PRINCETON
UNIVERSITY

Numerical Methods for 2D Incompressible Flow

Final Project

github.com/coursework-enb/APC523-Project

Date of Submission:
2025-05-09

Authors

Estéban Nocet–Binois
Marie Joe Sawma
Amirmasoud Amini
Kristin Paragian

Contents

1 Introduction

The Navier–Stokes equations describe the motion of viscous fluids based on Newton's second law and the conservation of momentum. Due to their nonlinear and time-dependent nature, they are analytically intractable in most cases and typically require numerical methods. This project focuses on the two-dimensional (2D), incompressible form of the equations for Newtonian fluids with constant density and no external forces.

The **2D incompressible Navier–Stokes equations** for constant-density flow and no external forces are given by the following system:

$$\begin{cases} \dfrac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} = -\dfrac{1}{\rho}\nabla p + \nu \nabla^2 \vec{u}, & \text{(momentum equation)} \\ \nabla \cdot \vec{u} = 0, & \text{(continuity equation)} \end{cases} \tag{1}$$

In this system:

- $\vec{u} = (u, v)$ is the velocity vector field in two spatial dimensions,
- p is the pressure field,
- ρ is the constant fluid density,
- ν is the kinematic viscosity.

Equation (1) expresses the conservation laws that govern incompressible fluid flow. The momentum equation accounts for acceleration driven by convective transport, pressure gradients, and viscous diffusion. The continuity equation enforces the incompressibility condition by requiring the velocity field to remain divergence-free throughout the domain.

1.1 Momentum Equation in the x-Direction

The x-momentum equation governs the horizontal component of the velocity field, accounting for how u evolves due to self-advection, cross-advection by v, pressure gradients in the x direction, and viscous diffusion. The equation is given by:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right). \tag{2}$$

The left-hand side includes time-dependent and nonlinear convective terms, while the right-hand side includes pressure gradients and diffusion due to viscosity. The balance of these terms determines whether the flow is dominated by inertia, pressure forces, or viscosity.
When ν is small (i.e., high Reynolds number), the convective terms dominate, which can lead to the formation of vortices and other complex structures.

1.2 Momentum Equation in the y-Direction

The equation y-momentum governs the vertical component of the velocity field. It is analogous to the x-momentum equation but describes how v changes over time and space:

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right). \tag{3}$$

Together with the continuity equation, this equation ensures a coupled, consistent evolution of the flow in two dimensions. Numerical schemes must discretize both momentum equations simultaneously while preserving the divergence-free constraint imposed by incompressibility.

1.3 Pressure-Poisson Equation

The pressure-Poisson equation governs the spatial distribution of pressure in incompressible flows and arises from applying the divergence operator to the momentum equations along with the incompressibility constraint:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = -\rho\left(\frac{\partial u}{\partial x}\frac{\partial u}{\partial x} + 2\frac{\partial u}{\partial y}\frac{\partial v}{\partial x} + \frac{\partial v}{\partial y}\frac{\partial v}{\partial y}\right) \tag{4}$$

This equation ensures that the velocity field remains divergence-free by enforcing mass conservation. The right-hand side of the equation captures the non-linear interactions of velocity gradients, including both self-advection and cross-advection terms. These terms represent how momentum is redistributed within the flow. The solution to this elliptic equation provides the pressure correction necessary to maintain incompressibility at each time step in numerical simulations.

2 Adaptive Time Stepping Strategy

To ensure both stability and efficiency in time-dependent simulations of the Navier–Stokes equations, adaptive time stepping is employed. Two mathematical strategies are used to dynamically adjust the timestep Δt: one based on the relative change in the velocity field and the other on the Courant–Friedrichs–Lewy (CFL) condition.

2.1 Adaptation Based on Relative Velocity Change

This method evaluates how much the velocity field has changed between consecutive time steps. Let $u_{i,j}^n$ and $v_{i,j}^n$ denote the horizontal and vertical velocity components, respectively, at spatial location (i, j) and time level n.
The relative change in each component is calculated as:

$$\varepsilon_u = \max_{i,j}\left|\frac{u_{i,j}^{n+1} - u_{i,j}^n}{u_{i,j}^n + \delta}\right|, \quad \varepsilon_v = \max_{i,j}\left|\frac{v_{i,j}^{n+1} - v_{i,j}^n}{v_{i,j}^n + \delta}\right|$$

where $\delta > 0$ is a small constant added to avoid division by zero (e.g., $\delta = 10^{-10}$).
The total maximum relative change is given by:

$$\varepsilon = \max(\varepsilon_u, \varepsilon_v)$$

A predefined tolerance $\tau > 0$ is used to determine whether the step is accepted:

$$\text{If } \varepsilon \leq \tau, \quad \text{accept the step and set } \Delta t_{\text{new}} = \min(2\Delta t, \Delta t_{\max})$$

$$\text{If } \varepsilon > \tau, \quad \text{reject the step and set } \Delta t_{\text{new}} = \max(0.5\Delta t, \Delta t_{\min})$$

This approach ensures that time steps are enlarged when the solution evolves slowly, and reduced when rapid changes are detected, balancing computational efficiency with temporal resolution.

2.2 Adaptation Based on the CFL Condition

The Courant–Friedrichs–Lewy (CFL) number is a dimensionless quantity defined as:

$$\text{CFL} = \max\left(\frac{|u|\Delta t}{\Delta x}, \frac{|v|\Delta t}{\Delta y}\right)$$

It quantifies the ratio of physical information propagation to the numerical grid resolution. For stability, especially in explicit schemes, the CFL number must remain below a certain threshold (commonly less than 1).

Given a target CFL value, the time step is updated based on the following rule:

$$\text{If CFL} \leq C_{\text{target}}, \quad \text{accept the step and set } \Delta t_{\text{new}} = \min(2\Delta t, \Delta t_{\max})$$

$$\text{If CFL} > C_{\text{target}}, \quad \text{reject the step and set } \Delta t_{\text{new}} = \max(0.5\Delta t, \Delta t_{\min})$$

This approach directly controls numerical stability based on the flow speed and grid spacing, ensuring the simulation adheres to the CFL condition at each time step.

3 Spatial Discretization

Spatial discretization transforms the continuous partial differential equations of fluid motion into a discrete system that can be solved numerically. In this solver, three approaches are implemented for computing the right-hand side of the momentum equations: central finite difference, upwind finite difference, and finite volume.

3.1 Finite Difference Method (FDM)

3.1.1 Central Scheme

The central finite difference scheme uses centered stencils for approximating spatial derivatives. Convective terms are computed using symmetric differences, ensuring second-order accuracy in space:

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} \tag{5}$$

$$\frac{\partial u}{\partial y} \approx \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} \tag{6}$$

and similarly for $\frac{\partial v}{\partial x}$ and $\frac{\partial v}{\partial y}$.
Diffusion terms are discretized using the standard 5-point stencil approximations for the Laplace operator ∇^2 applied to each velocity component:

$$\nabla^2 u \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} \tag{7}$$

and similarly for $\nabla^2 v$.
Lastly, the pressure gradient is also discretized using central differences:

$$\frac{\partial p}{\partial x} \approx \frac{p_{i+1,j} - p_{i-1,j}}{2\Delta x} \tag{8}$$

$$\frac{\partial p}{\partial y} \approx \frac{p_{i,j+1} - p_{i,j-1}}{2\Delta y} \tag{9}$$

This method provides a balance between simplicity and accuracy but may introduce numerical oscillations in convection-dominated regimes.

