Contents

List of Figures

List of Tables

Chapter 1

Installing OpenFOAM and Paraview

The First chapter deals with Installing OpenFOAM and Paraview. We are using Linux Operating System for installation and OpenFOAM-2.2.1 and Paraview-3.12.0. First we will look how to install OpenFOAM and paraview using Synaptic Package Manager. Then using the downlading it from the OpenFOAM website and lastly installing it using the source code. We will end this chapter with an example which shows running a simple problem in Paraview. As a basic requirement the user expected to have some basic knowledge of Computational Fluid Dynamics (CFD) and should be able to use basic Linux Commands.

1.1 Installation using Synaptic Package Manager

OpenFOAM and Paraview can be installed using Synaptic Package Manager. On the left side of your computer screen you can see the Launcher with the list of softwares. Click on the search box ,Fig.1.1 on top of the Launcher and type Synaptic. This will display the Synaptic Package Manager. Click on it to open.

Figure 1.1: Search Icon on top of Launcher

You will be interrupted to enter the system password.

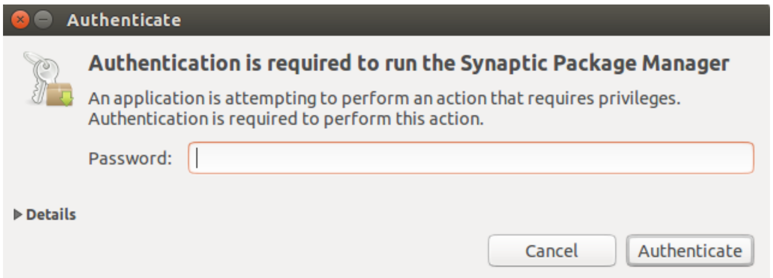


Figure 1.2: Enter system password to open Synaptic Package Manager

Once the Synaptic Package Manager is Opened, in the search box type OpenFOAM.

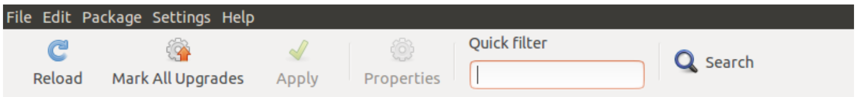


Figure 1.3: Search Box

You will see both OpenFOAM-2.3.0 and Paraview-4.1.0. Right Click Both of them for installation and click Apply to install, Fig 1.4. This might take some time to install depending upon your internet speed.

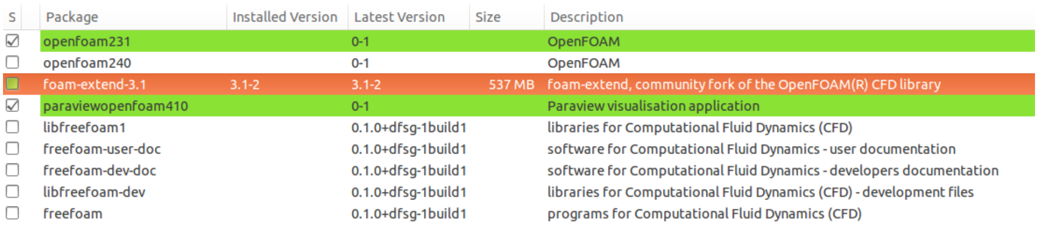


Figure 1.4: Install OpenFOAM and Paraview

1.2 Installtion from OpenFOAM website

OpenFOAM can also be downloaded and installed using the OpenFOAM website. Follow the steps given below for installation.

- On your browser type **www.openfoam.com/download**
- Go to Ubuntu Debian Installation
- Under the first point of Installation copy the command line and paste this in your terminal window
- Open the terminal window by pressing **Ctl+Alt+t** keys simultaneously on your keyboard or you can also open it using the search icon on top of the Launchbar

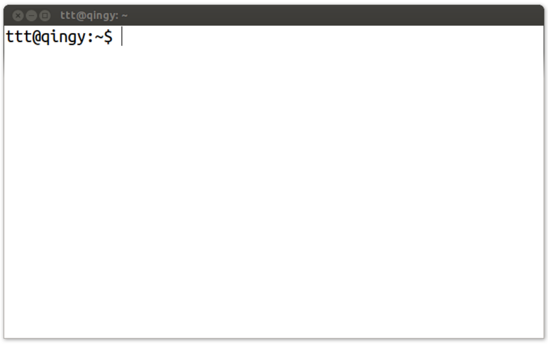


Figure 1.5: Terminal window

- For complete installation for OpenFOAM and Paraview follow the steps under Ubuntu installation page

To configure the installed software we need to edit the bashrc file. To do this open a new command terminal and type

gedit ~/.bashrc

and press $<$ *enter* $>$

After the bashrc file is opened scroll down to the bottom of the file. Then go back to your browser (OpenFOAM download page) and scroll down to **User Configuration**. Copy the line in point number 2

source /opt/openfoam230/etc/bashrc

and paste it at the bottom of the bashrc file. Save it and close the file.

To check if OpenFOAM is installed properly open a new command terminal and type

icoFoam -help

and press enter. You will see a "Usage" message on your terminal screen, Fig 1.6 which shows that the installation is done.

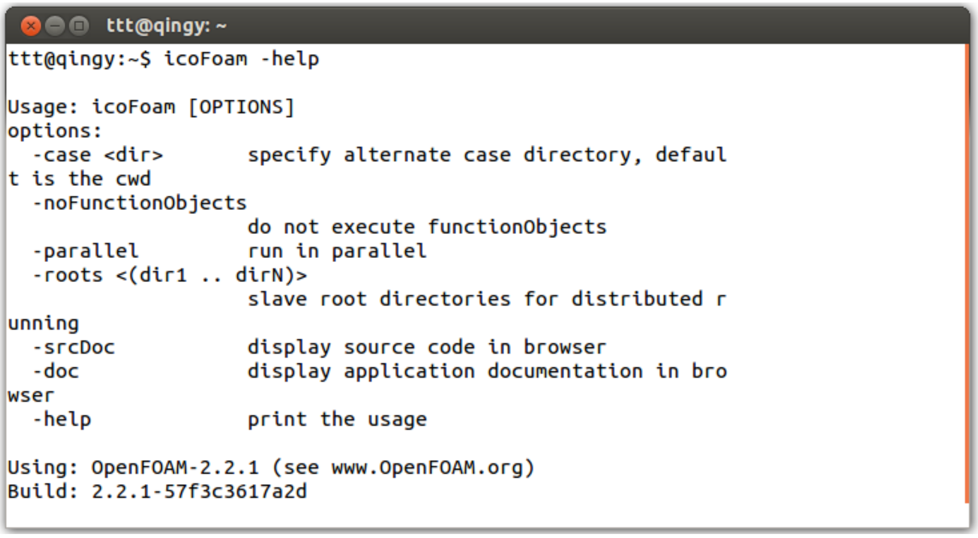


Figure 1.6: Usage Message

Now we will set up the working directory and copy the tutorial folder. Follow the steps given below.

1. Open up a new terminal and type **mkdir -p $FOAM_RUN** and press < *enter* >
2. Now type **cp -r $FOAM_TUTORIALS $FOAM_RUN** and press < *enter* >. This will copy the tutorials folder into the run directory.

Installation of OpenFOAM using the Debian package is now complete. Similarly you can download it for other linux OS such as Fredora, OpenSUSE.

1.3 Installation using Source Code

Alternate way to install OpenFOAM and Paraview is by Compiling the Source code available under the header of **Source Pack** Installation on the OpenFOAM website. Download the tar files available in **OpenFOAM.tar.gz** and **ThirdParty.tar.gz** format. Create a folder in your Home directory by the name OpenFOAM and paste the tar files in that folder and Extract the files in that folder. Follow the steps given on the OpenFOAM source pack installation page to complete the installation. Since we compile the source code it might take a few hours to complete.

1.4 Example Problem - Lid Driven Cavity

We will solve an problem here by the name Lid Driven Cavity. It is a two dimensional problem where the upper plate moves and other three sides of the plate are fixed / stationary, 1.7. The solver we use here is icoFoam which is an Transient solver for incompressible flow.

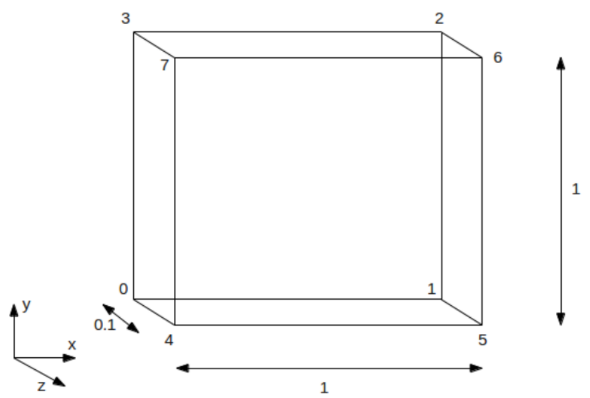


Figure 1.7: Lid Driven Cavity

In the terminal type the path given below :

cd OpenFOAM/OpenFOAM-2.3.0/run/tutorials/incompressible/icoFoam/cavity

Meshing the geometry

We need to mesh the geometry. This can be done using the blockMesh utility of OpenFOAM. In the command terminal type **blockMesh** and press < *enter* > which completes the meshing, Fig 7.3

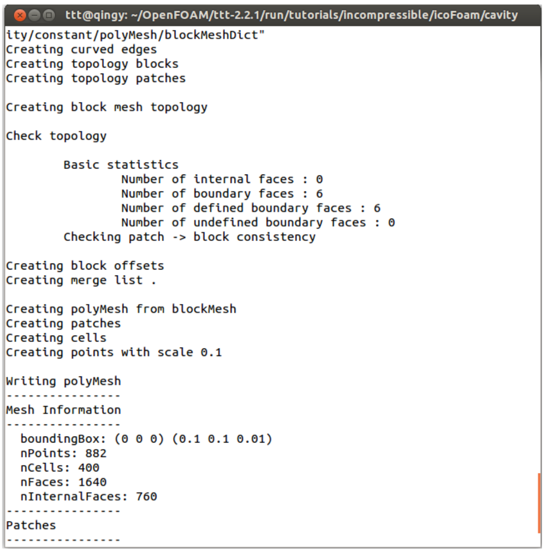


Figure 1.8: blockMesh for meshing

Solving

Once meshing is done we now run the solver by typing :

icoFoam

in the command terminal and press $< enter >$. The iteration running can be seen in the terminal window,Fig 1.9.

We have now solved the lid driven cavity case.

Visualization

To Visualize the results we use Paraview. To open paraview in your terminal type

paraFoam

and press $< enter >$. This will open up the paraview window, Fig 1.10.

Click on the Apply button on the left hand side of the **Object Inspector** Menu to view the Geometry, Fig7.6.

This brings us to the end of the first chapter. To summaries we have learnt to Install OpenFOAM and Paraview and ran a test example. The next chapter will cover about creating simple geometry in OpenFOAM.

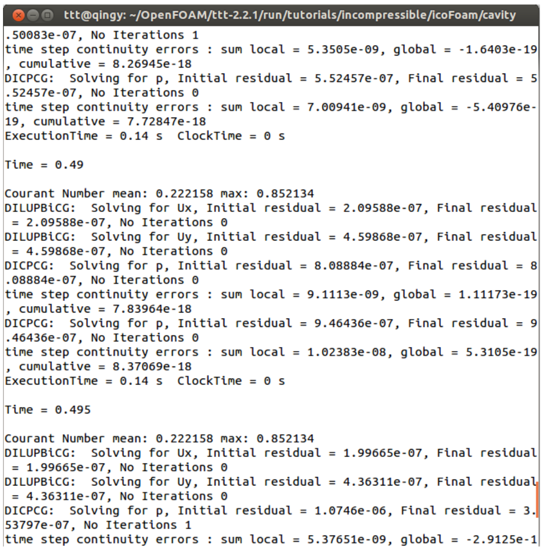


Figure 1.9: Iteration on Terminal Window

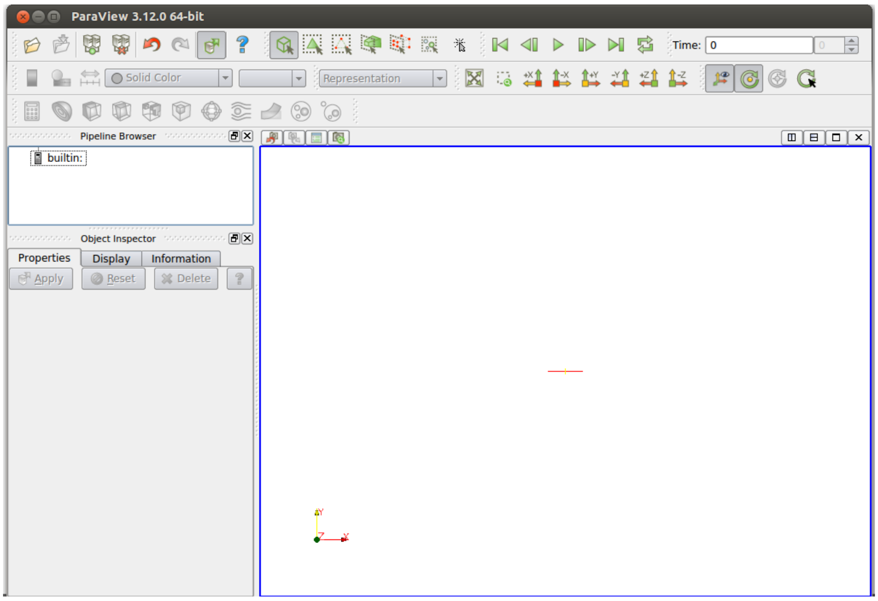


Figure 1.10: Paraview window

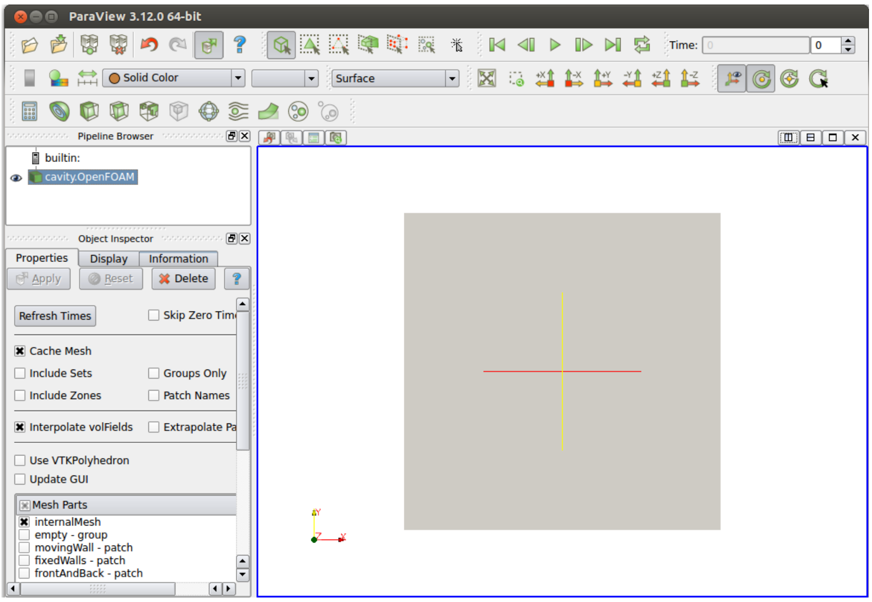


Figure 1.11: Geometry

Chapter 2

Creating a Simple Geomtery in OpenFoam

In this chapter we will learn how to create a simple geometry in OpenFOAM using the blockMeshDict utility of OpenFOAM. We can create simple geometries like a square, rectangle , circular cylinder using blockMeshDict.

