SIMPLE SOLVER FOR A LID-DRIVEN CAVITY FLOW

Stephen Hodson
Purdue University School of Mechanical
Engineering and Birck Nanotechnology Center
Purdue University
West Lafayette, IN 47907 USA
Email: slhodson@purdue.edu

Jonathan Claussen
Purdue University School of Mechanical
Engineering and Birck Nanotechnology Center
Purdue University
West Lafayette, IN 47907 USA
Email: jclausse@purdue.edu

ABSTRACT

The Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) algorithm is used to solve the two-dimensional (2D) Navier-Stokes (NS) equations to determine the velocity field in a lid-driven cavity. A discretization procedure of the NS equations that includes a third-order approximation for the face velocities is outlined. Reynolds numbers of 400, 1000, 3200, and 5000 are studied and benchmarked to previously published literature. Specifically, velocity profiles, stream function values and location of the primary vortex are compared. In addition, the convergence properties of SIMPLE are examined and a comparison to SIMPLEC is performed.

INTRODUCTION

The 2D driven cavity is a rudimentary flow problem that has served as a benchmark for conducting numerical methods on nonlinear dynamics of fluid flow[5]. The simplistic nature of the driven cavity geometry makes it ideal for code validation[5], while the enclosed cavity allows fluid flow to be analyzed even at high Re numbers. The driven cavity simulates a myriad of fluid flow phenomena including corner eddies, longitudinal vortices, transition, and turbulence.[5] The driven cavity flow problem also models scenarios found in nature and industry including current movements caused by wind flowing over large bodies of water (i.e. pond, lake, or ocean) and industrial applications such as use with the short-dwell coater that is used to make high-grade paper and photographic film[5].

Various numerical methods have been used to solve the 2D NS equations for a driven cavity flow problem such as the block-implicit multigrid[6] and lattice Boltzmann[8] methods. Ghia et. al used a multigrid method to solve the stream function vorticity formulation of the 2D NS equations,[2] while Liao et. al. used a second-order differencing equation.[8] Despite being studied for over 20 years, the lid-driven cavity flow problem has yielded different solutions especially for intermediate and high Reynolds numbers because of the various iterative approaches taken to solve the problem[9]. It has even been debated that steady solutions exist for these conditions[1].

This paper reports the numerical computation of the velocity field of a lid-driven cavity flow using the SIMPLE algorithm. The SIMPLE algorithm is a pressure-based method that iteratively and sequentially corrects the velocity field to satisfy both the momentum and continuity equations. A primitive variable formulation of the NS equations is considered and the flow is assumed to be steady, laminar, and incompressible. The third order accurate Quadratic Upwind Interpolation for Convective Kinetics (QUICK) scheme was used to upwind the face velocities of the respective u and v momentum meshes

NOMENCLATURE

Re	Reynolds number
ρ	density
b	source term of discrete equation
μ	viscous constant
∇	divergence operator
$J_\#$	flux (# subscript implies direction)
A	area vector
$D_\#$	diffusion coefficient (# subscript implies direction)
u_b	boundary velocity
u^*	x-direction velocity guess or current iterate
v^*	y-direction velocity guess or current iterate
p^*	pressure guess or current iterate
α_p	pressure under-relaxation
u, v	x and y velocity
p	pressure scalar
S_u	u-momentum equation source
S_v	v-momentum equation source
$F_\#$	face flow rate (# subscript implies direction)
V	velocity vector
$a_\#$	coefficient (# subscript implies direction)
a_{nb}	neighbor coefficient
u_{nb}	neighbor x-direction velocity
v_{nb}	neighbor y-direction velocity
$\Delta V^*_{ho,\#}$	QUICK higher order terms (# subscript implies direction)
$d_\#$	momentum eqn. constants (# subscript implies direction)

Δx	width of mesh cell
Δy	height of mesh cell
Ψ	stream function

NUMERICAL MODELING

COMPUTATIONAL DOMAIN

The computational domain is shown below in Figure 1. The boundary conditions include no penetration and no slip at the walls, thus the u and v velocities are zero at the three stationary walls. Accordingly, the boundary velocities at the moving lid include a constant u and v velocity of zero. Geometrically, an aspect ratio of 1:1 is used.

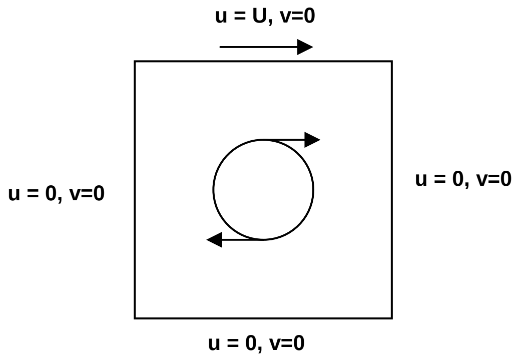

Fig. 1. Computational domain.

