SALOME tutorial for OpenFOAM geometry & mesh

Prasad ADHAV

University of Luxembourg

September 20, 2021

The current chapter provides a tutorial on the Salome platform for geometry and mesh generation for an OpenFOAM case. SALOME can be used as a standalone application for generation of the CAD model, its preparation for the numerical calculations and post-processing of the calculation results.

Contents

1 Introduction

In the current tutorial, Salome platform is used to create geometry and mesh for an OpenFOAM case, instead of the standard `blockMeshdict`. The execution of the OpenFOAM application is done as usual. More information on Salome platform can be found here. Current version of Salome can be downloaded from this link. If there are some issues with the current version, a universal binary package for Linux is available.

In this tutorial, a simple cube building is to be created, with another larger box around it representing fluid surrounding it. The patch names to be used for OpenFOAM as can be seen in figure 1.

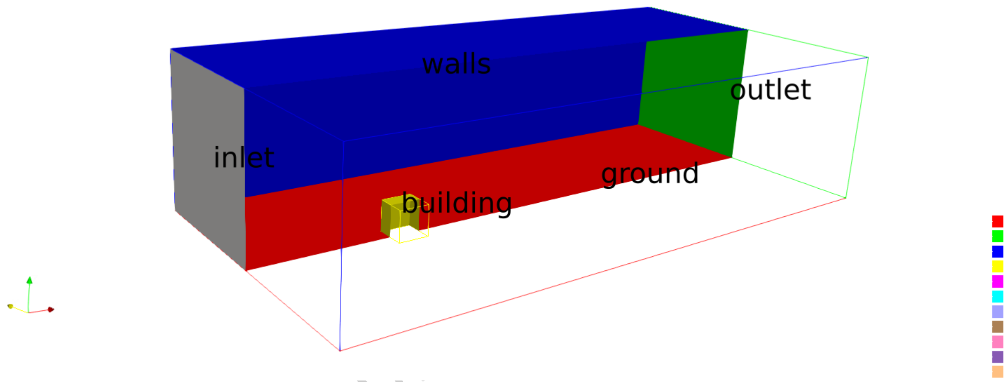


Figure 1: Problem setup for OpenFOAM with patch names

The tutorial demonstrates the following tasks:

- Basic operations to create geometry.
- Operations on geometry to prepare it for meshing.
- Meshing.
- Exporting and converting mesh files for OpenFOAM.

2 Basic features to create geometry

After a successful installation of Salome, you can launch Salome either from the desktop icon or from the command line. You should now see a the default window as shown in figure 2.

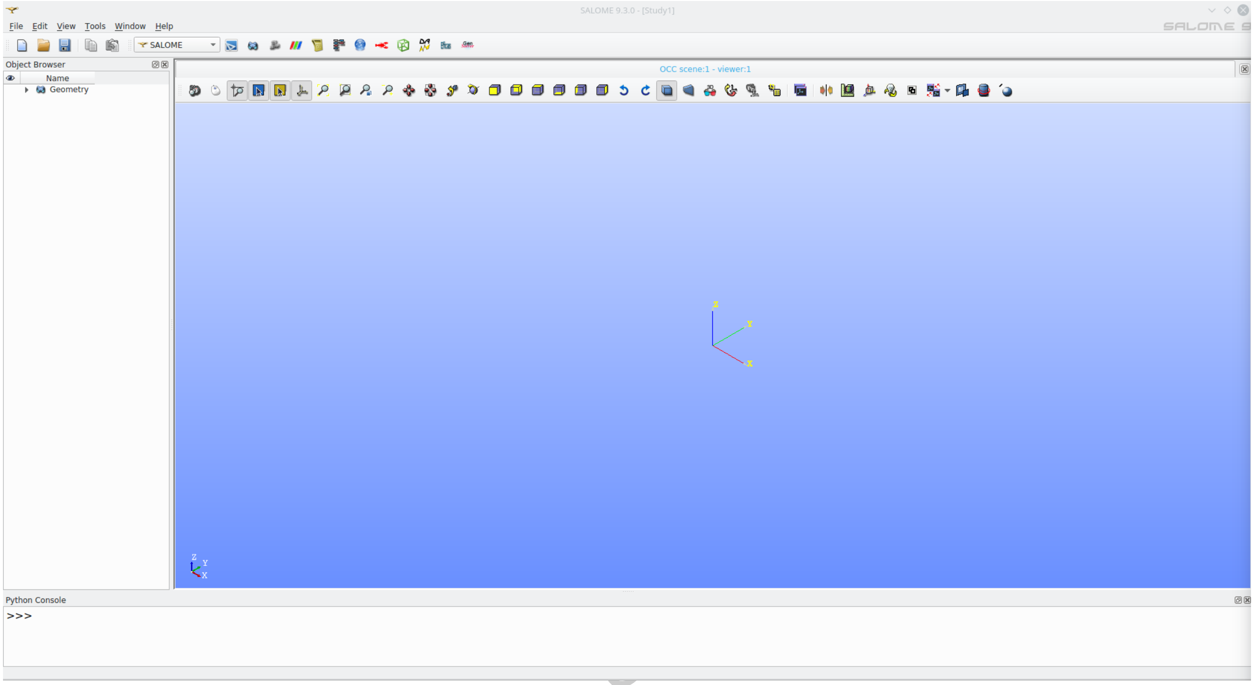


Figure 2: Salome default screen

In the tool bar, after the file options, a drop down `modules` shown in figure 3 menu is available (in current view **SALOME**. From this menu select Geometry (alternatively, the icons for same options available in drop down menu are available in the same row after modules option).

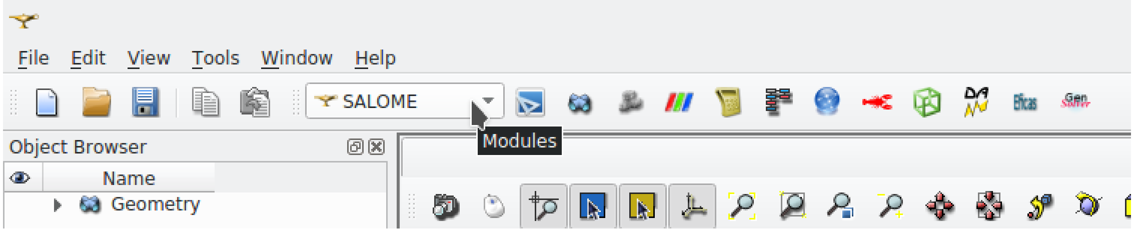


Figure 3: Salome modules option

The screen layout will change and now more options can be seen below the toolbar, as shown in the figure 4. In the third row, options to create geometry are available, which can also be accessed through the `New Entity` menu. From the `New Entity` menu, one can create point, line, sketch etc. `Primitives` are also available from which we can directly have box, cylinder, sphere etc. In the fourth row, (alternatively drop down menu `Operations`), various operations such as Boolean, Transformation operations can be done. It can be clearly seen that Salome has 3D geometry module similar to that of ANSYS classic. User has full access to elementary entities such as vertices, edges, wires, faces, shells, bodies and compound of bodies. For further learning one can follow the documentation or the video tutorials available from `Help` menu.

