

CFD OpenFOAM LAB

Assignments

CL455

Chemical Engineering Department

FOSSEE
IIT Bombay

Funded by Ministry of Education
(MHRD)

November 4, 2020





Turbulent Flow and Residence Time Distribution(RTD) plots in a Pressure Vessel using OpenFOAM

Assignment-1:

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

Assignment-2:

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

Problem statement

Geometry of pressure vessel is shown in fig. 1. It has 1 inlet and 1 outlet at both the end of cylindrical vessel. A wedge type axisymmetrical geometry with given data will be provided in a blockMesh file. Wedge type geometry of any cylindrical domain will save computation time and space in system. However, it won't affect results and give identical results of 3D simulations. In OpenFOAM, to create wedge type geometry, the wedge angle should be less than 5 degree.

For more details visit this link: <https://www.openfoam.com/documentation/user-guide/boundaries.php>

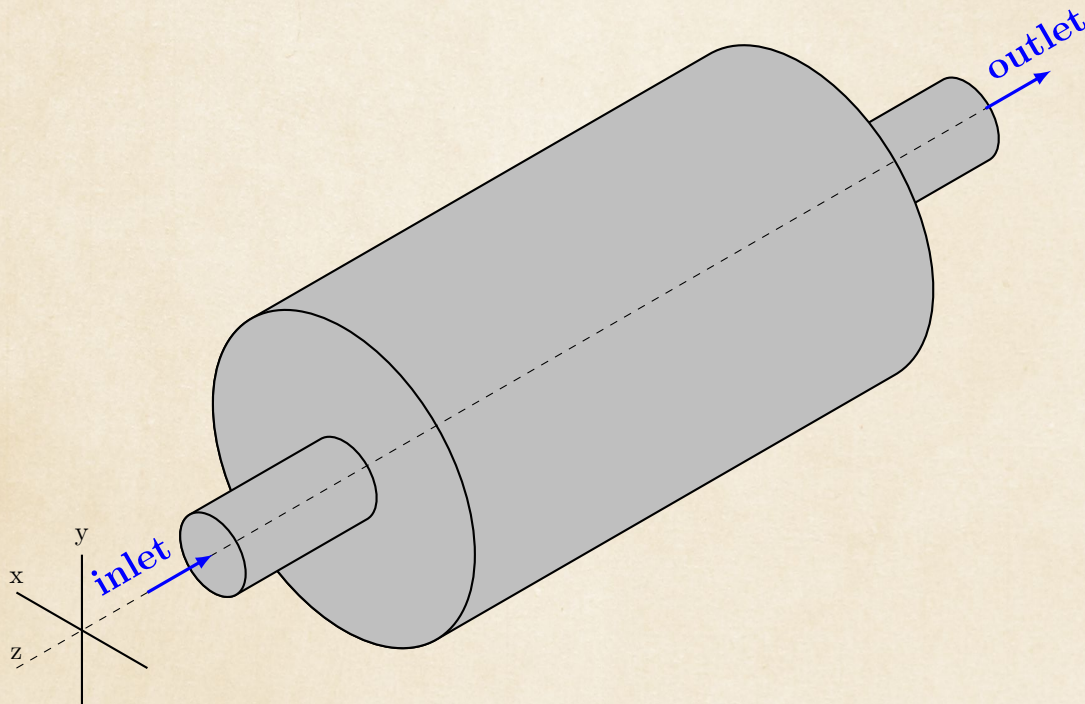


figure 1: 3D view of pressure vessel geometry

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

kinematic viscosity $\nu = 1\text{e-}6 \text{ m}^2/\text{s}$

Reynold's number based on the vessel diameter (50 cm) = $2500 + 100 \cdot (n/2)$

where, n is last 3 digits of your roll no.

For example, if your roll no is 170020015, $\text{Re} = 2500 + 100 \cdot (015/2) = 3250$

if your roll no is 170020115, $\text{Re} = 2500 + 100 \cdot (115/2) = 8950$



Assignment-1

Copying the folders

1. Download and extract the ZIP file from the moodle. (For backup, make a copy of it)
2. Copy the folder named CL455 from the extracted location into your run directory. To do so, type the following command once the run directory is your working directory

```
cp -r ~/Downloads/CL455 .
```

NOTE: The above command is for the ZIP file downloaded and extracted in the Downloads directory. Make necessary changes in the command depending upon your download location.