GOVERNING EQUATIONS

The governing equations are the NS momentum equations for a Newtonian fluid, which include convection, diffusion, and pressure gradient terms. By applying the steady state assumption and letting Φ = u,v and Γ = μ, the momentum equations in the x and y directions are analogous to the general scalar transport equation as shown below.

$$\frac{\partial(\rho\phi)}{\partial t}+\nabla\bullet(\rho V\phi)=\nabla\bullet(\Gamma\nabla\phi)+S$$
$$\nabla\bullet(\rho Vu)=\nabla\bullet(\mu\nabla u)-\frac{\partial P}{\partial x}+Su \qquad (1)$$
$$\nabla\bullet(\rho Vv)=\nabla\bullet(\mu\nabla v)-\frac{\partial P}{\partial x}+Sv$$

In addition, assuming constant properties and incompressible fluid, the source term S for the u velocity reduces to

$$S=-\frac{\partial P}{\partial x} \qquad (2)$$

The momentum equations are also coupled with the continuity equation shown below.

$$\nabla\bullet(\rho V)=0 \qquad (3)$$

DISCRETIZATION

A staggered grid approach was used to alleviate the problem of checkboarding that arises through co-located storage of the velocity vector and pressure at the same cell centroids. The staggered grid approach is illustrated in Figure 2.

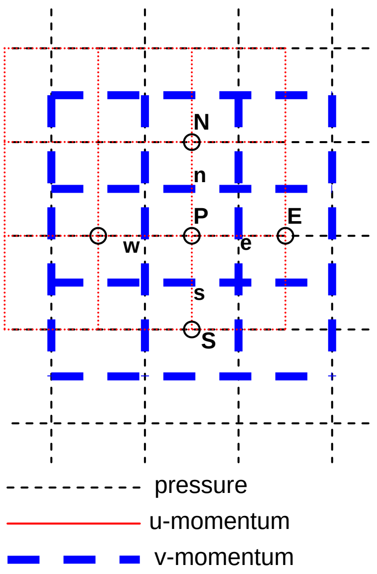

Fig. 2. v-momentum staggered mesh.

DISCRETIZATION OF INTERIOR CELLS

Figure 2 shows the discretization of an interior cell of the v-momentum mesh. A third-order accurate upwind difference scheme, QUICK, was used to upwind the face velocities. QUICK uses a parabolic correction scheme compared to a linear interpolation for first-order accurate upwind difference scheme. In addition, QUICK mitigates the spatial oscillations prevalent in the first-order upwind difference and second-order central difference schemes, while maintaining at least second-order accuracy.[3] Lastly, to ensure positive coefficients, a deferred correction strategy is implemented. Eq. 4 shows the resulting discretization:

$$\begin{aligned}
&a_P v_P=\sum a_{nb}v_{nb}+b\\
&a_E=D_u+\max[-F_e,0]\\
&a_W=D_u+\max[F_w,0]\\
&a_N=D_v+\max[-F_n,0] \qquad (4)\\
&a_S=D_v+\max[F_s,0]\\
&a_P=\sum a_{nb}+(F_e-F_w+F_n-F_s)\\
&b=(P_s-P_n)\Delta x-F_e\Delta v^{*}_{ho,e}+F_w\Delta v^{*}_{ho,w}-F_n\Delta v^{*}_{ho,n}+F_s\Delta v^{*}_{ho,s}
\end{aligned}$$

Assuming that $F_e > 0$,

$$\Delta v^*_{ho,e} = \frac{v^*_E + v^*_P}{2} - \frac{v^*_E + v^*_W - 2v^*_P}{8} - v^*_P \qquad (5)$$

where the starred terms are values from the previous iterate. The higher order terms are located in the source term, while the face mass flow rates that represent the mass balance for a cell is found in the a_P term. The mass flow rates out of the north and south faces are computed using continuity satisfying velocities, which are located at u and v momentum centroids. Figure 3 shows a typical stencil.

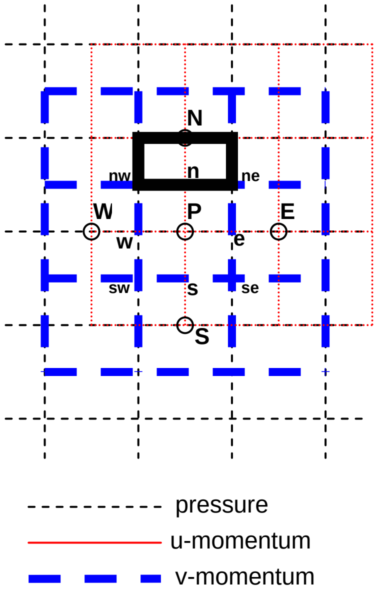

Fig. 3. Half control volume above the v-momentum interior cell.

By applying a mass balance on the half control volume highlighted black in Fig. 3, F_n can be determined by Eq. 6

$$F_n = \rho v_n \Delta x = F_N + \frac{1}{2}\left(F_{ne} - F_{nw}\right) \qquad (6)$$

where F_{ne} and F_{nw} are mass flow rates associated with u-momentum centroids. Conversely, the face mass flow rate F_e was found by Eq. 7.

$$F_e = \rho u_e \Delta y = \rho\left(\frac{u_{ne} + u_{se}}{2}\right)\Delta y = \frac{1}{2}F_{ne} + \frac{1}{2}F_{se} \qquad (7)$$

The face mass flow rates F_w and F_s are analogous to F_e and F_n, respectively.

DISCRETIZATION OF BOUNDARY CELLS

The fluxes through the northern and eastern boundaries were discretized according to Eqs. 8 and 9.

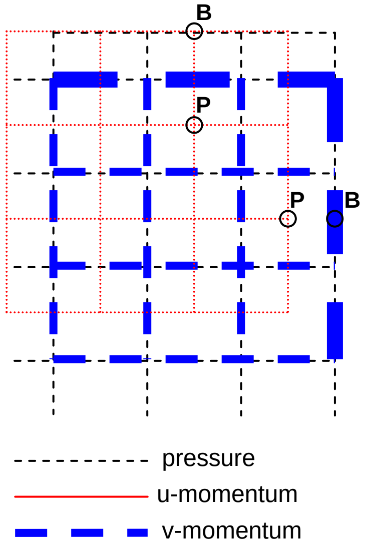

Fig. 4. Northern and eastern boundaries of the v-momentum mesh.

