Department of Chemical Engineering, IIT Bombay
Instruction Manual for Solving Turbulent Flow in a Pressure Vessel using OpenFOAM
CFD Lab, Semester-I, Year 2017-18

Turbulent Flow in a Pressure Vessel

Objective

To carry out CFD simulation using OpenFOAM, to obtain the velocity and pressure contours of pressure vessel aligned vertically.

The turbulent flow is simulated with the addition of gravity term due to vertical alignment of the vessel.

The mesh for the 3d geometry is created using **GMsh** and the mesh file is provided.

Geometry and Conditions

Length of the vessel = 100 cm

Diameter of the vessel = 50 cm

Axis along +x direction

Diameter of the inlet and the outlet pipes = 5 cm

Length of inlet and outlet pipes = 9 cm

Fluid to be used in simulations: water

Reynolds number based on vessel diameter (50 cm) = 2500+100*(n-1) where n is the group number

Turbulent Intensity = 10%

Turbulent length Scale = 0.001 m

Step-1

Group Numbers (GN)

```
For Group Numbers 1 to 15 : Actual Group Number
For Group Numbers 16 to 30 : Actual Group Number - 15
For Group Numbers 31 to 45 : Actual Group Number - 15
For Group Numbers 46 to 63 : Actual Group Number - 30
```

1. Press **ctrl+ Alt+T** to open the terminal
2. Type the following command to connect to the server.

```
ssh test<GN>@10.102.1.121 –X (for groups 01−15)
Password:test<GN>
ssh test<GN>@10.102.1.122 –X (for groups 16−30)
Password:test<GN>
ssh test<GN>@10.102.1.121 –X (for groups 31−45)
Password:test<GN>
ssh test<GN>@10.102.1.122 –X (for groups 46−63)
Password:test<GN>
```

3. Now you are in the home directory of your user.

4. To create the case folder type

```
For Division 1
mkdir pressurevessel<GN>_div1

For Division 2
mkdir pressurevessel<GN>_div2
```

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 vessel with the addition of buoyancy term, we choose **buoyantBoussinesqSimpleFoam** solver.

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

```
cd $FOAM_TUTORIALS
cd heatTransfer/buoyantBoussinesqSimpleFoam/hotRoom
```

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<GN>/pressurevessel<GN>_div<1 or 2>
```

Step-3

1. The mesh is made up of Unstructured elements using **GMsh**,which is an Open Source meshing software.

2. Copy the mesh file to your case directory by following command

```
cd ..
cd ..
cd opt
cp pressurevessel.msh /home/test<GN>/pressurevessel<GN>_div<1 or 2>
```

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

Step-4

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. Calculate the Velocity (m/s) based on the Reynold's number.

$$Re_D = \frac{\rho V D}{\mu} \tag{1}$$

where $\rho = 1000kg/m^3$, $\mu = 0.798 * 10^{-3} \frac{Ns}{m^2}$ and D = Vessel Diameter (m) (50 cm)

3. Now calculate the mass flow rate

$$\dot{m} = \rho A V \tag{2}$$

4. Since the mass flow rate will be constant according to mass conservation, apply the calculated mass flow rate at the inlet and find the inlet velocity.

$$\dot{m} = \rho A_i V_i \tag{3}$$

where Inlet Diameter is 5 cm

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 which is 10% for this case as given in specification.

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 the turbulence length scale which is 0.001 m for this case as given in specification.
$C_\mu = 0.09$

Step-5

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

   ```
   cd
   cd pressurevessel<GN>_div<1 or 2>
   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. The boundary patches are edited as shown below.

   ```
   inlet
       {
           type                fixedValue;
           value               uniform (<Calculated Value> 0 0);
       }

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

9. Save it and close the file.
10. Open the p_{rgh} file.

    ```
    gedit p_rgh
    ```

11. In incompressible flow solver in OpenFoam, the actual pressure will be divided by density and corresponding dimensions should be used.

12. keep the dimensions for the file as default.

```
dimensions          [0  2  −2  0  0  0  0];
```

13. The boundary patches are edited as shown below.

```
inlet
    {
        type                zeroGradient;
    }

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

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

```
internalField     uniform  0;
```

15. Save it and close the file.

16. Open the p file.

```
gedit  p
```

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

18. Still we maintain same as p_rgh file.

19. keep the dimensions for the file as default.

```
dimensions          [0  2  −2  0  0  0  0];
```

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

```
internalField     uniform  0;
```

21. The boundary patches are edited as shown below.

```
inlet
    {
        type                zeroGradient;
    }

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

22. Save it and close the file.

23. Open the k file.

    ```
    gedit k
    ```

24. 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];
    ```

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                kqRWallFunction;
            value               uniform <calculated value>;
        }
    ```

27. Save it and close the file

28. Open the epsilon file.

    ```
    gedit epsilon
    ```

29. 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];
```

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

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

31. 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>;
    }
```

32. Save it and close the file

33. Now rename the T file.

