

Universitat Politècnica de Catalunya
ESCOLA SUPERIOR D'ENGINYERIES INDUSTRIAL, AEROESPACIAL I
AUDIOVISUAL DE TERRASSA

ASSIGNMENT 1:
NUMERICAL SOLUTION OF THE
INCOMPRESSIBLE NAVIER-STOKES
EQUATIONS

Aerodinàmica, Mecànica de Vol i Orbital
2024-25 Q2

06/04/2025

GROUP 12

Lluís Ceinos, Marc
Gutiérrez Portillo, Guillem
Poposki Stanojkovska, Vedran



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

1 Introduction to the Problem

The objective of this problem is to determine the velocity and pressure fields as functions of spatial coordinates (x, y) and time (t) that satisfy the incompressible Navier-Stokes equations for arbitrary values of density (ρ) and kinematic viscosity (ν) . The problem is divided into the following parts:

- **Part A:** Implementation and verification of the convective and diffusive terms. An analytic solution will be used and the goal is to show second order convergence.
- **Part B:** Implementation and verification of the pressure-velocity coupling. An arbitrary velocity field with non-null divergence is employed instead of u^p .
- **Part C:** Implementation and verification of time integration.

1.a Preliminary Considerations

1. **Staggered mesh:** The spatial discretization follows a staggered grid approach to avoid numerical instabilities and ensure proper representation of velocity and pressure fields.
2. **Mesh for a periodic problem and halo update:** The domain is periodic, requiring careful treatment of boundary conditions and halo cells for data exchange between neighboring cells.

2 Part A

This first part of the problem is intended to compute and verify the numerical computation of convective and diffusive terms using a known analytical solution. The goal is to demonstrate second-order convergence by computing numerical errors at different grid resolutions and checking their convergence.

2.a Algorithm

Firstly, the initial condition is established a periodic velocity field as explained in the introduction. Then, the convective and diffusive terms for u and v are computed using symbolic differentiation.

The numerical computation is done for different grid resolutions ranging from 5 to 2560. The code loops over these grid sizes, repeating the following algorithm:

1. Addition of halo and calculation of cell size.
2. Computing numerical velocity field using set velocity field function with analytical u and v as inputs.
3. Computing numerical convective and diffusive terms using the dedicated functions and the calculated velocity field.
4. Compute analytical convective and diffusive fields on the grid

5. Compute the maximum error for each computed field convective u , convective v , diffusive u and diffusive v .
6. Store errors in preallocated storage arrays.
7. Plot errors.

Custom functions have been developed to compute, among other things, the convective and diffusive terms. The calculation of the convective term is done by firstly interpolating the velocities at the faces, then computing flow terms and the convective terms for each velocity direction. In the case of the diffusive term, it is first approximated using second-order finite differences (forward and backward).

It is also important to highlight the importance of performing a halo update after each domain-wide computation. This means that at the halo update function is called at the end of each of the other implemented functions, updating the values of the cells at the edges of the domain.

2.b Validation strategy

The basis for the validation of Part A is the comparison between numerical and analytical solutions for different grid sizes. The error between these two values is computed and plotted on a logarithmic scale, where it should appear with a slope of h^2 if it decreases quadratically, signifying second-order accuracy.

2.c Results

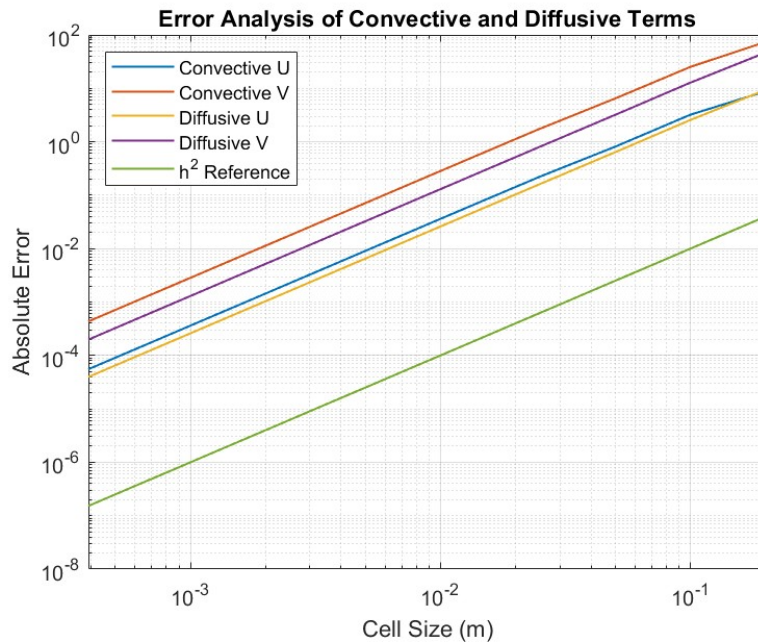


Figure 1: Error of computed terms

The error lines are parallel to the h^2 reference, confirming second-order accuracy.

