

1 INTRODUCTION

The stream function-vorticity formulation is a well-established method for solving the 2-D incompressible Navier Stokes equations. The "lid driven cavity problem" is often used as a benchmark to validate 2D incompressible flow algorithms Barragy and Carey (1997). In this paper, the lid driven cavity problem is approximately solved by means of a finite difference formulation. The scheme was evaluated for convergence in time and space, and the effect of increasing Reynolds number was investigated for Reynolds numbers up to 1000.

2 GOVERNING EQUATIONS

The 2-D unsteady incompressible Navier-Stokes equations along with conservation of mass are the equations governing the flow of an incompressible fluid. The equations are written as

Mass:

$$\frac{\partial u}{\partial \bar{x}} + \frac{\partial v}{\partial \bar{y}} = 0$$

X-momentum:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

Y-momentum:

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right)$$

These are the equations in non-conservative, or "primitive variable" form. Unfortunately there is no transport equations for updating the pressure. Instead we use the stream function, defined as standard with

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

This stream function will identically satisfy continuity.

Vorticity is defined in 2D as

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

Substitute the stream function into vorticity as defined above to get

$$\nabla^2 \psi = -\omega(t)$$

This is Poisson's equation that says the stream function at any instant is determined by the vorticity field at that instant.

Now we derive the vorticity equation. apply the 2D vorticity definition to the x and y momentum equations. the resulting equation is

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

Normalize the equations by applying the following normalization functions:

$$x = \frac{\bar{x}}{L} \quad y = \frac{\bar{y}}{L} \quad u = \frac{\bar{u}}{U_e} \quad v = \frac{\bar{v}}{U_e} \quad t = \frac{\bar{t}}{L} U_e \quad \omega = \frac{\bar{\omega}}{\omega_\infty}$$

Then the Stream Function Vorticity Transport Equation is:

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

Normalize the vorticity and all derivatives using the normalization functions. Substitute in all normalized first and second derivatives. Substitute in the non dimensional vorticity. Simplify. Use $R_e = \frac{\rho u_e L}{\mu}$ and the definition of kinematic viscosity:

$$\nu = \frac{\mu}{\rho}$$

The final formulation of the vorticity transport equation, using the now non-dimensional parameters, is now as follows (details available on request)

$$\frac{\partial \omega}{\partial t} + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = \frac{1}{R_e} \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right)$$

Stream function is similarly updated.

3 NUMERICAL FORMULATION

The overall scheme will act in two parts. One for vorticity transport and one for the stream function. First, vorticity is computed via the definition at the boundaries of the cavity. This will use a second order accurate forward difference, with appropriate sign changes on terms to account for the unit vectors at the walls. Then vorticity is updated via an approximate factorization, to be described below. After vorticity is updated by one time step, the stream function is updated via Gauss-Seidel iteration, to be explained as well. Finally, velocity is updated using the definition of the stream function, and the process is repeated until steady state is reached.

Convergence criteria in time is basically that vorticity stops changing with time on the interior. Convergence criteria for the stream function is that the stream function stops changing with each iteration. Convergence results and analysis are presented in more detail in the convergence section.

3.1 VORTICITY TRANSPORT

The equation for vorticity update is solved by means of an approximate factorization method. Specifically, an alternating direction implicit (ADI) method to get around the non-tridiagonal form that the 2D cavity implicit equations will take. This should give similar stability to a standard implicit method, but allow us to update the vorticity as a two stage process, with each

sweep involving the solution of a tri-diagonal system of equations. The method derivation is here reproduced from the work of Dr. Guillot, University of New Orleans.

First, define the operators:

$$L_x = u \frac{\partial}{\partial x} - v \frac{\partial}{\partial x^2} \quad L_y = v \frac{\partial}{\partial y} - v \frac{\partial}{\partial y^2}$$

The vorticity in operator form is then

$$\frac{\partial \omega}{\partial t} + (L_x + L_y) \omega = 0$$

Using $\Delta\omega^{n+1} = \omega^{n+1} - \omega^n$ a forward difference for the time term a two level time discretization is

$$\frac{\Delta\omega}{\Delta t} + (1 - \beta)(L_x + L_y)\omega^n + \beta(L_x + L_y)\omega^{n+1} = 0$$

Or, in delta form,

$$\frac{\Delta\omega^{n+1}}{\Delta t} + \beta(L_x + L_y)\Delta\omega^{n+1} = -(L_x + L_y)\omega^n$$

