



*Computational Fluid Dynamics*

## TUTORIAL GUIDE

## XFlow 2014 (Build 94)

© Copyright 2014 Next Limit Dynamics SL

XFlow is a registered trademark of Next Limit Dynamics SL

All other trademarks included in this document belong to their respective owners.

All rights reserved. This document, in whole or in part, may not be copied, reproduced, translated, transferred, transmitted or publicly performed, in any form or by any means - graphic, electronic, machine-readable, or mechanical, including photocopying, recording, or information storage and retrieval systems - without the prior written permission of Next Limit Dynamics SL.

All images in this book have been reproduced with the knowledge and prior consent of the artists concerned and no responsibility is accepted by Next Limit Dynamics SL, producer, publisher, or printer for any infringement of copyright or otherwise, arising from the contents of this publication. Every effort has been made to ensure that credits accurately comply with information supplied.

While every precaution has been taken in the preparation of this document, Next Limit Dynamics SL, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall Next Limit Dynamics SL, the publisher or the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Use of the XFlow software and its documentation has been provided under a software license agreement. Next Limit Dynamics SL assumes no responsibility or liability for any damages or data loss caused by installation or use of the software. Information described in this documentation is furnished for information only, is subject to change without notice, and should not be construed as a commitment by Next Limit Dynamics SL.

The software and its documentation contain valuable trade secrets and proprietary information and are protected by copyright laws. Unauthorized use of the software or its documentation can result in civil damages and criminal prosecution.

# Table of Contents

<b>Using this guide</b>	<b>5</b>
Conventions .....	5
<b>Tutorial 01 - Flow around a cylinder</b>	<b>7</b>
Step 0: Execute XFlow .....	8
Step 1: Create geometry .....	9
Step 2: Problem setup .....	14
Step 3: Run .....	18
Step 4: Post-processing .....	21
Step 5: Refine the resolution .....	29
Step 6: Moving cylinder - enforced motion .....	35
Step 7: Moving cylinder - rigid body dynamics .....	38
<b>Tutorial 02 - Vehicle aerodynamics</b>	<b>41</b>
Step 1: Import geometry .....	42
Step 2: Problem setup .....	45
Step 3: Run .....	49
Step 4: Post-processing .....	51
<b>Tutorial 03 - Advanced post processing</b>	<b>64</b>
Advanced post-processing .....	65
<b>Tutorial 04 - Dam break</b>	<b>75</b>
Step 1: Problem setup .....	76
Step 2: Post-processing .....	78
<b>Tutorial 05 - Breaking waves</b>	<b>82</b>
Step 1: Problem setup .....	83
Step 2: Post-processing .....	87
Step 3: Gravitational potential .....	88
Step 4: Porous volume .....	91
<b>Tutorial 06 - Ball check valve</b>	<b>93</b>
Step 1: No damping case .....	94
Step 2: Under-damping case .....	103
Step 3: Critical-damping case .....	104

Step 4: Over-damping case .....	105
<b>Tutorial 07 - Wind turbine</b>	<b>108</b>
Step 1: Geometry healing .....	109
Step 2: Enforced behaviour - Case setup .....	112
Step 3: Enforced behaviour - Post-processing .....	117
Step 4: Rigid body dynamics behaviour .....	120
<b>Tutorial 08 - Heat transfer</b>	<b>124</b>
Step 1: Problem setup .....	125
Step 2: Post-processing .....	129
<b>Tutorial 09 - Radiation</b>	<b>134</b>
Step 1: Problem setup .....	135
Step 2: Post-processing .....	136
<b>Tutorial 10 - Cyclone flow</b>	<b>140</b>
Step 1: Problem setup .....	141
Step 2: Post-processing .....	145
Step 3: Stream tracers .....	148
<b>Tutorial 11 - FMI standard co-simulation: OpenModelica Pendulum</b>	<b>157</b>
Step 1: Problem setup - XFlow .....	158
Step 2: Set FMI standard in XFlow .....	164
Step 3: Problem setup - OpenModelica .....	167
Step 4: Execution of Co-simulation .....	172
Step 5: Post-processing .....	175
<b>Tutorial 12 - MSC Nastran co-simulation: Turek Hron</b>	<b>177</b>
Step 1: Problem setup - SimXpert .....	177
Step 2: Problem setup - XFlow .....	191
Step 3: Execution of Co-simulation .....	197
Step 4: Post processing .....	200

# Using this guide

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

- [Tutorial 01](#) - Flow around a cylinder
- [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

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 Documentation section of the client area in XFlow website ([http://www.xflowcfd.com/index.php/client\\_area/documentation/view/1](http://www.xflowcfd.com/index.php/client_area/documentation/view/1)).

Please take into account that the tutorials presented in this guide have been set to compute in a short amount of time. To have more accurate results, you will have to lower the resolution.

## 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.

## Conventions



**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).

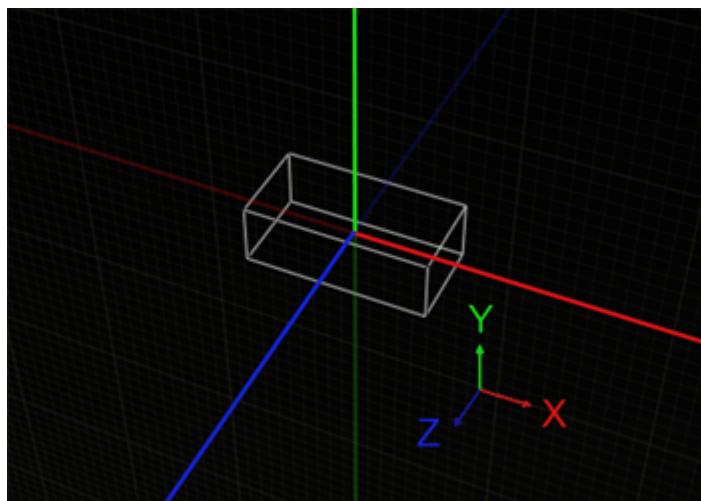
Measure	Symbol	Units
Mass		kg
Length		m
Time	t	s
Velocity	$v_x, v_y, v_z$	m/s
Pressure	p	Pa
Temperature	$\theta$	Kelvin
Acceleration		m/s <sup>2</sup>
Density	$\rho$	Kg/m <sup>3</sup>
Viscosity	$\mu$	Pa-s
Thermal Conductivity	k	W/m-K
Specific Heat	$C_p$	J/kg-K
Angle		degrees
Angular velocity		rad/s



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

## Coordinate system

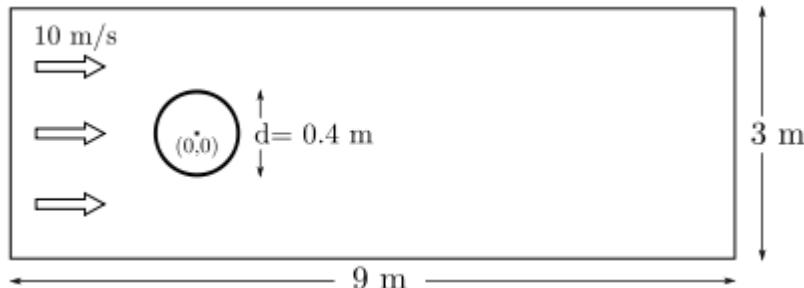
Special attention needs to be paid to the coordinate system. In XFlow the Y-axis is assumed to be vertical instead. The user may have thus to rotate the geometry when importing it from the CAD software. This can be done either in the CAD or in XFlow.



# Tutorial 01 - Flow around a cylinder

This first tutorial illustrates the setup and solution of the two-dimensional flow around a cylinder at Reynolds number  $Re_d = 4000$ .

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



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 part
- Set a simple fluid-structure interaction

## Contents

- [Step 1: Create geometry](#)
- [Step 2: Set up the problem](#)
- [Step 3: Run](#)
- [Step 4: Analyse results](#)
- [Step 5: Refine the resolution](#)
- [Step 6: Moving cylinder - enforced motion](#)

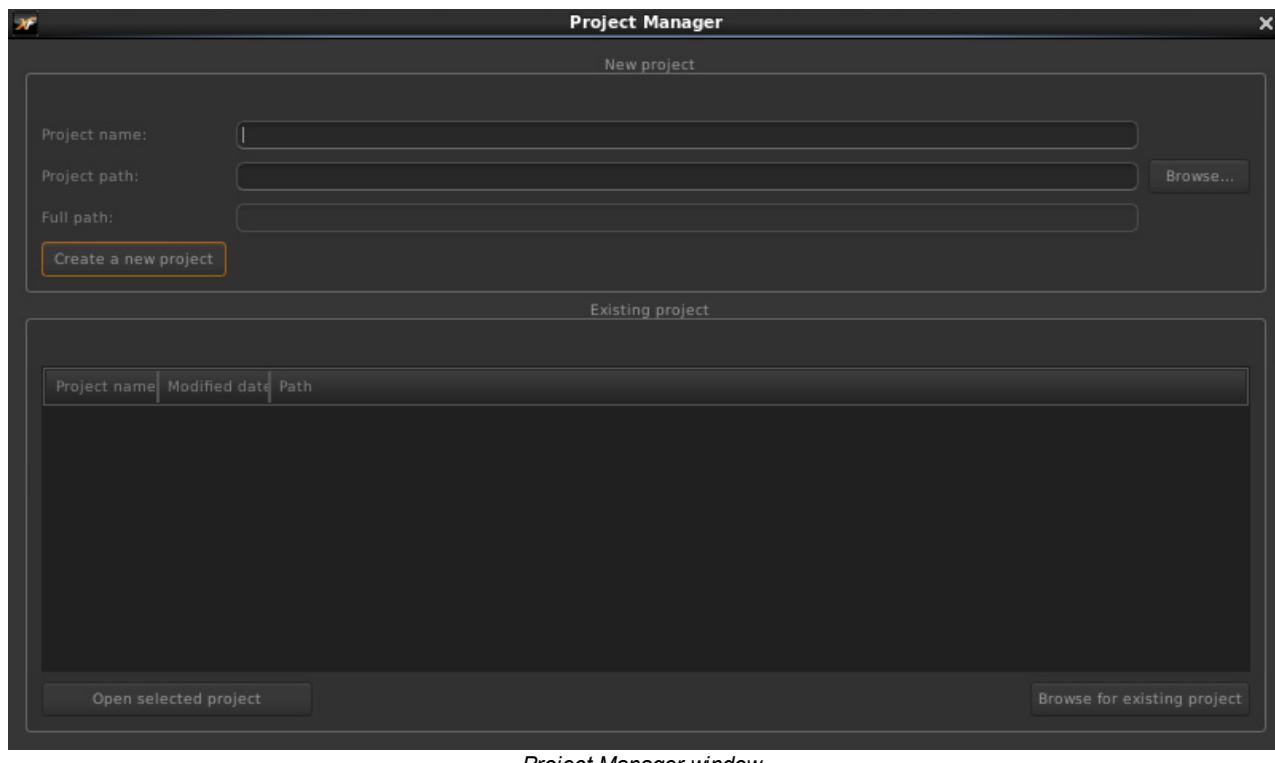
## [Step 7: Moving cylinder - rigid body dynamics](#)

## 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.sh`

In the *Project Manager* enter the project name, the project path (you may need to browse to specify the path or create a new folder) and press the button **Create a new project**.



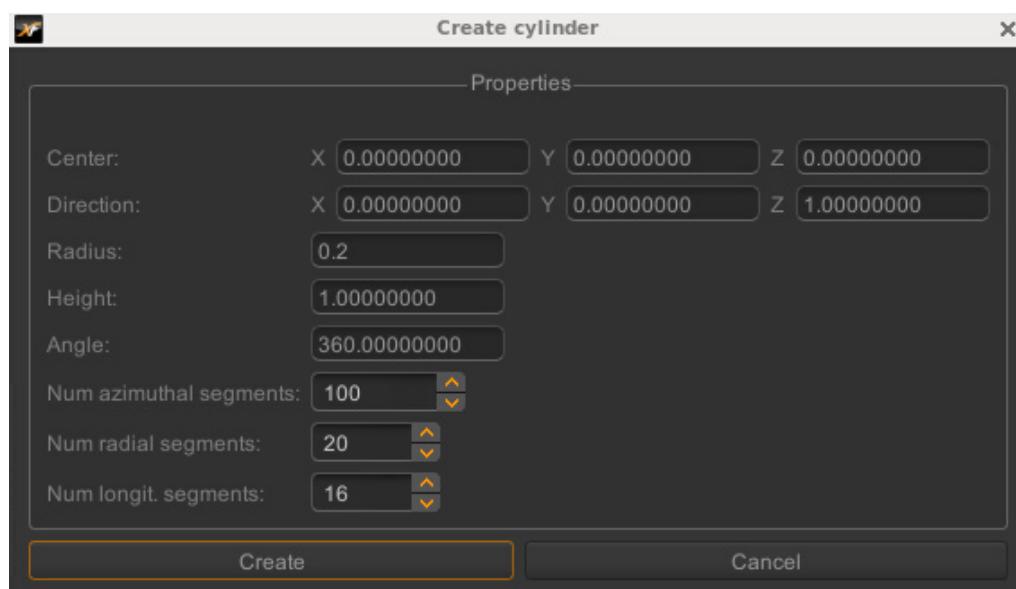
## Step 1: Create geometry

### 1.1 Create a cylinder

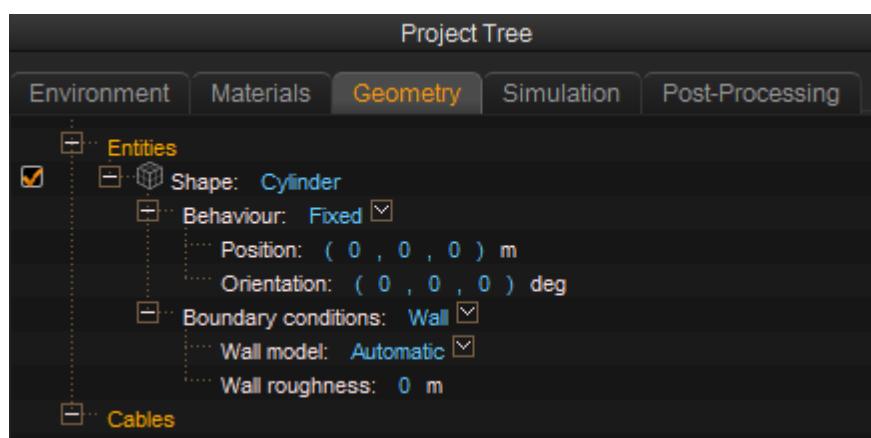
Create a cylinder centered at the origin, in the Z-direction, with radius 0.2 m and height 1.0 m, by means of:

Main menu > Geometry > Create object > Create cylinder, or  in Toolbar Object Creation.

Introduce the geometrical data of the cylinder in the dialogue box ([Units](#) in SI):



The cylinder appears as a **Shape** in Project Tree > Geometry > Entities:

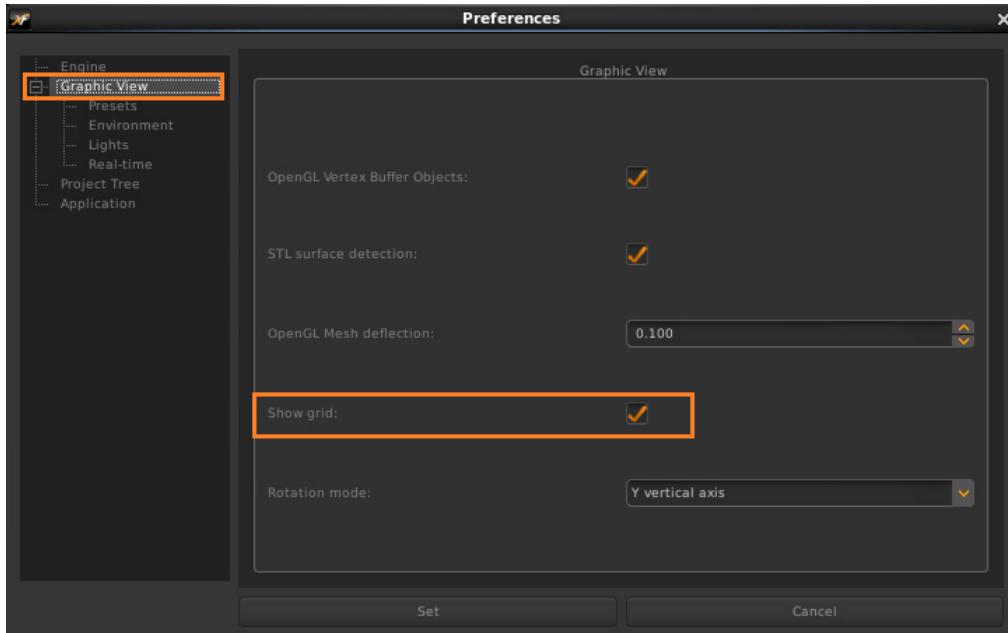


### 1.2 Check the position of the cylinder with the help of the grid

## Step 1: Create geometry

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

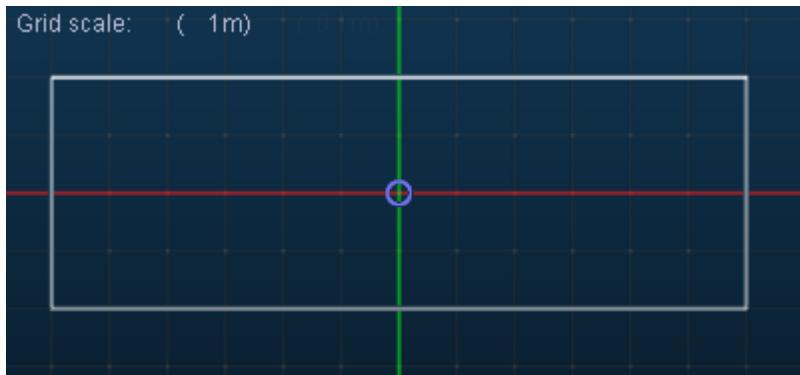
Enable the grid in: **Graphic View > Show grid**



Preferences > Graphic View > Show grid

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:

Top      Bottom      Left      Right      Front      Back      Perspective      Fit all



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.

**⚠ 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 cylinder

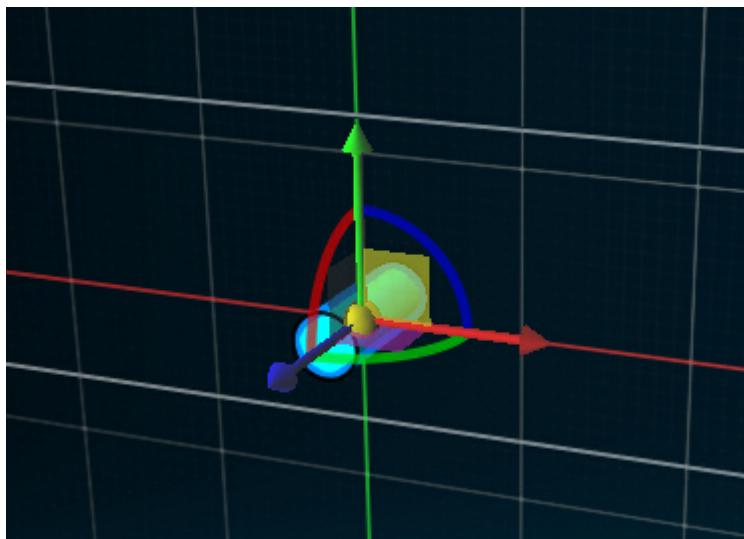
Select the cylinder geometry either by:



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



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



Selected cylinder - Gizmos shown

Gizmos allow the user to translate, rotate and scale the geometry. To familiarize yourself with the Gizmos please 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 cylinder.
- **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 cylinder.
- **click** on any of the Gizmos axes center - the scale dialogue box appears - and enter 2. Observe the new size of the cylinder (twice the original).

Select the cylinder and press **Delete** to delete the modified object and create the cylinder again as indicated in [Section 1.1](#)

## 1.5 Check the geometrical properties of the cylinder

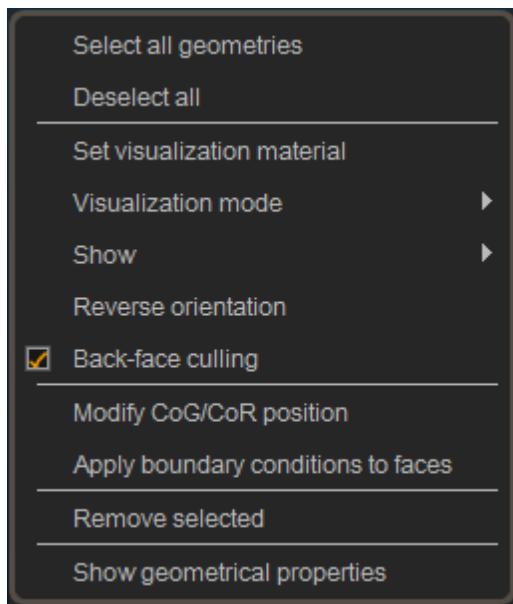
## Step 1: Create geometry

Select the cylinder and measure its dimensions: **Main menu > Geometry > Dimensions**, or  (

*Toolbar geometry*).

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

With the cylinder 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:



In this menu, select **Set visualisation material** to change the colour of the cylinder 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 cylinder as shading, wireframe, bounding box and mesh:

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

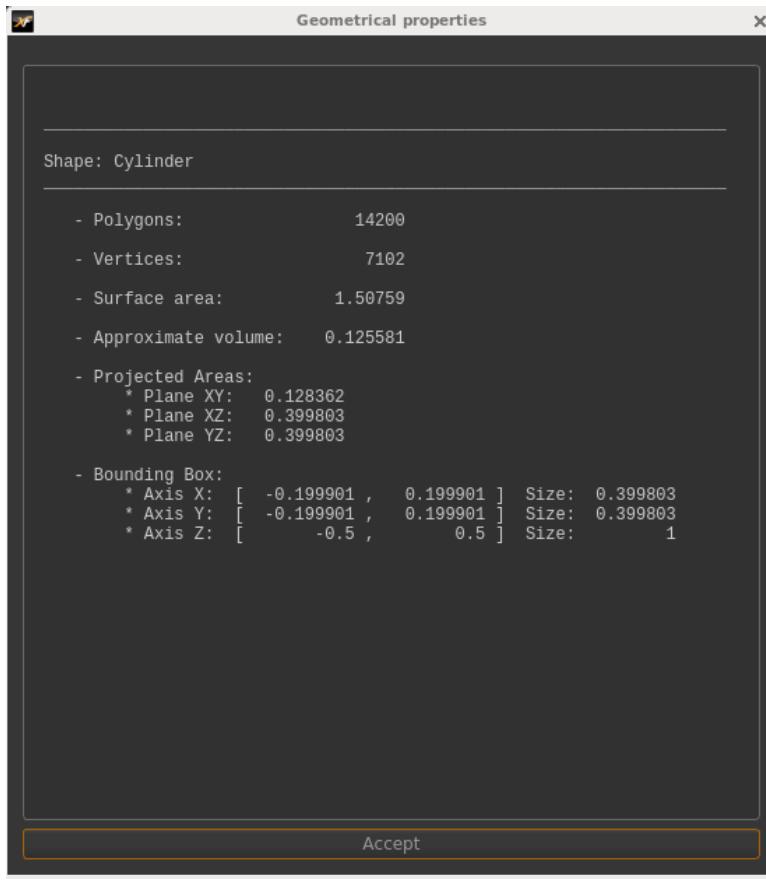
Select geometry > **right mouse button** in the *Graphic View* > **Graphic View menu > Visualisation mode > Mesh**

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 cylinder:

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



*Geometrical properties report*

The surface of the cylinder is automatically tessellated by XFlow when it is created. Therefore, the surface mesh properties (Polygons and Vertices) are also displayed. This tessellation depends on the parameter:

**Main menu > Options > Preferences > Graphic View: OpenGL mesh deflection**

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



**Check:** Make sure that **OpenGL 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 create it again (see [above](#)). The geometry has to be created again in order to take into account the change of the **OpenGL mesh deflection** value.

## Step 1: Create geometry

**⚠ 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.

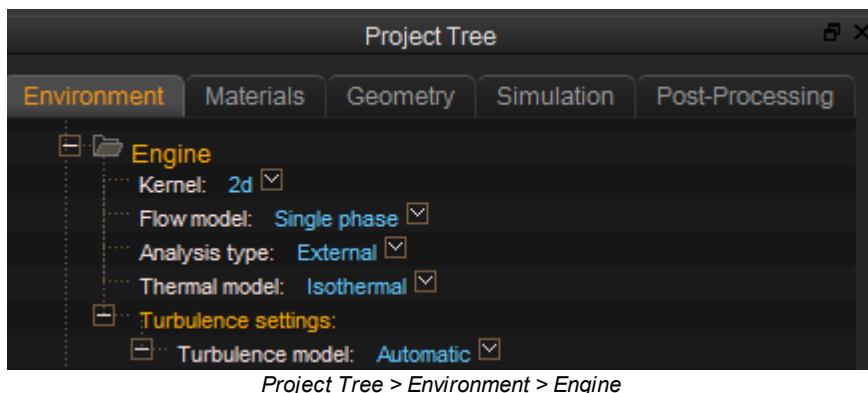
The setup of the problem is done in the following sections of the *Project Tree*:

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

### 2.1 Engine settings

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

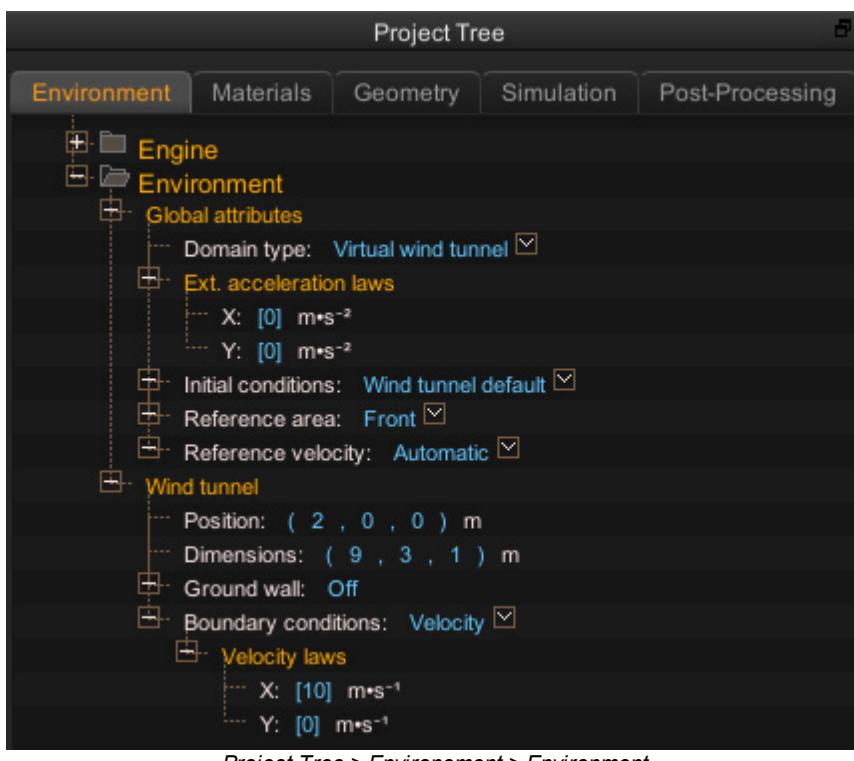
- (a) **Kernel : 2d**
- (b) **Flow model: Single phase**
- (c) **Analysis type: External**
- (d) **Thermal model: Isothermal**
- (e) **Turbulence settings:**
  - **Turbulence model: Automatic**



### 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 metres in the +X direction
- (g) Wind tunnel > Dimensions: 9 metres length (X), 3 metres height (Y) and 1 metre width (Z)
- (h) Wind tunnel > Ground wall: **Off**
- (i) Wind tunnel > Boundary conditions: **Velocity**
- (j) Wind tunnel > Boundary conditions > Velocity law:  **$10 \text{ m}\cdot\text{s}^{-1}$**  in +X direction. This boundary condition would be applied on the -X boundary of the wind tunnel (inlet).



Project Tree > Environment > Environment



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

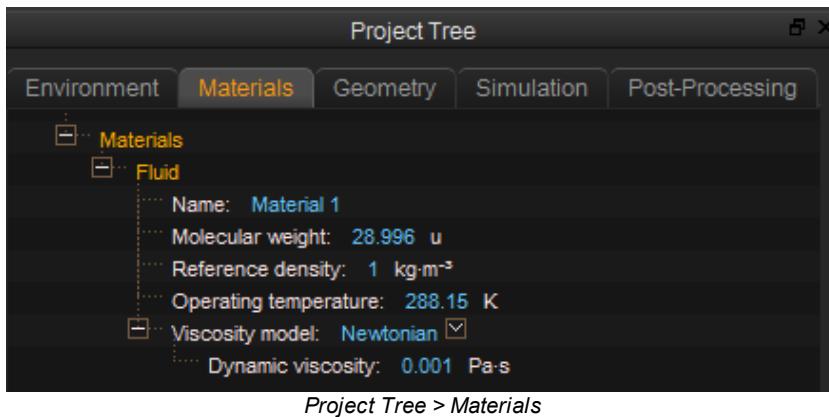


**Please note:** The initial condition **Wind tunnel default** allows to initialise the flow using the same wind tunnel conditions.

## 2.3 Materials settings

By default in single phase the fluid material is called *Material 1*, which is initialised with the air thermophysical properties. To get a  $Re_d = 4000$ , given a fluid velocity of  $10 \text{ m}\cdot\text{s}^{-1}$  and a cylinder diameter of 0.4 m, the *Material 1* properties have to be modified as:

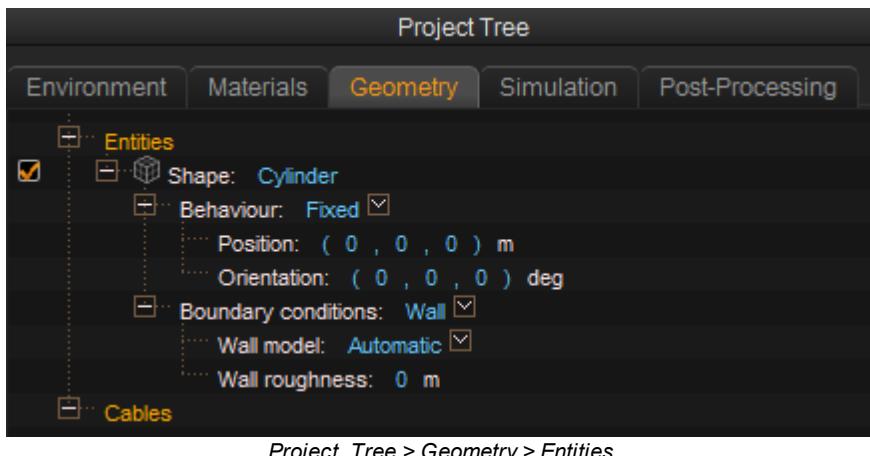
- (a) Project Tree > Materials > Fluid > Density :  **$1 \text{ kg}\cdot\text{m}^{-3}$**
- (b) Project Tree > Materials > Fluid > Viscosity model: **Newtonian**
- (c) Project Tree > Materials > Fluid > Viscosity model > Dynamic viscosity:  **$0.001 \text{ Pa}\cdot\text{s}$**



## 2.4 Geometry settings

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

- (a) Project Tree > Geometry > Entities > Shape: Cylinder > Behaviour: **Fixed**, leave blank position and orientation.
- (b) Project Tree > Geometry > Entities > Shape: Cylinder > Boundary conditions: **Wall**, with Automatic wall model and zero roughness

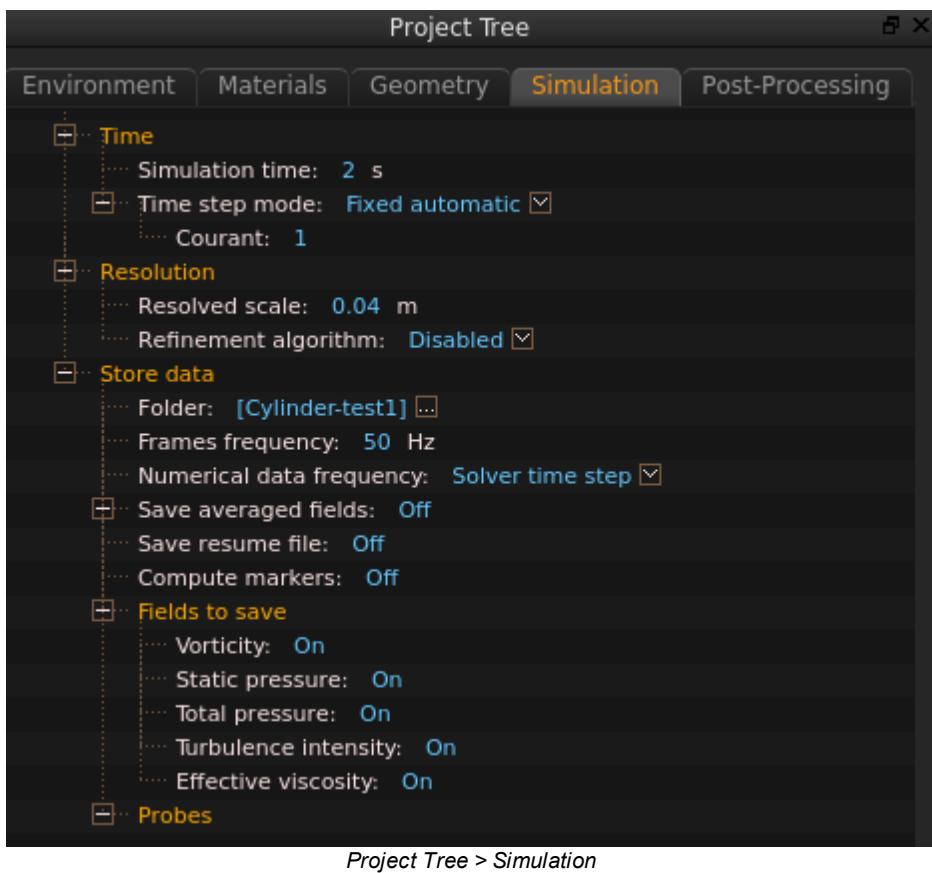


## 2.5 Simulation settings

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

- (a) **Time > Simulation time:** **2 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) **Resolution > Resolved scale:** **0.04 m**
- (e) **Resolution > Refinement algorithm:** **Disabled** (i.e. uniform resolution)
- (f) **Store data > Folder:** Cylinder-test1. This is the name of the folder where the numerical data will be stored.
- (g) **Store data > Frames frequency:** **50 Hz** (i.e. 50 frames per second, which means that a total of 100 frames will be saved for the 2 seconds of simulation time)
- (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**
- (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 > Fields to save:** Leave **On** all the fields, which means that all the flow fields will be saved on the hard disc.

## Step 2: Problem setup



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 Cylinder.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

### 3.1 Save project

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

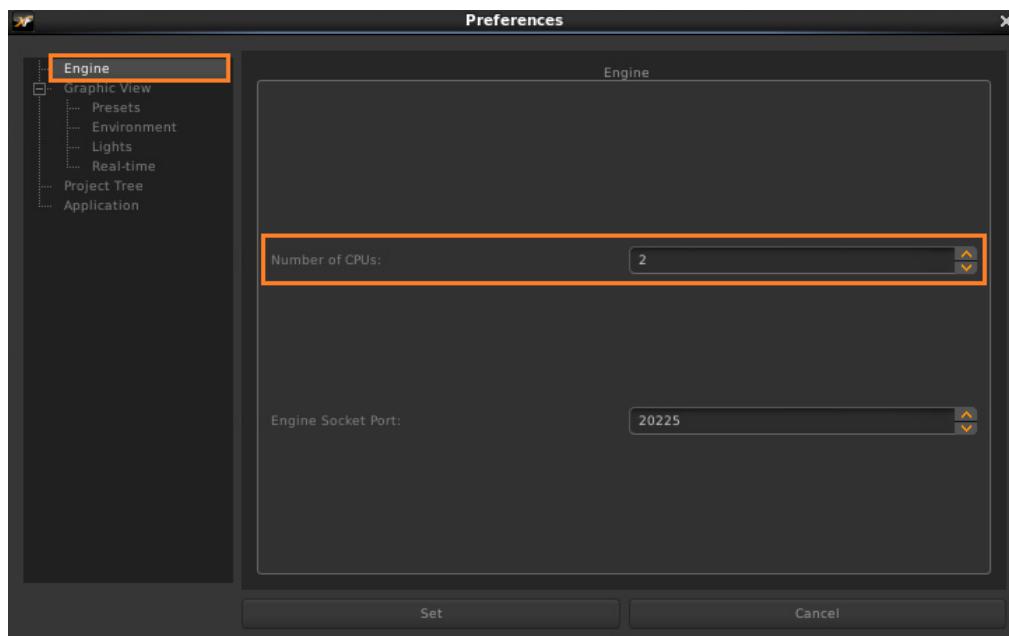
XFlow project files have the extension .xfp.

### 3.2 Set number of CPUs

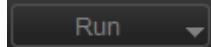
Set the number of CPUs, to be used for the calculation, in the preferences of the engine:

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 `cylinder-test1` in the same location as the project file. The results data and log files are saved in this folder.

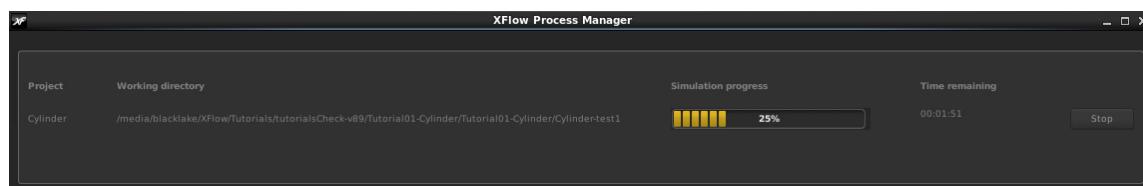
As XFlow is calculating, engine messages are shown in the *Message View* window.

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



### 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.



### Step 3: Run

Note that the *Message View* window shows the message "Connected to Engine Daemon on port 20222".

The port can be changed in

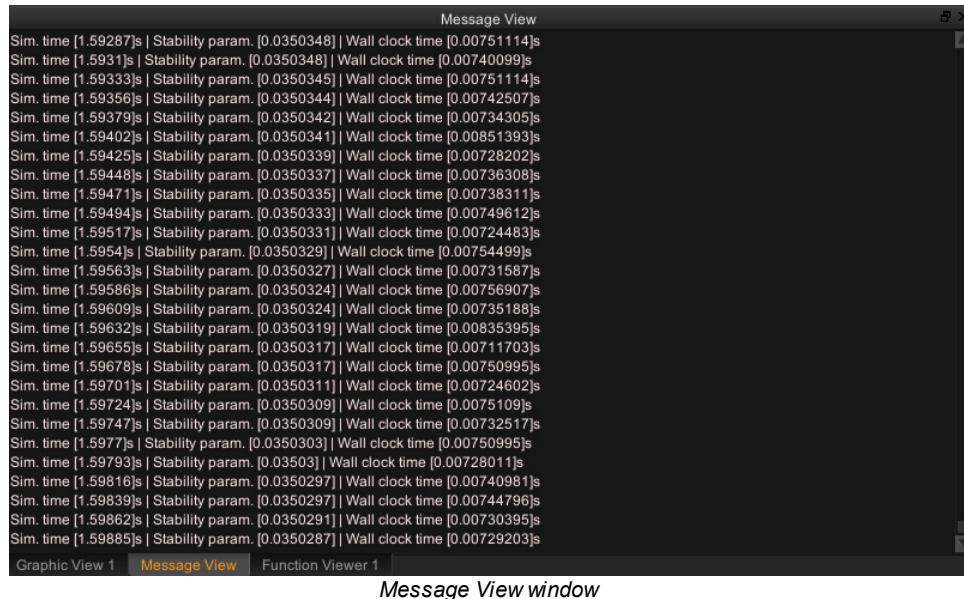
**Main menu > Options > Preferences > Engine: Engine Socket Port**

**⚠ 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 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:



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: Run](#)
- Wall clock time: the total time which has been required to compute the time step

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 1<sup>st</sup> 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

### 3.6 Stability parameter

When your simulation is running, the user has the possibility to monitor the evolution of the **Stability parameter** in 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–Lowy (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.

The stability parameter can be monitored in the *Function Viewer* window. To this end, do **right click** on the *Function Viewer* window and select Stability parameter.



*Stability parameter evolution, monitored in the Function Viewer*

## 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.

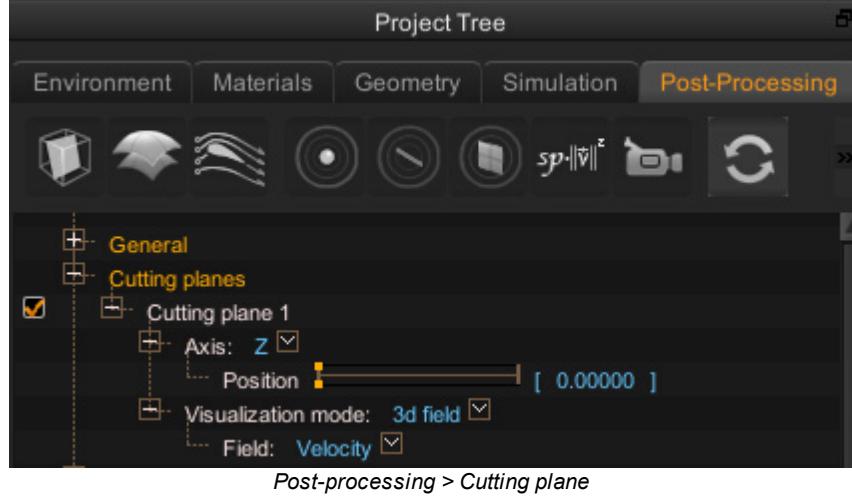


**Please note:** The project has to be saved before you are able to load the data.

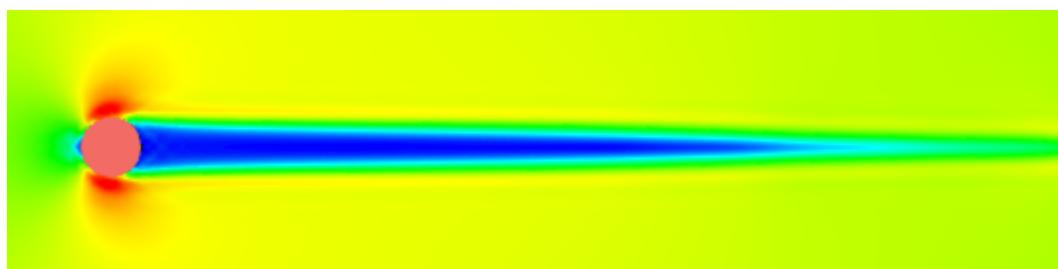
### 4.1 Visualise the velocity field

Create a cutting plane to visualise the velocity field. To do so, please 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**



- (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 frame = 100 corresponding to time = 2 s and observe the velocity field:

Interpolated velocity field at  $t = 2$  s

**Please note:** Even if the flow has already stabilised at  $t = 2$  s, the Karman vortex streets do not appear yet.

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

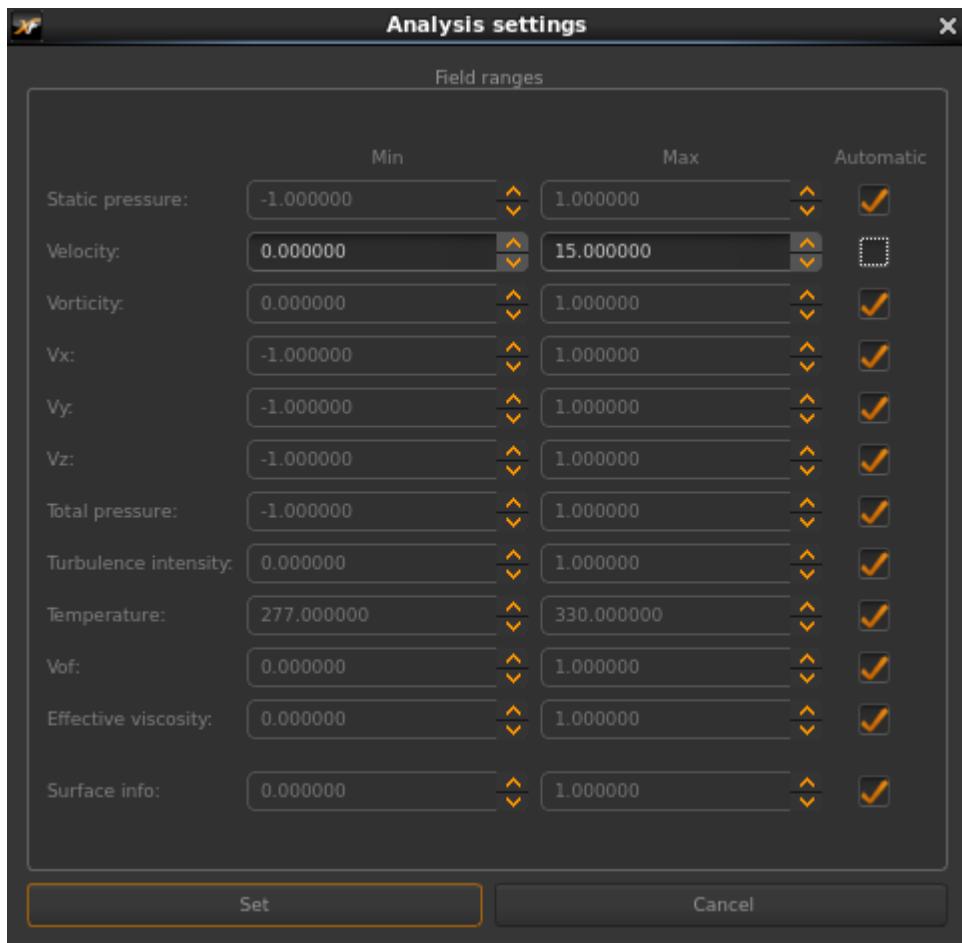
## 4.2 Use the playback controls to navigate through the transient results

- (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  $[0,15] \text{ m}\cdot\text{s}^{-1}$  by disabling the automatic range and inputting the minimum and maximum values.

## Step 4: Post-processing

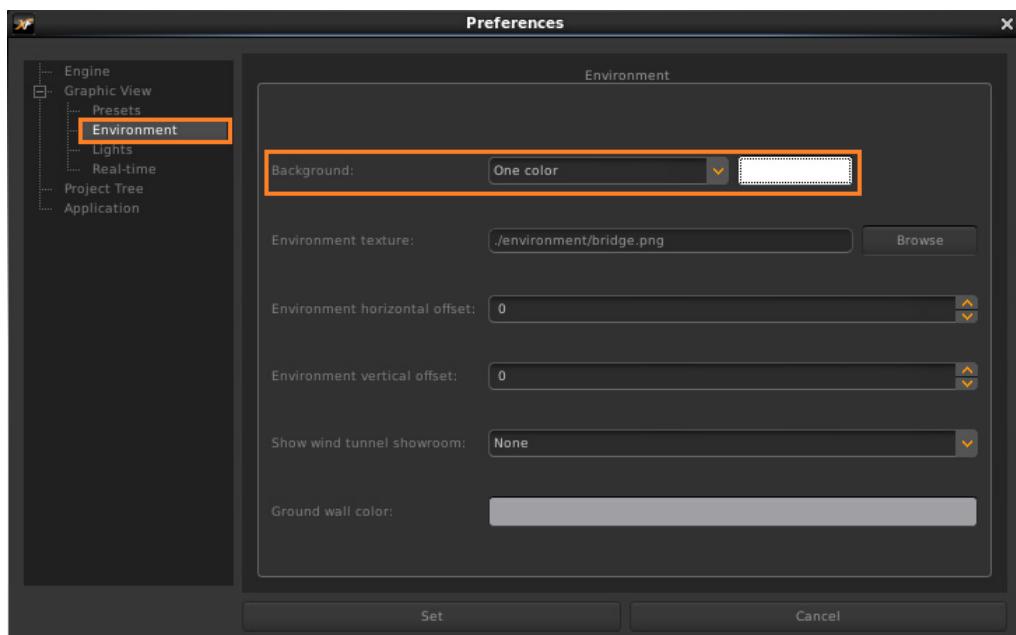


Play the frame sequence again.

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



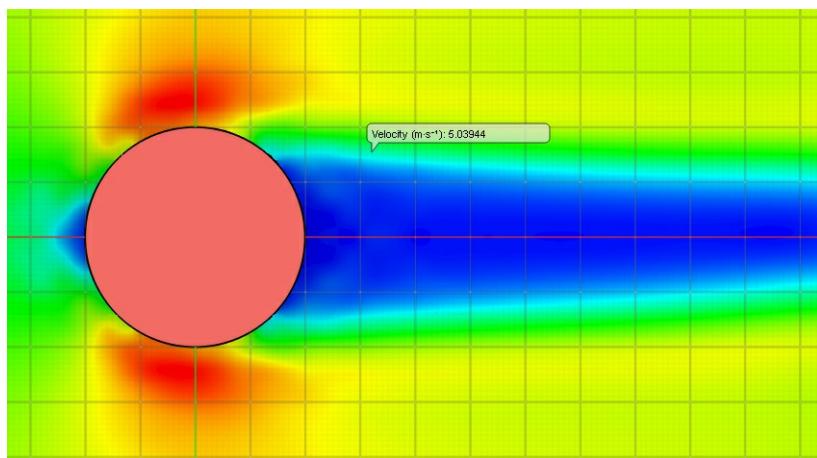
In the **Preferences** window, click on the cross of **Graphic View** to pull down its options Environment and Lights. Click on **Environment**, select **One colour** background, click on the coloured rectangle at the right hand side and pick the white colour in the palette.



