Department of Chemical Engineering, IIT Bombay

Instruction Manual for solving Turbulent Flow and obtain Residence Time Distribution(RTD) plots in a Pressure Vessel using OpenFOAM

CFD Lab, Autumn Semester, Year 2019-20

Session-1 : Meshing and setting up the Case Directory

Objective

To mesh the 2-D axisymmetric pressure vessel geometry using blockMesh (OpenFOAM utility for creating mesh).

Setting up the case directory to solve the turbulent flow through the pressure vessel.

Setting up the case directory to solve and obtain residence time distribution plots for the pressure vessel.

Geometry and Conditions

Length of the vessel = **100 cm**.

Diameter of the vessel = **50 cm**.

Axis along Z-axis

Length of inlet pipe = **5 cm**.

Length of outlet pipe = **5 cm**.

Fluid : **Liquid Water**. Density = 1000 kg/m^3

Reynold's number based on the vessel diameter(50 cm) = **2500 +100*(n-1)**, where n is your group number.

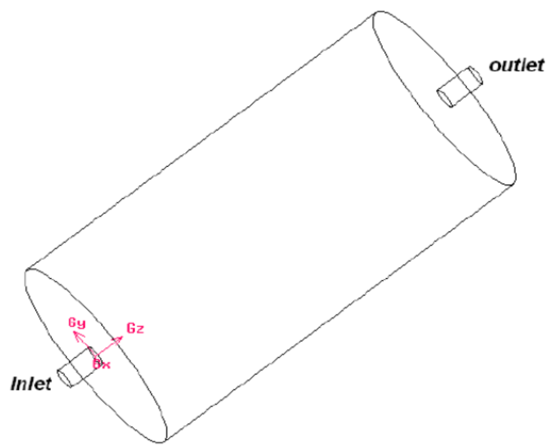

Figure 1: 3-D view of the pressure vessel geometry

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.**

Modification in Geometry

1. To simplify the calculation we will make this as axisymmetric wedge.

2. In OpenFOAM to make the axisymmetric wedge the angle should be **5 degrees**.

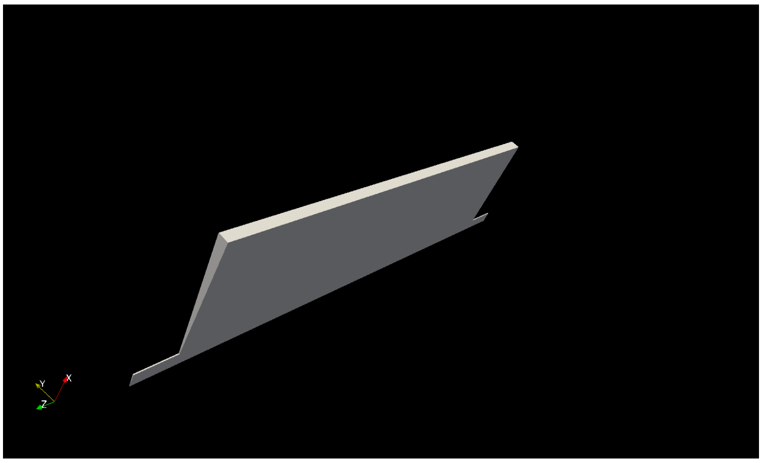

Figure 2: Axisymmetric pressure vessel geometry

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 pressurevessel<GN>_div1

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

For example

```
mkdir pressurevessel03_div1    (For Group-3)
mkdir pressurevessel38_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. The pressure vessel is aligned vertical and inlet is from the bottom. Therefore we need to add the effect of **gravity** to the simpleFoam solver, which we will do in the next session.

4. 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
```

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

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

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

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

For example,

```
Group03
cp -r 0 constant system /home/test03/pressurevessel03_div1
Group39
cp -r 0 constant system /home/test09/pressurevessel39_div2
```

Step-4

Creation of Geometry and Meshing

1. Open our case directory.

```
cd
cd pressurevessel<GN>_div<No>
```

2. To view the contents of the directory, type

```
ls
```

3. You can see the three folders namely **0**, **constant** and **system**.

   The **0** folder consists of files such as pressure, velocity etc where the boundary conditions are to be specified.

