MAE 820 Project 2

D. W. Gould

Preprint submitted to Annals of Torturing Grad Students *May 11, 2017*

Contents

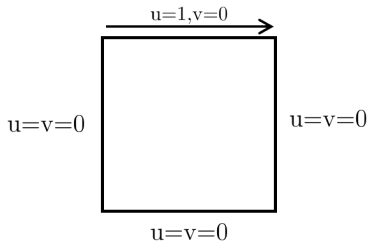


Figure 1: Diagram of model geometry

1. Introduction

In this project, the finite difference method was used to solve the lid driven cavity problem. The effectiveness of each solver used was also analyzed and additional suggestions for further reductions in computational times were presented.

2. Model

2.1. Geometry

In Figure 1, a basic diagram of the geometry is shown. In this problem, the south, east, and west walls of the enclosure are assumed to be stationary while the north wall is assumed to be sliding in the positive x-direction with a non-dimensional velocity of one.

2.2. Governing Equations

In this work, it is assumed that the fluid within the cavity is not rarefied and can be accurately considered to have both a constant density and viscosity. With these assumptions, the two-dimensional versions of the conservation of mass and conservation of momentum can be simplified to the forms shown in Equations 1, 2, and 3.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{1}$$

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial}{\partial y}(uv) + \frac{\partial P}{\partial x} = \frac{1}{\mathrm{Re}}\left[\frac{\partial u^2}{\partial x^2} + \frac{\partial v^2}{\partial y}\right] \tag{2}$$

$$\frac{\partial v}{\partial t} + \frac{\partial}{\partial x}(uv) + \frac{\partial v^2}{\partial y} + \frac{\partial P}{\partial y} = \frac{1}{\mathrm{Re}}\left[\frac{\partial u^2}{\partial x^2} + \frac{\partial v^2}{\partial y}\right] \tag{3}$$

3. Solution Method

3.1. Projection Method

Although the steady-state solution of the fluid pressure and velocity is desired, a transient method is used to

$$\frac{\partial u}{\partial t} = \frac{u^{n+1} - u^\star}{\Delta t} \tag{4a}$$

$$\frac{\partial v}{\partial t} = \frac{v^{n+1} - v^\star}{\Delta t} \tag{4b}$$

$$\frac{u^\star - u^n}{\Delta t} = -\frac{\partial u^2}{\partial x} - \frac{\partial}{\partial y}(uv) + \frac{1}{\mathrm{Re}}\left[\frac{\partial u^2}{\partial x^2} + \frac{\partial v^2}{\partial y}\right] \tag{5a}$$

$$\frac{v^\star - v^n}{\Delta t} = -\frac{\partial v^2}{\partial y} - \frac{\partial}{\partial x}(uv) + \frac{1}{\mathrm{Re}}\left[\frac{\partial u^2}{\partial x^2} + \frac{\partial v^2}{\partial y}\right] \tag{5b}$$

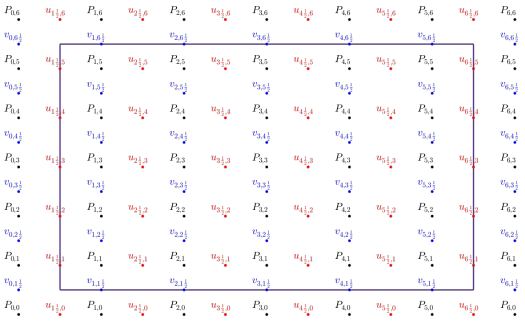


Figure 2: Example of staggered grid

$$\left[\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2}\right]^{n+1} = \frac{1}{\Delta t}\left[\frac{\partial u^\star}{\partial x} + \frac{\partial v^\star}{\partial y}\right] \tag{6}$$

The method used to solve Equation 6 is described in Section 3.3. Once the equation is solv

$$u^{n+1}_{i+\frac{1}{2},j} = u^\star_{i+\frac{1}{2},j} + \frac{\Delta t}{\Delta x}\left(P^{n+1}_{i+1,j} - P^{n+1}_{i.j}\right) \tag{7a}$$

$$v^{n+1}_{i,j+\frac{1}{2}} = v^\star_{i,j+\frac{1}{2}} + \frac{\Delta t}{\Delta y}\left(P^{n+1}_{i,j+1} - P^{n+1}_{i.j}\right) \tag{7b}$$

3.2. Discretization and Boundary Conditions

The governing equations of Section 2.2 were discretized using a staggered grid. Figure 2shows the node staggering method used in this work for an example grid consisting of 25 interior pressure nodes. In theory, use of the staggered mesh should eliminate any checkerboard error from occurring. In practice, it was found that the additional complexity of incorporating the staggered mesh into the computational algorithm actually often led to checkerboard style error being observed in the results.

