A Numerical Analysis for 2-D Navier-Stokes Equations by FDM and FVM

Lei LEI

Department of Mechanical Engineering, Columbia University

Abstract: Two-dimensional unsteady Navier-Stokes equation was numerically discretized using projection method in finite difference and SIMPLE algorithm in finite volume on a staggered grid system. The projection method was achieved via Python script and validated by standard CFD test case – driven cavity flow. Potential affecting factors, including grid size, boundary conditions, were analyzed and results matched pretty well with those from benchmark paper.

Keywords: 2-D Navier-Stokes equation, finite difference method, finite volume method

NOMENCLATURE

$\vec{u}$	local velocity vector with components *(u, v)*
p	local averaged pressure
ρ	fluid density, constant in this case
ν	kinematic molecular viscosity
ε	tolerance of iteration
Ω	solution domain of interest
O	order of magnitude of a quantity
∇	divergent operator, $\nabla = \frac{\partial \bullet}{\partial x} + \frac{\partial \bullet}{\partial y}$
Δ	finite change in quantity
$\vec{U}^n$	numerical solution of velocity at time step n, with element U^n and V^n

1. INTRODUCTION

Navier-Stokes (NS) equations, derived from Newton's 2nd law and 1st law of thermodynamics, are a set of nonlinear partial differential equations describing the motion and energy balance of the viscous fluid substances. Together with supplemental relations, conservation of mass for example, and well defined boundary conditions, NS equations have capability of modeling various scientific and engineering situations accurately and effectively. Regardless of their extensive applications in practice, the nonlinearity makes it troublesome (or sometimes impossible in 3D) to come up with analytical solutions, which introduces numerical analysis into the scheme.

The goal of the work presented is to numerically simulate the transient Navier-Stokes equations for two-dimension incompressible viscous flow with both *finite difference method* (FDM) and *finite volume method* (FVM) using Python script. The characteristics of both methods, including stability and convergence, are investigated in further detail by applying them to a classical test-case study of cavity flow.

2. METHODS

The NS momentum equations for 2D incompressible flow in the Cartesian coordinate can be expressed in the vector form as follows (Eq. 1), which would be discretized by both FDM and FVM respectively with second-order accuracy in space and at least first order in time.

$$\vec{u}_t + \left(\vec{u} \cdot \nabla\right)\vec{u} = -\frac{1}{\rho}\nabla p + \nu\nabla^2 \cdot \vec{u} \qquad \text{(Eq. 1)}$$

2.1 Finite Difference Method

In FDM, Ω is discretized into a staggered grid with the pressure in the cell centers while the velocities placed on the midpoints of interfaces (Fig.1). The projection method developed by Patenker and Spalding [4] was adopted to discretize the original Navier-Stokes equation. In this

scheme, an intermediate velocity u^* was introduced for momentum correction (Eq.2) which, however, did not satisfy the continuity equation. For both velocity components on staggered grids, the viscosity term was treated implicitly from (Eq.3), where U^{**} is a velocity correction term.

$$\frac{\vec{U}^{*} - \vec{U}^{n}}{\Delta t} = -\nabla \cdot \left(\vec{U}^{n}\right)^{2} \qquad \text{(Eq. 2)}$$

$$\frac{\vec{U}^{**} - \vec{U}^{*}}{\Delta t} = \nabla^{2} \cdot \left(\nu \vec{U}^{*}\right) \qquad \text{(Eq. 3)}$$