3 Part B

This second part of the problem aims to implement the pressure-velocity coupling and show how an arbitrary velocity field, after subtracting the gradient of a certain type of field, can become a null-divergence one.

3.a Algorithm

The implemented algorithm follows a simplified version of a predictor-corrector scheme. The objective is to begin with a velocity field containing artificial divergence and apply a correction using a scalar field (pseudo-pressure) whose gradient will subtract the divergence component from the velocity.

1. Create coordinate positions, initialize null velocity field and introduce artificial divergence at a specific grid point by setting its horizontal and vertical velocities to non-zero values.
2. Compute the divergence of the predictor velocity field using a custom function.
3. Construct Laplacian matrix for pressure Poisson equation, solve a linear system for pseudo-pressure using the Laplacian equation and the divergence field in vector form.
4. Convert the pseudo-P vector back to 2D field format, then compute its gradients. Correct the velocity field using the calculated pseudo-P gradients. Then, compute the divergence of this new velocity field.
5. Compare maximum values of divergence before and after correction.
6. Plot the results.

A relevant remark about this code are that the divergence and gradient are calculated in the physical domain excluding the halo, which is calculated after. Both divergence and gradient are calculated using finite difference approximation, forward difference in the case of the gradient and backward for the divergence, ensuring consistency with the staggered-grid scheme.

3.b Validation strategy

The validation of the pressure-velocity coupling is based on a two-step comparison of the divergence field. First, the divergence of the artificially perturbed velocity field is computed and its maximum absolute value is recorded. This provides a quantitative measure of how far the initial field is from being divergence-free. The maximum absolute divergence is again measured after correcting the velocity field.

A successful implementation should yield a corrected velocity field whose divergence is close to zero everywhere. Specifically, the maximum divergence should drop by several orders of magnitude, confirming that the projection step has effectively enforced incompressibility.

3.c Results

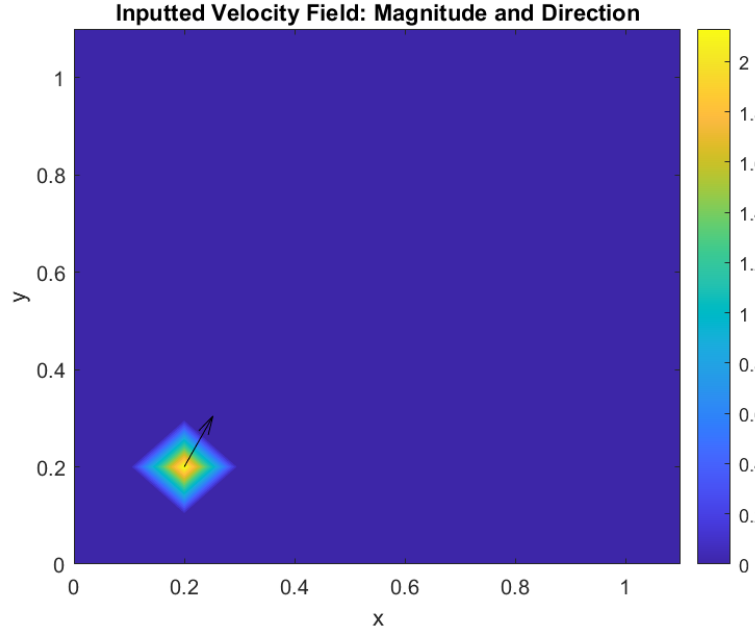


Figure 2: Input non-zero velocity shown in the physical domain.

The visualization is consistent with the injection of velocity in the null-velocity field.

