PROJECT - 3
AERO 523

Name: Faiz Fateh
UMID: 44561358

UNIVERSITY OF MICHIGAN

1. INTRODUCTION

In this project we will be solving a 2-dimensional led-driven cavity problem using incompressible Navier-Stokes equations using primitive variable (u, v, p). The geometry of the problem is shown in Figure 1. The length of the square cavity is 1 and the bottom, left and right walls are fixed. The top wall moves with a velocity of U_{wall} is set at 1.

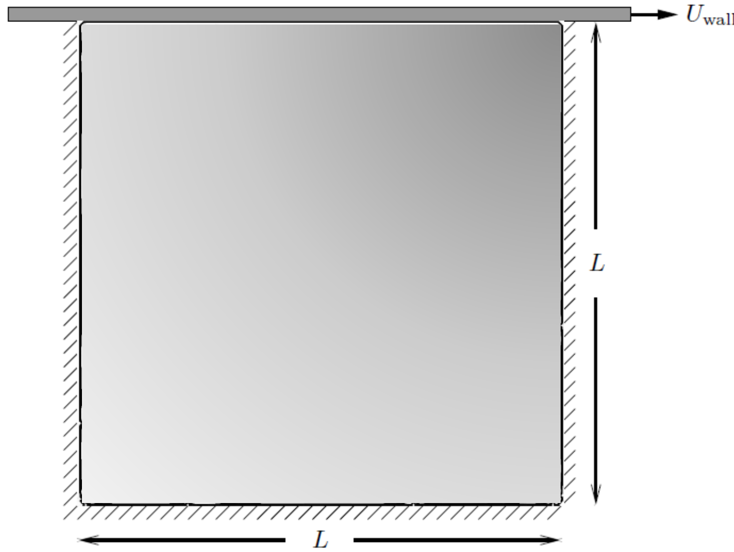


Figure 1: Setup for the led-driven cavity problem

1.1 Physical Model

To solve this problem we will be implementing the projection method. Basic steps involved in this method are:

- Compute an intermediate velocity field $\vec{v}^{n+\frac{1}{2}}$, from $\vec{v}^{n}$ using the momentum equations.
- Solve pressure Poisson equation:

$$\nabla^2 p^{n+1} = \rho \frac{\nabla \cdot \vec{v}^{n+\frac{1}{2}}}{\Delta t}$$

- Correct Velocity field using :

$$\rho \frac{\vec{v}^{n+1} - \vec{v}^{n+\frac{1}{2}}}{\Delta t} + \nabla p^{n+1} = 0$$

2. PROGRAMMING THE SOLVER

2.1 Variable Storage

We use a staggered storage to reduce the spurious oscillations while discretizing the primitive variable (u, v, p) for the calculations. Figure 2 shows a schematic representation of our storage method.

In our staggered grid:

- u (x-velocity) is stored on the midpoints of vertical edges.
- v (y-velocity) is stored on the midpoints of horizontal edges.
- p, F, G (pressure, x-direction flux, y-direction flux) stored on the cell centers.
- H_x , H_y stored on the grid nodes.

The corresponding matrices sizes of the variables will be as follows;

- u (x-velocity) is stored in (N+2) X (N+3) array includes ghost cells.
- v (y-velocity) is stored in (N+3) X (N+2) array includes ghost cells.
- p, F, G (pressure, x-direction flux, y-direction flux) stored in N X N array.
- H_x , H_y stored in (N+1) X (N+1).

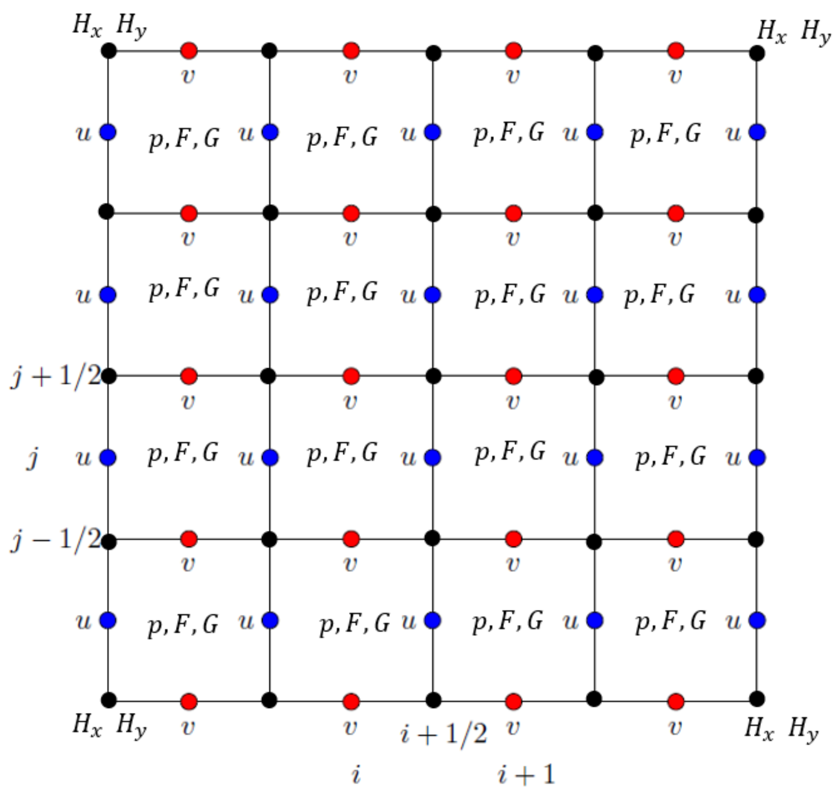


Figure 2: Schematic representation of the staggered store for various variable

2.2 Ghost cells

To calculate the p, F, g, H^x, H^y on the boundary of our container we need extra values of u, v which we can obtain using the concept of ghost cells. We assume an extra layer of cells surrounding the given grid and find the values of u, v on these extra cells using the boundary conditions provided. The boundary conditions are there is no slip conditions and no flow thru across all the boundaries and only the top wall has a velocity of 1in the x-direction. Using these conditions, we can derive equations to relate the ghost cells to the interior cells such that:

For left and right boundaries:

$$u_{out} = u_{in}$$
$$v_{out} = -v_{in}$$

For bottom boundary:

$$u_{out} = -u_{in}$$
$$v_{out} = v_{in}$$

For top boundary:

$$u_{out} = 2 * U_{wall} - u_{in}$$
$$v_{out} = v_{in}$$

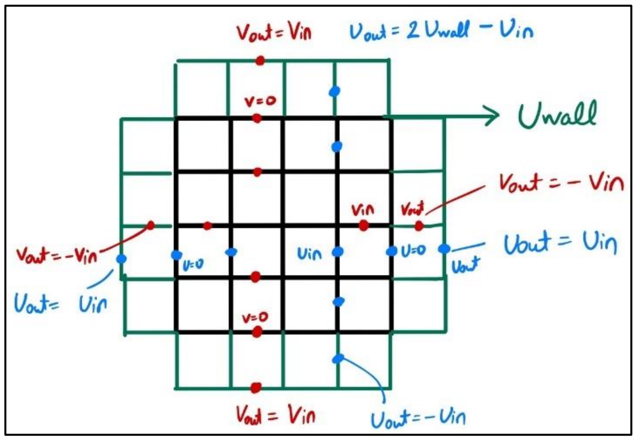


Figure 3: Schematic showing the ghost cells and their implementations

2.3 Conservation Equations

The 2-D Navier-Stokes equation can be written in conservation form such that,

$$u_t + (F + p)_x + H_y^x = 0$$
$$v_t + H_y^x + (G + p)_y = 0$$

Here,

$$F = u^2 - \nu u_x \quad G = v^2 - \nu v_y \quad H^x = uv - \nu u_y \quad H^y = vu - \nu v_x$$

These fluxes (F, G) are stored at cell centers while (H^x, H^y) are stored at the grid nodes. To calculate the fluxes we take a closer look at the meaning of the terms which reveals that :

F : uu = transport of u in the x-direction

G: vv = transport of v in the y-direction

H^x : uv = transport of u in the y-direction

H^y : vu = transport of v in the x-direction

2.4 SMART Limiter

To evaluate the flux, we use upwinding but with a limiter to reduce the oscillations. All these fluxes are in the form: flux = $q\phi$

where, q: transport velocity calculated using

$$q_{i,j} = (u_{i-\frac{1}{2},j} + u_{i+\frac{1}{2},j})/2 \text{ for x-direction}$$

$$q_{i,j} = (v_{i,j-\frac{1}{2}} + v_{i,j+\frac{1}{2}})/2 \text{ for y-direction}$$

ϕ: transported quantity

For using SMART, we first need to define a smoothness monitor, $\hat{\phi}$ which takes in three consecutive data values : $\phi_{j-1}, \phi_j, \phi_{j+1}$ and is defined as:

$$\hat{\phi} = \frac{\phi_j - \phi_{j-1}}{\phi_{j+1} - \phi_{j-1}}$$

These 3 consecutive values depend upon the sign of q. if q>0 the we take 3 upwind values else if q<0 then we take 3 downwind values.

