University of Naples Federico II
Naples, Italy

Optimal Energy-Driven Aircraft Design Under Uncertainty

Elisa Morales Tirado

Thesis for the degree of
Doctor of Philosophy
in Industrial Engineering
Cycle XXXIII

Supervisor: Renato Tognaccini
Supervisor: Domenico Quagliarella

Coordinator: Michele Grassi

February 2021

Acknowledgements

First of all, I would like to thank Domenico Quagliarella and Renato Tognaccini for their work, support, guidance, and patience. This was a tough and long journey but we managed to finish it. Besides, I would like to acknowledge Stefan Görtz and Thomas Bartz-Beielstein and Boris Naujoks alongside the colleagues of the German Aerospace Centre (DLR) and the Technische Hochschule Köln (THK) for the kind hospitality in their research groups and their collaboration. Furthermore, I thank the three reviewers of the dissertation Rauno Cavallaro, Valentino Pediroda, and Marco Fossati for their advice and helpful suggestions to improve the quality of the thesis.

Moreover, I had the pleasure to carry on my dissertation within the UTOPIAE project[1]. I would like to thank every ESR for their support and for all the great moments we had together along this journey. You made it smoother since we were able to share our fears and difficulties, helping each other to achieve our objectives and reach the end of this, not always easy, journey.

I would like to thank also my colleagues at my host institutions, the Italian Aerospace Research Centre (CIRA) and the University of Naples Federico II. Thank you Giuseppe Mingione for welcoming me in the research group. In addition, I would like to expressly thank Donato de Rosa, without your support and friendship I would have given up more than once, and, also, Riccardo Rocchio for your friendship and for making my social life existent.

I thank my friends. Thank you from listening to me, advising me, and making my life more easy going.

Furthermore, thank you very much Marco. You were by my side in the good and bad times. You have believed in me, sometimes, more than myself.

And last but not least, I thank my family. You were always supporting me from the distance. I need to especially thank my mother and my father. Without them I would have not achieved my goals. I have not enough lines to thank you for all the things you have made.

"The important thing is to not stop questioning. Curiosity has its own reason for existing."–
Albert Einstein.

[1] This work was supported by the H2020-MSCA-ITN-2016 UTOPIAE, grant agreement 722734.

Optimal Energy-Driven Aircraft Design Under Uncertainty

Abstract

Aerodynamic shape design robust optimization is gaining popularity in the aeronautical industry as it provides optimal solutions that do not deteriorate excessively in the presence of uncertainties. Several approaches exist to quantify uncertainty and, the dissertation deals with the use of risk measures, particularly the Value at Risk (VaR) and the Conditional Value at Risk (CVaR). The calculation of these measures relies on the Empirical Cumulative Distribution Function (ECDF) construction. Estimating the ECDF with a Monte Carlo sampling can require many samples, especially if good accuracy is needed on the probability distribution tails. Furthermore, suppose the quantity of interest (QoI) requires a significant computational effort, as in this dissertation, where has to resort to Computational Fluid Dynamics (CFD) methods. In that case, it becomes imperative to introduce techniques that reduce the number of samples needed or speed up the QoI evaluations while maintaining the same accuracy. Therefore, this dissertation focuses on investigating methods for reducing the computational cost required to perform optimization under uncertainty. Here, two cooperating approaches are introduced: speeding up the CFD evaluations and approximating the statistical measures.

Specifically, the CFD evaluation is sped up by employing a far-field approach, capable of providing better estimations of aerodynamic forces on coarse grids with respect to a classical near-field approach. The advantages and critical points of the implementation of this method are explored in viscous and inviscid test cases.

On the other hand, the approximation of the statistical measure is performed by using the gradient-based method or a surrogate-based approach. Notably, the gradient-based method uses adjoint field solutions to reduce the time required to evaluate them through CFD drastically. Both methods are used to solve the shape optimization of the central section of a Blended Wing Body under uncertainty. Moreover, a multi-fidelity surrogate-based optimization is used for the robust design of a propeller blade.

Finally, additional research work documented in this dissertation focuses on utilizing an optimization algorithm that mixes integer and continuous variables for the robust optimization of High Lift Devices.

Contents

List of figures

List of tables

Introduction

During the last decades, the aviation industry has substantially grown. Despite the developments for more efficient aircraft, the pollution due to air transport has rapidly increased. Hence, nowadays, the aeronautical community is called upon to enhance the environmental impact in the face of continuing expansion in demand for aviation. The Clean Sky 2 rationale states that the aviation sector is responsible for about 12% of transport emissions and 2% of all human-induced CO_2 emissions. The reduction of gas emissions and noise has been, and still is, a crucial point in programs like Clean Sky 2, Horizon 2020 or Flightpath 2050 vision of the Advisory Council for Aeronautics Research and Innovation in Europe. In particular, the target for 2050, compared to the year 2000 levels, is to achieve a 90% decrease in NOx emissions and a 75% reduction in CO_2 emissions per passenger-kilometer. Also, the perceived noise emission of flying aircraft reduction target is 65%.

To that end, the minimization of aircraft drag is a crucial point to diminish the emission of pollutants. Classically, deterministic aerodynamic design optimization problems were solved considering constraints in the aerodynamic performances and shape. However, the aerodynamics performance in real-world applications is inherently uncertain due to manufacturing tolerances, uncertain environmental conditions [1, 2], shape deformation under loads, debris accumulation on the wing surface, and other physical phenomena like the icing [3]. Therefore, uncertainties must be accounted for already during the design of aerodynamic shapes. Thus, robust and reliability-based design optimization is gaining popularity in the industrial context since this approach helps to obtain solutions that do not deteriorate excessively in the presence of uncertainties.

The language of probability theory is used to model these sources of uncertainty related to the real world or future events and states that are not explicitly foreseeable. In many approaches, random variables allow these types of uncertainty to be taken into account in optimization processes. Random variables are formally defined as measurable functions; consequently, objectives and constraints become functions that, in turn, must be remapped into real numbers representative of performance to proceed with the optimization process. Historically, several approaches have been used: the best estimate, the worst case, or the expected values with safety margins [4]. Recently, it has been introduced the use of risk measurement functionals, such as the Value-at-Risk (VaR) and the Conditional Value-at-Risk (CVaR), for robust design optimization [5]. The financial field introduced these measures to account for the risk of a portfolio. The focus of this dissertation is the efficient calculation of these risk measure functionals (or risk measures, for brevity). Their calculation requires evaluating the Empirical Cumulative Distribution Function (ECDF) of the Quantity of Interest (QoI). To accurately calculate the risk measure from the ECDF, numerous function evaluations are needed. Specifically, for high-fidelity aerodynamic shape design, the function evaluations require Computational Fluid Dynamics (CFD) runs as the QoIs are the aerodynamic characteristics. Typically, these CFD evaluations are expensive and burden the computational budget for the optimization problem.

Therefore, the research activity focus is investigating methods to reduce the CPU time for robust design optimization. To that end, two approaches to reduce computational time are investigated:

1. speeding up the CFD evaluations;
2. approximating the statistical measure.

The speeding up of the CFD computational time relies on improving the estimation error of aerodynamic forces on coarse meshes through a far-field approach. Indeed, CFD solvers use a near-field approach for the aerodynamic force calculation, but this requires a rather fine computational mesh to obtain acceptable results. Conversely, the introduction of a far-field formula allows calculating the drag coefficient with a precision similar to the near-field approach but using a coarser mesh. This result is possible because the far-field approach allows the identification and filtration of the spurious drag sources (truncation error and artificial dissipation). The far-field formula employed in the dissertation is based on entropy variations and was developed by Paparone and Tognaccini [6]. The Entropy Drag Concept was introduced by Oswatitsch [7]. Besides, in the literature, other authors, such as Destarac and Van Der Vooren [8] and Gariépy et al. [9], also studied and developed formulas for drag breakdown.

The approximation of the statistical measure relies on both intrusive and non-intrusive surrogate modelling approaches. The intrusive approach requires using the RANS adjoint solution to compute the gradient of the QoI with respect to the uncertainty sources. The computed gradient allows building a 1st order approximation of the QoI empirical cumulative distribution function (ECDF). The non-intrusive approach relies on an adaptive Gaussian processes approach to compute the ECDF approximation. Nevertheless, many other efficient uncertainty quantification methods are available in the scientific literature, although their consideration is beyond this dissertation scope. However, it is worth mentioning some of these methods to allow correct placement of this work in the context of the current state of the art of scientific research. Firstly, the Polynomial Chaos Expansion (PCE) [10–12]. PCE has excellent efficiency performance compared to the classical Monte Carlo (MC) sampling methods for industrial interest problems [13]. The major limitation of this method is the 'curse of dimensionality'. Indeed, for a large number of uncertainty sources, the method becomes computationally expensive. Hence, PCE can be a viable alternative to the methods proposed here when the number of random variables describing the uncertainty sources is low. Moreover, another non-intrusive method used for quantifying uncertainty is Stochastic Collocation (SC) [14, 15]. A comparison of Stochastic Collocation and Polynomial Chaos Expansion techniques is provided in [16]. Furthermore, other authors focused on developing advanced techniques based on the Monte Carlo sampling to overcome its slow convergence rate while still taking the advantage of its robustness and accuracy. An example of these methods is the Multi-Level Monte Carlo (MLMC) introduced by Heinrich [17] and Giles [18] and expanded to the computation of arbitrary order central statistical moments by Krumscheid et al. [19]. Also, an improvement of MLMC was presented by Pisaroni et al. [20] that introduced the Continuation Multi-Level Monte Carlo (C-MLMC) methodology.

The last part of the thesis deals with a different topic, even if it is closely related to what it is seen previously. In greater detail, the problem of robust design of High Lift Devices (HLD) is addressed. Hence the focus of the work is on the treatment, in the same optimization run, of configurations with a variable number of airfoil components. This implies that the optimizer must take into account both integer or enumerative variables as well as continuous variables. In particular, the discrete variables manage the topological changes of the configuration and the continuous ones the dimensioning and the reciprocal positioning of the components that make up the airfoil. Given the complexity and strong non-linearity of the search space, the angle of attack is the only uncertain parameter with respect to which to test the robustness of the solution. Classically, the design of HLDs is a two-steps procedure; first, a promising airfoil type (number of airfoil elements) is selected, and second, the optimal position and shape of the elements are obtained through an optimization run. Optimization algorithms that mix continuous and integer variables can improve classical design strategies by including configurational decisions, in this case, the airfoil type, in an automated optimization tool. The integration of these decisions, usually taken a-priori based on previous knowledge, represents a step forward for the multidisciplinary design optimization field. A surrogate-based approach is used to minimize the computational optimization cost, and a quadrature approach is employed to quantify uncertainty.

In summary, the dissertation follows the subsequent organization. Firstly, the advantages of robust optimization and the use of risk measures for uncertainty quantification are introduced in Chapter 1. Chapter 2 summarizes the optimization algorithms used and the tools needed for aerodynamic design optimization, with particular reference to the aerodynamic shape parametrizations, grid generators, and CFD flow solvers. The different methods to reduce the computational effort for robust optimization are the object of Chapter 3 to Chapter 5. In particular, the speeding up of the CFD evaluations through a far-field approach is discussed in Chapter 3. The statistical measure approximations through the adjoint method and Gaussian processes are in Chapter 4 and Chapter 5. Finally, the robust design of High Lift Devices through an optimization process that mixes continuous and integer variables is in Chapter 6. It should be emphasized that, although the thesis is entirely focused on increasing the computational efficiency of robust optimization methodologies, different paths have been explored which, however, are very interconnected and, in each of these, an original research contribution was presented, as shown in the highlights of the concluding chapter.

Summing-up, the key point addressed is the efficient use of novel and effective risk measures based on the cumulative distribution function, namely CVaR, and VaR, for robust aerodynamic design optimization using RANS high fidelity flow solvers. This goal is achieved through three converging strategies:

1. speeding up the CFD evaluation process using coarser grids while keeping reasonable precision in aerodynamic quantities estimation;

2. development of intrusive and non-intrusive methodologies for risk measure approximation;

3. reduction of the computational cost of robust aerodynamic optimization problems that combine continuous and discrete variables.

1
From deterministic to robust optimization

The theoretical background of optimization under uncertainty is introduced in the present chapter, starting from the classical deterministic optimization problem formulation. Several approaches to face robust or reliability-based optimization are given. Besides, the use of risk measures to solve these problems is addressed in more detail. In particular, the state-of-the-art Value at Risk and Conditional Value at Risk measures are introduced. At the end of the chapter, a 'toy' design problem example is provided to show the advantages of robust optimization. The purpose is to illustrate these new risk functions advantages with a practical example with issues very close to those of our interest. The chapter is based on Chapter 13 of the book Optimization Under Uncertainty with Applications to Aerospace Engineering [21].

1.1 Optimization under uncertainty

Classically, optimization problems are mathematically defined using the minimization formulation referred below:

$$\begin{cases} \min & f(\mathbf{z}) \\ \mathbf{z} \in S & \\ s.to: & \\ & c_i(\mathbf{z}) \leq 0 \quad i = 1, \ldots, m \\ & S \subseteq \mathbb{R}^n \end{cases} \tag{1.1}$$

where, the objective function is $f(\mathbf{z})$ and the vector $\mathbf{z}$ is referred to as the vector of design variables. Furthermore, the objective function may be subject to constraint functions, expressed by $c_i(\mathbf{z})$. Constraints functions can be linear or nonlinear functions of the design variables, and these functions can be either explicit or implicit in $\mathbf{z}$. A quite common convention, which does not affect the generality of the formulation, is to represent all the inequalities as non-positive ones. In addition, the problem (Eq. (1.1)) has been presented as a minimization problem, but some optimization problems might require maximization. Indeed, the maximization of $f(\mathbf{z})$ is always equivalent to the minimization of $-f(\mathbf{z})$ [22].

However, in most engineering problems, unknowns or future states must be considered. Moreover, they must account for the stochastic nature of the system and processes to be designed. For example, industrial manufacturing processes and real operating conditions inevitably introduce tolerances in the production and uncertainties in the working conditions, respectively, that will lead to deviations from the considerations taken at the design stage. Hence, random variables are introduced, and a stochastic optimization problem is defined to model the process under investigation correctly.

A random variable is defined as a measurable function $X : \Omega \mapsto \mathbb{R}$ that maps possible outcomes Ω to a measurable space $\mathbb{R}$, with $(\Omega, \mathcal{F}, P)$ a properly defined probability space, with $\omega \in \Omega$, $\mathcal{F} = 2^{\Omega}$, and P a probability measure [23].

Mathematically, the direct introduction of random variables into the optimization problem (Eq. (1.1)) introduces a functional dependency that must be appropriately treated to avoid inconsistencies. Indeed, the introduction of random variables leads to the following problem formulation:

$$\begin{cases} \min & f(\mathbf{z}, X) \\ \mathbf{z} \in S & \\ s.to: & \\ & c_i(\mathbf{z}, X) \leq 0 \quad i = 1, \dots, m \\ & S \subseteq \mathbb{R}^n \end{cases} \tag{1.2}$$

where the objective function and the constraints are now functions of functions. Therefore, a way to recast the problem into an optimization one must be searched by remapping these functions into real numbers representative of performance. Herein, several approaches are shown.

Best estimate: a particular outcome ($\bar{\omega} \in \Omega$) is chosen as the best estimate of the unknown status. As a consequence, the problem is, then, reconstructed as a deterministic optimization:

$$\begin{cases} \min & f(\mathbf{z}, X(\bar{\omega})) \\ \mathbf{z} \in S & \\ s.to: & \\ & c_i(\mathbf{z}, X(\bar{\omega})) \leq 0 \quad i = 1, \dots, m \\ & S \subseteq \mathbb{R}^n \end{cases} \tag{1.3}$$

Although this approach is attractive for its simplicity, this kind of alternative is very risky, as the choice of the typical outcome $\bar{\omega}$ is somewhat arbitrary and might not reflect at all what happens in reality.

On the other hand, this strategy can be extended by considering not a single outcome but a set of them. Notably, by choosing this set ad hoc, possibly resorting to an optimization process, it is possible to approximate more rigorous approaches, such as Monte Carlo sampling, while limiting the number of samples required. An example of this approach is given in the chapter on Gaussian processes.

Worst case: contrary to the best estimate strategy, the worst possible outcomes are identified for the unknown status. This leads to the following minmax problem:

$$\begin{cases} \min & sup f(\mathbf{z}, X(\omega)) \\ \mathbf{z} \in S & \omega \in \Omega \\ s.to: & \\ & c_i(\mathbf{z}, X(\omega)) \leq 0 \quad i = 1, \ldots, m \\ & S \subseteq \mathbb{R}^n \end{cases} \tag{1.4}$$

Two main disadvantages of this method are that minmax problems are generally very computationally expensive and that the obtained solution is too conservative. In addition, there is a high probability facing a non-feasible problem.

Expected values with safety margins: in this case, expectations, as well as standard deviations, are introduced in a weighted sum. Therefore, the robust design problem is cast in the following form:

$$\begin{cases} \min & E[f(\mathbf{z}, X)] + \lambda_0 \sigma(f(\mathbf{z}, X)) \\ \mathbf{z} \in S & \\ s.to: & \\ & E[c_i(\mathbf{z}, X)] + \lambda_i \sigma(c_i(\mathbf{z}, X)) \leq 0 \quad i = 1, \ldots, m \\ & S \subseteq \mathbb{R}^n \end{cases} \tag{1.5}$$

In this formulation, weighted sums of standard deviations can be interpreted as safety margins. Furthermore, this approach is widely used, although it can lead to severe problems such as the convergence to sub-optimal solutions due to the use of expectations that penalizes favorable and unfavorable candidate solutions in the same way.

Performance index on Cumulative Distribution Function (CDF): this approach is based on the definition of an ad-hoc performance index (or risk measure) as a function of the Cumulative Distribution Function related to the quantity of interest under investigation. For the sake of completeness, let us give the definition of the Cumulative Distribution Function (CDF). The CDF gives the area under the probability density function from minus infinity to x. It describes the probability that a real-valued random variable X with a given distribution will be found at a value less than or equal to x. Mathematically, this is expressed by

Eq. (1.6).

$$F_X(x) = P(X \leq x) \tag{1.6}$$

A performance index (or risk measurement) allows the comparison of different CDF shapes according to the risk criterion defined by the user. In this chapter, the risk measures used are the Value-at-Risk (VaR) and the Conditional Value-at-Risk (CVaR), also known as quantile and super-quantile, respectively (their definitions are given in Section 1.2).

1.2 Risk measures

When random events are modeled by random variables, as in the context of optimization under uncertainty, a way to measure risk should be figured out. With this purpose, a functional $\rho(X)$ is going to be defined for risk level quantification. Subsequently, an acceptable level of risk C must be decided, considering that there will, inevitably, be adverse events. Thus, the next inequality equation can be defined:

$$\rho(X) \leq C \tag{1.7}$$

Then, if the random variables representative of the cost depend on a deterministic decision vector **x** of size m, the following minimization problem can be stated.

$$\begin{cases} \min & \rho_0(X_0(x)) \\ \mathbf{x} \in S \subseteq \mathbb{R}^n & \\ s.to: & \\ & \rho_i(X_i(x)) \leq c_i \quad i = 1, \ldots, m \end{cases} \tag{1.8}$$

Within this framework, different definitions for risk functionals can be established. This will lead to different approaches to face optimization problems under uncertainty. The most immediate and familiar alternative of risk measure is the expected value. This means that, on average, it should be $X \leq C$:

$$\mu(X) \leq C \;\rightarrow\; \rho(X) = \mu(X) = EX \tag{1.9}$$

Being more stringent, a condition on the standard deviation or on variance could be imposed if there is a need to reduce the variation range of the quantity of interest:

$$\mu(X) + \lambda\sigma(X) \leq C \rightarrow \rho(X) = \mu(X) + \lambda\sigma(X) \tag{1.10}$$

$$\mu(X) + \lambda\sigma^2(X) \leq C \rightarrow \rho(X) = \mu(X) + \lambda\sigma^2(X) \tag{1.11}$$

Indeed, the classical robust design problem formulation is based on mean, μ, and variance, σ^2 [4], which can be treated as separated objectives in a multi-objective framework [24], as a weighted combination, or even cast into a constrained optimization format. However, the use of this classical approach may often generate some problems, since mean and variance are not independent measures, and it might be challenging to decide how much the mean must be penalized to get the desired reduction of variance.

Alternative risk measures offer better control on the desired features of the cumulative distribution function of interest. Here, in particular, the Value-at-Risk (VaR) and the Conditional Value-at-Risk (CVaR) are introduced and used. The main advantage of using these statistical measures is that they work asymmetrically. This is clearly beneficial for aerodynamic design optimization. For example, imagine that the objective of the optimization problem is to minimize the drag coefficient of an aerodynamic surface. The use of the mean value as a risk function penalizes any configuration that is far from the mean. Thus penalizing, in the same way, configurations that provide an increase or decrease on drag. However, only configurations that provide an increase on it should be penalized when dealing with the drag force. This can be done by using risk measures that work asymmetrically with the CDF, such as the VaR and CVaR.

Therefore, let us define the Value-at-Risk and the Conditional Value-at-Risk. Let X be a random variable and $F_X(x) = P(X \leq x)$ the Cumulative Distribution Function of X. Thus, the inverse CDF of X can be defined as $F_X^{-1}(\gamma) = inf\{x : F_X(x) \geq \gamma\}$. This function gives the minimum value of x that makes the CDF of X to be greater than or equal to γ. Hence, $\gamma - \mathrm{VaR}$, i.e. the Value-at-Risk for a given $\gamma \in (0, 1)$, is given by

$$\nu^\gamma = F_X^{-1}(\gamma) \tag{1.12}$$

In other words, VaR is the maximum loss that can be exceeded only in a $(1 - \gamma)100\%$ of cases. In its definition, the infimum is used since CDFs are, usually, weakly monotonic and right-continuous. The $\gamma - \mathrm{VaR}$ is shown in Fig. 1.2.1.

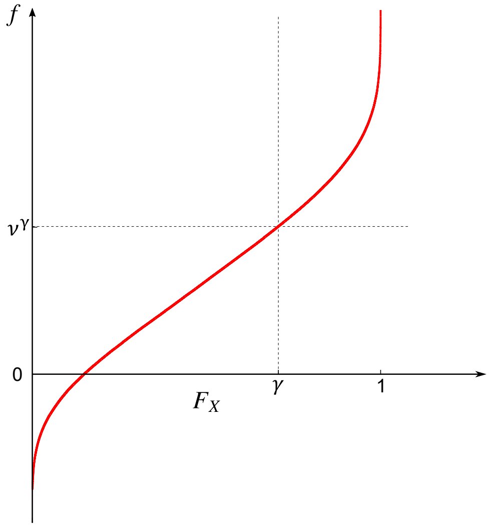


Figure 1.2.1: Value-at-Risk

The definition of Conditional Value at Risk is given below. Let X be a random variable, the $\gamma - \mathrm{CVaR}$ of X can be thought of as the conditional expectation of losses that exceed q_γ. From a mathematical point of view, CVaR is given by a weighted average between $\gamma - \mathrm{VaR}$ and the losses exceeding it.

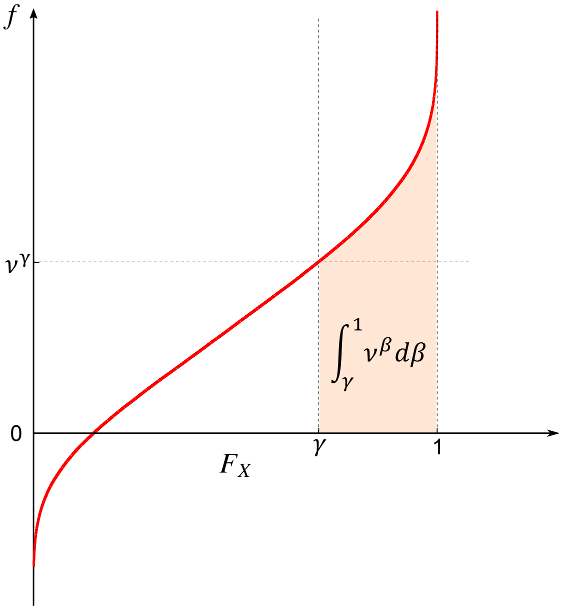


Figure 1.2.2: Conditional Value-at-Risk

The comparison of VaR and CVaR shows that the latter is more sensitive to the shape of the upper tail of the cumulative distribution. Summing up, the CVaR is expressed as:

$$c^{\gamma} = \frac{1}{1-\gamma} \int_{\gamma}^{1} \nu^{\beta} d\beta \tag{1.13}$$

The area measured by the integral of the $\gamma - \mathrm{CVaR}$ formula is highlighted in Fig. 1.2.2.

CVaR has the advantage, with respect to VaR, of being a coherent risk measure. The definition of coherency for a risk measure is a rigorous and well-defined mathematical concept that the interested reader can find in [25].

The properties that a risk measure must fulfill for being coherent [25] are enumerated below.

1. $\rho(C) = C$ for all constants C
2. *Convexity*: $\rho(1-\lambda)X + \lambda' \leq (1-\lambda)\rho(X) + \lambda\rho(X')$ for $\lambda \in (0,1)$
3. *Monotonicity*: $\rho(X) \leq \rho(X')$ if $X \leq X'$.
4. *Closedness*: $\rho \leq c$ when $X_k \rightarrow X$ with $\rho(X_k) \leq c$
5. *Positive homogeneity*: $\rho(\lambda X) = \lambda\rho(X)$ for $\lambda > 0$.

   From this last condition, the subsequent properties are derived:

   - *Translation invariance*: $\rho(X + C) = \rho(X) + C$.
   - *Sub-additivity*: $\rho(X + X') \leq \rho(X) + \rho(X')$.

Coherency properties offer several advantages in a robust optimization problem, and an actual robust aerodynamic shape design problem is used here to illustrate their meaning. The problem regards the improvement of the drag performance of a natural laminar flow wing and is described in detail in [26]. *Monotonicity* means that if the laminar performance of a generic wing X_2 is always better than X_1, then the risk of X_2 is always inferior to the risk of X_1. *Translation invariance* condition implies that a global delay of laminar to turbulent transition reduces the risk. Moreover, for a natural laminar flow wing-body, *sub-additivity* means that having two independent sources of laminarity (upper and lower wing surfaces) can only decrease risk. Summing up, coherency offers a way to consider the effect of desirable physical features in the risk measure used to formalize the robust design optimization problem to be solved.

Conversely, the Value-at-Risk is not a coherent measure since it does not respect the sub-additivity property.

1.3 Robust optimization problem using risk functions

Risk measures, likewise expectations and variance, are unknown parameters of a statistical model (estimands), which can only be approximated using estimators and finite samples of data. Consequently, the robust optimization problem has to be defined in terms of estimates of the risk functions within the framework of multi-objective optimization. Therefore, Eq. (1.8) becomes:

$$\begin{cases} \min & \hat{\rho}_{i;n}(\mathbf{z}) \quad i = 1, \dots p \\ \mathbf{z} \in Z \subseteq \mathbb{R}^n & \\ s.to: & \\ & \hat{\rho}_{i;n}(\mathbf{z}) \le c_i \quad i = p+1, \dots, p+q \end{cases} \tag{1.14}$$

where $\hat{\rho}_{i;n}$ is an estimate of the generic risk measure ρ_i obtained using a sample of size n and a proper estimator. Moreover, the constraints are also given taking into consideration a set of inequalities which are defined in terms of q further risk measure estimates.

In addition, it must be mentioned that the quality of the risk function estimate will directly influence the results of the optimization problem. Hence, some guidelines should be followed when formulating a robust optimization problem [26]:

- when estimating risk functions, use the largest possible number of samples considering the computational budget;
- use advanced techniques for sampling (i.e. multilevel Monte Carlo or Control Variates);
- low accuracy of the estimate can be perceived by the optimization algorithm as noise, thus select an optimization algorithm the least sensitive to noise as possible;
- employ advanced statistical methods for the evaluation of the estimate accuracy and confidence intervals. In particular, the bootstrap method will be adopted here (see Section 1.3.3).

The estimates of risk functions are here made either using the Empirical Cumulative Distribution Function (ECDF) or the Weighted Empirical Cumulative Distribution Function (WECDF), and the bootstrap is used to obtain accuracy and confidence intervals.

1.3.1 Estimation of risk functions using ECDF

The Empirical Cumulative Distribution Function (ECDF) is the distribution function associated with the empirical measure of a sample. Moreover, it can be seen as a step function that jumps up by $1/n$ at each of the n data points. It takes as value the fraction of observations of the variable that are less or equal to the specified value [27].

Mathematically speaking, let $X : \Omega \mapsto \mathbb{R}^d$ a random variable, $\mathbf{x}^i = (x_1^i, \dots, x_d^i)$ a random sample of X, μ a probability measure, and $\mathbf{t} = (t_1, \dots, t_d)$ a generic vector in $\mathbb{R}^d$. The ECDF is defined in Eq. (1.15) for n samples $\{\mathbf{x}^1, \dots, \mathbf{x}^n\}$.

$$\hat{F}_\mu^n = \frac{number\ of\ elements\ in\ the\ sample \leq \mathbf{t}}{n} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{\mathbf{x}^i \leq \mathbf{t}\} \qquad (1.15)$$

where $\mathbb{1}\{A\}$ is the indicator of event A:

$$\mathbb{1}_A(x) := \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \qquad (1.16)$$

and $\mathbf{x}^i \leq \mathbf{t}$ meaning $x_j^i \leq t_j, j = 1, \dots, d$. The last relation defines a partial order and if it is true, then $\mathbf{x}^i$ is either dominated by $\mathbf{t}$ or equal to it.

The estimation of Value at Risk and Conditional Value at Risk by means of the ECDF is explained in the following subsections.

Value at Risk (quantile) estimation using ECDF

Value at Risk for a scalar random variable X at a given confidence level γ can be directly computed from Eq. (1.15). Hence, if $X_1, X_2, \dots, X_n$ are n independent and identically distributed observations of the random variable X, then the estimation of the $\gamma - \mathrm{VaR}$ of X is given by

$$\hat{\nu}^{\gamma;n} = X_{\lceil n\gamma \rceil:n} = \hat{F}_n^{-1}(\gamma) \qquad (1.17)$$

where $X_{i:n}$ is the i-th order statistic from the n observations, and

$$\hat{F}_n(t) = \sum_{i=1}^{n} \mathbb{1}\{X_i \leq t\} \tag{1.18}$$

is the empirical CDF constructed from the sequence $\tilde{X}$ of $x_1, x_2, \ldots, x_n$. Note that the hat symbol (ˆ) indicates estimated quantities.

Cumulative Value at Risk (superquantile) estimation using ECDF

Regarding the estimation of the superquantile, according to [28], c^γ can also be written as a stochastic program:

$$c^\gamma = \inf_{t \in \mathbb{R}} \left\{ t + \frac{1}{1-\gamma} E[X - t]^+ \right\} \tag{1.19}$$

with $[a]^+ = \max\{0, a\}$. The set of optimal solutions to the stochastic program is $T = [\nu^\gamma, u^\gamma]$ with $u^\gamma = \sup t : F(t) \leq \gamma$. In particular, $\nu^\gamma \in T$, so

$$c^\gamma = \nu^\gamma + \frac{1}{1-\gamma} E[X - \nu^\gamma]^+ \tag{1.20}$$

When X has a positive density in the neighborhood of ν^γ, then $\nu^\gamma = u^\gamma$. Under these conditions, the above formula can also be directly derived from Eq. (1.13). So, in the case of a finite number of samples, with $X_1, X_2, \ldots, X_n$ independent and identically distributed (i.i.d.) observations of the random variable X, the estimation of c^γ is given by:

$$\hat{c}^{\gamma;n} = \hat{\nu}^{\gamma;n} + \frac{1}{n(1-\gamma)} \sum_{i=1}^{n} [X_i - \hat{\nu}^{\gamma;n}]^+ \tag{1.21}$$

1.3.2 Estimation of risk functions using WECDF

It was above stated that the ECDF is a step function that jumps up a fixed quantity, $1/n$, for each data point belonging to the sorted set of samples. Conversely, the WECDF can be considered as a step function that has a variable size jump, w_i:

$$\hat{F}^n_{\mu;\mathbf{w}}(\mathbf{t}) = \sum_{i=1}^{n} w_i \mathbb{1}\{\mathbf{x}^i \leq \mathbf{t}\} \tag{1.22}$$

with the related constraint

$$\sum_{i=1}^{n} w_i = 1 \tag{1.23}$$

The formula for VaR estimation starting from a WECDF is a generalization of Eq. (1.18), and requires two steps. Firstly, the k_γ index of the sorted sample set has to be chosen according to the following inequalities:

$$\sum_{k=1}^{k_\gamma} w_k \geq \gamma > \sum_{k=1}^{k_\gamma - 1} w_k \tag{1.24}$$

then $\hat{\nu}^{\gamma;n}$ $(\gamma - \mathrm{VaR})$ is simply given by choosing the k_γ^{th} element of ordered set:

$$\hat{\nu}^{\gamma;n}(x) = x_{(k_\gamma)} \tag{1.25}$$

Similarly, $\hat{c}^{\gamma;n}(x)$ $(\gamma - \mathrm{CVaR})$ is given by

$$\hat{c}^{\gamma;n}(x) = \frac{1}{1-\gamma}\left[\left(\sum_{k=1}^{k_\gamma} w_k - \gamma\right) x_{(k_\gamma)} + \sum_{k=k_\gamma+1}^{n} w_k x_{(k)}\right] \tag{1.26}$$

The use of WECDF becomes essential in cases in which the statistical sample has to be corrected or re-elaborated with some post-processing steps. This, for example, is the case of importance sampling, where the data set is sampled according to distributions that may differ substantially from those of the underlying random variables. Indeed, one of the possible approaches to the correct input distributions is the assignment of a different weight to each sample. In statistics, this method is called change of probability measure. In this field, several techniques have been developed [29], and, among these techniques, the one based on WECDF is thoroughly described in [30].

1.3.3 Bootstrap error analysis

As was previously mentioned, the robust optimization problem results are influenced by the quality of the risk function estimate. A possible approach to deal with this problem is the use of computational statistics methods, like the bootstrap, developed by Efron in 1979 [31]. As a general term, bootstrapping can be defined as an operation that will allow a system to self-generate from its small subsets. Hence, confining the definition to the statistical field, it is a computational re-sampling technique that provides the confidence intervals of statistics without a prior assumption about the type of the distribution function. In this work, it is used to assess the quality of the risk function estimates used in the optimization process.

Given a statistic $T(x_1, x_2, ..., x_n)$ evaluated on a set of data $\{x_1, x_2, ..., x_n\}$, the method consists of the following steps:

- Forming new sample sets $\{x_1^*, x_2^*, ..., x_n^*\}$, also known as bootstrap samples, of the same size of the real sample by performing a random selection of the original observation with replacement. Usually, the same observation is introduced several times in the bootstrap samples.
- Then, the statistic of interest $T(x_1^*, x_2^*, ..., x_n^*)$ is calculated for these new samples.

This statistic will show a probability distribution of its own. Thus, from this distribution, the confidence intervals of the risk functions, like VaR or CVaR, are obtained. In other words, the evaluation of confidence intervals would require repeated samples of a given population, but only one sample is available. Thus, the bootstrap method treats the real sample as a population, and the repeated samples needed for confidence interval estimation are obtained by re-sampling it with replacement.

Finally, it must be mentioned that, although very attractive for its simplicity, the bootstrap technique also has several disadvantages, thoroughly discussed in the scientific literature. Maybe the main drawback is that the bootstrap samples are related to the original (real) sample in the same manner that the original sample is related to the unknown population. Hence, if the original population sample is not sufficiently representative of the whole population features, then the confidence intervals computed by bootstrap might be completely misleading.

1.4 Application example

This section is aimed to give a simple but significant example of a robust aerodynamic design optimization problem. The example demonstrates the importance of using CVaR risk measure, one of the main points introduced in this dissertation for efficiently dealing with robust design in aerodynamics. The problem is focused on an airfoil in incompressible conditions subject to geometric and aerodynamic constraints. The goal is the improvement of the airfoil performance by changing its shape. When a robust version of this problem is faced, an optimal solution that is less vulnerable with respect to uncertainties in operating conditions and geometric shape is obtained. The baseline airfoil is the NACA 2412. The free-stream design conditions assumed are $M_\infty = 0$ and $Re_\infty = 0.5 \times 10^6$.

The airfoil performance is measured by a quantity of interest Q defined by drag coefficient c_d plus some constraints that are here considered as penalties. Consequently, the robust optimization problem requires the minimization of the $\gamma - \mathrm{CVaR}$ of Q, with γ set to 0.9. The equality constraints are the lift coefficient (c_l), which is fixed to 0.5, and the maximum thickness (t), which is fixed to the 12% of the airfoil chord (c). The inequality constraints are the trailing edge angle (TEA), which must be greater than or equal to 13°, the leading edge radius (LER), that must be greater than or equal to 0.7% of the chord, and the boundary layer transition point on the airfoil lower surface (XTR_{LOW}) that cannot be located at x/c greater than 0.95. A constraint on the pitching moment was not considered. In addition, an ERROR variable is set to 1 when the solver does not converge. In summary, the problem constraints are reported below:

$$\begin{cases} c_l = 0.5 \\ t/c = 0.12 \\ XTR_{LOW} \leq 0.95c \\ TEA \geq 13^\circ \\ LER \geq 0.007c \end{cases} \tag{1.27}$$

Hence, the robust optimization problem is formulated as:

$$\min_{\mathbf{z} \in Z \subseteq \mathbb{R}^n} \mathrm{CVaR}(Q) \tag{1.28}$$

with

$$Q = c_d + p^+(TEA, 13^\circ) + p^+(LER, 0.007c) \tag{1.29}$$

In this case, the constraints regarding the leading edge radius and the training edge angle are treated as quadratic penalties:

$$p^+(x,y) = \begin{cases} 0 & \text{if } x \geq y \\ (x-y)^2 & \text{if } x < y \end{cases} \tag{1.30}$$

Instead, the constraints on the c_l and on the thickness do not appear because they are automatically satisfied by the computation procedure by changing the angle of attack and by re-scaling the airfoil thickness to the assigned value.

The robust optimization problem is built by introducing uncertainties in the airfoil section shape that is parameterized as a linear combination of an initial geometry $(x_0(s), y_0(s))$, and some modification functions $y_i(s)$. Moreover, to describe geometry uncertainties, further $z_j(s)$ modification functions are introduced. So, the airfoil shape, including uncertainties, is described by

$$x(s) = x_0(s), \qquad y(s) = k\left(y_0(s) + \sum_{i=1}^{n} w_i y_i\right) + \sum_{j=1}^{m} U_j z_j \tag{1.31}$$

where the airfoil shape is controlled by the design parameters w_i and by the scale factor k. The uncertainty on shape and thickness of the airfoil is described by the U_j random variables. In this optimization problem 20 uniform random variables, in the range $[-0.1, 0.1]$, have been used. Moreover, the population is generated through a Monte Carlo algorithm and it has a size equal to 100. It is important to note that the airfoil is rescaled to the assigned thickness before the application of the random variables that describe the uncertainty in shape.

The performances of the parametric airfoil obtained by using Eq. (1.31) are computed by an aerodynamic analysis code, namely Prof. Drela's XFOIL code [32]. It is based on a second-order panel method interactively coupled to a boundary layer integral module. Moreover, the laminar to turbulent flow transition is predicted using the e^N method [33].

The optimization algorithm selected for solving the described design problem is the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [34], which is a stochastic optimization algorithm based on self-adaptation of the covariance matrix of a multivariate normal distribution. It is mainly used for design optimization problems up to a few hundreds of design variables. The parameters used for the optimization algorithm are the maximum number of allowed evaluations, the population size λ, and the initial standard deviation σ. The parameters set for this problem are reported in Table 1.4.1.

Maximum evaluations	Population size	Initial standard deviation
7000	20	0.1

Table 1.4.1: Optimization Parameters for the optimization

Furthermore, it was mentioned that in case of non-convergence of the solver, an ERROR flag was set equal to 1. When robust optimization is faced, the treatment of theses cases is crucial for the optimization process. The quantity of interest to be minimized, CVaR, depends on the upper tail of the Cumulative Distribution Function. As a consequence, assigning a high value to the objective in the cases where convergence is not achieved implies a too high CVaR value that could be detrimental for the optimization algorithm behavior. Hence, a proper value of the objective in these cases must be decided. In particular, in this optimization problem, the worst objective value selected between the properly converged cases, is assigned to those in which the error flag is set. Numerical tests lead to conclude that this was the setup with the lowest impact on the optimization process behavior.

1.4.1 Results

The obtained results are here commented on and compared with the baseline airfoil and the equivalent deterministic solution of the optimization problem. Firstly, in Fig. 1.4.1, the airfoil shape of the robustly optimized airfoil (dotted line) is compared with the baseline airfoil (solid line) and the deterministic optimized airfoil (dashed line).

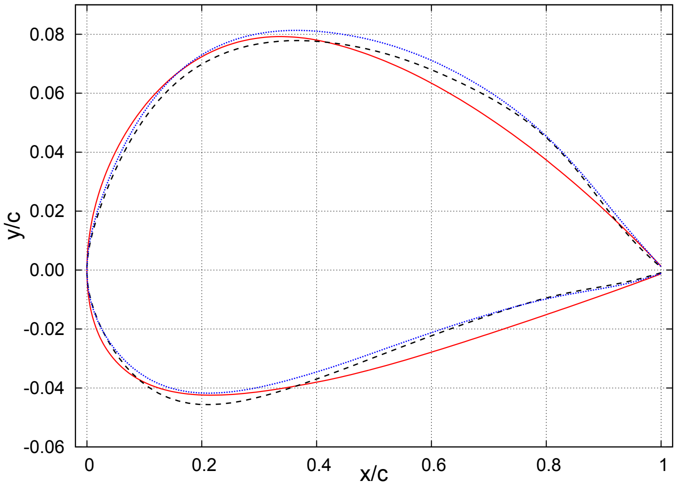


Figure 1.4.1: Airfoil shape comparison of the robust optimized airfoil (···), deterministic optimized airfoil (– – –) versus the baseline NACA 2412 airfoil (—)

In addition, the Cumulative Distribution Function obtained by introducing uncertainties in the airfoil shape is reported in Fig. 1.4.2 for the initial NACA 2412 airfoil (solid line), for the deterministic optimized airfoil (dashed line), and for the robustly optimized airfoil (dotted line).

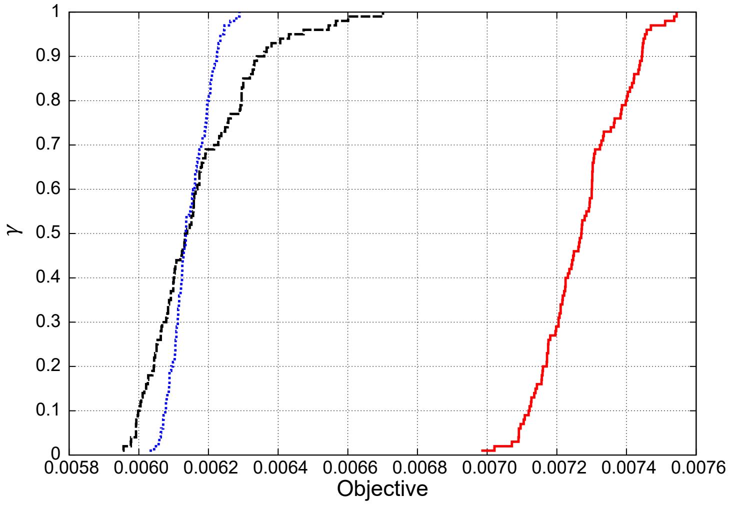


Figure 1.4.2: CDF obtained by the variation in airfoil shape using the robustly optimized airfoil (···), the deterministically optimized airfoil (– – –), and the baseline NACA 2412 airfoil (—)

The comparison of the CDFs related to the deterministic and robust airfoils highlights that the robust optimal solution is less vulnerable to uncertainties in geometric shape with respect to the deterministic one. The CVaR risk measure asymmetry allows a significant improvement of the cumulative distribution function upper tail without deteriorating, in the same manner, the lower one. This is a decisive advantage of CVaR compared to classical measures, such as mean or standard deviation. Working asymmetrically with the CDF is extremely important for aerodynamic optimization, where the quantity of interest is often the drag coefficient. Specifically, only configurations that provide an increase of c_d, compared to the nominal condition, have to be penalized since a decrease of drag is beneficial for the design. Moreover, the improve of the upper tail is also deduced by the observation of the value Value at Risk and Conditional Value at Risk with $\gamma = 0.9$ provided in Table 1.4.2.

	$0.9 - \mathrm{VaR} \cdot 10^4$	$0.9 - \mathrm{CVaR} \cdot 10^4$
Baseline NACA 2412 airfoil	74.5	75.0
Deterministic optimized airfoil	63.4	64.8
Robust optimized airfoil	62.2	62.5

Table 1.4.2: Risk measurement based on the obtained Cumulative Distribution Functions

While the advantages of CVaR are indisputable, it should also be noted that characterizing a probability tail is often much more challenging, from the point of view of the required samples, than estimating first and second-order moments. Here, for example, 100 samples are needed to get an acceptable description of the tail. Hence, this approach to robust optimization is practically inapplicable when the cost of a single sample evaluation is already expensive (as in the case of RANS computational fluid dynamic solvers). Thus, summarizing, CVaR based robust optimization is beneficial for aerodynamic design, but it is computationally expensive. Therefore, original methods that reduce the computational cost associated with robust aerodynamic design optimization problems are studied in this work.

2
Aerodynamic Design Optimization

This chapter copes with the two main ingredients needed for solving Aerodynamic Design Optimization (ADO) problems; a self-operating aerodynamic computational chain and the optimizer. First of all, the optimizer employed in the solved design problems is discussed in Section 2.1, and later, the components required for an autonomous aerodynamic computational chain are described in Section 2.2.

2.1 ADGLIB Library - Optimization algorithms

In this section, the ADGLIB [35] library used to solve the aerodynamic design problems in the subsequent chapters is introduced. ADGLIB (ADaptive Genetic algorithm LIBrary) is an in-house developed evolutionary optimization software library based on the hybridization concept. ADGLIB, indeed, allows to easily define evolutionary optimization algorithms that combine, for example, classical bit-string-based genetic algorithms with hill-climbing specialized operators. The choice is motivated by the need to improve the efficiency of genetic algorithms while keeping their flexibility in searching large design spaces. The specific implementation allows the use of the genetic algorithm hybridized either with gradient-based operators, like BFGS, or other sophisticated evolutionary algorithms like CMA-ES. These operators can also be used in a stand-alone fashion for a plain gradient-based or CMA-ES optimization run. ADGLIB is fully parallel at the level of population evaluation. The parallel programming model relies on shared memory multiprocessing, and the parallelism is implemented at the thread level using the standard POSIX thread interface. ADGLIB has a user interface that allows easy coupling with external codes, like flow solver or, more generally, numerical computation procedures. The interface relies on file templates that enable a fully automated customization of the input files required by the coupled numerical analysis procedures. The procedure output is retrieved through a flexible and parametric parsing interface that assigns the computation results to internal ADGLIB variables. ADGLIB allows the definition of scalar and vectorial input/output variables and constants. ADGLIB is capable of performing both single and multi-objective optimization.

Here, only two of the available optimization algorithms in the library are used, a classical Genetic Algorithm (GA) and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Both algorithms are Evolutionary Strategies (ES) that emulate the mechanisms typical of biological evolution. These techniques date back to the 50s [36] and have been used in many fields such as biology, economy, and engineering. In very general terms, evolutionary optimizers are loosely inspired by the biological principle of the survival of the fittest that underlies natural evolution. Of course, natural selection cannot be recast as an optimization process, but its principles work quite effectively to develop optimization algorithms useful in engineering practice.

In Sections 2.1.1 and 2.1.2, a brief description of the genetic algorithm and the covariance matrix adaptation evolution strategy algorithm is given.

2.1.1 Genetic Algorithm (GA)

Genetic algorithms (GAs) are population-based algorithms: they work with a population of candidate solutions according to an intrinsically parallel paradigm. The practicality of using the GA to solve complex problems was demonstrated by De Jong [37]. In Algorithm 1, the pseudocode of a simple GA is provided.

Algorithm 1 states that from a starting initial population, a variation is performed to create an offspring, then the whole combined population is evaluated, and finally, some individuals are selected to be kept for the following generation. This procedure is repeated

Algorithm 1 Genetic Algoritm

```
$t = 0.$
$P(t) \leftarrow$ Initialize the population with size $\lambda$
Evaluate $P(t)$
while Stopping criteria not fulfilled do
    $P^{'}(t) \leftarrow$ variation $P(t)$
    Evaluate $P^{'}(t)$
    $P(t+1) \leftarrow$ selection from $Q \cup P^{'}(t)$
    $t = t + 1$
end while
```

up to fulfilling the selected stopping criterion. A stopping criterion example is the maximum number of generations.

In a genetic algorithm, each individual is defined by a sequence of genes that form a chromosome. The chromosome encodes the value of the design variables of the problem. Classically, each design variable is encoded through a fixed-length binary number, and a single string, namely the chromosome, is generated by linking all the coded variables.

The evaluation of the fitness of each individual of a population is done to obtain a ranking based on a problem-dependent criterion.

The variation procedure allows the improvement of the individuals by means of two operators, namely crossover and mutation. The crossover operator recombines the chromosomes of two individuals (parents) to create offspring. Additionally, the mutation operator consists of a random change of the chromosome within an individual. There are several mutation and crossover operators, and their choice strongly influences the optimization problem [38].

Finally, the selection process consists of choosing two parent individuals from the population that are going to be reproduced. The probability of selecting one individual of the population will, generally, depend on its fitness. Therefore, the best individuals have a better chance to reproduce while the less fit ones are discarded. The selection mechanism implemented in the library is the random-walk [39, 40].

2.1.2 Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

The second optimization algorithm, and prevalently preferred in this dissertation, is the "Covariance Matrix Adaptation Evolution Strategy" (CMA-ES). The algorithm implemented in ADGLIB is the one developed by Hansen [34]. The CMA-ES is an evolutionary algorithm for difficult non-linear non-convex black-box optimization problems in the continuous domain. The CMA-ES algorithm is used for unconstrained or bounded constraint optimization problems, in which the dimension of the search space lies between three and a hundred.

As introduced in [41], at each generation of an Evolution Strategy (ES), a new population (of size λ) is generated in agreement with a multivariate normal distribution in $\mathbb{R}^n$, with n the number of design variables.

$$x_k^{(i+1)} \sim \mu^{(i)} + \sigma^{(i)}\mathcal{N}\left(0, C^{(i)}\right) \sim \mathcal{N}\left(\mu^{(i)}, (\sigma^{(i)})^2, C^{(i)}\right) \quad k = 1, \cdots, \lambda \quad (2.1)$$

In Eq. (2.1), $x_k^{(i+1)}$ indicates, inside the generation $i+1$, the $k-th$ individual. Moreover, $\mu^{(i)}$ denotes the mean of the distribution at the generation i and $\sigma^{(i)}$) is a scaling parameter or step-size. Finally, $C^{(i)}$ expresses a scaled covariance matrix of the distribution. The covariance matrix represents the dependencies of the variables in the distribution.

At each generation, a recombination operator is used to update the mean value of the distribution. For this algorithm, the Covariance Matrix Adaptation (CMA) is the method used to update the mean, the standard distribution, and the covariance matrix of the multivariate normal distribution at each generation. Particularly, to adapt the covariance matrix in the CMA-ES algorithm only a ranking of the candidate solutions is needed. Further details on the sorting, recombination, selection, covariance adaptation, and step-size control processes are, for example, in Pisaroni [41].

2.1.3 Robust optimization using ADGLIB

ADGLIB allows a relatively simple and flexible implementation of robust optimization algorithms. It is possible to operate on two levels. At the upper level ADGLIB allows to operate on the design variables and to manage the chosen optimization algorithm. In contrast, in the lower level, it takes care of generating the sampling of the random variables that characterize the uncertainty of the problem under examination. Therefore, according to this two-level approach, the upper level view as single individuals entire subpopulations of elements calculated at the lower level and defined by the sampling of random variables, whose distributions can be precisely defined. This procedure allows obtaining the ECDF

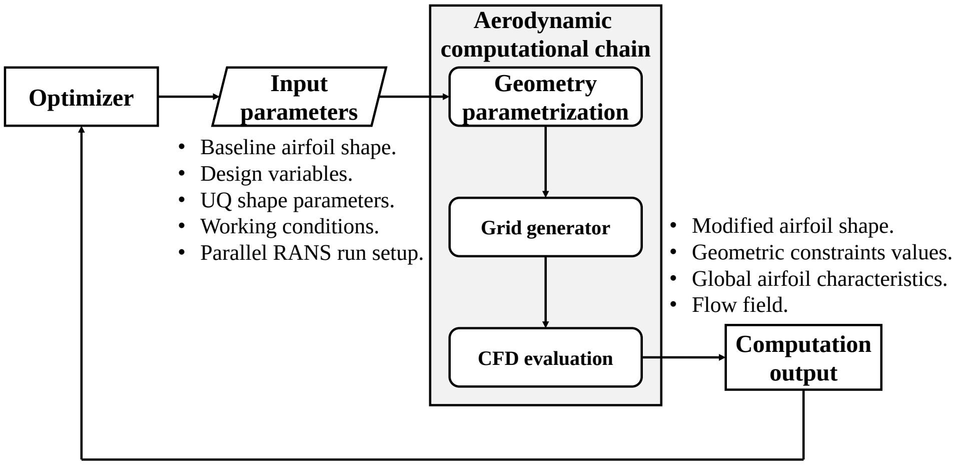


Figure 2.2.1: Aerodynamic design optimization (ADO) flowchart.

of the QoI to optimize. The performance of each individual subpopulation is calculated at the higher level through appropriately defined risk functions that operate on the aforementioned ECDF. Furthermore, it is also possible to evaluate the confidence intervals with the bootstrap technique. Sampling generation at the lower level can be done in various ways. In this thesis, the ECDF approximation modalities with the gradient-based method and the Gaussian processes have been added to the pre-existing Monte Carlo sampling

2.2 Aerodynamic computational chain

A fully automated aerodynamic computational chain is essential to face aerodynamic design optimization problems. In particular, the input design variables coming from the optimizer have to generate the geometric shape to be evaluated and, consequently, the related computational mesh. Then, the aerodynamic flow solver must calculate the aerodynamic performance of the candidate. Once the performance figures are obtained and did not occur errors in the computational chain, these values are feed to the optimizer. This process is depicted in Fig. 2.2.1, and each component is explained in depth in the following sub-sections.

2.2.1 Geometry parametrization

Depending on the chosen geometry parametrization, the number of shape design variables used for the optimization problem may vary. This block generates the candidate to be studied from the given design variables of the optimizer. Here-in, the used parametrization tools are described.

Airfoil parametrization using $wg2aer$ tool

The $wg2aer$ tool is an in-house developed program that generates a candidate airfoil from an initial geometry given some design parameters (optimization variables). When the design parameters are set to 0, the baseline airfoil is produced. This procedure works for single/multi-component airfoils. In the subsequent paragraphs, the parametrization for both types of airfoils is presented.

Regarding the single component airfoil, the shape description approach is derived from those presented in [42, 43] and is tailored to deal with uncertainties in the description of the airfoil shape. Particularly, the airfoil shape is parameterized as linear combination of an initial geometry, defined parametrically by $(x_0(s), y_0(s))$, and a number of modification functions $y_i(s)$ that may be defined analytically or by point distributions [44]. The possible modification functions are shown in Fig. 2.2.2. Moreover, the same technique is used to describe the uncertainties in the geometry introducing, additionally, $z_j(s)$ modification functions that are depicted in Fig. 2.2.3. So that, the airfoil is described as:

$$\begin{aligned} y(s) &= k\left(y_0(s) + \sum_{i=1}^{n} w_i y_i(s)\right) + \sum_{j=1}^{m} U_j z_j(s), \\ x(s) &= x_0(s) \end{aligned} \tag{2.2}$$

where the airfoil shape is controlled by the design parameters w_i and by the scale factor k for fulfilling the maximum thickness criterion. A maximum of 20 design parameters can be introduced. The uncertainty on shape is controlled by the U_j random variables.

Specifically, in order to consider the airfoil shape uncertainty, Hicks-Henne bump functions are introduced and are reported in Fig. 2.2.3. In particular, geometrical uncertainties

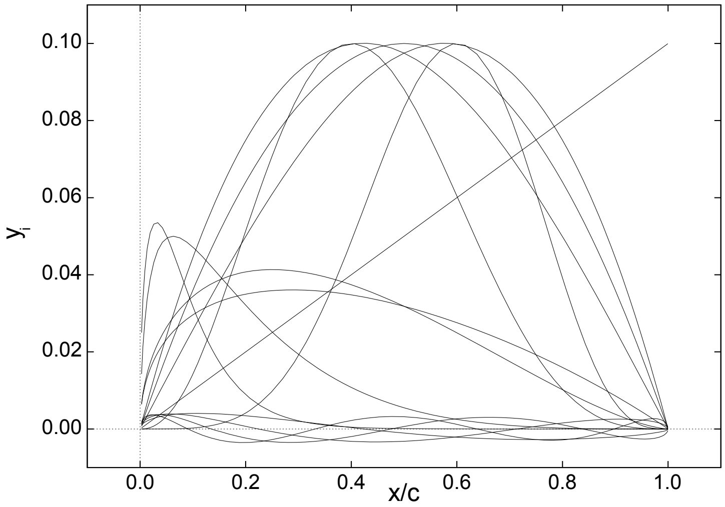


Figure 2.2.2: Modification functions that describe the variation of airfoil shape.

are modeled by means of the following expression:

$$z_j(s) = \sin^3\left(\pi x^{\frac{\log 0.5}{\log s_{b_j}}}\right) \tag{2.3}$$

where s_{b_j} is the position along the chord of the j-th bump. Note that the superscripts u and l in Eq. (2.4) denotes, respectively, the upper and lower surface.

$$s_b^u = s_b^l = \{0.05, 0.1, 0.2, 0.4, 0.6, 0.8\} \tag{2.4}$$

Some examples of how the modification of the design parameters influences the shape of the airfoil are now given. The baseline airfoil for these examples is the Martin Hepperle MH114 airfoil.

One possibility is to use the design parameters to define a thickness mode. To do so, a polynomial modification function is applied for the upper and lower airfoil surfaces. The design parameters that control the modification function for the upper and lower surfaces must have equal value but the opposite sign ($w_u = -w_l$). Moreover, a camber mode can also be obtained by applying a polynomial function but this time making equal, in value and sign, the design parameters ($w_u = w_l$). These airfoil modifications are de-

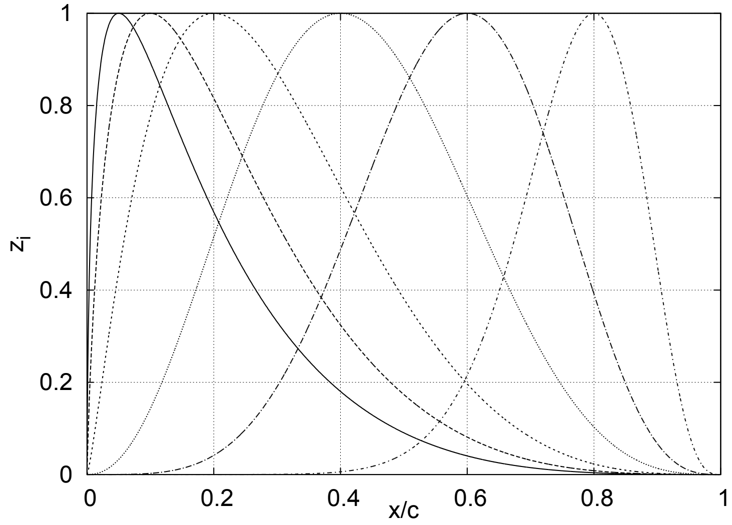
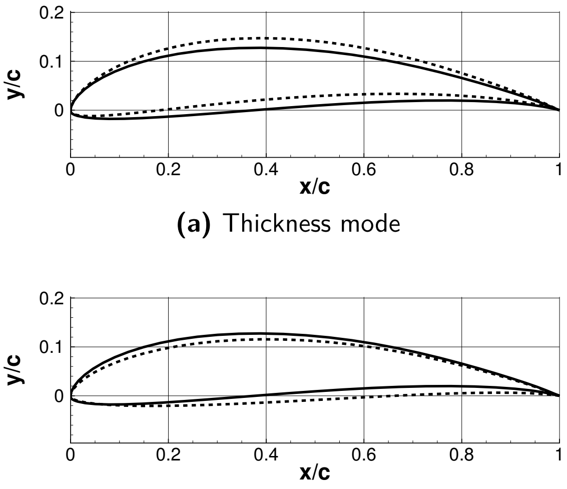


Figure 2.2.3: Hicks-Henne bump functions that describe the uncertainty in airfoil shape representation.