3.1.2 Upwind Scheme

To enhance stability in the presence of strong advection, an upwind finite difference scheme is provided. The directionality of the velocity field determines the stencil used for each derivative. For a positive velocity component $u_{i,j} > 0$, we compute the first derivatives as:

$$\frac{\partial u}{\partial x} \approx \frac{u_{i,j} - u_{i-1,j}}{\Delta x}, \quad \frac{\partial v}{\partial x} \approx \frac{v_{i,j} - v_{i-1,j}}{\Delta x} \tag{10}$$

Conversely, if $u_{i,j} < 0$, the stencil shifts forward:

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1,j} - u_{i,j}}{\Delta x}, \quad \frac{\partial v}{\partial x} \approx \frac{v_{i+1,j} - v_{i,j}}{\Delta x} \tag{11}$$

The same logic applies in the y-direction based on the sign of $v_{i,j}$. The diffusive and pressure terms remain central. This method introduces numerical diffusion, which prevents oscillations but can reduce accuracy in smooth regions.

3.2 Finite Volume Method (FVM)

The finite volume method discretizes the equations by integrating them over control volumes. Fluxes are computed at cell faces using averaged values to enforce conservation. This method is particularly well-suited for handling complex geometries and ensures global conservation of momentum.
Face-centered velocities are approximated using averaging of adjacent cell-centered values. The convective fluxes for the u- and v-momentum equations are then computed as:

$$\text{flux}_{\text{conv},u} = \frac{u_e^2 - u_w^2}{\Delta x} + \frac{u_{i,j+1} v_n - u_{i,j-1} v_s}{\Delta y} \tag{12}$$

$$\text{flux}_{\text{conv},v} = \frac{u_e v_e - u_w v_w}{\Delta x} + \frac{v_n^2 - v_s^2}{\Delta y} \tag{13}$$

The viscous diffusion terms and the pressure gradient are solved using the same central difference schemes.

4 Predictor

The prediction step advances the velocity fields in time without incorporating the pressure correction. Several time integration strategies are implemented, each designed to balance accuracy, stability, and computational efficiency.

4.1 Forward Euler Method

The forward Euler scheme is a first-order explicit method. Given the velocity fields u^n and v^n at time t^n, and the time derivatives $\partial_t u$, $\partial_t v$ computed from the right-hand side of the momentum equation, the update reads:

$$u^{n+1} = u^n + \Delta t \left(\frac{\partial u}{\partial t}\right)^n, \quad v^{n+1} = v^n + \Delta t \left(\frac{\partial v}{\partial t}\right)^n$$

This method is simple to implement but conditionally stable and generally requires small time steps for accuracy and stability.

4.2 Fourth-Order Runge–Kutta (RK4) Method

The classical Runge–Kutta method of fourth order (RK4) is an explicit integration scheme that achieves high accuracy by evaluating the time derivative at four intermediate stages

$$\begin{aligned}
k_1 &= f(u^n, v^n) \\
k_2 &= f\left(u^n + \frac{\Delta t}{2} k_1^u,\, v^n + \frac{\Delta t}{2} k_1^v\right) \\
k_3 &= f\left(u^n + \frac{\Delta t}{2} k_2^u,\, v^n + \frac{\Delta t}{2} k_2^v\right) \\
k_4 &= f\left(u^n + \Delta t\, k_3^u,\, v^n + \Delta t\, k_3^v\right)
\end{aligned}$$

The velocity is then updated by:

$$\begin{aligned}
u^{n+1} &= u^n + \frac{\Delta t}{6}(k_1^u + 2k_2^u + 2k_3^u + k_4^u) \\
v^{n+1} &= v^n + \frac{\Delta t}{6}(k_1^v + 2k_2^v + 2k_3^v + k_4^v)
\end{aligned}$$

This method provides high accuracy and is stable for moderately large time steps under non-stiff conditions.

4.3 Predictor–Corrector Method

This method uses a two-step procedure to achieve second-order temporal accuracy.

Predictor step:

$$u^* = u^n + \Delta t \left(\frac{\partial u}{\partial t}\right)^n, \quad v^* = v^n + \Delta t \left(\frac{\partial v}{\partial t}\right)^n$$

Corrector step:

$$u^{n+1} = u^n + \frac{\Delta t}{2}\left[\left(\frac{\partial u}{\partial t}\right)^n + \left(\frac{\partial u}{\partial t}\right)^*\right], \quad v^{n+1} = v^n + \frac{\Delta t}{2}\left[\left(\frac{\partial v}{\partial t}\right)^n + \left(\frac{\partial v}{\partial t}\right)^*\right]$$

Here, $(\partial_t u)^*$ is evaluated using the predicted fields u^*, v^*.

4.4 Semi-Implicit Scheme with Jacobi Diffusion Solver

In this approach, the nonlinear advection and pressure terms are treated explicitly, while the viscous (diffusion) terms are handled implicitly using a fixed number of Jacobi iterations. The method decouples advection from diffusion, improving stability for high Reynolds number flows.

Explicit advection step:

$$u^* = u^n + \Delta t \left(\frac{\partial u}{\partial t}\right)^n_{\text{adv}}, \quad v^* = v^n + \Delta t \left(\frac{\partial v}{\partial t}\right)^n_{\text{adv}}$$

Implicit diffusion step (Jacobi iteration):

To solve the diffusion equation:

$$\frac{u^{n+1} - u^*}{\Delta t} = \nu \left(\frac{\partial^2 u^{n+1}}{\partial x^2} + \frac{\partial^2 u^{n+1}}{\partial y^2}\right)$$

The Jacobi scheme updates the solution iteratively as:

$$u_{i,j}^{(k+1)} = \frac{1}{1 + 2\alpha\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)} \left[u^*_{i,j} + \alpha \left(\frac{u^{(k)}_{i+1,j} + u^{(k)}_{i-1,j}}{\Delta x^2} + \frac{u^{(k)}_{i,j+1} + u^{(k)}_{i,j-1}}{\Delta y^2}\right)\right]$$

with $\alpha = \nu \Delta t$.

The same update is applied to v, and after a fixed number of iterations, the resulting fields are accepted as u^{n+1} and v^{n+1}.

5 Boundary Conditions

Accurate boundary condition implementation is essential for ensuring the physical realism and numerical stability of fluid . This solver supports two benchmark configurations—*Taylor-Green Vortex (TGV)* and *Lid-Driven Cavity (LDC)*,each requiring specific boundary conditions for the velocity, pressure, and right-hand-side (RHS) derivatives in the momentum equations.