The diffusion term in the northern boundary flux term includes a Δy in the denominator, while the northern boundary flux contains a 0.5 Δx term in the denominator. This occurs due to the staggered nature of the v-momentum control volume observed in Figure 4.

$$J_N A_N = (pvv)_N \Delta y - \mu \Delta y\left(\frac{u_{bN} - u_P}{\Delta y}\right) \qquad (8)$$

$$J_E A_E = (puv)_E \Delta y - \mu \Delta y\left(\frac{u_{bE} - u_P}{\frac{1}{2}\Delta x}\right) \qquad (9)$$

The u-momentum equations are discretized in a similar fashion to the v-momentum equations.

THE SIMPLE AND SIMPLEC ALGORTITHMS

SIMPLE

First proposed by Patanker and Spalding in 1972, the SIMPLE algorithm is a numerical method that sequentially solves the coupled nonlinear NS and continuity equations.[4] An initial guess for pressure is made and the u and v velocities are calculated from the discrete momentum equations. A discrete 'pressure correction equation', which is derived from continuity, uses these velocities to create corrections for the pressure and velocity fields, making them continuity satisfying. The continuity satisfying terms are iterated continuously until they satisfy both the momentum and continuity equations. Qualitatively, the pressure correction equation nudges the u and v velocity fields to satisfy both the momentum and continuity equations.[3] The pressure correction equation is given in equation Eq. 10. Notice that the Scarborough criterion for the pressure correction equation is satisfied in the equality only. This implies that solution to the pressure correction is only accurate up to an arbitrary constant.

$$
\begin{aligned}
& a_p p_p^{'} = \sum_{nb} a_{nb} p_{nb}^{'} + b \\
& a_E = \rho_e d_e \Delta y \\
& a_W = \rho_w d_w \Delta y \\
& a_N = \rho_n d_n \Delta x \\
& a_S = \rho_s d_s \Delta x \\
& a_P = \sum a_{nb} \\
& b = F_w^* - F_e^* + F_s^* - F_n^*
\end{aligned}
\qquad (10)
$$

In Eq. 10,

$$
\begin{aligned}
& d_e = \frac{\Delta y}{a_e} \\
& d_n = \frac{\Delta x}{a_n}
\end{aligned}
\qquad (11)
$$

and the 'b' term represents the amount by which the velocity field does not satisfy continuity. It should be noted that $\Sigma a_{nb} u'_{nb}$ and $\Sigma a_{nb} v'_{nb}$ terms were dropped from the velocity correction equations (Eq. 12) to place the weight of the velocity correction terms on the pressure correction terms.

$$
\begin{aligned}
& a_e u_e^{'} \approx \Delta y \left(p_P^{'} - p_E^{'} \right) \\
& a_n v_n^{'} \approx \Delta x \left(p_P^{'} - p_N^{'} \right)
\end{aligned}
\qquad (12)
$$

This approximation for velocity correction equations changes the path to the final solution, but it does not change the final answer. The entire step-by-step process of the SIMPLE algorithm has been included as a flow chart in Figure 5.

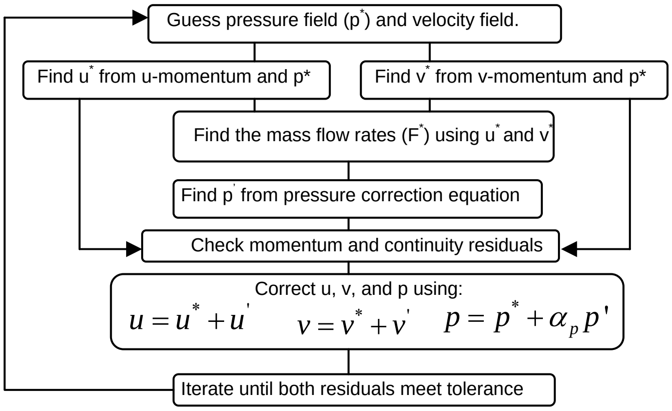
Guess pressure field (p*) and velocity field.
Find u* from u-momentum and p*
Find v* from v-momentum and p*
Find the mass flow rates (F*) using u* and v*
Find p' from pressure correction equation
Check momentum and continuity residuals
Correct u, v, and p using:
$u = u^* + u'$ $v = v^* + v'$ $p = p^* + \alpha_p p'$
Iterate until both residuals meet tolerance

Fig. 5. Flow chart of the SIMPLE algorithm.

