integration algorithm**: Dassl with Tolerance = 1e-4

Leave the remaining parameters to the default value.

Step 4: Execution of Co-simulation

4.2 Start FMU computation in XFlow

Click the **Run** button Run > **Start FMU computation**

XFlow waits until the coordinator start the computations.

```

Message View:
Full domain has 50800 elements.
Equivalent single-resolution domain has 160000
Equivalent single-resolution domain size is ( 400 x 400 x 1 )
-----
---

Computing boundary conditions map!
-----
---

Coarsest resolved length: 0.1
-----
    
```

Step 4: Execution of Co-simulation

```
---  
[WARNING] 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.  
Prandtl number: 0.741114  
Thermodynamic speed of sound: 340.112  
Reference area: 0.195704 m^2  
Reference velocity: 1 m/s  
Time step (level 0): 0.0020202 s  
-----  
---  
Waiting FMU initialization...
```

4.3 Start simulation in OpenModelica/Dymola

Run the simulation in the "Pendulum_equation" simulation parameter:

- In the case of OpenModelica by **selecting** the **Simulation setup** icon  in the main menu bar and press **Simulate**. Save the file `Pendulum_Coupling_XFlow_Openmodelica.mo` in the Pendulum folder generate by XFlow.
- In the case of Dymola by **selecting** the **Simulate icon**. Save the file `Pendulum_Coupling_XFlow_Dymola.mo` in the Pendulum folder generate by XFlow.

4.4 Message View

Check the *Message View* window. It is showing all the information XFlow needs to communicate, including the computation logs.

During the computation, the following messages will appear:

```
Message View:  
FMU simulation start time: 0 stop time: 1.79769e+308  
Waiting first FMU DoStep...  
Saving data...  
[[Data file]] 0 done!!! | Frame wall clock time[0]s | Overall wall clock  
time[0]s | Num elements[50800]  
Sim. time [2.020202e-003]s | Stability param. [1.460356e-005] | Wall clock  
time [1.250000e-001]s  
Sim. time [4.040404e-003]s | Stability param. [1.460356e-005] | Wall clock
```

```

time [1.410000e-001]s
  Sim. time [6.060606e-003]s | Stability param. [2.985308e-004] | Wall clock
time [2.180000e-001]s
  Sim. time [8.080808e-003]s | Stability param. [1.041214e-003] | Wall clock
time [1.880000e-001]s
  Sim. time [1.010101e-002]s | Stability param. [1.788178e-003] | Wall clock
time [2.340000e-001]s
  Sim. time [1.212121e-002]s | Stability param. [2.953553e-003] | Wall clock
time [2.350000e-001]s
  Sim. time [1.414141e-002]s | Stability param. [3.808381e-003] | Wall clock
time [2.970000e-001]s
  Sim. time [1.616162e-002]s | Stability param. [4.205828e-003] | Wall clock
time [2.960000e-001]s
  Sim. time [1.818182e-002]s | Stability param. [4.924741e-003] | Wall clock
time [2.500000e-001]s
  Sim. time [2.020202e-002]s | Stability param. [5.263237e-003] | Wall clock
time [2.820000e-001]s
  Sim. time [2.222222e-002]s | Stability param. [5.386357e-003] | Wall clock
time [2.650000e-001]s
  Sim. time [2.424242e-002]s | Stability param. [5.426862e-003] | Wall clock
time [2.190000e-001]s
  Sim. time [2.626263e-002]s | Stability param. [5.298300e-003] | Wall clock
time [2.190000e-001]s
  Sim. time [2.828283e-002]s | Stability param. [5.176308e-003] | Wall clock
time [2.340000e-001]s
  Sim. time [3.030303e-002]s | Stability param. [5.090734e-003] | Wall clock
time [1.250000e-001]s
  Sim. time [3.232323e-002]s | Stability param. [4.919204e-003] | Wall clock
time [1.250000e-001]s
  Sim. time [3.434343e-002]s | Stability param. [4.711709e-003] | Wall clock
time [1.410000e-001]s
  Sim. time [3.636364e-002]s | Stability param. [4.517341e-003] | Wall clock
time [1.560000e-001]s
  Sim. time [3.838384e-002]s | Stability param. [4.280114e-003] | Wall clock
time [1.090000e-001]s

```

Step 4: Execution of Co-simulation

```
Sim. time [4.040404e-002]s | Stability param. [4.230932e-003] | Wall clock
time [1.250000e-001]s
Sim. time [4.242424e-002]s | Stability param. [4.335968e-003] | Wall clock
time [1.880000e-001]s
Sim. time [4.444444e-002]s | Stability param. [4.554148e-003] | Wall clock
time [1.410000e-001]s
Sim. time [4.646465e-002]s | Stability param. [4.767117e-003] | Wall clock
time [1.400000e-001]s
Sim. time [4.848485e-002]s | Stability param. [4.976248e-003] | Wall clock
time [1.410000e-001]s
Sim. time [5.050505e-002]s | Stability param. [5.186340e-003] | Wall clock
time [1.250000e-001]s
Sim. time [5.252525e-002]s | Stability param. [5.397957e-003] | Wall clock
time [1.250000e-001]s
Sim. time [5.454545e-002]s | Stability param. [5.607738e-003] | Wall clock
time [1.870000e-001]s
Sim. time [5.656566e-002]s | Stability param. [5.812717e-003] | Wall clock
time [9.400000e-002]s
Sim. time [5.858586e-002]s | Stability param. [6.009043e-003] | Wall clock
time [1.090000e-001]s
Sim. time [6.060606e-002]s | Stability param. [5.887624e-003] | Wall clock
time [1.100000e-001]s
Sim. time [6.262626e-002]s | Stability param. [5.740904e-003] | Wall clock
time [1.250000e-001]s
Sim. time [6.464646e-002]s | Stability param. [5.696175e-003] | Wall clock
time [1.400000e-001]s
Sim. time [6.666667e-002]s | Stability param. [5.860001e-003] | Wall clock
time [1.100000e-001]s
Saving data...
[[Data file]] 1 done!!! | Frame wall clock time[5.813]s | Overall wall
clock time[5.813]s | Num elements[50800]
Sim. time [6.868687e-002]s | Stability param. [6.045215e-003] | Wall clock
time [9.400000e-002]s
Sim. time [7.070707e-002]s | Stability param. [6.197369e-003] | Wall clock
time [1.410000e-001]s
```

```

Sim. time [7.272727e-002]s | Stability param. [6.370204e-003] | Wall clock
time [1.090000e-001]s
Sim. time [7.474747e-002]s | Stability param. [6.505235e-003] | Wall clock
time [7.800000e-002]s
Sim. time [7.676768e-002]s | Stability param. [6.649560e-003] | Wall clock
time [1.250000e-001]s

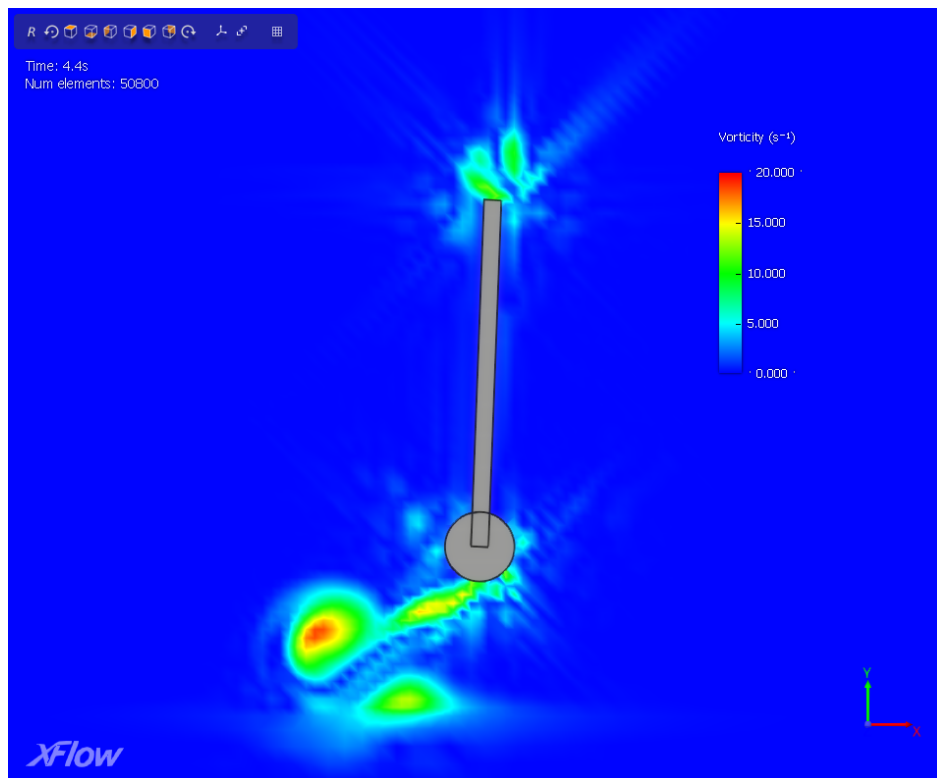
```

Step 5: Post-processing

It is possible to perform the entire post-processing of the simulation, including visualisation of the resultant motion of the pendulum, in XFlow. Nonetheless, this tutorial will also illustrate how to post-process the coupled simulation in OpenModelica by plotting the variable values of the Pendulum equation.

5.1 XFlow post-processing

Create a cutting plane to visualise the vorticity field which shows the vortices detached by Pendulum's movement. To do so, please go to **Project Tree > Post-processing > Cutting planes:**



Vorticity field at t=4.4s

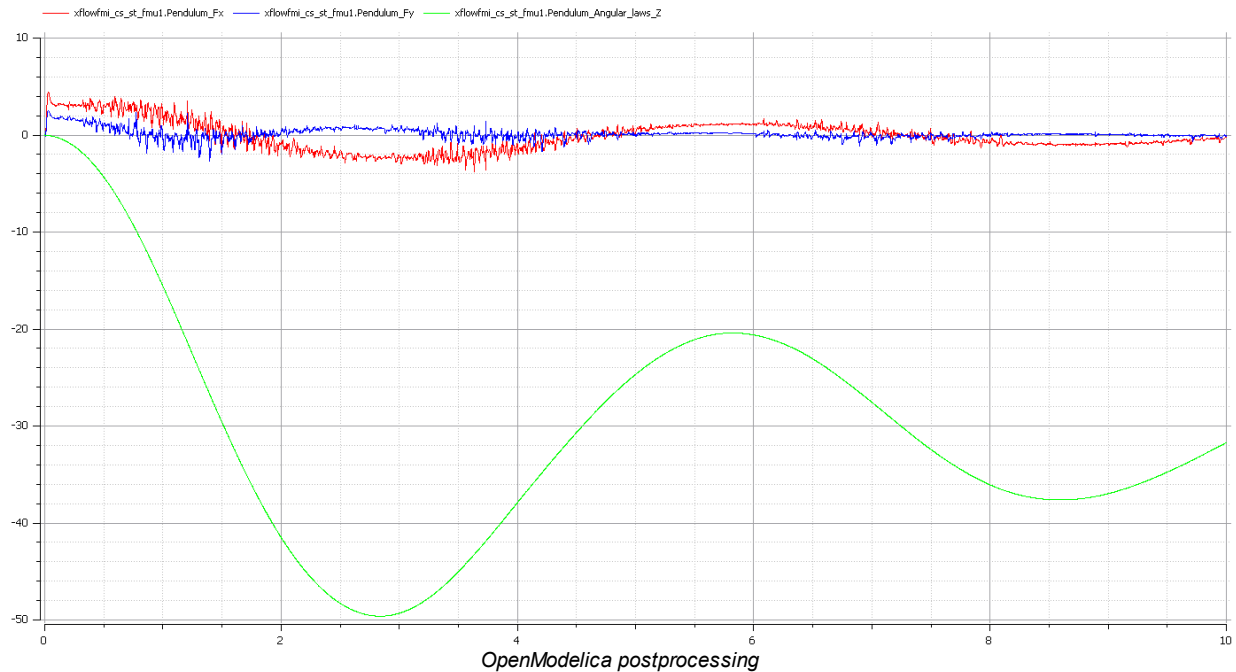
5.2 OpenModelica/Dymola post-processing

Once the simulation ends a post processing window will appear in both OpenModelica and Dymola.. The user

Step 5: Post-processing

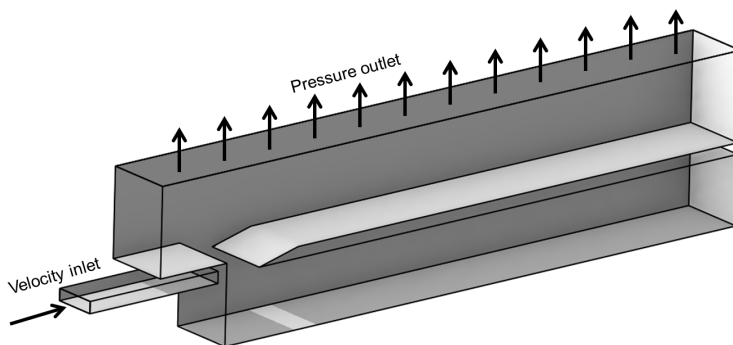
have access to the variables of both XFlow (limited to the inputs and outputs of the FMI settings) and OpenModelica/Dymola in the Variable Browser (right hand side window).

Activate the XFlow variables `Fx`, `Fy` and `Angular_laws_Z` in the Variable Browser. Firstly, it is possible to notice that the `angular_laws_Z` variable varies as a damped cosine function from a maximum of 0 to a minimum of -50 mm. This behaviour is consistent with a damped (by the air forces acting on the pendulum itself) pendulum simulation, where the initial angle of 30 deg is subtract from the output plot. Moreover, it can be observed how the forces' signals are synchronized with the Pendulum's motion.



Tutorial 12 - Helmholtz resonator

This tutorial illustrates the setup and solution of an internal flow inside a flute and its acoustics analysis. The flow impacts in the mouth of the flute generating an air resonance. It is produced by the interaction aerodynamic and acoustic phenomenon, well known as Helmholtz resonance.



This tutorial shows how to:

- Run an acoustic analysis.
- Observe the compressible unsteady pressure waves naturally captured by XFlow.
- Process the pressure signal to get the resonance frequency and the sound pressure level in frequency space.

It is assumed that the reader has completed Tutorial 01, 02 and 07. Some steps in the setup and post-process will not be described in detail.

Before starting the tutorial, please download the project data files from the XFlow[version]_TutorialsPack.zip folder.

Contents

[Step 1: Problem setup](#)

[Step 2: Post-processing](#)

Step 1: Problem setup

In order to access the acoustic analysis, the application mode must be set to **Expert**. Please make sure you are in Expert mode through the preferences options  : **Options > Preferences > Application > Application mode: Expert**.

1.1 Engine settings

In **Project Tree > Environment > Engine** set the following parameters:

- (a) **Kernel: 3d**
- (b) **Flow model: Single phase**
- (c) **Analysis type: Internal**. Now the wind tunnel will disappear and you will have to provide the geometry of the geometry that will bound the fluid domain.
- (d) **Turbulence settings > Turbulent model: Automatic**
- (e) **Acoustics analysis: On**.

The lattice-Boltzmann method is inherently compressible and unsteady, and thus physically deals with the sound pressure waves. The speed at which the pressure waves travel in the simulation is named

$$C_s = \frac{dx}{\sqrt{3}dt}$$

the numerical speed of sound and is defined by the relation: , where dx and dt are the resolution and time step locally.

Since the resolution is an input from the user, the time step can be adjusted in order to get the desired speed of sound. This is what happens when one switches on the acoustic mode: XFlow adjusts the time step in order to have the pressure waves to travel at the thermodynamic speed of sound.

The thermodynamical speed of sound is defined by the ideal gas relation: $C_{thermodynamic} = \sqrt{\frac{\gamma RT}{M}}$.

Therefore, once the acoustic mode is enabled, XFlow will force the time step to be defined such as:

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

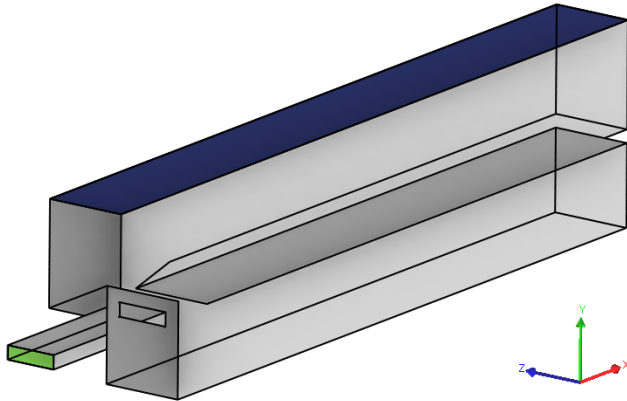
- (f) **Acoustics analysis > Refractive index: 1.0**

The refractive index can be useful if the speed of sound is not constant within a media, and can be a law of the spatial variables (x,y,z). In this tutorial, the media is air at 25°C and therefore the speed of sound is constant within the domain.

1.2 Import the geometry

Import the geometry of the flute from the files `flute.nfb`, `inlet.nfb` and `outlet.nfb`: **Main menu > Geometry > Import a new geometry** or press  in Toolbar Geometry. In the Import Geometry

pop-up window, leave the the default options and press **OK**. The geometry is then displayed in the *Graphic View 1*.



Correct flute orientation (Otherwise, reverse normals)

1.3 Environment settings

In **Project Tree > Environment > Environment** set the following parameters:

- (a) **Global attributes > Ext. acceleration laws:** **(0, 0, 0) m.s⁻²**
- (b) Leave the **Initial conditions** as **User defined**, and keep the **Initial velocity field** as **(0, 0, 0) m.s⁻¹**.

1.4 Material settings

In **Project Tree > Materials** leave the default **Material 1** properties. In single phase simulations, the default fluid is air.

1.5 Geometry settings

In **Project Tree > Geometry > Entities > Shape: flute** set the following parameters:

- (a) **Behaviour: Fixed**
- (b) **Boundary conditions: Wall**, with **Wall model: Automatic**

In **Project Tree > Geometry > Entities > Shape: inlet** set the following parameters:

- (c) **Behaviour: Fixed**
- (d) **Boundary conditions: Velocity**, with **Velocity laws** as **(7.76, 0, 0) m.s⁻¹**

In **Project Tree > Geometry > Entities > Shape: outlet** set the following parameters:

- (e) **Behaviour: Fixed**

- (f) **Boundary conditions:** Gauge pressure outlet, with Gauge pressure law as 0 Pa.

1.6 Simulation settings

In **Project Tree > Simulation** set the following parameters:

- (a) **Time > Simulation time:** 0.02 s
- (b) **Time > Time step mode:** Fixed automatic. Note the time step cannot be adjusted since it is unique when the acoustic mode is enabled due to the reasons explained in section 1.1.
- (c) **Lattice discretization > Resolved scale:** 0.00022 m, with the **Refinement algorithm:**

Disabled

In acoustics analysis, it is recommended to use a uniform resolution or use the less possible lattice levels between the acoustic source and the area of measurement as long as this is possible. This prevents from dissipation or reflections of the pressure waves when they go through lattice levels transition between two levels of the lattice. In this example, the simulation is internal and the volume is relatively small, therefore a uniform resolution is affordable and recommended.

- (d) **Lattice discretization > Options > Seed point:** Automatic

- (e) **Store data > Frames frequency:** 5000 frames per second (Hz)

XFlow will adjust this frequency to 5005.02807111 Hz in order to get an integer number of time steps between each frame. The time step is computed with the relation defined in section 1.1:

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

In this case the $dx = 0.00022$ m, and $C_{thermodynamic} = 340.112$ m/s. Hence $dt = 3.73456e-7$ s.

Therefore using $f = 5005.028$ Hz provides 535 time steps every frame.

- (f) **Store data > Numerical data frequency:** Solver time step

- (g) Leave **Off** Save averaged fields

- (h) Leave **On** all the Fields to save

- (i) Set **Probe 1 > Position** to (0.074, -0.003, 0) m. This probe will be used to measure the pressure fluctuations inside the flute cavity.

1.7 Launch the calculation

- (a) Save the project

 **Tip:** You can directly load the setup of this problem from the project file `helmholtz_resonator.xfp`

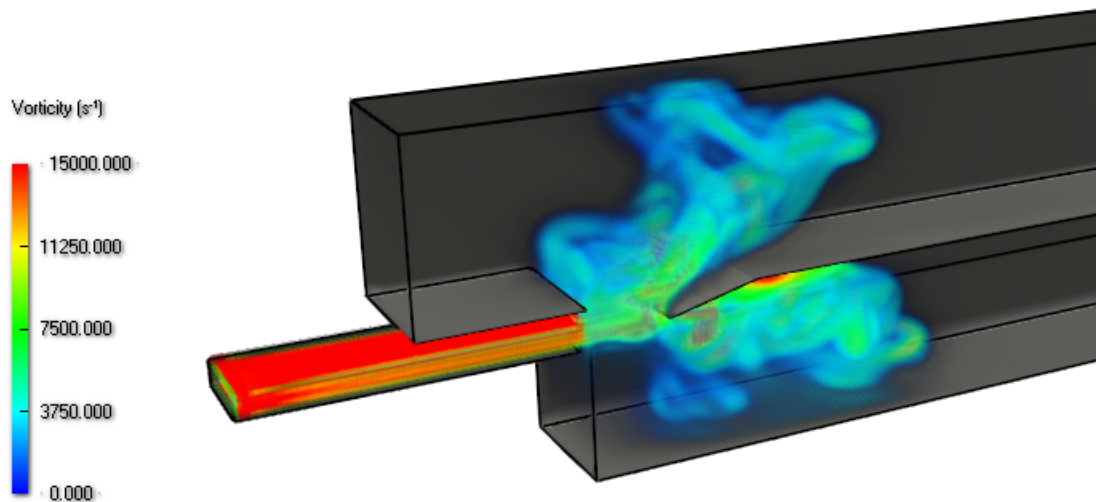
- (b) Set the number of CPUs in **Main menu > Options > Preferences > Engine**

- (c) Press **Run** button > **Start computation**.

Step 2: Post-processing

2.1 Flow visualisation

(a) Select the **Vorticity** field in **General > Show > Volumetric field > Visualisation field**, adjust its range to **[0, 15000]**, disable **General > Interpolation mode** and show the volumetric field with **Transference law: a**



3D vorticity field at frame 29

(b) In order to see the reflections of the pressure waves inside the flute cavity, create a cutting plane of static pressure: **Right click** on **Post-Processing > Cutting planes** and **Add cutting plane** with **Field: Static pressure**. Adjust the colour bar legend to **[-50, 80] Pa**. Then play backward the time control bar in order to visualize the reflections of the pressure waves in function of time.

2.2 Acoustic post-processing

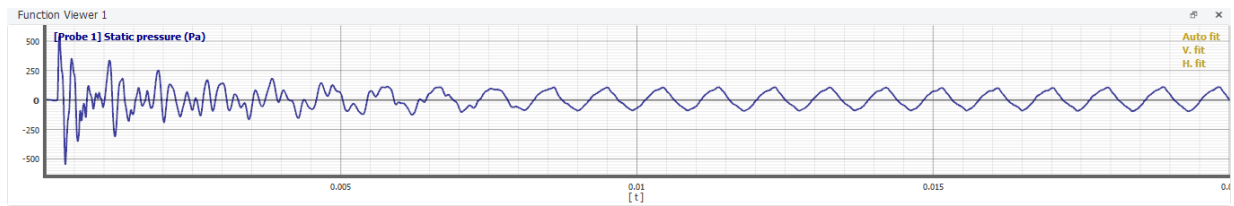
2.2.1 Spectral analysis

The purpose of this acoustic post-processing is to measure the resonance frequency of the flute and compare it with the theoretical value provided by the Helmholtz theory: **1197 Hz**.