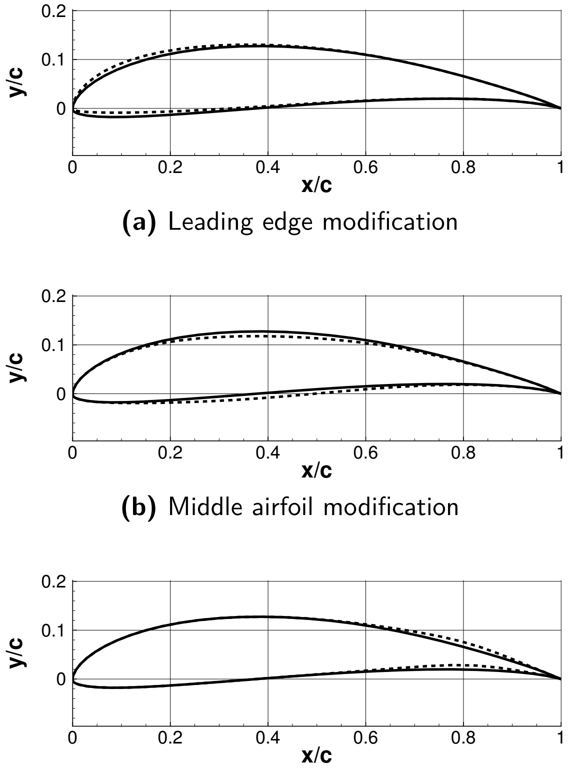


(a) Thickness mode

(b) Camber mode

Figure 2.2.4: Airfoil thickness and camber modification using $wg2aer$ tool. The baseline MH114 (**—**) and the modified airfoil (**- - -**).

picted in Fig. 2.2.4. Additionally, local airfoil modifications can be performed by using Hicks-Henne bump functions along the airfoil chord. In Fig. 2.2.5, modifications applied at different airfoil locations, x/c, are given.



(a) Leading edge modification

(b) Middle airfoil modification

(c) Trailing edge modification

Figure 2.2.5: Airfoil local modification using $wg2aer$ tool. The baseline MH114 (—) and the modified airfoil (– – –).

The presented tool can also be used to modify multi-component airfoils. In this case, the shape of the airfoil is invariant and the design variables are the settings of each airfoil component, which are the location in the 2-dimensional space and the rotation with respect to the starting airfoil ($\Delta X, \Delta Y,$ and, $\Delta\theta$). The reference point for the rotation is the leading edge of the corresponding element. Hence, the optimal position of the components of an airfoil can be found for a given aerodynamic design optimization problem.

An example of how the airfoil components can be translated and rotated is shown in Fig. 2.2.6. The baseline is a multi-component airfoil, obtained modifying the three element McDonnell Douglas (MDA) 30P-30N airfoil. In detail, the slat and main component are kept as the original, while the flap is modified into a triple slotted flap. The $wg2aer$ tool is then used to change the setting of the three flaps.

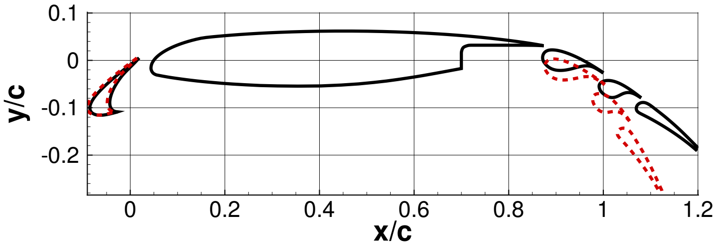


Figure 2.2.6: Airfoil components translation and rotation example. The baseline (—) and the modified airfoil (- - -).

Non-Uniform Rational B-Splines

Another parametric airfoil shape representation is the Non-Uniform Rational B-Splines (NURBS). This methodology allows the representation of complex geometries by means of a set of control points and an associated weight to each control point. In particular, a pth-degree NURBS curve can be mathematically defined by

$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \mathbf{P_i}}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \quad a \leq u \leq b \tag{2.5}$$

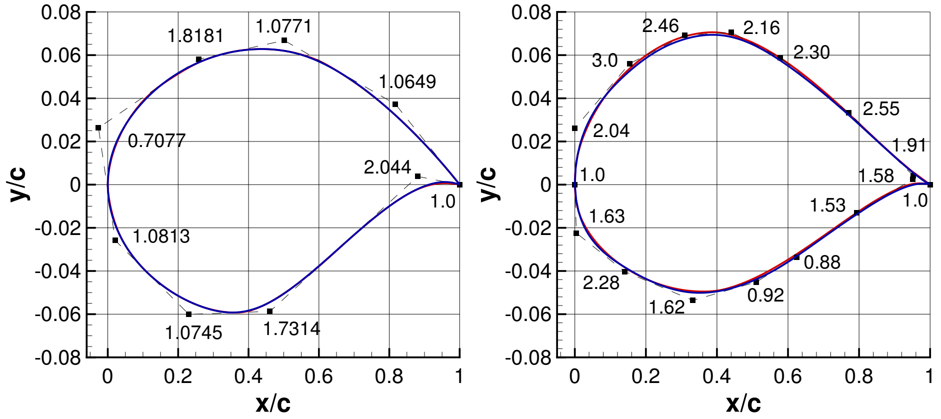
where w_i are the weights, $\mathbf{P_i}$ are the control points, and $N_{i,p}(u)$ are the pth-degree B-spline basis functions defined on the knot vector (U)

$$U = \underbrace{a, \ldots, a}_{p+1}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{b, \ldots, b}_{p+1} \tag{2.6}$$

For further details of the description of NURBS representation together with its mathematically properties refer to Ref.[45, 46]. The NURBS-Python library has been used[1].

Two examples of an airfoil (RAE 2822 and NACA $64(1) - 212$) representation by means of NURBS methodology are provided in Fig. 2.2.7. It can be seen how the original airfoil (*in red*) and the obtained airfoil (*in blue*) match perfectly. In addition, in Fig. 2.2.7, the control points and their associated weight used to represent the airfoil are also given. With this airfoil representation, the design optimization variables can be the control points position (x/c and y/c) and their associated weights.

[1] NURBS-Python Documentation: `http://nurbs-python.readthedocs.io/en/latest/`



(a) RAE 2822

(b) NACA 64(1)-212

Figure 2.2.7: Airfoil parametrization by means of NURBS representation. Original airfoil (—), NURBS airfoil representation (—), and NURBS control points (●). The x-axis and y-axis are not dependent for visualization purposes.

2.2.2 Grid generation

Once the aerodynamic surface candidate is obtained, the computational grid must be automatically generated. Several grid generators are used in this dissertation and are explained here in.

Grid generation using *Gmsh*

An automatic in-house procedure for mesh creation, suitable for Reynolds-averaged Navier–Stokes (RANS) simulations, was developed based on the open-source grid generator Gmsh [47]. The mesh generation procedure is able to create different types of unstructured mesh (O-grids and/or square grids) made of triangular and rectangular elements. The far-field can be placed at a selected position that is given as an input to the procedure. In addition, some peculiar features were also added. Particularly, the presence of a wake is considered, and also, the height of the near wall area region is estimated by the flat plate theory. The near wall area is made of rectangular elements, while the rest of the grid by triangles. In Fig. 2.2.8, an example of a squared grid with the far-field placed at $40c$ is shown. Moreover, the refinement at the wake is also present.

A similar process to the near wall area, is applied to calculate the proper value of the

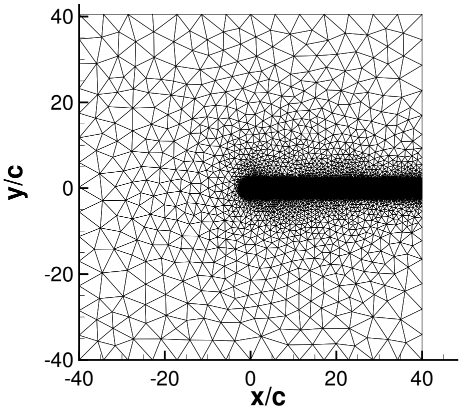
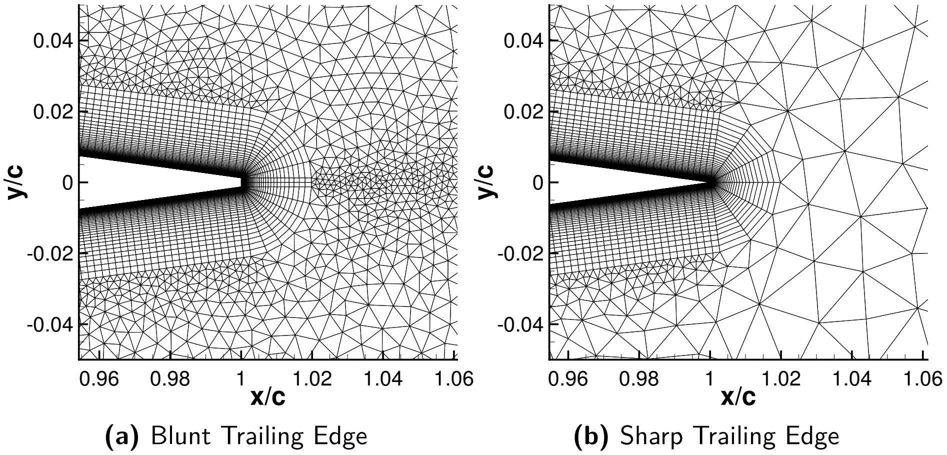


Figure 2.2.8: Example of a complete grid using *Gmsh*.

non-dimensional wall distance of the first grid element (y^+). Another control parameter is the size ratio between the cells at the far-field and the cell on the body surface to obtain coarser or finer grids.



(a) Blunt Trailing Edge

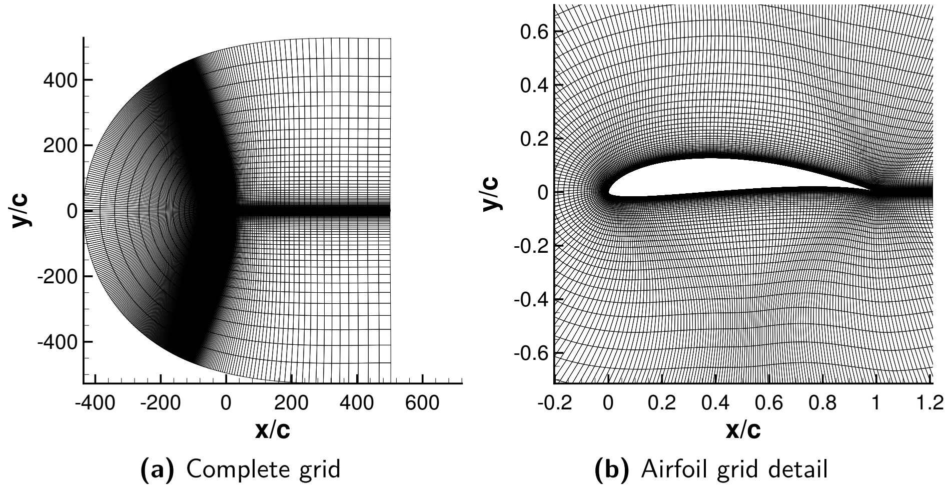
(b) Sharp Trailing Edge

Figure 2.2.9: Mesh detail at the trailing edge of an open and a sharp NACA 0012 airfoil using *Gmsh*.

Additionally, the procedure can manage airfoil with blunt (open) or sharp (closed) Trailing Edge (TE), and single and multi-component airfoils. In Fig. 2.2.9, grids generated with the *Gmsh* procedure around an open and sharp NACA 0012 airfoil are shown, particularly, a detail at the trailing edge of the airfoil. Furthermore, it is seen how the near wall area is made by rectangles, and after that, the triangular cells start.

Grid generation using *Construct2D*

The open-source grid generator *Construct2D* is designed to create 2D grids suitable for CFD computations on airfoils [48]. The hyperbolic or elliptic grids are generated in Plot3D format. However, this format is not appropriate for the flow solvers used (see Section 2.2.3); thus, the source code was modified to provide the grid in the needed formats. *Construct2D* is able to generate grids with O topology or C topology recommended for airfoils with blunt or sharp trailing edges, respectively. The generated grid is made of rectangular cells. Moreover, it is possible to generate hyperbolic or elliptic grids.



(a) Complete grid

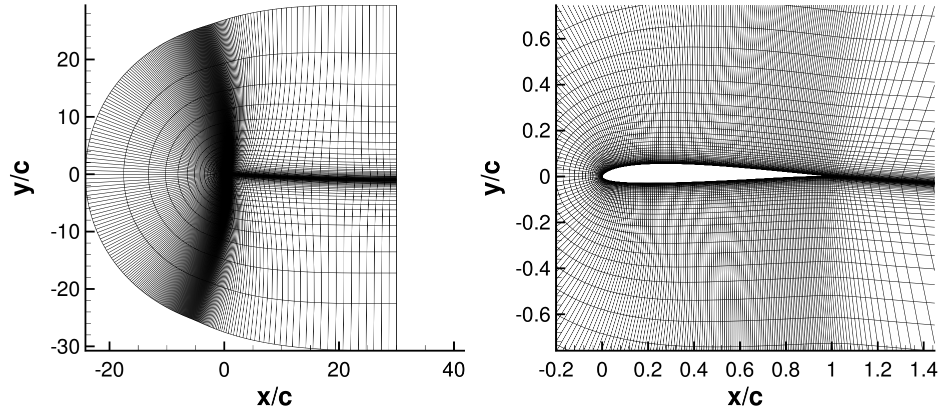
(b) Airfoil grid detail

Figure 2.2.10: C-grid generated with *Construct2D* for the MH 114 airfoil.

The required input is a file containing the set of coordinates defining the airfoil geometry, and a file containing some user-defined control parameters. Some of these parameters are the topology of the grid (C or O), the far-field position, the number of cells on the body surface, in the far-field direction, and in the wake. In addition, a smooth airfoil surface cell spacing can be controlled by setting the length of the cell at the leading edge and at the trailing edge. Finally, the Reynolds number of the CFD simulation and desired y^+ can also be given as an input, thus controlling the wall distance of the first grid element. An example of C-grid generated with *Construct2D* is given in Fig. 2.2.10.

Grid generation using *HypGen*

HypGen is an in-house developed automatic hyperbolic 2D grid generator, suited for structured meshes over a single component airfoil. It is able to generate C topology grids. It can be used to generate grids for both Euler and RANS applications. Contrary to the previous grid generators, *HypGen* only works with sharp trailing edge airfoils. If the airfoil has a blunt trailing edge, the procedure collapses a point in order to close the airfoil. The collapsing is performed automatically and with little control by the user.



(a) Complete grid

(b) Airfoil grid detail

Figure 2.2.11: C-grid generated with *HypGen* for the SD 7003 airfoil.

In order to generate the mesh, the required inputs are the airfoil coordinates, and optionally, some other grid information. The additional information is the far-field distance, the number of grid levels for multigrid/multilevel approach, and the numbers of grid points on the body surface, in the far-field direction, and in the wake. Furthermore, the height of the first cell can be selected as well. In Fig. 2.2.11, a grid generated using the explained procedure is provided. The grid is generated over the SD 7003 airfoil.

2.2.3 Computational Fluid Dynamics flow solver

In the following paragraphs, the Computational Fluid Dynamics (CFD) flow solvers used to calculate the aerodynamic quantities of interest for the optimization design problems are presented.

The Unsteady Zonal Euler Navier-Stokes flow solver

The Unsteady Zonal Euler Navier-Stokes (U-ZEN), an in-house flow solver for Reynolds-averaged Navier–Stokes (RANS) simulations, is a structured multi-block code for the analysis of complex configurations in the subsonic, transonic, and supersonic regimes [49]. The code is capable of simulating steady and unsteady RANS, and several turbulence models are available.

The governing equations are discretized by means of a standard cell-centered finite volume scheme with blended self-adaptive second and fourth-order artificial dissipation. The pseudo time-marching advancement is performed by using the Runge-Kutta algorithm with convergence accelerators such as the multi-grid and residual smoothing techniques. For unsteady simulations, the dual time stepping technique is adopted.

The turbulence equations are weakly coupled with the RANS equations and solved only on the finest grid level of a multi-grid cycle. Algebraic, one-equation, two-equations, and non linear eddy viscosity turbulence models are available. Transitional flows can be simulated using a one-equation transition model coupled to the turbulence model or fixing a transition line.

Stanford University Unstructured (SU2) tool suite

The Stanford University Unstructured (SU2) tool suite is an open-source computational analysis and design software aimed to solve complex, multi-physics analysis and optimization tasks using unstructured meshes. As described in Palacios et al. [50] and Economon et al. [51], the core of SU2 is a collection of C++ software tools that discretize and solve problems described by Partial Differential Equations (PDEs). Moreover, it is able to solve PDE-constrained optimization problems, such as optimal shape design.

In references [50, 51], the main C++ software tools are described. For this dissertation, the C++ modules of interest are the SU2_CFD (Computational Fluid Dynamics Code), SU2_CFD_AD (Discrete Adjoint code), and SU2_DOT_AD (Gradient Projection Code) software. SU2_CFD (SU2_CFD_AD) solves direct (adjoint) (steady or unsteady) problems for the Euler, Navier-Stokes, and Reynolds-Averaged Navier-Stokes (RANS), plasma, free-surface, electrostatic, etc., equation sets. It can be run serially or in parallel by using MPI. Moreover, explicit and implicit time integration methods are available with centered or upwinding spatial integration schemes. The software also has several advanced features to improve robustness and convergence, including residual smoothing, agglomeration multigrid, linelet and low speed preconditioning, and Krylov space methods for solving linear systems.

Furthermore, SU2_DOT_AD provides the gradient availability, which is used in Chapter 5. There are many applications such as optimization, response surface formulations, uncertainty quantification, among others, in which it is important to obtain gradients of the responses computed by SU2 to variations of design parameters. For this reason, SU2 relies on adjoint solver implementations that can be used to compute the necessary gradients. In this dissertation, the adjoint solver will be used for uncertainty quantification.

3
Aerodynamic Shape Optimisation using a Far-Field Analysis of the Drag Force

This chapter is based on the work presented at the AIAA Scitech 2019 Forum [52]. In particular, it is focused on the use of far-field analysis for drag prediction and decomposition in the context of aerodynamic shape design. Indeed, drag minimization is one of the fundamental objectives of Aerodynamics.

The straightforward computation of the aerodynamic force by stress integration on the body surface (*near-field* method) allows for the decomposition in friction and pressure force but does not give any information on the splitting among viscous, lift-induced and wave contributions which have a primary relevance for the aerodynamic designer. On the contrary, a *physical* decomposition is obtainable by *far-field* methods, based on formulae derived from the integral momentum equation. An additional feature of far-field techniques is the possibility to identify the local flow structures responsible for the generation of the aerodynamic force. Moreover, some of these methods showed the capability to improve the accuracy in the calculation of total drag from a given CFD solution by removing at least part of the so-called *spurious* drag implicitly or explicitly introduced by the artificial viscosity of the adopted numerical scheme. An extensive review of these far-field methods is reported in [53].

Nowadays, the methods developed for drag breakdown can be grouped into two families: thermodynamic-based and vorticity-based breakdown. The former is based on Oswatitsch's entropy drag concept and can decompose the *irreversible* drag in its viscous, wave and spurious components [6, 8]. Thermodynamic methods are now widely adopted in industrial and research environment [54, 55] and already contributed to the design of the last generation of transonic transport aircraft. They are limited to the computation and analysis of the force components of irreversible nature, therefore cannot compute and analyze lift. The lift-induced drag can only be derived indirectly by subtracting the irreversible drag to the near-field force.

More recent and less mature vorticity-based methods start from a general formulation of the vortex-force theory introduced by Wu et al. [56] and extract the reversible vortical-dominated process, therefore separating the lift-induced drag from the profile contribution [57]. These methods highlighted the role of the Lamb vector field (the cross product of vorticity times velocity) in the generation of the aerodynamic force. They have been extended to the analysis of compressible flows [58, 59]. The breakdown of the irreversible drag in viscous and wave component has also been proposed by Mele et al. [60], Ostieri and Tognaccini [61]. These methods can also be adopted for unsteady analyses and thrust-drag bookkeeping [62, 63]. Furthermore, these methods have also been applied to calculate the aerodynamic force of innovative aircraft design [64].

The far-field methods can help the aerodynamic shape optimization in two different ways:

1. The possibility to decompose drag in viscous, wave and lift-induced contributions allows for a selection of the objective function among these three terms. This particular characteristic has already been considered in [65].

2. The detection of at least part of the spurious drag contribution can allow for a reliable drag calculation even on very coarse grids thus implying a substantial reduction of the computational cost of the optimization process which is still a significant concern.

In this chapter, the second feature will be analyzed in detail.

The underlying assumption is that far-field approaches allow a good estimation of the aerodynamic characteristics of a given geometric configuration with a mesh size significantly smaller if compared to the one needed by near-field methods.

On the other hand, it should be noted that the flow field convergence level required for the correct estimation of the different sources of drag is higher for far-field methods with respect to near-field ones.

Nevertheless, even if some care is needed in the formulation of the objective function, the use of far-field methods allows a significant reduction of the computational effort required by the optimization loop. This advantage could be even more evident in robust design problems when the objective function and related constraints can be statistical quantities (like expected values, variances or generic risk measures [26]) which require a further sampling step for their estimation. The present work will demonstrate the feasibility of this approach to the modification of a symmetric airfoil for the reduction of wave drag in inviscid transonic condition at zero angle of attack. In particular, the first application will deal with the deterministic constrained problem described in [66] (AIAA Aerodynamic Design Optimization DG), and already solved using the far-field approach in [65]. It will be shown that the present far-field formulation, in conjunction with a suitable definition of the objective function, allows further improvement of drag with respect to the given reference obtained using a coarser computational mesh.

3.1 Far-Field Analysis of the Drag Force

The far-field method here described allows the decomposition of the drag force in three components: wave, viscous, and spurious drag. In particular, this method is based on the momentum integral balance applied to a control volume Ω surrounding the body.

The central idea of the method is the development of a far-field formula of the drag force based on entropy variations. In addition, the entropy drag is expressed as volume integral which will allow the decomposition of the drag in the components formerly mentioned. For that, the regions should be properly identified. A detailed description of the method is given in [6]. The entropy drag can be expressed as:

$$D_{\Delta s} = D_w + D_v + D_{sp} \tag{3.1}$$

where D_w, D_v, and D_{sp} are the wave, viscous, and spurious contributions, respectively.

D_{sp} is associated with the entropy introduced by the artificial dissipation and with the truncation error of the numerical scheme used in the Computational Fluid Dynamics solver. Therefore, a more accurate prediction of the drag coefficient for coarser grids is possible by computing this numerical error, D_{sp}, and subtracting it from Eq. (3.1). This is a considerable advantage for optimization purposes since the use of coarser grids allows significant computational time savings.

The three components of Eq. (3.1) are defined below:

$$\begin{cases} D_w = V_\infty \int_{\Omega_w} \nabla \cdot (\rho g \mathbf{V}) \, d\Omega \\ D_v = V_\infty \int_{\Omega_v} \nabla \cdot (\rho g \mathbf{V}) \, d\Omega \\ D_{sp} = V_\infty \int_{\Omega_{sp}} \nabla \cdot (\rho g \mathbf{V}) \, d\Omega \end{cases} \tag{3.2}$$

Being V_∞ the free stream velocity, ρ the density, $\mathbf{V}$ the velocity vector, and g a function of entropy variation Δs.

$$g\left(\Delta s/R\right) = -f_{s1}\left[\frac{\Delta s}{R}\right] - f_{s2}\left[\frac{\Delta s}{R}\right]^2 \tag{3.3}$$

where R is the gas constant and f_{s1} and f_{s2} are functions of the ratio between specific heats of the fluid, γ, and the free stream Mach number, M_∞ and are computed as follows.

$$f_{s1} = -\frac{1}{\gamma M_\infty^2} \qquad f_{s2} = -\frac{1 + (\gamma - 1) M_\infty^2}{2\gamma^2 M_\infty^4} \tag{3.4}$$

Furthermore, the volumes related with each drag region must be identified. This identification is based on a proper definition of shock-wave and boundary-layer plus wake sensors that are applied using the hierarchical criterion defined in [6]. The nondimensional function defined below is used to select the wave region Ω_w.

$$F_{shock} = \frac{\mathbf{V} \cdot \nabla p}{a |\nabla p|} > K_w \tag{3.5}$$

A cut-off value shall be decided for the particular problem under investigation. Usually, it takes values around 0.85.

Moreover, the viscous sensor (boundary-layer plus wake regions) must be defined:

$$F_v = \frac{\mu_l + \mu_t}{\mu_l} > K_v F_{v_\infty} \tag{3.6}$$

where μ_l is the dynamic viscosity, μ_t is the eddy or turbulent viscosity, F_{v_∞} is the viscous sensor value at the free-stream conditions, and K_v is the selected cut-off value. Classically, this sensor is used; however, it has been shown that it only works in fully turbulent regimes. Advanced definitions of the viscous sensor that also work in laminar and transition regimes are introduced and discussed in [67].

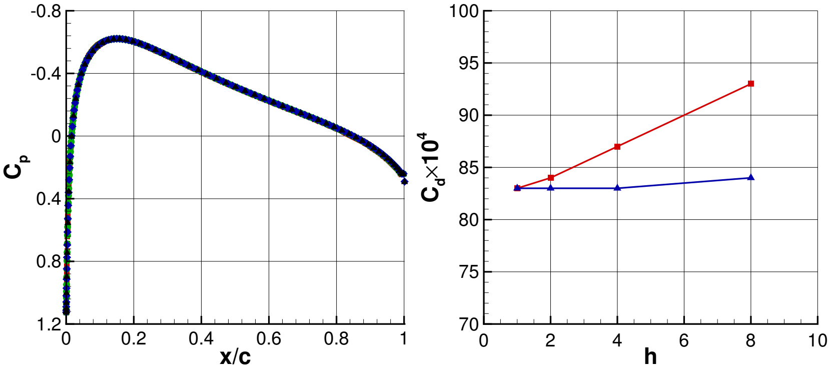
Finally, the volume that it is not selected, firstly, as wave or, in a second step, as viscous is going to be identified as spurious volume.

3.2 Advantages and disadvantages of using far-field methods

Some preliminary tests were performed in order to assess the proper implementation of the method and to show possible advantages of using the far-field analysis. The first test case was performed on the NACA 0012 airfoil, that will be used as baseline in the design optimization example, too.

This test case regards a viscous flow, and working conditions are $M_\infty = 0.7$ and $Re_\infty = 9 \times 10^6$ at $C_l = 0$. The mesh used is a single-block C-type grid with four grid levels: 224×64, 448×128, 896×256, and 1792×512 cells. For the test case, the selected cut-off values are $K_w = 0.9$ and $K_v = 2.7$. The near-field and far-field drag coefficients versus the mesh size are shown in Fig. 3.2.1, as well as the pressure coefficient distribution on the body surface at the different grid refinements. Particularly, the mesh size is given by the square root of the ratio between the number of cells of the finest grid and the grid under evaluation $\left(h = \sqrt{N_{h=1}/N_i}\right)$. Therefore, a decreasing h means an increase in grid size.

The pressure coefficient distributions on the airfoil obtained on all grids are proposed in Fig. 3.2.1a; differences are hardly visible, showing a good local accuracy even on the very coarse grid. In addition, Fig. 3.2.1b also shows that the near-field drag coefficient converges as the mesh size $h \longrightarrow 0$; specifically, there is a variation close to 11% in the computed near-field drag, which is clearly unacceptable if the coarse grids would be used in an optimization loop in which drag minimization is one of the objectives. On the contrary, the removal of the spurious drag by using the far-field approach allows to obtain, in



(a) Pressure Coefficient, c_p **(b)** Drag Coefficient, c_d

Figure 3.2.1: NACA 0012 test at $M_\infty = 0.7$, $Re_\infty = 9\times10^6$ and $\alpha = 0$. *Left*: Pressure coefficient distribution on the body surface at $h = 1$ (■), $h = 2$ (●), $h = 4$ (◆) and $h = 8$ (▲). *Right*: Near-field (–■–) and far-field (–▲–) drag coefficients versus mesh size.

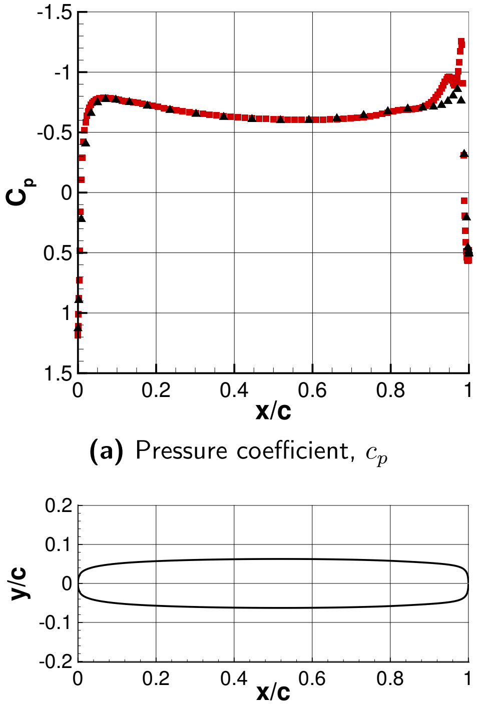
practice, a drag computation independent of the adopted grid. Taking into account that in a three-dimensional calculation, doubling the grid size implies a number of grid cells eight times lower, this example shows that a correct drag computation on grid level $h = 8$, in principle can lead to computational costs 512 times lower! Which is a clear advantage when facing optimization design problems.

Nevertheless, it must be pointed out that a fruitful adoption of the far-field technique in an optimization loop relies on two important requirements.

1. The local accuracy in the flow must be sufficient. In particular, if a coarse grid is adopted, the main features of the flow should be already correctly detected. Indeed, it is quite possible that some important characteristics of the flow, such as separated regions, vortices, shock waves require a sufficient grid resolution in order to be captured. A problem appeared in present study will be discussed below.

2. A correct detection of the spurious drag requires a proper selection of the boundary layer and shock wave regions. Significant progresses have been obtained towards a robust automatic region selection [67]. However, it is very likely that during an optimization loop with thousands drag calculations the far-field drag algorithm can fail as also discussed later. Clearly this possibility can lead to a wrong optimized solution.

Let us consider now an inviscid flow at $M_\infty = 0.85$ and $\alpha = 0$ around the airfoil depicted in Fig. 3.2.2b. In this case, the shock wave cut-off sensor, K_w, is set equal to 0.85.



(a) Pressure coefficient, c_p

(b) Airfoil

Figure 3.2.2: Inviscid test at $M_\infty = 0.85$ and $\alpha = 0$. *Upper*: Pressure coefficient distribution on the body surface at two grid levels: $h = 1$ (■) and $h = 4$ (▲).

The drag coefficient has been computed using the far-field formula at the coarsest $(h = 4)$ and finest $(h = 1)$ grid levels and the obtained values, c_{d_w} have been compared with the near-field results, $c_{d_{nf}}$. An underestimation of the computed drag coefficient using the coarsest mesh size is reported in Table 3.2.1. On the other hand, a good prediction of the drag coefficient is found at the finest grid size.

	Grid size	$c_{d_{nf}} \cdot 10^4$	$c_{d_w} \cdot 10^4$
$h = 4$	112×32	180	30
$h = 1$	448×128	63	61

Table 3.2.1: Mesh sizes and computed drag coefficients. Inviscid test at $M_\infty = 0.85$ and $\alpha = 0$

To understand why the far-field c_d was underestimated, the pressure coefficient distribution on the airfoil surface at both grid levels was plotted in Fig. 3.2.2a. It is seen how the coarsest grid is able to capture the expansion at leading edge of the airfoil. However, the shock wave at the trailing edge is not well simulated. Since a weaker shock wave was predicted, the far-field formula also gives a lower prediction of the drag coefficient. Hence, it is shown the importance of the detection of the main features of the flow by the grid to properly compute the drag coefficient with a far-field method.

An additional key point for the well functioning of the formula is the flow convergence. This issue is herein investigated. During an optimization test to solve the case study presented in Section 5.3 with a population based algorithm, a bad behavior of the drag coefficient calculation using the far-field formula was experienced. To explore the cause behind this behavior, two candidate airfoils of the optimization were selected. The airfoils are depicted in Fig. 3.2.3.

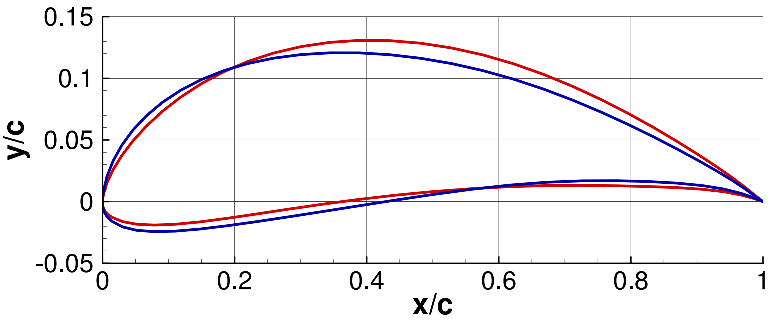


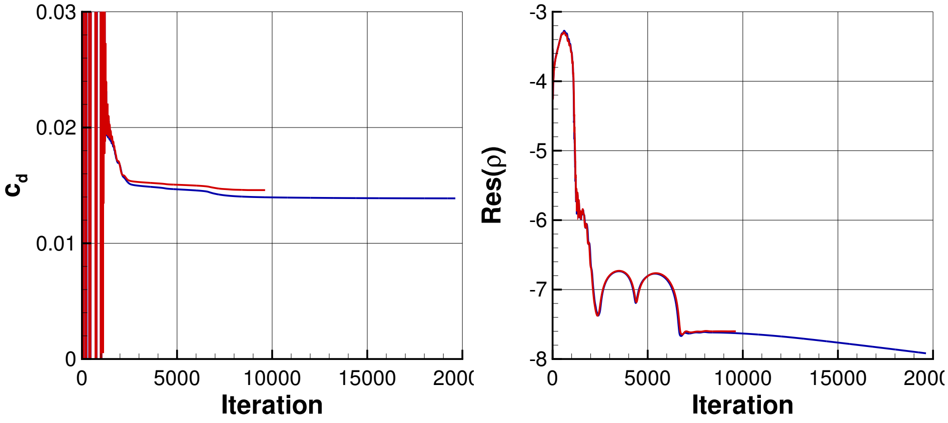
Figure 3.2.3: Airfoil comparison. Airfoil 1 (—) and Airfoil 2 (—). The axis are not dependent for visualization purposes.

The optimization test aims at the reduction of the c_d of a propeller blade airfoil under geometric and aerodynamic constraints. The starting airfoil is the Martin Hepperle MH 114. The operating conditions are $M_\infty = 0.2$, $Re = 4.97 \times 10^6$, and $\alpha = 2^\circ$. The lift coefficient of the airfoil is required to be greater or equal to one ($c_l \geq 1$). To calculate the aerodynamic performance, the flow solver used is SU2. A C-grid with 128 cells on the body surface and 64 cells on the wake and on the far-field direction is used (total number of cells equals to 16384). The convergence criterion used for the CFD analysis is that the cauchy criterion on c_d must be lower than 10^{-6}. With the output flow fields, the c_d is calculated with the far-field formula (only viscous drag c_{d_v} is found) and the result is compared with the near-field $c_{d_{nf}}$ value on a fine grid. The grid has 512 cells on the body surface and 256 on the far-field direction and on the wake (262144 total cells). The results are provided in Table 3.2.2.

	$c_{d_{nf}} \cdot 10^4$ (coarse)	$c_{d_v} \cdot 10^4$	$c_{d_{nf}} \cdot 10^4$ (fine)
Airfoil 1	145.9	88.6	124.4
Airfoil 2	138.8	117.8	120.6

Table 3.2.2: Far-field c_{d_v} and near-field drag coefficient comparison.

From Table 3.2.2, it is observed that an underestimation on the c_d is provided by the far-field formula using the solution of the Airfoil 1, whereas a good estimation is found when analyzing Airfoil 2. Both solutions were obtained after reaching the same convergence criterion. To understand the problem, the residual reduction value of the density was compared for both airfoils.



(a) Drag coefficient c_d convergence.

(b) Density (ρ) residual reduction.

Figure 3.2.4: Drag coefficient convergence and density residual reduction for Airfoil 1 (—) and Airfoil 2 (—).

In Fig. 3.2.4b, it is appreciated how the residual value for Airfoil 2 has further decreased than Airfoil 1, although the same convergence on c_d value (see Fig. 3.2.4a) is achieved. Therefore, it can be stated that it is not sufficient to make a control on the drag coefficient value. Particularly, it is observed that, for an accurate calculation of the drag coefficient with the far-field method, the flow field must reach a higher level of convergence. The solutions provided by computational fluid dynamic solvers converge firstly on the body surface and, only later, they converge far from the body. That is the reason why, although an accurate $c_{d_{nf}}$ was found for the Airfoil 1, the far-field approximation provides a poor result. Thus, this implies the need of a higher number of iterations.

3.3 Inviscid Airfoil Optimization Problem

The proposed design exercise is taken from [66], and it requires to find the symmetrical airfoil with minimum wave drag in a transonic inviscid flow, modeled by Euler equations, at $M_\infty = 0.85$ and zero angle of attack. The baseline geometry is the NACA 0012 airfoil, as mentioned in the previous section. The airfoil shape is represented parametrically using Non-Uniform Rational B-Splines (NURBS). This methodology allows the representation of complex geometries by means of control points and an associated weight to each control point (see Section 2.2.1). In this research, specifically, the airfoil upper surface is described using a quadratic NURBS curve with six control points. The lower surface is obtained by specular symmetry. The baseline shape is the NACA 0012 airfoil. According to the chosen parametrization, it is defined by the control points illustrated in Fig. 3.3.1.

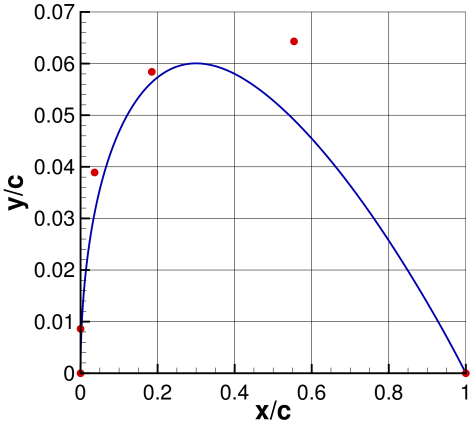


Figure 3.3.1: NURBS six control points for NACA 0012 airfoil representation. NACA 0012 upper surface (—) and NURBS control points (●). The y-coordinate is magnified and proportions are not respected.

In addition, the complete set of control points and weights needed for NURBS representation of NACA 0012 airfoil is given in Table 3.3.1.

The first and the last control points are the Trailing Edge (TE) and Leading Edge (LE) respectively that are kept fixed during the optimization. The remaining control points determine the shape variations during the optimization. In particular, x, y, and w parameters are changed in control points two to four, while the fifth control point is constrained to have the x-coordinate equal to zero in order to keep vertical the tangent at the leading edge.

	$x \cdot w$	$y \cdot w$	w
1	1	0	1
2	0.5948	0.0690	1.0735
3	0.3934	0.1243	2.1276
4	0.0494	0.0527	1.3551
5	0	0.0123	1.4271
6	0	0	1
Nodal vector			
[0 0 0 0.4066 0.6748 0.9 1 1 1]			

Table 3.3.1: NURBS representation of NACA 0012 airfoil

This leads to a total of 11 design variables for the optimization. The active design variables are those inside the box in Table 3.3.1.

A geometrical constraint is imposed requiring that the optimized airfoil completely includes the baseline. This is obtained imposing that, at each x-coordinate, the absolute value of the abscissa has to be greater than the abscissa of the zero thickness trailing edge NACA 0012 airfoil defined by Eq. (3.7).

$$y_{\text{baseline}}\left(x\right) = \pm 0.6\left(0.2969\sqrt{x} - 0.1260x - 0.3516x^{2} + 0.2843x^{3} - 0.1036x^{4}\right) \tag{3.7}$$

Actually, coherently with [65], the constraint was relaxed by allowing the profile to have, at a given x-coordinate, a slightly smaller absolute value of the y-coordinate than the NACA 0012 airfoil: $c(x) = \left|y\left(x\right)\right| - \left|y_{\text{baseline}}\left(x\right)\right| > -5 \cdot 10^{-4}$. The constraint is expressed as a penalty, and consequently the optimization problem can be mathematically expressed as

$$\min_{P}\ c_d + p\left(y\left(x\right)\right) \tag{3.8}$$

where P is the set of NURBS control points and weights, and the penalty function $p\left(y\left(x\right)\right)$

is given by

$$p\left(y\left(x\right)\right)=\begin{cases}0 & \text{if}\quad \left|y\left(x\right)\right|-\left|y_{\text{baseline}}\left(x\right)\right|>-5\cdot 10^{-4}\quad \forall x\in\left[0,1\right]\\ 0.5 & \text{otherwise.}\end{cases}\tag{3.9}$$

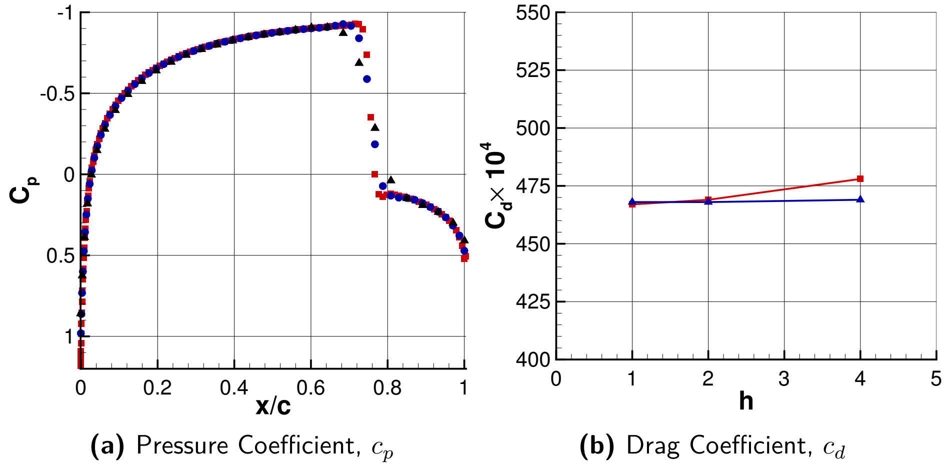
A further check requires the full convergence of the CFD solutions.

The optimization algorithm chosen for this design exercise is the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). A full description of the algorithm is provided in Section 2.1.2.

3.3.1 Preliminary analysis and mesh converging study

The CFD analyses were performed using the Zonal Euler Navier-Stokes flow solver (see Section 2.2.3) for an inviscid flow at $M_\infty = 0.85$ and $c_l = 0$. The mesh used is a structured C-grid with the far-field located at 100 chords. In the present test case, the shock wave cut-off value adopted in the far-field drag calculation is $K_w = 0.85$.

As in the previous viscous test, a grid convergence study was performed using three grids of increasing size.



(a) Pressure Coefficient, c_p **(b)** Drag Coefficient, c_d

Figure 3.3.2: NACA 0012, inviscid test at $M_\infty = 0.85$ and $\alpha = 0^\circ$. *Left*: Pressure coefficient distribution on the body surface at $h = 1$ (■), $h = 2$ (●), and $h = 4$ (▲). *Right*: Near-field (–■–) and far-field (–▲–) drag coefficients versus mesh size.

The comparison of the obtained pressure coefficient distribution on this baseline geometry is proposed in Fig. 3.3.2a. Again, differences are hardly visible in the smooth part of the flow. Some discrepancies are evidenced at the shock, but they are unavoidable because the *numerical* shock thickness depends on the grid resolution. Anyway, the shock appears sufficiently resolved on the coarse grid too. The comparison between the near-field and far-field coefficient drag calculations are reported in Fig. 3.3.2b and in Table 3.3.2. Again, the near-field drag converges as the mesh is refined and, as expected, the far-field formula with the spurious drag removal correctly computes drag even on the coarse grid. This result suggests that the grid resolution $h = 4$ can be successfully adopted in the optimization loop, but, as discussed in the following, the success is not straightforward.

The number of cells on the airfoil surface (N_b), along the far-field direction (N_j), and in the wake (N_w), together with the drag coefficient value obtained with near-field $(c_{d_{nf}})$ and the far-field (c_{d_w}) methods are reported in Table 3.3.2.

	N_b	N_j	N_w	$c_{d_{nf}} \cdot 10^4$	$c_{d_w} \cdot 10^4$
$h = 4$	64	32	24	478	469
$h = 2$	128	64	48	469	468
$h = 1$	256	128	96	467	468

Table 3.3.2: Mesh sizes and computed drag coefficients

3.3.2 Results

Two different sets of optimization runs have been performed. The first one used the near-field formulation to evaluate the drag coefficient, while the second one relied upon the far-field method.

Optimization Using Near-Field Drag Evaluation Method

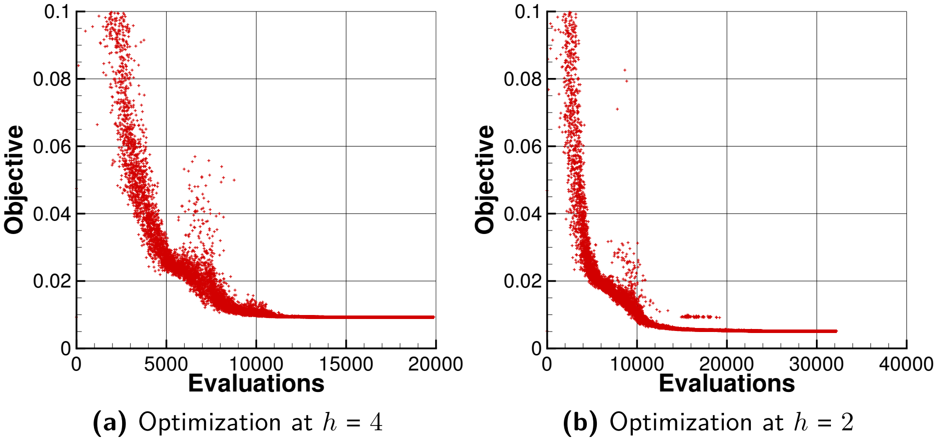
Two optimization problems were solved in which the drag force was evaluated by the near-field method. These were performed using the coarsest grid size, $h = 4$, and the subsequent mesh refinement $h = 2$. Several optimization processes were launched to assess the robustness of the set-up of the optimization algorithm. Only the best performance results are here reported. The optimization parameters used for the first phase are reported in Table 3.3.3.

The convergence histories of the two optimization problems performed are shown in Fig. 3.3.3. Note that both optimizations were stopped before reaching the maximum num-

Maximum evaluations	Population size, λ	Initial standard deviation, σ
36000	72	0.05

Table 3.3.3: Optimization Parameters for the optimization at $h = 4$ and $h = 2$

ber of evaluations since a further improvement of the objective value was not expected. In particular, the optimization at the coarsest grid level, $h = 4$, was stopped after 19872 evaluations, and the optimization at $h = 2$ after 32184 iterations.



(a) Optimization at $h = 4$

(b) Optimization at $h = 2$

Figure 3.3.3: Convergence histories of the near-field optimizations

The NURBS representation of both optimal airfoils is given in Table 3.3.4.

In addition, Fig. 3.3.4 shows the optimized airfoils at $h = 4$, at $h = 2$, as well as the starting NACA 0012 airfoil. The resulting optimized airfoils have a nearly constant thickness along the chord, the leading edge has been flattened, and the trailing edge is rounded. Moreover, it can be appreciated, also from Fig. 3.3.4, that the main effect of mesh refinement on the optimal airfoil solutions is located around the LE of the airfoil. The leading edge is flatter in the optimal airfoil found by the optimization at a finer grid.

In addition, after each optimization, the performance of the obtained airfoils has been evaluated by computing the drag coefficient at the finest mesh level $h = 1$. In Table 3.3.5, the objective value of the optimization (c_d calculated using the corresponding grid level of the optimization) and the drag coefficient computed at $h = 1$ are given. Furthermore, in

	Optimization at $h = 4$			Optimization at $h = 2$		
	$x \cdot w$	$y \cdot w$	w	$x \cdot w$	$y \cdot w$	w
1	1	0	1	1	0	1
2	4.0552	0.2014	4.0527	2.3032	0.1097	2.3051
3	0.7651	0.1889	2.7919	0.4020	0.0695	0.9702
4	0.0792	0.1747	4.1819	0.0009	0.0657	1.4720
5	0	0.0329	4.4046	0	0.0131	1.1963
6	0	0	1	0	0	1
	Nodal vector					
	[0 0 0 0.4066 0.6748 0.9 1 1 1]					

Table 3.3.4: NURBS representation of the optimized airfoils after the three optimization steps

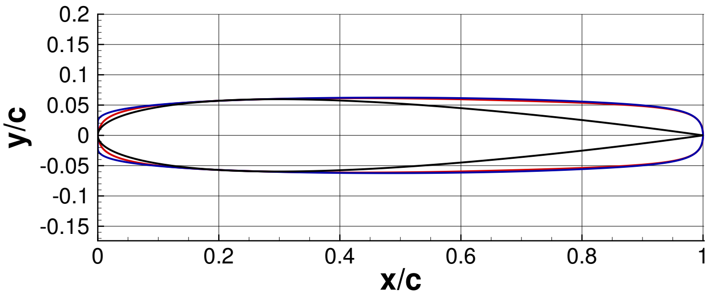


Figure 3.3.4: Comparison of NACA 0012 airfoil (—) with the optimized airfoils obtained at $h = 4$ (—) and $h = 2$ (—).

Table 3.3.5 the associated CPU cost of a single CFD evaluation at the specific grid level is reported. Recall that the costs are calculated based on the CPU cost of the CFD evaluation at the coarsest grid ($h = 4$). Besides, the associated CPU cost of a CFD evaluation at the finest grid level, $h = 1$, is 19.5.

	Objective value	$c_d \cdot 10^4$ at $h = 1$	CPU cost
Optimization at $h = 4$	95	69	1
Optimization at $h = 2$	53	33	4.5

Table 3.3.5: Objective value and CFD Drag coefficient using $h = 1$ grid size

From Table 3.3.5, it can be observed that after the optimization using the coarsest grid level a 85.2% drag reduction is achieved (taking as reference value 467 $d.c$, see Table 3.3.2). Performing an optimization with the second mesh size allows a further decrease of the drag coefficient (another 36 $d.c$), thus, reaching a 92.9% diminution from the starting NACA 0012 airfoil.

Figure 3.3.5 compares the pressure coefficient distribution on the body surface, c_p, of the NACA 0012 airfoil (in black) with respect to the optimized airfoil using $h = 4$ grid (in red), and using $h = 2$ grid (in blue) during the optimization. Note that the c_p distributions are calculated at the finest grid size $(h = 1)$ for all airfoils. The NACA 0012 presents a strong shock wave approximately at $3/4$ of its chord length. In the optimized airfoil, the shock wave has moved closer to the TE of the airfoil. Moreover, an expansion is present at the leading edge of the airfoil. A stronger flow expansion is found in the optimized airfoil using $h = 2$ that favors drag reduction. It is interesting to note that significant differences appear using different grid resolutions during the optimization not only at the shock, as was expected, but also in the suction peak in the leading edge region.

A main result of this analysis is that the optimization loop performed with the grid $h = 4$ is clearly not acceptable since the one performed with $h = 2$ gives a far better result.

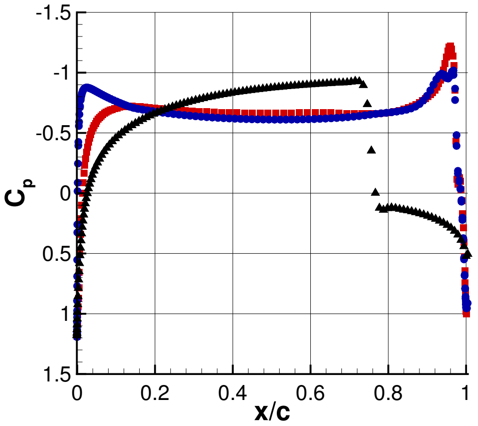


Figure 3.3.5: Comparison of the pressure coefficient distribution on the body surface of NACA 0012 airfoil (▲) with respect to the optimal airfoils obtained doing an optimization at $h = 4$ (■), and an optimization at $h = 2$ (●). Inviscid test at $M_\infty = 0.85$ and $\alpha = 0$.

In addition, Fig. 3.3.6 was provided to show the effect of grid refinement on c_p for the same airfoil. In particular, the pressure coefficient distribution on the airfoil surface is plotted at the three mesh resolutions for the obtained optimized airfoil by using $h = 2$ during the optimization process. The main difference is at the shock wave located in the trailing edge. In fact, the coarsest mesh $(h = 4)$ is not able to capture properly this flow feature. This will make the adoption of the far-field formula for the optimization at this grid level not straightforward.

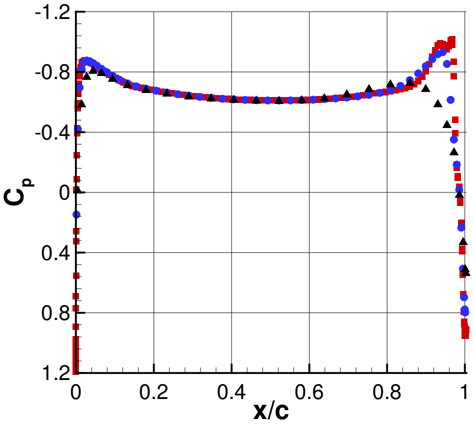


Figure 3.3.6: Comparison of the pressure coefficient distribution on the body surface of the optimized airfoil using $h = 2$ during the optimization at different grid levels: $h = 4$ (▲), $h = 2$ (●), and $h = 1$ (■). Inviscid test at $M_\infty = 0.85$ and $\alpha = 0$.

Optimization Using Far-Field Drag Evaluation Method

In this section, the far-field method introduced in Section 3.1 was used to estimate the wave drag coefficient to be minimized. In this particular case, the selected shock wave sensor is $k_w = 0.85$. The optimization problems were done at grid level $h = 4$. The objective is to check if the far-field method can provide more acceptable results at level $h = 4$ than the near-field approach. Preliminary experiments evidenced that potentially the far-field method was able to filter out the contribution of spurious drag effectively. Nevertheless, the first optimization runs evidenced that when unfeasible airfoils, characterized by numerical separations or instabilities, were generated by the design procedure, the far-field method gave unreliable results. Therefore, to face this problem, a new constraint, based on the near-field drag calculation, less sensitive to this problem, was introduced. This constraint is implemented as a step penalty function that is triggered when the difference between far-field and near-field drag coefficients is greater than a problem-dependent threshold. In addition, a check on the grid quality has also been introduced for avoiding anomalous behavior of the fluid dynamic solver. Thus, the optimization problem is redefined as

$$\min_{P} \; c_{d_w} + p\left(y\left(x\right)\right) + p\left(c_d \,\text{comparison}\right) \tag{3.10}$$

with

$$p\left(c_d \,\text{comparison}\right) = \begin{cases} 0 & \text{if} \quad c_{d_{nf}} - c_{d_w} < 0.0150 \\ 0.5 & \text{otherwise.} \end{cases} \tag{3.11}$$

The new optimization problem was run at the first grid level and using the algorithm parameters given in Table 3.3.3. The robustness of the set-up was checked launching the optimization process several times. In particular, five optimal solutions are here reported. The objective function formulation was always able to distinguish between feasible and unfeasible solutions, and the current optimization algorithm set-up was able to find a feasible optimal solution. The convergence history of the optimization that found a better performance airfoil is illustrated in Fig. 3.3.7.

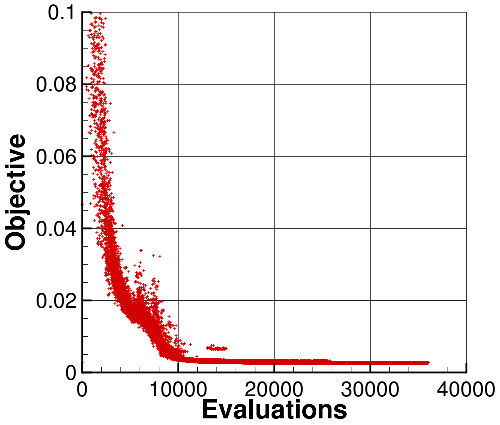


Figure 3.3.7: Convergence history of the far-field optimization

Analogously to the near-field study, the drag coefficient of the optimized airfoil was recomputed at the finest grid level, $h = 1$. This value together with the objective value (wave drag coefficient calculated with the far-field method at $h = 4$) are given in Table 3.3.6.

Table 3.3.6 shows that a higher drag reduction is always achieved than in the optimization made employing the near-field formulation at the same grid level, $h = 4$, which is

	Objective value	$c_d \cdot 10^4$ at $h = 1$
Trial $\#1$	29	39
Trial $\#2$	36	46
Trial $\#3$	28	39
Trial $\#4$	29	37
Trial $\#5$	30	39

Table 3.3.6: Objective value and CFD Drag coefficient using $h = 1$ grid size

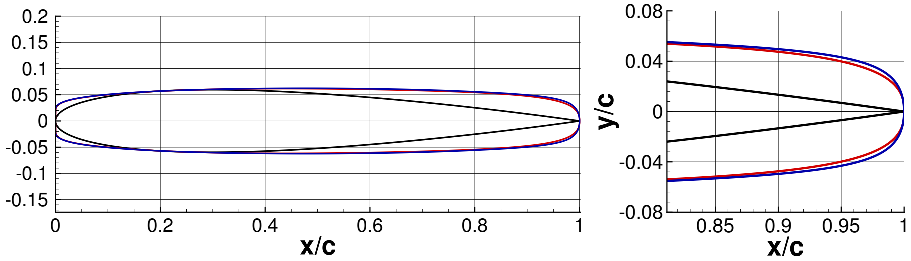
85.2%. In the worst case (trial $\#2$), a 90.1% reduction was reached, whereas, in trial $\#4$, the best-case scenario, the drag coefficient is reduced by 91.6%. This shows the advantage of using the far-field method in the optimization process since only the wave drag is minimized, contrary to the optimization cases where the near-field results of the drag coefficient are used where wave and spurious drag are optimized at the same time. In particular, by using the far-field formula, an optimum comparable to the solution of the optimization using the near-field approach and a denser grid ($h = 2$) is found. The potentiality of using the far-field formula are highly assessed, considering the good results obtained on a coarse grid and the consequent CPU time savings, with reference to a classical near-field approach.

The NURBS representation of the optimum airfoil of the best trial (Trial $\#4$) is provided in Table 3.3.7.

	$x \cdot w$	$y \cdot w$	w
1	1	0	1
2	2.4904	0.1292	2.4835
3	0.5910	0.1348	2.0143
4	0.0023	0.1118	2.6785
5	0	0.0012	1.2024
6	0	0	1
Nodal vector			
[0 0 0 0.4066 0.6748 0.9 1 1 1]			

Table 3.3.7: NURBS representation of the optimized airfoil of Trial $\#4$

The optimized airfoil (in red) of the best trial (Trial $\#4$) is compared in Fig. 3.3.8 with the baseline airfoil NACA 0012 (in black) and the optimized airfoil obtained using the near-field approach for drag evaluation (in blue). The optimized airfoil shows the same shape characteristics than those obtained using the near-field approach. The airfoil has approximately a constant thickness along $3/5$ of the chord. The trailing edge has been rounded, and the leading edge is almost flat. Furthermore, it can be observed that the optimal airfoil found with the near-field approach (in blue) has a rounder TE (see Fig. 3.3.8b) than the TE of the optimized airfoil using the far-field method (in red).



(a) Airfoil

(b) Trailing edge

Figure 3.3.8: Comparison of the optimized airfoil with the far-field approach (—) with the baseline NACA 0012 airfoil (—) and the optimal airfoil with the near-field approach using $h = 2$ (—).

In addition, from the observation of Table 3.3.6, it appears that the value of the drag coefficient recomputed at the finest grid level may be slightly different from the objective function, while, in principle, a closer value should be expected. This possible mismatch was already introduced in Section 3.1.