Then using the SMART limiter equalities we calculate $\hat{\phi}_{j+\frac{1}{2}}$

$$\hat{\phi}_{j+\frac{1}{2}} = \begin{cases} \hat{\phi}_j & \hat{\phi}_j < 0 \quad \text{or} \quad \hat{\phi}_j > 1 \\ 3\hat{\phi}_j & 0 < \hat{\phi}_j < \frac{1}{6} \\ \frac{3}{8}(2\hat{\phi}_j + 1) & \frac{1}{6} < \hat{\phi}_j < \frac{5}{6} \\ 1 & \frac{5}{6} < \hat{\phi}_j < 1 \end{cases}$$

After calculating the $\hat{\phi}_{j+\frac{1}{2}}$ we move ahead to calculate the final $\phi_{j+\frac{1}{2}}$ using

$$\hat{\phi}_{j+\frac{1}{2}} = \frac{\phi_{j+\frac{1}{2}} - \phi_{j-1}}{\phi_{j+1} - \phi_{j-1}}$$

2.5 Updating half velocities

With our fluxes calculated we go ahead and re-calculate our velocity without considering the pressure term and using Forward Euler time discretization. The equations for the update is as follows:

$$\begin{aligned} u^{n+\frac{1}{2}}_{i+\frac{1}{2},j} &= u^n_{i+\frac{1}{2},j} - \Delta t \left(F_x|_{i+\frac{1}{2},j} + H^x_y|_{i+\frac{1}{2},j} \right) \\ v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} &= v^n_{i,j+\frac{1}{2}} - \Delta t \left(H^y_x|_{i,j+\frac{1}{2}} + G_y|_{i,j+\frac{1}{2}} \right) \end{aligned}$$

Here, the fluxes are $F \text{ and } H^x_y$ are evaluated at $\left(i+\frac{1}{2},j\right)$ and $G \text{ and } H^y_x$ are evaluated at $\left(i,j+\frac{1}{2}\right)$. Using central difference, we get:

$$\begin{aligned} F_x|_{i+\frac{1}{2},j} &= \frac{1}{h}(F_{i+1,j} - F_{i,j}) \\ H^x_y|_{i+\frac{1}{2},j} &= \frac{1}{h}(H^x_{i+\frac{1}{2},j+\frac{1}{2}} - H^x_{i+\frac{1}{2},j-\frac{1}{2}}) \\ G_y|_{i,j+\frac{1}{2}} &= \frac{1}{h}(G_{i,j+1} - G_{i,j}) \\ H^y_x|_{i,j+\frac{1}{2}} &= \frac{1}{h}(H^y_{i+\frac{1}{2},j+\frac{1}{2}} - H^y_{i-\frac{1}{2},j+\frac{1}{2}}) \end{aligned}$$

Also Δt can be calculated using the formula provided in the problem statement as:

$$\Delta t = \beta \min \left(\frac{h^2}{4\nu}, \frac{4\nu}{U_{wall}^2}\right)$$

here, β is a stability factor

$$\nu = \frac{U_{wall} L}{Re}$$

2.6 Pressure Poisson

The pressure correction term is used to make velocity field at n+1 divergence free. Using central differencing to discretize the pressure equation we arrive at the following equations:

$$\delta_x^2|_{i,j} p^{n+1} + \delta_y^2|_{i,j} p^{n+1} = \frac{h}{\Delta t}\left(\delta_x|_{i,j} u^{n+\frac{1}{2}} + \delta_y|_{i,j} v^{n+\frac{1}{2}}\right)$$

Here,

$$\delta_x^2|_{i,j} p^{n+1} = p_{i-1,j}^{n+1} - 2p_{i,j}^{n+1} + p_{i+1,j}^{n+1}$$

$$\delta_y^2|_{i,j} p^{n+1} = p_{i,j-1}^{n+1} - 2p_{i,j}^{n+1} + p_{i,j+1}^{n+1}$$

$$\delta_x|_{i,j} u^{n+\frac{1}{2}} = u_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - u_{i-\frac{1}{2},j}^{n+\frac{1}{2}}$$

$$\delta_y|_{i,j} v^{n+\frac{1}{2}} = v_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - v_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}$$

We used the **Gauss-Seidel** iterative method with successive over relaxation (SOR) to solve the above-mentioned equations. The Gauss-Seidel was run for 30 iteration to minimize the error residual created while calculating the pressure at the cell centers.

Our iterative equation for Gauss-Seidel looks like:

$$p_{i,j}^{n+1} = \frac{\omega}{4}\left(\left(p_{i+1,j}^n + p_{i+1,j}^n + p_{i,j+1}^n + p_{i,j-1}^n\right) - \frac{h}{\Delta t}\left(\delta_x|_{i,j} u^{n+\frac{1}{2}} + \delta_y|_{i,j} v^{n+\frac{1}{2}}\right)\right) + (1-\omega) p_{i,j}^n$$

Also, at the corners and the boundaries the pressure beyond the wall is equal to the pressure inside the wall due to Neumann boundary conditions.

2.7 Correcting the velocity field

After updating our velocity field at n+1/2 and calculating the pressure field we move towards calculating the velocity at time step n+1. Pressure is added to make the velocity term divergence free. The update formula is given by:

$$\frac{\vec{v}^{n+1} - \vec{v}^{n+\frac{1}{2}}}{\Delta t} + \nabla p^{n+1} = 0$$

Which in discretized form can be written as:

$$
\begin{aligned}
u_{i+\frac{1}{2},j}^{n+1} &= u_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - \frac{\Delta t}{h}(p_{i+1,j} - p_{i,j}) \\
v_{i,j+\frac{1}{2}}^{n+1} &= v_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t}{h}(p_{i,j+1} - p_{i,j})
\end{aligned}
$$

2.8 Residual Calculation

When converging to steady state, we would like to monitor the value of Residual which could tell us how accurate our solver is. When using staggered storage, two types of control volumes are most convenient for defining the flux residual: the two shaded regions in Figure 4. We then have two residuals on our hands:

$$
\begin{aligned}
R_{i+\frac{1}{2},j} &= \int_{\partial A_{i+\frac{1}{2},j}} \vec{\mathbf{F}} \cdot \vec{n}\, dl \\
&= h(F_{i+1,j} + p_{i+1,j} - F_{i,j} - p_{i,j}) + h(H^x_{i+\frac{1}{2},j+\frac{1}{2}} - H^x_{i+\frac{1}{2},j-\frac{1}{2}})
\end{aligned}
$$

$$
\begin{aligned}
R_{i,j+\frac{1}{2}} &= \int_{\partial A_{i,j+\frac{1}{2}}} \vec{\mathbf{F}} \cdot \vec{n}\, dl \\
&= h(G_{i,j+1} + p_{i,j+1} - G_{i,j} - p_{i,j}) + h(H^y_{i+\frac{1}{2},j+\frac{1}{2}} - H^y_{i-\frac{1}{2},j+\frac{1}{2}})
\end{aligned}
$$

here, $R_{i+\frac{1}{2},j}$ is associated with vertical edge and $R_{i,j+\frac{1}{2}}$ is associated with horizontal edge. To obtain a single L_1 Residual norm we sum all the above calculated residual's absolute values via :

$$
|R|_{L_1} = \sum_{i=1}^{N-1}\sum_{j=1}^{N} |R_{i+\frac{1}{2},j}| + \sum_{i=1}^{N}\sum_{j=1}^{N-1} |R_{i,j+\frac{1}{2}}|
$$

2.9 Algorithm

To understand this more intuitively, a concise algorithm is created outlining all the necessary steps taken and further explaining the code structure in a simpler and easy to understand way.