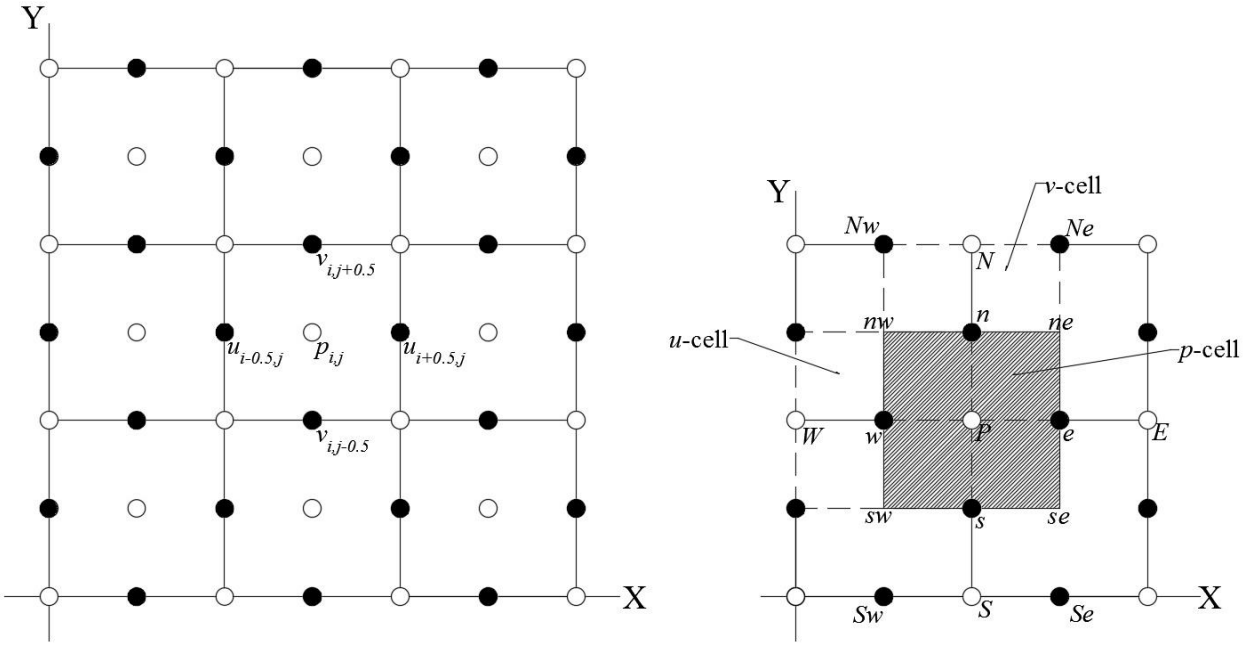


Fig. 1 Staggered Grids of Calculation for FDM (left) and FVM (right)

Pressure terms are evaluated using velocity correction terms and velocities at $(n+1)^{th}$ time step.

$$\frac{\vec{U}^{n+1} - \vec{U}^{**}}{\Delta t} = -\frac{1}{\rho} \nabla \cdot p^{n+1} \qquad \text{(Eq. 4)}$$

Such that (Eq.1) was the summation of (Eq.2), (Eq.3) and (Eq.4). As U^{n+1} and V^{n+1} together must fulfill the continuity equation (Eq.5) such that a Poisson pressure equation was established (Eq.6).

$$\nabla \cdot \vec{U}^{n+1} = U_{x}^{n+1} + V_{y}^{n+1} = 0 \qquad \text{(Eq. 5)}$$

$$\nabla^2 p^{n+1} = \frac{\rho}{\Delta t}\nabla \cdot \vec{U}^n \qquad \text{(Eq. 6)}$$

Central difference approximations were applied to second derivatives of both *x*- and *y*-directions (Eq.7a). The first derivatives, required for pressure calculation, were simulated differently at the center of gird where pressure field was stored (Eq.7b).

$$\nabla^2 \cdot U_{i,j}^n = \left(U_{xx}^n\right)_{i,j} + \left(U_{yy}^n\right)_{i,j} = \frac{U_{i+1,j}^n - 2U_{i,j}^n + U_{i-1,j}^n}{\Delta x^2} + \frac{U_{i,j+1}^n - 2U_{i,j}^n + U_{i,j-1}^n}{\Delta y^2} \qquad \text{(Eq. 7a)}$$

$$\left(U_x^n\right)_{i+\frac{1}{2},j} = \frac{U_{i+1,j}^n - U_{i,j}^n}{\Delta x} \qquad \text{(Eq. 7b)}$$

The approach to dealing with nonlinear term $\vec{U}^2$ was tricky as *x*- and *y*- velocities were defined at different positions (Fig.1). As an example of *U* updating, *UU* and *UV* were needed and should be evaluated at the locations of velocity *U*. An effective way is to average the adjacent values and take the derivative to averaged term as (Eq.2) to yield the desired expression (Eq.8).

$$\frac{U_{i,j}^* - U_{i,j}^n}{\Delta t} = -\left[\frac{1}{4}\left(U^n_{i+\frac{1}{2},j} + U^n_{i-\frac{1}{2},j}\right)^2\right]_x - \left[\frac{1}{4}\left(U^n_{i,j+\frac{1}{2}} + U^n_{i,j-\frac{1}{2}}\right)\left(V^n_{i+\frac{1}{2},j} + V^n_{i-\frac{1}{2},j}\right)\right]_y \qquad \text{(Eq. 8a)}$$

$$\frac{V_{i,j}^* - V_{i,j}^n}{\Delta t} = -\left[\frac{1}{4}\left(V^n_{i,j+\frac{1}{2}} + V^n_{i,j-\frac{1}{2}}\right)^2\right]_y - \left[\frac{1}{4}\left(U^n_{i,j+\frac{1}{2}} + U^n_{i,j-\frac{1}{2}}\right)\left(V^n_{i+\frac{1}{2},j} + V^n_{i-\frac{1}{2},j}\right)\right]_x \qquad \text{(Eq. 8b)}$$

The procedures below were followed in the numerical solution to 2-D NS equations.

(1) Evaluate intermediate velocity field (Eq.8) to get $\vec{U}^* = \vec{U}^n + \Delta t \cdot \vec{H}$.

(2) Solve Poisson pressure equation (Eq.6), yielding $p^{n+1}(x, y)$.

(3) Compute velocity correction term form (Eq.3) to get $\vec{U}^{**} = \vec{U}^* + \Delta t \cdot \nabla^2 \cdot \left(\nu \vec{U}^*\right)$.

(4) Update the velocity field with $\vec{U}^{n+1} = \vec{U}^{**} - \frac{\Delta t}{\rho}\nabla \cdot p^{n+1}$.

By expanding terms using Taylor series at point (i, j) and time *t*, it could be show that the FDM applied here was second-order accurate in space and first-order in time. The existence and uniqueness of strong solution to 2-D NS equations had already been proved by Mattingly [5].

