Channel Flow Over a Square Cylinder
ASE382-Q11 Final Project

William Jo

May 16, 2022

1 Introduction

The main goal of this class project is to outline the development of a modifiable computational fluid dynamics solver for unsteady channel flows based on two different fractional-step methods of both low and high order written in FORTRAN 90. This project attempts to solve the Navier-Stokes equations for a square cylinder in a channel flow geometry, but due to a time constraint of five weeks given for this project, work has been limited to the verification of the solver through verification of Plane Poiseuille flow and an attempt to solving a square cylinder in channel flow. While the project proposal involves the analysis of flow for a square cylinder in a channel flow geometry, this report will outline place a square cylinder or a rectangular cylinder within a channel and will walk through how boundary conditions are modified when such changes are to be added into the computational solver. This report will also discuss the issues involving the use of both lower order and higher order fractional step methods for flow analysis, the issue of ineffectual boundary conditions in channel flow written in prior literature and how they have been solved in this report, and finally existing issues of boundary conditions during the implementation of the square cylinder and potential solutions that can be modified in this project for a future student that might want to work on this project.

2 Literature Survey

2.1 Square Cylinder

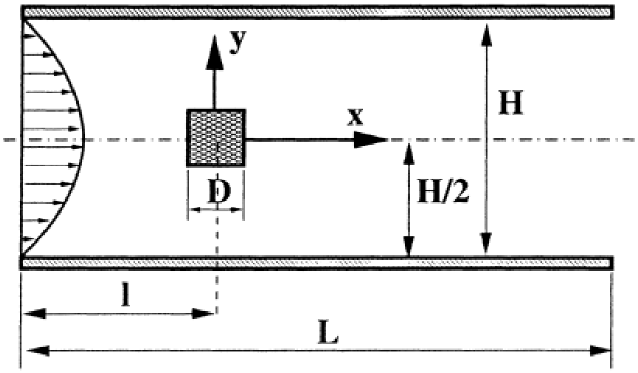

Figure 1: Description of the square cylinder in a channel problem, from Breuer et al. (1999) [1]

The objective of this project is to computationally solve flow past a square or rectangular cylinder that is placed halfway between two parallel plates, as described in figure 1. Deliverables that were proposed for this project includes the quantitative analysis of the accuracy of this proposed solver compared to previous research and literature through computations of force coefficients and recirculation lengths, and to also understand the effects of the downstream flow field when stretching or contracting in multiples of the dimension the square cylinder along the direction of the incoming channel flow. However, due to time constraints, the recirculation lengths from the computational solver are compared to data and analysis of the flow without the cylinder are conducted as a method of verification.

The setup and results produced by Breuer et al. (1999) serve as the benchmark for verifying the accuracy of the computational solver for the 2D square cylinder problem. Breuer et al. (1999) has established that the range of Reynolds numbers where laminar flow transitions from a steady to an unsteady form is between $56 \leq Re \leq 57$ [1]. For the steady case, that is $Re \leq 56$, Breuer et al.'s study has also established an empirical form to the recirculation zone length, with this equation: $L_r/D = 0.065 + 0.0554 \times Re$, where L_r is the length of the recirculation region, D is the length of the square cylinder, and Re the Reynolds number [1]. Literature from Dhiman et al. (2004) [2], Rahnama Hadi-Moghaddam (2005) [3], and Sen et al. (2010) [4], were able to verify the linear relationship between the reattachment bubble length versus Re for the Reynolds number range of $0 \leq Re \leq 50$ for a similar problem setup.

2.2 CFD Approaches

Prior, heavily cited works regarding similar flows such as plane Poiseuille flow and backwards facing step such as those from Breuer et al. (1999) and Kim et al. (2000) use the finite volume method (FVM) to solve the incompressible, unsteady Navier-Stokes equations [1][5]. Thus for the scope of this project, the Navier-Stokes equations will be solved using finite volume approaches. One of these FVM approaches to solving incompressible Navier-Stokes equations is through the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm from Spalding & Pathankar (1983) [6]. In this method, the pressure and velocity terms are linked, and the pressure is iteratively corrected until convergence is reached. In terms of grid properties, this grid will need to be staggered. Fluid properties, such as velocities are updated using updated pressures. The disadvantages of the SIMPLE algorithm is that it can only be used for steady-state problems, so while analysis of $Re \leq 50$ is valid using the SIMPLE algorithm, the unsteady cases when $Re > 60$ must include time-stepping.

A master's thesis written by Hines (2008) comparing SIMPLE and time integration methods for an incompressible, unsteady flow solver modifies the SIMPLE algorithm to include time stepping, providing algorithms for schemes such as SIMPLE, which solves for the pressure correction and velocity implicitly [7]. Another method related to the SIMPLE is a low order fractional step method from Saad (2019) [8] and a fractional-step algorithm pioneered by Kim and Moin (1985) [9]. Unlike in SIMPLE, both the lower order method from Saad and from Kim and Moin has the solution for the pressure solved implicitly, but the velocity components are solved explicitly (IMEX) and a semi-implicit additive Runge-Kutta method will help step the solution forward in time [5]. Due to the short scope to plan, code, and present deliverables through the study of the flow field created by the square/rectangular cylinder, the incompressible, unsteady Navier-Stokes equations will be solved through the fractional-step algorithms from both low order fractional step from Saad and from Kim and Moin due to its simplicity and solution speed using both explicit and implicit methods, rather than fully explicitly or implicitly.

3 Methodology

The main goal of this project is to develop a computational fluid dynamics solver for both laminar steady and unsteady flows. The following sections will discuss the methodology and modifications to create this flow solver, and the setup of an experimental test case for both channel flow and square cylinder cases. This derivation is based off of Bisetti (2021), but modified heavily to fit the flow description [10].

3.1 Governing Equations

The incompressible Navier Stokes equations for a Newtonian fluid can be written in vector form is as the following equation. These equations describe the momentum equation law for fluid motion.

$$\frac{\partial \vec{U}}{\partial t} + \vec{U} \cdot \nabla \vec{U} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 \vec{U} + \vec{F}$$

The momentum equation must be supplemented by the conditions of continuity equation, which can be written as the following equation when the flow is incompressible.

$$\rho\nabla \cdot \vec{U} = 0$$

Let u_i, $i = 1, 2, 3$, represent the three components of the velocity vector $\vec{U}$, the continuity equation can be written in the following compact form by using Einstein's summation convention:

$$\frac{\partial(\rho u_i)}{\partial x_i} = 0$$

By using the continuity equation, the Navier-Stokes equation may be written in the following conservative form.

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\mu \frac{\partial u_i}{\partial x_j}\right) + \rho F_i$$

3.2 Finite Volume Method

The finite volume method (FVM) is the mathematical process used to discretize the Navier Stokes partial differential equations by transforming them to a system of algebraic equations. Computational fluid dynamic solvers, based on the finite-volume method, break down the entire flow domain into components in the form of smaller squares or rectangles for discretization called finite volumes. In this method, any term that contains a divergence term is converted to a surface integral through the divergence theorem, which is then evaluated as fluxes at the surfaces of each volume. Typically, a staggered grid, seen in the next section, is used to discretize the Navier Stokes Equations. The divergence theorem is described below:

$$\oint_{\partial V} \vec{F} \cdot \hat{n} \mathrm{dS} = \int_V \vec{\nabla} \cdot \vec{F} \mathrm{d}V$$

3.2.1 Staggered Grid Formation

Within the setup of the staggered grid, vector variables such as velocities are stored within cell faces, while scalar variables such as pressure are stored within the cell centers. In 2D, the variables are staggered in a way where the momentum and the velocities are placed at the negative faces of the control volume, as described in figure 2. In terms of indexing, the velocities on the minus sides will be the same as the cell centered index.

The staggering of velocities u and v imply that different control volumes exist for different quantities. In terms of this project, velocities have their own control volumes. These volumes are centered at the faces of the scalar volumes, as described in figure 2. When expanded into a full grid, a discretized domain will look similar to that described in figure 3.

3.3 FVM on Governing Equations

3.3.1 Conservation of Mass

The mass conservation equation can be discretized to a FVM equation through the divergence theorem. The mass conservation equation for an incompressible fluid is given as the following equation.

$$\frac{\partial u_i}{\partial x_i} = \nabla \cdot u_i = 0$$

From the divergence theorem, a volume integral containing the mass conservation equation can be converted to a surface integral. Here, $\hat{n}$ is defined as the outward unit normal.

$$\int_V \nabla \cdot u_i \mathrm{d}V = \oint_{\partial V} u_i \cdot \hat{n} \mathrm{dS}$$

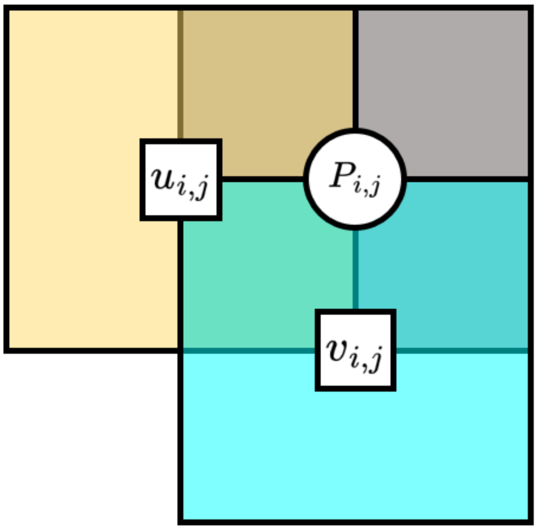

Figure 2: Placement of u and v velocities for the staggered grid formation.

The surface integral can be written as a summation, as described below, where f is the surface count and A_f is the surface area of interest. Here, since the cells are either rectangles or squares, dS when integrated becomes A.

$$\oint_{\partial V} u_i \cdot \hat{n} \mathrm{dS} = \sum_f (u_i \cdot \hat{n}) A_f$$

A finite volume cell at (i, j) is presented in figure 4. To evaluate the surface integral above, the velocities at each face are defined as u_e, u_w, v_n, and v_s, where e, w, n, s are the east, west, north, and south faces respectively. Taking the summation at each of the faces of the cell, the conservation of mass equation becomes the following equation, where h_x and h_y are the unit areas of the cell in the x and y direction respectively.

$$u_e h_y - u_w h_y + v_n h_x - v_s h_x = 0$$

By multiplying the above equation by $h_x h_y$ on the left and right hand sides, the following form of the conservation of mass equation is obtained.

$$\frac{u_e - u_w}{h_x} + \frac{v_n - v_s}{h_y} = 0$$

3.3.2 Conservation of Momentum

The momentum conservation equation can be discretized to a FVM equation through the divergence theorem. The momentum conservation equation for an incompressible fluid is given as the following equation.

