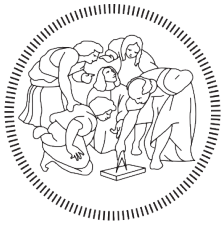

POLITECNICO
MILANO 1863

High Performance Computing Engineering

Solving the Navier-Stokes Equations: a Finite Element Solver using Deal.ii

Numerical Methods for Partial Differential Equations

Authors:

Gotti Mattia	10766863
Milani Michele	10776157

Professor: Alfio Quarteroni
Assistant Professor: Michele Bucelli
Academic Year: 2025-2026

Contents

Introduction

This project aims to numerically solve the unsteady, incompressible Navier-Stokes equations using the finite element method to simulate the benchmark problem "flow past a cylinder" in two and three dimensions, analyzing the aerodynamic properties of the obstacle across different Reynolds numbers through the computation of lift and drag coefficients. The report will discuss the employed numerical methods, their stability and accuracy, and computational efficiency. Furthermore, we explore preconditioning strategies tailored for high-performance computing applications.

1 The Navier-Stokes Equations

The motion of an unsteady, incompressible, viscous fluid is governed by the Navier-Stokes equations. Given a spatial domain $\Omega \subset \mathbb{R}^d$ (with $d = 2, 3$) and a time interval $(0, T]$, the system of equations for the velocity field $\mathbf{u}(\mathbf{x}, t)$ and the pressure field $p(\mathbf{x}, t)$ is given by:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega, t > 0 \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega, t > 0 \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma_D \subset \partial\Omega, t > 0 \\ \nu \nabla \mathbf{u} \cdot \mathbf{n} - p\mathbf{n} = \mathbf{h} & \text{on } \Gamma_N = \partial\Omega \setminus \Gamma_D, t > 0 \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0 & \text{in } \Omega, t = 0. \end{cases} \tag{1}$$

In this formulation, $\nu > 0$ denotes the kinematic viscosity of the fluid, while $\mathbf{f}$ represents the forcing term. The first equation expresses the conservation of momentum, including the non-linear convective term $(\mathbf{u} \cdot \nabla)\mathbf{u}$, while the second equation represents the continuity equation, enforcing the incompressibility constraint.

The boundary $\partial\Omega$ is partitioned into two disjoint subsets, Γ_D and Γ_N ($\Gamma_D \cup \Gamma_N = \partial\Omega$ and $\Gamma_D \cap \Gamma_N = \emptyset$), where Dirichlet and Neumann boundary conditions are applied via the functions $\mathbf{g}$ and $\mathbf{h}$, respectively. Here, $\mathbf{n}$ denotes the outward-pointing unit normal vector to the boundary Γ_N. Finally, $\mathbf{u}_0$ defines the initial velocity field at $t = 0$, which must satisfy the compatibility condition $\nabla \cdot \mathbf{u}_0 = 0$.

2 Mathematical model

To solve the Navier-Stokes equations numerically, we employ a *Finite Element Method* (FEM) for spatial discretization combined with a first-order Backward Differentiation Formula (Implicit Euler) for time stepping.

2.1 Weak formulation

We introduce the functional spaces for velocity and pressure. Let $H^1_D(\Omega) = \{v \in H^1(\Omega) \mid v = 0 \text{ on } \Gamma_D\}$, and let V denote the space $[H^1_D(\Omega)]^d$ and Q the space $L^2(\Omega)$.

To handle the non-homogeneous Dirichlet boundary condition $\mathbf{u} = \mathbf{g}$ on Γ_D, we employ a lifting technique. We define a lifting function $\mathbf{u}_g \in [H^1(\Omega)]^d$ such that its trace on the Dirichlet boundary matches the data, i.e., $\gamma_{\Gamma_D}(\mathbf{u}_g) = \mathbf{g}$. The total velocity field can then be decomposed as:

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_0(\mathbf{x}, t) + \mathbf{u}_g(\mathbf{x}, t) \tag{2}$$

where $\mathbf{u}_0 \in V$ is the unknown part of the velocity with zero trace on Γ_D.

Multiplying the momentum equation by a test function $\mathbf{v} \in V$ and the continuity equation by $q \in Q$, and integrating by parts the viscous and pressure terms, we obtain the following **weak formulation**:

$\forall t > 0$, find $\mathbf{u} \in \{\mathbf{u}_g + V\}, p \in Q$ such that:

$$\left(\frac{\partial \mathbf{u}}{\partial t}, \mathbf{v}\right) + a(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = F(\mathbf{v}) \quad \forall \mathbf{v} \in V, \tag{3}$$

$$b(\mathbf{u}, q) = 0 \quad \forall q \in Q, \tag{4}$$

where the bilinear and trilinear forms are defined as:

- $a(\mathbf{u}, \mathbf{v}) = \int_\Omega \nu \nabla \mathbf{u} : \nabla \mathbf{v} \, d\Omega$
- $c(\mathbf{w}, \mathbf{u}, \mathbf{v}) = \int_\Omega ((\mathbf{w} \cdot \nabla)\mathbf{u}) \cdot \mathbf{v} \, d\Omega$
- $b(\mathbf{v}, q) = -\int_\Omega q \nabla \cdot \mathbf{v} \, d\Omega$

The linear functional $F(\mathbf{v})$ accounts for external forces $\mathbf{f}$ and the Neumann boundary conditions $\mathbf{h}$ on Γ_N:

$$F(\mathbf{v}) = \int_\Omega \mathbf{f} \cdot \mathbf{v} \, d\Omega + \int_{\Gamma_N} \mathbf{h} \cdot \mathbf{v} \, d\gamma \tag{5}$$

2.1.1 Well-posedness and Stability

The well-posedness of the Navier-Stokes equations depends on the Reynolds number and the domain's properties. Existence and uniqueness are guaranteed for steady flows at low Re, while in 3D unsteady cases, global regularity remains an open challenge. Stability is ensured by the *coercivity* of the diffusion term and the divergence-free constraint on the convective term. In this benchmark, the pressure field is uniquely determined as the Neumann boundary condition at the outflow ($\Gamma_N \neq \emptyset$) removes the typical additive constant indeterminacy.

At the discrete level, stability requires the fulfillment of the *inf-sup* (or LBB) condition. This condition ensures compatibility between the velocity space V_h and the pressure space Q_h, preventing spurious pressure oscillations and ensuring a well-posed algebraic system. This requirement dictates the choice of stable finite element pairs, such as the Taylor-Hood elements used in this work.

