National Technical University of Athens
School of Mechanical Engineering
Fluids Department
Laboratory of Thermal Turbomachines
Parallel CFD & Optimization Unit

Non-Intrusive Polynomial Chaos Expansion for Aerodynamic Uncertainty Quantification & Robust Design with Manufacturing Imperfections

Diploma Thesis
by

Sergios Villette

Advisor:
Kyriakos C. Giannakoglou, Professor NTUA

Athens, February 2022

Acknowledgments

My deepest gratitude goes to my mother Lina and my grandmother Maria for the patience, encouragement and caring they showed for me, throughout the course of my life as a whole. My family has ever been a pillar of trust and support from the start of my education, up until the end of my student years. This work is, primarily, devoted to them, as a subtle show of appreciation for the sacrifices they underwent for me and the shining examples they set for me to follow.

I would, also, like to thank Prof. Kyriakos Giannakoglou for the time and guidance he offered to me, as well as for the trust he bestowed upon me, for the duration of our collaboration. From the time I spent under his tutelage in the various courses he taught at NTUA, up until the completion of this work, I strongly believe that his experience and know-how contributed to the increase of my interest and the deepening of my knowledge in the scientific domain of his work, for which I am grateful.

I should, also, like to express my gratitude to all the members of PCOpt/NTUA who contributed to this thesis, and especially PhD candidate Andreas Margetis for his constant support and reliable suggestions whenever I needed them. Furthermore, I would like to show appreciation for the help provided by Dr. Evaggelos Papoutsis Kiachagias and Dr. Varvara Asouti at certain stages of my work.

Last but not least, I would like to thank my friends and colleagues Dimitris Kalofolias, Fotis Sisanes and Charalampos Kapeleris for being beside me during the years of my studies, while my special appreciation goes to Π for the moments we shared.

Ευχαριστίες

Αρχικά θέλω να εκφράσω τη βαθιά ευγνωμοσύνη μου προς την μητέρα μου Λίνα και την γιαγιά μου Μαρία, για την υπομονή, ενθάρρυνση και φροντίδα που απλόχερα μου προσέφεραν, καθ΄ όλη τη διάρκεια της ζωής μου. Η οικογένεια μου υπήρξε, για μένα, πάντοτε ένα πυλώνας εμπιστοσύνης και στήριξης από την αρχή της σχολικής μου ζωής, μέχρι το τέλος των φοιτητικών μου χρόνων. Αυτή η εργασία, είναι πρωτίστως αφιερωμένη σε εκείνες, ως μία σεμνή ένδειξη ευγνωμοσύνης για τις θυσίες που έκαναν για εμένα καθώς και για το φωτεινό παράδειγμα που μου έδωσαν με τη στάση ζωής τους.

Νιώθω, ακόμη, την επιθυμία να ευχαριστήσω τον κ. Κυριάκο Γιαννάκογλου για το χρόνο και την καθοδήγηση που μου παρείχε καθώς και για την εμπιστοσύνη που επέδειξε στο πρόσωπό μου, κατά τη περίοδο της συνεργασίας μας. Από τον καιρό που παρακολουθούσα τα μαθημάτα που δίδασκε στο ΕΜΠ, μέχρι και την ολοκλήρωση της διπλωματικής μου εργασίας, θεωρώ πως η εμπειρία και η τεχνογνωσία του επέδρασαν καθοριστικά στη διεύρυνση του ενδιαφέροντος και των γνώσεών μου, όσον αφορά το γνωστικό του αντικείμενο, πράγμα για το οποίο του είμαι ευγνώμων.

Επιθυμώ, επίσης, να ευχαριστήσω όλα τα μέλη της ΜΠΥΡ&Β/ΕΜΠ που συνέβαλαν στην εκπόνηση της παρούσας εργασίας και ιδιαίτερα τον υποψήφιο Διδάκτορα Ανδρέα Μαργέτη για τη διαρκή βοήθεια και τις εύστοχες παρατηρήσεις που μου παρείχε, οποτεδήποτε τις χρείαστηκα. Αντίστοιχα, θέλω να ευχαριστήσω τους Δρ. Ευάγγελο Παπουτσή Κιαχαγιά και Δρ. Βαρβάρα Ασούτη για τη βοήθεια που μου προσέφεραν σε διάφορα στάδια της δουλειάς μου.

Τέλος, δε θα μπορούσα να μην ευχαριστήσω τους φίλους και συναδέλφους μου Δημήτρη Καλοφωλιά, Φώτη Σισανέ και Χαράλαμπο Καπελέρη που ήταν δίπλα μου μέχρι το τέλος της φοιτητικής μου ζωής.

National Technical University of Athens
School of Mechanical Engineering
Fluids Department
Laboratory of Thermal Turbomachines
Parallel CFD & Optimization Unit

Non-Intrusive Polynomial Chaos Expansion for Aerodynamic Uncertainty Quantification & Robust Design with Manufacturing Imperfections

Diploma Thesis

Sergios Villette

Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2022

Abstract

In the field of aerodynamics, the geometrical and flow conditions of a certain shape are usually considered to be constants, while in reality they exhibit some stochasticity, which can have a varying effect on its performance. This thesis, stresses aerodynamic cases in which the geometrical-manufacturing uncertainties of a certain shape are taken into account, by proposing a computational process capable to, firstly, evaluate the stochasticity of their performance (uncertainty quantification) and, secondly, to optimize their stochastic performance (robust design).

Therefore, this thesis presents the development of software, implementing the non-intrusive Polynomial Chaos Expansion and the Karhunen-Loève Transform theories, in order to perform aerodynamic uncertainty quantification and robust design optimization on 2D shapes with manufacturing uncertainties. The Karhunen-Loève Transform theory is used to simulate the real-time uncertainties that may occur during the manufacturing of aerodynamic shapes. The theory of Polynomial Chaos is based on the use of orthogonal polynomials to model the stochasticity of a certain phenomena, by analyzing its stochastic input and quantifying its stochastic output, though the form of its statistical moments. The Karhunen-Loève Transform software developed as well as the OpenFOAM© Computational Fluid Dynamics solvers are coupled to an in-house non-intrusive Polynomial Chaos Expansion code, so as to quantify the stochastic aerodynamic performance of 2D imperfect geometries. Additionally, robust design is performed on such imperfect geometries, parameterized through *Volumetric B-Splines*, by optimizing the statistical moments

of their performance, with respect to the design variables controlling the parameterized shape. This is achieved through the incorporation of the continuous adjoint optimization algorithm, developed by PCOpt/NTUA in the OpenFOAM environment, into the aforementioned Karhunen-Loève Transform and non-intrusive Polynomial Chaos coupled algorithm.

The Karhunen-Loève Transform code is designed to recreate imperfect perturbations on any 2D geometry and when combined the generalist nature of the non-intrusive Polynomial Chaos Expansion mathematical tool, it grants the ability to the proposed method to cope with a wide variety of aerodynamic cases with shape uncertainties. Simultaneously, the deterministic adjoint optimization method greatly mitigates the computational cost needed to perform the uncertainty quantification and robust design processes, when compared to other stochastic methods often employed in literature, such as the Evolutionary Algorithms.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Εργαστήριο Θερμικών Στροβιλομηχανών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής
& Βελτιστοποίησης

Μη-επεμβατικό Ανάπτυγμα Πολυωνυμικού Χάους για Αεροδυναμικό Στιβαρό Σχεδιασμό υπό Κατασκευαστικές Ατέλειες

Διπλωματική εργασία

Σέργιος Βιλλέτ

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, Φεβρουάριος 2022

Στην αεροδυναμική, οι γεωμετρικές και ροϊκές συνθήκες ενός σώματος που έρχεται σε επαφή με ρευστό, συνήθως, θεωρούνται σταθερές, ενώ στην πραγματικότητα εμφανίζουν κάποια στοχαστικότητα, η οποία μπορεί να έχει μικρότερες ή μεγαλύτερες επιπτώσεις στην απόδοσή τους. Η παρούσα διπλωματική εργασία, αναφέρεται σε αεροδυναμικά προβλήματα στα οποία οι γεωμετρικές-κατασκευαστικές ατέλειες των εν λόγω σωμάτων, λαμβάνονται υπόψη. Έτσι προτείνεται μια υπολογιστική μέθοδος για την ποσοτικοποίηση της στοχαστικής απόδοσης (ποσοτικοποίηση αβεβαιότητας) ατελών σωμάτων αλλά και τη βελτιστοποίηση της στοχαστικής τους απόδοσης (στιβαρός σχεδιασμός).

Για το λόγο αυτό στην παρούσα εργασία, αναπτύσσεται λογισμικό, που αξιοποιεί τις θεωρίες του μη-επεμβατικού Αναπτύγματος Πολυωνυμικού Χάους και του μετασχηματισμού Karhunen-Loève για την εκτέλεση αεροδυναμικής ποσοτικοποίησης αβεβαιότητας και στιβαρού σχεδιασμού σε διδιάστατες γεωμετρίες με κατασκευαστικές ατέλειες. Ο μετασχηματισμός Karhunen-Loève χρησιμοποιείται για την προσομοίωση των ατελειών που, πιθανά, διαμορφώνονται στην επιφάνεια αεροδυναμικών σωμάτων, κατά την κατασκευή ή τη λειτουργία τους. Η θεωρία του Πολυωνυμικού Χάους, βασίζεται στη χρήση ορθογωνικών πολυωνύμων για τη μοντελοποίηση της στοχαστικότητας ενός φαινομένου, αναλύοντας τις αβέβαιες μεταβλητές εισόδου και υπολογίζοντας τις στοχαστικές τους εξόδους, με τη μορφή της μέσης τιμής και της τυπικής απόκλισης. Ο κώδικας μετασχηματισμού Karhunen-Loève καθώς και οι επιλύτες υπολογιστικής ρευστοδυναμικής, σε περιβάλλον OpenFOAM©, ενοποιούνται με τον κώδικα μη-επεμβατικού Αναπτύγματος Πολυωνυμικού Χάους, με σκοπό την ποσοτικοποίση αβεβαιότητας σε διδιάστατες γεωμετρίες με κατασκευαστικές ατέλειες. Επιπλέον, ο

αεροδυναμικός στιβαρός σχεδιασμός εκτελείται σε τέτοιες ατελείς γεωμέτριες, παραμετροποιώντας τις τελευταίες με τη μέδοθο προσέγγισης Volumetric B-Splines και βελτιστοποιώντας τη μέση τιμή και την τυπική απόκλιση της αεροδυναμικής τους απόδοσης, ως προς τα σημεία ελέγχου των παραμετροποιημένων υπολογιστικών χωρίων. Τα παραπάνω πραγματοποιούνται με την ενσωμάτωση του λογισμικού συνεχούς συζυγούς μεθόδου βελτιστοποίσης, της ΜΠΥΡ& Β/ΕΜΠ, στον προαναφερόμενο πεπλεγμένο κώδικα, σε περιβάλλον OpenFOAM.

Το λογισμικό μετασχηματισμού Karhunen-Loève, σχεδιάζεται ώστε να δύναται να παράξει κατασκευαστικές ατέλειες σε οποιαδήποτε διδιάστατη γεωμετρία, γεγονός που σε συνδυασμό με τη γενικότητα του μη-επεμβατικού Αναπύγματος Πολυωνυμικού Χάους, προσδίδει τη δυνατότητα στην προτεινόμενη μέθοδο να εκτελέσει ποσοτικοποίηση αβεβαιότητας ή στιβαρό σχεδιασμό σε μία ποικιλία αεροδυναμικών προβλημάτων με γεωμετρικές ατέλειες. Ταυτόχρονα, η χρήση της συζυγούς μεθόδου μειώνει σημαντικά το κόστος των υπολογιστικών διεργασιών που παρέχει η προτεινόμενη μέθοδος, σε σχέση με αντίστοιχες προσπάθειες που χρησιμοποιούν στοχαστικές μεθόδους βελτιστοποίησης, όπως τους Εξελικτικούς Αλγορίθμους.

Nomenclature

CFD	Computational Fluid Dynamics
CAE	Computer Aided Engineering
NTUA	National Technical University of Athens
PCOpt	Parallel CFD & Optimization unit
GBM	Gradient Based Method
SOO	Single Objective Optimization
MOO	Multi Objective Optimization
RDO	Robust Design Optimization
UQ	Uncertainty Quantification
CPU	Central Processing Unit
DNS	Direct Numerical Simulation
NS	Navier-Stokes
RANS	Reynolds-Averaged Navier-Stokes
SD	Sensitivity Derivatives
PDE	Partial Differential Equation
gdPDE	grid displacement Partial Differential Equation
QoI	Quantity of Interest
CP	control point

FAE	Field Adjoint Equations
ABC	Adjoint Boundary Conditions
MC	Monte Carlo
PCE	Polynomial Chaos Expansion
niPCE	non-intrusive Polynomial Chaos Expansion
iPCE	intrusive Polynomial Chaos Expansion
PDF	Probability Density Function
GQ	Gauss Quadrature
GHQ	Gauss-Hermite Quadrature
KLT	Karhunen-Loève Transform
KLE	Karhunen-Loève Expansion
EFS	Equivalent Flow Solution
RHS	Right Hand Side
LHS	Left Hand Side
a.k.a.	also known as
w.r.t.	with respect to

Contents

Chapter 1

Introduction

In recent years, the ever improving performance of modern computer systems in combination with the enhancement of computational methods has led to an increasing use of Computational Fluid Dynamics (CFD) for the purpose of industrial Computer Aided Engineering (CAE) in applications related to fluid mechanics. Modern CFD software can simulate the flow around or inside the shapes faster and more accurately, making them available for the purpose of shape optimization procedure and aerodynamic robust design. Thus, the number and the cost of the experiments needed to evaluate new engineering designs have been thoroughly decreased.

1.1 Shape Optimization

In general, the goal of the optimization mathematical problem of a certain *objective function*, is to compute the values of it's variables that maximize or minimize the said function. These variables are referred to as *design* or *optimization variables.*

In the case of shape optimization, according to the control theory adapted to CFD-based optimization, the geometry to be optimized is controlled by a number of variables, which consist the *design variable vector* ($\vec{b} \in \mathbb{R}^N$) of the optimization process. For instance, these could be the control point coordinates of Bézier–Bernstein polynomials that parameterize the shape under consideration. The quality of the shape to be optimized is evaluated by computing the *objective function*, which is usually an integral quantity related to the fluid flow field. The objective function can be defined either at (part of) the boundaries, such as the total drag or lift force exerted on a body, or in a volume inside the geometry, such as the noise induced by the turbulent flow.

One way of classifying CFD-based optimization methods w.r.t. the number of objective functions, to: single objective optimization (SOO) and multi objective optimization (MOO) [1]. SOO applies when a single objective function is optimized whereas MOO applies when more than one objective functions are to be simultaneously optimized.

A second classification for optimization methods, is one that reflects the way the optimal set of design variables is computed: *stochastic* or *deterministic* [1]. This diploma thesis focuses exclusively on the latter. On the one hand, the stochastic optimization methods pseudo-randomly generate values for the *design variables* and heuristic algorithms are implemented in order to faster reach the optimal solution. The optimization process is concluded when either the objective function has converged to its optimal value or the user-defined maximum number of optimization cycles is reached. Examples of such methods would be the *evolutionary algorithms*. The deterministic or gradient-based optimization methods (GBM) rely on the computation of *sensitivity derivatives* (SD), which consist the gradient of the objective function w.r.t. the design variables. This allows a controlled convergence towards the optimal solution, faster than the one achieved with stochastic methods.

In deterministic shape optimization cases, the shape in question is updated in the direction dictated by the sensitivity derivatives. To do so, the flow field, the objective function value and the new SD field are computed on the current geometry. This process is repeated until the same criterion, as the one mentioned for stochastic methods, is met. Since the values of the sensitivity derivatives tend to be nullified near local minima, it is possible for this method to get entrapped into a local optimum. As a result, the algorithm may converge to a local rather than the global optimal solution, which is the main disadvantage of a GBM. Such a disadvantage does not occur in stochastic methods, as long as the maximum number of optimization cycles (generations) is not quite low.

The flow-chart for the generalized CFD-based optimization algorithm is presented in Figure 1.1, thus summarizing and visualizing everything that has been stated so far.

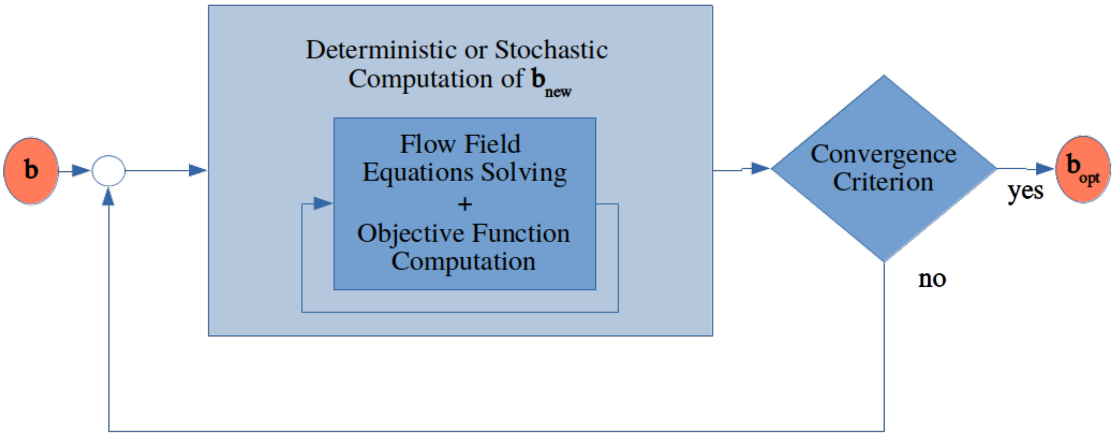

Deterministic or Stochastic Computation of $\mathbf{b}_{new}$
Flow Field Equations Solving + Objective Function Computation
Convergence Criterion
b
$\mathbf{b}_{opt}$
yes
no

Figure 1.1: *The generalized CFD-based optimization algorithm flow chart. A stochastic or deterministic method, either coupled or uncoupled to the fluid flow solver, is implemented in order to update the design variable vector* $\boldsymbol{b}_{new}$*. Depending on the optimization method, the fluid flow solver can be called either once or more times. This process repeats itself for the updated values of the design variables until a certain convergence criterion is met and only then the design variable vector that optimizes the objective function* $\boldsymbol{b}_{opt}$ *can be obtained.*

1.2 Aerodynamic Robust Design

In order to fully grasp the difference between the processes of shape optimization and robust design optimization (RDO), an additional classification concerning the input variables to the problem must be introduced. These variables can be grouped into two categories, according to the variation associated with their values. In the first class of variables their variation is assumed to be defined by a certain stochastic distribution and can, therefore, be quantified through the use of statistical measures such as the mean (μ) or the standard deviation (σ) of the input variable in question.

Variables with no stichasticity are already introduced as *design variables* ($\vec{b}$), while those that have some degree of variation are known as *environmental*, *robust* or *uncertain variables* ($\vec{c} \in \mathbb{R}^M$). It must be clear that in real-life scenarios every variable of a problem is governed by a certain degree of uncertainty, therefore in reality, every variable is an uncertain variable. Yet in the world of CAE, which variables of a simulation are considered to be uncertain is a decision made by the user. For instance, one case of airfoil aerodynamic RDO could focus on a robust design w.r.t. the Mach number of the flow, thus considering all other variables, such as the angle of attack or Reynolds number as design variables, with zero variance.

In contrast, a different airfoil RDO w.r.t. the angle of attack, would consider every other variable including the Mach number as design variables. In the scientific literature, such design cases are a.k.a. as design/optimization under uncertainties.

Therefore, a design case of which all variables consist of design variables, can be called a *shape optimization* (without uncertainties) case and its solution is known as *deterministic optimum* or *optimal design point*. In SOO, the deterministic optimum is defined as the point where the objective function displays is global minimum or maximum value, depending if the case in question requires minimization or maximization, respectively. Whereas, a *robust design* case has both design variables and uncertain variables as inputs ($\equiv (b_1, b_2, ..., b_N, c_1, c_2, ..., c_M) \in \mathbb{R}^{N+M}$) and the solution of such a case is named *robust optimum* or *robust design point*. The robust optimum may not necessarily be the aforesaid optimum, but is defined as the point around which the objective function exhibits extrema for all corresponding values of the uncertain variables within their respective distribution [2]. The *optimum* and *robust design points* of a SOO (maximization of the objective function $f(x) \in \mathbb{R}$) is displayed in Figure 1.2.

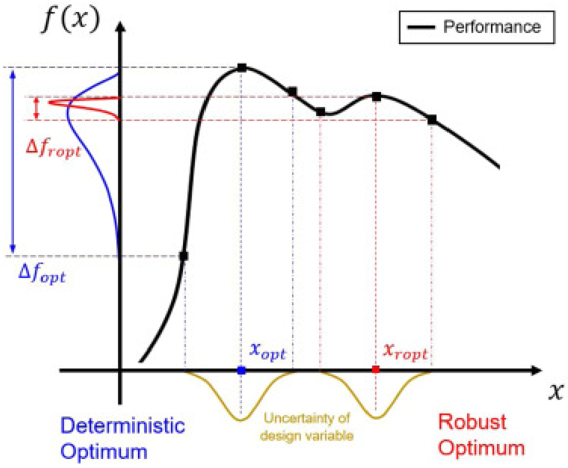


Figure 1.2: *The difference between the optimal design point (blue) and the robust design point (red) of the performance objective function $f(x)$ (black). In this specific case the robust optimum happens to be a local maximum while also displaying lower variation in it's close vicinity, compared to the deterministic optimum, for all the corresponding values of the uncertain variable x.*

Specifically, shape optimization cases of aerodynamic bodies opting to posses an optimized aerodynamic performance within a certain range of their respective design variables, can be established as *aerodynamic robust design* cases. In this category of cases, in order to properly describe the algorithmic steps needed to culminate in a robust design point, the definition of the *aerodynamic robustness metric* (F_R) must be introduced. This metric that represents the robustness of the original objective function (F) a.k.a. the Quantity of Interest (QoI), is dependent upon the statistical moments of F, most commonly referring to: the mean (μ_F) and standard deviation (σ_F) . In RDO cases, the aerodynamic robustness metric replaces the QoI as the quantity to be optimized by either a stochastic or deterministic optimization process. The value of F_R is dependent on the values μ_F and σ_F, which are computed through the Uncertainty Quantification (UQ) process.

The algorithmic steps needed to implement aerodynamic RDO are the following:

- Definition of the initial values of the design and uncertain variable vectors ($\mathbf{b}$ and $\mathbf{c}$ respectively)
- Application of deterministic or stochastic RDO method, utilized to update the design variable vector $\mathbf{b}_{new}$. Either way the statistical moments of the QoI must be computed through a process of UQ for the design variable vector $\mathbf{b}$ and the uncertain variable vector $\mathbf{c}$, under consideration. Thus, the process of numerically solving the flow field equations, in order to compute the flow field variables and the QoI, is executed as part of the UQ algorithm. Upon that, the statistical moments of F, computed through the UQ loop, are utilized to compute the aerodynamic robustness metric F_R.
- Application of an optimization convergence criterion. If the criterion is met, the process terminates by defining the optimal set of the design variables $\mathbf{b}_{opt}$ that result to the robust design point. Otherwise, this process repeats itself for the updated design variable vector $\mathbf{b}_{new}$.

The aforementioned steps of a typical aerodynamic robust design algorithm are, also, visualized through a flow-chart, in Figure 1.3.

More specifically, there is a variety of deterministic methods (e.i. *Method of Moments*) [3], [4] and stochastic (e.i. *Monte Carlo, intrusive or non-intrusive Polynomial Chaos Expansion*)[5], [6], [7], [8], [9] available in order to perform the task of UQ. Stochastic UQ methods determine a certain number values of uncertain variables to be evaluated, thus computing the moments μ_F and σ_F. On the other hand, deterministic UQ methods rely on the formulation of PDEs to be solved, either coupled or uncoupled from the CFD solver, in order to compute the statistical moments of the QoI.

Yet the implementation of the UQ process may still vary on the optimization method used to renew the design variable vector. For stochastic optimization methods the UQ is performed as described above. In contrast, for GBMs the computation of the derivatives of the robustness metric F_R, a.k.a. the robust SDs, is needed to renew the design variables. Subsequently the UQ process must also be implemented for the SDs of the QoI, in order to compute the gradients of the statistical moments of the QoI: $\nabla\mu_F \in \mathbb{R}^N$ and $\nabla\sigma_F \in \mathbb{R}^N$. The gradient of robustness metric ∇F_R is computed using these previously mentioned quantities.

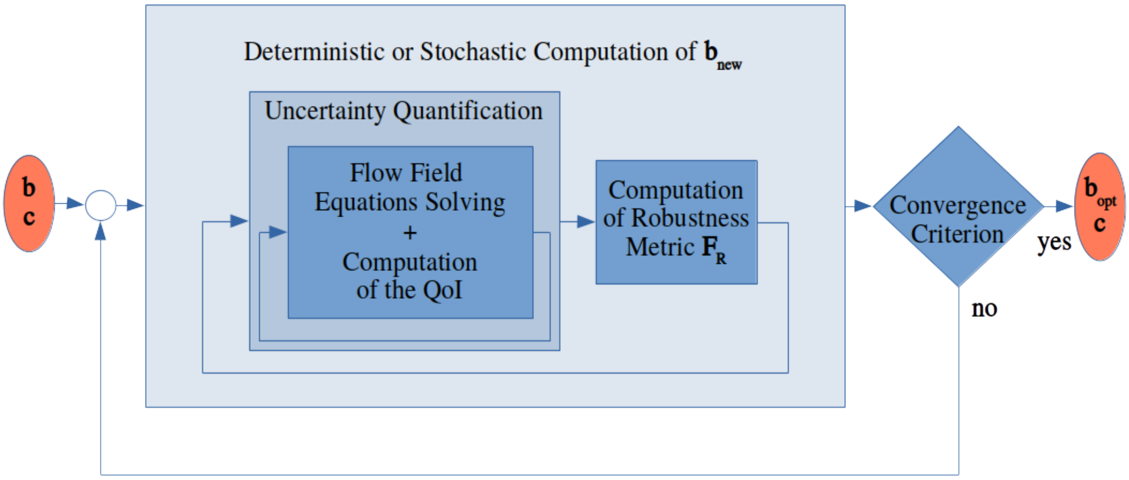

b
c
Deterministic or Stochastic Computation of $\mathbf{b}_{new}$
Uncertainty Quantification
Flow Field Equations Solving + Computation of the QoI
Computation of Robustness Metric $\mathbf{F}_R$
Convergence Criterion
yes
no
$\mathbf{b}_{opt}$
c

Figure 1.3: *The generalized CFD-based RDO algorithm flow-chart.*

In this thesis, the UQ and RDO process is performed through stochastic methods. Specifically, *Monte Carlo* and *non-intrusive Polynomial Chaos Expansion* are employed for UQ, while only the later are used for RDO. More information about these methods is disclosed in Chapter 3. Furthermore, only deterministic methods are implemented in order to compute and renew the design variables of the case under consideration. Hence, for deterministic RDO the computation of the SDs of the QoI is essential. The uncertain variables w.r.t. which the UQ and RDO is performed are considered to be the shape uncertainties, which are generated through the Karhunen-Loève Transform, further explored in Chapter 4.

1.2.1 Computation of Sensitivity Derivatives

Given that this thesis focuses solely on GBMs, this Subsection is dedicated to the different methods available, in order to compute the SD, mentioned in Section 1.1. This step is also instrumental in deterministic RDO methods, therefore the QoI F is displayed as dependent on both design $b_n,\ n \in [1, N]$ and uncertain variables $c_i,\ i \in [1, M]$. In general, the efficiency of GBMs is highly dependent on how the sensitivity derivatives are computed.

The Finite Differences Method

The most straightforward method of computing the SDs is by using *finite differences* (FD) [1]. The computation of a first derivative of F w.r.t. the design variable , then this design variable is perturbated by an infinitesimally small, user-defined quantity, ϵ. Thus, the QoI is re-evaluated for the perturbated design. For instance, a second-order, central difference FD scheme is formulated, as displayed in equation 1.1

$$\frac{\delta F}{\delta b_n} = \frac{F(b_1, ..., b_n + \epsilon, ..., b_N, c_1, ..., c_M) - F(b_1, ..., b_n - \epsilon, ..., b_N, c_1, ..., c_M)}{2\epsilon} \tag{1.1}$$

Despite its simple implementation, since it requires only the re-computation of the value of the objective function, this method poses great concerns because of its two main drawbacks. First and foremost, the cost of the FD method scales linearly with the number of the design variables, N, as it requires $2N$ evaluations of F by solving the flow equations, making it impracticable for optimization problems a large number of design variables. The second downside is the dependence of the computed derivatives from ϵ, the value of which cannot be determined a priori. The use of a too "small" value is not always the answer to the aforementioned problem as it can introduce round-off errors. In addition, for each design variable, the flow equations must be fully converged two additional times in order to compute the perturbated values of F, an event that should not be taken for granted in any CFD case.

The Complex Differences Method

An alternative method for the computation of the SDs is the *complex variable* (CV) method [1] according to which the computation of the SD is executed as

$$\frac{\delta F}{\delta b_n} = \frac{Im[F(b_1, ..., b_n + i\epsilon, ..., b_N, c_1, ..., c_M)]}{\epsilon} \tag{1.2}$$

where Im is the imaginary part of the complex function F and $i =\sqrt{-1}$.

From equation 1.2, it can be assumed that the round-off errors cease to exist, since there is no subtraction of two very close values as in the case of FD. Subsequently, this method is independent from the value of ϵ and, thus, there is no need for the flow equations to be fully converged. Nevertheless, the cost of the complex variable method still scales linearly with N, specifically requiring N evaluations of the QoI.

The Direct Differentiation Method

Another alternative, is the *direct differentiation* (DD) method [1], according to which the flow equations are differentiated w.r.t. the design variables and the N linear systems that arise are solved to define the derivatives of the flow variables w.r.t. to the design variables. Given that the SDs are expressed in terms of these fields, their final computation is straightforward. DD is harder to implement than FD, since a new flow solver has to be developed, increasing its implementation cost.

Moreover, the method's cost still scales with N, making it inadequate for large scale simulations.

The Adjoint Method

The *adjoint* method of computing the sensitivity derivatives required by GBMs is the alternative that has a cost practically independent from the number of the design variables N [1]. As a result, this method is a perfect choice for large industrial optimization and RDO problems. In order to achieve this independence, an augmented objective function is defined, by adding the volume integrals of the residuals of the flow equation (also referred to as the *primal* or *state* equations), multiplied by the adjoint (or *co-state* or *dual*) variable fields, to F. Considering that the residuals of the primal equations must be zero, $F \equiv F_{aug}$. After differentiating the augmented objective function and re-arranging the resulting terms, the system of field adjoint equations (FAE) and adjoint boundary conditions (ABC) is formulated, the numerical solution of which leads to a N-independent computation of the SDs. The numerical solution of the aforementioned system has a computational cost equivalent to the cost of the primal equations' solution.

There are two different approaches [10] on how the aforementioned adjoint method can be applied, that differ from each other in the sequence that the differentiation of the objective function and the discretization of the flow equations happen. In the *discrete* adjoint approach, the residuals of the primal equations that are added to the objective function are in their discrete form and the resulting system of adjoint equations and adjoint boundary conditions after the differentiation is already discretized and ready to be numerically solved. On the other hand, in the *continuous* adjoint approach, the residuals of the primal equations that are added to the objective function are in their continuous form and the resulting system of adjoint equations and the boundary conditions have to be discretized, in order to be numerically solved

There is a general consensus, that both discrete and continuous adjoint methods can produce sensitivity derivatives with sufficient accuracy to be used in common optimization problems. Nevertheless, the discrete approach is more accurate in computing the SD especially on coarse meshes, since it takes the primal discretization schemes into consideration, although its implementation can become cumbersome when higher discretization schemes are used. On the other hand, the continuous adjoint outweighs the discrete one in terms of CPU cost and memory requirements per iteration. Continuous approach also leads to better physical understanding of the adjoint system, since closed-form expressions exist for the field adjoint equations, their boundary conditions and the sensitivity derivatives expression.

In this diploma thesis, the continuous adjoint method is applied in aerodynamic robust design cases so as to compute the SDs of the QoI. The mathematical formulation and software programming of the adjoint solver for incompressible fluid flows has been performed by the PCOpt/NTUA within the OpenFOAM© environment.

1.2.2 Computation of Robustness Metric and its derivatives

Once the UQ process has been completed for the QoI as well as its SDs, the computation of the robustness metric F_R follows.The way of defining the robustness metric depends greatly on the stochastic distribution of the uncertain variables.

A common approach to tackle the uncertainty problem is known as *Design for Six Sigma* (DFSS) [11] [12], which is an engineering design process based on the assumption that every uncertain variable ($c_i,\ i \in [1, M]$) follows a normal distribution around it's mean value μ_i and within a certain range of six standard deviations σ_i of the respective variable c_i. It is a well known fact, according to the mathematical formulation of the normal distribution, that 99.73 % of the values of a stochastic variable, following such a distribution, can be found within the interval $[\mu{-}3\sigma, \mu{+}3\sigma]$ or, in other words, in the *six sigma* range.

Therefore, a way of defining the robust metric $F_R \in \mathbb{R}$ in accordance with the DFSS, is the following

$$F_R = \mu_F + \kappa\,\sigma_F \tag{1.3}$$

where the parameter $\kappa \in [-3, +3]$, is user-defined and its algebraic value determines the approach and the goal of the RDO process.

The absolute value of κ, acts as a weight, determining whether the μ_F or the σ_F is prioritized during the optimization. Meaning that, a small absolute value for κ indicates that an optimized mean value of the objective function is desired, while its variation (quantified through the standard deviation) is of lesser interest. In contrast, a selection of a larger absolute value for κ designates the desire for the robustness metric to display an optimized variation, while it's mean value is being regarded as secondary. Furthermore, the sign of κ indicates whether the designer's approach is pessimistic or optimistic. For example, in a minimization RDO case, a positive sign selection for κ signifies the worst case scenario and, consequently an pessimistic design approach, while a negative sign indicates the opposite.

Likewise, according to the DFSS, the gradient of the robustness metric, a.k.a. the *robust sensitivity derivative vector* $\nabla F_R \in \mathbb{R}^N$ is defined as

$$\nabla F_R = \nabla \mu_F + \kappa\,\nabla \sigma_F \tag{1.4}$$

1.2.3 Design Variable Update Method

There are various methods available to update the design variables b_n , $n \in [1, N]$, always dependent on the computation of either the sensitivity derivatives for optimization problems or the robust sensitivity derivatives for RDO problems. The method used in this thesis and one of the simplest GBMs, implementing 1st order derivatives, is the *steepest descent*.

A general expression for the renewal of the design variable vector $\vec{b} \in \mathbb{R}^N$, in RDO cases, is the following

$$\vec{b}^{new} = \vec{b}^{old} + \eta\, \vec{p} \tag{1.5}$$

where the definition of $\vec{p}$ distinguishes the different update methods, while η serves to scale the step length of each "descent". The value of η can be determined through the expression

$$\eta = \frac{\Delta b_{max}}{\Delta b_{act}} \tag{1.6}$$

where Δb_{max} is the user-defined maximum allowed displacement of the design variables b_i and Δb_{act} is the maximum displacement of each design variable, as computed by equation (1.5) for $\eta = 1$.

Steepest Descent Method

As mentioned, while this method is one of the simplest when it comes to its conception and its implementation, it often lacks efficiency when compared to other 2nd order derivative methods such as the *BFGS* method. Nevertheless, the steepest descent is considered a consistent benchmark method, essential to test an optimization or robust design process in its preliminary stages. This is the reason this method has been selected to meet the needs of this thesis.

In RDO cases, the vector $\vec{p}$ is defined as the gradient of the robustness metric F_R for the old value of the design variables b_n^{old}. As such, $\vec{p}$ is defined as

$$\vec{p} = \pm \nabla F_R^{old} \tag{1.7}$$

where the sign dictates the direction of the optimization, in other words whether it opts to the maximization or minimization of F_R. A positive sign signifies maximization, while a negative implies minimization.

Given that the adjoint solver developed by the PCOpt/NTUA classifies all optimization cases as de facto minimization cases, the incorporated steepest descent scheme takes on the form

$$\vec{b}^{new} = \vec{b}^{old} - \eta\, \nabla F_R^{old} \tag{1.8}$$

1.3 Thesis Objectives and Layout

The goal of this diploma thesis is the aerodynamic robust design of 2D bodies w.r.t. manufacturing imperfections. To be more specific, a code is developed implementing the *Karhunen-Loève Transform*, in order to model the imperfections of the geometries in question. This computational tool is then incorporated into the UQ and RDO processes, thus considering the imperfect geometry of the aerodynamic bodies as the only uncertain variables. The process of UQ is performed through the stochastic methods of non-intrusive Polynomial Chaos Expansion and Monte Carlo, by making use of an in-house code including both methods developed and validated by PCOpt/NTUA. Finally, the aerodynamic RDO employs the adjoint method for the computation of the SDs of the QoI, coupled with the niPCE method to compute the robustness metric F_R and its gradient ∇F_R, in accordance with the DFSS.

Furthermore, the rest of this diploma thesis is structured as follows:

- **Chapter 2**: The mathematical formulation and the numerical solution of the flow field, a.k.a. the *primal* equations and their respective *adjoint* equations is described, for the purpose of computing the SDs of the desired QoI. Additionally, a description of the grid displacement strategy following the displacement of the shape during the process of RDO, is presented.
- **Chapter 3**: The mathematical theory concerning the stochastic uncertainty quantification methods *Monte Carlo* and *non-intrusive Polynomial Chaos Expansion* is introduced and analyzed, for both one and multiple uncertain variables. For the second method, both *Full Grid* and *Smolyak/Sparse Grid* integration methods are included.
- **Chapter 4**: The mathematical formulation of the Karhunen-Loève Transform will be presented. In addition, a description of the algorithmic steps used to guide the development of software implementing the aforementioned transform for the the recreation of manufacturing imperfections in the shape of 2D aerodynamic bodies, will take place. Also, two application of the Karhunen-Loève Transform will be included.
- **Chapter 5**: The software generating Karhunen-Loève Transform shape uncertainties as well as OpenFOAM grid generators and flow solvers are incorporated in the in-house niPCE code for the purpose of performing UQ on 2D geometries with shape uncertainties. Moreover, the aforementioned coupled algorithm is applied to the NACA 0012 isolated airfoil, thus performing UQ for two QoI: the *lift* and *drag coefficients.* Furthermore, UQ is performed on the E387 airfoil with shape uncertainties, in order to compute the robust SDs of the drag coefficient. In both cases, cross-reference between the non-intrusive Polynomial Chaos Expansion and the Monte Carlo computed results take place, so as to validate the method.
- **Chapter 6**: Integration of the continuous adjoint solver and of a parameterized

mesh movement strategy into the aforementioned coupled algorithm. This all-encompassing code is implemented to perform aerodynamic RDO on the isolated E387 airfoil and the TU Berlin compressor stator cascade [13], considering the recurring KLT-modeled shape imperfections to be the uncertain variables of the problem.

Chapter 2

CFD Analysis and Adjoint Problem Formulation

In this chapter the mathematical formulation of steady state incompressible flow field equations and their respective adjoint field equations, as well as the numerical methods utilized to achieve their solution are thoroughly presented. This part is needed in order to further clarify the process used to compute the QoI and the SDs, for the needs of UQ and RDO, respectively.

2.1 Primal Problem

2.1.1 Flow Field Modeling

All CFD cases dealt with in this thesis, are flows around 2D aerodynamic bodies such as airfoils or turbomachinery blades. Such flows, if not highly turbulent beforehand, then turbulence is most likely to develop close to the solid surfaces and in their wake. Turbulence in a flow is defined by the semi-random development of unsteady pressure and velocity fluctuations, creating coherent vortex structures named eddy vortices.

As a general rule, the *Navier-Stokes* (NS) equations can fully predict viscous, steady and unsteady flows within a certain domain with defined boundary conditions. Yet the prediction of turbulent flow phenomena in their whole spatial and temporal scale spectrum, an infinitely small cell size during meshing, as well as painstakingly small time domain discretization. Such simulations, called direct numerical simulations, in which the NS are numerically solved without any turbulence modeling, has an unbearable computational and clock-time cost, thus making them unfeasible in most cases with limited computational resources. Specifically for the needs of this diploma thesis, the employment of DNS is unacceptable since the solution of the flow field, a.k.a. *primal* problem, is part of a larger procedure, either UQ or RDO.