(a) Monitor the evolution in time of the static pressure at the probe location:

right click in **Function Viewer > Probes > Probe 1 > Static pressure**

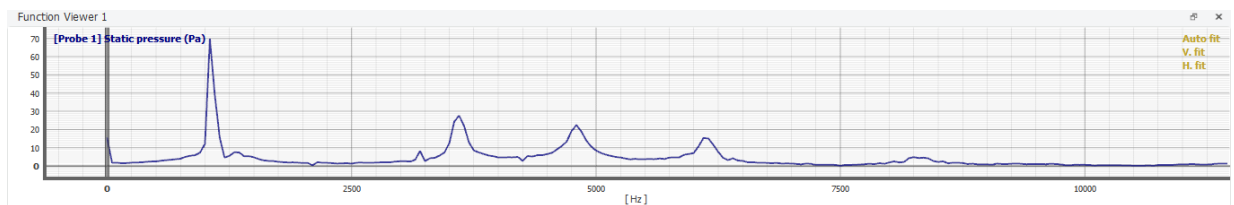
Step 2: Post-processing



Static pressure at Probe 1

In this plot, we can see the periodic fluctuations of the pressure field inside the flute. These fluctuations have a main frequency that can be measured using a Fast Fourier Transform (FFT).

(b) In order to apply the FFT on the time pressure signal measured at Probe 1, **right click** in *Function Viewer* with the Static pressure of Probe 1 displayed > **Data management** > **Set graph to [Frequency space] mode**. In the Signal filtering window, let all the options by default and press **Accept**.



Static pressure signal in frequency space (Hz)

Doing zoom at low frequencies, between 0 Pa and 10000 Pa, the main harmonics can be observed.

Moving the pointer to the higher peak, one can check that its frequency is around **1000 Hz**. This corresponds to the resonance frequency of the flute obtained by XFlow.

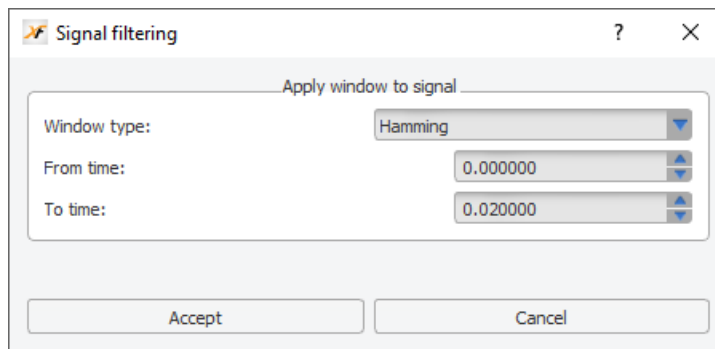
2.2.2 Windowing

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.

Furthermore, the window function can be applied in the relevant time range. For instance, the temporal pressure signal shows a transient regime where the pressure signal is not clear before 0.01 s, therefore it is convenient to apply the FFT and the window function from 0.01 s to 0.02 s. To proceed, please follow the steps:

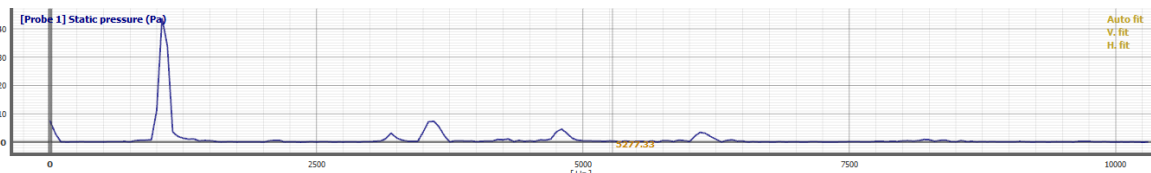
(c) Apply the FFT again on the probe signal: **Right click** in *Function Viewer* > **Data management** > **Set graph to [Frequency space] mode**.

The Signal filtering window, shown below, will appear automatically. Select "**Hamming**" window type for example. Then select the time range from **0.01 s** to **0.02 s** in order to remove the transient part of the signal and press **Accept**.



Signal filtering window

As you can see in the following plot, the obtained FFT is cleaner after applying the window filter. Some high frequencies due to the transient period have been reduced and the main frequency appears clearly now:



Static pressure signal in Frequency space applying Hamming window (t from 0.01 s to 0.02 s)

(d) Try to change the window type, with the same time range from **0.01 s** to **0.02 s**, and check the difference.

⚠ Please note: The window type affects the signal treated due to the distribution function used. For more information about these windows types and their formulations is available in the User Guide section **Post-processing > Function viewer > Data management** .

2.2.3 Signal filters

It is possible in XFlow to filter, partially or completely, the curves plotted in the Function viewer using Finite Element Response (FIR) filters. This allows to remove from the signal of interest all the non-desired frequencies, the numerical noise and any other interferences.

In this resonator case, we will try to filter the pressure signal to keep only the resonance frequency.

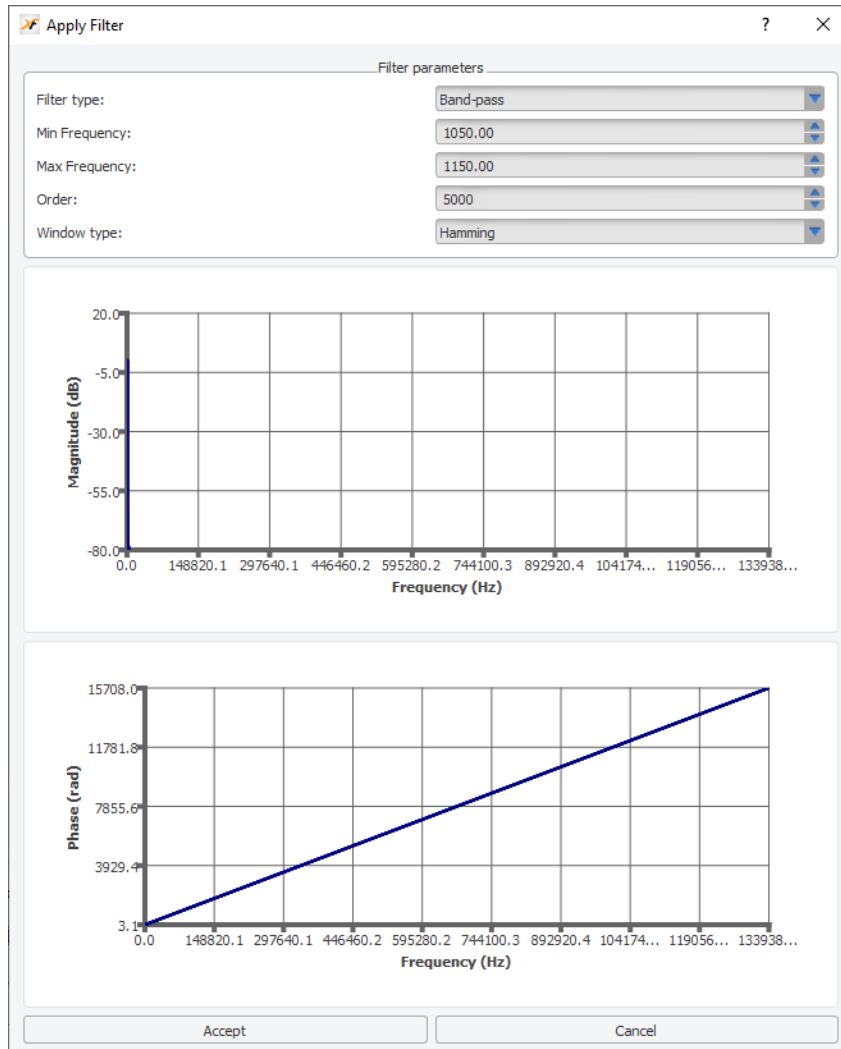
(e) Apply a filter to signal:

Plot the Static pressure signal measure by Probe 1 in time space, and **right click** in *Function Viewer* **> Data management > Apply signal to signal**.

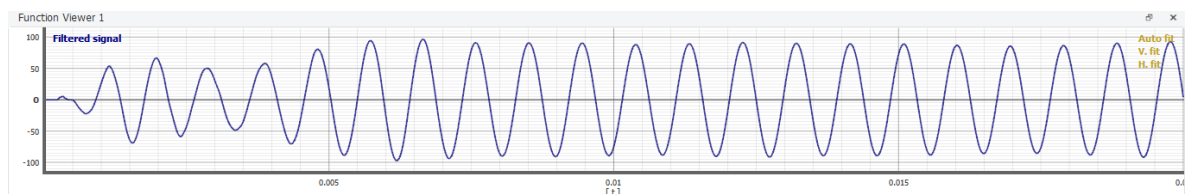
The **Signal filtering window**, shown in the figure below, will appear automatically. Select "**Band-pass**" filter type since we would like to keep only the resonance frequency and remove all the other frequencies. Therefore, set the frequency range from **1050 Hz** to **1150 Hz** in order to keep the resonance frequency 1100 Hz. Set the Order to **10000** and Window type "**Hamming**". Finally press **Accept** to apply the filter.

Step 2: Post-processing

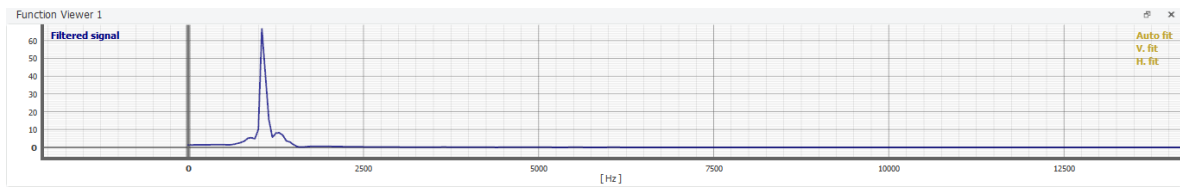
⚠ Please note: The higher is the order and the longer the computation of the filter, therefore it may take a while before you can apply higher order filters.



Once the filter is applied, you can see that the time evolution of the static pressure is cleaner and only contains 1 frequency (the main frequency). All the other frequencies have been removed.



You can also check it in the Frequency Space (**Right click** in *Function Viewer* > **Data management** > **Set graph to [Frequency space] mode**), where the main harmonic is represented filtering all the other harmonics:



⚠ Please note: For more information about the filters formulations, please check the User Guide section **Post-processing > Function viewer > Data management**.

2.2.4 Sound Pressure Level

In acoustics it is more relevant to work with Sound Pressure Level rather than with pressure in Pascal to measure the level of noise. The Sound Pressure Level (SPL) is in decibels (dB) unit and it is defined by the following relation:

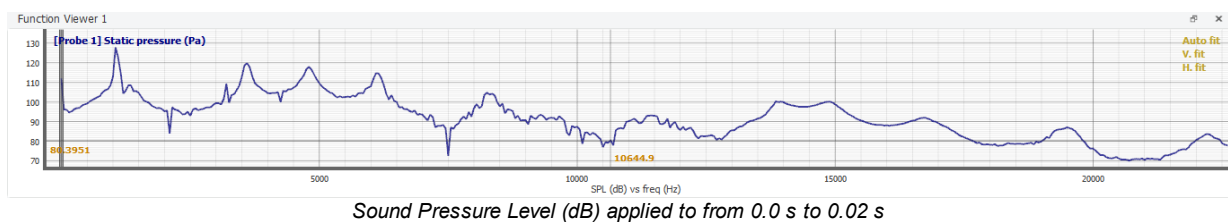
$$SPL = 10 \log_{10} \left(\frac{p^2}{p_{ref}^2} \right)$$

where p_{ref} is the reference pressure equal to $2 \cdot 10^{-5}$ Pa (i.e. threshold of human hearing).

To plot the SPL in frequency space one must:

- Plot a temporal pressure signal such as the pressure read at probe: **Right click** in *Function Viewer* > **Probes > Probe 1 > Static Pressure**.
- Plot the SPL in frequency space: **Right click** in *Function Viewer* > **Data management > Set graph to [SPL vs freq] mode**.
- Leave the window type to none and the time range from 0 to 0.02 s, and click **Accept**.
- Zoom in the frequency range [0; 20000] Hz which is the audible range and rescale the Y axis.

The Function Viewer will plot the following graph:



The mean harmonic peak at 1100 Hz is clearly shown and reaches an amplitude of noise of 130 dB approximately. Other harmonics seem to appear as well at higher frequencies. Mean noise seems around 90-100 dB.

As shown above, the pressure signal presents a transient period from 0 to 0.01 s and periodic oscillations after 0.01 s. For a better signal processing, it is recommended to remove this transient period as it may generate additional spurious noise and only consider the periodic part of the signal.

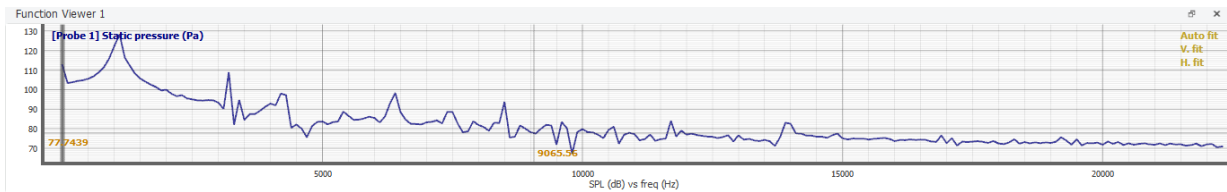
- Plot again the SPL graph to show the window type and time range to apply to the temporal signal

Step 2: Post-processing

again: **Right click** in *Function Viewer* > **Data management** > **Set graph to [SPL vs freq] mode**.

f) Change now the parameter "From time" to 0.01 s instead of 0.0 s. Click **Accept**.

The SPL is similar to the previous one, but some of the secondary peaks have lowered in intensity and the signal is more consistent.

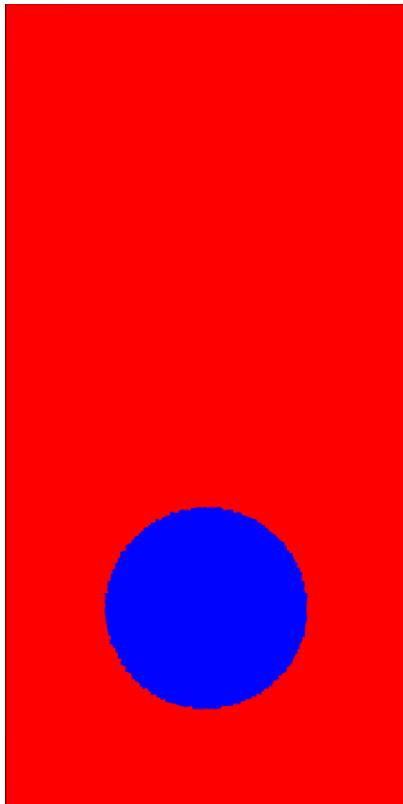


Sound Pressure Level (dB) applied to from 0.01 s to 0.02 s

⚠ Please note: For more explanations about this case, please check the Validation Guide, section **Helmholtz resonator: Flute**.

Tutorial 13 - Rising bubble

This tutorial illustrates the setup and solution of a bubble rising problem, which involves the simulation of the interaction between two different phases. The specific problem is relative to a fluid inclusion (bubble) rising in another fluid, as described by S. Hysing [*"Quantitative benchmark computations of two-dimensional bubble dynamics"*, International Journal for Numerical Methods in Fluids 2009, 60, pag. 1259-1288]. The inclusion and the surrounding fluids (blue and red color respectively in the picture below) are initially at rest. XFlow's Phase field multiphase solver will be used in this specific analysis.



Initial fluid 1 and 2 concentration.

This tutorial shows how to:

- Create simple geometries, such as boxes.
- Set a multiphase Phase field problem.
- Define liquid region.
- Set surface tension model.
- Make an animation.

- Define a custom field using the *vof* variable.
- Define a volume integral and visualize the rise velocity in the function viewer.

It is assumed that the reader has completed Tutorial 01, 02, 04 and 05. Some steps in the setup and post-process will not be described in detail.

Contents

[Step 1: Problem setup](#)

[Step 2: Post-processing](#)

Step 1: Problem setup

1.1 Engine settings

In **Project Tree > Environment > Engine** set the following parameters:

- (a) **Kernel:** 2d
- (b) **Flow model:** Multiphase
- (c) **Multiphase model:** Phase field
- (d) **Analysis type:** Internal
- (e) **Turbulence settings > Turbulent model:** Automatic

The **Phase Field** solver is chosen in this specific analysis (over the **Particle-based tracking** alternative) because of the important effect of the surface tension in determining the shape and rise velocity of the bubble: *"surface tension effects are strong enough to hold the bubble together and thus we do not expect any break up in this test case"* (S. Hysing, 2009). In fact, the relatively low value of the **Eotvos number**

$E_0 = \frac{4 \rho_1 g r^2}{\sigma}$ (also called the **Bond number**), equal to 10, indicates that the surface tension effects are as important as the gravitational forces. In the previous equation ρ_1 refers to the surrounding heavier fluid

density, r is the inclusion radius and σ the surface tension. In such applications, where an important effect of the surface tension is expected, it is usually advised to use the **Phase Field** solver.

1.2 Environment settings

In **Project Tree > Environment > Environment** set the following parameters:

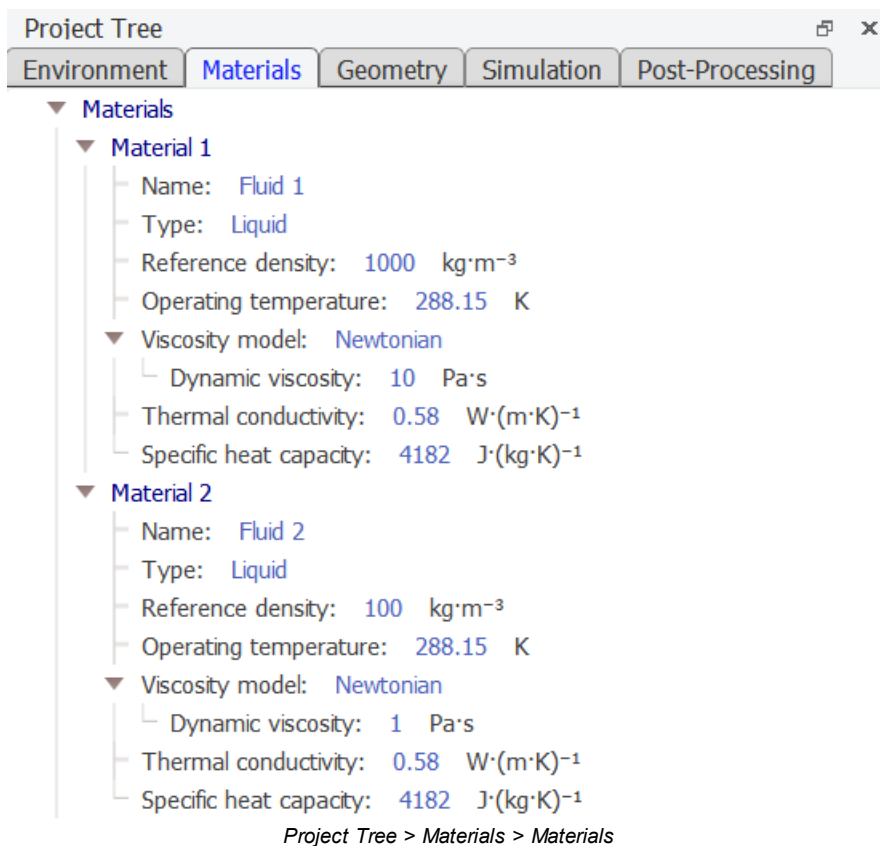
- (a) **Global attributes > Ext. acceleration laws:** (0, -0.98) m.s⁻², to set the Eotvos number value = 10.

- (b) Leave the **Initial conditions** as **User defined**, and keep the **Initial velocity field** as **(0, 0) m.s⁻¹**
- (c) **Liquid regions > Fluid 1 initial concentration: $(x^2+y^2>0.0625)(1)$** , to define the initial region of the surrounding fluid (inclusion radius $r=0.25$). The syntax **()()** is equivalent to the Boolean operation AND, and it is here used to express an if statement. In particular, the specified initial concentration expresses the intersection of a constant function **1** with the cylindrical region defined by **$x^2+y^2>0.0625$** . The result is, hence, a function which value is 1 only within the cylindrical region and 0 everywhere else. The same concentration could be expressed as **if($x^2+y^2>0.0625$,1,0)**.

1.3 Material settings

In **Project Tree > Materials > Materials** set the physical characteristic of both fluids. These

settings provide a Reynolds number $Re = \frac{\rho_1 \sqrt{g} (2r)^{\frac{3}{2}}}{\mu_1}$ equal to 35.

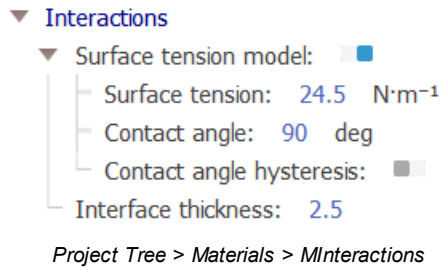


In **Project Tree > Materials > Interactions** the **Surface tension model** is always **On** when the **Multiphase model: Phase field** is selected. Set the following parameters:

- (a) **Surface tension: 24.5 N.m⁻¹**

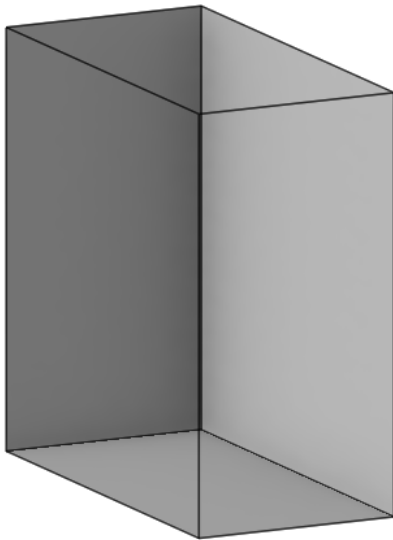
Step 1: Problem setup

- (b) **Contact angle: 90 deg**, leave default.
- (c) **Contact angle hysteresis: Off**
- (d) **Interface thickness: 2.5**, number of lattice elements used to model the interface thickness.



1.4 Create a geometry

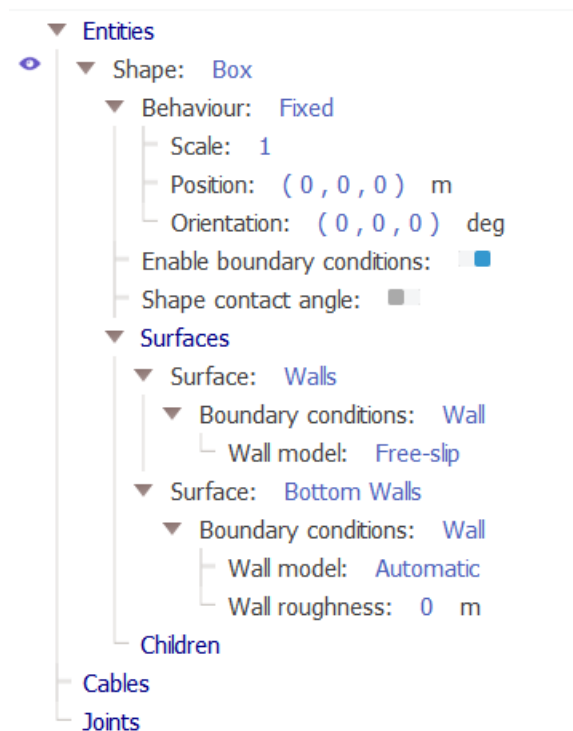
Create the cavity as a box () with lower corner **(-0.5, -0.5, -1)** and upper corner **(0.5, 1.5, 1)**. Check with the help of the **Back-face culling** that the box normals are orientated towards the interior.



