A fast and short Matlab code to solve the lid driven cavity flow problem using the vorticity-stream function formulation

By Joe Molvar

Introduction:

Below you will find the theory and the procedure to a solution to the Navier-Stokes equations for the incompressible lid driven cavity flow problem. A short (50 line) and fast (fully vectorized) Matlab code was written following the procedure outlined. A verification with this code was then performed. Feel free to use or modify this Matlab code.

Equation Derivation:

Shown below as Equation 1, 2, and 3 are the Incompressible Navier-Stokes Equations for 2 Dimensions.

$$\text{X Momentum} \rightarrow \rho\left(\frac{\delta(u)}{\delta t} + u\frac{\delta(u)}{\delta x} + v\frac{\delta(u)}{\delta y}\right) = -\frac{\delta P}{\delta x} + \mu\left(\frac{\delta^2 u}{\delta x^2} + \frac{\delta^2 u}{\delta y^2}\right) + \rho g_x \tag{1}$$

$$\text{Y Momentum} \rightarrow \rho\left(\frac{\delta(v)}{\delta t} + u\frac{\delta(v)}{\delta x} + v\frac{\delta(v)}{\delta y}\right) = -\frac{\delta P}{\delta y} + \mu\left(\frac{\delta^2 v}{\delta x^2} + \frac{\delta^2 v}{\delta y^2}\right) + \rho g_y \tag{2}$$

$$\text{2D continuity} \rightarrow \frac{\delta(u)}{\delta x} + \frac{\delta(v)}{\delta y} = 0 \tag{3}$$

Equation 1 and 2 can be rearranged with algebra and the body forces can be neglected. This is shown below as Equation 4, and 5.

$$\text{X Momentum} \rightarrow \frac{\delta(u)}{\delta t} + u\frac{\delta(u)}{\delta x} + v\frac{\delta(u)}{\delta y} = -\frac{1}{\rho}\frac{\delta P}{\delta x} + \frac{\mu}{\rho}\left(\frac{\delta^2(u)}{\delta x^2} + \frac{\delta^2(u)}{\delta y^2}\right) \tag{4}$$

$$\text{Y Momentum} \rightarrow \frac{\delta(v)}{\delta t} + u\frac{\delta(v)}{\delta x} + v\frac{\delta(v)}{\delta y} = -\frac{1}{\rho}\frac{\delta P}{\delta y} + \frac{\mu}{\rho}\left(\frac{\delta^2(v)}{\delta x^2} + \frac{\delta^2(v)}{\delta y^2}\right) \tag{5}$$

Equation 4 can be differentiated with respect to y. This is shown below as Equation 6.

$$\frac{\delta^2(u)}{\delta y\,\delta t} + \frac{\delta(u)}{\delta y}\frac{\delta(u)}{\delta x} + u\frac{\delta^2(u)}{\delta y\,\delta x} + \frac{\delta(v)}{\delta y}\frac{\delta(u)}{\delta y} + v\frac{\delta^2(v)}{\delta y^2} = -\frac{\delta^2 P}{\delta y\,\delta x}\frac{1}{\rho} + \frac{\mu}{\rho}\left(\frac{\delta^3(u)}{\delta y\,\delta x^2} + \frac{\delta^3(u)}{\delta y^3}\right) \tag{6}$$

Equation 5 can be differentiated with respect to x. This is shown below as Equation 7.

$$\frac{\delta^2(v)}{\delta x\,\delta t} + \frac{\delta(u)}{\delta x}\frac{\delta(v)}{\delta x} + u\frac{\delta^2(v)}{\delta x^2} + \frac{\delta(v)}{\delta x}\frac{\delta(v)}{\delta y} + v\frac{\delta^2(v)}{\delta x\,\delta y} = -\frac{\delta^2 P}{\delta x\,\delta y}\frac{1}{\rho} + \frac{\mu}{\rho}\left(\frac{\delta^3(v)}{\delta x^3} + \frac{\delta^3(v)}{\delta x\,\delta y^2}\right) \tag{7}$$

