



PhD in Design, Manufacturing and Operations Engineering

PhD Cycle XXXVII

Multi-Physics and Multi-Fidelity Approaches for Digital Twin Integration

Andrea Lopez

A.Y. 2024/2025

Tutor: Prof. Marco Evangelos Biancolini

Coordinator: Prof. Francesco Vivio

Multi-Physics and Multi-Fidelity Approaches for Digital Twin Integration

SUMMARY

Abstract	7
1. Introduction.....	8
2. Theoretical background.....	10
2.1. Radial Basis Functions (RBF)	10
2.2. Multi-physics and multi-fidelity methods.....	14
2.2.1. Multi-physics Analysis.....	14
2.2.1.1. Mapping Methods	14
Nearest neighbour interpolation.....	15
Weighted Residual Methods	15
Radial Basis Functions.....	16
2.2.1.2. Fluid-Structure Interaction (FSI)	16
One-way.....	17
Two-way.....	17
Modal superposition.....	17
2.2.1.3. Thermo-structural analyses	20
2.2.2. Multi-fidelity Analysis	22
2.2.2.1. Sub-modelling and homogenization.....	22
2.2.2.2. Low-fidelity and high-fidelity tool integration.....	23
2.3. Shape parametrization and Automatic workflows for Dataset Creation and Optimization.....	24
2.3.1. Shape Parametrization Methods	24
2.3.1.1. Parametric Cad.....	24
2.3.1.2. Mesh morphing.....	26
RBF Mesh Morphing	26
2.3.1.3. Hybrid Methods	27
OPAM Method.....	27
AeroSUV Method.....	30
2.3.2. Optimization methods	32
2.3.2.1. Optimization problem definition	32
2.3.2.2. Zero-order Methods	34
2.3.2.3. Gradient-based Methods.....	35
2.3.2.4. Parameter-less Methods	36
Adjoint	36
Biological growth method (BGM)	39
2.4. Big-data: Manage and transfer.....	41
2.4.1. Principal orthogonal decomposition (POD)	41

2.4.2.	Surrogate models.....	43
2.4.2.1.	Genetic Aggregation Response surface (GARS)	43
2.4.2.2.	Neural Network (NN)	43
2.4.2.3.	Gaussian Regression	44
2.4.2.4.	RBF network.....	44
2.4.3.	Functional mock-up unit (FMU).....	45
2.4.4.	Results Deploy.....	45
2.4.4.1.	MATLAB Dashboard	46
2.4.4.2.	VR Dashboard	47
3.	Multi-physics Analysis.....	50
3.1.	RBF Mapping for Steady FSI.....	50
3.1.1.	The Catenoid	50
3.1.2.	The RIBES wing.....	52
3.1.3.	Discussion	56
3.1.4.	Conclusions	57
3.2.	Comparison of two-way and modal superposition for Steady FSI: HIRENASD testcase.....	57
3.2.1.	Two-way method	59
3.2.2.	Modal Superposition.....	60
3.2.3.	Discussion	62
3.2.4.	Conclusions	63
4.	Multi-fidelity optimization.....	63
4.1.	Shape Optimization of Frame Structures through a Hybrid Analytical-2D and Numerical-3D Approach.....	63
4.1.1.	Introduction to the problem	64
4.1.2.	The Uniform-Strength Shape from the Timoshenko beam model	64
4.1.3.	Performance Factors.....	66
4.1.4.	Case 1: Frame structure	66
4.1.5.	Case 2: Lattice structure	71
4.1.6.	Case 3: Gear body	75
4.1.7.	Conclusions	80
5.	Adjoint Applications.....	81
5.1.	Adjoint sculpting with constraints	81
5.1.1.	CFD Model.....	82
5.1.2.	Results.....	82
5.1.3.	Conclusions	85
5.2.	Adjoint real time optimization.....	86
5.2.1.	CFD Model.....	88
5.2.2.	Shape Parameters	88

5.2.3.	Results.....	89
5.2.4.	Conclusions	90
5.3.	FSI – Adjoint coupling	91
5.3.1.	Testcase 1: cantilever beam.....	91
5.3.1.1.	Results.....	92
5.3.2.	Testcase 2: CRM Wing.....	94
5.3.2.1.	Results.....	96
5.3.3.	Testcase 3: F1 front wing	98
5.3.3.1.	Geometry	98
5.3.3.2.	CFD Model	99
5.3.3.3.	CFD Mesh.....	100
5.3.3.4.	FEM Model.....	102
5.3.3.5.	Results.....	103
	Case 1 : two-way vs. adjoint, velocity 50 m/s.....	104
	Case 2 : Speed-up, 65 m/s	105
	Case 3: Variation of Stacking Sequence.....	106
5.3.4.	Conclusions	107
6.	Reduced Order Models (ROM) Applications	108
6.1.	NACA0012 2D Aerofoil.....	108
6.1.1.	POD + GPR.....	108
6.1.2.	POD + RBF Network	110
6.1.3.	POD + Neural Network (NN)	111
6.1.4.	Conclusions	111
6.2.	ROM NACA Air Intake	111
6.2.1.	Preliminary Design	112
6.2.2.	Computational domain	114
6.2.3.	Boundary conditions	115
6.2.4.	Baseline Results	115
6.2.5.	Shape parameters.....	116
6.2.6.	ROM Building	118
6.2.7.	Ansys Twin Builder	118
6.3.	Scoop air intake ROM	123
6.3.1.	Preliminary Design	124
6.3.2.	CFD Optimization	125
6.3.3.	FEM Baseline.....	127
6.3.4.	ROM	130
6.4.	CUSP.....	134

6.4.1.	Payload Design.....	134
6.4.2.	Multi-physic Model.....	134
6.4.3.	Thermo-structural Analysis.....	135
6.4.4.	Temperature ROM.....	136
6.4.5.	Stress and displacements ROM.....	137
6.4.6.	Physical demonstrator.....	139
6.4.7.	Conclusions.....	140
6.5.	Open Parametric Aircraft Model (OPAM).....	140
6.5.1.	OPAM Project.....	140
6.5.2.	Hybrid Workflow.....	142
6.5.3.	Parametric CAD.....	143
6.5.4.	Design example according to the proposed workflow.....	143
6.5.4.1.	High fidelity CFD model of OPAM.....	143
6.5.4.2.	Parameters and DOE.....	144
6.5.4.3.	Response surface.....	146
6.5.4.4.	Mesh ROM.....	148
6.5.4.5.	Pressure ROM.....	149
6.5.4.6.	Optimization Results.....	151
6.5.4.7.	VR dashboard.....	153
6.5.5.	Conclusions.....	156
	Conclusions.....	157
	References.....	158

Abstract

The objective of this work is to develop a methodology for linking multi-physics and multi-fidelity analyses and to create workflows or optimization dashboards that make data accessible in real time for integration with digital twins. In particular, various problems have been investigated and different methods have been proposed.

In the context of multi-physics problems such as thermo-elastic or fluid-structure interactions, the main challenge is enabling communication between different tools and software. Here, mapping algorithms play a crucial role, as they allow the transfer of information between different CAE models. Another fundamental aspect is the shape parameterization of the model, where the same parameters must be applied to different CAE models. Various techniques have been explored in this regard, including CAD-based and mesh morphing approaches, and an innovative hybrid method has been proposed, combining the benefits of the aforementioned approaches.

Multi-fidelity approaches, on the other hand, refer to the possibility of integrating different tools, both analytical and numerical, within the same process, which represent the same problem with different levels of accuracy. Analytical methods are often used in the early design stages, providing quick results but with some simplifications. Numerical methods, such as high-fidelity CAE-based ones, offer more accurate results but come with much higher computational costs. Integrating these tools into a single workflow provides the right balance and optimizes the design process. In this context, an important challenge is how to transfer data from one level to the next to automate the entire workflow. Several solutions have been proposed.

Another critical aspect is managing large amounts of data to enable real-time predictive models. Various machine learning and AI techniques have been investigated. In particular, ROMs (reduced order models) and adjoint methods are two promising approaches that have been explored. ROMs allow the creation of a model from high-fidelity analyses that can be interacted with in real time. The idea is to reduce a problem by considering only the most influential features. ROMs are very precise and reliable but require large datasets for training and tuning the model. The adjoint method, on the other hand, allows the evaluation of the derivative of an objective function. This sensitivity can be used to predict the evolution of a complex system. The main limitation of this approach is the linearization around the operating point, so the prediction is highly reliable only for small variations of the considered model. For this reason, the adjoint method is often used in evolutionary and iterative methods, where the adjoint calculation is updated at each step. It is highly effective for fine-tuning around a design already in the final stage of the process. The main advantage of the adjoint method is that the computational cost does not depend on the number of parameters. Therefore, ideally, once the adjoint is calculated, all the shape modifications required can be tested without additional costs regardless their number.

Finally, in terms of the integration of the described tools and the deployment of results, FMUs (functional mock-up units) and VR (virtual reality) have been evaluated. The FMU standard defines a container and interface for exchanging models. It is supported by over 200 tools and is maintained as a project of the Modelica Association. It allows the integration of models developed with different simulation tools, facilitating collaboration between teams and organizations using different software. Thanks to standardization, models can be easily exchanged and reused, reducing development time and costs. The standard is suitable for the simulation of complex systems, allowing large models to be handled efficiently. ROMs can be encapsulated into FMUs getting at the same time the portability of the model outside the CAE software and very fast evaluation of new unknown states of the system. This approach opens to a variety of opportunities to exploit the reduced models, including VR experiences to offer a more immersive and intuitive interaction, enabling a better understanding of the physics involved in the studied problem.

1. Introduction

In recent decades, rapid advancements in simulation and computing technologies have profoundly transformed the way complex systems are designed, optimized, and analyzed. This progress has been driven by increasingly powerful computational tools capable of addressing multifaceted, multi-physics, and multi-fidelity problems, enabling the development of highly precise and sophisticated engineering solutions. Advanced methodologies, where data generated by different analyses can be integrated coherently and seamlessly, enhance both efficiency and predictive capabilities of models. High Performance Computing (HPC) lies at the core of this transformation (1) (2) (3) (4), as it offers the computational capacity to handle large-scale simulations and vast datasets, which are essential for tackling complex problems within limited time frames. The utilization of Graphics Processing Units (GPUs) has further accelerated this shift by enabling high parallelization of computations (5) (6), thereby facilitating simulations that demand high spatial and temporal resolution while reducing processing time considerably.

The increasing complexity of models and the massive amount of data generated by these simulations pose a significant challenge: effective management of big data. To address this need, innovative approaches such as reduced order models (ROMs) have been developed (7) (8). ROMs provide a solution by reducing the dimensionality of data while preserving essential information. This approach simplifies high-fidelity models, making real-time interaction and optimization feasible during the design and validation stages. By decreasing computational complexity without sacrificing precision, ROMs prove particularly advantageous for applications like digital twins, where rapid processing and real-time feedback are crucial for decision-making and predictive modeling.

In parallel, artificial intelligence (AI) and machine learning techniques are revolutionizing data analysis and interpretation (9) (10). Machine learning algorithms can capture and model the behavior of complex physical systems, improving predictive capabilities and optimizing decision-making processes. Integrating AI-driven techniques with physics-based models fosters a synergy that enhances accuracy, speed, and adaptability. This combination supports efficient data processing and offers actionable insights, helping engineers and researchers manage and leverage vast amounts of information with greater efficiency and precision.

In addition to data management and model optimization, the role of immersive visualization technologies such as virtual reality (VR) is gaining significant importance (11) (12). VR provides a transformative way to explore and interact with complex models, offering an intuitive and immersive experience for visualizing and analyzing multi-physics simulations. By immersing users within the simulated environment, VR enables a deeper understanding of physical interactions and structural behaviors, thus enhancing the ability to identify potential issues, optimize designs, and improve overall decision-making. Moreover, VR fosters collaborative work by allowing multidisciplinary teams to interact with models in shared virtual spaces, improving communication, accelerating development processes, and ensuring alignment across different aspects of a project.

This thesis aims to develop a comprehensive methodology for integrating multi-physics and multi-fidelity analyses, focusing on the creation of optimized workflows and dashboards that make data accessible and actionable in real-time, particularly in the context of digital twins. The research will explore various challenges, such as ensuring communication between diverse tools and software through mapping algorithms, parameterizing models for consistency across analyses, and employing advanced data compression techniques alongside AI-driven strategies to effectively manage large datasets. Special attention will be given to reduced order models (ROMs) for their ability to reduce the complexity of high-fidelity models without compromising on accuracy, as well as to adjoint methods, which offer a powerful means of evaluating the sensitivity of objective functions and refining designs. The study will also explore the integration of virtual reality (VR) to enhance model interaction, providing an immersive and intuitive environment for understanding complex phenomena, improving predictive capabilities, and fostering effective collaboration across disciplines.

The thesis is divided into two main parts. The first part describes the theoretical foundations, while the second presents various applications, structured as a collection of papers.

In the application part, several testcases that were addressed and studied are presented.

In the section 3, the focus was on multi-physics approaches, with particular attention to FSI problems. Two applications were proposed:

- The first involves a study on mapping methods (section 3.1), focusing on Radial Basis Functions (RBF). Two test cases of increasing complexity were analyzed: the catenoid and an airfoil from the literature.
- The second is an FSI study comparing two methods (section 3.2), two-way coupling and modal superposition, applied to a benchmark wing from the literature (HIRENASD).

In the section 4, the focus shifted to multi-fidelity approaches for optimization problems. Specifically, a mixed analytical-numerical optimization approach for lattice structures and frames was proposed.

The section 5 focuses on the adjoint method. In particular:

- The potential of the method is demonstrated for a classic optimization problem, incorporating the additional complexity of an external constraint, specifically an STL surface that must not be violated (section 5.1).
- The adjoint is used as a real-time prediction tool for any shape parameter (section 5.2). For this application, the aeroSUV car geometry from the literature was used, and a methodology was implemented to define a wrapper that envelopes the vehicle, manipulates any part of the wrapper, and transfers the deformation to the detailed mesh. For each parametric variation, the adjoint provides a real-time estimate of the observable's change. This tool was used to perform a "one-shot" optimization of the shape, improving the vehicle's aesthetics without compromising its aerodynamic performance.
- The adjoint is also used to predict the FSI behaviour of a structure (section 5.3). This method is similar to the previous one, but in this case, the deformation is imposed and corresponds to the structural deformation.

Finally, the last section showcases various ROM applications (section 6) for problems of increasing complexity and for multiphysics problems.

Specifically, regarding ROMs, several codes were tested, including open-source (Python-based) and commercial tools (Ansys Twin Builder), on cases of increasing complexity: a 2D airfoil (section 6.1), a NACA intake (6.2), a scoop intake (6.3), OPAM aircraft model (6.5) , and CUSP (6.4) (a CubeSat prototype developed in collaboration with INAF).

- The 2D airfoil was used to assess the accuracy of ROMs by comparing various algorithms for their generation.
- The NACA intake was employed to create an optimization dashboard enabling real-time evaluation of field quantities, such as pressure contours.
- The scoop intake was tested to evaluate the feasibility of integrating two different physics (fluid dynamics and structural) into the same real-time dashboard.
- For CUSP, a ROM was developed for temperature, stress, and deformation (thermo structural analysis). The aim was to create "virtual" sensors that provide temperature readings at any point in the structure, adding valuable information for potential structural considerations, such as stress and deformation fields. This approach simplifies the sensor system, reduces the CubeSat's weight, and enables temperature measurements at critical points where physical sensors cannot be installed.
- Finally, OPAM aircraft model was tested to create a real-time optimization dashboard within a VR environment.

2. Theoretical background

This section describes the theoretical concepts necessary for a full understanding of the application part. Many topics have been covered in the CAE framework, so the focus is on the essential notions needed to contextualize the presented applications.

2.1. Radial Basis Functions (RBF)

RBFs are a highly versatile and flexible tool, used in most of the applications presented. They are employed in any interpolation problem. The main applications proposed in this thesis are:

- In multi-physics problems, for mapping quantities across distinct physical domains. Specifically, in fluid-structure interaction (FSI) problems, RBFs can be used to map aerodynamic forces from the CFD mesh to the FEM mesh and to deform the mesh according to the computed structural deflection. More details on the FSI coupling can be found in Section 2.2 and in the application part (Section 3).
- For parameterizing a mesh in numerical simulations within optimization problems. Further details on this topic are provided in Section 2.3. In the application part, several use cases of RBFs for defining shape parameters for mesh morphing are presented, such as in the case of NACA and Scoop air intakes (Sections 6.2 and 6.3).
- This thesis also proposes a hybrid workflow (Section 2.3.1.3) in which the parameterization is defined at the CAD level, but RBFs are used to transfer the information from CAD to the mesh and perform mesh morphing. More details on this workflow are provided in Section 2.3.1.3, while the OPAM testcase is presented in the application part (Section 6.5).
- Finally, RBFs have been used to create surrogate models, such as response surfaces, to develop reduced-order models. More details on the use of RBFs for this purpose are reported in Section 2.4.2.4, while in the application part, they are tested in Section 6.1 on the reduced-order model of the 2D airfoil profile

Given two heterogeneous domains Ω_1 and Ω_2 , RBFs (13) (14) are used to interpolate quantities from Ω_1 to Ω_2 using a proximity criterion. The interpolation quality and its behaviour depends on the chosen RBFs. Typical radial functions are reported in Table 1.

Table 1: Typical radial functions

RBF	$\phi(\mathbf{r})$
Spline type	$ r ^n$
Multiquadric	$\sqrt{1 + r^2}$
Inverse multiquadric	$\frac{1}{\sqrt{1 + r^2}}$
Inverse quadratic	$\frac{1}{1 + r^2}$
Gaussian	e^{-r^2}

A linear system, with an order equal to the number of introduced source points, must be solved to calculate the coefficients. Once these unknown coefficients are determined, the value on any arbitrary point, whether inside or outside the domain (interpolation/extrapolation), is represented as the sum of the radial contributions from each source point, provided the point falls within the influence domain. Examples of RBF mapping of electromagnetic loads onto structures is given in (15), an evolution of the method has been developed in the RIBES project (16) and published in (17).

In mesh morphing, RBF find their ideal application, enabling efficient grid deformation while maintaining good mesh quality. Shape modifications are defined through a system of radial basis functions that generate nodal displacements based on a set of source points.

The process begins by defining the source points, or RBF points, where the user specifies known displacements. The RBF then interpolates these known displacements to the nodal positions using a proximity-based principle: each node's displacement is determined by its distance from adjacent source points. This allows the known displacement values to be defined anywhere within or outside the domain, at arbitrary positions. There is no requirement for a direct correspondence between source points and mesh nodes, making the process control-free and mesh-independent.

As a result, RBF-based methods are mesh-less, meaning the same solution can be applied to different meshes. This feature is particularly advantageous for reducing computational time: the problem can initially be analysed using a coarse mesh to find a solution through an appropriate optimization method. The displacements obtained can then be mapped onto a finer mesh, which is used for validating results and obtaining more precise estimates. Running the entire optimization process on a fine mesh would be computationally expensive, so this approach strikes an effective balance. However, this method is viable only if the physics underlying the optimization can be captured accurately enough by a coarser mesh.

The mesh-less nature of RBF methods also makes them well-suited for multi-physics problems. Furthermore, RBF offers a high level of expressiveness, allowing users to define source points with great flexibility and ease. This capability simplifies the parameter-setting process, enabling the definition of a parameter set without significant challenges.

Another advantage of RBF is the ability to control the interpolation behaviour of the function by selecting an appropriate basis. This flexibility ensures versatility and good mesh deformation quality, even for more complex cases.

From a mathematical perspective (21), solving the RBF problem involves calculating the coefficients of a linear system with an order equal to the number of source points. These coefficients enable the displacement of any arbitrary node on the grid to be expressed as the sum of the radial contributions from all source points. This approach allows the mesh deformation to be applied efficiently while preserving the topology of the grid, including the total number and types of elements.

Once the RBF fitting is performed, an analytical closed-form solution becomes available, defining the displacement field at any point in the domain. This function exactly matches the prescribed displacements at the source points and interpolates these values at the mesh nodes according to the behaviour dictated by the chosen radial function. If m is the number of source points, the interpolation function can be expressed as:

$$f(x) = \sum_{i=1}^m \gamma_i \phi(\|c_i - x\|) + p(x) \quad (2.1)$$

In Equation 2.1, the term $\phi(\|c_i - x\|)$ represents the chosen radial function, which depends on the distance between the source point c_i and the node x to be interpolated. γ_i is the weight associated with the radial basis function, and $p(x)$ is a polynomial term whose degree is determined by the selected basis. The polynomial term serves to regularize the solution.

It can be demonstrated (21) that a unique interpolating function exists if the RBF is conditionally positive definite. When this condition is satisfied, and if the order is equal to or less than 2, the polynomial term is defined as follows:

$$p(x) = \beta_1 + \beta_2 x_1 + \dots + \beta_{n+1} x_n \quad (2.2)$$

The unknowns of the system are the polynomial coefficients β_i and the weights γ_i of the radial basis functions. These can be determined by enforcing the following conditions:

- At the source points, the function takes the value of the known displacement: (g_i) :

$$f(c_i) = g_i \quad (2.3)$$

- Orthogonality of the coefficients:

$$\sum_{i=1}^m \gamma_i q(c_i) = 0 \quad (2.4)$$

In matrix form, the problem can be expressed as follows:

$$\begin{bmatrix} M & P \\ P^T & 0 \end{bmatrix} \begin{Bmatrix} \gamma \\ \beta \end{Bmatrix} = \begin{Bmatrix} g \\ 0 \end{Bmatrix} \quad (2.5)$$

The matrix M , also known as the interpolation matrix, contains all the distances between the RBF centres (source points). It is defined as: $M = \phi(\|c_i - c_j\|)$, for $1 \leq i \leq m$ and $1 \leq j \leq m$. P is a matrix $m \times (n+1)$ where each row j is defined as: $P_j = [1 \ x_1 \ x_2 \ \dots \ x_n]$.

For a 3d field the RBF interpolation is performed adopting three independent RBF, one for each component:

$$\begin{aligned} f^x(x) &= \sum_{i=1}^m \gamma_i^x \phi(\|c_i - x\|) + \beta_1^x + \beta_2^x x_1 + \beta_3^x x_2 + \beta_4^x x_3 \\ f^y(x) &= \sum_{i=1}^m \gamma_i^y \phi(\|c_i - x\|) + \beta_1^y + \beta_2^y x_1 + \beta_3^y x_2 + \beta_4^y x_3 \\ f^z(x) &= \sum_{i=1}^m \gamma_i^z \phi(\|c_i - x\|) + \beta_1^z + \beta_2^z x_1 + \beta_3^z x_2 + \beta_4^z x_3 \end{aligned} \quad (2.6)$$

It's worth noticing that multiple scalar interpolation (as the three component of a field) can be performed at a reduced cost as the linear system stays the same and the ability of linear system solvers to process multiple right vectors at the same time can be exploited.

A wide range of radial functions are available, each with distinct characteristics. As previously mentioned, the behaviour of the interpolation, while ensuring the prescribed values at the centre points, depends on the specific radial function chosen. The computational cost and the approaches to solving the RBF system also vary depending on the type of function. This diversity in behaviour is a feature that enhances the expressiveness of the obtainable shapes while strictly adhering to the constraints.

Generally, higher-degree functions allow for a more uniform modelling of surfaces, while spline functions tend to yield the best results when smoothing volumes. Figure 1 shows the most commonly used poly-harmonic functions. Additionally, Figure 2 illustrates the effect of increasing the degree of the RBF (top) or the radius (bottom). In both cases, the constraints are respected, and the source point is displaced by the same amount. However, significantly different surfaces are obtained in each case. The choice of RBF depends on the problem being analysed and the desired final surface.

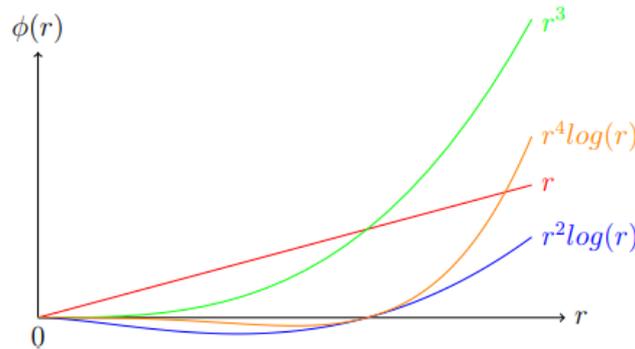


Figure 1: The most commonly used poly-harmonic functions, $r = \|c_i - x\|$

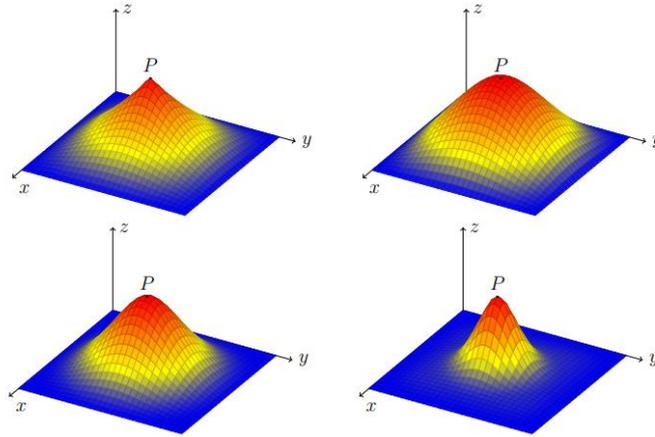


Figure 2: Interpolation using a linear poly-harmonic spline $\phi(r) = r$ (top left) and a cubic poly-harmonic spline $\phi(r) = r^3$ (top right). Interpolation using C2 Wendland functions with $r=0.7$ (bottom left) and $r=0.4$ (bottom right).

The numerical interpolation process can be divided into two phases: adaptation and evaluation. During the adaptation phase, which is the most computationally expensive, the centres are collected, and depending on the chosen RBF, the appropriate polynomial is defined. Then, the system in equation (2.5) is solved to obtain the polynomial coefficients and the base function weights. At this point, the calculation of the interpolated value at a given position is relatively inexpensive. This process is referred to as evaluation.

The main limitation of RBFs for industrial applications is the significant memory required for RBF solutions. The memory needed to simply store the square matrix in 2.5 increasing in the order of $O(N^2)$ for a $(N \times N)$ square matrix. This means that an RBF problem with 1000 centers requires 4MB of memory if using a double-precision floating-point format and 5GB for a 100000 source points problem. From a computational point of view using direct methods, for example exploiting a symmetric solver, requires $O(N^3)$ calculations. However, several strategies can be adopted to mitigate this problem. For example, one can exploit the distinction between sparse and dense matrices. Depending on the type of RBF used, the matrix M may contain many zero values. Sparse matrices can be easily compressed, requiring less memory.

Moreover, there are various strategies that can further reduce the adaptation or evaluation time and, consequently, the overall computational time. One commonly used method to accelerate the adaptation time when many source points are present is to approximate the interpolation by discarding some centres. This can be done using a distance-based algorithm that allows new source points to be gradually added if the error is too high. Another approach involves breaking the computationally expensive main problem into a set of smaller, more manageable subdomains (Partition Of Unity -POU- approach). Each solution to a subdomain is then combined with others using appropriate weighting functions to obtain the global solution. The final subdomains will typically be of varying sizes but contain an equal number of centres, resulting in comparable resolution times.

To achieve significant acceleration during the evaluation phase, one possible strategy is to use the Fast Multipole Method (FMM), derived from astrophysics to calculate the gravitational problem in which n bodies interact. The method distinguishes between near-field and far-field interactions, neglecting the influence of long-range interactions. A hierarchical data structure and proper organization of space are required to quickly evaluate near-field and far-field interactions at a given point in space. To evaluate an RBF in m given points using FMM the cost goes from $O(mN)$ of a direct evaluation to $O(m + N \log N)$.

Finally, as with other numerical problems, solving the problem using GPUs and AVX on the CPU can significantly speed up the computation. Iterative solvers are in fact based on multiple evaluations of the function and without entering the complexity of the solution algorithms, it's quite obvious that summations over a large cloud of points can be easily parallelized getting scalability efficiency very close to 100%.

2.2. Multi-physics and multi-fidelity methods

In recent decades, the increasing complexity of engineering and scientific systems has necessitated the development of innovative approaches for their modeling and simulation. Multi-physics and multi-fidelity methods have emerged as essential tools to tackle these challenges, allowing for the integration of different disciplines and levels of accuracy into a single coherent analysis. These approaches enhance the understanding of complex phenomena, optimize system performance, and reduce computational costs.

Multi-physics methods combine models that describe various physical disciplines, such as fluid mechanics, thermodynamics, structural mechanics and electromagnetism, to accurately capture interactions between different phenomena. On the other hand, multi-fidelity methods leverage models at different resolutions and levels of detail to balance precision and computational efficiency. The integration of these two approaches enables more reliable and detailed simulations, opening new possibilities for the design and analysis of complex systems. In this section, we will explore the fundamental principles of multi-physics and multi-fidelity methods, discuss their practical applications, and examine recent research advances. Through a series of examples and case studies, we will illustrate how these approaches can be used to address complex problems in various fields.

2.2.1. Multi-physics Analysis

In essence, a multi-physics system is composed of multiple components, each governed by its own principles for either evolution or equilibrium, usually through conservation or constitutive laws. A key classification in these systems is whether the coupling happens within the bulk (e.g., through source terms or constitutive relations that are active in the overlapping domains of the individual components) or across an idealized interface that is either lower dimensional or a narrow buffer zone (e.g., through boundary conditions that transmit fluxes, pressures, or displacements). Common examples of bulk-coupled multi-physics systems, which have their own extensively developed literature, include radiation with hydrodynamics in astrophysics (radiation hydrodynamics, or RHD (18)), electricity and magnetism with hydrodynamics in plasma physics (magnetohydrodynamics, or MHD (19)), and chemical reactions with transport in combustion or subsurface flows (reactive transport (20)). Examples of interface-coupled multi-physics systems include fluid–structure dynamics in aeroelasticity, core-edge coupling in tokamaks and thermoelastic coupling. In this section, interface coupling methods are addressed. In particular, mapping methods, which are essential in these approaches, are explored, and some applications in fluid-structure and thermoelastic analysis are examined.

2.2.1.1. Mapping Methods

In most multi-physics problems, a partitioned approach is commonly used, where separate solvers address the distinct physical phenomena individually, employing an appropriate coupling method. In this scenario, boundary conditions at the coupling interface for each single-physics problem must be specified. The interface values are then exchanged between the solvers. This method necessitates mapping techniques for physical variables between typically non-matching solver grids at the coupling surfaces. Important characteristics of such mappings include accuracy, consistency, and conservation of energy and momentum. Various methods for transferring data between non-matching meshes are discussed in the literature, such as nearest neighbour interpolation (21), projection methods (22), and spline-based interpolation methods (23).

The simplest and fastest method for information transfer is nearest neighbour interpolation, which involves obtaining information from the closest point in the other mesh. However, this method is only effective when the two grids are almost identical.

A more precise way to manage data transfer is through projection. Information from the other mesh can be obtained by orthogonally projecting a point onto that mesh, and the information at the projection point can then be used in the original point. This technique is useful when Gauss integration is applied in the calculation of integrals over the interface and is commonly used in fluid–structure interaction calculations. Similarly, an entire element can be orthogonally projected onto the other mesh, and the size of the intersection area can determine the extent to which the values of that element are considered.

The third method for data exchange involves spline-based methods, frequently employed in interpolation schemes in finite element methods, computer graphics, and medical imaging. Beckert and Wendland demonstrated that certain RBF with compact support have advantageous properties over other basis functions for data transfer (24). Smith et al. (25) identified and evaluated several radial basis functions for data transfer applications.

The previously mentioned methods handle coupling conditions directly. An alternative way to impose boundary conditions is through Lagrange multipliers (26). Lagrange multipliers methods are often used to couple displacements between two structural sub-domains. To manage non-conforming meshes, three-field and even four-field methods have been developed. However, when non-matching meshes are present, some interpolation or projection is still required. These dual methods are mainly used to couple two structural domains, such as in standard structural dynamics problems or contact problems. Only recently the coupling of two different physical domains, such as flow and structure, has been investigated. Currently, the flow is primarily modelled as a harmonic oscillator, and coupling a real flow and structure solver remains an open issue.

The mapping problem is explored in the application section 3.

Nearest neighbour interpolation

Nearest neighbour interpolation is a very simple method of transferring data from mesh A to mesh B. A search algorithm determines the point x_A in mesh A that is closest to a given point x_B in mesh B. The value for the variable in x_B is then taken to have the same value as x_A .

Weighted Residual Methods

The weighted residual method (27) can be used for mapping in multi-physics problems to handle the coupling and interpolation between different physical domains. Consider a multi-physics problem with two distinct domains Ω_1 and Ω_2 that interact through a common interface Γ . Suppose we want to transfer a physical quantity computed in the domain 1 and apply it on the domain 2. The residual R represents the difference between the quantity to be transferred, computed in the domain 1 f_1 , and the projected/interpolated quantity in the structural domain f_2 on the interface Γ :

$$R = f_1 - f_2 \quad (2.7)$$

We use a set of weight functions $\phi_i(x)$ defined on the interface Γ . These weight functions represent the bases with respect to which the residual is projected and must be chosen consistently with the problem discretization. The weighted residual method imposes that the weighted residual must be zero for each weight function ϕ_i :

$$\int_{\Gamma} R \phi_i(x) d\Gamma = \int_{\Gamma} (f_1 - f_2) \phi_i(x) d\Gamma = 0 \quad (2.8)$$

for every i . Rewriting the integral by separating the contributions gives:

$$\int_{\Gamma} f_1 \phi_i(x) d\Gamma = \int_{\Gamma} f_2 \phi_i(x) d\Gamma \quad (2.9)$$

When the problem is discretized, both f_1 and f_2 are represented as linear combinations of functions $\psi_j(x)$ associated with the mesh nodes:

$$f_1 \approx \sum_j c_j \psi_j(x), \quad f_2 \approx \sum_k d_k \psi_k(x) \quad (2.10)$$

where c_j and d_k are coefficients to be determined. The integral of the weighted residual becomes a sum of discrete contributions, which can be evaluated numerically using Gaussian integration:

$$\int_{\Gamma} f_1 \phi_i d\Gamma \approx \sum_g \omega_g f_1(x_g) \phi_i(x_g) \quad (2.11)$$

where x_g are the Gauss points and ω_g are their corresponding weights. Similarly, the contribution f_2 is evaluated using the same quadrature.

The integration results in a system of linear equations linking the coefficients c_j and d_k of the interpolating functions in the two domains, ensuring the consistent transfer of data between them. The system formulation can be expressed as:

$$\mathbf{Ac} = \mathbf{Bd} \quad (2.12)$$

where \mathbf{A} and \mathbf{B} are matrices resulting from the integral products of the basis and weight functions, and \mathbf{c} and \mathbf{d} represent the vectors of coefficients of the transferred data.

Radial Basis Functions

RBFs (13) (14) are used to interpolate quantities from Ω_1 to Ω_2 using a proximity criterion. The interpolation quality and its behaviour depends on the chosen RBFs. Typical radial functions are reported in Table 1. More details about RBF are reported in 2.1.

2.2.1.2. Fluid-Structure Interaction (FSI)

Advanced high fidelity simulations are more and more adopted at design stage with a strong interest to make the key predictions available at service as well. For systems interacting with flows, especially in situations where lightweight constraints affect substantially the flexibility, fluid structure interactions effect can't be neglected. In traditional CFD analyses, structures are typically assumed to be rigid, neglecting the FSI that could occur in real-world conditions. This interaction, where the fluid flow influences the structure and vice versa, is crucial for understanding the behaviour of systems exposed to aerodynamic forces. Historically, these interactions have been studied using wind tunnel tests, but such methods come with significant financial and time costs due to the need to build prototypes or scale models.

Aeroelastic and aerodynamic phenomena, though related, differ in their underlying mechanisms. Aerodynamic forces result from the interaction of the airflow with a rigid structure, while aeroelastic phenomena emerge when the structure deforms in response to these forces, potentially creating self-exciting oscillations. The general classification of aeroelastic and aerodynamic phenomena includes distinguishing between static and dynamic aeroelasticity. Among dynamic aeroelastic phenomena vortex shedding and lock-in, galloping and flutter are relevant. On the other hand, the main static aeroelasticity phenomenon considered in this work is torsional divergence, a static instability resulting from the loss of torsional stiffness due to a constant aerodynamic torque. The torsion of the structure increases the angle of attack (thus increasing the surface exposed to the wind), leading to a growth in the force acting on the structure, which in turn increases stress and deformation. If the twisting action exceeds the maximum torsional resistance, system instability can occur. Torsional divergence occurs at high flow speeds and affects slender structures when their elongated dimensions align with the incident wind direction.

To study FSI with CFD and FEM analysis there are generally 3 approaches:

- One-way (28);
- Two-way (29) (30);
- Modal Superposition (31) (32).

The one way method is generally used when deformation effects are negligible in the aerodynamic behaviour of the studied object. It is basically a mapping of the aerodynamic loads onto the FEM model.

The modal superposition method is less computationally onerous than the two-way method but also less accurate. Aerodynamic forces are projected in the direction of the main modes considered, and the resultant force is given by the linear superposition of the modal forces thus evaluated. Thus, a preliminary modal analysis on the FEM model is required. Next, an analysis of the forces exerted by the fluid on the structure is performed. To calculate the modal weights, a system of equations is solved for each natural mode. Each equation considers the balance of forces projected onto that mode. The modal weight for each mode represents the importance of that mode in the overall response of the structure to the fluid-structure interaction. The ultimate goal of this method is to simplify the analysis of structural response to the presence of fluid loads by reducing the number of degrees of freedom to be considered, but maintaining good accuracy in the results.

The two-way method is the most computationally onerous but also the most accurate. From the CFD analysis, the aerodynamic loads are calculated and exported. The loads are imported into the FEM model and the displacements are estimated. The displacements are used to deform the CFD mesh and have a more accurate estimate of the loads. The workflow is iterated until forces and displacements converge.

The FSI analysis and the comparison between different methods are also presented in the application part, in section 3

One-way

The one-way method is often used when a precise estimation of the elastic effects on the aerodynamics is not required, or when the aerodynamics is not significantly affected by structural deflection because the structure is particularly rigid, the loads are not particularly high, or the flow is especially stable. The method involves simply considering the CFD model as rigid to estimate the aerodynamic loads and using these loads to assess the stress state and deformation on the FEM model. Conversely, the effect of deformation on the CFD mesh is not evaluated. In this case, an algorithm to map the CFD loads onto the FEM mesh is sufficient, as the two meshes are non-matching.

Two-way

The two-way method is the most accurate approach for evaluating FSI. As mentioned in the previous section, this method is iterative. In the first step, the structure is assumed to be rigid, and the CFD analysis provides an initial estimate of the aerodynamic loads. These forces are then mapped onto the FEM mesh as nodal loads (or element loads), and the finite element analysis provides a first estimate of structural deformation. It is important to emphasize the need for a load mapping procedure from the fluid mesh to the structural mesh. Generally, the surfaces are overlapped, but due to the different element densities and types of the two meshes for computational reasons, the meshes themselves do not coincide.

At this point, the CFD geometry is corrected by accounting for the deformable geometry. In this phase, a methodology to deform the mesh according to the structural deformation is required. Usually, a mesh morphing technique is used, applying a mapping method to interpolate the FEM displacements onto the CFD mesh. In each iteration, the aerodynamic forces and structural displacements are monitored, and the process is repeated until convergence is reached. Typically, 4-5 iterations are sufficient.

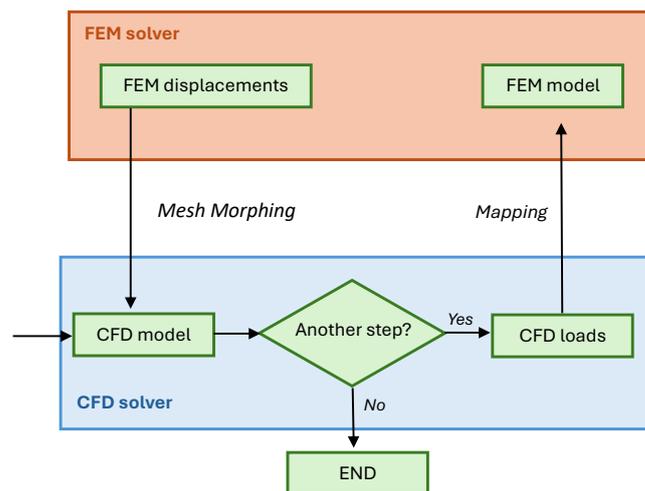


Figure 3: Two-way FSI workflow

Modal superposition

Modal analysis is a fundamental theory in structural mechanics that applies to both continuous and discrete systems. It focuses on identifying the vibration characteristics of a structure, specifically the undamped free vibration modes, each characterized by a distinct natural frequency and mode shape.

Natural frequencies refer to the specific frequencies at which a structure tends to vibrate when disturbed, while mode shapes describe the corresponding deformation patterns at these frequencies. The values of these vibration

parameters (natural frequencies and mode shapes) depend on the structural properties and boundary conditions. These parameters play a crucial role in designing structures subjected to dynamic loads, as they are integral for various dynamic analyses, including spectrum analysis, mode superposition, harmonic, and transient analyses. In finite element modelling (FEM), where continuous systems are represented by their discrete counterparts, modal analysis enables the assessment of a system's dynamic behaviour, provided that the assumption of linearity is valid. This assumption excludes nonlinearities stemming from material behaviour, large deformations, or contact interactions. The motion of a complex system, represented by n masses at specific locations (nodes of the mesh), is described by a system of second-order differential equations:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f} \quad (2.13)$$

Where \mathbf{M} , \mathbf{C} and \mathbf{K} respectively are the structural mass, viscous damping and stiffness matrices for the complete system, $\ddot{\mathbf{x}}$, $\dot{\mathbf{x}}$ and \mathbf{x} respectively are the nodal acceleration, velocity and displacement vectors at a specific time t , \mathbf{f} is the load vector of time-dependent loads that are applied to the masses constituting the system and vary in time with a known fashion.

The modal analysis formulation starts from the solution of a special case of the equation 2.13, namely free undamped condition:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{0} \quad (2.14)$$

The solution to this equation, which represents the system's response as a function of time, can be expressed in terms of harmonic functions of the following form:

$$\mathbf{x} = \mathbf{X} \sin(\omega_n t) \quad (2.15)$$

where \mathbf{X} is the eigenvector, or mode shape, and ω_n is the natural frequency. Substituting the equation 2.15 in the equation 2.14 the following relationship can be obtained:

$$(\mathbf{K} - \omega_n^2 \mathbf{M})\mathbf{X} = \mathbf{0} \quad (2.16)$$

that, apart from the null solution, has solution if the matrix $(\mathbf{K} - \omega_n^2 \mathbf{M})$ is singular.

Before continuing with the mathematical formulation of mode superposition, it is important to introduce a key characteristic of the previously mentioned eigenvectors \mathbf{X} . Eigenvectors corresponding to different natural frequencies are orthogonal with respect to both the structural stiffness and mass matrices. This implies that the eigenvectors are linearly independent. Another important property of the natural modes is that the scaling or magnitude of the eigenvectors is arbitrary. Since they are intrinsic to the structure, their shape remains unchanged during vibration, with only their amplitude varying. Therefore, for computational purposes, a convenient normalization of these eigenvectors is typically performed (33):

$$\mathbf{X}_i^T \mathbf{M} \mathbf{X}_i = 1 \quad (2.17)$$

Introducing the modal vector coordinates $\mathbf{q} = \mathbf{X}^{-1} \cdot \mathbf{x}$, for the stiffness matrix the following relationship is obtained:

$$\Omega = \mathbf{x}^T \mathbf{k} \mathbf{x} = \begin{bmatrix} \omega_{n,1}^2 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \omega_{n,n}^2 \end{bmatrix} \quad (2.18)$$

By substituting, in the hypothesis that \mathbf{C} is diagonal:

$$\ddot{q}_i + 2\zeta_i \omega_{n,i} \dot{q}_i + \omega_{n,i}^2 \cdot q_i = \frac{F_i(t)}{M_{ii}} \quad (2.19)$$

Where $\omega_{n,i} = \sqrt{\frac{K_{ii}}{M_{ii}}}$ and $\zeta_i = \frac{C_{ii}}{2M_{ii}\omega_{n,i}}$.

In cases where the deformation velocity is much greater than the velocity at which loads are applied, the physical phenomenon of FSI can be treated as a static problem. Under this assumption, equation 2.19 can be simplified and written as:

$$\omega_{n,i}^2 \cdot q_i = \frac{F_i(t)}{M_{ii}} \quad (2.20)$$

Taking into account the equation $\mathbf{q} = \mathbf{X}^{-1} \cdot \mathbf{x}$:

$$\mathbf{x} = \sum_{i=1}^N \mathbf{X}_i q_i \quad (2.21)$$

In other words, the total deformation of the structure under steady-state conditions can be determined by linearly combining a selected number of modes, using their modal coordinates as weights. For the purpose of the mode superposition technique, a subset of the lower modes is typically chosen, as mechanical systems are generally low-pass in nature. Consequently, the lowest frequency modes tend to have the highest energy levels and thus play a more significant role in the system's behaviour.

As a result, the complete solution to the eigenvalue problem can be approximated by retaining only a limited number of the lowest modes, which leads to a reduction in the DoF that need to be computed, while still adequately capturing the structure's vibration signature. The minimum number of modes to retain for an accurate FSI study can be determined by evaluating their modal participation factors, which are typically required to exceed a certain threshold value to ensure a reliable approximation.

To perform the steady FSI analysis using mode superposition, equation 2.21 is employed. The workflow followed in this method is based on three phases:

- Phase 1: The vibration modes of the deformable parts, normalized with respect to mass, are computed through a FEM analysis.
- Phase 2: The vibration modes are imported into the CFD model. This requires the introduction of a mesh morphing technique and a mapping algorithm to deform the CFD mesh according to the linear combination of the extracted modes.
- Phase 3: The weights of each mode are estimated based on equilibrium considerations. The mesh is then deformed, and the aerodynamic variation is evaluated.

Figure 4 illustrates the workflow followed with the mode superposition strategy.

The structural response is evaluated during the CFD calculation in the modal space. Based on equation 2.16, the mesh is made parametric:

$$\mathbf{x}_{CFD} = \mathbf{x}_{CFD0} + \sum_m \mathbf{X}_m q_m \quad (2.22)$$

Where m are the considered modes, \mathbf{x}_{CFD} is the final position of a generic node, \mathbf{x}_{CFD0} is the initial position of the same node.

The nodal forces f_m are integrated projecting the surface forces onto the modes that, for the discrete mesh results in the projection of nodal forces F_{CFD} onto the modal displacements over all the surfaces:

$$f_m = \sum_{i=1}^n \mathbf{X}_{m,i} \cdot F_{CFD_i} \quad (2.23)$$

The m -th modal load is obtained by summing the scalar product between the nodal displacement of the mode and the nodal load at each i -th node of the n nodes on the surface. It is also emphasized that the m -th modal load is a scalar. Since the extraction of modes involved the definition of a mass normalization criterion, the modal coordinates for the static problem are expressed according to the following relation:

$$q_m = \frac{f_m}{\omega_m^2} \quad (2.24)$$

By substituting into equation 2.16, the result is obtained as follows:

$$\mathbf{x}_{CFD} = \mathbf{x}_{CFD0} + \sum_m \frac{f_m}{\omega_m^2} \mathbf{X}_m \quad (2.25)$$

Equation 2.19 presents the relationship that governs the updating of the CFD mesh. It is important to highlight that, since the modes are precomputed, the indicated workflow does not require interaction between two software programs as in the two-way case, thereby significantly simplifying and accelerating the workflow. As a matter of fact we can say that once the modes are embedded the CFD model becomes flexible, i.e. capable to

self-adapt the shape according to the actual loads. This means that, once the structural modes are embedded, different flow conditions can be swiftly captured by the flexible CFD model.

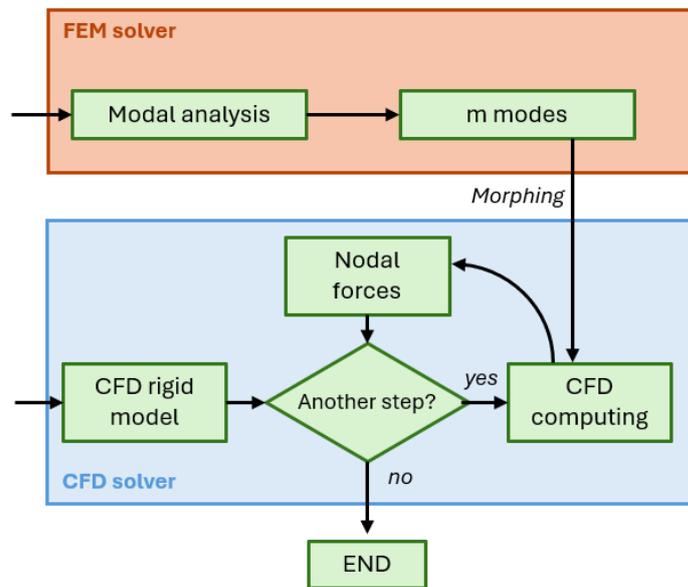


Figure 4: Modal superposition workflow

2.2.1.3. Thermo-structural analyses

Materials can be classified into five major categories:

- **Metallic Materials:** Composed of metallic elements, these materials exhibit a crystalline structure, high mechanical strength, and significant density. They are malleable, ductile, and good conductors of heat and electricity, while being generally opaque to light.
- **Ceramic Materials:** Consisting of oxides and/or silicates, ceramics typically have a crystalline structure (except for glass). They are hard and brittle, highly resistant to corrosion and wear, and poor conductors of heat and electricity. Additionally, they possess high melting temperatures.
- **Polymeric Materials:** These materials are macromolecular organic compounds, either synthetic or natural. Their structure consists of molecular chains, giving them low density and poor dimensional stability. They are generally amorphous, easily moldable, and tend to soften or combust under high temperatures.
- **Composite Materials:** Composites are formed by combining two or more mutually insoluble materials, where one acts as the matrix (binding phase) and the others as reinforcement or filler. The reinforcement enhances stiffness and strength in structural materials, while fillers can improve properties such as flame resistance, abrasion resistance, impact resistance, and electrical conductivity. The matrix is typically polymeric but can also be metallic or ceramic. The reinforcement phase can be composed of long fibers, short fibers, or particles.
- **Metamaterials:** These are a new class of engineered materials designed for specific functions. Their unique properties arise not from their chemical composition but from their micro- or nanoscale structural design.

Materials are characterized by mechanical properties (such as static strength, yield strength, hardness, resilience, and density) and thermal properties (such as thermal expansion, heat capacity, and thermal conductivity). Certain mechanical properties, such as the Young's modulus, can be influenced by temperature variations.

This section aims to provide an overview of thermoelastic problems, with a particular focus on static problems, while excluding thermal fatigue, which is not addressed in the applied analysis.

Thermoelastic analysis refers to the study of stress and strain induced by temperature gradients. To simplify the problem, we consider a plane strain state; analogous considerations can be extended to the three-dimensional

case. A temperature variation in isotropic elastic solids leads to deformations, as each material element tends to expand uniformly in all directions in proportion to the temperature change, following the equation:

$$\varepsilon_{x,y,z} = a\Delta T \quad (2.26)$$

In the presence of temperature variations, the first three constitutive equations are modified by adding the term from Equation 2.26:

$$\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \end{bmatrix} = \begin{bmatrix} 1 & -\nu & -\nu \\ -\nu & 1 & 1 \\ -\nu & -\nu & 1 \end{bmatrix} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \end{bmatrix} + a\Delta T \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (2.27)$$

σ_z has an additional term:

$$\sigma_z = \nu(\sigma_x + \sigma_y) - E a\Delta T \quad (2.28)$$

Finally, the congruence equation becomes:

$$\nabla_2(\sigma_x + \sigma_y) = E a \nabla_2(\Delta T) \quad (2.29)$$

Equation (2.29) shows that if the body is free from constraints or constrained in an isostatic manner, a uniform temperature variation ($\Delta T = \text{constant}$) or a variation that creates a linear temperature gradient ($\Delta T = cx + c_1$) does not give rise to a stress state. In fact, in this case, the term on the right-hand side of the equation cancels out.

Regarding thermo-elastic analyses, the problem is obviously twofold. On one side, it is necessary to calculate the temperature distribution, which becomes the boundary condition for structural calculations. For both physical phenomena, numerical or analytical approaches can be used to evaluate the quantities of interest. It is clear that analytical methods have the advantage of being very fast, but the main limitation is that they can only be applied to a relatively narrow range of geometries. On the other hand, numerical methods offer more flexibility.

On the thermal side, the problem to be solved is related to evaluating the temperature distribution. Generally, there are two scenarios:

- The temperatures at the material interfaces are known, and it is sufficient to evaluate the temperature distribution through the thickness. In this case, complications may arise from strongly non-homogeneous internal structures.
- The thermal fluxes exchanged with the environment are evaluated, considering conduction, convection, and radiation

Analytical approaches to thermoelastic problems frequently start with simplified models that capture essential behaviors using mathematical solutions to governing equations, such as those of heat conduction. Among these, the finite difference method (FDM) is a widely used technique for solving partial differential equations (PDEs) related to thermal analyses (34) (35) (36).

In the context of thermal problems, FDM discretizes the continuous domain into a grid of nodes, representing spatial variables and time intervals, enabling approximate solutions of the heat conduction equation. The governing equation for heat conduction in cylindrical coordinates, for example, is often employed in thermal analyses of components such as pipes, wheels, and other axisymmetric structures. These models rely on the general heat conduction equation in cylindrical coordinates, using explicit Euler integration for temporal discretization and second-order accuracy for spatial discretization. The explicit Euler method is a simple, first-order accurate scheme that calculates the temperature at the next time step T_{n+1} based on the current temperature T_n and a temperature change ΔT_n as described by:

$$T_{n+1} = T_n + \Delta T_n \quad (2.30)$$

$$\frac{\Delta T_n}{\Delta t} = MT_n + b_n \quad (2.31)$$

where Δt is the time step, n denotes the time index, M is the system matrix, and b_n is the forcing term (representing external heat inputs). The system matrix and forcing term are derived from the heat conduction equations applied at each node of the computational domain. In these models, the temperature field is organized

in vectors representing the distribution over nodes, with indexing based on radial, tangential, or axial coordinates. This method's strength lies in its simplicity and efficiency, making it particularly useful for problems with regular geometries and relatively low computational requirements.

Numerical approaches (37) (38), on the other hand, primarily include the CFD-FEM coupling, which enable the study of complex problems with generic geometries and boundary conditions. Through FEM, it is possible to simulate the behavior of three-dimensional components subjected to thermal and mechanical loads, accounting for nonlinear material properties and contact effects. In this context, the ability to couple thermal and mechanical models is essential to capture the overall thermoelastic response of the system.

Effective exchange of information between these domains is crucial to achieve reliable and realistic results. Material properties such as the elastic modulus, thermal expansion coefficient, and thermal conductivity can vary with temperature. Accurately modeling these properties is essential to correctly predict the thermoelastic response of a system. In thermoelastic analyses, a one-way approach can be adopted, wherein the thermal analysis provides the temperature field used in the structural analysis, or a two-way approach, which involves iterative interaction between the two domains. While the latter is more accurate, it can be computationally expensive. FEM can accommodate complex geometries, boundary conditions, and material properties, including nonlinearities that arise from temperature-dependent material behavior. High-fidelity FEM simulations offer precise results by capturing detailed interactions between thermal and mechanical fields, making it invaluable for engineering applications where accuracy is paramount, such as aerospace structures and high-temperature components.

The accuracy of thermoelastic analyses must be balanced with computational efficiency, especially for large-scale problems or those with tight time constraints. Techniques such as ROM and optimization algorithms can be employed to reduce computation times while maintaining high precision.

2.2.2. Multi-fidelity Analysis

Multi-fidelity methods in engineering are increasingly recognized as essential tools for addressing complex problems that require a balance between computational accuracy and cost efficiency.

Multi-fidelity approaches enable the connection of models with varying levels of accuracy. The objectives of these approaches can be manifold:

- To model in detail only specific areas of interest (sub-modelling, homogenization).
- To connect different physical phenomena that require varying degrees of accuracy (39).
- To link low-fidelity tools and analyses for preliminary design with high-fidelity numerical tools. For instance, combining analytical and numerical approaches to optimize the design process.

In this thesis, these approaches have been evaluated with a particular focus on the third point, which has been explored in greater detail in the industrial examples section.

2.2.2.1. Sub-modelling and homogenization

Sub-modelling (40) and homogenization (41) are two techniques often used simultaneously in FEM.

The idea behind homogenization is to simplify a highly heterogeneous structure by considering an equivalent but homogeneous material. This technique is widely applied to materials like composites (42) or metamaterials. For large structures, it would be unfeasible to model in detail the internal structure of composite fibres or the internal structure of a metamaterial. Homogenization techniques strongly depend on the problem being studied. For structural applications, the homogenized material must have equivalent structural properties. Remaining within the FEM framework one of the most used workflow is the Representative Volume Element (RVE) Method. This method uses a representative volume element that captures the microscopic properties of the heterogeneous material. The equivalent homogeneous material is characterized through experimental or numerical uniaxial load tests (43). Other papers suggest an energetic approach for estimating the equivalent properties (44).

These approaches greatly simplify the problem, making it possible to use much coarser meshes. However, the main drawback is that internal structure and stress concentrations are lost in detail. This is where sub-modelling

is used. The idea is to create a detailed model of only the area of interest (e.g., the most critical zones) by importing the boundary conditions from the entire model at the interface areas.

The submodel method is an effective approach for local stress analysis of large, complex structures and has been successfully applied in several engineering cases. For example, the submodel method has been used for the double-lap composite adhesive-bonded joint (45), bonded joints in ship structures (46), tokamak structural design (47) and other complex structures.

The basic steps of the submodel method are as follows:

- Performing a FE whole model analysis: The entire FE model should be meshed with coarse grids, ignoring local structural details. The displacement response of a specific section (boundary of the accurate FE submodel) should then be calculated based on the whole FE model.
- Building the accurate FE submodel: The accurate FE submodel should be constructed according to the dimensions and structures of the substructures, as well as the requirements of the analysis objectives. Appropriate elements should be chosen to simulate the FE submodel. Clearly, the grid density of the FE submodel is higher compared to the entire FE model.
- Calculating the boundary displacement of the accurate FE submodel: The displacement response of the specific section of the whole FE model should be automatically applied to the corresponding boundary of the accurate FE submodel. Linear interpolation should be used if there is no corresponding displacement from the whole model. This is a crucial step in the submodel method.
- Submodel analysis: Keep the boundary conditions and loads originally present in the FE submodel unchanged, and apply the corresponding boundary displacements from the whole model to carry out the submodel analysis.

2.2.2.2. Low-fidelity and high-fidelity tool integration

The integration of high-fidelity and low-fidelity tools is crucial for several reasons. First, in the generic workflow of designing complex machines or structures, the starting point is always the adoption of low-fidelity models. These are simplified models that capture the involved physics with approximations but in a very fast way. Such approaches typically allow for the development of a preliminary design, and various parametric approaches can be used to optimize the low-fidelity model.

However, low-fidelity approaches often have limitations and are sufficiently accurate only for very simple components. In some applications, higher precision is required, or it is insufficient to monitor an output parameter alone. Instead, it becomes necessary to monitor a field variable with greater accuracy. In these cases, high fidelity numerical methods (e.g., CFD, FEM) are generally used. The connection between these two domains remains a significant challenge.

The key aspect is to establish a common parameterization. Physical parameters can be easily managed in high-fidelity tools as boundary conditions. Geometric parameters, however, present a more complex issue. In high-fidelity models, there are generally two approaches to defining geometric parameterization:

- Parametric CAD: This involves creating edges, surfaces, and volumes of the reference geometry while parameterizing the dimensions of interest (e.g., radii, lengths, etc.).
- Mesh Morphing: This involves introducing parameterization directly into the mesh.

These two approaches will be explored in greater detail in the following section. In the industrial applications section, an example is provided to illustrate how to link analytical and numerical approaches for a structural optimization problem (section 4). Both techniques—parametric CAD and mesh morphing—have been explored. The proposed approach combines numerical and analytical optimization methods. The idea is to use the analytical approach to obtain an already optimized baseline for numerical optimization. This enables an optimized solution with fewer analyses while maintaining good accuracy. High fidelity methods, in fact, have the advantage of handling complex geometries with higher precision.

2.3. Shape parametrization and Automatic workflows for Dataset Creation and Optimization

In recent decades, computational capabilities have grown significantly, driven by the need to perform increasingly accurate and complex analyses. The availability of High-Performance Computing (HPC) infrastructure and the enhanced reliability of high-fidelity simulation software have dramatically reduced the time required to generate, process, and store large volumes of data.

The evolution of HPC (48) has revolutionized numerical analysis, enabling faster and more precise simulations. Parallel computing, a cornerstone of modern HPC, maximizes computational resource utilization by distributing workloads across multiple processors. In particular, Graphics Processing Units (GPUs) (49) have demonstrated substantial performance advantages due to their ability to execute a large number of operations in parallel. Unlike traditional Central Processing Units (CPUs), which are optimized for sequential instruction execution, GPUs excel at handling thousands of simultaneous operations, making them especially effective for computationally intensive tasks such as CFD (Computational Fluid Dynamics) simulations. The integration of GPUs into HPC systems has expanded the boundaries of numerical simulations, allowing increasingly complex problems to be solved with greater accuracy in significantly less time and with a substantial reduction in energy consumption.

The advancement of HPC has driven two major trends: first, it has enabled increasingly detailed and precise analyses (e.g., denser computational meshes, Large Eddy Simulation (LES) analyses) (50) (51). Second, it has facilitated the generation of large datasets required for optimization algorithms within reasonable timeframes. However, it is clear that this computational power's potential is significantly limited without automated methods for dataset generation.

Substantial progress has been made in the automation of CAD generation and re-meshing (52), as well as in mesh morphing techniques (53). These developments highlight the critical importance of automating the creation process for each design point, which is essential for leveraging HPC capabilities to their fullest extent.

2.3.1. Shape Parametrization Methods

As discussed in the previous sections, two main approaches can be used in numerical simulations to define shape parameters: parametric CAD and mesh morphing. The choice between these methods depends on the specific requirements of the application. In some cases, parametric CAD is more advantageous, while in others, mesh morphing offers better results.

Parametric CAD provides greater control over shape variations by directly modifying the mathematical representation of edges and surfaces. This approach allows for precise and systematic adjustments to the geometry. However, for each new parametric variation, a new computational mesh must be generated. This step is time-consuming, labor-intensive, and requires meticulous verification, making it challenging to fully automate. In contrast, mesh morphing is generally faster, more flexible, and easier to automate. Instead of creating a new mesh for each variation, the existing mesh is deformed to accommodate the changes. This eliminates the need for remeshing, significantly reducing computational noise, processing time, and manual intervention. As a result, mesh morphing is highly suitable for workflows that require rapid iteration and automation.

However, mesh morphing has limitations. It may not be viable for large parametric variations, as excessive deformation can degrade the mesh quality. Despite this drawback, mesh morphing preserves the original mesh topology, which is critical in certain optimization methods, such as adjoint optimization, or when creating reduced order models.

2.3.1.1. Parametric Cad

Since the early 1960s (54), the evolution of CAD systems has significantly transformed the manufacturing of mechanical devices. Initially, CAD, an acronym for Computer-Aided Drafting, referred to systems that allowed users to create drawings on a computer screen with pointing devices, mimicking traditional pencil-and-paper drafting. The primary advantage of these early systems was the precision they offered—straight lines, perfect circles, easy modifications, and error correction. However, these systems were limited to two-dimensional representations, requiring users to mentally translate these into three-dimensional concepts.

The next phase introduced Computer-Aided Drawing, which supported three-dimensional geometry. This advancement enabled drafters to generate multiple orthographic projections of an object, but it required the specification of three-dimensional coordinates, typically achieved by defining a sketching plane. Despite improvements, these systems could only produce wireframes, often leading to impractical models due to their lack of solid geometry understanding.

Subsequently, the acronym CAD came to represent Computer-Aided Design. This new generation introduced solid modelling, where objects were created by starting with a solid body and manipulating it through operations like adding, subtracting, or morphing material. The inclusion of parameters in these systems further allowed for easy modifications of dimensions, making models more flexible and enabling the automation of processes like tool-path generation, welding plans, and mold tool creation. These systems, tailored for manufacturing processes, are often categorized as mCAD systems (55).

Efficient communication between CAD systems and analysis tools is critical, particularly in a Multi-Disciplinary Analysis and Optimization (MDAO) context (56). Initially, data was exchanged using the IGES file format, which only contained geometric information, such as disjoint surfaces and curves, without any topological data. This limitation necessitated significant manual effort to ensure closed, watertight models for 3D meshing.

A more robust alternative is the STEP file format, which includes topology in addition to geometry, supporting the creation of Boundary Representations (BRep). BRep defines a model by combining geometric and topological entities to create a connected collection of surfaces and curves. These entities include:

- Nodes: Points in 3D space.
- Edges: Lines bounded by two nodes.
- Loops: Ordered collections of edges, potentially associated with surfaces.
- Faces: Surface segments bounded by loops.
- Shells: Collections of oriented faces.
- Bodies: Functional aggregates of entities.

BRep's hierarchical structure ensures unambiguous definitions. These representations encapsulate the hierarchy and connectivity of geometric components, making them preferable for comprehensive CAD data transfer.

Generally, there are two approaches to building a CAD model: bottom-up and top-down.

- Bottom-up approach: In this method, the construction begins with the most fundamental elements, such as points (nodes) and edges. Each component is incrementally built upon the previous one, with nodes serving as the starting point for edges, edges forming loops, loops defining faces, and so on. This hierarchical process ensures precision and unambiguous definitions, as required in Boundary Representation (BRep). However, it is labour-intensive and time-consuming, as each component must be explicitly defined.
- Top-down approach: This method focuses on defining the overall structure or intent of the model first, typically using parametric or feature-based design principles. Starting with a conceptual framework, designers define parameters and apply high-level operations like extrusions, revolutions, and boolean operations. This approach allows for greater flexibility, faster iterations, and easier modifications since changes in parameters automatically propagate through the model.

Parametric CAD systems go beyond static geometric definitions by using a master model that incorporates both a feature tree (build recipe) and design parameters. These systems store design intent, defining the design space and enabling iterative optimization in MDAO processes. However, commercial CAD systems typically use proprietary file formats, complicating external access and modification. Constructive Solid Geometry (CSG) is often considered a foundational approach to achieving these objectives, though its implementation demands a solid-modelling geometry kernel. To resolve this problem, OpenCASCADE (57) is a prominent open-source solid-modelling geometry kernel offering extensive capabilities, such as support for manifold and non-manifold geometry, bottom-up construction, CSG operations, and compatibility with IGES and STEP file formats. However, its steep learning curve, complexity, and limited documentation present challenges for new users. ACIS and Parasolid are the commercial standards

Parametric CAD systems leverage CSG principles to create complex models from simple shapes. Core features include:

- Solid Creation: Generation of basic shapes like boxes, spheres, and cylinders.
- Boolean Operations: Union, intersection, and subtraction of solids.
- Transformations: Operations such as revolve, extrude, sweep, and loft.

2.3.1.2. Mesh morphing

Mesh morphing is a technique that enables shape modification while preserving the original topology of the grid. This approach adjusts only the nodal positions, eliminating the need to recreate the CAD model and its associated mesh. As a result, parameterization is defined directly at the mesh level.

An ideal mesh-morphing algorithm should exhibit the following characteristics:

- The modified grid must retain its original topology.
- It should offer a high degree of expressiveness, ensuring a simple and intuitive interaction between the user and the graphical interface, allowing the parameter set to be easily defined.
- Boundary conditions and constraints must be respected.
- The algorithm should allow for the optimization of geometry through parameter adjustments, even in the case of complex baseline models.
- It must ensure a high-quality deformed mesh.
- Efficiency is critical, meaning the algorithm must be applicable to industrial-scale meshes with reasonable computational power and time requirements.

Among the most commonly used methods, the RBF approach has been explored in depth, as it is employed in this thesis work.

RBF Mesh Morphing

The use of RBFs is widely adopted in mesh morphing problems. More details on the mathematics of RBFs are provided in Section 2.1. The idea is to define a set of source points and apply known displacements to them. The displacement at the mesh nodes is then computed proportionally to their distance from the adjacent source points.

Fixed source points can be defined to lock a specific region, or displacements can be assigned to deform the mesh in a targeted area. Once the RBF solution associated with a given set of source points is computed, it can be linearly scaled. In this way, the mesh is parameterized.

With this approach, RBFs allow for preserving mesh quality, enforcing any required constraints, and, with appropriate parallelization techniques (described in Section 2.1), can be efficiently scaled for industrial-scale problems.

In Figure 5, an example is shown. In this case, the mesh nodes are directly selected as source points, but in general, they do not have to coincide. In the given example, the red points are fixed, while a horizontal displacement is applied to the green points. The final result is the deformed cube shown on the right. By linearly amplifying the base solution, it is possible to obtain parametric variations of the initial mesh.

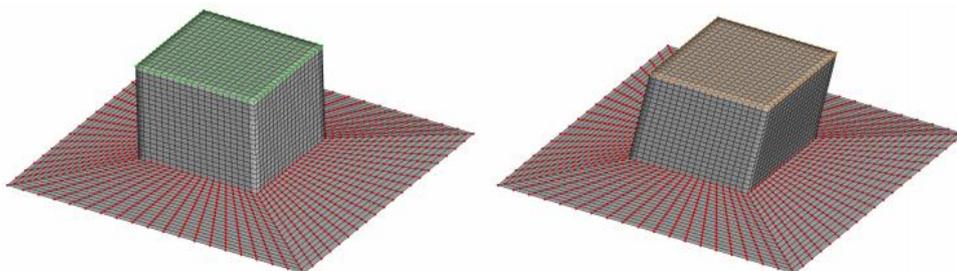


Figure 5: Example of RBF Mesh Morphing. Red points are fixed, green points are moved to the orange points.

In Figure 6, another example is shown of how RBF mesh morphing can be used to parameterize the mesh. In this case, the source points are located outside the mesh. This approach is more flexible and faster since source points can be defined anywhere, and being independent of the mesh, it allows for a parametric variation using a very limited number of points.

In this case, the box at the outlet of the air intake is fixed, meaning that a displacement of zero is applied to all source points in that region. The points at the base of the intake are also fixed, while the points of the cylinder are moved downward.

At the bottom, a preview of the mesh deformation associated with this parameterization is shown. As in the previous case, the RBF coefficients need to be computed only once, and the RBF solution can be linearly amplified to generate different parametric variations.

This technique was applied in sections 6.2 and 6.3.

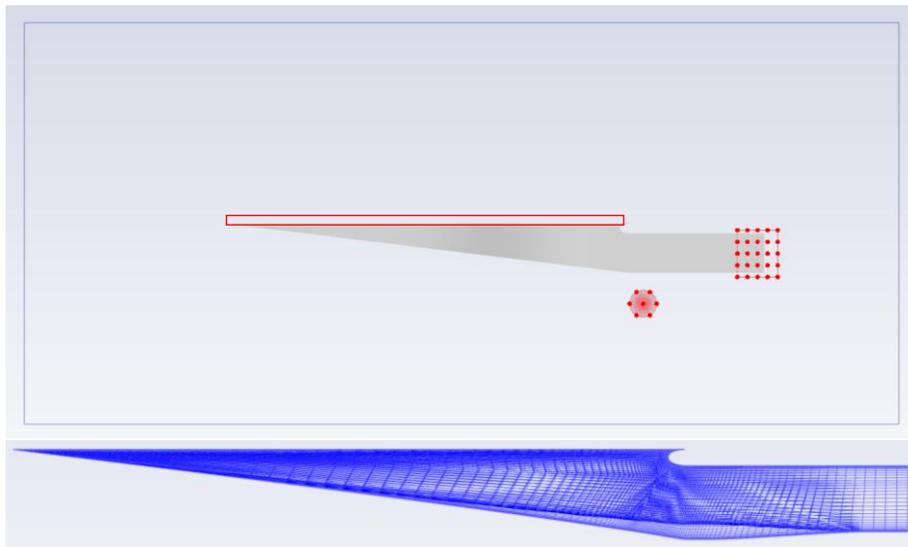


Figure 6: Example of RBF Mesh morphing. Upper, the box is fixed and is used to fix the outlet. A downward displacement is applied to the points of the cylinder. On the right, the preview shows how the mesh is deformed. The dark blue is the baseline, and the light blue is the deformed mesh

2.3.1.3. Hybrid Methods

This section presents two hybrid methods. The first was applied to the OPAM case study (Section 6.5), and the second to define the parameters in the aeroSUV testcase (Section 5.2).

