Computational Fluid Dynamics

SG2212 Project Report

David Ahnlund

Question 1

The kronecker operator (**kron**) is an operator for tensor product, which falls convinient when doing finite differences in higher dimensions than the 1D case. As we study the 2D case in this lid-driven cavity project the 5 point stencil is considered:

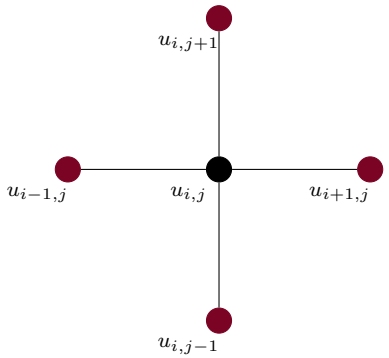


Figure 1: The five point stencil in 2D.

If we construct a finite difference scheme for the laplace operator we get:

$$\nabla^2 u = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2} \tag{1}$$

$$\left\{ \text{and if } h_x = h_y = h \implies \frac{u_{i+1,j} + u_{i-1,j} - 4u_{i,j} + u_{i,j+1} + u_{i,j-1}}{h^2} \right\} \tag{2}$$

The above expression we can transform into a vector form in size $(N_x, N_y) \mapsto N_x \cdot N_y$, using the indexing that one step in y-direction is equivalent to one row in the matrix *i.e.* $+N_x$ steps.

We can now define the second partial derivative in the separate directions:

$$S_x = \frac{1}{h_x^2}\begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & \ddots & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix} \qquad S_y = \frac{1}{h_y^2}\begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & \ddots & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{bmatrix} \tag{3}$$

Forming a matrix by the size $(N_x \cdot N_y) \times (N_x \cdot N_y)$ is now done by the kronecker product.

The operation done by the kronecker product essentially does the following in this case. Consider the identity matrix in the shape $N_y \times N_y$ (I_y), and the S_x from above. We get:

$$\text{kron}(I_y, S_x) = I_y \otimes S_x = \begin{bmatrix} [S_x] & 0 & \dots & \\ 0 & [S_x] & 0 & \dots \\ \vdots & & \ddots & \\ & & & [S_x] \end{bmatrix} \quad \text{shape is } (N_y \cdot N_x) \times (N_y \cdot N_x)$$

and for the other term in the sum of kronecker products we have:

$$S_y \otimes I_x = \begin{bmatrix} S_y^{1,1}[I_x] & 0 & \dots & \\ 0 & S_y^{2,2}[I_x] & 0 & \dots \\ \vdots & & \ddots & \\ & & & S_y^{N_y,N_y}[I_x] \end{bmatrix} \quad \text{shape is } (N_y \cdot N_x) \times (N_y \cdot N_x)$$

We can then see that in the simplified example where $N_x = N_y = N$ and $h_x = h_y = h = 1$, and no specific boundaries are set, we get:

$$I_y \otimes S_x + S_y \otimes I_x = \begin{bmatrix} -4 & 1 & \dots & 1 & \dots & & \\ 1 & -4 & 1 & \dots & 1 & \dots & \\ 0 & \ddots & \ddots & \ddots & & & \\ & & & 1 & \dots & 1 & -4 \end{bmatrix} \tag{4}$$

which describes the case stated in Equation 2. Note that the 1's in matrix in Equation 4 are N_x values apart, with zero-values in between.

Question 2

Experimental stability condition for Δt (with parameters $N_x = N_y = 30$, $L_y = L_x = 1$ and $Re = 25$) was empirically found to be

$$\Delta t_{\max} \approx 0.006971$$

when integrated up to $T = 50$.