2.1 Geometry creation

Here we will use the lid-driven cavity problem example mentioned in the previous chapter for the pre-processing. As previously mentioned you can type the following path in the command terminal to open the id-driven cavity problem: cd OpenFOAM/OpenFOAM-2.3.0/run/tutorials/incompressible/icoFoam/cavity

After this if you type ls in the command terminal would see three folder inside it given as:

- 0
- constant
- system

where the 0 folder gives the initial boundary conditions, constant gives the geomtery file and system folder gives the number of the iterations the solver would run along other important files. You can find the boundary of the problem in a polymesh folder inside constant. In order to open that type the following in the command terminal and then press $< enter >$:

cd constant/polymesh

Then type ls to in the command terminal and press $< enter >$. This shows the geomtery file given as blockMeshDict file. In order to view this file type the following in the command terminal:

gedit blockMeshDict

where gedit it the name of the editor we have used. Note that you may use any other text file editor to view and edit this file.

Now you can see the gedit window containing the geometry file. In order to draw a geomtery in OpenFoam you need to follow the below mentioned instructions.

In openFoam a geomtery is broken down into small blocks and are then numbered starting from 0, as shown in the Fig 5.1

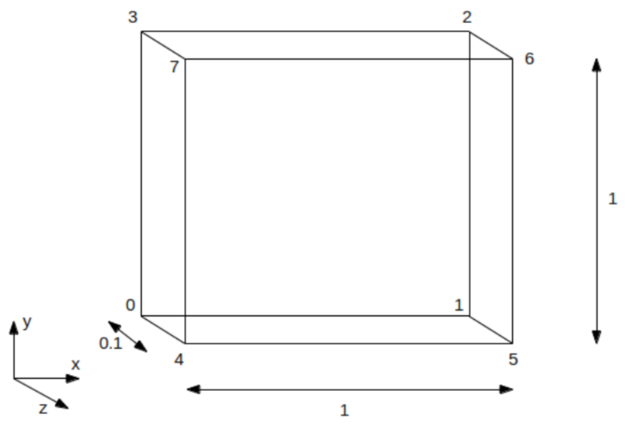


Figure 2.1: geomtery points of the lid driven cavity

2.2 blockMeshDict

Note that in openFoam to create a 2-D geometry you need to give a unit cell thickness in the Z axis. Now in order to create a new geomtry file open a new folder in destop and rename it a blockMeshDict.

A blockMeshDict file basically has the following parts:

- Foam File details
- vertices
- blocks
- edges
- boundary

- mergepatchpairs

Note that the line convertToMeter gives unit in which the geomtery is drawn. For example, as we are drawing the geomtery in meters for this problem we will keep convertToMeters as 1. Now after opening the new blockMeshDict file created in the desktop copy the lines from initial Foam File till convertToMeters from the old file and paste it. After this type vertices and then you can give the X, Y and Z co-ordinates of the boundary as shown below:

```
vertices
(
    (0 0 0)
    (1 0 0)
    (1 1 0)
    (0 1 0)
    (0 0 0.1)
    (1 0 0.1)
    (1 1 0.1)
    (0 1 0.1)
);
```

Figure 2.2: coordinates of boundary geomtery points of the lid driven cavity

Then type block, inside which you give the details of the boundary co-ordinates along with the number of mesh divisions in X, Y and Z direction in the following way, fig 3.4:

```
blocks
(
    hex (0 1 2 3 4 5 6 7) (30 30 1) simpleGrading (1 1 1)
);
```

Figure 2.3: block details of the geomtery

Here hex represents hexahedral block and the number next to that gives the names of the points at the boundary in clock-wise direction to form a block. Note that for more than one blocks the number of points would be more. The number of grid points can be modified as per requirement. For this problem we have used a 2-D mesh having 30x30 divisions and unit dept. Now since we have all straight edges in this geometry, we will keep the egdes empty.

```
edges
(
);
```

Figure 2.4: edge details of the geomtery

Next we give the details of the boundary. In the geometry we can see the following boundary conditions,as shown in fig 7.4:

- moving wall
- fixed wall
- front and back

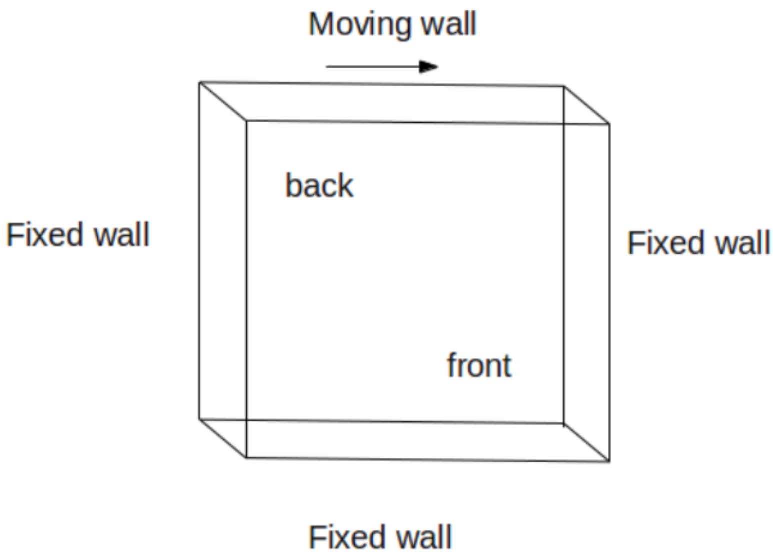


Figure 2.5: boundary names of the geomtery

where it has a top moving wall and three fixed wall. The front and back faces are kept empty as this is a 2-D problem.

Now in the blockMeshDict file you can type the boundary as shown in fig 2.6:

```
boundary
(
    movingWall
    {
        type wall;
        faces
        (
            (3 7 6 2)
        );
    }
    fixedWalls
    {
        type wall;
        faces
        (
            (0 4 7 3)
            (2 6 5 1)
            (1 5 4 0)
        );
    }
    frontAndBack
    {
        type empty;
        faces
        (
            (0 3 2 1)
            (4 5 6 7)
        );
    }
);
```

Figure 2.6: boundary details of the geomtery

Here within the boundary names enter the type of boundary used and then faces, giving the points of the block forming a particular boundary. Note that you should be very careful while writing the order of the points. The order should be such that if you place a folded palm on the surface of a boundary the thumb should be pointing normal to the surface and the fingers should be folded such that they make a curl in clockwise or anti-clockwise direction. Note that you should use either clockwise or anti-clockwise convention throughout the file and but not both. Also you should be very careful regarding openning and closing of brackets in this file.

After this, in a new line type mergePatchPairs. Since in this problem we do not have to merge any patches we will keep this empty, fig 2.7.

```
mergePatchPairs
(
);
```

Figure 2.7: merge patch details of the geomtery

Note that two P's are capital here.

After completing writing this file save it and close this file. Thus you have learned ho wto create a geomtery file.

Now go back to the command terminal and type the following twice to go back to cavity folder:

cd ..

Next you can mesh this geomtry by typing blockMesh in the command terminal. After this you can view the geometry by opening paraview. For this type paraFoam in the command terminal and press $< enter >$.

In the paraview window press Apply button on the left hand side of the Object Inspector Menu to view the Geometry, as shown in the fig 4.3:

As you as learned in the previous chapter, you can use different feature in the paraview window to check the details of the geometry.

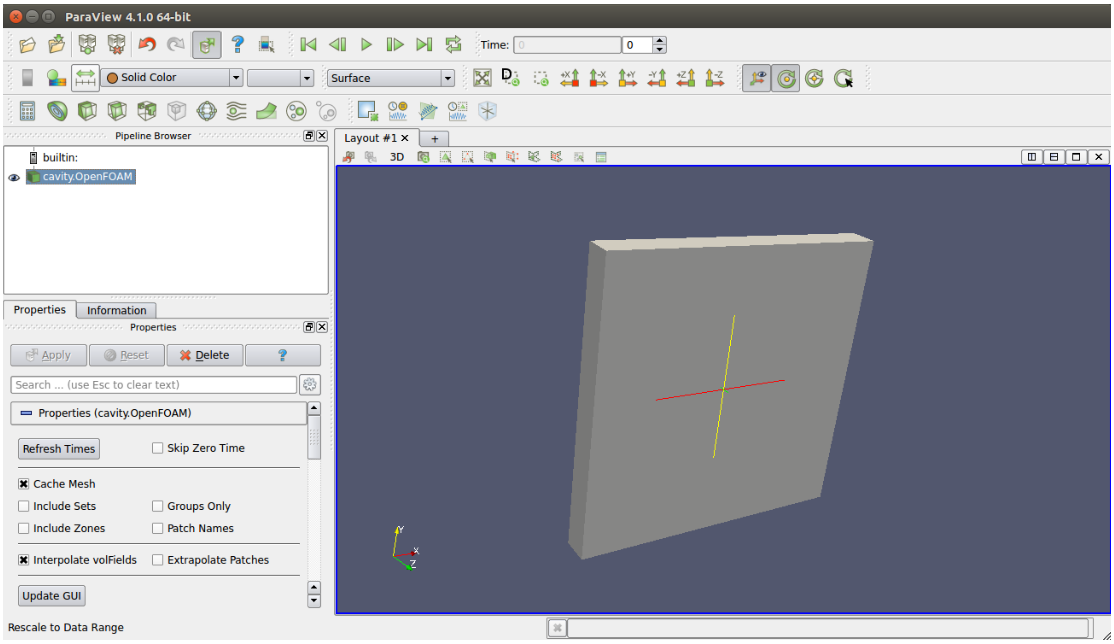


Figure 2.8: Paraview window showing the 2-D geometry

Chapter 3

Creating a Curved Geomtery in OpenFoam

In this chapter we will learn how to create a 2-D geometry for a flow over a cylinder. As it is an axis-symmetric geometry, we will consider the cylinder as a semi-circle and a rectangular domain around it. For meshing this geometry you should divide the domain into small hexahedral blocks. In this particular problem we have body fitted grid, as shown in Fig 3.1 and Fig 3.2:

Now, as mentioned in the previous chapter create a new blockMeshDict file and open it. In this geometry file copy the initial few lines till convertToMeters from previous lid-driven cavity problem blockMeshDict file and paste it. Here we will keep convertToMeters as 1 as we have given the geometry in meters. After this write down the coordinates of the vertices of the geometry in a similar fashion as given in the previous chapter.

For this particular problem we have used cosine and sine function to calculate the vertices on the curved edges as shown below:

where angle θ is calculated as:

$\sin(\theta)$ = perpendicular/hypotenuse

$\cos(\theta)$ = base/hypotenuse

Note that you should be very careful about the order of the vertex coordinates. It should start from 0 and be continued as 1,2,3.., as shown below:

vertices

(

(0.5 0 0) //0

(1 0 0) //1

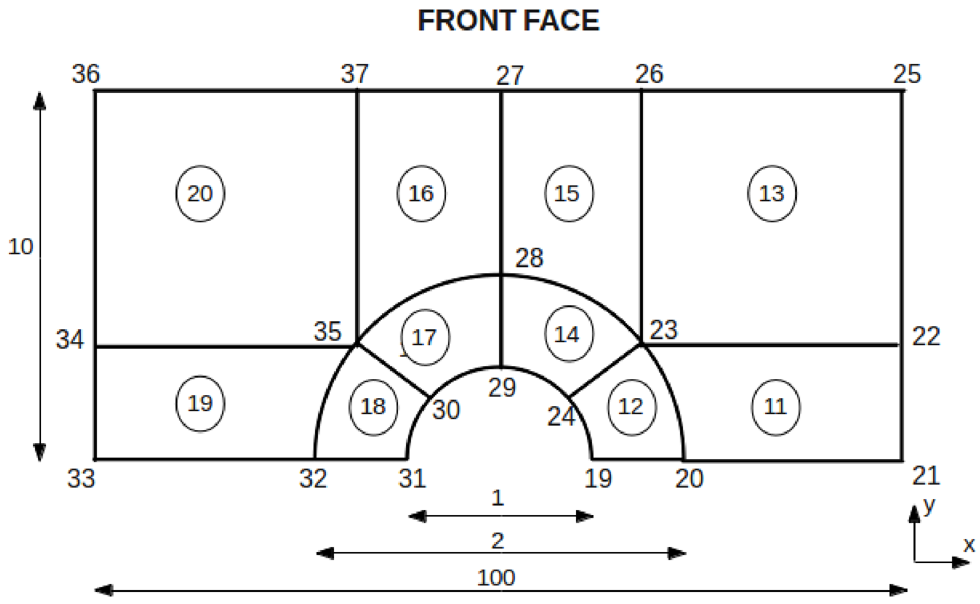


Figure 3.1: Geomtery points in the front face for 2-D flow over a cylinder

.

.

.

.

);

After this enter the details within blocks in a similar manner as shown in the previous chapter. Here in this problem, we have divided the geometry domain into ten small hexahedral blocks having structured body fitted grid points, as shown in fig 3.4. For further details on how to create blocks you may refer to chapter 2.