5.1 Velocity Boundary Conditions

In the *Taylor-Green Vortex*, periodic boundary conditions are applied by matching velocity values on opposing domain edges to maintain flow continuity. In the *Lid-Driven Cavity*, no-slip conditions are enforced on all walls except the top, where a horizontal lid motion with $u = 1$, $v = 0$ drives the flow.

5.2 Pressure Boundary Conditions

In the TGV case, periodic conditions are similarly applied to the pressure field, mirroring the implementation for velocity.

For the LDC problem, homogeneous Neumann conditions are imposed by setting the pressure gradient normal to each wall to zero. This avoids introducing artificial pressure buildup near the walls and aligns with the incompressibility constraint.

5.3 Time Derivative Boundary Conditions

In the *Taylor-Green Vortex*, periodicity is preserved by mirroring values across boundaries.
For the *Lid-Driven Cavity*, derivatives are set to zero at walls to uphold fixed velocity values and enforce the no-slip condition.

6 Projection Method

To enforce incompressibility in the numerical solution of the Navier–Stokes equations, a projection method is employed. This approach decouples the velocity and pressure fields by first computing an intermediate velocity (without enforcing divergence-free constraints) and then correcting it using a pressure Poisson equation. The process consists of three primary components: computing the right-hand side (RHS) for the pressure equation, solving the pressure Poisson equation, and correcting the velocity field.

6.1 Right-Hand Side of the Pressure Poisson Equation

The divergence of the intermediate velocity field (u^*, v^*) is used to construct the RHS of the pressure Poisson equation. Additionally, a nonlinear correction term is included to enhance pressure accuracy:

$$b = \frac{1}{\Delta t}\left(\frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y}\right) - \left(\frac{\partial u^*}{\partial x}\right)^2 - 2\frac{\partial u^*}{\partial y}\frac{\partial v^*}{\partial x} - \left(\frac{\partial v^*}{\partial y}\right)^2$$

6.2 Pressure Poisson Equation

To enforce the incompressibility constraint $\nabla \cdot \mathbf{u}^{n+1} = 0$, the pressure field p is computed by solving the Poisson equation:

$$\nabla^2 p = b$$

In discrete form, a five-point stencil is applied on a uniform grid, yielding:

$$\frac{p_{i+1,j} + p_{i-1,j}}{\Delta x^2} + \frac{p_{i,j+1} + p_{i,j-1}}{\Delta y^2} - 2\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right) p_{i,j} = b_{i,j}$$

This equation is solved by Jacobi method :

$$p_{i,j}^{(k+1)} = \frac{1}{2(\Delta x^2 + \Delta y^2)}\left[\Delta y^2(p_{i+1,j}^{(k)} + p_{i-1,j}^{(k)}) + \Delta x^2(p_{i,j+1}^{(k)} + p_{i,j-1}^{(k)}) - \Delta x^2 \Delta y^2 b_{i,j}\right]$$

6.3 Velocity Correction Step

After computing the pressure field, the intermediate velocity (u^*, v^*) is corrected to obtain a divergence-free field:

$$u^{n+1} = u^* - \Delta t \frac{\partial p}{\partial x}, \quad v^{n+1} = v^* - \Delta t \frac{\partial p}{\partial y}$$

These derivatives are also discretized using central differences:

$$\frac{\partial p}{\partial x} \approx \frac{p_{i+1,j} - p_{i-1,j}}{2\Delta x}, \quad \frac{\partial p}{\partial y} \approx \frac{p_{i,j+1} - p_{i,j-1}}{2\Delta y}$$

6.4 Semi-Implicit Diffusion Correction

In certain variants of the solver, the velocity field is further diffused using a semi-implicit scheme. The Laplacian of the velocity is applied to improve stability, especially in viscous-dominated regimes:

$$u_{i,j}^{n+1} \leftarrow u_{i,j}^{n+1} + \nu \Delta t \nabla^2 u_{i,j}, \quad v_{i,j}^{n+1} \leftarrow v_{i,j}^{n+1} + \nu \Delta t \nabla^2 v_{i,j}$$

with the discrete Laplacian defined as:

$$\nabla^2 u_{i,j} \approx \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{\Delta x^2}$$

6.5 Boundary Conditions

During each stage of the projection method, boundary conditions are applied:

- Velocity boundaries (e.g., no-slip or periodic) are enforced after both the advection and pressure correction steps.
- Pressure boundaries depend on the simulation case. For periodic domains, pressure values are copied across boundaries to ensure continuity.

7 Validation

7.1 Overview

To evaluate the accuracy and robustness of the numerical solver, two classical benchmark problems are used: the *Taylor-Green Vortex* and the *Lid-Driven Cavity*. These benchmarks allow for both qualitative and quantitative validation of the simulation results. The initial conditions are set using analytical expressions or physically motivated configurations, and validation is performed through comparison against known analytical solutions or reference data.

7.2 Initialization of Benchmark Problems

Initialization is handled based on the selected test case:

- **Taylor-Green Vortex:** The velocity and pressure fields are initialized using the analytical solution:

$$\begin{aligned} u(x, y, t) &= \sin(\pi x) \cos(\pi y) \exp(-2\nu\pi^2 t), \\ v(x, y, t) &= -\cos(\pi x) \sin(\pi y) \exp(-2\nu\pi^2 t), \\ p(x, y, t) &= -\frac{1}{4}(\cos(2\pi x) + \cos(2\pi y)) \exp(-4\nu\pi^2 t). \end{aligned}$$

  This provides a time-dependent analytical reference solution for validating kinetic energy decay.

- **Lid-Driven Cavity:** The domain is initialized with zero velocity and pressure, except for the top boundary where a constant tangential velocity is imposed: $u = 1$, $v = 0$. This setup mimics the classical lid-driven cavity configuration, widely used for assessing internal flow solvers under no-slip boundary conditions.

7.3 Validation Strategy

Each benchmark has its own validation metrics:

- **Taylor-Green Vortex:** At a given simulation time t, the simulated kinetic energy is compared against the analytical solution. The absolute error is computed as:

$$\text{Error}_{KE} = |KE_{\text{sim}} - KE_{\text{analytical}}|$$

  This comparison quantifies the solver's accuracy in capturing the decay of kinetic energy over time.

- **Lid-Driven Cavity:** At $t = 2.5$ seconds, the simulated minimum stream function value is compared against a reference value $\psi_{\min} = -0.061076605$ reported in literature. The absolute error is given by:

$$\text{Error}_{\psi} = |\min(\psi)_{\text{sim}} - \psi_{\min}|$$

  This provides a robust indicator of the solver's ability to reproduce correct vortex formation and boundary-layer behavior.

