Project 4: Navier-Stokes Solution to Driven Cavity and Channel Flow Conditions

R. S. Sellers

MAE 5440, Computational Fluid Dynamics

Utah State University, Department of Mechanical and Aerospace Engineering

The solution of the Navier-Stokes equation in the case of flow in a driven cavity and between parallel plates is developed within this paper. To solve the Navier-Stokes equation the SIMPLE algorithm is employed.

I. Introduction

Two sets of geometry with similar boundary conditions are given in the problem statement.

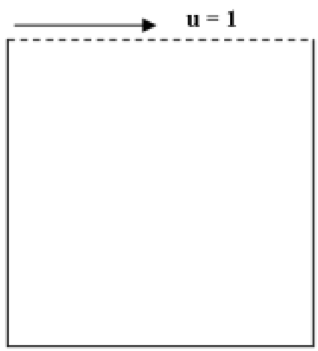

Figure1 . Driven cavity.

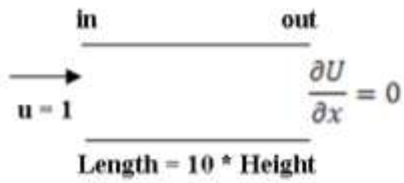

Figure 2. Flow between parallel plates.

A correct solution to the driven cavity problem will show vortical flow within the square cavity. Typical parabolic u-velocity distribution will be seen in a correct solution of the parallel plate flow geometry.

Both geometries assume constant property flow, $\mu = .01$ and $\rho = 1$.

II. SIMPLE Algorithm

The SIMPLE algorithm or **s**emi-**i**mplicit **m**ethod for **p**ressure-**l**inked **e**quations is a routine used to solve the pressure linked nonlinear momentum equations. In the case of incompressible fluids, a correct pressure field applied to the momentum equations should satisfy equilibrium.

Implementation is a process of guess-and-check. Steps are listed below.

1. Compute uncorrected (does not satisfy continuity) mass flux at the face of each velocity control volume.
2. Compute an intermediate velocity field via solving the discretized momentum equation.
3. Solve the pressure correction equation for each control volume.
4. Update pressure field.
5. Correct cell velocities to satisfy continuity via the gradient of the pressure correction.
6. Correct any other scalar variables.
7. Return to step 1 until convergence.

A more thorough, albeit brief, explanation of the algorithm can be seen in the following sections.

III. Discretization

A. Staggered Grid

Grid staggering is the process of purposefully misaligning the pressure & velocity control volumes in order to prevent a non-physical "checker board" pressure distribution. This project employs back staggering notation, seen in figure 3.

The main difficulty in writing the code for this project was to eliminate typos created by the sometimes confusing notation required for grid staggering.

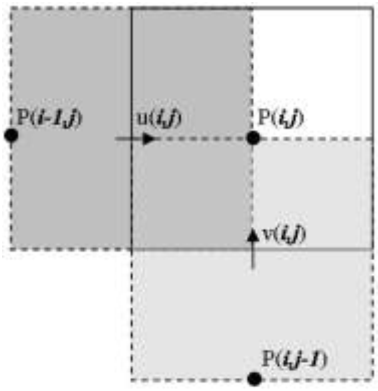

Figure 3. The misaligned pressure, u-momentum & v-momentum control volumes.

B. Momentum Terms

When the velocity field is not known or given, a solution to the momentum transport equations must be found. The three governing equations of steady laminar flow are given as:

$$\frac{\partial}{\partial x}(\rho uu) + \frac{\partial}{\partial y}(\rho vu) = \frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right) - \frac{\partial p}{\partial x} + S_u \tag{3.1}$$

$$\frac{\partial}{\partial x}(\rho uv) + \frac{\partial}{\partial y}(\rho vv) = \frac{\partial}{\partial x}\left(\mu\frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu\frac{\partial v}{\partial y}\right) - \frac{\partial p}{\partial y} + S_v \tag{3.2}$$

$$\frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0 \tag{3.3}$$

Difficulty in solving these equations lies in handling the nonlinear terms present in the first two equations, as well as the treatment of the u, v, p terms present in all three equations. The reader is assumed to be familiar with the SIMPLE algorithm as the following derivation is meant to give a brief overview of the process and implementation. For a rigorous derivation see chapter 6 of *Introduction to Computational Fluid Dynamics, Versteeg & Malalasekera, 2nd edition.*

