SG2212 Project: Development of a 2D Navier-Stokes Solver

Jonathan de Cruz

24th March 2024

1 Problem Formulation

In this project, we consider the problem of lid-driven cavity flow, which requires a solution to the Navier-Stokes equations and the accompanying scalar temperature equation. The flow domain is square, with no-slip boundary conditions on all the walls (Figure 1). The walls are all stationary except for the top, which is moving to the right with speed u_{top}, and the fluid is assumed to be incompressible. This flow is governed by the Navier-Stokes equations, which (in non-dimensionalised form) are:

$$\frac{\partial u_i}{\partial t} + \frac{\partial \left(u_i u_i\right)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i \tag{1}$$

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{2}$$

where f_i is a volume force. The corresponding transport equation for a scalar θ (such as temperature) reads:

$$\frac{\partial \theta}{\partial t} + \frac{\partial \left(u_j \theta\right)}{\partial x_j} = \frac{1}{Pe}\frac{\partial^2 \theta}{\partial x_j \partial x_j} \tag{3}$$

where $Pe = PrRe$ is the Péclet number and Pr the Prandtl number. Both of these systems of equations need boundary and initial conditions in order to become well-posed problems, and in the case of the Navier-Stokes equations these are comprised of the no-slip conditions on each wall, as well as the assumption of an initially non-moving fluid. We denote the domain by $\Omega := [0, \ell_x] \times [0, \ell_y]$, and the walls by Γ (Γ_t, Γ_b, Γ_l and Γ_r for the top, bottom, left and right walls respectively). Additionally, we specify Neumann boundary conditions all around the domain for the pressure. Then, the full Navier-Stokes problem is:

$$\frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} = -\frac{\partial \left(u^2\right)}{\partial x} - \frac{\partial (uv)}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \quad (x, y) \in \Omega, \quad t > 0$$

$$\frac{\partial v}{\partial t} + \frac{\partial p}{\partial y} = -\frac{\partial (uv)}{\partial x} - \frac{\partial \left(v^2\right)}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right), \quad (x, y) \in \Omega, \quad t > 0$$

$$\begin{aligned}
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0, \quad (x, y) \in \Omega, \quad t > 0 \\
u = v &= 0, \quad (x, y) \in \Gamma_l \cup \Gamma_r \cup \Gamma_b, \quad t > 0 \\
u = u_{top}, v &= 0, \quad (x, y) \in \Gamma_t, \quad t > 0 \\
\frac{\partial p}{\partial n} &= 0, \quad (x, y) \in \Gamma_t \cup \Gamma_l \cup \Gamma_b \cup \Gamma_r, \quad t > 0 \\
u = 0, v &= 0, \quad (x, y) \in \Omega, \quad t > 0
\end{aligned}$$

The n in the Neumann boundary conditions for the pressure refers to the inward-pointing normal direction to the walls. Similarly, the top and bottom walls are assumed to be at set temperatures, with the side walls being adiabatic. The initial condition for the temperature is a linear distribution in the vertical direction across the flow domain.

$$\frac{\partial \theta}{\partial t} = -\frac{\partial (u\theta)}{\partial x} - \frac{\partial (v\theta)}{\partial y} + \frac{1}{Pe}\left(\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2}\right), \quad (x, y) \in \Omega, \quad t > 0$$

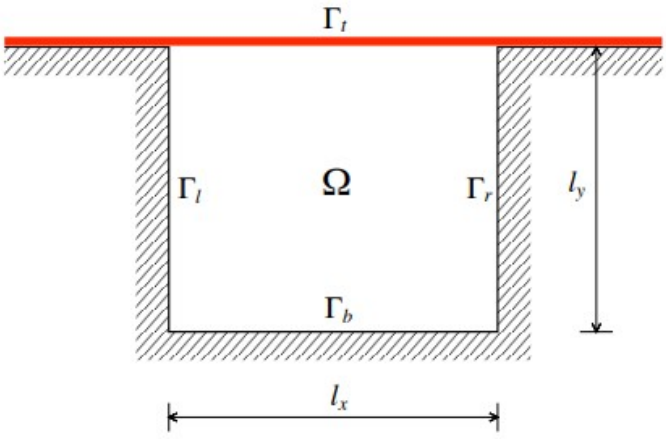

Figure 1: Flow Domain for the Lid-Driven Cavity Problem

$$\begin{aligned}
\frac{\partial \theta}{\partial x} &= 0, \quad (x,y) \in \Gamma_l \cup \Gamma_r, \quad t > 0 \\
\theta &= 0, \quad (x,y) \in \Gamma_t, \quad t > 0 \\
\theta &= 1, \quad (x,y) \in \Gamma_b, \quad t > 0 \\
\theta &= \frac{y}{\ell_y}, \quad (x,y) \in \Omega, \quad t = 0
\end{aligned}$$

Because the Navier-Stokes equations are the more difficult system to solve, we consider it first. Here, we shall solve the system numerically using the projection method. Being a numerical method, this requires discretisation of the equations and the domain, and we choose the spatial discretisation to be on a staggered grid, as shown in Figure 2. This avoids odd-even decoupling and the associated spurious solutions. This produces matrices `U` and `V` (of size $(N_x - 1) \times N_y$ and $N_x \times (N_y - 1)$ respectively), containing the velocity values at the inner nodes only. In order to add the boundary values and the ghost nodes, we define the extended velocity matrices `Ue` and `Ve`, which have sizes $(N_x + 1) \times (N_y + 2)$ and $(N_x + 2) \times (N_y + 1)$. The boundary conditions must be treated carefully here, since for each matrix, there is not necessarily a node present on the required boundary. Hence, looking at Figure 2, the boundary conditions can be applied directly to the left and right walls for `U`, and to the top and bottom walls for `V`. For the other boundaries, we must define ghost nodes which give the correct values on the boundaries by interpolation.

We also use the explicit (forward) Euler method for the time derivative, and transform the derivative operators in the equations into matrices acting on the solution vectors. With these, then, the Navier-Stokes equations become

$$\frac{\underline{u}^{n+1} - \underline{u}^n}{\Delta t} + \underline{\underline{N}}\left(\underline{u}^n, \underline{u}^n\right) - \frac{1}{Re}\underline{\underline{L}}\underline{u}^n + \underline{\underline{G}}\underline{p}^{n+1} = \underline{f}^n$$

$$\underline{\underline{D}}\underline{u}^{n+1}$$

where $\underline{u}$ denotes the velocity vector, $\underline{\underline{N}}(\cdot,\cdot)$ the discretised nonlinear advection operator, $\underline{\underline{L}}$ the Laplacian operator, $\underline{\underline{G}}$ the gradient, $\underline{\underline{D}}$ the divergence, $\underline{f}$ the discretised forcing terms including the boundary conditions, and Δt the time step. In this flow case, the forcing terms $\underline{f}$ are set to zero. Then, the projection method for this system proceeds as follows:

1. In the first half-step, an intermediate, non-divergence-free velocity field is computed by solving the following convection-diffusion equation:

$$\frac{\underline{u}^* - \underline{u}^n}{\Delta t} + \underline{\underline{N}}\left(\underline{u}^n, \underline{u}^n\right) - \frac{1}{Re}\underline{\underline{L}}\underline{u}^n = \underline{f}^n$$

   Note that in order to compute the nonlinear advection terms and the viscous terms ($\underline{\underline{L}}\underline{u}^n$), we need to average the extended velocity matrices `Ue` and `Ve` in the appropriate directions to ensure that the matrix dimensions are consistent throughout the operation.