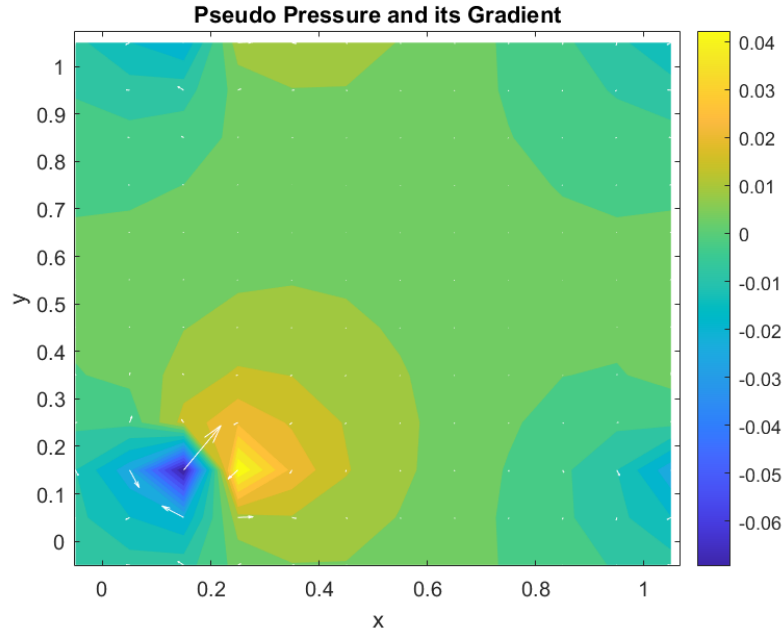


Figure 3: Pseudo-pressure in the physical domain.

The pressure gradient is actively correcting the divergence introduced by the velocity. The gradient vectors align with regions of high pseudo-P variation.

4 Part C

The objective of this final part is to simulate the temporal evolution of an incompressible, 2D fluid flow using a projection method. This approach separates the velocity update from pressure correction. The simulation is validated against an analytical solution, allowing for the assessment of accuracy and stability.

4.a Algorithm

The algorithm used follows a projection method, ensuring that the velocity field remains divergence-free at each step. The steps are explained below:

1. Initialize physical parameters including the time step and the analytical solution velocity and pressure fields. Calculate initial convective and diffusive terms, time step and R term.
2. Iterate for each timestep until the time limit is reached. For each time iteration:
 - Convective and diffusive terms are computed for both velocity components.
 - Calculate time step for each iteration.
 - Update R term and use it to find the predictor velocities.
 - Calculate divergence for predictor velocities.
 - A Poisson equation for the pseudo P field is solved using the Laplacian matrix.
 - Calculate pseudo P gradient and subtract it from predictor velocities to obtain corrected divergence-free predictor velocities.
 - Check divergence for predicted and corrected velocity fields. Find maximum values.
 - Update the iteration velocities with the corrected velocities and calculate the pressure difference field.
 - Compute analytical solution.
 - Store solutions and advance time step.
3. Calculate errors for each timestep and find maximums.
4. Visualize results. Plot errors for cell length, compare solutions at a point.

4.b Validation Strategy

At each time iteration, the maximum absolute error between the numerical and analytical solutions is computed over the physical domain for u , v , and p . The evolution of these errors over time shows the accuracy and consistency of the simulation.

The code generates error vs mesh size plots showing that the numerical error decreases with grid refinement. A second-order decay rate in space is expected.

Temporal evolution at a fixed point is compared between analytical and numerical values, confirming that the numerical solution accurately follows the behavior of the analytical one throughout the simulation.

Finally, particle trajectories are simulated through time and displayed in an animation together with velocities. This video is obtained through the code and an image of it is shown in Figure 7.

4.c Results

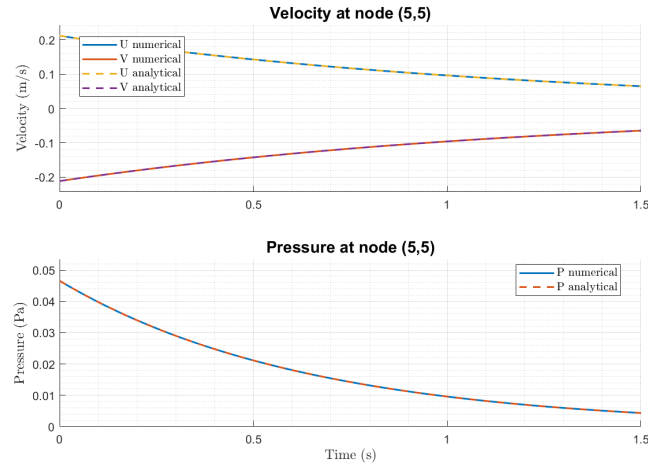


Figure 4: Comparison of numerical and analytical calculated values at a point.

Figure 4 confirms that the expectation of the numerical solution being a good approximation of the analytical one is correct.

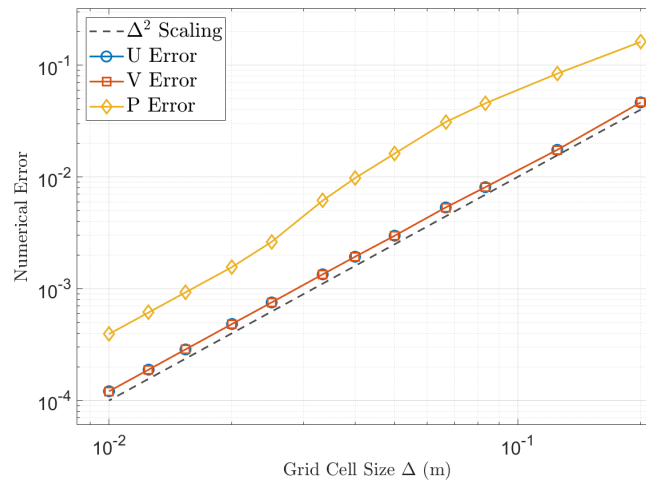


Figure 5: Convergence of error over the mesh.