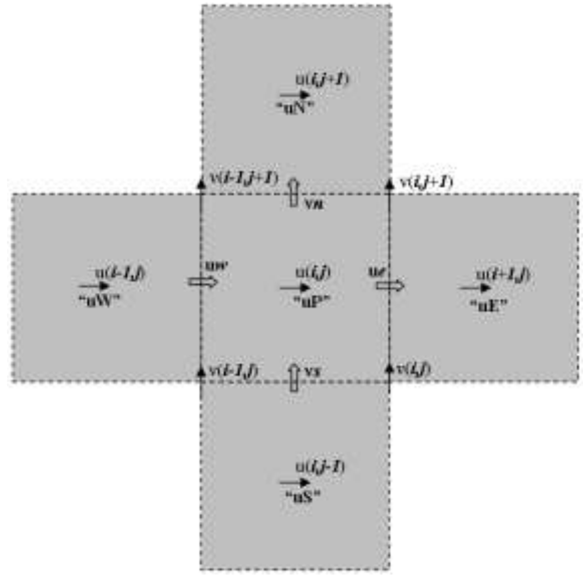

Figure 4. A collection of 5 u-momentum control volumes demonstrating the notation used in this derivation.

The discretization process will be developed for u-momentum equation (3.1). A similar process can be done for v-momentum equation (3.2). The left hand side is integrated over a *u* control volume in figure 4 and the right hand side diffusion and pressure terms are discretized.

Source terms in (3.1) are dropped for constant property flows. Equation 3.1 can be rewritten as:

$$u_e\dot{m}_e - u_w\dot{m}_w + u_n\dot{m}_n - u_s\dot{m}_s = (P_w - P_e)\Delta y + \mu\frac{u_E - u_P}{(\delta x)_e}\Delta y - \mu\frac{u_P - u_W}{(\delta x)_w}\Delta y + \mu\frac{u_N - u_P}{(\delta y)_n}\Delta x + \mu\frac{u_P - u_S}{(\delta y)_s}\Delta x \tag{3.5}$$

To facilitate an under relaxation scheme to find an iterative solution to the u-momentum equation (which do not yet satisfy continuity) the u_P terms are grouped together, and the convective flux terms are combined with diffusive conductance terms into "A" terms. $max[...]$ represents application of upwind differencing to the convective terms.

$$A_E^U = max[-\dot{m}_e, 0] + \mu\frac{\Delta y}{(\delta x)_e} \tag{3.5a}$$

$$A_N^U = max[-\dot{m}_n, 0] + \mu\frac{\Delta x}{(\delta y)_n} \tag{3.5b}$$

$$A_W^U = max[\dot{m}_w, 0] + \mu \frac{\Delta y}{(\delta x)_w} \quad (3.5c)$$

$$A_S^U = max[\dot{m}_s, 0] + \mu \frac{\Delta x}{(\delta y)_s} \quad (3.5d)$$

$$A_P^U = A_E^U + A_W^U + A_N^U + A_S^U \quad (3.5e)$$

$$\dot{m}_e = (\rho u)_e \Delta y \quad (3.6a)$$

$$\dot{m}_w = (\rho u)_w \Delta y \quad (3.6b)$$

$$\dot{m}_s = (\rho v)_s \Delta x \quad (3.6c)$$

$$\dot{m}_n = (\rho v)_n \Delta x \quad (3.6d)$$

Mass flow terms (3.5) across the control volume face are interpolated from control volume center values, for example:

$$u_e = \frac{u_P + u_E}{2} = \frac{u_{i,j} + u_{i+1,j}}{2}$$

Finally, a SOR type iteration scheme can be used and the final u-velocity term can be solved for and iterated on.

$$u_P = (1 - \Omega)u_P^{old} + \frac{\Omega}{A_P^U}[A_E^U u_E + A_W^U u_W + A_S^U u_S + A_N^U u_N + (P_w - P_e)\Delta y] \quad (3.6)$$

$$\Omega \cong .5$$

C. Pressure Terms

Since there is no transport equation similar to the u or v momentum equations for pressure, return to the integrated continuity equation and assume u, v, p terms are made up of a guessed and corrected value.

