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-2.3.0 and Paraview-4.1.0. 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 .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.

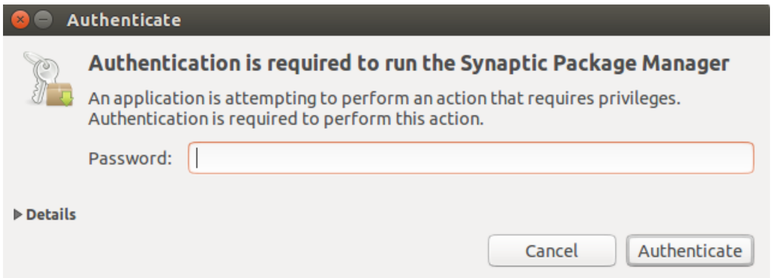


Figure 1.2: Enter system password to open Synaptic Package Manager

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

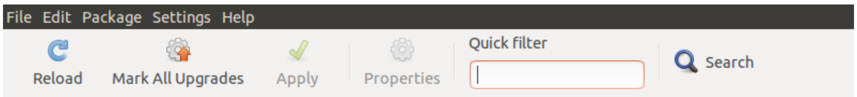
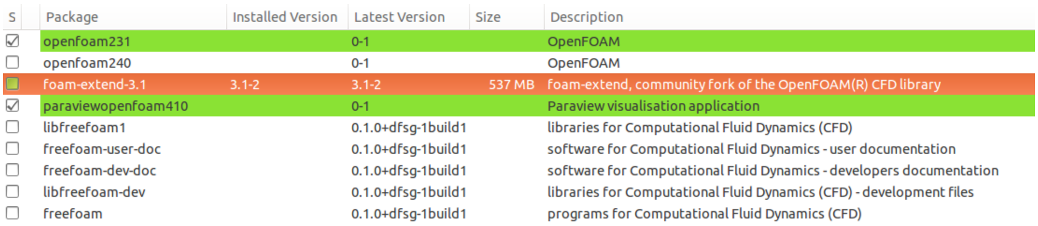


Figure 1.3: Search Box

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.



S	Package	Installed Version	Latest Version	Size	Description
☑	openfoam231		0-1		OpenFOAM
☐	openfoam240		0-1		OpenFOAM
	foam-extend-3.1	3.1-2	3.1-2	537 MB	foam-extend, community fork of the OpenFOAM(R) CFD library
☑	paraviewopenfoam410		0-1		Paraview visualisation application
☐	libfreefoam1		0.1.0+dfsg-1build1		libraries for Computational Fluid Dynamics (CFD)
☐	freefoam-user-doc		0.1.0+dfsg-1build1		software for Computational Fluid Dynamics - user documentation
☐	freefoam-dev-doc		0.1.0+dfsg-1build1		software for Computational Fluid Dynamics - developers documentation
☐	libfreefoam-dev		0.1.0+dfsg-1build1		libraries for Computational Fluid Dynamics (CFD) - development files
☐	freefoam		0.1.0+dfsg-1build1		programs for Computational Fluid Dynamics (CFD)

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

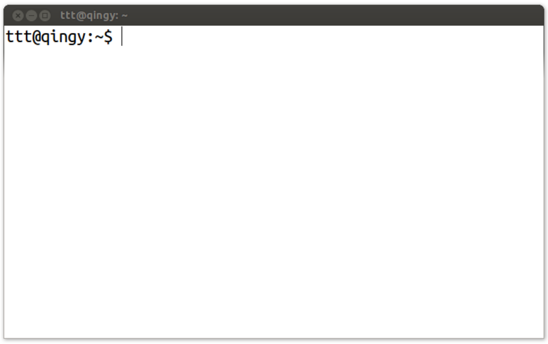


Figure 1.5: Terminal window

- 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.

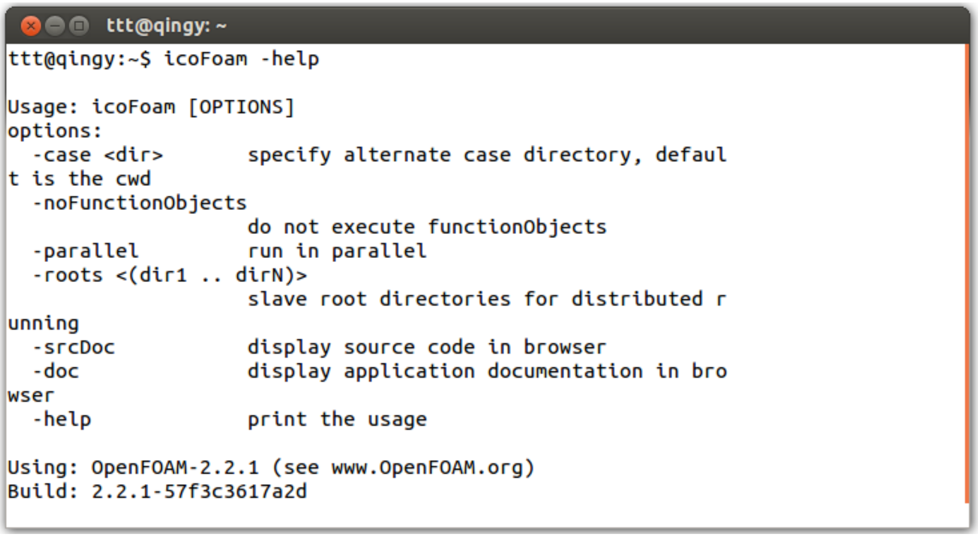


Figure 1.6: Usage Message

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.