Equation 6 can be subtracted from Equation 7. The result of this is shown below as Equation 8

$$\left(\frac{\delta^2(v)}{\delta x\,\delta t}-\frac{\delta^2(u)}{\delta y\,\delta t}\right)+\left(u\frac{\delta^2(v)}{\delta x^2}-u\frac{\delta^2(u)}{\delta y\,\delta x}\right)+\left(v\frac{\delta^2(v)}{\delta x\,\delta y}-v\frac{\delta^2(v)}{\delta y^2}\right)+\left(\frac{\delta(u)}{\delta x}\frac{\delta(v)}{\delta x}-\frac{\delta(u)}{\delta y}\frac{\delta(u)}{\delta x}\right)+\cdots$$
$$\left(\frac{\delta(v)}{\delta x}\frac{\delta(v)}{\delta y}-\frac{\delta(v)}{\delta y}\frac{\delta(u)}{\delta y}\right)=\frac{1}{\rho}\left(-\frac{\delta^2 P}{\delta x\,\delta y}--\frac{\delta^2 P}{\delta y\,\delta x}\right)+\frac{\mu}{\rho}\left(\frac{\delta^3(v)}{\delta x^3}-\frac{\delta^3(u)}{\delta y\,\delta x^2}+\frac{\delta^3(v)}{\delta x\,\delta y^2}-\frac{\delta^3(u)}{\delta y^3}\right) \tag{8}$$

With factoring, Equation 8 can be further simplified. This is shown below as Equation 9.

$$\frac{\delta}{\delta t}\left(\frac{\delta(v)}{\delta x}-\frac{\delta(u)}{\delta y}\right)+u\frac{\delta}{\delta x}\left(\frac{\delta(v)}{\delta x}-\frac{\delta(u)}{\delta y}\right)+v\frac{\delta}{\delta y}\left(\frac{\delta(v)}{\delta x}-\frac{\delta(u)}{\delta y}\right)+\cdots$$
$$\left(\frac{\delta(u)}{\delta x}+\frac{\delta(v)}{\delta y}\right)\left(\frac{\delta(v)}{\delta x}-\frac{\delta(u)}{\delta y}\right)=\frac{\mu}{\rho}\left(\frac{\delta^2}{\delta x^2}\left(\frac{\delta(v)}{\delta x}-\frac{\delta(u)}{\delta y}\right)+\frac{\delta^2}{\delta y^2}\left(\frac{\delta(v)}{\delta x}-\frac{\delta(u)}{\delta y}\right)\right) \tag{9}$$

Vorticity can be defined. This definition is shown below as Equation 10.

$$\text{Vorticity Defined} \rightarrow \zeta \equiv \frac{\delta(v)}{\delta x}-\frac{\delta(u)}{\delta y} \tag{10}$$

From taking Equation 9, and substituting in Equation 10, and Equation 3, the Vorticity Transport Equation can be created. This equation is shown below as Equation 11.

$$\text{Vorticity Transport Equation} \rightarrow \frac{\delta(\zeta)}{\delta t}+u\frac{\delta(\zeta)}{\delta x}+v\frac{\delta(\zeta)}{\delta y}=\frac{\mu}{\rho}\left(\frac{\delta^2(\zeta)}{\delta x^2}+\frac{\delta^2(\zeta)}{\delta y^2}\right) \tag{11}$$

Next the stream function Ψ can be defined and related to u and v velocity. These relations are shown below as Equation 12.

$$\frac{\delta\Psi}{\delta y}\equiv u;\quad \frac{\delta\Psi}{\delta x}\equiv -v \tag{12}$$

Using Equation 12 and Equation 10 the updated vorticity transport equation can be created. This is shown below as Equation 13.

$$\text{Vorticity Transport Equation Updated} \rightarrow \frac{\delta(\zeta)}{\delta t}=-\frac{\delta\Psi}{\delta y}\frac{\delta(\zeta)}{\delta x}+\frac{\delta\Psi}{\delta x}\frac{\delta(\zeta)}{\delta y}+\frac{\mu}{\rho}\left(\frac{\delta^2(\zeta)}{\delta x^2}+\frac{\delta^2(\zeta)}{\delta y^2}\right) \tag{13}$$

Next the relations in Equation 12, can be substituted into Equation 10 and a new equation, Equation 14 can be created for the Vorticity. This equation is known as the elliptical vorticity equation.

$$\text{Elliptical Vorticity Equation} \rightarrow -\zeta = -\frac{\delta\left(-\frac{\delta\Psi}{\delta x}\right)}{\delta x} + \frac{\delta\left(\frac{\delta\Psi}{\delta y}\right)}{\delta y} = \frac{\delta^2(\Psi)}{\delta x^2} + \frac{\delta^2(\Psi)}{\delta y^2} \tag{14}$$

Equation 13 and 14 are the two equations to be used for this analysis.