#### 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.



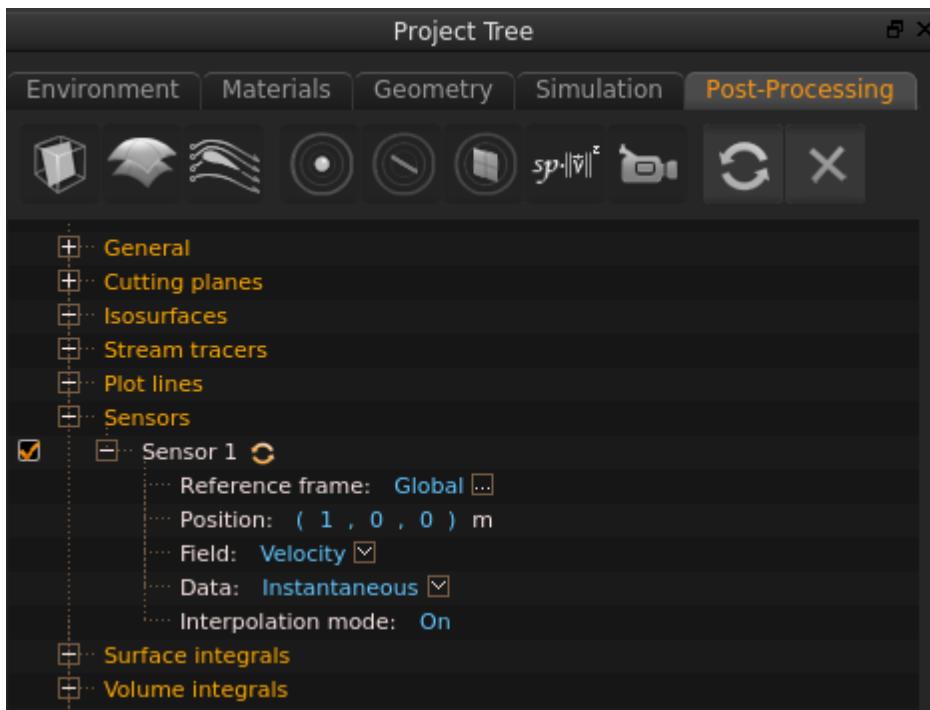
#### 4.5 Create a sensor

[Post-Processing > Sensors](#)

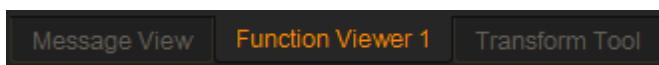
(a) Create a sensor: [Post-Processing > right clicking on Sensors > Add sensor](#)

The new sensor is automatically called **Sensor 1**. Change its position to **(1, 0, 0)** and set the **Field** to **Velocity**.

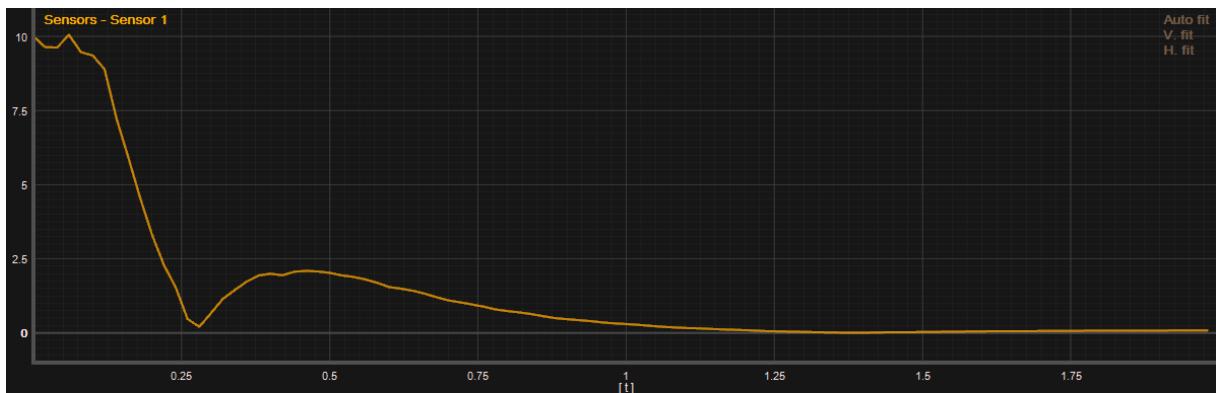
## Step 4: Post-processing



- (b) Hide **Cutting planes > Cutting plane 1** by unchecking the box in front of it  and make sure the checkbox of **Sensors > Sensor 1** is checked  in order to visualise the location of the sensor in the *Graphic View*.
- (c) Refresh the sensor: **Right click** on Post-Processing > Sensors > Sensor 1 > **Refresh icon**
- (d) Go to the *Function Viewer 1* window by switching to the **Function Viewer** tabulation:



- (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** > **Cylinder** > **Cx**

**Right click** on *Function Viewer* > **Shapes** > **Cylinder** > **Cy**

**Right click** on *Function Viewer* > **Shapes** > **Cylinder** > **Fx**

**Right click** on *Function Viewer* > **Shapes** > **Cylinder** > **Fy**

**⚠ Please note:** With this coarse resolution it is not possible to capture the correct physics and the numerical results are far from the correct ones ( $C_x \approx 1$ ). See the next section to learn how to refine close to the cylinder walls.

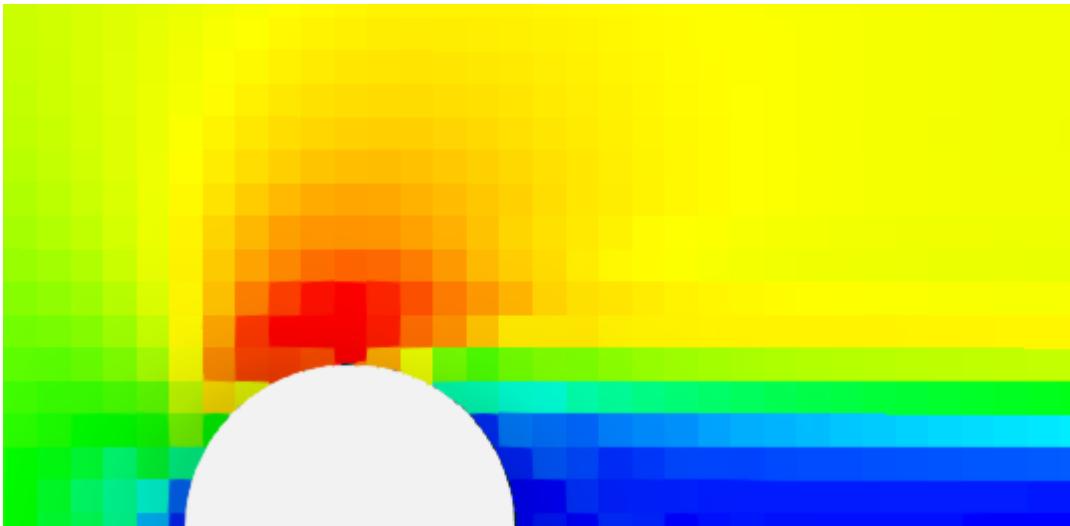
#### 4.7 Observe the resolution

- (a) Show the **Cutting plane 1** by checking its box

## Step 4: Post-processing

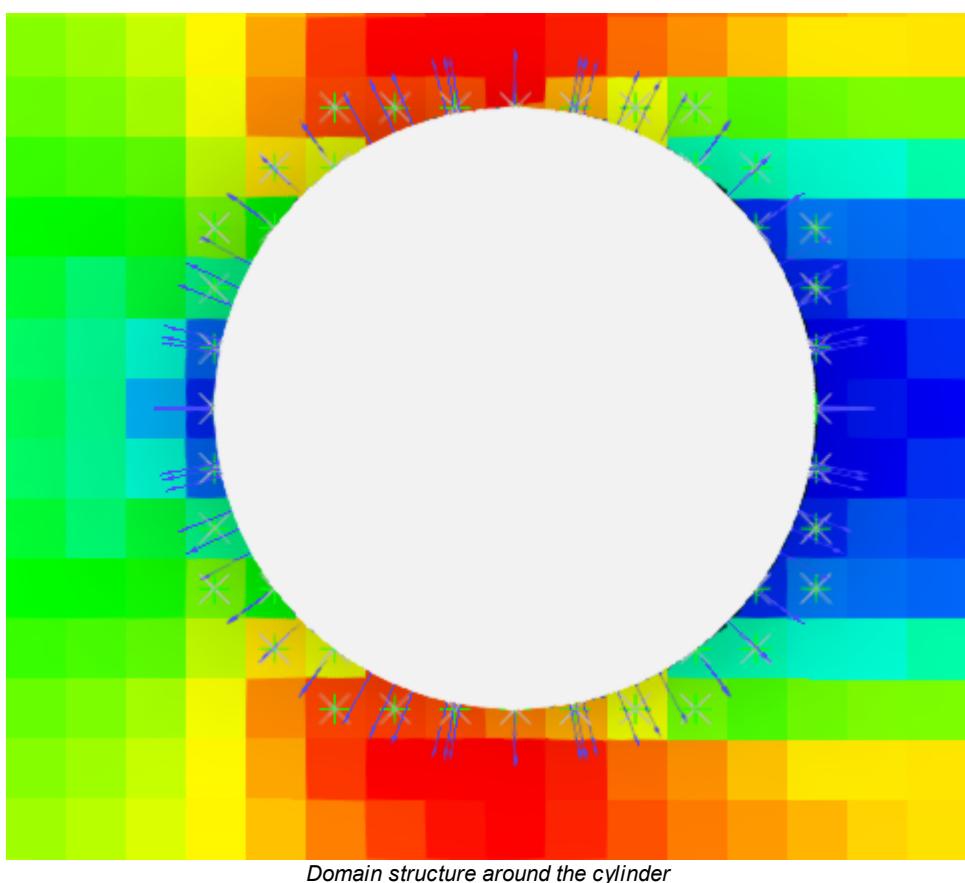
- (b) In **Post Processing > General > Show** set the **Interpolation mode** to **Off**.

Now the velocity field is displayed as a pixelation, being each "pixel" a cell of the lattice. In this way, you can distinguish the size of the cells.



*Velocity field with interpolation turned off*

- (c) plot the domain structure by checking the box  in **Post-Processing > General > Show > Domain structure**



The crosses "x" show the distance between each element of the lattice next to the walls and allows to estimate how fine or coarse is your simulation. The arrows show the wall normals.

In this case one can realise 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 cylinder must show a periodic flow motion developing in the wake of the cylinder as a result of boundary layer vortices being shed alternatively from either side of the cylinder. This regular pattern of vortices in the wake is called Karman vortex street.

The [Step 5: Refine the resolution](#) will teach how to refine the solution in order to observe the Karman vortex street.

## Step 5: Refine the resolution

We will now run the simulation using multi-resolution, i.e. a finer resolution close to the cylinder walls to better

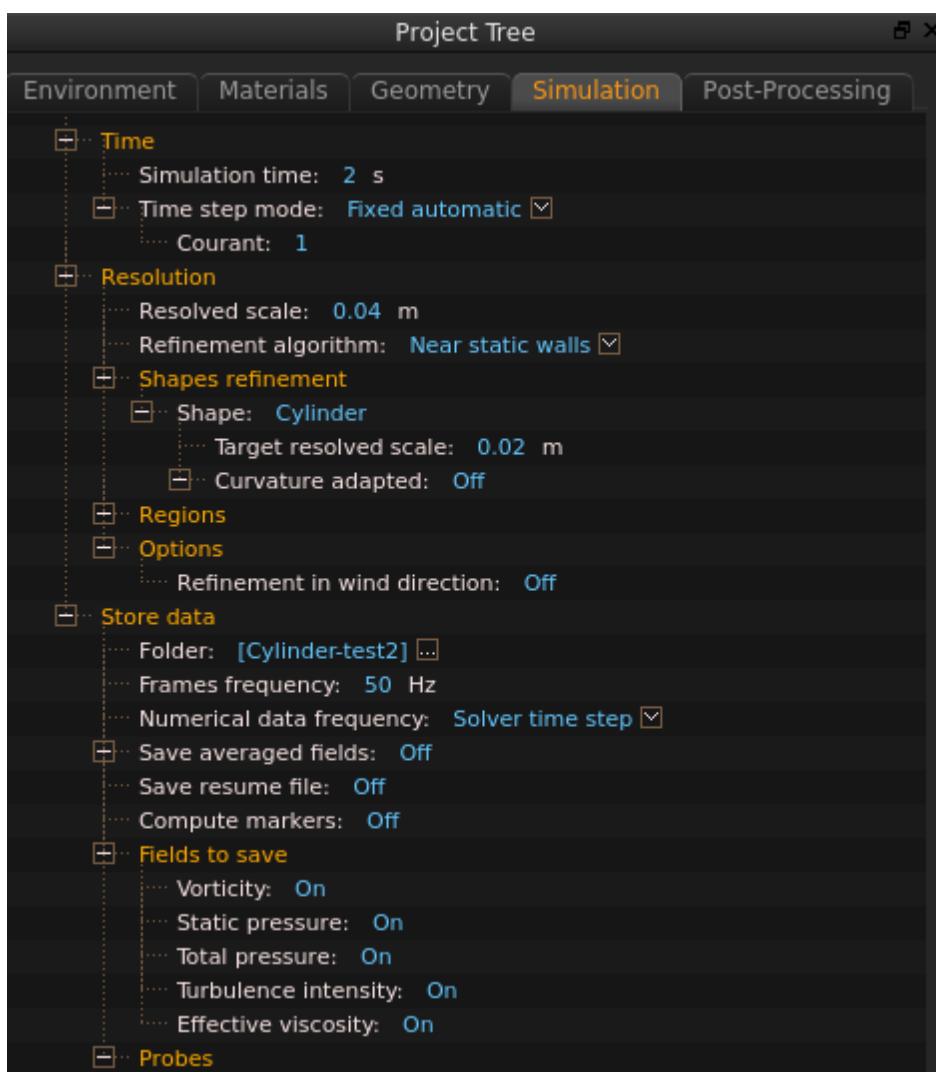
## Step 5: Refine the resolution

resolve the flow velocity gradients. As you have seen in the previous simulation, the flow has stabilized but the Karman vortex streets do not appear. The goal of this section is to capture the Karman vortex streets refining the solution.

### 5.1 Refinement algorithm

From the project defined in Step 3, change the following parameters in **Project Tree > Simulation**:

- (a) **Time > Simulation time:** **2 s** (as in the previous case)
- (b) **Resolution > Resolved scale:** **0.04 m** (as in the previous case)
- (c) **Resolution > Refinement algorithm:** **Near static walls**
- (d) **Resolution > Shapes refinement > Shape: Cylinder > Target resolved scale:** **0.02 m**
- (e) **Resolution > Shapes refinement > Shape: Cylinder > Curvature adapted:** **Off**
- (f) **Store data > Folder:** Change the name of the folder, where the results will be stored, to "**Cylinder-test2**"
- (g) **Store data > Frames frequency:** **50 Hz** (as in the previous case)
- (h) **Store data > Fields to save:** **On**, for all the fields



Project Tree &gt; Simulation

## 5.2 Save the project as a new project file

Main menu > File > Save project as or  in Toolbar File with a different name.

 **Tip:** It is recommended that you save the project with a different name because the settings have changed.

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

## 5.3 Run the new refined case

## Step 5: Refine the resolution

Press the **Run button > Start computation**

### 5.4 Post-process the results of the refined case

Display the structure of the multi-resolution:

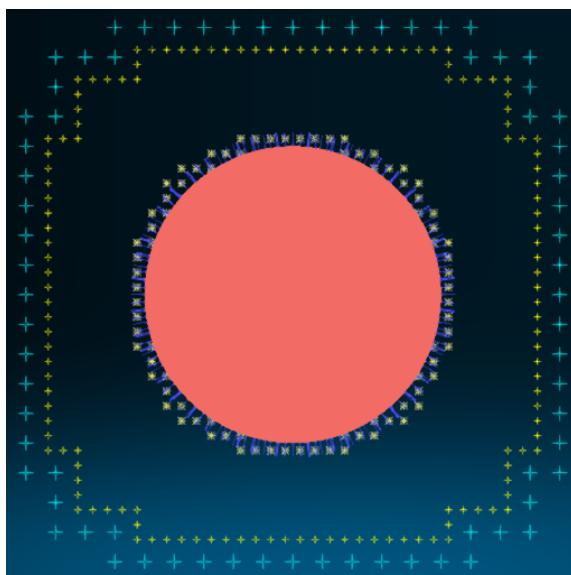
- (a) Locate the time bar on a calculated frame, e.g. frame 100
- (b) Uncheck (deactivate) every item of the **Post-Processing** tab of the *Project Tree*
- (c) Check (activate) **Post-Processing > General > Show > Domain structure**, to show the domain structure. This will be similar to that shown in the figure below:



*Domain structure at  $t = 2$  s, using the Near static walls refinement*

where, **x** indicates the border between fluid and wall, and **+** indicates the border of a region with uniform resolution and also indicates the size of the cells in this region.

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



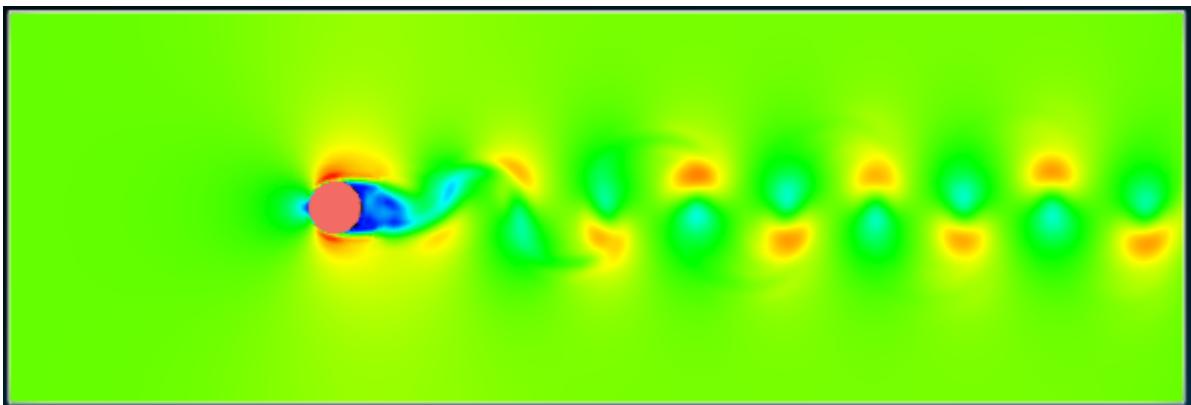
*Border between finest lattice level (yellow) and coarser lattice level (blue)*



**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 Display now the velocity cutting plane

- Disable the domain structure by unchecking the box: **Post-Processing > General > Show > Domain structure**
- Show the cutting plane 1 by checking the box: **Post-Processing > Cutting planes > Cutting plane 1**, and select:
  - Axis: Z**
  - Visualisation mode: 3d field**
  - Visualisation mode > Field: Velocity**
- Load the frame 100 to see the velocity field at time  $t = 2$  s, shown in the figure below:



*Velocity field showing the Karman vortex streets at frame 100 ( $t = 2$  s)*

The Karman vortex streets are now appearing because of the resolution that is fine enough around the cylinder walls to capture the vortex separation.

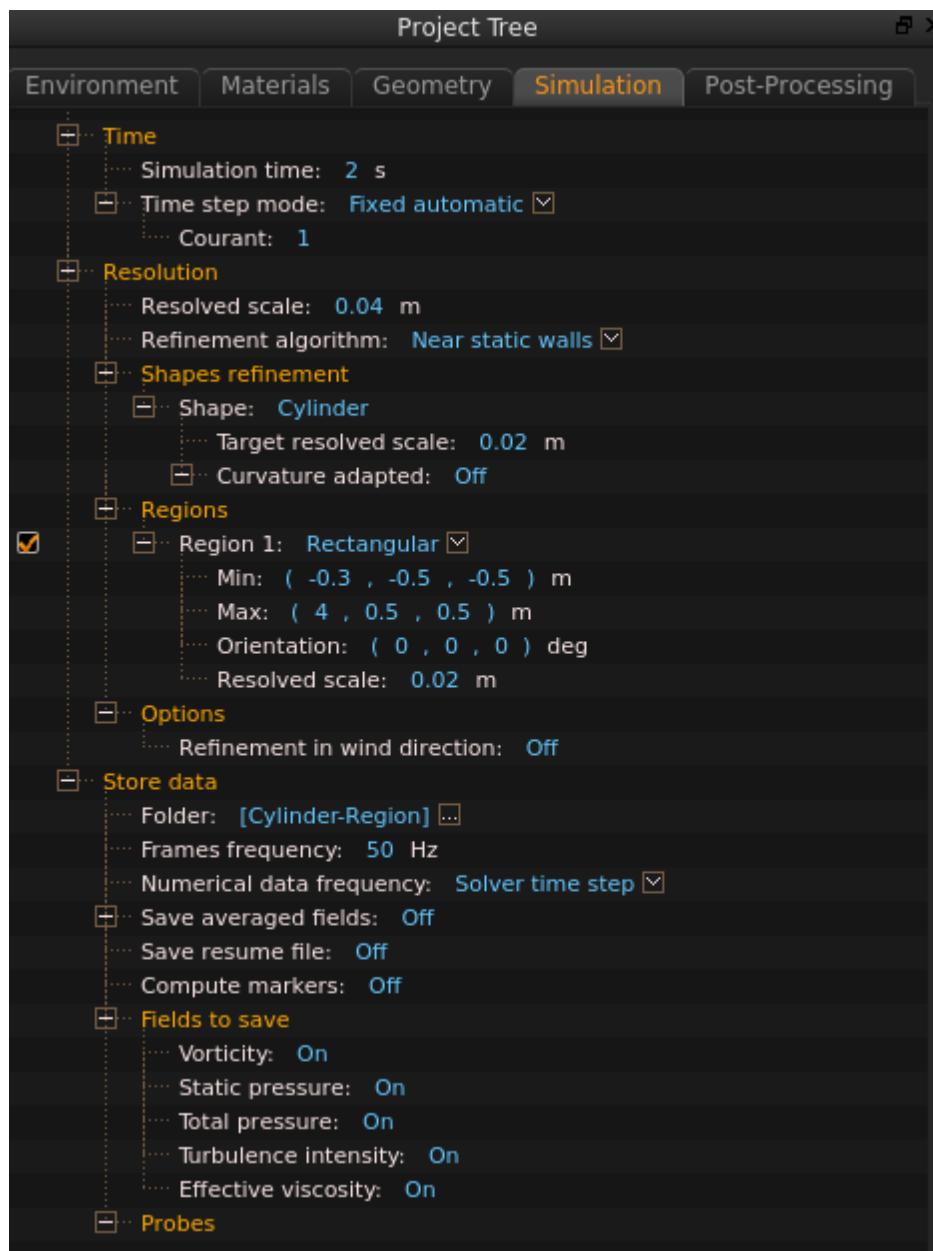
## 5.6 Create a region of refinement

Now we are going to define a fixed region of refinement to refine not only the cylinder but also the wake region downstream.

- Time > Simulation time: 2 s**
- Project Tree > Simulation > Resolution > Refinement algorithm: Near static walls**
- Project Tree > Simulation > Resolution > Shapes refinement: Cylinder > Target resolved Scale: 0.02 m**
- Project Tree > Simulation > Resolution > Regions: right click and select to Add region**

## Step 5: Refine the resolution

- (e) Project Tree > Simulation > Resolution > Regions > Region 1: **Rectangular**
- (d) Project Tree > Simulation > Resolution > Regions > Region 1 > Min: (-0.3, -0.5, -0.5) m
- (e) Project Tree > Simulation > Resolution > Regions > Region 1 > Max: (4, 0.5, 0.5) m
- (f) Project Tree > Simulation > Resolution > Regions > Region 1 > Orientation: (0, 0, 0) deg
- (g) Project Tree > Simulation > Resolution > Regions > Region 1 > Resolved scale: 0.02 m
- (h) Project Tree > Simulation > Store data > Folder: Change the name of the folder, where the results will be stored, to "**Cylinder-Region**"
- (i) Project Tree > Simulation > Store data > Frames frequency: 50 Hz



(j) Save the project.



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

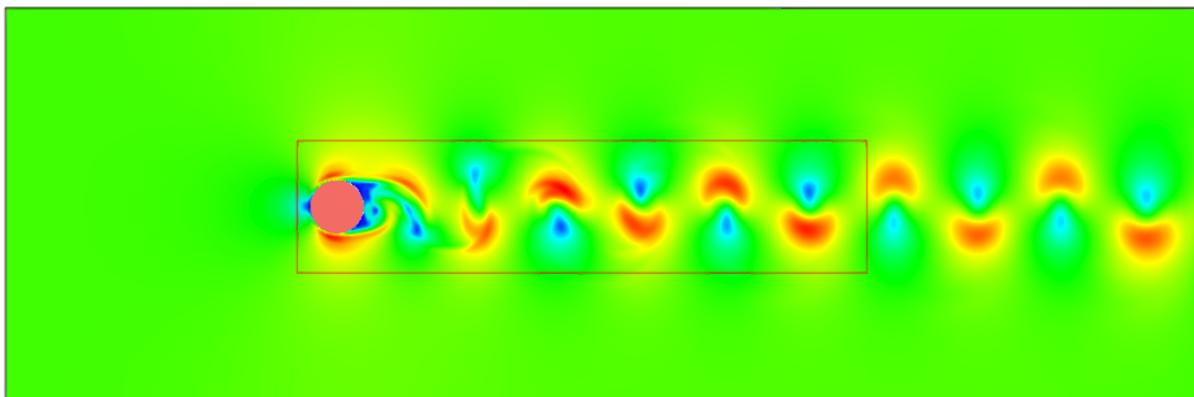
(k) Run the simulation

(l) Enable the domain structure by checking the box: Post-Processing > General > Show > **Domain structure**

(m) Show the cutting plane 1 by checking the box: Post-Processing > Cutting planes > **Cutting plane 1**, and select

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

(n) Load the frame 100 to see the velocity field at time  $t = 2$  s, shown in the figure below:



*Fixed region of refinement*

The Karman vortex streets are also obtained in the same way than the previous simulation.



**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 wake.

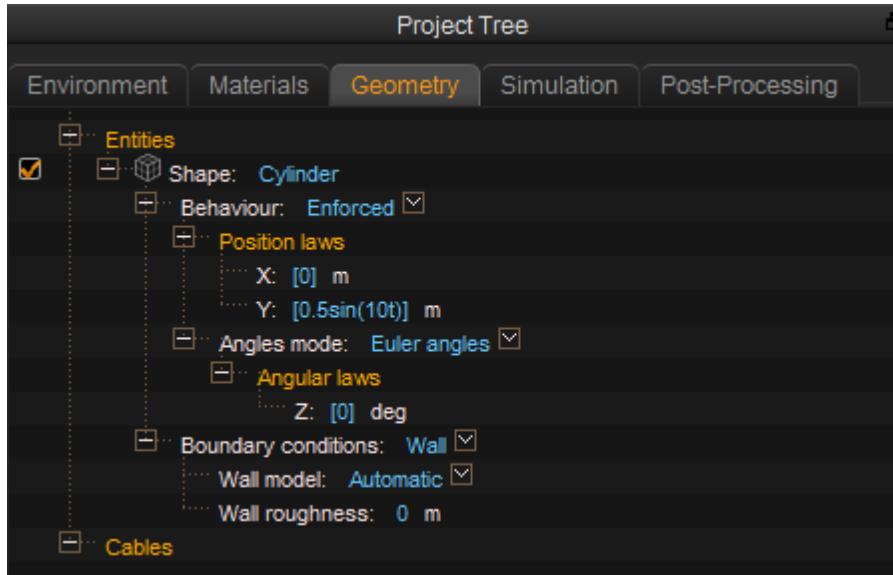
## Step 6: Moving cylinder - enforced motion

Now we are going to prescribe an enforced vertical motion to the cylinder through an analytic function.

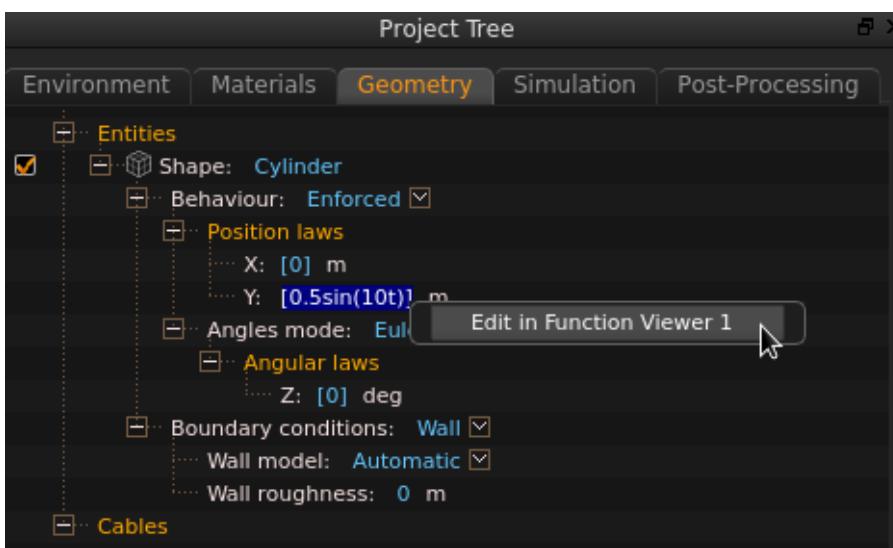
### 6.1 Set the shape motion

## Step 6: Moving cylinder - enforced motion

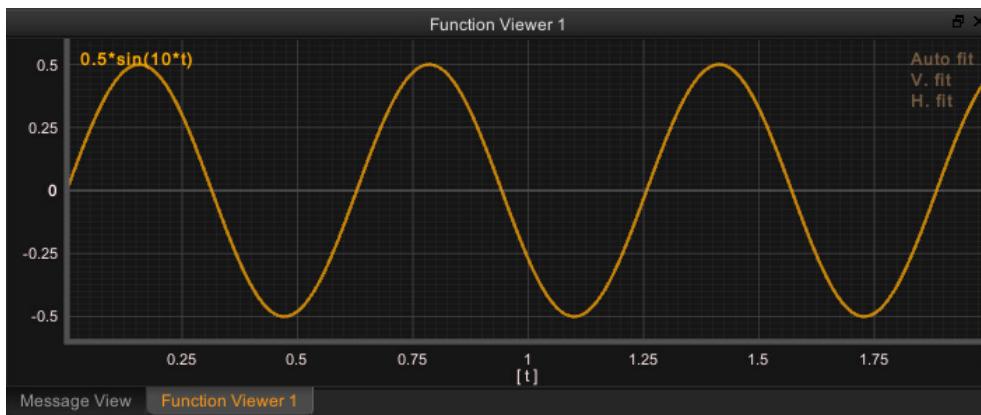
- (a) Project Tree > Geometry > Entities > Shape: Cylinder > Behaviour: Enforced
- (b) Project Tree > Geometry > Entities > Shape: Cylinder > Behaviour > Position law Y:  $0.5\sin(10t)$



- (c) Check the position law in the *Function Viewer* window by right clicking on the mentioned law and selecting: **Edit in Function Viewer**



- (d) Click **Auto fit** in the *Function Viewer* to adjust the scale of the graph to the size of the window. The position law is plotted below:



You can also check the behaviour of the cylinder if you play the frames timeline.

## 6.2 Set the refinement algorithm

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

- (a) Time > Simulation time: **2 s**
- (b) Resolution > Resolved scale: **0.04 m** (as in the previous case)
- (c) Resolution > Refinement algorithm: **Adaptive refinement**
- (d) Resolution > Shapes refinement > Shape: Cylinder > Target resolved scale: **0.02 m**
- (e) Resolution > Adaptive refinement > Wake resolution: **0.02 m**
- (f) Resolution > Adaptive refinement > Wake distance control: **Off**
- (g) Resolution > Regions: No refinement regions
- (h) Store data > Folder: Change the name of the folder to "**Cylinder-enforced**"
- (i) Store data > Frames frequency: **50 Hz**

## 6.3 Save the project as a new project file

Main menu > File > Save project as or  in Toolbar File with a different name.



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

## 6.4 Run the new refined case

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

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

Visualise the time evolution of the velocity field:

- (a) Show the cutting plane 1 by checking the box: Post-Processing > Cutting planes > Cutting

## Step 6: Moving cylinder - enforced motion

plane 1, and select:

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

(b) Go to the first frame  and press play 

(c) In the *Function Viewer* window, check that the vertical displacement of the cylinder is the prescribed one:

- Right click on the *Function Viewer* > Shapes > Cylinder > Py
- Go to the first frame and play forward to refresh the data in the *Function Viewer*

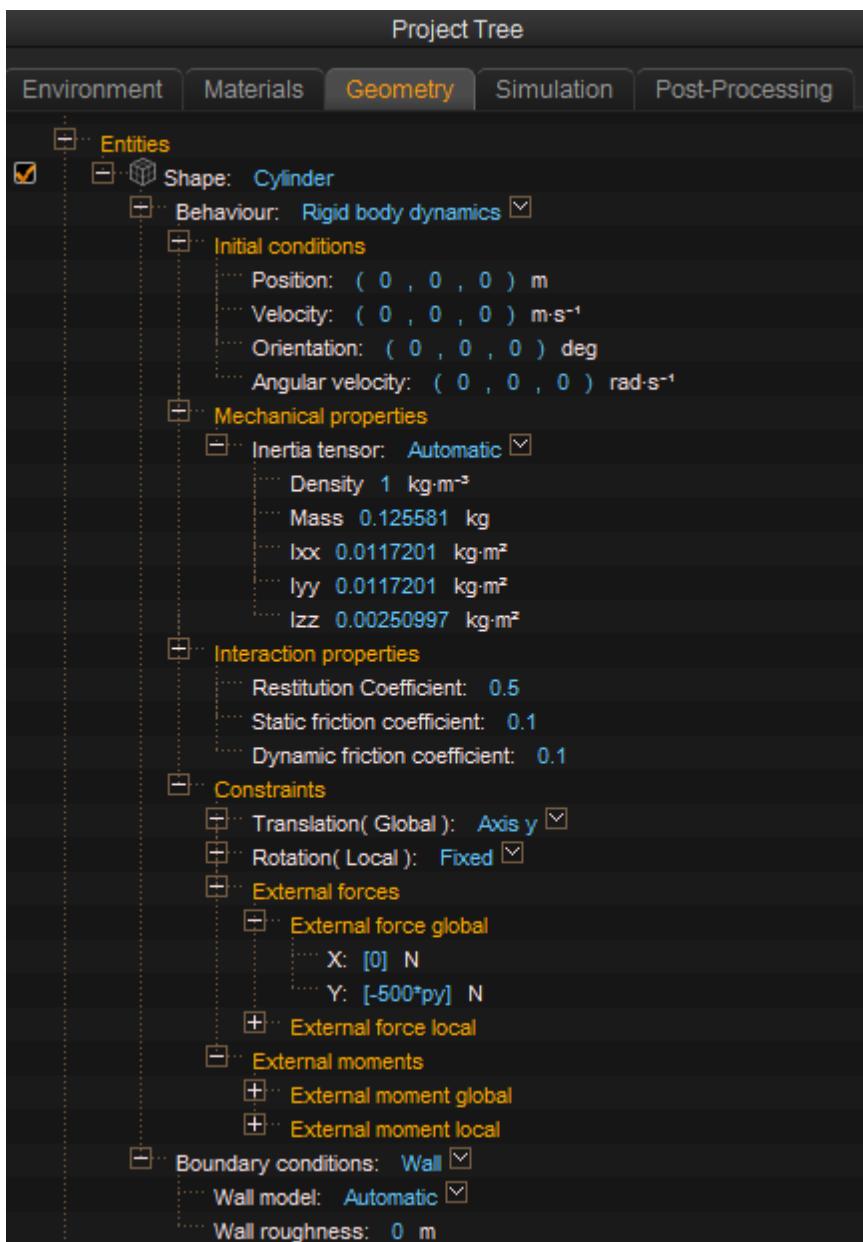
## Step 7: Moving cylinder - rigid body dynamics

We will now allow the cylinder to move vertically but only due to the forces exerted by the fluid. The cylinder movement will simultaneously affect the flow, and this is therefore a fluid-structure interaction problem.

### 7.1. Modelling the cylinder as joined to the coordinate origin by a spring:

Go to Project Tree > Geometry > Entities > Shape: Cylinder, and set up the tree as follows:

- Behaviour: Rigid body dynamics
- Behaviour > Mechanical properties > Inertia tensor: Automatic
- Behaviour > Mechanical properties > Inertia tensor > Density:  $1 \text{ kg}\cdot\text{m}^{-3}$
- Behaviour > Interaction properties: Leave the default values
- Behaviour > Constraints > Translation (Global): Axis y
- Behaviour > Constraints > Rotation (Local): Fixed (not allowed)
- Behaviour > Constraints > External forces global > Y:  $-500*py$  (external force of the spring, where py is the cylinder position in Y-axis direction).



## 7.2 Save the project as a new project file

(a) Change the folder name (**Simulation > Store data > Folder**) to "Cylinder-fsi".

(b) Save the project with a different name: **Main menu > File > Save project as** or in Toolbar File.

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

## 7.3 Run the case

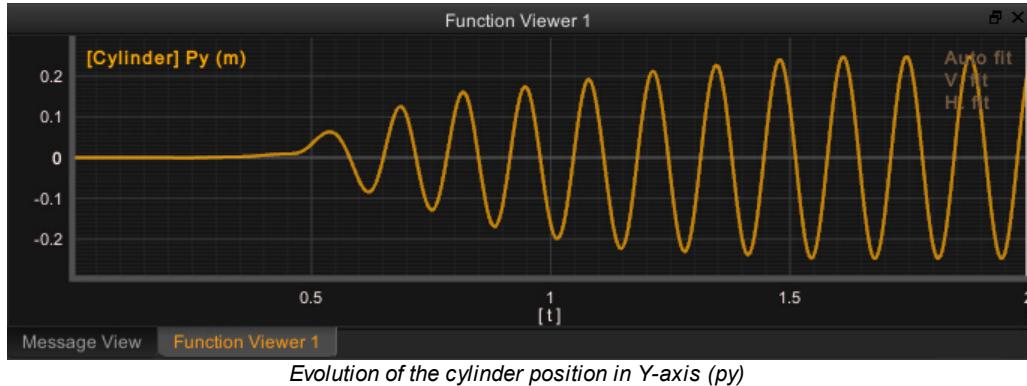
Press the **Run button > Start computation**

### 7.4. Post-process the results of the rigid-body-motion case

Once the calculation has finished, visualise the time evolution of the velocity field (see Step 6.5).

Display the vertical displacement of the cylinder:

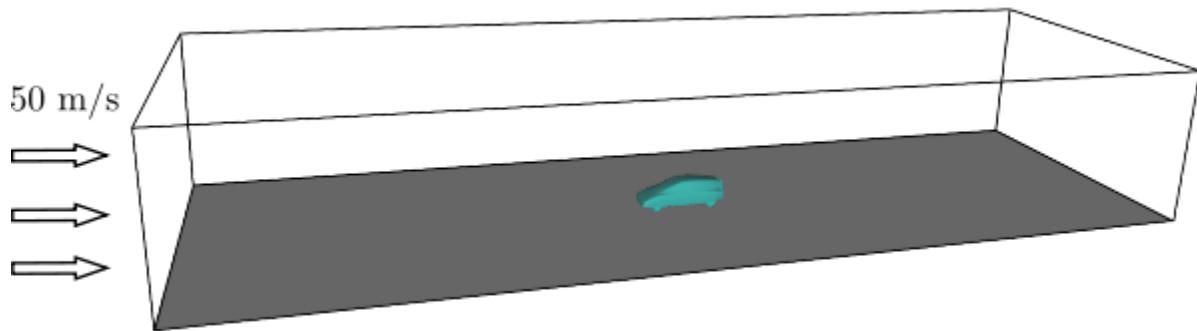
- Do **right click** on the *Function Viewer* window. Choose from the menu: **Shapes > Cylinder > Py** ; go to first frame and play forward to refresh the data in the *Function Viewer*



- Save the vertical displacement in a file, by **right clicking** on the *Function Viewer* window and choosing **Export current data**
- Check that the X and Z displacements, and all the rotational Euler angles are zero for the cylinder:
  - Right click** on *Function Viewer* window > **Shape > Cylinder > Px**
  - Right click** on *Function Viewer* window > **Shape > Cylinder > Pz**
  - Right click** on *Function Viewer* window > **Shape > Cylinder > Eux**
  - Right click** on *Function Viewer* window > **Shape > Cylinder > Euy**
  - Right click** on *Function Viewer* window > **Shape > Cylinder > 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
- Make 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.

Before starting the tutorial, please download the project data files from the Documentation section of XFlow website ([http://www.xflowcfd.com/index.php/client\\_area/documentation/view/1](http://www.xflowcfd.com/index.php/client_area/documentation/view/1)).

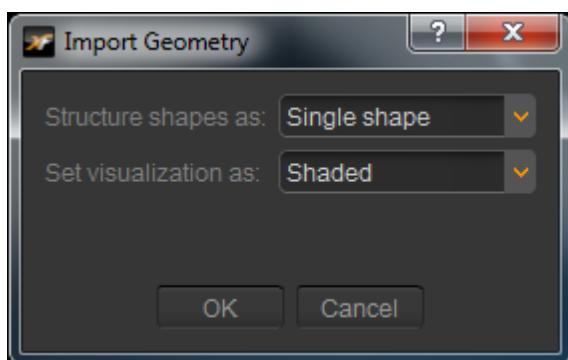
## Contents

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

## Step 1: Import geometry

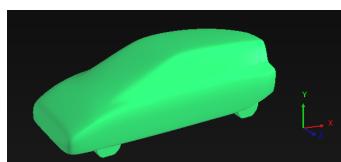
### 1.1 Import the geometry

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 "Single shape" structure, visualization "Shaded" and press **Apply to all**.

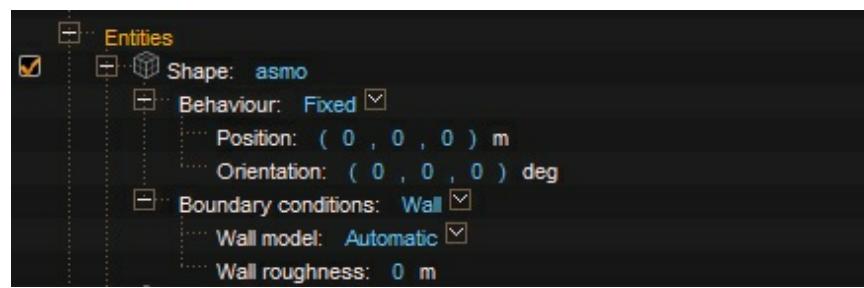


Import geometry dialog window

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*.



Asmo geometry shown in the Graphic View



Asmo geometry shown in the Geometry tree

### 1.2 Check the model

(a) 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

(b) 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

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

➤ Length (X direction): 0.809928 m

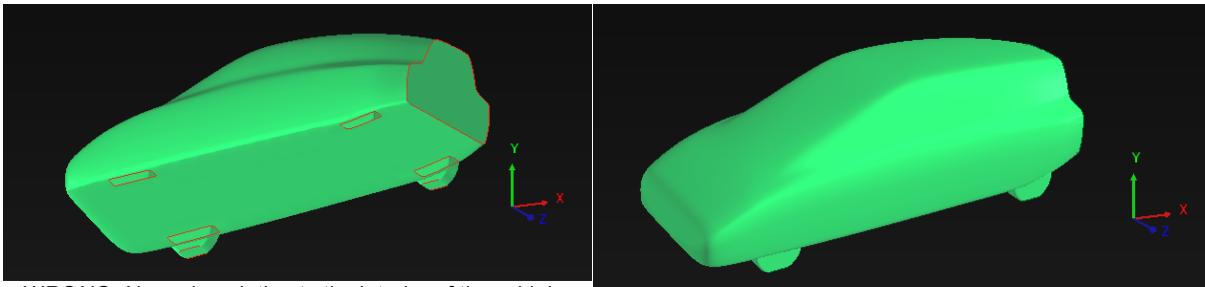
## Step 1: Import geometry

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

(e) Check the surface normals enabling the culling; thus select the geometry object, **right click** on the *Graphic View* window and choose **Graphic View Menu > Back-face culling**. Now reverse the orientation of the normals to learn the different representations of the vehicle depending on the normals orientation: **Graphic View Menu > 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. If the surface is coloured then its normals are pointing to the user, but if the surface looks like a wireframe, then the normals are pointing inside the geometry.

(f) Check that the model is free of holes: **Main menu > Geometry > Show/hide holes**, while displaying the geometry in bounding box mode: **Graphic View Menu > Visualisation mode > Bounding Box**

**⚠ Please note:** If the model has holes, fluid will leak inside. This fluid inside the geometry is initialised with the inlet velocity condition as the rest of the fluid but, because it is confined, generates pressure waves inside of the geometry that lead to wrong 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.

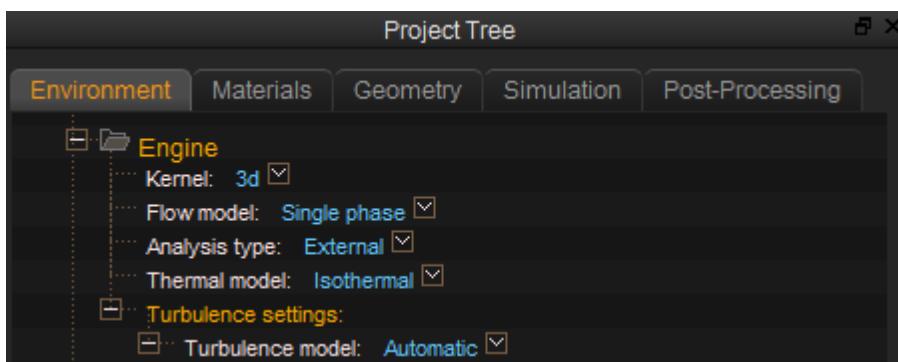
Some holes can be directly repaired in XFlow: **Main menu > Geometry > Healing**. See Tutorial 07 for more information about the healing feature.

## 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. You could 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 useful to simulate half vehicle

## Step 2: Problem setup

- (h) Wind tunnel > Boundary conditions: **Velocity**. Set the **Velocity laws** at inlet equal to **(50, 0, 0) m·s<sup>-1</sup>**



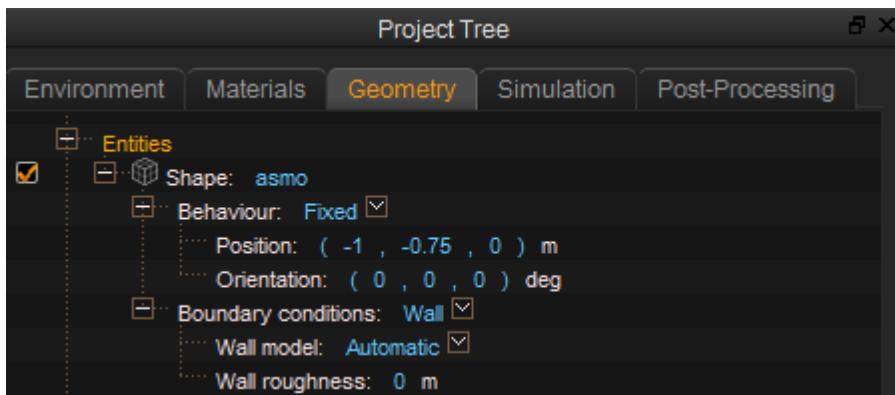
### 2.3 Material settings

Keep the default parameters of air in Project Tree > Materials.

### 2.4 Geometry settings

Project Tree > Geometry > Entities > Shape: **asmo**

- (a) The asmo geometry should be located just over the ground, leaving some space behind the vehicle for the wake. So, change the **Position** of the object to (**Behaviour > Position**) **(-1, -0.75, 0) m** and make sure that the wheels are touching the ground,
- (b) Set the **Boundary Conditions** to **Wall**, with **Wall model: Automatic** and **Wall roughness: 0 m**.



## 2.5 Simulation settings

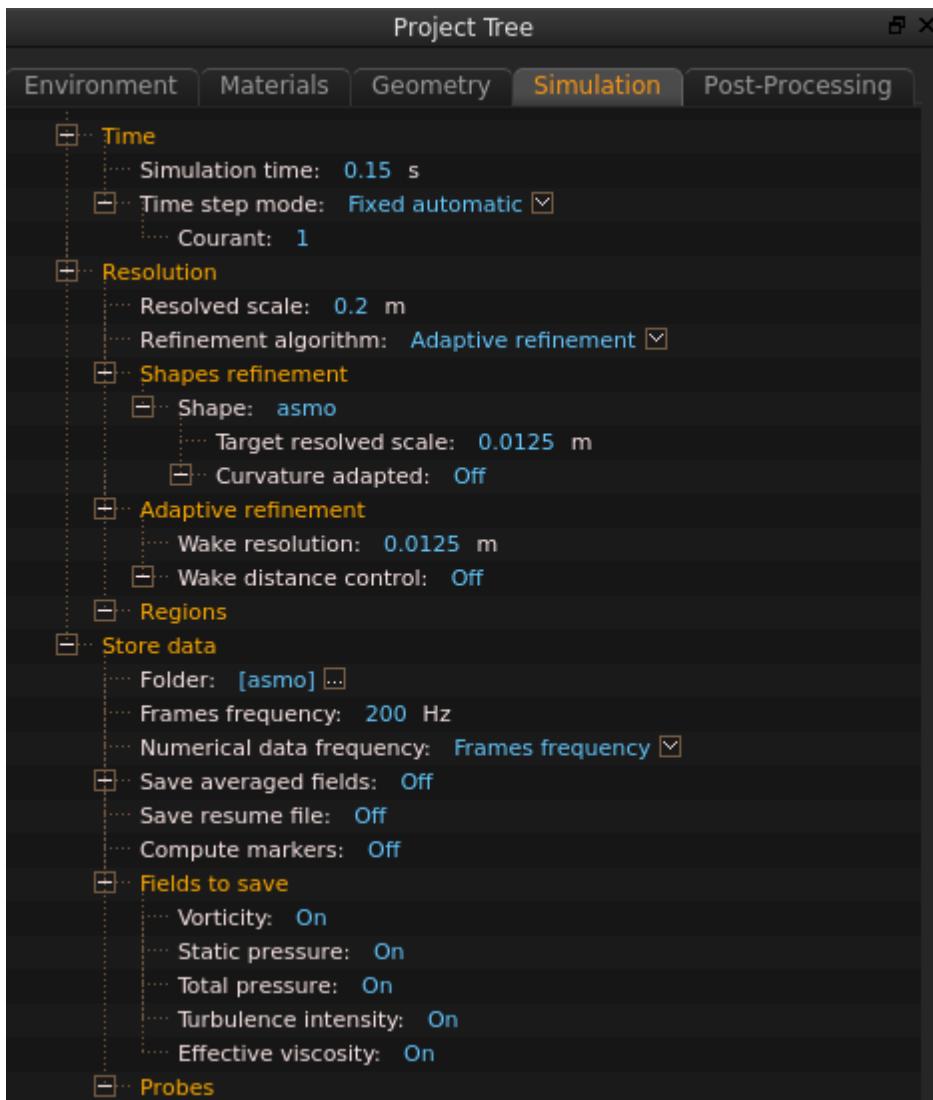
Project Tree > Simulation:

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

**!** **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) **Store data** in the **Folder "asmo"**, with 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**.