The pressure distribution on the body surface of the best trial at the coarsest $(h = 4)$ and finest $(h = 1)$ mesh sizes are plotted in Fig. 3.3.9. Contrary to the near-field optimization, the suction peak of the optimum solution is now correctly detected; however, the coarse grid resolution is not sufficient to correctly resolve the shock: the optimization problem here proposed is particularly critical because the main feature of the flow, the shock wave, cannot be accurately reproduced, one of the main requirements for a proper far-field drag analysis. The coarsest grid level is not able to properly capture the shock wave located closer to the TE of the profile and the weakening of the shock unavoidably leads to an underestimation of the wave drag coefficient, c_{d_w}.

To better understand the usefulness of the far-field method in aerodynamic shape design, it is necessary to focus not on the greater precision obtainable in the estimation of drag force, with a given computational mesh, but on the reduction of spurious drag con-

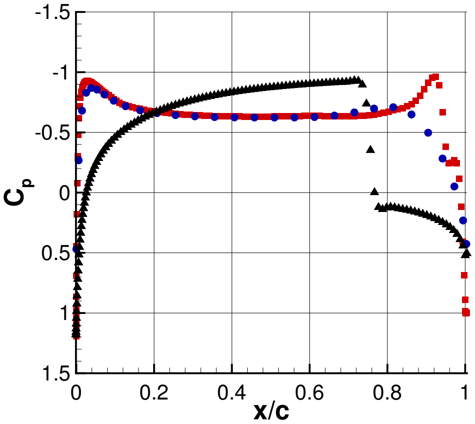


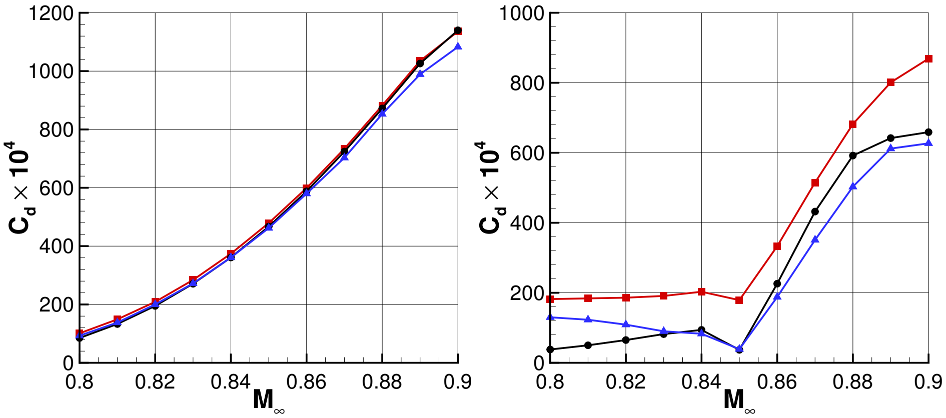
Figure 3.3.9: Pressure coefficient distribution on the body surface of the optimal airfoil (Trial #4) at two grid levels: h = 1 (■) and h = 4 (●), and the pressure coefficient distribution of the baseline NACA 0012 airfoil (▲). Inviscid test at M_∞ = 0.85 and α = 0.

tribution provided by this method. Indeed, for the convergence of an optimization process, what counts is the ability to discriminate between two different candidate optimal solutions, which of the two is better. If the optimization process aims to reduce the drag, the major obstacle to the use of a coarse grid is the spurious drag term. Indeed, this term adds a substantially random and non-controllable contribution to the resistance value that can deceive the optimizer, leading it to believe that an airfoil is better than another one incorrectly. In other terms, this means that the result of the comparison is not confirmed when the drag computation is performed using a denser mesh. Naturally, the use of a very coarse mesh inevitably introduces an error in the aerodynamic characteristics and, likewise, the optimizer will proceed up to a level of solution improvement such that it will no longer be possible to discriminate between two candidate configurations which one is better. However, the far-field method, as it has been demonstrated in this work, thanks to the elimination of the spurious contribution, allows a significant extension of the predictive and comparative capacity of a fluid dynamic solver for a given density level of the computational mesh. So, as has shown in the examples presented, the use of the far-field approach, even if not perfect, allows the optimization algorithm to reach to a significantly better solution refinement than that obtainable with the near-field method. Obviously, if grid density and quality increase, the use of the far-field approach in optimization becomes less useful, at the denser mesh levels, near-field and far-field techniques provide substantially the same difference in estimating the resistance of two aerodynamic configurations.

3.3.3 Toward an affordable Robust Optimization

Robust design optimization aims to obtain optimal solutions less vulnerable to different sources of uncertainties, such as operating conditions, here considered, or the shape of the airfoil. In particular, the aircraft may operate at speeds and load conditions that differ from the design ones, so that, they must be treated as uncertain parameters. These uncertainties are considered by introducing random variables. Hence, the objective function and the constraints of the problems will be functionals, and a method to map them into the set of real numbers has to be found. Lately, risk measures have been introduced for aerodynamic robust design. For an accurate calculation of these risk measures, a minimum of 20 additional evaluations of our quantity of interest (QoI) is necessary, implying a significant increase in the computational time. Here, the QoI is the airfoil drag coefficient in transonic inviscid flow conditions. Therefore, if the far-field method for the drag force prediction, used in the deterministic optimization, is suitable for the c_d calculation at different operating conditions (a range of Mach numbers), the computational cost of the robust optimization will be drastically reduced, making it affordable. Here, a preliminary study in the range of Mach numbers $0.8 - 0.9$ for the baseline airfoil and the optimum solution shows how the far-field method behaves.

Figure 3.3.10 shows the sensitivity of the computed drag coefficient to the freestream Mach number, for both the baseline and optimized airfoils. Drag monotonically increases with Mach number in the case of the baseline body, clearly due to the increasing of shock strength with Mach number. On the contrary, for the optimized airfoil, there is a sharp minimum at $M_\infty = 0.85$, the design Mach number for the optimization. It is interesting to note that the differences between the near-field and far-field drag calculations on the coarse grid (the one used for the optimization) are much more evident for the optimized airfoil, evidencing a much larger pollution introduced by the spurious drag in the optimum region. The far-field drag analysis on the coarse grid is in much better agreement when compared with the reference near-field value on the fine grid, in particular the drag minimum at the design Mach number (0.85) is more clear. For $M_\infty < 0.85$ the near-field drag on the coarse grid is substantially constant; therefore the optimum region is not well detected. This absence of sensitivity to flow conditions (included geometry) is, very likely, the reason why the optimization on the coarse grid by the near-field formula leads to an unsatisfactory result.



(a) Baseline airfoil (NACA 0012)

(b) Optimized airfoil

Figure 3.3.10: Sensitivity of the computed drag to the freestream Mach number. Near-field at $h = 4$ (–■–), near-field at $h = 1$ (–●–), and far-field at $h = 4$ (–▲–) drag coefficients versus mesh size.

It can be noted that, for the higher Mach number, the far-field formula underestimates drag on both the baseline and optimized airfoils. This is due to the reduction of shock strength introduced by the unavoidable reduction of grid resolution. In the present work, the problem is not serious, since the far-field formula correctly predicts the trends, but it is a further warning on the adoption of far-field methods: they are able to significantly improve drag calculations provided that the local features of the flow are correctly captured.

3.4 Conclusions

In this chapter, a far-field method (based on entropy variations) for drag force evaluation has been implemented within an optimization method. It allows the exclusion of the spurious drag contribution unavoidably introduced by the numerical scheme. In this way a better prediction of the c_d is achievable for coarser grids; hence CFD evaluations costs can be significantly saved. These possible advantages have been verified by trying an airfoil optimization in transonic inviscid flow, which used a very coarse grid resolution. The studied case was the optimization of the NACA 0012 airfoil in an inviscid flow at $M_\infty = 0.85$ subject to a geometrical constraint on the thickness. The adoption of the far-field formula allowed for an optimization result very near the best solution obtainable by a much denser grid. On the contrary, the optimization performed with the same resolution, but with the objective function computed by the standard near-field formula did not provide a satisfactory result. Indeed, the very large spurious drag present on the coarser grid can hide the optimum region.

While adding this method inside the optimization loop, several problems were evidenced and solved. The most critical issue faced was the applicability of the far-field method only in physically feasible solutions. It was addressed by setting up a control, in the form of a penalty in the objective function, on the near-field drag computation, which is more sensitive to the physicality of the solution. In addition, a check on the grid quality has been introduced to avoid anomalous behavior of the fluid dynamic solver.

From the presented results, it can be concluded that the use of a far-field method offers several advantages for aerodynamic design optimization since it also allows the optimization of only the physical drag components, here the wave drag, and avoids the optimization of the spurious drag inherent to the CFD calculations.

As opposed to the adoption of the standard near-field drag formula, the far-field formula allowed for a correct sensitivity analysis to Mach number variations, opening a road map towards a practically feasible robust design analysis, in the cases in which drag is one of the objective functions.

However, it should be noted that the possible grid coarsening is limited by the requirement that the local flow features, such as shock intensity in the present test, need to be sufficiently well resolved.

4 Adjoint based Robust Aerodynamic Shape Optimization

This chapter is based on the work presented at the journal article "*Gradient based empirical cumulative distribution function approximation for robust aerodynamic design*" [68].

4.1 Introduction

The purpose of this chapter is to illustrate an approach to robust optimization in aerodynamic design problems of industrial interest. The method relies on the Conditional Value at Risk (CVaR) function (see Sections 1.2 and 1.3), and it is applied to the robust aerodynamic design of the central airfoil of a Blended-Wing-Body (BWB) aircraft at transonic cruise conditions. The choice of the BWB is in line with the scientific community's research trend towards reducing of the environmental impact of air traffic. For this reason, in Europe, institutions like the Advisory Council for Aeronautics Research in Europe (ACARE) identified the Blended-Wing-Body (BWB) aircraft as one of the most promising layouts for the reduction of gas emissions and noise [69, 70]. Hence, the realization of an aerodynamic design optimization problem of this type of aircraft is of particular interest. A BWB aircraft has no clear dividing line between a central lifting-body that substitutes the classical fuselage and the wing. Furthermore, it often lacks the tail, so the pitch stability cannot be handled by elevators. Thus, regardless of the particular BWB or flying wing considered, it is mandatory to optimize the aerodynamic performance of the central airfoil, taking into account a constraint on the provided pitching moment coefficient c_m. Indeed, as remarked in [71], the static pitching stability of the whole BWB aircraft is usually accomplished by balancing a negative nose-down moment, related to the outer-wing, with a positive nose-up moment given by the central section.

In this applied research context, robust and reliability-based design optimization is gaining popularity since this approach helps to obtain solutions that do not deteriorate excessively in the presence of uncertainties on the airfoil contour or operating conditions. The uncertainties on the wing surface are, for example, due to machining tolerances, to shape deformation under load, or debris accumulation. Besides, the aircraft may operate at speed and load conditions quite different from the design ones, so they can be also uncertain parameters. Random variables are introduced to account for these unknown states. Therefore, the objective of the optimization design problem, as well as the constraints, are functionals. As a result, it is necessary to map them into the set of real numbers $\mathbb{R}$. As introduced in Chapter 1, several approaches are historically used like the worst-case [72] or the expected values with safety margins [73]. However, in this study, the Conditional Value-at-Risk (CVaR) estimator is considered to solve the presented robust design optimization (RDO) problem and it is estimated using an approach based on the empirical cumulative distribution function (ECDF). For further details on how to calculate the CVaR using the ECDF refer to Sections 1.3.1 and 4.3. The robust approach based on risk functions is, without doubt, more powerful and flexible than the deterministic multi-point methodology that is generally used in this type of problem, given that the high number of uncertain operational parameters makes the definition of the multi-point design problem quite tricky and cumbersome.

In the RDO problem here studied, the quantities of interest (QoI) of the risk function are the aerodynamic characteristics of the airfoil, such as lift, drag, and pitching moment coefficients, which are computed solving the compressible Reynolds-averaged Navier-Stokes equations with the open-source fluid-dynamic solver SU2 [50, 51] (the flow solver is explained in-depth in Section 2.2.3). Using a RANS solver makes the calculation of CVaR costly from the computational point of view. Therefore, computational cost reduction is the characterizing aspect of this dissertation. Above all the different possibilities, in this chapter, the estimate of the ECDF is here obtained by exploiting the QoI gradient with respect to the uncertain variables computed through the discrete adjoint module of SU2 [74]. Thus, the ECDF is approximated with a first-order series expansion using efficiently calculated gradients from SU2, which, thanks to the adjoint method, are available at almost the computational cost of one CFD solution whatever the number of uncertainties is. Thus, the adjoint-based approach presented here allows a substantial reduction of the computational cost needed to estimate the ECDF.

In synthesis, this chapter introduces a methodology for approximating the ECDF based on the efficient computation of gradients and illustrates its potentialities by applying it to an aerodynamic design problem of industrial interest. Specifically, a tailless BWB aircraft central section is considered for robust design optimization in transonic cruise conditions. The first step is the study of the aerodynamic characteristics of the baseline airfoil. Subsequently, an adjoint-based method aimed to reduce the computational cost of estimating the Cumulative Distribution Function is introduced in Section 4.4.1. After that, two consecutive optimizations are done, a deterministic one first (Section 4.5.2) and a robust one later (Section 4.5.3). In addition, before performing the robust optimization, a sensitivity analysis of the optimum deterministic airfoil to different uncertainty sources is done (Section 4.5.4). Finally, the baseline, the deterministic optimum, and the robust optimum airfoils are analyzed and compared from a robustness point of view.

4.2 Physical problem definition

As already introduced, the robust aerodynamic design problem of the central airfoil of a Blended-Wing-Body (BWB) aircraft is here faced. This design exercise working conditions are the typical cruise flight conditions for long-range commercial aircraft, which correspond here to a free-stream Mach number $M_\infty = 0.8$ at an altitude $h = 12000$ m. The design problem is tailored on an airfoil, which is conceived to work at fixed incidence $\alpha = 0^\circ$, providing a minimum amount of lift and a specific nose-up moment. The airfoil in question represents the central section of a BWB with a chord of approximately 30 meters. Then, the Reynolds number at cruise condition is $Re_c = 174 \times 10^6$.

In a real application, it is not possible to work constantly at the precise operating point, and some deviations from the operating parameters are likely. Moreover, it is known that some parameters would affect the airfoil performance more than others. Therefore, uncertainties in Mach number and Angle of Attack are introduced for taking into account the deviations before mentioned. However, perturbations in atmospheric pressure and temperature are assumed not to affect the airfoil drag consistently. The modeling of the working conditions uncertainties is given in Section 4.5.3.

4.2.1 Baseline airfoil shape and characteristics

In the following paragraphs, the central BWB section is presented, together with its aerodynamic characteristics. Also, a grid convergence study is provided.

Figure 4.2.1 shows the baseline airfoil shape characterized by some geometrical quantities that are crucial for the design process. One of the most important is the leading-edge radius (LER). Indeed, a small value of LER might induce an early stall of the wing. Hence, it is preferable to constraint it to safe values. Other geometrical quantities are the thickness-to-chord ratio t/c, the trailing edge angle (TEA), and the thickness ratio at the trailing edge (TAT). The latter is computed at 85% of the airfoil chord.

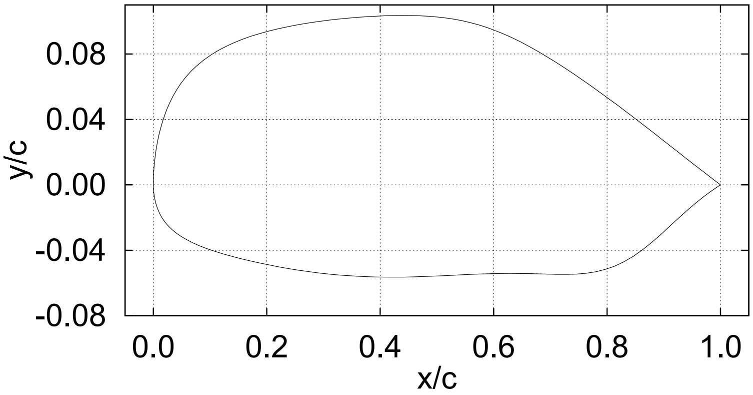


Figure 4.2.1: Baseline airfoil. BWB central section. The y-coordinate is magnified and proportions are not respected.

All these properties are specified for the baseline airfoil in Table 4.2.1. By controlling t/c, TEA and TAT, it is possible to maintain the inner space necessary to load the aircraft.

Figure 4.2.2 reports the baseline airfoil polar curves at Mach numbers representative of the airfoil operating conditions, that is, at the nominal condition and at the uncertainty region the boundaries ($M_\infty = 0.78, 0.80,$ and 0.82). The compressible RANS equations

t/c	LER	TEA	TAT
0.16	0.0156	27.57°	0.083

Table 4.2.1: Baseline airfoil geometrical parameters.

are solved using SU2 flow solver with the Spalart-Allmaras (SA) turbulence model [75]. For the spatial integration, JST central scheme with artificial dissipation coupled with an implicit Euler method for the pseudo-time stepping is used.

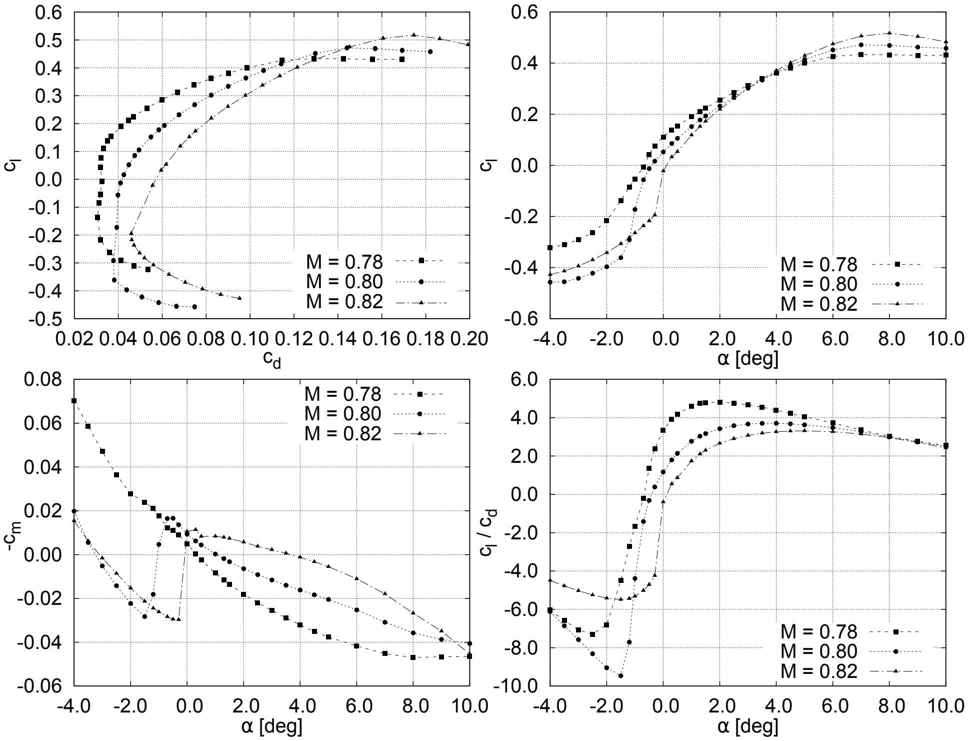
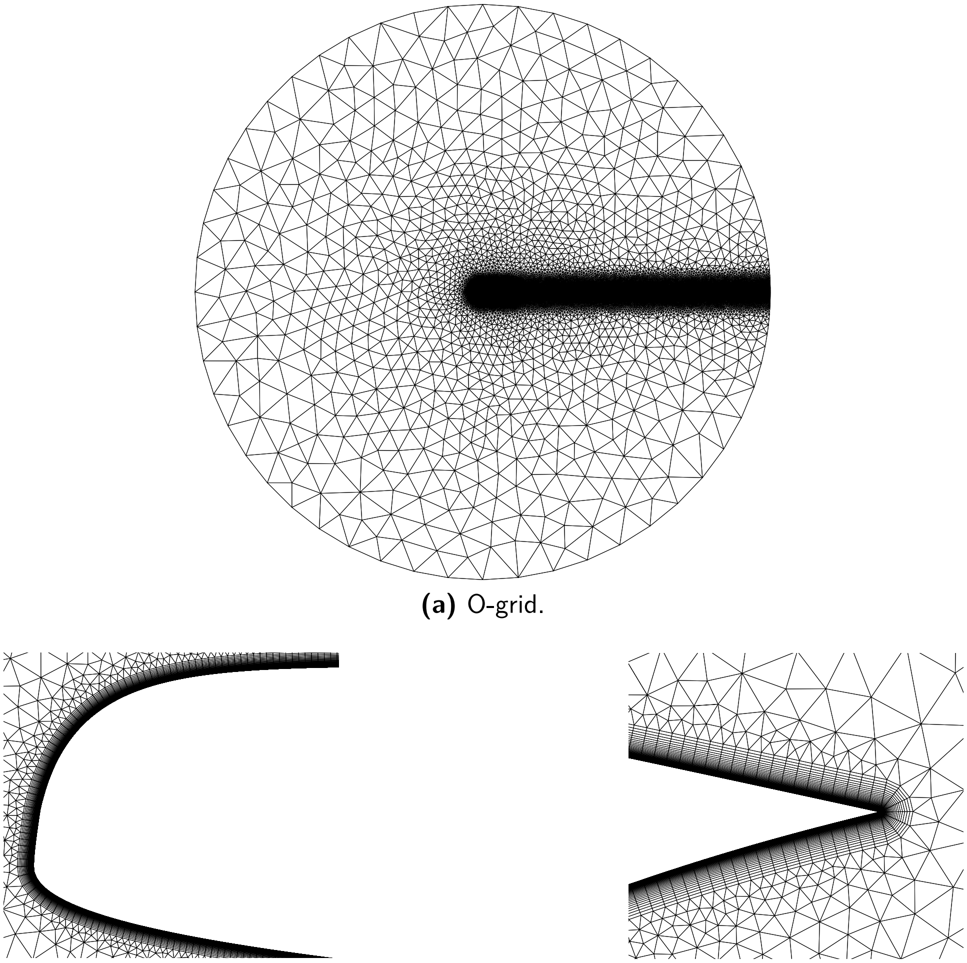


Figure 4.2.2: Baseline airfoil performance curves at the nominal Mach and at the boundaries of the Mach uncertainty region.

Concerning the grid generation, an automatic self-developed procedure based on the open source-grid generator *Gmsh* is used (more details of the procedure are given in Section 2.2.2). Specifically, the grid generation procedure creates an unstructured O-grid made of triangular and rectangular elements with a diameter of approximately 80 times the airfoil chord and some peculiar features. The complete grid is shown in Fig. 4.2.3a.

The rectangular elements build up a near-wall region whose thickness is estimated using the flat plate theory (see Fig. 4.2.3b). Furthermore, a closed sharp trailing edge (TE) has been adopted (see Fig. 4.2.3c). In sum, a whole grid has approximately 150000 elements at the discretization level chosen for this problem.



(a) O-grid.

(b) Close view of the airfoil leading edge.

(c) Close view of the airfoil trailing edge.

Figure 4.2.3: Computational domain. Total number of grid elements ~150000.

In the optimization process, the optimizer calls such a procedure after every shape perturbation to build a new grid from scratch. In the present work, the regeneration of the computational grid is preferred over its deformation. Some of the authors have also used this approach for multi-block structured grids for RANS [76]. Indeed, the generation of structured grids usually requires a computational effort comparable to that required for deformation but allows to maintain excellent control over the grid characteristics near the wall even for extensive changes in shape. Conversely, unstructured grids can require a high computational cost for their regeneration and make the deformation methods of existing grids much more attractive. In the present case, given the relative simplicity of the configuration considered, the regeneration of a 2D unstructured grid does not present computational disadvantages compared to the deformation. It also offers better control of the mesh quality near the boundary layer region. Furthermore, the optimization procedure gains greater flexibility because it does not require a re-meshing in correspondence with substantial airfoil shape changes.

The selected grid size is chosen after a mesh convergence study to ensure that the computational meshes generated in the optimization process allow a sufficiently precise evaluation of the QoI (c_d). Table 4.2.2 and Figure 4.2.4 report the preliminary grid convergence study results. In particular, Table 4.2.2 shows the drag coefficient expressed in drag counts [dc] as a function of the total number of grid elements (N) and, for better readability, the associated grid size (h). Grid size h is the square root of the ratio between the number of cells of the most refined grid and the grid under evaluation $\left(h = \sqrt{N_{h=1}/N_i}\right)$, and the drag count is $1\ \mathrm{dc} = 10^{-4} c_d$.

N	Grid size, h	c_d [dc]
99524	2.34	315.5
114532	2.18	292.2
1477424	**1.92**	**280.2**
270434	1.42	270.4
545272	1.00	269.0

Table 4.2.2: Grid convergence on the deterministic optimum airfoil (see Fig. 4.5.2). Test case: $M_\infty = 0.8$, $Re_{c_\infty} = 174 \times 10^6$ and $c_l = 0.1$.

Additionally, in Fig. 4.2.4, the drag coefficient is plotted as a function of the grid size. The selected grid refinement level ($h = 1.92$) introduces, at nominal conditions, a numerical uncertainty (due to discretization error) of 10 drag counts, compared to the most refined grid considered. This error is much lower than the variation in drag due to the operating condition and shape uncertainties, which is one order of magnitude higher (~ 200 dc), as the sensitivity studies reported in the following sections underline (see Fig. 4.5.4).

Hence, the selected grid size provides an acceptable trade-off between the precision of the risk function estimation and the computational cost.

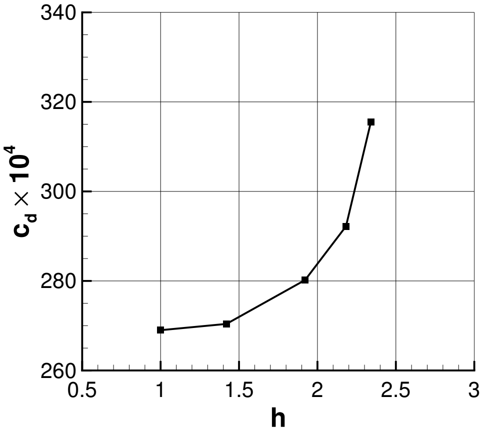


Figure 4.2.4: Drag coefficient (c_d) versus grid size on the deterministic optimum airfoil. Test case: $M_\infty = 0.8$, $Re_{c_\infty} = 174 \times 10^6$ and $c_l = 0.1$.

4.2.2 Shape handling, including uncertainties

A method able to handle the shape under exploration is essential to carry on robust design optimization problems, thus describing the shape as a function of both design and uncertain variables. The shape description approach used in this study is thoroughly described in Section 2.2.1. Particularly, the in-house developed tool $wg2aer$ is utilized. With this approach, the shape of the central section of the BWB aircraft is parameterized as a linear combination of an initial geometry and a number of modification functions. The same technique is used to describe the uncertainties in the geometry by introducing additional modification functions. Equation (2.2) analytically describes the airfoil shape and the related uncertainty. Moreover, for the airfoil shape uncertainty, Hicks-Henne bump functions are introduced (see Eq. (2.3)). The selected positions of the bumps along the chord at the upper and lower surfaces are $s_b^u = s_b^l = \{0.05, 0.1, 0.2, 0.4, 0.6, 0.8\}$.

Finally, it must be mentioned that the uncertain geometrical parameters are all characterized by uniform distribution and their range of variation is reported in Table 4.5.2.

4.3 Risk measures for robust design optimization problems

Section 1.2 reports an extensive discussion of the possible risk measures employed to solve robust design optimization. Here, for the sake of clarity, a short description of the statistical estimator used is summarized. Precisely, the Conditional Value at Risk (CVaR) measure [77] is employed, and its definition is:

$$c^{\gamma} = \frac{1}{1-\gamma}\int_{\gamma}^{1} \nu^{\beta}\mathrm{d}\beta \tag{4.1}$$

The CVaR can be thought of as a weighted average between γ-VaR [78, 79] (or Value-at-Risk or q_{γ} or ν_{γ}) and the losses exceeding it.

When dealing with a finite number of samples, that is, an ECDF [27] consisting of n independent and identically distributed (i.i.d.) observations of the random variable X $(X_1, X_2, \ldots, X_n)$, then the estimator

$$\hat{c}^{\gamma;n} = \inf_{t\in\mathbb{R}}\left\{t + \frac{1}{n(1-\gamma)}\sum_{i=1}^{n}[X_i - t]^{+}\right\} \tag{4.2}$$

is used to estimate the γ-CVaR of L. So, easily, the direct estimate of c^{γ} is obtained:

$$\hat{c}^{\gamma;n} = \hat{\nu}^{\gamma;n} + \frac{1}{n(1-\gamma)}\sum_{i=1}^{n}[X_i - \hat{\nu}^{\gamma;n}]^{+} \tag{4.3}$$

4.4 ECDF estimation and UQ techniques

The presented approach to the uncertainty quantification (UQ) analysis relies on the calculation of the cumulative empirical distribution (ECDF) [27] of the QoI (which in our case is the c_d or a function of c_d and c_m). The risk functions that define the optimization problem are then calculated based on the samples that make up the ECDF. The approximation technique used to estimate the ECDF relies on the QoI gradient information and is the key point that distinguishes the method described here.

The results, both in terms of accuracy of estimation and computational cost, are compared with the direct Monte Carlo sampling of the input random variables. The advantage of Monte Carlo method is that it is simple to implement, but, on the other hand, it converges very slowly $\left(1/\sqrt{N}\right.$, being N the number of samples$\left.\right)$. The latter can be a severe problem because often in optimization runs, it is necessary to use small sample sizes, and this can induce significant errors in the estimation of the QoI statistics.

4.4.1 Adjoint+Gradient

Gradient-related information is often usefully employed in probability theory, as happens, for example, in the First-order second-moment (FOSM) method [80, 81], and more in general in perturbation methods. The main difference compared to these types of methodologies is that, in this case, the gradient and, possibly, the Hessian are not used to approximate the statistical moments, but to obtain an approximation of the ECDF directly. Hence, the QoI (q) is approximated by a Taylor series at the nominal unperturbed vector $\mathbf{u}_0$:

$$
\begin{aligned}
q(\mathbf{u}) = q(\mathbf{u}_0) + \sum_{i=1}^{n} \frac{\partial q(\mathbf{u}_0)}{\partial u^{(i)}} \left(u^{(i)} - u_0^{(i)} \right) + \\
\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 q(\mathbf{u}_0)}{\partial u^{(i)}\, \partial u^{(j)}} \left(u^{(i)} - u_0^{(i)} \right) \left(u^{(j)} - u_0^{(j)} \right) + \\
\dots
\end{aligned}
\tag{4.4}
$$

with $u^{(i)}$ the i-th component of vector $\mathbf{u}$. Only the case in which the Taylor series stops at the first order is considered in the present work. Second-order methods will be introduced in the subsequent developments. The use of non-intrusive methods, such as finite differences (FD), is not convenient for obtaining the QoI gradients with respect to the uncertain variables at an acceptable computational effort. On the contrary, intrusive approaches [82], like the adjoint method introduced by Jameson [83] and widely used in aerodynamic shape design [84–86], provide those gradients at a cost comparable to a flow solution, regardless of the number of uncertain variables.

Consequently, the discrete adjoint method implementation based on Algorithmic Differentiation (AD), added in SU2 by Albring et al. [87], is used here. In this particular implementation, *Duality-Preserving* methods guarantee that the adjoint solution have the same convergence rate as the flow solver. Furthermore, advanced AD techniques as *expression templates* and *local preaccumulation* provide efficiency to the code.

The adopted computational strategy is summarized in Fig. 4.4.1 and explained in the following lines.

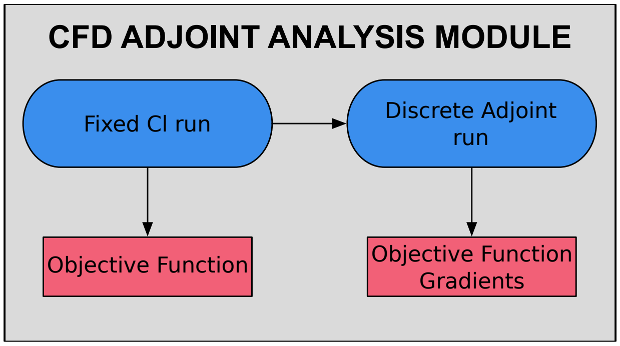


Figure 4.4.1: Computational model chain.

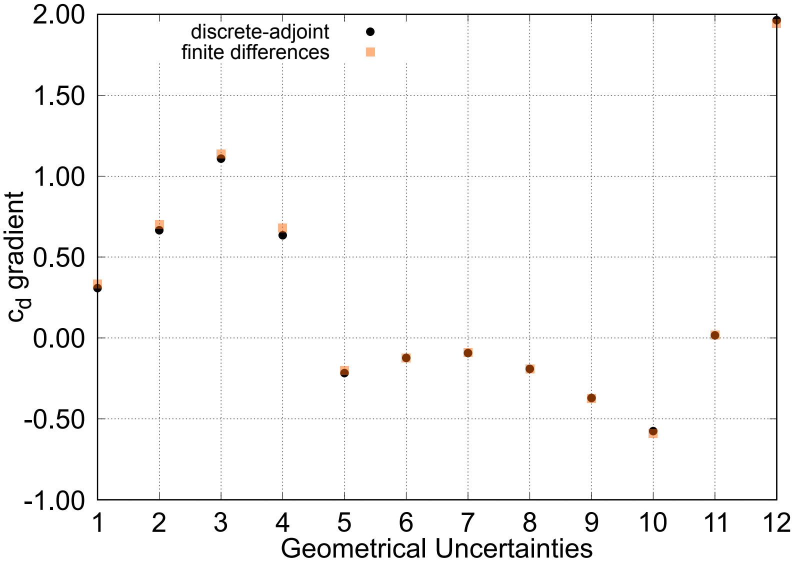


Figure 4.4.2: Example of discrete-adjoint gradients evaluated for the upper (1-6) and lower (6-12) surface geometrical uncertainties of the baseline airfoil.

1. The RANS flow solver gives the QoI of the problem (c_d) and additional aerodynamic characteristics needed for the problem constraints (i.e., c_m).

2. RANS adjoint solution returns the gradients of the QoI with respect to the uncertainty variables almost at the cost of one RANS flow solution.

3. A linear approximation of the QoI is built according to Eq. (4.4) using the extracted gradients.

4. Empirical Cumulative Distribution Function (ECDF) is then calculated.

5. Finally, CVaR is estimated from the ECDF by means of Eq. (4.3).

Figure 4.4.2 shows an example of the c_d gradients related to the geometrical uncertainties evaluated through the method mentioned above.

$(\partial c_d/\partial M)_0$	$(\partial c_d/\partial \alpha)_0$
0.6755454	0.0090269

Table 4.4.1: Example of discrete-adjoint gradients for operational uncertainties.

The result are in good agreement with the finite difference method. In addition, the discrete-adjoint gradients related to the operational uncertainties are reported in Table 4.4.1.

Adjoint Solution Convergence Issues

It is clear that when many uncertain parameters affect only one or a few cost functions, then the adjoint solution is the best choice for gradient evaluation, both in terms of effectiveness and efficiency. Nevertheless, it is crucial to obtain a well-converged flow solution before passing the solved flow field to an adjoint solver. Indeed, regardless of the particular adjoint method (i.e., continuous or discrete), the adjoint equations intrinsically depend on the flow field solution. Therefore, if an accurate evaluation of the QoI gradient through an adjoint solution is sought, it is very likely that the convergence level of the flow solution should reach a level much higher than the one usually suitable to compute the QoI only with accuracy suitable for optimization. To put it another way, this means that the convergence level, which is commonly proper for optimization purposes, would not be sufficient to find a suitable QoI gradient with a successive adjoint solution.

This point is a hindrance to the introduction of adjoint-based optimization methods in an industrial context where, often, RANS equations with complex geometries have to be solved without the possibility to reach very high convergence levels. Indeed, even in the 2D design application here studied, some gradient inaccuracies due to both the incomplete convergence of the flow field solution or to the divergence of the discrete adjoint solution may appear. Consequently, some extra CPU time to improve the average convergence level of the solutions has to be spent. For what concerns the presented strategy, it is mandatory to check the convergence of both the flow field and adjoint field solutions carefully to avoid leading the optimizer in the wrong direction.

According to the tests carried out for typical airfoil shapes, the trend found in the residual of the adjoint variables is a rapid descent of the residual, which often tends to flatten. These tests evidenced that an acceptable level of residual reduction is around 2.5 orders of magnitude, provided that there is no oscillation in the calculated sensitivities.

4.5 Design application example

In this section, the physical problem described in Section 4.2 is translated into a single-point constrained optimization problem. According to the engineering practice developed in previous works [88], the problem is first solved using a classic deterministic approach. Subsequently, the risk function based method, using CVaR, for robust design is applied. Finally, the baseline airfoil, the deterministically optimized one, and the airfoil resulting from the robust design loop are analyzed and compared from the point of view of uncertainty quantification.

4.5.1 Optimization problem setup

This design optimization example studies the improvement of the BWB central airfoil (see Fig. 4.2.1) performance by minimizing the drag coefficient (c_d), subjected to geometric and aerodynamic constraints. Regarding the geometrical constraints, the airfoil percentage thickness with respect to the chord is fixed at the base value ($t_{\%}$), while, to obtain realistic shapes, constraints on the leading edge radius (LER), the trailing edge angle (TEA), and the airfoil percentage thickness with respect to the chord at $x/c = 0.85$ (TAT) are introduced. In addition, considering the aerodynamic restrictions, special attention is dedicated to the airfoil pitching moment coefficient c_m, which for BWB configurations is a critical parameter due to the absence of the elevators. For this reason, two constraints for the pitching moment coefficient are used in order to keep its value properly confined, as required by trim aspects. For the sake of clarity, it has to be pointed out that the c_m is evaluated with respect to the aerodynamic center, and it is considered positive in the case of "*nose up*" pitching moment. In mathematical terms, the design optimization example reads:

$$\left\{\begin{array}{ll} \min\limits_{\mathbf{w}} & c_d(\mathbf{w}) \\ \text{subject to:} & \\ & t_{\%} = 16.00 \\ & \text{LER} \geq 0.00781 \\ & \text{TEA} \geq 22.0^{\circ} \\ & \text{TAT} \geq 0.06658 \\ & c_l = 0.1 \\ & c_m \leq 0.04 \\ & c_m \geq -0.04 \\ & \text{error} = 0 \end{array}\right. \tag{4.5}$$

The penalty approach is used to handle the constrained optimization problem as an unconstrained one:

$$\min_{\mathbf{w} \in W \subseteq \mathbb{R}^n} c_d(\mathbf{w}) + P(\mathbf{w}) \tag{4.6}$$

with

$$\begin{array}{rcl} P(\mathbf{w}) & = & k^1 p^+(\text{LER}, 0.00781) + \\ & & k^2 p^+(\text{TEA}, 22.0^{\circ}) + k^3 p^+(\text{TAT}, 0.06658) + \\ & & k^4 p^+(c_m, 0.4) + k^4 p^-(c_m, -0.04) + k^5 p^+(\text{error}, 0) \end{array} \tag{4.7}$$

In this case, all the constraints except those regarding the lift coefficient and the airfoil percentage thickness with respect to the chord are treated as quadratic penalties:

$$p^+(x, y) = \left\{\begin{array}{ll} 0 & \text{if } x \geq y \\ (x-y)^2 & \text{if } x < y \end{array}\right. \tag{4.8}$$

and

$$p^-(x, y) = \left\{\begin{array}{ll} (x-y)^2 & \text{if } x \geq y \\ 0 & \text{if } x < y \end{array}\right. \tag{4.9}$$

Instead, the c_l and the thickness constraints do not appear because they are automatically satisfied during the computation procedure by changing the angle of attack and by re-scaling the airfoil thickness to the assigned value, respectively. The numerical values chosen for the k^i coefficients are: $k^1 = 5000,\ k^2 = 10,\ k^3 = 30,\ k^4 = 1000,\ k^5 = 1000$. The transformation of a constrained optimization problem into an unconstrained one through the penalty approach is always a delicate process, as the choices of the weights of the penalization terms profoundly change the shape and features of the search space. Here, in particular, the violation of the constraint on leading-edge radius is highly penalized, as it is well known the tendency of the optimization processes to gain performance in these working conditions by reducing the leading edge radius. Unfortunately, this often induces bad off-design stall behavior, especially near stall, and should be avoided. It has to be also avoided that failed CFD computations may deceive the optimizer. Hence, a very high weight to the related constraint (error in Eqs. (4.5) and (4.7)) is added. Finally, a stiff penalty is also imposed on the pitching moment constraints, because a too marked change in this constraint would require a radical revision of the BWB's planform. This is one of the constraints that have the most significant impact on robust optimization.

4.5.2 Deterministic problem solution

Twenty design variables corresponding to the aforementioned w_i weights in Eq. (2.2) are used to describe the shape of the wing section. The parameters used for the CMA-ES [89] optimization algorithm, described in Section 2.1.2, are the maximum number of allowed evaluations, the population size λ, and the initial standard deviation σ.

This optimization task focuses on finding a starting point for robust optimization close enough to a deterministic global optimum. Hence, the deterministic aerodynamic optimization process efficiency is not a primary concern, even if the constraints connected to the available computational resources are carefully considered and in line with previous CFD-based evolutionary design work [76]. Table 4.5.1 reports the parameters set for this problem.

Maximum evaluations	Population size	Initial standard deviation
3000	40	0.15

Table 4.5.1: CMA-ES parameters for the deterministic optimization run.

The CMA-ES setup aims at a broad exploration of the search space. Therefore, some care is necessary to reduce the elapsed run time for the fluid dynamic solver. Since it is necessary to work at fixed c_l, SU2 works in the variable angle of attack mode for 5000 iterations. With this setup, each SU2 run requires about 35 minutes of running time on 6 Intel (R) Xeon (R) cores E5-2670 CPU at 2.60 GHz. Considering that each generation is evaluated in parallel, using 240 cores, the optimization run has required approximately 44 hours. Considering that c_d is more sensitive than c_l to the fluid dynamic field convergence level, the optimal solution has been re-evaluated with a variable angle of attack loop of 20000 iterations.

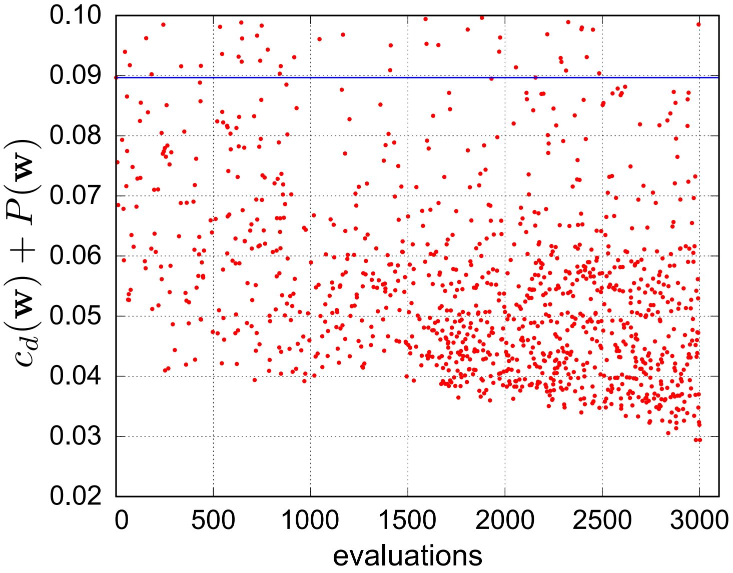


Figure 4.5.1: CMA-ES convergence history for the deterministic optimization. The blue line indicates the objective function value of the baseline airfoil (—).

As shown in Fig. 4.5.1, after 3000 function evaluations, the convergence is not completely reached, but a substantial improvement is achieved, and the shape in Fig. 4.5.2 is obtained. The objective function value for the optimized airfoil is 0.029409 (with the penalty term $P = 0$), which corresponds to a 67.2% reduction with respect to the baseline value of 0.089676 ($c_d = 0.049004$, $P = 0.040672$). The refined fixed lift loop re-evaluation led to $c_d = 0.028025$. Furthermore, all the aerodynamic and geometrical constraints are respected. Even if the absolute optimum is not reached, it was decided not to spend more computational resources on this problem because the deterministic optimization goal is to provide a good starting design for the robust optimization. Adopting this strategy, a robust design that is not only less vulnerable to the uncertainties, but shows a better aerodynamic performance too, is expected.

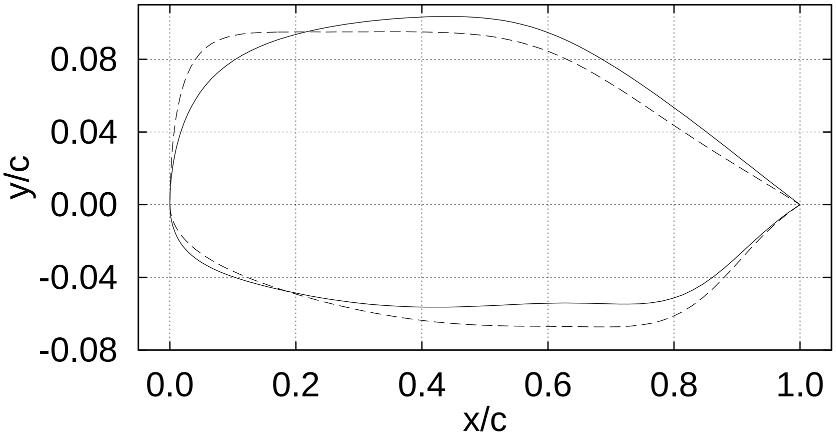


Figure 4.5.2: Airfoil shape comparison of the deterministic optimized airfoil (—) versus the baseline airfoil (– – –). The y-coordinate is magnified and proportions are not respected.

4.5.3 Robust Design Optimization

After solving the deterministic optimization, a robust one must be carried out to improve the airfoil performances under uncertainty. Only uncertainties in the airfoil shape and the operating conditions are considered. The first uncertainty source has been introduced in Section 4.2.2. In this particular case, 12 uniformly distributed random variables are used to represent the airfoil stochastically perturbed shape. Moreover, the working conditions that are considered uncertain are the Mach number and the Angle of Attack. The just mentioned uncertainties are modelled as four-parameter beta distribution, whose density function is given by

$$f(y;\eta,\theta) = \frac{\gamma(\eta+\theta)(y)^{\eta-1}(1-y)^{\theta-1}}{\gamma(\eta)\gamma(\theta)} \tag{4.10}$$

with shape factors η, θ, and a scale and translation given by $y = (x - \mathrm{loc})/\mathrm{scale}$. Mach is characterized by

$$\eta = 2, \quad \theta = 2, \quad \mathrm{scale} = 0.08, \quad \mathrm{loc} = 0.76$$

while the angle of attack is characterized by

$$\eta = 2, \quad \theta = 2, \quad \text{scale} = 1.0, \quad \text{loc} = -0.5$$

Table 4.5.2 reports the variation range of these uncertainties.

Uncertainty	Range
Mach, M	$[\,0.78\,, 0.82\,]$
Angle of Attack, $\Delta\alpha$	$[-0.15^\circ, 0.15^\circ]$
Geometry, U_j	$[-0.0007\,, 0.0007\,]\,,\ j = 1, \ldots, 12$

Table 4.5.2: Summary of uncertain parameter definition in the four benchmark cases.

As previously mentioned, by introducing random variables (in the airfoil geometry definition and in the operating conditions) the QoI is now a functional. Hence, the risk function CVaR is used to map the chosen QoI into $\mathbb{R}$. Specifically, the CVaR is estimated with a confidence level γ equal to 0.9. In addition, it is interesting to assess the impact of random perturbations only on the drag force. Thus, constraints are only computed at the nominal values of the design parameters (without considering the effect of uncertainties). These considerations led to the following choice for the objective function:

$$\min_{\mathbf{w} \in W \subseteq \mathbb{R}^n} \mathrm{CVaR}^{0.9}\left(c_d(\mathbf{w}, \mathbf{u})\right) + P(\mathbf{w}) \tag{4.11}$$

The procedure based on the adjoint method explained in Section 4.4.1 is used to solve this optimization problem.

4.5.4 Deterministic optimum sensitivity analysis

Before starting with the robust optimization, it is appropriate to carry out a sensitivity analysis for the sources of uncertainty [90], and it is necessary to use the same uncertainty parameterization of the robust optimization phase to avoid ambiguities and allow a fair comparison of robust and deterministic design. Given the availability of the gradient obtainable with the adjoint method, a local method was preferred. The starting point is the knowledge of the partial derivatives of the quantity of interest, in this case, the c_d, with respect to the parameters that define the uncertainty in the robust design problem:

$$\left.\frac{\partial c_d}{\partial u_i}\right|_{\mathbf{u}^0}$$

with $\mathbf{u}^0$ representing the nominal configuration, hence, without uncertainties both in the operating conditions and in shape. The gradient obtained is shown in Fig. 4.5.3, while Fig. 4.5.4 shows the maximum overall variation corresponding to the variation margins of the uncertainties.

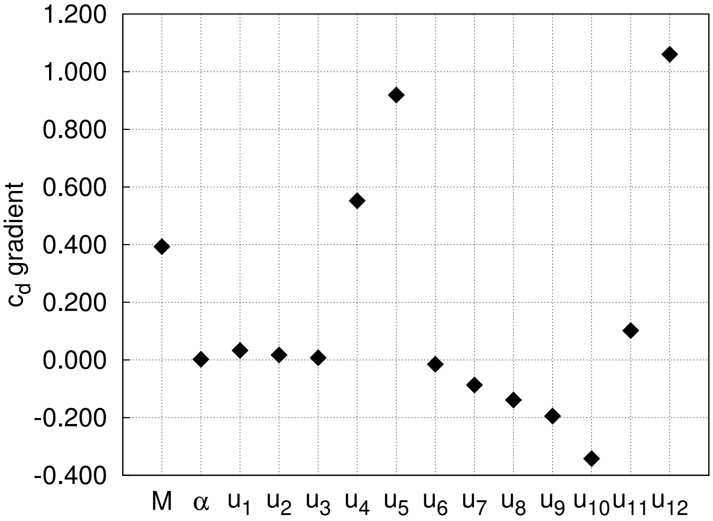


Figure 4.5.3: Gradient of the c_d with respect to the parameters representing the uncertainties for the baseline (deterministic optimum) configuration.

The choice of the range of variation of the uncertain parameters is linked to the physical plausibility of the robust optimization problem, and the information obtained from the sensitivity study helps to understand how significant the role of each of the uncertain parameters is. It is, immediately, observed that the preponderant effect is due to the uncertainty on the Mach number. However, the non-negligible effects due to uncertainty about geometry are also evident, which, therefore, must be considered in the design process. The contribution of the angle of attack uncertainty is smaller than that due to the other uncertainty sources. Nevertheless, it is considered non-negligible since it corresponds to a variation in drag counts of about 7, a significant variation for aerodynamicists.

Finally, it is observed that the actual value of the derivatives of the variables that control the shape of the profile directly depends on the scale of the modification functions that they control, as appear evident from Eq. (2.2). However, the sensitivity to shape uncertainty, as reported in Fig. 4.5.4, is not affected by this scaling effect.

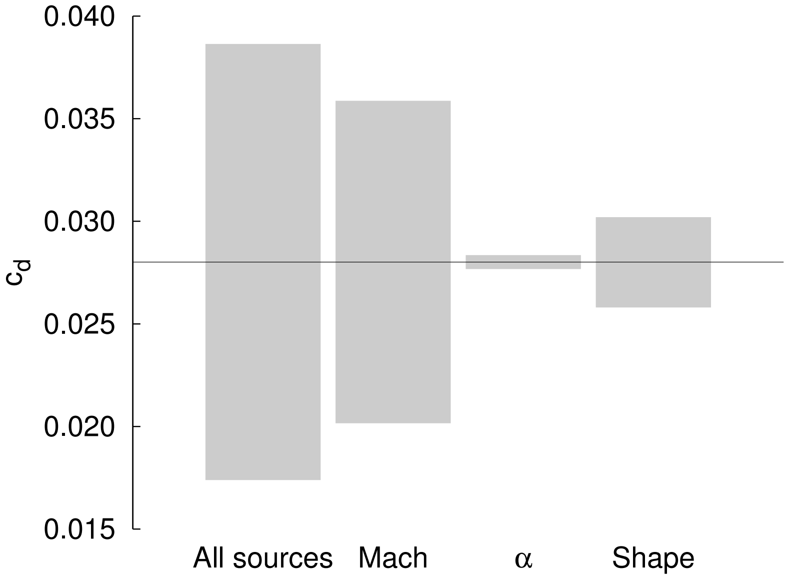


Figure 4.5.4: Linear approximation of the baseline (deterministic optimum) sensitivities.

4.5.5 Preliminary objective function tuning

Preliminary tests on this objective function have shown that the approximation technique used does not always lead to consistent results. In particular, the approximation may lead to an overestimation of the improvement of the objective function.

Figure 4.5.5 shows an example of this behavior. Indeed, the solution reported in blue (Solution 2) and the one shown in red (Solution 1) lead to very similar CVaR values if the approximate ECDF is used, while, when CVaR is recalculated using a classic Monte Carlo sampling, the red solution is much closer to the black one (Solution 0). Overall, the Monte Carlo estimate of the red solution turns out to be substantially worse than the blue solution. An in-depth analysis of these solutions has allowed understanding that the linear approximation implicit in the use of the gradient does not allow to model well the behavior of the airfoil when the Mach number is near the upper limit of the uncertainty range.

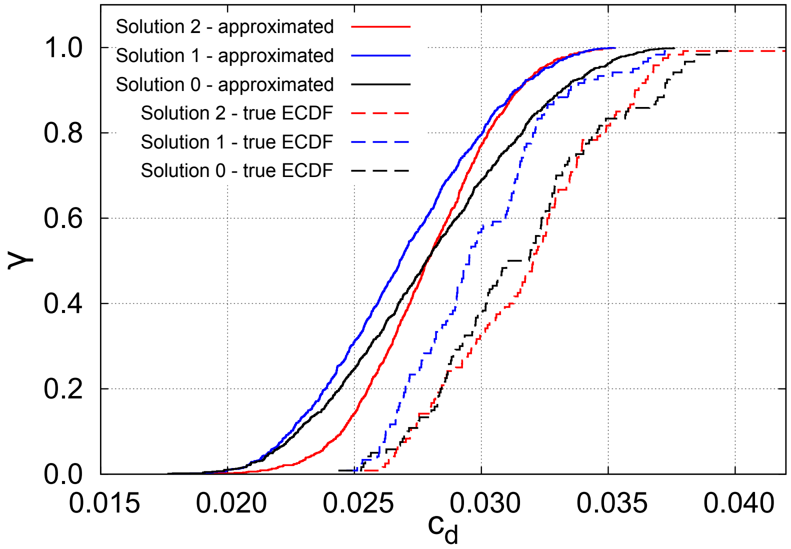


Figure 4.5.5: Comparison between approximated and "true" ECDFs for the drag coefficient (c_d).

Hence, before switching to a more precise and expensive second-order model, an ad-hoc robust optimization problem redefinition is introduced to overcome the highlighted problem. Considering that the part that is modelled the worst with the linear approximation of CVaR is the one at high Mach numbers, it was decided to introduce a second deterministic design point at these working conditions. Particularly, the drag coefficient with nominal values of the design parameters without taking into account uncertainties at a Mach number equal to 0.82, $c_{d,M=0.82}(\mathbf{w})$. Hence, the new robust optimization problem reads:

$$\min_{\mathbf{w}\in W\subseteq\mathbb{R}^n} \mathrm{CVaR}^{0.9}\left(c_d(\mathbf{w},\mathbf{u})\right) + P(\mathbf{w}) + c_{d,M=0.82}(\mathbf{w}) \qquad (4.12)$$

As for the deterministic optimization, the CMA-ES is the algorithm chosen for the optimization. The algorithm parameters are reported in Table 4.5.3. Smaller values for the population size and initial standard deviation are selected with respect to the deterministic optimization setup. The aim is to perform a local search (exploitation) of the optimal region found in the deterministic optimization. Moreover, when starting from a deterministic optimum design, a small value of the initial standard optimization may be preferred since the design may be close to an unfeasible region. Hence, if large perturbations are performed to the deterministic optimum design, the proposed designs might violate the constraints. Thus, the initial population may be composed, in a large proportion, of unfeasible candidates, and this is undesirable.

Maximum evaluations	Population size	Initial standard deviation
3000	12	0.02

Table 4.5.3: CMA-ES parameters adopted for robust design optimization.

By the way, the optimization process was stopped after 2653 evaluations at the end of the 221st generation because, after the optimal value found during the 160th generation, there were no further improvements. The evolution history of the optimization is reported in Fig. 4.5.6.

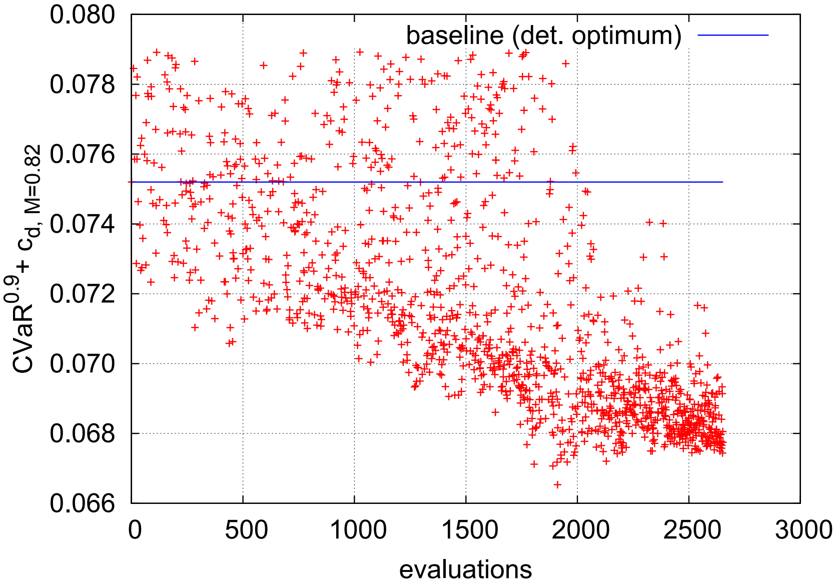


Figure 4.5.6: CMA-ES Evolution history for the robust optimization. The blue line indicates the objective function value of the deterministic optimum (baseline) airfoil (—).

4.5.6 Robust Optimization Solution

The optimization design problem is composed of two design points, namely, the risk measure $\mathrm{CVaR}^{0.9}\left(c_d(\mathbf{w},\mathbf{u})\right)$ and the drag coefficient at a different Mach number working condition $c_{d,M=0.82}(\mathbf{w})$ (the penalizing term $p(\mathbf{w})$ vanishes for feasible solutions). Hence, it is of special interest not only the study of the evolution history of the complete objective function (see Fig. 4.5.6), but also, the evolution history of the different design points (see Figs. 4.5.7 and 4.5.8). Moreover, both contributions are plotted together in Fig. 4.5.9.

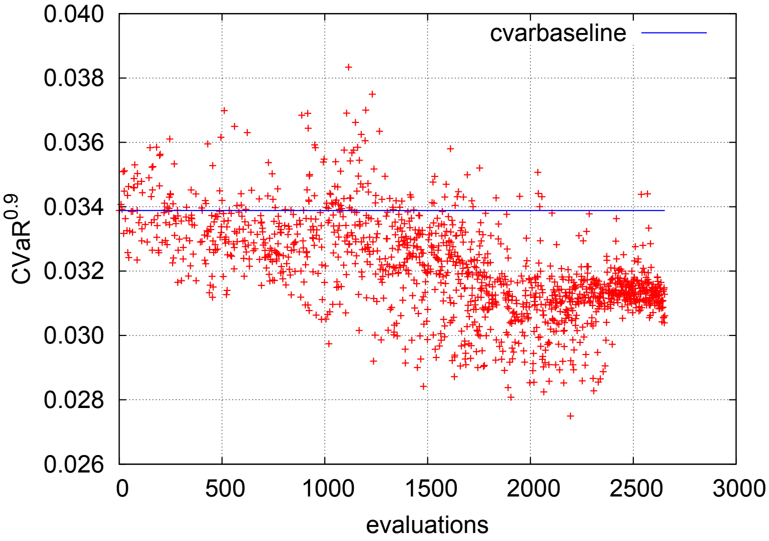


Figure 4.5.7: CMA-ES Evolution history of the $\mathrm{CVaR}^{0.9}\left(c_d(\mathbf{w},\mathbf{u})\right)$ contribution. The blue line indicates the $\mathrm{CVaR}^{0.9}\left(c_d(\mathbf{w},\mathbf{u})\right)$ value of the deterministic optimum (baseline) airfoil (—).