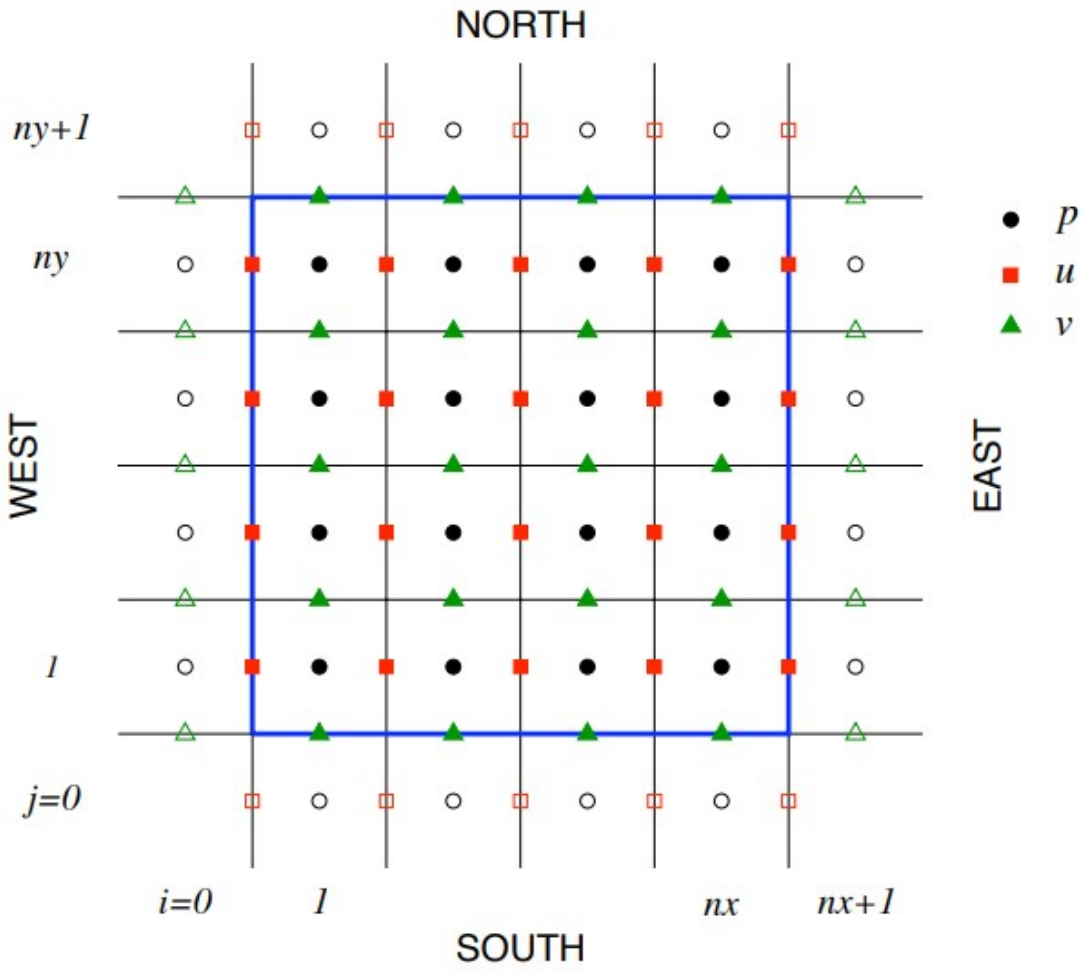

Figure 2: Staggered Grid for Spatial Discretisation

2. The second half-step consists of solving the following problem with both velocity and pressure as unknowns:

$$\frac{\underline{u}^{n+1} - \underline{u}^*}{\Delta t} + \underline{\underline{G}}\,\underline{p}^{n+1} = 0$$

$$\underline{\underline{D}}\,\underline{u}^{n+1} = 0$$

Applying the divergence operator to this step and and using the divergence-free condition gives a Poisson equation for $\underline{p}^{n+1}$:

$$\underline{\underline{D}}\,\underline{\underline{G}}\,\underline{p}^{n+1} = \underline{\underline{L}}\,\underline{p}^{n+1} = \frac{1}{\Delta t}\underline{\underline{D}}\,\underline{u}^*$$

In order to solve this, we make use of the matrix formulation of the equations, and simply multiply the inverse of the Laplacian matrix by the right-hand side of the equation (which can be made into a vector). The pressure thus computed can be used to obtain an approximation of $\underline{u}^{n+1}$.

$$\underline{u}^{n+1} = \underline{u}^* - \Delta t \underline{\underline{G}}\,\underline{p}^{n+1}$$

It should be noted that since we are only prescribing Neumann boundary conditions for the pressure, the numerical solution method will not specify it uniquely, leaving an arbitrary constant to be found. This problem can be alleviated by specifying the pressure (i.e. applying Dirichlet conditions) to a single point in the domain, and here we choose to set the pressure at the lower left-hand corner equal to zero.

Now, we can consider the scalar transport equation for the temperature in the domain. The temperature is defined initially in the matrix `T`, with size $N_x \times N_y$, representing the temperature field at the pressure nodes. Hence, because there are no temperature nodes directly lying on the boundaries, we need to apply ghost nodes all around the domain. The temperature at each of the ghost nodes is found by requiring that they interpolate to the required values on the top and bottom walls (since the boundary conditions are Dirichlet there). For the left and right walls, where there are only Neumann conditions, it is required that the numerical derivative taken over the boundaries vanishes. This gives unique values for each of the ghost nodes. It is important to realise also that we must add ghost temperature nodes outside the corners of the domain (even though there are no pressure nodes there), in order to keep the temperature matrix rectangular. Being a simpler formulation than the Navier-Stokes equations, the temperature equation can be solved by straightforward time integration, again choosing the explicit Euler method for the time discretisation. We have the following numerical equation:

$$\frac{\underline{\theta}^{n+1} - \underline{\theta}^n}{\Delta t} = -\underline{\underline{D}}(\underline{u}\underline{\theta})^n + \frac{1}{Pe}\underline{\underline{L}}\,\underline{\theta}^n$$

which can be solved for $\underline{\theta}^{n+1}$ at each time instant by simple algebraic manipulation.

2 Simulation Results

2.1 Case A

Case A involved running the lid-driven cavity problem with a Reynolds number of $Re = 25$ until $t = 50$, with the top wall moving at $u_{top} = 1$, and $\theta_{top} = 0, \theta_{bottom} = 1$. With the Reynolds number being so low, this flow is nearly symmetrical with respect to the vertical mid-line, as Figure 3 shows. There is a "vortex" core approximately positioned on this line, where the velocity is nearly zero and about which the flow rotates. There are regions of higher velocity near the top corners of the domain, and the maximum velocity is observed at the moving top wall.

The temperature field is relatively unchanged by the fluid motion - it can still be seen that the initial conditions are a linear distribution from the top to the bottom of the domain. Obviously, there is some asymmetry due to temperature advection, but the low Reynolds number ensures that the field as a whole is affected only slightly.

2.2 Case B

For Case B, the Reynolds number was changed to $Re = 250$. Here, Figure 4 shows that the flow is now asymmetrical about the vertical mid-line at $t = 50$. There is a large region of increased velocity at the top right-hand corner, with no such region on the left-hand side. The vortex core can still be seen, though it is now moved slightly down and to the right in comparison to Case A. The temperature is more affected in this case, with a large region of lower temperature in the middle of the domain - lower-temperature fluid is advected into this region by the velocity profile. The higher temperatures are confined to a smaller area near the bottom wall, with higher gradients.

2.3 Case C

At $Re = 5000$ and $t = 50$, the flow is no longer steady, and there is a "ring" of higher velocity around the vortex core, which is now situated approximately at the centre of the domain (Figure 5). There are very large velocity gradients at the top of the domain near the moving wall, which are necessary to ensure that the boundary conditions are met. In this case, higher domain resolution ($N_x = N_y = 100$) was needed to make sure that the boundary layers in the flow were correctly simulated, as exemplified by the boundary layer at the top and right-hand walls. As for the temperature, the central region of lower temperature is now dominant, with the higher temperatures confined to a very small area near the bottom corners of the domain.

In order to make sure that the simulation produced a steady flow, it was run until convergence, that is, until approximately $t = 300$s. These produced the velocity and temperature fields as seen in Figure 6, which are quite similar to the unconverged examples.