1.5 Geometry settings

In **Project Tree > Geometry > Entities > Shape: Box** set the following parameters:

- (a) **Behaviour: Fixed**
- (b) **Boundary conditions: Wall**. Set the Box bottom wall as **Wall model: Automatic** whilst setting the other walls as **Wall model: Free-slip**



1.6 Simulation settings

In **Project Tree > Simulation** set the following parameters:

- (a) **Time > Simulation time: 5 s**
- (b) **Time > Time step mode: Fixed custom**, with **Time step: 4e-5 s**
- (c) **Lattice discretization > Resolved scale: 0.00625 m**, with the **Refinement algorithm: Disabled**
- (d) **Lattice discretization > Options > Seed point: Automatic**
- (e) **Store data > Frames frequency: 50 frames per second (Hz)**
- (f) Leave Off **Save averaged fields**
- (g) Leave On all the **Fields to save**

1.7 Launch the calculation

- (a) Save the project
- (b) Set the number of CPUs in **Main menu > Options > Preferences > Engine**
- (c) Press **Run** button > **Start computation**

Step 2: Post-processing

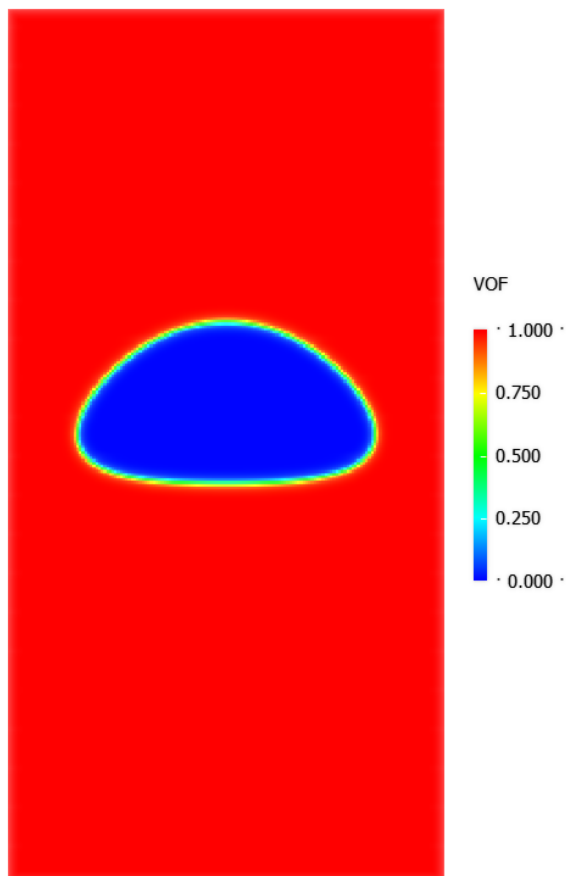
2.1 Load data

Load data: **Simulation data > Load simulation data** or 

2.2 Visualize the evolution of the bubble shape.

In **Post Processing > Cutting planes**

- (a) Create a **Cutting plane** on the Z axis
- (b) Choose **Visualisation mode: 3d field**
- (c) Choose **Field: VOF**
- (d) Select different frames in the time bar to visualize the evolution of the bubble shape

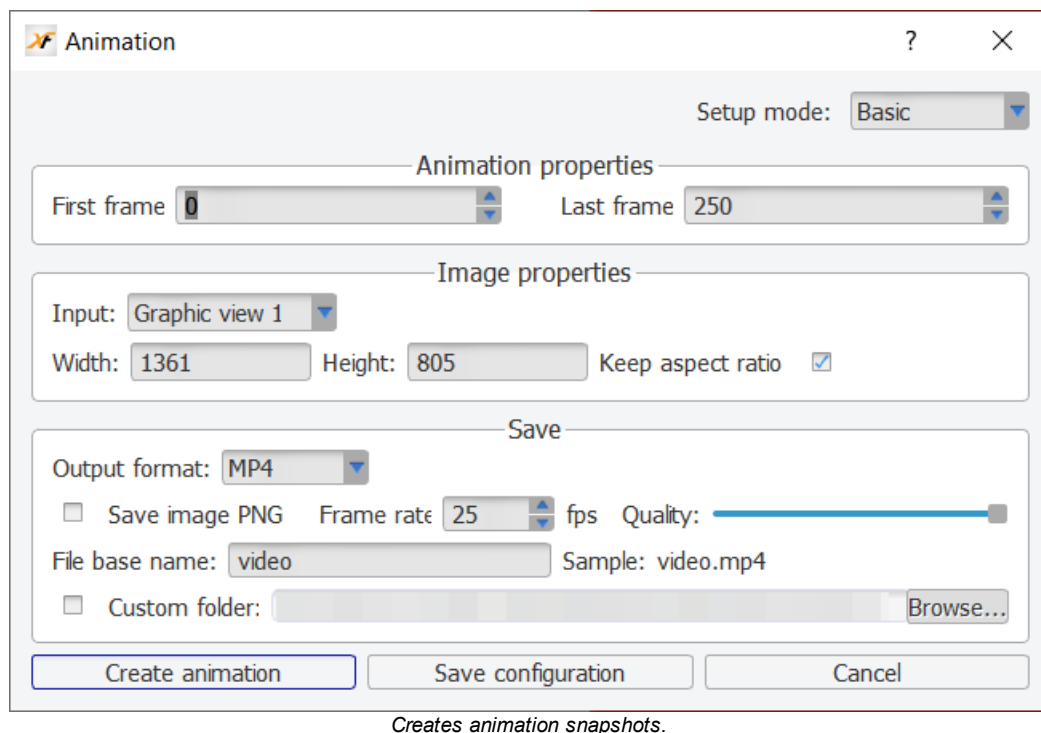


Bubble shape at t=3 s

2.3 Create a basic animation

- (a) **Main menu > Post-Processing > Animation**

- (b) Select **Setup mode: Basic**
- (c) Select **First frame** and **Last frame** for the sequence, e.g. 0 and 250
- (d) Select the **Input:** "Graphic View 1" to create a series of snapshots of the Graphic View, or "GUI view" to create images of the entire GUI
- (e) Use MP4 output format, frame rate of 25 fps, and highest quality
- (f) Enter a **File base name** to define the video filename, and leave unchecked the **Custom folder** in order to generate the animation files in the simulation folder. Press **Create Animation** in order to generate the animation. It will create the video in the directory C:/Users/XFlow/Desktop/Tutorial14/RisingBubble/animation.



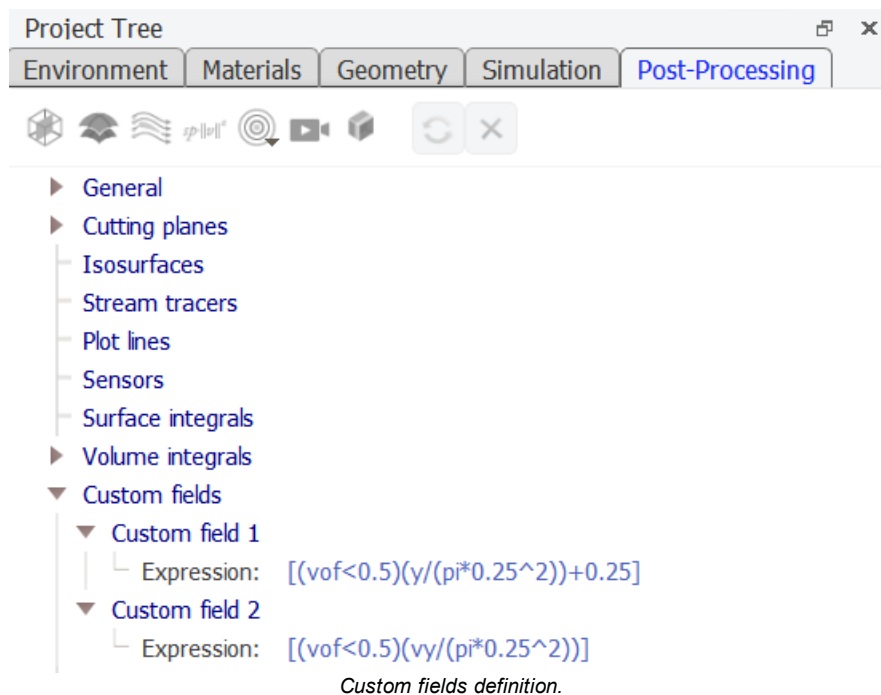
2.4 Create a custom field

- (a) To create a custom field go to **Project Tree > Post-Processing**, do **right-click** on **Custom fields** and choose **Add custom field** or press  in the Post-Processing Toolbar
- (b) In this specific case the following system variable will be used:
 - vof: volume of liquid phase representing the fluid 1 defined in the liquid regions.
 - vy: the Y-component of velocity at a discrete point of the domain, given by coordinates (x,y,z).
 - y: spatial coordinate
- (c) The user can define the custom field in **Custom Fields**, in this tutorial two custom fields are defined.

Step 2: Post-processing

- 1) To track the translation of the bubble the center of mass will be calculated as: **Custom Fields> Custom field 1 > Expression: $(vof < 0.5)(y/(pi * 0.25^2)) + 0.25$** . This means that if $vof < 0.5$ (bubble fluid region), the y-coordinate of the center of mass of the bubble is evaluated as $y/(pi * 0.25^2)$, otherwise the function will be set to 0. Furthermore, a constant value of 0.25 is added in order to easily compare the obtained results with the reference literature (S. Hysing, 2009).
- 2) To calculate the mean velocity with which the bubble is rising, defined as rise velocity, the following function will be used: **Custom Fields> Custom field 1 > Expression: $(vof < 0.5)(vy/(pi * 0.25^2))$** . This means that if $vof < 0.5$ (bubble fluid region), the mean bubble velocity is equal to $vy/(pi * 0.25^2)$, otherwise the velocity is 0.

Both those custom fields will be used as argument of a volume integral in the next session to calculate a scalar value.

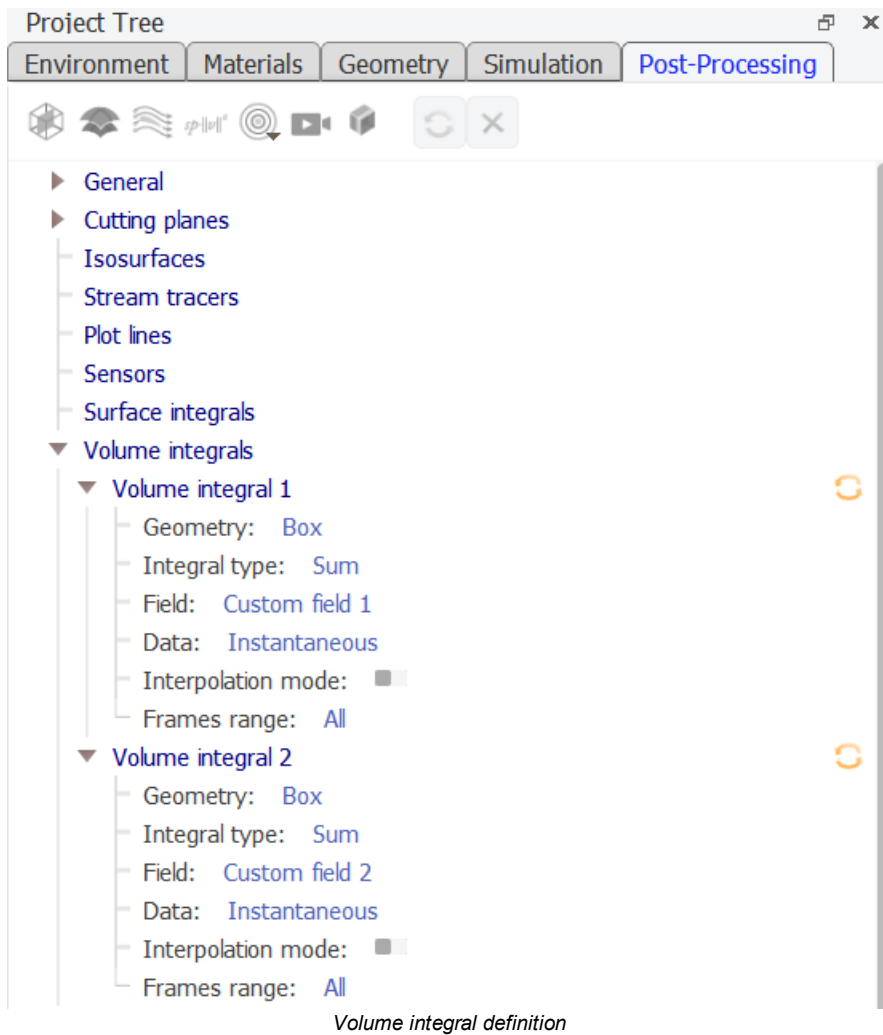


Custom fields definition.

2.4 Define a Volume integrals

To evaluate the rising speed and the evolution of the center of mass a volume integral has to be defined for each custom field earlier created.

To define a volume integrals go to **Project Tree > Post-Processing**, do **right-click** on **Volume integrals** and choose **Add volume integral**. In this tutorial two volume integrals will be added to evaluate both the bubble rising speed and its center of mass evolution.

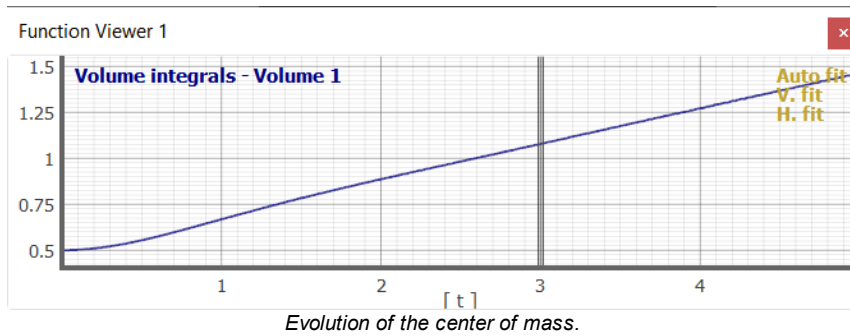


(a) Refresh the volume integrals with 

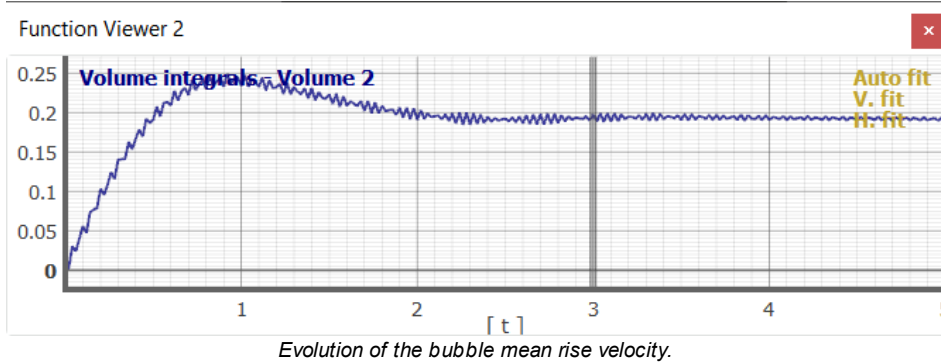
 **Tip:** It is possible to refresh more than 1 volume integral at a time by using a multiple selection (hold Ctrl key) and hit the Refresh button  at the top of the Post-Processing tab

(b) Display the Function Viewer window and measure the evolution of the center of mass in volume integral: **right click** on *Function Viewer 1* to show the Function Viewer menu, then choose **Volume integrals > Volume 1**. The curve illustrates that after an initial acceleration phase (parabolic trend up to $t = 0.75$ s) the bubble center of mass rises at a constant speed and a linear trend is observed. A value of 1.083 m is achieved at 3 [s] simulation, which is matching closely the data reported in the literature (S. Hysing, 2009) of 1.0813.

Step 2: Post-processing

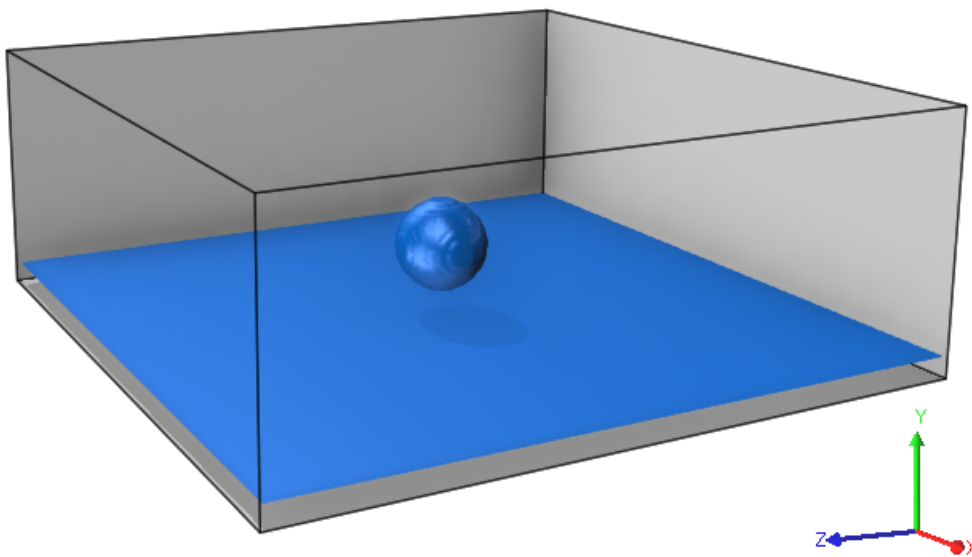


(c) Display the Function Viewer window and measure the rising speed in volume integral: **right click** on *Function Viewer 1* to show the Function Viewer menu, then choose **Volume integrals > Volume 2**. The plot shows that a maximum velocity of 0.243 is reached at $t = 0.88$ s, after which the velocity decreases and reaches a constant value of 0.194. This is consistent with the center of mass evolution described earlier, and again the data match closely the reference values of 0.2417 maximum velocity at $t = 0.92$ s (S. Hysing, 2009).



Tutorial 14 - Crown splashing

This second tutorial on multi-phase flows illustrates the setup and solution process of a droplet impacting on a thin film. The droplet has a radius of 2.5 mm and impacts the water film at a speed of 2.14 m/s. The film layer thickness is 1 mm. XFlow's Particle-based tracking multiphase solver will be used for the solution of this specific case. The data are compared to the following reference paper from Lebaigue, O and Jamet, D and Lemonnier E: *Test cases for interface tracking methods: methodology and current status.*^[1]



This tutorial shows how to:

- Create simple geometries, such as boxes.
- Set a multiphase problem using the Particle-based tracking method.
- Define initial velocity field.
- Define liquid region.
- Set surface tension model.
- Show particles.
- Create an isosurface.
- Make an animation.

It is assumed that the reader has completed Tutorial 01, 02, 04, 05 and 13. Some steps in the setup and post-process will not be described in detail.

Contents

[Step 1: Problem setup](#)

[Step 2: Post-processing](#)

Step1: Problem setup

1.1 Engine settings

In **Project Tree > Environment> Engine** set the following parameters:

- (a) **Kernel:** 3d
- (b) **Flow model:** Multiphase
- (c) **Multiphase model:** **Particle-based tracking**. The particle-based tracking multiphase model is recommended when inertial effects are more important than surface tension effects, which is the case in the present simulation given the droplet impacts at high speed the water film.
- (d) **Analysis type:** Internal.
- (e) **Turbulence settings > Turbulent model:** Automatic

1.2 Environment settings

In **Project Tree > Environment > Environment** set the following parameters:

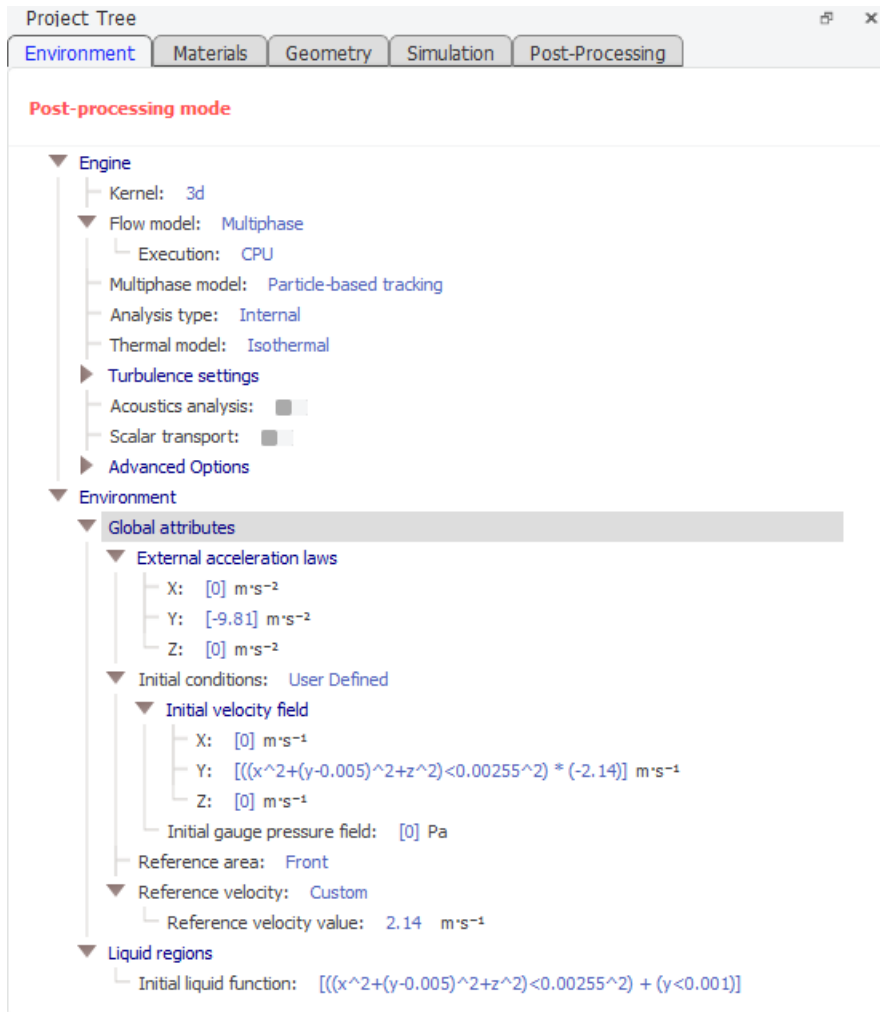
- (a) **Global attributes > Ext. acceleration laws:** **(0, -9.81, 0) m.s⁻²**, to take into account gravity.
- (b) Leave the **Initial conditions** as **User defined**, and set the Y-component of the **Initial velocity field** as:

$$((x^2+(y-0.005)^2+z^2) < 0.00255^2) * (-2.14) \text{ m.s}^{-1}$$

The multiplication operator "*" works as a Boolean function "**AND**" and the conditions are equal to 1 when the statement is true. Therefore this means if the condition defined by the equation $(x^2+(y-0.005)^2+z^2)<0.00255^2$ is verified, the initial velocity is equal to -2.14 m.s^{-1} in the y-direction. Hence, in the region of the droplet, the velocity will be initialized to -2.14 m.s^{-1} .

- (c) Set **Reference velocity** as **Custom**, and set the **Reference velocity value** as **2.14 m.s⁻¹**.
- (d) **Liquid regions > Fluid 1 initial function:** $((x^2+(y-0.005)^2+z^2)<0.00255^2) + (y<0.001)$, to define the initial water region (droplet + film region). The addition operator "+" works as a Boolean function "**OR**" therefore the Fluid 1 will be initialized in regions defined by the equation $(x^2+(y-0.005)^2+z^2)<0.00255^2$ or in the regions defined by $y<0.001$. The rest will be equal to zero, which means

no Fluid 1 is present in any other regions.



Project Tree > Environment

1.3 Material settings

By default in multiphase analyses the default **Fluid1** is water and the default **Fluid2** is air. We will keep those two fluids by default.