Discretization of Primary Equations:

The Vorticity transport equation (Equation 13) can be discretized. This is shown below as Equation 15.

$$\text{Vorticity Transport Equation Discritized} \rightarrow \left(\frac{\zeta_{i,j}^{new} - \zeta_{i,j}}{dt}\right) = -\left(\frac{\Psi_{i,j+1} - \Psi_{i,j-1}}{2dy}\right)\left(\frac{\zeta_{i+1,j} - \zeta_{i-1,j}}{2dx}\right) + \cdots$$
$$\left(\frac{\Psi_{i+1,j} - \Psi_{i-1,j}}{2dx}\right)\left(\frac{\zeta_{i,j+1} - \zeta_{i,j-1}}{2dy}\right) + \frac{\mu}{\rho}\left(\frac{\zeta_{i+1,j} - 2\zeta_{i,j} \pm \zeta_{i-1,j}}{dx^2} + \frac{\zeta_{i,j+1} - 2\zeta_{i,j} \pm \zeta_{i,j-1}}{dy^2}\right) \tag{15}$$

Equation 15 can be rearranged using algebra so that the new vorticity can be solved for the next step. This equation is shown below as Equation 16.

$$\text{Algebra} \rightarrow \zeta_{i,j}^{new} = \zeta_{i,j} + \begin{bmatrix} -\left(\frac{\Psi_{i,j+1} - \Psi_{i,j-1}}{2h}\right)\left(\frac{\zeta_{i+1,j} - \zeta_{i-1,j}}{2h}\right) + \left(\frac{\Psi_{i+1,j} - \Psi_{i-1,j}}{2h}\right)\left(\frac{\zeta_{i,j+1} - \zeta_{i,j-1}}{2h}\right) + \cdots \\ \frac{\mu}{\rho}\left(\frac{\zeta_{i+1,j} + \zeta_{i,j+1} - 4\zeta_{i,j} + \zeta_{i-1,j} + \zeta_{i,j-1}}{h^2}\right) \end{bmatrix} dt \tag{16}$$

The elliptical vorticity equation (Equation 14) can also be discretized. This is shown below as Equation 17.

$$\text{Elipitical Vorticity Equation Discritized} \rightarrow -\zeta_{i,j} = \frac{\Psi_{i+1,j} - 2\Psi_{i,j} + \Psi_{i-1,j}}{dx^2} + \frac{\Psi_{i,j+1} - 2\Psi_{i,j} + \Psi_{i,j-1}}{dy^2} \tag{17}$$

Equation 17 can be rearranged, and the stream function can be solved for. This is shown below as Equation 18.

$$\text{Solved for Stream Function} \rightarrow \Psi_{i,j} = \frac{\zeta_{i,j}h^2 + \Psi_{i+1,j} + \Psi_{i,j+1} + \Psi_{i,j-1} + \Psi_{i-1,j}}{4} \tag{18}$$

Boundary Conditions:

Boundary conditions must be implemented. The boundary conditions for one common CFD case, the Lid Driven cavity flow problem are used in this case.

The conditions for the lid driven cavity are shown below in terms of velocity.

$$\text{Top} \rightarrow u_{(:,N)} = \text{Uwall}; \quad v_{(:,N)} = 0$$

$$\text{Bottom} \rightarrow u_{(:,1)} = 0; \quad v_{(:,1)} = 0$$

$$\text{Left} \rightarrow u_{(1,:)} = 0; \quad v_{(1,:)} = 0$$

$$\text{Right} \rightarrow u_{(N,:)} = 0; \quad v_{(N,:)} = 0$$

At boundaries stream function is set to zero. This can be shown in adjusted form of Equation 10 below for each boundary. The non-normal component of Equation 10 is set to zero in the relations detailed below as Equation 18 through 21.

$$\text{Top} \rightarrow -\zeta_{(:,N)} = \frac{\delta^2(\Psi)}{\delta y^2}_{(:,N)} \tag{18}$$

$$\text{Bottom} \rightarrow -\zeta_{(:,1)} = \frac{\delta^2(\Psi)}{\delta y^2}_{(:,1)} \tag{19}$$

$$\text{Left} \rightarrow -\zeta_{(1,:)} = \frac{\delta^2(\Psi)}{\delta x^2}_{(1,:)} \tag{20}$$

$$\text{Right} \rightarrow -\zeta_{(N,:)} = \frac{\delta^2(\Psi)}{\delta x^2}_{(N,:)} \tag{21}$$

Shown below as Equation 22 through 25is Taylor Series Expansion for each adjacent boundary row or column.