## Step 2: Problem setup



**⚠ 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 length of the vehicle divided by the 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.



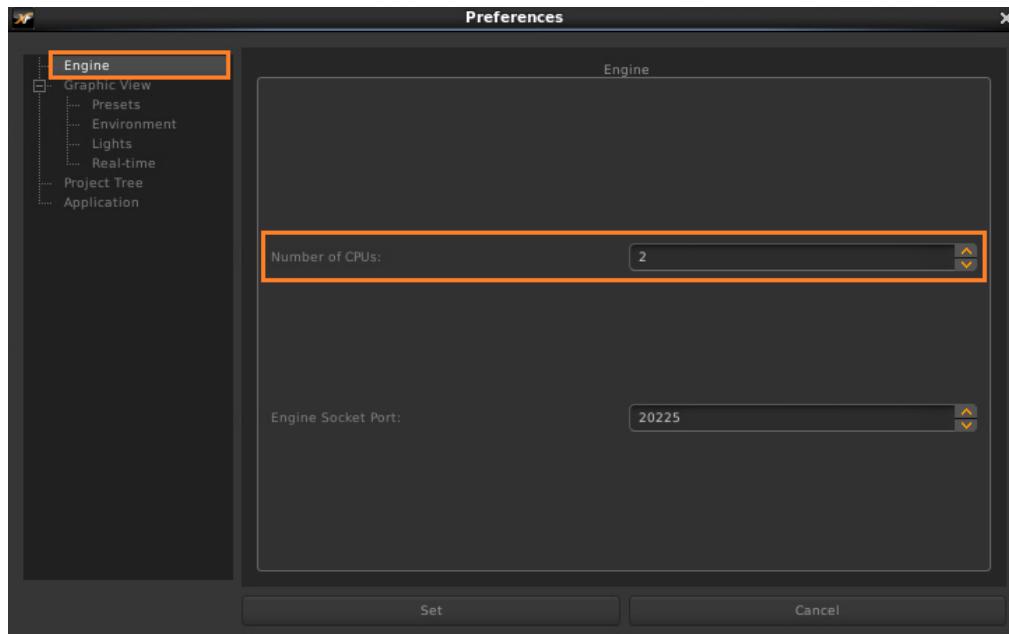
**Tip:** You can directly load the setup of this problem from the project file `asmo.xfp`

### 3.2 Set number of CPUs

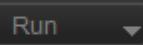
Set the number of CPUs, to be used for the calculation, in the preferences of the engine:

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%.

### Step 3: Run



#### 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 `asm0`.

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 for calculating aerodynamic coefficients)
- Time step size

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 as the **Refinement algorithm** is set to *Adaptive refinement*. Each line corresponds to a new solver time step.

##### Example: .log file output.

Saving data...

```
[[Data file]] 0 done!!! | Frame wall clock time[0]s | Overall wall clock time[0]s | Num elements[102747]
Sim. time [0.000333333]s | Stability param. [0.0618017] | Wall clock time [7.223]s | Num elems [127590]
Sim. time [0.000666667]s | Stability param. [0.0822159] | Wall clock time [4.961]s | Num elems [107437]
```

...

```
Sim. time [0.005]s | Stability param. [0.0649] | Wall clock time [4.96]s | Num elems [109425]
```

Saving data...

```
[[Data file]] 1 done!!! | Frame wall clock time[75.208]s | Overall wall clock time[75.208]s | Num elements [109425]
```

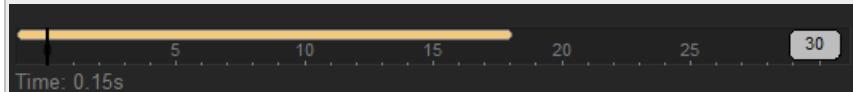
```
Sim. time [0.00533333]s | Stability param. [0.0630606] | Wall clock time [5.397]s | Num elems [109691]
```

...

Every time a new frame is calculated, XFlow advises you with the following message: `[[Data file]] #frame done!!!`; then you can select that frame 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 that you set the timeline to an empty frame when you are not post-processing, as shown below:



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

## Step 4: Post-processing

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

### 4.1 Visualise the solution

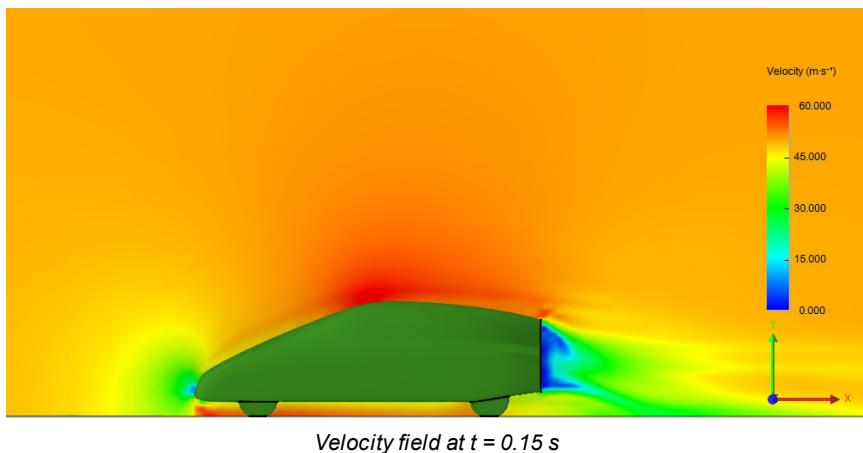
#### 4.1.1 Visualise a flow variable on a cutting plane

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

- (a) Right-click on Cutting planes > Add cutting plane or press in the Post-Processing toolbar
- (b) Plot the Cutting plane with a Z-axis normal, and set its position at the middle of the fluid domain:  
**Cutting plane 1 > Position: 0.5**
- (c) Visualise the velocity in the cutting plane by setting: **Cutting plane 1 > Visualisation mode: 3d field**, and choose the **Field: Velocity**
- (d) Activate the interpolation mode: **General > Interpolation mode: Convolution**
- (e) Set the velocity range to [0,60]: Main menu > Simulation data > Analysis settings or press in Toolbar Data Processing.
- (f) Select the right view
- (g) Press Play

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

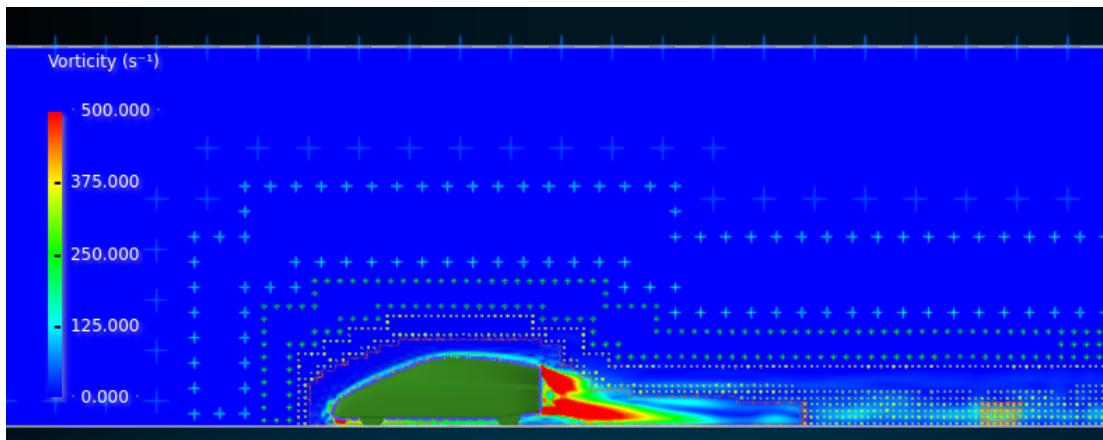
## Step 4: Post-processing



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

Repeat the previous step but now enable the **General > Show > Domain structure** option to observe the lattice domain evolution against the time. The wake is now refined dynamically behind the car based on the vorticity gradients.



Domain structure showing the adaptive wake refinement over a cutting plane of vorticity

### 4.1.3 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  $\pi$

$v_x(x, y, z)$	→ X-component of velocity at a discrete point of the domain, given by coordinates (x,y,z)
$v_y(x, y, z)$	→ Y-component of velocity at a discrete point of the domain, given by coordinates (x,y,z)
$v_z(x, y, z)$	→ Z-component of velocity at a discrete point of the domain, given by coordinates (x,y,z)
$v_{mod}(x, y, z)$	→ 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
$v_{mod}$	→ Module of the velocity
$\omega$	→ Vorticity
$cf$	→ Skin friction coefficient
$ti$	→ Turbulence intensity
$t_p$	→ Total pressure
$T$	→ Temperature
$n_x$	→ X-component of the surface normal
$n_y$	→ Y-component of the surface normal
$n_z$	→ 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

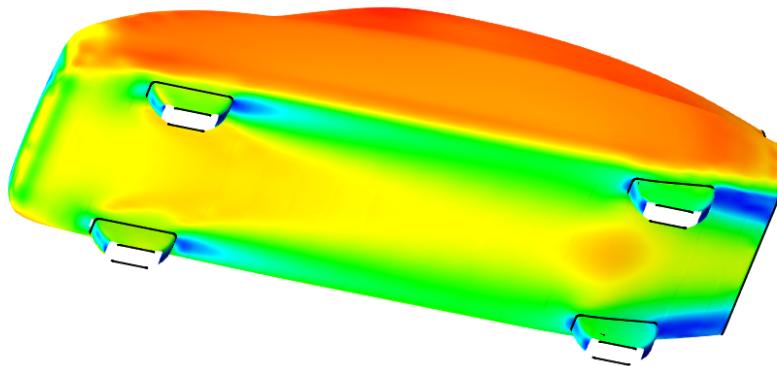
- (b) The user can define the custom field in **Custom Fields > Custom field 1 > Expression**. For instance, to define the pressure coefficient  $C_p$ : **Expression: [sp/(0.5\*rho\*50^2)]** ( $V_{ref} = V_{inlet} = 50 \text{ m}\cdot\text{s}^{-1}$ )
- (c) Switch the visualisation field of the **Cutting plane 1** to **Field: Custom field 1** to observe the pressure coefficient on the cutting plane.
- (d) Hide the cutting plane by unchecking its box .

#### 4.1.4 Visualise a flow variable on the vehicle surface

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 checking the box , to see the projection of the velocity over the vehicle in the **Graphic View**.

## Step 4: Post-processing



*Velocity field on the vehicle surface at  $t = 0.15 \text{ s}$*

(d) Hide the Surface info by unchecking its box

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

- **LIC** (Line Integral Convolution) of the velocity field
- **C<sub>p</sub>** pressure coefficient
- **C<sub>f</sub>** skin friction coefficient
- **Y<sub>+</sub>** dimensionless distance from the wall
- **P<sub>+</sub>** wall pressure
- **Velocity direction**

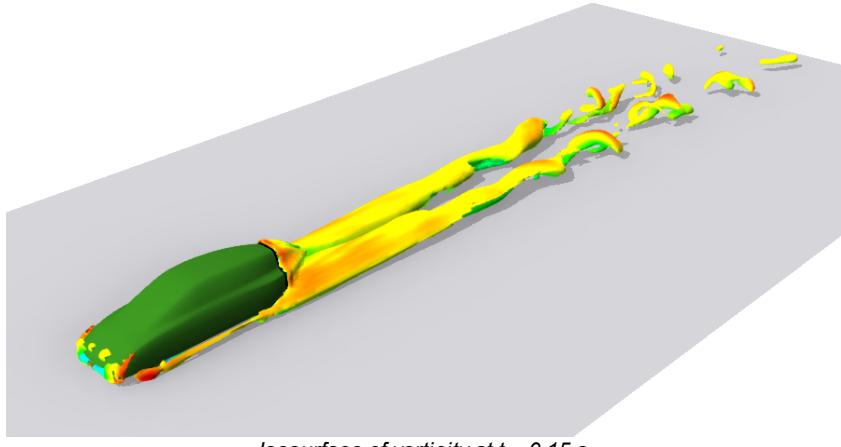


*LIC at  $t = 0.15 \text{ s}$*

### 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: Vorticity**
- (c) Set the **Value** to  $500 \text{ s}^{-1}$ . **Isosurfaces > Isosurface 1 > Value: 500**. XFlow will plot an isosurface representing all the fluid elements that have a vorticity value of  $500 \text{ s}^{-1}$ .
- (d) The user can colour the isosurface by other fields (e.g. velocity) rather than the isosurface field in: **Isosurfaces > Isosurface 1 > Coloured by field: Velocity**
- (e) Refresh the isosurface.

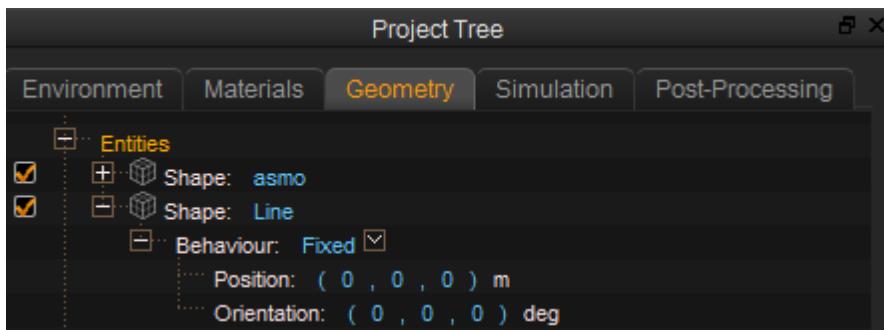


- (f) Go to point (c) and change the **Value** (between the min and the max values of vorticity colour bar). Recompute the isosurface by **right-clicking** on **Isosurface 1** and select > **Recompute isosurface** or press , which appears next to **Isosurface 1**. It may take a few seconds to calculate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

#### 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 **Geometry > Entities** section of the **Project Tree**.
- (b) Let the line **Behaviour as Fixed**.

## Step 4: Post-processing



(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

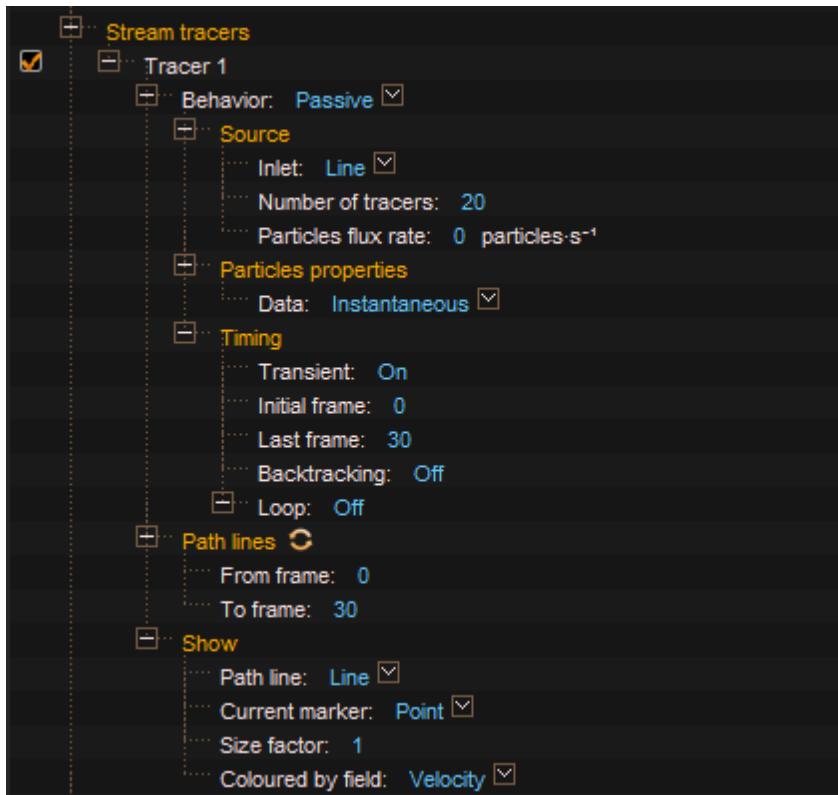
Transient: On

Initial frame: 0

Last frame: 30

Backtracking: Off

Loop: Off



- (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 red colour (over the yellow timeline), this is the timeline of to the stream tracers. Press play  and observe the particles trajectory.
- (g) Show the streamlines in **Post-Processing > Stream tracers: 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: Tracer 1 > Show > Path line: Line**
- (i) **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
- (j) Hide the stream tracer by unchecking its box



**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 checking the box
- (b) Select **Visualisation mode: Vectors**

## Step 4: Post-processing

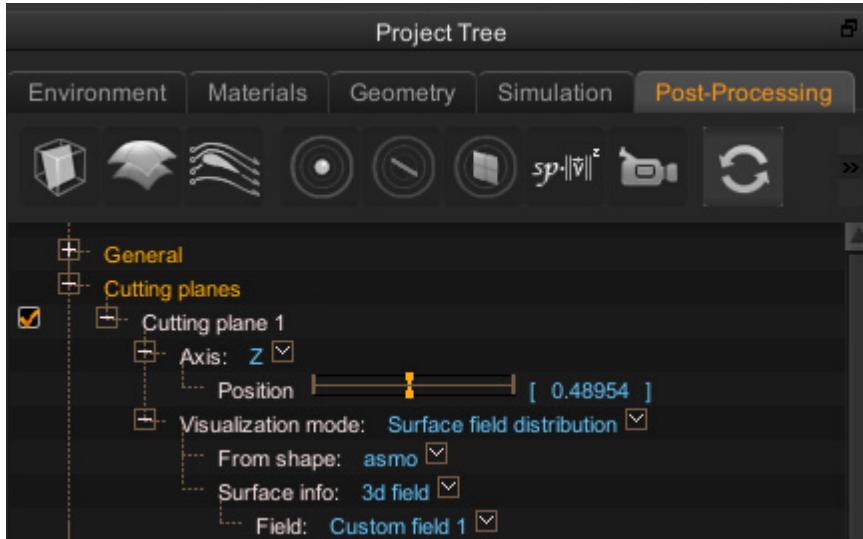
(c) Select **Field: Velocity**

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

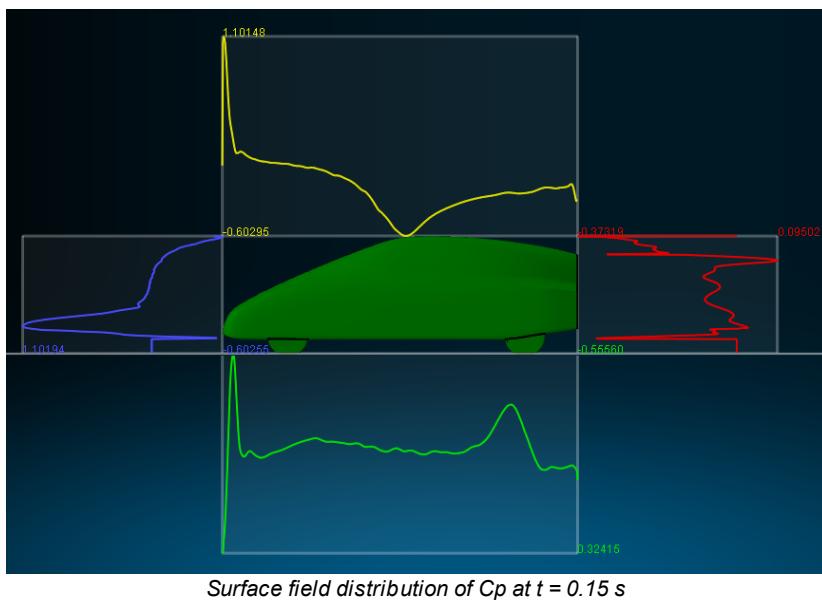
### 4.1.8 Visualise surface field distribution

To project on the vehicle surface the distribution of the pressure coefficient  $C_p$ , do:

(a) Switch the cutting plane visualisation mode to **Surface field distribution** and select **Surface Info:  $C_p$**



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



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).

(c) Hide the cutting plane by unchecking its box

#### 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 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*.

- (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:

$C_x, C_y, C_z$	→ Force coefficients in each direction
$F_x, F_y, F_z$	→ Force in each direction

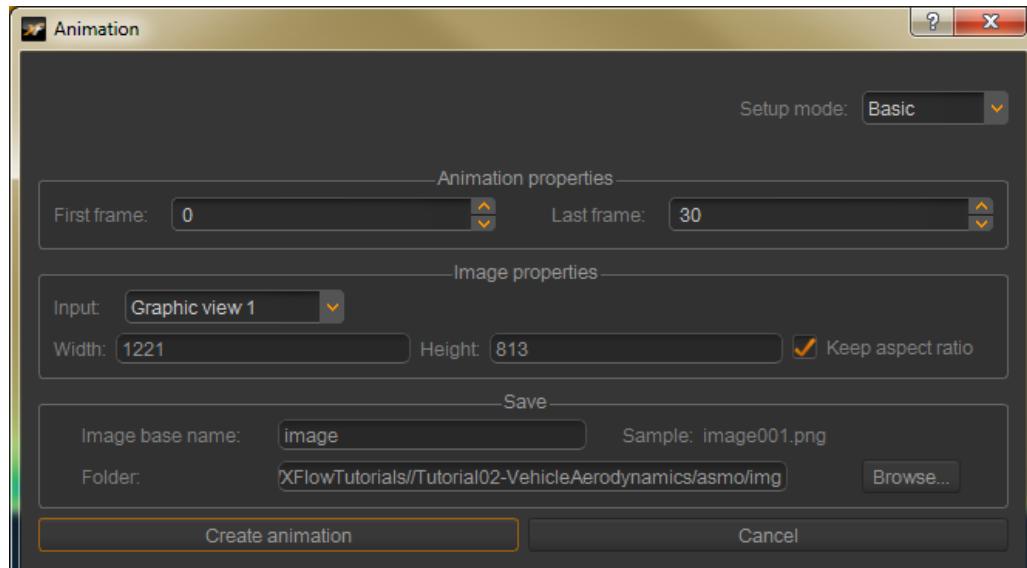
**⚠ Please note:** The drag coefficient obtained with this coarse resolution is far from the experimental values 0.153-0.158. You should refine the resolution near the walls and in the wake (e.g. target scale around 0.0025 m) to properly resolve the turbulent wake and obtain accurate drag values. The 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 >**

## Step 4: Post-processing

**Post-Processing > Animation**, or  in the Data Processing toolbar.



*Basic animation wizard*

### 4.3.1 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 30
- (d) Select the **Input:** "Graphic View 1" for creating images just of the Graphic View, or "GUI view" for creating images of the entire GUI
- (e) Enter an **Image base name** for the image sequence, the path of the **Folder** where to save it and press **Create Animation**. It will create 30 images in the directory `C:/XFlow/Tutorial02/asmo/img` that you can use to make an animation.

 **Tip:** The assemblage of the image sequence to a video file can be done e.g. with the open source software VirtualDubMod <http://virtualdubmod.sourceforge.net> :

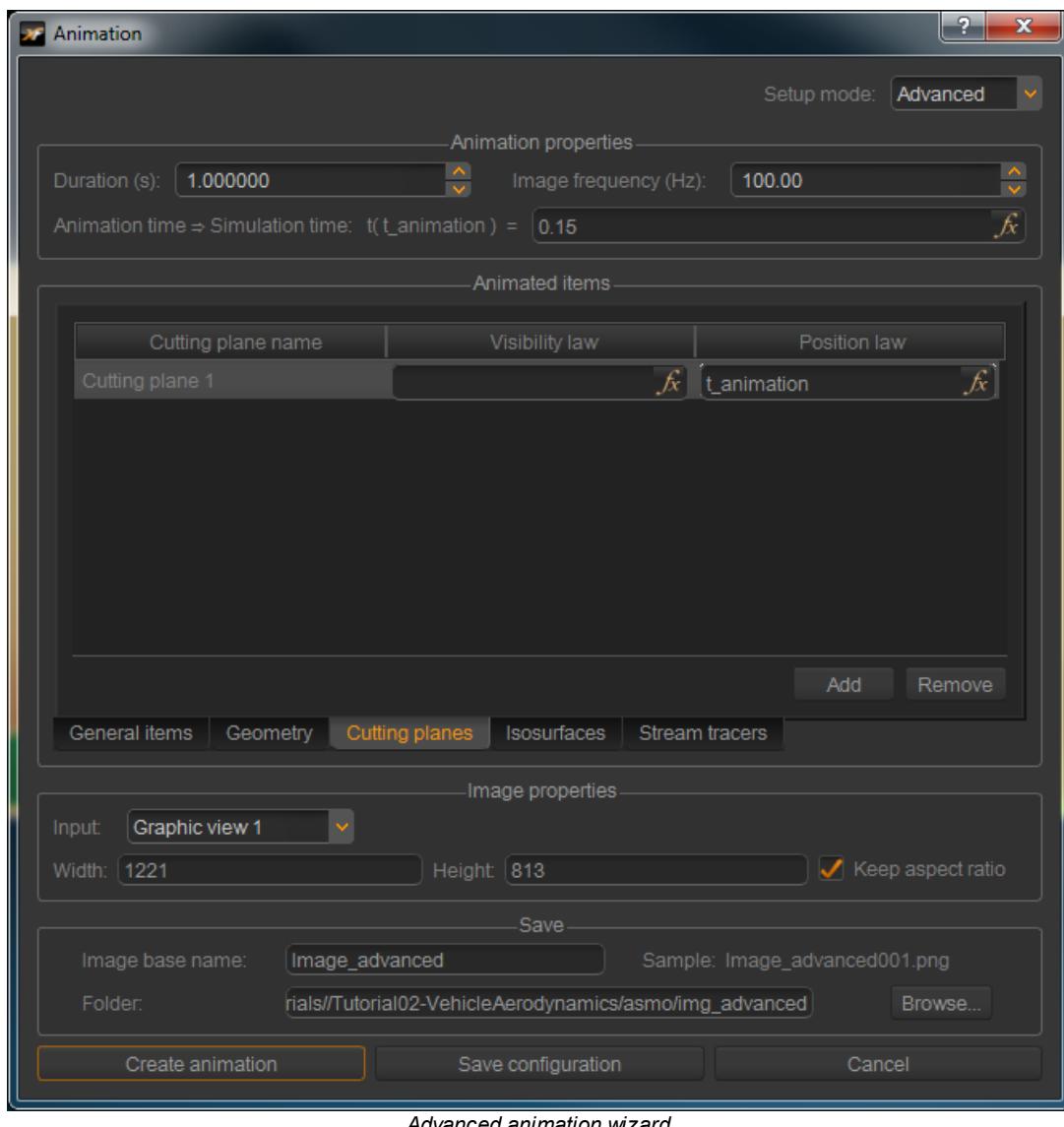
- drag the first image of the sequence into VirtualDubMod, it should find and append the rest of the image sequence
- go to Video > Frame Rate and enter the desired frame rate
- choose a processing mode and compression in the Video menu
- go to File > Save As

### 4.3.2 Advanced animation I

- (a) **Main menu > Post-Processing > Animation**

- (b) Select **Setup mode**: **Advanced**
- (c) **Duration**: **1 s**, this represents the duration (real time) of the animation
- (d) **Image frequency**: **100 Hz**, frames per second. Since the duration is 1 second, the animation will consist of a sequence of 100 images.
- (e) **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.
- (f) Go to **Animated items > Cutting planes** and press **Add**. Select **Cutting plane 1**.
- (g) Leave blank the **Visibility law** of the Cutting plane 1, this means that the cutting plane will be visible during the whole animation.
- (h) **Cutting plane 1 > Position law**: **t\_animation**. This variable ranges from 0 to the animation duration, in this case duration is 1 s. Hence, the Cutting plane will move from  $Z = 0$  to  $Z = 1$  sweeping the whole spatial domain.
- (i) **Image properties**, leave the default values.
- (j) **Save > Image base name**: **Image\_advanced**
- (k) Change the animation folder name to `C:/XFlow/Tutorial02/asmo/img_advanced` for example
- .
- (l) Press button **Create animation**

## Step 4: Post-processing



*Advanced animation wizard*

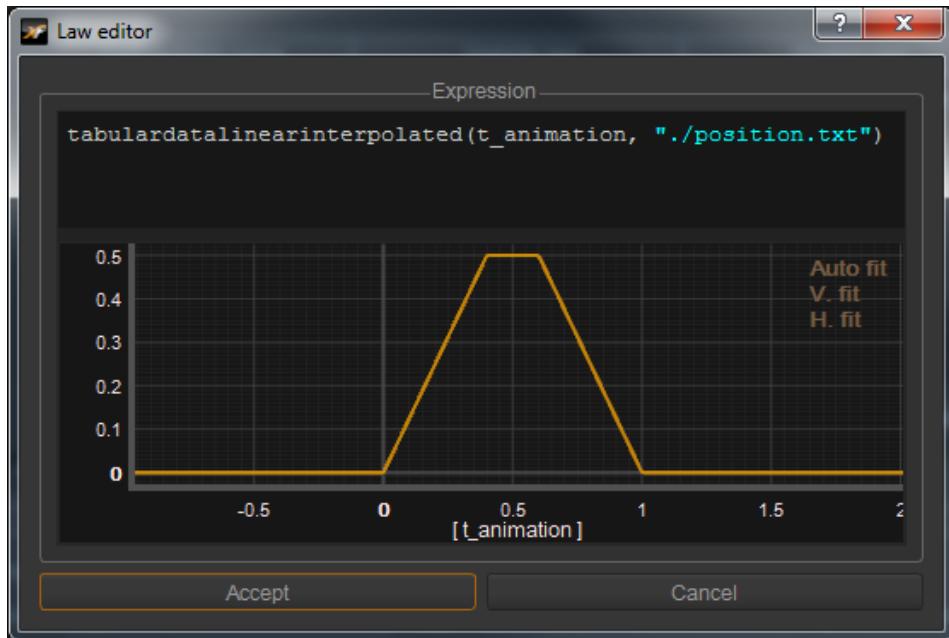
- (m) Check the new images stored in folder `C:/XFlow/Tutorial02/asmo/img_advanced/`. The images sequence shows the velocity contours at the Cutting plane 1 at a given time (0.15 s). The cutting plane moves from  $Z = 0$  to  $Z = 1$ , showing the velocity contours in the whole domain.
- (n) Create a text file named "*position.txt*" in the same folder where you have saved the project file (.xfp) 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 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:

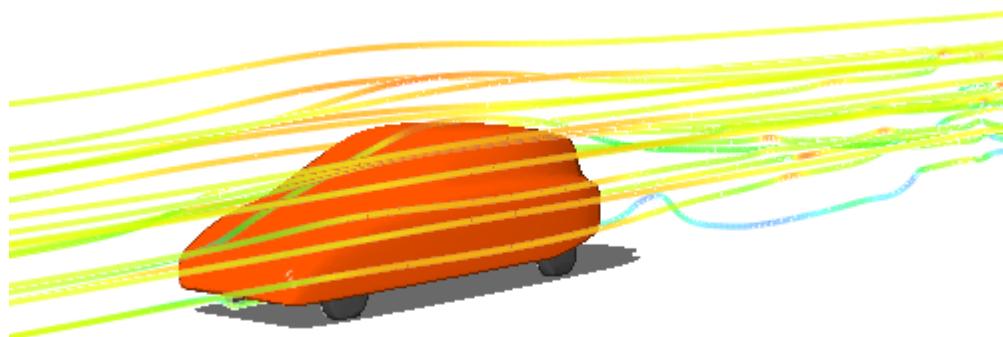
`tabulardatalinearinterpolated(t_animation, "./position.txt")` and click on to check if the data is read correctly by XFlow:



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

# Tutorial 03 - Advanced post processing

This third 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 Documentation section of XFlow

website ([http://www.xflowcfd.com/index.php/client\\_area/documentation/view/1](http://www.xflowcfd.com/index.php/client_area/documentation/view/1)).

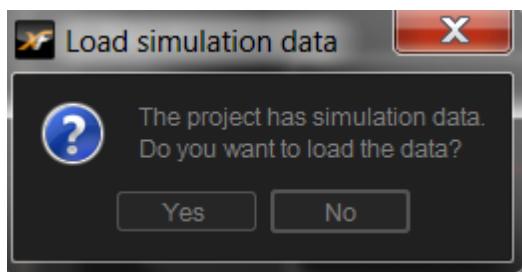
**Contents**

[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**.



- (b) Save the project with a new name in a new folder by hitting e.g.  
Tutorial03\_AdvPostProcessing.xfp

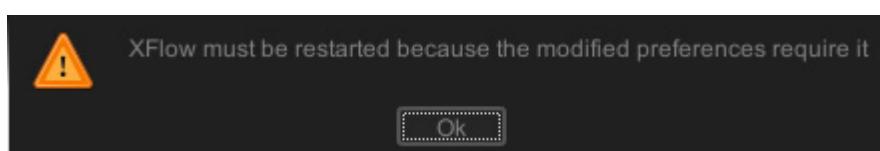
### 1.2 Change colour of the GUI

XFlow GUI colours can be customised in two styles:

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

Change the GUI style in: **Main menu > Options > Preferences, > Application > Application style: Classic**

The user will need to restart XFlow for the changes to be effective. 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 computation is done, you can superpose another geometry for visualisation purposes.

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) Remove the original geometry "asco" by selecting the **Shape: asmo** in the **Project Tree > Geometry > Entities**, [right clicking](#) in the *Graphic View* and selecting **Remove selected** in the *Graphic View* menu or just pressing key **Del**. To permanently remove this geometry, press **Ok** in the pop-up window.
- (d) Remove the original geometry "Line" by selecting the **Shape: Line** in the **Project Tree > Geometry > 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 > Geometry** by [right clicking](#) on the background a drop-down menu will appear, where the user should select **> Colapse all**
- (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 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 reflection index in metal or the transparency index in glass

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

One can also change the real-time visualisation properties in: [Main menu > Options > Preferences](#),  [> Graphic View > Real-time](#)

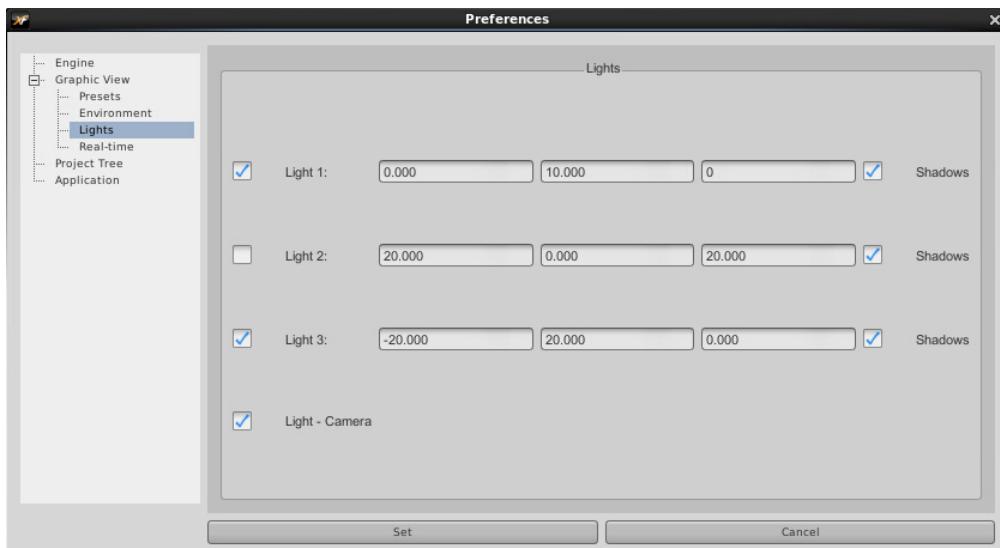
Lowering those parameters can speed up the visualisation of the *Graphic View*, but high parameters

improve the quality of the rendering. If the configuration of your Graphics Card is not enough good, we recommend to decrease these parameters, this way, lower graphical resources will be used.

## 1.5 Lights and ground wall colour

Shading is produced by the light sources when render is enabled. To define lights: **Main menu > Options > Preferences,  > Graphic View > Lights**

Please note that **Light 3** and **Light-Camera** are active by default. Activate **Light 1** at the location (0, 10, 0). Shadows for this light are active by default.

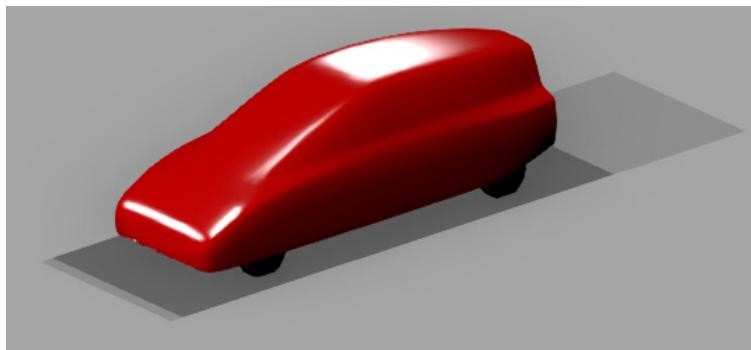


*Preferences window: Lights section*

Press the **Set** button. Notice that shadows are from two different lights now.

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.

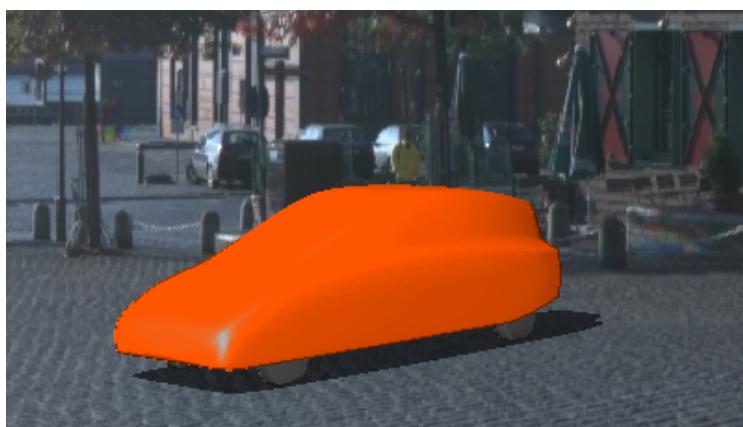


*Ground wall shown in gray*

### 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 fade between two colours or even have a texture as environment map.

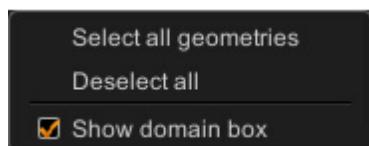
- (a) **Main menu > Options > Preferences,** **> Graphic View > Environment: Background**  
select **Two colours** and choose them by clicking on the coloured rectangles
- (b) Now, **Main menu > Options > Preferences,** **> Graphic View > Environment: Background** select **Texture**, and browse for a texture image (e.g citysquare.png)



### 1.7 Hide the domain box and the ground

To hide the domain box:

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



This hides the wireframe of the domain

To hide the ground wall: **Project Tree > Environment > Environment > Wind tunnel > Ground wall: Off.**

## 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.

(a) In **Post Processing > General > Show > Volumetric field** select the vorticity as visualisation field

(b) 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.

(c) Switch off the interpolation **Post-Processing > General > Interpolation mode: Off**. Otherwise the render of the volumetric field will be too slow.

(d) Show the volumetric field by ticking the **Volumetric field** checkbox  By default, the transference law is  $a^*a$ , which assigns a higher opacity to the higher vorticity. Try different transference laws, e.g.  $a$  and  $a^*a^*a$ .



*Volumetric field visualisation (vorticity)*

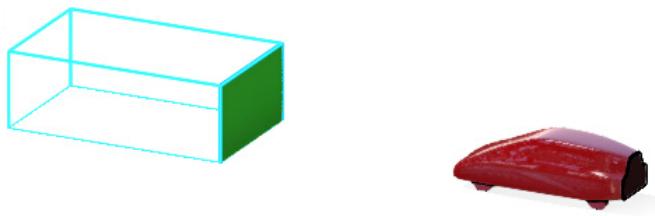
## 1.9 Visualise stream tracers

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, you need to define a geometry shape that will be the source of stream

tracers. Instead of a line like in [tutorial 02](#), now we will create a surface.

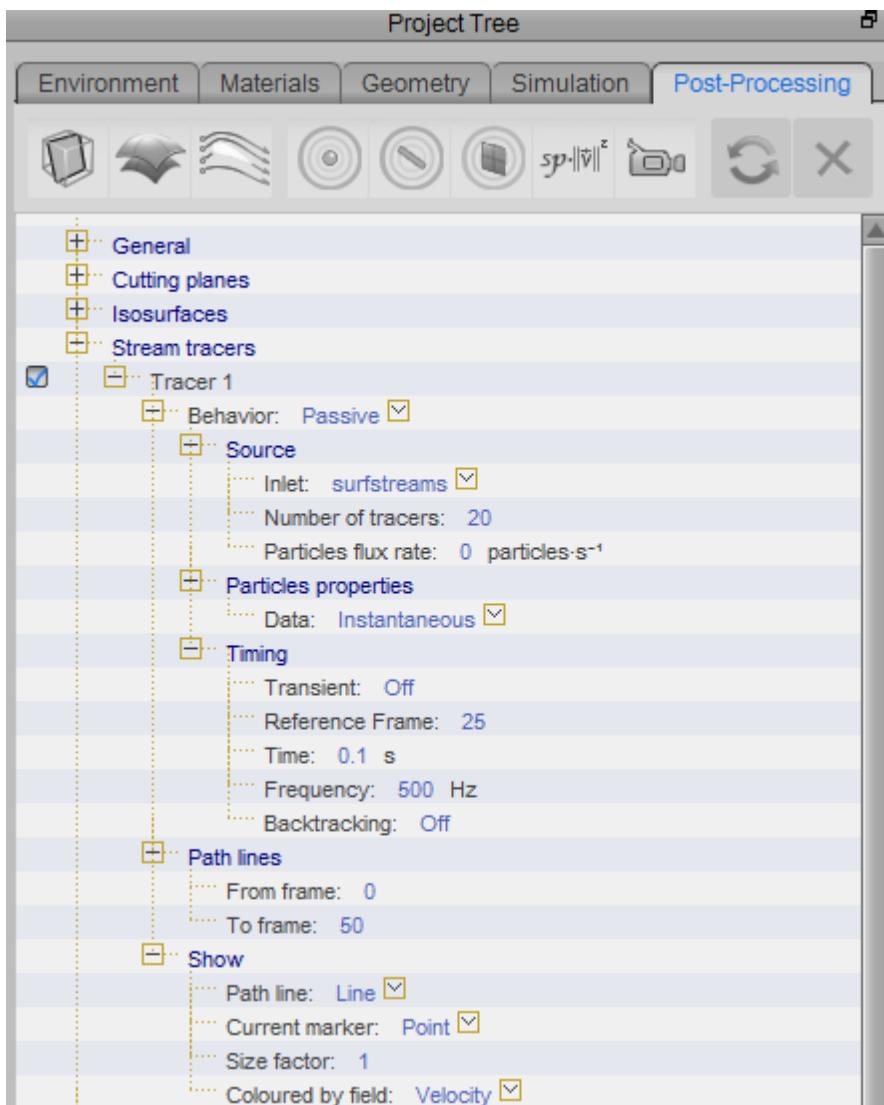
- (a) first create a box (**Main menu > Geometry > Create object > Create box** or just clicking on  in *Object Creation toolbar*) with lower corner (-3,-0.75,-0.3) and upper corner (-2,-0.4,0.3). Then eliminate all the box surfaces except the one at  $x = -2$  by selecting the face filter  in the toolbar, 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 > Geometry > Entities**, 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 **Tracer 1**.
- (c) Define **Tracer 1** 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



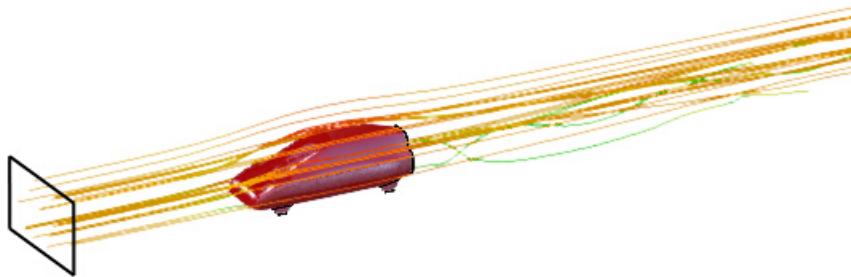
(d) Right click on **Tracer 1** and select **Recompute** or hit next to **Tracer 1**

(e) To visualise the path lines, right click again on **Tracer 1** and select **Generate path lines** or hit next to **Path lines**

(f) Choose Post-Processing > Stream tracers > Tracer 1 > Show > Path line: **Line** and **Size factor: 1**.

Set the velocity legend range to [20, 60] and recompute Tracer 1 (plus regenerate the path lines) to take effect.

Change **Path lines** style to Ribbon, 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) Untick the **Tracer 1** checkbox  to stop the stream tracer visualisation. The timeline will now be free.

### 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**:

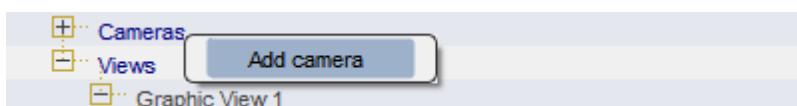
- **From** describes the position of the camera or eye point.
- **To** describes the position of the target, where the camera is looking at. It is the center of the view.
- **Up** describes the direction of the vertical axis.

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.

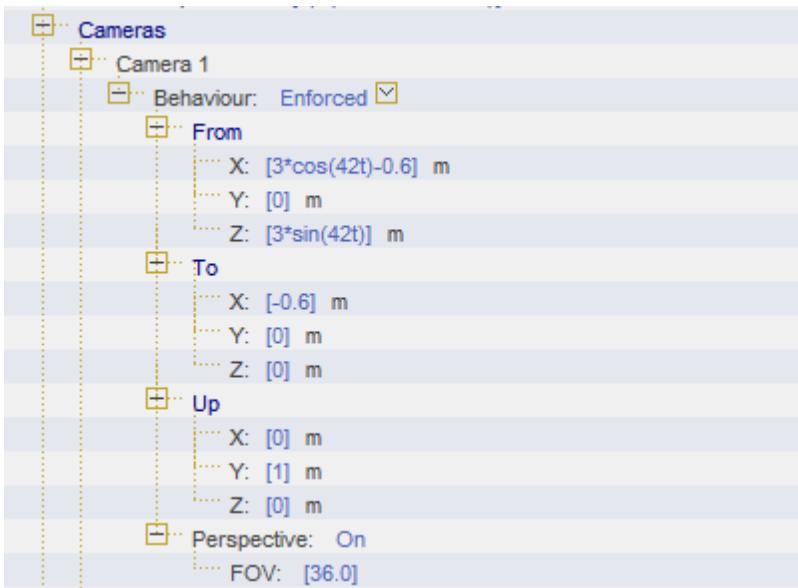


For example, define a camera from the top by defining **From = (0, 10, 0)**, **To = (0, 0, 0)**, **Up = (1, 0, 0)**, behaviour **Fixed** and perspective **On**. To activate this camera, go to **Post-Processing > Views > Graphic View 1 > Camera settings > Link to camera** and select **Camera 1**.

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:

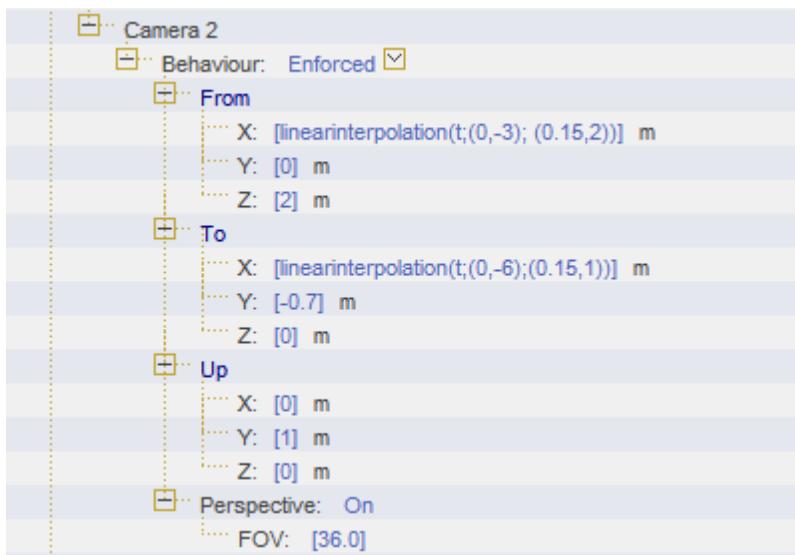


- (b) Display the graph of the From X-coordinate in the *Function Viewer* (you may need to create one) by right clicking the expression and selecting **Edit in Function Viewer 1**. Check that it is periodic with a period of 0.15 s.
- (c) Link the Graphic View to Camera 1: **Post-processing > Views> Graphic 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*.
- (b) Create a second camera with the following parameters:

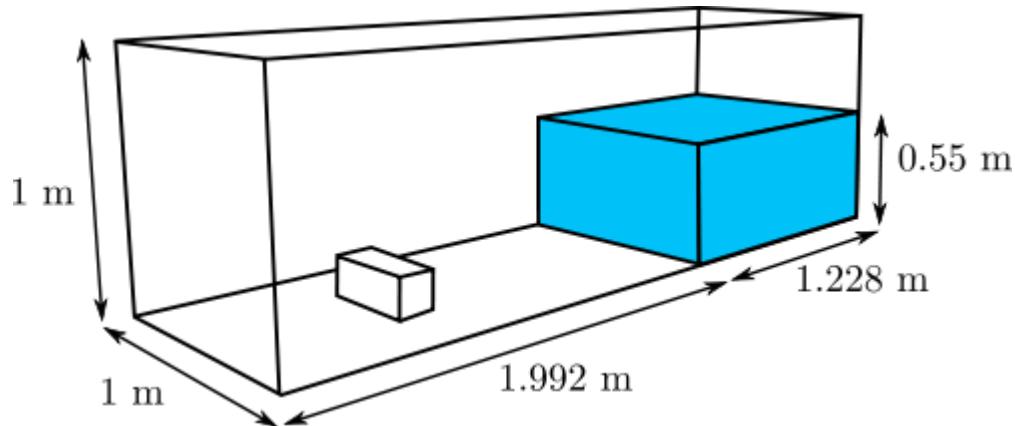


- (c) Link the Graphic View 2 to Camera 2. Select Show volumetric field as indicated in step 8 and play forward the results.

# 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 Documentation section of XFlow website ([http://www.xflowcfd.com/index.php/client\\_area/documentation/view/1](http://www.xflowcfd.com/index.php/client_area/documentation/view/1)).