During the final iteration the source term in the pressure correction is zero and produces constant pressure corrections, resulting in zero velocity corrections. Hence, the velocity field satisfies both the discrete momentum and continuity equations.[3]

SIMPLEC

The SIMPLEC algorithm was also used to solve the driven cavity flow problem because it converges to a solution faster with less computational expense than the SIMPLE algorithm. Instead of dropping the $\Sigma a_{nb} u'_{nb}$ and $\Sigma a_{nb} v'_{nb}$ terms from the velocity correction equations, SIMPLEC approximates these terms according to Eq. 13.

$$
\begin{aligned}
& \sum_{nb} a_{nb} u_{nb}^{'} \approx u_e^{'} \sum_{nb} a_{nb} \\
& \sum_{nb} a_{nb} v_{nb}^{'} \approx v_n^{'} \sum_{nb} a_{nb}
\end{aligned}
\qquad (13)
$$

The step-by-step flowchart for the SIMPLEC algorithm follows the same flow chart as for the SIMPLE program with exception that the pressure correction no longer needs to be relaxed.

$$
\begin{aligned}
& d_e = \frac{\Delta y}{(a_e - \sum_{nb} a_{nb})} \\
& d_n = \frac{\Delta x}{(a_n - \sum_{nb} a_{nb})}
\end{aligned}
\qquad (14)
$$

RESULTS

The velocity field was resolved for various Re: 400, 1000, 3200, and 5000. A uniform mesh of 130 x 130 centroids was used as the computational domain. Conservative under-relaxation values for the pressure correction ranged from $\alpha_p = 0.1 - 0.2$, while values for the momentum equations are converses of α_p. Lower relaxation values for the pressure correction, meaning less weight on the previous iterate, are needed at higher Re to suppress the possibility of large numerical oscillations that may cause the system to diverge.

The u and v velocity profiles along the geometric vertical and horizontal centerlines, respectively, are compared to the benchmark velocity profiles by Ghia et.al.[2] in Fig. 6 – 13. The location of the primary vortex is also compared to the benchmark locations. It should be noted that the benchmark paper used a vorticity stream function formulation of the NS equations in conjunction with a 129 x 129 mesh size which was solved using a multi-grid method.