Note: All the dimensions are in mm, in the later stages, this will be converted to meters m.

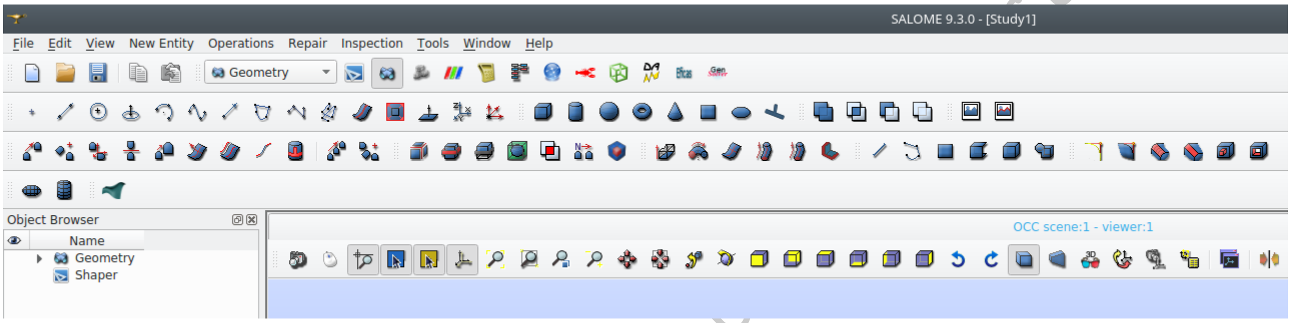


Figure 4: Salome geometry options

2.1 Quick walk-through on the Salome Layout

Please refer the figure 5 for useful icons and layout. Usually, the geometry object when created might not be visible, just click on the Reset button to view the object. If still the object is not visible use the Object On/Off (eye icon) to see it.

If the user hovers over the icon, the name of the icon can be seen. In the following tutorial, all options of **New Entity** are located in the **Basic Geometrical Tools** row. All options under **Operations** are available in **Operations on Geometry** row.

- Use `Ctrl + Left Mouse Button` for Zooming in and out
- Use `Ctrl + Right Mouse Button` for rotating the geometry

2.2 Basics of geometry

In this section, the basic commands are discussed to create a geometry using simple commands. The following will create a simple box geometry. First a point is created.

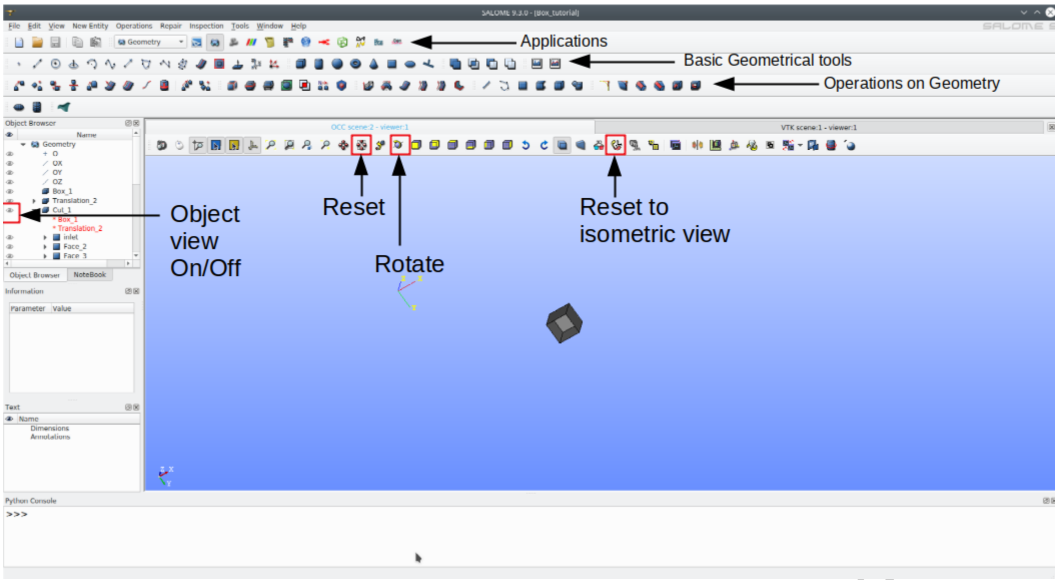
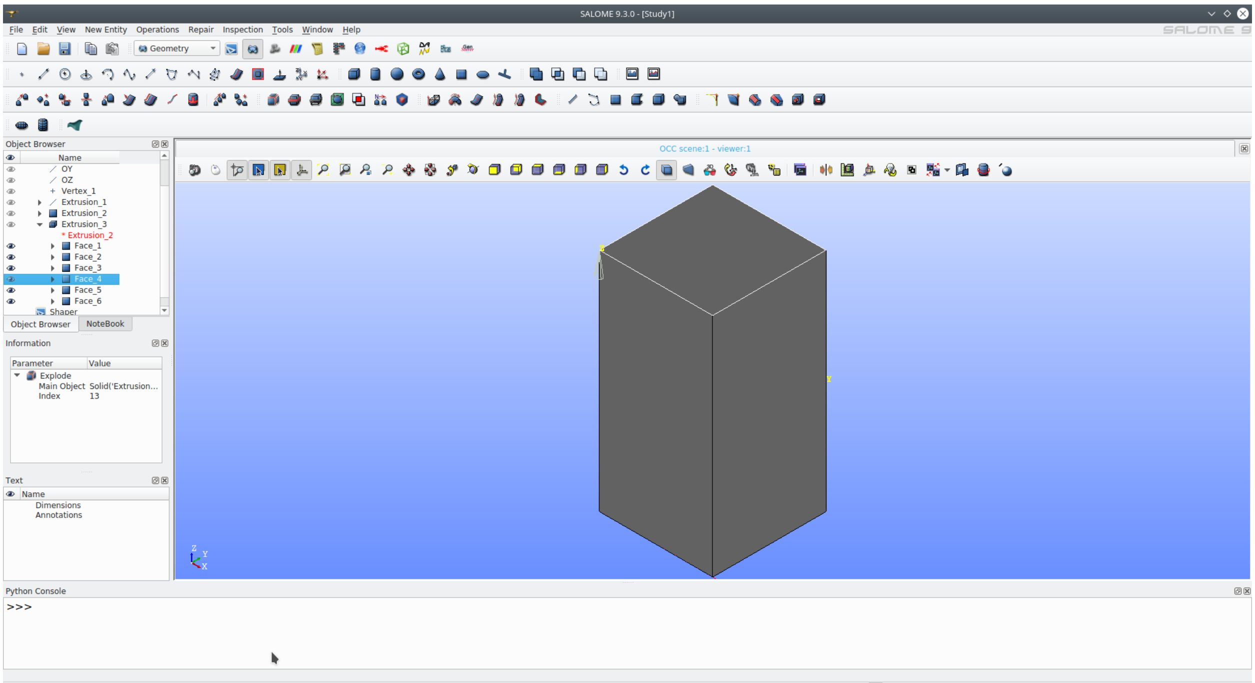


