POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

High Performance Simulation Lab for Mechanical Engineering

3D Fluid–Structure Interaction simulation of NASA SC(2)-0410 airfoil

Authors: Mattia Gotti, Michele Milani, Stefano Pedretti, Federico Pinto

Professors: Prof. Paolo Schito, Prof. Stefano Miccoli

Assistant professor: Ing. Francesco Moro

Academic year: 2025-2026

---

1. Introduction

Fluid-Structure Interaction (FSI) is the study of the mutual effects between deformable structures and a surrounding fluid. This interaction is a critical aspect in numerous engineering domains, from aerospace to civil engineering. Due to the high complexity of the coupled phenomena, the majority of detailed FSI studies have historically been constrained to two dimensions (2D). However, recent advancements in high-performance computing and coupling libraries have enabled the investigation of full three-dimensional (3D) configurations, following high-fidelity partitioned approaches for complex aerodynamic studies [1].

This report presents a 3D FSI simulation of a deformable wing. The simulation is managed using an implicit coupling approach employing the open-source software: OpenFOAM for the fluid domain, FEniCS for the structural domain, and preCICE as the coupling library.

The structure of the report is as follows: first, the numerical methods are detailed, addressing the solvers for both domains. The coupling between FEniCS and OpenFOAM is explained. Second, the simulation results are discussed, and an investigation of the frequencies that are characteristic of the fluid–structure interactions is presented.

2. Problem Description

This section defines the test case under investigation and summarizes the geometric, fluid-dynamic, and structural parameters that characterize the simulation.

2.1. Geometry Parameters

The geometry under investigation is the NASA SC(2)-0410 airfoil, chosen for its well-documented aerodynamic properties and because it is widely under research for UAV, drones and transonic speed vehicles including airplanes. The model consists of an extruded airfoil placed within a wind tunnel domain.

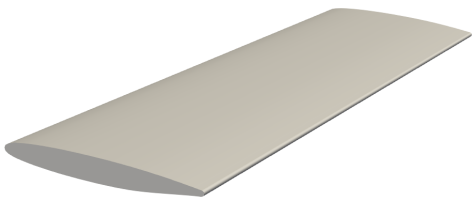

Figure 1: Wing geometry (3D).

The dimensions are chosen by scaling the approximate realistic geometry of a Boeing 737 wing. Since the reference chord is about 6 m, all dimensions have been normalized by this value to ensure a unitary chord for simplicity, obtaining:

- The airfoil chord (c) is 1 m.
- The wingspan (s) is 2.95 m.
- The angle of attack (α) is fixed at 5°.

The computational domain represents a wind tunnel with a total length of 20 m and a square cross-section of 8 m×8 m.

2.2. Flow Parameters

The fluid is air at standard conditions, characterized by a density $\rho_f \approx 1.225\,\text{kg/m}^3$ and a kinematic viscosity $\nu \approx 1.5 \times 10^{-5}\,\text{m}^2/\text{s}$. The fluid flows at a free-stream velocity (U_∞) of 100 m/s. The Reynolds number for this configuration is approximately 6.67×10^6, calculated as:

$$Re = \frac{U_\infty \cdot c}{\nu} \tag{1}$$

2.3. Structural Properties

The wing is modeled as a linear elastic material, assumed to be isotropic and homogeneous. The structural properties, corresponding to a standard aeronautical aluminum alloy, are:

- Young's modulus (E): 71.7 GPa.
- Poisson's ratio (ν): 0.33.
- Solid Density (ρ_s): $2810\,\text{kg/m}^3$.

The wing is fixed at its root (cantilever boundary condition), allowing for maximum displacement at the wingtip, primarily exhibiting bending motion along the vertical axis.

Fluid and Flow Properties		
Fluid Velocity	U_∞	100 m/s
Reynolds Number	Re	6.67×10^6
Fluid Density	ρ_f	$1.225\,\text{kg/m}^3$
Kinematic Viscosity	ν	$1.5 \times 10^{-5}\,\text{m}^2/\text{s}$
Geometrical Properties		
Chord Length	c	1 m
Wingspan	s	2.95 m
Angle of Attack	α	5°
Tunnel Length	L	20 m
Tunnel Section	$W \times H$	8 m × 8 m
Structural Properties		
Young's Modulus	E	71.7 GPa
Poisson's Ratio	ν	0.33
Solid Density	ρ_s	$2810\,\text{kg/m}^3$

Table 1: Summary of the test case parameters.

3. Mesh

This section details the mesh generation process for both the fluid and structural analyses.

3.1. Geometry Generation

The geometry was created in SolidWorks, starting from a sketch based on coordinate points contained in a `.dat` file retrieved from [2]. Originally, the dataset defined the upper and lower profiles as separate curves. The `.dat` file was manually modified to merge these points into a single continuous line defining the airfoil.

Since the raw data points did not form a closed loop at the trailing edge, determining the optimal closing strategy was a critical step. Initial attempts using a tight rounded arc resulted in poor quality for the fluid mesh. Conversely, a sharp trailing edge configuration caused issues with the structural mesh generation. Ultimately, a rounded arc with a larger radius was selected. This solution truncated the profile at a greater thickness compared to the first attempt, successfully resolving the meshing issues in both domains.

Once the 2D airfoil was defined, the 3D geometry was generated via extrusion and exported into two distinct file formats:

- `.STL`: used to generate the fluid mesh.
- `.STEP`: used to generate the solid mesh.

3.2. Fluid Mesh

The mesh generation process was carried out using the standard OpenFOAM utilities. The background domain was initially created using `blockMesh`. Then, the `snappyHexMesh` tool was used to carve the geometry—imported as an `.stl` file—out of the background grid and to locally refine the mesh near the walls, leveraging feature edges previously extracted by `surfaceFeatureExtract`.

To enhance the local flow resolution, two nested refinement regions were defined:

- `refinementBox`: a larger region surrounding the wing, set to 3 refinement levels.
- `innerRefinementBox`: a smaller region in the immediate vicinity of the airfoil, set to 4 refinement levels.

The mesh density follows a gradient to optimize computational efficiency: it transitions from high-resolution zones near the wing surface to coarser cells towards the far-field boundaries.

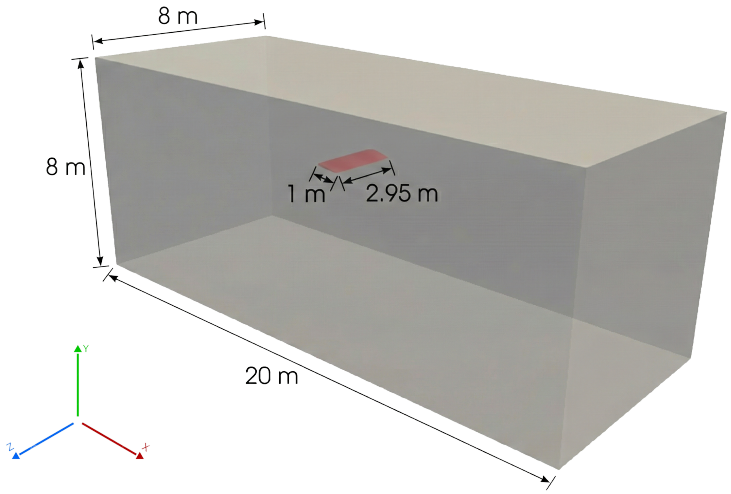

Figure 2: Geometric configuration.