Algorithm : Incompressible Navier-Stokes equations for lid-driven cavity problem

1. Initialize u,v,p to zeroes.
2. Define important parameters such as h, Δt, U_{wall} etc.
3. While Residual > 10^{-5}
4. Compute the ghost cells u & v values using the boundary conditions.
5. Compute the F, G, H_y^x, H_x^y using the SMART limiter
6. Update the half velocity $u_{i+\frac{1}{2},j}^{n+\frac{1}{2}}$ and $v_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}$
7. Calculate the Pressure Poisson using Gauss-Seidel
8. Using the pressure field correct the velocity to get $u_{i+\frac{1}{2},j}^{n+1}$ and $v_{i,j+\frac{1}{2}}^{n+1}$
9. Initialize Residual R_x and R_y

10. Calculate Residuals
11. Calculate the L_1 residual norm $|R|_{L_1}$
12. End While

3. RESULTS AND DISCUSSIONS

3.1 Convergence

Figure 4 and 5 shows the plot of L_1 residual norm vs the number of iterations it took to achieve that. Here we have used different Reynolds number for the two plots keeping everything else the same.

For Figure 4:

$$N=32, \quad Re=100, \quad U_{Wall}=1, \quad \beta=0.5, \quad \omega=1$$

And the graph is plotted using a semi-log on the y-axis

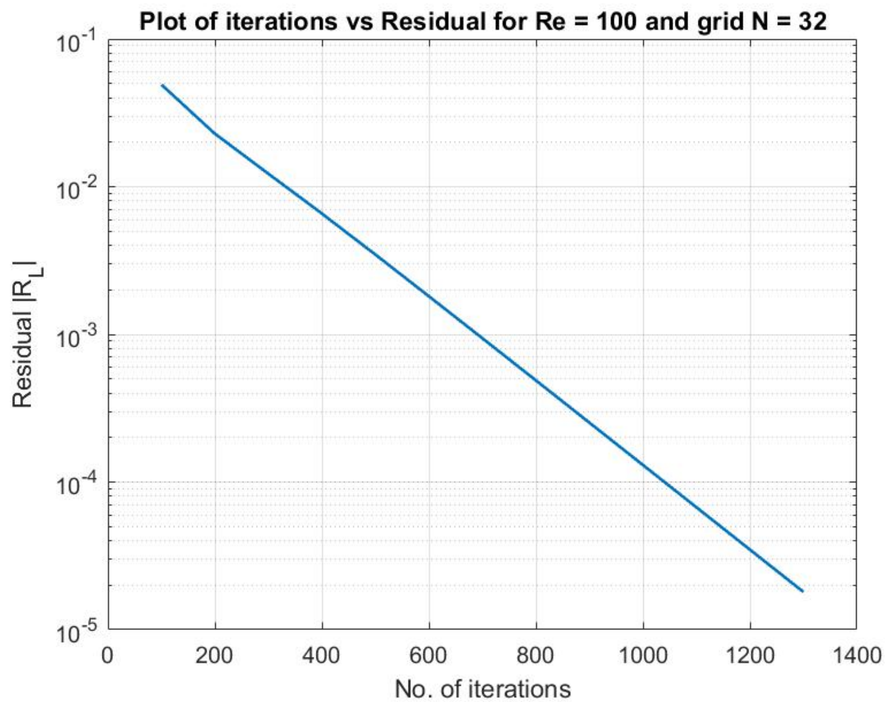


Figure 4: Plot of $|R|_{L1}$ vs iteration number for Re=100 and N=32

The next plot uses:

$$N=32, \quad Re=400, \quad U_{Wall}=1, \quad \beta=0.5, \quad \omega=1$$

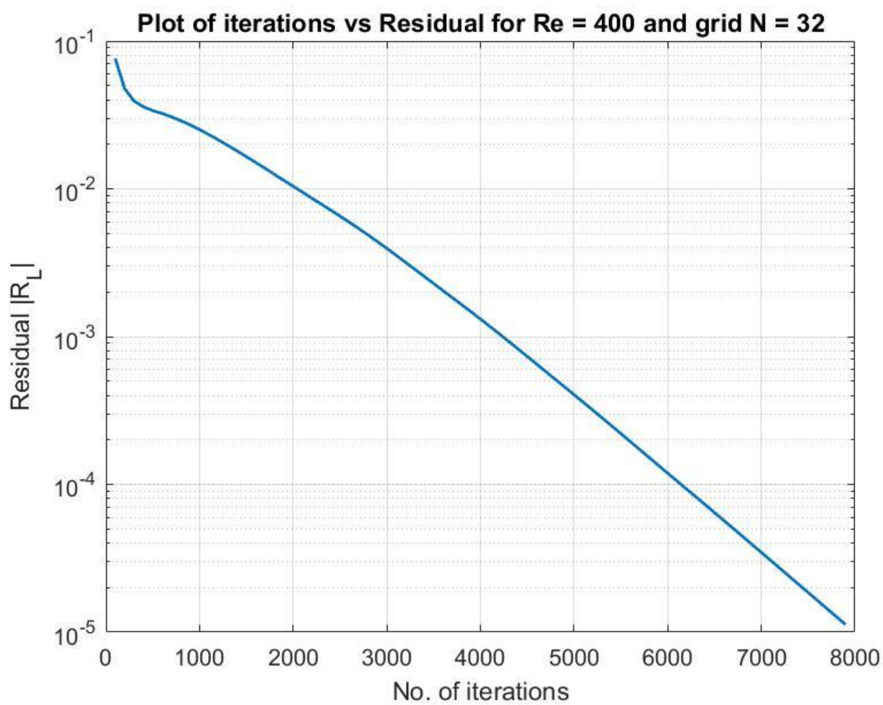


Figure 5: Plot of $|R|_{L1}$ vs iteration number for Re=400 and N=32

As we can see from the above two graphs as the Reynolds number increase the number of iterations it takes for the solution to converge below 10^{-5} increases. For Re=100 the plot converges after 1390 iterations while it takes 7994 iteration for Re-400 to converge. This can be attributed to the fact that as Reynolds number increase the fluid becomes more chaotic and the flow becomes harder to estimate causing the residuals to increase and hence increases the time for convergence.

3.2 Contour Plots

To create a contour plot we first need to define a streamfunction, ψ. With the streamfunction we can easily visualize the streamlines. ψ can be defined as:

$$\frac{\partial \psi}{\partial y} = u \,, \frac{\partial \psi}{\partial x} = -v$$

Integrating the streamfunction on a path A to B yields:

$$\begin{aligned} \psi(B) - \psi(A) &= \int_A^B \nabla\psi \cdot \vec{ds} \\ &= \int_A^B \vec{v} \cdot \vec{n} ds \end{aligned}$$

We can calculate the streamfunction at the nodes of our grid. First, pick a corner and put $\psi = 0$ there. From here we move horizontally to the next point and equate it like $\psi_{i+1,j} = \psi_{i,j} - v_{i+\frac{1}{2},j}$ and after completing one row we move up and while

moving up we use the equation: $\psi_{i,j+1} = \psi_{i,j} + u_{i,j+\frac{1}{2}}$. Due to no slip and no flow thru boundary conditions we can equate $\psi = 0$ at the walls.
Figure 6 and 7 shows the contour plots plotted for Re=100 and Re=400 respectively for a grid size of N=32. The contours are shown for the levels mentioned in the Ghia et al [1] Table III (Values for streamline and vorticity).
While comparing the contours with the plots shown in the Ghia et al papers we can easily identify the roughness and low accuracy of our contours which is caused due to a relatively smaller grid size (ours N=32 to Ghia N=129) which is inept in capturing all the minute details of the flow. Another aspect that reduces the accuracy of our contours can be the methods used to make the solver. We used explicit velocity updating and SOR Gauss-seidel while Ghia uses a combination of implicit methods and multi-grid techniques which improves the quality of the results.