2.2 Finite Volume Method

In contrast to finite difference method where computational domain is divided into hexahedral cells and discretized equations, derived from differential form of governing equation, are evaluated on the basis of dependent variable values stored at the nodes, a finite volume method constructs discretization via the integral form of PDE and approximates it with interpolated face-center values of each control volume (Fig.1). Advantages of FVM over FDM have been profoundly investigated and reported by many researches, including better suitability for complex geometry, stronger fulfillment of conservation law at grid scale [3] and etc. A staggered grid was adopted where pressure data was stored at the apexes (slashed dots) while velocity was evaluated at centroids (black dots) and face centers (hollow dots) were used as bridges to relate adjacent grids.

The first step of implementing FVM is to integrate governing equation with respect to time and space on the velocity control volume. For *x*-momentum equation in (Eq. 2), by assuming the face-averaged values equal to the values at center of faces, the integral form was expressed as follows.

$$\begin{aligned}&\left(u_w^{n+1}-u_w^n\right)\Delta x\Delta y+\Delta t\Delta y\left(u_{C,P}u_P^{n+1}-u_{C,W}u_W^{n+1}\right)+\Delta t\Delta x\left(v_{C,nw}u_{nw}^{n+1}-v_{C,sw}u_{sw}^{n+1}\right)\\&=-\frac{\Delta t\Delta y}{\rho}\left(p_P-p_W\right)+\nu\Delta t\Delta y\left[\left(\frac{\partial u}{\partial x}\right)_P-\left(\frac{\partial u}{\partial x}\right)_W\right]+\nu\Delta t\Delta x\left[\left(\frac{\partial u}{\partial y}\right)_{nw}-\left(\frac{\partial u}{\partial y}\right)_{sw}\right]\end{aligned}\qquad\text{(Eq. 8)}$$

Where $u_C's$ and $v_C's$ are known as convecting velocities, which are approximated by average of the centroids coincident to the face. u_W, u_P, u_{nw}, u_{sw} are convected velocities, a representative of convection momentum transferring through the surfaces. A second-order upwind method were

introduced in FVM such that for flow moving along positive *x*,*y*-direction, the convected velocities at face *P* and *sw* (u_P and u_{nw}) would be approximated respectively as (Eq.9).

$$u_P = \frac{3}{2}u_w - \frac{1}{2}u_{ww} \qquad u_{nw} = \frac{3}{2}u_w - \frac{1}{2}u_{Sw} \qquad \text{(Eq. 9)}$$

For the diffusion terms, central difference was applied such that

$$\left(\frac{\partial u}{\partial x}\right)_P = \frac{u_e - u_w}{\Delta x} \qquad \left(\frac{\partial u}{\partial y}\right)_{nw} = \frac{u_{Nw} - u_w}{\Delta y} \qquad \text{(Eq. 10)}$$

Substituting the above expressions into (Eq.8) yielded an explicit relation among neighboring grids

$$u_w^{n+1} = \sum_{i=e,ww,Sw,Nw} \alpha_i u_i^{n+1} + \alpha_w u_w^n + S_w^{n+1} \qquad \text{(Eq. 11)}$$

With the coefficients α_i's and source term S_P^n defined as follows

$$\alpha_w = 1 - \frac{3\Delta t}{2\Delta x}u_{C,P} - \frac{3\Delta t}{2\Delta y}v_{C,Nw} - \frac{2\nu\Delta t}{\Delta x^2} - \frac{2\nu\Delta t}{\Delta y^2}, \qquad \alpha_e = \frac{\nu\Delta t}{\Delta x^2}, \qquad \alpha_{Nw} = \frac{\nu\Delta t}{\Delta y^2}, \quad \text{(Eq. 12a)}$$

$$\alpha_{ww} = \frac{\Delta t}{2\Delta x}u_{C,P} + \frac{3\Delta t}{2\Delta x}u_{C,ww} + \frac{\nu\Delta t}{\Delta x^2}, \qquad \alpha_{Sw} = \frac{\Delta t}{2\Delta y}v_{C,nw} + \frac{3\Delta t}{2\Delta y}v_{C,sw} + \frac{\nu\Delta t}{\Delta y^2}, \qquad \text{(Eq.12b)}$$

$$S_p^{n+1} = -\frac{\Delta t}{2\Delta x}u_{C,W}\cdot u_{www}^{n+1} - \frac{\Delta t}{2\Delta y}v_{C,sw}\cdot u_{SSw}^{n+1} - \frac{\Delta t}{\rho\Delta x}\left(p_P^n - p_W^n\right) \qquad \text{(Eq.12c)}$$

Similar iteration expression can be derived for *y*-velocity. As in the FDM scheme, pressure still remains implicit, in order to proceed, the Semi-Implicit Pressure Linked Equations (SIMPLE) algorithm was applied. SIMPLE decomposes the velocities and pressure into guessed values and correction terms (Eq. 13) and simulates their distribution filed with two equations, one linking the velocity correction to the pressure correction and the other solving the pressure correction to update original guess. The illustrated procedures were taken to evaluate NS equations in this study.