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\mu \frac{\partial u_i}{\partial x_j}\right) + \rho F_i$$

The convective and diffusive terms of this equation need to be discretized in order to solve this equation. Through FVM, the convective term is discretized first. The following equation below represents the convective term in the Navier Stokes equation:

$$\frac{\partial(u_i u_j)}{\partial x_j} = \nabla \cdot (u_i u_j)$$

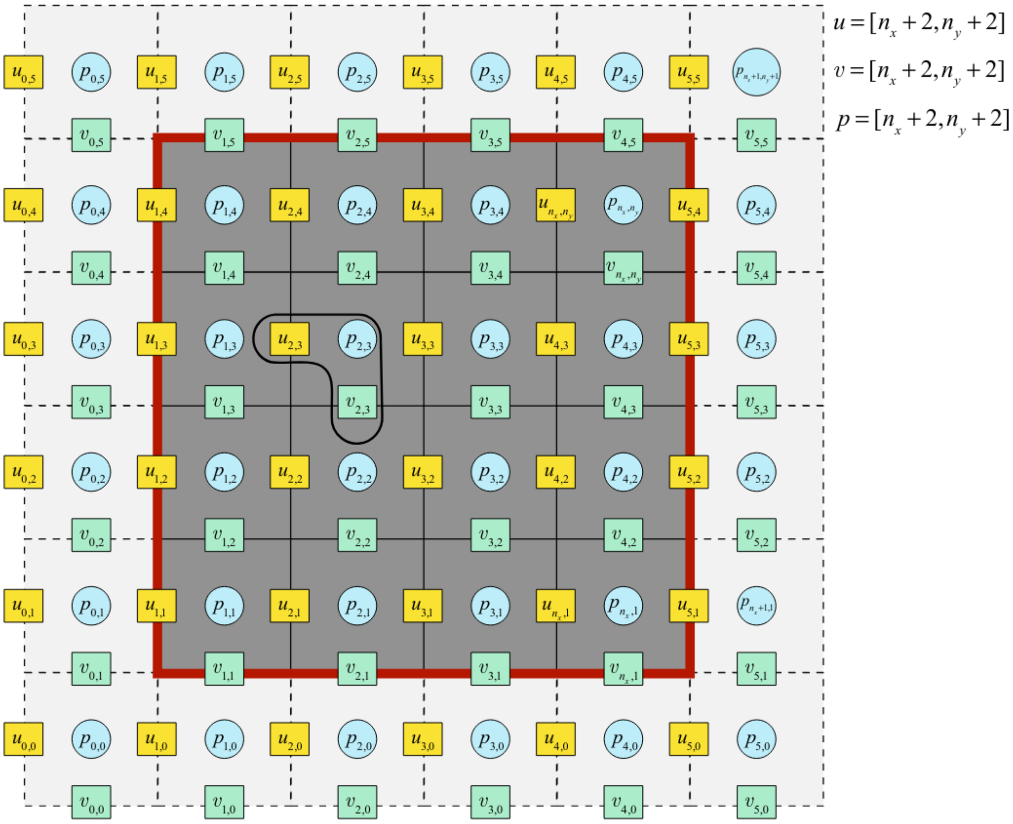

Figure 3: Geometry representing a 4x4 staggered grid domain to solve the Navier Stokes equations. Including ghost cells, this domain becomes to a 6x6 staggered grid (Saad, 2019) [8].

Using the divergence theorem, the convective term can be written as the following equation.

$$\int_V \nabla \cdot (u_i u_j)\mathrm{d}V = \oint_{\partial V} u_i(u_j \cdot \hat{n})\mathrm{dS} = \sum_f u_i(u_j \cdot \hat{n})A_f$$

A finite volume cell at (i, j) on u is presented in figure 4. Similar to how conservation of mass is derived, in this control volume, four velocities going through the north, south, east, and west faces, as described in figure 5. However, since the finite volume is not defined on a cell center, the four velocities at each face have to be derived. For u velocities in the east and west faces, a term $\hat{u}$ is defined, which is the average of two velocities, either left or right to the velocity u in question. In other words, $\hat{u}$ is the velocity of u at a cell center. Similarly, for v velocities in the north and south faces, a term is defined. However, this term is not $\hat{v}$ or known as the v velocity at a cell center. The discretization of the convective momentum flux in Navier Stokes equation can now be written as the following for the x-direction:

$$(\hat{u}_e u_{i,j} - \hat{u}_w u_{i,j})h_y + (\hat{v}_n u_{i,j} - \hat{v}_s u_{i,j})h_x = \oint_{\partial V} u_i(u_j \cdot \hat{n})\mathrm{dS} = \sum_f u_i(u_j \cdot \hat{n})A_f$$

The $\hat{u}$ and $\hat{v}$ terms are defined as the following equations for the x-direction convective momentum equation, where the i index represents a row in the array, and the j index represents the column

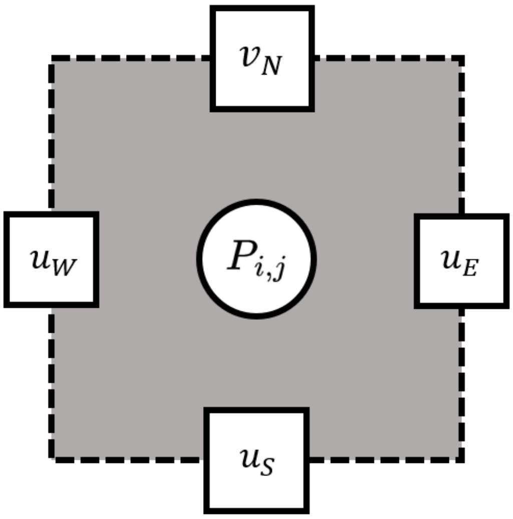

Figure 4: Geometry representing the control volume of a cell at (i, j) within the staggered grid.

in the array:

$$\begin{aligned}
\hat{u}_e = \hat{u}_{i,j} &= \frac{u_{i,j+1} + u_{i,j}}{2} \\
\hat{u}_w = \hat{u}_{i,j-1} &= \frac{u_{i,j} + u_{i,j-1}}{2} \\
\hat{v}_n &= \frac{v_{i+1,j} + v_{i+1,j-1}}{2} \\
\hat{v}_s &= \frac{v_{i,j} + v_{i,j-1}}{2}
\end{aligned}$$

A similar process can be written for the y-direction convective momentum, and a control volume diagram is presented in figure 6. The discretization is as follows:

$$(\hat{u}_e v_{i,j} - \hat{u}_w v_{i,j})h_y + (\hat{v}_n v_{i,j} - \hat{v}_s v_{i,j})h_x = \oint_{\partial V} v_i(u_j \cdot \hat{n})\mathrm{dS} = \sum_f v_i(u_j \cdot \hat{n})A_f$$

The $\hat{u}$ and $\hat{v}$ terms are defined as the following equations, where the i index represents a row in the array, and the j index represents the column in the array:

$$\begin{aligned}
\hat{u}_e &= \frac{u_{i,j+1} + u_{i-1,j+1}}{2} \\
\hat{u}_w &= \frac{u_{i,j} + u_{i-1,j}}{2} \\
\hat{v}_n = \hat{v}_{i,j} &= \frac{v_{i+1,j} + v_{i,j}}{2} \\
\hat{v}_s = \hat{v}_{i-1,j} &= \frac{v_{i,j} + v_{i-1,j}}{2}
\end{aligned}$$

The diffusive term is discretized next using the divergence theorem to obtain the diffusive fluxes for the x and y directions. Here, the diffusive term can be written as the following equation:

$$\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right) = \nabla \cdot \frac{\partial u_i}{\partial x_j}$$

Using the divergence theorem, the diffusive term can be written as the following:

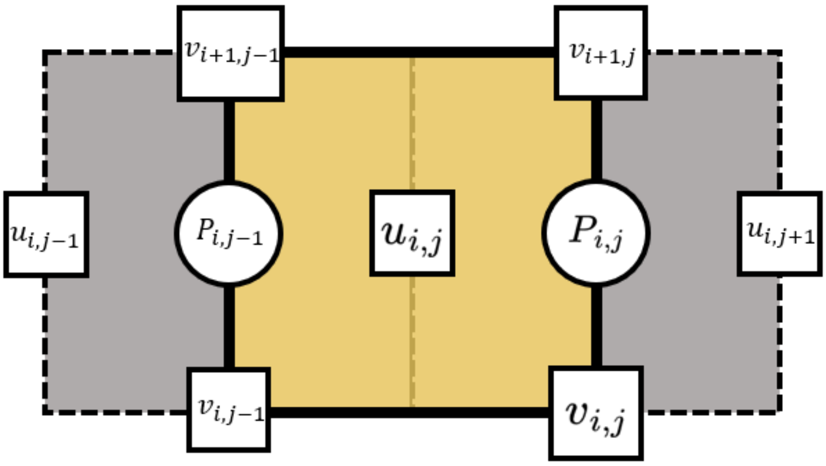

Figure 5: Geometry representing the control volume of a cell at $u_{i,j}$ within the staggered grid.

$$\int_V \nabla \cdot \left(\frac{\partial u_i}{\partial x_j}\right) \mathrm{d}V = \oint_{\partial V} \left(\frac{\partial u_i}{\partial x_j}\right) \cdot \hat{n} \mathrm{dS} = \sum_f \left(\frac{\partial u_i}{\partial x_j} \cdot \hat{n}\right) A_f$$

A finite volume cell at (i, j) on u is presented in figure 7 for the diffusive flux. The east and west fluxes are defined as:

$$\phi_e = \left(\frac{\partial u}{\partial x}\right)_e A_e = \left(\frac{u_{i,j+1} - u_{i,j}}{h_x}\right) h_y \quad \phi_w = \left(\frac{\partial u}{\partial x}\right)_w A_w = \left(\frac{u_{i,j} - u_{i,j-1}}{h_x}\right) h_y$$

The north and south fluxes in the finite volume cell are similarly defined:

$$\phi_n = \left(\frac{\partial u}{\partial y}\right)_n A_n = \left(\frac{u_{i+1,j} - u_{i,j}}{h_y}\right) h_x \quad \phi_s = \left(\frac{\partial u}{\partial y}\right)_s A_s = \left(\frac{u_{i,j} - u_{i-1,j}}{h_y}\right) h_x$$

When the fluxes from all directions are combined as follows, the x-direction diffusion flux is obtained:

$$\oint_{\partial V} \left(\frac{\partial u_i}{\partial x_j}\right) \cdot \hat{n} \mathrm{dS} = \sum_f \left(\frac{\partial u_i}{\partial x_j} \cdot \hat{n}\right) A_f = \phi_e - \phi_w + \phi_n - \phi_s$$

$$\oint_{\partial V} \left(\frac{\partial u_i}{\partial x_j}\right) \cdot \hat{n} \mathrm{dS} = \left(\frac{u_{i,j+1} - 2u_{i,j} - u_{i,j-1}}{h_x}\right) h_y + \left(\frac{u_{i+1,j} - 2u_{i,j} - u_{i-1,j}}{h_y}\right) h_x$$

Similarly, the y-direction diffusion flux for the Navier Stokes equation can also be obtained by following the same process. A finite volume cell at (i, j) on v is presented in figure 8 for the diffusive flux.

$$\oint_{\partial V} \left(\frac{\partial v_i}{\partial x_j}\right) \cdot \hat{n} \mathrm{dS} = \left(\frac{v_{i,j+1} - 2v_{i,j} - v_{i,j-1}}{h_x}\right) h_y + \left(\frac{v_{i+1,j} - 2v_{i,j} - v_{i-1,j}}{h_y}\right) h_x$$

To simplify the Navier Stokes equations for later, the convective flux discretization is set to the convective operator $A_{i,j}$ and the diffusive flux discretization set to the diffusive operator $D_{i,j}$ for

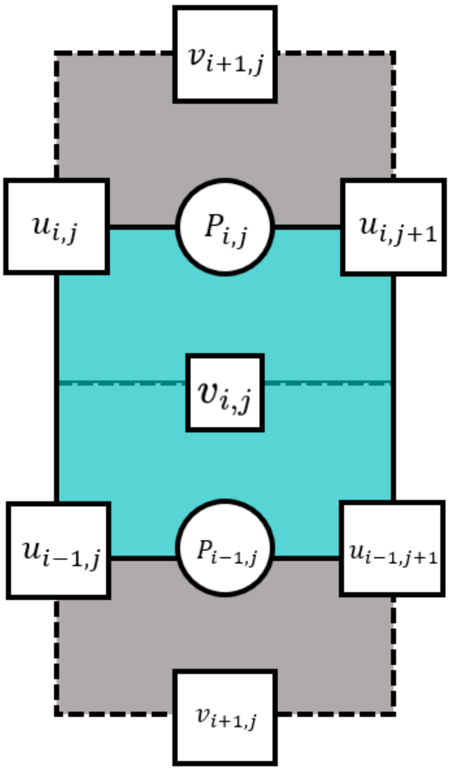

Figure 6: Geometry representing the control volume of a cell at $v_{i,j}$ within the staggered grid.

both u and v velocities.

