IMPLEMENTATION OF CHORIN'S FRACTIONAL STEP METHODS FOR SOLVING INCOMPRESSIBLE 2D NAVIER STOKES EQUATIONS

Boyuan Lu
blu38@wisc.edu

December 18, 2021

ABSTRACT

This paper presents the comprehensive details of a numerical method - fractional step method to solve the two-dimensional incompressible Navier Stokes equations under two specific condition: lid-driven cavity, and vortex initial flow. Comparing with the results from other previous researcher, we prove that the solver we implemented is reliable and accurate. Several tricks were proposed under the framework of fractional step methods in the discussion part, showing that FSM is an editable and extendable method which can be coupled with other methods.

1 Introduction

The Navier-Stokes equations are groups of partial differential equations describing the movement of Newtonian fluid in a given space and time. The formation of NS-equation are two parts: continuity and momentum. Assuming two-dimensional, incompressible, constant viscosity, the controlling equation can be expressed as following equation groups:
Continuity:

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

Momentum in x and y direction:

$$\frac{\partial u}{\partial t}+\frac{\partial uu}{\partial x}+\frac{\partial uv}{\partial y}=-\frac{1}{\rho}\frac{\partial P}{\partial x}+\nu(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}) \tag{2}$$

$$\frac{\partial v}{\partial t}+\frac{\partial uv}{\partial x}+\frac{\partial vv}{\partial y}=-\frac{1}{\rho}\frac{\partial P}{\partial y}+\nu(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}) \tag{3}$$

The equations are really difficult to solve analytically since both u,v and P appears in everywhere of the equations, which makes the equations nonlinear and complex. Thus, people usually solve the NS equations using numerical methods. To build a numerical model, the equations can be interpreted into three types of problem:

- the hyperbolic problem (wave equation) refer to the convection terms in the equations.
- the parabolic problem (heat diffusion) for viscous term.
- the elliptic problem (Poisson equation) for the pressure terms.

This mixed equation can be solved in many efficient and smart ways. In this project, we will implement the fractional step methods (FSM) to solve the 2-dimensional incompressible Navier-Stokes equations. The idea of fractional step methods (FSM) is to split up the equations into its constituent pieces and alternate between advance equation in time. The simplest form of FSM with two steps can be expressed as following:

$$U^*=N_A(U^n,k) \tag{4}$$

$$U^{N+1}=N_b(U^*,k) \tag{5}$$

where $N_A(U^n, k)$ represents some one-step numerical method that solves $u_t = A(u)$ over a time step of length k.

The objective of this project is to implement the basic Chorin's fractional step method and build up a specific solver for two-dimensional incompressible Navier-Stokes equations under a predefined initial and boundary condition. Beyond the implementation of algorithm, we expect to evaluate and analyze our work by validating the result with other people's work, and assess the error and computational time between the different methods based on FSM framework.

2 Related works

The numerical methods for approximately solving viscous and imcompressible Navier Stokes equation was founded and developed in the 1960 since the origin of computational fluid dynamics [1]. Among those schemes, one of the most classic one was come up by Chorin [2] in 1967, which was the fractional steps method or projection method. The basic idea behind the fractional steps method is to make some estimations of the vector field (in NS equation, it is velocity field in momentum equation, Eq.2 and Eq.3) at the next time step and then correct this estimations as little as possible so that it satisfies the equation constraint (in NS equation, it is incompressibility in mass conservation, Eq.1) as much as possible [3].

Lots of researchers have devoted their mind into analysing, and modifying FSM. Crandall and Majda analyzed the stability, accuracy, and convergence of the basic fractional step algorithm under various scheme such as monotone, Lax-Wendroff, and Glimm and they found out the weak stability among the FSM [4]. On the other hand, modified MSF like Exact FSM [3], Spline FSM [5], Lagrangian FSM [6] showed applicability of FSM to various practical issues.

3 Methods

3.1 Domain setup

The basic setup for the boundary and initial condition are predefined as below.
Domain

- use a rectangular domain with $L_x = 30.0$ and $L_y = 30.0$
- Boundary condition: left side: use an inlet condition of $u = 1$,$v = 0$;top and bottom: use a slip condition: $\frac{\partial u}{\partial y} = 0$,$v = 0$; right boundary: use the convection condition for all variables

initial conditions

- initial condition will be a vortex hat equation (shown in figure 1):

$$\psi(x, y, t = 0) = \Gamma \exp(\frac{-(x - x_0)^2 - (y - y_0)^2}{2\sigma^2}) \tag{6}$$

- parameter: kinematic viscosity is $\nu = 0.001$; $x_0 = 0.5L_x$ and $y_0 = 0.5L_y$;$\Gamma = 1$,$\sigma = 2.5$

To coupled with the velocities u, v and pressure P, we applied a staggered grid based on the idea of control volume methods. A staggered grid is one in which the velocities and pressure are located at different positions. Figure 1 presents the schematic layout of staggered grid units for the defined locations of velocities and pressures. Every cell has couple with one u velocity, and one v velocity, which defined as the surface velocity at the bottom and left boundary of cell, as well as the pressures will be defined in the center of cell. The advantage of such setting is that it allows for the u, v, P solution to have a uniform indices and easy for visualization by the grid. The disadvantage for staggered grids is that it requires the half index notation like $i + 1/2$ but most programming language only take integer indexing. However, we use the cells indexing rather than nodes indexing so that it won't be a trouble for indexing at all.
Besides the nx, ny cells presented in fig1, we also defined multiple "ghost cells" padding around the domain. The purpose of setting ghost cells is to define the boundary condition for the velocities and pressures. We will discuss more details of the boundaries conditions at session 3.5.