8 Results

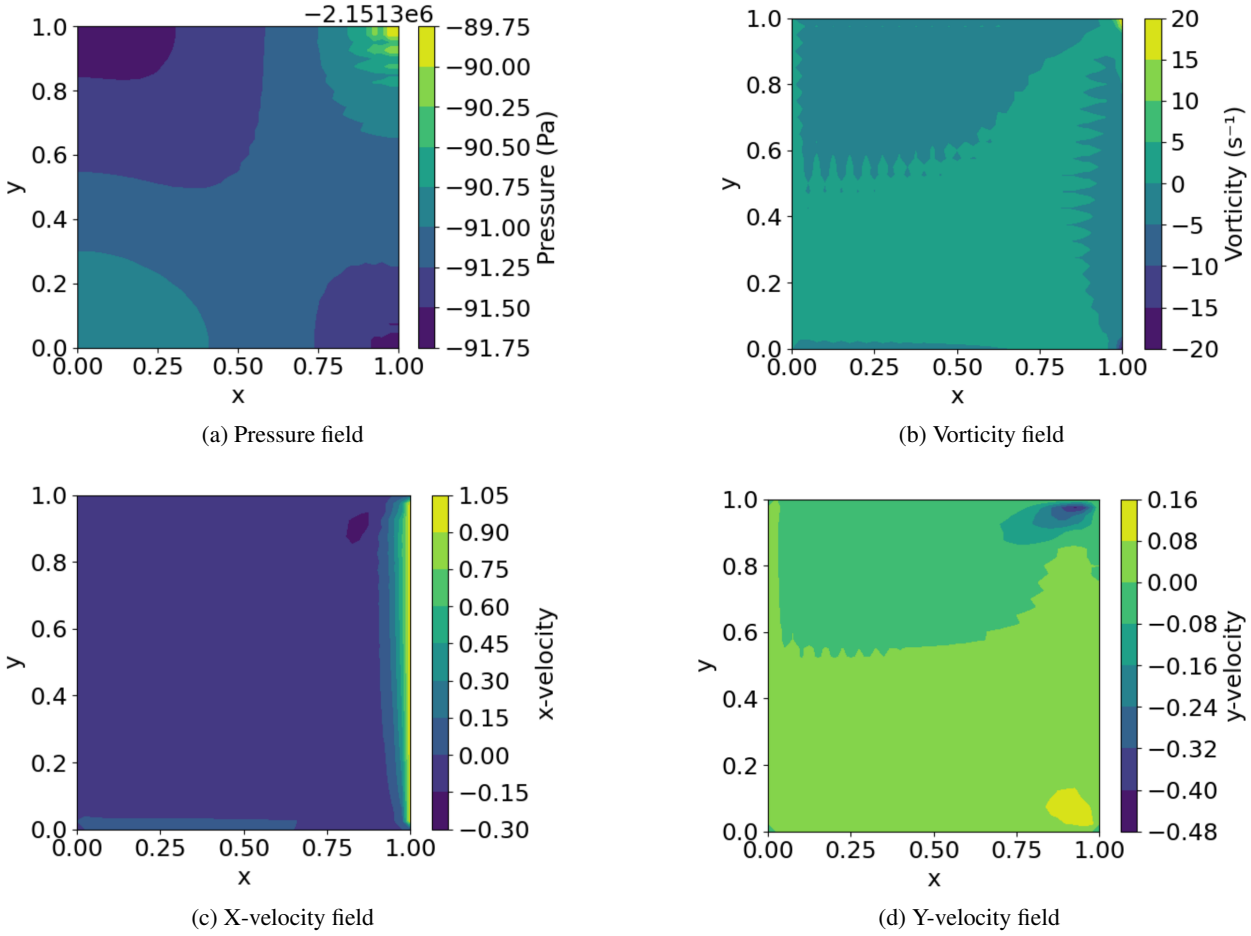
The code for the visual output in both benchmarks (Experiment 1 and 2) can be found directly in the in the demo notebook in our Github repository. The code for the last three experiments can be found in the output folder.

8.1 Experiment 1: LDC over 2.5s

Performing simulations for a full integration time requires a lot of resources: using 15 cores it took us 40 minutes to integrate for the Lid-Driven Cavity validation over 2.5s. We obtained:

- Simulated minimum stream function: -0.475840765
- Reference minimum stream function: -0.061076605
- Absolute error: 0.414764160

The results shown in Fig. 1 were obtained using a 2.1-based adaptive time step, with 10% rule and an initial dt of 0.004. The solver used a semi-implicit integrator with a finite difference upwind discretization of the RHS.
However, because of the amount of time simulating for the full integration period takes, we will mostly provide results for the Taylor-Green Vortex benchmark. Indeed since this problem has an analytical solution, we can easily employ it for any amount of time, or any number of iterations.