$$\begin{aligned} A^u_{i,j} &= (\hat{u}_e u_{i,j} - \hat{u}_w u_{i,j})h_y + (\hat{v}_n u_{i,j} - \hat{v}_s u_{i,j})h_x \\ where: \hat{u}_e = \hat{u}_{i,j} &= \frac{u_{i,j+1} + u_{i,j}}{2} \\ \hat{u}_w = \hat{u}_{i,j-1} &= \frac{u_{i,j} + u_{i,j-1}}{2} \\ \hat{v}_n &= \frac{v_{i+1,j} + v_{i+1,j-1}}{2} \\ \hat{v}_s &= \frac{v_{i,j} + v_{i,j-1}}{2} \end{aligned}$$

$$\begin{aligned} A^v_{i,j} &= (\hat{u}_e v_{i,j} - \hat{u}_w v_{i,j})h_y + (\hat{v}_n v_{i,j} - \hat{v}_s v_{i,j})h_x \\ where: \hat{u}_e &= \frac{u_{i,j+1} + u_{i-1,j+1}}{2} \\ \hat{u}_w &= \frac{u_{i,j} + u_{i-1,j}}{2} \\ \hat{v}_n = \hat{v}_{i,j} &= \frac{v_{i+1,j} + v_{i,j}}{2} \\ \hat{v}_s = \hat{v}_{i-1,j} &= \frac{v_{i,j} + v_{i-1,j}}{2} \end{aligned}$$

$$\begin{aligned} D^u_{i,j} &= \left(\frac{u_{i,j+1} - 2u_{i,j} - u_{i,j-1}}{h_x}\right) h_y + \left(\frac{u_{i+1,j} - 2u_{i,j} - u_{i-1,j}}{h_y}\right) h_x \\ D^v_{i,j} &= \left(\frac{v_{i,j+1} - 2v_{i,j} - v_{i,j-1}}{h_x}\right) h_y + \left(\frac{v_{i+1,j} - 2v_{i,j} - v_{i-1,j}}{h_y}\right) h_x \end{aligned}$$

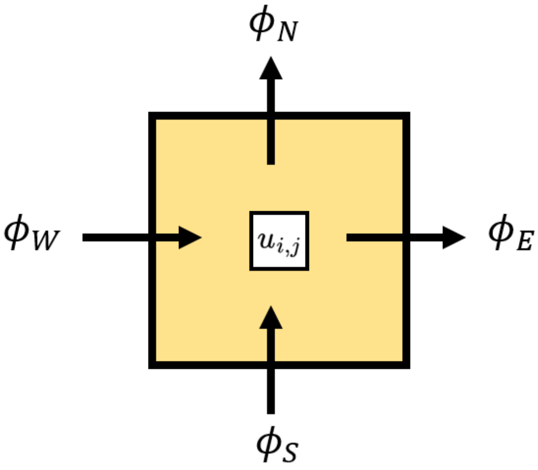

Figure 7: Geometry representing the control volume of a cell at $u_{i,j}$ within the staggered grid.

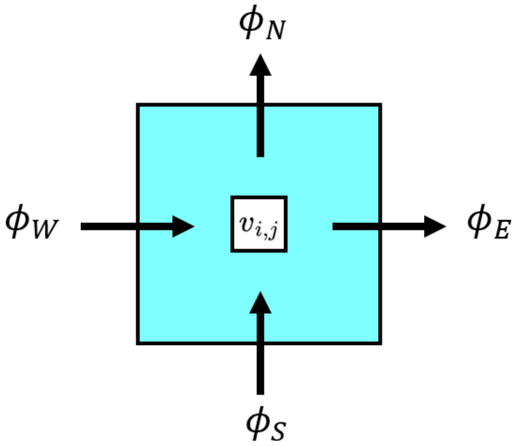

Figure 8: Geometry representing the control volume of a cell at $v_{i,j}$ within the staggered grid.

3.4 Navier Stokes Fractional Step Method

3.4.1 Derivation of Equations

The Navier Stokes momentum equation can be solved numerically through the fractional step method. This is a lower order method to solving the fractional step method from Saad [8], as section 3.5 will present a higher order fractional step method from Kim and Moin [9]. For simplicity, the Navier Stokes equations is presented in non-dimensionalized form as the following:

$$\frac{\partial \vec{U}}{\partial t} = -\vec{U} \cdot \nabla \vec{U} - \nabla P + \frac{1}{Re} \nabla^2 \vec{U}$$

The first term containing the time derivative is discretized into a first order finite difference expression, shown below:

$$\frac{\vec{U}^{n+1} - \vec{U}^n}{k} = -\vec{U} \cdot \nabla \vec{U} - \nabla P + \frac{1}{Re} \nabla^2 \vec{U}$$

Reordering the equation above so $\vec{U}^{n+1}$ is on the left hand side, and set all the variables to be solved at time-step n except for the pressure term P:

$$\vec{U}^{n+1} = \vec{U}^n + k(-\vec{U}^n \cdot \nabla \vec{U}^n - \nabla P^{n+1} + \frac{1}{Re} \nabla^2 \vec{U}^n)$$

If the divergence is set on both left and right hand sides, the following equation is obtained:

$$\nabla \cdot \vec{U}^{n+1} = \nabla \cdot (\vec{U}^n + (k(-(\vec{U}^n \cdot \nabla \vec{U}^n) - \nabla P^{n+1} + \frac{1}{Re}\nabla^2 \vec{U}^n))$$
$$\nabla \cdot \vec{U}^{n+1} = \nabla \cdot (\vec{U}^n + k(-(\vec{U}^n \cdot \nabla \vec{U}^n) + \frac{1}{Re}\nabla^2 \vec{U}^n)) - k\nabla^2 P^{n+1}$$

It can be also noted that $\nabla \cdot \vec{U}^{n+1} = 0$ to preserve the conservation of mass equations. Furthermore, to simplify the equation above, a variable $\vec{U}^*$ is used as a substitute for the first term on the right hand side.

$$\vec{U}^* = \vec{U}^n + k(-(\vec{U}^n \cdot \nabla \vec{U}^n) + \frac{1}{Re}\nabla^2 \vec{U}^n)$$

$$0 = \nabla \cdot \vec{U}^* - k\nabla^2 P^{n+1}$$

This returns a Poisson equation for P at time step $n+1$:

$$\nabla^2 P^{n+1} = \frac{\nabla \cdot \vec{U}^*}{k}$$

Furthermore, the Poisson equation can be discretized to a second order central finite difference expression:

$$\nabla^2 P_{i,j}^{n+1} = \frac{P_{i,j+1}^{n+1} - 2P_{i,j}^{n+1} + P_{i,j-1}^{n+1}}{h_x^2} + \frac{P_{i+1,j}^{n+1} - 2P_{i,j}^{n+1} + P_{i-1,j}^{n+1}}{h_y^2}$$

The term on the right hand side of the Poisson equation can be discretized to the following expression:

$$\frac{\nabla \cdot \vec{U}^*}{k} = \frac{1}{k}\left(\frac{u_{i,j+1}^* - u_{i,j}^*}{h_x} + \frac{v_{i+1,j}^* - v_{i,j}^*}{h_y}\right)$$

Therefore, the Poisson Equation can be written as:

$$\frac{P_{i,j+1}^{n+1} - 2P_{i,j}^{n+1} + P_{i,j-1}^{n+1}}{h_x^2} + \frac{P_{i+1,j}^{n+1} - 2P_{i,j}^{n+1} + P_{i-1,j}^{n+1}}{h_y^2} = \frac{1}{k}\left(\frac{u_{i,j+1}^* - u_{i,j}^*}{h_x} + \frac{v_{i+1,j}^* - v_{i,j}^*}{h_y}\right)$$

Returning back to the equation for $\vec{U}^{n+1}$, U^* is substituted and the following equation is obtained:

$$\vec{U}_{i,j}^{n+1} = \vec{U}_{i,j}^* - k\nabla P_{i,j}^{n+1}$$

In terms of writing $\vec{U}_{i,j}^{n+1}$ into its components $u_{i,j}^{n+1}$ and $v_{i,j}^{n+1}$, the velocity components at the next time-step can be found through the following discretization expression:

$$u_{i,j}^{n+1} = u_{i,j}^* - k\left(\frac{P_{i,j}^{n+1} - P_{i-1,j}^{n+1}}{h_x}\right)$$

$$v_{i,j}^{n+1} = v_{i,j}^* - k\left(\frac{P_{i,j}^{n+1} - P_{i,j-1}^{n+1}}{h_y}\right)$$

3.4.2 General Algorithm

With the equations found in section 3.4.1, a general algorithm can be established to solving the Navier Stokes equations for the next time-step at cell (i,j). The terms that represent the convective and diffusive terms are replaced with $A_{i,j}$ and $D_{i,j}$, respectively.

$$A_{i,j}^n = (\vec{U}^n \cdot \nabla \vec{U}^n) \qquad D_{i,j}^n = \nabla^2 \vec{U}^n$$

The solution algorithm is as follows:

1. Solve for a temporary velocity U^* at location (i,j): $\vec{U}^*_{i,j} = \vec{U}^n_{i,j} + k(-A^n_{i,j} + \frac{1}{Re} D^n_{i,j})$

2. With known $\vec{U}^*_{i,j}$, solve the system of equations for P^{n+1}: $\nabla^2 P^{n+1}_{i,j} = \dfrac{\nabla \cdot \vec{U}^*_{i,j}}{k}$

3. Using $P^{n+1}_{i,j}$ and $\vec{U}^*_{i,j}$, solve for the velocities at the next time step: $\vec{U}^{n+1}_{i,j} = \vec{U}^*_{i,j} - k\nabla P^{n+1}_{i,j}$

4. Repeat step 1 using $\vec{U}^{n+1}_{i,j}$ as $\vec{U}^n_{i,j}$ until desired time step is reached.

3.5 Kim and Moin Fractional Step Method

The following section outlines a fractional step method from Kim-Moin which is based off the textbook for ASE382-Q11 from Bisetti (2021) [10], which is a higher order method than the method outline in section 3.4. The methods shown here are modified for different grid sizes in the x and y directions.

3.5.1 Convective and Diffusive Expressions

The following equations represent the convective operator $N_{i,j}$ for Kim and Moin, recalling that the convective term can be written as the surface integral and sum in section 3.3.2.

$$N^u_{i,j} = -u_{i,j}\left(\frac{u_{i,j+1} - u_{i,j-1}}{2h_x}\right) - \left(\frac{\hat{v}_{i,j} - \hat{v}_{i,j-1}}{2}\right)\left(\frac{u_{i+1,j} - u_{i-1,j}}{2h_y}\right)$$

$$N^v_{i,j} = -v_{i,j}\left(\frac{v_{i+1,j} - v_{i-1,j}}{2h_y}\right) - \left(\frac{\hat{u}_{i,j} - \hat{u}_{i-1,j}}{2}\right)\left(\frac{v_{i,j+1} - u_{i,j-1}}{2h_x}\right)$$

The $\hat{u}$ and $\hat{v}$ terms are defined as follows:

$$\hat{u}_{i,j} = \frac{u_{i,j} + u_{i,j+1}}{2} \qquad \hat{v}_{i,j} = \frac{v_{i,j} + v_{i+1,j}}{2}$$

The following equations represent the diffusive operator for Kim and Moin:

$$\begin{aligned} L^u_{i,j} &= \left(\frac{u_{i,j+1} - 2u_{i,j} - u_{i,j-1}}{h_x^2}\right) + \left(\frac{u_{i+1,j} - 2u_{i,j} - u_{i-1,j}}{h_y^2}\right) \\ L^v_{i,j} &= \left(\frac{v_{i,j+1} - 2v_{i,j} - v_{i,j-1}}{h_x^2}\right) + \left(\frac{v_{i+1,j} - 2v_{i,j} - v_{i-1,j}}{h_y^2}\right) \end{aligned}$$

The difference between the Kim and Moin fractional step method and the method presented in section 3.4 is how the temporary velocities u^* and v^* are obtained. The expressions for u^* and v^* are as follows, obtained at (i,j):

$$u^* - \frac{k}{2}\frac{1}{Re}L^u(u^*) = \frac{k}{2}\left(3N^u(u^n, v^n) - N^u(u^{n-1}, v^{n-1}\right) + u^n + \frac{k}{2}\frac{1}{Re}L^u(u^n)$$

$$v^* - \frac{k}{2}\frac{1}{Re}L^v(v^*) = \frac{k}{2}\left(3N^v(u^n, v^n) - N^v(u^{n-1}, v^{n-1}\right) + v^n + \frac{k}{2}\frac{1}{Re}L^v(v^n)$$

3.5.2 General Algorithm

Note that through the Kim and Moin fractional step method, the pressure relation seen in 3.4.1 is not equal to pressure. Rather, a pseudo "pressure" ϕ replaces P. Thus, the general algorithm using the Kim and Moin method is as follows:

1. Repeat the algorithm listed in section 3.4.2 for a finite and small amount of timesteps.

2. Save the velocities $\hat{u}$ and $\hat{v}$ after each timestep.

3. Begin Kim and Moin Fractional Step Method after reaching the timestep criteria.

4. Solve u^* and v^* through a system of equation solver using expressions shown in section 3.5.1.

5. With known $\vec{U}^*_{i,j}$, solve the system of equations for ϕ^{n+1}: $\nabla^2\phi^{n+1}_{i,j} = \dfrac{\nabla \cdot \vec{U}^*_{i,j}}{k}$

6. Using $\phi^{n+1}_{i,j}$ and $\vec{U}^*_{i,j}$, solve for the velocities at the next time step: $\vec{U}^{n+1}_{i,j} = \vec{U}^*_{i,j} - k\nabla\phi^{n+1}_{i,j}$

7. Repeat step 3 using $\vec{U}^{n+1}_{i,j}$ as $\vec{U}^n_{i,j}$ until desired time step is reached.

For purposes of this project, the number of time steps set for the switch from a lower order fractional step method to a higher order fractional step method is 3.

