Department of Chemical Engineering, IIT Bombay
Instruction Manual for solving Turbulent Flow in a Horizontal Pipe problem using OpenFOAM
CFD Lab, Autumn Semester, Year 2019-20
Turbulent Flow in a Horizontal Pipe

Objective

To carry out CFD simulation of Turbulent flow in a Horizontal Pipe using OpenFOAM.

To obtain various contours such as Velocity, Pressure, Turbulent Kinetic Energy and Turbulent Dissipation Rate.

Geometry and Conditions

Straight horizontal pipe.

Radius of the pipe (r) = **2.5 cm**

Length of the pipe (L) = **100 cm**

Orientation of the pipe is such that the axis is in textbfz direction.

Fluid : **Liquid Water**.

Mass flow rate = **(0.2 + 0.03*n)** kg/sec, where **n is the group number for batch 1 and (group number<minus>30) for batch 2**.

Note:

1. **Please ignore the angular brackets in the whole manual.**

2. **Make sure of providing spaces, capital and small letters when typing command in the terminal window.**

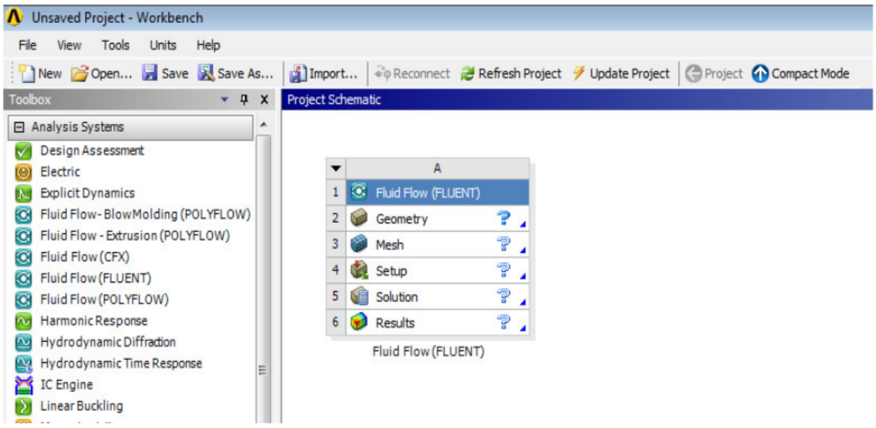
3. **Read the manual carefully before asking for help to the TAs.**

Procedure for Mesh Generation using FLUENT

1. Click on **OVD-EDC** and **log in** using your **LDAP ID** and **Password**.

2. Scroll down on the available software and click on **Workbench Workbench** will be opened.

3. Drag **Fluid Flow(FLUENT** from the **Toolbox** available in the right side of the Workbench window.

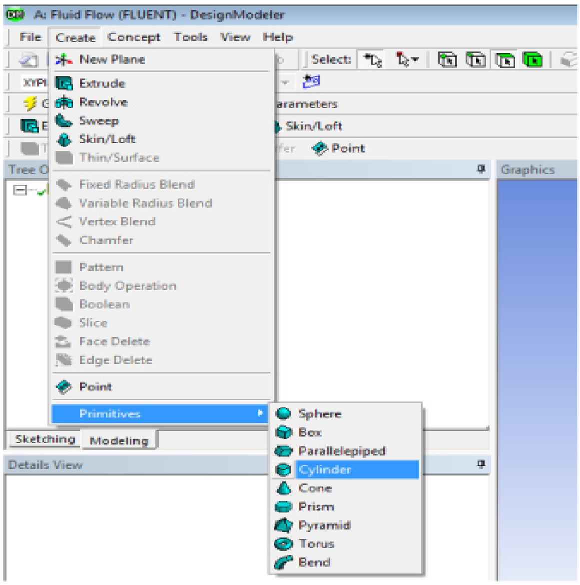


4. Double click on **Geometry**.

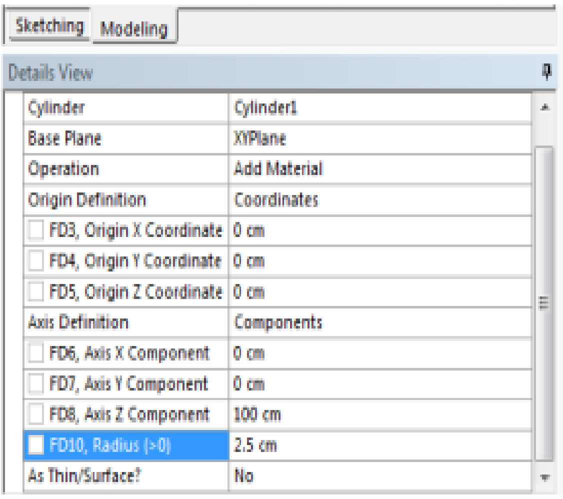
5. Click **Units** in the **Menu bar** and select **Meter** (Which will be default. Just ensure it is selected).

6. Click on **Create** in the **Menu bar**.

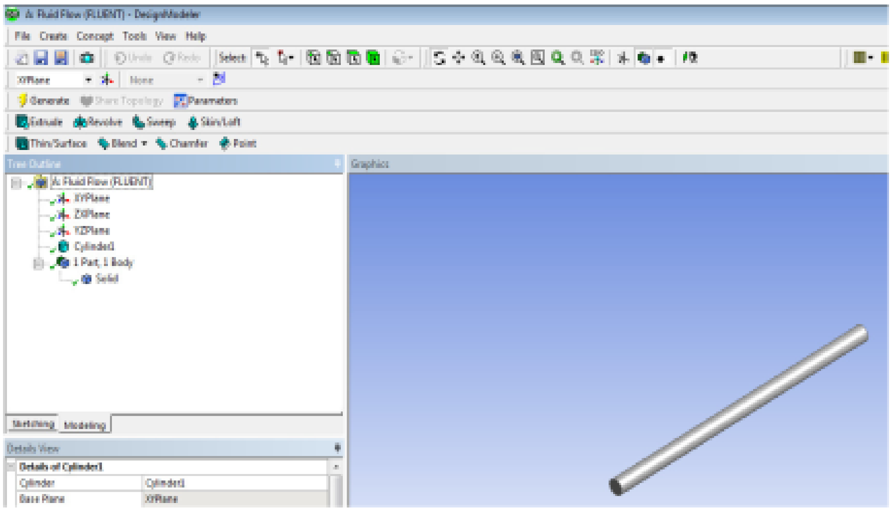
7. Select **Primitives** and click **Cylinder**.



8. Specify **Radius** as **0.025 m** and **Axis Z component** (length) as **1 m** in the **Sketching** Toolbox.



9. Right Click and click **Generate**.
10. Scroll the mouse to make the geometry visible.
11. In the **Tree Outline** window, **1 Part, 1 Body** should appear.

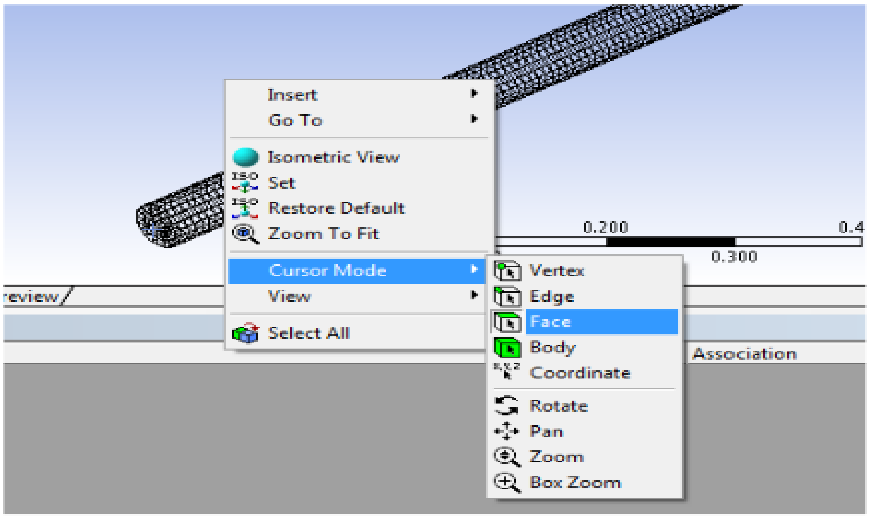


12. Close the **DesignModeler**.
13. Double click the **Mesh** in the **Project Schematic** box available in the **Work-bench** window.
14. Meshing window will open.
15. Double click the **Mesh** option available in the **Project** Outline box.
16. Expand **Sizing** option available in the **Details of "Mesh"** Toolbox.
17. In **Max Face Size** specify **5e-3**.
18. In **Max Size** specify **5e-3**.

