OpenFOAM

An Open source alternative to commercial CFD Package

Instruction Manual for Solving Turbulent Flow in a Horizontal Pipe

Prepared By:

CFD Team

FOSSEE

Indian Institute of Technology, Bombay

December 2017

Objective

To carry out CFD simulation using OpenFOAM, to obtain the velocity and pressure contours in the pipe.

To obtain the axial and radial velocity profile in the pipe.

Geometry

Straight horizontal pipe

Radius of the rod (r) = 0.5 cm

Length of the rod (L) = 30 cm

Rod is aligned along the z-axis

Flow Conditions

Fluid to be used in simulations: Air

Velocity at the inlet = 3 m/s

Density of the air = $1.225 kg/m^3$

Step-1

1. Press **ctrl**+ **Alt**+**t** to open the terminal
2. Now you are in the home directory of your user.
3. To create the case folder type

   ```
   mkdir pipe
   ```

Step-2

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 Incompressible Turbulent flow** inside the pipe, we choose **simpleFoam** solver.
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/pipeCyclic
   ```

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 by typing the following command.

```
cp -r 0 constant system /home/test<userId>/pipe
```

Copying the mesh file

1. Delete the existing blockMeshDict in the system folder.

```
cd pipe/system
rm blockMeshDict
```

2. Now we can copy the blockMeshDict file made for our problem to the system directory.
3. The mesh is made using **blockMeshDict**,which is an OpenFoam Meshing Utility.
4. Copy the **blockMeshDict**file to your case directory by following command

```
cd ..
cd ..
cd opt/pipemesh
cp blockMeshDict /home/test<userId>/pipe/system
```

5. The patch names defined in the mesh are **inlet**, **outlet** and **walls**.

Calculation of Flow 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. Velocity has been given as 3 m/s.
3. Now calculate the Turbulent Kinetic Energy (k) based on the inlet velocity as given below

$$k = \frac{3}{2}(V * I)^2 \tag{1}$$

where I is the turbulence intensity which is 1% for this case as given in specification.

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

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

where l is the turbulence length scale which is $l_{sc} = 0.07 * DimaterofPipe$.
$C_\mu = 0.09$

5. Calculate the mass flow rate

$$\dot{m} = \rho AV \tag{3}$$

where $\rho = 1.225kg/m^3$ and A = Cross sectional area of pipe.

Setting Boundary Conditions

1. Now we need to edit the **0** folder in our case directory.

2. To open the case directory

   ```
   cd
   cd pipe
   cd 0
   ```

3. Open the U file.

   ```
   gedit U
   ```

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

5. 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];
   ```

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

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

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

8. We will given Mass flow rate velocity inlet.

9. The boundary patches are edited as shown below.

   ```
   inlet
       {
           type                        flowRateInletVelocity;
           massFlowRate                constant <Calculated Value>;
           extrapolateProfile          no;
           rhoInlet                    1.225;
           value                       uniform (0 0 0);

       }

   outlet
       {
           type                zeroGradient;
       }

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

10. Save it and close the file.

11. Open the p file.

    ```
    gedit p
    ```

12. The **p** will be calculated from p_rgh, it doesn't matter what value we give in p file.

13. Still we maintain same as p_rgh file.

14. keep the dimensions for the file as default.

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

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

    ```
    internalField     uniform 0;
    ```

16. The boundary patches are edited as shown below.

    ```
    inlet
        {
            type                zeroGradient;
        }

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

17. Save it and close the file.

18. Open the k file.

    ```
    gedit k
    ```

19. 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];
    ```

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

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

21. The boundary patches are edited as shown below.

```
inlet
    {
        type            fixedValue;
        value           uniform <calculated value>;
    }
    outlet
    {
        type            zeroGradient;
    }
    walls
    {
        type            kqRWallFunction;
        value           uniform <calculated value>;
    }
```

22. Save it and close the file

23. Open the epsilon file.

    ```
    gedit epsilon
    ```

24. 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];
    ```

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

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

26. The boundary patches are edited as shown below.