3.6 Boundary Conditions

Because of the staggered grid setup to solving the Navier Stokes equations, some boundary conditions have to be discretized in order to obtain a solution. In general, the properties of both real ($\vec{U}$) and temporary velocities ($\vec{U}^*$) have to be defined at each wall, and the pressure (P) and pseudo-pressures (ϕ) must also be defined at each wall. This section will be divided into finding expressions for the walls for channel flow and the expressions for the walls for when a square cylinder is added within the flow. Two main boundary conditions exist in the flow field studied. These are Dirichlet and Neumann boundary conditions. On a no-slip boundary, the velocities are subjected to Dirichlet boundary conditions. The velocities u, v, u^*, and v^* are as follows:

$$u = 0 \quad v = 0 \quad u^* = 0 \quad v^* = 0$$

At the inlet, the u and v velocities are subjected to Dirichlet boundary conditions. Here, the values of u and v are as follows:

$$u = u_{inlet} \quad v = 0$$

At the outlet, the velocities u and v are subjected to Neumann boundary conditions and this is written as:

$$\frac{\partial u}{\partial x} = 0 \quad \frac{\partial v}{\partial x} = 0$$

In terms of pressure boundary conditions, the no-slip walls and the inlet will have a pressure gradient of $\hat{n} \cdot \nabla P = 0$, where $\hat{n}$ is the outward normal for each face, a Neumann boundary condition. At the outlet, the pressure is set to $P = 0$, a Dirichlet boundary condition.

3.6.1 Boundary Conditions Expressions for the Channel

The u^* and v^* boundary conditions are applied first. Because of the staggered grid formation, u and u^* can be directly applied to the inlet and the outlet, but not the lower and upper walls. On the other hand, v and v^* can be applied directly to the upper and lower walls, but not the inlet and outlet. At the inlet, $u = u_{inlet}$, so all u cells the represent the inlet can be defined as the following, where i is the row index, and j is the column index:

$$u(i,1) = u_{inlet} \quad u^*(i,1) = u_{inlet}$$

At the outlet, the condition $\dfrac{\partial u}{\partial x} = 0$ is required. To discretize this first derivative, a backward 2nd order finite difference expression is needed. Below is the backwards first derivative discretization of $\dfrac{\partial u}{\partial x} = 0$ at (i, j) accurate to the second order.

$$\frac{\partial u}{\partial x} = \frac{3u_{i,j} - 4u_{i,j-1} + u_{i,j-2}}{2h_x} = 0$$

Knowing that the updated velocity u^{n+1} is a function of the temporary velocity u^*, as shown below:

$$u^{n+1}_{i,j} = u^*_{i,j} - kG^x_{i,j}(\phi^n)$$
$$where \quad G^x_{i,j}(\phi^n) = \frac{\phi_{i,j} - \phi_{i,j-1}}{h_x}$$

The temporary velocity at the outlet can then be written as the following expression, where i is the row index and N_x+1 represents the column that is one bigger than the specified number of cells in x. Note that the staggered grid formation includes ghost cells, so i will range from cell index 0 to cell index N_y+1, and j will range from cell index 0 to cell index N_x+1:

$$u^*(i,N_x+1) = kG_{i,N_x+1}(\phi^n) + \frac{4}{3}(u^*(i,N_x) - kG_{i,N_x}(\phi^n)) - \frac{1}{3}(u^*(i,N_x-1) - kG_{i,N_x-1}(\phi^n))$$

The top and bottom boundaries are also Dirichlet boundary conditions, so in terms of the u velocity, $u=0$ at these walls. However, it is futile to set this boundary condition, as the staggered grid formation prevents this from happening, as described in figure 9, where each wall is split between two velocities, one inner and in the flow and one ghost. As a result, expressions need to be developed such that the average of the inner and ghost cells are equal to 0, a non-moving, no-slip boundary, represented as the following equation:

$$\frac{u_{N_y+1,j}+u_{N_y,j}}{2} = 0 \quad \textit{top wall}$$
$$\frac{u_{N_0,j}+u_{1,j}}{2} = 0 \quad \textit{bottom wall}$$

Using the G operator to convert from u^* to u, the equations for u^* for the top and bottom boundary ghost cells are found as follows:

$$u^*(N_y+1,j) = -u^*(N_y,j) + k(G^x_{N_y+1,j}(\phi^n) + G^x_{N_y,j}(\phi^n)) \quad \textit{top wall}$$
$$u^*(0,j) = -u^*(1,j) + k(G^x_{0,j}(\phi^n) + G^x_{1,j}(\phi^n)) \quad \textit{bottom wall}$$

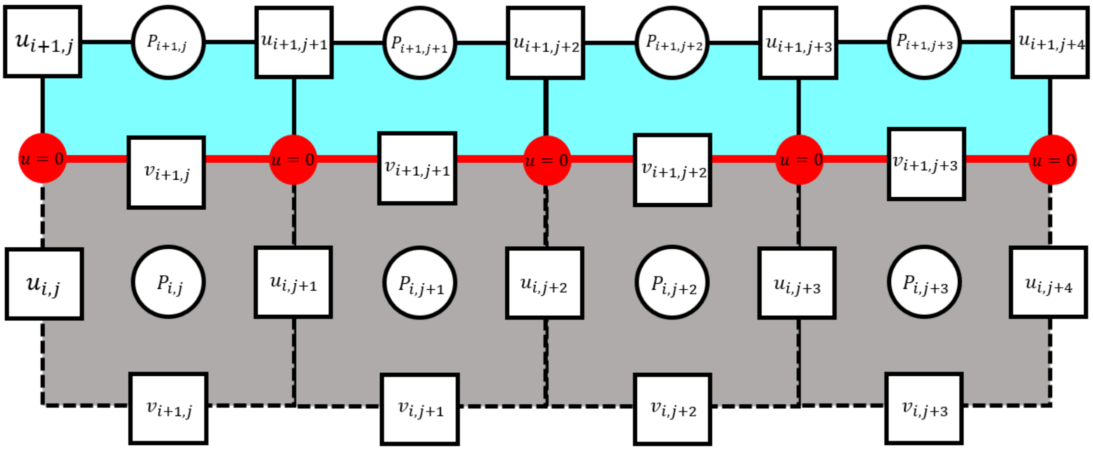

Figure 9: Geometry representing the top boundary within the staggered grid arrangement for channel flow.

Similarly, the top and bottom boundaries are directly declared with the v and v^* velocities, but the inlet and outlet boundaries are not. Below are already worked out correlations for the value of v^* at the top and bottom boundaries, and v^* for the ghost inlet and outlet boundaries. v^* for the top boundary and bottom boundaries are expressed as:

$$v^*(N_y+1,j) = v^*(0,j) = 0$$

$v*$ for the ghost inlet cells is expressed as the following equation. Note that the inlet condition is a Dirichlet boundary condition $v=0$:

$$v^*(i,0) = 2v_{inlet} - v^*(i,1) + k(G^y_{i,0}(\phi^n) + G^y_{i,1}(\phi^n))$$

The ghost outlet cells for v^* can be expressed as the following equation. Note that the outlet condition is a Neumann boundary condition $\dfrac{\partial v}{\partial n} = 0$:

$$v^*(i, N_x + 1) = v^*(i, N_x) + k(G^y_{i,N_x}(\phi^n) + G^y_{i,N_x+1}(\phi^n))$$

Note that the term G^y is the operator described below:

$$G^y_{i,j}(\phi^n) = \frac{\phi_{i,j} - \phi_{i-1,j}}{h_y}$$

Pressure boundary conditions are also applied to the flow. In general, the outlet is set to a pressure of $P = 0$ and $\nabla P = 0$ for the upper wall, lower wall, and the inlet. The form of the equation for the pressure boundaries, specifically the ghost cells, are set such that it can be solved through an extremely large matrix $Ax = b$, where A is the coefficient matrix, x is the solution for the value of P, and b is the solution if A and x are multiplied. For cells at the inlet, ∇P is applied. When discretized, the following equation is obtained:

$$-P(i, 0) + P(i, 1) = 0$$

At the outlet, $P = 0$ is applied. When discretized, the following equation for the pressure is obtained:

$$P(i, N_x) + P(i, N_x + 1) = 0$$

At the top wall, $\nabla P = 0$ is applied and the following equation for the pressure is obtained:

$$P(N_y, j) - P(N_y + 1, j) = 0$$

At the bottom wall, $\nabla P = 0$ is applied and the following equation for the pressure is obtained:

$$-P(0, j) + P(1, j) = 0$$

3.6.2 Boundary Conditions Expressions for the Cylinder

Defining boundaries for the cylinder is similar to finding boundary conditions for channel flow. At each wall of the cylinder, $u = 0$, $u^* = 0$, $v = 0$, and $v^* = 0$. The pressure conditions for the cylinder are $\hat{n} \cdot \nabla P = 0$. Within the context of this project, the cylinder has been defined at the following cell points:

$$N_{y_{lower}} \le i \le N_{y_{upper}}$$
$$N_{x_{lower}} \le j \le N_{x_{upper}}$$

$N_{y_{lower}}$ and $N_{y_{upper}}$ are the lowermost and uppermost cell bounds for the cylinder, and $N_{x_{lower}}$ and $N_{x_{upper}}$ are the leftmost and rightmost cell bounds for the cylinder. Figure 1 describes a square cylinder placed relative to the channel flow. Below are the boundary conditions for u^* and v^* at each cylinder wall.

$$u^*(N_{y_{lower}} : N_{y_{upper}}, N_{x_{lower}}) = 0$$
$$u^*(N_{y_{lower}} : N_{y_{upper}}, N_{x_{upper}}) = 0$$
$$v^*(N_{y_{lower}}, N_{x_{lower}} : N_{x_{upper}}) = 0$$
$$v^*(N_{y_{upper}}, N_{x_{lower}} : N_{x_{upper}}) = 0$$

Because of the staggered grid formation, the u velocities can be directly declared on the left and right cylinder walls while the v velocities can be directly declared on the top and bottom walls. Similar to the process found in section 3.6.1, ghost cells for $u*$ and $v*$ can be derived. In the case of the left and right cylinder walls, the u^* and $v*$ velocities can be found as:

$$u^*(N_{y_{lower}}, j) = -u^*(N_{y_{lower}} - 1, j) \ where \ N_{x_{lower}} + 1 \le j \le N_{x_{upper}} - 1$$
$$u^*(N_{y_{upper}}, j) = -u^*(N_{y_{upper}} + 1, j) \ where \ N_{x_{lower}} + 1 \le j \le N_{x_{upper}} - 1$$
$$v^*(i, N_{x_{lower}}) = -v^*(i, N_{x_{lower}} - 1) \ where \ N_{y_{lower}} + 1 \le i \le N_{y_{upper}} - 1$$
$$v^*(i, N_{x_{upper}}) = -v^*(i, N_{x_{upper}} + 1) \ where \ N_{y_{lower}} + 1 \le i \le N_{y_{upper}} - 1$$