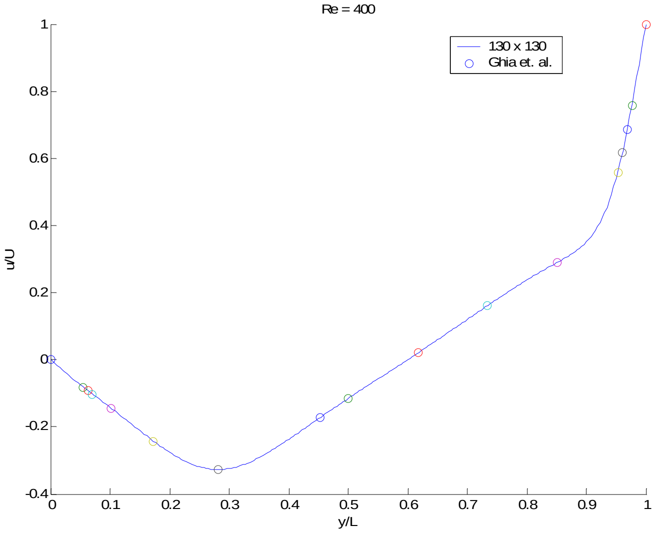

Fig. 6. u-velocity, Re = 400, Mesh size: 130x130

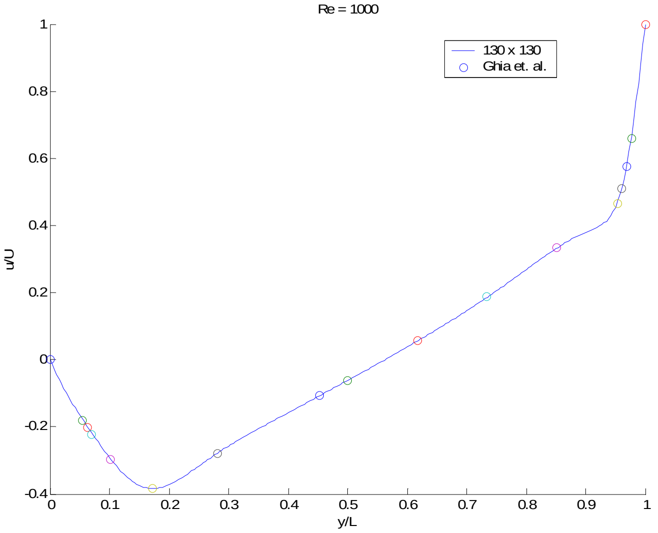

Fig. 8. u-velocity, Re = 1000, Mesh size: 130x130

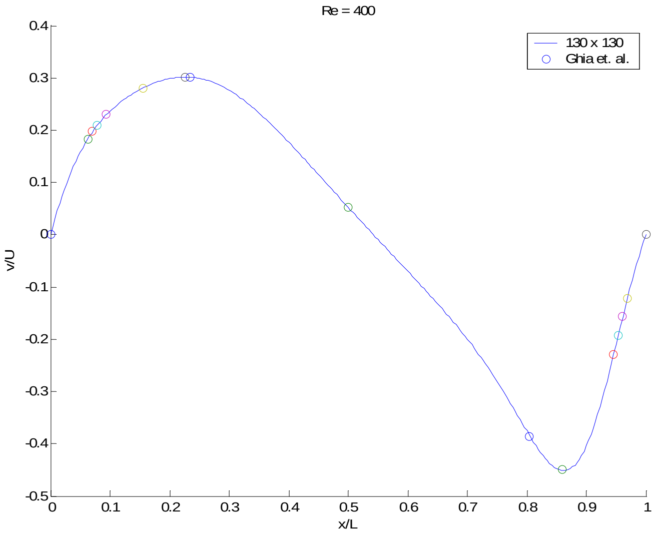

Fig. 7. v-velocity, Re = 400, Mesh size: 130x130

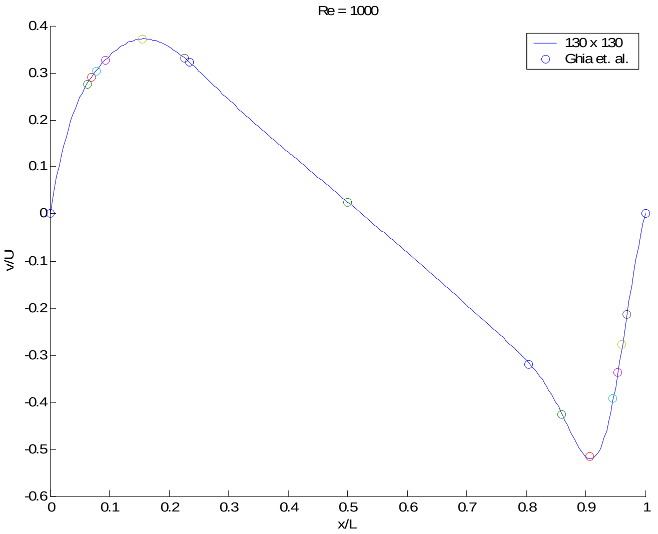

Fig. 9. v-velocity, Re = 1000, Mesh size: 130x130

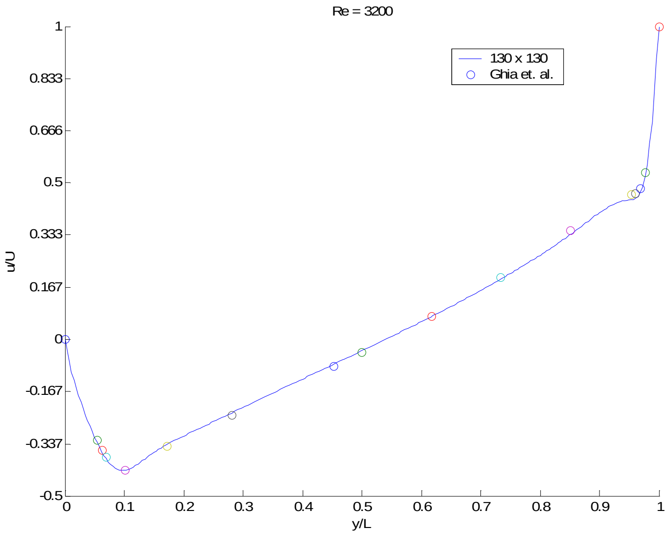

Fig. 10. u-velocity, Re = 3200, Mesh size: 130x130

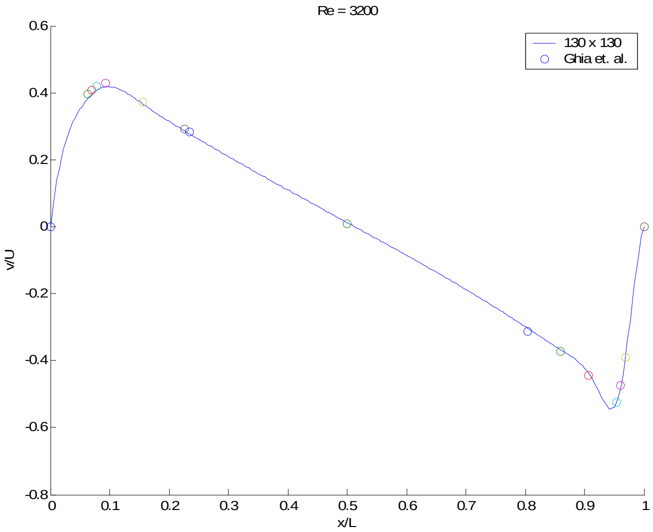

Fig. 11. v-velocity, Re = 3200, Mesh size: 130x130

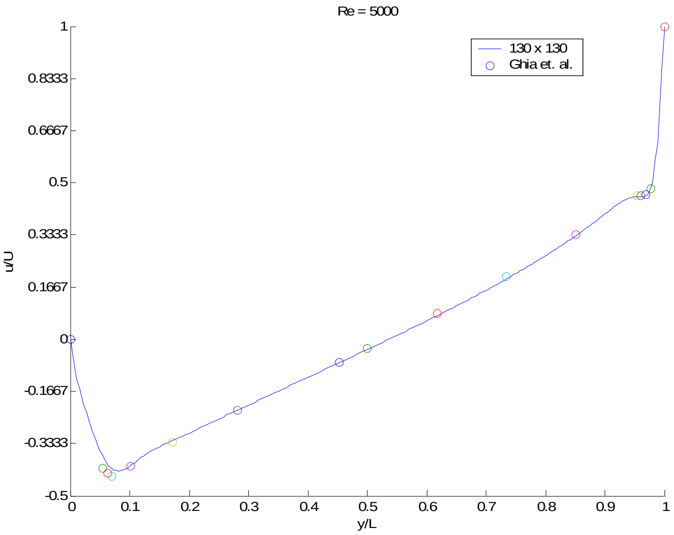

Fig. 12. u-velocity, Re = 5000, Mesh size: 130x130

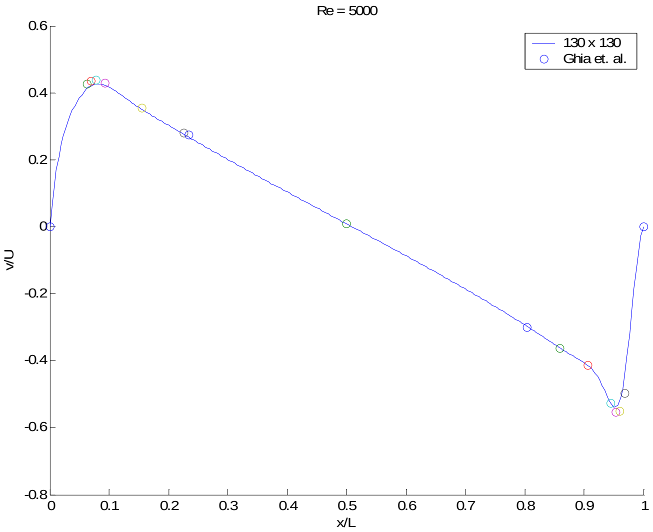

Fig. 13. v-velocity, Re = 5000, Mesh size: 130x130

From observation of the above plots, the resolved velocity fields exhibit exceptional congruence with the benchmark. As Re increases, the velocity profile deviates from the benchmark at sharp inflection points. This inaccuracy can be remedied with a finer mesh to absorb more information at those sites. Looking at the u-velocity plots, we notice that the velocity gradient becomes sharper