Figure 5: Salome geometry options

The point is extruded to create a line. The line is extruded to create a face. The face is extruded to create a body. Then the access of various elementary entities is shown.

- New Entity > Basic > Point > Point construction (dialogue box) > Enter Coordinates $(0, 0, 0)$ > Apply and close
- New Entity > Generation > Extrusion > Extrusion construction (dialogue box) > Select 3^{rd} option (Base shapes + DX DY DZ Vector) > Enter DZ= 10 > Apply and close
- New Entity > Generation > Extrusion > Extrusion construction (dialogue box) > Select 3^{rd} option (Base shapes + DX DY DZ Vector) > Enter DY= 5 > Apply and close
- New Entity > Generation > Extrusion > Extrusion construction (dialogue box) > Select 3^{rd} option (Base shapes + DX DY DZ Vector) > Enter DX= 5 > Apply and close
- Select Extrusion3 in pipeline > New Entity > Explode > Sub-shape Type (Drop down option) > Face > Apply and close

Now we have a box, with 6 faces as shown in figure 6. In the context of OpenFOAM, these faces will be used as inlet, outlet, walls etc. If the geometry is complete as needed, the next step is meshing. But before the geometry is meshed, the actual geometry from



Parameter	Value
Explode	
Main Object	Solid('Extrusion...
Index	13

Figure 6: Box created from the steps in this section

1, is to be created. For this, an alternative method is shown, where primitives are used to create the geometry drastically reducing the number of steps. Boolean operation is also introduced in the following section.

3 Operations on geometry to prepare it for meshing

Before proceeding to the next instructions, be sure to create a new study by

- File > New
- File > Save as > file_name.hdf

In the following instructions, a geometry of building with surrounding fluid (air) is created.

- New Entity > Primitives > Box > Box construction (dialogue box) > Enter Dimensions Dx= 20, Dy= 10, Dz= 5 > Apply and close
- New Entity > Primitives > Box > Box construction (dialogue box) > Enter Dimensions Dx= 1, Dy= 1, Dz= 1 > Apply and close

- Select Box2 in pipeline > Operation > Translation > Translation of an object (dialogue box) > Arguments > Objects > Click on Box2 > Dx= -0.5, Dy= -0.5, Dz= 0 > Apply and close

- Select Box2 in pipeline > Operation > Translation > Translation of an object (dialogue box) > Arguments > Objects > Click on Box2 > Dx= 5, Dy= 5, Dz= 0 > Apply and close

- In the pipeline, you will see Box_1, Box_2, Translation_1, Translation_2. Select Box_2, Translation_1 (Ctrl + Click) > Press Delete > Yes > Yes

- Select Box_1 in pipeline > Operation > Boolean > Cut > Cur of object (dialogue box) > Arguments > Main Object = Box_1 (Already selected) > Tool Objects = Select Translation_2 > Apply and close

- In the pipeline, you will see Box_1, Translation_2. You can delete these entities since they are no longer needed. Select Box_1, Translation_2 (Ctrl + Click) > Press Delete > Yes > Yes

- Select Cut_1 in pipeline > New Entity > Explode > Sub-shape Type (Drop down option) > Face > Apply and close

In the pipeline, 11 faces are available. When clicked on one face, the face edges will be highlighted in white. As an example, refer to figure 6. Some of these faces will be grouped together (building, walls) and some will be used as they are (inlet, outlet, ground). In the following instructions, the faces are renamed or if needed they are grouped. This will ease up the work in meshing.

- Select Face_1 in pipeline > Double click & Backspace > Type `inlet`

- Alternatively, Select Face_1 in pipeline > press F2 > Type `inlet`

- Select Face_11 in pipeline > press F2 > Type `outlet`

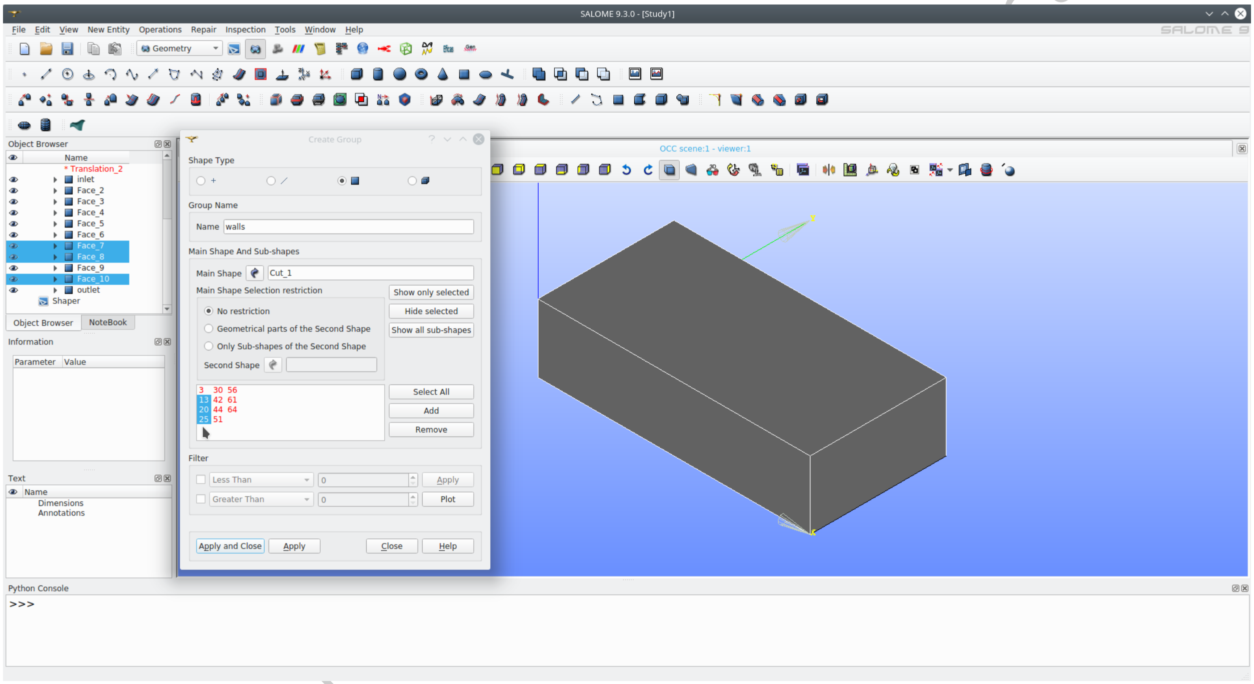
- Select Face_9 in pipeline > press F2 > Type `ground`