OPAM Method

Previous sections mentioned the limitations and the merits of CAD based and mesh based shape parameterisation approaches. In this thesis, a novel hybrid method is proposed to link CAD and mesh for the generation of each dataset to get at the same time the key advantages of CAD parameterisation and mesh morphing. The idea is to use a parametric CAD and a method for transferring shape modification information from the CAD level to the mesh level, combining the advantages of mesh morphing and parametric CAD. Specifically, the parameters are defined in a Python script that generates the CAD model. Each CAD variant is then compared with the baseline, creating a point cloud that spans from the initial state to the new CAD design. This point cloud is used as an RBF field to transfer information from the CAD level to the mesh level. Finally, the mesh is deformed, and a simulation is conducted for each DP. Figure 7 illustrates the workflow followed in this study.

To generate the baseline mesh, a convergence analysis is necessary, progressively refining the mesh to find a balance between very dense meshes, which are computationally slow, and coarser meshes. This starting point remains crucial for the workflow. Once the baseline mesh is generated, the other DPs morph the already created mesh, so it is only necessary to monitor mesh quality to avoid highly distorted cells. From this perspective, RBFs provide an excellent solution, as they help preserve mesh quality during the deformation process.

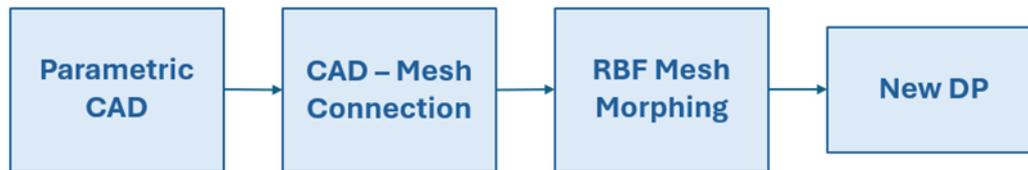


Figure 7: Hybrid workflow

The proposed method combines the advantages of both previously described approaches. Specifically, CAD-based parameterization is retained, which provides excellent control over the final shape, while mesh morphing is employed to update the baseline mesh to the desired variant. This hybrid workflow not only presents an appealing alternative to more established methods but also enables the creation of ROMs from a parametric CAD model. This, in turn, facilitates the transfer of high-fidelity simulation data to CAD editors, allowing for the development of smarter design platforms and significantly reducing the time required for design.

In the section 6.5, this workflow was implemented on the Open Parametric Aircraft Model (OPAM). However, the method is general and can be extended to any model, not just those in the aeronautical domain.

The process begins with a parametric CAD model, and the workflow can be applied to any iso-topological parametric CAD model. For the generation of parametric CAD, the open-source CAD editor ESP was used, though the workflow is highly flexible and easily customizable for integration with various other CAD editors. Trials have also been conducted using different tools, including CadQuery, JPAD (58), AxCent (59) and Rhino (60).

One of the critical steps in the workflow is transferring information from the CAD model to the mesh. This requires an algorithm that encodes the shape variations defined in the geometry, making it usable for mesh deformation. To facilitate this, a tool based on the OpenCascade geometrical modeler was developed to create a point cloud on the baseline geometry surface and a corresponding set of points on the CAD variant. Several challenges arise in this process. A key aspect is the recognition of homologous surfaces and edges. Using scriptable CAD editors provides a significant advantage here. The same operations are performed sequentially by changing the input parameters, ensuring that the generated surfaces and edges maintain a consistent order. For the successful coupling of homologous entities between different CADs, it is essential that the CADs are iso-topological so that homologous points on homologous surfaces and edges can be defined. This approach builds upon the method proposed in (61), wherein the surfaces of the baseline geometry are covered with a regular grid of points in the two-dimensional U-V space. Similarly, the edges are covered with one-dimensional points spaced evenly. The final position of the points on the edges is determined by prescribing the same parametric positions; the final position of the points within the surfaces is obtained by transforming the U-V parametric position of each node according to a 2D RBF deformation defined by the surface parametric position of the points on the edges in both the original and deformed shape. Since the number of points remains the same in both point clouds, the grids can be put into one-to-one correspondence. Additionally, managing the interface zone between two surfaces is crucial. To ensure good mesh quality and proper overlap of the deformed mesh with the corresponding geometry, points on edges and surfaces are independently defined, and a buffer zone is introduced between them. Surface interior points within a certain buffer distance from the edges are removed. The interior points' spacing is kept constant in both the U and V directions, calculated at the centre of each surface and imposing specific Δu and Δv values to achieve the desired spacing in 3D space. This approach does not require an external meshing tool and allows for the generation of high-quality, uniformly spaced clouds suitable for 3D RBF mesh morphing.

Once the two point clouds are generated, they can be used to drive surface mesh morphing by adopting the points on the baseline geometry as source points and the difference between the updated cloud and the baseline as the prescribed field. The point cloud is generated only on the surfaces of the CAD models, and its density can be controlled by setting the spacing between adjacent points. This approach is summarized in Figure 8, where a CAD variation of a simple wing is shown, in Figure 9, where the CAD-based source points extracted from the edges and surfaces are depicted, in Figure 10 (top), where the two final clouds used for surface morphing are shown, and in Figure 10 (bottom), where the deformed mesh is represented. The resulting surface mesh, which matches the

target B-rep shape, is shown in Figure 11, where the morphed mesh is superimposed on the CAD, and the projection of its points onto the CAD is displayed.

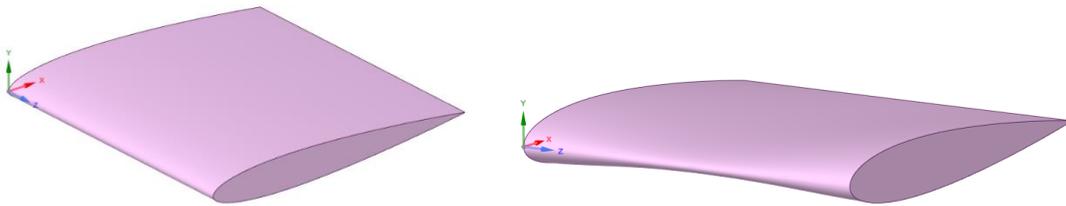


Figure 8: Example of NACA profiles (baseline-left and deformed-right)

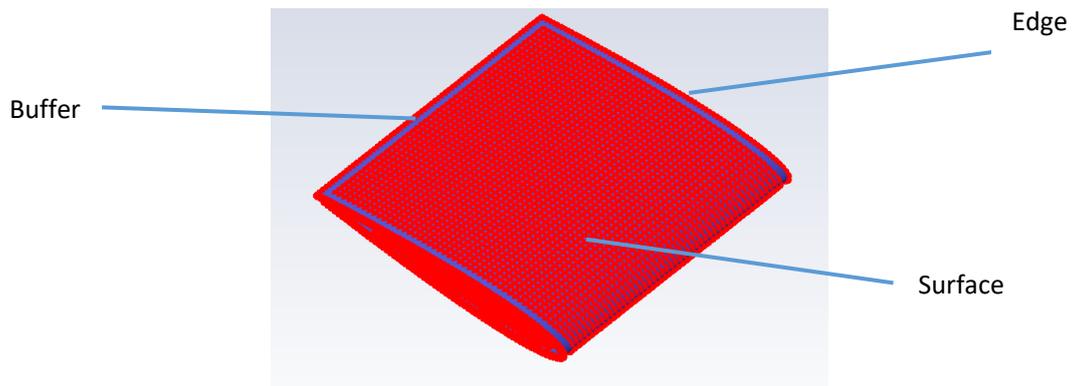


Figure 9: Cloud of points generated on the baseline

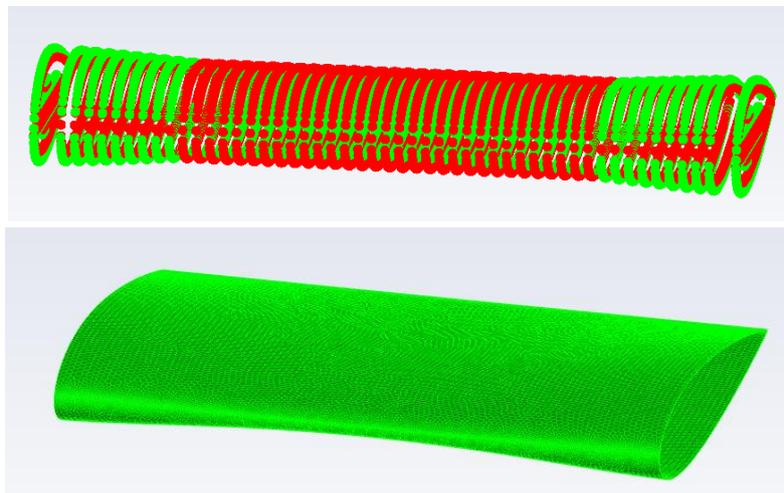


Figure 10: Comparison of points on the baseline (red) and on the deformed geometry (green) (top). Deformed mesh controlled by the source points (bottom).

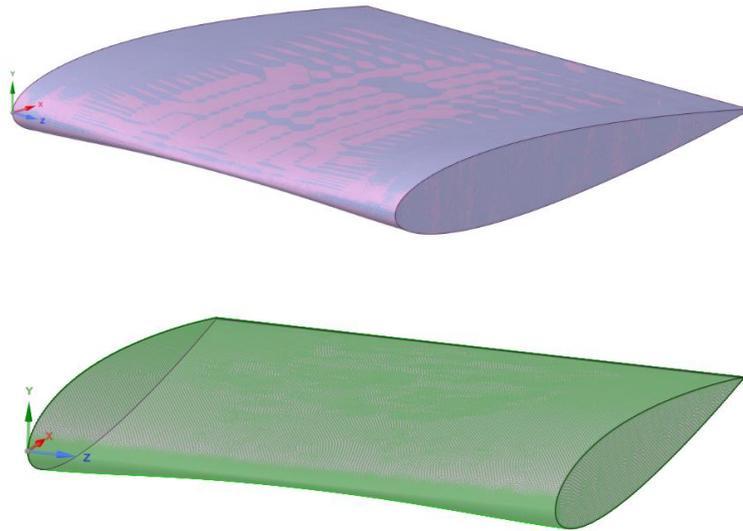


Figure 11: Comparison of deformed mesh and updated CAD (top) .Comparison of deformed mesh and updated CAD, green points have deviation under 0,01 mm (bottom).

In order to complete the proposed workflow we need to propagate the shape variation known at surfaces into the volume. To warp the full volume where the CFD mesh is defined, the mesh morphing technique based on RBFs was utilized to update each DP. Specifically, the point clouds computed at surfaces were translated into an RBF field which is used as source points in an RBF problem. This process effectively transfers the parameterization defined at the CAD level to the mesh level. It's worth considering that with respect to the workflow proposed in (62), where all the nodes of the CFD mesh are used to control the morphed shape by combining simple operations, the new proposed approach relies on a smaller cloud of uniformly spaced RBF points. This allows to use the RBF cloud as a space warper with a point spacing intended to match the geometrical details rather than the flow solution, as in the CFD mesh. It's well known that a regular cloud allows to get smoother shapes. This concept can be better understood considering the RBF cloud as a companion spline of the geometrical model rather than of the CAE one. Once this accuracy is guaranteed, the ability of the RBF to smoothly interpolate the data allows to control both a fine spaced CFD mesh and a coarse one as the final geometry . The effectiveness can be easily verified by projecting the deformed mesh onto the corresponding CAD to assess the achieved tolerance. For the example case shown, Figure11 illustrates how all points of the deformed mesh, projected onto the deformed CAD, have a deviation smaller than the tolerance (0.01 mm), and therefore are displayed as green points.

AeroSUV Method

The second hybrid method proposed was applied to define the shape parameters for the aeroSUV testcase (Section 5.2). In this case, a tool was developed to define a support mesh. The support mesh is used to define source points (nodes of the support mesh). Displacements are defined on these source points. In this way, there is a direct correlation between the CAD elements (edges and surfaces) and the source points, allowing the direct manipulation of the entities of interest. By using a coarser mesh, the number of source points is significantly reduced, which results in a smaller RBF problem.

To go into more detail on the developed workflow, the first step was to define a wrapper that envelops the defined geometry (Figure 12). This way, the CAD and, consequently, the support mesh are sufficiently general to potentially be extended to any SUV.

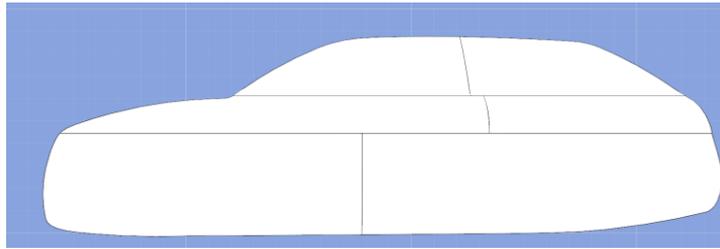


Figure 12: Wrapper CAD di aeroSUV

A support mesh was built on the wrapper. The mesh is not used for any CFD analysis; it is only used to obtain source points that are correlated with the CAD entity. The support mesh is shown in Figure 13. In particular, by using a relatively coarse mesh, the number of source points will be smaller, thereby reducing the time required for the RBF solution.

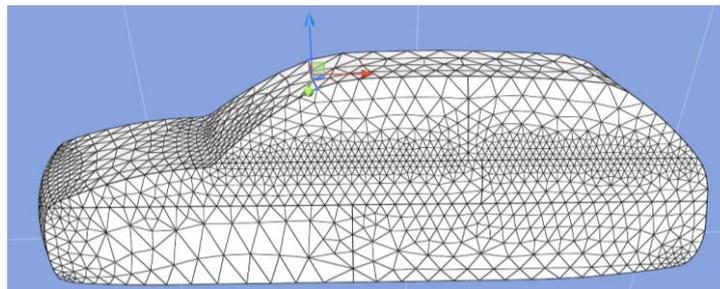


Figure 13: Support mesh

Once the support mesh is defined, displacements can be specified on some of the source points. The advantage of using this approach is that CAD entities such as edges and surfaces are indirectly controlled, but with a very limited number of source points. The use of the wrapper allows the parameterization to be generalized theoretically for any SUV model. Specifically, displacements were defined on edges 1, 2, and 3 in Figure 14, while some surfaces (blue points in Figure 14) are fixed.

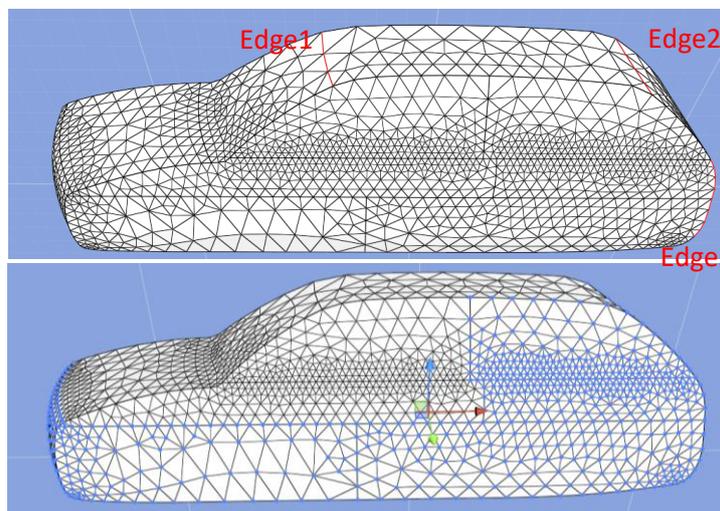


Figure 14: Selected edges (upper), and fixed points (below)

As with the previously described cases, once source points and displacements are defined, the parameterization can easily be transferred to the mesh in detail and can be amplified parametrically for exploration of the parameter space. To run simulations in an HPC environment, it is sufficient to transfer the files with the source points and displacements. Given n parameters, there is no need for an n -tuple of files for each design point (DP); instead, the n files with the original displacements will be sufficient. After that, the RBF solutions associated with each parameter can be amplified and linearly combined.

In Figure 15, previews are shown for the displacement of each edge in the indicated directions.

In Section 5.2, however, this parameterization is not linked to a classic DOE but rather to a one-shot optimization with the adjoint (Section 2.3.2.4), so no further CFD analyses are required.

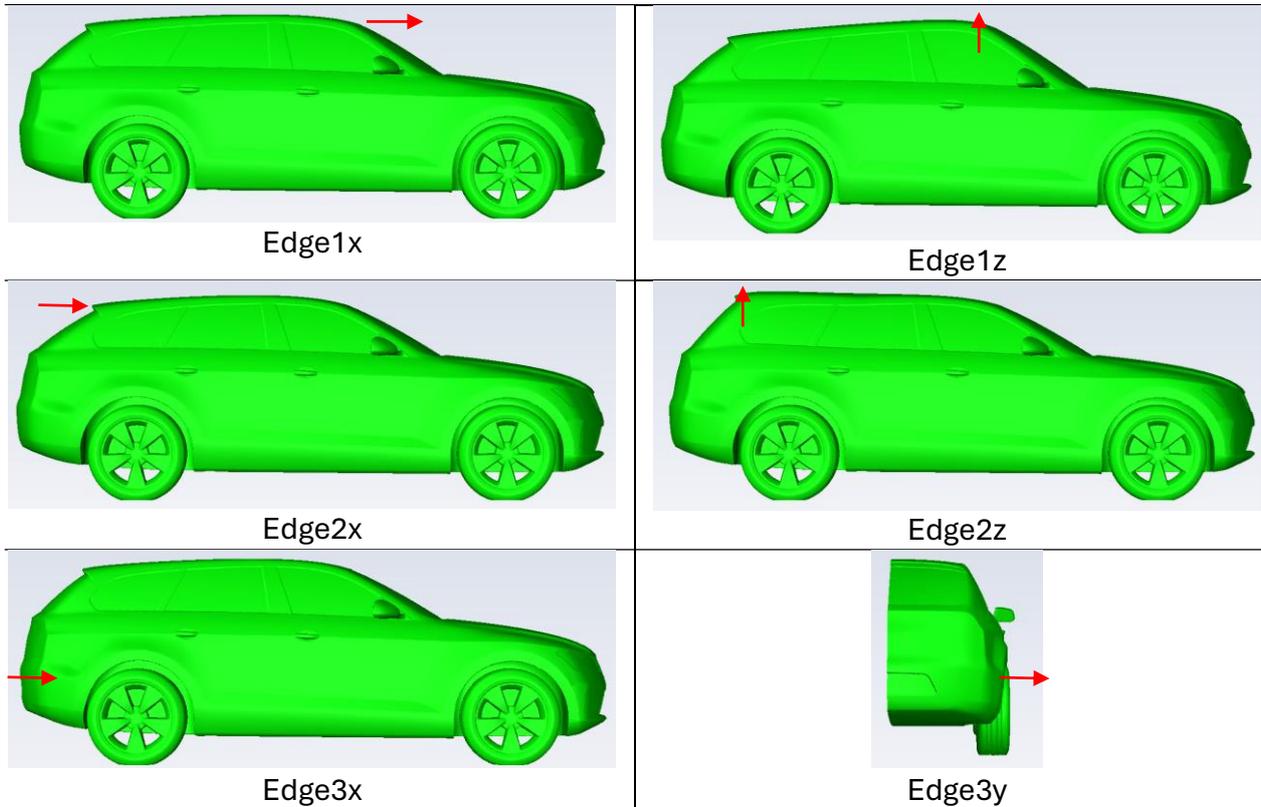


Figure 15: Morphing of the detailed mesh for each shape parameter

2.3.2. Optimization methods

Optimization methods boast a long tradition, with a history spanning over two centuries. The earliest studies in this field date back to the late 1600s, carried out by mathematicians, physicists, and scientists of the calibre of Bernoulli, Euler, Newton, Leibniz, Gauss, and Weierstrass. Since the mid-20th century, the advent of computers has enabled the evolution of optimization algorithms, enhancing existing methods and developing new ones that leverage the increased computational power. Today, optimization algorithms remain central, applied across nearly all branches of engineering.

This thesis focuses specifically on shape optimization methods. Optimal Shape Design (OSD) encompasses the set of methods, algorithms, and procedures aimed at optimizing a shape. The optimization objective depends on the application and is defined by the objective function to be minimized or maximized.

In recent years, advances in computational power and the emergence of new manufacturing techniques have made shape optimization even more relevant. For example, additive manufacturing methods allow the creation of highly complex shapes, increasing the range of design variations and thereby expanding the problem's degrees of freedom. Testing such a wide range of possibilities solely through experimental methods would be impractical, requiring excessive time and costs to produce the necessary prototypes. In this context, the use of computers and automated methods becomes essential. These tools allow the consideration of a large number of parameters and shape variations, selecting only the most promising configurations for subsequent experimental validation.

2.3.2.1. Optimization problem definition

The first step in any optimization problem is to establish the objective to be achieved. This step is crucial as it influences all subsequent phases of the process. The objective is defined both in terms of the quantities to

optimize and the required variations. The quantity (or set of quantities) to optimize represents the objective function, which is the observable to be measured and monitored.

For example, in the context of external flow optimization, the objective function might be drag, lift, or their ratio. Additionally, a minimum required variation of the observable is often set during this phase. In many optimization problems, the goal is to improve performance relative to a reference shape. For instance, in the case of external flow optimization, the target could be a 10% reduction in drag compared to the initial geometry. In such cases, the aim is not necessarily to find a global minimum of the objective function but rather a local minimum, depending on the designer's preferences regarding computational time and performance improvement. While the global minimum represents the ideal solution, achieving a local minimum may suffice if it meets the optimization target, is simpler to manufacture, or requires significantly less computational time for only marginal improvements.

It's worth commenting that, especially when costly functions are evaluated, the optimisation ends not because an optimum has been found but because the available budget is over. The "optimal" optimisation method is the one capable to get the best result within the available budget (usually expressed in time, computational power and CAE software licenses availability).

Another critical aspect of an optimization problem is the selection of design variables, which are the parameters that can be adjusted to achieve the desired target. The choice of these variables is strongly influenced by the objective function.

It is essential to select design variables that have a significant impact on the observable, as this leads to more meaningful variations and simpler final shapes. For example, the same benefit can be achieved with fewer modifications to the initial geometry by choosing the most influential variables. Design variables can vary widely, including thickness, material properties, or kinematic properties of the simulation. Generally, these variables can be categorized as continuous, discrete, or boolean, with continuous variables being the easiest to manage during the optimization process.

Constraints are integral to any optimization problem and define the conditions that must be satisfied during the optimization process. These constraints can be geometric, design-related, or functional:

- Geometric Constraints: Ensure that the geometric properties of the object are preserved. For instance, if the initial geometry is symmetric, the optimized shape might be required to maintain this symmetry.
- Design Constraints: Often related to computational power and costs, these constraints influence the time required to achieve the specified objective.
- Functional Constraints: Guarantee that the final shape remains functional for the intended application. These constraints are frequently associated with size, weight, or performance limits.

Finally, the choice of the simulation models significantly impacts the optimization problem. Selecting the most appropriate and adequate model is essential to simulate the problem effectively and guide the optimization process toward the desired outcome.

As mentioned in the previous section, optimization refers to the process of minimizing a given function to obtain the optimal values of the design variables.

According to Weierstrass's Theorem, if a generic function $\psi(x)$ is continuous and bounded, then it must have an absolute minimum. This minimum can be identified by ensuring that the gradient of the function is zero:

$$\nabla\psi(x_0) = 0 \tag{2.32}$$

This condition alone identifies a critical point, which could be a minimum, maximum, or saddle point. To confirm that the critical point is a minimum, it is necessary to verify that the Hessian matrix of second partial derivatives is positive definite. Mathematically, this means:

$$H(\psi(x_0)) > 0 \tag{2.33}$$

For H to be positive definite, all its leading principal minors must be greater than zero. This ensures that the critical point is indeed a minimum.

In the context of constrained optimization, the primary method is the Lagrange multiplier method.

Given a constraint $g(x) = 0$, the optimization problem can be reformulated by introducing Lagrange multipliers λ . This leads to the definition of the Lagrangian function:

$$L(x, \lambda) = \psi(x) + \lambda g(x) \quad (2.34)$$

Minimizing the auxiliary function, the following system of equations is obtained, which, when solved, provides the solution to the constrained optimization problem:

$$\begin{aligned} \nabla \psi(x) + \lambda \nabla g(x) &= 0 \\ g(x) &= 0 \end{aligned} \quad (2.35)$$

There are various techniques for finding the optimum, which can first be classified based on the type of result obtained. From this perspective, optimization problems can be divided into local minimization and global minimization problems.

- $x^* \in S \subseteq \mathbb{R}^n$ is a global minimum if $f(x^*) \leq f(x), \forall x \in S$;
- $x^* \in S \subseteq \mathbb{R}^n$ is a local minimum point if there exists a neighbourhood $B(x^*, \epsilon) \cap S$ such that: $f(x^*) \leq f(x), \forall x \in B(x^*, \epsilon) \cap S$.

Global minimization involves exploring the solution over a wide domain, where several local minima or maxima might be encountered before identifying the final configuration. This approach allows for examining vastly different solutions within a large region of interest.

In contrast, local minimization methods are used when the goal is to find an improved solution that remains close to the initial configuration. Here, the domain is typically more restricted, resulting in shorter computation times. Gradient-based methods, for example, often converge to a local minimum, making them suitable for situations where the search is limited to a small region. Zero-order methods can help locate a global minimum, but these are often more computationally intensive. As noted earlier, it is not always preferable to find the global minimum. A local minimum generally requires fewer iterations and thus lower computational costs. Moreover, in some applications, a solution that is slightly less optimal but closer to the initial configuration might be preferred for technical or economic reasons.

2.3.2.2. Zero-order Methods

Zero-order methods explore potential solutions to the problem stochastically, employing strategies to select design points while minimizing computational cost. These methods are particularly effective in identifying stationary points that are not local minima. They are often employed for an initial evaluation of the entire domain or to improve a solution when gradient-based methods converge to a local minimum. However, as the number of design parameters increases, the computational cost of zero-order methods grows significantly. Examples of zero-order methods include global techniques such as response surface methods, decomposition methods, and population-based methods like genetic algorithms.

As the name suggests, response surface methods (63) involve constructing a response surface by evaluating simulation results at various points in the domain. These points are selected using appropriate algorithms. Given N parameters, there are ∞^N possible combinations, so selecting a suitable set of points is essential to reduce computational resources and time. This is achieved using a DOE (Design Of Experiment).

Through DOE, design points are selected carefully—too many points result in excessive computation times, while too few yield unreliable results. Once the design points are tested through numerical analysis, a response surface is created by solving a least-squares problem to determine the coefficients of an analytical expression. Once the analytical expression for the response surface is known, the global minimum can be found by ensuring the gradient is zero and the Hessian matrix of second partial derivatives is positive definite. Genetic algorithms (GAs) (64) are inspired by the theory of biological evolution, where a species' survival depends on its fitness relative to others. These algorithms excel in identifying globally optimal solutions. In this approach:

- Each design point represents an individual defined by its chromosomes.
- Chromosomes are combined to create new design points.
- A population is the set of all individuals, i.e., design points.

- For each design point, a numerical analysis is conducted, and a fitness value is assigned based on how well the solution matches the target. The next generation of the population is created by randomly pairing parents. The process continues until convergence to a single individual with a high match to the objective.

Genetic algorithms are particularly useful for finding global optima in problems with numerous local minima. However, their high computational cost limits their application, making them most suitable for 2D optimization problems.

2.3.2.3. Gradient-based Methods

As previously mentioned, the main limitation of zero-order methods lies in their computational cost, which increases rapidly with the number of design parameters. For a substantial number of variables, a large number of design points must be defined to comprehensively represent the problem and obtain reliable results. This issue can be mitigated by employing gradient-based algorithms—evolutionary methods that leverage the gradient of the objective function to guide the optimization process and adjust the geometry towards a better configuration. Gradient-based algorithms generally require fewer iterations to reach the minimum compared to zero-order methods, but they are prone to converging to a local minimum. Additionally, they necessitate second-order continuity of the cost functions.

Gradient-based techniques are iterative in nature. At each step, the starting position is updated based on the result of the previous calculation until the minimum of the objective function is reached. For a generic DP at iteration i its position can be expressed as:

$$x_i = x_{i-1} + k\delta_{i-1} \quad (2.36)$$

Where k is the step size and δ_{i-1} is the direction evaluated in x_{i-1} .

The Steepest Descent Method is a straightforward gradient-based optimization technique that is simple to implement. For a given function $f(x)$, the gradient vector $\nabla f(x)$ indicates the direction of the steepest ascent of the objective function. To move towards the minimum of $f(x)$, the optimization process follows the direction opposite to the gradient:

$$\delta = -\nabla f(x) = -\left[\frac{\partial f}{\partial x_1} \dots \frac{\partial f}{\partial x_n}\right]^T \quad (2.37)$$

In optimization, when defining the descent direction δ , it is common practice to refer to the normalized gradient to ensure that the magnitude of the step depends primarily on the step size:

$$\delta = -\frac{\nabla f(x)}{\|\nabla f(x)\|} \quad (2.38)$$

In reference to Figure 16 (illustrating a contour plot of the objective function):

- The gradient vector $\nabla f(x)$ is always perpendicular to the contour lines of the function $f(x)$
- This orthogonality ensures that following the direction opposite to the gradient, i.e., $-\nabla f(x)$, guides the optimization towards the minimum of the function.

At each iteration, the descent direction δ is updated by inserting the new gradient value, and the update equation (Equation 2.35) is applied until convergence, which occurs when the condition $\|x_i - x_{i-1}\| \leq \varepsilon$ is satisfied.

In this method, the choice of k , defined by the user, is very important. Indeed, k essentially determines the step size at each iteration: larger values of k allow for faster convergence with fewer iterations; however, if k is too large, the algorithm may skip over a local minimum. Generally, in many applications, a solution involves defining a variable k , which takes larger values in the initial steps to speed up convergence and smaller values in the later steps to accurately find the local minimum.

Another approach to ensure the identification of a local minimum is to introduce a check on the output value of the algorithm. Specifically, if the value of the observable is greater than the previous step, this indicates that a local minimum has been skipped. To prevent this, a simple bisection-based approach can be implemented. This method adjusts the step size by halving it until a reduction in the observable is achieved compared to the previous step, ensuring the local minimum is properly identified.

An evolution of the steepest descent method is the conjugate gradient method, in which an additional term is introduced in the definition of the search direction $\delta(x)$:

$$\delta_i = -\frac{\nabla f(x)}{\|\nabla f(x)\|} + \beta \delta_{i-1} \quad (2.39)$$

The added term takes into account the gradient evaluated in the previous step and helps accelerate convergence, reducing the number of iterations. β can be defined in several ways. The most commonly used definition is that of Fletcher (65):

$$\beta = \frac{\nabla f(x_i)^T \nabla f(x_i)}{\nabla f(x_{i-1})^T \nabla f(x_{i-1})} \quad (2.40)$$

Finally, to obtain a more accurate estimate of the optimization direction and achieve faster convergence, second-order derivatives can also be utilized. In the Newton method, $\delta(x)$ is defined as:

$$\delta = -H(f(x))^{-1} \nabla f(x) \quad (2.41)$$

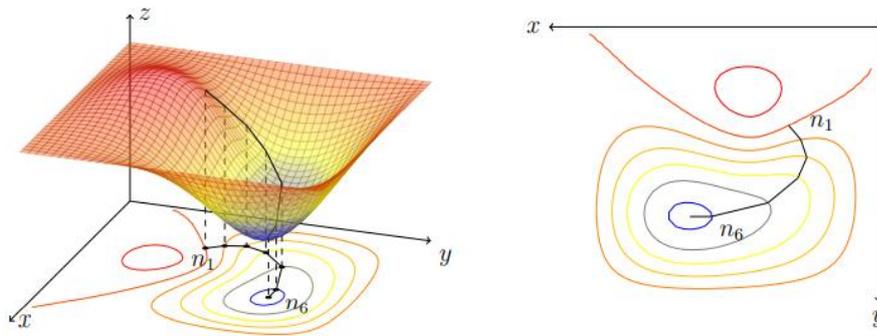


Figure 16: Gradient-based method applied on 2D problem

2.3.2.4. Parameter-less Methods

In the field of shape optimization, parameter-less methods have been playing an increasingly important role in recent years. Unlike parametric methods, these approaches are evolutionary, and at each step, the shape is updated based on the value of the target function. The significant advantage is that they are fully automated and deform the mesh, allowing the identification of new and original shapes. Among these methods, two algorithms have been particularly studied in the application part: the Adjoint method for CFD simulations and the BGM (Biological Growth Method) for the FEM ones.

Adjoint

The data and results from CFD analysis can be used to guide the optimization process and identify the direction to proceed. The most commonly used optimization methods in engineering problems are zero-order methods. As seen in the section 2.3.2.2, these methods allow for the identification of an absolute minimum and are relatively easy to implement. However, they require very high computational resources when the problem involves a significant number of parameters. In such cases, the problem can be circumvented by reducing the number of design parameters, identifying the most effective ones, or using gradient-based evolutionary methods. Unlike zero-order methods, first-order algorithms offer advantages in terms of efficiency and performance when the number of design variables is high, but they have the drawback of finding a local optimum and require sensitivity information. Sensitivity refers to the derivative of the objective function with respect to each parametric variation and can be evaluated in different ways.

In general, the derivative of a function is defined using the concept of the limit of the incremental ratio:

$$\frac{df}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (2.42)$$

This definition cannot be used in a numerical code; it must be discretized. There are several discretization methods. The simplest technique for sensitivity calculation is finite differences. The method used in this thesis is the adjoint method.

In gradient-based methods, it is crucial to evaluate the sensitivity in reduced time for the method to be considered efficient. The adjoint method allows for the sensitivity of an objective function in a problem with N parameters to be calculated with a single computation. In other words, with a single calculation, it is possible to estimate the influence of each parameter by evaluating the gradient of the observable with respect to the different parameters. This calculation must be repeated for each observable. From this perspective, the adjoint method is highly efficient and cost-effective, especially in optimization problems with a large number of parameters. In the adjoint method instead of solving a generic linear system, $Au = f$, the dual (adjoint) problem $Bv = g$, where v is the adjoint variable, is solved. It can be proven (66) that, from a logical point of view, there is no difference between solving the primal problem or its dual. The difference lies in the computational aspect. In fact, f and g , in general, have different dimensions. Let p and m represent the dimensions of the vectors f and g , respectively. It is evident that the adjoint problem is advantageous in cases where $m \ll p$, meaning when there is a large number of parameters and a small number of observables.

The adjoint method (67) can be viewed as an approach to solving constrained optimization problems. From this perspective, the method of Lagrange multipliers can be employed. Let f be the objective function, which generally depends on both physical variables x and design variables α :

$$f = f(x, \alpha).$$

Since the observable depends on both x and α , its differential can be written as follows:

$$\delta f_i = \frac{\partial f_i}{\partial x_j} \delta x_j + \frac{\partial f_i}{\partial \alpha_k} \delta \alpha_k \quad (2.43)$$

So, to evaluate the variation of the observable, it is necessary to know the variation of the fluid dynamic quantities and the design parameters. Considering the residuals of the governing equations, it must be verified that:

$$R(x, \alpha(x)) = 0.$$

This equation can also be differentiated, yielding:

$$\delta R = \left[\frac{\partial R}{\partial x} \right] \delta x + \left[\frac{\partial R}{\partial \alpha} \right] \delta \alpha = 0 \quad (2.44)$$

The problem can thus be reduced to a constrained optimization problem, where the goal is to minimize the objective function f , with the function R serving as the constraint. By introducing Lagrange multipliers and differentiating the auxiliary function, we obtain:

$$\delta f = \frac{\partial f}{\partial x} \delta x + \frac{\partial f}{\partial \alpha} \delta \alpha - \lambda^T \left\{ \left[\frac{\partial R}{\partial x} \right] \delta x + \left[\frac{\partial R}{\partial \alpha} \right] \delta \alpha \right\} = \left[\frac{\partial f}{\partial x} - \lambda^T \frac{\partial R}{\partial x} \right] \delta x + \left[\frac{\partial f}{\partial \alpha} - \lambda^T \frac{\partial R}{\partial \alpha} \right] \delta \alpha \quad (2.45)$$

λ is the added vector, and the values of its components are arbitrary. Therefore, a vector is chosen in such a way that the terms related to δx are cancelled, making the observable function dependent only on the external parameters:

$$\left[\frac{\partial R}{\partial x} \right]^T \lambda = \frac{\partial f}{\partial x} \quad (2.46)$$

Equation 2.46 is the equation of the adjoint problem. This equation depends only on the derivatives with respect to the fluid dynamics variables. The solution to the adjoint problem, therefore, allows the calculation of the adjoint variables λ . Once these variables are known, it is then possible to construct the gradient using expression 2.45, which can be reformulated as follows:

$$\delta f = G \delta \alpha, \text{ con } G = \frac{\partial f}{\partial \alpha} - \lambda^T \frac{\partial R}{\partial \alpha} \quad (2.47)$$

The expression of sensitivity is valid for each chosen parameter; in fact, the term $\delta \alpha$ represents the vector of parameter variations, which, when multiplied by G (the gradient), allows the calculation of the sensitivities of the

observable. Therefore, equation 2.47 allows the creation of the sensitivity map for each observable. Equation 2.47 can be rewritten as follows, considering N parameters:

$$\delta f = \sum_{k=1}^N G_k \delta \alpha_k \quad (2.48)$$

This expression highlights how the variation of the observable is linear, meaning that the principle of superposition applies, so the total variation of the observable is the sum of the variations obtained for each shape modification. Additionally, the variation δf is proportional to the gradient. In other words, the parameters with the highest influence are those with the highest gradient values, and conversely, those with low gradient values have less influence. Therefore, the variations of δf are linked to the gradient, which can provide a linear and local estimate of the observable's variation.

The adjoint methodology can be applied in a continuous or discrete manner. In recent years, both techniques have been developed, yielding nearly identical results despite the differing methodologies. In the continuous approach (also known as derive-then-discretize), the governing equations are derived and then discretized. Conversely, in the discrete approach (also known as discretize-then-derive), the governing equations are discretized before being differentiated by applying the adjoint technique directly to the discrete equations. Both approaches, while maintaining the characteristic of low-cost evaluation for many design parameters, present advantages and disadvantages relative to each other. The continuous formulation is considered very flexible, while the discrete method is a simpler and more easily validated approach. To overcome the issues of both the continuous and discrete methods, some authors suggest hybrid methods that combine the advantages of both.

In this thesis, a discrete adjoint solver was used, which is the method implemented in most commercial codes. The adjoint method is employed to derive the sensitivity maps for each observable. Indeed, the gradient can provide a linear and local estimate of the observable's variation and thus guide the optimization process. In other words, each sensitivity map helps understand how the observable evolves as the quantities of interest or design parameters change. Therefore, this information allows insights into the fluid dynamics of the problem and its evolution, and provides guidance for making design decisions.

A very important result of the adjoint calculation is the shape sensitivity, that is, the derivative of the observable with respect to surface displacements. By analysing this sensitivity map, regions where the gradient is higher or lower can be identified. The shape can thus be modified more in the areas where the sensitivity is higher to achieve a more significant variation in the observable for the same displacement. The great advantage of the adjoint method is that it provides the sensitivities of the observable and the optimal directions with respect to the infinite parameters of interest by solving just one problem—the adjoint one. The gradient evaluation is independent of the number of chosen design variables because, once the adjoint problem is solved, the cost to obtain the sensitivities for each parameter is negligible compared to the CFD calculation cost. The cost of solving the adjoint problem is slightly higher but comparable to a CFD analysis. This highlights the advantage of using the adjoint technique when many design parameters are involved. From this, the connection between the adjoint method and gradient-based optimization methods, such as steepest descent, becomes evident. Indeed, the adjoint method allows for evaluating the sensitivity of the objective function, and this information can be used to implement the gradient descent method, guiding the optimization process toward a local minimum. This approach is especially advantageous for problems with a large number of parameters.

Figure 17 shows the workflow used in optimization processes that employ the adjoint method. There is an initial phase where the optimization goal is set, and based on this, design parameters, constraints, and the most suitable models for CFD and adjoint simulations are chosen. Then, the fluid dynamic and adjoint problems are solved. At this point, the sensitivity can be evaluated, and based on its value, the parameter amplification is selected. Once the shape modifications and amplifications for each step are determined, morphing is performed, and the mesh is updated. The method is iterative, and the steps are repeated until the gradient is close to zero.

Thus, for each optimization cycle, solving the fluid dynamic and adjoint problems is required before modifying the mesh. This can be computationally expensive, so in some cases, it is possible to proceed by avoiding solving the adjoint problem at each step and using the sensitivity information evaluated in the first step. Obviously, this

method is less accurate and introduces errors, but in some cases, these errors do not significantly affect the predictions.

Adjoint applications are explored in the section 5.

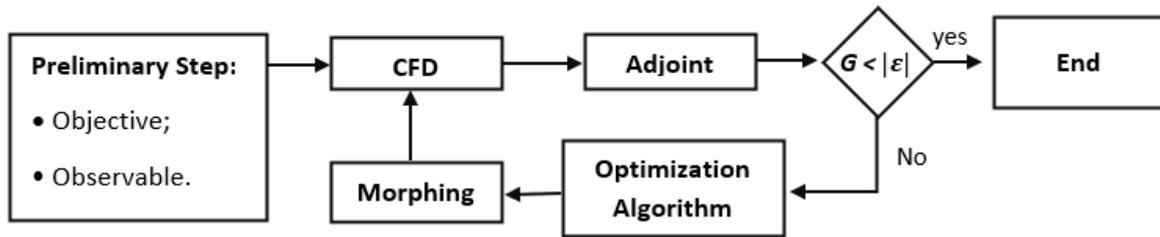


Figure 17: Adjoint workflow

Biological growth method (BGM)

The BGM is a stress-driven approach inspired by the adaptive behaviours of biological structures, aimed at optimizing structural components. Proposed by Mattheck and Burkhardt in 1990 (68) and recently extended by Porziani et al (69) (70), BGM is based on the observation that natural systems, such as bones and tree trunks, adjust their shapes in response to external loads by adding material in regions of high stress and removing it in low-stress areas. This process results in an optimized geometry with a uniform von Mises stress distribution on free surfaces.

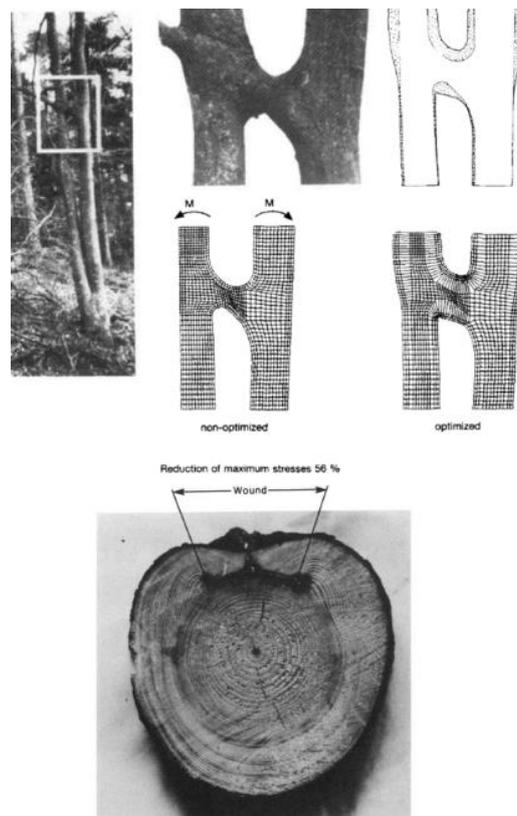


Figure 18: BGM Algorithm explained by Mattheck (74) using a tree as a reference

The growth mechanism in BGM follows a linear relationship between surface stress and material adaptation, governed by a threshold stress value (σ_{th}). Material is added or removed based on local stress levels, with the magnitude and direction of the displacement determined by the BGM stress data and limited by a maximum displacement. The modification of the surface shape is achieved through an offset technique, where node displacements are applied in the direction of the surface normal:

$$S_{node} = \frac{\sigma_{node} - \sigma_{th}}{\sigma_{max} - \sigma_{min}} d \tag{2.49}$$

Where S_{node} is the displacements applied in each node, σ_{node} is the node stress, σ_{th} is the reference stress, σ_{max} , σ_{min} are the maximum and minimum value of the stress respectively, and d is a scale parameter.

Automatic optimization integrates BGM stress data, using mesh morphing tools to adjust the structure. This allows for efficient, data-driven modifications that align with the stress distribution, resulting in improved structural performance.

An application of the method is reported in 4.1.

2.4. Big-data: Manage and transfer

In the previous chapters, various workflows for automatically creating datasets from numerical simulations were introduced. This chapter presents methods for generating reduced-order models (ROMs) capable of real-time interaction, starting from these datasets. ROMs are mathematical models designed to approximate the behaviour of complex systems with an accuracy comparable to the one of high fidelity simulations while significantly reducing computational requirements. They are particularly useful in engineering where high-fidelity simulations of physical systems can be computationally expensive and time-consuming. In fact, once large datasets are created, algorithms are needed to manage these data and make them usable in real time.

ROM methods could be grouped into two families, according to the intrusive or the non-intrusive approaches adopted. The focus of this work is on non-intrusive methods which are completely decoupled from numerical solvers. For a detailed review of the intrusive ROMs, the readers are referred to the work by Benner et al (71). Intrusive methods for ROMs are based on projecting the governing equations onto a reduced space. Advantages of Intrusive Methods for ROMs are:

- Higher accuracy – Since they derive directly from the full-order system equations, they retain the original physical and mathematical structure.
- Preservation of physical properties – They allow the conservation of fundamental system characteristics, such as energy preservation and stability.
- Reliability in complex scenarios – They are effective for well-understood systems with known dynamics, especially in engineering and scientific contexts.

Disadvantages of Intrusive Methods for ROMs are:

- High initial programming effort – They require direct manipulation of the original equations..
- Difficult application to commercial or complex codes – Accessing the internal equations of commercial or legacy models is not always possible to implement the reduction.
- Limited generalizability – An intrusive ROM is often specific to a particular configuration and may not be easily adaptable to different conditions.

An additional distinction can be made between static ROMs and dynamic ROMs. In this work, the focus is on static ROMs. Dynamic ROMs explicitly account for system evolution over time, making them suitable for transient simulations. Static ROMs, on the other hand, do not consider time evolution. They focus on steady-state solutions. In this thesis work ROMs are based on data compression techniques to create high-fidelity real-time models. Specifically, the application delves into methods based on Proper Orthogonal Decomposition (POD). The main goal is to simplify the problem by identifying the most significant features and extracting the most relevant modes, thereby facilitating easier interaction. Once these modes are identified, a surrogate model is required to represent the response of the system in the reduced space, i.e. by computing the modal weights and update the solution. Different algorithms can be used for calculating these weights and modes, as detailed in paragraph 6.

For interaction with these models, the functional mock-up interface (FMI) standard was chosen. This standard allows the export and import of the created models into major programming environments (such as MATLAB, Python, etc.), enabling the development of real-time design platforms.

2.4.1. Principal orthogonal decomposition (POD)

One of the most popular linear techniques is POD, which is also known as Karhunen–Loève procedure, principal component analysis (PCA), Hotelling analysis, empirical component analysis, quasi-harmonic modes, and empirical eigenfunction decomposition in different fields (72) (73). POD is a dimensionality reduction technique used to analyse complex data and identify the main modes of variation. There are various methods to perform POD, each with its own characteristics and specific applications. Figure 19 illustrates the principle on which the POD method is based on.

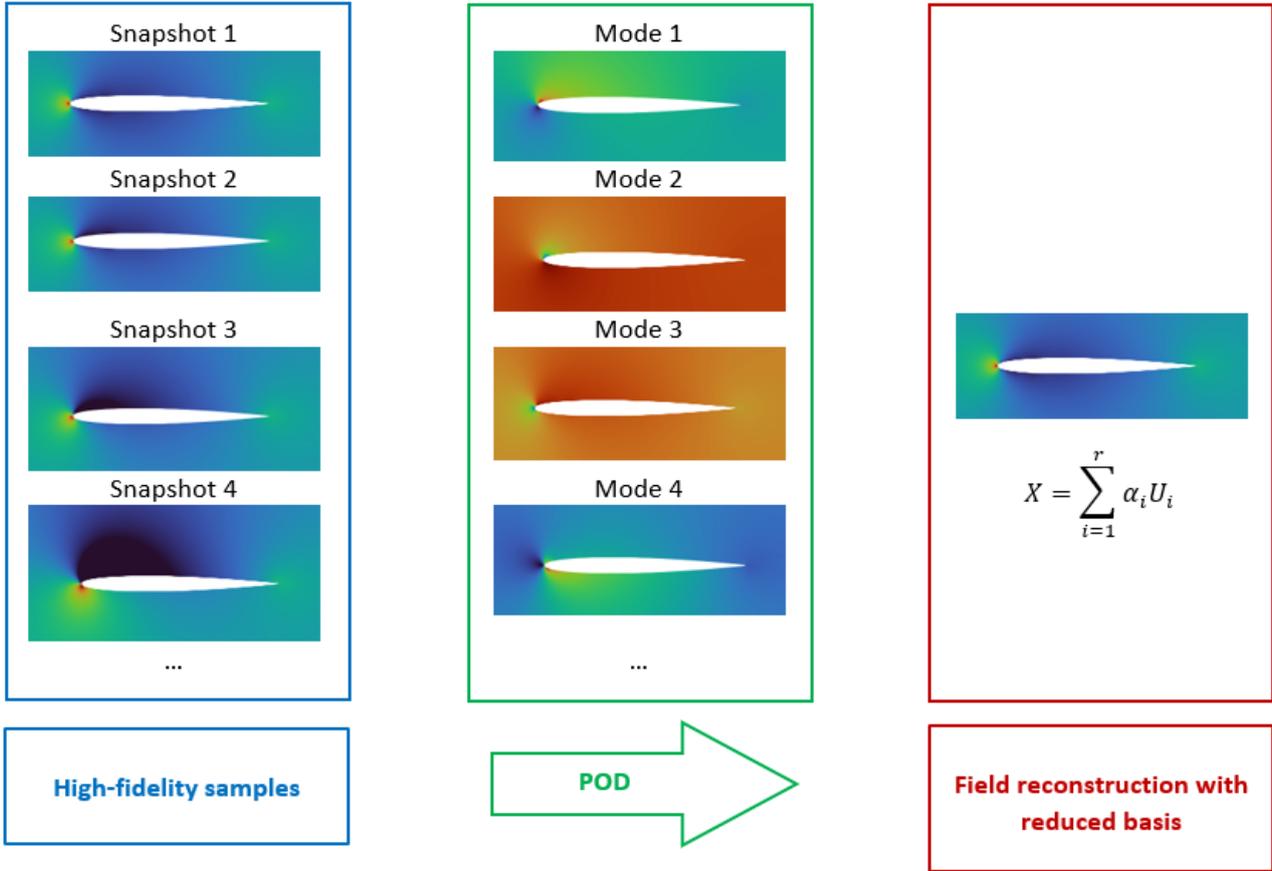


Figure 19: POD principle

One of the most effective methods used alongside POD is SVD (74), which compresses the dataset into a small number of modes. These modes correspond to the dominant features of the system's behaviour and are derived from the covariance matrix diagonalization of the dataset. Each DP of the dataset is a snapshot. Given that M is the matrix composed of the learning data where each column represents the values of an individual snapshot, the SVD matrix can be expressed as:

$$M = \begin{bmatrix} \vdots & \dots & \vdots \\ s_1 & \dots & s_n \\ \vdots & \dots & \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \dots & \vdots \\ u_1 & \dots & u_n \\ \vdots & \dots & \vdots \end{bmatrix} \times \begin{bmatrix} \sigma_1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \sigma_n \end{bmatrix} \times \begin{bmatrix} \vdots & \dots & \vdots \\ v_1 & \dots & v_n \\ \vdots & \dots & \vdots \end{bmatrix}^T = U \Sigma V^T \quad (2.50)$$

Where:

- Σ is a diagonal matrix composed of σ_i the singular values of matrix M .
- U and V are orthogonal matrices such that: $U * U^T = Id$ and $V * V^T = Id$.
- s_i are the learning solution vectors (learning snapshots i)

The field solution X in the design space can then be approximated as a linear combination of r modes:

$$X = \sum_{i=1}^r v_i \sigma_i U_i \quad (2.51)$$

r is a critical parameter that balances accuracy and compression while excluding noise. It's worth noticing that in the equation (2.50) each snapshot s_i is approximated by spanning the retained modes using as new modal coordinates v_i in the reduced space coordinates the retained components of the companion column of V resulting from the SVD.

Generally, an energetic approach is used to select the number of modes, which corresponds to imposing the condition:

$$\frac{\sum_{i=1}^r \sigma_i^2}{\sum_{i=1}^n \sigma_i^2} = 0.99 \quad (2.52)$$

In equation 2.52, it is imposed that the considered energy must be 99% of the total energy considering all modes. In general, the energy contained in each mode is expected to decay rapidly with increasing ranks.

Note that order of U and V is of the same order of grid points, which frequently achieves $O(10^7)$ in practice. Consequently, the decomposition process equation 2.50 would become prohibitively expensive. Therefore, an alternative version, known as reduced SVD (75) can be performed.

2.4.2. Surrogate models

A surrogate model aims to replace the original full-order system to achieve computational efficiency. In the context of ROM, a surrogate model is constructed for the reduced degrees of freedom based on samples generated from the high-fidelity simulations. In particular, once the modes have been extracted and the optimal number of modes r has been selected, it is necessary to identify a correlation between the model's input parameters and the mode weights. While various surrogate models are available in the literature, each model owns its pros and cons. In the following, a few typical models are described to provide brief introductions to this field.

2.4.2.1. Genetic Aggregation Response surface (GARS)

Response Surface Methodology (RSM) (76) is an extension of DOE that focuses on modelling output responses determined by a set of input parameters. It involves using various mathematical techniques to model a response surface that approximates the relationship between input and output variables to create a continuous response over the design space.

In this study, GARS method are used to find the correlations between scalar inputs and scalar outputs. In fact, this method is very robust and accurate, and it is widely used in the literature (76). The GARS method employs genetic algorithms to determine an optimal combination of multiple response surfaces, ensuring that the aggregated model captures the essential characteristics of the provided data. It iteratively refines parameters to enhance accuracy. Initially, a population of potential solutions is generated and evaluated based on fitness criteria. Through successive generations, individuals with higher fitness are selected, undergo crossover and mutation, producing offspring solutions. The process continues until convergence, yielding optimized weights for the response surface. This iterative approach efficiently navigates the solution space, improving the accuracy of weights evaluation for enhanced model performance.

Mathematically, the algorithm can be expressed as follows:

$$G(x) = \sum_{i=1}^n \omega_i R_i(x) \quad (2.53)$$

Where :

- $G(x)$ is the aggregated response surface,
- $R_i(x)$ are the individual response surfaces,
- ω_i are the optimized weights.

The weights are evaluated by integrating a multi-objective genetic algorithm (MOGA) (77) to meet the optimization objective requirements. A specific RS function can take various forms and its ability to fit the data depends on the type of problem and the simulation model used.

2.4.2.2. Neural Network (NN)

Artificial NN are computational models inspired by the functioning of the human brain. They consist of a set of artificial neurons connected to each other via synaptic weights. Each neuron processes the input, applies an activation function and passes the result on to the other connected neurons. It has at least one input layer, one or more hidden layers and one output layer. The main goal of this network is to approximate an unknown function, i.e. to find a relationship between inputs and outputs. The Levenberg-Marquardt (LM) backpropagation algorithm (78) is an advanced variant of the classical backpropagation algorithm. Initially, the network weights are assigned randomly or using a specific initialisation technique. Input is fed through the network, and intermediate results are calculated for each layer until output. The output of a neuron is calculated:

$$y = \sigma(\sum_{i=1}^n \omega_i x_i + b) \quad (2.54)$$

The error between the expected output and the desired output is calculated using an error function, such as mean square error (MSE). The error is propagated backwards through the network. The algorithm calculates the gradients of the error with respect to the weights of the neurons:

$$\omega_{k+1} = \omega_k - \frac{J^T J + \lambda I}{J^T e} \Delta \omega_k \quad (2.55)$$

This algorithm dynamically adapts the learning steps according to the geometry of the error curves, ensuring faster and more stable convergence than stochastic gradient optimisation. Training continues until a stopping criterion is met, e.g. when the error reaches a predefined threshold or stops decreasing.

2.4.2.3. Gaussian Regression

Gaussian process regression (GPR) is a non-parametric statistical technique used to model complex relationships between input and output variables. It is based on the concept of a Gaussian process, a multidimensional generalization of the normal distribution, where each point in the input domain is associated with a random variable, and the correlation between these variables is described by a covariance function (or kernel). Formally, a Gaussian process is defined as a collection of random variables, $\{f(x)\}_{x \in \mathcal{X}}$, such that any finite subset of these variables follows a multivariate normal distribution:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \quad (2.56)$$

where $m(x)$ is the mean function, often assumed to be zero ($m(x)=0$), and $k(x, x')$ is the covariance function or kernel, which represents the correlation between two points x and x' . The most commonly used is the radial basis function (RBF) kernel.

Given a training dataset $D = \{X, y\}$ where X are the inputs and y are the observed outputs, the Gaussian process uses the joint distribution of the observations and the points to be predicted to compute the predictive distribution. The posterior distribution for a new point x^* is also a normal distribution:

$$f^*(x^*) | D, x^* \sim N(\mu^*(x^*), \sigma^{*2}(x^*)) \quad (2.57)$$

where the predictive mean and variance are given by:

$$\mu^*(x^*) = k(x^*, X) K^{-1} y \quad (2.58)$$

$$\sigma^{*2}(x^*) = k(x^*, x^*) - k(x^*, X) K^{-1} k(X, x^*) \quad (2.59)$$

Here, K is the covariance matrix computed on the training points X , and $k(x^*, X)$ is the vector of covariances between the new point x^* and the training points.

GPR is particularly suitable for supervised learning problems because it provides not only a point estimate of the predicted value but also a measure of uncertainty.

2.4.2.4. RBF network

In this study, radial basis functions are tested as the interpolating functions to link input parameters to mode weights, owing to their excellent approximation and smoothing properties, as demonstrated by Ostrowski et al (79).

Equation 2.51 can be rewritten as:

$$a(p) = B \cdot f(p) \quad (2.60)$$

Where p is a generic vector of input parameters, $a(p)$ is the magnitude of the evaluated quantity (for example, the pressure field evaluated at each node), $f(p)$ represents the r modes extracted using POD, and B is the matrix of unknown coefficients

As proposed by Ostrowski et al., the inverse multiquadric radial functions have been employed here. The i -th function has a form of:

$$f_i(p) = f_i(|p - p^i|) = \frac{1}{\sqrt{|p - p^i|^2 + k^2}} \quad (2.61)$$

where k stands for user-defined smoothing factor and p^i is a known i -th node of the RBF network. In the proposed approach, the nodes $p^i, i = 1, 2, \dots, r$ of the RBFs are identical with the vectors p 's used when generating the snapshots. It can be seen, that the argument of the i -th RBF is the distance $|p - p^i|$, i.e. distance between its node and current point p .

Since the entries of parameter vector p are representing various quantities, usually of diversified magnitude, all entries should be normalized. Usually, the user-defined smoothing parameter k is chosen to be constant for all functions f_i 's, and is equal to the mean value of all normalized parameters.

To use efficiently approximation 2.60, the matrix B should be evaluated. This can be accomplished by making this equation formula exact for all snapshots. This requirement leads to a matrix equation:

$$\bar{A} = BF \quad (2.62)$$

The columns of matrix \bar{A} are the vectors of amplitudes corresponding to subsequent snapshots (it should be underlined that at this stage the matrix \bar{A} is known); F is the matrix of interpolation functions defined as set of r vectors $f_i(p)$.

Transposition of (2.62):

$$F^T B^T = \bar{A}^T \quad (2.63)$$

being a set of linear equations with respect to the columns of matrix B^T . Solution of (2.63) presents no difficulty and requires single factorisation of F^T .

2.4.3. Functional mock-up unit (FMU)

FMU is a standard component designed to facilitate the integration of heterogeneous models according to the functional mock-up interface (FMI). FMUs are modular and reusable components that implement standardized interfaces for the simulation of numerical models, facilitating interoperability between different modelling tools. Developed as part of the European Modelica project and formalized under the FMI standard, FMUs are employed in a wide range of applications (80) (81), including automotive, aerospace, energy, and robotics. An FMU is a self-contained functional unit that includes both a mathematical model and a communication interface. It is distributed as a zip file containing an XML description of the model, a binary executable file (which implements the mathematical model), and configuration data. There are two main types of FMUs:

- Model Exchange (ME): Allows the exchange of models described as systems of differential or algebraic equations. This type of FMU is integrated directly into the host simulator.
- Co-Simulation (CS): Includes an internal solver, enabling the FMU to run the simulation autonomously while communicating with other models or simulations through standardized interfaces.

FMUs can be generated to make ROMs reusable. In our approach we encapsulate the ROM and the global results into FMUs. We adopt three types of FMUs to manage the geometrical model, the field results and the global results. FMUs are intended to manipulate sets of scalar quantities and in our implementation we receive as input the new state x to be represented in the parametric space (each individual parameter as a scalar CAD input) and we get as output the state of the system in the reduced coordinate system $v_i(x)$ as in (2.51). The evaluation happens real-time and allows to get from the corresponding FMUs the full state: geometry, field results and global results. As far as the portability of the modes (large fields of data) instead of accessing directly the data from the FMU we have the ability to write the full field (i.e. a complete span of the modes) on file. This is not efficient for interactivity and so we adopt the solution to dump the modes on file from the FMU and then load them on the device (i.e. the GPU of the VR or the one of the laptop). This approach is a workaround to the data structures limitations of FMU up to 2.0, vector data can't be directly exposed or manipulated; scalars (integer, real) are supported along with strings. At run-time the FMU can receive as input the weights of the mode and then dump on file the combination of the modes. As file data exchange can be a bottle neck, we use the same approach to dump on file all the modes invoking the file export by setting all the coefficients zero but the one of the mode to be written.

2.4.4. Results Deploy

Another advantage of using the FMU standard is that ROMs can be imported into any environment for deploying the results. Specifically, in the application part, two deployment environments were proposed: MATLAB and Unity. MATLAB is a standard in the engineering field, and having a real-time model in MATLAB/Simulink allows the integration of other models to create a more complex system. Moreover, MATLAB makes it easy to develop applications for real-time interaction.

Unity, on the other hand, not only provides an environment with much higher graphical features but also enables virtual reality (VR). In the application chapter, a test case is proposed in which a real-time design environment in VR was developed.

2.4.4.1. MATLAB Dashboard

MATLAB/Simulink is a widely used design environment in engineering. The purpose of this section is to demonstrate how to create a black-box system that provides, in real time, all results of a CFD analysis (both scalar quantities and field variables) within the MATLAB/Simulink.

The first step is importing the FMUs, which is done in Simulink using a dedicated block. For the RS, a single FMU is imported, encapsulating the relationship between the scalar input and scalar output of interest. For the ROM, two different FMUs are imported:

One FMU manages the generation of modes, which is executed only once, with the modes being stored as vectors. The other FMU manages the generation of weights, using the RS to output the weights corresponding to each n-tuple of inputs (as shown in Figure 20).

Once the weights and modes are available, a simple linear combination allows to compute the desired field quantity. This approach, which separates modes and weights, enables real-time responsiveness. The modes are stored in memory and evaluated only once, while the linear combination requires minimal computational time.

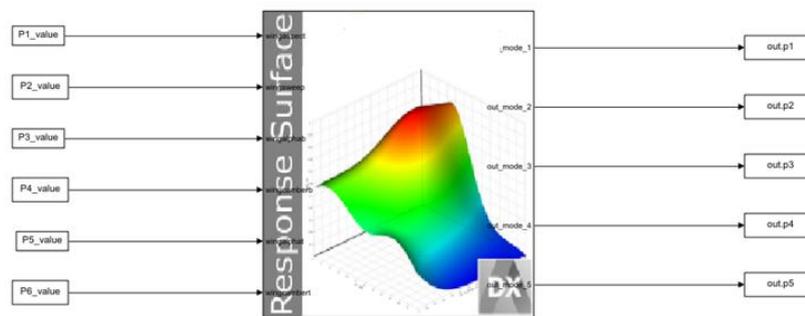


Figure 20: FMU imported on Simulink

The collected data are processed by a MATLAB script that facilitates optimization at various levels. Scalar values can guide the optimization process and help identify a global optimum through response surfaces.

For multi-objective optimization, a Pareto front can be determined, representing the best trade-off solutions. Once potential targets are identified, ROMs (Reduced-Order Models) are utilized to select the optimal solution. Real-time parameter exploration allows users to observe how parameters influence aircraft aerodynamics. This approach supports a more robust and informed optimization process by offering a clear understanding of the underlying physics and the contribution of each parameter.

Additionally, analysing 3D contours or 2D profiles can help discard design points that, while improving efficiency, might lead to greater flow instability during CFD optimization. In other fields, such as thermo-structural analysis, average values or peak magnitudes are often used to identify optimal solutions. However, having access to the entire solution field enables more reasonable assessments and allows for lower safety factors to be applied.

Once all models are imported into MATLAB, an application can be developed using the dedicated tools. Figure 21 shows a preview of the interface for the NACA intake case study, which is described further in the application chapter.

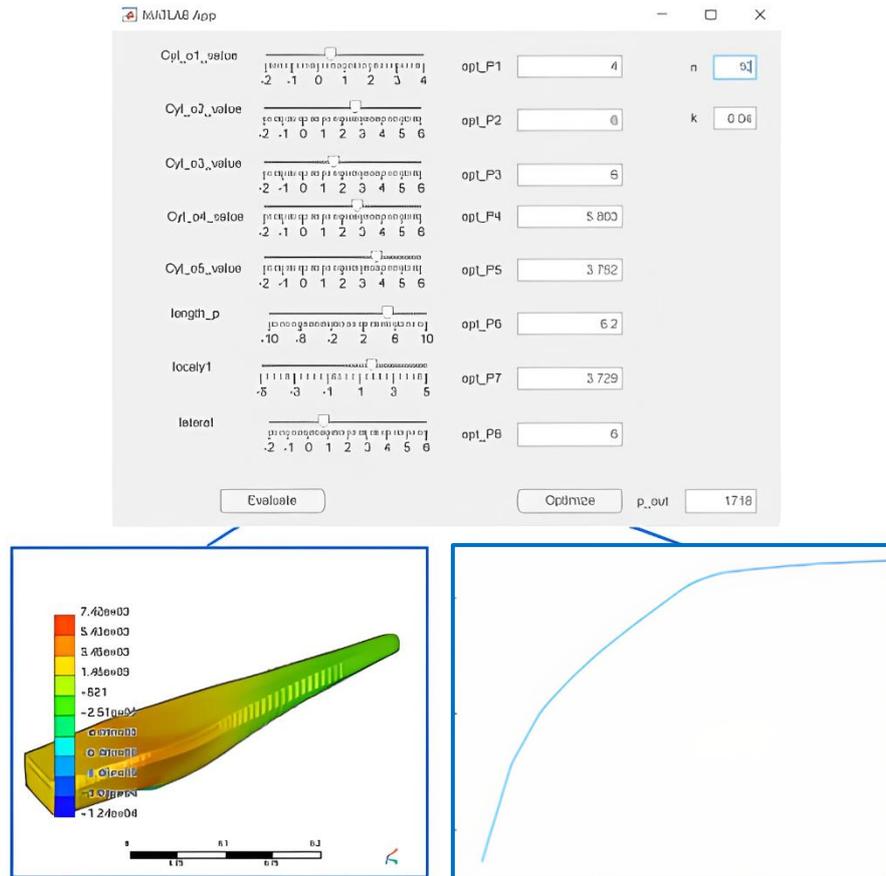


Figure 21: Example of Matlab Dashboard

The idea is to interact with the dashboard on two levels:

- Real-time exploration of the parameter space: users can adjust input parameters using sliders and evaluate the solution in real time. Both the geometry and the pressure field are updated dynamically.
- Performance optimization: optimization can be performed using the ROM itself. For example, in the reported case, the ROM can be used to evaluate the average pressure at the outlet, and the gradient descent method applied to find the maximum. Alternatively, another FMU containing only the RS for the scalar quantities of interest can be used for optimization.

This dashboards are developed in the sections 6.2 and 6.3.

2.4.4.2. VR Dashboard

VR is a rapidly growing technology that is now being applied across various fields (82). In engineering, the potential of this tool is still being explored, but its key advantages are already evident. These include:

- Ease of interaction with objects and models,
- Immersive environments,
- The ability to create collaborative spaces where experts and non-experts from different sectors can interact to share ideas and projects.

Using the graphical dashboard, the user can see the deformation of the mesh and the field values for all the combinations of the available parameters in an interactive UI and/or in a virtual reality environment, in which the interaction is managed using a digital representation of the user's hands.

The dashboard has been developed using Unity Engine by leveraging the latest features offered by its VR packages. It consists of three main modules: the import module, the visualization module, and the interaction module.

The import module is responsible for managing the reading of the data needed by the visualization module. The data needed are as follows:

- Baseline 3D mesh (STL file)
- FMUs containing transformations from input parameters to mode coefficients
- Mode matrix of the mesh ROM (CSV file)
- ROM modes matrix of the field quantity evaluated (CSV file)

The baseline 3D mesh is imported using a custom format that stores the collection of 3D vertices and the collection of triangle indices. As the connectivity of the mesh is not known because FMUs store cloud of points, at import time, the number of vertices is first checked to be sure that FMU data are compatible with the mesh, and then the connectivity between the mesh and the cloud is reconstructed with a fast search algorithm. We used a NNS (Nearest Neighbor Search) algorithm based on a k-d tree with $k = 3$.

The visualization module is responsible for managing the rendering of the imported data in real-time, both for the mesh deformation and texturing. The mesh texturing is used to represent the pressure values.

The diagram below (figure 22) shows what happens in each frame when the user updates one or more of the available parameters.

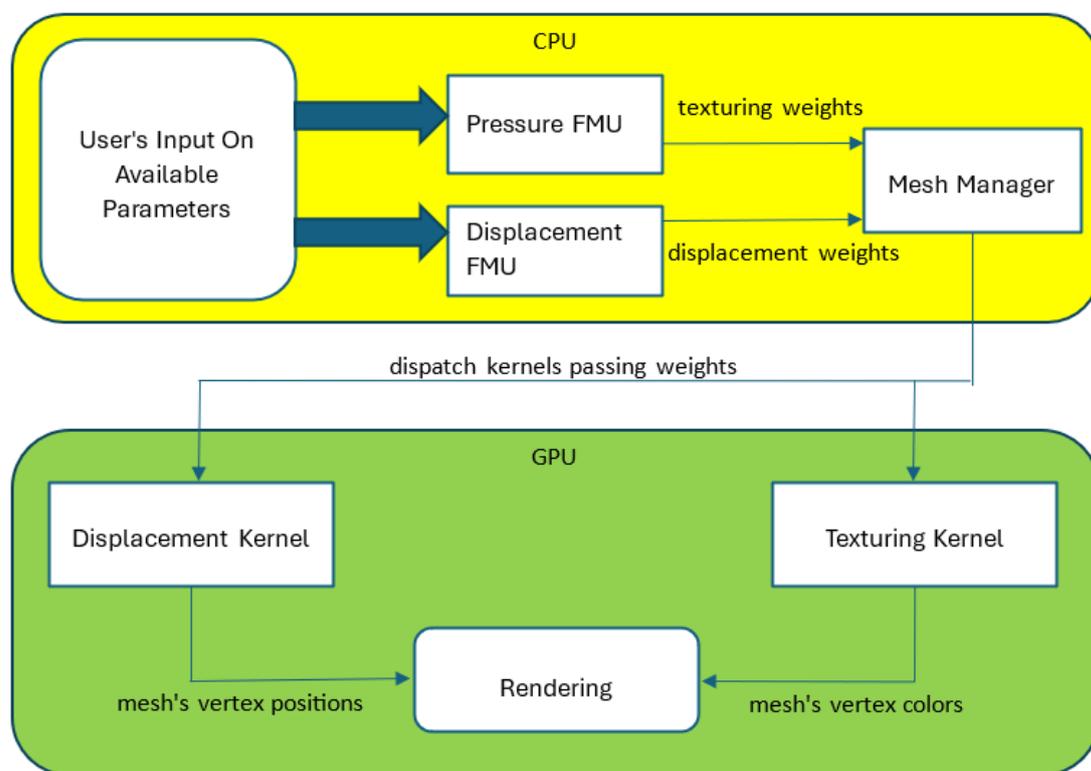


Figure 22: The diagram showing what happens in each frame when the user updates one or more of the available parameters.

All the necessary calculations, except those of the FMU weight retrieval, are performed on the GPU so that the data structures that contain the results are passed directly to the rendering pipeline.

The displacement and the texturing kernel are initialized with their respective modes so that the final vertex positions and colours can be computed as a weighted sum of those modes.

The colour of the single pixel is computed using the interpolated value generated during the rasterization phase of the rendering pipeline.

The interaction module is responsible for managing the user's interactions with the virtual environment. Its VR implementation is based on Unity's VR Interaction Toolkit. The main purpose of this module is to allow the user to modify the value of the parameters using a 3D user interface and/or 3D handles.

As soon as the user modifies one of the parameter's values, the new value is propagated to the visualization module to update the mesh displacement and texture, as shown in the paragraph 6.5.