Like the lower and upper walls of the channel, the pressure boundary condition along the walls is set to $\hat{n} \cdot \nabla P = 0$. The following equations represent the equations for the pressure condition on each wall. These equations are implemented in the system of equations to solve for the pressure. For the lower and upper walls, the pressure equations for the square cylinder can be written as:

$$P(N_{y_{lower}} - 1, j) - P(N_{y_{lower}}, j) = 0 \;\; where \; N_{x_{lower}} \leq j \leq N_{x_{upper}}$$
$$-P(N_{y_{upper}}, j) + P(N_{y_{upper}} + 1, j) = 0 \;\; where \; N_{x_{lower}} \leq j \leq N_{x_{upper}}$$

The left and right wall pressure equations for the square cylinder can be written as:

$$P(i, N_{x_{lower}} - 1) - P(i, N_{x_{lower}}) = 0 \;\; where \; N_{y_{lower}} + 1 \leq i \leq N_{y_{upper}} - 1$$
$$-P(i, N_{x_{upper}}) + P(i, N_{x_{upper}}) + 1 = 0 \;\; where \; N_{y_{lower}} + 1 \leq i \leq N_{y_{upper}} - 1$$

4 Results

4.1 Problem Setup

For verification of the solver, channel flow has to be verified first. For this setup, the length of the plane channel is set to 0.2 meters and the distance between each plane is set to 0.1 meters. An arbitrary inlet velocity is introduced $U_{inlet} = 10$. The verification of the u velocity profile at steady state is given by the following analytical solution, from Sumesaraayi and Ihmsen (2021) [11], where y is the distance away from the bottom horizontal plate and h is the separation between the two plates.

$$u(y) = 1.5U_{inlet}\left(1 - \left(\frac{y - h/2}{h/2}\right)^2\right)$$

The grid size for this domain is 50 by length and 25 by height. The timestepping is 10^{-5} seconds. The flow domain is discretized through FORTRAN 90 and the flow domain is obtained through Paraview. The grid for this setup is described below in figure 10. For purposes of this project, the placement of the cylinder follows that of Breuer et al. (1999) [1], where a fully-developed fluid flows through a plane channel with a square cylinder mounted halfway between the top and bottom planes. The length of the plane channel is set to 30 meters and the distance between each plane is set to 8 meters. The leading edge of the square cylinder is placed exactly one-third the length of the plane channel from the inlet to the center of the cylinder. In the following sections, figure 23 depicts a detailed description of the setup of the square cylinder in channel flow.

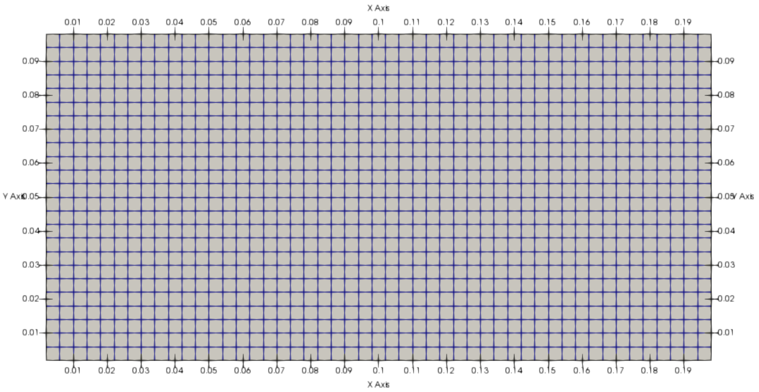

Figure 10: 50x25 grid representing discretization of the channel flow domain.

4.2 Channel Flow Results

4.2.1 Low Order Fractional Step

The flow domain at startup $t = 0$ for the u and v velocities is presented in figures 11 and 12 below, respectively.

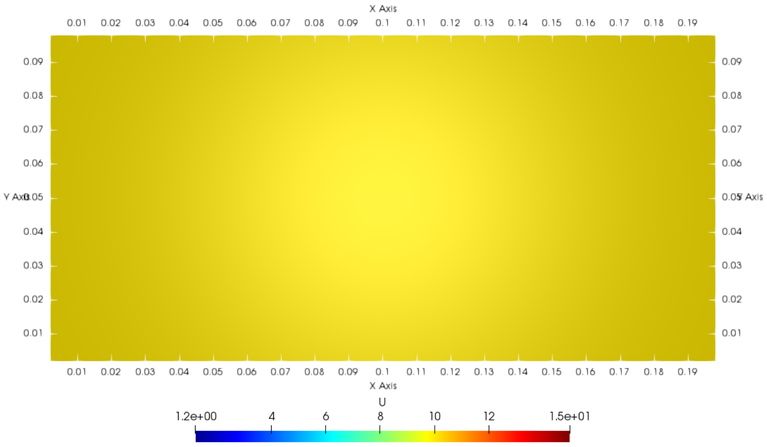

Figure 11: Channel flow with $u_{inlet} = 10\,m/s$ at $t = 0\,s$ using low order fractional step method for flow in the u direction.

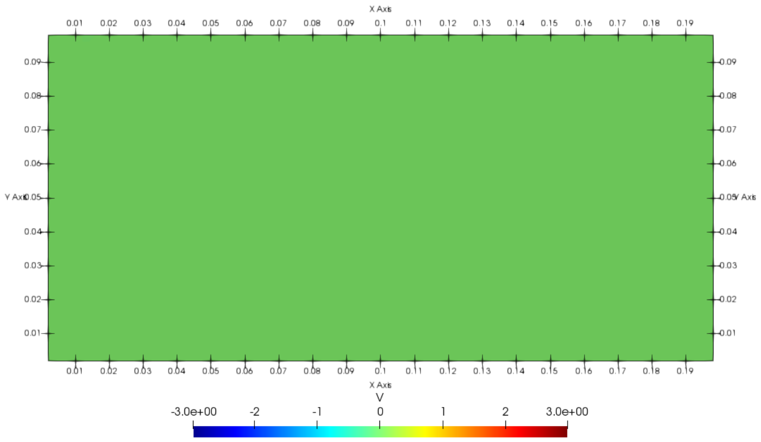

Figure 12: Channel flow with $u_{inlet} = 10\,m/s$ at $t = 0\,s$ using low order fractional step method for flow in the v direction.

The flow domain at startup $t = 10^{-3}$ for the u and v velocities is presented in figures 13 and 14 below, respectively.

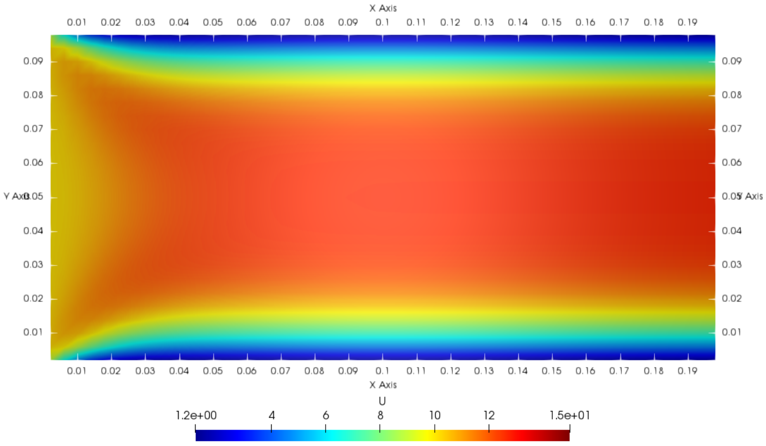

Figure 13: Channel flow with $u_{inlet} = 10\,m/s$ at $t = 1 \times 10^{-3}\,s$ using low order fractional step method for flow in the u direction.

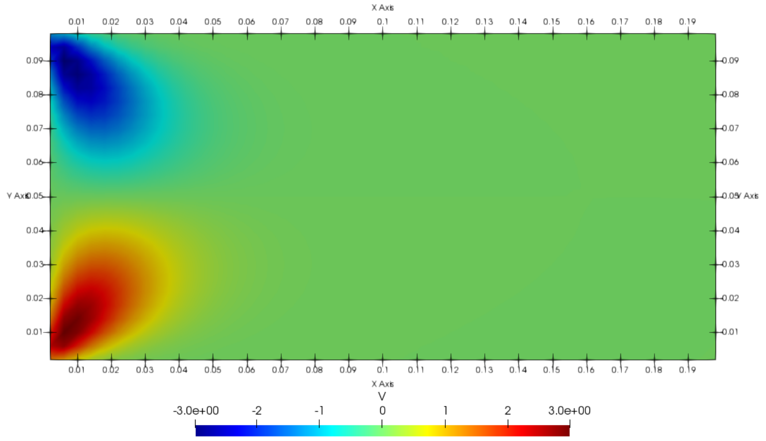

Figure 14: Channel flow with $u_{inlet} = 10\,m/s$ at $t = 1 \times 10^{-3}\,s$ using low order fractional step method for flow in the v direction..

The flow domain at startup $t = 10^{-3}$ for the u and v velocities is presented in figures 15 and 16 below, respectively.

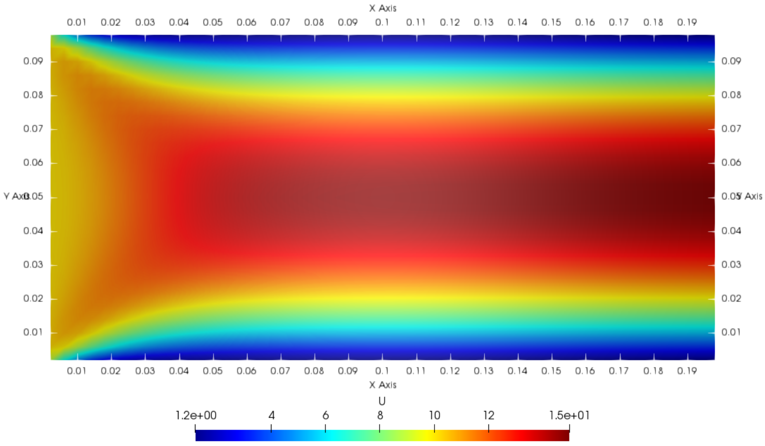

Figure 15: Channel flow with $u_{inlet} = 10\,m/s$ at $t = 1 \times 10^{-2}\,s$ using low order fractional step method for flow in the u direction.

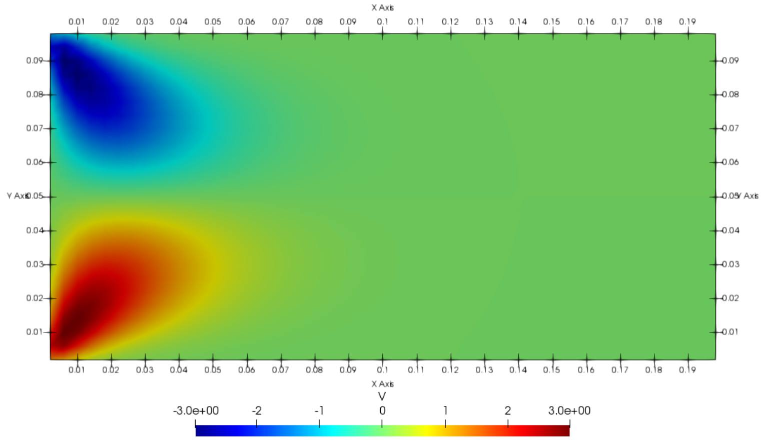

Figure 16: Channel flow with $u_{inlet} = 10\,m/s$ at $t = 10^{-2}\,s$ using low order fractional step method for flow in the v direction..

A plot comparing the analytical steady state u velocity profile at steady state and the simulated u velocity profile at $t = 10^{-2}\,s$ is presented in figure 17 below. A probe located at $x = 0.1\,m$ in the flow records the u velocity along the height of the domain y. Using the lower order fractional step method from Saad, channel flow results can be accurately produced and it can be said that the flow is steady state by $t = 10^{-2}\,s$.