Observing the evolution of the evolutionary histories of the objective function and its components allows us to infer, albeit limited to the problem under consideration, some key features of the presented approach to robust optimization. Figure 4.5.6, related to the complete objective function, shows that its improvement stops after about 2000 evaluations. The comparison of Fig. 4.5.7, related only to the CVaR component, and Fig. 4.5.8, related to the $c_{d,M=0.82}(\mathbf{w})$ component, shows that after the 2000 evaluation threshold

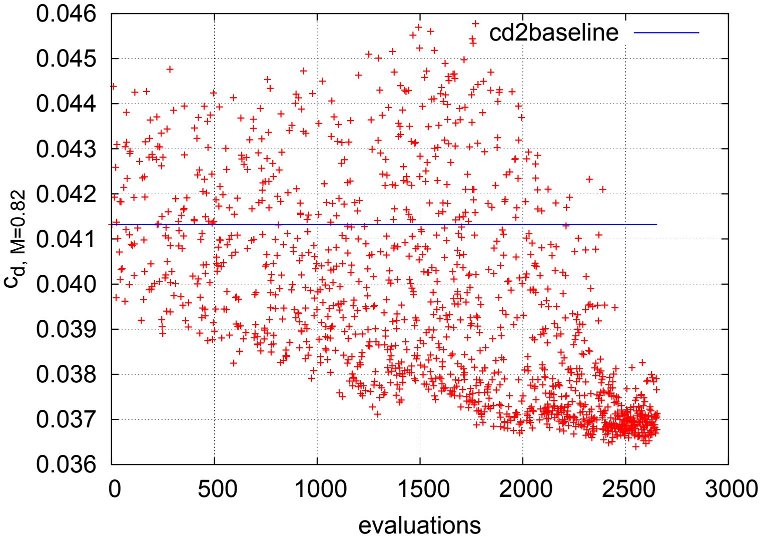


Figure 4.5.8: CMA-ES Evolution history of the $c_{d,M=0.82}(\mathbf{w})$ contribution. The blue line indicates the $c_{d,M=0.82}(\mathbf{w})$ value of the deterministic optimum (baseline) airfoil (—).

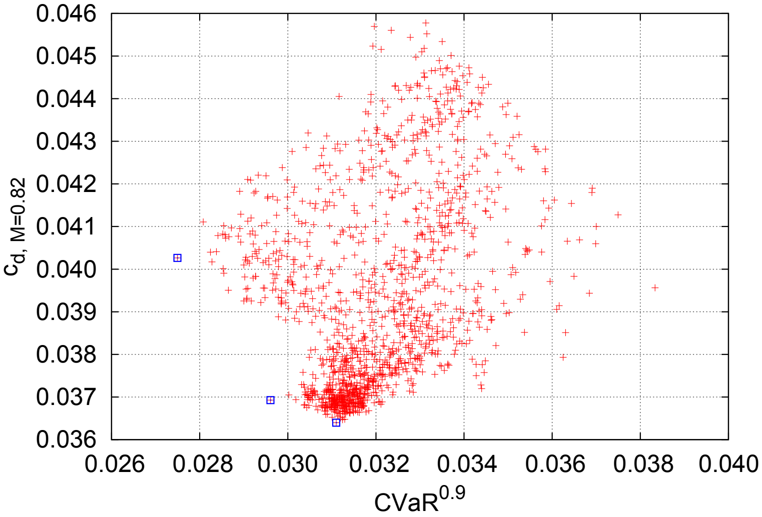


Figure 4.5.9: $\mathrm{CVaR}^{0.9}\,(c_d(\mathbf{w},\mathbf{u}))$ and $c_{d,M=0.82}(\mathbf{w})$ contributions. The solutions marked with a blue square are those corresponding to the best $\mathrm{CVaR}^{0.9}\,(c_d(\mathbf{w},\mathbf{u}))$, the best objective value, and the best $c_{d,M=0.82}(\mathbf{w})$ (□).

the barrier effect of the second design point prevails because, probably, the best CVaR values that appear beyond this limit are excessively affected by the error related to the linearity of the approximation method. Therefore, in the present case, the optimization trend confirms that $c_{d,M=0.82}$ behaves as a safety parameter for controlling the approximation of the risk measure.

The ECDF comparison of the deterministic and robust optimal solutions are reported in Fig. 4.5.10, with the plots of both the approximated and the exact distributions. The ECDF based on gradient approximation is estimated with 1000 samples, while the "exact" one is obtained using 120 samples. Specifically, the figure compares three different solutions obtained from the optimization process previously reported. In particular, they are the one with the lowest value of the objective function (SOL 1), the one with the best $c_{d,M=0.82}\,(x)$ (SOL 2), and, finally, the one with the best approximate CVaR (SOL 3).

Furthermore, the comparison between the baseline (deterministic optimum) and these three selected robust solutions in terms of approximated and "true" CVaR, the $c_{d,M=0.82}$, and the objective value is given in Table 4.5.4.

$\langle$ID$\rangle$	Approx. $\mathrm{CVaR}^{0.9}$	$c_{d,M=0.82}$	Objective	True $\mathrm{CVaR}^{0.9}$	Comment
SOL 0	0.03388	0.04132	0.07520	0.03805	Determ. Opt.
SOL 1	0.02961	0.03693	$\mathbf{0.06653}$	0.03408	Best Objective
SOL 2	0.03110	$\mathbf{0.03640}$	0.06750	$\mathbf{0.03313}$	Best $c_{d,M=0.82}$
SOL 3	$\mathbf{0.02749}$	0.04026	0.06776	0.03471	Best CVaR

Table 4.5.4: Approximated and "true" CVaR, $c_{d,M=0.82}$, and objective value comparison between the deterministic optimum and the three selected robust solutions.

The first thing to be observed in Fig. 4.5.10 is that the addition of the second control point at $M_\infty = 0.82$ makes all three solutions acceptable, in the sense that all three have an appreciable improvement in the Monte Carlo CVaR compared to the starting one. However, the solution with the best approximate CVaR, despite having an overall excellent performance in terms of c_d reduction, has a deterioration in the high tail of the ECDF, which worsens its CVaR value calculated with Monte Carlo sampling. Finally, the inversion in SOL 1 and SOL 2 between the approximate CVaR and "true" CVaR values (see Table 4.5.4) is probably due to the effect of c_d curve translation induced by the reduction of $c_{d,M=0.82}$.

Generally, the comparison shows an improvement of the upper tail shape, without de-

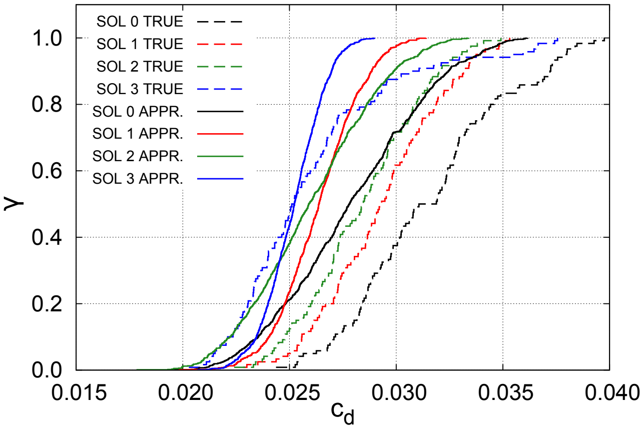


Figure 4.5.10: ECDF comparisons of deterministic and robust optimal solutions.

terioration of the lower tail. This point is a definite advantage of the proposed method compared to classical approaches based on mean μ and standard deviation σ computation. Despite a visible shift, the approximated solution captures the trend of the true ECDF correctly, and therefore the approximated $\mathrm{CVaR}^{0.9}$ is seamlessly usable for robust optimization.

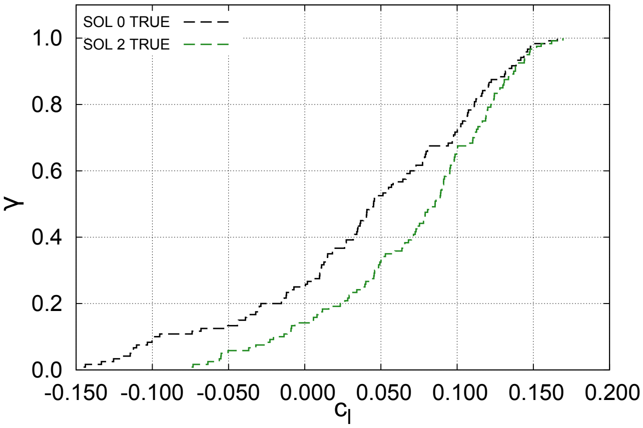


Figure 4.5.11: ECDF for the lift coefficient (c_l) comparison between the deterministic (SOL 0) and robust (SOL 2) optimal solutions.

From now on, the solution with the lowest value of $c_{d,M=0.82}$ (SOL 2) will be referred to as the robust optimum since it provides the lowest "true" CVaR value, which is our design target. On the other hand, it must be pointed out that a designer could prefer the solution that has the best approximated CVaR (SOL 3) since, on average, it shows better performance (lower c_d). In particular, there is a probability less than the 10% that it provides a worse performance than SOL 2. However, SOL 2 is here preferred since the target of this design exercise is the improvement of the upper tail of the ECDF through the CVaR measure.

Another significant result is shown in Fig. 4.5.11. Here, the true ECDF for c_l of the deterministic and the robust solutions are compared. Indeed, although no risk measure was applied to c_l, the presented strategy reduces its variability remarkably. Thus, the optimal ECDF for c_l exhibits a higher slope, and this is more significant in the proximity of the c_l nominal value. Since it is clear the link between AoA and c_l for an airfoil, this behavior is due to the leverage of AoA variability effects, which indirectly influences the c_l variability too. Moreover, as a result of no applying a risk measure to the lift coefficient, the optimized airfoil can provide a negative value of c_l under the effect of the uncertainty.

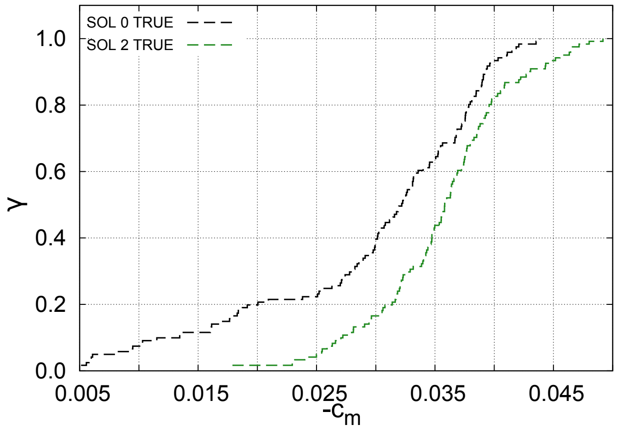


Figure 4.5.12: ECDF for the pitching moment coefficient (c_m) comparison between the deterministic (SOL 0) and robust (SOL 2) optimal solutions.

Likewise, the robust optimization formulation did not consider the uncertainty of the pitching moment. Therefore the constraints on c_m are fulfilled only in the nominal condition. Hence, Fig. 4.5.12 shows that, under uncertainty, lower constraint ($-c_m \leq 0.04$) is violated with a probability close to 20%. Indeed, the introduction of a risk measure applied to the pitching moment coefficient might be beneficial, although it would likely reduce the margin of improvement of the airfoil performances. Incidentally, the c_m ECDF, although shifted, exhibits a reduced variation range with respect to the deterministic optimum airfoil, as it occurs with the lift coefficient.

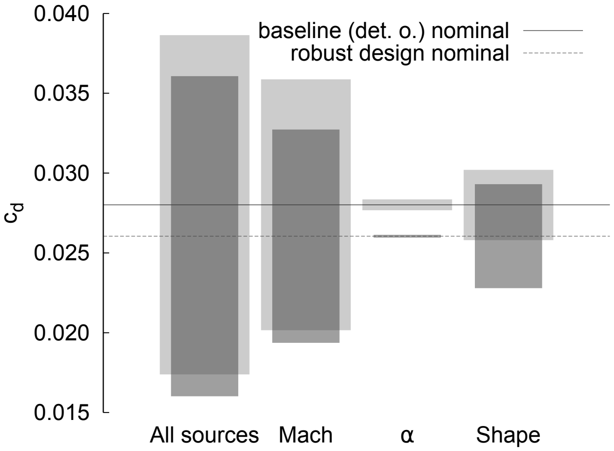


Figure 4.5.13: Deterministic (SOL 0) and robust (SOL 2) optimum sensitivities comparison.

Finally, a linear sensitivity analysis of the robustly optimized airfoil has been performed, and Fig. 4.5.13 shows the comparison with the deterministic optimum. The drag coefficient is reduced with respect to the baseline value. This behavior was expected since, as already mentioned, the deterministic optimization did not reach the absolute optimum. Specifically, the robustly optimized airfoil drag coefficient is 0.0261, which corresponds to a 6.8% reduction compared to the deterministic optimum airfoil ($c_d = 0.0280$). Furthermore, all the aerodynamic and geometrical constraints are respected. As expected, the c_d variation due to all the sources of uncertainty (namely, Mach, angle of attack, and airfoil shape uncertainties) is reduced after the robust optimization. More in detail, the variation caused by the Mach number, although reduced, is still preponderant. Also, the effect due to the angle of attack has nearly disappeared. Finally, the maximum overall variation of c_d as a result of a change in shape has increased. It seems that the variability contribution due to the working conditions has shifted towards the uncertainty in shape. Hence, it could mean that by improving the manufacturing procedures and reducing manufacturing tolerances, the sensibility of the optimized airfoil to uncertainties in working conditions could be significantly reduced.

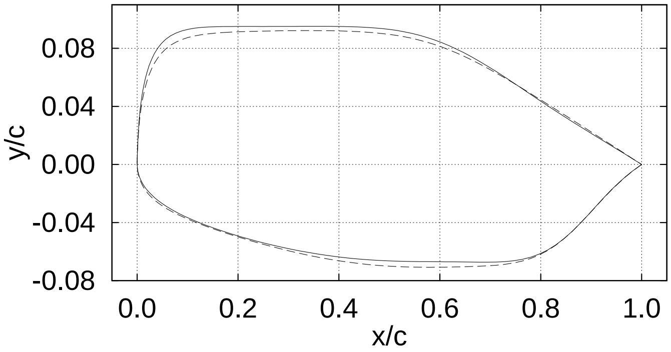


Figure 4.5.14: Airfoil shape comparison of the deterministic optimized airfoil (SOL 0) (—) versus the chosen robust optimized airfoil (SOL 2) (- - -). The y-coordinate is magnified and proportions are not respected.

Figure 4.5.14 shows the shape comparison of the deterministic and robust optimal airfoils.

Moreover, in Fig. 4.5.15-Fig. 4.5.17, the aerodynamic characteristics of the baseline, the deterministic and robust optimum airfoils are compared at the nominal Mach number and at the Mach uncertainty boundaries.

Besides, in Fig. 4.5.18, the drag coefficient is plotted against the Mach number in its uncertainty range for the original airfoil and both deterministic and robust optimum designs. The CFD runs were performed at constant lift coefficient ($c_l = 0.1$). The comparison of the deterministic optimal and the robust one shows that the increase in drag is more gradual and moderate in the latter one, and this is the kind of behavior expected from a robust design. However, when analyzing the drag rise curves, what stands out is that at the nominal design point, the c_d of the robust solution is slightly better (20 dc) than the deterministic one. Normally, it is reasonable to expect the opposite, i.e., that the deterministic optimum is better than the robust one in the nominal condition. However, this behavior is explainable because the baseline solution was quite far (209 dc) from the optimal solution found. Indeed, the deterministic optimization process aimed to explore the optimization landscape rather than refining the solution found. Therefore, it would have been appro-

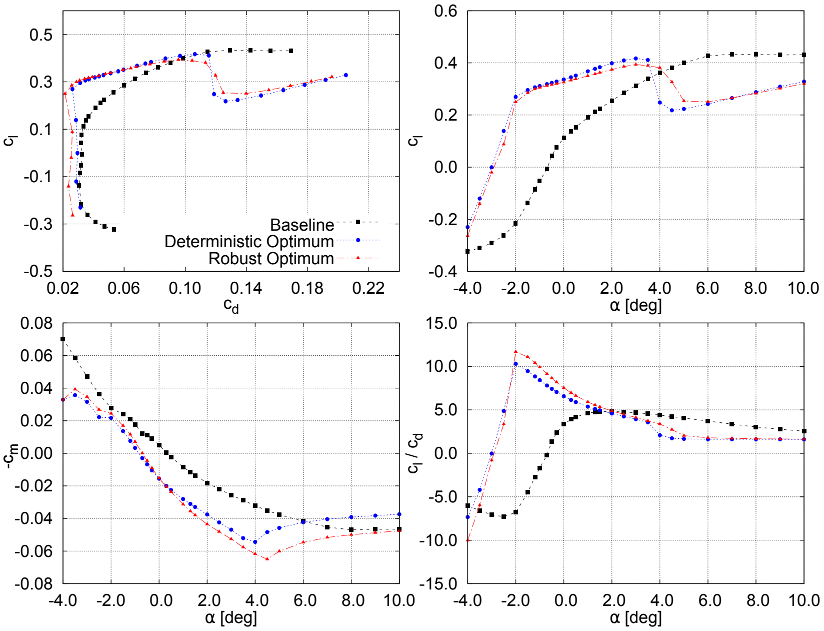


Figure 4.5.15: Aerodynamic characteristic curves of the baseline, deterministic optimum and robust optimum airfoils at $M = 0.78$.

priate to continue the deterministic optimum research phase with a phase of refinement and exploitation. Still, it was considered out of the scope for this research work since the purpose of the deterministic optimization was mainly to provide an appropriate starting point for the robust optimization process. Table 4.5.5 reports drag coefficient values at the nominal Mach for baseline, deterministic optimum, and the three solutions extracted from the robust optimization run for completeness.

⟨ID⟩	$c_{d,M=0.80}$	Comment
BASE	0.04890	Baseline
SOL 0	0.02803	Determ. Opt.
SOL 1	0.02640	Best Objective
SOL 2	0.02605	Best $c_{d,M=0.82}$
SOL 3	0.02524	Best CVaR

Table 4.5.5: Comparison of $c_{d,M=0.80}$ (nominal Mach) of baseline, deterministic optimum and the three selected robust solutions.

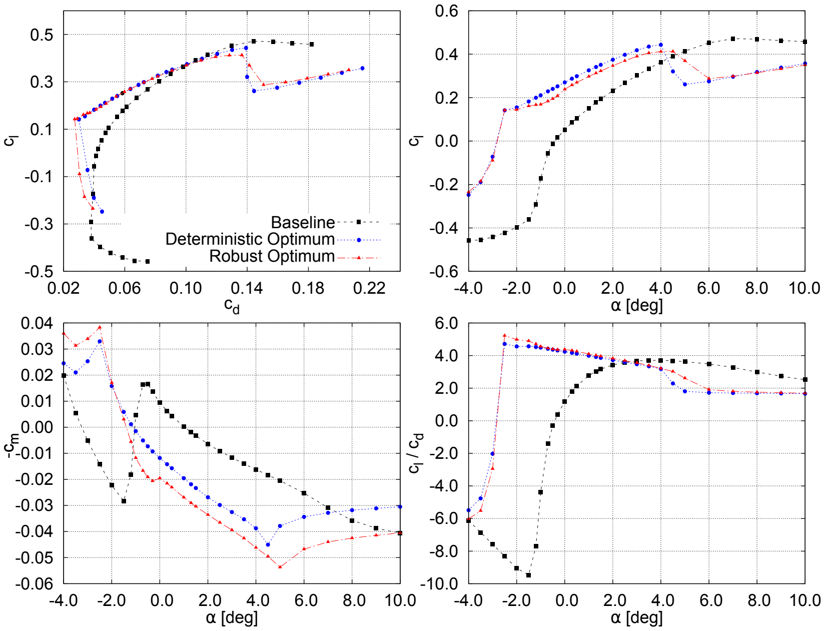


Figure 4.5.16: Aerodynamic characteristic curves of the baseline, deterministic optimum and robust optimum airfoils at M = 0.80.

Finally, in Fig. 4.5.19 the pressure coefficient (c_p) contours for the baseline, deterministic optimum, and robust optimum airfoils are given, together with, the comparison of the c_p on the body surface for the three airfoil. First of all, Fig. 4.5.19a shows that the baseline configuration presents one shock wave on the upper side and three on the lower side. Particularly, the second shock wave on the lower side is weak, whereas the other shocks are quite strong. This is also visible in the pressure coefficient distribution on the body surface. The two optimized airfoils provide a highly different distribution of shock waves. The shock wave presented at the upper surface has a similar structure for the two optimized airfoils. The contour fields (Figs. 4.5.19b and 4.5.19c) show a more extended structure of shock wave on the upper side of the geometry compared to the baseline. By looking at Fig. 4.5.19d, it can be appreciated that a sudden expansion on pressure coefficient precedes a plateau for the baseline airfoil and, then, a compression located at ~ 60% of the chord. Contrary, the optimal solutions have a steeper expansion at $x/c = 0.1$ followed by a smooth compression up to the 40% of the chord where an expansion begins, followed, eventually, by a strong compression at 70% approximately. On the lower side, a different behavior is observed for the three airfoils. For the deterministic optimum airfoil, the three

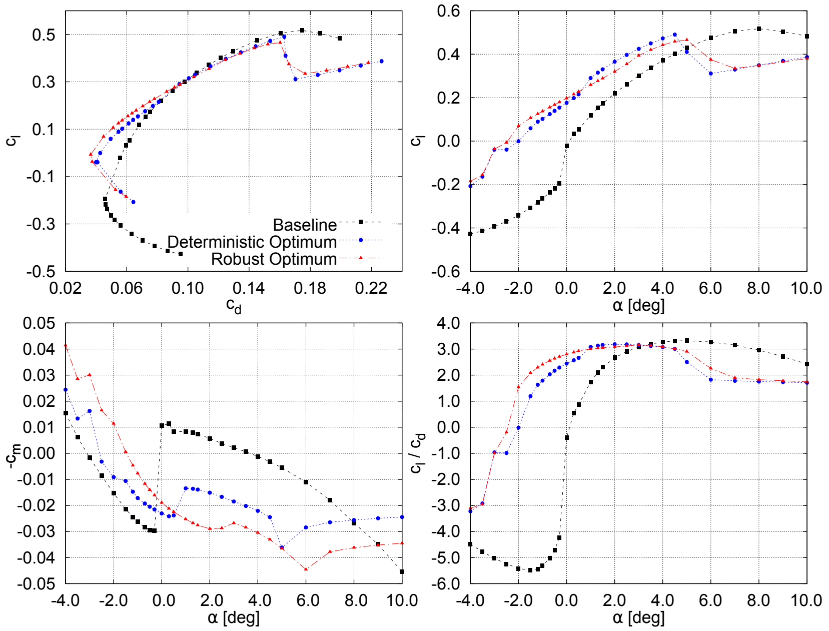


Figure 4.5.17: Aerodynamic characteristic curves of the baseline, deterministic optimum and robust optimum airfoils at $M = 0.82$.

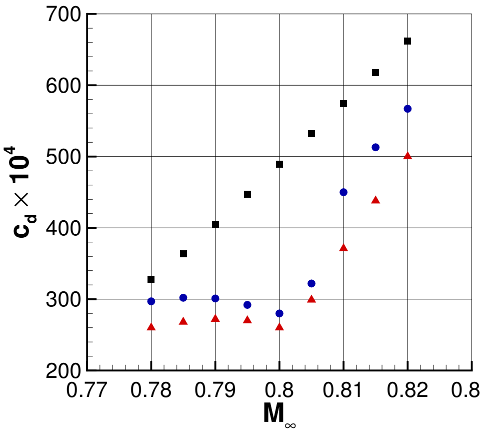
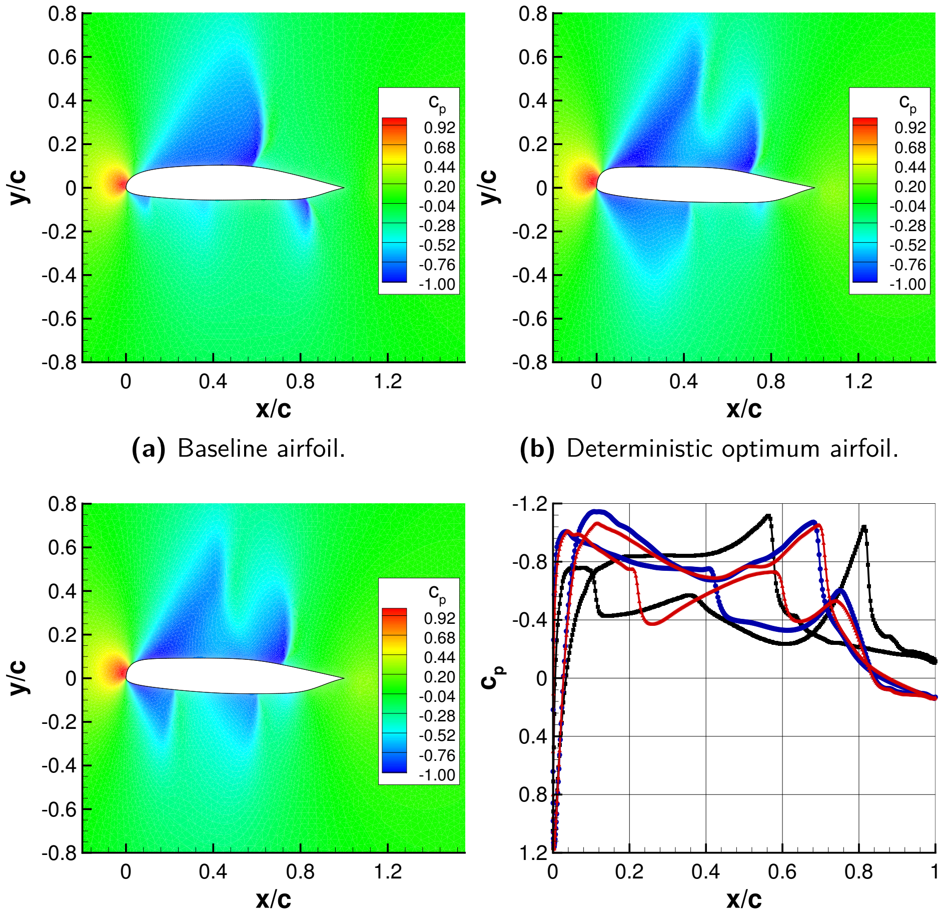


Figure 4.5.18: Drag coefficient (c_d) versus Mach number (M_∞) for the baseline airfoil (■) and deterministic (●) and robust (▲) optimum airfoils. Test case: $Re_{c_\infty} = 174 \times 10^6$ and $c_l = 0.1$.

shock waves presented at the lower surface in the baseline airfoil become two shock waves. Specifically, it has a strong expansion at the leading edge followed by a compression after the 40% of the chord. In addition, there is a second weak shock at $x/c = 0.75$. Besides, in the robust optimum airfoil three shock waves are still present but with different structures with respect to the baseline airfoil. There is an expansion at the leading edge of the airfoil and a sudden compression at the 20% of the chord. After that, a smooth expansion starts at $x/c = 0.4$ that is followed by a compression at $x/c = 0.6$. Finally, a third and weak shock wave is located at the 75% of the airfoil chord.



(a) Baseline airfoil.

(b) Deterministic optimum airfoil.

(c) Robust optimum airfoil.

(d) Body surface c_p

Figure 4.5.19: Figures (a) to (c) provide the c_p contour for the baseline, deterministic optimum, and robust optimum airfoils. Figure (d) shows the pressure coefficient on the body surface for the baseline (−■−), deterministic optimum (−●−), and robust optimum (−▲−) airfoils. Nominal conditions ($M_\infty = 0.80$, $Re_{c_\infty} = 174 \times 10^6$, and $c_l = 0.1$).

From the previous considerations, it can be stated that the reduction of the c_d obtained by the optimal airfoils is mainly due to the weakness of the shock wave located at $x/c = 0.75$ on the lower surface, and due to the downstream movement of the compression at the upper surface shock wave.

4.5.7 Computational Cost

Finally, the following section reports a comparison of the computational effort for calculating the ECDF of both methods (simple Monte Carlo and gradient-based).

When using the plain Monte Carlo approach to estimate CVaR through the empirical cumulative distribution, the number of samples needed to get a satisfactory accuracy can be very high. In the problem here presented, a minimum of 20 samples is necessary to obtain a barely acceptable estimation of CVaR. In particular, the construction of the ECDF requires a single fixed lift coefficient CFD run at the nominal condition point ($M_\infty = 0.8$ and $c_l = 0.1$) plus 20 fixed angle-of-attack CFD runs constructed by sampling the random variables that describe the problem uncertainty. The computational cost is normalized with respect to the average execution time of a fixed α run. The average cost of a fixed c_l run at proper convergence is instead roughly doubled with respect to the fixed α case. The total cost of the Monte Carlo method is summarized in Table 4.5.6.

	20 fixed α runs	c_l fixed run	Total
Monte Carlo ECDF cost	20	2	22

Table 4.5.6: Computational cost of CVaR estimation using Monte Carlo sampling.

The adjoint-based ECDF approximation requires a first CFD run at fixed c_l, like in the Monte Carlo approach. After that, the discrete adjoint solver computes the gradient of the c_d respect to the uncertain variable. The computational cost of this run is equivalent to the fixed lift coefficient run. Finally, the CFD run to compute $c_{d,M=0.82}$ is carried out. The convergence criterion of this run is relaxed. Thus, only one-tenth of the iterations of the fixed c_l run is necessary on average. In Table 4.5.7, the computational cost of the proposed approach is given.

The ratio of the computational costs between Monte Carlo and gradient-based approaches is $22/4.2$. Hence, the gradient-based approximation for CVaR calculation provides a CPU cost reduction of around 80%. This savings estimation is very conservative because

	c_l fixed run	Adjoint run	$c_{d,M=0.82}$ run	Total
Adjoint based ECDF cost	2	2	0.2	4.2

Table 4.5.7: Computational cost of CVaR estimation using a gradient-based approximation.

the Monte Carlo samples used are just 20, which is the lowest allowable limit to get significant results in the current design problem.

4.6 Conclusions

In this chapter, a robust optimization approach based on CVaR risk measure has been successfully employed for an aerodynamic design problem of industrial interest. The CVaR introduces the possibility of working asymmetrically on the ECDF, which allows improving the upper tail of the distribution without deterioration of the lower one. In other words, the risk measure adopted introduces the possibility of working asymmetrically on the ECDF, while this would not be possible with other classical techniques such as those based on the mean value and the standard deviation. The gradient-based strategy for the linearization of the QoI has allowed the effective use of UQ for an aerodynamic design optimization problem because only one more adjoint-flow solution for each individual is necessary to get an acceptable approximation of the risk function. However, it must be borne in mind that the convergence level that the adjoint solver must reach to calculate the gradient with sufficient precision is higher than that required by the field solver for a sufficiently precise estimate of the overall aerodynamic characteristics. Add to this the fact that the fluid dynamic field, which is an input of the adjoint solver, must also reach a higher convergence level, again for the gradient calculation accuracy. However, even considering this increase in computational time, the overall balance clearly leans in favor of the method presented here compared to the classic Monte Carlo, as is extensively discussed in the previous section. In addition, the shift found between the true ECDF and the gradient-based approximation is a problem that will be the focus in subsequent developments, possibly, using a second-order approximation of the QoI. To this end, various approaches are possible that range from approximated Hessian estimation using gradient information [91] to algorithmic differentiation [92] and "tangent-on-reverse" methods [93]. In any case, it is necessary to find a compromise between the need for accuracy in estimating the risk function and the efficiency of the optimization procedure, given that the evaluation of the second derivatives considerably increases the computational load, even using an intrusive, adjoint field-based approach. However, although a higher-order approximation is desirable, the presented method shows quite a good trade-off between cost efficiency and the optimization process effectiveness.

5

Surrogate-Based Robust Optimization of Aerodynamic Shapes

This chapter addresses robust optimization efficiency from a complementary point of view to that of the previous chapter. Indeed, the previous chapter focused on using an intrusive method for reducing the computational time of robust optimization. However, it is not always easy to have all the information needed to implement an intrusive method and, in particular, as in our case, it might be difficult or expensive to obtain the derivatives of the quantity of interest with respect to the uncertain variables. Thus, in this chapter, a new non-intrusive and adaptive method is proposed that, although having a slightly lower efficiency than the gradient-based intrusive one, still significantly improves the robust optimization process efficiency. In more detail, among all the possible methodologies, such as Polynomial Chaos Expansion or Stochastic Collocation, this chapter investigates the use of surrogate models to perform robust optimization, and the focus is, in particular, on Gaussian Processes regression. The first part is an introduction to Surrogate-Based Optimization (SBO). The DLR's SMARTy toolbox is used to illustrate the basic features of SBO. DLR made available SMARTy for this Ph.D. research work within the UTOPIAE project framework. Thanks to UTOPIAE, the author spent a research secondment at the Institute of Aerodynamics and Flow Technology. She joined the $C^2A^2S^2E$ (Center for Computer Applications in AeroSpace Science and Engineering) group, where she learned to use SMARTy. After giving some illustrative examples, two real-world problems are solved in Sections 5.2 and 5.3. The work presented in Section 5.3 was performed in collaboration with ESTECO and the University of Trieste. Both studies were presented at the International Conference on Uncertainty Quantification & Optimization UQOP2020 (in Press). In Section 5.2, the surrogate model is used to approximate the cumulative distribution function to compute the CVaR, while a population-based optimization algorithm searches the design space. Instead, a surrogate assisted search algorithm is used in Section 5.3.

5.1 Surrogate-based optimization overview

A summary of surrogate-based optimization (SBO) methods is the subject of this section, while more exhaustive reviews of SBO foundare in the following references: Queipo et al. [94], Simpson et al. [95], and Forrester and Keane [96].

Aerodynamic optimization design problems of industrial interest require calculating performance metrics (such as lift, drag, and pitching moment) of many candidate designs that are usually very computationally expensive. The introduction of uncertain parameters further increases these costs since they require estimating a statistical measure computed on a population of candidates. This increment in required evaluations may easily make the problem unsolvable. Surrogate models, also called response surfaces, may substantially reduce the computational budget and make the robust design approach viable.

The basics steps to perform surrogate assisted optimization are explained in the following steps and depicted in Fig. 5.1.1.

1. Design of Experiments (DOE) or sampling plan: An initial sample of designs is analyzed. There are several techniques for making a smart selection of the points.

2. Surrogate model construction. A wide variety of response surfaces exists, the most suitable for each optimization must be selected. This dissertation only considers the use of Gaussian Processes (GP).

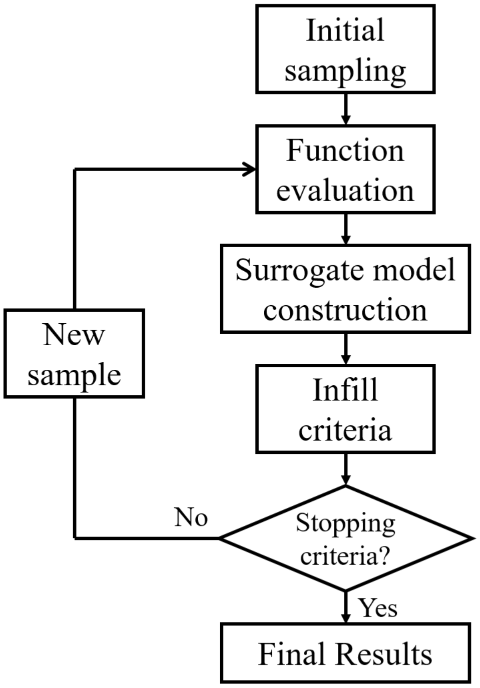


Figure 5.1.1: Surrogate-based optimization framework.

3. Infill criteria: acquisition strategy to determine where to sample next if the computational budget is not exceeded or some convergence criteria is achieved. Then, the new point is evaluated.

The second and third steps iterated until the stopping criterion (computational budget or convergence criteria) is satisfied.

The following sections report a brief introduction to each of theses steps. The illustrative examples here shown were obtained using the German Aerospace Center's (DLR) SMARTy toolbox (Surrogate Modeling for Aero Data Toolbox in Python). This toolbox has been widely used for solving aerodynamic design optimization [97, 98]. The examples are performed in test functions with one and two design variables.

For the 1-dimensional test case, the function is given by:

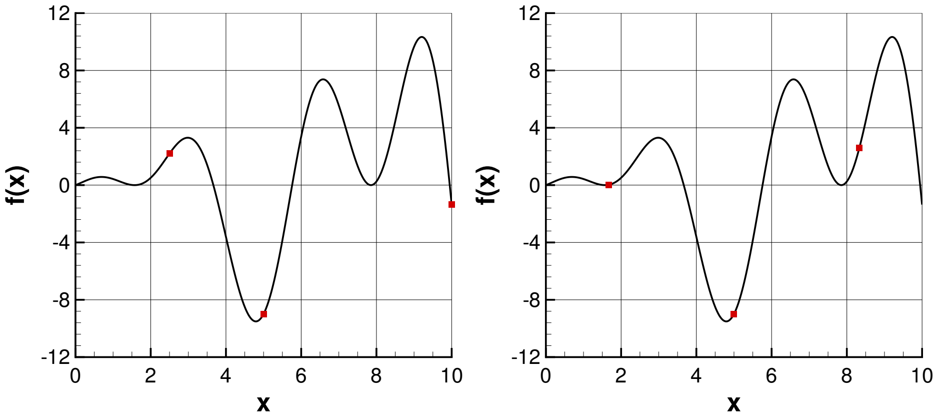
$$f(x) = x\sin(x) + x\cos(2x). \tag{5.1}$$

Moreover, to illustrate the 2-dimensional test case, Himmelblau's equation is used:

$$f(x_1, x_2) = \left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2 \tag{5.2}$$

5.1.1 Design of Experiments (DOE)

Once the problem design space and the design variables are identified, the initial designs to be evaluated are chosen. This process is referred to as the design of experiments or sampling plan. A smart selection of this initial sampling must be made. Intuitively, a uniform, non-regular spread of points across the search space is preferred. With this aim, several approaches have been developed, such as the Sobol sequences [99] or the Latin Hypercube Sampling (LHS) [100] techniques. In the following figures, the initial sampling on the test functions given in Eq. (5.1) and Eq. (5.2) is shown using Sobol sequences and LHS.



(a) Sobol sequence.

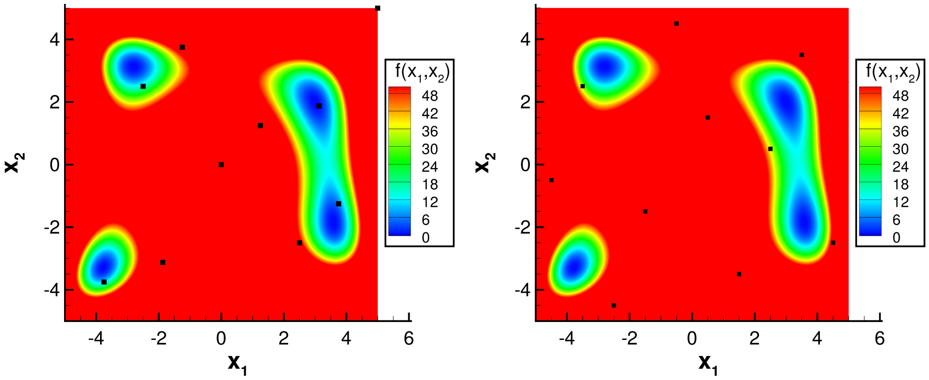
(b) Latin Hypercube Sampling.

Figure 5.1.2: 1D test function (—) (Equation (5.1)) and sampling points (🟥).

The initial sampling for the 1-dimensional test case is made by three points, whereas in the 2-dimensional case, ten initial samples were selected.

5.1.2 Surrogate model construction - Gaussian Process

Nowadays, there is a wide variety of surrogate models. Radial Basis Functions (RBFs) [101, 102] or Support Vector Regression (SVR) [103, 104] are just two of these possibil-



(a) Sobol sequence.

(b) Latin Hypercube Sampling.

Figure 5.1.3: 2D test function contour plot (Equation (5.2)) and sampling points (■).

ities. However, in this dissertation only Gaussian Processes (GPs) are studied. For an in-depth discussion on GPs, please refer to the book of Rasmussen and Williams [105]. Nevertheless, the theoretical principles of Gaussian Processes are explained here [106, 107].

Let us consider a set of noisy data, $\mathcal{D}$, which is composed of N pairs of L-dimensional input vectors and scalar outputs $\left(\{\boldsymbol{x}_n, q_n\}_1^N\right)$. The objective of the Gaussian process is to provide a prediction of the output q_{N+1} at a new point $(\boldsymbol{x}_{N+1})$ by employing the current available data. Besides, a prior to model the data $\mathcal{P}\,(f|\xi)$ and another to model the noise $\mathcal{P}\,(\upsilon|\phi)$ can be defined. Being ξ and ϕ two sets of hyperparameters, and υ a noise vector. Hence, the probability of the data is defined by Eq. (5.3).

$$\mathcal{P}\,(\boldsymbol{q}_N|\{\boldsymbol{x}_n\},\xi,\phi) = \int df d\upsilon\, \mathcal{P}\,(\boldsymbol{q}_N|\{\boldsymbol{x}_n\},f,\upsilon)\,\mathcal{P}\,(f|\xi)\,\mathcal{P}\,(\upsilon|\phi) \qquad (5.3)$$

Being, $\boldsymbol{q}_N = (q_1,\ldots,q_N)$. Moreover, by defining a vector $\boldsymbol{q}_{N+1} = (q_1,\ldots,q_N,q_{N+1})$, constructed with the vector $\boldsymbol{q}_N$ and the prediction at $\boldsymbol{x}_{N+1}$ (q_{N+1}), the conditional distribution of q_{N+1} can be determined.

$$\mathcal{P}\,(q_{N+1}|\mathcal{D},\xi,\phi) = \frac{\mathcal{P}\,(\boldsymbol{q}_{N+1}|\{\boldsymbol{x}_n\},\xi,\phi)}{\mathcal{P}\,(\boldsymbol{q}_N|\{\boldsymbol{x}_n\},\xi,\phi)} \qquad (5.4)$$

Usually, the resolution of Eq. (5.3) is a difficult task. Hence a method based on Gaussian Process priors was considered [107].

A Gaussian Process can be defined as a sample of normally distributed random variables $\boldsymbol{q} = (q(\boldsymbol{x}_1), \ldots, q(\boldsymbol{x}_n))$ such that the joint distribution is a multivariate Gaussian. The joint distribution is defined as:

$$\mathcal{P}\left(\boldsymbol{q}|\boldsymbol{K}, \{\boldsymbol{x}_n\}\right) = \frac{1}{Z} \exp\left(-\frac{1}{2}\left(\boldsymbol{q} - \mu\right)^T \boldsymbol{K}^{-1}\left(\boldsymbol{q} - \mu\right)\right). \tag{5.5}$$

With Z a normalizing constant. The Gaussian Process is specified by a mean vector (μ) and a covariance matrix, or kernel, $(\boldsymbol{K}(\boldsymbol{x}_m, \boldsymbol{x}_n; \boldsymbol{\theta}))$, where $\boldsymbol{\theta}$ is a vector of parameters that can be learned from data to obtain regression. The covariance matrix determines the variance along each dimension (diagonal terms) and shows how the random variables are correlated between them (off-diagonal terms). Moreover, it is positive semi-definite.

Now, let us define the vector of data $\boldsymbol{q}_N$ defined by its covariance matrix $\boldsymbol{K}_N$, and a zero mean vector $(\mu = \boldsymbol{0})$. That is zero-mean prior Gaussian distribution. The conditional Gaussian distribution of $\boldsymbol{q}_{N+1}$ can be obtained using Eq. (5.4).

$$\begin{aligned}\mathcal{P}\left(q_{N+1}|\mathcal{D}, K\left(\boldsymbol{x}_m, \boldsymbol{x}_n; \boldsymbol{\theta}\right), \boldsymbol{x}_{N+1}, \boldsymbol{\theta}\right) &= \frac{\mathcal{P}\left(\boldsymbol{q}_{N+1}|K\left(\boldsymbol{x}_m, \boldsymbol{x}_n; \boldsymbol{\theta}\right), \boldsymbol{\theta}, \boldsymbol{x}_{N+1}, \{\boldsymbol{x}_n\}\right)}{\mathcal{P}\left(\boldsymbol{q}_{N+1}|K\left(\boldsymbol{x}_m, \boldsymbol{x}_n; \boldsymbol{\theta}\right), \boldsymbol{\theta}, \{\boldsymbol{x}_n\}\right)} \\ &= \frac{Z_N}{Z_{N+1}} \exp\left(-\frac{1}{2}\left(\boldsymbol{q}_{N+1}^T \boldsymbol{K}_{N+1}^{-1} \boldsymbol{q}_{N+1} - \boldsymbol{q}_N^T \boldsymbol{K}_N^{-1} \boldsymbol{q}_N\right)\right)\end{aligned} \tag{5.6}$$

Defining Eq. (5.7) and Eq. (5.8),

$$\hat{q}_{N+1} = \boldsymbol{k}_{N+1}^T \boldsymbol{K}_N^{-1} \boldsymbol{q}_N \tag{5.7}$$

$$\sigma_{i_{N+1}}^2 = \kappa - \boldsymbol{k}_{N+1}^T \boldsymbol{K}_N^{-1} \boldsymbol{k}_{N+1} \tag{5.8}$$

and being the vector $\boldsymbol{k}_{N+1}^T = \left(K\left(\boldsymbol{x}_1, \boldsymbol{x}_{N+1}; \boldsymbol{\theta}\right), \cdots, K\left(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1}; \boldsymbol{\theta}\right)\right)$ and the scalar $\kappa = K\left(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1}; \boldsymbol{\theta}\right)$, thus, Eq. (5.6) can be rewrite as:

$$\mathcal{P}\left(q_{N+1}|\mathcal{D}, K\left(\boldsymbol{x}_m, \boldsymbol{x}_n; \boldsymbol{\theta}\right), \boldsymbol{x}_{N+1}, \boldsymbol{\theta}\right) = \frac{1}{Z} \exp\left(-\frac{\left(q_{N+1} - \hat{q}_{N+1}\right)^2}{2\sigma_{i_{N+1}}^2}\right) \tag{5.9}$$

Note that, $\hat{q}_{N+1}$ is the predictive mean at the new point and $\sigma^2_{i_{N+1}}$ the variance on this prediction.

Finally, the selected covariance matrix must be specified. In particular, the chosen form for the covariance matrix is given in Eq. (5.10).

$$K\left(\boldsymbol{x}_m, \boldsymbol{x}_n; \boldsymbol{\theta}\right) = \nu_1 \exp\left\{-\frac{1}{2}\sum_{l=1}^{L}\frac{\left(x_m^{(l)} - x_n^{(l)}\right)^2}{r_l^2}\right\} + \nu_2 + \delta_{mn}\mathcal{N}\left(\boldsymbol{x}_m; \boldsymbol{\theta}\right) \quad (5.10)$$

with $x^{(\ell)}$ the ℓ-th component of vector $\boldsymbol{x}$. The vector of hyperparameters is given by $\boldsymbol{\theta} = \{\nu_1, \nu_2, r_1, \ldots, r_L\}$ and N defines the noise model.

Figures 5.1.4 and 5.1.5 show the first approximation of the 1 and 2-dimensional test functions (Eqs. (5.1) and (5.2)) using the Sobol sequence approach to select the initial sampling as depicted in Figs. 5.1.2 and 5.1.3a.

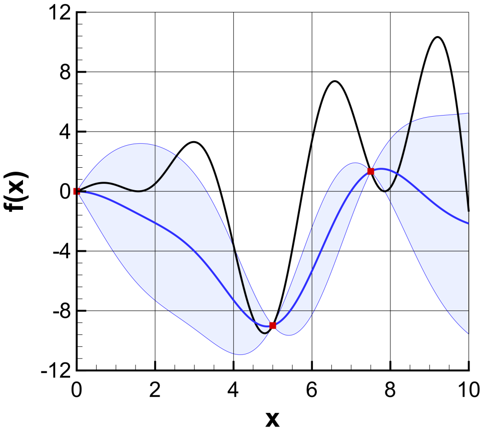


Figure 5.1.4: First 1D test function approximation (Eq. (5.1)) (—), the true function (—), and the sampling points (■). The filled space is the area between the predicted values plus and minus the standard deviation ($\hat{q} \pm \sigma$).

5.1.3 Infill criteria

The evaluation of the true function must always confirm the results provided by the surrogate. As mentioned, the true function evaluation is, usually, costly (such as CFD analyses).

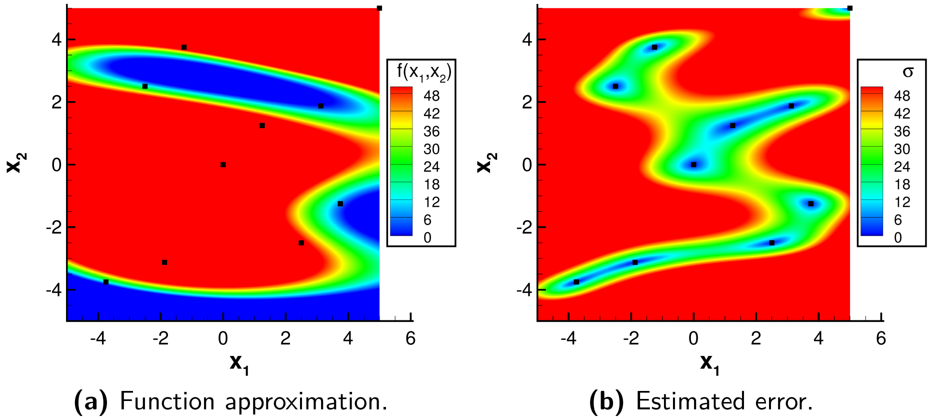
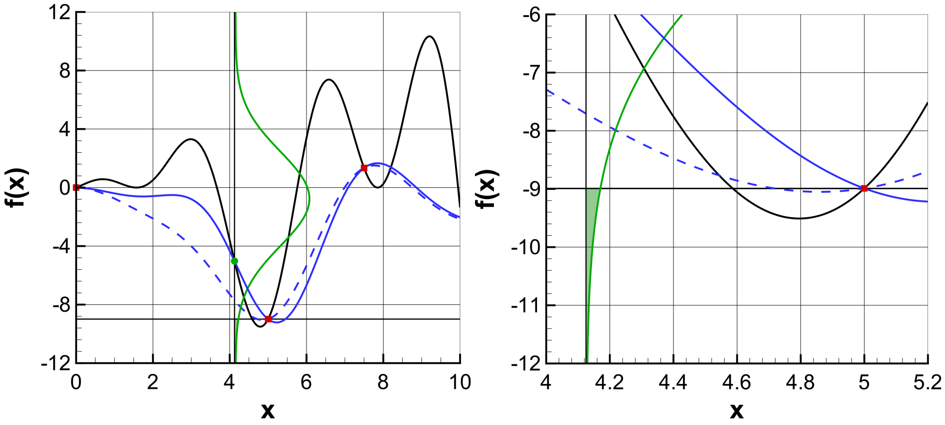


Figure 5.1.5: First 2D test function approximation (Eq. (5.2)) and the estimated error contour plots and sampling points (■).

Therefore, a judicious selection of the new point (*infill point*) in which to evaluate the true function is needed. The infill point is chosen based on some infill criteria. According to this technique, known as *adaptive sampling*, the promising areas of the objective function are sampled on a continually changing surrogate. Different types of infill criteria there exist depending on the aim of the search in the design space. The infill criteria are divided into two main categories: exploration and exploitation.

Design space *exploration* means filling the gaps between the existing sampled points. Infill criteria based on exploration have a role in producing a globally accurate model, providing the visualization and comprehension of the design space. However, this technique could waste time modeling sub-optimal regions when the global optimum is required for optimization purposes. Moreover, infill criteria based on the design space *exploitation* are appealing methods only to perform local optimization. Thus, there must be a balance between exploration and exploitation of the design space. Firstly, an exploration of the design space is performed, followed by exploiting of the most promising area to find the optimum.

One technique that balances exploration and exploitation is the *Expected Improvement* (EI) introduced by Jones et al. [108]. Before evaluating the true function at a point of the design space $\boldsymbol{x}$, its value $\hat{q}(\boldsymbol{x})$ is uncertain. The uncertainty of $\hat{q}(\boldsymbol{x})$ can be modeled by treating it as a normally distributed random variable Q. The Gaussian process gives the mean and standard deviation. Let us define f_{min} as the current minimum value. Hence, there will be a probability that the value of the function improves the current best (f_{min}).



(a) Full area. **(b)** Zoomed area.

Figure 5.1.6: Graphical interpretation of the probability of improvement. The 1D test function (given in Eq. (5.1)) (—), initial sampling points (■), infill point (●), first (- - -) second (—) function approximation, and normal density function (—). The green filled area is the probability of improvement.

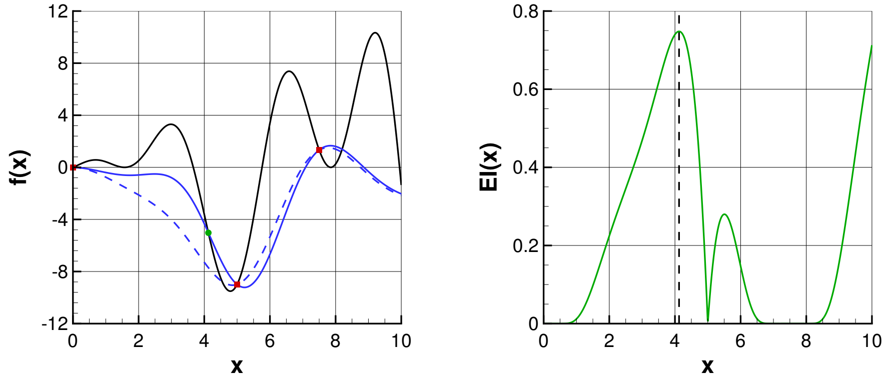
The probability of improvement is depicted by the green area of Fig. 5.1.6. The improvement at a point $\boldsymbol{x}$ of the design space is defined by $I = \max(f_{min} - Q, 0)$. To obtain the expected improvement the expected value of this expression is taken. In other words, the expected improvement can be seen as the first moment of the area enclosed by the normal distribution and the current minimum value (green area in Fig. 5.1.6).

$$\mathrm{E}\left[I(\boldsymbol{x})\right] = \mathrm{E}\left[\max(f_{min} - Q, 0)\right] \tag{5.11}$$

Therefore, the infill point is selected where the expected improvement is maximum. In Fig. 5.1.7b, the EI after the approximation of the 1D test function with the initial sampling points is represented together with the point of maximum expected improvement. Furthermore, Fig. 5.1.7a provides the new function approximation after the evaluation of the infill point using the true function.

Moreover, the expected improvement obtained after the surrogate construction using the initial ten samples of the 2-dimensional test function is shown in Fig. 5.1.8.

The point with maximum expected improvement is placed at the lower boundary of the two design variables ($\boldsymbol{x} = [-4.9994, -4.9987]$). This infill point is now evaluated with



(a) Function approximation

(b) Expected Improvement.

Figure 5.1.7: The 1D test function (given in Eq. (5.1)) (—), initial sampling points (■), infill point (●), second function approximation (—). The green filled area is the probability of improvement

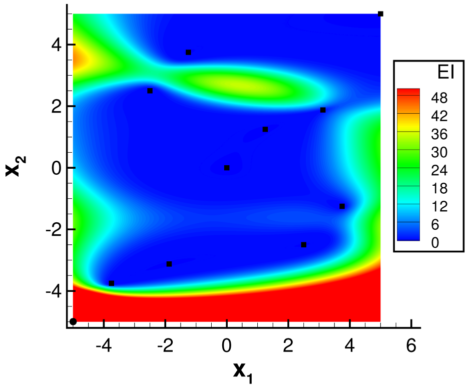
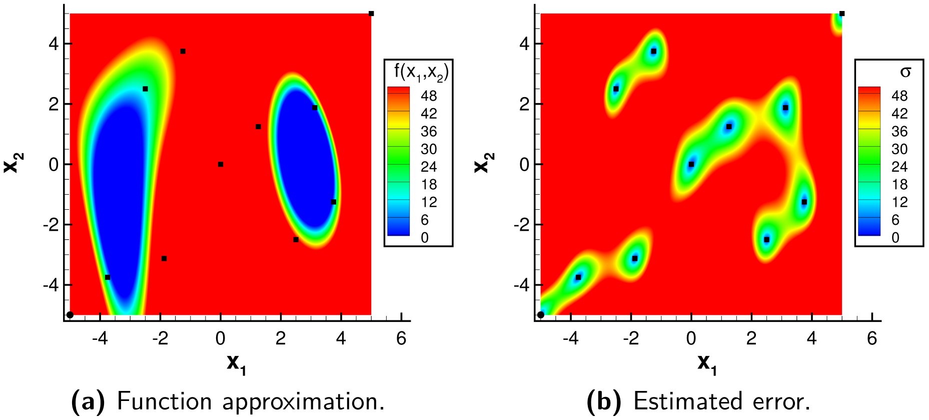


Figure 5.1.8: Expected Improvement contour plot after the GP construction using the initial sampling point (■) and the calculated infill point (●).

the true function and added to the training samples to built up the new surrogate. The new surrogate model for Eq. (5.2) is presented in Fig. 5.1.9a, and the estimated error obtained is given in Fig. 5.1.9b.

By repeating this process of finding the new infill point, which maximizes the expected improvement, and rebuilding the surrogate model, the value of the minimum function can be found. The selection of infill points is repeated until reaching a convergence criterion,



(a) Function approximation. **(b)** Estimated error.

Figure 5.1.9: Second 2D test function approximation (Eq. (5.2)) and the estimated error contour plots. The initial sampling points (■) and the first infill point (●).

like maximum EI below a selected threshold, or the difference between the true value of the function at the new infill point and the current best (f_{min}) is lower than a tolerance, or the allocated computational budget is reached. In Fig. 5.1.10, the final surrogate model for the 1D test function is presented.

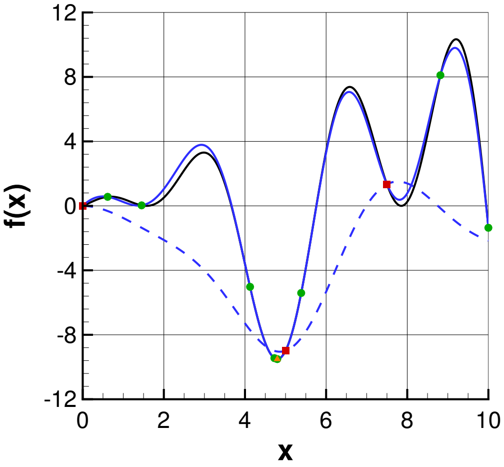


Figure 5.1.10: Final surrogate model (—) of the 1D test function (—) and the initial function approximation (- - -). The initial sampling (■) and the infill points (●). The minimum point (▲).

The 1D test function given in Eq. (5.1) has a global minimum equal to -9.50835 at

$\boldsymbol{x} = [4.79507]$. With the surrogate-based optimization, the minimum is found at $\boldsymbol{x} = [4.7959012]$ and is equal to -9.5083475. Thus, an error of the 0.02% is made in the location of the minimum, which leads to a negligible error in the value of the global minimum.

Sometimes, after a first exploration of the search space based on an Expected Improvement infill criterion, exploitation of the promising areas of the design space is done by using an infill criteria based of the minimum provided by the surrogate model. In other words, the new point to be evaluated with the true function is the predicted optimum by the surrogate. An example of this two-step infill criteria is shown using the 2-dimensional test function.

Equation (5.2) has four global analytical minima at $\boldsymbol{x}^1 = [-3.77931, -3.283186]$, $\boldsymbol{x}^2 = [3, 2]$, $\boldsymbol{x}^3 = [-2.805118, 3.131312]$, and $\boldsymbol{x}^4 = [3.584428, -1.848126]$. The value of the function at these points is equal to 0. A surrogate-based optimization using the two aforementioned infill criteria was performed. The initial sampling is performed using the Sobol sequence technique with 10 samples (see Fig. 5.1.3a). Later, new infill points are chosen using the maximum expected improvement approach until the convergence criterion is achieved. In this case, that the maximum expected improvement is below 0.0001.