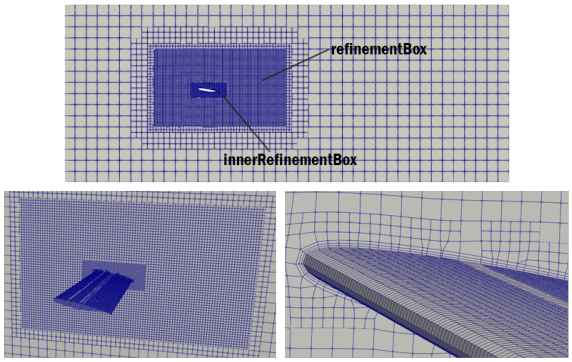

Figure 3: Fluid mesh discretization around the NASA SC(2)-0410 wing.

3.3. Solid Mesh

The structural mesh of the wing was generated using the finite element mesh generator Gmsh, starting from the CAD model in `.STEP` format.

The primary objective was to ensure high geometric fidelity of the airfoil profile while optimizing the computational cost for the finite element analysis in FEniCS.

3.3.1 Meshing Strategy

The solid domain was discretized using first-order (linear) tetrahedral elements, consistent with the function space formulation defined in the solver (`VectorFunctionSpace` with `'P', 1` elements).

For seamless integration with the `preCICE` adapter, specific *Physical Groups* were defined within Gmsh:

- **Volume:** The 3D body of the wing used for solving the linear elasticity and structural dynamics equations.
- **Fixed boundary (Root):** Located at the wing root, where a homogeneous Dirichlet boundary condition (clamped) is applied.
- **Coupling boundary (FSI Interface):** The external surface of the wing, which is subject to pressure and wall shear stress loads communicated by the fluid solver.

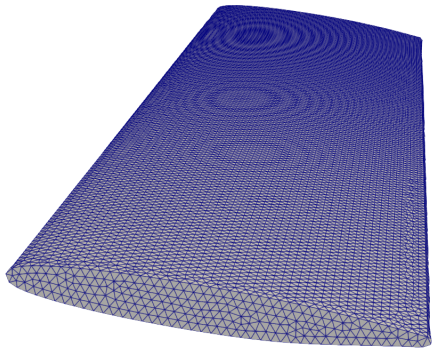

Figure 4: Solid mesh of the NASA SC(2)-0410 wing.

3.3.2 Mesh Properties and Conversion

A non-matching mesh approach was intentionally adopted between the fluid and solid interfaces. The transmission of data (displacements and forces) is efficiently handled by preCICE through radial basis function (RBF) mapping, ensuring high accuracy even on misaligned grids.

The generated mesh was converted using a script python to the `.XDMF` format to support parallel I/O and native integration within the FEniCS environment.

3.4. Mesh Independence Analysis

Two versions of the computational grid were generated for both the fluid and solid domains. These versions correspond to different levels of spatial refinement, providing a baseline configuration and a high-resolution configuration:

- **Coarse configuration:** used for preliminary validation.
- **Refined configuration:** employed for the final FSI simulations.

The fluid domain was discretized considering the different Angles of Attack (AoA) required for the study. The mesh metrics for each configuration are summarized in Table 7.

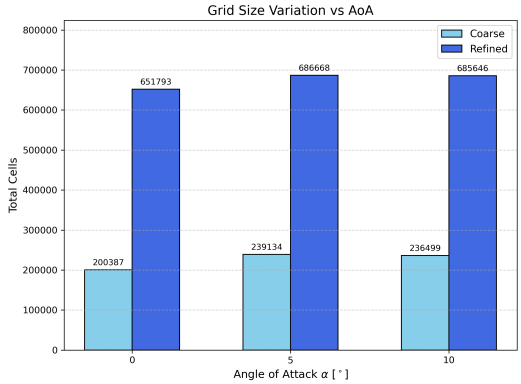

Figure 5: Variation of the total number of cells for Coarse and Refined mesh configurations as a function of the Angle of Attack (α).

The structural mesh characteristics are reported in Table 2 and a visual comparison of the refinement levels is provided in Figure 6.

The detailed *Mesh Independence Analysis* is presented in Section 7.1.

Refinement	Vertices	Elements
Coarse	26531	117151
Refined	103873	514608

Table 2: Structural mesh configurations properties.

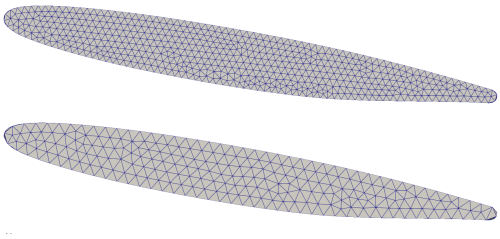

Figure 6: Cross-sectional comparison (XY plane) between the *Refined* (top) and *Coarse* (bottom) solid mesh.

4. Numerical Methods and Modelling Assumptions

This section details the numerical strategies adopted to address the fluid dynamics, structural mechanics, and their interaction. It also defines the modeling assumptions, specifying the boundary and initial conditions applied to each computational domain.

4.1. Fluid Simulation: OpenFOAM

The fluid dynamic simulation is conducted using **OpenFOAM** (v2306). The following paragraphs detail the mathematical model, the dynamic mesh and the solver configuration.

4.1.1 Governing Equations

The fluid behavior is described by the incompressible Navier-Stokes equations formulated in the **Arbitrary Lagrangian-Eulerian (ALE)** frame to account for the domain deformation. The conservation of mass and momentum are expressed as:

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 \\ \dfrac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} - \mathbf{u}_\Omega) \cdot \nabla \mathbf{u} = -\dfrac{1}{\rho} \nabla p + (\nu + \nu_t) \Delta \mathbf{u} \end{cases} \tag{2}$$

where $\mathbf{u}$ represents the fluid velocity, $\mathbf{u}_\Omega$ is the mesh velocity (computed based on the FSI interface displacement), p is the kinematic pressure, ρ is the density, ν is the kinematic viscosity, and ν_t denotes the turbulent eddy viscosity.

4.1.2 Dynamic Mesh

The deformation of the fluid domain is managed via the `displacementLaplacian` solver. The mesh update is performed by solving a Laplace equation for the mesh displacement:

$$\nabla \cdot (\gamma \nabla \mathbf{u}_\Omega) = 0 \tag{3}$$

The diffusivity coefficient γ is configured using a `quadratic inverseDistance` model relative to the `NASA` airfoil patch. This approach ensures that the boundary layer cells move rigidly with the airfoil to preserve their aspect ratio and quality, while the deformation is gradually absorbed by the coarser cells in the far-field.

4.1.3 Turbulence Modelling

Given the high Reynolds number, turbulence effects are modeled using the **Reynolds-Averaged Navier-Stokes (RANS)** approach with the $k-\omega$ SST model. It is a two equation model for the turbulence kinetic energy k and turbulence specific dissipation rate ω. It aims to overcome the defficiencies of the standard k-omega model wrt dependency on the freestream values of k and omega and it is able to capture flow separation. The source code is in `Foam::RASModels::kOmegaSST` and `Foam::kOmegaSSTBase`.

4.1.4 Solver Configuration

With an operational Mach number of $Ma \approx 0.3$, the flow is treated as incompressible. The transient solver `pimpleFoam` was employed. The PIMPLE algorithm (a combination of PISO and SIMPLE) is particularly advantageous for FSI applications as it ensures numerical stability even at Courant numbers (Co) greater than unity, allowing for larger time steps without compromising the accuracy of the transient coupling.

4.1.5 Boundary Conditions

The computational domain is defined by four primary boundary patches:

- `inlet`: Uniform velocity inflow.
- `outlet`: Static pressure outlet.
- `walls`: Impermeable wind tunnel boundaries.
- `NASA`: The wing surface, acting as the FSI interface.