A central differencing scheme was used to discretize the derivatives in Equations 47. Interpolation was used whenever the scheme required information at a location between nodes.

Examples of this discretization for some of the terms in Equations 47 are given in Equations 8 and 9.

$$\left[\frac{\partial}{\partial x}\left(u^2\right)\right]_{i+\frac{1}{2},j} = \frac{\left(u_{i+1,j}\right)^2 - \left(u_{i,j}\right)^2}{\Delta x} \tag{8}$$

$$\left[\frac{\partial}{\partial y}\left(uv\right)\right]_{i+\frac{1}{2},j} = \frac{\left(uv\right)_{i+\frac{1}{2},j+\frac{1}{2}} - \left(uv\right)_{i+\frac{1}{2},j-\frac{1}{2}}}{\Delta y} \tag{9}$$

For the u and v nodes, the nodes lying along their south, east, and west boundaries were defined so that fluid velocity was zero at the walls. If the outer velocity node was located outside of the cavity, its value was defined so as the average between the outer node and its nearest interior node was equal to zero.

For the north boundary, a similar method was employed, however in this case the outer u velocity nodes were set so that the average between themselves and their nearest interior node would be equal to 1. The v velocity nodes along the north boundary were again set to zero, representing a no slip condition along the sliding wall.

3.3. Poisson Solver

Using the calculated $u^\star$and $v^\star$, the pressure Poisson equation was solved using the Successive Over-Relaxation method.

$$C_{i,j}^k = \frac{1}{\Delta t}\left[\frac{u^\star_{i+\frac{1}{2},j} - u^\star_{i-\frac{1}{2},j}}{\Delta x} + \frac{v^\star_{i,j+\frac{1}{2}} - v^\star_{i,j-\frac{1}{2}}}{\Delta y}\right] \tag{10}$$

$$\frac{P_{i+1,j} - 2P_{i,j} + P_{i-1,j}}{(\Delta x)^2} + \frac{P_{i,j+1} - 2P_{i,j} + P_{i,j-1}}{(\Delta y)^2} = C_{i,j} \tag{11}$$

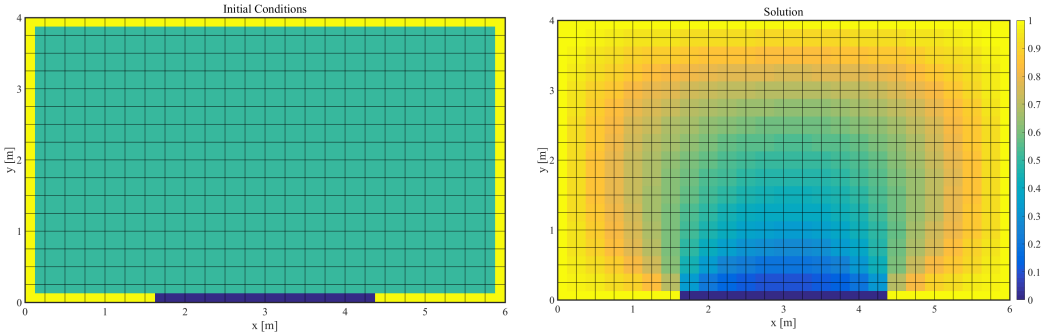
$$\beta = \frac{\Delta x}{\Delta y}$$

$$P_{i,j}^{k+1} = (1-\omega)\,P_{i,j}^k + \omega\frac{P_{i+1,j}^k + P_{i-1,j}^{k+1} + \beta^2\left(P_{i,j+1}^k + P_{i,j-1}^{k+1}\right) - (\Delta x)^2\,C_{i,j}^k}{2\left(1+\beta^2\right)} \tag{12}$$

Equation 12 shows the general formula for the Successive Over-Relaxation (SOR) method. The SOR method improves on the GS solution method by incorporating over - or, if increased stability is desired, under - relaxation into the equation using the relaxation parameter ω. When ω is equal to 1, the SOR method becomes identical to the GS method. Raising ω to values greater than one can increase the rate of convergence at the cost of stability. Conversely, lowering the relaxation parameter below unity will result in dramatically increased computational times, but provides greater stability. In this work, values of ω of between .9 and 1.3 were used.