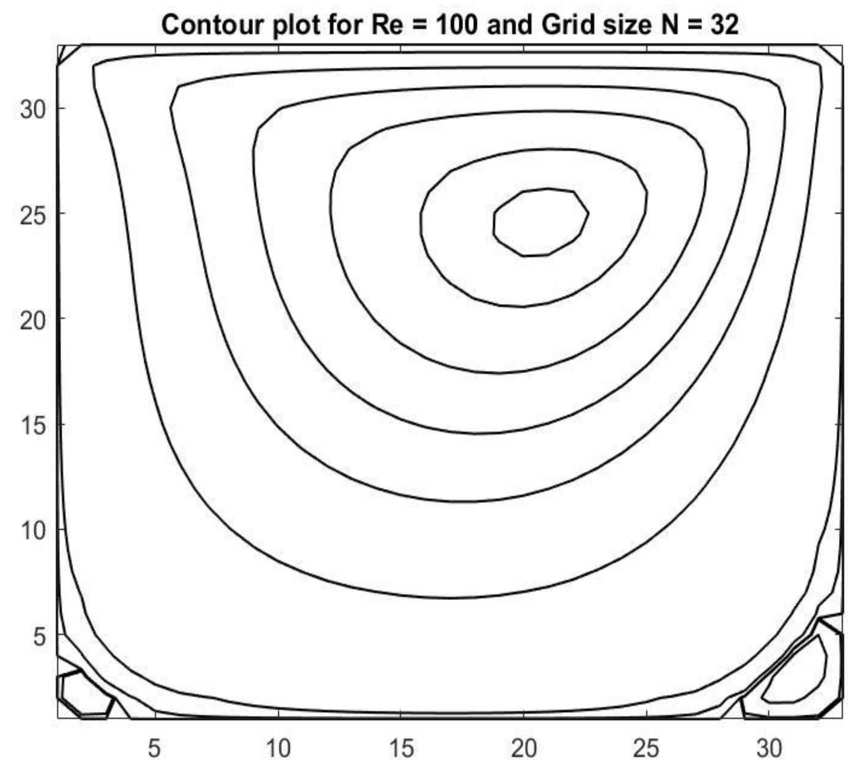


Figure 6: Contours of streamfunction for Re=100 and N=32

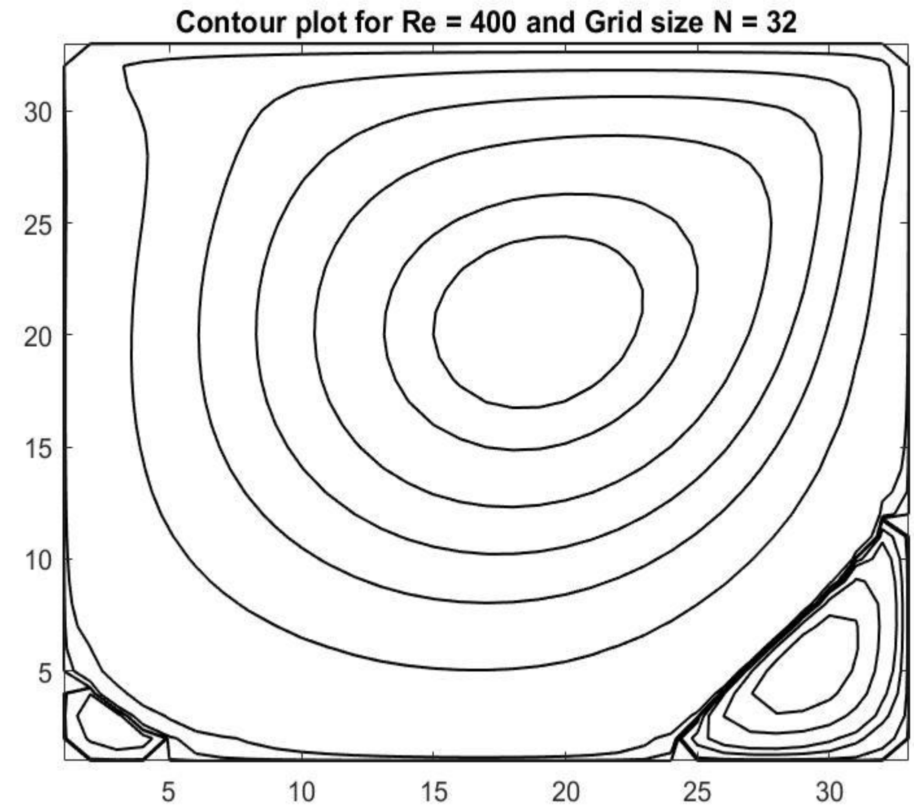


Figure 7: Contours of streamfunction for Re=400 and N=32

3.3 Velocity Profiles

We are asked to plot u-velocity along the vertical line through the geometric center and v-velocity along the horizontal line through the geometric center of our grid and overlay values corresponding to the same Reynolds number from Ghia et al paper Table I and II.

From Figure 8 and 9 we can see that the u and v – velocities through the geometric center as solved by us follows the same trend as seen with Ghia et al paper values but our plots couldn’t hit the maximum nor the minimum value mentioned in the Ghia et al papers. This could be because we are running our solver at a lower resolution, grid size of 32 as compared to his grid size of 129. Another possible reason for un-similarity could be the different methodologies used for the solver i.e. we are using a explicit solver using Gauss-Seidel to calculate Pressure Poisson and Ghia et al uses an implicit method with multi-grid optimization to reach at the results. As we increase our grid size we get closer to the values Ghia et al presents in his paper.