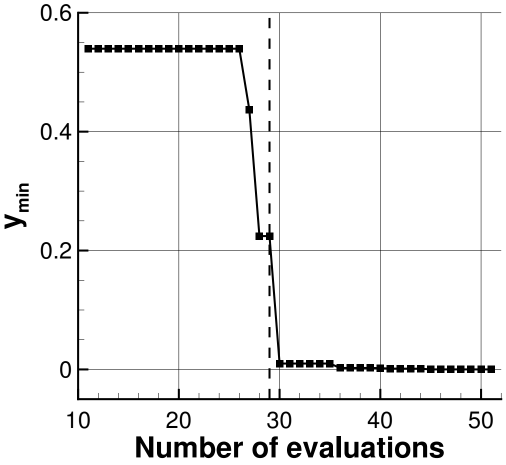
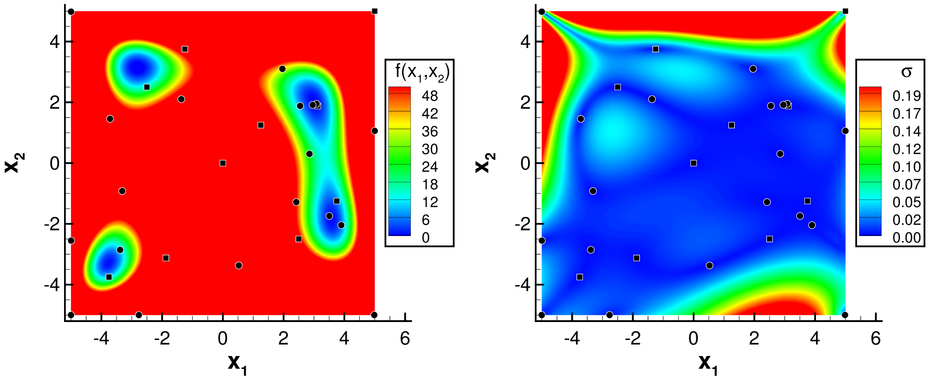


Figure 5.1.11: Current minimum convergence.

When convergence is reached, the infill points are selected at the locations where the surrogate model predicts the lower value of the function. The stopping criteria, in this case, is that the absolute difference between the true value at the new point and the current optimum is lower than a threshold and, at the same time, the absolute difference between the design variables at this point and the design variables of the current best is below a threshold. The convergence of the current minimum value is shown in Fig. 5.1.11. The separation line in Fig. 5.1.11 divides the convergence into two parts: the first one uses the maximum expected improvement, while the second one the minimum predicted value infill criteria. Note that, the number of evaluations starts at eleven since the initial sampling required ten function evaluations. The maximum expected improvement stopping criteria was fulfilled after 19 iterations. The output Gaussian process and the associated error after these iterations are shown in Fig. 5.1.12.



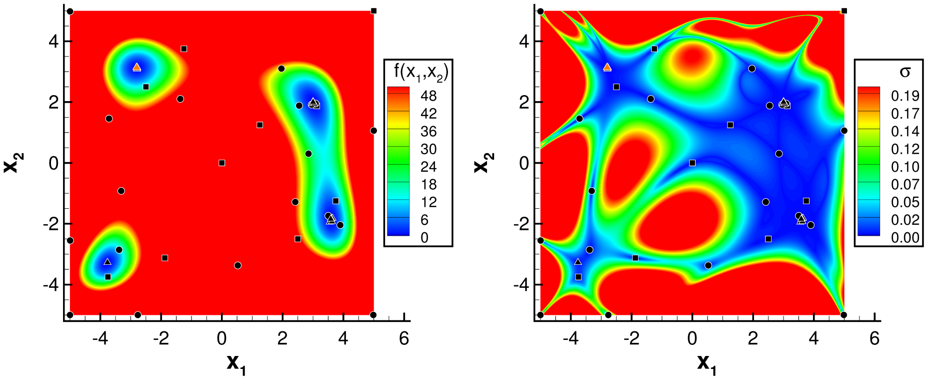
(a) Function approximation.

(b) Estimated error.

Figure 5.1.12: 2D test function approximation (Eq. (5.2)) and the estimated error contour plots after the maximum EI infill criteria. The initial sampling points (■) and the infill points using EI criteria (●).

Then, additional 22 new points, obtained with the second infill criteria, are evaluated until reaching convergence. In Fig. 5.1.13, the final surrogate model and its estimated error are provided.

In Figs. 5.1.12 and 5.1.13, it is seen how the infill points provided by the maximum EI criteria aim the exploration of the design, and a global approximation of the function is obtained. After that, the infill criteria based on the minimum value predicted by the Gaussian process exploit the promising areas of the design space. The black circles (maximum EI infill points) are placed along with the entire design space, whereas the black triangles



(a) Function approximation.

(b) Estimated error.

Figure 5.1.13: Final 2D test function approximation (Eq. (5.2)) and the estimated error contour plots. The initial sampling points (■), the infill points using EI criteria (●), and the infill points using the minimum prediction criteria (▲). The optimum point (▲).

(minimum prediction infill points) are only located at the areas where the global minimums are located. Moreover, it must be mentioned that by locating more points on the optimal regions, a better approximation of these areas is obtained at the expense of having a more significant error estimation elsewhere. This is observed by comparing Figs. 5.1.12b and 5.1.13b.

Finally, it must be mentioned that a trust-region method could also be employed after the two aforementioned infill criteria to prove that the convergence is not to a local optimum.

These steps must be followed when the aim of the surrogate model is to find the optimum in a design search space. This use of the surrogate model is applied in Sections 5.3 and 6.6. Despite this, if a globally accurate surrogate model is needed, an infill criterion based on the estimated error given by the Gaussian process has to be used. New points are evaluated where the estimated error is maximum until a convergence criterion is reached. This approach is shown herein for the 1-dimensional test function.

In Fig. 5.1.4, it is observed that after the initial sampling, the estimated error of the surrogate is at $x = [10.0]$. Thus, it is the first infill point. The resulting function approximation and the estimated error after the addition of the first infill point is given in Fig. 5.1.14. New points are, successively, selected until the maximum estimated error is below a threshold.

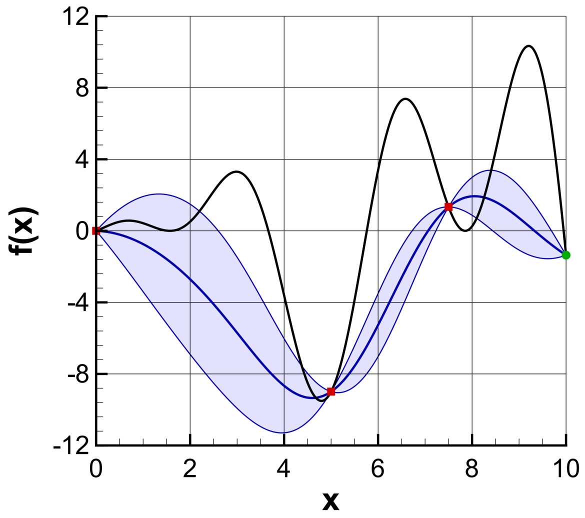


Figure 5.1.14: 1D test function approximation (—) after the first infill point given by the maximum error criteria. True function (—), sampling points (■), and the first infill point (●). The filled space is the area between the predicted values plus and minus the standard deviation ($\hat{q} \pm \sigma$).

In this case, it is equal to 0.3. The final approximation is shown in Fig. 5.1.15. The difference between the approximated function and the true function is negligible by evaluating only 13 times the true function.

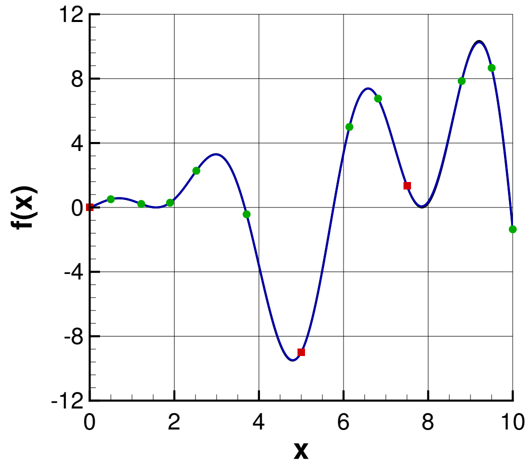


Figure 5.1.15: 1D test function final approximation (—). True function (—), sampling points (■), and the infill points (●).

A globally accurate Gaussian process is needed in Section 5.2. A particular optimization process is carried on (see Section 5.2.2) to select the sampling points to build the GP.

5.2 Gaussian processes for CVaR approximation in robust aerodynamic shape design

One of the latest studies presented at the International Conference on Uncertainty Quantification & Optimization (UQOP2020) is introduced in this section. In particular, the potential of Gaussian processes to reduce the computational load of robust optimization problems faced with the risk function methodology is investigated. The focus is on the Conditional Value-at-Risk function (CVaR), already presented in Section 1.2, and shows how a reduced number of samples can be used to obtain an approximation of CVaR usable in a robust optimization loop based on evolutionary algorithms.

As already mentioned, the disadvantage in the use of risk functions is all in their nature as statistical estimators, in the sense that to obtain a sufficiently significant estimate of the risk function, it is often necessary to carry out large samplings of the related quantity of interest (QoI). Therefore, if the QoI requires significant computational effort to evaluate it (such as CFD evaluations), the total computational cost of robust optimization can quickly become unsustainable. Consequently, limiting the number of samples necessary to obtain a reasonable estimation of risk functions is a research topic of paramount importance. Various approaches are possible for this purpose ranging from the introduction of sophisticated sampling strategies such as multilevel Monte Carlo [109, 110], or the use of advanced quadrature schemes [111, 112], to the use of the adjoint solution as presented in Chapter 4.

Thus, the characterizing elements of this study compared to the current literature are that the approximation obtained with the Gaussian Process is applied to evaluate the CVaR function which gives greater importance to the tails of the probability distribution compared to the first two statistical standard moments, and that the approximation with the GPs aims at making the optimization process more efficient and quick. Therefore, the number of QoI evaluations is minimized with very light training. An update strategy of the GP training point is developed, hence the Gaussian Process is refined during the optimization process.

The application used to illustrate the potential of the method is the optimization of the aerodynamic shape of an airfoil in the transonic field. The QoI is the drag coefficient, c_d, that is calculated by solving the RANS equations.

5.2.1 Robust design and CVaR risk function

In robust aerodynamic optimization problems, the Quantity of Interest (QoI) is a statistical measure. As beforehand mentioned, the advantage of these types of problems is the offer of an optimal design less vulnerable to different uncertainty sources. Here, an optimization approach based on risk measures is employed. The risk measure, $\rho(X)$, depends on random variables X that represent the uncertainty of the problem. At the same time, these random variables can also depend on the design parameters, $\mathbf{x}$. Thus, the optimization problem can be written as in Eq. (1.8).

An approach based on the CVaR function is used here. The CVaR gives greater importance to the probability distribution tails compared to the first two statistical standard moments (μ and σ^2). The approaches based on mean and variance penalize any configuration far from the mean value. For example, consider that the drag coefficient (c_d) has to be minimized. Hence, these measures penalize, in the same manner, configurations that provide an increase in drag and configurations that decrease it. Instead, it is worth the penalization of only configurations leading to an increase in (c_d) since lower drag configurations are desirable. Therefore, risk measures that work asymmetrically, likewise CVaR, are preferred. The Conditional Value-at-Risk is mathematically described in Eq. (5.12) (a detailed description is given in Section 1.2).

$$c^{\gamma} = \frac{1}{1-\gamma}\int_{\gamma}^{1} \nu^{\beta} d\beta \tag{5.12}$$

Figure 5.2.1 shows how CVaR relates to the distribution function. The two filled areas, respectively in solid colour and squares, are proportional to the CVaR calculated for the distribution that represents the robust optimum and for that related to the deterministic optimum. The robust optimum, characterized by a more flattened distribution, is also the one with the lower CVaR.

5.2.2 Risk function approximation

The work aims to introduce surrogate models to reduce the computational load associated with the evaluation of the Empirical Cumulative Distribution Function (ECDF) needed

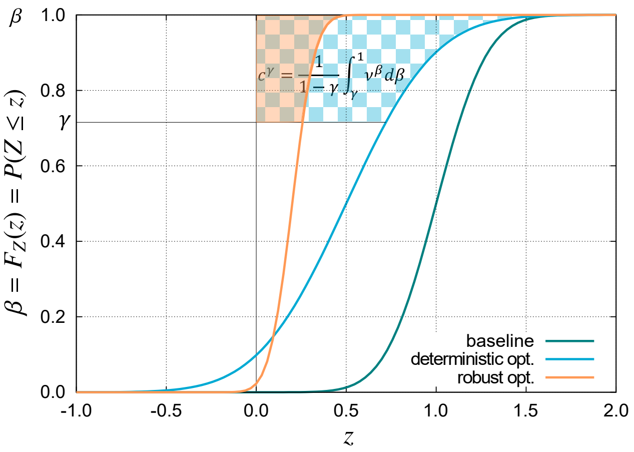


Figure 5.2.1: Cumulative Distribution Function and related CVaR Risk Function at a given γ threshold value (confidence level).

to calculate the CVaR. However, there are some considerations regarding accuracy and precision obtainable in estimating the parameters of a population.

Before going into detail on the use of Gaussian Processes for approximating of risk functions, it is necessary to reflect on what it means to use approximation techniques for the calculation of statistical estimators. When a statistical parameter is evaluated, the main problem is accuracy, intended as the proximity of the estimator obtained through a sample to the real population parameter.

Accuracy depends on the number of samples available to calculate the statistics. To calculate it, however, it is necessary to know the actual value of the population parameter. Realistically, a value obtained with a huge number of samples is considered as the true parameter value. So, it should proceed according to the following steps:

1. Compute the quantity of interest as many times as possible as allowed by the computational budget.

2. Consider the obtained estimate as being the true value of the parameter.

3. Consider a smaller population (realistically the smaller that one can afford in the experiments) and do a high number of times (with different samples) the calculation of the estimator (one could also randomly extract subsamples from the large initial population to avoid doing other calculations).

4. If the standard deviation or variance of the estimates obtained is calculated in this

way, an estimate of the precision of the method is found, but not the accuracy. If instead, the average value of the estimates is replaced with the true value, an estimate of the accuracy is obtained.

Surrogate models allow high precision in estimating a statistical quantity because the evaluation of a huge sample is computationally very cheap. On the other hand, they may also introduce a non easily estimable bias error. This bias can be determined when exact values are available for a large population sample. This study aims to reduce as much as possible the bias error due to surrogate models using a limited number of exact samples. Indeed, it is essential not to exceed the number of samples needed to train the approximator (in our case, the GPs) not to frustrate the advantages that surrogates offer in computational cost.

Therefore, a GP-based regression method is used to approximate the ECDF [27] for the QoI (the c_d in this benchmark case). After approximating the empirical cumulative distribution function with the Gaussian Process, the statistical of interest, the CVaR, is calculated. The particular Gaussian Process used in this study is explained in Section 5.1.

Training methodology

The training procedure followed to instruct the Gaussian processes to approximate risk functions is the characterizing point of this work. The goal is to obtain a convenient compromise between the effort required for training and the Gaussian process predictive ability. The starting point is a set of ECDF calculated with a sufficiently large number of samples of the quantity of interest q. In particular, q is a function of a vector $\boldsymbol{x} \in \boldsymbol{X}$ of deterministic parameters (design parameters) and a vector of random variables $\boldsymbol{u} \in \boldsymbol{U}$ that model uncertainty:

$$q_{ki} = q\left(\boldsymbol{x}_k, \boldsymbol{u}_i\right) \tag{5.13}$$

For this study, it is chosen and fixed once and for all a congruous number of samples $\boldsymbol{u}_i$ of the random vector according to the assigned joint distribution function, and this set is always used when the distribution function of q is calculated for an assigned deterministic input vector $\boldsymbol{x}_k$. So, having fixed an assigned design vector $\boldsymbol{x}_k$, the set of $Q_k \triangleq \{q(\boldsymbol{x}_k, \boldsymbol{u}_i),\ \ i = 1 \ldots n\}$, that is obtained with the random sampling of $\boldsymbol{u}$, is computed. This set of values constitutes the reference ECDF. The goal is to build a Gaussian process capable of accurately approximating both the q values of the training set and those not belonging to it. The fundamental difference compared to the classic approach is that instead of approximating the whole set of data globally, the focus is on a single ECDF. In this way, a tiny and predetermined sample of the input CDF is evaluated, say $\boldsymbol{U}_\sigma \subset \boldsymbol{U}$, and this subset is the training set. Then, the Gaussian process that approximates it $\hat{q}_k = q_k^{\mathrm{GP}}(\boldsymbol{u}; \boldsymbol{x}_k, \boldsymbol{U}_\sigma)$ is built, and the quality of the approximation depends on the training set $\boldsymbol{U}_\sigma$, the choice of which is a combinatorial optimization problem. Defining with

$$\Delta(Q_k, \hat{q}_k) \tag{5.14}$$

a function that evaluates the distance between the complete ECDF and that obtained by the Gaussian process using the $\boldsymbol{U}_\sigma$ training subset, for a given set size $m < n$, the following minimization is obtained:

$$\min_{\boldsymbol{U}_\sigma \subset \boldsymbol{U}} \Delta(Q_k, \hat{q}_k) \tag{5.15}$$

with $\boldsymbol{U}_\sigma \triangleq \{\boldsymbol{u}_{\sigma_1}, \dots, \boldsymbol{u}_{\sigma_m}\}$ and $\sigma_i \in \{1, \dots, n\}$ without repetitions. So it is a question of finding the σ_i indices each in the range $(1 \dots N)$ (without repetitions) that minimize the distance function Δ. In this work, a genetic algorithm (see Section 2.1.1) is used with an appropriate encoding of the bit string. However, the solution obtained has the disadvantage of not being general. That is, for an arbitrary ECDF $Q_p \neq Q_k$, the approximation obtained is not necessarily satisfactory. For this, it is necessary to expand the verification set by adding more ECDFs. In this case, the objective function becomes:

$$\min_{\boldsymbol{U}_\sigma \subset \boldsymbol{U}} \sum_{k=1}^{T} \Delta(Q_k, \hat{q}_\star) \tag{5.16}$$

The critical point is that the set of random input vectors $\boldsymbol{u}$ must be the same for each deterministic vector $\boldsymbol{x}_k$. In this way, the optimal subset of $\boldsymbol{U}_{\sigma^\star}$ points corresponds to different values of q because it also depends on the deterministic vector, and it will be the optimal choice for the approximation of the whole ECDF set. Therefore, once obtained the optimal $\boldsymbol{U}_{\sigma^\star}$ subset, the computation of a new ECDF approximation requires the exact calculation of q only for the values of this subset. So, excluding the GP training time, the computational cost of the new ECDF calculation is of the order of $O(m)$. If the cardinality of the $\boldsymbol{U}_\sigma$ training set is increased, the time required to define the next optimal local GP also increases, and the combination of GP training time increment and the $O(m)$ evaluations may render the GP usage no more convenient. To remedy, at least partially, this drawback, an objective function can be defined, which privileges more the approximation of the ECDF in the areas essential for the variation of the risk function considered. In this work, the objective function introduced to this end is the weighted sum of both ECDFs and risk functions differences. The ideal size of the subset of samples to approximate an ECDF must respond to a tradeoff between the accuracy of the GP approximation and the computational cost necessary to evaluate the quantity of interest at the input values of the subset. It is certainly possible to introduce the satisfaction of this tradeoff as a further goal of the training process, but, at present, the size of the GP training subset is decided through empirical considerations and a process of trial and error.

As a final remark, the need to keep the same sample for the input variables governing the uncertainty is certainly a disadvantage of the method since it can introduce a bias in the approximation process. However, this problem is mitigated by the fact that the training samples, although they are all obtained with the same set of input variable samples, are relatively large, and the GP training process allows to select a small set of samples which, at least regarding the original training set, minimizes this bias effect.

5.2.3 Numerical analysis tools

For the aerodynamic design optimization problem faced in this work, the autonomous aerodynamic computational chain is composed of the following blocks. Firstly, it generates the candidate airfoil through the *wg2aer* program that, from some input design variables, modifies a baseline airfoil considering a set of modification functions as Hicks–Henne, Wagner, or Legendre functions. Afterward, an unstructured square mesh is created employing a self-developed procedure based on the open-source grid generator *Gmsh* [47]. Finally, the aerodynamic solver is run. In this case, the open-source Computational Fluid Dynamic (CFD) SU2 [113] solver was used. It solves the compressible Reynolds-averaged Navier-Stokes (RANS) equations using SA turbulence model [75]. Each of these components is explained in Section 2.2.

5.2.4 Design application example

The proposed design application example is thoroughly introduced in Chapter 4 in Sections 4.2 and 4.5. A summarized problem recap is given here-in. In particular, the different steps followed to transform the constrained deterministic problem to an unconstrained robust design exercise are explained.

Specifically, for this study, both aerodynamic and geometric constraints are used. The airfoil percentage thickness with respect to the chord is fixed at the base value ($t_{\%}$), while some constraints on the leading edge radius (LER), the trailing edge angle (TEA) and the airfoil percentage thickness with respect to the chord at $x/c = 0.85$ (TAT) are introduced to obtain realistic shapes. Moreover, special attention is dedicated to the airfoil pitching moment coefficient c_m, which for BWB configurations is a critical parameter due to the absence of the elevators. For this reason, two constraints for the c_m coefficient are imposed to keep its value properly confined, as required by trim aspects. The c_m coefficient is calculated with respect to the aerodynamic center, and it is considered positive in the case of "*nose up*" pitching moment.

The deterministic optimization problem reads:

$$\begin{cases} \quad \min\limits_{\mathbf{x}} & c_d(\mathbf{x}) \\ \text{subject to:} & \\ & t_{\%} = 16.00 \\ & c_l = 0.1 \\ & \text{LER} \geq 0.00781 \\ & \text{TEA} \geq 22.0^\circ \\ & c_m \geq -0.04 \\ & c_m \leq 0.04 \\ & \text{TAT} \geq 0.06658 \\ & \text{error} = 0 \end{cases} \tag{5.17}$$

The penalty approach is used to obtain an unconstrained problem:

$$\min_{\mathbf{x}\in X\subseteq\mathbb{R}^n} c_d(\mathbf{x}) + P(\mathbf{x}) \tag{5.18}$$

with

$$\begin{aligned} P(\mathbf{x}) \quad = \quad & k^1 p^+(\text{LER}, 0.00781) + \\ & k^2 p^+(\text{TEA}, 22.0^\circ) + k^3 p^+(\text{TAT}, 0.06658) + \\ & k^4 p^+(c_m, -0.04) + k^4 p^-(c_m, 0.04) + k^5 p^+(\text{error}, 0) \end{aligned} \tag{5.19}$$

All the constraints except those regarding the lift coefficient and the airfoil percentage thickness with respect to the chord are treated as quadratic penalties:

$$p^+(x,y) = \begin{cases} 0 & \text{if } x \geq y \\ (x-y)^2 & \text{if } x < y \end{cases} \quad \text{and} \quad p^-(x,y) = \begin{cases} (x-y)^2 & \text{if } x \geq y \\ 0 & \text{if } x < y \end{cases} \tag{5.20}$$

The c_l and the thickness constraints are not included because they are automatically satisfied by the computation procedure by changing the angle of attack and re-scaling the airfoil thickness to the assigned value.

The numerical values chosen for the k^i coefficients are: $k^1 = 5000,\ k^2 = 10,\ k^3 = 30,\ k^4 = 1000,\ k^5 = 1000$. The transformation of a constrained optimization problem into an unconstrained one through the penalty approach is always a delicate process, as the choices of the weights of the penalization terms profoundly change the shape and features of the search space.

After solving the deterministic optimization, Section 4.5.2, the robust one must be carried out as in Chapter 4. The objective of the robust optimization is to improve the airfoil performances under uncertainties in the airfoil shape and in the operating conditions.

To account for uncertainty, 12 uniformly distributed random variables were used to represent the stochastically perturbed shape of the airfoil. Moreover, the Mach number and the angle of attack are considered as the uncertain working conditions, and they are modeled as four-parameter beta distribution, whose density function is given by

$$f(y;\eta,\theta) = \frac{\gamma(\eta+\theta)(y)^{\eta-1}(1-y)^{\theta-1}}{\gamma(\eta)\gamma(\theta)} \tag{5.21}$$

with shape factors η, θ, and a scale and translation given by $y = (x - \mathrm{loc})/\mathrm{scale}$. Mach is characterised by $\eta = 2,\ \theta = 2,\ \mathrm{scale} = 0.08,\ \mathrm{loc} = 0.76$, while the angle of attack is characterised by $\eta = 2,\ \theta = 2,\ \mathrm{scale} = 1.0,\ \mathrm{loc} = -0.5$. Table 5.2.1 reports the variation range of these uncertainties.

Uncertainty	Range
Mach, M	$[\ 0.78\ , 0.82\ ]$
Angle of Attack, $\Delta\alpha$	$[-0.15^\circ, 0.15^\circ]$
Geometry, U_j	$[-0.0007\ , 0.0007]\ ,\ j = 1, \ldots, 12$

Table 5.2.1: Summary of uncertain parameter definition in the four benchmark cases.

After introducing random variables, the QoI is a function. Thus, the risk functional CVaR is used to map the chosen QoI into $\mathbb{R}$. It is estimated with a confidence level γ equal to 0.9. Note that, the constraints are computed only at the nominal values because it is interesting to assess the impact of random perturbations only on the drag force.

So, the objective function of the robust optimization problem is:

$$\min_{\mathbf{x}\in X\subseteq\mathbb{R}^n} \mathrm{CVaR}^{0.9}\left(c_d(\mathbf{x})\right) + P(\mathbf{x}) \tag{5.22}$$

5.2.5 Optimization process and robust design results

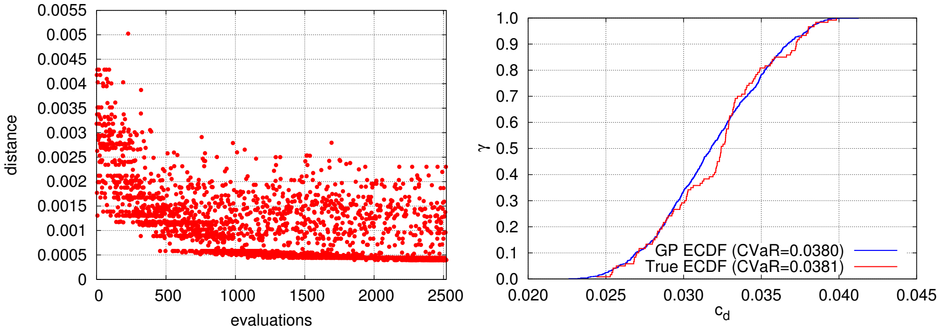
The robust optimization method is based on an adaptive surrogate model (GP) updated in multiple cycles. Following the procedure explained in Section 5.2.2, the solution of an optimization problem determines the training subset for the Gaussian process. Thus, selecting the optimal sets of uncertain variables. In the first step, the selected points are those that minimize the distance between the ECDF of the deterministic airfoil calculated with a sufficiently high number of CFD samples and the ECDF predicted by the Gaussian process. This first optimal set is used to start the initial robust design optimization loop.

Subsequently, the "exact" ECDF is computed for the best solution of this initial robust design optimization, together with the proposed candidates that have maximum Euclidean distance with respect to the baseline airfoil and the up-to-now best solution. These new ECDFs feed another GP training loop to determine the new sets of uncertain variables. Nevertheless, after this step, it was observed that the minimization of the distance between the predicted ECDFs and the "exact" ECDFs was not sufficient to obtain a good approximation of the CVaR measure. Therefore, a new objective function to find the optimal training set is introduced. It considers, both the distance between the ECDFs and the difference between the "exact" and the approximated CVaR value. Moreover, a penalty function is added to the objective function to fulfill an order relationship. Thus, after finding the new GP training subset, a final robust optimization is carried out. Each step, here summarized, is explained now in detail and the most significant results are provided.

Step 1 – First GP training

The initial step of the procedure requires the characterization of the deterministic solution robustness. The deterministic optimization is given in Section 4.5.2. For this purpose, the design variables that describe the uncertainty in the airfoil shape and the operating conditions are sampled using a classical Monte Carlo technique. Subsequently, the ECDF of the quantity of interest, which is the drag coefficient (c_d), is built. In particular, an ECDF with 120 samples is used for the training. Each point calculated to build the ECDF required, on average, 80 minutes of elapsed time to obtain a convergence solution using SU2 V$6.2.0$ on eight cores of a cluster equipped with Intel (R) Xeon (R) cores E5-2670 CPU at 2.60 GHz.

The minimization of the distance between the 120-sample hi-fi ECDF and its approximation via the Gaussian Process was performed using a simple genetic algorithm. The basic concepts of the genetic algorithm used are introduced in Section 2.1.1. Specifically, the algorithm had a population of 120 elements that evolved over 21 generations and used a bit string encoding with Gray code. The crossover operator is the classic one-point binary with a triggering probability of 100%. Bit-mutation has a 2.4% chance of changing the state of a single bit. The string of bits of the genetic algorithm encodes the selection of five elements extracted from the hi-fi ECDF. The Gaussian process constructs a response surface using these five elements and generates an approximated ECDF with a thousand Monte Carlo samples. The objective function is the distance of this approximated ECDF from the original one. The objective function evaluation process is very fast because the setup of a Gaussian process with five elements is rapid. Also, the sampling of the approximating ECDF with a thousand samples is almost immediate.



(a) Evolution history.

(b) GP approximation ECDF training set matching.

Figure 5.2.2: GP preliminary training — 5 variables — one design point.

The history of the evolution process is shown in Fig. 5.2.2a. The visual comparison between the 120-samples ECDF (on which the Gaussian process was trained) and the ECDF approximated with the Gaussian process using the five samples selected by the optimizer is depicted in Fig. 5.2.2b.

Step 2 – First robust optimization run

The first robust optimization step used the resulting GP approximator. Each approximation of the ECDF requires five evaluations of the QoI with five fluid dynamics solver runs. Of these five runs, the first is related to the nominal conditions (without random perturbations) and is at constant c_l. With the settings used, the constant c_l run lasts about twice as long as those with the assigned angle of attack. Although the GP setup does not require the nominal point to be present in the training subset, this point must be calculated to evaluate the initial angle of attack on which to apply the perturbations due to uncertainty $(\Delta\alpha)$. Consequently, it is useful for the computational efficiency of the whole process to force its inclusion in the GP setup subset.

Twenty design variables describe the airfoil shape, and CMA-ES is the algorithm chosen for robust optimization. The CMA-ES algorithm is described in Section 2.1.2. The algorithm parameters for this optimization phase are reported in Table 5.2.2.

Maximum evaluations	Population size	Initial standard deviation
129	8	0.02

Table 5.2.2: CMA-ES parameters adopted for the first robust design optimization step.

The evolution history of this first robust optimization step is reported in Fig. 5.2.3.

The GP approximation quality for elements that are not part of its training set must be verified. For that, three additional elements are evaluated with the pre-defined dense sampling (120 points). In particular, the selected solutions are the one that the GP classifies as best, the population element having maximum Euclidean distance from the baseline in the design variable space and the one with maximum Euclidean distance from the GP best. The last two elements where selected because they represent a search space region not widely explored by the search algorithm. The predicting capabilities of the first Gaussian process are reported in Fig. 5.2.4.

The Gaussian process performance is relatively low and deteriorates quickly when the distance from the baseline increases. Regarding the best candidate predicted by the GP, a

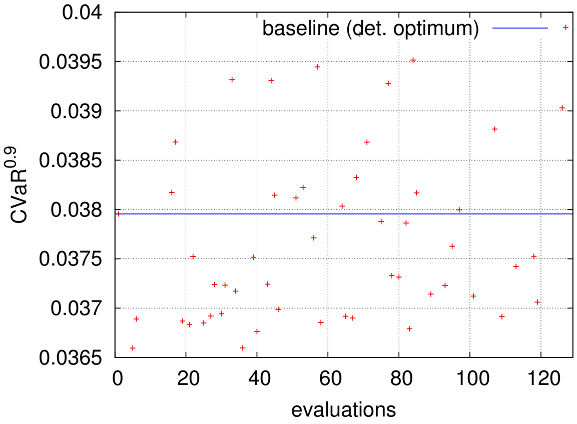
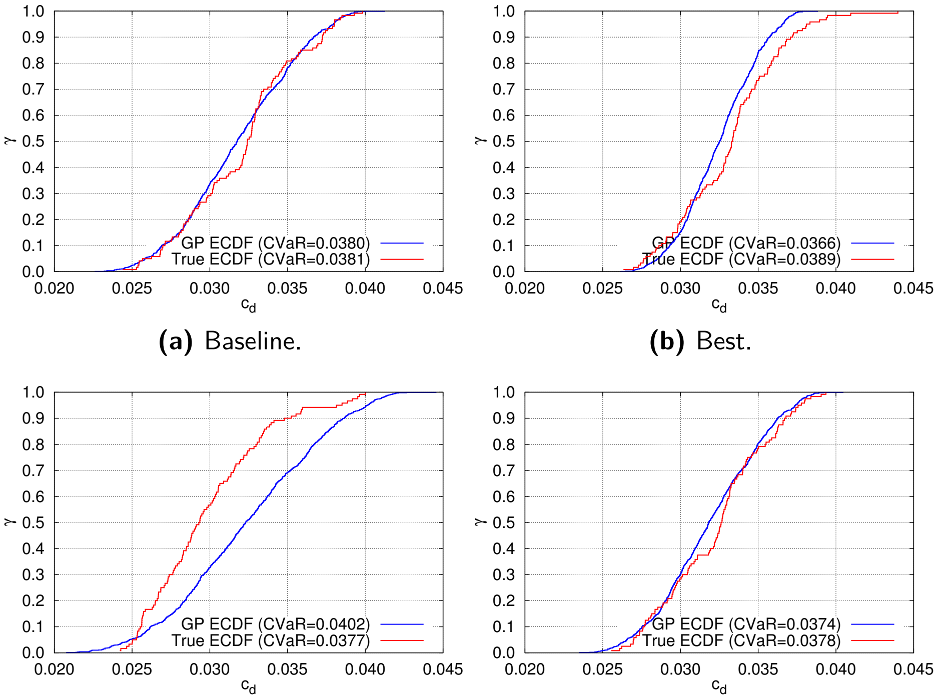


Figure 5.2.3: First optimization run using the initial GP approximation.



(a) Baseline. **(b)** Best.

(c) Maximum euclidean distance baseline. **(d)** Maximum euclidean distance best.

Figure 5.2.4: First optimization run using the initial GP approximation.

good approximation of the lower tail of the ECDF is found but, for the CVaR calculation, the tail of interest is the upper one. Thus, the approximated CVaR value is underestimated. Moreover, the “true” value of CVaR not only provides a higher value than the baseline but it has the worst CVaR in the verification set. Hence, the GP cannot give an accurate prediction both in terms of distance between the ECDFs and of CVaR. After all these considerations, it is evident that a new GP training step is needed.

Step 3 – Second GP training

In this second phase of the Gaussian process training, the goal is to choose a subset of input elements that can guarantee a better approximation for a more significant number of ECDFs. The 3 ECDFs that were not well represented with the old approximation scheme are added to the training pool. The objective function is now constructed by adding the distances between the High-Fidelity ECDFs and the ECDFs constructed with the Gaussian process in the 4 training cases. Therefore the same subset of input vectors of uncertain variables will be associated with the corresponding values of the QoI that correspond to the different training cases considered. Again, the process of evaluating the objective function is very fast since the quantities of interest were previously calculated in the evaluation phase of the High-fidelity ECDFs.

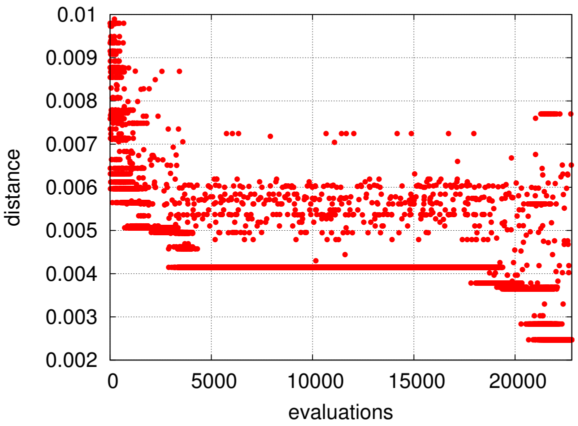
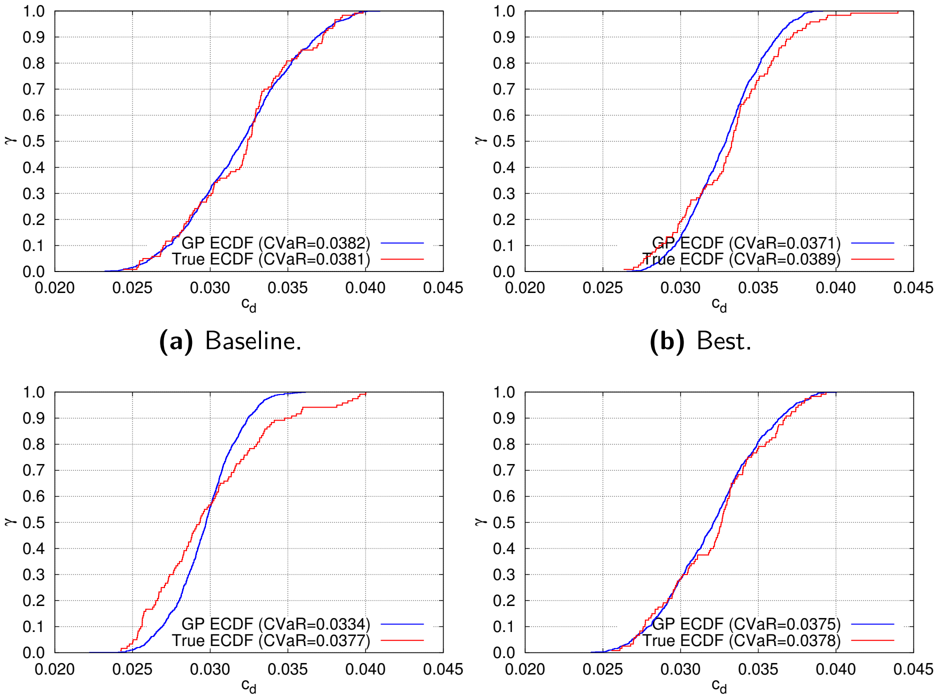


Figure 5.2.5: Evolution history – retraining step 5 variables.

The same genetic algorithm used for the first training phase is used here. However, in this step, the algorithm had a population of 240 elements that evolved over 95 generations. The one-point binary crossover had a triggering probability of 80%, and the bit-mutation had a 1.2% chance of changing the state of a single bit.

Figure 5.2.6 compares, for each training point, the ECDF obtained with fine Monte Carlo sampling and the approximated one with the Gaussian process.



(a) Baseline.

(b) Best.

(c) Maximum euclidean distance baseline.

(d) Maximum euclidean distance best.

Figure 5.2.6: ECDFs comparison on the training set for the second GP training step.

A better overall approximation for the four ECDFs is found. However, the concern about the approximation of the upper tail of the ECDF, the most important part for a good CVaR prediction, is still present, as observed in Figs. 5.2.6b and 5.2.6c. Although the underestimation of the Conditional Value at Risk is higher for the ECDF in Fig. 5.2.6c, the situation most problematic from an optimization point of view is the one depicted in Fig. 5.2.6b. The "true" CVaR value is the highest, while the predicted one is low. Thus, the optimizer will prefer these types of candidates when, in reality, they should be discarded.

Step 4 – Third GP training

Two problems arose from the last Gaussian process training step. The first one concerns the fact that, while obtaining a good overall agreement between the empirical distribution calculated with the Gaussian process and the Monte Carlo sampled distribution, the upper tail of the distribution, which is the most critical part for a good estimate of the CVaR, is not always well approximated.

Indeed, this second point is essential when comparing distributions that are very close to each other. In this case, even in the presence of a good approximation, the error, even if small, in the estimate of the CVaR can lead to the overturning of the order relationship. In other words, it could happen that while the Monte Carlo sampling indicates, hypothetically, $\mathrm{CVaR}_1 < \mathrm{CVaR}_2$, the GP approximation indicates $\overline{\mathrm{CVaR}}_1 > \overline{\mathrm{CVaR}}_2$, reversing the order relation.

To remedy the first problem, it was decided to introduce a new term in the objective function proportional to the distance between the CVaR computed with the Monte Carlo samples and the CVaR calculated with the GP. This term is added to the objective function for each of the training distributions. For the second problem, instead, it has been chosen to introduce a penalty in the objective function, which is increased for each violation of the order relation introduced by the Gaussian process compared to that defined by the Monte Carlo sampling.

The calculation of the penalty value is immediate but involves the sorting and comparison operation detailed below. Let us consider, for each element of the training set, the pair $\left(\mathrm{CVaR}, \overline{\mathrm{CVaR}}\right)$ constituted by the values of the cumulative value at risk calculated, respectively, according to the Monte Carlo sampling and according to the approximation based on Gaussian processes. Then reorder the set of pairs so that $\mathrm{CVaR}_i \leq \mathrm{CVaR}_{i-1}$. The penalty value is given by:

$$P_{\mathrm{tset}} = w \sum_{i=2}^{n} \mathbb{1}_C \left(\overline{\mathrm{CVaR}}_i > \overline{\mathrm{CVaR}}_{i-1}\right) \tag{5.23}$$

where n is the size of the training set, w a weight constant, and $\mathbb{1}_C$ is the indicator function of subset C: $\mathbb{1}_C(x) := \{1 \text{ if } x \in C, \quad 0 \text{ if } x \notin C\}$, with C the subset of all the true inequalities among the set of all the possible ones X between the training set $\overline{\mathrm{CVaR}}$ values calculated with the Gaussian process.

A new optimization run is done with the genetic algorithm. In this case, the genetic algorithm had a population of 240 elements again and evolved over 50 generations. The one-point binary crossover had a triggering probability of 80%, and the bit-mutation had a 1.2% chance of changing the state of a single bit, as in the previous training run. The penalty term weight w was set to 0.01. In Fig. 5.2.7, the optimization run evolution history is given.

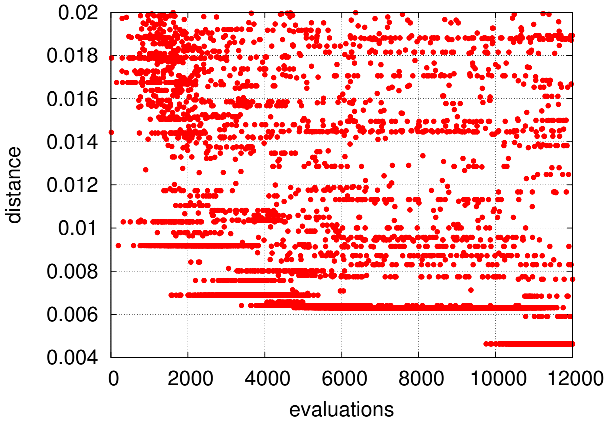


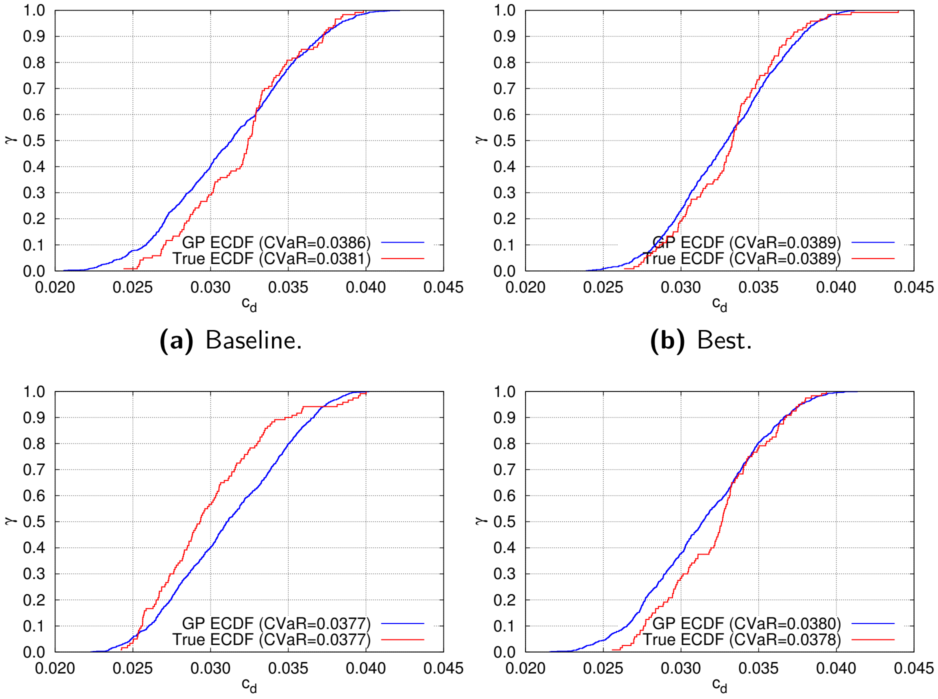
Figure 5.2.7: Evolution history – retraining step 5 variables.

Table 5.2.3 compares, for each training point, the conditional value at risk obtained with fine Monte Carlo sampling (CVaR) and the approximated one with the Gaussian process $(\overline{\mathrm{CVaR}})$. The table shows the CVaR values of the training set sorted in descending order. This then allows evaluating the penalty term associated with the $\overline{\mathrm{CVaR}}$ simply by comparing its value with that of the table previous row.

Solution ID	Distance	CVaR	$\overline{\mathrm{CVaR}}$	penalty term
Best	5.312×10^{-4}	0.03892	0.03886	–
Baseline	9.824×10^{-4}	0.03860	0.03813	0
Max. distance w.r.t. best	8.495×10^{-4}	0.03783	0.03798	0
Max. distance w.r.t. baseline	1.553×10^{-3}	0.03769	0.03766	0

Table 5.2.3: CVaR and $\overline{\mathrm{CVaR}}$ values comparison in the training set after the third Gaussian process training step.

In addition, Fig. 5.2.8 compares the ECDF obtained with the Gaussian process using the new training set and with the fine Monte Carlo sampling. It is observed how, after adding the new terms into the objective function, the upper tail of the empirical cumulative distribution function is better predicted. The new objective function introduced in this training step allows a good calculation of the Conditional Value at Risk, at the cost of a slight deterioration of the overall agreement between the ECDFs.



(a) Baseline.

(b) Best.

(c) Maximum euclidean distance baseline.

(d) Maximum euclidean distance best.

Figure 5.2.8: ECDFs comparison on the training set for the third GP training step.

Step 5 – Second robust optimization run

The robust optimization run is performed using the Gaussian process obtained after the last training described above (Step 4). The starting point of the optimization is the up-to-now studied best configuration. It is not the best obtained in the first robust optimization, but the candidate which variables gave maximum Euclidean distance with respect to the deterministic optimum airfoil variables. The CMA-ES settings are not changed, except the maximum limit of evaluations of the objective function (set at 801). The evolution history of this second robust optimization is given in Fig. 5.2.9.

The quality of the approximation of the GP must be verified. With this aim, the solution that the GP classified as best is evaluated with the pre-defined dense sampling of 120

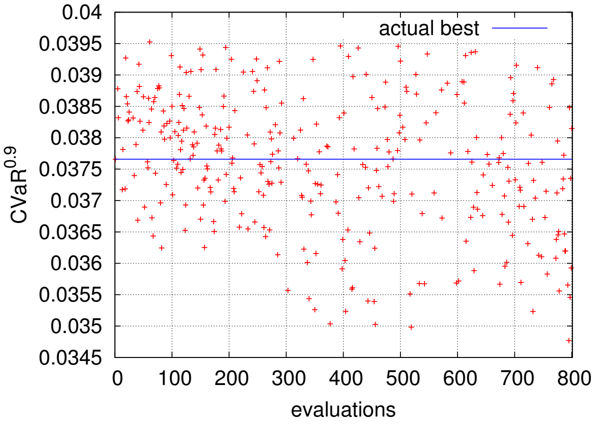


Figure 5.2.9: Second optimization run using the last GP approximation.

points. The "true" and the approximated ECDFs of the best are reported in Fig. 5.2.10. Contrary to what happened at the first robust optimization run, the GP still shows a good capability to predict the empirical cumulative distribution function. Therefore, it can be stated that it is worth including the additional terms in the objective function, even though the training step complexity increases.

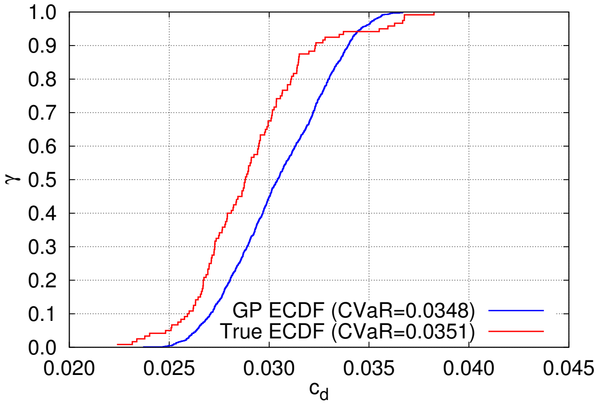


Figure 5.2.10: ECDFs comparison for the best using the third GP training step.

The analysis of the evolutionary history of Fig. 5.2.10 shows that a further improvement of the objective function would be possible, even if an examination of Fig. 5.2.11 suggests that a further phase of GP training is appropriate before restarting the optimization.

Moreover, the results related to the approach based on the first-order approximation (see Section 4.5.6) also confirm that a further reduction of the CVaR is possible. However, a possible increase in the number of samples to be given as input to the GP should be carefully considered to improve the approximation, with a further increase in the computational load of optimization.

On the other hand, it is believed that it is more useful to carry out an in-depth comparison of the robustness characteristics of the solution obtained with respect to the deterministic optimum used as a baseline. The ECDFs of the robust and the deterministic optimum airfoils are shown in Fig. 5.2.11.

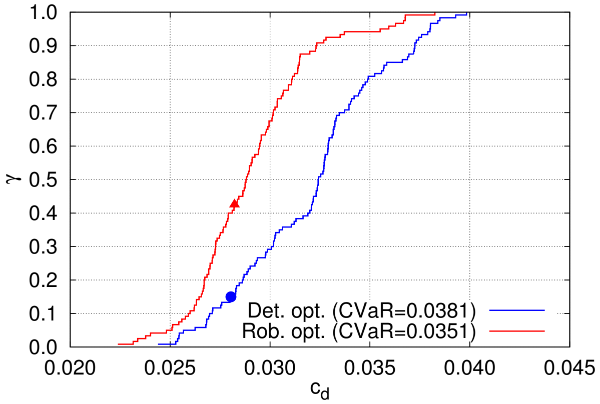


Figure 5.2.11: ECDFs comparison for the deterministic and robust optimum. Nominal conditions $Re_{c_\infty} = 174 \times 10^6$, $M_\infty = 0.8$, and $c_l = 0.1$ and no uncertainty in shape for the deterministic (●) and robust (▲) optimum airfoils.

Figure 5.2.11 shows the improvement of the upper tail of the robust ECDF with respect to the deterministic one. In this case, an improvement of the whole ECDF is found. In addition, in Fig. 5.2.11, the point at nominal conditions for both airfoils is provided. When working in conditions of uncertainty, the airfoil obtained with robust optimization has a 40% probability of obtaining better performance than the nominal one (i.e., without perturbations not known a priori) compared to the baseline in which this probability stops at 15%. This is a clear advantage of robust optimization. Additionally, Fig. 5.2.12 shows that the ECDF of c_l relative to the robust optimum is much better centered around the nominal value than the baseline.

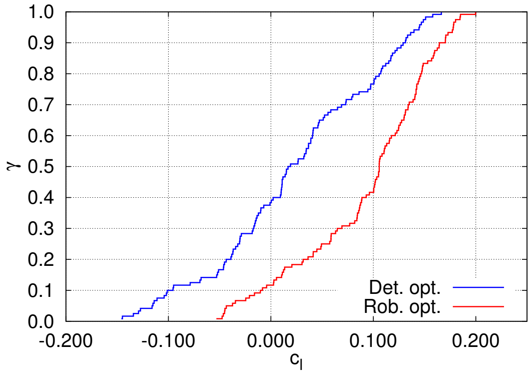
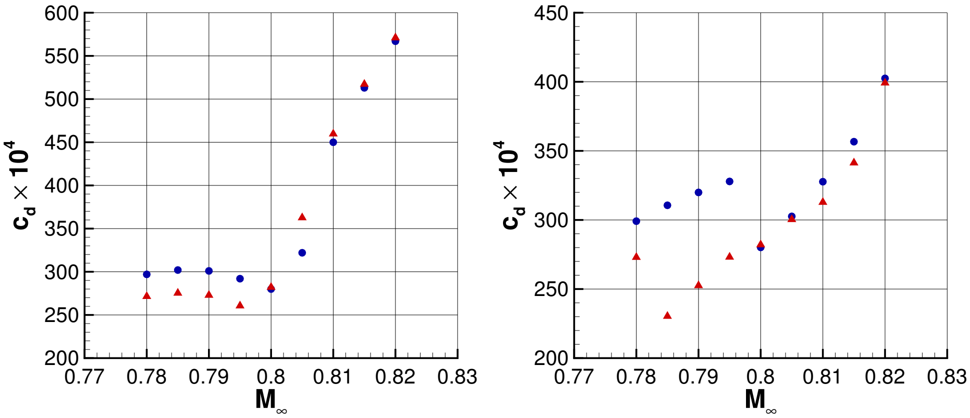


Figure 5.2.12: Lift coefficient ECDFs comparison for the deterministic and robust optimum.

Conversely, the advantages of the robust optimum are not evident in terms of drag rise. Basically, both the curves calculated at nominal c_l (Fig. 5.2.13a) and those at a constant angle of attack (Fig. 5.2.13b) show that the robust optimum performs better than the baseline only for values of the Mach number lower than or equal to the nominal one.

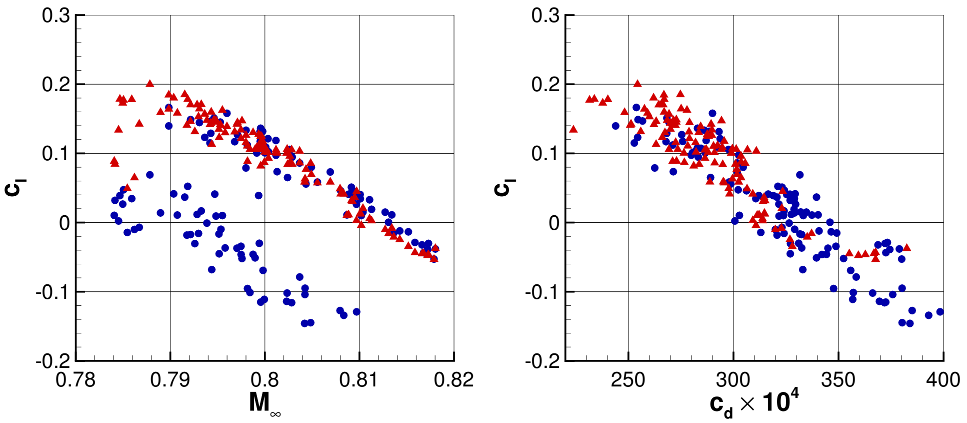


(a) Fixed lift coefficient.

(b) Fixed angle of attack.

Figure 5.2.13: Drag coefficient (c_d) versus Mach number (M_∞) for the deterministic (●) and robust (▲) optimum airfoils. Test case Figure (a): $Re_{c_\infty} = 174 \times 10^6$ and $c_l = 0.1$. Test case Figure (b): $Re_{c_\infty} = 174 \times 10^6$ and $\alpha = -2.86^\circ$ for the deterministic airfoil and $\alpha = -2.37^\circ$ for the robust airfoil.

Figure 5.2.14 suggests a possible explanation for this behavior. Indeed, Fig. 5.2.14a shows the lift coefficient graph as a function of Mach for the elements used for the ECDF. It shows that lift is anticorrelated to Mach and, similarly, Fig. 5.2.14b shows that it is also anticorrelated with drag force. Therefore, the drag rise curve (Fig. 5.2.13a) shows an evident criticality as the Mach number increases. This criticality also appears in the calculation performed at a constant angle of attack (Fig. 5.2.13b), albeit to a lesser extent. This suggests that if one wants to keep drag increment under control, it might be necessary to add an additional design point, similar to what was done in Section 4.5.5.



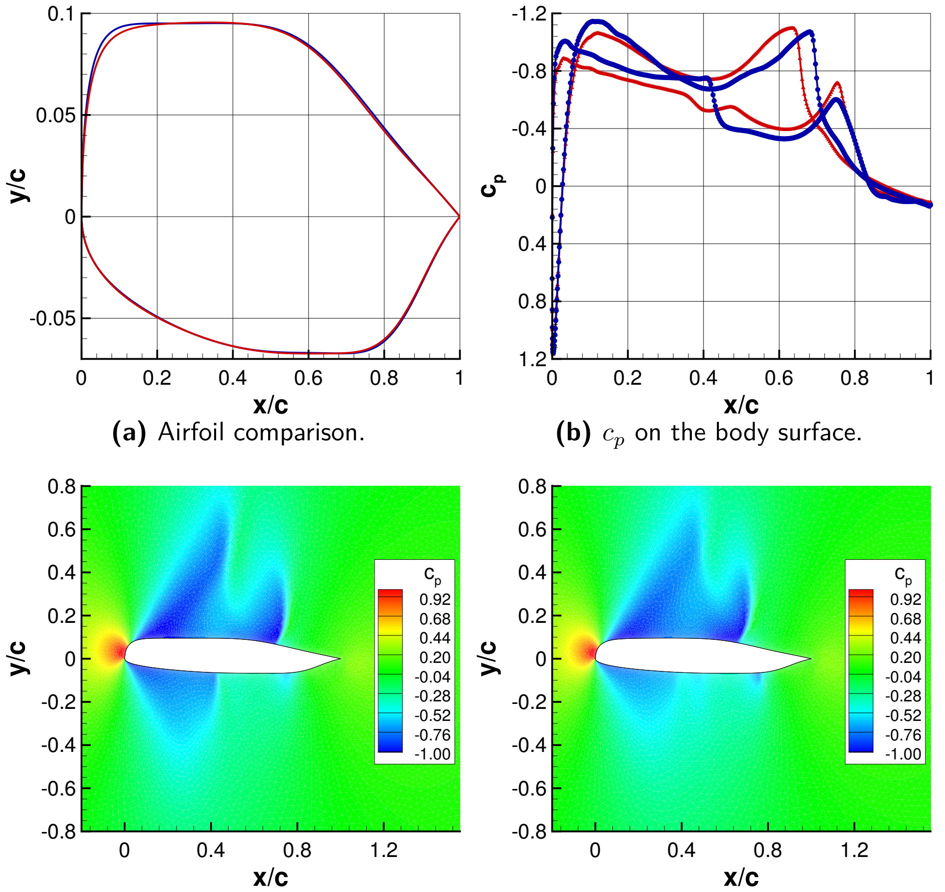
(a) c_l vs M_∞.

(b) c_l vs c_d.

Figure 5.2.14: Deterministic (●) and robust (▲) optimum airfoils.

By the way, the robust optimization effectiveness appears evident in the reduction of the c_d expected value.

Finally, the deterministic and robust optimum airfoils are compared in Fig. 5.2.15a. They present slight differences in shape, the most evident being in the leading edge region on the upper side of the airfoil. This difference is clearly appreciated in the pressure coefficient distribution, where the deterministic optimum presents a greater expansion with respect to the robust one, as shown in Fig. 5.2.15b. On the upper side of the airfoil, a second difference, barely visible in the geometry, is evidenced from $x/c = 40\%$, where a second expansion begins, but with different slopes. On the lower side, the expansion peak is higher for the deterministic optimum airfoil. In addition, a sudden compression at $x/c = 40\%$ is presented at the deterministic optimum. The shock wave at $x/c = 80\%$ is present on both airfoils, but it is weaker for the deterministic optimum airfoil. These differences in the pressure coefficient are also visible in Figs. 5.2.15c and 5.2.15d. The data is calculated at the nominal conditions ($M_\infty = 0.80$, $Re_{c_\infty} = 174 \times 10^6$, and $c_l = 0.1$).