## Contents

[Step 1: Problem setup](#)

[Step 2: Post-process](#)

# Step 1: Problem setup

## 1.1 Engine settings

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

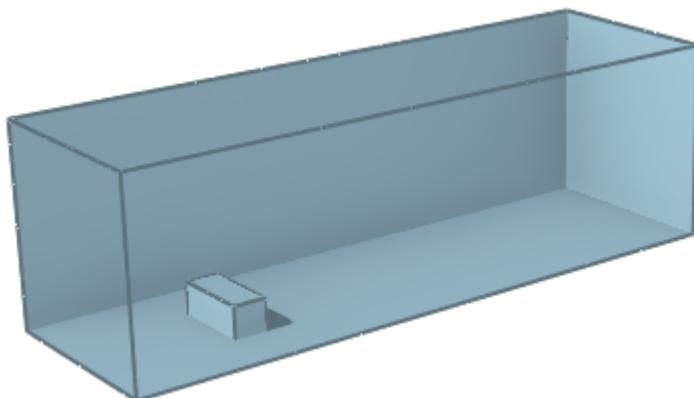
- (a) **Kernel:** **3d**
- (b) **Flow model:** **Free surface**
- (c) **Analysis type:** **Internal**. Now the wind tunnel will disappear and you will have to provide the geometry of the tank that will contain the fluid
- (d) **Turbulence settings > Turbulent model:** **Automatic**

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

Show the grid (**Main menu > Options > Preferences > Graphic View: Show grid**) to observe that the coordinate origin is at the location where the water column begins.



*Correct tank orientation (Otherwise, reverse normals)*

### 1.3 Environment settings

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

- (a) Global attributes > Ext. acceleration laws: (0, -9.81, 0) m.s<sup>-2</sup>, to take into account the gravity.
- (b) Leave the Initial conditions as User defined, and keep the Initial velocity field as (0,0,0)
- (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.

### 1.4 Material settings

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

### 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) Resolution > Resolved scale: 0.03 m, with the Refinement algorithm: Disabled
- (d) Resolution > Options > Seed point: Automatic
- (e) Store data > Folder: DamBreak
- (f) Store data > Frames frequency: 50 frames per second (Hz)
- (g) Leave off Save averaged fields
- (h) Enable Compute markers
- (i) 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 DamBreak.xfp

## Step 1: Problem setup

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

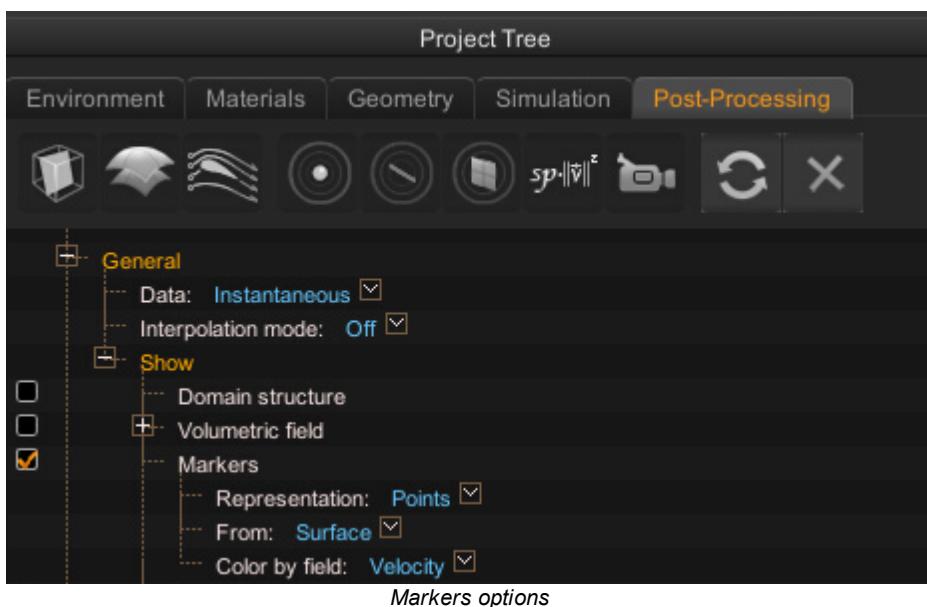
## Step 2: Post-processing

### 2.1 Load data

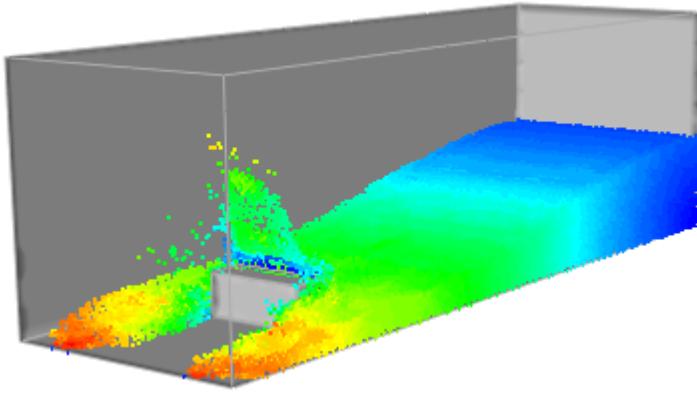
Load data: **Simulation data > Load simulation data** or 

### 2.2 Show particles

Tick the checkbox  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, select **Arrows** and **Sphere** in **Representation**.
- (c) Play the simulation by pressing 



## 2.3 Create a clipping plane

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

- 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.
- Check the box  in front of 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.
- Change the Origin of the plane, e.g. (0, 0, -0.2) and the Normal vector to (0, 0, 1). Now the other side of the clipping plane can be visualised.
- Disable the clipping plane by unchecking Plane 1

## 2.4 Create data sensors

Post-Processing > Sensors

- 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, 0.101, 0) and other on the top of the obstacle at position (-1.2685, 0.161, 0)
- Select: Field: Static pressure for both sensors
- Refresh the sensors with
- 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
- 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

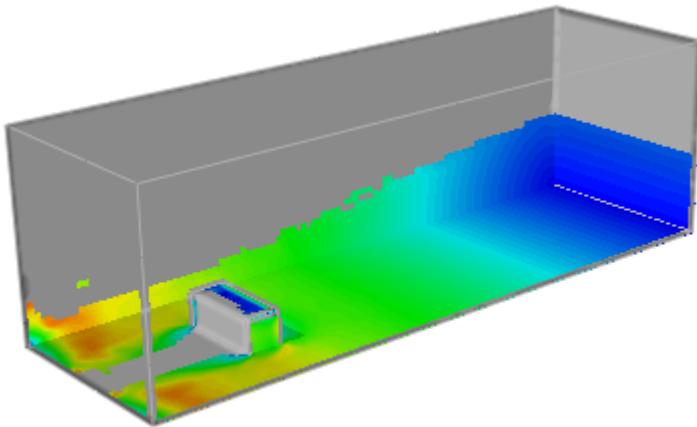
## Step 2: Post-processing

instead, **Simulation > Store data > Probes**. However, the Probes need to be defined before running your computation.

### 2.5 Show surface info

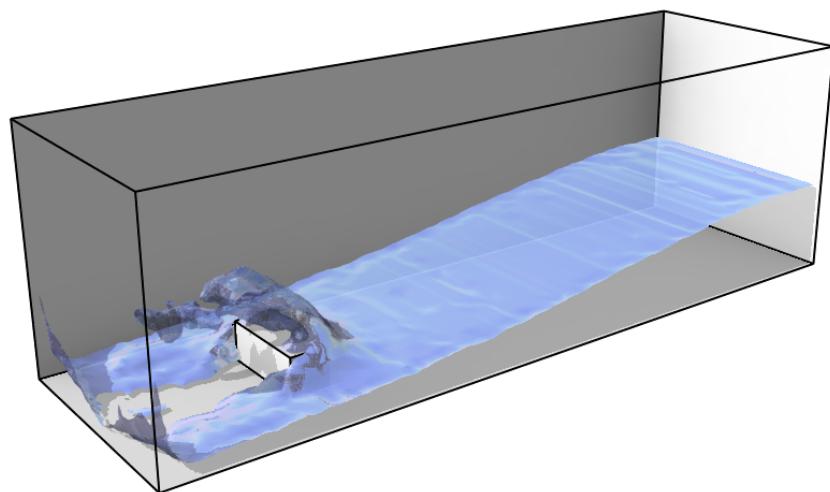
**Post-Processing > General > Show > Surface info**

- (a) Select **3d field** and a **Field: Velocity**
- (b) Check the box



### 2.6 Create an isosurface

- (a) Create an isosurface **Right clicking** on **Post-Processing > Isosurfaces** and **Add isosurface** with **Field: Volume of liquid phase** and **Value: 0.5**. This way, the isosurface will represent the free-surface of water.
- (b) Change the visualisation material of the Isosurface to get visula effect similar to water: Select the Isosurface in the **Graphic View 1 > Right click > Set visualisation material**. Select the **Glass** material type and a blue colour:



*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.

It is 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 a free surface external problem
- Set progressive waves with the help of the wizard
- Set center of gravity for moving objects
- Set the gravity as an external acceleration laws 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 Documentation section of XFlow website ([http://www.xflowcfd.com/index.php/client\\_area/documentation/view/1](http://www.xflowcfd.com/index.php/client_area/documentation/view/1)).

## Contents

- [Step 1: Problem setup](#)
- [Step 2: Post-process](#)
- [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 default options and press **OK**

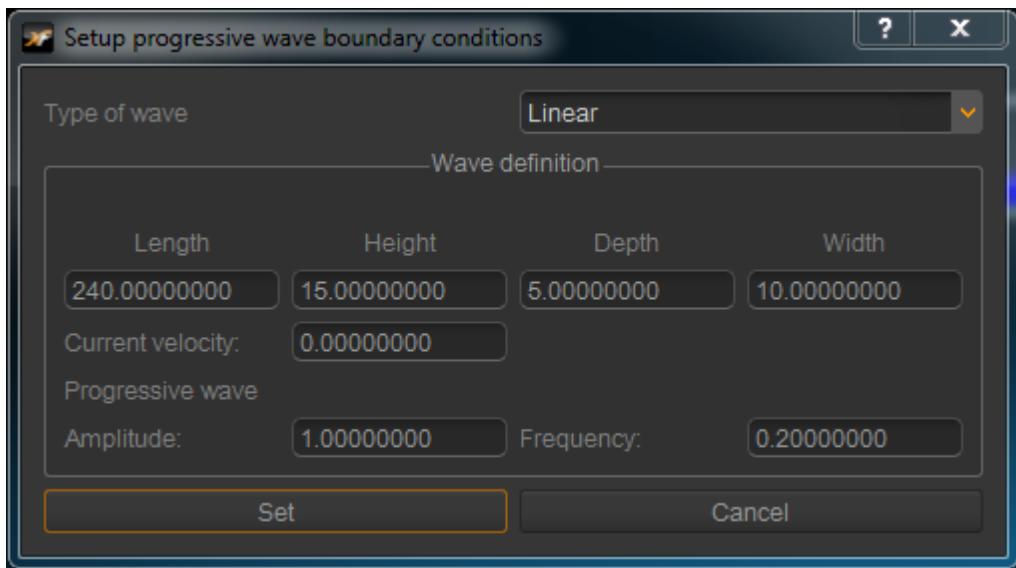
(b) Check the correct orientation of the wedge surface, i.e. make sure the normals are pointing to the outside. Otherwise, reverse orientation (Select the geometry, **right click** on the *Graphic View* window to show the **Graphic View Menu** and choose **Reverse orientation**).

### 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<sup>-2</sup>**) 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<sup>-1</sup>, 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 ground wall to automatic type 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:**

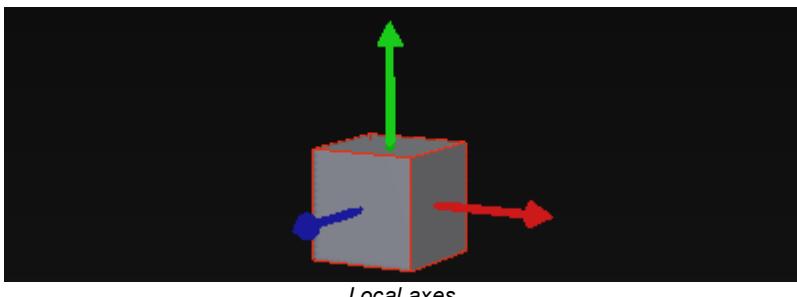
- (a) Set the **Time > Simulation time: 20 s** and the **Time > Time step mode** to **Fixed automatic** with **Courant: 1**
- (b) Set the **Resolution > Resolved scale: 0.2 m**, with the **Refinement algorithm: Disabled**
- (c) **Store data** at a **Frame frequency** of **5 Hz** and leave the numerical 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** or  ) 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**

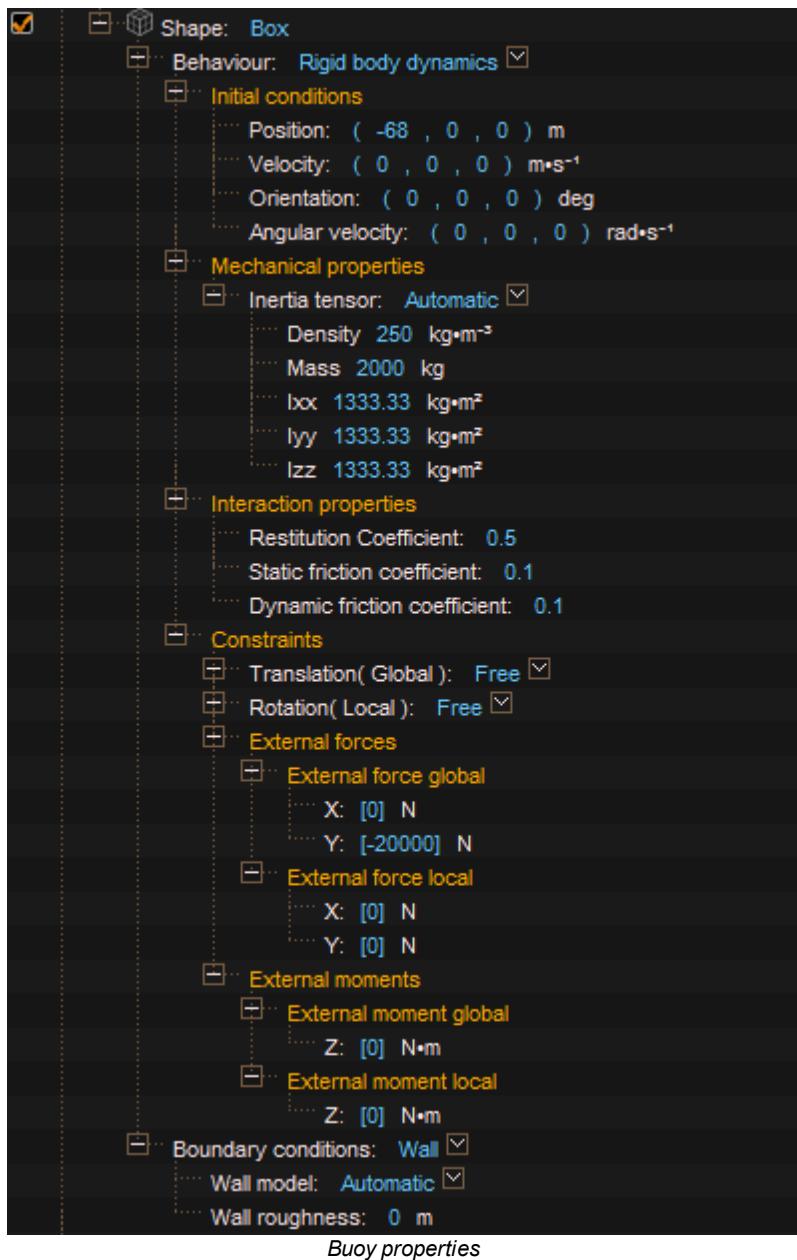
**⚠ Please note:** You could also create the cube directly in the right position, i.e. lower corner **(-69, -1, -1)** and upper corner **(-67, 1, 1)**; but observe that then the center of gravity would be in the global origin **(0, 0, 0)**, which would make the simulation to fail. In this case you should 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.



- (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<sup>-3</sup>** (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



### 1.8 Launch the calculation

- Save the project



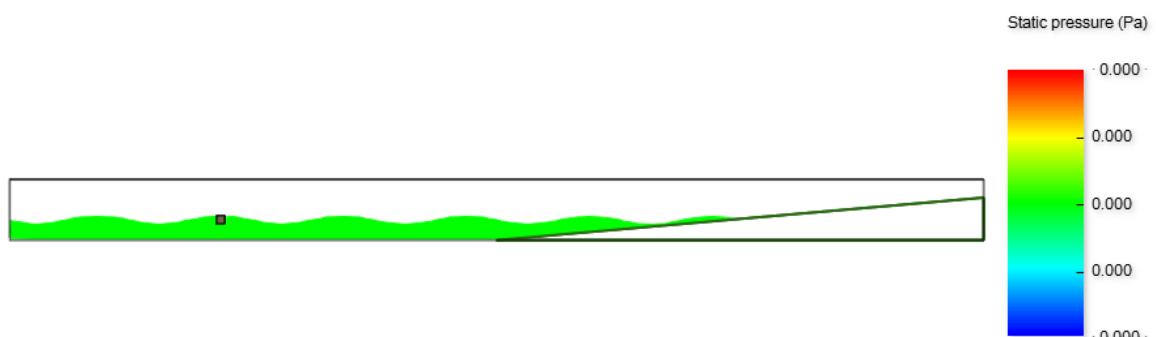
**Tip:** You can directly load the setup of this problem from the project file `BreakingWaves.xfp`

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

## Step 2: Post-processing

### 2.1 Check pressure initial 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

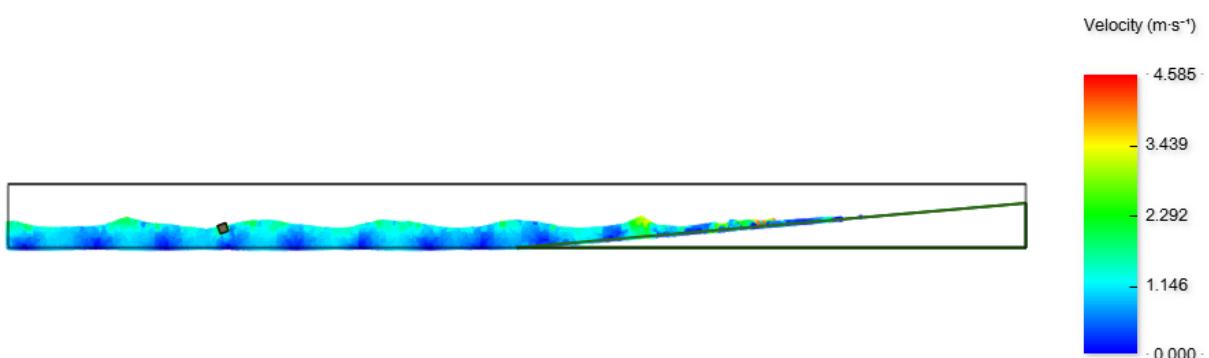


*Markers coloured by the initial static pressure field (time = 0).*

**⚠ 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 coloured 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

## Step 2: Post-processing

up breaking due to shoaling.

- (c) Visualise vorticity and static pressure fields (Post-Processing > Show > Markers > Coloured by field).
- (d) Select the beach and visualise it in wireframe mode (**Graphic View Menu > Visualisation Mode > Wireframe**). Observe that there is no fluid inside.

### 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 the gravity.

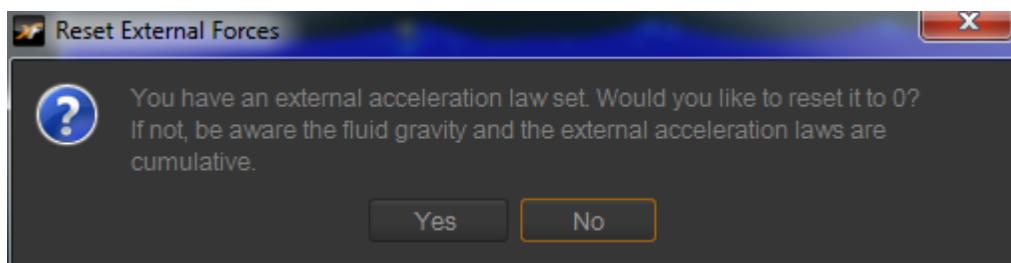
In this section, the feature **Gravitational potential** is used to compute the [Breaking waves](#) case.

### 3.1 Gravitational potential setup

Load the project generated in [Step 1](#),  *BreakingWaves.xfp*, and save it 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**. It

should appear a pop-up window like the one shown in the figure below:

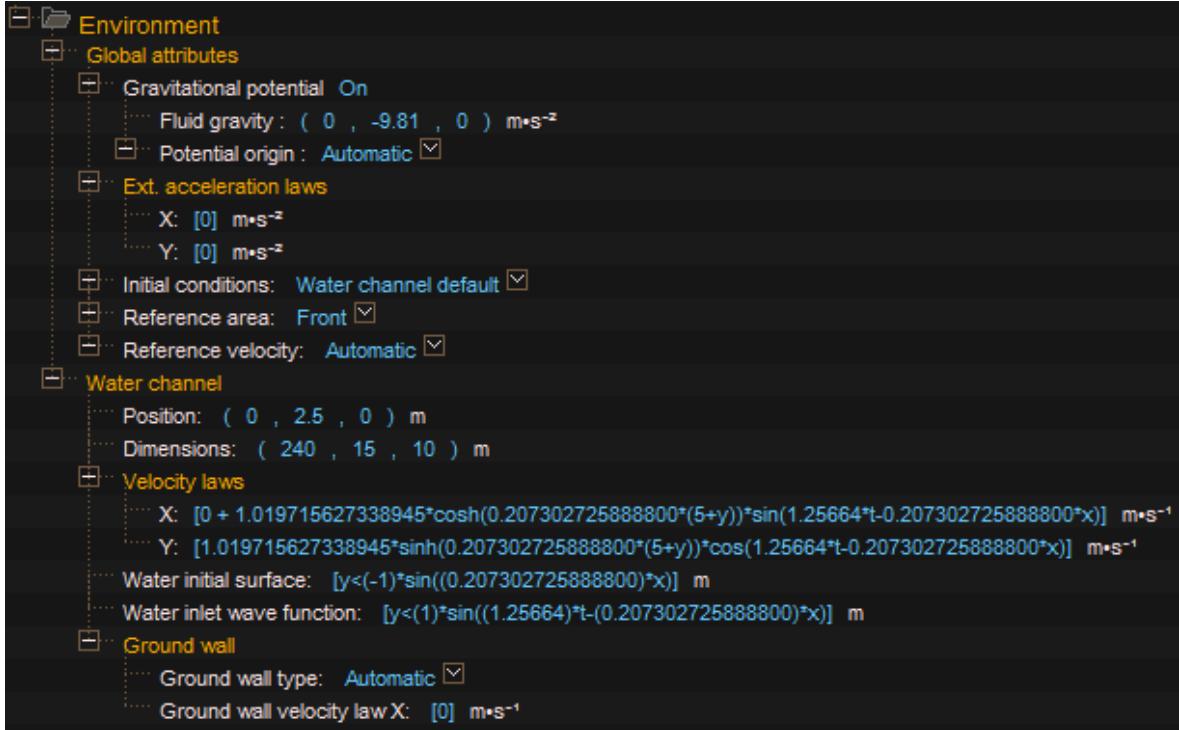


This message indicates that the setup contains a non-zero **External acceleration law** together with the **Gravitational potential**, which accounts for the gravity. In this case, the **Ext. acceleration**

**laws** represent the gravity ( $(0, -10) \text{ m}\cdot\text{s}^{-2}$ ) introduced by hand in [Step 1](#). Since the user only wants to consider the gravity effects once, press **Yes** to reset the **Ext. acceleration law** to zero.

(b) Leave the earth gravity as the **Fluid gravity:  $(0, -9.81, 0) \text{ m}\cdot\text{s}^{-2}$**

(c) Leave the **Gravitational potential > Potential origin: Automatic**, which by default sets the potential origin at the free surface.



### 3.2 Launch the calculation

(a) Rename the **Simulation > Store data > Folder** as **BreakingWaves\_GravitationalPotential**

(b) Save changes,



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

`BreakingWaves_GravitationalPotential.xfp`

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

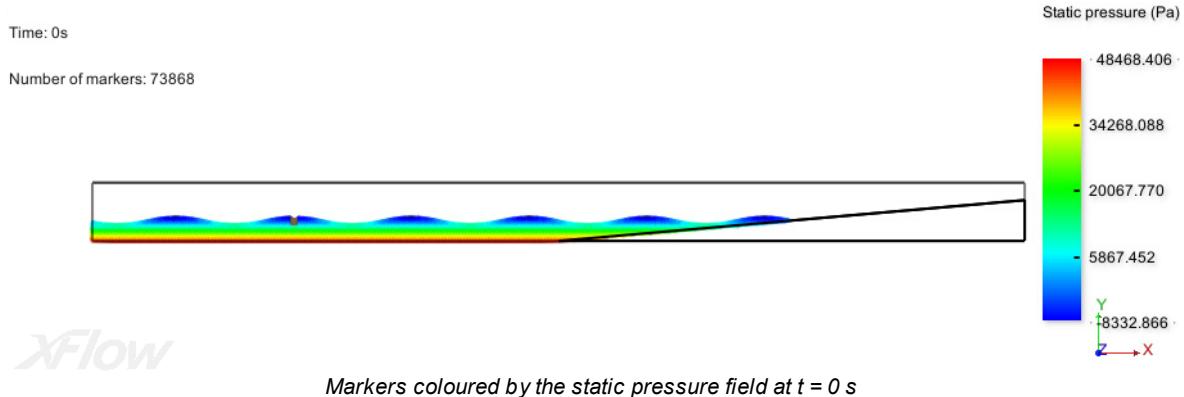
### 3.3 Post-processing

(a) Visualise the initial static pressure field (frame 0). Activate the markers visualisation 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

### Step 3: Gravitational potential

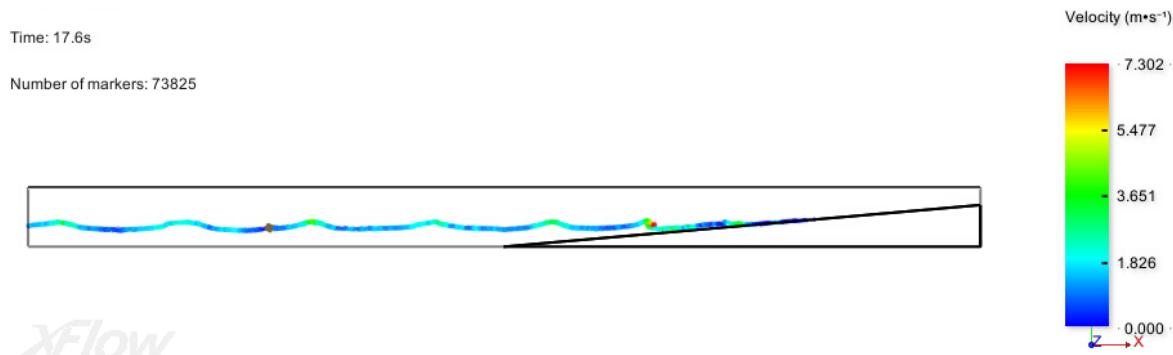
Markers > Color by field.



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

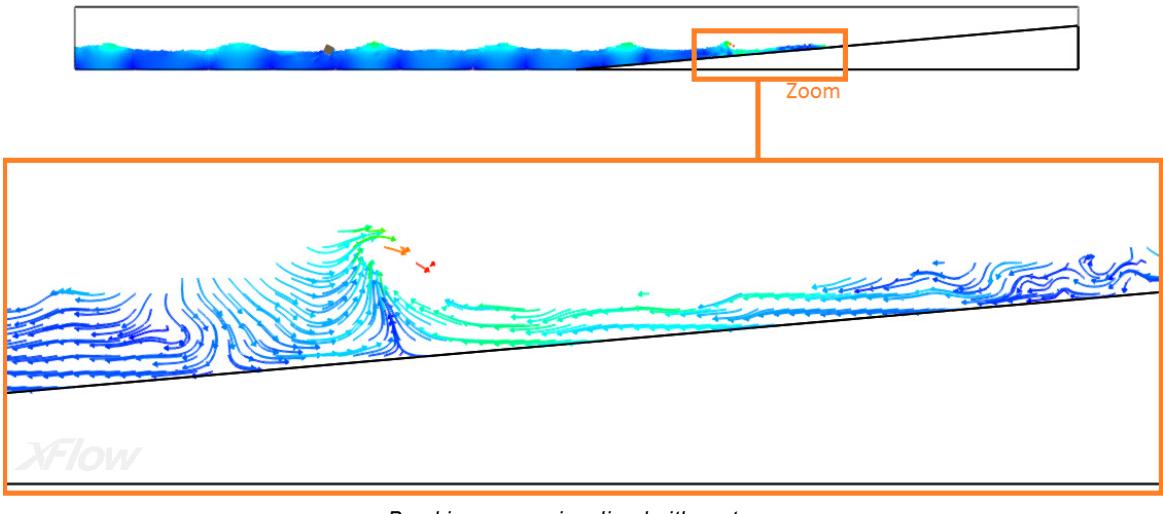
(b) Visualise the free surface by activating the markers visualisation as follows:

- 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: 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



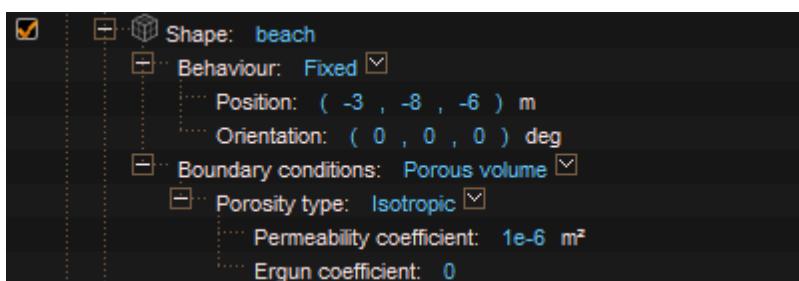
## Step 4: Porous volume

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

### 4.1 Porous region setup

Load the project generated in [Step 3](#), `BreakingWaves_GravitationalPotential.xfp`, and save it 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 zero **Ergun coefficient**



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

## 4.2 Launch the calculation

- (a) Rename the **Simulation > Store data > Folder** as **BreakingWaves\_GravitationalPotential** Save the project
- (b) Save the project
- (c) Set the number of CPUs in **Main menu > Options > Preferences > Engine**
- (d) Press **Run** button > **Start computation**

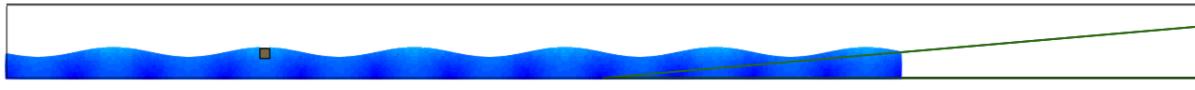


**Tip:** You can directly load the setup of this problem from the project file **BreakingWaves\_Porous.**

**xfp**

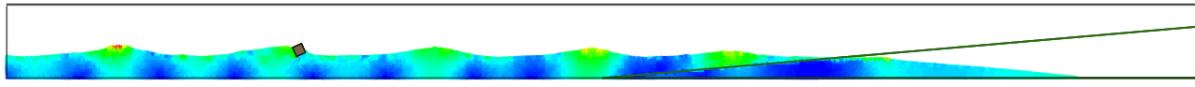
## 4.3 Post-processing

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



*Markers coloured by velocity at t = 0 s*

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

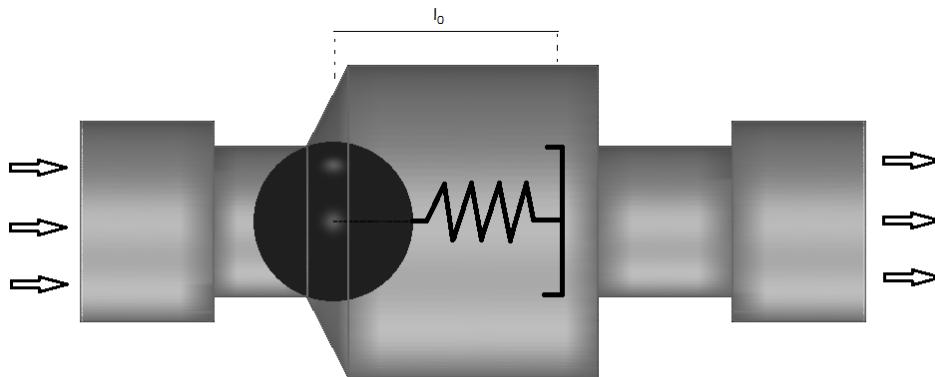


*Markers coloured by velocity at t = 15 s*

# 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 that is against the spring. If the fluid stops coming from 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 in X-direction with and without damping to simulate different regimes of the spring.



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

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

The parameter  $\xi$  is called the damping ratio and determines the behavior of the system, and  $k$  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 Documentation section of XFlow website ([http://www.xflowcfd.com/index.php/client\\_area/documentation/view/1](http://www.xflowcfd.com/index.php/client_area/documentation/view/1)).

## Contents

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

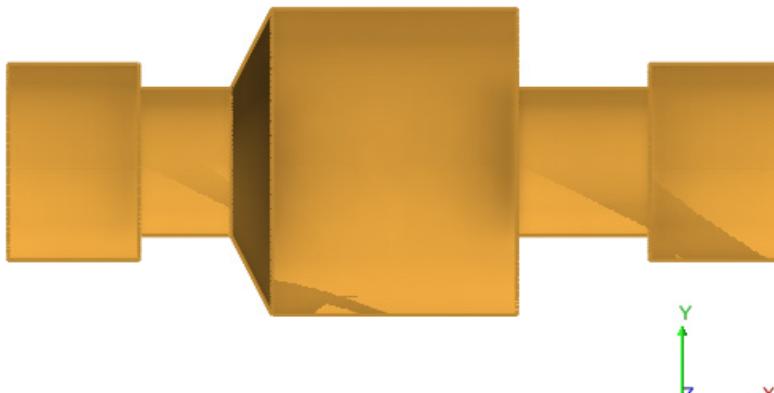
## Step 1: No damping case

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

### 1.1 Import/Create the geometry

Load and setup the valve geometry as indicated below:

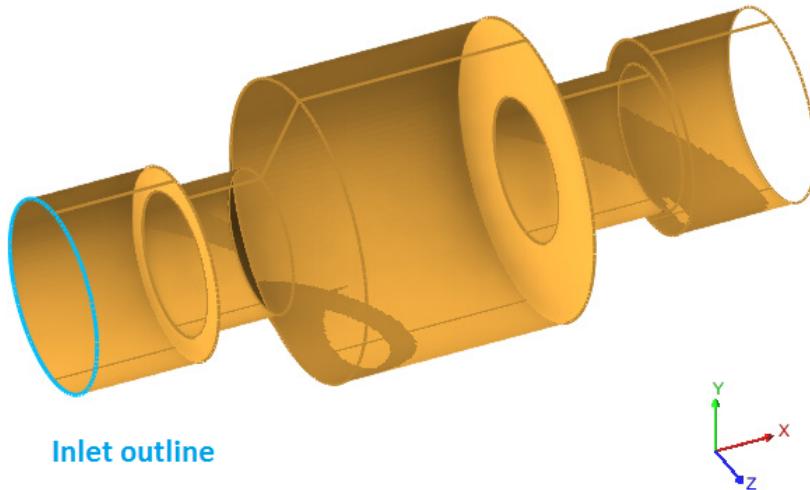
- (a) Import the geometry file named `Valve.stp` 
- (b) In order to orientate the flow from -X to +X, make a rotation of 180 degrees around Y-axis by changing to **(0, 180, 0)** the geometry orientation: **Geometry > Entities > Shape: Valve > Behavior > Orientation**. The valve is now correctly oriented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



Valve geometry

- (c) Check the orientation of the geometry. Remember that the 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 to the fluid region. To reverse the normals orientation, **right click** on the *Graphic View* window and in the **Graphic View Menu** choose **Reverse orientation**
- (d) You can notice that the valve inlet and outlet surfaces are not defined. Create the inlet boundary

surface: use the "Edge filter" button to select the circular inlet contours of the inlet cylinder as shown below. You have to maintain the **Ctrl** key to select several edges.



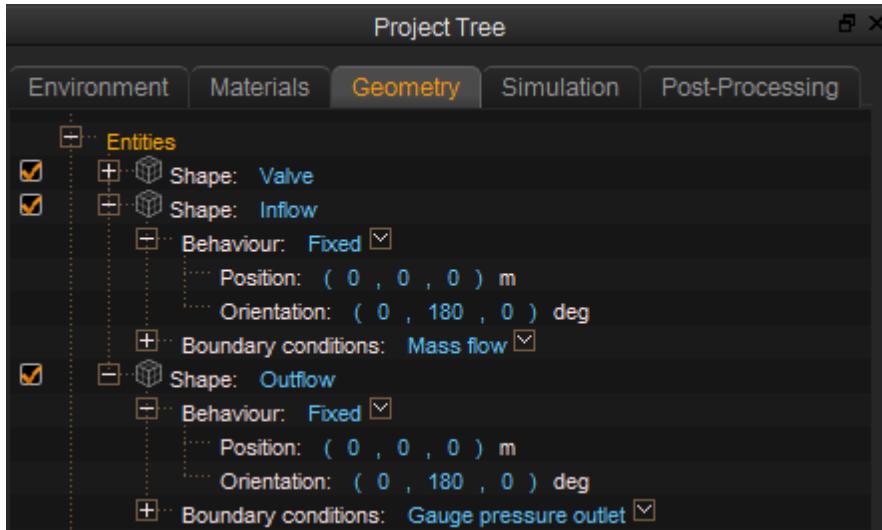
Inlet outline

- (e) Now click on the "Create surface" button from the *Toolbar Object Creation* and a new circular surface will appear in the geometry tree: **Geometry > Entities > Shape: Surface**.
- (f) Repeat steps (d) and (e) at the valve outlet to get the outlet closing surface.
- (g) It is recommended to rename the geometries created with more explicit names. Here, for instance,

## Step 1: No damping case

it is suggested to call the surfaces "**Inflow**" and "**Outflow**":

- **Geometry > Geometries > Entities > Shape: Inflow**
- **Geometry > Geometries > Entities > Shape: Outflow**



Create the ball:

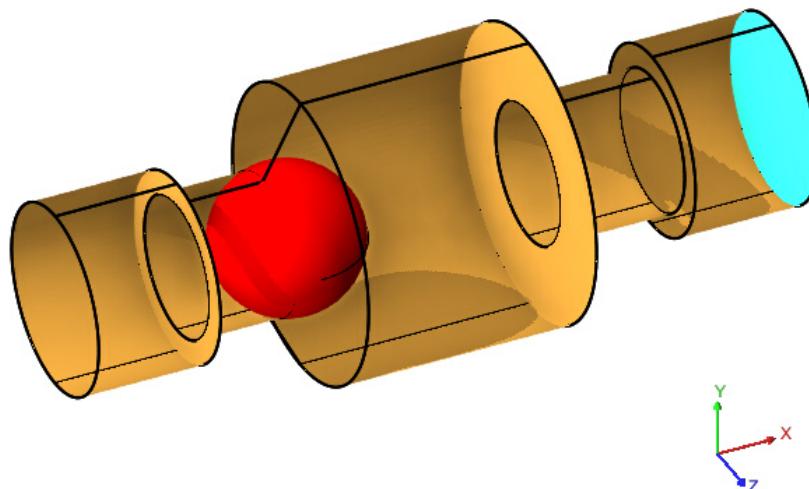
(a) Create a sphere of radius **0.008 m** at the location **(0, 0, 0)**. To do so, **Main Menu > Geometry >**

**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

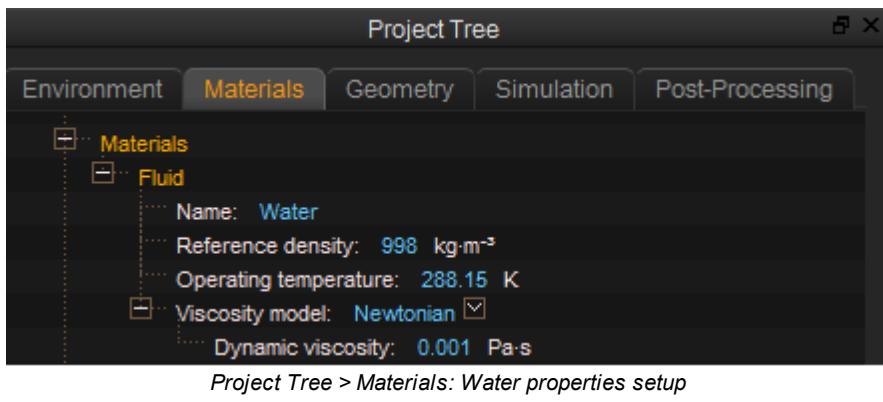
Leave the default configuration, this means: (i) no **External acceleration laws (0, 0, 0)**; and (ii) **user defined Initial conditions** with zero initial values. The system should be at rest at the first frame.

## 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:  $998 \text{ kg}\cdot\text{m}^{-3}$**
- (c) **Operating temperature: 288.15 K**
- (d) **Viscosity model: Newtonian**, with **Dynamic viscosity: 0.001 Pa·s**

## Step 1: No damping case

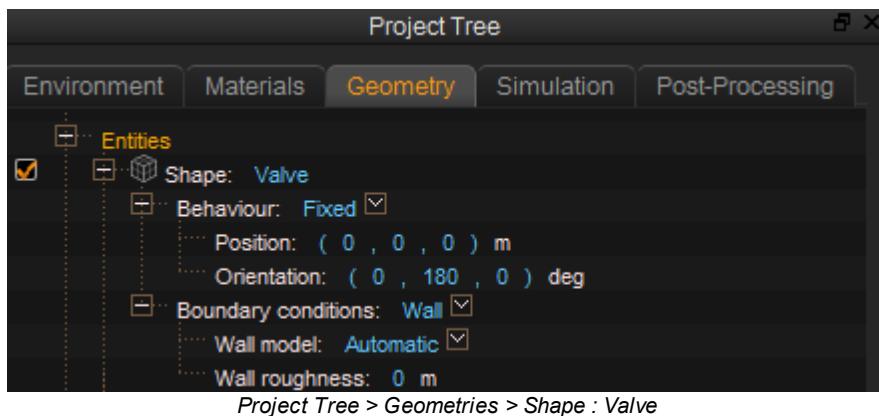


## 1.5 Geometry settings

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

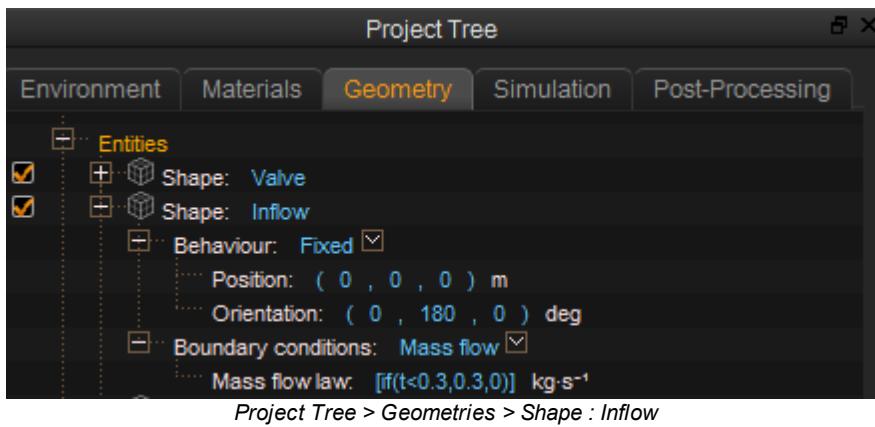
### (a) Valve geometry - Shape: Valve:

- 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**.



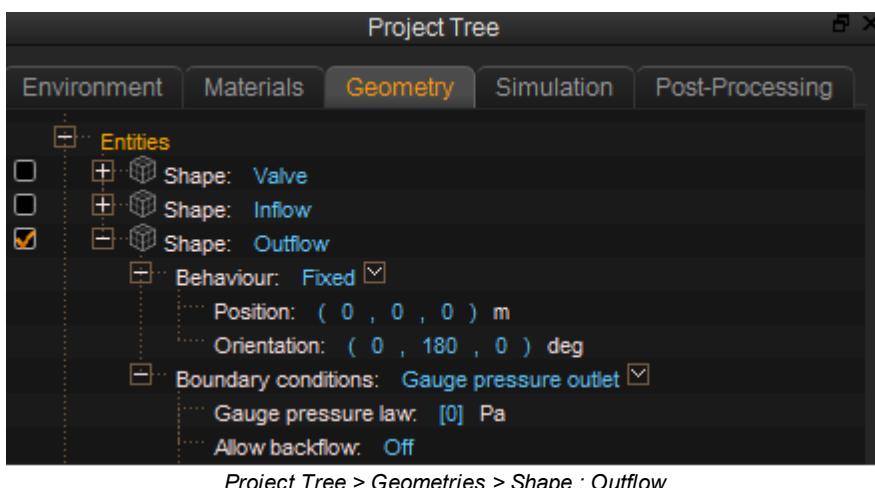
### (b) Inflow surface - Shape: Inflow:

- 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<sup>-1</sup>** 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<sup>-1</sup>**.



(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 **Allow backflow** option disabled.

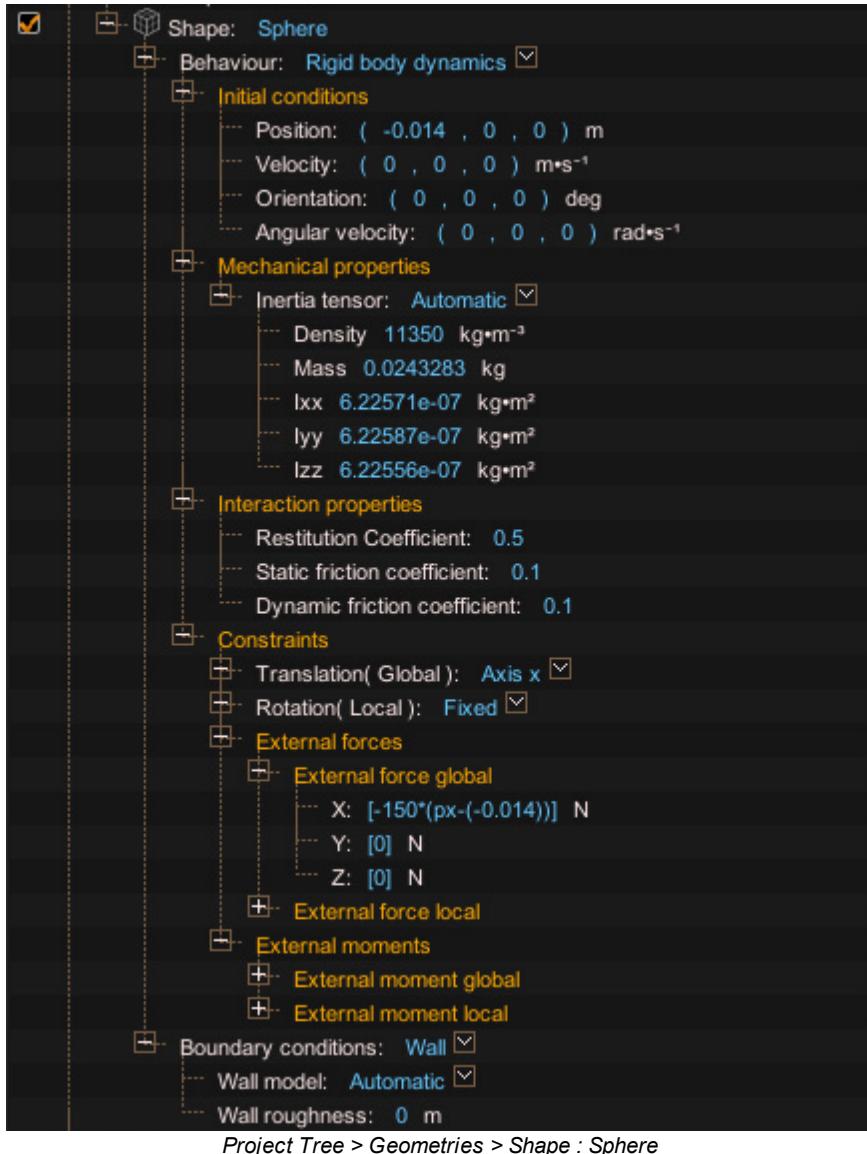


(d) Ball - **Shape: Sphere**: set up the Sphere to simulate the spring. In this section the spring is modeled with no damping:

- **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.

## Step 1: No damping case

- 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<sup>-1</sup>**
- **Boundary conditions: Wall** with **Automatic Wall model** and **0** **Wall roughness**.