A more widespread approach for the simulation of turbulent flows is the use a method first proposed by Reynolds in 1985, the *Reynolds-Averaged Navier-Stokes* (RANS) equations [1, 14, 15]. The concept of this method, still valid today, is based on the decomposition of the flow field variables into their mean and their fluctuating components. Thus, the random perturbation of the flow variables caused by the turbulence of the flow is taken into account through the use of *turbulence model* equation, thus bypassing the costly DNS for turbulent flows. For this thesis, the system of the RANS and turbulence model equations constitute the *primal problem.*

In accordance with the Einstein convention, for which repeated indices imply summation, the RANS system of equation for incompressible steady flows is presented in equations (2.1),(2.2) in non-conservative form:

- The conservation of mass, a.k.a. continuity equation is

$$\frac{\partial \overline{v}_j}{\partial x_j} = 0 \tag{2.1}$$

- The conservation of momentum equation is

$$\overline{v}_j \frac{\partial \overline{v}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\nu\left(\frac{\partial \overline{v}_i}{\partial x_j} + \frac{\partial \overline{v}_j}{\partial x_i}\right)\right] + \frac{\partial}{\partial x_j}\left(-\overline{v_i' v_j'}\right) \quad , \quad i = 1,2,3 \tag{2.2}$$

where $\overline{v_i}$ indicate the mean velocity components, $\overline{p}$ stands for the mean pressure and $\rho = \overline{\rho}$ designates the constant density of the fluid. An overbar '$^-$' symbolizes the mean value , while the $''$ symbolizes the perturbation of a flow variable. In equation (2.2), the only term including flow variable fluctuations is the *Reynolds stress* or *turbulent shear stress tensor* $\tau_{ij}'/\rho = -\overline{v_i' v_j'}$. In order to fully get rid of velocity fluctuations that appear only in the RANS equations, an expression modeling the Reynolds stresses as variables wholly dependent on the mean velocity components is needed. One way to overcome any reference to the turbulent fluctuations comes through the acceptance of the *Boussinesq Hypothesis* [14],[15], suggested by Boussinesq in 1877 and still widely adopted today:

$$\tau_{ij}'/\rho = -\overline{v_i' v_j'} = \nu_t \left[\left(\frac{\partial \overline{v}_i}{\partial x_j} + \frac{\partial \overline{v}_j}{\partial x_i}\right) - \frac{2}{3}\, k\, \delta_i^j\right] \tag{2.3}$$

where δ_i^j indicates the Kronecker delta, ν_t the turbulent kinematic viscosity, a.k.a. eddy viscosity (measured in m^2/s) and k the turbulent kinetic energy, defined as

$$k = \frac{1}{2}\overline{v_i' v_i'} \tag{2.4}$$

Hence, after including the Boussinesq hypothesis (2.3) into the momentum equation (2.2), the Reynolds stress tensor is expressed in terms of the turbulent viscosity ν_t, as a new unknown field variable. Thus the conservation of momentum equation is expressed by

$$\overline{v}_j \frac{\partial \overline{v}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[(\nu + \nu_t)\left(\frac{\partial \overline{v}_i}{\partial x_j} + \frac{\partial \overline{v}_j}{\partial x_i}\right)\right] \ , \quad i = 1, 2, 3 \tag{2.5}$$

The introduction of the aforementioned ν_t field causes the need of one more equation, so as to close the system. This closure problem can be treated with the addition of algebraic or differential turbulence models in the system of primal equations [14]. Given that turbulence models do not directly simulate the turbulent fluctuations, they are imbued with decades of experimental data on turbulent flows. Therefore, some models can be more fine-tuned than others for use in certain application fields, i.e. the $k-\epsilon$ model is genrally preferred in heat transfer applications. The turbulence models used to compute the turbulent viscosity, therefore taking the Boussinesq hypothesis, into account, are referred to as eddy viscosity models (EVMs). Such a model is the one used exclusively in this thesis, the *Spalart–Allmaras* turbulence model [16], presented in Subsection 2.1.2.

2.1.2 The Spalart–Allmaras Turbulence Model

The Spalart–Allmaras is a relatively simple and low cost mixing length model, implementing one transport partial differential equation (PDE) for the computation of the turbulent viscosity ν_t. It is calibrated on empirical data from 2D wall-bounded flows such as flat plates, wakes and mixing layers [16]. The model provides satisfactory predictions for boundary layers with severe pressure gradients, as well as fair to good results in aerospace applications, such as airfoils and wings [16]. Its governing PDE is assembled as a function of the Spalart–Allmaras variable $\widetilde{\nu}$ and is defined as

$$v_j \frac{\partial \widetilde{\nu}}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\widetilde{\nu}}{\sigma}\right)\frac{\partial \widetilde{\nu}}{\partial x_j}\right] - \frac{c_{b2}}{\sigma}\left(\frac{\partial \widetilde{\nu}}{\partial x_j}\right)^2 - \widetilde{\nu}\, P\left(\widetilde{\nu}\right) + \widetilde{\nu}\, D\left(\widetilde{\nu}\right) = 0 \tag{2.6}$$

where the first and second terms on the LHS of the PDE (2.6) correspond to the convection and diffusion (bulk and turbulent) terms of $\widetilde{\nu}$, while ν_t being the turbulent viscosity, formulated as a function of the later

$$\nu_t = f_{v1}\, \widetilde{\nu} \tag{2.7}$$

In addition, the production $P(\widetilde{\nu})$ and dissipation terms are modeled as

$$P(\widetilde{\nu}) = c_{b1}\widetilde{Y} \ , \qquad D(\widetilde{\nu}) = c_{w1} f_w(\widetilde{Y}) \frac{\widetilde{\nu}}{\Delta^2} \tag{2.8}$$

where $\widetilde{Y}$ is given by

$$\widetilde{Y} = Y f_{v_3} + \frac{\widetilde{\nu}}{\Delta^2 \kappa^2} f_{v_2} \ , \qquad Y = \left| e_{ijk} \frac{\partial v_k}{\partial x_j} \right| \tag{2.9}$$

with Y standing for the vorticity magnitude and Δ being the distance of cell of vertex centres from the wall boundaries. Given that OpenFOAM uses a finite volume cell-centered discretization of the governing equations [17], Δ represents the respective distance of cell centres.

The model functions read

$$\begin{aligned}
f_{v_1} &= \frac{\chi^3}{\chi^3 + c_{v_1}^3} \ , \quad f_{v_2} = \frac{1}{\left(1 + \frac{\chi}{c_{v_2}}\right)^3} \\
f_{v_3} &= \frac{(1 + \chi f_{v1})}{c_{v2}} \left[3\left(1 + \frac{\chi}{c_{v2}}\right) + \left(\frac{\chi}{c_{v2}}\right)^2 \right] \left(1 + \frac{\chi}{c_{v2}}\right)^{-3} \\
\chi &= \frac{\widetilde{\nu}}{\nu} \ , \quad f_w = g \left(\frac{1 + c_{w_3}^6}{g^6 + c_{w_3}^6} \right)^{1/6} \\
g &= r + c_{w_2}(r^6 - r) \ , \quad r = \frac{\widetilde{\nu}}{\widetilde{Y} \kappa^2 \Delta^2}.
\end{aligned} \tag{2.10}$$

The constants of the model are $c_{b1} = 0.1355$, $c_{b2} = 0.622$, $\kappa = 0.41$, $\sigma = 2/3$, $c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{(1+c_{b2})}{\sigma}$, $c_{w2} = 0.3$, $c_{w3} = 2$, $c_{v1} = 7.1$ and $c_{v2} = 5$. The Levi–Civita symbol, e_{ijk}, used in the vorticity magnitude Y, is

$$e_{ijk} = \begin{cases} +1 & (i,j,k) \in (1,2,3),\ (2,3,1),\ (3,1,2) \\ -1 & (i,j,k) \in (1,3,2),\ (3,2,1),\ (2,1,3) \\ 0 & i = j,\ j = k,\ k = i \end{cases} \tag{2.11}$$

2.1.3 The Low and High Reynolds Number Models

The Spalart–Allmaras model as described until now, is sufficient in areas of fully developed turbulent flow, far from solid-fluid interaction, where turbulent shear stresses dominate over bulk stresses. Yet, near the solid boundary, where viscous phenomena thrive, specified terms need to be added to the turbulence model, in order to better simulate the effect of such phenomena.

One approach is the *Low Reynolds number* model, according to which the laminar or viscous sublayer of the boundary layer is resolved numerically by including the effects of bulk viscosity into the formulation of turbulence model in use, in the form of near wall damping additional terms. This method requires very dense grids near the wall to produce adequate results, due to the steep velocity gradients appearing in the viscous sublayer and the buffer zone, as displayed in 2.2. This method though accurate when properly implemented is quite costly.

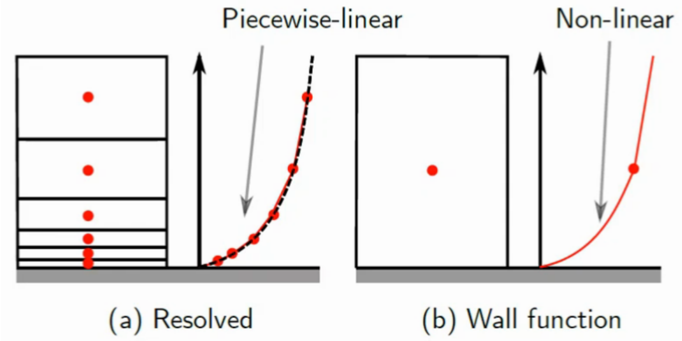


Figure 2.1: *Comparison of near-wall mesh quality between Low-Re (a) and High-Re number (b) turbulence models.*

An alternative solution and the one adopted for the CFD analysis present this thesis, is referred to as *High Reynolds number* turbulence model. In this method, wall functions, meaning analytical expressions combined with experimental data are introduced for the computation of the mean velocity on the cell centres of the near-wall regions. For this method, the distance of the first cell centre off the wall, should lie in the logarithmic region of the turbulent boundary layer, hence the value of the non-dimensional wall distance y^+ (defined in (2.12)) can be up to 100, in order not to compromise the method's accurately.

$$y^+ = \frac{u_T\, y}{\nu} \ , \quad u_T = \sqrt{\frac{\tau_w}{\rho}} \tag{2.12}$$

where y is the cell centre from the wall and τ_w is the wall stress.

Consequently, given that the laminar sublayer is not resolved, the near-wall meshing does not necessarily need to be as fine as that required in the previously mentioned Low-Re model, as displayed in Figure 2.1. In the Spalart–Allmaras model, the wall functions, used to approximate the value of ν_t at the cell closest to the wall, are formulated as $\nu_t = \frac{u_\tau^2}{\partial U/\partial n}$, where u_τ is computed based on the y^+ value. The formulation for the computation of y^+, as programmed in OpenFOAM©, follows *Spalding's Law* [18]. This models the inner sublayer and the logarithmic region of the boundary layer with a single equation:

$$y^+ = v^+ + e^{-\kappa B}\left[e^{\kappa v^+} - 1 - \kappa v^+ - \frac{(\kappa v^+)^2}{2} - \frac{(\kappa v^+)^3}{6}\right] \tag{2.13}$$

where κ is the von-Karman constant equal to 0.41 and $B \approx 5.5$.

This equation came as a result of best fit between the curve of $y^+ = u^+$ which is valid in the viscous sublayer and $u^+ = Ey^+/\kappa$ which is valid in the logarithmic

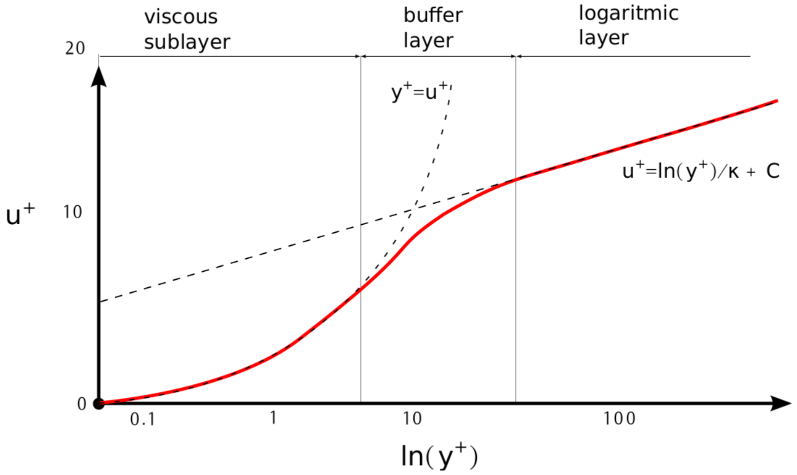


Figure 2.2: *The non-dimensional velocity u^+ profile w.r.t. the non-dimensional wall distance y^+ spanning form the solid wall to the log-law of a turbulent boundary layer.*

region. $E = e^{-\kappa B}$ is an empirical constant equal to 9.793.

2.1.4 Primal Equations and Boundary Conditions

All the previous analysis culminates in the full determination of the primal problem equations used to simulate the a steady-state turbulent flow of an viscous, incompressible fluid, by rearranging equations (2.1),(2.5) and (2.6).

$$R^p = -\frac{\partial v_j}{\partial x_j} = 0 \tag{2.14a}$$

$$R_i^v = v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial}{\partial x_j}\left[(\nu + \nu_t)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right] + \frac{\partial p}{\partial x_i} = 0 \;, \qquad i = 1,2,3 \tag{2.14b}$$

$$R^{\widetilde{\nu}} = v_j \frac{\partial \widetilde{\nu}}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\widetilde{\nu}}{\sigma}\right)\frac{\partial \widetilde{\nu}}{\partial x_j}\right] - \frac{c_{b2}}{\sigma}\left(\frac{\partial \widetilde{\nu}}{\partial x_j}\right)^2 - \widetilde{\nu}\, P\left(\widetilde{\nu}\right) + \widetilde{\nu}\, D\left(\widetilde{\nu}\right) = 0 \tag{2.14c}$$

Last but not least, the boundary conditions needed for the closure of the primal problem are presented, in a generalized formulation for 2D computational meshes exclusively used in the present thesis, as follows:

- **Inlet**: at the inlet to the computational domain, Dirichlet boundary conditions are imposed on the velocity components v_i according to the wanted freestream velocity value and zero-Newmann condition is used on the pressure p. For the Spalart–Allmaras model variable $\widetilde{\nu}$, a Dirichlet condition is imposed.
- **Outlet**: at the exit boundary of the computational domain, zero-Neumann boundary conditions are imposed on the velocity components v_i and the Spalart–Allmaras variable $\widetilde{\nu}$, while zero-Dirichlet conditions are imposed on the pressure p.

- **Solid Walls**: at the wall boundaries of the domain, zero-Dirichlet, a.k.a. *no-slip*, boundary conditions are imposed for v_i and zero-Neumann on p. Finally, a zero-Dirichlet boundary condition is utilized for $\widetilde{\nu}$.

The aforementioned boundaries are visualized for computational meshes around airfoils in Figure 2.3.

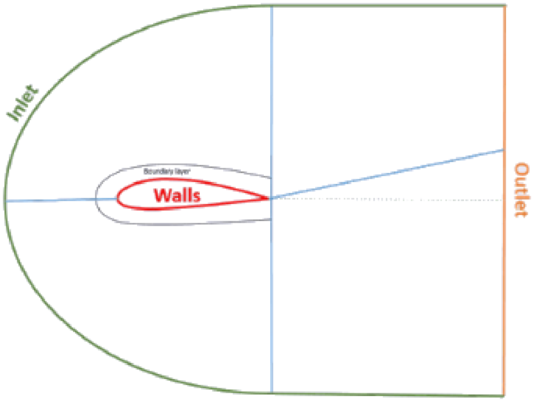


Figure 2.3: *Generalized 2D boundaries of computational domains around isolated airfoils.*

Outside of Figure 2.3, for meshes generated in turbomachinery blade cascades, one more boundary condition must be imposed. In the mesh boundary above and below the blades, periodic conditions are imposed, according to which all the primal problem variables v_i, p, $\widetilde{\nu}$ on each and every cell centre are mirrored between these two boundaries.

2.2 Adjoint Problem

As previously mentioned, the continuous adjoint method is used in this thesis for the computation of the SDs of a certain objective function F, referred to as QoI for the purposes of UQ and RDO. These SDs are the derivatives of F w.r.t. the design variables of the shape optimization case in question. This is achieved by adding to F the volume integrals of the primal problem's PDEs multiplied with the adjoint variable fields, thus creating the augmented objective function F_{aug}. The key feature of the adjoint solver, developed by PCOpt/NTUA and utilized in this thesis, is that it does not neglect the variations of the turbulent viscosity ν_t as often assumed in literature. This assumption, according to which the only the mean flow quantities are affected by the shape, may lead to inaccurate SDs and seriously affect the optimization process [19]. The adjoint model formulated by taking this assumption into account is referred to as *frozen turbulence* model and culminates to system of adjoint equation without the adjoint to the turbulence model PDEs. The more meticulous approach, followed by PCOpt/NTUA, requires the differentiation of the turbulence model PDEs w.r.t. the design variables b_n, consequently reaching to the formulation and solution of the adjoint to turbulence model.

2.2.1 The Three Continuous Adjoint Formulations

According to the literature, three distinct approaches are available for the formulation of the continuous adjoint method: the FI, the SI and the E-SI approaches. All approaches culminate to the same expression for the *field adjoint equations* (FAE) and the *adjoint boundary conditions* (ABS). Yet, they differ on the final expression for the SDs of F.

The FI method, being the first chronologically to be proposed, includes both boundary and field integrals in the formulation of the SDs, hence the name *Field Integrals* (FI). It is clearly the most accurate as well as the most costly of the three approaches, due to the integration of the entire field and the need of computing the *grid* or *mesh sensitivities* $\delta x_k/\delta b_n$ [20]. The second formulation, known as *reduced gradient*, takes its name from the fact that only boundary, a.k.a. *Surface Intervals* (SI approach) are contained in the SD formulation, getting rid of the costly computation of the grid sensitivities. Consequently this approach is characterized by a low computational cost, especially for problems with many design variables, making it an interesting prospect. However this simplification does not guarantee an accurate SD prediction [21], particularly for coarse meshes. Finally, the third approach, the *Enhanced Surface Integrals* (E-SI), combines the advantages of both previous formulation, producing accurate SDs at a reduced computational cost [21].

For a more detailed analysis on the three aforementioned formulations, the reader is pointed to the bibliography [20] and [21]. Though all adjoint formulations are briefly analyzed, only the E-SI approach is implemented in this thesis, so as to compute the SDs, and thus it is described more thoroughly.

The generalized augmented objective function is defined, according to the Einstein convention, as follows

$$F_{aug} = F + \int_{\Omega} \Psi_i \, R_i \, d\Omega \tag{2.15}$$

where $R_i \approx 0$ the residuals of the state PDEs, Ψ_i their corresponding adjoint variables, $i = 1, 2, ..., E$ with E the number of state equations, while Ω the computational domain. By differentiating w.r.t. the design variable b_n, the total derivatives $\delta/\delta b_n$ of F_{aug} appear

$$\frac{\delta F_{aug}}{\delta b_n} = \frac{\delta F}{\delta b_n} + \frac{\delta}{\delta b_n} \int_{\Omega} \Psi_i \, R_i \, d\Omega \tag{2.16}$$

The different ways the total derivative of the integral in eq. (2.16) is expanded, cause the distinct FI and SI (enhanced or not) formulations.

The FI Approach

The FI approach dictates the development of the aforementioned term as follows

$$\frac{\delta}{\delta b_n}\int_\Omega \Psi_i\, R_i\, d\Omega = \int_\Omega \Psi_i\, \frac{\delta R_i}{\delta b_n}\, d\Omega + \int_\Omega \Psi_i\, R_i\, \frac{\delta(d\Omega)}{\delta b_n} \tag{2.17}$$

According to [19], the derivative of $d\Omega$ domain in the RHS of eq. (2.17) assumes the form

$$\frac{\delta(d\Omega)}{\delta b_n} = \frac{\partial}{\partial x_k}\left(\frac{\delta x_k}{\delta b_n}\right) d\Omega \tag{2.18}$$

Therefore, the inclusion of eq. (2.17) and (2.18) in eq. (2.16) yields

$$\left.\frac{\delta F_{aug}}{\delta b_n}\right|_{FI} = \frac{\delta F}{\delta b_n} + \int_\Omega \Psi_i\, \frac{\delta R_i}{\delta b_n}\, d\Omega + \int_\Omega \Psi_i\, R_i\, \frac{\partial}{\partial x_k}\left(\frac{\delta x_k}{\delta b_n}\right) d\Omega \tag{2.19}$$

Eq. (2.19) comprises the basis for the FI adjoint formulation and contains the grid variations of x_k. By developing the terms in eq. (2.19) according to the primal problem state equations described in Subsection 2.1.4, the derivative of F_{aug} is formulated as

$$\begin{aligned}\left.\frac{\delta F_{aug}}{\delta b_n}\right|_{FI} &= \frac{\delta F}{\delta b_n} + \int_\Omega \left(u_i \frac{\delta R_i^v}{\delta b_n} + q\frac{\delta R^p}{\delta b_n} + \widetilde{\nu_a}\frac{\delta R^{\tilde{\nu}}}{\delta b_n}\right) d\Omega\, + \\ &+ \int_\Omega \left(u_i R_i^v + q R^p + \widetilde{\nu_a} R^{\tilde{\nu}}\right) \frac{\partial}{\partial x_k}\left(\frac{\delta x_k}{\delta b_n}\right) d\Omega\end{aligned} \tag{2.20}$$

where u_i the adjoint velocity components, q the adjoint pressure and $\widetilde{\nu_a}$ the adjoint Spalart–Allmaras variable, whereas R_i^v, R^q and $R^{\tilde{\nu}}$ the residuals of the primal problem PDEs, as presented in (2.14).

Since the residuals of the primal equations must be close to zero over the whole domain, $F_{aug} = F$ and consequently $\frac{\delta F_{aug}}{\delta b_n} = \frac{\delta F}{\delta b_n}$. The development of the total derivatives of R_i^v, R^q and $R^{\tilde{\nu}}$ w.r.t. to b_n yields the corresponding derivatives of the state variables v_i, p, τ_{ij} (stress tensor components) and $\tilde{\nu}$ as well as their corresponding spatial derivatives [21]. By differentiating the objective function F w.r.t. to b_n and by employing the chain rule, the eq. (2.21) proven in [19], as well as the Green-Gauss theorem, integrals of expressions multiplied by $\delta v_i/\delta b_n$, $\delta p/\delta b_n$ or $\delta\tilde{\nu}/\delta b_n$ arise.

$$\frac{\delta}{\delta b_n}\left(\frac{\partial \Phi}{\partial x_k}\right) = \frac{\partial}{\partial x_k}\left(\frac{\delta \Phi}{\delta b_n}\right) - \frac{\partial \Phi}{\partial x_k}\frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right) \tag{2.21}$$

The SI Approach

The appearance of the spatial gradients of the grid sensitivities $\frac{\partial}{\partial x_k}\left(\frac{\delta x_k}{\delta b_n}\right)$ is the main drawback of the FI formulation, since their numerical computation by the FD in clock-time expensive for a large number of design variables [22]. The SI method circumvents this drawback by applying the Leibniz theorem for the differentiation of volume integrals with variable boundaries. For a quantity $\Phi \in \Omega$ and controlled boundaries $S = S(\vec{b}) = \partial\Omega$, the application of the Leibniz theorem yields

$$\frac{\delta}{\delta b_n}\int_\Omega \Phi \delta\Omega = \int_\Omega \frac{\partial \Phi}{\partial b_n}\delta\Omega + \int_S \Phi n_k \frac{\delta x_k}{\delta b_n}\delta S \tag{2.22}$$

With the use of the Leibniz theorem on eq.(2.16), it yields

$$\left.\frac{\delta F_{aug}}{\delta b_n}\right|_{SI} = \frac{\delta F}{\delta b_n} + \int_\Omega \Psi_i \,\frac{\delta R_i}{\delta b_n}\, d\Omega + \underbrace{\int_S \Psi_i \; R_i \; n_k \,\frac{\delta x_k}{\delta b_n} dS}_{LB-term} \tag{2.23}$$

Eq. (2.23) constitutes the base of the SI method and in its RHS the last integral is labeled Leibniz (LB) term. In literature [23], the LB-term is often neglected in SI adjoint formulation, assuming that the primal equations are satisfied also close to the moving boundaries of the computational domain. This indeed happens in fine grids, where this assumption may not compromise the accurate computation of the SDs. In contrast, depending on the case or the grid's coarseness along the boundary, the inclusion of the LB-term may be critical for the correct computation of the SDs. An interesting proposition of replacing the LB-term with the expression present in eq. (2.24), is proven in [23]

$$\begin{aligned}\int_S \Psi_i R_i n_k \frac{\delta x_k}{\delta b_n} dS = -\int_\Omega \frac{\partial}{\partial x_j}\left(-u_i v_j \frac{\partial v_i}{\partial x_k} - u_j \frac{\partial p}{\partial x_k} - \right. \\ \left. -\tau^a_{ij}\frac{\partial v_i}{\partial x_k} u_i \frac{\partial \tau_{ij}}{\partial x_k} + q\frac{\partial v_j}{\partial x_k}\right)\frac{\delta x_k}{\delta b_n}\delta\Omega\end{aligned} \tag{2.24}$$

where $\tau_{ij} = \nu\left(\frac{\partial v_i}{\partial x_j}\frac{\partial v_j}{\partial x_i}\right)$ is the stress tensor and $\tau^a_{ij} = \nu\left(\frac{\partial u_i}{\partial x_j}\frac{\partial u_j}{\partial x_i}\right)$ is the adjoint stress tensor.

This adaptation of the SI approach can yield adequate results, rivaling those of the FI approach [23]. However, an inclusion of this transformed LB-term a.k.a. the V-term requires the computation of $\partial x_k/\partial b_n$ in Ω, causing the method's cost to scale linearly with the number of design variables, thus leading to a cost comparable to that of the SI method.

The E-SI Approach

The E-SI formulation aims to abolish the computation the field integrals containing the $\partial x_k/\partial b_n$ terms, by solving the adjoint to a hypothetical grid displacement PDE [22], [23]. A *Laplace* equation is assumed to be the grid displacement PDE (gdPDE) for this thesis. This grPDE is formulated as follows

$$R_i^m = \frac{\partial^2 m_i}{\partial x_j^2} = 0 \tag{2.25}$$

where m_i are the Cartesian displacements of the grid nodes. Along the boundary, m_i represents the displacement of the boundary points. Following the adjoint methodology, to derive the adjoint gdPDE, a new term is added to the augmented function of eq. 2.15, containing the field integral of the laplacian grid displacement PDE multiplied by the adjoint to m_i variable. The resulting $\delta/\delta b_n$ field integrals are expanded using the Leibniz theorem.

The extra field integral of the laplacian grid displacement PDE is also included since the analysis is based on the E-SI continuous adjoint approach.

$$F_{aug} = F + \int_\Omega u_i R_i^v d\Omega + \int_\Omega q R^p d\Omega + \int_\Omega \widetilde{\nu_a} R^{\widetilde{\nu}} d\Omega + \int_\Omega m_i^a R_i^m d\Omega \tag{2.26}$$

where Ω is the computational domain, u_i the adjoint velocity, q the adjoint pressure, $\widetilde{\nu_a}$ the adjoint turbulence (or adjoint Spalart-Allmaras) variable and m_i^a the adjoint to m_i variable. It should be noted that the third integral of eq. 2.26 would be excluded if the 'frozen turbulence' assumption were made.

By employing the Leibniz and the Green-Gauss theorem we receive

$$\left.\frac{\delta F_{aug}}{\delta b_n}\right|_{E-SI} = \frac{\delta F}{\delta b_n} + \frac{\delta}{\delta b_n}\int_\Omega \left(u_i R_i^v + q R^p + \widetilde{\nu_a} R^{\widetilde{\nu}} + m_i^a R_i^m\right) d\Omega \tag{2.27}$$

$$\begin{aligned} &= \frac{\delta F}{\delta b_n} + \int_\Omega u_i \frac{\partial R_i^v}{\partial b_n} d\Omega + \int_\Omega q \frac{\partial R^p}{\partial b_n} d\Omega + \int_\Omega \widetilde{\nu_a} \frac{\partial R^{\widetilde{\nu}}}{\partial b_n} d\Omega \\ &+ \int_S m_i^a n_j \frac{\partial}{\partial x_j}\left(\frac{\delta x_i}{\delta b_n}\right) dS - \int_{S_W} \frac{\partial m_i^a}{\partial x_j} n_j \frac{\delta x_i}{\delta b_n} dS + \int_\Omega \frac{\partial^2 m_i^a}{\partial x_j^2} \frac{\delta x_i}{\delta b_n} \delta\Omega \\ &+ \int_S \left(u_i R_i^v + q R^p + \widetilde{\nu_a} R^{\widetilde{\nu}} + m_i^a R_i^m\right) n_k \frac{\delta x_k}{\delta b_n} dS \end{aligned} \tag{2.28}$$

where S is the boundary of the computational domain, $S = S_I \cup S_O \cup S_W \cup S_{W_P}$. The boundaries S_I, S_O, S_W and S_{W_P} refer to the inlet, outlet, fixed and controlled (thus parameterized) wall boundaries of the domain, respectively. Also, n_k stands for the components of the unit outward vector which is normal to the surface. Since the only parameterized boundary is S_{W_P} and for the non-controlled boundaries $\delta x_k/\delta b_n = 0$, we have

$$\left.\frac{\delta F_{aug}}{\delta b_n}\right|_{E-SI} = \frac{\delta F}{\delta b_n} + \int_{\Omega} u_i \frac{\partial R_i^v}{\partial b_n} d\Omega + \int_{\Omega} q \frac{\partial R^p}{\partial b_n} d\Omega + \int_{\Omega} \widetilde{\nu}_a \frac{\partial R^{\widetilde{\nu}}}{\partial b_n} d\Omega$$
$$+ \int_{S_{W_P}} m_i^a n_j \frac{\partial}{\partial x_j}\left(\frac{\delta x_i}{\delta b_n}\right) dS - \int_{S_{W_P}} \frac{\partial m_i^a}{\partial x_j} n_j \frac{\delta x_i}{\delta b_n} dS + \int_{\Omega} \frac{\partial^2 m_i^a}{\partial x_j^2} \frac{\delta x_i}{\delta b_n} \delta\Omega$$
$$+ \int_{S_{W_P}} \left(u_i R_i^v + q R^p + \widetilde{\nu}_a R^{\widetilde{\nu}} + m_i^a R_i^m\right) n_k \frac{\delta x_k}{\delta b_n} dS \tag{2.29}$$

The FAE are derived by zeroing the terms that multiplied with the aforementioned state variable derivatives in the volume integrals of eq. 2.20, while the ABC are derived by zeroing these terms that manifest in the respective surface integrals. The remaining terms yield the final formulation for the SDs. For a more generalized expression for the SDs, achieved through the FI formulation, the reader is referred to [21].

At this point, a sharp distinction must be made between symbols $\delta()/\delta b_n$ and $\partial()/\partial b_n$. $\delta\Phi/\delta b_n$ denotes the total (or material) derivative of an arbitrary quantity Φ and represents the total change in Φ by varying b_n, whereas $\partial\Phi/\partial b_n$ denotes the partial derivative of Φ and represents the variation in Φ due to changes in the flow variables excluding the contributions from the space deformation. $\delta\Phi/\delta b_n$ and $\partial\Phi/\partial b_n$ are related with the following expression depending if they are computed on the interior of Ω or on the boundary of Ω.

Interior of Ω

$$\frac{\delta\Phi}{\delta b_n} = \frac{\partial\Phi}{\partial b_n} + \frac{\partial\Phi}{\partial x_k}\frac{\delta x_k}{\delta b_n} \tag{2.30}$$

Surface - Boundary of Ω

$$\frac{\delta_s\Phi}{\delta b_n} = \frac{\partial\Phi}{\partial b_n} + \frac{\partial\Phi}{\partial x_k} n_k \frac{\delta x_m}{\delta b_n} n_m \tag{2.31}$$

Before proceeding with analysing the integrals appearing on the RHS of eq. (2.20) the following observation must be made. Since $\partial()/\partial b_n$ takes into account only changes in the flow variables and excludes changes in the shape/volume of the flow domain, spatial differentiation and partial differentiation w.r.t. the design variables can commute, i.e.

$$\frac{\partial}{\partial b_n}\left(\frac{\partial\phi}{\partial x_j}\right) = \frac{\partial}{\partial x_j}\left(\frac{\partial\phi}{\partial b_n}\right) \tag{2.32}$$

In general, this is not valid for the total derivative, i.e.

$$\frac{\delta}{\delta b_n}\left(\frac{\partial\phi}{\partial x_j}\right) \neq \frac{\partial}{\partial x_j}\left(\frac{\delta\phi}{\delta b_n}\right) \tag{2.33}$$

2.2.2 Differentiation of the Objective Function

In this thesis, two different objective functions are used as QoI for the purposes of UQ or RDO. These objective functions are defined either on volume of the computational domain, either specified surfaces of it. Therefore, a generalized objective function F formulation, encompassing both surface and volume integrals, with S and Ω their respective domains, can be defined as

$$F = \int_{\Omega} F_{\Omega} d\Omega + \int_{S} F_{S_i} n_i dS \tag{2.34}$$

where n_i the unit surface normal vector, while F_{S_i} and F_{Ω} the integrands on either a surface or a volume of the domain, respectively. The differentiation of F w.r.t. b_n yields

$$\frac{\delta F}{\delta b_n} = \frac{\delta}{\delta b_n} \int_{\Omega} F_{\Omega} d\Omega + \frac{\delta}{\delta b_n} \int_{S} F_{S_i} n_i dS \tag{2.35}$$

The differentiation of the surface integral on the RHS of eq. (2.35) can be expanded as follows

$$\begin{aligned} \frac{\delta}{\delta b_n} \int_{S} F_{S_i} n_i dS = & \int_{S} \left(\frac{\partial F_{S_i}}{\partial p} \frac{\partial p}{\partial b_n} + \frac{\partial F_{S_i}}{\partial v_k} \frac{\partial v_k}{\partial b_n} + \frac{\partial F_{S_i}}{\partial \tau_{kj}} \frac{\partial \tau_{kj}}{\partial b_n} + \frac{\partial F_{S_i}}{\partial \tilde{\nu}} \frac{\partial \tilde{\nu}}{\partial b_n} \right) n_i dS + \\ & + \int_{S} n_i \frac{\partial F_{S_i}}{\partial x_k} \frac{\partial x_k}{\partial b_n} n_k dS + \int_{S} F_{S_i} \frac{\partial n_i}{\partial b_n} dS + \int_{S} F_{S_i} n_i \frac{\delta (dS)}{\delta b_n} \end{aligned} \tag{2.36}$$

Furthermore, by applying the Leibniz integral rule for the differentiation of volume integrals with moving boundaries, the respective integral in eq. (2.35) assumes the form

$$\frac{\delta}{\delta b_n} \int_{\Omega} F_{\Omega} d\Omega = \int_{\Omega} \frac{\partial F_{\Omega}}{\partial b_n} d\Omega + \int_{S} F_{\Omega} \, n_k \frac{\delta x_k}{\delta b_n} dS \tag{2.37}$$

Given the F is dependent upon the state variables of the primal problem, eq. (2.37) can be expanded as follows

$$\begin{aligned} \frac{\delta}{\delta b_n} \int_{\Omega} F_{\Omega} d\Omega = & \int_{\Omega} \acute{F}_{\Omega}^{v_i} \frac{\partial v_i}{\partial b_n} d\Omega + \int_{\Omega} \acute{F}_{\Omega}^{p} \frac{\partial p}{\partial b_n} d\Omega + \int_{\Omega} \acute{F}_{\Omega}^{\tilde{\nu}} \frac{\partial \tilde{\nu}}{\partial b_n} d\Omega + \int_{S} \acute{F}_{S}^{v_i} \frac{\partial v_i}{\partial b_n} dS + \\ & + \int_{S} \acute{F}_{S}^{p} \frac{\partial p}{\partial b_n} dS + \int_{S} \acute{F}_{S}^{\tilde{\nu}} \frac{\partial \tilde{\nu}}{\partial b_n} dS + \int_{S} F_{\Omega} \, n_k \frac{\delta x_k}{\delta b_n} dS \end{aligned} \tag{2.38}$$

where F_{Ω}^{Φ} includes the partial derivative $\partial F / \partial \Phi$ as well as any term resulting the implementeation of the Green-Gauss theorem for integrals of the form $\int_{\Omega} \frac{\partial}{\partial b_n} \left(\frac{\partial \Phi}{\partial x_j} \right) d\Omega$.

Finally, by substituting eq. (2.36) and (2.38) in eq. (2.35), the final expression for $\delta F/\delta b_n$ arises

$$\begin{aligned}\frac{\delta F}{\delta b_n} &= \int_\Omega \acute{F}_\Omega^{v_i} \frac{\partial v_i}{\partial b_n} d\Omega + \int_\Omega \acute{F}_\Omega^{p} \frac{\partial p}{\partial b_n} d\Omega + \int_\Omega \acute{F}_\Omega^{\tilde{\nu}} \frac{\partial \tilde{\nu}}{\partial b_n} d\Omega + \int_S \left(\acute{F}_S^{v_i} + \frac{\partial F_{S_k}}{\partial v_i} n_k \right) \frac{\partial v_i}{\partial b_n} dS + \\ &+ \int_S \left(\acute{F}_S^{p} + \frac{\partial F_{S_i}}{\partial p} n_i \right) \frac{\partial p}{\partial b_n} dS + \int_S \left(\acute{F}_S^{\tilde{\nu}} + \frac{\partial F_{S_i}}{\partial \tilde{\nu}} n_i \right) \frac{\partial \tilde{\nu}}{\partial b_n} dS + \int_S \frac{\partial F_{S_i}}{\partial \tau_{kj}} \frac{\partial \tau_{kj}}{\partial b_n} n_i dS + \\ &+ \int_S n_i \frac{\partial F_{S_i}}{\partial x_k} \frac{\partial x_k}{\partial b_n} n_k dS + \int_S F_{S_i} \frac{\partial n_i}{\partial b_n} dS + \int_S F_{S_i} n_i \frac{\delta (dS)}{\delta b_n} + \int_S F_\Omega \, n_k \frac{\delta x_k}{\delta b_n} dS \end{aligned} \tag{2.39}$$

The generalized expression for $\delta F/\delta b_n$ presented in eq. (2.39), includes the partial derivatives of the flow variables w.r.t. the design variables. This expression is later specified for the three different objective functions implemented in this thesis. The numerical computation of such derivatives would require the solution of N systems of equations similar to the primal equations. It is, therefore, clearly stated why methods such as the adjoint differentiation are ought to be employed in shape optimization problems, given that they circumvent the costly direct computation of the derivatives in question by solving the adjoint equations.

Force Coefficient

The objective function used to optimize the total force exerted on a certain solid wall boundary of the computational domain can be formulated as a dimensionless coefficient as follows

$$C_F = \frac{\int_{S_w} \rho \left(-\tau_{ij} n_j + p n_i \right) r_i dS}{\frac{1}{2} \rho A_w U_\infty^2} \tag{2.40}$$

where p denotes the mean static pressure, ρ the fluid density, v_i the mean velocity vector components, τ_{ij} the stress tensor components and n_i the outward pointing unit normal vector components. In addition, r_i is the user-defined direction in which the force vector should be projected (e.g. parallel to the farfield velocity to optimize drag), while S_w indicates the aforementioned solid wall boundary and A_w its respective surface area.

Given that only incompressible fluid flows ($\rho = const.$) are simulated in this thesis, eq. (2.40) can be simplified as follows

$$C_F|_{incomp.} = \frac{\int_{S_w} \left(-\tau_{ij} n_j + p n_i \right) r_i dS}{\frac{1}{2} A_w U_\infty^2} \tag{2.41}$$

Differentiation of the incompressible force objective function (2.41) w.r.t. the design variable yields

$$
\begin{aligned}
\left.\frac{\delta C_F}{\delta b_n}\right|_{incomp.} = \frac{2}{A_w U_\infty^2} \Bigg[& \int_{S_w} \frac{\partial}{\partial b_n} \left(-\tau_{ij} n_j + p n_i\right) r_i dS \ + \\
& + \int_{S_w} \left(-\tau_{ij} n_j + p n_i\right) \frac{\partial r_i}{\partial b_n} dS + \int_{S_w} \left(-\tau_{ij} n_j + p n_i\right) r_i \frac{\delta(dS)}{\delta b_n} \Bigg]
\end{aligned}
\tag{2.42}
$$

The force projection vector $\vec{r} \in \mathbb{R}^3$ is constant, causing the second integral in the RHS of eq. (2.42) to be to zero, because $\partial r_i/\partial b_n = 0$, and it assumes the final form

$$
\begin{aligned}
\left.\frac{\delta C_F}{\delta b_n}\right|_{incomp.} = \frac{2}{A_w U_\infty^2} \Bigg[& \int_{S_w} \left(-n_j \frac{\partial \tau_{ij}}{\partial b_n} + -\tau_{ij} \frac{\partial n_j}{\partial b_n} + n_i \frac{\partial p}{\partial b_n} + p \frac{\partial n_i}{\partial b_n}\right) r_i dS \ + \\
& + \int_{S_w} \left(-\tau_{ij} n_j + p n_i\right) r_i \frac{\delta(dS)}{\delta b_n} \Bigg]
\end{aligned}
\tag{2.43}
$$

Total Pressure Losses

The objective function used to minimize the total pressure losses between two boundaries of the computational domain is given by the expression

$$
F_{P_t} = -\int_{S_{I,O}} \left(p + \frac{v_i^2}{2}\right) v_j n_j dS \tag{2.44}
$$

where v_i indicates the mean velocity vector components, whereas the $S_{I,O}$ indicate the inlet and outlet boundaries of the domain, respectively. The units of F_{P_t} as defined in eq. (2.44) are m^5/s^3, thus signifying power losses per units of density.