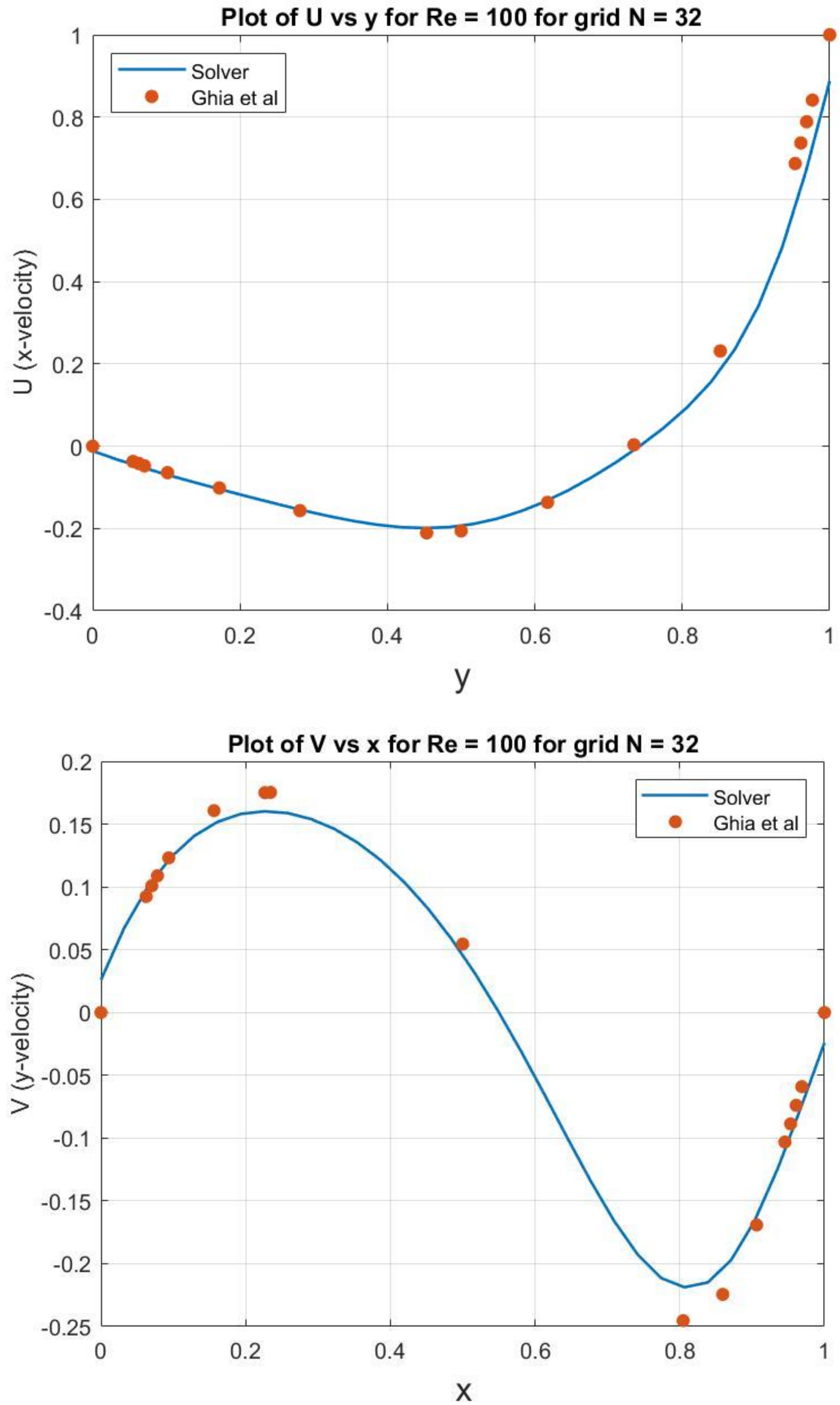


Figure 8: Plots of a) u-velocity vs y b) v-velocity vs x through the geometric center as well as values from Ghia et al for Re=100

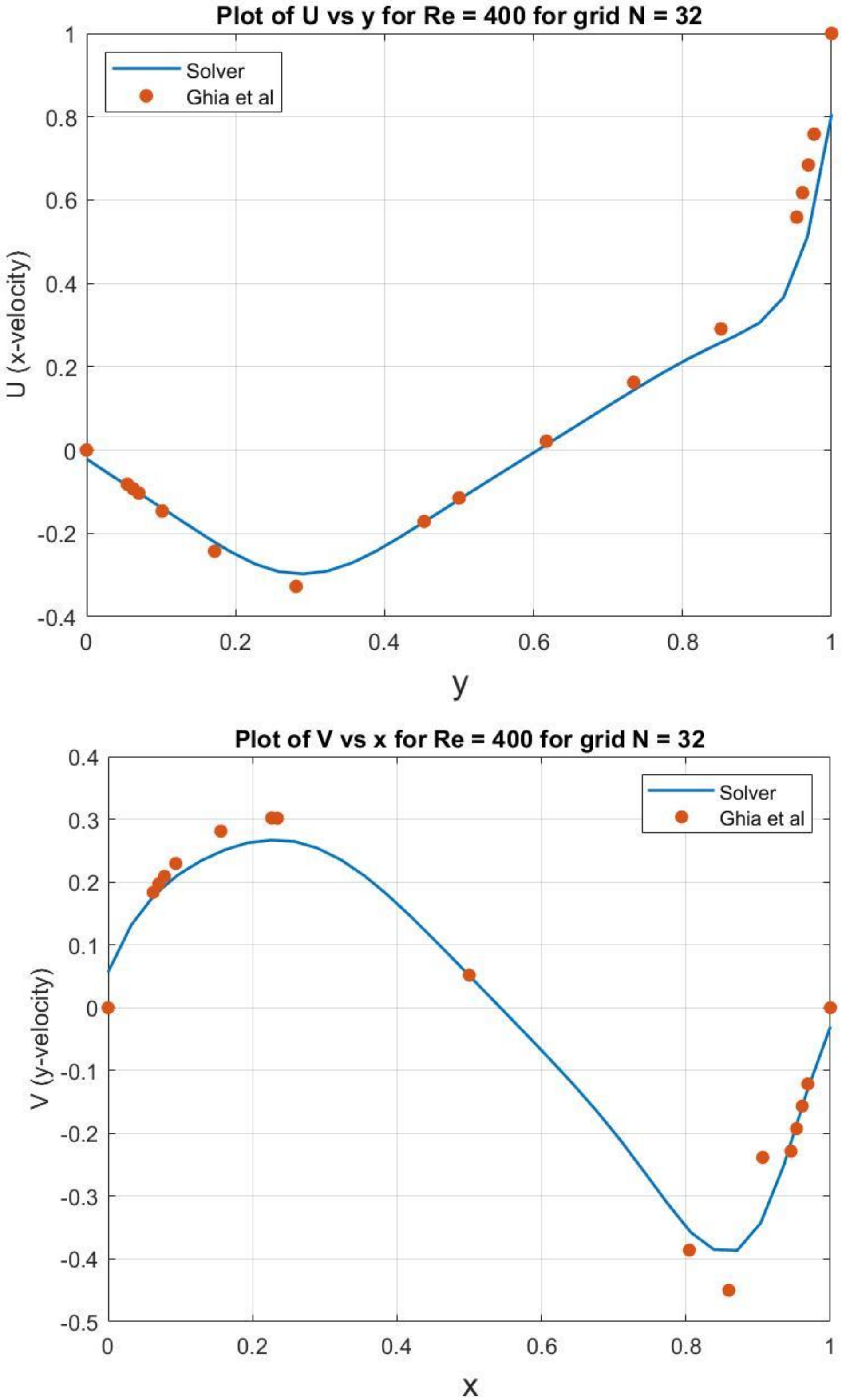


Figure 9: Plots of a) u-velocity vs y b) v-velocity vs x through the geometric center as well as values from Ghia et al for Re=400

3.4 Results for Higher Grid sizes and Higher Reynolds number

3.4.1 Grid Size 64 and Reynolds No. 1000

In this grid size our input variable changes. We can run at significantly high ω and β values to make our solution converge faster.