For the `NASA` patch, a `movingWallVelocity` condition is applied. This enforces the no-slip condition where the fluid velocity at the boundary matches the structural deformation velocity calculated by FEniCS.

Patch Name	Velocity (u)	Pressure (p)
`inlet`	`fixedValue`	`zeroGradient`
`outlet`	`zeroGradient`	`fixedValue` (0)
`walls`	`noSlip`	`zeroGradient`
`NASA`	`movingWallVelocity`	`zeroGradient`

Table 3: Summary of the boundary conditions applied in OpenFOAM.

4.1.6 preCICE Adapter

The interaction between OpenFOAM and the coupling library is managed by the `preciceAdapterFunctionObject`, loaded within the `controlDict`. The use of a `functionObject` allows the standard `pimpleFoam` solver to be "coupling-ready" without any modification to its original C++ source code.

The behavior of this adapter is governed by the dictionary `preciceDict`. This file identifies the `NASA` wing surface as the interface and manages the bidirectional exchange of forces and displacements, ensuring that the fluid stresses are correctly integrated over the interface and communicated to the coupling environment.

4.2. Solid Simulation: FEniCS

The structural response of the wing is simulated using the open-source finite element (FEM) framework **FEniCS**. The structure is modeled as a linear elastic body subjected to dynamic loads and volumetric forces.

4.2.1 Governing Equations

The dynamics of the solid body are governed by the momentum balance equation in the Lagrangian formulation:

$$\rho_s \ddot{\mathbf{u}} - \nabla \cdot \boldsymbol{\sigma} = \mathbf{f} \quad \text{in } \Omega_s \tag{4}$$

where ρ_s is the material density, $\mathbf{u}$ is the displacement vector, and $\mathbf{f}$ represents the body forces.

For an isotropic, homogeneous, linear elastic material, the Cauchy stress tensor $\boldsymbol{\sigma}$ follows Hooke's law:

$$\boldsymbol{\sigma}(\mathbf{u}) = \lambda(\nabla \cdot \mathbf{u})\mathbf{I} + 2\mu\boldsymbol{\epsilon}(\mathbf{u}) \tag{5}$$

The infinitesimal strain tensor $\boldsymbol{\epsilon}(\mathbf{u})$ is defined as:

$$\boldsymbol{\epsilon}(\mathbf{u}) = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) \tag{6}$$

The elastic properties are defined by the Lamé constants λ and μ, which are related to Young's modulus (E) and Poisson's ratio (ν) by:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)} \tag{7}$$

4.2.2 Boundary Conditions

The structural domain features two primary types of boundary conditions:

- **Dirichlet (Clamped):** A rigid constraint ($u = 0$) is applied at the wing root, represented in blue in Figure 7.
- **Neumann (Coupling):** A variable surface load is applied to the airfoil interface. In this region, shown in red in Figure 7, preCICE maps the aerodynamic forces (pressure and skin friction) calculated by OpenFOAM.

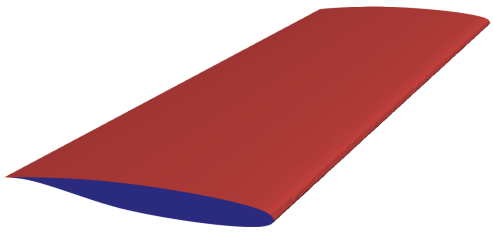

Figure 7: Structural boundary conditions: clamped root (blue) and FSI coupling interface (red).

4.2.3 Generalized-α Method

Time integration is performed using the Generalized-α scheme [3], a family of one-step, three-stage algorithms designed for structural dynamics. This method possesses numerical dissipation that can be controlled by the user through the spectral radius, denoted as ρ_∞.

It is specifically engineered to achieve high-frequency dissipation while minimizing unwanted low-frequency damping.

The algorithm evaluates the equation of motion at intermediate stages defined as:

$$\mathbf{u}_{n+1-\alpha_f} = (1-\alpha_f)\mathbf{u}_{n+1} + \alpha_f \mathbf{u}_n \tag{8}$$

$$\mathbf{v}_{n+1-\alpha_f} = (1-\alpha_f)\mathbf{v}_{n+1} + \alpha_f \mathbf{v}_n \tag{9}$$

$$\mathbf{a}_{n+1-\alpha_m} = (1-\alpha_m)\mathbf{a}_{n+1} + \alpha_m \mathbf{a}_n \tag{10}$$

To ensure the algorithm is unconditionally stable, second-order accurate, the parameters are mapped as follows:

$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1}, \quad \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1} \tag{11}$$

$$\gamma = \frac{1}{2} - \alpha_m + \alpha_f \tag{12}$$

$$\beta = \frac{1}{4}(1 - \alpha_m + \alpha_f)^2 \tag{13}$$

In this work, the parameters are set to $\alpha_m = 0.2$ and $\alpha_f = 0.4$, which satisfies the optimal case for $\rho_\infty = 2/3$. This specific configuration ensures that the low-frequency dissipation is minimized for the chosen level of high-frequency damping.

4.2.4 Variational Formulation

To implement the model in FEniCS, the governing equations are converted into a *weak form*. Instead of solving the equations at every point, the framework finds a solution that satisfies the overall balance of forces across the volume.

The implementation defines two main symbolic forms:

- **Mass Form (m):** Represents the inertia of the wing.
- **Stiffness Form (k):** Represents the internal elastic resistance of the aluminum.

These forms are combined into a single *residual* equation. By evaluating this residual at the intermediate α-stages defined by the Generalized-α method, the solver achieves second-order accuracy and ensures that high-frequency numerical noise is effectively damped, leading to a more stable solution during the transient analysis.

4.2.5 Checkpointing Logic

Given that the fluid and structural domains are coupled implicitly, the solvers must iteratively reach a synchronized solution within each time step. This process is managed through a **checkpointing** logic provided by the preCICE adapter:

1. At the beginning of a time window, FEniCS saves a "snapshot" of the the current structural state ($\mathbf{u}_n, \mathbf{v}_n, \mathbf{a}_n$).
2. If the coupling convergence criteria are not met at the end of an iteration, the adapter triggers a rollback. FEniCS discards the failed step, reloads the snapshot, and retries with updated force data.
3. Once convergence is reached, the fields are updated, and the simulation moves to the next time step.

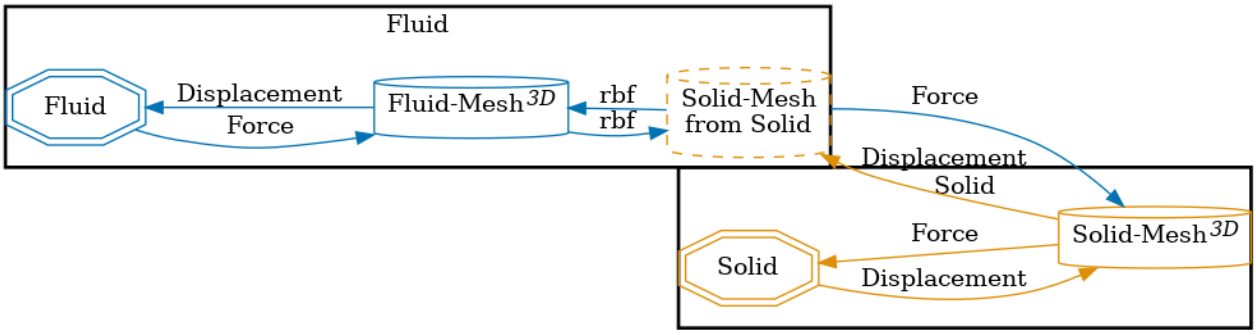

Figure 8: Data flow diagram and participant interaction within the preCICE environment.