2.2 Semi-discrete problem

To transition from the continuous weak formulation to a computationally tractable problem, we introduce a spatial discretization based on the Finite Element Method. In order to reduce the infinite degrees of freedom of the continuous problem, we introduce the space of finite elements:

$$[X_h^r]^d = \{v_h \in [C^0(\overline{\Omega})]^d : v_h|_{K_i} \in [\mathcal{P}^r]^d, \forall K_i \in \mathcal{T}_h\} \tag{6}$$

where $\mathcal{T}_h$ is a triangulation composed of simplicial elements (triangles in 2D). Following the Taylor-Hood requirement for the Inf-Sup stability, we choose the discrete spaces for velocity and pressure as $V_h = [X_h^2]^2 \cap V$ and $Q_h = X_h^1 \cap Q$, respectively (i.e., $\mathbb{P}_2 - \mathbb{P}_1$ elements).

The semi-discrete form obtained through the Galerkin approximation reads as follows: $\forall t > 0$, find $\mathbf{u}_h \in V_h, p_h \in Q_h$:

$$(\partial_t \mathbf{u}_h, \mathbf{v}_h) + a(\mathbf{u}_h, \mathbf{v}_h) + c(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = F(\mathbf{v}_h) \quad \forall \mathbf{v}_h \in V_h \tag{7}$$

$$b(\mathbf{u}_h, q_h) = 0 \quad \forall q_h \in Q_h \tag{8}$$

with $\mathbf{u}_h(\mathbf{x}, 0) = \mathbf{u}_{h,0}(\mathbf{x})$. Denoting by $\{\boldsymbol{\varphi}_j\}_{j=1}^{N_h}$ the finite element basis of V_h and by $\{\psi_k\}_{k=1}^{M_h}$ the basis of Q_h, the discrete solution is:

$$\mathbf{u}_h(\mathbf{x}, t) = \sum_{j=1}^{N_h} u_j(t)\boldsymbol{\varphi}_j(\mathbf{x}), \quad p_h(\mathbf{x}, t) = \sum_{k=1}^{M_h} p_k(t)\psi_k(\mathbf{x}) \tag{9}$$

2.3 Fully-discrete problem

The semi-discrete formulation is discretized in time using a semi-implicit method for the nonlinear convective term, i.e., $(\mathbf{u} \cdot \nabla)\mathbf{u} \approx (\mathbf{u}^k \cdot \nabla)\mathbf{u}^{k+1}$. Let $\mathbf{u}_h^k \in V_h$ and $p_h^k \in Q_h$ be approximations of the solutions $\mathbf{u}$ and p at time t^k. We approximate the solution at time t^{k+1} using the Backward Differentiation Formula of order 1 (BDF1), also known as the Implicit Euler method, which offers first-order accuracy in time.

The semi-implicit BDF1 scheme reads: $\forall k \geq 0$, find $\mathbf{u}_h^{k+1} \in V_h, p_h^{k+1} \in Q_h$ such that:

$$\int_\Omega \frac{\mathbf{u}_h^{k+1} - \mathbf{u}_h^k}{\Delta t} \cdot \boldsymbol{\varphi}_i \, d\Omega + a(\mathbf{u}_h^{k+1}, \boldsymbol{\varphi}_i) + c(\mathbf{u}_h^k, \mathbf{u}_h^{k+1}, \boldsymbol{\varphi}_i) + b(\boldsymbol{\varphi}_i, p_h^{k+1}) = F^{k+1}(\boldsymbol{\varphi}_i) \tag{10}$$

$$b(\mathbf{u}_h^{k+1}, \psi_m) = 0 \tag{11}$$

where the convective velocity is taken from the previous time step $\mathbf{u}_h^k$ to linearize the problem. This semi-implicit treatment allows us to solve a single linear system at each time step, avoiding the higher computational costs of fully implicit methods or iterative Newton schemes.

2.4 Algebraic problem

Exploiting the finite element bases, the problem is equivalent to a linear system of the form:

$$\begin{bmatrix} \frac{1}{\Delta t}M + \mathcal{A} + \mathcal{C}[\mathbf{u}^k] & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{k+1} \\ \mathbf{p}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}^{k+1} + \frac{1}{\Delta t}M\mathbf{u}^k \\ 0 \end{bmatrix} \tag{12}$$

where:

- $M = (m_{ij}) = \int_\Omega \boldsymbol{\varphi}_j \cdot \boldsymbol{\varphi}_i \, d\Omega$ (Mass matrix)
- $\mathcal{A} = (a_{ij}) = \int_\Omega \nu \nabla \boldsymbol{\varphi}_j \cdot \nabla \boldsymbol{\varphi}_i \, d\Omega$ (Stiffness matrix)
- $\mathcal{C}[\mathbf{u}^k] = (c_{ij}[\mathbf{u}_h^k]) = \int_\Omega (\mathbf{u}_h^k \cdot \nabla)\boldsymbol{\varphi}_j \cdot \boldsymbol{\varphi}_i \, d\Omega$ (Convection matrix)
- $B = (b_{mj}) = \int_\Omega q_h \nabla \cdot \boldsymbol{\varphi}_j \, d\Omega$ (Divergence matrix)

3 C++ Implementation

The numerical solver is implemented in C++ using the `deal.II`[1] finite element library. The code is designed to be general for both 2D and 3D simulations by leveraging C++ templates on the geometric dimension d.

3.1 Parameter Management

To ensure reproducibility and flexibility without the need for recompilation, the solver integrates the `ParameterHandler` class provided by `deal.II`. All physical and numerical parameters are defined in an external configuration file (`.prm`).

- **Physical Properties**: Defines the Reynolds number, kinematic viscosity ν, fluid density ρ, and the specific test case (e.g., Turek 2D-1 or 2D-2).
- **Numerical Parameters**: Sets the polynomial degrees for velocity (r_u) and pressure (r_p), the time step Δt, and the total simulation time T.
- **Solver Control**: Allows the selection of the preconditioning strategy (Yosida or SIMPLE) and sets convergence tolerances for the GMRES linear solver.

3.2 Class Structure and Parallelism

The core logic is encapsulated within the `NavierStokes` class, which manages the simulation lifecycle from mesh initialization to data output. To address large-scale problems, the implementation leverages distributed memory parallelism via MPI and utilizes the `TrilinosWrappers` library for managing distributed sparse matrices and vectors. The mesh is partitioned using a distributed triangulation, and the degrees of freedom (DoFs) are renumbered component-wise. This reordering ensures that all velocity DoFs precede the pressure DoFs, creating a naturally blocked algebraic structure essential for the application of efficient block-preconditioning.

