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-2 : Solving and Obtaining Residence Time Distribution(RTD) plots

Objective

Modifying the simpleFOAM solver to add the effect of gravity in to it

Running the simulation for flow through pressure vessel problem using the modified solver with the case directory created in the last session.

Running the simulation using scalarTransportFoam to obtain the Residence Time Distribution plot at outlet by introducing the scalar concentration at the inlet of the vessel

Geometry and Conditions

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

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

Axis along Z-axis

Length of inlet pipe = **9 cm**.

Length of outlet pipe = **9 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.

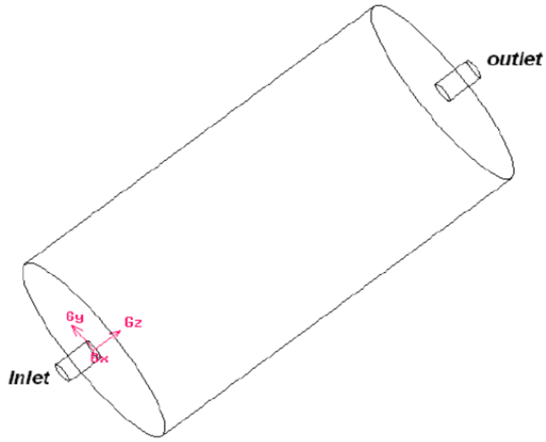


Figure 1: 3-D view of the pressure vessel geometry

Note:

1. **The below section is a copy of the source code of the solver which is C++**

2. **Make sure of providing spaces, semicolon, comma and upper,lower cases of alphabets as shown in the manual to avoid compilation errors.**

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

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
```

Modifying the simpleFOAM solver to add the effect of Gravity

1. Open the solver in the source directory. Type the following command

   ```
   cd ../..
   cd opt/openfoam5/applications/solvers/incompressible
   ls
   ```

2. Now copy the **simpleFoam** folder and paste it in the directory we created for our problem.

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

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

3. Now open our case directory

   ```
   cd
   cd pressurevessel<GN>_div<N>
   ```

4. Rename the folder as **simpleFoamGravity**

   ```
   mv simpleFoam simpleFoamGravity
   ```

5. Now open the folder **simpleFoamGravity**

   ```
   cd simpleFoamGravity
   ls
   ```

6. Delete the folders named **porousSimpleFoam** and **SRFSimpleFoam**.

   ```
   rm -rf porousSimpleFoam SRFSimpleFoam
   ```

Editing createField.H file

1. Now open the the file **createFields.H**

   ```
   gedit createFields.H
   ```

2. Below the line

```
#include "createPhi.H"
```

comment the next four lines by typing // infront of each line.

3. After the last line

   ```
   #include "createMRF.H"
   ```

   Add the following sections one below other.

4. We are adding the effect of gravity

   ```
   #include "readGravitationalAcceleration.H"
   ```

5. Second

   ```
   Info<< "Calculating field g.h\n" << endl;
   ```

6. Followed by

   ```
   volScalarField gh
       (
           IOobject
           (
               "gh",
               runTime.timeName(),
               mesh,
               IOobject::NO_READ,
               IOobject::AUTO_WRITE
           ),
           g & mesh.C()
       );
   ```

7. Adding total pressure field which is the sum of static pressure and acceleration due to gravity.

   ```
   volScalarField p_tot
       (
           IOobject
           (
               "p_tot",
               runTime.timeName(),
               mesh,
               IOobject::NO_READ,
               IOobject::AUTO_WRITE
           ),
           p + gh
       );
   ```

8. For reference cell in the mesh

```
label pRefCell = 0;
    scalar pRefValue = 0.0;
    setRefCell
    (
        p_tot,
        p,
        mesh.solutionDict().subDict("SIMPLE"),
        pRefCell,
        pRefValue
    );
```

9. Followed by

```
if (p.needReference())
    {
        p_tot += dimensionedScalar
        (
            "p_tot",
            p_tot.dimensions(),
            pRefValue - getRefCellValue(p_tot, pRefCell)
        );

        p = p_tot - gh;
    }
```

10. Save the file and close it.

Editing pEqn.H

1. Now open the the file **pEqn.H**