In the previous chapter we have seen, the geometry domain had staright edges, thereby we had kept the details within edges keyword empty (which is the default condition). But here, in this geometry we have curved edges. In OpenFOAM we can use the following types of edges in blockMeshDict file, as shown in fig 3.5:

For this problem we have used arc edges. Therfore the details within edges can be given as:

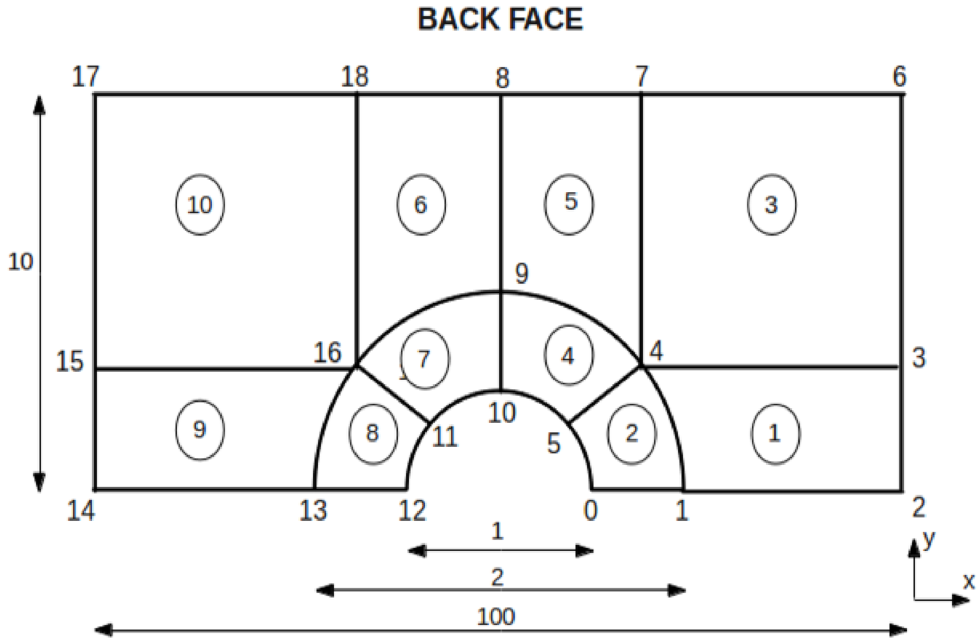


Figure 3.2: Geomtery points in the back face for 2-D flow over a cylinder

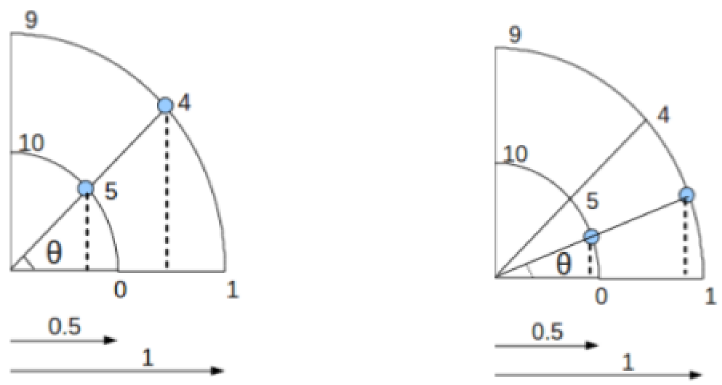


Figure 3.3: Geomtery points for 2-D flow over a curved body

edges

(

```
blocks
(
    hex (5 4 9 10 16 15 20 21) (10 10 1) simpleGrading (1 1 1)
    hex (0 1 4 5 11 12 15 16) (10 10 1) simpleGrading (1 1 1)
    hex (1 2 3 4 12 13 14 15) (20 10 1) simpleGrading (1 1 1)
    hex (4 3 6 7 15 14 17 18) (20 20 1) simpleGrading (1 1 1)
    hex (9 4 7 8 20 15 18 19) (10 20 1) simpleGrading (1 1 1)
);
```

Figure 3.4: Block details for the blockMeshDict file

Keyword selection	Description	Additional entries
arc	Circular arc	Single interpolation point
simpleSpline	Spline curve	List of interpolation points
polyLine	Set of lines	List of interpolation points
polySpline	Set of splines	List of interpolation points
line	Straight line	—

Figure 3.5: Types of edges used in BlockMeshDict dictionary

<keyword> <vertices joining the edge> (interpolation points)

);

The details of edges for our current problem is given in fig 3.6:

```
edges
(
    arc 0 5 (0.469846 0.17101 0)
    arc 5 10 (0.17101 0.469846 0)
    arc 1 4 (0.939693 0.34202 0)
    arc 4 9 (0.34202 0.939693 0)
    arc 11 16 (0.469846 0.17101 0.5)
    arc 16 21 (0.17101 0.469846 0.5)
    arc 12 15 (0.939693 0.34202 0.5)
    arc 15 20 (0.34202 0.939693 0.5)
);
```

Figure 3.6: Edge details for the blockMeshDict file

After this enter the boundary patches under the keyword boundary. You may refer to the previous chapter to get know more details regarding how to write the boundary patches.

Similar to the lid-driven cavity problem , even this geometry does not have any patches to be merged. Therefore we will keep the mergePatchPairs empty.

After completing the blockMeshDict file, you can save it within the required case file in polymesh folder inside constant folder. Thereafter switch back to the command terminal and open the required case file as mentioned in the previous chapter.

After this enter blockMesh in the command terminal and press $< enter >$. Thus you can see geometry is meshed. After this type paraFoam in the command terminal and press<enter>. This opens the ParaView window. Now on the ParaView window press apply on the left hand side of the Object Inspector Menu to view the new geometry.

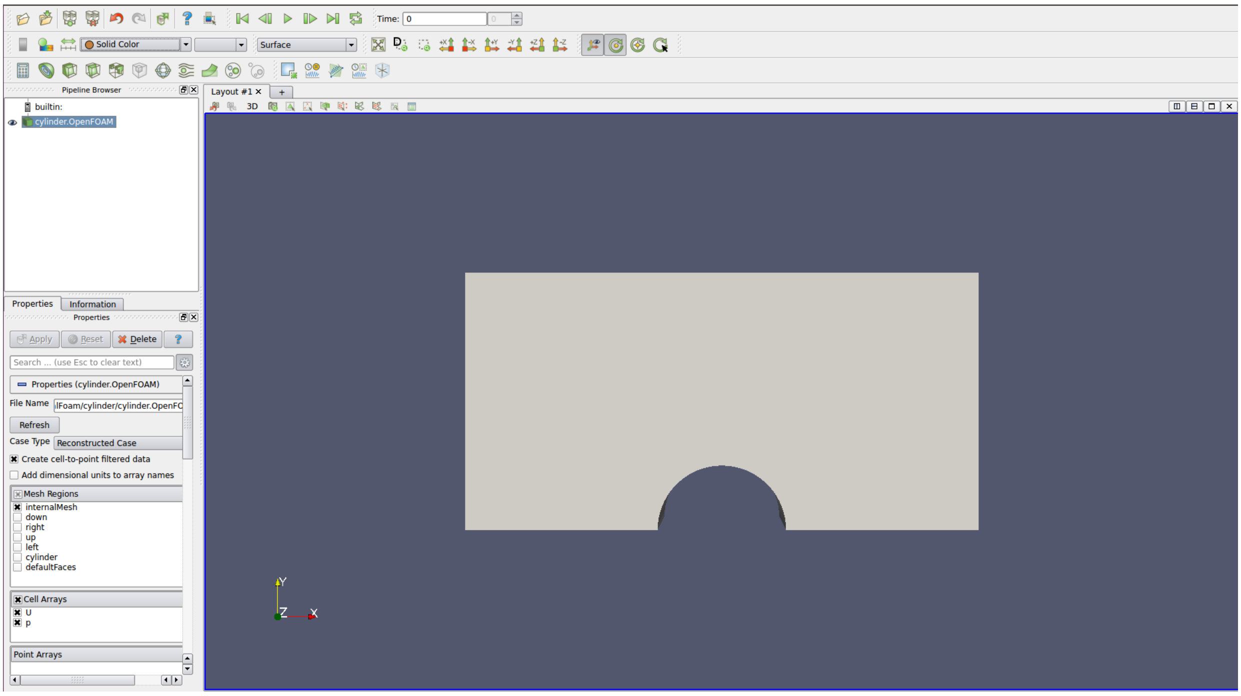


Figure 3.7: ParaView window showing the 2-D geometry

Now in the ParaView window you can check or uncheck the different regions within the mesh region in the Object Inspector Menu to visualize diffrent regions on the geomtery. You can also visualize the geometry in wire-frame instead of surface by changing it from the down-down Active Variable Control Menu.

Thus in this chapter we learned how to create a curved geometry in OpenFOAM and visualize it in different ways using ParaView.

Chapter 4

Simulating flow in a Lid Driven Cavity using OpenFOAM

In this chapter we would learn how to solve a lid driven cavity problem using icoFoam solver and viewing the results in ParaView. As you might recollect from the 2nd chapter we, learned how to create the case file in OpenFoam and write its blockMeshDict file. You may review the previous chapter to see the problem statement for a lid driven cavity. Here in this problem we have a rectangular box of unit thickness with top surface moving with a velocity of 1 m/s and the other three sides fixed, as shown in fig 5.1: Here we have considered a Re of 100.

Now to solve our present problem open acommand terminal by pressing <ctrl>, <Alt> and <T> simulataneously. After this enter the path for the current case file as shown below:

cd run/tutorial/incompressible/icoFoam/cavity

After openning the required case file enter 'ls'. This would show the folders within this case file. Here as mentioned in chapter 2 you would find three folders by the name:

- 0
- constant
- system

Now if you press 'cd 0' ¡enter¿ and then 'ls' ¡enter¿ in the command terminal, you would find two folders given as:

- P
- U

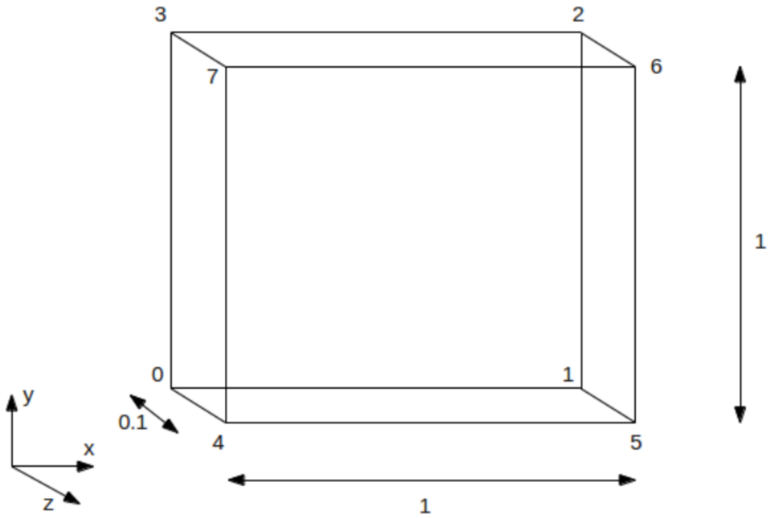


Figure 4.1: Geomtery of 2-D lid driven cavity

These folders gives the initial boundary conditions for pressure (i.e. P), velocity (i.e. U), temperature, etc of the geomtery. After this go back to the cavity folder by typing 'cd ..' <enter>.

Now if you open the constant folder by entering 'cd constant' ¡enter¿ in the command terminal followed by 'ls' ¡enter¿ you would find two folder:

- polyMesh
- transportProperties

Here polyMesh folder contains the blockMeshDict file. You can open this by typing 'cd poymesh' <enter> followed by 'ls' <enter> in the command terminal and then type 'gedit blockMeshDict'. This would show you the blockMeshDict file in text editor which contains the veritces, blocks and boundary patches of the geometry. The tranportProperties contain properties of the fluid medium used in this this problem.

Now you can go back to the cavity folder by entering cd .. twice in the command terminal. After this type 'cd system' ¡enter¿ followed by 'ls' <enter>. This would show you the three folders within system file.

- ControlDict
- fvSchemes

- fvSolution

Here controlDict file contains control parameters for start and stop of the number of iterations, fvSolution contain discretization schemes used for simualtion of this problem and fvSchemes contains equations for solver, tolerance, etc.

To mesh the geometry go back to the cavity folder in the command terminal and enter the following and press <enter>:

blockMesh

This would mesh the geometry as shown in the fig 4.2:

As you can see, we have used course mesh for this problem. If there are some error in the blockMesDict file, it would be shown in the command terminal. You can also type 'checkMesh' <enter> in the command terminal to check the different properties of the meshed geometry like number of cells, skewness, etc.

After this to view the meshed geometry, you can type 'paraFoam' <enter> in the command terminal. This woould open the ParaView window. Now on the ParaView window press apply on the left hand side of the Object Inspector Menu to view the meshed geometry, as shown in fig 4.3. After inspecting the geomety you may close the ParaView window and switch back to the command terminal.

Now for simulating this particular problem we have used icoFoam solver. This is a Transient solver for incompressible, laminar flow of Newtonian fluids. To solve this problem type 'icoFoam' ¡enter¿ in the command terminal. You can see the progressing iterations in the terminal window along with the residual values. After it ends type 'paraFoam' <enter> in the terminal window for post-processing.

This would open the ParaView window. As mentioned previously press apply on the left hand side of the Object Inspector Menu to view the new geometry. After this you can check or uncheck the different regions within the mesh region in the Object Inspector Menu to visualize differnt regions on the geomtery. Now to check the velocity contours select U from the drop-down Active Variable Control Menu, from the visible toolbar. This will show the initial velocity countour of the cavity, as shown in fig 4.4. Along with this you may also select the Toggle Colour Legend from the toolbar to visualize the legend.

Now in the paraView window press the 'play' buttom from the VCR control. This would visualize the changing velocity countour along with the progressing iterations. You can see the final velocity contour as shown in fig 4.5.

Now to validate our result we need to plot the u and v velocity in the domain. To do this click on the 'plot over line' object from the Active Variable Control Menu bar. Now for our validation we need to plot u/U versus l/L in the X and Y axis. To do this select X-axis on the left hand side Object Inspector Menu. This will show you the Pressure and velocity plot along X-direction.

On selecting the Yaxis you will see the velocity along Y-axis.