And rearrange as

$$[1 + \beta(L_x + L_y)]\Delta\omega^{n+1} = -\Delta t(L_x + L_y)\omega^n$$

With $\beta = 0$, the method is explicit, and for $\beta = 1$, the method is implicit. Factor the term in brackets for the crucial step in the approximate factorization method.

$$[1 + \beta(L_x + L_y)]\Delta\omega^{n+1} = [1 + \beta\Delta t L_x][1 + \beta\Delta t L_y] - \beta^2\Delta t^2 L_x L_y$$

Neglect the last term in the above equation. Since the time discretization is only $\mathcal{O}[\Delta t]$ accurate, this will cause no loss of accuracy. Substitute this result back into the scheme as

$$[1 + \beta\Delta t L_x][1 + \beta\Delta t L_y]\Delta\omega^{n+1} = -\Delta t(L_x + L_y)\omega^n$$

Now if we define

$$[1 + \beta\Delta t L_y]\Delta\omega^{n+1} = \omega^*$$

Then we can update to the new time level in two steps as follows

$$[1 + \beta\Delta t L_x]\Delta\omega^* = -\Delta t(L_x + L_y)\omega^n$$

$$[1 + \beta\Delta t L_y]\Delta\omega^{n+1} = \Delta\omega^*$$

The operators L_x and L_y are still non-linear, so this will have to be addressed before the scheme is finished.

There are various methods in the literature for handling nonlinear velocity terms. Some that have previously been discussed include

1. Lag the Velocities: Produces a Δt restriction
2. Iterate for the Velocities within every time step (e.g. A matrix Newton's method)
3. Extrapolate, e.g. 2 previous time levels to find the updated velocities.

For our scheme, we chose to lag the velocities.

The vorticity operators are restated for convenience as

$$L_x = u \frac{\partial}{\partial x} - v \frac{\partial}{\partial x^2} \quad L_y = v \frac{\partial}{\partial y} - v \frac{\partial}{\partial y^2}$$

So the scheme for the 1st part of the ADI method will be

$$\left[1 + \beta \Delta t \left(u \frac{\partial}{\partial x} - v \frac{\partial}{\partial x^2} \right) \right] \Delta \omega^* = -\Delta t \left(u \frac{\partial}{\partial x} - v \frac{\partial}{\partial x^2} + v \frac{\partial}{\partial y} - v \frac{\partial}{\partial y^2} \right) \omega^n$$

Using second order accurate finite differences, the scheme for this part can be written as

$$\Delta \omega_{i,j}^* + \beta \Delta t \left[u \left(\frac{\omega_{i+1,j}^* - \omega_{i-1,j}^*}{\Delta x} \right) - v \left(\frac{\omega_{i-1,j}^* - 2\omega_{i,j}^* + \omega_{i+1,j}^*}{\Delta x^2} \right) \right] = RHS^n$$

With the RHS as

$$RHS^n = -\Delta t \left[u_{i,j}^n \left(\frac{\Delta \omega_{i+1,j}^n - \Delta \omega_{i-1,j}^n}{2\Delta x} \right) - v \left(\frac{\Delta \omega_{i-1,j}^n - 2\Delta \omega_{i,j}^n + \Delta \omega_{i+1,j}^n}{\Delta x^2} \right) \right] \\ - \Delta t \left[v_{i,j}^n \left(\frac{\Delta \omega_{i,j+1}^n - \Delta \omega_{i,j-1}^n}{2\Delta x} \right) - v \left(\frac{\Delta \omega_{i,j-1}^n - 2\Delta \omega_{i,j}^n + \Delta \omega_{i,j+1}^n}{\Delta x^2} \right) \right]$$

This tri-diagonal system is then inverted to find ω^* . The scheme for the 2nd part is then as follows

$$\Delta \omega^{n+1} + \beta \Delta t \left[u \left(\frac{\omega_{i+1,j}^{n+1} - \omega_{i-1,j}^{n+1}}{\Delta x} \right) - v \left(\frac{\omega_{i-1,j}^{n+1} - 2\omega_{i,j}^{n+1} + \omega_{i+1,j}^{n+1}}{\Delta x^2} \right) \right] = \Delta \omega^*$$

Note one important aspect of the scheme that is visible here: There are no convective terms on the main diagonal. This could lead to problems with a non-dominant main diagonal at high Reynolds numbers, where convection tends to dominate the terms on the left hand side. In such cases, unwinding provides a more stable scheme.