$$\text{Top} \rightarrow \Psi_{(:,N-1)} \equiv \Psi_{(:,N)} - \frac{\delta\Psi}{\delta y}_{(:,N)} dy + \frac{\delta^2(\Psi)}{\delta y^2}_{(:,N)} \frac{dy^2}{2} \ldots \tag{22}$$

$$\text{Bottom} \rightarrow \Psi_{(:,2)} \equiv \Psi_{(:,1)} + \frac{\delta\Psi}{\delta y}_{(:,1)} dy + \frac{\delta^2(\Psi)}{\delta y^2}_{(:,1)} \frac{dy^2}{2} \ldots \tag{23}$$

$$\text{Left} \rightarrow \Psi_{(2,:)} \equiv \Psi_{(1,:)} + \frac{\delta\Psi}{\delta x}_{(1,:)} dx + \frac{\delta^2(\Psi)}{\delta x^2}_{(1,:)} \frac{dx^2}{2} \ldots \tag{24}$$

$$\text{Right} \rightarrow \Psi_{(N-1,:)} \equiv \Psi_{(N,:)} - \frac{\delta\Psi}{\delta x}_{(N,:)} dx + \frac{\delta^2(\Psi)}{\delta x^2}_{(N,:)} \frac{dx^2}{2} \ldots \tag{25}$$

Equations 12 and 18 through 21 can be substituted into Equation 22 through 25. Also, the boundary stream function is set to zero. Equations are then algebraically manipulated. These new equations are shown below as Equation 26 through 29.

$$\text{Top} \rightarrow \Psi_{(:,N-1)} = -u_{(:,N)}dy - \zeta_{(:,N)}\frac{dy^2}{2} \rightarrow \zeta_{(:,N)}\frac{dy^2}{2} = -\Psi_{(:,N-1)} - u_{(:,N)} * dy \tag{26}$$

$$\text{Bottom} \rightarrow \Psi_{(:,2)} = u_{(:,1)}dy - \zeta_{(:,1)}\frac{dy^2}{2} \rightarrow \zeta_{(:,1)}\frac{dy^2}{2} = -\Psi_{(:,2)} + u_{(:,1)} * dy \tag{27}$$

$$\text{Left} \rightarrow \Psi_{(2,:)} = -v_{(1,:)}dx - \zeta_{(1,:)}\frac{dx^2}{2} \rightarrow \zeta_{(1,:)}\frac{dx^2}{2} = -\Psi_{(2,:)} - v_{(1,:)} * dx \tag{28}$$

$$\text{Right} \rightarrow \Psi_{(N-1,:)} = v_{(N,:)}dx - \zeta_{(N,:)}\frac{dx^2}{2} \rightarrow \zeta_{(N,:)}\frac{dx^2}{2} = -\Psi_{(N-1,:)} + v_{(N,:)} * dx \tag{29}$$

After continued algebra Equation 26 through 29 can be put into their final boundary condition form, with the boundary velocities accounted for. These final boundary equations are shown below as Equation 30 through 33.

$$\text{Top} \rightarrow \zeta_{(:,N)} = \frac{-2\Psi_{(:,N-1)}}{dy^2} - \frac{Uwall * 2}{dy} \tag{30}$$

$$\text{Bottom} \rightarrow \zeta_{(:,1)} = \frac{-2\Psi_{(:,2)}}{dy^2} \tag{31}$$

$$\text{Left} \rightarrow \zeta_{(1,:)} = \frac{-2\Psi_{(2,:)}}{dx^2} \tag{32}$$

$$\text{Right} \rightarrow \zeta_{(N,:)} = \frac{-2\Psi_{(N-1,:)}}{dx^2} \tag{33}$$

Create Velocities from Stream Function:

From the relations in Equation 12, the following expressions can be created to obtain the u and v velocities. These expressions are detailed as Equation 34 and 35.

$$u = \frac{\delta\Psi}{\delta y} = \frac{\Psi_{i,j+1} - \Psi_{i,j-1}}{2h} \tag{34}$$

$$v = -\frac{\delta\Psi}{\delta x} = \frac{-\Psi_{i+1,j} + \Psi_{i-1,j}}{2h} \tag{35}$$

Matlab Approach:

The steps I used to solve this Lid Driven Cavity problem in Matlab are shown below.