3.3 Finite Element Setup

The simulation is initialized by the `setup()` method, which reads the mesh from a `.msh` file, partitions it among MPI processes, and distributes the DoFs using the `FE_SimplexP` elements. Following the Taylor-Hood requirement for stability, we employ quadratic elements for velocity and linear elements for pressure.

3.4 Assembly of the System

To optimize the time-stepping loop, the assembly process is split into two distinct functions to avoid redundant computations:

- **Initial Assembly**: The function `assemble()` computes the time-independent operators: the mass matrix M, the stiffness matrix $\mathcal{A}$, and the pressure coupling blocks B and B^T. These are stored to build the base of the global system matrix.
- **Time-Step Assembly**: Since we use a semi-implicit BDF1 scheme, only the convection matrix $\mathcal{C}[\mathbf{u}^k]$ and the right-hand side (RHS) need to be updated at each iteration. The function `assemble_time_step()` updates the global system matrix incrementally by subtracting the convective contribution from the previous step and adding the newly computed one:

$$\mathbf{F}^{k+1} = \mathbf{F}_{base} + \mathcal{C}[\mathbf{u}^k]$$

where $\mathbf{F}_{base} = \frac{1}{\Delta t} M + \mathcal{A} + B + B^T$.

3.5 Linear Solver and Preconditioning

The resulting saddle-point problem is solved using the GMRES algorithm. Given the ill-conditioning of the Navier-Stokes system, we implemented two block-preconditioning strategies. The Yosida preconditioner provides an algebraic approximation of the Schur complement, while the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) preconditioner acts as an effective fractional-step solver within the Krylov subspace iteration. Both strategies are designed to accelerate convergence by exploiting the block-partitioned structure of the linear system.

4 Preconditioners

Here we discuss the preconditioners implemented and their role in High Performance Computing (HPC). They are both taken from [2] and they are made ad-hoc for Navier Stokes equations on large scale parallel architectures.

4.1 SIMPLE Preconditioner

The Semi-Implicit Method for Pressure Linked Equations (SIMPLE) preconditioner is defined by the following block factorization:

$$P_{SIMPLE} = \begin{bmatrix} F & 0 \\ B & -\tilde{S} \end{bmatrix} \begin{bmatrix} I & D^{-1}B^T \\ 0 & \alpha I \end{bmatrix} \tag{13}$$

where F is the momentum matrix, D is the diagonal of F, and $\tilde{S} = BD^{-1}B^T$ represents the approximated Schur complement. A relaxation parameter $\alpha \in (0, 1]$ is introduced to control the pressure update; in our setup, this is conservatively set to $\alpha = 0.5$.

Relying on `deal.II` and `TrilinosWrappers`, the application of the preconditioner is algorithmically decoupled into two main stages: a prediction step and a subsequent correction step.

During the **prediction step**, the block lower triangular system is solved to find an intermediate solution $(\tilde{u}, \tilde{p})$. First, the intermediate velocity $\tilde{u}$ is obtained by solving the linear system $F\tilde{u} = r_u$. Because F embeds convective terms and is non-symmetric, this subsystem is solved using the Generalized Minimal Residual (GMRES) method. Next, the intermediate pressure $\tilde{p}$ is computed by solving the approximate Schur complement system $-\tilde{S}\tilde{p} = r_p - B\tilde{u}$. The operator $-\tilde{S}$ is explicitly assembled via matrix-matrix multiplication during the initialization phase. Since it is symmetric, the associated linear system is solved using the Conjugate Gradient (CG) method. To ensure high computational efficiency, both inner Krylov solvers (GMRES and CG) are preconditioned with an Incomplete LU (ILU) factorization and operate with a relative tolerance of 10^{-2}.

In the **correction step**, the block upper triangular system is applied to update the intermediate vectors and yield the final preconditioned solution (u, p). The pressure is scaled by the relaxation parameter:

$$p = \frac{1}{\alpha}\tilde{p} \tag{14}$$

Finally, the velocity field is updated using the pre-computed inverse diagonal matrix D^{-1} to account for the pressure correction:

$$u = \tilde{u} - D^{-1}B^T p \tag{15}$$

4.2 Yosida Preconditioner

The Yosida preconditioner is an alternative splitting approach that structurally resembles the SIMPLE preconditioner but differs in its formulation of the approximate Schur complement and the velocity correction step.

In the Yosida formulation, the exact Schur complement is approximated by replacing the inverse of the momentum matrix (F^{-1}) with the inverse of the mass matrix. Specifically, our implementation approximates the Schur complement as:

$$\tilde{S} = B[\text{diag}(M/\Delta t)]^{-1}B^T = \Delta t B[\text{diag}(M)]^{-1}B^T \tag{16}$$

During the initialization phase, the operator $-\tilde{S}$ is explicitly assembled via matrix-matrix multiplication. The inverse diagonal elements are extracted directly from the scaled mass matrix ($M/\Delta t$), making the assembly *highly efficient*.

Algorithmically, the application of the `PreconditionYosida` operator is divided into a prediction step, a pressure calculation step, and a final correction step. The required tolerances for the inner solvers are set to 10^{-2}, and Incomplete LU (ILU) factorizations are used to precondition all inner Krylov solvers.

1. **Velocity Prediction:** Similar to the SIMPLE approach, an intermediate velocity field y_u is computed by solving the momentum subsystem:

   $$Fy_u = r_u \tag{17}$$

   This non-symmetric system is solved using an ILU-preconditioned GMRES solver.

2. **Pressure Solve:** The pressure field y_p is then obtained by solving the approximate Schur complement system. The right-hand side is constructed using the intermediate velocity y_u:

   $$-\tilde{S}y_p = By_u - r_p \tag{18}$$

   Because $-\tilde{S}$ is symmetric and positive definite, this linear system is solved using the Conjugate Gradient (CG) method, again accelerated by an ILU preconditioner.

3. **Correction Step:** The final pressure update is applied directly without any relaxation parameter ($p = y_p$). However, the velocity correction in the Yosida scheme involves an additional exact inversion of the momentum matrix. A second linear system is solved to compute the velocity increment:

   $$F \cdot \text{res} = B^T y_p \tag{19}$$

   This requires a second ILU-preconditioned GMRES solve. Once the residual vector res is obtained, the final velocity field is updated as:

   $$u = y_u - \text{res} \tag{20}$$

From a computational perspective, the Yosida preconditioner requires two separate GMRES solves for the F matrix per application, compared to the single solve required by the SIMPLE preconditioner. While this makes Yosida potentially more expensive per iteration, the exact inversion in the correction step can lead to enhanced robustness and a reduction in the total number of outer Krylov iterations.