5. preCICE configuration

The coupling between the fluid solver (OpenFOAM) and the structural solver (FEniCS) is managed using the **preCICE** (Precise Code Interaction Coupling Environment) library. The simulation workflow is defined in the `precice-config.xml` file, which specifies the data exchange methods, communication protocols, and convergence criteria. The coupling structure and data flow between participants are illustrated in Figure 8.

5.1. Participants

The simulation involves two main participants interacting through dedicated interfaces:

- **Fluid:** Manages the fluid domain by solving the RANS equations to calculate the aerodynamic loads (pressure and wall shear stress) acting on the airfoil surface.
- **Solid:** Operates on the structural domain, receiving the fluid loads to determine the elastic response and deformation of the structure.

To monitor the dynamic behavior of the wing, a *watch-point* was defined on the solid participant, located at the wingtip (`Flap-Tip`), to record displacement history over time.

5.2. Data Exchange and Mapping

The coupling requires the bidirectional exchange of two fundamental vectors: **Forces** (from fluid to solid) and **Displacements** (from solid to fluid). Since the meshes at the interface are non-conformal, preCICE handles the data interpolation using Radial Basis Function (**RBF**) mapping with a `compact-polynomial-c6` kernel [4]. This method provides superior accuracy compared to nearest-neighbor or nearest-projection mappings, as it constructs a global or local interpolant that respects the mesh topology.

To ensure physical consistency during the data transfer, two different mapping constraints are implemented according to the nature of the exchanged fields [4]:

- **Conservative mapping** for the Force vector: This constraint is essential for preserving the integral of the total loads (energy conservation) when transferring data between meshes of different resolutions. It ensures that the total work done by the fluid pressure on the solid surface remains consistent.
- **Consistent mapping** for the Displacement vector: This approach aims at maintaining the geometric integrity and smoothness of the deformed profile. It ensures that a constant displacement field is mapped exactly (reproduction of rigid body motions), preventing artificial gaps or overlaps at the interface.

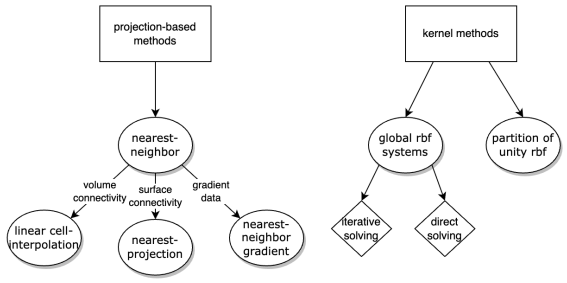

Figure 9: Mapping Options in preCICE

5.3. Coupling Logic and Numerical Acceleration

Due to the strong added-mass effect inherent in air-structure interactions at high velocities, a

parallel-implicit coupling scheme was adopted. Unlike explicit schemes, this configuration requires both solvers to iterate within the same time window until sub-cycling convergence is achieved.

To optimize the computational cost of these iterations, the **IQN-ILS** (Interface Quasi-Newton, Inverse Least-Squares) acceleration algorithm was employed. This quasi-Newton method significantly accelerates convergence by estimating the interface Jacobian without requiring internal information from the solvers' underlying equations.

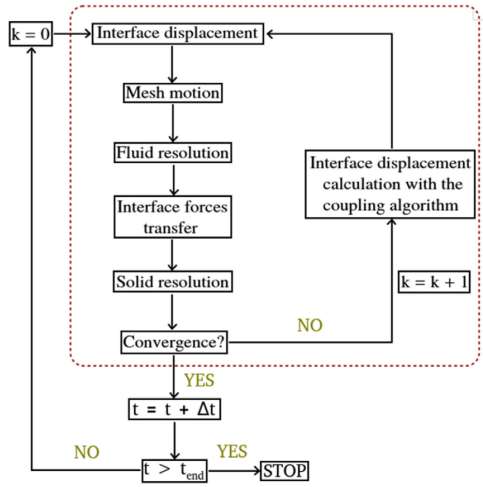

Figure 10: Numerical process of an implicit fsi simulation.

5.4. Communication Protocol

The communication logic is defined via the `<m2n:sockets>` tag. This tag and the adopted method will be explained here. It orchestrates how the parallel fluid solver interacts with the structural solver over the network. The specific attributes adopted for this simulation are:

- `m2n:sockets` which means "Many-To-None Sockets" is designed specifically for partitioned simulations where *at least* one participant (OpenFOAM in our case) is parallelized, while the participants reside on disjoint memory spaces (separate Docker containers in our case).
- The followed architecture is a client-server architecture used to establish the connection. `acceptor="Fluid"` is the server which binds the port and wait for a client connection. `connector="Solid"` is the client which actively initiates the connection request to the fluid participant.
- `network="eth0"` bridge the internal container network (`fsinet`), allowing the solvers to "see" each other using their service names.
- Last attribute is data aggregation, the most critical setting for the parallel efficiency of the coupling. It is also the most HPC one, it uses parallel pattern studied in many of our master courses. Since OpenFOAM runs in parallel (domain decomposition), the interface mesh is split among multiple processor cores. This option forces a centralized communication pattern:
  1. *Gather*: the interface forces computed by *each* OpenFOAM processor are gathered to the master rank.
  2. *Send/Receive*: the master rank exchanges the complete data vector with the Solid solver over the single TCP socket.
  3. *Scatter*: the received displacements are scattered back from the master rank to the respective OpenFOAM processors. This prevents the complexity of managing multiple socket connections from every individual MPI rank to the structural solver.

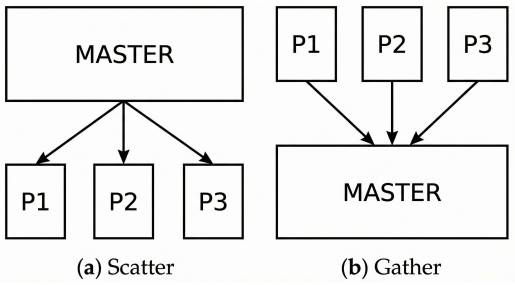

Figure 11: Simple schema of Scatter and Gather parallel patterns from a master rank

6. Computational Environment

The complexity of the FSI coupling, which requires the simultaneous execution of OpenFOAM, FEniCS, and the preCICE library, needed a reproducible and consistent computational environment. The simulations were performed using two primary infrastructures: local workstations and PoliMi's "Calimero" High-Performance-Computing cluster.

6.1. Containerization with Docker

To manage the complex dependencies of OpenFOAM, FEniCS, and preCICE simultaneously (different MPI versions, PETSc libraries, and Python environments), the entire computational workflow was containerized using **Docker** and orchestrated via Docker Compose.

Instead of installing all solvers in a single, heavy environment, we developed two specialized containers that act as independent "participants" in the FSI simulation.

- **Fluid Participant Container:** Derived from `opencfd/openfoam-default:2306`. This container is dedicated to the CFD solver. It requires a C++ build-chain to compile the preCICE library (v3.3.0) from source and to link the OpenFOAM-preCICE adapter.
- **Solid Participant Container:** Based on the `fenics-gmsh` image. This environment provides the Python 3 stack and the FEniCS finite element framework. It is optimized for structural mechanics and handles mesh conversion from GMSH to XDMF.