Similarly to the previous parts of the code, the error for the numerical solution for decreasing grid cell size decreases quadratically, confirming second-order accuracy.

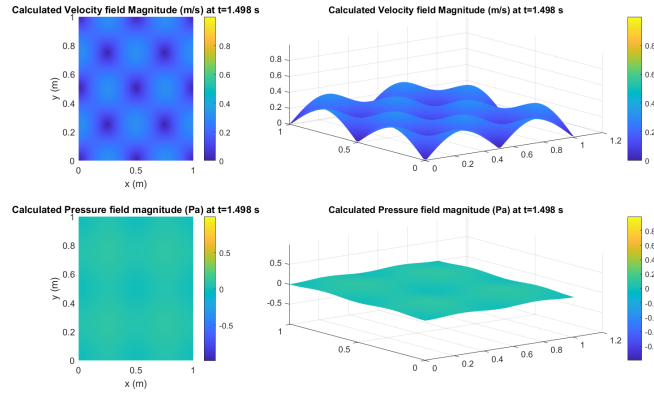


Figure 6: Calculated velocity and pressure fields at a point in time.

Shows a symmetric, periodic variation in the x-component of velocity across the domain, in line with the initial conditions.

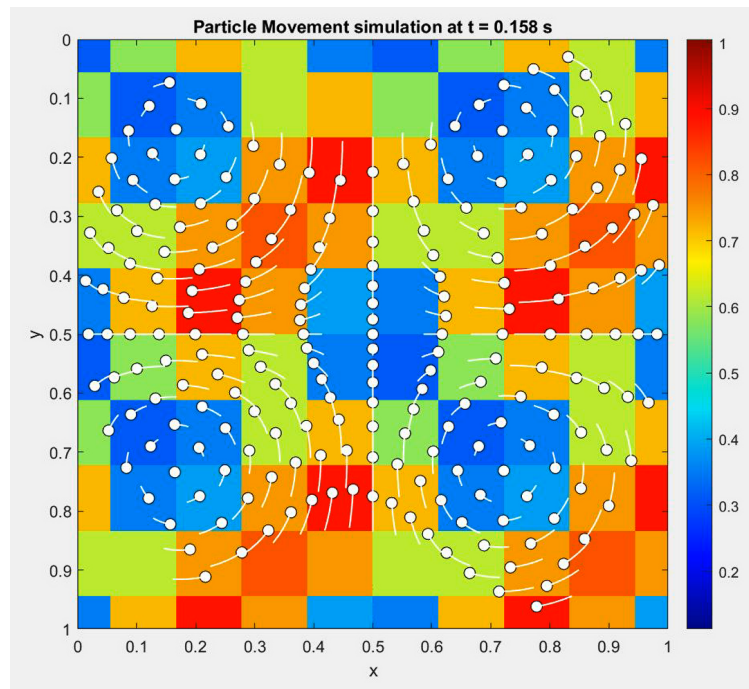


Figure 7: Screenshot of the particle movement simulation.

Depicts particle trajectories overlaid on a colored velocity grid, confirming coherent particle motion along the established vector field with evident symmetry and smooth flow paths.

5 Conclusions

The numerical resolution of the incompressible Navier-Stokes equations has been implemented and validated in three stages.

In the first part, a known analytical solution was used to validate the implementation of convective and diffusive terms. By computing numerical errors over a range of grid sizes, the error plot confirmed a second-order convergence rate. This validates the finite-difference discretization schemes used for both convective and diffusive terms.

The second part focused on enforcing incompressibility using a projection method. Starting from a velocity field with artificial divergence. The pseudo-pressure gradients aligned correctly with the divergence sources, and the visualizations confirmed that the velocity field became divergence-free after correction.

The final part combined the previous components into a full time-dependent simulation. The temporal evolution was validated against the analytical solution, showing a close match to the analytical one in both velocity and pressure over time, and the convergence study confirmed second-order spatial accuracy. Divergence remained low throughout the simulation, confirming the effectiveness of the projection step. Additionally, particle trajectory animations showed smooth motion consistent with the velocity field.

Overall, the solver performs accurately and reliably across all tested aspects: It achieves second-order convergence in space, it correctly enforces the divergence-free condition through a pressure correction step and it reproduces time-dependent flow behavior with accuracy compared to analytical references.