Differentiation of this objective function w.r.t. the design variable yields

$$
\begin{aligned}
\frac{\delta F_{P_t}}{\delta b_n} = & -\int_{S_{I,O}} \frac{\partial}{\partial b_n} \left(p + \frac{v_i^2}{2}\right) v_j n_j dS - \int_{S_{I,O}} \left(p + \frac{v_i^2}{2}\right) \frac{\partial v_j}{\partial b_n} n_j dS \ - \\
& - \int_{S_{I,O}} \left(p + \frac{v_i^2}{2}\right) v_j \frac{\delta(n_j dS)}{\delta b_n}
\end{aligned}
\tag{2.45}
$$

Given that the boundaries serving as the domain of all surface integrals in eq. (2.45) constitute immovable, non-controlled boundaries during the omptimization process, the third integral of the RHS in the aforementioned eq. is equal to zero, since $\delta(n_j dS)/\delta b_n = 0$. Therefore, the final form for $\delta F_{P_t}/\delta b_n$ arises

$$
\frac{\delta F_{P_t}}{\delta b_n} = -\int_{S_{I,O}} \frac{\partial p}{\partial b_n} v_j n_j dS - \int_{S_{I,O}} \left(v_i v_j + p + \frac{v_i^2}{2}\right) \frac{\partial v_j}{\partial b_n} n_j dS \tag{2.46}
$$

Velocity Angle

The objective function used to optimize the angle of the velocity vector of a certain boundary of a 2D computational domain is defined as

$$\alpha = atan\left(\frac{\int_{S_O} v_2\, dS}{\int_{S_O} v_1\, dS}\right) \tag{2.47}$$

where S_O indicates the immovable outlet boundary of the domain, as used in later simulations. The units of α as defined in eq. (2.47) are *radians*.

Differentiation of this objective function w.r.t. the design variable yields

$$\frac{\delta \alpha}{\delta b_n} = \frac{\int_{S_O} v_1\, dS \cdot \frac{\delta}{\delta b_n}\left(\int_{S_O} v_2\, dS\right) + \int_{S_O} v_2\, dS \cdot \frac{\delta}{\delta b_n}\left(\int_{S_O} v_1\, dS\right)}{\left(\int_{S_O} v_1\, dS\right)^2 + \left(\int_{S_O} v_2\, dS\right)^2} \tag{2.48}$$

The surface integrals domain present in 2.48, is a non-controlled boundary and, therefore, it is not affected by the design variables displacement ($\delta(dS)/\delta b_n = 0$), yielding

$$\frac{\delta}{\delta b_n}\left(\int_{S_O} v_j\, dS\right) = \int_{S_O} \frac{\delta v_j}{\delta b_n}\, dS + \int_{S_O} v_j\, \frac{\delta(dS)}{\delta b_n} = \int_{S_O} \frac{\delta v_j}{\delta b_n}\, dS\ ,\quad j = 1, 2 \tag{2.49}$$

According to (2.49), eq. (2.48) becomes

$$\frac{\delta \alpha}{\delta b_n} = \frac{\int_{S_O} v_1\, dS \cdot \int_{S_O} \frac{\delta v_2}{\delta b_n}\, dS + \int_{S_O} v_2\, dS \cdot \int_{S_O} \frac{\delta v_1}{\delta b_n}\, dS}{\left(\int_{S_O} v_1\, dS\right)^2 + \left(\int_{S_O} v_2\, dS\right)^2} \tag{2.50}$$

2.2.3 Differentiation of the Primal Equations

Now that the expression for the derivatives of the objective function is defined, the partial derivatives of the primal equation w.r.t. the design variables have to be formulated, as dictated by eq. (2.20).

The differentiation of the continuity eq. (2.14a) yields

$$\frac{\partial R^p}{\partial b_n} = -\frac{\partial}{\partial x_j}\left(\frac{\partial u_j}{\partial b_n}\right) \tag{2.51}$$

while the respective partial derivative of the momentum eq. (2.14b) assumes the form

$$\begin{aligned}\frac{\partial R_i^v}{\partial b_n} = \frac{\partial v_j}{\partial b_n}\frac{\partial v_i}{\partial x_j} + v_j\frac{\partial}{\partial x_j}\left(\frac{\partial v_i}{\partial b_n}\right) - \frac{\partial}{\partial x_j}\left[(\nu+\nu_t)\frac{\partial}{\partial b_n}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)\right] - \\ -\frac{\partial}{\partial x_j}\left[\frac{\partial \nu_t}{\partial b_n}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)\right] + \frac{\partial}{\partial x_i}\left(\frac{\partial p}{\partial b_n}\right) \;, \quad i=1,2,3\end{aligned} \tag{2.52}$$

where $\partial \nu_t/\partial b_n$ can be computed as follows

$$\frac{\partial \nu_t}{\partial b_n} = \frac{\partial \nu_t}{\partial \widetilde{\nu}}\frac{\partial \widetilde{\nu}}{\partial b_n} \quad with \quad \frac{\partial \nu_t}{\partial \widetilde{\nu}} = f_{v_1} + \widetilde{\nu}\frac{\partial f_{v_1}}{\partial \widetilde{\nu}} = f_{v_1} + \frac{3c_{v_1}^3\chi^3}{\left(\chi^3+c_{v_1}^3\right)^2} \tag{2.53}$$

Moreover, the partial derivative of the Spalart–Allmaras turbulence model eq. (2.14c) is formulated as

$$\begin{aligned}\frac{\delta R^{\widetilde{\nu}}}{\delta b_n} = \frac{\partial \widetilde{\nu}}{\partial x_j}\frac{\partial v_j}{\partial b_n} + v_j\frac{\partial}{\partial x_j}\left(\frac{\partial \widetilde{\nu}}{\partial b_n}\right) - \frac{\partial}{\partial x_j}\left[\left(\nu+\frac{\widetilde{\nu}}{\sigma}\right)\frac{\partial}{\partial x_j}\left(\frac{\partial \widetilde{\nu}}{\partial b_n}\right)\right] \\ -\frac{1}{\sigma}\frac{\partial}{\partial x_j}\left(\frac{\partial \widetilde{\nu}}{\partial b_n}\frac{\partial \widetilde{\nu}}{\partial x_j}\right) - 2\frac{c_{b2}}{\sigma}\frac{\partial \widetilde{\nu}}{\partial x_j}\frac{\partial}{\partial x_j}\left(\frac{\partial \widetilde{\nu}}{\partial b_n}\right) \\ +\widetilde{\nu}\left(-\frac{\partial P}{\partial b_n}+\frac{\partial D}{\partial b_n}\right) + (-P+D)\frac{\partial \widetilde{\nu}}{\partial b_n}\end{aligned} \tag{2.54}$$

The differentiation of the production and dissipation terms presented in eq. (2.8), yields

$$-\frac{\partial P}{\partial b_n} + \frac{\partial D}{\partial b_n} = \mathcal{C}_{\widetilde{\nu}}\frac{\partial \widetilde{\nu}}{\partial b_n} + \mathcal{C}_{\Delta}\frac{\partial \Delta}{\partial b_n} + \mathcal{C}_Y\frac{1}{Y}e_{mjk}\frac{\partial v_k}{\partial x_j}e_{mli}\frac{\partial}{\partial b_n}\left(\frac{\partial v_i}{\partial x_l}\right) \tag{2.55}$$

where

$$\mathcal{C}_Y = \left(-c_{b_1} - c_{w_1}\mathcal{C}\frac{r}{\widetilde{Y}}\right) f_{v_3} \tag{2.56}$$

$$\mathcal{C}_\Delta = -\frac{2}{\Delta^3}\left[c_{w_1} r\, \mathcal{C}\left(\Delta^2 - \frac{\widetilde{\nu} f_{v_2}}{\kappa^2 \widetilde{Y}}\right) + c_{w_1} f_w \widetilde{\nu} - c_{b_1}\frac{f_{v_2}}{\kappa^2}\widetilde{\nu}\right] \tag{2.57}$$

$$\mathcal{C}_{\widetilde{\nu}} = \left(-c_{b_1} - c_{w_1}\mathcal{C}\frac{r}{\widetilde{Y}}\right)\left(\frac{\partial f_{v_3}}{\partial \widetilde{\nu}} Y + \frac{f_{v_2}}{\kappa^2\Delta^2} + \frac{\partial f_{v_2}}{\partial \widetilde{\nu}}\frac{\widetilde{\nu}}{\kappa^2\Delta^2}\right) + c_{w_1}\mathcal{C}\frac{r}{\widetilde{\nu}} + c_{w_1}\frac{f_w}{\Delta^2} \tag{2.58}$$

$$C = \frac{c_{w_1}\widetilde{\nu}^2}{\Delta^2}\left[1 + c_{w_2}(6r^5 - 1)\right]\frac{c_{w_3}^6}{g^6 + c_{w_3}^6}\left(\frac{1 + c_{w_3}^6}{g^6 + c_{w_3}^6}\right)^{1/6} \tag{2.59}$$

$$\frac{\partial f_{v_2}}{\partial \widetilde{\nu}} = -\frac{3}{\nu c_{v_2}}\left(1 + \frac{\chi}{c_{v_2}}\right)^{-4} \tag{2.60}$$

$$\begin{aligned}\frac{\partial f_{v_3}}{\partial \widetilde{\nu}} = &\frac{1}{c_{v_2}}\left(\frac{f_{v_1}}{\nu} + \chi\frac{\partial f_{v_1}}{\partial \nu}\right)\left[3\left(1 + \frac{\chi}{c_{v_2}}\right) + \left(\frac{\chi}{c_{v_2}}\right)^2\right]\left(1 + \frac{\chi}{c_{v_2}}\right)^{-3}\\ &+ \frac{1}{\nu c_{v_2}^2}(1 + \chi f_{v_1})\left(3 + 2\frac{\chi}{c_{v_2}}\right)c_{v_2}^2\left(1 + \frac{\chi}{c_{v_2}}\right)^{-3}\\ &- 3\frac{(1 + \chi f_{v_1})}{\nu c_{v_2}^2}\left[3\left(1 + \frac{\chi}{c_{v_2}}\right) + \left(\frac{\chi}{c_{v_2}}\right)^2\right]\left(1 + \frac{\chi}{c_{v_2}}\right)^{-4}\end{aligned} \tag{2.61}$$

A more thorough analysis of the continuous adjoint equation to the Spalart-Allmaras model can be found in [19], [24].

2.2.4 Field Adjoint Equations and Adjoint Boundary Conditions

By substituting eqs. 2.51, 2.52, 2.54 and (2.39) into eq. 2.29 we receive the final expression of the material derivative of the augmented objective function w.r.t. the design variables.

$$\begin{aligned}\frac{\delta F_{aug}}{\delta b_n} = &\int_S \mathcal{BC}_i^u \frac{\partial v_i}{\partial b_n} dS + \int_S \mathcal{BC}^p \frac{\partial p}{\partial b_n} dS + \int_S \mathcal{BC}^{\widetilde{\nu_a}} \frac{\partial \widetilde{\nu}}{\partial b_n} dS + \int_S \mathcal{BC}^{m_a} \frac{\partial}{\partial x_j}\left(\frac{\delta x_i}{\delta b_n}\right) dS\\ &+ \int_S (-u_i n_j + \frac{\partial F_{S_k}}{\partial \tau_{ij}} n_k)\frac{\partial \tau_{ij}}{\partial b_n} dS - \int_S \widetilde{\nu_a}\left(\nu + \frac{\widetilde{\nu}}{\sigma}\right)\frac{\partial}{\partial b_n}\left(\frac{\partial \widetilde{\nu}}{\partial x_j}\right) n_j dS\\ &+ \int_\Omega R_i^u \frac{\partial v_i}{\partial b_n} d\Omega + \int_\Omega R^q \frac{\partial p}{\partial b_n} d\Omega + \int_\Omega R^{\widetilde{\nu_a}} \frac{\partial \widetilde{\nu}}{\partial b_n} d\Omega + \int_\Omega R_k^{m^a} \frac{\delta x_k}{\delta b_n} d\Omega\\ &+ \int_{S_{W_p}} n_i \frac{\partial F_{S_{W_p,i}}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k dS + \int_{S_{W_p}} F_{S_{W_p,i}} \frac{\delta n_i}{\delta b_n} dS + \int_{S_{W_p}} F_{S_{W_p,i}} n_i \frac{\delta(dS)}{\delta b_n}\\ &- \int_{S_{W_P}} \frac{\partial m_i^a}{\partial x_j} n_j \frac{\delta x_i}{\delta b_n} dS + \int_\Omega \widetilde{\nu}\widetilde{\nu}_a \mathcal{C}_\Delta \frac{\partial \Delta}{\partial b_n} d\Omega + \int_S m_i^a R_i^m n_k \frac{\delta x_k}{\delta b_n} dS\end{aligned} \tag{2.62}$$

where

$$\begin{aligned}
\mathcal{BC}^u_i = & \, u_i v_j n_j + (\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) n_j - q n_i + \widetilde{\nu_a}\widetilde{\nu}\frac{\mathcal{C}_Y}{Y} e_{mjk}\frac{\partial v_k}{\partial x_j} e_{mli} n_l \\
& + \frac{\partial F_{S_k}}{\partial v_i} n_k + \acute{F}^v_{S,i} \qquad (2.63)
\end{aligned}$$

$$\mathcal{BC}^p = u_j n_j + \frac{\partial F_{S_i}}{\partial p} n_i + \acute{F}^p_S \qquad (2.64)$$

$$\mathcal{BC}^{\widetilde{\nu_a}} = \widetilde{\nu_a} v_j n_j + \left(\nu + \frac{\widetilde{\nu}}{\sigma}\right)\frac{\partial \widetilde{\nu_a}}{\partial x_j} n_j - \frac{\widetilde{\nu_a}}{\sigma}\left(1 + 2c_{b_2}\right)\frac{\partial \widetilde{\nu}}{\partial x_j} n_j + \frac{\partial F_{S_k}}{\partial \widetilde{\nu}} n_k + \acute{F}^{\widetilde{\nu}}_S \qquad (2.65)$$

$$\mathcal{BC}^{m_a} = m^a_i n_j \qquad (2.66)$$

After setting the multipliers of $\partial v_i/\partial b_n$, $\partial p/\partial b_n$, $\partial\widetilde{\nu}/\partial b_n$ and $\delta x_k/\delta b_n$, in the volume integrals of eq. 2.62 to zero, the field adjoint equations are derived.

$$R^q = -\frac{\partial u_j}{\partial x_j} + \acute{F}^p_\Omega = 0 \tag{2.67}$$

$$\begin{aligned} R^u_i = & u_j \frac{\partial v_j}{\partial x_i} - \frac{\partial (v_j u_i)}{\partial x_j} - \frac{\partial}{\partial x_j} \left[(\nu + \nu_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] + \frac{\partial q}{\partial x_i} + \acute{F}^v_{\Omega,i} \\ & + \widetilde{\nu_a} \frac{\partial \widetilde{\nu}}{\partial x_i} - \frac{\partial}{\partial x_l} \left(\widetilde{\nu_a} \widetilde{\nu} \frac{\mathcal{C}_Y}{Y} e_{mjk} \frac{\partial v_k}{\partial x_j} e_{mli} \right) = 0 \ , \ i = 1,2,3 \end{aligned} \tag{2.68}$$

$$\begin{aligned} R^{\widetilde{\nu_a}} = & -\frac{\partial (v_j \widetilde{\nu_a})}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\widetilde{\nu}}{\sigma} \right) \frac{\partial \widetilde{\nu_a}}{\partial x_j} \right] + \frac{1}{\sigma} \frac{\partial \widetilde{\nu_a}}{\partial x_j} \frac{\partial \widetilde{\nu}}{\partial x_j} + 2 \frac{c_{b2}}{\sigma} \frac{\partial}{\partial x_j} \left(\widetilde{\nu_a} \frac{\partial \widetilde{\nu}}{\partial x_j} \right) \\ & + \widetilde{\nu_a} \widetilde{\nu} \mathcal{C}_{\widetilde{\nu}} + \frac{\partial \nu_t}{\partial \widetilde{\nu}} \frac{\partial u_i}{\partial x_j} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + (-P + D) \, \widetilde{\nu_a} + \acute{F}^{\widetilde{\nu}}_\Omega = 0 \end{aligned} \tag{2.69}$$

$$R^{m^a}_k = \frac{\partial^2 m^a_k}{\partial x_j^2} + \frac{\partial}{\partial x_j} \left\{ u_i v_j \frac{\partial v_i}{\partial x_k} + u_j \frac{\partial p}{\partial x_k} + \tau^a_{ij} \frac{\partial u_i}{\partial x_k} - u_i \frac{\partial \tau_{ij}}{\partial x_k} - q \frac{\partial v_j}{\partial x_k} \right\} = 0 \tag{2.70}$$

After satisfying the field adjoint equations, the remaining terms in eq. 2.62 are

$$\begin{aligned} \frac{\delta F_{aug}}{\delta b_n} = & \int_S \mathcal{BC}^u_i \frac{\partial v_i}{\partial b_n} dS + \int_S \mathcal{BC}^p \frac{\partial p}{\partial b_n} dS + \int_S \mathcal{BC}^{\widetilde{\nu_a}} \frac{\partial \widetilde{\nu}}{\partial b_n} dS + \int_S \mathcal{BC}^{m_a} \frac{\partial}{\partial x_j} \left(\frac{\delta x_i}{\delta b_n} \right) dS \\ & + \int_S (-u_i n_j + \frac{\partial F_{S_k}}{\partial \tau_{ij}} n_k) \frac{\partial \tau_{ij}}{\partial b_n} dS - \int_S \widetilde{\nu_a} \left(\nu + \frac{\widetilde{\nu}}{\sigma} \right) \frac{\partial}{\partial b_n} \left(\frac{\partial \widetilde{\nu}}{\partial x_j} \right) n_j dS \\ & + \int_{S_{W_p}} n_i \frac{\partial F_{S_{W_p,i}}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k dS + \int_{S_{W_p}} F_{S_{W_p,i}} \frac{\delta n_i}{\delta b_n} dS + \int_{S_{W_p}} F_{S_{W_p,i}} n_i \frac{\delta (dS)}{\delta b_n} \\ & - \int_{S_{W_P}} \frac{\partial m^a_i}{\partial x_j} n_j \frac{\delta x_i}{\delta b_n} dS + \int_\Omega \widetilde{\nu} \widetilde{\nu_a} \mathcal{C}_\Delta \frac{\partial \Delta}{\partial b_n} d\Omega + \int_S m^a_i R^m_i n_k \frac{\delta x_k}{\delta b_n} dS \end{aligned} \tag{2.71}$$

The system of the field adjoint PDEs is closed with the adjoint boundary conditions. The ABC are imposed aiming to eliminate the surface integrals that contain the partial derivatives of the state variables w.r.t. the design variables, namely the first six and the last integral of eq. 2.71 which contain the surface integrals of $\partial v_i/\partial b_n$, $\partial p/\partial b_n$, $\partial v_i/\partial b_n$, $\partial \widetilde{\nu}/\partial b_n$,$\partial(\delta x_i/\delta b_n)/\partial x_j$, $\partial \tau_{ij}/\partial b_n$, $\partial(\partial \widetilde{\nu}/\partial x_j)/\partial b_n$ and $\delta x_i/\delta b_n$. For the sake of completeness these terms are rewritten as follows.

$$I_1 = \int_S \mathcal{BC}_i^u \frac{\partial v_i}{\partial b_n} dS \tag{2.72}$$

$$I_2 = \int_S \mathcal{BC}^p \frac{\partial p}{\partial b_n} dS \tag{2.73}$$

$$I_3 = \int_S \mathcal{BC}^{\widetilde{\nu_a}} \frac{\partial \widetilde{\nu}}{\partial b_n} dS \tag{2.74}$$

$$I_4 = \int_S \mathcal{BC}^{m_a} \frac{\partial}{\partial x_j} \left(\frac{\delta x_i}{\delta b_n} \right) dS \tag{2.75}$$

$$I_5 = \int_S (-u_i n_j + \frac{\partial F_{S_k}}{\partial \tau_{ij}} n_k) \frac{\partial \tau_{ij}}{\partial b_n} dS \tag{2.76}$$

$$I_6 = \int_S \widetilde{\nu_a} \left(\nu + \frac{\widetilde{\nu}}{\sigma} \right) \frac{\partial}{\partial b_n} \left(\frac{\partial \widetilde{\nu}}{\partial x_j} \right) n_j dS \tag{2.77}$$

$$I_7 = \int_S m_i^a R_i^m n_k \frac{\delta x_k}{\delta b_n} dS \tag{2.78}$$

At this point, the final expressions of the adjoint boundary conditions are presented, whereas their detailed derivation can be found in [19]. The adjoint boundary condition of the adjoint gdPDEs is the same for all boundaries, namely $m_k^a = 0$, so that integral I_4 is eliminated. Also, since $m_k^a = 0$ along all boundaries, integral I_7, which is the equivalent of LBterm discussed in 2.2.1, also vanishes in all boundaries.

Inlet Boundaries S_I

At the inlet boundaries since Dirichlet boundary conditions are imposed on v_i and $\widetilde{\nu}$, $\delta v_i/\delta b_n = 0$ and $\delta\widetilde{\nu}/\delta b_n = 0$. Since S_I in a non-controlled boundary, $\delta x_k/\delta b_n = 0$ and taking into consideration eq. 2.31, $\partial v_i/\partial b_n = 0$ and $\partial\widetilde{\nu}/\partial b_n = 0$. This means that $I_1 = I_3 = 0$.

Integrals I_2 and I_5 are eliminated by demanding

$$u_{\langle n \rangle} = -\frac{\partial F_{S_I,j}}{\partial p} n_j \tag{2.79a}$$

$$u_{\langle t \rangle}^I = \frac{\partial F_{S_I,k}}{\partial \tau_{ij}} n_k t_i^I n_j + \frac{\partial F_{S_I,k}}{\partial \tau_{ij}} n_k t_j^I n_i \tag{2.79b}$$

$$u_{\langle t \rangle}^{II} = \frac{\partial F_{S_I,k}}{\partial \tau_{ij}} n_k t_i^{II} n_j + \frac{\partial F_{S_I,k}}{\partial \tau_{ij}} n_k t_j^{II} n_i \tag{2.79c}$$

where t_i^I, t_i^{II} are the components of the tangent to the surface unit vectors. The first tangent vector t_i^I can be defined as arbitrary unit vector parallel to S_I, whereas t_i^{II} forms an orthogonal system with n and t_i^I. Quantities $u_{\langle t \rangle}^I$ and $u_{\langle t \rangle}^{II}$ are the components of the adjoint velocity in the t_i^I, t_i^{II} directions respectively. It should be noted that if F is not defined at the inlet of the computational domain, the adjoint velocity components are zero along S_I. Integral I_6 is zeroed by imposing a zero Dirichlet condition to $\widetilde{\nu_a}$, i.e. $\widetilde{\nu_a} = 0$.

Finally, since no boundary condition for q results from the elimination of any of the seven boundary integrals already discussed, a zero Neumann boundary condition is employed.

Outlet Boundaries S_O

At the outlet boundaries since a Dirichlet boundary condition is imposed on p, $\delta p/\delta b_n = 0$. Since S_O is fixed, $\delta x_k/\delta b_n = 0$ and taking into consideration eq. 2.31, $\partial p/\partial b_n = 0$. As a result, integral I_2 vanishes automatically. Due to the distance of the outlet boundary from the controlled area, an almost uniform velocity profile can be assumed along S_O, meaning that $\delta \tau_{ij}/\delta b_n = 0$ along S_O. Hence, integral I_5 can be neglected.

In order to eliminate I_1, its integrand quantity is set equal to zero, i.e.

$$\begin{aligned}\mathcal{BC}_i^u = & u_i v_j n_j + (\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) n_j - q n_i + \widetilde{\nu_a}\widetilde{\nu}\frac{\mathcal{C}_Y}{Y} e_{mjk}\frac{\partial v_k}{\partial x_j} e_{mli} n_l \\ & + \frac{\partial F_{S_k}}{\partial v_i} n_k + \acute{F}_{S,i}^{v} = 0\end{aligned} \tag{2.80}$$

Eq. 2.80, which can be analysed in three scalar equations, $i = 1, 2, 3$, includes four unknown quantities (the adjoint pressure q and the three components of the adjoint velocity u_i). Therefore, one of them may take on an arbitrary value. This is chosen to be the normal component of the adjoint velocity $u_{\langle n \rangle}$, on which a zero Neumann boundary condition is imposed. By multiplying equation 2.80 with n_i a Dirichlet condition for the adjoint pressure is derived

$$\begin{aligned} q = & u_{\langle n \rangle} v_{\langle n \rangle} + 2(\nu + \nu_t)\frac{\partial u_{\langle n \rangle}}{\partial n} + \frac{\partial F_{S_{O,k}}}{\partial v_i} n_i n_k + \acute{F}_{S_{O,i}}^{v} n_i \\ & + \widetilde{\nu_a}\widetilde{\nu}\frac{\mathcal{C}_Y}{Y} e_{mjk}\frac{\partial v_k}{\partial x_j} e_{mli} n_l n_i = 0\end{aligned} \tag{2.81}$$

The outlet adjoint tangential velocity is computed by multiplying eq. 2.80 with the tangent to the surface vectors t_i^l , $l = 1, 2$.

$$\begin{aligned} 0 = & v_{\langle t \rangle} u_{\langle t \rangle}^l + (\nu + \nu_t)\left(\frac{\partial u_{\langle t \rangle}^l}{\partial n} + \frac{\partial u_{\langle n \rangle}}{\partial t^l}\right) + \frac{\partial F_{S_{O,k}}}{\partial v_i} n_k t_i^l + \acute{F}_{S_{O,i}}^{v} t_i^l \\ & - \widetilde{\nu_a}\widetilde{\nu}\frac{\mathcal{C}_Y}{Y} e_{mjk}\frac{\partial v_k}{\partial x_j} e_{mzi} n_z t_i^l \ , \ l = 1, 2\end{aligned} \tag{2.82}$$

Finally, a Robin-type boundary condition is imposed on $\widetilde{\nu_a}$ in order to eliminate integral I_3.

$$BC^{\widetilde{\nu_a}} = \widetilde{\nu_a} v_j n_j + \left(\nu + \frac{\widetilde{\nu}}{\sigma}\right) \frac{\partial \widetilde{\nu_a}}{\partial x_j} n_j + \frac{\partial F_{S_{O,k}}}{\partial \widetilde{\nu}} n_k + \acute{F}^{\widetilde{\nu}}_{S_O} = 0 \tag{2.83}$$

It must be noted that term $\frac{\widetilde{\nu_a}}{\sigma}\left(1 + 2c_{b_2}\right) \frac{\partial \widetilde{\nu}}{\partial x_j} n_j$ has been eliminated from eq. 2.83 with regards to eq. 2.65 where $BC^{\widetilde{\nu_a}}$ was originally defined, because of the zero Neumann boundary condition imposed on $\widetilde{\nu}$ for the outlet boundaries.

Unparameterized Wall Boundaries S_W

Since $\widetilde{\nu}$ is equal to zero on the wall boundaries, integral I_3 vanishes. However, this is not the case for the gradient of $\widetilde{\nu}$ and in order to eliminate integral I_6 a zero Dirichlet boundary condition imposed on $\widetilde{\nu_a}$. The boundary conditions imposed on the adjoint velocity conditions are derived following the same procedure presented for the inlet boundaries. For the sake of completeness these boundary conditions are

$$u_{\langle n \rangle} = -\frac{\partial F_{S_W,j}}{\partial p} n_j \tag{2.84a}$$

$$u^{I}_{\langle t \rangle} = \frac{\partial F_{S_W,k}}{\partial \tau_{ij}} n_k t^{I}_i n_j + \frac{\partial F_{S_W,k}}{\partial \tau_{ij}} n_k t^{I}_j n_i \tag{2.84b}$$

$$u^{II}_{\langle t \rangle} = \frac{\partial F_{S_W,k}}{\partial \tau_{ij}} n_k t^{II}_i n_j + \frac{\partial F_{S_W,k}}{\partial \tau_{ij}} n_k t^{II}_j n_i \tag{2.84c}$$

Finally, a zero Neumann boundary condition is imposed on q.

Parameterized Wall Boundaries S_{W_P}

The main difference between parameterized and non-parameterized wall boundaries is the fact that the parameterized boundaries may change during the optimization. Thus, $\delta x_k/\delta b_n \neq 0$ and the total and partial derivatives of the flow quantities are different and are linked through eq. 2.31. In addition, the total variations in the normal and tangent surface vectors are not zero, contributing extra terms during the formulation of the adjoint boundary conditions [19].

2.2.5 Adjoint to the Distance Equation

After satisfying the field adjoint equations along with their adjoint boundary conditions, eq. 2.71 takes the form of eq. 2.85. To this equation are included some extra terms that arise from the derivation of the adjoint boundary conditions at the controlled boundaries [19, 25].

$$
\begin{aligned}
\frac{\delta F_{aug}}{\delta b_n} = {} & T_{SD}^{WF} - \int_{S_{W_p}} \mathcal{SD}_1 \frac{\partial \tau_{ij}}{\partial x_m} n_j t_i^I n_m n_k \frac{\delta x_k}{\delta b_n} dS - \int_{S_{W_p}} \mathcal{SD}_1 \tau_{ij} \frac{\delta (n_j t_i^I)}{\delta b_n} \frac{\delta x_k}{\delta b_n} dS \\
& + \int_{S_{W_p}} \mathcal{SD}_{2,i} v_{\langle t \rangle}^I \frac{\delta t_i^I}{\delta b_n} dS - \int_{S_{W_p}} \mathcal{SD}_{2,i} \frac{\partial v_i}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS \\
& - \int_{S_{W_p}} \left[\left(\nu + \frac{\widetilde{\nu}}{\sigma} \right) \frac{\partial \widetilde{\nu}_a}{\partial x_j} n_j + \frac{\partial F_{S_z}}{\partial \widetilde{\nu}} n_z + \acute{F}_S^{\widetilde{\nu}} \right] \frac{\partial \widetilde{\nu}}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS \\
& - \int_{S_{W_p}} (-u_{\langle n \rangle} + \phi_{\langle n \rangle \langle n \rangle}) \left(\tau_{ij} \frac{\delta (n_i n_j)}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k n_i n_j \right) dS \\
& - \int_{S_{W_p}} \phi_{\langle t^I \rangle \langle t^I \rangle} \left(\tau_{ij} \frac{\delta (t_i^I t_j^I)}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k t_i^I t_j^I \right) dS \\
& - \int_{S_{W_p}} (\phi_{\langle t^{II} \rangle \langle t^I \rangle} + \phi_{\langle t^I \rangle \langle t^{II} \rangle}) \left(\tau_{ij} \frac{\delta (t_i^{II} t_j^I)}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k t_i^{II} t_j^I \right) dS \\
& - \int_{S_{W_p}} \phi_{\langle t^{II} \rangle \langle t^{II} \rangle} \left(\tau_{ij} \frac{\delta (t_i^{II} t_j^{II})}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k t_i^{II} t_j^{II} \right) dS \\
& + \int_{S_{W_p}} n_i \frac{\partial F_{S_{W_p,i}}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k dS + \int_{S_{W_p}} F_{S_{W_p,i}} \frac{\delta n_i}{\delta b_n} dS + \int_{S_{W_p}} F_{S_{W_p,i}} n_i \frac{\delta (dS)}{\delta b_n} \\
& - \int_{S_{W_P}} \frac{\partial m_i^a}{\partial x_j} n_j \frac{\delta x_i}{\delta b_n} dS + \int_{S_{W_p}} \mathcal{A}_{\Delta}^{WF} \frac{\partial \Delta^P}{\partial b_n} dS + \int_{S_W} \mathcal{A}_{\Delta}^{WF} \frac{\partial \Delta^P}{\partial b_n} dS \\
& + \int_{\Omega} \widetilde{\nu} \widetilde{\nu}_a \mathcal{C}_{\Delta} \frac{\partial \Delta}{\partial b_n} d\Omega
\end{aligned}
\tag{2.85}
$$

where

$$\mathcal{SD}_1 = -u^I_{\langle t\rangle} + \phi_{\langle t^I\rangle\langle n\rangle} + \phi_{\langle n\rangle\langle t^I\rangle} \tag{2.86}$$

$$\mathcal{SD}_{2,i} = (\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) n_j - q n_i + \frac{\partial F_{S_{W_p,k}}}{\partial v_i} n_k + \acute{F}^v_{S_{W_p,i}} \tag{2.87}$$

$$\phi_{ij} = \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{ij}} n_k \tag{2.88}$$

T^{WF}_{SD}, $\int_{S_{W_P}} \mathcal{A}^{WF}_{\Delta} \frac{\partial \Delta^P}{\partial b_n} dS$ and $\int_{S_W} \mathcal{A}^{WF}_{\Delta} \frac{\partial \Delta^P}{\partial b_n} dS$ summarize the contribution of the wall functions differentiation to the sensitivity derivatives [26]. As can be seen, all but the last term of eq. 2.85 are surface integrals, which can be computed at a cost that is, practically, negligible when compared to the solution of the primal or the adjoint equations. However, this is not the case for the last field integral which contains the distance variation for the entire domain w.r.t. the design variables. The simplest way to compute this variation is through finite differences, i.e. by perturbating each of the design variables by an infinitesimally small quantity ϵ in the positive and negative directions and re-computing nodal distances for the entire domain. Then, the total distance variation would be

$$\frac{\delta \Delta}{\delta b_n} = \frac{\Delta(b_n + \epsilon) - \Delta(b_n - \epsilon)}{2\epsilon} \tag{2.89}$$

Having computed the total distance variation, the partial variation of Δ appearing in the last field integral of 2.85 can be calculated through eq. 2.30 as follows

$$\frac{\partial \Delta}{\partial b_n} = \frac{\delta \Delta}{\delta b_n} - \frac{\partial \Delta}{\partial x_k} \frac{\delta x_k}{\delta b_n}$$

Nevertheless, the finite differences method has the same issues as the ones described in the introduction of this diploma thesis, namely the requirement to make $2N$ computations of the distance field (for instance by an exhaustive search of all cell centers with all boundary faces) and the sensitivity of the result from the value of ϵ.

An alternative and more cost-effective way to deal with $\partial \Delta / \partial b_n$ is to apply the adjoint methodology in order to eliminate the term containing this variation. There are various PDEs that can be used to compute the distances field Δ. Hamilton-Jacobi equation has shown to produce a very good approximation to the Euclidean distance field and to be numerically robust [19, 27]. Hamilton-Jacobi equation reads

$$R^{\Delta} = \frac{\partial (c_j \Delta)}{\partial x_j} - \Delta \frac{\partial^2 \Delta}{\partial x_j^2} - 1 = 0 \tag{2.90}$$

where $c_j = \partial \Delta / \partial x_j$. The boundary conditions of eq. 2.90 consist of a zero Dirichlet condition for the solid wall boundaries and $\frac{\partial \Delta}{\partial x_i} n_i = 1$ for the rest of the domain boundaries. This equation can be viewed as an additional primal PDE to be solved, meaning that it should be added to the augmented objective function, eq. 2.26 which now becomes

$$F_{aug} = \underbrace{F + \int_{\Omega} u_i R_i^v d\Omega + \int_{\Omega} q R^p d\Omega + \int_{\Omega} \widetilde{\nu}_a R^{\widetilde{\nu}} d\Omega + \int_{\Omega} m_i^a R_i^m d\Omega}_{T_1} + \underbrace{\int_{\Omega} \Delta_{\alpha} R^{\Delta} d\Omega}_{T_2} \tag{2.91}$$

where Δ_{α} is the adjoint to the distance field variable. The differentiation of F_{aug} follows the same methodology presented in section 2.2.1. We have

$$\frac{\delta F_{aug}}{\delta b_n} = \frac{\delta T_1}{\delta b_n} + \frac{\delta T_2}{\delta b_n} \tag{2.92}$$

The development of $\delta T_1/\delta b_n$ led to eq. 2.85. $\delta T_2/\delta b_n$ is developed using the Leibniz theorem, as follows

$$\frac{\delta T_2}{\delta b_n} = \frac{\delta}{\delta b_n} \int_{\Omega} \Delta_a R^{\Delta} d\Omega = \int_{\Omega} \Delta_a \frac{\partial R^{\Delta}}{\partial b_n} d\Omega + \int_{S_{W_p}} \Delta_a R^{\Delta} n_k \frac{\delta x_k}{\delta b_n} dS \tag{2.93}$$

After differentiating the Hamilton-Jacobi equation and substituting the result in eq. 2.93, we receive

$$\frac{\delta T_2}{\delta b_n} = \int_S 2\Delta_a \frac{\partial \Delta}{\partial x_j} n_j \frac{\partial \Delta}{\partial b_n} dS + \int_{S_{W_p}} \Delta_a R^{\Delta} n_k \frac{\delta x_k}{\delta b_n} dS - \int_{\Omega} 2 \frac{\partial}{\partial x_j} \left(\Delta_a \frac{\partial \Delta}{\partial x_j} \right) \frac{\partial \Delta}{\partial b_n} d\Omega \tag{2.94}$$

By integrating eq. 2.94 into 2.85, the expression where the multiplier of $\partial\Delta/\partial b_n$ in the resulting volume integrals should be set to zero, is derived. Thus, the adjoint to the distance field equation is derived

$$R^{\Delta_\alpha} = -2 \frac{\partial}{\partial x_j} \left(\Delta_\alpha \frac{\partial \Delta}{\partial x_j} \right) + \widetilde{\nu}\widetilde{\nu}_a C_{\Delta} = 0 \tag{2.95}$$

where the first of the terms in the RHS of eq. 2.95 is contributed by the differentiation of the Hamilton-Jacobi equation 2.90, whereas the second one from the differentiation of the Spalart-Allmaras equation.

Having satisfied the field adjoint distance equation along with the proper boundary condition [25], the terms that should be added to the sensitivity derivatives expression replacing the last field integral $\int_{\Omega} \widetilde{\nu}\widetilde{\nu}_a \mathcal{C}_{\Delta} \frac{\partial \Delta}{\partial b_n} d\Omega$ of equation 2.85 are

$$\int_{\Omega} \widetilde{\nu}\widetilde{\nu}_a \mathcal{C}_{\Delta} \frac{\partial \Delta}{\partial b_n} d\Omega = \int_{S_{W_p}} \Delta_a R^{\Delta} n_k \frac{\delta x_k}{\delta b_n} dS - \int_{S_{W_p}} 2\Delta_a \frac{\partial \Delta}{\partial x_j} n_j \frac{\partial \Delta}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS \tag{2.96}$$

2.2.6 Final Expression of the Sensitivity Derivatives

Taking everything into consideration, the final expression for the sensitivity derivatives reads

$$
\begin{aligned}
\frac{\delta F_{aug}}{\delta b_n} = & T_{SD}^{WF} - \int_{S_{W_p}} \mathcal{SD}_1 \frac{\partial \tau_{ij}}{\partial x_m} n_j t_i^I n_m n_k \frac{\delta x_k}{\delta b_n} dS - \int_{S_{W_p}} \mathcal{SD}_1 \tau_{ij} \frac{\delta (n_j t_i^I)}{\delta b_n} \frac{\delta x_k}{\delta b_n} dS \\
& + \int_{S_{W_p}} \mathcal{SD}_{2,i} v_{\langle t \rangle}^I \frac{\delta t_i^I}{\delta b_n} dS - \int_{S_{W_p}} \mathcal{SD}_{2,i} \frac{\partial v_i}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS \\
& - \int_{S_{W_p}} \left[\left(\nu + \frac{\widetilde{\nu}}{\sigma} \right) \frac{\partial \widetilde{\nu}_a}{\partial x_j} n_j + \frac{\partial F_{S_z}}{\partial \widetilde{\nu}} n_z + \acute{F}_S^{\widetilde{\nu}} \right] \frac{\partial \widetilde{\nu}}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS \\
& - \int_{S_{W_p}} (-u_{\langle n \rangle} + \phi_{\langle n \rangle \langle n \rangle}) \left(\tau_{ij} \frac{\delta (n_i n_j)}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k n_i n_j \right) dS \\
& - \int_{S_{W_p}} \phi_{\langle t^I \rangle \langle t^I \rangle} \left(\tau_{ij} \frac{\delta (t_i^I t_j^I)}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k t_i^I t_j^I \right) dS \\
& - \int_{S_{W_p}} (\phi_{\langle t^{II} \rangle \langle t^I \rangle} + \phi_{\langle t^I \rangle \langle t^{II} \rangle}) \left(\tau_{ij} \frac{\delta (t_i^{II} t_j^I)}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k t_i^{II} t_j^I \right) dS \\
& - \int_{S_{W_p}} \phi_{\langle t^{II} \rangle \langle t^{II} \rangle} \left(\tau_{ij} \frac{\delta (t_i^{II} t_j^{II})}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k t_i^{II} t_j^{II} \right) dS \\
& + \int_{S_{W_p}} n_i \frac{\partial F_{S_{W_p,i}}}{\partial x_m} n_m \frac{\delta x_k}{\delta b_n} n_k dS + \int_{S_{W_p}} F_{S_{W_p,i}} \frac{\delta n_i}{\delta b_n} dS + \int_{S_{W_p}} F_{S_{W_p,i}} n_i \frac{\delta (dS)}{\delta b_n} \\
& - \int_{S_{W_P}} \frac{\partial m_i^a}{\partial x_j} n_j \frac{\delta x_i}{\delta b_n} dS + \int_{S_{W_p}} \mathcal{A}_{\Delta}^{WF} \frac{\partial \Delta^P}{\partial b_n} dS + \int_{S_W} \mathcal{A}_{\Delta}^{WF} \frac{\partial \Delta^P}{\partial b_n} dS \\
& + \int_{S_{W_p}} \Delta_a R^{\Delta} n_k \frac{\delta x_k}{\delta b_n} dS - \int_{S_{W_p}} 2 \Delta_a \frac{\partial \Delta}{\partial x_j} n_j \frac{\partial \Delta}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS
\end{aligned}
\tag{2.97}
$$

2.3 Mesh Parameterization and Movement

In GBM shape optimization, once the SDs of a certain objective function are computed and can be utilized to improve the current geometry, so as to improve the performance of the geometry w.r.t. the current objective function. In GBM robust design shape optimization cases, the *robust* SDs of a certain QoI are computed through either a stochastic or a deterministic uncertainty quantification method and, then, in the same manner, the geometry in question is updated according the robust SDs, leading to a more robust performance w.r.t. to the selected QoI. In order for this process to take place, a mesh movement tool must be employed, with the ability to accurately morph the nodes of the meshed geometry towards the direction dictated by the respective SDs (robust or not). In order to accomplish this task, the computational grid can either be re-meshed according to the updated geometry or it can be deformed in the area affected by the new geometry. Due to the fact, that later option yields by far a lower computational cost, especially when it comes to complicated 2D meshes, it is most often preferred.