Compiling the modified simpleFoam solver

To simulate the flow in a pressure vessel, the **simpleFoam** solver needs to be modified to handle gravity. All the necessary changes have been made in the folder named **simpleGravityFoam** in the CL455 folder. The same needs to be compiled before it can be used to simulate problems. To do so

1. Move into the **simpleGravityFoam** directory
2. Execute the following commands

```
wclean  
wmake
```

Setting up Case Directory for turbulent flow in a pressure vessel

Follow the steps below to set up the case directory to solve turbulent flow in a pressure vessel

1. The case directory for assignment-1 is named pressureVessel in the CL455 folder. Go into this directory to make necessary changes.
2. The case files have been set up for a Reynolds number of 5000 and an inlet velocity of 1 m/s. Make necessary changes in the files in the 0 directory based on the Reynolds number at the pipe inlet.

NOTE: The Reynolds number (5000) and the inlet velocity (1 m/s) used to set up the files are trivial values. All turbulence parameters have to be calculated based on the Reynolds number for the pipe inlet and the inlet pipe diameter. Use the principle of mass conservation to calculate the velocity/Reynolds number at the pipe inlet based on the given Reynolds number for the vessel. Refer to the spoken tutorial on Simulating turbulent flow through a channel for more details.

3. In the constant folder, there's a gravitational acceleration file named g. The direction of the acceleration is given in the negative y-direction. Make necessary changes in the file such that the acceleration is $9.81 \text{ m}^2/\text{s}$ in the **negative z-direction**.
4. Once the case files have been set up, mesh the geometry using the **blockMesh** command and then simulate the problem using **simpleGravityFoam** command.

What to submit by today?

- Open case file in paraview, clip mesh along Y-Axis.
- Press ctrl+space in your keyboard. One window will open in foreground.
- Type *Rotational Extrusion* and select that filter.
- Click on Apply button from properties panel and you will see while cylindrical geometry rendered.
- For the last time-step save screenshot of velocity and pressure profile and upload on moodle.



Assignment-2

python installation

NOTE: Before solving the assignment, python3 and some of its packages need to be installed. Use the following commands to install/update python3, numpy and matplotlib.

```
sudo apt-get install python3
sudo apt-get install python3-numpy
sudo apt-get install python3-matplotlib
```

Setting up Case Directory for Residual Time Distribution(RTD)

The case directory for assignment-2 is named RTD in the CL455 folder. Go into this directory to make the necessary changes.

1. The RTD plots are based on the flow field solved in the previous assignment for the pressure vessel.
2. Therefore, the first step would be to include the final velocity field (at the 500 th iteration) in the RTD case directory. Use the following command, from the RTD case directory, to copy the final U file from the pressureVessel folder to the 0 folder of RTD,

```
cp $FOAM_RUN/CL455/pressureVessel/500/U 0
```

3. Once the case files have been set up, mesh the geometry using the **blockMesh** command and then simulate the problem using **scalarTransportFoam** command.
4. Once the solver iterations are completed, the data for residual time distribution is created as .dat file in the location CL455/RTD/postProcessing/RTD/0
5. To plot the residual time distribution, the python script needs to be copied to the location of .dat file. Run the following command from the RTD case directory to do so.

```
cp plotRTD.py $FOAM_RUN/CL455/RTD/postProcessing/RTD/0
```

6. Now navigate to the location of the residual time distribution data using the below command

```
cd postProcessing/RTD/0
```

7. Then, execute the python script using the command

```
python3 plotRTD.py
```

8. Save the plots.
9. Your submission ZIP file should include the two plots.



Appendix

Modifying the simpleFoam solver to add the effect of Gravity

1. The solver source code is available in the directory \$FOAM_SOLVERS. To modify the simpleFoam solver, first the solver source code needs to be copied into the run directory.

```
cp -r $FOAM_SOLVERS/incompressible/simpleFoam simpleGravityFoam
```

All the files/folder hereafter are available in the simpleGravityFoam folder.

