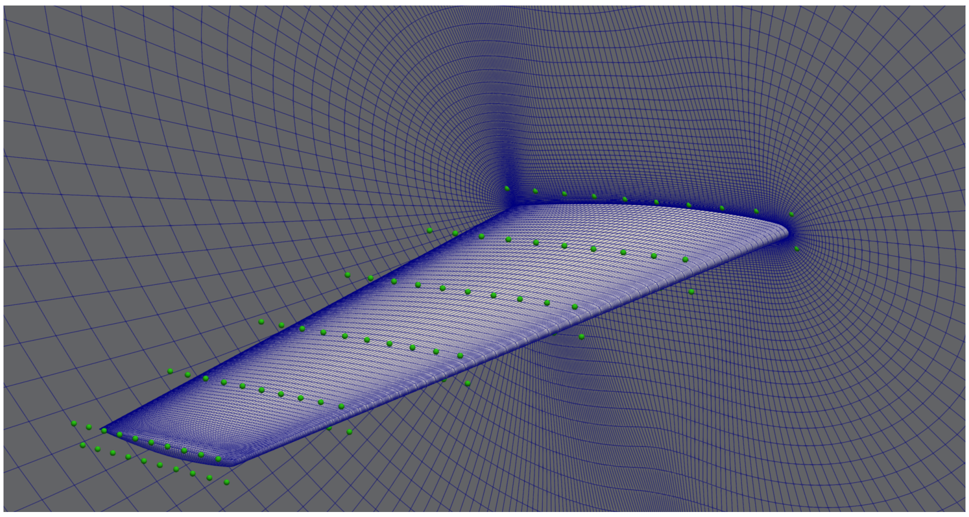
University of Liège - School of Engineering

Master's thesis completed in order to obtain the degree of Master of Science in Aerospace Engineering

Aerostructural optimization of an aircraft wing assisted by dimensionality reduction methods



Author:
Julien Caudron

Supervisors:
Pr. Koen Hillewaert (ULiège)
Dr. Rajan Filomeno Coelho (Cenaero)

Academic year 2023-2024

"It's not the plane, it's the pilot."

Maverick

Abstract

Advancements in numerical methods have increasingly allowed for the modeling and simulation of complex problems with greater accuracy. However, as these simulations grow in complexity, they become computationally expensive, especially as the growth in computer processing speed appears to have plateaued. In industrial applications, where computational resources are limited, it is essential to find techniques that reduce these costs without compromising accuracy.

This thesis presents a framework that integrates dimensionality reduction methods into an aerostructural optimization process, aiming to improve the performances of the Onera M6 wing, a well-known wing that operates under transonic conditions where shockwaves occur. The wing geometry is modified using the Free-Form Deformation (FFD) method, which allows for smooth deformations based on a parameterization involving 125 design variables. However, this high-dimensional (HD) problem poses challenges for optimization, as the vast design space makes it difficult for the optimizer to locate the global maxima that would enhance the wing performances.

To address this, the study employs dimensionality reduction techniques to simplify the optimization problem. Specifically, the feature selection method, particularly the filter method, is used to rank design variables by their impact on wing performances, setting aside the less influential ones. This approach reduces the dimensionality of the problem, making the optimization process more manageable and cost-effective.

The research is organized into five key sections. The first section focuses on developing a Computational Fluid Dynamics (CFD) setup to accurately model the Onera M6 wing performances, using OpenFOAM to simulate the transonic, compressible, and steady-state fluid-body interaction. The second section details the FFD method and its parameterization. The third section introduces and applies dimensionality reduction methods to the problem at hand. The fourth section covers the optimization process, using two different algorithms to optimize the wing performances. Finally, the last section provides recommendations for further research and potential enhancements to the framework.

This study demonstrates that even with a reduction in dimensionality, the optimization process can yield significant performance improvements while maintaining accuracy, offering valuable insights for industrial applications where computational efficiency is critical.

Keywords: Dimensionality Reduction — Feature Selection — Multidisciplinary Design Optimization — Free-Form Deformation — Computational Fluid Dynamics — Onera M6 wing — Aerodynamic Shape Optimization

Acknowledgements

I would like to express my deepest gratitude to my parents, whose unwavering support and encouragement have been the foundation of my entire academic journey. Despite the challenges along the way, they never doubted my ability to succeed. Becoming an engineer is a fulfilling achievement, and I hope it brings them as much pride as it brings me. A special thank you to my father, who helped me correct and finalize this master's thesis; your assistance was invaluable.

I extend my heartfelt thanks to Professor Hillewaert for his invaluable guidance throughout the construction of the CFD setup. His insightful feedback and numerous suggestions greatly contributed to the progress and refinement of my work. His encouragement to explore a broad range of parameters was instrumental in enhancing the quality and depth of this research.

I would also like to thank Cenaero for the opportunity to undertake this insightful internship. In particular, I am grateful to Rajan, who was incredibly supportive, attentive, and resourceful. Our weekly meetings were a constant source of inspiration, motivating me to continually improve and persevere toward the completion of this master's thesis. I sincerely appreciate your time and the promptness with which you responded to my queries, ensuring I always had the support I needed. Your valuable advice and observations significantly enhanced the comprehensiveness of this work.

Finally, I wish to thank all the professors, students, and friends who have supported me throughout my studies. Your companionship made this journey not only successful but also unforgettable.

Contents

List of Figures

List of Tables

List of Abbreviations

FFD	**F**ree-**F**orm **D**eformation
CFD	**C**omputational **F**luid **D**ynamics
RANS equations	**R**eynolds-**A**veraged **N**avier-**S**tokes equations
DoE	**D**esign **o**f **E**xperiment
DV	**D**esign **V**ariable
HD	**H**igh **D**imensional
LD	**L**ow **D**imensional
COBYLA	**C**onstrained **O**ptimization **BY** **L**inear **A**pproximation
DE	**D**ifferential **E**volution

To my dear and lovely Laura

Chapter 1

Introduction

1.1 Context

Since the dawn of aviation, aircraft manufacturers have been working on improving their creations. The pursuit of the perfect aircraft is driven by a desire for innovation. Crafting planes that are not only technologically advanced and comfortable but, most importantly, highly efficient. Efficiency in aviation has a wide range of applications. Manufacturers aim at extending the ranges of flights, increasing payload capacity and cruise velocity, reducing both construction and maintenance costs, enhancing safety and ensuring reliability. Yet, among those goals, there is one other aspect that is crucial in today's world: minimizing fuel consumption.

The aviation industry faces huge challenges driven by global efforts in reducing and eventually annihilating CO_2 emissions. With recent technological advances, the future of aviation will provide airplanes that are no longer powered by fossil fuels. This bright vision cannot be achieved at large scale with today's technology. Alternating sources of energy are being explored but their reliability remain a challenge. Consequently, the aviation sector must continue to push the actual planes boundaries by fine-tuning every aspect of the aircraft.

The analysis presented in this study focuses on a crucial airplane component: the wings. These play a major role in its performance as they are the primary source of lift, the very essence of flying. Lift, however, comes hand in hand with drag, and the balance between these two forces is a key parameter when talking about plane efficiency. This thesis delves into the lift-to-drag ratio, a parameter that must be optimized to enhance wing performances.

This optimization of 3D wings involves dealing with numerous parameters leading to a high dimensional problem that can be challenging for optimizers. This well-known mathematical issue, called the *curse of dimensionality*, is a common challenge in machine learning or computational assisted methods. Hopefully, there exist methods that can mitigate this issue by reducing the dimensionality of the problem. Those methods must be analysed to asses their reliability.

1.2 Objectives

This work is dedicated to the development of a robust framework designed to optimize the performance of an aircraft wing. While the primary goal is wing optimization, this framework introduces several key objectives. Optimization is a well-known topic with many algorithms designed for a wide range of applications. However it still offers plenty of opportunities for innovation, especially in industrial settings. This thesis aims to propose a process or methodology that could be smoothly integrated into industrial applications.

In addition to presenting this framework, the study also introduces a more and more popular machine learning technique known as *dimensionality reduction*. This technique is precious for simplifying complex problems by reducing the number of parameters involved. In many applications, optimization algorithms are challenged by the high dimensionality of problems, sometimes involving thousands of parameters. The objective here is to illustrate how dimensionality reduction can be effectively applied and to highlight the benefits it could offer. While there are plenty dimensionality reduction techniques, more or less complex, this work focuses more on introducing this dimensionality reduction approach rather than comparing techniques. However, determining and integrating the best method, for this case, into the framework would enhance accuracy.

Reducing the dimensionality of a problem leads to a loss of information, which must be carefully managed. The main limitation of those techniques is to avoid introducing significant errors in the low dimensional model. This thesis seeks to explore the balance between simplification and accuracy, demonstrating how the careful application of these techniques simplify the optimization process.

In summary, the main research question arises here: **How can dimensionality reduction methods be applied to optimize wing performances?**

1.3 State of the art

In the context of high-fidelity aerostructural optimization, Kenway et al. [1] presented a CAD-free approach aimed at addressing the limitations of traditional CAD-based methods in multidisciplinary design optimization (MDO). The proposed method employs a free-form deformation (FFD) volume approach, enabling the efficient computation of analytic derivatives, which are essential for gradient-based optimization. The authors integrate Computational Fluid Dynamics (CFD) and Computational Structural Mechanics (CSM) into the optimization process, highlighting the advantages of the CAD-free method in managing complex geometries and multidisciplinary interactions. Compared to traditional methods, they conclude that the CAD-free, high-fidelity aerostructural optimization approach provides significant improvements. However, it is important to note that they still requires costly numerical simulations, which may be impractical for low-budget projects.

To improve the computational time of the optimization process, Queipo et al. [2] proposed Surrogate-Based Optimization (SBO). This approach is particularly beneficial when the objective function requires high-fidelity numerical simulations, such as CFD, where evaluating the objective can take several hours. The high computational cost associated with numerous evaluations makes the optimization process expensive and sometimes impractical. SBO addresses this issue by drastically reducing computational time, relying on a metamodel built upon previous observations. However, constructing accurate metamodels can become challenging due to the curse of dimensionality, especially when dealing with a large number of parameters.

To mitigate the dimensionality issue, integrating dimensionality reduction techniques into the optimization process proves effective, as suggested by Gaudrie et al. [3]. They proposed a dimensionality reduction approach using eigenshapes to characterize CAD parameters. In the context of wing shape optimization, the number of CAD parameters required to accurately define the geometry can be substantial. Directly working with these parameters poses challenges for the optimization process, potentially leading to costly and inaccurate results, especially with limited computational budgets.

The mapping between CAD parameters and eigenshapes is performed using Principal Component Analysis (PCA) [4]. This method, applied to the high-dimensional feature space $\mathbf{\Phi}$, identifies the eigenvectors $\mathbf{v}^j$ of the matrix $\mathbf{\Phi}^\top \mathbf{\Phi}$. These eigenvectors are ranked according to the variance (or dispersion) they capture, as indicated by their corresponding eigenvalues λ_j. Consequently, any design can be represented as a combination of these eigenvectors, facilitating a more efficient optimization process.

In other words, the eigenshape space effectively compresses the CAD parameters into a smaller number of significant components, allowing the surrogate model to focus on the most relevant dimensions. With the mentioned strategy, Gaudrie et al. [3] provided a more accurate metamodel and faster optimization compared to traditional approaches that operate directly in the full CAD parameter space.

PCA is not the only technique available for dimensionality reduction. Originally introduced as a preprocessing step in data analysis, this approach is critical for eliminating redundant, noisy, and irrelevant data, as explained by van der Maaten et al. [5] and confirmed by Velliangiri et al. [6]. This reduction is instrumental in improving the accuracy of learning models and decreasing training time. These methods have a broad range of applications, including deep learning, machine learning, medical imaging, and optimization. By simplifying the data, dimensionality reduction enhances the performance of algorithms, making it easier to visualize, manipulate, and classify.

While van der Maaten emphasizes the importance of dimensionality reduction for creating efficient and accurate models, Hou et al. [7] further validate this by summarizing various techniques and underscoring the need for more sophisticated non-linear methods capable of handling complex problems. They highlight that integrating dimensionality reduction with surrogate modeling offers significant potential for improving computational efficiency and accuracy across various engineering fields. However, both van der Maaten and Hou agree that ongoing research is essential to refine these methods and expand their applicability to manage the increasing complexity and size of modern datasets.

1.4 Methodology and flowchart

As stated in the objectives, the methodology consists of several interconnected modules, each with its own specific goals, all working together to achieve the primary objective: optimizing the performances of a wing.

The process begins with selecting a baseline wing. A widely recognized choice for aerodynamic validation is the Onera M6 wing [8]. This transonic wing is small enough for wind tunnel testing and is frequently used to validate CFD simulations due to available experimental data.

With the selected wing, the next step is to deform it using the Free-Form Deformation (FFD) method. FFD allows for smooth and controlled morphing of the baseline wing. Before deformation, a parameterization must be established to guide the process. In this case, 125 design variables are chosen, and the reasoning behind this choice and its implications are discussed in chapter 3.

Once the surface mesh is deformed, a 3D mesh is generated and used in the CFD simulation. The CFD simulation is a critical component of the methodology, as it takes the mesh as input and produces the lift-to-drag ratio as output; this metric is used here for evaluating performances. The setup for this CFD simulation is carefully constructed using OpenFOAM [9] as detailed in chapter 2.

To achieve the goal of finding a *better* wing, the next module in the process is optimization. The primary objective here is to maximize the lift-to-drag ratio, which requires navigating a high-dimensional problem involving 125 parameters. This is where dimensionality reduction comes into play, reducing the problem complexity. The reduction is achieved through feature selection, which ranks the design variables based on their influence on the lift-to-drag ratio. The most impactful parameters are retained, while less significant ones are set aside, as described in chapter 4.

To apply dimensionality reduction, a Design of Experiment (DoE) must first be generated. This involves creating a set of samples (deformed wings) that are linked to the CFD output (lift-to-drag ratio). With this DoE, a regression model is trained to build a metamodel. The optimization process then uses this metamodel to predict the lift-to-drag ratio based on given parameters. This entire process is detailed in chapter 5. This optimization process focuses solely on the lift-to-drag ratio, which, while important, does not consider several key parameters critical to evaluating the overall performances of the wing. A more comprehensive study should also account for factors such as stability effects, control mechanisms, manufacturability, resistance to loads, and many other vital flight parameters.

This methodology is represented as a flowchart in Figure 1.1.

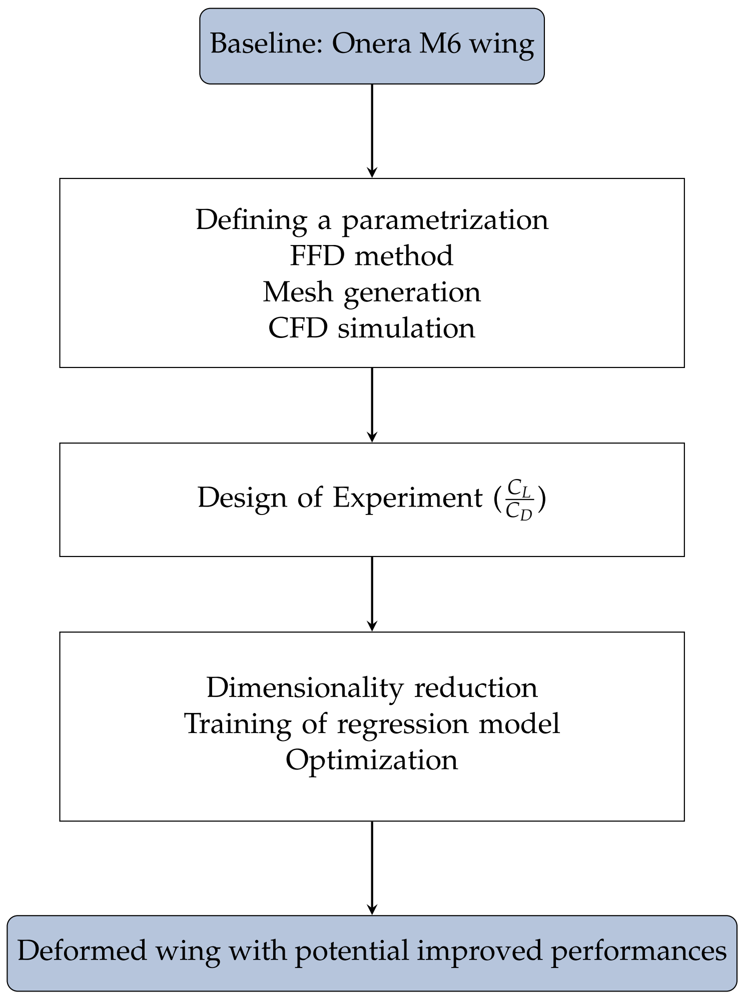

Baseline: Onera M6 wing
Defining a parametrization
FFD method
Mesh generation
CFD simulation
Design of Experiment ($\frac{C_L}{C_D}$)
Dimensionality reduction
Training of regression model
Optimization
Deformed wing with potential improved performances

FIGURE 1.1: Methodology flowchart.

1.5 DAFoam

DAFOAM [10] [11] is a tool that will be referenced frequently throughout the discussion. Understanding DAFoam is crucial for anyone interested in advancing this research. Developed by renowned experts in the field of numerical simulation: H. Ping, C. Mader, J. Martins, and K. Maki. DAFoam stands for Discrete Adjoint with OpenFOAM for High-Fidelity Multidisciplinary Design Optimization. It is an open-source software based on OpenFOAM, designed to perform optimization using the FFD method.

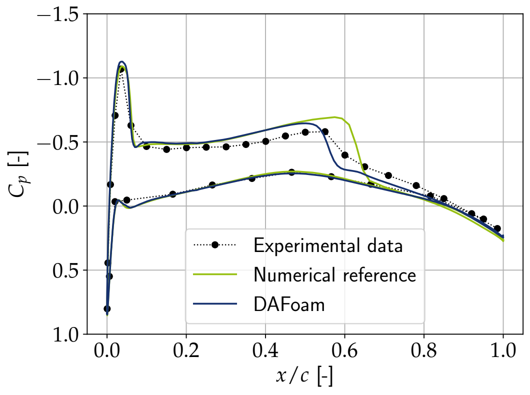
DAFoam is an excellent tool that closely aligns with the framework described in this thesis. Initially, the plan was to use DAFoam as a reference and foundation for this work, which would have saved significant time and resources in establishing a reliable CFD setup. However, DAFoam's baseline case does not align with the specified requirements as the Onera M6 case appears to be corrupted.

DAFoam documentation provides files for the Onera M6 wing case, allowing users to run CFD simulations directly. However, the developers made modifications to these files, most notably by coarsening the mesh. Even after refining the mesh, the results did not align with DAFoam's claims. For instance, DAFoam documentation promises a drag coefficient of 1.66×10^{-2}. The best result achievable, by modifying mesh refinement, numerical schemes and even residual control, produces a drag coefficient of 1.79×10^{-2}, which is a 7.5% error. Moreover, obtaining this result was computationally expensive, requiring three hours on 32 CPUs with 100 GiB of memory. Reducing computational resources led to further increases in the drag coefficient.

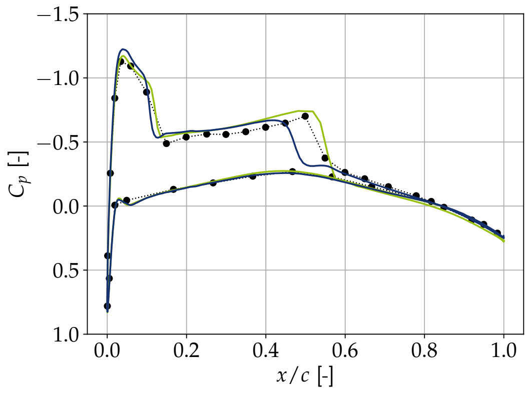
While the drag coefficient discrepancy is within a reasonable range compared to NASA's numerical data [12], the primary issue with DAFoam lies in its inability to accurately capture shock waves. These shocks were either shifted towards the leading edge or entirely absent. The pressure coefficient curves Figure 1.2, particularly at station $\eta = 0.8$ (Figure 1.2d), clearly show the missing shocks. For comparison, the pressure coefficient curves obtained with the CFD setup used in this thesis are shown in Figure 2.13.

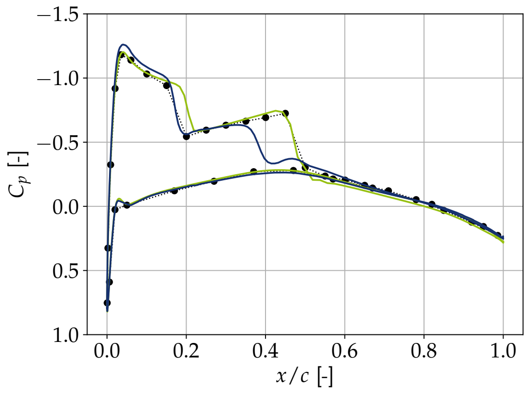
Another challenge with DAFoam is its use of a custom solver, *DARhoSimpleCFoam*, a variant of *rhoSimpleFoam*. Ensuring the reliability of this solver is not straightforward, especially since it lacks some of OpenFOAM's built-in utilities (such as tools for obtaining y^+ or τ_w) that are essential for verifying and debugging results.

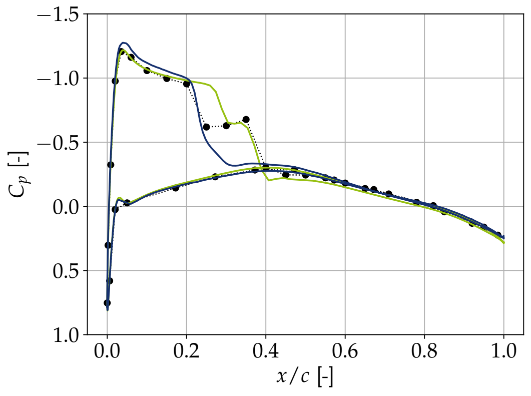
Integrating dimensionality reduction methods within the DAFoam framework could have produced a highly impactful scientific report and saved time for exploring more advanced dimensionality reduction techniques. It is possible that DAFoam functioned well under previous conditions or could still be effective with the right adjustments. Additionally, more experienced CFD and OpenFOAM users might be able to identify and rectify the underlying problems. As a general guideline, the use of DAFoam should therefore be carefully analyzed, and its results compared with reference data, before using it for data processing or optimization purposes.

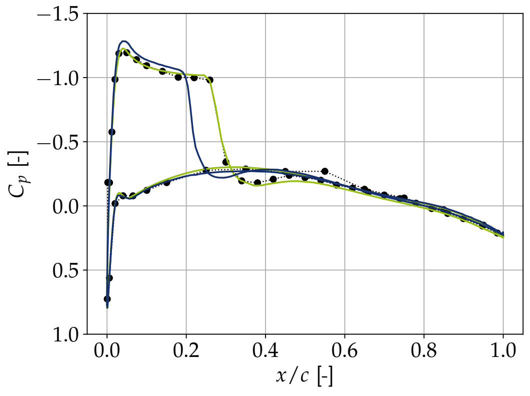


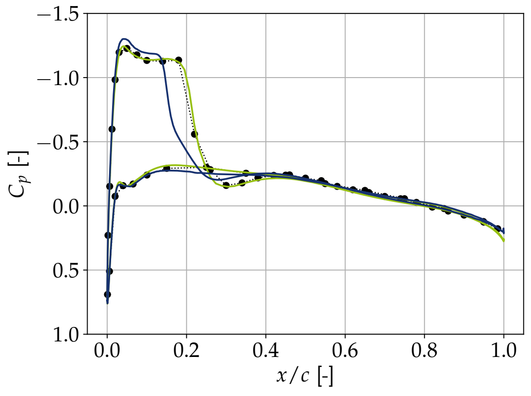
(A) C_p at span station $\eta = 0.2$.