3.2 Fractional steps for the time discretization

To discretize the partial t terms in two momentum equation in Eq.2 and Eq.3, we firstly applied the Explicit Euler scheme which can be written as,

$$\frac{u^{n+1} - u^n}{dt} = -u^n \frac{\partial u^n}{\partial x} - v^n \frac{\partial u^n}{\partial y} - \frac{1}{\rho} \frac{\partial P^{n+1}}{\partial x} + \nu(\frac{\partial^2 u^n}{\partial x^2} + \frac{\partial^2 u^n}{\partial y^2}) \tag{7}$$

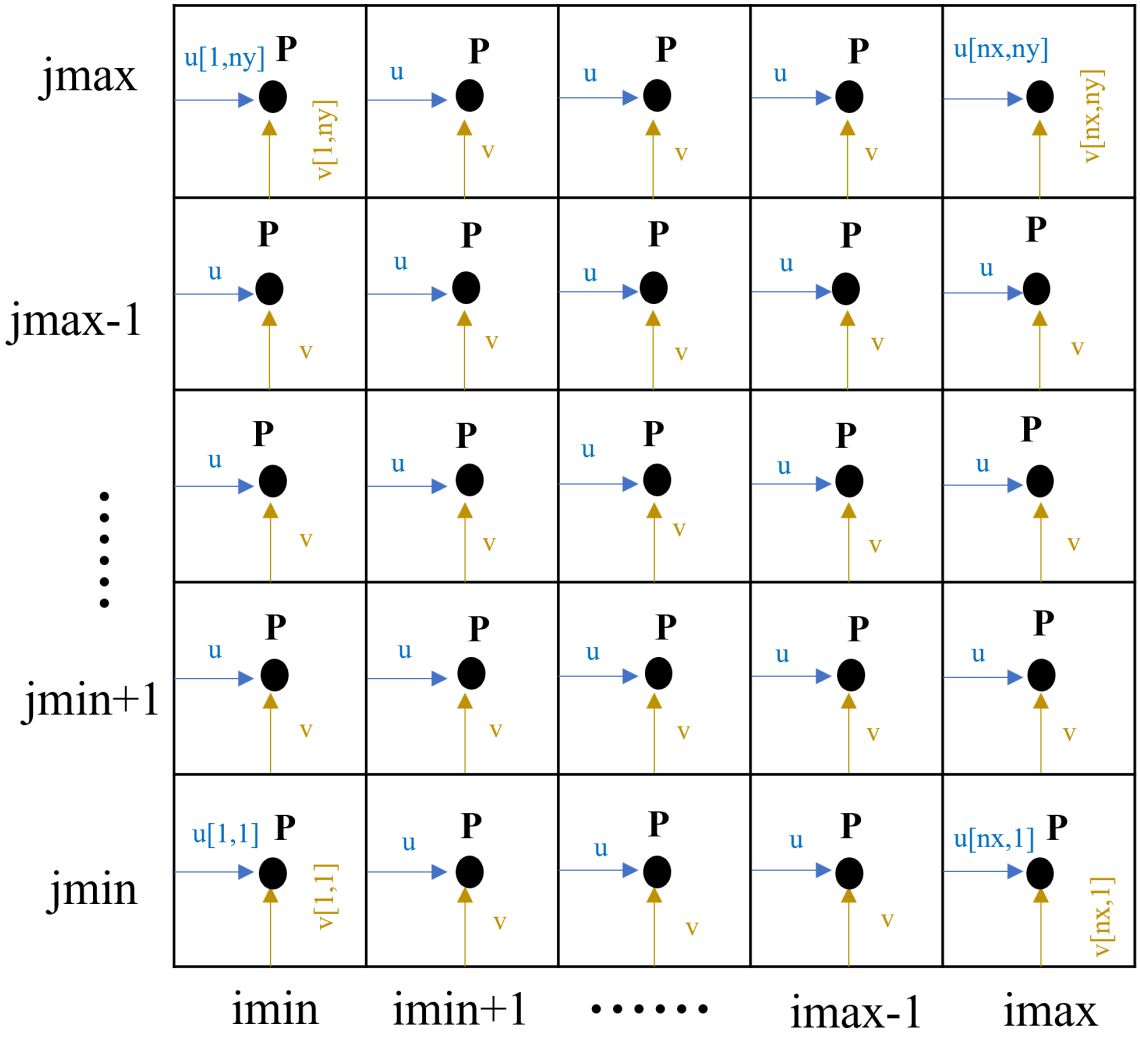


Figure 1: computational staggered grids for our CFD solver. The locations of u, v, P are presented in each cell with indices.

$$\frac{v^{n+1}-v^n}{dt} = -u^n\frac{\partial v^n}{\partial x} - v^n\frac{\partial v^n}{\partial y} - \frac{1}{\rho}\frac{\partial P^{n+1}}{\partial x} + \nu\left(\frac{\partial^2 v^n}{\partial x^2} + \frac{\partial^2 v^n}{\partial y^2}\right) \tag{8}$$

However, ordinary Explicit Euler has its weakness as it cannot be directly solved because P^{n+1} is not known so far, and solution might not be guaranteed to be computed out due to instability of the Explicit Euler. To solve the temporal discretization, we need to introduce predictor-corrector schemes coupling with the Eq.1 of continuity conservation by splitting the Explicit Euler into two steps.