```
gedit pEqn.H
```

2. Below the line

```
fvOptions.correct(U)
```

Add the following lines

```
    p_tot == p + gh;
    if (p.needReference())
    {
        p_tot += dimensionedScalar
        (
           "p_tot",
            p_tot.dimensions(),
            pRefValue - getRefCellValue(p_tot, pRefCell)
        );
        p = p_tot - gh;
        // note that grad(p) is unchanged, since we have
        // "p = (p + gh) - gh = p" in the overall above procedure.
    }
```

3. Ignore the lines start with //

4. Save the file and close it.

Editing Make folder

1. There will be file named **simpleFoam.C**. Rename the file **simpleFoam.C** to **simpleFoamGravity.C**

   ```
   mv simpleFoam.C simpleFoamGravity.C
   ```

2. Open the Make folder

   ```
   cd Make
   ls
   ```

3. There will be two files namely **files** and **options**.

4. Open the file name **files**

   ```
   gedit files
   ```

5. Edit the file name **files** as shown below.

   ```
   simpleFoamGravity.C
   EXE = $(FOAM_USER_APPBIN)/simpleFoamGravity
   ```

6. Save the file and close it.

Compiling the new solver:

1. Go to one folder back by typing

   ```
   cd ..
   ```

2. Now will compile the solver. Type the following

   ```
   wclean
   wmake
   ```

Editing the controlDict file:

1. Open the **controlDict** file

   ```
   cd
   cd pressurevessel<GN>_div<n>/system
   gedit controlDict
   ```

2. Change the **endTime** from **1000** to **500**.

3. Save the file and close it.

Solving flow through Pressure Vessel

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

2. Now we need to run the case.

3. Type the following

```
cd
cd pressurevessel<GN>_div<n>
simpleFoamGravity
```

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

Step-11

1. Open the case folder.

```
cd
cd pressurevessel<GN>_div<n>
```

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.

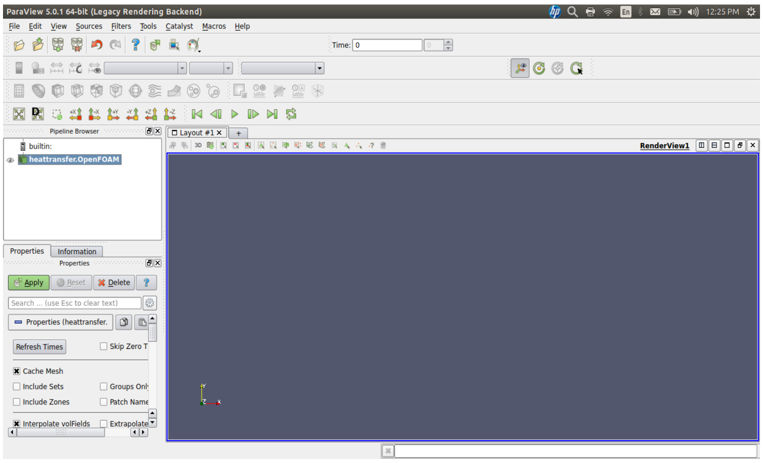


Figure 2: Paraview window

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

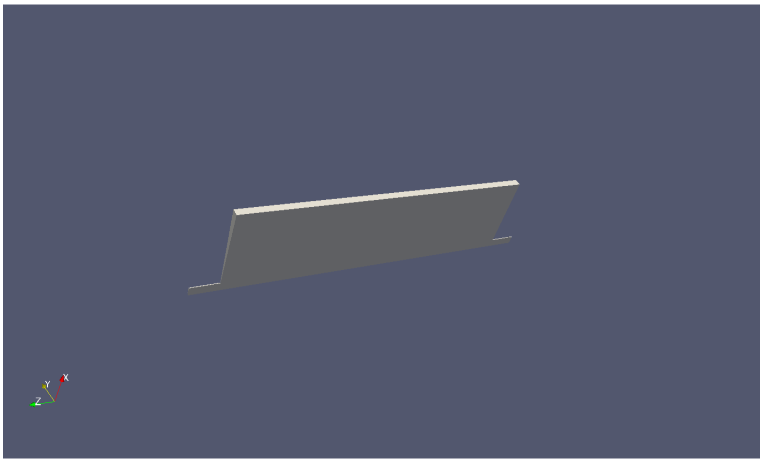


Figure 3: Solid model of the Geometry

5. In the pipeline browser, by default **p** and **U** will be selected. Check also the field p_{tot}
6. Click **Apply**.
7. In the drop down list select **U**. Go to last time step using VCR control.
8. Now click on the icon adjacent to the drop down menu which indicates (Rescale to Data range) when you move the cursor over it.
9. Keep the contour in proper orientation and save the screen shot.

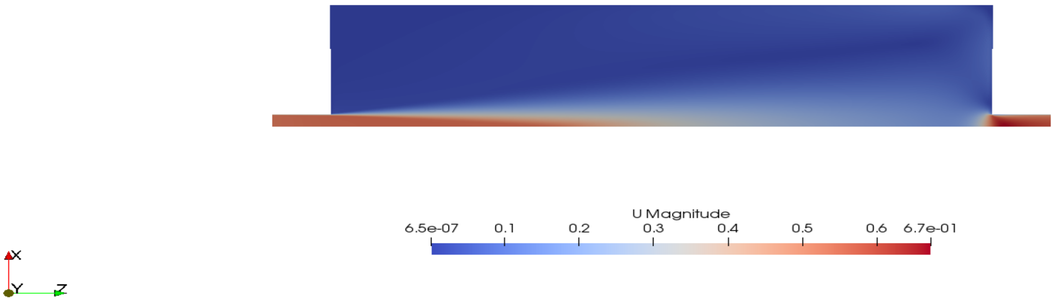


Figure 4: Velocity Contour

10. Do the same procedure for saving the screen shot for total pressure.

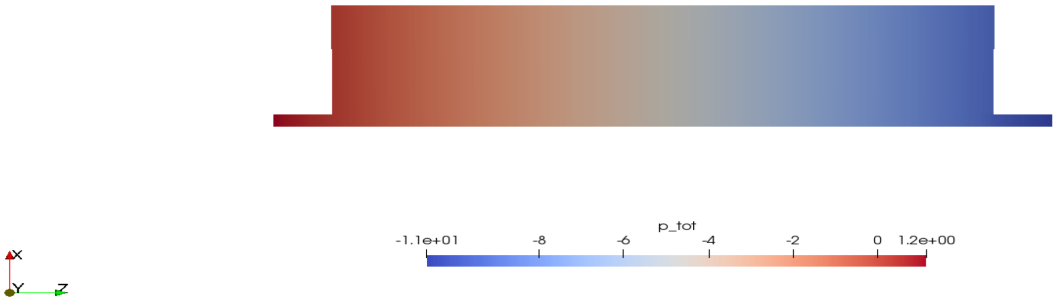


Figure 5: Total pressure Contour

Solving for RTD plots

Copying the Velocity file:

1. The flow field is solved in the previous simulation.
2. With the calculated flow field (i.e Velocity) we will put a scalar particle at the inlet and monitor the concentrating of the introduced particle at the outlet with respect to time.
3. To do so, we need to copy the **U** file of the last iteration of the previous simulation. (i.e **500**) to the **0** folder of the **RTD** folder.
4. To do so

   ```
   cd
   cd pressurevessel<GN>_div<n>/500
   cp U /home/test<GN>/pressurevessel<GN>_div<n>/RTD/0
   ```

Editing the Scalar(T) file:

1. The **T** file in the **0** folder is the scalar.

   ```
   cd
   cd pressurevessel<GN>_div<n>/RTD/0
   gedit T
   ```

2. Edit the **T** file to introduce the scalar at the inlet as shown below

   ```
   inlet
       {
           type            fixedValue;
   ```