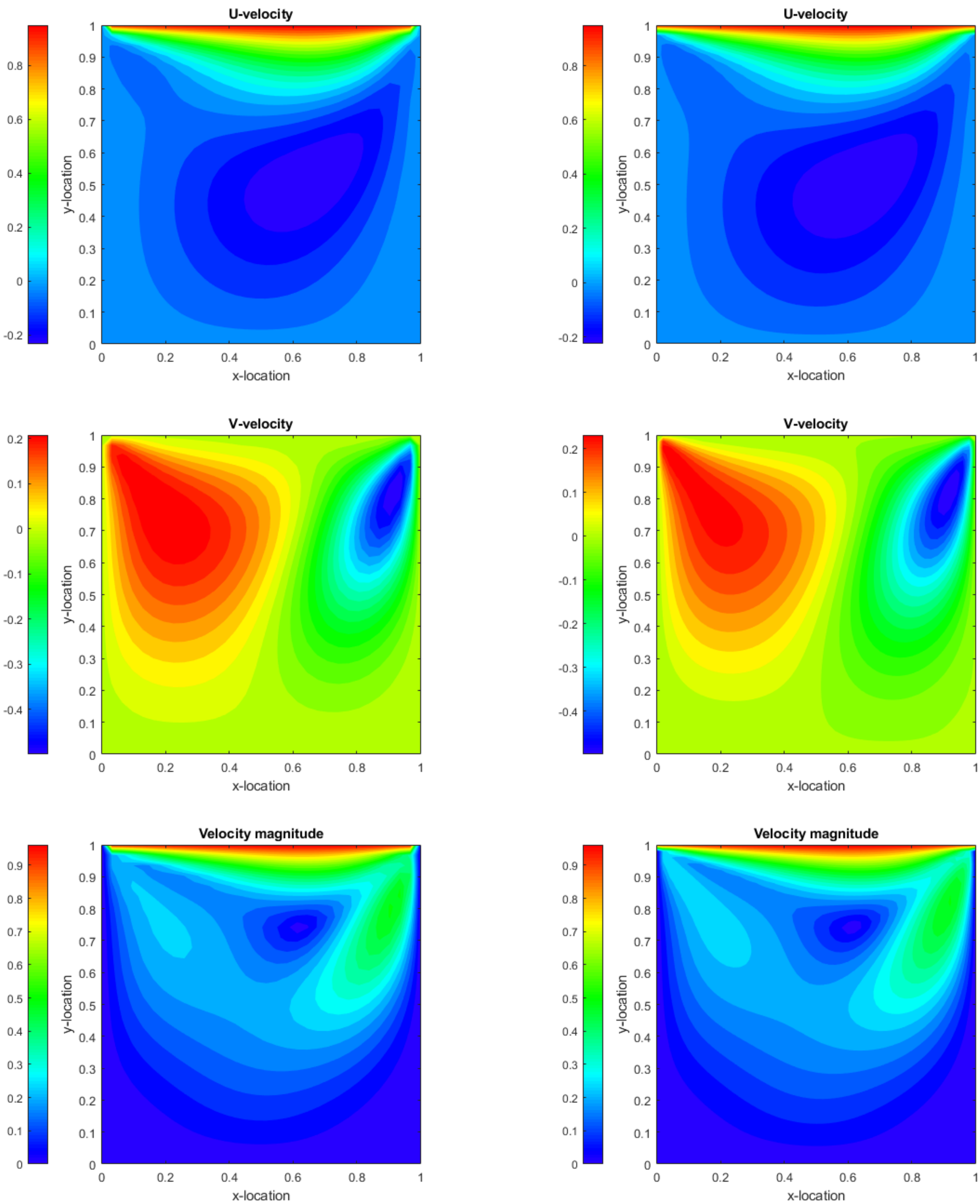
1. Givens (Nodes, Domain Size, Velocity, Density, Dynamic Viscosity, Time Step, Max iter, Max error)
2. Setup 1D Grid
3. Prelocate Matrixes
4. Solve Loop
    I. Create Boundary Conditions (Equation 30 to 33)
    II. Partially Solve Vorticity Transport Equation (Equation 16)
    III. Partially Solve Elliptical Vorticity Equation for Stream Function (Equation 18)
    IV. Check for Convergence (Difference in Max Vorticity)
5. Create Velocities from Stream Function (Equation 34 and 35)
6. Plots

I solve Equation 16 and 18 using the solve loop based around the change in time, to partially solve the vorticity transport equation for every loop. I then partially solve the elliptical vorticity equation for every loop. This partial solve method I developed is similar to the Gauss-Seidel Method. With more iterations the difference in each loop's solution for the vorticity and the stream function become less. Convergence is approached for both the Vorticity Transport equation and Elliptical Vorticity equation. Eventually the max change in vorticity from the previous iteration compared with current iteration becomes less than the max error factor and the code exits the solve loop. Next velocity components are created from the stream function. I wrote the code to be fully vectorized, for faster performance.