   The **constant** folder consists of file where the properties of fluid, turbulence models will be specified.

   The **system** folder consists of files

4. We need to rename the 0 folder to some name. Let us rename to **org**

```
mv 0 org
```

5. Now we need to open system folder.

```
cd system
```

6. To view the contents of the directory, type

```
ls
```

7. Open the **blockMeshDict** file using the text editor gedit

```
gedit blockMeshDict
```

8. Delete the contents of the blockMeshDict file to edit for our case as shown below.

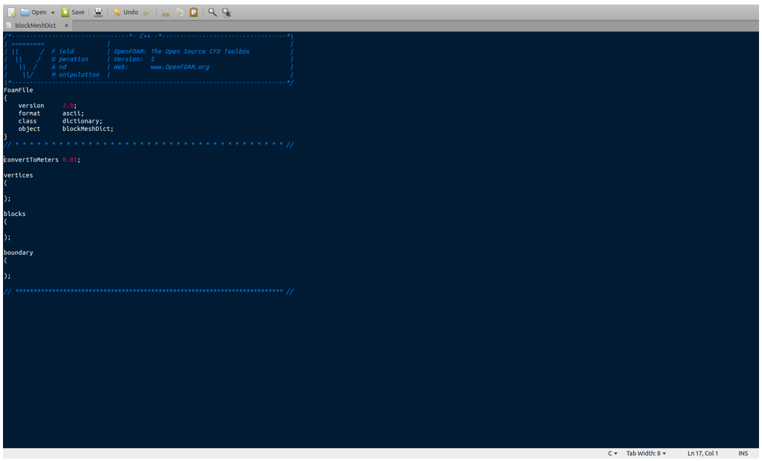

Figure 3: Vertices

9. The sections like **negY**, **posY**, **posYR** and **edges** can be removed.

Step-5

1. In OpenFOAM we create geometry using point in space. This is similar to the way we use a graph and plot points in it.
2. Check for **convertToMeters** in the very first line of **blockMeshDict** file.
3. By default the units used in OpenFOAM are in meters. Since our geometry is in centimetres we need to use the conversion factor from meters to centimetres.
4. Replace **0.001** to **0.01**.

Defining Vertices of the Geometry

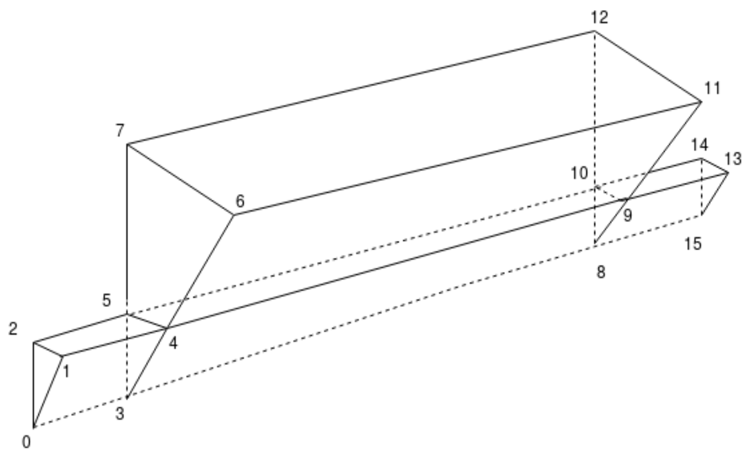

Figure 4: Vertices

1. In OpenFOAM the point numbering starts from **0** and continues till the last point number. In this case the last point number is **15**. So total there are 16 points.

2. Enter the coordinates of vertices as (X Y Z).

3. In OpenFOAM to make the geometry in to axis symmetric wedge the angle should be 5 deg.

4. In the vertices the values are directly given in centimetres as we mentioned the conversion factor in the first line.

5. **X** represents the radial distance, **Y** represents the angular axis and **Z** represents the axial length of the geometry.