(B) C_p at span station $\eta = 0.44$.


(C) C_p at span station $\eta = 0.65$.


(D) C_p at span station $\eta = 0.8$.


(E) C_p at span station $\eta = 0.9$.


(F) C_p at span station $\eta = 0.96$.

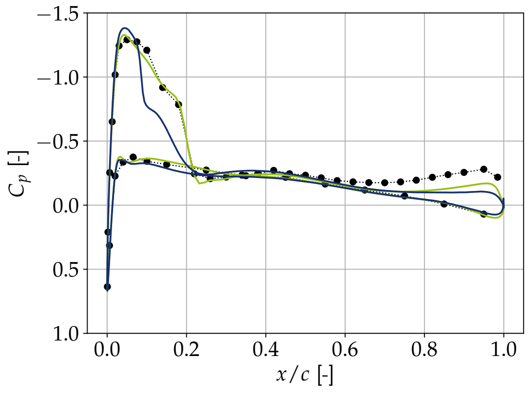

(G) C_p at span station $\eta = 0.99$.

FIGURE 1.2: C_p comparisons between DAFoam and reference data.

Chapter 2

Computational Fluid Dynamics

This chapter aims to present and describe the numerical simulation. Computational fluid dynamics (CFD) allows to approximate the behavior of the interaction between a fluid and a given body by solving the governing laws of motion of fluids. These numerical methods are a key step in every aeronautical (and other sector) manufacturing process. Even if those methods have strong limitations due to the amount of approximations, they are useful to guide and help designers find the best designs. It must be noted that those numerical simulations are less expensive (depending on the accuracy required) than experimental testing. CFD is a great tool but it cannot be used as a black-box, the user must ensure the theoretical validity of the simulation, and every CFD output experimentally validated. CFD is used to approximate complex phenomena but it cannot ensure a perfect accuracy. Every CFD user must keep in mind that the CFD output might be very far from experimental observations.

This chapter will describe the whole simulation performed to obtain the wing performances. There are different types of CFD simulations, but this research will only focus on solving the Reynolds-Averaged Navier-Stokes (RANS) equations. More advanced numerical computations can be conducted but are not adapted for the current presentation.

2.1 CFD: usage and limitations

During CFD simulations a trade-off between accuracy and computational cost must always be made. The goal of this work is to optimize the performances of a wing. This can only be achieved by giving the optimizer an extended database, called a Design of Experiment (DoE). For the record the DoE, described in chapter 4, is composed of about 800 wings. Due to this huge amount of samples, and due to the limitation in time and computational resources, the overall accuracy had to be reduced.

Another point that should be mentioned is that the Onera M6 wing is a transonic wing. Therefore there are shockwaves that will develop on the extrados. This is a complex phenomenon that can generate errors if the models are not wisely selected. Moreover, at such high speed the flow becomes turbulent, particularly in the boundary layer, and is no more uniformly laminar. The presence of turbulence is another source of errors in the CFD simulation as turbulence is one of the most complex phenomena. Nowadays there are no perfect models that could approximate these chaotic and randomness flows. Therefore a major source of the global errors might come from these approximated turbulence models.

To evaluate the required lift and drag forces, it could be useful to explain in a few lines how they are computed. The mesh is composed of cells and nodes. The job of the CFD solver is to solve the RANS equations within each cell. At the end of the whole iteration process of the solver, the relevant fields are available. Amongst them are the pressure and the wall shear stress fields. From those values, it is possible to extract the two relevant forces for this work: lift and drag. These are computed by integrating the pressure and the shear stress distributions over the surface of the wing. The pressure force contribution can be computed using,

$$\mathbf{F_p} = \int_S p\mathbf{n}dS \,. \tag{2.1}$$

The viscous force contribution is obtained by using

$$\mathbf{F}_\tau = \int_S \tau dS \,. \tag{2.2}$$

Bringing these equations together, the total force acting on the wing can be expressed:

$$\mathbf{F} = \mathbf{F_p} + \mathbf{F}_\tau \,. \tag{2.3}$$

From this, the lift and drag forces can be obtained using their respective unit vector direction.

$$\mathbf{L} = \mathbf{F} \cdot \mathbf{e_L} \tag{2.4}$$

$$\mathbf{D} = \mathbf{F} \cdot \mathbf{e_D} \tag{2.5}$$

2.2 Computational resources

This thesis is part of an aero-structural demonstrator developed in Cenaero [13], a Belgian applied research center. Thanks to Cenaero the simulations were performed on their high performance computing resource: Lucia [14]. The supercomputer is composed of 300 CPU and 50 GPU nodes. The final baseline case runs in 30 minutes on 32 CPU cores, using less than 15 GiB of memory. This case can runs on a personal computer, however it might take a lot more time. For information, Figure 2.1 shows the computational time evolution as a function of the number of CPU used.

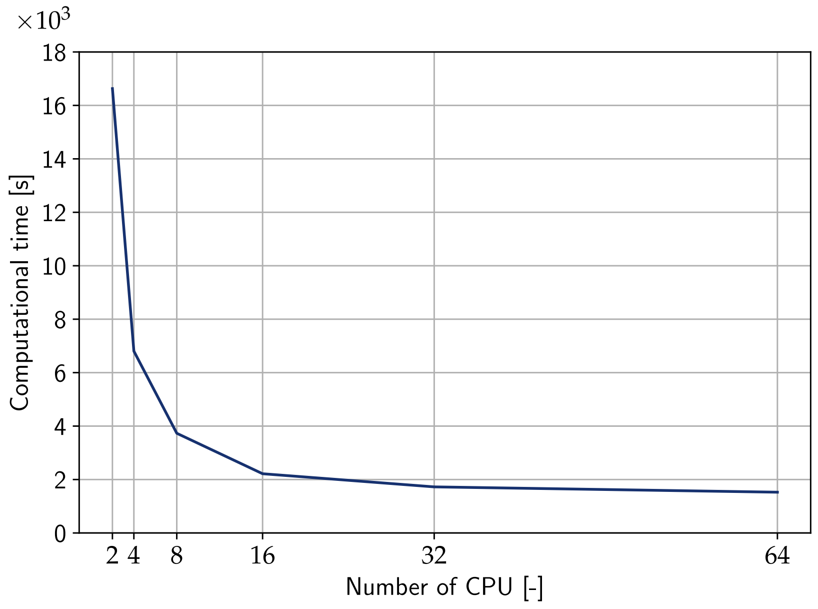


FIGURE 2.1: Computational resources required to run the baseline case.

2.3 Physical and computational domain

The goal of the CFD analysis is to obtain the lift-to-drag ratio of the Onera M6 wing that will be used later for the optimization process. Beside this measure of the performances, other quantities can be obtained from the simulation (pressure fields, velocity fields, etc.). The optimization part, chapter 5, focuses on a single operating point, that is described later on. For that reason, quantities like Mach number, angle-of-attack or Reynolds number remain constant. The description of this operating point is described in this section, but in short, it is a transonic simulation at ground atmospheric conditions.

There are several assumptions that are made to simplify the problem. One of them, is that only the flow over the half wing is modeled and there is no interaction taken into account between the fuselage of the plane and the wing. Moreover the wing is assumed to have no ailerons, spoilers, slats or flaps. The potential aero-elasticity, deflection and bending of the wing are not taken into account. This work is made assuming a steady-state behavior (i.e., there is no time dependence). This allows a lighter computational cost and makes the wing comparisons easier, as only the fully developed flow is studied. The fluid is air, which is considered as a calorically perfect gas at room temperature.

One of the main characteristics of this wing is that it is designed to fly at transonic speed. The value of the Mach number, defined as the ratio between flow velocity and speed of sound, is 0.8395. With this value, compressibility effects cannot be neglected as the Mach number is higher than the commonly used 0.3 threshold. These compressibility effects will be illustrated through the formation of multiple shocks on the extrados of the wing.

Another point to describe the flow is through the Reynolds number. In this work is has a value of $11.72 \cdot 10^6$ based on the mean aerodynamic chord (Table 2.2). With such high values, a transition from laminar to turbulence occurs in the boundary layer. This transition will interact with the shocks, and must be correctly modeled.

The last element describing the operating point is the angle-of-attack that has a strong influence over the wing performances. A value of 3.06 degree is chosen, and will remain unchanged throughout the whole analysis. This is a key parameter as it has a strong impact on the lift and the drag. The value here is not the one leading to the maximal lift-to-drag ratio. For practical implementation reasons, the chord remains aligned with the x-axis, and it is the incoming flow that changes direction. Therefore all the figures in this thesis will be aligned with the x-axis (the airflow coming slightly from the bottom). However this is only for simplification the implementation process but does not change the behavior of the fluid-body interaction. These three flow conditions are gathered in Table 2.1

TABLE 2.1: Flow conditions from NASA[12].

Dimension	**Value**
Mach number (M) [-]	0.8395
Reynolds number (Re) [-]	$11.72 \cdot 10^6$
Angle-of-attack (AoA) [°]	3.06

The Onera M6 wing is a famous wing that is well documented. It is a quite small wing that can fit inside a wind tunnel making it very attractive for validation. The wing was designed based on the airfoil represented in Figure 2.2. This is called the Onera D profile, that is here non-dimensional. This profile can be found by slicing the Onera M6 dorsal view (Figure 2.3) using a 30°angle. For more details, the original Onera's drawing can be found in Appendix A.

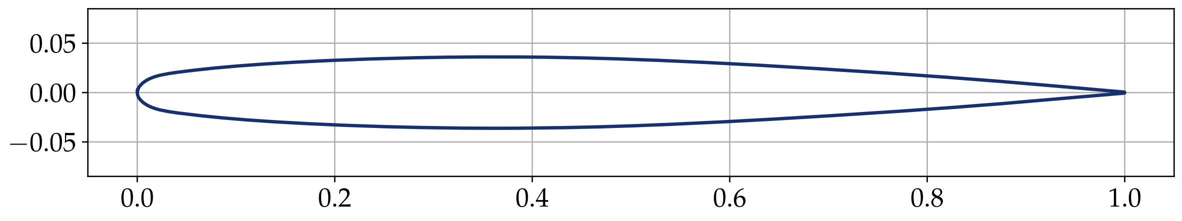


FIGURE 2.2: Onera M6 wing 'D' section.

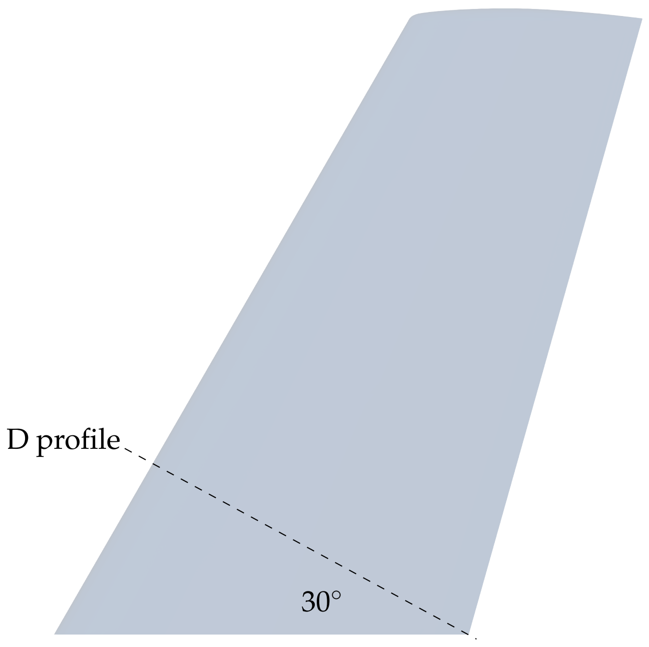


FIGURE 2.3: Dorsal view of the Onera M6 wing, with the D profile cut.

All the characteristics of this wing can be found in Table 2.2.

2.4 Turbulence model

This simulation is performed at a really high speed and the appearance of shockwaves is expected as previously explained. Moreover, at such high velocity, the boundary layer will be turbulent. It is therefore needed to use equations that can approximate the turbulence. The most suitable turbulence model for the Onera M6 wing, is the well known Spalart-Allmaras [15], that is used in number of referenced cases including the NASA Turbulence Modeling Resource [12] that will be used to asses the validity of the results. The simulation is thus performed in a fully turbulent regime.

TABLE 2.2: Onera M6 wing dimensions according to Onera[8] and NASA[12].

Dimension	**Value**
Mean Aerodynamic Chord (MAC) [m]	0.64607
Aspect Ratio (AR) [-]	3.8
Taper Ratio (λ) [-]	0.562
Sweep Ratio ($\Lambda_{0.25}$) [°]	26.7
Semi-Span (b) [m]	1.1963

2.5 Numerical methods

2.5.1 OpenFOAM

The current study utilizes the well-known OpenFOAM [9] CFD software. The main reason is that it is free and open source. However, the main drawback is that it is low level (only terminal interaction) and the absence of graphical interface might be a source of wasted time for unfamiliar users, as errors might be hard to catch and understand.

Numerous solvers are implemented in OpenFOAM and choosing the appropriate solver can be tricky. To determine the best solver, one needs to highlight the key property of the studied case. Here the goal is to perform a steady-state simulation of a compressible and turbulent flow. The solver that meets those criteria is *rhoSimpleFoam* that is based on the *SIMPLE* algorithm. SIMPLE stands for Semi-Implicit Method for Pressure-Linked Equations. This algorithm is an iterative process that is inherently segregated. In fact, the momentum equation is solved first and injected into the pressure equation. Once the latter is derived, the new pressure is used to solve a second time the momentum equation. Finally the turbulence momentum equations can be solved. This whole procedure is explained by Sim-flow [16]. It would have been more efficient to use a solver that is coupled, where all the equations are solved simultaneously, but there is no such solver available within OpenFOAM library adapted to this problem.

2.5.2 Numerical schemes

The selection of numerical schemes is a critical factor in the accuracy and stability of a CFD simulation. The goal is to strike a balance between precision and robustness, avoiding the pitfalls of numerical oscillations and instability while capturing the essential physics of the flow.

In this simulation, the flow is assumed to be steady-state, which simplifies the governing equations by eliminating the time derivative $\frac{\partial}{\partial t}$. The spatial gradients within the domain are computed using the Gauss Gradient Theorem (Equation 2.6), a fundamental principle in vector calculus that converts a volume integral into a surface integral. This approach is coupled with a second-order accurate linear interpolation scheme, ensuring a higher degree of precision in capturing the variations within the field.

$$\int_V \nabla p \; \mathrm{d}V = \oint_S p \; \mathrm{d}\mathbf{S} \tag{2.6}$$

The convective terms, which describe the transport of quantities like momentum, energy, and species concentration due to fluid motion, are approximated using the Gauss Divergence Theorem (Equation 2.7). This theorem transforms the volume integral of a divergence into a surface integral, making it a powerful tool for CFD. The interpolation schemes applied to these terms vary depending on the specific field. For instance, the convective flux of velocity employs a second-order accurate linear upwind scheme, which combines the accuracy of central differencing with the stability of upwind schemes. Meanwhile, fields such as energy and enthalpy are discretized using the first-order accurate upwind scheme. The later is less accurate but highly robust as it avoids spurious oscillations in regions with steep gradients.

$$\int_V \nabla \cdot (\mathbf{u}\mathbf{u}) \; \mathrm{d}V = \oint_S \mathbf{u}\mathbf{u} \cdot \mathrm{d}\mathbf{S} \tag{2.7}$$

Viscous terms, which represent the diffusive transport of momentum due to viscosity, are discretized using a second-order accurate linear scheme. This choice ensures that the momentum diffusion is captured with high fidelity, preserving the smoothness of the solution without introducing artificial diffusion.

Laplacian terms, commonly arising in diffusion equations and the pressure Poisson equation, are discretized using the Gauss theorem with a second-order accurate linear corrected scheme. This approach accounts for mesh non-orthogonality, ensuring that the solution remains accurate even on complex or distorted meshes.

A summary of the numerical schemes applied in this CFD simulation is presented in Table 2.3.

TABLE 2.3: Summary of numerical schemes.

Term	**Scheme**	**Order of accuracy**
Time derivative	`steadyState`	-
Gradient	`Gauss linear`	Second-order
Convective term for velocity	`Gauss linearUpwindV grad(U)`	Second-order
Convective term for energy	`Gauss upwind`	First-order
Viscous term	`Gauss linear`	Second-order
Other divergence terms	`Gauss upwind`	First-order
Laplacian	`Gauss linear corrected`	Second-order
Interpolation	`linear`	Second-order

2.6 Boundary and initial conditions

The boundary conditions are summarized in a schematic diagram in Figure 2.4, where all the faces of the domain are inlet or outlet flows except the front face (in blue) that is a symmetry plane. This symmetry plane acts as a mirror and allows to model the influence of the other wing and the aircraft body. This boundary condition involves zero normal velocity and gradients. The inlet boundary condition assumes a uniform velocity profile, while the outlet condition supposes a zero gradient. The last boundary condition is the wing body that is considered as an adiabatic solid wall, ensuring a no-slip condition, i.e., the velocity at the wall is zero and the fluid sticks to it.

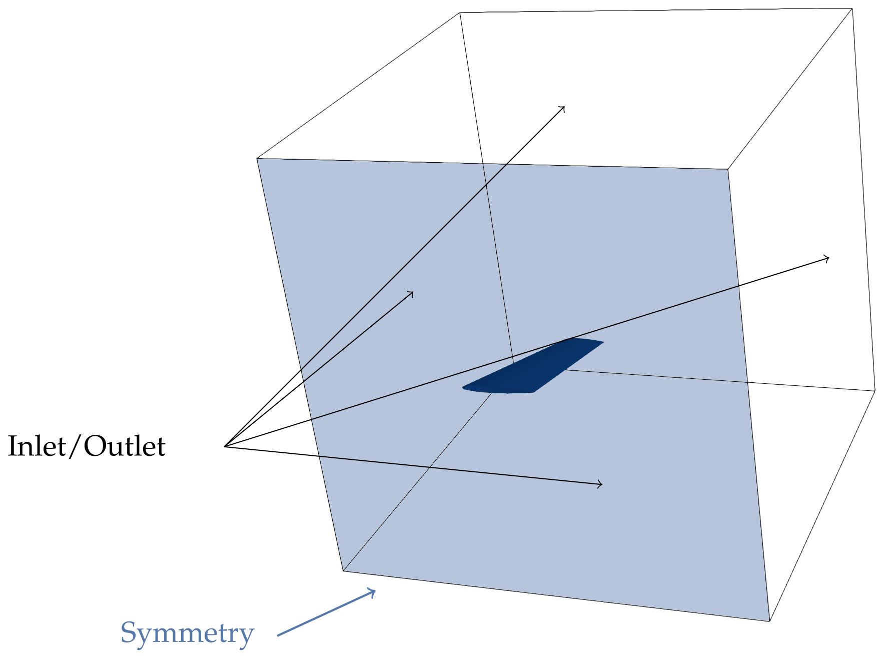


FIGURE 2.4: Representation of the boundary conditions.

The initial conditions are summarized in Table 2.4 and are based on the classic atmospheric conditions of air at room temperature.

TABLE 2.4: Summary of the initial conditions.

Field	**Value**
Pressure (p) [Pa]	101325
Temperature (T) [K]	300
Velocity ($\mathbf{U}$) [m/s]	291.65
Kinematic viscosity (ν) [m^2/s]	$4.5 \cdot 10^{-5}$
Turbulent kinematic viscosity ($\tilde{\nu}$) [m^2/s]	$4.5 \cdot 10^{-5}$
Turbulent thermal diffusivity (α_t) [m^2/s]	10^{-4}

2.7 Generating an appropriate mesh

The mesh is a crucial aspect of the CFD simulation. Indeed if the mesh is too coarse then the results won't be accurate, but if it is too refined then the simulation will be too computationally expensive. The objective is thus to generate a mesh that allows a good accuracy whilst being moderately resource demanding.

The first element to be modeled is the Onera M6 surface wing. There are many different surface meshes available online, as software that can produce this kind of geometry. However to be able to deform the wing, every step must be achieved from command line, and using software is therefore not the best way to automate the meshing process. Therefore the other option is implemented: taking a surface mesh from online resources. The target of this work being the optimization process and the dimensionality reduction, the focus is not on the meshing nor the CFD simulation. Luckily DAFoam [10][11] is an open-source framework developed for this kind of optimization with the automatization required to perform this work. DAFoam is made of several modules that are extracted for this process. The first part of the meshing strategy comes from DAFoam.

First of all, the surface mesh is obtained, and it must be refined to ensure accuracy. But again, *what is good enough?* To determine the best refinement of the surface, it is common practice to perform a mesh refinement study. Before presenting the results of this study, the whole meshing strategy is discussed and described in details. Once the surface mesh has the desired refinement it is deformed following a given parametrization using the Free-Form Deformation method as presented later in chapter 3. Finally DAFoam proposes a method to extrude a 3D mesh based on a given surface mesh. However it required some parameters that also need to be studied. The three most important parameters of this tool are the total height of the mesh H, the number of layers N and the first (adjacent to the wall) layer thickness y_1. These parameters are important because, if not well defined, they will cause a loss of information. These three parameters are linked through the expansion ratio r, that is the ratio at which the thickness of each successive layer increases, i.e., how the the cells grow in the wall normal direction. This ratio is defined thanks to expression 2.8.

$$H = y_1 \frac{1 - r^N}{1 - r} \tag{2.8}$$

There is no exact value for this parameter to be found in the literature. However in this case r must be greater than 1 as fine grids are required in the boundary layer, while coarser cells have less influence over the response far away from the walls. Depending on the case that is studied the maximal value can vary up to values of approximately 1.2 (from CFD Online forum [17]). For this setup, fine cells are required in the boundary layer to capture the complex phenomena described previously. After a discussion with an experienced CFD user, Pr. Koen HILLEWAERT (ULiège), the value of the expansion ratio should be in the range [1.05 - 1.1]. It could have been possible to implement another strategy in order to have really low cell sizes near the wall and shifting to higher cells far away. Such a strategy is beyond reach for the presented framework.

Before diving into this mesh refinement analysis, some points about the mesh are worth discussing. First of all, the mesh is a non-uniform structured O-type grid. This is illustrated in a very coarse mesh in Figure 2.5.