(1) Decomposition of target values into guess (*) and correction (') terms.

$$u = u^* + u' \qquad v = v^* + v' \qquad p = p^* + p' \tag{Eq. 13}$$

The guessed values must fulfill the discretized momentum PDE (Eq.11) such that guessed velocity at time step *n+1* can be evaluated by (Eq.14).

$$\left(u_w^{n+1}\right)^* = \sum_{i=e,ww,Sw,Nw} \left(\alpha_i u_i^{n+1}\right)^* + \alpha_w u_w^n + \left(S_w^{n+1}\right)^* \tag{Eq. 14}$$

Where α_i's were the same as (Eq.11) while $\left(S_w^n\right)^*$ was computed from guessed values.

(2) Construction of velocity-pressure correction relations by subtracting (Eq.14) from original expression (Eq.11), yielding the correction relations (Eq. 14 for *x*-component).

$$\left(u_w^{n+1}\right)' = \sum_{i=e,ww,Sw,Nw,SSw} \left(\alpha_i u_i^{n+1}\right)' - \frac{\Delta t}{\rho \Delta x}\left[\left(p_P^{n+1}\right)' - \left(p_W^{n+1}\right)'\right] \tag{Eq. 15}$$

SIMPLE algorithm assumes zero velocity corrections for previous time steps and drops all other correction terms of neighboring cells at current time step, above expressions can be reduced into a straightforward form as follows.

$$\left(u_w^{n+1}\right)' = -\frac{\Delta t}{\rho \Delta x}\left[\left(p_P^{n+1}\right)' - \left(p_W^{n+1}\right)'\right] \qquad \left(v_n^{n+1}\right)' = -\frac{\Delta t}{\rho \Delta y}\left[\left(p_N^{n+1}\right)' - \left(p_P^{n+1}\right)'\right] \tag{Eq. 16}$$

(3) Solving for pressure correction terms and updating velocities. To do so, continuity equation (Eq.17) were considered and integrated along the pressure cells (Eq.18).

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{Eq. 17}$$

$$\left(u_e^{n+1} - u_w^{n+1}\right)\Delta y + \left(v_n^{n+1} - v_s^{n+1}\right)\Delta x = 0 \tag{Eq. 18}$$

By decomposing velocities terms and substituting the corresponding expressions in (Eq.16), the pressure-correction equation was given by (Eq.19).

$$-\frac{\Delta t}{\rho \Delta x}\left(P_E' + P_W'\right) + \frac{2\Delta t}{\rho}\left(\frac{1}{\Delta x} + \frac{1}{\Delta y}\right)P_P' - \frac{\Delta t}{\rho \Delta y}\left(P_N' + P_S'\right) = 0 \qquad \text{(Eq. 19)}$$

The pressure correction terms can then be solved implicitly and plugged back into (Eq.16) for velocity corrections and (Eq.13) for calculated velocities at $(n+1)^{th}$ time step.

(4) Use velocities at time step *(n+1)* as new guessed values, repeat procedures (1) to (3) until the prescribed convergence tolerance is achieved.

3. CASE STUDY SET-UP

Numerical methods described above were justified by examining a standard and classical case in computational fluid dynamics (CFD) where upper boundary kept moving above a rectangular cavity. The geometry and boundary conditions for the cavity flow were illustrated in Fig.2.

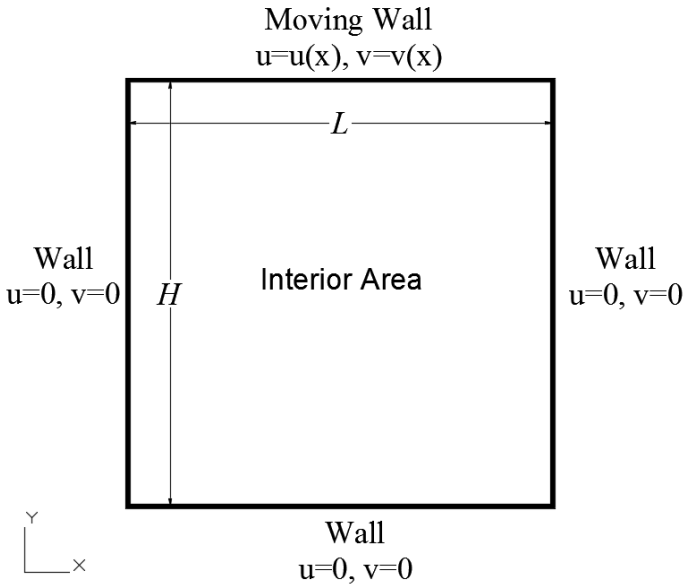


Fig. 2 Geometry and Boundary conditions of Cavity Flow

The validation began with the simplest case where solution domain was set to be a square with $L = H = 1.0$ and a Dirichlet boundary condition of $u = 1.0$ and $v = 0$ was assumed for the upper surface. The process proceeded with various combinations of upper boundary velocities, meshing

strategies, aspect ratios and time intervals (Table 1) to explore the stability and reliability of given methods with a tolerance of $\varepsilon = 0.001$.