```
mv T.orig T
```

34. For the temperature, alphat and nut files we give the default values as zero, since we are not solving heat transfer case.

35. Open the T file.

```
gedit T
```

36. keep the dimensions for the file as default.

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

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

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

```
internalField     uniform 0;
```

39. The boundary patches are edited as shown below.

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

40. Save it and close the file

41. Open the alphat file.

    ```
    gedit alphat
    ```

42. keep the dimensions for the file as default.

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

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

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

    ```
    internalField    uniform 0;
    ```

45. The boundary patches are edited as shown below.

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


        walls
        {
            type                alphatJayatillekeWallFunction;
            Prt                 0.7;
            value               uniform 0;
        }
    ```

46. Save it and close the file
47. Open the nut file.

```
gedit nut
```

48. keep the dimensions for the file as default.

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

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

```
internalField     uniform 0;
```

51. 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;
    }
```

52. Save it and close the file

Step-6

1. Now we create the mesh file for our case using the **pressurevessel.msh** file
2. Type the following command to create the mesh.

```
cd
cd pressurevessel<GN>_div<1 or 2>
gmshToFoam pressurevessel.msh
```

3. Mesh will be created.
4. Type the following command to check the mesh

```
checkMesh
```

5. Now open the **polymesh** folder and edit the patch name for walls.

   ```
   cd
   cd pressurevessel<GN>_div<1 or 2>/constant/polyMesh
   gedit boundary
   ```

6. In the walls boundary, instead of the **type** as **patch**, edit it as **wall** and remove the line **physicalType**. It should be like

   ```
   walls
       {
           type            wall;

           nFaces          6478;
           startFace       84973;
       }
   ```

7. Save it and close the file

Step-7

1. Now edit the constant directory.

   ```
   cd
   cd pressurevessel<GN>_div<1 or 2>/constant/
   ```

2. Open **g** file.

   ```
   gedit g
   ```

3. Since the length of vessel is aligned along +x axis, give the value as

   ```
   (-9.81 0 0);
   ```

4. Save it and close the file.

5. Open the **transportProperties** file.

   ```
   gedit transportProperties
   ```

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

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

7. Do not change the other values.

8. Save it and close the file.

9. In the **turbulenceProperties** file, by default **kEpsilon** would have been given. Leave it as such and close the constant folder.

Step-8

1. Now edit the system directory.

   ```
   cd
   cd pressurevessel<GN>_div<1 or 2>/system/
   ```

2. Remove the blockMeshDict file.

   ```
   rm blockMeshDict
   ```

3. Open **controlDict** file.

   ```
   gedit controlDict
   ```

4. Change the **endTime** as 5000.

   ```
   endTime          5000;
   ```

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

   ```
   gedit fvSolution
   ```

7. Edit the residual control values as shown below

   ```
   SIMPLE
   {
       nNonOrthogonalCorrectors 0;
       pRefCell          0;
       pRefValue         0;

       residualControl
       {
           p_rgh             1e-5;
           U                 1e-5;
           T                 1e-2;

           // possibly check turbulence fields
           "(k|epsilon|omega)" 1e-5;
       }
   }
   ```

8. Edit the relaxationFactors as shown below.

   ```
   relaxationFactors
   {
       fields
       {
           p_rgh             0.3;
       }
   ```

```
    equations
    {
        U                   0.3;
        T                   0.5;
        "(k|epsilon|R)"     0.3;
    }
}
```