Figure 23 previews the appearance of the VR dashboard for an aircraft model (section 6.5) . Users can interact with the model through both the menu sliders and handles positioned on the model itself, located in front of the actual geometry. In real time, updating the input values allows users to observe how the quantities of interest change dynamically.

This example is specific to a CFD problem, but the methodology can be easily extended to other physical domains. For example, Figure 24 shows another application case (CUSP), where the structural behaviour (stress and deformations) is monitored (section 6.4).

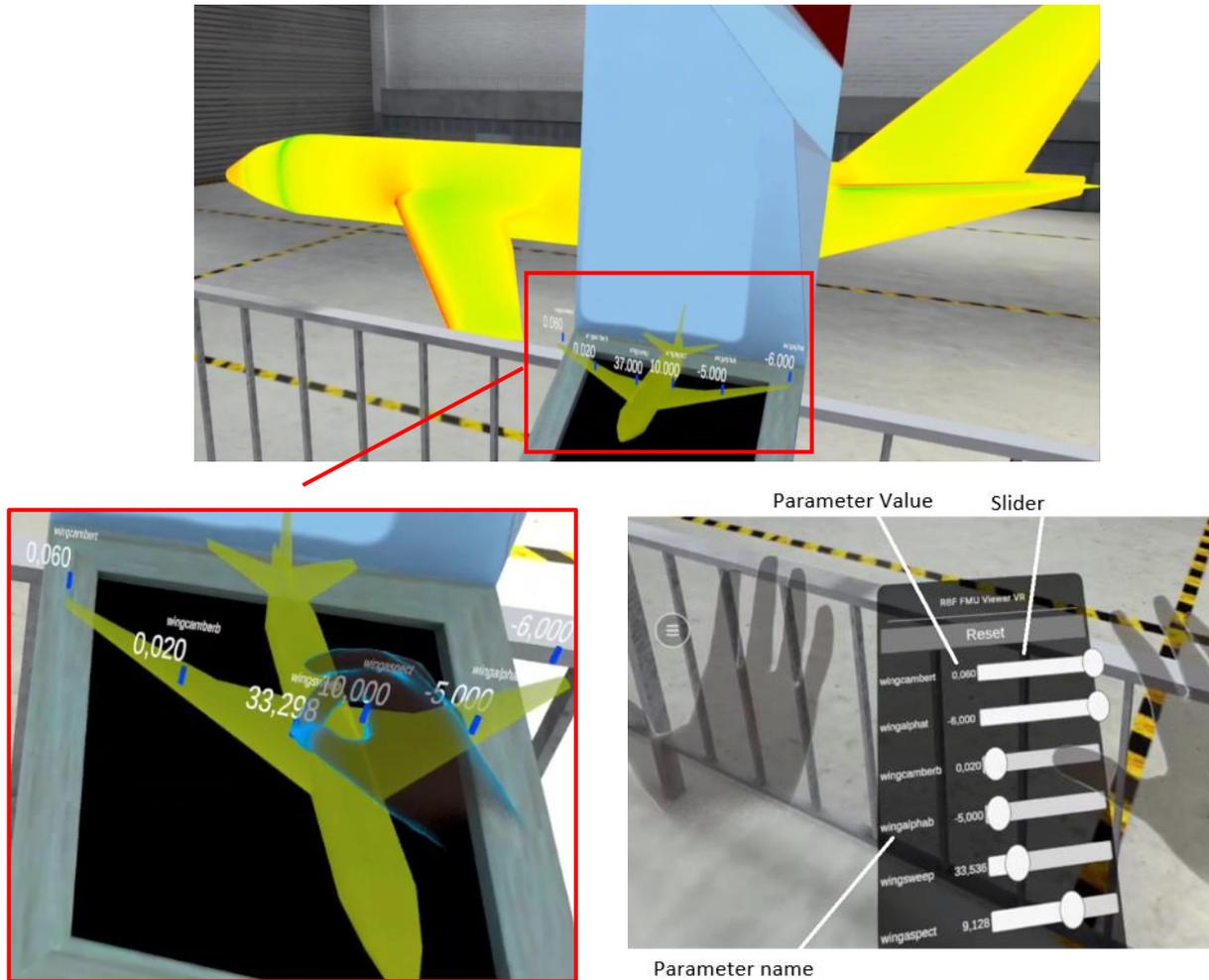


Figure 23: VR Dashboard for a CFD testcase

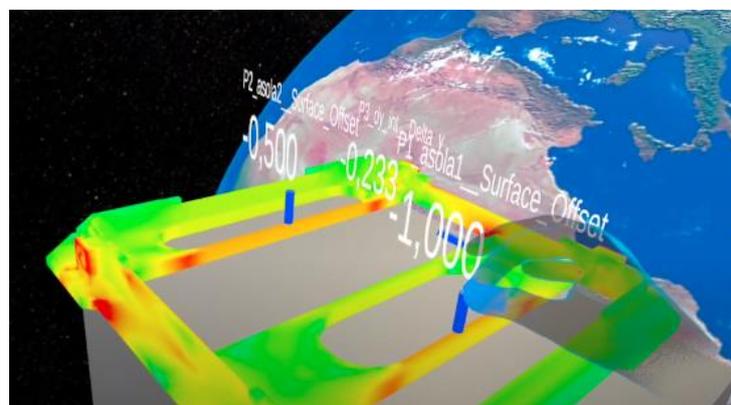


Figure 24: VR Dashboard for a FEM testcase

3. Multi-physics Analysis

This paragraph presents several fluid-structure interaction analyses, exploring various methods and approaches. The initial focus was on the problem of mapping, as the main challenge with the two-way coupling approach lies in transferring information between the CFD and FEM tools. Specifically, a mapping algorithm is used to transfer CFD loads onto the FEM model, while FEM displacements must be applied to deform the CFD mesh.

The first proposed application investigates mapping algorithms. The second application compares the classical two-way coupling method with the modal method. The latter offers significant computational advantages at the expense of some precision loss. However, the proposed applications demonstrate that the modal method achieves a high level of accuracy despite this trade-off.

3.1. RBF Mapping for Steady FSI

The theoretical background of the proposed method has been presented in the section 2.2.1.1. This work is published in (83). Reliable data exchange is a critical challenge in the loose coupling of CFD and Computational Structural Mechanics (CSM) modules in FSI applications. This work compares two methods for mapping the pressure field across mismatched grids: the RIBES method (84) and the preCICE algorithm (85), both based on RBF interpolation.

The two methods offer varying levels of control over balance preservation during mapping, with the RIBES algorithm demonstrating superior efficacy. The test cases include a parametric double-curved geometry and a wind tunnel mock-up. In the latter case, the mapped forces are used to load a CSM model to derive stress and displacement fields. While differences in FEM results are noticeable, they are not significant, nonetheless highlighting a correlation between the accuracy of balance preservation during data mapping and the structural outputs.

3.1.1. The Catenoid

The first test bench used to validate the soundness of the algorithms under comparison is the catenoid (86), a double curvature hull parametrically described by these equations:

$$x = \cos(u) \cosh(v) \tag{3.1}$$

$$y = v \tag{3.2}$$

$$z = \sin(u) \cosh(v) \tag{3.3}$$

with $u \in [0, \pi]$ and $v \in [-1.5, 1.5]$. The pressure field acting on the catenoid is also given an analytical form:

$$p = \sin(3x + 3y) \tag{3.4}$$

The parametric shell is discretized into triangular patches, with the total number of nodes or elements determining the refinement level. Mapping is carried out from source to target meshes according to the matchings detailed in the rows of Table 2. This approach ensures comprehensive coverage of all coupling scenarios, including transitions from coarse to fine, fine to coarse, and between similar discretization levels.

Figure 25 illustrates the grid, which consists of 196 nodes and is constructed based on the catenoid geometry. The colormap shown represents the pressure field as defined in Equation 3.4.

Table 2: Row-wise: discretization size of mesh pairs used for mapping.

ID	Source		Target	
	Number of nodes	Number of elements	Number of nodes	Number of elements
1	36	50	2116	4050
2	196	338	1444	2738
3	484	882	900	1682
4	900	1682	484	882
5	1444	2783	196	338
6	2116	4050	36	50

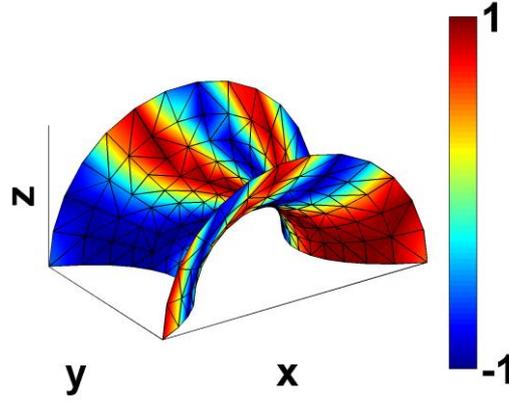


Figure 25: Grid comprising 196 nodes constructed on the catenoid geometry. Colormap refers to the pressure field.

The RIBES method employs Radial Basis Function (RBF) interpolation to compute source pressure field values at target nodes. These interpolated values are then subjected to an optimization process, guided by fuzzy sets, to ensure balance conservation between the source and target.

Each node of the coarser mesh acts as the centre of a spherical fuzzy subset, encompassing a defined number of nodes from the coarser mesh. Similarly, identical fuzzy subset arrangements are created over the finer mesh, with nodes grouped within the same pre-defined spherical volumes. Balance between homologous subsets in the source and target domains is enforced using corrective coefficients that vary smoothly over the target mesh.

Determining the optimal number of points to include in each fuzzy subset is a challenging task. While the mathematical properties of fuzzy subsets inherently guarantee translational balance, selecting the appropriate number of nodes can help minimize errors in rotational balance, ensuring a more accurate mapping.

In most cases involving shell meshes, this minimum lies within the interval $[lbs, ub_s]$, with

$$\begin{aligned} lbs &= \lceil 3 \log_{10} \min(P_S, P_T) \rceil \\ ub_s &= \min(2lbs, P_S, P_T) \end{aligned} \quad (3.5)$$

Let P_S and P_T denote the number of points in the source and target grids, respectively. For each scenario in Table 2, every integer within the range specified in Equation 3.5 is used to fine-tune the correction of the RBF interpolation. Only the configuration that minimizes rotational imbalance is retained.

The workflow described is implemented using MATLAB. Table 3 summarizes the balance errors encountered with the RIBES method for the scenarios outlined in Table 2. The RBF kernel employed is a linear spline with no polynomial supplement, and the pole for moment calculations is positioned at the centre of gravity of the target mesh.

As anticipated, larger errors are observed at the extremities of Table 3, where the most dissimilar meshes are interconnected. However, the maximum error in the magnitude of the moment vector remains below 5%, a level considered acceptable for practical applications.

Table 3: Percentage errors on global resultants ,component-wise ($\frac{F_{i,target} - F_{i,source}}{F_{i,source}} \cdot 100$) and for vectors

($\frac{\|F_{target} - F_{source}\|}{\|F_{source}\|} \cdot 100$), RIBES method

ID	ΔF_x	ΔF_y	ΔF_z	ΔF	ΔM_x	ΔM_y	ΔM_z	ΔM
1	0%	0%	0%	0%	-6.158%	0.241%	-9.208%	5.071%
2	0%	0%	0%	0%	0.240%	-1.677%	-11.178%	1.389%
3	0%	0%	0%	0%	-0.779%	-0.646%	-16.148%	0.727%
4	0%	0%	0%	0%	-1.180%	-0.244%	-17.619%	0.752%
5	0%	0%	0%	0%	-5.639%	-0.548%	-101.42%	3.520%
6	0%	0%	0%	0%	2.650%	-4.247%	-651.52%	3.786%

The same tests were conducted using the preCICE algorithm. It is important to note that preCICE provides the user with a choice between conservative and consistent mapping approaches.

The conservative method uses nodal forces as input and strives to maintain translational balance between the source and target meshes. However, in coarse-to-fine mapping scenarios, forces are applied only to a subset of target nodes, selected based on their proximity to the source nodes. While this approach may achieve satisfactory balance results, it lacks consistency in output. Specifically, a continuous pressure field is transformed into a discontinuous, spiky field, with abrupt variations between zero and peak values.

Following the preCICE guide's recommendations for mapping physical (non-cumulative) variables such as pressure, the authors chose the consistent mapping option and evaluated its performance in terms of balance preservation. Table 4 summarizes the balance errors for the catenoid test bench, corresponding to the mesh couplings listed in Table 2.

Table 4: Percentage errors on global resultants ,component-wise ($\frac{F_{i,target}-F_{i,source}}{F_{i,source}} \cdot 100$) and for vectors

($\frac{\|F_{target}-F_{source}\|}{\|F_{source}\|} \cdot 100$), preCICE

ID	ΔF_x	ΔF_y	ΔF_z	ΔF	ΔM_x	ΔM_y	ΔM_z	ΔM
1	8.006%	17.073%	-100.19%	14.968%	23.644%	10.488%	-100.237%	16.982%
2	2.935%	0.056%	-94.773%	2.430%	7.536%	0.077%	-98.372%	5.459%
3	0.511%	0.139%	-109.41%	0.458%	1.021%	0.115%	-123.26%	2.237%
4	-0.284%	-0.656%	-950.94%	0.577%	-1.472%	0.110%	-494.64%	2.368%
5	-0.353%	-5.087%	1474%	3.828%	-10.965%	2.500%	2181%	7.785%
6	-23.257%	6.657%	8552.9%	17.126%	-7.174%	-7.750%	-7071.1%	7.926%

PreCICE consistent algorithm introduces discrepancies in balance for both translational and rotational equilibria, which, for the most dissimilar meshes exceed 15% of vector magnitude.

3.1.2. The RIBES wing

The RIBES wing model, used in the RIBES experimental campaign as a wind tunnel mock-up, features a wingspan of 1600 mm (root to tip). Both CAD and numerical models are available online (16).

The comparison between mapping algorithms now encompasses the entire CFD–CSM workflow, which is interrupted after one iteration. CFD analysis is performed within the SU2 environment, an open-source CFD solver widely recognized for its high reliability in both compressible and incompressible analyses, particularly in the aeronautical field. In this study, a "standard air" model is used, incorporating compressibility effects and assuming air behaves as an ideal gas. An SST turbulence model is applied, and the boundary conditions from the RIBES project are adopted: steady analysis with a Mach number of 0.11762, an angle of attack of 6°, a Reynolds number of 1.41E6, and a Reynolds length of 0.5153.

The results, particularly the pressure field (used for mapping), are compared with those obtained using the commercial solver ANSYS Fluent, which serves as the reference solver. As shown in Figure 26, there is good agreement between the results.

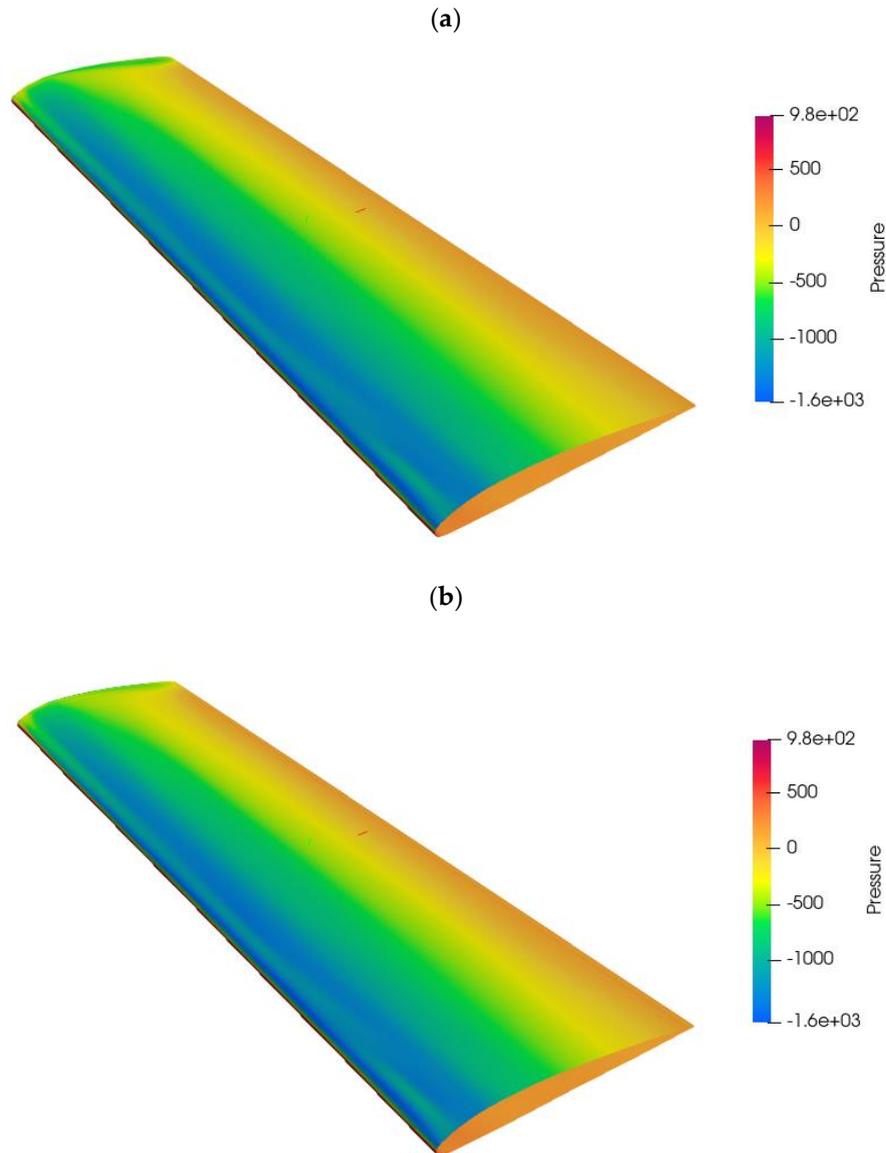


Figure 26: Comparison of the pressure fields for the RIBES wing obtained with Fluent (a) and SU2 (b).

The CFD wing boundary is extracted from the full model by removing the wing cap, which mismatches with the CSM model, and the trailing edge (Figure 27). The trailing edge occupies a small portion of the overall model and contains almost coincident nodes, making its contribution negligible from the perspective of global balance. The resulting wing hull consists of 8673 quadrilateral patches and 8900 nodes.

The CSM model incorporates a wing skin reinforced with ribs and spars. Only the mesh of the wing skin undergoes the mapping process, while the reinforcement structure remains intact for the final stress assessment. The target mesh consists of 47082 triangular and quadrilateral elements and 45846 nodes.

The RIBES method employs RBF interpolation and scans the range specified in Equation 3.5 to determine the optimal correction scheme for rotational balance.

Figure 28 presents a triple comparison: the pressure field on the source mesh (a), and the pressure on the target domain before (resulting from basic RBF interpolation) and after correction (left and right, respectively, for (c) and (b)). At first glance, all the pressure fields appear similar, yet solutions (b) and (c) for the target mesh lead to significantly different sets of force and moment resultants. The difference in the pressure field on the suction side is visible in the corrected pressure field, as it adjusts to the local mesh refinement.

Table 5 reports the percentage errors in balance for the pressure field mapped with the RIBES method against the source field.

Stress assessment is conducted in FEMAP, using Nastran as the solver. The complete CSM model is loaded with the corrected mapped field, and the structural model is anchored at the base, simulating the attachment of the wing to the fuselage, similar to a simple cantilever beam. The maximum values of von Mises stress and vertical deflection are 69.8 MPa and 8.1 mm, respectively. Figure 29 shows the stress and displacement maps resulting from the FEM analyses.

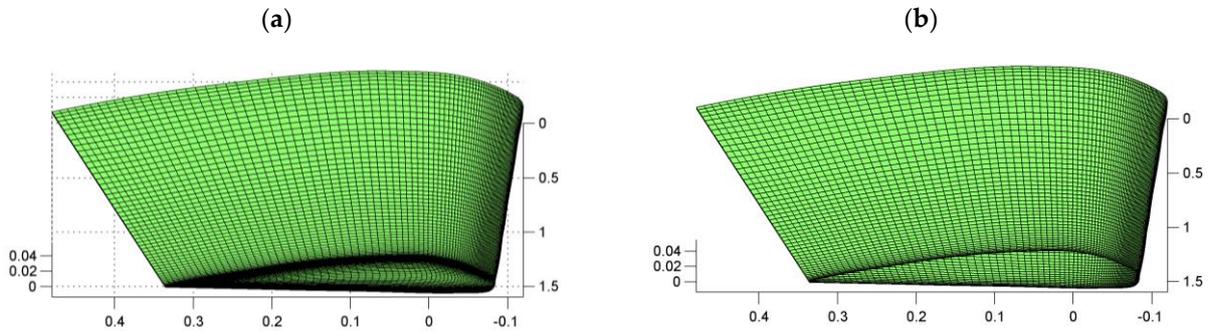


Figure 27: CFD mesh before (a) and after (b) the removal of the wing cap and the trailing edge.

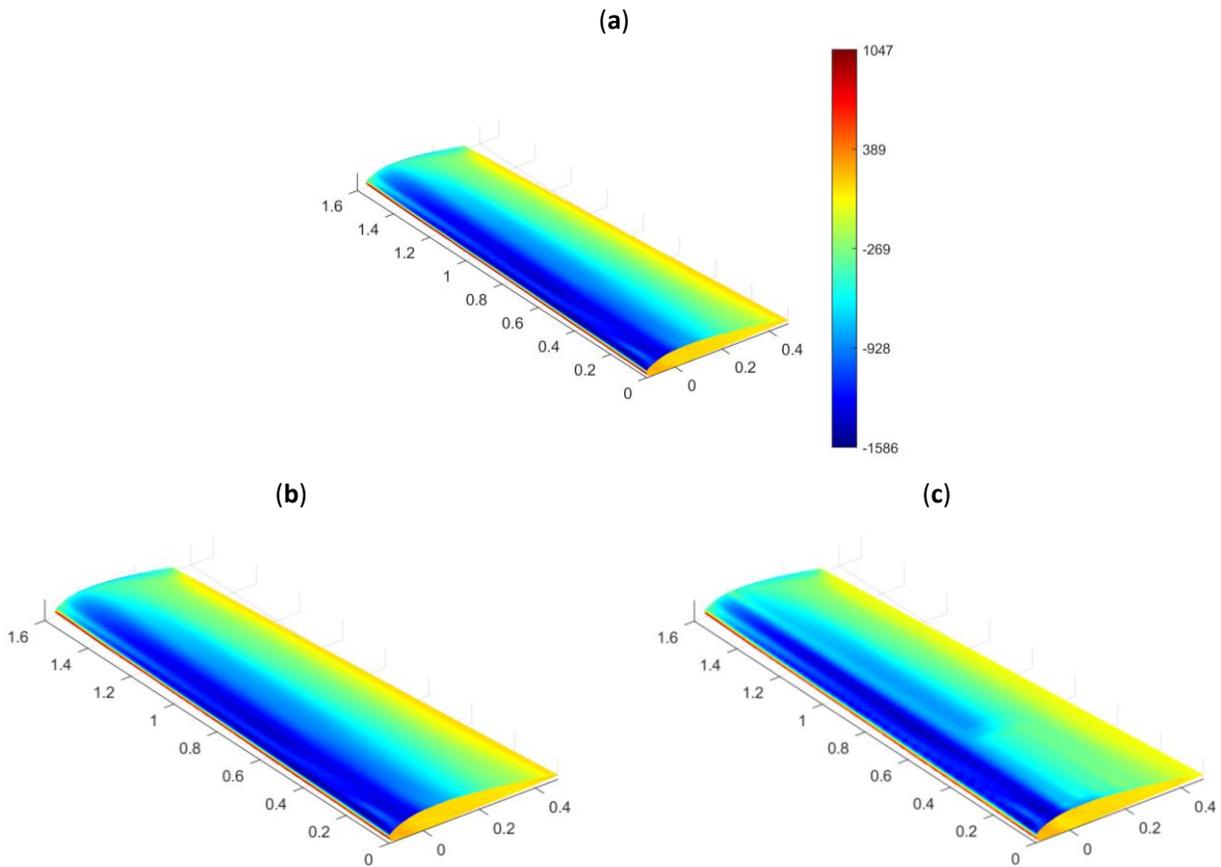


Figure 28: Pressure field on RIBES wing: CFD source model (a), CSM target model with pressure field interpolated by plain RBF (c), CSM target model with pressure field mapped with RIBES method (b).

Table 5: Percentage errors on global resultants ,component-wise ($\frac{F_{i,target}-F_{i,source}}{F_{i,source}} \cdot 100$) and for vectors

($\frac{\|F_{target}-F_{source}\|}{\|F_{source}\|} \cdot 100$), RIBES method

ΔF_x	ΔF_y	ΔF_z	ΔF	ΔM_x	ΔM_y	ΔM_z	ΔM
0%	0%	0%	0%	-0.081%	-1.285%	-0.208%	1.216%

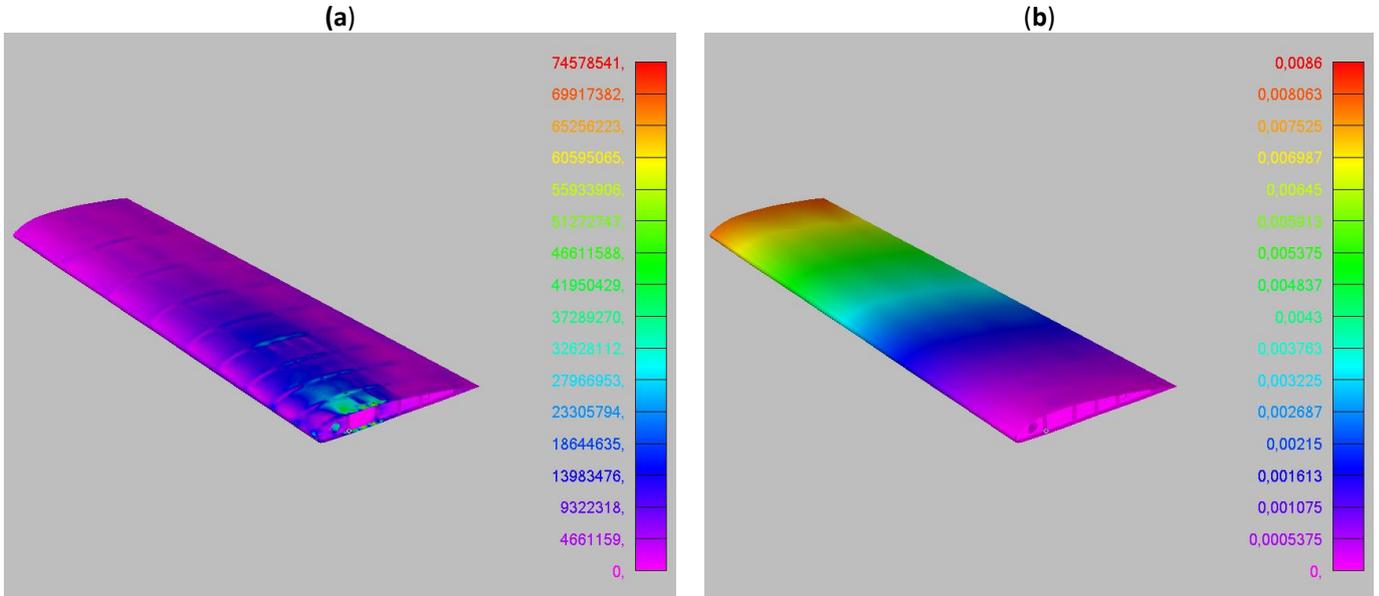


Figure 29: FEM output for the mapped loads from the RIBES method: (a) Von Mises stress, maximum value 69.8 MPa, (b) vertical deflection, maximum value 8.1 mm.

The same mapping instance is then addressed using the preCICE algorithm, with the consistency-preserving option selected. Table 6 presents the balance errors, while Figure 30 displays the target pressure fields. Furthermore, Figure 31 shows the results from the FEM analysis in terms of displacements and stresses, with the maximum values of von Mises stress and vertical displacements reaching 72.5 MPa and 8.25 mm, respectively.

Table 6: Percentage errors on global resultants ,component-wise ($\frac{F_{i,target}-F_{i,source}}{F_{i,source}} \cdot 100$) and for vectors

($\frac{\|F_{target}-F_{source}\|}{\|F_{source}\|} \cdot 100$), preCICE

ΔF_x	ΔF_y	ΔF_z	ΔF	ΔM_x	ΔM_y	ΔM_z	ΔM
8.744%	4.959%	6.078%	6.091%	70.65%	18.48%	-22.31%	28.45%

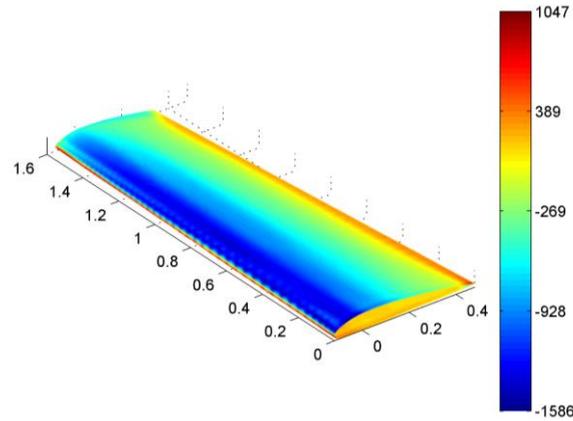


Figure 30: Pressure field on RIBES wing CSM target model with pressure field interpolated by consistent preCICE algorithm

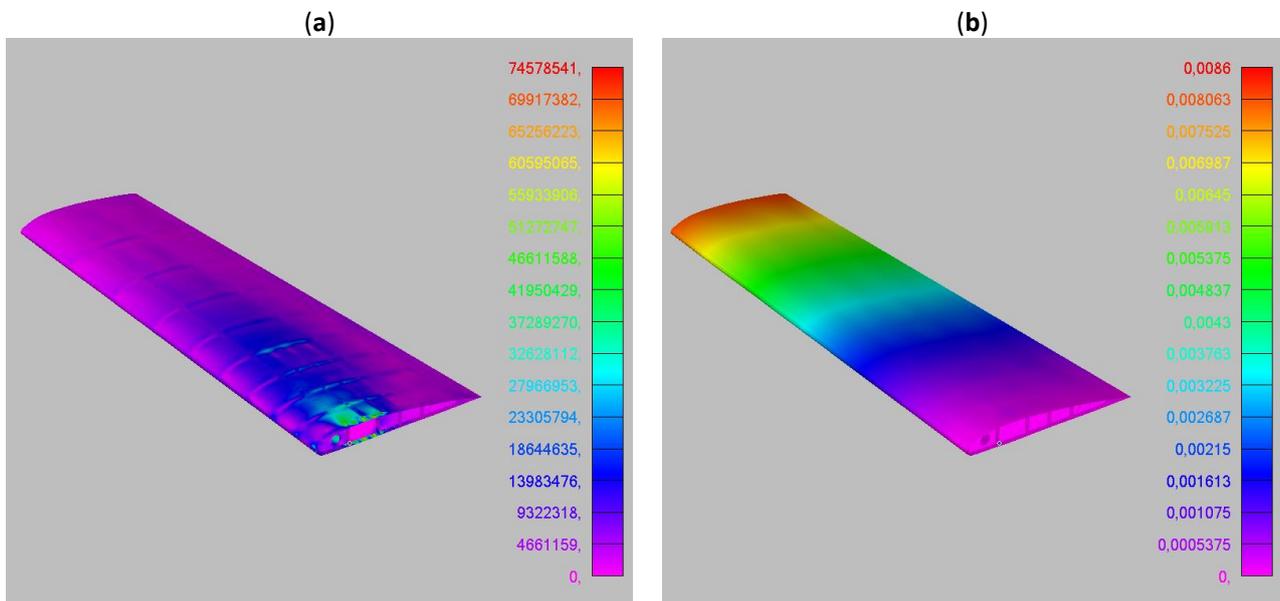


Figure 31: FEM output for the preCICE mapped loads: (a) Von Mises stress, maximum value 72.5 MPa, (b) vertical deflection, maximum value 8.25 mm.

3.1.3. Discussion

A well-designed mapping algorithm should transfer information across mismatching grids in a conservative manner, meaning no data loss or alteration should occur. Simultaneously, it should be consistent, ensuring that a pointwise source field, varying within a specific interval, maintains its overall trend across the target mesh, with values remaining within similar bounds. Additionally, the mapping algorithm should be efficient, achieving these goals in a reasonable time frame and without excessive computational resource usage.

The RIBES method strives to meet these criteria by combining the consistency of Radial Basis Function (RBF) interpolation with a subsequent phase using fuzzy sets to enforce balance conservation retrospectively. Significant time savings can be achieved by adopting boundary decomposition methods and computational parallelization.

On the other hand, the preCICE algorithm allows the user to choose between consistency and conservation, revealing a limitation in guaranteeing both simultaneously. In this study, the authors compared the RIBES method against the preCICE consistent algorithm across two distinct test scenarios.

Regarding consistency, both approaches demonstrated high quality. The mapped fields closely resembled the source field, with colour plots showing similar variations across domains, without significant overshoots or undershoots. From a balance perspective, the RIBES method holds an advantage due to its underlying

mathematical formulation, which ensures translational balance. Moreover, the local correction procedure results in smaller discrepancies in rotational balance compared to the preCICE approach.

In terms of performance, while the RIBES method achieved considerably faster data transfer, the preCICE algorithm, in its basic form, follows a plain serial approach. Therefore, this aspect should be excluded from the comparison, as the underlying conditions differ.

For the FEM results of the RIBES wing, von Mises stress peaks differ by 3.87%, while the maximum deflections differ by 1.85% between the two mapped fields. The stress and displacement fields largely coincide, and although the discrepancies in their maximum values are noticeable, they are not significant. However, it is worth noting that the model considered is a wing mock-up with a limited wingspan. The observed discrepancies may vary for full-scale models or with different boundary conditions. Therefore, the discussed methods remain valuable as precise tools for investigation.

3.1.4. Conclusions

This work provides a comparative analysis of two load mapping techniques used to transfer data between CFD and CSM modules in FSI applications. RBF are key components of both algorithms, namely, the RIBES method and the one integrated into the preCICE library. These approaches were evaluated in two different scenarios: a simpler case involving the parametric geometry of the catenoid, and the practical application of the RIBES wing, utilized in the aeroelasticity experimental campaign as part of the RIBES EU project. The RIBES method exhibited higher precision in preserving balance between the source and target domains. Utilizing the mapped pressure fields on the wing to individually load the structural model in FEM analyses revealed that variations in balance could impact the obtained fields to an extent that, in this specific case, proved to be noticeable yet not significant.

3.2. Comparison of two-way and modal superposition for Steady FSI: HIRENASD testcase

In this section, a comparison between the classical 2-way method and the modal approach is presented. The 2-way method is the most accurate but requires several steps and the interaction of different tools at each step. The modal method significantly speeds up the calculations. The theoretical background of the proposed method has been presented in the section 2.2.1.2. In this section, both methods are demonstrated on an industrial case study, the HIRENASD wing (87) (88). Full details of CFD and FEM model can be find in (89).

For these analyses, the commercial solvers ANSYS Fluent and Nastran were used. In the two-way workflow, the pressures are mapped onto the FEM model using ANSYS Fluent's direct mapping tool, while the commercial tool RBF Morph was used for RBF mesh morphing.



Figure 32: HIRENASD real model in the wind tunnel

The FEM model of HIRENASD is shown in the figure 33. It can be observed that a shaker is present at the base of the model, aimed at reproducing the natural modes of the structure by vibrating it at the frequencies at which these occur. The FEM model is a literature model that has been experimentally validated up to the 20th mode (90).

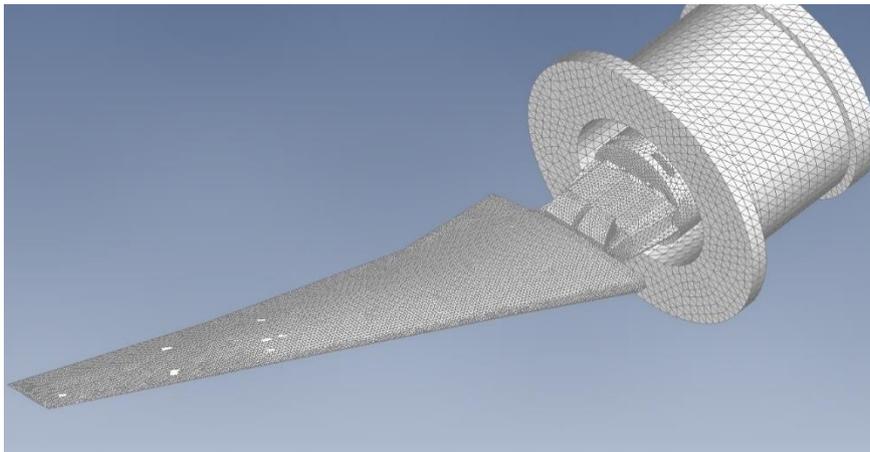


Figure 33: HIRENASD FEM model

The coarse configuration from those available in the repository was used for the CFD mesh. The CFD model has a mesh consisting of 13169981 cells, 32398946 faces, and 7206319 nodes; Figure 34 shows the surface mesh of the airfoil. The flight condition studied was Mach 0.8 and an AoA of 1.5°. y^+ is less than 20 and y^+ mean is 0.45.

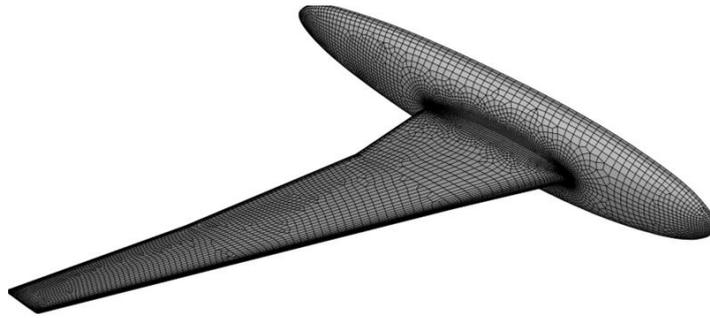


Figure 34: CFD Surface mesh

3.2.1. Two-way method

The first approach considered is the Two-way approach. Iteratively, the CFD loads are computed and imported into the FEM model to calculate the displacements, which are then used to deform the CFD mesh. Source points are reported in figure 36. To solve an RBF problem of 14406 source points, 46 s are required using 48 cores CPU. A domain is defined to limit the morphing area (blue in figure 36). In each iteration, drag, lift, and maximum displacements are monitored. Nastran is used for FEM analysis, Ansys Fluent for CFD and their coupling. The following figures show the results obtained. Table 7 reports a comparison of C_L and C_D for the baseline and the coupled analysis. RBF were used to deform the CFD mesh according to the FEM displacements. Figure 37 shows the plot of the tip displacement. It can be observed that the method converges by the third iteration.

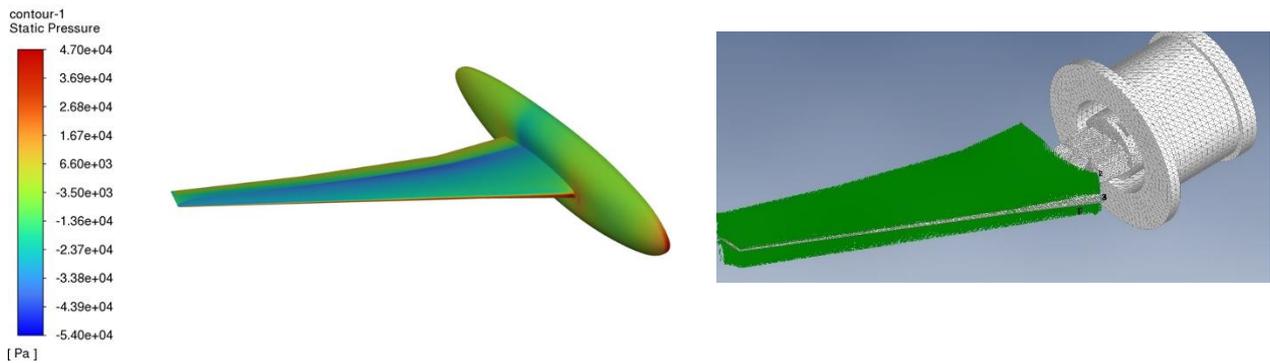


Figure 35: CFD static pressure field (left) and load imported on FEM model (right)

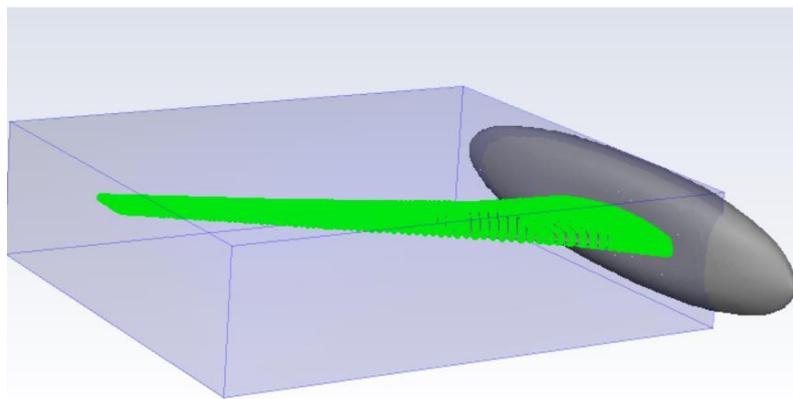


Figure 36: RBF source points (green) and domain box (blue) for RBF mesh morphing

Table 7: Comparison of baseline and two-way results

	Baseline	Two-Way	$\Delta\%$
C_L	0.3568	0.3395	-4.85%
C_D	0.0137	0.0144	+5.11%

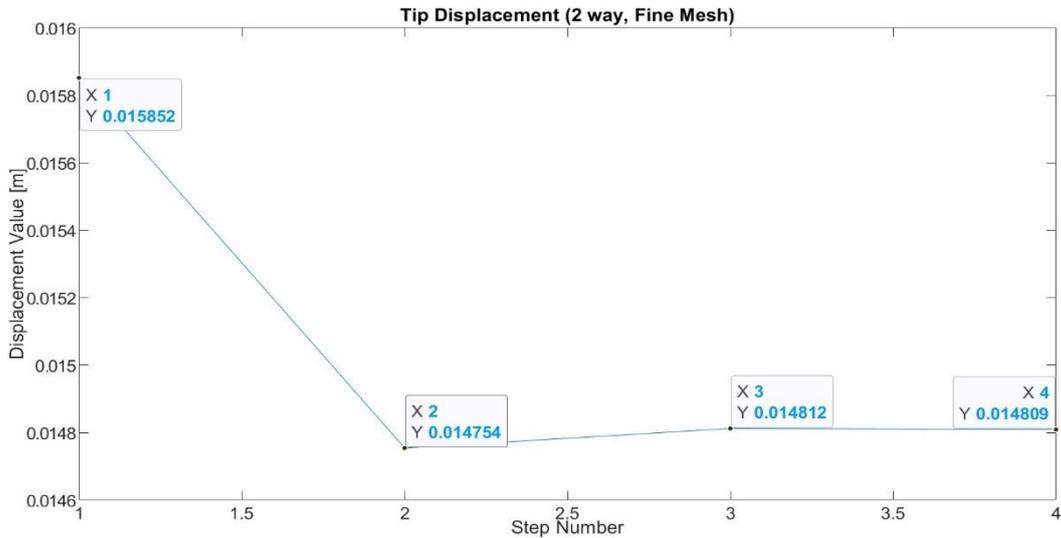


Figure 37: Convergence plot for two-way method, tip displacement

3.2.2. Modal Superposition

In this case, a modal analysis was first performed using Nastran. The figure 38 shows the first six vibration modes of the structure, arranged in increasing order of frequency.

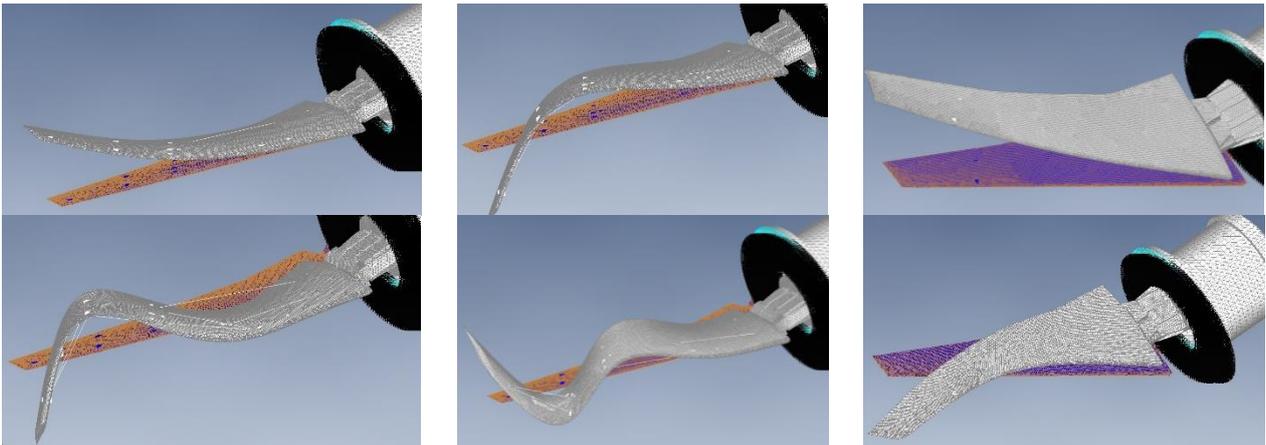


Figure 38: First six vibration modes

Once the generation of RBF solutions for the various modal shapes was completed, the setup of the modal superposition analysis was initiated. Since this is an iterative process, it was necessary to define the iteration interval for which the modal shape amplification factors would be updated. The advantage of this procedure lies in the inherent automation it allows: in fact, continuous interfacing between the two solvers is not required.

The table 8 presents the tip displacements obtained as the number of modes considered varies, along with the percentage difference compared to the displacement obtained with the Two-Way analysis. The results presented in Table 8 regarding the influence of the vibration modes can be verified through modal basis qualification. This process involves superimposing the modal shapes, appropriately amplified by their modal coordinates, with the static solution from the first step of the Two-Way analysis. The last modal shape's coordinate (in this case, the sixth) is then nullified, and any significant changes in the geometry are checked. It is important to note that at this step, the modal shapes have already been calculated, and only a surface mesh preview is required. Therefore, this procedure is very fast. It can be observed that the displacements converge with the use of 4 modes.

Table 8: Tip displacements obtained as the number of modes considered varies, along with the percentage difference compared to the displacement obtained with a single mode

# Modes	Displacements [mm]	$\Delta\%$
1 mode	15.94	/
2 modes	14.79	-7.21%
3 modes	14.81	-7.09%
4 modes	14.89	-6.59%
5 modes	14.88	-6.65%
6 modes	14.88	-6.65%

Figure 39 shows the error obtained by progressively increasing the number of modes. Figure 39-a illustrates the difference (scaled by a factor of 500 for better visualization) between the Two-Way solution and the first mode with an appropriate calculated weight. As the number of modes increases, the error decreases. By comparing Figure 39-a with Figure 38, it can be seen that this result is similar to the second vibration mode.

In Figure 39 b, the error obtained by considering only the first two modes is shown, which is similar to the fourth modal shape presented in Figure 38, and so on. Based on this, it can be concluded that displacement convergence occurs from the fourth mode onward, confirming the results shown in Table 8.

It is worth noting that modal basis qualification is a procedure that can be implemented before running the FSI analysis by calculating the modal coordinates from the baseline CFD solution.

In this way, a modal superposition analysis can be launched that will certainly converge in terms of displacements.

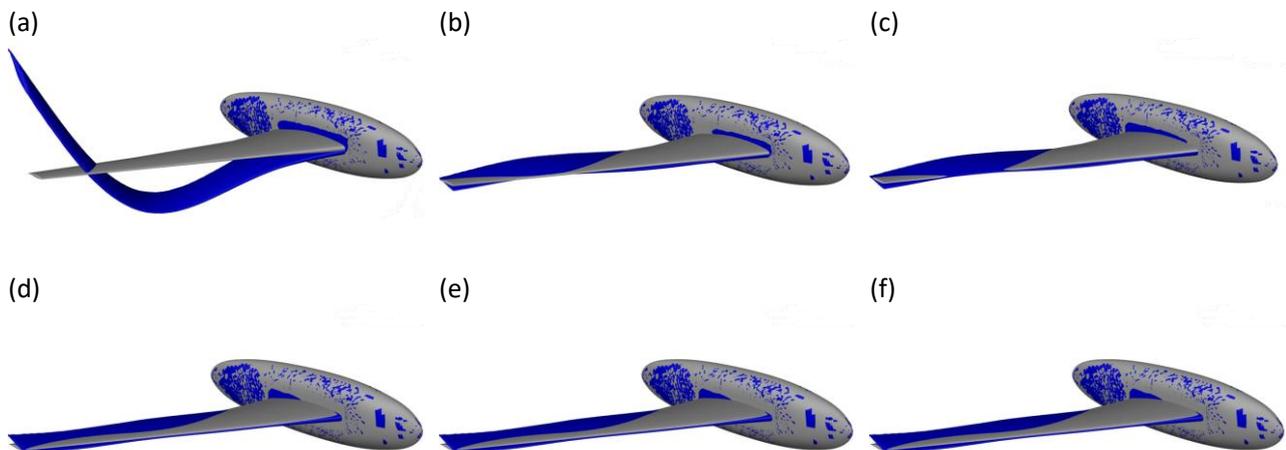


Figure 39: Displacements error for increasing number of modes

The tip displacement trend obtained through this FSI analysis methodology is shown in the figure 40.

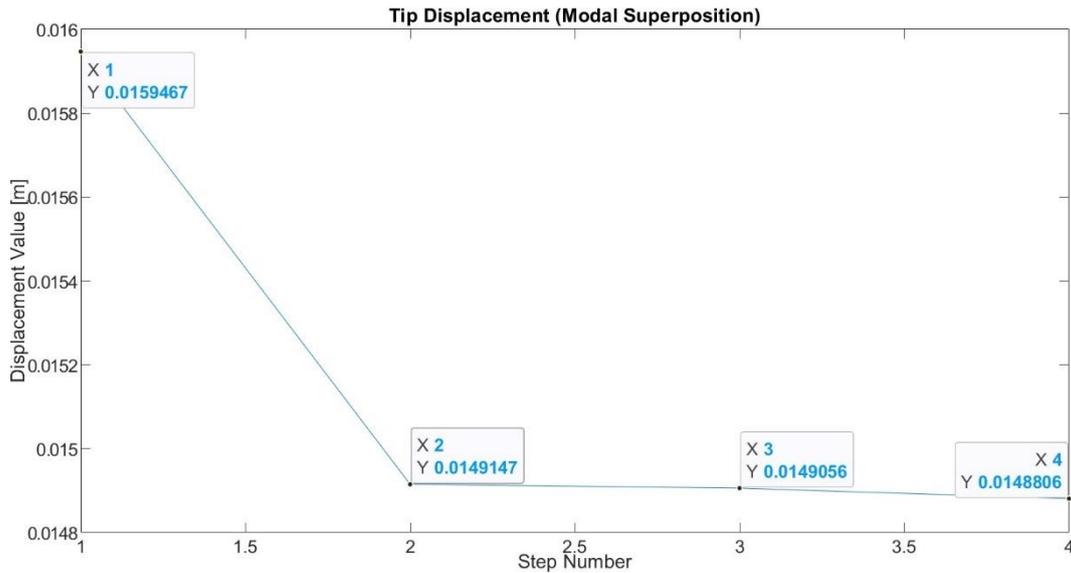


Figure 40: Modal superposition method convergence plot

Table 9: Comparison of baseline and modal superposition results

	Baseline	Modal	$\Delta\%$
C_L	0.3568	0.3395	-4.85%
C_D	0.0137	0.0144	+5.11%

3.2.3. Discussion

Tables 10-11 present a comparison between the two procedures. For the HIRENASD case, the chosen analysis methodology has no significant impact on the results obtained.

Table 10: Comparison of drag and lift coefficients between two-way and modal methods

	Two-Way	Modal
C_L	0.3395	0.3395
C_D	0.0144	0.0144

Table 11: Comparison of tip displacements between two-way and modal methods

	Two-Way	Modal
Disp [mm]	14.81	14.88

Considering that the angle of attack to be used is 1.5° , the comparison between the displacements calculated in this case and those available in the literature (90), where a value of 14.26 mm was obtained, indicates that the findings are in line with the expected results (there is a discrepancy of 0.6 mm, which is 4.2%), confirming the validity of the analyses conducted.

Finally, the computational time differences observed in the execution of the analyses are examined. These analyses were conducted on a remote server equipped with an Intel Xeon Gold 6152, all within a Linux environment. In both cases, 20 threads were used. There were 256 GB of RAM available, which was never fully utilized. Table 12 presents the timing results obtained from the Two-Way process and the modal superposition process. It can be observed that the first method requires more time to complete compared to the second.

Table 12: Computational time summary

	Two-Ways	Modal
Time	4h 15min 2s	58min 59s

Based on the information presented in Table 12, it can be stated that the modal superposition strategy allows for analyses to be performed in a shorter time.

3.2.4. Conclusions

In this study, two methodologies for steady-state FSI analysis were compared, using the HIRENASD case, a benchmark from the literature. Both methods were found to be highly effective and robust, and the results obtained were very representative when compared to the literature values. The two-way method is a well-established approach, while the modal superposition method is less commonly used in the literature. An approach to evaluate the number of modes to consider before launching the coupled analysis is proposed. By progressively increasing the number of modes and monitoring the displacement field, it is possible to estimate the number of modes required. This operation is very quick and does not significantly affect the overall computational time. As a result, the modal method is highly accurate. The major advantage of the modal superposition method is the simplicity of the workflow and automation. The FEM solver is only called in the first step, and the entire analysis is performed inside the CFD solver. Finally, computational times are lower for the modal case which is about 4 time faster than the 2-way one.

4. Multi-fidelity optimization

The aim of this study is to present a multi-fidelity approach for optimizing a design process. One of the most challenging aspects when transitioning from an analytical to a numerical approach is the transfer of information between different levels. In this study the two methods introduced in 2.3.1 are demonstrated. The first involves mesh morphing: the analytical solution is converted into a point cloud and used as an RBF field to deform the baseline mesh. The second method involves CAD automation directly using a scriptable editor.

4.1. Shape Optimization of Frame Structures through a Hybrid Analytical-2D and Numerical-3D Approach

In this study, we propose a method for the shape optimization of frame structures using a mixed analytical-numerical approach. The goal is to achieve a uniform-strength frame structure, ensuring optimal material utilization and weight minimization. The optimization is performed in two calculation steps. The first step uses an analytical model based on the Timoshenko beam theory, where through appropriate mathematical steps give the uniform-strength shape of the entire structure. Depending on the type of cross-section analysed, the exact uniform-strength profile of each element is derived by solving for three parameters related to the forces and moments acting on the element. These parameters are obtained by solving a nonlinear system of equations, which includes the external and internal kinematic constraints of the structure, as well as equilibrium equations for each element. However, the solution obtained using the one-dimensional theory is not valid in areas affected by boundary effects, such as the interconnection regions between elements and those near the supports, for at least a decay distance equal to the characteristic diameter of the section.

To address this limitation, the second optimization step involves incorporating solutions that account for a triaxial stress field. This is typically done by discretizing the structure using the finite element method. The frame geometry obtained from the previous analytical solution is constructed, and the regions affected by boundary effects are optimized using the BGM (described in the section 2.3.2.4). This is an iterative, bio-inspired method modelled after the growth of trees, which increases trunk diameter in proportion to the loads experienced. The method is applied simultaneously to all regions where three-dimensional effects are significant, with the aim of achieving uniform strength in areas influenced by boundary effects. An important aspect of applying the BGM is maintaining the topology of the initial mesh, which is ensured through the use of mesh morphing techniques. The results of the two-step optimization process are shown on simple geometries involving few elements and on more complex geometries of mechanical interest.

4.1.1. Introduction to the problem

This study is partially published in (91). Frame structures are among the most popular and versatile engineering solutions in mechanical and building design. These systems, composed of interconnected beams, offer an optimal combination of strength and rigidity making them suitable for a wide range of applications in a variety of engineering fields (92). The principle behind frame structures is to distribute loads efficiently by exploiting the geometry and rigid connection of structural elements.

In recent decades, with the advent of new materials and advanced analysis methods, frame structures have seen a significant evolution, improving both in terms of performance and sustainability. In particular, the development of additive manufacturing techniques (93) and homogenization algorithms (94) has made it possible to greatly extend the scope of application of these structures on ever smaller scales.

On the basis of the above, it is evident that the optimization of frame structures is a very topical and pressing problem.

In this work, an innovative optimization workflow was proposed to combine analytical and numerical methods for the optimization of frame structures. The workflow idea is illustrated in Figure 41.

In the first optimization step, an analytical approach was adopted: starting from the Timoshenko beam theory, the uniform strength condition was imposed to compute the optimal beam cross-section. This method makes it possible to obtain a first optimized geometry in a very short time. However, the main limitation is the failure of Timoshenko beam theory to consider three-dimensional effects, making the analytical solution less effective in the presence of striction or joint zones where a triaxial stress state appear. To take edge effects into account, finite element analysis must be introduced.

To this end, an FEM model and a numerical method capable of optimizing the most critical areas is required. The BGM is chosen because it is compatible with the analytical approach described above, as it allows the introduction of a target stress, is automatable and very robust. Therefore, the second optimization step is based on a numerical method that allows the critical areas to be resolved, also taking into account three-dimensional effects. Applying the BGM to the already optimized geometry allows the optimal solution to be found with significantly fewer iterations.

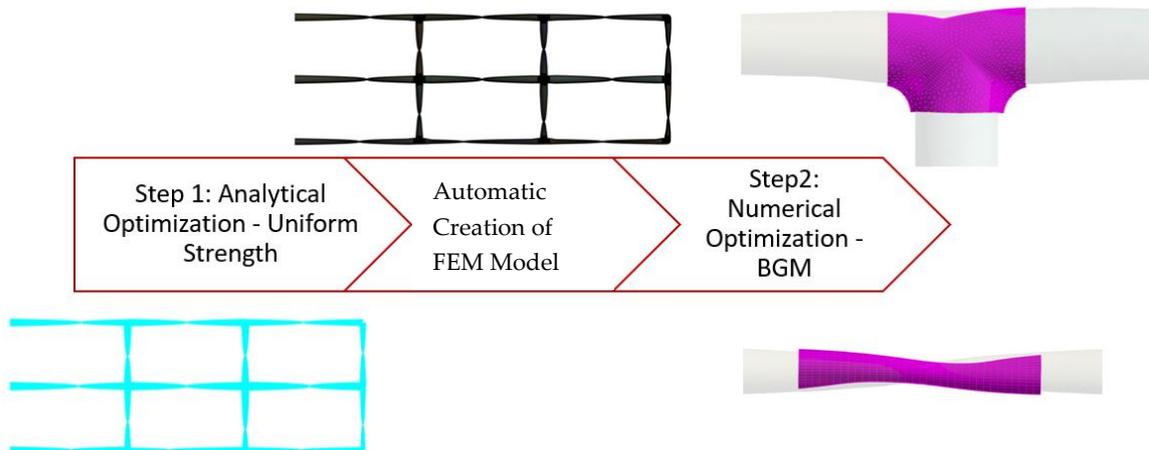


Figure 41: Workflow

4.1.2. The Uniform-Strength Shape from the Timoshenko beam model

Many structural optimization algorithms share a significant limitation: the need to repeatedly perform Finite Element Analysis (FEA) at each evaluation of the objective function, i.e., at every iteration. To substantially accelerate the optimization process, we propose an approach that first obtains an optimized configuration from a 1D analytical model based on beam theory. This allows the subsequent 3D optimization, using the BGM method, to start from an already optimal state and focus on refining the solution in only a few calculation steps. These

refinements are applied selectively, targeting areas where they are most needed specifically, regions of striction and junctions between elements.

The detailed analytical procedure followed is described in (95), and the key concepts will be summarized below. As a reference, consider the beam shown in figure 42, with length L , variable height $h(x)$, and subject to forces and moments (F_1, \dots, F_6) at the end-nodes.

The six forces and moments must be fulfilling the equilibrium equations:

$$F_4 = -F_1 \quad (4.1)$$

$$F_5 = -F_2 \quad (4.2)$$

$$F_6 = F_2 L - F_3 \quad (4.3)$$

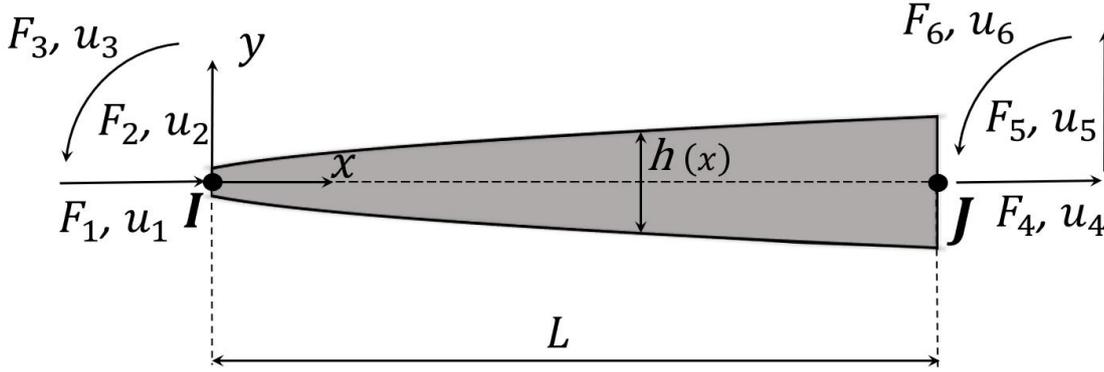


Figure 42: Generic beam element with variable cross-section

As results of the planar Timoshenko beam model (96), the stress state is biaxial, where the axial and mean shear stresses approximation are:

$$\sigma_x(x, y) = \frac{N(x)}{A} - \frac{M(x)}{I} y \quad (4.4)$$

$$\tau_x = \frac{T(x)}{A} \quad (4.5)$$

Where the axial and shear forces N , T and the bending moment M are linked with the applied forces as follow:

$$N(x) = -F_1 \quad (4.6)$$

$$T(x) = -F_2 \quad (4.7)$$

$$M(x) = F_2 x - F_3 \quad (4.8)$$

The shape $h(x)$ that ensures the iso-stress condition can be obtained from Eq. 4.5 by setting $\sigma_x(x, y = \pm h(x)/2) = \sigma_{iso} = constant$, obtaining:

$$\sigma_{iso} = \frac{|N(x)|}{A(x)} + \frac{|M(x)|}{I(x)} \frac{h(x)}{2} \quad (4.9)$$

where σ_{iso} is the uniform-strength (iso-resistance) stress, $A(x)$ and $I(x)$ are the variable area and moment of inertia of the cross-section.

Eq. 4.9 consider only the axial stress, i.e. do not take into account the shear stress. This is not a problem, since according to the Jourawsky's approximation (96), shear stress vanishes at top and bottom of a beam cross-section. The information of the mean shear in Eq. 4.5 will be used to set a minimum thickness of the cross-section in the region in which the bending moment is null, avoiding the absence of material.

Considering a circular cross-section of radius $R(x)$, inserting into Eq. 4.9 the expression of $A(x), I(x)$ as functions of the radius and solving by respect $R(x)$, one obtain:

$$R(x) = \sqrt[3]{-\frac{2|M(x)|}{\pi \sigma_{iso}}} \left[\sqrt[3]{-1 + \sqrt{\Delta(x)}} + \sqrt[3]{-1 - \sqrt{\Delta(x)}} \right] \quad (4.10)$$

where:

$$\Delta(x) = 1 - \frac{1}{108 \pi \sigma_{iso}} \frac{|N(x)|^3}{M^2(x)} \quad (4.11)$$

It is worth to point out that Eq. 4.10 simplify a lot if the axial force is neglected, which is a recurring case in frame of lattice structures when the stress due to the bending moment is dominant.

Eq. 4.11 is the analytical expression of the beam shape that ensure the iso-stress condition. It can be used to integrate analytical or numerical the kinematic of the beams (elastic line) to obtain the six nodal displacements and rotations (u_1, \dots, u_6) (Figure 42) that are function of the nodal forces of the elements. The analytical equations of the kinematics can be used to form a nonlinear system of equations, function of the unknown nodal forces (F_1, \dots, F_N), in which at each node the conditions of equilibrium, constraint and kinematic congruence are imposed. The solution of the nonlinear system is obtained by means the trust-region algorithm.

4.1.3. Performance Factors

To evaluate the effectiveness of the proposed method, several factors were introduced to compare the geometries obtained at various optimization steps. Specifically, two factors were introduced that assess the stress distribution on the surface (factor f_1) and throughout the entire volume (factor f_2), respectively:

$$f_1 = \frac{\sqrt{\sum_{i=1}^{n_S} (\sigma_{iso} - \sigma_{vm})^2}}{n_S} \frac{1}{\sigma_{iso}} \quad (4.12)$$

$$f_2 = \frac{\sqrt{\sum_{i=1}^{n_V} (\sigma_{iso} - \sigma_{vm})^2}}{n_V} \frac{1}{\sigma_{iso}} \quad (4.13)$$

Where σ_{iso} is the uniform resistance stress used as a target, σ_{vm} is the von Mises stress evaluated on each node, n_S and n_V are the number of nodes on surface and volume respectively. So f_1 and f_2 are the root mean square deviations of the von Mises stress evaluated on all surface nodes and on all volume nodes, respectively.

Additionally, an energy factor f_3 was introduced, defined as the ratio between the elastic strain energy and the maximum energy that could be accumulated if all points be stressed at the maximum stress value:

$$f_3 = \frac{\xi * 2E}{\sigma_{iso}^2 V} \quad (4.14)$$

Where ξ is the elastic strain energy, E is the Young's module and V is the structure's volume.

f_1 , f_2 and f_3 are scalar values that allow us to make global considerations about the analyzed structures.

In the following figures, the same terminology is used to indicate the values of these quantities evaluated at each node. These contour plots enable local considerations on the areas where the optimization has been more or less effective.

4.1.4. Case 1: Frame structure

Case 1 is a generic frame structure. The initial geometry and dimensions are shown in the Figure 43. Nodes 1, 5 and 9 are fixed and a vertical force of 7kN is applied at node 12.

A 3D CAD model of this baseline configuration was created from the diagram. It is emphasized that the model, being extremely simple, consisting of a series of beams with a constant circular cross-section, can be easily generated with a scriptable CAD editor. Specifically, Ansys Spaceclaim® was used to automatically generate the baseline CAD.

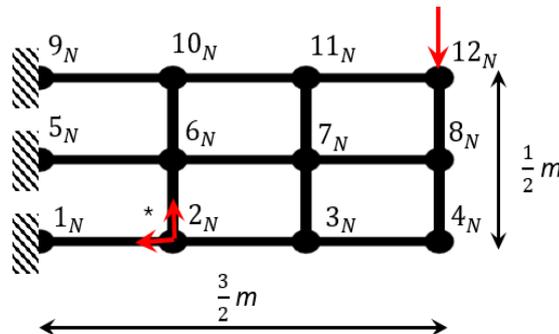


Figure 43: Baseline geometry

Finally, the FEM model used is shown in Figure 44:



Figure 44: Baseline mesh

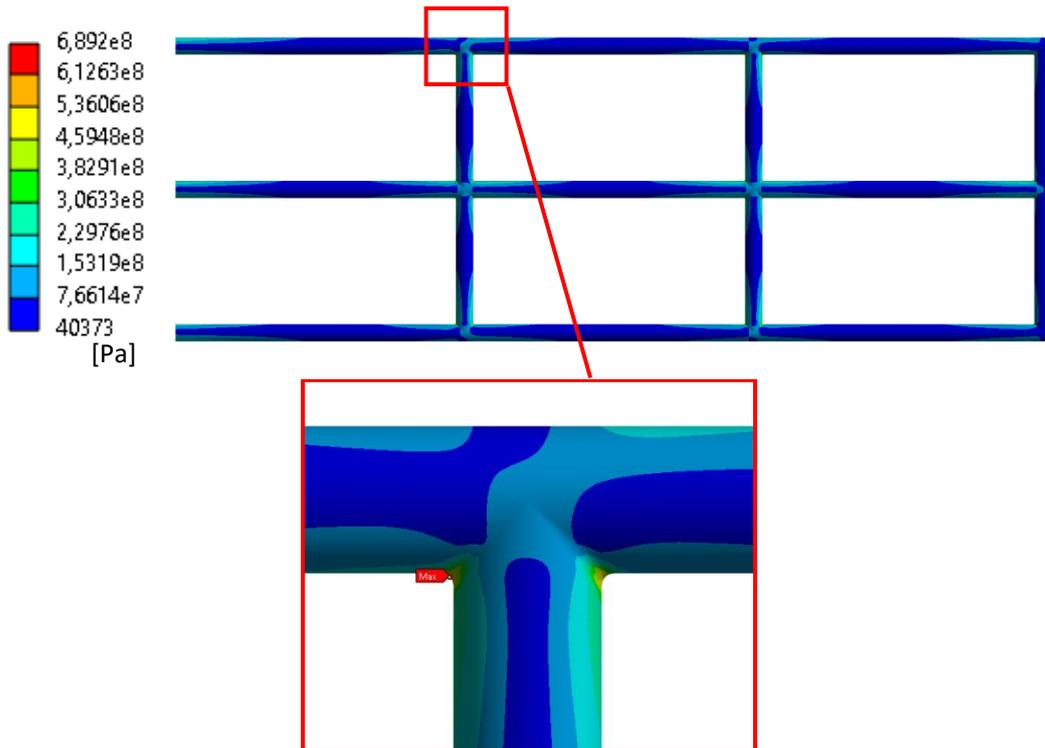


Figure 45: VM stresses

Analysing the von Mises stresses, it can be seen that the material is not exploited to the full, in fact there are unloaded zones and stress peaks at the joints. The analytical method described was applied to optimize the beam section by imposing a uniform resistance stress of 250 MPa, resulting in the geometry shown in Figure 46.



Figure 46: Analytic optimized geometry

It can be seen that the geometry obtained has two limitations for a CAD representation:

- There is no continuity of section in the joining zones of two or more elements;
- There are areas of zero cross-section.

For this testcase, mesh morphing was used to create the FEM model of the analytically optimized geometry. From Timoshenko's beam theory, by imposing the uniform strength condition, the analytical relation of the optimal section can be obtained (Eq. 4.10). Eq. 4.10 is computed in a local reference system of each beam element and originating at the node with the lowest node index.

Noting this correlation, it is possible to create a point cloud from the initial geometry (constant radius section) to the optimized geometry. At this stage, the only critical aspect to consider is the transition from the local reference system of each beam, in which the correlation of $R(x)$ is defined to the global one. Furthermore, in order to have good mesh quality and to avoid the problem of continuity in the connection zone, buffers are created between each beam in which no points are defined. Finally, a minimum section is set. Figure 47 shows the point cloud used for this case (36000 points). RBF solution with 48 cores CPU requires more or less 2 minutes

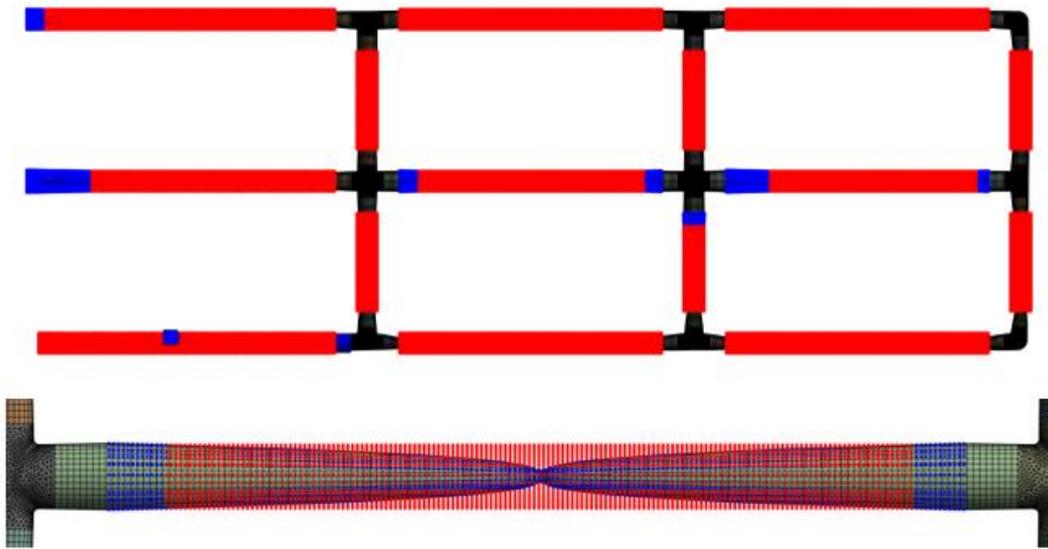


Figure 47: Cloud of points, red on the baseline, blue on the new shape

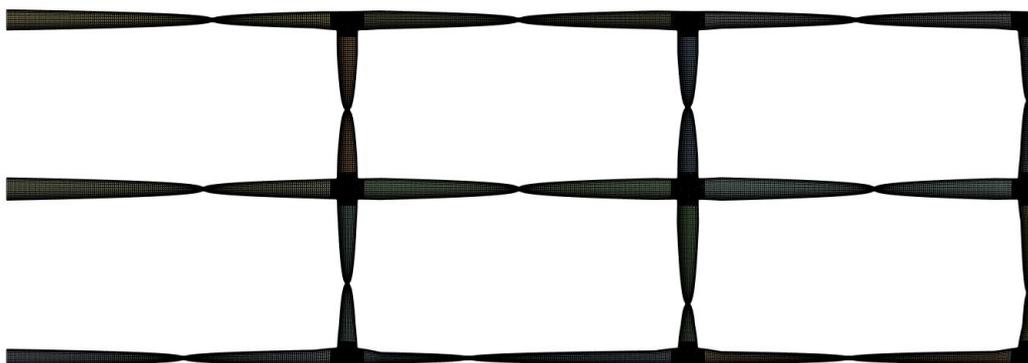


Figure 48: Step 1 optimized mesh

This point cloud is used as an RBF field. In fact, using RBFs, the displacements defined on the source points are mapped onto the mesh nodes. In this way, the optimized geometry can be obtained automatically.