(a) Airfoil comparison.

(b) c_p on the body surface.

(c) c_p contour deterministic optimum.

(d) c_p contour robust optimum.

Figure 5.2.15: Figure (a) provides the comparison of the deterministic (—) and robust (—) optimum airfoils (axis are not dependent), Figure (b) the comparison of the pressure coefficient distribution on the body surface for the deterministic (–●–) and robust (–▲–) optimum airfoils, and Figures (c) and (d) the c_p contour plots. The c_p is calculated at nominal conditions ($M_\infty = 0.80$, $Re_{c_\infty} = 174 \times 10^6$, and $c_l = 0.1$).

5.2.6 Computational cost

Finally, analogously Section 4.5.7, a comparison of the computational effort for calculating the ECDF of both methods, simple Monte Carlo and surrogate-based, is reported here.

When using the plain Monte Carlo approach, the same CPU cost estimation as in Section 4.5.7 is used for the CVaR calculation through the empirical cumulative distribution. The number of samples needed to get a satisfactory accuracy of the risk measure can be very high. Here, a minimum of 20 samples is assumed to be necessary to obtain a barely acceptable estimation of CVaR. Specifically, the construction of the ECDF requires a single fixed lift coefficient CFD run at the nominal condition point ($M_\infty = 0.8$ and $c_l = 0.1$) plus 20 fixed angle-of-attack CFD runs constructed by sampling the random variables that describe the problem uncertainty. The computational cost is normalized with respect to the average execution time of a fixed α run. The average cost of a fixed c_l run at proper convergence is instead roughly doubled with respect to the fixed α case.

The total cost of the Monte Carlo method is summarized in Table 5.2.4.

	20 fixed α runs	c_l fixed run	Total
Monte Carlo ECDF cost	20	2	22

Table 5.2.4: Computational cost of CVaR estimation using Monte Carlo sampling.

Regarding the surrogate-based ECDF, it requires a first CFD run at fixed c_l, like in the Monte Carlo approach. After that, 4 additional runs, at fixed α, are performed at the optimal points found in the training phase. Table 5.2.5 shows the computational cost of the proposed approach.

	4 fixed α runs	c_l fixed run	Total
Surrogate-based ECDF cost	4	2	6

Table 5.2.5: Computational cost of CVaR estimation using Monte Carlo sampling.

The computational costs ratio between plain Monte Carlo and GP-based approaches for building the ECDF is $22/6$. Note that, the cost of calculating the four "true" ECDF for the training phase must be added for the surrogate-based method. Furthermore, to build a "true" cumulative distribution function, one c_l fixed plus 119 fixed angle of attack runs are needed. Therefore, the total cost of the training set is $4 \times 121 = 484$. This cost is distributed among the whole robust optimization run, not only for the construction of a single ECDF. Finally, the optimization costs related to the search of the optimal sets of training points are negligible.

When including the true ECDF computation costs, the surrogate-based approach is more computationally expensive than the adjoint-based one. However, the method described here is less intrusive and, since it does not depend on a linear approximation, it is less subject to the need to introduce ad hoc corrections to take into account any nonlinearities. Moreover, the sample size used for the true ECDF used in the training process is rather conservative, and in a further step of this work, a smaller number of sampled points for the "true" ECDFs will be investigated.

5.2.7 Conclusions

In this study, an adaptive approach that allows an efficient implementation of robust optimization based on the use of risk functions and on Gaussian processes as a strategy for constructing response surfaces was presented. In particular, CVaR was used in a robust aerodynamic optimization loop to improve a transonic airfoil performance. The critical point of the procedure was introducing an iterative approach in which the training phase of the Gaussian process alternates with that of robust optimization, and the results of the latter are then used in the next phase to improve the Gaussian process. One of the training phase peculiarities is that one can choose between different objective functions to be optimized. In the most straightforward approach, the minimization of the distance between the cumulative distributions calculated with fine Monte Carlo sampling and those calculated with the Gaussian process is required. The minimization of the distances between the CVaR measures calculated with the two different methods can then be added. Finally, a penalty can be imposed if the Gaussian process fails to obtain the same order relationship between risk functions as the reference solutions obtained by the Monte Carlo sampling. At present, the switch between the various phases of the procedure is not automatic. Still, it must be carried out manually based on the user's empirical considerations on the progress of every single step of the robust optimization. Further developments of the methodology will consider constructing an integrated algorithm in which the transition between the robust optimization phase and the training phase of the Gaussian process can take place automatically. Despite the limitations indicated above, the presented procedure allowed the implementation of an effective and efficient robust design loop with a level of computational resources that is a fraction of that required by risk function-based approaches that do not approximate the objective function.

5.3 Multi-fidelity surrogate assisted design optimization of an airfoil under uncertainty using far-field drag approximation

In the previous section, Gaussian processes were employed to approximate the empirical cumulative distribution function. Thus, reducing the computational cost of calculating the risk measure of interest. Contrary, in this section, a surrogate-based optimization approach, as explained in Section 5.1, is used to solve a robust aerodynamic shape design case study.

Robust optimization techniques provide optimal designs that are less vulnerable to the presence of uncertainty. These techniques are already used at early stage of the design process, since the aerodynamics performance of real-world applications is inherently uncertain due to manufacturing errors, uncertain environmental conditions [1, 2] and other physical phenomena like icing [3]. Therefore, uncertainties should be considered already during the aerodynamic design of airfoils. However, as already mentioned, such optimization techniques require numerous expensive function evaluations since the quantity of interest of the problem is a statistical measure. Notably, the quantities of interests in aerodynamic shape design are calculated by solving the Reynolds-averaged Navier-Stokes equations with computational fluid dynamic solvers that heavily impact the computational budget.

A multi-fidelity surrogate-based method is proposed here to reduce the computational time required for robust design. The accuracy of a surrogate highly depends on the size of its training data, as shown earlier in this chapter (Section 5.1). Therefore, when the function evaluations are costly, such as a CFD simulation with a very fine grid, the training data can be complemented by function evaluations cheaper and with lower fidelity. The information coming from various fidelity levels can be fused together with multi-fidelity Gaussian process regression (MF-GPR). This technique was introduced by [114].

In this work, the drag coefficient (c_d) of the MH 114 [115] propeller airfoil is minimized by a multi-fidelity surrogate assisted optimization technique. The open-source fluid-dynamic solver SU2 is used for calculating the c_d. SU2 solves the compressible RANS equations numerically and calculates c_d by integrating the stress over the body surface with the so-called near-field method.

The drag coefficient c_d can have different levels of fidelity by using diverse grid refinements. A calculation with a fine grid provides a high-fidelity c_d prediction. However, fine meshes are very demanding from a computational point of view. Coarse grids are computationally cheap, but they introduce a higher proportion of spurious drag. This numerically introduced drag stems from the truncation error of the used numerical methods and the artificial dissipation of solving the RANS equations with a coarse grid. The artificial dissipation is added in the numerical schemes to boost the convergence of the flow and to stabilize the scheme. Hence, the prediction of the near-field c_d deteriorates.

Nevertheless, there are far-field methods for the estimation of the drag force that allow the c_d prediction with a level of accuracy similar to a fine grid by identifying the spurious drag sources. A review of all these methods is given in [53]. In this work, the formulation described in Chapter 3 has been implemented.

The prediction of the drag coefficient using the far-field method will be used for the low-fidelity level on a coarse grid. This procedure will allow a better estimation of the drag coefficient with respect to the near-field value computed on the same grid, thus resulting in an increased accuracy while preserving the computational cost. In addition, the near-field c_d estimation obtained with a fine grid will be used for the high-fidelity level.

5.3.1 Multi-fidelity Gaussian Process Regression

The multi-fidelity Gaussian process based optimization framework used in this research work was made available by ESTECO within the UTOPIAE project framework. The capabilities of this approach have already been proven in [116, 117].

There are several strategies to obtain the different levels of fidelity of the drag coefficient calculation. One can use various flow solvers that provide diverse accuracy of the c_d value. Besides, in this work, it was decided to introduce the levels of fidelity by using different grid levels. Coarse grids provide, due to spurious drag, a low-fidelity prediction of the c_d at a low computational cost. Whereas, fine grids give an accurate prediction of the drag coefficient but are computational expensive. In addition, to allow a better estimation of the c_d at the coarse grid, the far-field method introduced in Chapter 3 is used.

Therefore, the drag coefficients obtained with SU2 flow solver using a fine grid (high-fidelity) and the far-field c_d approximation using the coarse mesh (low-fidelity) are fused together into a single surrogate by multi-fidelity Gaussian process regression (MF-GPR).

The recursive formulation is adapted here as proposed by [118]:

$$\tilde{f}_1(\mathbf{x}) = \mathbf{h}_1^T(\mathbf{x})\boldsymbol{\beta}_1 + \tilde{\delta}_1(\mathbf{x}), \tag{5.24a}$$

$$\tilde{f}_2(\mathbf{x}) = \rho(\mathbf{x})\tilde{f}_1(\mathbf{x}) + \mathbf{h}_2^T(\mathbf{x})\boldsymbol{\beta}_2 + \tilde{\delta}_2(\mathbf{x}), \tag{5.24b}$$

$$\rho(\mathbf{x}) = \mathbf{g}^T(\mathbf{x})\boldsymbol{\beta}_\rho, \tag{5.24c}$$

where indices 1 and 2 denote the low and high-fidelity levels, respectively. The mean trend of the fidelity level is formulated as a least-squares regression $\mathbf{h}_i(\mathbf{x})\boldsymbol{\beta}_i$ with the vector of regression functions $\mathbf{h}_i(\mathbf{x})$ and the vector of regression coefficients $\boldsymbol{\beta}_i$. The local variations of the model are modeled as zero-mean Gaussian distributions with σ_i^2 variance and incorporated into $\tilde{\delta}_i(\mathbf{x}) \sim \mathcal{N}(0, \sigma_i^2)$. This recursive formulation, first, trains a standard GPR surrogate using the low-fidelity samples calculated by the far-field method. Then, the posterior of the low-fidelity GPR is combined with the high-fidelity observations of SU2 by training an additional GPR. This recursive formulation avoids the need to construct a large covariance matrix containing the low- and high-fidelity designs as in [114]. Even if the training cost of the surrogate is negligible compared to the aerodynamic design evaluation, the reduced covariance size is advantageous as the model is frequently re-trained during the optimization process.

5.3.2 Aerodynamic computational chain

The self-operating aerodynamic computational chain, used in this work, takes as input the design variables given by the optimizer and generates the candidate to be evaluated, builds the computational mesh, and runs the computational fluid dynamic flow solver. Finally, the obtained performance of the candidate is provided to the optimizer. In the design problem studied here, the performance of the candidate is the airfoil drag coefficient. Moreover, for the low-fidelity runs, instead of providing directly the c_d given by the CFD solver, the drag coefficient is calculated with a far-field approach and later is provided to the optimizer. This pipeline is shown in Fig. 5.3.1.

The candidate airfoils are generated using *wg2aer* that, from a set of input design variables values, modifies the starting airfoil MH 114 accordingly to some modification functions. The functioning of *wg2aer* is provided in Section 2.2.1. Specifically, ten design parameters are considered. The first and second design parameters describe a thickness

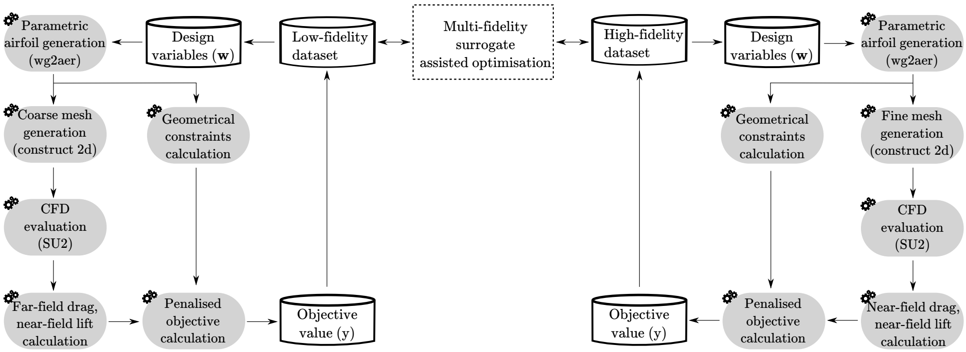

Parametric airfoil generation (wg2aer)
Design variables (w)
Low-fidelity dataset
Multi-fidelity surrogate assisted optimisation
High-fidelity dataset
Design variables (w)
Parametric airfoil generation (wg2aer)
Coarse mesh generation (construct 2d)
Geometrical constraints calculation
Geometrical constraints calculation
Fine mesh generation (construct 2d)
CFD evaluation (SU2)
CFD evaluation (SU2)
Far-field drag, near-field lift calculation
Penalised objective calculation
Objective value (y)
Objective value (y)
Penalised objective calculation
Near-field drag, near-field lift calculation

Figure 5.3.1: Aerodynamic computational chain

mode; hence they have the same value but opposite sign ($w_2 = -w_1$). Also, the third and fourth design parameters represent a camber mode; thus both are equal ($w_4 = w_3$). Therefore, this will lead to eight optimization design variables instead of ten. The range for the design parameters are:

$$\begin{aligned} w_1, w_2 \in [-2, 2], \quad & w_3, w_4 \in [-2, 2], \\ w_5, w_6, w_7, w_8 \in [-1, 1], \quad & w_9, w_{10} \in [-0.2, 0.2] \end{aligned} \tag{5.25}$$

Regarding the modification functions, the first four are polynomials affecting the whole airfoil, while the rest are Hicks-Henne bump functions that have the location of the bump at different position of the airfoil chord. In Fig. 5.3.2, an example of airfoil modifications is provided.

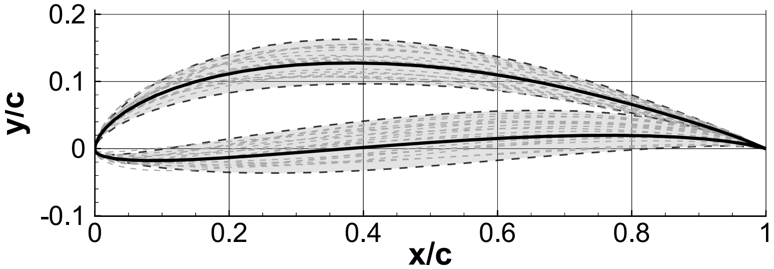


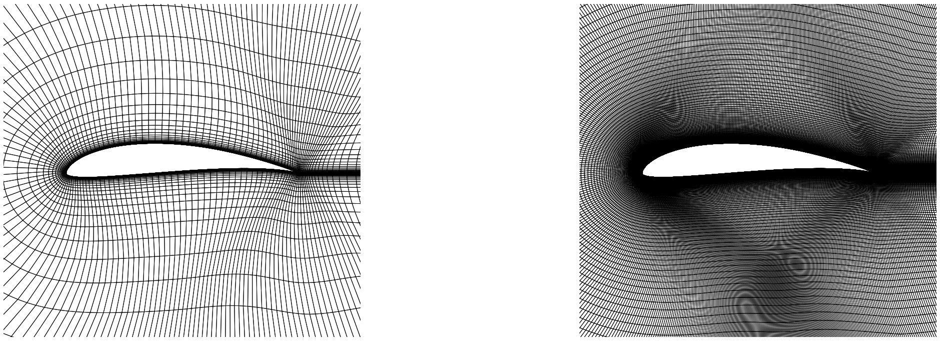
Figure 5.3.2: Modified airfoils example. Baseline airfoil (—)

In addition, the open-source grid generator *Construct2D* is used to create the 2D meshes. In particular, given the coordinates of the modified airfoil, a C-type grid is generated. The number of cells of the mesh depends on the level of fidelity that has to be run. The possible mesh sizes are provided in Table 5.3.1.

	N_b	N_w	N_j	N_{total}
Low fidelity (LF)	96	48	48	9216
High fidelity (HF)	512	256	256	262144

Table 5.3.1: Mesh size parameters for low- and high-fidelity simulations. (N_b: number of cells on the body surface, N_w: number of cells in the wake, N_j: number of cells in far-field direction, N_{total}: total number of cells)

For both grid refinements, the far-field is located at 500 airfoil chords. The possible computational meshes are depicted in Fig. 5.3.3.



(a) Low-fidelity grid

(b) High-fidelity grid

Figure 5.3.3: Possible grids depending on the fidelity level.

Finally, the CFD solver used for the aerodynamic shape design optimization problem is the open-source fluid-dynamic solver SU2. The compressible Reynolds-averaged Navier-Stokes (RANS) equations are solved. The turbulence model used is the Spalart-Allmaras (SA) [75]. Furthermore, for the spatial integration, JST central scheme with artificial dissipation coupled with an implicit Euler method for the pseudo-time stepping is used.

5.3.3 Far-field Drag Coefficient Calculation

The far-field method implemented in this work was introduced in [6] and exhaustively explained in Chapter 3. It allows the decomposition of the drag force in three components:

wave, viscous, and spurious drag. Specifically, the method is based on entropy variations. The entropy drag is expressed as volume integral, which allows the decomposition of the drag into the above components. Hence, a proper selection of each region is needed. The entropy drag is defined as:

$$D_{\Delta s} = D_w + D_v + D_{sp} \tag{5.26}$$

where D_w, D_v, and D_{sp} are the wave, viscous, and spurious contributions, respectively.

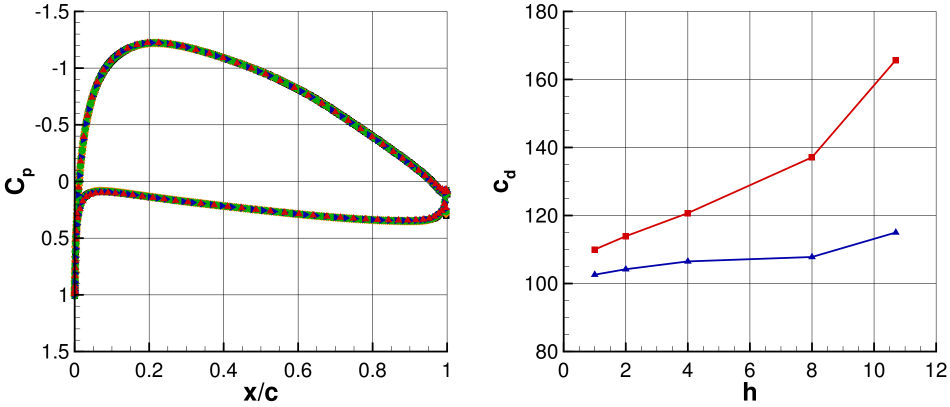
D_{sp} is the drag source related to the entropy introduced by the truncation error and the artificial dissipation of the numerical schemes used by the Computational Fluid Dynamics flow solver. Hence, by identifying the D_{sp} contribution and subtracting it from Eq. (5.26), a prediction of the drag coefficient, with an accuracy close to fine grids, is obtained on a coarser mesh. This implies a considerable advantage while facing aerodynamic optimization problems since coarser grids allow a significant reduction of the required computational time. This advantage for optimization has also been shown in [65].

Furthermore, a test to verify this advantage for the design problem of the propeller blade airfoil, here studied, has been carried out. In particular, a viscous flow with working conditions $M_\infty = 0.2$, $Re_\infty = 4.97 \times 10^6$ and $c_l = 1.0$ is performed on the MH 114 airfoil. The compressible RANS equations are solved using the SU2 flow solver with the SA turbulence model. Five C-type grids of an increasing number of cells are studied. The grid size is obtained by the square root of the ratio between the number of cells of the finest grid and the grid under evaluation $\left(h = \sqrt{N_{h=1}/N_i}\right)$. The number of cells on the body surface (N_b), on the wake (N_w), in the far-field direction (N_j), and the total number of cells (N_{total}), as well as the near-field value of the drag coefficient ($c_{d_{nf}}$), and the far-field value (c_{d_v}) are given in Table 5.3.2. Note that in this test case, the only drag contribution is the viscous (D_v), and the drag values are expressed in drag counts ($1\text{dc} = 10^{-4}$).

	N_b	N_w	N_j	N_{total}	$c_{d_{nf}}$ [dc]	c_{d_v} [dc]
$h = 10.7$	96	48	48	9216	165.7	115.0
$h = 8$	128	64	64	16384	137.1	107.8
$h = 4$	256	128	128	65536	120.7	106.5
$h = 2$	512	256	256	262144	113.9	104.2
$h = 1$	1024	512	512	1048576	110.0	102.6

Table 5.3.2: Mesh sizes and computed drag coefficients. Viscous test at $M_\infty = 0.2$, $Re_\infty = 4.97 \times 10^6$ and $c_l = 1.0$.

In Fig. 5.3.4, the comparison between near-field and far-field drag coefficients versus the grid size is given. In addition, the pressure coefficient distribution on the body surface (c_p) at the different grid refinements is also plotted.



(a) Pressure Coefficient, c_p

(b) Drag Coefficient, c_d

Figure 5.3.4: MH 114 test at $M_\infty = 0.2$, $Re_\infty = 4.97 \times 10^6$ and $c_l = 1.0$. *Left*: Pressure coefficient distribution on the body surface at $h = 1$ (■), $h = 2$ (◆), $h = 4$ (●), $h = 8$ (▶), and $h = 10.7$ (▲). *Right*: Near-field (−■−) and far-field (−▲−) drag coefficients versus mesh size.

In Fig. 5.3.4a it can be observed that the differences between the c_p on the body surface are barely visible. Thus, a good local accuracy of the solution is also demonstrated at the coarsest grid level. However, Fig. 5.3.4b shows how the near-field value of the drag coefficient converges as the grid is refined $(h \longrightarrow 0)$. The variation on c_d between the coarsest and finest mesh sizes is given by the spurious drag source introduced by the numerical method and the artificial dissipation. Contrarily, using the far-field analysis of the drag force, the spurious drag contribution is removed. Hence, a better estimation of the c_d is found, and applying Richardson extrapolation, at $h = 0$, the difference between the two methods is approximately 6.6 dc.

Hence, with this method, the drag coefficient value for the lower fidelity of the multi-fidelity surrogate model is improved in accuracy while keeping the same computational time.

5.3.4 Optimization design problem description

The shape optimization design problem aims at the minimization of the drag coefficient (c_d) of a propeller blade airfoil. Geometric and aerodynamic constraints are considered. The baseline design is the propeller airfoil Martin Hepperle MH 114. The flow conditions are $M_\infty = 0.2$, $Re = 4.97 \times 10^6$, and $\alpha = 2^\circ$. The lift coefficient of the airfoil is required to be greater or equal to one $(c_l \geq 1)$. Geometrical constraints are imposed for obtaining realistic shapes. The percentage thickness with respect to the airfoil chord $(t_\%)$ is fixed to the value of the baseline. The Leading Edge Radius (LER) and the Trailing Edge Angle (TEA) are constrained by minimum values not to fall below their baseline values by more than 10%.

In mathematical terms, the deterministic optimization example reads:

$$\begin{cases} \min\limits_{\mathbf{w}} & c_d(\mathbf{w}) \\ \text{subject to:} & \\ & c_l(\mathbf{w}) \geq 1.0 \\ & t_\%(\mathbf{w}) = 13.05 \\ & \mathrm{LER}(\mathbf{w}) \geq 0.011377 \\ & \mathrm{TEA}(\mathbf{w}) \geq 6.0^\circ \end{cases} \tag{5.27}$$

A penalty approach is used to handle the constrained problem.

$$\begin{aligned} \min_{\mathbf{w} \in W \subseteq \mathbb{R}^n} c_d(\mathbf{w}) + p_{c_l} \max\left(0, 1 - c_l(\mathbf{w})\right) + p_{\mathrm{LER}} \max\left(0, 0.011377 - \mathrm{LER}(\mathbf{w})\right) + \\ + p_{\mathrm{TEA}} \max\left(0, 6 - \mathrm{TEA}(\mathbf{w})\right), \end{aligned} \tag{5.28}$$

where the $p_{c_l} = 1000$, $p_{\mathrm{LER}} = 100000$ and $p_{\mathrm{TEA}} = 100$. The equality constraint of the thickness is imposed by scaling the modified airfoil shape to the given value, and this step is directly fulfilled using *wg2aer*.

Furthermore, to improve the performance of the airfoil under uncertainty, a robust optimization must be performed. Here, the angle of attack is the only parameter considered uncertain, thus representing uncertainty in the environmental conditions. Particularly, the uncertainty has been modeled with a four-parameter beta distribution. The variability range is $\alpha_{ref} \pm 0.25$. The statistical measure chosen to solve the robust design problem is the Conditional Value at Risk (CVaR) measure at a confidence level γ equal to 0.95.

$$\begin{cases} \min\limits_{\mathbf{w}} & \mathrm{CVaR}^{0.95}\left(c_d(\mathbf{w},\mathbf{u})\right) \\ \text{subject to:} & \\ & \mathrm{CVaR}^{0.95}_{loss}\left(c_l(\mathbf{w},\mathbf{u})\right) \geq 1.0 \\ & t_{\%}(\mathbf{w}) = 13.05 \\ & \mathrm{LER}(\mathbf{w}) \geq 0.011377 \\ & \mathrm{TEA}(\mathbf{w}) \geq 6.0^{\circ} \end{cases} \tag{5.29}$$

Therefore, the robust optimization problem reads:

$$\begin{aligned} \min_{\mathbf{w}\in W\subseteq\mathbb{R}^n} \mathrm{CVaR}^{0.95}\left(c_d(\mathbf{w},\mathbf{u})\right) + p_{c_l}\max\left(0, 1-\mathrm{CVaR}^{0.95}_{loss}\left(c_l(\mathbf{w},\mathbf{u})\right)\right) + \\ + p_{\mathrm{LER}}\max\left(0, 0.011377-\mathrm{LER}(\mathbf{w})\right) + p_{\mathrm{TEA}}\max\left(0, 6-\mathrm{TEA}(\mathbf{w})\right), \end{aligned} \tag{5.30}$$

where $\mathrm{CVaR}^{0.95}_{loss}\left(c_l\right) = -\mathrm{CVaR}^{0.95}\left(-c_l\right)$ is the loss Conditional Value at Risk. The random perturbations of the angle of attack impacts only the aerodynamic force requirements $(c_d$ and $c_l)$. The geometric constraints can be evaluated for each design configuration $(\mathbf{w})$ deterministically. The same penalty parameters have been used as in the deterministic case.

5.3.5 Optimization pipeline

To reduce the computational cost of the shape design case study presented in Section 5.3.4, a tailored multi-fidelity surrogate based optimization strategy introduced by Korondi et al. [117] is applied here. Particularly, the far-field drag predictions are used to populate the training dataset as the computational budget affords only few high-fidelity RANS simulations. The proposed optimization pipeline is presented in Fig. 5.3.5.

In surrogate based optimization, an important step is the selection of the infill criterion to update the response surface at each iteration. For this study, the acquisition methodology is based on the constrained Expected Improvement (cEI). It can be seen as a combination of the expected improvement of the objective and the probability that the constraint

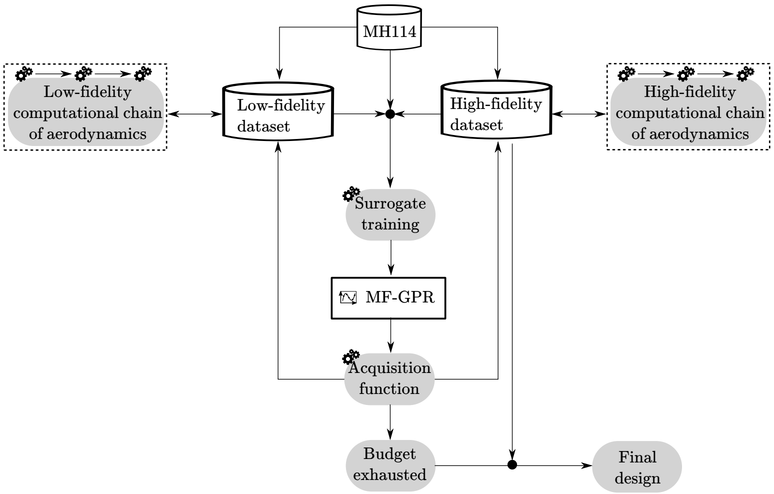


Figure 5.3.5: Optimization pipeline with multi-fidelity surrogate

on lift coefficient is not satisfied. The cEI is formulated as:

$$
\begin{aligned}
\text{cEI} &= \mathbb{E}\left[\max\left(0, F^*_{obj} - F_{obj}(\tilde{c}_d, \tilde{c}_l)\right)\right]\mathbb{P}\left[\tilde{c}_l \geq 1\right] \\
&= \left(\left(F^*_{obj} - F_{obj}(\hat{c}_d, \hat{c}_l)\right)\Phi\left(\frac{F^*_{obj} - F_{obj}(\hat{c}_d, \hat{c}_l)}{\hat{\sigma}^2_{c_d}}\right) + \right. \\
&\quad \left. + \hat{\sigma}^2_{c_d}\phi\left(\frac{F^*_{obj} - F_{obj}(\hat{c}_d, \hat{c}_l)}{\hat{\sigma}^2_{c_d}}\right)\right)\Phi\left(\frac{\hat{c}_l - 1}{\hat{\sigma}^2_{c_l}}\right),
\end{aligned}
\tag{5.31}
$$

where $\hat{\sigma}_{c_d}$ and $\hat{\sigma}_{c_l}$ are the standard deviations of the drag and lift coefficient, respectively. F_{obj} is the penalized objective given by Eq. (5.28) or Eq. (5.30), in order, for the deterministic and robust optimization studies. The best evaluated objective value is given by F^*_{obj}. The Φ and ϕ symbols denote the cumulative distribution function and probability density function of the standard normal distribution, respectively. $\mathbb{E}$ is the expected value, and $\mathbb{P}$ is the probability operator.

The new candidate to be evaluated by the true function is selected at the point where the constrained expected improvement is maximum.

$$
\mathbf{w}_{new} = \max_{\mathbf{w} \in W \subseteq \mathbb{R}^n} \text{cEI} \tag{5.32}
$$

For multi-fidelity surrogate optimization, in addition to the acquisition methodology, a strategy that chooses the fidelity level at which to evaluate the most promising design candidate $\mathbf{w}_{new}$ needs to be defined. In this work, the fidelity is chosen based on the Scaled Expected Variance Reduction (SEVR) measure [117]. Specifically, SEVR balances the cost associated with evaluating a candidate with a given fidelity level and the expected variance reduction.

$$l = \max_{LF,HF} \text{SEVR}_l \tag{5.33}$$

Where, the SEVR is defined as:

$$\text{SEVR}_{\text{LF}} = \frac{\rho^2(\mathbf{w}_{new})\hat{\sigma}^2_{c_d,\text{LF}}(\mathbf{w}_{new})}{c_{\text{LF}}}, \tag{5.34}$$

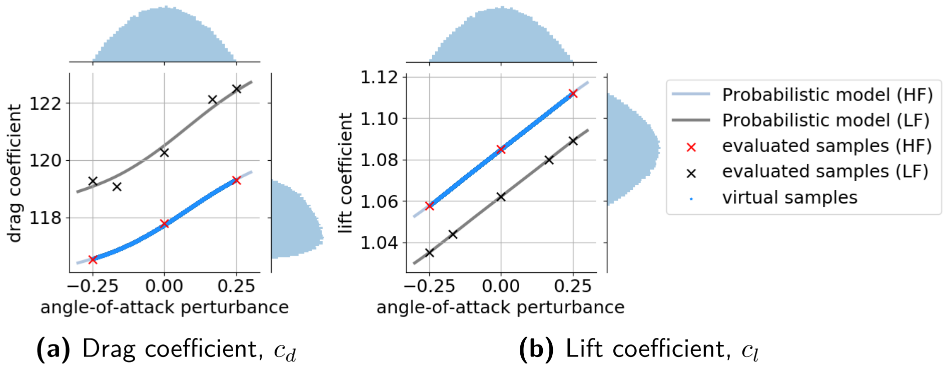
$$\text{SEVR}_{\text{HF}} = \frac{\rho^2(\mathbf{w}_{new})\hat{\sigma}^2_{c_d,\text{LF}}(\mathbf{w}_{new}) + \hat{\sigma}^2_{c_d,\delta_{\text{HF}}}(\mathbf{w}_{new})}{c_{\text{HF}}}, \tag{5.35}$$

where $c_{LF} = 1$ and $c_{HF} = 10$ are the costs of the low- and high-fidelity simulations, respectively.

Regarding the robust optimization, the computational chains of low- and high-fidelity have to be complemented by an additional step for computing the lift and drag coefficients CVaR risk measures. Due to the fact that this process is computationally demanding, the risk measure is calculated with a surrogate-based uncertainty quantification approach. At the low fidelity level, for each design configuration, five LF samples are used for constructing a local GPR model, while at the high-fidelity level, five LF samples and three HF samples are used for constructing a local MF-GPR model. These models can then be used to draw a statistically significant number of samples to calculate the risk measure. The number of HF samples is set to the minimum required samples necessary for training the local probabilistic model. Arbitrarily, it was decided to increase the number of LF samples by 20% w.r.t. the HF approximation.

The constructed local probabilistic models of the baseline configuration are presented in Fig. 5.3.6. Furthermore, in Fig. 5.3.7, the convergence of the risk measure value of the aerodynamic force coefficients in relation to the number of virtual samples is depicted. Based on the CVaR convergence, 100000 virtual samples of the local probabilistic models are generated to calculate the CVaR values for both aerodynamic coefficients (c_l and c_d) in the present work.

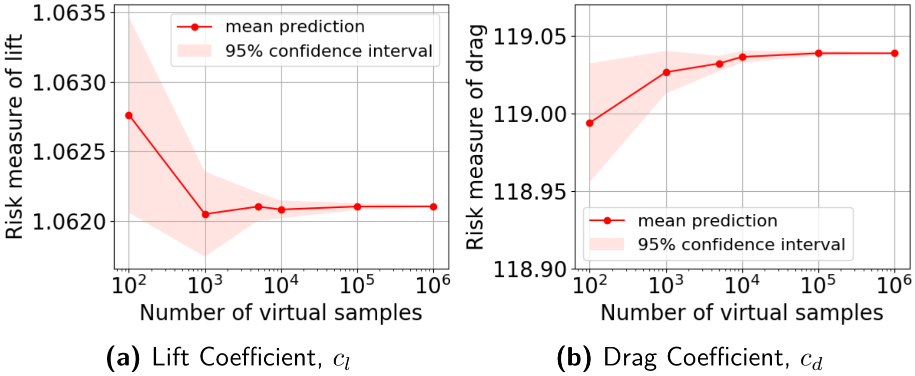
Note that the computational costs of the fidelity levels are set according to their true



(a) Drag coefficient, c_d

(b) Lift coefficient, c_l

Figure 5.3.6: Local probabilistic models of the aerodynamic force coefficients of the baseline configuration.



(a) Lift Coefficient, c_l

(b) Drag Coefficient, c_d

Figure 5.3.7: Convergence of the risk measure value of the aerodynamic force coefficients.

computational time required for the probabilistic optimization. The cost of running a low-fidelity CFD evaluation is considered as 1. According to the computational time, the cost for high-fidelity is 16 times greater than the low-fidelity runs. Considering that five LF samples are needed to build the low-fidelity probabilistic model, the total cost of the LF model is 5. On the other hand, to construct the high-fidelity probabilistic model, three HF and five LF evaluations are required; thus, the total cost is 53. Therefore, a 1 to 10 cost-ratio is used in this study. Note that the computational costs of training the surrogates and calculating the acquisition function are considered negligible compared to the CFD evaluations.

Finally, the computational chain of the aerodynamic forces with the probabilistic model is shown in Fig. 5.3.8.

5.3.6 Results

The most promising obtained results are shown here-in. A two-step optimization was carried out. Firstly, a deterministic optimization, and, later, a probabilistic one.

Deterministic optimization

The deterministic design optimization problem posed in Eq. (5.27) is solved with a multi-fidelity and a single-fidelity surrogate-based technique, and, also, with a population-based technique. A resume of all the optimization results is presented in Table 5.3.3.

	c_d	c_l	TEA	LER	Samples	Cost
	[dc]		[deg]		[LF, HF]	
baseline (MH 114)	117.8	1.09	6.60	0.014324	$[0, 1]$	10
best MF-GPR	112.0	1.00	6.03	0.013228	$[120, 18]$	300
best GPR	117.6	1.08	7.76	0.011973	$[0, 30]$	300
best CMA-ES ($h = 10.7$)	117.1	1.02	6.17	0.018025	$[1800, 0]$	1800
best CMA-ES ($h = 8$)	115.6	1.02	6.34	0.020900	$[1800, 0]$	2700

Table 5.3.3: Comparison of multi-fidelity, single-fidelity surrogate-based and population-based optimization results. (The cost of a single high- and low-fidelity simulations are 10 and 1 respectively.)

The same computational budget is used to compare the single- and multi-fidelity methodologies regarding the surrogate-based approaches. In both cases, 10 high-fidelity samples are used for constructing the initial surrogate. In the case of MF-GPR, the surrogate is complemented with 100 low-fidelity samples. Therefore, to keep the same computational budget, only 20 additional samples are generated to complement the high-fidelity surrogate. In Table 5.3.3, it can be observed that the optimization that uses MF-GPR is able to find a better solution. This is due to the fact that the computational budget was severely limited, hence this did not allow us to have enough high-fidelity samples. The lack of HF samples prohibits the construction of an accurate GPR model. However, by introducing low-fidelity information obtained from computationally cheaper samples, the MF-GPR

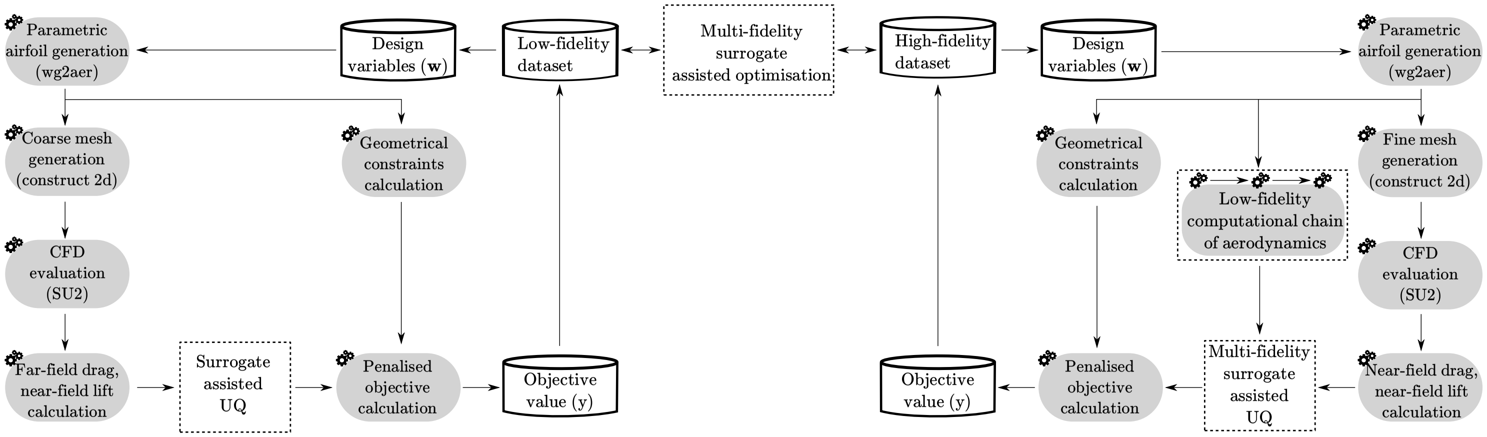

Parametric airfoil generation (wg2aer)
Design variables (w)
Low-fidelity dataset
Multi-fidelity surrogate assisted optimisation
High-fidelity dataset
Design variables (w)
Parametric airfoil generation (wg2aer)
Coarse mesh generation (construct 2d)
Geometrical constraints calculation
Geometrical constraints calculation
Low-fidelity computational chain of aerodynamics
Fine mesh generation (construct 2d)
CFD evaluation (SU2)
CFD evaluation (SU2)
Far-field drag, near-field lift calculation
Surrogate assisted UQ
Penalised objective calculation
Objective value (y)
Objective value (y)
Penalised objective calculation
Multi-fidelity surrogate assisted UQ
Near-field drag, near-field lift calculation

Figure 5.3.8: Computational chain of the aerodynamic forces with the probabilistic model.

could better approximate the performance landscape. Clearly, this handful of new samples are not enough to find a sufficiently good design.

From an aerodynamic perspective, the optimal airfoil of the MF-GPR approach has a lower drag coefficient since its camber line is lower than the camber lines of the other two airfoils, the MH114 and the optimal airfoil obtained at the optimization using the GPR approach. This can be observed in Fig. 5.3.9. Note that, for visualization purposes, the axes in Figs. 5.3.9 and 5.3.10 are not dependent. Specifically, by decreasing the camber and keeping the free stream angle of attack, the effective angle of attack that the airfoil actually perceives decreases. Thus, a lower lift coefficient is obtained. This implies a reduction of the lift-induced drag coefficient, hence of the total drag.

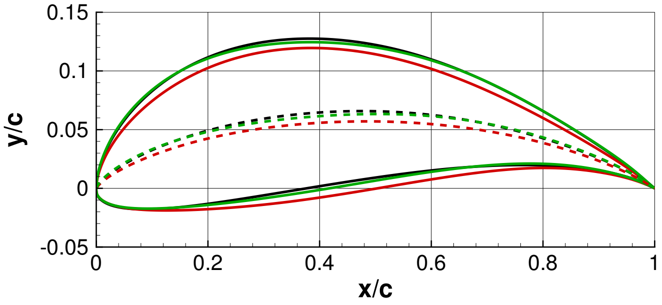


Figure 5.3.9: Baseline and deterministic optimal airfoil comparison. MH114 (—), MF-GPR optimal airfoil (—), and GPR optimal airfoil (—). The *dashed* lines are the camber of each airfoil. Axes are not dependent.

In addition, the presented design optimization approach is compared with a well-known population-based algorithm, namely CMA-ES (see Section 2.1.2). The optimization is performed using only low-fidelity CFD evaluations. The evolutionary algorithm is not able to find a similar optimal design to the one given by the MF-GPR approach. Contrarily, the airfoil is barely optimized. Therefore, it is decided to increase the mesh size to $h = 8$ (instead of $h = 10.7$) and redo the optimization. In this case, the algorithm found a similarly best design configuration to the presented method (MF-GPR). The cost of a CFD evaluation on $h = 8$ grid size is 1.5 times the cost of a low-fidelity one. In addition, to perform the population-based optimization, 1800 evaluations are needed. This implies that the computational cost is significantly increased with respect to the multi-fidelity approach presented in this paper.

Therefore, the presented optimization method presents an advantage with respect to single-fidelity surrogate-based optimization since by adding low-fidelity samples, the performance landscape is better approximated. This results in a better allocation of computational resources. The multi-fidelity surrogate-based approach can find better airfoil designs compared to classical population-based optimization, and single-fidelity techniques as the HF evaluations are performed only for promising design candidates.

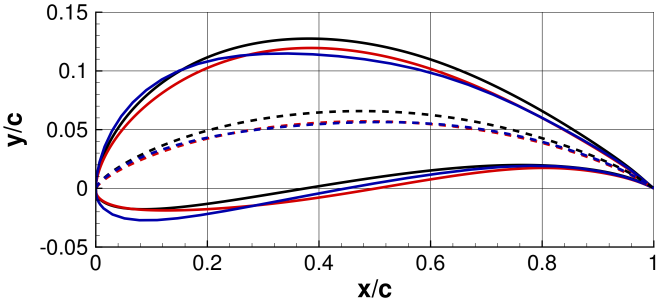


Figure 5.3.10: Baseline and deterministic optimal airfoil comparison. MH114 (—), MF-GPR optimal airfoil (—), and CMAES optimal airfoil using $h = 8$ grid size (—). The *dashed* lines are the camber of each airfoil. Axes are not dependent.

Furthermore, Fig. 5.3.10 shows the comparison of the optimal airfoils obtained with CMA-ES and with MF-GPR approaches with respect to the baseline. In addition, the camber line of each airfoil is also plotted. In this case, both optimal airfoils have a similar camber line. However, the obtained optimal design with CMA-ES has the maximum airfoil thickness placed at a forward position ($x/c = 0.227$) with respect to the MF-GPR best design ($x/c = 0.297$). This implies that the effective angle-of-attack is greater than the one perceived by the MF-GPR optimal airfoil. Hence, higher c_l and, consequently, c_d are found.

	$\hat{c}_d$	$\hat{c}_l$	$\hat{F}_{obj}$	$F_{obj}(\hat{c}_d, \hat{c}_l, \text{TEA}, \text{LER})$	HF iterations
MF-GPR	2.044%	0.71%	34.51%	5.43%	8
GPR	3.11%	5.65%	17.14%	11.53%	20

Table 5.3.4: Comparison of prediction error of multi- and single-fidelity surrogate models. (Prediction error is defined as the arithmetic mean value of the relative error of the high-fidelity predictions during the course of optimization.)

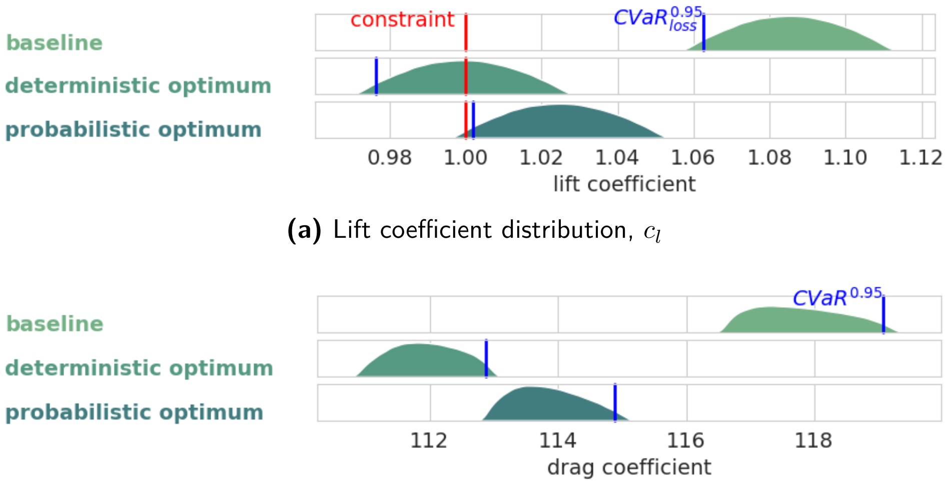
To evaluate the quality of the produced surrogate models, the mean prediction error is calculated. The results are summarised in Table 5.3.4. Particularly, the prediction error was calculated for each HF iteration by using the prediction and true values calculated at the new infill design point. It can be seen that MF-GPR predicted the aerodynamic forces of the new designs significantly better. The poor design configuration found by the GPR-based optimization is also due to the penalized approach which was employed here. The lift coefficient is not well predicted which drives the algorithm to waste computations on unfeasible designs and have high objective values. Moreover, in Table 5.3.4, it is also shown that the objective cannot be accurately predicted by a surrogate directly. However, by independently predicting the aerodynamic forces and calculating the objective afterwards based on these predictions, it was more accurately predicted.

Probabilistic optimization

After solving the deterministic problem, a probabilistic optimization is carried out to improve the performance of the airfoil under uncertainty. Specifically, only uncertainty on the angle of attack is considered. Taking into account that the advantages of using MF-GPR are already shown in the deterministic optimization, the probabilistic one is made using the proposed optimization approach.

The predicted lift and drag distributions of the optimal designs and the baseline airfoil are shown in Fig. 5.3.11. Figure 5.3.11a shows that, if the angle of attack is perturbed, the optimal design obtained in the deterministic problem violates the constraint imposed for the lift coefficient. Contrary, by considering the uncertainty in the angle of attack during the optimization using the MF-GPR approach, the obtained design respects the constraint imposed on the lift coefficient. Particularly, for the probabilistic design optimization, it is decided that $\mathrm{CVaR}_{loss}(\tilde{c}_l)$ must be greater or equal to one. To fulfill the constraint, the robust optimum design has to increase the lift coefficient with respect to the deterministic optimum, as shown in Fig. 5.3.11a. Consequently, the robust design has a higher drag (see Fig. 5.3.11b).

In Table 5.3.5, a comparison of the baseline airfoil with the optimal designs is given in terms of lift, drag, and shape characteristics. Besides, in order to calculate the CVaR, the MF-GPR technique requires the computation of three high-fidelity samples. Hence, the computational budget is triple. However, this is still limited as the size of the budget is equal to the cost of only 90 high-fidelity simulations.



(a) Lift coefficient distribution, c_l

(b) Drag coefficient distribution, c_d

Figure 5.3.11: Prediction of the distributions for the baseline and optimal designs.

	Objective	CVaR($\tilde{c}_d$) [dc]	CVaR$_{loss}$($\tilde{c}_l$)	TEA [deg]	LER	Samples [LF, HF]	Cost
baseline (MH 114)	119.07	119.07	1.063	6.60	0.014324	$[5, 3]$	35
deterministic opt.	136.55	112.87	0.976	6.03	0.013228	$[5, 3]$	35
probabilistic opt.	114.89	114.89	1.002	6.38	0.016937	$[370, 51]$	4880

Table 5.3.5: Comparison of the requirements considering environmental uncertainty.

The obtained optimal airfoils are compared in Fig. 5.3.12. It is appreciated that both the deterministic and the probabilistic optimization resulted in a smaller camber line curvature airfoil than the baseline. The MH 114 airfoil generates a significantly higher lift coefficient than the required constraint value; hence the optimization tends to reduce the camber curvature, so that the lift reduces, and so does the drag. Moreover, by comparing the deterministic and probabilistic optimal designs, it can be seen that the probabilistic optimum has a stronger S-shaped lower side. This increases drag and lift coefficient; hence, resulting in a feasible airfoil design.

Finally, in Fig. 5.3.13, a comparison of the pressure coefficient distribution and friction coefficient on the body surface of the baseline, deterministic and probabilistic optimal designs is presented. Analysing both optimal designs, it can be observed that the deterministic optimum presents a smoother expansion rate on the upper surface of the airfoil.

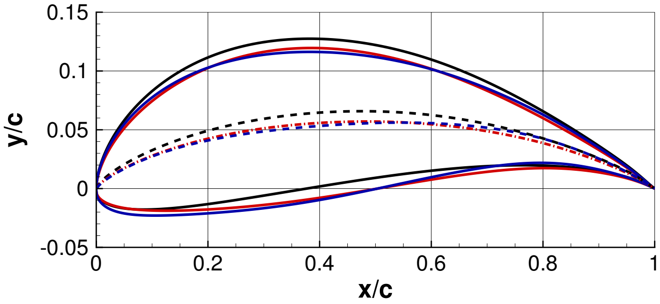
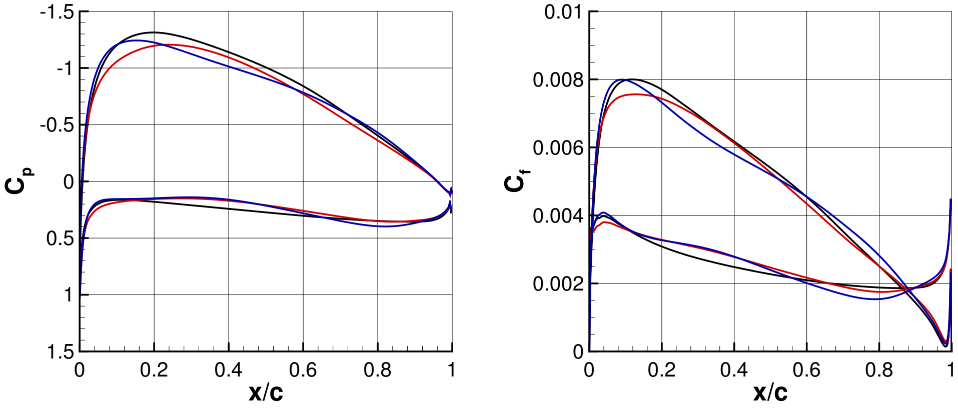


Figure 5.3.12: Baseline, deterministic optimum, and robust optimum airfoil comparison. MH 114 (—), deterministic optimum (—), and probabilistic optimum (—). The *dashed* lines are the camber of each airfoil. Axes are not dependent.

Specifically, the maximum is reached at 30% of the chord, whereas the probabilistic optimum has the peak at 10% of the chord. Comparing the pressure coefficient distribution of the optimal airfoils, the contribution of pressure to drag coefficient is clearly higher for the probabilistic optimum. Besides, for the contribution of friction on the drag coefficient, a similar conclusion can be applied. The maximum value of skin friction coefficient is higher for the probabilistic optimum on the suction side of the airfoil and, also, to a minor extent on the pressure side. Thus, the friction drag is higher for the probabilistic solution, mainly due to the peak of maximum c_f.



(a) Pressure Coefficient, c_p

(b) Friction Coefficient, c_f

Figure 5.3.13: Pressure coefficient *(left)* and friction coefficient *(right)* on the body surface comparison. MH 114 (—), deterministic optimum (—), and probabilistic optimum (—).

5.3.7 Conclusion

In the presented study, a surrogate-based optimization under uncertainty workflow able to solve expensive aerospace applications has been developed. The optimization pipeline is employed to find an optimal propeller blade airfoil that minimizes the drag produced but satisfying the requirement imposed on the lift coefficient, together with some geometrical constraints to obtain feasible designs. To predict the aerodynamic forces, the compressible Reynolds Averaged Navier-Stokes equations are solved using the open-source CFD flow solver SU2. The computational time required to solve these equations is high; thus, a multi-fidelity surrogate technique is used to decrease the computational effort. Two levels of grid refinement are used to build the multi-fidelity surrogate. In addition, to increase the correlation of the high- and low-fidelity drag predictions, the spurious drag at the low-fidelity is compensated by performing the drag prediction using far-field formulation.

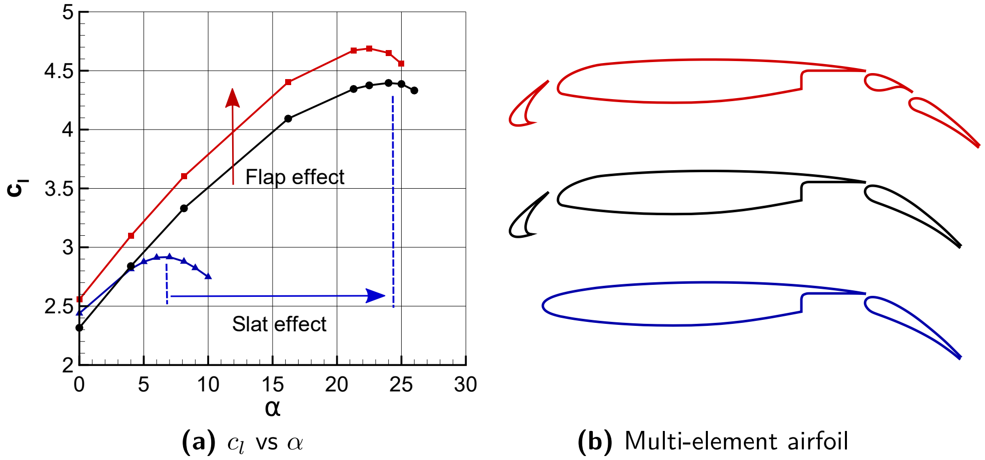
The multi-fidelity Gaussian process approach has been compared against classical single-fidelity surrogate-based and evolutionary algorithms. The results showed that single-fidelity surrogate methods could struggle to find significantly improved designs due to the limited computational budget, that it is solved by introducing the information obtained at a lower fidelity. Moreover, it has been demonstrated that the computation budget has to be increased to obtain similar results with a population-based algorithm. Finally, the highest potential of the multi-fidelity surrogate-based approach presented in this case study relies on solving problems under uncertainty where the required numerous probabilistic samples can be efficiently obtained by introducing a multi-fidelity probabilistic model.

6 Machine Learning Assisted for High-Lift Devices Topology Robust Optimization

This chapter copes with the optimization under uncertainty of the topology of high-lift devices. The chapter has two main parts. Firstly, a deterministic optimization containing continuous, integers, and nominal categorical variables is solved using an heuristic algorithm, and, secondly, the same problem is carried on but introducing uncertainty. A surrogate-based approach plus some machine learning techniques are used to make it computationally treatable. This work was made in collaboration with the University of Strathclyde and Cologne University of Applied Sciences (TH Köln) and has been published in [119] and presented at the International Conference on Uncertainty Quantification & Optimization UQOP2020 (in Press).

6.1 Introduction

Recently, in the field of commercial aircraft design, the interest regarding the optimal design of High-Lift Devices has increased [120, 121]. High-Lift Devices (HLDs) are components located on the aircraft's wing that aim to increase the lift force produced by the wing during slow flight phases, mainly take-off and landing. A wide variety of HLD types exists, and the most common are slats and flaps that are the object in this work. The slat is located at the airfoil leading edge and has the effect of delaying stall by increasing the angle of attack at which the maximum lift is attained. In contrast, the flap is a trailing edge device that increases the lift coefficient even at low angles of attack [122, 123]. The effects of the mentioned HLDs are shown in Fig. 6.1.1. The *black* airfoil is the McDonnell Douglas (MDA) 30P-30N airfoil, the *blue* airfoil is the 30P-30N with the slat not deployed, and the *red* airfoil is the 30P-30N, but the flap has been split to make a double-slotted airfoil.



(a) c_l vs α **(b)** Multi-element airfoil

Figure 6.1.1: Slat and flap lift coefficient effect. 30P-30N airfoil (–●–), double-slotted flap (–■–), and 30P-30N airfoil with no slat (–▲–).

As discussed in [124], the HLD design is an arduous task because it is multidisciplinary (aerodynamics, structures, systems integration disciplines are involved), multi-objective (generally in conflict) [125], and multi-point (designed for take-off, climbing, approach, landing flight phases) [126]. Thus, HLD design is very demanding from a computational point of view. One critical point is the proper convergence of the Computational Fluid Dynamic (CFD) solver because configurations at high angles of attack, close to stall conditions, must be simulated. Generally, these configurations lead to separate flows that make the accurate prediction of the aerodynamic performance very challenging. As a consequence, experimental results to validate the CFD are necessary.

Classically, the design of HLDs is a two-stage procedure. First, a promising configuration type (number of airfoil elements) is selected, and second, the position and shape of the elements are obtained through optimization, often employing heuristic algorithms [124]. Furthermore, because most of them are limited to continuous numeric variables and fixed-length search spaces, optimization algorithms are assigned to solve a circumscribed part of the original design problem exclusively. It is challenging to know beforehand which configuration is optimal in terms of performance, weight, and cost. As a result, this approach is likely to lead to sub-optimal solutions. This work aims to demonstrate the feasibility of including the HLDs configuration as a design variable of the optimization phase. In particular, the optimizer explores the HLD search space changing the number of elements (or configuration) and their position and rotation. For simplicity, the optimization objective is limited to maximize the lift generated by the HLD at a given angle of attack regardless of its drag, weight, costs, and practical construction feasibility. Because of this, the problem falls under the area of mixed-variable single objective global optimization.

Dealing with these kinds of problems implies tackling many challenges. Indeed, even a proper definition of the design problem may be very difficult, as well as finding an efficient mapping and encoding of the design variables to facilitate the optimizer efficiency. This problem is even more critical when variables of a different type, like continuous (numerical) and categorical (nominal), simultaneously encode topological and dimensional features. Among the different algorithms, one of the most suited to face this particular problem is Genetic Algorithms (GAs) [127].

Various strategies for facing variable-size global optimization are described in the literature [128, 129]. The *hidden gene* adaptation of GA for the optimization of interplanetary trajectories is introduced in [129]. Here the maximum number of genes that can describe a candidate is set. Each candidate is then represented using all the possible genes plus a set of *activation genes*, indicating which genes to consider when computing the objective function. However, in the cases where the activation of genes depends on the activation of other ones or on the values they assume, defining an effective problem formulation can be very tough or even impossible.

A more complex, but efficient and flexible adaptation of GA is proposed in [128]. In this case, a hierarchical multi-level chromosome structure is adopted in place of the standard *string-based* one. Unlike the standard GA formulation, both vicinity and hierarchy relationships link the genes of each chromosome. However, as presented in [130], the operators implemented can result ineffective or even destructive because they act regardless of the variable type.

In the current work, a mixed-variable with dynamically varying search space global optimizer based on a genetic algorithm, the Structured-Chromosome Genetic Algorithm [130–133] is used. This approach consists of an adaptation of GA that allows structured-chromosomes definitions. The algorithm takes advantage of hierarchical problem formulations and uses revised operators that behave considering the type of variables and the history of the optimization [130, 132, 133]. Moreover, this offers the possibility to define dynamic bounds and dynamic variables dependencies, and this makes it suitable for a wide variety of problem formulations and applications.