9. Save it and close the file.
10. Close the system folder.

Step-9

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 pressurevessel<GN>_div<1 or 2>
   buoyantBoussinesqSimpleFoam
   ```

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

Step-10

1. Open the case folder.

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

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.

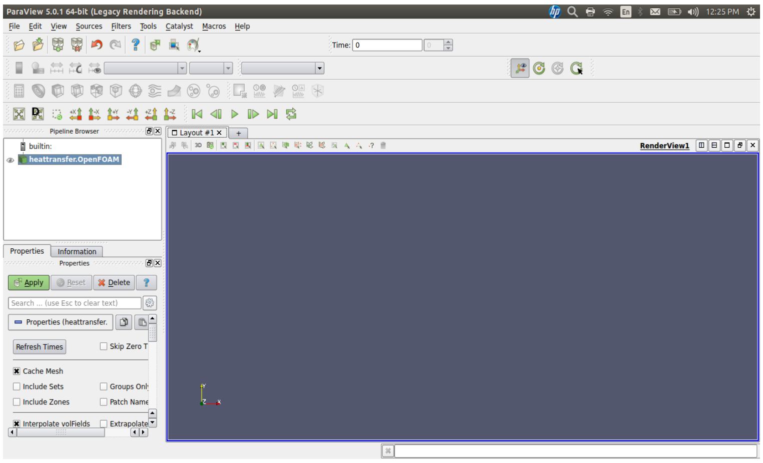


Figure 1: Paraview window

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 as shown in figure below.
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.

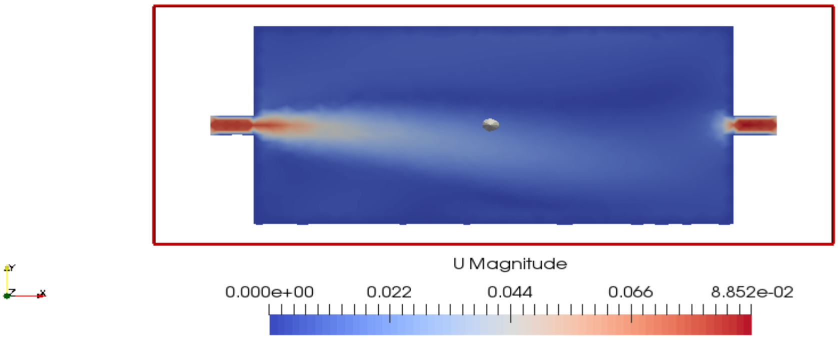
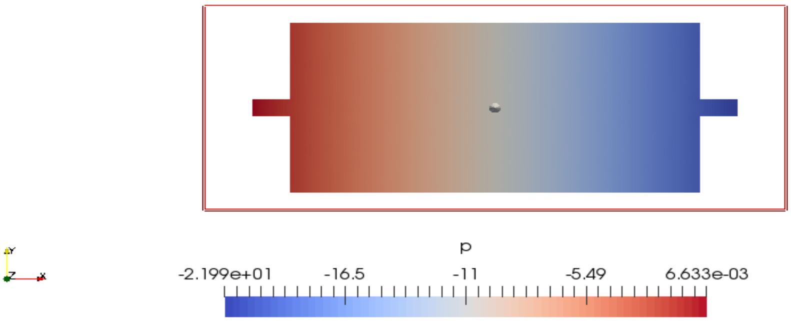


Figure 2: Paraview window

11. Take the screenshot.

12. Similarly select **p** from the drop down menu and save the screenshot of pressure contour.



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 X-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.

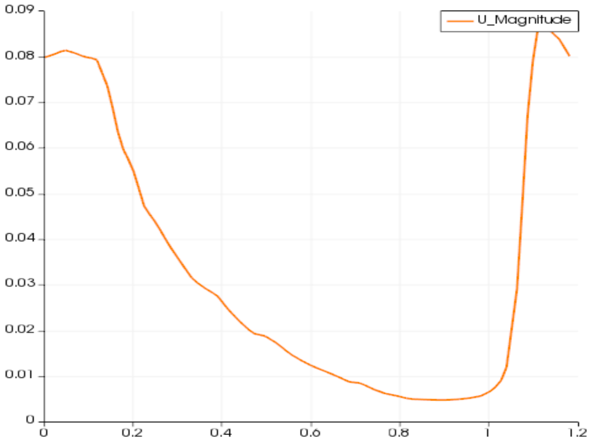


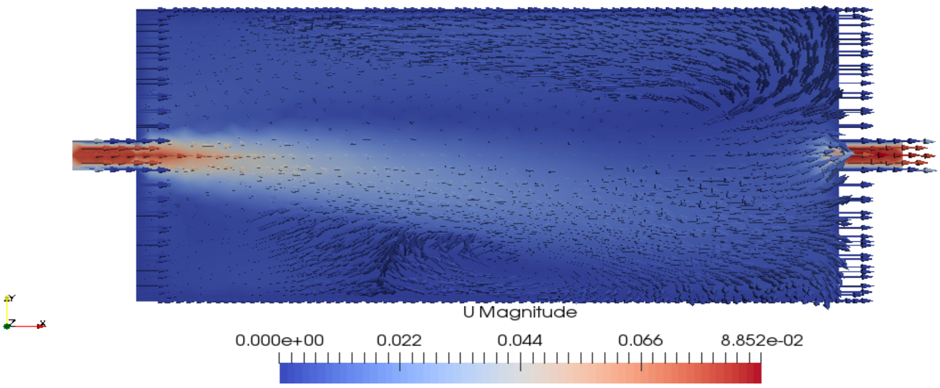
Figure 3: Axial Variation of velocity

6. Take screenshot of the plot.

Step-12

1. Now we need to plot the velocity streamlines.

2. To do so, select the **slice** menu and in pipeline browser select Z-Normal.
3. Click Apply.
4. Now click **Glyph** icon on the menu bar.
5. In the pipeline browser, select U in vectors.
6. Click Apply.
7. In the scale factor give **0.05**
8. Click Apply.



9. Take screenshot of the image.
10. This brings us to the end of this tutorial.