

0.1 question 1

The PISO algorithm (Pressure Implicit with Splitting of Operator) is an algorithm used to find pressure and velocity in steady and unsteady flow. It uses one predictor step and two corrector steps. It can be seen as an extension of the SIMPLE algorithm. Consider the discretised momentum equation and the discretised continuity equation. Using a guessed pressure p^* into the momentum equation, the velocity component (u^*, v^*, w^*) are derived. Since the pressure is guessed, the derived velocity field will not satisfy the continuity equation. To solve this problem, in a first corrector step, a corrector pressure p' is defined as the difference between the guess pressure and the real pressure. Using the continuity equation a pressure correction equation is derived in terms of the pressure p^{**} . Introducing this pressure into the discretised momentum equation a velocity field (u^{**}, v^{**}, w^{**}) satisfying the discretised continuity equation is computed. In the second corrector step a corrected pressure p^{***} is introduced in the discretised momentum equation to derive a twice corrected velocity field $(u^{***}, v^{***}, w^{***})$.

The PISO algorithm implementation in openFoam:

1)define equation for U

```
fvVectorMatrix UEqn
(
    fvm::ddt(U)
    + fvm::div(phi, U)
    - fvm::laplacian(nu, U)
);
```

2)solve the momentum predictor:

```
solve (UEqn == -fvc::grad(p));
```

3)Calculate a_p coefficient and calculate U

```
volScalarField rUA = 1.0/UEqn().A();
U = rUA*UEqn().H();
```

4)Calculate the flux

```
phi = (fvc::interpolate(U) & mesh.Sf())
      + fvc::ddtPhiCorr(rUA, U, phi);
adjustPhi(phi, U, p);
```

5)Solve the pressure equation

```
fvScalarMatrix pEqn
(
    fvm::laplacian(rUA, p) == fvc::div(phi)
);
pEqn.setReference(pRefCell, pRefValue);
pEqn.solve();
```

6)Correct the flux

```

if (nonOrth == nNonOrthCorr)
{
    phi -= pEqn.flux();
}

```

7) Correct the flux

```

if (nonOrth == nNonOrthCorr)
{
    phi -= pEqn.flux();
}

```

8) calculate continuity error

```

# include "continuityErrs.H"

```

9) Perform the momentum corrector step

```

U -= rUA*fvc::grad(p);
U.correctBoundaryConditions();

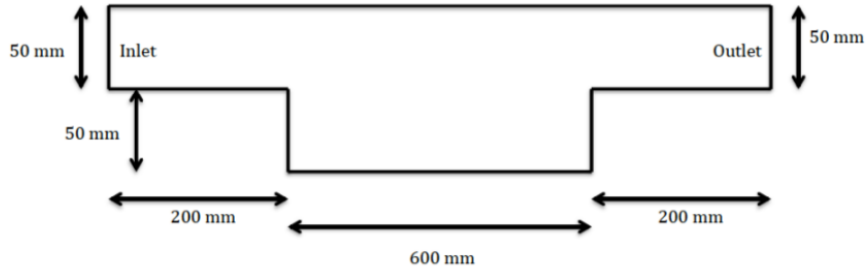
```

10) repeat from 3) for the prescribed number of PISO corrector step.

0.2 question 2

consider the domain given by

Figure 1: Computation domain



We want to solve this flow using the pisoFoam solver for LES model. The governing equations are given by the LES continuity equation and the LES momentum equations:

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho \bar{\mathbf{u}}) = 0 \quad (1)$$

$$\frac{\partial(\rho \bar{u})}{\partial t} + \text{div}(\rho \bar{u} \bar{\mathbf{u}}) = -\frac{\partial \bar{p}}{\partial x} + \mu \text{div}(\text{grad}(\bar{u})) - (\text{div}(\rho \bar{u} \bar{\mathbf{u}}) - \text{div}(\rho \bar{u} \bar{\mathbf{u}})) \quad (2)$$

$$\frac{\partial(\rho\bar{v})}{\partial t} + \text{div}(\rho\bar{v}\bar{\mathbf{u}}) = -\frac{\partial\bar{p}}{\partial y} + \mu\text{div}(\text{grad}(\bar{v})) - (\text{div}(\rho\bar{v}\bar{\mathbf{u}}) - \text{div}(\rho\bar{v}\bar{\mathbf{u}})) \quad (3)$$

where \bar{u}, \bar{v} are the mean velocities in the x and y direction. ρ is the density of the fluid. μ is the viscosity and \bar{p} is the average pressure.