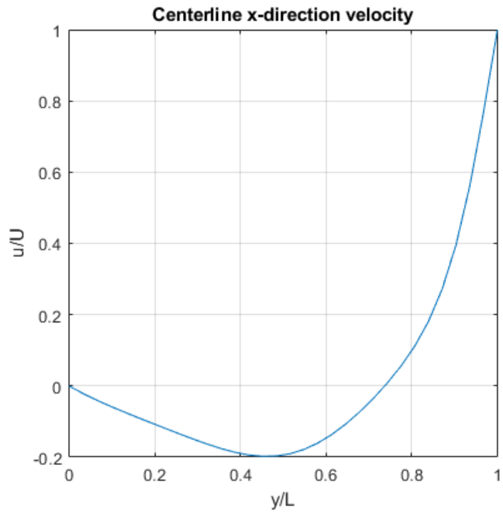
Verification:

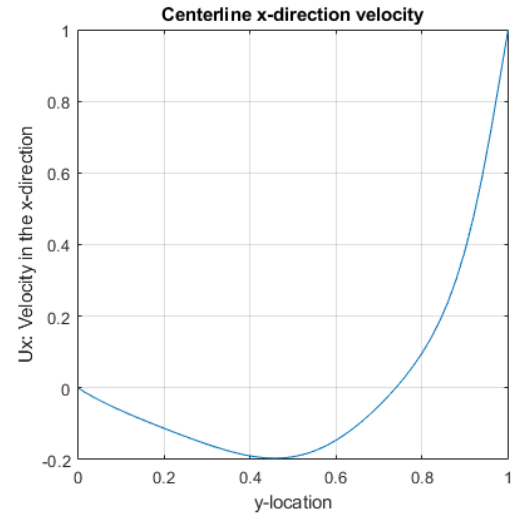
- 32x32 Nodes
- Density = 1; Dynamic Viscosity = 0.01; Corresponding to Reynolds Number of 100

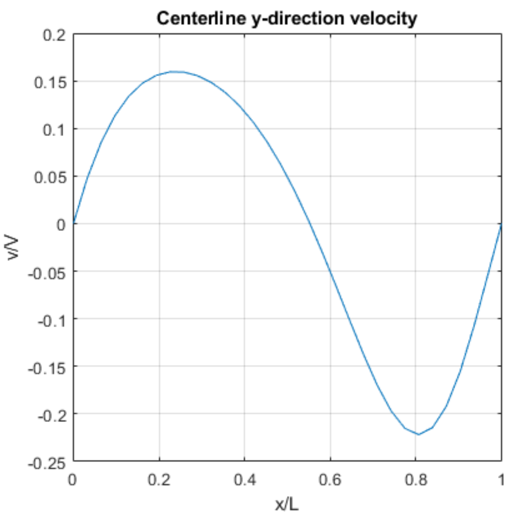
Below on the left is results from my code, and on the right are results from the code written by Benjamin Seibold during his time at MIT. [1] Note the attached code has reduced plotting.

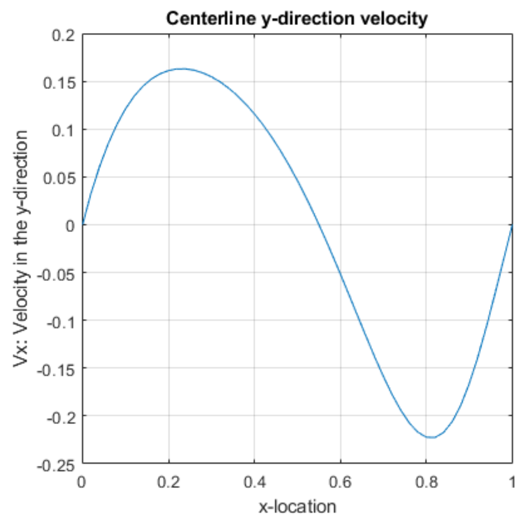


U-velocity
U-velocity
V-velocity
V-velocity
Velocity magnitude
Velocity magnitude
x-location
y-location