```
            value               uniform 1.0;
        }
    outlet
        {
            type                zeroGradient;
        }
    walls
        {
            type                zeroGradient;
        }
    axis
        {
            type                empty;
        }
    frontAndBack_pos
        {
            type                wedge;
        }
    frontAndBack_neg
        {
            type                wedge;
        }
```

Copying the Mesh file

1. Now we need to copy the **polyMesh** directory which contains the mesh to the RTD folder. To do that

```
cd
cd pressurevessel<GN>_div<n>
cp -r constant/polyMesh /home/test<GN>/pressurevessel<GN>_div<n>/RTD/constant
```

Editing the Transport Properties file

1. Now open the transportProperties file

```
cd
cd pressurevessel<GN>_div<n>/RTD/constant
gedit transportProperties
```

2. Change the diffusion co-efficient **DT** as **0.0** to make the transport from inlet to the outlet solely due to convection.

```
DT                  DT [0 2 -1 0 0 0 0] 0.0;
```

Editing controlDict file

1. Now open the controlDict file

```
cd
cd pressurevessel<GN>_div<n>/RTD/system
gedit controlDict
```

2. Change the **endTime** as **10**.

3. Change the **deltaT** as **0.01**.

4. Change the **writeInterval** as 10.

5. Below the line **runTimeModifiable true;** add the following which will calculate the integrate value of the **concentration T** at the outlet patch

```
functions
{

RTD
{
    functionObjectLibs ("libfieldFunctionObjects.so");
    type               surfaceFieldValue;
    enabled            true;
    writeControl       outputTime;
    writeInterval      1;
    writeFields        false;
    name               outlet;

    regionType         patch;

    operation          areaIntegrate;

    fields
    (
        T
    );
}
}
```

6. Now setting up the case directory for the **RTD** has been completed.

7. Now we need to run the case.

8. Type the following