In this thesis, the only method used both parameterize and to morph the mesh around the to-be optimized shape are the *volumetric B-Splines* [28], integrated into the OpenFOAM© code in conjunction with the continuous adjoint solver by PCOpt/NTUA.

2.3.1 Volumetric B-Splines

For the parameterization of a specific part of the computational mesh that includes the shape to-be optimized, the user defines a 3D structured control grid a.k.a. *control* or *morphing box*. The box is defined by defining the Cartesian coordinates b_m^{ijk} , $m \in [1,3]$, $i \in [0,I]$, $j \in [1,J]$, $k \in [1,K]$ of the ijk-th control point, as well as the number $I,\ J,\ K$ of control points per grid direction.

The Cartesian coordinates $\vec{x} = [x_1, x_2, x_3]^T = [x, y, z]^T$ of the computational mesh points within the aforementioned control box are defined as

$$x_m(u,v,w) = U_{i,pu}\, V_{j,pv}\, W_{k,pw}\, b_m^{ijk} \ , \quad m = 1,2,3 \tag{2.98}$$

where $U,\ V,\ W$ indicate the B-Splines basis polynomial functions, $pu,\ pv,\ pw$ their respective degrees and $\vec{u} = [u_1, u_2, u_3]^T = [u, v, w]^T$ the mesh point parametric coordinates.

Once the parametric coordinates vector $\vec{u}$ are known, the computation of the Cartesian coordinates vector $\vec{x}$ of any parameterized flow field mesh point is effortless and can be conducted at a negligible computational cost. In order to accurately compute the mesh parametric coordinates a mapping form $\mathbb{R}^3(x,y,z) \to \mathbb{R}^3(u,v,w)$ is needed, thus allowing the volumetric B-Splines to reproduce any geometry with machine accuracy [28]. Therefore, the parametric coordinates (u,v,w) of a said point with Cartesian coordinates $\vec{r} = [x_r, y_r, z_r]^T$ can be computed by solving the system (2.99), as long as the user-defined control points, knot vectors and basis function degrees are known.

$$\mathbf{R}(u,v,w) = \begin{bmatrix} x(u,v,m) - x_r = 0 \\ y(u,v,m) - y_r = 0 \\ z(u,v,m) - z_r = 0 \end{bmatrix} \tag{2.99}$$

where $x_m(u,v,w)$ are computed by utilizing eq. (2.98), based on the given b_m^{ijk} control points coordinates. The 3x3 system of eq. (2.99) can be solved independently for each and every parameterized mesh point through the Newton-Raphson method, once the Jacobian matrix $\partial x_m / \partial u_j$, $m, j = 1,2,3$ is computed and inverted. The Jacobian is computed analytically through a closed form expression resulting by differentiating eq. (2.98) w.r.t. the components of $\vec{u}$.

The aforementioned process must be performed only once at the beginning of the optimization loop, justifying the name ”training phase” of the method. Then, once the displacement of the control points $\partial \vec{b}^{ijk}$ is defined (either through the SDs or the robust SDs), the Cartesian coordinates of each computational mesh point that resides within the morphing box can be inexpensively computed through th use of eq. (2.98).

2.4 The SIMPLE Finite Volume Method

The numerical solution of the primal and the adjoint problem, as thoroughly presented in the respective Subsections 2.1 and 2.2, is performed by employing the SIMPLE pressure based method [29]. First proposed by B. Spalding and S. Patankar in the early 1970s, it is nowadays a generalized and widespread finite volume algorithm, finding application in a variety of CFD cases. The algorithm, culminates to a numerical solution for the NS-Spalart–Allmaras equations by iterating the following steps:

1. Initialization of the pressure field p^*.
2. Computation of the uncorrected velocity $\vec{v}^{\,*}$ and Spalart–Allmaras variable fields $\widetilde{\nu}^{\,*}$ through the solution of the discretized momentum and Spalart–Allmaras equations.
3. Computation of the uncorrected mass fluxes at cell faces.
4. Computation of the pressure-based correction.
5. Computation of the corrected pressure p with the selected under-relaxation factor.
6. Correction of the face mass fluxes.
7. Correction of the velocity $\vec{v}$ and $\widetilde{\nu}$ fields from their uncorrected field values $\vec{v}^{\,*}$ and $\widetilde{\nu}^{\,*}$ by implementing the velocity correction formulas.
8. Reiteration of process by assuming the corrected pressure field p to be the new initial pressure field p^*.

The same algorithmic steps can be implemented for the solution of the adjoint system, where the aforementioned flow variables are switched to their respective adjoints.

The iterative process comes to end once the predefined convergence criterion is met or when the maximum number of iteration is reached. The convergence criterion, being the residual between the old and renewed value of the flow variable, is to be selected so as not to compromise the final solution’s accuracy. For the version of the SIMPLE algorithm available in OpenFOAM© a maximum convergence criterion to achieve a trustworthy result is equal to 10^{-6}.

The convergence process may be complicated by several factors such as the density of mesh not being sufficient in areas with steep velocity gradients or the discretisation schemes used for the NS equations. Another parameter greatly affecting the convergence are the under-relaxation factors. These are weights used to conduct a linear interpolation between the old and the renewed values of the field variables. Their values practically vary between 0.25 to 0.8. The greater the under-relaxation factor the quicker the convergence, yet the lesser the error stability. Meaning that a selection of a high relaxation factor may backfire, by causing an instability to occur in some, if not all, of the flow variable's residuals.

An integral part in the SIMPLE algorithm, is the discretization schemes of the primal or adjoint field equations. More about this topic can be found in [29], for it is not covered in this thesis.

Chapter 3

Stochastic Uncertainty Quantification Methods

In this chapter the two stochastic methods of UQ, utilized in this diploma thesis, are presented and explained. The first, *Monte Carlo* (MC) being the easiest in implementation and yet the most costly, is mostly used to verify the results of the faster and more robust second method, the *non-intrusive Polynomial Chaos Expansion* (niPCE), in later stages of this thesis. Other deterministic UQ methods, such as the *Method of Moments* [19], [3], [4] are not explored in the present thesis.

3.1 Monte Carlo

The MC method [5], [6], being the most typical and straightforward stochastic UQ method, is based on the seemingly simple concept that if an exceedingly large variety of inputs values are given to the uncertain variables of a problem then the statistical moments of each exited QoI will converge towards their real statistical moments. Therefore, the larger the stochastic input pool for the uncertain variables, the more accurate the prediction. The values of the input pool are computed through the use of random number generators, following the predetermined distribution if the uncertain variables.

In CFD-based UQ as well as RDO cases, this method is rarely used due to the several thousands of replicates often needed for the method to produce accurate results. Consequently, the employment of such a method nested into a greater optimization loop is mostly considered unacceptable, due to its great computational cost. Especially, for this thesis, the implementation of the MC method nested into an RDO loop, utilizing the adjoint GBM to compute the SDs, signifies that each optimization step requires the numerical solution of the primal and the adjoint (also measured in two EFS) problems multiple thousands of times, thus mitigating the advantage of the adjoint method. This is mainly the reason for which the MC method is only used for the UQ results verification.

3.2 Polynomial Chaos Expansion

The *Polynomial Chaos expansion* (PCE) was first proposed by N. Wiener in 1938 [30], then only encompassing uncertain variables following a normal distribution through the use of orthogonal *Hermite polynomials*. However, D. Xiu and G. E. Karniadakis [7] developed the generalized Polynomial Chaos theory, based on the Wiener-Askey approach [31] on the generalization of the orthogonal polynomials. The method today can be implemented for uncertain variables following any known statistical distribution. Yet, in the current diploma thesis, since the PCE is applied in conjunction with the DFSS approach, thus accepting a normal distribution for the uncertain variables of the RDO problem, only the initial Hermite polynomials are employed.

In general two methods of implementing the PCE exist, the *intrusive* (iPCE) and *non-intrusive* methods [8]. For the first, the expansion is applied on the inputs and the outputs of the case in question, thus generating a new set of governing PDEs, an boundary conditions. Therefore, these inputs and outputs are modeled as polynomials of order k. It is clear that this method lacks the ability to be generalized, given that for different cases, chaos order k and uncertain variables the governing equations are altered and with them, their solution approach, be it either analytical or numerical. In contrast, the niPCE methods produce a finite number of sample values for the uncertain variables of the case under consideration for which the case's governing equations should be solved, a process a.k.a. sampling. Consequently, the governing PDEs and governing conditions are treated as a black box. Especially for CFD-based UQ or RDO the NS equations are solved independently, for all the value-sets of the uncertain variables designated by the niPCE method. Nevertheless, it is implied that the computational cost, w.r.t. the clock-time, of niPCE methods scales with the number of sampling flow evaluations, dictated by them.

To conclude, the main drawback of niPCE methods, when compared with the intrusive ones, is their much higher computational cost due to the need for multiple evaluations. Yet, niPCE methods are de facto generalized due to their decoupled nature from the case in question, avoiding the painstaking ad-hoc implementation of their iPCE counterparts. Consequently, the same non-intrusive approach is employed in this thesis, on two different RDO scenarios.

3.3 1D non-intrusive PCE

According to the Polynomial Chaos Expansion theory, a function $F = F(x) \in \mathbb{R}$ (referred to as the QoI of the case under consideration) with $x \in \mathbb{R}$ being an uncertain variable following a specified stochastic distribution, can be expanded through an appropriate series of polynomials. This expansion offers the potential to analytically compute the statistical moments of the QoI F, by selecting a suitable cut-off point for it. The computation of the first and the second statistical moments, a.k.a. the mean value and standard deviation of the QoI are of interest in most cases.

Assuming an uncertain variable x distributed according to a probability density function $w(x)$ and $\mathcal{P} = \{p_0(x), p_1(x), ..., p_i(x), ...\}$ a family of polynomials p_i, with i the maximum rank of each polynomial, respectively. In accordance with the PCE theory, $F(x)$ can approximated by a different function $f(x)$ with the same stochastic input x, defined as a linear combination of the polynomials belonging in $\mathcal{P}$:

$$F(x) \approxeq f(x) := \sum_{i=0}^{\infty} a_i p_i(x) \tag{3.1}$$

where $a_i \in \mathbb{R}$ and $f : \mathbb{R} \longrightarrow Y \subseteqq \mathbb{R}$.

The n-th statistical moment of the set Y can be computed as

$$\begin{aligned} \langle y^n \rangle &= \int_D (f(x))^n \, w(x) dx = \int_D \left(\sum_{i=0}^{\infty} a_i p_i \right)^n w(x) dx \\ \Rightarrow \quad \langle y^n \rangle &= \sum_{i_1=0}^{\infty} \cdots \sum_{i_n=0}^{\infty} a_{i_1} \cdots a_{i_n} \int_D p_{i_1} \cdots p_{i_n} w(x) dx \end{aligned} \tag{3.2}$$

The aforementioned integrals, though complex, can be solved analytically due to their polynomial nature. Hence, theoretically every statistical moment of a function F can be calculated. However, equation (3.2) can be simplified by defining $\mathcal{P}$ as a specific family of polynomials, known as *orthogonal polynomials* [32].

3.3.1 Orthogonal Polynomials

The most important feature of orthogonal polynomials is that the inner product (Galerkin projection) of whichever two polynomials ($p_i(x)$) of the same family $\mathcal{P}$, with their corresponding weight function $w(x)$ within their domain D, is equal to zero, unless the two aforementioned polynomials are identical. This property, is depicted as

$$\langle p_i(x), p_j(x) \rangle_w = \int_D p_i(x) p_j(x) w(x) dx = \langle p_i(x), p_j(x) \rangle_w \, \delta_i^j \tag{3.3}$$

where δ_i^j is the *Kronecker Delta.* The definition (3.3) greatly facilitates the solution of equation (3.2).

In case $i = j$, the inner product (3.3) assumes the form of the $w-norm$ of polynomial $p_i(x)$, as depicted below

$$\langle p_i(x), p_j(x) \rangle_w = ||p_i(x)||_w^2 = \gamma_i \tag{3.4}$$

where the square root of the new variable $\sqrt{\gamma_i}$, signifies the *normality metric* of the p_i polynomial. This parameter is not necessarily equal to 1 though, when it is, the orthogonal polynomial family is defined as *canonical.*

A common characteristic of all the various stochastic distributions and their corresponding families of orthogonal polynomial p_n, is that their first (zero degree) polynomial is $p_0(x) = 1\ ,\ \ \forall\ \ w(x)$.

Each stochastic distribution corresponds to a specific family of orthogonal polynomials $p_n(x)$, each defined in a certain domain $x \in D$ and with a certain probability density function $w(x)$. For example, the normal distribution, used exclusively in this thesis, is associated with by the *Hermite* polynomial family $He_n(x)$, defined in the domain $D = (-\infty,\ +\infty)$. Also, the probability density function of a single uncertain variable following a normal distribution with a mean value μ and a standard deviation σ, is given

$$\mathcal{N}(\mu, \sigma^2) :\ w(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{3.5}$$

3.3.2 1st and 2nd Statistical Moments

In order to calculate the first statistical moment, a.k.a. the mean value (μ_F) or expectation ($\mathbb{E}(F)$), equation (3.2) is displayed for $n = 1$

$$\mu_F = \mathbb{E}(F) = \langle y^1 \rangle = \sum_{i_1=0}^{\infty} a_{i_1} \int_D p_{i_1} w(x)dx = \sum_{i=0}^{\infty} a_i \int_D p_i w(x)dx \tag{3.6}$$

The convenience of orthogonal polynomials stems from the following property, according to which the computation of any statistical moment (3.2), can be greatly simplified $\forall\ i > 0$

$$\int_D p_i(x)w(x)dx = \int_D p_i(x) \cdot 1 \cdot w(x)dx = \langle p_i(x), p_0(x) \rangle_w \delta_i^0 = 0 \tag{3.7}$$

Furthermore, given that $\int_D w(x)dx = 1$, by taking into account that the total possibility of any distribution is equal to 1 as well as $p_0(x) = 1$, eq. (3.6) assumes the form

$$\mu_F = a_0 \int_D p_0 w(x)dx + \sum_{i=1}^{\infty} a_i \int_D p_i w(x)dx = a_0 \tag{3.8}$$

The second statistical moment, a.k.a. the *variance* ($var()$), can be computed as

$$var(F) = \left\langle y^2 \right\rangle - \mu_F^2 \tag{3.9}$$

while the *standard deviation* (σ), being the square root of the variance, is defined as

$$\sigma_F = \sqrt{var(F)} = \sqrt{\langle y^2 \rangle - \mu_F^2} \tag{3.10}$$

According to the PCE theory, $\langle y^2 \rangle$ is given by

$$\left\langle y^2 \right\rangle = \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} a_{i_1} a_{i_2} \int_D p_{i_1} p_{i_2} w(x)dx \tag{3.11}$$

By using the orthogonality properties described in eqs. (3.3) and (3.4), eq. (3.11) is simplified as

$$\left\langle y^2 \right\rangle = \sum_{i=0}^{\infty} a_i^2 \int_D p_i^2 w(x)dx = a_i^2 \gamma_i \tag{3.12}$$

The γ_i parameter can be further simplified through the use of canonical orthogonal polynomial families, a.k.a. *ortho-canonical.* A way to generate such families comes by dividing the polynomials p_n by their normality metric γ_n, as displayed

$$\tilde{p_n}(x) = \frac{p_n(x)}{||p_n(x)||} = \frac{p_n(x)}{\gamma_n} \tag{3.13}$$

Consequently, $||\tilde{p_i}(x)||_w = 1$, and by using ortho-canonical polynomials, eq. (3.12) becomes

$$\left\langle y^2 \right\rangle = \sum_{i=0}^{\infty} a_i^2 \tag{3.14}$$

and the standard deviation from eq. (3.10) becomes

$$\sigma_F = \sqrt{\sum_{i=0}^{\infty} a_i^2 - a_0^2} = \sqrt{\sum_{i=1}^{\infty} a_i^2} \tag{3.15}$$

Thus, by knowing the probability distribution the uncertain variable x follows and its corresponding ortho-canonical polynomials, the PCE coefficients a_i can be computed. A critical choice is the cut-off points of the series in eqs. (3.8) and (3.15), in order to compute the mean value and standard deviation of F.

3.3.3 Polynomial Chaos Expansion Coefficients

So far, the statistical moments of QoI function $F(x)$ were defined for an infinite number of terms in their respective expansions. Thus, prior to the calculation of the PCE coefficients a_i, the cut-off point k, a.k.a. as *chaos order* of the expansion must be decided. It can be assumed, that the higher the value of k, the higher becomes the accuracy of PCE-computed statistical moments, while simultaneously the computational cost is expected to increase. The importance of the choice of chaos order k, stems from the aforementioned equilibrium between accuracy and cost.

Thus, for a cut-off point k for the expansion, the function F and its mean value μ_F and standard deviation σ_F can be expanded as

$$F(x) \approxeq \sum_{i=0}^{k} a_i \tilde{p}_i(x) \tag{3.16}$$

$$\mu_F \approxeq a_0 \tag{3.17}$$

$$\sigma_F \approxeq \sqrt{\sum_{i=1}^{k} a_i^2} \tag{3.18}$$

Thus, eq. (1.3) for the computation of the robustness metric F_R according to the DFSS, assumes the form

$$F_R \approxeq a_0 + \kappa \sqrt{\sum_{i=1}^{k} a_i^2} \tag{3.19}$$

The computation of the $k+1$ PCE coefficients a_i, the *Galerkin projection* as well as the ortho-canonical polynomials $\tilde{p}_i$ are reused in equation (3.20)

$$\begin{aligned} \langle f(x), \tilde{p}_i(x) \rangle_w &= \left\langle \sum_{j=0}^{k} a_j \tilde{p}_j(x), \tilde{p}_i(x) \right\rangle_w = \sum_{j=0}^{k} a_j \int_D \tilde{p}_j(x) \tilde{p}_i(x) w(x) dx \\ \Rightarrow \quad \langle f(x), \tilde{p}_i(x) \rangle_w &= a_i ||\tilde{p}_i(x)||_w^2 = a_i \ , \quad i = 0, 1, ..., k \end{aligned} \tag{3.20}$$

In addition, according to eq. (3.1) the QoI function $F(x)$ can be replaced as

$$\langle f(x), \tilde{p}_i(x) \rangle_w = \int_D f(x) \tilde{p}_i(x) w(x) dx \approxeq \int_D F(x) \tilde{p}_i(x) w(x) dx \ , \ \ i = 0, 1, ..., k \tag{3.21}$$

Thus, eqs. (3.20) and (3.21) lead to the final equations for the a_i coefficients

$$a_i = \int_D F(x) \tilde{p}_i(x) w(x) dx \ , \quad i = 0, 1, ..., k \tag{3.22}$$

In order to successfully compute the PCE coefficients by making use of eq. (3.22), it is essential to assume the chaos order k and the stochastic distribution of the uncertain variables. Therefore, the PDF, weight functions and ortho-canonical polynomial family should, thus, all be known to user.

The integration present in eq. (3.22), when not solved analytically requires the call to the QoI function $F(x)$ a finite number of times. In this diploma thesis, any call to $F(x)$ refers to the solution of the primal problem as described in Section 2.1. Hence, in aerodynamic UQ and RDO cases, the computational cost of methods using PCE coefficients, scales with the number of times the QoI (F) is computed.

3.3.4 Differentiation w.r.t. the Design Variables

Since a gradient based RDO is used in this thesis, the gradients ($\nabla\mu_F$, $\nabla\sigma_F$) of the 1st and 2nd statistical moments must be computed as mentioned in Subsection 1.2.2. These, can be computed by differentiating the statistical moments w.r.t. the design variables b_n, as follows

$$\frac{\partial \mu_F}{\partial b_n} \approxeq \frac{\partial a_0}{\partial b_n} \ , \quad n = 1, 2, ..., N \tag{3.23}$$

$$\frac{\partial \sigma_F}{\partial b_n} \approxeq \frac{\sum_{i=1}^{k} a_i \frac{\partial a_i}{\partial b_n}}{\sqrt{\sum_{i=1}^{k} a_i^2}} = \frac{1}{\sigma_F} \sum_{i=1}^{k} a_i \frac{\partial a_i}{\partial b_n} \ , \quad n = 1, 2, ..., N \tag{3.24}$$

where the derivatives of the PCE coefficients can be computed by directly differentiating eq. (3.22) w.r.t. to the design variables, resulting to

$$\frac{\partial a_i}{\partial b_n} = \frac{\partial}{\partial b_n} \left(\int_D F(x, \vec{b}) \tilde{p}_i(x) w(x) dx \right) = \int_D \frac{\partial F(x, \vec{b})}{\partial b_n} \tilde{p_i}(x) w(x) dx \ , \quad i = 0, 1, ..., k \tag{3.25}$$

given that, according to the previous analysis, the QoI is dependent on a single uncertain variable $x \in \mathbb{R}$ and many design variables comprising the design variable vector $\vec{b} \in \mathbb{R}^N$.

Finally, the derivatives of the robust metric F_R, a.k.a. the robust SDs, can be formulated by combining eqs. (3.23) and (3.24) with eq. (1.4) of Subsection 1.2.2

$$\frac{\partial F_R}{\partial b_n} = \frac{\partial \mu_F}{\partial b_n} + \kappa \frac{\partial \sigma_F}{\partial b_n} \approxeq \frac{\partial a_0}{\partial b_n} + \kappa \frac{\sum_{i=1}^{k} a_i \frac{\partial a_i}{\partial b_n}}{\sigma_F} \ , \quad n = 1, 2, ..., N \tag{3.26}$$

3.3.5 Gauss Quadrature Integration

Having defined the PCE coefficients analytically in eq. (3.22), a common way of numerically computing them, referred to as *Gauss Quadrature* (GQ) [32], are explained.

According to the base GQ integration method, the integral of a function $h(x)$ within a domain D can numerically be computed as the sum of N_{GQ} terms, each term being a product of a weight r_i and a value of the function h, computed for specific values of its variable x, referred to as *Gauss nodes* z_i'. This definition is formulated as follows

$$\int_D h(x)dx = \sum_{i=1}^{N_{GQ}} r_i \, h(z_i') \tag{3.27}$$

The greater the number of nodes, the higher the method's accuracy and computational cost, given that more calls of the $h(x)$ function are needed.

Let $h(x) = w(x)f(x)$, where $w(x)$ denotes the probability function (defined according to the stochastic distribution of the uncertain variable x) and f the polynomial approximation of the QoI function F from eq. (3.1). The weights and Gauss nodes are re-defined as ω_i and z_i, respectively, thus converting eq. (3.27) to

$$\int_D h(x)dx = \int_D w(x)f(x)dx = \sum_{i=1}^{N_{GQ}} \omega_i \, f(z_i) \tag{3.28}$$

For one uncertain variable, the GQ is a easy to implement and affordable method for the integration of equations (3.22) and (3.25), partly due to the fact that ortho-canonical polynomials are for the expansion of $f(x)$. The roots of orthogonal polynomials are all are simple, real and within their respective domain D, while their number is equal to the polynomial's degree. These roots define the Gauss nodes z_i and constitute the best possible distribution of nodes for the minimization of the GQ method's error [32]. Therefore, for the selected cut-off point k a.k.a. chaos order, $N_{GQ} = k + 1$ Gauss nodes are needed for maximum accuracy and the values of the nodes z_i are defined the roots of the polynomial $p_{k+1}(x)$. In the meantime, the weights ω_i are defined as

$$\omega_i = \frac{A_{k+1}}{A_k} \cdot \frac{\gamma_k}{p'_{k+1}(z_i)p_k(z_i)} \tag{3.29}$$

where A_k the coefficient of the x^k term of the orthogonal polynomial of degree k and so on.

The *probabilists'* Hermite polynomial family $He_n(x)$ is implemented for the solution of integrals (3.22) and (3.25). Thus the integration method takes the name *Gauss Hermite Quadrature* (GHQ). This orthogonal polynomial family, differing a bit from the *physicists'* Hermite polynomial family, is thoroughly explored in Appendix B and simply defined as a recurring formula

$$He_{k+1}(x) = x\, He_k(x) - k\, He_{k-1}(x)\ ,\quad He_0(x) = 1\ and\ He_1(x) = x \tag{3.30}$$

where the coefficient of the x^k term of the polynomial $He_k(x)$ is equal to $A_k = 1$ and the w-norm of this family is defined as

$$||He_k(x)||_w^2 = \gamma_k = \langle He_k(x), He_k(x)\rangle_w = \int_{-\infty}^{+\infty} (He_k(x))^2\, w(x)dx = k! \tag{3.31}$$

The *Hermite* polynomials presented in eq. (B) correspond to the *standardized* normal distribution (with $\mu = 0$ and $\sigma = 1$). Hence, their probability density function, from equation (3.5), assumes the form

$$w(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{3.32}$$

According to the eq. (3.13), the probabilists' Hermite polynomials can be converted to an ortho-canonical through this process

$$\widetilde{He}_k(x) = \frac{He_k(x)}{||He_k(x)||} = \frac{He_k(x)}{\sqrt{k!}} \tag{3.33}$$

and their w-norm is formulated as follows

$$||\widetilde{He}_k(x)||_w^2 = 1\ ,\quad \forall\ \ k = 1, 2, ... \tag{3.34}$$

causing $A_k = 1/\sqrt{k!}$.

Given that not every normal distribution is standardized, a specific transform must be applied so as to convert the uncertain variable from the standardized normal distribution $z \in \mathcal{N}(0, 1)$ to the generalized normal distribution $x \in \mathcal{N}(\mu, \sigma^2)$. This is done through the linear transform

$$\frac{x - \mu}{\sigma} = z\ \rightarrow\ dx = \sigma dz \tag{3.35}$$

Therefore, according to the transform (3.35), the weight function (3.5), eq. (3.22), for $i = 0, 1, ..., k$ assumes the analytical form

$$a_i = \int_{-\infty}^{+\infty} F(x)\widetilde{He}_i(\frac{x-\mu}{\sigma})w(x)dx = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} F(\sigma z + \mu)\widetilde{He}_i\,(z)\, e^{-\frac{z^2}{2}} dz \tag{3.36}$$

and by implementing GHQ to numerically solve the integral (3.36), the final form emerges

$$a_i = \int_{-\infty}^{+\infty} F(x)\widetilde{He}_i(\frac{x-\mu}{\sigma})w(x)dx = \sum_{j=1}^{k+1} \omega_j^{(k+1)} F(\sigma z_j^{(k+1)} + \mu)\widetilde{He}_i\left(z_j^{(k+1)}\right) \tag{3.37}$$

Meanwhile, by employing the GHQ method, the integral for the computation of the derivatives of the PCE coefficients, eq. (3.25), becomes

$$\begin{aligned} \frac{\partial a_i}{\partial b_n} &= \frac{\partial}{\partial b_n}\left(\int_{-\infty}^{+\infty} F(x)\widetilde{He}_i(\frac{x-\mu}{\sigma})w(x)dx\right) = \\ &= \sum_{j=1}^{k+1} \omega_j^{(k+1)} \frac{\partial}{\partial b_n}\left(F(\sigma z_j^{(k+1)} + \mu)\right)\widetilde{He}_i\left(z_j^{(k+1)}\right) \end{aligned} \tag{3.38}$$

where the $(k+1)$ index on the terms ω_j and z_j signify that they constitute solutions of the probabilists' Hermite polynomial He_{k+1}, i.e. the one of degree $k+1$, where k the selected chaos order for the niPCE method.

Additionally, the weights ω_j, according to equation (3.29), can be computed as

$$\omega_j^{(k+1)} = \frac{A_{k+1}}{A_k} \cdot \frac{\gamma_k}{\widetilde{He}'_{k+1}(z_j^{(k+1)})\widetilde{He}_k(z_j^{(k+1)})} = \frac{1}{(k+1)\widetilde{He}_k^2(z_j^{(k+1)})} \tag{3.39}$$

Finally, the niPCE mean value of a QoI function $F(x)$, for x following $\mathcal{N}(\mu, \sigma)$, by implementing the GHQ integration, is formulated as

$$\mu_F = a_0 = \sum_{j=1}^{k+1} \omega_j^{(k+1)} F(\sigma z_j^{(k+1)} + \mu)\widetilde{He}_0\left(z_j^{(k+1)}\right) = \sum_{j=1}^{k+1} \omega_j^{(k+1)} F(\sigma z_j^{(k+1)} + \mu) \tag{3.40}$$

given that $\widetilde{He}_0 = 1$, while the respective mean value's gradient (3.23) w.r.t. the design variable b_n is computed as

$$\frac{\partial \mu_F}{\partial b_n} = \frac{\partial a_0}{\partial b_n} = \sum_{j=1}^{k+1} \omega_j^{(k+1)} \frac{\partial}{\partial b_n}\left(F(\sigma z_j^{(k+1)} + \mu)\right) \tag{3.41}$$

Furthermore, the niPCE standard deviation of a QoI function $F(x)$, for x following $\mathcal{N}(\mu, \sigma)$, by implementing the GHQ integration, is formulated as

$$\sigma_F = \sqrt{\sum_{i=1}^{k} a_i^2} = \sqrt{\sum_{i=1}^{k}\left[\sum_{j=1}^{k+1} \omega_j^{(k+1)} F(\sigma z_j^{(k+1)} + \mu)\widetilde{He}_i\left(z_j^{(k+1)}\right)\right]^2} \tag{3.42}$$

while the corresponding standard deviation's gradient eq. (3.24), w.r.t. the design variable b_n, assumes the form

$$\begin{aligned} \frac{\partial \sigma_F}{\partial b_n} = \frac{1}{\sigma_F} \sum_{i=1}^{k} \left[\sum_{j=1}^{k+1} \omega_j^{(k+1)} F(\sigma z_j^{(k+1)} + \mu) \widetilde{He}_i \left(z_j^{(k+1)} \right) \right] \cdot \\ \cdot \left[\sum_{j=1}^{k+1} \omega_j^{(k+1)} \frac{\partial}{\partial b_n} \left(F(\sigma z_j^{(k+1)} + \mu) \right) \widetilde{He}_i \left(z_j^{(k+1)} \right) \right] \end{aligned} \tag{3.43}$$

3.4 Multi-dimensional non-intrusive PCE

In this Subsection the niPCE method is generalized for a QoI function $F = F(\vec{x})$ dependent on multiple uncertain variables, constituting the uncertain variable vector $\vec{x} = [x_1, x_2, ..., x_M]^T \in \mathbb{R}^M$. For $D_1, D_2, ..., D_M$ the domain of each of the respective components of $\vec{x}$, the domain of the QoI function can be defined as

$$\vec{D} = D_1 \times D_2 \times \cdots \times D_M \tag{3.44}$$

Given that each uncertain variable x_n follows a certain stochastic distribution, with a corresponding probability density function $w_n(x_n)$, $n = 1, 2, ..., M$, according to the PCE theory F can be approximated through a linear combination of polynomials being part of the family $\Psi = \{\psi_0(\vec{x}), \psi_1(\vec{x}), ..., \psi_M(\vec{x}), ...\}$, as follows

$$F(\vec{x}) \approxeq f(\vec{x}) := \sum_{i=0}^{\infty} a_i \psi_i(\vec{x}) \tag{3.45}$$

where $a_i \in \mathbb{R}$ and $f : \vec{D} \subseteqq \mathbb{R}^M \longrightarrow Y \subseteqq \mathbb{R}$.

Let the multidimensional probability density function a.k.a. the product of the aforementioned distributions $w_j(x_j)$ be defined as

$$W(\vec{x}) = \prod_{j=1}^{M} w_j(x_j) = w_1(x_1)\, w_2(x_2) \cdots w_M(x_M) \tag{3.46}$$

Therefore, the n-th statistical moment of the set Y is formulated

$$\begin{aligned} \langle y^n \rangle = \int_{\vec{D}} (f(\vec{x}))^n W(\vec{x}) d\vec{x} = \int_{\vec{D}} \left(\sum_{i=0}^{\infty} a_i \psi_i(\vec{x}) \right)^n W(\vec{x}) d\vec{x} = \\ = \int_{D_1} \int_{D_2} \cdots \int_{D_M} \underbrace{\left(\sum_{i=0}^{\infty} a_i \psi_i(\vec{x}) \right) \cdots \left(\sum_{i=0}^{\infty} a_i \psi_i(\vec{x}) \right)}_{n \ times} \prod_{j=1}^{M} w_j(x_j) \; dx_1 dx_2 \cdots dx_M \end{aligned} \tag{3.47}$$

And by applying the distributive property, eq. (3.47) becomes

$$\langle y^n \rangle = \sum_{i_1=0}^{\infty} \cdots \sum_{i_n=0}^{\infty} a_{i_1} \cdots a_{i_n} \int_{D_1} \cdots \int_{D_M} \psi_{i_1}(\vec{x}) \cdots \psi_{i_n}(\vec{x}) w_1(x_1) \cdots w_M(x_M)\, dx_1 \cdots dx_M \tag{3.48}$$

Again, the definition of ψ_i as multi-dimensional orthogonal polynomials may greatly simplify the expressions of the statistical moments (3.48).

3.4.1 Multi-dimensional Orthogonal Polynomials

Multi-dimensional orthogonal polynomials possess the same properties with their one-dimensional counterparts described in Subsection 3.3.1, if each polynomial psi_n of order k is defined as a product of one-dimensional orthogonal p_{i_l}, in such a fashion that the sum of their degrees i_l is equal to the desired order k. These polynomials assume the form

$$\psi_n(\vec{x}) = \psi_{m_j^k}(\vec{x}) = \prod_{l=1}^{M} p_{i_l}(x_l) \ , \quad \sum_{l=1}^{M} i_l = k \tag{3.49}$$

The index m_j^k is included for now and will be explained later. Thus, the simpler n index is used to describe the degree a multi-dimensional polynomial ψ.

According to the definition of polynomials belonging in the Ψ family, given in eq. (3.49), the Galerkin projection (inner product) of any two of these polynomials is equal to zero, except if the two are identical. Two polynomials of the Ψ family are identical only if all consecutive indexes i_l of all the polynomials p_il that comprise them are equal. This inner product between two multi-dimensional polynomials, of degrees i and j, yields

$$\begin{aligned}
&\langle \psi_i(\vec{x}), \psi_j(\vec{x}) \rangle_W = \int_{\vec{D}} \psi_i(\vec{x}) \psi_j(\vec{x}) W(\vec{x})\, d\vec{x} = \\
&= \int_{D_1} \cdots \int_{D_M} \prod_{l=1}^{M} p_{i_l}(\vec{x_l}) \prod_{l=1}^{M} p_{j_l}(\vec{x_l}) \prod_{l=1}^{M} w_l(\vec{x_l})\, dx_1 \cdots dx_M = \\
&= \prod_{l=1}^{M} \left(\int_{D_l} p_{i_l}(x_l) p_{j_l}(x_l) w_l(x_l) dx_l \right) = \prod_{l=1}^{M} \left(\delta_{i_l}^{j_l} \int_{D_l} p_{i_l}^2(x_l) w_l(x_l) dx_l \right) = \\
&= \prod_{l=1}^{M} \left(\delta_{i_l}^{j_l} \right) \int_{D_1} \cdots \int_{D_M} \prod_{l=1}^{M} \left[p_{i_l}^2(x_l) w_l(x_l) \right] dx_1 \cdots dx_M = \\
&= \delta_i^j \int_{\vec{D}} \prod_{l=1}^{M} \left(p_{i_l}^2(x_l) \right) W(\vec{x})\, d\vec{x} = \delta_i^j \int_{\vec{D}} \left(\psi_i(\vec{x}) \right)^2 W(\vec{x})\, d\vec{x}
\end{aligned} \tag{3.50}$$

Therefore, concluding to the expression

$$\langle \psi_i(\vec{x}), \psi_j(\vec{x}) \rangle_W = \int_{\vec{D}} \psi_i(\vec{x}) \psi_j(\vec{x}) W(\vec{x})\, d\vec{x} = \langle \psi_i(\vec{x}), \psi_i(\vec{x}) \rangle_W \, \delta_i^j \tag{3.51}$$

Now, the Galerkin projection of two identical polynomials ($i = j$) of the Ψ family, is equal to the W-norm of the ψ_i polynomial, which is proven by

$$\begin{aligned}
\langle \psi_i(\vec{x}), \psi_i(\vec{x}) \rangle_W &= \int_{\vec{D}} \psi_i^2(\vec{x}) W(\vec{x})\, d\vec{x} = \\
&= \int_{D_1} \cdots \int_{D_M} \prod_{l=1}^{M} p_{i_l}^2(\vec{x_l}) \prod_{l=1}^{M} w_l(\vec{x_l})\, dx_1 \cdots dx_M = \\
&= \prod_{l=1}^{M} \left(\int_{D_l} p_{i_l}^2(x_l) w_l(x_l) dx_l \right) = \prod_{l=1}^{M} \left(||p_{i_l}(x_l)||_{w_l}^2 \right) = \\
&= \left(\prod_{l=1}^{M} ||p_{i_l}(x_l)||_{w_l} \right)^2 = ||\psi_i(\vec{x})||_W^2
\end{aligned} \tag{3.52}$$

where the W-norm of ψ_i is defined as $||\psi_i(\vec{x})||_W = \prod_{l=1}^{M} ||p_{i_l}(x_l)||_{w_l}$.

To summarize, eq. (3.52) is simplified to

$$\langle \psi_n(\vec{x}), \psi_n(\vec{x}) \rangle_W = ||\psi_n(\vec{x})||_W^2 = \gamma_n \tag{3.53}$$

where $\sqrt{\gamma_n}$ depicts the normality metric, now defined for multidimensional orthogonal polynomials ψ_n. As already stated in Subsection 3.3, if $\gamma_n = 1$ then the ψ_n polynomials are part of the ortho-canonical families, a subset of orthogonal polynomial families. Also, if, in all expression concerning multi-dimensional orthogonal polynomials, the number of uncertain variables is set $M = 1$, then the corresponding expressions for 1D polynomials, explored in 3.3, will emerge.

In eq. (3.49) it is stated that a multi-dimensional polynomial ψ of order k, is to a product of M 1D polynomials p_{i_l}, of which the sum of their degrees i_l is equal to the initial order k. There is, thus, a need to describe the different combinations of M number of integers in which the integer k can be expanded. According to the mathematical field of set theory and combinatrics, these combinations are referred to as *multiset* of k and the greater the value of k, the greater the multiset or the number of different combinations of integers. Specifically, number of possible combinations is given by expression (3.54).

$$\left(\!\! \binom{M}{k} \!\! \right) = \binom{k + M - 1}{k} = \frac{(k + M - 1)!}{k!(M - 1)!} = \frac{M(M + 1)(M + 2) \cdots (M + k - 1)}{k!} \tag{3.54}$$

Additionally the index j used in eq. 3.49, serves to classify the aforementioned combination as follows

$$1 \leqslant j \leqslant \left(\begin{array}{c} k+M-1 \\ k \end{array} \right) \tag{3.55}$$

This classification is relative and does not take into account the possible combinations of integers with a sum smaller than the respective k.

According to combinatorics, the number of combinations of M integers with a sum lower or equal to k_{max} is calculated as

$$\left(\begin{array}{c} k_{max}+M \\ k_{max} \end{array} \right) = \frac{(k_{max}+M)!}{k!M!} = \frac{M(M+1)(M+2)\cdots(M+k_{max})}{k_{max}!} \tag{3.56}$$

The, aforementioned and previously not explained, index m_j^k of eq. (3.49), is used to provide an absolute classification for all possible combinations of integers, adding up to $k = 0, 1, 2, ..., k_{max}$. This index scales as follows

$$0 \leqslant m_j^k \leqslant \frac{(k_{max}+M)!}{k_{max}!M!} - 1 \tag{3.57}$$

This algorithmic method of classification is known as *Full Factorial Design* is based on the restriction dictating that the sum of the indexes of 1D polynomials p_{i_l} (producing the $\psi_{m_j^k}(\vec{x})$) cannot be greater than the maximum desired chaos order k_{max}. In this manner, the surplus indexes are eliminated and the right number of combinations remain.

3.4.2 1st and 2nd Statistical Moments

At long last, now that multi-dimensional orthogonal polynomials are defined, it is high time to apply their simplifying properties in eq. (3.48) in order to formulate the first and second statistical moments of $F(\vec{x})$, without the application of any cut-off points to the respective series, for now.