Here, I've used: N=64, Re=1000, ω=1.5, β=3

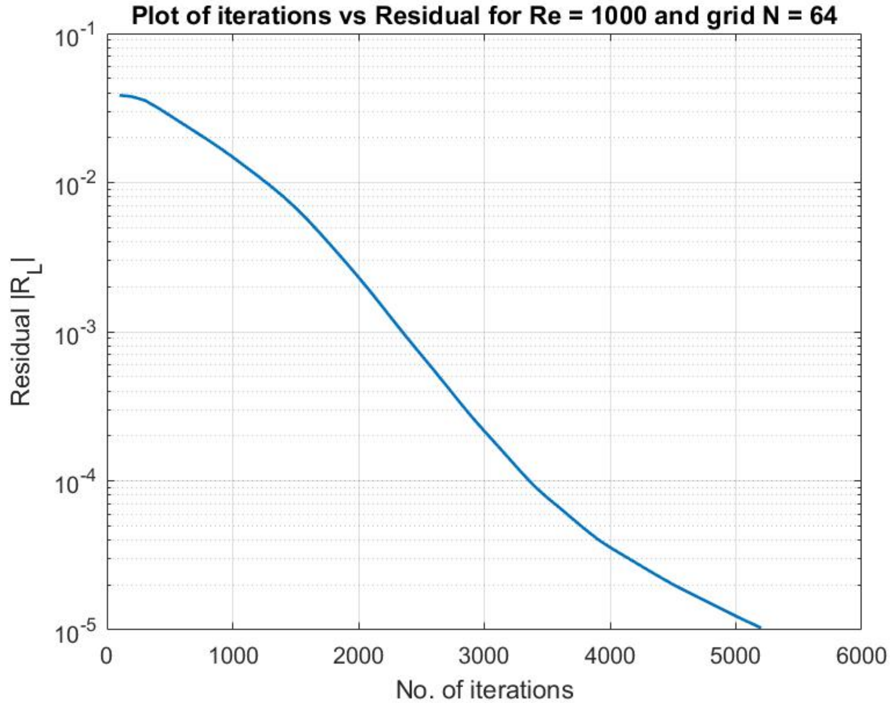


Figure 10: Plot of $|R|_{L1}$ vs iteration number for Re=1000 and N=64

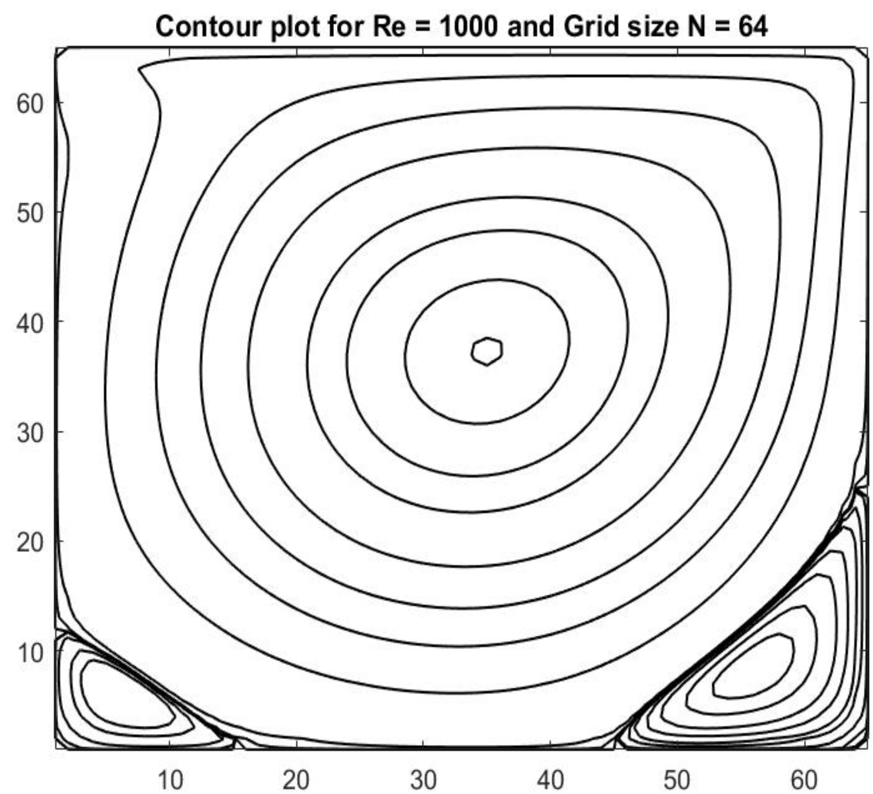


Figure 11: Contours of streamfunction for Re=1000 and N=64

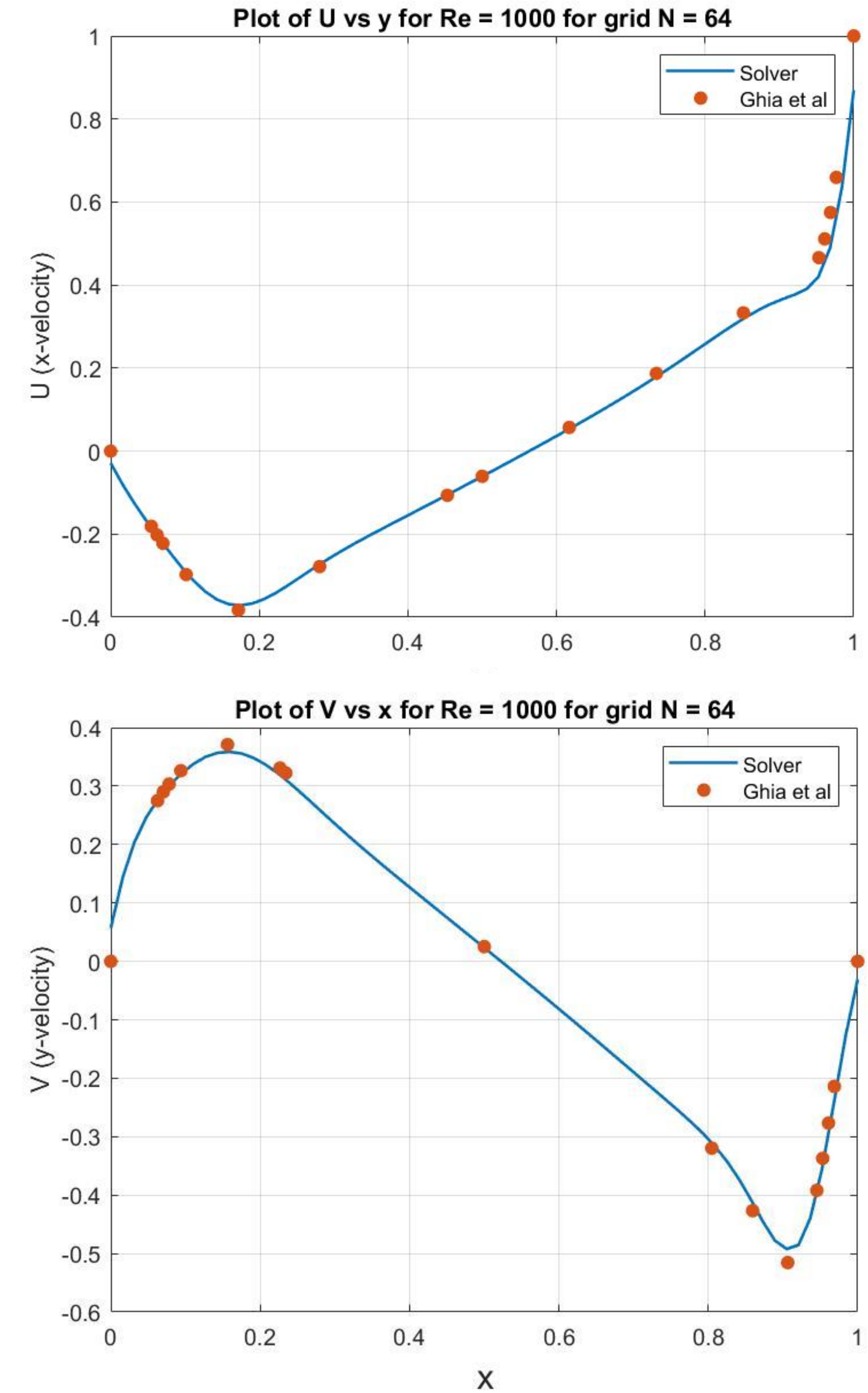


Figure 12: Plots of a) u-velocity vs y b) v-velocity vs x through the geometric center as well as values from Ghia et al for Re=1000

From these results we can easily see that as the grid size increase the accuracy of the result also increases. Also, a high ω, relaxation factor also helped us in reducing the error while calculating the PPE. This solution converges at around 5200 iterations.