Table 1 Various Combinations for Validation

Combination	Moving Boundary	Mesh $(X \times Y)$	Aspect Ratio $(L:H)$	Time Interval
1	$u = 1.0, v = 0$	41×41	1:1	0.01*s*
2	$u = 1.0, v = 0$	82×41	1:1	0.01*s*
3	$u = 3.0, v = 0$	41×41	1:1	0.01*s*
4	$u = 1.0, v = 0$	41×41	1:1	0.001*s*
5	$u = 1.0, v = 0$	82×41	2:1	0.01*s*

4. RESULTS AND DISCUSSION

Note that because of time limit, FVM code could not be debugged as expected. The presented results all came from FDM. But efforts would be continuously spent in improving FVM.

4.1 Convergence

For the prescribed $\varepsilon = 0.001$, combinations 1 to 5 respectively took 1092, 807, 940, 1288, and 3826 times to reach convergence, residual curves (Fig.3) were monotonously decreasing smooth for both *u* and *v*, indicating that the FDM was fairly stable and reliable, which is in accordance with the previous analysis.

The results demonstrated that although the shape of the curves were quite similar to each other, the impacts of various factors on convergence were incomparable. The refinement on grid size would decrease the number of iteration but dramatically increase the calculation for every step and consequent computing time. A higher upper boundary velocity would accelerate convergence due to a greater energy input and a severer mixing of the fluid inside cavity. The changes in dimensions of Ω but not of grids barely affected convergence behavior, reflecting a weak dependence of

stability performance on solution geometry at least for simple cases. Refinement on Δt would obviously decelerate convergence, while the nominal converged flow time was reduced from 10.92 seconds to 3.826 seconds, providing an approach to short-time investigation.

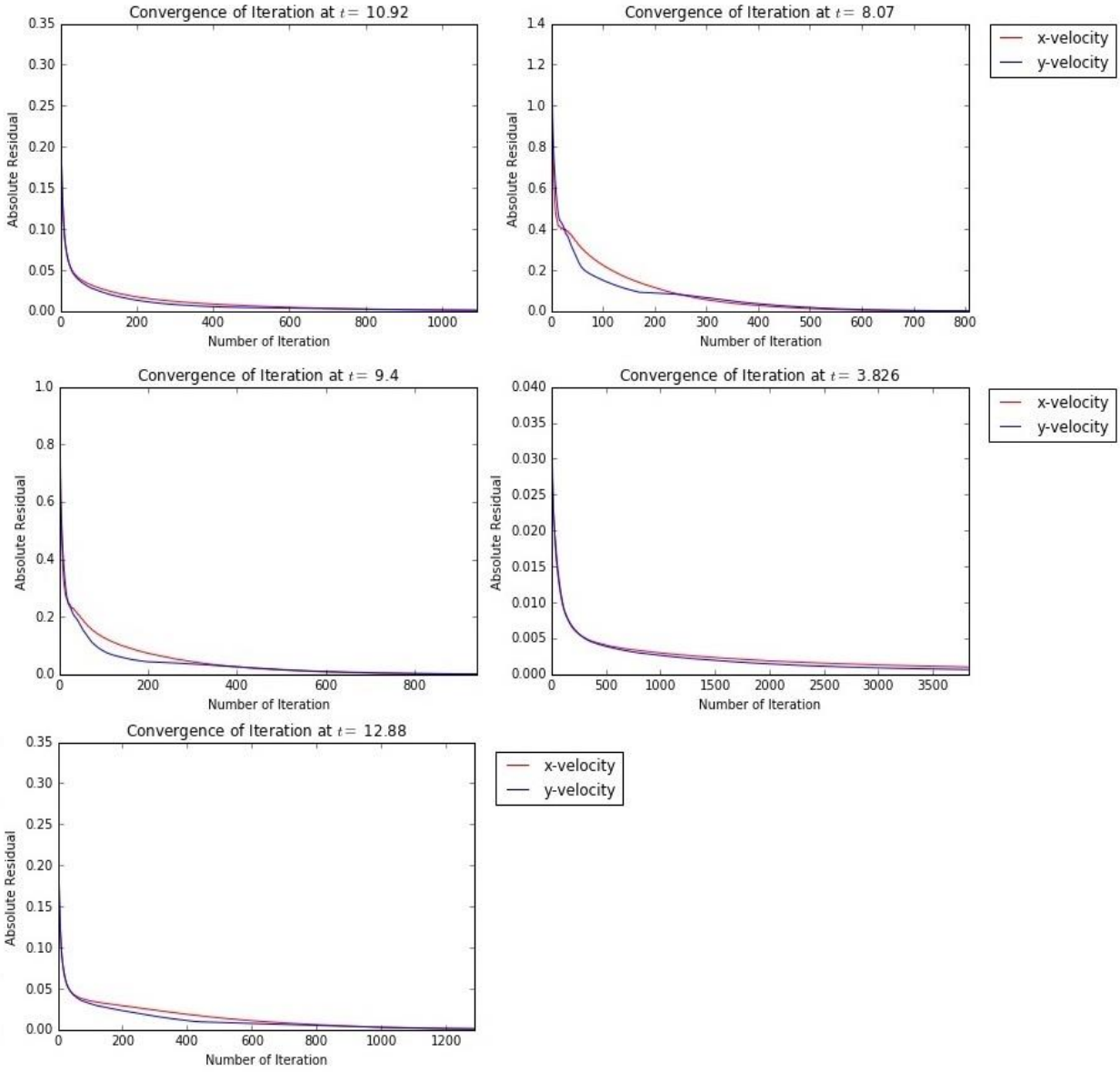


Fig. 3 Typical Converging Curve of Absolute Residuals (Order of Combinations)