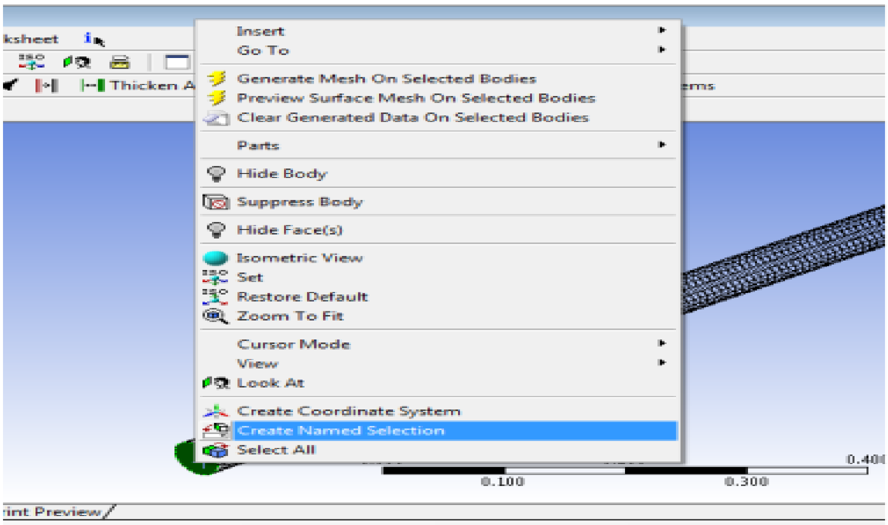
19. Right click on **Mesh** option in **Project** Outline box and click Generate Mesh.

20. Right click on **Mesh** option in **Project** Outline box and click **Update**.

21. Right click in the graphic window. Select **Cursor Mode** and then select **Face**.



22. Select one of the two end faces. Right click and select **Create Named Selection**.



23. In **Selection name** specify the name as **inlet**.

24. **Similarly select the "other end face"**. Right click and select **Create Named Selection**. In **Selection name** specify the name as **outlet**.

25. Select the periphery of the cylinder. Right click and select **Create Named Selection**. In **Selection name** specify the name as **walls**.

26. In Fluent Meshing module click

    Tools –> Options –> Click Expand Meshing –> Export

27. In the **Format of Input File (*.msh)** change to **ASCII** format.

28. Click **OK**.

29. Click

    ```
    File -> Export
    ```

30. Change the **Save as type** as **FLUENT Input Files (*.msh)**

31. Mention the file name as

    ```
    turbpipe_group<Number>
    Please ignore the angular brackets
    ```

32. Select the location as

    ```
    This PC -> Desktop on guest-desktop
    ```

33. Click **Save** button.

34. Close the Meshing window.

35. Go back to Workbench window.

36. Save the project using appropriate Group name for future use.

Step-1

Connecting to the Server

1. Division 1 is Group 01 to Group 30.

2. Division 2 is Group 31 to Group 60

3. Press **ctrl**+ **Alt**+**T** to open the terminal

4. Type the following command to connect to the server.

   For Division-1
   (For groups 01-15)

   ```
   ssh test<GroupNumber>@10.102.1.121 -X
   Password:test<GroupNumber>
   ```

   (For groups 16-30)

   ```
   ssh test<GroupNumber>@10.102.1.122 -X
   Password:test<GroupNumber>
   ```

   For example
   Group-06

   ```
   ssh test06@10.102.1.121 -X
   Password:test06
   ```

Group-27

```
ssh test27@10.102.1.122 -X
Password:test27
```

For Division-2
(For groups 31-45)

```
ssh test<GroupNumber-30>@10.102.1.121 -X
Password:test<GroupNumber-30>
```

(For groups 46-60)

```
ssh test<GroupNumber-30>@10.102.1.122 -X
Password:test<GroupNumber-30>
```

For example
Group-39

```
ssh test09@10.102.1.121 -X
Password:test09
```

Group-58

```
ssh test28@10.102.1.122 -X
Password:test28
```

Step-2

1. Now you are in the home directory of your user.
2. To create the our case folder type the following

```
For Groups 1-30
mkdir turbpipe<GN>_div1

For Groups 31-60
mkdir turbpipe<GN>_div2
```

For example

```
mkdir turbpipe03_div1    (For Group-3)
mkdir turbpipe38_div2    (For Group-38)
```

Step-3

Choosing the Solver

1. Before starting to problem we need to select our solver according to the flow conditions.

2. For this case since we are dealing with steady state, incompressible, turbulent flow we can choose **simpleFoam** solve in the OpenFOAM inside the incompressible solver category.

3. To create the case directory of our problem, we make use of the tutorial case for **simpleFoam** solver, which is opened by typing the following command in the terminal

```
cd $FOAM_TUTORIALS
cd incompressible/simpleFoam/pitzDaily
```

4. Now type **ls** command in the terminal to display the contents inside the folder.

5. To copy the files **0, constant and system** folders to our case folder **turbpipe** type the following command.

```
For Group Number(GN) 01-30
cp -r 0 constant system /home/test<GN>/turbpipe<GN>_div1

For Group Number(GN) 31-60
cp -r 0 constant system /home/test<GN-30>/turbpipe<GN>_div2
```

For example,

```
Group03
cp -r 0 constant system /home/test03/turbpipe03_div1
Group39
cp -r 0 constant system /home/test09/turbpipe39_div2
```

Step-4

Choosing the Turbulence Model and Calculating the Parameters

1. We use **k-**ϵ turbulence model for this case. Hence we need to calculate the k and ϵ values corresponding to the specifications.

2. Now calculate the inlet velocity from the mass flow rate

$$\dot{m} = \rho A V \tag{1}$$

where Inlet Diameter is 5 cm

3. Calculate Reynolds number based on the diameter of the pipe.

$$Re = \frac{V * D}{\nu} \tag{2}$$

where ν is the Dynamic Viscosity of water. In this case it is $10^{-6} m^2/s$

4. Calculate Turbulent Intensity (I) using the expression given below

$$I = 0.16 * (Re)^{(-1/8)} \tag{3}$$

5. Now calculate the Turbulent Kinetic Energy (k) based on the inlet velocity as given below

$$k = \frac{3}{2}(V * I)^2 \tag{4}$$

where I is the turbulence intensity

6. Now calculate the Turbulence Dissipation rate (ϵ) based on the **k** as given below

$$\epsilon = (C_\mu)^{\frac{3}{4}} * \frac{k^{\frac{3}{2}}}{l} \tag{5}$$

where,
l is $0.07d_h$ is the turbulent length scale, where d_h is the hydraulic diameter, which for circular pipes is equal to the diameter d.
$C_\mu = 0.09$

Step-5

Setting up the Case Directory

1. Now we need to make changes in the case folder.

2. It consists of three files, which are **0**,**constant** and **system**.

3. We need to make changes in files of the **0** folder to setup initial boundary conditions for our case.

4. Open the case folder and view the files available. To do so in terminal type

```
cd
cd turbpipe<GN>_div<No>/0
ls
```

5. There will be several files such as **epsilon**, **k**, **nut**,**p**,**U**,**f**,**omega** and **v2**.

6. Please delete the files **f**, **omega** and **v2**, which doesn't required for the turbulence model we have chosen.

7. To do so, in the terminal type

```
rm f omega v2
```

8. Now type **ls** command in terminal. There will be **epsilon**, **k**, **nut**, **p** and **U** files only.

Setting up of k (Turbulent Kinetic Energy) file

1. Since we have three boundary patches in our mesh :- inlet, outlet and walls, we need to enter these three face types in these as well.

2. Open the **k** file using editor of your choice. In this case we use Gedit.

```
cd
cd turbpipe<GN>_div<No>/0
gedit k
```

3. The first line shows dimensions. These dimensions set consists of 7 basic units such as [Mass Length Time Temperature Quantity Current Luminosity].