6. Therefore the coordinate axis can be obtained by simple trigonometric formula of $tan\theta = \frac{OppositeSide}{AdjacentSide}$. Remember the angle should be **5 degrees**. So the half wedge angle is **2.5 degrees**, which is used for the calculation.

7. Inside the vertices section start entering the coordinates for these points.

```
vertices
(
    (0      0              -9)    // 0
    (2.5   -0.109152357    -9)    // 1
    (2.5    0.109152357    -9)    // 2
    (0      0               0)    // 3
    (2.5   -0.109152357     0)    // 4
    (2.5    0.109152357     0)    // 5
    (25    -1.091523573     0)    // 6
    (25     1.091523573     0)    // 7
    (0      0             100)    // 8
    (2.5   -0.109152357   100)    // 9
    (2.5    0.109152357   100)    // 10
```

```
    (25     −1.091523573  100)     //  11
    (25      1.091523573  100)     //  12
    (2.5    −0.109152357  109)     //  13
    (2.5     0.109152357  109)     //  14
    (0       0            109)     //  15

);
```

Defining Blocks of the Geometry

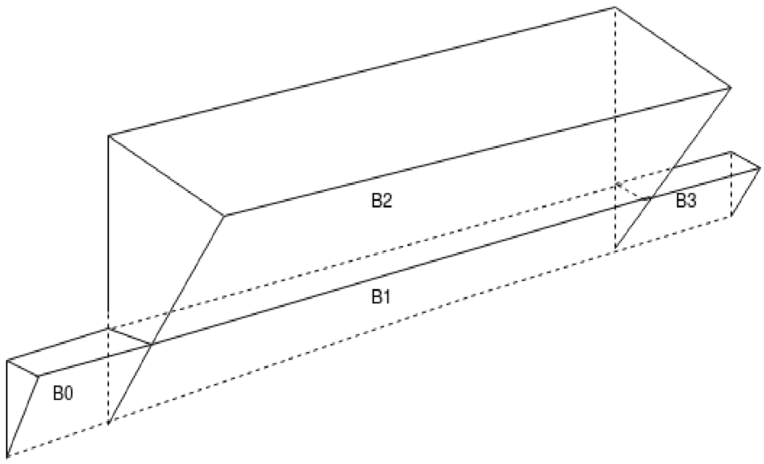

Figure 5: Blocks

1. Our geometry is divided in to blocks for Meshing purposes.

2. There are totally **4** blocks.

3. For block number **B0** we need to enter as

   ```
   blocks
   (
       hex (0 1 2 0 3 4 5 3)      (15 1 80)    simpleGrading (1 1 1)
   );
   ```

4. Here **hex** stands for **Hexahedral Mesh**.

5. The next **(0 1 2 0 3 4 5 3)** are the vertices. Here the anti-clockwise numbering should be followed when the geometry is viewed from the front.

6. The next **(15 1 80)** is the number of mesh elements in **(X Y Z)** directions. Since we are not solving the problem in angular axis we need to define unit cell.

7. The next **simpleGrading (1 1 1)** is used to define the progression of mesh.

8. For all the blocks **B0**,**B1**,**B2** and **B3** we need to type as

```
blocks
(
    hex (0 1 2 0 3 4 5 3)        (15 1 80)   simpleGrading (1 1 1)
    hex (3 4 5 3 8 9 10 8)       (15 1 180)  simpleGrading (1 1 1)
    hex (4 6 7 5 9 11 12 10)     (150 1 180) simpleGrading (1 1 1)
    hex (8 9 10 8 15 13 14 15)   (15 1 80)   simpleGrading (1 1 1)
);
```