The synchronization between these two entities is handled by **Docker Compose**. A dedicated bridge network (`fsi_net`) was established to handle TCP/IP communication between the participants.

This solution ensures **reproducibility**, as the complete software stack is immutable and host indipendent. Guarantees also **dependency isolation**, preventing critical version conflicts between the incompatible system libraries required by the heterogeneous solvers.

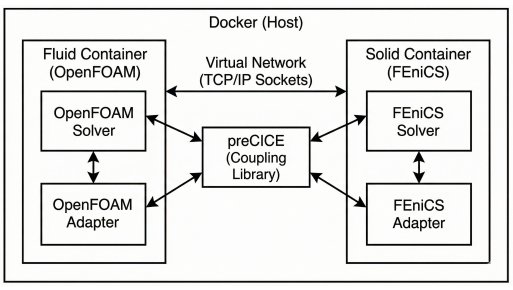

Figure 12: Containerized FSI Coupling Architecture

6.2. Solver Adapters

Since neither OpenFOAM nor FEniCS natively "speak" the preCICE protocol, an intermediate layer called an **Adapter** is required. These layers act as translators that extract data (forces or displacements) from one solver and translate it for the preCICE library.

- **OpenFOAM Adapter:** Compiled from source within the Docker container, this C++ *Function Object* allows the fluid solver to exchange forces and displacements at the interface. It works as an external plugin, meaning the core OpenFOAM source code remains untouched.
- **FEniCS Adapter:** Installed via `pip` as a Python package (`fenicsprecice`), it integrates directly into the structural script. It receives aerodynamic loads to calculate the wing's deformation and sends the updated nodal positions back to the fluid solver.

6.3. preCICE on Calimero

An attempt was made to build the preCICE coupling library (v3.3.0) from source within a user-space environment. A dedicated `Conda` environment was provisioned with the GNU Compiler Collection (GCC 14.3.0), OpenMPI (v4.1.6), and necessary dependencies including `LibXml2`, `Eigen`, and `Boost`. The build configuration was managed via CMake, utilizing explicit flags to enforce the usage of the environment's MPI wrappers and to disable optional bindings to minimize complexity. The configuration aimed to *isolate* the build from host system libraries. Even afer a successful compilation, the installation could not be finalized due to persistent **runtime linkage errors** and ABI incompatibilities:

- **Boost Version Mismatch:** A conflict persisted between build-time headers (v1.82.0) and runtime shared objects. The binaries attempted to link against newer ABI versions (v1.83+) present in the dynamic loader path, causing execution failures.
- **Persistent Dependency Linking:** Although the PETSc mapping feature was explicitly disabled via CMake flags (`-DPRECICE_FEATURE_PETSC_MAPPING=OFF`), the build system produced artifacts linked against `libpetsc.so`. This indicated an issue within the package discovery modules, resulting in *missing shared object* errors.

7. Automated Analysis

The fluid dynamics simulations were performed on the *Calimero* cluster at Politecnico di Milano. This high-performance computing resource, primarily dedicated to educational activities, is composed of three computing nodes equipped with Intel Xeon E5-2680 CPUs (Sandy Bridge architecture, 2.70GHz) running on Rocky Linux 8 [5].

To automate the case setup and manage different configurations (angle of attack, mesh refinement, and core count), a custom Python script named `setupCase.py` was developed. Based on command-line arguments, the script retrieves the necessary template files from dedicated source directories and overwrites the corresponding files in the active simulation folder. The directory structure is organized as follows:

- `meshes`: contains the specific `snappyHexMeshDict` dictionaries required for the mesh generation process.
- `stlFiles`: stores the geometry files (`.stl`) for each specific angle of attack.
- `cpuFiles`: includes the decomposition dictionaries (`decomposeParDict`) and the job submission scripts tailored for different core counts.

An auxiliary script, `infoSetup.py`, was also implemented to query and verify the currently loaded configuration parameters.

7.1. Mesh Independence Analysis

The mesh independence study was conducted by testing both the *Coarse* and *Refined* configurations across three different operating points (AoA of 0°, 5°, and 10°). The fluid mesh metrics for each case are summarized in Table 7.

To validate the numerical setup, complete simulations were performed for every combination. The average aerodynamic coefficients (C_l and C_d) were computed using a Python script, excluding the first 50 steps of each run to eliminate initial transient effects. The results are detailed in Table 8 and shown in Figure 13.

7.2. Domain Independence Analysis

A domain independence analysis was conducted to ensure that the finite size of the computational domain does not influence the results. This study was performed at a fixed angle of attack of 10° using the efined mesh configuration. The background mesh dimensions defined in the `blockMeshDict` file were varied as follows:

- **Original domain:** $40 \times 16 \times 16$ divisions.
- **1.25× scale:** $50 \times 20 \times 20$ divisions.
- **1.5× scale:** $60 \times 24 \times 24$ divisions.

We expect the main aerodynamic parameters (C_l, C_d) to remain independent of the domain size. The results are reported in Table 9.

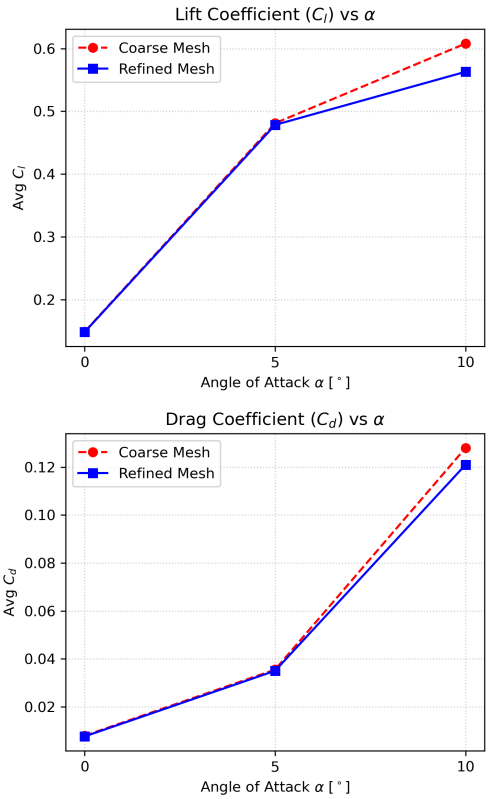

Figure 13: Comparison of lift (C_l, top) and drag (C_d, bottom) coefficients as a function of the angle of attack (α) for the two mesh configurations (Coarse and Refined).

7.3. Scalability Analysis

A scalability analysis was conducted to evaluate the performance gains achievable on the cluster. All simulations included in this study were executed on the same cluster node (queue `dida.q`) following this two-step procedure:

1. **Warm-up phase:** A preliminary setup phase was carried out for each combination of AoA and mesh type. This step involved meshing and advancing the simulation up to 0.25 s (500 iterations). This duration was selected to bypass the initial flow transient, a limit determined by monitoring the convergence of lift and drag coefficients. An arbitrary number of cores (typically 16) was

used for this step, as it does not influence the scalability metrics.

2. **Run phase:** This phase consists of a benchmark run of 200 iterations. The simulation restarts from the 0.25 s checkpoint saved during the warm-up phase and proceeds until 0.35 s. In this phase, the number of CPUs is the variable under analysis.

The scalability analysis was performed in two separate phases. The initial phase was performed under nominal cluster conditions on a representative subset of the domain configurations (AoA 0° Coarse/Refined and AoA 5° Coarse). These tests, covering core counts from 1 to 16, exhibited optimal scaling properties and are summarized in Table 10 and Figure 14.