5 Results: Flow Past a Cylinder in 2D and 3D

In this section, we present the numerical results for the "flow past a cylinder" benchmark. The primary objective is to validate the C++ implementation by comparing the computed drag (C_d) and lift (C_l) coefficients against the established reference values provided by Turek and Schäfer [3]. The analysis covers both steady and unsteady regimes in 2D and 3D, evaluating the solver's ability to capture complex phenomena such as Von Kármán vortex shedding.

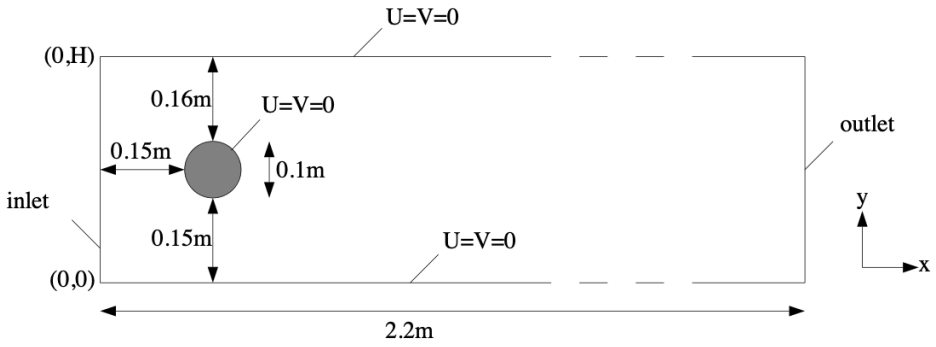

Figure 1: Geometry of 2D test cases with boundary conditions.

5.1 2D Test Cases

The 2D simulations are conducted on a rectangular domain containing a circular obstacle. The fluid properties and boundary conditions are varied to explore different Reynolds numbers (Re), defined based on the cylinder diameter $D = 0.1$ and the mean velocity $U(0, y)$.

5.1.1 Test Case 2D-1 (Steady)

The inflow condition is given by:

$$U(0, y) = \frac{4U_m y(H - y)}{H^2}, \quad V = 0 \tag{21}$$

with $U_m = 0.3$ m/s, yielding the Reynolds number $Re = 20$. Velocity increases once flowing around the bluff body to keep the mass flow rate constant. In addition, pressure decreases along the x-axis, accordingly with the energy dissipation due to viscous stresses. Lift and drag coefficients are evaluated, the expected outcome is a first transitory phase followed by a constant value for both of them. This is due to the *steady* nature of the flow at this low Reynold number.

As we can see from the figure, the simulation results match the expected outcome and are consistent with the benchmark. The first image has a very thin tolerance band, but the relative error will be computed and shown later in this report.

5.1.2 Test Case 2D-2 (Unsteady)

The inflow condition is:

$$U(0, y, t) = \frac{4U_m y(H - y)}{H^2}, \quad V = 0 \tag{22}$$

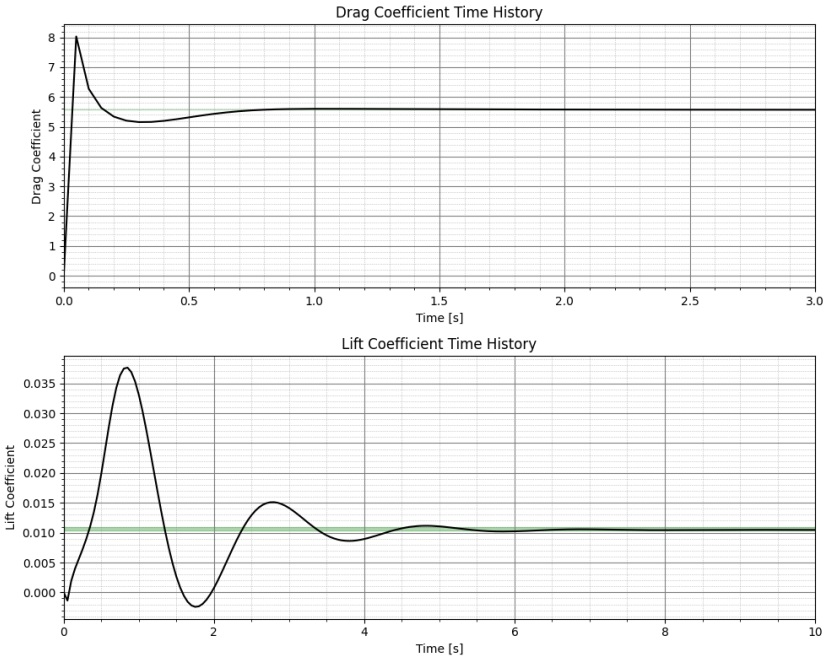

Figure 2: Time history of Drag and Lift coefficients for Test Case 2D-1. The steady state values fall within the expected benchmark [3] tolerance bands.

with $U_m = 1.5$ m/s, yielding $Re = 100$. The inlet velocity is kept constant throughout the period $(0, T)$ and the initial velocity is null over the whole domain. We set $T = 10\ s$. The value of the lift coefficient tends to decrease over time due to the non-stationarity of the test. This test cannot be considered stationary because the static initial condition combined with a high Reynolds number implies a strong effect of the time derivative $\frac{\partial u}{\partial t}$ on the system matrix. Lift and drag coefficients are evaluated, the expected outcome is a first transitory phase followed by an **oscillation phase** for both of them. This is due to the *unsteady* nature of the flow and the Von-Karman vortexes.

The figure clearly shows the oscillation that occurs to both coefficients after the transition phase.

For the lift coefficient in particular, we can see that the magnitude of the oscillation progressively increases until about $4.5s$. At this time an equilibrium is reached and the oscillations keep occurring but with the same period and magnitude. In green we see the tolerance bands taken from [3] and also reported in 1. We will show them better in figure 4.

As we can see our solver is very close to the upper and lower bounds given in the benchmark paper. The Strouhal number is also computed in the paper. The Strouhal number is defined as

$$St = \frac{Df}{U_m} \tag{23}$$

where f is the frequency of separation. D is the diameter of the cylinder, U_m is the average velocity. In our case $D = 0.1\ m$, $U_m = 1.0\ \frac{m}{s}$.
To find the frequency we did a FFT (Fast Fourier Transform) of the signal. This operation allows us to pass from the time domain to the frequencies domain and find the exact frequency of the oscillation. The peak can be seen in figure 7. In that figure we can also see the tolerance bands on the frequency that correspond to the Strouhal number bands on [3], calculated with the inverse of 23.