4. Open the k file.

```
gedit k
```

5. keep the dimensions for the file as default. In the **k** (m^2/s^2) file it will be

```
dimensions          [0 2 -2 0 0 0 0];
```

6. The initial data for the overall domain can be initialized by the following line defined after dimensions

```
internalField    uniform <calculated value>;
```

7. The boundary patches are edited as shown below.

```
inlet
    {
        type                fixedValue;
        value               uniform <calculated value>;
        //Omit the angular brackets
        //There should be a space between uniform and the number
        //to be entered.
    }
    outlet
    {
        type                zeroGradient;
    }
    walls
    {
        type                kqRWallFunction;
        value               uniform <calculated value>;
        //Omit the angular brackets
        //There should be a space between uniform and the number
        //to be entered.
    }
```

8. Save it and close the file.

Setting up of Epsilon (Turbulent Dissipation rate) file

1. Open the epsilon file.

```
gedit epsilon
```

2. keep the dimensions for the file as default. In the $\epsilon(m^2/s^3)$ file it will be

```
dimensions          [0 2 -3 0 0 0 0];
```

3. The initial data for the overall domain can be initialized by the following line defined after dimensions

```
internalField    uniform <calculated value>;
```

4. The boundary patches are edited as shown below.

```
inlet
    {
        type                fixedValue;
        value               uniform <calculated value>;
        //Omit the angular brackets
        //There should be a space between uniform and the number
        //to be entered.
    }
    outlet
    {
        type                zeroGradient;
    }


walls
    {
        type                epsilonWallFunction;
        value               uniform <calculated value>;
         //Omit the angular brackets
         //There should be a space between uniform and the number
        //to be entered.
    }
```

5. Save it and close the file

Setting up of nuTilda file

1. Open the nuTilda file.

```
gedit nuTilda
```

2. keep the dimensions for the file as default.

```
dimensions          [0 2 -1 0 0 0 0];
```

3. We keep the same conditions for this file. Just change the patch name
4. The initial data for the overall domain can be initialized by the following line defined after dimensions

```
internalField    uniform 0;
```

5. The boundary patches are edited as shown below.

```
    inlet
    {
        type            calculated;
        value           uniform 0;
    }
    outlet
    {
        type            zeroGradient;
    }


    walls
    {
        type            zeroGradient;
    }
```

6. Save it and close the file

Setting up of nut file

1. Open the nut file.

```
gedit nut
```

2. keep the dimensions for the file as default.

```
dimensions          [0 2 -1 0 0 0 0];
```

3. We keep the same conditions for this file. Just change the patch name
4. The initial data for the overall domain can be initialized by the following line defined after dimensions

```
internalField    uniform 0;
```

5. The boundary patches are edited as shown below.

```
    inlet
    {
        type            calculated;
        value           uniform 0;
    }
    outlet
```

```
        {
            type                calculated;
            value               uniform 0;
        }

        walls
        {
            type                nutkWallFunction;
            value               uniform 0;
        }
```

6. Save it and close the file

Setting up of Velocity (U) file

1. Open the U file.

   ```
   gedit U
   ```

2. The first line shows dimensions. These dimensions set consists of 7 basic units such as [Mass Length Time Temperature Quantity Current Luminosity].

3. keep the dimensions for the file as default. In the **U (m/s)** file it will be

   ```
   dimensions          [0 1 -1 0 0 0 0];
   ```

4. Since velocity is vector we need to define three components for it.

5. The initial data for the overall domain can be initialized as 0 by the following line defined after dimensions

   ```
   internalField     uniform (0 0 0);
   ```

6. The boundary patches are edited as shown below.

   ```
   inlet
       {
           type                    fixedValue;
           value                   uniform (0 0 <-Calculated Velocity>);
           //since flow direction is in negative Z axis.
           //Omit the angular brackets
       }

   outlet
       {
           type                zeroGradient;
       }
   walls
       {
           type                fixedValue;
           value               uniform (0 0 0);
       }
   ```

7. Save it and close the file.

Setting up of Pressure (p) file

1. Open the p file.

```
gedit p
```

2. keep the dimensions for the file as default.

```
dimensions          [0 2 -2 0 0 0 0];
```

3. The initial data for the overall domain can be initialized as 0 by the following line defined after dimensions

```
internalField     uniform 0;
```

4. The boundary patches are edited as shown below.

```
inlet
    {
        type                zeroGradient;
    }

    outlet
    {
         type                fixedValue;
         value               uniform 0;
    }
    walls
    {
        type                      zeroGradient;
    }
```

5. Save it and close the file.

Step-6

1. The kinematic viscosity of water in this case we consider as $1 * 106{-}06$

2. Open the constant folder to edit the kinematic viscosity value.

```
cd
cd turbpipe<GN>_div<No>/constant
gedit transportProperties
```

3. Change the value as 1e-06.

```
nu                  [0 2 -1 0 0 0 0] 1e-06;
```

4. Save the file and close it.

5. This completes the setup of our case. Let us begin to solve it.

Step-7

1. We will make changes to **controlDict** file of the **system** folder.
2. Go inside the system folder and open the controlDict file.

   ```
   cd
   cd turbpipe<GN>_div<No>/system
   gedit controlDict
   ```

3. Change the **endTime** as **1000**.
4. Save this and close the file.

Step-8

1. We will make changes to **fvSolution** file of the **system** folder.
2. Go inside the system folder and open the fvSolution file.

   ```
   cd
   cd turbpipe<GN>_div<No>/system
   gedit fvSolution
   ```

3. Change the residualControl as shown below

   ```
   SIMPLE
   {
       nNonOrthogonalCorrectors 0;
       consistent         yes;

       residualControl
       {
           p                      1e-5;
           U                      1e-5;
           "(k|epsilon|omega|f|v2)" 1e-3;
       }
   }
   ```

4. Save this and close the file.
5. **Don't close the current terminal**.

Step-9

Copying the Mesh file to Case Directory

1. Open a new terminal by typing **Ctrl+Alt+t** keys together.
2. Now type

```
cd Desktop
```

3. Type **ls** command to view the contents. You can see the mesh file

   ```
   turbpipe_group<Number>.msh
   ```

4. To copy the mesh file to our case directory,

   For Division-1

   (For groups 01-15)

   ```
   scp turbpipe<G>.msh test<n>@10.102.1.121:/home/test<n>/turbpipe<G>_div<N>
   ```

   (For groups 16-30)

   ```
   scp turbpipe<G>.msh test<n>@10.102.1.122:/home/test<n>/turbpipe<G>_div<N>
   ```

   n is your group number

   G is your group number

   N is your division number

   In the above command ignore angular brackets

   For Division-2

   (For groups 31-45)

   ```
   scp turbpipe<G>.msh test<n>@10.102.1.121:/home/test<n>/turbpipe<G>_div<N>
   ```

   (For groups 46-60)

   ```
   scp turbpipe<G>.msh test<n>@10.102.1.122:/home/test<n>/turbpipe<G>_div<N>
   ```

   n is your (group number<minus>30)

   G is your group number

   N is your division number

   In the above command ignore angular brackets

Meshing the Geometry

Converting fluent Mesh to OpenFOAM

1. Now we create the mesh file for our case using the **turbpipe.msh** file

2. Type the following command to create the mesh.

   ```
   cd
   cd turbpipe<GN>_div<1 or 2>
   fluent3DMeshToFoam turbpipe.msh
   ```

3. Mesh will be created.

4. Type the following command to check the mesh

```
checkMesh
```

5. It will display whether the **mesh is OK** of it has any errors. In case of any error have a better look at the error in the terminal and make changes accordingly.

   You have now finished creating the geometry, meshing it and setting up the boundary faces.