(a) Pressure field

(b) Vorticity field

(c) X-velocity field

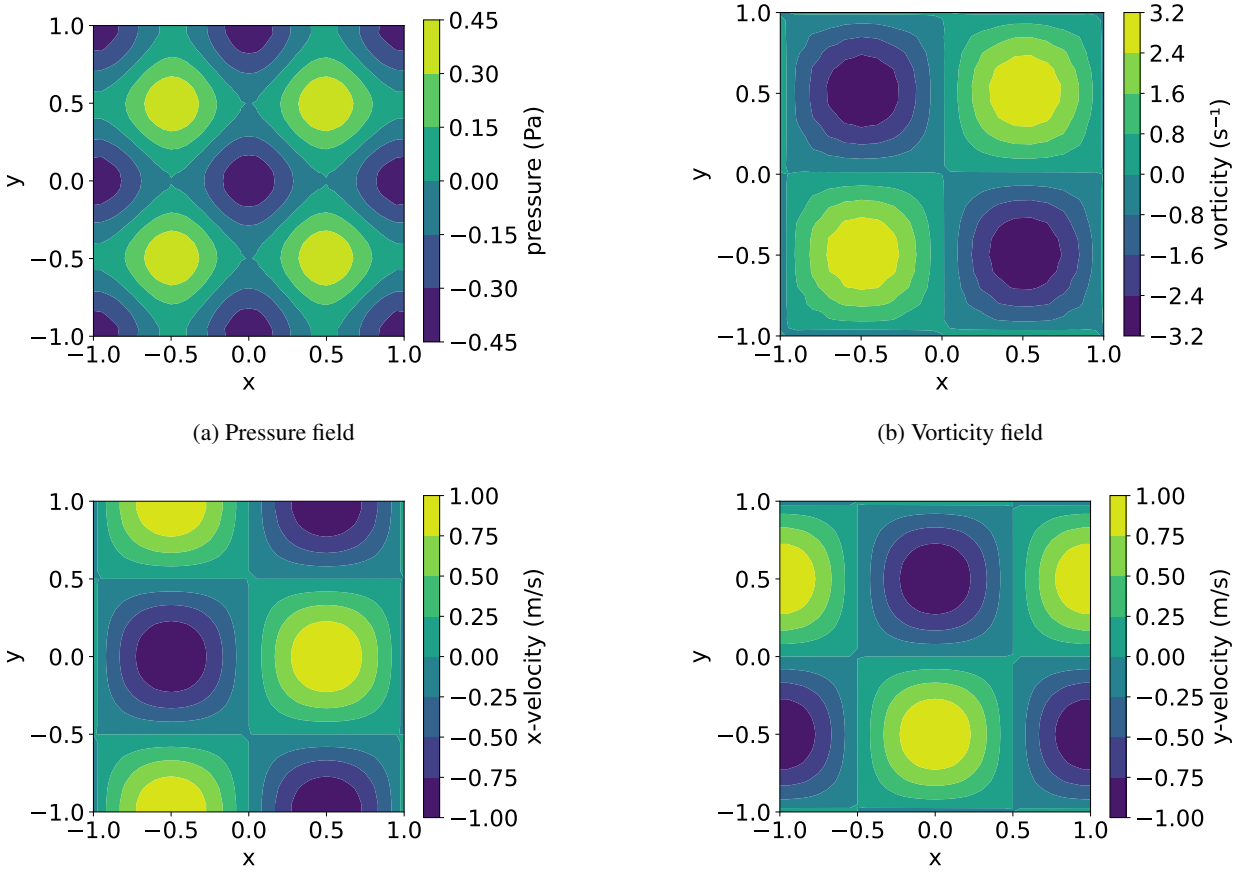
(d) Y-velocity field

Figure 1: Lid-driven cavity simulation results for the pressure, vorticity, and velocity components after 2.5 seconds. Results were obtained using adaptive time stepping (10% rule) with initial $dt = 0.004$, a viscosity of 1×10^{-3}, with a semi implicit integrator and a finite difference upwind scheme

8.2 Experiment 2: TGV over 10^4 Steps

In Fig. 2 we display the results when testing with the TGV problem for 10^4 time steps. We will be using this as main benchmark for the next three experiments as it is more flexible and less expensive. We obtain:

- Simulated kinetic energy: 1.046098648
- Analytical kinetic energy: 1.049823078
- Absolute error: 0.003724430



(a) Pressure field

(b) Vorticity field

(c) X-velocity field

(d) Y-velocity field

Figure 2: Taylor-Green vortex simulation results for the pressure, vorticity, and velocity components after 10000 time steps. Results were obtained using adaptive time stepping (10% rules) with initial $dt = 0.004$, viscosity of 1×10^{-3}, with a semi implicit integrator and a finite difference upwind scheme.

8.3 Experiment 3: Comprehensive Comparison

In this experiment, we conducted a comprehensive performance comparison across various combinations of spatial discretization schemes and time integrators, for two viscosity values: $\nu = 10^{-3}$ and $\nu = 10^{-5}$. We fixed the grid size and number of time steps to isolate the effect of discretization and integration strategies. The left bar chart in Figure 3 reports compute time, while the red curve overlays the corresponding error.

Overall, we observe in Fig. 3 that higher-order integrators resulted in lower error but sometimes at the cost of higher compute time. On the other side, more stable diffusion schemes such as the upwind scheme led to lower compute time (and more stable behavior which was leveraged in the last experiment), but at the cost of accuracy. Notably, when $\nu = 10^{-5}$ (i.e., low viscosity and more stiff systems), schemes with explicit integrators like Euler or RK4 tend to perform poorly or even fail to converge, whereas semi-implicit schemes remain stable and accurate.

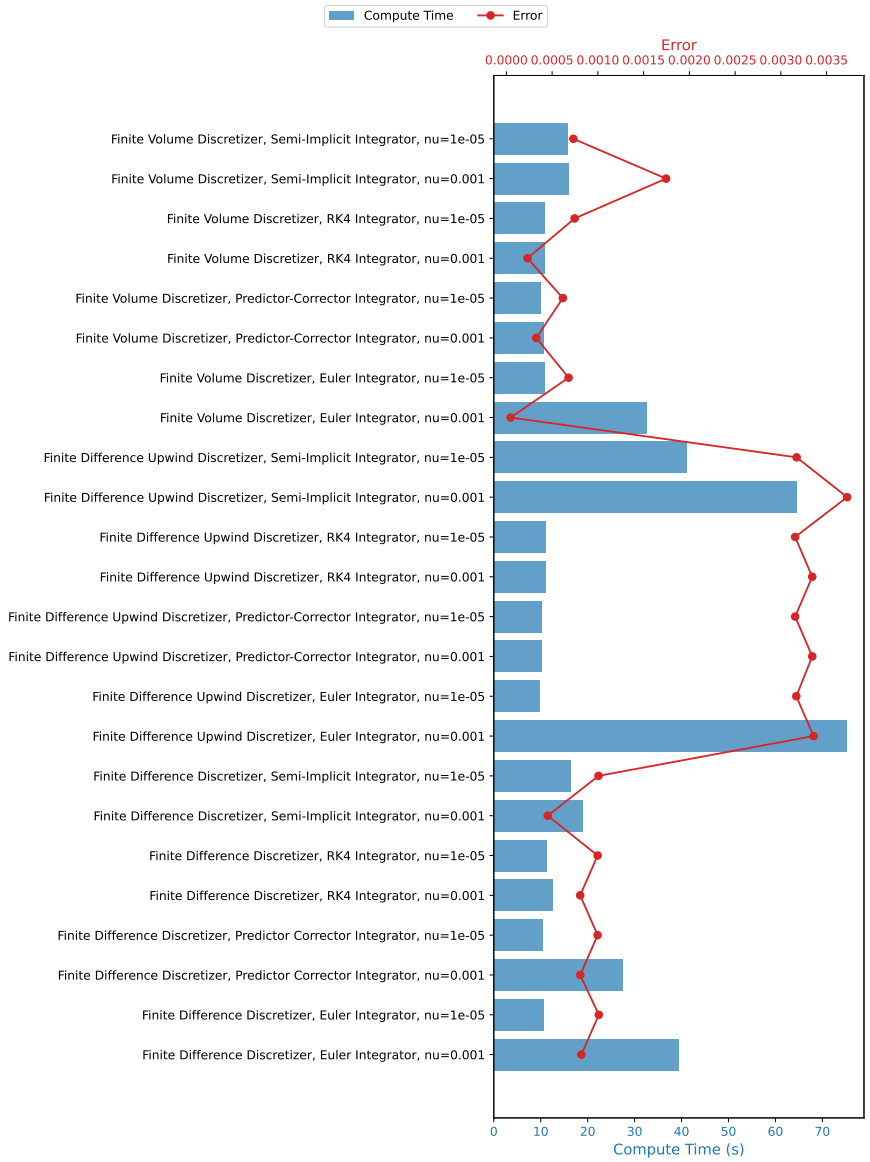


Figure 3: Comparison of accuracy and compute time for all scheme combinations with two different values of viscosity $1 \times 10^{-3}, 1 \times 10^{-5}$. We benchmarked all of these schemes against the TGV problem and ran it for 10000 steps, using adaptive time steps (which was necessary as most schemes were failing for fixed time step).

8.4 Experiment 4: Effect of Time Step Size

The results of Experiment 4 (cf. Fig. 4) demonstrate that reducing the time step size dt consistently lowers the validation error, highlighting the stability and accuracy of the predictor–corrector scheme. This trend aligns with the expected temporal convergence behavior of the numerical method. As dt increases, the error increases noticeably,

particularly for values exceeding 0.01, where the impact of time discretization becomes more significant. Despite this, the solver remains stable across the entire range of tested dt values, confirming the efficiency of the implemented scheme.