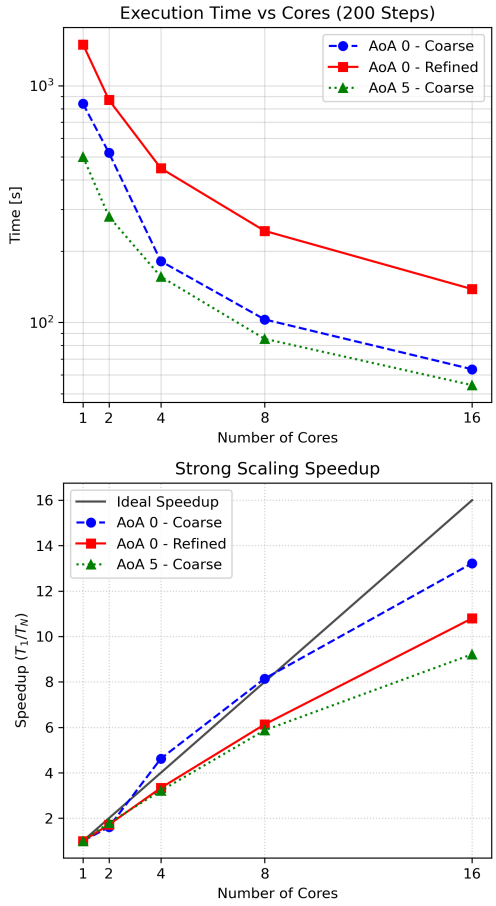

Figure 14: Execution time (top) and speedup (bottom) versus the number of cores for the second analysis.

A second campaign was subsequently undertaken to complete the analysis for the remaining configurations. During this phase, a generalized performance degradation was observed, characterized by increased execution times and reduced speedup. Consequently, the entire test suite was executed again—including the configurations from the first phase—to ensure a consistent dataset under these new load conditions.

For completeness and to highlight the impact of the computational environment on performance metrics, the results from both campaigns are reported. It is worth noting that, despite the difference in execution times, the simulations were performed following the identical methodology and numerical setup in both phases. The data from the second phase are detailed in Table 11 and Figure 15.

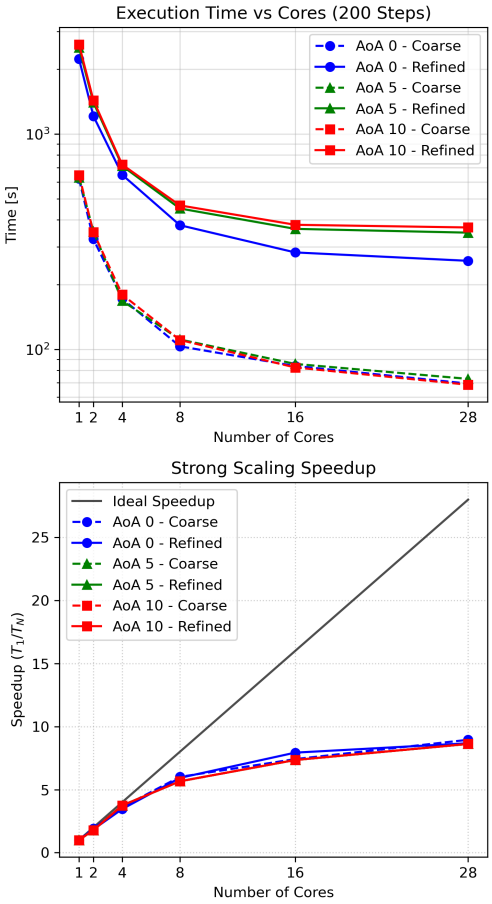

Figure 15: Execution time (top) and speedup (bottom) versus the number of cores for the second analysis.

8. Validation and Results

This section presents the validation of the numerical results obtained from both the coupled Fluid-Structure Interaction (FSI) simulations and the standalone Computational Fluid Dynamics (CFD) analysis. In addition, key physical parameters are reported, analyzed, and discussed.

8.1. Fluid dynamics analysis

The aerodynamic coefficients used as a baseline for the analyses in Section 7 are validated through a comparison with literature data. Additionally, a detailed analysis of the flow physics and key fluid dynamic parameters extracted from the simulations is presented to characterize the wing behavior.

8.1.1 Validation against Literature

The aerodynamic coefficients C_l and C_d obtained from the 3D simulations have been compared against reference data retrieved from [6]. It is important to note that these reference values, detailed in Table 4, correspond to a 2D airfoil profile and were computed using potential flow codes such as JavaFoil and XFOIL.

Consequently, a noticeable discrepancy between the datasets is observed. This deviation is physically consistent and is inherent to the difference between 2D (infinite span) and 3D (finite span) aerodynamics. In the 3D simulation, the pressure difference between the pressure side and the suction side causes the formation of wingtip vortices at the free end of the wing, as visualized in Figure 16.

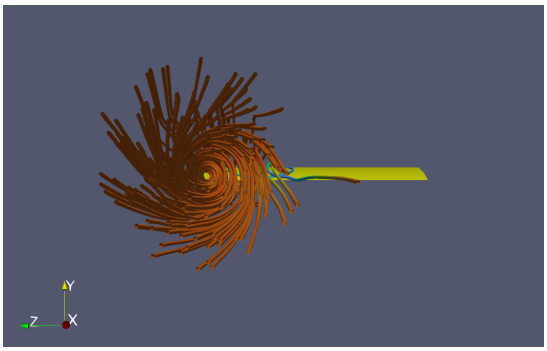

Figure 16: Visualization of the wingtip vortices.

These vortices induce a downwash velocity field that modifies the local flow angle. This phenomenon results in the generation of *induced drag* ($C_{D,i}$), which significantly increases the total C_d, and simultaneously reduces the effective angle of attack, leading to a lower lift coefficient (C_l) compared to the 2D predictions.

Analytical Verification To quantitatively assess the consistency of the 3D simulation results, a theoretical estimation is performed using

Table 4: Reference 2D aerodynamic coefficients.

AoA	**XFoil**		**JavaFoil**	
[°]	C_l	C_d	C_l	C_d
0	0.246	0.0054	0.274	0.0110
5	0.823	0.0079	0.833	0.0147
10	1.372	0.0124	1.176	0.0195

Table 5: Computed 3D aerodynamic coefficients.

AoA	**3D Simulation**	
[°]	**Avg** C_l	**Avg** C_d
0	0.148	0.0076
5	0.478	0.0350
10	0.563	0.1210

Prandtl's Lifting Line Theory. Since the 2D coefficients ($C_{d,2D}$) represent only the profile drag, the total 3D drag ($C_{D,3D}$) can be estimated by adding the induced drag component ($C_{D,i}$):

$$C_{D,3D}^{theory} \approx C_{d,2D} + \frac{C_{L,3D}^2}{\pi e AR} \tag{14}$$

where $C_{L,3D}$ is the lift coefficient obtained from the current simulation, AR is the wing Aspect Ratio ($AR = s/c = 2.95$), and e is the Oswald efficiency factor (assumed $e \approx 0.8$ for a rectangular wing).

By applying Equation 14 to all the three AoA configurations, using the XFoil profile drag as a baseline, the theoretical values show a closer agreement with the numerical results, as detailed in Table 6, confirming that the discrepancy is largely due to the induced drag phenomenon.

Table 6: Comparison between computed 3D Drag and theoretical estimation.