4. Results - Part 1

The inital conditons used and results obtained for part one can be seen in Figure 3. In each of these figures, the mesh used is overlayed.



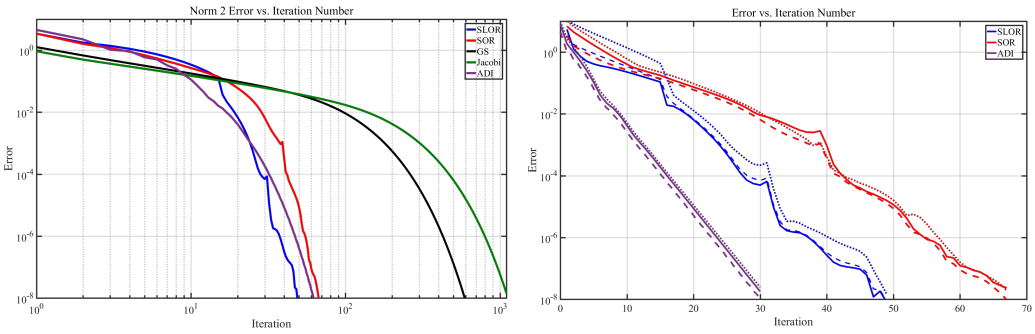
(a) Part 1 initial conditions with mesh overlay

(b) Part one solution with mesh overlay

Figure 3: Solution of part 1

In Figure 4, the effects each solver on the convergence of the solution is shown. A convergence criteria of 1E-8 was used solvers in this project.

For the second part of the this project, the box geometry was modified so that a portion of the right corner was removed. The line defining the removed corner is shown as the dotted line in 1. This change in the size of the domain was implemented by adjusting the initial conditions and boundary conditions used when solving for ψ. The new wall was applied by initially assigning all nodes beyond the new boundary line as boundary nodes possessing constant values of $\omega = 1$. A visual depiction of this application can be seen in Figure 5a.



(a) Error vs iteration number for each solver

(b) Comparison of norm 1, norm 2, norm∞ error

Figure 4: Relative error from examined solution methods

	Jacobi	**G-S**	**SOR**	**SLOR**	**ADI**
Time per Iteration [s]	5.75E-5	5.78E-5	6.58E-5	2.0E-4	2.34E-4
Solution Time [s]	6.5E-2	3.4E-2	4.5E-3	1.0E-2	7.25E-3
Optimal ω	N/A	N/A	1.7189	1.2421	1.3
Minimum Iterations	1127	592	68	50	31

Table 1: Solver effectiveness

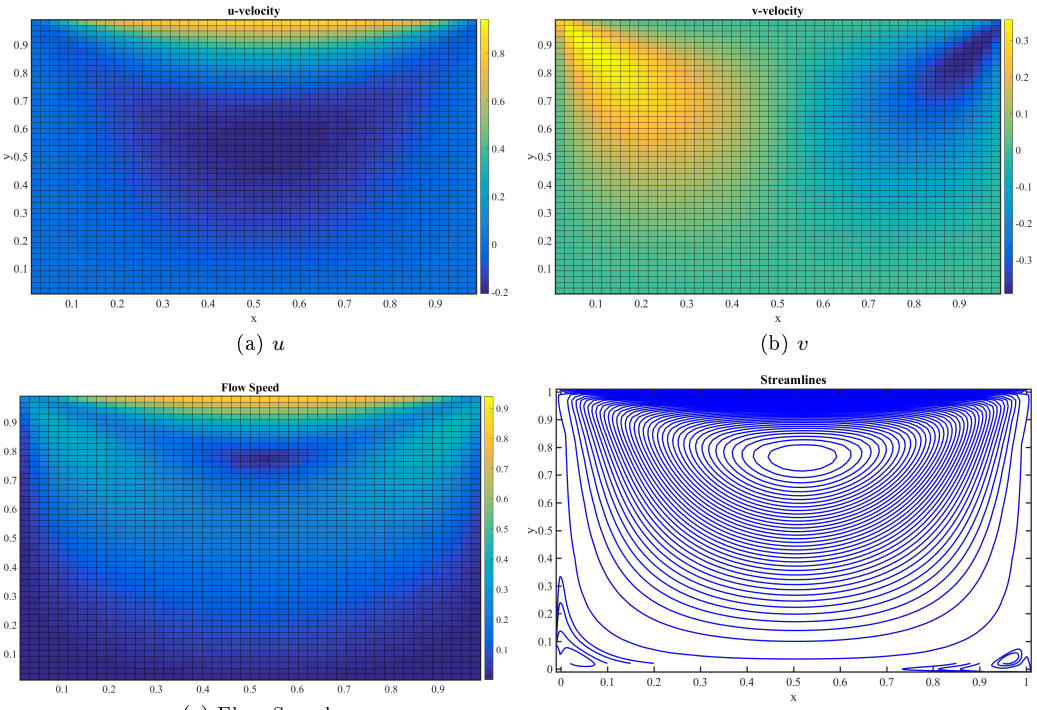
The restricted domain was seen to slightly modify the stream function calculated within the domain. This change can be seen in Figure 5b in which the a contour plot with the results from the geometry of part 1 is overlayed with the new results from the domain with the removed corner. In Figure 5b, the black contours are those from the geometry of part 1, and the colored contours are from the results calculated with the new geometry.