at the edges as Re increases, which is indicative of boundary layer thinning. Also, at higher Re, both the u and v velocity profiles develop kinks near y = 1 and x =1, respectively. As previously reported, this behavior arises in fine meshes and implies that velocity distributions near walls of the cavity are not sensitive to mesh size[2].

In Fig. 14 – 17, streamline plots of the flow are shown to visualize the flow field. In addition, the center location of the primary vortex and its associated minimum stream function value is compared to the benchmark in Table 1.

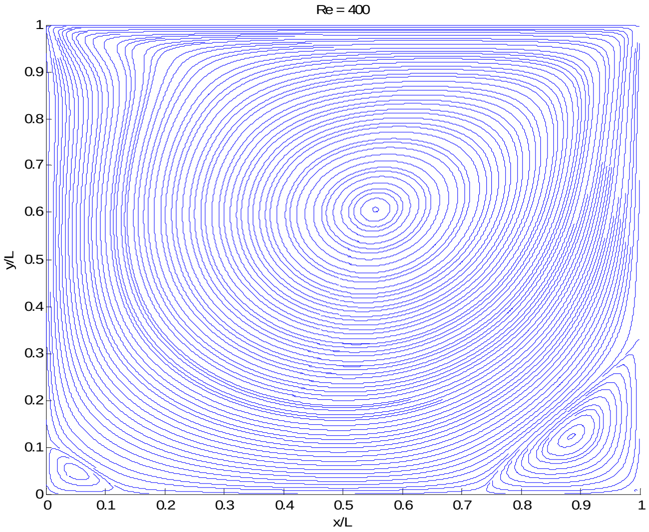

Fig. 14. Streamlines, Re = 400, Mesh size: 130x130

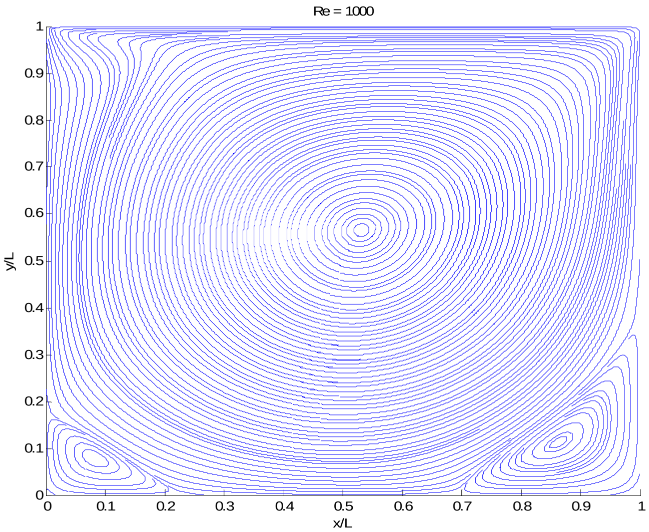

Fig. 15. Streamlines, Re = 1000, Mesh size: 130x130

Re	**Ψ_{min}**	**Ψ_{min} (bm.)**	**x/L**	**x/L (bm.)**	**y/L**	**y/L (bm.)**
400	-0.1130	-0.1139	0.5562	0.5547	0.6077	0.6055
1000	-0.1171	-0.1179	0.5310	0.5313	0.5660	0.5625
3200	-0.1167	-0.1204	0.5161	0.5165	0.5424	0.5469
5000	-0.1155	-0.1190	0.5131	0.5117	0.5400	0.5352

Table 1. Primary vortex: stream function and location (Note: benchmark used 257 x 257 mesh)