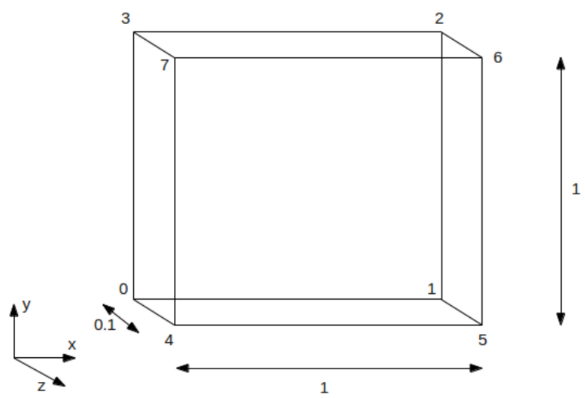


Figure 1.7: Lid Driven Cavity

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 3.3

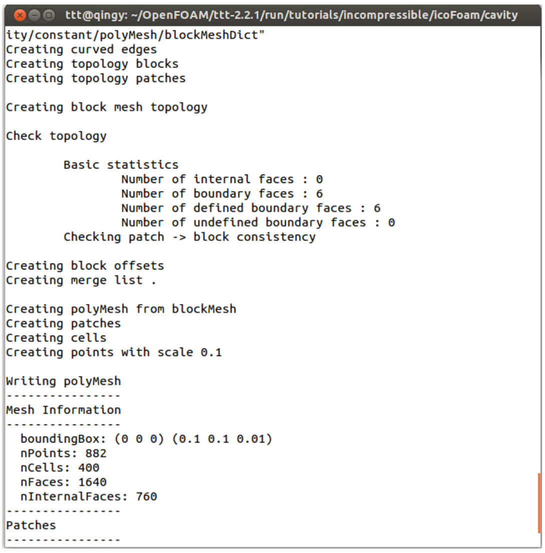


Figure 1.8: blockMesh for meshing

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

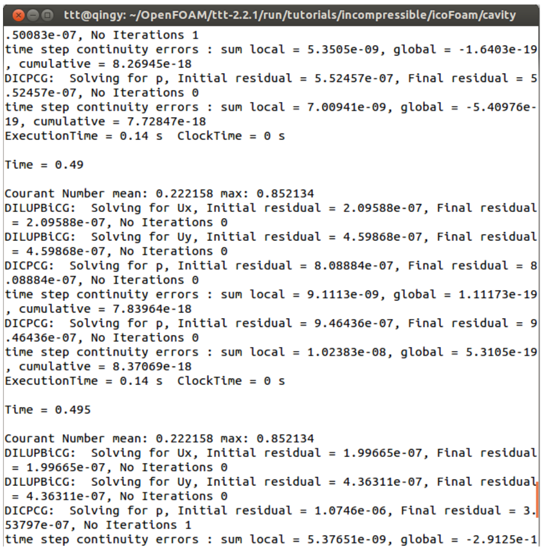
To Visualize the results we use Paraview. To open paraview in your terminal type

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, Fig3.6.

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.



```
ttt@qingy: ~/OpenFOAM/ttt-2.2.1/run/tutorials/incompressible/icoFoam/cavity
.50083e-07, No Iterations 1
time step continuity errors : sum local = 5.3505e-09, global = -1.6403e-19
, cumulative = 8.26945e-18
DICPCG:  Solving for p, Initial residual = 5.52457e-07, Final residual = 5
.52457e-07, No Iterations 0
time step continuity errors : sum local = 7.00941e-09, global = -5.40976e-
19, cumulative = 7.72847e-18
ExecutionTime = 0.14 s  ClockTime = 0 s

Time = 0.49

Courant Number mean: 0.222158 max: 0.852134
DILUPBiCG:  Solving for Ux, Initial residual = 2.09588e-07, Final residual
 = 2.09588e-07, No Iterations 0
DILUPBiCG:  Solving for Uy, Initial residual = 4.59868e-07, Final residual
 = 4.59868e-07, No Iterations 0
DICPCG:  Solving for p, Initial residual = 8.08884e-07, Final residual = 8
.08884e-07, No Iterations 0
time step continuity errors : sum local = 9.1113e-09, global = 1.11173e-19
, cumulative = 7.83964e-18
DICPCG:  Solving for p, Initial residual = 9.46436e-07, Final residual = 9
.46436e-07, No Iterations 0
time step continuity errors : sum local = 1.02383e-08, global = 5.3105e-19
, cumulative = 8.37069e-18
ExecutionTime = 0.14 s  ClockTime = 0 s

Time = 0.495

Courant Number mean: 0.222158 max: 0.852134
DILUPBiCG:  Solving for Ux, Initial residual = 1.99665e-07, Final residual
 = 1.99665e-07, No Iterations 0
DILUPBiCG:  Solving for Uy, Initial residual = 4.36311e-07, Final residual
 = 4.36311e-07, No Iterations 0
DICPCG:  Solving for p, Initial residual = 1.0746e-06, Final residual = 3.
53797e-07, No Iterations 1
time step continuity errors : sum local = 5.37651e-09, global = -2.9125e-1
```