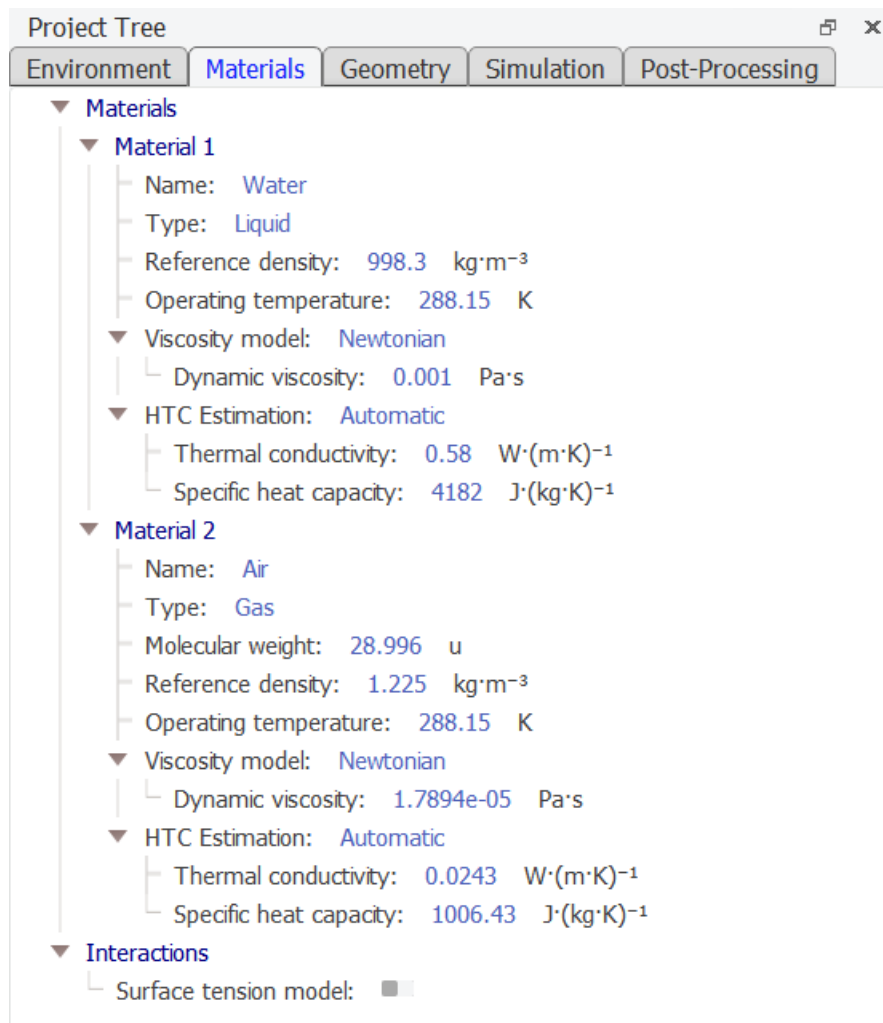
In **Project Tree > Materials > Materials** set the names of both fluids.

(a) **Fluid1 > Name: Water**

(b) **Fluid1 > Name: Air**

Leave the default **Water** and **Air** properties.

Step1: Problem setup



Project Tree > Materials > Materials

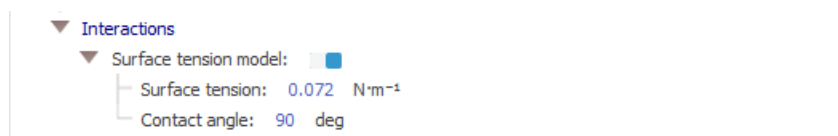
In **Project Tree > Materials > Interactions** set the following parameters.

(a) Enable **Surface tension model**: **On**

and leave by default:

(b) **Surface tension**: **0.072 N·m⁻¹**

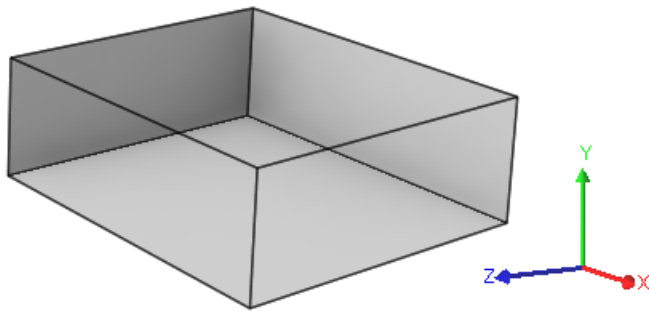
(c) **Contact angle**: **90 deg**



Project Tree > Materials > Interactions

1.4 Create a geometry

Create a box () with lower corner **(-0.02, 0, -0.02)** and upper corner **(0.02, 0.015, 0.02)**. Check with the help of the **Back-face culling** that the box normals are orientated towards the interior.



Domain shape

1.5 Geometry settings

In **Project Tree > Geometry > Entities > Shape: Box** set the following parameters:

- (a) **Behaviour: Fixed**
- (b) **Boundary conditions: Wall**, with **Wall model: Automatic**

1.6 Simulation settings

In **Project Tree > Simulation** set the following parameters:

- (a) **Time > Simulation time: 0.02 s**
- (b) **Time > Time step mode: Fixed custom**, with **Time step: 1e-6 s**
- (c) **Lattice discretization > Resolved scale: 0.00025 m**, with the **Refinement algorithm: Disabled**
- (d) **Lattice discretization > Options > Seed point: Automatic**
- (e) **Store data > Frames frequency: 10000 frames per second (Hz)**
- (f) Leave off **Save averaged fields**
- (g) Enable **Compute markers**
- (h) Leave On all the **Fields to save**

1.7 Launch the calculation

- (a) Save the project

 **Tip:** You can directly load the setup of this problem from the project file `CrownSplasing.xfp`

- (b) Set the number of CPUs in **Main menu > Options > Preferences > Engine**
- (c) Press **Run** button > **Start computation**

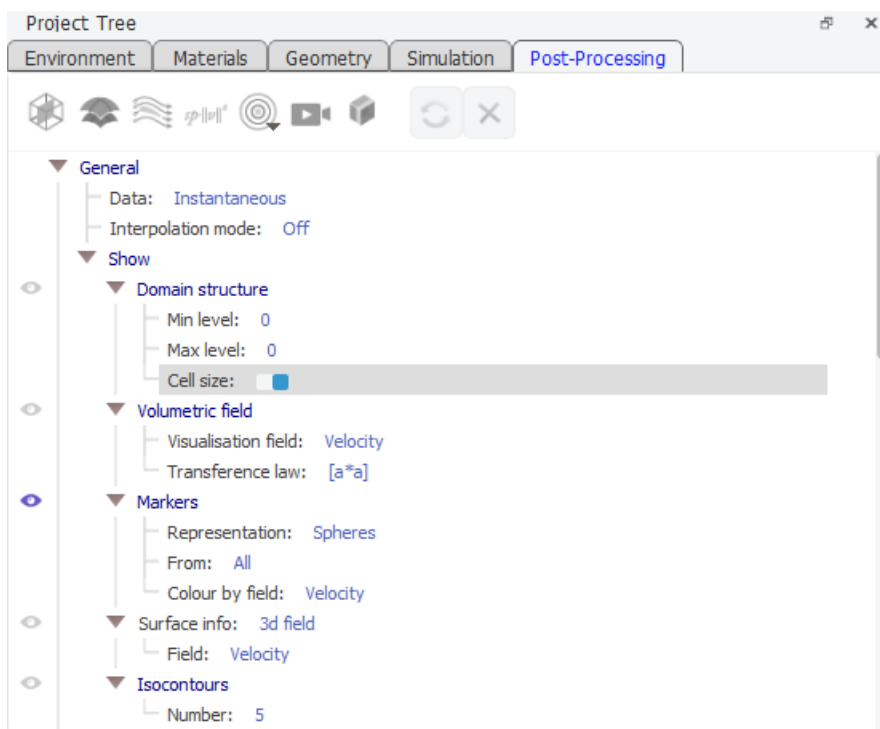
Step2: Post-processing

2.1 Load data

Load data: **Simulation data > Load simulation data** or 

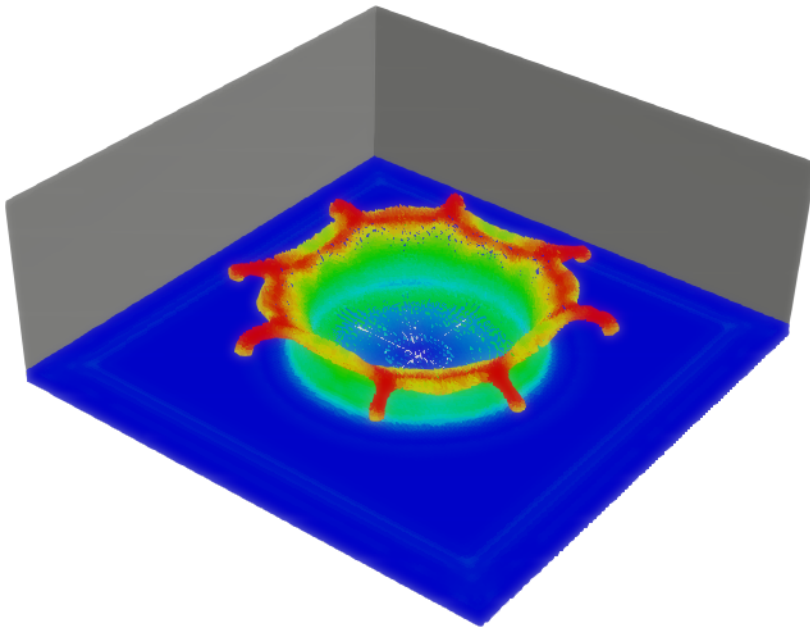
2.2 Show water particles

Click on the icon  at **Post processing > General > Show > Markers**



Post-processing > Markers

- (a) To show the particles of the water phase switch on the **Markers**.
- (b) To represent the particles with different shapes, go to the last frame, and select Arrows and Sphere in **Markers > Representation**.
- (c) Play the simulation by pressing 

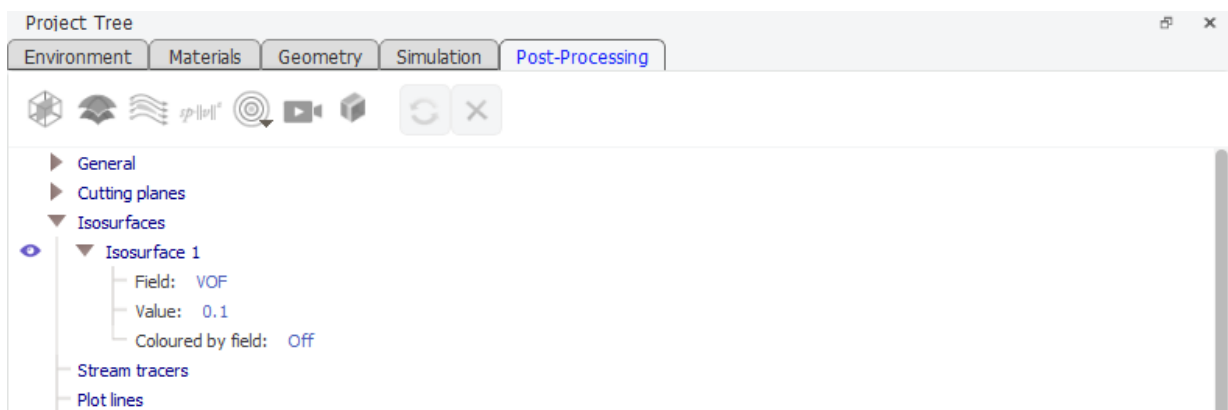


Water markers colored by velocity at $t = 0.01$ s

The crown formation can be appreciated as per the reference data [1].

2.3 Create an isosurface of the interface

(a) Create an isosurface **Right clicking** on **Post-Processing > Isosurfaces** and **Add isosurface** with:



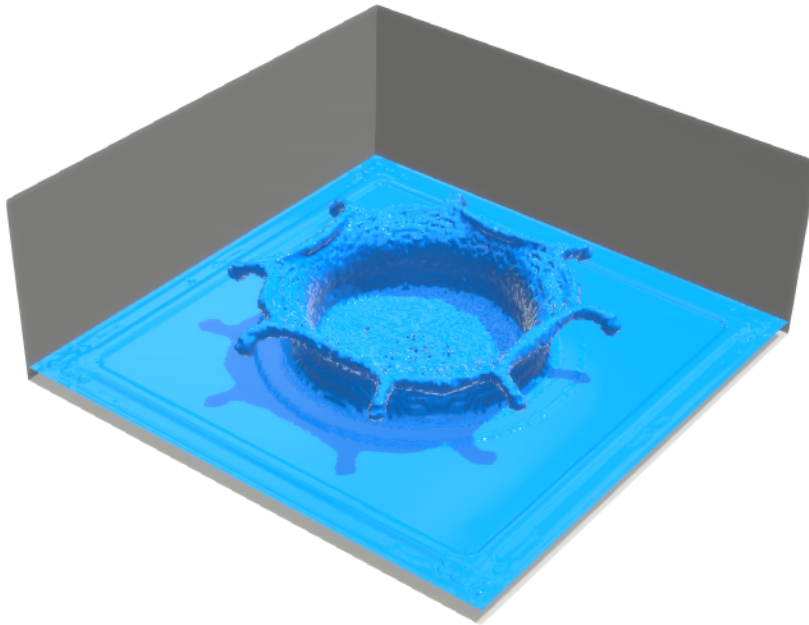
Post-processing > Isosurfaces

The isosurface will represent the water interface since the isosurface is created from a low volume of liquid phase value (0.1).

(b) Change the visualisation material of the Isosurface to get visual effect similar to water: Select the Isosurface in the **Graphic View 1 > Right click > Set visualisation material**. Select the **Plastic** material

Step2: Post-processing

type and a blue colour:



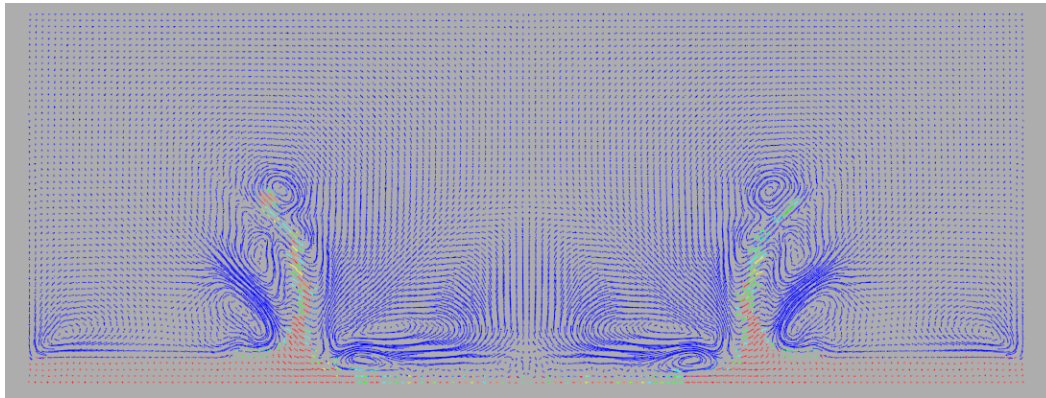
Isosurface of Volume of liquid phase at $t = 0.01$ s

(c) Select the isosurface in the Project Tree. Export this isosurface as a geometry (.stl) in the **Main menu > Simulation data > Export selected isosurfaces**. You can measure the diameter of the crown at each time step in an external CAD software for instance. The crown diameter can be compared with the data from [1].

2.4 Visualize the second phase

The difference between a free-surface simulation and a multiphase simulation is the latter includes a second phase. The second phase in the present case is the air and is also computed to interact with the water. It becomes interesting to see the air entrainment induced by the water splashing, and also the impact it has on the crown formation.

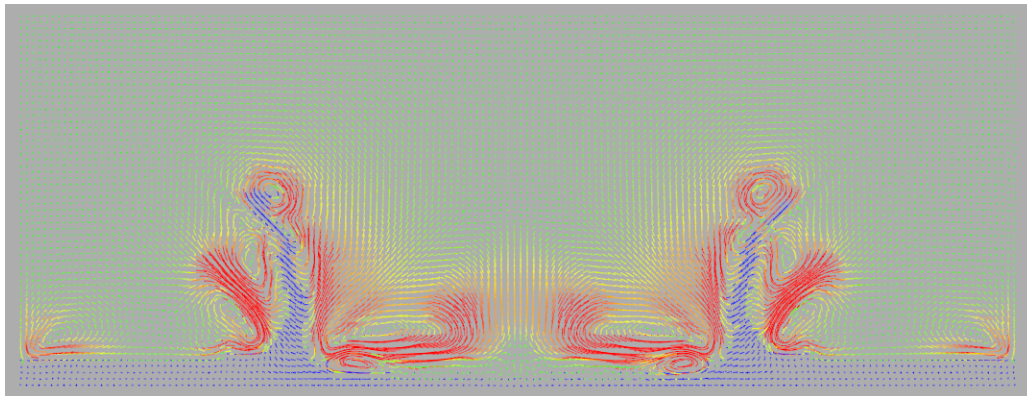
- (a) Create a cutting plane by clicking on the cutting plane icon  (or right-click on **Cutting planes > Add cutting plane**)
- (b) Set **Visualization mode: Vectors** in order to visualize the vectors on the plane
- (c) Adjust the **Arrows density** and the **Arrow length** to 1.0.
- (d) Select the **Field: Volume of liquid phase** in order to appreciate the two phases.



Vectors colored by Volume of liquid phase at $t = 0.01$ s

The recirculations can be appreciated especially around the crown fins. In order to check the velocity patterns for the air phase only, we can take benefit of the custom fields and the Volume of liquid phase field.

- (e) Create a custom field by clicking on the Custom fields icon  (or right-click on **Custom fields > Add custom field**)
- (f) Set the custom field law to: **if(vof<0.1,vmod,-999)**. This allows to discard the water phase ($\text{vof} > 0.1$) setting the velocity to a high negative value of -999.
- (g) Display the custom field by selecting **Field: Custom field 1** in the **Cutting plane 1**.
- (h) Adjust the range from -1 to 1 by clicking on the maximum and minimum values of the color bar. This allows to appreciate the values of the water as a blue color, the fluid at rest as a green color, and the air velocity components with the rest of the colors.



Vectors colored by Custom field 1 at $t = 0.01$ s

Higher velocity regions can be appreciated around the crown fins indeed especially on the inner side of the crown. This is the velocity induced by the water droplet splashing.

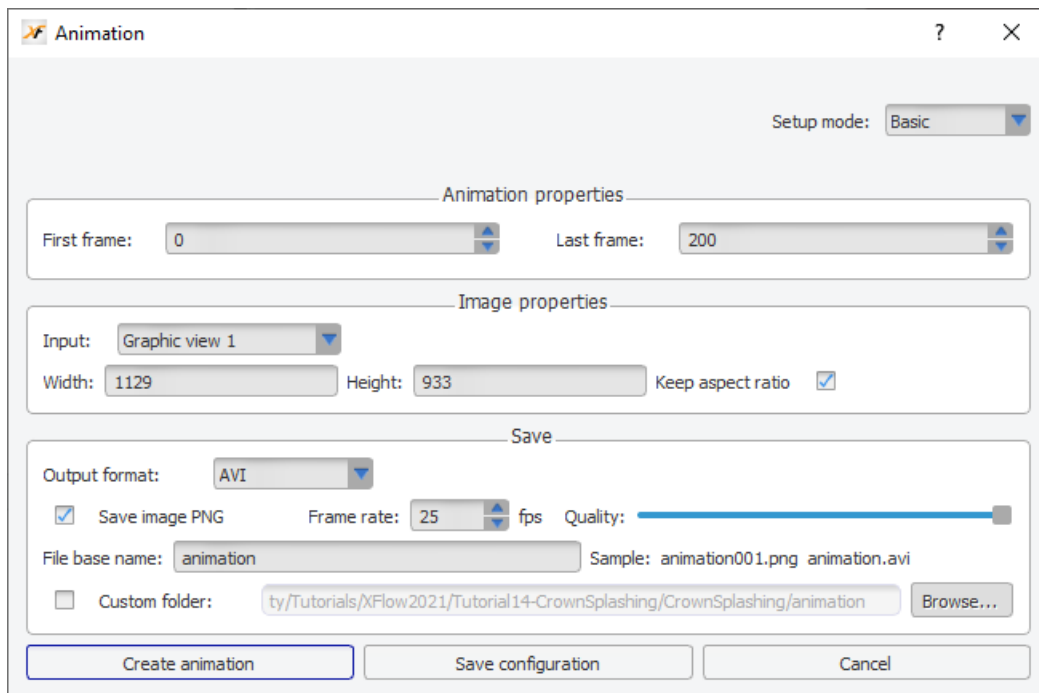
2.5 Create a basic animation

As in the previous multi-phase case we will make an animation to better visualize the evolution of the

Step2: Post-processing

splashing.

- (i) **Main menu > Post-Processing > Animation**
- (j) Select **Setup mode: Basic**
- (k) Select **First frame** and **Last frame** for the sequence, e.g. 0 and 200
- (l) Select the **Input**: "Graphic View 1" to create a series of snapshots of the Graphic View, or "GUI view" to create images of the entire GUI
- (m) Select the video **Output format** as .AVI and switch on the option to **Save image PNG**
- (n) Enter an **File base name** which will be used as a basename for both the video output and the image sequence, the path of the **Folder** where to save it and press **Create Animation**. It will create an output video at 25 FPS (.AVI, .MP4, or .OGV) and 200 images (.PNG format) in the directory `$Working-dir/CrownSplashing/animation`. If required, the image sequence can be used in an third-party software to create a more customized animation.



Tutorial 15 - Abaqus co-simulation: Flexible bar Dam-break

This tutorial illustrates how to set up and run an XFlow / Abaqus coupled simulation. The specific case here illustrated constitutes of a 3D dam-break simulation which includes a flexible bar.

The tutorial shows how to:

- Setup an Abaqus simulation in Abaqus\CAE which will be used in a FSI simulation with XFlow
- Setup the correspondent XFlow simulation
- Run the co-simulation and analyse the results

It is assumed that the reader has completed Tutorial 01, 02 and 03. Some steps in the setup will not be described in detail.

 **Please note:** This tutorial requires having activated the Labs mode. The Labs mode can be activated in: **Main menu > Options > Preferences > Application mode**

 **Please note:** Please refer to section 8.3.1 from the XFlow User's Guide for compatibilities between different versions of XFlow and Abaqus.

Contents

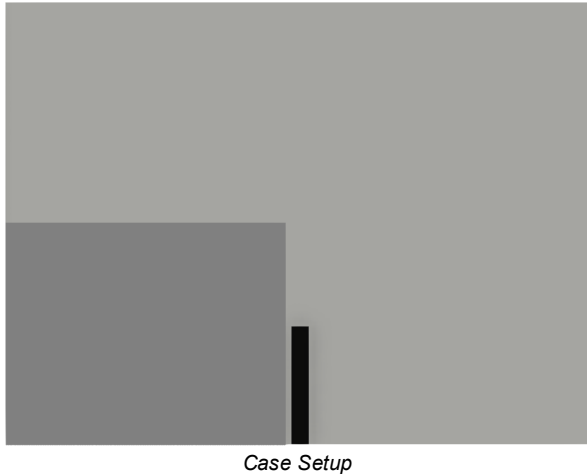
- [Step 1: Problem setup - Abaqus\CAE](#)
- [Step 2: Problem setup - XFlow](#)
- [Step 3: Execution of Co-simulation](#)
- [Step 4: Post-processing](#)

Step 1: Problem setup - Abaqus/Cae

The first step of the tutorial is the setup of the Abaqus structural analysis. The Abaqus/CAE pre-processing software is here used to setup the case and generate the Abaqus analysis file (.inp).

1.1 Definition of the Setup Geometry

The tutorial consists of a flexible bar under the load of a bam-break, as illustrated in the following figure. In this analysis the fluid is considered to be incompressible.