## 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) **Resolution > Resolved scale: 0.001 m** with no refinement (**Refinement algorithm: Disabled**)

- (c) Store data > Frames frequency: **300 Hz**
- (d) Store data > Folder: NoDamping
- (e) Leave the Numerical data frequency to **Solver time step**
- (f) 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 numerical data will be automatically loaded. Otherwise, load the

numerical data 

## 1.9 Post-processing: Vectors

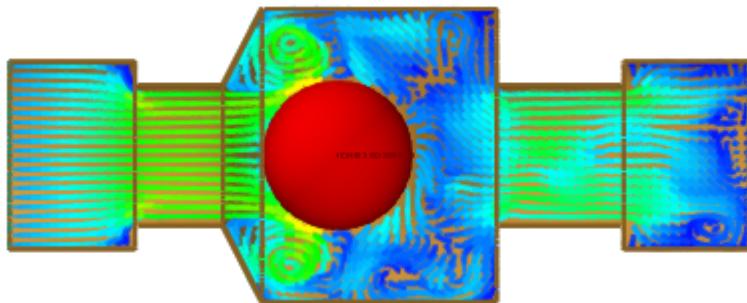


- (a) Create a cutting plane by clicking on the icon  on the Post-processing toolbar. A **Cutting plane 1** will appear in the *Post-processing* tab. Change its position to **0.5** to visualise the flow at the mid section of the Valve geometry.
- (b) Select **Vectors** as **Visualisation mode** of the **Cutting plane 1**. Adjust **Arrows density** to **1.000** and **Arrows length** to **0.8**.
- (c) Change the velocity range to **[0; 4.5]** by hitting 

## Step 1: No damping case

(d) Set the view to Right view by pressing 

(e) Play forward the simulation by pressing 



*Velocity vectors cutting plane*

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: Sphere position

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



*Function Viewer: Visualization of Shape > Sphere > Px*

Note that the evolution is periodic until 0.3 seconds 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 Run the under-damping case

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

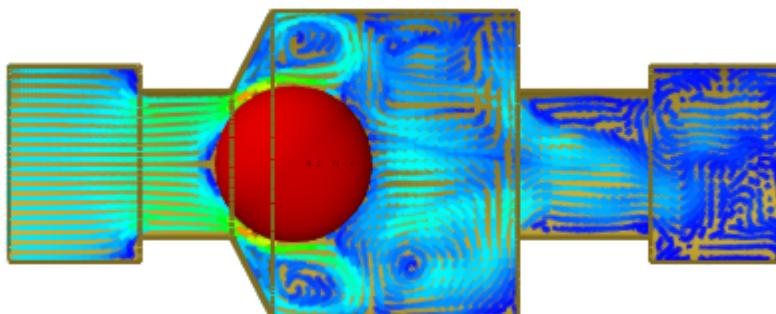
- (a) In **Geometry > Entities > Shape: Sphere > Behavior: Rigid body dynamics > Constraints > External forces** replace **External force global > X: [-150\*(px-(-0.014))]** by **[-150\*(px-(-0.014))-1.91\*vx] N**. The new term corresponds to the damping term and depends on the velocity. According to the [equations](#) and taking  $\xi = 0.5$ , the calculation leads to  $2\xi(km)^{1/2} = 1.91$ .
- (b) Change the name of the folder in **Simulation > Store data > Folder**, for instance "UnderDamping"
- (c) 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](#)

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

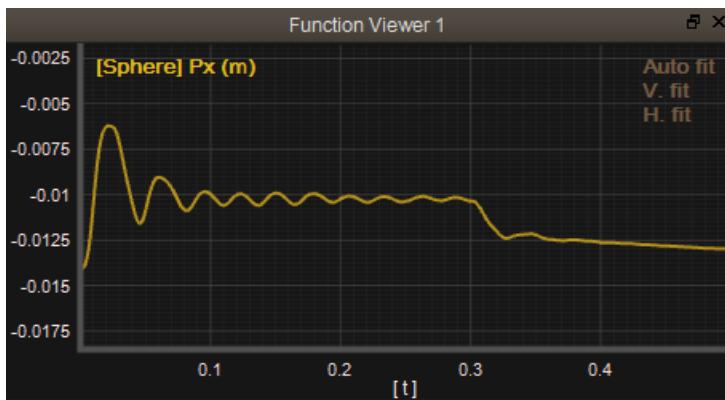
### 2.2 Post-process the results

- (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



- (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 which is the behavior expected from a spring that has a spring constant of 0.5.

## Step 2: Under-damping case



## 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 Run the under-damping case

Taking the "[Under 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))-1.91\*vx]** by **[-150\*(px-(-0.014))-3.82\*vx]** N. According to the [equations](#) and taking  $\xi = 1$ , the calculation leads to  $2\xi(km)^{1/2} = 3.82$ .
- Change the name of the folder in **Simulation > Store data > Folder**, for instance "CriticalDamping"
- Save the case, check the number of CPUs



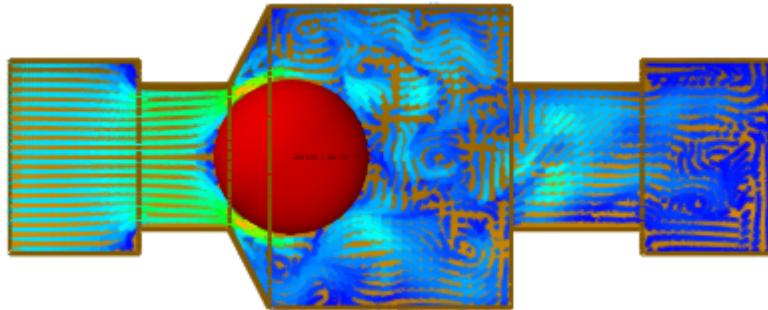
**Tip:** You can directly load the setup of this problem from the project file `BallCheckValve_CriticalDamping.xfp`

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

### 3.2 Post-process the results

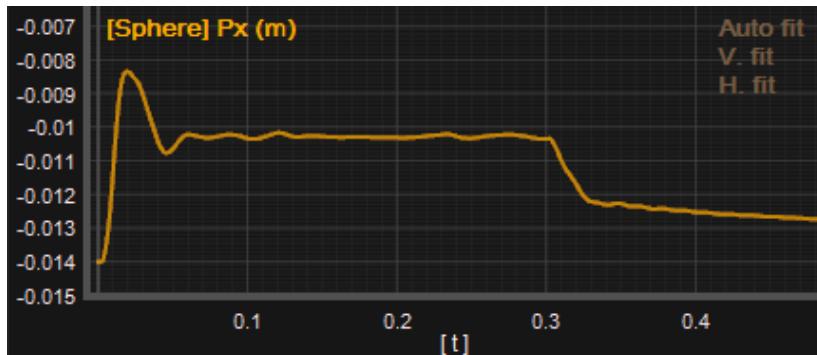
- Observe the velocity vectors in the **Cutting plane 1** as done in [section 1.9](#). Note that the sphere is reaching an equilibrium position directly without completing a full oscillation. The flow is smooth until the mass flow stops and the sphere is then going back to its original location slightly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

previous cases



(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 executed 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 an extreme case.

### 4.1 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) Change the name of the folder in **Simulation > Store data > Folder**, for instance "OverDamping"

#### Step 4: Over-damping case

- (c) 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`

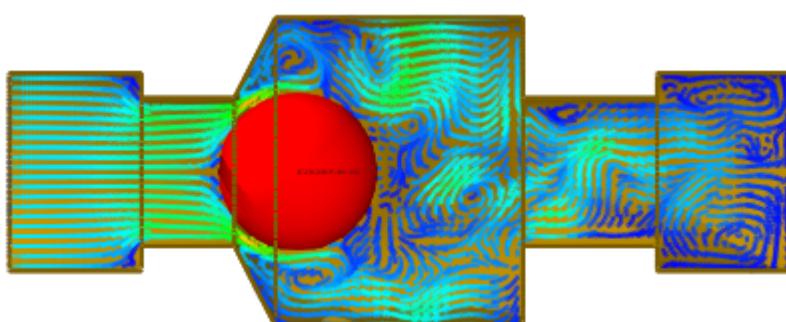
- (d) Press **Run** button > **Start computation**

#### 4.2 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! Time step too big. Please, try to run again the computation with a lower Courant number!!!" is output in the Message View between 0.015 and 0.021 s. This means the velocities are too high for the time step and the resolution of the lattice used locally in the domain. The Courant–Friedrichs–Lowy (CFL) condition is therefore not respected, which 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 instead, which will lower the time step.
- (d) Press **Run** button > **Start computation**

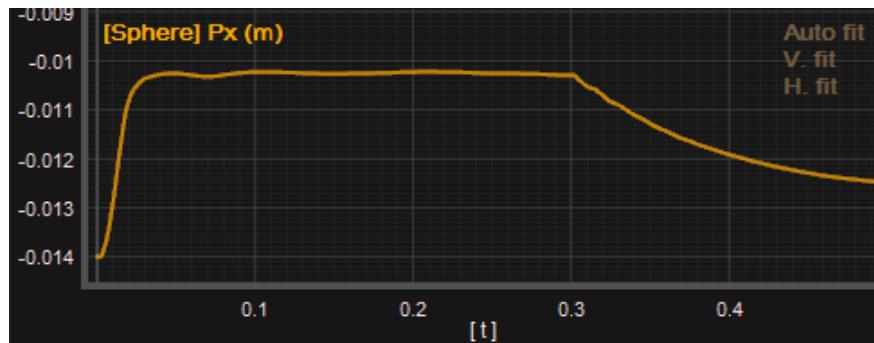
#### 4.3 Post-process the results

- (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 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 Xlocation. When the mass flow stops the sphere is then going back to its original location much slower than previous cases



- (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

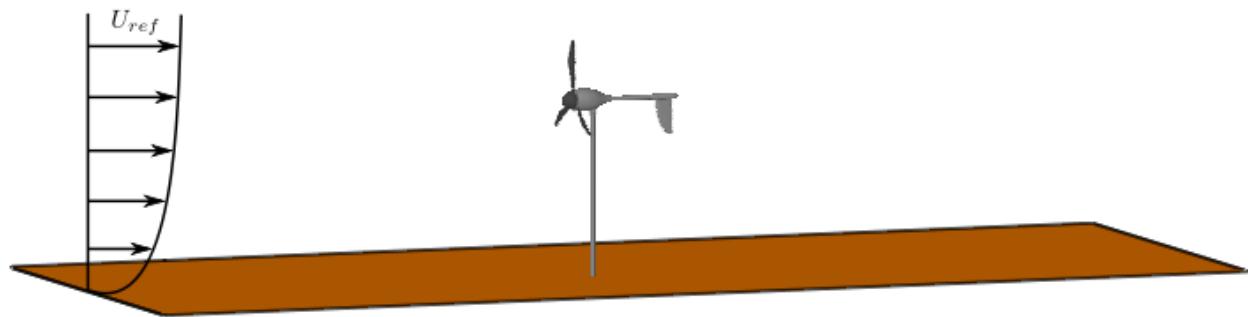
effective. Observe that no oscillation occurs: this is the over-damping. The drawback of such an over-damped system is that the sphere struggles to go back to its original location, and therefore does not seal the inlet passage properly.



Function Viewer: Visualization of Shape > Sphere > Px

# Tutorial 07 - Wind turbine

This tutorial illustrates the setup and solution of the flow around a wind turbine and its aeroacoustics 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 basic aeroacoustics analysis 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 Documentation section of XFlow website ([http://www.xflowfd.com/index.php/client\\_area/documentation/view/1](http://www.xflowfd.com/index.php/client_area/documentation/view/1)).



**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

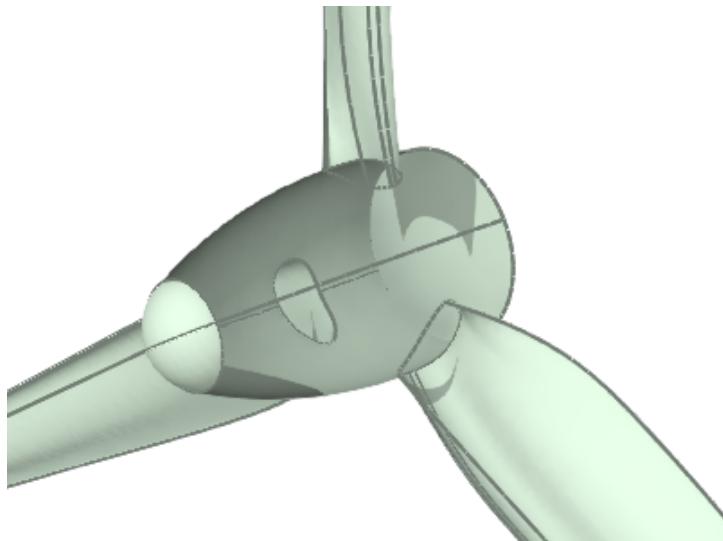
### 1.1 Import the CAD file blades-to-heal.igs

- (a) Set the OpenGL mesh deflection to 1 in; **Main menu > Options > Preferences > Graphic View > OpenGL Mesh Deflection.** This parameter (between 0 and 2) determines the size of the automatic tessellation for parametric geometries
- (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 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 7900 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 41000 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.

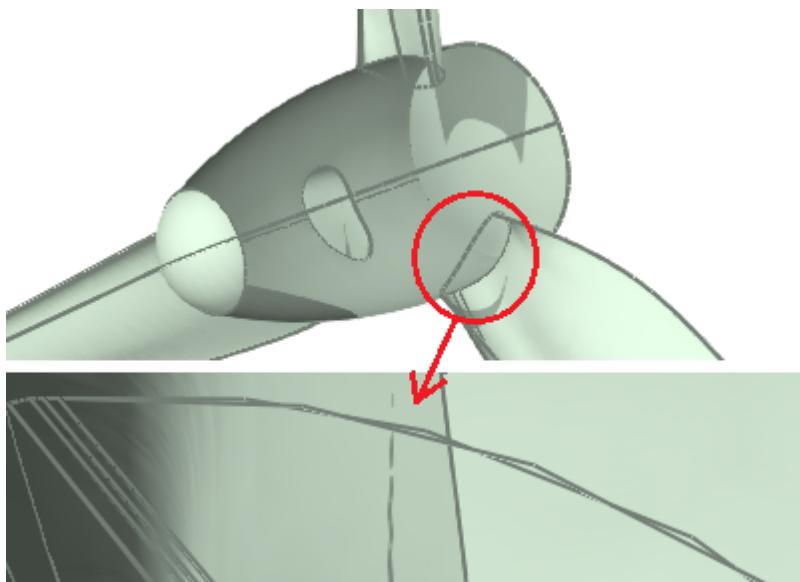
## Step 1: Geometry healing



Geometry: blades-to-heal (Shading)

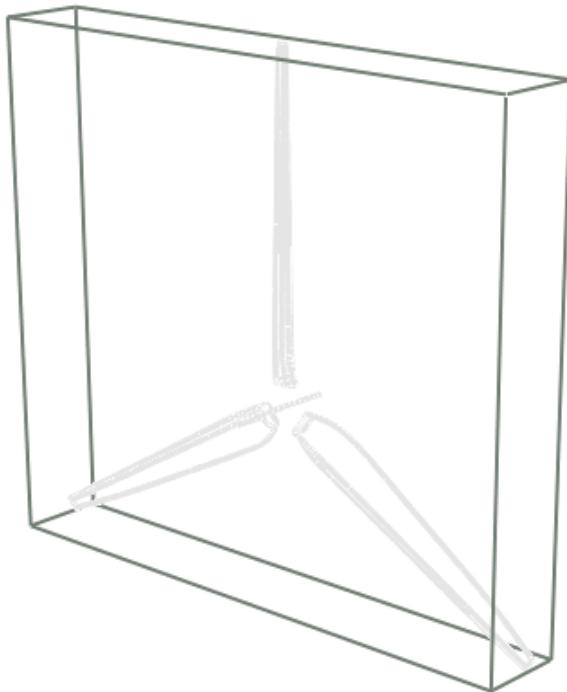
### 1.3 Show holes

Furthermore, the surfaces (patches) defining the geometry are not sewed, there are gaps between them. For instance, the blades are not properly joined to the central part. 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**).



*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 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 operations (be careful that the model is unselected after finishing an operation, so you will have to select it again):

**Main menu > Geometry > Healing > Fix small lines**

**Main menu > Geometry > Healing > Sew faces**

**Main menu > Geometry > Healing > Complete healing**

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

## Step 1: Geometry healing

**⚠ Please note:** If the geometry has holes, fluid will leak inside and it will be initialised with the inlet velocity condition as the rest of the fluid but, because it is confined, generates pressure waves inside of the geometry that lead to wrong 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`. In model units, select metres and press **Apply to all**.

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

### 2.2 Engine settings

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

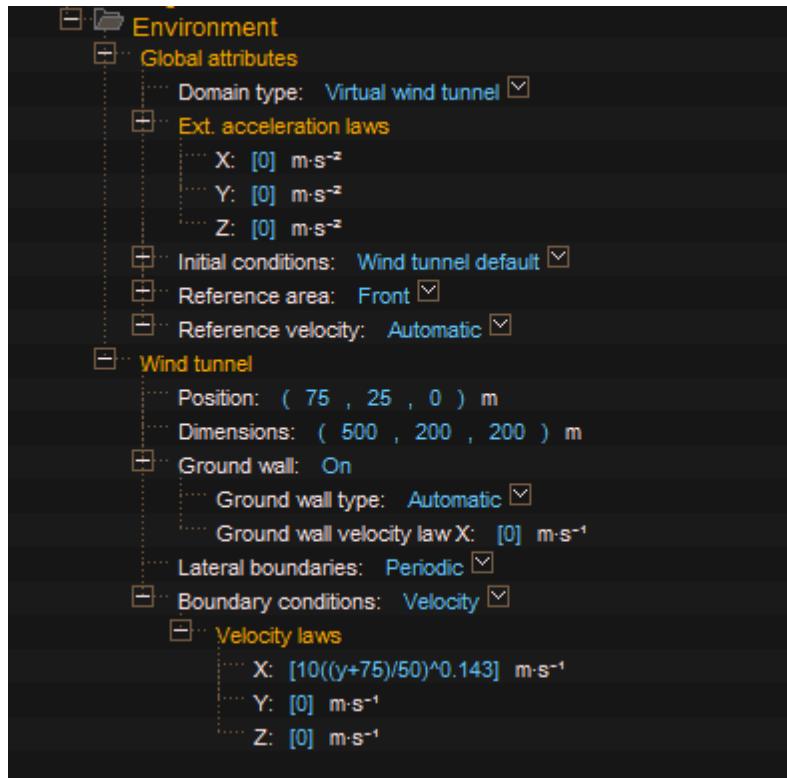
### 2.3 Environment settings

- (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 **Automatic Reference velocity**
- (e) Introduce the wind tunnel **Dimensions**: **(500, 200, 200) m**. By default the wind tunnel is centered 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<sup>-1</sup>**
- (h) Set the **Lateral boundaries** to **Periodic** and the inlet **Boundary conditions** as **Velocity** defined by the following power-law profile

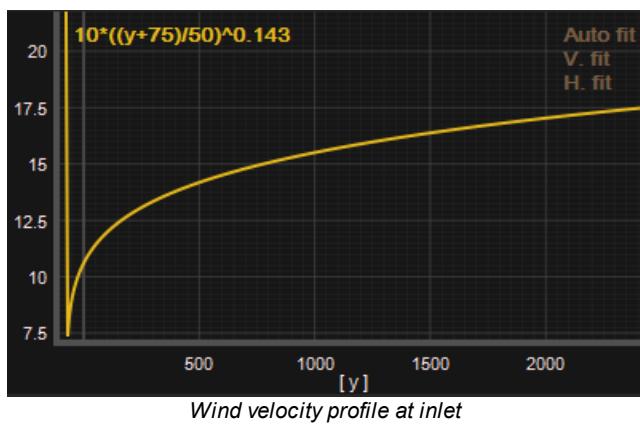
$$u = u_{ref} \left( \frac{y - y_0}{y_{ref}} \right)^\alpha$$

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 m** and coefficient  $\alpha = 0.143$  corresponding to neutral conditions.



- (i) Display the graph of the X-velocity law in the *Function Viewer* by right clicking the expression and selecting **Edit in Function Viewer 1**. 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**



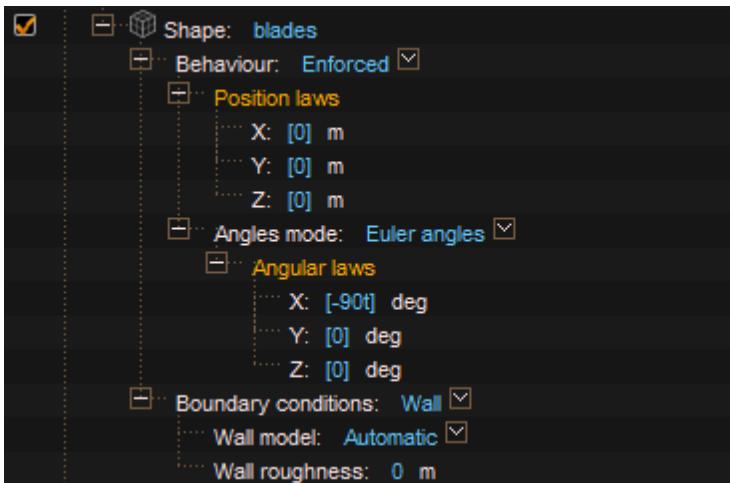
## 2.4 Material settings

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:

- (a) **Geometry > Entities > Shape: tower**: Behaviour **Fixed** and **Boundary conditions: Wall** with **Automatic Wall model** and **0 Wall roughness**.
- (b) **Geometry > Entities > Shape: blades**: **Behaviour: Enforced**
- (c) 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.
- (d) **Geometry > Entities > Shape: blades**: set the **Angular law** for the X-axis to **[ -90t ] deg** (15 rpm = 90 deg·s<sup>-1</sup>)
- (e) Check the enforced motion of the blades by pressing or dragging the current frame along the timebar



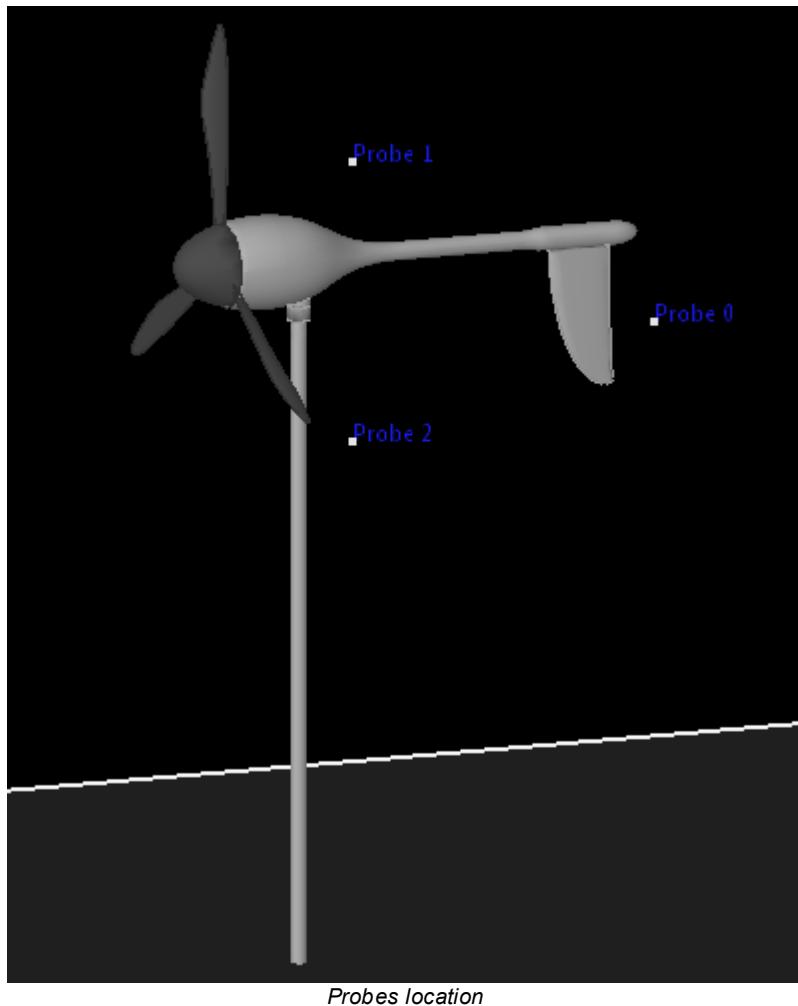
## 2.6 Simulation settings

- Set the **Simulation time** to **12 s**
- Time step mode** will be **Fixed automatic** with **Courant: 1**
- Set **8 m** for the **Resolved scale** (far field resolution)
- Select **Refinement algorithm: Adaptive refinement**
- 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**)
- Store data** with a **Frames frequency** of **25 Hz** and set the **Numerical data frequency** to **Solver time step**.
- Set the **Store data > Folder:** WindTurbine
- Leave **Save averaged fields** and **Compute markers** disabled

## 2.7 Probes

Probes are predefined points where data are measured and saved during computation. They allow the user to save numerical data at the solver time step frequency instead of the frames frequency as with the sensors.

- Create a probe by **right clicking** the string **Probes** and choose **Add probe**
- Set **Probe 1 > Position** to **(50, -10, 0) m**
- Create two additional probes at locations:  
**Probe 2 > Position: (15, 10, 0) m**  
**Probe 3 > Position: (15, -20, 0) m**
- Show the probes position by selecting them 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.

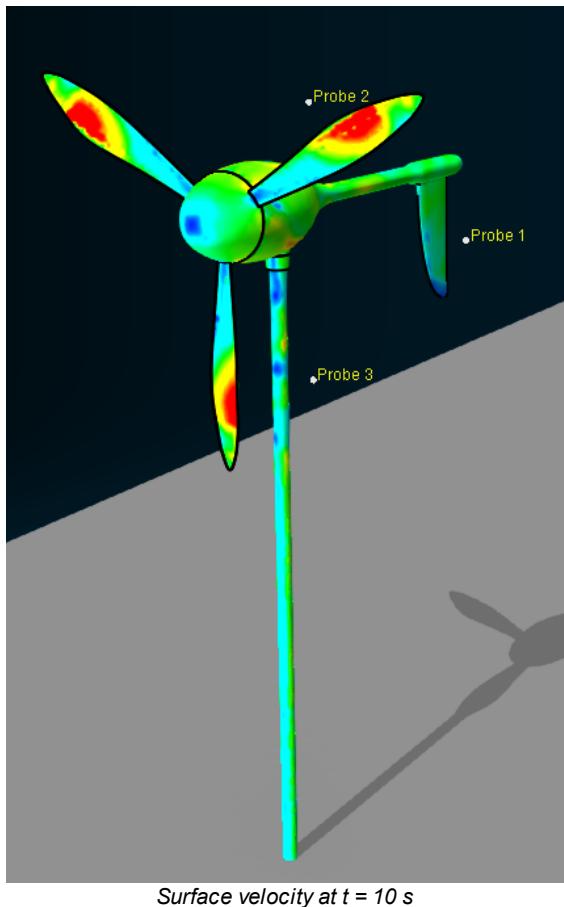
Load the frames 10, 100 and 200 and check the number of particles in the *Message View*.

	Frame 10	Frame 100	Frame 200
<b>Level 0</b>	38016	37434	36652
<b>Level 1</b>	7691	9728	12072
<b>Level 2</b>	17389	24426	35666
<b>Level 3</b>	41509	91724	168336
<b>Level 4</b>	174044	662884	1333572
<b>Total</b>	278649	826196	1586298

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

- (a) Enable Post-processing > Show > Surface info: **3d field** and choose **Velocity** as **Field**
- (b) 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.

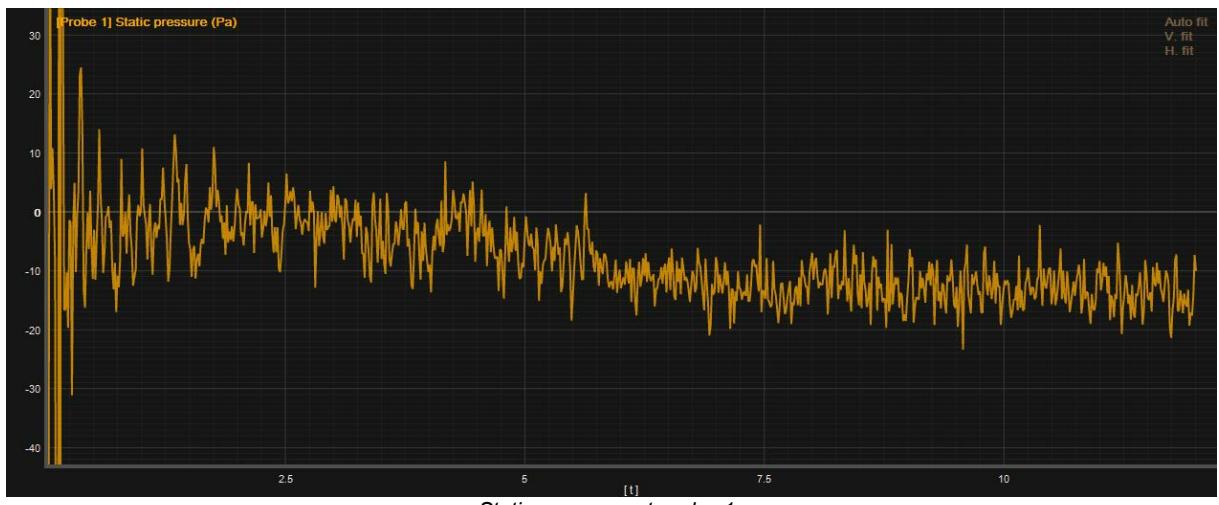


Surface velocity at  $t = 10$  s

- (c) Show the pressure coefficient ([Show > Surface info: Cp](#)). Adjust the range to [-5, 5] in [Main menu > Simulation data > Analysis settings > Surface info](#).

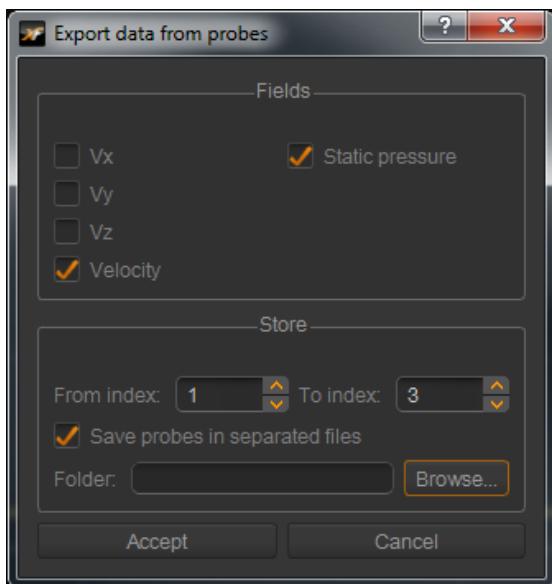
### 3.3 Probes measurements

- (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**



(b) You can export the data of the probe **right clicking** in *Function Viewer* > **Export all**. A pop-up window appears and asks you to select which fields and which probes would you like to export. Select the following options:

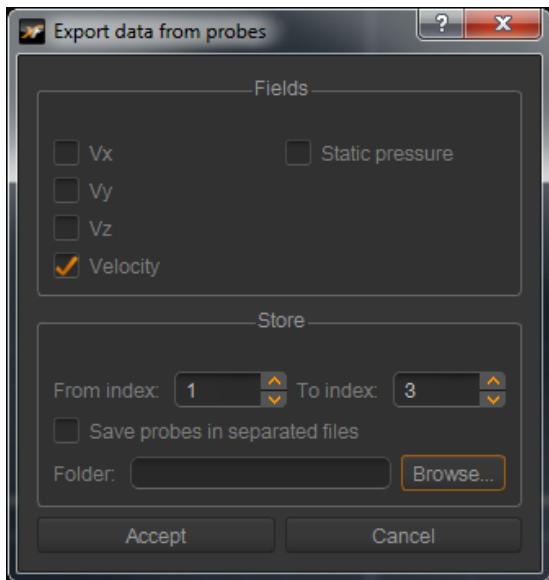
- **Fields:** **Velocity** and **Static pressure**
- **Store:** **From index:1, To index: 3.** This means that you will export the data of the Probe1, Probe 2 and Probe 3.
- Enable the "**Save probes in separated files**" option. This means that the data of each probe will be exported in separated files 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 now only the **Velocity** field measured by the probes 1 to 3 and leaving the "**Save probes in**

## Step 3: Enforced behaviour - Post-processing

"separated files" option unchecked:



XFlow will export now the Velocity measurements of the three probes to one file called "probe\_1\_3.txt".

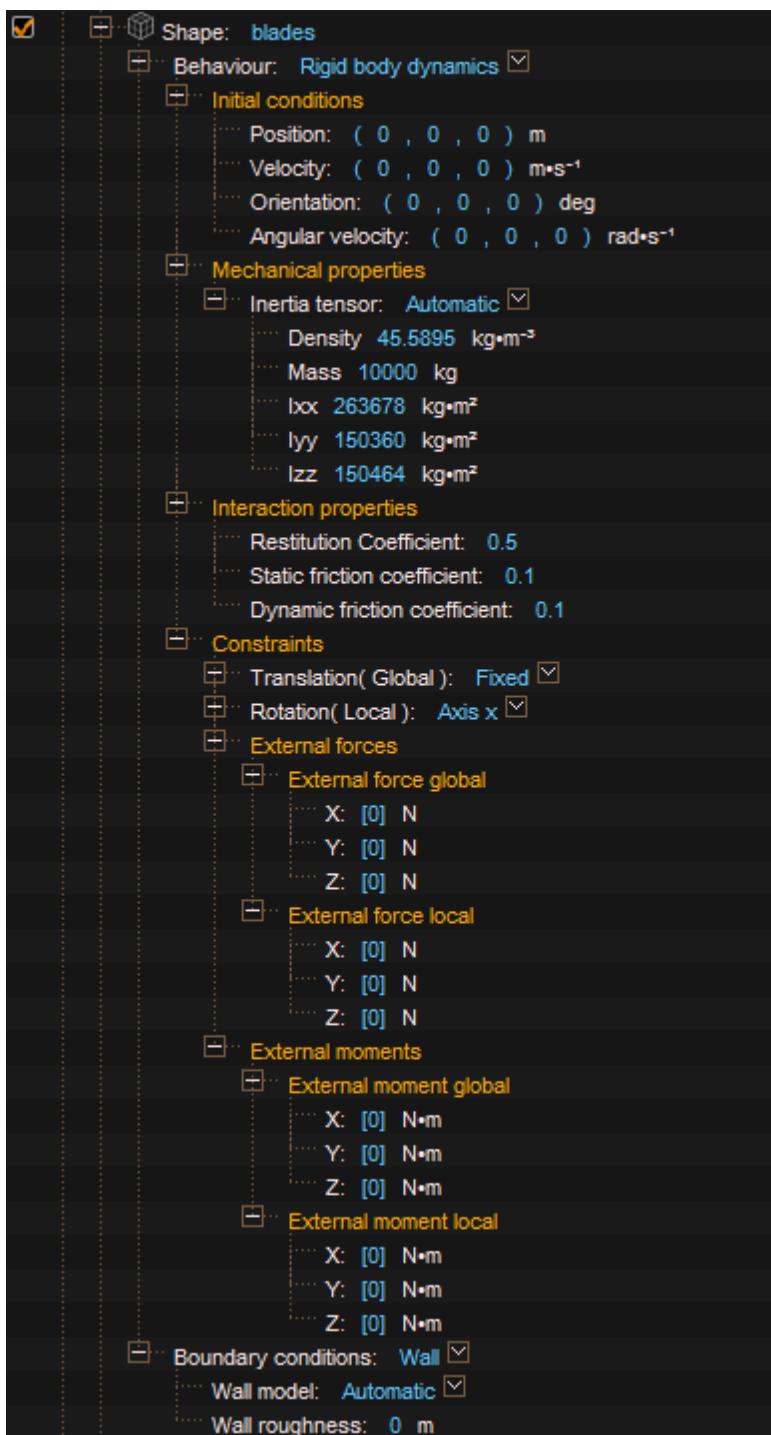
**⚠ Please note:** The Probes are working in the same way as the Sensors (used in the Tutorial 04 for example). The difference is that the Probes need to be defined before running the calculation to be accessible in the Function Viewer, but then XFlow does not need to refresh every frame to load the data as with the Sensors. Therefore, this allows you to save time for post-processing when you know in advance where to probe, and furthermore the probes can measure the information at the solver time step instead of the frames frequency as for the sensors.

## Step 4: Rigid body dynamics behaviour

### 4.1 Case setup

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

- In **Geometry > Shape: blades** set the **Behaviour** to **Rigid body dynamics**, with:
  - Density:** **45.5895 kg·m<sup>-3</sup>** which corresponds to a mass of **10000 kg**
  - Translation (global): Fixed, Rotation (local): Axis x**



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

- In **Simulation**, set the **Simulation time** to **30 s**
- Change the **Store data > Folder**, e.g. to "WindTurbine\_RBD"

## Step 4: Rigid body dynamics behaviour

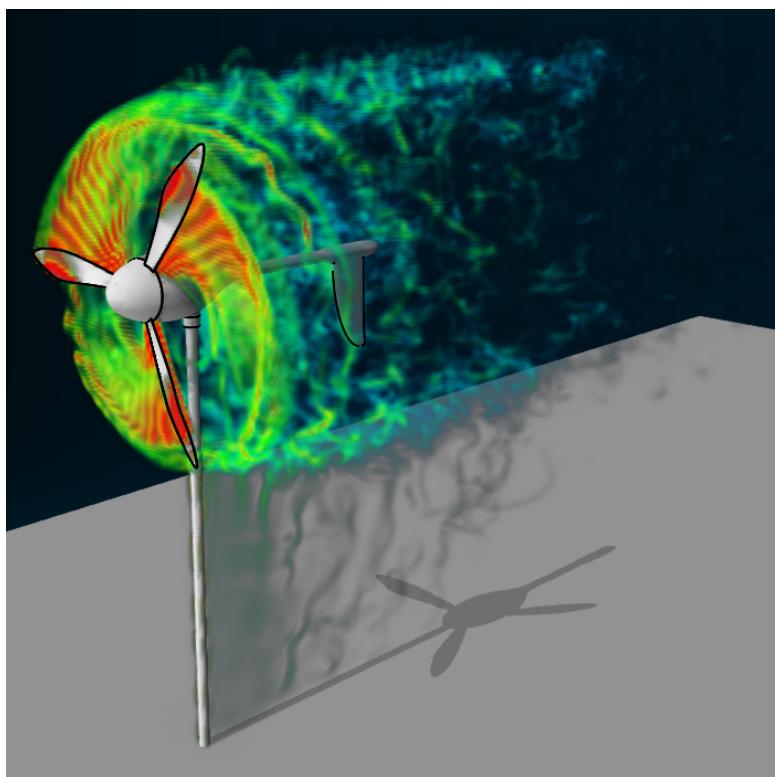
- (d) In this case, a frequency of **10 Hz** will be enough
- (e) Remove all the probes by **right clicking** the **Probe #** and selecting **Remove probe**
- (f) Save the project with a different name and run the calculation



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

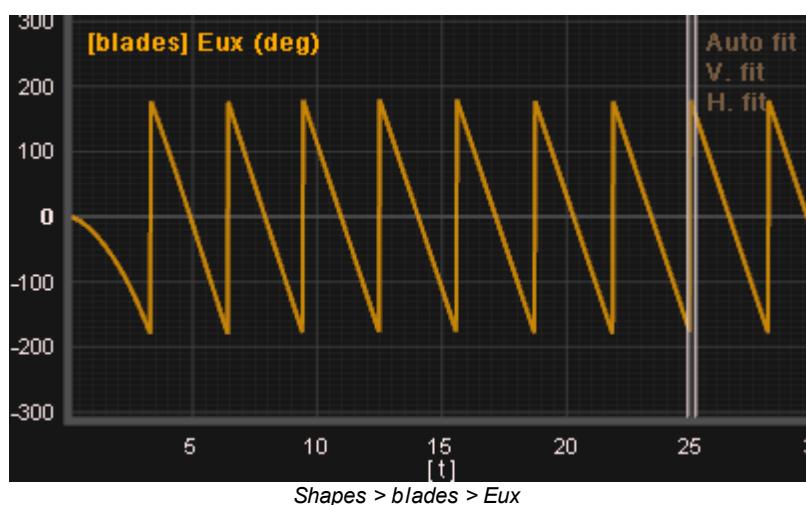
## 4.2 Post Processing

- (a) Play the simulation from the first frame to see how the blades start to rotate
- (b) Choose the **Vorticity** in **General > Show > Volumetric field > Visualisation field**, adjust its range to **[0, 20]**, disable **General > Interpolation mode** and show the volumetric field with **Transference law:  $a^*a$**



3D vorticity field at  $t = 30$  s

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



Observe that the blades achieve a uniform rotational speed of approximately 2 rad/s = 19 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.

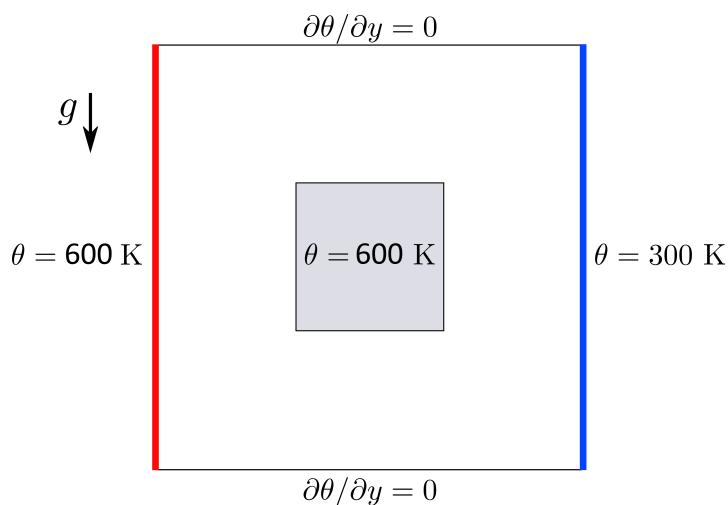
It consists of a square cavity with differentially heated vertical walls and adiabatic horizontal walls. The cavity is filled with air and a solid box is placed in its center. Fluid convection is driven by buoyancy forces, which are modeled using the Boussinesq approximation:

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

where  $\alpha$  is the thermal expansion coefficient,  $\theta_0$  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-process](#)

## Step 1: Problem setup

### 1.1 Create the geometry

(a) Create the cavity as a box (  ) with lower corner **(-1.5, -1.5, -1.5)** and upper corner **(1.5, 1.5, 1.5)**.

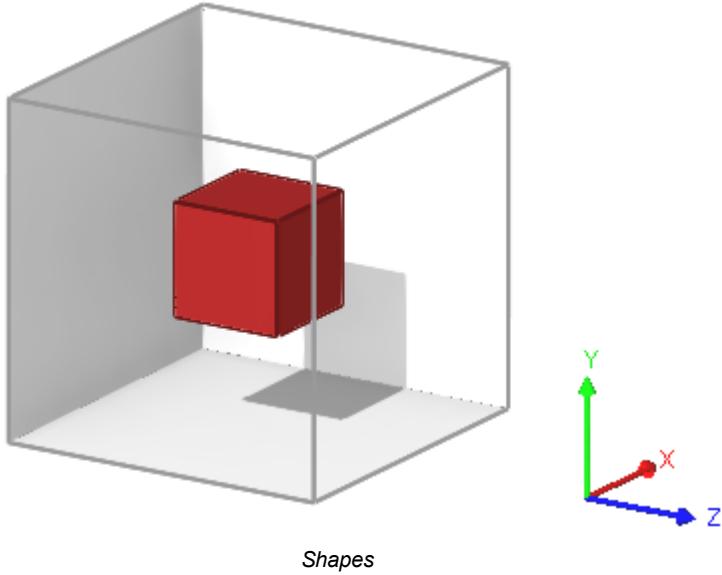
Check with the help of the **Back-face culling** that the box normals are orientated towards the interior and rename this **Shape** as **Cavity**.

(b) 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 significance in this example.
- (f) Leave the default **Turbulence settings**, **Acoustics analysis**, **Passive scalar transport** and **Advanced options**. Note that **Advanced options > Viscous term in energy equation** is enabled by default.

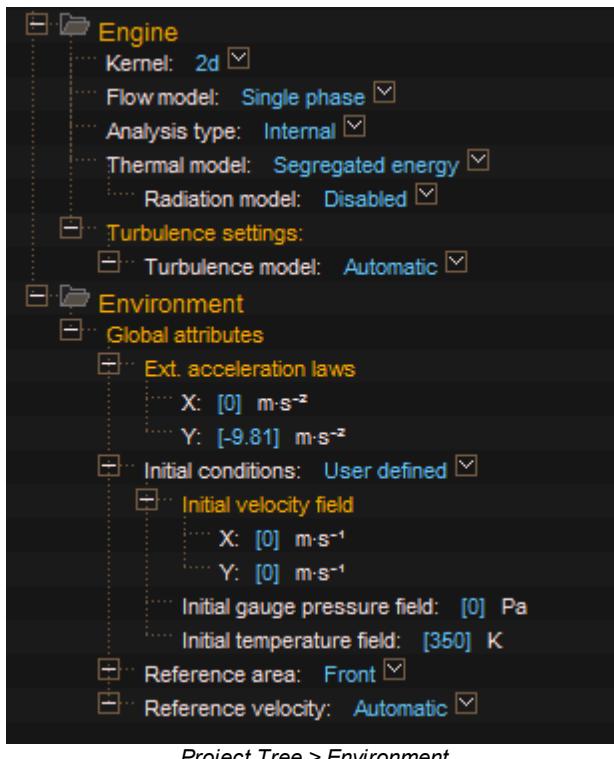
The effect of viscous dissipation in natural convection is appreciable only when the induced kinetic energy becomes comparable to the amount of heat transferred, and this occurs when either the buoyancy force is large or when the convection region is extensive. Viscous heat dissipation will not be relevant 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) Set the Initial temperature field to 350 K



Project Tree > Environment

## 1.4 Material settings

- (a) Set the Material 1 State equation to **Boussinesq**, with Density: **1 kg·m⁻³** and Thermal expansion coefficient: **0.1 K⁻¹**
- (b) Set the **Newtonian Dynamic viscosity** to **1e-5 Pa·s**, the **Thermal conductivity** to **10 W·(m·K)⁻¹** and the **Specific heat capacity** to **200 J·(kg·K)⁻¹**

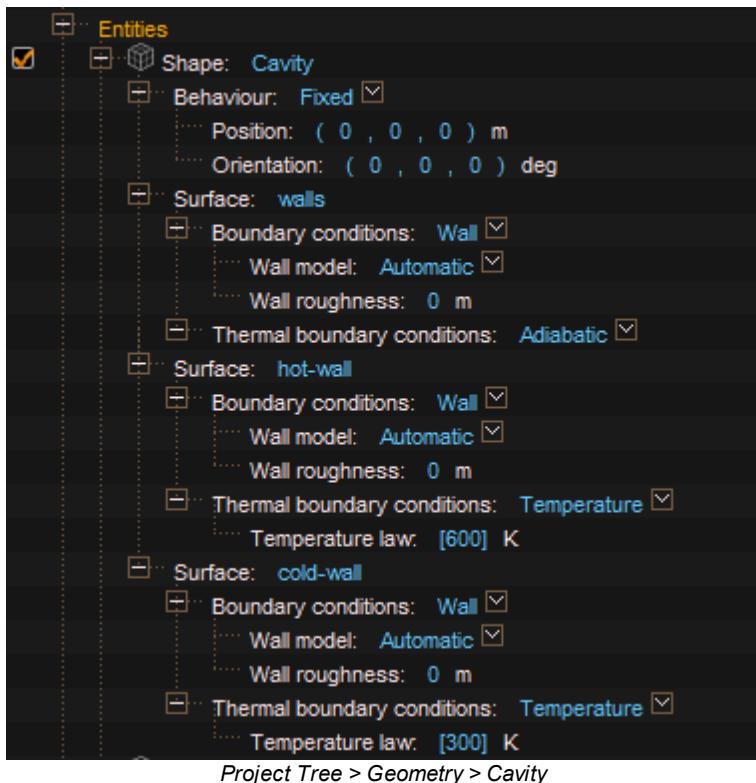
## 1.5 Geometry settings

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

- (a) Choose the selection mode "Face filter"  in the toolbar . Select a vertical face of the Shape: **Cavity** (it highlights), do **right click** on the *Graphic View* and choose **Apply boundary conditions to faces** from the contextual menu
- (b) 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
- (c) Select the opposite face, **right click** in the *Graphic View*, and choose **Apply boundary conditions to faces**

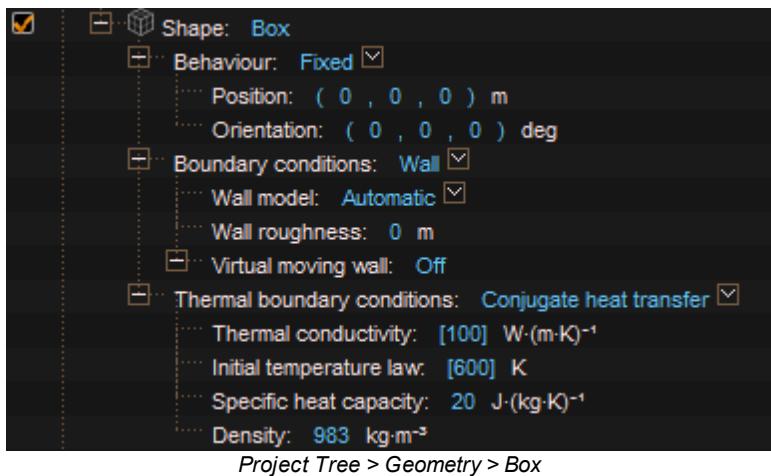
## Step 1: Problem setup

- (d) Check that a new surface called "surface 2" appears in the *Project Tree*
- (e) Rename "surface 0" to "walls", "surface 1" to "hot-wall" and "surface 2" to "cold-wall"
- (f) Return to the View only selection mode by choosing  in the toolbar
- (g) Set all surfaces to **Wall Boundary conditions**, with **Automatic Wall model** and zero **Wall roughness**
- (h) 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**



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

- (a) Set the whole **Shape** to **Wall Boundary conditions**, with **Automatic Wall model** and zero **Wall roughness**
- (b) Set the **Thermal boundary conditions** to **Conjugate heat transfer**, with a **Thermal conductivity** of **100 W·(m·K)<sup>-1</sup>**, **Initial temperature law: 600 K**, **Specific heat capacity: 20 J·(kg·K)<sup>-1</sup>** and **Density: 983 kg·m<sup>-3</sup>**



## 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 both **Shapes**).
- Store data > Folder:** HeatTransfer
- 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
- Tip:** You can directly load the setup of this problem from the project file `HeatTransfer.xfp`
- Set the number of CPUs in **Main menu > Options > Preferences > Engine**
  - Press **Run** button > **Start computation**

## Step 2: Post-processing

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

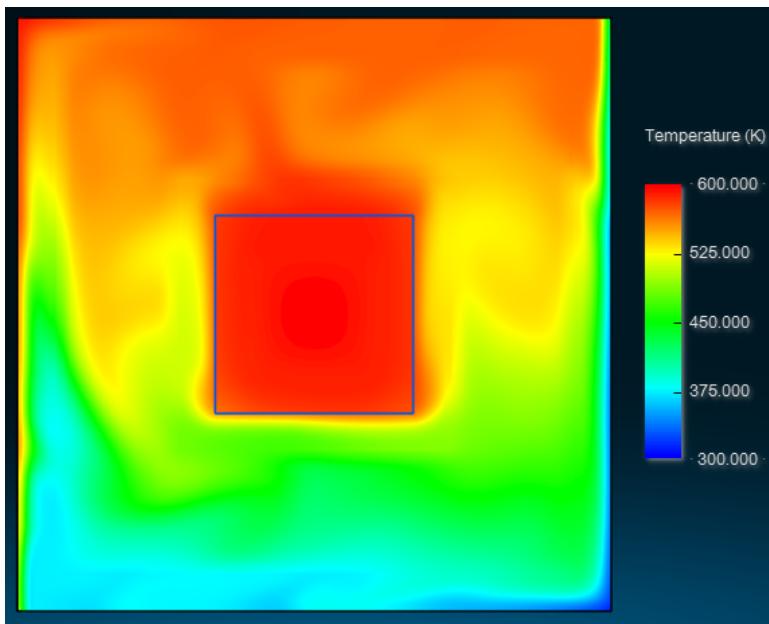
### 2.1 Temperature field analysis

- Post-Processing > Cutting planes.** Create a cutting plane on the Z axis and visualise the

## Step 2: Post-processing

**Temperature field.**

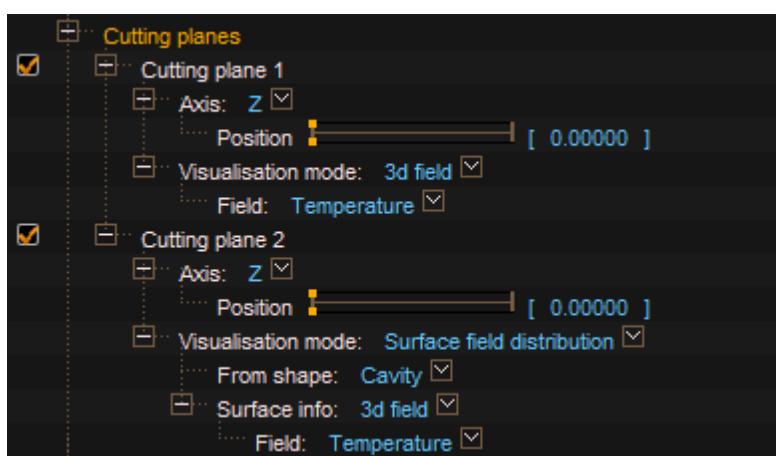
- (b) Select the solid box and display it in wireframe mode to be able to visualise the results at its interior  
Right click in *Graphic View* > **Visualisation mode** > **Wireframe**. Play forward the results and observe the influence of the solid's temperature in the overall solution.