Another issue is the vorticity at the walls. Here we choose to write the wall vorticity as explicit functions of the other portions of the tridiagonal matrix.

For example, let A, B, and C represent the sub, main, and super diagonals, with D as the right hand side, the scheme at the “left” wall is written as

$$B_2 \Delta \omega_{2,j}^* + C_2 \Delta \omega_{3,j}^* = D_2^n - A_2 \Delta \omega_{1,j}^n$$

Similarly for the right wall

$$A_{I-1} \Delta \omega_{I-2,j}^* + B_I \Delta \omega_{I,j}^* = D_{I-1}^n - C_{I-1} \Delta \omega_{I,j}^*$$

These explicit lagged terms will limit the maximum size of our time step, despite our mostly implicit method.

After the computation of the $\Delta\omega^{n+1}$, terms, update ω one the interior with $\Delta\omega^{n+1} = \omega^{n+1} - \omega^n$.

3.2 STREAM FUNCTION

The Poisson's Equation for the stream function is updated via Gauss-Seidel iteration.

Using central differencing for the 2nd derivative terms, the initial discretization is

$$\frac{\psi_{i-1,j} - 2\psi_{i,j} + \psi_{i+1,j}}{\Delta x^2} + \frac{\psi_{i,j-1} - 2\psi_{i,j} + \psi_{i,j+1}}{\Delta y^2} = -\omega_{i,j}^{n+1}$$

Where the center stencil point $\psi_{i,j}$ is evaluated at the $k+1$ level, and the other stencil points are at the k level. Rearrange terms according to iteration level

$$\frac{\left[\omega_{i,j}^k + \left(\frac{\psi_{i-1,j}^k + \psi_{i+1,j}^k}{\Delta x^2} + \frac{\psi_{i,j-1}^k + \psi_{i,j+1}^k}{\Delta y^2} \right) \right]}{\left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} \right)} = \psi_{i,j}^{k+1}$$

The solution sweeps through the interior points, using values of the neighboring points, some of which may be updated, to calculate new values of the stream function. This is a Gauss-Seidel technique. The solution continues until convergence is reached, defined as the max norm of the difference in ψ over the interior of the plate from one iteration level to the next.

$$Max_{interior} \left\| \psi_{i,j}^{k+1} - \psi_{i,j}^k \right\|$$

When the max norm falls below some threshold, the algorithm is assumed to have converged. For the time dependent flow studied here, final Poisson's calculations are done with convergence tolerances equal to 10^{-16} , effectively machine zero.

Finally, to improve performance of this portion of the code, the successive over relaxation (SOR) technique was employed. After each ψ value is calculated, it is modified by the following formula:

$$\psi_{i,j}^{k+1} = (1 - \beta) \psi_{i,j}^k + \beta \psi_{i,j}^{k+1}$$

β values for the SOR technique were documented for each run. Additionally, tests to examine the effect of time step size on stability were performed without SOR.

After the stream function is calculated on the plate, u and v are updated on the interior using second order accurate central differences in space. At this point, the u and v values are used to update vorticity at the walls (using the 2nd order accurate forward differences), and the process repeats, with the ADI stages to update vorticity and Poisson's equation to update the stream function, until convergence to steady state. Note that this will not imply a convergence to any analytic value, but only convergence to steady values.

4 RESULTS

A computer code was written in Fortran to solve the stream function-vorticity formulation, as described above, on a cavity domain ($0 \leq x \leq L$), ($0 \leq y \leq L$), $L = 1$ with initial conditions most succinctly shown in the following figure:

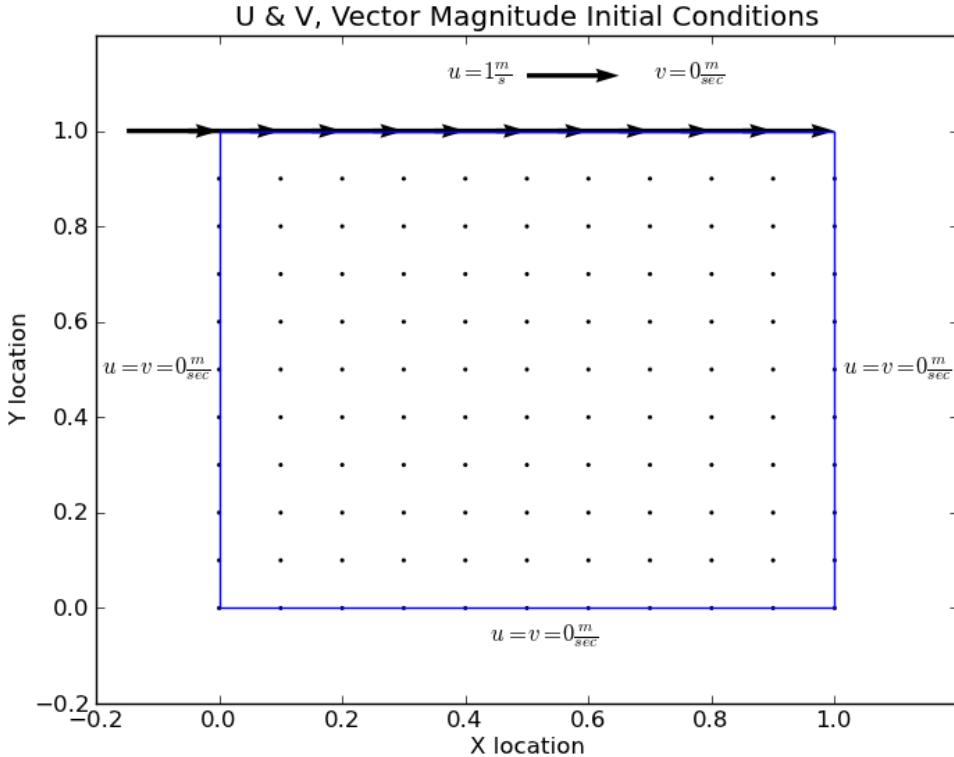


Figure 4.1: **Initial Conditions**

The figure above shows the initial conditions for a domain discretized by 11 points in x and 11 points in y. The important thing to note is that the vertical velocity, v is set to zero at the walls. Horizontal velocity, u , is set to zero at all walls except the upper lid, which is set to 1. Hence the name "driven cavity flow". Note, all velocities are initialized to zero on the interior. Additionally, All stream function and vorticity values are set to zero on the interior of the plate. Vorticity is then computed on the boundaries using the given u and v values, then the scheme is ready to iterate until steady state following the process described in the previous sections.

5 CONVERGENCE

There are 3 types of convergence for our scheme. They are as follows:

1. Convergence of Discretization
2. Convergence to steady state
3. Iterative Convergence

Because there is no analytic solution, we have no test of the convergence of our scheme to the true solution. We still care whether our scheme is performing well, however. First we test the convergence of the scheme as the number of points are increased in the cavity. Accordingly, a series of trials were computed where the scheme was tested on meshes of increasing point density, for a Reynolds number of 10. Results are shown next.

5.1 CONVERGENCE OF DISCRETIZATION

First, we will examine the velocity plots in the cavity as Reynolds number increases.

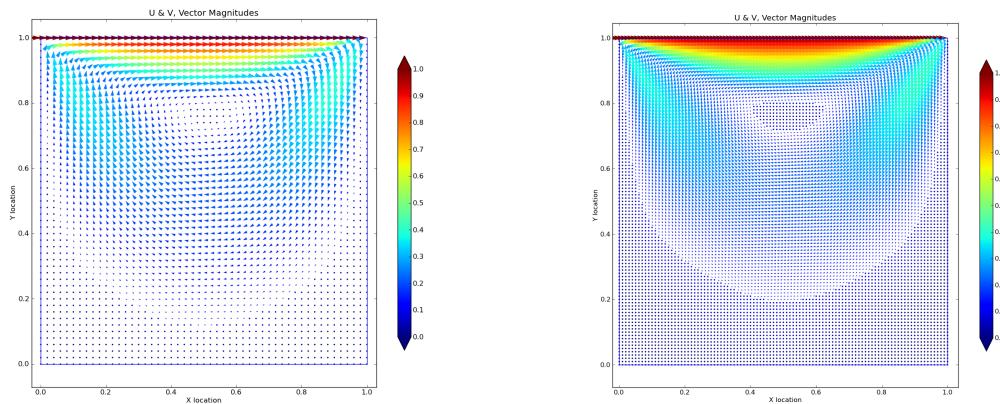


Figure 5.1: Plots of the 51×51 Mesh (Left), with the 101×101 Mesh (Right) Velocity Vectors