- the first step, also known as predictor step is to calculate the intermediate velocity u^*, v^* but only include convection and diffusion terms using the Explicit Euler,

$$\frac{u^*-u^n}{dt} = -\left(u^n\frac{\partial u^n}{\partial x} + v^n\frac{\partial u^n}{\partial y}\right) + \nu\left(\frac{\partial^2 u^n}{\partial x^2} + \frac{\partial^2 u^n}{\partial y^2}\right) \tag{9}$$

$$\frac{v^*-u^n}{dt} = -\left(u^n\frac{\partial v^n}{\partial x} + v^n\frac{\partial v^n}{\partial y}\right) + \nu\left(\frac{\partial^2 v^n}{\partial x^2} + \frac{\partial^2 u^n}{\partial y^2}\right) \tag{10}$$

- the second step, also known as corrector step, is to solve velocities u^{n+1}, v^{n+1} at the next moment, but includes the pressure terms exclusively,

$$\frac{u^{n+1}-u^*}{dt} = -\frac{1}{\rho}\frac{\partial P^{n+1}}{\partial x} \tag{11}$$

$$\frac{v^{n+1}-v^*}{dt} = -\frac{1}{\rho}\frac{\partial P^{n+1}}{\partial y} \tag{12}$$

Taking the derivative of the Eq.11, Eq.12, and combine with Eq.1, we can derive the gradient of pressure projects u, v onto a divergence free field that satisfies continuity:

$$\frac{\partial^2 P^{n+1}}{\partial x^2} + \frac{\partial^2 P^{n+1}}{\partial y^2} = \frac{\rho}{dt}(\frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y}) \tag{13}$$

The Eq.13 actually is in the form of Poisson equation in two-dimension: $u_{xx} = f$, which is known as pressure Poisson equation. By solving the pressure Poisson equation, pressure can be computed and updated. Then, apply the pressure into Eq.11 and Eq.12 to compute the u^{n+1} and v^{n+1} .

3.3 Momentum discretization for predictor step

The Eq.9 and Eq.10 can be computed using the finite difference method. For the convection term, we apply the center difference scheme, while for the diffusion term, we applied the 2nd order center difference scheme. Here are the discretization for the u velocity in predictor step,

$$u^* = u^n + dt(\nu(\frac{\partial^2 u^n}{\partial x^2} + \frac{\partial^2 u^n}{\partial y^2}) - (u^n\frac{\partial u^n}{\partial x} + v^n\frac{\partial u^n}{\partial y})) \tag{14}$$

where the viscous term and diffusive terms can be written in each cell (i, j) we define in 3.1 as:

$$\frac{\partial^2 u^n}{\partial x^2} = \frac{u(i-1,j) - 2u(i,j) + u(i+1,j)}{dx^2} \tag{15}$$

$$\frac{\partial^2 u^n}{\partial y^2} = \frac{u(i,j-1) - 2u(i,j) + u(i,j+1)}{dy^2} \tag{16}$$

$$u^n\frac{\partial u^n}{\partial x} = u(i,j)(\frac{u(i+1,j) - u(i-1,j}{2dx}) \tag{17}$$

$$v^n\frac{\partial u^n}{\partial y} = \frac{1}{4}(v(i,j) + v(i-1,j+1) + v(i,j+1) + v(i-1,j))(\frac{u(i+1,j) - u(i-1,j)}{2dx}) \tag{18}$$

The same approach can be applied to v velocity,

$$v^* = v^n + dt(\nu(\frac{\partial^2 v^n}{\partial x^2} + \frac{\partial^2 v^n}{\partial y^2}) - (u^n\frac{\partial v^n}{\partial x} + v^n\frac{\partial v^n}{\partial y})) \tag{19}$$

where the viscous terms and diffusive terms are,

$$\frac{\partial^2 v^n}{\partial x^2} = \frac{v(i-1,j) - 2v(i,j) + v(i+1,j)}{dx^2} \tag{20}$$

$$\frac{\partial^2 v^n}{\partial y^2} = \frac{v(i,j-1) - 2v(i,j) + v(i,j+1)}{dy^2} \tag{21}$$

$$u^n\frac{\partial v^n}{\partial x} = \frac{1}{4}(u(i,j) + u(i+1,j) + u(i,j+1) + u(i+1,j+1))(\frac{v(i+1,j) - v(i-1,j)}{2dx}) \tag{22}$$

$$v^n\frac{\partial v^n}{\partial y} = v(i,j)(\frac{v(i+1,j) - v(i-1,j}{2dx}) \tag{23}$$

Notice that, in Eq.18 and Eq.22, the viscous term $v\frac{\partial u}{\partial x}$ was discretized as the central average among the four points $\frac{1}{4}(v(i,j) + v(i-1,j+1) + v(i,j+1) + v(i-1,j))(\cdots)$. The reason for this is to couple the u, v velocities into the same location, which shown in fig 2 below.