- New Entity > Group > Create Group > Select the Shape Type as Face (3^{rd} option) > Name `walls` > ...

- . . . Select Main shape as Cut_1 by clicking in the pipeline > Select All > Numbers will appear in the dialogue box as shown in figure 7. > 13, 20, 25 are the needed numbers > Select the rest and click on Remove > Apply and close

- New Entity > Group > Create Group > Select the Shape Type as Face (3^{rd} option) > Name `building` > . . .

- . . . Select Main shape as Cut_1 by clicking in the pipeline > Select All > Numbers will appear in the dialogue box as shown in figure 7. > 44, 51, 56, 61, 64 are the needed numbers > Select the rest and click on Remove > Apply and close



Create Group

Shape Type: ○ + ○ / ● ■ ○

Group Name
Name: walls

Main Shape And Sub-shapes
Main Shape: Cut_1
Main Shape Selection restriction
● No restriction
○ Geometrical parts of the Second Shape
○ Only Sub-shapes of the Second Shape
Second Shape:

3 30 56
13 42 61
20 44 64
25 51

Show only selected | Hide selected | Show all sub-shapes | Select All | Add | Remove

Filter
□ Less Than 0 Apply
□ Greater Than 0 Plot

Apply and Close | Apply | Close | Help

Figure 7: Creating a group of faces in Salome representing `walls` for OpenFOAM simulation

Note: The numbers representing faces in the Create Group dialogue box might differ. To select valid faces, start hiding faces by clicking the `eye` icon left of the face.

4 Meshing

In this section, the meshing of the previously created geometry will be seen. Firstly a basic mesh tetrahedral mesh is created.

4.1 Simple Tetrahedral Mesh

- Modules > Mesh > A blank window will appear. Note that different Menus are available.

- Select Cut_1 > Mesh (One of the new Menus) > Create Mesh > Create Mesh dialogue box appears > Mesh type > From the drop down menu select Tetrahedral > Algorithm > NETGEN 1D−2D−3D > In the 'Hypothesis' row, click on the gear icon > Select NETGEN 3D Parameters > Hypothesis Construction Dialogue box appears

- Hypothesis Construction Dialogue box > Max. Size = 1, Min. Size = 0.2, Fineness = Fine > Ok > Apply and close

- Select Mesh_1 in pipeline > Right click > Select Compute from drop down menu

A simple tetrahedral mesh can be now seen in the window, as seen in figure 8. By adjusting the Max. size and min. size, a finer mesh can be obtained. Other options, such as 'Second order' can be selected in the 'Hypothesis Construction Dialogue box' for 2^{nd} order mesh.

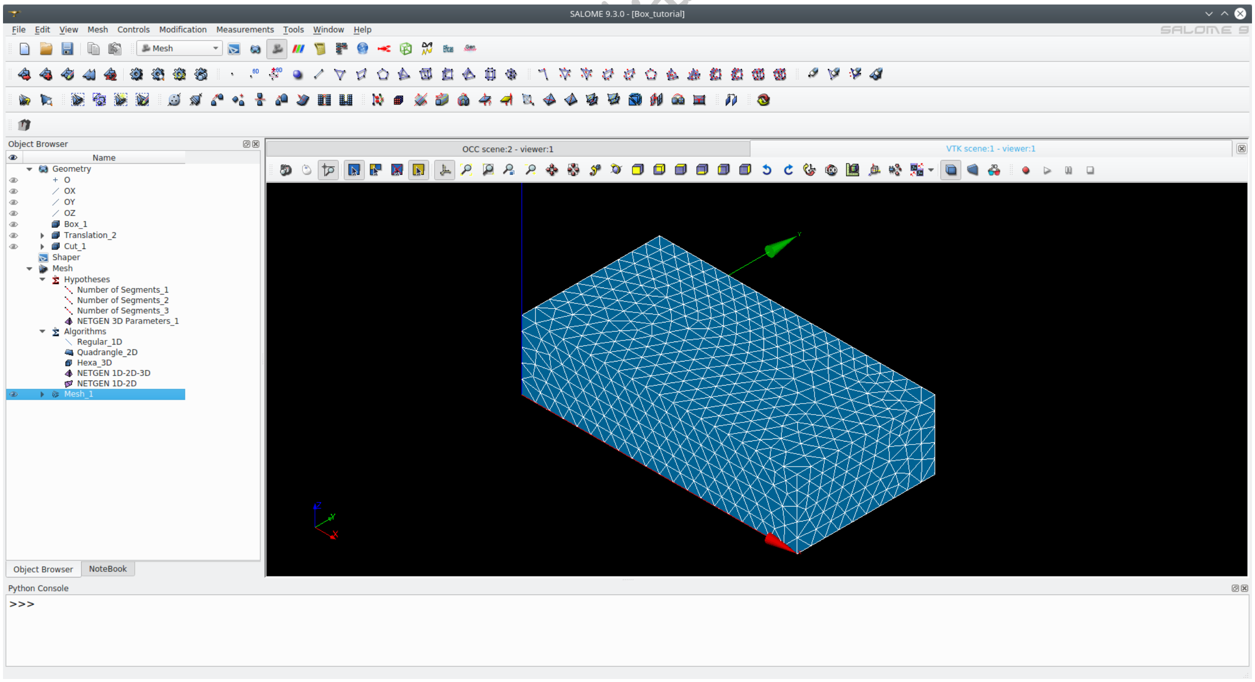


Figure 8: Simple tetrahedral mesh

4.2 Refined Mesh with viscous layer

In the following mesh, the mesh size is chosen smaller for refined mesh and viscous layers hypothesis is added for better resolution at the boundaries.