According to eq. (3.48) for $n = 1$, the first statistical moment, aka the mean value, yields

$$\mu_F = \langle y^1 \rangle = \int_{\vec{D}} f(\vec{x})W(\vec{x})d\vec{x} = \sum_{i=0}^{\infty} a_i \int_{\vec{D}} \psi_i(\vec{x})W(\vec{x})d\vec{x} \tag{3.58}$$

Given $\int_{D_i} w(x_i)dx_i = 1 \; \forall \; i$ (as explained in Subsection 3.3.2), then, regardless of the distribution, the product of them becomes

$$\int_{\vec{D}} W(\vec{x})d\vec{x} = \int_{D_1} w(x_1)dx_1 \cdots \int_{D_M} w(x_M)dx_M = 1 \cdot 1 \cdots 1 = 1 \tag{3.59}$$

thus proving that the total possibility of the appearance of all phenomena is equal to 1.

In addition, the zero degree polynomial is equal to $\psi_0(\vec{x}) = 1$, given that is a product of 1D unit polynomials, regardless of the stochastic distribution $W(\vec{x})$. Thus this property in conjunction with the definition described in eq. (3.51) gives

$$\int_{\vec{D}} \psi_i(\vec{x}) W(\vec{x}) d\vec{x} = \int_{\vec{D}} \psi_i(\vec{x}) \cdot 1 \cdot W(\vec{x}) d\vec{x} = \int_{\vec{D}} \psi_i(\vec{x}) \psi_0(\vec{x}) W(\vec{x}) d\vec{x} = 0 \ \forall \ i > 0 \tag{3.60}$$

Therefore, by utilizing eq. (3.59) and (3.60), eq. (3.58) assumes the form

$$\begin{aligned} \mu_F = a_0 \int_{\vec{D}} \psi_0(\vec{x}) W(\vec{x}) d\vec{x} + \sum_{i=1}^{\infty} a_i \int_{\vec{D}} \psi_i(\vec{x}) W(\vec{x}) d\vec{x} = \\ = a_0 \int_{\vec{D}} 1 \cdot W(\vec{x}) d\vec{x} \implies \mu_F = a_0 \end{aligned} \tag{3.61}$$

Furthermore, it is reminded that the second statistical moment is used to compute the variance of of F. The standard deviation of a quantity, according to eq. (3.9) is defined as the square root of its variance. Therefore, the second statistical moment is formulated from eq. (3.48) for $n = 2$ as follows

$$\langle y^2 \rangle = \int_{\vec{D}} \left(f(\vec{x})\right)^2 W(\vec{x}) d\vec{x} = \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} a_{i_1} a_{i_2} \int_{\vec{D}} \psi_{i_1}(\vec{x}) \psi_{i_2}(\vec{x}) W(\vec{x}) d\vec{x} \tag{3.62}$$

By applying the orthogonality property (3.51) and the multi-dimensional W-norm definition (3.53), eq. (3.62) becomes

$$\langle y^2 \rangle = \sum_{i=0}^{\infty} a_i^2 \int_{\vec{D}} \psi_i^2(\vec{x}) W(\vec{x}) d\vec{x} = \sum_{i=0}^{\infty} a_i^2 \ ||\psi_i(\vec{x})||_W^2 = \sum_{i=0}^{\infty} a_i^2 \ \gamma_i \tag{3.63}$$

The use of ortho-canonical polynomials $\widetilde{\psi}_i(\vec{x})$ gives $\gamma_i = 1$. This can be easily done, by dividing the orthogonal polynomials of any family with their respective normality metric, as stated below

$$\widetilde{\psi}_i(\vec{x}) = \frac{\psi_i(\vec{x})}{||\psi_i(\vec{x})||} = \frac{\psi_i(\vec{x})}{\sqrt{\gamma_i}} \tag{3.64}$$

Thus, giving rise to the most simplified formulation for the 2nd moment:

$$\langle y^2 \rangle = \sum_{i=0}^{\infty} a_i^2 \tag{3.65}$$

According to eq. (3.10) and (3.65), the standard deviation of $F(\vec{x})$ assumes the form

$$\sigma_F = \sqrt{\sum_{i=0}^{\infty} a_i^2 - a_0^2} = \sqrt{\sum_{i=1}^{\infty} a_i^2} \tag{3.66}$$

3.4.3 Polynomial Chaos Expansion Multi-Dimensional Coefficients

In the same manner as in Subsection 3.3.3, in order to compute the PCE coefficients a_i for a finite number of terms, a cut-off point must be set by the user, to the expansion of $f(\vec{x})$ from (3.45). According to the previous analysis, f can be expanded into $N_{cut} = \begin{pmatrix} k_{max} + M \\ k_{max} \end{pmatrix}$ terms, comprising all viable combinations of 1D orthogonal polynomials, with the sum of their degrees being lesser or equal to the desired chaos order k. The expansion of f along with, the mean value and the standard deviation of F assume the forms

$$F(\vec{x}) \approxeq f(\vec{x}) = \sum_{i=0}^{N_{cut}} a_i \widetilde{\psi}_i(\vec{x}) \tag{3.67}$$

$$\mu_F \approxeq a_0 \tag{3.68}$$

$$\sigma_F \approxeq \sqrt{\sum_{i=1}^{N_{cut}} a_i^2} \tag{3.69}$$

and the expression for the robustness metric F_R for multiple uncertain variables assumes the same form (3.19), with the one presented in Subsection 3.3.3.

The Galerkin projection of f and multi-dimensional ortho-canonical polynomials $\widetilde{\psi}$ is expanded in two distinct manners

$$\left\langle f(\vec{x}), \widetilde{\psi}_i(\vec{x}) \right\rangle_W = \int_{\vec{D}} f(\vec{x})\widetilde{\psi}_i(\vec{x})W(\vec{x})\, d\vec{x} \ \approxeq \int_{\vec{D}} F(\vec{x})\widetilde{\psi}_i(\vec{x})W(\vec{x})\, d\vec{x} \tag{3.70}$$

$$\left\langle f(\vec{x}), \widetilde{\psi}_i(\vec{x}) \right\rangle_W = \left\langle \sum_{l=0}^{N} a_l \widetilde{\psi}_l(\vec{x}), \widetilde{\psi}_i(\vec{x}) \right\rangle_W = a_i ||\widetilde{\psi}_i(\vec{x})||_W^2 = a_i \tag{3.71}$$

Eqs. (3.70) and (3.70) are utilized to compute the niPCE coefficients a_i, as follows

$$a_i \approxeq \int_{\vec{D}} F(\vec{x})\widetilde{\psi}_i(\vec{x})W(\vec{x})\, d\vec{x}\ , \quad i = 1, 2, ..., N \tag{3.72}$$

It is reminded that, in this thesis, all uncertain variables follow normal distributions. Therefore, their corresponding probabilists' Hermite polynomials are implemented, in their canonical form ($\widetilde{He}$). Consequently, the approximate polynomial expansion of the QoI function, for M uncertain variables x_l, each with their respective mean value μ_l and standard deviation σ_l for $l = 1, 2, ..., M$, becomes

$$F(\vec{x}) \approxeq \sum_{i=0}^{N_{cut}} \left(a_i \prod_{l=1}^{M} \widetilde{He}_{i_l} \left(\frac{x_l - \mu_l}{\sigma_l} \right) \right) \tag{3.73}$$

For M uncertain variables following standardized normal distributions $\mathcal{N}$ (with $\mu = 0$ and $\sigma = 1$), the multidimensional probability density function, according to (3.46) and (3.32), is formulated as

$$W_{He}(\vec{x}) = \prod_{j=1}^{M} \frac{1}{\sqrt{2\pi}} e^{-\frac{x_j^2}{2}} = \frac{1}{(2\pi)^{\frac{M}{2}}} \cdot e^{-\frac{1}{2}\sum_{j=1}^{M} x_j^2} \tag{3.74}$$

Additionally, given that the Hermite polynomials' domain is $D = [-\infty, +\infty]$ and that all uncertain variables follow a standardized normal distribution, the analytical expression for the niPCE coefficients a_i become

$$a_i = \frac{1}{(2\pi)^{\frac{M}{2}}} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} F(\vec{x}) \widetilde{\psi}_i(\vec{x}) e^{-\frac{\sum_{j=1}^{M} x_j^2}{2}} \, dx_1 dx_2 \cdots dx_M \tag{3.75}$$

On the other hand, according to eq. (3.5), when each of the M uncertain variables follows a generalized normal distribution ($x_j \in \mathcal{N}(\mu_j, \sigma_j^2)$, $j = 1, 2, ..., M$), the the generalized analytical expression for the coefficients formulates as

$$a_i = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} F(\vec{x}) \widetilde{\psi}_i(\vec{z}) \left(\prod_{j=1}^{M} \frac{e^{-\frac{z_j^2}{2}}}{\sigma_j \sqrt{2\pi}} \right) dx_1 dx_2 \cdots dx_M \tag{3.76}$$

where each component of the vector $\vec{z} \in \mathbb{R}^M$ is defined, in order to denote the linear transform between generalized x_j and standardized z_j normal distributions, as

$$z_j = \frac{x_j - \mu_j}{\sigma_j} \ , \quad j = 1, 2, ..., M \tag{3.77}$$

3.4.4 Differentiation w.r.t. the Design Variables

The derivatives of the 1st and 2nd statistical moments w.r.t. the design variables (b_n , $n = 1, 2, ..., N$), for the multiple uncertain variables (x_j , $j = 1, 2, ..., M$), are formulated as

$$\frac{\partial \mu_F}{\partial b_n} \approxeq \frac{\partial a_0}{\partial b_n} \ , \quad n = 1, 2, ..., N \tag{3.78}$$

$$\frac{\partial \sigma_F}{\partial b_n} \approxeq \frac{\sum_{i=1}^{N_{cut}} a_i \frac{\partial a_i}{\partial b_n}}{\sqrt{\sum_{i=1}^{N_{cut}} a_i^2}} = \frac{1}{\sigma_F} \sum_{i=1}^{N_{cut}} a_i \frac{\partial a_i}{\partial b_n} \ , \quad n = 1, 2, ..., N \tag{3.79}$$

Given that the QoI function is also dependent upon the design variables $F = F(\vec{b}, \vec{x})$, the derivatives of the PCE coefficients can be computed by directly differentiating eq. (3.76) w.r.t. to the design variables, resulting to

$$\frac{\partial a_i}{\partial b_n} = \frac{\partial}{\partial b_n} \left(\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} F(\vec{b}, \vec{x}) \widetilde{\psi}_i(\vec{z}) \left(\prod_{j=1}^{M} \frac{e^{-\frac{z_j^2}{2}}}{\sigma_j \sqrt{2\pi}} \right) dx_1 dx_2 \cdots dx_M \right) =$$
$$= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \frac{\partial F(\vec{b}, \vec{x})}{\partial b_n} \widetilde{\psi}_i(\vec{z}) \left(\prod_{j=1}^{M} \frac{e^{-\frac{z_j^2}{2}}}{\sigma_j \sqrt{2\pi}} \right) dx_1 dx_2 \cdots dx_M \ , \quad i = 1, 2, ..., N_{cut} \tag{3.80}$$

The derivatives of the robust metric $F_R = F_R(\vec{b}, \vec{x})$, a.k.a. the robust SDs, can be formulated by combining eqs. (3.78) and (3.79) with eq. (1.4), as

$$\frac{\partial F_R}{\partial b_n} = \frac{\partial \mu_F}{\partial b_n} + \kappa \frac{\partial \sigma_F}{\partial b_n} \approxeq \frac{\partial a_0}{\partial b_n} + \kappa \frac{\sum_{i=1}^{N_{cut}} a_i \frac{\partial a_i}{\partial b_n}}{\sigma_F} \ , \quad n = 1, 2, ..., N \tag{3.81}$$

3.4.5 Cubature Integration

The numerical solution of the integrals in eqs. (3.76) and (3.80) is achieved through the Gauss Quadrature, a method a.k.a. *Cubature* when implemented in multi-dimensional integrals.

This method is defined in a similar manner with the definition used for simple integrals eq. (3.28), in Subection 3.3.5. For a function $h(\vec{x}) = W(\vec{x}) f(\vec{x}) \in \mathbb{R}$, its Cubature integration is formulated as

$$\int_{\vec{D}} h(\vec{x}) d\vec{x} = \int_{\vec{D}} W(\vec{x}) f(\vec{x}) dx = \sum_{j=1}^{N_{GQ}} \Omega_j \ f(\vec{z}_j) \tag{3.82}$$

where $\Omega_j = \prod_{l=1}^{M} \omega_{j_l}$ indicates the weight product corresponding to a certain Gauss Node combination $\vec{z}_j = [z_{j_1}, z_{j_2}, ..., z_{j_M}]$.

The numerical solution of the integral in eq. (3.82), requires all possible Gauss Node combinations in all dimensions present in the $\vec{z}_j$ vector. This means that the f function has to be computed for N_{GQ} number of combinations, given that $j \in [1, N_{GQ}]$, thus directly affecting the method's computational cost. The sum of all Gauss Nodes, for the solution of integral (3.82), is referred to as *Full Grid*.

In Subsection 3.3.5, it is stated that when implementing the GQ method for the computation of the niPCE coefficients a_i and their derivatives, the number of Gauss Nodes needed for the simple integrals is dependent on the chaos order k and is equal to $k+1$. Generalizing the aforementioned statement, for M-dimensional integrals the number of Gauss Nodes as well as the number of the F functions calls in eqs. (3.76) and (3.80) are equal to $(k+1)^M$. Therefore, the computational cost of the niPCE method using Full Grid GQ integration scales exponentially with the number M of uncertain variables that are taken into account. The exponential scaling of the number of nodes needed for the numerical solution of an integral, w.r.t. the number of its dimensions, is known as *curse of dimensionality.* This property of numerical integration has to be taken into account for problems with a large number of uncertain variables, given that a Full Grid integration can lead to a practically unfeasible implementation, due to its great computational and clock-time cost. The way the curse of dimensionality has an impact on the CPU cost of niPCE coefficients computation is displayed in Table 3.1.

	M					
k	1	2	3	4	5	6
0	1	1	1	1	1	1
1	2	4	8	16	32	64
2	3	9	27	81	243	729
3	4	16	64	256	1024	4096
4	5	25	125	625	3125	15625
5	6	36	216	1296	7776	46656

Table 3.1: *QoI function calls for the computation of niPCE coefficients through use of Full Grid GQ numerical integration, for different values of chaos order k and uncertain variables M. The scaling of the function call with the number of uncertain variables, a.k.a. the number of dimensions in the integral, is exponential, following the $(k+1)^M$ rule.*

Nevertheless, the use of Full Grid GQ integration, alongside with the employment of multi-dimensional canonical Hermite polynomials $\widetilde{\psi}$ for the solution of eqs. (3.76) and (3.80) yields

$$a_i = \sum_{j_1=1}^{k+1} \sum_{j_2=1}^{k+1} \cdots \sum_{j_M=1}^{k+1} \left(\prod_{l=1}^{M} \omega_{j_l} \right) F(\vec{b}, x_{j_1}, x_{j_2}, ..., x_{j_M}) \; \widetilde{\psi}_i(z_{j_1}, z_{j_2}, ..., z_{j_M}) \tag{3.83}$$

$$\frac{\partial a_i}{\partial b_n} = \sum_{j_1=1}^{k+1} \sum_{j_2=1}^{k+1} \cdots \sum_{j_M=1}^{k+1} \left(\prod_{l=1}^{M} \omega_{j_l} \right) \frac{\partial F(\vec{b}, x_{j_1}, x_{j_2}, ..., x_{j_M})}{\partial b_n} \widetilde{\psi}_i(z_{j_1}, z_{j_2}, ..., z_{j_M}) \quad (3.84)$$

where z_{j_l} the Gauss Nodes, computed as the roots of the $\widetilde{\psi}_{k+1}$ polynomials, ω_{j_l} their corresponding weights (more data in Appendix B) and z_{j_l} are defined as

$$x_{j_l} = \sigma_l \cdot z_{j_l} + \mu_l \ , \quad j_l = 1, 2, ..., k+1 \ \ and \ \ l = 1, 2, ..., M \quad (3.85)$$

To conclude, through Full Grid GHQ integration, the niPCE mean value of F assumes the form

$$\mu_F = a_0 = \sum_{j_1=1}^{k+1} \sum_{j_2=1}^{k+1} \cdots \sum_{j_M=1}^{k+1} \left(\prod_{l=1}^{M} \omega_{j_l} \right) F(\vec{b}, x_{j_1}, x_{j_2}, ..., x_{j_M}) \, \widetilde{\psi}_0(z_{j_1}, z_{j_2}, ..., z_{j_M}) \quad (3.86)$$

while its derivatives w.r.t. the design variables b_n becomes

$$\frac{\partial \mu_F}{\partial b_n} = \frac{\partial a_0}{\partial b_n} = \sum_{j_1=1}^{k+1} \sum_{j_2=1}^{k+1} \cdots \sum_{j_M=1}^{k+1} \left(\prod_{l=1}^{M} \omega_{j_l} \right) \frac{\partial F(\vec{b}, x_{j_1}, x_{j_2}, ..., x_{j_M})}{\partial b_n} \widetilde{\psi}_0(z_{j_1}, z_{j_2}, ..., z_{j_M}) \quad (3.87)$$

Meanwhile, the computation of the niPCE standard deviation of F, by makinguse of the Full Grid GHQ integration, yields

$$\sigma_F = \sqrt{\sum_{i=1}^{N_{cut}} a_i^2} =$$

$$= \sqrt{\sum_{i=1}^{N_{cut}} \left[\sum_{j_1=1}^{k+1} \sum_{j_2=1}^{k+1} \cdots \sum_{j_M=1}^{k+1} \left(\prod_{l=1}^{M} \omega_{j_l} \right) F(\vec{b}, x_{j_1}, x_{j_2}, ..., x_{j_M}) \, \widetilde{\psi}_i(z_{j_1}, z_{j_2}, ..., z_{j_M}) \right]^2} \quad (3.88)$$

and its derivatives w.r.t. the design variables are formulated as

$$\frac{\partial \sigma_F}{\partial b_n} = \frac{1}{\sigma_F} \sum_{i=1}^{N_{cut}} a_i \frac{\partial a_i}{\partial b_n} =$$

$$= \frac{1}{\sigma_F} \sum_{i=1}^{N_{cut}} \left[\sum_{j_1=1}^{k+1} \sum_{j_2=1}^{k+1} \cdots \sum_{j_M=1}^{k+1} \left(\prod_{l=1}^{M} \omega_{j_l} \right) F(\vec{b}, x_{j_1}, x_{j_2}, ..., x_{j_M}) \, \widetilde{\psi}_i(z_{j_1}, z_{j_2}, ..., z_{j_M}) \right] \cdot$$

$$\cdot \left[\sum_{j_1=1}^{k+1} \sum_{j_2=1}^{k+1} \cdots \sum_{j_M=1}^{k+1} \left(\prod_{l=1}^{M} \omega_{j_l} \right) \frac{\partial F(\vec{b}, x_{j_1}, x_{j_2}, ..., x_{j_M})}{\partial b_n} \widetilde{\psi}_i(z_{j_1}, z_{j_2}, ..., z_{j_M}) \right] \quad (3.89)$$

where the number of niPCE coefficients is given by the expression

$$N_{cut} = \left(\begin{array}{c} k+M \\ k \end{array} \right) = \frac{M(M+1)(M+2)\cdots(M+k)}{k!} \tag{3.90}$$

3.4.6 Smolyak Sparse Grid Integration

In this Subsection, a less costly method under certain circumstances, is introduced, for the numerical computation of the integrals, present in niPCE coefficients and their derivatives ((3.76) and (3.80), respectively).

Up until now, these integrals where computed by employing the GHQ method, while using all $(k+1)^M$ Gauss Nodes for the integration. These nodes are the roots z_{j_l} of the multi-dimensional Hermite polynomials $\psi(\vec{z}_j)$, which are products of 1D Hermite polynomials $He_{j_l}(z_{j_l})$, of degree j_l. The amount of different combinations of degrees, and therefore roots, is equal to $(k+1)^M$, given that:

$$j_l = 1, 2, ..., k+1 \ \ \forall \ \ l = 1, 2..., M$$

This set of nodes used for the GQ method is referred to as *Full Grid* GQ integration. According to Table 3.1, the curse of dimensionality, makes the implementation of Full Grid integration practically impossible for problems with a large number of uncertain variables ($M \geq 4$).

The *Smolyak Sparse Grids* [33] first proposed by Smolyak in 1963, so as to circumvent the curse of dimensionality in GQ integration, by requiring a smaller number of Gauss Nodes for the computation of the integral. Consequently, when applied in UQ or RDO problems, less calls to the QoI function F are needed (an action synonymous with the costly solution of either only the primal problem, if UQ is preformed, or also of the adjoint problem, if RDO takes place), leading to a reduced computational cost. The real advantage of the Smolyak Grids lies, especially in problems with a large number of uncertain variables M, given that, with this method, the integrals can be computed with a slightly reduced accuracy, but, on the other hand, with a far lesser number of Gauss Nodes and therefore for an inferior cost.

Nevertheless, the Smolyak Grids are not introduced in this work, so as to totally replace the Full Grid GQ integration. Their use is, mainly, to complete the GQ integration method as a whole, by switching between one method, when the other yields an unfeasible computational cost.

That said, the process needed for the creation of Smolyak Sparse grid for a specific M-dimensional integral, is the following:

1. For the selected chaos order k (the higher its value, the higher the accuracy along with the cost), the possible combinations of polynomial degrees j_l, also

culminating to the number N_{SM} of Gauss Nodes of the grid, have to fullfill the criterion

$$M \leq \sum_{l=1}^{M} j_l \leq M + k \tag{3.91}$$

The addition of definitions $|j| = \sum_{l=1}^{M} j_l$ and $M + K = q$ to (3.91), yield

$$M \leq |j| \leq q \tag{3.92}$$

According to criterion (3.92), the span of each of the degrees j_l is defined as

$$\left\{ \begin{array}{c} j_1 = 1, ..., d_1 \\ j_2 = 1, ..., d_2 \\ \cdots \\ \cdots \\ \cdots \\ j_M = 1, ..., d_M \end{array} \right\} \tag{3.93}$$

where the product of maximum degrees yields the total number of nodes $N_{SM} = \prod_{l=1}^{M} d_l = d_1\, d_2 \cdots d_M$.

2. The Full Grid GQ weights ω_{j_l} of the orthogonal polynomials defined in (3.82) of Subsection 3.4.5, are utilized to produce the Smolyak redefined weights $W_n \;\forall\; n = 1, 2, ..., N_{SM}$, as follows

$$\Omega_n = (-1)^{q-|j|} \begin{pmatrix} M-1 \\ q-|j| \end{pmatrix} \prod_{l=1}^{M} \omega_{j_l} \tag{3.94}$$

3. The integral is computed with a Sparse Grid of Gauss Nodes as

$$\begin{gathered} \int_{\vec{D}} f(\vec{x})\, d\vec{x} = \sum_{n=1}^{N_{SM}} \Omega_n f(\vec{z}_n) = \\ = \sum_{j_1=1}^{d_1} \sum_{j_2=1}^{d_2} \cdots \sum_{j_M=1}^{d_M} \left[(-1)^{q-|j|} \begin{pmatrix} M-1 \\ q-|j| \end{pmatrix} \prod_{l=1}^{M} \omega_{j_l} \right] f(z_{j_1}, z_{j_2}, ..., z_{j_M}) \end{gathered} \tag{3.95}$$

According to step 2, the amount of nodes used both in Full and Sparse grids are displayed in Table 3.2, on order to fully understand the capabilities and shortcomings of the two methods. It is clear, that the Full Grid has a smaller implementation costand is therefore a better choice for $M = 1, 2, 3 \;\forall\; k$. For $M = 4$ the cost of the two methods is in the same order of magnitude, giving the ability to both methods interchangeably. On the other hand, for a greater number dimensions, the cost of the Full Grid integration, for any value of $k > 1$, is a least an order of magnitude greater than the respective cost of Sparse grid integration.

Full Grid / Smolyak Sparse Grid						
	M					
k	1	2	3	4	5	6
0	1/1	1/1	1/1	1/1	1/1	1/1
1	2/3	4/5	8/7	16/9	32/11	64/13
2	3/5	9/13	27/25	81/41	243/61	729/85
3	4/9	16/29	64/69	256/137	1024/241	4096/389
4	5/17	25/65	125/177	625/401	3125/801	15625/1457
5	6/33	36/145	216/441	1296/1105	7776/2433	46656/4865

Table 3.2: *Gauss Nodes for the computation of niPCE coefficients through use of Full Grid and Smolyak Sparse grid GQ numerical integration, for different values of chaos order k and uncertain variables M.*

Finally, the implementation of Smolyak Sparse grid GHQ integration, for the computation of the niPCE coefficients a_i from (3.76), yields

$$a_i = \sum_{j_1=1}^{d_1} \cdots \sum_{j_M=1}^{d_M} \left[(-1)^{q-|j|} \begin{pmatrix} M-1 \\ q-|j| \end{pmatrix} \prod_{l=1}^{M} \omega_{j_l} \right] F(\vec{b}, x_{j_1}, ..., x_{j_M})\, \widetilde{\psi}_i(z_{j_1}, ..., z_{j_M}) \tag{3.96}$$

while for the derivatives of a_i w.r.t. the design variables b_n

$$\frac{\partial a_i}{\partial b_n} = \sum_{j_1=1}^{d_1} \cdots \sum_{j_M=1}^{d_M} \left[(-1)^{q-|j|} \begin{pmatrix} M-1 \\ q-|j| \end{pmatrix} \prod_{l=1}^{M} \omega_{j_l} \right] \frac{\partial F(\vec{b}, x_{j_1}, ..., x_{j_M})}{\partial b_n}\, \widetilde{\psi}_i(z_{j_1}, ..., z_{j_M}) \tag{3.97}$$

where $\vec{z}_j = (z_{j_1}, z_{j_2}, ..., z_{j_M})$ the Gauss Nodes, $q = M + k$, while

$$x_{j_l} = \sigma_l \cdot z_{j_l} + \mu_l \;\; and \;\; |j| = \sum_{l=1}^{M} j_l$$

Finally, from eqs. (3.96) and (3.97), the mean value and standard deviation of F, as well as their derivatives, are formulated accordingly, using the same strategy as the one followed in Subsection 3.4.5.

Chapter 4

Modeling of Manufacturing Imperfections

The mathematical formulation of the stochastic processes used to model the inevitable shape imperfections generated on mechanical parts during the manufacturing procedure, is presented in this chapter.

4.1 Karhunen-Loève Transform

During the manufacturing of mechanical parts, the occurrence of a certain differentiation between the shapes of the designed and the finished product is generally expected. The stochasticity of the manufacturing process, such as the quality degradation of cutting tools, is largely responsible for such occurrences. This phenomena appears, for example, during the large-scale production of lifting bodies such as aircraft wings, hydrofoils or turbomachinery blades. The shape differentiation of the finished product when compared with the original design must respect the tolerances specified by the manufacturer, otherwise the product is deemed a failure and be disposed. Yet, even if the imperfections do not overstep their specified tolerance boundaries, these have a non-trivial impact on the aerodynamic performance of the manufactured lifting body.

In this diploma thesis, the mathematical tool used to model these imperfections is known as the *Karhunen-Loève Transform* (KLT) [34, 35, 36, 37, 38]. The KLT comprises a stochastic process used over a finite space or time span.

4.1.1 Karhunen-Loève Expansion

The *Karhunen-Loève Expansion* (KLE) is used to compute stochastic perturbations on the surface of an imperfect part. This perturbation $E(s)$ is expanded into an orthogonal set of deterministic functions $f_n(s)$ according to the following formula

$$E(s) = \sum_{n=1}^{\infty} \sqrt{\lambda_n} c_n f_n(s) \tag{4.1}$$

where s indicates the dimensional curvilinear coordinate of the shape in question, c_n indicates a set of random variables to be determined and λ_n indicates some constants, yet to be defined.

In eq. (4.1), the probabilistic $\sqrt{\lambda_n} c_n$ and deterministic $f_n(s)$ parts of the expansion, must be considered separately.

In order to recreate a randomly generated imperfect shape realization $X(s, \vec{c})$, the perturbation expansion $w(s)$ is superimposed on the nominal shape $\bar{X}(s)$ in the direction of the surface normal vector. Specifically in the case of 2D aerodynamic bodies, such as airfoils, the perturbation is superimposed on the mean nominal airfoil geometry $\bar{X}(s)$ using the following equation

$$X(s, \vec{c}) = \bar{X}(s) + E(s)\vec{n}(s) = \bar{X}(s) + \sum_{n=1}^{\infty} \sqrt{\lambda_n} c_n f_n(s)\vec{n}(s) \tag{4.2}$$

where $s \in [0, s_{max}]$ and s_{max} is defined as the total length of the contour of the shape under consideration (in this case an airfoil), whereas $\vec{n}(s)$ indicates the surface unit normal vector.

4.1.2 Covariance Kernel

It is essential to explain and define the statistical moments that describe the stochastic perturbation process described in eq. (4.1). Evidently, the process has zero mean due to the fact the mean nominal geometry is included in eq. (4.2). However, the variance off all the possible of the expansion $E(s)$, from eq. (4.1), can be modeled through the covariance kernel or covariance function.

By definition of the covariance function is bounded, symmetric and positive definite [35]. Thus, it has the spectral decomposition [37, 38]

$$C(s_1, s_2) = \sum_{n=1}^{\infty} \lambda_n f_n(s_1) f_n(s_2) \tag{4.3}$$

The type of covariance function used in the KLT process greatly impacts the eigenvalues and the mathematical formulation of the eigenfunctions used to describe the stochastic perturbations of eq. (4.2). In this thesis, the covariance kernel used to generate the imperfections of 2D shapes is exponential [34, 35, 36, 37] and is represented by the formula

$$C(s_1, s_2) = \sigma^2 e^{-\frac{|s_1 - s_2|}{l}} \tag{4.4}$$

where $s_1,\ s_2 \in [0, s_{max}]$ represent positions on the airfoil profile and l indicates the correlation length, which in return, describes the frequency of the perturbations. The σ parameter is known as the standard deviation of the perturbations and is a user-defined dimensional parameter that serves to scale the absolute magnitude of the perturbations. The value of this parameter has no physical significance, as the scaling of the perturbations can be achieved also through the range or the statistical moments that define the distribution of the uncertain variables c_n. For this thesis, given that the standard deviation of the imperfections is not defined, its value is assumed to be $\sigma = 1$ and can, therefore, be practically omitted.

The eigenvalues λ_n and the eigenfunctions f_n used in the KLE model eq. (4.2), are obtained by solving the following integral equation

$$\int_D C(s_1, s_2) f_n(s_2) ds_2 = \lambda_n f_n(s_1) \tag{4.5}$$

where D indicates the domain where the coordinates $s_1,\ s_2$ are defined.

Due to the fact that the covariance kernel has the properties mentioned above, the eigenvalues and eigenfunctions that consist the solution of the integral eq. (4.5) have the following properties [35]:

- The set of eigenfunctions is orthogonal and complete. The eigenfunctions can be normalized in the sense

  $$\int_D f_n(s) f_m(s) ds = \delta_n^m \tag{4.6}$$

  where δ_n^m is the Kronecker delta.

- Each eigenvalue corresponds to a finite number of linearly independent eigenfunctions, at most. In the present case of exponential kernels, each eigenvalue corresponds to a single eigenfunction.

- There are at most a countably infinite number of eigenvalues.

- All eigenvalues are positive real numbers.

- Every covariance kernel admits to the uniformly convergent expansion of eq. (4.3)

In the case of the exponential kernel (4.4) and of a non-symmetric interval $D = [a, b]\,,\ a, b \in \mathbb{R}$ for which $s_1,\ s_2 \in D$, the integral eq. (4.5) takes the following form

$$\int_a^b \sigma^2 e^{-\frac{|s_1-s_2|}{l}} f_n(s_2) ds_2 = \lambda_n f_n(s_1) \tag{4.7}$$

Since the correlation length l, meaning the frequency of the manufacturing imperfections, is a value difficult to define even by the manufacturer, for the purpose of shape imperfections modeling, it is often assumed that $l = |a - b| = b - a$. Especially in the case of aerodynamic bodies, for which the limits of the interval D are often defined as $a = 0$ and $b = s_{max}$, the correlation length is, therefore, defined as $l = s_{max}$ (with accordance to the previous assumption).

4.1.3 Solution of the Integral Equation

At this point, it would be useful to specify the known and unknown data of the KLT process. Whenever a new nominal geometry is introduced, in order to create an imperfect realisation of it, the contour s_{max} of the nominal geometry is computed. Thus, the range of the variables s_1, s_2 and, consequently, the interval $D = [a, b]$ of the integral eq. (4.7) can be defined. In addition, if no more empirical data are available, the correlation length l and the standard deviation σ can be defined, respectively, in consonance with the assumptions mentioned in Subsection 4.1.2. Finally, the vector of uncertain variables $\vec{c}$ must be designated either by a random number generator or by a specific process such as niPCE if, for example, UQ must be performed.

Once all the above parameters are known, we can proceed to the analytical solution of eq. (4.7), so as to calculate the needed eigenvalues and eigenfunctions. Equation (4.7) can be written in the following form

$$\int_a^{s_1} \sigma^2 e^{-\frac{s_1-s_2}{l}} f_n(s_2) ds_2 + \int_{s_1}^b \sigma^2 e^{\frac{s_1-s_2}{l}} f_n(s_2) ds_2 = \lambda_n f_n(s_1) \tag{4.8}$$

Differentiating eq. (4.8) w.r.t. $s_1 \in [a, b]$ and applying the Leibniz integral rule gives

$$\begin{aligned} & -\frac{\sigma^2}{l}\int_a^{s_1} e^{\frac{s_2-s_1}{l}} f_n(s_2) ds_2 + \frac{\partial s_1}{\partial s_1}\sigma^2 f(s_1) + \frac{\sigma^2}{l}\int_{s_1}^b e^{\frac{s_1-s_2}{l}} f_n(s_2) ds_2 - \frac{\partial s_1}{\partial s_1}\sigma^2 f(s_1) = \lambda_n \frac{df_n(s_1)}{ds_1} \\ & \Rightarrow\ -\int_a^{s_1} e^{\frac{s_2-s_1}{l}} f_n(s_2) ds_2 + \int_{s_1}^b e^{\frac{s_1-s_2}{l}} f_n(s_2) ds_2 = \frac{l\lambda_n}{\sigma^2}\frac{df_n(s_1)}{ds_1} \end{aligned} \tag{4.9}$$

Differentiating eq. (4.9) again w.r.t. s_1 and using again the Leibniz rule yields

$$\frac{1}{l}\int_a^{s_1} e^{\frac{s_2-s_1}{l}} f_n(s_2)ds_2 - \frac{\partial s_1}{\partial s_1} f(s_1) + \frac{1}{l}\int_{s_1}^{b} e^{\frac{s_1-s_2}{l}} f_n(s_2)ds_2 - \frac{\partial s_1}{\partial s_1} f(s_1) = \frac{l\lambda_n}{\sigma^2}\frac{df_n^2(s_1)}{ds_1^2}$$
$$\Rightarrow \; -\int_a^{s_1} e^{\frac{s_2-s_1}{l}} f_n(s_2)ds_2 + \int_{s_1}^{b} e^{\frac{s_1-s_2}{l}} f_n(s_2)ds_2 - 2lf(s_1) = \frac{l^2\lambda_n}{\sigma^2}\frac{df_n^2(s_1)}{ds_1^2} \tag{4.10}$$

After rearranging and replacing eq. (4.8) into eq. (4.10), the following differential equation is obtained

$$\lambda_n \frac{d^2 f_n(s_1)}{ds_1^2} = (\frac{-2\sigma^2 l + \lambda_n}{l^2}) f_n(s_1) \tag{4.11}$$

By defining the new variable ω_n as

$$\omega_n = \frac{2\sigma^2 l - \lambda_n}{l^2\lambda_n} > 0 \tag{4.12}$$

and, hence, the eigenvalues [37] can be calculated by

$$\lambda_n = \sigma^2 \frac{2l}{1+(\omega_n l)^2} > 0 \tag{4.13}$$

By setting $s_1 = s$, eq. (4.11) assumes the following form

$$\frac{d^2 f_n(s)}{ds^2} + \omega_n^2 f_n(s) = 0\;, \quad a \leq s \leq b \tag{4.14}$$

To find the boundary conditions of the differential eqs. (4.14), (4.8) and (4.9) are evaluated at the integral boundaries, at $s = a$ and $s = b$. Therefore, the boundary conditions become

$$\begin{cases} f_n(a) - l\frac{df_n(a)}{ds} = 0 \\ \\ f_n(b) + l\frac{df_n(b)}{ds} = 0 \end{cases} \tag{4.15}$$

Furthermore, eq. (4.14) has four distinct solution forms [36]:

1. $\lambda_n = 0$
2. $0 < \lambda_n < 2\sigma^2 l$
3. $\lambda_n = 2\sigma^2 l$
4. $\lambda_n > 2\sigma^2 l$

For case (1), if $\lambda_n = 0$ then, according to eq. (4.12), ω_n is ill defined, due to division by zero.

For case (2), if $0 < \lambda_n < 2\sigma^2 l$ then

$$2\sigma l - \lambda_n > 0 \quad \Rightarrow \quad \omega_n = \frac{2\sigma^2 l - \lambda_n}{l^2 \lambda_n} > 0 \tag{4.16}$$

which respects the definition of ω_n, according to eq. (4.12).

For case (3), if $\lambda_n = 2\sigma^2 l$ then $\omega_n = 0$ which violates the definition of ω_n, according to eq. (4.12).

For case (4), if $\lambda_n > 2\sigma^2 l > 0$ then

$$2\sigma^2 l - \lambda_n < 0 \quad \Rightarrow \quad \omega_n = \frac{2\sigma^2 l - \lambda_n}{l^2 \lambda_n} < 0 \tag{4.17}$$

which again violates the definition of ω_n, according to eq. (4.12).

Thus, integral eq. (4.7) cannot be satisfied for the cases (1), (3) and (4).

For $0 < \lambda_n < 2\sigma^2 l$, it is assumed that the solution can be given by

$$f_n(s) = c_1 \cos\left[\omega_n \left(s - \frac{a+b}{2}\right)\right] + c_2 \sin\left[\omega_n \left(s - \frac{a+b}{2}\right)\right] \tag{4.18}$$

where $c_1,\ c_2 \in \mathbb{R}$ to be defined.

By substituting the eq. (4.18) into the boundary conditions (4.15), the following equations are formulated

$$\begin{cases} c_1[1 - \omega_n l \tan(\omega_n \tfrac{b-a}{2})] - c_2[\tan(\omega_n \tfrac{b-a}{2}) + \omega_n l] = 0 \\ \\ c_1[1 - \omega_n l \tan(\omega_n \tfrac{b-a}{2})] + c_2[\tan(\omega_n \tfrac{b-a}{2}) + \omega_n l] = 0 \end{cases} \tag{4.19}$$

Non-trivial solutions can be achieved only when the determinant J of the homogeneous system (4.19) is equal to zero [35]. Meaning that, if $J \neq 0$, then the inevitable solution of eqs. (4.19) is $c_1 = c_2 = 0$ Thus, J takes the following form

$$\begin{aligned} J &= 2[\omega_n l + (1 - (\omega_n l)^2) \tan(\omega_n \frac{b-a}{2}) - \omega_n l \tan^2(\omega_n \frac{b-a}{2})] \ \Rightarrow \\ J &= 2[\omega_n l + \tan(\omega_n \frac{b-a}{2})][1 - \omega_n l \tan(\omega_n \frac{b-a}{2})] = 0 \end{aligned} \tag{4.20}$$

By setting the determinant of eq. (4.20), to zero and after rearranging, the following transcendental equations [34] are derived

$$\begin{cases} \omega_n l + \tan(\omega_n \frac{b-a}{2}) = 0 \\ \\ 1 - \omega_n l \tan(\omega_n \frac{b-a}{2}) = 0 \end{cases} \tag{4.21}$$

For n being an even number, the value of $\omega_n > 0$ is given by the ordered solution to the transcendental equation

$$\omega_n l + \tan(\omega_n \frac{b-a}{2}) = 0 \tag{4.22}$$

and for n being an odd number, the value of $\omega_n > 0$ is given by solving the transcendental equation

$$1 - \omega_n l \tan(\omega_n \frac{b-a}{2}) = 0 \tag{4.23}$$

At this point, the first 20 eigenvalues λ_n are computed through the solution of the transcendental equations (4.22),(4.23) by making use of eq. (4.13) and are plotted in Figure 4.1 for different values of input parameters a, b, σ, l.