This empirically found stability condition can now be compared to the theoretical limit presented in the course:

For a general diffusion equation in 2D we have:

$$\frac{\partial u}{\partial t} = \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

Which can be discretized as the following (note that this is a general case, hence no staggered grid):

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = \nu \left[\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{h_x^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{h_y^2} \right] \tag{5}$$

Performing a *von Neumann* analysis on 5, where one proceed as defining

$$u_{i,j}^n = \hat{u}_{kl}^n \cdot e^{ik\alpha x_i} \cdot e^{il\beta y_j}$$

where k and l are the wave number in x and y direction respectively, and inserting the expression in the discretized diffusion equation, one eventually ends up with the following amplification factor:

$$\hat{G} := \frac{\hat{u}_{kl}^{n+1}}{\hat{u}_{kl}^n} = 1 + 2\frac{\nu \Delta t}{h_x^2}(\cos(\phi_k) - 1) + 2\frac{\nu \Delta t}{h_y^2}(\cos(\phi_l) - 1) \tag{6}$$

where $\phi_k := k\alpha h_x$ and $\phi_l = l\beta h_y$.

Considering the condition for stability: $|\hat{G}| \leq 1$, the following condition can be derived:

$$\frac{1}{2} \geq \frac{\nu \Delta t}{h_x^2} + \frac{\nu \Delta t}{h_y^2} \tag{7}$$

In our case, the constant in the general case ν corresponds to $\frac{1}{Re}$ in our setting.

Inserting this to Equation 7, together with the fact that $h_y = h_x = h$ yields the condition:

$$\frac{1}{2} \geq 2\frac{\Delta t}{Re \cdot h^2} \tag{8}$$

In Case A we have $N = 30 \implies h = \frac{1}{30}$, $Re = 25$. If we now insert our estimated stablility limit as Δt, we get the following value:

$$2\frac{0.006971}{25 \cdot \frac{1}{30^2}} \approx 0.501912$$

i.e it above the theoretical limit. If time integrated for longer we would therefore probably get an unstable solution.

For the advective terms, we can arrive at the following stability condition in 2D using the general case:

$$\frac{\partial u}{\partial t} + \vec{a} \cdot \nabla \vec{u} = 0 = \frac{\partial u}{\partial t} + a_x \frac{\partial u}{\partial x} + + a_y \frac{\partial u}{\partial y}$$

Using a similar *von Neumann* analysis as in the diffusive case, one ends up with the following condition:

$$\frac{\Delta t |a_x|}{h_x} + \frac{\Delta t |a_y|}{h_y} \leq 1 \tag{9}$$

which is the sums of the CFL conditions from the separate directions. In our specific setting, the advective terms has a correpsonding a_x and a_y value as 1 in both cases.

Therefore we can write the stability condition for the advective terms as

$$2\frac{\Delta t}{h} \leq 1$$

Comparing the limits for all three cases we get the following:

$$\text{Convective: } \Delta t \leq \frac{h}{2}, \quad \text{Diffusive: } \Delta t \leq \frac{Re \cdot h^2}{4}$$

$$\begin{aligned}
\implies A: \quad & \Delta t \leq \frac{1/30}{2} \approx 0.01667, \quad \Delta t \leq \frac{25 \cdot 1/30^2}{4} \approx 0.006944 \\
B: \quad & \Delta t \leq \frac{1/50}{2} = 0.01, \quad \Delta t \leq \frac{250 \cdot 1/50^2}{4} = 0.025 \\
C: \quad & \Delta t \leq \frac{1/100}{2} = 0.005, \quad \Delta t \leq \frac{5000 \cdot 1/100^2}{4} = 0.125
\end{aligned}$$

We conclude that as the Re-number increases, the advective terms are the ones setting the stability limit.