3.4.2 Grid Size 128 and Reynolds number 3200

Here, I've used: N=128, Re=3200, ω=1.6, β=5

Figure 13: Plot of $|R|_{L1}$ vs iteration number for Re=3200 and N=128

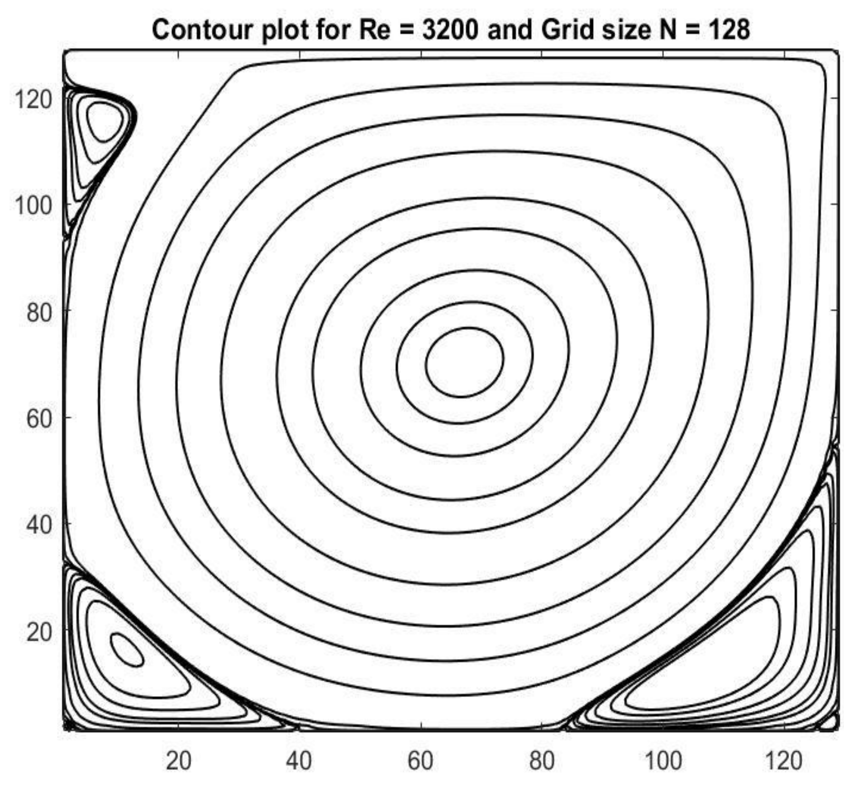
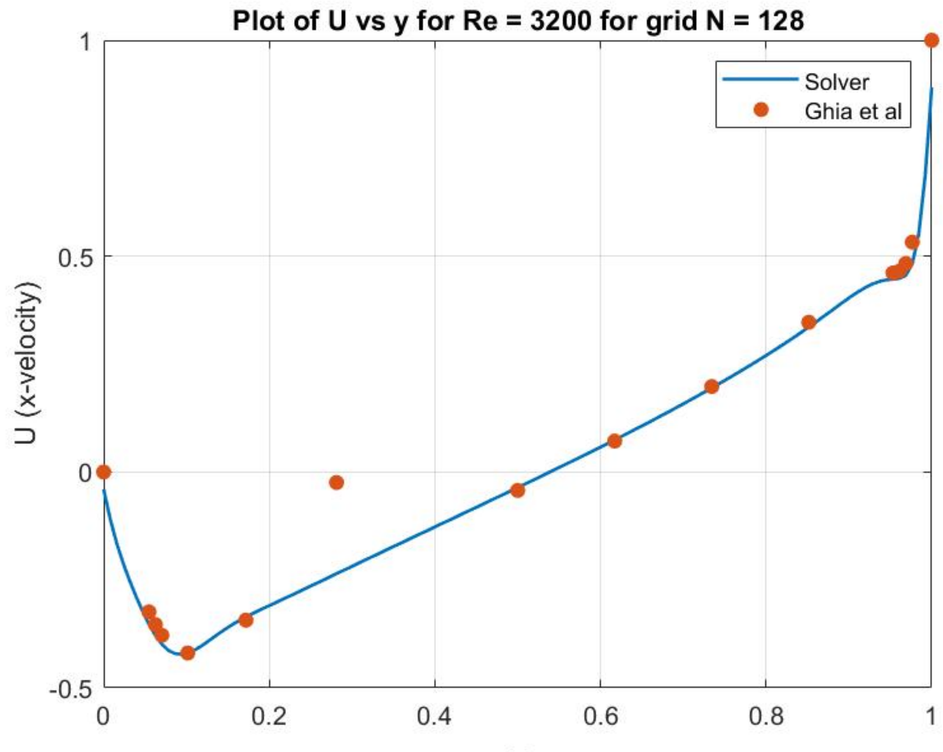


Figure 14: Contours of streamfunction for Re=3200 and N=128



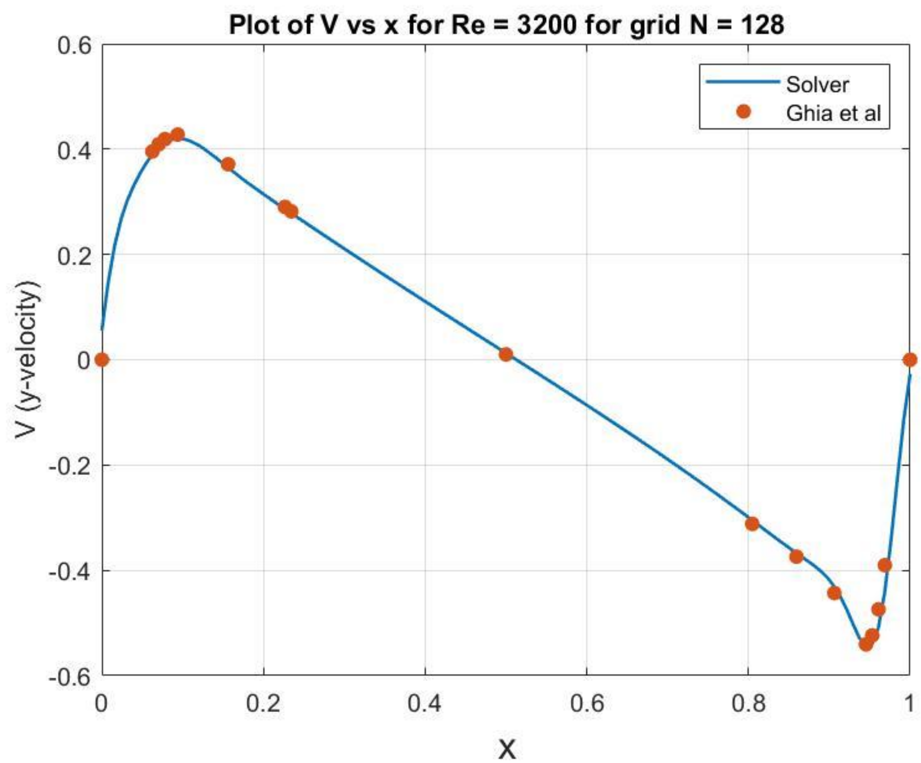


Figure 15: Plots of a) u-velocity vs y b) v-velocity vs x through the geometric center as well as values from Ghia et al for Re=3200

Since the code is not greatly optimized we can oscillations in our convergence plot. Our u & v values now exactly match the Ghia et al provided values. In the contour plot, we can start observing some disturbances in the top left corner which was absent in lower Reynolds number. This is mostly caused due to the high chaotic nature of the fluid at 3200 Re.