2.4 Case D

Once the flow analysis using the developed MATLAB code was finished, it could be validated using OpenFOAM, which is a tried-and-tested software suitable for a great range of CFD applications. The lid-driven cavity case was run on OpenFOAM in order to mimic Case C, using a Reynolds number of $Re = 5000$, a rectangular grid with spacing $\Delta x = \Delta y = 0.01$, and a time step of $\Delta t = 0.001$s. Running the simulation until $t = 50$s produced the velocity field presented in Figure 7, which was then imported into MATLAB for post-processing and plotting. It should be noted that in OpenFOAM, the Reynolds number may not be set directly, so the kinematic viscosity ν was chosen to ensure $Re = 5000$:

$$Re = \frac{u_{top}\ell_x}{\nu} \quad \therefore \quad \nu = \frac{u_{top}\ell_x}{Re} = \frac{1 \times 1}{5000} = 0.0002$$

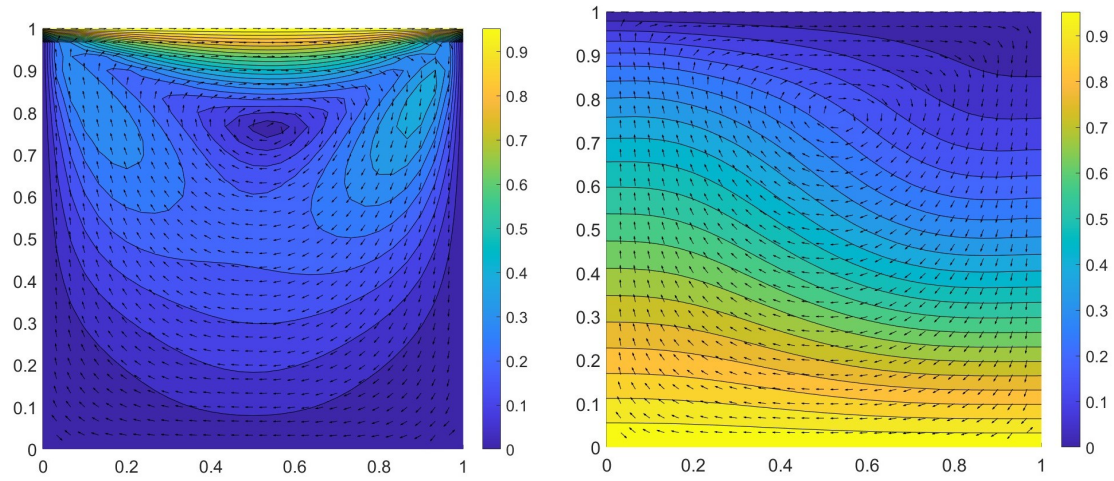

Figure 3: Velocity (left) and Temperature (right) Fields for Case A

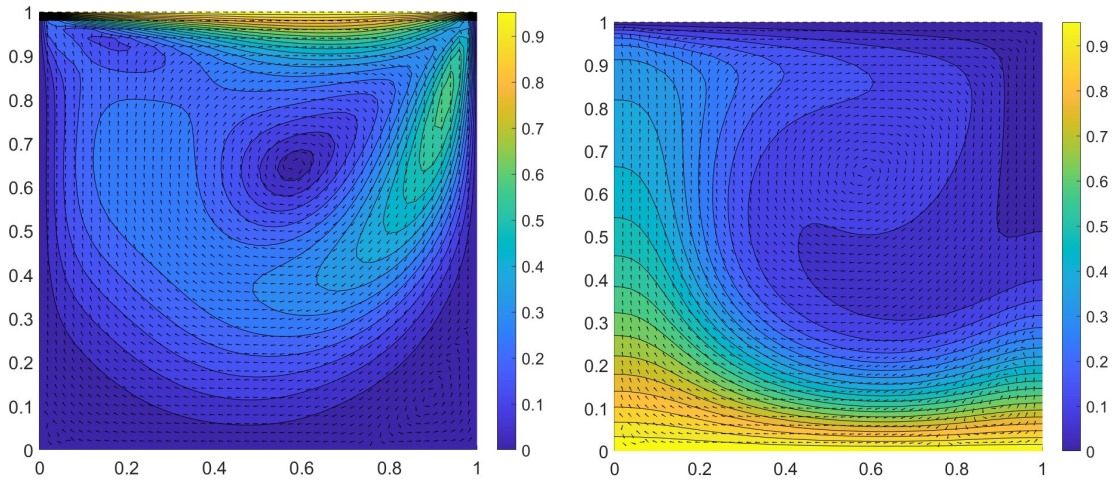

Figure 4: Velocity (left) and Temperature (right) Fields for Case B

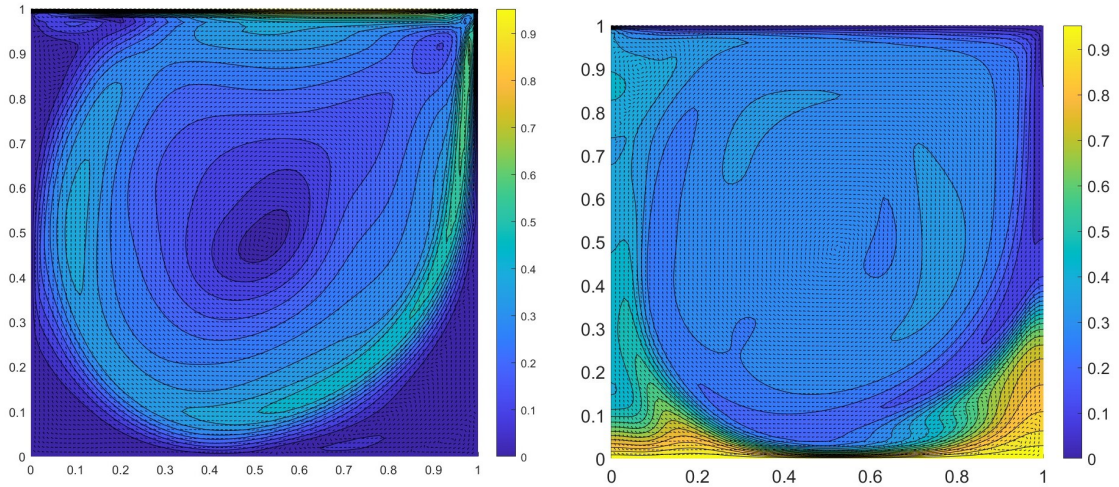

Figure 5: Velocity (left) and Temperature (right) Fields for Case C until $t = 50$s

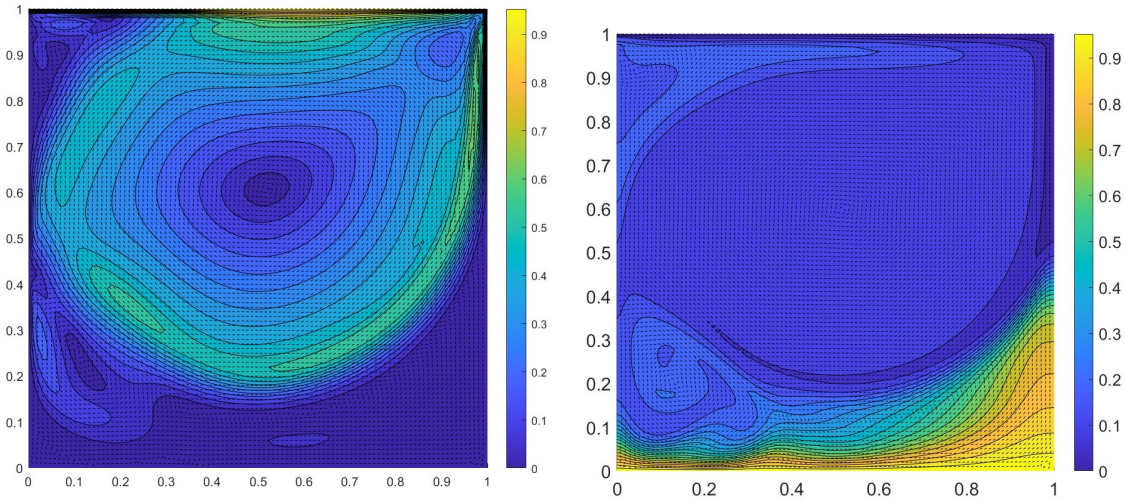

Figure 6: Velocity (left) and Temperature (right) Fields for Case C until $t = 300$s

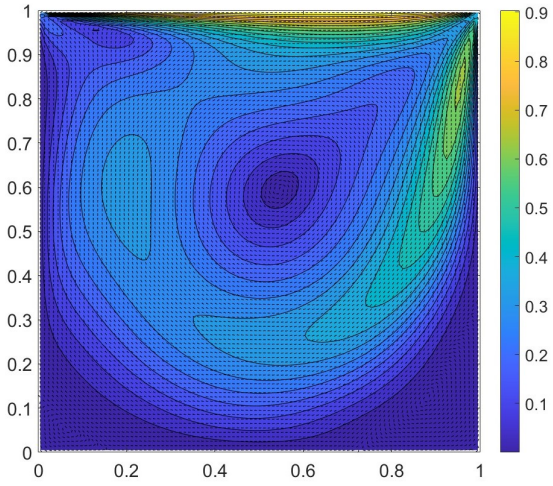

Figure 7: Velocity Field at $t = 50$s with $Re = 5000$ Using OpenFOAM

3 Additional Questions

Question 1

Discuss why the `kron` *operator (Kronecker tensor product) returns the matrix for the second derivative in the two-dimensional case.*