- Select Cut_1 > Mesh > Create Mesh > Create Mesh dialogue box appears > Mesh type > From the drop down menu select Tetrahedral > Algorithm > NETGEN 1D−2D−3D > In the 'Hypothesis' row, click on the gear icon > Select NETGEN 3D Parameters > Hypothesis Construction Dialogue box appears

- Hypothesis Construction Dialogue box > Max. Size = 0.5, Min. Size = 0.1, Fineness = Fine > Ok > Apply and close

- Add. Hypothesis row > click on the gear icon > Select Viscous layers > Hypothesis Construction Dialogue box appears

- Hypothesis Construction Dialogue box > Total thickness= 0.5 > Number of layers= 10 > Leave Stretch factor and Extrusion method as thet are > Specified faces are > Select Faces with layers(walls) > In pipeline select faces/face group `walls, ground, building` > Click Add in the Hypothesis Construction Dialogue box > Ok > Click on the Plus icon > Apply and close

- Select Mesh_1 in pipeline > Right click > Select Compute from drop down menu

Now, groups are created for mesh from the faces. By doing this all the mesh information is available for that face. These mesh group names will be used as boundaries in OpenFOAM (Eg. inlet, outlet, walls etc). If all the steps were followed until now, there are 2 options, user may select any of the options.

- Select Mesh (This case Mesh_1) from pipeline > Mesh > Create Group > Elements type = Face > Group type = Group on geometry > Click on the Arrow next to 'Geometrical Object' > Click on `inlet` face from geometry part of pipeline > Apply

- Note that the dialogue box stays open. This is because a mesh group is to be created for all the named faces/face groups created in geometry. Repeat the step given above for `ground, outlet, walls, building` > Close

- Under the Mesh_1, `Group of Faces` can be seen. Expand `Group of Faces` by clicking the arrow next to it. There are 5 groups available `Group_1` . . . to . . . `Group_5`.

By clicking the arrow next to the Group, the parent entity can be seen. According name the Groups.

- Select `Group_1` in pipeline > Press F2 > Type `inlet`
- Similarly rename the remaining groups according to their parent faces/face groups. (`ground, outlet, walls, building`). The groups fo Faces should have something similar as seen for Mesh_1 in figure 9.
- Repeat the above steps for Mesh_2

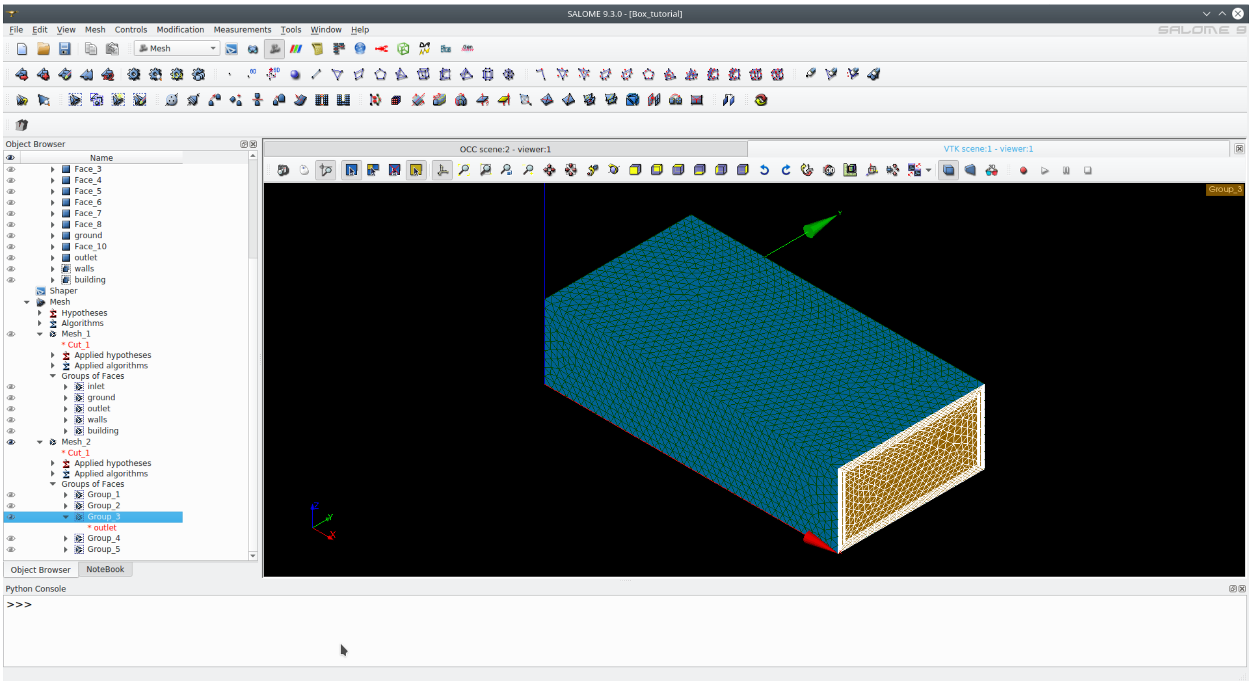


Figure 9: Mesh groups created for faces. Outlet group highlighted.

4.3 Hexahedral Mesh

For hexahedral meshing, the body is needs to be divided into simple parts. The current geometry might seem it is already simple, but the building cut in the box doesn't make it so. For making simple bodies, the geometry is to be divided (Partitioned, `Operation > Partition`). For making these partitions, planes are to be created along the faces of the `building`. To make the process easy, only ground and building is made visible using the eye icon in front of the face/face group. Now, only ground and the building should be visible. An alternative is also given in the following instructions.

- Select `ground` and `building` in the pipeline > Right click > Show Only
- New Entity > Basic > Plane > Plane construction by face (3^{rd} option) > Select Face_2 > Size of plane = 50 > Apply and close
- Repeat the above step for Face_3 through Face_6 > Close the dialogue box
- Select Cut_1 in the pipeline > Operations > Partition > Click on arrow icon next to Tool Objects > Press Ctrl and select Plane_1 through Plane_5 > Resulting Type = Solid > Apply and close
- In pipeline right click on Partition_1 > Show Only

Currently, the geometry is ready for hex meshing. From module go to meshing window, and follow the instructions:

- Select Partition_1 in the pipeline > Mesh > Create Mesh > 3D Algorithm = Hexahedron (i,j,k) > 2D Algorithm = Quadrangle: Mapping, Hypothesis = Quadrangle Parameters = Standard = Ok > 1D Algorithm = Wire Discretisation, Hypothesis = Number of segments = 10 > Apply and close
- Select Mesh_3 in pipeline > Compute

A basic hex mesh is now visible, as seen in figure 10. This mesh can be further improved. Two improvement options will be discussed here. In one option and edge will be divided further (equidistant) and in second option, the mesh size increases from a small value and increases to max. value, giving a kid of gradient along the long edge.