$$[(\rho u)_e - (\rho u)_w]\Delta y + [(\rho v)_n - (\rho v)_s]\Delta x = 0 \quad (3.7)$$

$$u = u_{guess}^* + u'_{correction} \quad (3.8a)$$

$$v = v_{guess}^* + v'_{correction} \quad (3.8b)$$

$$p = p_{guess}^* + p'_{correction} \quad (3.8c)$$

From here the u, v correction values can be seen to be in terms of pressure correction values. A key approximation made by the SIMPLE algorithm when finding correction values at the center of a control volume is the omission of surrounding node correction values.

$$u'_w = \frac{\Delta y}{A_W^U}(p'_W - p'_P) \quad (3.9a)$$

$$u'_e = \frac{\Delta y}{A_E^U}(p'_P - p'_E) \quad (3.9b)$$

$$v'_n = \frac{\Delta x}{A_N^V}(p'_P - p'_N) \quad (3.9c)$$

$$v'_S = \frac{\Delta x}{A_S^V}(p'_S - p'_P) \quad (3.9d)$$

To create notation which is clearer, "a" values are created.

$$a_E = \frac{\rho \Delta y^2}{A_E^U} \quad (3.10a)$$

$$a_W = \frac{\rho \Delta y^2}{A_W^U} \quad (3.10b)$$

$$a_N = \frac{\rho \Delta x^2}{A_N^V} \quad (3.10c)$$

$$a_S = \frac{\rho \Delta x^2}{A_S^V} \quad (3.10d)$$

$$a_P = a_E + a_W + a_N + a_S \quad (3.11)$$

The correction equations (3.8) and the u, v correction values (3.9) can be plugged back into the integrated continuity equation (3.7) and the "a" values (3.10) can be factored out. Solve for p'_P via SOR scheme and the resulting pressure correction is:

$$p'_P = p'_P + \frac{\Omega_P}{a_P}(a_E p'_E + a_W p'_W + a_N p'_N + a_S p'_S - S - a_P p'_P) \quad (3.12)$$

$$\Omega_P \cong 1.5$$

$$S = [\rho u_e - \rho u_w]\Delta y + [\rho v_n - \rho v_s]\Delta x \quad (3.13)$$

Following the steps laid out in section II, p'_P should be solved after the non corrected momentum equations (3.6) are solved. Once p'_P has been reasonably converged, apply the corrections (3.8) and (3.9) to the velocity and pressure terms. Correction of the pressure term (3.8c) should be under relaxed.

The source term, S (3.13) is crucial for determining when the solution is converged. Once the source term calculated before the velocity corrections is reasonably close to zero, it can be assumed that the problem has converged and satisfies continuity over the whole mesh.

IV. Boundary Conditions

A. Driven Cavity

It is known that the u and v velocities will always be zero on the three wall boundaries present in the driven cavity, therefore it is necessary to only iterate on the interior nodes. Care must be taken to select the correct interior nodes as it depends on the staggering scheme used.

The combined diffusion and convection terms in (3.5) must be scaled along wall nodes because the δx or δy must be divided by two when solving the v & u momentum equations, respectively.

The u = 1 boundary is simple to incorporate by initializing the top row of the u matrix to 1.

In both problems, the $a_E$and $a_W$values are set to zero on east and west faces and the $a_N$and $a_S$values are set to zero on north and south faces.

B. Channel Flow

The u=1 inflow condition is easy to implement by initializing the left side of the u-momentum matrix. The $\frac{\partial u}{\partial x}$ boundary requires a more consideration if continuity is to be satisfied. The following steps will enable this boundary condition and cause it to satisfy continuity (based on the grid notation used in project 4 code).

$$u(imax+1,j) = u(imax,j)$$

$$\dot{m}_{in} = \frac{\Delta y(u(imax-2,j)+u(imax-1,j))}{2}$$

$$\dot{m}_{out} = \frac{\Delta y(u(imax,j)+u(imax+1,j))}{2}$$

$$u(imax,j) = \frac{\dot{m}_{in}}{\dot{m}_{out}} u(imax,j)$$

The scaling of the diffusion and convection terms (3.5) along walls are the same in the channel flow problem as in the driven cavity problem.