2. Make changes to the createField.H file. The changes to the code are in line 31-34 and 45-95.

```
1  Info<< "Reading field p\n" << endl;
2  volScalarField p
3  (
4      IOobject
5      (
6          "p",
7          runTime.timeName(),
8          mesh,
9          IOobject::MUST_READ,
10         IOobject::AUTO_WRITE
11     ),
12     mesh
13 );
14
15 Info<< "Reading field U\n" << endl;
16 volVectorField U
17 (
18     IOobject
19     (
20         "U",
21         runTime.timeName(),
22         mesh,
23         IOobject::MUST_READ,
24         IOobject::AUTO_WRITE
25     ),
26     mesh
27 );
28
29 #include "createPhi.H"
30
31 //label pRefCell = 0;
32 //scalar pRefValue = 0.0;
33 //setRefCell(p, simple.dict(), pRefCell, pRefValue);
34 //mesh.setFluxRequired(p.name());
35
36 singlePhaseTransportModel laminarTransport(U, phi);
37
38 autoPtr<incompressible::turbulenceModel> turbulence
39 (
40     incompressible::turbulenceModel::New(U, phi, laminarTransport)
41 );
42
43 #include "createMRF.H"
44 #include "createFvOptions.H"
45 #include "readGravitationalAcceleration.H"
46
47 Info << "Calculating field g.h\n" << endl;
```



```
48 volScalarField gh
49 (
50     IOobject
51     (
52         "gh",
53         runTime.timeName(),
54         mesh,
55         IOobject::NO_READ,
56         IOobject::AUTO_WRITE
57     ),
58     g & mesh.C()
59 );
60
61 volScalarField p_tot
62 (
63     IOobject
64     (
65         "p_tot",
66         runTime.timeName(),
67         mesh,
68         IOobject::NO_READ,
69         IOobject::AUTO_WRITE
70     ),
71     p + gh
72 );
73
74 label pRefCell = 0;
75 scalar pRefValue = 0.0;
76 setRefCell
77 (
78     p_tot,
79     p,
80     mesh.solutionDict().subDict("SIMPLE"),
81     pRefCell,
82     pRefValue
83 );
84
85 if (p.needReference())
86 {
87     p_tot += dimensionedScalar
88     (
89         "p_tot",
90         p_tot.dimensions(),
91         pRefValue - getRefCellValue(p_tot, pRefCell)
92     );
93
94     p = p_tot - gh;
95 }
```

3. Make changes to the pEqn.H file. The changes to the code are in line 51-62.

```
1 {
2     volScalarField rAU(1.0/UEqn.A());
3     volVectorField HbyA(constrainHbyA(rAU*UEqn.H(), U, p));
4     surfaceScalarField phiHbyA("phiHbyA", fvc::flux(HbyA));
5     MRF.makeRelative(phiHbyA);
6     adjustPhi(phiHbyA, U, p);
7
8     tmp<volScalarField> rAtU(rAU);
9
10    if (simple.consistent())
```




```
11 {
12     rAtU = 1.0/(1.0/rAU - UEqn.H1());
13     phiHbyA +=
14         fvc::interpolate(rAtU() - rAU)*fvc::snGrad(p)*mesh.magSf();
15     HbyA -= (rAU - rAtU())*fvc::grad(p);
16 }
17
18 tUEqn.clear();
19
20 // Update the pressure BCs to ensure flux consistency
21 constrainPressure(p, U, phiHbyA, rAtU(), MRF);
22
23 // Non-orthogonal pressure corrector loop
24 while (simple.correctNonOrthogonal())
25 {
26     fvScalarMatrix pEqn
27     (
28         fvm::laplacian(rAtU(), p) == fvc::div(phiHbyA)
29     );
30
31     pEqn.setReference(pRefCell, pRefValue);
32
33     pEqn.solve();
34
35     if (simple.finalNonOrthogonalIter())
36     {
37         phi = phiHbyA - pEqn.flux();
38     }
39 }
40
41 #include "continuityErrs.H"
42
43 // Explicitly relax pressure for momentum corrector
44 p.relax();
45
46 // Momentum corrector
47 U = HbyA - rAtU()*fvc::grad(p);
48 U.correctBoundaryConditions();
49 fvOptions.correct(U);
50
51 p_tot == p + gh;
52 if (p.needReference ())
53 {
54     p_tot += dimensionedScalar
55     (
56         "p_tot",
57         p_tot.dimensions(),
58         pRefValue - getRefCellValue(p_tot, pRefCell)
59     );
60     p = p_tot - gh;
61 }
62 }
```

4. Rename the file `simpleFoam.C` to `simpleGravityFoam.C`.

```
mv simpleFoam.C simpleGravityFoam.C
```

5. In the `Make` folder, make changes to the file named `files`.

```
1 simpleGravityFoam.C
2 EXE = $(FOAM_USER_APPBIN)/simpleGravityFoam
```