So, for n even

$$c_2 = \frac{\omega_n l + \tan(\omega_n \frac{b-a}{2})}{1 - \omega_n l \tan(\omega_n \frac{b-a}{2})} c_1 = 0 \ \ and \ \ f_n(s) = c_1 \cos\left(\omega_n (s - \frac{a+b}{2})\right) \tag{4.24}$$

Also, for n odd

$$c_1 = \frac{1 - \omega_n l \tan(\omega_n \frac{b-a}{2})}{\omega_n l + \tan(\omega_n \frac{b-a}{2})} c_2 = 0 \ \ and \ \ f_n(s) = c_2 \sin\left(\omega_n (s - \frac{a+b}{2})\right) \tag{4.25}$$

Furthermore, c_1, c_2 can be computed using the orthogonality eq. (4.6) of the eigenfunctions

$$For\ n\ even:\ c_1^2 \int_a^b \cos^2\left(\omega_n (s - \frac{a+b}{2})\right) ds = 1 \ \Rightarrow \ c_1 = \frac{1}{\sqrt{\frac{b-a}{2} + \frac{\sin(\omega_n (b-a))}{2\omega_n}}} \tag{4.26}$$

$$For\ n\ odd:\ c_2^2 \int_a^b \sin^2\left(\omega_n (s - \frac{a+b}{2})\right) ds = 1 \ \Rightarrow \ c_2 = \frac{1}{\sqrt{\frac{b-a}{2} - \frac{\sin(\omega_n (b-a))}{2\omega_n}}} \tag{4.27}$$

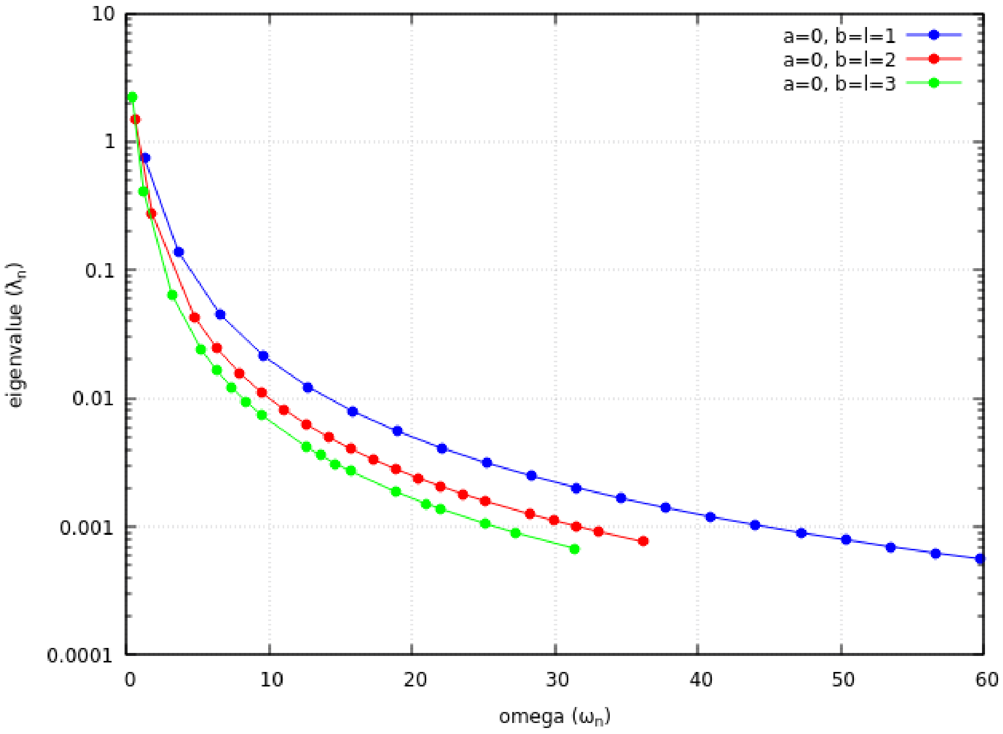


Figure 4.1: *KLT eigenvalue decay for the exponential kernel of eq.* (4.4) *and, also, for* $\sigma = 1$, $a = 0$ *and* $b = l = 1,\ 2,\ 3$.

To summarize, the eigenfunctions [34] are determined by the following formula

$$f_n(s) = \begin{cases} \dfrac{\cos\left(\omega_n(s-\frac{a+b}{2})\right)}{\sqrt{\frac{b-a}{2}+\frac{\sin(\omega_n(b-a))}{2\omega_n}}}\ , & for\ n\ even \\ \\ \dfrac{\sin\left(\omega_n(s-\frac{a+b}{2})\right)}{\sqrt{\frac{b-a}{2}-\frac{\sin(\omega_n(b-a))}{2\omega_n}}}\ , & for\ n\ odd \end{cases} \tag{4.28}$$

By making use of eq. (4.28), the first six eigenfunctions are computed and plotted in Figure 4.2, for $s \in [0,\ 1]$.

4.1.4 KLT Algorithmic Formulation

Now that the mathematical formulation the Karhunen-Loève Transform has been fully defined, it is possible to describe the algorithmic steps needed to compute a random imperfect realization of a 2D geometry.

First and foremost, the nominal 2D geometry must be given in the form of either a number of discrete coordinate points or a function of s. Either way, this input defines the variable $\bar{X}(s)$ in eq. (4.2). Subsequently, the contour s_{max} of the nominal

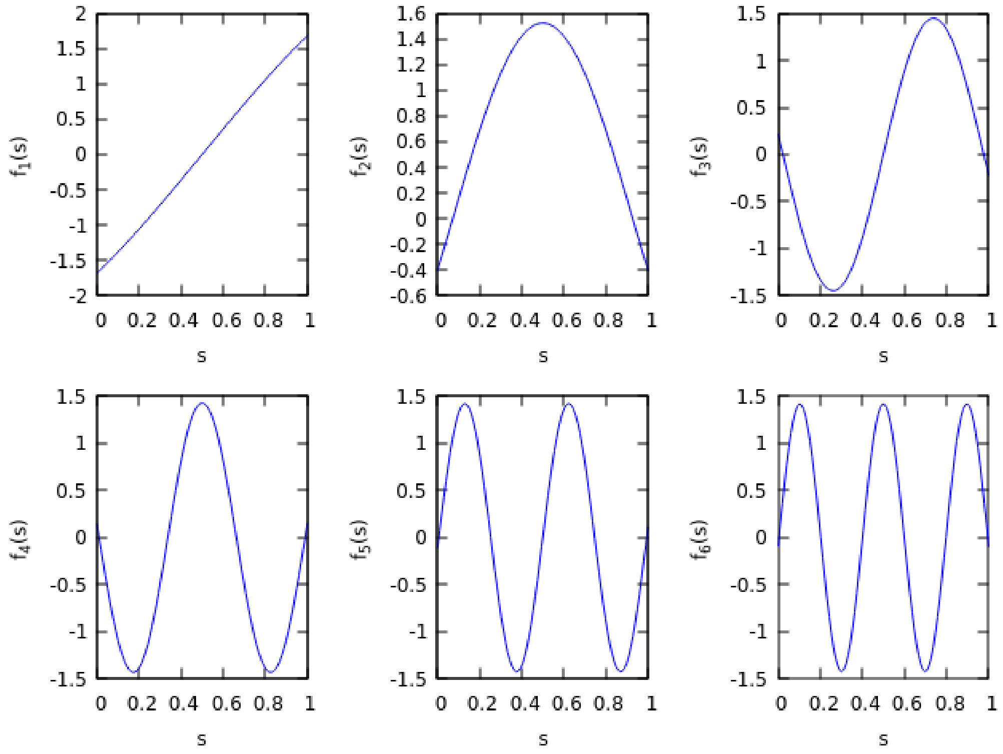


Figure 4.2: *First six KLT eigenfunctions for the exponential kernel* (4.4) *and, also, for* $s \in [a,\, b] = [0,\, 1]$, $b = l$ *and* $\sigma = 1$.

geometry and, thus, the interval $D = [0,\, s_{max}]$ of the integral eq. (4.7) is computed, under the assumption that the covariance kernel as the one in (4.4), is employed. If no more data is shared by the designer or the manufacturer of the geometry, then the correlation length can be assumed $l = s_{max}$ (according to the assumptions made in Subsection 4.1.2).

Once all this is defined, the algorithm, implementing the KLT to generate an imperfect geometry, can be described by the following steps:

1. Analytical or numerical solving of the transcendental equations (4.21) for a vast range of the variable $\omega_n > 0$, in order to find a finite number of eigenvalues λ_n, by making use of eq. (4.13).

2. Specifying the number of the first eigenvalues and their respective eigenfunctions that is included in the KLE of eq. (4.1), according to either a predefined criterion or via other *ad-hoc* methods.

3. Computation of the surface unit normal vectors $\vec{n}$ via an analytical or a numerical method, depending on the description method for the nominal geometry.

4. Randomizing the values of the KLT uncertain variable vector $\vec{c}$, by means of

a random number generator. The dimension of $\vec{c}$ was defined in *Step 2* and is equal to the number of eigenvalues included in the KLE.

5. Computation of the random geometry perturbation through the KLE formulas and superimposition of this perturbation on the nominal geometry, according to eq. (4.29)

Steps 4 and 5 can be repeated independently within a loop, so as to compute more than one random imperfect 2D geometries.

4.2 KLT Applications

4.2.1 Flat Plate

First, the KLT process is applied on a flat plate, for simplicity reasons, in order to understand the basic functions of the process on a plain 2D geometry. To put it simply, the KLE perturbations are superimposed on the a straight line, in the direction perpendicular to it, hence, creating a number of oscillating continuous lines.

The parameters of the applications are defined simply, by using all assumptions mentioned in Section 4.1. Therefore, for a flat plate of 1 *meter* in length, with no curvature and parallel to the x-axis of the Cartesian coordinate system the KLT parameters are set as $s \equiv x \in [a,\ b] = [0,\ 1]\ m,\ l = b = 1\,m$ and $\sigma = 1\,m$. The input file, containing the plate's equally spaced coordinates, is comprised by 100 coordinate points, meaning that the discretization step of the plate is equal to $\Delta s = \Delta x = 0.01\,m$.

For the implementation of KLT on the flat plate, an algorithm that can apply the KLE on a 2D geometry is developed in C++. According to the transform's properties presented in Subsection 4.1.2 the solution of the integral eq. (4.5) can produce at most an infinite number of eigenvalues and their respective eigenfunctions. It is self-evident that an algorithm must use a finite number of eigenvalues in order to compute the KLE of the imperfections (4.1). Thus, the question of how many modes should be used during the KLT process has arisen. Given that, an acceptable number $M \in \mathbb{N}$ (as defined in eq. (4.29)) of eigenvalues and eigenfunctions to be included in the KLE is not pre-determined, a parametric analysis is w.r.t. M taking place. The value of the cut-off point M, simultaneously, determines the dimension of the vector of uncertain variables $c_n,\ n \in [0, M]$ needed for the process. For this application and the next (see in Subsection 4.2.2), the values of the uncertain variables c_n are produced by the pseudo random number generator *rand()*, of C++.

Here, eq. (4.2) is rewritten with defined cut-off point M:

$$X(s, \vec{c}) = \bar{X}(s) + \sum_{n=1}^{M} \sqrt{\lambda_n} c_n f_n(s) \vec{n}(s) \tag{4.29}$$

The purpose of the parametric analysis is to better understand the effect that the value of M has on the shape of the imperfect realizations of the nominal geometry. Seven imperfect realization of the flat plate are generated, each time for 5, 10 and 20 modes included in the KLE, thus, M is assuming the values 5, 10, and 20, accordingly.

The maximum magnitude of the perturbations in flat plate KLT imperfect renderings is purposely set as 50% of the initial plate's length, for display reason, given that the complexity of the perturbation's osculation is of interest here and not its magnitude.

The imperfect plates computed by including the 5, 10 and 20 first eigenvalues and their corresponding eigenfunctions in the KLE modeling the perturbations can be found in Figures 4.3, 4.4 and 4.5, respectively.

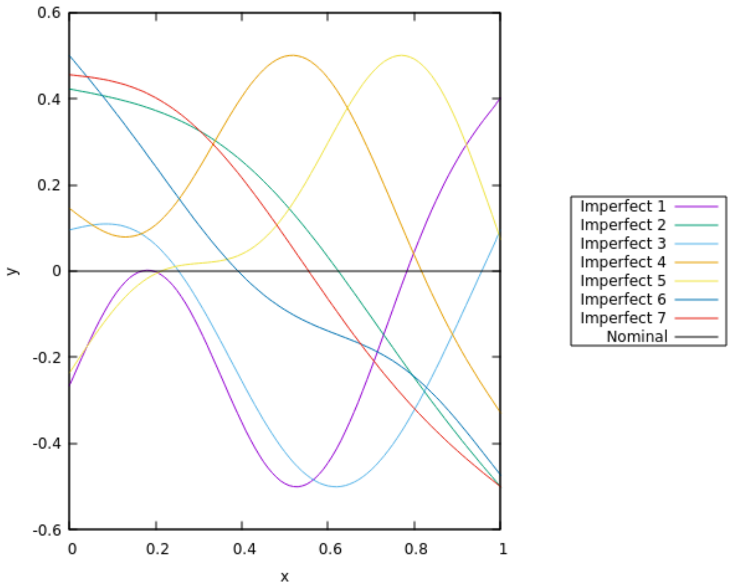


Figure 4.3: *The nominal flat plate (black) and seven imperfect realizations of it (other colors), generated through KLT by including the first 5 modes ($M = 5$) and plotted in scale.*

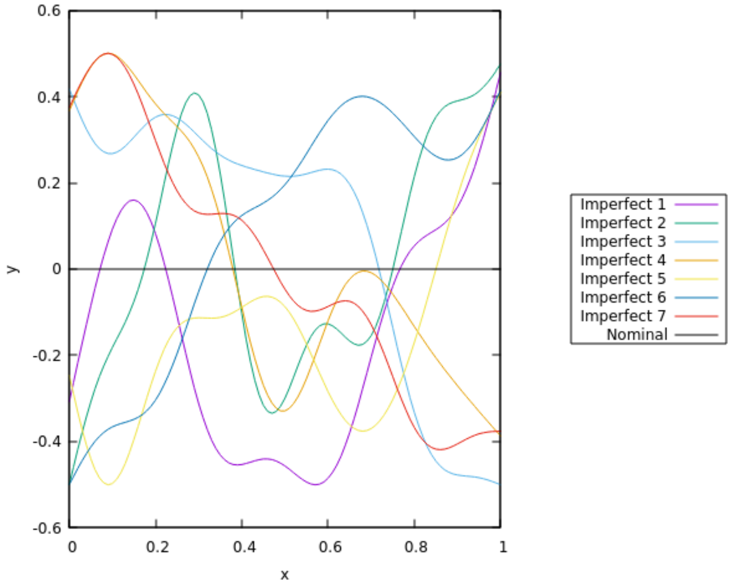


Figure 4.4: *The nominal flat plate (black) and seven imperfect realizations of it (other colors), generated through KLT by including the first 10 modes ($M = 10$) and plotted in scale.*

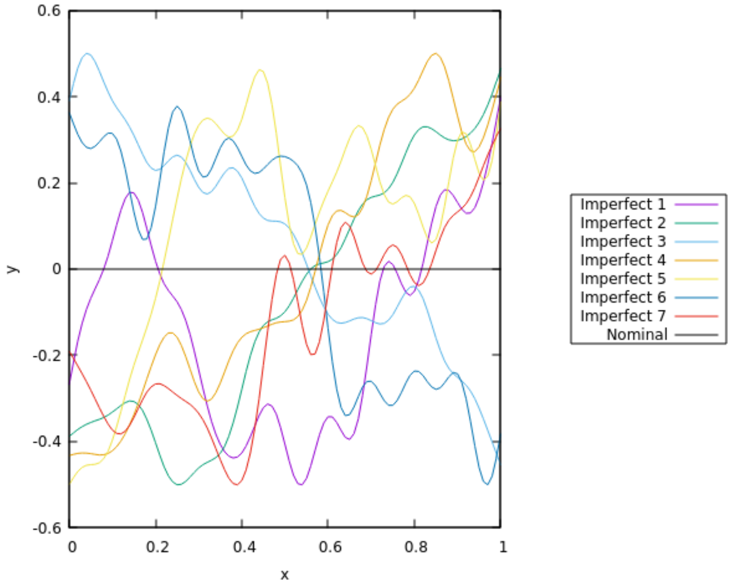


Figure 4.5: *The nominal flat plate (black) and seven imperfect realizations of it (other colors), generated through KLT by including the first 20 modes ($M = 20$) and plotted in scale.*

As expected, the random imperfect realizations of the flat plate get more complex as more modes are included in the KLE. This observation contradicts the fact of eigenvalue decay, as stated previously in Figure 4.1. To be more precise and to further strengthen this point, for this application, the sum of the first 5, 10 and 20 eigenvalues are presented as a percentage of the sum of all calculated eigenvalues, in Table 4.1. Yet, as we progress from Figures 4.3 to 4.5, it is clear that the increase of M causes the imperfect lines to be become more oscillated, given that more and more high frequency eigenfunctions (Figure 4.2) are added to the expansion that simulates the imperfections.

Number of Modes (M)	$\sum_{n=1}^{M} \lambda_n / \sum_n \lambda_n$
5	**98.15** %
10	**99.25** %
20	**99.68** %

Table 4.1: *Sum of the first KLT eigenvalues of the flat plate, for* $M = 5,\ 10,\ 20$.

4.2.2 NACA 0012 Airfoil

The KLT process is, now, applied on a 2D aerodynamic body, such as an airfoil, in order to showcase the capabilities of the process in recreating a number of stochastic imperfect realizations of the said body, thus, simulating the final manufactured product. The KLT is implemented on the NACA 0012 airfoil and, thus, the parameters of the process must be defined.

The domain of the integral eq. (4.5) is defined as $D = [0, s_{max}]$, where $s_{max} = 2.03918\ m$ represents the perimeter of a NACA 0012 airfoil with a unit chord. The correlation length of the covariance kernel is set $l = s_{max}$. In addition, the number d of eigenvalues and corresponding eigenfunctions that are used in the KLE to form the stochastic perturbations must be defined. After some attempts on different airfoils it can be assumed that only the first five eigenvalues and eigenvalues might be sufficient, given that the following criterion is satisfied with 5 modes, in most cases.

$$\frac{\sum_{n=1}^{M} \lambda_n}{\sum_n \lambda_n} \geq 0.96 \tag{4.30}$$

Specifically, for the NACA 0012 airfoil, the sum of the first five eigenvalues consists the 97.26 % of the sum of all calculated eigenvalues. Consequently, to define each imperfect airfoil, five uncertain variables c_n are needed.

Therefore, for this application, eq. (4.29) assumes the following form

$$X(s, \vec{c}) = \bar{X}(s) + \sum_{n=1}^{M=5} \sqrt{\lambda_n} c_n f_n(s) \vec{n}(s) \tag{4.31}$$

For the implementation of KLT on a specified airfoil, the same software as the one mentioned in Subsection 4.2.1 is used. A similar algorithm is coupled with the niPCE in-house code developed by the PCOpt/NTUA in FORTRAN and used in later stages of this diploma thesis, so as to perform UQ [3] and aerodynamic RDO w.r.t. manufacturing imperfections. The input file, that describes the NACA 0012 profile, contains 200 coordinate points.

For this application only, the standard deviation of the perturbations σ of the covariance kernel (4.4) is replaced by a different parameter that serves to define the maximum absolute magnitude of the KLE perturbation (4.1), so as to simulate the equivalent manufacturing tolerance of the lifting body. Thus, this new parameter renders parameter σ obsolete, given its only actual purpose is to scale the KLE perturbations (4.1). Nevertheless, this post-processing is bypassed, in later use of this software (for the purposes of UQ and RDO), considering that the magnitude of the perturbations are dictated by the statistical moments defining the uncertain variables. For this application, the maximum imperfection magnitude is assumed, arbitrarily, to be the 0.3 % of the airfoil's chord length.

Additionally, a Hann-like weighted cosine function, such as those used in signal processing, is optionally utilized to damp the KLE perturbation close to the trailing edge of the airfoil. In this application, this post-processing is required to keep the trailing edge coordinates unchanged and always maintain a sharp trailing edge. The formula used for the computation of the weighted *Hanning window* coefficient C_{Hann} can be described in eq. (4.32)

$$C_{Hann}(s) = \begin{cases} 1-\cos(\frac{\pi}{2\,p}\frac{s}{s_{max}}) \ , \quad for \ \frac{s}{s_{max}} < \ p \\ \\ 1 \ , \quad for \ p \leqslant \frac{s}{s_{max}} \leqslant \ (1-p) \\ \\ 1-\cos(\frac{\pi}{2\,p}\frac{1-s}{s_{max}}) \ , \quad for \ \frac{s}{s_{max}} > \ 1-p \end{cases} \tag{4.32}$$

where $p \in [0,1]$ indicates the user-defined percentage of the contour s_{max} of the airfoil. The coefficient produced from eq. (4.32) is plotted, for different values of p, in Figure 4.6.

It is assumed that for this particular airfoil, the damping of the perturbations should take place, for each side of the airfoil, at the 8 % of it's contour (p = 8%). Thus, the final form of the KLE formula used in this algorithm is given by

$$X(s,\vec{c}) = \bar{X}(s) + C_{Hann}(s) \sum_{n=1}^{M=5} \sqrt{\lambda_n} c_n f_n(s) \vec{n}(s) \tag{4.33}$$

Finally, by defining the rest of the parameters as described previously, seven random realisations of the NACA 0012 airfoil are being computed and displayed in Figure 4.7.

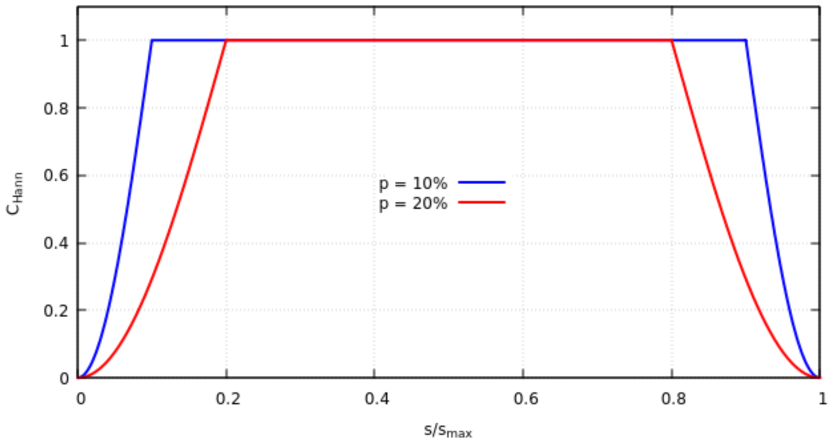


Figure 4.6: *The Hanning Window coefficient C_{Hann} computed and plotted for $p = 10\%$ (blue) and $p = 20\%$ (red).*

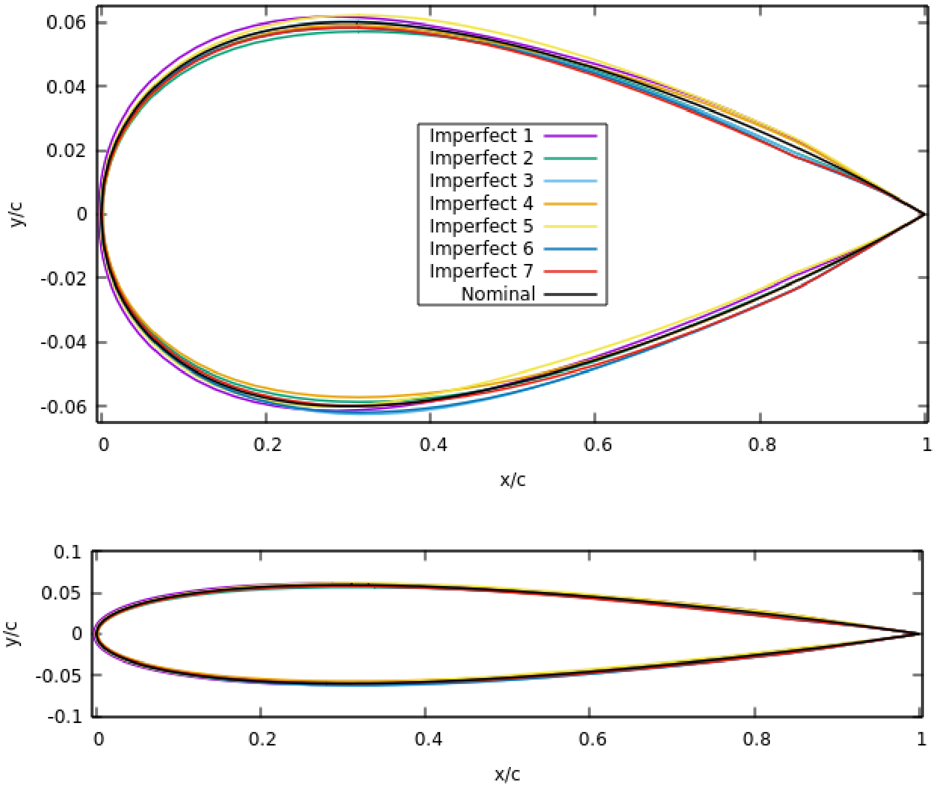


Figure 4.7: *The nominal (black) and seven imperfect (other colors) NACA 0012 airfoils, generated through KLT, not in scale (above) and in scale (below).*

Chapter 5

Uncertainty Quantification with Manufacturing Imperfections

In this chapter, a description of the integration of the previously mentioned KLT software (used to compute imperfect 2D geometries) coupled with the OpenFOAM mesh generators and fluid flow solvers into the in-house niPCE software takes place. Also, various simulations concerning the UQ of the aerodynamic performance of 2D bodies with their manufacturing imperfections, are presented.

5.1 Integration of KLT software and OpenFOAM solvers into the niPCE software

The software developed in C++ that implements the KLT onto a 2D geometry, in order to generate an imperfect shape, thus simulating a manufactured final mechanical part, is named *foilKLT*. As the name reveals, this software specializes into 2D aerodynamic bodies such as isolated or turbomachinery blade airfoils. The core algorithm is vastly based upon the generalized algorithm of KLT application on 2D shapes, as presented in Subsection 4.1.4. Its purpose is to be coupled with a CFD solver, so as to compute the QoI needed for the process of UQ, performed through the non-intrusive Polynomial Chaos Expansion code developed by the PCOpt/NTUA.

The data defining the nominal aerodynamic body is given to *foilKLT* in the form of discrete points (coordinates) in a date file (set **airfoil.dat** for now). Then KLT is implemented to create an imperfect realization of the nominal geometry. Another data file, named **KLT.ini**, is utilized to define the KLT process parameters, the input and output file names, while, also, giving the option whether or not to apply on the final imperfect geometry, the Hanning Window or the absolute KLE perturbations magnitude post-processing subroutines, that were thoroughly described in Subsection 4.2.2. The imperfect body outputted by the program has the same format as the input geometry file and has a variable file name (set **impFoil** for this thesis).

The most important difference of *foiKLT* w.r.t. the aforementioned KLT algorithm, is that the random number generator, mentioned in Step 4 (see Subsection 4.1.4), used to designate the values of the uncertain variables c_n, is replaced by an additional input file of variable name. As such, the uncertain variable vector $\vec{c}$ used in the KLE, can be defined through an outside source, in this specific case by the niPCE in-house code through a data file (usually named **task.dat**).

Because the input and the output files of *foilKLT* are 2D Cartesian coordinate data files, the problem, of integrating them into the OpenFOAM 3D mesh configuration, arises. The easiest solution to this problem would be to integrate the imperfect 2D geometry coordinates into the input file of the OpenFOAM mesh generator (e.g. *blockMeshDict* for the *blockMesh* mesher), used to create the mesh in the first place, thus re-meshing for every call of *foilKLT*. While practical, this solution is costly, especially in scenarios with complex geometries that require mesh generation of significant computational cost. To avoid this shortcoming, the source OpenFOAM *moveDynamicMesh* solver can be implemented, so as to only re-mesh the solid boundaries defining the geometry under consideration, as well as the mesh close to them. The *moveDynamicMesh* can only accept certain triangulated surface file formats (i.e. **.stl, .obj, .vtk, .tri** etc) as inputs, in order move specific mesh regions. Thus, an additional software is developed in C++, named *preSucOBJ*, for the purpose of converting the imperfect 2D geometry file into a triangulated 3D surface file, specifically in the **.obj** file format. In order for the aforementioned process to take place, the to-be imperfect wing length must be specified. Finally, once all the above are executed, the displacement Laplacian solver *moveDynamicMesh* has renewed the **polyMesh** directory, re-defining the mesh region of the flow field. Then, the OpenFOAM solver runs, which for this specific chapter is the incompressible flow solver *simpleFoam*. Once the flow solver has converged to a solution for the flow field variables, one or more QoI (e.g. drag coefficient etc) are computed and written into a final communication file (usually named **task.res**) which is, then, inputted into the *niPCE* software thus closing the process of computing the QoI.

This whole process is called and performed several times when the niPCE code is executed, while it is highly recommended to run the CFD code in *parallel execution*, in order to dramatically reduce the wall-clock time, a.k.a. response time, of the operation in question. In Figure 5.1, the flow-chart of the QoI computation process is presented.

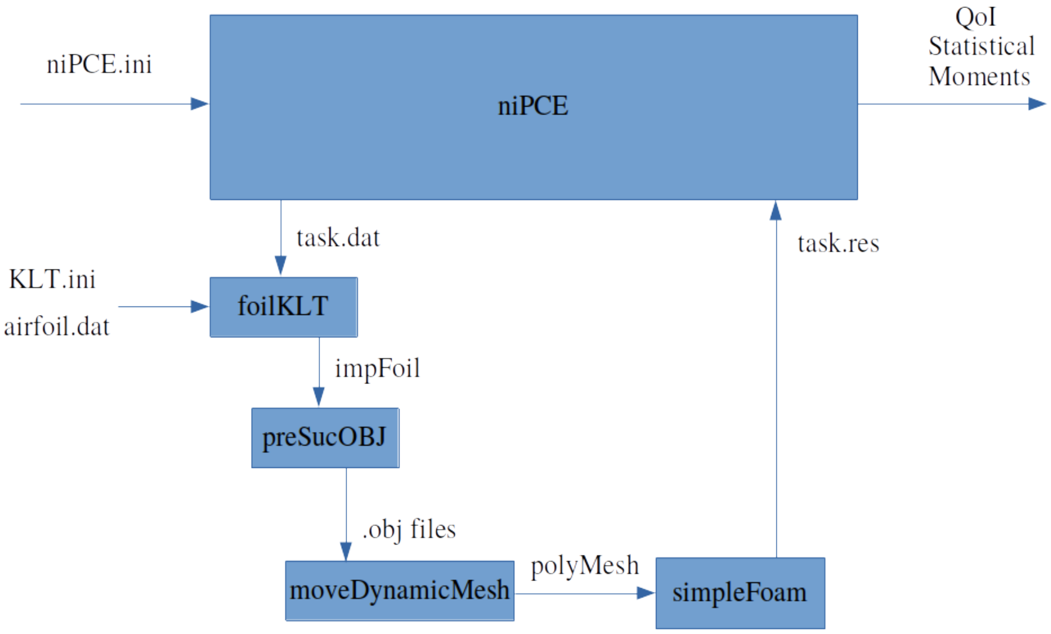


Figure 5.1: *The QoI computation process flow-chart, including the names of the communication/data files of the software used in it. This process is repeated as many times as dictated by the niPCE algorithm, depending mostly on the dimension M of the uncertain variable vector and the order k of the Chaos Expansion polynomials. For simplicity reasons, the niPCE algorithm is displayed as a black box, in order to better focus on the computation of the QoI.*

5.2 Uncertainty Quantification: NACA 0012 airfoil

In this section the initial conditions, the mesh, the flow field and the QoI results (without uncertainties) of the NACA 0012 airfoil CFD case are presented. The algorithm described in Section 5.1, is then applied on the case in question, in order to assess the potential of the KLT in UQ with manufacturing uncertainties. UQ results, computed for several values of the chaos order parameter k, are displayed and compared with MC results, due to its simplicity and accuracy [3], if, at least, an adequate number of replicates is used.

5.2.1 Flow Field Initial Conditions and Mesh

The flow field around the airfoil is considered to be steady, incompressible and turbulent. The NACA 0012 chord has been set to 1 *meter*. The flow initial conditions and properties are displayed in Table 5.1. The flow is solved on a structured, 2D and cell centered C-type grid, consisting of 160 × 135 quadrilateral elements, generated through *blockMesh*. The farfield boundaries of the mesh around the airfoil are set to a 10 chord lengths away form it. The mesh in question is

Farfield Velocity Magnitude	U_∞	$26\ m/s$
Farfield Velocity Angle	α	$2.0\,^o$
Kinematic Viscocity	ν	$10^{-5}\ m^2/s$
Density	ρ	$1.225\ m^3/kg$
Reynolds Number	Re_c	$2.6 \cdot 10^{+6}$

Table 5.1: *The constant initial condition of the NACA 0012 CFD-case.*

visualized in Figures 5.2 and 5.3.

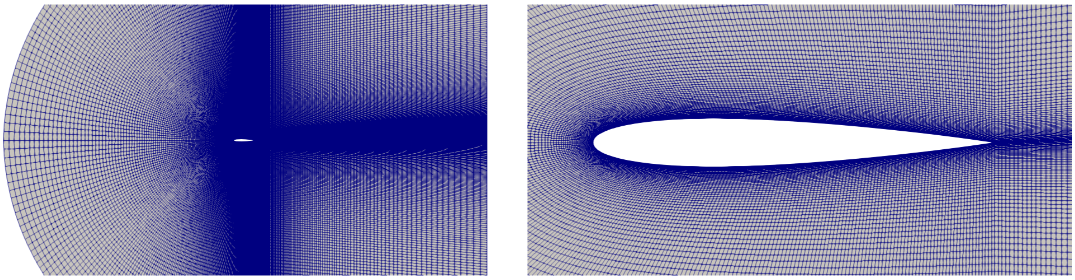


Figure 5.2: *NACA 0012 airfoil: structured C-type mesh.*

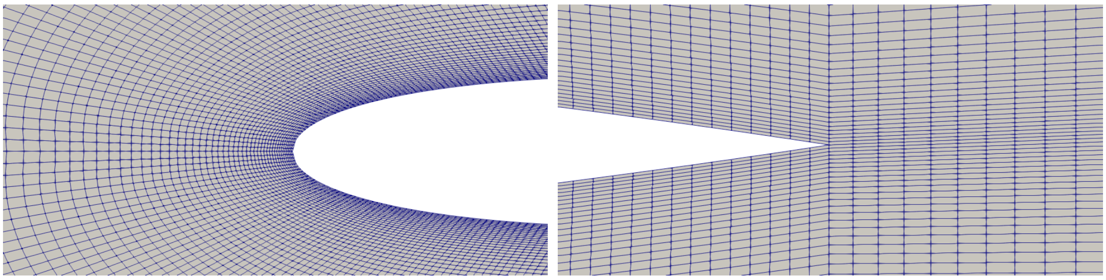


Figure 5.3: *NACA 0012 airfoil: structured C-type mesh, close up view of the airfoil's leading (above) and trailing edges (below).*

The turbulence model used in this CFD case, is the Spalart–Allmaras one [16], while a common assumption in airfoil flow field is used, which dictates that $\tilde{\nu}/\nu = 5$, at the inlet boundary. Hence, the initial condition for the turbulence model are: Spalart–Allmaras model variable $\tilde{\nu} = 5 \cdot 10^{-5}\ m^2/s$ and turbulent kinematic viscosity $\nu_t = 1.29 \cdot 10^{-5}\ m^2/s$. The OpenFOAM's *nutUSpaldingWallFunction* High-Re wall function [18] is utilized to compute the velocity of the near-wall cell centers, that are affected by viscous flow phenomena, further analyzed in Subsection 2.1.3. For this model to have an acceptable accuracy, the first cell center must lie into the log-law region of the boundary layer, meaning that the non-dimensional distance between its cell center and the wall must be $y^+ < 100$. In order to verify the use of

High-Re wall functions for this particular mesh, the distribution of y^+ is plotted for the pressure and suction sides of the airfoil, in Figure 5.4.

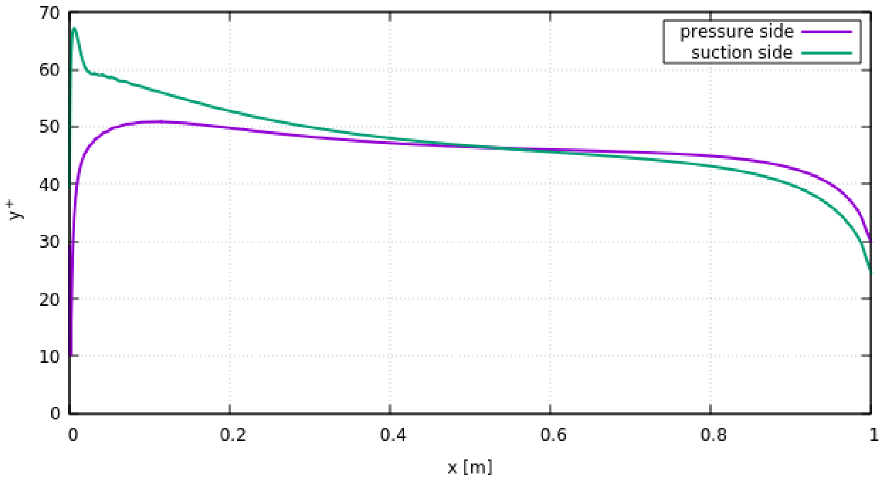


Figure 5.4: *Non-dimensional wall distance y^+, plotted for the pressure and the suction side of the mesh around NACA 0012 airfoil.*

5.2.2 CFD Results without Uncertainties

The flow field equations system is solved by making use of the SIMPLE finite volume algorithm [29], implemented through OpenFOAM©'s executable *simpleFoam*, as described in Subsection 2.4. 2^{nd} order finite volume schemes are used to discretize the $div()$ and $grad()$ operators present into the RANS and Spalart–Allmaras equations. A converged solution is reached in 1400 iterations and the convergence chart of the mean flow variables is presented in Figure 5.5.

The lift (C_L) and drag (C_D) coefficients are used as QoIs in Subsection 5.2.3 and their current values, computed for no uncertainties are displayed in Table 5.2. Furthermore, the velocity magnitude U and the turbulent kinematic viscosity ν_t fields, close to the airfoil, are visualized in Figure 5.6.

$NACA$ 0012 $Aerodynamic\ Coefficients$	
C_L	C_D
0.210755	0.011127

Table 5.2: *NACA 0012 airfoil: lift and drag coefficients for the constant flow conditions described in Table 5.1 ($\alpha = 2\,^o$, $Re_c = 2,600,000$).*

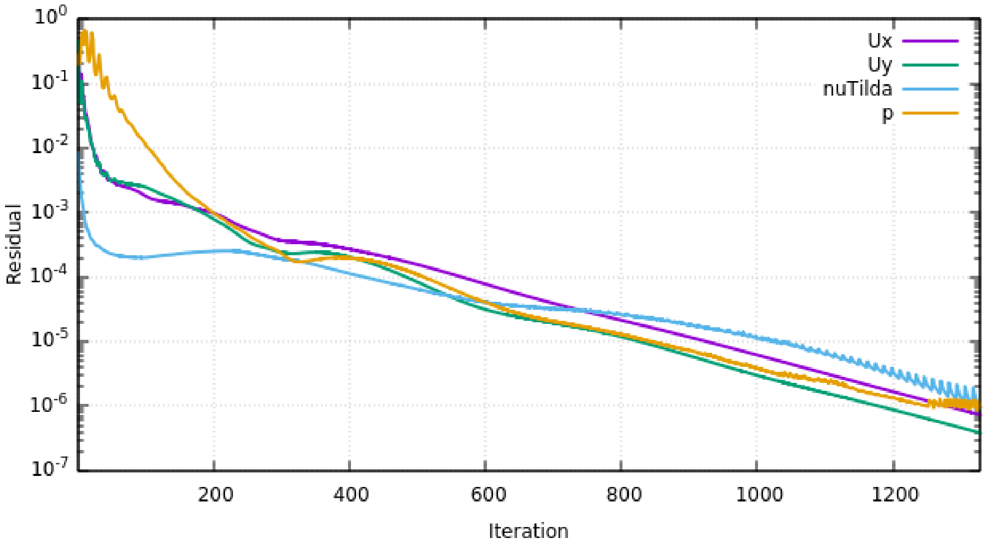


Figure 5.5: *NACA 0012 airfoil: convergence plot of the flow field equations. The convergence criterion is set at a residual equal to* 10^{-6}.

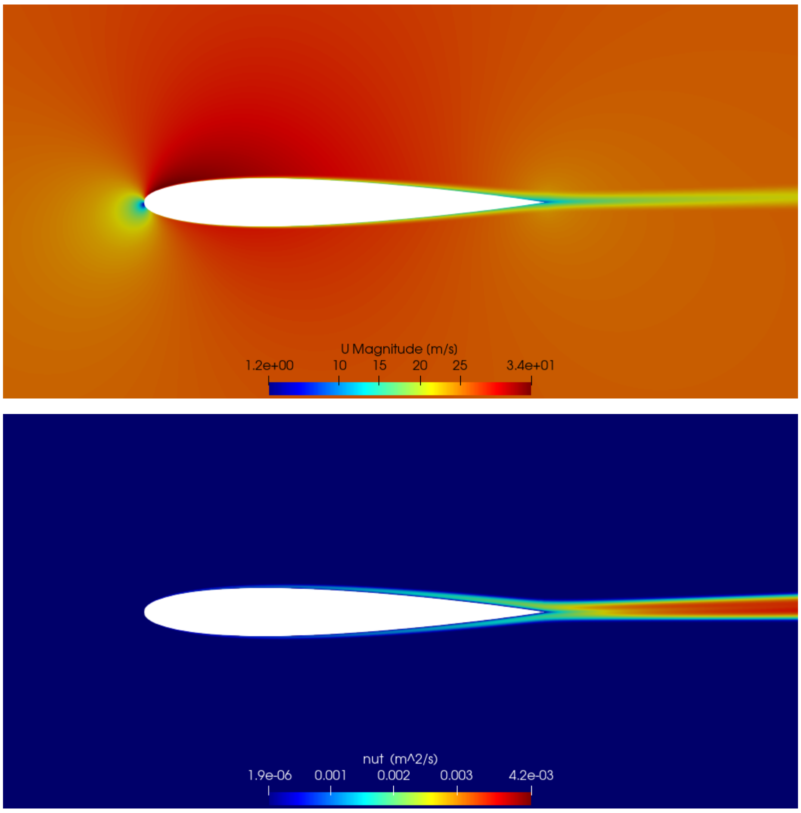


Figure 5.6: *NACA 0012 airfoil: velocity magnitude* U *(above) and turbulent viscosity* ν_t *(below) contours around the airfoil, computed for the nominal shape.*

5.2.3 Chaos Order Parametric Analysis & Results Validation

The UQ process is performed through niPCE for the flow around an imperfect NACA 0012 airfoil. The imperfections are modeled with the KLT algorithm presented in Section 5.1. The solution of the flow equations is achieved with the configuration thoroughly described in Subsection 5.2.2.