```
inlet
    {
        type            fixedValue;
        value           uniform <calculated value>;
    }
    outlet
    {
        type            zeroGradient;
    }


walls
    {
        type            epsilonWallFunction;
        value           uniform <calculated value>;
    }
```

27. Save it and close the file

28. Open the nut file.

    ```
    gedit nut
    ```

29. keep the dimensions for the file as default.

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

30. We keep the same conditions for this file. Just change the patch name

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

    ```
    internalField     uniform 0;
    ```

32. 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;
        }
    ```

33. Save it and close the file

Setting Physical Properties

1. Now edit the constant directory.

    ```
    cd
    cd pipe/constant/
    ```

2. Open the **transportProperties** file.

    ```
    gedit transportProperties
    ```

3. Edit the **nu** value (kinematic viscosity). i.e Divide the dynamic viscosity of the air by the density of the air.

    ```
    nu                  [0 2 -1 0 0 0 0] <calculated value>;
    ```

4. Do not change the other values.
5. Save it and close the file.
6. In the **turbulenceProperties** file, by default **kEpsilon** would have been given. Leave it as such and close the constant folder.

Setting Simulation Parameters

1. Now edit the system directory.

   ```
   cd
   cd pipe/system/
   ```

2. Open **controlDict** file.

   ```
   gedit controlDict
   ```

3. Change the **endTime** as 2000.

   ```
   endTime          2000;
   ```

4. Save it and close the file.
5. Open the **fvSolution** file.

   ```
   gedit fvSolution
   ```

6. Edit the relaxationFactors as shown below.

   ```
   relaxationFactors
   {
       fields
       {
           p              0.3;
       }
       equations
       {
           U              0.3;
           k              0.3;
           "epsilon.*"    0.3;
       }
   }
   ```

7. Save it and close the file.
8. Close the system folder.

Meshing

1. Type the following command to create the mesh.

   ```
   cd
   cd pipe
   blockMesh
   ```

2. Mesh will be created.

3. Type the following command to check the mesh

   ```
   checkMesh
   ```

4. Save it and close the file

Solving

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

2. Now we need to mesh the case.

3. Type the following

   ```
   cd
   cd pipe
   blockMesh
   ```

4. To run the case.

5. Type the following

   ```
   cd
   cd pipe
   simpleFoam
   ```

6. The iterations will be running slow wait will the specified convergence/timeStep is reached.

Post-Processing

1. Open the case folder.

   ```
   cd
   cd pipe
   ```

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

   ```
   paraFoam
   ```

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

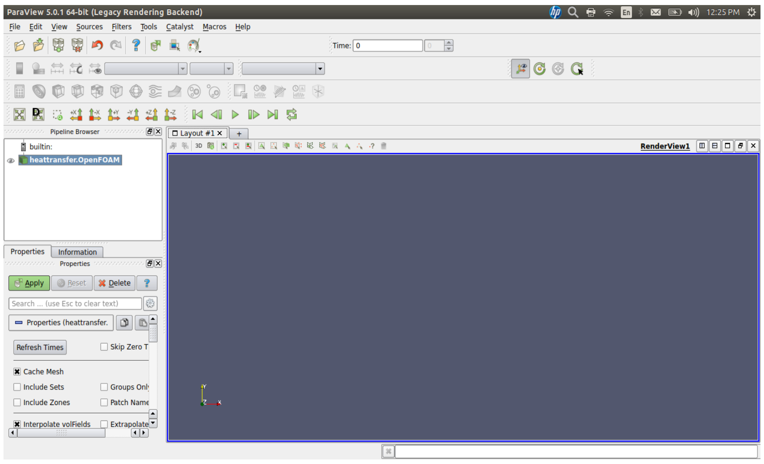

Figure 1: Paraview window

4. In the pipeline browser, by default p and U will selected. Check all the properties except T, alphaT and nuT.
5. Click Apply to view the geometry
6. Select the **slice** menu and in pipeline browser select Z-Normal.
7. Click Apply
8. Now from the Drop down menu in the top menu bar select **U**.
9. Now to see the velocity contour click on the play button on the top menu bar.
10. Now click on the button adjacent to the drop down menu which says (Rescale to data range) when you move the cursor over it.

Plotting Profile in paraFoam

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 Z-Axis on the **Pipeline Browser** and then scroll down
4. Select velocity in the pipeline browser and click Apply.
5. You will see the velocity profile along the length of the vessel as shown below.

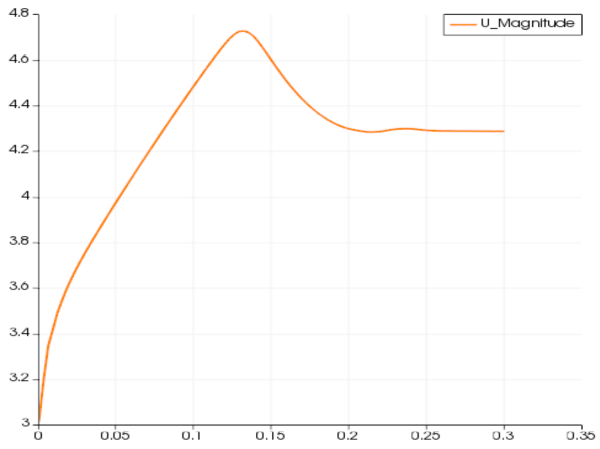

Figure 2: Axial Variation of velocity

6. Now we need to plot the variation of velocity along radius of the pipe.
7. To do so, on the left most top of the paraview window go to Filters → Data Analysis → Plot Over Line
8. Click on the Y-Axis on the **Pipeline Browser** and then scroll down
9. Select velocity in the pipeline browser and click Apply.
10. You will see the velocity profile along the radius of the pipe as shown below.

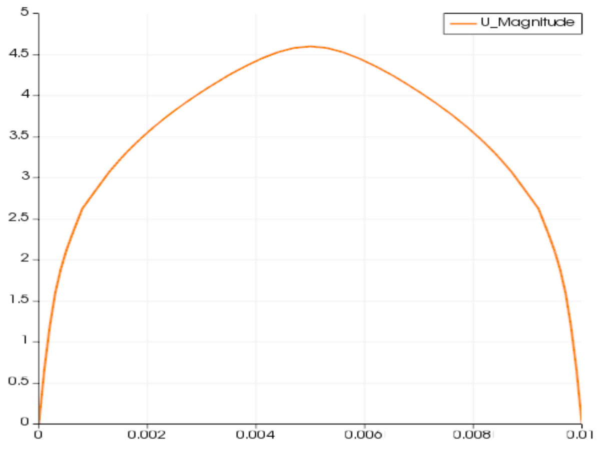

Figure 3: Axial Variation of velocity

Parallel Simulation

1. We delete all the time directories created during the previous simulation.