The dimension of the external domain are:

$$L = 0.4 \text{ [m]}$$

$$H = 0.3 \text{ [m]}$$

$$W = 0.012 \text{ [m]}$$

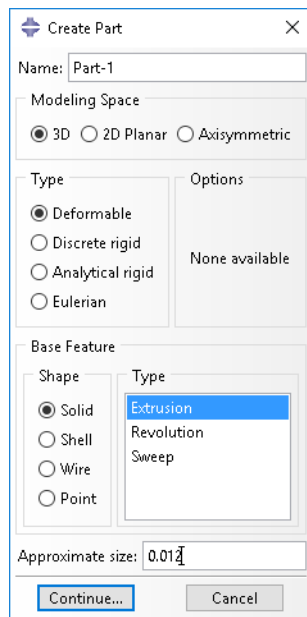
The bar geometry is defined by the following geometrical quantities:

$$l = 0.0012 \text{ [m]}$$

$$h = 0.08 \text{ [m]}$$

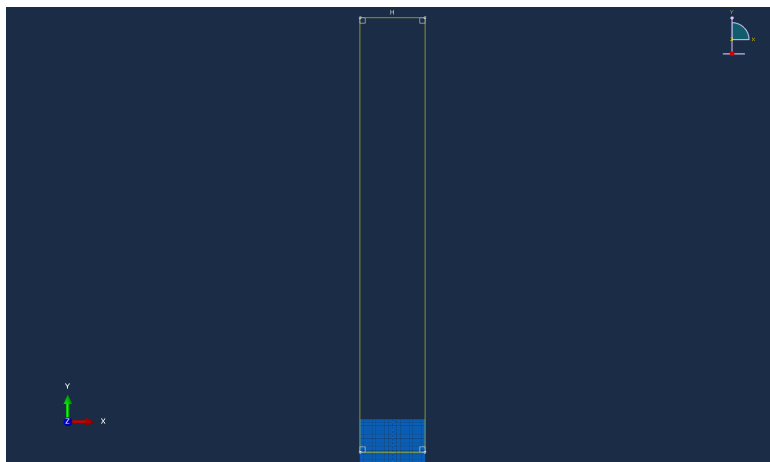
1.2 Creation of the Geometry in Abaqus\CAE

a) Create a New Part  with the following settings (Approximate size 0.012):



Creation of a new Part in Abaqus Cae.

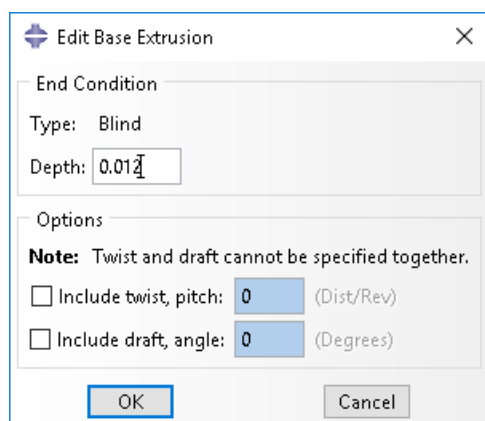
- b) Use the sketcher to create a rectangle  with the following vertices: $(-0.006, 0)$ and $(0.006, 0.08)$. Press Esc once the two vertices have been specified.



Rectangle in sketcher.

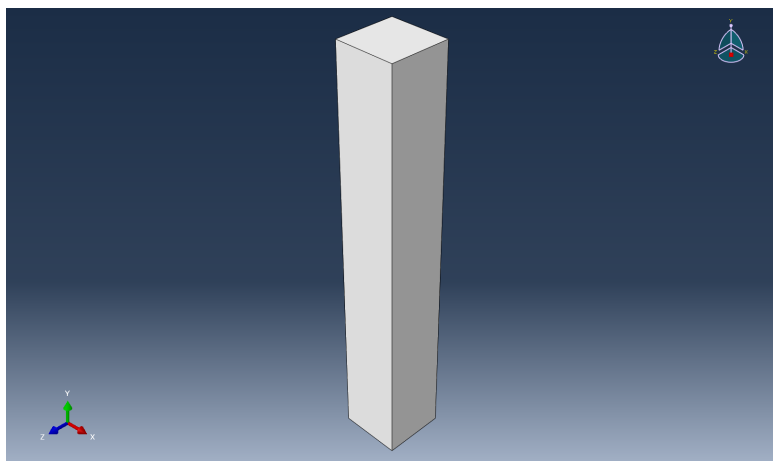
- c) Hit the Done button at the bottom of the graphic view  Edit the section sketch  . Specify an extrusion Depth of 0.012

Step 1: Problem setup - Abaqus/Cae



Extrusion Depth.

d) The bar geometry in Abaqus/CAE should look like the following snapshot:

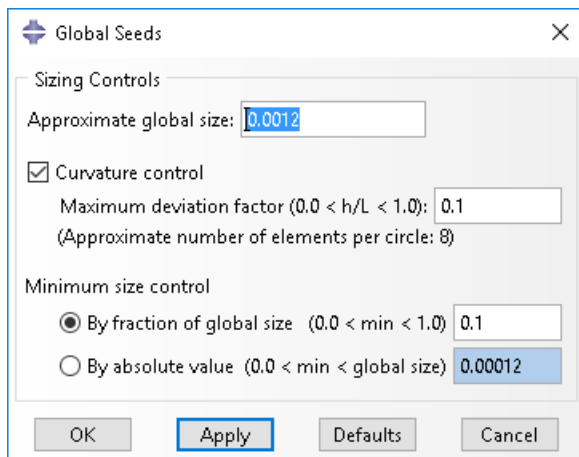


Bar geometry in Abaqus Cae.

1.3 Generation of the Structural Mesh

Under **Parts > Part-1** select **Mesh**:

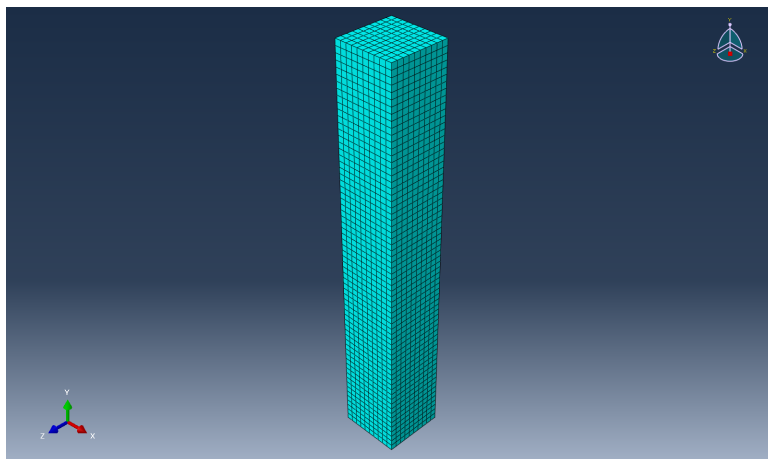
a) Seed the part  with an uniform spacing of 0.0012



Uniform seeding of geometry.

b) Mesh the part .

c) The resulting mesh should look like:



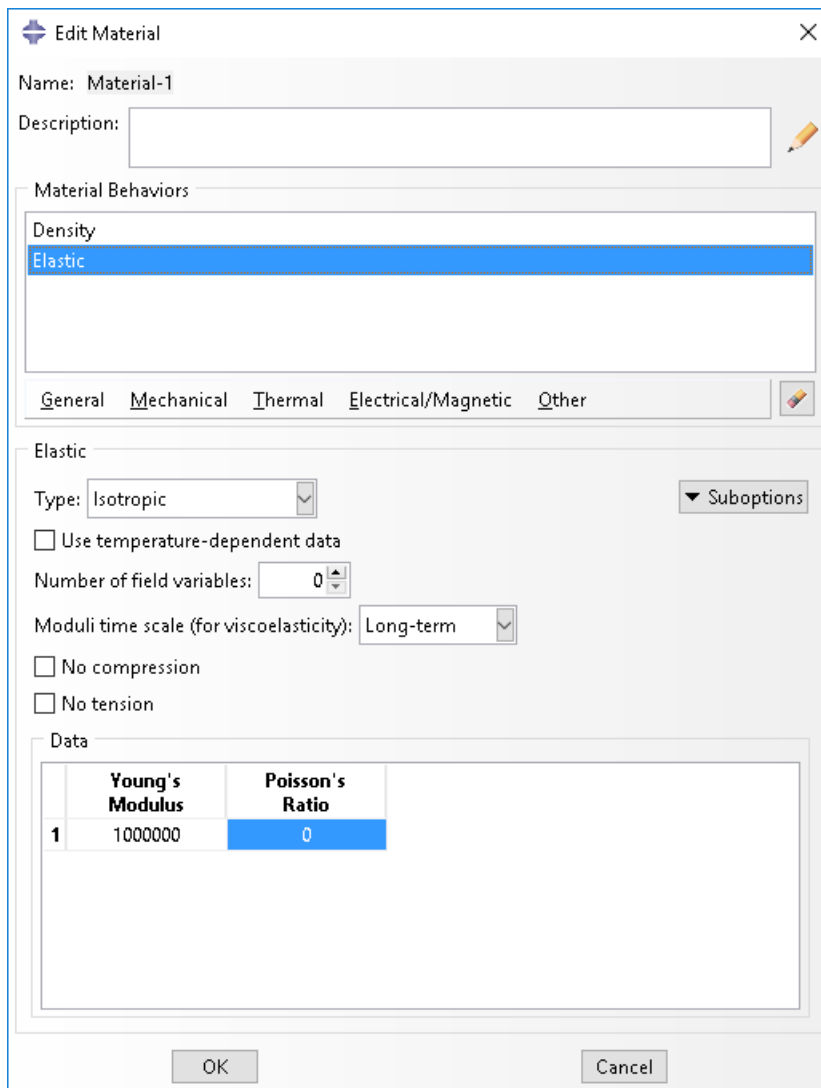
Uniform mesh of bar geometry.

1.4 Definition of Material, Elements Properties and Boundary Conditions

a) Under **Materials** create a new material  with the following parameters:

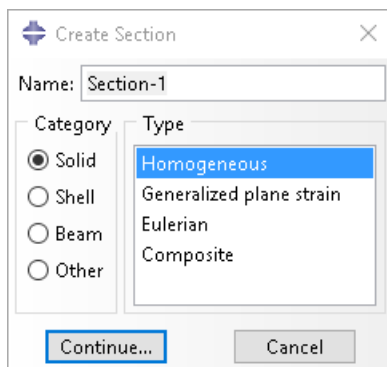
- General > Density: 2700
- Mechanical > Elasticity > Elastic > Young's Modulus: 1000000
- Mechanical > Elasticity > Elastic > Poisson's Ration: 0

Step 1: Problem setup - Abaqus/Cae



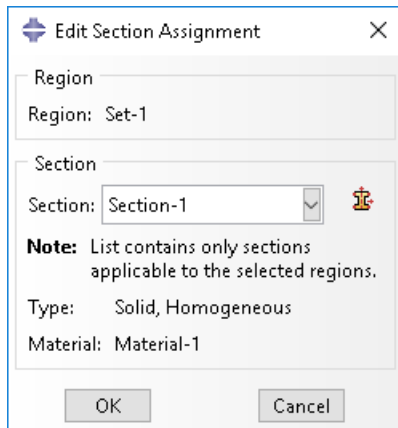
Isotropic, perfectly elastic material definition.

b) Create an Homogeneous solid section  which uses the above created Material-1



Creation of the homogeneous section.

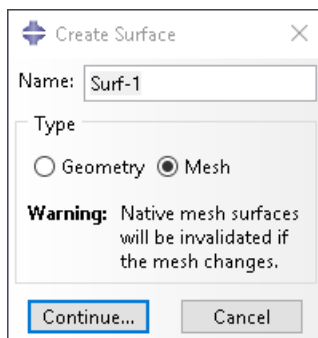
c) Assign the created section to the Part  and select the geometry from the graphical view:



Section assignment to Part.

d) Under **Assembly** create a new Assembly  with an instance of the Part-1

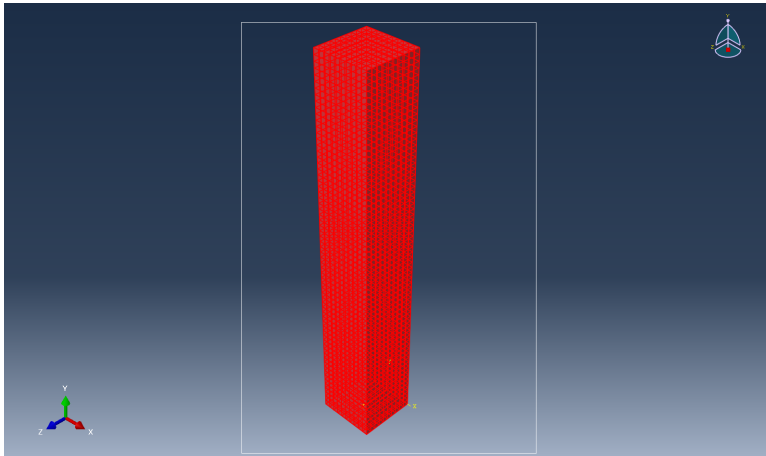
e) Under **Assembly > Surfaces** create a Mesh Surface which will be the region exchanging loads and displacements during the co-simulation:



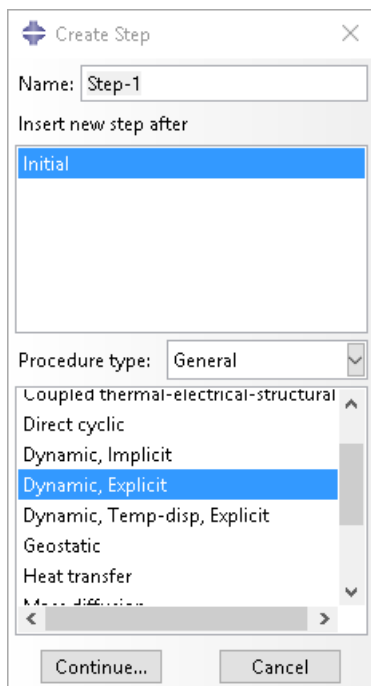
Create Surface (Mesh) where data will be exchanged during the co-simulation process.

Select all the mesh elements using a mouse drag-select. Abaqus will automatically select only surfaces elements, ignoring the inner solid elements.

Step 1: Problem setup - Abaqus/Cae



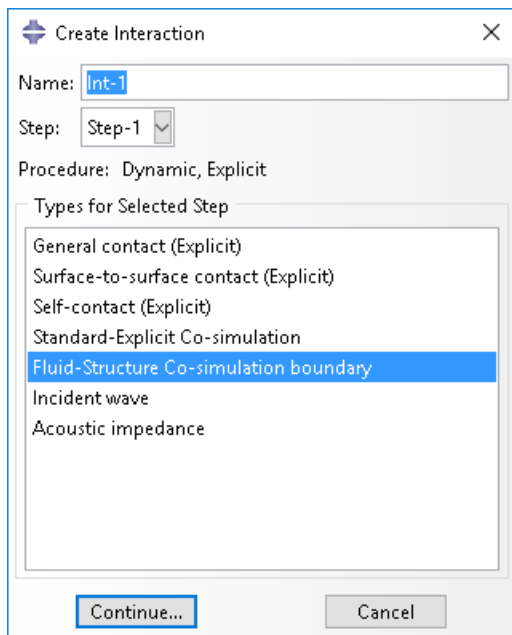
f) Under **Steps** create a new Dynamic, Explicit Step :



Creation of a dynamic explicit step.

Set a Time period of 0.5 (this will be the overall simulation time for both XFlow and Abaqus) and leave all the other settings to the Default values.

g) Under **Interaction** create a new Interaction  and select Fluid-Structure Co-simulation boundary:



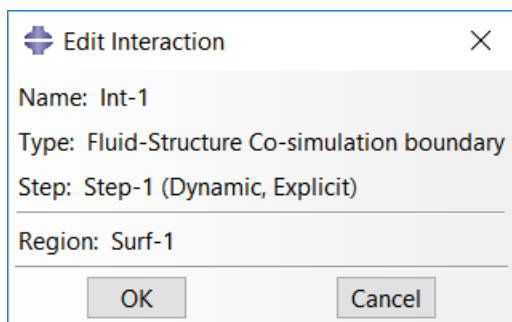
Create interaction.

⚠ Please note: This option may not be available for versions earlier than Abaqus 2018. If this the case, select any interaction type. This will be edited in the INP later on.

Select Surface and select the Surf-1 entry from the Surfaces... list on the bottom toolbar leaving the default settings.



Surface selection for interface definition.

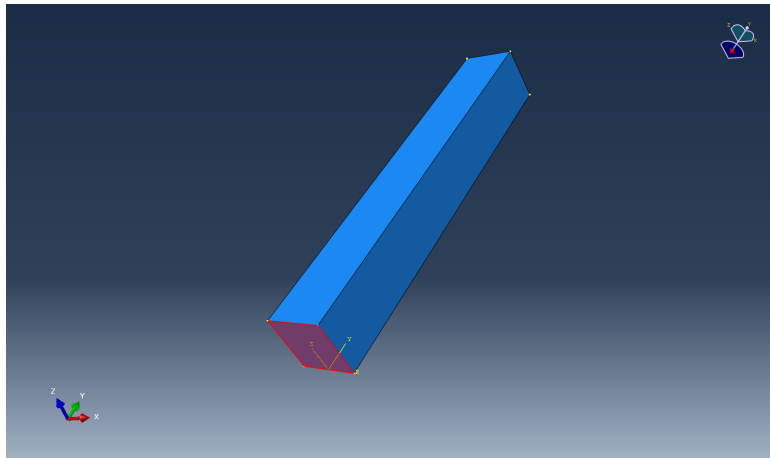
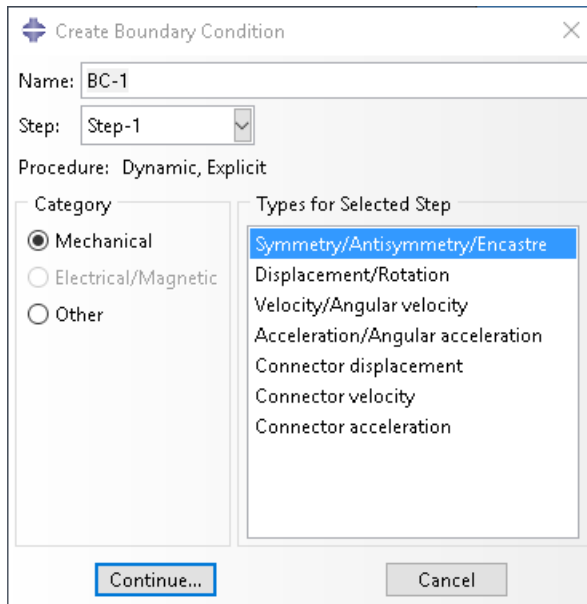


Default settings for interface.

h) Under **BCs** create new boundary conditions :

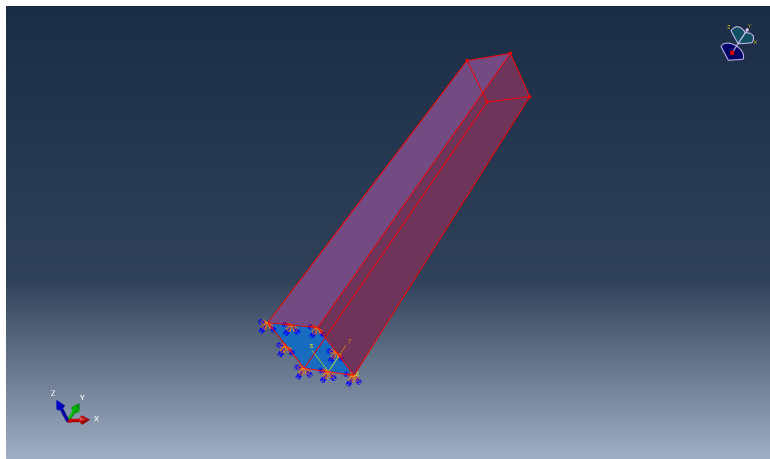
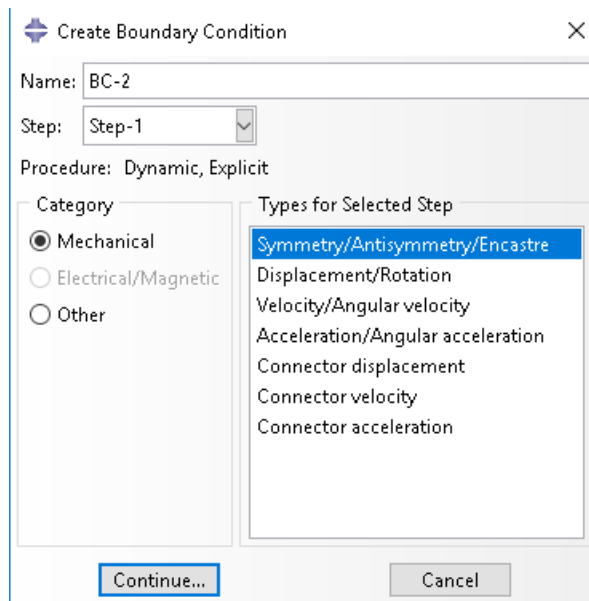
- Encaastre for the bottom side surface

Step 1: Problem setup - Abaqus/Cae

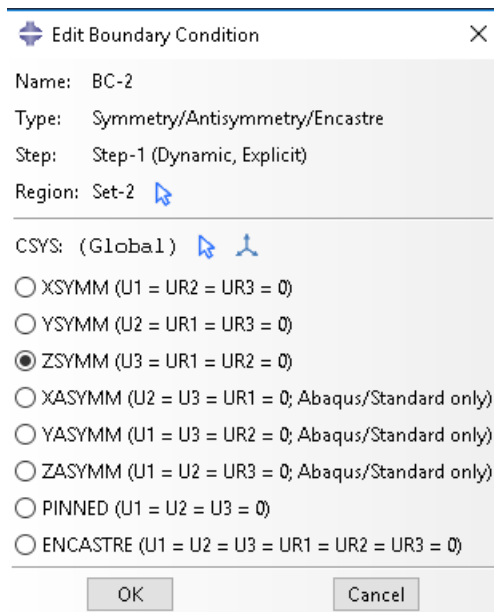


Encastre BC on the bottom surface.

- Symmetry in the Z-direction for the remaining surfaces to prevent deformation of the bar in the z-direction



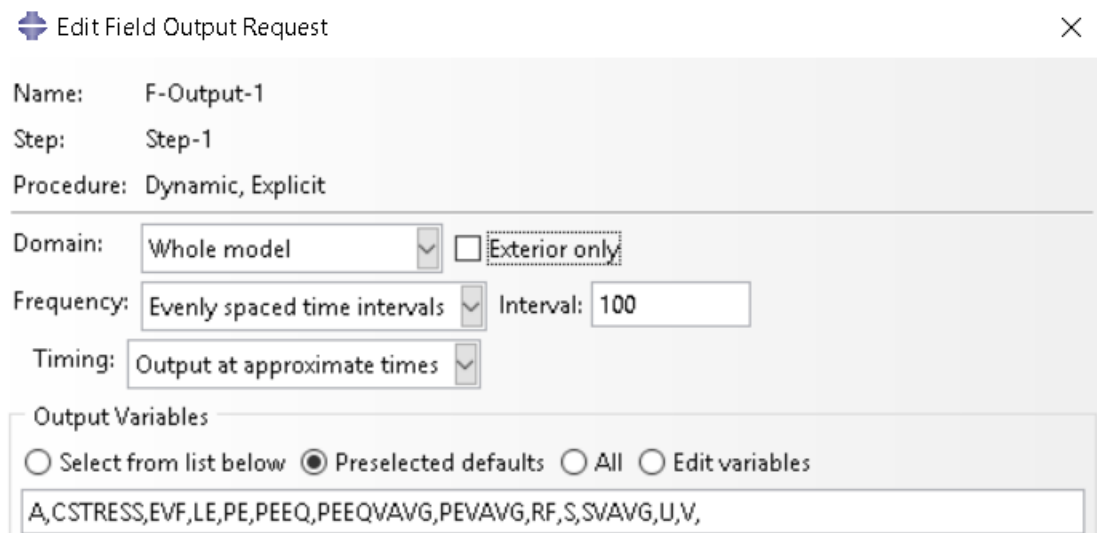
Step 1: Problem setup - Abaqus/Cae



2D displacement settings on remaining surfaces.