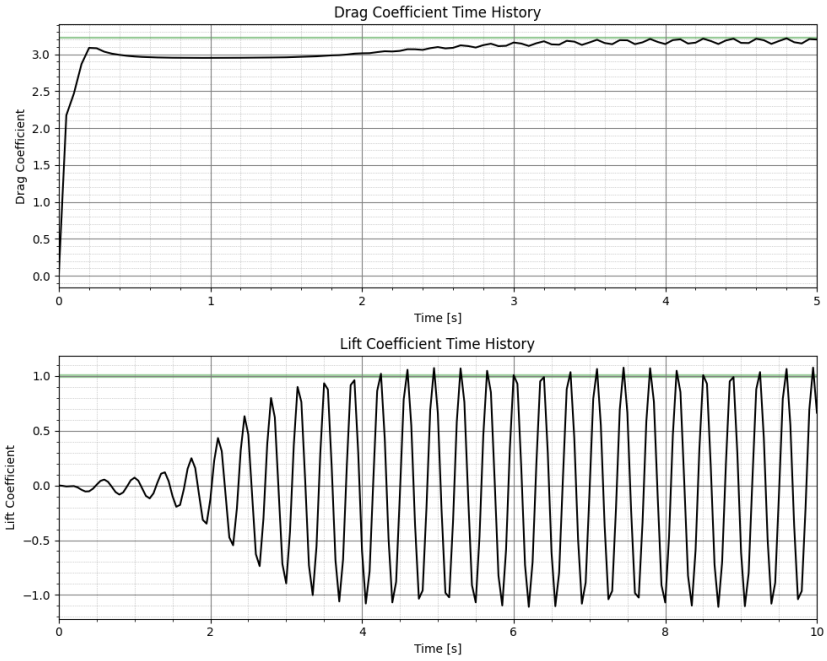

Figure 3: Time history of Drag and Lift coefficients for Test Case 2D-2.

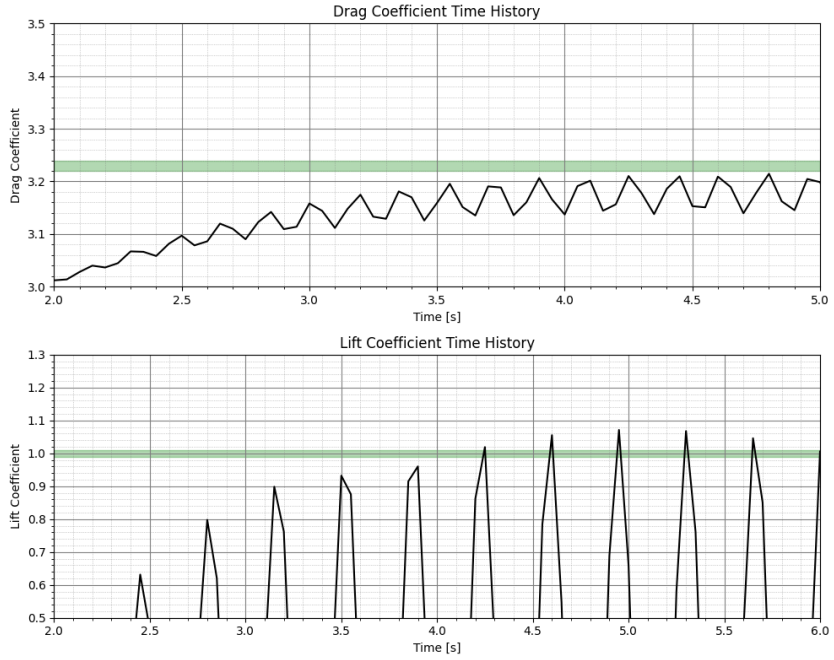

Figure 4: Time history of Drag and Lift coefficients for Test Case 2D-2, with a zoom to highlight tolerance bands.

5.1.3 Test Case 2D-3 (Unsteady)

The inflow condition is given by:

$$U(0, y, t) = \frac{4U_m y(H - y)\sin(\frac{\pi t}{8})}{H^2}, \quad V = 0 \tag{24}$$

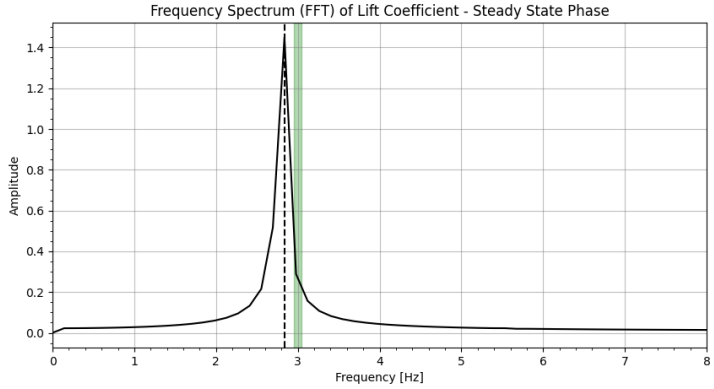

Figure 5: Frequency of oscillation, obtained with a FFT over the lift signal.

with $U_m = 1.5$ m/s and a time interval $0 \leq t \leq 8$ s. This results in a time-varying Reynolds number $0 \leq Re(t) \leq 100$. From $t = 4$ s onward, the results are influenced by the gradual decrease of the inlet velocity, indicating an overall deceleration of the flow. Lift and drag coefficients are evaluated, because the flow is driven by a variable inlet velocity, the behavior of the aerodynamic coefficients differs from a simple transitory-to-periodic phase. As shown in Figure 6, the drag coefficient predominantly follows the macroscopic sinusoidal trend of the fluid velocity. Conversely, the lift coefficient remains near zero at the beginning and end of the cycle, developing a localized **oscillation phase** only in the middle of the time window. This is due to the *unsteady* nature of the flow and the von Kármán vortex street, which triggers only when the instantaneous Reynolds number exceeds the critical threshold for vortex shedding.

As requested by the benchmark, the maximum values reached during the 8-second cycle are evaluated. The reference tolerance bounds are $c_{Dmax} \in [2.93, 2.97]$ and $c_{Lmax} \in [0.47, 0.49]$. Figure 6 demonstrates that the simulated peaks properly match the expected benchmark bands.

5.1.4 Error Analysis

In addition to the previously discussed results, we decided to condense the relative errors of each test into a single radar chart for the 2D cases, and another for the 3D cases.

