Tutorial 1: Basics

Learning Outcomes:

Running `icoFoam`, `pisoFoam`, `simpleFoam`; postprocessing with `foamCalc`, `sample` and `xmgrace`, setting up a new case, conversion from Fluent.

I.1 Lid Driven Cavity

Directory `tutorial1` contains a case called `cavity` which is one of the tutorial cases distributed with OpenFOAM. It simulates the flow in a lid-driven cavity using the code `icoFoam`, which solves the time-dependent Navier-Stokes equations for laminar incompressible flow (the name is short for *incompressible Foam*). Go into the `tutorial1` directory and run `blockMesh` :

```
cd cavity
blockMesh
```

which generates the mesh; then run `icoFoam`. Also run `postProcess -func ‘‘mag(U)’’` which takes the output and generates the magnitude of the velocity in each timestep. You should have 6 timestep directories going `0...0.5`.

[Q.I.1] Using `paraFoam`, generate plots of pressure, velocity magnitude and velocity vectors on the mid-plane of the simulation.
[Q.I.2] Use `postProcess -func sampleDict` to find the pressure variation along the cavity lid, and compare this with the pressure variation across the mid-line (plot graphs using `xmgrace`).
[Q.I.3] What is the Reynolds number for this flow?
[Q.I.4] Copy the whole case to a new name `turbCavity`. Alter the flow conditions to give a Reynolds number of $10,000$ (this can be done either by changing the lid speed – edit the `U` field in the `0` timestep directory – or changing the fluid viscosity – edit the value of `nu` in `constant/transportProperties`). Turn on the turbulence model in `constant/RASProperties` and rerun the case using `pisoFoam`; you will need to change the timestep however. How have the results changed?

The next exercise is to use `fluentMeshToFoam` to convert a Fluent mesh into an OpenFOAM case. The mesh we will use is one used in the 3rd year course; see figure 1.

This geometry can be created using ANSYS Mesher and exported as a Fluent .msh file (make sure it is saved in ASCII format; in Mesher look for `Tools` → `Options` → `Export`, and switch the output format to ASCII). The syntax of the `fluentMeshToFoam` command takes an argument :

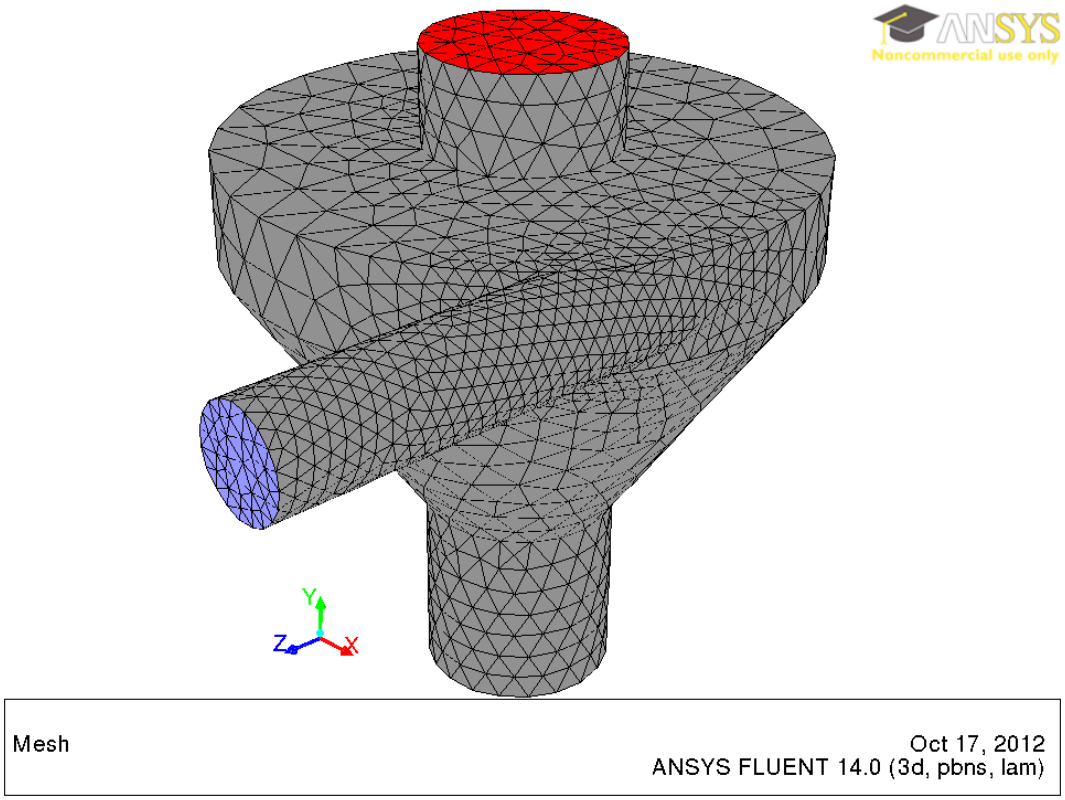

Figure 1: Geometry (and mesh) for vortex separator.

```
fluentMeshToFoam <fluentMeshFile>.msh
```

For this to work the basic structure of the case must already exist; a case directory containing `system/controlDict`. This is provided in the case `VFC` together with the appropriate .msh file. Run
`fluentMeshToFoam`, which will generate the `constant/polyMesh` directory.

Having done this it is necessary to create a `0` timestep directory. One has been provided with the case, but the boundary conditions need to be modified. The inlet velocity of 4 m/s must be specified, as do the inlet values of k and ϵ. Generally these values are not known and so must be estimated; conventionally the following 'rule of thumb' is used;

$$k = \frac{3}{2}\left(U_{ref} T_i\right)^2 \qquad \epsilon = C_\mu^{3/4} \frac{k^{3/2}}{l} \qquad \text{where} \quad l = 0.07L$$

where L is a characteristic inlet scale, U_{ref} the inlet flow velocity, T_i the turbulent intensity, and $C_\mu = 0.09$ one of the coefficients from the $k - \epsilon$ turbulence model.

Check that the boundaries are correctly labelled by looking in
`constant/polyMesh/boundaries`, and set up the boundaries for `p, U, k, epsilon`. Then run `simpleFoam` to perform the calculation.

Often we want to inspect the residuals from the calculation. Whenever any OpenFOAM solver solves and equation it prints out information about the matrix residuals on the terminal line, and it is often worth saving this information and displaying it graphically. The command

```
simpleFoam > log
```

will redirect output from running `simpleFoam` to the file `log` which can then be examined afterwards. Alternatively the commands

```
simpleFoam > log &
tail -f log
```

start `simpleFoam` as a background process (so it runs independently of anything you do afterwards in the command line); `tail` displays the end of the file `log` whilst `tail -f` updates this every time `log` gets updated. Having done this you can then run the command

```
foamLog log
```

which systematically extracts all the residual data from the log file and collates it into different files, one for each variable. Traditionally we examine the initial residual for each variable, plotted logarithmically against iteration number.

[Q.I.5] Using `paraFoam`, generate plots of pressure, velocity magnitude and velocity vectors.
[Q.I.6] Plot out the residuals for pressure and the velocity components using xmgrace (or other plotting software).
[Q.I.7] Use `sample` to find the pressure variation along the mid-line of the domain (plot graphs using `xmgrace`).