If $\underline{\underline{A}}$ is an $m \times n$ matrix and $\underline{\underline{B}}$ is a $p \times q$ matrix, then the Kronecker tensor product $\underline{\underline{A}} \otimes \underline{\underline{B}}$ is a $pm \times qn$ block matrix:

$$\underline{\underline{A}} \otimes \underline{\underline{B}} = \begin{bmatrix} a_{11}\underline{\underline{B}} & \cdots & a_{1n}\underline{\underline{B}} \\ \vdots & \ddots & \vdots \\ a_{m1}\underline{\underline{B}} & \cdots & a_{mn}\underline{\underline{B}} \end{bmatrix}$$

In order to take the two-dimensional Laplacian of the pressure field, we use central differences in both directions:

$$\nabla^2 p \approx \frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{\Delta x^2} + \frac{p_{i,j+1} - 2p_{i,j} + p_{i,j-1}}{\Delta y^2}$$

However, as detailed in the problem formulation (Section 1), the Laplacian operator can be thought of as a matrix, with the pressure and velocity values at each of the nodes being vectors of length $N_x \times N_y$. For the pressure, for example, the vector will be:

$$\underline{p} = \left[p_{1,1}, p_{2,1}, ..., p_{N_x,1}, p_{1,2}, p_{2,2}, ..., p_{N_x-1,N_y}, p_{N_x,N_y}\right]^T$$

Then, the Laplacian operator $\underline{\underline{L}}$ needs to be a $N_x N_y \times N_x N_y$ matrix with the appropriate values. We have the following matrices for the second derivative in the x- and y-directions:

$$\underline{\underline{D}}_{2x} = \frac{1}{\Delta x^2}\begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 \\ 0 & 0 & 0 & \cdots & 1 & -1 \end{bmatrix}, \quad \underline{\underline{D}}_{2y} = \frac{1}{\Delta y^2}\begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -2 & 1 \\ 0 & 0 & 0 & \cdots & 1 & -1 \end{bmatrix}$$

Note that these derivative matrices already include homogeneous Neumann boundary conditions, which are discussed further in Question 3 below.

We then want to know why the two-dimensional Laplacian matrix is given by the following Kronecker sum:

$$\underline{\underline{L}} = \underline{\underline{I}}_{N_y \times N_y} \otimes \underline{\underline{D}}_{2x} + \underline{\underline{D}}_{2y} \otimes \underline{\underline{I}}_{N_x \times N_x}$$

We may split this sum into two separate Kronecker products, as below.

$$\underline{\underline{I}}_{N_y \times N_y} \otimes \underline{\underline{D}}_{2x} = \begin{bmatrix} \underline{\underline{D}}_{2x} & \cdots & \underline{\underline{0}} \\ \vdots & \ddots & \vdots \\ \underline{\underline{0}} & \cdots & \underline{\underline{D}}_{2x} \end{bmatrix},$$

$$\underline{\underline{D}}_{2y} \otimes \underline{\underline{I}}_{N_x \times N_x} = \frac{1}{\Delta y^2}\begin{bmatrix} -\underline{\underline{I}}_{N_x \times N_x} & \underline{\underline{I}}_{N_x \times N_x} & \underline{\underline{0}} & \cdots & \underline{\underline{0}} \\ \underline{\underline{I}}_{N_x \times N_x} & -2\underline{\underline{I}}_{N_x \times N_x} & \underline{\underline{I}}_{N_x \times N_x} & \cdots & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{I}}_{N_x \times N_x} & -2\underline{\underline{I}}_{N_x \times N_x} & \cdots & \underline{\underline{0}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{0}} & \cdots & -\underline{\underline{I}}_{N_x \times N_x} \end{bmatrix}$$

Then, for a general point $p_{i,j}$ in the flow domain (assuming for the moment that it is located away from the boundaries), corresponding to the entry $\underline{p}[i+(j-1)N_x]$ in the pressure vector, the Laplacian will be given by multiplying $\underline{\underline{L}}$ and $\underline{p}$:

$$\underline{\underline{L}}\underline{p}[i + (j-1)N_x] = \sum_{k=1}^{N_x} p_{k,j-1}\left[\frac{1}{\Delta y^2}\underline{\underline{I}}_{N_x \times N_x}(i,k)\right] + \sum_{k=1}^{N_x} p_{k,j}\left[-\frac{2}{\Delta y^2}\underline{\underline{I}}_{N_x \times N_x}(i,k) + \underline{\underline{D}}_{2x}(i,k)\right] + \sum_{k=1}^{N_x} p_{k,j+1}\left[\frac{1}{\Delta y^2}\underline{\underline{I}}_{N_x \times N_x}(i,k)\right]$$

$$\underline{\underline{L}}\underline{p}[i+(j-1)N_x] = \frac{1}{\Delta y^2}p_{i,j-1} - \frac{2}{\Delta y^2}p_{i,j} + \frac{1}{\Delta x^2}p_{i-1,j} - \frac{2}{\Delta x^2}p_{i,j} + \frac{1}{\Delta x^2}p_{i+1,j} + \frac{1}{\Delta y^2}p_{i,j+1}$$

$$\underline{\underline{L}}\underline{p}[i+(j-1)N_x] = \frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{\Delta x^2} + \frac{p_{i,j+1} - 2p_{i,j} + p_{i,j-1}}{\Delta y^2}$$

Thus

$$(\nabla^2 p)_{i,j} \approx \underline{\underline{L}}\underline{p}[i+(j-1)N_x]$$

as required. Hence, the Laplacian matrix must be, at least for the inner nodes, as given by the Kronecker sum referred to above:

$$\underline{\underline{L}} = \underline{\underline{I}}_{N_y \times N_y} \otimes \underline{\underline{D}}_{2x} + \underline{\underline{D}}_{2y} \otimes \underline{\underline{I}}_{N_x \times N_x}$$

By a similar argument, we may also show that this formulation is suitable for the boundary values, since the one-dimensional second derivative matrices ($\underline{\underline{D}}_{2x}$ and $\underline{\underline{D}}_{2y}$) already include the required values. The boundary conditions for the pressure, for example, are homogeneous Neumann as below:

$$\frac{\partial p}{\partial n} = 0 \quad \text{on } \partial\Omega$$

where the n refers to the inward-pointing normal to each successive boundary wall. While more details on this are given in Question 3, this means that we have the following formulations for the ghost pressure nodes:

$$\begin{aligned} \text{North} &: p_{i,N_y+1} = p_{i,N_y} \\ \text{South} &: p_{i,0} = p_{i,1} \\ \text{West} &: p_{0,j} = p_{1,j} \\ \text{East} &: p_{N_x+1,j} = p_{N_x,j} \end{aligned}$$

Then, taking the boundary condition at the bottom-left corner as an example and substituting these formulations into the discretised Laplacian, we are left with

$$(\nabla^2 p)_{1,1} \approx \frac{p_{2,1} - p_{1,1}}{\Delta x^2} + \frac{p_{1,2} - p_{1,1}}{\Delta y^2}$$

and this is what we require from the Laplacian matrix. Multiplying $\underline{\underline{L}}$ by $\underline{p}$ then gives:

$$\underline{\underline{L}}\underline{p}(1) = \sum_{k=1}^{N_x} p_{k,1}\left[-\frac{1}{\Delta y^2}\underline{\underline{I}}_{N_x \times N_x}(1,k) + \underline{\underline{D}}_{2x}(1,k)\right] + \sum_{k=1}^{N_x} p_{k,2}\left[\frac{1}{\Delta y^2}\underline{\underline{I}}_{N_x \times N_x}(1,k)\right]$$

$$\underline{\underline{L}}\underline{p}(1) = -\frac{1}{\Delta y^2}p_{1,1} + \frac{1}{\Delta x^2}\left(-p_{1,1} + p_{2,1}\right) + \frac{1}{\Delta y^2}p_{1,2}$$

$$\underline{\underline{L}}\underline{p}(1) = \frac{p_{2,1} - p_{1,1}}{\Delta x^2} + \frac{p_{1,2} - p_{1,1}}{\Delta y^2}$$

as we required. Thus, the Laplacian matrix is also valid at the boundaries as well as at the inner points of the domain. Therefore, it is completely justified to use the Kronecker tensor product to build the two-dimensional Laplacian matrix. QED.