Since the benchmark literature provides an acceptable tolerance band rather than a single exact value for the aerodynamic coefficients and Strouhal number, the reference value is taken as the arithmetic mean between the lower and upper bounds. The relative percentage error is thus computed as:

$$Err_{\%} = \frac{|\phi_{sim} - \phi_{ref}|}{|\phi_{ref}|} \times 100 \tag{25}$$

where ϕ_{sim} is the value obtained from our simulations and $\phi_{ref} = \frac{\phi_{min}+\phi_{max}}{2}$ is the reference value derived from the benchmark bounds. Figure 5 shows the resulting error distribution for the 2D test cases. Table 2 has accurate the computed values.

As depicted in the radar chart (Figure 5), the numerical solver demonstrates a highly overall agreement with the benchmark data, though with distinct behaviors depending on the specific quantity and flow regime.

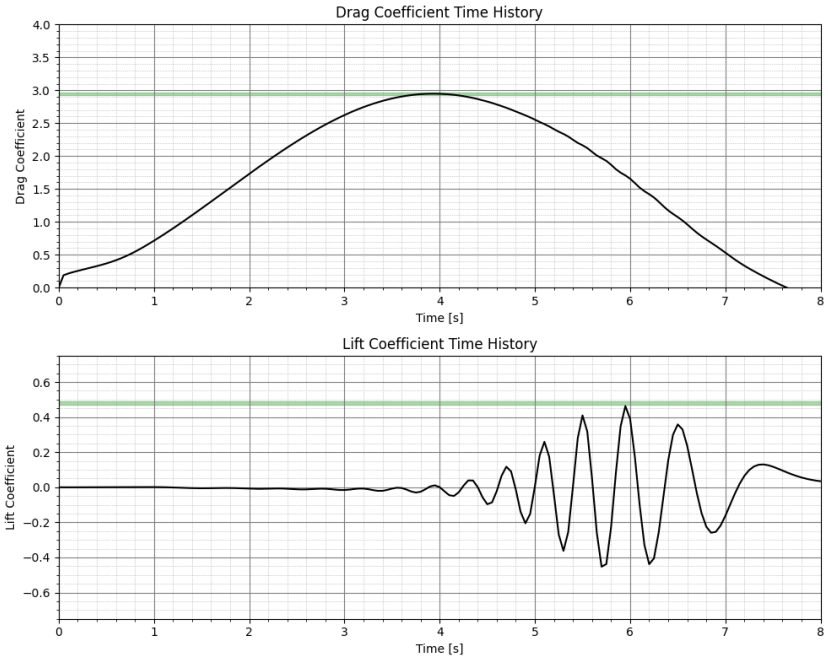

Figure 6: Time history of Drag and Lift coefficients for Test Case 2D-3.

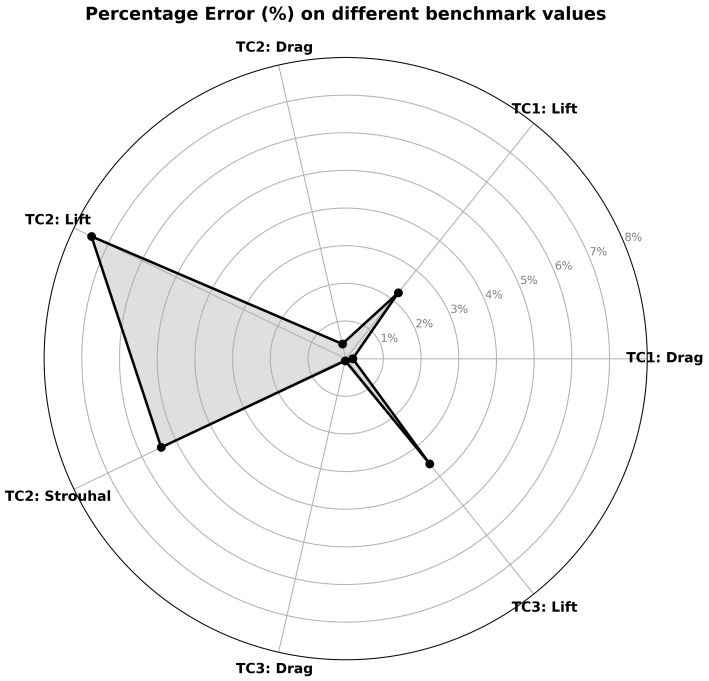

Figure 7: Radar error distribution for TC1,TC2 and TC3

First, the drag coefficient (c_D) is captured with excellent accuracy across all test cases (steady, unsteady, and variable), showing negligible relative errors (below 1%). Conversely, the lift coefficient (c_L) and the Strouhal number (St) prove to be more sensitive and challenging to predict accurately, particularly in the unsteady regimes (TC2 and TC3). Indeed, the maximum discrepancies are recorded for the TC2 lift and Strouhal number, peaking at roughly 7.5% and 6%, respectively.

This behavior aligns with the general findings of the benchmark authors: the lift coefficient and the shedding frequency are highly sensitive to the accurate resolution of the wake dynamics and boundary layer separation. Any numerical dissipation introduced by the spatial discretization, or the grid resolution itself, tends to slightly dampen the amplitude of the von Kármán vortex street, affecting c_{Lmax} and St more severely than the mean streamwise drag force. Keep in mind that these are errors referred to the mean value of the bounds. We might have a result which is in the correct window and still the error is not considered 0%.

5.2 3D Test Cases

The 3D simulations extend the benchmark to a three-dimensional channel with a cylindrical obstacle. The computational complexity increases significantly, requiring efficient parallel execution and robust preconditioning.

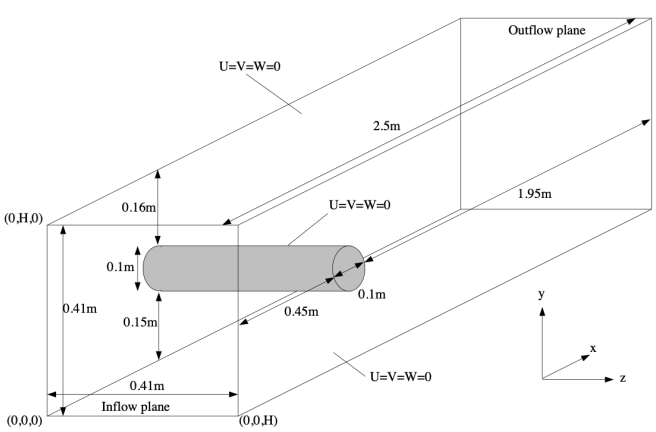

Figure 8: Configuration and boundary conditions for flow around a cylinder with circular cross–section.