Step-10

1. Now setting up the case directory for the problem has been completed.

2. Now we need to run the case.

3. Type the following

```
cd
cd turbpipe<GN>_div<1 or 2>
simpleFoam
```

4. The iterations will be running. Wait till the specified **endTime** is reached.

Step-11

1. Open the case folder.

```
cd
cd turbpipe<GN>_div<1 or 2>
```

2. To view the results in again open paraview from your terminal using the command

```
paraFoam
```

3. This will Open up the paraview window as shown in the figure.

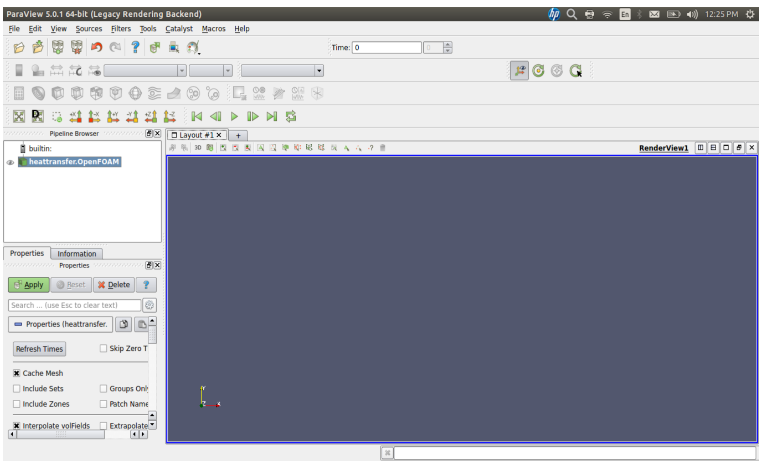


Figure 1: Paraview window

4. Now click on **Apply** in the pipeline browser to view the geometry. The geometry will be displayed as shown below

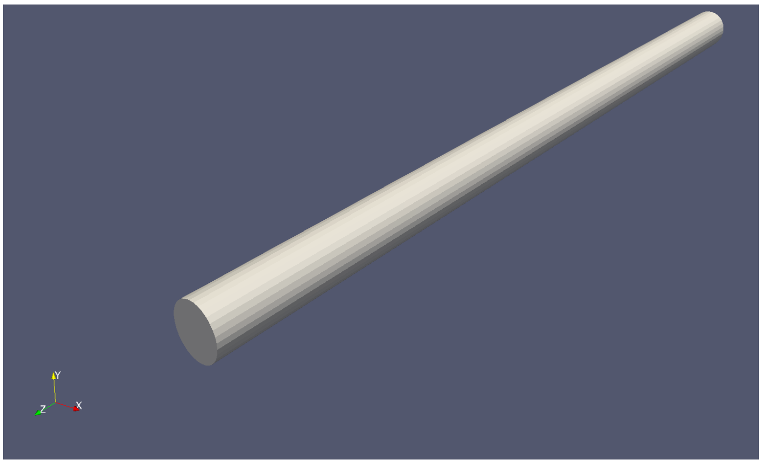


Figure 2: Solid model of the Geometry

5. In the drop down list select **Surface with Edges** to display the mesh as shown below.

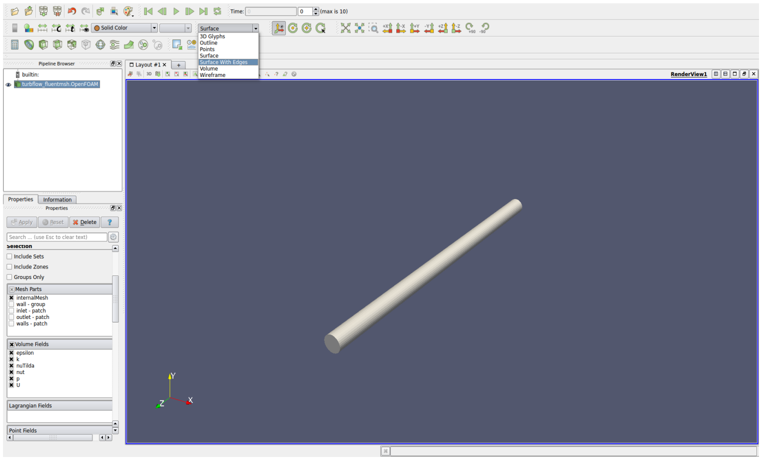


Figure 3: Mesh model of the Geometry

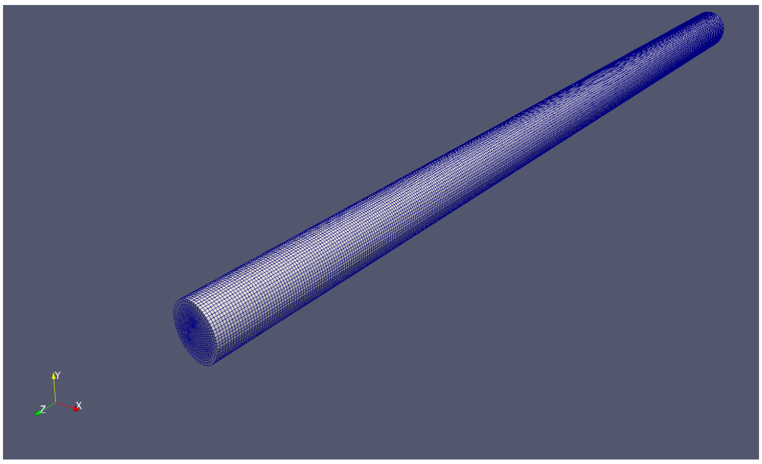


Figure 4: Mesh model of the Geometry

6. Go to **File** > **Save Screenshot** and give names according to your group number.
7. In the pipeline browser, by default **p** and **U** will selected. Check all the fields **epsilon**, **k**, **nuTilda**, **nut**.