1.5 Setting Number of Frames

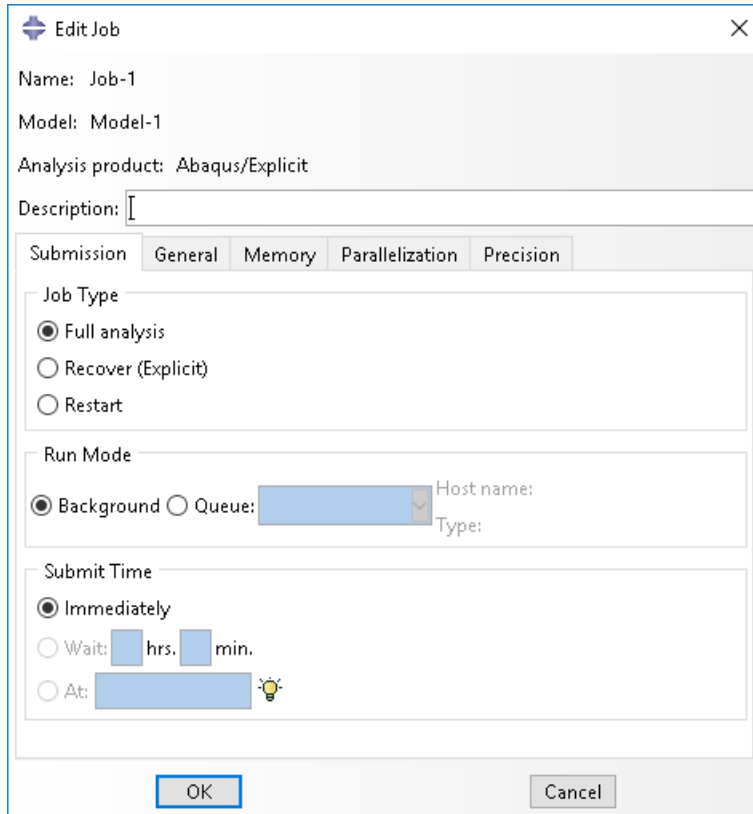
Under the Field Output Requests, open the F-Output-1 output request and modify the Interval number to 100, as per snapshot below.



Field Output Requests

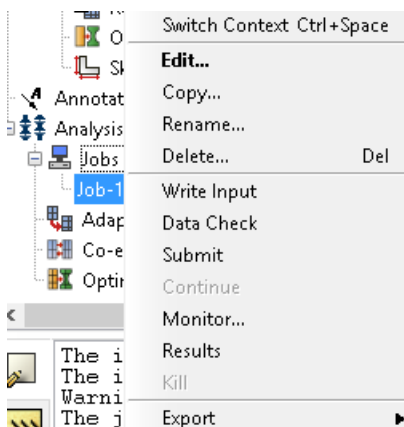
1.6 Creation of Abaqus job

a) Create an Abaqus Job  under **Analysis > Jobs**.



New Abaqus job.

b) Save the Abaqus setup (Save as > Flexible-bar_Dam-break.cae) and write the Abaqus input file (right mouse click on the Job name):



c) This last step will write an Abaqus input file Job-1.inp in the Abaqus\CAE selected working directory (File >

Step 1: Problem setup - Abaqus/Cae

Set working directory) which will be used to launch the co-simulation process.

d) Open the Job-1.inp file just created with a text editor. Complement the lines:

```
** Interaction: Int-1
*Co-simulation, name=Int-1, program=MULTIPHYSICS

with

** 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
```

 **Please note:** If the surface name is different from Surf-1 in your .cae file, you should edit the above text accordingly. The same applies with the interaction name Int-1.

e) Export the structural mesh created as a .stl file to be later imported in XFlow. Navigate to **Parts > Part-1 > Mesh** select **Plug-ins > Tools > STL Export** from the main menu bar and export the file as Bar.stl.

Step 2: Problem setup - XFlow

The second step of the tutorial is the setup of the simulation in XFlow. A 3D simulation will be setup, using water as fluid.

1.1 Environment

a) Open a new XFlow window and create a new project named Flexible-bar_Dam-break.xfp.

b) In **Project Tree > Environment > Engine** set the following parameters:

(a) **Kernel:** 3d

(b) **Flow model:** Free surface

(c) **Analysis type:** Internal

(d) **Advanced Options > Co-simulation:** Abaqus/Simpack

Host name: localhost

Port: 1024

Co-simulation force scheme: Automatic

Limiting pressure on co-sim surfaces: Off

Lead co-simulation: Off

(e) Initial liquid function: $(y < 0.15)(x < -0.01)$

(f) External Acceleration laws Y: -9.81 ms^{-2}

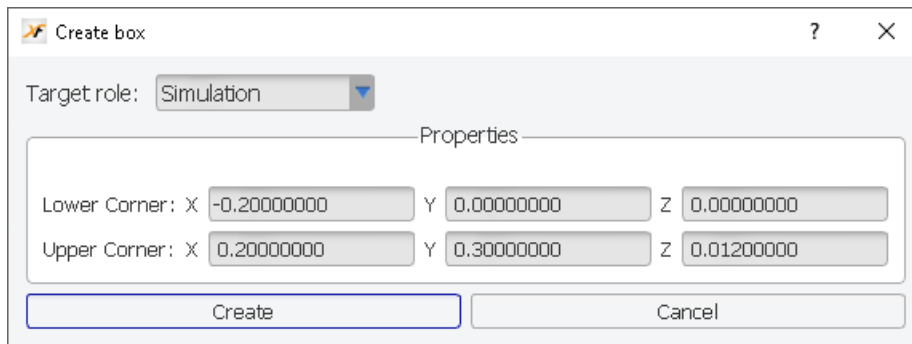
Please note: The Co-simulation option is only available in Labs mode.

1.2 Materials

c) In **Project Tree > Materials** leave the default water properties

1.3 Geometry

d) In **Project Tree > Geometry** create a box which will represent the exterior domain of the simulation using the command **Geometry > Create object > Box** and specifying the following dimensions:

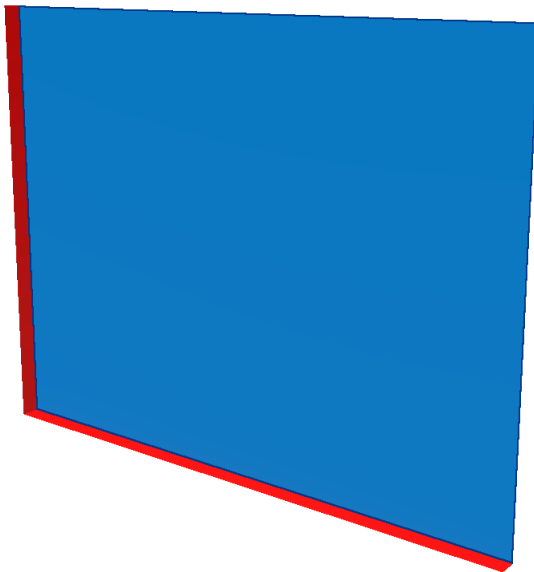


Box geometry dimensions.

e) Set the side surfaces as Free-Slip, selecting the relevant surfaces in the graphic view using the Face filter

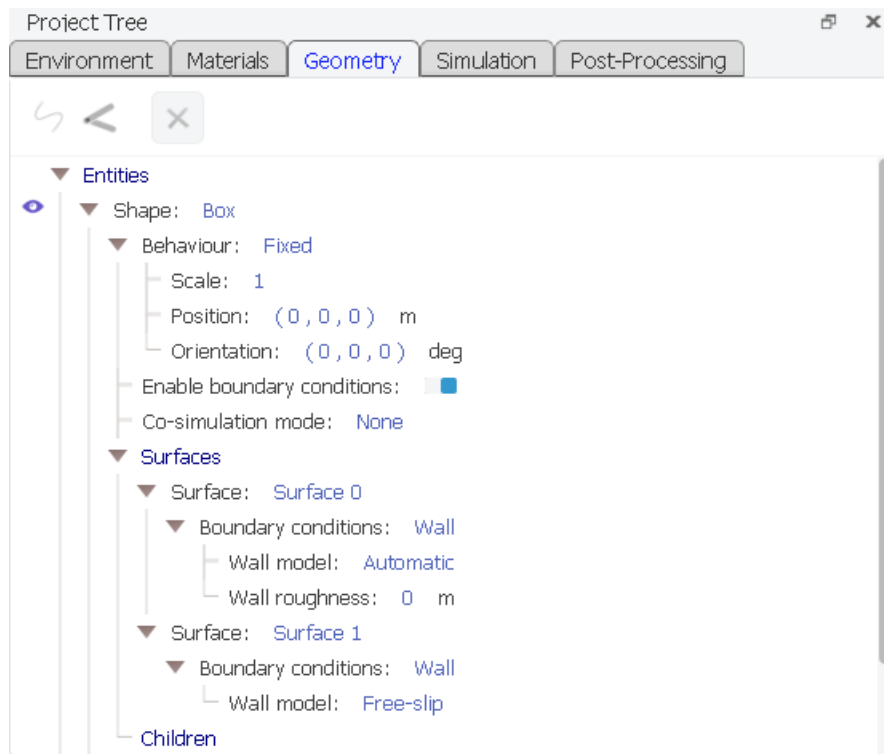


, right mouse click and **Apply boundary conditions to faces.**



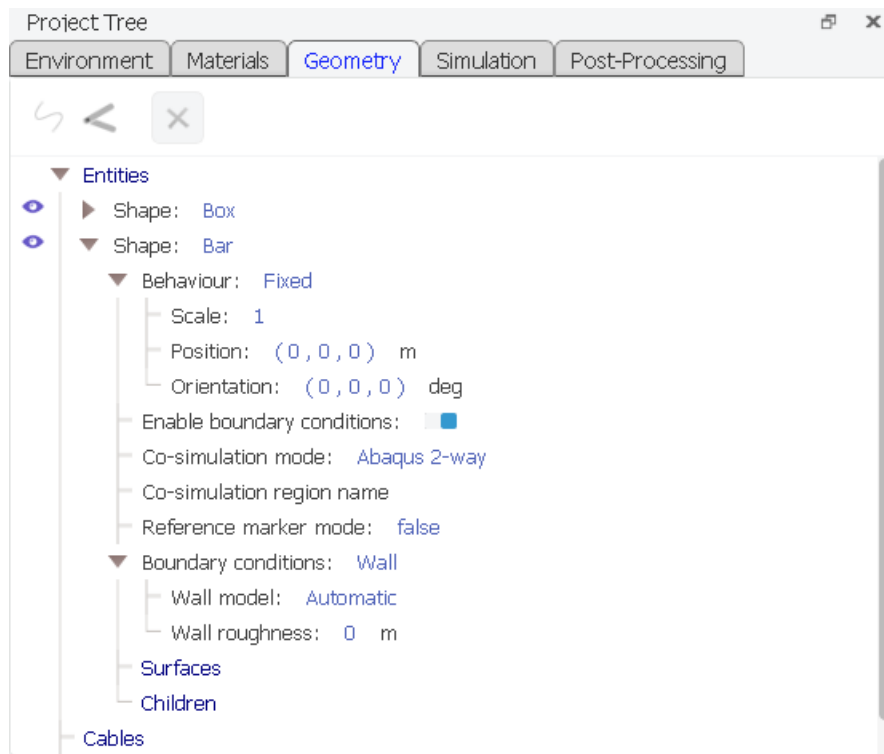
Side surfaces to set as Free-slip walls.

Step 2: Problem setup - XFlow



Side surfaces set as Free-slip walls.

- f) Import the `Bar.stl` geometry create in [Step 1](#) of this tutorial using the command **Geometry > Import a new geometry**. Use the default import settings.
- g) Set the following parameters:
- (a) **Co-simulation mode: Abaqus 2-way**



Geometry Settings in XFlow

1.4 Simulation

h) In **Project Tree > Simulation** set the following parameters

(a) **Simulation time: 0.5 s**

⚠ Please note: The **Simulation time** must always match the step time in Abaqus in order to synchronize the co-simulation.

(b) **Time step:**

Fixed custom: 2e-5 s

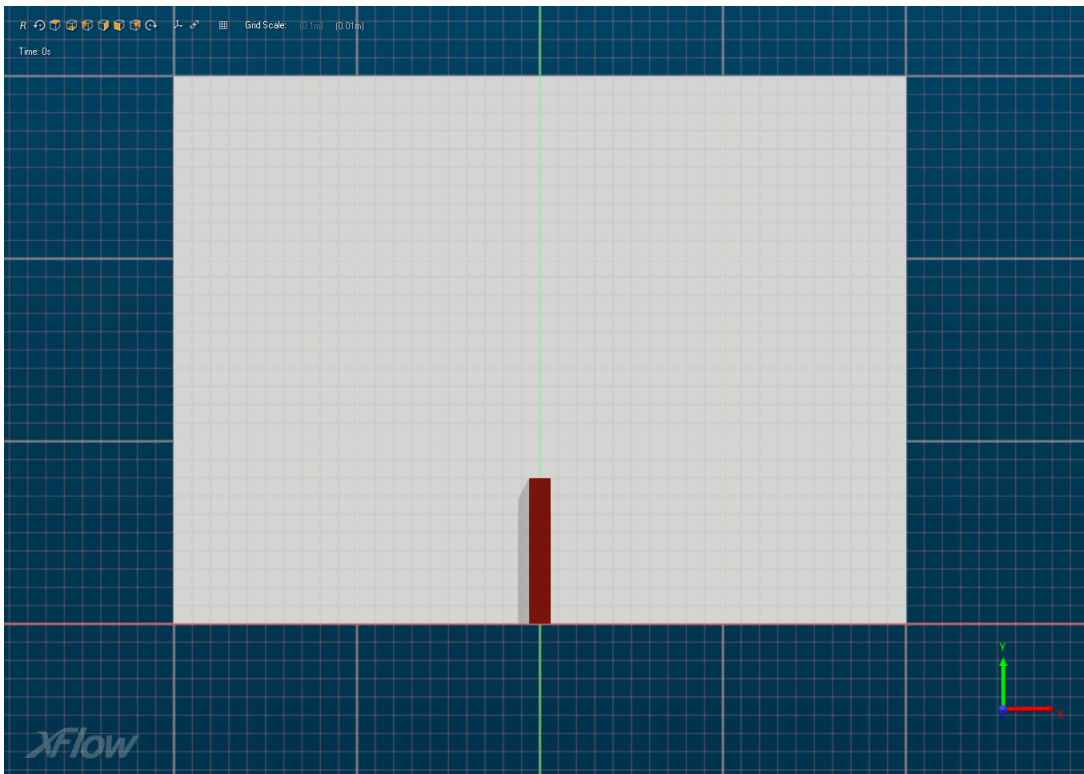
(c) **Resolved scale: 0.0012 m**

(d) **Frames frequency: 200 Hz**

(e) **Compute markers: On**

The resulting setup should look like this:

Step 2: Problem setup - XFlow



XFlow setup for the flexible bar Dam-break case.

- i) Save the project **File > Save project as** Flexible-bar_Dam-break.xfp

Step 3: Execution of Co-simulation

Once the setup of the simulation in Abaqus and XFlow are completed it is possible to run the coupled simulation.

- a) Copy the file `FSI_II_std_css.xml` from the `cse` folder under XFlow's installation folder to the working directory where the `.xfp` and `.inp` files are located
- b) Modify the `FSI_II_std_css.xml` file to reflect the actual simulation time by changing the line:
`<duration>0.5</duration>`
- c) Modify the `FSI_II_std_css.xml` file to reflect the name of the interaction geometry by changing the lines:

```
<connector name="Connector-CFD">
  <componentInstance>Fluid</componentInstance>
  <variables>
    <input>
      <variable>coordinates.Bar</variable>
      <variable>displacement.Bar</variable>
```

```

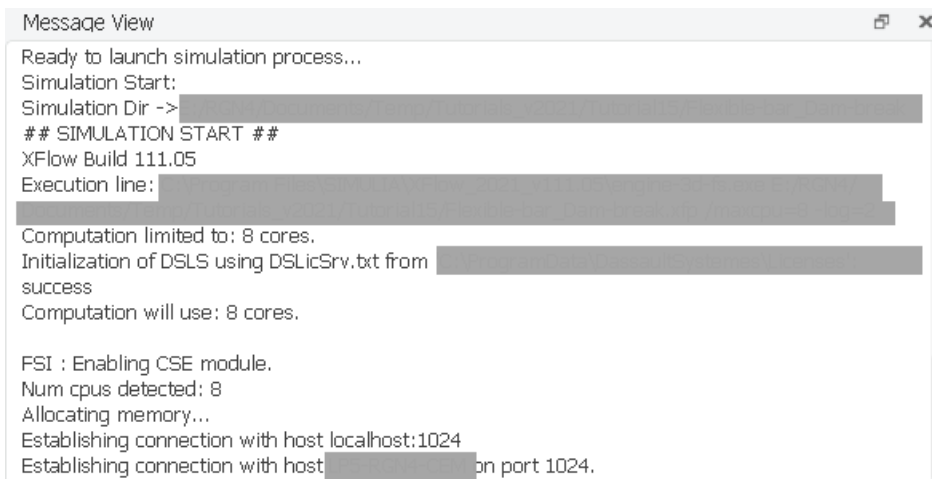
                <variable>velocity.Bar</variable>
            </input>
            <output>
                <variable>force.Bar</variable>
            </output>
        </variables>
    </connector>

```

⚠ 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.

d) Start XFlow simulation by selecting Run > Start Simulation

A message will be printed out in the Message viewer: "Establishing connection with host \$LOCALHOST on port 1024".



e) Start the Co-Simulation engine by executing in a shell the command: `abaqus cse -configure FSI_II_std_css -listenerport 1024`

f) Start Abaqus simulation by executing in a shell the command: `abaqus -job FSI_II_std -input Job-1.inp -double -csdirector localhost:1024 -int`

The co-simulation will start and the two software will synchronize at specified point in time to exchange loads and deformations, as indicated in XFlow's logs.

💡 Tip: If XFlow is run by command lines instead of using the User Interface, the above step c) is then modified as follow.

Start XFlow simulation by executing the following commands in a shell:

- `$XFlow-Dir/generateDomain3d Flexible-bar_Dam-break.xfp`
- `$XFlow-Dir/engine-3d-fs Flexible-bar_Dam-break.xfp`

⚠ Please note: If the XFlow process is launched via command line, the following path has to be exported

Step 3: Execution of Co-simulation

prior to the execution of the co-simulation to correctly locate the CSE libraries:

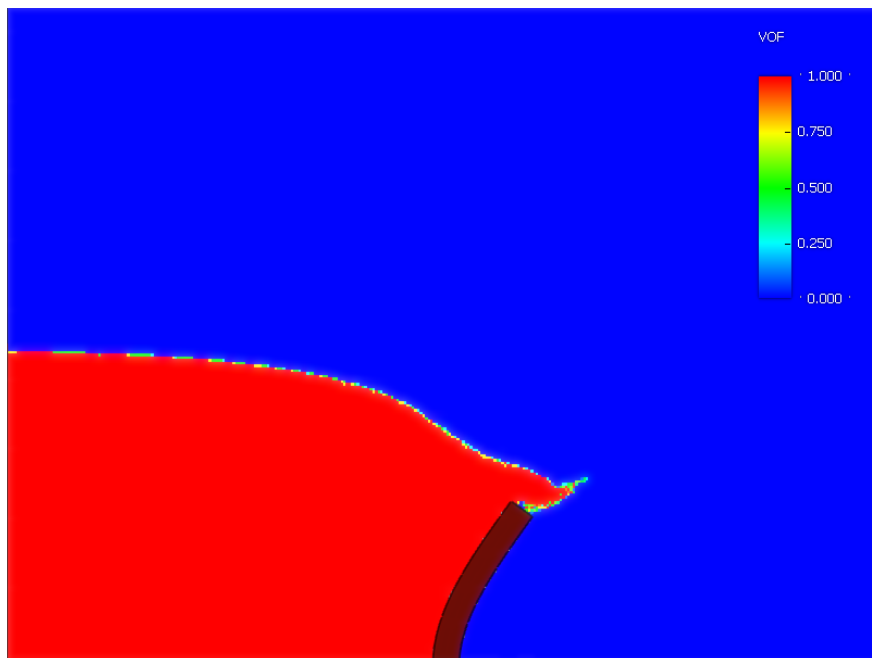
- Linux OS: add the following to the library path the co-simulation libraries from the Abaqus installation:
 - `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:[SIMULATION_SERVICE_DIR]/abaqus[VERSION]/linux_a64/code/bin/`
- Windows OS: run `engine-3d-fs.bat` which will automatically load the route to the co-simulation libraries.

Step 4: Post processing

As with any XFlow simulation it is possible to perform the post-processing on the fly (while the co-simulation is running on the background). The same is true for the Abaqus structural analysis, loading the .odb file into the Abaqus\CAE interface.

1. Post-process the co-simulation in XFlow

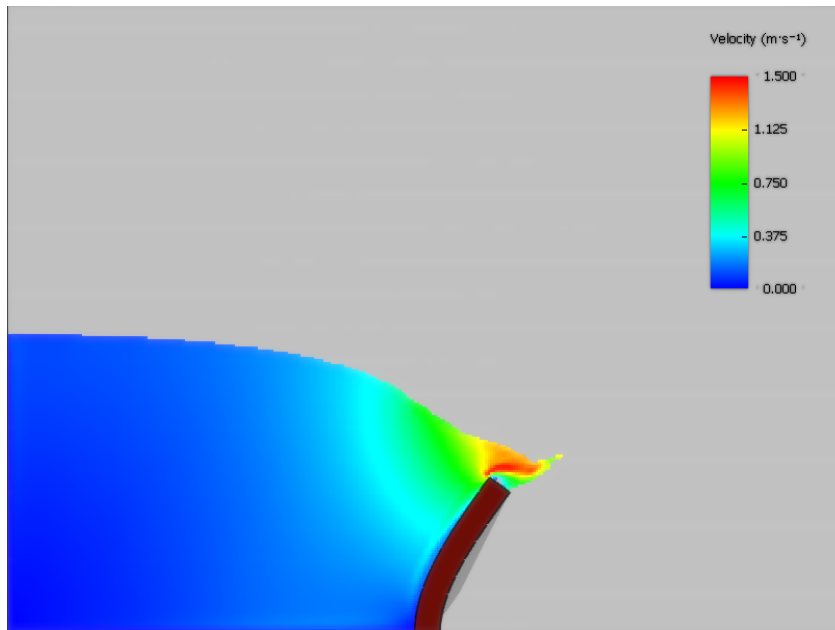
- a) In **Project Tree > Post Processing > Cutting Planes** generate a cutting plane in the Z axis, change the Position value to 0.006 (mid of the domain) and use VOF as field to visualize.
- b) Select frame 20 on the Time bar and visualize the Volume Of Fluid distribution



Simulation results (VOF) at frame 20 ($t=0.1$ [s]).

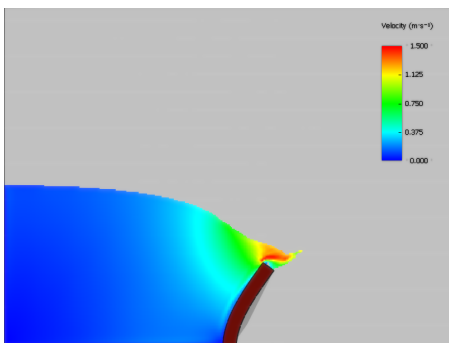
- c) Turn off the cutting plane and turn on the Markers visualization and observe the elastic deformation of the

bar under the effect of the water pressure.