Question 2

For case A, determine "experimentally" the maximum time step Δt for which the simulation is stable. Can you relate that maximum time step to the grid resolution and Reynolds number by considering stability conditions for both convective and diffusive terms? Which of the two is more strict for cases A–C? Choosing $Pr = 0.71$, is the scalar equation less stable than the momentum equations? Why?

For the convective part of the Navier-Stokes equations, we study the two-dimensional advection equation given below.

$$\frac{\partial U}{\partial t} + a_x\frac{\partial U}{\partial x} + a_y\frac{\partial U}{\partial y} = 0$$

(Here, U may stand for either the horizontal or vertical components of velocity). The quantities a_x and a_y are the convective velocities in the x- and y-directions respectively, which for the non-linear Navier-Stokes equations can be taken as the maximum absolute values of u and v. Then, in order

for the numerical discretisation of this equation to be stable, we require that the CFL condition is satisfied. This condition states that the domain of dependence of the finite-difference scheme should completely contain the domain of dependence of the continuous PDE. For the flow considered in this project, this is written as:

$$a_x \frac{\Delta t}{\Delta x} + a_y \frac{\Delta t}{\Delta y} \leq 1$$

where we can define the Courant numbers σ_x and σ_y:

$$\sigma_x = a_x \frac{\Delta t}{\Delta x}, \quad \sigma_y = a_y \frac{\Delta t}{\Delta y}$$

$$\implies \sigma_x + \sigma_y \leq 1$$

When $\Delta x = \Delta y$, we have:

$$\Delta t \leq \frac{\Delta x}{a_x + a_y}$$

For the diffusive part of the Navier-Stokes equations, we study the two-dimensional diffusion equation, which in continuous form is given by:

$$\frac{\partial U}{\partial t} = \frac{1}{Re} \nabla^2 U$$

Discretising this using a central scheme in space and a forward Euler scheme in time, we obtain

$$\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\Delta t} = \frac{1}{Re} \left(\frac{U_{i+1,j}^n - 2U_{i,j}^n + U_{i-1,j}^n}{\Delta x^2} + \frac{U_{i,j+1}^n - 2U_{i,j}^n + U_{i,j-1}^n}{\Delta y^2} \right)$$

Then, to carry out von Neumann analysis, we make the ansatz $U_{i,j}^n = \hat{U}_{k\ell}^n e^{ikx_i} e^{i\ell y_j}$ and define the amplification factor as $\hat{G}(k,\ell) = \hat{U}_{k\ell}^{n+1} / \hat{U}_{k\ell}^n$. Substituting this into the discretisation gives:

$$\frac{\hat{U}_{k\ell}^{n+1} - \hat{U}_{k\ell}^n}{\Delta t} = \frac{1}{Re} \left(\frac{\hat{U}_{k\ell}^n e^{ik\Delta x} - 2\hat{U}_{k\ell}^n + \hat{U}_{k\ell}^n e^{-ik\Delta x}}{\Delta x^2} + \frac{\hat{U}_{k\ell}^n e^{i\ell\Delta y} - 2\hat{U}_{k\ell}^n + \hat{U} e^{-i\ell\Delta y}}{\Delta y^2} \right)$$

$$\frac{\hat{U}_{k\ell}^{n+1}}{\hat{U}_{k\ell}^n} = \hat{G}(k,\ell) = 1 + \frac{\Delta t}{Re} \left[\frac{1}{\Delta x^2} (2\cos k\Delta x - 2) + \frac{1}{\Delta y^2} (2\cos \ell\Delta y - 2) \right]$$

Defining:

$$\beta_x = \frac{1}{Re} \frac{\Delta t}{\Delta x^2} \quad \text{and} \quad \beta_y = \frac{1}{Re} \frac{\Delta t}{\Delta y^2}$$

we have the stability condition $\left|\hat{G}(k,\ell)\right| \leq 1$. Since $\left|\hat{G}(k,\ell)\right|$ is wholly real, this reduces to:

$$-1 \leq 1 + 2\beta_x (\cos k\Delta x - 1) + 2\beta_y (\cos \ell\Delta y - 1) \leq 1$$

$$0 \leq \beta_x (1 - \cos k\Delta x) + \beta_y (1 - \cos \ell\Delta y - 1) \leq 1$$

For the most stringent limit on β_x and β_y, we choose $k\Delta x = \ell\Delta y = \pi$. This gives us:

$$\beta_x + \beta_y \leq \frac{1}{2}$$

In this case, we have $\Delta x = \Delta y$, meaning that $\beta_x = \beta_y$. Hence, the diffusive stability limit on Δt is:

$$\beta = \frac{1}{Re} \frac{\Delta t}{\Delta x^2} \leq \frac{1}{4}$$

$$\implies \Delta t \leq \frac{Re \Delta x^2}{4}$$

It is clear that the diffusive stability limit is dependent on the Reynolds number and the grid spacing, while the CFL condition is only dependent on the grid spacing. Therefore, for lower Reynolds numbers (Cases A and B), the diffusive stability limit is more restrictive than the CFL condition, since Re is fairly small and Δx is very small. However, when the Reynolds number is large (such as

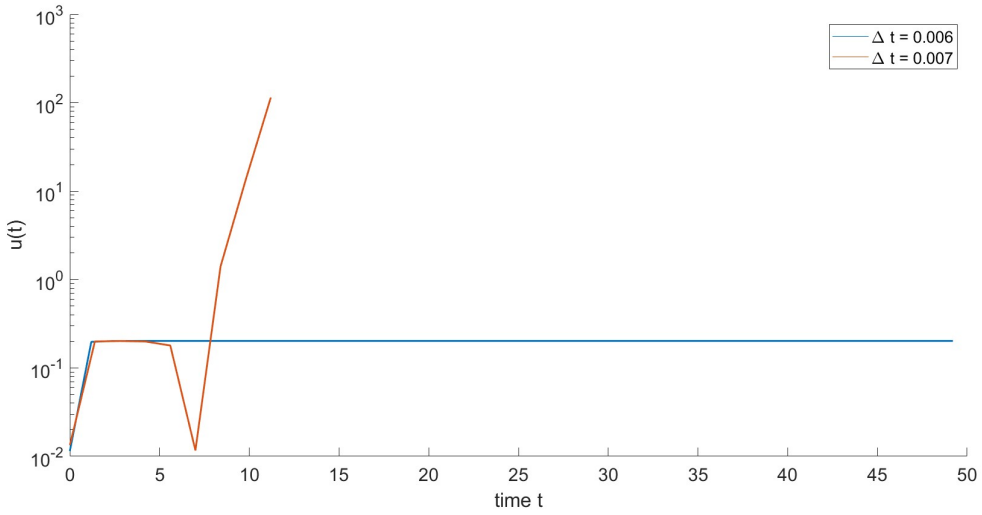

Figure 8: Time Histories for Case A, $\Delta t = 0.006$ and $\Delta t = 0.007$

in Case C), the diffusive limit becomes large, and (as long as the grid spacing is kept constant) the CFL condition becomes the more stringent limit.