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

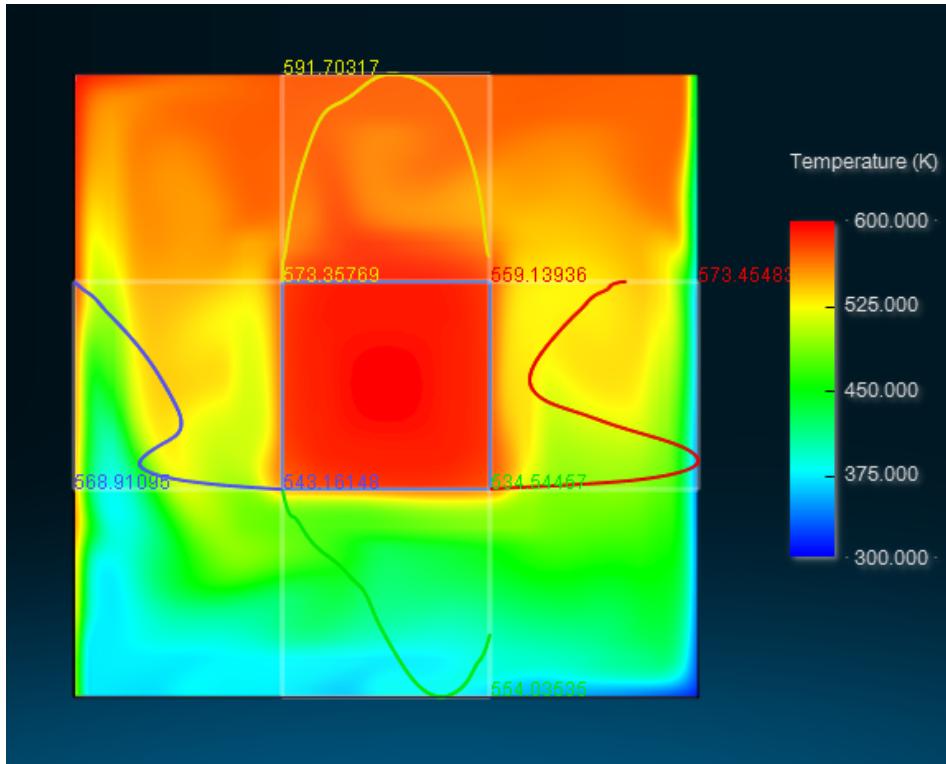
- (c) Create a second cutting plane and display the temperature distribution along the Box walls by selecting **Post-Processing** > **Cutting planes** > **Cutting plane 2** > **Visualisation mode: Surface field distribution**

- (d) Adjust the the temperature range to be [300, 600] K



**Please note:** This is the temperature of the fluid touching the wall, not the wall 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.



Temperature distribution on the Box geometry at frame  $t = 3\text{ s}$

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

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

(e) Show the isocontours: **Post-Processing > General > Show > Isocontours**

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

Refresh the **Plot lines > Right click on Line 1 > Refresh** or **click on** . Then, in the **Function Viewer** do **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, 600] in the Y-coordinate. To do so, remember that:

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

## Step 2: Post-processing

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

To adjust the X and Y axis:

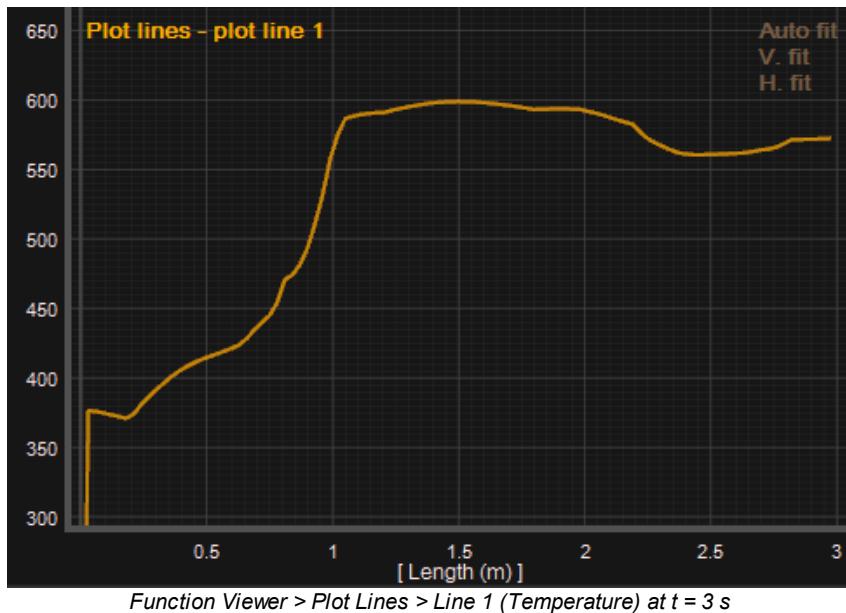
**Alt + right mouse button** + drag rightward      Horizontal zoom in

**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.



- (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, 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) Note that the flux is constant at the top and bottom walls. Check that it is zero, as corresponds to an adiabatic wall.

- (c) 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.
- (d) 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

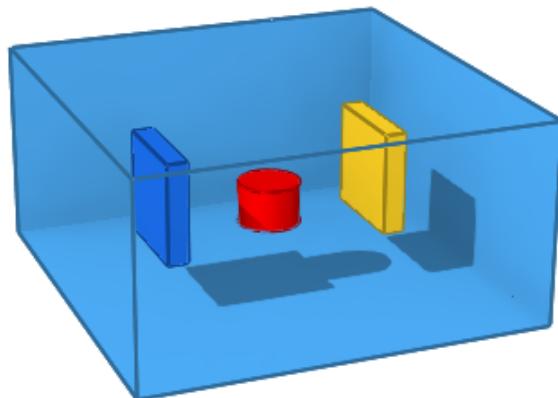
- (a) Enable again the **Cutting plane 1**, choose the **Velocity** as **Visualisation field** and show it as particles in: **Post-Processing > Cutting planes > Cutting plane 1 > Visualisation mode: Markers**

Play the results forward to see the movement of the particles.

# 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 geometry 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-process](#)

## 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**. 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<sup>-2</sup>** 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**.

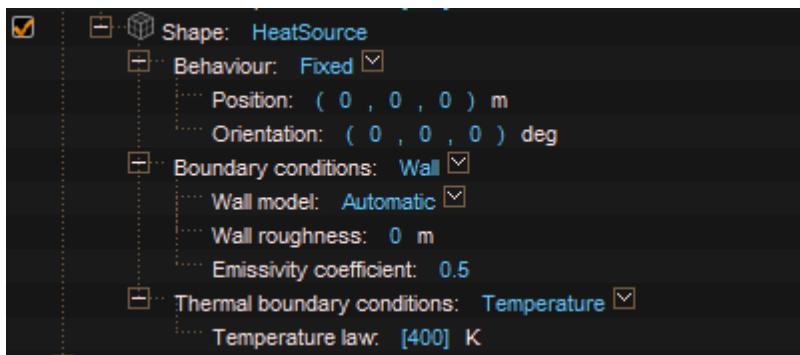
### 1.4 Material settings

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**, and **Thermal boundary conditions** with prescribed **Temperature** at **400 K**.

## 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) **Store data > Folder:** Radiation
- (d) Set the **Frames frequency** to **10000** frames per second and leave the **Numerical data frequency** by default
- (e) 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.

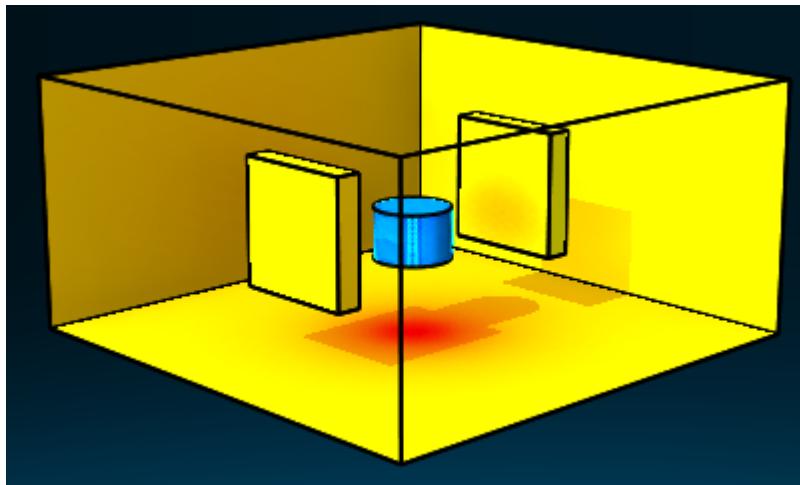
## 2.1 Read the Message View output

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 (= 0.00005 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
-----
-----
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.
WARNING::Reference pressure undefined!!
-----
-----
Thermodynamic speed of sound: 340.112
Reference area: 1 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: 42%
Progress: 53%
Progress: 64%
Irradiance map generated!
Saving data...
```

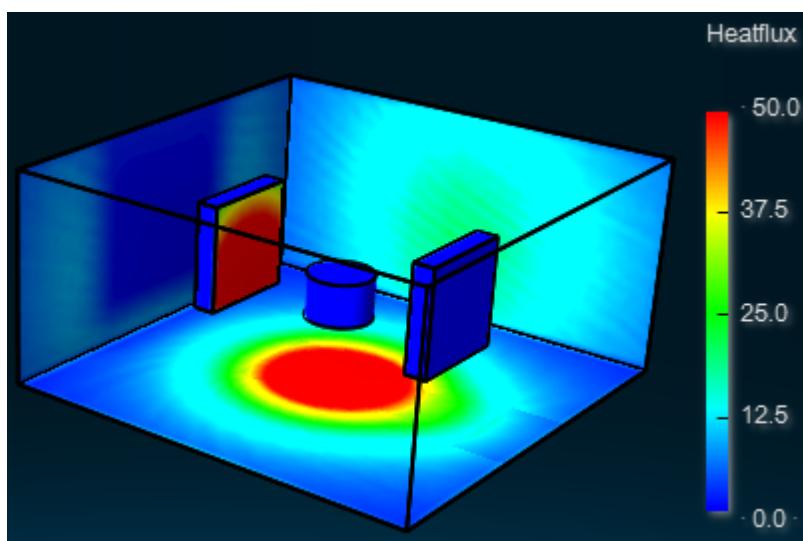
## 2.2 Analyse the heat flux at the surfaces

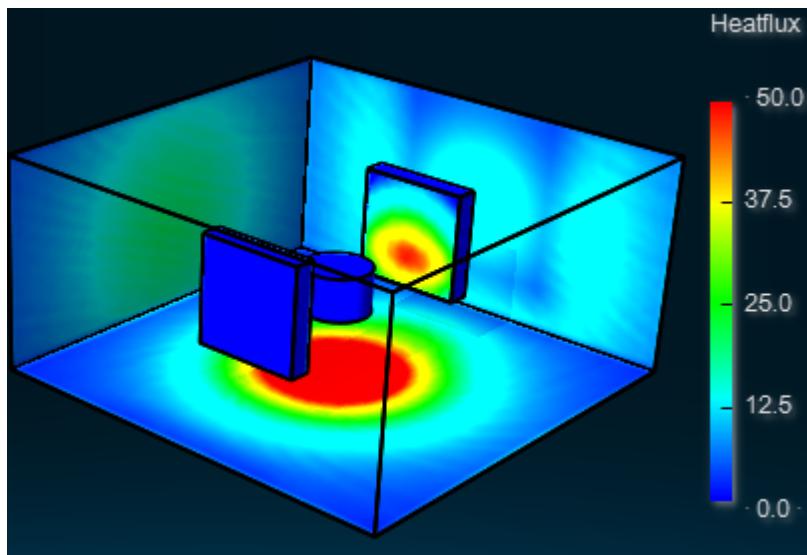
- (a) Go to frame 0 and show the total heat flux at the surfaces: **Post-Processing > General > Show > Surface info: Heat flux**



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 pure radiative, while in frames 1 and 2 the heat flux has also contributions from 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 **Visualisation 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.

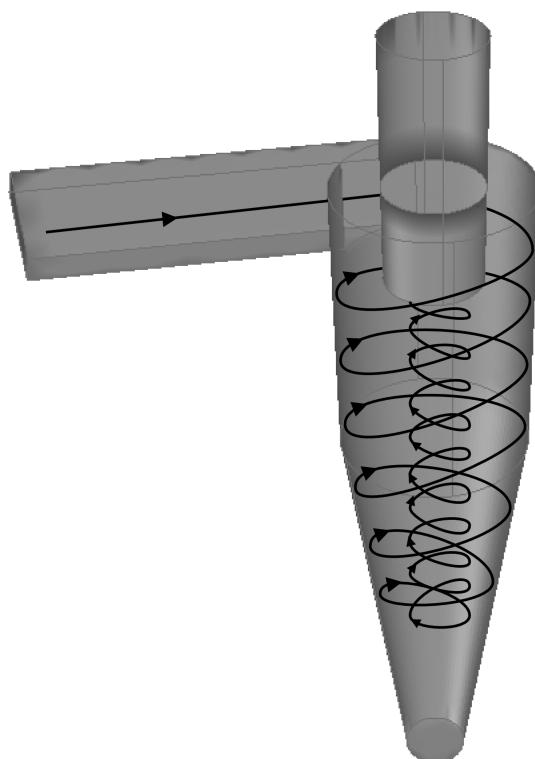
# Tutorial 10 - Cyclone flow

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: Run](#)

## Step 1: Problem setup

### 1.1 Engine settings

- Select the **3d Kernel**
- Select the **Flow model Single phase**
- 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
- Leave the default options for **Thermal model** and **Turbulence settings**

### 1.2 Import the geometry

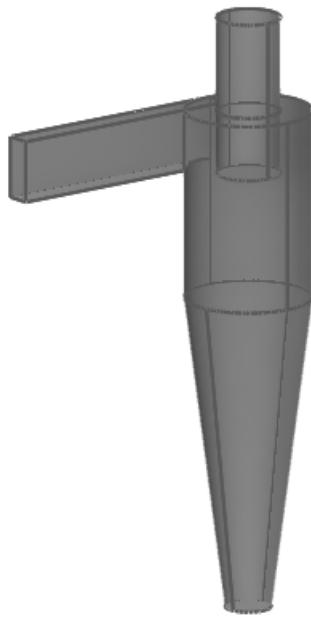
- Import the geometry of the chamber from the file **CycloneChamber.stp** (**Main menu > Geometry > Import a new geometry** or ). Note that the vertical direction of the chamber, when it is imported, is in the Z-axis.
- Rotate the chamber so that the vertical direction is parallel to Y-axis: **Geometry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



- Check the orientation of the normals and make sure they point to the inside, i.e. to the region with

## Step 1: Problem setup

fluid. If required, select the geometry and do **right click** in *Graphic View* > **Reverse orientation**



*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

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

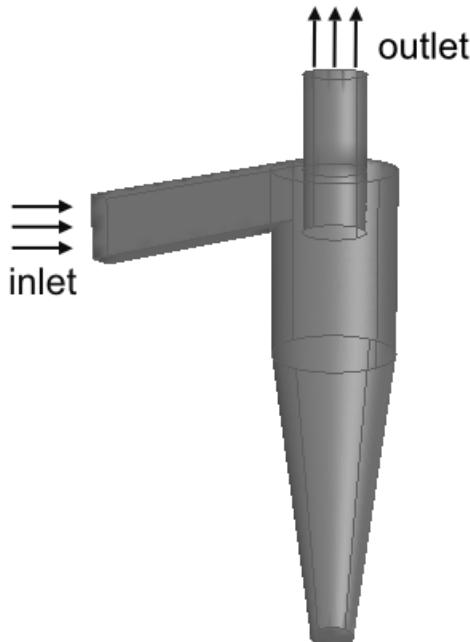
### 1.5 Geometry settings

- Set the **CycloneChamber Behaviour** as **Fixed**

- Set the inlet and outlet boundary conditions:

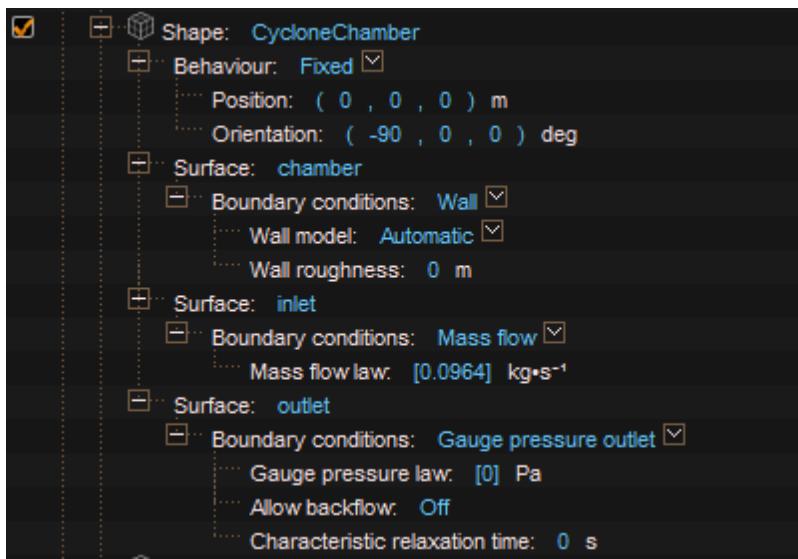
- Choose the "Face filter" mode  in the toolbar.
- Select the inlet surface (it highlights) and by **right clicking** on the *Graphic View*, choose **Apply boundary conditions to faces** from the contextual menu
- Check that in the **Geometry** section of the *Project Tree*, two surfaces appear in the **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

*Boundaries schematic*

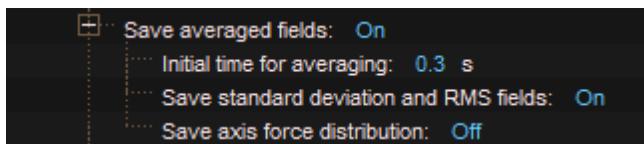
- (c) Leave the **Wall Boundary condition** at the **CycloneChamber ( Automatic )**
- (d) Set the **Boundary condition > Inlet > Mass flow** to the **inlet** surface. It will be assumed that the air inflow rate is  $0.08 \text{ m}^3 \cdot \text{s}^{-1}$ . This is equivalent to a **Mass flow law** of (air density)\*(volumetric flow rate) =  **$0.0964 \text{ kg} \cdot \text{s}^{-1}$**
- (e) Set the **Gauge pressure outlet** to the **outlet** surface, with **Gauge pressure law: 0 Pa**, and **Allow backflow: Off** and **Characteristic relaxation time: 0 s**

## Step 1: Problem setup



### 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**
- Store data > Folder:** Cyclone
- 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



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

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



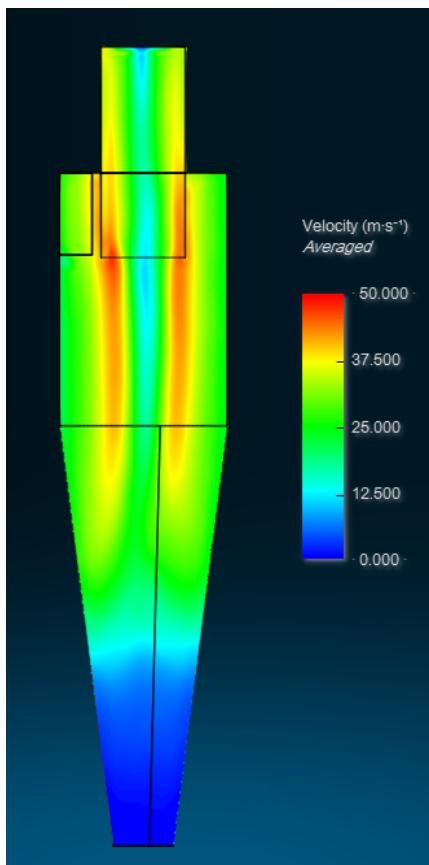
**Please note:** Depending on your hardware, the computation of this tutorial may take a few hours and around 5 GB of hard disk.

## Step 2: Post-processing

### 2.1 Visualise the velocity field on a cutting plane

In Post Processing > Cutting planes

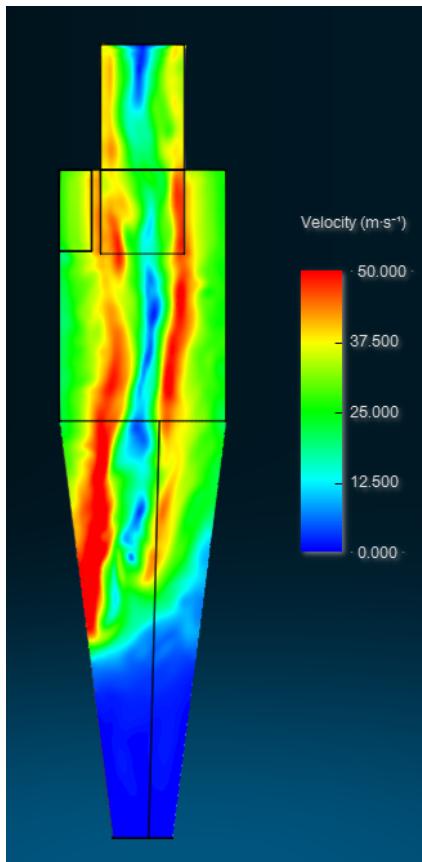
- (a) Create a **Cutting plane 1** on the Z axis and adjust its **Position** to **0.2**
- (b) Choose **Visualisation mode: 3d field**
- (c) Choose **Field: Velocity**
- (d) Set the **General > Interpolation mode: Convolution**
- (e) Select view from right 
- (f) Choose **General > Data: Averaged**
- (g) Adjust the legend range of velocity to **[0, 50] m·s<sup>-1</sup>** (**Main menu > Simulation data > Analysis settings** or - (h) 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  
 $t = 1$  s

## Step 2: Post-processing

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



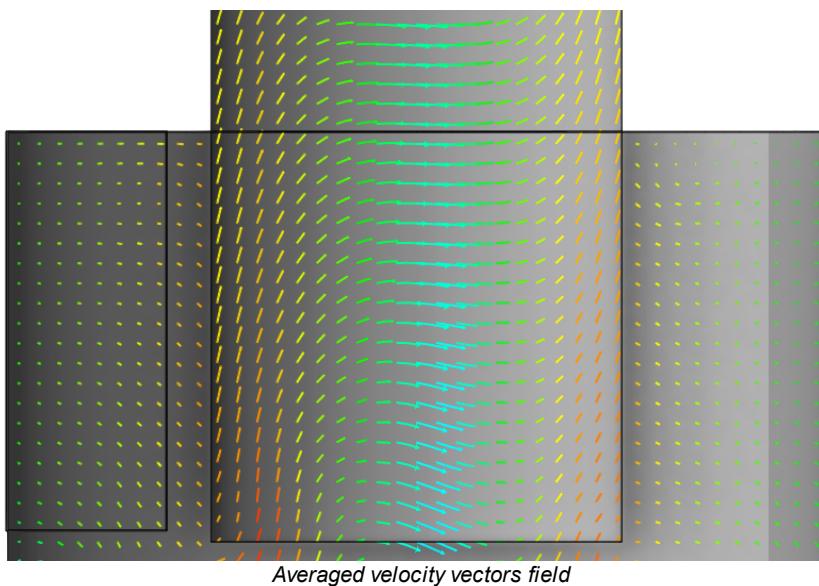
*Instantaneous velocity field on cutting plane 1 at  $t = 1\text{ s}$*

(i) Show also static pressure, vorticity,  $V_y$ ,  $V_z$  and turbulence intensity fields. For each field, switch 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 Visualise the velocity vector field on a cutting plane

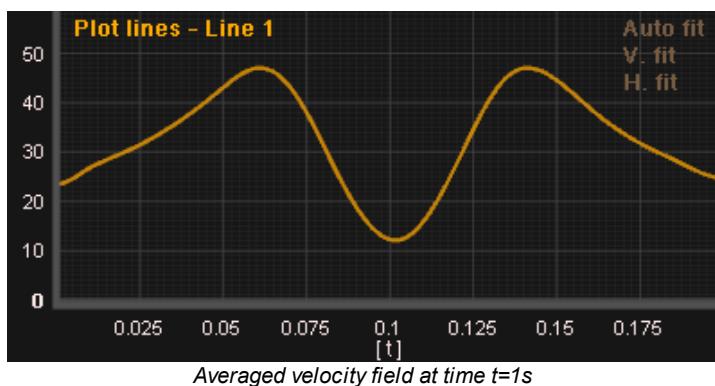
In Post-Processing > Cutting planes > Cutting plane 1

- (a) Go to the last frame available
- (c) Choose **Visualisation mode:** **Vectors** with **Uniform distribution:** **On** 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



### 2.3 Measure velocity field along a line

- Create a plot line by **right clicking** on **Post-Processing > Plot lines** and selecting **Add plot line**
- Line goes from **Vertex 1: (-0.1, -0.3, 0)** to **Vertex 2: (0.1, -0.3, 0)**
- Set the **Number of samples** to **1000**. This is the number of points used to draw the curve, therefore the higher the more accurate
- Select the **Velocity** field and **Averaged** data, with interpolation **On**
- Right click** in **Function Viewer > Plot lines > Line 1** to display the evolution of the visualisation field along the line:

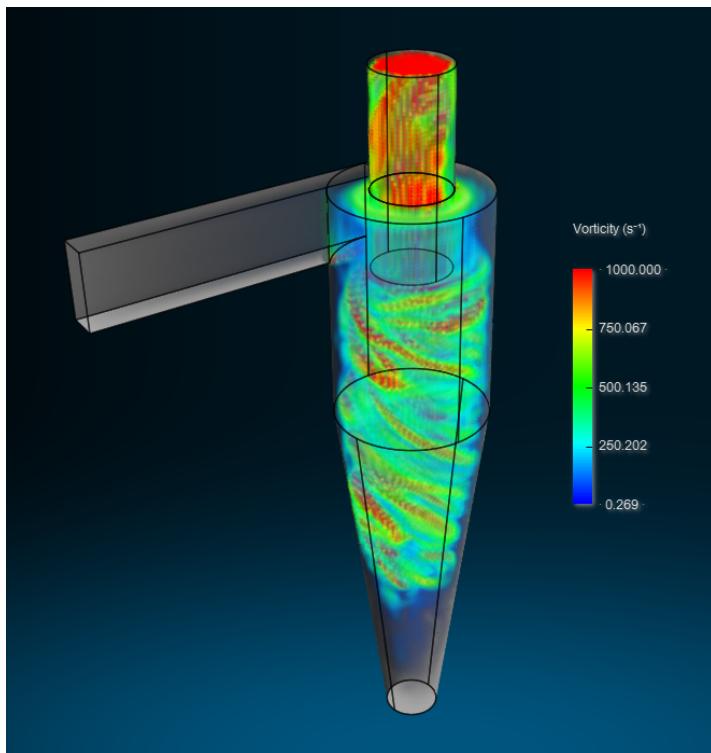


### 2.4 Visualise the vorticity volumetric field

- Deactivate any cutting plane or plot line
- Set the **Interpolation mode** to **Off**
- Set the vorticity range to **[0, 1000]** (**Main menu > Simulation data > Analysis settings** or

## Step 2: Post-processing

- (e) Set Post-Processing > General > Data: Instantaneous
- (f) Enable the visualisation of the Volumetric field and visualise the Vorticity field with a transference law:  $\mathbf{a}^*\mathbf{a}$
- (g) Press play 
- (h) Change the Transference law to  $\mathbf{a}$



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

## Step 3: Stream tracers

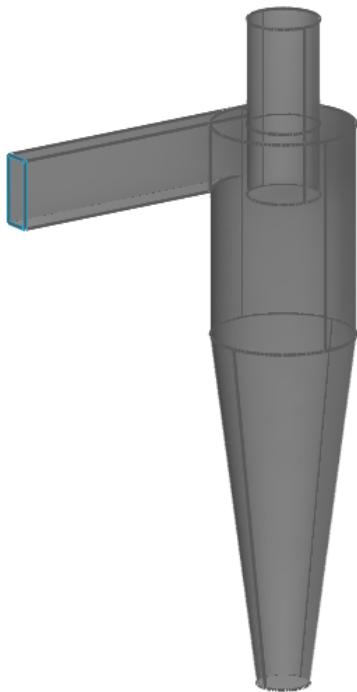
Stream tracers integrate the streamlines in the velocity field and so allow to track the trajectory of Lagrangian particles along the flow.

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) The new surface (called "CycloneChamber-Duplicated") appears as a **Shape** in the **Geometry** section of the *Project Tree*. Rename it to "**Surface**" and move it **1 cm** to the interior.
- (d) Set the surface **Behaviour** to **Disabled** to avoid that it affects the stream tracers. If the behaviour would be fixed/enforced/rigid-body-dynamics, the surface could act as a wall for the stream tracers



- (e) 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 **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**

### Step 3: Stream tracers

(d) Set the other parameters to:

Number of tracers: **1** (Only one particle)

Particles flux rate: **0 particles·s<sup>-1</sup>**

Data: **Averaged**

Turbulent dispersion: **Off**

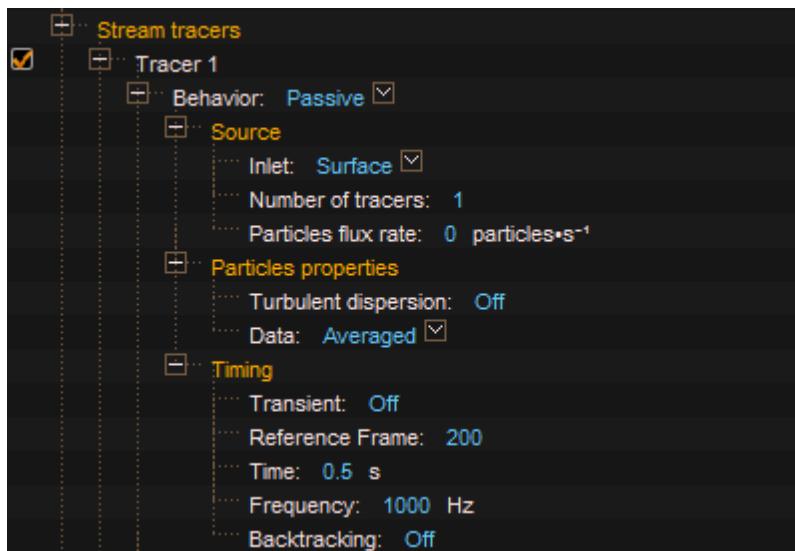
Transient: **Off**

Reference frame: **200**

Time: **0.5 s**

Frequency: **1000 Hz**

Backtracking: **Off**



(d) Right click on **Tracer 1** and select **Recompute** or

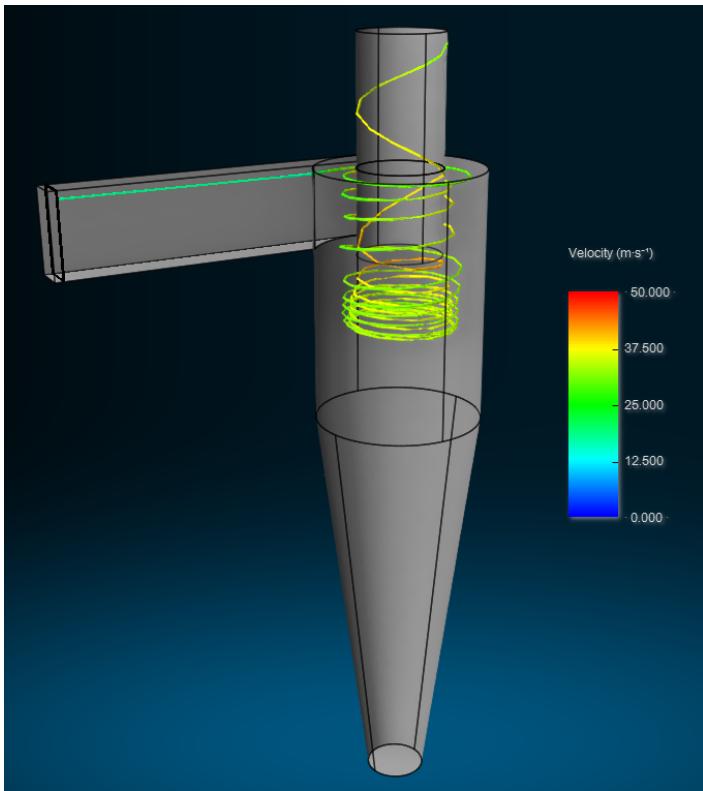
Observe that appears a second timeline (in red colour) related to the stream tracers, and that the frame 200 is locked. To unlock the visualisation of the Tracer 1, you must untick its checkbox.

(e) Choose **Tracer > Show > Current marker: Point** 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: 500** and refresh the **Path lines** by the refresh icon or by **Right clicking** on **Tracer 1** and **Generate path lines**

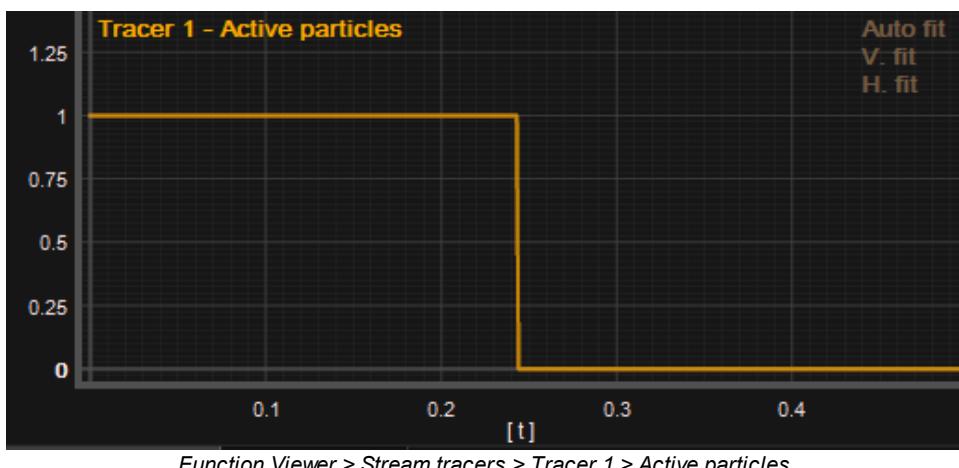
(g) When the generation finishes, display the line **Post-Processing > Stream Tracers > Tracer 1 > Show > Path line: Line**. You can also choose Ribbon, Sphere chain or Tube. You may need to

adjust the size factor to visualise these path lines correctly.



You can observe the streamline of one particle which goes to the outlet.

- (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 from time 0 s to 0.244 s, at this moment this particle exits the domain through the outlet and no particles remain



- (i) Untick the checkbox of **Tracer 1** to disable its visualisation and leave the timeline free

### 3.2.2 Transient tracer

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

(b) Set the same parameters than before except

**Data: Instantaneous**

**Transient: On**

**Initial frame: 0**

**Last frame: 50**

In transient tracers makes more sense to use instantaneous data.

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

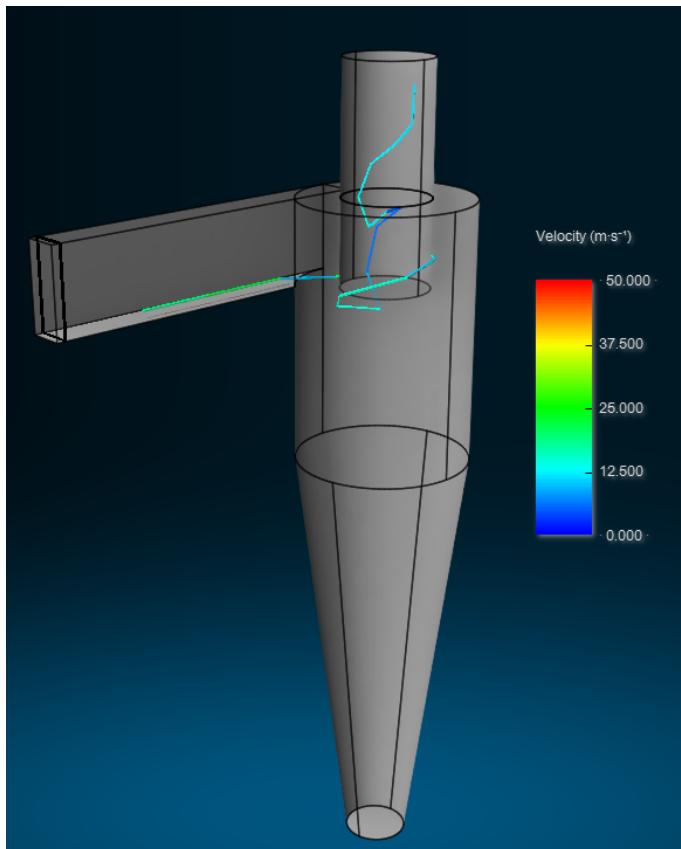
(d) Again **right click** on **Tracer 2** and select **Generate path lines**. From frame: **0** and To frame: **50**

(e) Show the line and the marker

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

Post-Processing > Stream tracers > Tracer 2 > Show > Current marker: **Point**

Post-Processing > Stream tracers > Tracer 2 > Show > Size factor: **0.02**



Observe that now 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 1000 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.

Restitution coefficient refers to the collision of particles with the surfaces. Restitution coefficient = 1 means perfectly elastic collision, Restitution coefficient = 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, _y, _z (particle position)	fluid_px, _py, _pz (fluid position)
particle_vx, _vy, _vz (particle	
velocity)	fluid_vx, _vy, _vz (fluid velocity)
particle_t (particle time)	fluid_sp (fluid static pressure)
	fluid_vrt (fluid vorticity)

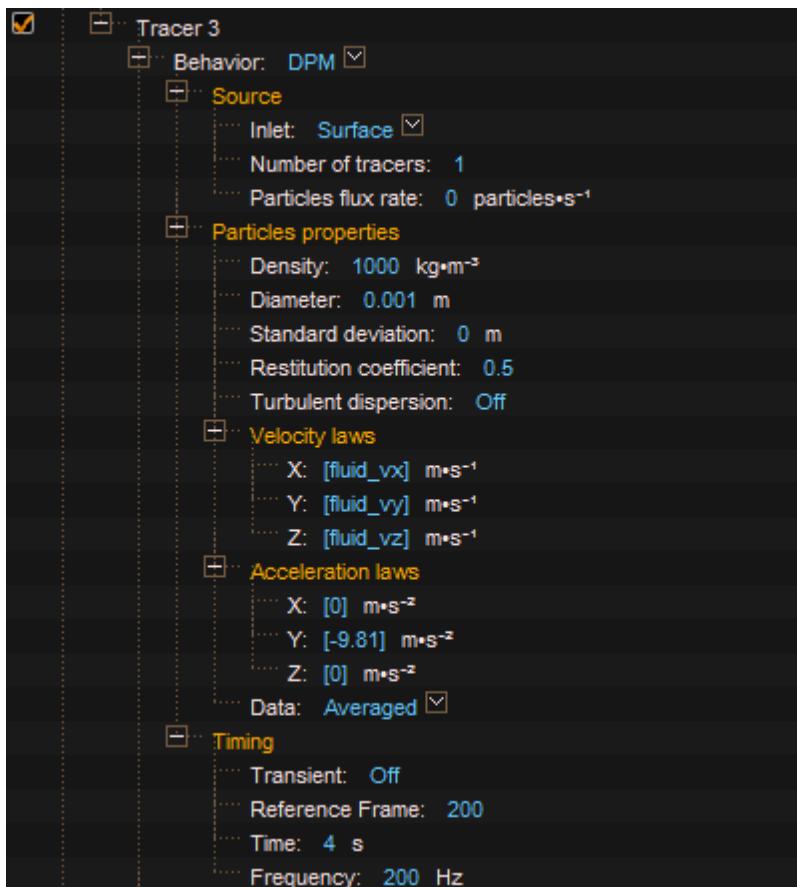
#### 3.3.1 Big particle

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

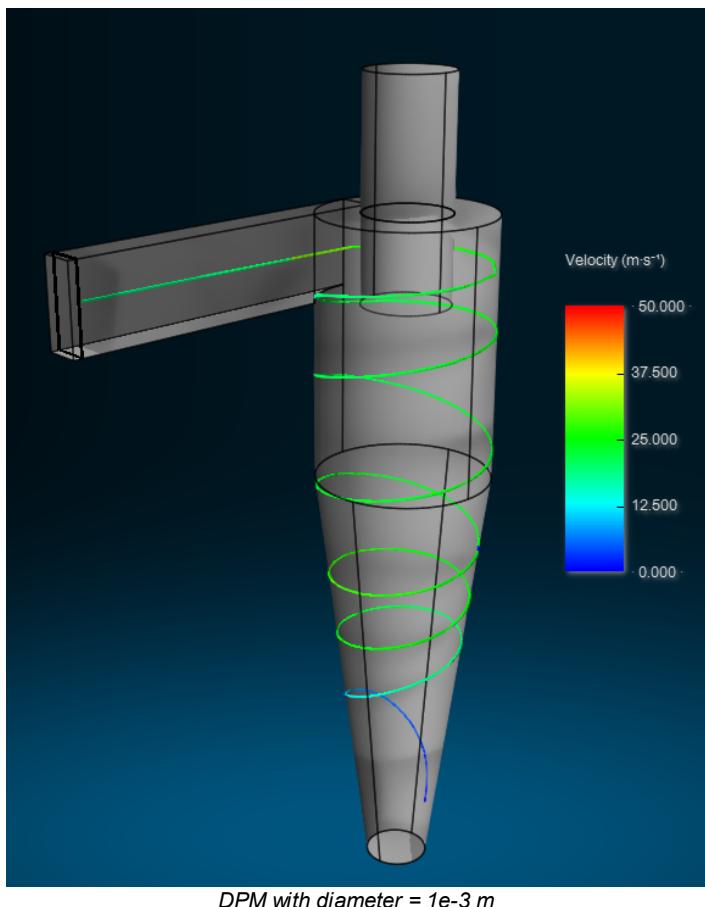
(a) Create a new stream tracer (right click **Stream tracers > Add stream tracer**). It is called **Tracer 3**.

(b) Set the following parameters

### Step 3: Stream tracers



- (c) Right click on **Tracer 3** and select **Recompute**
- (d) Again right click on **Tracer 3** and select **Generate path lines** from frame 0 to 800.
- (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 and deposits.

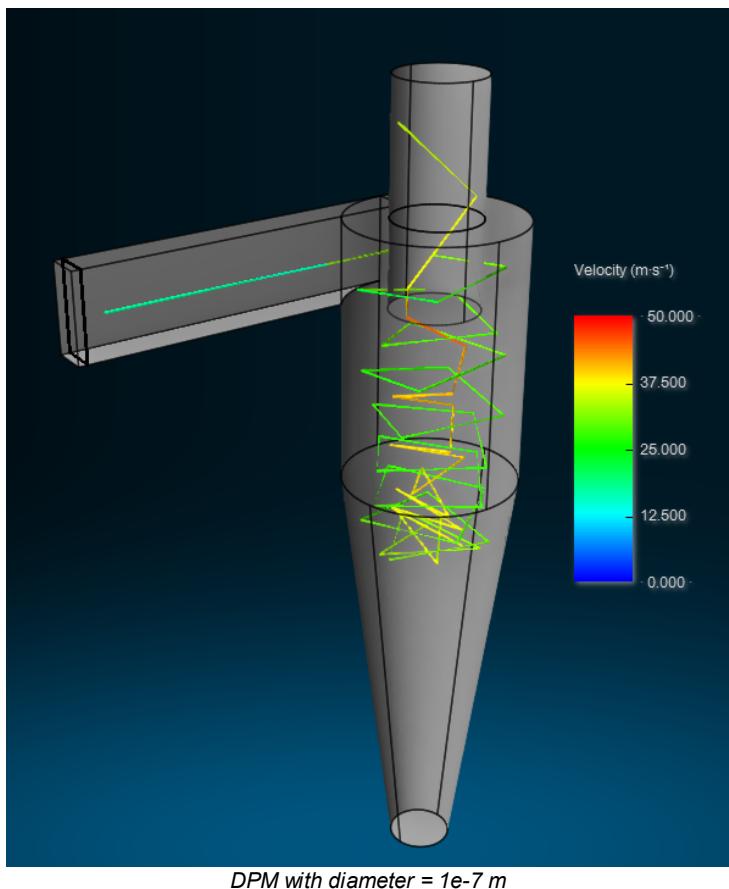


### 3.3.2 Other cases

Taking the previous case as basis:

- 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.

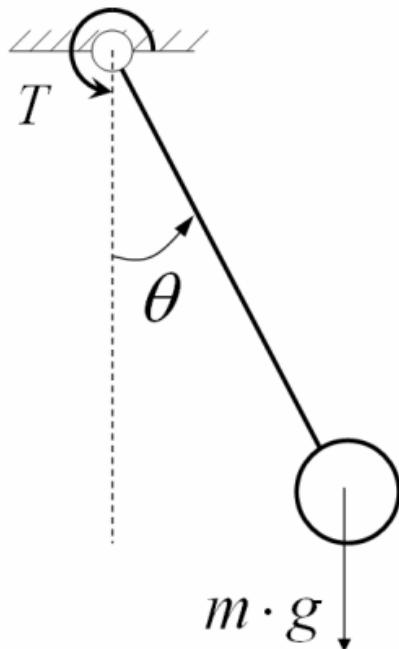
### Step 3: Stream tracers



- (b) Vary the **Restitution coefficient**: **1** and 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) Vary the **Restitution coefficient** to **0** and recompute. The collision is perfectly inelastic (the particle does not bounce off the wall and sticks to it)
- (d) Vary the **Number of tracers** to **4** and the **Standard deviation** (in diameter size) to **0.005**. Now four tracers will be created at the source surface and at the initial time. The particle sizes follow a normal distribution with mean value **0.001** and **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) Vary 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: OpenModelica Pendulum

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



FMI Standard works with the master-slave concept: the slaves simulate sub-problems whereas the master is responsible for both coordinating the overall simulation as well as transferring data. XFlow works just as a slave, so it can be connected to external software defined as a master.

OpenModelica is an open-source Modelica-based modeling and simulation environment intended for industrial and academic usage. It is able to work as master so the coupling with XFlow is possible.

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

- Coupling with a master software: OpenModelica



**Please note:** The version used of OpenModelica is 1.9.1 (r22929) available online.

## Contents

- [Step 1: Problem setup](#)
- [Step 2: Set FMI standard in XFlow](#)
- [Step 3: OpenModelica setup](#)
- [Step 4: Run](#)
- [Step 5: Post-processing](#)

## Step 1: Problem setup - XFlow

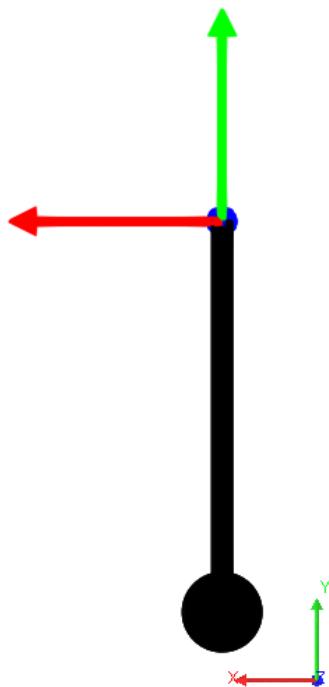
### 1.1 Import pendulum

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 **Apply to all**.

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

Environment   Materials   **Geometry**   Simulation   Post-Processing

Entities

- Shape: Pendulum
  - Behaviour: Fixed
  - Position: ( 0 , 0 , 0 ) m
  - Orientation: ( 0 , 0 , 0 ) deg
- Boundary conditions: Wall
  - Wall model: Automatic
  - Wall roughness: 0 m
  - Virtual moving wall: Off

Cables

Project Tree &gt; Geometry &gt; Entities

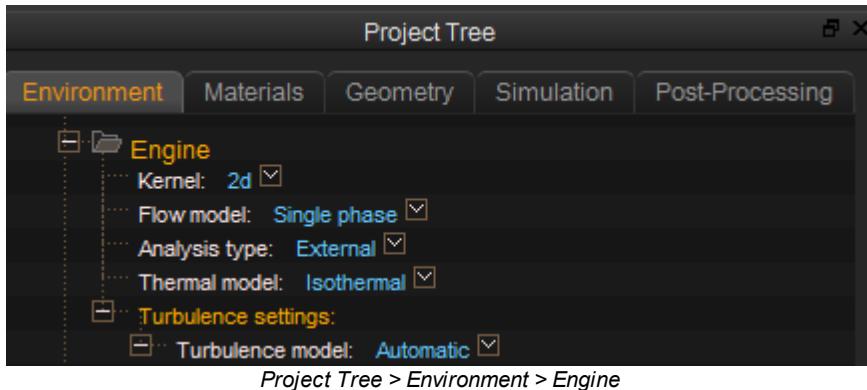
## 1.2 Engine settings

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:

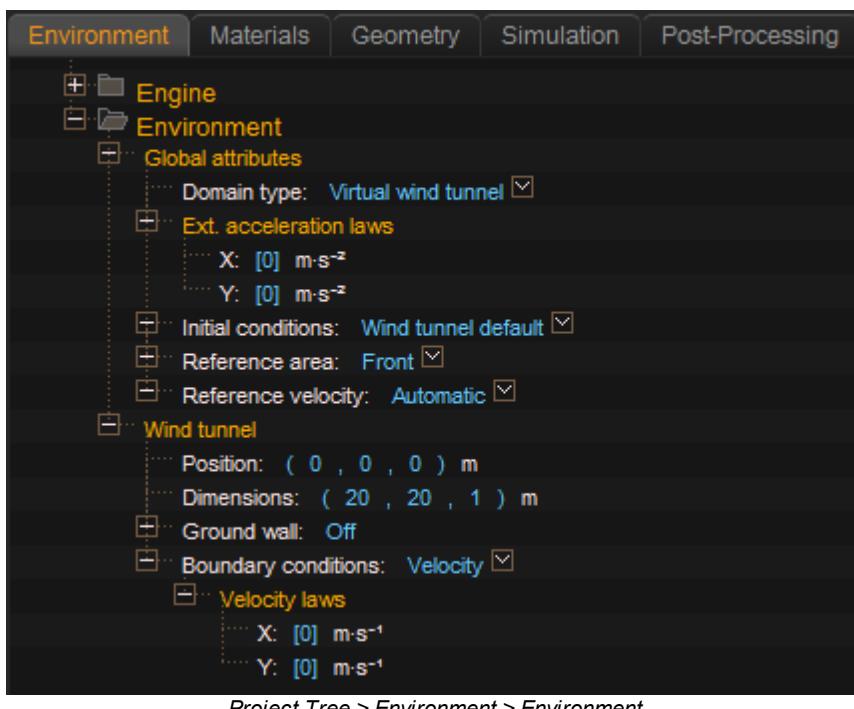
- Turbulence model: **Automatic**



### 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 (gravity is included in the OpenModelica pendulum equation).
- (c) Global attributes > Initial conditions: **Wind tunnel default**
- (d) Global attributes > Reference area: **Front**
- (e) Global attributes > Reference velocity: **Automatic**
- (f) Wind tunnel > Position: leave it to 0, 0, 0.
- (g) Wind tunnel > Dimensions: 20 metres length (X), 20 metres height (Y) and 1 metre 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.



Project Tree &gt; Environment &gt; Environment

## 1.4 Materials settings

By default in single phase the fluid material is called *Material 1*, which is initialised with the air thermophysical properties.



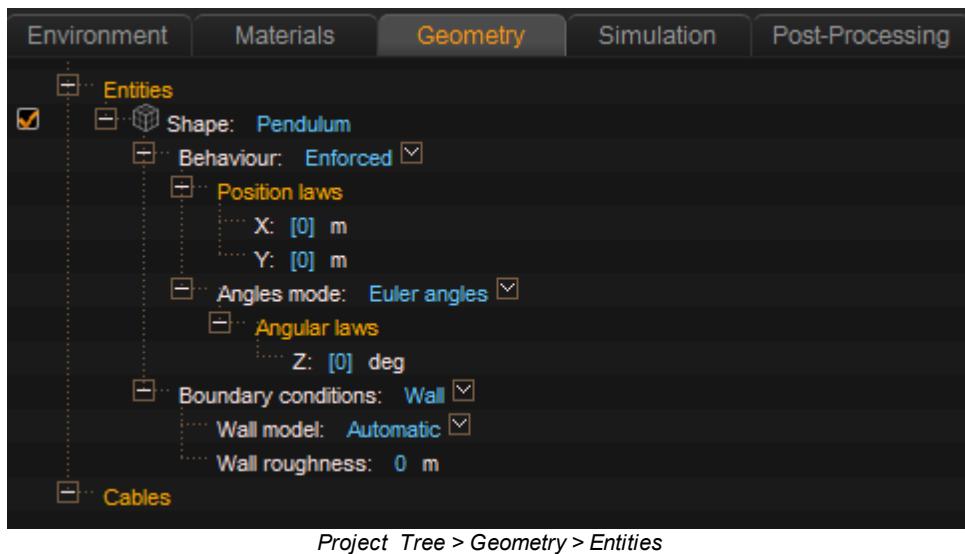
Project Tree &gt; Materials

## 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 orientation.
- [Project Tree > Geometry > Entities > Shape: Pendulum > Boundary conditions: Wall](#), with Automatic wall model and zero roughness

**⚠ Please note:** Currently XFlow only is able to simulate geometries with **Enforced** behaviour.



## 2.5 Simulation settings

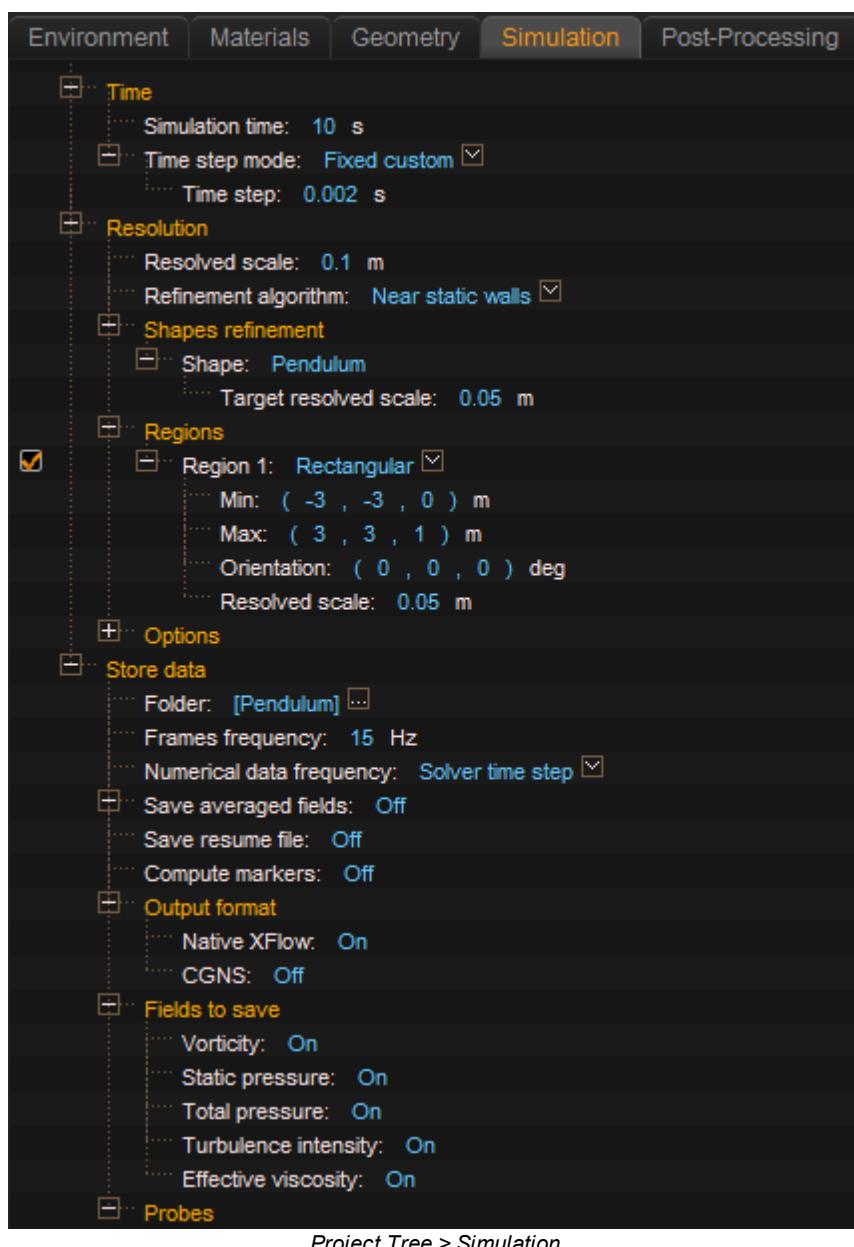
Configure the section Project Tree > Simulation > as follows:

- (a) Time > Simulation time: **10 s**
- (b) Time > Time step mode: **Fixed Custom**
- (c) Time > Time step mode > Time step: **0.002 s**
- (d) Resolution > Resolved scale: **0.1 m**
- (e) Resolution > Refinement algorithm: **Near static walls**
- (f) Resolution > Shapes refinement > Shape: **Pendulum** > Target resolved scale: **0.05 m**
- (g) Resolution > Region: **Rectangular** > Min: (-3,-3,0) ; Max: (3,3,1) ; Resolved scale: **0.05 m**
- (h) Store data > Folder: Pendulum. This is the name of the folder where the numerical data will be stored.
- (i) Store data > Frames frequency: **15 Hz** (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

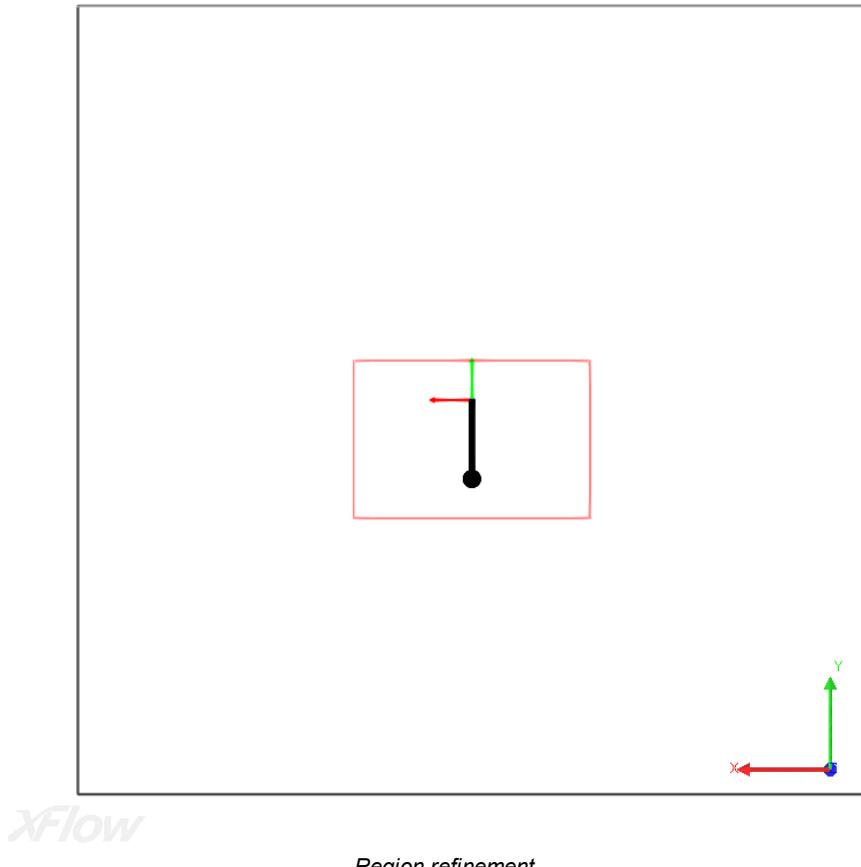
you can switch it to On, however this consumes more hard disk space.

(m) **Store data > Compute makers:** Off

(n) **Store data > Fields to save:** Leave **On** all the fields, which means that all the flow fields will be saved on the hard disc.



## Step 1: Problem setup - XFlow



*XFlow*

*Region refinement*

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

**⚠ Please note:** Before executing the **Export to FMI standard** command it is necessary to generate the binary files to create the folder where XFlow saves the FMI standard files.

**Main menu > Simulation data > Generate binary files**

This last command is available only in **Expert** or **Labs** mode.

Export FMI Standard file which contains the input/output variables to communicate with OpenModelica :

**Main menu > Options > Export to FMI standard**

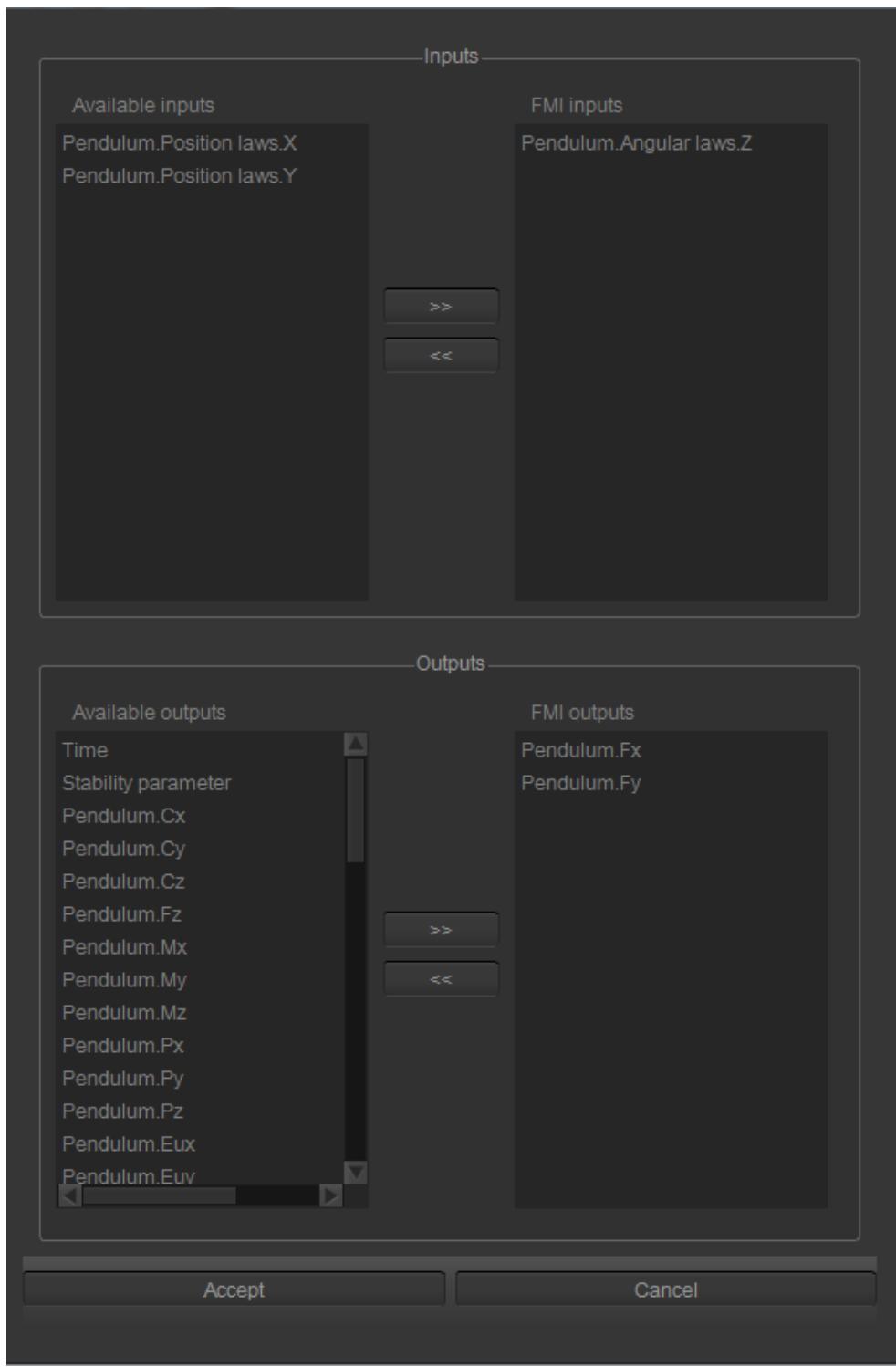
The Pendulum modeling needs an angular position as input and the internal forces (Fx and Fy) as output.

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 positions as FMI inputs, and any variable stored in the numerical data as FMI outputs.

## Step 2: Set FMI standard in XFlow



## Step 3: Problem setup - OpenModelica

The user needs to install OpenModelica in its computer.

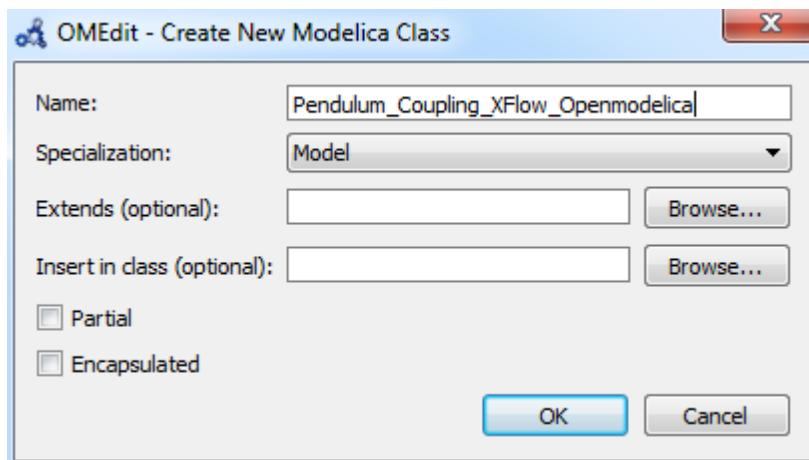


**Please note:** The version used of OpenModelica is 1.9.1 (r22929) available online.

### 3.1 Create new Modelica Class

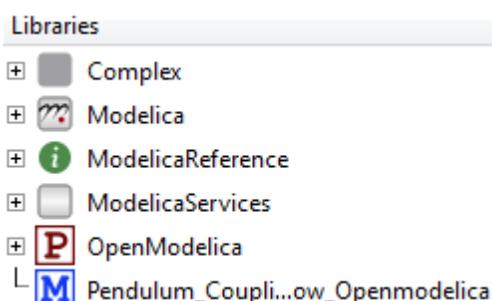
Execute OpenModelica through the direct link in your desktop or by double-clicking the `OMEdit.exe` file located in the installation folder. The application displays the *Graphical User Interface*.

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*

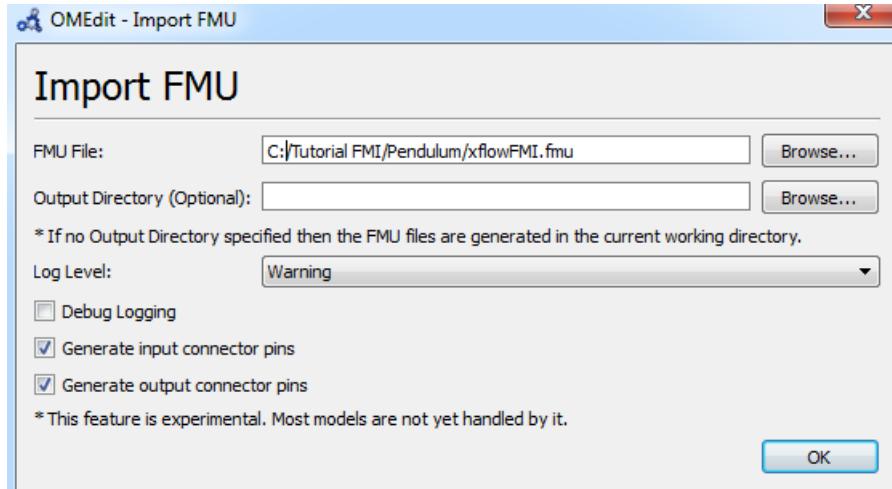
In the Libraries tree (left) a new Modelica Class has appeared, called "Pendulum\_Coupling\_XFlow\_Openmodelica".



### 3.2 Import FMU

### Step 3: Problem setup - OpenModelica

Import the file `xflowFMI.fmu` created by XFlow to OpenModelica **Main menu > FMI > Import FMU**.



### 3.3 Set pendulum equation

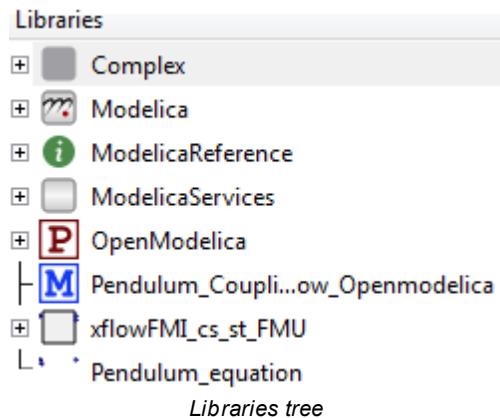
In this simulation the Pendulum equation is going to be introduced as an extern library. It contains the following code:

```
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
6 Modelica.Blocks.Interfaces.RealOutput theta annotation(Placement(transformation(extent = {{100, 60}, {120, 80}}));
7 Modelica.Blocks.Interfaces.RealInput extFx annotation(Placement(transformation(extent = {{-120, 60}, {-100, 80}}));
8 Modelica.Blocks.Interfaces.RealInput extFy annotation(Placement(transformation(extent = {{-100, 35}, {-120, 55}}));
9
10 output Real thetaRad;
11 //theta_initialization of 30 degrees from equilibrium.
12 Real y(start = -1.905255888, fixed = true);
13 Real x(start = 1.1, fixed = true);
14 output Real F;
15 output Real vx, vy;
16 equation
17  //X axis
18 m * der(vx) = (-x / L * F) + extFx;
19  //Y axis
20 m * der(vy) = (-y / L * F) - m * g + extFy;
21  der(x) = vx;
22  der(y) = vy;
23 x ^ 2 + y ^ 2 = L ^ 2;
24 thetaRad = atan2(x, -y);
25 //Angular position in degrees
26 theta = Modelica.SIunits.Conversions.to_deg(thetaRad);
27 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));
28 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:



### 3.4 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:

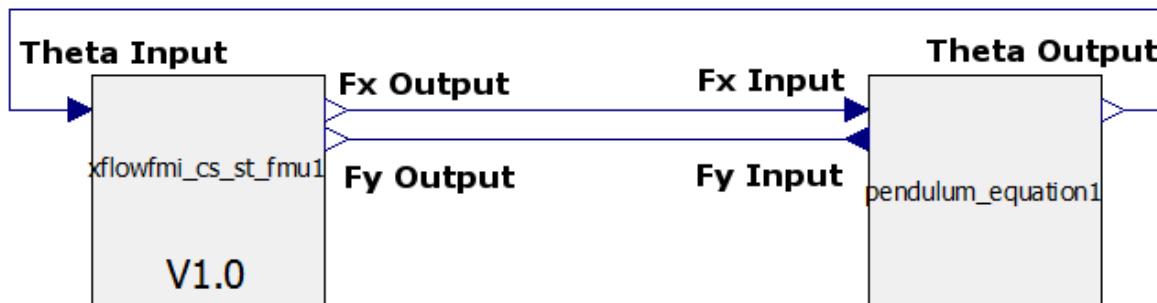


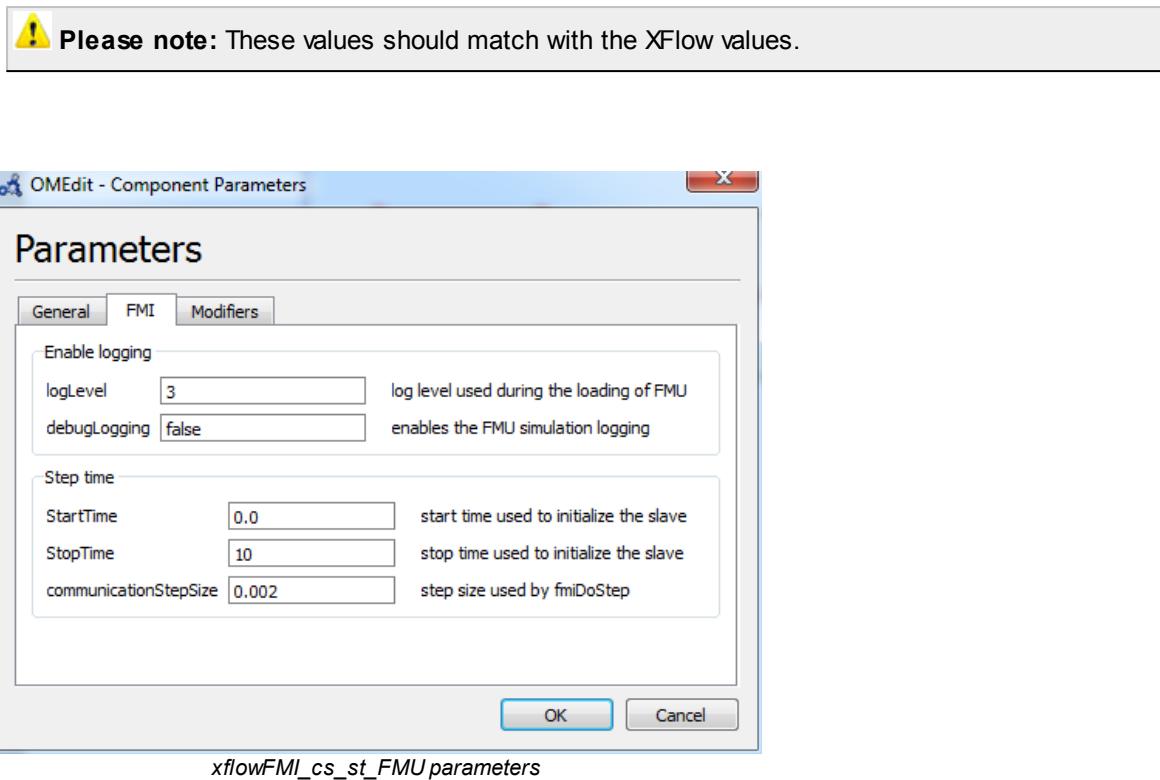
Diagram OpenModelica

### 3.5 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 **Paramenters > FMI** as follows:

- StartTime:** 0.0 s
- StopTime:** 10 s.
- communicationStepSize:** 0.002 s.

### Step 3: Problem setup - OpenModelica



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.

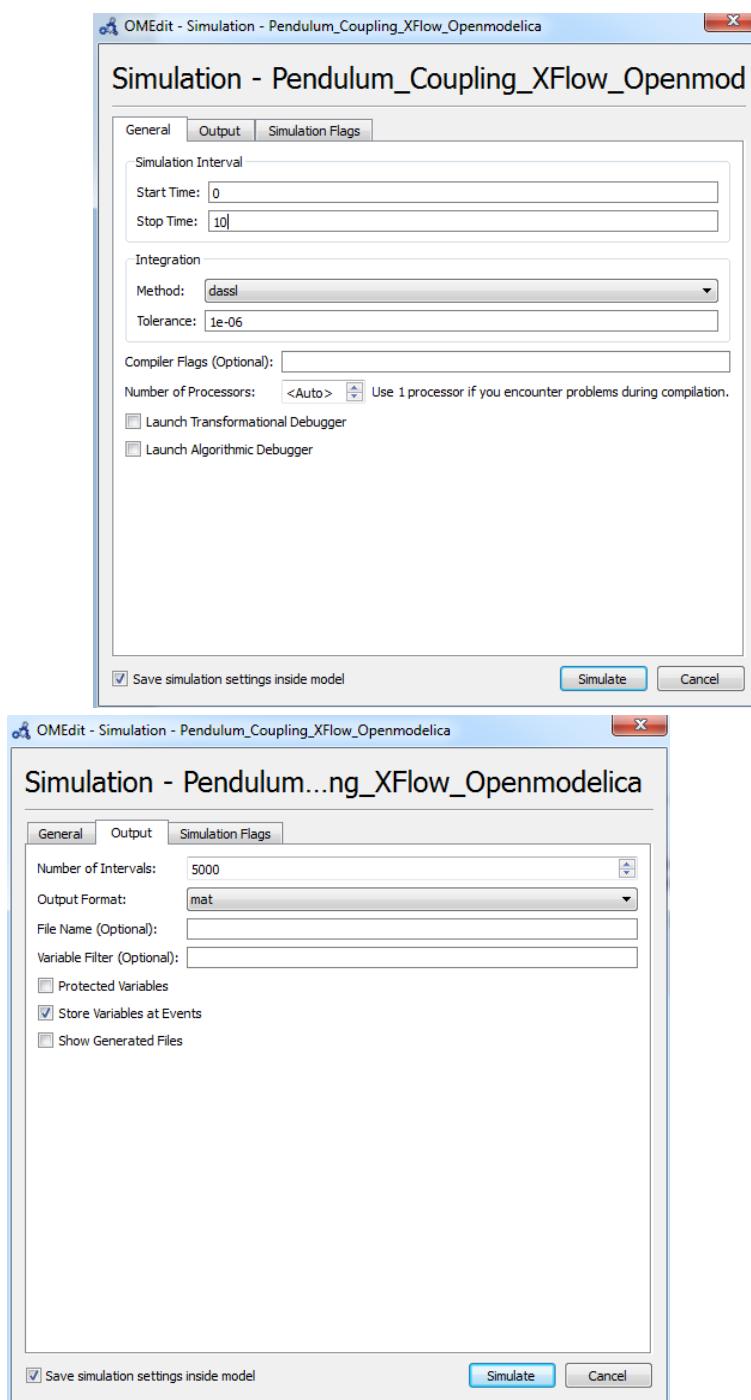
Configure the section **Simulation > Output** as follows:

- (f) **Number of iterations:** 5000

which is the result of the current operation(**StopTime - StartTime**) / **communicationStepSize**.

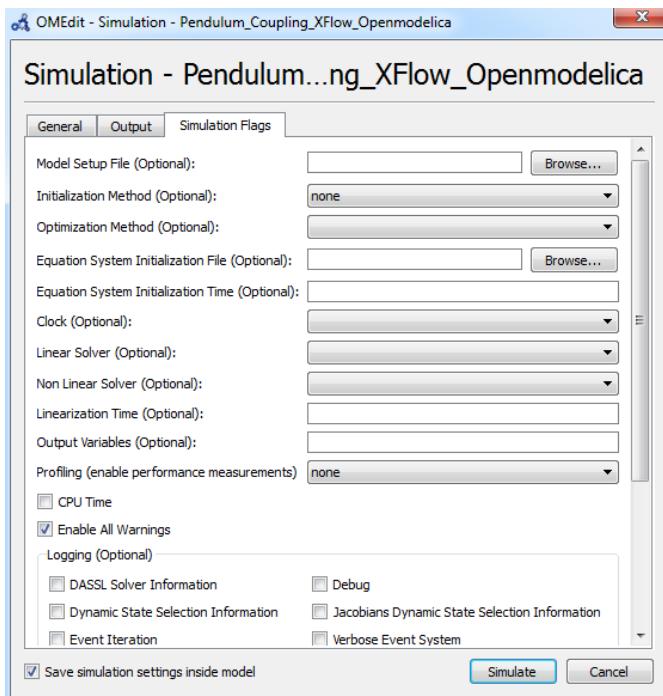
Configure the section **Simulation > Simulation Flags** as follows:

- (g) **Initialization Method:** none.



Pendulum\_equation simulation parameters

### Step 3: Problem setup - OpenModelica



Leave the remaining parameters to the default value.

## Step 4: Execution of Co-simulation

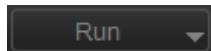
### 4.1 Save project and generate binary files

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

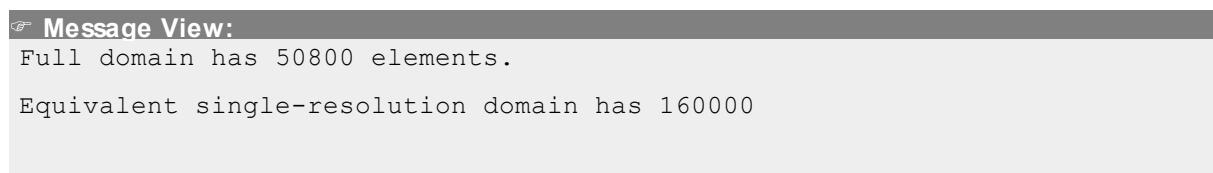
XFlow project files have the extension `.xfp`.

If user has change some simulation parameter in XFlow has to generate binary files: **Main menu > Simulation data > Generate binary files**

### 4.2 Start FMU computation in XFlow

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

XFlow waits until the master start the computations.



```

Equivalent single-resolution domain size is ( 400 x 400 x 1 )

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

Coarsest resolved length: 0.1

-----
-- 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.

Thermodynamic speed of sound: 340.112
Reference area: 0.22 m^2
Reference velocity: 1 m/s
Time step (level 0): 0.004 s

-----
-- Waiting FMU initialization...

```

### 4.3 Start simulation in OpenModelica

Run the simulation in the "Pendulum\_equation" simulation parameter 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.