Question 3

In the given lid-driven cavity problem we have the following boundary condition that should be fulfilled in each time point:

$$\begin{aligned}
& u = U_{\text{bottom}}, \quad v = 0, \quad (x, y) \in \Gamma_b \\
& u = U_{\text{top}}, \quad v = 0, \quad (x, y) \in \Gamma_t \\
& u = v = 0, \quad (x, y) \in \Gamma_r \cup \Gamma_l \\
& \frac{\partial p}{\partial x_i} n_i = 0, \quad (x, y) \in \Gamma_b \cup \Gamma_t \cup \Gamma_l \cup \Gamma_r
\end{aligned}$$

Now, since a staggered grid is used in the discretization, we set these conditions in the following way:

First we have the boundary conditios that we are supposed to look at, to then determine the necessary ghost points.

$$u_{i+\frac{1}{2},\frac{1}{2}} = U_{\text{bottom}}, \quad v_{\frac{1}{2},j+\frac{1}{2}} = 0 \tag{10}$$

These boundary conditions are as seen given. We can also set up the two following conditions:

$$\begin{aligned} u_{i+\frac{1}{2},\frac{1}{2}} &= \frac{u_{i+\frac{1}{2},1} + u_{i+\frac{1}{2},0}}{2} = U_{\text{bottom}} \\ v_{\frac{1}{2},j+\frac{1}{2}} &= \frac{v_{0,j+\frac{1}{2}} + v_{1,j+\frac{1}{2}}}{2} = 0 \\ \implies &\begin{cases} u_{i+\frac{1}{2},0} = 2U_{\text{bottom}} - u_{i+\frac{1}{2},1} \\ v_{0,j+\frac{1}{2}} = 2 \cdot 0 - v_{1,j+\frac{1}{2}} = -v_{1,j+\frac{1}{2}} \end{cases} \end{aligned}$$

where $v_{0,j+\frac{1}{2}}$ and $u_{i+\frac{1}{2},0}$ are the ghost points that we use to fulfill the boundare conditions.

Similarily, for the ghost points for the pressure: $p_{0,j}$ and $p_{i,0}$ we proceed as follows:

We have the Neumann condition that the derivative normal to the boundary, at the boundary is zero:

$$\begin{aligned} \frac{\partial p}{\partial n}|_{0,j} &\approx \frac{p_{1,j} - p_{0,j}}{h_x} = 0 \\ \frac{\partial p}{\partial n}|_{i,0} &\approx \frac{p_{i,1} - p_{i,0}}{h_y} = 0 \\ \implies &\begin{cases} p_{0,j} = p_{1,j} \\ p_{i,0} = p_{i,1} \end{cases} \end{aligned}$$

Combining the above with the Laplace equation for p, we get the following:

$$\begin{aligned} \frac{p_{0,j} - 2p_{1,j} + p_{2,j}}{h_x^2} &= \frac{-p_{1,j} + p_{2,j}}{h_x^2} \\ \frac{p_{i,0} - 2p_{i,1} + p_{i,2}}{h_y^2} &= \frac{-p_{i,1} + p_{i,2}}{h_y^2} \end{aligned}$$

Therefore, the second derivative operator DD in the python code, gets its first and last element changed to $-1/h^2$. The full matrices are shown in Equation 3.

Question 4

Placing a probe in the domian centre gives the following velocity U along the time line to $T = 100$:

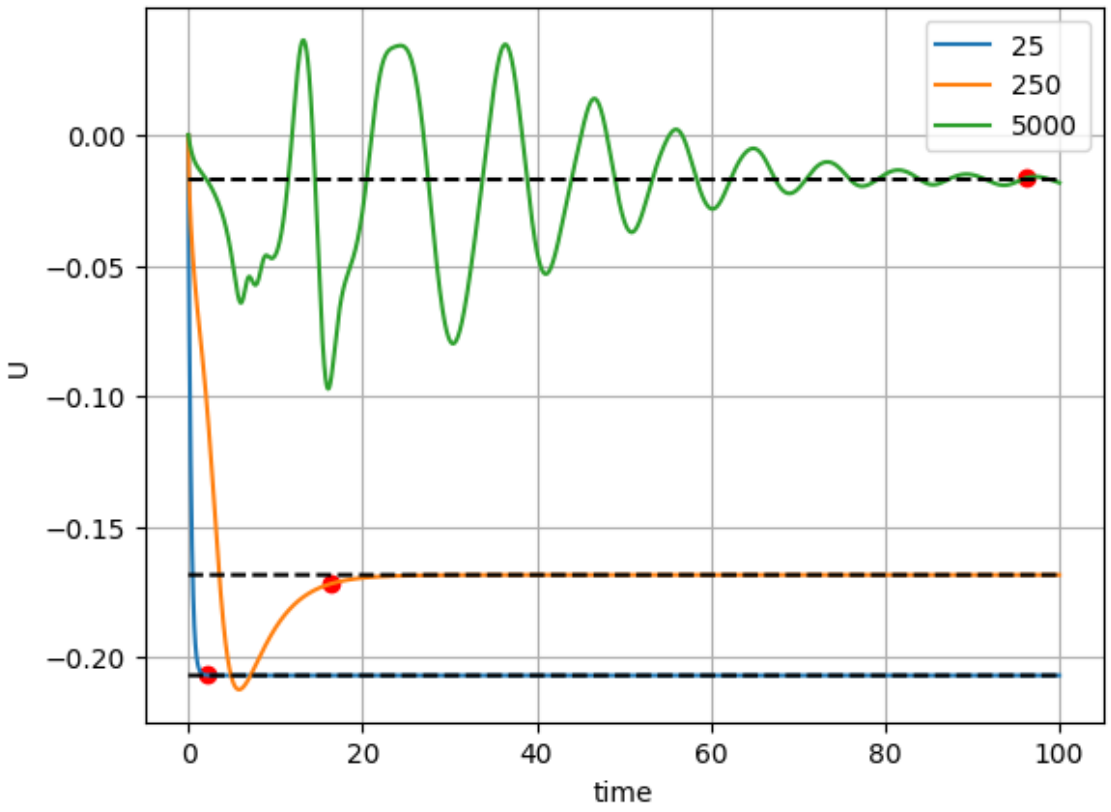


Figure 2: U in middle of domain for different Re-values

Clearly the cases with lower Reynold's number converges to a steady flow quicker than the cases with a higher number. This is due to the amount of turbulance is greater in the case with higher Re; the intertial forces are greater in comparison to the viscous forces when the Re increases.

To in a more concrete way decide when a steady flow is reached in the location of the probe, the first and second derivative of U w.r.t t is taken into account. When setting up three conditions, which measures when the derivatives doesn't overshoot the value 0.001 and deviation from the median doesn't overshoot 5%. With the following lines of code:

```
du_vec = np.diff(u_vec, axis = 0)/np.diff(t)[0]
ddu_vec = np.diff(u_vec, axis = 0, n = 2)/(np.diff(t)[0])**2

for col in range(u_vec.shape[1]):
    cond1 = np.abs(u_vec[2:,col]-np.median(u_vec,axis = 0)[col]) < \
        0.05 * np.abs(np.median(u_vec,axis = 0)[col])
    cond2 = np.abs(du_vec[1:,col]) < 0.001
    cond3 = np.abs(ddu_vec[:,col]) < 0.001

    cond = cond1*cond2*cond3

    index = np.argmax(cond)
```

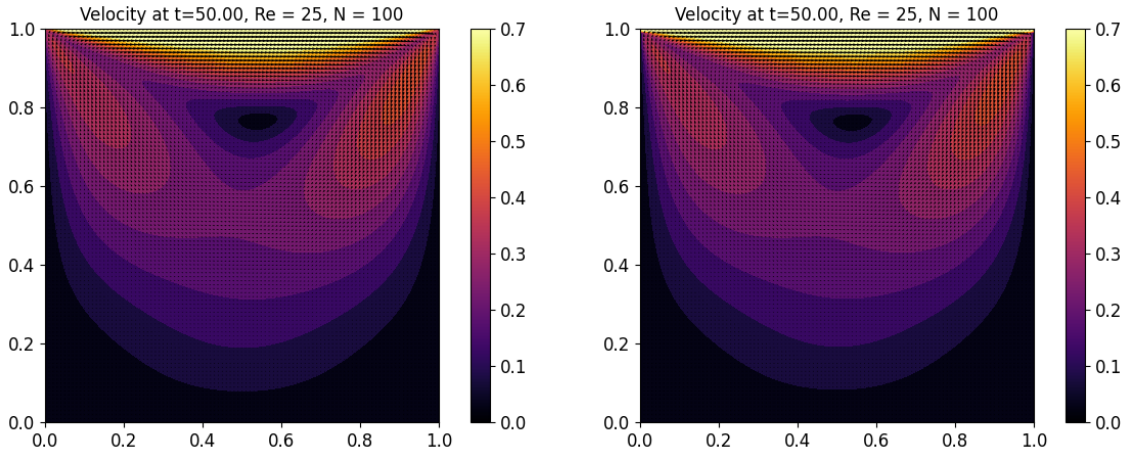
the times for steady flow was estimated to be:

```
Time to steady flow when Re = 25: 2.165102165102165 s
Time to steady flow when Re = 250: 16.342916342916343 s
Time to steady flow when Re = 5000: 96.2003962003962 s
```