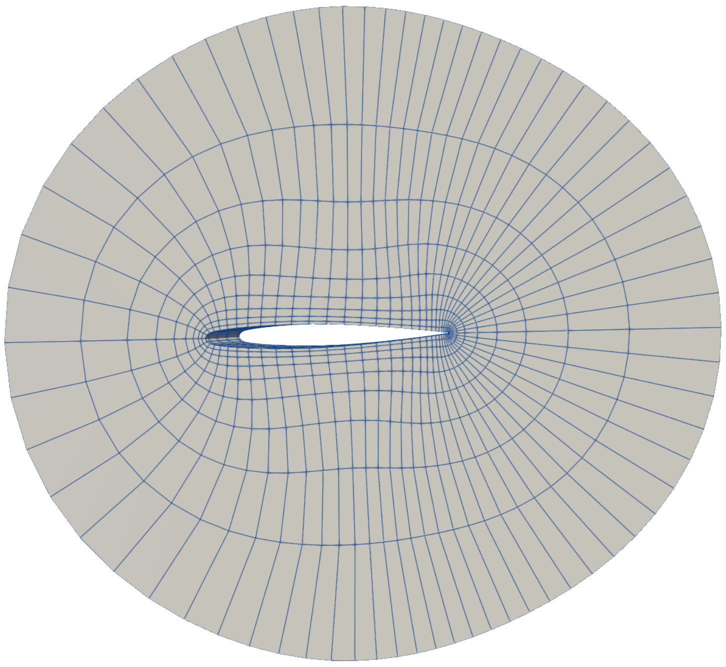


FIGURE 2.5: Illustration of the non-uniform structured O-type grid on a very coarse mesh.

2.7.1 Surface mesh

In this part of the study, four different meshes are compared to each other. Once again the objective is to determine the mesh refinement that performs best whilst being not too computationally expensive. This is summarized in Table 2.5. Note that the very fine case needs more tuning to be able to converge. Here it ran for 4000 iterations and has not reached the residual threshold. Despite not achieving convergence, it provides valuable insights and predictions for this case.

TABLE 2.5: Summary of mesh refinement data.

	Mesh refinement levels			
	Coarse	**Middle**	**Fine**	**Very fine**
Number of cells	216,840	867,360	3,469,440	13,877,760
Number of nodes	223,300	883,260	3,513,580	14,015,820
Computational time (32 CPU) [hh:mm:ss]	00:01:39	00:04:47	00:46:32	06:44:32
C_L	0.2725	0.2720	0.2712	0.2709
C_D	0.0368	0.0222	0.0185	0.0180
C_L/C_D	7.40	12.27	14.68	15.07

These results can also be plotted to have a better view of the convergence. This is done in Figure 2.6, where both C_L and C_D are represented.

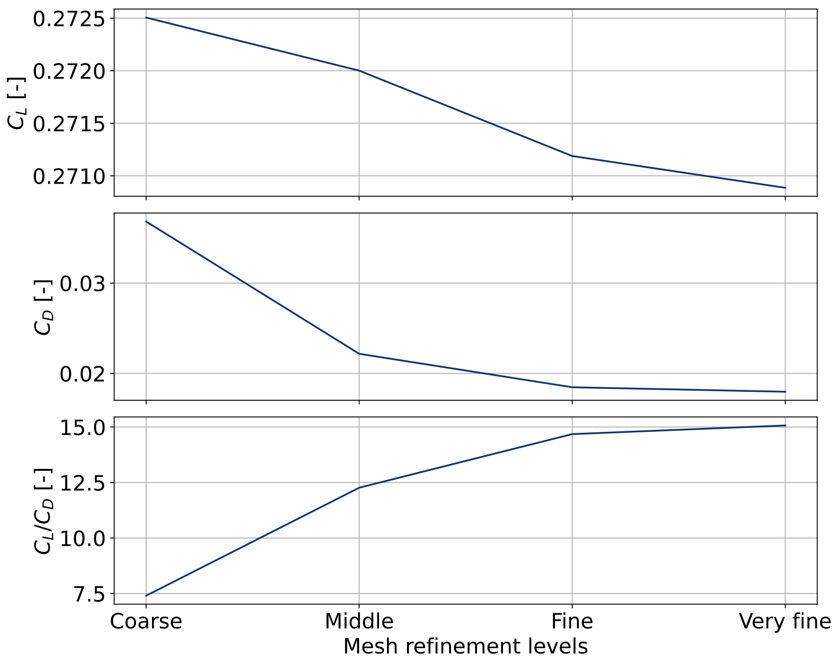


FIGURE 2.6: Quantities of interest depending on the mesh refinement level.

The conclusions are quite straightforward. On one hand the coarse and middle meshes are not sufficiently accurate. On the other hand, the very fine mesh is too computationally expensive. Therefore the fine mesh is the one retained to perform the optimization.

2.7.2 First layer thickness

The first layer thickness is a parameter that can be tuned to be able to capture more information in the boundary layer. However it is more common to analyse its influence through y^+ the dimensionless number that characterizes the first layer that is defined as:

$$y^+ = \frac{u_\tau y_1}{\nu} \tag{2.9}$$

Where y_1 is the first layer thickness, u_τ is the friction velocity and ν is the kinematic viscosity of the fluid. Note that y^+ is an output of the CFD simulation and there exist empirical formulas to predict the y^+ that can be used for the first run. Every CFD user aims for a value of y^+ that is lower than 1. In fact, the velocity near the surface is much lower than the free-stream velocity. It is therefore required (in this kind of application) to have very thin cells near the walls to accurately capture the velocity profile. However such a low value increases the required computing resources. Another strategy is to aim for a larger y^+ and apply correction methods to approximate this complex velocity profile near the wall. This allows to reduce computational resources but the price to pay is a loss of accuracy. A trade-off must again be made but the complexity of the flows forces to use very thin cells as it is shown hereafter. Moreover there are three important zones: the viscous sublayer ($y^+ \lessapprox 5$), the log-law region ($y^+ \gtrapprox 40$) and the in-between buffer layer ($5 \lessapprox y^+ \lessapprox 40$). The intent is not to dive into details but it must be said that every CFD user should avoid the buffer region[18] because this zone might not be accurately modeled. The two other zones are fine but in the log-law region, the use of wall function is required and may lead to errors. Therefore it is best practice to solve governing equations in the viscous sublayer.

For this analysis only the average value of the y^+ is taken. Indeed, there is one y^+ value for each cell. The average value is a good point of comparison if the maximal value is in a reasonable range. Moreover, the high y^+ cells cannot be located in regions of interest, e.g., the leading edge and the shocks location zone. In Figure 2.7 the plot of the lift coefficient confirms that having y^+ below 1 results in the best performances. The drag coefficient does not behave well, as the value increases in the viscous sublayer and decreases afterwards. The orange zone is the buffer region that should be avoided. Hopefully the lift-to-drag ratio gives a better insight. Indeed two regions emerge: the first one is at $y^+ < 1$ and the other is at $y^+ \approx 64$. Both values appear to give good performances, but for the rest of this study, being in the viscous sublayer is preferable. In the boundary layer complex phenomena take place and must be accurately captured. A plateau behavior is observed when $y^+ < 1$, which shows a stability to converge to the same value. Therefore it is decided to keep a low value of the first layer thickness so that y^+ stays in this zone.

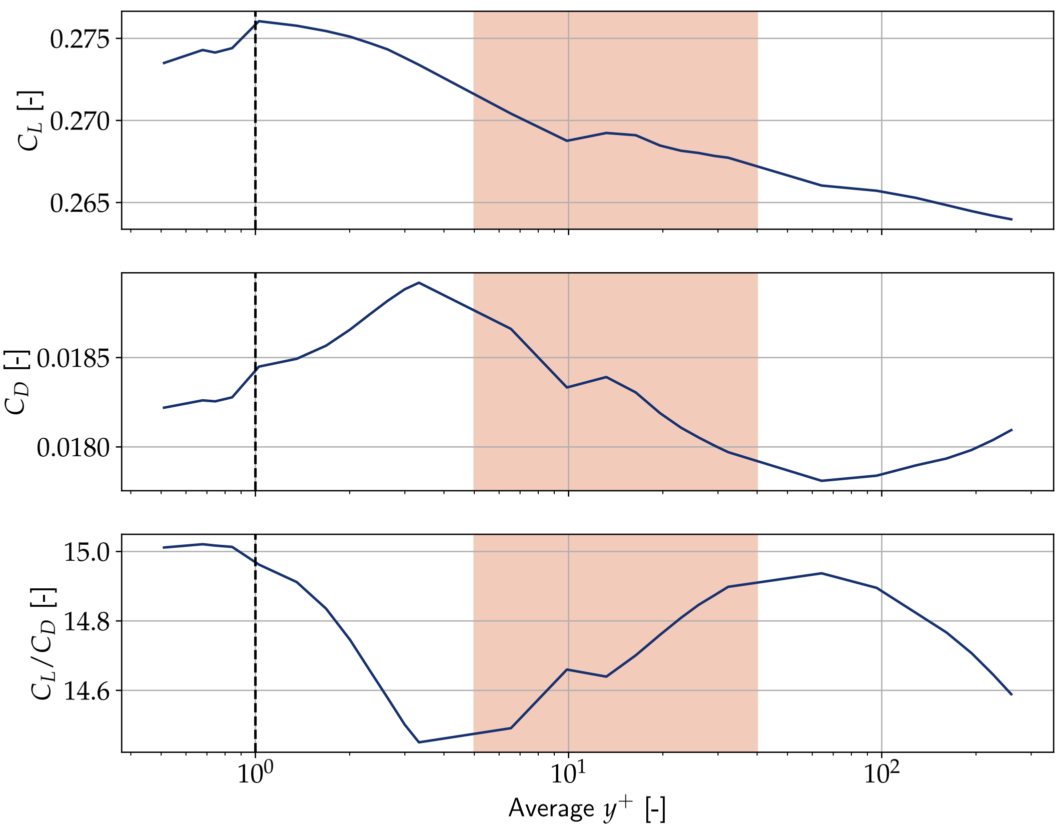


FIGURE 2.7: Evolution of quantities of interest with averaged y^+ values.

A final glance into computational resources confirms the choice to have low y^+. Actually, the case corresponding to $y^+ \approx 64$ takes 30% more time. This can be explained by the high cells size that couldn't capture enough information to converge rapidly. In the plateau zone, the computational time is almost constant which allows to set the first layer thickness with ease.

2.7.3 Number of layers

Taking into account the previous results, this analysis focuses on the importance of the number of layers. The height of the mesh is fixed to a value of 15 times the chord length, which should be reasonable to capture a large part of the far-field effects. With both values of H and y_1 constant, the values of N can be determined by keeping the expansion ration r in its range. The results of this part of the study are summarized in the Figure 2.8.

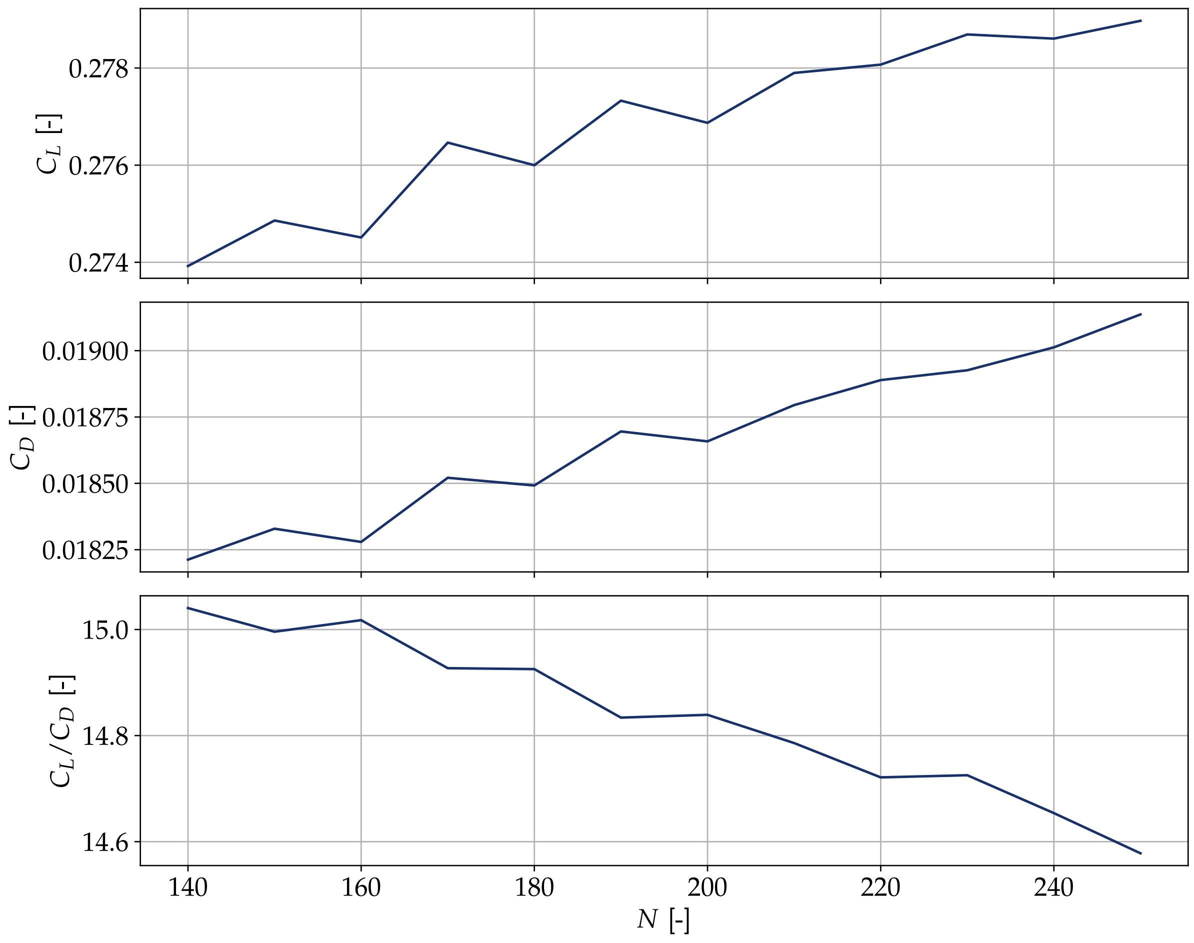


FIGURE 2.8: Evolution of quantities of interest as function of the numbers of layers.

As expected, the more layers there are, the more information is captured. This leads to a lower overall value of the lift-to-drag ratio due to a higher accuracy. However, this has a non-negligible cost. In fact, there is a 250% time increase between $N = 140$ and $N = 250$. This is a huge limitation for the rest of the study. Indeed keeping a high N value would not be affordable for the scope of this study. If one would want to increase the accuracy, a higher value of N would be better. But here, it is decided to reduce the required computational resources by having a low number of layers and therefore a high expansion ratio.

2.7.4 Mesh height

This last part of the mesh refinement study will focus on the meshing distance: the height. The objective is to progressively increase this value to assess the good capture of far-field effects. To be meaningful, this comparison must keep a constant stretching ratio value. In fact, changing the cell size will influence the results. Therefore r is fixed at an arbitrary value of 1.05. The reader must therefor keep in mind that even if the results are presented with increasing height values, the number of layers also vary according to expression 2.8 previously introduced.

Obviously an infinite mesh height would be the most accurate case, but it is not computationally feasible. Again a trade-off must be made between accuracy and computational time. Increasing the height of the mesh leads to a reduced value of the lift-to-drag ratio as can be observed in Figure 2.9 with the height nondimensionalized by the mean aerodynamic chord (MAC). These three curves show a rapid convergence to a constant value without taking into account the bump shape at $H/MAC = 80$. The main conclusion is that the more the mesh height, the more far-field effects are captured, and the more accurate is the setup.

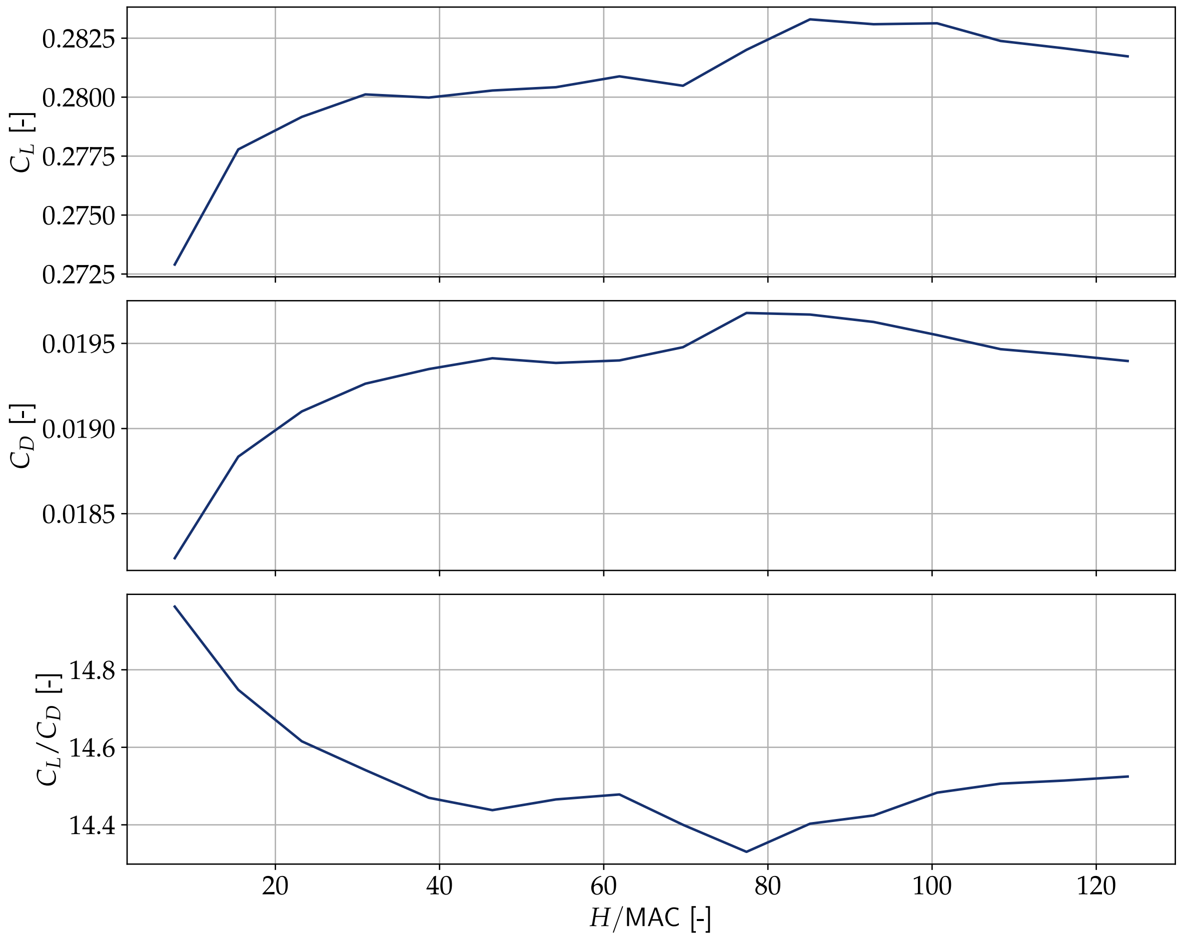


FIGURE 2.9: Evolution of quantities of interest for multiple heights.

An interesting phenomenon is represented in Figure 2.10. When the height increases, the computational time decreases. This might be explained by the fact that the extra information helps to solve the equations.

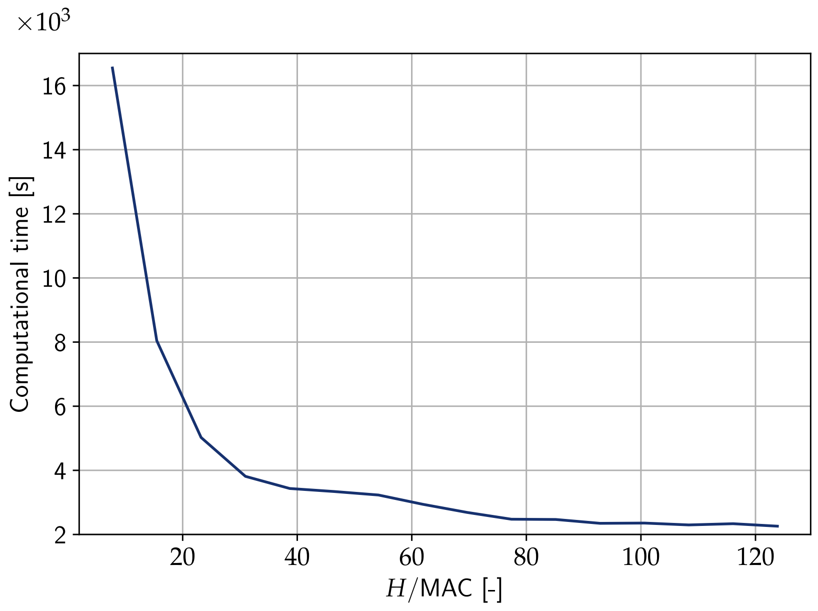


FIGURE 2.10: Evolution of computational time as function of the mesh height.

2.7.5 Conclusion

Thanks to this refinement mesh study, the most suitable configuration can be highlighted. The final mesh that is used in this whole research can now be fully described. Its main characteristics and properties are summarized in Table 2.6. The reader must keep in mind that this setup is selected to give the most accurate results within the allocated budget. Keeping in mind that an optimization process will be applied to this wing and a large number of CFD simulations need to be performed. If one only wants to simulate the baseline case, without being computationally limited, then the setup must be modified to reach a better overall accuracy. Some ideas to improve the setup can be found at the end of this thesis in chapter 6.

TABLE 2.6: Summary of mesh properties.

Property	Value
Grid	non-uniform structured O-type grid
Averaged y^+	≈ 0.67
Number of layers	160
Mesh height	$\approx 122\ MAC$
Expansion ratio	1.1
Number of cells	3,968,640
Number of nodes	4,015,520

2.8 Convergence criteria

Convergence in a CFD simulation is of paramount importance. If the output of the simulation does not converge it is useless as the results cannot be trusted. It is therefore required to specify convergence criteria to ensure the CFD solver does not stop until satisfied. In OpenFOAM this is achieved through the residuals that are a measure of the error. The solver will stop if the residuals drop below a given tolerance. These tolerances are summarized in Table 2.7.

TABLE 2.7: Summary of the convergence criteria.

Field	Value
Pressure (p)	$5 \cdot 10^{-7}$
Velocity (**U**)	$5 \cdot 10^{-7}$
Energy (e)	$5 \cdot 10^{-6}$
Turbulent viscosity $(\tilde{\nu})$	$5 \cdot 10^{-7}$

The convergence of the four fields are represented in Figure 2.11.