V. Solution

A. Convergence Criteria

Convergence is determined by monitoring the source value, S. Once the source term is below 1e-10 before velocity corrections are made, the iterative loop will exit. In other words:

$$sqrt(sum(sum(S_{before}.*S_{before}))) < 1e^{-10}$$

B. Flux calculation

Flux calculations along the top surface of the driven cavity problem are made using a second order accurate central difference scheme. See p. 278 of *Versteeg* for more details.

$$Flux = \sum_{k=1}^{N-1} -\mu \frac{(u_{k,N}-1)\Delta x}{\frac{1}{2}\Delta y}$$

VI. Results

A. Driven Cavity

The following diagrams are comparisons between FLUENT generated contour plots of u-momentum and Matlab generated u-momentum plots. See appendix A for more contour plots and momentum vector plots.

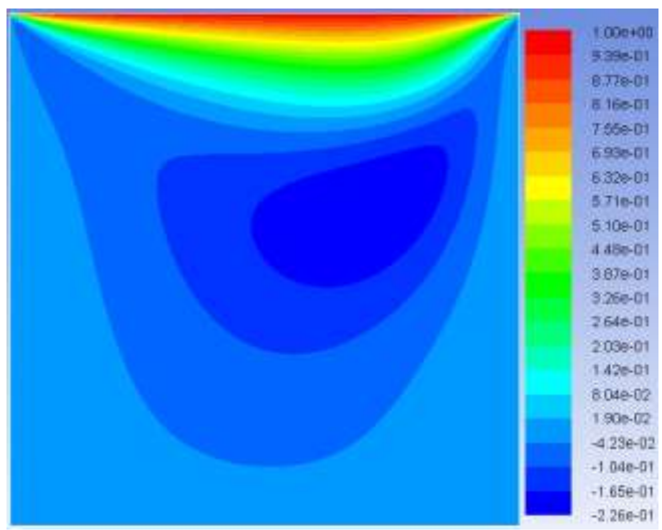

Figure 5. FLUENT contour plot of u-momentum on a 200x200 mesh.

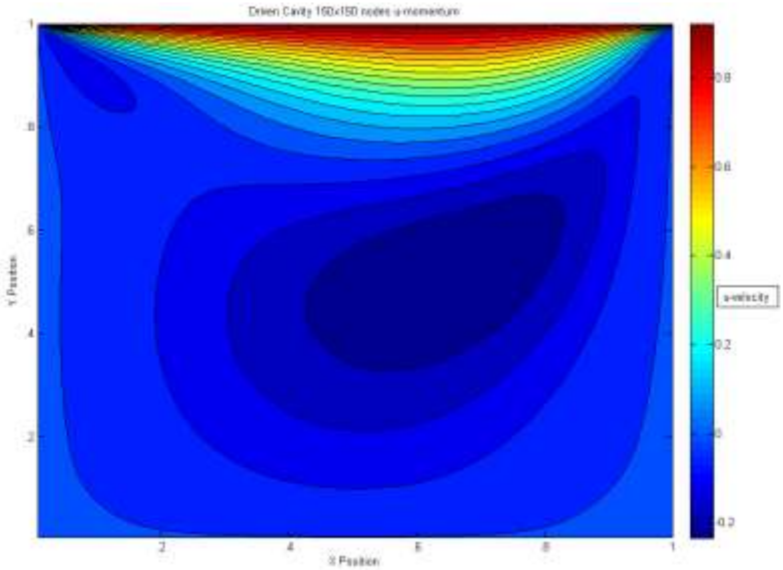

Figure 6. Matlab generated plot of u-momentum on a 150x150 mesh.

Both plots are practically identical. The vortex has the same general shape and position in both charts. The Matlab plot uses more contours and has a coarser grid, which could be the reason for the slight discrepancies between the two.