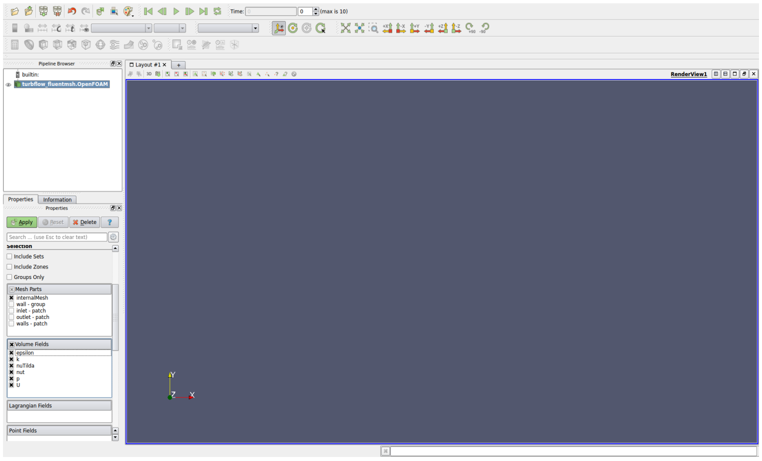


Figure 5: Mesh model of the Geometry

8. Now in the drop down list select **Surface**

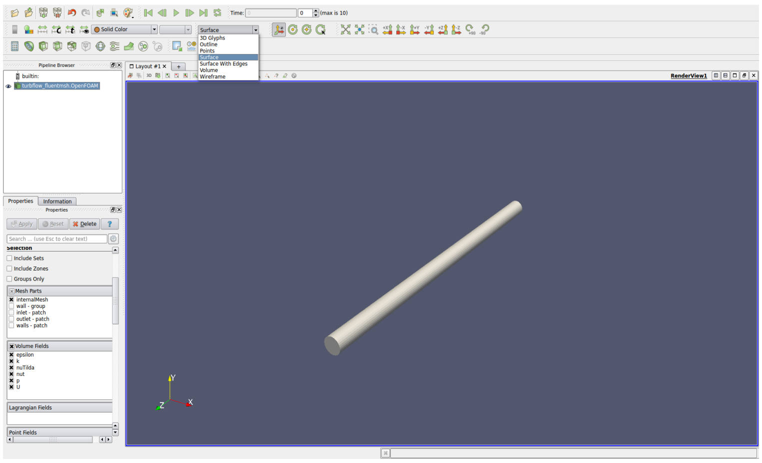


Figure 6: Mesh model of the Geometry

9. Select the **slice** menu and in pipeline browser select X-Normal.

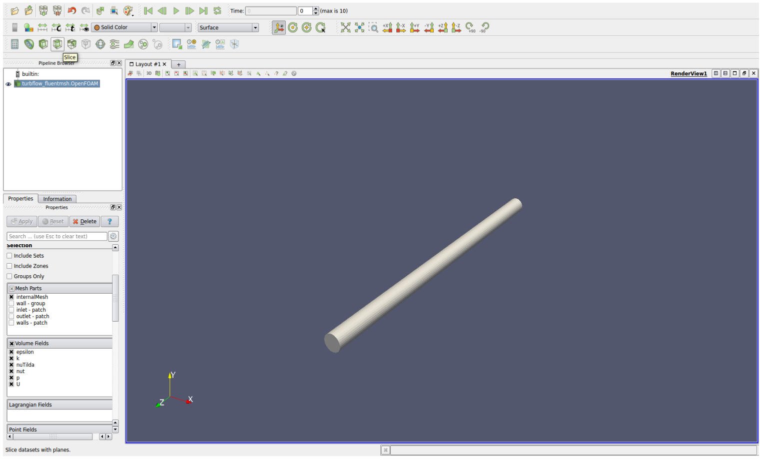


Figure 7: Mesh model of the Geometry

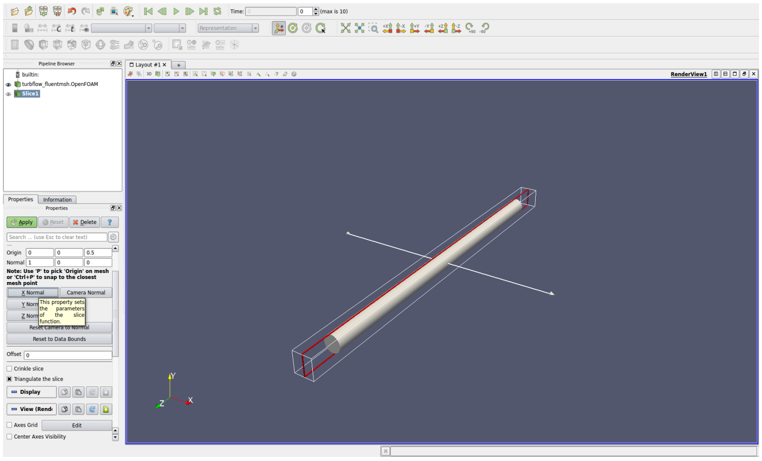


Figure 8: Mesh model of the Geometry

10. Click Apply
11. Now from the Drop down menu in the top menu bar select **U**.

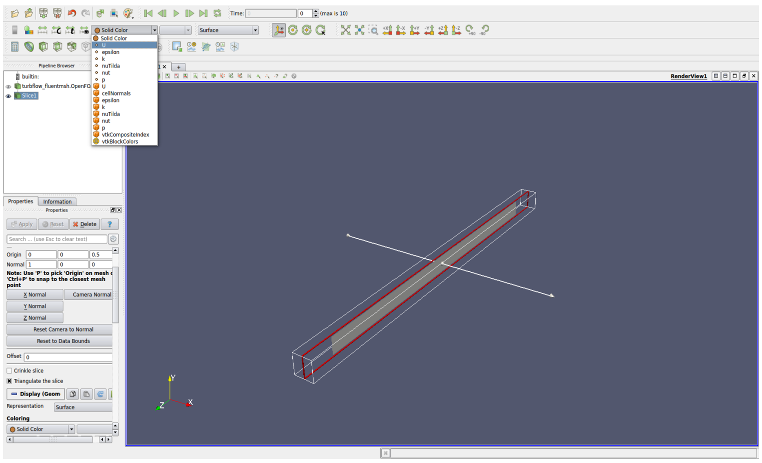


Figure 9: Mesh model of the Geometry

12. Now to see the velocity contour click on last time step option on the top menu bar in the VCR control as shown in figure below.

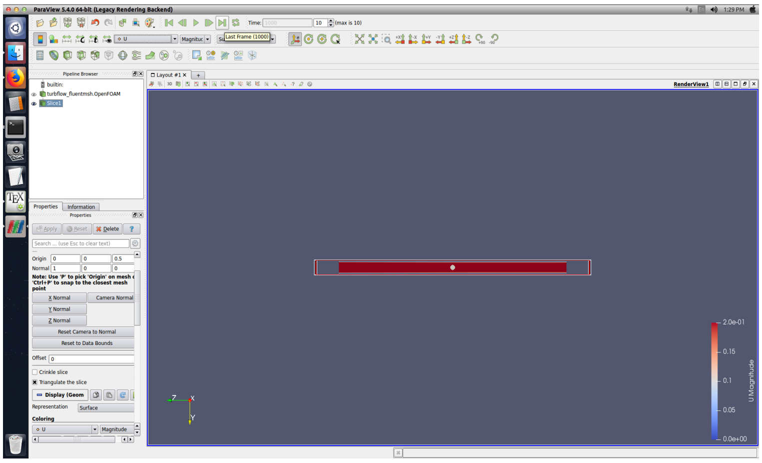


Figure 10: Mesh model of the Geometry

13. Now click on the button adjacent to the drop down menu which says (Rescale to data range) when you move the cursor over it.