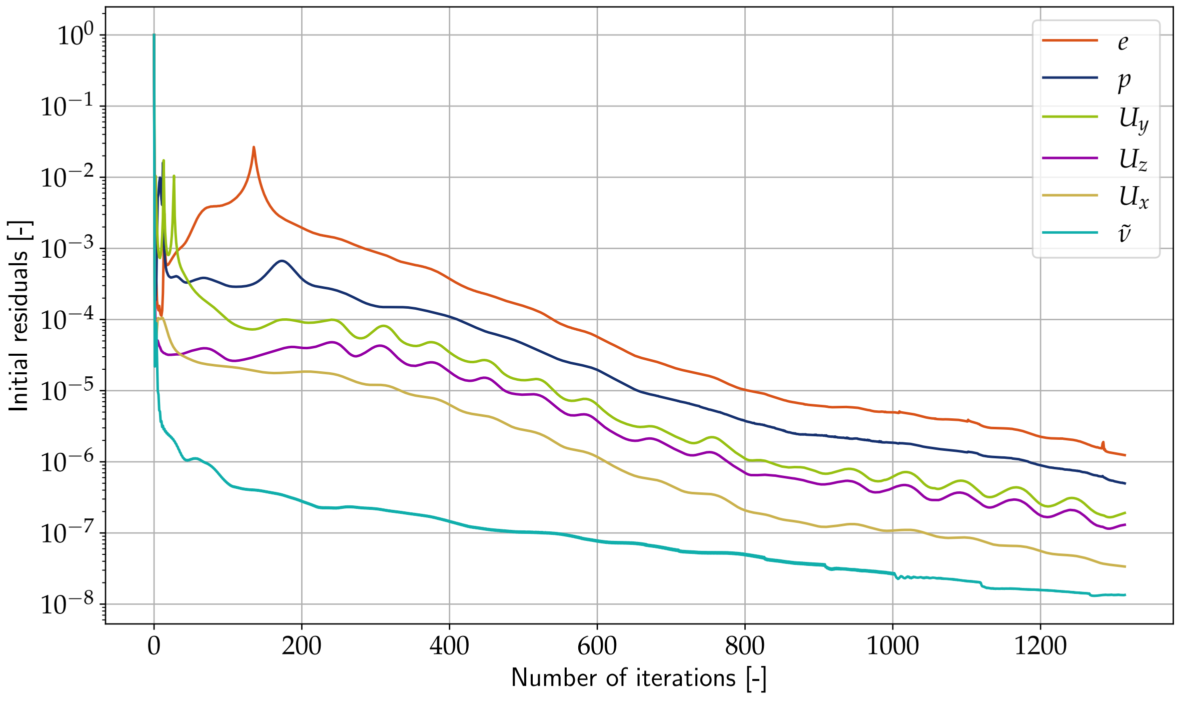


FIGURE 2.11: Convergence of the residuals.

However very low tolerances might not always lead to converged simulations. Therefore analysing quantities of interest help asses the good convergence of the solver. The lift and drag coefficient evolution are shown in Figure 2.12.

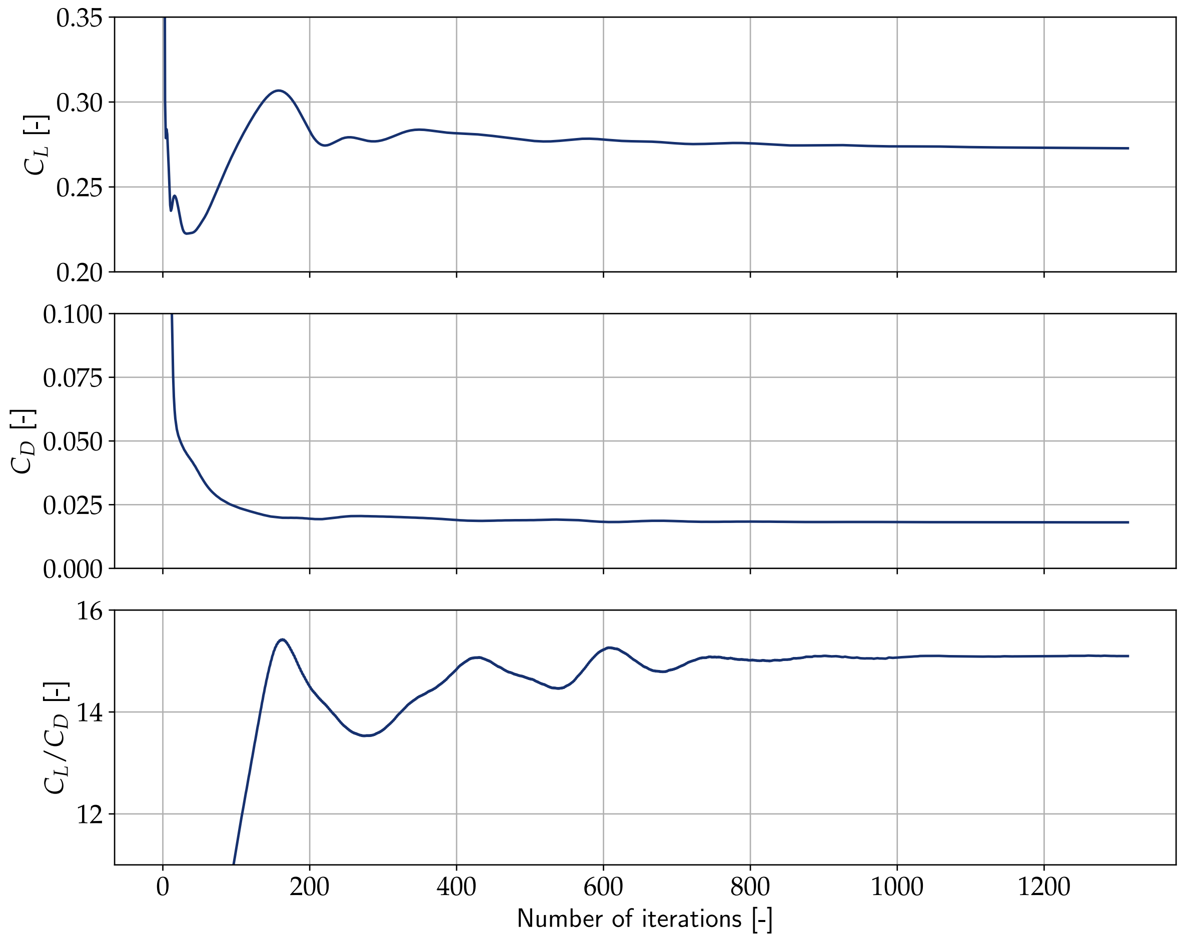


FIGURE 2.12: Convergence of quantities of interest.

The previous figures indicate that the case converged in about 1300 iterations.

2.9 Validation and verification

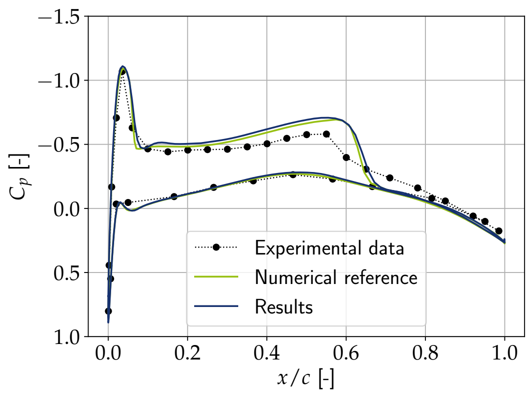
To validate the whole setup, the relevant data are compared to the experimental, from AGARD [19], and numerical data from NASA Turbulence Modeling Resource [12]. The data are available for 7 span stations: $\eta = y/b$. Before diving into the results, the reader must keep in mind that the experimental data cannot be numerically obtained without performing very complex and intensive computation. In fact CFD is only an approximation of the solution. Moreover, the numerical data provided by the NASA won't be matched in this simulation. They performed more advanced simulations, with finer meshes and other solvers and software. The main goal of this whole study being an optimization process, reaching this kind of accuracy is not required and would be too computationally demanding.

First of all, the pressure coefficient C_p can be compared. The main target is to be able to capture the shocks on the extrados. Recall that a positive C_p corresponds to the pressure side, while a negative value is obtained on the suction side. Then the stream-wise skin-friction coefficient can be analysed. Both coefficients are compared to the numerical data. As a reminder, these coefficients are computed based on the free-stream conditions such as:

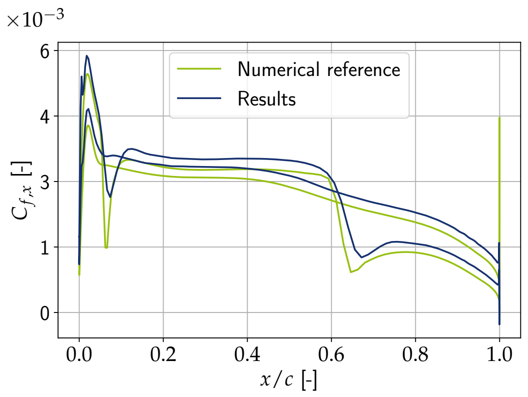
$$C_p = \frac{p - p_\infty}{0.5\rho_\infty U_\infty^2}\ , \tag{2.10}$$

$$C_f = \frac{\tau_\mathrm{w}}{0.5\rho_\infty U_\infty^2}\ . \tag{2.11}$$

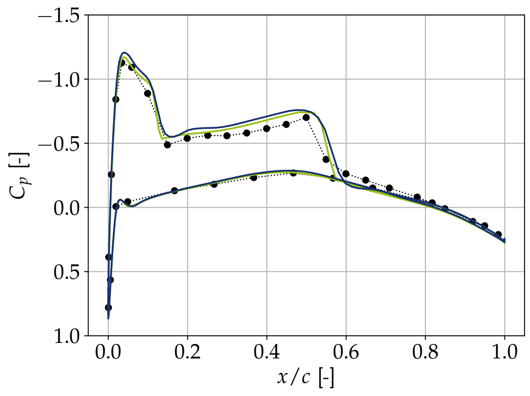
With τ_w the wall shear stress, ρ_∞ the free-stream density and U_∞ the free-stream velocity. The comparisons can be found in the graphs in Figure 2.13.



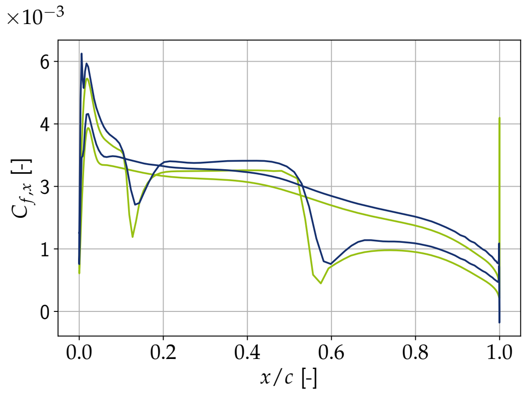
(A) C_p at span station $\eta = 0.2$.



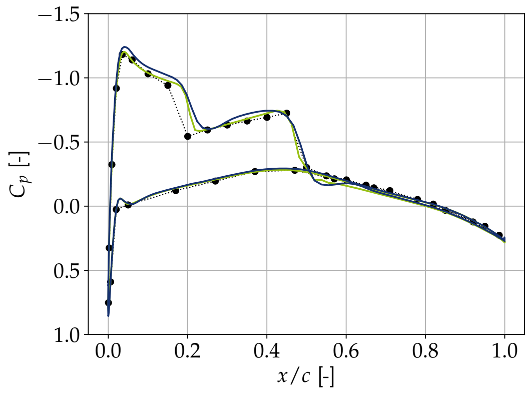
(B) $C_{f,x}$ at span station $\eta = 0.2$.



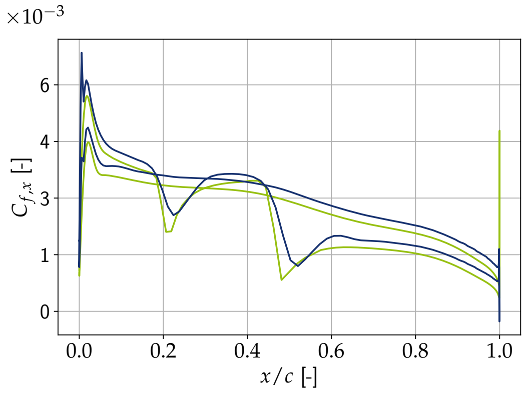
(C) C_p at span station $\eta = 0.44$.



(D) $C_{f,x}$ at span station $\eta = 0.44$.

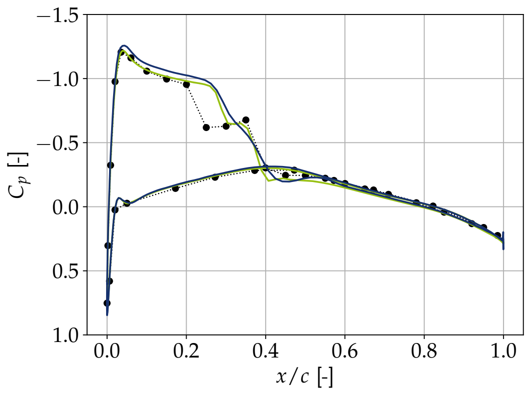


(E) C_p at span station $\eta = 0.65$.

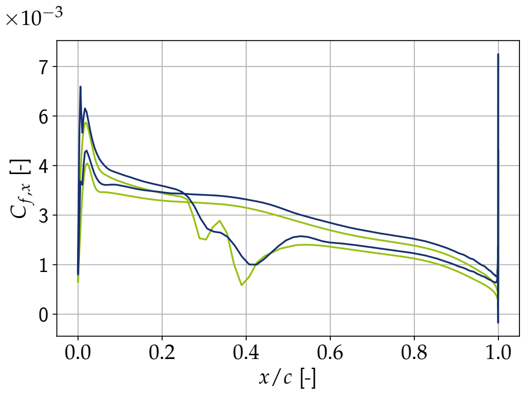


(F) $C_{f,x}$ at span station $\eta = 0.65$.

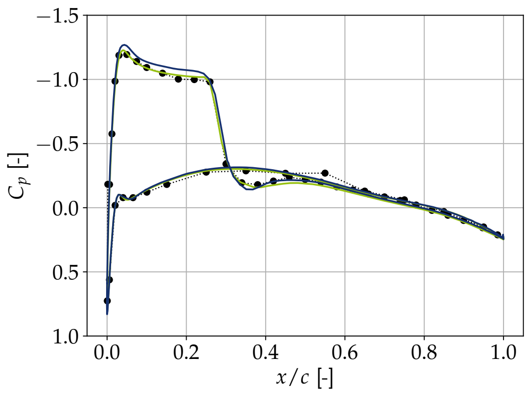
FIGURE 2.13: Comparisons of C_p and $C_{f,x}$ with reference data (Part 1).



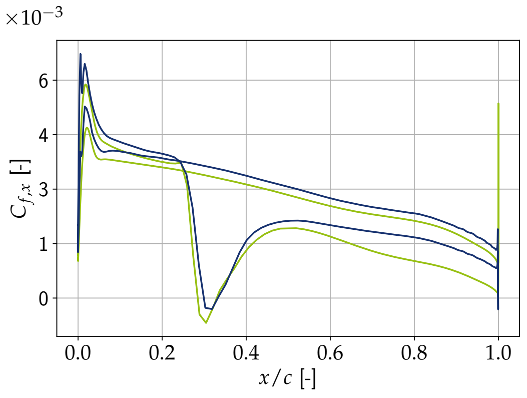
(G) C_p at span station $\eta = 0.8$.



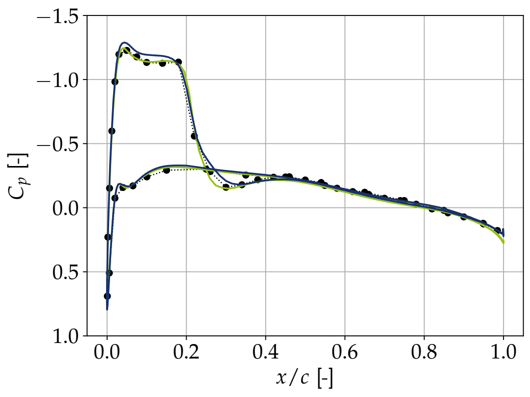
(H) $C_{f,x}$ at span station $\eta = 0.8$.



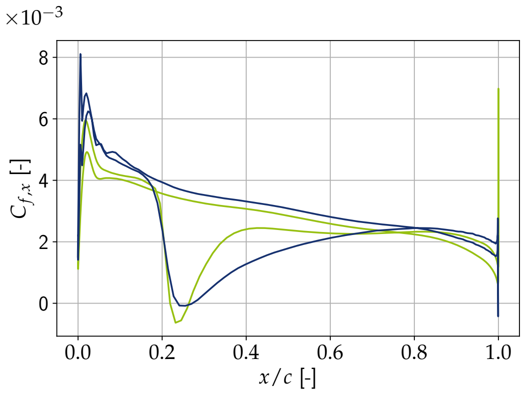
(I) C_p at span station $\eta = 0.9$.



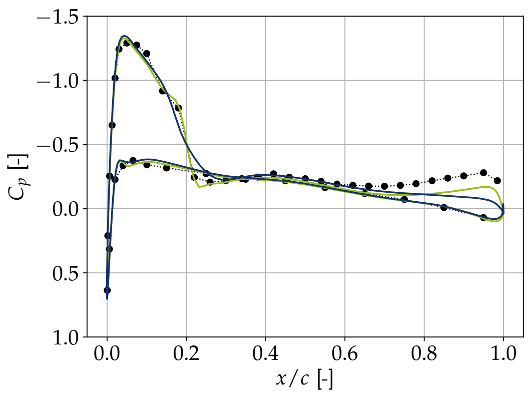
(J) $C_{f,x}$ at span station $\eta = 0.9$.



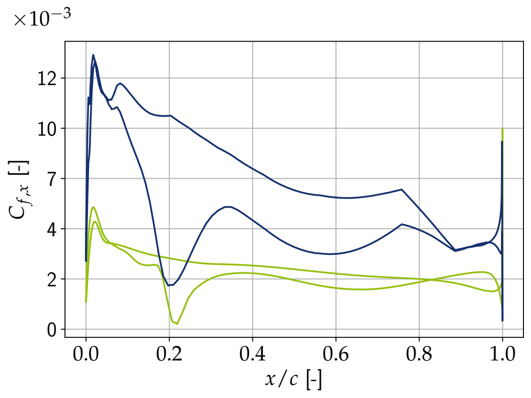
(K) C_p at span station $\eta = 0.96$.



(L) $C_{f,x}$ at span station $\eta = 0.96$.



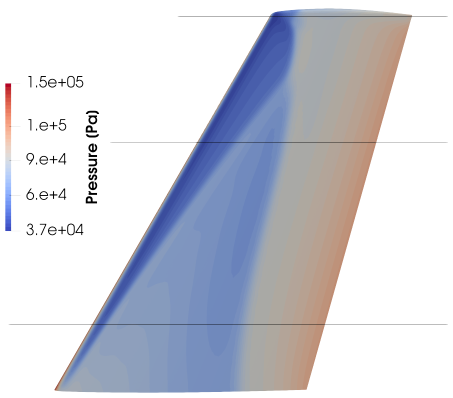
(M) C_p at span station $\eta = 0.99$.



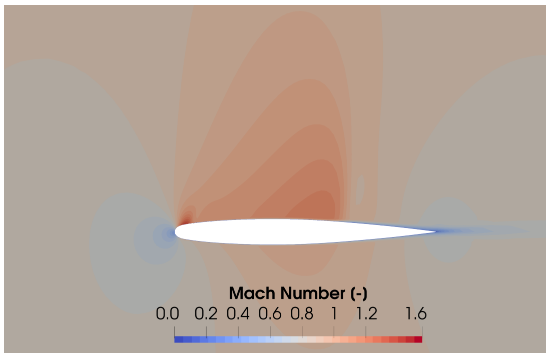
(N) $C_{f,x}$ at span station $\eta = 0.99$.

FIGURE 2.13: Comparisons of C_p and $C_{f,x}$ with reference data (Part 2).

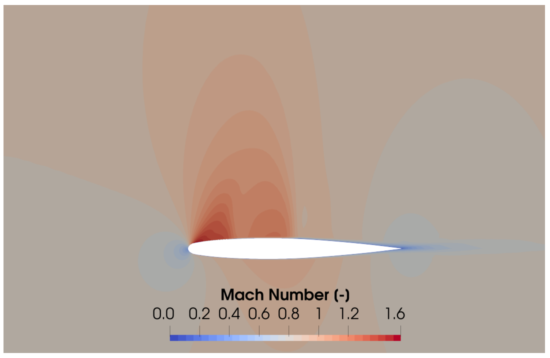
These results are encouraging, as the CFD setup provides results that are reliable. The next aspect that must be verified is the presence of shocks. For this a dorsal view representing the pressure can be found in Figure 2.14a. As expected, a lambda shape can be observed on the extrados which is typical with shocks. Another way that might help visualize the shocks is by representing the Mach number at different span stations in Figure 2.14.



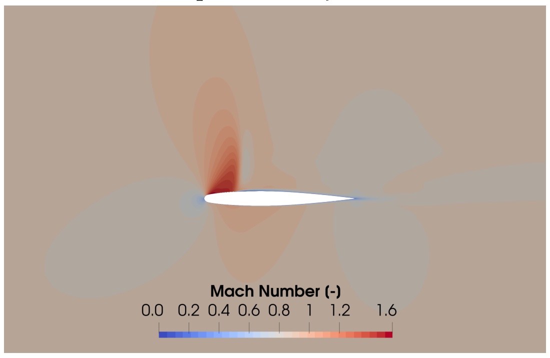
(A) Dorsal view.



(B) Span station $\eta = 0.2$.



(C) Span station $\eta = 0.65$.



(D) Span station $\eta = 0.96$.

FIGURE 2.14: Illustration of the shocks obtained with ParaView [20].

From all these comparisons, the CFD setup can be considered reliable, and will be used for the rest of this thesis. However this setup can still be improved as discussed in chapter 6.

Chapter 3

Free-Form Deformation

3.1 Introduction

Free-Form Deformation (FFD) is a modeling technique that was introduced by Sedeberg and Parry in 1986 [21]. The main goal is to allow solid modeling to deform a given volume. The first step is to define FFD points that will form the FFD volume. The object/volume that the users want to transform should be embedded in the FFD volume. To have a better understanding, one simple tutorial from PYGEO [22] is presented in the followings lines. Figure 3.1 show an example of a cylinder inside a FFD box.

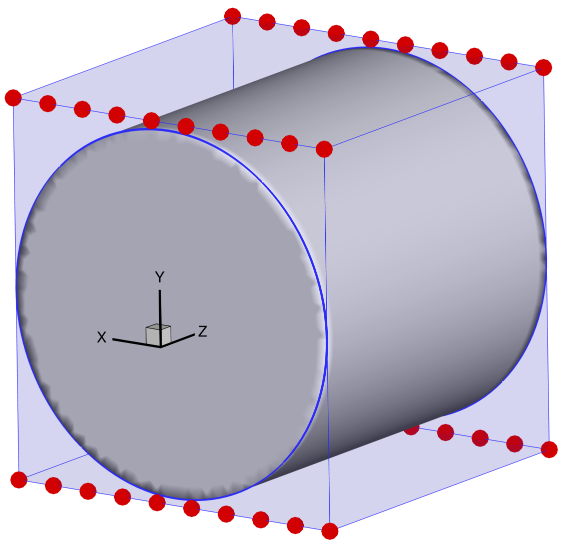
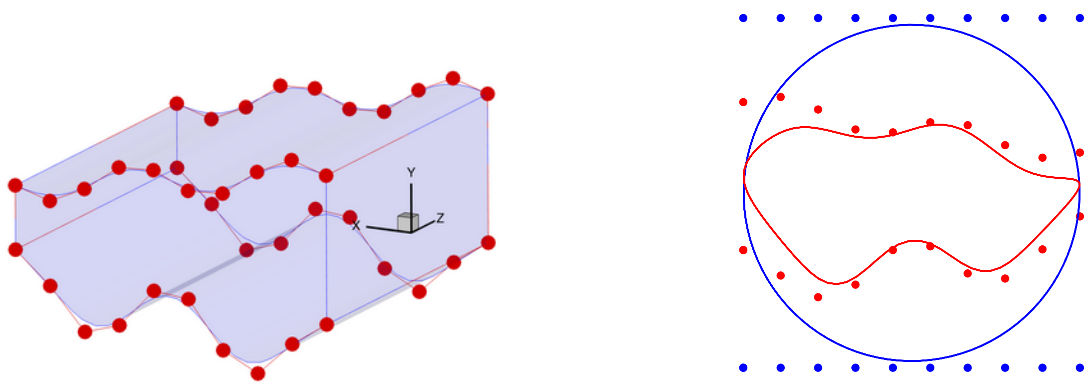


FIGURE 3.1: Cylinder embedded in a FFD box [22].