Figure 1.9: Iteration on Terminal Window

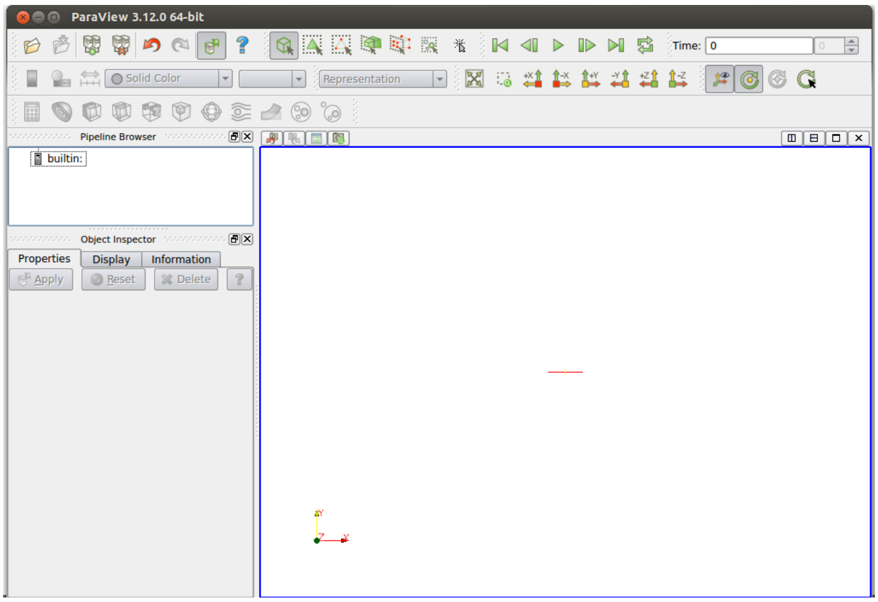


Figure 1.10: Paraview window

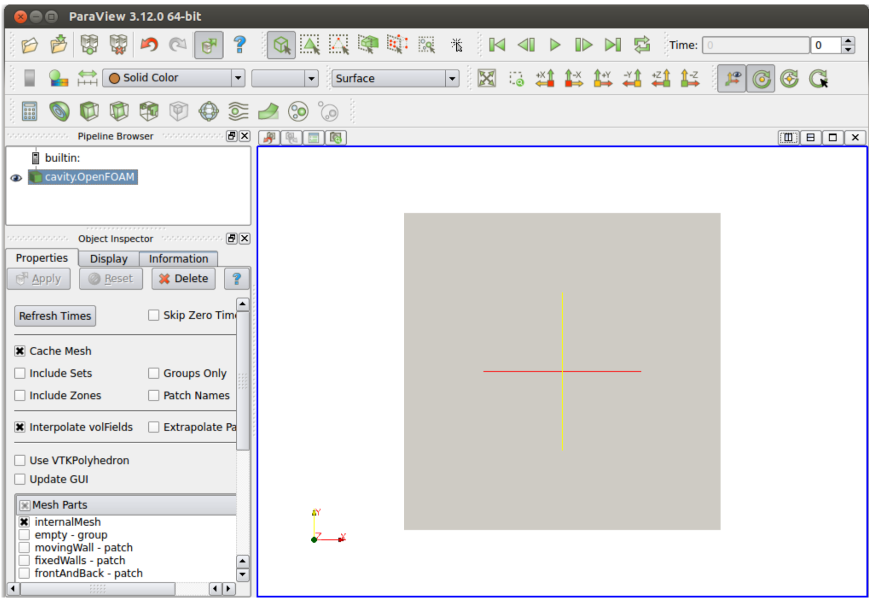


Figure 1.11: Geometry

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 2.1

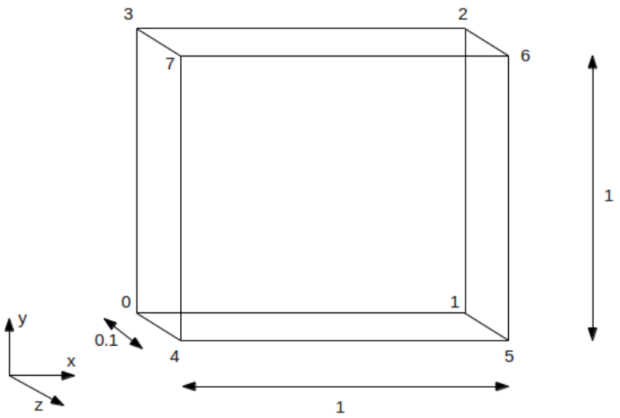


Figure 2.1: geomtery points of the lid driven cavity

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 2.3:

```
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 3.4:

- moving wall
- fixed wall
- front and back

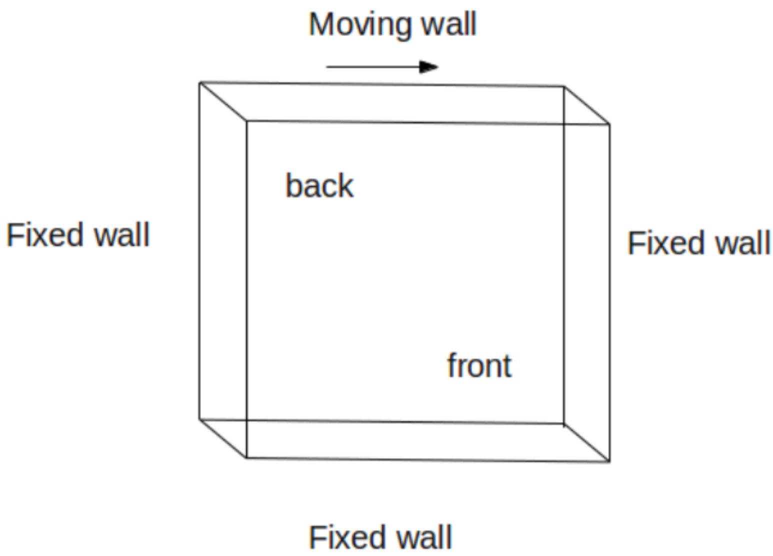


Figure 2.5: boundary names of the geomtery

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 2.8:

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

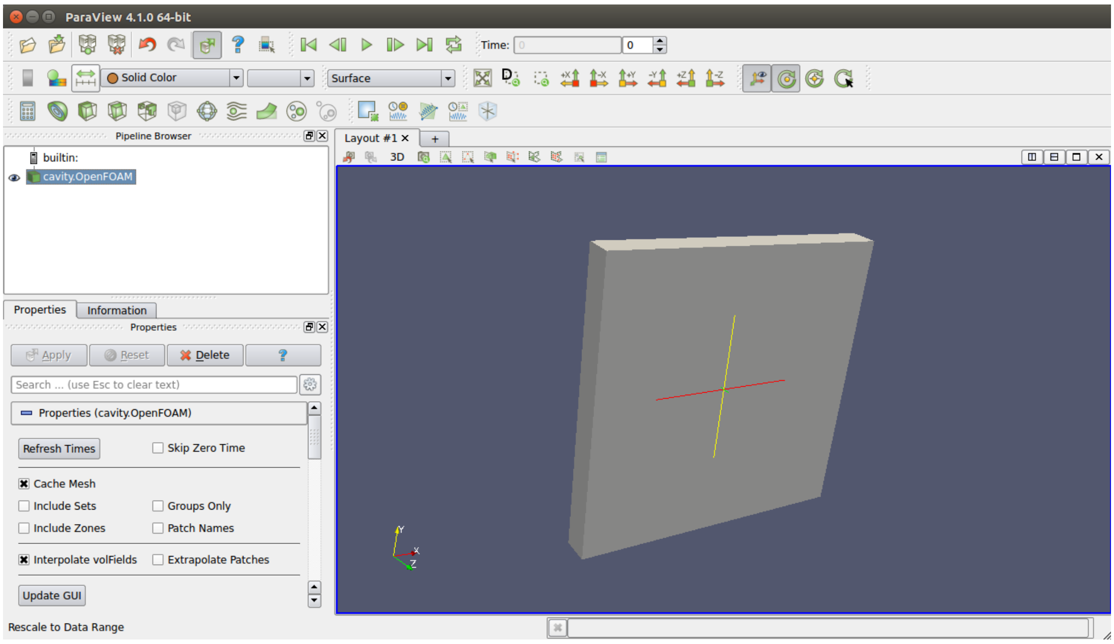


Figure 2.8: Paraview window showing the 2-D geometry

Chapter 3

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.

3.1 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 3.1 below.

3.2 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 3.3

3.3 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