Analysing the von Mises stresses, it can be seen that there is a better utilization of the material, in the areas away from the neutral section the stress is about equal to the imposed uniform resistance stress, but there are problems in the striction zones and near the fittings, where the analytical method used does not give a valid solution.

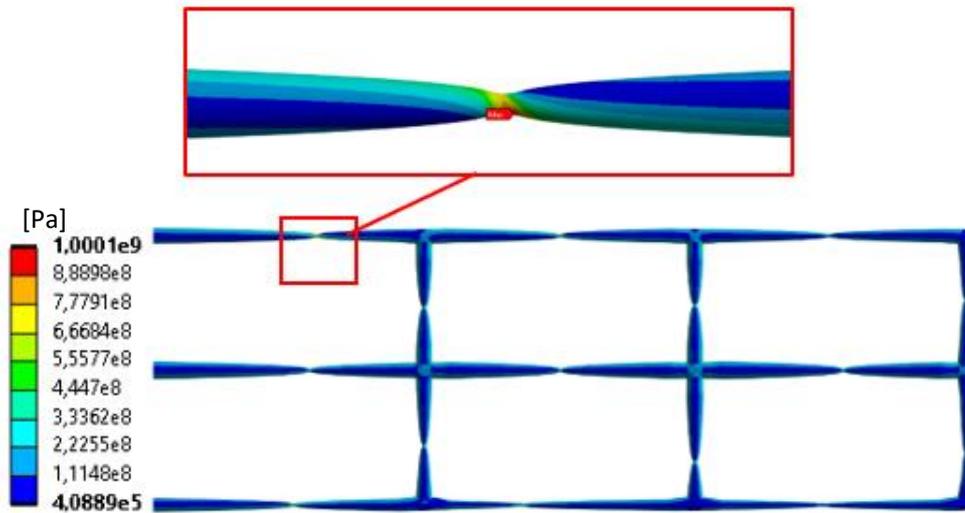


Figure 49: VM stress on step 1 optimized geometry

In order to solve problems at the edges where the Timoshenko beam theory is not valid, and thus to achieve an optimized 3D geometry, BGM was used. BGM was only applied in the critical zones, i.e. zones of striction and junction between several elements. The following image compares the geometry of step 1 and the final geometry obtained with BGM. It is emphasized that BGM is an iterative method, so in each step the mesh is updated according to the measured surface stress. It is evident, therefore, how starting from an already optimized geometry and intervening only in certain areas allows the time required for optimization to be considerably reduced.

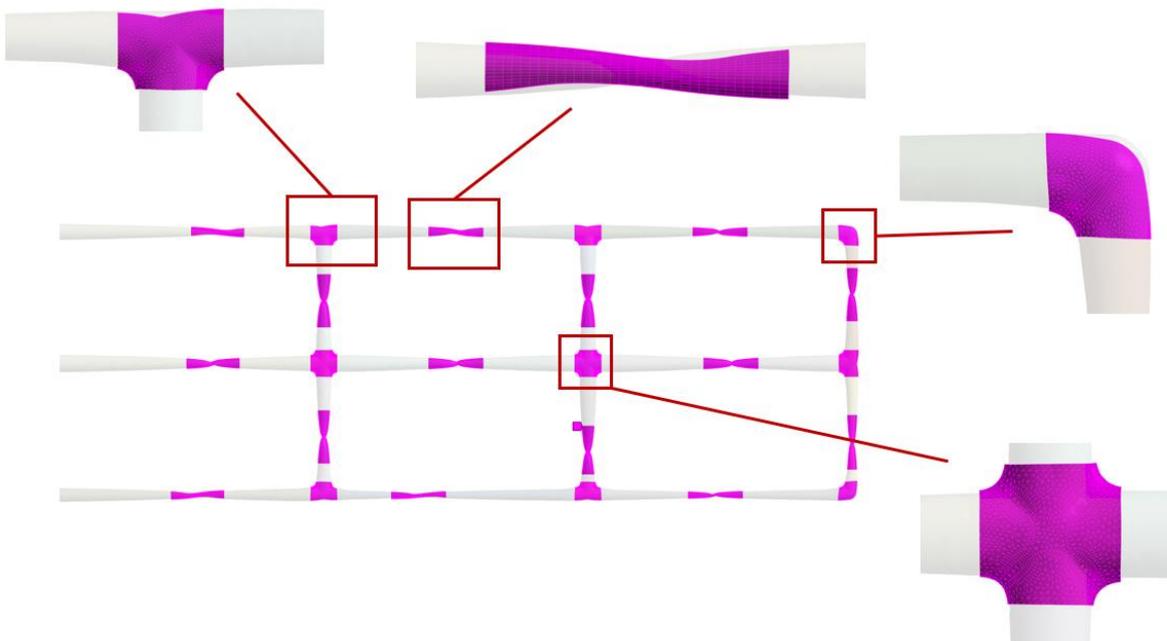


Figure 50: Comparison of step 1 and step 2 optimized geometries

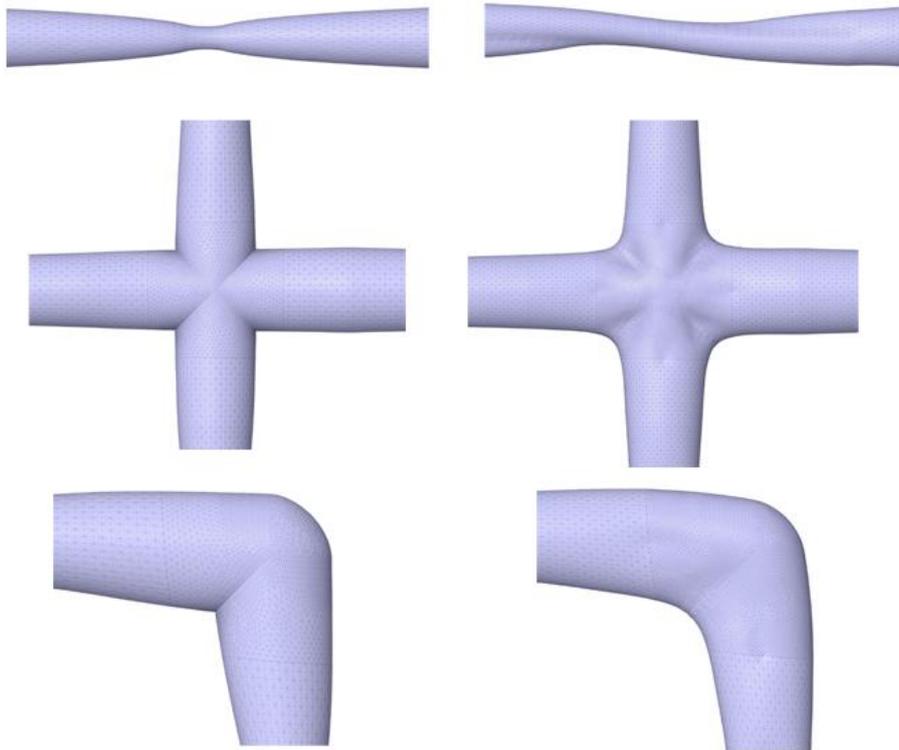


Figure 51: Comparison of step 1 and step 2 optimised geometries

Analysing the von Mises stresses, it can be seen that overall, there are no major differences, but the main criticalities are resolved, material is added in the squeeze zones and the fillets are smoothed, excess material is removed (especially in the z-direction) resulting in organic and more efficient shapes.

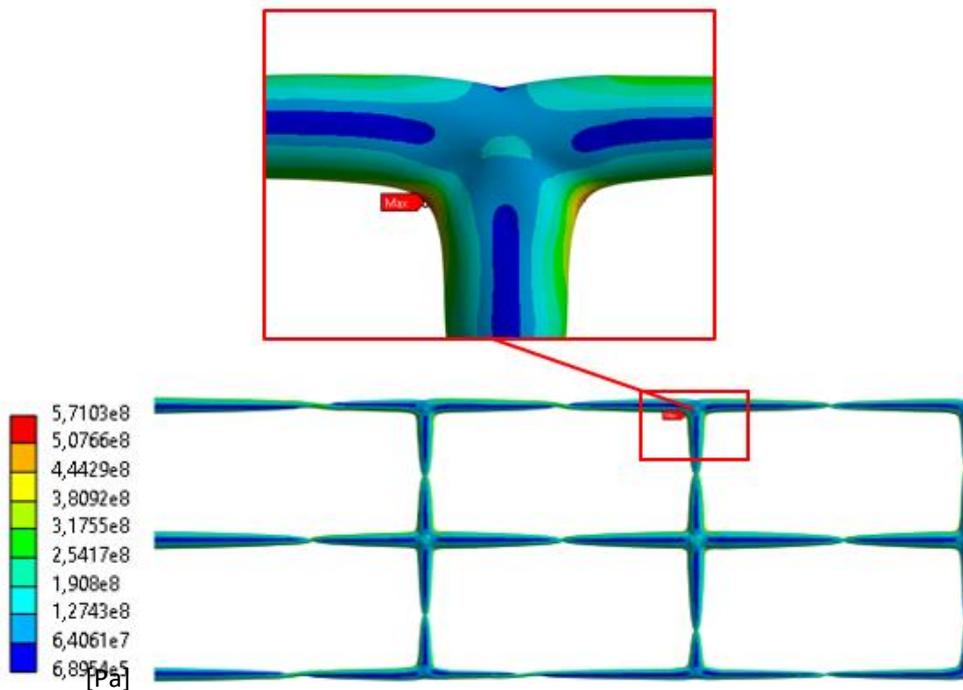


Figure 52: VM stress on optimised geometry

To compare the results, the values of f_1 at each node and the scalar values of the factors f_1 , f_2 and f_3 were compared (Eq. 4.12,4.13,4.14). In particular, these values are scalars, so the corresponding nodal quantities have been plotted for the plot. From the contours of f_1 it can be observed that from baseline to step 1 the material utilization improves greatly, but there are some problem areas that are completely solved in step 2. Analysing the scalar values in Table 13 and 14, a drastic reduction of the f_1 and f_2 factors can be observed, the volume (and thus the weight) decreases a lot. The energy factor also improves greatly due to the combined effect of the volume reduction and the increase in deformation energy.

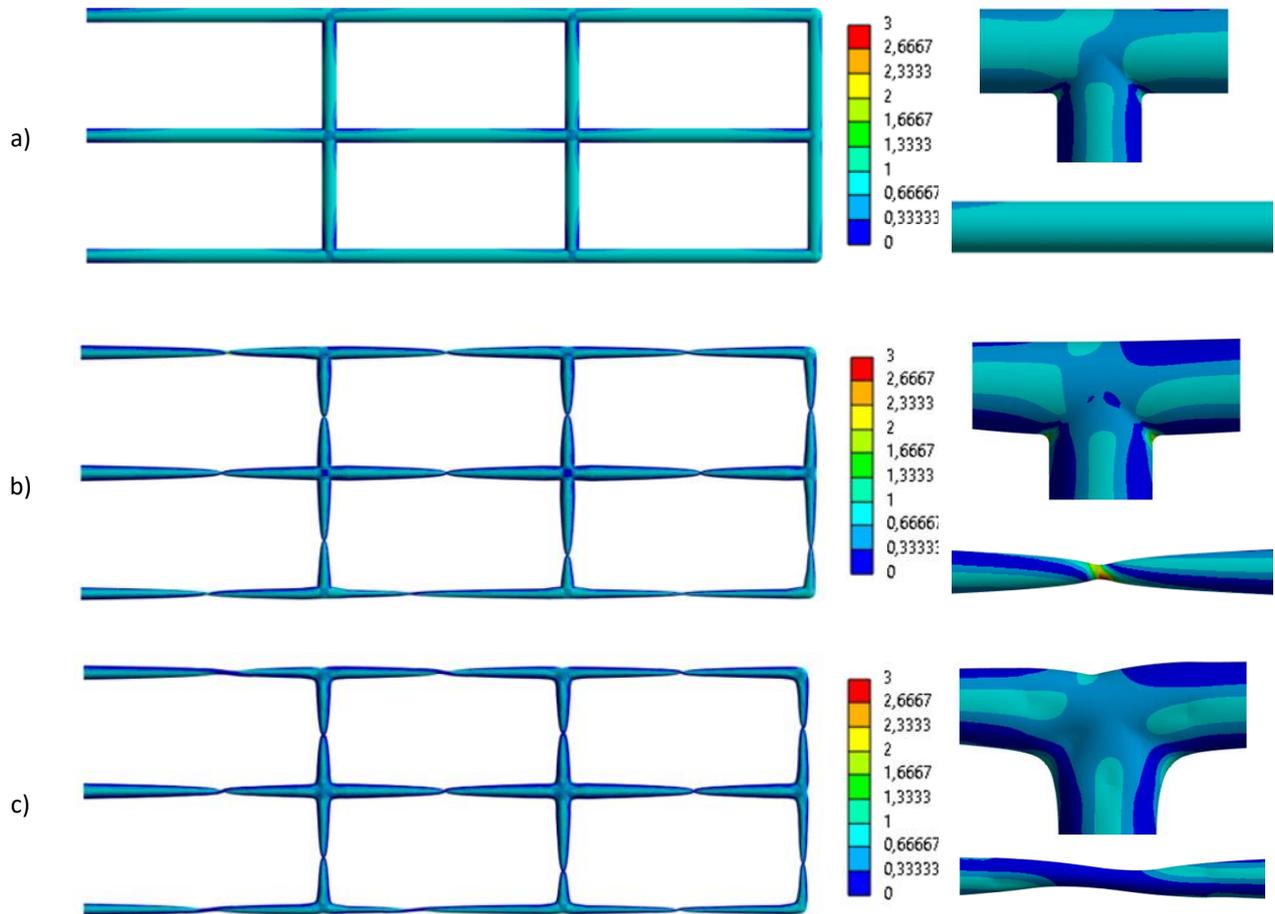


Figure 53: Comparison of f_1 factor of a) baseline; b) step 1 optimized shape; c) step2 optimized shape

Table 13: Comparison of stress factors

	VM_{max} [Pa]	Vol [m^3]	f_1	f_2
Baseline	$6.9e8$	$4.16e-003$	0.72	0.8
Step 1	$1e9$	$2.24 e-003$	0.53	0.64
Step 2	$5.7e8$	$1.95 e-003$	0.48	0.612

Table 14: Comparison of energy factors

	Strain energy	f_3
Baseline	$48.221 J$	0.075
Step 1	$98.108 J$	0.28
Step 2	$102.23 J$	0.335

4.1.5. Case 2: Lattice structure

The second case analysed is a lattice structure. Only one layer was considered and a pure shear condition was applied.

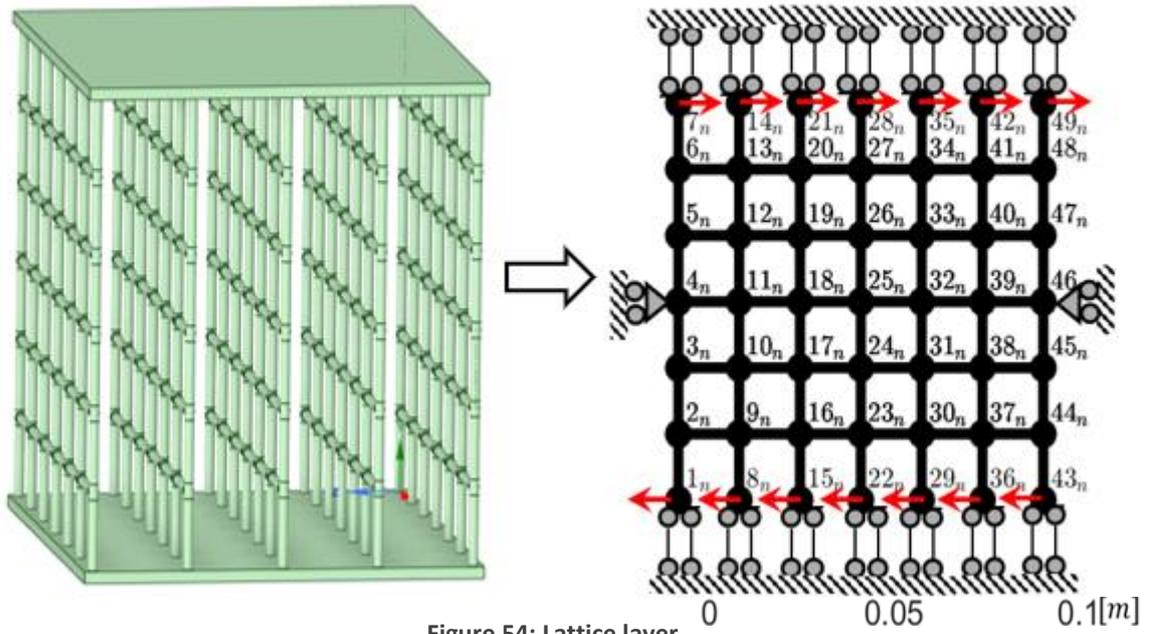


Figure 54: Lattice layer

Analysing the von Mises stresses (Figure 55), it can be seen that some areas are unloaded, while some stress peaks are present. Therefore, the material is not fully utilized and we are very far from a condition of uniform resistance.

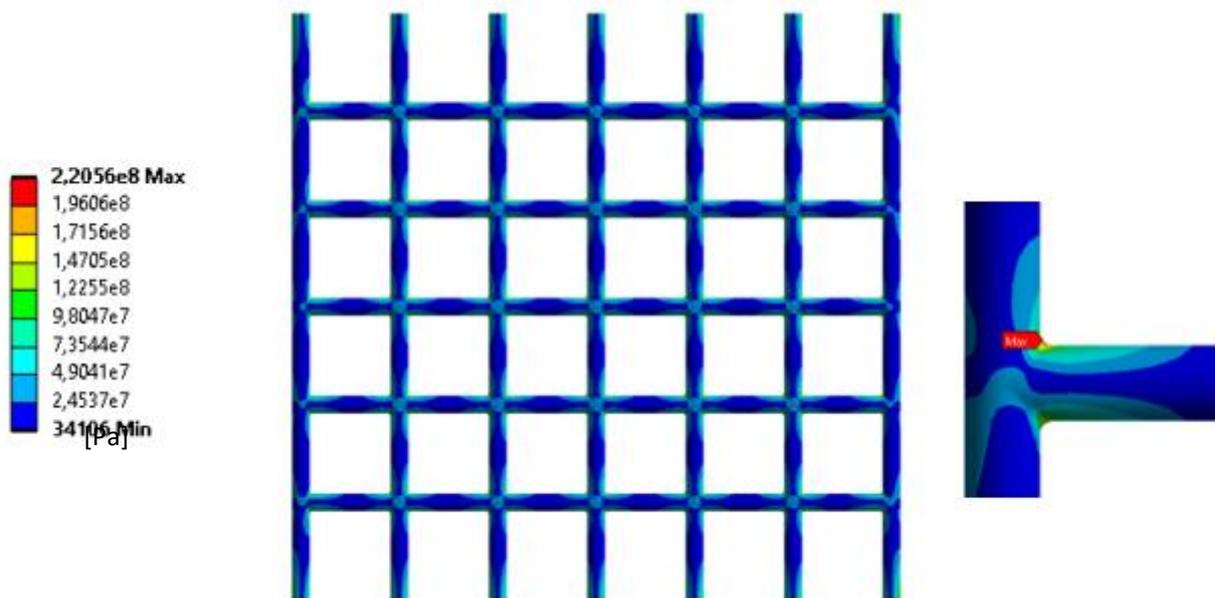


Figure 55: VM stress on baseline

As in the previous case, the analytical method was applied, obtaining a correlation between the radius of each beam and the x-axis along the beam axis. This correlation was used to create the point cloud to obtain the optimized geometry. As in the previous case, buffers were created between two successive beam elements and a minimum section was set. RBF points are used to deform the mesh.

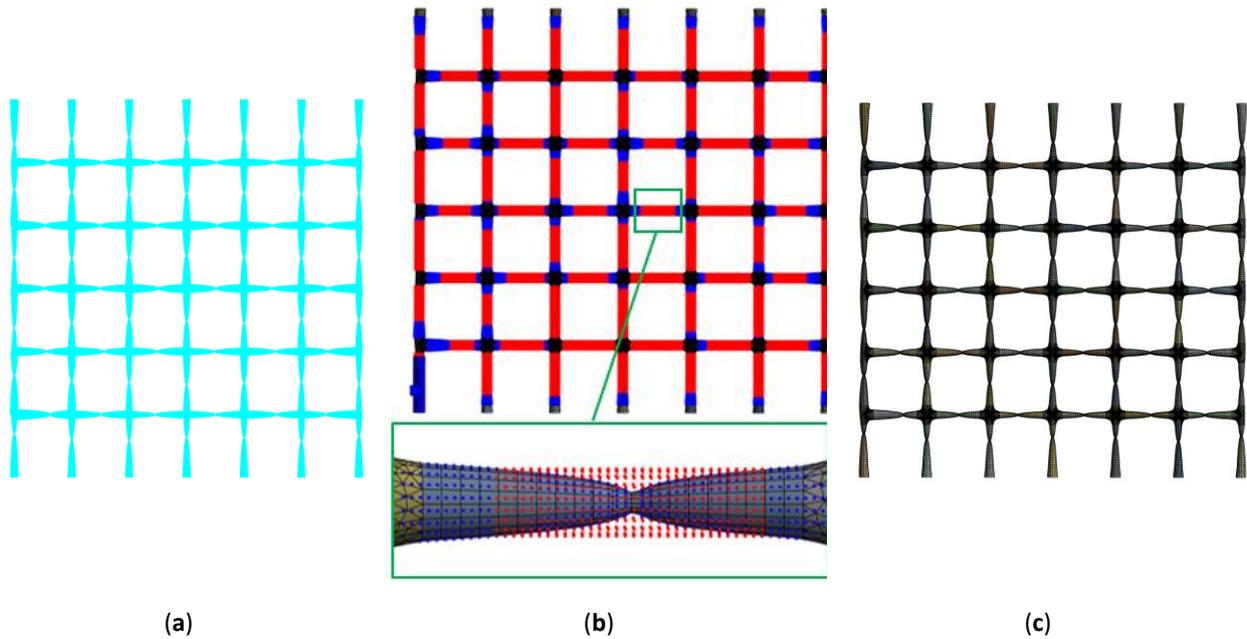


Figure 56: a) Analytical solution; b) RBF points; c) Step 1 optimized mesh

Looking at the von Mises stresses (Figure 57), it can be seen that excluding the neutral section zone, the stresses are close to the imposed iso-resistance value. However, there are critical zones with stress peaks at strictions and junctions of several elements.

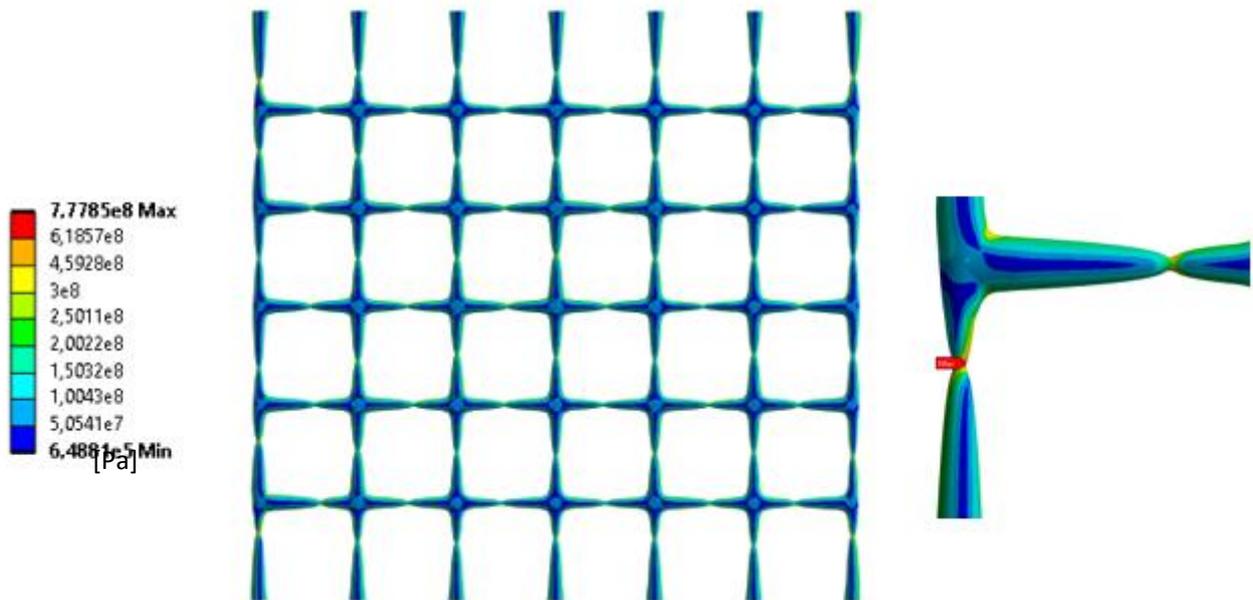


Figure 57: VM stress on Step 1 optimized shape

As in the first case, BGM was used to resolve critical areas and further optimize the geometry, adding material where the surface tensions are very high and conversely removing material where it is not needed. A comparison of step 1 and step 2 geometry is shown in the figures below. It can be seen how the BGM acts mainly in the strictions and fillets, resulting in a more organic geometry, with less volume (and therefore weight), with much reduced tension peaks.

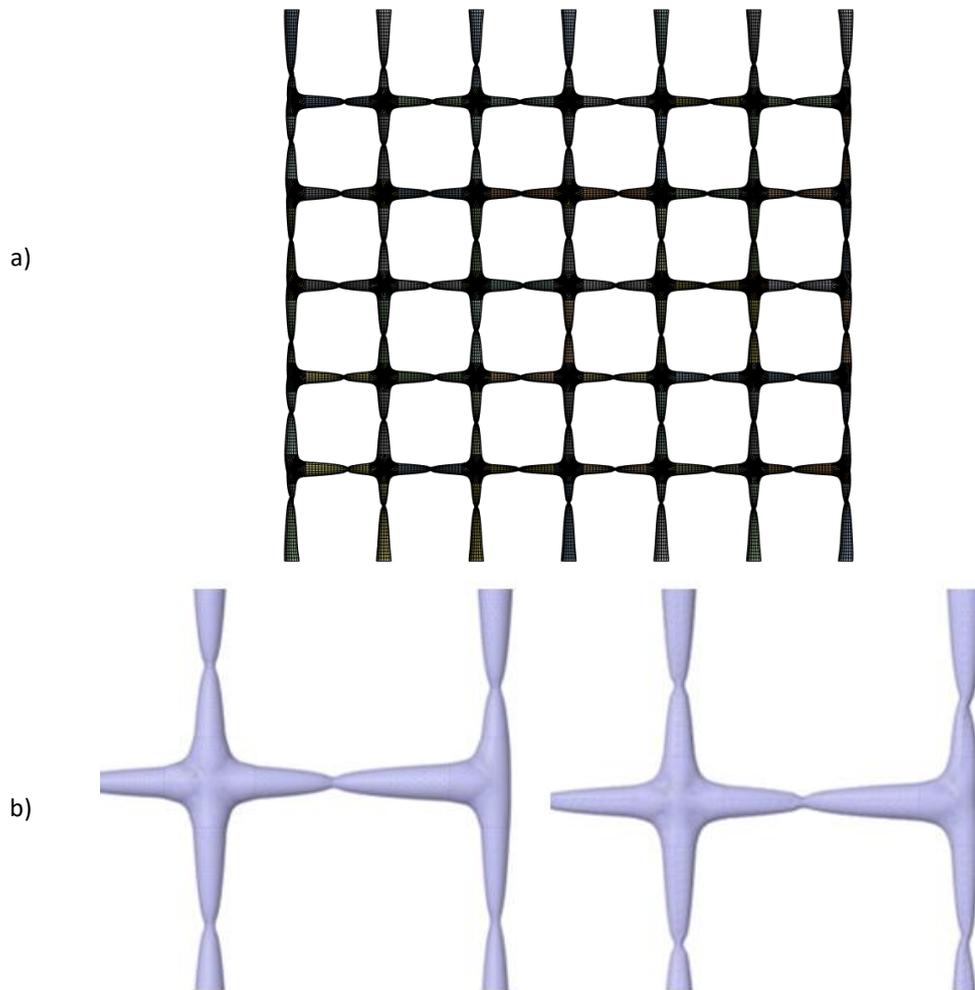


Figure 58: a) BGM optimized shape; b) Comparison of step 1 (left) and step 2 (right) geometry

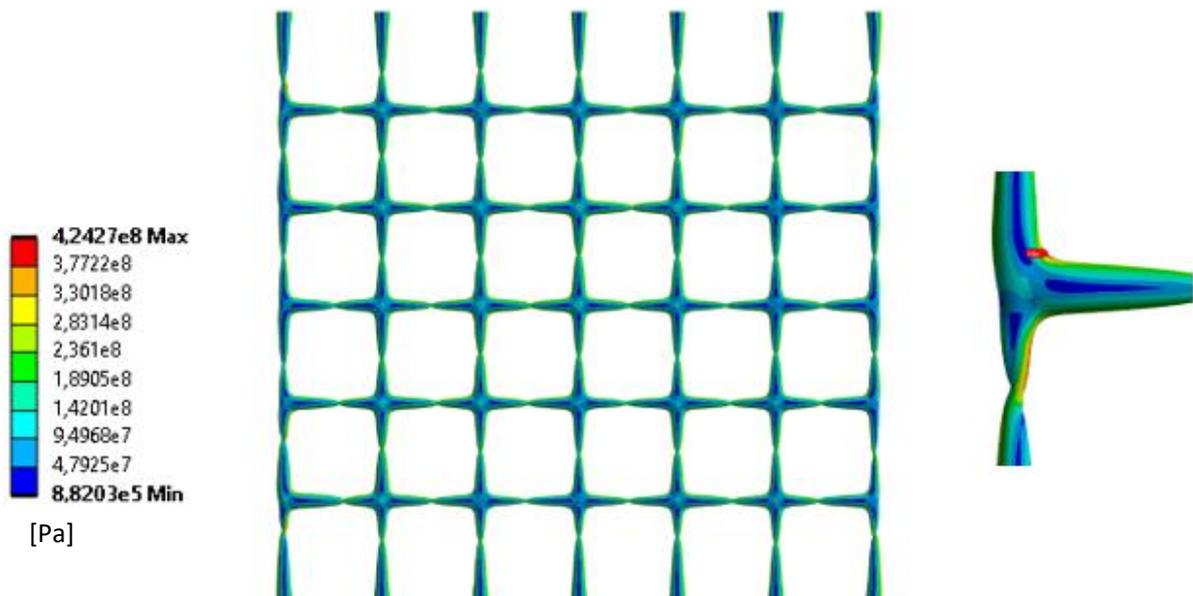


Figure 59: VM stress on step 2 optimized shape

To compare the results, the nodal values of f_1 and the scalar values of factors f_1 , f_2 , and f_3 were analysed. The f_1 contour plots show a significant improvement in material utilization from the baseline to step 1, with some

problem areas fully resolved in step 2. A closer look at the scalar values in Tables 15 and 16 reveals a substantial reduction in both f_1 and f_2 , accompanied by a significant decrease in weight. The energy factor also improves considerably, thanks to the combined effect of volume reduction and increased deformation energy

Table 15: Comparison of stress factors

	VM_{max} [Pa]	Vol [m^3]	f_1	f_2
Baseline	$2.2e8$	$7.9553e-006$	0.85	0.87
Step 1	$7.8e8$	$2.8534e-006$	0.6	0.69
Step 2	$4.2e8$	$2.3051e-006$	0.55	0.64

Table 16: Comparison of energy factors

	Strain energy	f_3
Baseline	$1.8075e-002$ J	0.014
Step 1	$9.1444e-002$ J	0.2
Step 2	0.10943 J	0.3

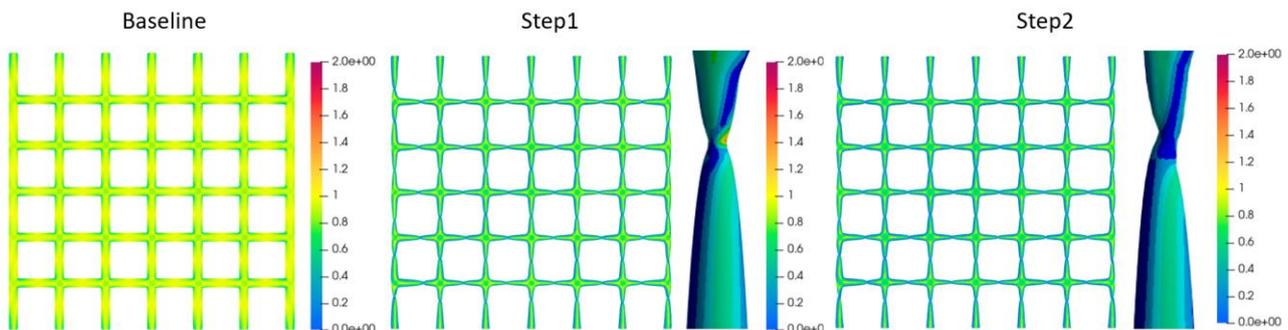


Figure 60: Comparison of f_1 factor contour plot

4.1.6. Case 3: Gear body

The final case analysed is that of a PEEK gear. In this instance, a slightly different approach was used. The first step remains the 1D optimization using Timoshenko beam theory. In this case, it must be considered that only one tooth is engaged, but the engaged tooth changes as the gear rotates. Therefore, both in the analytical and numerical optimization, an iterative approach was employed: in each step, the optimized shape is calculated only on the circular sector that includes the engaged tooth, the optimized geometry is evaluated, and then propagated cyclically over the entire gear. This way, the stiffness of the entire gear is updated, and the next optimization step is performed. Furthermore, compared to the previous cases, a 2D model was used for the numerical analysis in this test case, creating the CAD directly from the configuration obtained through numerical optimization.

For the analysis, only the gear body was considered. The baseline geometry was obtained by arranging beams in a reasonable manner. Of course, various configurations can be explored, but in this work, the focus is not on layout optimization or topological optimization, but on shape optimization. The following figures show the baseline model used. Specifically, the nodes on the innermost circumference are considered fixed, while at the node corresponding to the engaged tooth, the following is applied:

$$F_1 = -1,303e + 02 \text{ N}$$

$$F_2 = 3,581 + 02 \text{ N}$$

$$F_3 = 2.511 \text{ Nm}$$

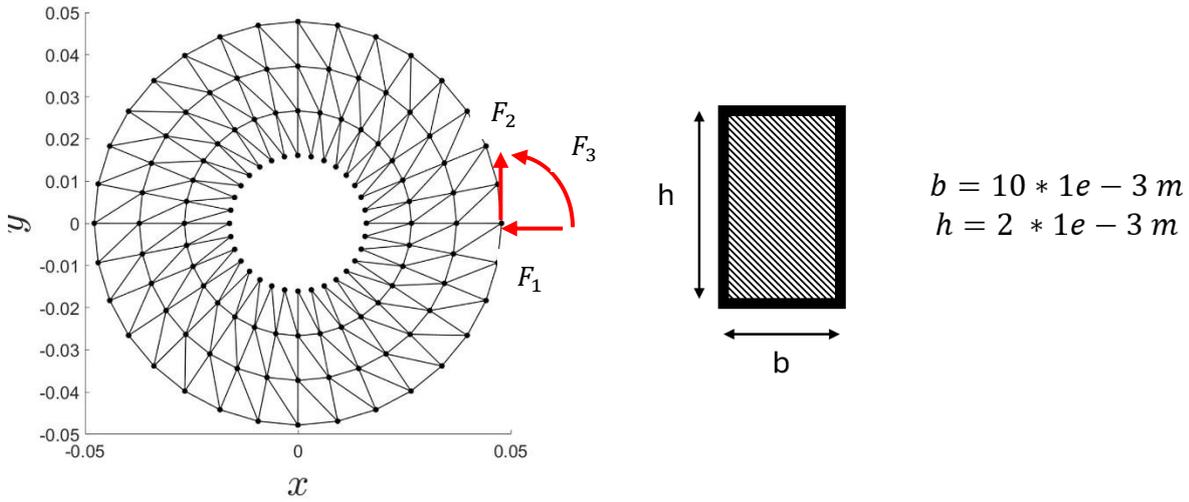


Figure 61: Gear body baseline

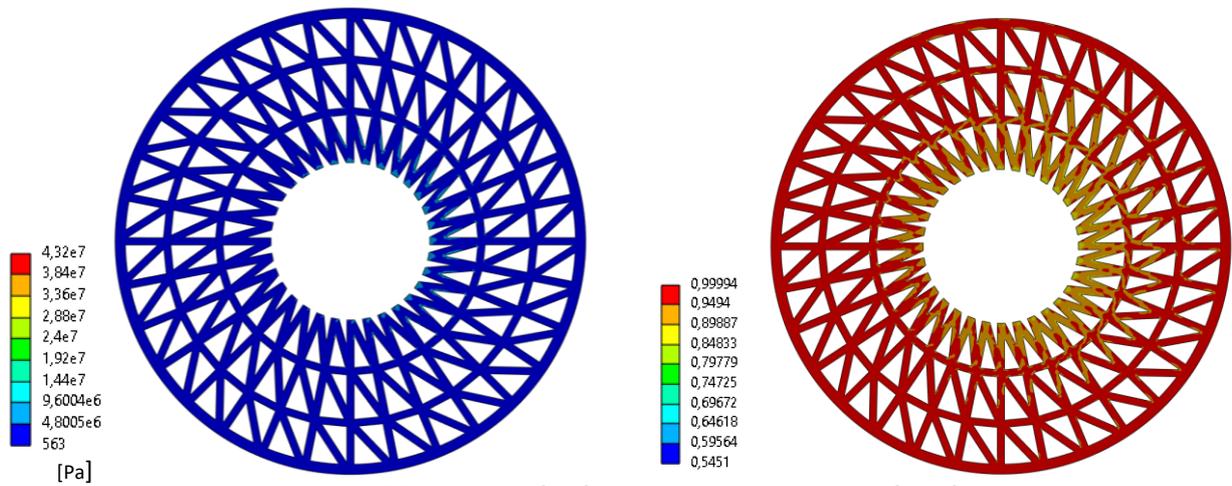


Figure 62: VM stress (left) and f1 factor contour plot (right)

As mentioned in the previous paragraph, in this case, having a 2D geometry, a different approach was chosen. In the initial analysis, the same analytical approach was used to identify the optimized geometry. It is important to note that the method is iterative, and at each optimization step, only the circular sector of the engaged tooth is analysed, with the modifications then propagated across the entire gear body before proceeding to the next step. By solving equation 4.10 for a rectangular section, the following result is obtained:

$$h(x) = \frac{|c_1|}{2b\sigma_0} + \sqrt{\left(\frac{c_1}{2b\sigma_0}\right)^2 + \frac{6|-c_3+c_2x|}{b\sigma_0}} \quad (4.15)$$

This relation is used to generate a CAD model using a Python-based extension of the Ansys Spaceclaim. Therefore, compared to previous cases, morphing is only used for implementing the BGM and not for reproducing the optimized geometry with the analytical method. However, as in the previous cases, the method is fully automated. The figures below show the optimized geometry and the results in terms of VM stress and f_1 factor.

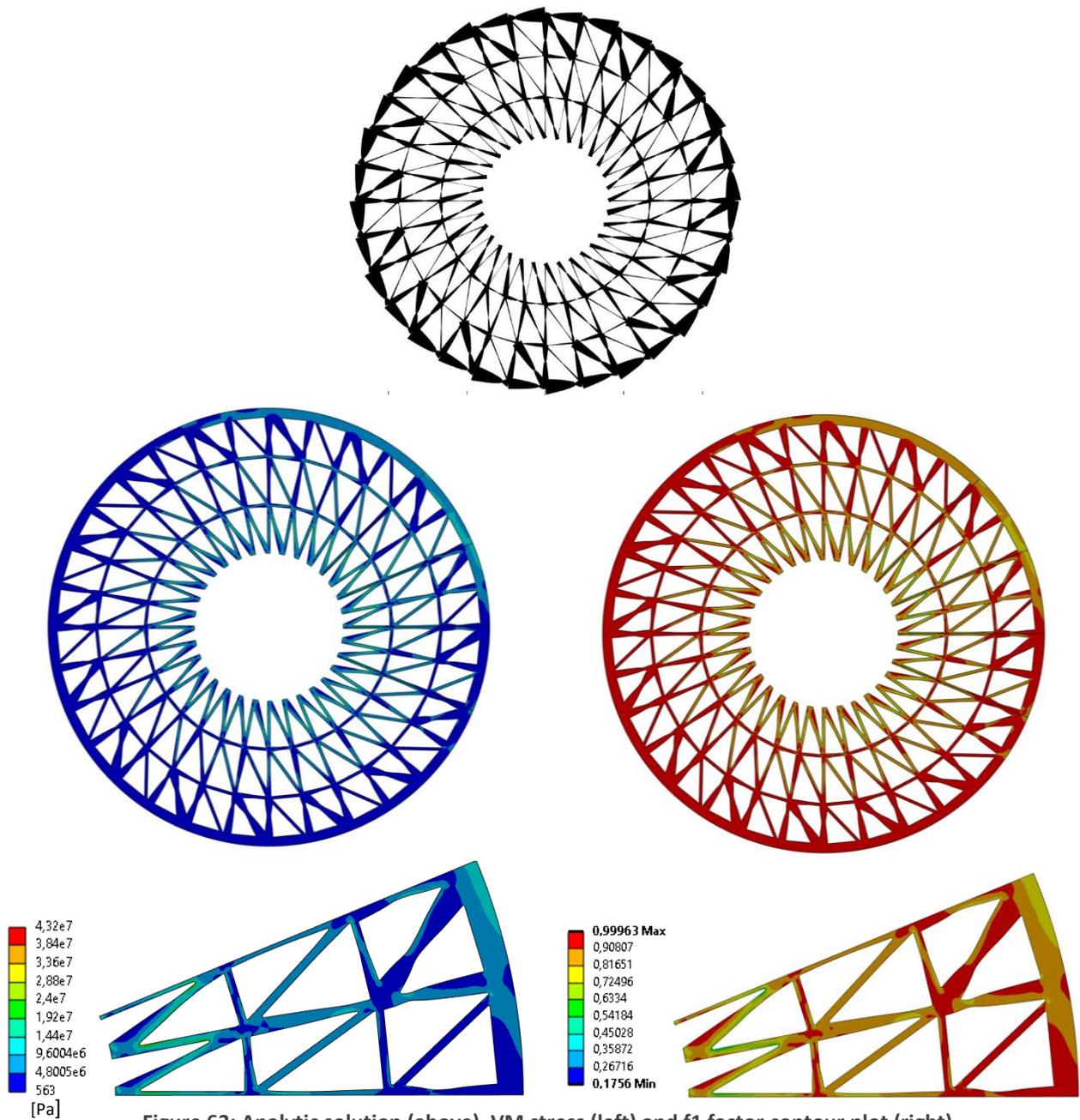


Figure 63: Analytic solution (above), VM stress (left) and f1 factor contour plot (right)

The final step is the application of the BGM. A circular sector was identified where, when the tooth is engaged, stress peaks occur. The surface stresses were calculated and used to guide the optimization towards a uniform strength stress target, which was set at 50MPa. The displacements calculated on one circular sector were then propagated over the entire gear. It is important to note that in this case, being a 2D model, the 'surface' stresses are actually evaluated on the edges, and the direction of the displacements is along the normal to the curves.

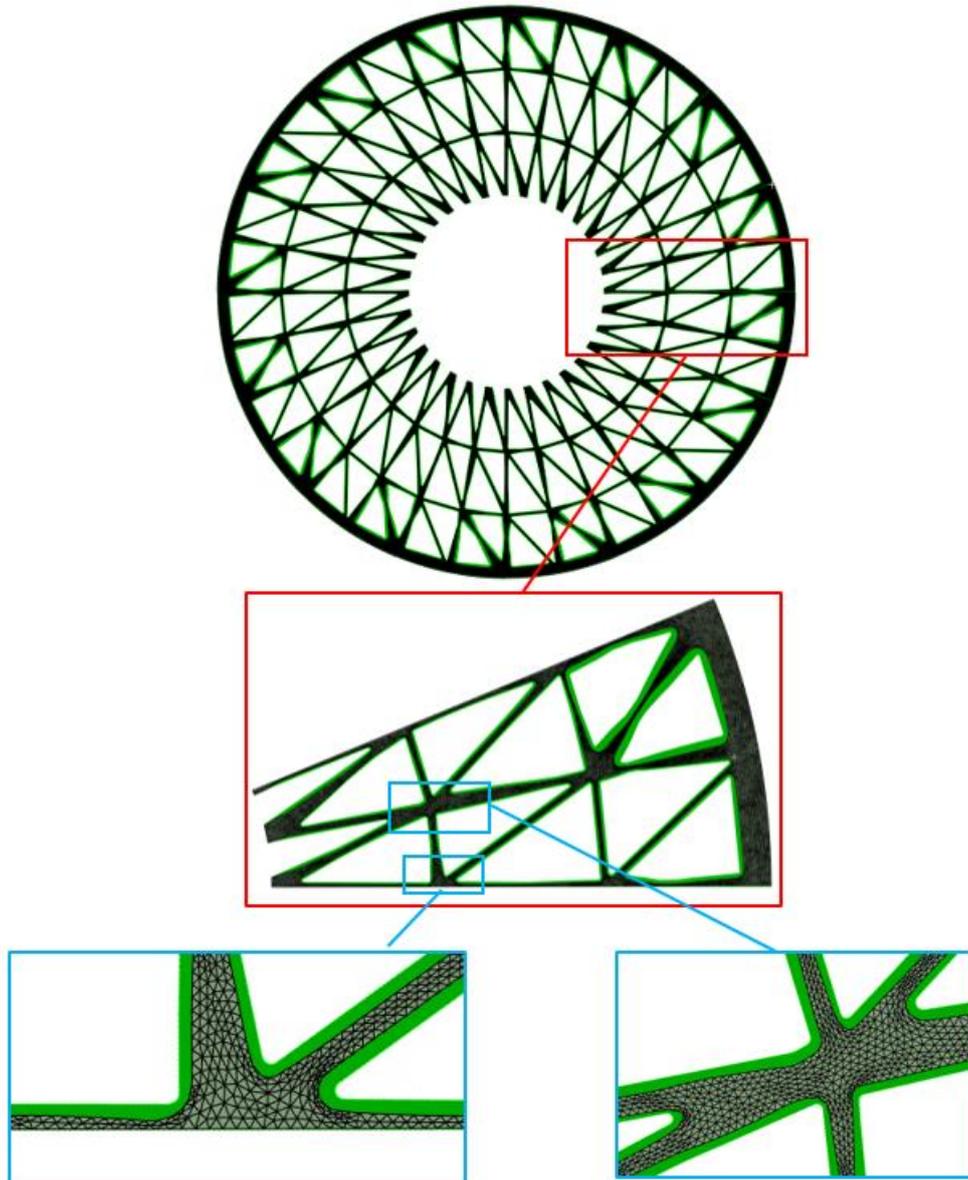


Figure 64: Comparison of step 1 (green) and step 2 (grey) mesh

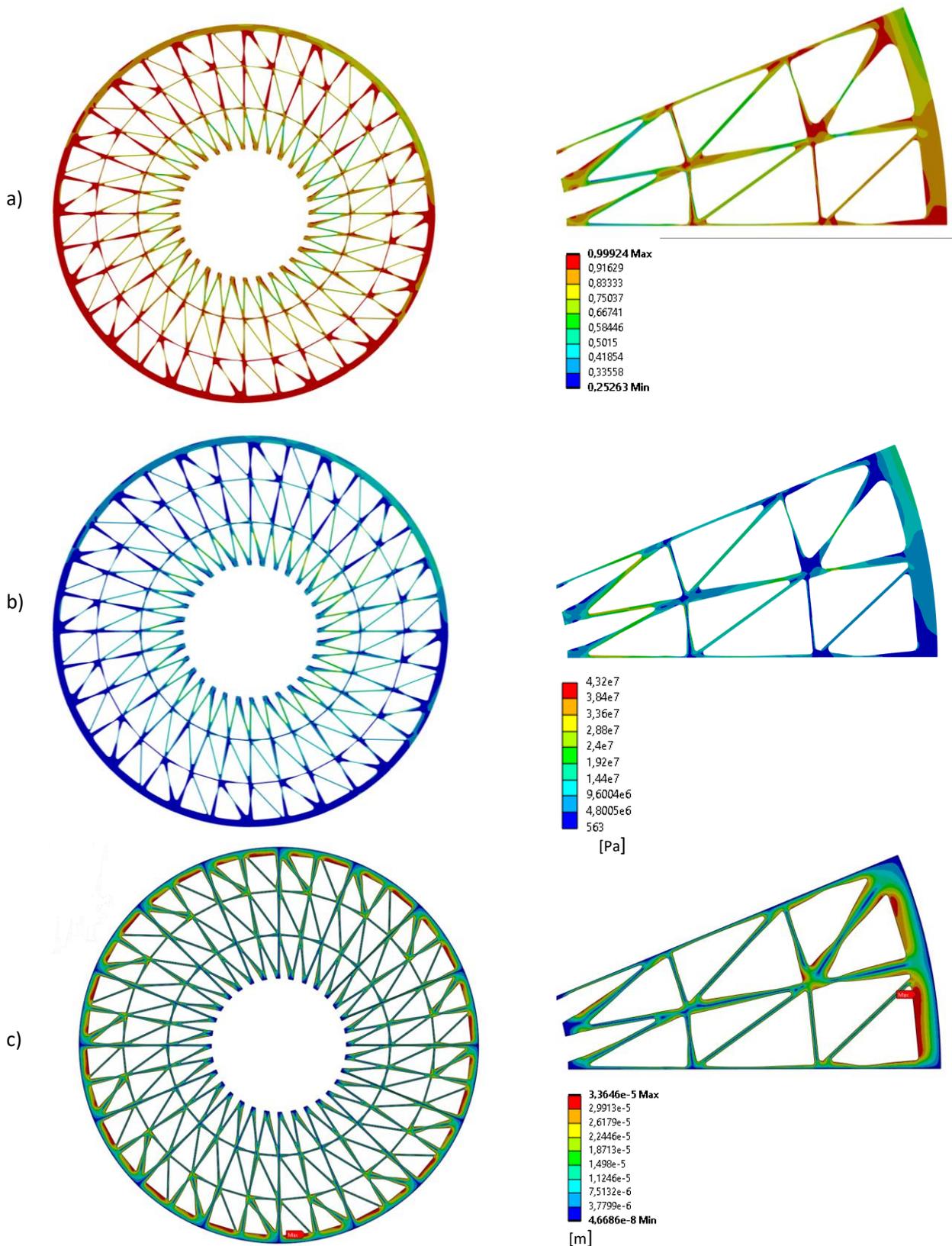


Figure 65: BGM displacements (c), VM stress (b), f_1 factor contour plot (a)

Comparing the results, it can be observed that in this final case, the action of the BGM is very significant, as the entire gear is oversized relative to the design conditions. The final geometry allows for better material utilization, but the f_1 , f_2 and f_3 factors remain low compared to previous cases because we are optimizing only a circular

sector. On the other hand, this approach is essential, considering that, alternatively, all the various areas would be stressed under the considered load.

Table 17: Comparison of stress factors

	Vol [m^3]	f_1	f_2
Baseline	<i>3.723e-005</i>	<i>0.95</i>	<i>0.96</i>
Step 1	<i>2.4385e-005</i>	<i>0.89</i>	<i>0.92</i>
Step 2	<i>1.5765e-005</i>	<i>0.81</i>	<i>0.86</i>

Table 18: Comparison of energy factors

	Strain energy	f_3
Baseline	<i>4.157e-004 J</i>	<i>3.75e-05</i>
Step 1	<i>1.6892e-003 J</i>	<i>2.33e-04</i>
Step 2	<i>2.9936e-003 J</i>	<i>6.38e-04</i>

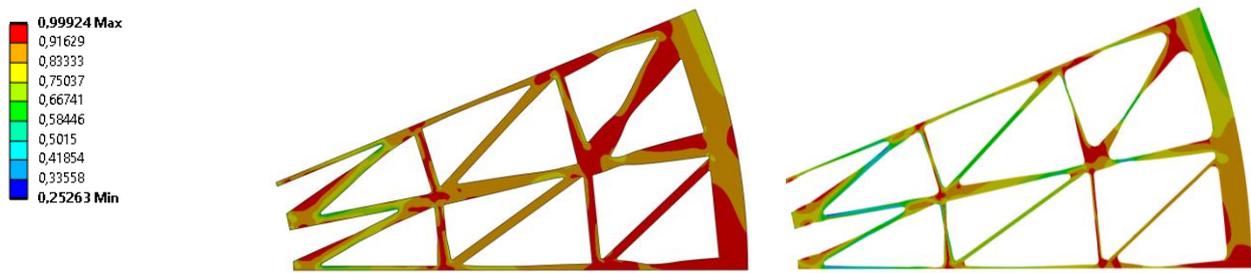


Figure 66: f_1 comparison of analytic solution (left) and BGM (right)

4.1.7. conclusions

This study introduced a novel two-step workflow that integrates both 2D analytical and 3D numerical methods for the optimized design of frame structures. By leveraging the strengths of each approach, the proposed method efficiently addresses the challenge of achieving uniform strength while minimizing material usage and weight. The initial step, based on the Timoshenko beam theory, rapidly generates an optimized geometry for the entire structure using an analytical model. This method is highly effective for uniform-strength profiles but is limited by edge effects in regions where three-dimensional phenomena, such as strictions and joint zones, play a critical role. To overcome this limitation, the second step incorporates the biological growth method (BGM), a bio-inspired numerical approach that refines the optimization process in regions affected by triaxial stress fields. This method ensures that areas near boundaries and joints, where 2D analysis may fall short, are optimized for strength and stability while maintaining the structural topology. The combined approach thus ensures a comprehensive optimization across the entire structure, balancing both efficiency and accuracy in areas of complex stress distribution.

The results of the test cases are promising, demonstrating improved efficiency in terms of stress distribution and energy factors. Optimized geometries generated using this hybrid method exhibit superior performance compared to those designed solely through traditional approaches. This dual approach holds significant potential for advancing the design of frame structures, offering a reliable and efficient tool for engineers in the field of structural optimization and to develop an automated tool that can input any frame structure and output the optimized geometry ready for additive manufacturing.

The analytical model currently applies to planar frame structures but can be extended to 3D structures following the same idea.

5. Adjoint Applications

This paragraph summarizes several activities related to the adjoint method. Specifically, the potential of the adjoint (section 2.3.2.4) has been explored to predict how the model behaves in the vicinity of the operating point. In the context of integration with digital twins, the adjoint method is a very interesting alternative to ROMs because it enables predictions even for "off-design" points. In other words, it allows predictions for any shape parameter without requiring a model training phase.

Three applications have been proposed:

- The first involves using the adjoint method for a constrained optimization problem, enabling automatic sculpting of the model while ensuring that the geometric constraint is respected at each optimization step.
- The second focuses on using the adjoint method to create an interactive design dashboard, where parameters can be defined, and the adjoint is used to predict variations in the observable in real time.
- Finally, the possibility of using the adjoint method to predict fluid-structure interaction coupling has been explored.

5.1. Adjoint sculpting with constraints

This study explored the capability of the adjoint method to predict the evolution of a complex system and its interaction with external constraints, such as packaging or maximum allowable space constraints. As a test case, an exhaust duct was considered, and its geometry is shown in the figure. During the optimization process, the duct must not exceed the packaging constraint (yellow surfaces in the figure 67).

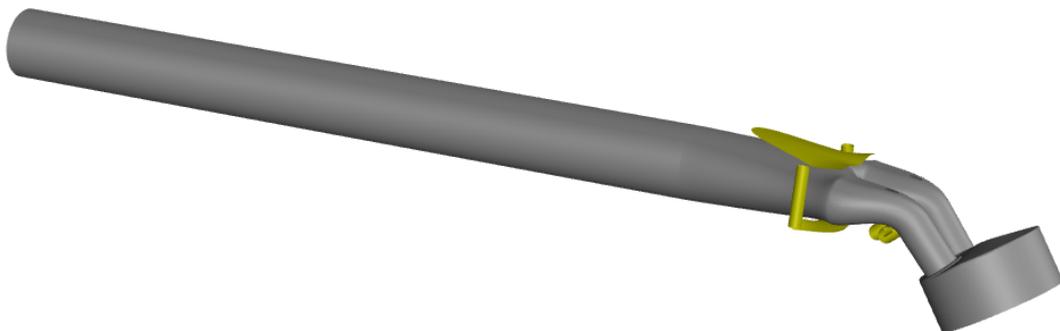


Figure 67: Baseline geometry and external constraints

For the optimization of the duct, the following iterative workflow was used:

- CFD and adjoint calculation:
The sensitivity in the area of interest, namely the region around the two ducts, is computed. At this stage, data smoothing is necessary to obtain sufficiently smooth surfaces and a usable solution.
- Sensitivity-based mesh update:
The sensitivity is used to update the mesh. In this step, each node undergoes a displacement proportional to the sensitivity in the direction normal to the surface. RBF mesh morphing is employed to update the mesh.
- Constraint correction:
It is checked that the final solution does not invade the constraint. The constraint is imported as a file with a .stl extension. For each node, the distance to the constraint surface is calculated using the direction of the previous displacement both before and after the adjoint displacement (d_0 and d_1 in Figure 68). If the two vectors are aligned, the constraint has not been violated, and the adjoint displacement is not corrected. Conversely, if the two vectors are misaligned, the constraint has been violated, and the adjoint displacement is corrected by the quantity $|d_1 - d_0|$.
- The workflow is repeated until convergence is achieved.

The analysis was performed in ANSYS Fluent. The adjoint displacements are computed by the solver itself, and RBF Morph was used for morphing.

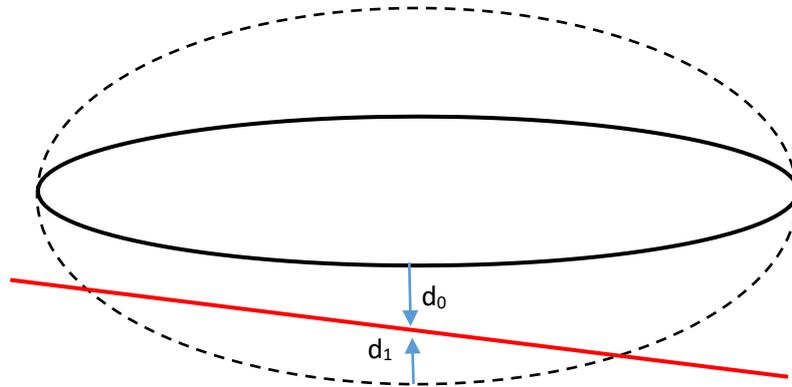


Figure 68: Constraint correction scheme

5.1.1. CFD Model

The mesh used is shown in the figure 69. It consists of mixed tetrahedral and hexahedral elements. Specifically, it includes 1103647 elements, 2583766 faces, and 436900 nodes. Two valves are located within the two ducts.

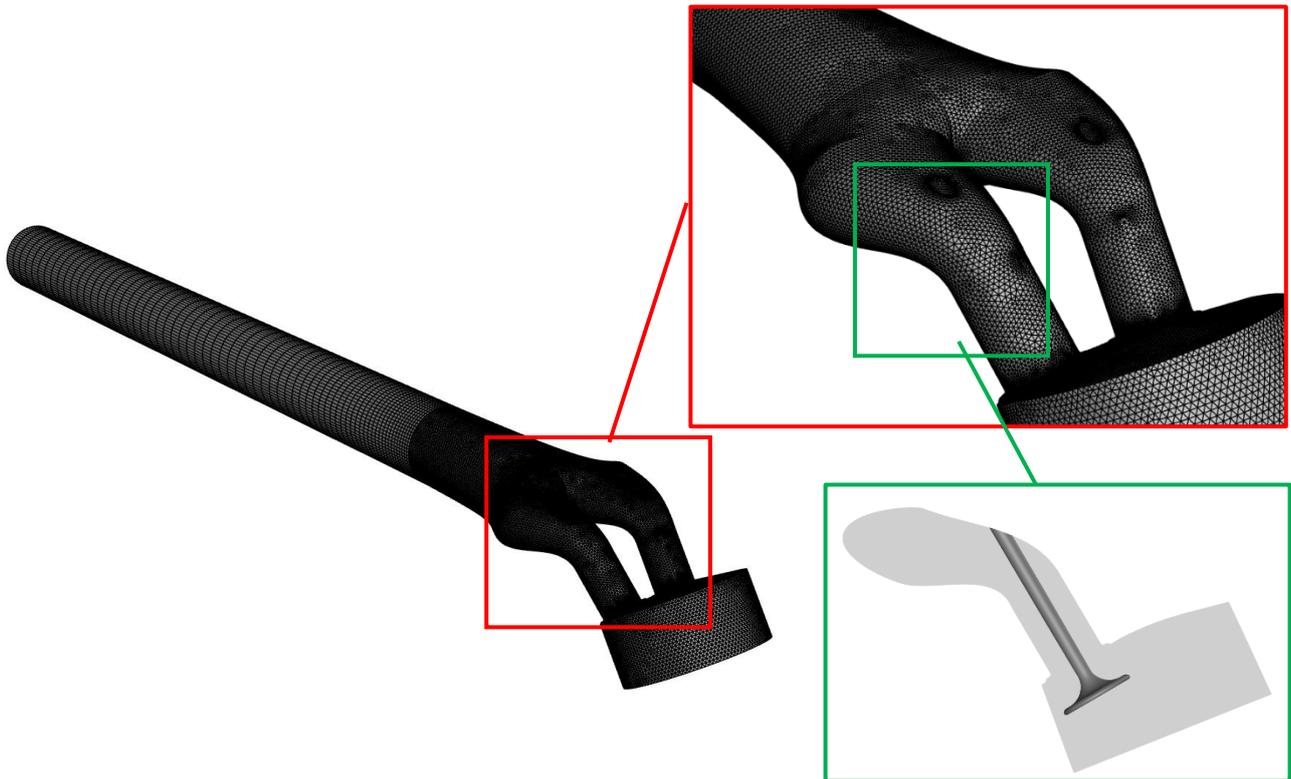


Figure 69: Baseline mesh

The velocity at the inlet is set to 30 m/s, and atmospheric pressure is imposed at the outlet. The objective of the optimization is to minimize the total pressure drop. y^+ is less than 30 and y^+ mean is 12.14

5.1.2. Results

The objective of the optimization is to minimize the total pressure losses by acting only in the area of the two ducts, while keeping the valves fixed. More or less 15000 source points are moved and valves are fixed. Automatic buffers are created between fixed (red points on figure 70) and movable areas (green points). RBF problem is solved in 60s using RBF Morph with 48 cores CPU.

After 15 steps (Figure 74), an improvement of 12.5% in the observable is achieved. Specifically, by observing the velocity and static pressure contours (Figures 71,72,73), it can be noted that the flow becomes more regular, and flow separation is limited. Finally, in Figure 75, it is evident that the constraint is respected, and in the area where the adjoint violated the constraint, the surface is pressed against the constraint itself.

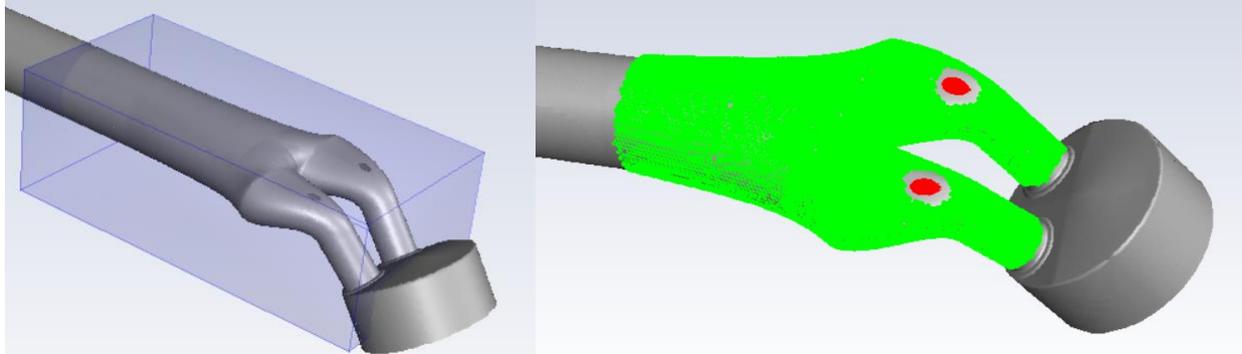


Figure 70: Left, domain box; Right, source points

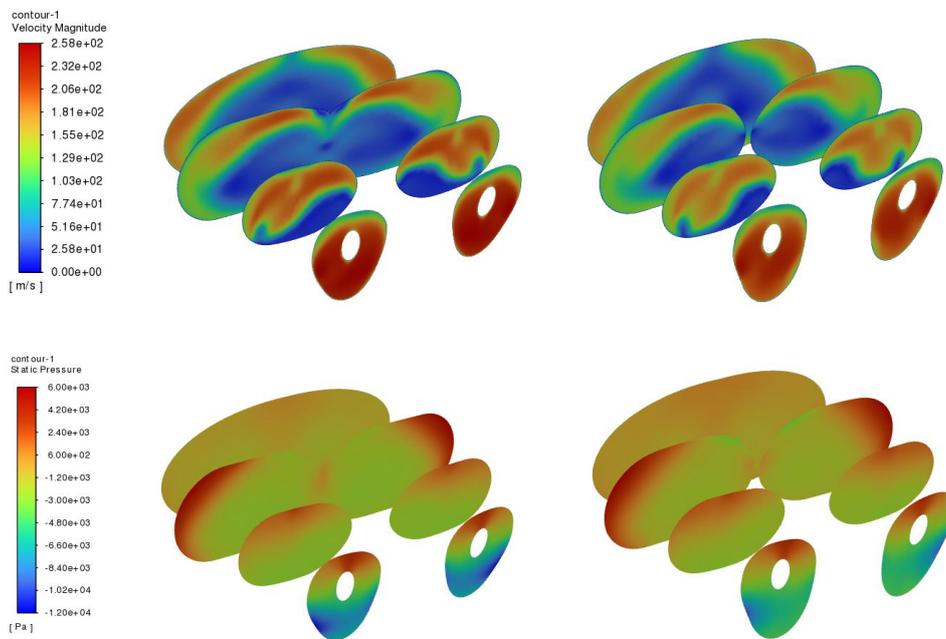


Figure 71: Comparison of velocity ,static pressure contour on baseline (left) and optimized shape (right)

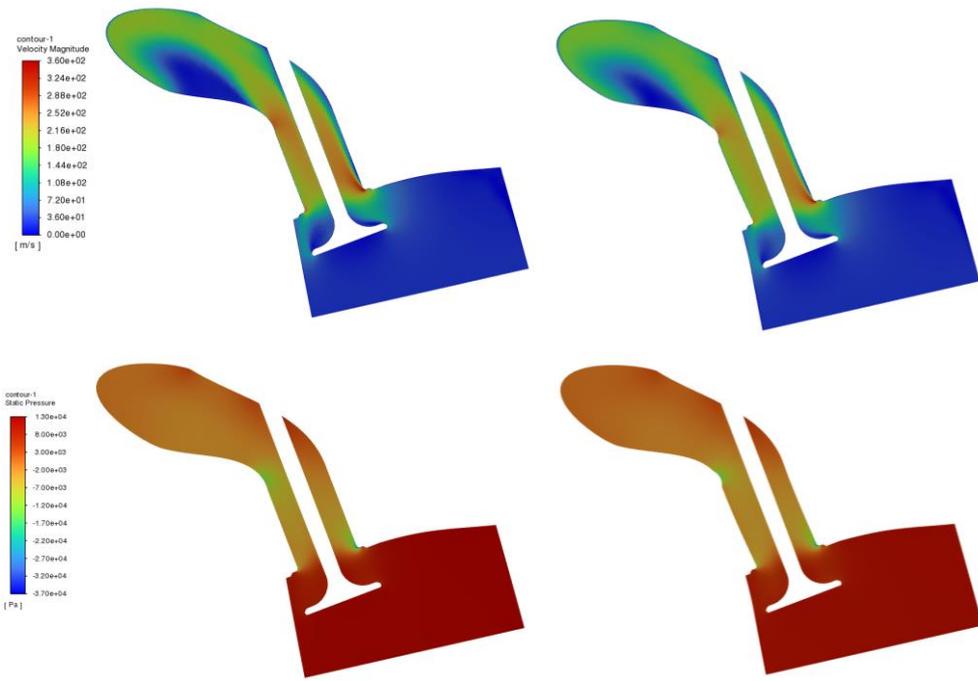


Figure 72: Comparison of velocity and static pressure contour on baseline (left) and optimized shape (right)

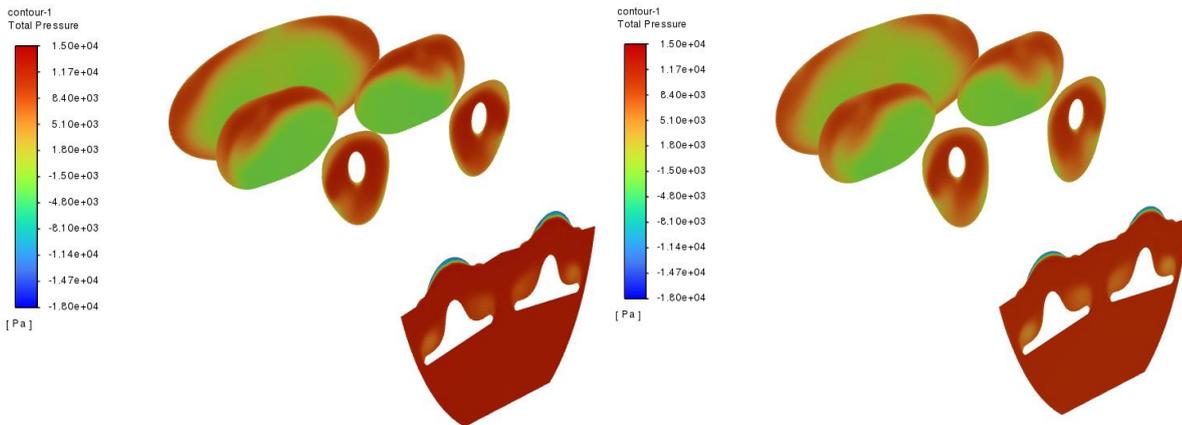


Figure 73: Total pressure comparison. Left baseline, right optimized

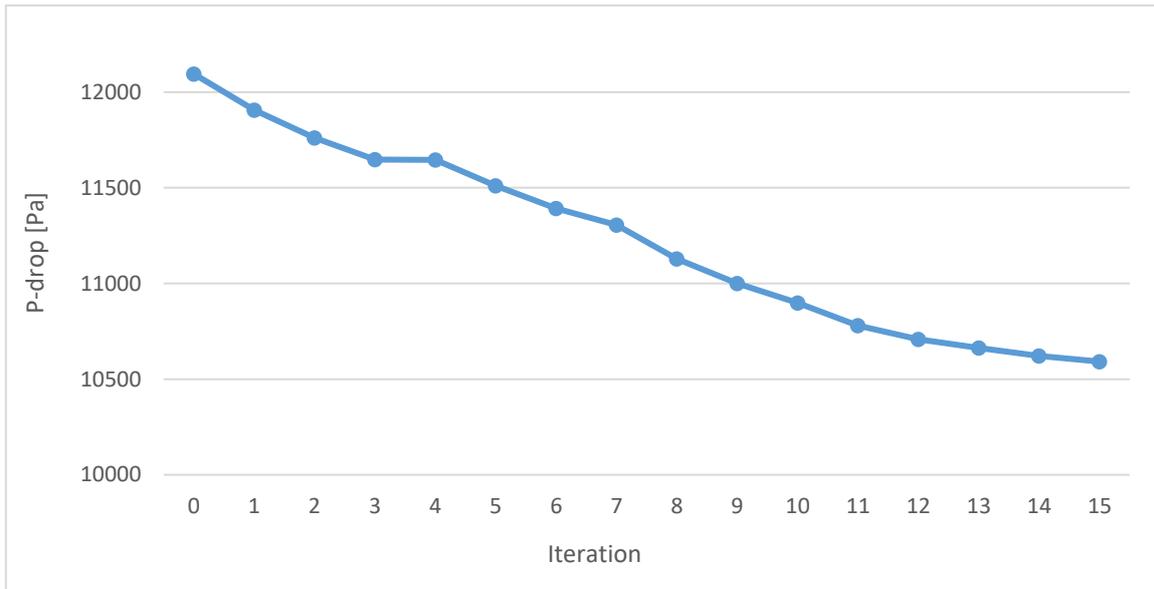


Figure 74: Observable plot during optimisation

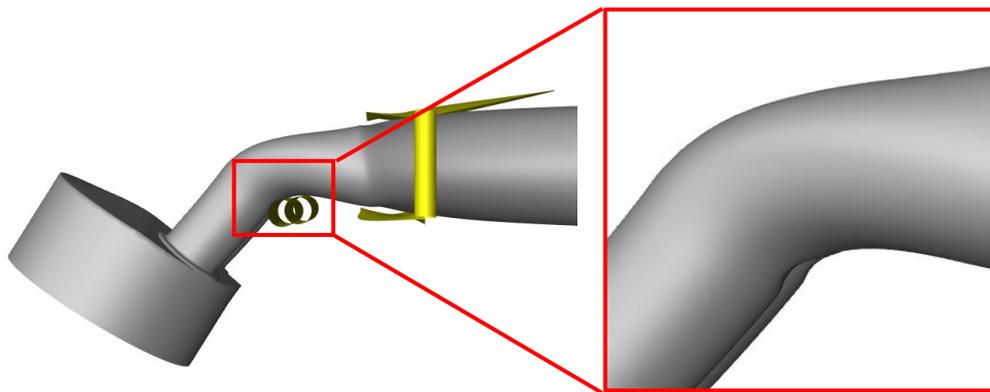


Figure 75: Final shape with focus on constraint zone

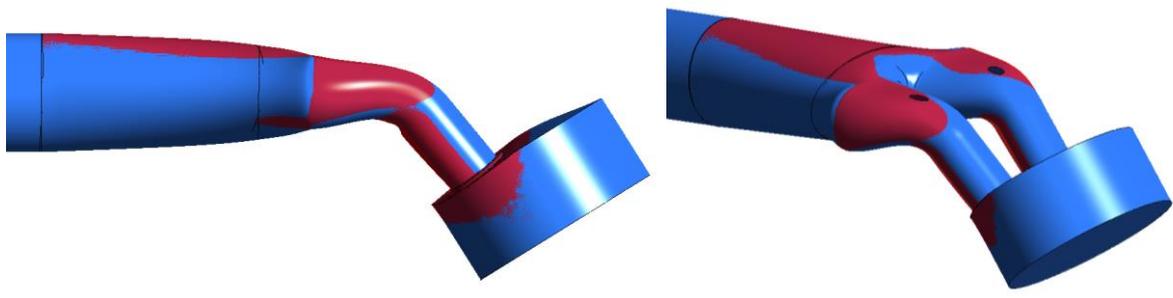


Figure 76: Comparison of baseline(blue) and optimized shape (red)

5.1.3. Conclusions

In this study, a workflow leveraging the adjoint method was proposed to explore new geometries. The adjoint approach is a highly interesting tool that uses the derivative of the monitored quantity to estimate its evolution. This sensitivity can be exploited to modify the geometry, considering that once the adjoint problem is solved, it becomes possible to monitor n shape parameters without additional computational costs. The proposed workflow takes advantage of this by considering a number of shape parameters equal to the number of nodes in the mesh. Each node is then displaced by an amount proportional to the gradient of the observable (filtered to ensure

smoother surfaces). This approach is particularly intriguing because it allows for the exploration of new solutions and geometries, while the method itself is parameter-less and automatic.