This work aims to show that an appropriately crafted optimization algorithm can handle configurational decisions that are usually the responsibility of experienced designers. Indeed, this achievement can represent a turning point not only in the field of design optimization but also in multidisciplinary design optimization, structural optimization and many other design disciplines.

6.2 Aerodynamic computational chain

As explained in Section 2.2, it is essential to have a complete autonomous aerodynamic computational chain to solve aerodynamic design optimization problems. The computational chain used to solve this design problem is depicted in Fig. 6.2.1 and briefly explained in the following paragraphs since all the tools are thoroughly described in Section 2.2. Moreover, in Section 6.2.1 the CFD solver is validated. In Section 6.2.2, the design of the different multi-element airfoil used as baseline configurations for the optimization are generated.

During the optimization, the candidate airfoils are generated using *wg2aer*. As explained in Section 2.2.1, it is an in-house developed program that accepts as input a set of design variable values and modifies a specified starting airfoil. For obtaining the baseline airfoil, the design variable values must be equal to 0. In this study, the introduced design variables are the settings of each airfoil element, which are the translation in the 2-dimensional space (ΔX and ΔY) and the rotation ($\Delta\theta$) with respect to the starting airfoil. The reference point for rotation is the leading edge of the corresponding element.

After generating the candidate airfoil, the grid is automatically built using a self-developed procedure based on the open-source grid generator *Gmsh*. It generates an unstructured square grid combining triangular and rectangular elements. The far-field is located at 40

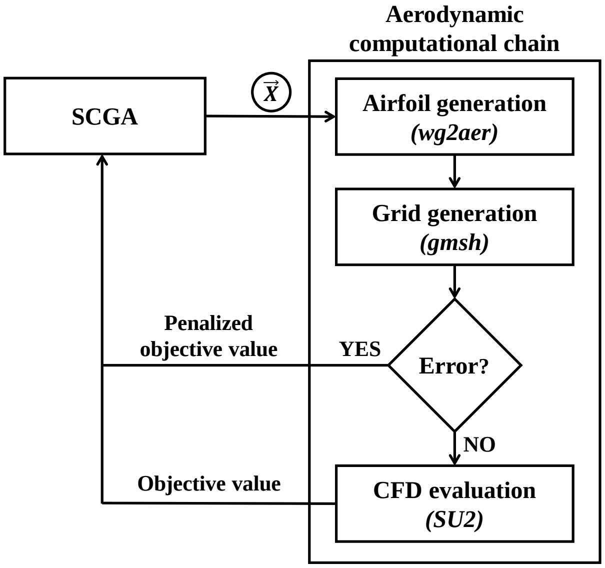


Figure 6.2.1: Aerodynamic computational chain representation.

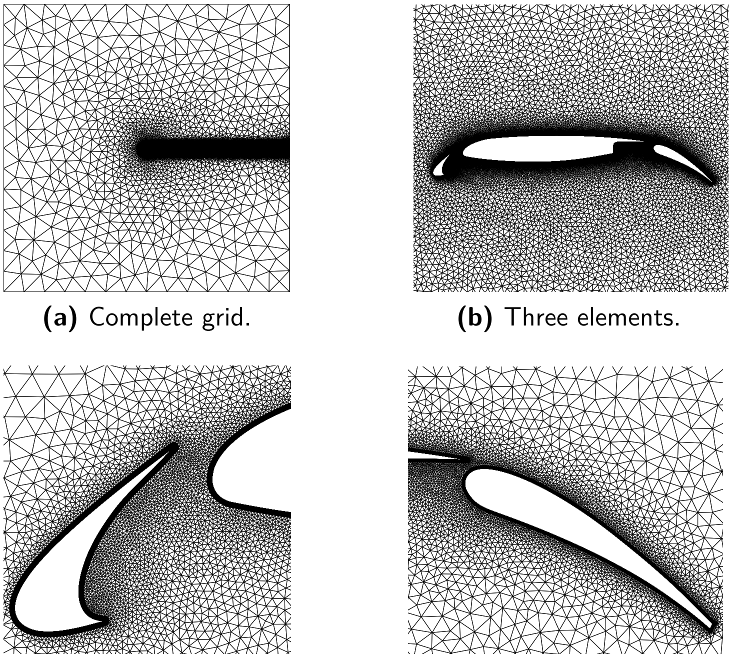
airfoil chords. An example of the computational mesh used for validating the CFD flow solver is shown in Fig. 6.2.2.

Finally, the Computational Fluid Dynamic solver used for the HLD design optimization is the open-source fluid-dynamic solver SU2 (see Section 2.2.3). The compressible Reynolds-averaged Navier-Stokes (RANS) equations are solved using the SA turbulence model [75]. Also, JST central scheme with artificial dissipation coupled with an implicit Euler method for the pseudo-time stepping is used for spatial integration.

6.2.1 CFD results validation and design point selection

Before performing an optimization run, the CFD solver must be validated. This step is essential when designing HLDs. During their design, configurations at high angles of attack are studied. These configurations could present highly separated flows, hence, predicting the airfoil performance is a challenging task for the solver. Therefore, a comparison with experimental results is required.

A series of wind-tunnel experiments were conducted at NASA Langley Research Center for the three-elements airfoil McDonnell Douglas (MDA) 30P-30N [134, 135]. Thus, the 30P-30N airfoil (Fig. 6.2.3) was selected as the starting configuration for this design



(a) Complete grid.

(b) Three elements.

(c) Main element & slat.

(d) Main element & flap.

Figure 6.2.2: Computational mesh.

optimization problem. These experimental data are used to validate the numerical results.

The considered working conditions are: free-stream Mach number $M_\infty = 0.2$ and Reynolds number $Re_c = 5 \times 10^6$ (based on the airfoil chord, c) at several angles of attack, namely, $\alpha = [0^\circ, 4^\circ, 8.12^\circ, 16.21^\circ, 21.29^\circ, 22.5^\circ, 24^\circ, 25^\circ, 26^\circ]$. A computational mesh with ~ 200000 cells is used (see Fig. 6.2.2).

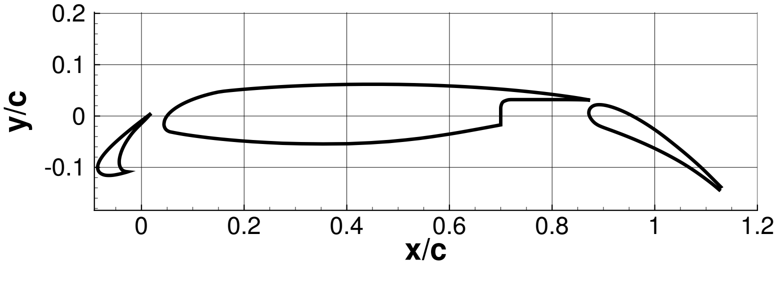
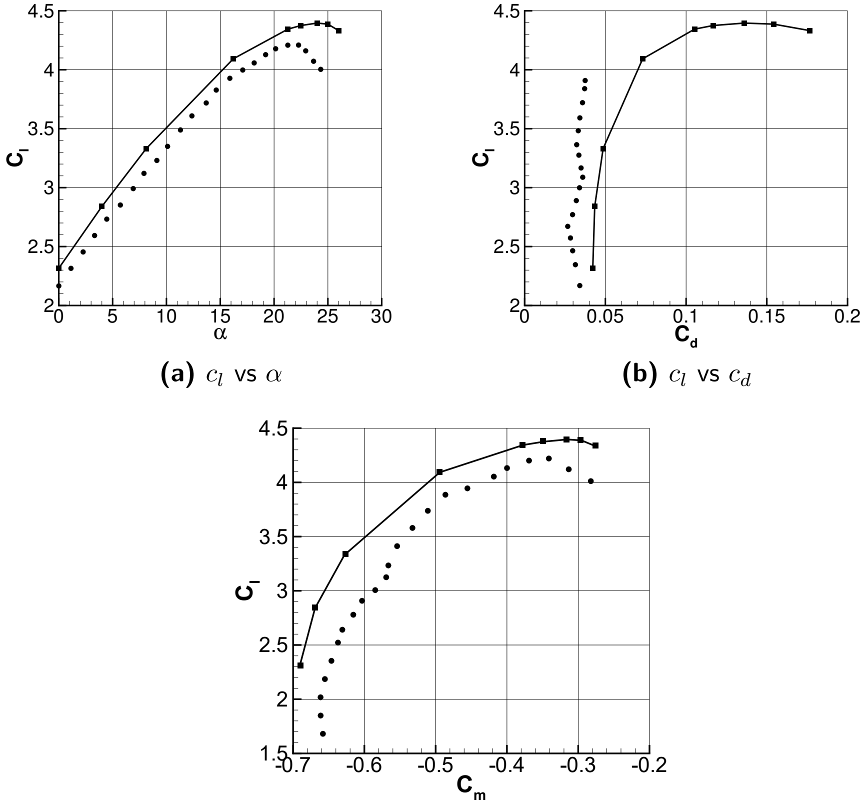


Figure 6.2.3: 30P-30N airfoil.

An important concern is to demonstrate the ability of the CFD solver to assess stall. Thus, a comparison of experimental and numerical lift coefficient (c_l) versus angle of attack (α) curves is shown in Fig. 6.2.4a. It is observed that the computed lift is slightly higher than the experimental data. In addition, the computed stall angle $\left(\alpha \simeq 24^\circ\right)$ is higher than the experimental $\left(\alpha \simeq 21^\circ\right)$.



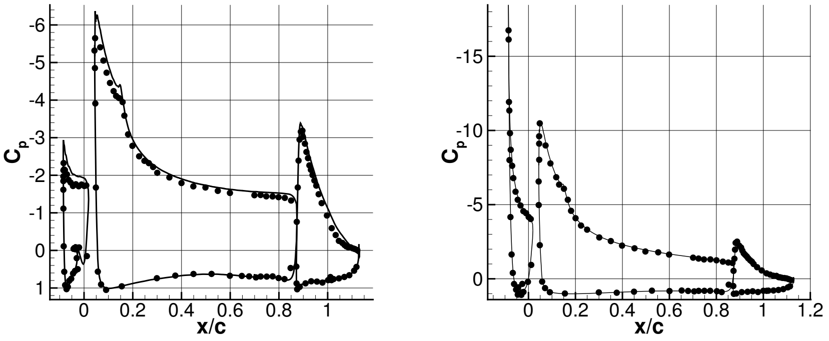
(a) c_l vs α

(b) c_l vs c_d

(c) c_l vs c_m

Figure 6.2.4: SU2 (–■–) and experimental (•) data comparison.

The drag polar (Fig. 6.2.4b) shows that numerical simulations predict a higher drag coefficient (c_d). This trend is expected since a fully turbulent approach was adopted while the experiments were conducted in free transition. Finally, it must be mentioned that the pitching moment c_m is computed at 25% of the chord, and a negative value indicates that it is a nose-down moment. Both, experimental and numerical results show a decrease of the nose-down moment with increasing angles of attack. Moreover, computed pressure distributions (c_p) at the body surface for $\alpha = 8.12^{\circ}$ and $\alpha = 21.29^{\circ}$ are compared with the experiments in Fig. 6.2.5. A satisfactory match between both is observed.



(a) c_p at $\alpha = 8.12^{\circ}$

(b) c_p at $\alpha = 21.29^{\circ}$

Figure 6.2.5: Comparison of the pressure coefficient at the body surface. SU2 results (—) and experimental data (●).

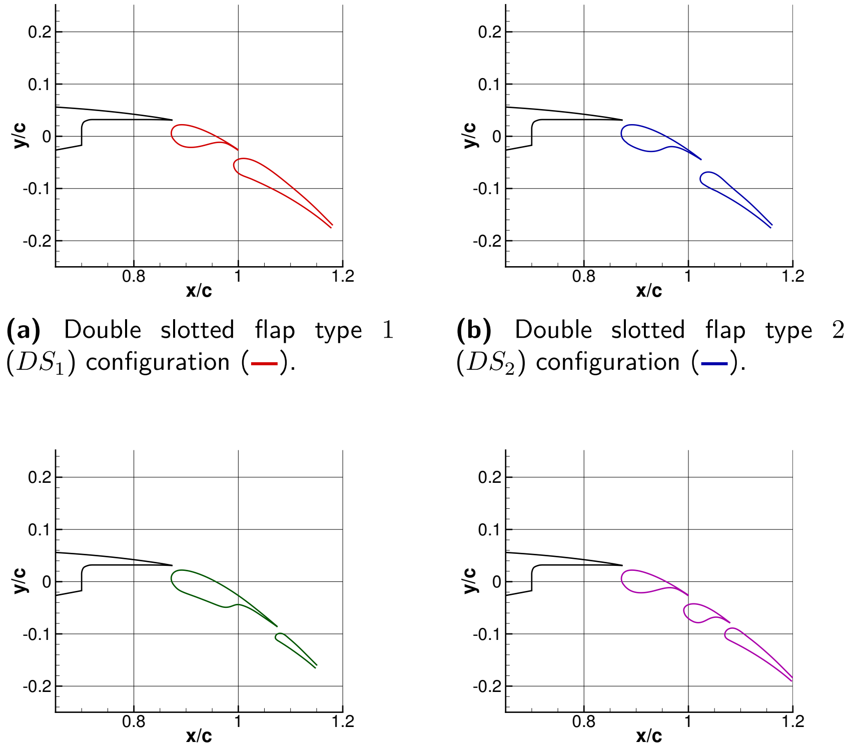
Summing up, it can be concluded that the CFD solver is capable of predicting the airfoil performance, although there are some variations between numerical and experimental results.

6.2.2 Flap topology generation

The proposed design optimization aims to select the airfoil topology most suitable for the considered working conditions. Hence, several multi-element airfoil configurations must be considered as candidates for the HLD optimization. To do so, the 30P-30N airfoil was set as the reference baseline to design four different flap configurations, in particular, three different types of double-slotted flap and one triple slotted flap.

In order to split the single flap of the baseline airfoil in several flap components, the multi-element airfoil manipulation program AIRSET [136] was used. Once the new elements were generated, XFOIL [32] was employed for smoothing the splitting surfaces.

Thus, during the HLD optimization, five different High-Lift Device candidates are considered: a single flap configuration (*SF*), which is the 30P-30N airfoil, three types of



(a) Double slotted flap type 1 (DS_1) configuration (—).

(b) Double slotted flap type 2 (DS_2) configuration (—).

(c) Double slotted flap type 3 (DS_3) configuration (—).

(d) Triple slotted flap (TS) configuration (—).

Figure 6.2.6: Illustration of all the configurations considered in the optimization.

double-slotted flap $(DS_1$, DS_2, and $DS_3)$, and a triple slotted flap (TS) (see Fig. 6.2.6). Note that the slat and the main body parts are the same for all the configurations. In a further step, configurations without slat can also be taken into consideration. However, this study aims to demonstrate that the proposed optimization procedure is able to solve topology selection design problems. Hence, a more challenging problem definition is led to future works.

6.3 SCGA

In this work, the optimizer used is not the one explained in Section 2.1 since continuous and discrete optimization variables must be treated. Therefore, a brief explanation of the optimizer, taken from [119], is here-in included.

The Structured-Chromosome Genetic Algorithm (SCGA) [133] is a heuristic algorithm optimization that aims at coping with mixed-variable optimization problems with dynamically varying search space. This algorithm aims at dealing with a class of problems as broad as possible. Notably, it can effectively manage variable-size problem formulations containing continuous, integers, and nominal categorical variables.

SCGA is a general-purpose optimizer, able to deal with problems from different fields and which may present different specific characteristics. For this reason, it consists of a revised version of the popular GA [137]. In this work, the flexibility of the SCGA is leveraged to incorporate the decision about the HLDs topology in the optimization loop. Contrary to standard GAs where the chromosomes are encoded as strings, so the variables are independent, SCGA encodes the chromosome as trees allowing the definition of dependencies between variables. This creates a hierarchy between variables. For instance, the value of one variable can determine the presence or absence of other variables, or even their permissible values. This fact is advantageous in the cases where the optimizer is controlling a configurational decision where each possibility requires the specification of different additional parameters. In the presented HLD optimization, the variable selecting the type of flap impacts the number of variables needed to specify the position and rotation of all the flap elements. In light of these considerations, it is clear that standard genetic operators cannot be adopted. SCGA, makes use of genetic operators revised to handle hierarchical problem definitions. The customized crossover and mutation operators permit meaningful transformations that respect the interdependence between variables and permit to take advantage of insights the user can have about the problem [130, 132, 133]. The crossover operation consists of swapping genes between two different chromosomes (parents) to produce two new candidates (children). This operation aims at merging the information contained in the parents into the children. In this way, hopefully, the children inherit the characteristics that originated the performance of their parents and, combining them, they can reach even better performances. The crossover implementation in SCGA takes advantage of the hierarchy operating not only on the selected variables but also on all the dependent variables. This operation is necessary to create meaningful solutions and preserve the overall information encapsulated by the selected variables and the associated ones [130]. The mutation operator aims at avoiding premature convergence and increases the diversity in the population. The variables that undergo mutation are perturbed from their current state. The strength of the perturbation determines the quantity of randomness introduced, and it should be varied during the optimization considering a variety of factors, i.e., the diversity in the population and the type of variable [138, 139]. However, this is strongly problem dependent and difficult to foresee without an in-depth knowledge of the problem. The mutation operator employed in SCGA implements a self-adaptive step size that aims to adjust the strength of the perturbations autonomously [133]. The operation differs significantly depending on the type of the variable undergoing the mutation. In the case of continuous or integer variables, the operator perturbs the value to change with a small perturbation. A normal distribution generates the perturbation for the continuous variables, while the difference of two geometrical distributions is used for the integer ones [140]. In the case of nominal categorical variables, since all the possibilities are equally likely, the mutation operator simply substitutes the current value with another one. Particularly, this is re-sampled (uniform randomly) from the set of possible values deprived of the current assumed (for further details, please refer to [140]).

6.3.1 SCGA validation

Before solving the High-Lift Device topology design problem, the algorithm has been tested on a preliminary *'toy'* problem to validate SCGA capabilities when solving airfoil optimization problems. Indeed, although the potential of SCGA was shown in references [130, 132, 133], it has never been applied to problems with characteristics similar to those of HLD design. In particular, the purpose of this validation is to check the capacity of finely tuning continuous variables. This prerequisite is inescapable for the HLD design problem object of this study, where the continuous variables are predominant (see Table 6.4.1).

Another goal of this preliminary study is to test the robustness of the optimizer to different settings. Due to computational budget limitations, only one run of SCGA for the mixed-variable optimization is possible. Consequently, an accurate algorithm parameter tuning is out of the reach of this study. Nevertheless, some assessment work on a similar problem is necessary to check the real performance of the algorithm.

Problem description

The *'toy'* problem here presented was introduced in Section 1.4, and it is the deterministic aerodynamic design optimization problem of an airfoil in incompressible flow conditions and subject to geometric and aerodynamic constraints. The objective is to improve the airfoil performance, drag minimization in this case, by modifying its shape. The chosen baseline airfoil is the NACA 2412, and the working conditions are $M_\infty = 0$ and $Re_c = 5 \times 10^5$ at fixed lift coefficient $c_l = 0.5$. To obtain a fixed c_l, the angle of attack is adjusted.

The airfoil parametrization is made using $wg2aer$ tool explained in Section 2.2.1. In particular, it is parametrized by a linear combination of an initial geometry $(x_0(s), y_0(s))$, and modification functions $y_i(s)$. The airfoil shape is given by Eq. (2.2). The shape is controlled by the design parameters w_i and the scale factor k used to scale the airfoil to the required thickness. Specifically, in this optimization problem the number of design parameters is $n = 20$ in the range $w_i \in [-5, 5]$. Moreover, the maximum thickness (t) of the airfoil is fixed to 12% of the chord (c).

The quantity of interest (QoI) J to be optimized is the drag coefficient, c_d. The airfoil optimization is subjected to some geometrical and aerodynamic constraints. In particular, the imposed constraints are: trailing edge angle (TEA) greater than or equal to 13°, leading edge radius (LER) greater than or equal to 0.7% of the chord, and the boundary layer transition point on the lower surface (XTR_{LOW}) of the airfoil cannot take place at x/c greater than 0.95. Furthermore, an $Error$ flag, if equal to 1, will indicate the no convergence of the solver. Bearing this in mind, the objective function is given by

$$J = \begin{cases} c_d & \text{if } Error = 0 \text{ \& fulfilled constraints} \\ p\, c_d & \text{if } Error = 0 \text{ \& no fulfilled constraints} \\ p & \text{if } Error = 1, \end{cases} \tag{6.1}$$

with $p = 1000$. Thus, the deterministic problem reads:

$$\min J\,. \tag{6.2}$$

The drag coefficient is computed using XFOIL aerodynamic analysis code [32]. In a fully interactive way, it couples a high-order panel method with an integral boundary layer code.

Experiment

Five different settings of SCGA have been tested on the *'toy'* problem validate the capacity of SCGA of tuning continuous variables finely. Specifically, the parameters varied are the *population size* and the *tournament size*. The increase of the former helps to keep diversity in the population and preventing from premature convergence. The latter is the size of the

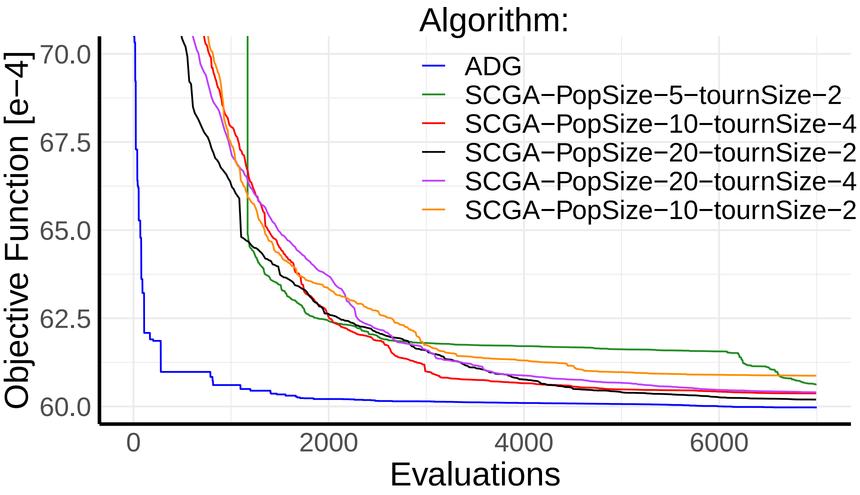


Figure 6.3.1: Best found solution history. All the instances are compared to the ADG run.

pool of candidates tested during a selection operation. This size heavily impacts on the diversity in the mating pool for the *crossover* operator [137]. The smaller the *tournament size* is, the higher the chances for a low-fitness candidate to bequeath its information to candidates of the next population. Therefore, the higher the *tournament size*, the higher the pressure to select only the high-fitness candidates, with the possible drawback of letting the population collapse toward a local optimum. The values tested for the *population size* are $[5, 10, 20]$ and for the *tournament size* are $[2, 4]$. However, the combination with *population size* equal to 5 and *tournament size* equal to 4 cannot be run because of the tournament selection implementation in SCGA. The number of maximum function evaluations has been set to 7000 to match the reference run ones. The remaining parameters of SCGA have been left to the default values [131].

The reference run is made using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) available in the Adaptive Genetic algorithm (ADG) optimization library (see Section 2.1). This library has already been tested for many aerodynamic design optimization problems facing multi-element airfoils [124, 125]. The parameters for this optimization algorithm are the maximum number of allowed evaluations, equal to 7000, the population size ($\lambda = 4$), and the initial standard deviation ($\sigma = 0.1$). For each setting of SCGA, 50 independent runs have been performed to have statistically significant results.

All the tested instances perform adequately well, as shown in Fig. 6.3.1, and are summarized in Table 6.3.1. Particularly, the values of the best solution found of each instance of SCGA are compared in terms of the relative difference between the minimum, the mean and the maximum value found in the 50 independent runs with respect to the reference value. The algorithm validation step is positive, also taking into account the differences between the two classes of optimization algorithms considered. In addition, the initial population of the ADG reference run was built by randomly mutating the solution representing the nominal airfoil (NACA 2412), whereas with SCGA a totally random initial population is generated.

Population Size	Tournament Size	Min diff [%]	Mean diff [%]	Max diff [%]
5	2	0.17	1.08	7.44
10	2	0.21	0.66	1.50
10	4	0.33	1.07	3.00
20	2	0.14	0.72	1.68
20	4	0.12	1.84	20.9

Table 6.3.1: Results of the optimization runs.

Thus, after validating SCGA, the robustness of the two main ingredients needed to carry on an aerodynamic design optimization problem (the CFD flow solver in Section 6.2.1 and, here, the optimizer) has been proven.

6.4 Optimization design variables

One of the many challenges of mixed-discrete variable-size optimization problems is to implement the appropriate mapping between the design variables of the problem. The HLD optimization degrees of freedom are the type of flap and the settings of each flap component (position in the 2-dimensional space and rotation). The settings of the slat are also considered. The optimizer receives this information encoded as a set of design variables.

The SCGA allows the definition of hierarchical problem formulations. In this problem, the top of the hierarchy is composed of four independent variables: one for indicating the flap typology (SF, DS_1, DS_2, DS_3 and TS) and three variables indicating the slat settings, namely $\Delta\theta_S$, ΔX_S, and ΔY_S. Specifically, they indicate the deviance of a proposed position from the nominal one. The second level of the hierarchy is composed of all the variables determining the flap's settings. They are all dependent on the flap typology variable, and its value determines their presence. They are three in the case of the single flap, six in the case of the double-slotted flap (whatever the type), and nine in the case of the triple slotted flap. A graphical representation of the hierarchy of the variables is shown in Fig. 6.4.1. *Level* 1 is constituted by the independent variables (referring to Table 6.4.1, the variables 1-4). In addition, *Level* 2 is made by all the remaining variables dependent on variable 1. Solid lines indicate variables that are present in all the candidates. In contrast, dashed lines indicate variables that can also not be present.

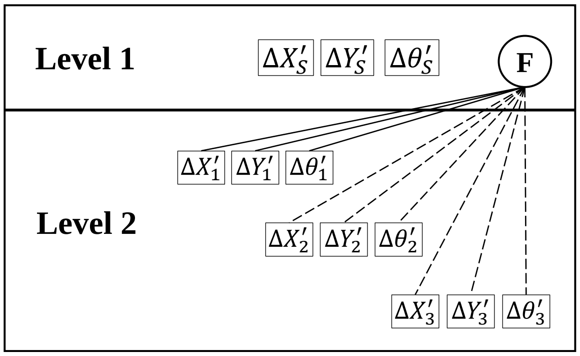


Figure 6.4.1: Hierarchy of the High-Lift Device topology design optimization variables.

Furthermore, the description of each variable, its number for identifying it in the hierarchy, its type, the possible value that it can assume (as possibility or bounds), and the dependency of any variables to it are given in Table 6.4.1.

Concerning the encoding of the variables indicating the settings of the flap elements, the most straightforward and *naive* problem formulation would be to use the same approach used for encoding the settings of the slat. However, this would imply significant difficulties from the optimization perspective. For example, let us suppose that one candidate undergoes the mutation operator and only the variable defining the type of the flap is mutated. A desirable outcome would be an HLD with different elements but a similar overall topology. However, the example of this transformation is given in Fig. 6.4.2a. Here, two possible outcomes of the mutation operating on the flap type of a solution adopting a double-slotted DS_2 flap are shown. As one can see, by using *naive* problem formulation (*dashed blue* airfoil components), the mutation of one single variable would result

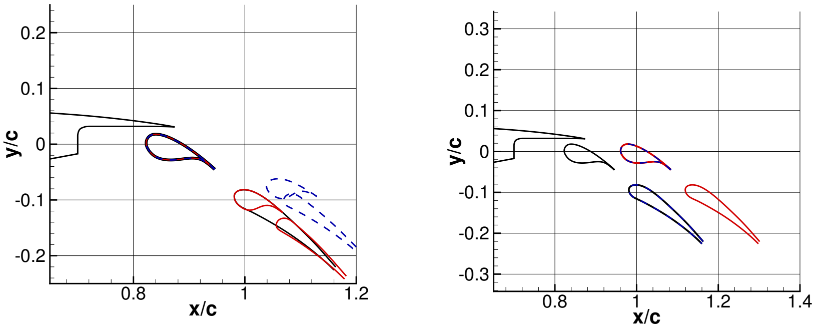
Description	N	Variable type	Possibilities		Dep
Flap type	1	Nominal discrete	$[SF, DS_1, DS_2\ ,$ $DS_3, TS]$		$[5 - 13]$
Description	N	Variable type	Lower Bound	Upper Bound	Dep
$\Delta\theta_S'$	2	Continuous	-15	15	–
$\Delta X_S'$	3	Continuous	-0.1	0.1	–
$\Delta Y_S'$	4	Continuous	-0.025	0.025	–
$\Delta\theta_1'$	5	Continuous	-15	15	–
$\Delta X_1'$	6	Continuous	-0.2	0.2	–
$\Delta Y_1'$	7	Continuous	-0.05	0.05	–
$\Delta\theta_2'$	8	Continuous	-15	15	–
$\Delta X_2'$	9	Continuous	-0.1	0.1	–
$\Delta Y_2'$	10	Continuous	-0.025	0.025	–
$\Delta\theta_3'$	11	Continuous	-15	15	–
$\Delta X_3'$	12	Continuous	-0.1	0.1	–
$\Delta Y_3'$	13	Continuous	-0.025	0.025	–

Table 6.4.1: Problem formulation.

in a considerable unwanted change in the HLD overall topology. The resulting topology would even be unfeasible.

Another example is depicted in Fig. 6.4.2b. In this case, the variable determining the x-coordinate of the first flap element is mutated. If the *naive* problem formulation is used (*dashed blue* airfoil components), the second component of the resulting flap completely loses its alignment with the first component, leading the first flap component to be over the second. Therefore, the *naive* problem formulation may lead very often to undesirable outcomes because a small variable perturbation can induce considerable topology transformations.

In light of these considerations, another problem formulation has been proposed. Its backbone is to make the flap elements be logically connected to the preceding elements in terms of rotation and translation. The absolute difference between the proposed solutions



(a) From double slotted type 1 (—) to triple slotted.

(b) Translation of one element. Initial position (—).

Figure 6.4.2: Examples of the mutation operator adopting the *naive* (- - -) and the proposed (—) formulations.

and the nominal solutions is described as follows:

$$\boldsymbol{\Delta X} = \begin{cases} \Delta X_S = \Delta X_S^{'} \\ \Delta Y_S = \Delta Y_S^{'} \\ \Delta\theta_S = \Delta\theta_S^{'} \\ \begin{cases} \Delta X_i = \Delta X_i^{'} \\ \Delta Y_i = \Delta Y_i^{'} \\ \Delta\theta_i = \Delta\theta_i^{'} \end{cases} & \text{if } i = 1 \\ \begin{cases} \Delta X_i = \Delta X_i^{'} + \Delta X_{i-1}^{'} \\ \qquad + RX_{i-1} - X_i^{LE} \\ \Delta Y_i = \Delta Y_i^{'} + \Delta Y_{i-1}^{'} \\ \qquad + RY_{i-1} - Y_i^{LE} \\ \Delta\theta_i = \Delta\theta_{i-1}^{'} + \Delta\theta_i^{'} \end{cases} & \text{if } i \neq 1. \end{cases} \tag{6.3}$$

Where RX_{i-1} and RY_{i-1} are the translation due to the rotation of the previous element:

$$\begin{bmatrix} RX_{i-1} \\ RY_{i-1} \end{bmatrix} = \begin{bmatrix} X_{i-1}^{LE} \\ Y_{i-1}^{LE} \end{bmatrix} + \begin{bmatrix} \cos(\Delta\theta_{i-1}^{'}) & -\sin(\Delta\theta_{i-1}^{'}) \\ \sin(\Delta\theta_{i-1}^{'}) & \cos(\Delta\theta_{i-1}^{'}) \end{bmatrix} \begin{bmatrix} d_X \\ d_Y \end{bmatrix} \tag{6.4}$$

and d_X and d_Y are the distance between the leading edges of the i-th and $i-1$-th elements along the two axes:

$$\begin{aligned} d_X &= X_{i-1}^{LE} - X_i^{LE} \\ d_Y &= Y_{i-1}^{LE} - Y_i^{LE}. \end{aligned} \tag{6.5}$$

From Fig. 6.4.2, the impact of this new problem formulation (*red* components) on the mutation operations can be appreciated. In the example shown in Fig. 6.4.2a, if the proposed formulation is used, the resulting airfoil presents somewhat similar overall settings to the original one. Therefore, the changes are only due to the different type of flap adopted. In the example shown in Fig. 6.4.2b, the relative alignment between the flap's elements is preserved. In particular, the change in the topology is accompanied by a translation along the x-axis of the HLD as a whole.

By using this new variable definition, the occurrence of unfeasible solutions is prevented; thus, the waste of computational resources is also avoided.

6.5 Deterministic optimization

A deterministic mixed variable-size aerodynamic design optimization problem of a multi-element airfoil has been solved. The baseline configurations are those shown in Section 6.2.2, and the working conditions are $M_\infty = 0.2$ and $Re_c = 5 \times 10^6$ at a fixed angle of attack $\alpha = 21.29^\circ$. The goal of the optimization is the improvement of the multi-element airfoil performance, maximizing lift coefficient, by selecting the most suitable flap typology and the settings (position and rotation) for the flap elements, as for the slat.

The quantity of interest to be optimized is the lift coefficient, c_l. However, the failure of the grid generation or the non-convergence of the CFD solver have to be treated. This latter is considered to happen when the difference between the up-to-date c_l and the mean lift coefficient in the last 1000 iterations of the CFD solver $(c_{l_{AVG}})$ is lower or equal to 0.005 $\left(|c_l - c_{l_{AVG}}| \leq 0.005\right)$. CFD analysis of configurations close to stall or even post-stall is notoriously a complex task, and the results present a high margin of uncertainty even in the case of fully converged solutions. Consequently, it is not wise and appropriate to use CFD results that have not reached full convergence because they would introduce an unacceptable margin of uncertainty and imprecision. The aerodynamic computational chain returns an indication of the occurrence of one of these errors as an error flag. This flag is 1 in case of error, 0 in case of a successful evaluation. Only successful candidates can return a correct prediction of c_l. Therefore, the objective function is reformulated by introducing a step penalty as follows:

$$Q(\mathbf{\Delta X}) = \begin{cases} c_l & \text{if } Error = 0 \\ p & \text{if } Error = 1, \end{cases} \tag{6.6}$$

with $p = -1000$ (to be noted that it is a maximization problem and values $\mathcal{O}(1)$ are expected). Hence, the optimization problem reads:

$$\max Q(\mathbf{\Delta X}) \tag{6.7}$$

Finally, the optimization algorithm used is SCGA. A list of the algorithm parameters is given in Table 6.5.1.

Table 6.5.1: SCGA parameters

size	*tournamentSize*	*maxEvaluations*	*elitism*	*mutRate*	*probability*
19	3	1550	1	0.05	$[3, 1, ..., 1]/16$

Specifically, in Table 6.5.1, *size* refers to the population size, *mutRate* is the mutation rate, and *probability* is the probability of a design variable to be selected by the operators. The remaining parameters of SCGA have been left as default [131]. The optimization stopped when the available computational budget was exhausted. In particular, this happened after 86 iterations.

6.5.1 Deterministic optimization results

The most notable results are analyzed here. Firstly, the convergence history of the best solution found is commented. From Fig. 6.5.1, it can be seen that, starting from a randomly generated population, only 355 evaluations were needed to find an HLD topology that performs better than the 30P-30N airfoil, and 505 to improve DS_1 baseline airfoil. The optimum airfoil is a double-slotted of type 1 (DS_1). In particular, the airfoil type DS_1 prevailed as the best configuration after a short initial phase of exploration. The evolution of the presence of the different flap types in the populations is represented in Fig. 6.5.2. Here, it is shown that all the configurations are fairly equally represented in the initial population. Later, the good performance of candidates having the flap type DS_1 makes that a

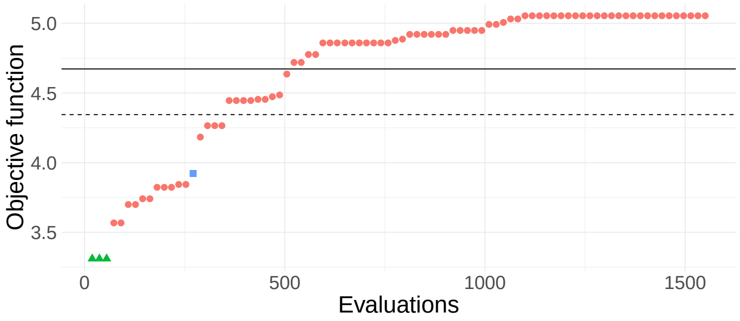


Figure 6.5.1: Best found objective function history. Airfoil types: DS_1 (●), TS (■), and DS_3 (▲). Lift coefficient of the baseline airfoil of type DS_1 (—) and c_l of the 30P-30N airfoil (- - -).

significant part of the candidates assumes this airfoil type. However, the other configurations never disappear; thus, the effect of the configurational variable is investigated during all the optimization. Moreover, it must be specified that 41% of the evaluations returned

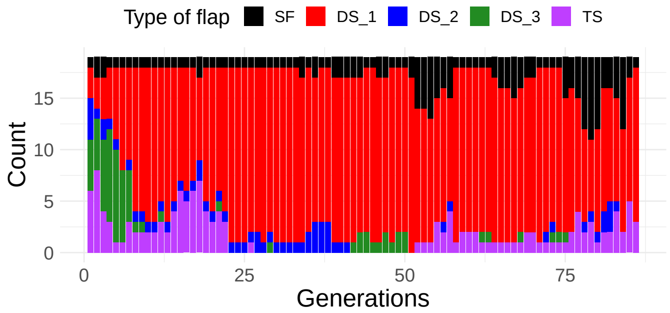


Figure 6.5.2: Presence of candidates with different airfoil typology in every generation.

$Error$ = 1. The majority of these cases were due to the no convergence of the CFD analysis. This behavior is typical when studying near stall conditions. Although the failure rate was high, the optimizer was able to find a configuration improving the airfoil performance at the studied working conditions. This evidences the potential of SCGA.

In addition, Fig. 6.5.3 shows the history of each variable of the best element in each population. If variables $\Delta X_3^{'}$, $\Delta X_3^{'}$ or $\Delta \theta_3^{'}$ are missing, then the best solution is not a triple slotted configuration. Moreover, it is appreciated that part of the optimum values were rapidly identified, others, i.e., $\Delta X_1^{'}$ and $\Delta \theta_2^{'}$, changed significantly during the whole

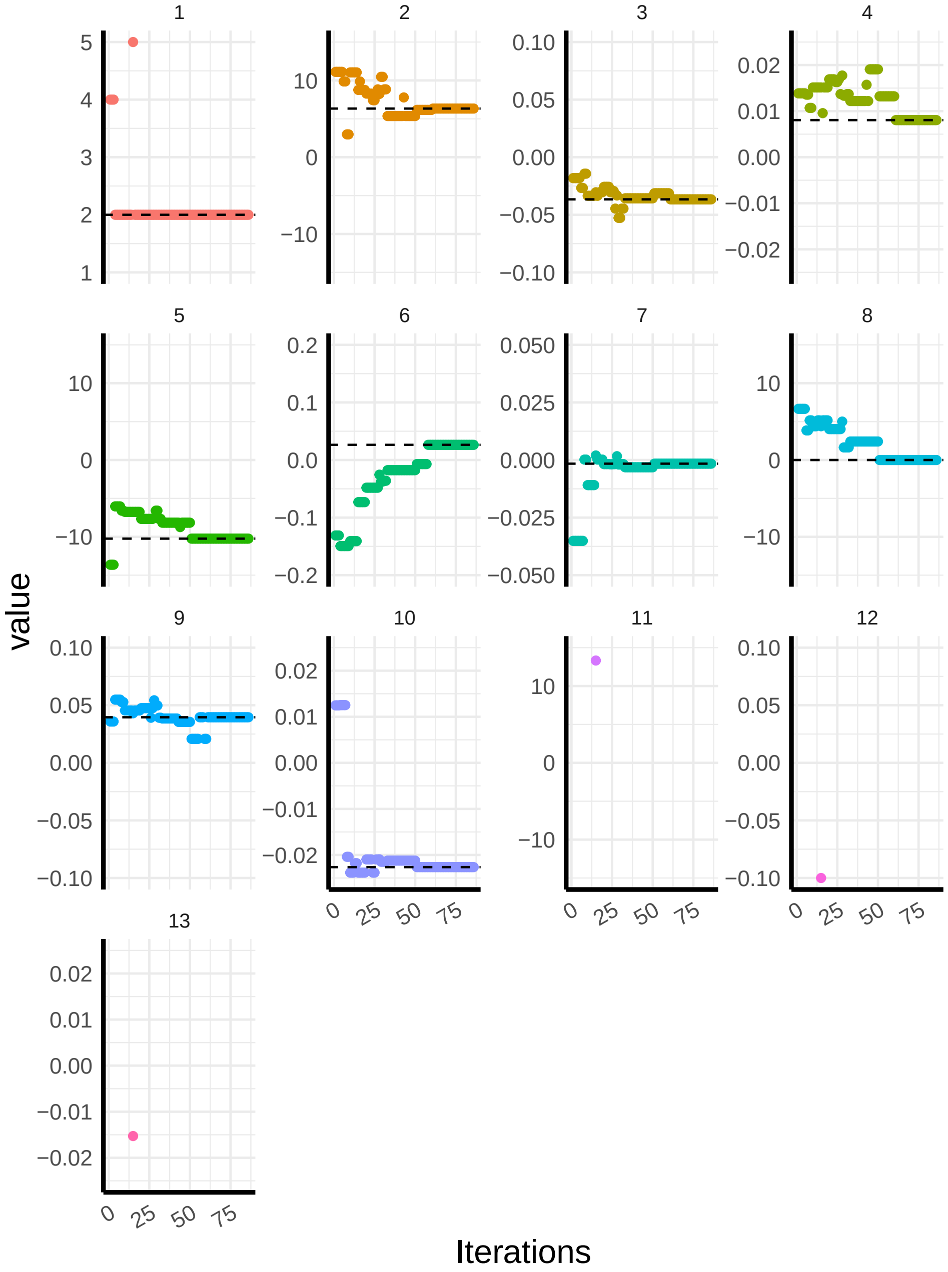


Figure 6.5.3: Best found history. Variables enumeration follow the one given in Table 6.4.1.

optimization.

The deterministic optimal multi-element airfoil is depicted in Fig. 6.5.4 and is compared with the baseline double-slotted airfoil of type one (DS_1) and the starting airfoil, the 30P-30N airfoil. The lift coefficient of the optimized airfoil at $\alpha = 21.29^\circ$ is $c_l = 5.04838$. Therefore, at this angle-of-attack, the airfoil presents an increment of 16.18% with respect to the 30P-30N airfoil and an 8.04% with regard to the baseline double-slotted airfoil of type one (DS_1). Regarding the increase of maximum lift coefficient, the deterministic optimum airfoil provides an increment equal to 14.82% and 7.67%, respectively. Moreover, the aerodynamic performance of the optimal airfoil must be contrasted to the baseline airfoils aforementioned. Thus, the polars of the three airfoils are given in Fig. 6.5.5a and Fig. 6.5.5b. The comparison among the three configurations shows a clear improvement, in terms of lift coefficient, for the optimal configuration.

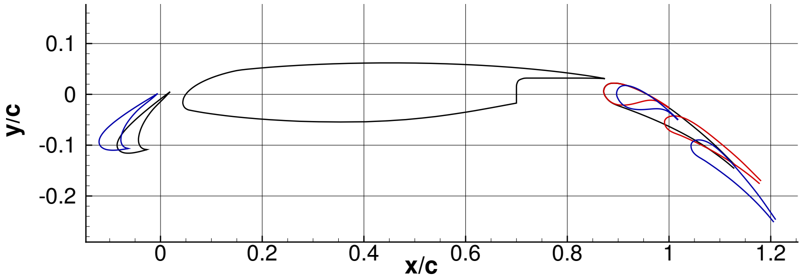
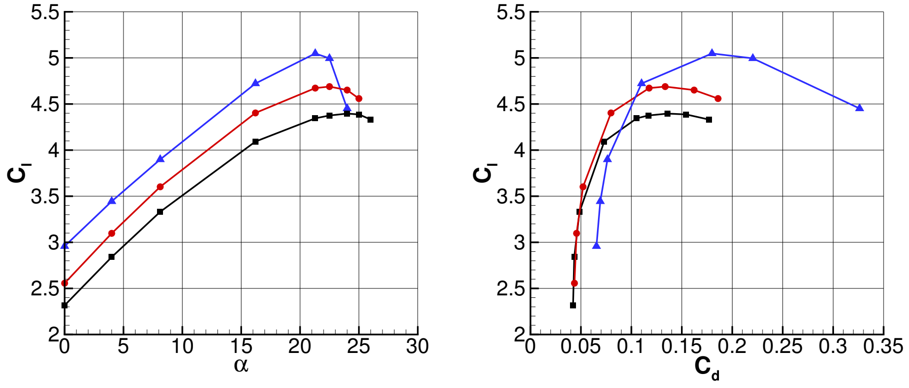


Figure 6.5.4: Airfoil comparison between the 30P-30N (—), the baseline airfoil of the double slotted flap of type one (DS_1) (—), and the optimum deterministic airfoil (—).

Also, the optimized airfoil reaches the stall condition at a lower angle of attack. Notably, the optimal airfoil shows a pronounced drop of c_l at an angle-of-attack equals to 24° that it is not present in the other two airfoils. This behavior could compromise the aircraft performances in landing conditions. Therefore, the landing design condition should be explicitly introduced in the optimization problem to overcome this problem. Indeed, the overcome of this undesirable behavior is the objective of the optimization solved in Section 6.6.

Finally, in Fig. 6.5.6, the flow fields of the pressure coefficient, c_p, of the double-slotted airfoil of type 1 (DS_1) and the deterministic optimum airfoil are given for the optimization working conditions $(M_\infty = 0.2,\ Re_c = 5 \times 10^6,$ and $\alpha = 21.29^\circ)$. Besides, in Fig. 6.5.7, the pressure coefficient and the friction coefficient, c_f, at the body surface of both airfoils are compared. Observing Fig. 6.5.4, it is seen that the main variation between



(a) c_l vs α **(b)** c_l vs c_d

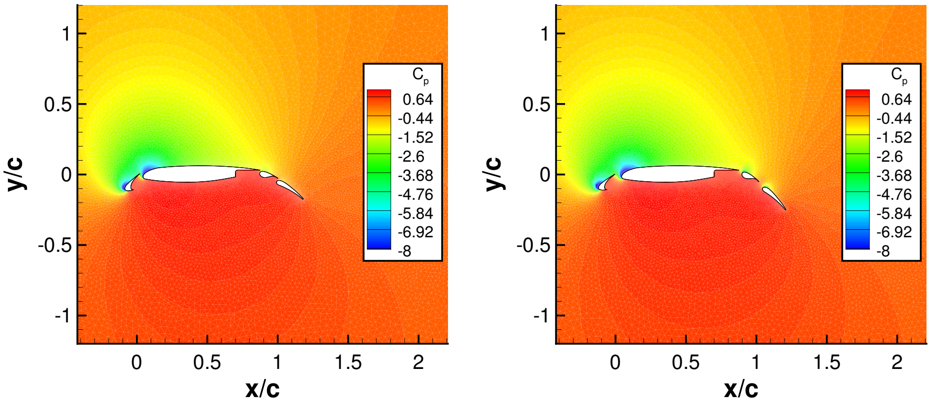
Figure 6.5.5: Comparison of the aerodynamic performance between the 30P-30N airfoil (–■–), the baseline DS_1 airfoil (–●–), and the deterministic optimum airfoil(–▲–). M_∞ = 0.2 and Re_c = 5×10^6.

the airfoils is the upstream shift of slat and the downstream movement of flaps that are aimed to increase the actual surface of the airfoil, thus increasing the lift coefficient. Additionally, another output of the optimization is the clockwise rotation of the flap elements. The rotation produces a higher expansion on the upper surface of the flap components producing a rise on the c_l. This is appreciated in Figs. 6.5.6 and 6.5.7a. Moreover, the c_p contributions of the main airfoil component and the slat are similar for the DS_1 and the deterministic optimum airfoil.

The drag coefficient of the deterministic optimum is remarkably penalized, and an increase of 53% is obtained. This was an expected result because, inevitably, an increase in lift leads to a rise in drag, and there was not an imposed constraint on the c_d to control its increment. Moreover, the growth in drag is also observed in Fig. 6.5.7. As mentioned, Fig. 6.5.7a shows a greater c_p contribution of the flap components, and in Fig. 6.5.7b, higher skin friction values on the upper side of the flap elements are appreciated.

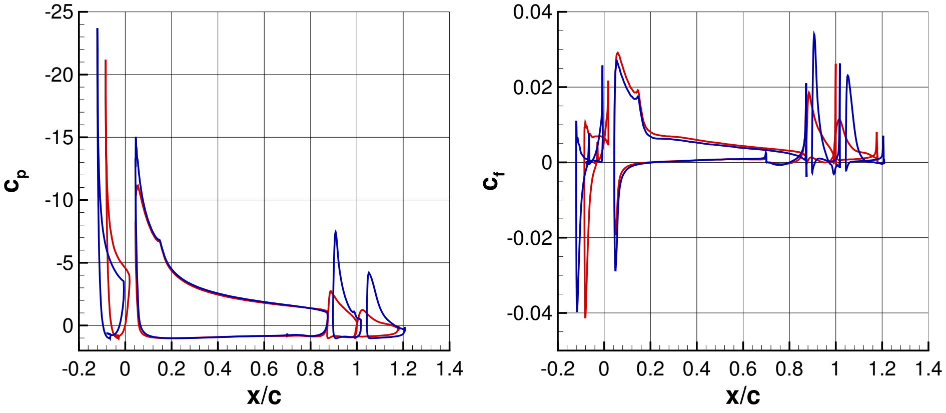
6.6 Machine learning assisted robust optimization

Previously, a deterministic optimization was performed where the objective function was limited to the maximization of the lift coefficient generated by the HLD at a given angle-



(a) Double slotted type 1 (DS_1). **(b)** Deterministic optimum.

Figure 6.5.6: Pressure coefficient c_p flow field for the double slotted airfoil of type 1 and the deterministic optimum. $M_\infty = 0.2$, $Re_c = 5 \times 10^6$, and $\alpha = 21.29°$.



(a) Pressure coefficient c_p. **(b)** Friction coefficient c_f.

Figure 6.5.7: Pressure c_p and friction c_f coefficients on the body surface comparison for the double slotted airfoil of type 1 DS_1 (—) and optimum deterministic (—) airfoils. $M_\infty = 0.2$, $Re_c = 5 \times 10^6$, and $\alpha = 21.29°$.

of-attack regardless of its drag, weight, costs and practical construction feasibility. The use of CFD for predicting the aerodynamic performances of the airfoil is costly. However, these kinds of analyses are affected by several uncertainties. In particular, the uncertainty affecting the operational conditions, such as the angle-of-attack of the airfoil, must be considered since a small change on the angle-of-attack may cause an abrupt stall of the airfoil. On the other hand, the introduction of uncertainty variables can make standard optimization unaffordable in terms of computational costs. Thus, a method to reduce the CPU time is required. Therefore, the use of surrogate models is introduced to optimize the usage of computational resources further.

The general idea is to adopt the search algorithm employed in the previous work in an Efficient Global Optimization (EGO)-like optimization framework [108] enhanced by data-driven models. Particularly, a classifier is used to predict the feasibility of the proposed candidate. Moreover, the computational cost is further reduced by means of a quadrature approach that makes the uncertainty quantification relatively inexpensive.

The multi-element airfoils used in the deterministic optimization are employed as baseline configurations for the optimization under uncertainty presented here in. The objective of the optimization is to design an HLD that maximizes the lift coefficient considering the angle-of-attack as an uncertain parameter. Therefore, a quadrature approach [141] is implemented to quantify the uncertainty. Hence, instead of maximizing the c_l directly, a statistical measure of it is maximized.

6.6.1 Machine Learning Assisted optimization

To solve this optimization problem, a method that accounts for the stochastic and expensive nature of the problem is required. Hence, a Machine Learning Assisted Optimization approach, which stems from the EGO strategy [108], is used. Specifically, this consists of an iterative approach as shown in Algorithm 2.

Algorithm 2 Machine Learning Assisted Optimization

```
$t = 0$. $P(t)$ = SetInitialPopulation().
Evaluate($P(t)$) on $f$.
while not TerminationCriterion() do
    Use $P(t)$ to build a model $M(t)$ and a classifier $C(t)$.
    Define objective function $F$ based on $M(t)$ and $C(t)$.
    $P'(t+1)$ = GlobalSearch($F(t)$).
    Evaluate($P'(t+1)$) on $f$.
    $P(t+1) = P(t) \cup P'(t+1)$.
    $t = t + 1$.
end while
```

The first step in Algorithm 2 (Line 1) is the determination of the initial data-set that is used to train the first surrogate model M and classifier C. Once the first data-set has been created and observed (in Line 2), it is used to train a surrogate model M to replicate the behavior of the objective function, and a classifier C to distinguish feasible and unfeasible candidates Line 4. A composition of C and M yields the objective function that is optimized to propose a new promising solution in Lines 5 and 6. Finally, the data-set is enlarged with the new proposed point and its observation on f Lines 6 and 8. Further details are given in Section 6.6.3, and a representation can be found in Fig. 6.6.1.

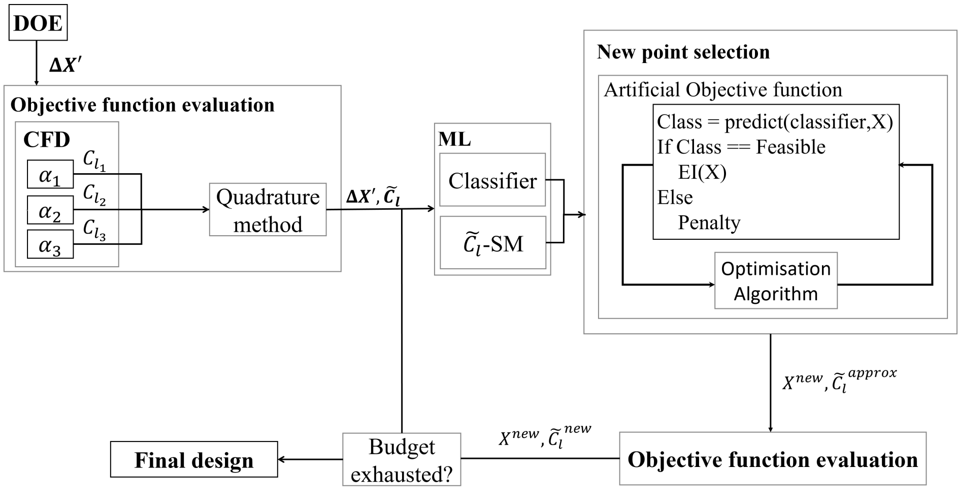


Figure 6.6.1: Machine Learning Assisted Optimization Flowchart.

First, a Design of Experiment (DOE) is created and observed. The observation, objective function evaluation, consists of computing the c_l coefficient of a candidate given three different angles of attack through high fidelity CFD analyses. These values are used to estimate a robust measure of lift coefficient as described in Section 6.6.3. Once the data are collected, two data-driven models are trained. One aims to mimic the objective function, one at identifying feasible from unfeasible solutions. A combination of these two is used as a target of the optimization algorithm SCGA to find a new promising HLD design. The actual performance of the proposed point is evaluated and stored. If the evaluation budget is not exhausted, the process is repeated. Else, the best solution found is returned as the optimal final design.

Surrogate Model

To reduce the computational effort and efficiently pursue the identification of optimal solutions, optimization is often enhanced by the use of surrogate models. These models mimic and replace the original expensive objective function providing a cheap target for the optimization routines. In this research, a Gaussian process model (or Kriging) [142] has been chosen as a surrogate model. An appreciated feature of Kriging is that it also provides an estimate of its own prediction uncertainty. This estimate can be used to balance exploration and exploitation by computing the Expected Improvement (EI) of candidate solutions [143]. Kriging assumes that data follow a multivariate Gaussian distribution, where errors are spatially correlated. This is encoded within a kernel function. To account for the hierarchical nature of the search space, the Wedge kernel [144] is used. This kernel employs a mapping function. With any standard kernel $\mathrm{k}(\mathbf{x}, \mathbf{x})$, the mapping function can be employed such that $\mathrm{k}_{wedge}(\mathbf{x}, \mathbf{x}) = \mathrm{k}(\mathrm{h}(\mathbf{x}), \mathrm{h}(\mathbf{x}))$, where $\mathrm{h}(\mathbf{x})$ applies h_i to each dimension x_i. The mapping function of the Wedge kernel is

$$\mathrm{h}_i(\mathbf{x}) = \begin{cases} \begin{bmatrix} 0 & 0 \end{bmatrix}^T, & \text{if } \delta_i(\mathbf{x}) = \text{false} \\ \begin{bmatrix} \theta_{1,i} + v\left(\theta_{2,i}\cos(\rho_i) - \theta_{1,i}\right) & v\theta_{2,i}\sin(\rho_i) \end{bmatrix}^T, & \text{otherwise,} \end{cases} \tag{6.8}$$

with the scaled variable value $v = (x_i - l_i)/(u_i - l_i)$, and lower / upper bounds l_i, u_i in each dimension.

From a geometrical view, this function maps the variable values in each dimension of the input vector **x** to a triangular shape (the Wedge) in a two-dimensional space. If the hierarchical variable in that dimension is inactive ($\delta_i(\mathbf{x}) = \text{false}$), then x_i is mapped to the origin $\left(\begin{bmatrix} 0 & 0 \end{bmatrix}^T\right)$. Else, x_i is mapped to a line segment. The parameters $\rho_i \in [0, \pi]$, $\theta_{1,i} \in \mathbb{R}^+$, and $\theta_{2,i} \in \mathbb{R}^+$, specify the angle and two adjacent side lengths of a triangle (spanned by the origin, and the line segment).

When training a standard Kriging model, it is assumed that the uncertainty in the already sampled locations is zero. However, this does not hold in noisy problems in the presence of uncertainty. One way to account for noise is to introduce the so-called nugget effect. This essentially adds a constant value η to the diagonal of the kernel matrix. The parameter η is determined by Maximum Likelihood Estimation. The nugget effect enables the model to regress the observed data, and hence smoothens noisy observations. Furthermore, it may now produce a non-zero estimate of the uncertainty at observed locations.

Classifier

In the developed framework, the surrogate model is fed only by solutions that did not report any error. Thus, the landscape is not compromised by artificially assigned penalty values. Further details about the types of errors that might occur are given in Section 6.6.3. In addition to the surrogate model, a classifier is used to predict and filter out candidates that may produce an error in the CFD analysis. Hence, the unfeasible regions are, in principle, detected and excluded in the new point selection step (Fig. 6.6.1).

The classification method adopted is the Random Forest model [145, 146]. This type of model ensembles a learning method for classification that operates by constructing a multitude of decision trees at training time. When predicting, the ensemble outputs the class that is the mode of the classes predicted by the individual trees.

6.6.2 Quadrature Approach for Uncertainty Quantification

In this work, among all the possible uncertain parameters, only the uncertainty of the angle of attack has been considered. Classically, Monte Carlo (MC) methods [147] are used for uncertainty quantification. To accurately obtain the probabilistic distribution by using MC methods, a large number of samples is required. Considering that only one uncertain variable is introduced in this optimization design problem, a different approach was thought to be more suitable. Particularly, the integral of the c_l coefficient over the interval of angles of attack $\left[21.29^{\circ},\ 24^{\circ}\right]$ has been adopted as the measure of interest. Indeed, it is a metric that expresses the overall quality in the whole range of the uncertain parameter. To estimate the integral, the Simpson quadrature rule [141] has been used. One derivation replaces the integrand $f(x)$ by the quadratic polynomial (i.e. parabola) which takes the same values as $f(x)$ at the end points a and b and the midpoint m as follows:

$$P(x) = f(a)\frac{(x-m)(x-b)}{(a-m)(a-b)} + f(m)\frac{(x-a)(x-b)}{(m-a)(m-b)} + f(b)\frac{(x-a)(x-m)}{(b-a)(b-m)}. \tag{6.9}$$

One can show that

$$\int_a^b P(x)dx = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right], \tag{6.10}$$

introducing the step size $h = (b - a)/2$ can be rewritten as

$$\int_a^b P(x)dx = \frac{h}{3}\bigg[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\bigg]. \qquad (6.11)$$

Therefore, the quantity of interest can be estimated by computing only three c_l values at the extremes and midpoint of the considered range of the angle of attack.