While Figure 5b does make it clear that *some* change in the results did indeed occur due to the modified corner boundary, in order to both better understand the effect of geometry on the flow patterns within the box and provided validation for the geometry adjustment method used, the domain of analysis was further restricted by removing all nodes beyond the boundary identified in Figure 1 as the "Modified Wall". Unsurprisingly, moving the wall this far into the domain proved to have a much larger effect.

Finally, to further disturb the flow, a partial barrier was added in the region between the inlet and outlet by assigning a line of nodes at x=3m to be equal 0. This assignment created a new wall attached to the wall between points A and C. This is seen in Figure 10a. Unsurprisingly, the addition of the barrier worked in conjunction with the extended wall and resulted in a highly disturbed flow path. A contour and pseudocolor plot depicting this is presented in Figure 10b. The code calculating this effect is provided in Appendix ??.

		dt	dx	Nodes	Tolerance	Computational Time
Re	1					
	10					
	100					
	400					
	400	2.0E-4				
	400	0.005	0.0101	100x100	5.0E-7	270
	1000	0.0036	0.0204	50x50	5.0E-7	363

Table 2: Results



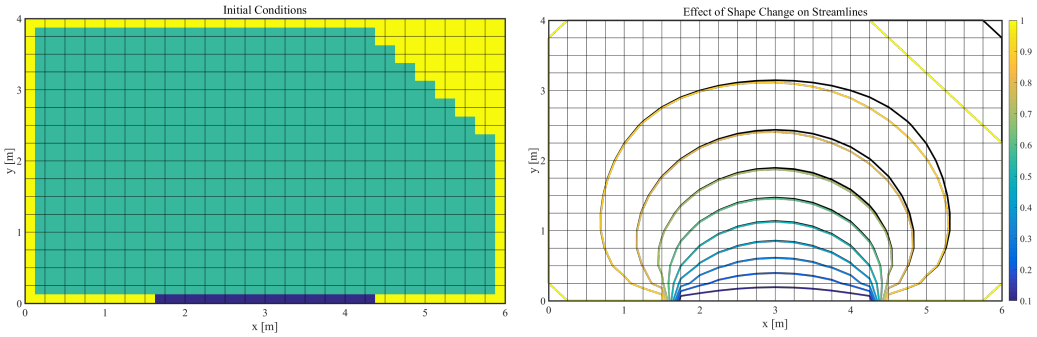
(a) u (b) v

(c) Flow Speed (d) ψ

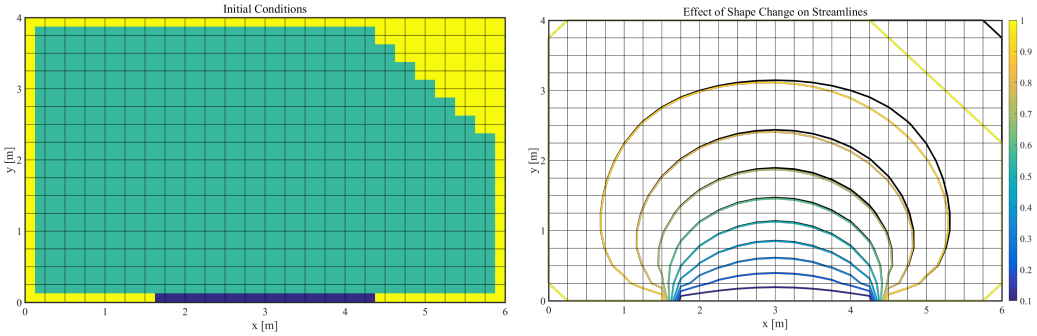
Figure 5: Results - Re 10

5. Conclusion

Despite the limited resolution, the methods employed in solving the Laplace equation for the stream function within the domain of interest proved to be very effective. Regarding the different solution methods, it was found that when using Matlab, the ability to vectorize an equation is of significant importance. This is admittedly a significant drawback if one is going to be using the language for iterative solutions to finite difference equations. Unfortunately, it was also discovered by the author that all other coding options are horrible and evil. Thus, in the future, this author would recommend that FDE equations can be best solved by either using working in Matlab and using Matlab's built in, vectorized functions as much as possible, or simply buying some computer science grad student a 6 pack of beer to program the solution code for you in a awful, but fast language such as C or Fortran.