4.2 Pressure and Velocity Distribution

When upper boundary began to flow above the still fluid inside cavity, fluid in the upper layer got accelerated by molecular viscosity, resulting in a pressure drop relative to the rest of the cavity as implied by Bernoulli equation. The pressure gradients then drove the cavity fluid floating up,

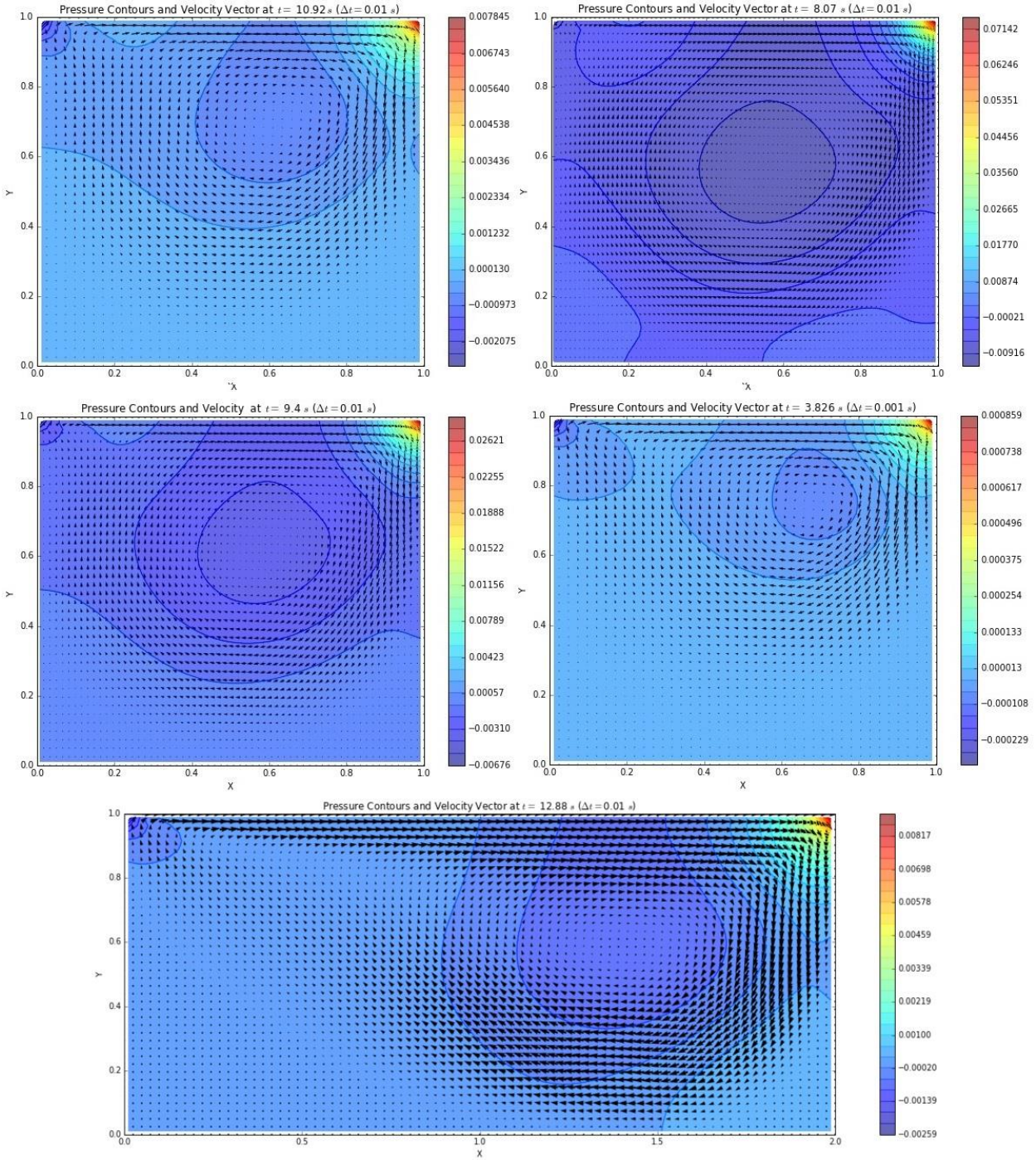


Fig. 4 Predicted Pressure and Velocity Distribution (Order of Combinations)

leaving “empty space” that was immediately and continuously filled by downstream fluid. The phenomenon kept happening until the flow reached rigid and impenetrable right boundary where it stagnated and sank to complement leaving fluid. Near the stagnation point, the pressure grew dramatically as a natural result of kinetic energy conversion. As a result, vortex was formed in the middle of the cavity. The predicted pressure contours and velocity vector distribution shown below were coincident with the theoretical analysis.