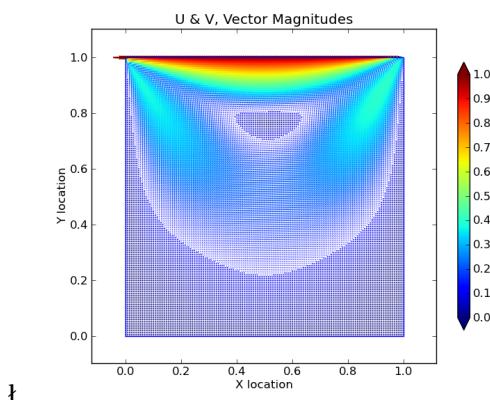


Figure 5.2: Plots of the 151×151 Mesh Velocities

The three plots shown in figures 5.1, and 5.2 on this page display the resultant vector of the U and V velocities on the plate, with the arrows (admittedly small arrows, I hope you can see them) pointing in the direction of the overall velocity vector, showing the overall swirl pattern of the flow. As will be explained in the "Convergence to Steady State" subsection, each plot in the figures shown here represents a steady state solution for the given mesh. Convergence as mesh density is increased is clearly visualized as the distribution of velocity magnitudes show in the color plots.

Velocity doesn't appear to change much between the 101×101 and 151×151 grid.

In general, the direction of the velocity vectors appeared to converge as well, though this is more difficult to visualize, due to the density of the vectors on the plot. For additional clarity, the plots above are enlarged to more clearly show the vector arrows denoting velocity directions.

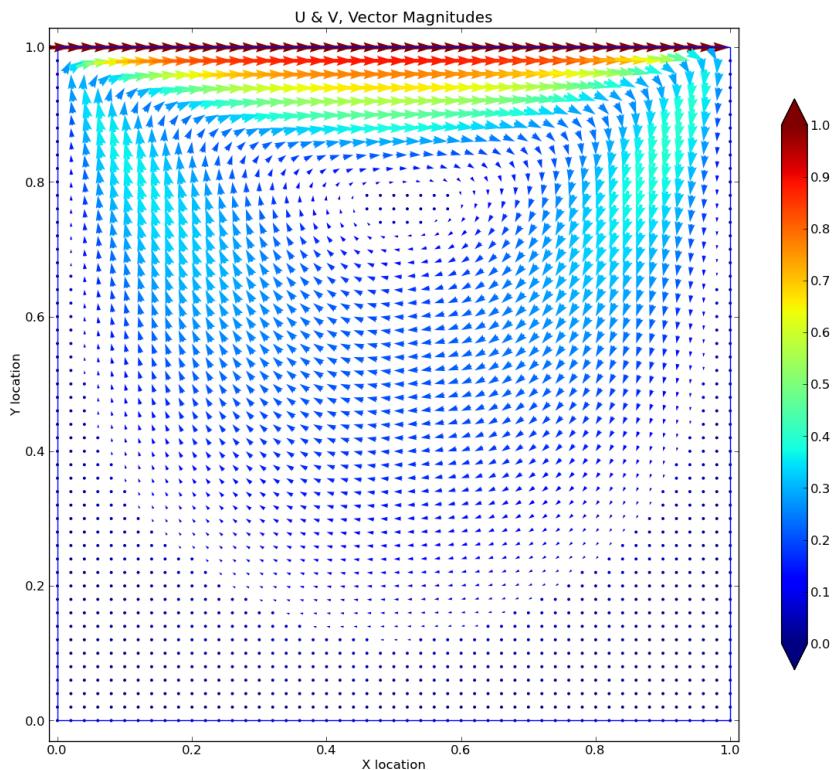


Figure 5.3: Enlarged plot showing vector direction. $Re=10$, 51×51 Mesh

For the figure 5.3 above, with a 51×51 mesh, note the lower extent of the region of moving flow.