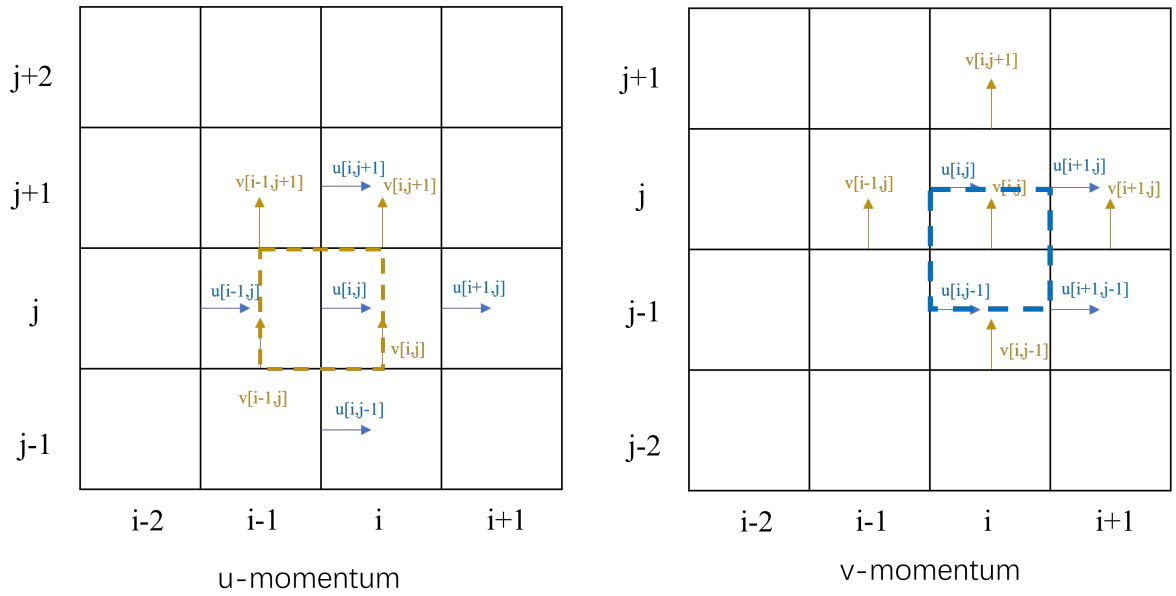


Figure 2: the momentum discretization for u(i,j) (left) and v(i,j) (right) directions. Each mementum discretization will involve nearest 4 velocities with the same directions plus 4 velocities with orthogonal direction.

3.4 Pressure discretiztion and velocities updates for corrector step

The most computing-consuming part is to compute the pressure Poisson equation,

$$\nabla^2 P = R \tag{24}$$

where ∇^2 is the Laplatian operator, and R is referred to right hand side of pressure Poisson equation which is $R = -\frac{\rho}{dt}\nabla u^*$.
To solve Eq.24, we applied the 2d finite difference scheme in the format of Jacobi matrices for all interior nodes. To discretize the Laplacian pressure gradient, 2nd order central difference is applied as following,

$$\nabla^2 P = \frac{P(i-1,j) - 2P(i,j) + P(i+1,j)}{dx^2} + \frac{P(i,j-1) - 2P(i,j) + P(i+1,j+1)}{dy^2} \tag{25}$$

For the right hand side R, the u^*, v^* have already been computed out in predictor step. Therefore, we can apply the upwind difference directly as,

$$R = \frac{\rho}{dt}\left(\frac{u^*(i+1,j) - u^*(i,j)}{dx} + \frac{v^*(i,j+1) - v^*(i,j)}{dy}\right) \tag{26}$$

With the help of the pressure Poisson solver using the Jabobi matrices, we can solve the pressure at each cells in the form of $Ax = B$ as following,

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \frac{-1}{dx^2} & D & \frac{-1}{dx^2} & \cdots & \cdots & \cdots & \frac{-1}{dy^2} & \cdots & 0 \\ \vdots & \cdots & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \frac{-1}{dy^2} & \ddots & \frac{-1}{dx^2} & D & \frac{-1}{dx^2} & \ddots & \frac{-1}{dy^2} & 0 \\ \vdots & \cdots & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots & \vdots \\ 0 & \frac{-1}{dy^2} & \cdots & \cdots & \cdots & 0 & \frac{-1}{dx^2} & D & \frac{-1}{dx^2} \\ 0 & 0 & 0 & \cdots & \frac{-1}{dy^2} & \cdots & 0 & \frac{-1}{dx^2} & D \end{bmatrix} \begin{bmatrix} P(1,1) \\ P(2,1) \\ \vdots \\ P(i,j) \\ \vdots \\ P(nx*ny, nx*ny-1) \\ P(nx*ny, nx*ny) \end{bmatrix} = \begin{bmatrix} R(1,1) \\ R(2,1) \\ \vdots \\ R(i,j) \\ \vdots \\ R(nx*ny, nx*ny-1) \\ R(nx*ny, nx*ny) \end{bmatrix} \tag{27}$$

Notice that terms D in Eq.27 are different at different location of boundaries. For the boundary pressure, there are some extra jobs to do and we will introduce the boundary condition in the session 3.5.