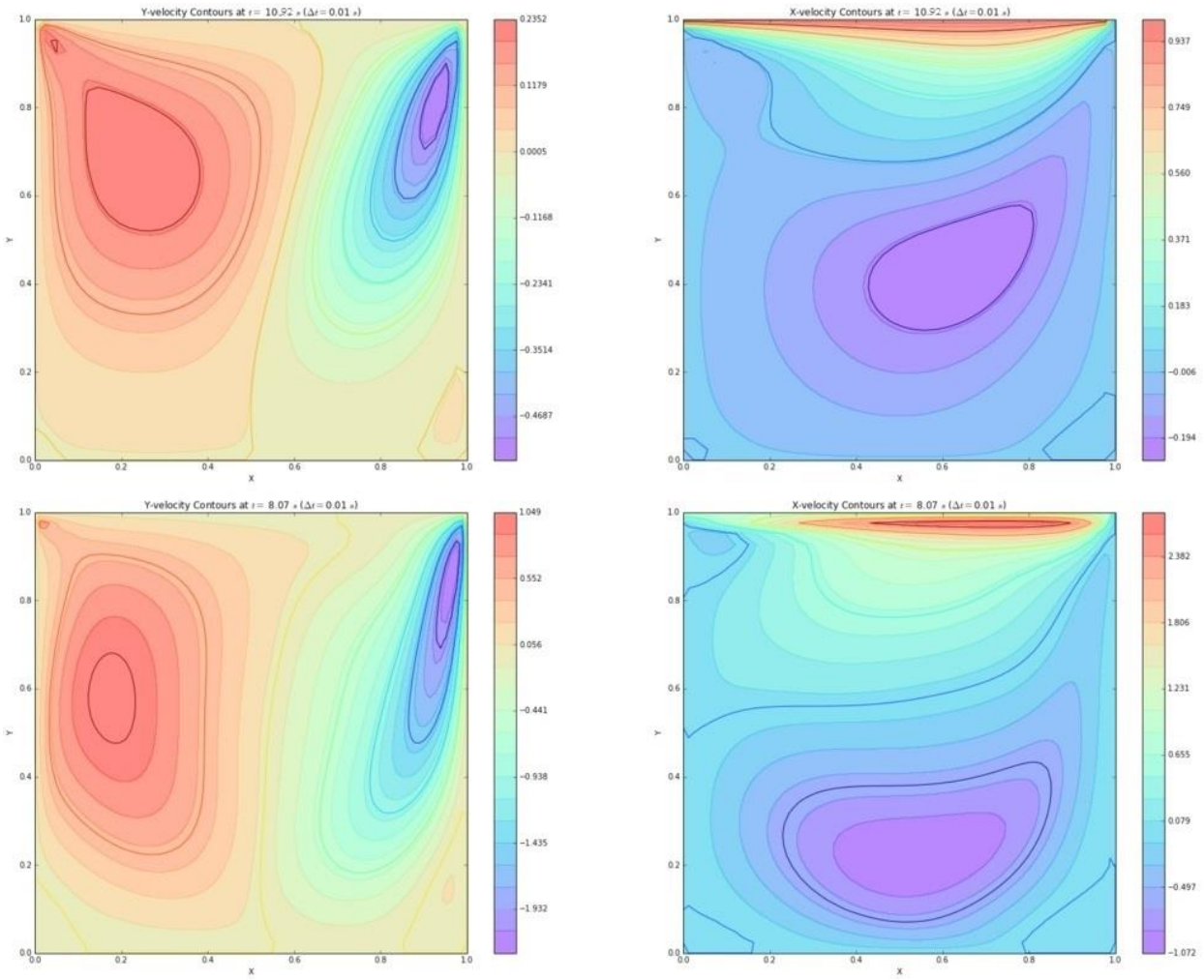
As before, grid size, time steps and upper boundary velocity were three significant dominants to performance of the predictions. A finer mesh would definitely produce more representative and somehow more accurate results. The influence of boundary conditions on the pressure and velocity distribution came from physical essence of the problem, higher velocity, higher pressure gradients and consequently severer mixing. Observation was also made that the maximum pressure, which occurred at stagnation point, would also increase with finer grids or higher boundary velocity. This was kind of responsible for the difference between combination 1 and 2. As stated previously, the smaller Δt provided more information on the development of the phenomenon in the cost of computing effort. The optimal combination of the factors depends on the specific problem and purposes and interest of the research.

4. 3 Velocity Components Contours

Combinations 1 and 4, 2 and 3 separately shared the similar velocity components (u and v) distribution, thus only three velocity contours were displayed in Fig.5. Basically, v was positive at the left half of Ω, meaning the fluid was floating, and negative at the rest half where flow reached stagnation point and sank. The circular shape of the velocity contours indicated that there were actually vortices formation within the cavity. The v contour became denser around both upper corners for all cases, implying a great velocity gradient due to the stagnation effect. While all other

v contours were concentrated, the contours for combination 5 got detached near $x = 0.5$, which could result from a slower bottom velocity and a consequently weaker *y*-pressure gradient due to enlarging of the domain. For *x*-component velocity, the contours were simple determined by competition of the pressure gradients and stagnation effects. Therefore, *u* contours possessed alike shapes and distribution, denser near the upper surface due to pressure and neat the corners because of stagnation. Special attention should be cast to the bottom corners where secondary vortex and even tertiary vortex, if velocity was large enough, would take place.