```
cd
cd pressurevessel<GN>_div<n>/RTD
scalarTransportFoam
```

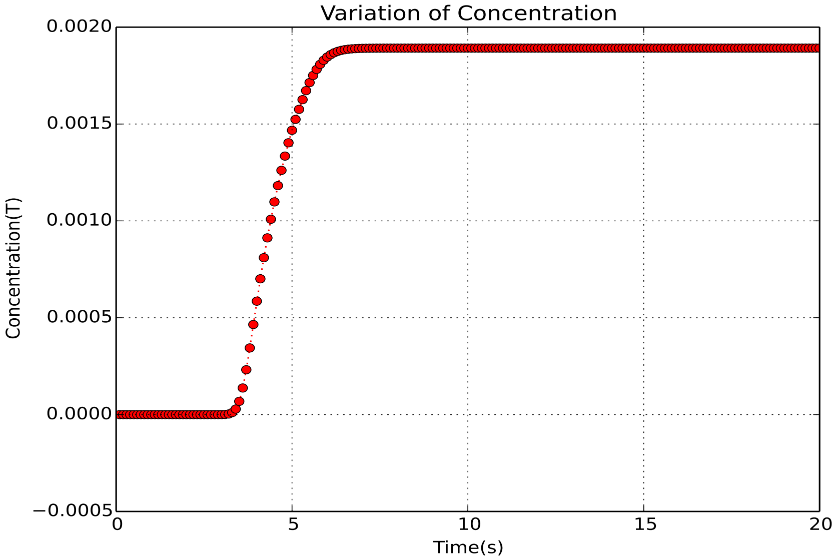
9. The iterations will be running. Wait till the specified **endTime** is reached.

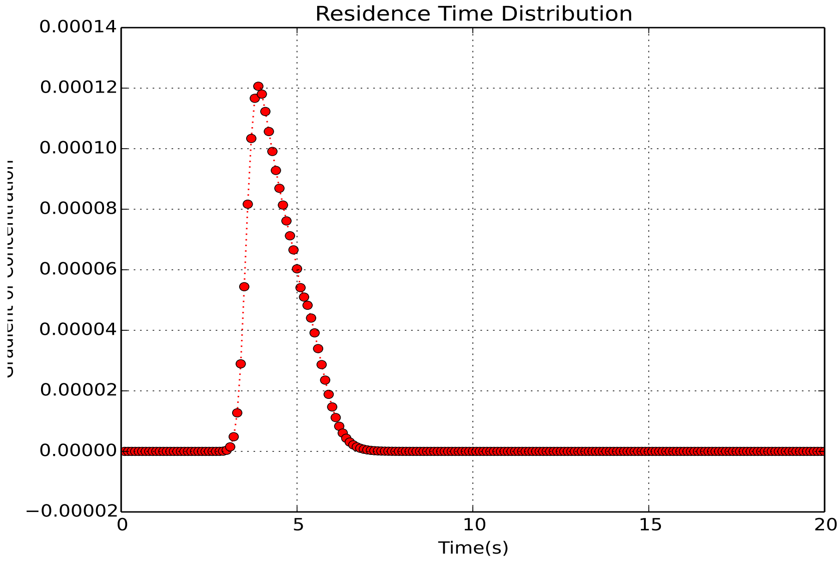
Obtaining RTD plots

1. There will be folder named postProcessing will be created.
2. Open that folder

   ```
   cd
   cd pressurevessel<GN>_div<n>/RTD/postProcessing/RTD/0
   ```

3. You can find a file named **surfaceFieldValue.dat**
4. Use the python code shown below to obtain the variation of concentration at the outlet patch due to the scalar concentration introduced at the inlet with respect to time.
5. The gradient of the change of concentration with respect to time will give the Residence Time Distribution plot.





6. With this the problem is completed. In the terminal type the following

```
cd
cd pressurevessel<GN>_div<n>
firefox
```

7. Open your gpo mail account.

8. Attach the contours and **surfaceFieldValue.dat** file and mail for your report preparation.

Python code for obtaining RTD plots

```
import numpy as np
from matplotlib import pyplot as plt

data=np.loadtxt('surfaceFieldValue.dat')
X=data[:,0]
Y=data[:,1]
Y= (360/5)*Y  //To convert for 360 degrees.
b=np.gradient(Y)

plt.figure(1)
plt.plot(X,Y,':ro')
legend = plt.legend(loc='upper␣right', shadow=True)
plt.ylabel('Concentration(T)')
plt.xlabel('Time(s)')
plt.grid('on')
plt.title('Variation␣of␣Concentration')
plt.savefig('Concentration.eps')
plt.show()

plt.figure(2)
plt.plot(X,b,':ro')
legend = plt.legend(loc='upper␣right', shadow=True)
plt.ylabel('Gradient␣of␣Concentration(T)')
plt.xlabel('Time(s)')
plt.grid('on')
plt.title('RTD␣Plot')
plt.savefig('RTD.eps')
plt.show()
```