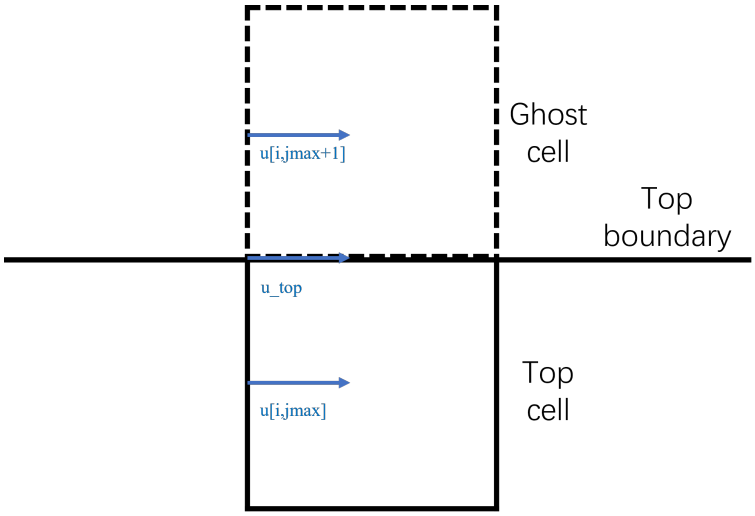


Figure 3: Use the top cell and ghost cell to determine the u velocity boundary condition.

At the final step of corrector, we can updates Eq.11 and Eq.12 for the u, v using the downward difference of pressure,

$$u^{n+1}(i,j) = u^*(i,j) - \frac{dt}{\rho}\frac{P(i,j) - P(i-1,j)}{dx} \tag{28}$$

$$v^{n+1}(i,j) = v^*(i,j) - \frac{dt}{\rho}\frac{P(i,j) - P(i,j-1)}{dy} \tag{29}$$

3.5 Boundary conditions

In this case, we will introduce how to specify the boundary condition for both velocities fields and pressures fields. First of all, it was because the staggered cells that some of velocities might not be defined at boundaries of the domain. The most trick part, for example, is that the velocities u at the top of cells are defined $dy/2$ away from the top boundary. To deal with this tissue, here comes with the ghost cells we mentioned at 3.1. We assumed that the velocities at the boundary can be interpolated as the average values between the velocity at top cell and ghost cell out of top cells. Figure 3 shows the schematic of the how to use ghost cell to interpolate the boundary velocities. Therefore, with help the ghost cell and ghost velocites, we can right boundary condition as,

$$u_{top} = (u(i,jmax) + u(i,jmax+1))/2 \tag{30}$$

For the pressure boundary condition, we can also applied the pressure in ghost cell in fig 4.

In pressure Poisson discretization (Eq. 25 and Eq. 26) we have the following,

$$\frac{(\cdots)}{dx^2} + \frac{P(i,jmax+1) - 2P(i,jmax) + P(i,jmax)}{dy^2} = R(i,jmax) \tag{31}$$

By assuming the pressure gradient is zeros at boundary, we can modified the Eq.30 as following,

$$\frac{(\cdots)}{dx^2} + \frac{P(i,jmax) - P(i,jmax-1)}{dy^2} = R(i,jmax) \tag{32}$$

With the help of Eq.32, we can interpret the terms D in Eq.27,

$$D = \begin{cases} \frac{2}{dx^2} + \frac{2}{dy^2}\,, & \text{interior cells} \\ \frac{1}{dx^2} + \frac{2}{dy^2}\,, & \text{vertical (y) boundaries} \\ \frac{2}{dx^2} + \frac{1}{dy^2}\,, & \text{horizontal (x) boundaries} \\ \frac{1}{dx^2} + \frac{1}{dy^2}\,, & \text{corner cells at boundaries} \end{cases} \tag{33}$$

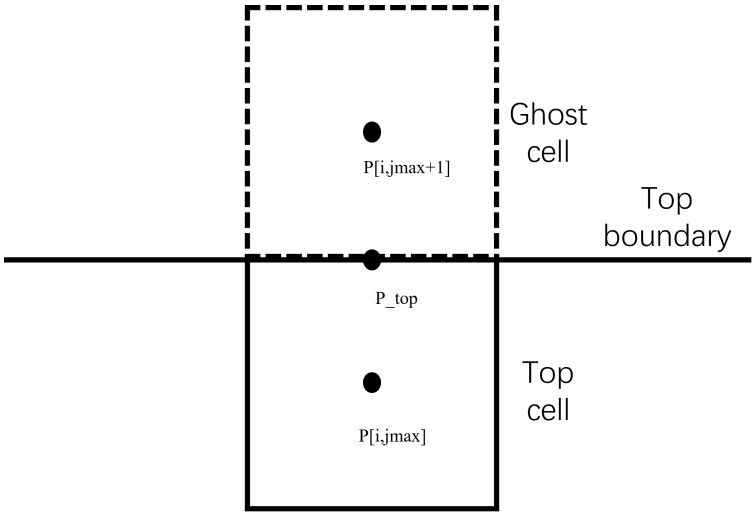


Figure 4: using the top cell and ghost cell to determine the pressure bondary condition