6.6.3 Problem Formulation

The goal of the presented study is to modify the airfoil topology of a baseline HLD through a Machine Learning Assisted Optimization (MLAO) framework to improve the starting performance. The three-elements airfoil McDonnell Douglas (MDA) 30P-30N [148, 149] has been adopted as a baseline.

For the present robust design optimization problem, the working conditions of the multi-element airfoil are $M_\infty = 0.2$ and $Re_c = 5 \times 10^6$ at a range of angles-of-attack $\alpha = [21.29^\circ, 24^\circ]$. The goal of the optimization is to find the multi-element airfoil configuration which guarantees the best performance (maximum estimation of the integral of the c_l coefficient) in the full range of working conditions, by selecting the most suitable flap type and the settings (position and rotation) for the flap elements, as for the slat. Hence, the aim of the design optimization problem is to improve the stall performance of the multi-element airfoil with respect to the deterministic optimum found in Section 6.5.1.

Original Objective function

The purpose of this work is to maximize a robust measure of the lift coefficient: its integral with respect to the uncertain angle of attack over its range. The method presented in Section 6.6.2 has been used to compute a robust measure of the lift coefficient $\tilde{c_l}$.

With that said, using Eq. (6.11), the adopted objective function can be defined as:

$$\tilde{c_l} = \int_{21.29^\circ}^{24^\circ} 0.5\bigg[c_l(21.29^\circ) + 4c_l(24^\circ - (24^\circ - 21.29^\circ)/2) + c_l(24^\circ)\bigg]. \qquad (6.12)$$

When dealing with CFD analysis, two errors might occur. Firstly, the generation of the computational grid might be impossible given certain HLD configurations such as intersections between airfoil elements. Secondly, the CFD analysis might not converge. The convergence of CFD is considered to happen when the difference between the up-to-date

c_l and the mean lift coefficient in the last 1000 iterations of the CFD solver $(c_{l_{AVG}})$ is lower or equal to 0.005 $\left(|c_l - c_{l_{AVG}}| \leq 0.005\right)$. In addition, it must be mentioned that Computational Fluid Dynamic analysis of configurations close to stall or even post-stall is notoriously a complex task, and the results present a high margin of uncertainty even in the case of fully converged solutions. Consequently, it is not wise and appropriate to use CFD results that have not reached full convergence because they would introduce an unacceptable margin of uncertainty and imprecision.

The aerodynamic computational chain returns an indication of the occurrence of one of these errors as an error flag.

This flag is equal to -1 when the mesh is not generated, equal to -2 when the convergence of the flow field is not achieved, and equal to 0 in the case of all successful evaluations. Only candidates successfully tested at all angles of attack return a correct estimation of $\tilde{c_l}$.

Therefore, the objective function is reformulated as follows:

$$Q(\mathbf{\Delta X}) = \begin{cases} \tilde{c_l}, & \text{if } ErrorFlag = 0 \\ ErrorFlag, & \text{if } ErrorFlag \neq 0, \end{cases} \tag{6.13}$$

where $ErrorFlag$ refers to the three possible values returned by CFD analyses used to compute $\tilde{c_l}$. Referring to Algorithm 2, it can be said that $Q \equiv f$.

Artificial Objective function

In the presented framework, the search for optimal solutions relies on the application of an optimizer to an *artificial* function that mimics the behavior of the original function. In traditional Surrogate Based Optimization (SBO), this *artificial* function is the prediction of a surrogate model trained with all the collected data.

Here, it is determined by the predictions of two distinct models. The first, the surrogate model M, aims at the prediction of $\tilde{c_l}$. The second, the classifier C, identifies feasible and unfeasible solutions. In particular, the classifier is trained to distinguish configurations leading to mesh creation failures, CFD no convergence, or correct c_l evaluations. This is based on the reported $ErrorFlag$. To train the models, all configurations needed to be mapped into a common rectangular data structure. Therefore, all the structured chromosome have been flattened, and the ones containing less than the maximum number of possible variables (single and double slotted configurations) have been padded to match the dimensionality of the triple slotted configuration.

Finally, the artificial objective function is reformulated by introducing a step penalty as follows:

$$Q_{artificial}(\mathbf{\Delta X}) = \begin{cases} EI_M(\mathbf{\Delta X}), & \text{if } C(\mathbf{\Delta X}) = 0 \\ p, & \text{if } C(\mathbf{\Delta X}) \neq 0, \end{cases} \tag{6.14}$$

with $p = -10$ (note, that it is a maximization problem and values $\mathcal{O}(1)$ are expected), $EI_M(\mathbf{\Delta X})$ is the EI computed by the $M(\mathbf{\Delta X})$ prediction model and $C(\mathbf{\Delta X})$ is the prediction of the classifier. Referring to Algorithm 2, $Q_{artificial} \equiv F$.

6.6.4 Optimization Setup

As a first step of the MLAO process, an initial design is generated. In this case, an initial DOE of 200 individuals composed of the 5 baseline configurations and 195 randomly generated configurations has been evaluated. Then, new random individuals have been generated until a total of 200 cases with successful grid generation have been proposed. The cases with failure on the grid generation are computationally inexpensive, hence ignored in the evaluations counting. The R `CEGO` package [150] is used to train the Kriging model, while the R `ranger` package [151] is used to train the classifier. The optimization has been stopped after 325 function evaluations (grid generation errors excluded) to assign to the DOE and optimization respectively 60% and 40% of the total computational budget. The new point selection step is entrusted to SCGA (also available as an R package). The list of adopted settings is given in Table 6.6.1. In particular, *size* is the population size, *mutRate* is the mutation rate, and *probability* is the probability of a design variable to be selected by the operators. The remaining parameters of SCGA have been left as default [152].

size	*tournamentSize*	*maxEvaluations*	*elitism*	*mutRate*	*probability*
200	3	50000	1	0.05	$[3, 1, ..., 1]/16$

Table 6.6.1: SCGA parameters

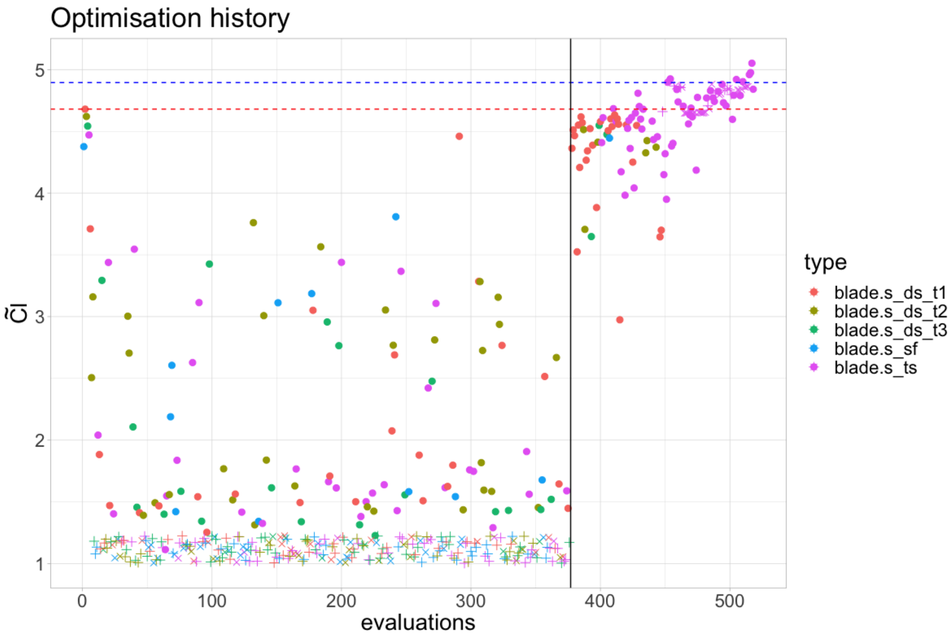


Figure 6.6.2: Optimization convergence history.

6.6.5 Results

The most relevant information coming from the obtained results are presented and discussed. Firstly, the convergence of the optimization process is provided in Fig. 6.6.2. Particularly, in Fig. 6.6.2, the filled dots depict the true value given by the CFD analyses, and the *crosses* and *plus* symbols indicate an artificial value assigned to the solutions in which, respectively, the grid generation or the convergence of the CFD solution was impossible. The solid black line separates the solutions composing the DOE and the ones proposed by the algorithm. The dashed red line indicates the best solution found in the initial design. Finally, the dashed blue line marks the reference solution obtained using the deterministic optimum found in Section 6.5.1. In addition, an important consideration, from this figure, is that achieving flow field convergence is a severe problem for SU2 flow solver. Notably, a considerable part of the DOE (21%) reported a convergence error in at least one simulation. Moreover, the largest part of the solutions randomly generated in the DOE (47%) proposes a multi-element airfoil that prevents the generation of the computational grid. Hence, only 32% of the cases have converged in all the three CFD cases performed. These results indicate how strongly the constraints shrink the feasible search space.

It is also worth noting that, in the initial design, only one randomly generated candidate performs comparably to the baseline solutions (firsts five in Fig. 6.6.2). This underlies the difficulty of the investigated optimization problem.

When the evaluation of the randomly generated points is terminated, the optimization process starts. Intuitively, in the very first iterations, the model's landscape attracts the optimizer towards the region of the best solution found so far. So the first points suggested are all characterized by the double-slotted topology of type 1 (DS_1). Then, the concentration of points in that region and the consequent decrease of the model's uncertainty leads the search into a more exploratory optimization phase. Configurations with different flap types alternate until a new best solution with triple slotted flap configuration (TS) is found. From that point on, the search mainly focused on exploiting this region of the search space. It must be highlighted that the triple slotted flap is the most challenging configuration to design because all the variables are active.

Another interesting observation regards the number of unfeasible solutions found in the optimization process. The largest part (67%) of the suggested points have been correctly classified as feasible. Notably, only 10% of the solutions led to an error in the computational grid generation. The remaining 26% reported a convergence error in at least one simulation. These results testify the overall high quality of the trained classifiers and show that it is relatively easy to identify configurations with unfeasible geometry in respect to the ones reporting CFD convergence anomalies. This behavior was expected since, the flow conditions to be analyzed (high angles of attack at stall conditions) are characterized by separated flows. Thus, representing an arduous task for any numerical flow solver.

In Fig. 6.6.3, the obtained robust optimum airfoil is compared with the deterministic optimum found in Section 6.5.1 and the baseline MD 30P-30N airfoil. Regarding the slat of the proposed airfoil, it has nearly moved with respect to the baseline, contrary to the deterministic that has enlarged the gap with the main body.

Looking at the optimal HLD resulting from this process, one can see how this outperforms the deterministic optimum found in Section 6.5.1(see Table 6.6.2). Notably, this is true for both the $\tilde{c_l}$ and $c_l(21.29^\circ)$ which was the objective of Section 6.5.1. This important achievement validates the MLAO framework and confirms the superiority of the methods based on data-driven models over plain optimization routines when very limited computational budgets are available.

The lift curves of each airfoil are provided in Fig. 6.6.4 to compare their performance.

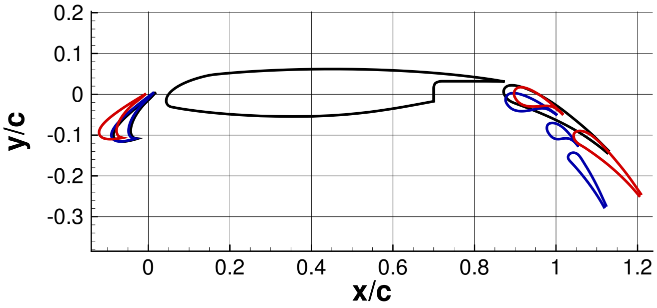


Figure 6.6.3: Comparison of the triple slotted robust optimum airfoil (—), the double slotted deterministic optimum (—), and the 30P-30N airfoil (—).

Both optimum airfoils provide higher c_l than the 30P-30N airfoil. Comparing the robust optimum airfoil and its baseline TS airfoil, an increase of c_l is observed over the complete range of angle-of-attack. The maximum lift coefficient of the robust airfoil is $c_{l_{max}} = 5.18466$ and it is obtained at $\alpha = 20^{\circ}$, as for the TS airfoil, providing an increment of maximum lift coefficient equal to 13.96%.

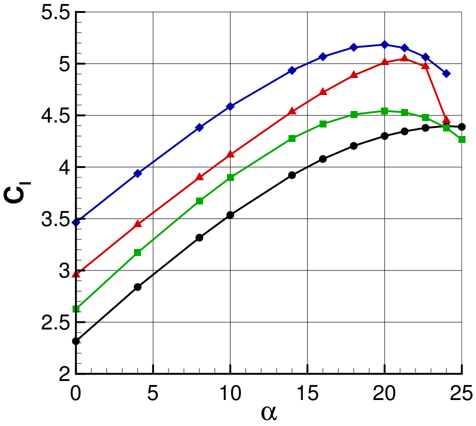
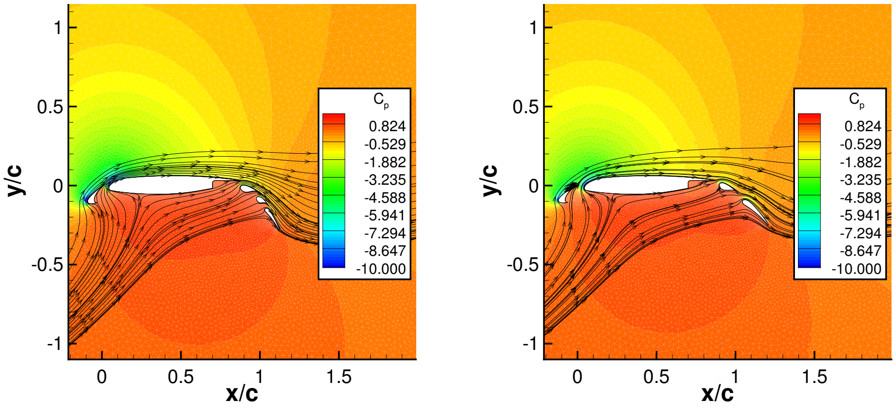


Figure 6.6.4: Polar curves (c_l vs α) for the 30P-30N airfoil (–●–), TS baseline airfoil (–■–), deterministic optimum airfoil (–▲–) found in Section 6.5.1, and robust optimum airfoil (–◆–).

Moreover, the performance of the deterministic and robust optimum airfoils must be compared. The maximum lift of the deterministic optimum airfoil is achieved at 21.29°, since it was the α at which the deterministic optimization was performed. However, it is observed that the c_l at 21.29° of the robustly optimized airfoil is higher than the one provided by the deterministically optimized airfoil. Thus, this demonstrates that the deterministic optimization gave a sub-optimal solution. Particularly, the robust airfoil has a maximum lift coefficient 2.7% higher. Furthermore, the $\tilde{c_l}$ value for the MD 30P-30N, the baseline TS, and robust and deterministic optimum airfoils is given in Table 6.6.2. Regarding the measure of robustness, $\tilde{c_l}$, the optimal robust airfoil found improves the performance of the deterministic optimum and the baseline MD 30P-30N respectively by a 3.1% and 15.4%.

Airfoil	$\tilde{c_l}$	$c_l(21.29^\circ)$	$c_l(22.645^\circ)$	$c_l(24^\circ)$
MD 30P-30N	4.376	4.345	4.379	4.396
Baseline TS	4.471	4.529	4.479	4.379
Deterministic optimum	4.899	5.048	4.973	4.451
Robust optimum	5.052	5.152	5.063	4.905

Table 6.6.2: Comparison of the $\tilde{c_l}$ and c_l obtained for the MD 30P-30N, the baseline TS and the deterministic and robust optimum airfoils.



(a) Robust optimum airfoil **(b)** Deterministic optimum airfoil

Figure 6.6.5: Pressure coefficient c_p flow field at 24° for the robust and deterministic optimum airfoil. The black lines are the streamtraces.

Finally, the pressure coefficient (c_p) flow field at 24° of the robust and deterministic airfoil is given in Fig. 6.6.5. The streamtraces provided for the deterministic optimum airfoil flow field show that there is a vortex over the slat of the airfoil. Although both airfoils are after maximum lift condition at 24°, the presence of the vortex determines a higher decrease of lift coefficient in the deterministic optimum. Hence, by performing the robust optimization this behaviour is prevented, and stall and post stall performances are improved.

6.7 Conclusions

In this chapter, a novel method to improve classical design strategies by including configurational decisions in an automated optimization tool is presented. The integration of these decisions, usually taken a-priori based on previous knowledge, could represent a step forward for the multidisciplinary design optimization field. The potential of the method has been shown in a High-Lift Device design problem, an application that presents many difficulties itself. Among the others, the considerable computational cost and the demanding constraints of the design problem represented a challenge. Furthermore, it was quite challenging to obtain an adequately converged Computational Fluid Dynamic solution for the majority of population elements due to the near-stall conditions at which the airfoil had to operate. An automatic estimation routine consisting of an aerodynamic computational chain based on the SU2 solver has been adopted to cope with these problems. This procedure has been coupled with the Structured-Chromosome Genetic Algorithm for optimizing the design of an HLD equipped with slat and flap devices based on the McDonnell Douglas 30P-30N airfoil.

In the first part of the chapter, the goal of the optimization is to deterministically identify the typology, for the multi-element airfoil that maximizes the lift coefficient at a given working near-stall condition ($M_\infty = 0.2$, $Re_c = 5 \times 10^6$, and $\alpha = 21.29°$). The results of the optimization show that, given an appropriate problem formulation, the optimizer was able to improve the reference performance (30P-30N airfoil) by 16.18% in a very limited number of function evaluations. Nevertheless, the optimizer performed well, although a significant amount of failed evaluations.

After solving the deterministic optimization, the HLD design problem is further analyzed, considering the angle of attack as an uncertain parameter. To cope with the consequent growth of the computational effort, a completely novel automated Machine Learning Assisted Optimization framework able to deal with configurational decisions has been developed. The presented Machine Learning Assisted Optimization makes use of data-driven models (Kriging, Random Forest) and the Simpson quadrature. The aim of the optimization was to obtain the topology for the multi-element airfoil that maximizes a robust measure of the lift coefficient, $\tilde{c}_l$. The results of the optimization show that the optimization process is able to improve the baseline 30P-30N performance, in terms of $\tilde{c}_l$, by 15.4% in few function evaluations.

For future research, some improvements are possible. Regarding the problem design variables, the shape of each element can be introduced using the approaches described in Section 2.2.1. This would lead to a significant increase in the number of variables and, consequently, of the problem complexity. Furthermore, the optimization could consider a higher number of performance indicators as the weight or the structural complexity. Besides, a control on the drag coefficient can be added to satisfy the constraints during take-off and climbing phases. This implementation would imply the resolution of multi-objective optimization problems. Moreover, better tuning of the CFD analysis stopping and convergence criteria may lead to an increasing number of successful evaluations.

Considering the robust design, a different quadrature approach able to estimate the mean and standard deviation of a distribution, as the one introduced by Padulo et al. [112], could be used in order to maximize the expected performance while maintaining the standard deviation within a limiting interval. Additionally, a rigorous analysis of the model and classifier performances should be done. Therefore, ensuring that shortcomings in these components of the Machine Learning Assisted Optimization framework do not compromise the execution of the overall process. On the other hand, it should be investigated whether different approaches are more suitable for the presented problem.

Finally, an assessment of the advantages offered by the presented approach with respect to traditional design strategies and optimizers need to be performed.

Conclusions

In this dissertation, the advantages of optimization under uncertainty with respect to the classical deterministic approach have been shown. In addition, state-of-the art risk measures, namely Value at Risk and Conditional Value at Risk, were introduced for aerodynamic shape design. Finally, several methods for reducing the computational cost of robust optimization, which is the characterizing aspect of this dissertation, have been investigated. Particularly, two approaches to reduce computational time were studied; speeding up the CFD evaluations and approximating the statistical measure.

A far-field method (based on entropy variations) for drag force evaluation has been implemented regarding the speeding up of the CFD runs. It allows the exclusion of the spurious drag contribution unavoidably introduced by the numerical scheme and it is particularly suited for solutions on coarse grids, where it provides a better prediction of the c_d with respect to the near-field method on the same grid. The far-field analysis was adopted for an inviscid test case and the optimization result was close to the best obtainable with a finer grid using the near-field calculation, whereas the optimization with the near-field evaluation of the drag force on the coarse grid did not provide a satisfactory result. Indeed, the very large spurious drag present on the coarser grid can hide the optimum region. However, attention must be paid to two aspects for the proper functioning of the far-field formula. Firstly, the grid can not be arbitrarily coarse but must be capable to capture all the physical aspects of the flow. Secondly, the far-field methods need an adequately solved field far from the body, implying that the solution must converge further. Indeed, in a converged solution, disturbances dissipate firstly close to the body surface and, later, far from it. Thus, additional iterations to the converged solution are needed with respect to a near-field approach for a good calculation of the drag force.

To approximate the risk measure, intrusive and non-intrusive methods were explored. The intrusive approach requires the RANS adjoint solution, which is used to compute the gradient of the quantity of interest (QoI) of the problem with respect to the uncertainty sources. The gradients allow the 1st order approximation of the QoI empirical cumulative distribution function (ECDF), and from this approximation, the conditional value at risk, CVaR, is computed. This approach has been applied to solve the aerodynamic shape optimization of a Blended Wing Body (BWB) central section at typical cruise flight conditions $\left(M_\infty = 0.8,\ c_l = 0.1 \text{ and } Re_c = 174 \times 10^6\right)$. Particularly, the drag coefficient of the airfoil has to be minimized under aerodynamic and geometrical constraints. It was observed that the part modeled the worst with a linear approximation is the one at high Mach numbers. Therefore, a second deterministic design point was introduced, in particular, the drag coefficient with nominal values of the design parameters without taking into account uncertainties at a Mach number equal to 0.82, $c_{d,M=0.82}(\mathbf{w})$. This is an ad-hoc problem definition; thus, it must be revised for any other optimization study. Nevertheless, the approach provides a substantial reduction of computational time with respect to the Monte Carlo method for the ECDF approximation. Specifically, if only 20 Monte Carlo samples are considered, a CPU cost reduction of around 80% for CVaR calculation is obtained.

The same aerodynamic problem was solved, but, in this case, a non-intrusive method was employed. Precisely, Gaussian processes (GPs) are used to approximate the ECDF. Only five true points are used for building the GP. The selection of these five points of the design space is not random. Indeed their selection came from an optimization process using a Genetic Algorithm. Firstly, the points are selected based on the minimization of the distance between the true and the approximated ECDFs of the deterministic optimum airfoil of this problem. After a first robust optimization, three true ECDFs are added to the training phase, and new points to build the Gaussian process are selected. Again, the objective is to minimize the distance between ECDFs, but this time for the whole set. However, it was appreciated that this objective was not sufficient for a proper approximation of the CVaR. Thus, it was decided to include the distance between the approximated and true risk measures and a penalty term to the order relationship. With this new set of points, a robust optimization run was carried out. A good approximation of the ECDF and the CVaR was obtained for the optimal result. This approach provides reduction around the 70% with respect to the Monte Carlo method. In comparison with the adjoint method, the cost reduction is lower, but the definition of the problem is independent of the case study.

Moreover, surrogate models have also been applied to solve the aerodynamic shape design optimization problem of a propeller blade at a viscous flow, typical for its working conditions $\left(M_\infty = 0.2 \text{ and } Re_c = 4.97 \times 10^6\right)$. The drag produced by the airfoil is minimized while satisfying the requirement imposed on the lift coefficient, together with some geometrical constraints. Particularly, surrogate-based optimization was carried out using a multi-fidelity Gaussian process. The levels of fidelity are introduced using different grid sizes. In addition, at the low-fidelity level, the far-field formula has been included to provide a better approximation of the c_d. The method showed considerable advantages with respect to classical single-fidelity surrogates and to population based optimization algorithms from a computational cost point of view.

Finally, an optimization algorithm that mixes continuous and integer variables has been used for the robust design of High Lift Devices (HLD) at $M_\infty = 0.2$ and $Re_c = 5 \times 10^6$. The aim was to demonstrate that a crafted optimization algorithm can handle configurational decisions that are usually the responsibility of experienced designers. A deterministic optimization was, in a first step, done to demonstrate the proper behavior of the algorithm. Then, to make the robust problem treatable, a surrogate-based optimization approach was introduced, together with a classifier that aimed to predict if the candidate will fulfill the convergence criteria beforehand. Thus, if the classifier predicts that the candidate will not converge, the CFD evaluation is not performed, providing a saving of CPU time. In addition, to quantify the uncertainty, a quadrature approach has been implemented. Satisfactory results were obtained; hence it can be said that this approach for taking configurational decisions by the optimizer can represent a turning point in multidisciplinary design optimization.

For the sake of clarity, it is finally summarized what, in the author opinion, were the innovative aspects in the field of robust design for aerodynamic configurations treated in this thesis:

1. The applicability of far-field formulas to aerodynamic optimization problems, both deterministic and robust, was shown. The main obstacle faced was that, especially with very coarse grids, these formulas tend to give unreliable results for aerodynamic configurations with separate flows or characterized by geometries with many curvature changes, which often appear in evolutionary optimization processes. To solve this problem, it was developed an ad-hoc formulation of the aerodynamic optimization problem that allows the recognition of such situations by introducing an ad-hoc constraint.

2. The introduction of a new intrusive methodology for the quantification of uncer-

tainty, based on the gradient calculation through the adjoint field, has allowed a cost reduction of about $80\%$for the estimation of the CVaR risk measure compared to the Monte Carlo sampling. It was also extensively discussed and demonstrated the benefits of an asymmetrical risk measure, such as CVaR, in robust aerodynamic design.

3. A non-intrusive approach was developed, complementary to the intrusive one, based on a Gaussian process iterative training. The application of this methodology to the same aerodynamic design problem has allowed us to show that, even with an efficiency slightly lower than that obtainable with the adjoint-based method in the CVaR approximation, an improvement of about 70% was obtained compared to a typical implementation of the Monte Carlo method.

4. A multi-fidelity surrogate-based optimization chain, proposed in collaboration with a Ph.D. candidate of Trieste University, showed significant advantages with respect to classical surrogate-based and population-based approaches to robust aerodynamic shape design.

5. Finally, in collaboration with a Ph.D. from the University of Strathclyde, it was explored how to extend robust optimization methodologies to complex aerodynamic configurations, such as multi-component airfoils. Simultaneously solving both topological and dimensioning aspects required the introduction of new elements in the optimization computational chain. In particular, we began experimenting with machine intelligence and classifier systems to increase the effectiveness of our approximators.

References

[1] Thomas A Zang. *Needs and opportunities for uncertainty-based multidisciplinary design methods for aerospace vehicles*. National Aeronautics and Space Administration, Langley Research Center, 2002.

[2] Xiaodong Wang, Charles Hirsch, Zhiyi Liu, Shun Kang, and Chris Lacor. Uncertainty-based robust aerodynamic optimization of rotor blades. *International journal for numerical methods in engineering*, 94(2):111–127, 2013.

[3] Barbara Arizmendi, Tommaso Bellosta, Ana Isabel del Val, Giulio Gori, Mariana O Prazeres, and Joao Reis. On real-time management of on-board ice protection systems by means of machine learning. In *AIAA Aviation 2019 Forum*, page 3464, 2019.

[4] Gyung-Jin Park, Tae-Hee Lee, Kwon Hee Lee, and Kwang-Hyeon Hwang. Robust design: An overview. *AIAA Journal*, 44(1):181–191, January 2006. ISSN 0001-1452. doi: 10.2514/1.13639. URL https://doi.org/10.2514/1.13639.

[5] Domenico Quagliarella. Value-at-risk and conditional value-at-risk in optimization under uncertainty. In Charles Hirsch, Dirk Wunsch, Jacek Szumbarski, Łukasz Łaniewski-Wołłk, and Jordi Pons-Prats, editors, *Uncertainty Management for Robust Industrial Design in Aeronautics: Findings and Best Practice Collected During UMRIDA, a Collaborative Research Project (2013–2016) Funded by the European Union*, pages 541–565. Springer International Publishing, Cham, 2019. ISBN 978-3-319-77767-2. doi: 10.1007/978-3-319-77767-2_34. URL https://doi.org/10.1007/978-3-319-77767-2_34.

[6] Luigi Paparone and Renato Tognaccini. Computational fluid dynamics-based drag prediction and decomposition. *AIAA Journal*, 41(9):1647–1657, 2003.

[7] Klaus Oswatitsch. *Gas dynamics*, volume 1. Academic Press, 1956.

[8] D Destarac and J Van Der Vooren. Drag/thrust analysis of jet-propelled transonic transport aircraft; definition of physical drag components. *Aerospace Science and Technology*, 8(6):545–556, 2004.

[9] Martin Gariépy, Benoit Malouin, Jean-Yves Trépanier, and Éric Laurendeau. Far-field drag decomposition applied to the drag prediction workshop 5 cases. *Journal of Aircraft*, 50(6):1822–1831, 2013.

[10] Norbert Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60 (4):897–936, 1938.

[11] Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, 24(2): 619–644, 2002.

[12] Thierry Crestaux, Olivier Le Maıtre, and Jean-Marc Martinez. Polynomial chaos expansion for sensitivity analysis. *Reliability Engineering & System Safety*, 94(7): 1161–1172, 2009.

[13] Lucia Parussini, Valentino Pediroda, and Carlo Poloni. Prediction of geometric uncertainty effects on fluid dynamics by polynomial chaos and fictitious domain method. *Computers & fluids*, 39(1):137–151, 2010.

[14] Jeroen AS Witteveen and Gianluca Iaccarino. Simplex stochastic collocation with random sampling and extrapolation for nonhypercube probability spaces. *SIAM Journal on Scientific Computing*, 34(2):A814–A838, 2012.

[15] Sethuraman Sankaran, Charles Audet, and Alison L Marsden. A method for stochastic constrained optimization using derivative-free surrogate pattern search and collocation. *Journal of Computational Physics*, 229(12):4664–4682, 2010.

[16] Michael Eldred and John Burkardt. Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. In *47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 976, 2009.

[17] Stefan Heinrich. Monte carlo complexity of global solution of integral equations. *Journal of Complexity*, 14(2):151–175, 1998.

[18] Michael B Giles. Multilevel monte carlo path simulation. *Operations research*, 56 (3):607–617, 2008.

[19] S Krumscheid, Fabio Nobile, and M Pisaroni. Quantifying uncertain system outputs via the multilevel monte carlo method—part i: Central moment estimation. *Journal of Computational Physics*, 414:109466, 2020.

[20] Michele Pisaroni, Fabio Nobile, and Pénélope Leyland. A continuation multi level monte carlo (c-mlmc) method for uncertainty quantification in compressible inviscid aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 326: 20–50, 2017.

[21] Elisa Morales Tirado and Domenico Quagliarella. *Risk Measures in the Context of Robust and Reliability Based Optimization*, pages 411–427. Springer International Publishing, Cham, 2021. ISBN 978-3-030-60166-9. doi: 10.1007/978-3-030-60166-9_13. URL https://doi.org/10.1007/978-3-030-60166-9_13.

[22] G. N. Vanderplaats. *Numerical Optimization Techniques for Engineering Design: with Applications.* Mc Graw–Hill, 2001.

[23] Joseph K Blitzstein and Jessica Hwang. *Introduction to probability.* Chapman and Hall/CRC, 2014.

[24] D.S. Lee, L.F. Gonzalez, J. Périaux, and K. Srinivas. Efficient hybrid-game strategies coupled to evolutionary algorithms for robust multidisciplinary design optimization in aerospace engineering. *Evolutionary Computation, IEEE Transactions on*, 15 (2):133–150, April 2011. ISSN 1089-778X. doi: 10.1109/TEVC.2010.2043364.

[25] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 7 1999. ISSN 1467-9965. doi: 10.1111/1467-9965.00068. URL http:https://doi.org/10.1111/1467-9965.00068.

[26] D. Quagliarella and E. Iuliano. Robust design of a supersonic natural laminar flow wing-body. *IEEE Computational Intelligence Magazine*, 12(4):14–27, Nov 2017. ISSN 1556-603X. doi: 10.1109/MCI.2017.2742718.

[27] A. W. van der Vaart. *Asymptotic Statistics.* Cambridge University Press, 1998. URL http://dx.doi.org/10.1017/CBO9780511802256.

[28] A. Alexandre Trindade, Stan Uryasev, Alexander Shapiro, and Grigory Zrazhevsky. Financial prediction with constrained tail risk. *Journal of Banking & Finance*, 31(11):3524–3538, 2007. URL http://EconPapers.repec.org/RePEc:eee:jbfina:v:31:y:2007:i:11:p:3524-3538.

[29] Patrick Billingsley. *Probability and Measure.* Wiley, New York, NY, third edition, 1995.

[30] Sergio Amaral, Douglas Allaire, and Karen Willcox. Optimal L_2-norm empirical importance weights for the change of probability measure. *Statistics and Computing*, pages 1–19, 2016. ISSN 1573-1375. doi: 10.1007/s11222-016-9644-3. URL http://dx.doi.org/10.1007/s11222-016-9644-3.

[31] Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer, 1992.

[32] Mark Drela and Harold Youngren. *XFOIL 6.94 User Guide*. MIT Aero & Astro, December 2001.

[33] J Van Ingen. The eN method for transition prediction. Historical review of work at TU Delft. In *38th Fluid Dynamics Conference and Exhibit*, page 3830, 2008.

[34] Nikolaus Hansen. The CMA evolution strategy: A comparing review. In Jose A. Lozano, Pedro Larrañaga, Iñaki Inza, and Endika Bengoetxea, editors, *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*, pages 75–102. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-32494-2. doi: 10.1007/3-540-32494-1_4. URL http://dx.doi.org/10.1007/3-540-32494-1_4.

[35] Domenico Quagliarella. *Aerodynamic Shape Design Using Evolutionary Computation: A Tutorial with Examples and Case Studies*, pages 529–581. American Institute of Aeronautics and Astronautics (AIAA), 2014.

[36] David B Fogel. Nils barricelli-artificial life, coevolution, self-adaptation. *IEEE Computational Intelligence Magazine*, 1(1):41–45, 2006.

[37] Kenneth Alan De Jong. Analysis of the behavior of a class of genetic adaptive systems. Technical report, 1975.

[38] David E Goldenberg. Genetic algorithms in search, optimization and machine learning, 1989.

[39] Alessandro Vicini and Domenico Quagliarella. Inverse and direct airfoil design using a multiobjective genetic algorithm. *AIAA journal*, 35(9):1499–1505, 1997.

[40] Alessandro Vicini and Domenico Quagliarella. Airfoil and wing design through hybrid optimization strategies. *AIAA journal*, 37(5):634–641, 1999.

[41] Michele Pisaroni. Multi level monte carlo methods for uncertainty quantification and robust design optimization in aerodynamics. Technical report, EPFL, 2017.

[42] D. Quagliarella, G. Petrone, and G. Iaccarino. Optimization under uncertainty using the generalized inverse distribution function. In W. Fitzgibbon, editor, *AISTATS*, volume 34 of *Computational Methods in Applied Sciences*, pages 171–190. Springer, NL, June 2014.

[43] Samineh Bagheri, Wolfgang Konen, Richard Allmendinger, Jürgen Branke, Kalyanmoy Deb, Jonathan Fieldsend, Domenico Quagliarella, and Karthik Sindhya. Constraint handling in efficient global optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference 2017 (GECCO '17)*, page 8 pages, Berlin, Germany, July 15–19 2017. GECCO. doi: DOI:10.475/1234.

[44] R. Hicks and P. A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407–412, 1978.

[45] Les Piegl and Wayne Tiller. *The NURBS book*. Springer Science & Business Media, 2012.

[46] Wayne Tiller. Rational b-splines for curve and surface representation. *IEEE Computer Graphics and Applications*, (6):61–69, 1983.

[47] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. doi: 10.1002/nme.2579. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2579`.

[48] Construct2d. `https://sourceforge.net/projects/construct2d/`.

[49] Claudio Marongiu, Pietro Catalano, Marcello Amato, and Gianluca Iaccarino. U-ZEN: a computational tool solving U-RANS equations for industrial unsteady applications. In *34th AIAA Fluid Dynamics Conference and Exhibit*, page 2345, 2004.

[50] Francisco Palacios, Juan Alonso, Karthikeyan Duraisamy, Michael Colonno, Jason Hicken, Aniket Aranake, Alejandro Campos, Sean Copeland, Thomas Economon, Amrita Lonkar, Trent Lukaczyk, and Thomas Taylor. Stanford University Unstructured (SU^2): An open-source integrated computational environment for multi-physics simulation and design. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, 2013. doi: 10.2514/6.2013-287. URL `https://doi.org/10.2514/6.2013-287`.

[51] Thomas D Economon, Francisco Palacios, Sean R Copeland, Trent W Lukaczyk, and Juan J Alonso. SU2: An open-source suite for multiphysics simulation and design. *Aiaa Journal*, 54(3):828–846, 2016.

[52] Elisa Morales Tirado, Domenico Quagliarella, and Renato Tognaccini. Airfoil optimization using far-field analysis of the drag force. In *AIAA Scitech 2019 Forum*, page 0972, 2019.

[53] Yitong Fan and Weipeng Li. Review of far-field drag decomposition methods for aircraft design. *Journal of Aircraft*, 56(1):11–21, 2019.

[54] D. Hue and S. Esquieu. Computational drag prediction of the dpw4 configuration using the far-field approach. *Journal of Aircraft*, 48(5):1658–1670, 2011.

[55] Makoto Ueno, Kazuomi Yamamoto, Kentaro Tanaka, Mitsuhiro Murayama, and Renato Tognaccini. Far-field drag analysis of nasa common research model simulation. *Journal of Aircraft*, 50(2):388–397, 2013.

[56] J-Z Wu, X-Y Lu, and L-X Zhuang. Integral force acting on a body due to local flow structures. *Journal of Fluid Mechanics*, 576:265–286, 2007.

[57] Claudio Marongiu, Renato Tognaccini, and Makoto Ueno. Lift and lift-induced drag calculation by lamb vector integration. *AIAA Journal*, 51(6):1420–1430, 2013.

[58] B. Mele and R. Tognaccini. Aerodynamic force by Lamb vector integrals in compressible flow. *Physics of Fluids*, 26:056104–1–16, 2014. doi: 10.1063/1.4875015.

[59] B. Mele, M. Ostieri, and R. Tognaccini. Vorticity based breakdown of the aerodynamic force in three-dimensional compressible flows. *AIAA Journal*, 54(4):1198–1208, 2016.

[60] Benedetto Mele, Mario Ostieri, and Renato Tognaccini. Aircraft lift and drag decomposition in transonic flows. *Journal of Aircraft*, 54(5):1933–1944, Sep 2017. ISSN 1533-3868. doi: 10.2514/1.c034288. URL `http://dx.doi.org/10.2514/1.C034288`.

[61] Mario Ostieri and Renato Tognaccini. On a recently proposed vorticity-based definition of wave drag. *Journal of Aircraft*, 55(6):2521–2523, 2018.

[62] Mario Ostieri, Benedetto Mele, and Renato Tognaccini. Linear and nonlinear decomposition of aerodynamic force acting on an oscillating plate. *AIAA Journal*, 56(2):594–608, Feb 2018. ISSN 1533-385X. doi: 10.2514/1.j056129. URL `http://dx.doi.org/10.2514/1.J056129`.

[63] Lorenzo Russo, Mario Ostieri, and Renato Tognaccini. Thrust extraction from vorticity fields in steady and unsteady flows. In *AIAA 2018-3967*, pages 2018–3967–1–21. American Institute of Aeronautics and Astronautics, June 2018. ISBN 9781624105593. doi: 10.2514/6.2018-3967. URL `http://dx.doi.org/10.2514/6.2018-3967`.

[64] Lorenzo Russo, Renato Tognaccini, and Luciano Demasi. Box wing and induced drag: Compressibility effects in subsonic and transonic regimes. *Journal of Aircraft*, Article in advance, 2020. doi: 10.2514/1.J059080.

[65] Martin Gariepy, Jean-Yves Trepanier, Eddy Petro, Benoit Malouin, Charles Audet, Sébastien LeDigabel, and Christophe Tribes. Direct search airfoil optimization using far-field drag decomposition results. In *53rd AIAA Aerospace Sciences Meeting*,

AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, January 2015. doi: 10.2514/6.2015-1720. URL https://doi.org/10.2514/6.2015-1720.

[66] S Nadarajah. Aerodynamic design optimization: Drag minimization of the naca 0012 in transonic inviscid flow. *Retreived from https://info.aiaa.org/tac/ASG/APATC/AeroDesignOpt-DG/default.aspx*, 201, 2013.

[67] Marco Lanzetta, Benedetto Mele, and Renato Tognaccini. Advances in aerodynamic drag extraction by far-field methods. *Journal of Aircraft*, 52(6):1873–1886, 2014.

[68] Elisa Morales, Andrea Bornaccioni, Domenico Quagliarella, and Renato Tognaccini. Gradient based empirical cumulative distribution function approximation for robust aerodynamic design. *Aerospace Science and Technology*, 112:106630, 2021.

[69] Advisory Council for Aeronautical Research in Europe. Flightpath 2050: Europe's vision for aviation, 2011.

[70] Advisory Council for Aeronautical Research in Europe. Strategic research & innovation agenda, 2017.

[71] M. A. Sargeant, T. P. Hynes, W. R. Graham, J. I. Hileman, M. Drela, and Z. S. Spakovszky. Stability of hybrid-wing-body-type aircraft with centerbody leading-edge carving. *Journal of Aircraft*, 47(3):970–974, 2010. doi: 10.2514/1.46544. URL https://doi.org/10.2514/1.46544.

[72] Abraham Wald. Statistical decision functions which minimize the maximum risk. *Annals of Mathematics*, 46:265–280, 1945.

[73] Alberto Clarich and Rosario Russo. Formulations for robust design and inverse robust design. In Charles Hirsch, Dirk Wunsch, Jacek Szumbarski, Łukasz Łaniewski-Wołłk, and Jordi Pons-Prats, editors, *Uncertainty Management for Robust Industrial Design in Aeronautics: Findings and Best Practice Collected During UMRIDA, a Collaborative Research Project (2013–2016) Funded by the European Union*, pages 447–462. Springer International Publishing, Cham, 2019. ISBN 978-3-319-77767-2. doi: 10.1007/978-3-319-77767-2_28. URL https://doi.org/10.1007/978-3-319-77767-2_28.

[74] Tim A. Albring, Max Sagebaum, and Nicolas R. Gauger. Efficient aerodynamic design using the discrete adjoint method in SU2. In *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. AIAA, 2015. doi: 10.2514/6.2016-3518. URL https://arc.aiaa.org/doi/abs/10.2514/6.2016-3518.

[75] Philippe Spalart and Steven Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439, 1992.

[76] Emiliano Iuliano and Domenico Quagliarella. Efficient aerodynamic optimization of a very light jet aircraft using evolutionary algorithms and reynolds-averaged navier-stokes flow models. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 225(10):1109–1129, October 2011. doi: 10.1177/0954410011412450.

[77] Carlo Acerbi and Dirk Tasche. Expected shortfall: A natural coherent alternative to value at risk. *Economic Notes*, 31(2):379–388, 2002. ISSN 1468-0300. doi: 10.1111/1468-0300.00091. URL `http://dx.doi.org/10.1111/1468-0300.00091`.

[78] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000.

[79] R. Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26:1443–1471, 2002.

[80] Achintya Haldar and Sankaran Mahadevan. *Probability, Reliability, and Statistical Methods in Engineering Design*. John Wiley & Sons Inc, 1999.

[81] Luis Crespo and Sean Kenny. A first and second order moment approach to probabilistic control synthesis. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, August 2005. doi: 10.2514/6.2005-6133. URL `https://arc.aiaa.org/doi/abs/10.2514/6.2005-6133`.

[82] Alain Dervieux. Introduction to intrusive perturbation methods. In Charles Hirsch, Dirk Wunsch, Jacek Szumbarski, Łukasz Łaniewski-Wołłk, and Jordi Pons-Prats, editors, *Uncertainty Management for Robust Industrial Design in Aeronautics: Findings and Best Practice Collected During UMRIDA, a Collaborative Research Project (2013–2016) Funded by the European Union*, pages 327–334. Springer International Publishing, Cham, 2019. ISBN 978-3-319-77767-2. doi: 10.1007/978-3-319-77767-2_20. URL `https://doi.org/10.1007/978-3-319-77767-2_20`.

[83] Antony Jameson. Aerodynamic design via control theory. *Journal of scientific computing*, 3(3):233–260, 1988.

[84] J Brezillon and NR Gauger. 2d and 3d aerodynamic shape optimisation using the adjoint approach. *Aerospace Science and Technology*, 8(8):715–727, 2004.

[85] Thomas D Economon, Francisco Palacios, and Juan J Alonso. Unsteady continuous adjoint approach for aerodynamic design on dynamic meshes. *AIAA Journal*, 53 (9):2437–2453, 2015.

[86] Gaetan KW Kenway and Joaquim RRA Martins. Multipoint aerodynamic shape optimization investigations of the common research model wing. *AIAA Journal*, 54(1):113–128, 2016.

[87] Tim A Albring, Max Sagebaum, and Nicolas R Gauger. Efficient aerodynamic design using the discrete adjoint method in SU2. In *17th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 3518, 2016.

[88] Domenico Quagliarella and Emiliano Iuliano. Robust design of a supersonic natural laminar flow wing-body. *IEEE Computational Intelligence Magazine*, 12(4): 14–27, 2017.

[89] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

[90] Guangda Yang, Andrea Da Ronch, Jernej Drofelnik, and Zheng-Tong Xie. Sensitivity assessment of optimal solution in aerodynamic design optimisation using SU2. *Aerospace Science and Technology*, 81:362–374, 2018.

[91] J. Nocedal and S. J. Wright. *Numerical Optimization (2nd ed.)*. Springer-Verlag, New York, 2006.

[92] Alain Dervieux. Algoritmic differentiation for second derivatives. In Charles Hirsch, Dirk Wunsch, Jacek Szumbarski, Łukasz Łaniewski-Wołłk, and Jordi Pons-Prats, editors, *Uncertainty Management for Robust Industrial Design in Aeronautics: Findings and Best Practice Collected During UMRIDA, a Collaborative Research Project (2013–2016) Funded by the European Union*, pages 327–334. Springer International Publishing, Cham, 2019. ISBN 978-3-319-77767-2. doi: 10.1007/978-3-319-77767-2_21. URL https://doi.org/10.1007/978-3-319-77767-2_21.

[93] Marcin Wyrozębski, Łukasz Łaniewski-Wołłk, and Jacek Rokicki. Second-order derivatives for geometrical uncertainties. In Charles Hirsch, Dirk Wunsch, Jacek Szumbarski, Łukasz Łaniewski-Wołłk, and Jordi Pons-Prats, editors, *Uncertainty Management for Robust Industrial Design in Aeronautics: Findings and Best Practice Collected During UMRIDA, a Collaborative Research Project (2013–2016) Funded by the European Union*, pages 327–334. Springer International Publishing, Cham, 2019. ISBN 978-3-319-77767-2. doi: 10.1007/978-3-319-77767-2_22. URL https://doi.org/10.1007/978-3-319-77767-2_22.

[94] Nestor V Queipo, Raphael T Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, and P Kevin Tucker. Surrogate-based analysis and optimization. *Progress in aerospace sciences*, 41(1):1–28, 2005.

[95] Timothy Simpson, Vasilli Toropov, Vladimir Balabanov, and Felipe Viana. Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come-or not. In *12th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 5802, 2008.

[96] Alexander IJ Forrester and Andy J Keane. Recent advances in surrogate-based optimization. *Progress in aerospace sciences*, 45(1-3):50–79, 2009.

[97] Daigo Maruyama, Stefan Görtz, and Dishi Liu. A framework for robust and reliability-based design optimization of airfoils considering geometrical uncertainties. 2016.

[98] Christian Sabater and Stefan Görtz. An efficient bi-level surrogate approach for optimizing shock control bumps under uncertainty. In *AIAA Scitech 2019 Forum*, page 2214, 2019.

[99] Ilya M Sobol', Danil Asotsky, Alexander Kreinin, and Sergei Kucherenko. Construction and comparison of high-dimensional sobol'generators. *Wilmott*, 2011 (56):64–79, 2011.

[100] Michael Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987.

[101] Jooyoung Park and Irwin W Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.

[102] Martin D Buhmann. *Radial basis functions: theory and implementations*, volume 12. Cambridge university press, 2003.

[103] Harris Drucker, Christopher J Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161, 1996.

[104] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[105] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. ISBN 0-262-18253-X.

[106] Mark N. Gibbs. *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, University of Cambridge, 1997.

[107] Mark Gibbs and David J.C. MacKay. Efficient implementation of gaussian processes. Technical report, Cambridge University Engineering Department, 1997.

[108] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4): 455–492, 1998.

[109] Stefan Heinrich. Multilevel Monte Carlo methods. In Svetozar Margenov, Jerzy Waśniewski, and Plamen Yalamov, editors, *Large-Scale Scientific Computing: Third International Conference, LSSC 2001 Sozopol, Bulgaria, June 6–10, 2001 Revised Papers*, pages 58–67. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. ISBN 978-3-540-45346-8. doi: 10.1007/3-540-45346-6{_}5. URL `http://dx.doi.org/10.1007/3-540-45346-6{_}5`.

[110] Michael B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 5 2015. ISSN 1474-0508. doi: 10.1017/S096249291500001X. URL `http://journals.cambridge.org/article{_}S096249291500001X`.

[111] JA Witteveen, Alireza Doostan, Tonkid Chantrasmi, Rene Pecnik, and Gianluca Iaccarino. Comparison of stochastic collocation methods for uncertainty quantification of the transonic rae 2822 airfoil. In *Proceedings of workshop on quantification of CFD uncertainties*, 2009.

[112] Mattia Padulo, M Sergio Campobasso, and Marin D Guenov. Novel uncertainty propagation method for robust aerodynamic design. *AIAA journal*, 49(3):530–543, 2011.

[113] Thomas D. Economon, Francisco Palacios, Sean R. Copeland, Trent W. Lukaczyk, and Juan J. Alonso. SU2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, 54(3):828–846, 2016. doi: 10.2514/1.J053813. URL `https://doi.org/10.2514/1.J053813`.

[114] Marc C Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

[115] Martin hepperle mh 114 for a propeller for ultralight. `http://airfoiltools.com/airfoil/details?airfoil=mh114-il`. Accessed: 2020-06-28.

[116] Peter Zeno Korondi, Lucia Parussini, Mariapia Corrada Marchi, and Carlo Poloni. Multi-fidelity gaussian process regression for propeller optimisation under uncertainty. In *EUROGEN 2019*. A. Gaspar-Cunha, 2019.

[117] Péter Zénó Korondi, Mariapia Marchi, Lucia Parussini, and Carlo Poloni. Multi-fidelity design optimisation strategy under uncertainty with limited computational budget. *Optimization and Engineering*, pages 1–26, 2020.

[118] Loic Le Gratiet and Josselin Garnier. Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *International Journal for Uncertainty Quantification*, 4(5), 2014.

[119] Lorenzo Gentile, Elisa Morales, Domenico Quagliarella, Edmondo Minisci, Thomas Bartz-Beielstein, and Renato Tognaccini. High-lift devices topology optimisation using structured-chromosome genetic algorithm. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–9. IEEE, 2020.

[120] Ralf Rudnik and Heiko Geyr. The european high lift project eurolift ii-objectives, approach, and structure. In *25th AIAA applied aerodynamics conference*, page 4296, 2007.

[121] CP Van Dam. The aerodynamic design of multi-element high-lift systems for transport airplanes. *Progress in Aerospace Sciences*, 38(2):101–144, 2002.

[122] Apollo Milton Olin Smith. High-lift aerodynamics. *Journal of Aircraft*, 12(6):501–530, 1975.

[123] Peter KC Rudolph. High-lift systems on commercial subsonic airliners. 1996.

[124] Pierluigi Iannelli, Frédéric Moens, Mauro Minervino, Rita Ponza, and Ernesto Benini. Comparison of optimization strategies for high-lift design. *Journal of Aircraft*, 54(2):642–658, 2017.

[125] Pierluigi Iannelli and Domenico Quagliarella. Multi-objective/multi-point shape and setting high-lift system optimization by means of genetic algorithm and 2D Navier-Stokes equations. In *EUROGEN 2011 Conference proceedings, Capua, Italy*, 2011.

[126] Frédéric Moens and Christelle Wervaecke. Multi-point optimization of shapes and settings of high-lift system by means of evolutionary algorithm and Navier-Stokes equations. *Engineering Computations*, 2013.

[127] Julien Pelamatti, Loïc Brevault, Mathieu Balesdent, El-Ghazali Talbi, and Yannick Guerin. How to deal with mixed-variable optimization problems: An overview of algorithms and formulations. In *World Congress of Structural and Multidisciplinary Optimisation*, pages 64–82. Springer, 2017.

[128] Hui Meen Nyew, Ossama Abdelkhalik, and Nilufer Onder. Structured-chromosome evolutionary algorithms for variable-size autonomous interplanetary

trajectory planning optimization. *Journal of Aerospace Information Systems*, 12(3): 314–328, 2015.

[129] Shadi A Darani and Ossama Abdelkhalik. Space trajectory optimization using hidden genes genetic algorithms. *Journal of Spacecraft and Rockets*, 55(3):764–774, 2017.

[130] Cristian Greco, Lorenzo Gentile, Gianluca Filippi, Edmondo Minisci, Massimiliano Vasile, and Thomas Bartz-Beielstein. Autonomous generation of observation schedules for tracking satellites with structured-chromosome GA optimisation. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 497–505. IEEE, 2019.

[131] L. Gentile. LorenzoGentile/SCGA: SCGA second release. Update. https://doi.org/10.5281/zenodo.3627555, January 2020. URL `https://doi.org/10.5281/zenodo.3627555`.

[132] Lorenzo Gentile, Cristian Greco, Edmondo Minisci, Thomas Bartz-Beielstein, and Massimiliano Vasile. An optimization approach for designing optimal tracking campaigns for low-resources deep-space missions. In *70th International Astronautical Congress*, 2019.

[133] Lorenzo Gentile, Cristian Greco, Edmondo Minisci, Thomas Bartz-Beielstein, and Massimiliano Vasile. Structured-chromosome GA optimisation for satellite tracking. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1955–1963, 2019.

[134] Vincent Chin, David Peters, Frank Spaid, and Robert Mcghee. Flowfield measurements about a multi-element airfoil at high Reynolds numbers. In *23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference*, page 3137, 1993.

[135] Steven M Klausmeyer and John C Lin. Comparative results from a CFD challenge over a 2D three-element high-lift airfoil. 1997.

[136] Mark Drela. A user's guide to MSES 3.05. *Massachusetts Institute of Technology (MIT), Cambridge*, 2007.

[137] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[138] Gregor Papa. Parameter-less algorithm for evolutionary-based optimization. *Computational Optimization and Applications*, 56(1):209–229, 2013.

[139] Imtiaz Korejo and Shengxiang Yang. Comparative study of adaptive mutation operators for genetic algorithms. In *in MIC 2009: The VIII Metaheuristics International Conference, 2009. 196 Journal of Emerging Technology and Advanced Engineering Website: www. ijetae. com (ISSN 2250-2459, Volume 2, Issue 2*. Citeseer, 2012.

[140] Rui Li, Michael TM Emmerich, Jeroen Eggermont, Thomas Bäck, Martin Schütz, Jouke Dijkstra, and Johan HC Reiber. Mixed integer evolution strategies for parameter optimization. *Evolutionary computation*, 21(1):29–64, 2013.

[141] Kendall E Atkinson. *An introduction to numerical analysis*. John wiley & sons, 2008.

[142] Alexander Forrester, Andras Sobester, and Andy Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

[143] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. *Towards Global Optimization 2*, chapter The application of Bayesian methods for seeking the extremum, pages 117–129. North-Holland, 1978.

[144] Daniel Horn, Jörg Stork, Nils-Jannik Schüßler, and Martin Zaefferer. Surrogates for hierarchical search spaces: The wedge-kernel and an automated analysis. In Manuel López-Ibáñez, editor, *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO'19*, GECCO '19, pages 916–924, Prague, Czech Republic, 2019. ACM. ISBN 978-1-4503-6111-8. doi: 10.1145/3321707.3321765. URL `http://doi.acm.org/10.1145/3321707.3321765`.

[145] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[146] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.

[147] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

[148] Vincent Chin, David Peters, Frank Spaid, and Robert Mcghee. Flowfield measurements about a multi-element airfoil at high Reynolds numbers. In *23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference*, page 3137, 1993.

[149] Steven M Klausmeyer and John C Lin. Comparative results from a CFD challenge over a 2D three-element high-lift airfoil. 1997.

[150] Martin Zaefferer, Joerg Stork, Martina Friese, Andreas Fischbach, Boris Naujoks, and Thomas Bartz-Beielstein. Efficient global optimization for combinatorial problems. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation (GECCO'14)*, pages 871–878, New York, NY, USA, 2014. ACM. URL `http://doi.acm.org/10.1145/2576768.2598282`.

[151] Marvin Wright and Andreas Ziegler. ranger: A fast implementation of random forests for high dimensional data in c++ and r. *Journal of Statistical Software, Articles*, 77(1):1–17, 2017. ISSN 1548-7660. doi: 10.18637/jss.v077.i01. URL `https://www.jstatsoft.org/v077/i01`.

[152] Lorenzo Gentile. LorenzoGentile/SCGA: SCGA second release. Update, January 2020. URL `https://doi.org/10.5281/zenodo.3627555`.

List of Author’s Publications

JOURNAL ARTICLES

- E. Morales, D. Quagliarella, R. Tognaccini (2021). Gradient based empirical cumulative distribution function approximation for robust aerodynamic design. Aerospace Science and Technology, 112, 106630.

CONFERENCE PAPERS[1]

- E. Morales, D. Quagliarella and R. Tognaccini. Airfoil Optimization Using Far-Field Analysis of the Drag Force. *AIAA Scitech 2019 Forum,* San Diego (USA), 2019.

- L. Gentile, E. Morales, D. Quagliarella, E. Minisci, T. Bartz-Beielstein, R. Tognaccini. High-Lift Devices Topology Optimisation using Structured-Chromosome Genetic Algorithm. *IEEE CEC 2020,* Glasgow (UK), 2020. (*peer-reviewed conference*).

BOOK CHAPTERS

- D. Quagliarella, E. Morales, and A. Bornaccioni. (2020). Risk Measures Applied to Robust Aerodynamic Shape Design Optimization. In Flexible Engineering Toward Green Aircraft (pp. 153-168). Springer, Cham.

- E. Morales and D. Quagliarella. (2021). Risk Measures in the Context of Robust and Reliability Based Optimization. In Optimization Under Uncertainty with Applications to Aerospace Engineering (pp. 411-427). Springer, Cham.

- E. Morales, D. Quagliarella, and R. Tognaccini (in printing). Gaussian Processes for CVaR approximation in Robust Aerodynamic Shape Design. In Advances in Uncertainty Quantification and Optimization Under Uncertainty with Aerospace Applications. Springer, Cham.

- E. Morales, PZ. Korondi, D. Quagliarella, R. Tognaccini, M. Marchi, L. Parussini, and C. Poloni (in printing). Multi-fidelity Surrogate Assisted Design Optimisation of an Airfoil under Uncertainty using Far-Field Drag Approximation. In Advances in Uncertainty Quantification and Optimization Under Uncertainty with Aerospace Applications. Springer, Cham.

[1] Speaker.

- L. Gentile, E. Morales, M. Zaefferer, E. Minisci, D. Quagliarella, T. Bartz-Beielstein, and R. Tognaccini (in printing). High-Lift Devices Topology Robust Optimisation using Machine Learning Assisted Optimisation. In Advances in Uncertainty Quantification and Optimization Under Uncertainty with Aerospace Applications. Springer, Cham.

Colophon

This thesis was typeset using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 11 point Arno Pro, designed by Robert Slimbach in the style of book types from the Aldine Press in Venice, and issued by Adobe in 2007. A template, which can be used to format a PhD thesis with this look and feel, has been released under the permissive MIT (X11) license, and can be found online at github.com/suchow/ or from the author at suchow@post.harvard.edu.