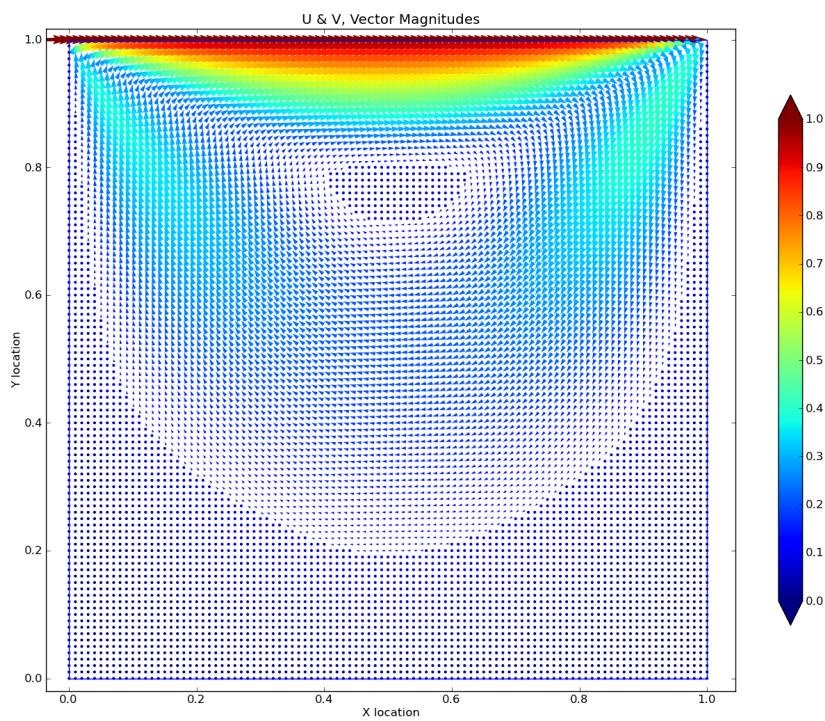


Figure 5.4: **Enlarged plot showing vector direction. Re=10, 101 x 101 Mesh**

For the figure 5.4 above, with a 101 x 101 mesh, note the flow has tightened up a little as compared to the 51 x 51 mesh.

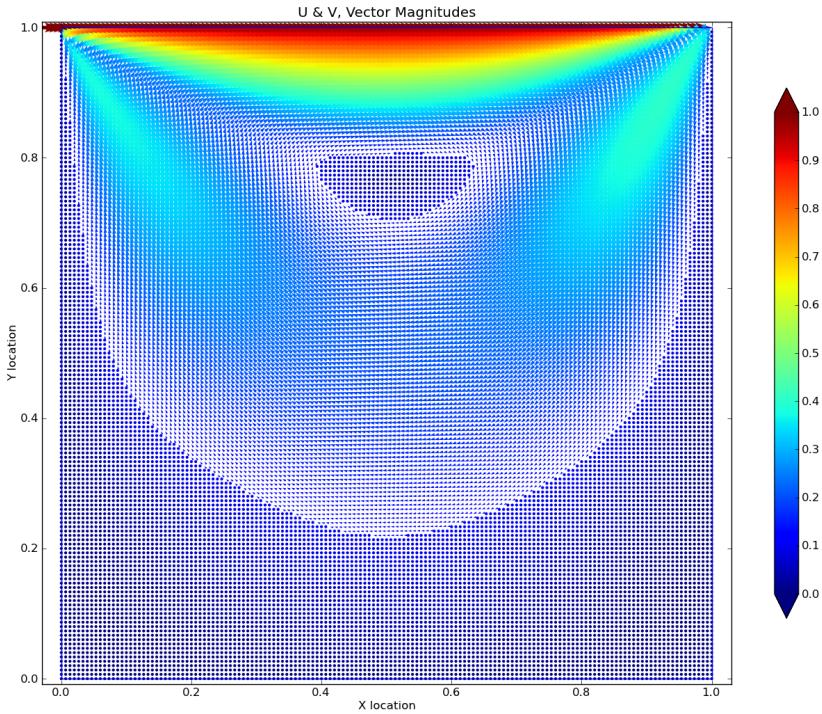


Figure 5.5: Enlarged plot showing vector direction. Re=10, 151 x 151 Mesh

For figure 5.5 above, with a 151 x 151 mesh, note the flow appears virtually identical to the flow from the 101 x 101 mesh.

However, these visualizations, while useful in showing that the flow appears to be converging, are not perfect at establishing convergence. A quick check near the middle of the cavity was performed to see how one value for vorticity performed as mesh spacing decreased.

5.2 CONVERGENCE TO STEADY STATE

Convergence to steady state was judged by finding the max percentage change of vorticity on the plate for each time step. When that change fell below 10^{-11} , the solution was found to be as converged as possible given my code, because vorticity percentage change would then oscillate about that some mean value of approximately 5^{-11} .

The formula to compute convergence is as follows:

$$Max_{interior} \left(\frac{\|\omega_{i,j}^{k+1} - \omega_{i,j}^k\|}{\omega_{i,j}^k} \right)$$

For each point on the interior of the cavity, after each time step update is complete, the magnitude percentage change from one time step to the next is computed, and the maximum

value of that computation found over the entire cavity is used as the criteria for stopping the solver. The solver continues to run until this magnitude falls below the 10^{-11} listed above.