In the first option of mesh refinement, the long edge is divided into 20 parts instead of 10 as done previously. This can be done on the existing hex mesh created in the previous steps. But it is advised to duplicate the previous steps and create another mesh same as seen in figure 10. If another mesh was created, this should be the Mesh_4.

- Go to Geometry through module
- Select Partition_1 in the pipeline > Right click and select Show Only
- Select Partition_1 in the pipeline > New Entity > Explode > Sub-shape Type = Edge, Check box for Select Sub-Shapes, select the longest edge available (see figure 11 > Apply and close

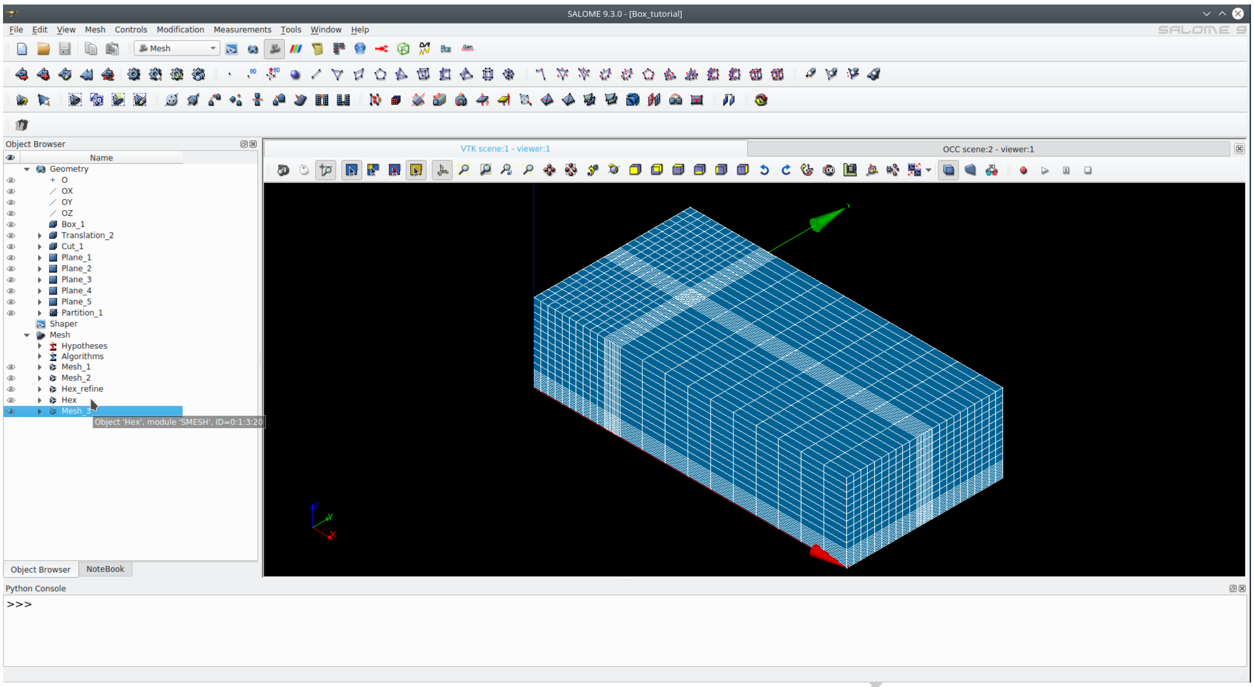
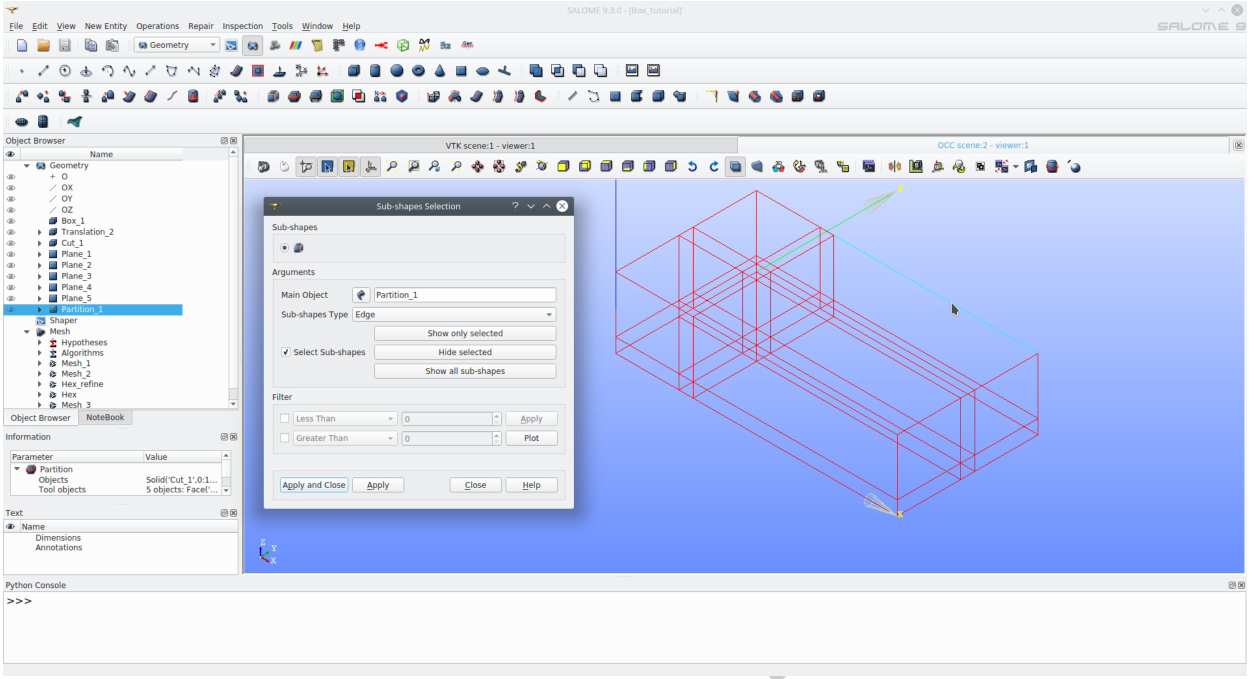


Figure 10: Basic hex mesh with 10 divisions along each edge

- Other edges (as seen in figure 11), may as well be selected for further refinement of mesh
- Go to mesh module
- Select Mesh_4 in pipeline > Mesh > Create sub-mesh > Mesh = Mesh_4 > Geometry = Edge_1 (Under Partition_1) > 1D > Algorithm = Wire Discretisation > Hypothesis = Number of segments = 20 > Apply and close
- Select Mesh_3 in pipeline > Compute

From the previous sections, the long edge was divided into 20 equidistant parts, as seen in the left image of figure 12. Now the same edge will be divided into various parts. This division is done by describing the Start length (smallest element length) and End length (Max length of element). The size of element length is gradually increased along the edge from Start (smallest) length to End (largest) length, as seen in the right image of figure 12. It is advised to follow the instructions for the basic hex mesh. The first three steps are repetition from previous section, hence the operations done on geometry to obtain the edge can be ignored.