Defining Patches for the Geometry

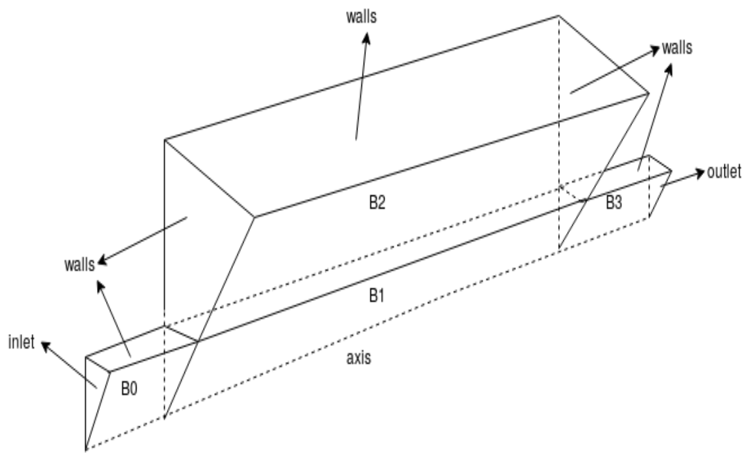

Figure 6: Patch names

1. In our geometry we have the patches as **inlet**,**outlet**, **walls**, **axis** and **frontAndBack**.
2. We define the patch by using the vertex numbers.
3. For the **inlet** patch we define as

```
boundary
(
    inlet
    {
        type patch;
        faces
        (
            (1 0 0 2)
        );
    }
);
```

4. The type should be kept as **empty** for the patch **axis**, **wall** for the patches **walls** and **wedge** for the patches **frontAndBack** (since we are not solving the problem in angular axis).

5. For all the patches we define as

```
boundary
(
    inlet
    {
        type patch;
        faces
        (
            (1 0 0 2)
        );
    }

    walls
    {
        type wall;
        faces
        (
            (6 7 12 11)
            (1 2 5 4)
            (6 4 5 7)
            (11 9 10 12)
            (9 10 14 13)
        );
    }

    outlet
    {
        type patch;
        faces
        (
            (13 15 15 14)
        );
    }

    axis
    {
        type empty;
        faces
        (
            (0 3 3 0)
            (3 8 8 3)
            (8 15 15 8)
        );
    }

    frontAndBack
    {
        type wedge;
```

```
        faces
        (
            (2 0 3 5)
            (5 3 8 10)
            (7 5 10 12)
            (10 8 15 14)

            (0 1 4 3)
            (3 4 9 8)
            (4 6 11 9)
            (8 9 13 15)
        );
    }

);
```

6. Now save the file and close it.

Step-6

Meshing the Geometry

1. Now to return to the case directory type the following

   ```
   cd ..
   (Note: Please note there is space between cd and dots)
   ```

2. To mesh the geometry type

   ```
   blockMesh
   ```

3. Your terminal window will display your mesh parameters.

4. Now in the terminal type

   ```
   checkMesh
   ```

5. It will display whether the **mesh is Ok** or it has any errors.

6. With this we completed the meshing of our geometry **2-D Axis symmetric Pressure Vessel**

Step-7

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. We know the Reynolds number from the problem specification. It is based of the vessel diameter (50 cm).

3. Now calculate the velocity from the Reynolds number formula. Note that this velocity can be given as inlet velocity.

$$Re = \frac{V * D}{\nu} \tag{1}$$

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

4. Now calculate the mass flow rate

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

5. 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.

6. Use this calculated inlet velocity in the above step for further calculations.

7. Now calculate the Reynolds number using the inlet velocity calculated above.

$$Re_i = \frac{V_i * D}{\nu} \tag{4}$$

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

$$I = 0.16 * (Re_i)^{(-1/8)} \tag{5}$$

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

$$k = \frac{3}{2}(V_i * I)^2 \tag{6}$$

   where I is the turbulence intensity

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

$$\epsilon = (C_\mu)^{\frac{3}{4}} * \frac{k^{\frac{3}{2}}}{l} \tag{7}$$

   where,
   l is $0.07 d_h$ is the turbulent length scale, where d_h is the hydraulic diameter, which for circular pipes is equal to the diameter d. In this case take the inlet diameter which is 5 cm
   $C_\mu = 0.09$