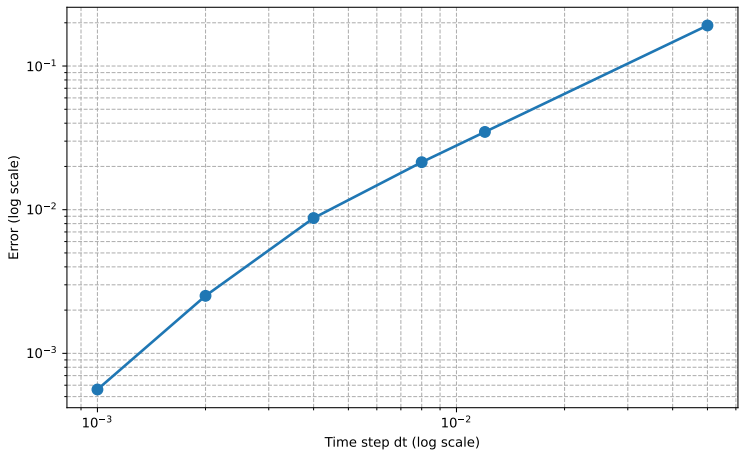


Figure 4: The temporal accuracy of our solver were evaluated by varying the time step size dt while keeping the grid size fixed. The predictor–corrector scheme was employed, with a finite volume discretization for the spatial terms. The simulation is run up to a fixed physical time of $T = 0.3$ for each configuration, and the validation error is recorded at the end of the simulation.

8.5 Experiment 5: Effect of Grid Size

Here we want to explore how our solver scales with grid size (dx, dy), running it for increasingly refined grids. Since we want to explore the effect on compute time and accuracy through only changing the grid resolution (keeping the same dt, and amount of steps), we chose an upwind scheme, which was one of the few schemes to consistently work with fixed dt (the combination used in previous experiment being the other example). Indeed, one major challenge we faced when running batches of simulations was overflow issues, easily arising from quadratic terms in velocities. These terms can rapidly exceed machine representation when using unstable schemes that exhibit artificial growth.
We observed in Fig. 5 that error decreased with smaller grid spacing, indicating improved spatial accuracy. Compute time generally increased with finer grids due to the larger number of operations required.
However, because we were running all our experiments using 28 cores, we observe a loss in performance at coarsest resolutions which can be attributed to parallel computation overhead (increased idle waiting time). We assume that the workload per thread became too small to fully utilize the available cores. This analysis highlights the trade-off between accuracy and computational cost.

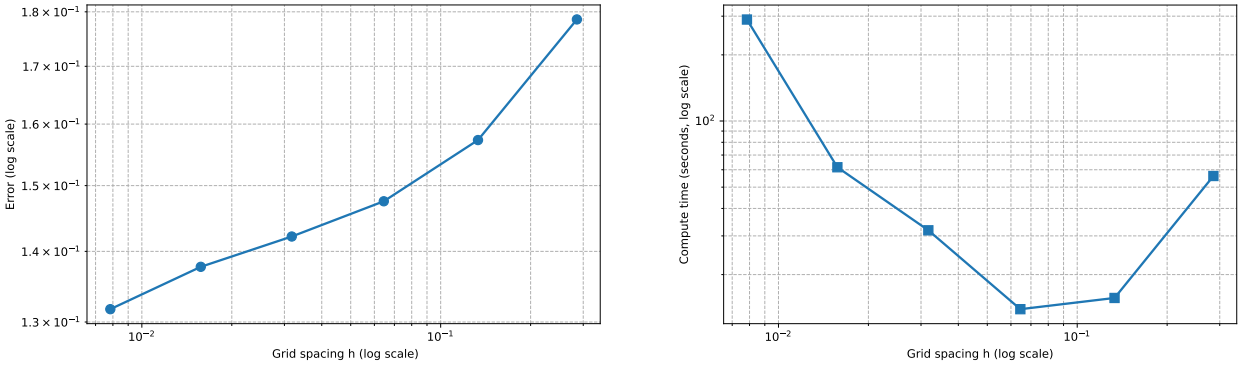


Figure 5: Performance (compute time and error) with respect to grid spacing $h = dx = dy$. We used a semi-implicit integrator with a finite difference upwind discretization of the spatial terms.

A Appendix

A.1 Semi-implicit Method Derivation

We begin with the 2D incompressible Navier-Stokes momentum equation for the u-velocity component (ignoring pressure for now):

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{14}$$

We discretize in time using a semi-implicit scheme:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} + \left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right)^{n} = \nu\nabla^2 u_{i,j}^{n+1} \tag{15}$$

Next, use first-order upwind or central differences for the advection terms at time n:

$$\left(u\frac{\partial u}{\partial x}\right)_{i,j}^{n} \approx u_{i,j}^{n} \cdot \frac{u_{i,j}^{n} - u_{i,j-1}^{n}}{\Delta x} \tag{16}$$

$$\left(v\frac{\partial u}{\partial y}\right)_{i,j}^{n} \approx v_{i,j}^{n} \cdot \frac{u_{i,j}^{n} - u_{i-1,j}^{n}}{\Delta y} \tag{17}$$

So the full advection term becomes:

$$\text{Advection}_{i,j}^{n} = u_{i,j}^{n} \cdot \frac{u_{i,j}^{n} - u_{i,j-1}^{n}}{\Delta x} + v_{i,j}^{n} \cdot \frac{u_{i,j}^{n} - u_{i-1,j}^{n}}{\Delta y} \tag{18}$$

Next, multiply both sides by Δt:

$$u_{i,j}^{n+1} - \Delta t \cdot \nu\nabla^2 u_{i,j}^{n+1} = u_{i,j}^{n} - \Delta t \cdot \text{Advection}_{i,j}^{n} \tag{19}$$

Then we discretize the Laplacian:

$$\nabla^2 u_{i,j}^{n+1} \approx \frac{u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1}}{\Delta x^2} + \frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{\Delta y^2} \tag{20}$$

Then plug into the time discretization

$$\begin{aligned} u_{i,j}^{n+1} - \Delta t \cdot \nu &\left(\frac{u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1}}{\Delta x^2} + \frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{\Delta y^2}\right) \\ &= u_{i,j}^{n} - \Delta t \cdot \text{Advection}_{i,j}^{n} \end{aligned} \tag{21}$$

Lastly, we solve for $u_{i,j}^{n+1}$:

$$\begin{aligned} \left(1 + 2\Delta t\nu\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)\right) u_{i,j}^{n+1} = u_{i,j}^{n} &+ \Delta t\left[-u_{i,j}^{n}\frac{u_{i,j}^{n} - u_{i,j-1}^{n}}{\Delta x} - v_{i,j}^{n}\frac{u_{i,j}^{n} - u_{i-1,j}^{n}}{\Delta y}\right] \\ &+ \Delta t\nu\left(\frac{u_{i+1,j}^{n+1} + u_{i-1,j}^{n+1}}{\Delta x^2} + \frac{u_{i,j+1}^{n+1} + u_{i,j-1}^{n+1}}{\Delta y^2}\right) \end{aligned} \tag{22}$$