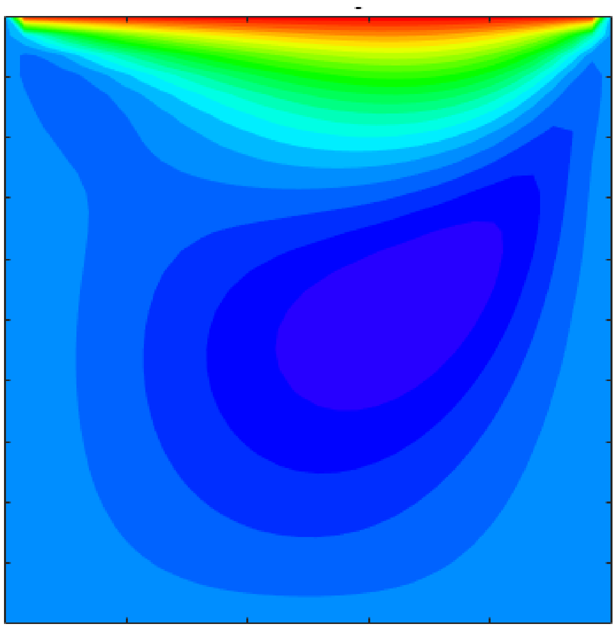


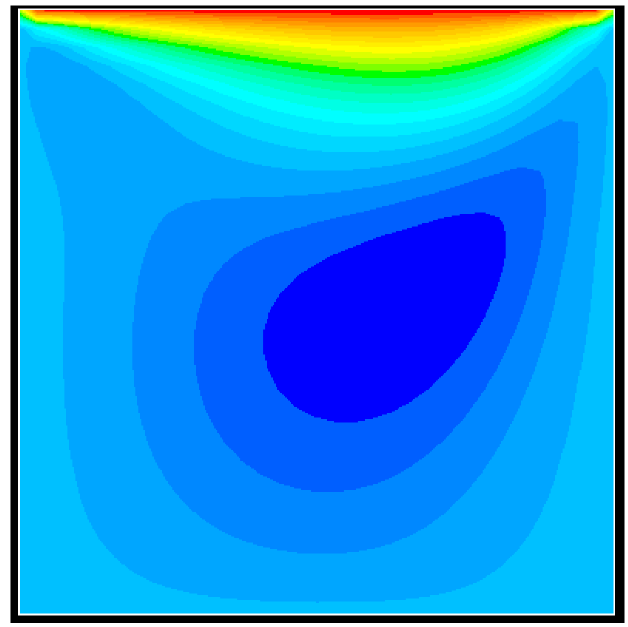


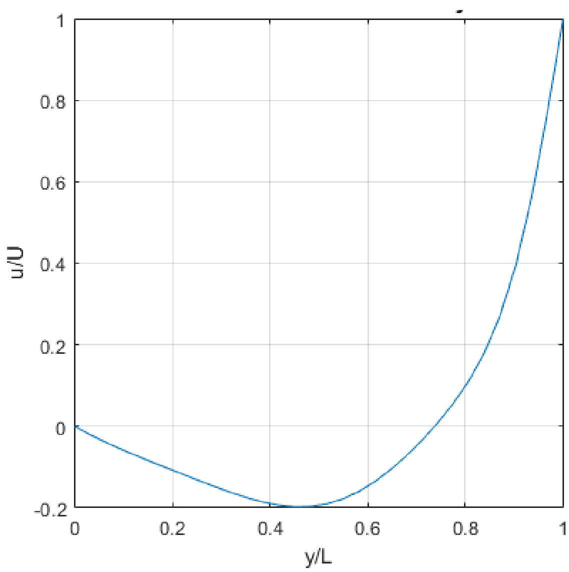


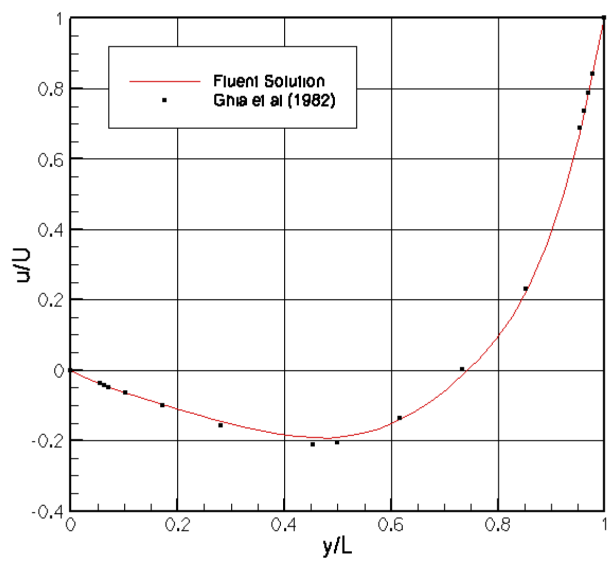


As an additional check, I compared my code results with the results on the CFD wiki for a lid driven cavity flow problem [2]. My code results left, CFD wiki results right below.