Now that this cylinder is embedded in a surrounding volume, every perturbation applied to the FFD points (red dots on the Figure 3.1) will be spread to the whole volume, as if it was a block of jelly. FFD focuses on the delta of deformation, in other words, to better understand the method, "FFD volumes parameterize the *geometry change* rather than the geometry itself" as stipulated by Kenway et al. [1]. At this point it is clear that the more FFD points there are, the more the shape modifications become precise and detailed. However it must be noted that even with a small number of FFD points, the cylinder will still be smoothly deformed.

The next step is to define a parametrization that defines how the FFD points will be disturbed. They can move independently of each other or together. This is illustrated in the following example. The cylinder is squeezed with a wave-like shape applied to the FFD points.

This can be seen in Figure 3.2 where the deformed and the original cylinder are represented.



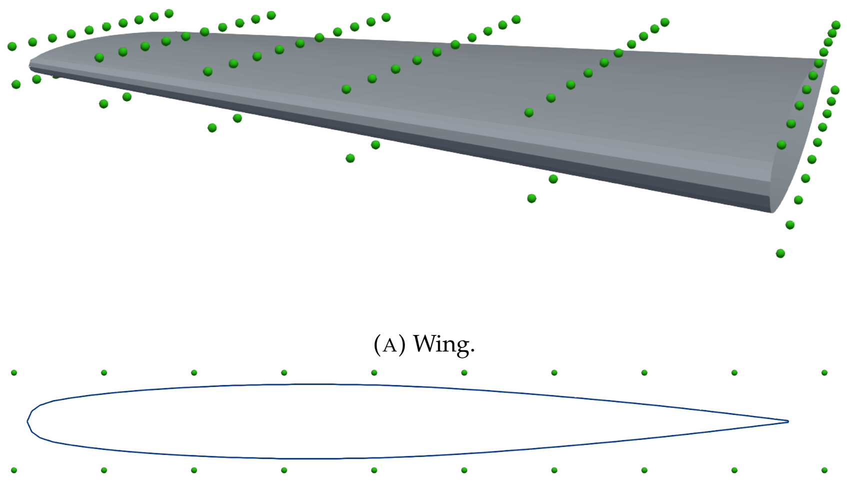
(A) 3D deformed volume. (B) 2D perturbed geometry.

FIGURE 3.2: Application of a more complex deformation on the cylinder. Blue being the original cylinder and FFD points, while red are the deformed ones.

3.2 Applying FFD to the Onera M6 wing

3.2.1 Defining the FFD box

As previously said, the FFD points must enclose the whole wing surface. The number of points for this framework is set to be 120, which is divided in six airfoils containing ten points stream-wise on the extrados and ten other on the intrados. In the purpose of deforming the wing to increase its performances, it is preferable to have more points aligned with the flow direction. This will allow to model the airfoil shapes more rigorously. Figure 3.3 illustrates the FFD points (green dots) that are all around the Onera M6 wing.



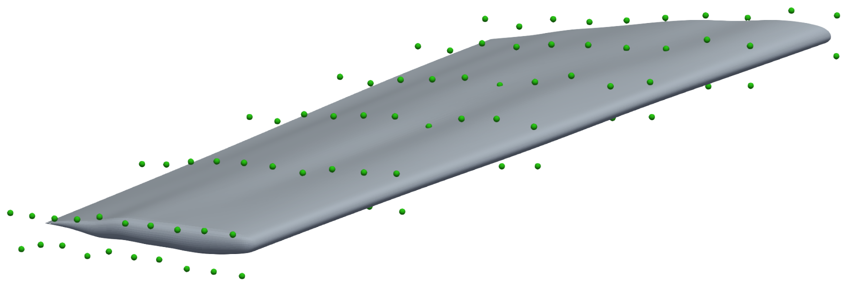
(A) Wing.

(B) Airfoil.

FIGURE 3.3: Baseline wing embedded in its FFD box.

3.2.2 Parametrization

The parametrization can be defined as the set of all geometric variables and their effects on the baseline geometry. In the context of wing geometry, many parameters are known to impact the performances. Amongst them the most known are camber, twist, sweep, thickness, taper, dihedral, chord, span. Those parameters are known by the aerodynamic experts, however choosing only these parameters could influence the optimization. To be able to keep the objectivity, the philosophy here is to let the optimizer find the best design, and eventually put in evidence some of the main aerodynamic wing parameters. Hence the parametrization in this work will remain brute. The choice of the parametrization is therefore quite straightforward: each FFD point can move along the z-axis, which is normal to the chord-span plane. The surface of the wing shouldn't be perturbed above a given limit. Indeed, by imposing a limit on the amplitude of the deformation, the wing surface can remain smooth enough to avoid an abrupt drag increase. This limit on the amplitude is taken to be 10 %. For illustration, one random parametrization is shown on Figure 3.4 (with an increased limit up to 30% for better comprehension).


(A) Perturbed wing.

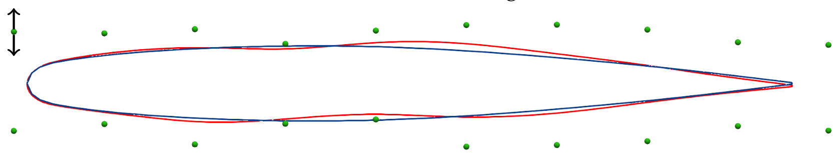
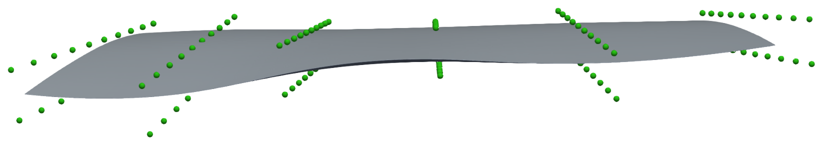

(B) Baseline (blue) and perturbed (red) airfoils.

FIGURE 3.4: Application of the 120 design variables with arrows in the deformation direction.

In order to help the optimizer find a better design, five more design variables are added, as performed by DAFoam, leading to a total of 125 parameters. These additional design variables are the independent twisting of the FFD airfoils. They are all modified except the root airfoil that remains untwisted. The twist is applied along the quarter chord line. This is illustrated on Figure 3.5 where only the five twist design variables are applied. Again a constraint is set to limit the twist amplitude to 2°. Once more, the figures have larger amplitudes for illustration purpose.


(A) Perturbed wing.

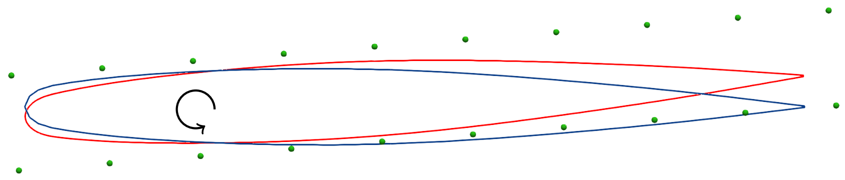
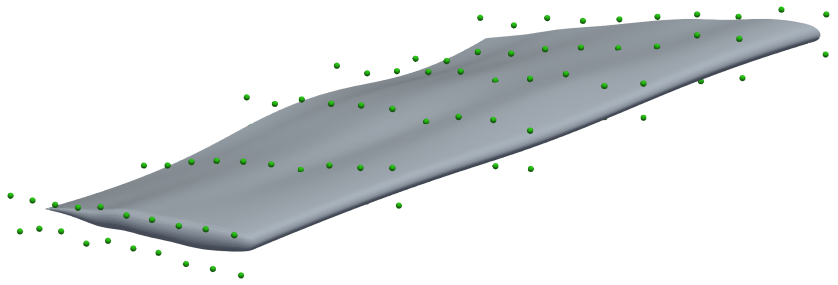

(B) Baseline (blue) and perturbed (red) airfoils.

FIGURE 3.5: Application of the 5 twist design variables.

Bringing it all together, Figure 3.6 show one random perturbed wing using the whole set of parameters.


(A) Perturbed wing.

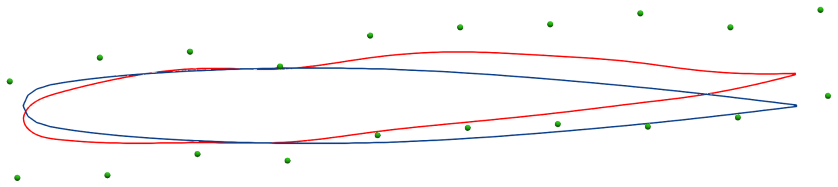

(B) Baseline (blue) and perturbed (red) airfoils.

FIGURE 3.6: Application of the 125 design variables.

Keep in mind that the order in which these design variables are applied is critical: first, the deformation along the z-axis, followed by the twist. Reversing this sequence (twisting first and then deforming) will modify the chord length. To maintain consistency in this study, it is important to preserve certain features, such as the chord, span, and sweep.

In fact, since the flow condition is transonic, sweep significantly influences the velocity the wing experiences and, as a result, affects drag. Although a more detailed optimization could focus on this parameter, it is crucial to maintain the sweep constant. Its impact on the shocks would complicate the high-level parametrization desired here, which focuses on the overall design rather than intricate details.

3.3 Differences with DAFoam

The application of Free-Form Deformation closely resembles the deformations implemented in DAFoam. Both frameworks use 120 variables in the direction normal to the chord-span plane. Additionally, both frameworks incorporate five twisted sections. However, DAFoam also includes a modification for the angle of attack, which significantly impacts performances and optimization. Since this impact is more pronounced on the lift-to-drag ratio compared to the other 125 design variables, it was decided to exclude it from the current study. Furthermore, modifying the angle of attack complicates the design validation process, as it also significantly affects the shocks position and magnitude.

Chapter 4

Dimensionality reduction

Dimensionality reduction is a technique that aims at reducing the number of variables in a dataset while retaining as much relevant information as possible. It is used in various scientific and engineering applications to simplify problems that have many parameters (high-dimensional). This method can improve performances and helps mitigate the *curse of dimensionality*. Indeed, it can be very costly for some regression models to build a metamodel with more than about twenty parameters. Kriging-based metamodels, for example, are known to be exponentially resource-consuming as the dimensionality increases, as described by Ulaganathan et al. [23].

4.1 Main families of dimensionality reduction methods

There are many methods to effectively reduce the dimensionality of a given problem. These methods can be arranged into three main families as presented in Figure 4.1. A brief overview of these families and methods is made to help the reader understand the wide range of applications of such methods as a detailed explanation goes beyond the scope of this study.

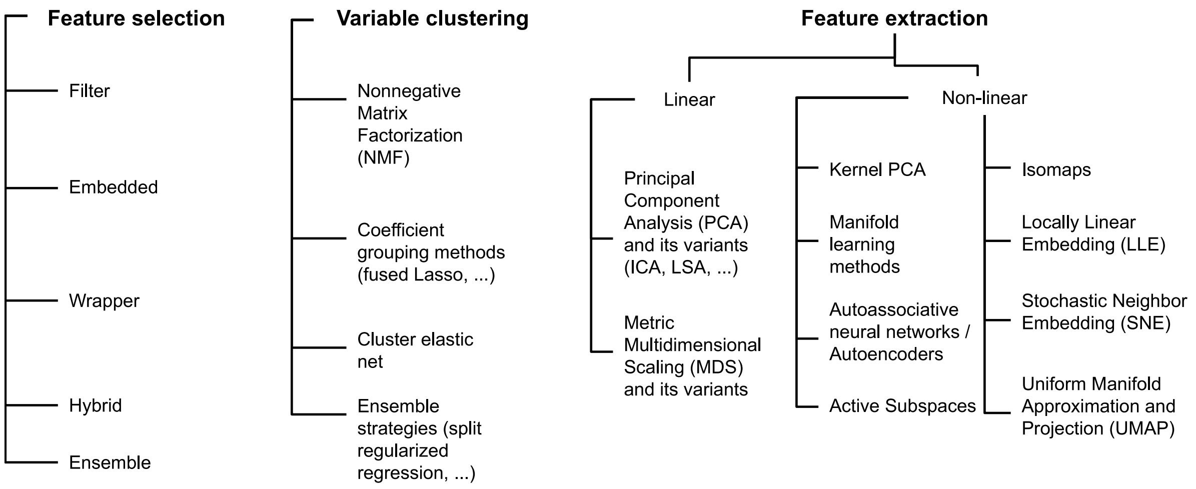

Feature selection
Filter
Embedded
Wrapper
Hybrid
Ensemble
Variable clustering
Nonnegative Matrix Factorization (NMF)
Coefficient grouping methods (fused Lasso, ...)
Cluster elastic net
Ensemble strategies (split regularized regression, ...)
Feature extraction
Linear
Principal Component Analysis (PCA) and its variants (ICA, LSA, ...)
Metric Multidimensional Scaling (MDS) and its variants
Non-linear
Kernel PCA
Manifold learning methods
Autoassociative neural networks / Autoencoders
Active Subspaces
Isomaps
Locally Linear Embedding (LLE)
Stochastic Neighbor Embedding (SNE)
Uniform Manifold Approximation and Projection (UMAP)

FIGURE 4.1: Taxonomy of dimensionality reduction techniques [24].

4.1.1 Feature extraction

The feature extraction methods apply the dimensionality reduction by creating combinations of the original features. These new features, also called components, lie into a lower-dimensional space.

One famous method is the Principal Component Analysis (PCA) [4], that is widely used. It is a linear technique that projects the data onto a new set of axes (principal components), where each axis represents a direction of maximum variance in the data. It allows to reduce the dimensionality of the problem by retaining the most important variance in the data.

Another technique that is more advanced is the Autoencoders [25]. These are a neural network-based methods that compress data into a lower-dimensional space and then reconstruct the original data from this compressed representation. This complexity allows to model non-linear relationships, but is computationally intensive.

4.1.2 Variable clustering

Variable clustering methods implement the dimensionality reduction through elimination of correlated or similar variables. They try to identify groups of similar features in the dataset. This is not an easy task because the clusters are not known in advance, as explained by R. Marion [26]. Once the clusters are defined, it is possible to select representative features from each cluster to compose the new dataset.

4.1.3 Feature selection

The feature selection philosophy is to reduce the dimensionality by choosing a subset from the original data. Therefore the conserved data are not transformed which preserves the most informative features. The irrelevant, or less influential data are discarded.

The Filter method uses statistical techniques to evaluate the importance, or relevance, of each feature independently of the model. This is a method that selects the most influential variables and discard the ones that are considered to have less impact on the response. This means that the features that aren't retained cause a loss of information.

This last method is used in this thesis because it is easy to implement and not computationally resources demanding. Even if this method is not the more complex one, it serves as a pre-processing step, that is, according to Frénay [27], required to create the desired framework. Once this framework is developed, the dimensionality reduction step can be modified to implement more robust, complex and advanced methods that will allow to improve the optimization. This first approach should really be considered as a draft mode, that gives an overview of this tool.

4.2 Applying dimensionality reduction

Before reducing the dimension of the problem, two steps are required: generating a Design of Experiment (DoE) and train a regression model. Once this regression model is ready, the selection process can take place. First, the impact of each Design Variable on the lift-to-drag ratio must be computed. Then the most influential parameters are selected.

Before going into the details, it must be said that the dimensionality reduction process can also be made using a classification instead of a regression model. However this depends on the problem studied. Here the output being continuous, a regression model must be used. If the output was discontinuous, for example *True* or *False*, then a classification algorithm should have been employed.

4.2.1 Generating a DoE

A Design of Experiment (DoE) is a step applied in a wide range of research fields. The main goal of DoE is to generate samples that cover the entire space of parameter values. The DoE can be seen as a relationship or function f between the input variables $\mathbf{X}$ and the output y as in Equation 4.1.

$$f(\mathbf{X}) = y \tag{4.1}$$

The input is the parametrization that allows to deform and perturb the wing surface. The output is the lift-to-drag ratio that is used to characterise the wing performances which is obtained with the CFD simulation.

To comprehensively cover the entire parameter space, a Latin Hypercube Sampling (LHS) method [28] is employed. LHS is a type of stratified sampling that divides each of the dimensions of the design space into non-overlapping subsets, used to obtain random samples covering these subsets more uniformly.

The next question that must be answered is *What should be the size of the DoE?*. This is a complex question that again introduces a trade-off. Indeed, the more samples in the DoE, the more trained the regression model will be. However an infinite number of samples is obviously not achievable due to computational resources. Therefore the objective is to determine a required number of samples that allows building an accurate model that can approximate as the solution within the simulation budget. This depends on the case that is studied as the number of samples is related to the number of parameters. At low dimension one could aim for a DoE that is composed of five to ten times the number of parameters. In this case, the dimension is so high that the DoE must remain close to the lower bound. That is the reason why it is decided to build a DoE of about 800 samples. However this choice is validated later on at the end of section subsection 4.2.2. This DoE is represented in Figure 4.2 where each point represents a deformed wing. The dashed line allows to observe the wings that perform better or worse than the baseline.

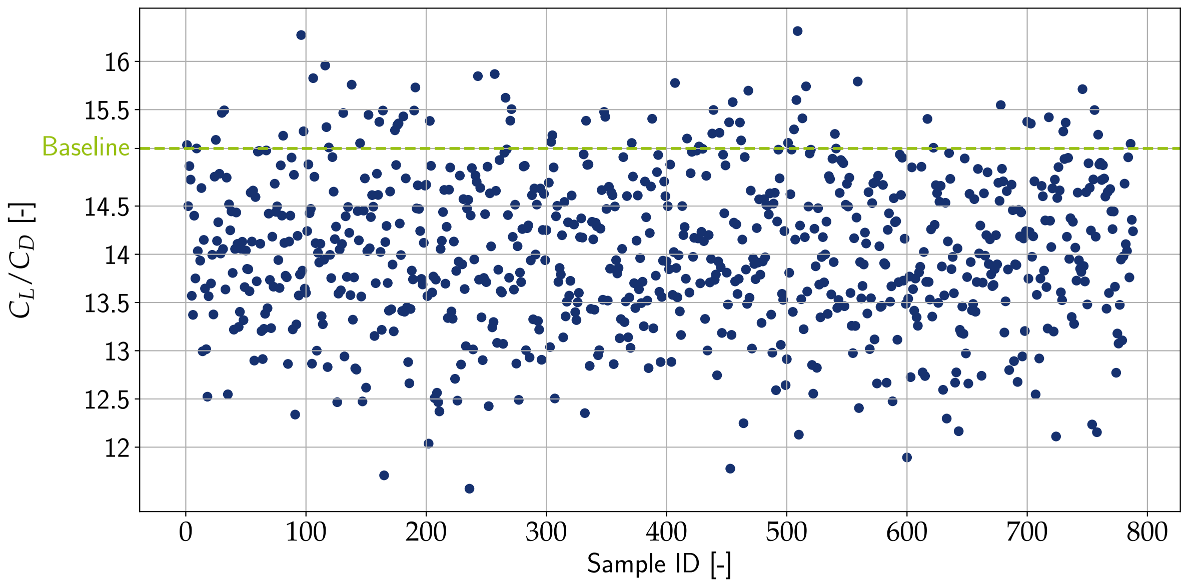


FIGURE 4.2: Illustration of the DoE compared to the baseline case.

The DoE already identifies a wing parametrization that improves the performances up to 8%. The goal of chapter 5 is to push the performances further.

4.2.2 Training a regression model

The regression model has an important impact on the results. Indeed, its goal is to build an inexpensive model to provide an estimate of the output with respect to any input values. In other words, it tries to create a metamodel that will be able to predict the more accurately possible the lift-to-drag ratio based on a given parametrization.

There are many regression algorithms, some linear but others non-linear are more complex. In order to find a good regression algorithm, the models must be compared to each other. This performance aspect is a key parameter of the whole dimensionality reduction process. Indeed, the model must perform well to be able to accurately predict the response. There are various ways to evaluate the performances. One way is to analyse the R^2 value. The R^2 is a measure of the error, 1 being a perfect fit while below the accuracy decreases (note that R^2 can be negative).

To have the more adapted model, the latter should avoid both overfitting and underfitting. Each model relies on different options and parameters that can be tuned in order to avoid both cases. The best parameters can be obtained via a grid search that will highlight the best setup for the model. In this study, the parameters are modified and tested using a cross-validation method. The cross-validation is a well-known method that splits the dataset into multiple training and testing sets. In this analysis, 20 different splits are used. When the model is trained, its predictions are evaluated against the observed (testing) dataset. By applying this with different splits (changing the training and testing sets) the robustness and fidelity of the model can be improved.

Once all the parameters and options of the models are well defined, a final comparison can be done to illustrate the algorithms performances. This comparison involves several regression models and is summarized in Table 4.1 via their R^2 score. These values are obtained by training the models using first 80% of the data set, while the 20% remaining are used for the testing phase. A second comparison is made using 90% of the samples for training and 10% for the testing. This shows how they perform on predicting the tested values.

TABLE 4.1: Comparison of the R^2 for various regression algorithms.

Regression algorithm	$\mathbf{R^2}$ (80-20)	$\mathbf{R^2}$ (90-10)
RIDGECV [29]	0.020	0.022
LASSOCV [30]	0.545	0.553
RANDOMFORESTREGRESSOR [31]	0.246	0.252
GRADIENTBOOSTINGREGRESSOR [32]	0.527	0.543
XGBREGRESSOR [33] [34]	0.538	0.571
SVR [35]	0.547	0.556

The different models can be briefly described:

- RIDGECV and LASSOCV are both linear models based on classical RIDGE and LASSO, with integrated cross-validation (that allow automatic tuning of the regularization parameter).

- RANDOMFORESTREGRESSOR is a more advanced model that combines decision trees and average their predictions to improve the accuracy.

- The GRADIENTBOOSTINGREGRESSOR is also composed of multiple trees, but each tree tries to correct the errors made by the previous ones.