The final form of the semi-implicit update is:

$$u_{i,j}^{n+1} = \frac{u_{i,j}^n + \Delta t \left[-u_{i,j}^n \cdot \frac{u_{i,j}^n - u_{i,j-1}^n}{\Delta x} - v_{i,j}^n \cdot \frac{u_{i,j}^n - u_{i-1,j}^n}{\Delta y} \right] + \Delta t \cdot \nu \left(\frac{u_{i+1,j}^{n+1} + u_{i-1,j}^{n+1}}{\Delta x^2} + \frac{u_{i,j+1}^{n+1} + u_{i,j-1}^{n+1}}{\Delta y^2} \right)}{1 + 2\Delta t \nu \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)} \tag{23}$$

A.2 Modified Equations Derivation

Taylor expansions are used to derive the modified equations of the momentum equations and pressure-Poisson equation.
We begin with the u-momentum equation:

$$\begin{aligned} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} + u_{i,j}^n \left(\frac{u_{i,j}^n - u_{i-1,j}^n}{\Delta x} \right) + v_{i,j}^n \left(\frac{u_{i,j}^n - u_{i,j-1}^n}{\Delta y} \right) \\ = -\frac{1}{\rho} \left(\frac{p_{i+1,j}^n - p_{i-1,j}^n}{2\Delta x} \right) \\ + \nu \left(\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{\Delta x^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{\Delta y^2} \right) \end{aligned} \tag{24}$$

To derive the modified equation, the Taylor expansions for each relevant term are shown below.

$$\begin{aligned} u_{i,j}^{n+1} &= u_{i,j}^n + \Delta t \frac{\partial u}{\partial t} + \frac{\Delta t^2}{2!} \frac{\partial^2 u}{\partial t^2} + \frac{\Delta t^3}{3!} \frac{\partial^3 u}{\partial t^3} + \cdots \\ &= u_{i,j}^n + \Delta t \frac{\partial u}{\partial t} + \frac{\Delta t^2}{2} \frac{\partial^2 u}{\partial t^2} + \frac{\Delta t^3}{6} \frac{\partial^3 u}{\partial t^3} + \cdots \end{aligned} \tag{25}$$

$$u_{i+1,j}^n = u_{i,j}^n + \Delta x \frac{\partial u}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial^2 u}{\partial x^2} + \frac{\Delta x^3}{6} \frac{\partial^3 u}{\partial x^3} + \cdots \tag{26}$$

$$u_{i-1,j}^n = u_{i,j}^n - \Delta x \frac{\partial u}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial^2 u}{\partial x^2} - \frac{\Delta x^3}{6} \frac{\partial^3 u}{\partial x^3} + \cdots \tag{27}$$

$$u_{i,j+1}^n = u_{i,j}^n + \Delta y \frac{\partial u}{\partial y} + \frac{\Delta y^2}{2} \frac{\partial^2 u}{\partial y^2} + \frac{\Delta y^3}{6} \frac{\partial^3 u}{\partial y^3} + \cdots \tag{28}$$

$$u_{i,j-1}^n = u_{i,j}^n - \Delta y \frac{\partial u}{\partial y} + \frac{\Delta y^2}{2} \frac{\partial^2 u}{\partial y^2} - \frac{\Delta y^3}{6} \frac{\partial^3 u}{\partial y^3} + \cdots \tag{29}$$

$$p_{i+1,j}^n = p_{i,j}^n + \Delta x \frac{\partial p}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial^2 p}{\partial x^2} + \frac{\Delta x^3}{6} \frac{\partial^3 p}{\partial x^3} + \cdots \tag{30}$$

$$p_{i-1,j}^n = p_{i,j}^n - \Delta x \frac{\partial p}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial^2 p}{\partial x^2} - \frac{\Delta x^3}{6} \frac{\partial^3 p}{\partial x^3} + \cdots \tag{31}$$

We next plug these Taylor expansions into the u-momentum equation one by one, as shown below.

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \frac{\partial u}{\partial t} + \frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2} + \frac{\Delta t^2}{6}\frac{\partial^3 u}{\partial t^3} + \cdots \tag{32}$$

$$u_{i,j}^{n}\left(\frac{u_{i,j}^{n} - u_{i-1,j}^{n}}{\Delta x}\right) = u_{i,j}^{n}\left(\frac{\partial u}{\partial x} - \frac{\Delta x}{2}\frac{\partial^2 u}{\partial x^2} + \frac{\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3} + \cdots\right) \tag{33}$$

$$v_{i,j}^{n}\left(\frac{u_{i,j}^{n} - u_{i,j-1}^{n}}{\Delta y}\right) = v_{i,j}^{n}\left(\frac{\partial u}{\partial y} - \frac{\Delta y}{2}\frac{\partial^2 u}{\partial y^2} + \frac{\Delta y^2}{6}\frac{\partial^3 u}{\partial y^3} + \cdots\right) \tag{34}$$

$$\frac{1}{\rho}\left(\frac{p_{i+1,j}^{n} - p_{i-1,j}^{n}}{2\Delta x}\right) = \frac{1}{\rho}\left(\frac{\partial p}{\partial x} + \frac{\Delta x^2}{6}\frac{\partial^3 p}{\partial x^3} + \cdots\right) \tag{35}$$

$$\nu\left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\Delta x^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta y^2}\right) = \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{36}$$

We can now combine everything to obtain the full u-momentum equation.

$$\begin{aligned}\frac{\partial u}{\partial t} + \frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2} + \frac{\Delta t^2}{6}\frac{\partial^3 u}{\partial t^3} + u_{i,j}^{n}\left(\frac{\partial u}{\partial x} - \frac{\Delta x}{2}\frac{\partial^2 u}{\partial x^2} + \frac{\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3} + \cdots\right)& \\ + v_{i,j}^{n}\left(\frac{\partial u}{\partial y} - \frac{\Delta y}{2}\frac{\partial^2 u}{\partial y^2} + \frac{\Delta y^2}{6}\frac{\partial^3 u}{\partial y^3} + \cdots\right)& \\ = -\frac{1}{\rho}\left(\frac{\partial p}{\partial x} + \frac{\Delta x^2}{6}\frac{\partial^3 p}{\partial x^3} + \cdots\right)& \\ + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)&\end{aligned} \tag{37}$$

Rearranging this, we obtain the modified equation, as shown below.

$$\begin{aligned}\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + \frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2} + \frac{\Delta t^2}{6}\frac{\partial^3 u}{\partial t^3}& \\ + u\left(-\frac{\Delta x}{2}\frac{\partial^2 u}{\partial x^2} + \frac{\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3}\right) + v\left(-\frac{\Delta y}{2}\frac{\partial^2 u}{\partial y^2} + \frac{\Delta y^2}{6}\frac{\partial^3 u}{\partial y^3}\right) - \frac{\Delta x^2}{6\rho}\frac{\partial^3 p}{\partial x^3}&\end{aligned} \tag{38}$$