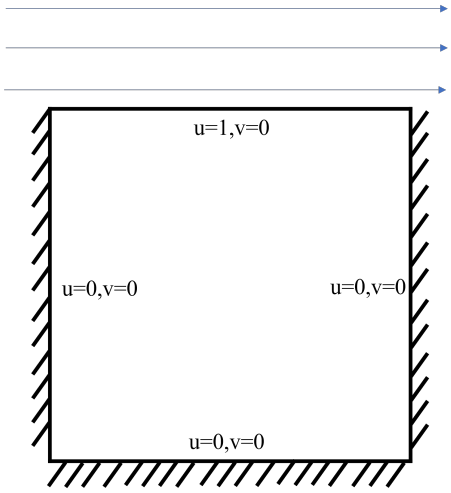


Figure 5: the space and velocities setups of two-dimensional lid-driven cavity problem

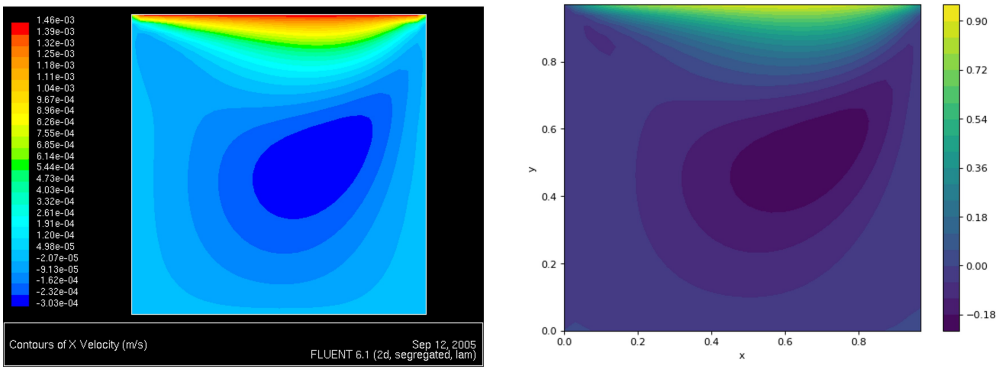
4 Result

4.1 Validation for the solver

The analytical solution of Navier Stokes equations is still an unresolved puzzle in the world. Therefore, it is impossible to validate the solver with the pre-define initial condition because the solution is too hard to present. Fortunately, there are a couples of scenarios that been used widely in the past decades as the validation scene for the NS equation, one of them is known as the lid-driven cavity problem. The lid-driven cavity problem is simple and in two-dimensional. The standard case is fluid contained in a square domain with Dirichlet boundary conditions on all sides, with three stationary sides and one moving side (See Figure 5).

This problem is a nice one for testing for several reasons. First, as mentioned above, there is a great deal of literature to compare with. Second, the (laminar) solution is steady. Third, the boundary conditions are simple and compatible with most numerical approaches. Note that this is not necessarily the case for finite element methods, in which difficulties may arise at the corner intersections of the moving wall and the stationary wall. The Fig 6 Shown below is the comparison between the results of u velocities from solver we wrote verse the results form website[1] which computed by the commercial Fluent code.

[1] https://www.cfd-online.com/Wiki/Liddriven_cavity_problem



Fluent results for u velocity

Our results for u velocity

Figure 6: Comparison between Fluent results (left figure) with our results (right figure) in terms of u velocities for $Re = 100$ at $T = 5s$.