5.3 ITERATIVE CONVERGENCE

Convergence of the Gauss Seidel update routine for the stream function in the cavity is evaluated after each sweep over the plate in the following manner: The magnitude percentage change from one iteration step to the next is computed for the stream function at each point, and the maximum value of that computation found over the entire cavity is used as the criteria for stopping the Gauss Seidel update.

The formula to compute the convergence is as follows:

$$Max_{interior} \left(\frac{\|\psi_{i,j}^{k+1} - \psi_{i,j}^k\|}{\psi_{i,j}^k} \right)$$

This routine is run to approximately numerical zero, or just below 10^{-16} for fortran double precision. Note that a cheaper criteria for convergence to steady state might be to simply count the number of iterations needed in the Gauss Seidel updates. When that Poisson's equation no longer needs to iterate to convergence for each time step, the scheme is effectively converged. This was not the criteria selected for my study, but did appear to be viable option, at least from the heuristic standpoint.

6 TIME STEP

The code was tested to find the maximum allowable time step for certain mesh sizes at a Reynolds number of 10, 100, and 1000, for grids of 11×11 , 51×51 , and 101×101 . A plot of the data is shown below.

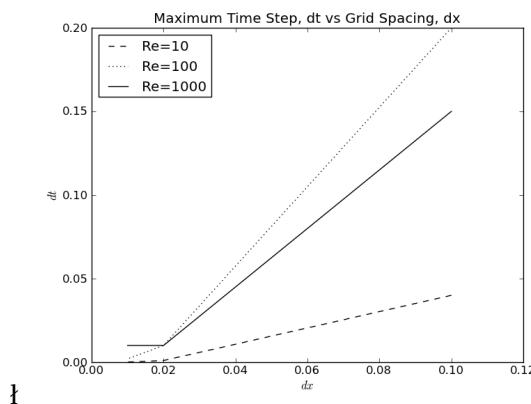


Figure 6.1: Maximum Allowable Time Step as Reynolds number and Mesh Size Vary

In figure 6.1 above, and for all tests with this code, mesh discretization was equal in the x and y directions, with the cavity always being of equal length, 1. Generally, as can be seen from this graph, as mesh density increases, the maximum time step allowable decreases. This matches with the intuition that for smaller spatial step sizes, at equal time steps, the solution might tend to become unstable if, for example, velocities or vorticity gradients are too high. In effect, transport is more likely to "outrun" the stencil as dx decreases and time step remains constant. Therefore, the plot above seems to show the correct trend.

7 REYNOLDS NUMBER EFFECT

With scheme behavior established for variations in mesh size and time step, the effect of Reynolds number on flow solutions was investigated. Reynolds number was particularly interesting because of the aforementioned structure of the tridiagonal system of equations. Specifically, the lack of convection terms on the main diagonal of the tridiagonal systems mean that as Reynolds number is increased, convective terms tend to dominate and promote instability within the system.

Other effects, such as the Hopf bifurcation, in which the solution becomes periodic in time, take place at an estimated $\text{Re} > 7500$, Barragy and Saad (2006) and thus could not be investigated because of the expected lack of stability of the scheme at Reynolds numbers much above 1000, again due to the convective terms off the main diagonal. For this study, the plate was fixed at 101 x 101 mesh points. 1st, Reynolds number was set to 100, then 1000. Results are graphed below.

7.1 REYNOLDS NUMBER = 100

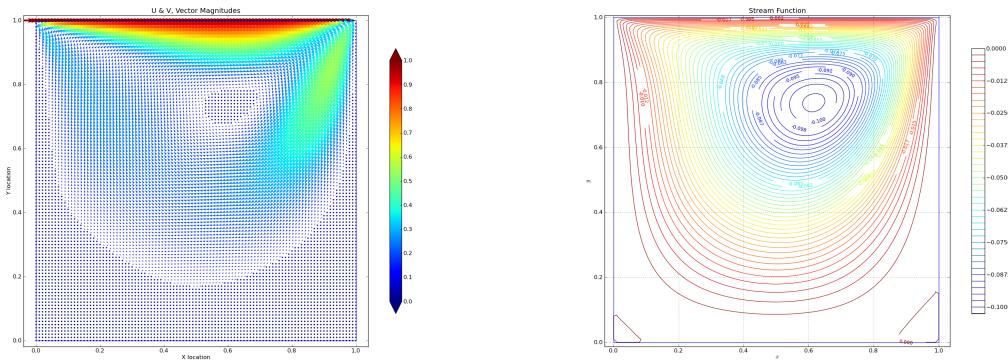


Figure 7.1: $\text{Re}=100$, 101 x 101 Mesh - Velocity Plot (Left), with the Stream Function Plot (Right)