AoA	**Simulated** C_d	**Theoretical** C_d
0°	0.0076	0.0084
5°	0.0350	0.0387
10°	0.1210	0.0552

8.1.2 Velocity

The flow behavior around the wing is qualitatively analyzed through the velocity field visualizations presented below.

Figure 17 presents the streamlines of the flow field. While the fluid remains attached to the geometry over the majority of the chord, a visible detachment occurs near the trailing edge. This

behavior signals the presence of separated flow, consistent with the simulation conditions.

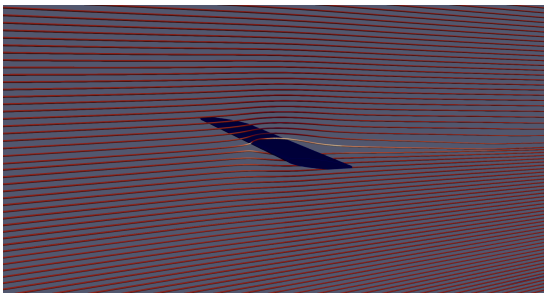

Figure 17: Visualization of the streamlines of the velocity field.

Figure 18 provides a contour map of the **velocity magnitude** (U). Three key aerodynamic regions can be identified:

- **Stagnation Point:** At the leading edge, a small region of zero velocity (dark blue) is visible, corresponding to the point where the flow impacts the nose of the airfoil.
- **Suction Side Acceleration:** On the upper surface (suction side), the fluid accelerates significantly to follow the surface curvature, reaching peak velocities indicated by the red area.
- **Wake Region:** Downstream of the trailing edge, a low-velocity zone (blue wake) is observed. The thickness of this wake is indicative of the energy loss and the resulting drag forces acting on the wing.

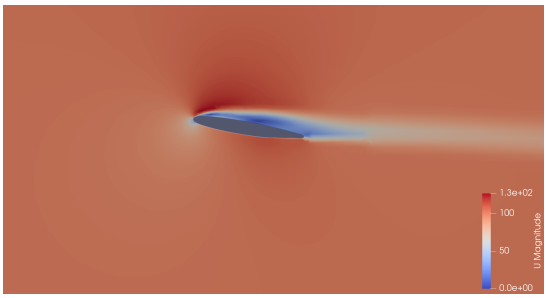

Figure 18: Visualization of the velocity field.

8.1.3 Pressure

The aerodynamic loads acting on the wing are driven by the static pressure distribution (p), which is visualized in Figure 19. The contour plot highlights the pressure field on a 2D slice of the domain.

The following aerodynamic features characterize the pressure field:

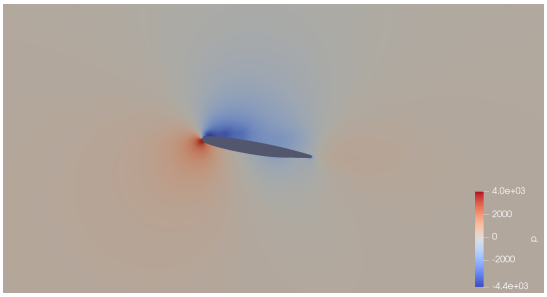

Figure 19: Visualization of the pressure field.

- **Stagnation Region:** At the leading edge, a localized area of maximum pressure (indicated in red) is clearly visible. This corresponds to the stagnation point, where the incoming fluid impacts the nose of the airfoil and the flow velocity decelerates to zero.
- **Suction Side (Upper Surface):** The dorsal side of the wing is dominated by a large region of negative pressure (indicated in blue). In accordance with Bernoulli's principle, this pressure drop is induced by the local acceleration of the fluid over the airfoil curvature and is the primary mechanism responsible for lift generation.
- **Pressure Recovery:** Moving downstream along the chord, the pressure gradually increases from the suction peak, recovering towards the ambient free-stream pressure near the trailing edge. The smoothness of this gradient suggests a stable flow attachment in the anterior section of the profile.

8.2. Effects of the fluid on the solid

This section explores the structural response of the wing under the influence of aerodynamic forces.

8.2.1 Tip Displacement

The temporal evolution of the vertical displacement provides a direct measure of the wing's deformation under the aerodynamic loads. Data extracted from the `Flap-Tip` watch-point show that the wing exhibits a primary bending motion along the vertical axis.
Based on the displacement time-history, the following observations can be made:

- **Transient and Periodic Behavior:** After an initial transient phase due to the impulsive start of the aerodynamic load, the structure settles into a stable periodic oscil-

lation. This suggests that the system has reached a dynamic limit cycle rather than a static deflection.

- **Displacement Range:** The vertical displacement fluctuates within a steady range of $12 - 13$ mm. The horizontal displacement remains significantly lower (approximately 2 mm), confirming that vertical bending is the primary structural response.
- **Numerical Stability:** The integration performed via the Generalized-α method ensures that high-frequency numerical noise is effectively dissipated. This is evidenced by the clean signal profile, which allows for a clear characterization of the low-frequency physical response.

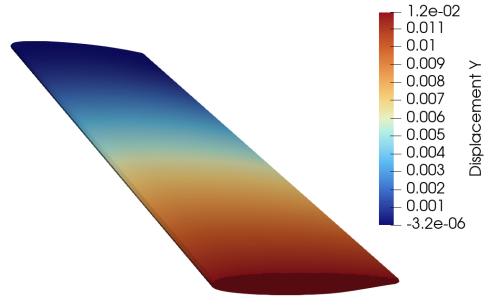

Figure 20: Tip displacement along y at $t = 0.05$ s.

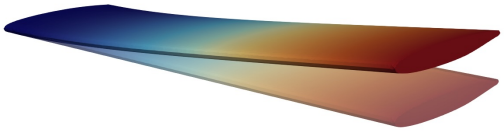

Figure 21: Warped displacement visualization (scale factor = 20).

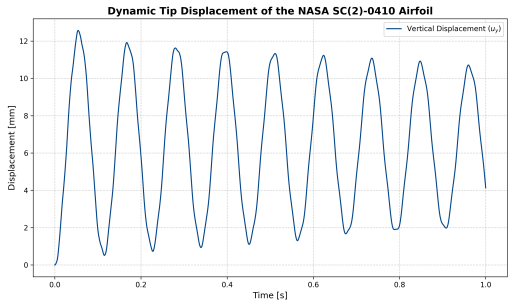

Figure 22: Tip Displacement over time

8.2.2 Spectral Analysis

To characterize the dynamic response of the wing under aerodynamic loading, a frequency-domain analysis was performed on the displacement signals recorded at the wingtip watchpoints. The study focused on three locations: the Leading Edge (LE-Tip), the Center (C-Tip), and the Trailing Edge (TE-Tip).

Firstly the best windowing method had to be chosen. A preliminary study was conducted to determine it by using the Power Spectral Density (PSD) calculation, aiming to minimize spectral leakage. For this reason three different windowing methods were plotted and analyzed. As we can see from figure 23, "*boxcar*" windowing method was not suitable for our purpose because it introduced a lot of noise which hides the interested frequencies. *Hamming* method has smooth curves, but higher noise than *blackman*. In the following graph we have chosen to plot with both Hamming and blackman windowing methods.

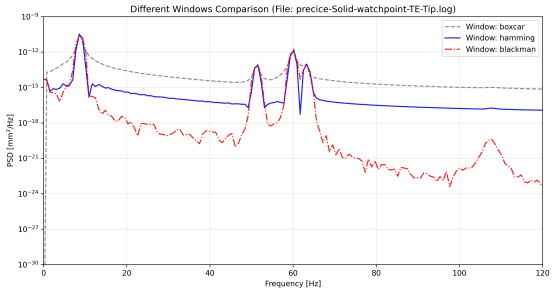

Figure 23: A comparison to see the best windowing method to choose.