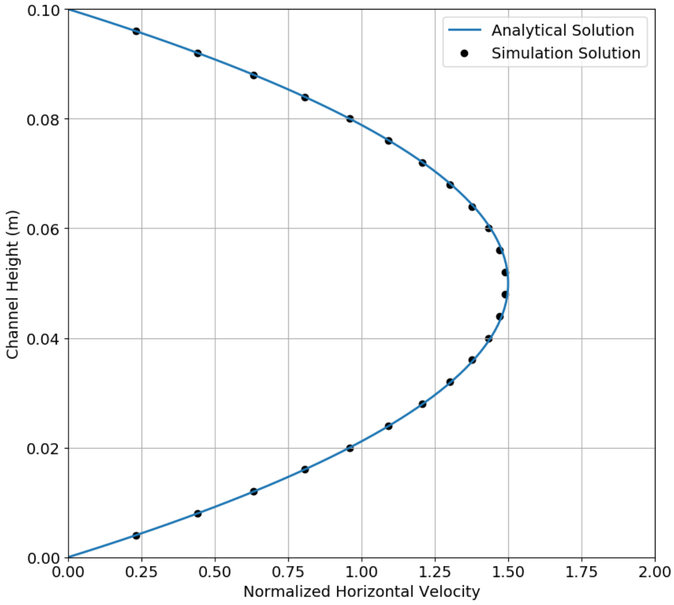

Figure 17: Plot comparing the analytical solution for steady state 2D Poisuelle flow and the simulation at $t = 10^{-2}$ seconds.

4.2.2 Kim and Moin Fractional Step

The results when the flow is solved through the Kim and Moin method is presented in this section. It is expected that the Kim and Moin method will produce results similar to the lower order method presented in the previous section. However, the computation for Kim and Moin is more intensive, as it requires a solution for a set of equations for u^*, v^*, and ϕ^* than a solution for a set of equations for just P. Thus, it is recommended that channel flow analysis with the square cylinder be done by using the lower order method if time is a limiting factor.

The following are simulation results for channel flow using Kim and Moin using the same flow conditions used in the previous section. However, due to an unforeseen bug in the code, the grid size is modified such that N_x and N_y are both even and equal. The grid domain is presented in figure 18 below, using a 50 x 50 grid, and results are presented in figures 19 and 20 for the x and y direction velocities respectively. To compare the Kim-Moin Fractional step method with the lower-order fractional step method, a probe measuring the u and v velocities is set at $x = 0.1\,m$ for u and $x = 0.025\,m$ for v for both discretization schemes and compared. The time is set at $t = 1.25 \times 10^{-3}\,s$. Viewing figures 21 and 22, the Kim and Moin solution to the horizontal and vertical velocities at a location x along the direction y is unsymmetrical, whereas the lower order solution is nearly symmetric halfway between the two horizontal stationary plates $y = 0.05$. Thus, current and future work shall be done using the lower order fractional step method from Saad et al (2019) [8].

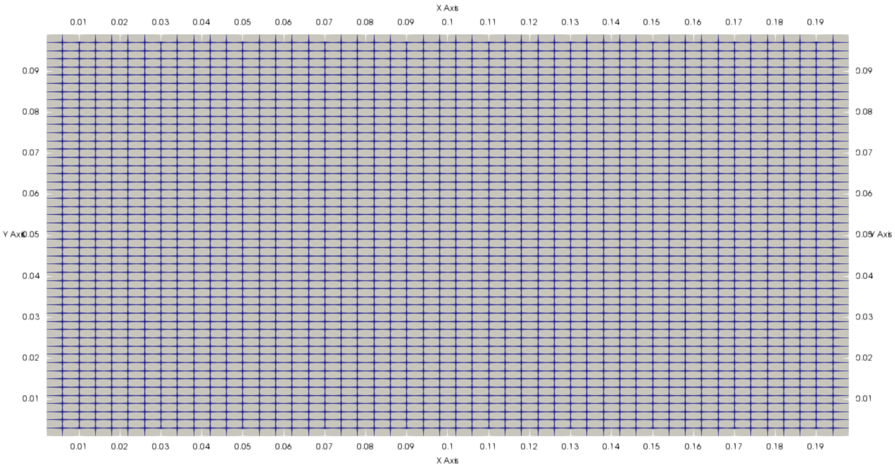

Figure 18: 50x50 grid representing discretization of the channel flow domain for Kim and Moin.

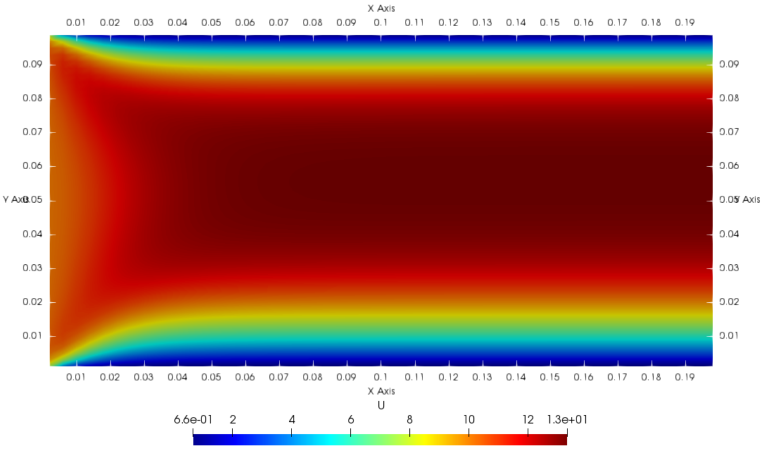

Figure 19: Channel flow with $u_{inlet} = 10\,m/s$ at $t = 1 \times 10^{-2}\,s$ using the Kim-Moin method for flow in the u direction.

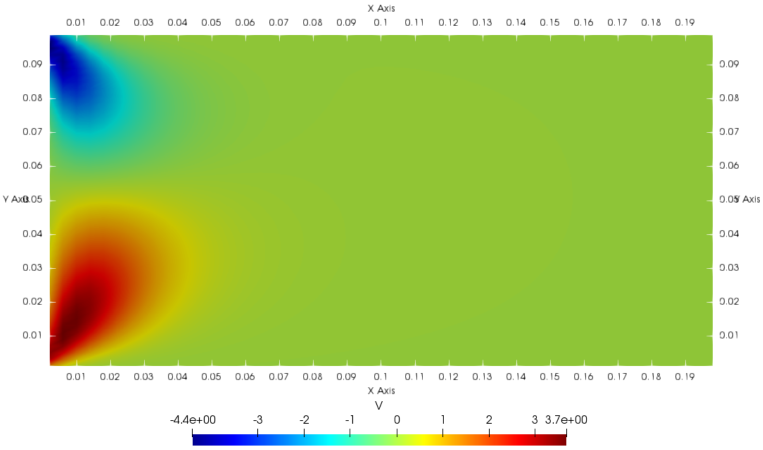

Figure 20: Channel flow with $u_{inlet} = 10\,m/s$ at $t = 1 \times 10^{-2}\,s$ using the Kim-Moin method for flow in the v direction.

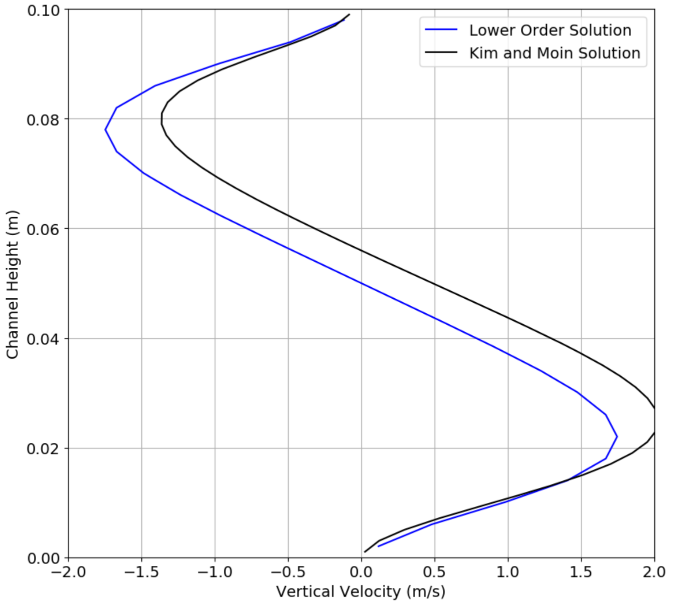

Figure 21: Plot comparing the vertical velocities found through both Kim and Moin and lower order fractional step methods at $t = 1.25 \times 10^{-3}\,s$ and $x = 0.025\,m$.

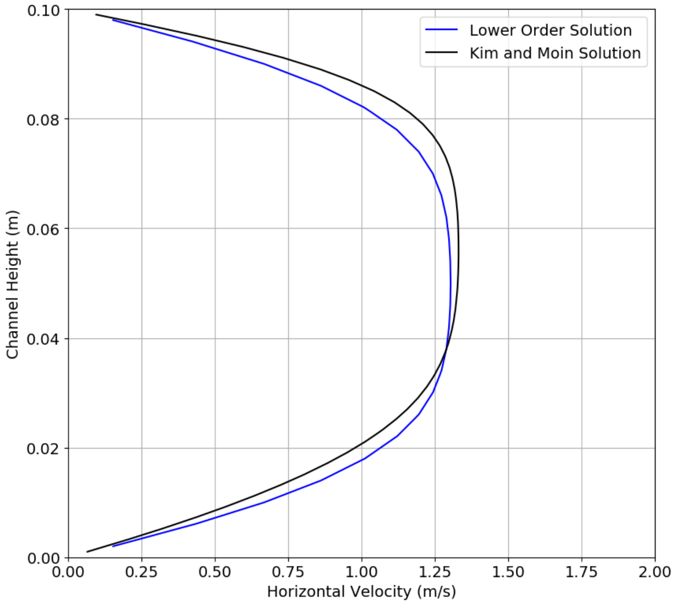

Figure 22: Plot comparing the vertical velocities found through both Kim and Moin and lower order fractional step methods at $t = 1.25 \times 10^{-3}\,s$ and $x = 0.1\,m$.

4.3 Channel Flow with Square Cylinder Results

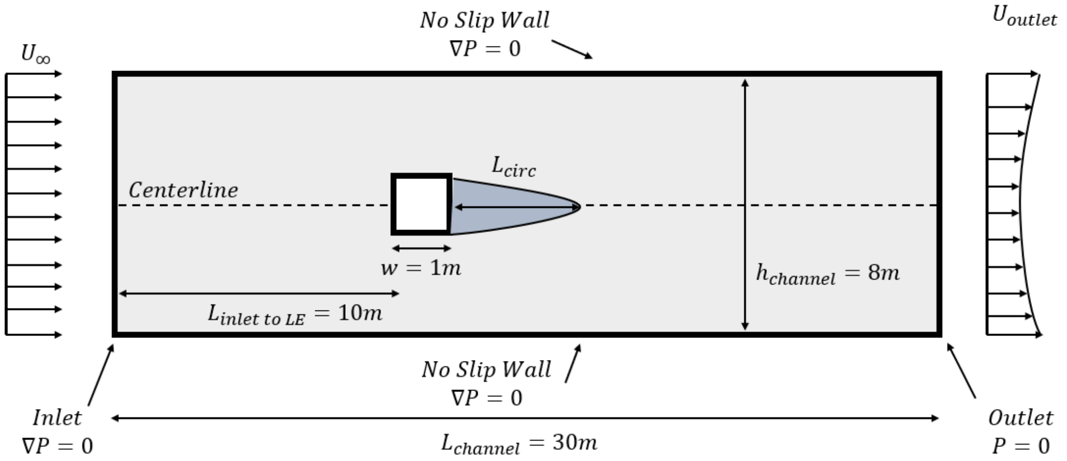

Figure 23: Simplified visualization of the domain setup, adapted from Breuer et al. (1999) [1].