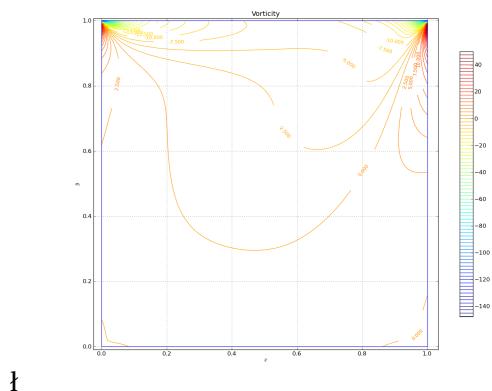


Figure 7.2: Vorticity, $\text{Re}=100$, 101 x 101 Mesh

This plot ran to iterative and steady state convergence in approximately 31 simulated seconds. Notice that the center of circulation has shifted to the right versus the plots at $\text{Re}=10$ in the previous section. Also, possible eddies are visible on the stream function plot. These are expected to become more prevalent at the Reynolds number is increased, as long as the scheme and flow remain stable.

208 simulated seconds.

7.2 REYNOLDS NUMBER = 1000

The plots below were computed on a mesh size of 101 x 101 nodes for a Reynolds number of 1000. Convergence to numerical steady state occurred at approximately

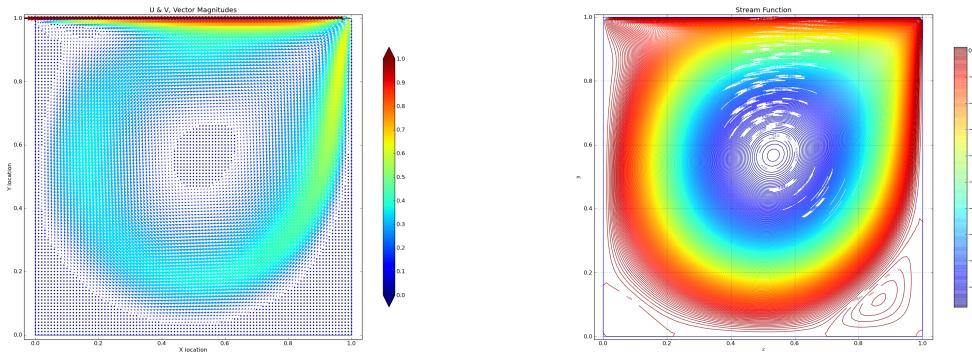


Figure 7.3: **Re=1000, Velocity Plot (Left), and Stream Function Plot (Right)**

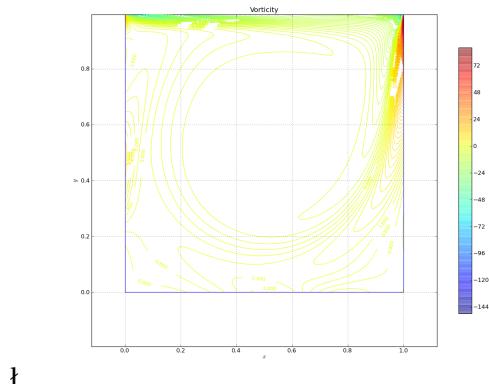


Figure 7.4: **Vorticity Plot. Re=1000**

In figure 7.3 on the right, possible secondary or tertiary eddies are seen to be forming near the bottom corners of the cavity. Note also, that for figure 7.3 and figure 18, the number of contour lines has been increased to better show the behavior of the flow. If additional time was available, it would be interesting to test the code on flows of higher Reynolds number, to see if results might match with plots of Barragy and Saad (2006), though it is doubtful that very accurate results would be obtained. A possible next step for this code would involve computing "better", convective terms, and using upwinding instead of central differences, or more stable difference schemes.

REFERENCES

- Barragy, C. and Saad, M. (2006). 2d lid-driven cavity problem revisited. *Computers and Fluids*, 35:326–348.
- Barragy, E. and Carey, G. (1997). Stream function-vorticity driven cavity solution using p finite elements. *Computers and Fluids*, 26(5):453–468.