3.4.4 Grid Size 128 and Reynolds number 5000

Here, I've used: N=128, Re=5000, ω=1.6, β=5

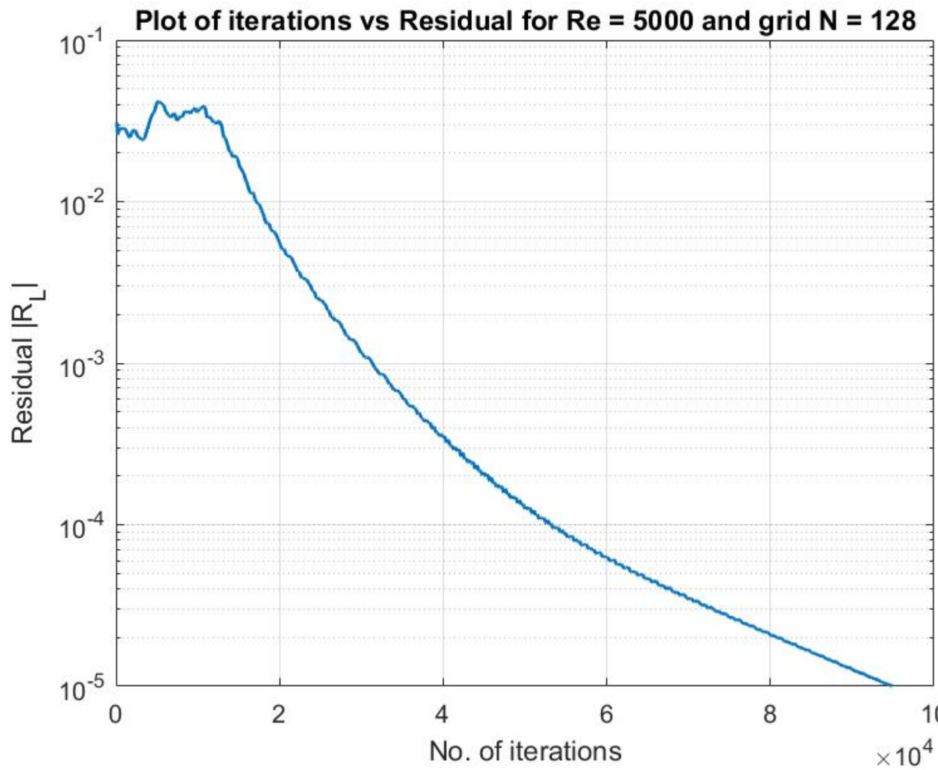


Figure 16: Plot of $|R|_{L1}$ vs iteration number for Re=5000 and N=128

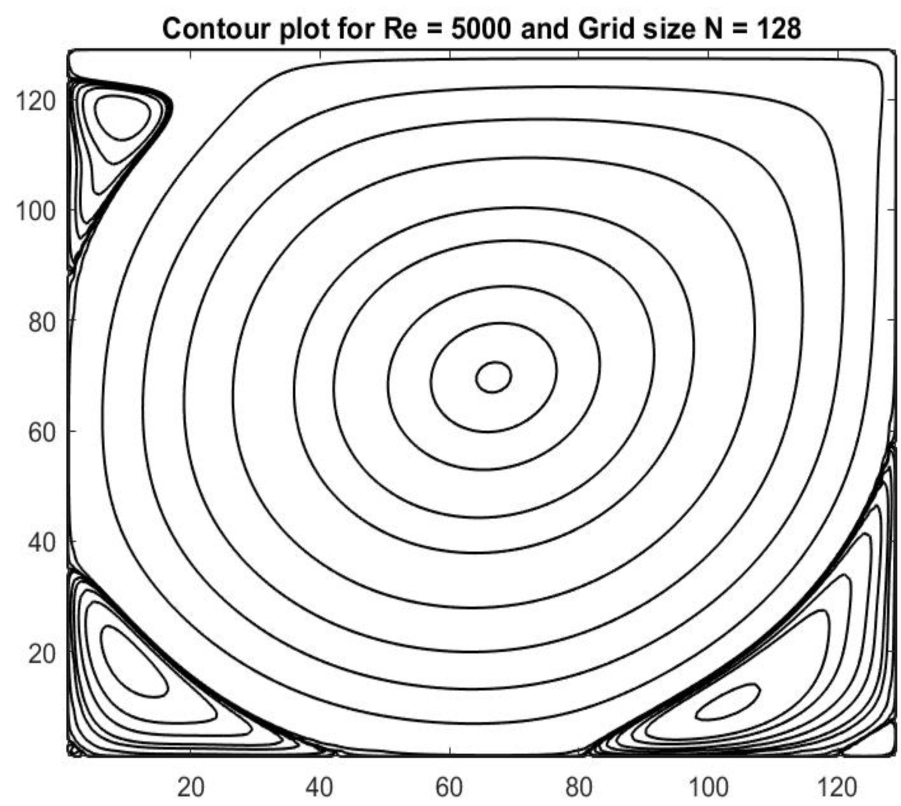
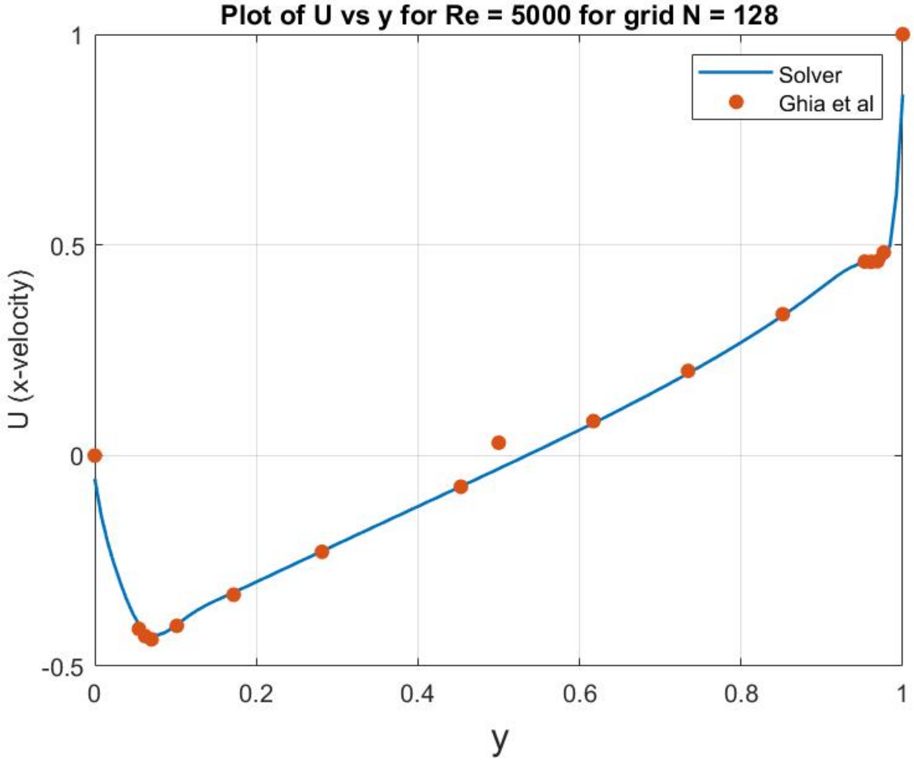


Figure 17: Contours of streamfunction for Re=5000 and N=128



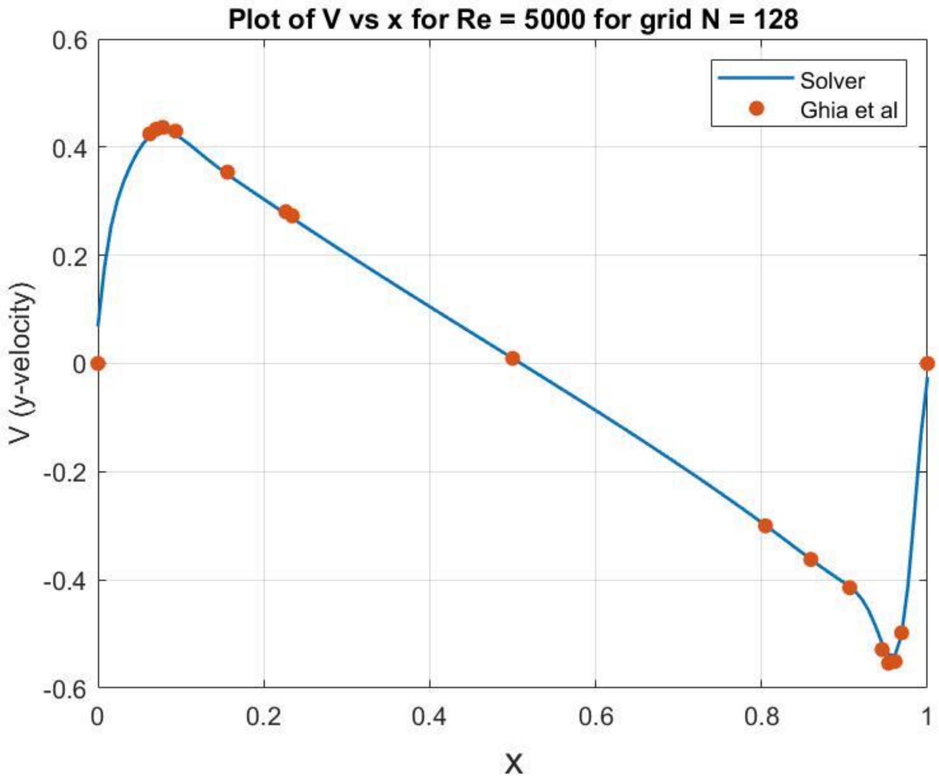


Figure 15: Plots of a) u-velocity vs y b) v-velocity vs x through the geometric center as well as values from Ghia et al for Re=1000

From the contour plots we can see that the turbulence of the fluid is further increased, and it took an astonishing 94700 iterations to converge to the required solution albeit with a lot of oscillations in between.