This is also marked with red points in Figure 2.

Question 5

Using OpenFoam in each of the cases A-C the following data were obtained: Figure 3 shows both a solution from the Python implementation as well as OpenFOAM solution with the data visualized using matplotlib.



(a) OpenFoam solution

(b) Python Solution

Figure 3: Case A

For a more detailed comparison we can observe the velocity magnitude along the diagonal of the domain, so called a *plot over line* visualization. For case A we have the following:

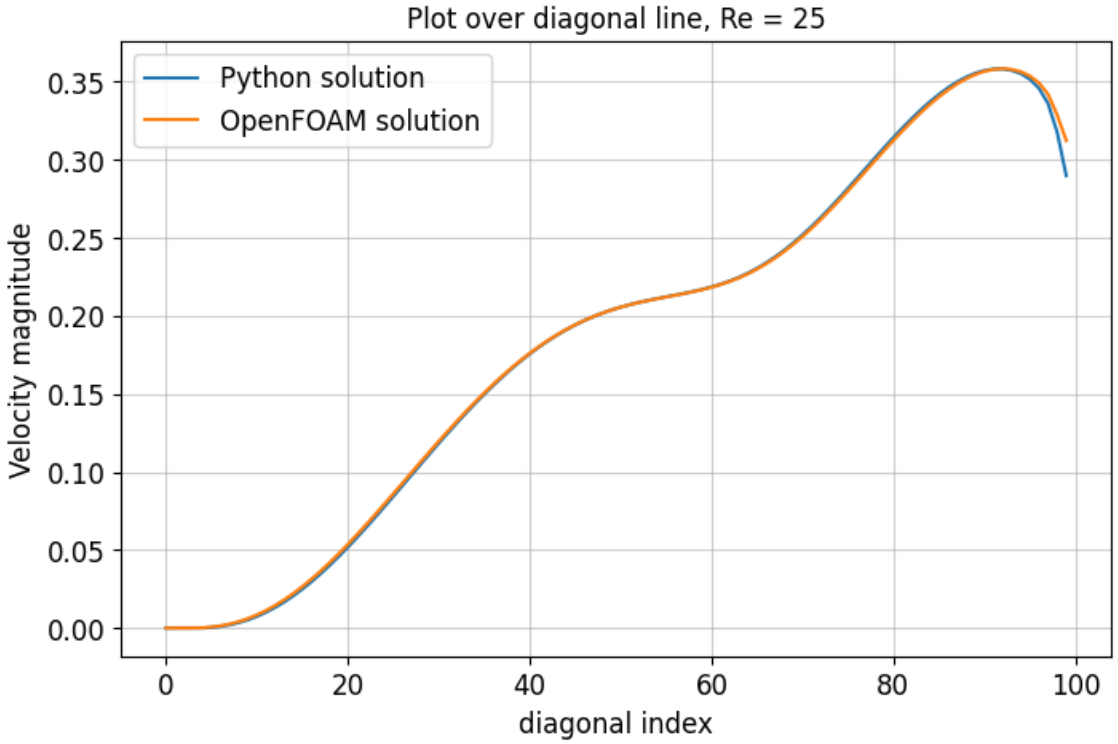
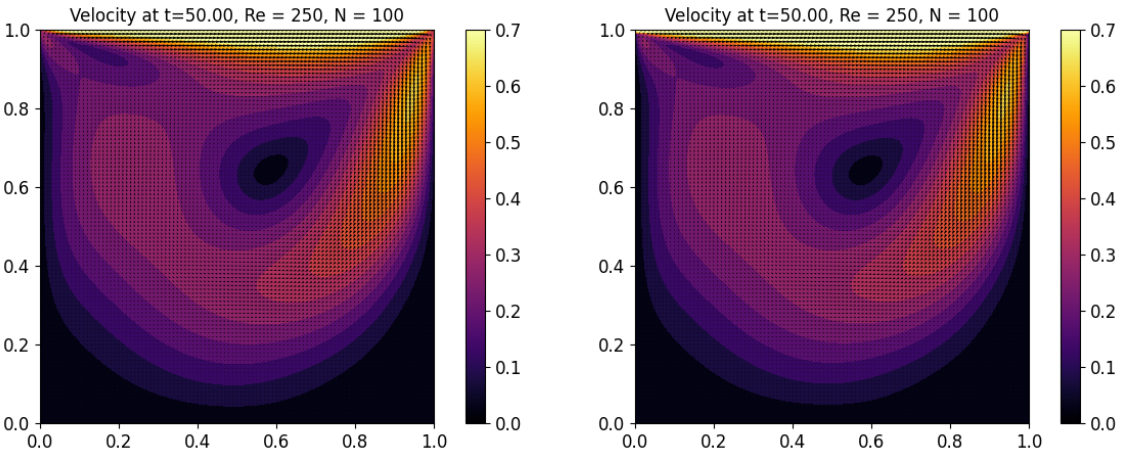