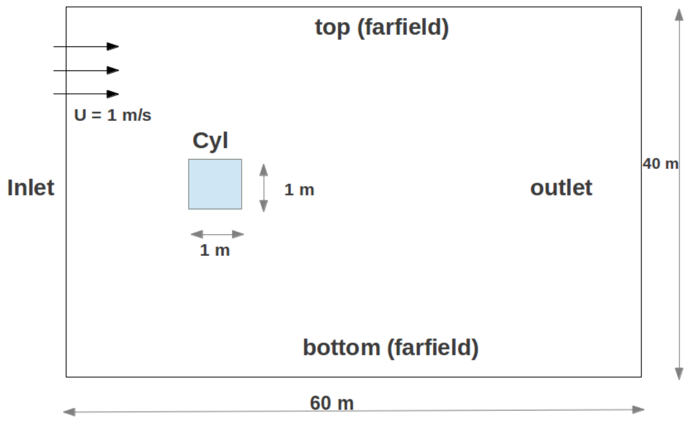


Figure 3.1: Flow over square Cylinder

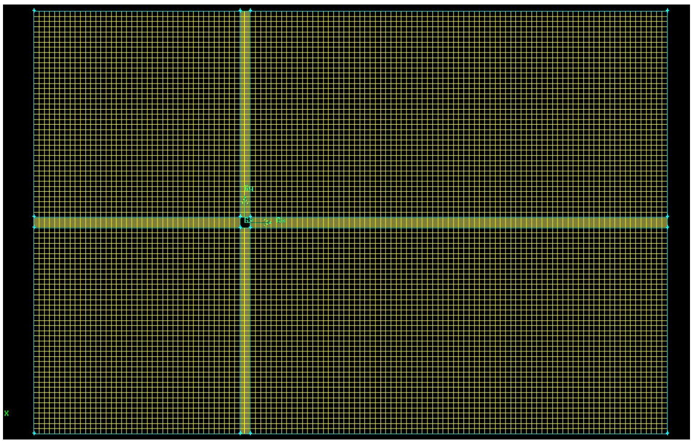


Figure 3.2: Mesh

- 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.

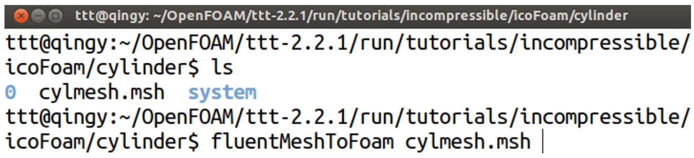


Figure 3.3: convert

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

3.4 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 3.4.

The boundary names will be as shown in the domain shown above, Fig 3.1. 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 3.4: Boundary file

3.5 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 3.5 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 3.5: 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.

3.6 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 3.6. In the active varialble control menu change from Solid Color to Velocity (U). You can now see the initial conditions for velocity, Fig 3.7. 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 3.8.

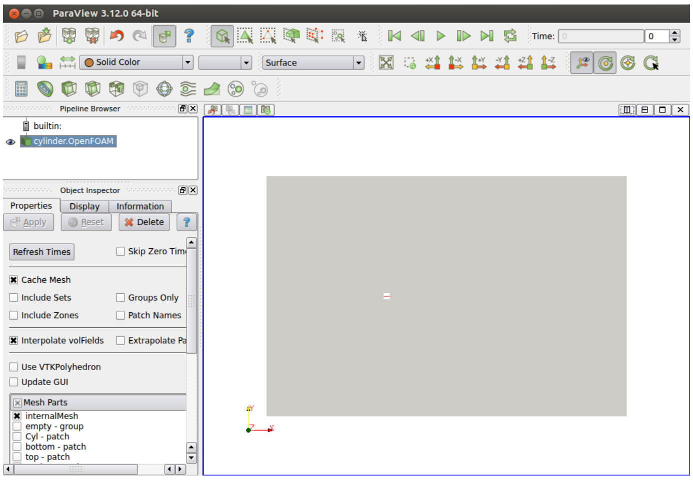


Figure 3.6: Geometry in Paraview

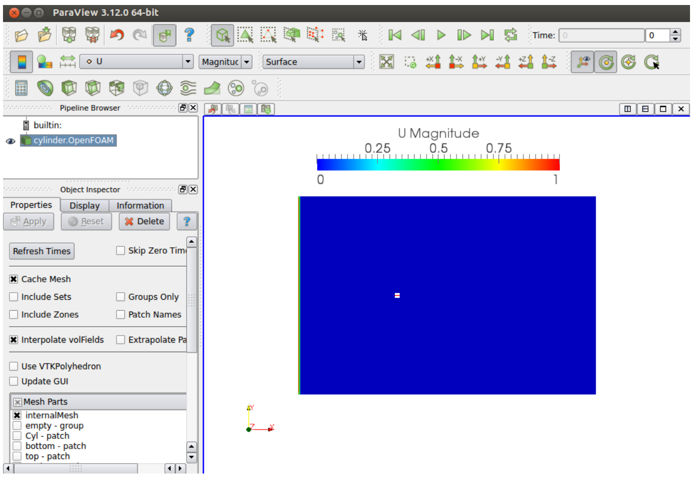