Now you may save these data by selecting File¿Save As¿ and then give appropriate file name. These data will be saved in .csv format. After this open this file from home folder in libreoffice sheet. In the office sheet you can copy paste U:0 and P:1 on a new sheet and calculate it as U:0/U and P:1/L. And them plot them using chart utility. This would give you the following result.

```
/*---------------------------------------------------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | foam-extend: Open Source CFD                    |
|  \\    /   O peration     | Version:     3.1                                |
|   \\  /    A nd           | Web:         http://www.extend-project.de       |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
Build    : 3.1
Exec     : blockMesh
Date     : Oct 21 2015
Time     : 02:40:03
Host     : subhasree-B85M-DS3H-A
PID      : 24966
CtrlDict : /home/subhasree/foam/foam-extend-3.1/etc/controlDict
Case     : /home/subhasree/foam/foam-extend-3.1/tutorials/incompressible/icoFoam/cavity
nProcs   : 1
SigFpe   : Enabling floating point exception trapping (FOAM_SIGFPE).

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
Create time


Creating block mesh from
    "/home/subhasree/foam/foam-extend-3.1/tutorials/incompressible/icoFoam/cavity/constant/polyMesh/blockMeshDict"

Creating curved edges
Creating topology blocks
Creating topology patches

Creating block mesh topology

Check topology

        Basic statistics
                Number of internal faces : 0
                Number of boundary faces : 6
                Number of defined boundary faces : 6
                Number of undefined boundary faces : 0
        Checking patch -> block consistency

Creating block offsets
Creating merge list .

Creating polyMesh from blockMesh
Creating patches
Creating cells
Creating points with scale 0.1
    Block 0 cell size :
        i : 0.005 .. 0.005
        j : 0.005 .. 0.005
        k : 0.01 .. 0.01


Writing polyMesh
----------------
Mesh Information
----------------
  boundingBox: (0 0 0) (0.1 0.1 0.01)
  nPoints: 882
  nCells: 400
  nFaces: 1640
  nInternalFaces: 760
----------------
Patches
----------------
  patch 0 (start: 760 size: 20) name: movingWall
  patch 1 (start: 780 size: 60) name: fixedWalls
  patch 2 (start: 840 size: 800) name: frontAndBack

End
```

Figure 4.2: Meshing of the 2-D geometry in OpenFOAM

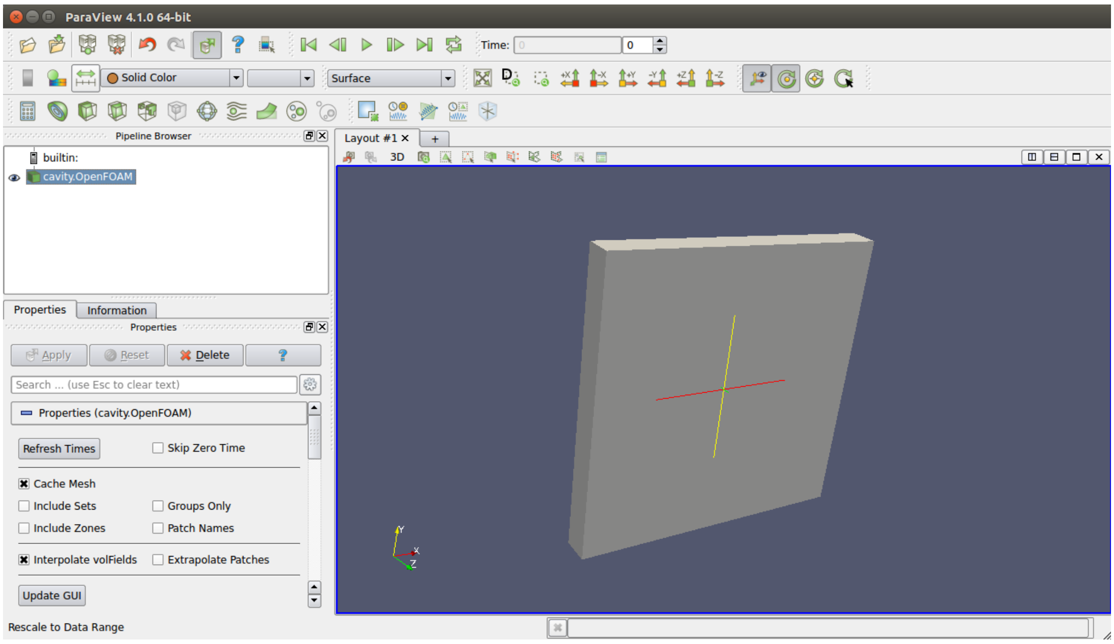


Figure 4.3: Visualization of the 2-D geometry in ParaView

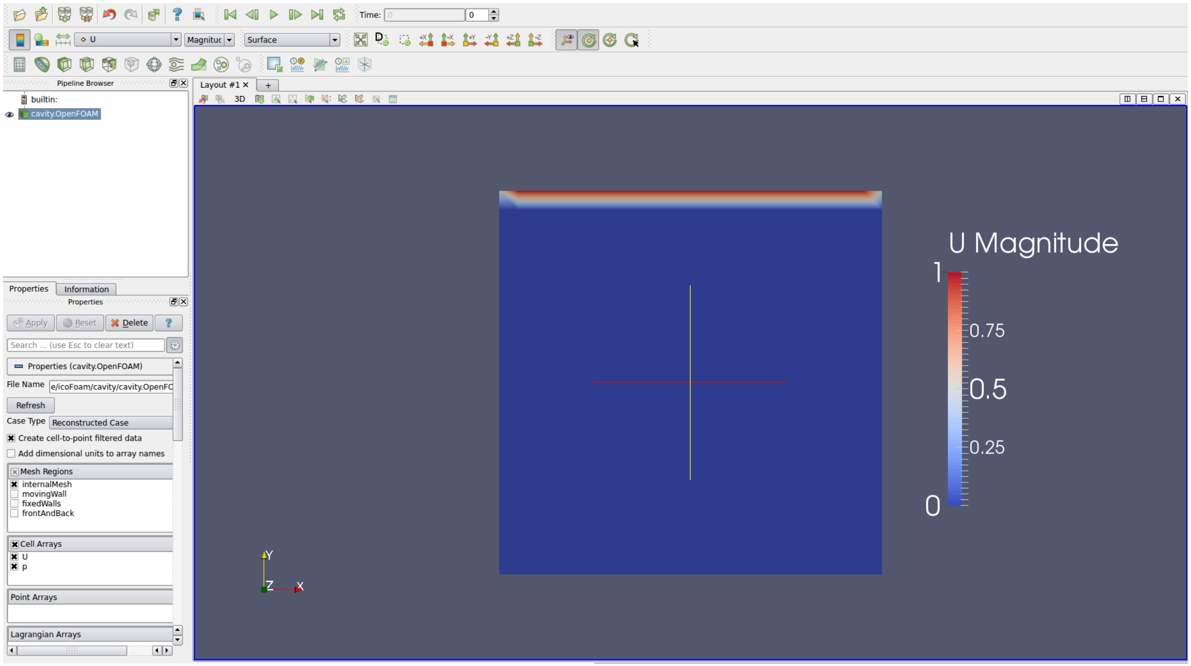


Figure 4.4: Velocity contour in the 2-D geometry at time 0 sec in ParaView

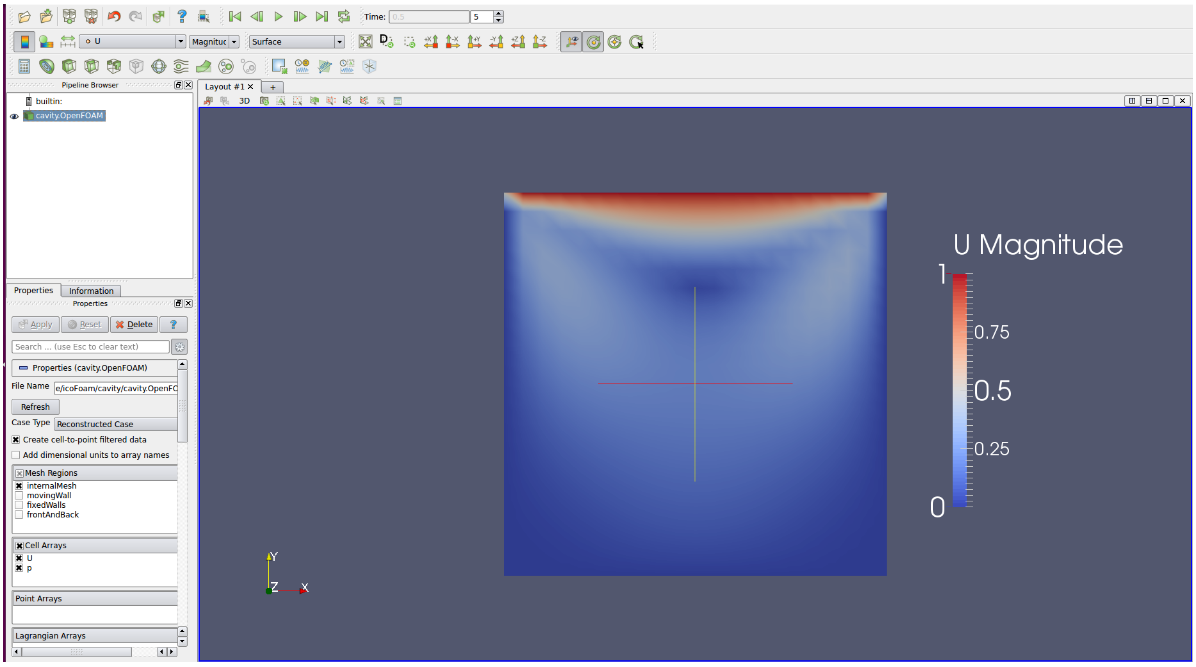


Figure 4.5: Velocity contour in the 2-D geometry at final state in ParaView

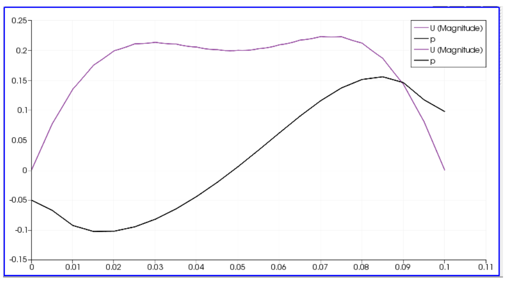


Figure 4.6: Velocity contour in the 2-D geometry at final state in ParaView

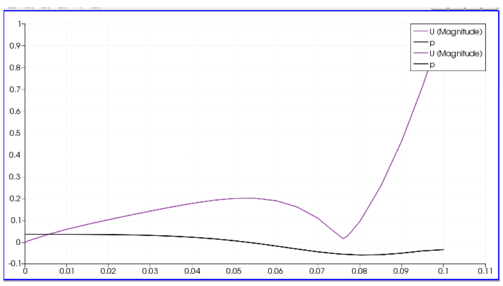


Figure 4.7: Velocity contour in the 2-D geometry at final state in ParaView

Chapter 5

Supersonic flow over a wedge using OpenFOAM

In this chapter we would learn how to simulate a supersonic flow over a wedge and post-process it using ParaView. Here the domain for simulation is given as shown in fig 5.1:

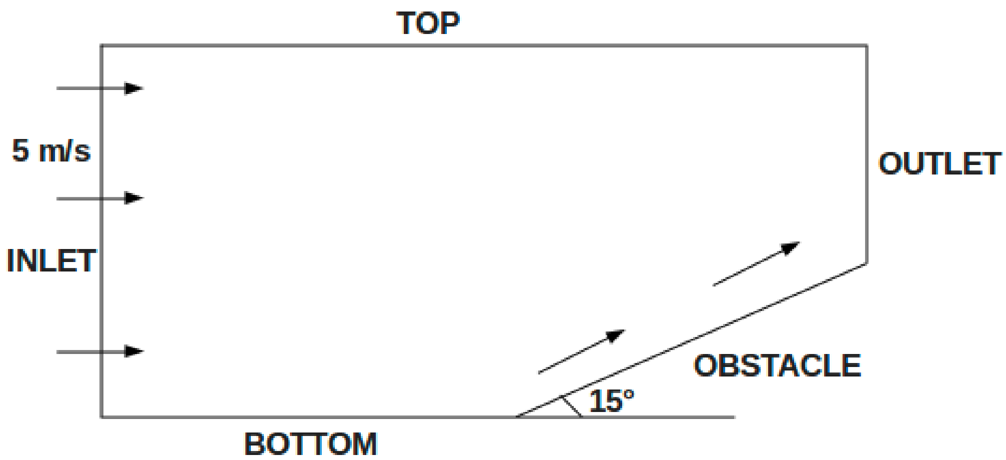


Figure 5.1: Geomtery of 2-D wedge along with boundary conditions

Please note that this chapter in written assuming the readers have some prior know on compressible flows and gas dynamics.

As shown in the figure, in this problem statement we have a wedge at 15 degree with the horizontal and a flow from the inlet at 5 m/s. The boundary conditions used in the geometry file are similar to that as shown in the figure. Now this test case is already pready in the tutorial directory of OpenFoam. Thereby in this chapter we would mainly focus on how to simulate compressible flow over a given test case rather than creating a geometry.

Now to solve our present problem open acommand terminal by pressing <ctrl>, <Alt> and <T> simulataneously. After this enter the path for the current case file as shown below:

cd run/tutorial/compressible/rhoCentralFoam/wedge15Ma5

After openning the required case file enter 'ls'. This would show the folders within this case file. Here as mentioned in chapter 2 you would find three folders by the name:

- 0
- constant
- system

Now if you press 'cd 0' ¡enter¿ and then 'ls' <enter> in the command terminal, you would find two folders given as:

- P
- U
- T

These folders gives the initial boundary conditions for pressure (i.e. P), velocity (i.e. U), temperature (i.e. T), etc of the geomtery. After this go back to the wedge folder by typing 'cd ..' <enter>.

Now if you open the constant folder by entering 'cd constant' <enter> in the command terminal followed by 'ls' ¡enter¿ you would find two folder:

- polyMesh
- transportProperties

Here polyMesh folder contains the blockMeshDict file. You can open this by typing 'cd poymesh' <enter> followed by 'ls' <enter> in the command terminal and then type 'gedit blockMeshDict'. This would show you the blockMeshDict file in text editor which contains the veritces, blocks and boundary patches of the geometry. The tranportProperties contain properties of the fluid medium used in this this problem.

Now you can go back to the wedge folder by entering cd .. twice in the command terminal. After this type 'cd system' ¡enter¿ followed by 'ls' <enter>. This would show you the three folders within system file.

- ControlDict

- fvSchemes
- fvSolution

Here controlDict file contains control parameters for start and stop of the number of iterations, fvSolution contain discretization schemes used for simualtion of this problem and fvSchemes contains equations for solver, tolerance, etc.

To mesh the geometry go back to the cavity folder in the command terminal and enter the following and press <enter>:

blockMesh

This would mesh the geometry in a similar manner as shown in the previous chapter.If there are some error in the blockMesDict file, it would be shown in the command terminal. You can also type 'checkMesh' ¡enter¿ in the command terminal to check the different properties of the meshed geometry like number of cells, skewness, etc.

After this to view the meshed geometry, you can type 'paraFoam' <enter> in the command terminal. This would open the ParaView window. Now on the ParaView window press apply on the left hand side of the Object Inspector Menu to view the meshed geometry.

Now in the ParaView window you can check or uncheck the different regions within the 'mesh region' in the Object Inspector Menu to visualize differnt regions on the geomtery. You can also visualize the geometry in wire-frame instead of surface by changing it from the down-down Active Variable Control Menu, to check out the quality of mesh.After inspecting the geomety you may close the ParaView window and switch back to the command terminal.

Now to run the solver switch back to the command terminal and type 'rhoCentralFoam' ¡enter¿. The solver 'rhoCentralFoam' is a density-based compressible flow solver based on central-upwind schemes of Kurganov and Tadmor. You can see the progressing iterations in the terminal window along with the residual values. After the iteration ends type 'paraFoam' ¡enter¿ in the terminal window for post-processing.

This would open the ParaView window. As mentioned previously press apply on the left hand side of the Object Inspector Menu to view the new geometry. After this you can check or uncheck the different regions within the mesh region in the Object Inspector Menu to visualize differnt regions on the geomtery. Now to check the velocity contours select U from the drop-down Active Variable Control Menu, from the visible toolbar. This will show the initial velocity countour of the cavity, as shown in fig 5.4. Along with this you may also select the Toggle Colour Legend from the toolbar to visualize the legend.

Now in the paraView window press the 'play' buttom from the VCR control. This would visualize the changing velocity countour along with the progressing iterations. You can see the final velocity contour as shown in fig 7.7. Now on the ParaView window press apply on the left hand side of the Object Inspector Menu to view the meshed geometry.