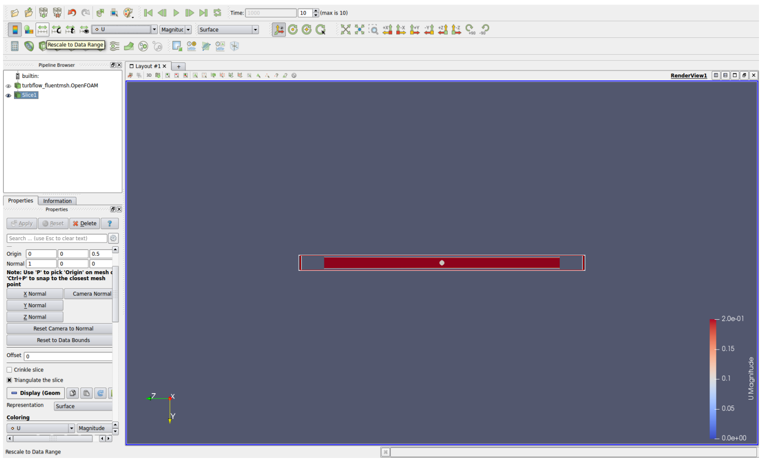


Figure 11: Mesh model of the Geometry

14. To fix the geometry in the proper orientation follow the images as given below.

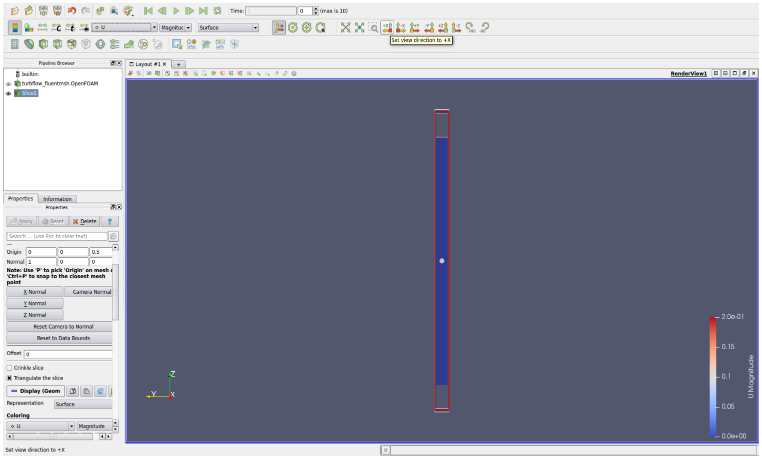


Figure 12: Mesh model of the Geometry

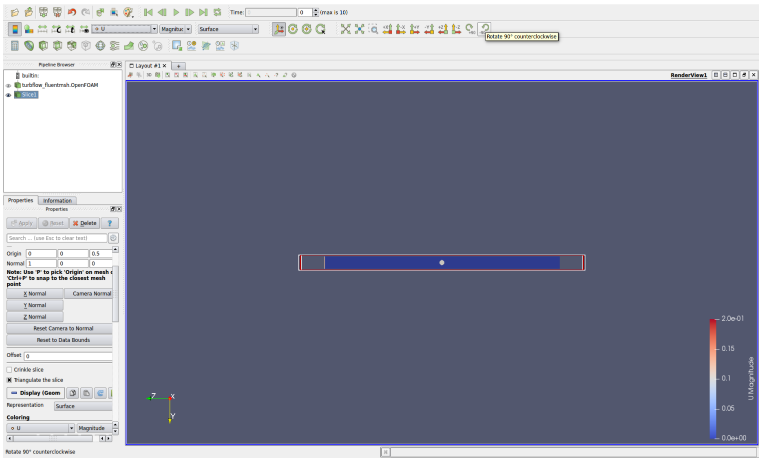


Figure 13: Mesh model of the Geometry

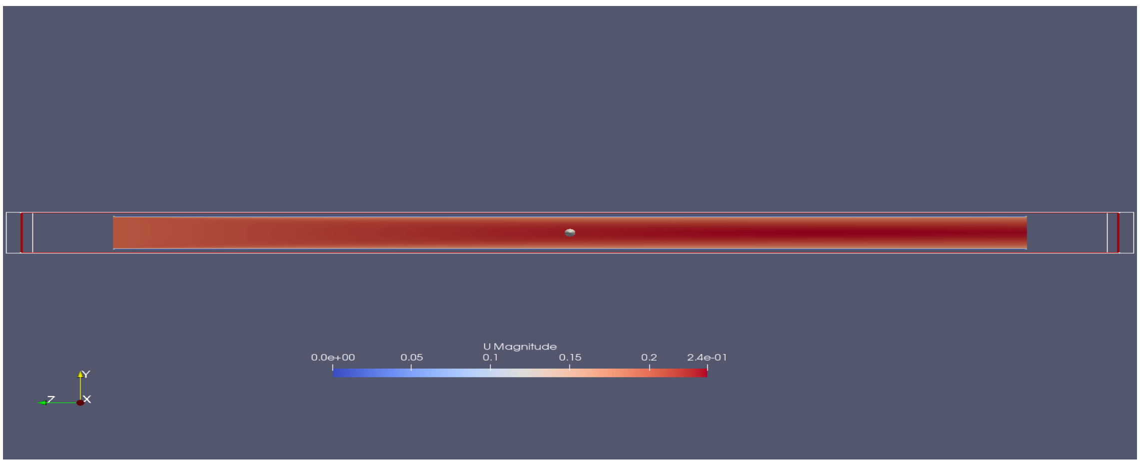


Figure 14: Velocity Contour

15. The velocity contour is shown above.

16. Take the screenshot.

17. Similarly select **p** from the drop down menu

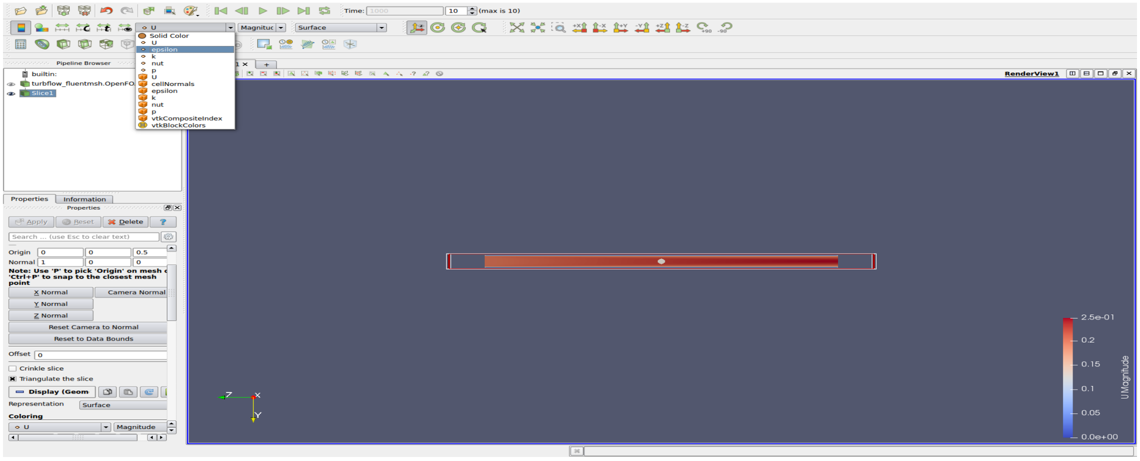


Figure 15: Pressure Contour

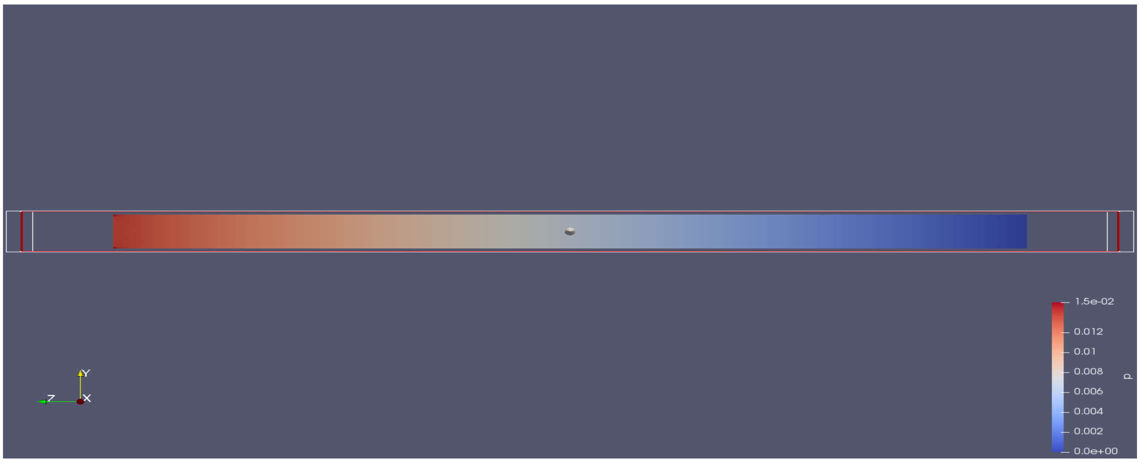


Figure 16: Pressure Contour

18. The velocity contour is shown above.

19. Take the screenshot.

20. Select **epsilon** from the drop down menu

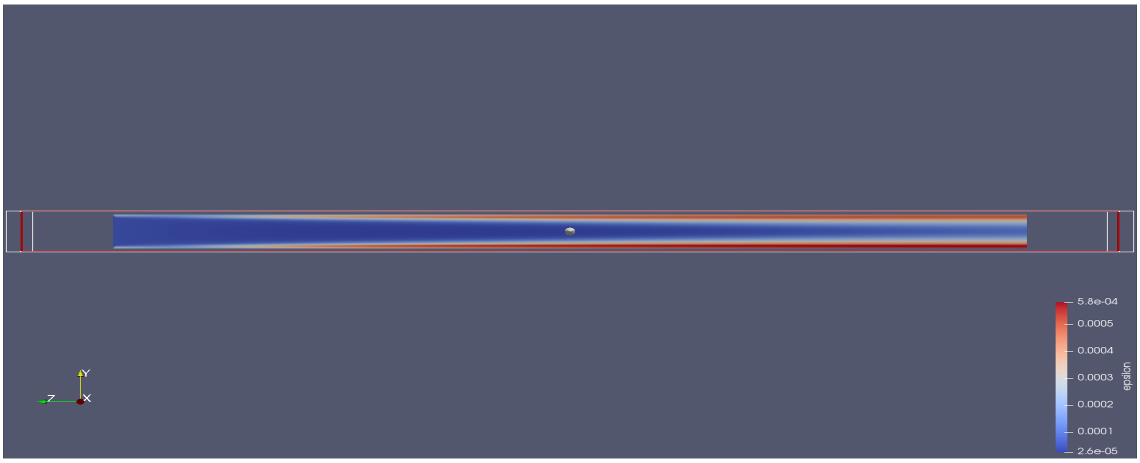


Figure 17: Turbulence Dissipation Rate Contour

21. The Turbulence Dissipation Rate is shown above.
22. Take the screenshot.
23. Select **k** from the drop down menu

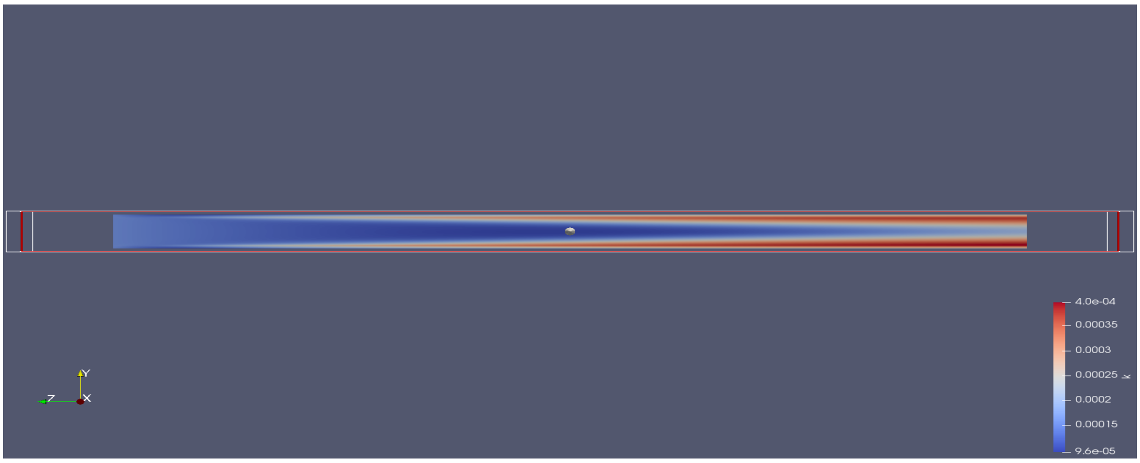


Figure 18: Turbulence Kinetic Energy Contour

24. The Turbulence Kinetic Energy Contour is shown above.
25. Take the screenshot.

Step-11

1. Now we need to plot the axial variation of velocity (along the length of the pipe).
2. To do so, on the left most top of the paraview window go to Filters → Data Analysis → Plot Over Line
3. Click on the Y-Axis on the **Pipeline Browser** and then scroll down