Figure 3.7: Initial velocity condition

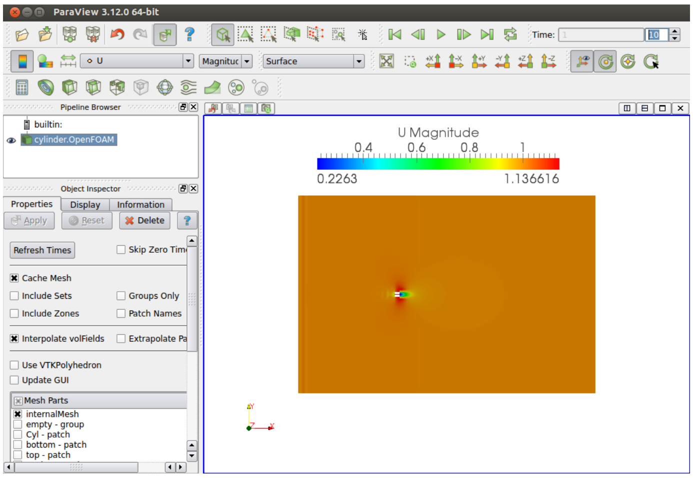


Figure 3.8: Velocity at 1 sec

3.7 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 4

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.

4.1 Installing Gmsh

Gmsh can be installed using Synaptic Package Manager. Open Gmsh in your system by typing your system passowrd. In the search box type Gmsh and install it, Fig 4.1. This might take some time depending on your internet speed.

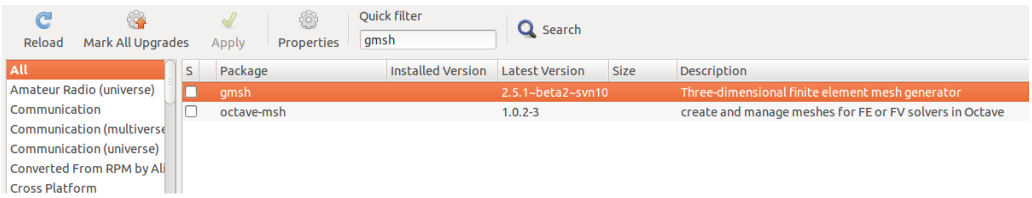


Figure 4.1: Install Gmsh

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 4.2 according to your Operating System (OS).

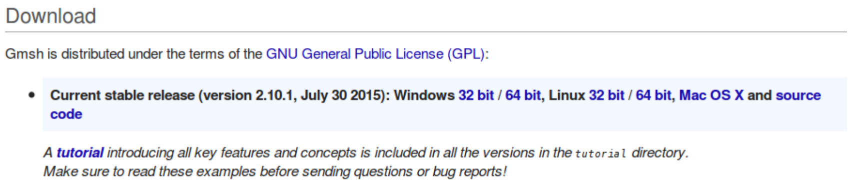


Figure 4.2: Download stable release

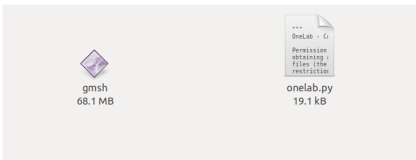


Figure 4.3: gmsh-icon

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 4.3. Double click on it to launch the Gmsh Start screen, Fig 4.4

As a pracice to learn Gmsh we will create a cube of sides 1 unit as seen in the Fig, 4.5. 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 4.6. Now enter points for all the remaining 7 vertices to complete the cube, Fig 4.5. 