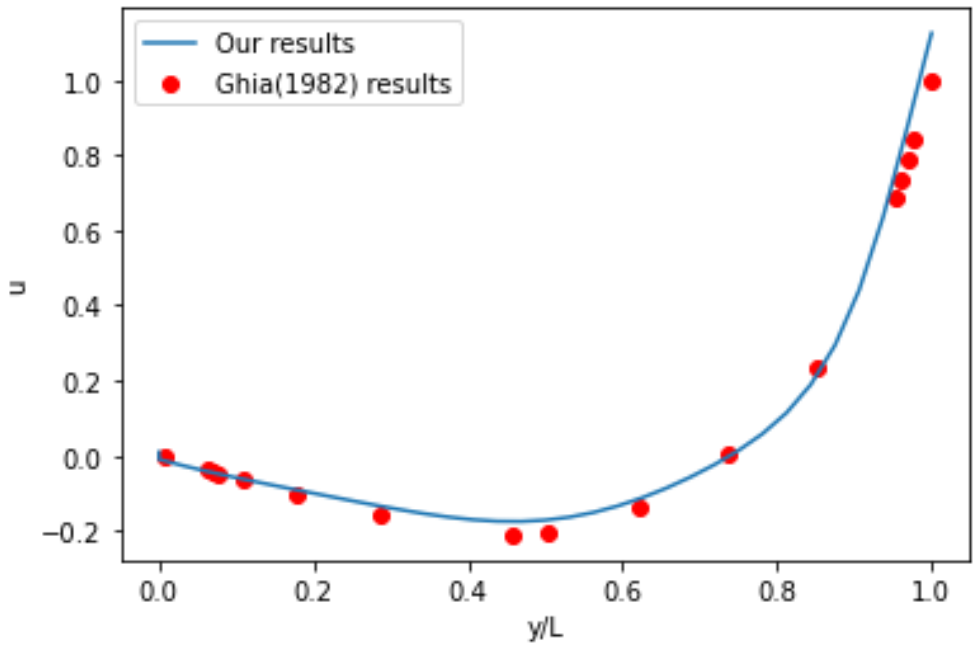
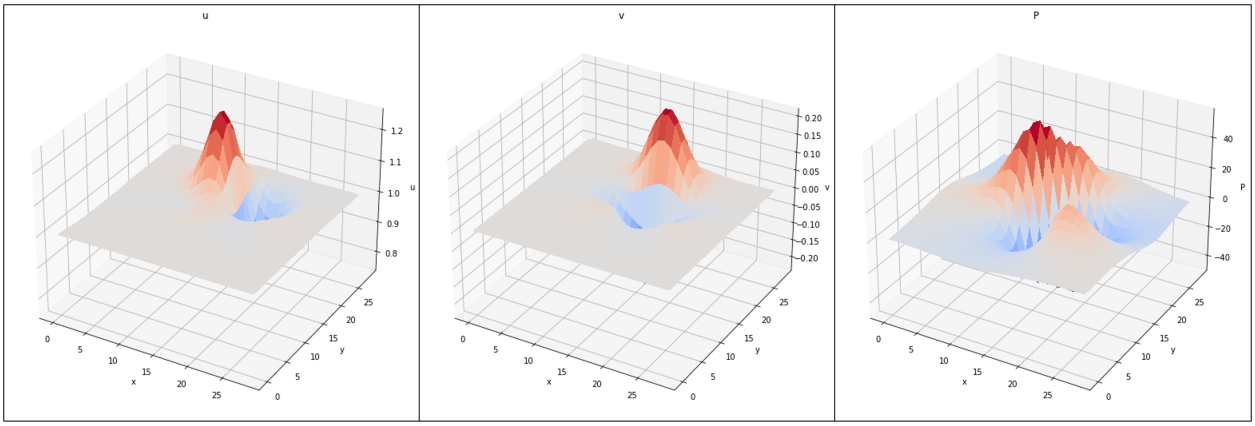


Figure 7: u velocities distribution at center vertical line of domain between two results (our vs Ghia's)

Notice that the results are sampled at $t = 5s$ at step size $dt = 0.001$ with Reynolds number $Re = 100$ (Reynolds number defines the degree of fluids turbulence) under 32*32 grids for $L = 1$ spatial domain. Our results roughly are the similar with the results from the Fluent software. Unfortunately, we do not have the exact time domain of Fluent setting so that we cannot compare two results from grid to grid. Good thing is, as we mentioned previously, there are lots of literature we can somehow use to check the validity of our results. With the help from the previous research[7], we can match out results in a reasonably good agreement. Figure 7 shows the u velocities varied in the center line of domain between our results and Ghia (1982) results, and our results matches with Ghia's results, which means that our solver is accurate and valid.

4.2 Results under the pre-specified initial condition

For the vortex condition we pre-defined in session 3.1, we can derive the u, v velocities based on the derivatives of streamline function,



u velocity v velocity pressure velocity

Figure 8: u velocity(left), v velocity(middle), and pressure (right) fields under the open vortex condition.

$$u_0 = \frac{\partial \psi}{\partial y} = \Gamma \exp(\frac{-(x-x_0)^2 - (y-y_0)^2}{2\sigma^2}) \frac{-(y-y_0)}{\sigma^2} \tag{34}$$

$$v_0 = -\frac{\partial \psi}{\partial x} = \Gamma \exp(\frac{-(x-x_0)^2 - (y-y_0)^2}{2\sigma^2}) \frac{(x-x_0)}{\sigma^2} \tag{35}$$

Here is the results of pressure fields under the Eq.34 and Eq.35 conditions in $T = 5$s. The u, v velocities are corrected based on the my knowledge. However, notice that the plain of pressure field around the vortex seems a little not smooth. We believed that the possible reasons for that might related to the coarse grids or some tiny mistakes to the open boundary on the left. Due to limitation of time, we were not able to fix this error completely.

5 Discussion for the modifications of FSM

In Eq.7, we discretize the time step using the Explicit Euler, which utilize the current stages to anticipate the future stage. According to its Courant-Friedrichs-Lewy (CFL) condition, it becomes instable when $u\Delta t/\Delta x > 1$. Facing with this issue, we can somehow improve the stability by introducing the Adams Bashforth and Crank Nicolson scheme for the time splitting when predicting the u^*. In general, we applied the Adams Bashforth for the convection terms applying the current time n and previous time $n-1$. On the other hand, we also applied the Crank Nicolson for the diffusion term ultilizing the current time n and future time $n+1$. Here is the scheme in details,

$$\frac{u^* - u^n}{dt} = -(\frac{3}{2}C^n - \frac{1}{2}C^{n-1}) + \frac{1}{2}(D^{n+1} - D^n) \tag{36}$$

Where C refers to the viscous term and D refers to the diffusion term,

$$C = u\frac{\partial u}{\partial x} + v\frac{\partial v}{\partial y} \tag{37}$$

$$D = \nu(\frac{\partial^2 u^n}{\partial x^2} + \frac{\partial^2 u^n}{\partial y^2}) \tag{38}$$

Notice that the Courant-Friedrichs-Lewy (CFL) condition for the AB2-CN scheme is unconditional stable due to its implcicit-explicit complexity zone.