However, one limitation is that geometric constraints are often challenging to manage with this approach. To address this, a workflow was developed in this study to handle even complex geometric constraints effectively. The results are very promising. Despite the constraints, an improvement of over 10% in the monitored quantity (pressure-drop) was achieved. The constraint was perfectly respected, and computationally, the mesh control and correction did not lead to a significant increase in computational effort.

5.2. Adjoint real time optimization

The goal of this work is to create a real-time adjoint-based design procedure that enables the definition of any shape parameter and provides a prediction of the variation in the monitored observable. This application is particularly intended for the automotive sector, specifically for making "style modifications"—changes aimed at improving aesthetics, which inevitably also impact aerodynamics. The idea is to develop a tool that, for any proposed shape variation, estimates the impact on the observable, ensuring that aesthetic changes do not excessively degrade aerodynamic performance.

The adjoint method is a powerful tool that provides insights into the behaviour of complex systems. Sensitivity data can be leveraged to estimate how any shape variation affects the selected observable. The core concept is to apply a known displacement, multiply the nodal displacement by the sensitivity, and sum all contributions to estimate the variation in the observable due to the proposed shape change. A key prerequisite for this method is maintaining a consistent mesh structure, making mesh morphing essential in defining shape parameters.

Compared to the application for the duct (discussed previously), this approach uses the adjoint method more deliberately. For the duct, the goal was parameter-less optimization, employing adjoint sensitivity to guide surface sculpting. Here, on the other hand, parameters are set by the user, and the adjoint provides an estimate of their influence. Since this estimation relies only on simple multiplications and summations, it operates in real-time, making the approach particularly appealing for a real-time design platform. The vision is to create a dashboard where users can define any shape variation and immediately obtain an estimate of its impact on the requested observable.

Compared to the ROM-based methods proposed in this thesis, this approach is fundamentally different and, in some respects, complementary. ROMs and machine learning algorithms require large datasets for training and often lack reliability for out-of-sample points or novel parameters. In contrast, the adjoint method, with a single computation, can provide estimates for any shape variation. The main limitation of the adjoint approach is its reduced accuracy for large shape changes, as the prediction is based on a linearization around the baseline point. This work employs RBFs to define shape parameters. As a mesh-less method, RBFs are highly general and do not depend on the baseline mesh.

For instance, considering a simple problem with a single observable and a single shape modification parameter, it is possible to derive the trend of the objective function as the parameter varies. The orange curve in Figure 77 is obtained by varying the amplification of the parameter and modifying the mesh at each step, followed by evaluating the observable through CFD analysis. In contrast, the blue curve is obtained for the same problem using the same amplifications but utilizing the adjoint method. It can be observed that the blue curve is tangent to the objective function at the origin. In fact, the blue curve represents the rate of mesh deformation as the parameter varies.

This approach is particularly efficient when dealing with shape parameters that can be amplified linearly, such as translation or scaling. In these cases, the rate of the mesh used to evaluate the variation of the observable is defined as follows:

$$v_{x_i} = \frac{dx_i}{da} \quad (5.1)$$

This equation is applied for a given node and direction, where a is the amplification of the parameter and x_i is the nodal displacement.

However, when rotations or complex parameterizations that involve hierarchical methods are necessary, this approach can no longer be utilized. In such cases, using finite differences, the mesh velocity around a given amplification a can be calculated as follows:

$$v_{x_i} = \frac{x_i(a+\delta a) - x_i(a)}{\delta a} \quad (5.2)$$

Where $x_i(a+\delta a)$ and $x_i(a)$ are the results of two distinct calculations. By multiplying each nodal velocity by the sensitivity of each node with respect to the objective function (Ψ), the variation of the observable associated with the imposed parameter amplification can be calculated as follows:

$$\frac{\delta \Psi}{\delta a} = \frac{\delta \Psi}{\delta x} \cdot \frac{\delta x}{\delta a} \quad (5.3)$$

In other words, the gradient of the observable with respect to the parametric variation is evaluated. This information is generally used to guide an optimization process, and a local minimum can be identified using a gradient-based optimization method.

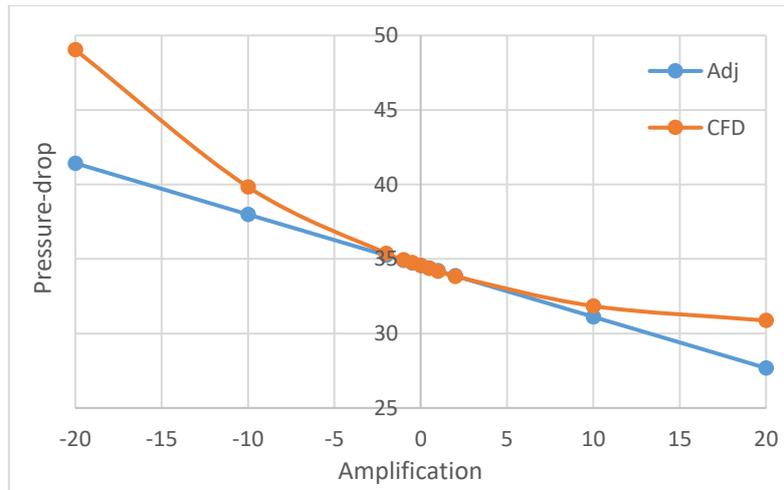


Figure 77: The orange line shows the trend of the observable as the parameter amplification varies, evaluated using CFD. The blue line shows the trend of the observable evaluated with adjoint for the same problem

Specifically, a CAD-based approach was used to define the parameters, and an RBF-based workflow was developed to transfer information from the CAD model to the mesh.

The AeroSUV model (97) , an open-source SUV design, was chosen as the test case (Figure 78). The workflow followed for this study is outlined below:

- A wrapper geometry—a simplified shape enclosing the AeroSUV model—was defined.
- A surface mesh was created on the wrapper for support. This mesh was not used for analysis but served as a set of source points linked to the CAD entities.
- Source points were employed to define displacements.
- RBF interpolation was applied to transfer the known displacements to the mesh nodes.
- The adjoint method was used to preview the variation in the observable (drag, in this case).



Figure 78: AeroSUV baseline

5.2.1. CFD Model

Figures 79 and 80 illustrate the mesh utilized in the study, which comprises a mix of tetrahedral and hexahedral elements. Specifically, the mesh consists of 7453979 cells, 23285526 faces, and 8377264 nodes. The k- ω SST turbulence model was employed for the analysis.

Boundary conditions include a velocity of 25 m/s at the inlet and atmospheric pressure at the outlet. y^+ is less than 30 and y^+ mean is 8.7.

The optimization objective is to minimize the drag coefficient, ensuring that aerodynamic performance improves while maintaining compatibility with the proposed shape variations.

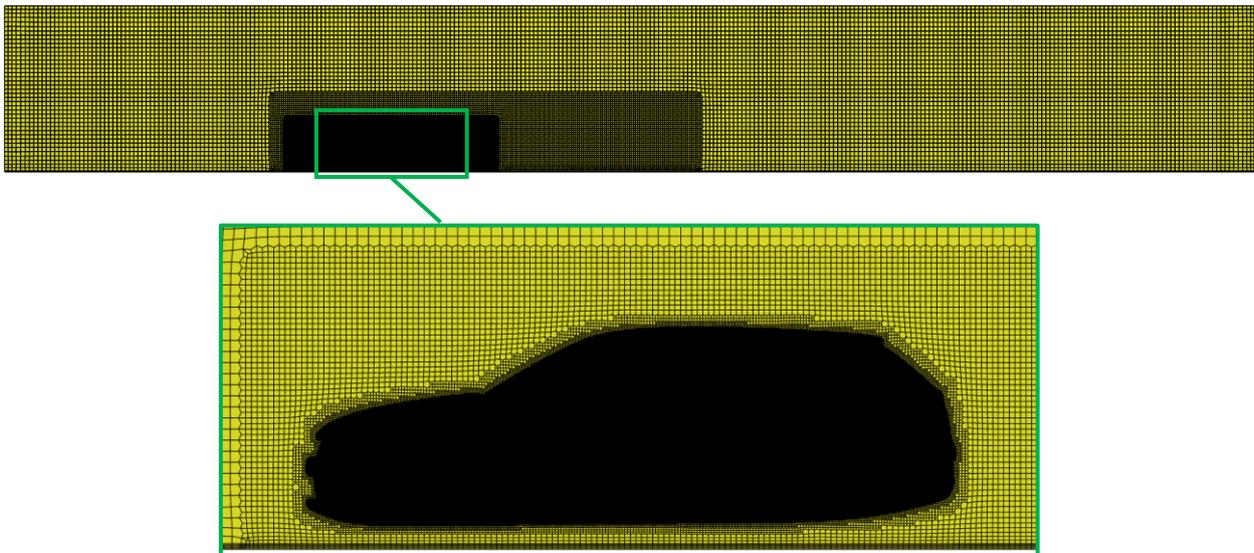


Figure 79: Volume mesh

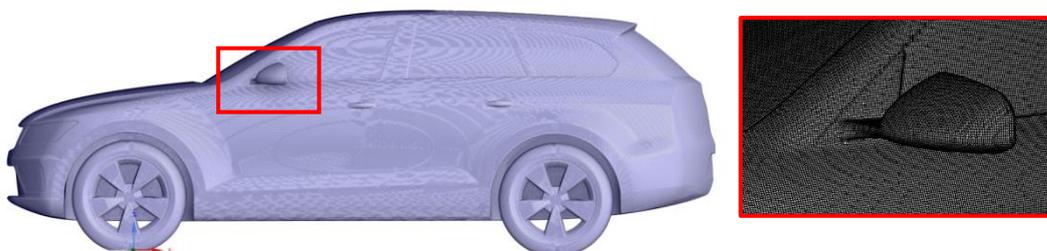


Figure 80: Surface mesh

5.2.2. Shape Parameters

As mentioned in the previous section, an hybrid procedure was used to define the shape parameters. The entire workflow is described in section 2.3.1.3. Figure 81 illustrates the steps involved in controlling the displacement

of an edge. Starting from the wrapper, the support mesh is used to define displacements in the x and y directions of the edges ("edge1", "edge2", and "edge3" in Figure 81). This results in the definition of 6 parameters. The displacement can occur in both directions (positive and negative). Figure 82 summarizes the defined parameters.

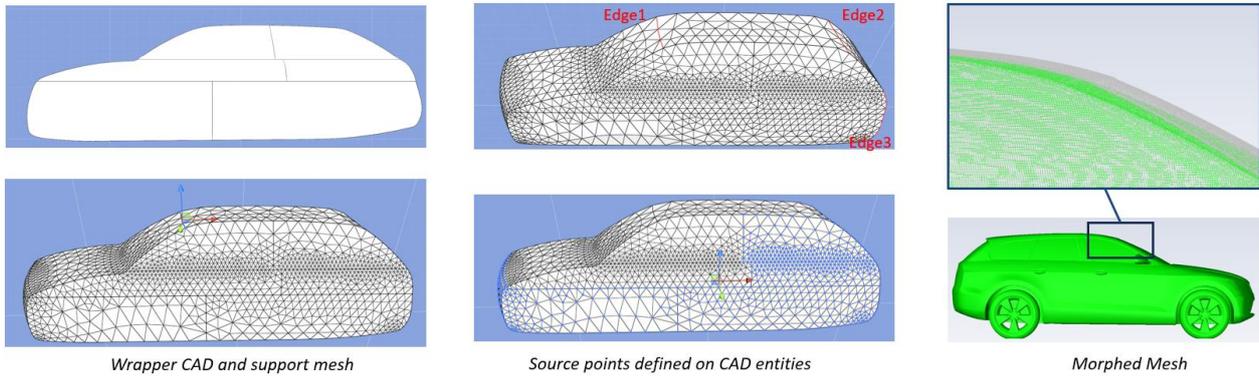


Figure 81: Steps of the described workflow

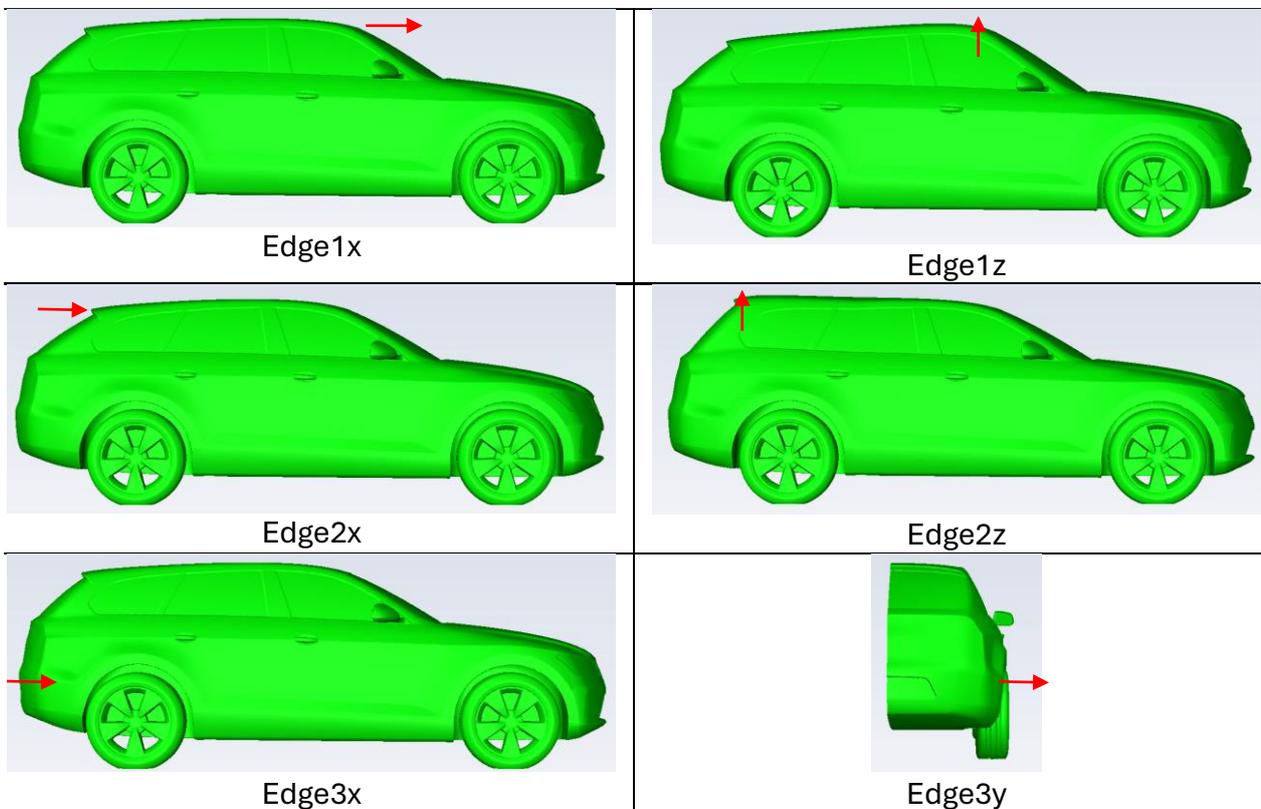


Figure 82: Six shape parameters defined

5.2.3. Results

Figure 83 shows the sensitivity for drag in the considered case. The sensitivity provides valuable information both during the definition of parameters and for choosing the amplification of those parameters. For example, observing Figure 83, we can identify the areas where the sensitivity is greatest, meaning where deforming the mesh would be most beneficial. If the sensitivity is higher, it indicates that for the same displacement, a larger change in the observable can be achieved. In other words, it implies that the considered modification is more efficient in improving performance.

The sensitivity also provides crucial information about the direction in which to act. As it is a vector quantity, it has both a normal direction to the surface and a positive or negative direction. In this work, the parameters are predefined and are "stylistic" parameters, meaning they are not designed for drag improvement but rather for

modifying the vehicle's aesthetics. The adjoint is thus used to understand in which direction the parameters should be amplified in order to avoid worsening performance and, if possible, improve it.

In the table 19, the sensitivity value for each defined parameter is reported. These values are simply obtained by multiplying the nodal displacement by the adjoint sensitivity and summing all the contributions. Based on the sensitivity values and the vehicle's aesthetic, amplifications were proposed, resulting in a modified geometry. A further CFD analysis was conducted to evaluate the Cd for the new configuration. The result is an improvement of approximately 3%. Figure 84 presents a comparison between the baseline and the modified geometry.

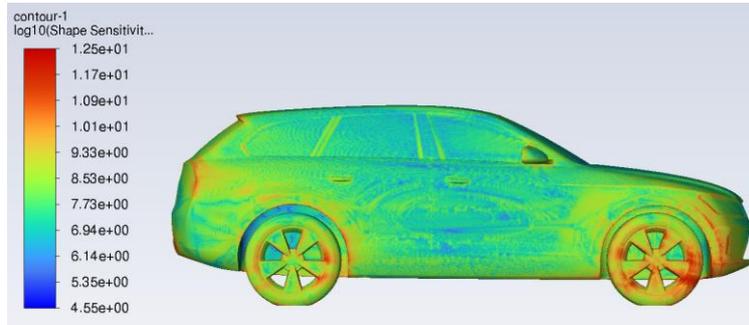


Figure 83: Adjoint shape sensitivity

Table 19: Range, Sensitivity and final value for each shape parameter

	Edge 1x	Edge 1z	Edge 2x	Edge 2z	Edge 3x	Edge 3z
Range	$-1 \div 1$					
Sensitivity	0.04	-0.33	0.54	-0.9	0.2	0.1
Final Value	-0.12	1	-1	1	-0.6	-0.3

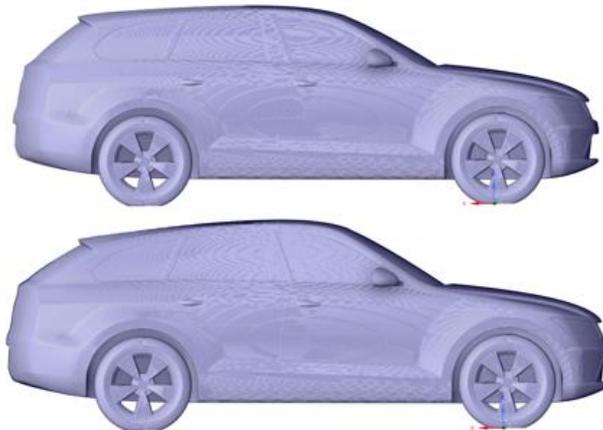


Figure 84: Comparison of baseline (upper) and optimized shape

Table 20: Cd comparison

	Cd
Baseline	0.3
Optimized	0.29 (-3.33%)

5.2.4. Conclusions

In this study, a workflow is proposed that starts from a generic CAD wrapper to define shape parameters that directly influence the mesh. The idea behind this approach is that the parametrization is generic and could theoretically be applied to any SUV. Additionally, the adjoint is used to provide real-time predictions of shape

variation. Thus, the adjoint drives the optimization process, guiding the direction of change for each parameter. So far, these operations have been carried out within the CFD solver, but the goal is to create an external dashboard that performs the same functions as those encapsulated in the workflow (wrapper, parametrization, adjoint sensitivity, and observable variation estimation). Once the CFD and adjoint solution is calculated, the CFD solver becomes redundant, and the operations described no longer require additional computations.

The results obtained are very promising. It should be emphasized that, in this case, the shape parameters were not defined with the goal of improving drag, but rather to modify the aesthetics without negatively affecting the drag. From this perspective, the 3% improvement should be contextualized. By defining more effective shape parameters, it would certainly be possible to further enhance the aerodynamic performance.

5.3. FSI – Adjoint coupling

This work presents a study on using CFD adjoint calculations combined with FEA structural mechanics to predict how aerodynamics are influenced by structural flexibility. The proposed approach was first tested on a simple cantilever beam to clarify its fundamental principles and then applied to two industrial cases: the HIGH LIFT CRM benchmark wing in the aeronautical sector and the front section of a 2022-regulation F1 car in the motorsport sector.

The adjoint method calculates sensitivities, which are then multiplied by the nodal displacements associated with structural deflection to determine sensitivity to deformation. This assumes a linear system response near the operating point.

The results show that for the cases studied, adjoint evaluation achieves sufficient accuracy, with a single CFD simulation of the primal flow and adjoint enabling estimation of how aerodynamics are impacted by structural elasticity.

The FSI-Adjoint coupling has been studied in various papers (98) (99) (100) (101) (102), but in this work, a different approach has been proposed. The idea of this work is to leverage adjoint calculations to predict the variation in drag and lift associated with the elastic behaviour of the structure. This prediction is computed using sensitivity data.

The idea behind this work is similar to that presented in *aeroSUV*, but the displacement of the mesh is not guided by optimization but by the elastic behaviour of the structure. Thus, the nodal displacement is the FEM deformation, and the observables are the forces F_x and F_z . Given the sensitivities δF_x and δF_z , it is possible to estimate the variation in drag and downforce/lift associated with structural deformation. It's worth emphasize that the update of aerodynamic loads due to changes in structural deformation is not performed by deforming the mesh and running a new CFD analysis, but by using the adjoint calculated on the baseline, multiplying the known sensitivity by the updated displacement field. Therefore, the cost in terms of CFD analysis is reduced to a single analysis. The method has been compared with a classical two-way approach.

5.3.1. Testcase 1: cantilever beam

As the first test case, a relatively simple scenario was considered: a thin cantilever beam with one end fixed. The dimensions of the beam are $L = 1.6$ cm, $t = 0.1$ cm, $L_c = 20$ cm, and $h_c = 3.2$ cm.

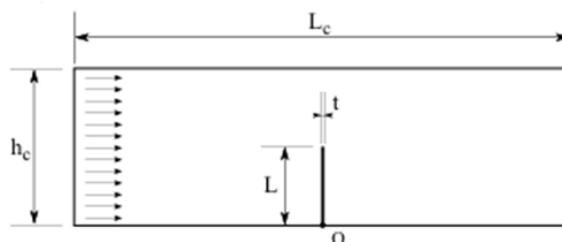


Figure 85: 1st testcase

The following 3D model was constructed (Figure 86), considering a fluid volume thickness of 20 cm. Boundary conditions were set as follows: an inlet velocity of 1 m/s, an outlet pressure equal to atmospheric pressure (101325 Pa), symmetry conditions on the lateral faces of the fluid domain, and a no-slip wall condition on the thin beam surfaces and the ground. A RANS analysis was performed using the $k-\omega$ SST turbulence model, with second-order discretization applied.

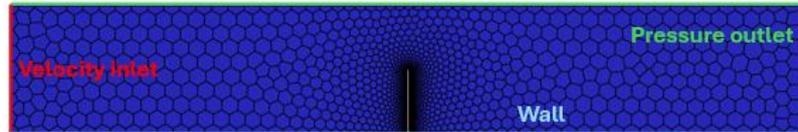


Figure 86: CFD Volume mesh

The FEM model is shown in Figure 87. A fictitious Young's modulus of 40000 Pa and a Poisson's ratio of 0.33 were assigned to ensure a significant FEM deformation. Large displacements were considered, meaning the model is non-linear. Aerodynamic forces were mapped onto the lateral surfaces.



Figure 87: FEM Model

5.3.1.1. Results

A complete two-way analysis was performed using the described models. Table 21 shows the evolution of the monitored quantities, where it can be observed that the solution converges after the first step. Figure 88 displays the displacement contour for the beam at convergence. Mesh morphing was employed to deform the CFD mesh and update the aerodynamic loads accordingly.

Table 21: Two-way methods drag values

Drag [N]	Step
0.001050443	0
0.000673901	1
0.0006738	2

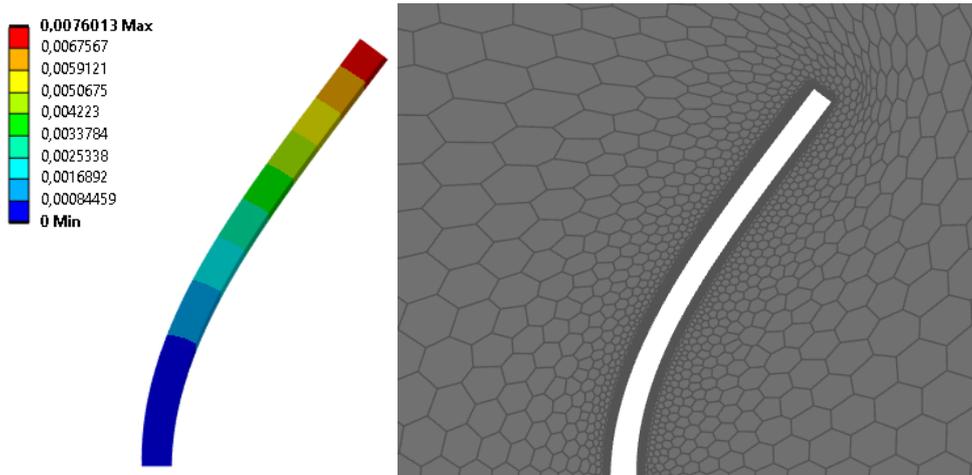


Figure 88: FEM displacements (left), deformed CFD mesh (right)

The adjoint was then calculated to assess the variation in drag associated with the structural deflection. In Figure 89, the adjoint prediction is shown in orange, while the drag value, modulated by amplifying the FEM displacement from 0 to 1, is shown in blue. It can be observed that the error is very small and almost negligible up to a 50% amplification of the FEM displacement, but it becomes more significant for larger deformations.

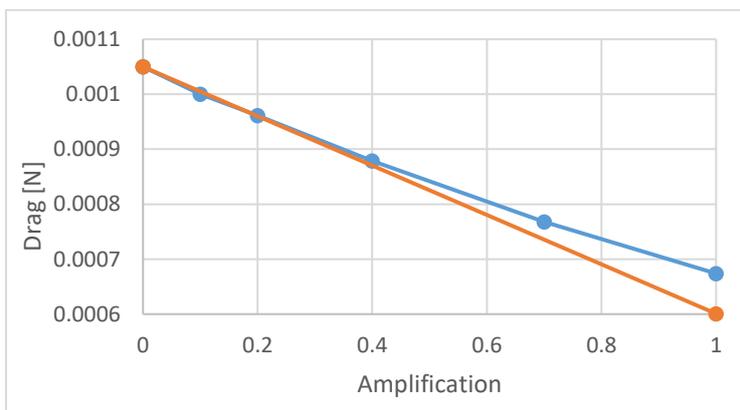


Figure 89: The blue line shows the trend of the observable as the parameter amplification varies, evaluated using CFD. The orange line shows the trend of the observable evaluated with adjoint for the same problem

The diagram in Figure 89 clearly shows that for such large deformations, the adjoint does not provide a sufficiently accurate response. However, the range in which the adjoint's response is meaningful remains considerable. In fact, with a deformation approximately half of the one considered, that is, about 20% of the beam's length, the adjoint prediction is very reliable. For this reason, an additional test was conducted by modifying the elastic modulus of the beam to a value of 90000 Pa. With the new elastic modulus, the previously described analysis was repeated, comparing the full two-way method with the adjoint. In this case, the displacement at the tip of the beam is approximately 3.5 mm (Figure 90), or about 20% of the beam's length. Therefore, we remain in the realm of large displacements, and the deflection remains very interesting. In this case, the adjoint prediction is highly accurate, and the error is reported in Table 22.

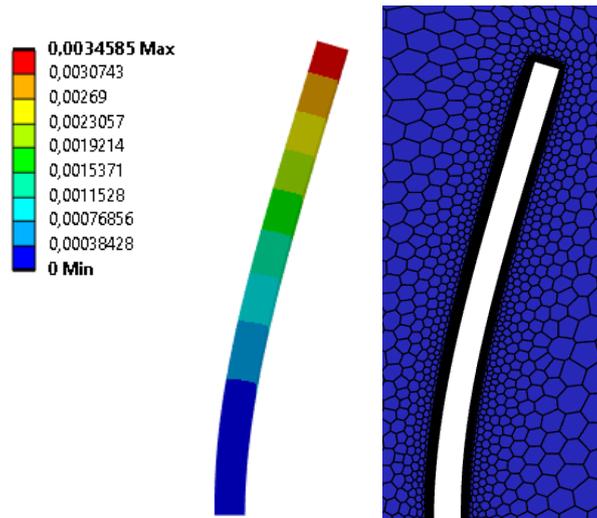


Figure 90: FEM displacements (left), deformed CFD mesh (right)

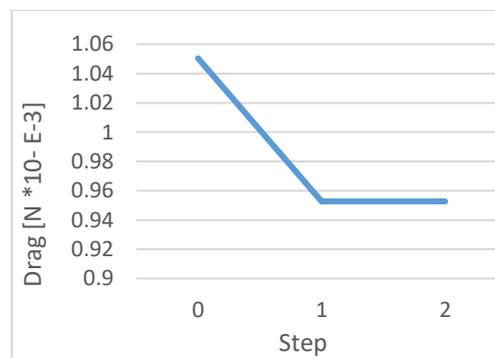


Figure 91: Drag convergence history for Two-way method

Table 22: Comparison of adjoint and Two-way method for drag evaluation

ΔD Two-way	ΔD Adjoint	Err abs	Err rel
$-9.52E-05$	$-9.77E-05$	$1.93E-04$	$2.68E-02$

5.3.2. Testcase 2: CRM Wing

The second testcase considered an aerodynamic scenario. In particular, a benchmark from the literature was selected (103), for which CAD models, FEM models, and wind tunnel results are available (104). Compared to the analyses in the literature, a coarser mesh was used, and only the wing of the FEM model was selected. The goal is to achieve a realistic FEM deformation (14 mm based on literature data) and to obtain a good comparison between the full two-way method and the adjoint prediction in terms of drag and lift.

In Figure 92, the CFD model used is shown, with a focus on the mesh around the aircraft. The following boundary conditions were considered:

- Pressure far-field with a pressure of 125,481 Pa, $Ma = 0.85$, $AoA = 2.5^\circ$, Temperature = 310.93 K
- Symmetry in the plane of symmetry
- Wall no-slip on the surface of the aircraft
- The reference area is 0.13987 m^2 (1/2 wing reference area), while the reference length is 0.18923 m (mean aero chord).
- K- ω SST turbulence model

y^+ is less than 200 and y^+ mean is 43.2.

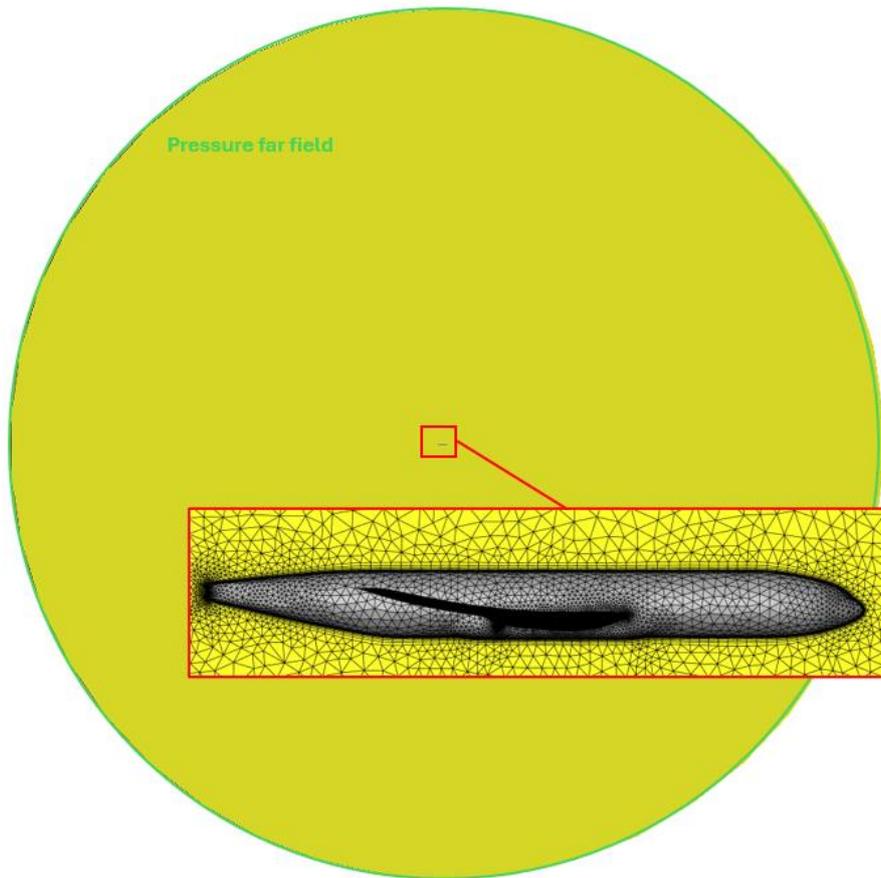


Figure 92: CFD Volume mesh

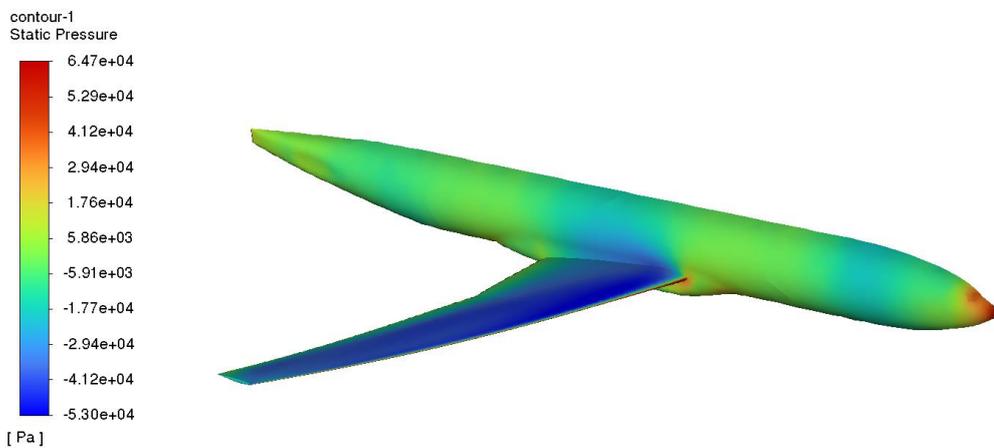


Figure 93: CFD static pressure contour plot

The FEM model was taken from the model available in the literature. Compared to the complete model (104) (Figure 94), only the wing was used (Figure 95), which was fixed at one end, while the CFD loads were applied to the top and bottom surfaces of the wing. The areas that were excluded are few and not significant for evaluating the deformations. The forces, on the other hand, are evaluated using the complete CFD model. From this

perspective, using RBF for mapping in both directions is very advantageous. In fact, the method, being mesh-less, does not require a 1-to-1 correspondence between the FEM and CFD models.



Figure 94: Entire FEM Model from literature (104)

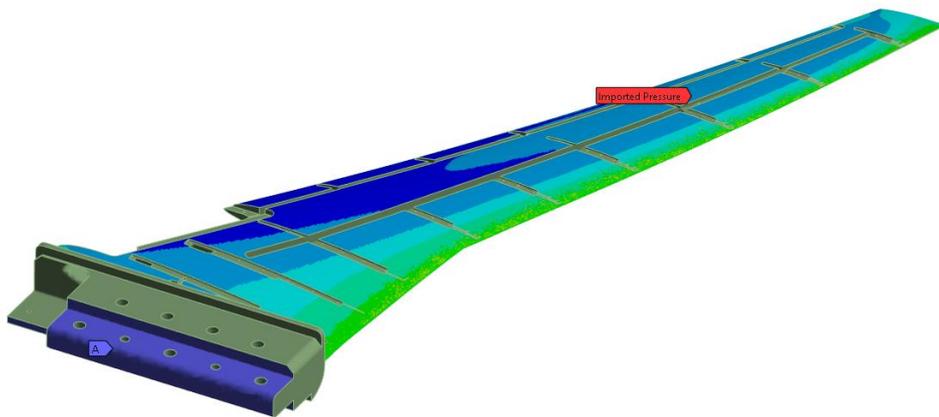


Figure 95: FEM model with aerodynamic loads

5.3.2.1. Results

As in the previous case, the two-way method was compared with the adjoint prediction on the variation of drag and lift. Using the complete two-way method, a converged tip deflection of approximately 13 mm was obtained, which is consistent with the literature value of 14 mm (104). The difference is attributable to the simplifications made. However, in evaluating the effectiveness of the proposed method, a comparable structural deflection is required. Therefore, the difference of 1 mm is considered acceptable. The RBF problem has 106000 source points and it is solved in 308 s using 48 cores CPU.

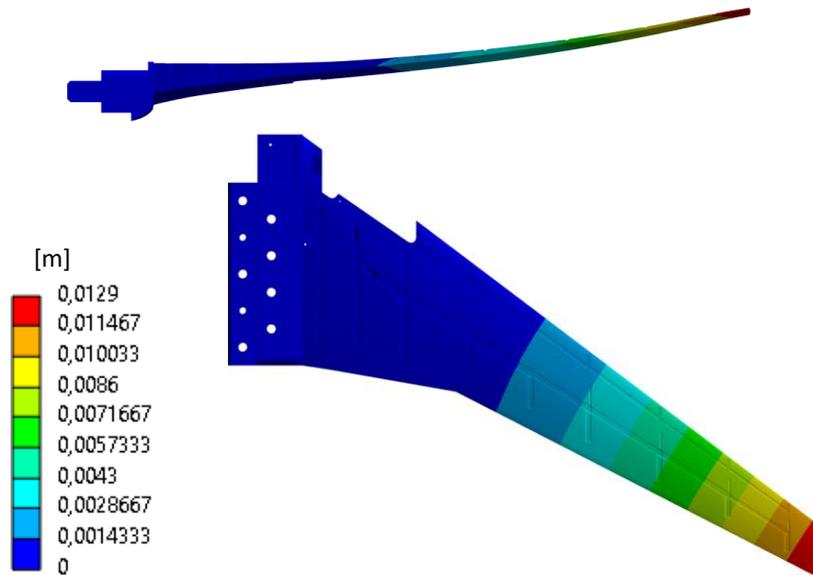


Figure 96: FEM displacements

Table 23: Cd and Cl convergence typo

Step	Cd	Cl
0	0.052	0.47
1	0.049	0.44
2	0.049	0.439

Finally, the two-way method was compared with the adjoint prediction. Table 24 provides a comparison between the evaluation using the two-way method and the approach proposed in this work, based on the adjoint. It can be observed that the error is very low, and the adjoint prediction proves to be highly accurate. Additionally, as with the previous case, the FEM displacement was modulated from 0 to 1 to verify the tangency of the adjoint solution with respect to the CFD solution (Figure 97).

Table 24: Comparison of two-way and adjoint

ΔC_d Two-way	ΔC_d Adjoint	ΔC_l Two-way	ΔC_l Adjoint
-0.003	-0.003	-0.031	-0.032

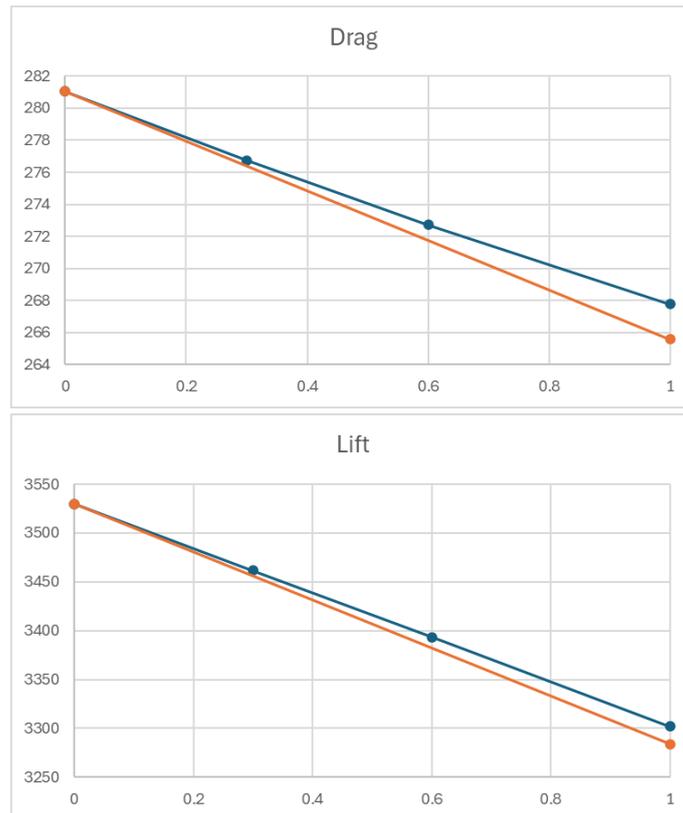


Figure 97: The blue line shows the trend of the observable as the parameter amplification varies, evaluated using CFD. The orange line shows the trend of the observable evaluated with adjoint for the same problem

5.3.3. Testcase 3: F1 front wing

The objective of this work is to explore the possibility of using adjoint calculations to obtain a sufficiently accurate estimate of how the aerodynamics is affected by structural deflection. For this testcase the material specifications, thicknesses, and stacking sequences of the composite layers are highly confidential information. However, strict adherence to real models is not necessary. In this phase, it is sufficient to have a plausible estimate of the structural deformation. For this reason, a generic carbon fibre composite with a stacking sequence and number of layers was initially considered, such that applying a static vertical load of 1000 N results in a maximum deflection of less than 20 mm (as per regulations). Once the FEM model was established, a CFD model was created, and the variation in drag and downforce was evaluated using the adjoint method. In particular, the adjoint sensitivity was multiplied by the nodal displacement to estimate how the deflection influences the selected observables. This estimate was compared with the value calculated two-way approach.

5.3.3.1. Geometry

While simulating the entire car would provide a more accurate representation, doing so would demand significant computational resources. However, since the frontal elements, particularly the front wing, have a considerable influence on the race car's downforce and drag characteristics, valuable insights into the aerodynamic performance of front wings can still be obtained from a simplified geometry simulation. The central focus of this study is the simulation of front wings, emphasizing how the aerodynamic forces are distributed across them.

In general, certain regulations are followed in the design of front wings. The front wings must have a multi-element configuration, with a maximum of four elements and must be angled backward from the nose. This reduces the pressure difference on a given surface, allowing the cars to achieve higher speeds before encountering compressibility issues. The endplates must be integrated with the elements of the front wing using a curved profile. This reduces the efficiency losses that occurred in previous designs. For other specifications regarding maximum dimensions, refer to the complete documentation (105).

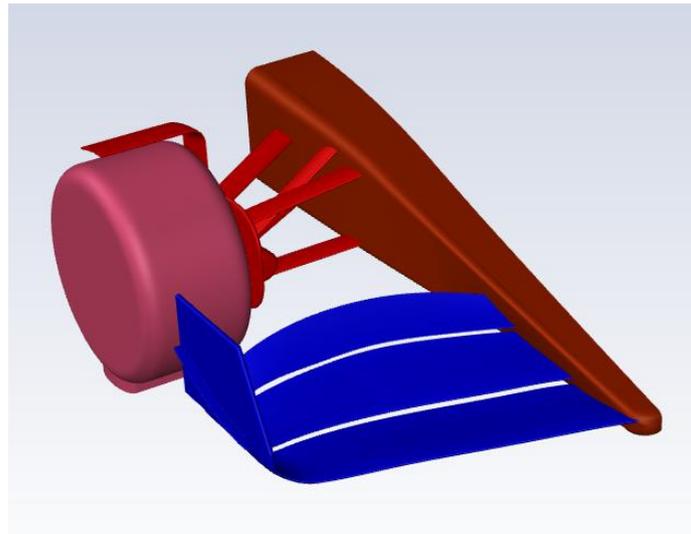


Figure 98: Geometry

5.3.3.2. CFD Model

For the CFD model, various works in the literature were referenced to define the computational domain, obtain plausible values for the drag coefficient (C_x) and downforce coefficient (C_z), and establish the boundary conditions. This approach ensured that the model was grounded in existing research and that the results were realistic and comparable to those observed in similar studies (106) (107) (108). The computational domain's length is set at 10 times that of the front wing, with the distance from the inlet to the reference axis being 4 times the front wing's length. The outlet is located 4 times the width of the front wing from the exit. Given the vehicle's perfect symmetry, only half of the car is simulated, and the distance between the longitudinal symmetry plane and the side wall is approximately 3 times half the width of the car model.

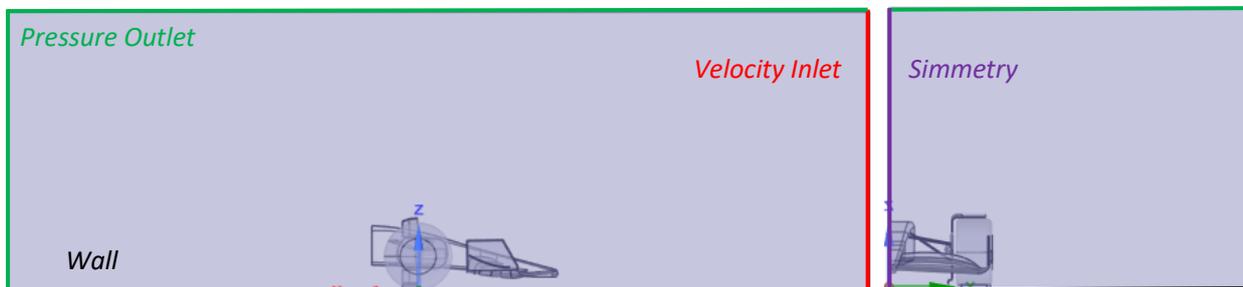


Figure 99: CFD domain and boundary conditions

The numerical problem's boundary conditions must be defined to compute the specific solutions of the partial differential equations using CFD software. These boundary conditions, which are described below, correspond to those applied in ANSYS Fluent:

- *Velocity-inlet*: A uniform horizontal velocity profile is prescribed at the domain's inlet, where the inlet velocity is set to 50 m/s, consistent with values used in validation studies from the literature. The working fluid in this case is air, characterized by a density of 1.225 kg/m^3 and viscosity of $1.81\text{E-}05 \text{ kg/m-s}$. Consequently, the Reynolds number in the simulation is $Re = 1.2\text{E}07$. Turbulent intensity at the inlet is fixed at 0.15%, following the same references. The reference area is set to 1 and the reference length is the wing chord.
- *Pressure-outlet*: At the outlet, atmospheric pressure is applied, ensuring the correct outflow conditions for the domain. This boundary condition is applied to all domain boundaries except the inlet and the symmetry plane.
- *Wall*: This boundary condition is applied to all solid surfaces in the model, including the front wing, tires, and ground. The wall boundaries on certain components have both translational and rotational movement. The ground surface is given a translational speed of 50 m/s, matching the inlet velocity.

Additionally, the tires are modelled with rotational motion, where the angular velocity is calculated as $\omega=V/R_w$, with R_w representing the rolling radius of the front wheels. This ensures that the contact point between the tires and the ground has matching velocities.

- *Symmetry*: This condition ensures zero flux across the centreline of the computational domain, taking advantage of the car's symmetry. The symmetry boundary condition is thus enforced along the centreplane of the domain.
- In line with the studies found in the literature, a RANS $k-\omega$ SST turbulence model was used. This method is well-established for simulating turbulent flows and provides a good balance between accuracy and computational efficiency, making it suitable for aerodynamic analyses of the given application.

Drag and lift are evaluated only on the wing surfaces. The analyses were carried out in ANSYS Fluent, and the forces were calculated as the integral of pressure and viscous stresses on the selected surfaces.

The analyses were carried out in ANSYS Fluent, and the forces were calculated as the integral of pressure and viscous stresses on the selected surfaces.

5.3.3.3. CFD Mesh

To construct the CFD mesh, we started from literature considerations indicating that a y^+ value between 40 and 150 is sufficient to obtain a reasonably accurate estimate of drag and downforce (106). To validate this consideration for the studied geometry, a mesh convergence analysis was performed. Specifically, three meshes with increasing levels of accuracy were evaluated, and for each mesh, C_x and C_z were assessed as performance indicators. The values were compared with those found in the literature.

It is important to note that, due to the absence of a CAD model similar to those in the literature, the comparison can only be qualitative. The objective of this phase was to develop a CFD model that does not depend on mesh density while obtaining reasonable C_x and C_z values for the problem at hand.

Below are the main parameters of the meshes considered, all of which are polyhedral mosaic meshes created in Fluent. In line with the considerations found in the literature, we chose to proceed with the coarser mesh (with y^+ values between 40 and 150). Additionally, some details of the chosen mesh for subsequent analyses are provided.

Table 25: Mesh statistics

Nodes	Cells	Elements	Y^+ Range	Y^+ mean	$SC_x [m^2]$	$SC_z [m^2]$
<i>Mesh 1</i>						
971114	4613672	2863894	40 ÷ 150	76.69	0.16	- 0.8
<i>Mesh 2</i>						
8558417	40872764	25522897	1 ÷ 15	7.53	0.15	- 0.8
<i>Mesh 3</i>						
11993766	56730109	35093950	0.2 ÷ 5	1.5	0.152	-0.82

Table 26: Inflation layers statistics

Inflation Layers	1st layer thickness [mm]	Growth Rate
<i>Mesh 1</i>		
13	1	1.2
<i>Mesh 2</i>		
15	0.1	1.2
<i>Mesh 3</i>		
15	0.02	1.2

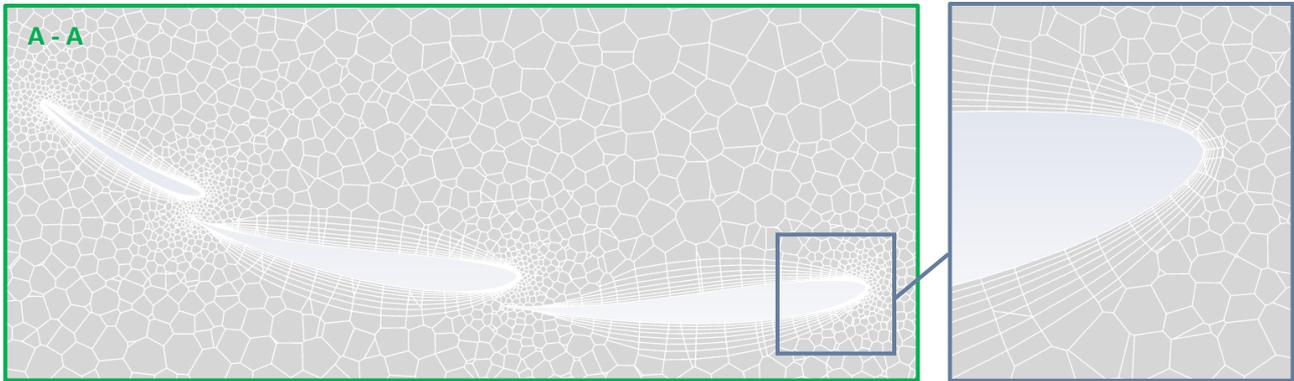
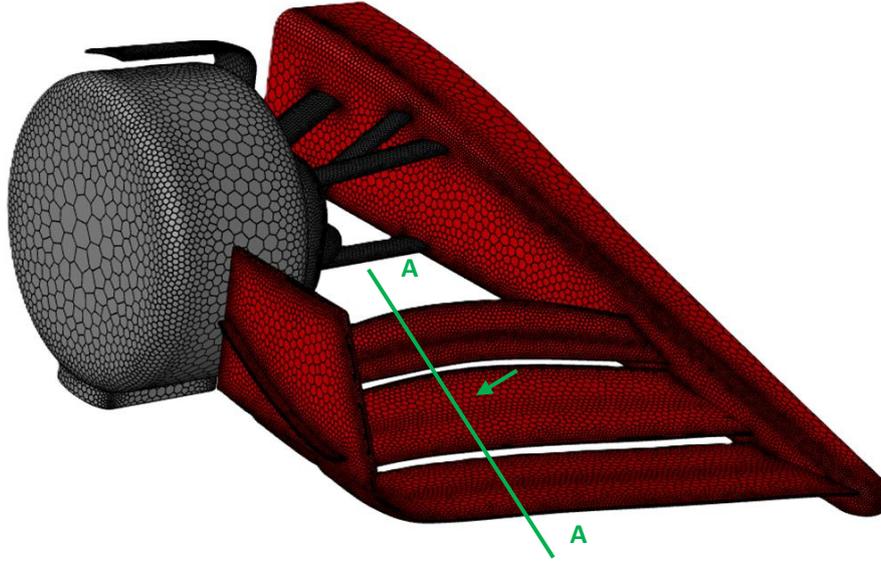


Figure 100: CFD Mesh

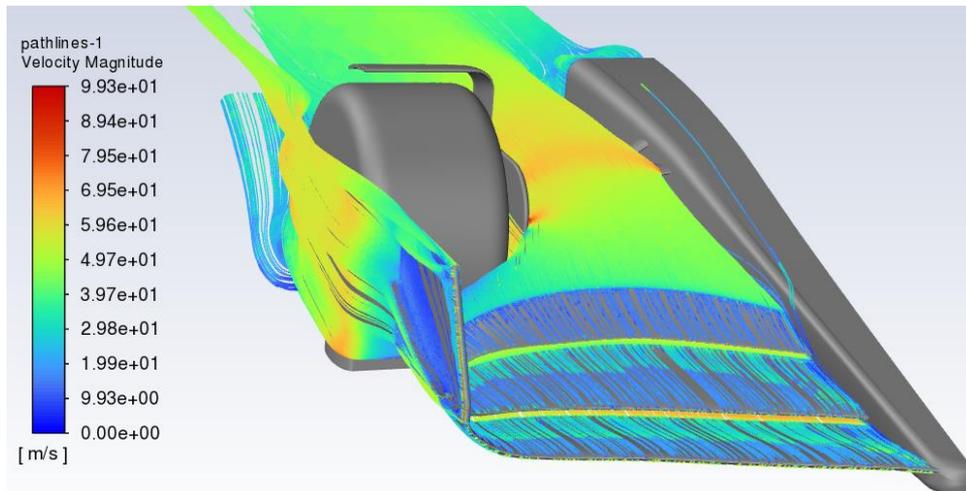


Figure 101: Velocity path lines

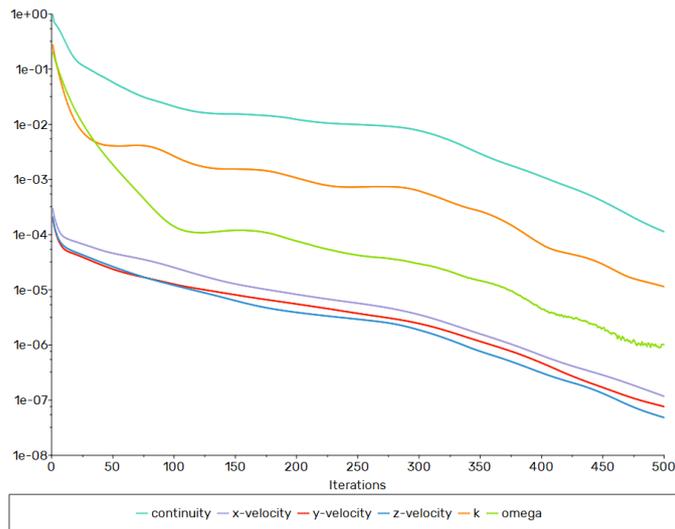


Figure 102: Residuals convergence plot

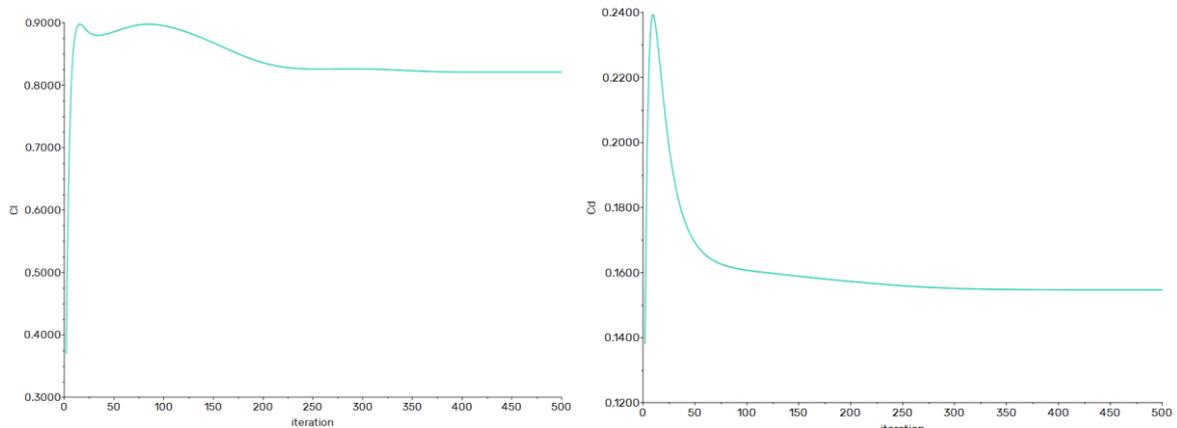


Figure 103: Cx and Cl convergence plots

5.3.3.4. FEM Model

As mentioned in the previous section, many of the necessary details for characterizing the FEM model are not available. However, it is known that regulations dictate that the maximum deflection of the wing under a static load of 1000 N must not exceed 20 mm. Based on these considerations, a generic carbon fibre composite was reconstructed (table 27) with a stacking sequence and number of layers designed to achieve the required deformation. The model was created using Ansys ACP. The final model has a thickness of 5 mm and consists of 4 layers with the stacking sequence (0, 45, -45, 0), each with a thickness of 1.25 mm. The direction of the fibres was set using the edges of the geometry. The layers are made of carbon fibre prepreg, an orthotropic material, whose properties are reported in Table 27. Below are the main structural characteristics of the material. It is important to emphasize that in this study, having a highly accurate FEM model is not critical; the main goal is to obtain a plausible structural deflection to allow for a reasonable comparison between the conventional FSI approach and the estimated predictions using the adjoint method.

For boundary conditions, the connection point of the nose to the rest of the vehicle was fixed, and the plane of symmetry was defined accordingly. Initially, a static load of 1000 N was applied to dimension the composite layers. In a second analysis, the pressures calculated from the CFD analysis were mapped onto the FEM model.

Below, the FEM model used and the results obtained from the static calibration test of the model are presented. In particular, the deformations are reported, as they serve as the reference parameter for sizing the layers.



Figure 104: FEM Model

Table 27: Material properties

Young's Modulus X direction	2.09E+11	Pa
Young's Modulus Y direction	9.45E+09	Pa
Young's Modulus Z direction	9.45E+09	Pa
Poisson's Ratio XY	0.27	
Poisson's Ratio YZ	0.4	
Poisson's Ratio XZ	0.27	
Shear Modulus XY	5.5E+09	Pa
Shear Modulus YZ	3.9E+09	Pa
Shear Modulus XZ	5.5E+09	Pa
Tensile X direction	1.979E+09	Pa
Tensile Y direction	2.6E +07	Pa
Tensile Z direction	2.6E +07	Pa
Compressive X direction	-8.93E+08	Pa
Compressive Y direction	-1.39E+08	Pa
Compressive Z direction	-1.39E+08	Pa
Shear XY	1E+08	Pa
Shear YZ	5E +07	Pa
Shear XZ	1E+08	Pa

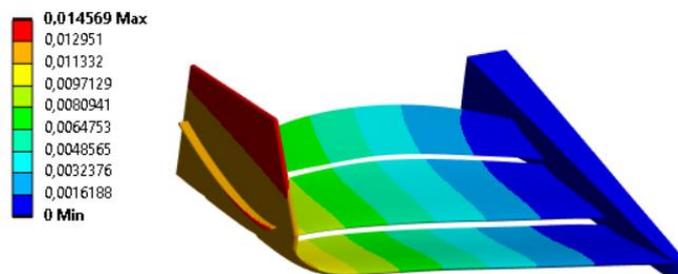


Figure 105: FEM Displacements

5.3.3.5. Results

This section presents the results related to the F1 front wing. Specifically, three scenarios were explored. In the first, the adjoint calculated at a fixed inlet velocity v_0 was used to estimate the impact of flexibility on drag and downforce. In the second, the adjoint at v_0 was employed to predict the aeroelastic response of the same geometry at a different speed. Finally, the adjoint method was applied to assess the influence of the stacking sequence in composite material layers.

Case 1 : two-way vs. adjoint, velocity 50 m/s

To test the method, a comparison was made against a classical approach, the two-way method. Specifically, a speed of 50 m/s was fixed for this analysis. The following Ansys tools were used:

- Ansys ACP for creating the FEM model
- Ansys Mechanical for FEM analyses
- Ansys Fluent for CFD analyses
- Ansys Workbench as the orchestrator for mapping aerodynamic loads onto the FEM mesh
- RBF Morph for deforming the CFD mesh based on the FEM displacements

RBF mesh morphing was employed to deform the CFD mesh. The FEM displacements were read as an RBF field, which was then used to deform the mesh accordingly.

Below are the results obtained from this validation. Notably, it can be observed that the method converges as early as the second step.

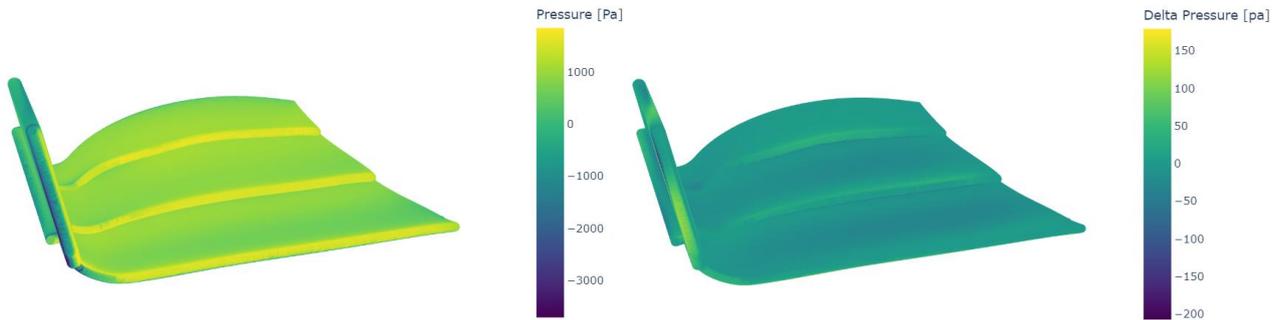


Figure 106: Baseline static pressure and pressure variation in the converged configuration using the two-way method

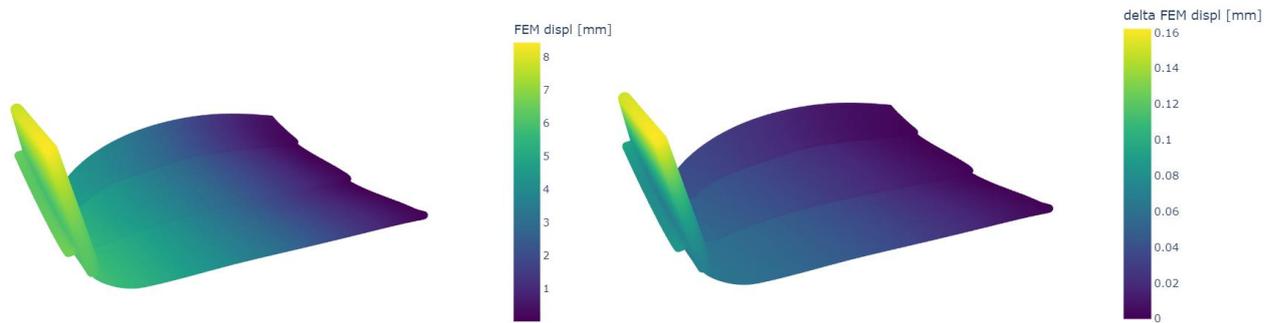


Figure 107: baseline FEM displacements and variation in the converged configuration using the two-way method

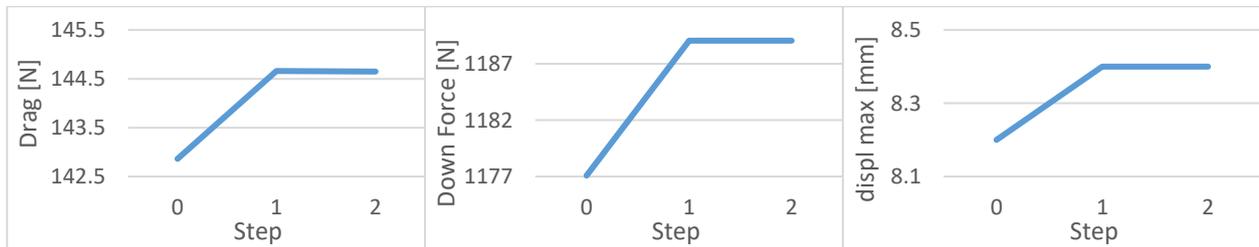


Figure 108: Convergence history for two-way method

Table 27: Convergence history for two-way method

	Drag [N]	Down Force [N]	Max Displ [mm]
Step 0	142.86	1177.08	28.96
Step 1	144.66	1189.05	29.33
Step 2	144.68	1189.11	29.56

The results obtained were compared with the adjoint prediction. It is important to note that to evaluate the variations in drag and downforce, the adjoint calculation must be performed twice, once for each observable. The sensitivities calculated in this way can then be multiplied by the nodal displacements to estimate the changes in drag and downforce.

Finally, these results were compared with those obtained from the two-way analysis to estimate the error.

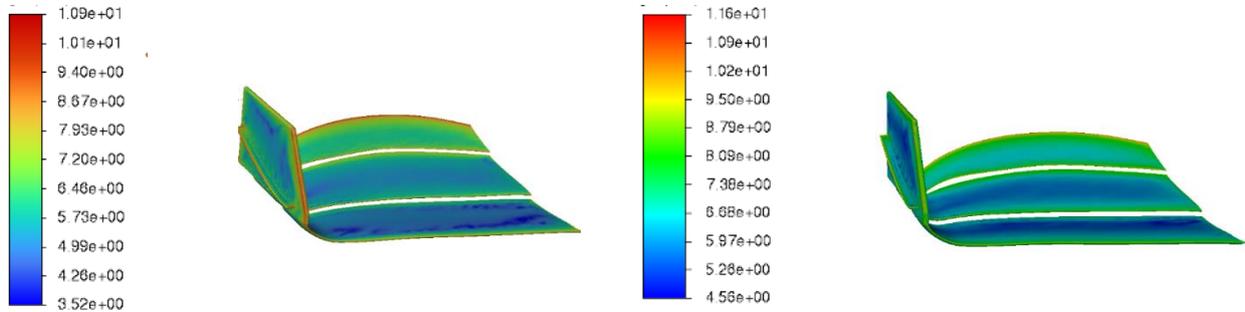


Figure 109: Log10 Shape Sensitivity for drag (left) and downforce (right)

Table 28: Two-way and adjoint comparison

	Drag Variation [N]	Downforce Variation [N]
Two-way	1.79	11.75
Adjoint	1.83	11.87

Case 2 : Speed-up, 65 m/s

In this case, the adjoint method was used to predict the variation in drag and downforce using the adjoint solution evaluated at 50 m/s, while aiming to estimate the aeroelastic behaviour at a speed of 65 m/s. Specifically, it was observed that the non-dimensional parameters C_x and C_z remained nearly constant when transitioning from 50 to 65 m/s. This suggests that there are no significant changes in the pressure distribution with the increase in speed.

Table 29: Comparison of C_x and C_z for different velocities

	C_x	C_z
65 m/s	0.154	0.821
50 m/s	0.154	0.813

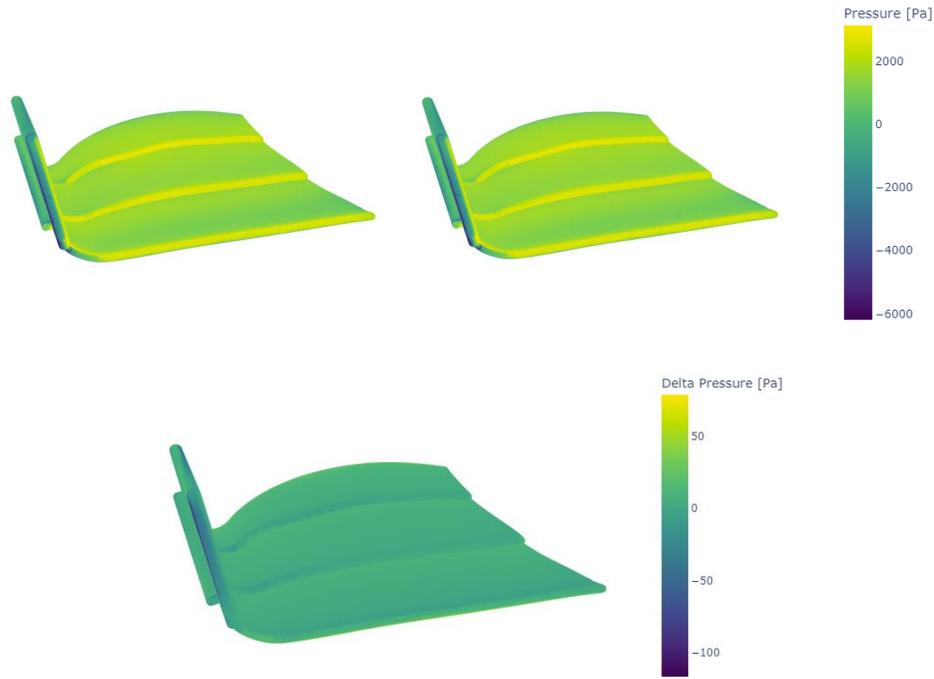


Figure 110: Comparison of static pressure contours 65 m/s solution vs. scaled 50 m/s (above), and error (below)

Therefore, the pressure field can be estimated simply by scaling the solution obtained at 50 m/s by a factor related to the velocity change : $\frac{V_0^2}{V_1^2} = \frac{50^2}{65^2}$. This scaling approach relies on the assumption that the aerodynamic characteristics do not vary significantly over the examined range of velocities. Once the updated pressure field is established, it becomes feasible to evaluate the displacement field through a FEM analysis. This analysis provides the necessary information on how the structure deforms under the influence of aerodynamic loads. Finally adjoint sensitivity have to be scaled by the factor : $\frac{V_0}{V_1} = \frac{50}{65}$ to be used for predict drag and downforce variation. To compare the results from the adjoint sensitivity analysis with those obtained from a traditional two-way FSI approach, table 31 summarizes the key performance indicators.