From the comparisons it is safe to say that the code I wrote creates an accurate approximation for the fluid flow for the lid driven cavity flow problem.

References:

[1] http://math.mit.edu/~gs/cse/codes/mit18086_navierstokes.pdf

[2] https://www.cfd-online.com/Wiki/Lid-driven_cavity_problem

Appendix: Matlab Code

```matlab
%%  VORTICITY/STREAM FUNCTION LID DRIVEN CAVITY FLOW SOLVER   JOE MOLVAR
clear;  close all
%%% GIVENS
Nx = 32;  L = 1; Wall_Velocity = 1;       % Nodes X; Domain Size; Velocity
rho = 1;   mu = 0.01;                      % Density; Dynamic Viscosity;
dt = 0.001;  maxIt = 50000;  maxe = 1e-7; % Time Step; Max iter; Max error
%%% SETUP 1D GRID
Ny = Nx;  h=L/(Nx-1);  x = 0:h:L;  y = 0:h:L;
im = 1:Nx-2; i = 2:Nx-1; ip = 3:Nx;  jm = 1:Ny-2; j = 2:Ny-1; jp = 3:Ny;
%%% PRELOCATE MATRIXES
Vo = zeros(Nx,Ny);  St = Vo;  Vop = Vo;  u = Vo;  v = Vo;
%%% SOLVE LOOP SIMILAR TO GAUSS-SIEDEL METHOD
for iter = 1:maxIt
    %%% CREATE BOUNDARY CONDITIONS
    Vo(1:Nx,Ny) = -2*St(1:Nx,Ny-1)/(h^2) - Wall_Velocity*2/h; % Top
    Vo(1:Nx,1)  = -2*St(1:Nx,2)   /(h^2);                     % Bottom
    Vo(1,1:Ny)  = -2*St(2,1:Ny)   /(h^2);                     % Left
    Vo(Nx,1:Ny) = -2*St(Nx-1,1:Ny)/(h^2);                     % Right
    %%% PARTIALLY SOLVE VORTICITY TRANSPORT EQUATION
    Vop = Vo;
    Vo(i,j) = Vop(i,j) + ...
        (-1*(St(i,jp)-St(i,jm))/(2*h) .* (Vop(ip,i)-Vop(im,j))/(2*h)+...
            (St(ip,j)-St(im,j))/(2*h) .* (Vop(i,jp)-Vop(i,jm))/(2*h)+...
     mu/rho*(Vop(ip,j)+Vop(im,j)-4*Vop(i,j)+Vop(i,jp)+Vop(i,jm))/(h^2))*dt;
    %%% PARTIALLY SOLVE ELLIPTICAL VORTICITY EQUATION FOR STREAM FUNCTION
    St(i,j) = (Vo(i,j)*h^2 + St(ip,j) + St(i,jp) + St(i,jm) + St(im,j))/4;
    %%% CHECK FOR CONVERGENCE
    if iter > 10
        error = max(max(Vo - Vop))
        if error < maxe
            break;
        end
    end
 end
%%% CREATE VELOCITY FROM STREAM FUNCTION
u(2:Nx-1,Ny) = Wall_Velocity;
u(i,j) = (St(i,jp)-St(i,jm))/(2*h);  v(i,j) = (-St(ip,j)+St(im,j))/(2*h);
%%% PLOTS
cm = hsv(ceil(100/0.7));  cm = flipud(cm(1:100,:));
figure(1);  contourf(x,y,u',23,'LineColor','none');
title('U-velocity');  xlabel('x-location');  ylabel('y-location')
axis('equal',[0 L 0 L]);  colormap(cm);  colorbar('westoutside');
figure(2);  plot(y,u(round(Ny/2),:));
title('Centerline x-direction velocity');
xlabel('y/L');  ylabel('u/U');  axis('square');  xlim([0 L]);  grid on
N = 1000;  xstart = max(x)*rand(N,1);  ystart = max(y)*rand(N,1);
[X,Y] = meshgrid(x,y);
figure(3);  h=streamline(X,Y,u',v',xstart,ystart,[0.1, 200]);
title('Stream Function');  xlabel('x-location');  ylabel('y-location')
axis('equal',[0 L 0 L]);  set(h,'color','k')
```