- Go to Geometry through module



Parameter	Value
Partition	
Objects	Solid('Cut_1',0:1...
Tool objects	5 objects: Face('...

Sub-shapes Selection

Sub-shapes

Arguments

Main Object: Partition_1

Sub-shapes Type: Edge

☑ Select Sub-shapes

Figure 11: Basic hex mesh with 10 divisions along each edge

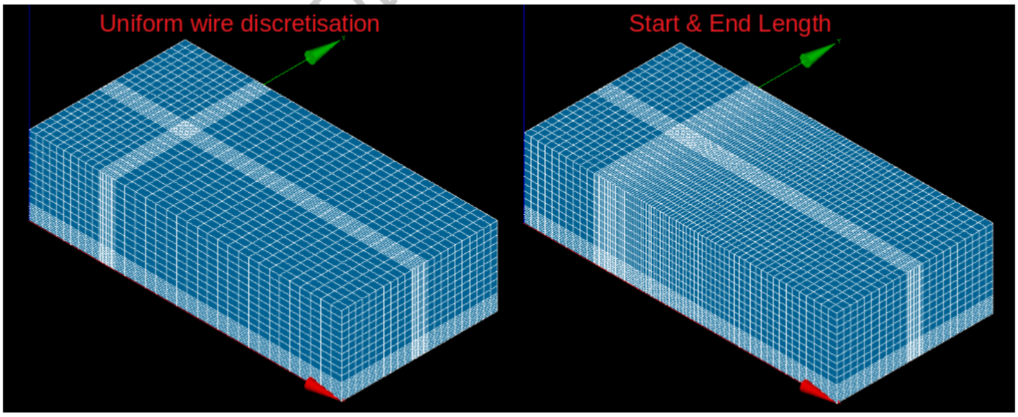


Figure 12: Hex mesh with 20 divisions along long edge(left) and Hex mesh with Start & End length

- Select Partition_1 in the pipeline > Right click and select Show Only
- Select Partition_1 in the pipeline > New Entity > Explode > Sub-shape Type = Edge, Check box for Select Sub-Shapes, select the longest edge available (see figure 11 > Apply and close
- Other edges (as seen in figure 11), may as well be selected for further refinement of mesh
- Go to mesh module
- Select Mesh_5 in pipeline > Mesh > Create sub-mesh > Mesh = Mesh_4 > Geometry = Edge_1 (Under Partition_1) > 1D > Algorithm = Wire Discretisation > Hypothesis = Start and End length > Start and End length dialogue box > Start length = 0.1, End length= 0.5 > Ok > Apply and close
- Select Mesh_3 in pipeline > Compute

The refined mesh can be seen in the right of figure 12.

5 Exporting and converting mesh files for OpenFOAM

After the previous step, all the necessary components for a mesh needed for OpenFOAM simulation are available. The mesh is to be exported in `.unv` format and converted so that OpenFOAM can read it.

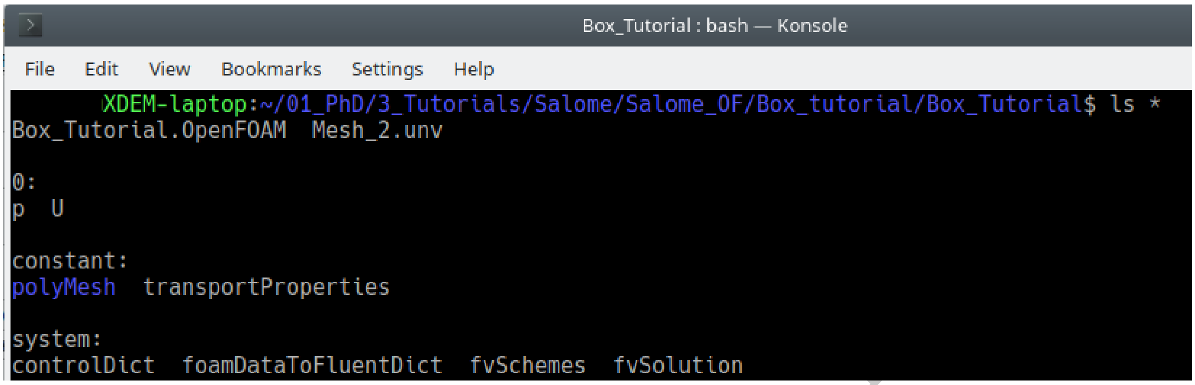
- Select Mesh_1 in Pipeline > Right click > Export > UNV file > Enter file name (here Mesh_1) > Save

To convert the mesh, OpenFOAM command is used through terminal. But before that, a project directory is to be made.

- Open terminal > `mkdir Case_name` > `cd Case_name`
- Copy `0` and `system` folder from any of the OpenFOAM tutorials. (The next command to be used detects the `0` and `system` folders. It is not actually dependent on the contents of these folders. These can be changed or replaced later on as needed.)
- For the current tutorial they are copied from the `elbow` tutorial.

- In terminal enter following command > `(Comp/Case_name)$> ideasUnvToFoam Mesh_1.unv`

After using this command, the `constant` folder is created, with `polymesh` folder in it. Copy the `transportProperties` from the tutorial folder, in the `constant` directory. In the figure 13, the file structure of the case setup is seen.



```
XDEM-laptop:~/01_PhD/3_Tutorials/Salome/Salome_OF/Box_tutorial/Box_Tutorial$ ls *
Box_Tutorial.OpenFOAM  Mesh_2.unv

0:
p  U

constant:
polyMesh  transportProperties

system:
controlDict  foamDataToFluentDict  fvSchemes  fvSolution
```

Figure 13: File structure of the OpenFOAM case set up.

The current geometry was create with meters, so instead of entering dimensions as 20000 as length of the bounding box, it was entered as 20, and so on for other dimension. When the geometry is created with `mm` as dimensions, use the following command to convert the dimensions from `mm` to `m`.