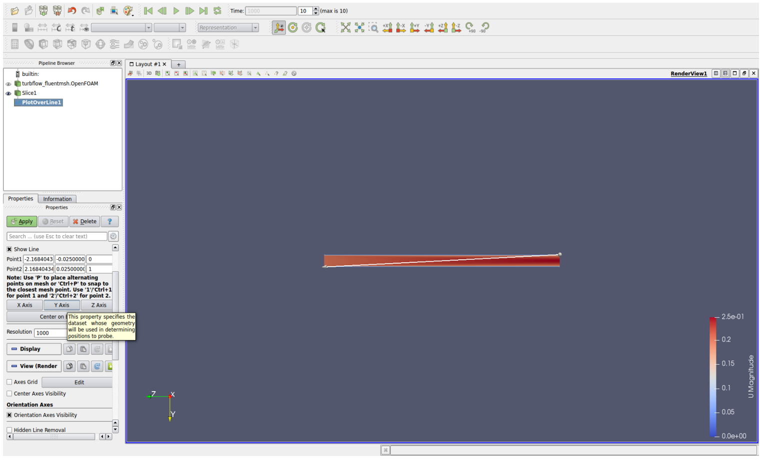


Figure 19: Mesh model of the Geometry

4. Click **Apply**.
5. Select **Umagnitude** in the pipeline browser and click Apply.

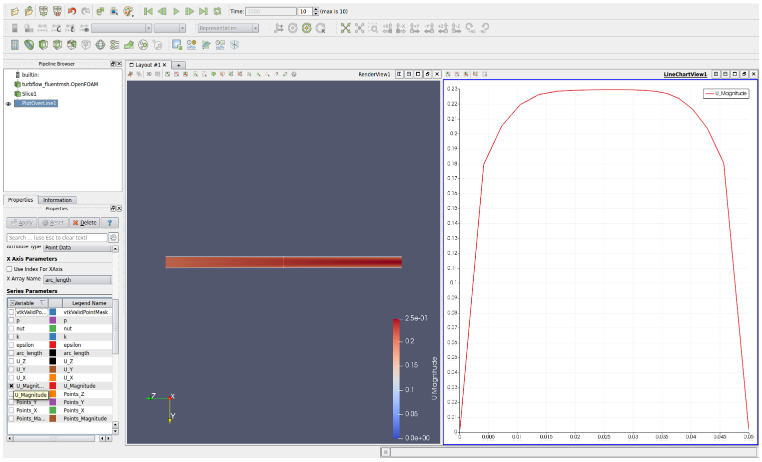


Figure 20: Mesh model of the Geometry

6. You will see the velocity profile across the radial direction of the pipe as shown below.

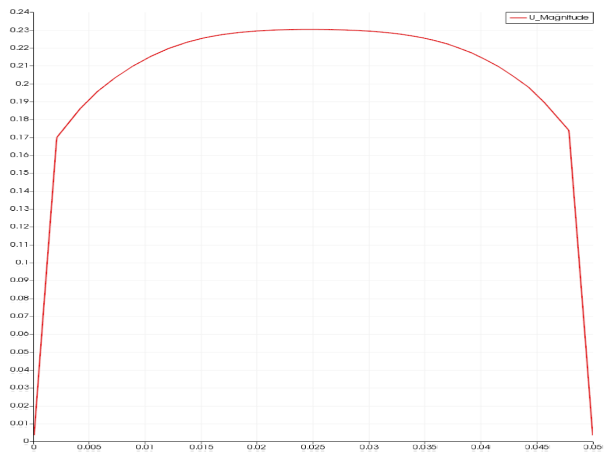


Figure 21: Radial Velocity Profile at the middle of the pipe

7. Take screenshot of the plot.

Step-12 (For the report)

1. Now we need to export the data of radial profile at the middle of the pipe.
2. In the VCR control option, go to the last time step.
3. On the left most top of the paraview window go to Filters → Data Analysis → Plot Over Line
4. Click on the Y-Axis on the **Pipeline Browser** and then scroll down
5. In the top side of the plot, Click the option **Split Vertical**.

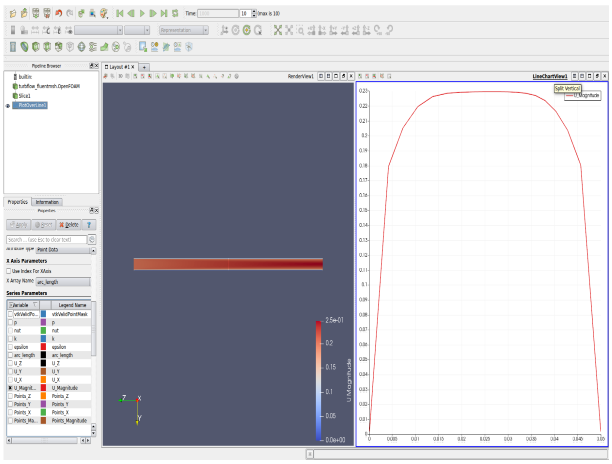


Figure 22: Radial Velocity Profile at the middle of the pipe

6. Create view window will appear in the bottom. Scroll down and click SpreadSheet view option.

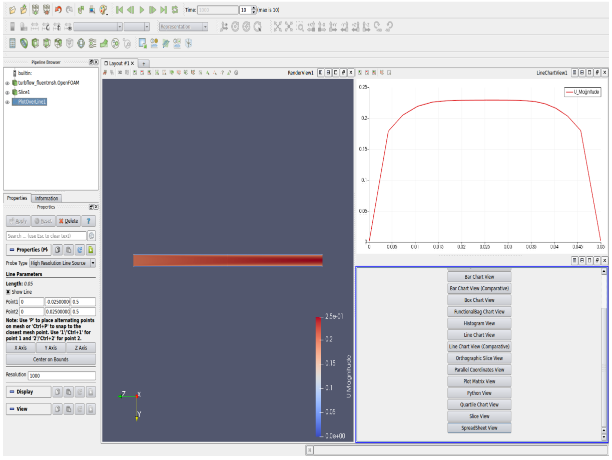


Figure 23: Radial Velocity Profile at the middle of the pipe

7. Now click File → Save Data

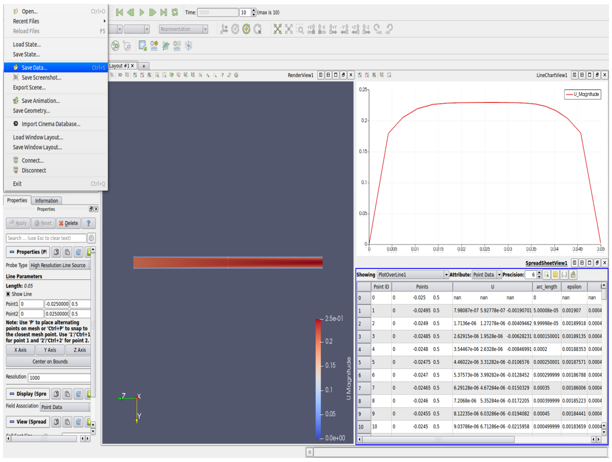


Figure 24: Radial Velocity Profile at the middle of the pipe

8. Give the name as

   turbpipe<GN>_div<1 or 2>_radial

   and save as the csv file.

9. Use this csv file and compare the results with analytical expression using Python script or LibreCalc.

10. The analytical expression for the radial profile of the axial velocity is given by

$$V_s = 1.25 * V_{avg} * \{1 - \frac{r}{R}\}^{\frac{1}{7}} \tag{6}$$

   where,
   V_z is the axial velocity.
   R is the radius of the pipe.
   V_{avg} is the average velocity in the pipe.

11. We can calculate the pressure drop along the length of the pipe with the analytical solution given by

$$\Delta p = f * \frac{\rho * u^2 * l}{2 * d} \tag{7}$$

where f is the friction factor. Using the following graph to find the value based on your Reynold's number.