The PSD analysis for the three watchpoints reveals a structural response characterized by distinct resonant peaks:

1. Primary Bending Mode (8 Hz): a dominant high-energy peak is observed in the low-frequency range. This corresponds to the first bending mode of the structure, which is consistent with the primary vertical motion observed in the time domain.
2. Higher-Order Modes ($50 - 62$ Hz): tree distinct secondary peaks are identified in the medium frequency range. The presence of these peaks across all three monitoring points (LE, C, and TE) indicates that global structural bending and torsional modes are being excited by the fluid flow.

To verify the **stability** of the interaction over

time, a *Short-Time Fourier Transform (STFT)* was performed, generating spectrograms for the simulation duration. The spectrograms (shown in the bottom panels of the analysis figures) display bright, horizontal bands corresponding to the identified frequencies. These bands remain *constant* in frequency throughout the simulation window. This temporal stability indicates that:

- The system operates in a stable regime without aeroelastic instability (flutter), which would otherwise manifest as diverging or shifting frequencies.
- The coupling between OpenFOAM and FEniCS successfully captures the damped oscillation of the aluminum structure as it settles around its aeroelastic equilibrium.

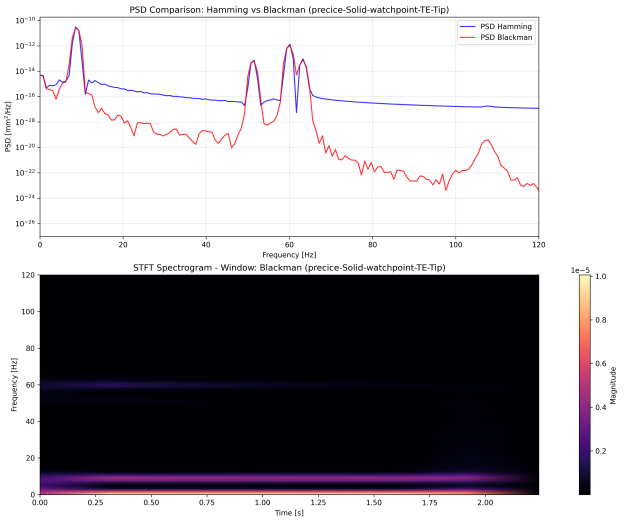

Figure 24: PSD and STFT analysis: Trailing Edge.

9. Conclusions

Analyzing a fluid-structure interaction is a complex problem, both regarding the required *software infrastructure* and the interaction of different physical phenomena; we succeeded in achieving a robust implementation that allowed us to study and validate the wing's behavior from purely fluid-dynamic, structural, and finally coupled perspectives, although the latter faced computational limits due to the difficulty of installing preCICE on the cluster and, more generally, the coupling environment.

Despite this, we studied different aspects such as velocity, pressure fields, and the wing deformation both in time and frequency space. We have learned to properly connect to and run different software in a **cluster environment**, which is important for our future careers. We have obtained results that are consistent with our expectations and, more importantly, the result obtained with this study provides a usable base framework from which to start simulating cases for various applications.

References

[1] Fabien Salmon and Ludovic Chatellier. 3d fluid–structure interaction simulation of an hydrofoil at low reynolds number. *Journal of Fluids and Structures*, 111:103573, 2022.

[2] Airfoil Tools. SC (2) 0410 airfoil details. URL `http://airfoiltools.com/airfoil/details?airfoil=sc20410-il`.

[3] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method. *Journal of Applied Mechanics*, 60(2):371–375, 1993.

[4] preCICE Developers. Mapping configuration - precice documentation, 2024. URL `https://precice.org/configuration-mapping.html`.

[5] CFDHub - Politecnico di Milano. CFDHub User Guide. URL `https://www.cfdhub.polimi.it/userguide/`.

[6] BigFoil. BigFoil Airfoil Database. URL `https://www.bigfoil.com/B/1793e619-1836-4920-a82e-a91210846f73_infoB1.php`.

Appendix

Supplementary tabulated results derived from the computational analyses are reported below.

AoA	Mesh	Points	Cells	Max Non-Ortho	Max Skew	Max AR	Min Vol $[m^3]$
0	Coarse	229443	200387	60.50°	2.95	27.04	2.33×10^{-8}
0	Refined	695615	651793	63.45°	2.94	27.04	2.91×10^{-8}
5	Coarse	274572	239134	61.91°	2.83	28.46	2.32×10^{-8}
5	Refined	736779	686668	61.90°	2.83	29.13	2.32×10^{-8}
10	Coarse	273251	236499	52.57°	2.97	26.25	2.98×10^{-8}
10	Refined	736939	685646	52.56°	3.15	26.25	2.98×10^{-8}

Table 7: Fluid mesh metrics across different Angles of Attack.

AoA	Mesh	Avg C_l	Avg C_d	Co Max	Co Mean	Avg y^+ (Airfoil)	Avg y^+ (Walls)
0	Coarse	0.1488	0.0079	12.12	0.13	104.5	664.7
0	Refined	0.1483	0.0076	13.46	0.18	105.3	408.3
5	Coarse	0.4811	0.0355	11.24	0.13	237.5	895.4
5	Refined	0.4784	0.0350	11.22	0.18	237.3	583.4
10	Coarse	0.6080	0.1280	10.84	0.13	225.6	1540.8
10	Refined	0.5630	0.1210	11.45	0.18	222.0	1101.1

Table 8: Aerodynamic coefficients and simulation metrics for the mesh independence analysis.

Domain Configuration	Avg C_l	Avg C_d
Original (Baseline)	0.563	0.121
1.25× Extended	0.593	0.123
1.5× Extended	0.585	0.123

Table 9: Aerodynamic coefficients for the domain independence analysis.

	AoA 0 - Coarse		AoA 0 - Refined		AoA 5 - Coarse	
Cores	Time [s]	SpeedUp	Time [s]	SpeedUp	Time [s]	SpeedUp
1	838.69	1.000	1493.14	1.000	500.69	1.000
2	520.24	1.612	869.79	1.717	279.73	1.790
4	181.38	4.624	447.77	3.335	156.00	3.210
8	102.96	8.146	243.59	6.130	85.28	5.871
16	63.41	13.226	138.26	10.800	54.32	9.217

Table 10: First scalability analysis results.

	AoA 0 - Coarse		AoA 0 - Refined		AoA 5 - Coarse		AoA 5 - Refined		AoA 10 - Coarse		AoA 10 - Refined	
Cores	Time [s]	S.U.	Time [s]	S.U.	Time [s]	S.U.	Time [s]	S.U.	Time [s]	S.U.	Time [s]	S.U.
1	621.29	1.000	2238.14	1.000	629.31	1.000	2522.37	1.000	645.72	1.000	2610.87	1.000
2	325.28	1.910	1212.55	1.846	352.98	1.783	1403.21	1.798	350.06	1.845	1435.93	1.818
4	172.10	3.610	646.89	3.460	168.18	3.742	711.53	3.545	179.68	3.594	721.09	3.621
8	103.38	6.010	377.53	5.928	111.20	5.659	451.39	5.588	110.63	5.837	466.61	5.595
16	83.76	7.418	282.17	7.932	85.64	7.348	363.02	6.948	82.46	7.831	379.22	6.885
28	69.41	8.951	257.93	8.677	72.98	8.623	348.59	7.236	68.55	9.420	368.50	7.085

Table 11: Second scalability analysis results.