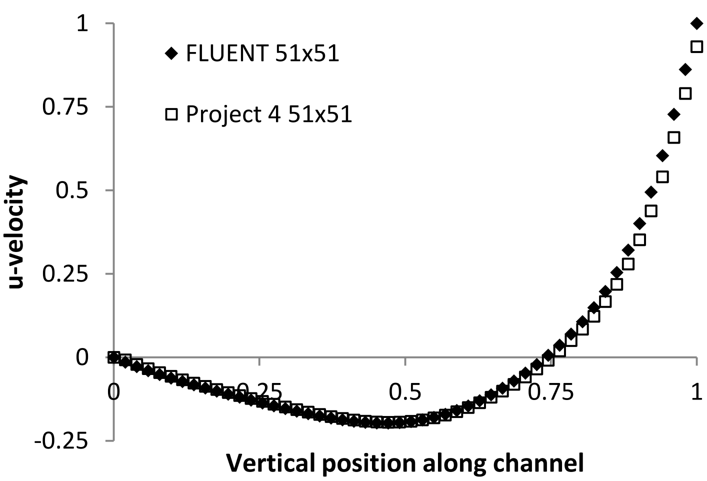

Figure 7. A comparison of the center line u-velocity computed with Matlab and FLUENT.

On a 51x51 grid the centerline of both solution methods lay on top of each other – further proof of a correct implementation of the SIMPLE algorithm as applied to a driven cavity.

Mesh size:	50	100	150
FLUENT drag:	0.188	0.215	0.233
Matlab drag:	0.1763	0.198	0.22

Drag forces computed on variable mesh sizes can be seen above. The Matlab computed drag forces are slightly lower than FLUENT calculations, which is to be expected based off discussions held in class.

B. Channel Flow

Below are contour plots for channel flow calculated in Matlab and in FLUENT. The cross section of the u-velocity in an ideal channel flow solution will have the expected laminar parabolic shape with the maximum velocity being 1.5*u-wall.

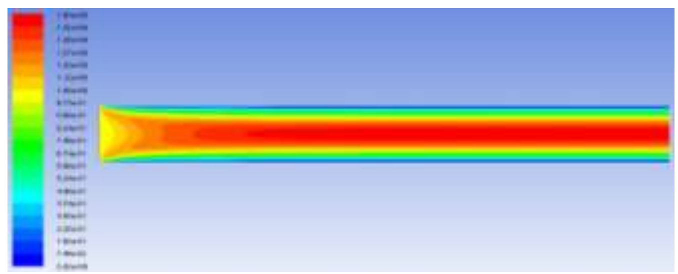

Figure 8. FLUENT generated plot on a fine grid.

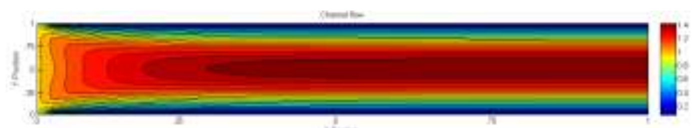

Figure 9. Matlab generated plot of the channel flow problem.

Both channel flow contour plots are nearly identical, leading to the assurance that the algorithm was implemented correctly and boundary conditions were applied correctly.

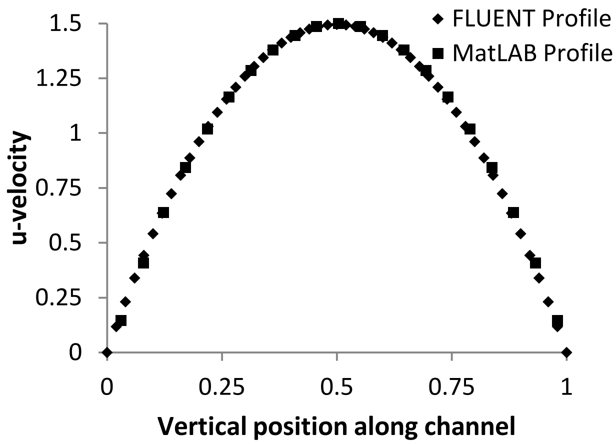

Figure10 . A comparison of velocity curves. Maximum for FLUENT = 1.498, maximum for Matlab = 1.4989

The velocity profile of the Matlab generated plot lays on the top of the FLUENT profile, further proof of a correct implementation. Both were generated on odd grid spacing to ensure the absolute highest velocity was recorded.

VII. Summary

Overall, my results seem to fit very well with the accepted values of FLUENT. In both cases, the Matlab results fit the contour plots as well as the line plots of FLUENT results. This is an indication that my code was properly written and implements the SIMPLE algorithm correctly. For more contour plots, see Appendix A.