Consider now Flow Case A, for which the Reynolds number is $Re = 25$ and the domain is discretised with $N_x = N_y = 30$. This gives grid sizes of $\Delta x = \Delta y = 0.0\dot{3}$, and we assume that the characteristic speeds in the x- and y-directions are $a_x = a_y = 1$. These can be substituted first into the CFL condition, which is:

$$\Delta t \leq \frac{\Delta x}{a_x + a_y} = \frac{0.0\dot{3}}{1+1} = 0.01\dot{6}$$

Then, the diffusive stability condition gives a similar maximum value for the time step:

$$\Delta t \leq \frac{Re\Delta x^2}{4} = \frac{25 \times 0.0\dot{3}^2}{4} = 6.9\dot{4} \times 10^{-3}$$

As we expect, the diffusive stability limit is the more restrictive condition in this case where the Reynolds number is low, giving us a lower value for Δt than the CFL condition. Hence, we expect that using a time step above $\Delta t = 6.9\dot{4} \times 10^{-3}$ will produce an unstable simulation, while time steps below this value will converge. This is backed up by the actual simulation results when we set $\Delta t = 0.006$ and $\Delta t = 0.007$, i.e. two values which are just below and just above the stability limit, respectively (Figure 8). It is clear that the time history for the simulation using $\Delta t = 0.006$ is stable and converges quite quickly to the expected behaviour (as in Section 2), while the velocity for $\Delta t = 0.007$ diverges considerably, and even fails to produce non-finite results after $t = 11.2$s. These time histories provide solid evidence not only that it is absolutely necessary to satisfy both the diffusive stability condition and the CFL condition, but also that the diffusive stability condition is more restrictive than the CFL condition, as expected.

It should be noted, though, that the scalar convection equation for the temperature constitutes a separate problem, and so may not have the same stability limit as the momentum equations. Motivated by the stability analysis we have carried out for the Navier-Stokes equations, we study the following diffusion equation, including both the Reynolds and Prandtl numbers:

$$\frac{\partial \theta}{\partial t} = \frac{1}{PrRe}\nabla^2\theta$$

Just like the momentum equation, we discretise this equation with a forward Euler scheme in time and a central scheme in space:

$$\frac{\theta_{i,j}^{n+1} - \theta_{i,j}^n}{\Delta t} = \frac{1}{PrRe}\left(\frac{\theta_{i+1,j}^n - 2\theta_{i,j}^n + \theta_{i-1,j}^n}{\Delta x^2} + \frac{\theta_{i,j+1}^n - 2\theta_{i,j}^n + \theta_{i,j-1}^n}{\Delta y^2}\right)$$

and this leads us to a nearly identical stability condition as we have for the momentum equations:

$$\beta_x + \beta_y \leq \frac{1}{2}$$

where

$$\beta_x = \frac{1}{PrRe}\frac{\Delta t}{\Delta x^2}, \quad \beta_y = \frac{1}{PrRe}\frac{\Delta t}{\Delta y^2}$$

For this flow case, where $\Delta x = \Delta y$ and $Pr = 0.71$, the maximum time step becomes

$$\Delta t \leq \frac{PrRe\Delta x^2}{4} = \frac{0.71 \times 25 \times 0.0\dot{3}^2}{4} = 4.930\dot{5} \times 10^{-3}$$

The inclusion of the Prandtl number (which here is less than 1) makes this condition stricter than the conditions on the momentum equation, meaning that one must be careful when considering coupled velocity-temperature problems. Hence, for this case, the scalar transport equation is less stable than the momentum equations.

Question 3

The boundary conditions for u on the top and bottom boundaries as well as for v on the left and right boundaries should be imposed by choosing the correct values for the ghost nodes. Write the expressions for $\underline{u}_{i+\frac{1}{2},0}$ and $\underline{u}_{0,j+\frac{1}{2}}$ at the ghost nodes. The values for the pressure at the ghost nodes $\underline{p}_{0,j}$ and $\underline{p}_{i,0}$ should also be chosen such that the Neumann boundary condition is satisfied. Write the expression for the pressure values on the ghost nodes and for the discretised Laplace operator $\underline{\underline{L}}\underline{p}_{1,j}$ for $j = 2, ..., N_y - 1$.

Since the ghost nodes are defined differently for the u- and v-components of the velocity, they must be separated and treated differently. In practice, this means that the boundary conditions cannot be applied in the same way to all of the boundaries for u or v, and that they must be applied differently for each velocity component. Starting with u, Figure X shows that there are nodes defined on the left- and right-hand boundaries, allowing the boundary conditions to be defined directly there. However, there are no nodes on the top and bottom walls, and we are given ghost nodes instead, meaning that the boundary condition needs to be applied by ensuring that the velocity interpolates to the correct value over the walls. For u, we have homogeneous Dirichlet conditions on the left and right walls, and non-homogeneous Dirichlet conditions on the top and bottom. This gives the following values for the boundary and ghost nodes:

$$\begin{aligned}
&\text{North:} && \tfrac{1}{2}\left(u_{i+\frac{1}{2},N_y} + u_{i+\frac{1}{2},N_y+1}\right) = u_{top} \implies u_{i+\frac{1}{2},N_y+1} = 2u_{top} - u_{i+\frac{1}{2},N_y} \\
&\text{South:} && \tfrac{1}{2}\left(u_{i+\frac{1}{2},0} + u_{i+\frac{1}{2},1}\right) = u_{bottom} \implies u_{i+\frac{1}{2},0} = 2u_{bottom} - u_{i+\frac{1}{2},1} \\
&\text{West:} && u_{\frac{1}{2},j} = 0 \\
&\text{East:} && u_{N_x+\frac{1}{2},j} = 0
\end{aligned}$$

For v, we have homogeneous Dirichlet conditions all around $\partial\Omega$. There are boundary nodes on the top and bottom walls, and ghost nodes on the left- and right-hand sides. Thus, we proceed in a similar way to the u boundary conditions:

$$\begin{aligned}
&\text{North:} && v_{i,N_y+\frac{1}{2}} = 0 \\
&\text{South:} && v_{i,\frac{1}{2}} = 0 \\
&\text{West:} && \tfrac{1}{2}\left(v_{0,j+\frac{1}{2}} + v_{1,j+\frac{1}{2}}\right) = 0 \implies v_{0,j+\frac{1}{2}} = -v_{1,j+\frac{1}{2}} \\
&\text{East:} && \tfrac{1}{2}\left(v_{N_x,j+\frac{1}{2}} + v_{N_x+1,j+\frac{1}{2}}\right) = 0 \implies v_{N_x+1,j+\frac{1}{2}} = -v_{N_x,j+\frac{1}{2}}
\end{aligned}$$

For the pressure, we define the ghost nodes as shown in Figure X, which lie outside the boundaries. We stipulate that the normal derivative of the pressure across each of the boundaries should vanish. That is, we impose homogeneous Neumann boundary conditions:

$$\frac{\partial p}{\partial n} = 0 \quad \text{on } \partial\Omega$$

Noting that there are no pressure nodes on the boundaries themselves, this allows us to define the pressure values at the ghost nodes as:

$$\begin{aligned}
&\text{North:} && \left(p_{i,N_y} - p_{i,N_y+1}\right)/dy = 0 \implies p_{i,N_y+1} = p_{i,N_y} \\
&\text{South:} && \left(p_{i,1} - p_{i,0}\right)/dy = 0 \implies p_{i,0} = p_{i,1} \\
&\text{West:} && \left(p_{1,j} - p_{0,j}\right)/dx = 0 \implies p_{0,j} = p_{1,j} \\
&\text{East:} && \left(p_{N_x,j} - p_{N_x+1,j}\right)/dx = 0 \implies p_{N_x+1,j} = p_{N_x,j}
\end{aligned}$$

In order to use the projection-correction method to solve the Navier-Stokes equations, we will need to numerically solve a Poisson equation for the pressure at each time step, which we accomplish by turning the PDE into a system of linear equations that can be written as a matrix. Hence, we need a Laplacian operator matrix $\underline{\underline{L}}$ which already includes the ghost-node values found above. The effect of the boundary conditions is thus pre-built into the Laplacian matrix, and to check this we can find the expression that this matrix gives at the left-hand boundary. In effect, then, we want to find $\left[\underline{\underline{L}}\underline{p}\right]_{1,j}$ for $j \in [2, N_y - 1]$, which we can do by using the general expression we found in Question 1:

$$\underline{\underline{L}}\underline{p}[i + (j-1)N_x] = \sum_{k=1}^{N_x} p_{k,j-1}\left[\frac{1}{\Delta y^2}\underline{\underline{I}}_{N_x \times N_x}(i,k)\right] + \sum_{k=1}^{N_x} p_{k,j}\left[-\frac{2}{\Delta y^2}\underline{\underline{I}}_{N_x \times N_x}(i,k) + \underline{\underline{D}}_{2x}(i,k)\right] + \sum_{k=1}^{N_x} p_{k,j+1}\left[\frac{1}{\Delta y^2}\underline{\underline{I}}_{N_x \times N_x}(i,k)\right]$$