Figure 4: Plot over line (Case A): Python against OpenFOAM

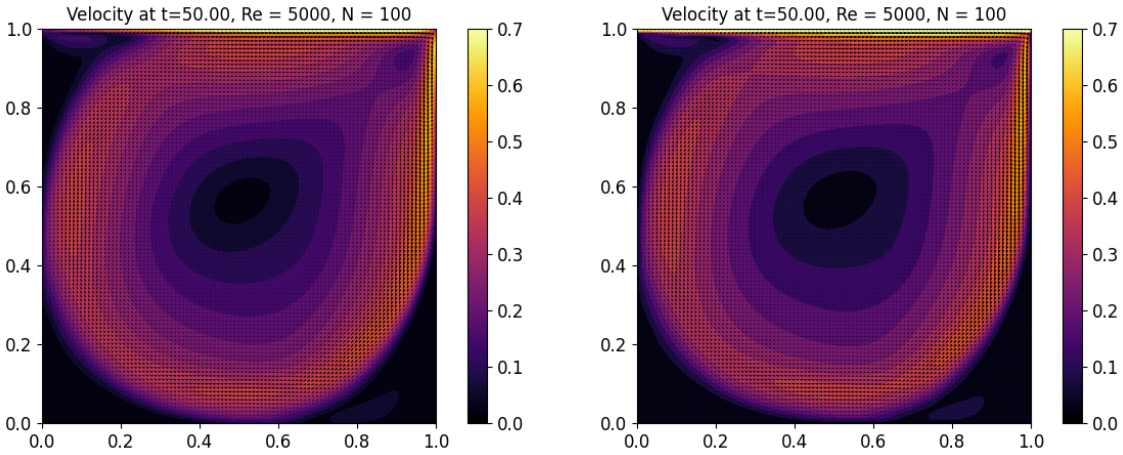
Likewise we can compare case B and C in a similar manner:



(a) OpenFoam solution

(b) Python Solution

Figure 5: Case B



(a) OpenFoam solution (b) Python Solution

Figure 6: Case C

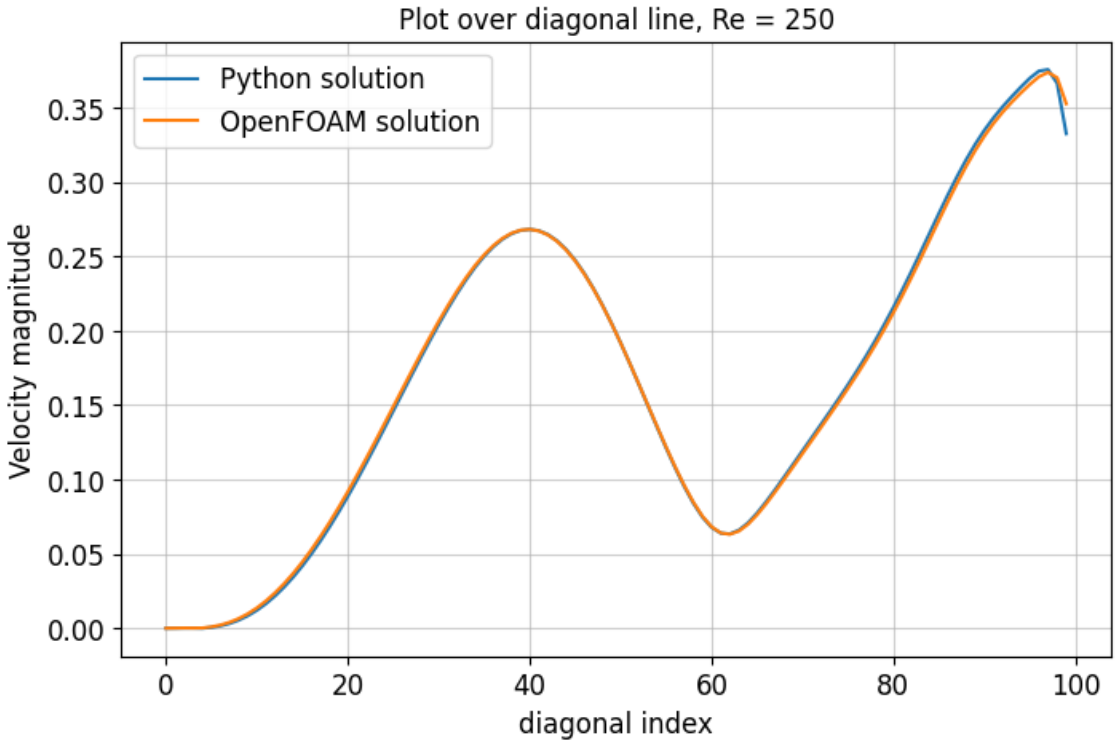


Figure 7: Plot over line (Case B): Python against OpenFOAM

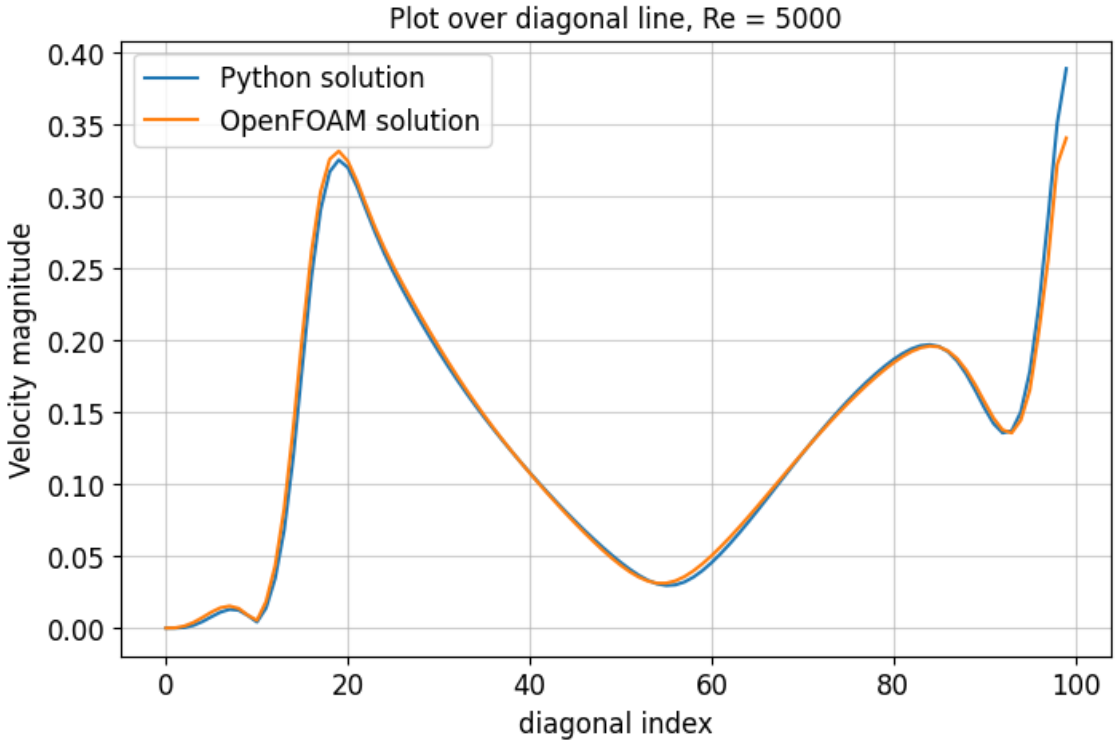


Figure 8: Plot over line (Case C): Python against OpenFOAM

We can conclude that the Python and OpenFOAM solutions converge to the same solution, and notably the largest differences were found close to the boundaries. In the above plots the boundary point (L_x, L_y) was actually excluded due to OpenFOAM didn't include it in the solution.