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 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, 4.7 below. As you can see on the Gmsh screen you can press e to end selection and q to abort.

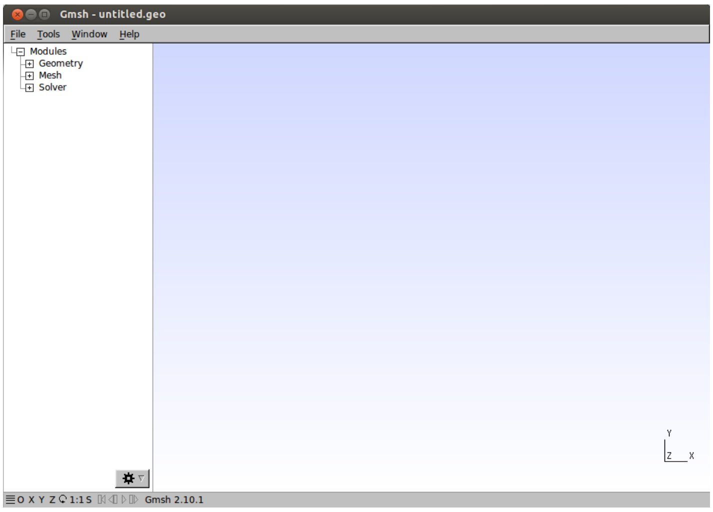


Figure 4.4: gmsh-icon

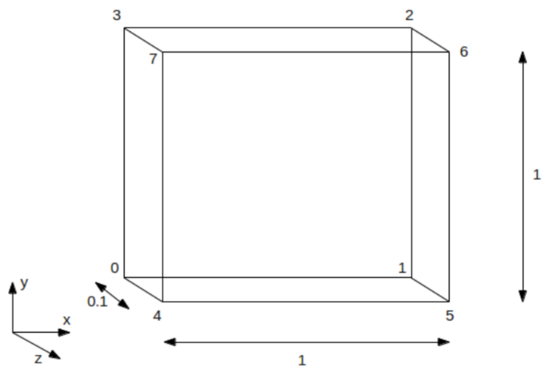


Figure 4.5: Cube of unit dimension

4.1.1 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 first.Once you select the edges they will turn red in color, Fig 4.8. 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 4.9. Repeat this procedure for remaining faces, Fig 4.10 and finally press q to abort.

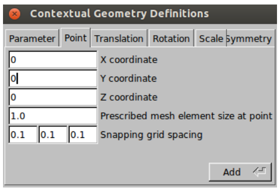


Figure 4.6: Points window

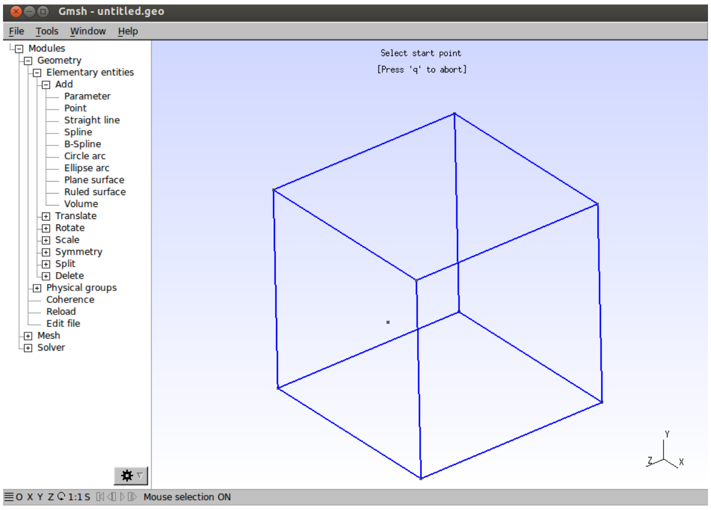


Figure 4.7: Join the points using line

4.1.2 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.

4.1.3 Physical Groups

In Gmsh we need to 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 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 faces and press q to abort. Also we need to select the Physical Volume. Click on Volume under Physical Groups and select the yellow dot at