Letting $i = 1$ and assuming that j is away from the boundary values, we have:

$$\left[\underline{\underline{L}}\underline{p}\right]_{1,j} = \sum_{k=1}^{N_x} \frac{1}{\Delta y^2} p_{k,j-1}\delta_{1k} + \sum_{k=1}^{N_x} p_{k,j}\left[-\frac{2}{\Delta y^2}\delta_{1k} + \underline{\underline{D}}_{2x}(1,k)\right] + \sum_{k=1}^{N_x} \frac{1}{\Delta y^2} p_{k,j+1}\delta_{1k}$$

$$\left[\underline{\underline{L}}\underline{p}\right]_{1,j} = \frac{1}{\Delta y^2} p_{1,j-1} - \frac{2}{\Delta y^2} p_{1,j} + \sum_{k=1}^{N_x} p_{k,j}\underline{\underline{D}}_{2x}(1,k) + \frac{1}{\Delta y^2} p_{1,j+1}$$

$$\left[\underline{\underline{L}}\underline{p}\right]_{1,j} = \frac{p_{2,1} - p_{1,1}}{\Delta x^2} + \frac{p_{1,j+1} - 2p_{1,j} + p_{1,j-1}}{\Delta y^2}$$

Looking at the ghost node formulations for the pressure, this is precisely what we expect at a general point on the left-hand boundary of the flow domain.

Question 4

Place a probe at the domain centre and compare the time history of the horizontal velocity (u) for flow cases A–C. Try to estimate the time it takes for each run to establish a steady solution. How does this time correlate with the Reynolds number?

For Case A, where the Reynolds number was set to $Re = 25$, the numerical solution converged very quickly to a steady solution. As shown in Figure 9, the horizontal velocity at the centre of the domain ($x = 0.5$, $y = 0.5$) remains constant over the entire simulation from approximately $t = 2$s, showing that the solution is steady after only a very short simulation.

For Case B (Figure 10), the convergence only happens after a longer time period in the simulation. This case, at $Re = 250$, converged to a steady solution only after approximately $t = 22$s.

For Case C, with a Reynolds number of $Re = 5000$, the simulation is no longer steady at $t = 50$s, as can be seen from the time history in Figure 11. Since this is a considerably longer time than the convergence periods needed for the two previous simulations, it was necessary to check that the stability conditions were still satisfied for a time step of $\Delta t = 0.001$s. Checking the CFL condition with $Re = 5000$ and $\Delta x = \Delta y = 0.01$:

$$\Delta t \leq \frac{0.01}{2} = 0.005$$

Checking the diffusive stability condition:

$$\Delta t \leq \frac{5000 \times 0.01^2}{4} = 0.125$$

Since both of these conditions are still fulfilled, the unsteadiness at $t = 50$ is a genuine artifact of the simulation, and the convergence time for large Reynolds numbers does indeed increase significantly. If we wait for the simulation to converge, we only get steady flow after approximately 250 seconds as Figure 12 shows. Thus, the initial unsteady flow will eventually die away, even though this may take a very long time.

Hence, it is clear that the convergence time goes up as the Reynolds number is increased. A larger Reynolds number corresponds to lower fluid viscosity, which means that there is less diffusion of momentum throughout the fluid. Hence, it will take more time for the local accelerations at each point to decay, and hence more time is needed for the flow to stabilise.

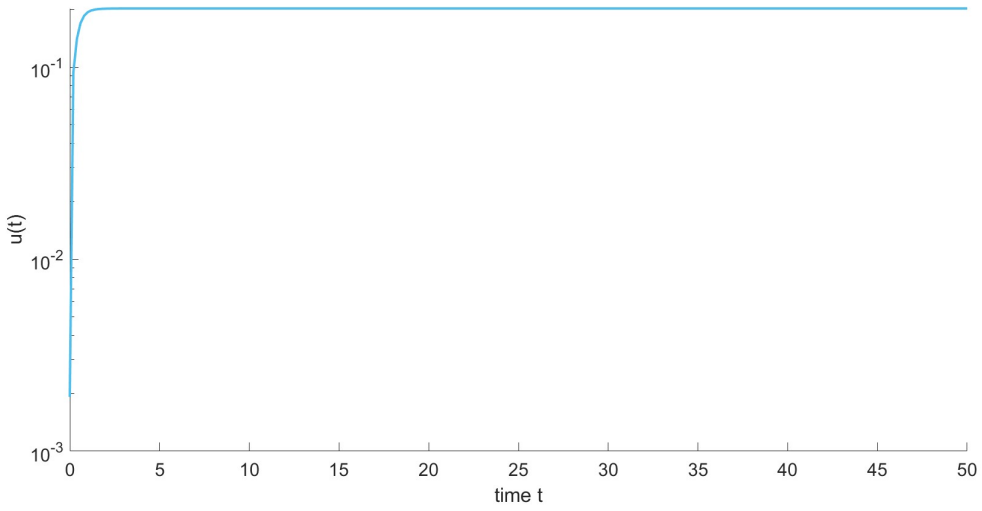

Figure 9: Time History for Case A ($Re = 25$)

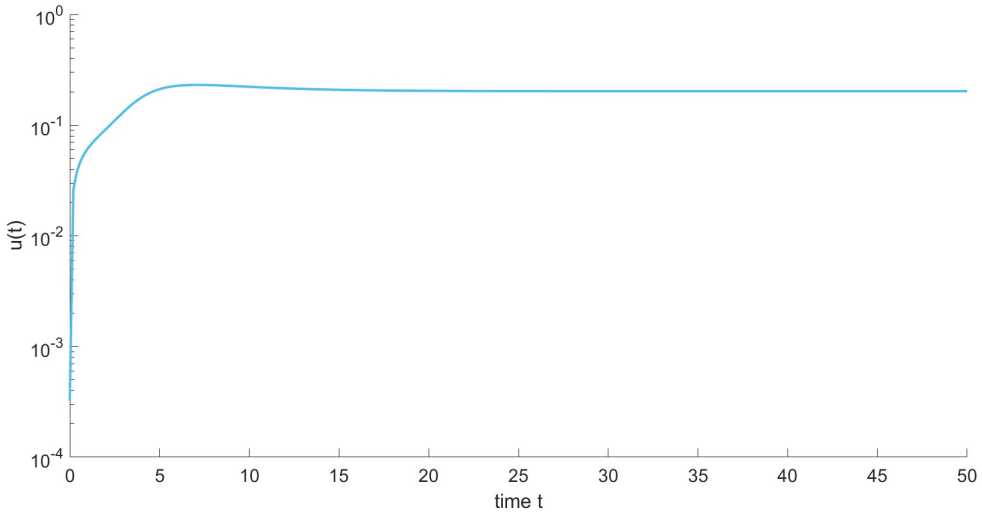

Figure 10: Time History for Case B ($Re = 250$)