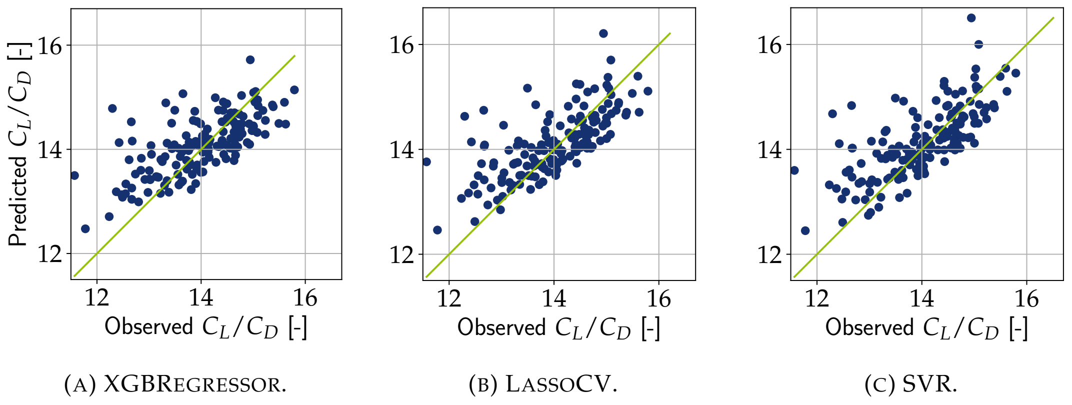
- The XGBREGRESSOR is an improved version (eXtreme-Gradient Boosting), using machine learning methods with regularization techniques, of the GRADIENTBOOSTINGREGRESSOR. This method was first introduced by Chen and Guestrin [34].

- Finally, Support Vector Regression (SVR) is a more sophisticated model that seeks to identify a hyperplane that best fits the data within a defined margin of tolerance, prioritizing a balance between accuracy and simplicity, rather than just minimizing the prediction error directly.

These models are well documented online if the reader is interested in further algorithms descriptions.

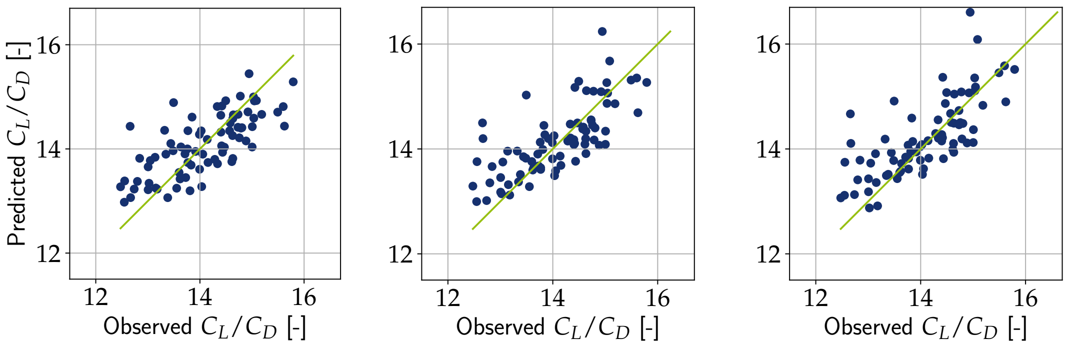
It can be observed from Table 4.1 that there are three models that stand out from the others and a more in depth comparison must be made to select the best one. However, it must be noted that they are just above $R^2 = 0.5$ which is not a very satisfying reliability. This will impact the results in the dimensionality reduction and in the optimization processes. Indeed, even if they are better than the other models, they still perform quite poorly, as the residual errors are non-negligible. These low R^2 values can be explained by the complexity of the problem. On one hand, the physical behavior is very sensitive, as a small wing perturbation can destroy the performances. On the other hand, the number of design variables (125) is really high, and it is hard-work to define a metamodel that is accurate.

Another way to show the models performances is to analyse the error graphically. This is shown in Figure 4.3 and Figure 4.4 for the three remaining models, where the green line represent a perfect fit (i.e., where $y_{\text{predicted}} = y_{\text{observed}}$).



(A) XGBREGRESSOR. (B) LASSOCV. (C) SVR.

FIGURE 4.3: Graphical illustration of R^2 for the three remaining models (80-20).



(A) XGBREGRESSOR. (B) LASSOCV. (C) SVR.

FIGURE 4.4: Graphical illustration of R^2 for the three remaining models (90-10).

From these graphs, there is no clear model that appears to perform better than the others. To be able to select a model, the cross-validation can be modified to get other scoring techniques. This second comparison focuses on both the Root Mean Squared Error (*RMSE*) and the Mean Absolute Error (*MAE*).

$$RMSE = \sqrt{\frac{1}{Q}\sum_{i}^{Q}\left(y_{\text{observed},i} - y_{\text{predicted},i}\right)^2} \tag{4.2}$$

$$MAE = \frac{1}{Q}\sum_{i}^{Q}\left|y_{\text{observed},i} - y_{\text{predicted},i}\right| \tag{4.3}$$

where Q is the number of samples. *RMSE* measures the average magnitude of the prediction errors, penalizing larger error more heavily. The *MAE* measure the average magnitude of all prediction errors, providing a balanced view of overall accuracy.

Table 4.2 summarizes the values obtained with the second cross-validation, where again 80% of the samples are used for training first and then in a second step with 90%, for a total of 20 different splits.

TABLE 4.2: Advanced cross-validation for the three remaining models.

	RMSE **[%]**		***MAE*** **[%]**	
Regression algorithm	**80-20**	**90-10**	**80-20**	**90-10**
LASSOCV	52.84	52.48	40.90	40.98
XGBREGRESSOR	52.73	50.17	41.29	39.58
SVR	52.45	51.61	40.23	39.61

From these results, it seems that they all have a similar *RMSE* and *MAE*. There is again no clear reason to select one model over the others even if a greater improvement with the XGBREGRESSOR can be observed. Therefore it is required to perform a third comparison: bootstrapping analysis. This method helps assessing the stability and reliability of model predictions by repeatedly resampling the dataset with replacement. The process involves creating multiple bootstrap samples by randomly sampling with replacement from the original data, where each sample may include duplicates and miss some original data points. The model is then trained and tested on each bootstrap sample, and this is repeated many times to obtain a range of performance metrics. The results are presented in terms of mean $RMSE$ and standard deviation $RMSE$, and can be found in Table 4.3.

TABLE 4.3: Bootstrapping results for the three remaining models.

Regression algorithm	**mean *RMSE* [%]**	**std *RMSE* [%]**
LASSOCV	40.99	1.49
XGBREGRESSOR	4.94	0.16
SVR	27.38	1.49

Unlike the two previous comparisons, the bootstrapping method clearly shows that the XGBREGRESSOR performs better than the other model. This bootstrapping analysis indiquates that the regression model is more adequate at predicting new, unseen data. Therefore it is the XGBREGRESSOR algorithm that is selected for the rest of this work.

Obviously, when the good performances of the regressor are confirmed, all the samples from the dataset are used to train the final model for an increased reliability. Indeed, as can be seen in Figure 4.5, the more samples in the DoE there are, the more accurate the model is. For the sake of simplicity, this graph is made with the same model parameters, to illustrate the influence of the number of samples on the predictions accuracy. From this graph it is expected that having a larger DoE, with more than 800 samples, will improve the models, and tend towards a R^2 score of 1. Recall that n is the number of DVs (125 here).

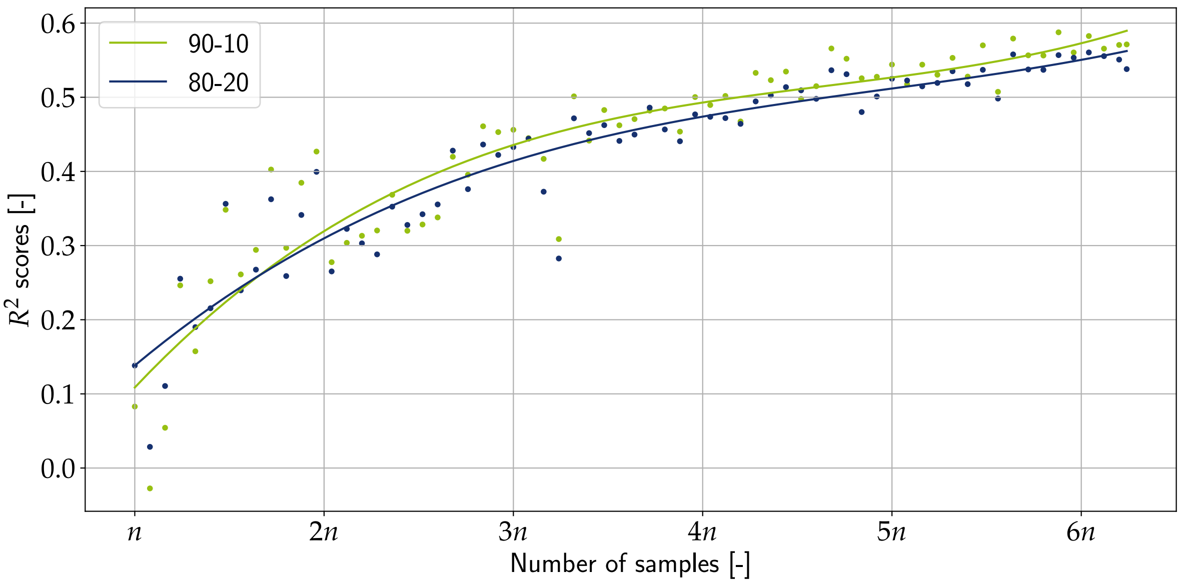


FIGURE 4.5: Influence of DoE size on the predictions accuracy.

4.2.3 Selecting the most important features

Using the trained regressor, a ranking can be made. Indeed it is possible to order the design variable by their importance, which is a measure of their influence on the output. This means that the DV with the highest importance, has the most influence on the lift-to-drag ratio, in this case. On the contrary, the one with the lowest importance, has nearly no impact on the performances of the wing. The reader must keep in mind that this is highly dependent on both the regression algorithm and on the DoE, as increasing the number of samples will improve the accuracy. This ranking is shown on Figure 4.6, where design variables are selected (green point) thanks to their importance. The blue dots are the DV that are considered less influential by the regression model. This graph is only for illustration purposes as the number of selected points is here totally arbitrary. Moreover, it might be good to recall that the DV numbered from 0 to 119 are related to the z perturbation, while the last 5 DV are related to the twist.

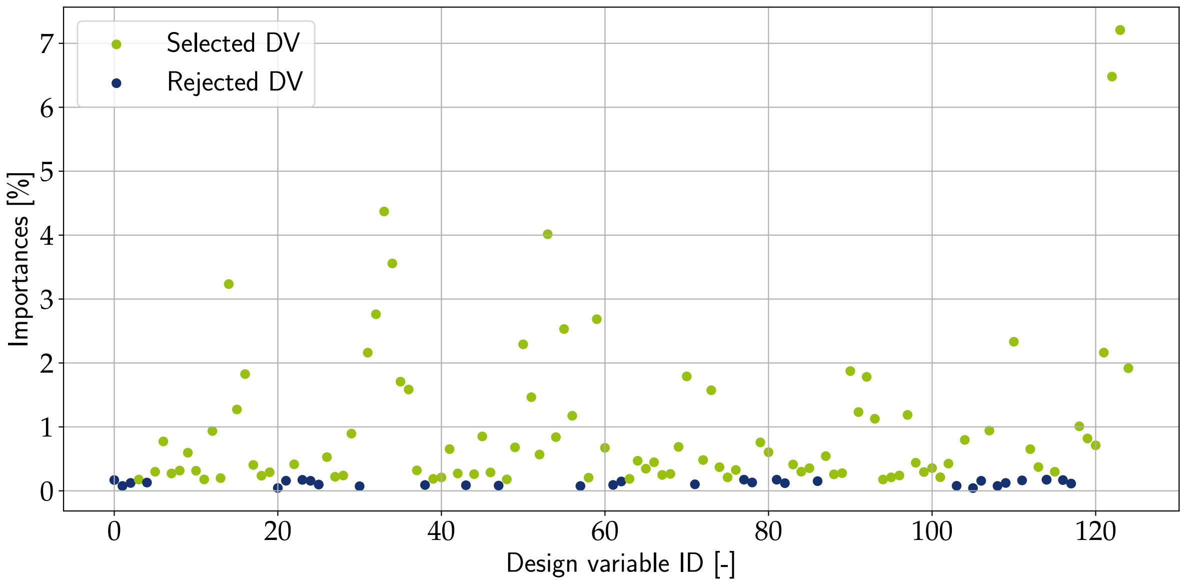


FIGURE 4.6: Graphical view of the feature selection.

One key observation is that a significant portion of the DV has an importance value below 1%. As a result, even though a ranking exists, the differences between the DVs are so minimal that this selection process may not be the most suitable for this case. This suggests that using feature selection alone might not be the most effective method for dimensionality reduction, especially given the large number of features with nearly identical importance values.

In this study however, this ranking will still be used to select the most influential DVs and modify the dataset. The reduced dataset can be obtained by simply deleting or omitting the unselected parameters for each sample. This process can be schematically represented with n

being the number of DVs (125 in this case):

$$\forall \mathbf{x} \in \mathbf{X}: \quad \mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{bmatrix} \xrightarrow{\text{Dimensionality Reduction}} \mathbf{x}_{\text{reduced}} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{bmatrix}$$

The reduced dataset $\mathbf{X}_{\text{reduced}}$ can be used to train again the regression model creating the low dimension LD problem. For improved reliability and accuracy the model should again be determined with its parameters. However, for the sake of simplicity, the same regression algorithm XGBREGRESSOR is used. This LD setup must be compared to the HD setup in order to assess the reliability of the reduced problem. This comparison can be found in the following section.

4.2.4 Results

The performances of the high and low dimensional problems will be compared more clearly in the optimization part (chapter 5). However the two methods presented above can still be applied to obtain the R^2 score and the different measures of the error. The main goal is to assess that even if the LD problem has a reduced number of DV, it does not lead to a high loss of information. In this section the reduced number of DV is set to 100, to keep the most of the parametrization variables. The R^2 scores can be found in Table 4.4. Again the training set is composed first of 80% of the samples and then 90%.

TABLE 4.4: Comparison of the R^2 for the HD and LD models.

Regression algorithm	**R^2** (80-20)	**R^2** (90-10)
HD	0.538	0.571
LD	0.561	0.591

The same more in depth cross-validation can be made. Its output in summarized in Table 4.5.

TABLE 4.5: Advanced cross-validation for the HD and LD models.

	RMSE [%]		*MAE* [%]	
Regression algorithm	**80-20**	**90-10**	**80-20**	**90-10**
HD	52.73	50.17	41.29	39.58
LD	51.46	49.80	40.13	39.41

Finally, the Bootstrapping technique demonstrates that the LD problem yields results similar to those of the HD (Table 4.6), which is promising for the remainder of the work.

TABLE 4.6: Bootstrapping results for the HD and LD models.

Regression algorithm	**mean** *RMSE* [%]	**std** *RMSE* [%]
HD	4.94	0.16
LD	5.31	0.17

When observing the bootstrapping results it appears that the HD model has a lower error. However, the reader might be surprised to see that the LD seems to perform better than the HD in terms of R^2 score. These good performances from the LD setup are not fully understood, although it can be partially explained by the well-known difficulty of non-linear metamodels to reach good performances when the dimensionality increases. Further investigations of potential factors might help to explain this observation. Indeed, if the total number of samples is increased, it is acceptable to think that the HD will perform better. The other aspect that could explain this behavior is the complexity of the problem, as it should be "harder" for the regression model to build a good metamodel. Therefore reducing the dimensionality might help the algorithm.

Chapter 5

Optimization

The optimization of the wing performances is made through the maximization of the lift-to-drag ratio. As a reminder, the objective is to optimize at a single operating point. This means that even if the results of the optimization lead to an improvement of the wing lift-to-drag ratio, it might degrade the overall performances of the whole range of the polar. An interesting research would be to analyse the optimized wing at other operating points. Moreover, the lift-to-drag ratio is a good index to predict the performances of the wing but there are other parameters that play an important role. Even if those parameters are not taken into account, the reader must keep in mind that it would be better (and required) to integrate them in the optimization.

In this chapter, the high dimensional optimization is compared to the reduced one. The objective is to approximate the HD solution with the LD case. The error that is related to the dimensionality reduction must be analysed to conclude the effectiveness of this method. For this whole chapter, the term *predicted* represent the output of the optimizer while the term *observed* is reserved to a value that is validated through a CFD analysis.

5.1 Optimization algorithm

In the world of optimization, there are plenty different algorithms. Each one has its pros and cons. The first element that helps restricting the options is the use, or not, of gradients. Gradient-based optimizer should perform better with a high number of parameters, which is the case in this study. However, their objective is to find the maxima based on the gradient and require to have a continuous and smooth function. Here the function is not continuous as there is a limited number of samples. Moreover the complexity of the problem causes the function to be unstable. In fact, a small change in the magnitude of a parameter can directly influence the performances. This consideration tends to make use of a non gradient-based optimizer.

For this study two optimizers are compared to each other. The first one is COBYLA [36], proposed by Powell [37], one of the most easy to use, often used as a reference algorithm in numerical optimization. The second algorithm, a more advanced optimizer, is the DIFFERENTIAL EVOLUTION (DE), conceptualized by Storn and Price [38].

The optimization process is driven by the objective function, the lift-to-drag ratio predicted by the trained regressor. There are only bound constraints on each design variable. As previously explained, if the magnitude of the deformation goes beyond a certain threshold the actual wing performances would be severely degraded. Therefore, each DV must remain in a given range of value. Both optimizers can take an initial set of values, $\mathbf{x_0}$ which is the starting point of the optimization. This starting point must be wisely chosen as its value can impact the solution.

5.1.1 COBYLA

The COBYLA (Constrained Optimization BY Linear Approximation) optimizer is designed for solving non-linear problems without requiring the derivatives of the objective function. It is a popular choice for many industrial applications as it can deal with constraints. It works by iteratively approximating the objective function using linear models. It starts from the initial set of values and changes each variable until it converges to the solution. It is suited for the case studied here knowing the complexity of the problem. However it remains a relatively simple algorithm that does not guarantee to converge to the global maximum. In fact it can be stuck in a local maxima and never be able to find the best solution. This optimizer can be described as a local search algorithm, as it will improve its solution based on local information rather than exploring the whole space.

Despite its simplicity and ease of implementation, COBYLA might not always be the most efficient, particularly when dealing with a large number of variables as stated by Powell [37].

5.1.2 Differential Evolution

Differential Evolution (DE) is an advanced and robust optimization algorithm designed to tackle complex problems. Unlike COBYLA, which is better suited for simpler problems, DE can handle high-dimensional cases, although this comes at the cost of increased computational time. DE is a population-based method, meaning it evolves a population of candidate solutions across generations. DE is considered an *NP-problem*, as each additional variable requires an increase in the population size, leading to a significant increase in the number of candidates that must be evaluated. According to Eltaeib and Mahmood [39], DE relies on three main processes: mutation, crossover, and selection.

Mutation: This process involves creating a new vector by combining or recombining existing vectors from the population. There are various techniques for selecting these vectors, but a random strategy performs better for the presented case. Three random vectors are typically chosen to form a new vector called the "trial vector".

Crossover: The mutation process is further refined by a crossover technique, which mixes elements from the trial vector and a mutant vector. The crossover process determines which elements are taken from the trial vector and which from the mutant vector. This allows the algorithm to generate new vectors that combine information from previous candidates, enabling a more effective exploration of the search space where potential improved solutions may exist.

Selection: After mutation and crossover, the resulting vector is compared to the current candidate solution (often the best vector found so far). This step ensures that the population either remains the same or improves with each iteration.

Unlike COBYLA, DE explores the entire search space and has a greater likelihood of converging to a global maximum. However, DE also has its challenges, particularly its sensitivity to control parameters and the risk of stagnation, where the population fails to improve after several iterations. Variants of the classic DE algorithm exist that can address these issues and improve performances, but a detailed discussion of these variants is beyond the scope of this thesis.

In conclusion, while DE may offer superior performances compared to COBYLA for complex and high-dimensional problems, it requires higher computational costs.

5.1.3 Determining the starting point

Both optimizers introduced above can utilize an initial vector as a starting point. This vector, denoted as $\mathbf{x_0}$, has a significant impact on the optimization path and, consequently, on the final solution. Practically, $\mathbf{x_0}$ can be any arbitrary set of values. However, certain strategic points should be tested to understand their influence on the optimization outcome.

In this problem, two different initial vectors $\mathbf{x_0}$ are compared. The first is the baseline case that needs to be optimized. The second choice is to start the optimization with the best parametrization obtained from the Design of Experiments (DoE). This second approach, while potentially advantageous, can sometimes lead the optimization process in a inadequate direction. Inadequate in the sense that it may conduct the solution toward a local maximum rather than the global maximum.

The goal of this section is to identify the optimal starting point for both optimizers. To achieve this, two optimizations are conducted: the first uses the baseline case as the initial vector $\mathbf{x_0}$, while the second uses the best case derived from the DoE. The results for both scenarios are summarized in Table 5.1, where the improvements achieved with the baseline performances are presented for both optimizers. (For simplicity, these results are obtained using the HD setup.)

TABLE 5.1: Impact comparison of the two optimization initial points.

	Improvement [%]			
	Predicted		**Observed**	
$\mathbf{x_0}$	**COBYLA**	**DE**	**COBYLA**	**DE**
Baseline	2.519	18.988	5.032	8.066
Best case from DoE	7.949	19.068	9.930	12.228

From these results, it appears that the best case is the more suitable option for the starting point of the optimization. Additionally, both initial vectors $\mathbf{x_0}$ predict positive improvements, indicating that, regardless of the choice, the optimized wing will perform better than the original. Another noteworthy observation is that COBYLA tends to underestimate its predictions, while DE tends to overestimate them. This suggests that COBYLA may be more stable and accurate than DE, even though its observed solutions result in lower lift-to-drag ratios.

It is important to note that the discrepancies between predicted and observed outcomes are likely due to the limitations of the DoE size, as the regression algorithm lacked sufficient information to fully cover the parameter space. The larger errors observed with DE could be attributed to its tendency to explore the entire search space more extensively, which might backfire in the form of increased prediction errors.