Appendix A

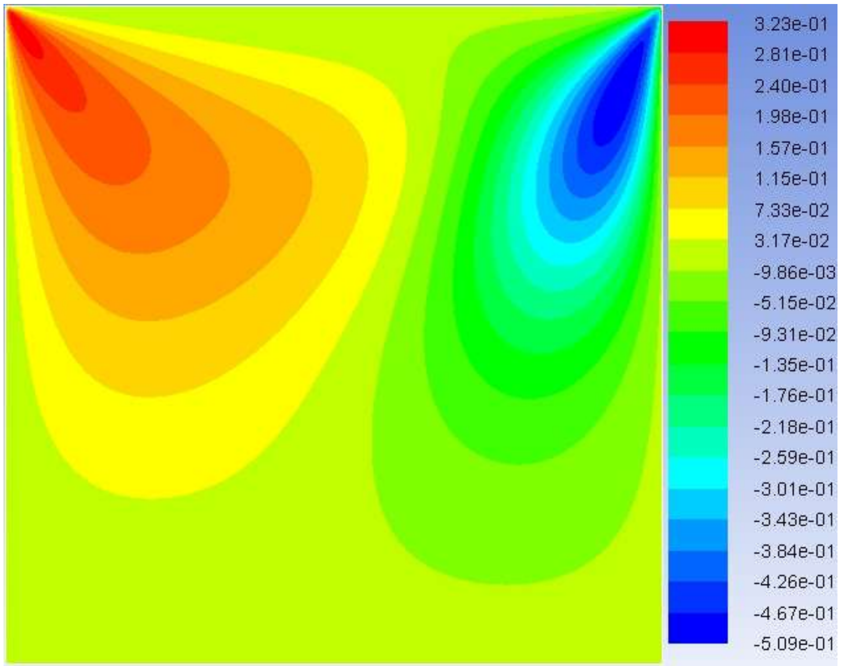

Figure A1. FLUENT results of v-momentum on a 200x200 grid

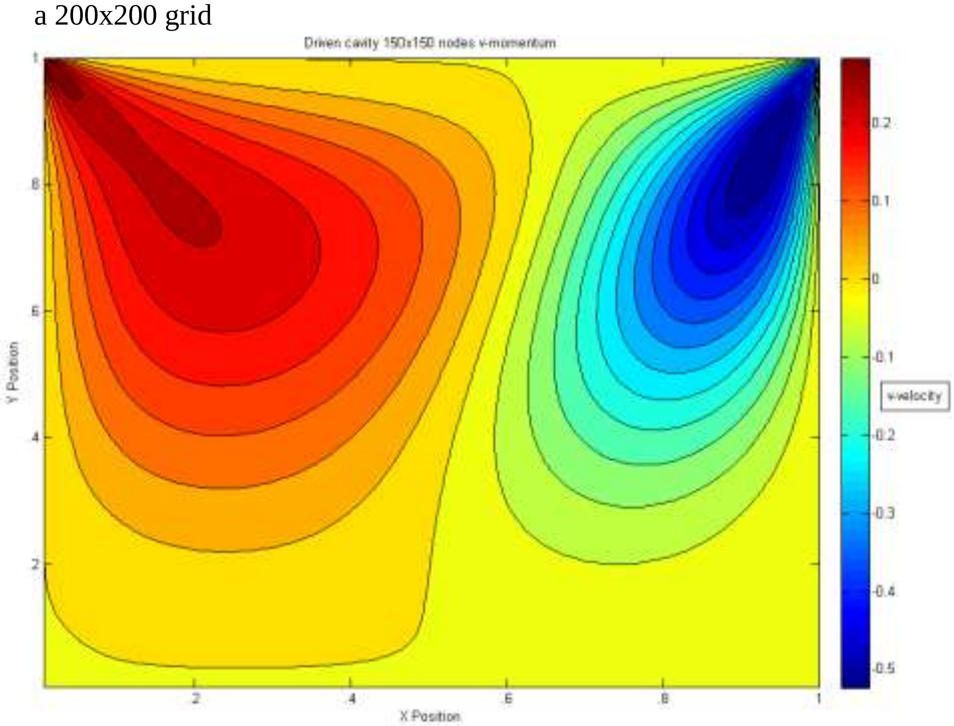

Figure A2. Matlab results of v-momentum on a 150x150 grid