- `(Comp/Case_name)$> transformPoints -scale` ′(0.010.010.01)′
- The mesh can be checked by the usual command `(Comp/Case_name)$> check Mesh`. This is a good way to confirm the conversion from `mm` to `m` worked.
- Depending on which solver is being used, now we can actually start the simulation. In this case it is `icoFoam`

6 Results and comparison

6.1 Meshing

The comparison of two meshes (coarse and refined is done) in figure 14.

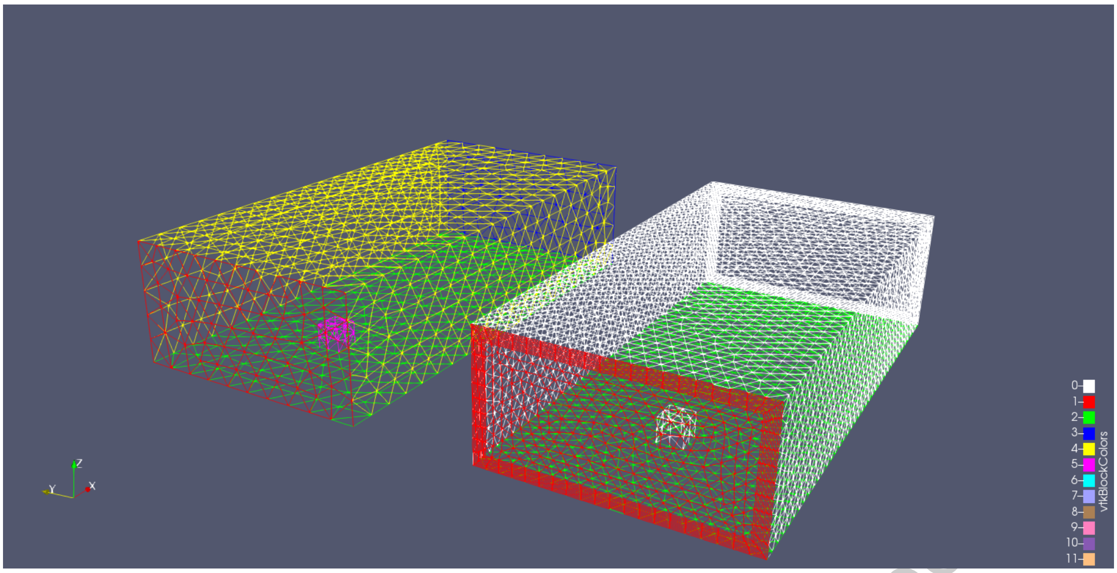


Figure 14: Comparison of meshes

6.2 Velocity

The velocity result is shown for coarse and refined mesh in the figure 15. The velocity result is seen much better in the fine mesh with the viscous layer seen in figure 16. Of course the result is not as good around the `building`, but this can be solved by local refinement of the mesh around the `building`.

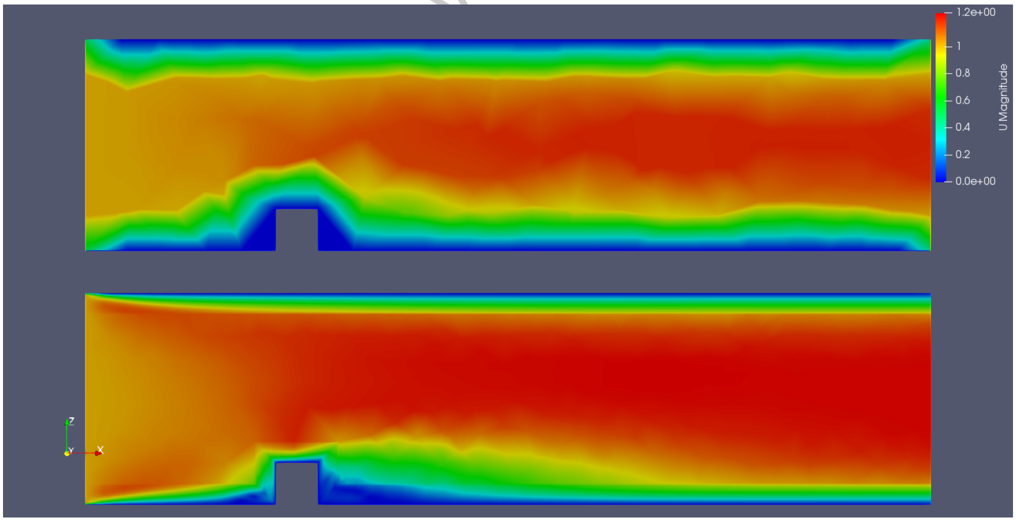


Figure 15: Comparison of meshes through velocity field. Coarse Mesh (Top), Fine Mesh (Bottom)

The object was just to demonstrate the meshing possibilities.

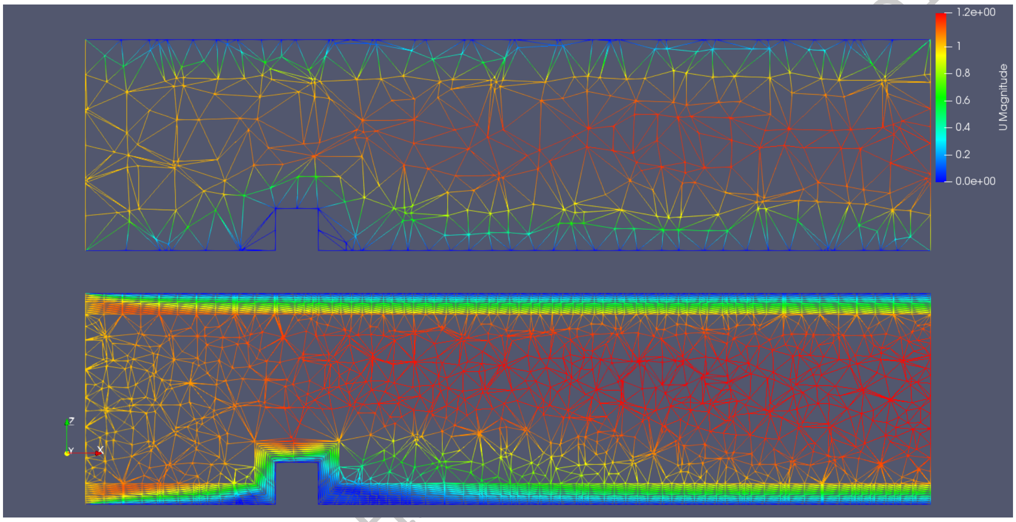


Figure 16: Comparison of meshes through velocity field wire-frame. Coarse Mesh (Top), Fine Mesh (Bottom)