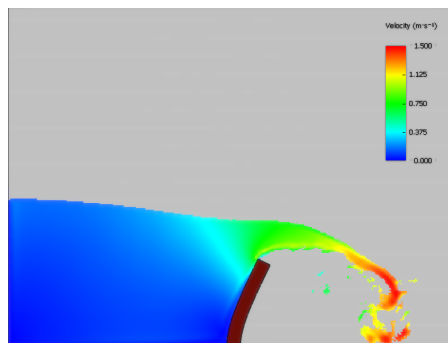


Simulation results at frame 20 ($t=0.1$ [s]).

d) The sequence of snapshots below illustrates the deformation of the bar under the pressure loads of the dam-break and its feedback effect on the evolution of the fluid flow.

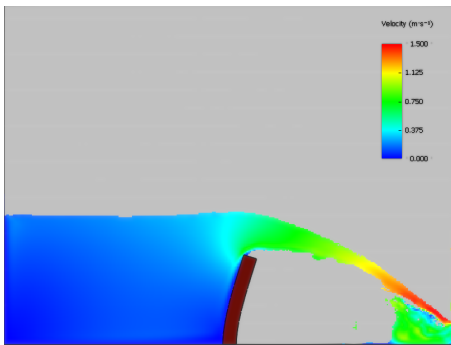


Simulation results at frame 20 ($t=0.1$ [s]).

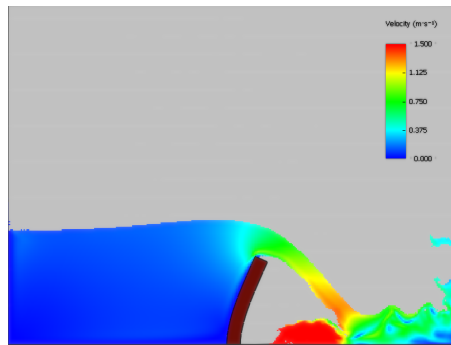


Simulation results at frame 40 ($t=0.2$ [s]).

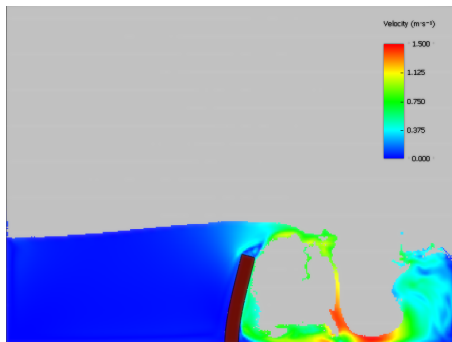
Step 4: Post processing



Simulation results at frame 60 ($t=0.3$ [s]).



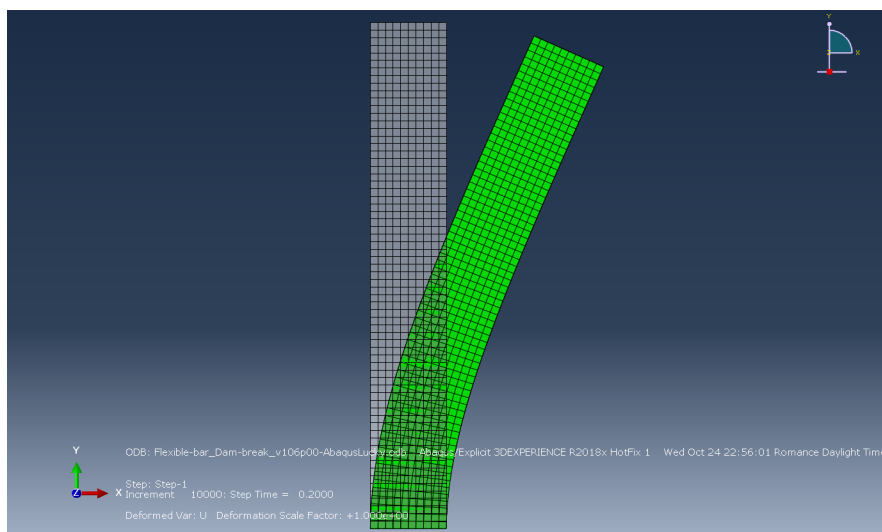
Simulation results at frame 80 ($t=0.4$ [s]).



Simulation results at frame 100 ($t=0.5$ [s]).

2. Post-process the co-simulation in Abaqus\CAE

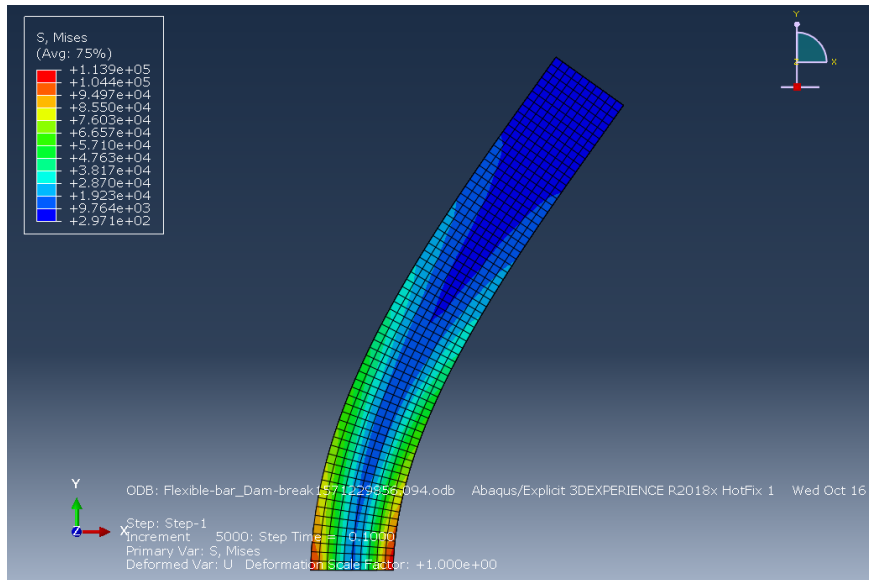
- Load the data into Abaqus\CAE by selecting **File > Open** and select the `FSI_II_std.odb` file.
- Plot the body deformed shape by navigating to the **Results** tab and selecting **Plot deformed shape** .
- Select the fourth frame saved (time step = 0.1) and plot the deformed shape, as per picture below



Bar deformation at time step 0.1 [s].

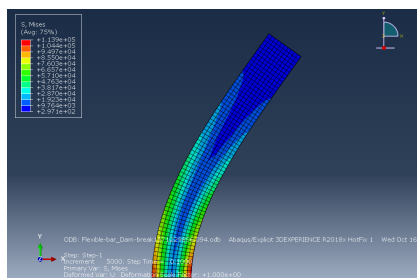
d) It is also possible to visualize the stress generated on the bar element by the fluid forces. Select **Plot**

contours on deformed shape  and plot the **Von Mises** stress value at the same time step. The stress concentration areas are clearly visible in the plot: the attachment area of the beam to the floor, where the bending moment is maximum.

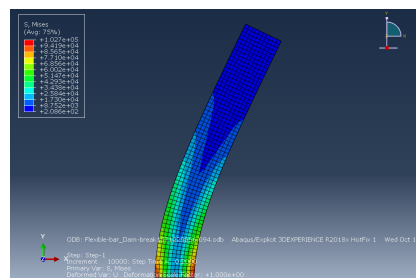


Von Mises stress on the bar at time step 0.1 [s].

e) The sequence of snapshots below illustrates the deformation of the bar under the pressure loads of the dam-break.

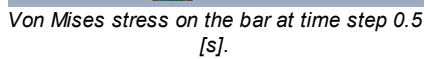
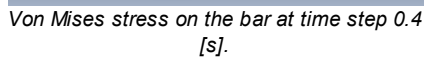
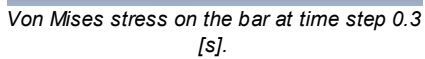


Von Mises stress on the bar at time step 0.1 [s].



Von Mises stress on the bar at time step 0.2 [s].

Step 4: Post processing



Tutorial 16 - Simpack co-simulation: Powertrain Lubrication

This tutorial illustrates how to set up and run an XFlow / Simpack coupled simulation. The specific case here illustrated constitutes of a Powertrain Lubrication case, in which the motion is governed by Simpack.

The tutorial shows how to:

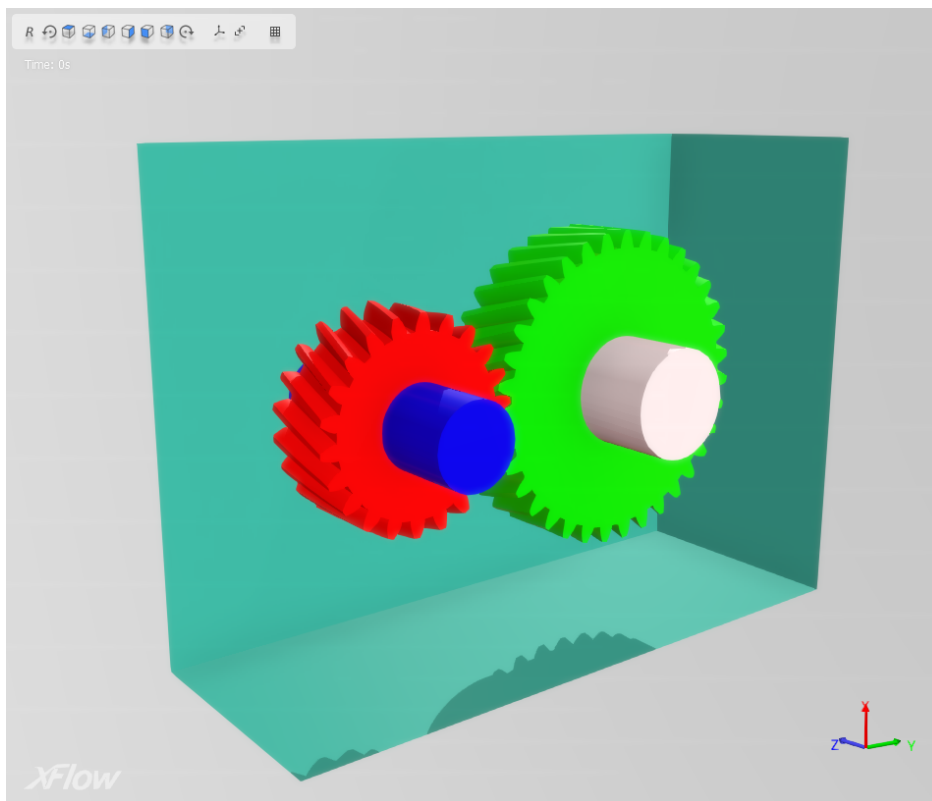
- Setup a Simpack simulation in Simpack - Pre which will be used in a FSI simulation with XFlow
- Setup the correspondent XFlow simulation
- Run the co-simulation and analyse the results

It is assumed that the reader has completed Tutorial 01, 02 and 03. Some steps in the setup will not be described in detail.

 **Please note:** This tutorial requires having activated the Labs mode. The Labs mode can be activated in: **Main menu > Options > Preferences > Application mode**

Contents

- [Step 1: Problem setup - Simpack](#)
- [Step 2: Problem setup - XFlow](#)
- [Step 3: Execution of Co-simulation](#)
- [Step 4: Post-processing](#)

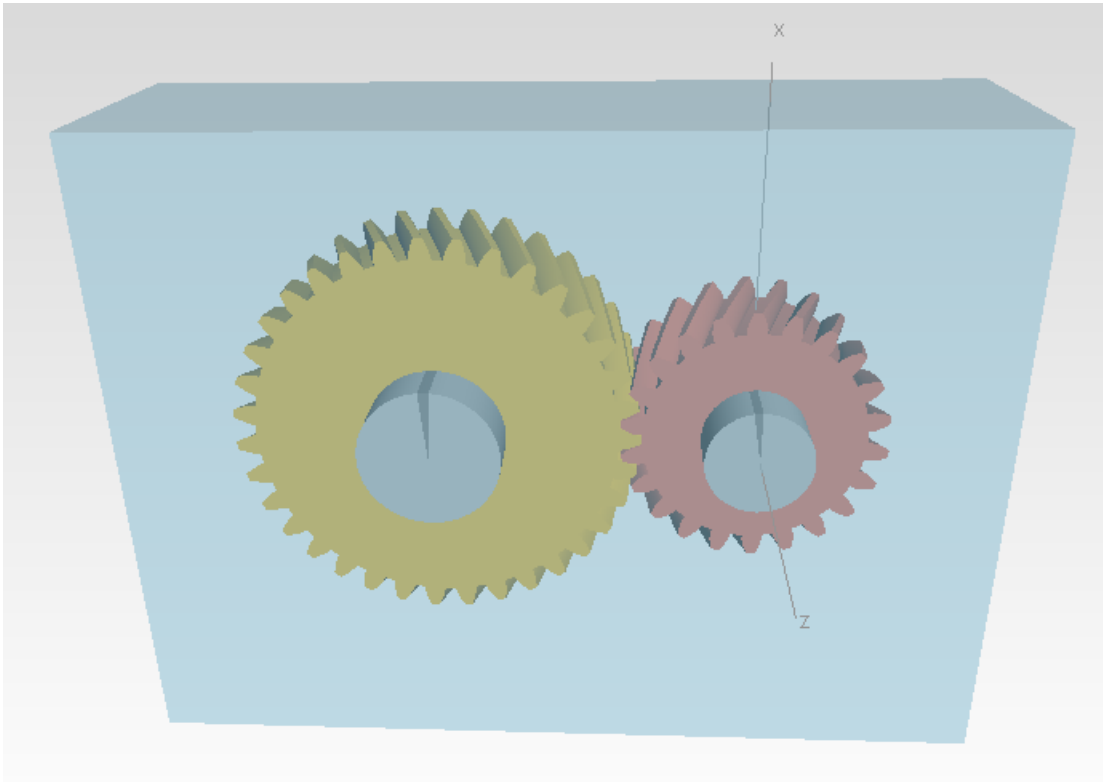


Gear lubrication example

Step 1: Problem setup - Simpack

The first step of the tutorial is the setup of the Simpack multibody analysis. The Simpack-Pre pre-processing software is here used to setup the case and generate the Simpack analysis file (.spck).

To setup the case in Simpack, please follow the guidelines in the documentation of the Simpack Force Element 113



Simpack setup of the gearbox lubrication case.

Step 2: Problem setup - XFlow

The second step of the tutorial is the setup of the simulation in XFlow. A 3D free surface simulation will be setup, using oil as a liquid.

1.1 Environment

- a) Open a new XFlow window and create a new project named SimpleGearBox.xfz.
- b) Import the Simpack project through **Options > Import Simpack model** and navigate to the directory where the xxx.spck model is located (it is also possible to use relative path, e.g. ../xxx.spck)

 **Tip:** Ensure that the Simpack installation folder is correctly set in **Main menu > Options > Preferences > Co-simulation > Simpack installation**

- c) In **Project Tree > Environment > Engine** set the following parameters:

- (a) **Kernel:** 3d
- (b) **Flow model:** Free surface
- (c) **Analysis type:** Internal
- (d) **Thermal model:** Isothermal
- (e) **Advanced Options:** These options will be automatically setup when the Simpack model has been

Step 2: Problem setup - XFlow

imported

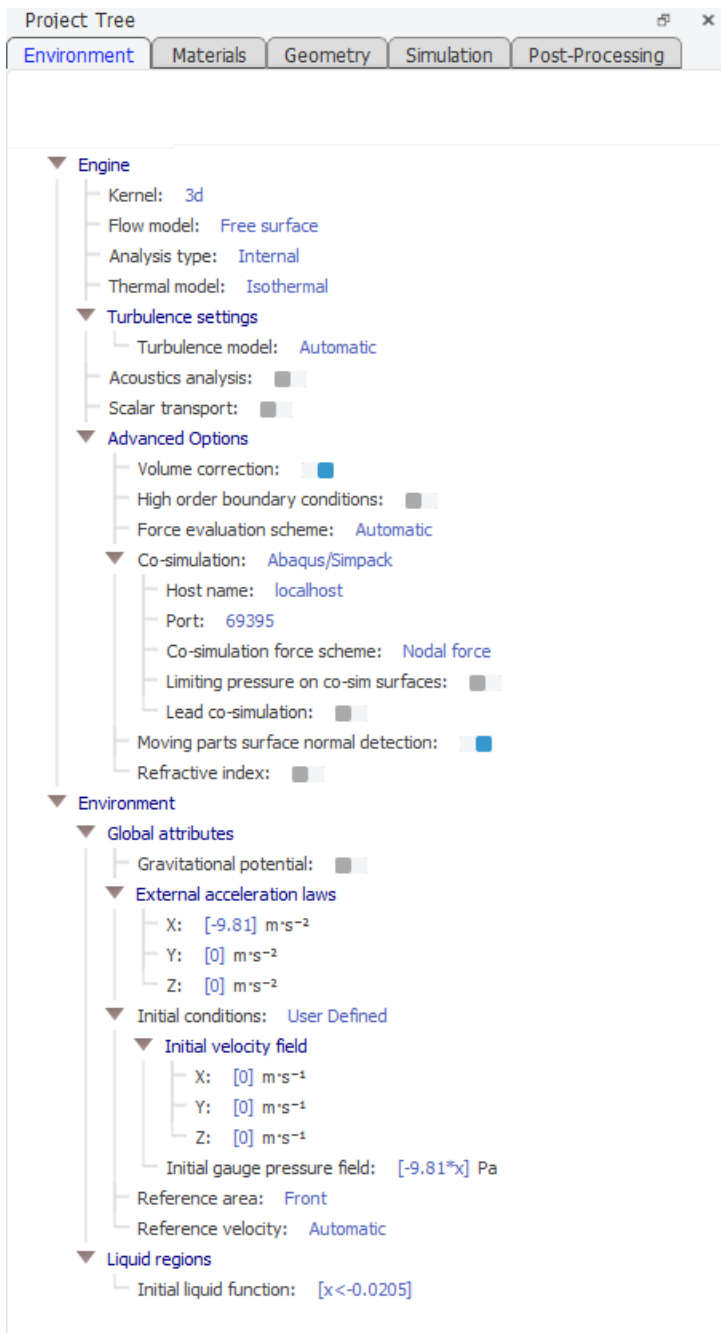
(f) External acceleration laws: X: -9.81 m/s^2

(g) Initial gauge pressure field: $-9.81 \cdot x$

(h) Initial liquid function: $x < -0.0205$

 **Please note:** The **Co-Simulation** option is only available in Labs mode.

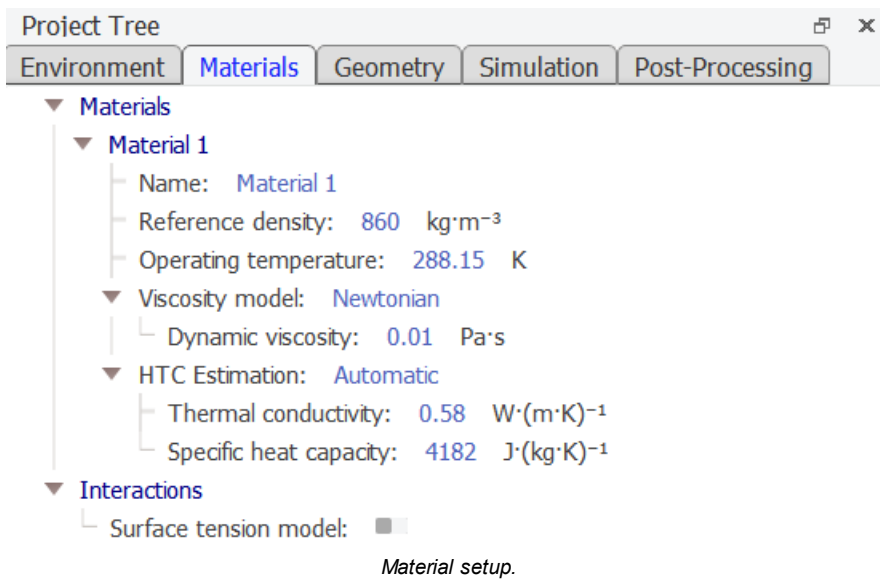
When completed, the Environment tab should look like the caption below:



Environment tab setup

1.2 Materials

a) In **Project Tree > Materials** setup the materials to model Oil in the following way:



1.3 Geometry

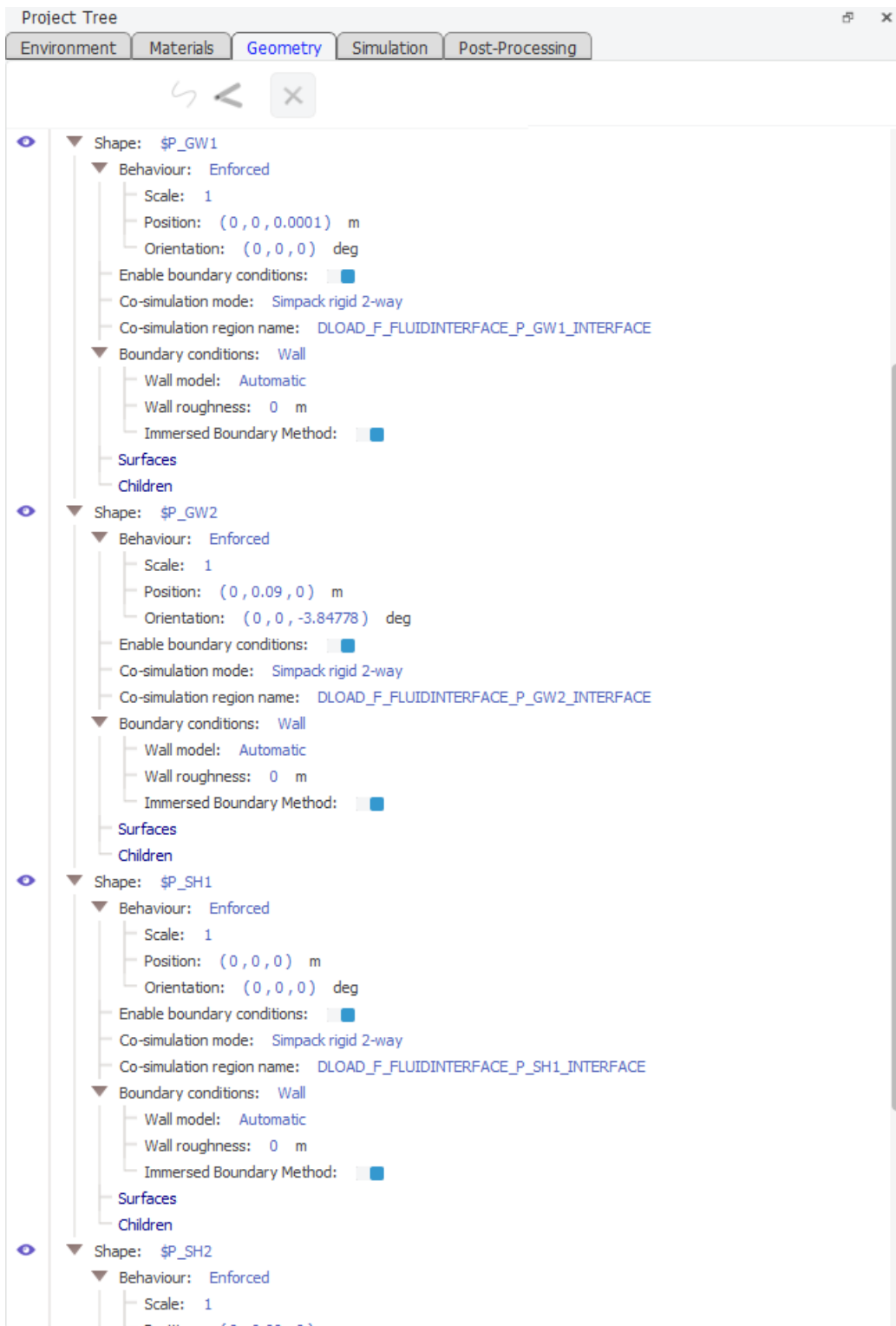
In **Project Tree > Geometry** you will find the gears and shafts imported from the Simpack model, which will be acting as the co-simulation regions on where the loads will be transferred, and are named as follows:

- \$P_GW1
- \$P_GW2
- \$P_SH1
- \$P_SH2

Like with the Advanced Options, all the Co-simulations fields related to the Behaviour of the geometry are automatically selected, based on the inputs received from the Simpack setup.