Boundary condition at the inlet:

$$(\bar{u}, \bar{v}) = (10\text{m/s}, 0)$$

$$\bar{p} = 0$$

Boundary condition at the outlet

$$\bar{p} = 0$$

In figure 2 and 3 the average or mean velocity and pressure are given :

Figure 2: average pressure

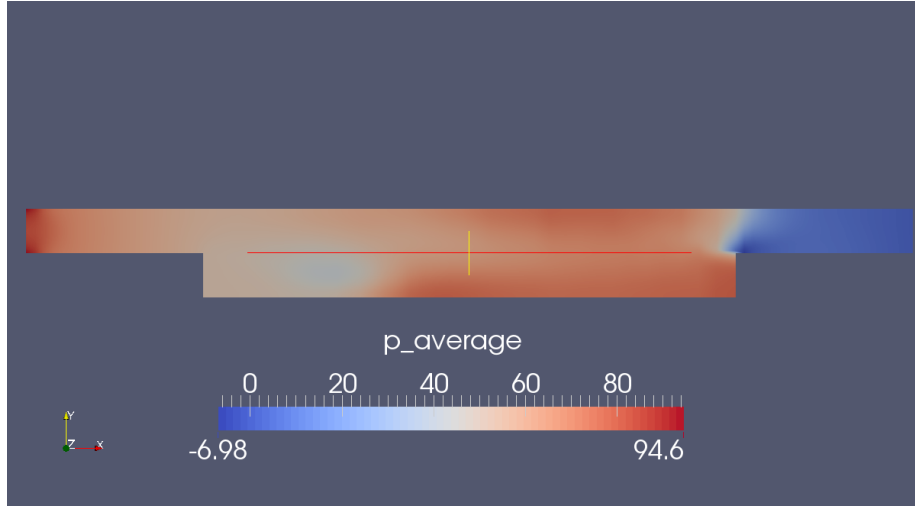
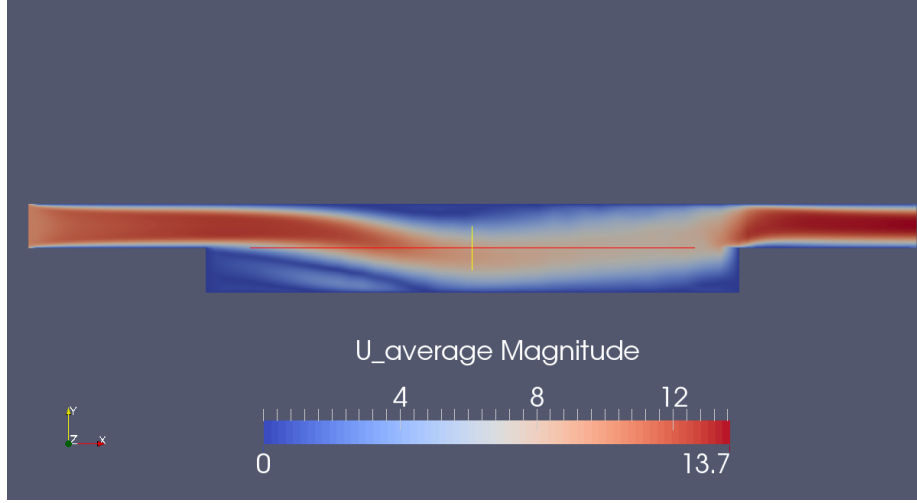


Figure 3: average velocity



The size of the mesh will drive the quality of the solution. The convection term is discretised by the Gauss scheme (second order Gaussian integration) with a second order linear interpolation scheme (central differencing scheme). The central differencing scheme is conservative, bounded and second order accurate for Peclet number less than 2. Even though it does not possess transportiveness it requires less computational time per time step