An identical derivation can be completed for the v-momentum equation to obtain the following modified equation:

$$\begin{aligned}\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + \frac{\Delta t}{2}\frac{\partial^2 v}{\partial t^2} + \frac{\Delta t^2}{6}\frac{\partial^3 v}{\partial t^3}& \\ + u\left(-\frac{\Delta x}{2}\frac{\partial^2 v}{\partial x^2} + \frac{\Delta x^2}{6}\frac{\partial^3 v}{\partial x^3}\right) + v\left(-\frac{\Delta y}{2}\frac{\partial^2 v}{\partial y^2} + \frac{\Delta y^2}{6}\frac{\partial^3 v}{\partial y^3}\right) - \frac{\Delta y^2}{6\rho}\frac{\partial^3 p}{\partial y^3}&\end{aligned} \tag{39}$$

A similar process can be completed for the discretized pressure-Poisson equation:

$$\frac{p^n_{i+1,j} - 2p^n_{i,j} + p^n_{i-1,j}}{\Delta x^2} + \frac{p^n_{i,j+1} - 2p^n_{i,j} + p^n_{i,j-1}}{\Delta y^2} = \rho \left[\frac{1}{\Delta t} \left(\frac{u_{i,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j} - v_{i,j-1}}{2\Delta y} \right) - \left(\frac{u_{i,j} - u_{i-1,j}}{2\Delta x} \right) \left(\frac{u_{i,j} - u_{i-1,j}}{2\Delta x} \right) - \left(\frac{v_{i,j} - v_{i,j-1}}{2\Delta y} \right) \left(\frac{v_{i,j} - v_{i,j-1}}{2\Delta y} \right) - 2 \left(\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} \right) \left(\frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x} \right) \right] \tag{40}$$

Taylor expansions are needed for each relevant term, which can be obtained in a similar fashion to the expansions from the u-momentum equation. We plug them into the equation term by term as shown below.

$$\frac{p^n_{i+1,j} - 2p^n_{i,j} + p^n_{i-1,j}}{\Delta x^2} = \frac{\partial^2 p}{\partial x^2} \tag{41}$$

$$\frac{p^n_{i,j+1} - 2p^n_{i,j} + p^n_{i,j-1}}{\Delta y^2} = \frac{\partial^2 p}{\partial y^2} \tag{42}$$

$$\frac{1}{\Delta t} \left(\frac{u_{i,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j} - v_{i,j-1}}{2\Delta y} \right) = \frac{1}{\Delta t} \left(\frac{\partial u}{\partial x} + \frac{\Delta x^2}{6} \frac{\partial^3 u}{\partial x^3} + \frac{\partial v}{\partial y} + \frac{\Delta y^2}{6} \frac{\partial^3 v}{\partial y^3} \right) \tag{43}$$

$$\left(\frac{u_{i,j} - u_{i-1,j}}{2\Delta x} \right) \left(\frac{u_{i,j} - u_{i-1,j}}{2\Delta x} \right) = \left(\frac{\partial u}{\partial x} + \frac{\Delta x^2}{6} \frac{\partial^3 u}{\partial x^3} \right) \left(\frac{\partial u}{\partial x} + \frac{\Delta x^2}{6} \frac{\partial^3 u}{\partial x^3} \right) \tag{44}$$

$$2 \left(\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} \right) \left(\frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x} \right) = 2 \left(\frac{\partial u}{\partial y} + \frac{\Delta y^2}{6} \frac{\partial^3 u}{\partial y^3} \right) \left(\frac{\partial v}{\partial x} + \frac{\Delta x^2}{6} \frac{\partial^3 v}{\partial x^3} \right) \tag{45}$$

$$\left(\frac{v_{i,j} - v_{i,j-1}}{2\Delta y} \right) \left(\frac{v_{i,j} - v_{i,j-1}}{2\Delta y} \right) = \left(\frac{\partial v}{\partial y} + \frac{\Delta y^2}{6} \frac{\partial^3 v}{\partial y^3} \right) \left(\frac{\partial v}{\partial y} + \frac{\Delta y^2}{6} \frac{\partial^3 v}{\partial y^3} \right) \tag{46}$$

Next, we combine everything to obtain the full pressure-Poisson equation.

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = \rho \left[\frac{1}{\Delta t} \left(\frac{\partial u}{\partial x} + \frac{\Delta x^2}{6} \frac{\partial^3 u}{\partial x^3} + \frac{\partial v}{\partial y} + \frac{\Delta y^2}{6} \frac{\partial^3 v}{\partial y^3} \right) - \left(\frac{\partial u}{\partial x} + \frac{\Delta x^2}{6} \frac{\partial^3 u}{\partial x^3} \right) \left(\frac{\partial u}{\partial x} + \frac{\Delta x^2}{6} \frac{\partial^3 u}{\partial x^3} \right) - 2 \left(\frac{\partial u}{\partial y} + \frac{\Delta y^2}{6} \frac{\partial^3 u}{\partial y^3} \right) \left(\frac{\partial v}{\partial x} + \frac{\Delta x^2}{6} \frac{\partial^3 v}{\partial x^3} \right) - \left(\frac{\partial v}{\partial y} + \frac{\Delta y^2}{6} \frac{\partial^3 v}{\partial y^3} \right) \left(\frac{\partial v}{\partial y} + \frac{\Delta y^2}{6} \frac{\partial^3 v}{\partial y^3} \right) \right] \tag{47}$$

Finally, we rearrange to obtain the modified pressure-Poisson equation.

$$\begin{aligned}\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2} = &-\rho\left(\frac{\partial u}{\partial x}\frac{\partial u}{\partial x}+2\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}+\frac{\partial v}{\partial y}\frac{\partial v}{\partial y}\right)\\ &-\rho\Bigg[-\frac{1}{\Delta t}\left(\frac{\partial u}{\partial x}+\frac{\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3}+\frac{\partial v}{\partial y}+\frac{\Delta y^2}{6}\frac{\partial^3 v}{\partial y^3}\right)\\ &+\frac{\Delta x^2}{3}\frac{\partial u}{\partial x}\frac{\partial^3 u}{\partial x^3}+\frac{\Delta x^4}{36}\left(\frac{\partial^3 u}{\partial x^3}\right)^2\\ &+2\left(\frac{\Delta x^2}{6}\frac{\partial u}{\partial y}\frac{\partial^3 v}{\partial x^3}+\frac{\Delta y^2}{6}\frac{\partial v}{\partial x}\frac{\partial^3 u}{\partial y^3}+\frac{\Delta x^2\Delta y^2}{36}\frac{\partial^3 u}{\partial y^3}\frac{\partial^3 v}{\partial x^3}\right)\\ &+\frac{\Delta y^2}{3}\frac{\partial v}{\partial y}\frac{\partial^3 u}{\partial y^3}+\frac{\Delta y^4}{36}\left(\frac{\partial^3 v}{\partial y^3}\right)^2\Bigg]\end{aligned} \tag{48}$$