5.2.1 Quantities of Interest

To validate the simulation against established benchmark data, the following physical quantities are evaluated:

- **Aerodynamic Coefficients:** The drag (C_D) and lift (C_L) coefficients are derived from the total force $\mathbf{F} = (F_D, F_L, F_W)$, computed by integrating the stress tensor (pressure and viscous components) over the cylinder surface S:

$$F_D = \int_S \left(\rho\nu \frac{\partial v_t}{\partial n} n_y - p n_x \right) dS, \quad F_L = -\int_S \left(\rho\nu \frac{\partial v_t}{\partial n} n_x + p n_y \right) dS$$

  The dimensionless coefficients are then normalized as $C_{D,L} = \frac{2F_{D,L}}{\rho \bar{U}^2 DH}$.

- **Pressure Difference:** The pressure drop ΔP is monitored between the stagnation point and the wake, specifically at $A(0.45, 0.2, 0.205)$ and $B(0.55, 0.2, 0.205)$.

5.2.2 Inflow Condition and Numerical Strategy

For the steady-state benchmark (Case 3D-1), a parabolic velocity profile is imposed at the inlet:

$$U(0, y, z) = \frac{16 U_m y z (H - y)(H - z)}{H^4}, \quad V = W = 0 \tag{26}$$

With $U_m = 0.45$ m/s, the Reynolds number is $Re = 20$. The numerical solution utilizes the SIMPLE preconditioner with a time step of $\Delta t = 0.05$. Convergence behavior typically shows an initial overhead that stabilizes as the flow reaches a steady state, leading to a reduction in the required iterations per time step.

6 Conclusions

In this project, a robust and scalable finite element solver for the incompressible Navier-Stokes equations was successfully developed and evaluated. Leveraging the `deal.II` library and MPI-based distributed memory parallelism, the solver was validated against the established Schäfer and Turek benchmark[3] for flow past a circular cylinder.

For the 2D test cases, the numerical results demonstrated excellent agreement with the reference data. The aerodynamic drag was predicted with high precision (relative errors below 1%) across steady, unsteady, and time-varying regimes. While the lift coefficient and Strouhal number exhibited slightly higher deviations—peaking at a 7.5% relative error in the unsteady case—this behavior is expected. It highlights the inherent sensitivity of lift and shedding frequencies to the spatial discretization and the numerical dissipation naturally introduced when resolving complex wake dynamics and boundary layer detachments.

From a computational standpoint, the SIMPLE and Yosida preconditioners, ensured stable and robust convergence of the GMRES solver when tackling the ill-conditioned saddle-point problem.

The extension to 3D domains highlighted both the capabilities and the limitations of the current numerical setup. While the low-Reynolds regime (Test Case unsteady 1) was successfully resolved, reaching a stable solution, the higher-velocity unsteady regime (Test Case unsteady 2) exhibited severe numerical instability, leading to divergence. This outcome underscores the steep non-linearities and computational demands of 3D transient flows. It suggests that resolving complex 3D vortex shedding at higher Reynolds numbers with standard Galerkin formulations requires either significantly *finer* spatio-temporal resolutions or the introduction of *specific stabilization techniques*, such as SUPG (Streamline Upwind Petrov-Galerkin) and VMS (Variational Multiscale) methods, which remain a primary objective for future developments.

A Tabulated Results Summary

Test Case	Quantity	Lower Bound	Upper Bound	Mean Value
2D-1 (Steady)	Drag (c_D)	5.5700	5.5900	5.5800
	Lift (c_L)	0.0104	0.0110	0.0107
2D-2 (Unsteady)	Max Drag (c_{Dmax})	3.2200	3.2400	3.2300
	Max Lift (c_{Lmax})	0.9900	1.0100	1.0000
	Strouhal (St)	0.2950	0.3050	0.3000
2D-3 (Variable)	Max Drag (c_{Dmax})	2.9300	2.9700	2.9500
	Max Lift (c_{Lmax})	0.4700	0.4900	0.4800

Table 1: Reference bounds and their computed arithmetic mean for the 2D benchmark test cases provided by Schäfer and Turek.

Test Case Errors	c_D	c_L	St
2D-1 (Steady)	0.19%	2.24%	-
2D-2 (Unsteady)	0.40%	7.49%	5.43%
2D-3 (Variable)	0.06%	3.57%	-

Table 2: Relative percentage errors of the computed aerodynamic quantities for the 2D benchmark test cases.

B Simulation Snapshots

B.1 Two Dimensions Test Case 1

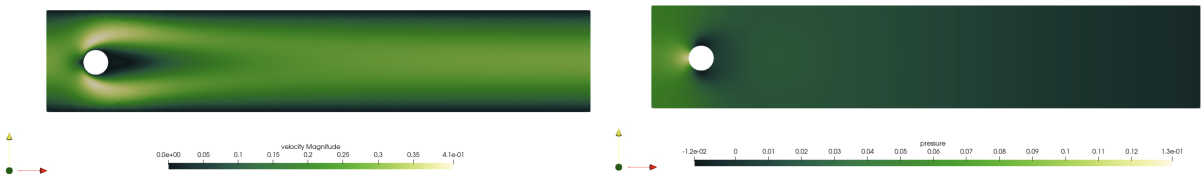

Figure 9: Overview of the fluid dynamics simulation of 2D Test Case 1. The left column shows the velocity magnitude field, while the right column displays the corresponding pressure distribution.