Step-8

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 pressurevessel<GN>_div<No>/0
ls
```

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

6. Please delete the files **f**, **omega**, **nuTilda** 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 nuTilda
```

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 pressurevessel<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.
    }
     frontAndBack
    {
        type                wedge;

    }
     axis
    {
        type                empty;

    }
```

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.
}

frontAndBack
{
    type            wedge;
}
axis
{
    type            empty;
}
```

5. Save it and close the file

Setting up of nut (Turbulent Viscosity)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;
}
frontAndBack
{
    type              wedge;
}

axis
{
    type              empty;
}
```

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 inlet Velocity>);
    //since flow direction is in positive Z axis.
    //Omit the angular brackets
}

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

frontAndBack
{
    type                wedge;
}

axis
{
    type                empty;
}
```

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;
    }

    frontAndBack
    {
        type                wedge;
    }

    axis
    {
        type                empty;
    }
```

5. Save it and close the file.

Step-9

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 pressurevessel<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.

Step-10

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 pressurevessel<GN>_div<No>/system
gedit controlDict
```

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

Step-11

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 pressurevessel<GN>_div<No>/system
gedit fvSolution
```

3. Change the residualControl as shown below. Note we are changing only the values. The remaining part of the file should remain as it is.

```
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.

Step-12

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

```
cd
cd pressurevessel<GN>_div<No>/system
gedit fvSchemes
```

3. In the bottom the file add the following lines. Note we are adding the lines shown below. The remaining part of the file should remain as it is.

```
fluxRequired
{
    default         no;
    p               ;
}
```

4. Save this and close the file.

Step-13

1. Now we need **g** file to include the effect of gravity.
2. The **g** file should be place inside **constant** folder.
3. Type the following command

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

4. Now type **ls** command in the terminal to display the contents inside the folder.
5. To copy the **g** file to our case folder type the following command.

```
For Group Number(GN) 01-30
cp g /home/test<GN>/pressurevessel<GN>_div1/constant

For Group Number(GN) 31-60
cp g /home/test<GN-30>/pressurevessel<GN>_div2/constant
```

For example,

```
Group03
cp g /home/test03/pressurevessel03_div1/constant
Group39
cp g /home/test09/pressurevessel39_div2/constant
```

6. Open the g file and edit the value as shown below

```
cd
gedit pressurevessel<GN>_div<No>/constant/g
```

7. It will open the **g** file.
8. Change the value from **(0 -9.81 0)** to **(0 0 -9.81)**. Since the vessel is aligned along positive z-axis and inlet is at the bottom.

Step-14

1. To do RTD simulation we need another simulation using **scalarTransportFoam** solver.
2. To do that we will set up the case directory for that in this session.
3. Type the following command to create the directory

```
cd
cd pressurevessel<GN>_div<No>
mkdir RTD
```

4. Type the following command to copy the files required to our directory RTD

```
cd
cd $FOAM_TUTORIALS
cd basic/scalarTransportFoam/pitzDaily
```

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

6. To copy the **0**,**constant** and **system** folders to our case folder **RTD** type the following command.

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

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

For example,

```
Group03
cp -r 0 constant system /home/test03/pressurevessel03_div1/RTD
Group39
cp -r 0 constant system /home/test09/pressurevessel39_div2/RTD
```

Step-15

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 pressurevessel<GN>_div<No>/RTD/system
gedit controlDict
```

3. Change the **endTime** as **120**.

4. Change the **deltaT** as **0.1**.

5. Change the **writeInterval** as **10**.

6. Save this and close the file.

- This brings us to the end of this session of Meshing and setting up of Case files required for simulation.

- In the next session we will modify the solver **simpleFoam** to add the effect of gravity and solve the flow problem and simulation required to obtain Residence Time Distribution(RTD) plots.