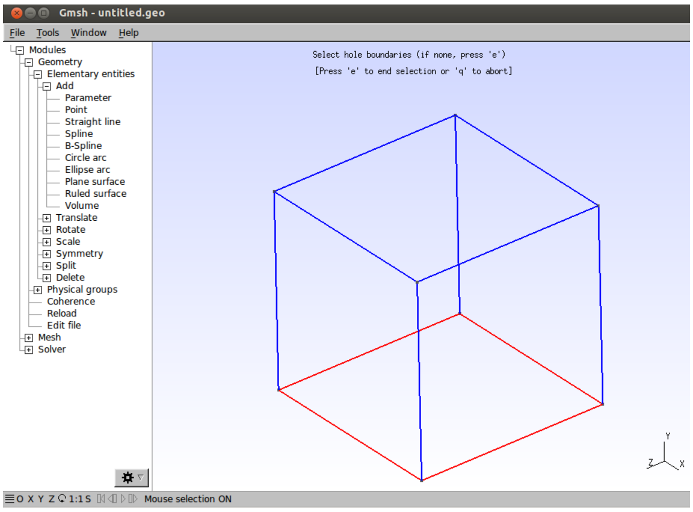


Figure 4.8: Selct edges

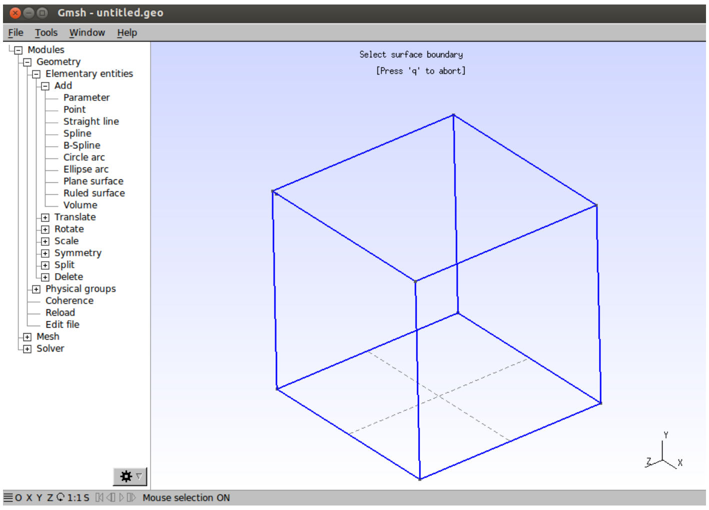


Figure 4.9: Bottom Face

the center of the cube. The yellow dot will turn red in colora dn 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.

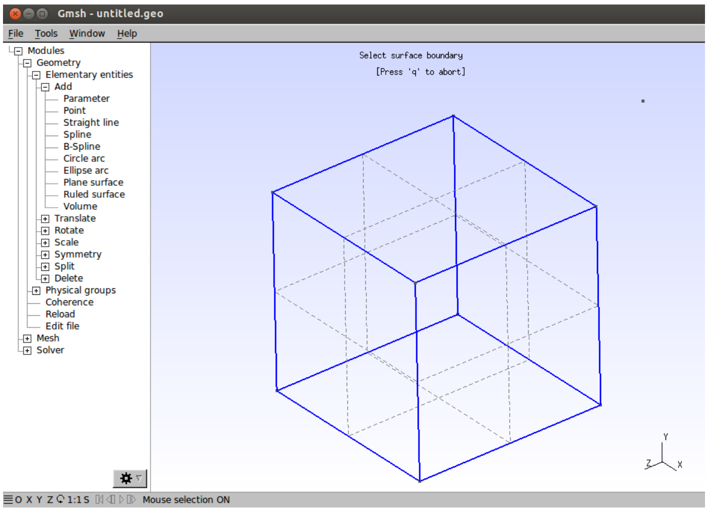


Figure 4.10: Create faces for all surfaces

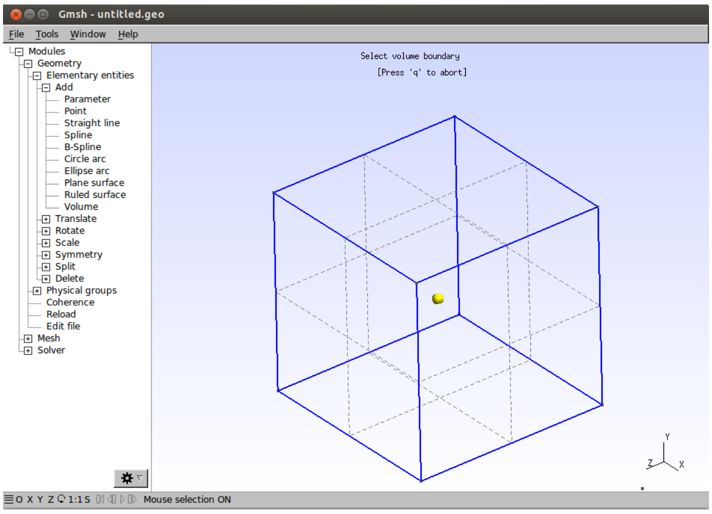


Figure 4.11: Volume

Chapter 5

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.

5.1 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 5.1, 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 5.2. 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, 5.3. 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 5.4. 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 or Ubuntu 14.04 64-bits binary, Fig 5.5. Click on it and Save the file. Since the file size is big it will take some time to download. 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.