The only input parameter of the case in question that is considered uncertain is the airfoil geometry. Specifically, the NACA 0012 airfoil is assumed to display a variation within its manufacturing tolerances. This uncertainty is quantified through the uncertain variable vector $\vec{c} \in \mathbb{R}^M$ of the KLE. The dimension of $\vec{c}$ is set equal to $M = 5$, given that it is considered to be a middle ground solution between the complexity of the KLT-generated imperfect geometries and the mitigation of computational cost. The KLE standard deviation is set equal to $\sigma = 1$ (so as to simplify this variable, for which no empirical data are given), while the statistical moments of the uncertain variables c_l, used to produce the KLT stochastic shape imperfection (eq. (4.29)), are defined as

$$\mu_l = 0\, m \quad , \quad \sigma_l = 2 \cdot 10^{-3}\, m \quad \forall \quad l = 1, 2, ..., 5 \tag{5.1}$$

Another decision taken, so as to reduce the computational cost of the UQ, is the use of Smolyak Sparce grids [33] instead of full grids to define the uncertain variables (further analysis in other Chapters not yet completed). This method greatly reduces the sample points needed for the niPCE, mitigating consequently the computational cost of the GQ integration. Thus, a parametric analysis is held on the order k of the Chaos Expansion polynomials, while MC is used as benchmark method for verifying the computed results [3]. The computational cost of each niPCE run for both the Smolyak and the full grids, measured in Equivalent Flow Solution (EFS) time given that it is equal to the number of Gauss Nodes, is displayed in Table 3.2.

The UQ parametric analysis of the niPCE method (with Smolyak grids), for the QoIs, C_D and C_L, are computed and displayed in the Figures 5.7 and 5.8, accordingly. The MC-computed UQ, for 5000 sample runs, is also included, for the purposes of result comparison and validation. The same data is also presented, in the aggregate, in Table 5.3, along with the specific error of the niPCE results when compared with the corresponding MC results.

niPCE − *Smolyak* $(M = 5)$				
k	μ_{C_D} (% Δ)	μ_{C_L} (% Δ)	σ_{C_D} (% Δ)	σ_{C_L} (% Δ)
1	$1.125 \cdot 10^{-2}$ (0.05)	$2.137 \cdot 10^{-1}$ (**0.43**)	$9.028 \cdot 10^{-5}$ (**0.84**)	$1.207 \cdot 10^{-2}$ (9.60)
2	$1.124 \cdot 10^{-2}$ (0.05)	$2.159 \cdot 10^{-1}$ (0.61)	$9.452 \cdot 10^{-5}$ (5.57)	$1.479 \cdot 10^{-2}$ (10.73)
3	$1.129 \cdot 10^{-2}$ (0.39)	$2.133 \cdot 10^{-1}$ (0.57)	$9.584 \cdot 10^{-5}$ (7.05)	$1.373 \cdot 10^{-2}$ (**2.79**)
4	$1.124 \cdot 10^{-2}$ (**0.04**)	$2.157 \cdot 10^{-1}$ (0.55)	$9.570 \cdot 10^{-5}$ (6.89)	$1.436 \cdot 10^{-2}$ (4.59)
MC	$1.125 \cdot 10^{-2}$	$2.146 \cdot 10^{-1}$	$8.953 \cdot 10^{-5}$	$1.335 \cdot 10^{-2}$

Table 5.3: *NACA 0012 airfoil UQ with manufacturing imperfections. UQ case with $M = 5$ and $k = 1$ to $k = 4$. Mean value and standard deviation of the C_D and C_L computed through niPCE with Sparse grid GHQ and compared with the outcome of MC with 5000 replicates. For each column, the niPCE result with the lowest relative error is highlighted in* ***bold****.*

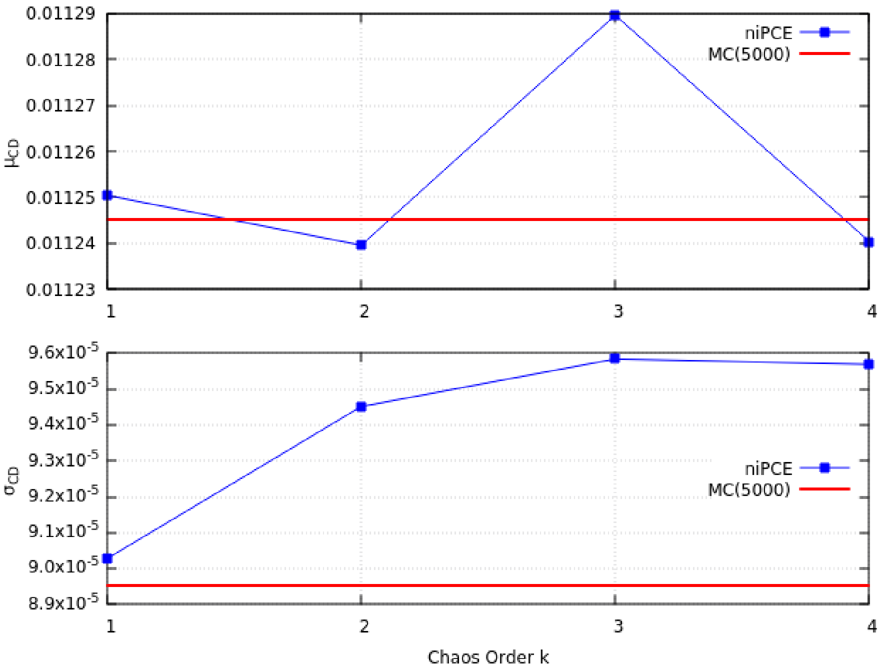


Figure 5.7: *NACA 0012 airfoil drag coefficient UQ with manufacturing imperfections. Results computed with MC, for 5000 samples, and with niPCE, for chaos order $k = 1$ to $k = 4$.*

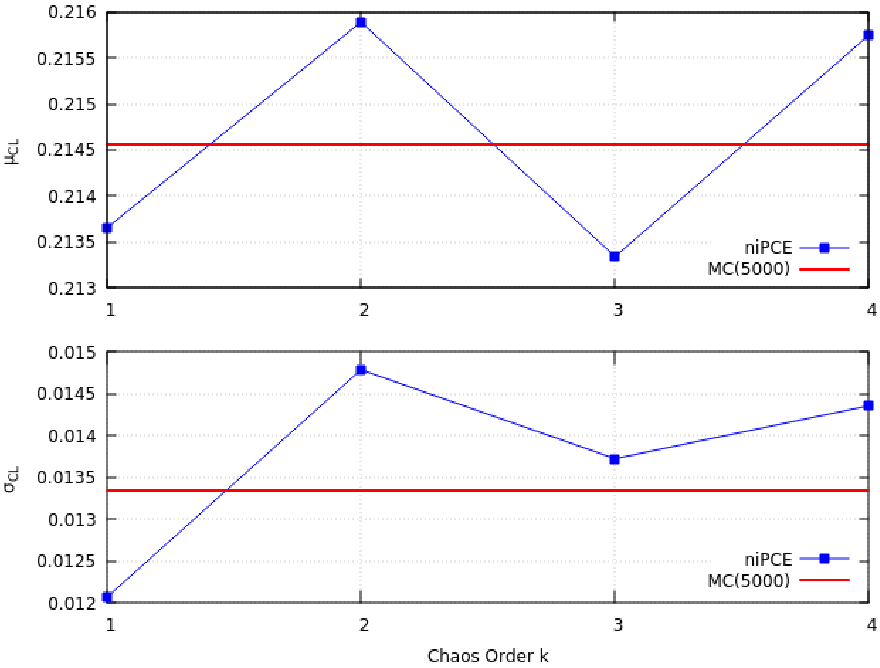


Figure 5.8: *NACA 0012 airfoil lift coefficient UQ with manufacturing imperfections. Results computed with MC, for 5000 samples, and with niPCE, for chaos order $k = 1$ to $k = 4$.*

Finally, enough data has been gathered in order to select the optimal value for the chaos order k, so as to continue with the aerodynamic RDO of lifting bodies with manufacturing imperfections. This analysis has took both the increase of CPU cost and the fluctuating accuracy of the niPCE method into account, in response to the increase of k. Therefore, the illustrated of k consists of a MOO optimization problem with two objectives: the minimization of the cost and the error of the UQ results. The average specific error of all niPCE-computed UQ results when compared to the ones computed through the MC method, is plotted w.r.t. their computational cost in Figure 5.9 (according to Tables 3.2 and 5.3).

As illustrated in Figure 5.9, the dominant solutions, defining the Pareto Front of this parametric analysis, are the UQ results for $k = 1$ and $k = 3$, with mean relative errors of 2.73 % and 2.70 %, respectively. These two solutions combine the lowest CPU cost and the highest average accuracy. Any of the two dominant solutions for the chaos order is a viable option for the user in this specific application. Yet, the difference in their mean accuracy can be considered unnoticeable, while their respective CPU cost difference is quite vast (by an order of magnitude). Given that both solutions are relatively accurate when it comes to the mean value of the QoIs, the one for $k = 1$ computes more accurately the standard deviation of the drag coefficient, while the other for $k = 3$ is more accurate for the computation of the standard deviation of the lift coefficient (as assumed form Table 5.3). Yet, this result may simply be circumstantial and thus no general rule can be deduced from this analysis without further investigation. Anyhow, the lowest chaos order $k = 1$ for the niPCE method with Smolyak integration, has proven to be relatively viable in its UQ results, while simultaneously boasting the lower possible clock-time cost. It is the main reason for which it becomes the selected chaos order value for the simulations to come, in this thesis.

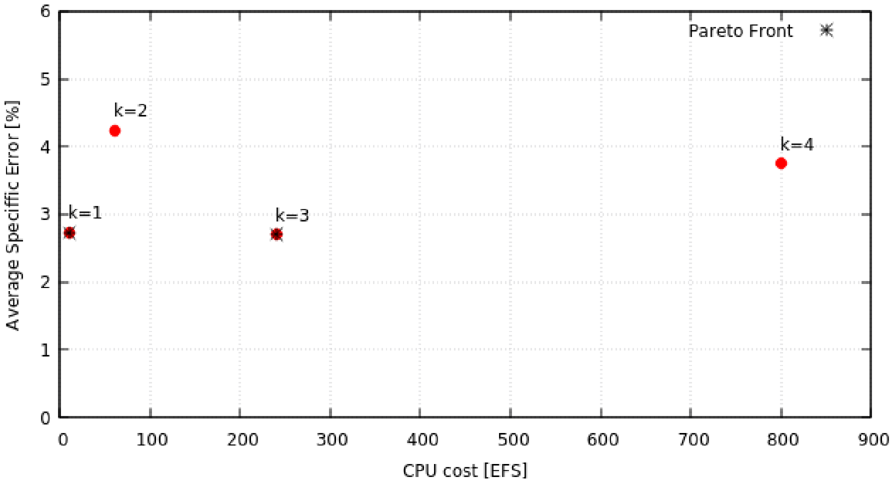


Figure 5.9: *NACA 0012 airfoil: average specific error and CPU cost of niPCE-computed UQ results, for $k = 1$ to 4.*

5.3 Uncertainty Quantification: E387 airfoil

In this section, the same process is applied for the E387, an airfoil most often used in sailplanes and other low Mach applications. The same initial conditions and flow solver are applied, with the exception that, this time, the MC and niPCE methods are employed for the computation and verification of the derivatives of the statistical moments of a certain QoI w.r.t. to the selected design variables. For this analysis, the algorithm, presented in Section 5.1, is executed with one slight modification: the primal problem solver is followed by the adjoint problem solver, both included in the OpenFOAM© executable *adjointOptimisationFoam* developed by PCOpt/NTUA, so as to conduct the sampling of the SDs needed to perform the UQ process.

5.3.1 CFD Analysis without Uncertainties

The flow field around the E387 airfoil is, again, considered to be steady, incompressible, viscous and turbulent, while the chord has been set to 1 *meter* in length. The flow initial conditions and properties are assumed to be identical to those of Subsection 5.2.1, as displayed in Table 5.1.

The computational C-type mesh generated around the E387 has the same exact properties (160 × 135 quads) with the one described in Subsection 5.2.1, and is presented in Figure 5.10: The incompressible primal and adjoint problems for are

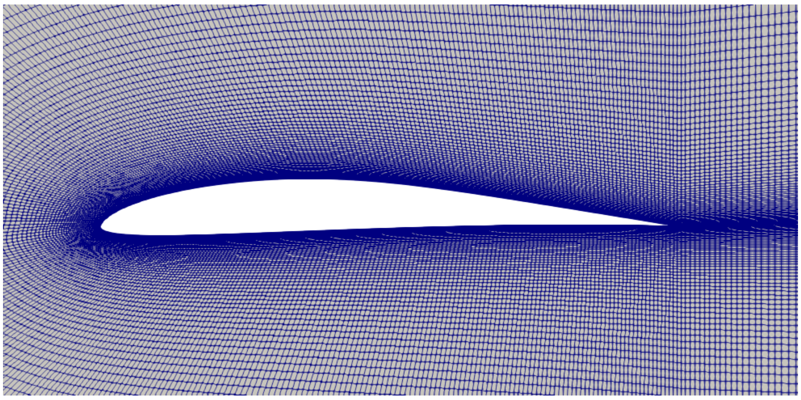


Figure 5.10: *E387 airfoil: structured C-type mesh.*

both solved through the use of the SIMPLE finite volume algorithm by implementing the same configuration as described in Subsections 5.2.1 and 5.2.2. For a single run without any uncertainties, the primal and adjoint problems reach a converged solution after 1665 and 1854 iterations, respectively. The convergence chart for both problems is presented in Figure 5.11.

Furthermore, the C_L and C_D coefficients, computed for no uncertainties are displayed in Table 5.4, while only the second is used as a QoI, mainly for the computation of it's derivatives w.r.t. the design variables.

$E387\ Aerodynamic\ Coefficients$	
C_L	$C_D\ (QoI)$
0.587956	**0.012986**

Table 5.4: *E387 airfoil: lift and drag coefficients for the constant flow conditions described in Table 5.1 ($\alpha = 2\,^o$, $Re_c = 2,600,000$).*

Additionally, the static pressure p and the adjoint pressure p_a, as well as the $\widetilde{\nu}$ and the $\tilde{\nu}_a$ fields, close to the airfoil, are visualized in Figures 5.12 and 5.13, respectively.

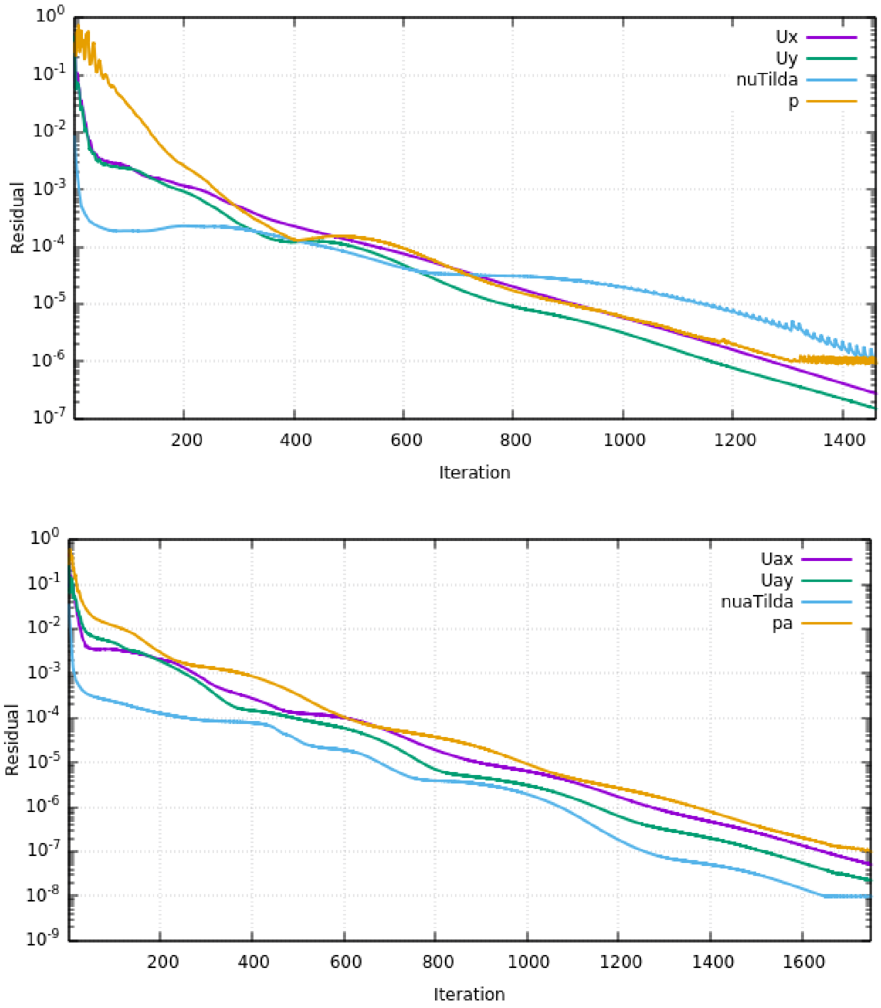


Figure 5.11: *E387 airfoil: convergence chart for primal (above) and the adjoint (below) problem variables. The convergence criterion is set at a residual equal to* 10^{-6} *for the primal and* 10^{-7} *for the adjoint problem.*

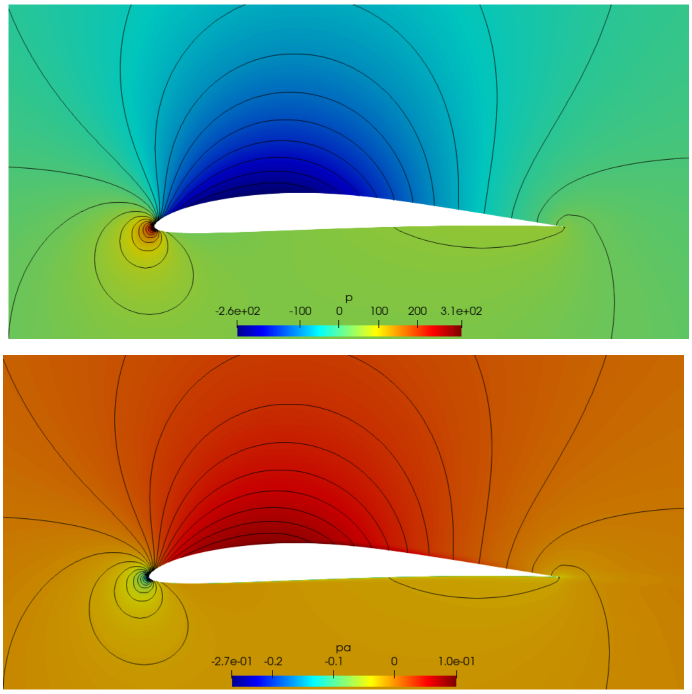


Figure 5.12: *E387 airfoil: static (above) and adjoint pressure (below) contours, close to the airfoil.*

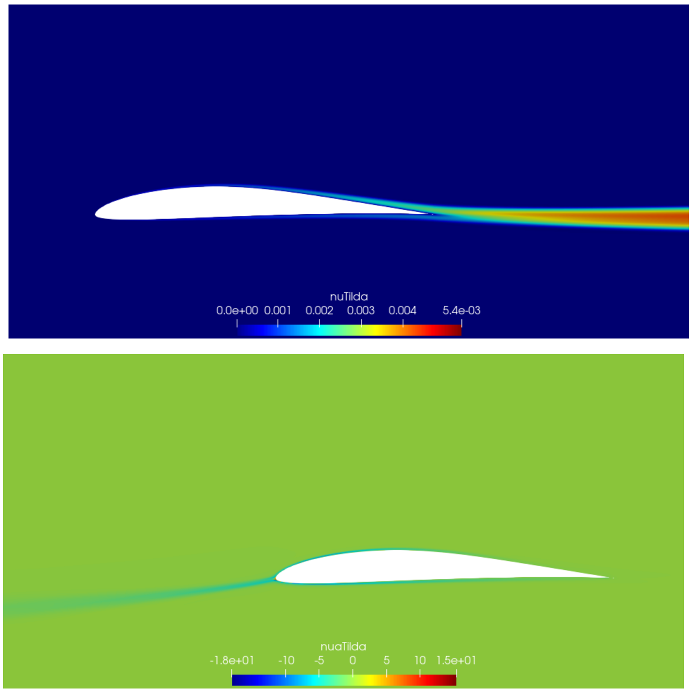


Figure 5.13: *E387 airfoil: Spalart–Allmaras variable $\widetilde{\nu}$ (above) and adjoint Spalart–Allmaras variable $\tilde{\nu}_a$ (below) contours, close to the airfoil.*

Finally the mesh close the surface of the airfoil is parameterized through the use of *Volumetric B-Splines* [28]. The control box, a.k.a. the grid of control points dictating the deformation of the airfoil in accordance with the computed sensitivities and the selected update method, is displayed in Figure 5.14.

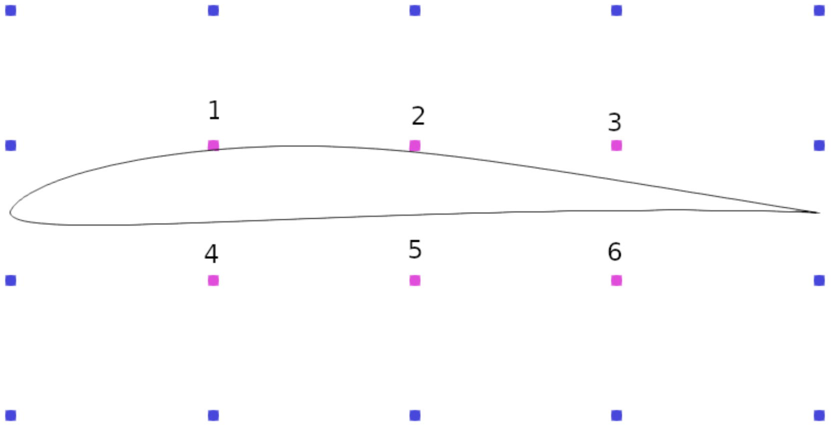


Figure 5.14: *E387 airfoil: Volumetric B-Spline 2D control box for the mesh paramaterization. The blue colored control points are inactive (= immovable), while the magenta colored ones are active, thus constituting the design variables for this problem. The outline of the E387 is colored in black, while the active control points are also enumerated from 1-6.*

5.3.2 Derivatives of the Robustness Metric Verification

In order to compute the robust SDs, according to equation (1.4), the derivatives of the mean value and the standard deviation of a QoI w.r.t. the design variables, must be primarily computed. The selected QoI is the drag coefficient C_D, while the desired design variables are the active Volumetric B-Splines control points, as presented in Figure 5.14. The aforementioned method for the execution of this task, are the niPCE and MC methods for the computation of the statistical moments' derivatives, sampled through the use of the adjoint method.

So as to culminate to relatively accurate result, the MC method is employed for 1000 replicates, i.e. 2000 EFS given that each run requires the solution of both the primal and the adjoint problems once. Meanwhile, the niPCE method is configured for a chaos order of $k = 1$ and for both Full as well as Smolyak Sparse sampling grids. Therefore, according to Table 3.2, the niPCE sampling runs have a CPU cost of $2 \cdot 32 = 64$ EFS for the Full Grid integration and $2 \cdot 11 = 22$ EFS for the Smolyak grid integration. The use of higher chaos orders, for the niPCE, could prove costly, especially when included into a RDO algorithm in latter stages of the thesis. Similarly with Section 5.2, the niPCE and MC result are compared for the purpose of validating the first. The UQ results of $\frac{\partial \mu_{C_D}}{\partial b_n}$ and $\frac{\partial \sigma_{C_D}}{\partial b_n}$ are presented in Tables 5.5 and 5.5, respectively.

$\partial\mu_{C_D}/\partial b_n \quad (M=5)$						
	$x-dir$			$y-dir$		
	$niPCE \quad (k=1)$		MC	$niPCE \quad (k=1)$		MC
n	*Full Grid*	*Smol. Grid*	1000 *sam.*	*Full Grid*	*Smol. Grid*	1000 *sam.*
1	$2.529 \cdot 10^{-4}$	$2.862 \cdot 10^{-4}$	$1.916 \cdot 10^{-4}$	$2.539 \cdot 10^{-3}$	$2.521 \cdot 10^{-3}$	$2.606 \cdot 10^{-3}$
2	$3.189 \cdot 10^{-4}$	$3.203 \cdot 10^{-4}$	$3.155 \cdot 10^{-4}$	$3.415 \cdot 10^{-3}$	$3.419 \cdot 10^{-3}$	$3.405 \cdot 10^{-3}$
3	$8.382 \cdot 10^{-4}$	$8.423 \cdot 10^{-4}$	$8.274 \cdot 10^{-4}$	$9.521 \cdot 10^{-3}$	$9.572 \cdot 10^{-3}$	$9.393 \cdot 10^{-3}$
4	$4.469 \cdot 10^{-4}$	$4.834 \cdot 10^{-4}$	$3.769 \cdot 10^{-4}$	$3.176 \cdot 10^{-3}$	$3.146 \cdot 10^{-3}$	$3.270 \cdot 10^{-3}$
5	$4.654 \cdot 10^{-4}$	$4.671 \cdot 10^{-4}$	$4.611 \cdot 10^{-4}$	$4.506 \cdot 10^{-3}$	$4.509 \cdot 10^{-3}$	$4.497 \cdot 10^{-3}$
6	$9.549 \cdot 10^{-4}$	$9.594 \cdot 10^{-4}$	$9.433 \cdot 10^{-4}$	$1.078 \cdot 10^{-2}$	$1.084 \cdot 10^{-2}$	$1.064 \cdot 10^{-2}$

Table 5.5: *E387 airfoil UQ with manufacturing imperfections, with* $M = 5$. C_D ***mean value derivatives*** *w.r.t. the design variables* b_n *computed with the niPCE* ($k = 1$) *method for both Full and Smolyak Sparse grids, as well as MC for 1000 sample runs.*

$\partial\sigma_{C_D}/\partial b_n \quad (M=5)$						
	$x-dir$			$y-dir$		
	$niPCE \quad (k=1)$		MC	$niPCE \quad (k=1)$		MC
n	*Full Grid*	*Smol. Grid*	1000 *sam.*	*Full Grid*	*Smol. Grid*	1000 *sam.*
1	$-8.557 \cdot 10^{-6}$	$-2.693 \cdot 10^{-6}$	$-4.055 \cdot 10^{-6}$	$-3.358 \cdot 10^{-5}$	$-1.923 \cdot 10^{-5}$	$-3.576 \cdot 10^{-5}$
2	$1.273 \cdot 10^{-5}$	$1.209 \cdot 10^{-5}$	$1.332 \cdot 10^{-5}$	$1.043 \cdot 10^{-4}$	$1.037 \cdot 10^{-4}$	$1.056 \cdot 10^{-4}$
3	$5.546 \cdot 10^{-5}$	$5.338 \cdot 10^{-5}$	$5.715 \cdot 10^{-5}$	$4.632 \cdot 10^{-4}$	$4.421 \cdot 10^{-4}$	$4.835 \cdot 10^{-4}$
4	$-2.443 \cdot 10^{-5}$	$-2.010 \cdot 10^{-5}$	$-1.749 \cdot 10^{-5}$	$-4.399 \cdot 10^{-5}$	$-2.368 \cdot 10^{-5}$	$-5.751 \cdot 10^{-5}$
5	$1.491 \cdot 10^{-5}$	$1.417 \cdot 10^{-5}$	$1.581 \cdot 10^{-5}$	$9.420 \cdot 10^{-5}$	$9.428E-5$	$9.451 \cdot 10^{-5}$
6	$5.813 \cdot 10^{-5}$	$5.609 \cdot 10^{-5}$	$6.034 \cdot 10^{-5}$	$3.573 \cdot 10^{-4}$	$3.357 \cdot 10^{-4}$	$3.814 \cdot 10^{-4}$

Table 5.6: *E387 airfoil UQ with manufacturing imperfections, with* $M = 5$. C_D ***standard deviation derivatives*** *w.r.t. the design variables* b_n *computed with the niPCE* ($k = 1$) *method for both Full and Smolyak Sparse grids, as well as MC for 1000 sample runs.*

The conclusion, from Tables 5.5 and 5.6, is that the niPCE statistical moment derivatives are practically accurate. Though the derivatives computed through the different methods are not identical, the corresponding derivatives have comparable magnitudes and the same algebraic sign. Given that, according to the implemented Steepest Descent method mentioned in Subsection 1.2.3, the design variable update is scaled w.r.t. a user defined maximum displacement, it is easy to assume that this change in the design variables, for all employed UQ methods, may become quite similar. Therefore, this analysis makes it is safe to assume that for these particular 2D airfoil applications the use of the economic niPCE method for the purposes of RDO with manufacturing uncertainties can be conducted with a relatively accurate computation of the 1st degree robust SDs.

When it comes to the values of the derivatives, it is clear that all UQ methods dictate that in order for the airfoil to minimize drag for the specified range of manufacturing imperfection generated through KLT, a displacement of the E387 airfoil's trailing edge is expected. This interpretation can further be backed by Figures 5.15 and 5.16, while cross-referencing with the index of each active control point from Figure 5.14.

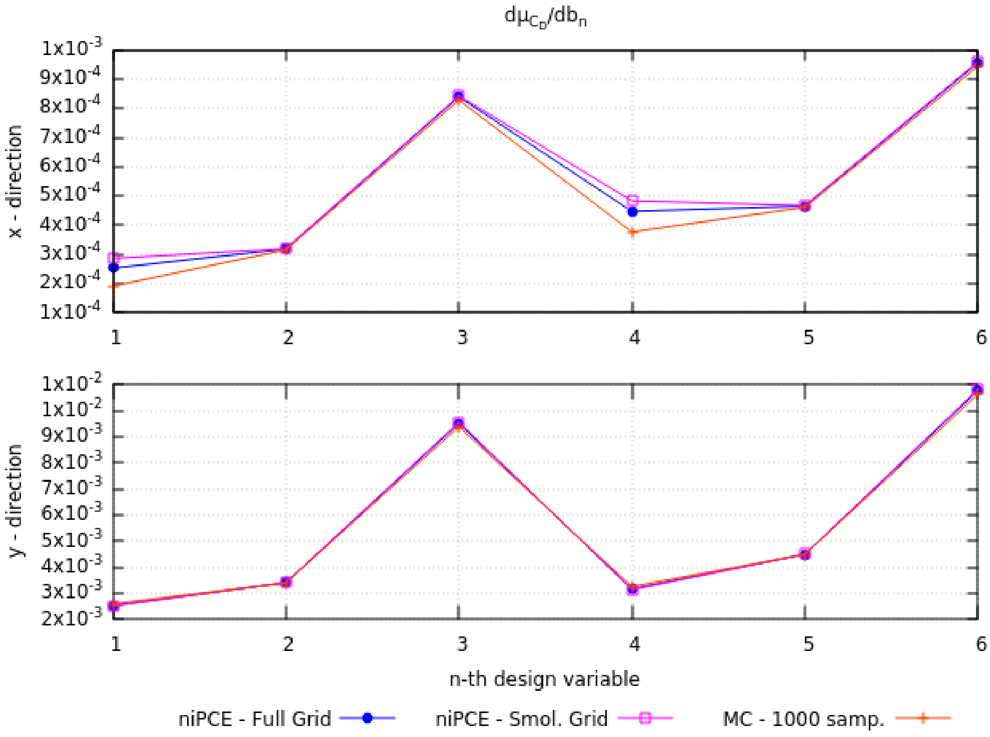


Figure 5.15: *E387 airfoil C_D mean value derivatives w.r.t. the design variables b_n computed with the niPCE method for both Full and Smolyak Sparse grids, as well as MC for 1000 sample runs.*

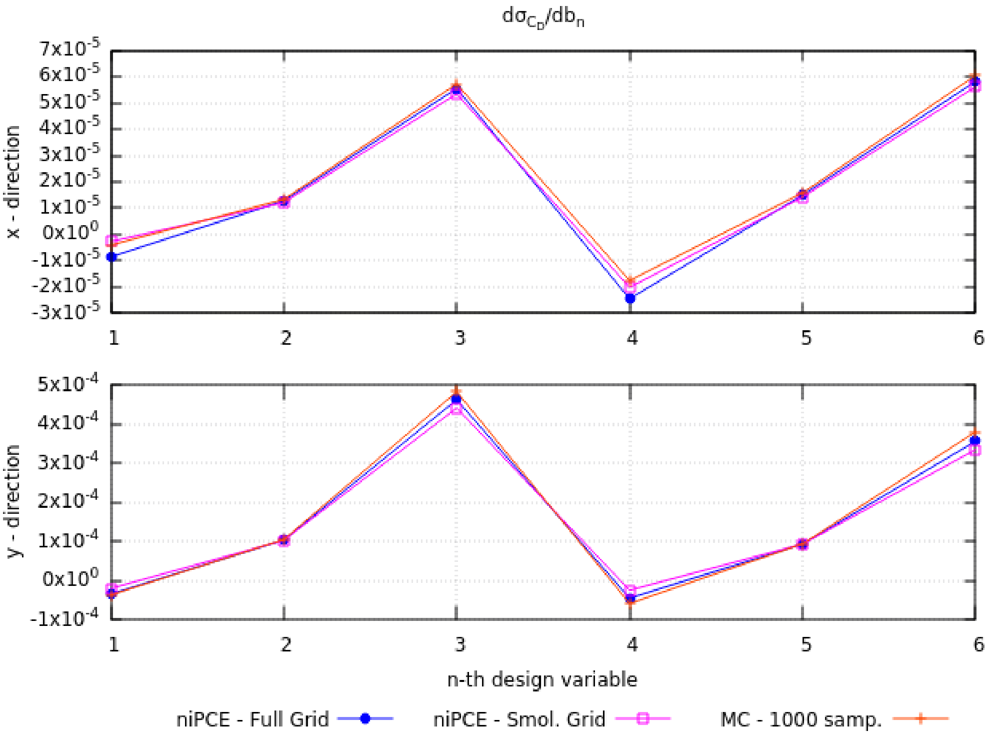


Figure 5.16: *E387 airfoil C_D standard deviation derivatives w.r.t. the design variables b_n computed with the niPCE method for both Full and Smolyak Sparse grids, as well as MC for 1000 sample runs.*

Chapter 6

Robust Design Optimization with Manufacturing Imperfections

In this chapter, the RDO with manufacturing uncertainties of two 2D shapes is presented. The manufacturing imperfections are recreated through the KLT software introduced in Chapter 4, while the primal and adjoint problems are solved though OpenFOAM$^{©}$'s *adjointOptimisationFoam* solver. In order to perform the manufacturing imperfections RDO, the algorithm described in Subsection 5.1 is inserted into a loop and with some additions, is repeated as many times as the maximum optimization cycles, selected by the user. In each loop, the following steps are performed:

1. The niPCE algorithm is called, so as to execute UQ on the user-selected QoIs (computed through the solution of the primal problem), as well as their respective SDs (computed through the solution of the adjoint problem).
2. The robust metric (F_R) and the robust SDs ($\delta F_R/\delta b_n$) are computed according to the DFSS rule (explained in 1.2.2), for the user-defined constant κ.
3. The parameterized initial geometry of the current loop, is displaced, according to the *Steepest Descent* method, in the direction dictated by the robust SDs, by moving the control points of the *Volumetric B-Splines* morphing box (described in Subsection 2.3).

The implementation of the aforementioned algorithmic process, on the E387 airfoil and the TU Berlin TurboLab Stator [13] cascade, embody the two main Sections of the current Chapter.

6.1 Robust Design: E387 airfoil

For the E387 airfoil the primal problem is formulated in the same manner as in Section 5.3, meaning that the same *simpleFoam* configuration, mesh and boundary conditions are used. Furthermore, the adjoint problem is formulated as described in

Subsection 5.3.2, with the only exception being that a denser control box is used, as displayed in Figure 6.1. For every RDO held of the E387 in this Subsection, the maximum control point displacement is set to $\Delta b_{max} = 0.5 \cdot 10^{-2}\ m$. Finally the configuration of KLT shape imperfection generator is unchanged, while, according to the analysis held in Subsection 5.3.2, the niPCE chaos order is set to $k = 1$, for which Smolyak Sparse Grid GHQ integration is used.

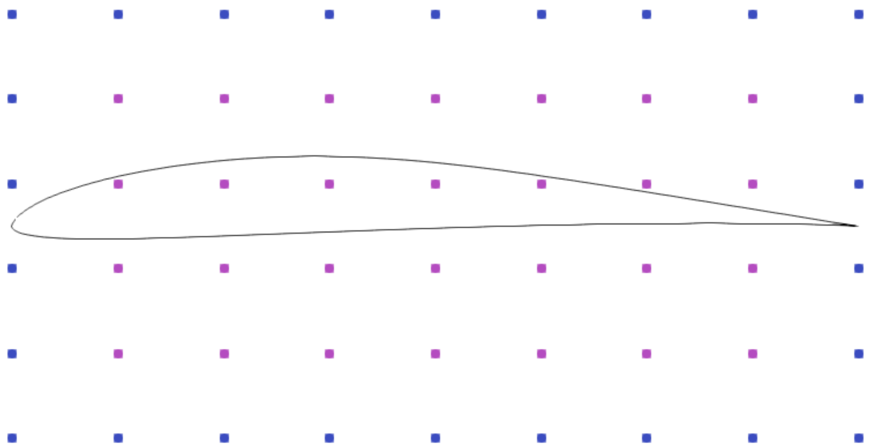


Figure 6.1: *E387 airfoil: Volumetric B-Spline 2D control box for the mesh paramaterization for the purpose of RDO. The blue colored control points are inactive, while the magenta colored ones are active, thus constituting the design variables for this problem. Consequently, a* 9×6 *overall control mesh is used, of which all boundary control points are set as inactive.*

6.1.1 Single-Objective RDO: Robust Drag

Firstly, the E387 airfoil is subjected to single-objective RDO with manufacturing uncertainties while the selected QoI are the drag coefficient C_D, so as to achieve a minimized drag performance within the user-defined shape imperfection spectrum. As set in Subsection 4.2.2, the KLT-modes, i.e. the shape uncertain variables, used are $M = 5$, while their mean value and standard deviation are defined as

$$\mu_l = 0\ ,\quad \sigma_l = 2c \cdot 10^{-3} = 2 \cdot 10^{-3}\ m \ \ \forall \ \ l = 1, 2, ..., 5$$

where $c = 1$ denotes the airfoil chord.

For $M = 5$, $k = 1$ and Sparse Grid integration, according to Table 3.2, in each optimization cycle the primal and adjoint problems are solved 11 times. Therefore, the computational cost of each cycle amounts to $2 \times 11 = 22\ EFS$.

In addition, the robustness metric, for this specific QoI, is also referred to as *robust C_D* and is formulated as follows

$$C_D^{(robust)} = \mu_{C_D} + \kappa\, \sigma_{C_D} \tag{6.1}$$

where κ indicates the DFSS coefficients, defined in Subsection 1.2.2.

Accordingly, the robust SDs, according to which the controlled surfaced is displaced, assume the form

$$\frac{\partial}{\partial b_n} C_D^{(robust)} = \frac{\partial \mu_{C_D}}{\partial b_n} + \kappa\, \frac{\partial \sigma_{C_D}}{\partial b_n} \tag{6.2}$$

Finally, in order to fully monitor the aerodynamic performance of the to-be optimized airfoil, in each RDO loop, UQ is also be performed on the lift coefficient, yielding the *robust C_L*, as follows

$$C_L^{(robust)} = \mu_{C_L} + \kappa\, \sigma_{C_L} \tag{6.3}$$

The RDO process is, therefore, executed for five optimization cycles, for three different values of the DFSS coefficient: $\kappa = -2, 0, +2$. Therefore, three different robust airfoils are yielded, each with a different prioritization over the σ_{C_D} as well as the design approach (pessimistic for $\kappa > 0$ or optimistic for $\kappa < 0$, for minimization problems). In Tables 6.1, 6.2, as well as Figure 6.2, the mean value and the standard deviation of C_D and C_L, respectively, are presented, for the different values of κ, for each of the five optimization cycles.