For these four Co-simulation geometries, select the Immersed Boundary Option to ON.

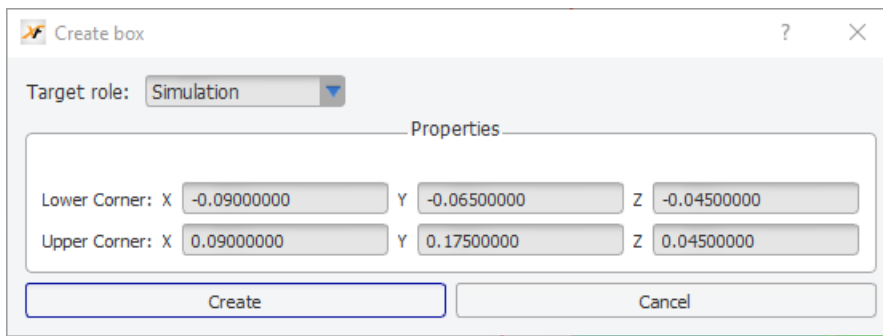
In the screenshot below you can see how the Co-Simulation geometries should be setup:



Geometry setup.

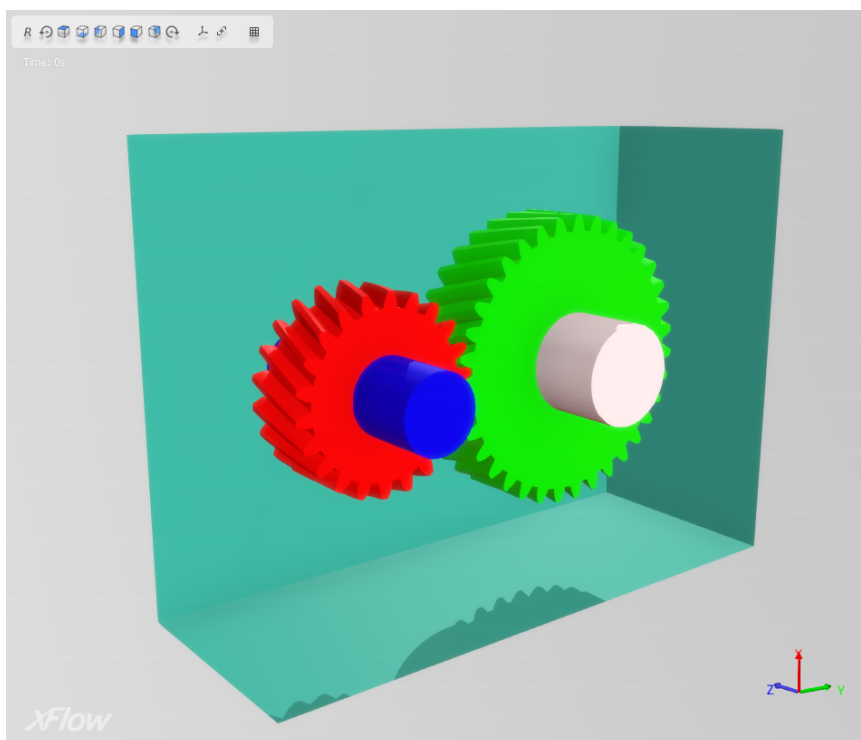
a) Create a Box which will act as the casing of the gearbox with the following properties:

Step 2: Problem setup - XFlow



b) Rename it to "Case" and ensure that the normals are pointing inwards.

The image below illustrates how the geometry should be in the Graphic View:



Geometry setup in XFlow.

1.4 Simulation

In **Project Tree > Simulation** set the following parameters

(a) **Simulation time: 1 s**

⚠ Please note: The **Simulation time** must always match the step time in Simpack in order to synchronize the co-simulation.

(b) **Time step:**

Fixed custom: **5e-4 s**

(c) Resolved scale: **0.004 m**

(d) Frames frequency: **100 Hz**

(e) Numerical data frequency: **Solver time step**

 **Tip:** By default, the numerical data is saved at the solver frequency. Lubrication cases with moving parts normally produce large numerical data files due to the low time step required to capture the physics. The frames frequency option produces significantly lighter and manageable data files.

(f) Compute markers: **On**

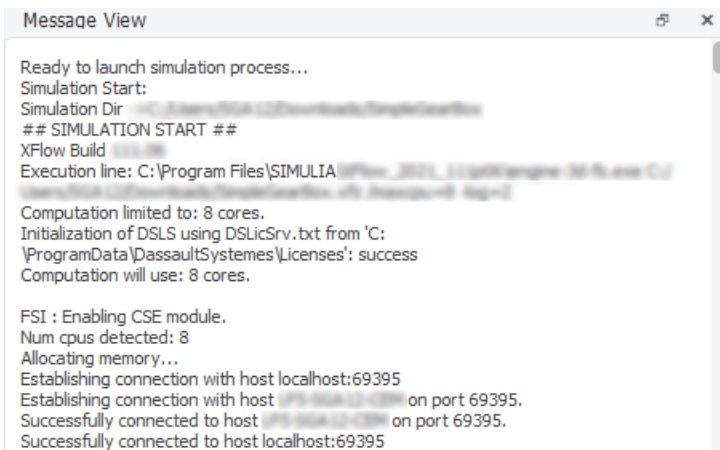
Step 3: Execution of Co-simulation

Once the setup of the simulation in Simpack and XFlow are completed it is possible to run the coupled simulation.

 **Please note:** 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)

Start the XFlow simulation by selecting Run > Start Simulation.

Before the simulation starts the following message should be printed out in the Message Viewer:



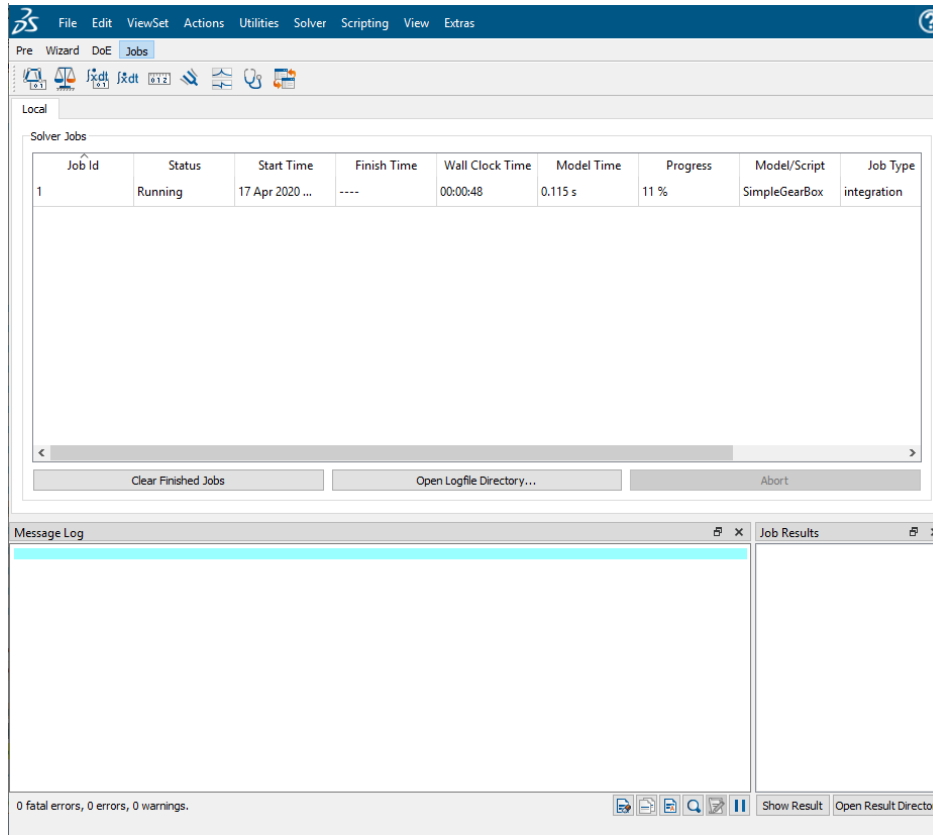
This message indicates that Co-Simulation Engine has started and that XFlow has been able to connect to it. After this step the Simpack solver will start and both simulations will begin, synchronizing at the time step specified in the XFlow simulation.

By default a co-simulation configuration .xml file is generated which will be used by the CSE (SimpleGearBox-cosim.xml) and can be found in the simulation data folder.

Step 3: Execution of Co-simulation

In XFlow, the Message View and completion bar will show you that the simulation is running and its percentage of completion.

For Simpack, you can go to **Jobs** in the Simpack - Pre GUI and check that the **Status** is running



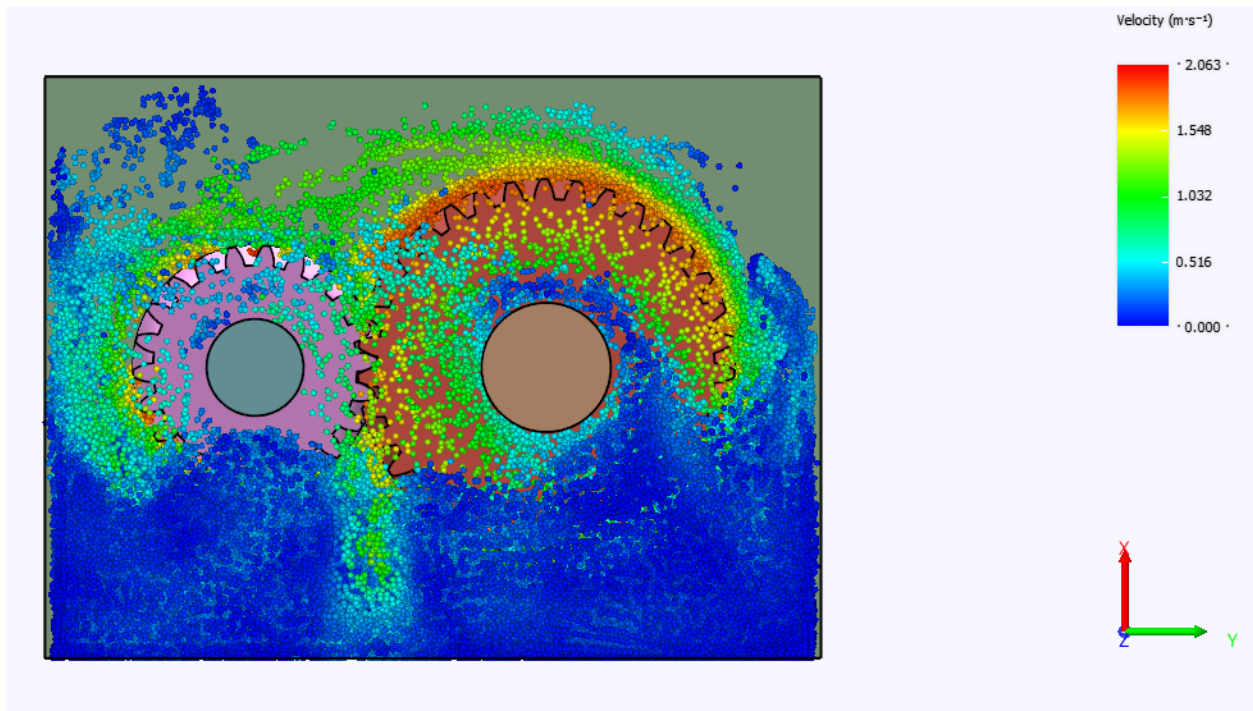
Simpack GUI to launch Simpack solver.

Step 4: Post processing

Post-processing of the XFlow simulation can be done live, whilst the Co-Simulation Engine and Simpack are running in the background.

1. Post-process the co-simulation in XFlow

- In **Project Tree > Post Processing > Markers** choose **Representation: Spheres** to view the oil splashing within the casing and the velocity of the fluid particles on the last computed frame.



Marker visualization at t=1s

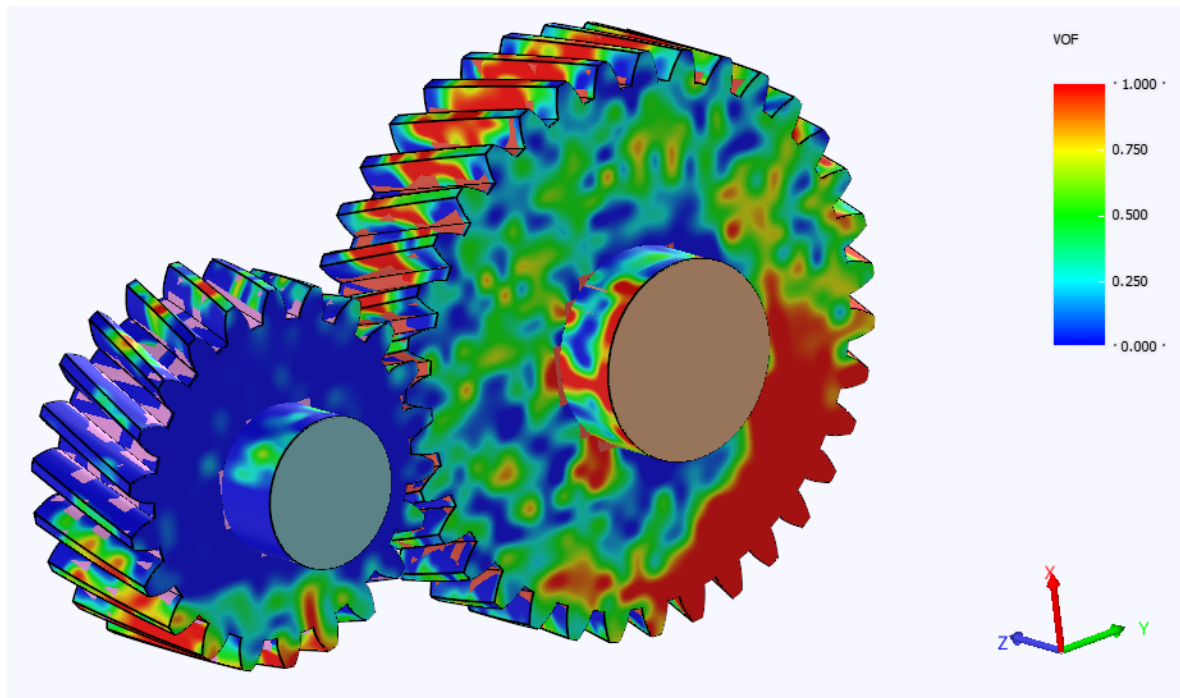
b) In **Project Tree > Geometry** hide the Case geometry.

Go back to **Project Tree > Post Processing > General Data > Interpolation mode:**

Convolution to smooth out the VoF field with the coarse lattice.

Then select **Project Tree > Post Processing > Surface info > Field: VOF**

This shows how much of each of the gears and shaft surface is lubricated with the oil phase on the last computed frame.



Surface VOF visualization at $t=1s$.

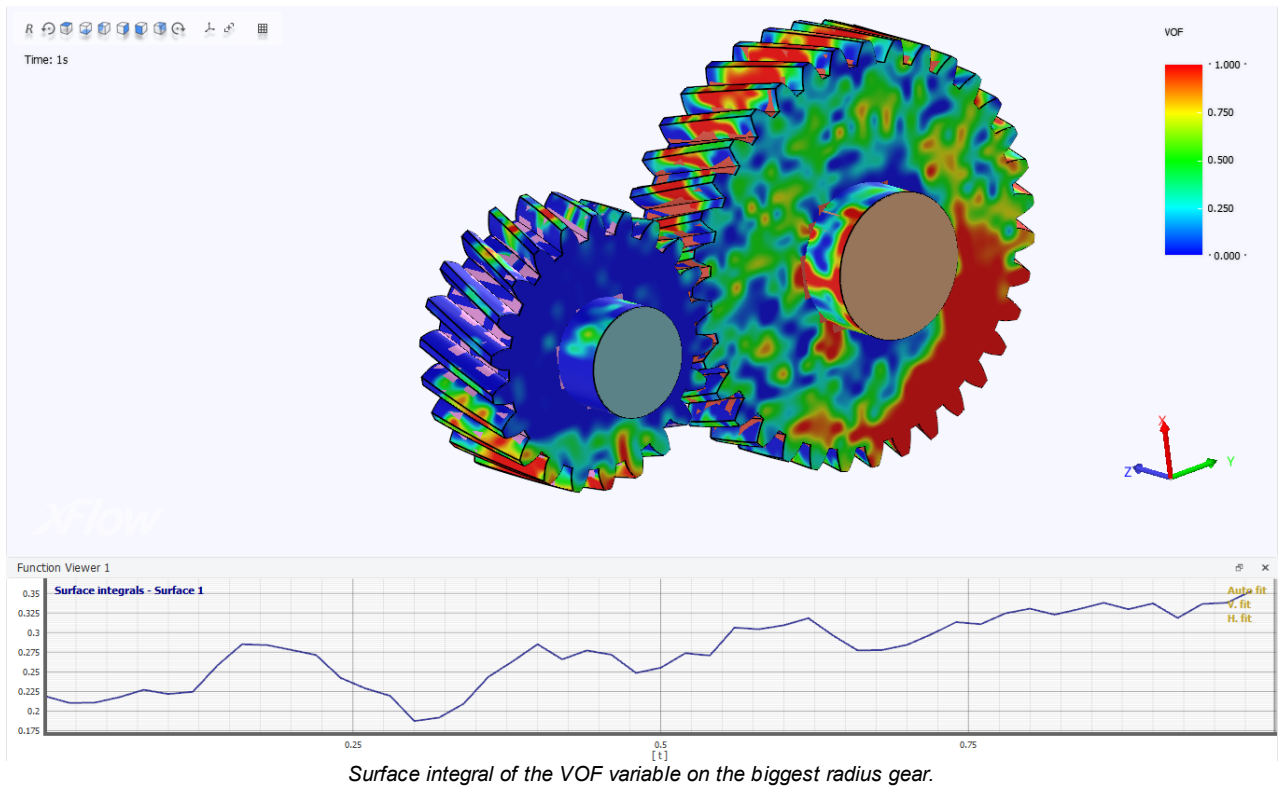
c) This same visualization can be quantified in a numerical way through Surface Integrals, showing the percentage of the gear that is wet. For this right click on **Project Tree > Post Processing > Surface integrals** to add a surface integral.

- (a) **Geometry:** \$P_GW2
- (b) **Integral type:** Average
- (c) **Field:** VoF

The rest of the surface integral fields can be left as default.

Once setup click on **Project Tree > Post Processing > Surface integrals > Surface integral 1** and select refresh surface integral

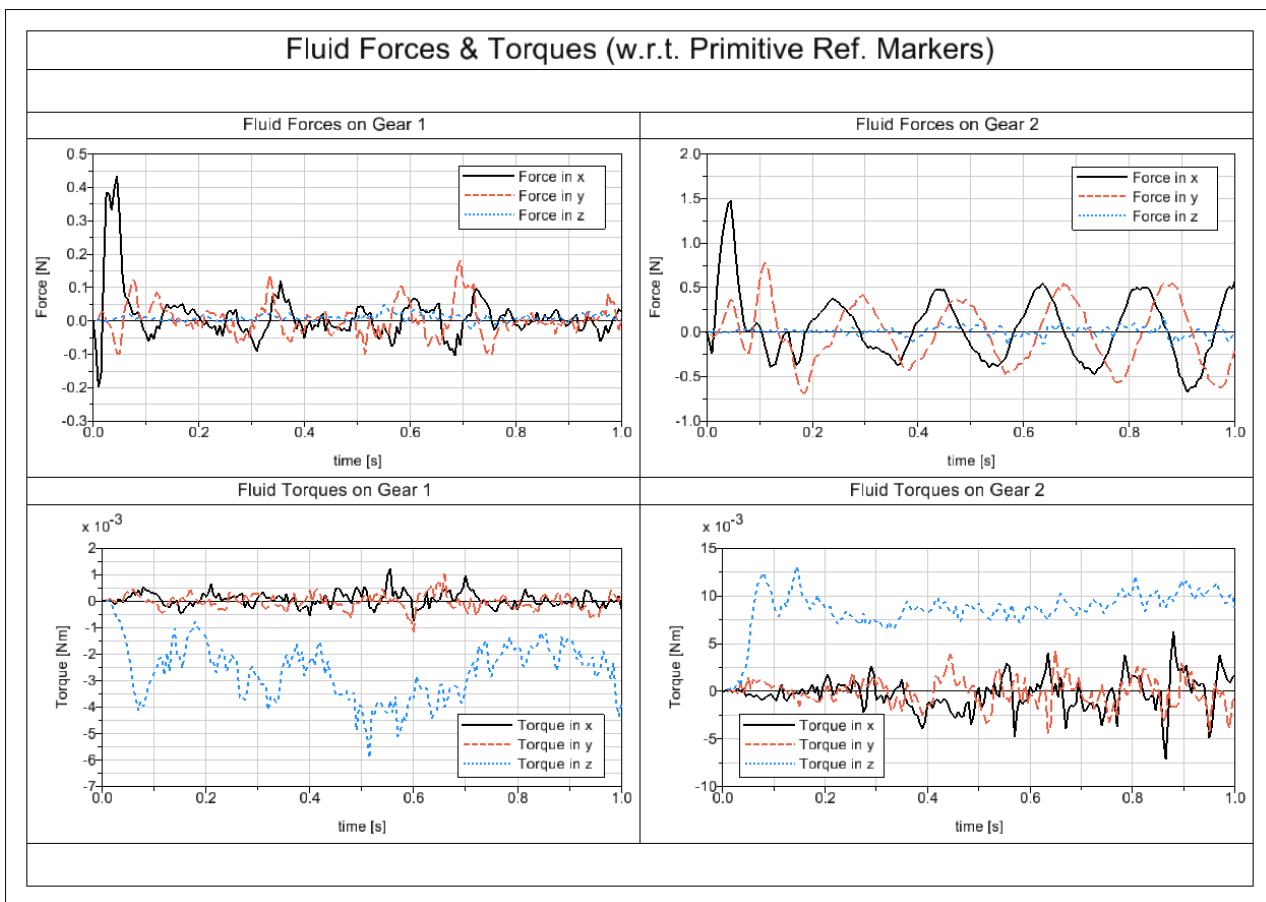
If you right click on the *Function Viewer* and select **Surface integrals > Surface 1** you will see the percentage of oil covering the larger gear's surface.



2. Post-process the co-simulation in Simpack Post

The screenshot below contains the forces and torques evaluated in the post-processor Simpack Post.

Step 4: Post processing



Simpack post-processing.

Index

- C -

Co-sim 263
Coupling 263
CSS 263

- F -

Fluid Structure Interaction 263
FSI 263

- L -

Lubrication 263

- P -

PowerTrain 263

- S -

Simpack 263

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®

DASSAULT SYSTEMES | The **3DEXPERIENCE®** Company

Americas
Dassault Systèmes
175 Wyman Street
Waltham, Massachusetts
02451-1223
USA

Europe/Middle East/Africa
Dassault Systèmes
10, rue Marcel Dassault
CS 40501
78946 Vélizy-Villacoublay Cedex
France

Asia-Pacific
Dassault Systèmes K.K.
ThinkPark Tower
2-1-1 Osaki, Shinagawa-ku,
Tokyo 141-6020
Japan

©2021 Dassault Systèmes. All rights reserved. 3DEXPERIENCE, the Compass icon, the 3DS logo, CATIA, BIOVIA, ENOVIA, EXALEAD, NETVIBES, MEDIDATA, CENTRIC PLM, 3DEXCITE, SIMULIA, DELMIA, and IFWE 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 United States 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.