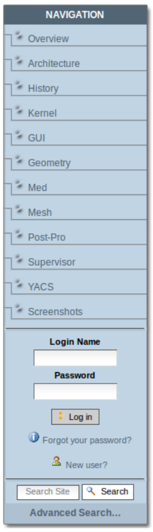


Figure 5.1: Navigation Bar

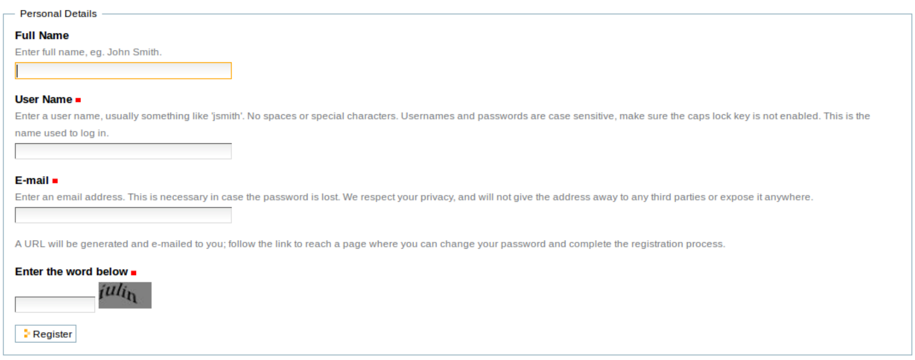
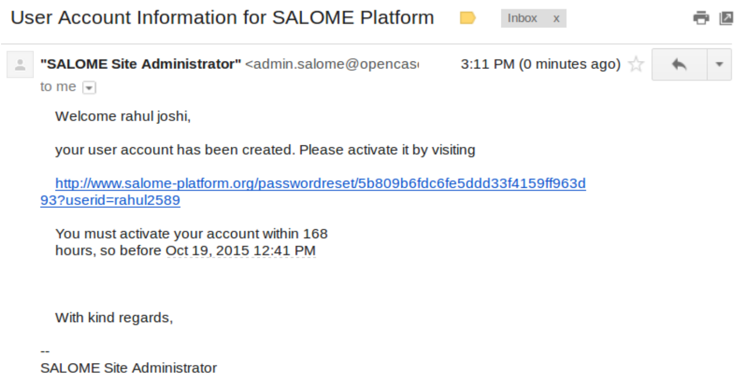


Figure 5.2: User Details



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

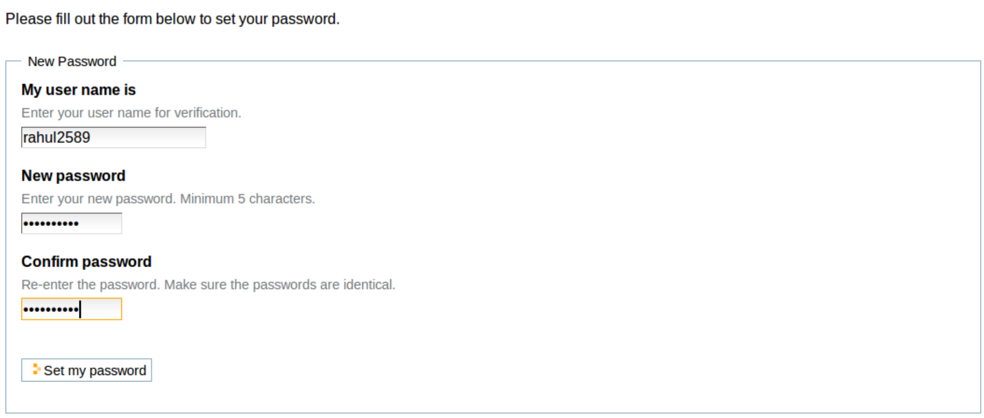
http://www.salome-platform.org/passwordreset/5b809b6fdc6fe5ddd33f4159ff963d93?userid=rahul2589

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

With kind regards,

--
SALOME Site Administrator

Figure 5.3: 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.

••••••••••

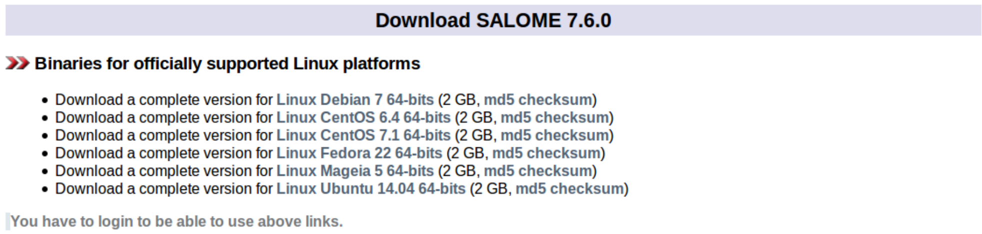
Confirm password

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

••••••••••

Set my password

Figure 5.4: Enter Password



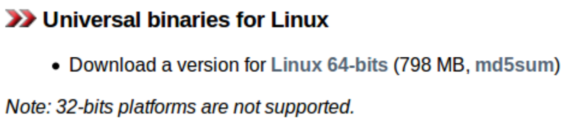
Download SALOME 7.6.0

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 5.5: Salome Linux Debain 7 64 bit binary



Universal binaries for Linux

- Download a version for **Linux 64-bits** (798 MB, **md5sum**)

Note: 32-bits platforms are not supported.

Figure 5.6: Universal Binaries

Chapter 6