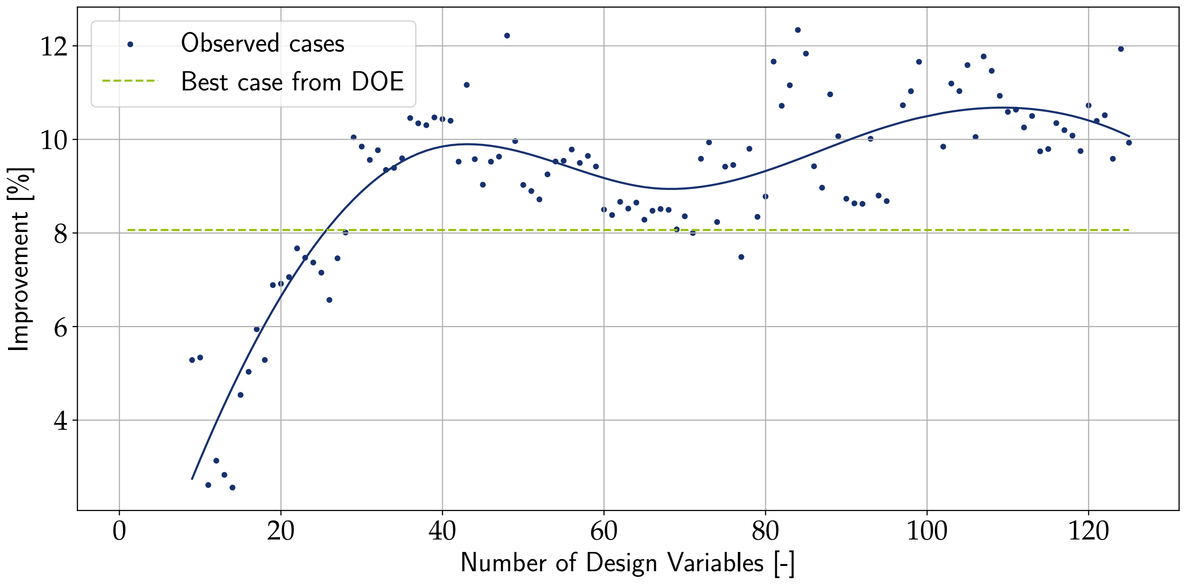
5.2 Optimization with the LD setup

5.2.1 Determining the number of selected DVs

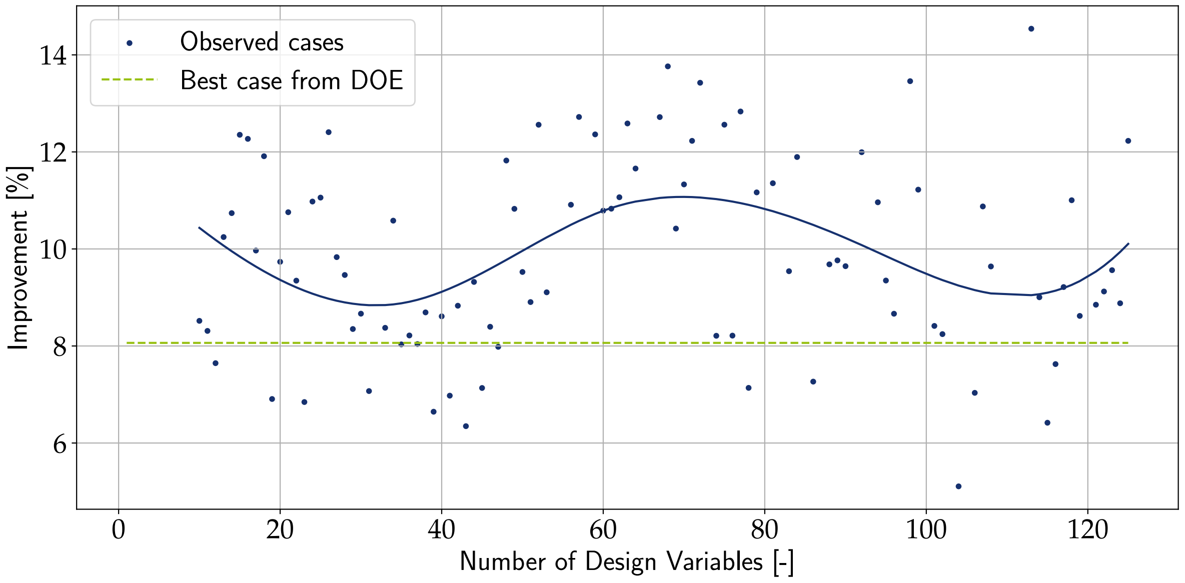
Once the starting point is determined, the actual optimization can proceed. At this stage, both HD and LD optimizations can be performed. However, the number of selected design variables for the LD setup is not predetermined. This number can be set arbitrarily, but it is important to remember that excluding too many DVs results in the loss of valuable information. To better understand the influence of the number of design variables, a small analysis can be conducted. Specifically, it may be beneficial to perform multiple optimizations, progressively increasing the number of selected DVs. This approach serves as an illustrative example, demonstrating how the choice of DVs affects the optimization outcome.

This analysis is performed with both optimizers to help identify an optimal choice of design variables. However, this dimensionality reduction significantly impacts the optimization outputs. Regardless of the optimizer used, the results tend to be very noisy, making it difficult to discern a clear pattern. To aid in visualizing the comparison results, a spline (represented by a solid blue line) can be drawn to provide an approximate indication of performances trend. However, this visualization should be interpreted cautiously, as each design variable, even the less influential ones, contributes to the final solution.

Figure 5.1 illustrates the evolution of performance improvements with respect to the number of parameters. Each data point represents the result of a single optimization. Each optimization relies on its own metamodel constructed by a regression algorithm, that was trained on the corresponding number of design variables, to predict the output. Since the best case from the DoE is used as the starting point for the optimization (indicated by the green dashed line), it serves as a threshold. The goal of the optimization is to achieve better performances than this initial $\mathbf{x_0}$.



(A) COBYLA.



(B) Differential Evolution.

FIGURE 5.1: Improvement predicted by the optimization as a function of the number of design variables.

The results are widely scattered across the graphs, but the spline in Figure 5.1a once again highlights the stability of COBYLA. There is a smooth and expected relationship between accuracy (measured as the error compared to the HD case) and the number of design variables. However, this relationship is much less clear with DE (Figure 5.1b), where even at very low dimensions (e.g., 10 DVs), the results are unexpectedly aligned with the HD case.

It is important for the reader to remember that the prediction model is not 100% accurate, which can occasionally lead to seemingly "lucky" outcomes, i.e., having a high observed value that wasn't expected.

Unfortunately, there is no definitive number of DVs that clearly stands out. Therefore, the choice to use 94 DVs is somewhat arbitrary. However, the splines in both graphs suggest that when the number of DVs is around 94, the solution closely aligns with the HD setup. Reducing the number of DVs to 94 in the LD setup corresponds to a 25% reduction in dimensionality.

A potential question that the reader might ask is: *How are unselected variables treated in the CFD simulation?* The first option is to set zero deformation for the unselected DVs. However, this approach is not well-suited to the complexity of the case, as it may lead to abrupt modifications of the wing surface which must be avoided to maintain a smooth wing profile. Therefore, an alternative approach is adopted.

This approach involves applying a gradual reduction towards zero using a smoothing factor of 20%. To illustrate this technique, a small example can be more effective than a lengthy explanation.

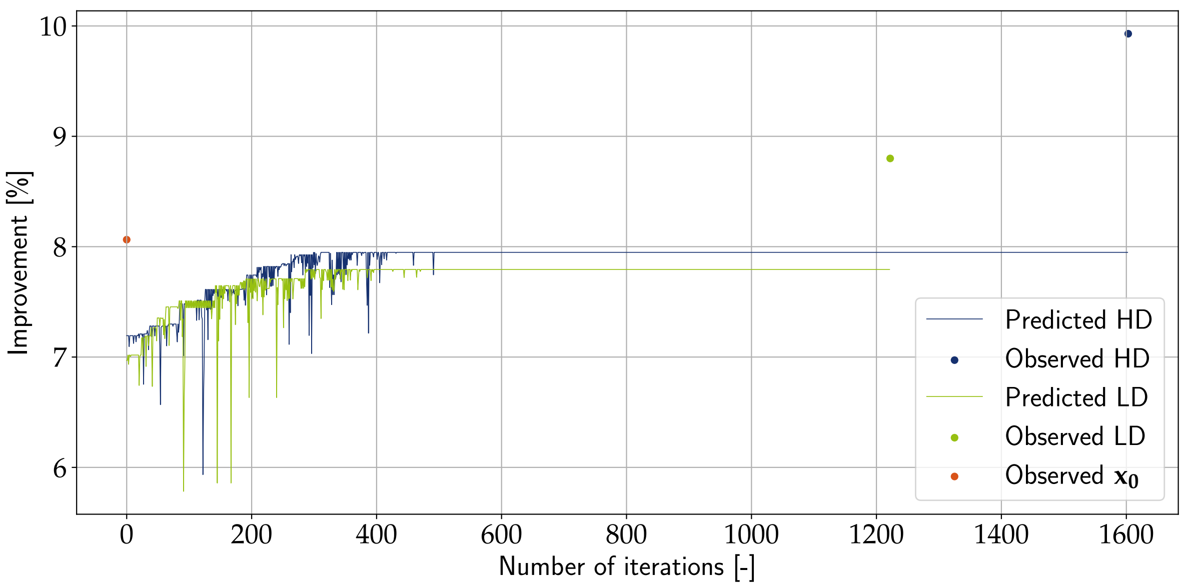
In a problem involving five parameters that is reduced to two variables, if the two remaining variables are x_1 and x_2, the application of the smoothing reconstruction might look something like this:

$$\begin{tikzcd}
\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \arrow[r, "\text{Dimensionality Reduction}"] & \mathbf{x}_{\text{reduced}} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \arrow[d, "\text{Optimization}"] \\
\mathbf{x}_{\text{recomposed}} = \begin{bmatrix} 1.6 \\ 2 \\ 6 \\ 4.8 \\ 3.84 \end{bmatrix} & \mathbf{x}_{\text{reduced}} = \begin{bmatrix} x_0 \\ 2 \\ 6 \\ x_3 \\ x_4 \end{bmatrix} \arrow[l, "\text{Smoothing reconstruction}"']
\end{tikzcd}$$

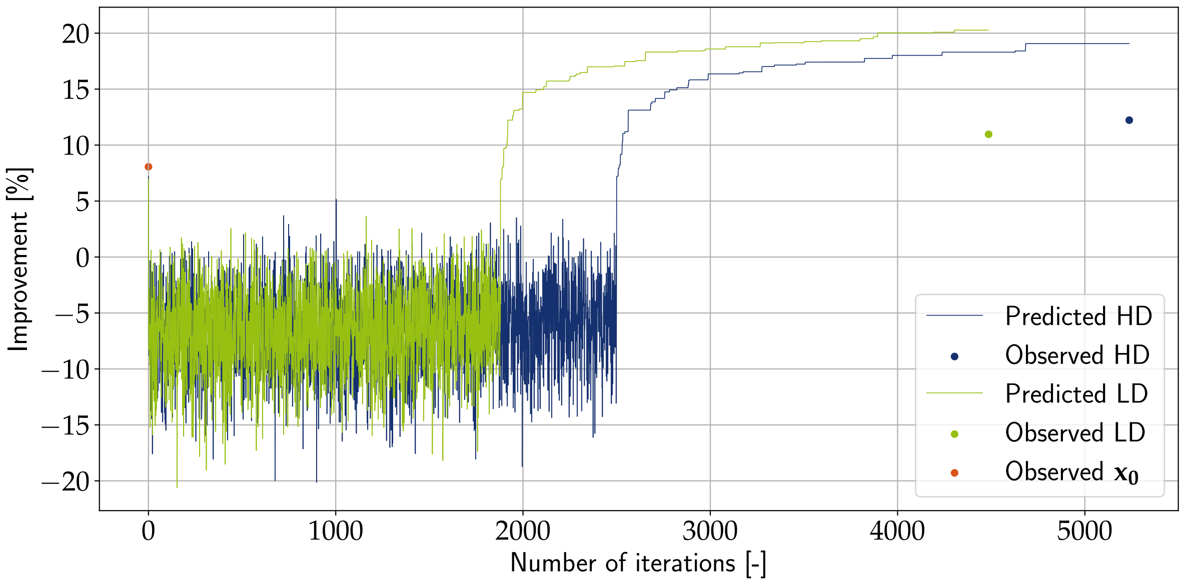
5.2.2 Comparison of HD and LD setup

The LD setup is now fully determined and can be compared to the HD optimization. The objective of this analysis is to investigate whether dimensionality reduction methods can simplify complexity without introducing significant errors. The best way to compare the two setups is by quantifying the final error in the observed lift-to-drag ratios.

First, the convergence of both optimizers is illustrated in Figure 5.2. The solid lines represent the predicted improvement relative to the baseline lift-to-drag ratio, while the bullet points denote the observed values. In both graphs, the starting point $\mathbf{x_0}$ is shown in orange, indicating that both optimizers are capable of producing outcomes better than this initial point.



(A) COBYLA.



(B) Differential Evolution.

FIGURE 5.2: Performance improvements predicted by the optimization for both HD and LD setup.

COBYLA once again demonstrates its stability throughout the incremental process. Figure 5.2a shows that both the HD and LD setups converge rapidly to their respective solutions. The LD setup clearly exhibits information loss by predicting a lower lift-to-drag ratio compared to the HD prediction. This loss is less apparent with DE, as the converged LD setup appears above the HD prediction in Figure 5.2b. DE is designed to explore the entire parameter space, which explains the oscillations in the graph, where predictions fluctuate between -20% and +5%. However, once DE identifies the direction of the global maximum, there is a rapid increase in prediction improvement.

The range of values tested by DE does not allow for a clear comparison of the observed values. Therefore, these results are summarized in Table 5.2.

TABLE 5.2: Comparisons of the observed optimization output.

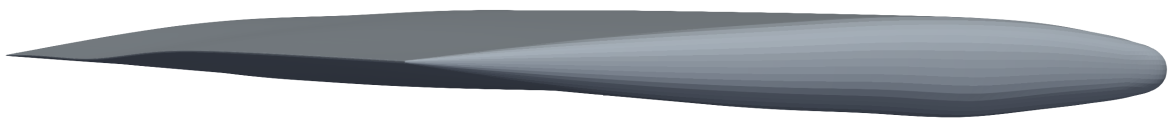
	Observed improvement [%]	
	LD	**HD**
COBYLA	8.800	9.930
DE	10.962	12.228

Table 5.2 shows that the DE optimizer performs better and produces a wing with improved performances. Regarding dimensionality reduction, both cases are close to the unreduced output, indicating that the LD setup reliably provides results comparable to the HD case.

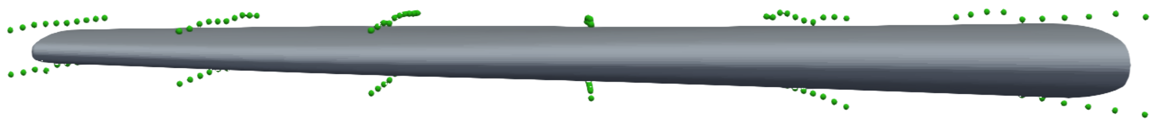
5.3 Visualizing the optimized wing

This section presents the wing obtained using the DE optimizer, illustrating both the wing deformation and the pressure fields with the shocks. The figures in this section depict the results of the reduced case.

Figure 5.3 shows the deformations applied to the wing. However, the magnitudes are so small that they may be difficult to visualize.



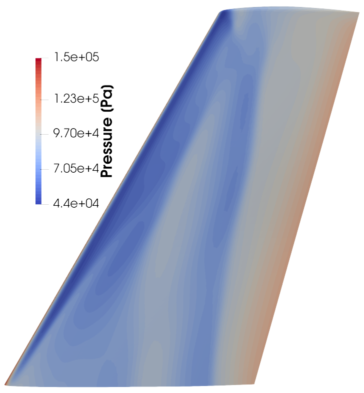
(A) View from the tip.



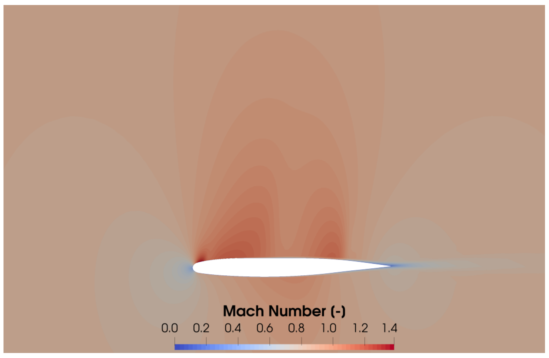
(B) View from the leading edge.

FIGURE 5.3: Representation of the optimized wing deformations.

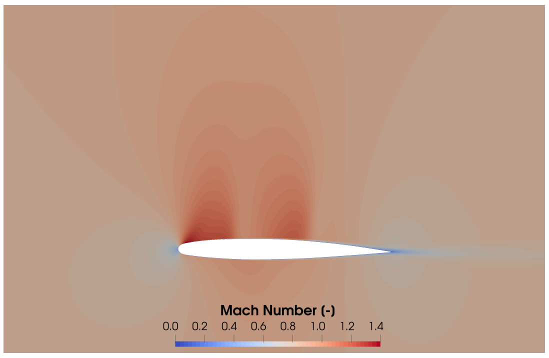
Concerning the shocks, they can be observed in Figure 5.4.



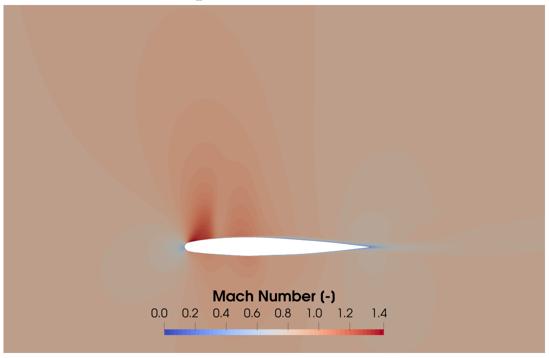
(A) Dorsal view.



(B) Span station $\eta = 0.2$.



(C) Span station $\eta = 0.65$.

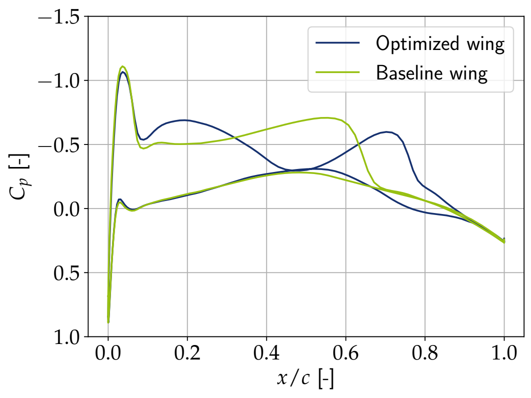


(D) Span station $\eta = 0.96$.

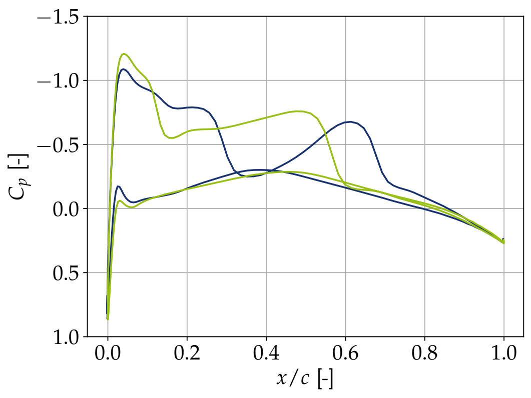
FIGURE 5.4: Illustration of the shocks obtained with ParaView [20].

One can observe that the shocks are more prominent across the entire extrados, particularly at the span station $\eta = 0.65$ (Figure 5.4c), where the shocks are more pronounced at mid-chord. This indicates that the wing deformation affects the shock patterns. It would be valuable to investigate this behavior experimentally.

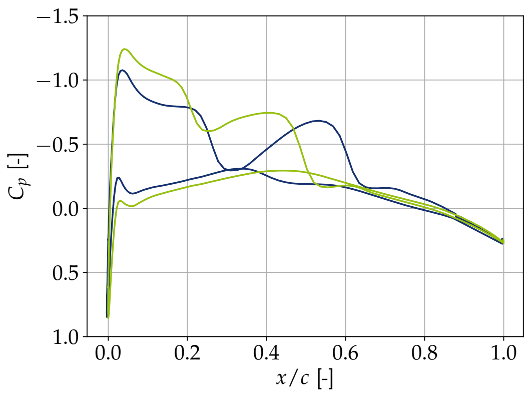
Finally, the C_p curves are shown in Figure 5.5, comparing the optimized wing to the baseline wing.



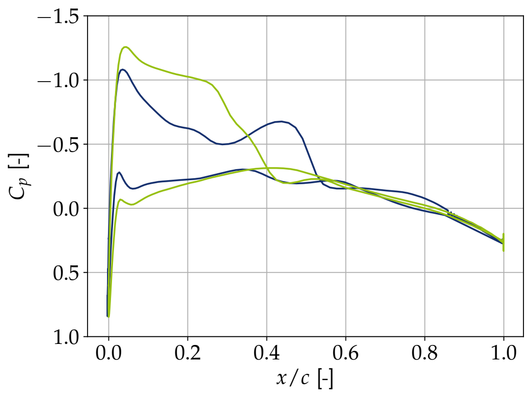
(A) C_p at span station $\eta = 0.2$.



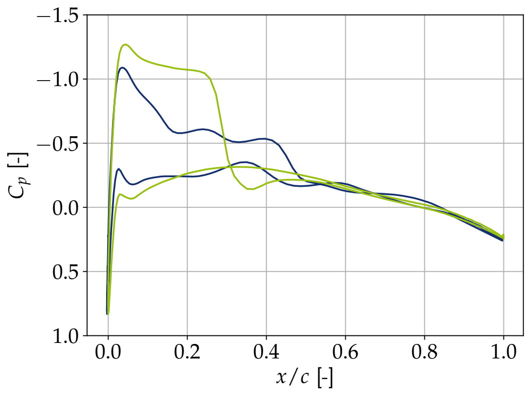
(B) C_p at span station $\eta = 0.44$.



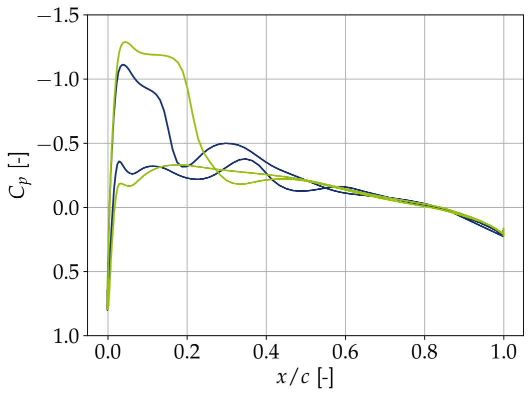
(C) C_p at span station $\eta = 0.65$.



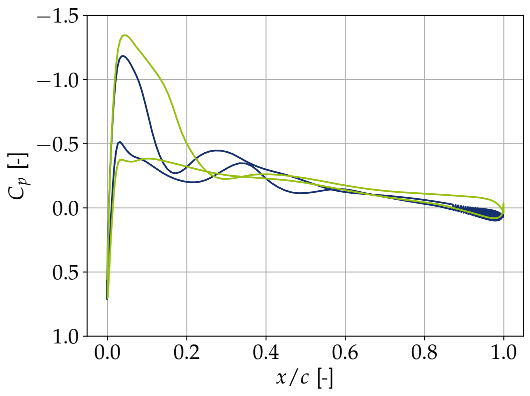
(D) C_p at span station $\eta = 0.8$.



(E) C_p at span station $\eta = 0.9$.



(F) C_p at span station $\eta = 0.96$.



(G) C_p at span station $\eta = 0.99$.

FIGURE 5.5: C_p comparisons between optimized and baseline wings.

As expected, the pressure coefficient curves reveal that the optimized wing affects the shocks. Specifically, the shocks are more abrupt and unstable. These pronounced changes in the shocks are likely to have a significant impact on performances, beyond what is observed in the CFD simulation. Therefore, it would be highly valuable to validate this wing design experimentally.

Chapter 6

Future directions and enhancements

The framework presented in this master's thesis proposes integrating dimensionality reduction into the optimization process. However, several aspects require additional research to enhance the framework reliability and accuracy. This chapter outlines various ideas and key considerations for further improvement. Each section is examined and analyzed to offer targeted recommendations that address the specific issues discussed.