### 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...

```

## Step 4: Execution of Co-simulation

```
[[Data file]] 0 done!!! | Frame wall clock time[0]s | Overall wall clock
time[0]s | Num elements[50800]
Sim. time [4.000000e-003]s | Stability param. [1.734903e-004] | Wall clock
time [2.810000e-001]s
Sim. time [8.000000e-003]s | Stability param. [1.734903e-004] | Wall clock
time [4.060000e-001]s
Sim. time [1.200000e-002]s | Stability param. [2.347426e-003] | Wall clock
time [4.520000e-001]s
Sim. time [1.600000e-002]s | Stability param. [7.856966e-003] | Wall clock
time [3.740000e-001]s
Sim. time [2.000000e-002]s | Stability param. [1.385617e-002] | Wall clock
time [4.370000e-001]s
Sim. time [2.400000e-002]s | Stability param. [2.012526e-002] | Wall clock
time [4.840000e-001]s
Sim. time [2.800000e-002]s | Stability param. [2.625810e-002] | Wall clock
time [3.900000e-001]s
Sim. time [3.200000e-002]s | Stability param. [3.169038e-002] | Wall clock
time [4.210000e-001]s
Sim. time [3.600000e-002]s | Stability param. [3.618427e-002] | Wall clock
time [4.370000e-001]s
Sim. time [4.000000e-002]s | Stability param. [3.939046e-002] | Wall clock
time [4.840000e-001]s
Saving data...
[[Data file]] 1 done!!! | Frame wall clock time[4.213000e+000]s | Overall
wall clock time[4.213000e+000]s | Num elements[50800]
Sim. time [4.400000e-002]s | Stability param. [4.137910e-002] | Wall clock
time [5.930000e-001]s
Sim. time [4.800000e-002]s | Stability param. [4.209257e-002] | Wall clock
time [3.590000e-001]s
Sim. time [5.200000e-002]s | Stability param. [4.167101e-002] | Wall clock
time [4.520000e-001]s
Sim. time [5.600000e-002]s | Stability param. [4.090447e-002] | Wall clock
time [4.680000e-001]s
...
...
```

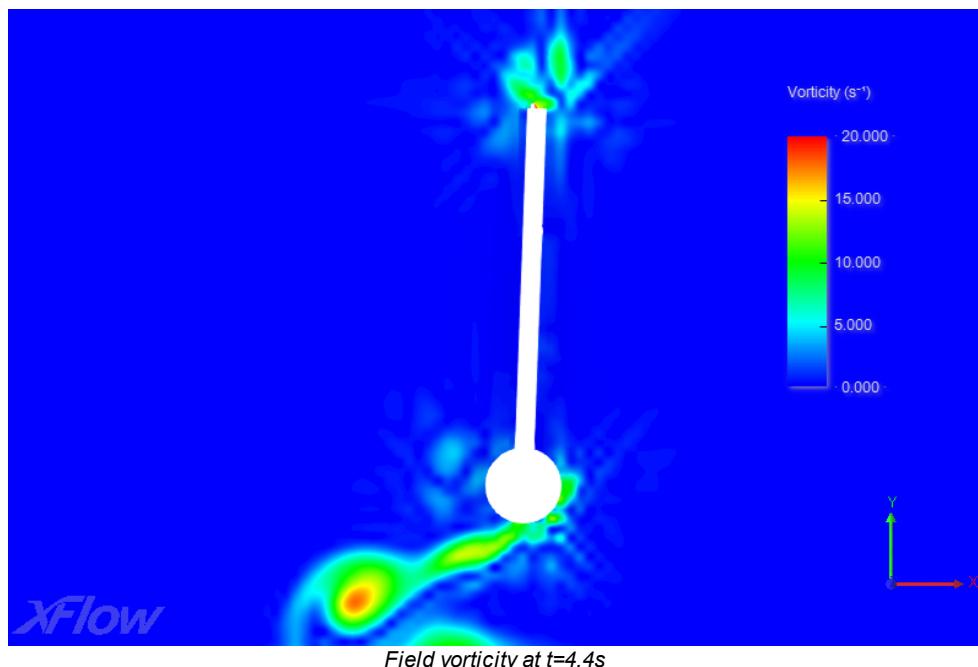
**⚠ Please note:** The stop time `1.79769e+308` in the log is given by OpenModelica ignoring the value fixed in the Simulation parameters. The effect that this behavior produces is that XFlow's simulation could get to the last frame but do not stop, it does not affect the simulation's results.

## Step 5: Post-processing

The post-processing is possible in XFlow completely managed from the **Post-Processing** section of the Project Tree and OpenModelica allows to do a similar Post-processing adding the calculated variables by the Pendulum equation.

### 5.1 XFlow post-processing

Create a cutting plane to visualise the vorticity field which shows the vortex detached by Pendulum movement. To do so, please go to **Project Tree > Post-processing > Cutting planes**:

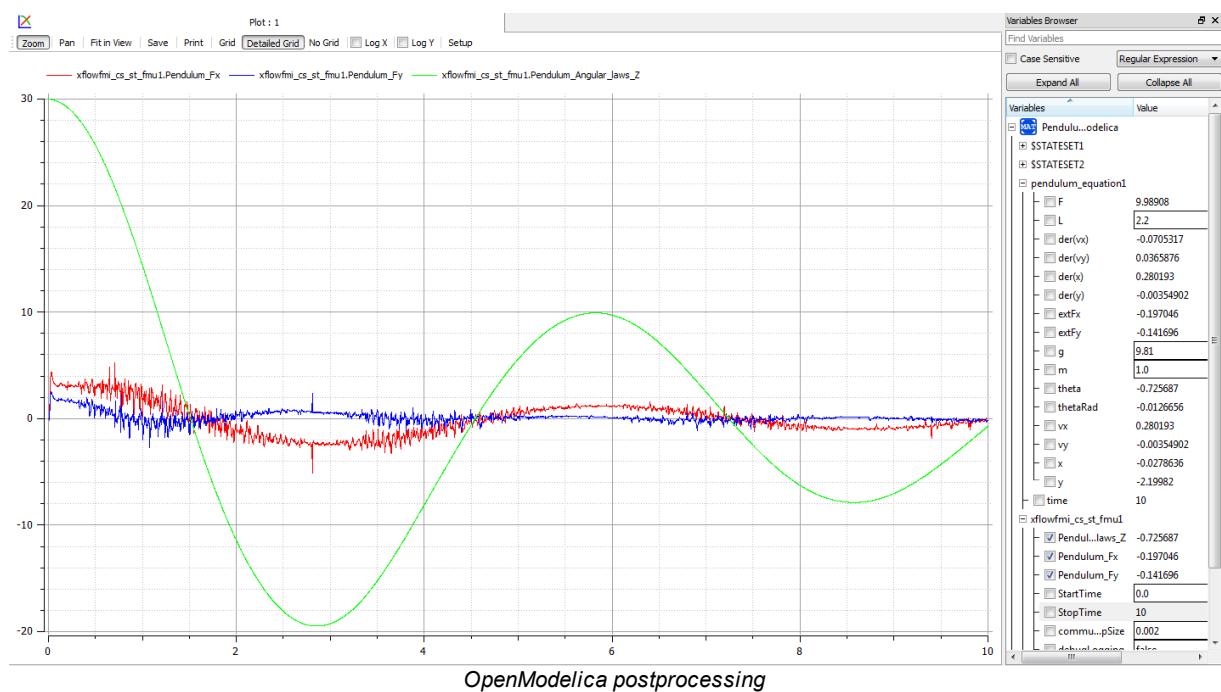


### 5.2 OpenModelica post-processing

In OpenModelica will appear a post processing window once the simulation ends. The user have access the variables of both XFlow and OpenModelica in the Variable Browser (right).

Activate the XFlow variables `Fx`, `Fy` and `Angular_laws_Z` in the Variable Browser. It can be observed how the sign of the forces changes when the Pedulum passes over the equilibrium position.

## Step 5: Post-processing



# Tutorial 12 - MSC Nastran co-simulation: Turek Hron

This tutorial illustrates how to setup and run a XFlow / MSC Nastran coupled simulation. The specific case here illustrated is the Turek-Hron FSI (Fluid-Structure Interaction) benchmark, available in literature [*"Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow"* in Lecture Notes in Computational Science and Engineering, Turek Stefan and Hron Jaroslav, Springer Berlin Heidelberg, 2006, pp. 371-385].

This tutorial shows how to:

- Setup a MSC Nastran simulation in SimXpert to 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:** Check the OpenFSI installation in your computer. Please see in the "XFlow 2014 User Guide" section **Co-simulation > MSC Nastran > XFlow OpenFSI Service Installation**.

## Contents

- [Step 1: Problem setup - SimXpert](#)
- [Step 2: Post-process - XFlow](#)
- [Step 3: Execution of Co-simulation](#)
- [Step 4: Post-processing](#)

## Step 1: Problem setup - SimXpert

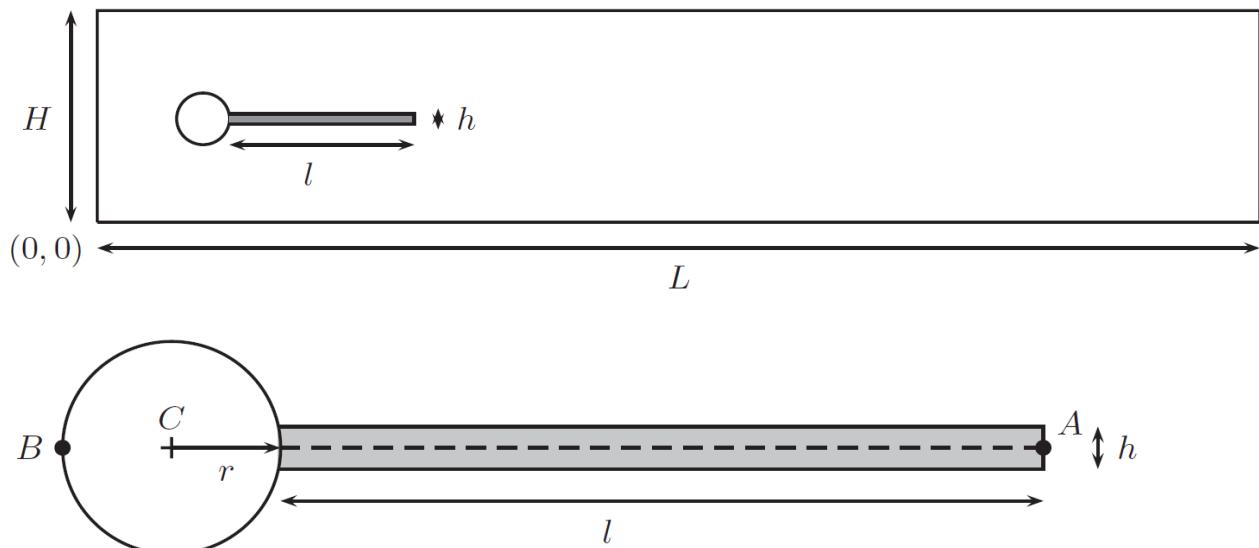
The first step of the tutorial is the setup of the MSC Nastran structural analysis. The MSC SimXpert pre-

## Step 1: Problem setup - SimXpert

processing software is here used to setup the case and generate the MSC Nastran analysis file (.bdf).

### 1.1 Definition of the Setup Geometry

The Turek Hron benchmark consists of a solid cylinder and an attached elastic bar submerged in a channel flow, as illustrated in the following figure. The characteristics of the solid and fluid materials, as well as the inlet velocity, are chosen so that self-induced oscillations in the fluid and the deformable part of the structure are obtained. In this analysis the fluid is considered to be incompressible and in the laminar regime.



*Solid cylinder and flexible bar setup*

The dimension of the external domain are:

$$L = 2.5 \text{ [m]}$$

$$H = 0.41 \text{ [m]}$$

The cylinder and bar geometries are defined by the following geometrical quantities:

$$C = (0.2, 0.2) \text{ [m]}$$

$$r = 0.05 \text{ [m]}$$

$$l = 0.35 \text{ [m]}$$

$h = 0.02 \text{ [m]}$

$$A(0) = (0.6, 0.2)$$

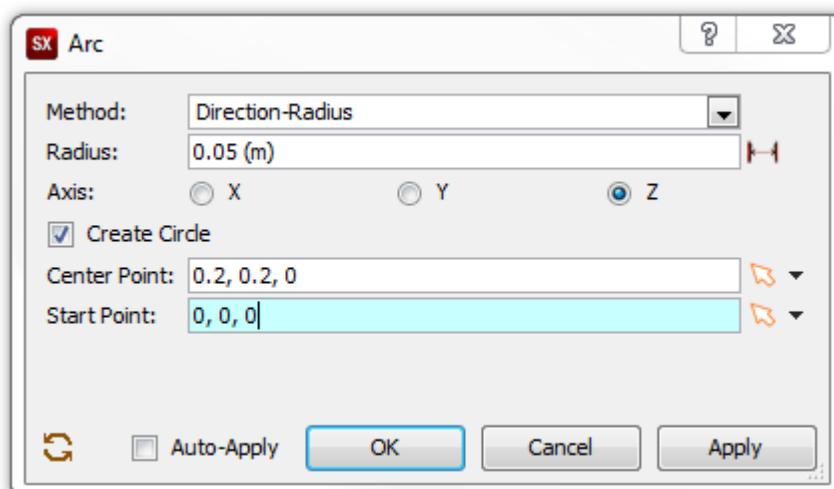
$$B = (0.15, 0.2)$$

## 1.2 Creation of the Geometry in SimXpert

- a) Set the unit to [m] in **Tools > Options > Units Manager > Length: m**

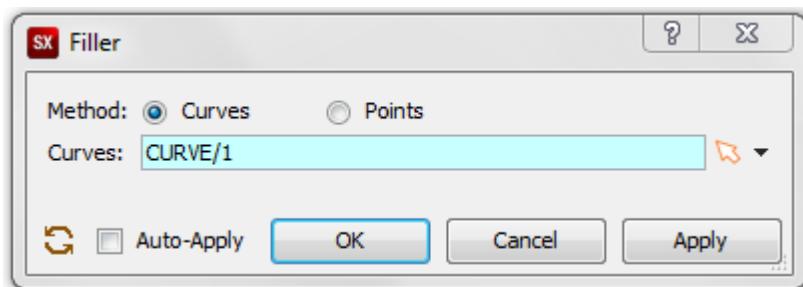
In the **Geometry** tab:

- b) Create the cylinder geometry with the Arc command  and using the settings:



*Creation of the cylinder perimeter*

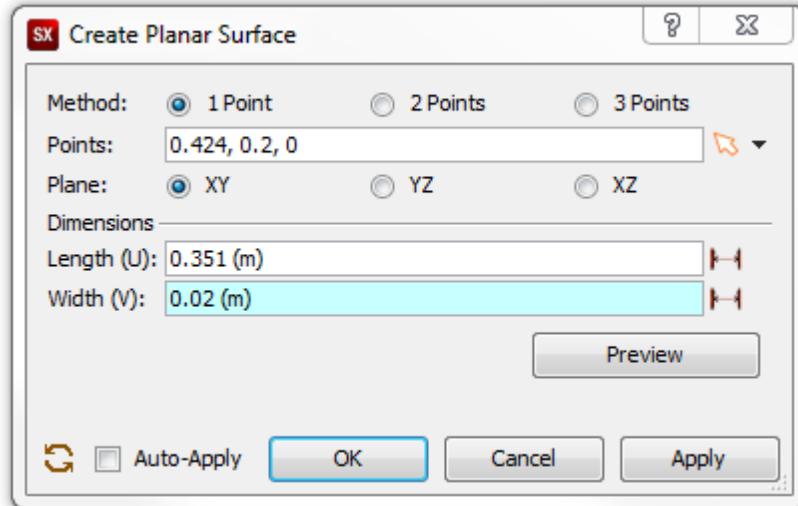
- c) Create a surface of the Cylinder. Select the Cylinder perimeter and use the filler command  :



*Creation of the cylinder surface*

- d) Create the Bar geometry using the planar command  and the settings:

## Step 1: Problem setup - SimXpert



*Creation of the bar surface*

The resulting geometry should look like:

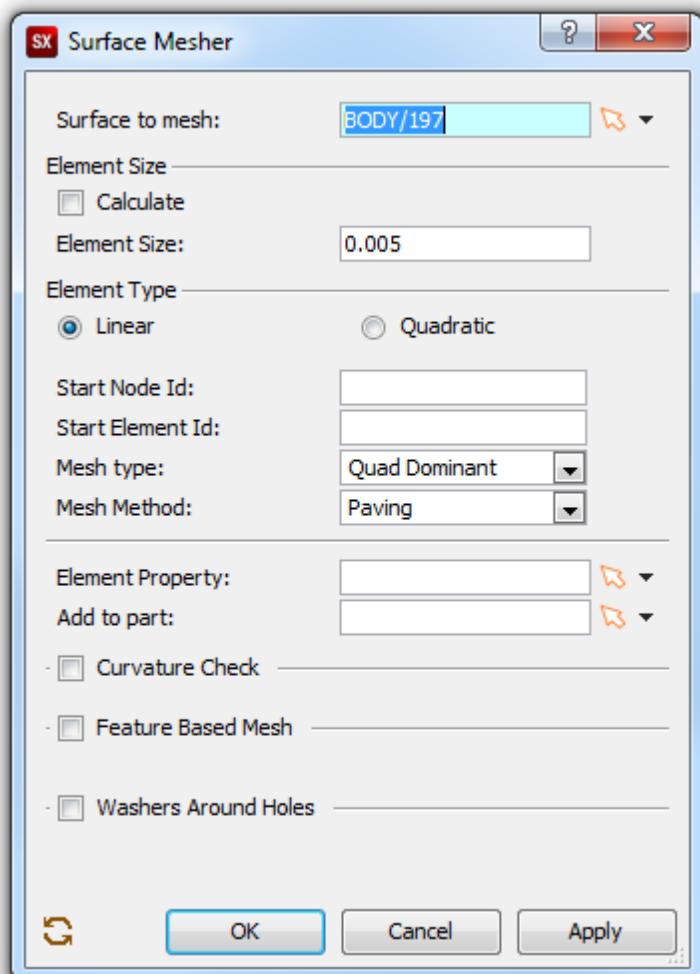


*Cylinder and Bar geometries*

### 1.3 Generation of the Structural Mesh

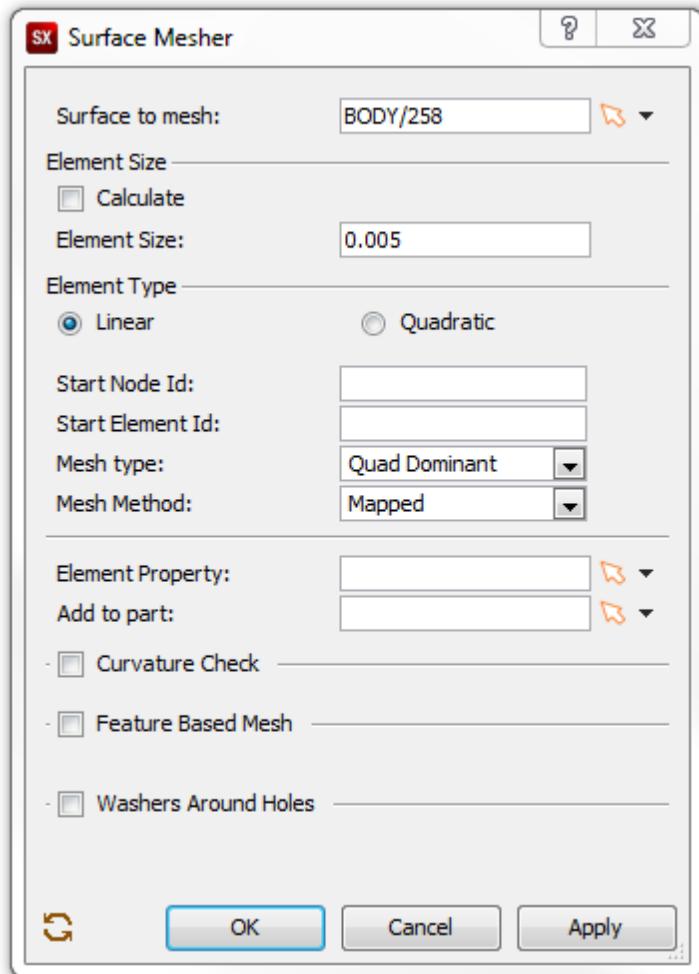
In the **Meshing** tab:

- Generate a mesh of the Cylinder and Bar surfaces using the Surface command and the settings shown below:



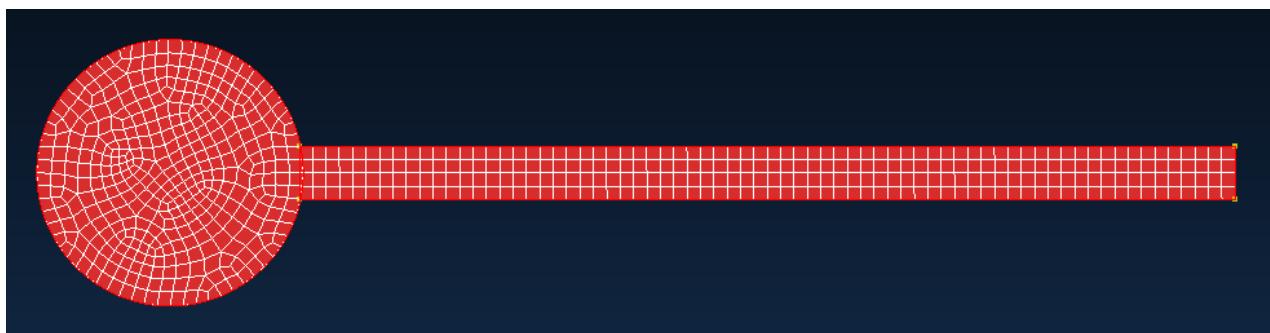
Cylinder Surface

## Step 1: Problem setup - SimXpert

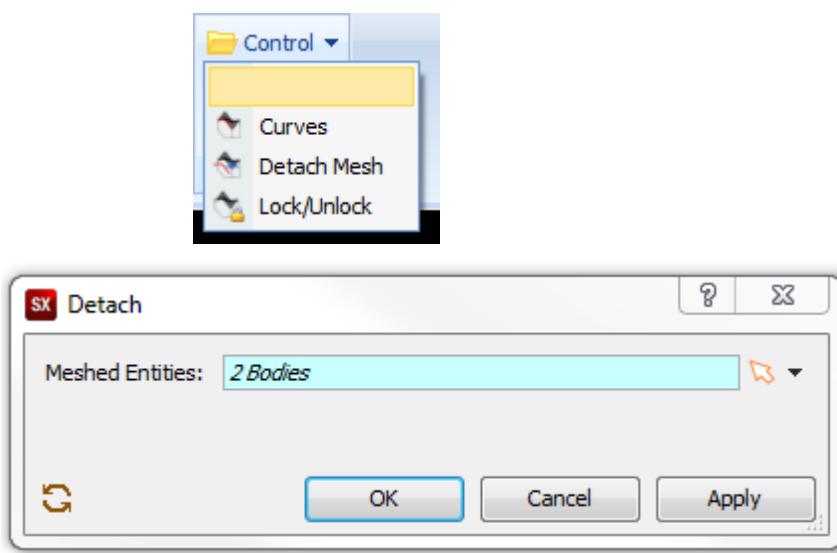


Bar Surface

The resulting mesh should look like:

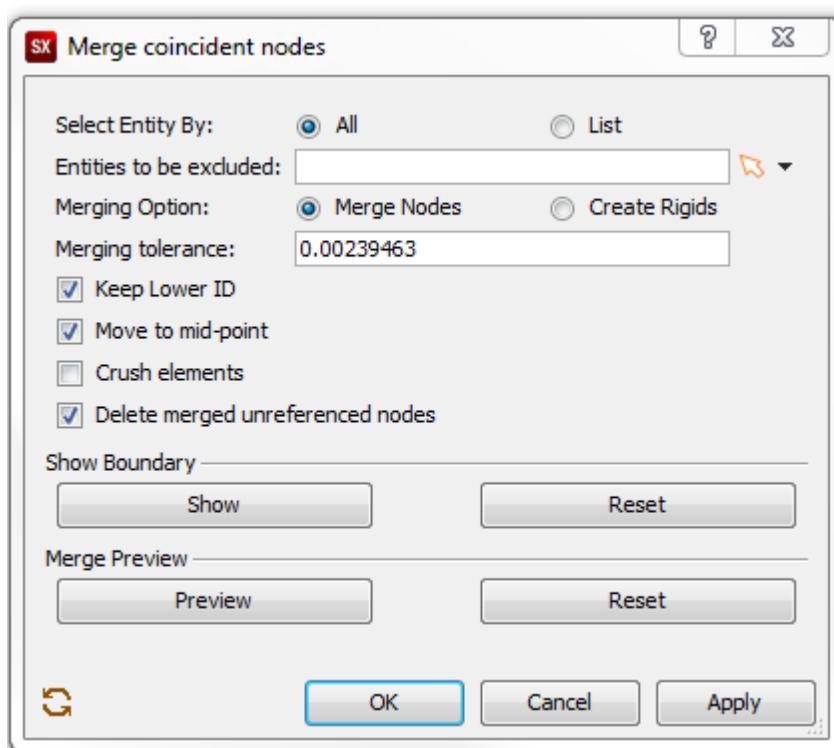


- b) Detach the created mesh from the geometry, using the command Detach Mesh and selecting the two geometries (Cylinder and Bar):



In the **Nodes/Elements** tab:

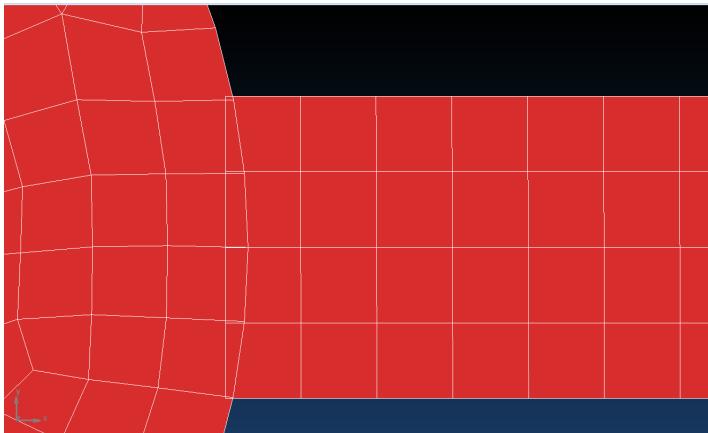
- c) Match the Cylinder and Bar Meshes using the equivalence command  , leaving the default Settings:



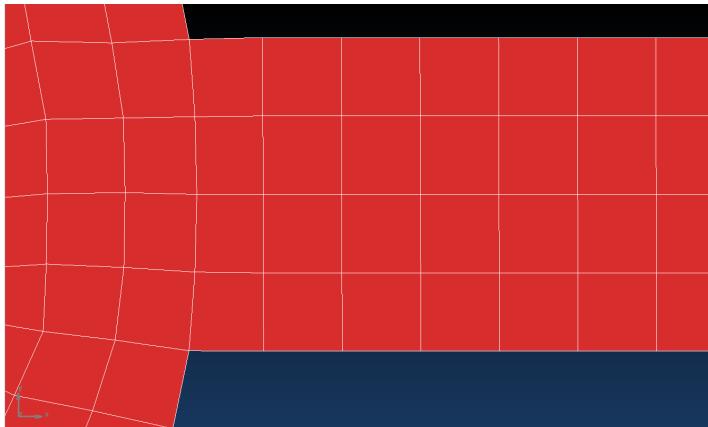
*Settings for the Equivalence of the Cylinder and Bar Meshes*

## Step 1: Problem setup - SimXpert

The mesh of the Cylinder and Bar should now match at the interface, as shown below:



*Before Equivalence*

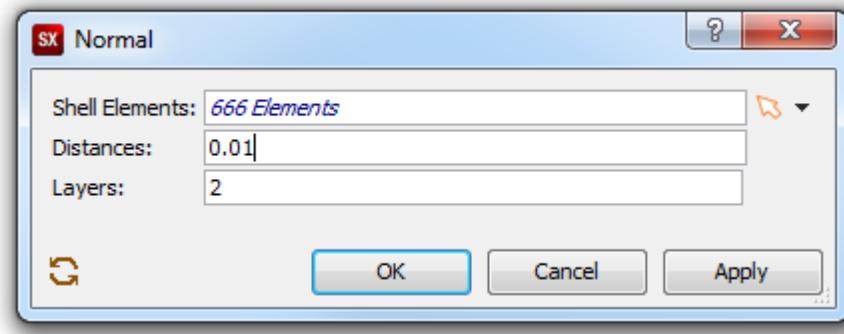


*After Equivalence*

In the **Meshing** tab:

d) Extrude the surface mesh in the z-axis to create 3D solid elements using the normal command 

Select all the 2D elements and input the values shown in the image below:



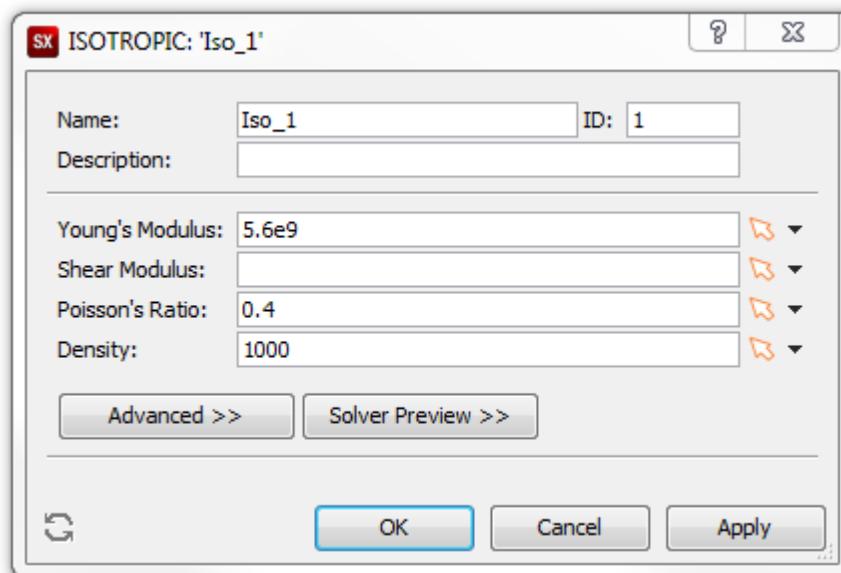
*Extruding Surface elements to generate 3D Solid ones*

- e) Create a new Part named MESH. Select only 3D solid elements as visible in the graphic viewer, (icon ). Select all elements in the GUI and move them to the newly generated MESH part by right-click **Assign Part...**
- f) Delete the original Part.

## 1.4 Definition of Material, Elements Properties and Boundary Conditions

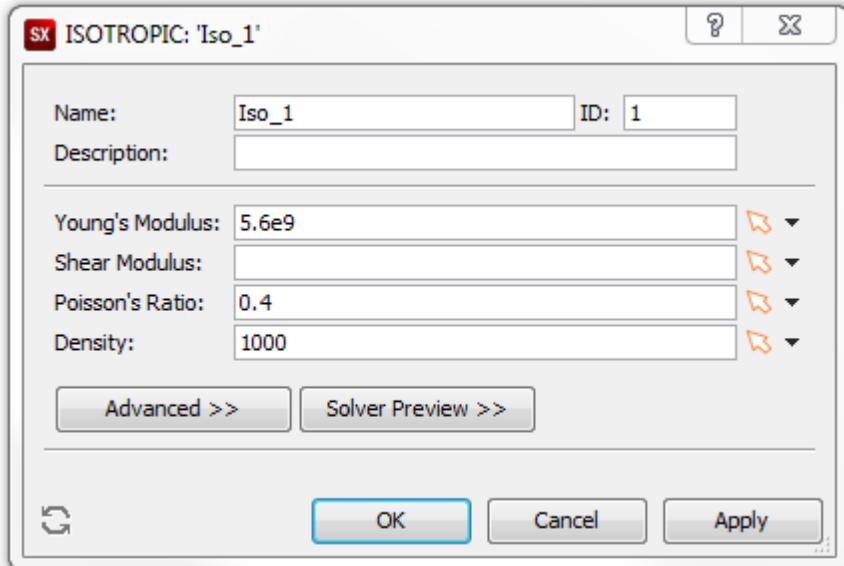
In the **Material and Properties** tab:

- a) Define an isotropic material , with the following elastic characteristics:



- b) Apply Material and Properties to the 3D Solid Elements using the command **Solid** . Select all the 3D Solid elements:

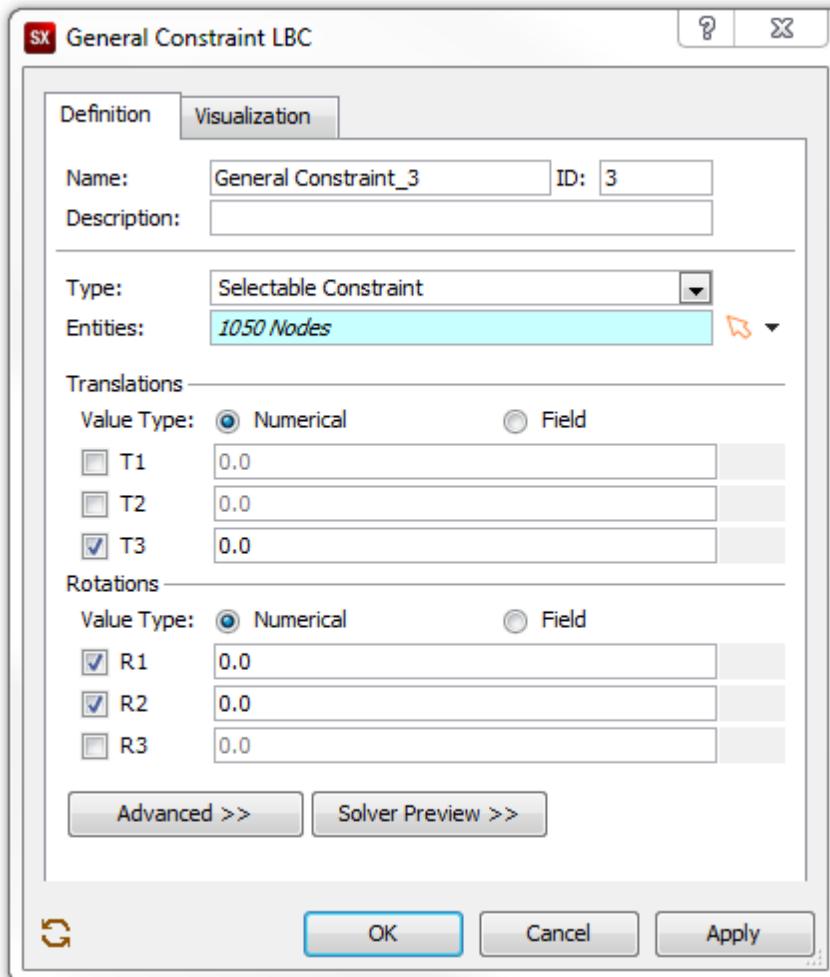
## Step 1: Problem setup - SimXpert



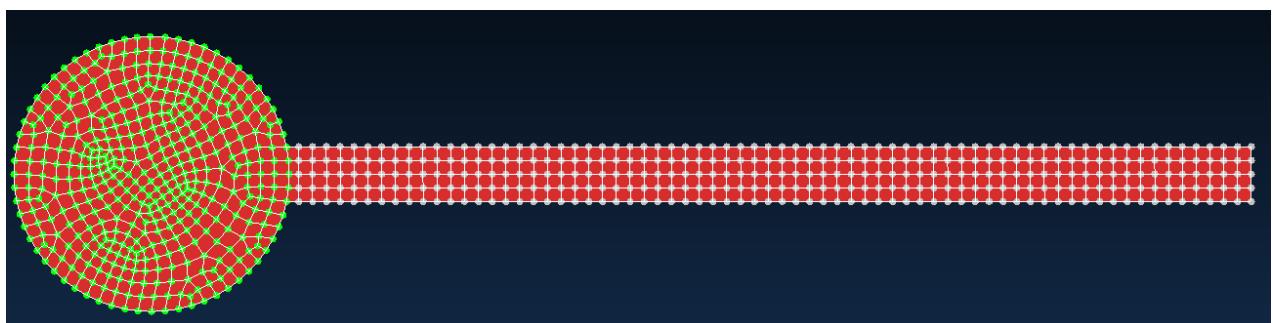
*Applying Material Properties to 3D Solid Elements*

In the **LBCs** tab:

- c) Apply Boundary Conditions. We are going to fix the cylinder's nodes, selecting them in the viewer and using the fixed BC . Also, we are allowing only the X and Y translation and Z rotation of the Bar nodes, setting a General Constraint BC .

*General Constraint BC applied on the Bar nodes*

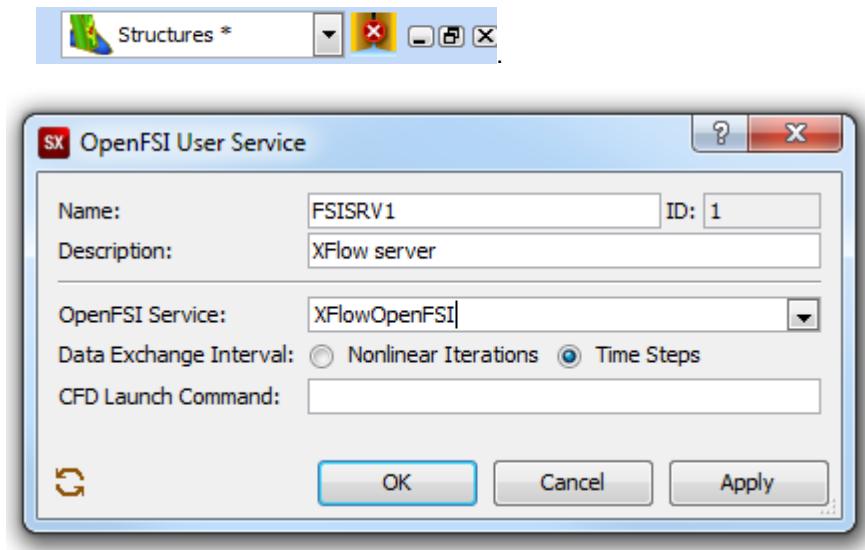
The applied Boundary Conditions should look as below, where the green nodes are Fixed and the grey ones are allowed only XY plane movements (the colors shown are not by default).

*Constraints visualization on Mesh*

## 1.5 Creation of XFlowOpenFSI Server

In the **User Services** tab:

- a) Create an OpenFSI server, under **User Services**  , using the settings below indicate. If the **User Services** tab is not visible, please make sure you are in **Structures** mode

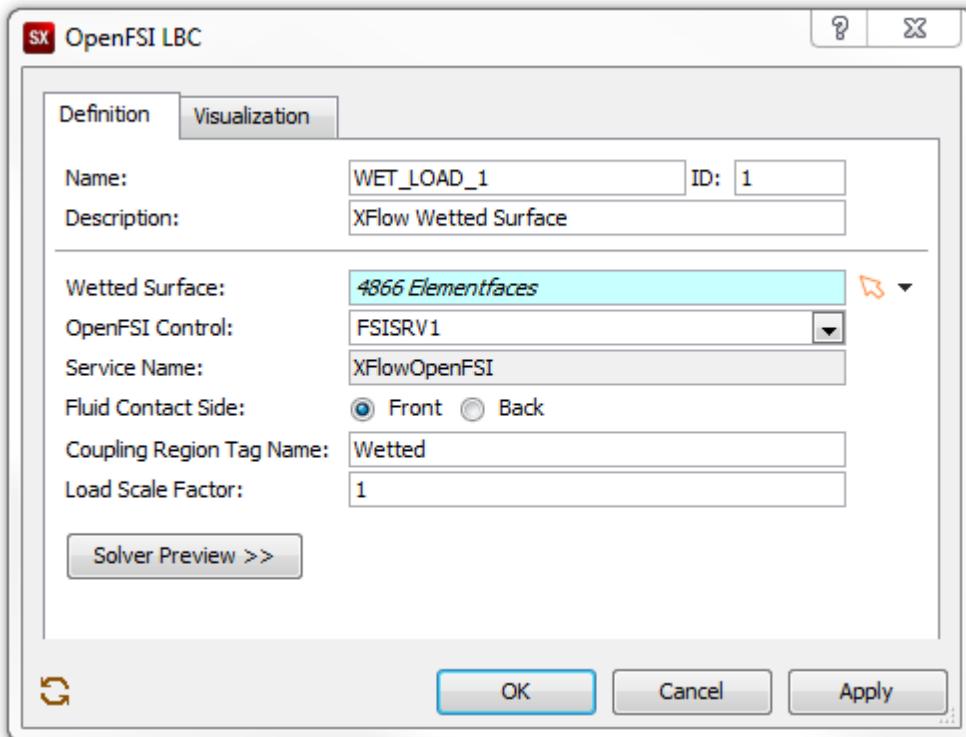


*Creation of XFlowOpenFSI Server*

In the **LBCs** tab:

- b) Create the Wetted Surface which will be exchanging the load and displacement information with XFlow.

From the LBCs tab use the OpenFSI command  :

*Definition of Wetted Surface*

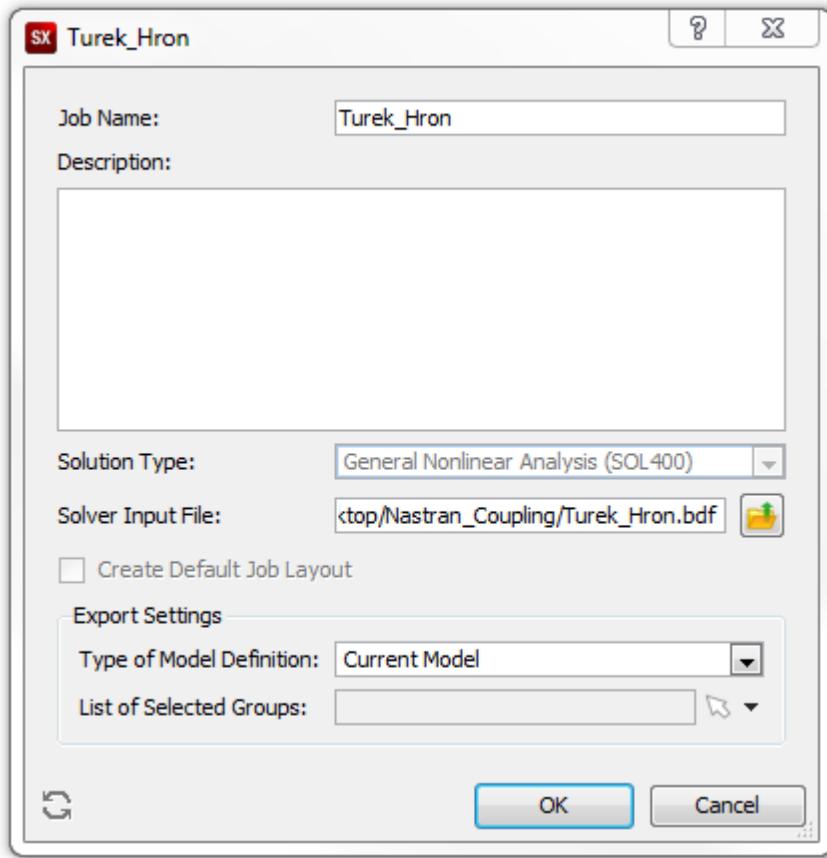
In order to select all the 3D Solid faces please select the Element Faces filter ONLY in the Wetted Surface



## 1.6 Creation of MSC Nastran job

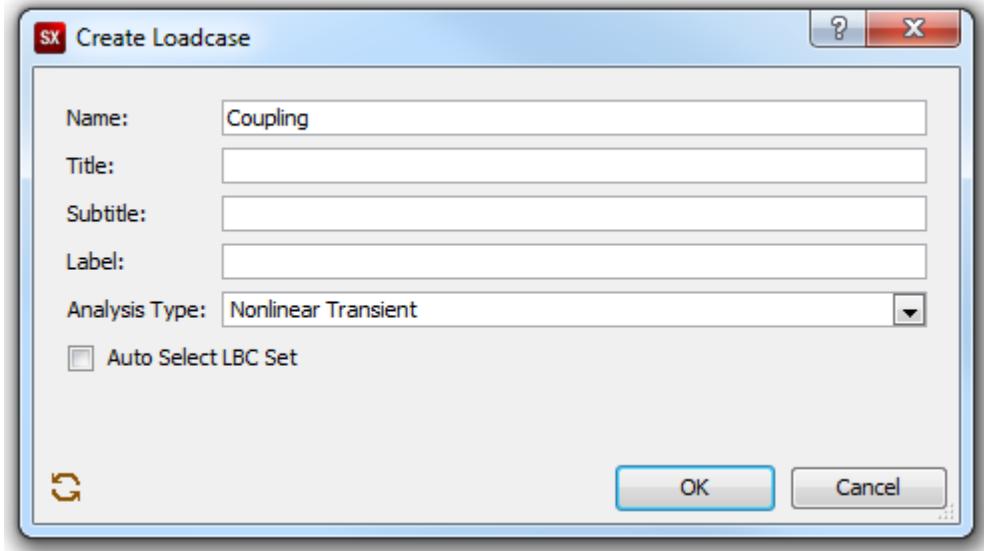
- Create a Nastran Job by right click the top branch of the project tree (FileSet) and selecting **Create new Nastran job** command. Select General Nonlinear Analysis (SOL 400) as **Solution Type** and deselect the **Create Default Job Layout** option.

## Step 1: Problem setup - SimXpert



*Create new Nastran job*

- b) In the project tree **Simulations > Turek Hron > Solver Control**, right click and select **Properties**. Under the **Analysis Option** tab select **Large Displacement and Follower Force** to model the Large Displacement Effects and Apply.
- c) Right click on **Load cases** and create a new Loadcase selecting **Create Loadcase**.



Create Loadcase

- d) Right click on **Simulations > Turek Hron > Load Cases > Coupling > Load Case Control** and set the following parameters under **Generic Control**:
- Total Time:** 20 (s)
  - Minimum Iterations for each Increment:** 6
  - Maximum Iterations for each Increment:** 25



**Please note:** The **Total Time** parameter set in MSC Nastran will override the XFlow **Simulation time**.

- e) Right click **Output Requests** and select **Nodal Output Requests > Create Displacement output request**. Leave the Default values.
- f) Right click **Output Requests** and select **Nodal Output Requests > Create Applied Load output request**. Leave the Default values.
- g) Right click in **LBC Container > Select LBC Set** and select the default boundary condition set **DefaultLbcSet**.
- h) Export the Nastran **Turek\_Hron.bdf** job from **File>Export > Nastran Model**

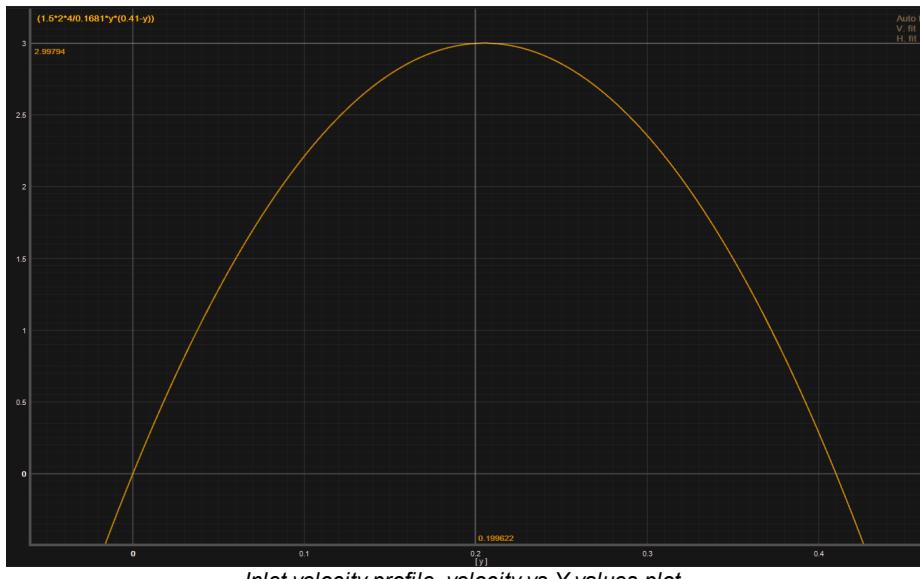
## Step 2: Problem setup - XFlow

The second step of the tutorial is the setup of the simulation in XFlow. A 2D simulation will be setup, using glycerine as fluid and setting the input parabolic velocity profile as specified in the Turek-Hron benchmark case:

## Step 2: Problem setup - XFlow

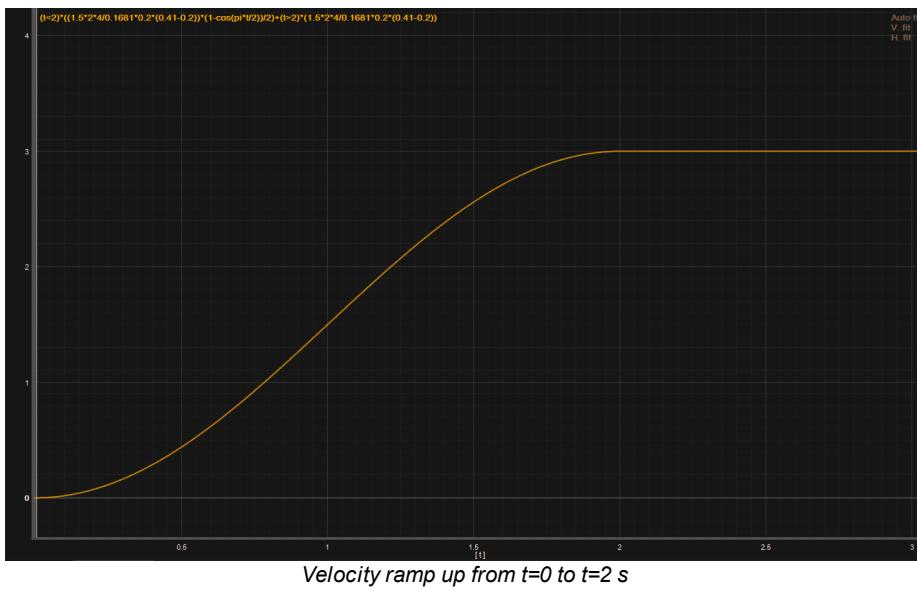
$$v^f(0, y) = 1.5 \bar{U} \frac{y(H-y)}{\left(\frac{H}{2}\right)^2}$$

The velocity profile results in zero velocity at the boundary of the domain, and a maximum value at the center, as illustrated in the picture below.



The velocity initialized to zero in the whole domain, and a ramp up function is used to slowly increase it to its maximum value:

$$v^f(t, 0, y) = \begin{cases} v^f(0, y) \frac{1-\cos(\frac{\pi}{2}t)}{2} & \text{if } t < 2.0 \\ v^f(0, y) & \text{otherwise} \end{cases}$$



## 1.1 Environment

a) Open a new XFlow window and create a new project called Turek\_Hron.xfp.

b) In Project Tree > Environment > Engine set the following parameters:

(a) Kernel: **2d**

(b) Advanced Options > Structural Analysis: **Nastran**

c) In Project Tree > Environment > Environment set the following parameters:

(a) Domain type: **Generic rectangular domain**

(b) Position: **(1.25,0.205,0.005)**

(c) Dimension: **(2.5,0.41,0.15)**

(d) X periodic: **Off**

a. -X Boundary condition: **Inlet > Velocity**

Velocity laws

X: 
$$[(t<2)((1.5*2^4/0.1681*y*(0.41-y))*(1-\cos(\pi*t/2))/2)+(t>2)(1.5*2^4/0.1681*y*(0.41-y))] \text{ ms}$$
  
-1

b. +X Boundary condition: **Outlet > Gauge pressure outlet**

(e) Y periodic: **Off**

c. -X Boundary condition: **Wall**

Wall model: **Automatic**

d. +X Boundary condition: **Wall**

Wall model: **Automatic**



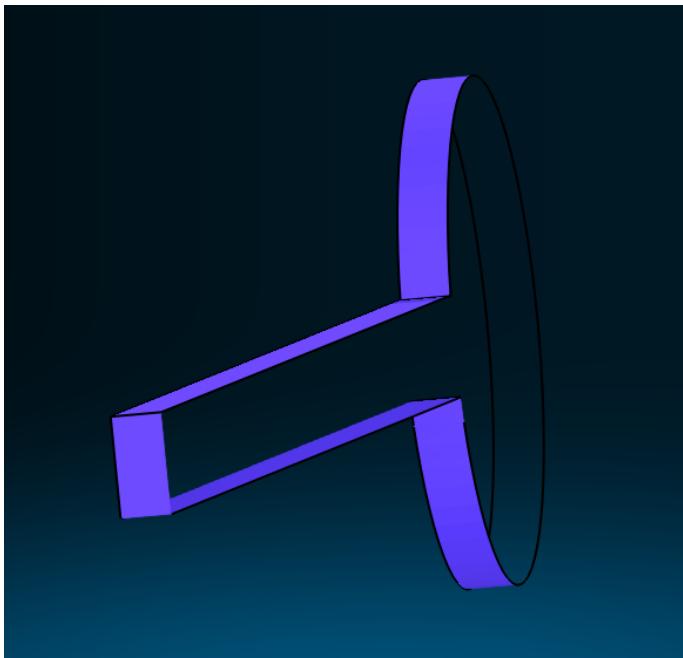
**Please note:** The **Structural Analysis** option is only available in Labs mode.

## 1.2 Materials

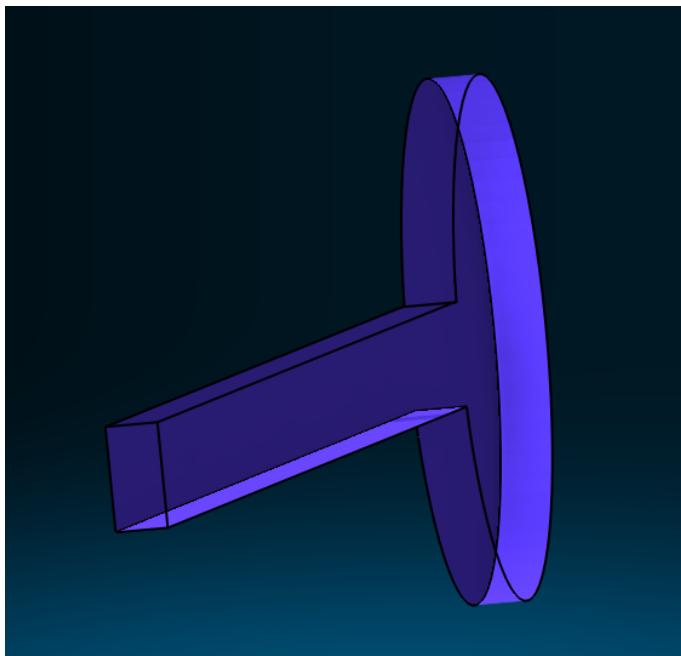
- d) In **Project Tree > Materials** set the following parameters for the **Fluid** material:
- (a) **Molecular weight:** **92.09**
  - (b) **Reference density:** **1000**
  - (c) **Viscosity model:** **Newtonian**  
**Dynamic viscosity:** **1 Pa s**

## 1.3 Geometry

- e) In **Project Tree > Geometry** import the geometry using the command **Geometry > Import a new geometry** and selecting the `MSC_Nastran_Turek_Hron.bdf` file generated in [Step 1](#) of this tutorial.
- f) Fix the normals orientation of the imported geometry by selecting it and using the command **Geometry > Reorientate normals**

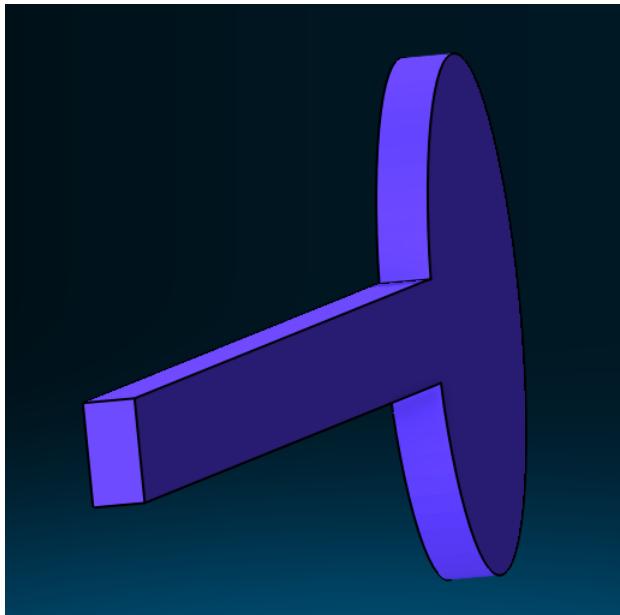


*Inconsistent normals orientation*



*Consistent normal orientation, but pointing inward*

- g) Reverse the normals so that they point outward by selecting the geometry, **right click** with the mouse and selecting **Reverse orientation**



*Outward pointing and consistent normals*

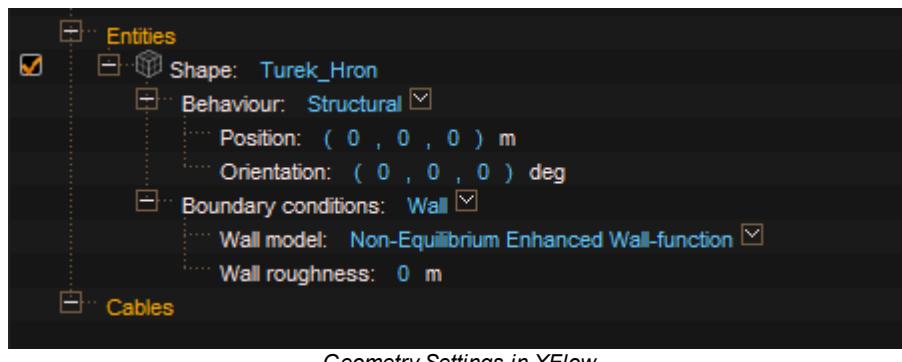
- h) Set the following parameters:

- (a) **Behaviour: Structural**

## Step 2: Problem setup - XFlow

(b) Boundary conditions: Wall

(c) Wall model: Non-Equilibrium Enhanced Wall-function



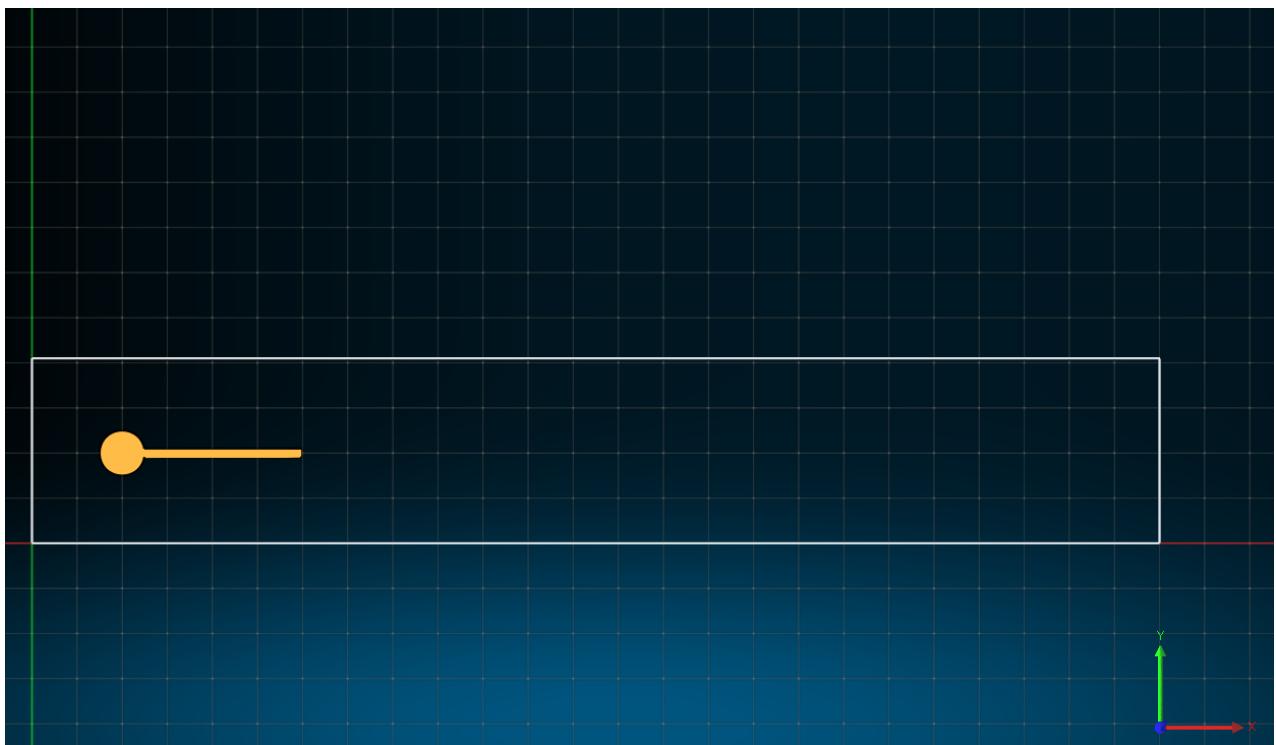
Geometry Settings in XFlow

## 1.4 Simulation

i) In Project Tree > Simulation set the following parameters

- (a) Simulation time: 20 s
- (b) Time step: 1.5e-4 s
- (c) Resolved scale: 0.005 m
- (d) Folder: Turek\_Hron
- (e) Frames frequency: 10 Hz

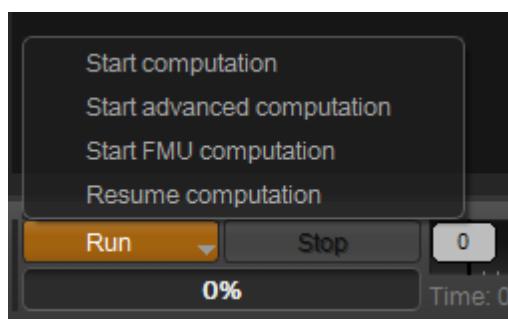
The resulting setup should look like that:



### Step 3: Execution of Co-simulation

Once the setup of the simulation in SimXpert and XFlow are completed it is possible to run the coupled simulation.

- Start XFlow simulation by selecting **Run > Start computation**



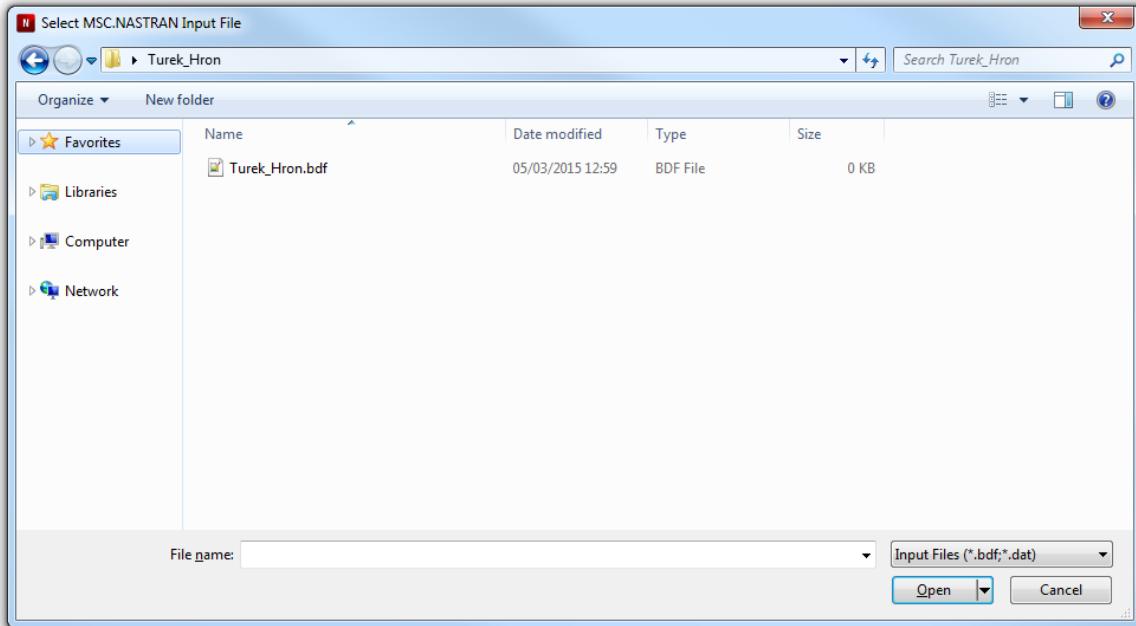
XFlow's Message View will prompt the "Waiting for initial sync point with Nastran" message.

### Step 3: Execution of Co-simulation

```
XFlow num of wet nodes: 1742
Successfully attached to: NastXFBarrier
Successfully attached to: XFNastBarrier
Successfully attached to: NumWetNodes
Successfully attached to: TokenA
Successfully attached to: TokenB
Successfully attached to: TokenC
Successfully attached to: DeltaTNASTRAN
Successfully attached to: Coordinates
Successfully attached to: Ids
Successfully attached to: Displacements
Successfully attached to: Velocities
Successfully attached to: EndSimulation
Successfully attached to: Message
Successfully attached to: TimeNASTRAN
Successfully attached to: Forces
FSI: Cleaning up all shared memory resources
Successfully attached to: XFNastBarrier
Successfully attached to: NumWetNodes
Successfully attached to: TokenA
Successfully attached to: TokenB
Successfully attached to: TokenC
Successfully attached to: DeltaTNASTRAN
Successfully attached to: Coordinates
Successfully attached to: Ids
Successfully attached to: Displacements
Successfully attached to: Velocities
Successfully attached to: EndSimulation
Successfully attached to: Message
Successfully attached to: TimeNASTRAN
Successfully attached to: Forces
Successfully attached to: NastXFBarrier
Waiting for initial sync point with Nastran
```

XFlow Message View

- Start MSC Nastran simulation by executing MSC Nastran and selecting the `Nastran_Turek_Hron.bdf` file



Execution of MSC Nastran simulation

- c) The co-simulation will start and the two software will synchronize at each timestep to exchange loads and deformations, as indicated in XFlow's Message View.

```
Showing first coordinates ...
#0      Id: 9387      0.00542      0.00501      0.043125
Enter computeFSIGeometryMapping ...
After computeFSIGeometryMapping ...
----- 20
Setup for OpenFSI with Shared Memory is done.
Initializing variables for TIME_WAITING_NASTRAN.
Reading domain data...
Reading domain structure with 1 levels...
  Level 0 has 24600 active elements
Processing domain data...
Domain loaded successfully
Reading domain data...
Reading domain structure with 1 levels...
  Level 0 has 24600 active elements
Processing domain data...
Domain loaded successfully
Saving data...
[[Data file]] 0 done!!! | Frame wall clock time[0]s | Overall wall clock time[0]s | Num elements[24600]
Sim. time [9.615385e-005]s | Stability param. [5.710058e-002] | Wall clock time [4.700000e-002]s
Sim. time [1.923077e-004]s | Stability param. [5.710058e-002] | Wall clock time [7.800000e-002]s
Sim. time [2.884615e-004]s | Stability param. [5.710058e-002] | Wall clock time [4.600000e-002]s
```

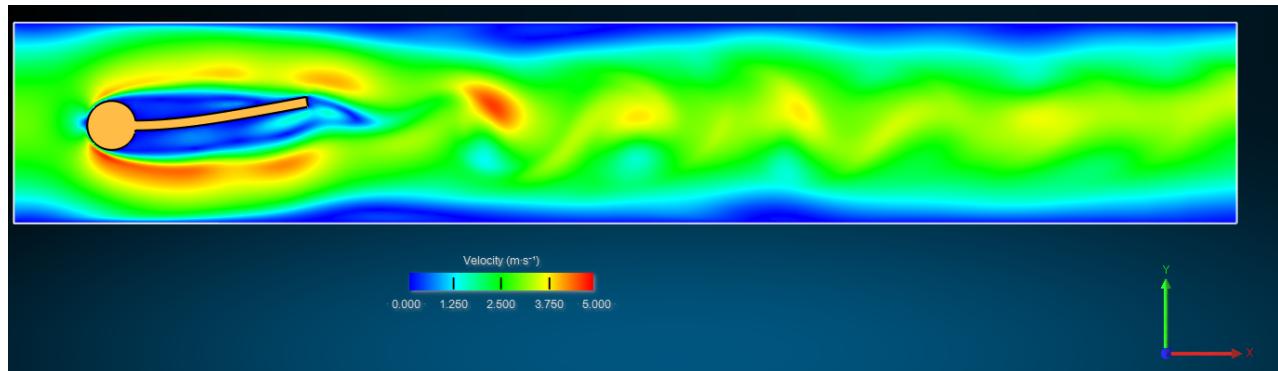
XFlow Message View showing the progress of the coupled simulation

## Step 4: Post processing

As with any XFlow simulation it is possible to perform the post-processing on the fly.

### 1. Post-process the co-simulation

- a) In **Project Tree > Environment > Engine** generate a cutting plane in the Z axis and use Velocity as field to visualize.



*Cutting plane at  $t=20$  s showing the Velocity contours.*

It is possible to see how the beam deforms under the effects of the fluid loads, but as well the influence of the beam deformation on the way the vortices are shed.