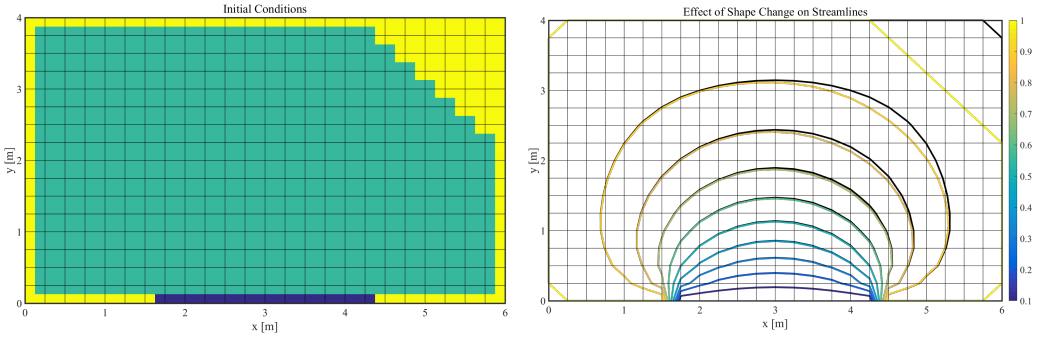


(a) u (b) v

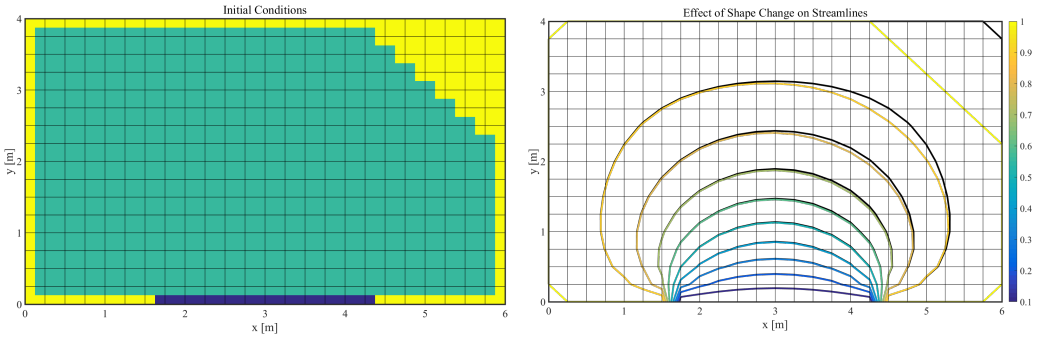


(c) Flow Speed (d) ψ

Figure 6: Results - Re 100

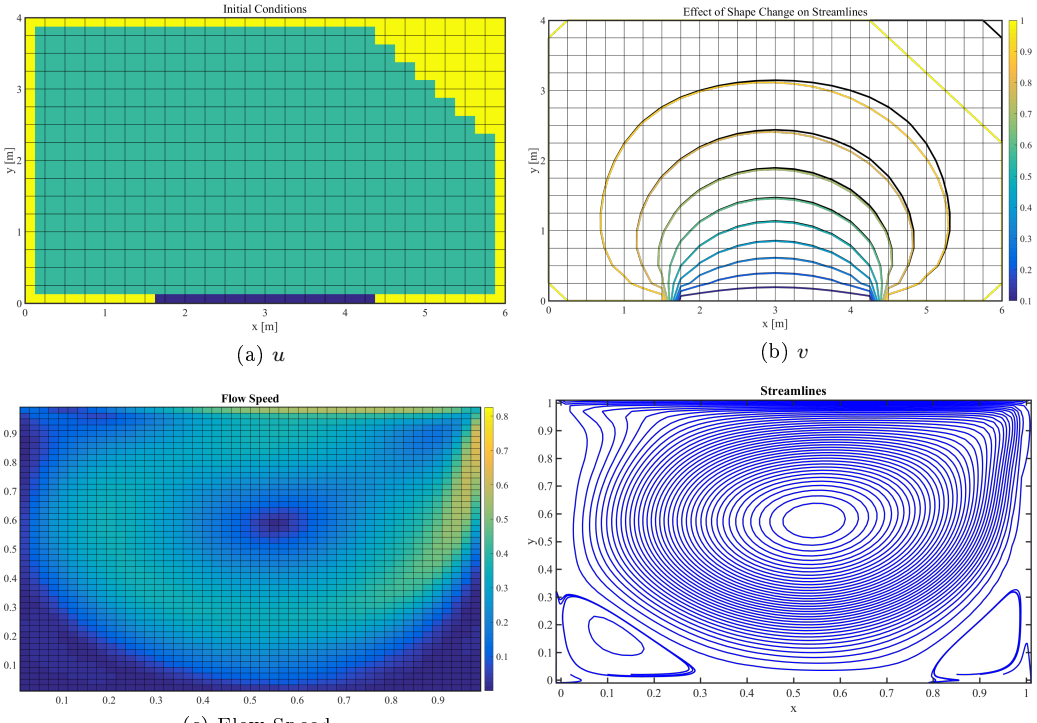


(a) u (b) v



(c) Flow Speed (d) ψ

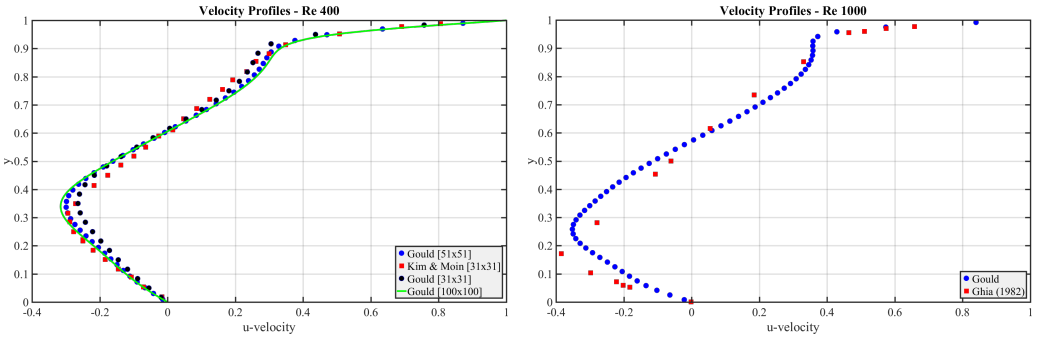
Figure 7: Results - Re 400



(a) u (b) v