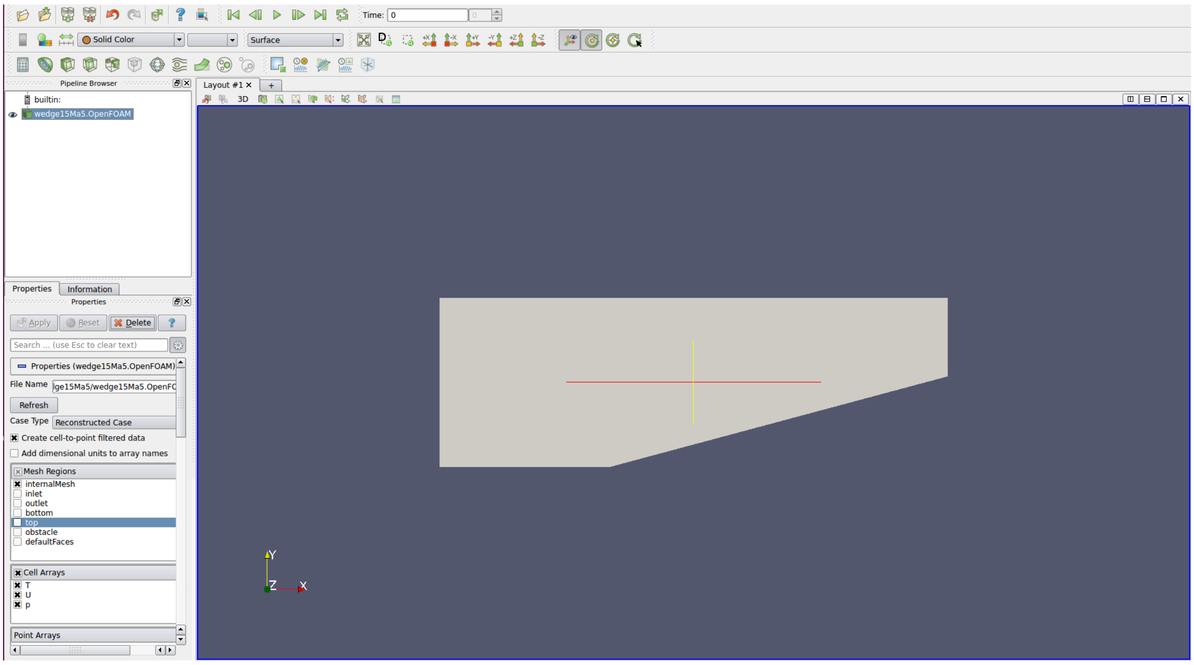


Figure 5.2: Surface visualization of the 2-D wedge domain in ParaFView

Similarly, you can also plot the pressure and temperature contours.

Now we can also calculate the mach number in this flow using a utility function in OpenFoam. To do this close the paraView window and switch to the command terminal. Here type 'Mach' and press <enter>. Note that M is capital here. This would calculate the Mach number in the flow for every time step.

Now to view the Mach number contour type 'paraFoam' in the command terminal and press <enter>. Here select Mach in the left hand side properties window to include Mach number for post-processing. Now select 'Ma' from the drop-down Active Variable Control Menu, from the visible toolbar. And then press the play button on the VCR control toolbar. Along with this you may also select the Toggle Colour Legend from the toolbar to visualize the values to Mach number.

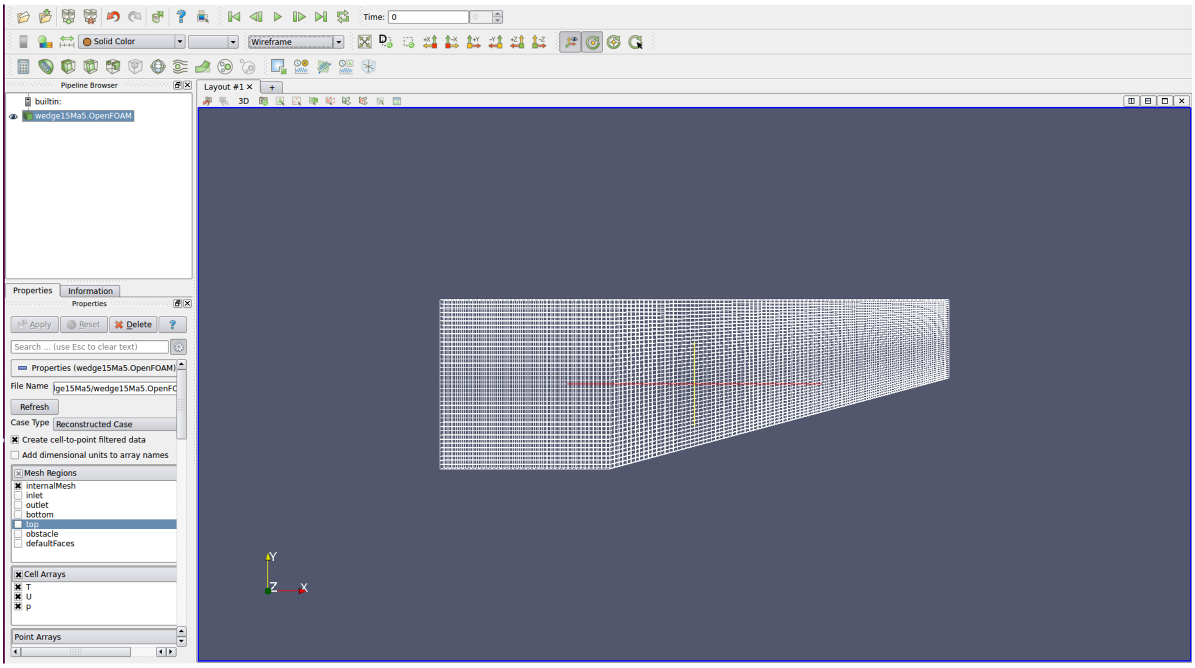


Figure 5.3: Wireframe visualization of the 2-D wedge domain in ParaFView

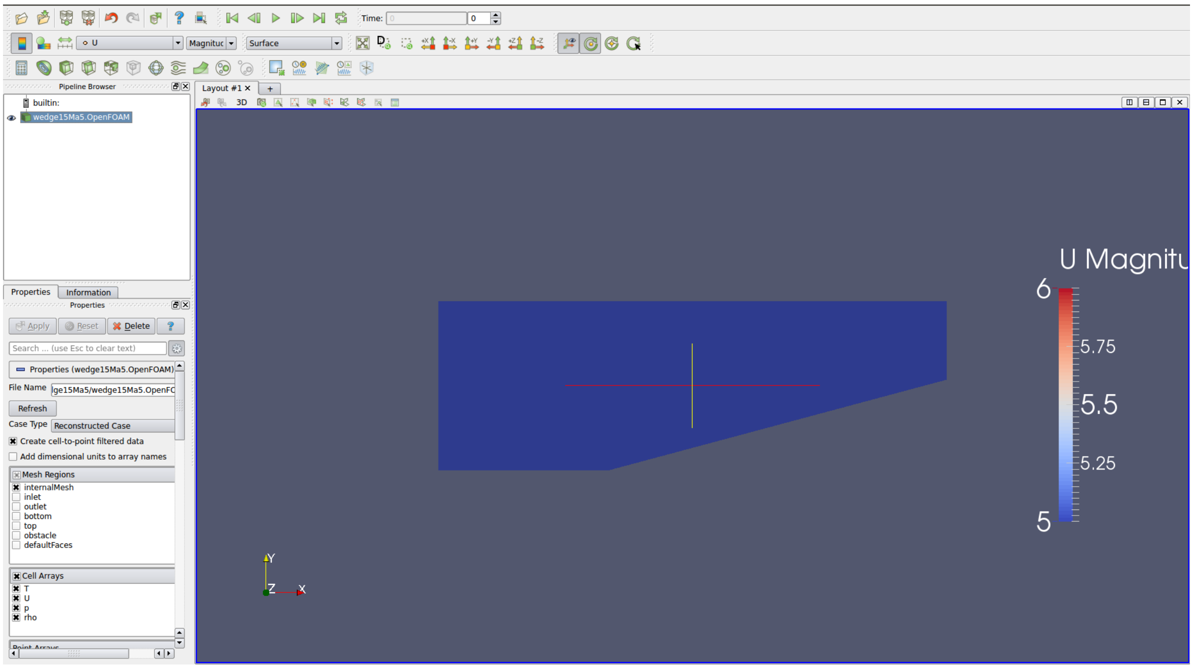


Figure 5.4: Velocity contour in the 2-D wedge domain at initial state in ParaFView

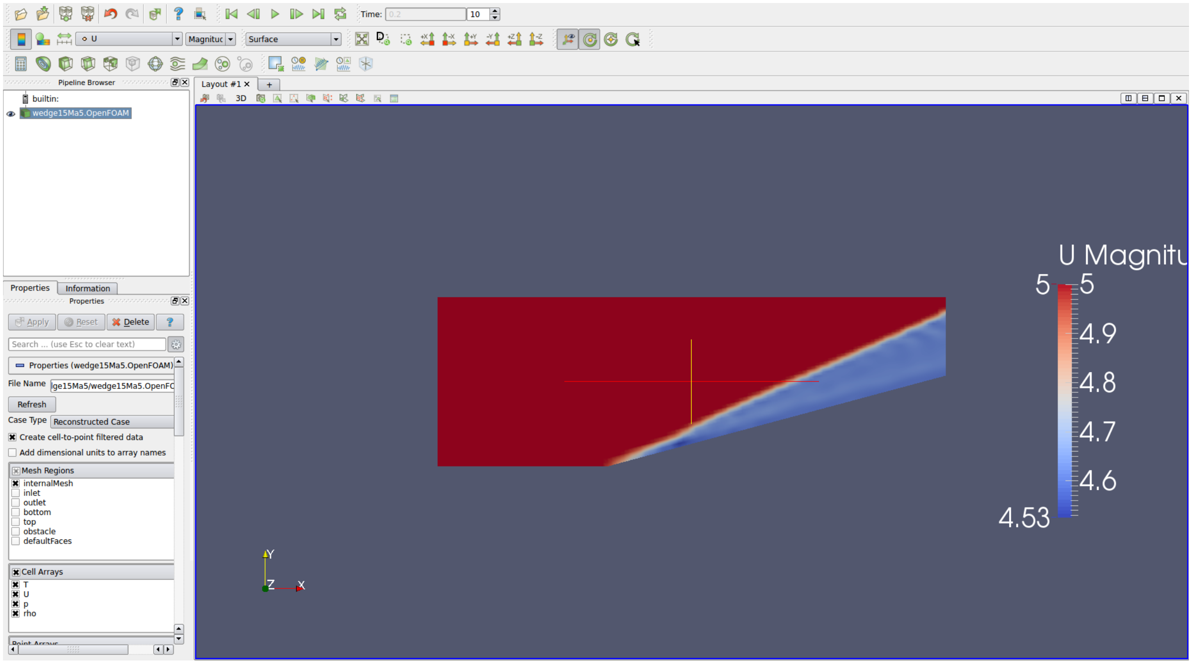


Figure 5.5: Velocity contour in the 2-D wedge domain at final state in ParaFView

Chapter 6

Downloading and Installing Salome

Salome is a Free and Open Source CAD (Computer Aided Drawing), Meshing and Visualization Software for Numerical simulation. We can Create/modify, import/export (IGES, STEP, BREP), repair/clean CAD models and Mesh CAD models, edit mesh, check mesh quality, import/export mesh (MED, UNV, DAT, STL) using Salome. In this chapter we will learn how to download and intall Salome in any Operating system.

6.1 Download Salome

Open your browser and in the address bar type the url given below,

www.salome-platform.org

To Download Salome the user needs to create a account on the salome site. To do this on the left hand side of the salome screen website scroll down to the bottom of the **Navigation** bar, Fig 6.1, where you can see the new user option. Click on it and enter the required personal details.

After you enter the details click on the register button at the bottom as shown in, Fig 6.2. Once done you will be directed to a screen showing that you have been registered. This also states thatv once you have done with registration you have to login to your email. Now open the mail sent by Salome and click on the link shown in Fig, 6.3. This link will direct you to a window where you need to set your password for your Salome account. Enter the password and confirm it and press set my password button, Fig 6.4. After this it will direct you to a window which says your password has been set successfully. You may now login with your username and password.

In the Navigation bar click on Downloads after which you will be directed to a page which will show various bianaries for various Linux distributions. You can choose according to your Operating System and 32/64 bit size. Since in this book we are working on a 64 bit platform we will download Linux Debian 7 64-bits binary, Fig 6.5. Click on it and Save the file. Downloading may take some time due to the large file size. After this scroll down to Universal Binaries and click on the **Linux 64-bits** to download it. Note that 32-bit version of binaries are no more supported for the latest version of Salome.

6.2 Installation

Create a new folder in your home directory by the name Salome. Once these files are downloaded go to your Downloads folder and copy the tar file and a Self Extracting file and paste this inside your Salome folder. To extract the tar file right click on the tar file and select extract here. We can now see the extract wizard folder.

Open a new terminal window and type in the path for the Installation Wizard folder in the Salome folder of your home directory.

cd /home/Salome/InstallationWizard_7.6.0_Debain_64bits

Type ls to view the content inside the file. An executable file by the name **runInstall** can be seen here. The Installation Wizard can be launched in two modes : GUI and Batch. The default installation settings can be overridden by using command line options. Each option has a short and a long notation. In the command line type :

./runInstall [options]

Options include

- -g / –gui : Runs the Installation Wizard in the GUI mode (this is the default mode).
- -b / –batch : Runs the Installation Wizard in the terminal mode.

Once the user has finished installation by either the GUI or Batch mode we now need to install the universal library. Since the binary is available in the same folder in the command terminal we can type :

./Salome-V7_6_0-LGPL-x86-64.run and press enter

Enter the default path for the installation file. After this type N to use Salome in English.The installation prcocess now starts and might take a while. Close the terminal once this is done. Now you can see the Salome icon on your Desktop, Fig 6.7. Double click on this to open the Salome working window, Fig 6.8. This brings us to the end of the chapter. In the next chapter we will learn more about how to create Geometries using Salome and using it for our OpenFOAM Simulation.