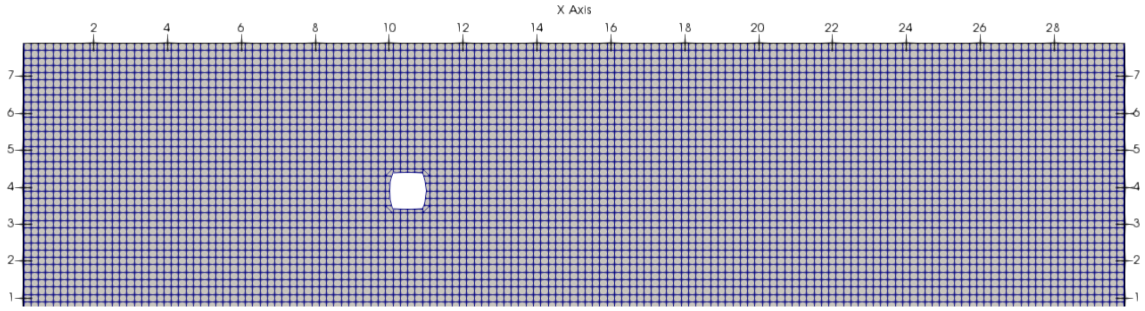

Figure 24: 150 x 40 cell setup for a square cylinder in channel flow.

The above figures are that of a simplified setup of the problem and the grid setup with the square cylinder put in place, as shown in figures 23 and 24 respectively. The grid setup is 150 cell sizes in length and 40 cell sizes in height. The 1 meter grid placed in the middle of the grid domain is 5 cells in length and 5 cells in height. The bottom left cell of the cylinder is placed at row 18 and column 51. Post processing is done on Paraview, and cells representing the square cylinder are patched over through a clip filter. Due to system settings, the clip that represents the square cylinder is somewhat curved unintentionally. The solution is done with the lower order, fractional step method from Saad (2019) to limit the amount of time it takes to achieve a solution. The following flow fields presented below is taken when the solution reaches steady state and at $Re = 1, 10, 50, 100$.

At $Re = 1$:

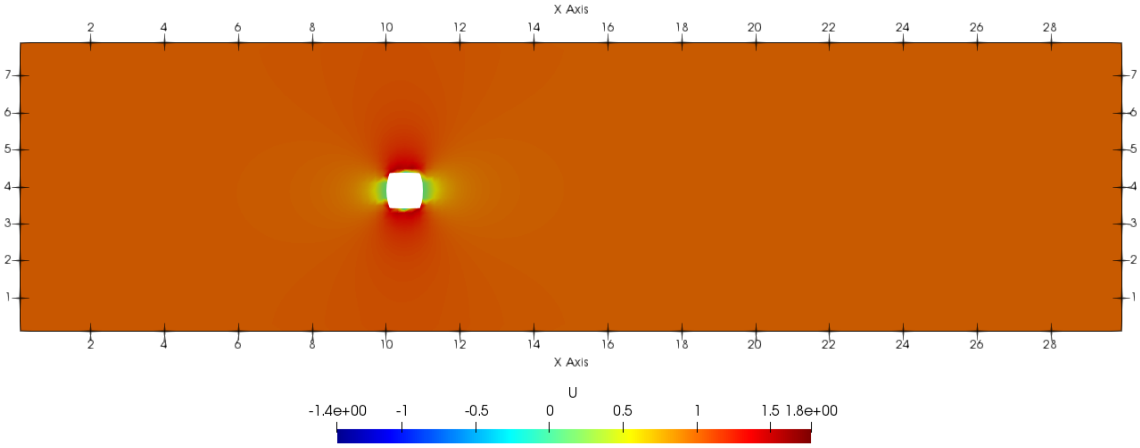

Figure 25: Flow over a square cylinder at $Re = 1$ in the x direction.

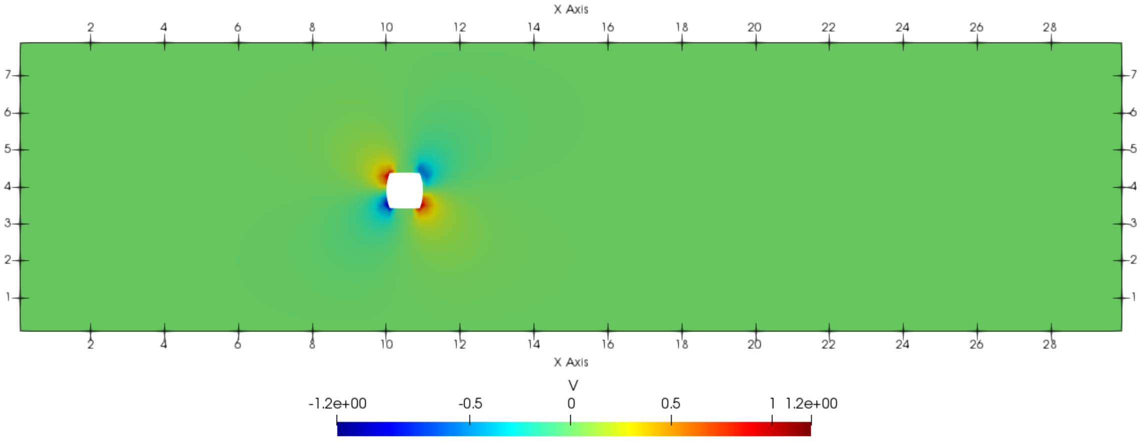

Figure 26: Flow over a square cylinder at $Re = 1$ in the y direction.

At $Re = 10$:

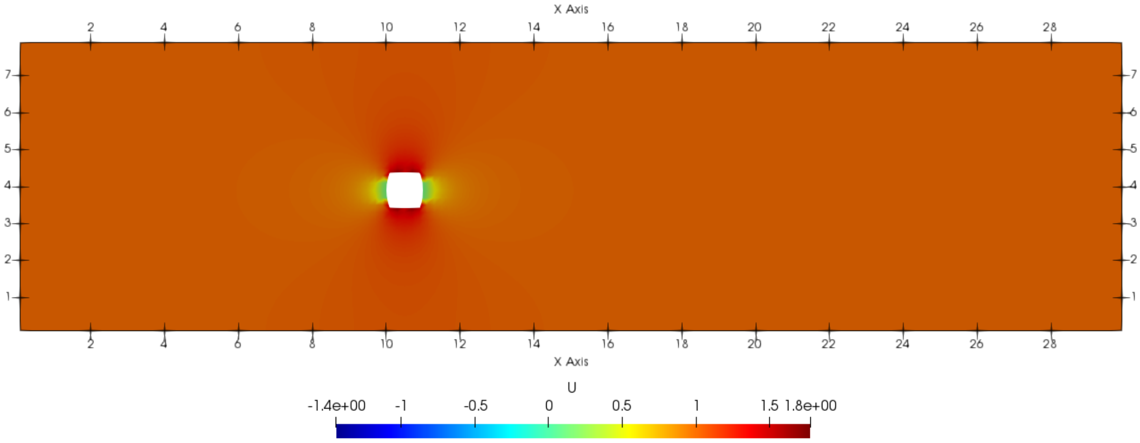

Figure 27: Flow over a square cylinder at $Re = 10$ in the x direction.

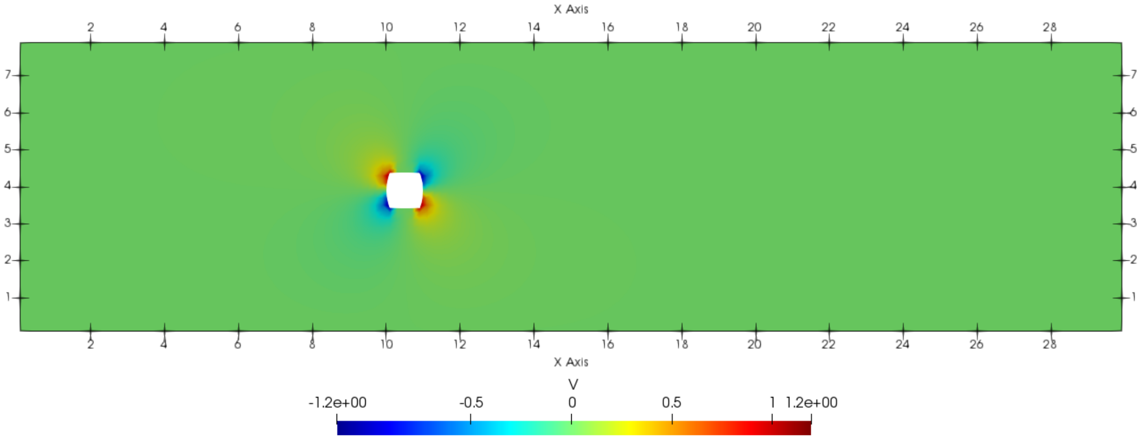

Figure 28: Flow over a square cylinder at $Re = 10$ in the y direction.

At $Re = 50$:

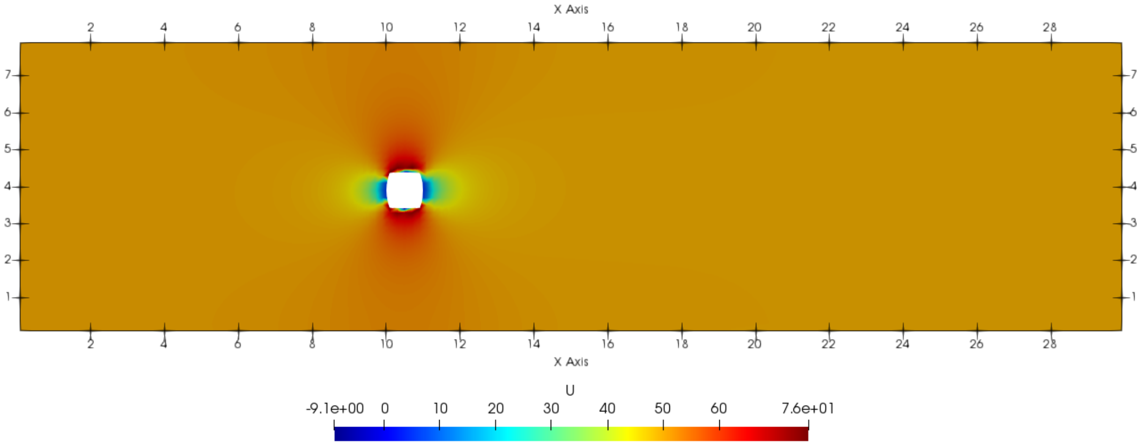

Figure 29: Flow over a square cylinder at $Re = 50$ in the x direction.

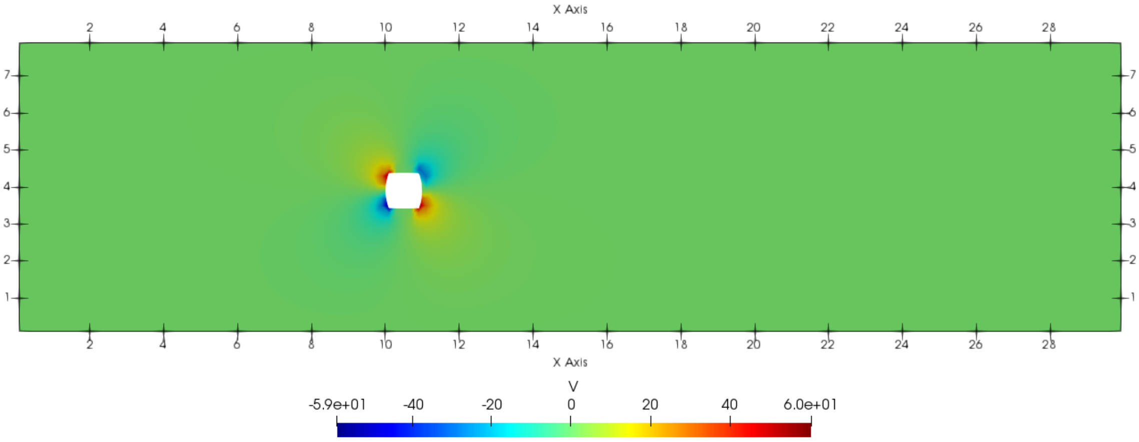

Figure 30: Flow over a square cylinder at $Re = 50$ in the y direction.