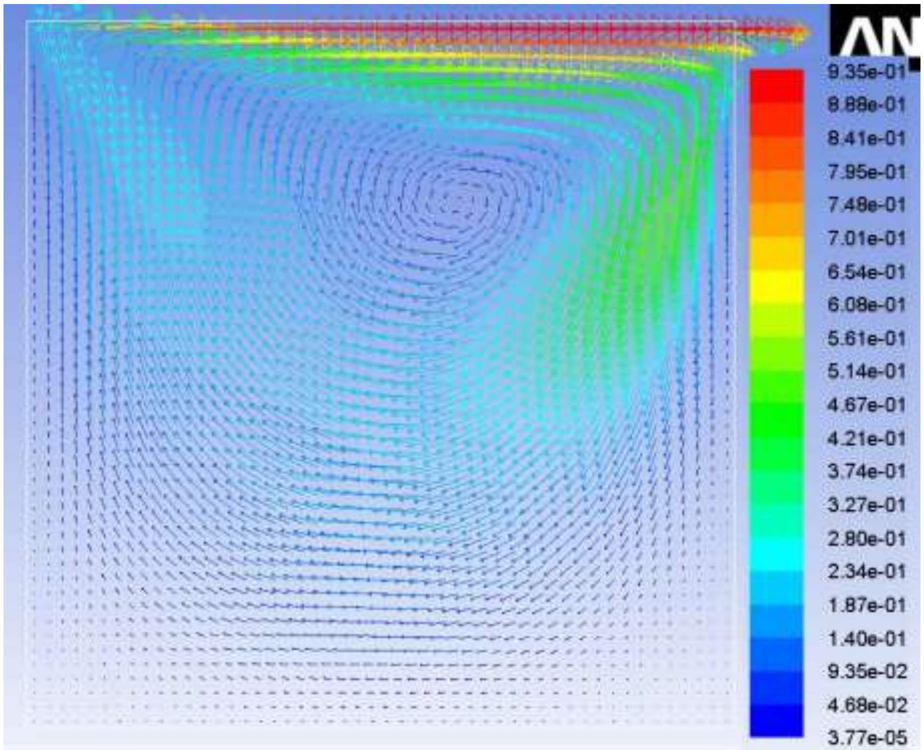

Figure A3. FLUENT velocity vector on a 50x50

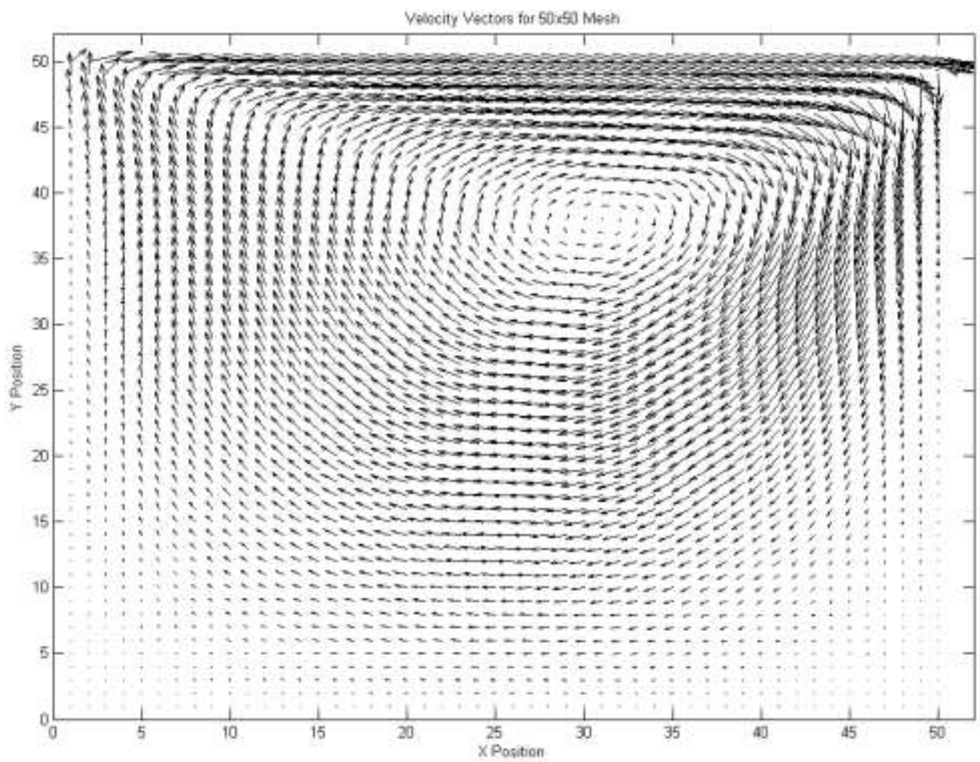

Figure A4. Matlab velocity vector results on a 50x50 driven cavity.