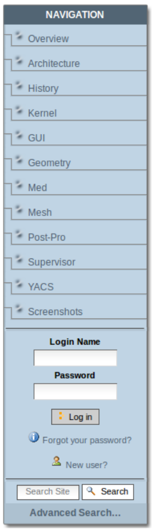


Figure 6.1: Navigation Bar

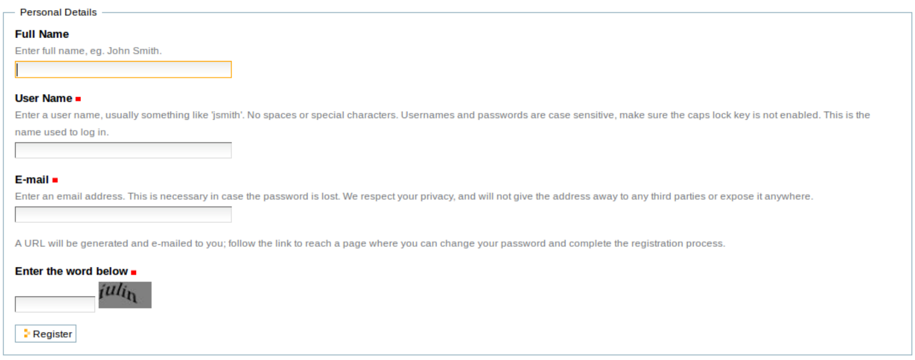
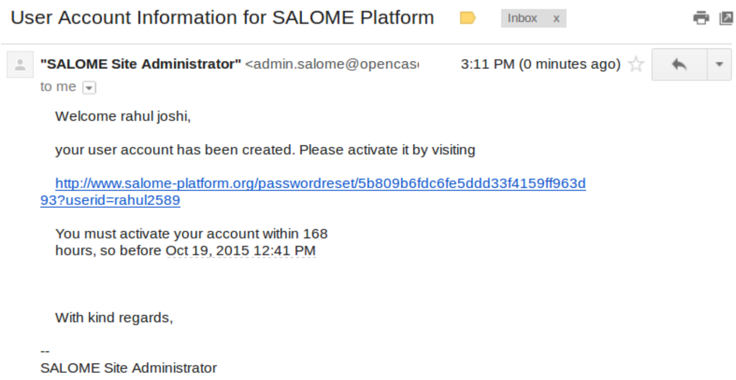


Figure 6.2: User Details


User Account Information for SALOME Platform Inbox x

"SALOME Site Administrator" <admin.salome@opencas 3:11 PM (0 minutes ago)

to me

Welcome rahul joshi,

your user account has been created. Please activate it by visiting

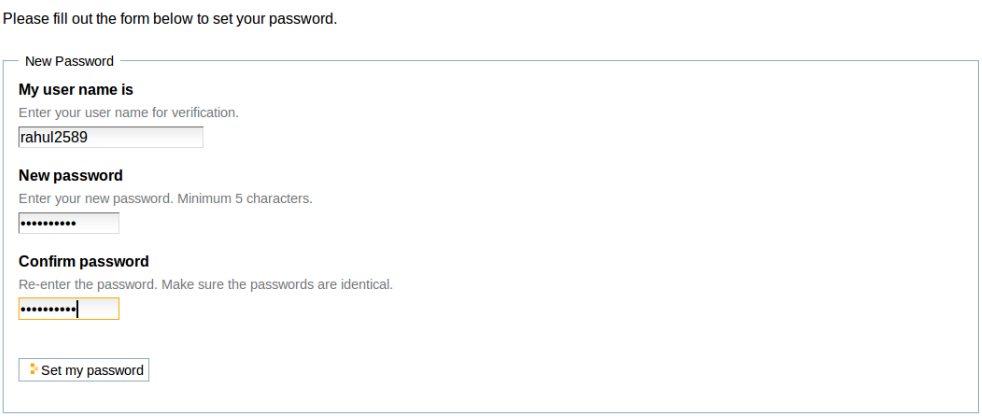
http://www.salome-platform.org/passwordreset/5b809b6fdc6fe5ddd33f4159ff963d93?userid=rahul2589

You must activate your account within 168 hours, so before Oct 19, 2015 12:41 PM

With kind regards,

--
SALOME Site Administrator

Figure 6.3: Salome Link


Please fill out the form below to set your password.

New Password

My user name is

Enter your user name for verification.

rahul2589

New password

Enter your new password. Minimum 5 characters.

••••••••••

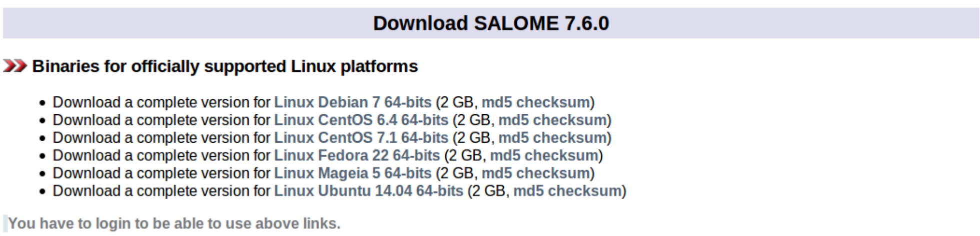
Confirm password

Re-enter the password. Make sure the passwords are identical.

••••••••••

Set my password

Figure 6.4: Enter Password


Download SALOME 7.6.0

Binaries for officially supported Linux platforms

- Download a complete version for **Linux Debian 7 64-bits** (2 GB, **md5 checksum**)
- Download a complete version for **Linux CentOS 6.4 64-bits** (2 GB, **md5 checksum**)
- Download a complete version for **Linux CentOS 7.1 64-bits** (2 GB, **md5 checksum**)
- Download a complete version for **Linux Fedora 22 64-bits** (2 GB, **md5 checksum**)
- Download a complete version for **Linux Mageia 5 64-bits** (2 GB, **md5 checksum**)
- Download a complete version for **Linux Ubuntu 14.04 64-bits** (2 GB, **md5 checksum**)

You have to login to be able to use above links.

Figure 6.5: Salome Linux 7 64 bit binary

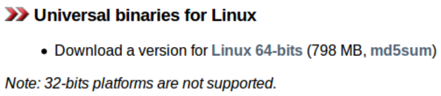


Figure 6.6: Universal Binaries

Figure 6.7: Salome icon

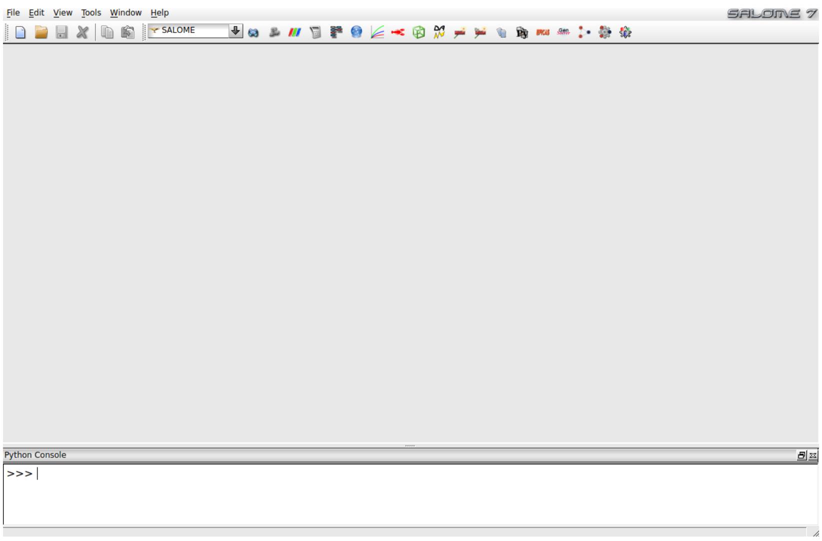


Figure 6.8: Salome working window

Chapter 7

Importing Mesh From Third Party Software in OpenFOAM

OpenFOAM can be used for creating and meshing geometrical shapes like Box, Pipe. When dealing with complex geometries like a turbine blade, aircraft, ship etc, we cannot use the blockMesh utility. In such cases it is always better to create the geometry and mesh in dedicated CAD and Meshing softwares and solve those usiing OpenFOAM. As a prerequisite it is expected the user should have knowledge about creating geometry and generating mesh in softwares like Gmabit, Gmsh, Salome, ICEM etc. This chapter deals with the steps involved in importing mesh files in OpenFOAM using different mesh conversion tools.

7.1 Geometry

We will use the above problem of Flow over a square cylinder as an example for importing mesh file in OpenFOAM. Here we have a square cylinder of length 1m and height 1 m. Inlet velocity is set at 1 $\frac{m}{s}$ for Reynolds number (Re) 100. The size of the domain choosen is 60 m by 40 m. The boundary conditions are as shown in the , Fig 7.1 below.

7.2 Meshing

We have generated a hexhedral mesh for the above geometry with 40000 cells and saved the mesh file as cylmesh.msh. The mesh generated is as shown below, Fig 7.3

7.3 Importing the mesh file

In incompressibel solvers go to icoFoam and create a solver inside it by the name **cylinder**. Now go inside the cavity case and copy the

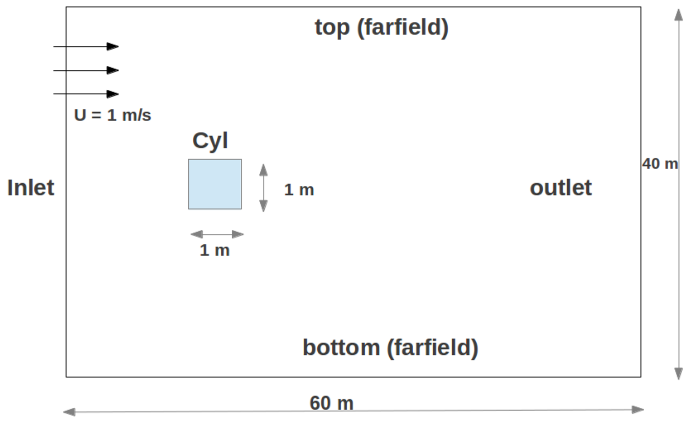


Figure 7.1: Flow over square Cylinder

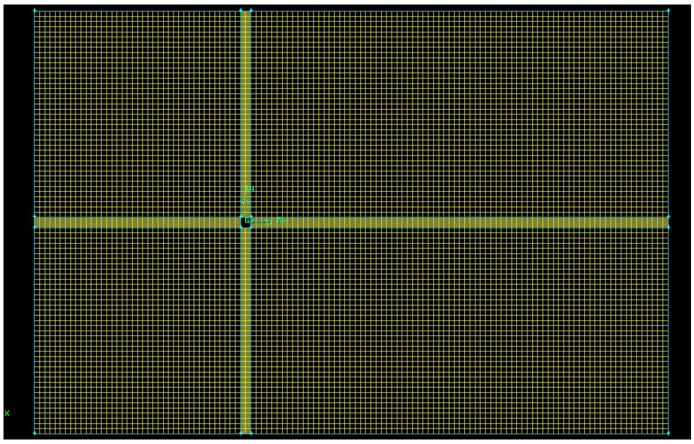


Figure 7.2: Mesh

- 0
- system

folder and paste it inside the cylinder folder. Please make a not here that we do not need the **constant** folder here. After this copy the cylmesh.msh mesh file create earlier and paste this inside this folder. Thus the our case file is now ready. Now open the command terminal and type the path for the cylinder folder. Now since we have a Fluent (.msh) mesh file we will use the mesh conversion command as shown below followed by the file name

fluentMeshToFoam file-name.msh

In the terminal window type the above command with the file name and press enter.

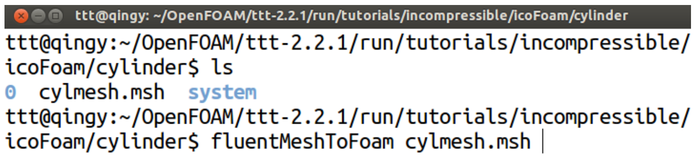


Figure 7.3: convert

In case you have a 3D mesh file then you can use the command

fluent3DMeshToFoam file-name.msh

The Fluent mesh file is converted into OpenFOAM mesh file. Now if we look back into our cylinder folder we can see that the "constant" folder is now generated. When we open the constant folder we will see that the transport properties file is missing. Since we had converted the fluent mesh file into openfoam the fluid property files were missing. Copy the transport property file from the constant folder of cavity case and paste this inside the constant folder of cylinder. The trasnportProperties file contains the value of fluid viscosity, we can either change it or keep it default.

Make a note here that we do not use the **blockMesh** command here

7.4 Boundary Conditions

When we import the geometry in OpenFOAM we need to be very careful with the boudnary names used while creating the mesh file. Since OpenFOAM is case sensitive in case of any mistake with the boundary names can create an error while running the solver. To view the boundary names in the command terminal go to polyMesh folder inside the constant. Inside polyMesh you can see a file by the name **boundary**. Open this file in any editor of your choice, eg, gedit boundary, Fig 7.4.

The boundary names will be as shown in the domain shown above, Fig 7.1. In case of any error with the boundary names you can always refer to this boundary file. Now in your command terminal go to the 0 folder and open the pressure file. Make sure that the boundary names match exactly the names in the boundary file, in case of errors make the necessary changes.

```
ttt@qingy: ~/OpenFOAM/ttt-2.2.1/run/tutorials/incompressible/icoFoam/cylinder/constant/polyMesh
ttt@qingy:~/OpenFOAM/ttt-2.2.1/run/tutorials/incompressible/icoFoam/cylind
er/constant/polyMesh$ ls
boundary   faces      neighbour  points
cellZones  faceZones  owner      pointZones
ttt@qingy:~/OpenFOAM/ttt-2.2.1/run/tutorials/incompressible/icoFoam/cylind
er/constant/polyMesh$ gedit boundary
```

Figure 7.4: Boundary file

7.5 Solver settings

In the terminal window go to the controlDict file inside system and open it in any editor of your choice. Change the endTime from 0.5 to 1.5 seconds. Save the file and close it, Fig 7.5 and come back to the cylinder folder.

```
application     icoFoam;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime         1.5;

deltaT          0.005;

writeControl    timeStep;

writeInterval   20;

purgeWrite      0;
```

Figure 7.5: controlDict file

After making the necessary changes we can now run the solver. In the temrinal window type the name of the solver **icoFoam** and press enter. The iterations will be seen running on the terminal window. After the iterations stop we can now start with the visualization.

7.6 Post-Processing

Launch paraview by typing **paraFoam** in the terminal window and once it opens click on the Apply button to view the geometry, Fig 7.6. In the active varialble control menu change from Solid Color to Velocity (U). You can now see the initial conditions for velocity, Fig 7.7. To view the animation on the right hand top of paraview click on the play button of VCR menu. You can see the change in velocity in the paraview window with the passage of time, Fig 7.8.