Table 30: Two-way and adjoint comparison

	Drag Variation [N]	Downforce Variation [N]
Two-way	3.09	20.1
Adjoint	2.99	20.96

Case 3: Variation of Stacking Sequence

As the last case study, the possibility of modifying the material and using the adjoint method to estimate the variation of drag and downforce was evaluated. This possibility is particularly interesting because, in standard FSI optimization problems, it is necessary to iteratively switch between solvers for each DP. This makes the workflow very lengthy and laborious. In contrast, with this approach, it is possible to obtain an estimate of the variation of the quantities of interest for any considered material with just a single CFD calculation and the adjoint method. Since the rigid geometry remains the same, the pressure field from the first step is always the same. Therefore, it is possible to evaluate the displacement field by updating the material properties and performing the FEM analysis. Once the displacement field is known, it is sufficient to multiply the nodal displacement by the adjoint sensitivity to obtain an estimate of the variation of the quantities of interest. Compared to case 1, the ply configuration has changed from $(0^\circ, 45^\circ, -45^\circ, 0^\circ)$ to $(0^\circ, 45^\circ, 0^\circ, 45^\circ)$, resulting in the deformed shape shown in the figure 111. Figure 112 shows a comparison between the traditional workflow and the adjoint-based workflow for optimizing the

stacking sequence of the composite in the case of FSI problems. The proposed approach allows for evaluating how the observable changes with just a single CFD and adjoint analysis by simply updating the FEM displacements for the different configurations studied. This approach would significantly reduce computation times for this type of problem.

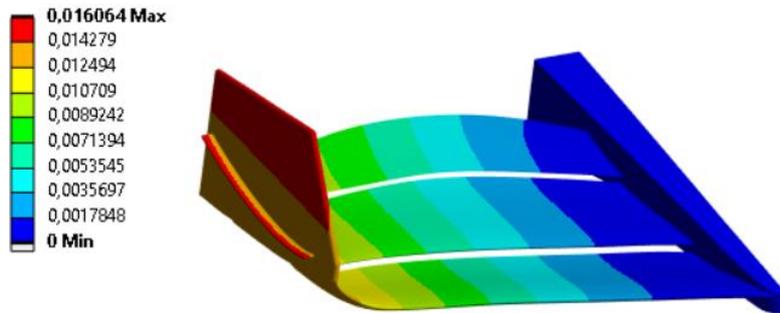


Figure 111: FEM displacements for the new stacking sequence with a static load of 1000 N

Table 31: Two-way and adjoint comparison

	Drag Variation [N]	Downforce Variation [N]
Two-way	1.4	23.4
Adjoint	1.3	22.7

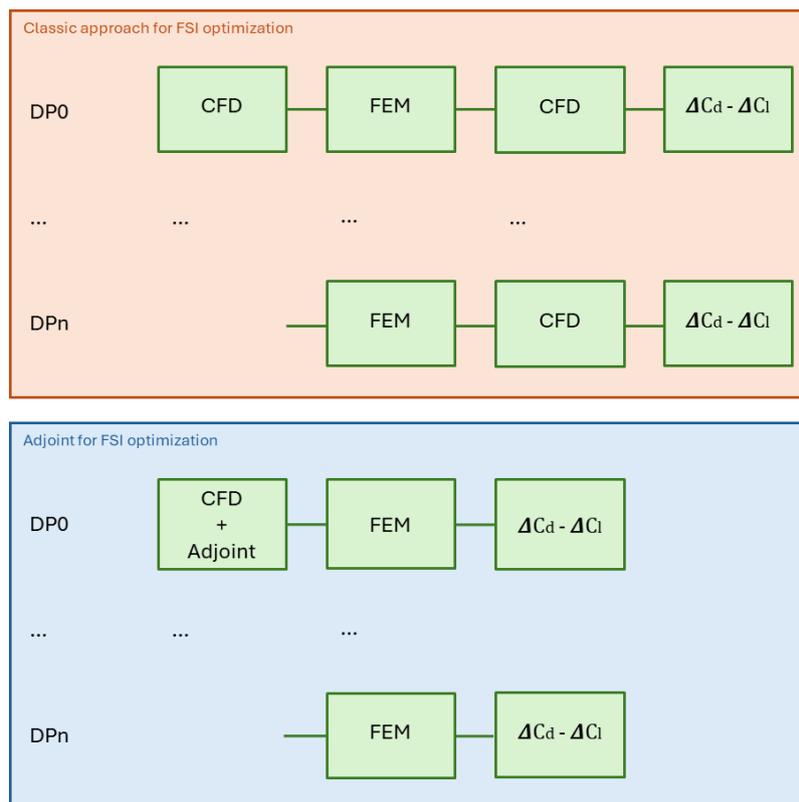


Figure 112: Comparison of classic and adjoint based workflow for stacking sequence optimization

5.3.4. Conclusions

In this study, a method to estimate the influence of structural deformation on aerodynamics was proposed. The method was applied to geometries of increasing complexity, achieving good results. Specifically, for the last case, the front wing, three different usage scenarios were proposed. Particularly interesting is the ability to estimate the influence of material changes on aerodynamics, within the context of a reliable optimization workflow with

low computational impact. This approach becomes even more compelling when placed within the F1 environment, where each CFD simulation must be carefully considered within the overall budget.

6. Reduced Order Models (ROM) Applications

Regarding ROMs (section 2.4), several codes were tested, both open-source (Python-based) and commercial (Ansys Twin Builder), on cases of increasing complexity: a 2D aerofoil, a NACA intake, a scoop intake, OPAM, and CUSP (a CubeSat prototype developed in collaboration with INAF). In particular, the 2D aerofoil was used to assess the accuracy of ROMs by comparing various algorithms for their generation; the NACA intake was employed to create an optimization dashboard where real-time evaluation of field quantities, such as pressure contours, is possible; the scoop intake was tested to evaluate the possibility of loading two different physics (fluid dynamics and structural) into the same real-time dashboard; for CUSP, a ROM was developed for temperature, stress, and deformation (thermo-structural analysis). The idea was to create "virtual" sensors that provide the temperature at any point in the structure, adding useful information for potential structural considerations, such as the stress and deformation fields. This not only simplifies the sensor system and lightens the CubeSat, thanks to the achievable optimised design, but also allows temperature measurement at critical points where sensors cannot be installed. Finally, the OPAM aircraft model was tested to create a real-time optimization dashboard on virtual reality (VR).

6.1. NACA0012 2D Aerofoil

A simple 2D NACA0012 aerofoil was considered, and a dataset was created by keeping the geometry fixed while varying the boundary conditions (AoA and Mach). The analyses were run using the open-source code SU2 for Mach numbers between 0.3 and 0.5, and AoA between 0° and 7°, employing the RANS S-A turbulence model. The focus of this test case was on automating the process to streamline analyses in SU2 and developing ROMs to create predictive models for real-time interaction.

Specifically, a dataset of 135 DPs was generated by varying the angle of attack and Mach number, using a full-factorial DOE with steps of 0.5° for AoA and 0.5 for Mach number. SU2 requires an ASCII configuration file with all input values, so the boundary conditions were modified accordingly to run the simulations. For each DP, the velocity (horizontal component) values were saved at every node of the 2D domain. The generated dataset was used to create ROMs. Various algorithms (in both MATLAB and Python environments) were tested, and the most relevant results are presented below.

Below are the results for the normalized velocity, which is obtained by dividing by the reference value (calculated based on the Mach number of the simulation). Similar considerations can be made for the pressure fields.

6.1.1. POD + GPR

The first approach involves using SVD and GPR (singular value decomposition and gaussian process regression described in the section 2.4.2.3). The code leverages the TensorFlow library, the Gaussian Processes library "GPy" (109), and other libraries for data manipulation and graphical visualization. The total number of samples is 135, with 123 used for training and the remaining for testing. Compared to the entire CFD domain, only elements near the aerofoil were selected (Figure 113). This allows us to focus on the relevant area, reducing the ROM error and computation time.

The extended domain is necessary for CFD analysis to capture the full flow field and improve solution convergence. However, it poses a limitation when building the ROM. The region far from the airfoil, with nearly constant pressure and velocity values, negatively impacts ROM estimation. Additionally, by evaluating fewer nodes, the creation of the ROM becomes faster. For this reason, only the cells near the airfoil were selected. The first 20 layers are selected.

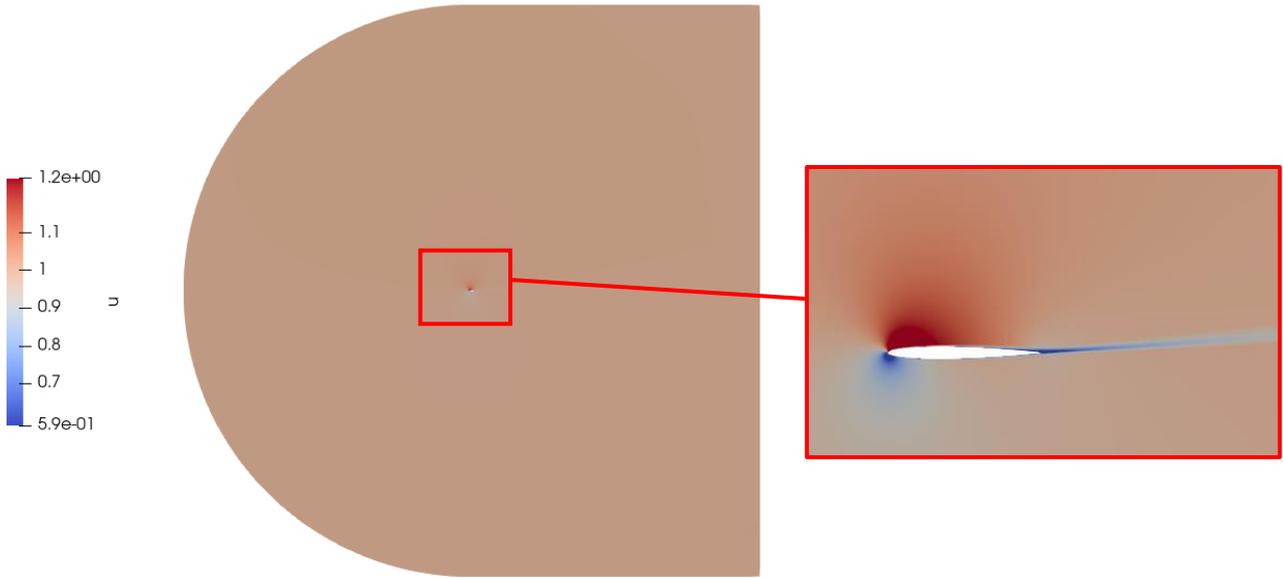


Figure 113: Entire CFD domain, normalized velocity [m/s]

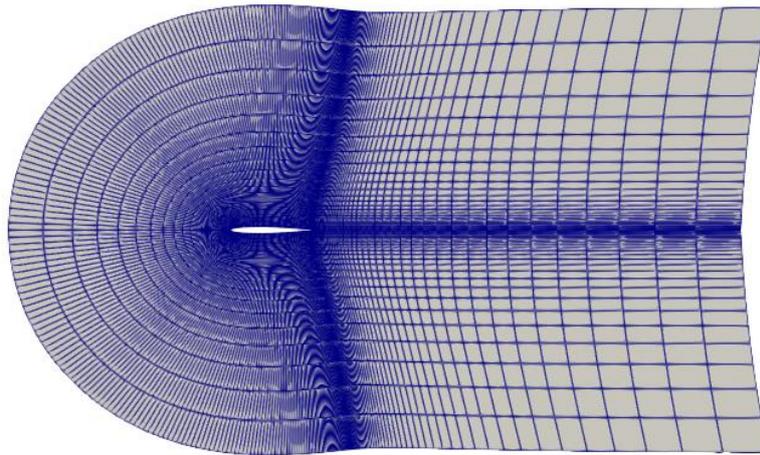


Figure 114: Selected domain for ROM training and test

For the points in the test set, the prediction error for velocity (u components) and pressure was evaluated. The table below shows the maximum error in the estimation of u and p , while the figures below provide a comparison between the CFD results and the ROM estimates. The error is defined as:

$$e = \sum_{i=1}^n e_{1i} \quad (6.1)$$

Where n is the number of snapshots, e_1 is evaluated for each snapshot:

$$e_1 = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (pred_i - exact_i)^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N (exact_i - \overline{exact})^2}} \quad (6.2)$$

N represents the number of considered nodes, $pred_i$ refers to the predicted pressure and velocity values at each node, $exact_i$ refers to the exact values at each node, \overline{exact} is the mean value of $exact_i$.

Finally, the maximum absolute error is also considered, which is simply defined as:

$$e_m = \max(|pred - exact|) \quad (6.3)$$

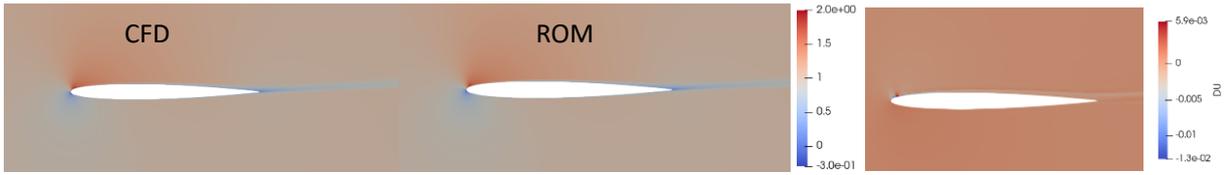


Figure 115: : Comparison of normalized velocity [m/s] evaluation of CFD and ROM (left) and error (right)

Table 32: ROM error

e velocity	e_{\max} velocity
$3.08 e-03$	$1.22 e-02$ m/s

6.1.2. POD + RBF Network

Using the same dataset described earlier, the POD + RBF Network (radial basis function) algorithm was also applied (Section 2.4.2.4). If the data size is below a predefined memory threshold, the method uses singular value decomposition (SVD) to obtain the orthogonal basis of the principal modes. Alternatively, if the data size is too large, an approach based on the eigenvalues of the covariance matrix is used. In terms of accuracy, the error is minimal and negligible, especially considering the truncation applied to the POD basis. In fact, in both cases, cumulative energy is calculated to determine how many modes to include in the truncated basis, adhering to the set energy threshold. A cumulative energy threshold of 99% was imposed.

A key aspect of the implementation is the optimal calculation of the shape parameter for the RBF (79). The method employs a bisection algorithm to determine the optimal shape parameter that minimizes the condition number of the collocation matrix. This ensures that the solution obtained from the RBF is numerically stable. Finally, the RBF coefficient matrix is constructed, representing the normalized quadratic distance between the training points. The matrix is based on the Inverse Multiquadric (IMQ) function, a common choice in the context of RBFs due to its ability to produce smooth and well-conditioned solutions. This method is essential for the subsequent inference in the solution reconstruction process.

The results in terms of computational time and ROM accuracy are comparable to those obtained using the method mentioned above, though the results are slightly less accurate.

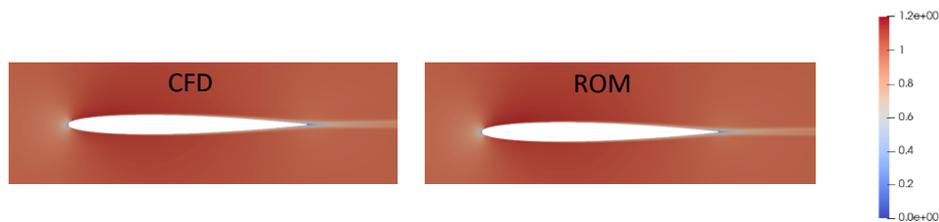


Figure 116: Comparison of normalized velocity [m/s] evaluation of CFD analysis and ROM

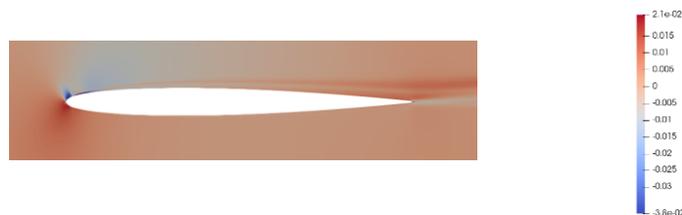


Figure 117: ROM error on velocity evaluation [m/s]

Table 33: ROM error

e velocity	e_{\max} velocity
$8.22 e-03$	$5.05 e-02$ m/s

6.1.3. POD + Neural Network (NN)

The final method tested was POD + NN (Section 2.4.2.2). The same dataset described earlier was used, and a MATLAB code was implemented that utilizes SVD decomposition to identify the modes and the cumulative energy to determine the number of modes to evaluate. As in the previous case, a cumulative energy threshold of 0.99 was set. The LM-based neural network is developed to link input parameters and modes weights. This algorithm is implemented using MATLAB's `trainlm` function (110). The accuracy and computational times are consistent with the methods previously reported.

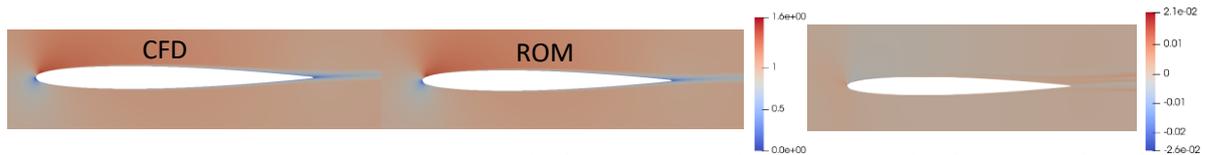


Figure 118: Comparison of normalized velocity [m/s] evaluation of CFD and ROM (left) and error (right)

Table 34: ROM error

e velocity	e_{\max} velocity
$8.41 e-03$	$5.85 e-02$ m/s

6.1.4. Conclusions

In this study, various algorithms for generating Reduced Order Models (ROM) were compared. The focus was particularly on surrogate models that link input parameters to the weights of the extracted modes. All the methods examined for the problem at hand yielded excellent results in terms of both accuracy and training time. Although this study is preliminary, based on a relatively simple testcase with only physical parameters, it allowed us to familiarize ourselves with the described methods and select the most promising approaches.

6.2. ROM NACA Air Intake

Given the increasing electrification of aircraft and the need to dissipate high amounts of heat, the design and optimization of air intakes is a central and highly relevant topic. In the aeronautical field, there are various solutions available. NACA air intakes are an interesting option since they typically have a lower drag impact, but are usually used for lower flow rates compared to scoop intakes. For an electric aircraft, having low aerodynamic resistance is essential to increase autonomy. For this reason, attention has been focused on the design of NACA air intakes, while fixing the required flow rate and optimizing pressure drop.



Figure 119: NACA air intakes

The following sections present the entire workflow that led to the development of a ROM for a NACA-type air intake. Specifically, a multi-fidelity approach is used. The preliminary sizing is performed using a Python code based on semi-empirical analytical relationships. The output of this code provides the sizing parameters of the air intake and an estimate of its performance. The geometric data is then used to generate the baseline CAD model. In this step, the approach transitions from analytical to numerical. The CFD analysis provides much more detailed information, enabling a refined optimization process using RBF Mesh Morphing to define a parametrization directly on the mesh. The advantages of this approach are twofold:

- Reduced computational time
- Iso-topological mesh

Having an iso-topological mesh is crucial for creating a ROM. The ROM is exported in FMU format for easy transportability. Specifically, it was imported into a MATLAB environment, where an optimization dashboard was created. The workflow is completely agnostic to the CFD solver used. To create the dataset, two CFD solvers were employed: one open-source (SU2) and one commercial (Fluent).

6.2.1. Preliminary Design

The typical preliminary design approach of air intakes makes use of the ESDU86002 standard (111). It is based on semi-empirical formulations that allows to size and to estimate the performance of intakes for given input requirements and operative conditions. The main physical parameters involved are:

- Mass flow rate
- Momentum thickness
- Boundary layer thickness
- Mach number

Figure 120 sketches the topology of a flush NACA intakes with a curved-divergent ramp. Its highly swept planform causes vortices to develop along the ramp sides introducing losses but generating a pressure recovery larger than the theoretical two-dimensional value. Its performance in general exceeds also the pressure recovery than can be obtained by other flush inlets. The intake was dimensioned to operate with a mass-flow rate of 0.23 kg/s at a Mach number of 0.52. In ESDU86002 it is suggested a ramp angle (α) of 7° and an inlet width to depth ratio (w/d_t) of 4. Geometrical parameters w and d_t represent the width and height of the inlet section, as shown in Figure 121. Such values usually allow to achieve better drag and pressure recovery performance. The other geometric values are to be evaluated based on the physical properties under examination.

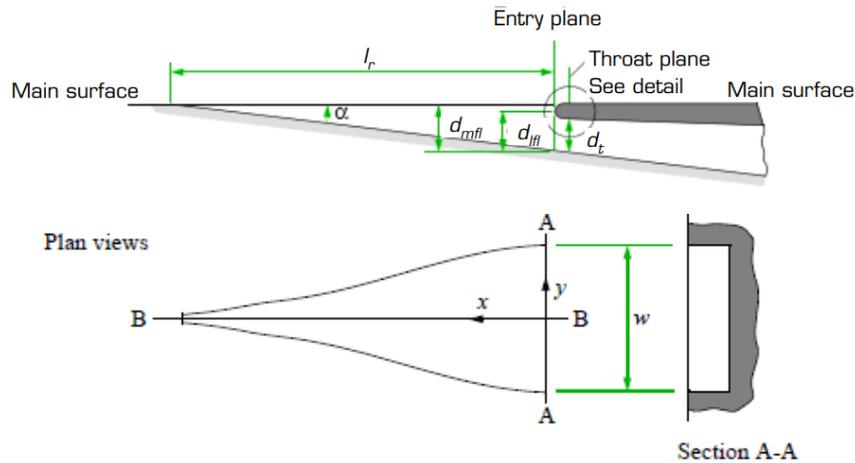


Figure 120: Scheme of NACA air intake from ESDU86002

The method is iterative, and the first step is to determine the inlet section. It starts with a first attempt value:

$$A_1 = \frac{\dot{m}}{\rho_0 V_0} \quad (6.4)$$

Where \dot{m} is the requested flow rate, while ρ_0 and V_0 are respectively the density and the velocity of the undisturbed fluid. A_1 is the initial value of the inlet area of the air intake

Among the curves shown in the standard, only those associated with the range of interest were considered. The design point is the cruise point ($M = 0.52$). So, for example, in the diagram in Fig.121, the points on the curves for $M = 0.4$ and $M = 0.6$ were evaluated, and the points were interpolated for $M = 0.52$ to derive a mathematical law.

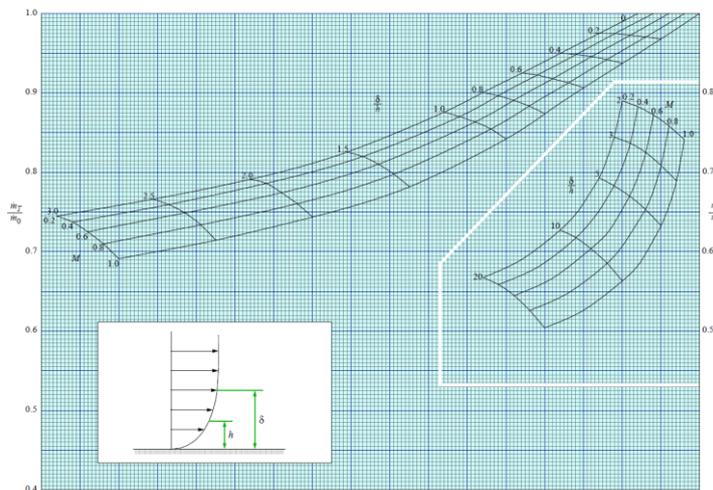


Figure 121: Curves taken from ESDU standard

Here are the main steps followed for the sizing. For a more detailed explanation of all the steps, please refer to the reported standard (111).

Considering an elliptical lip:

$$d_t = \sqrt{\frac{0.25A_1}{1.125 - 0.25 \tan \alpha}} \quad (6.5)$$

$$x = \frac{\delta}{dt} \rightarrow \frac{\dot{m}_t}{\dot{m}_0} \quad (6.6)$$

$$\dot{m}_i = \frac{\dot{m}_t}{\dot{m}_0} \cdot V_0 \cdot A_{1(i-1)} \quad (6.7)$$

$$A_{1i} = A_{1(i-1)} \cdot \frac{\dot{m}_{req}}{\dot{m}_i} \quad (6.8)$$

In these equations, the same nomenclature as the standard has been used. δ represents the boundary layer thickness and is an imposed boundary condition. In step (6.6), the diagram in Figure 121 is used to calculate the ratio $\frac{\dot{m}_t}{\dot{m}_0}$, which represents the actual flow rate entering the air intake compared to the flow rate that could be processed.

These equations are iterated until A_1 and \dot{m} converge. At the end of this process, section A_1 is obtained such that, considering losses and the shape of the air intake, the required flow rate is precisely processed. Once section A_1 have been identified from simple geometric considerations, the entire intake can be characterized. The final geometry obtained is given in Figure 122.

This procedure was used for a preliminary design. The use of CFD allows for exploring a wider range of scenarios and identifying the optimum design more accurately

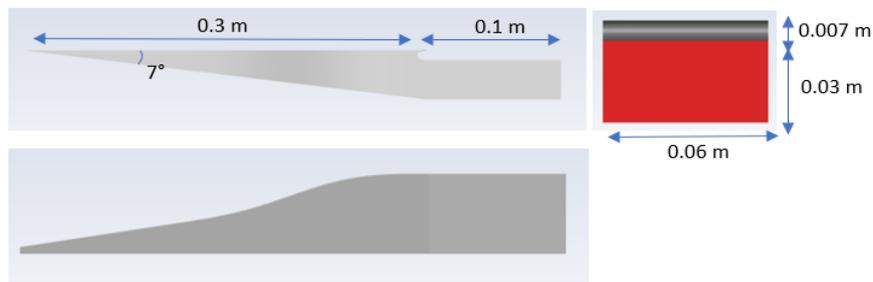


Figure 122: Baseline geometry

6.2.2. Computational domain

To reduce the size of the computational domain, we leveraged the geometrical symmetry property of the duct to generate a half-domain, assuming the development of a symmetric flow field. The NACA air intake was placed inside a volume 2 m long, 0.5 m wide and 0.5 m tall (Figure 123). A multiblock structured grid composed of about 600.000 hexahedral cells was generated. Figure 124 displays the mesh on the symmetry plane and reports details of the cells clustering in the boundary layer and on the lip of the intake. The layering of the cells close to the wall was dimensioned assuming to adopt wall functions to model the boundary layer. In this way, it was possible to use a coarser mesh and reduce computational times.



Figure 123: Computational domain

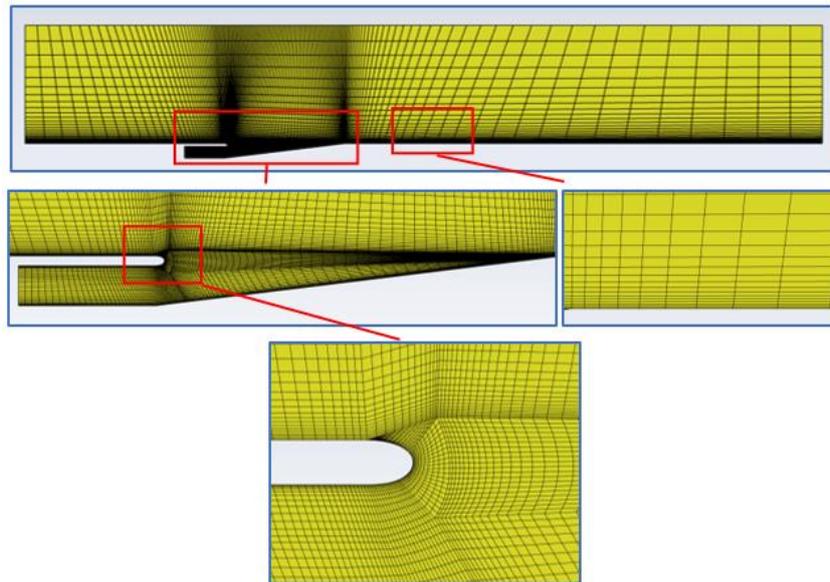


Figure 124: CFD Volume mesh

6.2.3. Boundary conditions

Steady compressible fully turbulent RANS analyses were performed utilizing the $k-\omega$ SST turbulence model. The coupled scheme was employed for the pressure-velocity coupling. Second-order schemes were utilized for spatial discretization. Gradients were computed using the Least-Squares-Cell-Based method. A *velocity inlet* condition was imposed at the inlet of the domain assuming a uniform velocity equal to 167 m/s. A *pressure outlet* was set at the outlet. All walls were modelled as smooth no-slip walls. Zero static gauge pressure was imposed at the outlet of the air-intake imposing the required target mass flow rate of 0.125 kg/s.

6.2.4. Baseline Results

The analyses were carried out using two CFD solvers, one open-source (SU2) and one commercial (Fluent). The idea is to demonstrate the flexibility of the method, which is completely agnostic to the CFD solver used. This section presents a comparison between the results obtained using the baseline mesh

Table 35: Comparison of Fluent and SU2 results

	Fluent SST	SU2 SST
Total Pressure [Pa]:		
Inlet	48887.49	48893.08
Outlet	48752,75	48732.54
Outlet NACA	46979,22	46607.94
Static Pressure [Pa]:		
Inlet	41797.56	41805.78
Outlet	41770	41769.96
Outlet NACA	42434.49	42327.79
Density [kg/m³]:		
Outlet NACA	0.505	0.505
Velocity [m/s]:		
Outlet NACA	130.26	130.16
Massflow [kg/s]:		

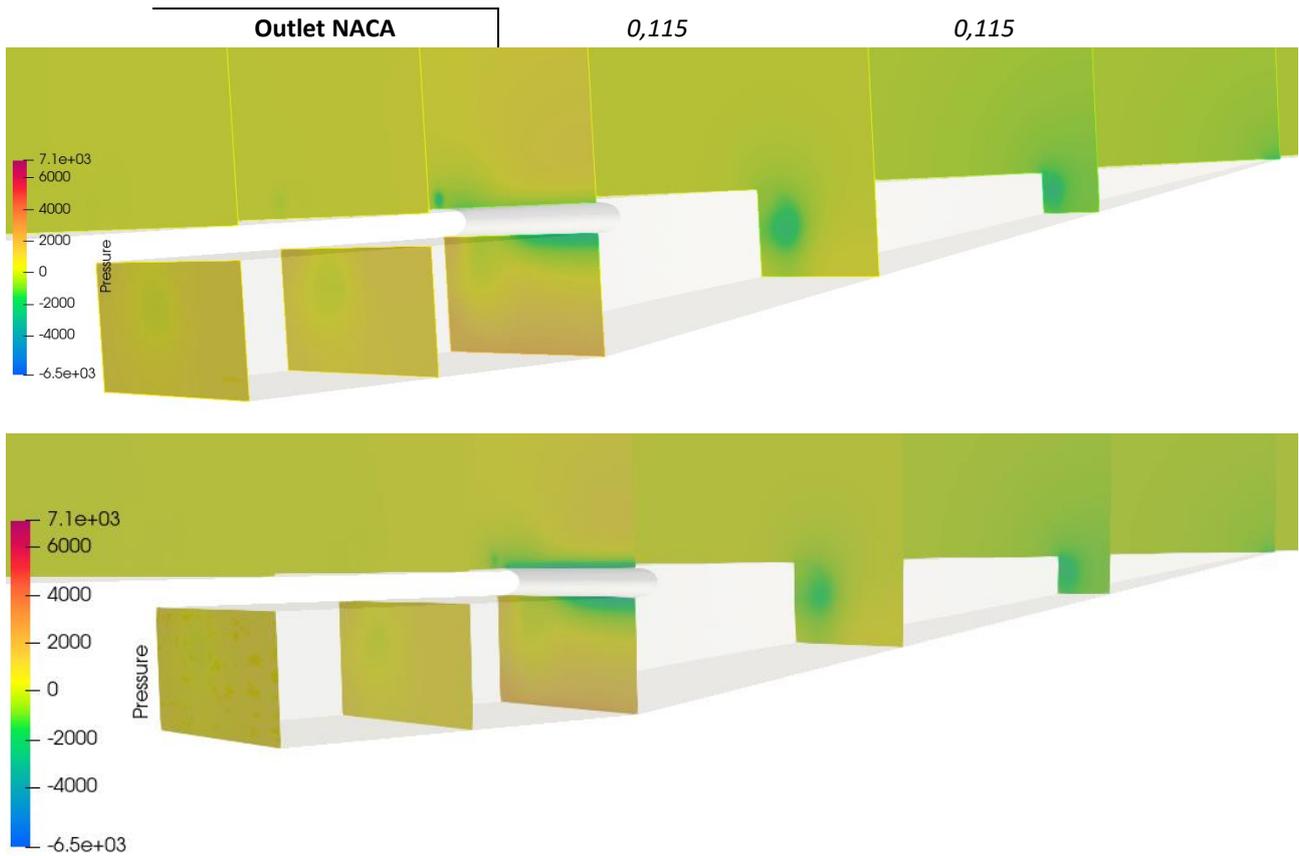


Figure 125: Static pressure [Pa] contour plot comparison between Fluent (upper) and SU2

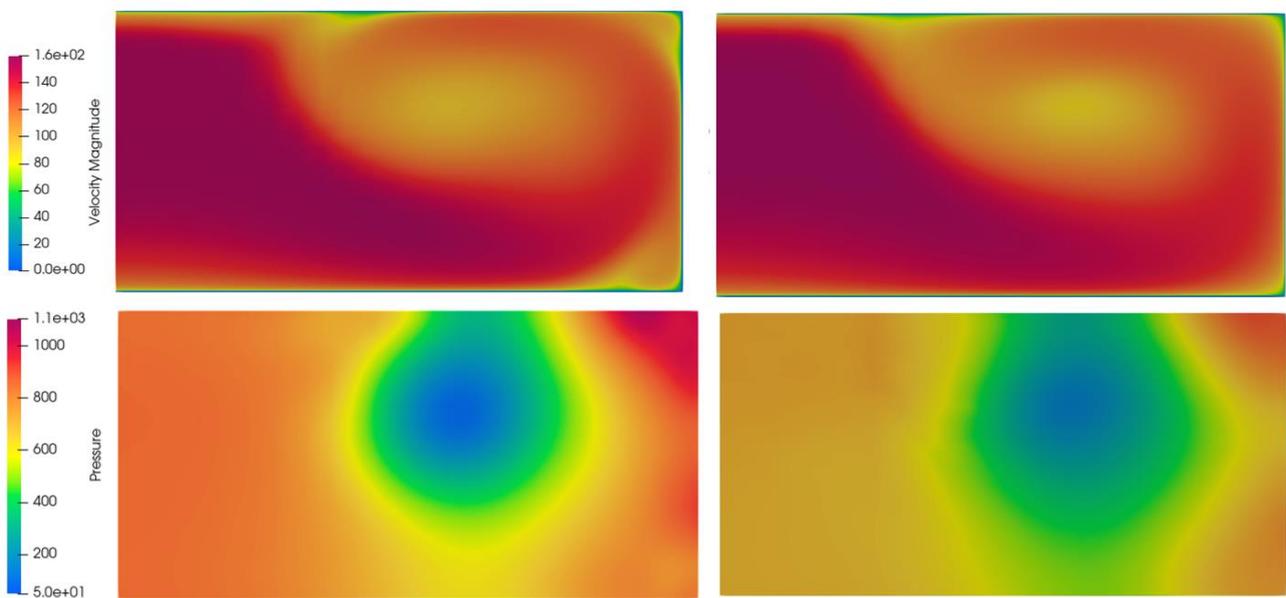


Figure 126: NACA Outlet results comparison between Fluent (left) and SU2 (right), Velocity [m/s] and Static pressure [Pa]

6.2.5. Shape parameters

The RBF Morph software (112) facilitates the intuitive definition of shape modifications by employing RBFs to interpolate displacements specified on mesh nodes. Source points can be defined by directly selecting nodes on surfaces or by utilizing "encaps". Encaps refer to reference solids (spheres, cylinders, or parallelepipeds) within which points are positioned at user-defined uniform intervals. These encaps can serve to specify displacements

(moving encaps) or delineate the region affected by the morphing process (domain encaps). The RBF Morph software was used to define 8 shape parameters. The first 5 parameters control the ramp profile. Each of them acts locally. The overlap of the 5 parameters allows to change the shape of the ramp profile. The preview of these 5 parameters is shown in Figure 127÷130. The source points for the first parameter are shown, and the same logic is extended to the other parameters.

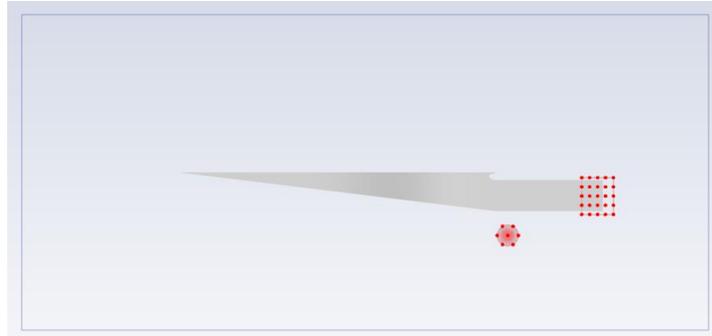


Figure 127: Defined source points. A vertical displacement is defined on the points of the cylinder, while the points on the outlet of the air intake remain fixed. The blue box delimits the morphing domain

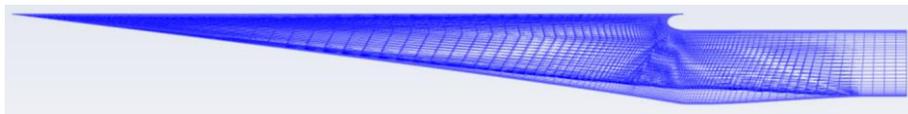


Figure 128: Cyl_o1 preview

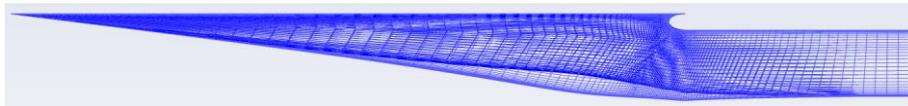


Figure 129: Cyl_o2, preview

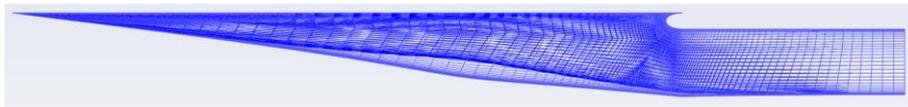


Figure 130: Cyl_o3, preview

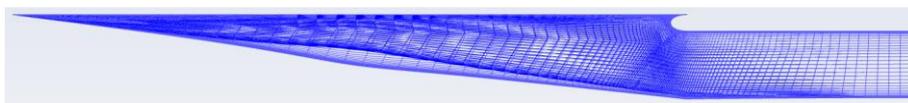


Figure 131: Cyl_o4, preview

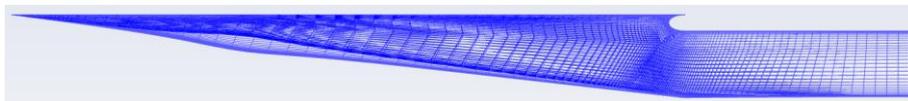


Figure 132: Cyl_o5, preview

One parameter controls the ramp length and its angle. Again, a cylindrical moving encap was used to define the source points and displacements, fixed points were defined by directly selecting surfaces. The preview is shown in figure 133.

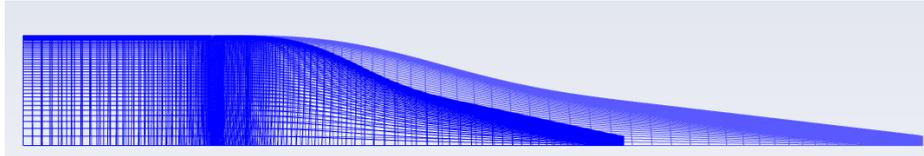


Figure 133: Length preview

One parameter controls the constricted area on the lateral surface (Figure 134).

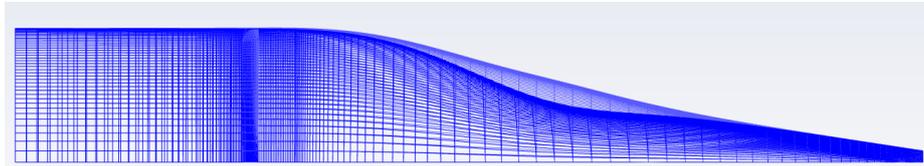


Figure 134: Lateral striction preview

Finally, the last parameter moves the entire lateral surface (Figure 135).

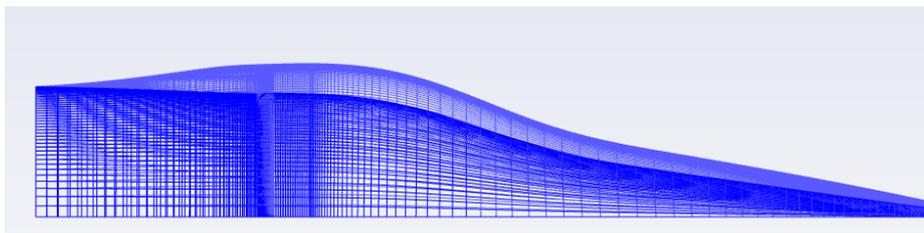


Figure 135: Lateral, preview

6.2.6. ROM Building

The goal is to have an optimization platform where the best combination of shape parameters can be identified visualizing the flow field in real-time. To accomplish this, a ROM of the studied case was extracted. The following workflow was followed to create the ROM:

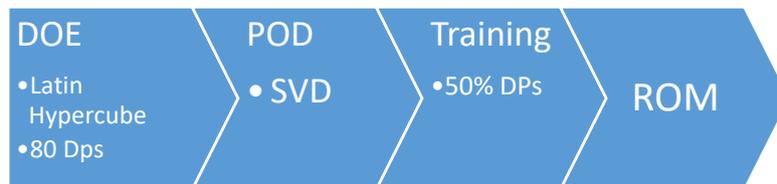


Figure 136: ROM building workflow

DOE is the first step. For this study, 80 snapshots were used. The snapshots were identified using Latin Hypercube sampling. Each snapshot comes from the solution of a CFD analysis of a geometry generated morphing the baseline mesh with a combination of shape parameter amplification factors representing a point of the variables design space.

6.2.7. Ansys Twin Builder

The next step is data compression, aimed at reducing the number of variables in the system and simplifying the problem. Ansys Twin Builder (113) was utilized to perform POD and create the ROM. Ansys Twin Builder is a software designed for creating ROMs. In our case, each snapshot contains pressure values for every node on the air intake surface. The tool allows selection of the number of snapshots to be used for training, as well as the number of modes for reduction, while monitoring error at each step. The choice of these hyperparameters is then made based on a trade-off between error (considering underfitting and overfitting) and model complexity. So 14 modes were selected (Fig. 137) and the model was trained using 50% of the snapshots. The number of snapshots for training is set based on the obtained error. Using 50% of the points ensures a low error and a sufficiently large test set for evaluations. Once the ROM was created, it was possible to update the shape and

the physical parameters. The solution field can be now visualized in real-time. Figure 138 reports the Ansys Twin Builder interface with which is possible to interactively update the shape parameters amplification factors and to visualize in real time the updated geometry with its corresponding pressure field.

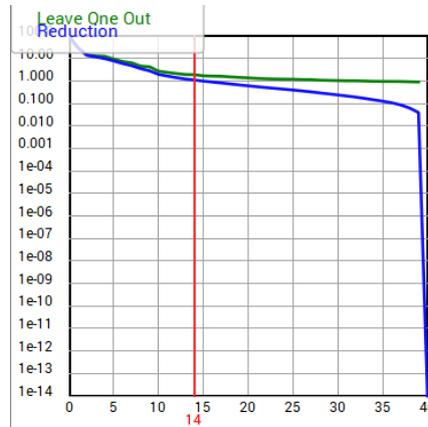


Figure 137: Reduction and LOO error for NACA ROM, y axis is the error, x axis is the number of selected modes

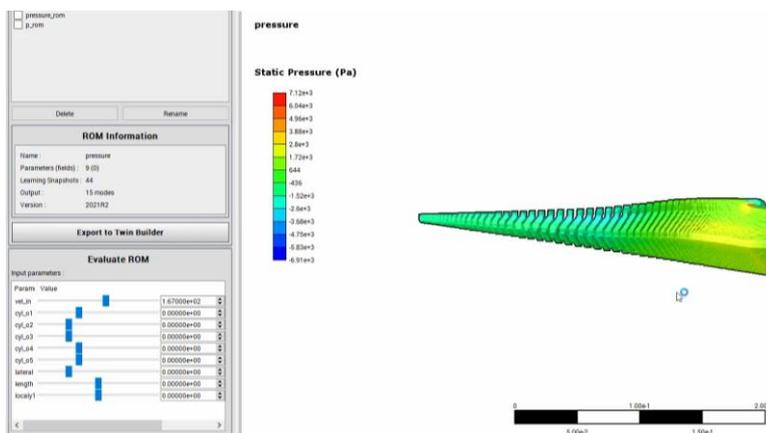


Figure 138: Ansys Twin Builder Interface

The obtained ROM was exported as a FMU, imported into the MATLAB-Simulink environment and coupled to an optimization algorithm. In this case, the gradient descent method was used to identify the optimum. So, at each optimization step, a parameter variation of $da = 0.1$ is applied, and the ROM calculates the outlet static pressure P_i . Then, the gradient is computed using finite differences: $g = \frac{(P_i - P_{i-1})}{da}$. A proportional amplification is then applied:

$$a_i = a_{i-1} + kg \quad (6.9)$$

Where k is a user-set multiplicative parameter. A larger k leads to faster convergence, but excessively large k values may result in poor sensitivity in detecting the maximum. This process is repeated for each parameter, and only after calculating the amplifications i for each parameter, the optimization proceeds to the next step.

An interface was created in MATLAB environment (Figure 139). The right part of the interface allows to maximize the pressure setting fixing the maximum number of iterations and the proportionality coefficient of the steepest descent method (k). During each iteration the values of parameters are updated in the proper boxes and pressure output calculated using the ROM. In the left part, it is possible to set the parameter values manually. By the "Evaluate" button, the pressure field for the defined set of parameters is shown in real-time.

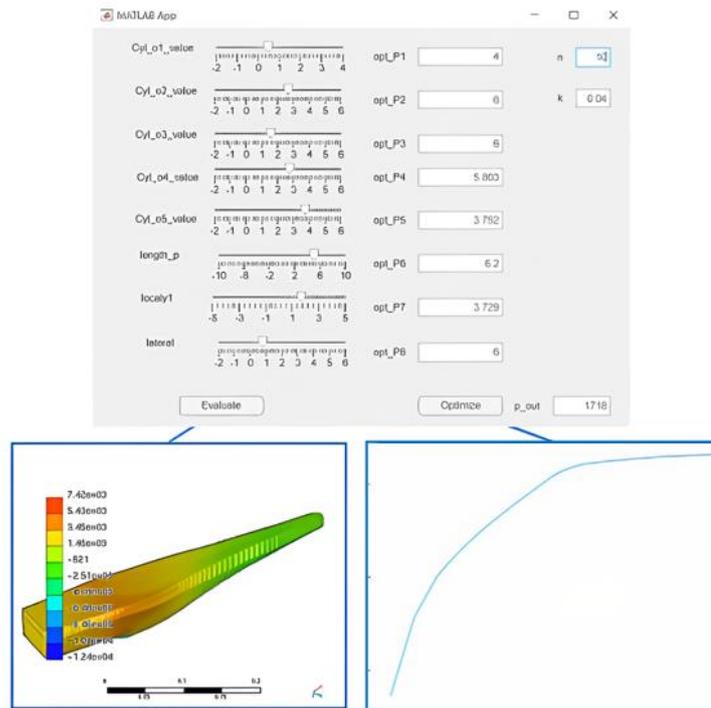


Figure 139: Matlab Optimization dashboard

To assess the quality of the proposed methodology, the results obtained with the use of ROM were compared to those achieved using plain CFD evaluations. In Figure 140 the pressure contours obtained by the ROM are shown on the left, while the ones calculated using CFD are shown on the right for 3 points. Each point is a different combination of input parameters (the same value for CFD and ROM).

The total error has two parts: reduction error, which can be enhanced by increasing modes and learning data set, and interpolation error, minimized by increasing the learning data set. For each of the 80 DPs, the total error (for both the points used for training and the others) is calculated. The maximum error evaluated is in the order of 4%.

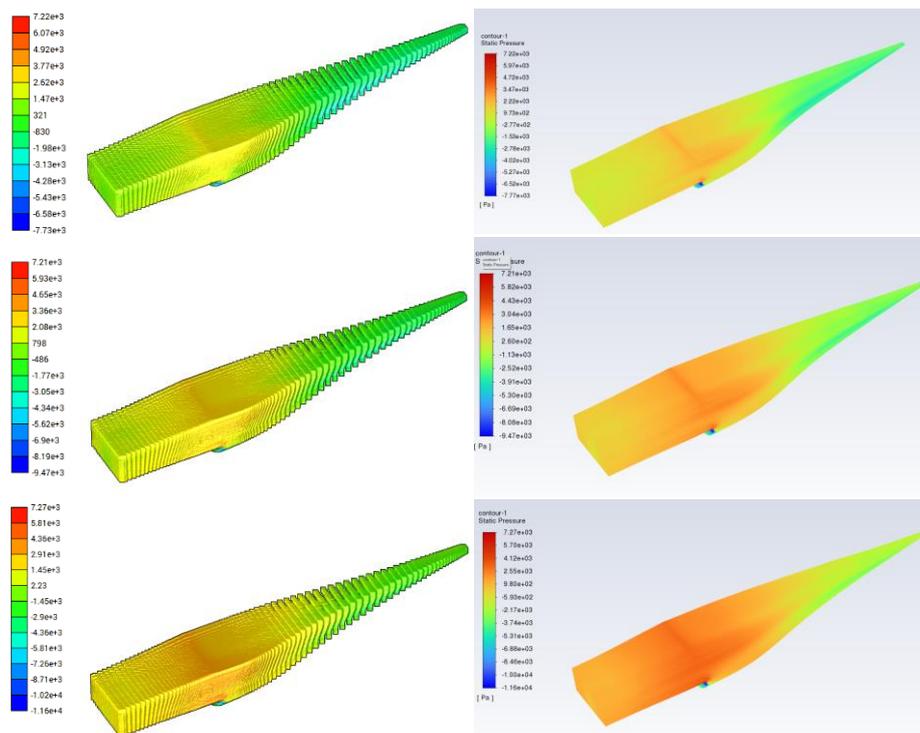


Figure 140: Comparison between ROM (left) and CFD solutions (right) for 3 points

Table 36: Comparison of outlet static pressure [Pa] between ROM and CFD for 3 points

ROM	CFD
722.2	743.1
1120.5	1165.3
1432.1	1482.2

Figure 141 compares the optimal design with the baseline geometry. Figure 142 compares the flow field and Figures 143-144 displays the velocity and the pressure contours. The outlet static pressure of the optimized geometry is 160% larger than the pressure on the outlet of the baseline geometry. Table 38 reports the averaged values of the pressure on the outlets. Finally, Table 39 reports the input parameter values corresponding to the optimized geometry and the ranges considered for each parameter. The range for each parameter was considered both by evaluating mesh quality (excessively large amplifications significantly reduce mesh quality) and by considering physical/topological assessments (excessively large amplifications may result in deformed final geometries with less meaningful shapes).

Looking at the reported results, it can be seen that in the optimized geometry, the flow is better guided at the duct inlet, resulting in a reduction in the maximum velocity and separation recorded in the lip zone. In the baseline geometry, the flow tube has a strong bottleneck in the duct start zone that acts as a concentrated leak resulting in a significant pressure drop. This effect is limited in the optimized geometry.

Table 37: Outlet pressure of baseline and optimized shape

	Baseline	Optimized
P_Out [Pa]	664.5	1718.5

Table 38: Range and optimal value of each shape parameter

	Range	Optimum
P1	-2 ÷ 4	4
P2	-2 ÷ 6	6
P3	-2 ÷ 6	6
P4	-2 ÷ 6	5.8
P5	-2 ÷ 6	3.8
P6	-5 ÷ 5	3.7
P7	-10 ÷ 10	6.2
P8	-2 ÷ 6	6

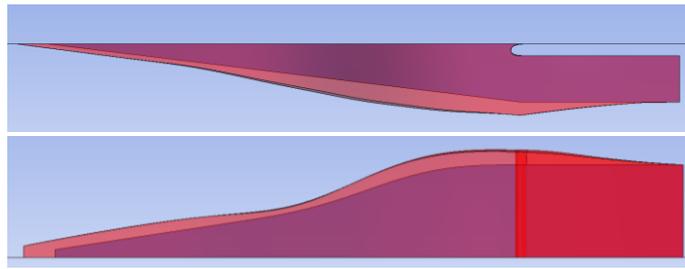


Figure 141: Comparison of baseline (blue) and optimized shape (red)

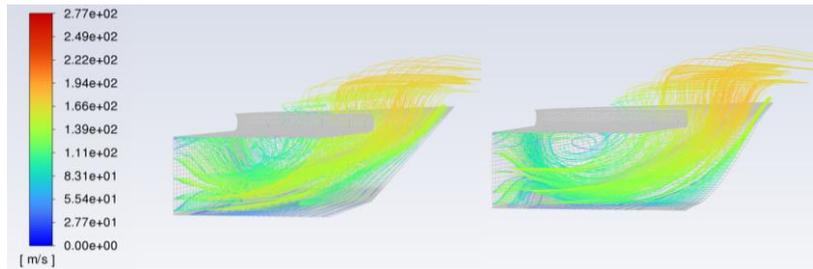


Figure 142: Comparison of velocity streamlines of baseline (left) and optimized (right)

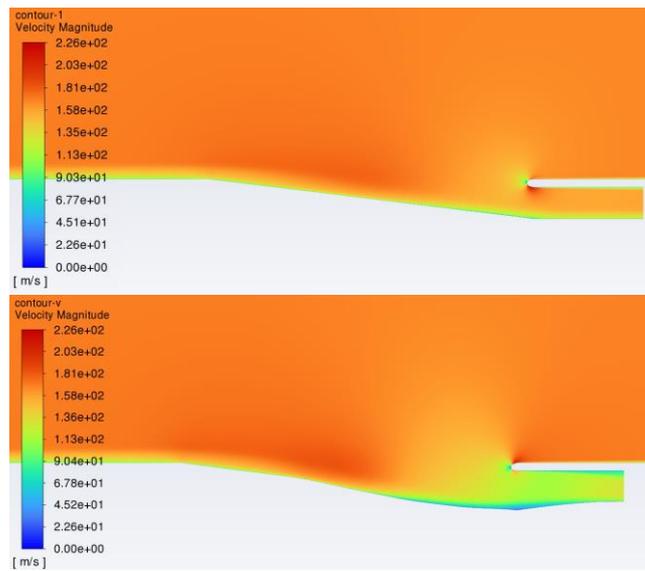


Figure 143: Comparison of velocity contour of baseline (above) and optimized (below)

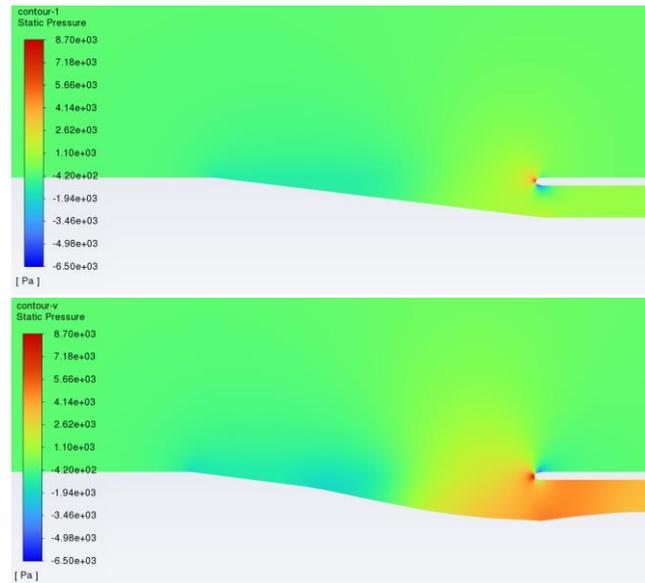


Figure 144: Comparison of static pressure contour of baseline (above) and optimized (below)

In this study, a workflow was developed for the design and optimization of NACA-type air intakes. This problem allowed us to address two challenges. On one hand, the workflow was designed with a multi-fidelity approach, starting from a semi-empirical preliminary design to fine optimization using numerical tools. The second part of the workflow focused on the development of ROMs to condense all the gathered information into a single dashboard. Specifically, the FMU standard was used to transfer information from one tool (Ansys Twin Builder) to another (MATLAB). The final result is a dashboard that allows real-time evaluation of the aerodynamics of any DP and, once the physical variables (such as inlet velocity) are imposed, identifies the optimal geometry. The proposed workflow appears very robust, and the developed methodology can potentially be applied to any problem.

6.3. Scoop air intake ROM

This work is published in (114) where the feasibility of using ROMs for optimizing a scoop air intake for aeronautical applications was evaluated. Since the air intake is exposed to aerodynamic loads, a two-way fluid-structure interaction workflow was used to characterize the component. The goal is to create an optimization dashboard that allows both scalar quantities (the parameters that are intended to be monitored during the design and optimization of the air intake) and field quantities to be evaluated in real time. In this way, the designer can have a full understanding of the physics of the problem and make more informed design choices. In addition, in this way it is possible to visualize results from different physics in a single dashboard, linking different components, interacting with models in real time. A mesh morphing technique based on RBFs was used. The result was very interesting both from a structural point of view (mass reduction over 90% and maximum strain reduction of 36%) and from a fluid dynamic point of view (outlet static pressure 150% higher and drag 32% lower) and the generated ROMs proved to be a very accurate (ROM relative error in the order of 7%) and flexible tool. In this study, the ROM was created from CFD and FEM analysis. Thus, in the final dashboard, fluid-dynamic and structural field quantities can be visualized in real time as geometric and physical parameters change.

A Scoop air intake was considered as a testcase for two reasons:

- As an exposed, external air intake, it is subject to aerodynamic loads to be considered for sizing and structural optimization. Therefore, it is necessary to consider multi-physics analysis and study fluid-structure interaction. In fact, one of the goals of this work is to test the flexibility of ROMs and create a single environment for the different physics involved;
- Given the increasing electrification in aviation, cooling of electronic systems is a central issue. Therefore, developing an updated workflow for air intake design covers a strategic role.

The workflow in the figure 145 was used for the design of the baseline. The reference standard was used for a preliminary design of the air intake. The resulting CAD model was used as a baseline for a CFD optimisation. Finally,

the optimised geometry was studied with a FEM model to identify the correct number of composite plies. In this way, the FEM baseline was realised.

The fundamental assumption underlying this workflow is that the number of composite plies was fixed to ensure a sufficiently rigid structure, such that the pressure field does not impact aerodynamic performance. For this reason, the geometry optimizing aerodynamic performance was identified first, followed by tweaking the number and orientation of the composite plies to minimize structural deformation.

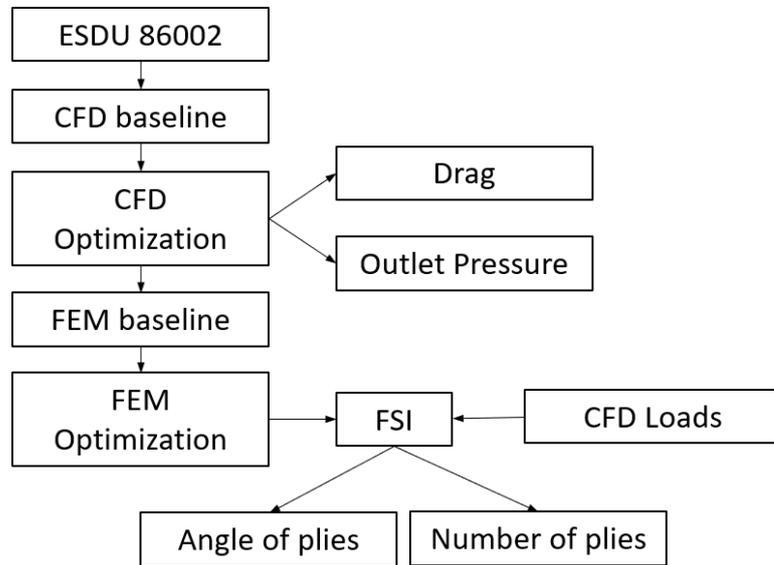


Figure 145: Design Workflow

6.3.1. Preliminary Design

The ESDU86002 (111) standard was used for the preliminary sizing of a Scoop air intake according to defined operative working conditions. Such a geometry was used as baseline on which to generate the computational domain and to implement the shape parameterization. The intake was dimensioned to operate with a mass-flow rate of 0.86 kg/s and cruise conditions (angle of attack of 0 degrees, Mach number of 0.52).

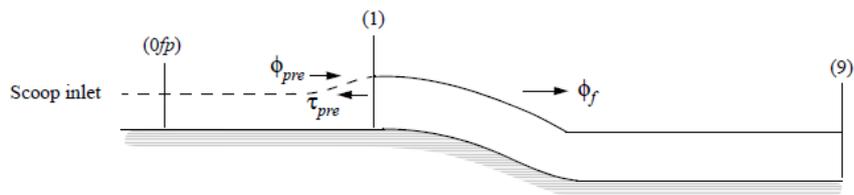


Figure 146: Figure extracted from the standard

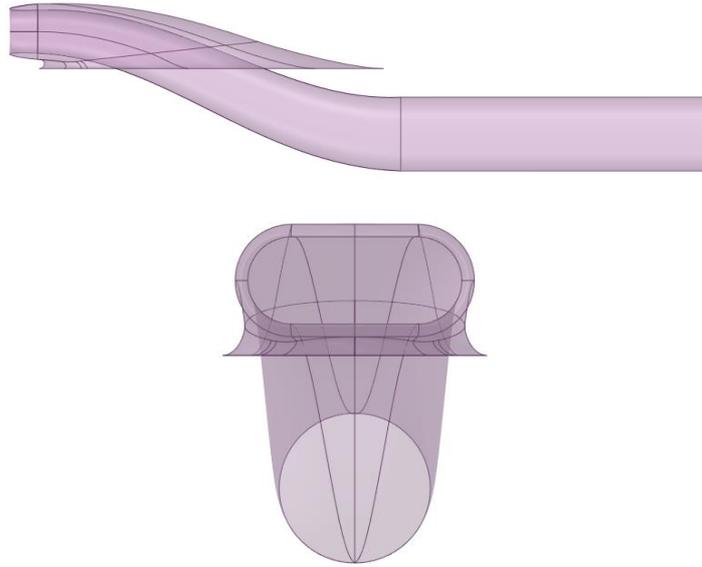


Figure 147: Geometry baseline

6.3.2. CFD Optimization

For the shape optimization, mesh morphing was used to define five shape parameters that act on the most sensitive areas, namely the lip and the junction zone with the ground. The preview of the defined parameters is shown in the figures 148-152, the left and right pictures indicate opposite perturbation.

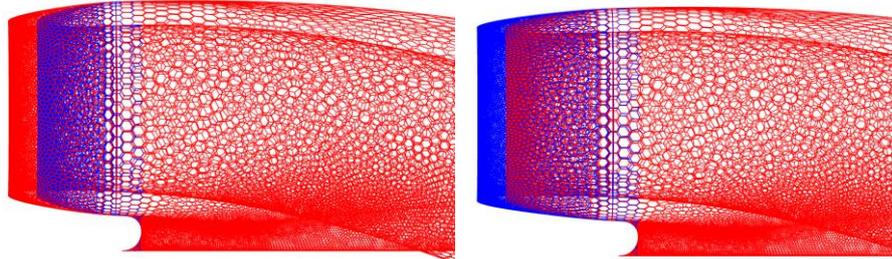


Figure 148: P1, lip elongation

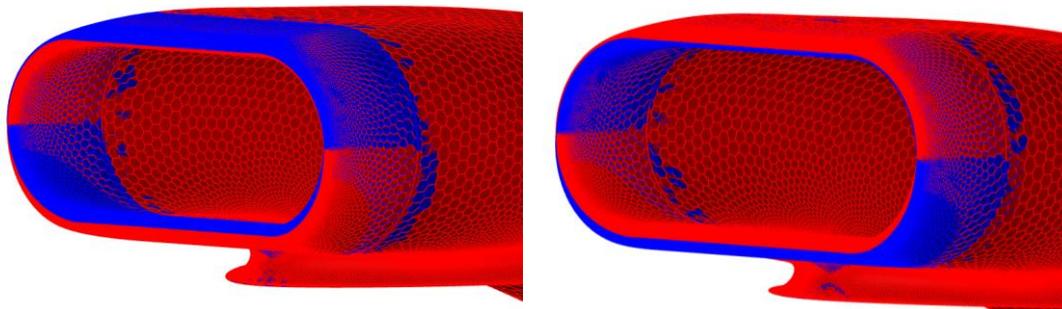


Figure 149: P2, rotation of the lower wall of the lip

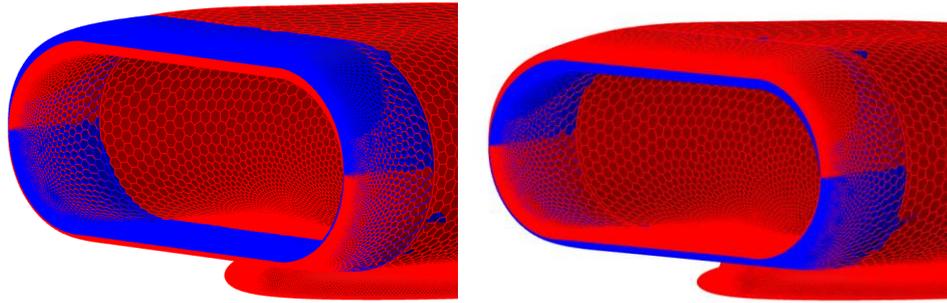


Figure 150: P3, rotation relative to the horizontal symmetric axis of the inlet section of the upper wall of the lip.

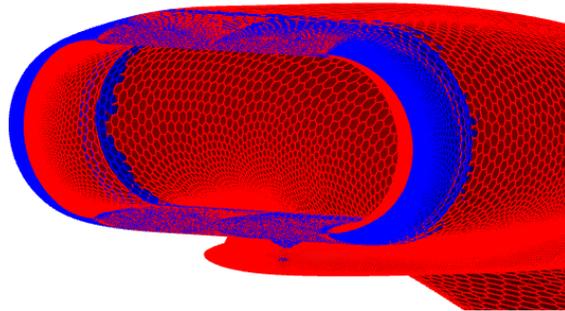


Figure 151: P4, variation of the major axis of the ellipse

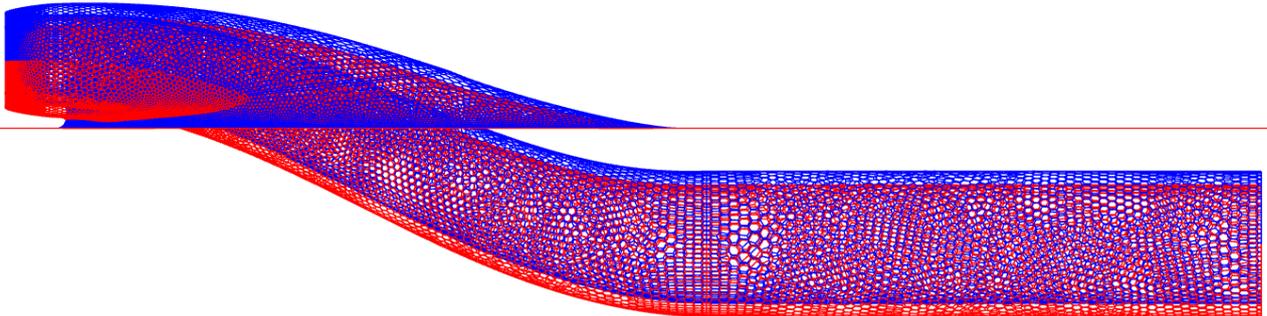


Figure 152: P5, raising/lowering of the air intake

The objective of the optimization is to minimize drag and maximize static pressure at the outlet (linked to pressure recovery). ANSYS Fluent was used for the CFD analyses.

Drag is defined as the integral of static pressure forces and viscous stresses, projected in the horizontal direction, calculated on the outer section of the intake. Static pressure at the outlet is a key parameter for optimization, as it is closely related to pressure recovery and the actual drag of the intake (horizontal forces).

The optimal geometry was identified using the response surface method (Section 2.3.2), leveraging the MOGA algorithm.

Finally, the optimized geometry is presented, which achieves a reduction in both pressure drop and aerodynamic drag. The figures and tables below summarize the key results, showcasing the improvements obtained through the proposed workflow.

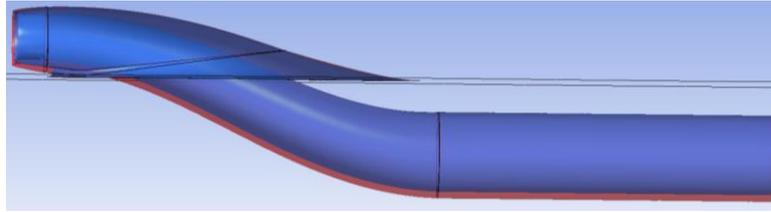


Figure 153: Comparison of baseline (blue) and optimized shape (red)

Table 39: Comparison of drag and static pressure in the outlet surface

	Drag [N]	P_out [Pa]
Baseline	12.8	-1352.5
Optimized	9.18 (-22.9%)	742 (+156%)

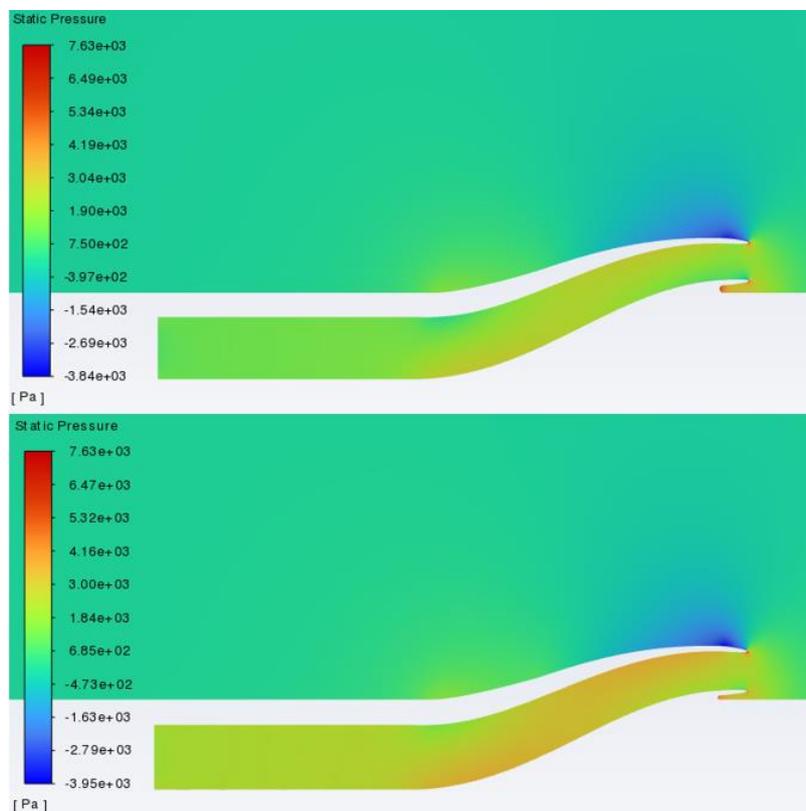


Figure 154: Comparison of static pressure baseline (upper) and optimized (lower)

It can be seen that the separation zone on the lip of the air intake is less developed (drag reduction) and the flow within the duct is better guided. The optimization allows to obtain a reduction of drag about 32% and an increasing of outlet static pressure about 150% .