0.3 question 3

We use *pimplefoam* to solve the same flow. The k - ϵ model equations:

$$\frac{\partial(\rho k)}{\partial t} + \nabla \cdot (\rho k \vec{U}) = \nabla \cdot \left(\frac{\mu_t}{\sigma_k} \nabla k \right) + 2\mu_t S_{ij} \cdot S_{ij} - \rho \epsilon \quad (4)$$

$$\frac{\partial(\rho \epsilon)}{\partial t} + \nabla \cdot (\rho \epsilon \vec{U}) = \nabla \cdot \left(\frac{\mu_t}{\sigma_\epsilon} \nabla \epsilon \right) + C_{1\epsilon} \frac{\epsilon}{k} 2\mu_t S_{ij} \cdot S_{ij} - C_{2\epsilon} \rho \frac{\epsilon^2}{k} \quad (5)$$

with the eddy viscosity defined by

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \quad (6)$$

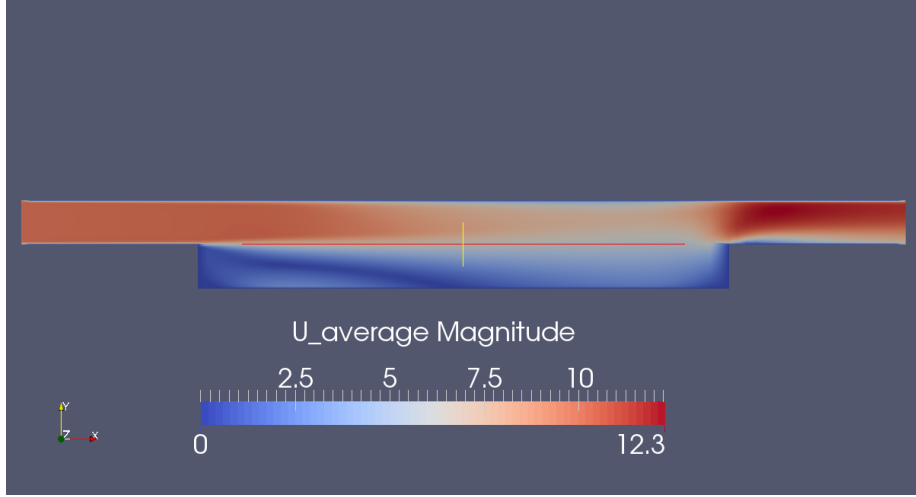
In equation (4, 5) k is the turbulent kinetic energy and ϵ is the rate of viscous dissipation. ρ is the density. $c_{1\epsilon} = 1.44$, $c_{2\epsilon} = 1.92$, $C_\mu = 0.09$ and S_{ij} is the rate of deformation. the boundary condition for the turbulent model are: at the inlet we have:

$$k = \frac{2}{3} (U_{ref} T_i)^2$$

$$\epsilon = C_\mu^{3/4} \frac{k^{3/2}}{0.07L}$$

where L is a characteristic length and T_i is the turbulent intensity. The initial kinetic energy is set to $k = 0.375m^2/s^2$ and the initial dissipation energy rate to $\epsilon = 10.781m^2/s^3$ at the inlet. In figure 4 the average velocity is given. The convective term is discretised by using an upwinding scheme satisfying transportiveness, boundness, conservativeness and is first order accurate.

Figure 4: average velocity



0.4 question 4

pimpleFoam uses large time step. It is more suitable for RANS because RANS can use coarser mesh than LES which uses more finer mesh. In RANS you use larger time step then in LES.

0.5 Comparison of result from RANS and LESS

Figure 4 shows the magnitude of the average velocity from PimpleFoam solver using RANS. The velocity field reach steady state faster than the velocity field obtain from PISOFoam using LES from figure 2 and 3. The LES model shows more turbulence than the RANS model because there is no dissipation to dump the kinetic energy. (see video). In LES the recirculation bubbles are far from the inlet some where in the middle of the domain. In RANS the recirculation bubbles are closer to the backward step and dies out quickly as the flow reaches steady state. In the LES model there is no dissipation to dump the kinetic energy while in the RANS model the rate of dissipation ϵ is much larger than the energy used to create turbulence. As the flow progress in the RANS model, the kinetic energy used to maintain turbulence will dissipate very fast and the flow will reach steady state faster.

0.6 question 6

LES is not usually used as a two D model because turbulence is a 3D phenomenon.