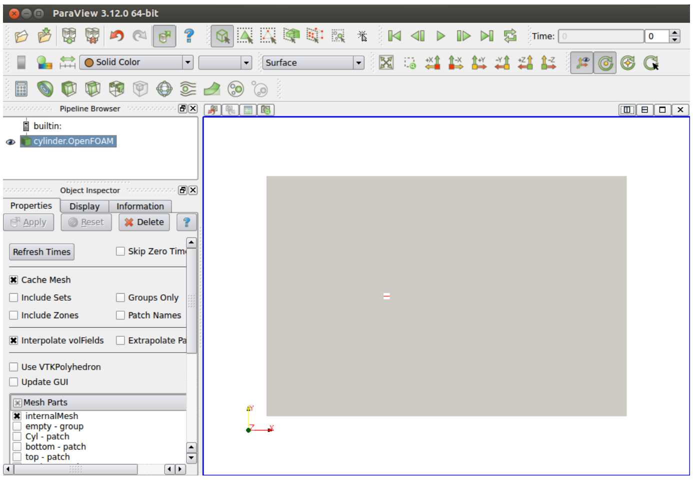


Figure 7.6: Geometry in Paraview

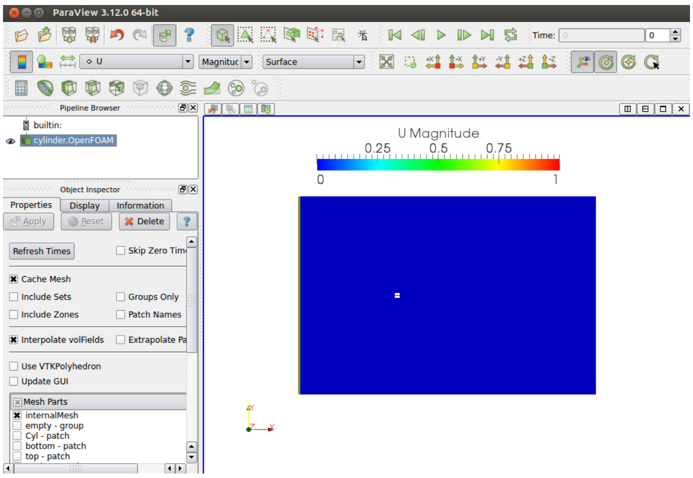


Figure 7.7: Initial velocity condition

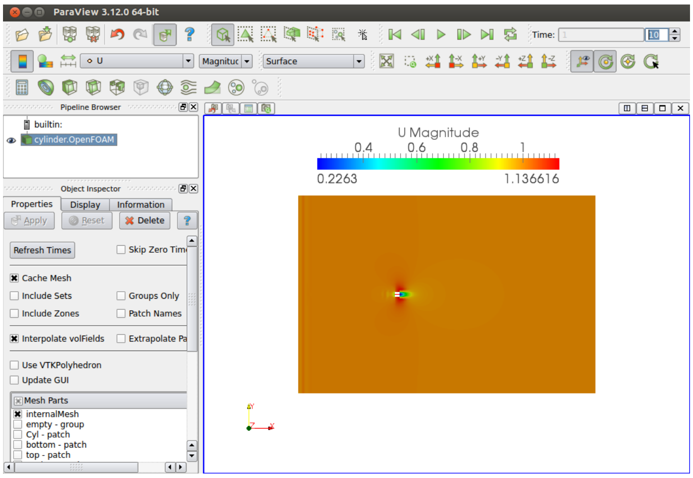


Figure 7.8: Velocity at 1 sec

7.7 Mesh Conversion Commands

The user can also import mesh files from other meshing softwares as well. Here is a list of commands to import mesh files in OpenFOAM.

- ANSYS : ansysToFoam file-name
- IDEAS : ideasToFoam file-name
- CFX : cfxToFoam file-name
- SALOME : ideasUnvToFoam file-name

Chapter 8

Installing and Running Gmsh

Gmsh is a Free and Open Source three dimensional finite element grid generator with a build-in CAD engine and post- processor. There are four modules available in Gmsh such as Geometry, Meshing, Solver and Post-Processiing. Using Gmsh we can mesh the geometry and import it in OpenFOAM using the mesh conversion utilities (see chapter 17 for more info). In this chapter we will cover how to install Gmsh and create a simple geometry.It is expected that the user should have knowledge about Meshing.

8.1 Installing Gmsh

Gmsh can be installed using Synaptic Package Manager. Open Gmsh in your system by typing your system password. In the search box type Gmsh and install it, Fig 8.1. This might take some time depending on your internet speed.

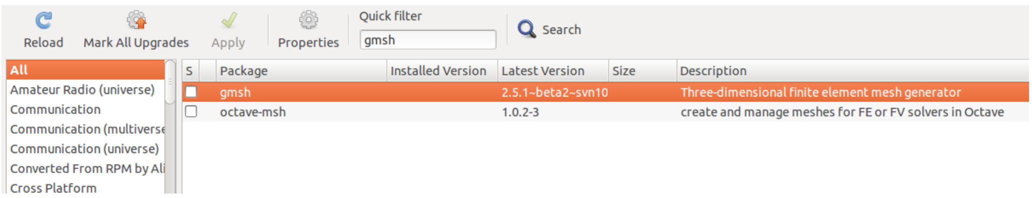


Figure 8.1: Install Gmsh

Alternately we can also install Gmsh from the gmsh website given below,

http://geuz.org/gmsh/

Open this website in your browser and scroll down to download. Now Download Gmsh according to the given current stable release Fig 8.2 according to your Operating System (OS).

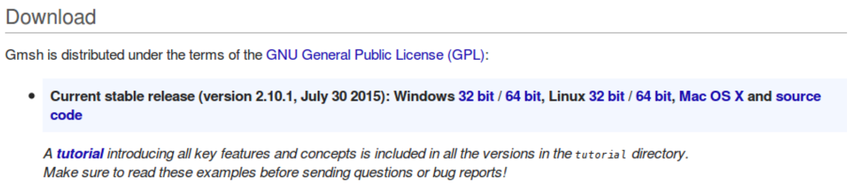


Figure 8.2: Download stable release

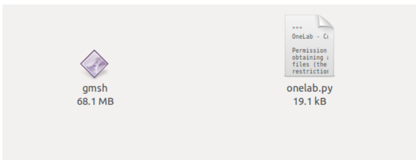


Figure 8.3: gmsh-icon

8.2 Running Gmsh

In the Download folder extract the downloaded gmsh tar file. After you open the folder you will see folder named bin, click on it. Inide the bin folder you will see the Gmsh icon, Fig 8.3. Double click on it to launch the Gmsh Start screen, Fig 8.4

As a pracice to learn Gmsh we will create a cube of sides 1 unit as seen in the Fig, 8.5. On the left hand side in the Gmsh window you can see three modules namely,

- Geometry
- Mesh
- Solver

Click on the Geometry module, then go to Elementary Entities, inside elementary entities go to add and then click on points. This will open up a window where you can enter the X, Y and Z co-ordinates starting with 0 inside each box and press Enter, Fig 8.6. Now enter points for all the remaining 7 vertices to complete the cube, Fig 8.5. In the Gmsh screen we can see the eight points, you can move those points using the left mouse click. To join these points click on Straight-line option under Elementary Entities. Now select

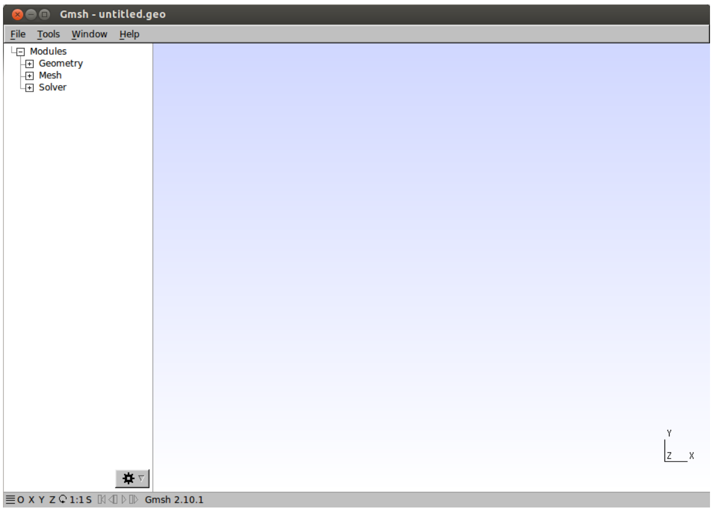


Figure 8.4: Gmsh Start window

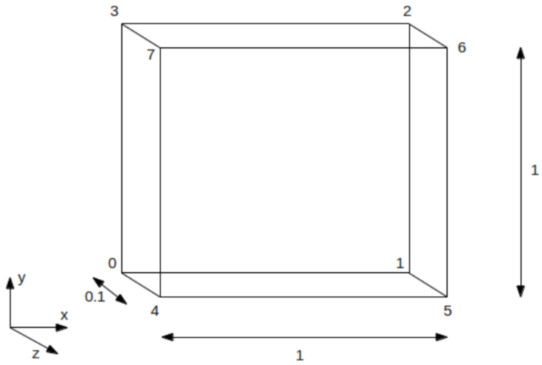


Figure 8.5: Geometry for Gmsh

any two points to create a straight line, click on the start point and then the second point to create a line. Similarly join all the other points to create a cube as shown in the Fig, 8.7 below. As you can see on the Gmsh screen you can press e to end selection and q to abort.

8.2.1 Create Faces

To create faces for the cube click on plane-surface unde elementery enetities. After this select the outer booundaries of the face of a rectangle. Select the edges of the bottom face

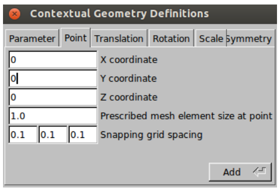


Figure 8.6: Points window

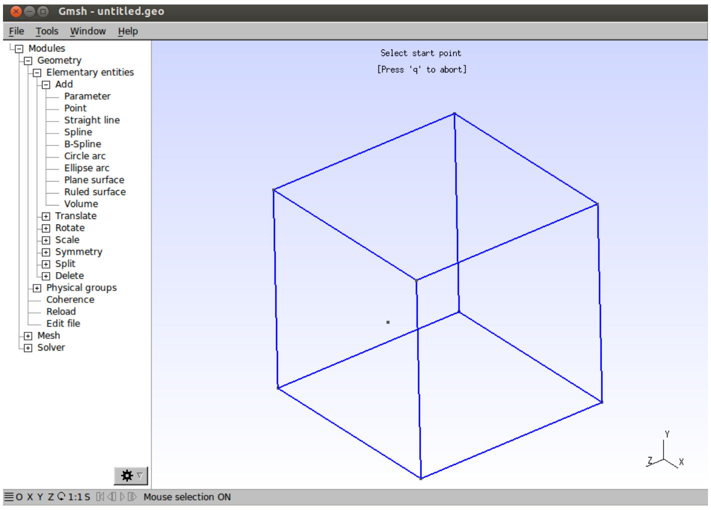


Figure 8.7: Join points using line

first.Once you select the edges they will turn red in color, Fig 8.8. Check in case if there is any hole in the face, if none then press e to end selection. You will notice that a face will appear with dasshed center lines, Fig 8.9. Repeat this procedure for remaining faces, Fig 8.10 and finally press q to abort.

8.2.2 Creating Volume

We now need to create volume boundary. We need to select the Volume boundary similar to selecting boundary for faces. Click on the Volume boundary under elementery entities and click on boundary surface of the cube and press e to end selection. A yellow dot will appear at the center of the cube which represents volume in Gmsh. Press q to abort the selction.

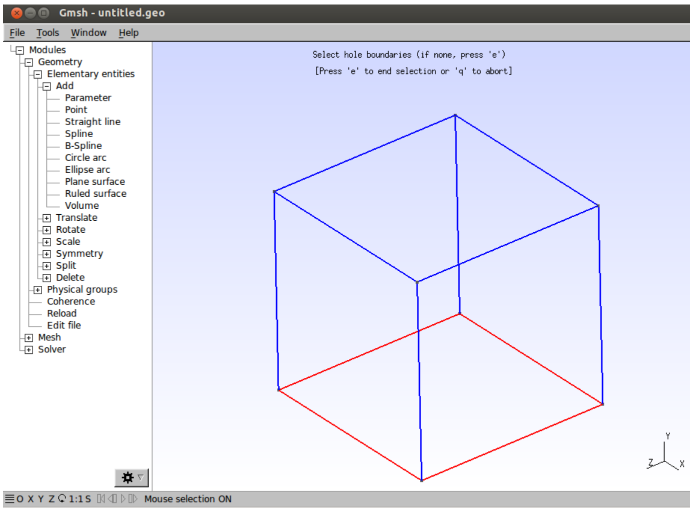


Figure 8.8: Selct edges

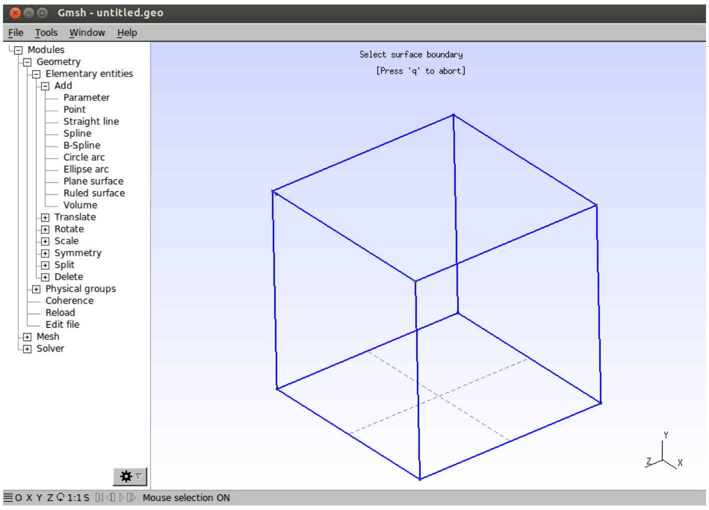


Figure 8.9: Bottom Face

8.2.3 Physical Groups

Physical groups heps us to identify a set of points, lines, faces, volume with a unique Identification number. We create physical groups which will be useful for exporting the Mesh file to OpenFOAM.To do so click on Physical Group under Geometry Module. Click

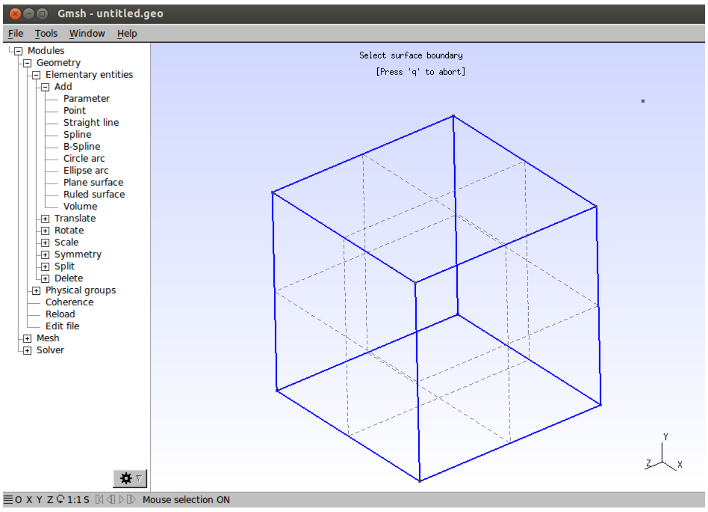


Figure 8.10: Create faces for all surfaces

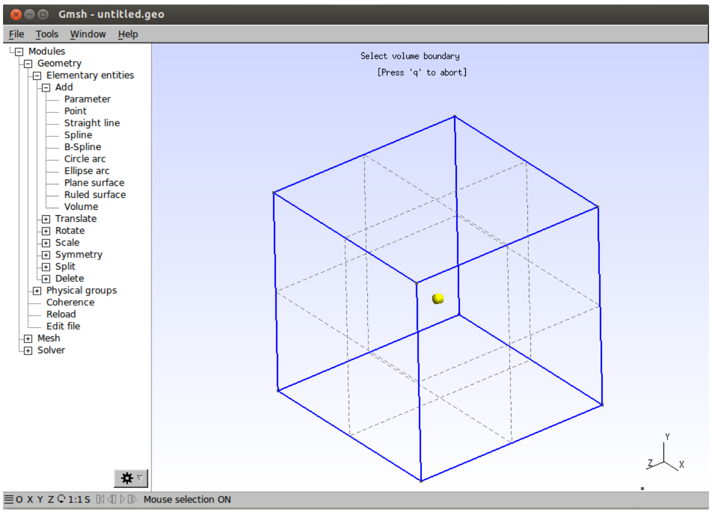


Figure 8.11: Volume

on Add and then Surface. Upon selection of any face it will turn red. Now press e to end selection. Do this procedure for all the remaining facesand press q to abort. Also we need to select the Physical Volume. Click on Volume under Physical Groups and select the yellow dot at the center of the cube.The yellow dot will turn red in colour adn press e to end selection and q to abort.