At $Re = 100$:

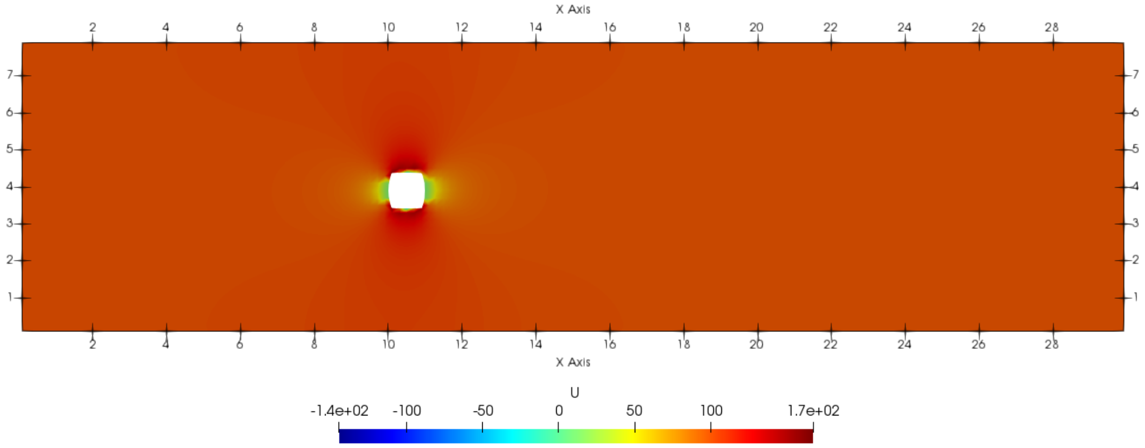

Figure 31: Flow over a square cylinder at $Re = 100$ in the x direction.

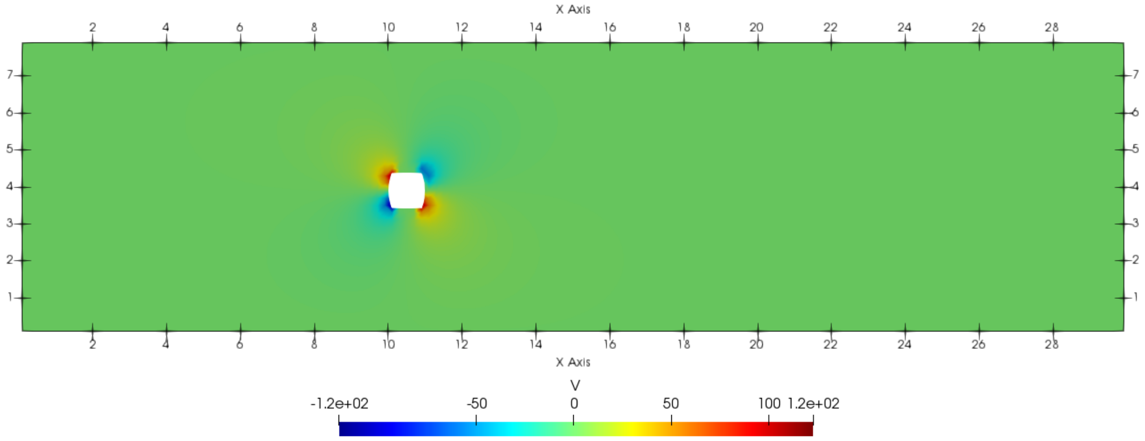

Figure 32: Flow over a square cylinder at $Re = 100$ in the y direction.

Centerline profiles of normalized horizontal velocity versus distance along x are plotted below in figure 33 for $Re = 1, 10, 50, 100$. It has been noticed that comparing the data to the analytical solution $L_r/D = 0.065 + 0.0554 \times Re$, where L_r is the length of the recirculation region, D is the length of the square cylinder, and Re the Reynolds number [1], the computed normalized profiles for this flow for all these Reynolds numbers are almost exactly the same. As shown in figure 33, there is no difference in the shape of the flow field when increasing or decreasing Re. This is incorrect, as the recirculation region is known to grow with increasing Reynolds numbers from Breuer et al. (1999) [1].

If the value of $D = 1\,m$, the value of L_r is $3\,m$ at $Re = 57$, which is the transition Re where flow begins to shed vortexes similar to a Von Kármán vortex street. Looking at figure 33, the normalized velocities stabilize around $x = 14\,m$, which suggests that the flow field being studied is held at $Re = 57$, even though the inlet velocity increases for increasing Re. This heavily implies that there's an error within the code, and further work has be be done polishing and removing such errors.

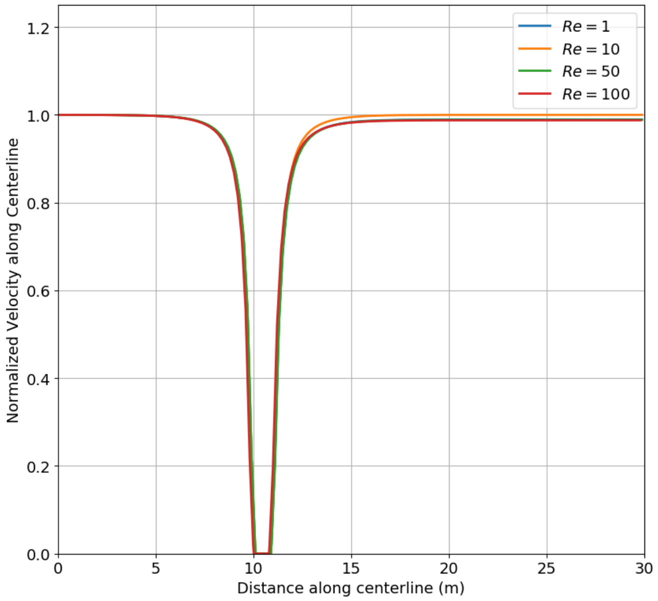

Figure 33: Centerline velocities for $Re = 1, 10, 50, 100$ for square cylinder flow in a channel.

5 Discussion

In terms of errors within the Kim and Moin fractional step method channel flow code, one error that has to be addressed is that uneven grid sizes, that is choosing an $N_x \neq N_y$, will cause the flow to diverge. Most likely, an error was undetected, and if one were to look into debugging the code, the convective discretization outlined in section 3.5.1 should be looked into, as the discretization accounts for uneven grid sizes N_x and N_y, and not even grid sizes N in x and y as is in Bisetti (2021) [10]. For convergence with the current code, the user must input an $N_x = N_y$. This will converge even if the length of the domain is different from the height $L_x \neq L_y$.

Further issues arise with the Generalized Minimal Residual Method (GMRES) FORTRAN 90 Code from Burkardt (2020) [12] used to solve the system of equations for the pressure update matrix in the lower-order fractional step method from Saad (2019), and u^* and v^* along with the pressure update matrix for the Kim and Moin method. The open-source code works well solving just one system of equations, as seen in figure 17. However when solving for 3 different system of equations per time step as in Kim and Moin, the real time for discretization takes three times longer than Saad and the channel flow becomes unsymmetrical with respect to the horizontal centerline, as seen in figure 19. It is not suggested to use the Kim and Moin unless supercomputer is available, since the code can be easily parallelized to solve for u^* and v^* simultaneously suggested.

In terms of discretization, the staggered grid is very complex to understand and solve without confusion and intense book keeping. Other methods to solving the Navier Stokes equations can be used, like Finite Difference Methods, as seen in Barba and Forsyth (2017) [13]. Although there are a myriad of resources for problems such as lid cavity flow for the staggered grid, as only Dirichlet boundary conditions are defined on the boundaries, there is not much resources on the methodology of Neumann boundary conditions for inlet and outlet conditions. Likewise, not much is placed on objects placed within the flow domain, such as a square cylinder. Though Neumann boundary conditions for channel flow have been found and solved using both low-order and Kim and Moin methods on a staggered grid, this is not the case for the square cylinder.

One problem that arises to finding the boundary condition for the cylinder is that of the corners. In the code, including the corner cells for the square cylinder will cause the pressure system of equations to be over-defined and unsolvable. This is due to the condition that at the wall, the condition $\nabla P = 0$. At the corners, P has to be solved with the upper/lower cell, $P_{i+1,j} - Pi, j = 0$ and the left/right cells $Pi, j + 1 - P_{i,j} = 0$. This creates two equations for every one corner cell, making the system over defined. One remedy to this problem is to enforce pressure boundary conditions to each corner cell to just its corresponding upper and lower cells, but this will cause velocities to "leak" into the cylinder, as the pressure condition is not enforced for one corner side exposed in the flow.

A remedy to this problem is to treat the corners as if they are not part of the cylinder. Using this method helps to achieve the flow patterns in u and the four petal flower pattern in v consistent with square cylinder in channel flow. However, the boundary layer profiles are not present in channel flow, which is untrue. The Reynolds number is not good either, as the centerline profile velocities are similar with increasing Re. This problem likely comes from the discretizing the non-dimensionalized form of the Navier Stokes equations as a way of simplifying the problem and doesn't reflect on density and viscosity well for the cylinder flow. Nevertheless, channel flow for the staggered grid is verified and accurate based on comparison of velocity profiles in the x direction and more work needs to be done with boundary conditions for the square cylinder.

6 Conclusion

In terms of the square cylinder, working with the staggered grid formation for the corner boundary conditions is very challenging, and the work presented up above on the square cylinder reflects the progress to achieving accurate low Re flow over the square cylinder. Modifications to the boundary conditions, such as the corners, were modified based on intuition and not on prior research on the topic.

In the case of this project, it has been found that if corners are to be included, this will cause the system of equations to solve for pressure to be over defined. Assigning the corner to only one side will retain the solvability of the pressure update system of equations, but will cause flow to "leak" into the cylinder. Therefore, future work should be done with the corner cells excluded.

It would be more beneficial if a "teach code" was given as a starting point to solving a problem in the Navier Stokes equations and more attention was given on solving the boundary conditions over the discretization methodology, since they are both equally important for students trying to solve special flow conditions much easily.

Because of time constraints and limited resources on how to treat boundary conditions, most deliverables proposed have been pushed back and need revision. Understanding and breaking down the staggered grid has been the most time consuming process, but a suitable code has been developed for channel flow, and can be easily suited for modification for other channel like flows, such as the backwards step problem.

References

[1] Michael Breuer, J Bernsdorf, Thomas Zeiser, and Franz Durst. Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-boltzmann and finite-volume. *International journal of heat and fluid flow*, 21(2):186–196, 2000.

[2] AK Dhiman, RP Chhabra, and V Eswaran. Flow and heat transfer across a confined square cylinder in the steady flow regime: effect of peclet number. *International Journal of Heat and Mass Transfer*, 48(21-22):4598–4614, 2005.

[3] Mohammad Rahnama and Hakimeh Hadi-Moghaddam. Numerical investigation of convective heat transfer in unsteady laminar flow over a square cylinder in a channel. *Heat transfer engineering*, 26(10):21–29, 2005.

[4] Subhankar Sen, Sanjay Mittal, and Gautam Biswas. Flow past a square cylinder at low reynolds numbers. *International Journal for Numerical Methods in Fluids*, 67(9):1160–1174, 2011.

[5] Dongjoo Kim and Haecheon Choi. A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids. *Journal of computational physics*, 162(2):411–428, 2000.

[6] Suhas V Patankar and D Brian Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion*, pages 54–73. Elsevier, 1983.

[7] Jonathan Hines. A comparative study of the simple and fractional step time integration methods for transient incompressible flows. Master's thesis, University of Waterloo, 2008.

[8] Tony Saad. ucfd: Four steps to navier-stokes. *TonySaad.net*, 2019.

[9] John Kim and Parviz Moin. Application of a fractional-step method to incompressible navier-stokes equations. *Journal of computational physics*, 59(2):308–323, 1985.

[10] Fabrizio Bisetti. Introduction to computational fluid dynamics. *N/A*, 2021.

[11] Saba Golshaahi Sumesaraayi and Markus Ihmsen. Planar poiseuille flow (2-d). *PreonLab*, 2021.

[12] John Burkardt. Mgmres. *Department of Math, University of South Carolina*, 2020.

[13] Lorena A. Barba and Gilbert F. Forsyth. 12 steps to navier stokes. *Blog*, 2013.