Another popular modification for fractional step methods is the RK2, which refers to the 2nd order Runge Kutta methods. The general idea of RK2 trick is to apply a multistage at preditor step to acheive a higher accuracy. More specific, in the first stage, generate an intermediate value that approximate the u^* using explicit Euler,

$$\frac{u^{*(i)} - u^n}{dt} = H_x^n \tag{39}$$

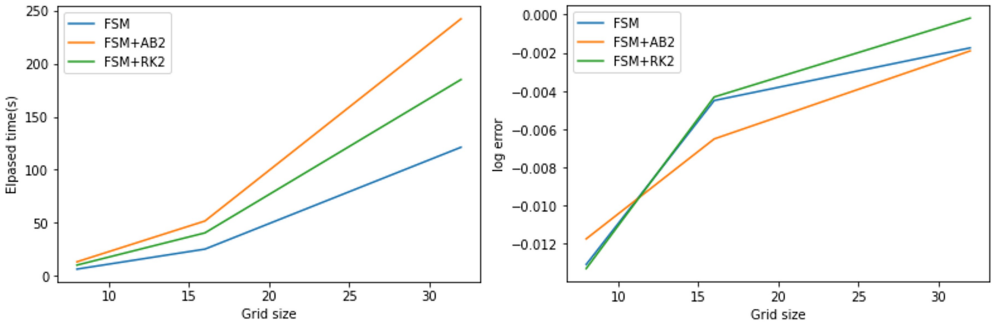


Figure 9: The elapsed time (left) and log error (right) under different grid size for three FSM methods

where H_x^n is the combination of the viscous term and diffusion term $-C_x^n + D_x^n$. In the second step, evaluate the function at the mid point to estimate the whole slope of time over full time step,

$$\frac{u^{*(ii)} - u^n}{dt} = \frac{1}{2}(H_x^{(i)} + H_x^n) \tag{40}$$

where H_x^n is in Eq. 37 and Eq. 38, and $H_x^{(i)}$ is expressed as,

$$H_x^{(i)} = -(u^{*(i)}\frac{\partial u^{*(i)}}{\partial x} + v^{*(i)}\frac{\partial v^{*(i)}}{\partial y}) + \nu(\frac{\partial^2 u^{*(i)}}{\partial x^2} + \frac{\partial^2 u^{*(i)}}{\partial y^2}) \tag{41}$$

Here are the results of elapsed time and log error in terms of different grid size for three methods - Basic FSM, FSM + AB2, FSM + RK2, respectively. For the time count, it is a little hard to trace the iterations for the computations so that we used the elapsed time in replaces. Among the three methods, the AB2 is the most time consuming methods and RK2 is the most accurate method. Notice that the RK2 use moderate computing time but achieve the highest accuracy, which makes the 2nd Runge Kutta method popular in the field of Computational Fluid Mechanics.

6 Conclusion

In this paper, we introduced how to use the ideas of predictor-corrector scheme,also known as fractional step method, to solve two-dimensional incompressible Navier Stokes equations. We defined two scenarios. One is testing case, called the lid-driven cavity problem. Compared with results from commercial Fluent and previous studies, we validated and tested our solver. Slightly modified the initial conditions and boundaries conditions, we applied the second scenario - vortex case. The results is reasonable and good. We also discussed several tricks for time intergrations: the 2nd order Adams Bashforth/Crank Nicolson, and 2nd order Runge Kutta. Three methods showed their pros and cons, and we decide that the 2nd order is the highest accuracy but in a economical computing time in three methods. Throughout this report, we developed the knowledge of one specific numerical method, which will guide our research in the future.

References

[1] J Blasco, R Codina, and A Huerta. A fractional-step method for the incompressible Navier-Stokes equations related to a predictor-multicorrector algorithm. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, 28(10):1391–1419, DEC 30 1998.

[2] AJ CHORIN. NUMERICAL SOLUTION OF NAVIER-STOKES EQUATIONS FOR AN IMCOMPRESSIBLE FLUID. *BULLETIN OF THE AMERICAN MATHEMATICAL SOCIETY*, 73(6):928–&, 1967.

[3] J. Perot and Venkataramanan Subramanian. Exact fractional step methods for solving the incompressible navier-stokes equations. 01 2002.

[4] M CRANDALL and A MAJDA. THE METHOD OF FRACTIONAL STEPS FOR CONSERVATION-LAWS. *NUMERISCHE MATHEMATIK*, 34(3):285–314, 1980.

[5] Wang Pu. Spline method of fractional steps in numerical model of unsteady natural convection flow at high rayleigh number. *Numerical Heat Transfer*, 11(1):95–108, 1987.

[6] Christoph Börgers and Charles S Peskin. A lagrangian fractional step method for the incompressible navier-stokes equations on a periodic domain. *Journal of Computational Physics*, 70(2):397–438, 1987.

[7] U GHIA, KN GHIA, and CT SHIN. HIGH-RE SOLUTIONS FOR INCOMPRESSIBLE-FLOW USING THE NAVIER STOKES EQUATIONS AND A MULTIGRID METHOD. *JOURNAL OF COMPUTATIONAL PHYSICS*, 48(3):387–411, 1982.