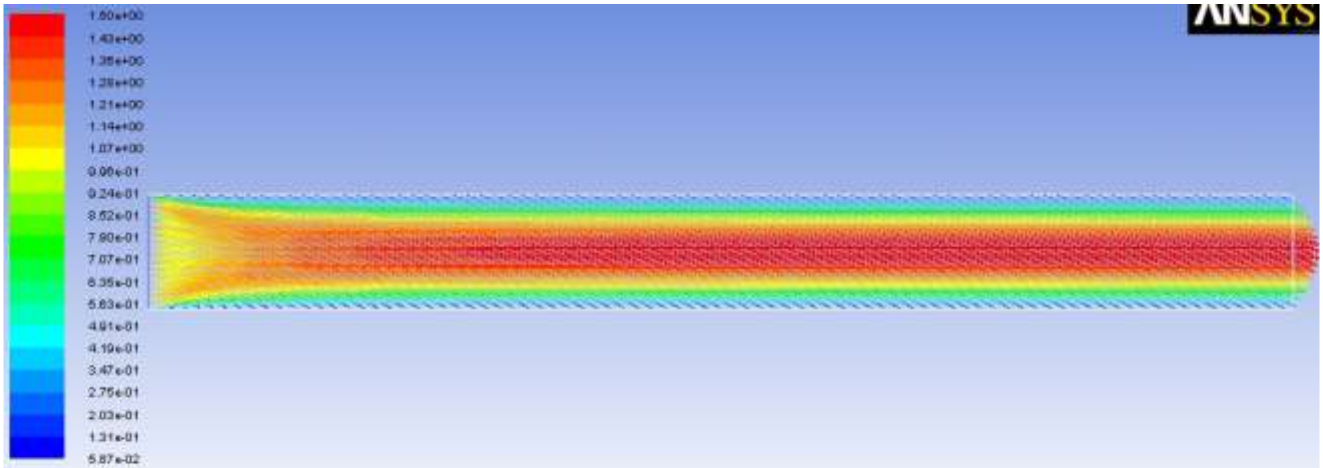

Figure A5. FLUENT velocity vector results on a channel geometry

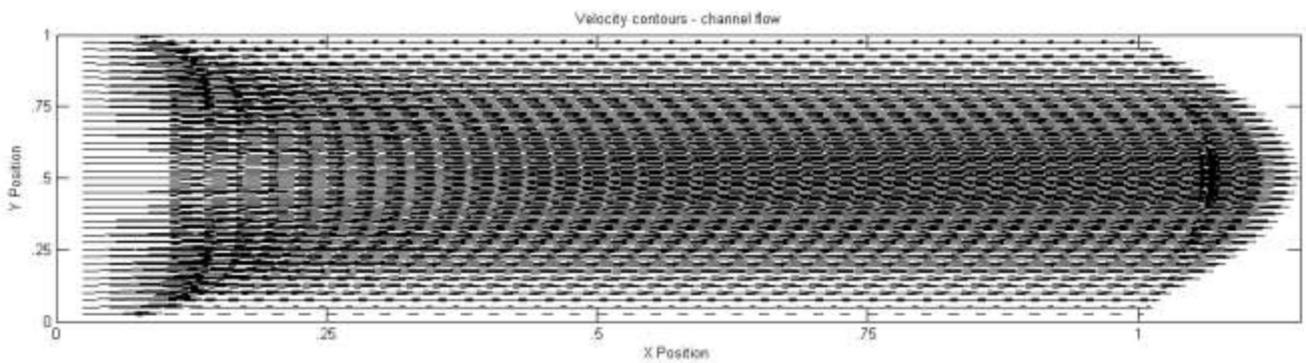

Figure A6. Matlab velocity vector results on a channel geometry