The above results were last compared to the those from benchmark papers [6], significant agreement was observed, demonstrating the reliability of the FDM proposed.



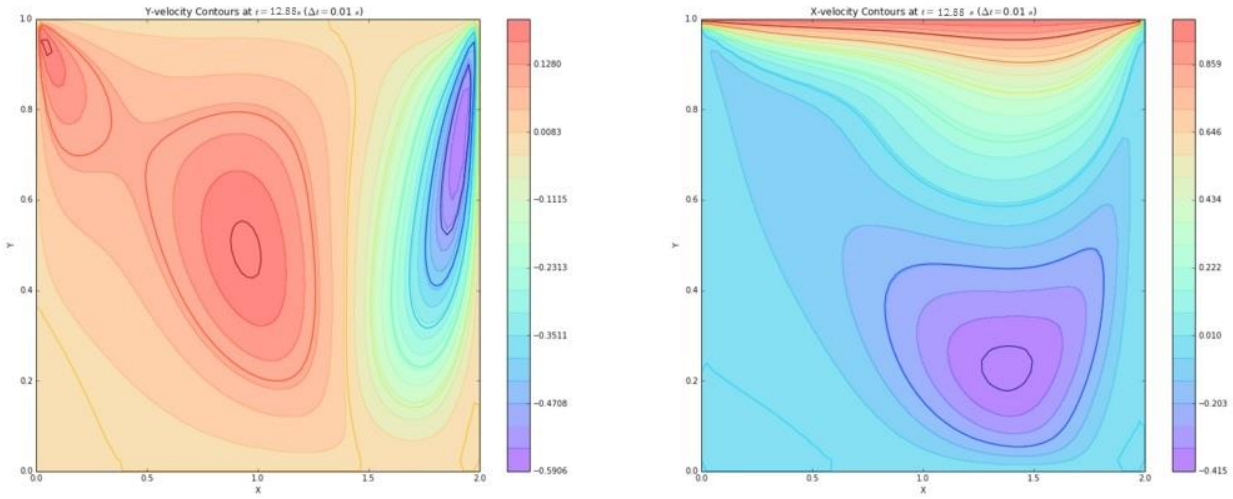


Fig .5 Velocity Components for Combination 1, 2, 5

5. CONCLUSION

This paper presented the derivation of two numerical discretization of 2-D unsteady Navier-Stokes equation using either FDM or FVM on a staggered grid system. The FDM discretization was based on the projection method with central difference in space and forward difference in time. While the FVM one was derived from SIMPLE algorithm with second-order upwind difference for convection terms, central difference for diffuse terms and forward difference for time. By applying Taylor series expansion for discretized equations, it can be shown that the methods were first-order accurate in time and second-order accurate in space.

The FDM was coded using Python script and validated by analyzing driven cavity flow. Results shows that grid sizes and boundary conditions were the dominant factors to influence pressure and velocity fields. The dimension of solution domain would have impact on velocity profiles but not on pressure or convergence behavior. Refinement on time step would decelerate the converging but could provide additional information on flow development.

Future efforts would be cast in achieving SIMPLE algorithm and optimizing compatibility and overall performance of both FVM and FDM by testing and improving the program on more complicated and various geometries and boundary conditions.

REFRENCE

[1] Thien Hiep Le et al. Pegase: A Navier-Stokes Solver for Direct Numerical Simulation of Incompressible Flows, International Journal for Numerical Methods in Fluids, 1997, VOL. 24, 833–861.

[2] Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Phys. Fluids. 1965; 8:2182-2189.

[3] Joaquim Peiro and Spencer Sherwin, Finite Difference, Finite Element and Finite Volume Methods for Partial Differential Equations, Handbook of Materials Modeling. Volume I: Methods and Models, 1–32.

[4] S. V. Patankar and D. B. Spalding, A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows, Int. J. Heat Mass Transfer, vol. 15, p. 1787, 1972.

[5] J. C. Mattingly and Ya. G. Sinai, An Elementary Proof of the Existence and Uniqueness Theorem for the Navier-Stokes Equations, Commun. Contemp. Math. 01, 497 (1999).

[6] Erturk E., Corke T.C. and Gokcol C., "Numerical Solutions of 2-D Steady Incompressible Driven Cavity Flow at High Reynolds Numbers", International Journal for Numerical Methods in Fluids, Vol 48, pp 747 - 774, 2005.

[7] Tu, Yeoh, and Liu, Computational Fluid Dynamics: A Practical Approach, Elsevier, 2013.

[8] Numerical Python Manual 1.10, http://docs.scipy.org/doc/numpy/reference/.

[9] Matplotlib User Guide, Release 1.5.1, http://matplotlib.org/Matplotlib.pdf.

APPENDIX

Please refer to attachment for Python source code for FDM.