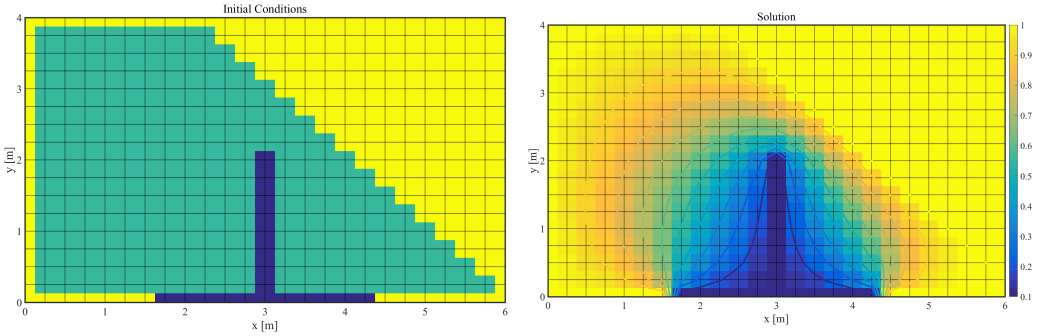
(c) Flow Speed (d) ψ

Figure 8: Results - Re 1000



(a) Re 400 (b) Re400

Figure 9: Comparison of calculated u velocity along cavity centerline



(a) Initial conditions for simulation with barrier (b) Effect of barrier on solution

Figure 10: Modified geometry - shape and barrier