To save the geometry under the file menu click on Save as and save the geometry by the name cube.geo. Here ”geo” stands for geometry. Click OK twice to save the geometry.

Chapter 9

Creating a Sphere using Gmsh

In the earlier chapter we learnt about how to Install and Run Gmsh by creating a simple geometry. As the earlier tutorial shows about basic geometry construction in Gmsh the user is expected to know it before starting this chapter. This chapter deals with creating a Sphere using Gmsh and meshing it. The chapter we will focus on creating the spherical geometry and the domain surrounding it and in the next chapter we will look into how to mesh this geometry. Since this is a spherical geometry we will learn about how to create a circular arc, how to create ruled surface and doing basic manipulation with the .geo file which generated.

9.1 Points

You can start Gmsh by either double clicking on the gmsh-icon or from the terminal by typing **gmsh sphere1.geo** , this will open up the gmsh window. The first step after starting Gmsh is to mark the co-ordinates for our sphere, we define the center of the Sphere and points surrounding it. Co-ordinates for the sphere (7 points) are as given below :

- (0, 0, 0)
- (-1, 0, 0)
- (1, 0, 0)
- (0, -1, 0)
- (0, 1, 0)
- (0, 0, -1)
- (0, 0, 1)

The points will appear on the Gmsh window as shown in the Fig 9.1 below.

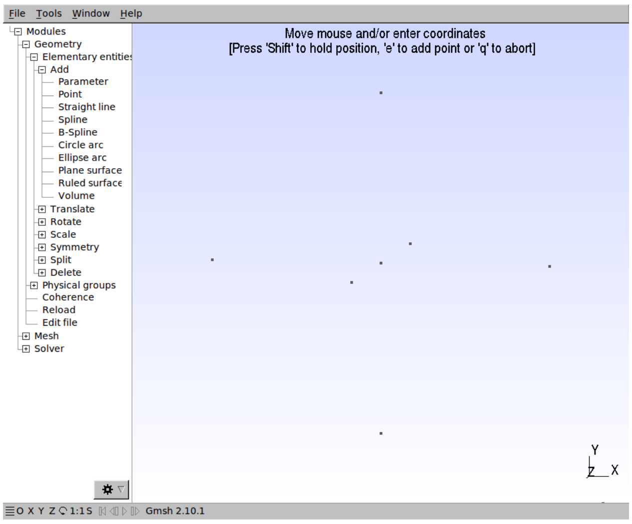


Figure 9.1: Sphere coordinates

9.2 Circular Arc

Creating an Arc is a three step process where we a start point, center and an end point . An important point to be noted here is that in Gmsh a Circular Arc is strictly created less than pi. Now to create an circular arc select Circle Arc option in the left hand side menu of Gmsh under Add. Click on the right most point in the Gmsh Window, it will turn red in colour, Fig 9.2.

Now click on the center of the Sphere as shown in Fig 9.3 and that too will turn red in colour.

For the end point click on the point above the center. As soon as we click on this point a arc is created as shown in the Fig 9.4.

Repeat this process for all the points to complete the sphere. Create the Arcs keeping the same center point. The completed geometry of the sphere is as shown , Fig **??**

9.3 Surface Creation

To stich the arcs together we now need to create surfaces. To do so, click on the Ruled Surface option under Add menu. Now select bounding edges for the surface as shown in Fig 9.6. After the selction they will turn red in colour. Press e on your keyboard to execute this selection. A crossed dotted line will be visible which shows that the surface has been created, Fig 9.7. Repeat the process to create all eight surfaces of the sphere, Fig 9.8.

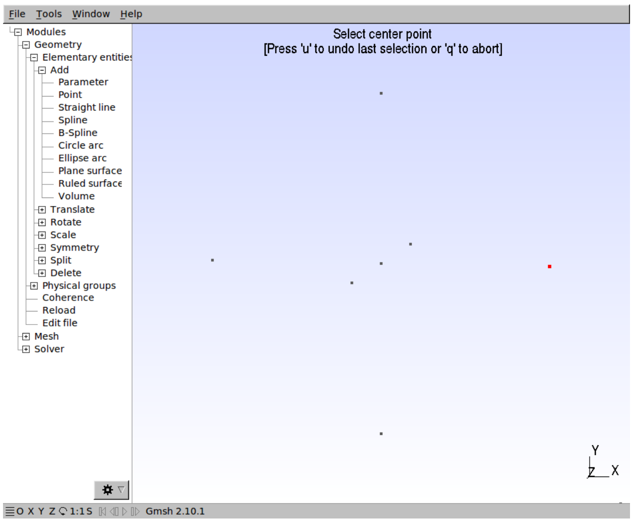


Figure 9.2: Rightmost point

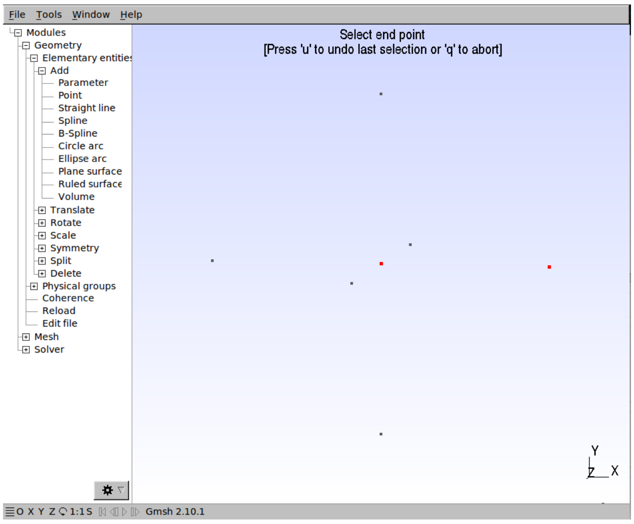


Figure 9.3: Center of the sphere

9.4 Editing .geo file

Gmsh provides us with an option of editing the saved file. Open sphere1.geo file in any editor of your choice. Information related to the geometric entities we create using Gmsh are stored here. General syntax under gmsh is as given below,

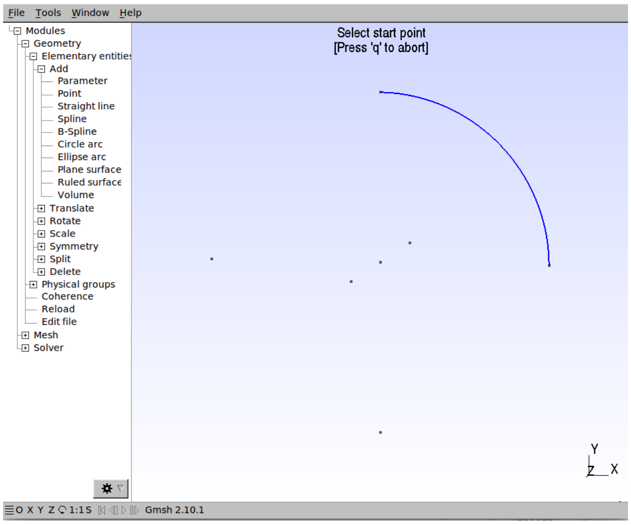


Figure 9.4: End point of the Arc

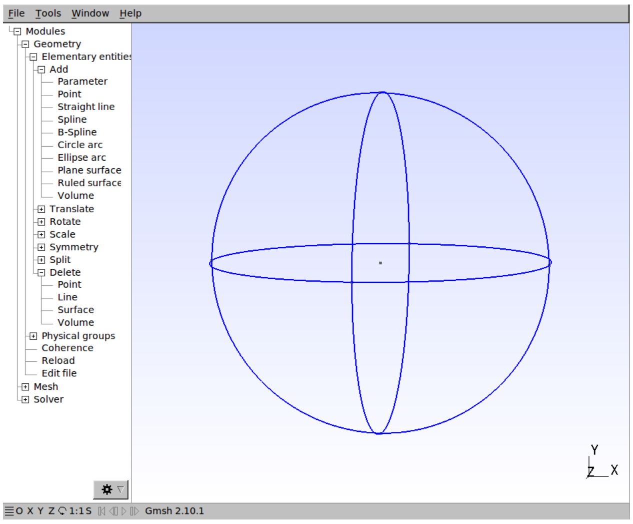


Figure 9.5: Sphere

Point (1) = 0,0,0,1;

here Point stand for the Geometrical Entity, (1) stands for Identification number inside the parenthesis next number starting from one which is equal to an expression. For points in expression we have the X, Y and Z co-ordinate followed by the value of desired mesh

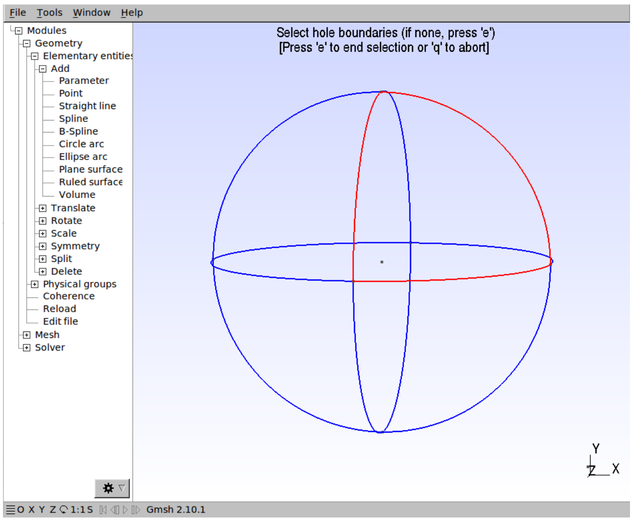


Figure 9.6: Surface Bounding edges

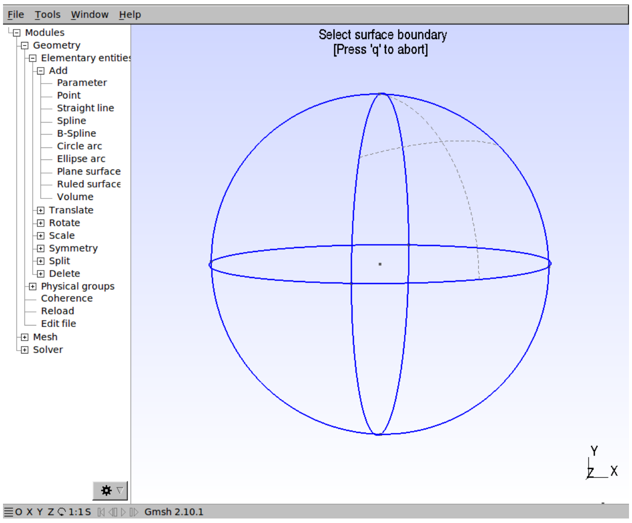


Figure 9.7: Surface Creation

element size. The size of the mesh element will then be computed by linearly interpolating these values on initial mesh. We can change the mesh element size here by a variable which can then take different value. Change the mesh element size from 1 to s and on top of the file type s = 0.1; as shown in the Fig 9.9

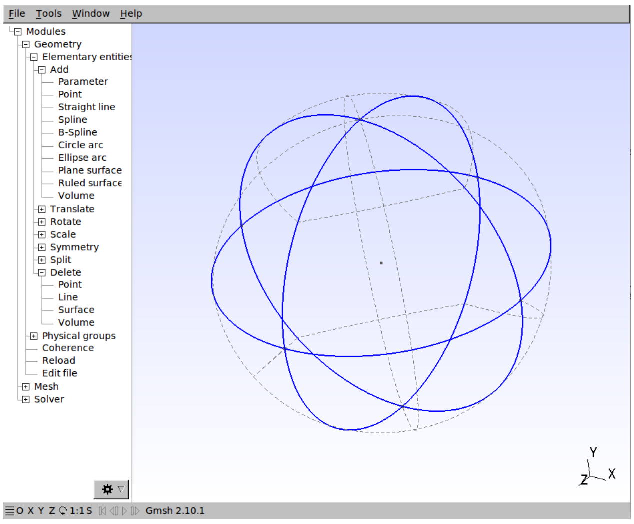


Figure 9.8: Complete Surface Creation

```
s= 0.1;

Point(1) = {0, 0, 0, s};
Point(2) = {-1, 0, 0, s};
Point(3) = {1, 0, 0, s};
Point(4) = {0, -1, 0, s};
Point(5) = {0, 1, 0, s};
Point(6) = {0, 0, -1, s};
Point(7) = {0, 0, 1, s};
```

Figure 9.9: Mesh Element size variable

9.5 Boundary Layer

Flow over any bluff body we are more interested in capturing the Boundary Layer. In this problem to capture the boundary layer in the geo file add a line after the mesh characteristic length varialble as

Mesh.CharacteristicLengthFromCurvature = 0.05;

This line above will adapt the mesh to the curvature w.r.t the geometrical entities.

9.6 Volume Creation

To create a volume we need all the bounding surfaces. This can also be done manually by typing at the end of the file

Surface Loop (identity) = identities of sphere surface within braces;

Now save this file and close it

9.7 Physical Groups

The geometry created now needs a physical meaning to it which helps us during simulation. To do this under Physical Groups go to Surface and select all the 8 surfaces of the sphere. It will turn red in colour, Fig 9.10. Press e to end selction and q to abort. Now again open the sphere1.geo file. It can be noticed that a new line has been added to the file which describes the physical surface. Here replace the Identification number by the name sphere within double quotes as this can be used as a boundary identification during simulation or postprocessing.

Physical Surface("Sphere") = 26, 27, 28, 30, 31, 32, 33, 34;

Save and close the file.

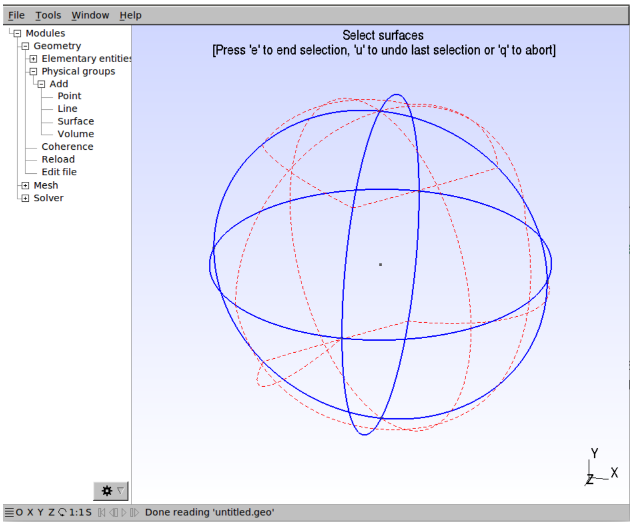


Figure 9.10: Physical Groups : Surface