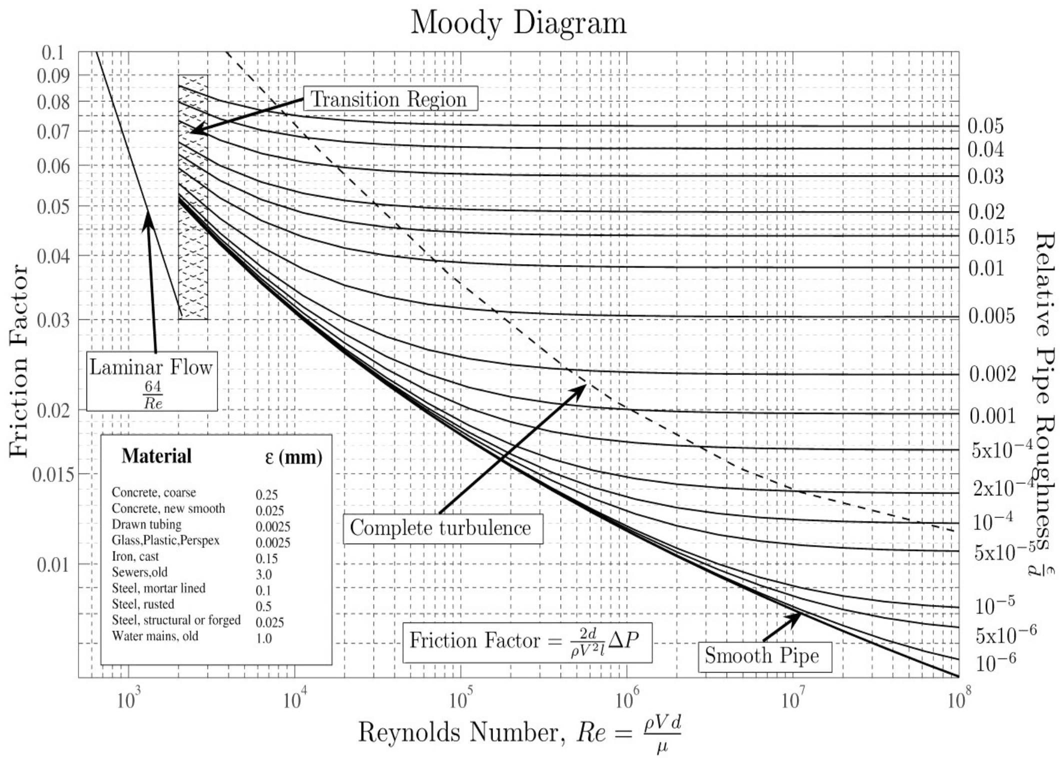


Figure 25: Moody diagram for estimating the Darcy friction factor

12. You can compare the pressure drop plot with the OpenFOAM results.
13. To do so, on the left most top of the paraview window go to Filters → Data Analysis → Plot Over Line
14. Click on the Z-Axis on the **Pipeline Browser** and then scroll down
15. In the top side of the plot, Click the option **Split Vertical**.
16. Create view window will appear in the bottom. Scroll down and click SpreadSheet view option.
17. Now click File → Save Data
18. Give the name as

```
turbpipe<GN>_div<1 or 2>_axial
```

and save as the csv file.

19. Use this csv file and compare the results with analytical expression using Python script or LibreCalc.

20. The Pressure drop along the length of the pipe is shown below

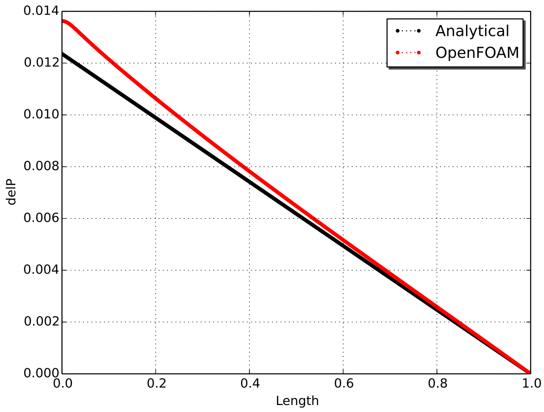


Figure 26: Pressure Drop along the length of the pipe

21. The radial profile of the axial velocity at middle of pipe is shown below

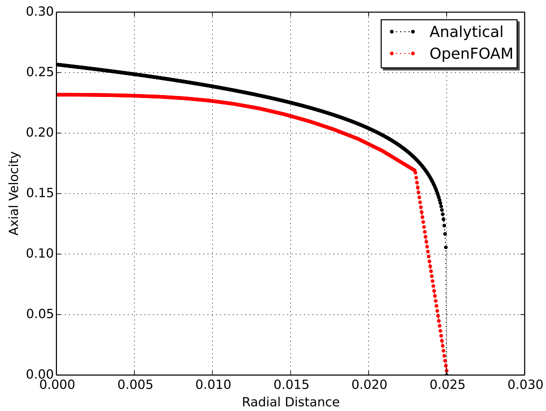


Figure 27: Radial Velocity Profile at the middle of the pipe

Python code for comparison of Pressure drop along the length of the pipe

```
import numpy as np
from matplotlib import pyplot as plt
import csv
with open('test.csv') as csvfile:
    readCSV = csv.reader(csvfile, delimiter=',')
    next(readCSV)
    p=[];Ux=[];Uy=[];Uz=[];l=[];usq=[];vsq=[];wsq=[];P=[]

    for row in readCSV:


            p=np.append(p,row[0])
            Ux=np.append(Ux,row[1])
            Uy=np.append(Uy,row[2])
            Uz=np.append(Uz,row[3])
            l=np.append(l,row[5])
U = [float(i) for i in Ux]
V = [float(j) for j in Uy]
W = [float(k) for k in Uz]
l = [float(m) for m in l]

def Reverse(lst):
    return [ele for ele in reversed(lst)]

P = [float(m) for m in p]
P = Reverse(P)
usq = [x**2 for x in U]
vsq = [y**2 for y in V]
wsq = [z**2 for z in W]

#Umag= [sum(x) for x in zip(usq, vsq, wsq)]

f=0.0309  #Darcy Friction factor for Re = 10,000 from Moody chart
rho=1000.0 #Density of Water
d=0.05  #Diameter of pipe
velocity=0.2
factor=((f*rho*velocity*velocity)/(2*d))/(rho)  #we are dividing by rho fo
result = [x * factor for x in l]


result = Reverse(result)


plt.figure(1)
plt.plot(l, result, 'k.:',label='Analytical')
```

```
plt.plot(l, P, 'r.:',label='OpenFOAM')
legend = plt.legend(loc='upper right', shadow=True)
plt.ylabel('$\Delta p$')
plt.xlabel('Length of the pipe')
plt.grid('on')
plt.title('Pressure Drop along the length of the pipe')
plt.savefig('Pressure_drop_comparison.eps')
plt.show()
```

Python code for comparison of Radial profile of axial velocity at middle of the pipe

```
import numpy as np
from matplotlib import pyplot as plt
import csv
with open('radial.csv') as csvfile:
    readCSV = csv.reader(csvfile, delimiter=',')
    p=[];Ux=[];Uy=[];Uz=[];r=[];

    for row in list(readCSV)[2:502]:


            p=np.append(p,row[0])
            Ux=np.append(Ux,row[1])
            Uy=np.append(Uy,row[2])
            Uz=np.append(Uz,row[3])
            r=np.append(r,row[5])


#Analytical
r = [float(m) for m in r]    #Radial Distance
Uz = [float(n) for n in Uz]
d=0.05                        #Diameter of the pipe
R=d/2.0                       #Radius of the pipe
a = np.divide(r, R)
a= 1.0 - a
b=0.142857142                  #(1/7)
def Average(lst):
    return sum(lst) / len(lst)
U = Average(Uz)                #Average radial velocity at mid section
bracket = [x**b for x in a]
factor = 1.25*U
result = [x * factor for x in bracket]
result=np.negative(result)


#OpenFOAM
W = [float(k) for k in Uz]
```

```python
Uz = np.negative(W)
def Reverse(lst):
    return [ele for ele in reversed(lst)]
Uz = Reverse(Uz)


plt.figure(1)
plt.plot(r, result, 'k.:',label='Analytical')
plt.plot(r, Uz, 'r.:',label='OpenFOAM')
legend = plt.legend(loc='upper right', shadow=True)
plt.ylabel('Axial Velocity (m/s)')
plt.xlabel('Radial Distance (m)')
plt.grid('on')
plt.savefig('Axialvel_at Radiallocation_comparison.eps')
plt.show()
```