6.1 Computational Fluid Dynamics

The CFD component represents a significant source of potential errors, as the analysis presented relies heavily on the results from CFD simulations. The adage in engineering, "garbage in, garbage out," underscores the importance of accurate inputs. To clarify and simplify the discussion, it is divided into two topics: the mesh and the OpenFOAM software.

6.1.1 The mesh

Mesh generation is the starting point of any simulation and must be clean, reliable, and efficient. To improve results, the mesh should be adapted for the case studied, featuring very thin cells in the boundary layer and larger cells further from the wall. Adding more cells in regions where shocks occur, such as the extrados and leading edge, is crucial. Overall mesh refinement generally yields more accurate results, though this is practical only if the simulation budget allows. Alternative grid types, such as H-type or C-type grids, or even a combination of different types, may be more appropriate. There are numerous ways to enhance the solution simply by modifying the grid construction.

To simplify the automation process, a thick trailing edge was used. Investigating the effects of a sharp trailing edge might be beneficial.

Additionally, with the current setup, the first layer thickness is significantly smaller compared to other cell dimensions, leading to a warning about high-aspect ratio cells, with a factor of 1000 between two cell dimensions. While this is a concern for transient problems, there is no clear indication that it affects this steady-state analysis. However, it would be prudent to assess this issue further, as it could impact convergence rates and computational time.

6.1.2 OpenFOAM

OpenFOAM is a versatile, open-source tool that facilitates automation. However, it is not inherently designed for compressible solvers. While correction functions are implemented to address compressibility effects, the accuracy of the results may be compromised. It is recommended to consider alternative software, such as SU2 [40], which may be better suited for analyzing the Onera M6 wing.

If continuing with OpenFOAM, numerous parameters, built-in functions, and options can be tuned to enhance the setup. A thorough analysis of these parameters can improve convergence rates and accuracy, though this requires in-depth knowledge of the software.

Another critical aspect is the choice of numerical schemes used in OpenFOAM. Various schemes are available, and each field may require a specific divergence scheme. Careful selection of these schemes is essential, as they can significantly influence simulation behavior. For instance, an in-depth analysis of limiters and the order of accuracy of the schemes is necessary to ensure a high-quality setup. Similarly, the turbulence model should be optimized; different versions of the Spalart-Allmaras model exist, and the standard version might not be the most suitable.

Additionally, there are no widely available, reliable test cases for the Onera M6 wing in OpenFOAM, either in the literature or online. Developing a robust and accurate CFD setup for this wing would be a valuable contribution.

6.2 Free-Form Deformation

For the execution of this master's thesis, 120 FFD points were utilized. It would be beneficial to explore the response using a greater number of FFD points, as well as fewer. This could potentially allow for more precise and nuanced tuning of the wing shape.

Regarding parameterization, there are numerous alternative approaches and choices that could be tested. Allowing for modifications with a greater magnitude might lead to a higher lift-to-drag ratio if executed carefully. The current approach employed a broad and generalized parameterization. However, if the focus is specifically on enhancing the wing performances, applying aerodynamic knowledge to guide the optimization could be advantageous. For example, adjusting the sweep angle is known to significantly impact drag generation at high velocities.

6.3 Dimensionality reduction

The primary objective of this thesis was to introduce dimensionality reduction and integrate it into the optimization process. However, as discussed, numerous techniques exist, each varying in complexity. This represents the main area for potential improvement. The complexity of the problem cannot be reduced merely by eliminating less influential variables. Moreover, alternative methods might be better suited for this purpose. A preliminary study could involve examining the Principal Component Analysis (PCA) [4] method and its impact on the optimization process and results. However, since PCA has already been used in the literature [3], it may be more insightful to explore the application of other methods and techniques.

Variable clustering offers significant advantages for this case. Grouping parameters through clustering can help reduce noise, providing a more effective approach than removing variables with the feature selection method. Variable clustering also allows for the grouping of parameters without requiring specific aerodynamic insight, thereby minimizing subjectivity and facilitating automation by analyzing DoE samples.

The choice of regression algorithm is also crucial, as its predictions drive the optimization process. Increasing the size of the DoE can enhance the model and its predictions, though practical constraints often limit this. Various regression algorithms are available, and more advanced models may improve results. For instance, developing a neural network could be costly but might offer improved accuracy. Such machine learning approaches require fine-tuning and a good understanding of algorithms, but they have the potential to significantly enhance overall accuracy.

Finally, in industrial applications, the simulation budget is a critical constraint. Addressing this limitation is essential. Dimensionality reduction methods offer a viable option to reduce process costs. A promising approach could be to use a global iterative process, as illustrated in Figure 6.1.

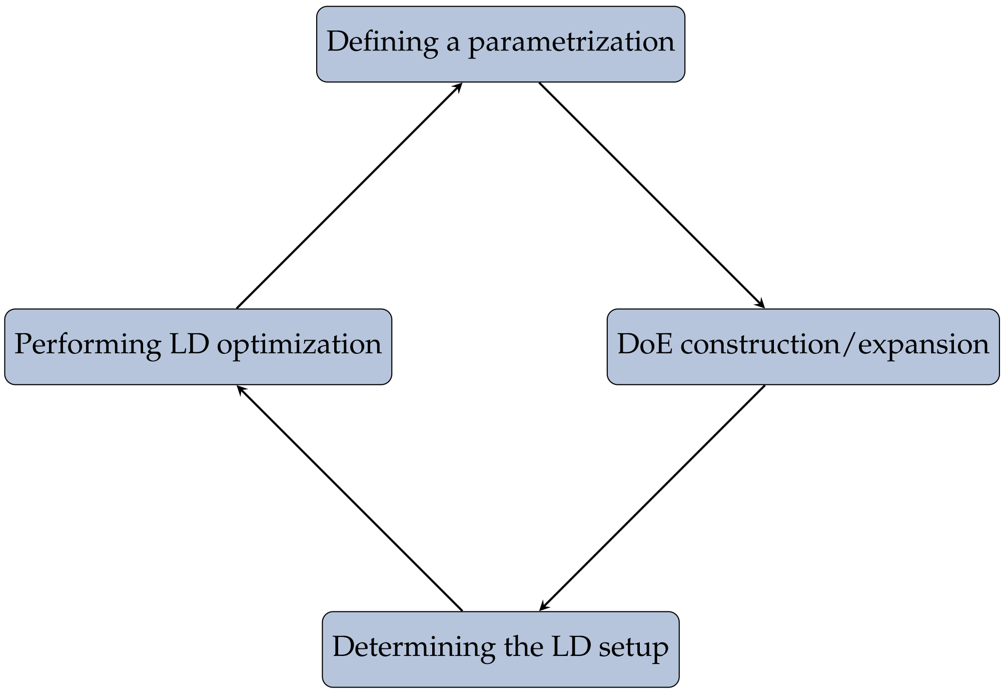

Defining a parametrization
DoE construction/expansion
Determining the LD setup
Performing LD optimization

FIGURE 6.1: Potential industrial iterative process.

The methodology is straightforward. Each iteration begins with the LD setup to restrict the design space. This approach helps converge towards a promising region where the optimal wing design is likely to be found. Once convergence is achieved with the LD setup, or when the simulation budget is nearing its limit, a final iteration can be performed using the HD setup. This final iteration utilizes the entire parameterization space to enhance accuracy and capture detailed characteristics of the design.

6.4 Optimization

In the optimization chapter, two different algorithms are introduced. Both are derivative-free, and other optimizers might offer better performances. For instance, DAFoam provides three gradient-based algorithms, e.g. SLSQP (Sequential Least Squares Quadratic Programming) [41] , SNOPT (Sparse Nonlinear OPTimizer) [42], and IPOPT (Interior Point OPTimizer) [43], which are well-regarded in the literature. Each of these algorithms has its own advantages and limitations, which should be considered to match the complexity of this optimization problem. Additionally, the choice of the optimizer influences the available settings, such as tolerance levels and maximum number of iterations, which require careful analysis to fine-tune these settings for more accurate and efficient solutions.

The optimization process was conducted offline. An improvement could be to implement an online mode, known as Surrogate-Based Optimization (SBO) [2], which allows the optimizer to run CFD simulations at specific points during the iteration process. This approach could enhance convergence and improve the optimized wing, but it may be costly, as each CFD simulation is not instantaneous.

The current optimization focuses on a single operating point, which, while useful, is not fully realistic. Ideally, the optimization should encompass the entire range of the polar curve. For example, while optimization might target the cruise operating point, it should not lead to performances degradation at other operating points.

In a broader application context, additional constraints are necessary to ensure the wing reliability beyond achieving a high lift-to-drag ratio. While this ratio is a valuable aerodynamic metric, it cannot fully characterize a wing. A more comprehensive optimization process should consider parameters such as maneuverability, total mass, structural integrity (including load resistance), shock position and strength, control effectiveness, and stability. A coupled fluid-structure approach is essential to ensure the wing viability.

Finally, to validate the optimization results, experimental testing of the optimized wing is necessary. This will provide an assessment of the performances and verify the effectiveness of the optimization process.

Chapter 7

Conclusion

This master's thesis aimed to develop a framework to enhance the performances of a given wing. The wing selected for this study was the renowned Onera M6 wing, which operates under transonic conditions where shockwaves occur. The performances of this wing were evaluated using Computational Fluid Dynamics (CFD) simulations conducted with OpenFOAM software.

A key objective of this thesis was to incorporate dimensionality reduction techniques into the optimization process. The wing shape was deformed using a chosen parameterization involving 125 design variables, with deformation applied through the Free-Form Deformation (FFD) method, which allows for smooth modifications. The high dimensionality of this problem impacts the optimization process, and reducing dimensionality simplifies the problem, approximating results with reduced complexity. The dimensionality reduction technique employed in this study was feature selection, specifically the filter method, which ranks design variables based on their influence on the response.

The optimization process was carried out using two derivative-free optimizers: COBYLA and Differential Evolution (DE). COBYLA was found to be more stable but yielded less optimal results. In contrast, DE, though more unstable, explored a broader search space and achieved better performances. DE demonstrated a 12% improvement with the high-dimensional setup and nearly 11% improvement with a reduced setup containing 94 design variables. This indicates that even with dimensionality reduction, the approximation maintains a low error, proving that dimensionality reduction methods can be valuable in industrial contexts where simulation budgets are a major constraint. These methods offer significant cost reductions without substantial loss of accuracy.

Although this study introduces promising techniques for dimensionality reduction, further research is needed. The preliminary framework developed here can be enhanced in several manners. The ideas and specific points of attention outlined in this thesis are intended to guide future research and improvements in this area.

Appendix A

Drawing of the Onera M6 wing

Figure A.1 shows the schematic of the Onera M6 wing, as it can be found on the Onera website [8]. The main characteristics of the wing are represented here, with all the relevant dimensions. All this information are not required for this work but can help to create a mesh.

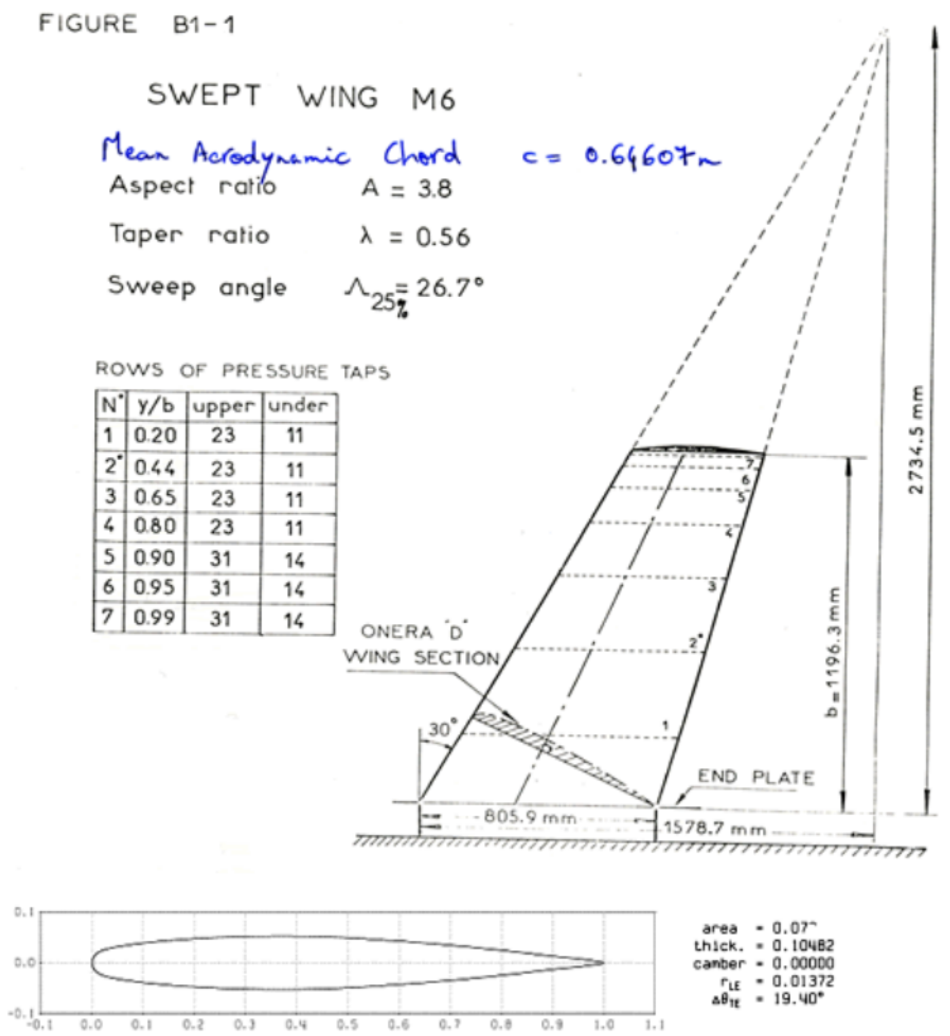


FIGURE B1-1

SWEPT WING M6

Mean Aerodynamic Chord		c = 0.64607m
Aspect ratio	A = 3.8	
Taper ratio	λ = 0.56	
Sweep angle	$\Lambda_{25\%}$ = 26.7°	

ROWS OF PRESSURE TAPS

N°	y/b	upper	under
1	0.20	23	11
2*	0.44	23	11
3	0.65	23	11
4	0.80	23	11
5	0.90	31	14
6	0.95	31	14
7	0.99	31	14

area	= 0.07~
thick.	= 0.10482
camber	= 0.00000
r_{LE}	= 0.01372
$\Delta\theta_{TE}$	= 19.40°

FIGURE A.1: Onera M6 wing and its ’D’ section (from Onera[8]).

Bibliography

[1] Gaetan Kenway, Graeme Kennedy, and Joaquim Martins. "A CAD-Free Approach to High-Fidelity Aerostructural Optimization". In: *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*. Sept. 2010. DOI: 10.2514/6.2010-9231.

[2] Nestor V. Queipo et al. "Surrogate-based analysis and optimization". In: *Progress in Aerospace Sciences* 41.1 (2005), pp. 1–28. ISSN: 0376-0421. DOI: https://doi.org/10.1016/j.paerosci.2005.02.001. URL: https://www.sciencedirect.com/science/article/pii/S0376042105000102.

[3] David Gaudrie et al. "From CAD to Eigenshapes for Surrogate-based Optimization". In: *13th World Congress of Structural and Multidisciplinary Optimization*. Beijing, China, May 2019. URL: https://hal.science/hal-02142492.

[4] G Berkooz, P Holmes, and J L Lumley. "The proper orthogonal decomposition in the analysis of turbulent flows". In: *Annual Review of Fluid Mechanics* 25 (1993), pp. 539–575.

[5] Laurens van der Maaten, Eric Postma, and H. Herik. "Dimensionality Reduction: A Comparative Review". In: *Journal of Machine Learning Research - JMLR* 10 (Jan. 2007).

[6] S. Velliangiri, S. Alagumuthukrishnan, and S Iwin Thankumar joseph. "A Review of Dimensionality Reduction Techniques for Efficient Computation". In: *Procedia Computer Science* 165 (2019). 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019, pp. 104–111. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2020.01.079. URL: https://www.sciencedirect.com/science/article/pii/S1877050920300879.

[7] Chun Kit Jeffery Hou and Kamran Behdinan. "Dimensionality Reduction in Surrogate Modeling: A Review of Combined Methods". In: *Data Science and Engineering* 7.4 (Dec. 2022), pp. 402–427. ISSN: 2364-1541. DOI: 10.1007/s41019-022-00193-5. URL: https://doi.org/10.1007/s41019-022-00193-5.

[8] ONERA website. URL: https://www.onera.fr/en. (accessed: 29.04.2024).

[9] OpenFoam website. URL: https://www.openfoam.com/. (accessed: 29.04.2024).

[10] Ping He et al. "DAFoam: An open-source adjoint framework for multidisciplinary design optimization with OpenFOAM". In: *AIAA Journal* 58.3 (2020), pp. 1304–1319.

[11] Ping He et al. "An aerodynamic design optimization framework using a discrete adjoint approach with OpenFOAM". In: *Computers & Fluids* 168 (2018), pp. 285–303.

[12] Turbulence Modeling Resource - Onera M6. URL: https://turbmodels.larc.nasa.gov/onerawingnumerics_val_sa.html. (accessed: 29.04.2024).

[13] Cenaero website. URL: https://www.cenaero.be/. (accessed: 28.07.2024).

[14] Lucia documentation. URL: https://doc.lucia.cenaero.be/. (accessed: 28.07.2024).

[15] Turbulence Modeling Resource - The Spalart-Allmaras Turbulence Model. URL: https://turbmodels.larc.nasa.gov/spalart.html. (accessed: 31.07.2024).

[16] SimFlow Computational Fluid Dynamics Software website. URL: https://help.sim-flow.com/solvers/simple-foam#_simple_algorithm. (accessed: 29.04.2024).

[17] CFD Online - Forum. URL: https://www.cfd-online.com/Forums/main/107399-question-about-large-expansion-ratio-influence-numerical-accuracy.html. (accessed: 28.07.2024).

[18] LEAP CFD Team. URL: https://www.computationalfluiddynamics.com.au/y-plus_part1_understanding-the-physics-of-boundary-layers/. (accessed: 28.07.2024).

[19] V. SCHMITT. "Pressure Distributions on the ONERA M6-Wing at Transonic Mach Numbers, Experimental Data Base for Computer Program Assessment". In: *AGARD AR-138* (1979), pp. 327–370. URL: https://cir.nii.ac.jp/crid/1570009749345531648.

[20] Paraview website. URL: https://www.paraview.org/. (accessed: 28.07.2024).

[21] Thomas W. Sederberg and Scott R. Parry. "Free-form deformation of solid geometric models". In: *SIGGRAPH Comput. Graph.* 20.4 (Aug. 1986), pp. 151–160. ISSN: 0097-8930. DOI: 10.1145/15886.15903. URL: https://doi.org/10.1145/15886.15903.

[22] Hannah M. Hajdik et al. "pyGeo: A geometry package for multidisciplinary design optimization". In: *Journal of Open Source Software* 8.87 (2023), p. 5319. DOI: 10.21105/joss.05319.

[23] S. Ulaganathan et al. "High dimensional Kriging metamodelling utilising gradient information". In: *Applied Mathematical Modelling* 40.9 (2016), pp. 5256–5270. ISSN: 0307-904X. DOI: https://doi.org/10.1016/j.apm.2015.12.033. URL: https://www.sciencedirect.com/science/article/pii/S0307904X15008458.

[24] R Filomeno Coelho, C Sainvitu, and T Benamara. "UMAP-based Dimensionality Reduction for Variable Grouping: Application to the Design Optimization of Truss Structures". In: *13th ASMO UK / 2nd ASMO-EUROPE / ISSMO Conference on Engineering Design Optimization Product and Process Improvement, Cenaero, Belgium, July 8-9*. 2024.

[25] Y Wang, H Yao, and S Zhao. "Auto-encoder based dimensionality reduction". In: *Neurocomputing* 184 (2016), pp. 232–242.

[26] Rebecca Marion. "Statistical and machine learning methods for identifying clusters of variables : with applications in omics, ecology and psychology". Anglais. PhD thesis. UCL UCL - SSH/LIDAM/ISBA - Institut de Statistique, Biostatistique et Sciences Actuarielles Faculté des Sciences, 2021. URL: http://hdl.handle.net/2078.1/252866.

[27] Benoît Frénay, Gauthier Doquire, and Michel Verleysen. "Is mutual information adequate for feature selection in regression?" In: *Neural Networks* 48 (2013), pp. 1–7. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2013.07.003. URL: https://www.sciencedirect.com/science/article/pii/S0893608013001883.

[28] Latin Hypercube Sampling from Scipy. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.qmc.LatinHypercube.html. (accessed: 28.07.2024).

[29] RidgeCV documentation from scikit-learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeCV.html. (accessed: 28.07.2024).

[30] LassoCV documentation from scikit-learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoCV.html. (accessed: 28.07.2024).

[31] RandomForestRegressor documentation from scikit-learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html. (accessed: 28.07.2024).

[32] GradientBoostingRegressor documentation from scikit-learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html. (accessed: 28.07.2024).

[33] XGBRegressor documentation from xgboost. URL: https://xgboost.readthedocs.io/en/stable/python/python_api.html#xgboost.XGBRegressor. (accessed: 28.07.2024).

[34] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794. ISBN: 9781450342322. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

[35] Support Vector Regression documentation from scikit-learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html. (accessed: 28.07.2024).

[36] COBYLA documentation from SciPy. URL: https://docs.scipy.org/doc/scipy/reference/optimize.minimize-cobyla.html. (accessed: 28.07.2024).

[37] M. J. D. Powell. "A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation". In: *Advances in Optimization and Numerical Analysis*. Ed. by Susana Gomez and Jean-Pierre Hennart. Dordrecht: Springer Netherlands, 1994, pp. 51–67. ISBN: 978-94-015-8330-5. DOI: 10.1007/978-94-015-8330-5_4. URL: https://doi.org/10.1007/978-94-015-8330-5_4.

[38] Rainer Storn and Kenneth Price. "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces". In: *Journal of Global Optimization* 11.4 (Dec. 1997), pp. 341–359. ISSN: 1573-2916. DOI: 10.1023/A:1008202821328. URL: https://doi.org/10.1023/A:1008202821328.

[39] Tarik Eltaeib and Ausif Mahmood. "Differential Evolution: A Survey and Analysis". In: *Applied Sciences* 8.10 (2018). ISSN: 2076-3417. DOI: 10.3390/app8101945. URL: https://www.mdpi.com/2076-3417/8/10/1945.

[40] SU2 website. URL: https://su2code.github.io/. (accessed: 19.08.2024).

[41] Dieter Kraft. "A software package for sequential quadratic programming". In: *Forschungsbericht-Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt* (1988).

[42] Philip E. Gill, Walter Murray, and Michael A. Saunders. "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization". In: *SIAM Review* 47.1 (2005), pp. 99–131. DOI: 10.1137/S0036144504446096. eprint: https://doi.org/10.1137/S0036144504446096. URL: https://doi.org/10.1137/S0036144504446096.

[43] Andreas Wächter and Lorenz T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". eng. In: *Mathematical programming* 106.1 (2006), pp. 25–57. ISSN: 0025-5610.