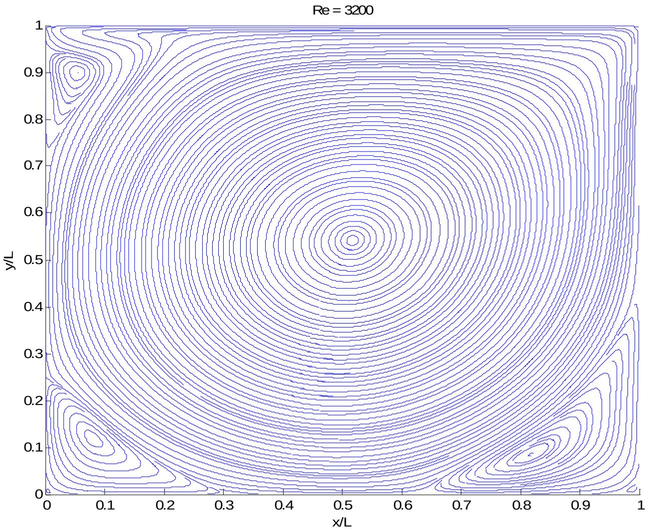

Fig. 16. Streamlines, Re = 3200, Mesh size: 130x130

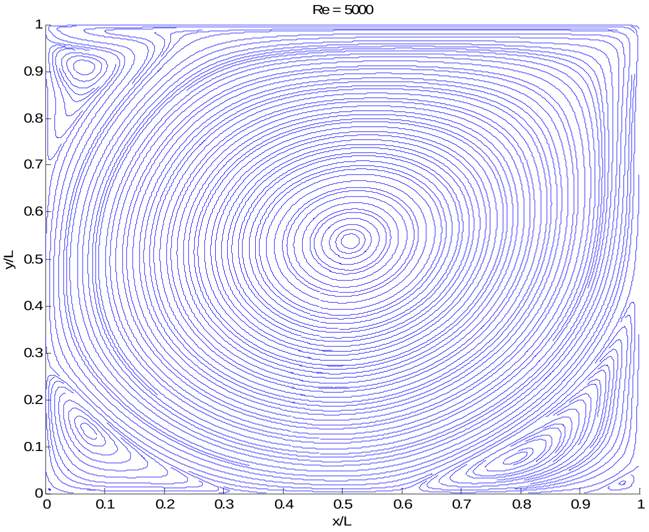

Fig. 17. Streamlines, Re = 5000, Mesh size: 130x130

Qualitatively, the flow field develops into a primary vortex near the geometric center with secondary vortices forming in the lower left and right corners. The secondary vortices develop due to their proximity to two boundaries which exhibit the no-slip condition. As Re increases, the primary vortex shifts towards the geometric center of the cavity, which is similar behavior to the benchmark. More so, secondary vortices increase in size and a new vortex forms in the upper right corner as the Re increases above 3200. For Re = 5000, another small vortex forms in the south east corner. As shown in Table 1, the resolved minimum stream function value and location of the primary vortex match very well with the benchmark. However, differences between the resolved and benchmark stream function values tends to be larger for Re = 3200 and 5000. This discrepancy could be a result of larger numerical oscillations and higher velocity corrections than seen at lower Re. More accurate results could be obtained with a more stringent convergence tolerance and finer mesh.

CONVERGENCE PROPERTIES

While the above results are promising, the accuracy was accompanied by a computational cost due to the use of a Gauss-Seidel (GS) method in solving the pressure correction equation each non-linear, outer loop iteration. This cost, in terms of number of iterations and elapsed CPU time to convergence is tabulated below. Note that the convergence tolerance was based upon a summation of the normalized momentum and continuity residuals. While a liberal convergence tolerance of 10^{-2} was used, the maximum u and v velocity corrections were observed to be $O(10^{-6})$. Convergence to a solution seemed strongly dependent not only on the under-relaxation values, but also the GS tolerance used while resolving pressure correction values. Using a conservative under-relaxation values for the pressure correction to prevent divergence, numerical oscillations of the continuity residual arose and continued to iterations upwards of ~10000 with high GS tolerances(~10^{-2}). Therefore, a dynamic scheme was implemented to monitor the continuity residual and reduce the GS tolerances accordingly.

Re	**Mesh size**	**Iterations**	**Elapsed CPU time (hrs)**	**Min. GS tol.**
400	130 x 130	1468	3.08	10^{-4}
1000	130 x 130	1455	3.10	10^{-4}
3200	130 x 130	1120	7.96	10^{-5}
5000	130 x 130	1729	9.80	10^{-5}