B.2 Two Dimensions Test Case 2

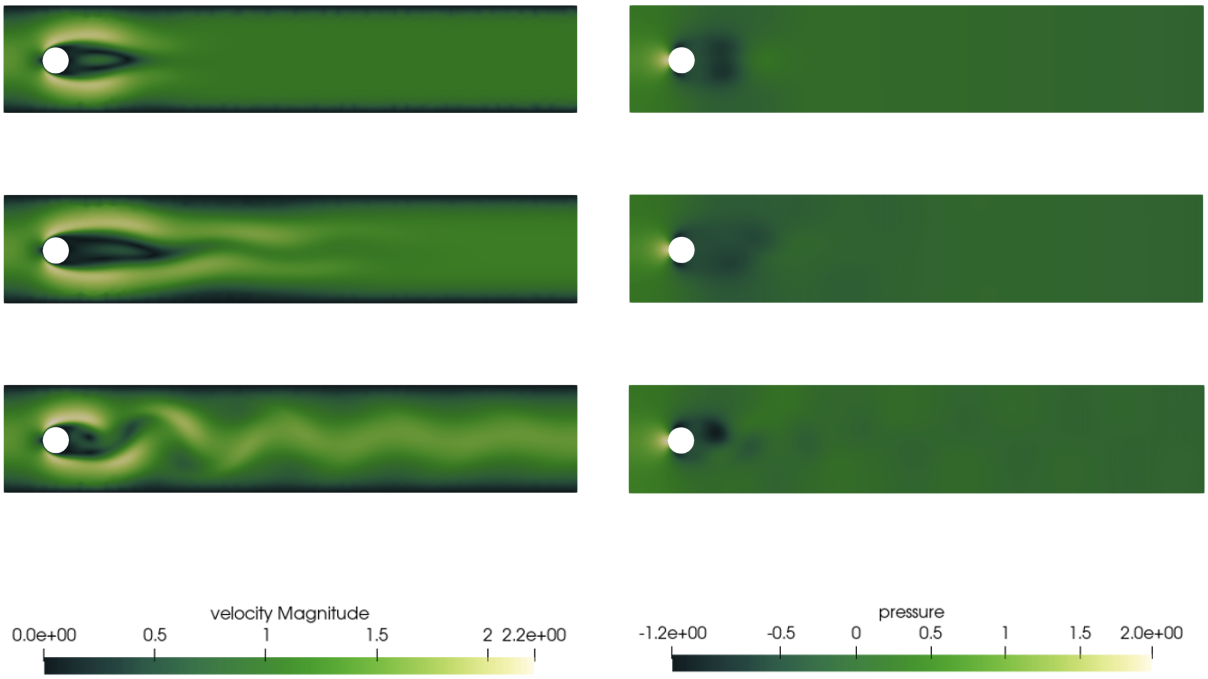

Figure 10: Overview of the fluid dynamics simulation of 2D Test Case 2. The left column shows the velocity magnitude fields, while the right column displays the corresponding pressure distributions.

B.3 Two Dimensions Test Case 3

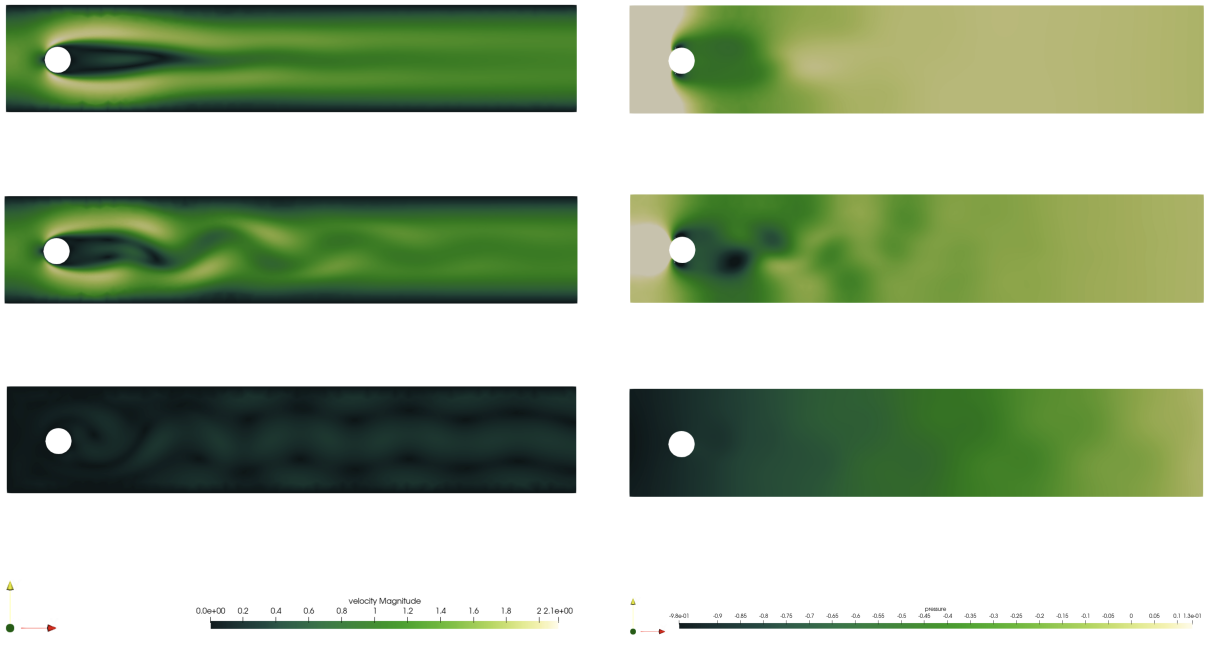

Figure 11: Overview of the fluid dynamics simulation of 2D Test Case 3. The left column shows the velocity magnitude fields, while the right column displays the corresponding pressure distributions at time 4, 5 and 8 seconds .

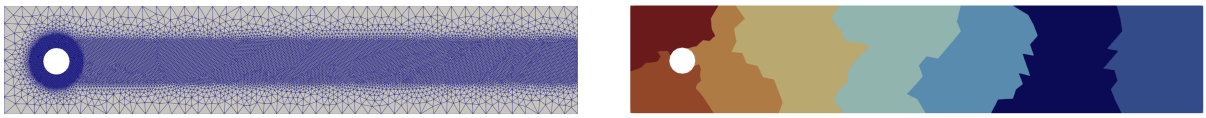

Figure 12: 2D mesh and partitioning among cores with `mpirun -n 8 ./main`, which is, 8 cores

B.4 Three Dimensions Test Case

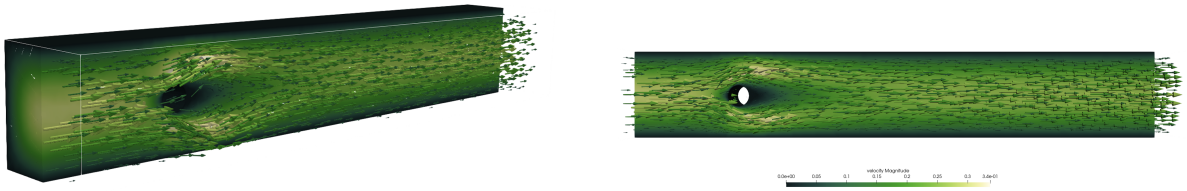

Figure 13: 3D simulation of TC1 with *Glyph* and *clip* filters on ParaView

References

[1] The deal.ii finite element library: https://dealii.org.

[2] Simone Deparis, Gwenol Grandperrin, and Alfio Quarteroni. Parallel preconditioners for the unsteady navier-stokes equations and applications to hemodynamics simulations. *Computers & Fluids*, 92:253–273, 2014.

[3] Michael Schäfer and Stefan Turek. Benchmark computations of laminar flow around a cylinder. In *Flow Simulation with High-Performance Computers II*, volume 52 of *Notes on Numerical Fluid Mechanics (NNFM)*, pages 547–566. Vieweg, 1996.