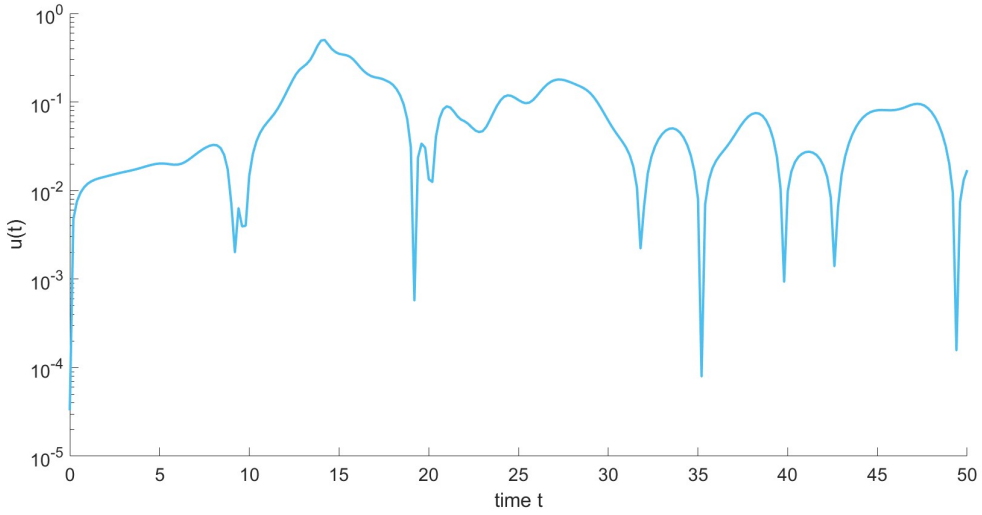

Figure 11: Time History for Case C ($Re = 5000$) until $t = 50$s

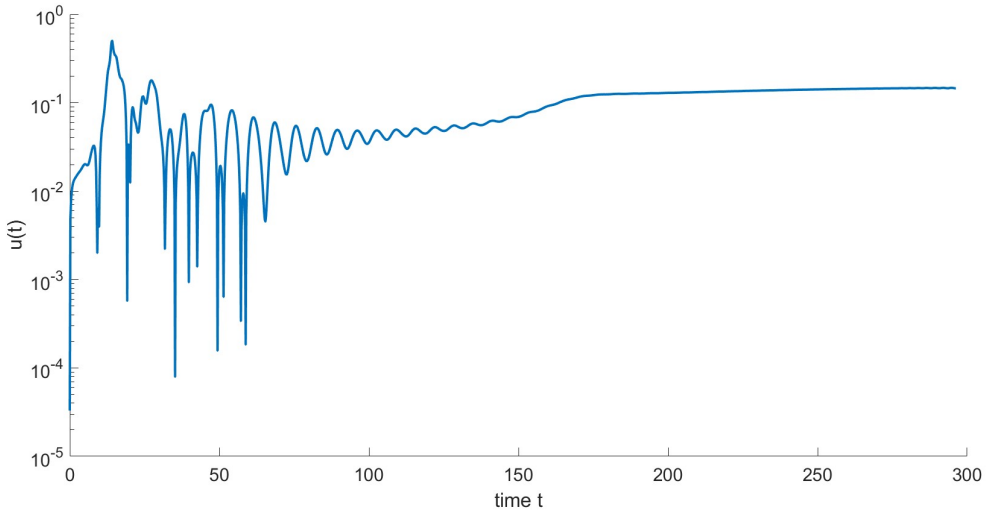

Figure 12: Converged Time History for Case C ($Re = 5000$) until $t = 300$s

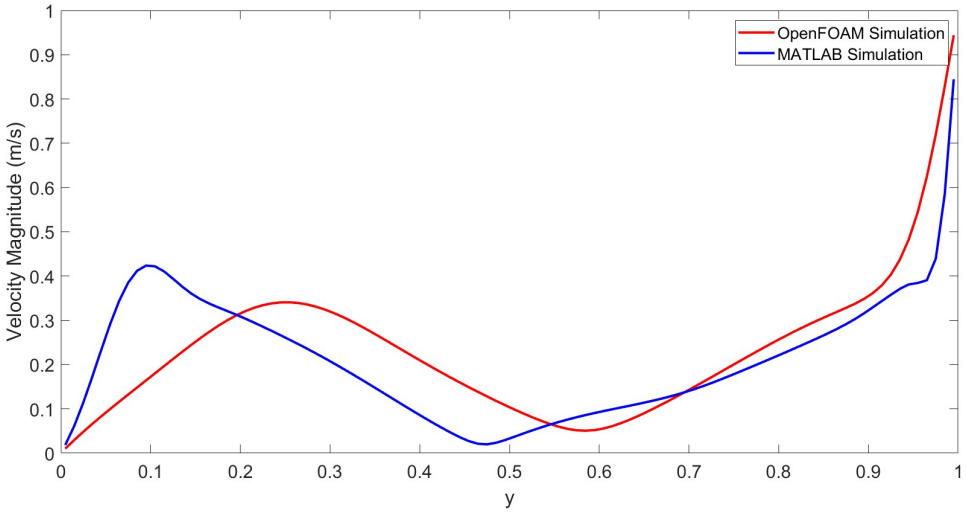

Figure 13: Velocity Magnitude along the Vertical Mid-Line for Cases C and D

Question 5

Compare the velocity profile along the centre plane that you obtained using OpenFOAM with what you obtain using your MATLAB code.

Since we have run Case C twice, both on the developed MATLAB code and in OpenFOAM, we may compare the two sets of results to see how far they agree. In particular, rather than simply looking at the contour maps of the velocity fields, we may consider the velocity distribution over a pre-defined line of interest, which in this case we choose to be the vertical mid-line of the domain (i.e. $x = 0.5$). At $t = 50$s, the velocity profiles along this line are shown in Figure 13. Here, the vortex core (the region of decelerated flow around which the rest of the fluid rotates) is clearly visible in both simulations, being represented by the local minima in these plots. However, the two simulations do differ: the vortex core and local maximum velocity occur at slightly lower y-valuesin the MATLAB results than in the OpenFOAM results, and MATLAB predicts a higher local maximum velocity than OpenFOAM does. However, in terms of boundary conditions, both simulations tend to zero velocity at the fixed bottom wall and to $V = 1$ at the moving top wall, which is what we require physically. Additionally, the two profiles have the same overall shape, indicating that while the specifics of the flow field may be subject to some inaccuracy in the MATLAB program, the large-scale flow features are modelled reliably.

Question 6

In your MATLAB code, implement a moving lower wall using appropriate boundary conditions. Try at least one case at $Re = 100$ with the lower wall moving with the same velocity as the upper one. Include the plot of the final velocity field, and discuss the symmetry properties of the resulting flow.

By implementing another moving wall at the bottom of the domain, we change the flow from the already-discussed cases. The no-slip condition must still be satisfied on the bottom wall, but this now means that the horizontal velocity there must be equal to the wall velocity (which we choose here to be equal to the top wall velocity). The vertical velocity v on the bottom wall must still be zero, as we expect from the zero-normal-flow boundary condition. Figure 14 shows the velocity and temperature fields for a lid-driven cavity flow at $Re = 100$ with $u_{top} = u_{bottom} = 1$, at time $t = 50$s. Here, the velocity field is now symmetrical about the horizontal mid-line, and there are accordingly two vortex cores about which the local flow rotates. Along the horizontal mid-line, the velocity vectors are all parallel to the top and bottom walls, and there are two stagnation points along this line at the intersections with the left and right walls. Because the Reynolds number is moderately high, the flow is no longer symmetrical about the vertical mid-line, with noticeable areas of higher and lower velocity at the right- and left-hand corners, respectively. As for the temperature, the boundary and initial settings were kept as in previous simulations: $\theta_{bottom} = 1$, $\theta_{top} = 0$. It is clear that, like the velocity field, the lines of constant temperature (isotherms) are symmetrical about the horizontal mid-line.

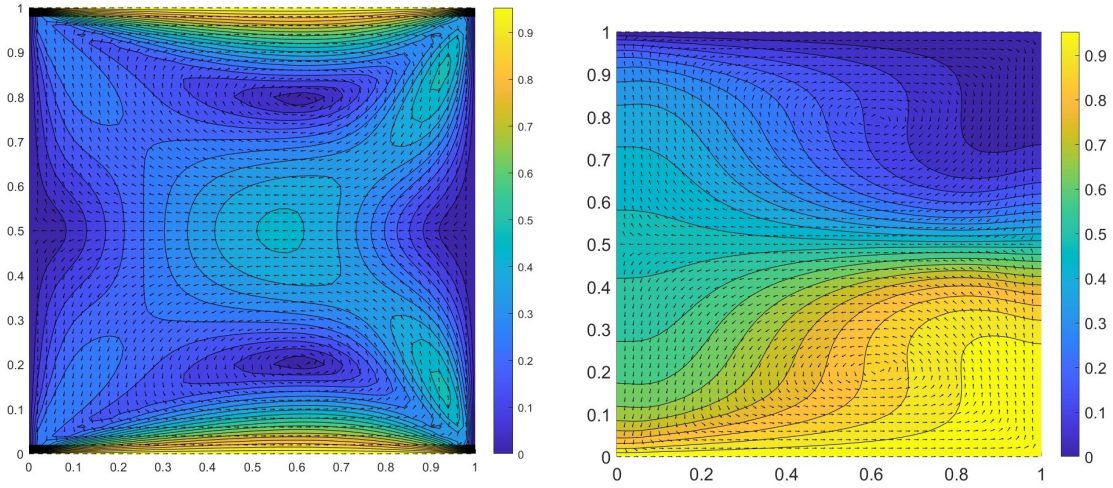

Figure 14: Velocity (left) and Temperature (right) Fields for $Re = 100$ with $u_{bottom} = 1.0$