6.3.3. FEM Baseline

Aerodynamically optimised geometry was used as the FEM baseline. A carbon-fibre-reinforced composite material for aeronautical applications (115) was used, the main characteristics of which are shown in table 40. The Nastran solver was used for the FEM analysis. Table 41 summarises the main mesh statistics and boundary conditions used. The most severe condition was considered for the aerodynamic loads, i.e. $Ma = 0.6$ and angle of attack of 4 degrees (Figure 156). The orientation of the material follows the main flow direction.

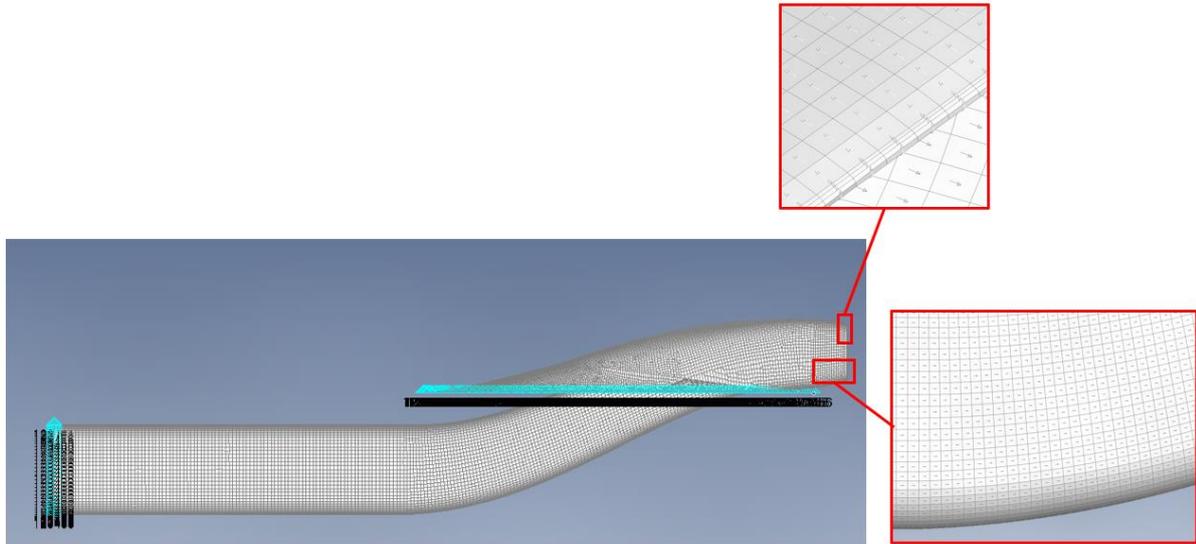


Figure 155: FEM baseline, focus on the orientation of the material

Table 40: Material properties

Density [kg/m³]	<i>1530</i>
E11 [Pa]	<i>1.83 E+11</i>
E22 [Pa]	<i>9 E+9</i>
G12 [Pa]	<i>8 E+9</i>
G1z [Pa]	<i>8 E+9</i>
G2z [Pa]	<i>4 E+9</i>
X1t [Pa]	<i>1.5 E+9</i>
X2t [Pa]	<i>9.5 E+8</i>
X1c [Pa]	<i>7 E+7</i>
X2c [Pa]	<i>2 E+8</i>
S12 [Pa]	<i>8 E+7</i>
nu	<i>0.35</i>

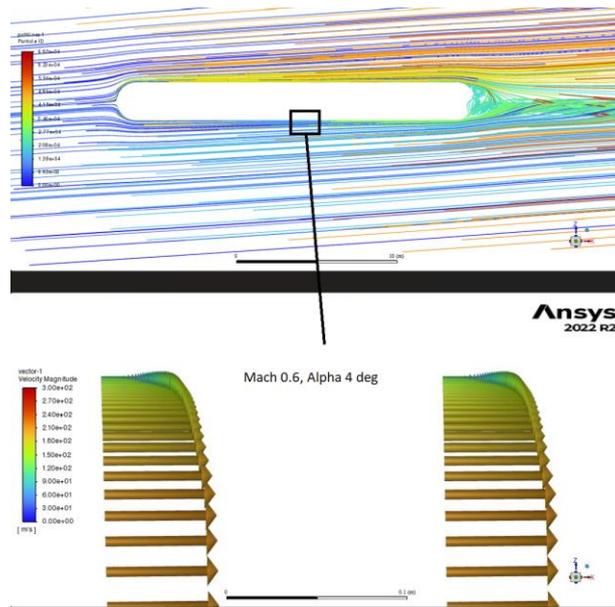


Figure 156: CFD boundary condition

Table 41: Main mesh statistics and Border Condition resume

Laminate Plate,	48874
Linear elements	
Failure Theory	<i>Tsai-Wu</i>
Constraints	<i>Fixed curves</i>
Loads	<i>Pressure field of the most critical point</i>
Parameters:	<i>Number and angle of plies</i>

Finally, an iterative process was used to decide the number of plies. The starting point was the literature configuration: 24 symmetric plies with stacking sequence $[45/-45/0/90/0/45/-45/0]_s$.

The maximum deformation was identified as the output parameter. In fact, the stresses for all the configurations studied are below the limit value and the Tsai-Wu coefficient is always less than 0.5. Conversely, with too high deformations there is a change in the fluid-dynamic behaviour. Therefore, the number of plies below which the structural deformations have a significant influence was identified. In this way, the initial configuration with 24 symmetrical plies with a thickness of 0.25 mm was changed to the final configuration with 4 symmetrical plies with a thickness of 0.125 mm. Reducing the number of plies leads to a reduction in mass of around 92%.

In the graphs shown in figure 157, it can be seen that the final configuration provides good stiffness. In fact, the resulting deformations are not such as to induce significant changes in the fluid-dynamic behaviour and the FSI workflow converges to the first step. Therefore, a simple mapping of the aerodynamic loads (one way method) is sufficient to characterise the structural behaviour.

The last step considers the optimisation of the stacking sequence to minimise the maximum deformation.

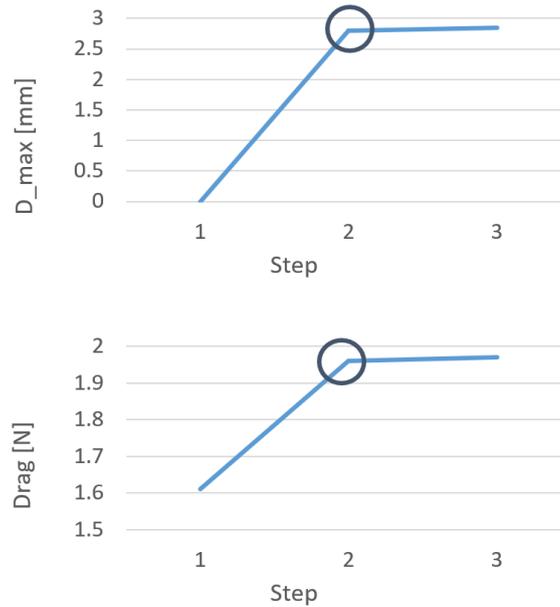


Figure 157: Convergence of FSI workflow for configuration with 4 x 0.125 mm plies [0/90]s

Table 42: Variation of CFD performances for configuration with 4 x 0.125 mm plies [0/90]s

Drag	+0.3 N
Outlet Pressure	+0.3%

6.3.4. ROM

The performed analyses were used to create a ROM. Ansys Twin Builder was utilized as the commercial code, and the obtained results were compared with a MATLAB code employing a neural network whose weights were optimized through LM backpropagation (section 2.4.2.2).

Eighty DPs were employed, with the angle of the 2 plies ranging from -90 to 90. Forty DPs were used for training, while the remaining forty were used as a test set. A comparison of the results reveals that the deformation field predicted by the ROM obtained through Ansys Twin Builder is highly reliable, with a relative error of 6%.

The ROM can be exported as a FMU, and it can be imported into Simulink as a black-box that takes input values and provides the desired field quantity as output. This enables the creation of a dashboard in the MATLAB-Simulink environment, where all the involved physics can be visualized in a unified setting, or different components can be connected to build a more complex model.

Using the same procedure described, the CFD pressure ROM was also built. Both geometric parameters (the 6 shape parameters used for optimization) and physical parameters (cruise velocity) were considered as input parameters. In this case as well, the results are highly accurate, with a maximum relative error of 7%.

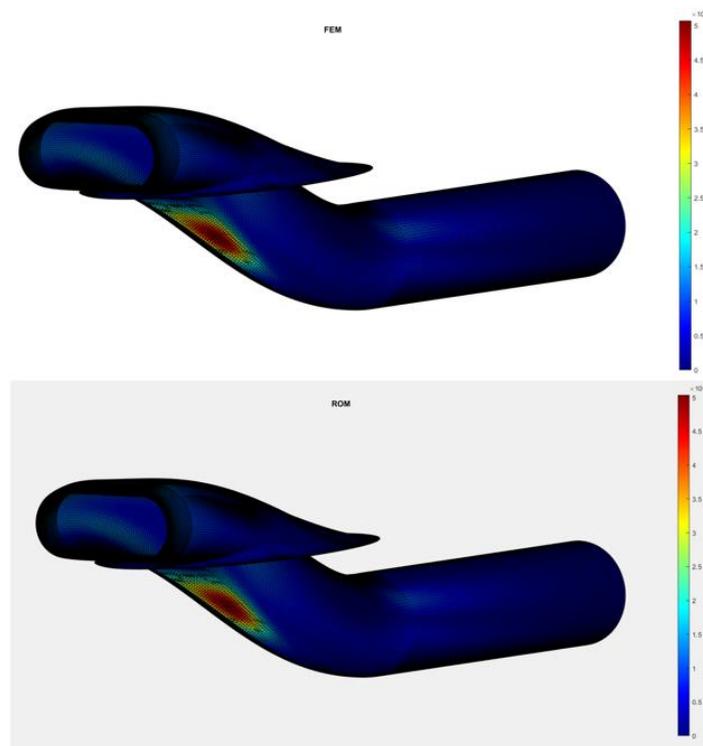


Figure 158: Comparison of FEM (up) and ROM (below) displacements for a random point of the test set

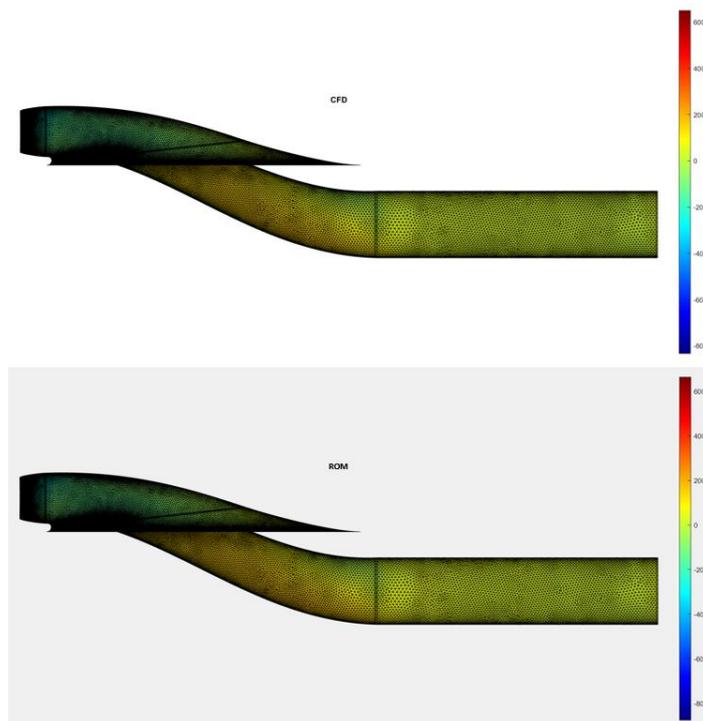


Figure 159: Comparison of CFD (up) and ROM (below) static pressure field for a random point of the test set

Finally, the previously described Matlab code was used. The results are consistent with those obtained using commercial software. In particular, the following two graphs depict the displacement values for each node of a random design point from the test set. It can be observed that the predicted and expected values overlap almost entirely. Furthermore, a comparison of the displacement distribution also shows a good correspondence.

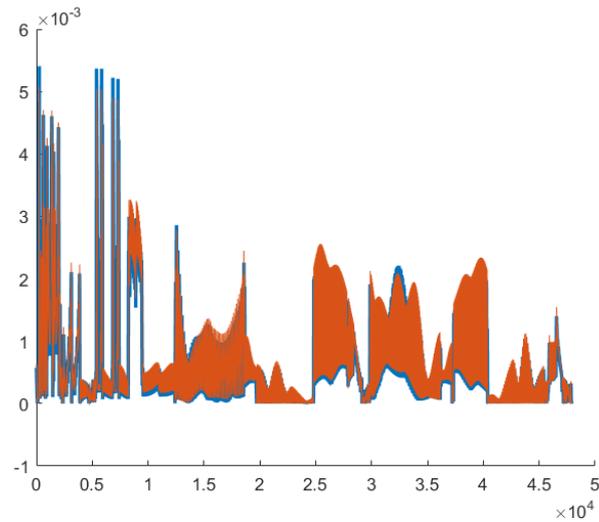


Figure 160: Displacement value for each node real (blue) and calculated by the neural network (orange)

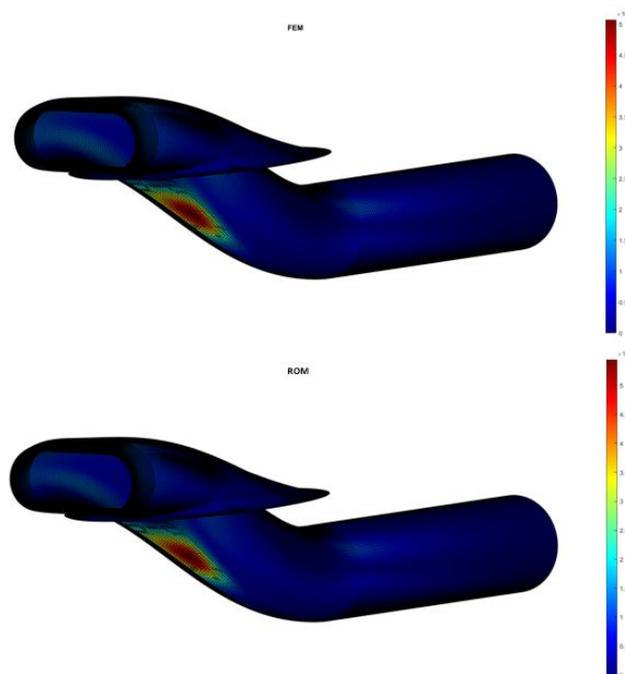


Figure 161: Comparison of FEM (up) and ROM (below) displacements for a random point of the test set

The results from Ansys Twin Builder were exported as an FMU and imported into the Matlab-Simulink environment, where a real-time optimization dashboard was created. In the CFD section, it is possible to modify the shape parameters and visualize their effect on the pressure contour. The results of the ROM for each design point (DP) can not only be visualized but also exported and mapped onto an FEM model for further analysis.

The FEM ROM, on the other hand, was developed with a fixed geometry, varying only the ply angles. This allows the observation of displacement field evolution for each configuration but does not permit shape parameter variation.

The FEM ROMs described were used to identify the optimal ply angle configuration. The optimal sequence, [-90/0]_s, resulted in a 36% reduction in maximum deformation.

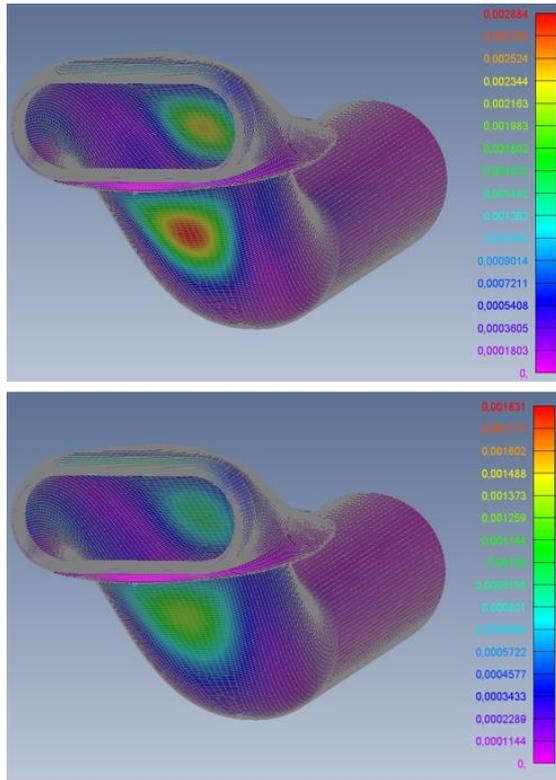


Figure 162: Displacements baseline (above) and optimized (below)

Table 43: Variation of max displacements

Initial	2.88 mm
Optimized	1.83 mm

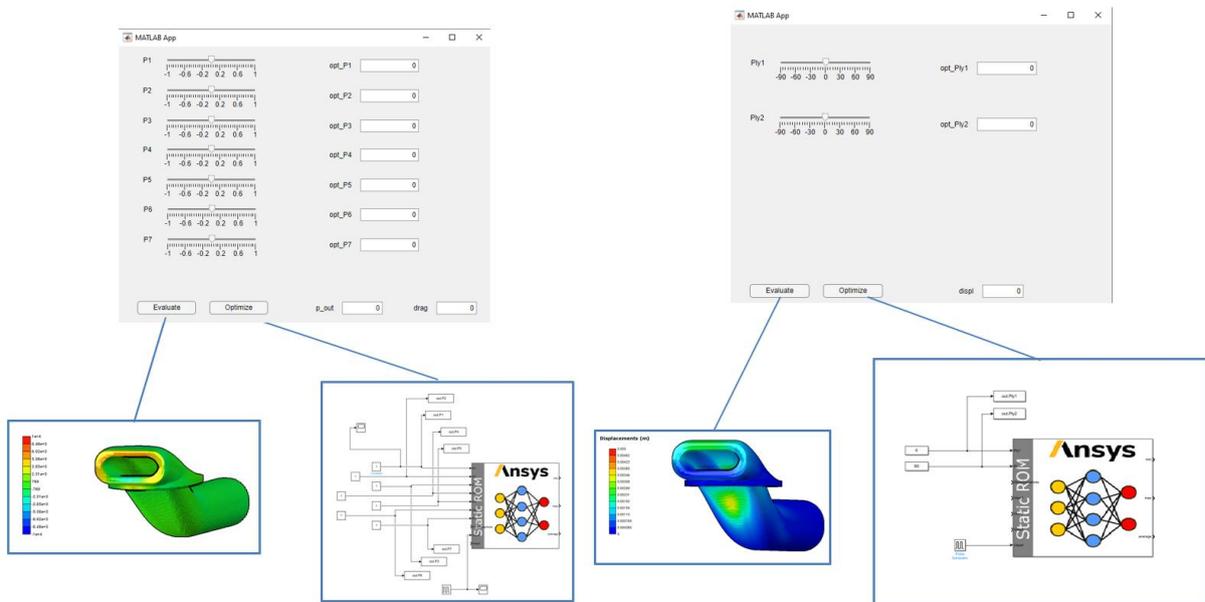


Figure 163: FEM (left) and CFD (right) dashboard

This study focused on the development of ROMs from different physical domains. Specifically, two domains were considered: CFD and FEM. The idea is to leverage ROMs and the FMU standard to create a unified design environment where complex analysis results can be visualized in real time. The methodology developed has

proven to be highly robust and effective for a wide range of problems. The results are very promising in terms of ROM accuracy and workflow flexibility.

6.4. CUSP

This work is partially published in (116). The CUBesat Solar Polarimeter (CUSP) project is a CubeSat mission orbiting the Earth aimed to measure the linear polarization of solar flares in the hard X-ray band by means of a Compton scattering polarimeter. CUSP is a project in the framework of the Alcor Program of the Italian Space Agency aimed to develop new CubeSat missions. In this study, the use of ROMs has been explored to create virtual sensors for measuring temperature in locations where physical sensors cannot be installed. Additionally, by coupling thermal and structural ROMs, it is possible to monitor not only the temperature but also the stress state or associated deformations. A prototype has also been developed to conduct tests. The objective is to create a digital twin and use the ROMs to monitor temperature and other relevant quantities during the mission.

6.4.1. Payload Design

The mechanical design, from the inception of Phase A of the mission in figure 164 (left), was already represented in detail since the sensors and sensitive elements of the payload had already been identified. After completion of Phase A some design improvements are under analysis while waiting for the start of Phase B (Figure 164 – right). Redout sensors (photomultiplier tubes and Avalanche photodiode) were already selected, thus this analysis coincided with an in-depth examination of the individual mechanical components, the interfaces among these components, and their integration with the overall platform. The payload consist of:

- Two identical aluminium interfaces with the platform;
- An aluminium alloy top, bottom lid and 4x panels with a passive tungsten film clamped;
- The collimator tray that contain 4 collimators and filters for scatterer and 4 collimators and filters for absorber;
- 32x absorber made from GAGG scintillator hosted on APD board;
- In the same mechanical and electrical framework of the scintillator tray are located 64x scatterer made from plastic scintillator and leaned on the Detector frame;
- A set of 4x ribs contributing to the mechanical connection of the parts.



Figure 164: Left phase A, right phase B

6.4.2. Multi-physic Model

The preliminary multi-physics simulation work was conducted using numerical analysis with ANSYS software, specifically ANSYS Workbench. This involved creating a simplified model that maintains the overall envelope and mass of the system while reducing geometric complexity and non essential details. A simplified model is crucial at this stage to ensure reasonable computation times without sacrificing simulation accuracy. The model includes all major structural and thermal characteristics, ensuring reliable analysis of multi-physical loads. A key aspect of this work was setting up a fine mesh, figure 165, a detailed mesh is essential for accurately capturing physical phenomena such as mechanical deformations and temperature distributions. The fine mesh provides high resolution numerical solutions, ensuring precise and reliable results. The mesh setup involved selecting the

element type, distributing mesh density in critical regions, and optimizing mesh quality to avoid distortions. The combination of a simplified yet accurate model and a fine mesh enabled effective preliminary analysis, providing a solid foundation for further detailed studies and design improvements.



Figure 165: FEM mesh

A specific material is attributed to each component of the model setting its mechanical properties. Each element that makes up the module is subject to different physical and thermal loads and at the same time will have its own function within the experiment. In table 44 the database of materials used for this study is reported.

Several numerical analyses were conducted to ensure the structural integrity and performance of the payload under various conditions. These analyses included:

- Quasi-static analysis with loads of 15g per axis to simulate extreme mechanical stress scenarios.
- Random vibration analysis using an interpolated curve derived from both European (ECSS) and American (GEVS) standards, ensuring compliance with relevant guidelines and launch stresses.
- Thermoelastic analysis incorporating thermal loads obtained from orbital simulations performed using Thermica, to evaluate the payload’s response to thermal stresses in orbit.
- Modal analysis to verify that the first resonance mode did not exceed 120 Hz, a critical mission requirement (100 Hz plus margin).

These comprehensive analyses provided a robust understanding of the payload’s behaviour under the expected operational conditions.

Table 44: Material properties

Material	Density [kg/m ³]	Young’s Modulus(GPa)	PoissonRatio	Thermal Expansion Coefficient [10 ⁶ /K]	Yield Strength [MPa]	Ultimate Strength [MPa]
<i>Aluminium 6082</i>	<i>2700</i>	<i>70</i>	<i>0.33</i>	<i>23.5</i>	<i>260</i>	<i>310</i>
<i>Aluminium 7075</i>	<i>2810</i>	<i>71.7</i>	<i>0.33</i>	<i>23.6</i>	<i>503</i>	<i>572</i>
<i>Ti6Al4V</i>	<i>4430</i>	<i>113.8</i>	<i>0.34</i>	<i>8.6</i>	<i>880</i>	<i>950</i>
<i>3D Printed PEEK</i>	<i>1320</i>	<i>3.6</i>	<i>0.36</i>	<i>47</i>	<i>90</i>	<i>100</i>
<i>Tungsten</i>	<i>19300</i>	<i>400</i>	<i>0.28</i>	<i>4.5</i>	<i>550</i>	<i>750</i>
<i>Scatterer</i>	<i>1023</i>	<i>1.86</i>	<i>0.41</i>	<i>80</i>	<i>28</i>	<i>29.2</i>
<i>Scintillator GAGG</i>	<i>6630</i>	<i>1.86</i>	<i>0.4</i>	<i>75</i>	<i>28</i>	<i>29.2</i>

6.4.3. Thermo-structural Analysis

In this work, the focus is not on the design and optimization of the component itself but on developing a ROM that provides reliable temperature data at each point of the cubesat. For each studied configuration, the ROM also offers reference values for stress and deformation to monitor the cubesat’s structural health throughout the mission, minimizing the need for a large number of sensors. In particular, two worst-case scenarios have been

identified: “hot” and “cold.” Depending on its position in orbit, the cubesat can be directly exposed to solar radiation or not.

Figure 166 summarizes the boundary conditions for the thermo-elastic analysis. Specifically, the connection areas with the external support are fixed, while temperatures are imposed at the top and bottom interfaces (A and B in Figure 166), on the connecting plates (C), and within the internal area of the cubesat where electronic components are located. The temperatures at these interfaces are listed in Table 45 for both the hot and cold cases. These temperatures were set considering the cubesat’s orientation during its orbit. For example, since Face A is more exposed than Face B, it experiences a much larger temperature variation, while Block D was designed to operate within a temperature range of 20-35°C.

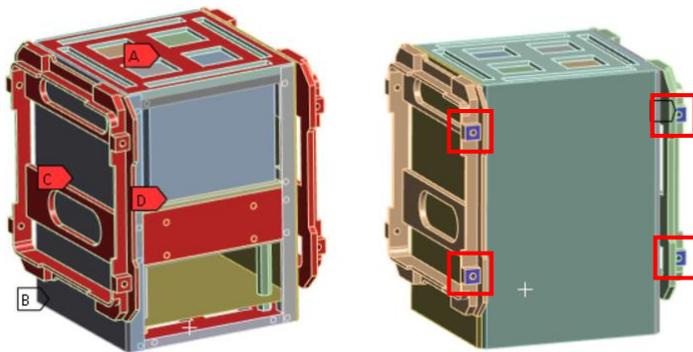


Figure 166: Temperature interfaces (left) and fixed surfaces (right)

Table 45: Temperature for hot-case and cold-case

	Temp D	Temp B	Temp A	Temp C
Hot-case	20	0	-20	10
Cold-case	35	20	50	20

6.4.4. Temperature ROM

With the temperature ranges defined in Table 45, 62 snapshots were generated using Latin Hypercube Sampling. The objective is to create a single ROM applicable both to cases when the cubesat is exposed to solar radiation and when it is not. Half of the snapshots are used for training, with the remaining 50% allocated to the test set. Ansys Twin Builder was employed to develop the ROM. Four modes were selected in the PCA, and the GARSM algorithm was used to interpolate the modal weights based on the input parameters. The error reported by the software is below 1%.

In the following figures, comparisons are shown between the temperatures predicted by the model and those calculated by FEM analysis for the hot case, the cold case, and the dataset point with the maximum error. A good correlation can be observed.

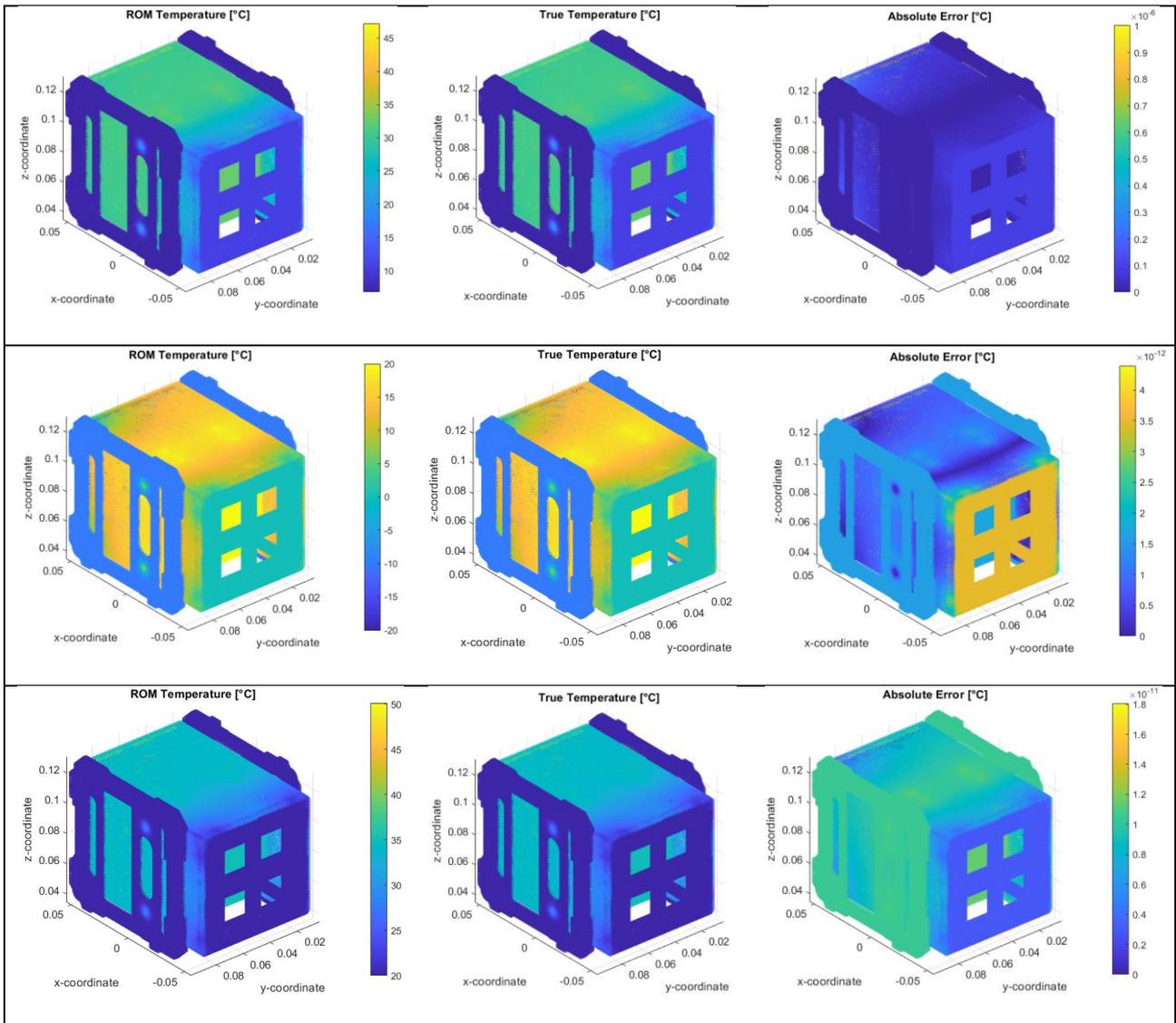


Figure 167: Comparison of ROM prediction and FEM temperature for maximum error DP (above), cold-case (centre) and hot-case (below)

6.4.5. Stress and displacements ROM

The same DOE used for the temperature ROM was also applied to develop ROMs for displacements and von Mises stress. The goal is not only to use the temperature ROM as a “virtual sensor” to provide temperature data at any point on the cubesat for operational monitoring but also to obtain additional information that enables more efficient monitoring. For this purpose, displacement and stress ROMs were also created. As with the temperature ROM, 50% of the snapshots are used for training and the remaining 50% for the test set. In both cases, nine modes were selected, and the software-reported error remains below 1%.

The following images present a comparison between the results from the ROM and those from FEM analysis.

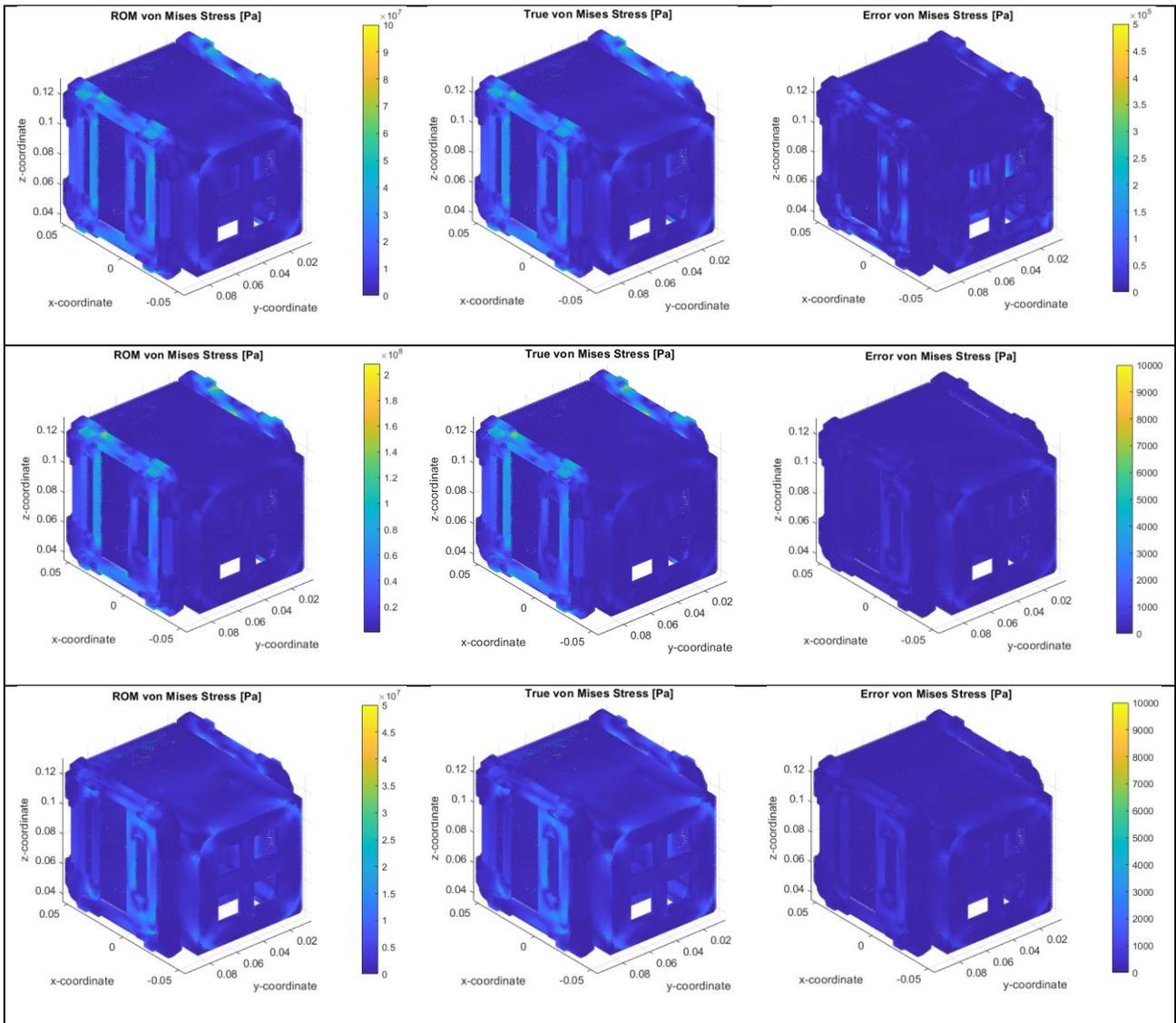


Figure 168: Comparison of ROM prediction and FEM von Mises stress for maximum error DP (above), cold-case (centre) and hot-case (below)

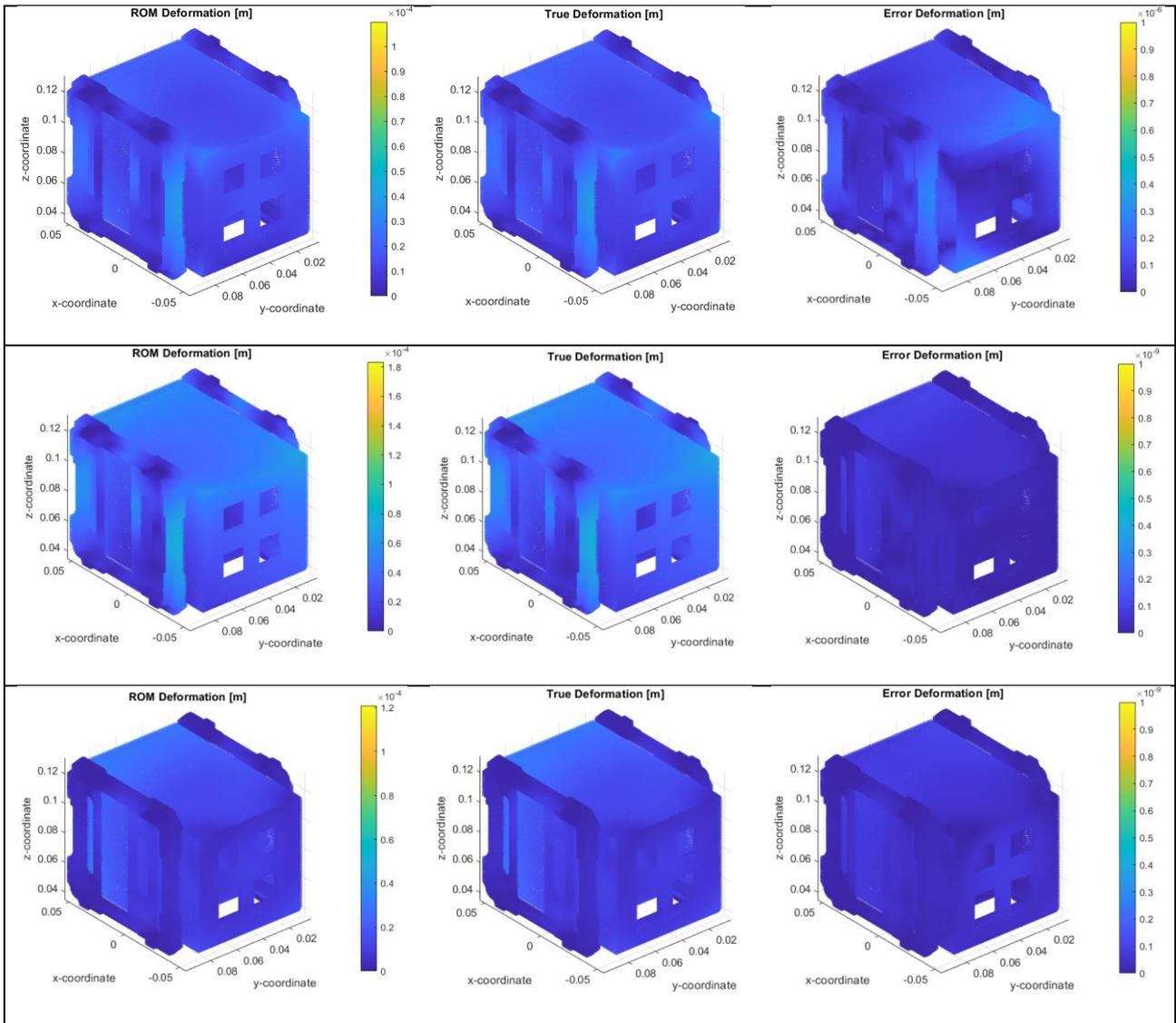


Figure 169: Comparison of ROM prediction and FEM deformation for maximum error DP (above), cold-case (center) and hot-case (below)

6.4.6. Physical demonstrator

Following the completion of these analyses, a technological demonstrator was developed and manufactured. A demonstrator like this, in figure 170-left incorporating the optimized components, will be tested in the subsequent phases of the mission to validate its performance under real-world conditions. Some parts were 3D printed, see figure 170-right, leveraging the precision and flexibility of additive manufacturing to achieve the desired specifications.

The idea is to connect the physical object with the ROM to create a fully functional digital twin. By using sensors to monitor temperature at the interfaces and critical points, a successful comparison would enable a reduction in the number of required sensors, saving weight and, crucially, allowing for precise monitoring of sensitive internal areas where installing sensors during operation may be nearly impossible. While temperature monitoring is straightforward at the interfaces, placing sensors inside the cubesat is much more challenging. Furthermore, with the ROMs interconnected, it is possible to gain additional insights (such as stress and deformation states), thereby

facilitating easier monitoring of the cubesat's behaviour or enabling mission simulations to prevent potential operational failures.



Figure 170: Technological demonstrator

6.4.7. Conclusions

The objective of this study was to develop ROMs from thermo-elastic multi-physics analyses. In this way, it is possible to build a virtual sensor that provides temperature, stress, and deformation data for any point of CUSP. This allows for more efficient monitoring of the CubeSat's performance, saving weight and the number of physical sensors, and, most importantly, monitoring even the most critical areas where it is difficult to install sensors. The ROMs have been developed into a very robust, flexible, and accurate tool. The physical demonstrator will validate the models and develop a routine to leverage the ROMs as digital sensors, thus creating a true digital twin.

6.5. Open Parametric Aircraft Model (OPAM)

This work is partially published in (117) where a workflow for creating advanced aerodynamics design dashboards is proposed. A CAD modeler is directly linked to CFD simulation results so that the designer can explore in real time, assisted by VR, how shape parameters affect the aerodynamics and choose the optimal combination to optimize performance. In this way, the time required for the conception of a new component can be drastically reduced because, even at the preliminary stage, the designer has all the necessary information to make more thoughtful choices. The OPAM (Open Parametric Aircraft Model), a simplified model of the Boeing 787, was considered as a case study. CAD parameterization and mesh morphing were combined to generate the DPs, while ROM were developed to link the results of CFD analyses to the chosen parameterization. The ROMs were exported as FMUs to be easily managed in any environment. Finally, a VR design dashboard was created in Unity environment enabling the interaction with the geometric model to observe in a fully immersive and intuitive environment how each shape parameter affects the physics involved. The MetaQuest 3 headset has been selected for these tests.

6.5.1. OPAM Project

The GMGW (Geometry and Mesh Generation Workshops) aimed to identify suitable improvements and best practices for generating more accurate and reliable meshes. For the second GMGW meeting (118), a parametric aircraft model inspired by the Boeing 787-800 Dreamliner, a twin-engine civilian aircraft for medium to long-haul routes, was provided (119). The simplified model (Figure 171) is based on 53 parameters and scaled approximately 1:300 relative to the B787. The entire aircraft body is composed of spheres, cylinders, cones, and ruled surfaces. Cross-sections for the ruled surfaces are either NACA 4-series aerofoils or super-ellipses. Full details on the generation of wings, fuselage, and other parts of the aircraft can be found in (119). As a CAD editor, ESP (Engineering Sketch Pad), an open-source editor developed by the Massachusetts Institute of Technology, was used (120).

The workshop challenge involved starting with a parametric CAD model and developing an automatic meshing procedure that maintains mesh quality for each design parameter (DP). In this study, a different perspective was taken to address the challenge. Instead of generating a new mesh for each DP, the CAD modifications are

transferred to the existing mesh, deforming the baseline mesh accordingly. This innovative and creative approach offers several computational advantages. Notably, it makes the workflow well-suited for generating ROM, where it is crucial to keep the mesh topology unchanged, and provides a significant step forward compared with the mesh morphing based approach presented at GMGW (19) adding the full automation of the process according to the CAD parametrization.

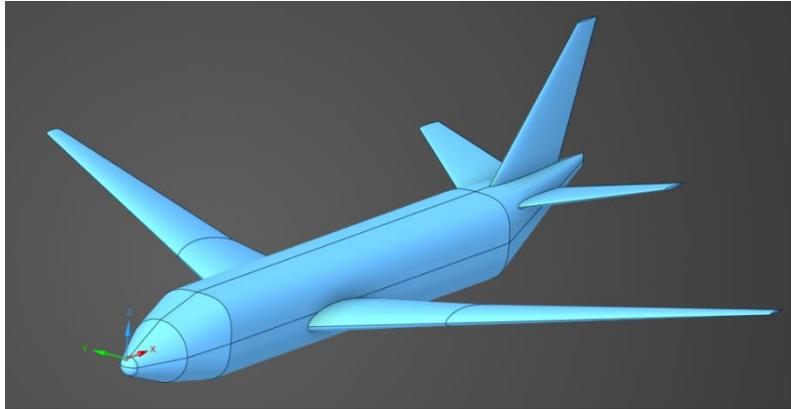


Figure 171: OPAM baseline

The focus of the OPAM-1 challenge was automatic mesh creation. In fact, the ESP tool is fully scriptable, so once the parametric model is defined each CAD variant can be generated with a Python script. Having also a procedure for automatic mesh creation makes the workflow very attractive, greatly reducing the time to prepare the CFD analysis. Several studies (121) (122) (123) have been conducted in this regard demonstrating the possibility of automating the mesh creation process. However, some critical issues remain. The processes used are often not fully automated and always require a preliminary step in which the parts that form the aircraft are described and grouped and the reference dimensions for the problem are set. In addition, the mesh creation phase remains a fairly computationally expensive phase. Finally, even automating the mesh creation process, it remains very difficult to ensure that the mesh is iso-topological (a key aspect of creating reduced models).

Another approach is mesh morphing, which involves using the same mesh while defining parameters at the mesh level. Notably, this approach was also used for the OPAM-1 testcase (124), emphasizing the flexibility of this approach that can also be applied to important geometric variations. The mesh used for this testcase and the variations considered are shown in the figures below.

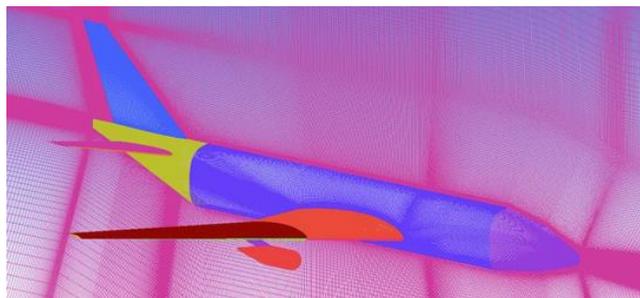


Figure 172: Structured mesh used in (124)

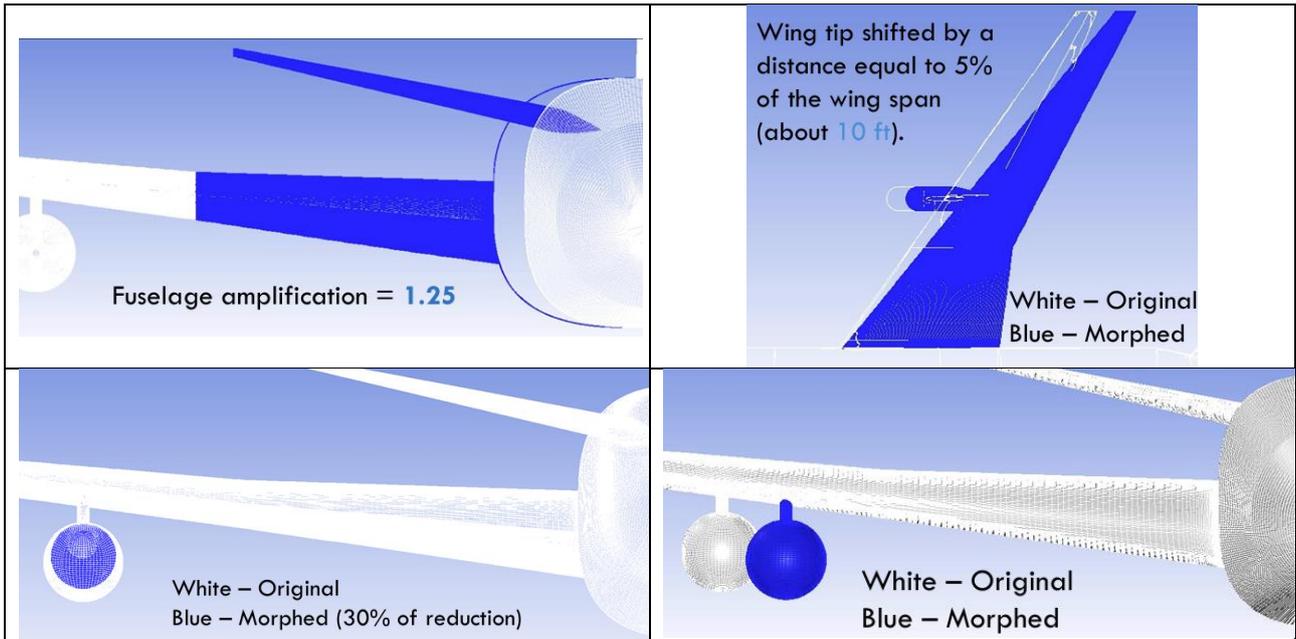


Figure 173: Shape parameters defined using mesh morphing in (124)

The main advantages of this approach are related to the fact that the same computational grid is always used, so there is no re-meshing noise, the mesh topology does not change (crucial for ROM generation), the convergence of the CFD analysis is faster, and in general the mesh update takes less time. In particular, comparing the time required for the two studies (124) (123), even considering the different facility used, mesh morphing is very advantageous. Conversely, the main limitation is related to the set-up. In fact, to obtain these shape variations complex set-ups are required, there is often a limit on the maximum amplifications to ensure a good mesh quality, and the shape variations must be iso-topological. In addition, in some cases it is difficult to define the displacements of source points when there is a more complex shape change and the fundamental operations (rigid displacements, scaling, offsets, etc...) fail to capture with sufficient accuracy (specifically STL corrections are required to preserve the geometry).

6.5.2. Hybrid Workflow

The innovative aspect of this study lies in the parameterization and the algorithm used to create each DP in the Design Of Experiment (DOE). A hybrid method was employed (125) (126), combining the advantages of mesh morphing and parametric CAD. The workflow is reported in the section 2.3.1.3. Specifically, the parameters are defined in a Python script that generates the CAD. Each CAD variant is then compared with the baseline, creating a cloud of points that spans from the initial state to the new one. This point cloud is used as an RBF field to transfer information from the CAD level to the mesh level. Finally, the mesh is deformed, and a CFD simulation is conducted for each DP. Figure 174 illustrates the workflow followed in this study.



Figure 174: Hybrid workflow

The proposed method has the advantages of both previously described methods. In fact, CAD-based parameterization is preserved, which provides excellent control over the final shape, but mesh morphing is used to update the baseline mesh onto the desired variant. This hybrid workflow, in addition to being a very attractive alternative to more established approaches, makes it possible to create ROMs from a parametric CAD model, thus allowing CFD information to be transferred to CAD editors to create smarter design platforms and reduce the time required for design. It is emphasized that the proposed workflow is an innovative and computationally efficient

solution to the problem of automatic mesh generation. Instead of creating a new mesh, the pre-existing mesh is updated with CAD-based modifications.

Once the dataset was created, the procedure described in Section 2.4.4 was used to create deployment and design dashboards in MATLAB and Unity environments. Specifically, the FMU standard was used to extract the ROM. The FMU standard can be imported into major programming environments as a black box, simplifying the creation of dashboards as described in the following sections.

6.5.3. Parametric CAD

For the generation of parametric CAD, the already mentioned open-source CAD editor ESP was employed (120). The workflow demonstrates remarkable flexibility and is easily customizable for integration with various CAD editors. Trials have been conducted using diverse tools, including CadQuery (127) (128), JPAD (129), AxCent (130) and Rhino (131).

ESP is a browser-based design software designed for creating, modifying, and generating three-dimensional solid models. It uses OpenCascade (132) as the engine for geometry generation and reads input ASCII files containing parameters, invoking the necessary functions to generate the CAD. In this case, scripting the modification of these ASCII files is needed. ESP has a graphical interface but can also be run in batch mode.

6.5.4. Design example according to the proposed workflow

This paragraph presents the complete workflow applied to the case of OPAM-1. Six parameters affecting the wings were selected, and the hybrid workflow was used to generate each DP. It is highlighted that the entire workflow, is fully automated, allowing for the generation of DPs ready to be run on any CFD solver automatically. Therefore, it can be seen as a more computationally efficient alternative to traditional automatic mesh generation tools. The described workflow was used to generate the dataset. For each DP, the surface node coordinates and the surface pressure values, drag, lift, and efficiency values were saved. The node coordinates of each DP's surface mesh and the pressure values were used to create ROMs, while scalar values were interpolated to obtain a response surface. In the final dashboard, these datasets can be utilized across multiple levels. Specifically, the response surfaces can be used to identify the optimal solution. Within the best identified DP, field quantities (specifically pressure fields) can be examined to gain deeper insights into the physics involved. Real-time exploration around the selected DP allows for the discovery of potentially more interesting solutions. Additionally, extracting pressure coefficient profiles at any section provides confirmation of the findings.

6.5.4.1. High fidelity CFD model of OPAM

As mentioned in previous paragraphs, the OPAM case study was chosen because it represents a sufficiently interesting testcase, especially due to its comprehensive parametrization with 53 parameters describing every part of the aircraft.

The first step was to define the baseline and create the CFD mesh of the initial geometry. Table 46 outlines the main characteristics of the mesh, and Figure 175 provides some details of the mosaic mesh. Symmetry was leveraged to reduce the computational domain.

Table 46: Details of CFD Mesh

Number of faces	<i>4979888</i>
Number of cells	<i>957205</i>
Number of nodes	<i>3366691</i>
Min. Orthogonal Quality	<i>1.50172e-01</i>
Max. Aspect Ratio	<i>1.38865e+02</i>
y+	<i><10</i>

The following main options are configured:

- Steady-state simulation;
- Density-based solver;
- k-omega SST turbulence model;

- Air as an ideal gas with the Sutherland viscosity law;
- Boundary conditions:
 - Inlet [pressure-far-field]: Mach equal to 0.7 inclined by $\alpha = 0^\circ$
 - Outlet [pressure-outlet]: Pressure and temperature standard (101325 Pa, 298 K)
 - Side [pressure-far-field]: Same conditions as the Inlet.
 - Symmetry [symmetry]: Symmetry plane.
 - Plane [wall].

The implicit Roe-FDS formulation with second-order discretization was employed.

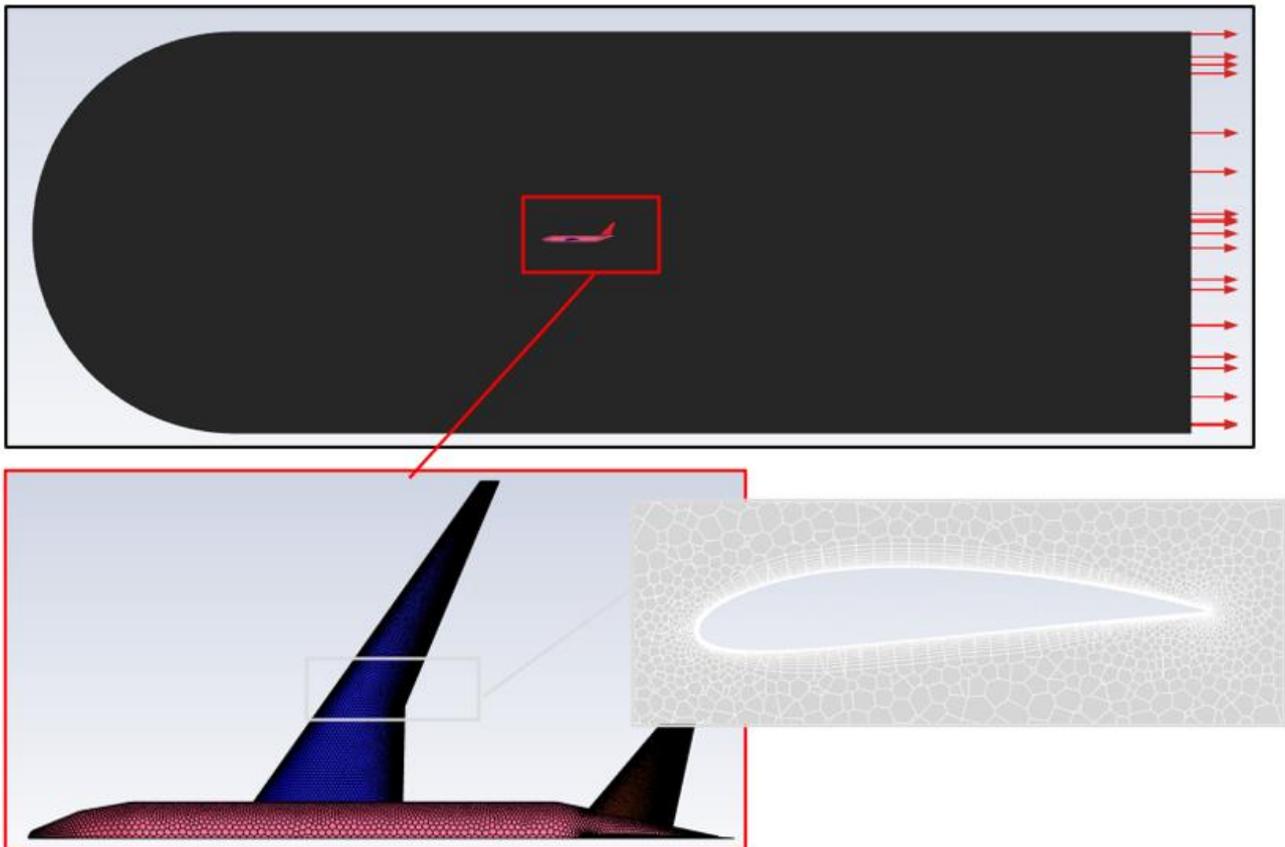


Figure 175: CFD Mesh

6.5.4.2. Parameters and DOE

Out of the 53 model parameters, 6 were chosen to modify the wing shape, while maintaining constant wing area. Variations were made in the aspect ratio, leading edge sweep, setting angle, and camber ratio at the break and tip. A DOE consisting of 66 DPs was generated using the LHS. Specifically, DP 65 and 66 are highlighted as they represent the extreme values of the parameters within their respective ranges when compared to the baseline configuration.

- DP65: 8, 33, -5, 0.02, -10, 0.02
- DP66: 10, 37, -1, 0.06, -6, 0.06

In this phase, it was verified that the mesh quality did not degrade. Specifically, DP 65 exhibited the lowest mesh quality. Below are some values of the minimum orthogonal quality for certain DPs, as reported in the table 48.

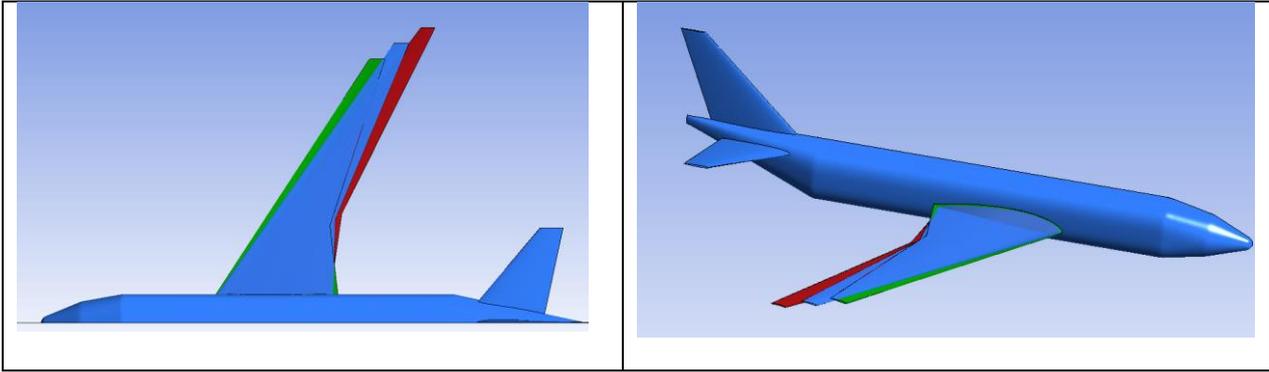


Figure 176: Blue,baseline, green DP 65, red DP 66

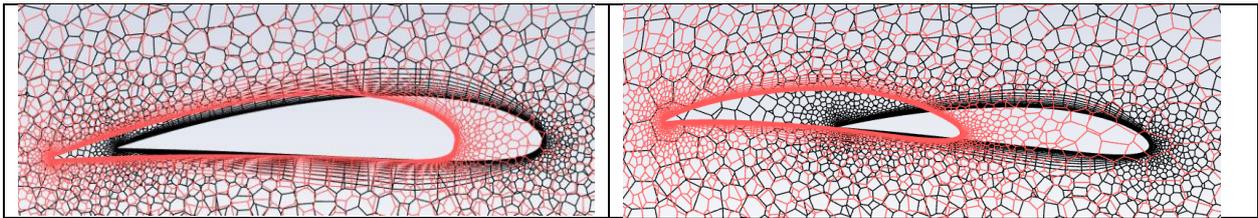


Figure 177: Design Point 66 with cord at 0.5 and 0.75 compared to the baseline (black line).

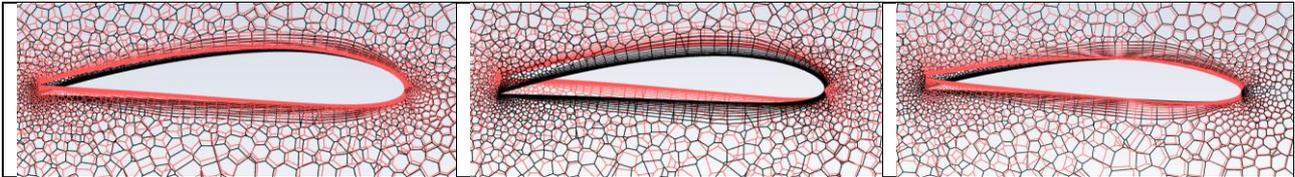


Figure 178: Alpha B effect: Alpha B = -5 with cord at 0.25,0.5,0.75 compared to the baseline (black line).

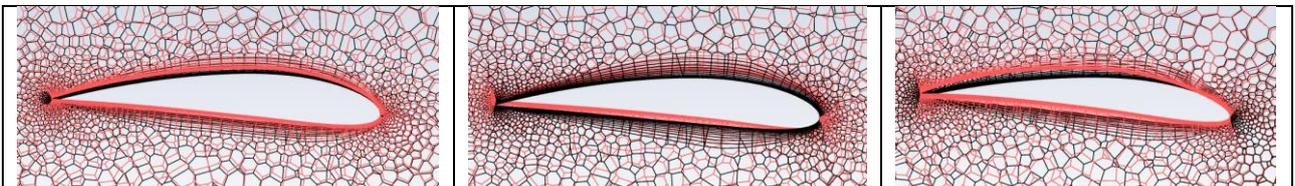


Figure 179: Camber B effect: Camber B = 0.06 with cord at 0.25,0.5,0.75 compared to the baseline (black line).

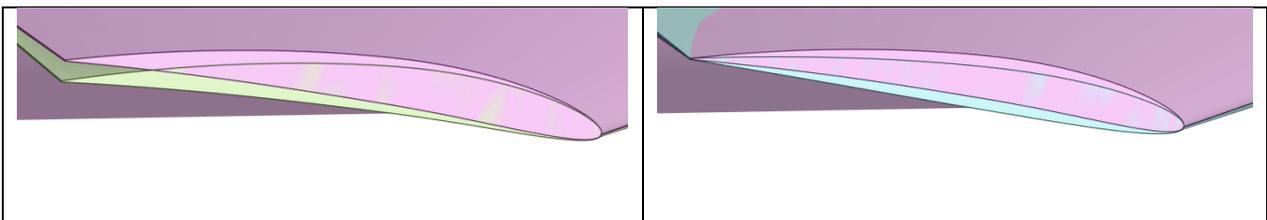


Figure 180: Effect of alpha_t (-6) and camber_t (0.02) at the tip compared to the baseline (pink).

Table 47: Range and baseline value for each parameter

	Aspect R	Sweep	Alpha B	Camber B	Alpha T	Camber T
Range	8 ÷ 10	33 ÷ 37	-5 ÷ -1	0.02 ÷ 0.06	-10 ÷ -6	0.02 ÷ 0.06
Baseline	9	35	-3	0.04	-8	0.04

Table 48: Mesh Quality

Min Orthogonal Quality	DP
1.45437e-01	baseline
8.80044e-02	1
9.94842e-02	10
9.51205e-02	20
1.02042e-01	30
1.02038e-01	40
9.15076e-02	50
9.12111e-02	60
5.18119e-02	65
9.16307e-02	66

Finally, it was verified that there is indeed a good correspondence between CAD and mesh for each DP. The figure below shows a comparison between mesh and CAD for DP 65 where the maximum deviation has been observed. It can be observed that the CAD and the surface mesh perfectly overlap. In Figure 182, the comparison between the surface mesh and the corresponding CAD is shown. For the reported DP, there is a deviation ranging from -0.09 mm to 0.07 mm. The green areas indicate points with deviation under the tolerance (0.01 mm).

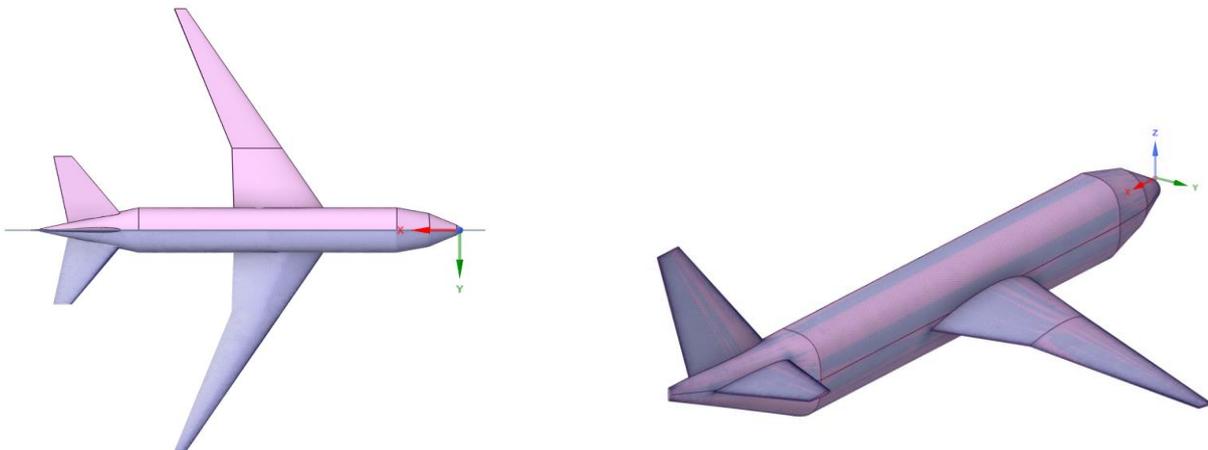


Figure 182: Comparison of CAD and mesh for DP 65

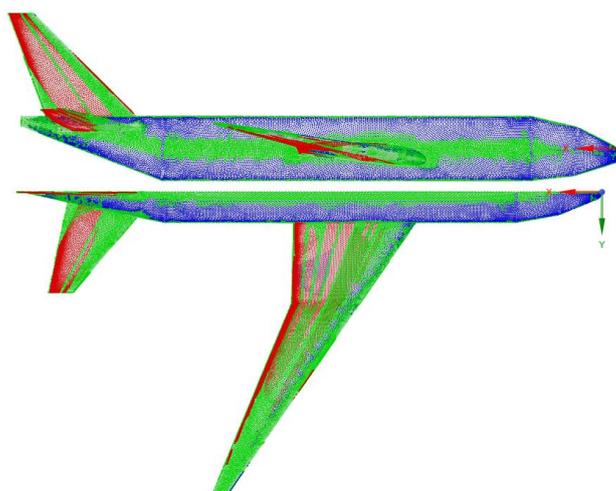


Figure 183: Deviation of the mesh from the deformed CAD for DP 65

6.5.4.3. Response surface

FMU are used to interpolate scalar quantities such as drag and lift. In the figures below the main results about precision of the RSs are reported.