Table 2. Re, Iterations, Elapsed CPU time, GS tolerance

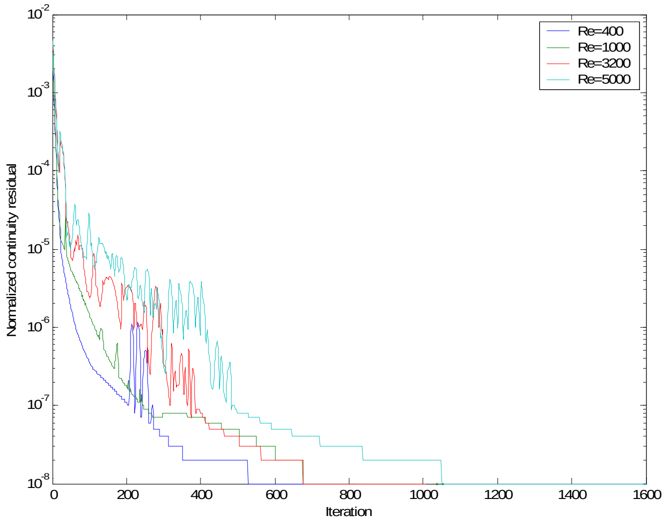

Fig. 18. Normalized continuity residual vs. non-linear outer loop iteration

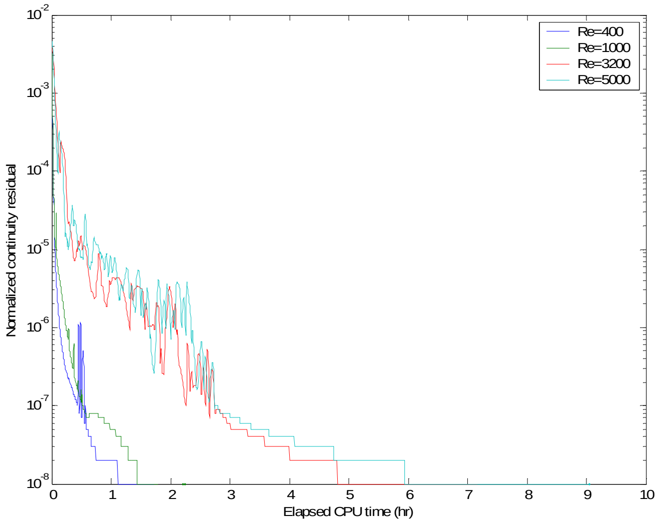

Fig. 19. Normalized continuity residual vs. CPU time

For purposes of the following discussion, iterations refer to non-linear outer loop iterations. Fig. 18 illustrates how convergence is dependent on Re. As expected, steeper convergence rates occur at lower Re. In congruence with this trend, it should be expected that lower Re should converge in a shorter amount of iterations, but as seen in Table 2, Re = 3200 converged at a lower iteration number than Re = 1000 and 400. It has been postulated that at lower GS tolerances per iteration, more accurate pressure corrections are obtained, resulting in more accurate velocity corrections . Hence, convergence is achieved in a smaller amount of iterations for lower GS tolerances. However, the number of iterations for Re = 5000 does not corroborate this hypothesis and we suspect that large numerical oscillations become more of a factor at higher Re.

In contrast, the discrepancy between convergence rates at different Re becomes more accentuated when plotted against elapsed computation time as shown in Fig. 19. Elapsed computation times at higher Re more than double that at lower Re, which is due to larger numerical oscillations, or highly over-corrected velocities per iteration. More so, the over-corrected velocities, which appear in the source term in the pressure correction equation, could cause the GS method to require more time (TDMA iterations) to reach a steady solution when solving the pressure correction equation.

SIMPLE VERSUS SIMPLEC

A comparison of the convergence rates between SIMPLE and SIMPLEC are tabulated in the Table 3. Re = 400 with a 130 by 130 mesh size was used as the test case. SIMPLEC was tested with under-relaxation values of 0.8, 0.9, and 0.95 for the momentum equations, while 0.8 was used for SIMPLE. It should be mentioned that SIMPLE and SIMPLEC represent paths to a solution. Therefore, the rate of convergence to a solution may vary between them, but the solutions will not differ.

Method	α_u	Iterations	Elapsed CPU time (hr.)
SIMPLE	0.8	1468	3.08
SIMPLEC	0.8	1900	2.62
SIMPLEC	0.9	690	1.33
SIMPLEC	0.95	380	0.85

Table 3. SIMPLE vs. SIMPLEC

From the results, it is seen that SIMPLEC out-performs SIMPLE for momentum under-relaxation values greater than 0.8. This is expected since the pressure correction under-relaxation value is unity. For α_u = 0.8, SIMPLEC only reduces the elapsed CPU time. Further investigation may exhibit that SIMPLEC performs better than SIMPLE for larger momentum relaxation values.

CONCLUSIONS

The SIMPLE algorithm has been implemented to resolve the velocity field from the 2D Navier-Stokes equations. Face velocities were approximated with the third-order QUICK scheme. The velocity profiles, location of the primary vortex and corresponding stream function values for Re of 400, 1000, 3200, and 5000 were studied. For each Re, the results strongly match the benchmark data. The convergence properties of the SIMPLE algorithm were examined and showed strong dependence on under-relaxation values and Re. In conclusion, the SIMPLE algorithm accompanied by QUICK yields accurate results for relatively coarse meshes, but requires large iterations and computation time using Gauss-Seidel as an iterative and sequential solver of the discrete momentum and pressure correction equations. For more accurate results, finer meshes and higher-order schemes should be used in conjunction with a multi-grid method to increase the rate of convergence.

REFERENCES

1. E. Erturk, T.C.C.C.G., *Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers.* International Journal for Numerical Methods in Fluids, 2005. **48**(7): p. 747-774.
2. Ghia, U., K.N. Ghia, and C.T. Shin, *High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method.* Journal of Computational Physics, 1982. **48**(3): p. 387-411.
3. Murthy, J.Y., *Numerical Methods in Heat, Mass, and Momentum Transfer*. 2002, West Lafayette.
4. Patankar, S.V., *Numerical Heat Transfer and Fluid Flow*. Computational and Physical Processes in Mechanics and Thermal Sciences, ed. W.J.M.a.E.M. Sparrow. 1980: Taylor & Francis.
5. Shankar, P.N. and M.D. Deshpande, *Fluid Mechanics in the Driven Cavity.* Annual Review of Fluid Mechanics, 2000. **32**(1): p. 93-136.
6. Vanka SP. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *Journal of Computational Physics* 1986; **65**: 138-158
7. Liao SJ, Zhu JM. A short note on higher-order streamfunction-vorticity formulation of 2-D steady state Navier-Stokes equations. *International Journal for Numerical Methods in Fluids* 1996; **22**: 1-9.
8. Hou S, Zou Q, Chen S, Doolen G, Cogley AC. Simulation of cavity flow by the lattice Boltzmann method. *Journal of Computational Physics* 1995; **118**: 329-347
9. Bruneau, C.-H. and M. Saad, *The 2D lid-driven cavity problem revisited.* Computers & Fluids, 2006. **35**(3): p. 326-348.