Comparing the relative L_2-norm between the different steady solutions (OpenFOAM against Python) we got the following data:

```
Running case for Re = 25
Relative error: 0.04912371466615295

Running case for Re = 250
Relative error: 0.05397593894314182

Running case for Re = 5000
Relative error: 0.09147688319684655
```

Question 6

When also including a lower wall with the same velocity as the upper lid, the following solution was obtained at $T = 50$:

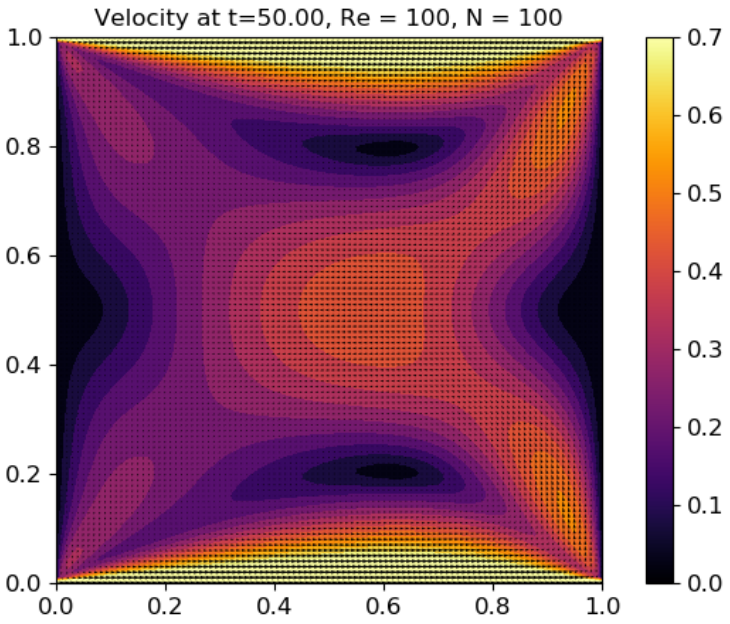


Figure 9: Case D

The visualization in Figure 9 clearly shows the symmetry in the velocity field at $y = 0.5$, as the fluid flow generated from the top and bottom meet in the middle of the domain.