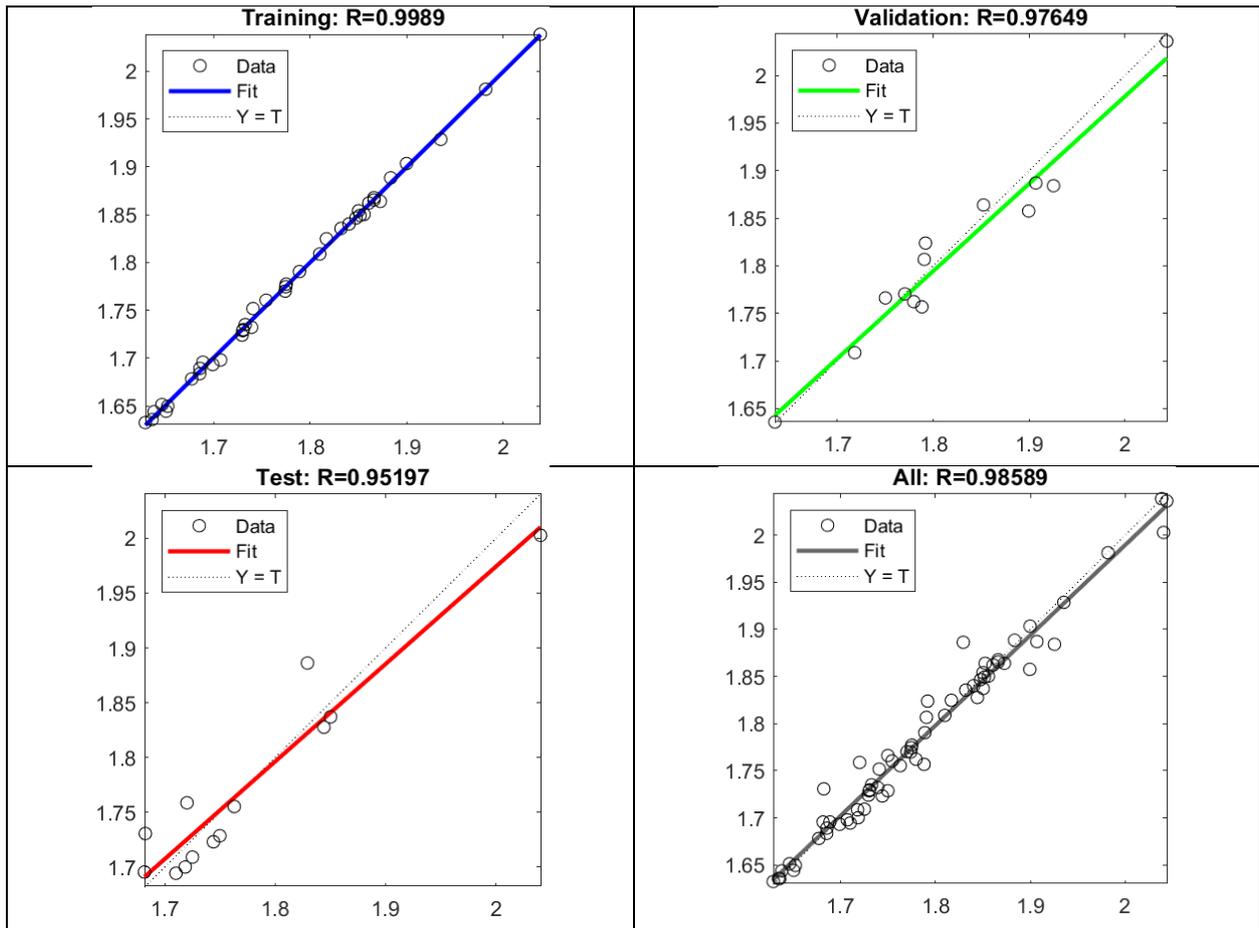


Figure 184: R value for drag evaluation

Table 49: performance of FMU for drag evaluation

	Observations	MSE	R
Train	40 (60%)	2.15E-05	0.9989
Validation	13 (20%)	5.37E-04	0.9765
Test	13 (20%)	8.36E-04	0.952

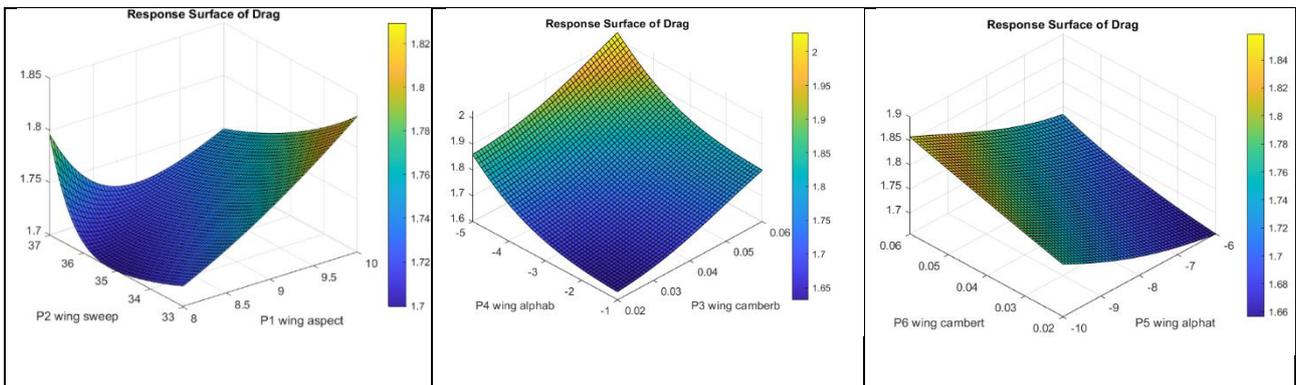


Figure 185: Response surfaces for drag evaluation

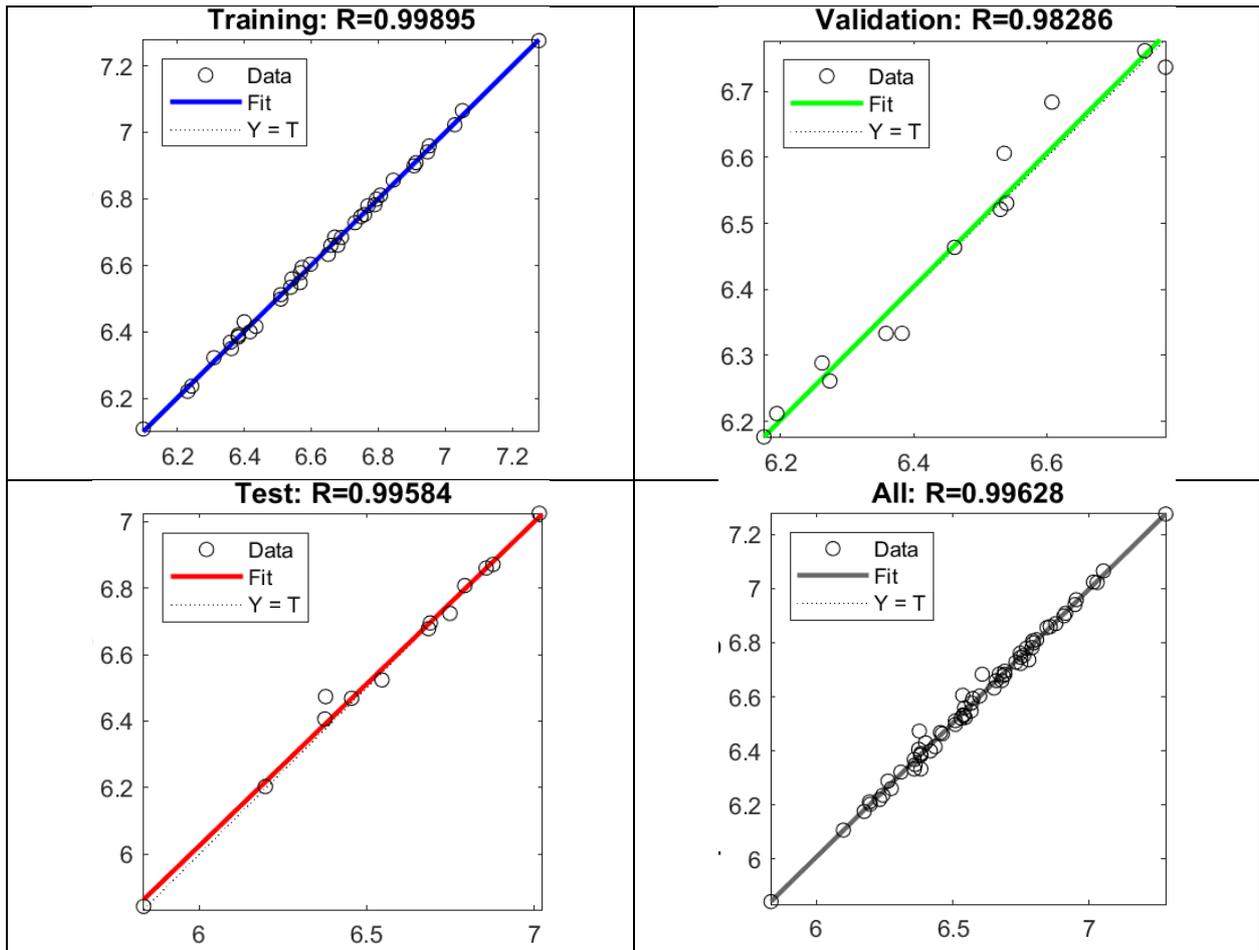


Figure 186: R value for lift evaluation

Table 50: performance of FMU for lift evaluation

	Observations	MSE	R
Train	40 (60%)	0.0001	0.9989
Validation	13 (20%)	0.0013	0.9829
Test	13 (20%)	0.0009	0.9958

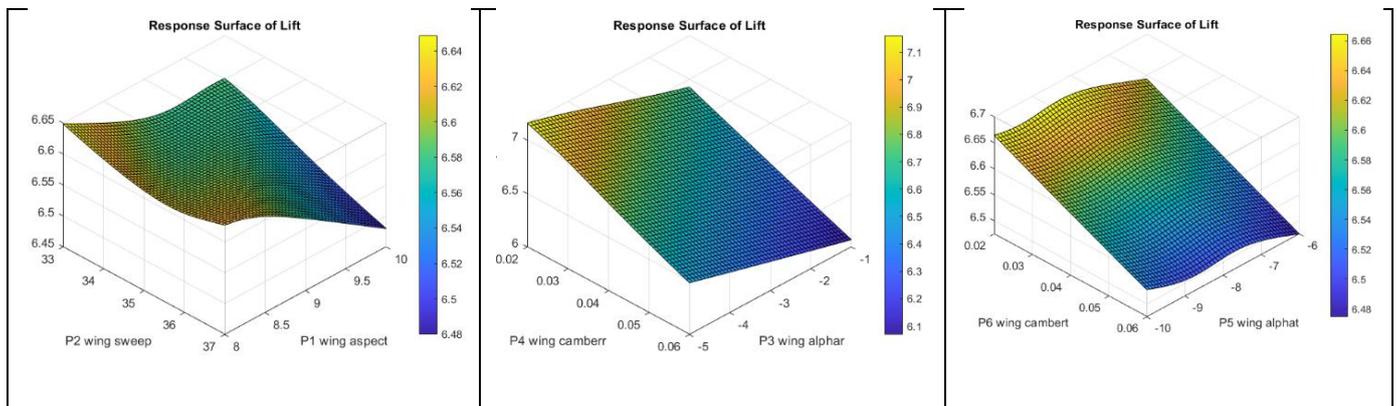


Figure 187: Response surfaces for lift evaluation

6.5.4.4. Mesh ROM

The ROM of the geometry allows real-time evaluation of node positions for any aircraft configuration, i.e., for any combination of selected parameters. Specifically, all nodes of the wing, the part of interest for monitoring, were selected. A model was trained to assess how each parameter affects the final geometry. Six modes were selected,

and 50% of the DPs were used for training. The error of the ROM evaluated by the software is less than 0.48%. In particular, the DP in the test set where the error is maximum (0.48%) is reported. The blue points represent those of the CFD mesh, while the red points are the corresponding points evaluated with the ROM. Excellent overlap is observed both globally and locally (profiles at 0.25, 0.5, and 0.75 chord positions are specifically shown). Figure 189 shows the percentual error defined as:

$$e_p = \left| \frac{x_{ref} - x}{x_{ref}} \right| \quad (6.10)$$

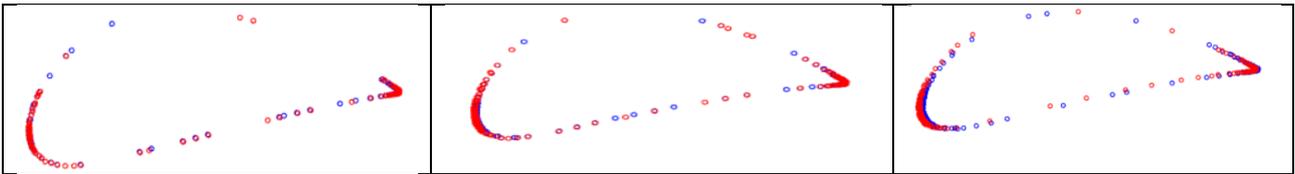
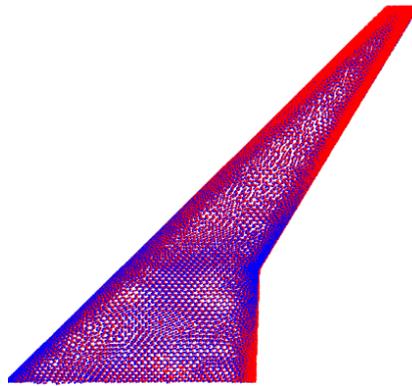


Figure 188: The blue points represent those of the CFD mesh, while the red points depict the corresponding points evaluated with the ROM for the profiles at 0.25, 0.5, and 0.75 chord positions.

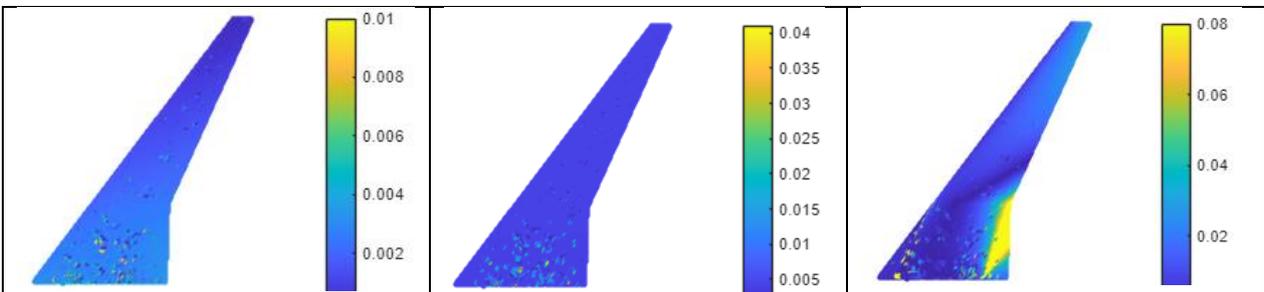


Figure 189: Relative error for coordinate x,y and z

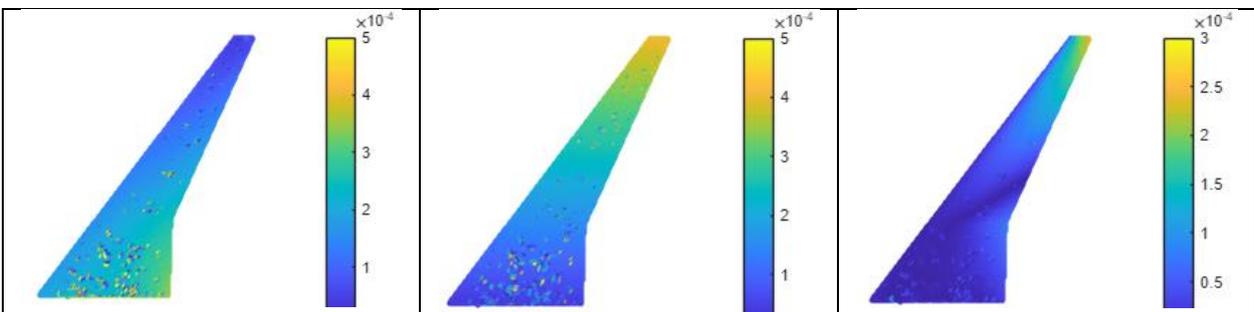


Figure 190: Absolute error [m] for coordinate x,y and z

6.5.4.5. Pressure ROM

The ROM of the pressure allows real-time evaluation of pressure values (or associated quantities like C_p) for any combination of selected parameters. All wing nodes, the part of interest for monitoring, were selected. A model was trained to assess the influence of each parameter on the final geometry. Five modes were selected (Figure 191), and 50% of the DPs were used for training. The error of the ROM evaluated by the software is less than

5.1%. Again, a comparison was made between the values predicted by the ROM and those computed by CFD for the DP in the test set with maximum error (5.1%). A good correspondence is observed.

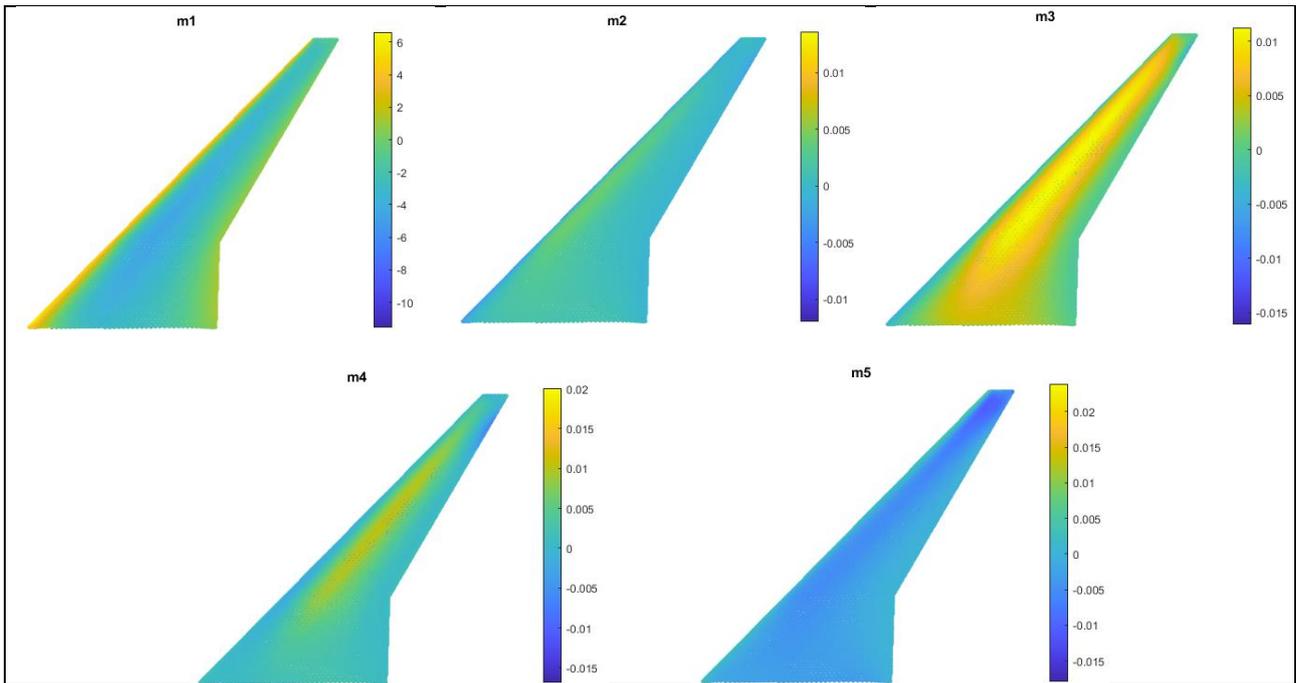


Figure 191: Pressure modes

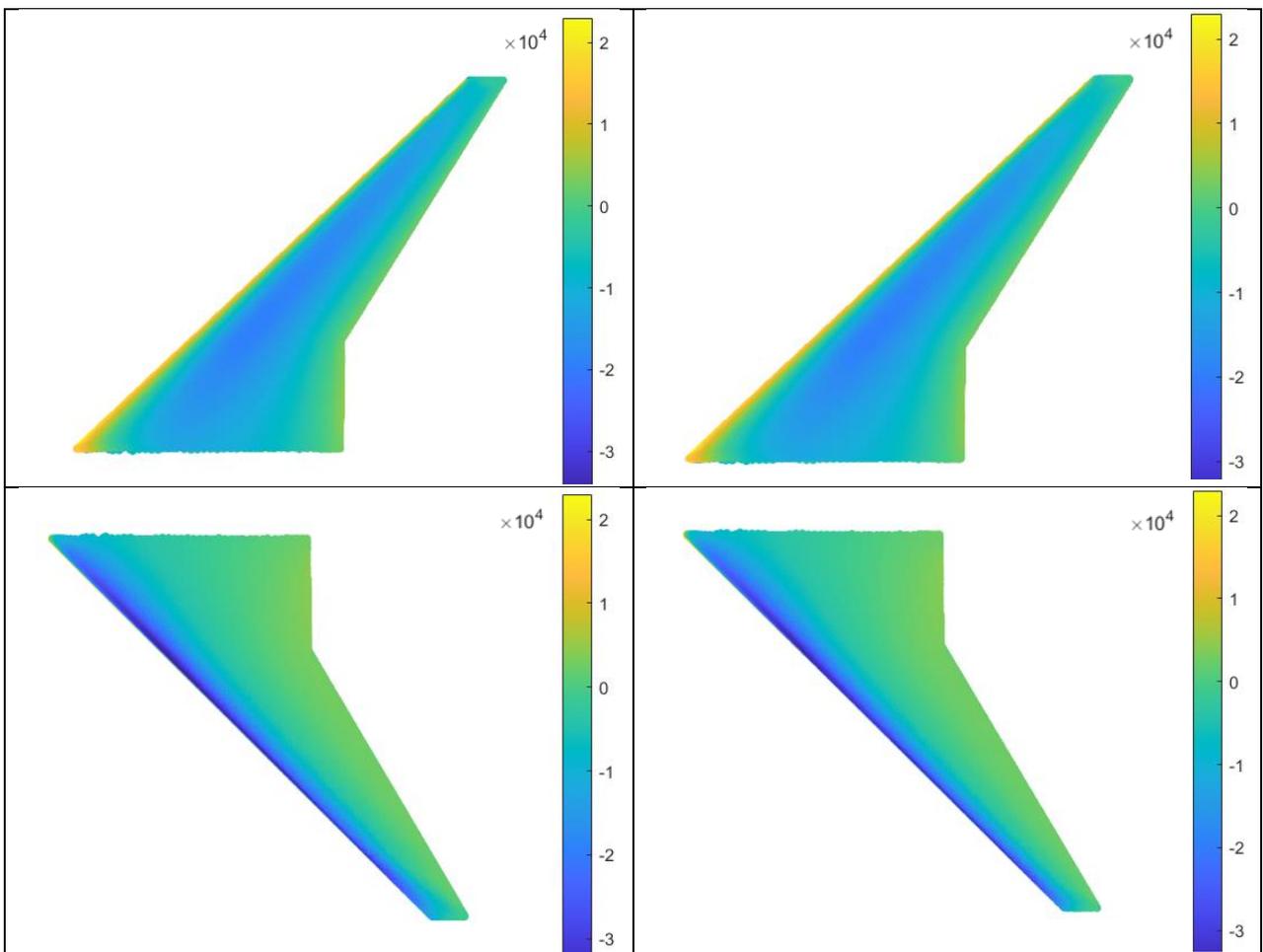


Figure 192: Comparison of pressure contours: CFD (left) vs ROM (right) for the DP with maximum error in the test set.

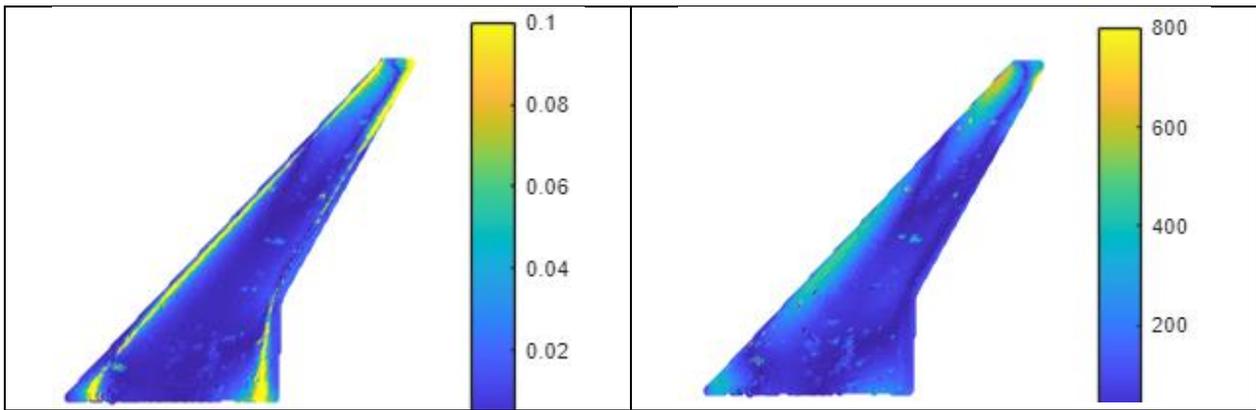


Figure 193: Percentual error (left) and absolute error [Pa] (right) for the DP in the test set with maximum error

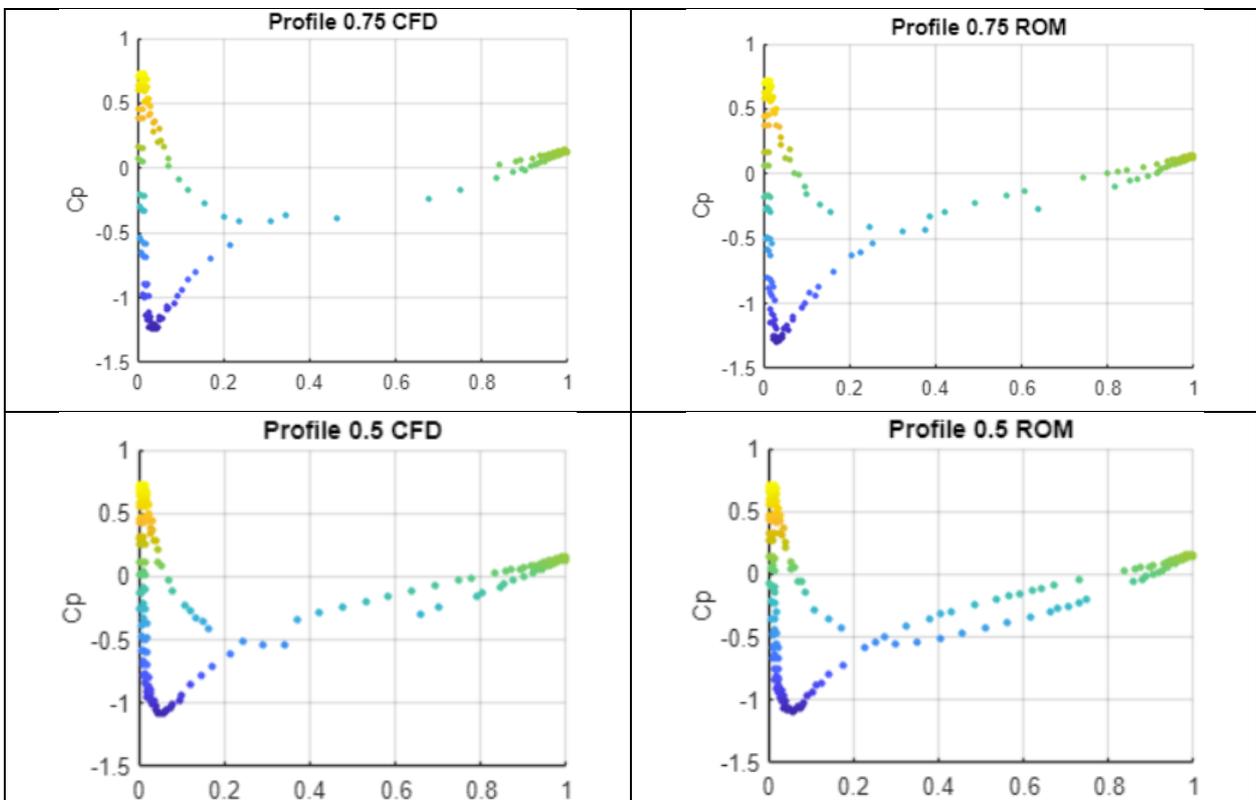


Figure 194: CP profile on the chord at 0.75 and 0.5, comparison between CFD and ROM for the DP in the test set with maximum error

6.5.4.6. Optimization Results

The collected data is managed by a MATLAB script that enables optimization at various levels. Scalar values can be used to guide optimization and identify a global optimum using response surfaces. Real-time parameter exploration allows observation of their influence on aircraft aerodynamics. This approach facilitates a more robust and informed optimization process because it provides a clear understanding of the physics involved and the contribution of each parameter. For example, it allows for discarding DPs that, despite improving efficiency, lead to greater flow instability.

Using this approach, an optimal point was identified that significantly improves performance. Specifically, efficiency is increased by approximately 10%. It is observed that the low-pressure zone on the rear part of the profile is less developed, and the separation zone at the wing's trailing edge is reduced.

The following figures show a comparison between the baseline and optimized geometries, as well as a comparison between CFD results and predictions from the ROM.

Table 51: Comparison baseline and optimized input parameters value

	Aspect r	Sweep	Alpha b	Camber b	Alpha t	Camber t
Baseline	9	35	-3	0.04	-8	0.04
Optimized	9.31	34.39	-2,93	0.025	-6.2	0.027

Table 52: Comparison Baseline and optimized out parameters value

	Cl	Cd	Eff
Baseline	0.505	0.134	3.77
Optimized	0.535 (+6%)	0.129 (-4%)	4.15 (+10%)

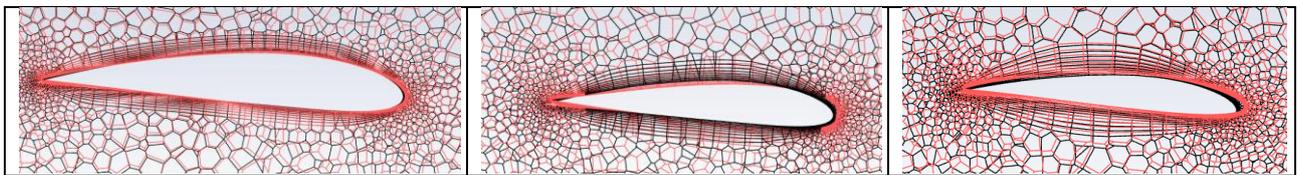
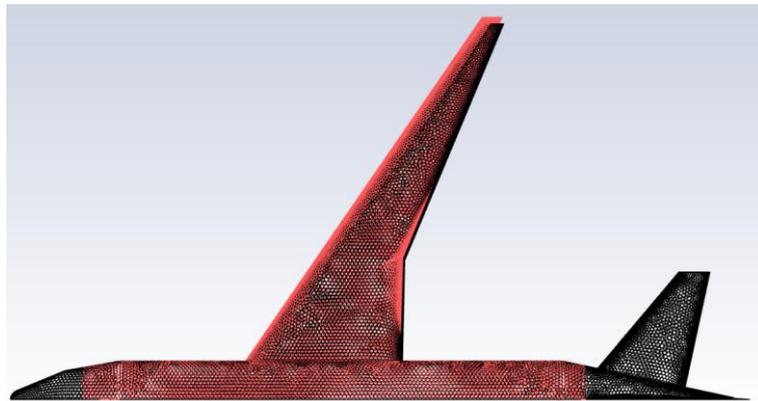


Figure 195: Comparison of baseline (black) and optimized (red) mesh

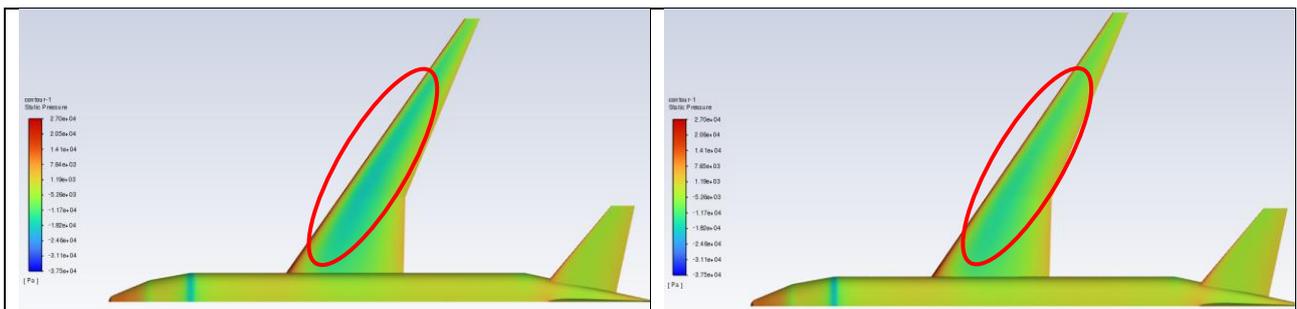


Figure 196: Comparison of pressure contours: baseline (left) and optimized (right).

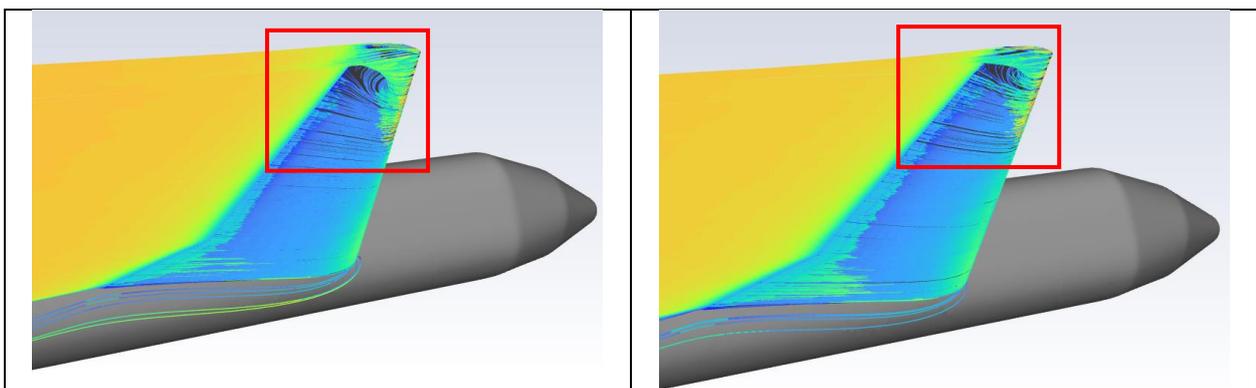


Figure 181: Comparison of velocity path-lines: baseline (left) and optimized (right).

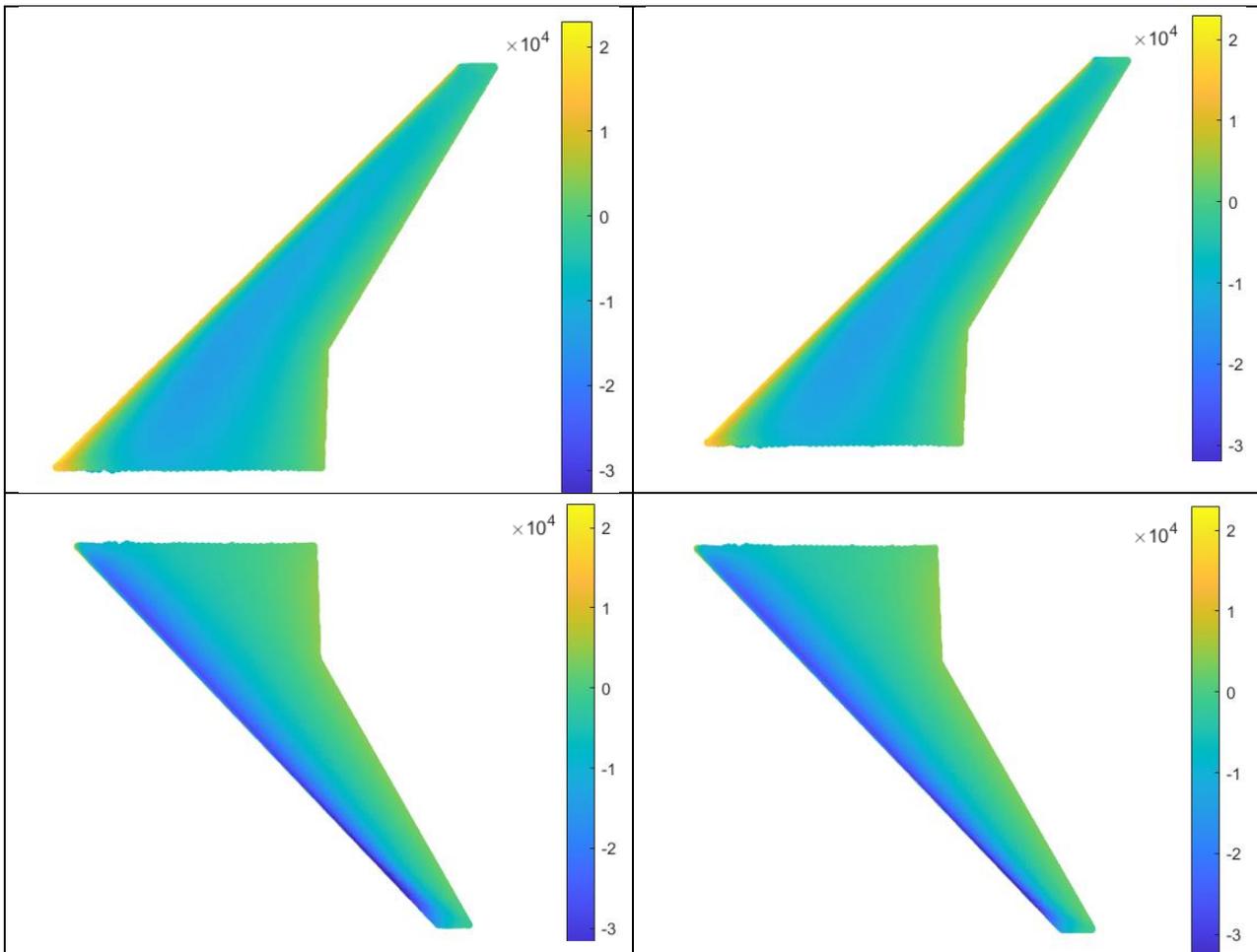


Figure 198: Comparison of CFD (left) and ROM (right) on the front and rear of the wing for the optimized point

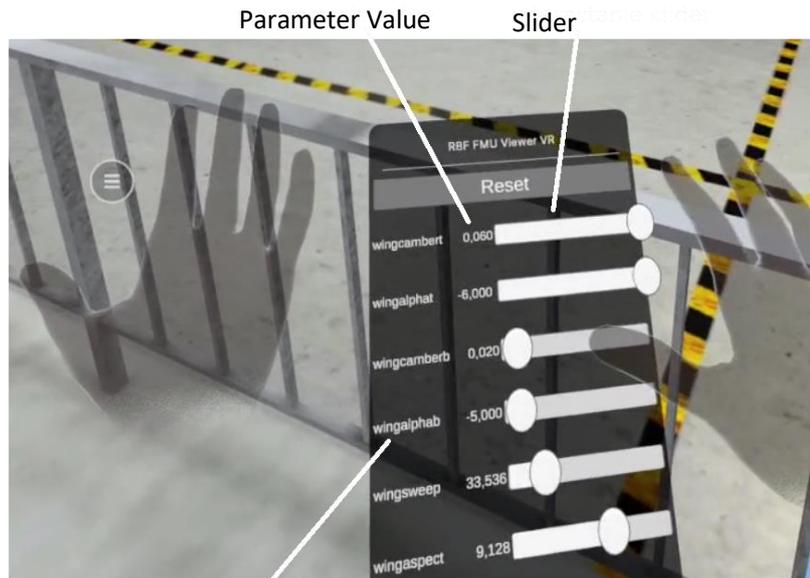
6.5.4.7. VR dashboard

The generated Reduced Order Models have been loaded into the interactive virtual reality dashboard. The Meta Quest 3 headset has been used to explore in real time how the CAD-based parameters affect the aerodynamics of the aircraft in a fully immersive environment.

As described in the previous sections, the data need to be prepared for import into the viewer. The modes for the geometry and for the pressure are loaded into the GPU memory at initialization. The FMUs are loaded into the CPU and allow for the updated weights of the modes (both geometry and results) to be obtained in almost real-time. The achieved visualization performance was stable at 60 frames per second, resulting in a very good immersive experience. The parameter menu is activated by making the left-hand palm face towards the camera. The users can interact with the model in two ways: using sliders (Figure 199) or with a smaller model of the aircraft (Figure 200). The sliders can be interacted with by using the index finger (Figure 199). Simply “grabbing” a slider with the index and thumb and dragging it updates the parameter value, which in turn updates the geometry and the pressure field.

It is also possible to interact with a smaller model, with handles positioned at the points where the parameters act, or with the sliders (Figure 199). The handles have a fixed range of movement and can be manipulated by dragging them using the index finger and thumb. Moving the handles within the predefined range updates both the smaller model and the larger model with the pressure field. The VR App has been developed formatting numbers according to the European decimal separator, i.e. a comma instead of a point.

The real-time results are displayed on the full model. An aircraft hangar setting (Figure 201) was used to make the experience even more realistic and immersive.



Parameter name

Figure 199: Slider interface

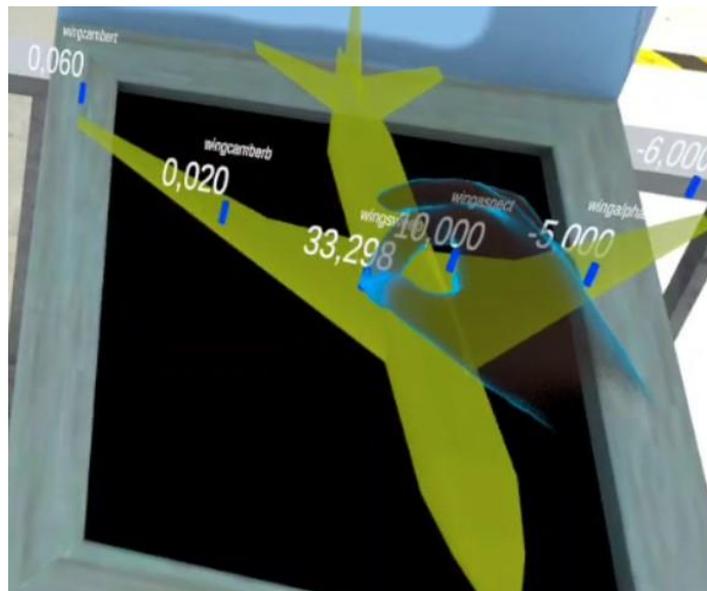


Figure 200: 3D interactive handles used to modify in real-time the value of the parameters.

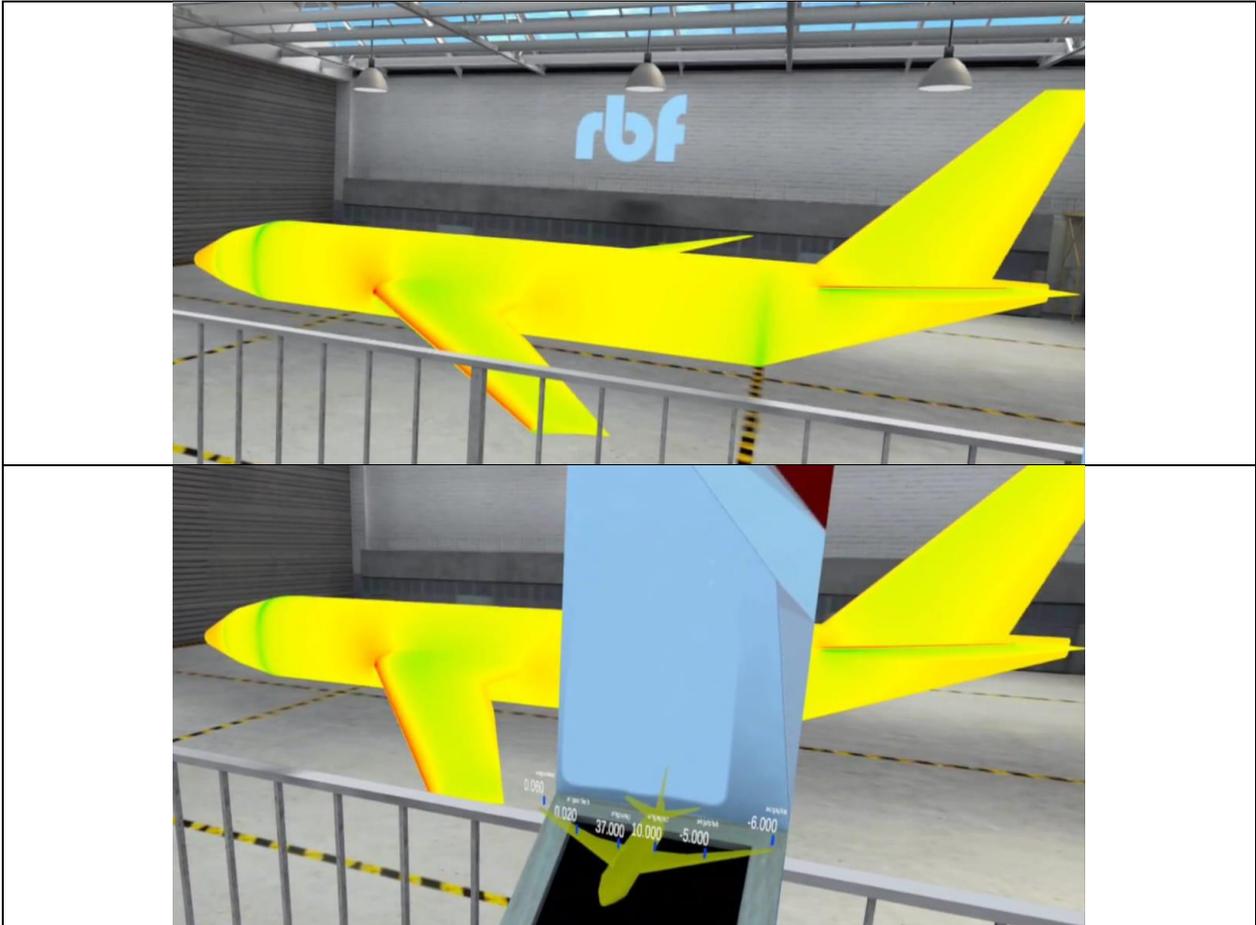


Figure 201: Full plane in the hangar with CFD results

6.5.5. Conclusions

In this study, a workflow has been demonstrated to connect the CAD-based parameterization with the CFD analysis results, creating an advanced design dashboard for real-time and VR applications. Designers can explore the parameter space in mixed reality to identify the optimal design by observing both scalar outputs, such as Cd or Cl, and field quantities, like 3D pressure fields or 2D Cp profiles across various chords.

A specific workflow was developed to transfer the CAD-based parameterization to the mesh level. This approach proved highly effective as it preserves the mesh topology and maintains its quality. The mesh quality is monitored by evaluating the Minimal Orthogonal Quality for each DP. Table 48 shows the Minimal Orthogonal Quality for some points of the DOE (including the most critical ones). It can be observed that it remains within acceptable limits even for very significant shape variations. This method overcomes the main limitations of other conventional approaches. In fact, it allows the mesh and CAD to remain connected at all times, offering a computationally efficient solution to the problem of automatic mesh generation. CAD and mesh are almost perfectly overlapping. This aspect is evaluated by projecting the mesh onto the corresponding CAD surface. As shown in Figures 182 and 183, the distance between the mesh points and the CAD surfaces is always very small.

ROMs were used to create metamodels for real-time interaction, allowing visualization of the field quantities of interest. This method is robust and sufficiently accurate. Specifically, two ROMs are required: one for the mesh and one for the pressure field. The ROM error is always less than 5%, while the maximum pointwise error is 10% for the pressure field. The results were considered sufficiently accurate, but greater precision could be achieved by increasing the number of modes and DPs. Future studies will likely explore these scenarios to further improve the accuracy of the reduced-order models.

Finally, the ROMs were imported into the MetaQuest3 viewer. The result is a platform where the CAD editor is directly linked to the CFD results. Mixed reality adds interactivity, ease of data visualization, immersion, and the ability to share information and insights. Figures 199-201 show previews of the real-time dashboard: the user can interact with the geometric model through sliders or by moving knobs on the airplane model, and the geometry and aerodynamics are updated in real time.

Future developments will involve exploring larger datasets and evaluating a greater number of parameters for more complex geometries and meshes. Additionally, ROMs can be used to evaluate other field quantities, such as velocity, to visualize other entities like streamlines in mixed reality.

Conclusions

This work developed a methodology to link multi-physics and multi-fidelity analyses, aiming to create workflows and optimization dashboards that make data accessible in real time for integration with digital twins. Various problems were explored, and different methods were proposed, focusing on connecting high-fidelity analyses, such as CFD and FEM, with low-fidelity preliminary design analyses using a multi-fidelity approach.

A crucial aspect of this study was the integration of different physical phenomena within a single workflow, addressing challenges such as communication between different software tools and the use of mapping algorithms, particularly RBF, to transfer information between different CAE models.

The use of RBF for mesh morphing showed significant advantages, including reduced meshing noise, lower computational times, and the ability to automate workflows without needing to create a new mesh for each design point. However, it was also recognized that there could be limitations in controlling shape variations, especially when transitioning between complex profiles. To address this challenge, a hybrid workflow combining the benefits of mesh morphing and parametric CAD was proposed.

The study also delved into the use of numerical and analytical methods in multi-fidelity approaches, with a particular focus on data management.

For real time predictions, techniques such as ROMs and adjoint methods emerged as promising solutions. ROM simulations, used to analyze and optimize models of increasing complexity, proved to be highly accurate but required large datasets for model training. Similarly, the adjoint method enabled real-time optimization of complex systems, though its main limitation lies in the linearization of results.

Finally, the creation of real-time optimization dashboards, integrating CAE models, opened new opportunities for the analysis and design of complex structures, as demonstrated with the OPAM prototype. The use of tools such as FMU and VR further facilitated transfer of data and simulation visualization and understanding, improving interaction.

In conclusion, the approach developed in this work provides a solid framework for integrating multi-physics and multi-fidelity analyses, optimizing design processes, and improving real-time data accessibility. The use of advanced techniques such as ROMs and adjoint methods, combined with automated workflows, offers significant opportunities for engineering design, reducing computational times and increasing overall efficiency.

References

1. *Open-source complex-geometry 3D fluid dynamics for applications with unpredictable heterogeneous dynamic high-performance-computing loads.* Bakosi, J. s.l. : *Computer Methods in Applied Mechanics and Engineering*, 2024, *Computer Methods in Applied Mechanics and Engineering*. <https://doi.org/10.1016/j.cma.2023.116586>.
2. *High-order space–time parallel computing of the Navier–Stokes equations.* Meiyuan Zhen, Xuan Liu, Xuejun Ding, Jinsheng Cai. s.l. : *Computer Methods in Applied Mechanics and Engineering*, 2024. <https://doi.org/10.1016/j.cma.2024.116880>.
3. *fortissimo project.* [Online] <https://www.rbf-morph.com/services/fortissimo-eu-project/>.
4. *HPC in the medical field: Overview and current example.* Koch, M., et al. s.l. : *Technol. Health Care* , 2023.
5. *Subspace graph networks for real-time granular flow simulation with applications to machine-terrain interactions.* Amin Haeri, Daniel Holz, Krzysztof Skonieczny. s.l. : *Engineering Applications of Artificial Intelligence*, 2024. <https://doi.org/10.1016/j.engappai.2024.108765>.
6. *Asynchronous GPU-based DEM solver embedded in commercial CFD software with polyhedral mesh support.* Alireza Kianimoqadam, Justin L Lapp. s.l. : *Powder Technology*, 2024. <https://doi.org/10.1016/j.powtec.2024.120040>.
7. *Forced response analysis of the rotor blade rows with the ROM-based aeroelastic model.* Jingyuan Yang, Weiwei Zhang. s.l. : *Aerospace Science and Technology*, 2023. <https://doi.org/10.1016/j.ast.2023.108366>.
8. *Accelerating unsteady aerodynamic simulations using predictive reduced-order modeling.* Zilong Li, Ping He. s.l. : *Aerospace Science and Technology*, 2023. <https://doi.org/10.1016/j.ast.2023.108412>.
9. *Utilizing neural networks and genetic algorithms in AI-assisted CFD for optimizing PCM-based thermal energy storage units with extended surfaces.* A. Ali Rabienataj Darzi, S. Morteza Mousavi, Milad Razbin, Ming Li. s.l. : *Thermal Science and Engineering Progress*, 2024. <https://doi.org/10.1016/j.tsep.2024.102795>.
10. *Machine-learning aided calibration and analysis of porous media CFD models used for rotating packed beds.* Ahmed M. Alatyar, Abdallah S. Berrouk. s.l. : *International Journal of Thermofluids*, 2024. <https://doi.org/10.1016/j.ijft.2024.100845>.
11. *Development and evaluation of an immersive VR-CFD-Based tool for dust exposure mitigation in underground tunnelling operations.* Ting Ren, Ming Qiao, Jon Roberts, Jennifer Hines, Yang-Wai Chow, Wei Zong, Adrian Sugden, Mark Shepherd, Matthew Farrelly, Gareth Kennedy, Faisal Hai, Willy Susilo. s.l. : *Tunnelling and Underground Space Technology*, 2024. <https://doi.org/10.1016/j.tust.2023.105496>.
12. *Interactive CFD simulations with virtual reality to support learning in mixing.* Serkan Solmaz, Tom Van Gerven. s.l. : *Computers & Chemical Engineering*, 2022. <https://doi.org/10.1016/j.compchemeng.2021.107570>.
13. *Biancolini, Marco E. ast Radial Basis Functions for Engineering Applications.* s.l. : Springer, 2017.
14. *Advanced RBF Methods for Mapping Aerodynamic Loads onto.* Andrea Chiappa, Andrea Lopez, Corrado Groth. s.l. : *Fluids*, 2024.
15. *A new meshless approach to map electromagnetic loads for FEM analysis on DEMO TF coil system.* Marco Evangelos Biancolini, Carlo Brutti, Francesco Giorgetti, Luigi Muzzi, Simonetta Turtù, Alessandro Anemona. s.l. : *Fusion Engineering and Design*, 2015.
16. *RIBES Project.* [Online] <http://ribes-project.eu/>.
17. *A balanced load mapping method based on radial basis functions and fuzzy sets.* M.E. Biancolini, A. Chiappa, F. Giorgetti, C. Groth, U. Cella, P. Salvini. <https://doi.org/10.1002/nme.5850>.

18. A smoothed particle radiation hydrodynamics code for modeling high intensity laser-plasma interactions. Min Ki Jung, Hakhyeon Kim, Su-San Park, Eung Soo Kim, Yong-Su Na, Sang June Hahn. s.l. : *Journal of Computational Physics*, 2024. <https://doi.org/10.1016/j.jcp.2024.113000>.
19. Numerical computations of magnetohydrodynamic (MHD) thermal fluid flow in a permeable cavity: A time dependent based study. Bai Mbye Cham, Shams-ul Islam, Afraz Hussain Majeed, Mohamed R. Ali, Ahmed S. Hendy. s.l. : *Case Studies in Thermal Engineering*, 2024. <https://doi.org/10.1016/j.csite.2024.104905>.
20. Effect of reaction rate and pore structure heterogeneity on reactive solute transport in porous media. Mengzi Ren, Liping Qiao, Zhechao Wang. s.l. : *Gas Science and Engineering*, 2025. <https://doi.org/10.1016/j.jgsce.2024.205498>.
21. A novel version of k nearest neighbor: Dependent nearest neighbor. Ömer Faruk Ertuğrul, Mehmet Emin Tağluk. s.l. : *Applied Soft Computing*, 2017. <https://doi.org/10.1016/j.asoc.2017.02.020>.
22. A projection-based Characteristic Mapping method for tracer transport on the sphere. Seth Taylor, Jean-Christophe Nave. s.l. : *Journal of Computational Physics*, 2023. <https://doi.org/10.1016/j.jcp.2022.111905>.
23. Comparing Different Methods for the Coupling of Non-Matching Meshes in Fluid-Structure Interaction Computations. Aukje Boer, Hester Bijl and Alexander Zuijlen. s.l. : *AIAA conference*, 2017. <https://doi.org/10.2514/6.2005-4620>.
24. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. Armin Beckert, Holger Wendland. s.l. : *Aerospace Science and Technology*, 2001. [https://doi.org/10.1016/S1270-9638\(00\)01087-7](https://doi.org/10.1016/S1270-9638(00)01087-7).
25. An evaluation of computational algorithms to interface between CFD and CSD. M Smith, C Cesnik, D Hodges, K Moran, K. s.l. : *Proceedings of the 37th Structure, Structural Dynamics and Materials Conference*.
26. On the automatic construction of interface coupling operators for non-matching meshes by optimization methods. Radim Dvořák, José A. González. s.l. : *Computer Methods in Applied Mechanics and Engineering*, 2024. <https://doi.org/10.1016/j.cma.2024.117336>.
27. The modified weighted residual formulation in the wave based method for plate bending problems: A general formulation for different types of edge restraints. Ling Liu, Francesco Ripamonti, Roberto Corradi, Zhushi Rao. s.l. : *Journal of Sound and Vibration*, 2021. <https://doi.org/10.1016/j.jsv.2021.116329>.
28. Wind Tunnel Model Design and Aeroelastic Measurements of the RIBES Wing. U Cella, P Della Vecchia, C Groth, S Porziani, F Giorgetti, F Nicolosi, M E Biancolini. s.l. : *Journal of Aerospace Engineering*, 2021. [https://doi.org/10.1061/\(ASCE\)AS.1943-5525.0001199](https://doi.org/10.1061/(ASCE)AS.1943-5525.0001199).
29. Comparison between an uncoupled one-way and two-way fluid structure interaction simulation on a super-tall slender structure. K. Wijesooriya, D. Mohotti, A. Amin, K. Chauhan. s.l. : *Engineering Structures*, 2021. <https://doi.org/10.1016/j.engstruct.2020.111636>.
30. C Groth, Cella U, Costa E, Biancolini M E. Validation of High Fidelity Computational Methods for Aeronautical FSI Analyses. *Flexible Engineering Toward Green Aircraft*. s.l. : Springer.
31. A. Martinez-Pascual, Marco Evangelos Biancolini & J. Ortega-Casanova. Fluid Structure Modelling of Ground Excited Vibrations by Mesh Morphing and Modal Superposition. *Flexible Engineering Toward Green Aircraft* . s.l. : Springer , 2021.
32. Fast high fidelity CFD/CSM fluid structure interaction using RBF mesh morphing and modal superposition method. Corrado Groth, Ubaldo Cella, Emiliano Costa, Marco Evangelos Biancolini. s.l. : *Aircraft Engineering and Aerospace Technology*, 2019.
33. —.Groth, C., Cella, U., Costa, E. and Biancolini, M.E. s.l. : *Aircraft Engineering and Aerospace Technology*, 2019. <https://doi.org/10.1108/AEAT-09-2018-0246>.

34. *two-dimensional finite difference models for the thermal study of freight car braking*. Cantone, Siniscalchi. s.l. : AIAS Conference, 2023.
35. *Simulation of the thermo-mechanical behaviour of tread braked railway wheels by means of a 2D finite element model*. N. Bosso, L. Cantone, G. Falcitelli, R. Gjini, M. Magelli, F.M. Nigro, E. Ossola, N. Zampieri. s.l. : Tribology International, 2023. <https://doi.org/10.1016/j.triboint.2022.108074>.
36. *Generation of Finite Difference Formulas*. Fornberg, B.
37. *Application of thermoelasticity in the frequency-domain multiaxial vibration-fatigue criterion*,. Jaša Šonc, Klemen Zaletelj, Janko Slavič. s.l. : Mechanical Systems and Signal Processing, 2025. <https://doi.org/10.1016/j.ymsp.2024.112002>.
38. *CUSP: a two CubeSats constellation for space weather and solar flares x-ray polarimetry*. Giovanni Lombardi, Andrea Lopez and all. s.l. : SPIE, 2022. <https://doi.org/10.1117/12.2629233>.
39. *Ducted heat exchanger thermal and aerodynamic shape optimization*. Eytan J. Adler, Andrew H. R. Lamkin, and Joaquim R. R. A. Martins. s.l. : ICAS 2024.
40. *Hao Wang, AiQun Li, Tong Guo & Shuang Ma. Accurate stress analysis on rigid central buckle of long-span suspension bridges based on submodel method*. s.l. : Springer, 2009.
41. *Structural assessment of TF superconducting magnet of the DTT device*. all, Andrea Chiappa et. s.l. : AIAS, 2019. <https://doi.org/10.1016/j.prostr.2020.02.079>.
42. *Advancements and Challenges of Micromechanics-based Homogenization for the Short Fiber Reinforced Composites*. Hugon Lee, Sangryun Lee & Seunghwa Ryu. 2023. <https://doi.org/10.1007/s42493-024-00100-2>.
43. *Finite element homogenization and experimental evaluation of additively manufactured lattice metamaterials*. Zhmaylo, M., Maslov, L., Borovkov, A. et al. 2023. <https://doi.org/10.1007/s40430-023-04210-0>.
44. *Evaluation of equivalent stiffness properties of corrugated board*. Biancolini, Marco E. s.l. : Composite Structures, 2004.
45. *Three-dimensional finite element analysis of double-lap composite adhesive bonded joint using sub-modeling approach*. Bogdanovich A E, Kizhakkethara I. s.l. : Composites, 1999.
46. *Numerical methods for design of bonded joints for ship structures*. Wang X, Mieth U, Cappelletti A M. 2004.
47. *Crack propagation analysis of ITER Vacuum Vessel port stub with Radial Basis Functions mesh morphing*. Edoardo Pompa, Gabriele D'Amico, Stefano Porziani, Francesco Giorgetti, Corrado Groth, Alfredo Portone, Marco E. Biancolini. s.l. : Fusion Engineering and Design, 2020.
48. *Using HPC for Computational Fluid Dynamics*. Jamshed, S. s.l. : Shamoan Jamshed, 2015. <https://doi.org/10.1016/B978-0-12-801567-4.00008-8>.
49. *CPU and GPU parallel efficiency of ARM based single board computing cluster for CFD applications*. Di Pierro, B. and Hank, S. s.l. : Computers & Fluids, 2024, Vol. 272. <https://doi.org/10.1016/j.compfluid.2024.106187>.
50. *Parametric aeroelastic modeling, maneuver loads analysis using CFD methods and structural design of a fighter aircraft*. Voß, A. and Klimmek, T. s.l. : Aerospace Science and Technology, 2023, Vol. 136. <https://doi.org/10.1016/j.ast.2023.108231>.
51. *CFD design capabilities for next generation high-speed aircraft*. Viviani, A., et al. s.l. : Acta Astronautica, 2021, Vol. 178. <https://doi.org/10.1016/j.actaastro.2020.09.006>.
52. *Progress in Geometry Modeling and Mesh Generation Toward the CFD Vision 2030*. Taylor, J. R. and Chawner, N. J. s.l. : AIAA 2019. <https://doi.org/10.2514/6.2019-2945>.
53. *Sails trim optimisation using CFD and RBF mesh morphing*. Biancolini, M. E., Viola, I. M. and Riotte, M. s.l. : Computers & Fluids, 2014, Vol. 93. <https://doi.org/10.1016/j.compfluid.2014.01.007>.

54. Mann, S A Coons and R W. *Computer-Aided Design Related to the Engineering Design Process*.
55. C, Waguespack. *Mastering Autodesk Inventor 2015 and Autodesk Inventor LT 2015*. 2014.
56. *OpenCSM: An Open-Source Constructive Solid Modeler for MDAO*. Dannenhofer, J F. s.l. : AIAA 2013, 2013.
57. [Online] <https://dev.opencascade.org/>.
58. *Automatic modeling of aircraft external geometries for preliminary design workflows*. . s.l. : *Aerospace Science and Technology*, 2020, Vol. 98. 10.1016/j.ast.2019.105667.
59. *Multi-objective optimization for centrifugal compressor of mini turbojet engine*. . Guo, S., et al. s.l. : *Aerospace Science and Technology*, 2014. <https://doi.org/10.1016/j.ast.2014.04.014>..
60. *CAD-integrated parametric design and analysis of lightweight shell structures*. . Goldbach, A.-K. and C., Lázaro. s.l. : *Structures*, 2024. <https://doi.org/10.1016/j.istruc.2024.106566>..
61. *RBF-based morphing of B-Rep models for use in aerodynamic shape optimization*. . Gagliardi, F. and Giannakoglou, K. C. s.l. : *Advances in Engineering Software*, 2019. <https://doi.org/10.1016/j.advengsoft.2019.102724>.
62. *Opencascade*. [Online] <https://dev.opencascade.org/doc/overview/html/>.
63. *Multidisciplinary design optimization for aircraft wing using response surface method, genetic algorithm, and simulated annealing*. Xuan-Binh Lam, D. s.l. : *Journal of Science and Technology in Civil Engineering*, 2019. <https://doi.org/10.31814/stce.nuce2020->.
64. *Surrogate-Based Aerodynamic Shape Optimization of a Civil Aircraft Engine Nacelle*. Keane, Wenbin Song and Andy J. s.l. : *AIAA Journal*, 2007. 10.2514/1.30015.
65. *Function minimization by conjugate gradients*. Fletcher, R. s.l. : *The Computer Journal*, 1964.
66. *Adjoint-based shape optimization workflows using RBF*. Groth, Corrado. 2016.
67. *A COUPLED-ADJOINT METHOD FOR HIGH-FIDELITY AERO-STRUCTURAL OPTIMIZATION*. Martins, Joaquim R. R. A. 2002.
68. *Design in nature: learning from trees*. Mattheck, C. s.l. : Springer.
69. *Automatic shape optimisation of structural parts driven by BGM and RBF mesh morphing*. Porziani, S., Groth, C., Waldman, W., & Biancolini, M. E. s.l. : *International Journal of Mechanical Sciences*, 2021. <https://doi.org/10.1016/j.ijmecsci.2020.105976>.
70. *Automatic Optimization Method Based on Mesh Morphing Surface Sculpting Driven by Biological Growth Method: An Application to the Coiled Spring Section Shape*. Porziani S, De Crescenzo F, Lombardi E, Iandiorio C, Salvini P and Biancolini M E. s.l. : *Computer Science*, 2021. 10.1007/978-3-030-77977-1_38.
71. *A survey of projection-based model reduction methods for parametric dynamical systems*. Benner P, Gugercin S and Willcox K. s.l. : *SIAM*, 2015.
72. *Reduced basis methods for partial differential equations: an introduction*. Quarteroni A, Manzoni A and Negri F. s.l. : Springer, 2015.
73. *Certified reduced basis methods for parametrized partial differential equations*. Hesthaven JS, Rozza G, Stamm B, et al. s.l. : Springer, 2016.
74. *Fusing wind-tunnel measurements and CFD data using constrained gappy proper orthogonal decomposition*. *Aerospace Science and Technology*. Mifsud, M., et al. 2019, Vol. 86. <https://doi.org/10.1016/j.ast.2018.12.036>.
75. *Modal analysis of fluid flows: an overview*. Taira K, Brunton SL, Dawson ST, et al. s.l. : AIAA , 2017.

76. *Multidisciplinary design optimization for aircraft wing using response surface method, genetic algorithm, and simulated annealing.* Xuan-Binh Lam, D. s.l. : *Journal of Science and Technology in Civil Engineering*, 2019. [https://doi.org/10.31814/stce.nuce2020-14\(1\)-03](https://doi.org/10.31814/stce.nuce2020-14(1)-03).
77. *Multi-objective optimization of a mini-channel heat sink with non-uniform fins using genetic algorithm in coupling with CFD models.* Ge, Y., et al. s.l. : *Applied Thermal Engineering*, 2022.
78. *Simulation of hybrid boiling nano fluid flow with convective boundary conditions through a porous stretching sheet through Levenberg Marquardt artificial neural networks approach.* . Noreen Sher Akbar, Tayyab Zamir, Javaria Akram, Tayyaba Noor, Taseer Muhamm. s.l. : *International journal of Heat and Mass transfers*, 2024.
79. *Solving inverse heat conduction problems using trained POD-RBF network inverse method.* Z. Ostrowski, R. A. Bialecki & A. J. Kassab. 2008. <http://dx.doi.org/10.1080/17415970701198290>.
80. *Modelling and Simulation of Thermal Power Generation Systems with Modelica.* Casella, F. and Leva, A. 2024. <https://doi.org/10.1016/j.ifacol.2024.07.487>.
81. *Modelica-based simulation of electromagnetic transients using Dynawo: Current status and perspectives.* Masoom, A., et al. 2021. <https://doi.org/10.1016/j.epsr.2021.107340>.
82. *A CFD-based numerical virtual flight simulator and its application in control law design of a maneuverable missile model.* Laiping ZHANG, Xinghua CHANG, Rong MA, Zhong ZHAO, Nianhua WANG. s.l. : *Chinese Journal of Aeronautics*, 2019, Vol. 32. <https://doi.org/10.1016/j.cja.2019.07.001>.
83. *Advanced RBF Methods for Mapping Aerodynamic Loads onto Structures in High-Fidelity FSI Simulations.* Chiappa, A., Lopez, A., Groth C. s.l. : *Fluids*, 2024. <https://doi.org/10.3390/fluids9060137>.
84. *balanced load mapping method based on radial basis functions and fuzzy sets.* Biancolini, M.E., et al. 2018.
85. *preCICE v2: A Sustainable and User-Friendly Coupling Library.* . [Online] <https://open-research-europe.ec.europa.eu/articles/2-51/v2>.
86. *Numerical Methods for Fluid-Structure Interaction - A Review.* Hou G, Wang J, Layton A. s.l. : *Communications in Computational Physics*, 2012. 10.4208/cicp.291210.290411s.
87. *Experimental analysis of high Reynolds number structural dynamics in ETW.* Ballmann, J. s.l. : *AIAA*, 2008.
88. *FSI HIRENASD tesi.* [Online] https://www.rbf-morph.com/storage/2024/01/D.-Regine-Presentation_RBF-MORPH.pdf.
89. *High Reynolds Number Aerostructural Dynamics (HIRENASD).* [Online] <http://heinrich.lufmech.rwth-aachen.de/en/windtunnel-assembly>.
90. *Validation of High Fidelity Computational Methods for Aeronautical FSI Analyses.* all, Groth C et. s.l. : *Springer*, 2020, Vol. *Flexible Engineering Toward Green Aircraft*.
91. *Shape Optimization of Frame Structures through a Hybrid Analytical-2D and Numerical-3D Approach.* Lopez A, Iandiorio C, Milani D, Salvini P, Biancolini M E. s.l. : *AIAS*, 2024.
92. *Comprehensive study on the heat transfer characteristics of windward bend lattice frame structure.* Xiaohui, B., et al. s.l. : *Propulsion and Power Research*, 2022. <https://doi.org/10.1016/j.jprr.2022>.
93. *A systematic review of design for additive manufacturing of aerospace lattice structures: Current trends and future directions.* Khan, N. and Riccio, A. s.l. : *Progress in Aerospace Sciences*, 2024. <https://doi.org/10.1016/j.paerosci.2024.101021>.
94. *Evaluation of computational homogenization methods for the prediction of mechanical properties of additively manufactured metal parts.* March, N.G., Gunasegaram, D.R. and Murphy, A.B. s.l. : *Additive Manufacturing*, 2023. <https://doi.org/10.1016/j.addma.2023.103415>.

95. Shape optimization of 2D beam-structures to obtain the uniform strength. Iandiorio, C., Milani, D. and Salvini, P. s.l. : Mater. Sci. Eng. DOI 10.1088/1757-899X/1275/1/012030.
96. Nostrand, Timoshenko SP. D. Van. *Company Strength of Materials (parts I & II)*. 1940.
97. AeroSUV. [Online] <https://www.ecara.org/drivaer-1>.
98. Rocco Bombardieri, Rauno Cavallaro, Ruben Sanchez, Nicolas R. Gauger. *Aerostructural wing shape optimization assisted by algorithmic differentiation. Structural and Multidisciplinary Optimization*. s.l. : Springer, 2021.
99. Coupled adjoint fluid-structure interaction technique for flexible wing shape optimization. K. Tsiakas, G. Rogé, X. Trompoukis, V. Asouti, G. Fougeron,. s.l. : ECCOMAS Congress, 2020.
100. Aerostructural Adjoint Method for FlexibleWing Optimization. I. Ghazlane, G. Carryery , A. Dumont and J.A. Desideri. s.l. : AIAA, 2012.
101. Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations. Gaetan K. W. Kenway, Graeme J. Kennedy, and Joaquim R. R. A. Martins. s.l. : AIAA, 2014.
102. Coupled Adjoint-Based Sensitivities in Large-Displacement Fluid-Structure Interaction using Algorithmic Differentiation. R. Sanchez, T. Albring2, R. Palacios , N. R. Gauge, T. D. Economon, J. J. Alonso. s.l. : International Journal for Numerical Methods in Engineering, 2017. //doi.org/10.1002/nme.5700.
103. <https://commonresearchmodel.larc.nasa.gov/fem-file/wing-body-alone-configuration/>. [Online]
104. Validation of Wing Deformation Simulations for the NASA CRM Model using Fluid-Structure Interaction Computations. s.l. : AIAA, 2015. <https://doi.org/10.2514/6.2015-0619>.
105. 2022 FORMULA 1 TECHNICAL REGULATIONS. [Online] 2022. https://www.fia.com/sites/default/files/formula_1_-_technical_regulations_-_2022_-_iss_9_-_2022-02-18.pdf.
106. 3D CFD simulation of the interaction between front wheels&brake ducts and optimised five-element F1 race car front wings under regulations. F.-J. Granados-Ortiz, P. Morales-Higueras, J. Ortega-Casanova. s.l. : Alexandria Engineering Journal, 2023. //doi.org/10.1016/j.aej.2023.02.011.
107. Aerodynamic Study of the Wake Effects on a Formula 1 Car. Castilla, Alex Guerrero and Robert. s.l. : Energies, 2020. //doi.org/10.3390/en13195183.
108. Aerodynamic simulation of a 2017 F1 car. U. Ravelli, M. Savini. s.l. : JOURNAL OF TRAFFIC AND TRANSPORTATION ENGINEERING, 2018.
109. Gaussian Regression on TensorFlow. [Online] https://www.tensorflow.org/probability/examples/Gaussian_Process_Regression_In_TFP?hl=it.
110. trainlm MATLAB. [Online] <https://it.mathworks.com/help/deeplearning/ref/trainlm.html>.
111. ESDU 86002. Drag and pressure recovery characteristics of auxiliary air inlets at subsonic speeds. 2004.
112. RBF Morph. [Online] <https://www.rbf-morph.com/>.
113. Ansys Twin Builder. [Online] <https://www.ansys.com/products/digital-twin/ansys-twin-builder>.
114. Design And Optimization Of Aeronautical Components And Digital Twins Development. Lopez A, Cella U, Groth C, Biancolini M E. s.l. : AIAS, 2023.
115. Machining FEM model of long fiber composites for aeronautical components. C. Santiuste, X. Soldani, M. H. Miguélez. s.l. : Composite Structures, 2010.
116. The payload design of the CUBesat Solar Polarimeter (CUSP), for Space Weather and Solar flares X-ray polarimetry. Lombardi G, Lopez A et all. s.l. : AIAS, 2024.

117. *Development of a Reduced Order Model-Based Workflow for Integrating Computer-Aided Design Editors with Aerodynamics in a Virtual Reality Dashboard: Open Parametric Aircraft Model-1 Testcase.* Lopez A, Biancolini M E. s.l. : Applied Sciences, 2024.
118. *Summary of Exascale and Remeshing Efforts for the Second Geometry and Mesh Generation Workshop.* Carolyn Woeber, James S. Masters and David R. McDaniel. s.l. : AIAA, 2019.
119. *Analysis of GMGW2 Case 3: Design Variations.* Dannenhoffer, John F. s.l. : AIAA, 2019.
120. *On The Construction of Aircraft Conceptual On The Construction of Aircraft Conceptual.* Robert Haimes, Mark Drela. s.l. : AIAA, 2012.
121. *Automatic Unstructured Mesh Generation with Geometry Attribution.* Steve Karman, Nick Wyman. s.l. : AIAA, 2019.
122. *Progress in Geometry Modeling and Mesh Generation Toward the CFD Vision 2030.* John R. Chawner, Nigel J. Taylor. s.l. : AIAA, 2019.
123. *Automation of Overset Structured Surface Mesh Generation on Complex Geometries.* William M. Chan, Shishir A. Panday, Robert Haimes. s.l. : AIAA, 2019.
124. Cella U., Biancolini M. E. [Online] RBF Morph, 2019. <https://www.rbf-morph.com/wp-content/uploads/2019/02/GMGW2-Case3-Wade.pdf>.
125. Magri G, Lopez A, Della Barba F. *Ottimizzazione aerodinamica di un velivolo mediante CAD parametrico e Mesh Morphing.* 2023.
126. *RBF-based morphing of B-Rep models for use in aerodynamic shape optimization.* Flavio Gagliardi, Kyriakos C. Giannakoglou. s.l. : Advances in Engineering Software, 2019.
127. *Cadquery documentation.* [Online] <https://cadquery.readthedocs.io/en/latest/>.
128. *Rapidly predicting the effect of tool geometry on the wrinkling of biaxial NCFs during composites manufacturing using a deep learning surrogate model.* J.V. Viisainen, F. Yu, A. Codolini, S. Chen, L.T. Harper, M.P.F. Sutcliffe. s.l. : Composites Part B: Engineering, 2023, Vol. 253.
129. *Automatic modeling of aircraft external geometries for preliminary design workflows.* s.l. : Aerospace Science and Technology, 2020, Vol. 98.
130. *Multi-objective optimization for centrifugal compressor of mini turbojet engine.* Shuai Guo, Fei Duan, Hui Tang, Seng Chuan Lim, Mee Sin Yip. s.l. : Aerospace Science and Technology, 2014. <https://doi.org/10.1016/j.ast.2014.04.014>.
131. *CAD-integrated parametric design and analysis of lightweight shell structures.* Ann-Kathrin Goldbach, Carlos Lázaro. s.l. : Structures, 2024. <https://doi.org/10.1016/j.istruc.2024.106566>.
132. [Online] <https://dev.opencascade.org/doc/overview/html/>.