```
cd
cd pipe
mv 0 org
rm -r [0-9]*
```

2. To run the simulation in parallel, copy the decomposeParDict file in the system directory.

   ```
   cd ..
   cd ..
   cd opt/parallel
   cp decomposeParDict /home/test<GN>/pipe/system
   ```

3. Parallel simulation will divide the domain and do parallel simulation for each domain. In the file provided, the pipe geometry will be divided in to four regions and four processors will be used for simulation.

   ```
   decomposePar
   ```

4. To run the simulation, type

   ```
   mpirun−np 4 simpleFoam −parallel > log &
   ```

5. Once the similation completed, type

   ```
   reconstructPar
   ```

6. For more details about decomposition of mesh https://cfd.direct/openfoam/user-guide/running-applications-parallel/

Plotting Residuals

1. We delete all the time directories created during the previous simulation.

   ```
   cd
   cd pipe
   mv 0 org
   rm −r [0−9]*
   rm −r processor*
   ```

2. The residuals of the simulation during runtime can be plotted using gnuplot. (makesure gnuplot has been installed in the system)

3. Copy the residuals file in to the case directory.

   ```
   cd ..
   cd ..
   cd opt/gnuplot
   cp residuals /home/test<GN>/pipe
   ```

4. While running the case

   ```
   cd
   cd pipe
   nohup simpleFoam > log &
   gnuplot residuals
   ```