	$\kappa = +2$		$\kappa = 0$		$\kappa = -2$	
Cycle	μ_{C_D}	μ_{C_L}	μ_{C_D}	μ_{C_L}	μ_{C_D}	μ_{C_L}
1	$1.300 \cdot 10^{-2}$	$5.881 \cdot 10^{-1}$	$1.300 \cdot 10^{-2}$	$5.881 \cdot 10^{-1}$	$1.300 \cdot 10^{-2}$	$5.8814 \cdot 10^{-1}$
2	$1.227 \cdot 10^{-2}$	$5.102 \cdot 10^{-1}$	$1.226 \cdot 10^{-2}$	$5.099 \cdot 10^{-1}$	$1.226 \cdot 10^{-2}$	$5.093 \cdot 10^{-1}$
3	$1.179 \cdot 10^{-2}$	$4.490 \cdot 10^{-1}$	$1.179 \cdot 10^{-2}$	$4.496 \cdot 10^{-1}$	$1.179 \cdot 10^{-2}$	$4.502 \cdot 10^{-1}$
4	$1.147 \cdot 10^{-2}$	$4.006 \cdot 10^{-1}$	$1.147 \cdot 10^{-2}$	$4.027 \cdot 10^{-1}$	$1.148 \cdot 10^{-2}$	$4.048 \cdot 10^{-1}$
5	$1.124 \cdot 10^{-2}$	$3.619 \cdot 10^{-1}$	$1.125 \cdot 10^{-2}$	$3.654 \cdot 10^{-1}$	$1.126 \cdot 10^{-2}$	$3.686 \cdot 10^{-1}$

Table 6.1: *E387 airfoil RDO with manufacturing imperfections: mean values of C_D and C_L for 5 optimization cycles and $\kappa = -2, 0, +2$.*

It is clear that for the the varying values of κ, the final aerodynamic coefficients have minor differences. For all cycles and for both the mean value and standard deviation, the three κ values yield practically the same result. The cause can be found in Subsection 5.3.2, where it can be observed that the SD standard deviation ($\partial \sigma_F / \partial b_n$) is a least an order of magnitude lower than the SD mean values ($\partial / \partial b_n$), for the same CP. The same takes place for the current control grid (Figure 6.1), making the subtraction or the addition of $\partial \sigma_F / \partial b_n$ in eq. (6.2), to cause a nearly insignificant

	$\kappa = +2$		$\kappa = 0$		$\kappa = -2$	
Cycle	σ_{C_D}	σ_{C_L}	σ_{C_D}	σ_{C_L}	σ_{C_D}	σ_{C_L}
1	$2.035 \cdot 10^{-4}$	$1.697 \cdot 10^{-2}$	$2.035 \cdot 10^{-4}$	$1.697E - 2$	$2.035 \cdot 10^{-4}$	$1.697 \cdot 10^{-2}$
2	$1.774 \cdot 10^{-4}$	$1.695 \cdot 10^{-2}$	$1.766 \cdot 10^{-4}$	$1.696E - 2$	$1.765 \cdot 10^{-4}$	$1.695 \cdot 10^{-2}$
3	$1.581 \cdot 10^{-4}$	$1.686 \cdot 10^{-2}$	$1.587 \cdot 10^{-4}$	$1.686E - 2$	$1.589 \cdot 10^{-4}$	$1.686 \cdot 10^{-2}$
4	$1.431 \cdot 10^{-4}$	$1.677 \cdot 10^{-2}$	$1.439 \cdot 10^{-4}$	$1.678E - 2$	$1.449 \cdot 10^{-4}$	$1.678 \cdot 10^{-2}$
5	$1.314 \cdot 10^{-4}$	$1.672 \cdot 10^{-2}$	$1.326 \cdot 10^{-4}$	$1.672E - 2$	$1.339 \cdot 10^{-4}$	$1.673 \cdot 10^{-2}$

Table 6.2: *E387 airfoil RDO with manufacturing imperfections: standard deviation of C_D and C_L for 5 optimization cycles and $\kappa = -2, 0, +2$.*

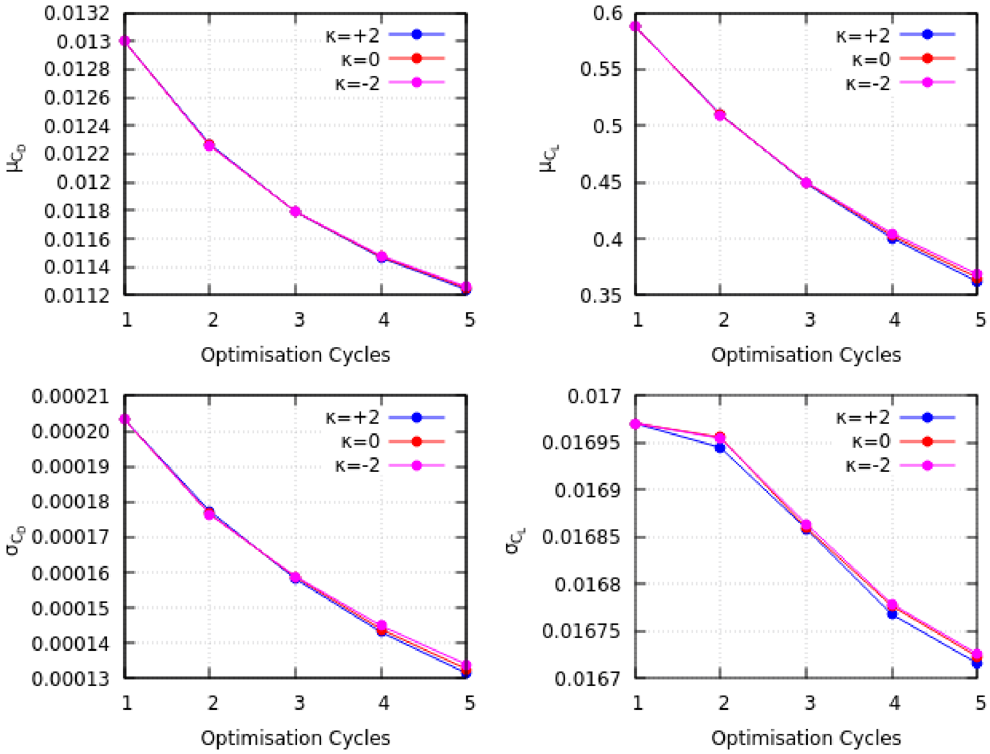


Figure 6.2: *E387 airfoil RDO with manufacturing imperfections:mean value and standard deviation of C_D and C_L, respectively, for 5 optimization cycles and $\kappa = -2, 0, +2$.*

change to the final values of the robust SDs ($\partial F_R/\partial b_n$). Hence, the generated robust airfoils as well as their aerodynamic performance is virtually indifferent, yet this must not be considered a generality in RDO. Yet in this case, greater values of κ should be used in order to see significant changes. The final robust airfoils are displayed in Figure 6.3.

Nevertheless, the goal of this RDO analysis, to minimize the mean value as well as the spread (standard deviation) of the airfoil's drag is achieved. For all values of κ, the μ_{C_D} and σ_{C_D} are reduced by approximately 13.5 % and 35 %, respectively. Naturally, the robust geometries generate lower lift average and variation, due to the inverse flap-like shape generated in all robust airfoils' trailing edge that changes the airfoil's camber (as observed in Figure 6.3). The μ_{C_L} and σ_{C_L} are reduced by approximately 38 % and 1.5 %, respectively. The seemingly small reduction in σ_{C_L} is expected, given that the RDO algorithm is set to minimize the statistical moments of drag, not lift.

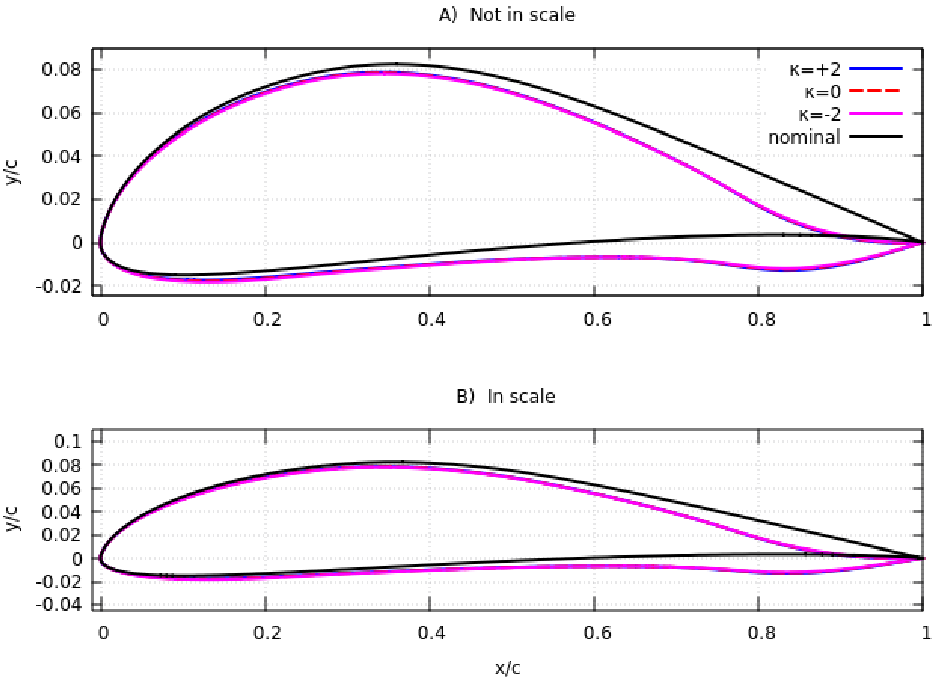


Figure 6.3: *E387 airfoil RDO with manufacturing imperfections: three robust airfoil geometries after 5 optimization cycles, each for a different value of* $\kappa = -2, 0, +2$. *The robust airfoils are visually compared with the initial airfoil, displayed in black, both not in scale (A) and in scale (B).*

The relative deviation of the aerodynamic coefficients is defined the relative difference between each coefficient and their respective mean values, as computed for the 11 KLT imperfect airfoils, generated for the 11 Gauss Nodes of the Smolyak Grid GHQ integration. These values, are presented in Figure 6.4, for the tree final robust airfoils and for the initial E387 airfoil. This is done in order to properly visualize the dispersion of the aerodynamic coefficients, caused by the KLT-generated shape uncertainty, as well as to see, to what extent, this variation is mitigated for the robust drag airfoils.

To conclude, the maximum relative deviation of C_D is reduced by approximately

39.6 % for all robust airfoils, while the maximum relative deviation of C_L is increased by approximately 1 %, which is anticipated given that σ_{C_L} is reduced by a relatively small amount. Overall, the conclusion drawn from these results is that the RDO with shape uncertainties, designed and executed for this thesis, can successfully optimize the stochastic performance of a force objective (i.e. drag). The reduction in the dispersion and, therefore, the standard deviation of a force objective is achievable, by a significant amount, even though the KLT shape imperfections have a minor impact on the lift and drag forces exerted on an airfoil, exhibiting maximum relative variations lower than 3 %.

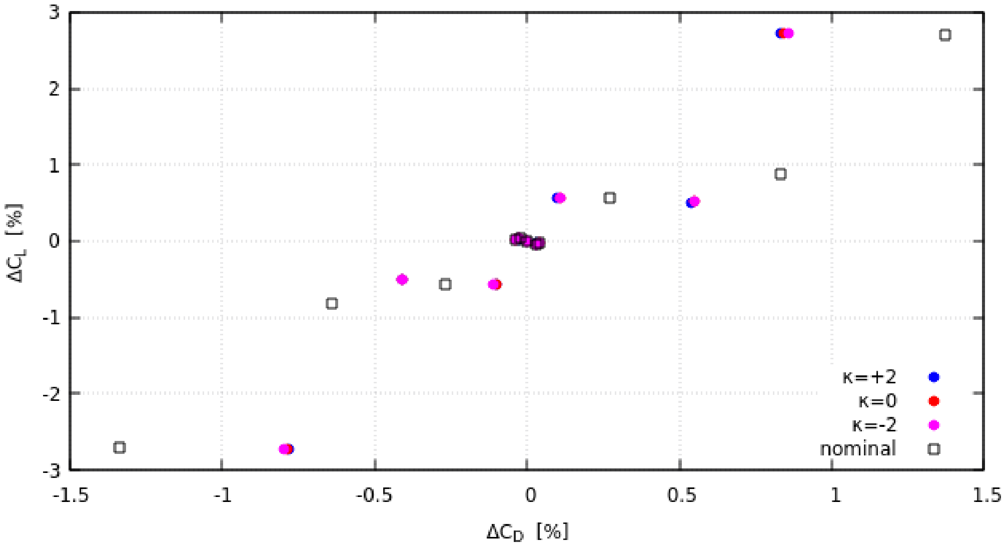


Figure 6.4: *E387 airfoil RDO with manufacturing imperfections: relative deviation ΔC_L, ΔC_D for the three robust airfoil, as well as for the initial E387 airfoil.*

6.1.2 Multi-Objective RDO: Weighted Objectives

Now the E387 airfoil is subjected to multi-objective RDO with manufacturing uncertainties. For this process to take place, the QoI (F) is defined as a weighted sum of the aerodynamic coefficients, thus requiring one adjoint solver instead of two, as formulated in (6.4). The weights (w) serve to quantify the importance of each of the coefficients. The goal of this method is mainly to display the ability to perform RDO with shape uncertainties, in order to achieve a robust drag performance while still being able to contain the lift reduction.

$$F = w_D\, C_D + w_L\, C_L = w_D\, C_D - (1 - w_D)\, C_L \tag{6.4}$$

where $w_D \in [0, 1]$, while the lift weight is set to $w_L = -(1 - w_D) \in [-1, 0]$, so as to have only one weight to denote the lift-drag prioritization.

Additionally, the lift weight is defined as $w_L \leq 0$, to indicate the intention to maximize the lift exerted on the airfoil, given that PCOpt/NTUA's adjoint solver is a de facto minimization algorithm. Consequently, only one objective function is formulated and therefore one adjoint solver is needed, leading to a lower computational cost.

The mean value of this weighted QoI yields

$$\mu_F = w_D\, \mu_{C_D} - (1 - w_D)\, \mu_{C_L} \tag{6.5}$$

while its standard deviation yields

$$\sigma_F = w_D\, \sigma_{C_D} - (1 - w_D)\, \sigma_{C_L} \tag{6.6}$$

Thus, the robustness metric assumes the form

$$\begin{aligned} F_R = \mu_F + \kappa\, \sigma_F = w_D\, \mu_{C_D} - (1 - w_D)\, \mu_{C_L} + \kappa\, [w_D\, \sigma_{C_D} - (1 - w_D)\, \sigma_{C_L}] = \\ = w_D\, (\mu_{C_D} + \kappa\, \sigma_{C_D}) - (1 - w_D)\, (\mu_{C_L} + \kappa\, \sigma_{C_L}) \end{aligned} \tag{6.7}$$

And by including eqs. (6.1) and (6.3) into eq. (6.7), it can be rewritten as

$$F_R = w_D\, C_D^{(robust)} - (1 - w_D) C_L^{(robust)} \tag{6.8}$$

Correspondingly, the derivatives of the statistical moments are formulated as

$$\frac{\partial \mu_F}{\partial b_n} = w_D\, \frac{\partial \mu_{C_D}}{\partial b_n} - (1 - w_D)\, \frac{\partial \mu_{C_L}}{\partial b_n} \tag{6.9}$$

$$\frac{\partial \sigma_F}{\partial b_n} = w_D\, \frac{\partial \sigma_{C_D}}{\partial b_n} - (1 - w_D)\, \frac{\partial \sigma_{C_L}}{\partial b_n} \tag{6.10}$$

Thus, according to eqs. (6.2), the derivative of the robustness metric a.k.a. the robust SD assumes the form

$$\begin{gathered} \frac{\partial F_R}{\partial b_n} = \frac{\partial \mu_F}{\partial b_n} + \kappa\, \frac{\partial \sigma_F}{\partial b_n} = \\ = w_D \left(\frac{\partial \mu_{C_D}}{\partial b_n} + \kappa\, \frac{\partial \sigma_{C_D}}{\partial b_n} \right) - (1 - w_D) \left(\frac{\partial \mu_{C_L}}{\partial b_n} + \kappa\, \frac{\partial \sigma_{C_L}}{\partial b_n} \right) \implies \\ \implies \frac{\partial F_R}{\partial b_n} = w_D\, \frac{\partial}{\partial b_n} C_D^{(robust)} - (1 - w_D) \frac{\partial}{\partial b_n} C_L^{(robust)} \end{gathered} \tag{6.11}$$

where $C_L^{(robust)}$, similar to its drag counterpart defined as

$$\frac{\partial}{\partial b_n} C_L^{(robust)} = \frac{\partial \mu_{C_L}}{\partial b_n} + \kappa\, \frac{\partial \sigma_{C_L}}{\partial b_n} \tag{6.12}$$

The RDO process is, again, executed for five optimization cycles, for $\kappa = +2$ and three different weights: $w_D = 100$, 99.35, 99.25 %. Clearly, for $w_D = 100$ % the lift terms are removed from eqs. (6.8) and (6.11), thus, swifting to the single-objective RDO expressions of Subsection 6.1.1. In addition, the lift weights ($|w_L| < 0.01$) are set to be at least two orders of magnitude lower than the drag weights, because such is the difference between the robust SDs of the two QoI. Lift weights with a greater absolute values, would lead to robust lift being prioritized over robust drag optimization.

Ultimately, three different robust airfoils are designed, each with a different prioritization over the lift stochastic distribution. For the different values of w_D, the robust C_D and C_L (as formulated in eqs. (6.1) and (6.3), respectively) are presented in Table 6.3 and Figure 6.5, for each of the five optimization cycles.

	$w_D = 100.00\,\%$		$w_D = 99.35\,\%$		$w_D = 99.25\,\%$	
Cycle	$C_D^{(robust)}$ (% Δ)	$C_L^{(robust)}$ (% Δ)	$C_D^{(robust)}$ (% Δ)	$C_L^{(robust)}$ (% Δ)	$C_D^{(robust)}$ (% Δ)	$C_L^{(robust)}$ (% Δ)
1	$1.341 \cdot 10^{-2}$ (0.0)	$5.542 \cdot 10^{-1}$ (0.0)	$1.341 \cdot 10^{-2}$ (0.0)	$5.542 \cdot 10^{-1}$ (0.0)	$1.341 \cdot 10^{-2}$ (0.0)	$5.542 \cdot 10^{-1}$ (0.0)
2	$1.263 \cdot 10^{-2}$ (5.8)	$4.764 \cdot 10^{-1}$ (14.0)	$1.260 \cdot 10^{-2}$ (6.0)	$4.745 \cdot 10^{-1}$ (14.4)	$1.259 \cdot 10^{-2}$ (6.1)	$4.731 \cdot 10^{-1}$ (14.6)
3	$1.211 \cdot 10^{-2}$ (9.7)	$4.152 \cdot 10^{-1}$ (25.1)	$1.233 \cdot 10^{-2}$ (8.0)	$4.454 \cdot 10^{-1}$ (19.6)	$1.251 \cdot 10^{-2}$ (6.7)	$4.673 \cdot 10^{-1}$ (15.7)
4	$1.175 \cdot 10^{-2}$ (12.4)	$3.671 \cdot 10^{-1}$ (33.8)	$1.220 \cdot 10^{-2}$ (9.0)	$4.315 \cdot 10^{-1}$ (22.2)	$1.241 \cdot 10^{-2}$ (7.4)	$4.580 \cdot 10^{-1}$ (17.4)
5	$1.150 \cdot 10^{-2}$ (**14.22**)	$3.285 \cdot 10^{-1}$ (40.7)	$1.211 \cdot 10^{-2}$ (9.7)	$4.224 \cdot 10^{-1}$ (23.8)	$1.233 \cdot 10^{-2}$ (8.0)	$4.503 \cdot 10^{-1}$ (**18.7**)

Table 6.3: *E387 airfoil RDO with manufacturing imperfections: $C_D^{(robust)}$ and $C_L^{(robust)}$ values, for 5 optimization cycles, $\kappa = +2$ and three different objective weights. The relative divergence (% Δ) from the initial value is also included.*

The corresponding final robust airfoils produced for the different values of w_D are displayed in Figure 6.6, and compared with the initial E387 airfoil.

It can be stated that, the goal of this RDO process is met, given that the implementation of weighted objectives, can halt the ever decreasing values of the robust C_L. Yet a robust airfoil exhibiting both a lower robust drag and a greater lift than the initial airfoil is not achieved. This is generally expected, given that these two objectives are *incompatible* and, thus "conflicting". Therefore, as observed in Figure 6.6, any change in the airfoil's camber generating a lower drag, simultaneously results to a lower lift force excreted on the airfoil and vice-versa. Thus, the greatest robust drag reduction is achieved for $w_D = 100$ %, by 14.2 %, while the lowest robust lift reduction is achieved for $w_D = 99.25$ %, by 18.7 %.

Finally, the relative deviation of the aerodynamic coefficients (C_L, C_D), as computed for the 11 KLT-generated imperfect airfoils, are presented in Figure 6.7, for the three final robust airfoils, as well as for the initial E387 airfoil. As anticipated, the high w_D values lead to a relatively unaffected maximum C_L deviation, while the maximum deviation of C_D increases, when the w_D value decreases.

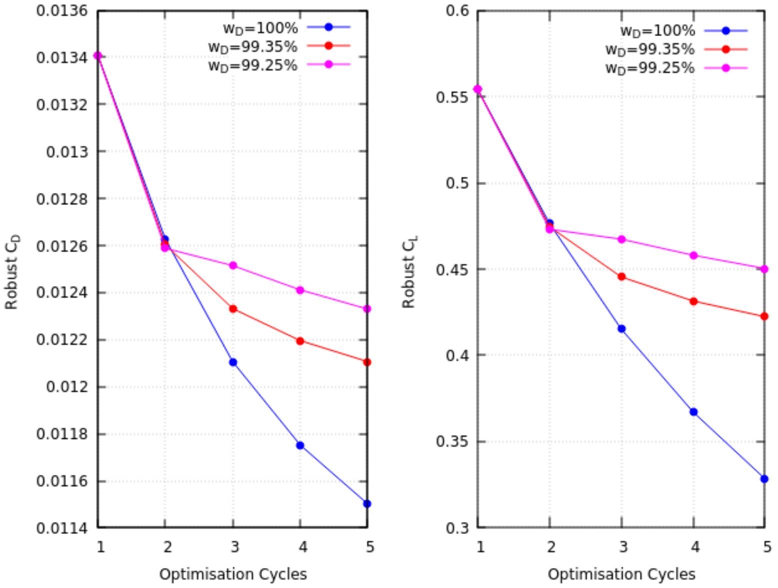


Figure 6.5: *E387 airfoil RDO with manufacturing imperfections:* $C_D^{(robust)}$ *and* $C_L^{(robust)}$ *values, for 5 optimization cycles,* $\kappa = +2$ *and three different objective weights.*

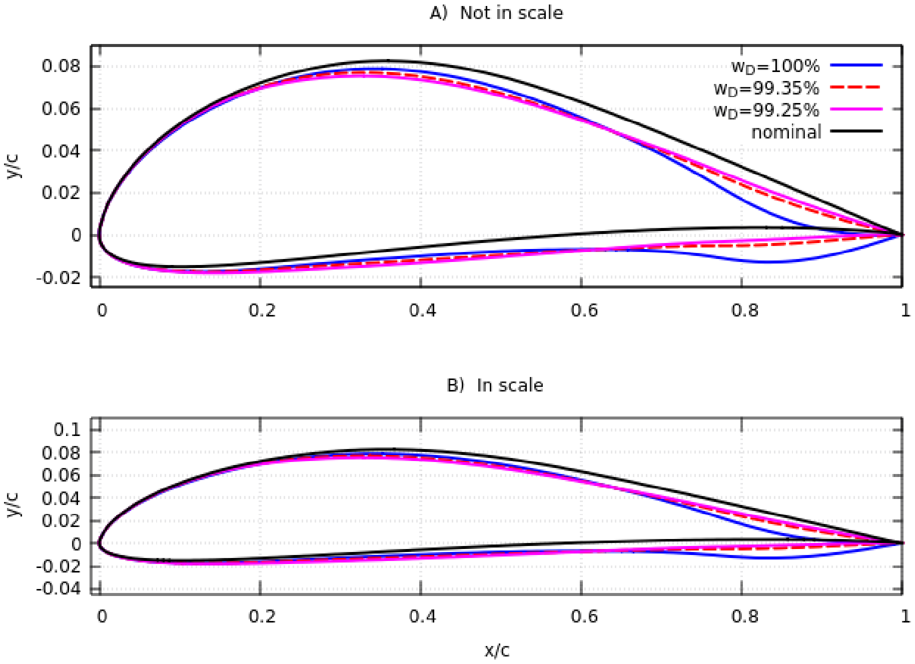


Figure 6.6: *E387 airfoil RDO with manufacturing imperfections: three robust airfoils after 5 optimization cycles, each for a different value of* $w_D = 100, 99.35, 99.25$ % *and for* $\kappa = +2$. *The robust airfoils are visually compared with the initial airfoil, displayed in black, both not in scale (A) and in scale (B).*

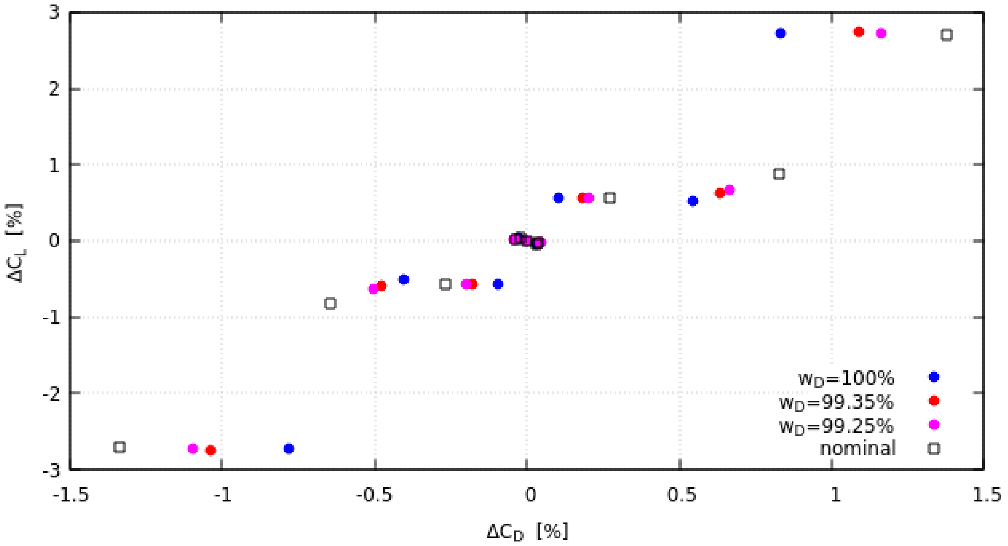


Figure 6.7: *E387 airfoil RDO with manufacturing imperfections: relative divergence ΔC_L, ΔC_D for the three robust airfoil for the three drag weights, as well as for the initial E387 airfoil.*

6.2 Robust Design: TU Berlin Compressor Stator

In this section, a CFD analysis of the *Test Case 3: TU Berlin Turbolab Stator* cascade [13], is held. Generally a compressor cascade is a simplified 2D model of real axial compressor stator blade. The boundary conditions, the computational mesh as well as the solution for the primal problem, without uncertainties, are presented.

Thereafter, the stator is subjected to multi-objective RDO with manufacturing imperfection, for two QoI: the *total pressure losses* (F_{P_t}) between the inlet and outlet boundaries and the *velocity* or *flow angle* (α) of the outlet boundary (as described in Subsection 2.2.2). The KLT software generating the shape uncertainties is configured for $M = 5$ modes, while the Hanning window post-processing function is enabled. The mean value and standard deviation of the KLT uncertain variables (modes) are defined as follows

$$\mu_l = 0\,m \quad , \quad \sigma_l = 7 \cdot 10^{-4}\,m \quad \forall \quad l = 1, 2, ..., M \tag{6.13}$$

6.2.1 Compressor Cascade Initial Conditions and Mesh

The flow around the stator is considered to be steady, incompressible and turbulent. The blade's chord is equal to 0.1876 *meters*. A representation of the blade's intersection, as well and some geometrical properties are displayed in Figure 6.8, taken form [13].

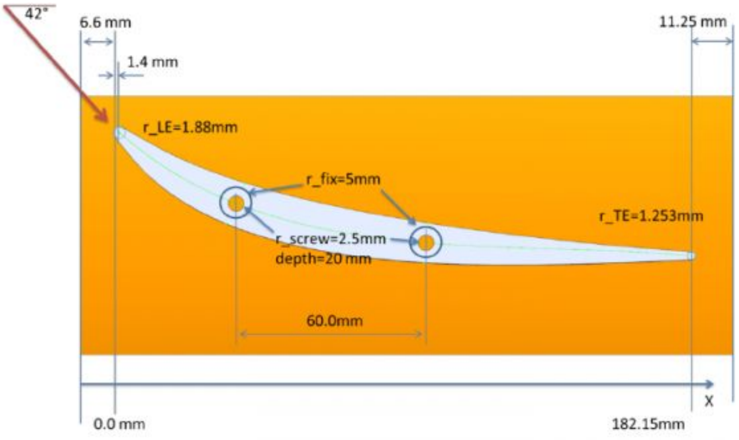


Figure 6.8: *TU Berlin compressor stator intersection.*

The flow initial conditions and properties are assumed to be constant and their values are presented in Table 6.4.

$Inlet\ Velocity$	U_∞	$48\ m/s$
$Inlet\ Velocity\ Angle$	α_1	$-42^{\,o}$
$Kinematic\ Viscocity$	ν	$1.339 \cdot 10^{-5}\ m^2/s$
$Reynolds\ Number$	Re_c	$6.72 \cdot 10^5$

Table 6.4: *The constant initial condition of the TU Belrin compressor stator.*

The cascade flow is solved on a hybrid, 2D and cell-centered mesh, consisting of 80039 quadrilateral and 189 triangular elements , generated through the OpenFOAM© meshers *blockMesh* and *snappyHexMesh*. The mesh is visualized in Figures 6.9 and 6.10.

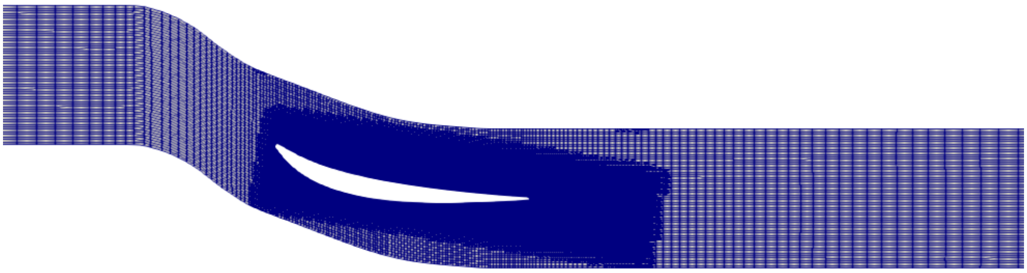


Figure 6.9: *TU Berlin compressor cascade: computational mesh.*

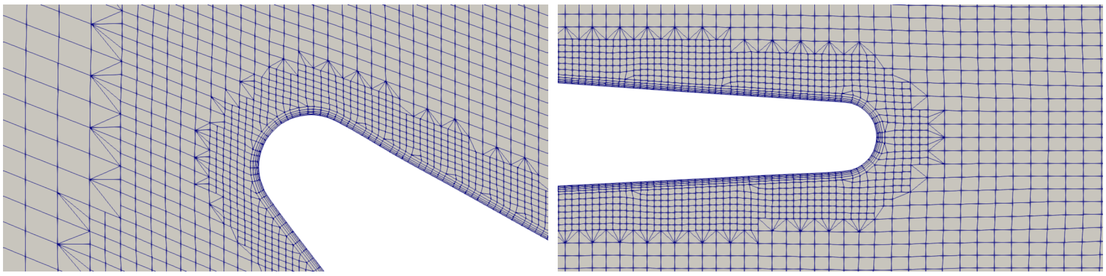


Figure 6.10: *TU Berlin compressor cascade: computational mesh, close up view of the blade's leading (left) and trailing edges (right).*

The turbulence model used in this CFD case, is once more, the Spalart–Allmaras model. The initial conditions for the turbulence model eqs. are: Spalart–Allmaras kinematic viscocity $\tilde{\nu} = 2.793 \cdot 10^{-4}\ m^2/s$ and turbulent kinematic viscosity $\nu_t = 2.678 \cdot 10^{-4}\ m^2/s$. OpenFOAM's *nutUSpaldingWallFunction* High-Re wall function [18] is utilized as wall treatment, given that for all first cell centers off the solid wall, the non-dimensional wall distance amounts to $y^+ < 100$. This claim is verified in Figure 6.11, where the y^+ distributions are plotted.

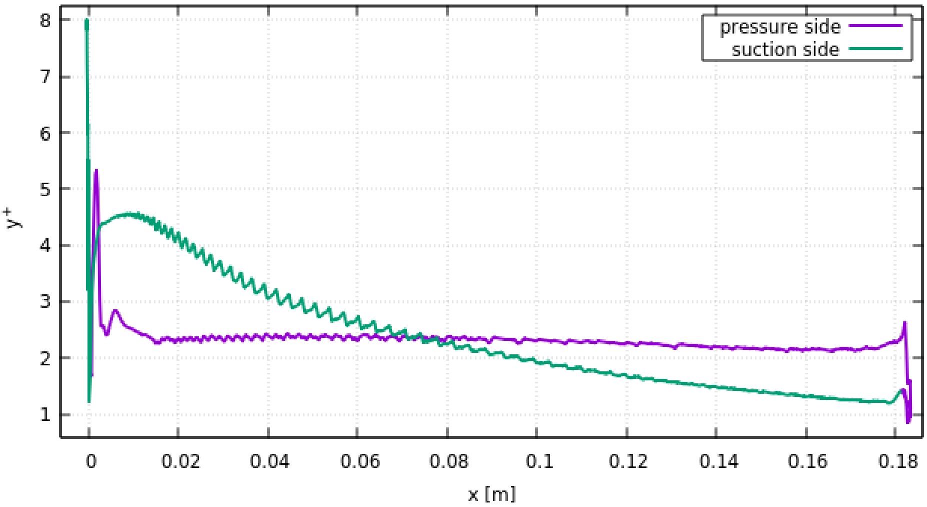


Figure 6.11: *Non-dimensional wall distance y^+, plotted for the pressure and the suction side of the mesh around TU Berlin compressor stator blade.*

6.2.2 CFD Analysis without Uncertainties

The primal equations is solved through use of the executable *simpleFoam*, discretized with 2^{nd} order finite volume schemes. A converged solution is reached in 1469 iterations and the convergence chart of the mean flow variables is presented in Figure 6.12.

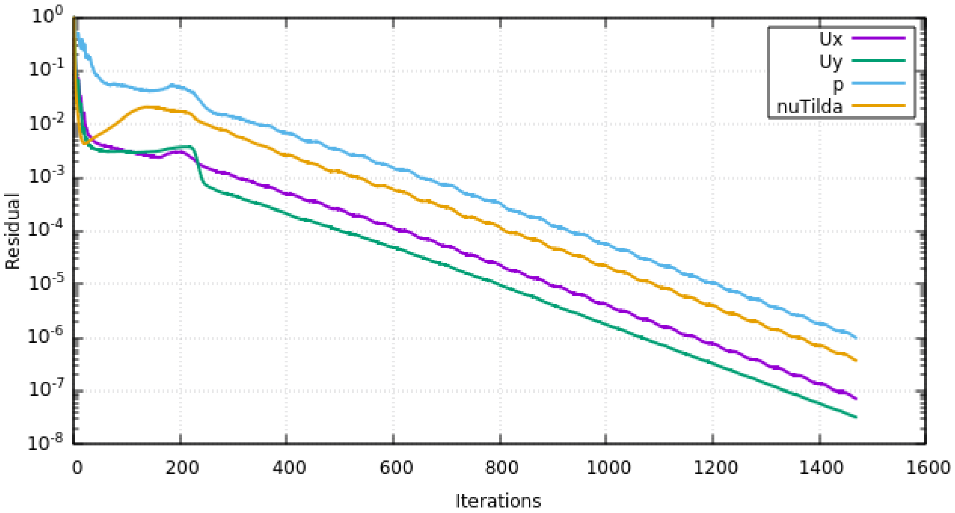


Figure 6.12: *TU Berlin stator cascade: convergence plot of the flow variables. The convergence criterion is set at a residual equal to 10^{-6}, so as not to compromise the solution's accuracy.*

The total pressure losses (F_{P_t}) and the outlet velocity angle (α_2), computed for no uncertainties are displayed in Table 6.5. Furthermore, the static pressure p and the turbulent kinematic viscosity ν_t contours, close to the blade, are visualized in Figure 6.13. Additionally, the velocity magnitude U close to the trailing and leading edges are displayed in Figure 6.14.

TU Berlin Compressor Stator Cascade	
F_{P_t}	$0.109231\ m^5/s^3$
α_2	-2.206^o

Table 6.5: *TU Berlin compressor cascade: total pressure losses and the velocity angle for the flow conditions described in Table 6.4 ($\alpha_1 = -42\,^o$, $Re_c = 672,000$).*

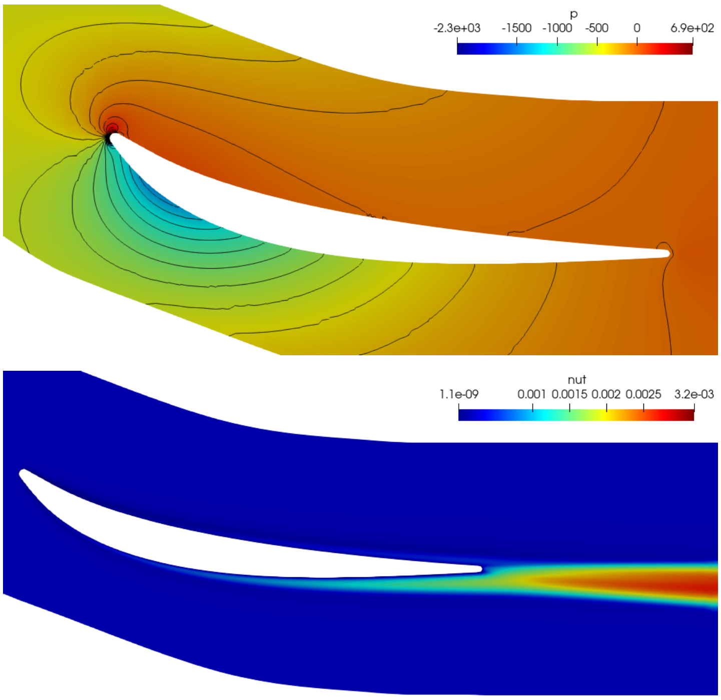


Figure 6.13: *TU Berlin compressor cascade: static pressure p (above) and turbulent kinematic viscosity ν_t contours (below), for the flow conditions described in Table 6.4.*

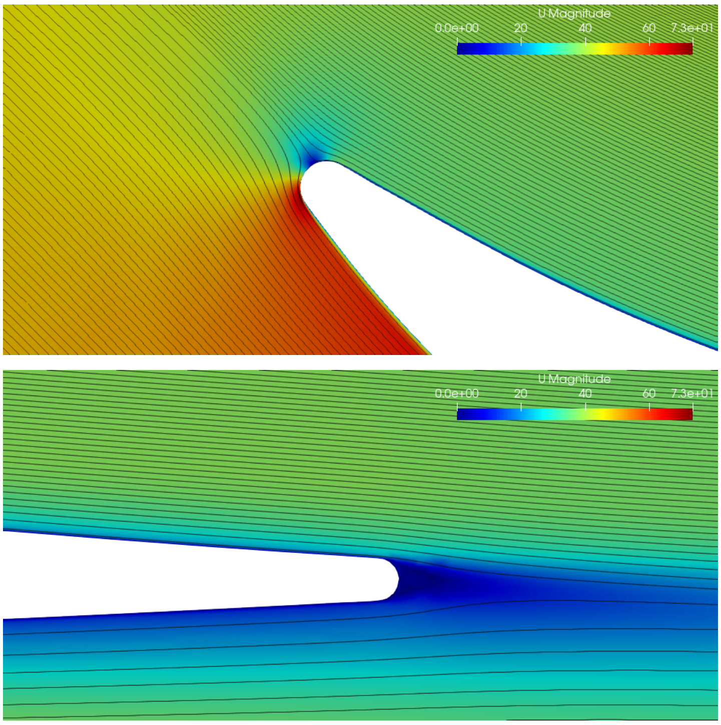


Figure 6.14: *TU Berlin compressor cascade: velocity magnitude U, close up view of the blade's leading (above) and trailing edges (below), with the streamlines also present.*

6.2.3 Multi-Objective RDO: Weighted Objectives

For the purpose of multi-objective RDO with shape imperfections, the QoI (F) is defined as a weighted sum of the pressure losses and the outlet velocity angle, as formulated in (6.14).

$$F = w_p \, F_{p_t} + w_\alpha \, \alpha_2 = w_p \, F_{p_t} - (1 - w_p) \, \alpha_2 \tag{6.14}$$

where $w_p \in [0, 1]$ the total pressure loss weight, while the outlet velocity angle weight is set to $w_\alpha = -(1 - w_p) \in [-1, 0]$.

Generally, some the key purposes of stator blade is to achieve a high flow turning, while keeping the total pressure losses as low as possible. These two functions are quantified though the *flow deviation angle*, formulated as $\theta = \alpha_1 - \alpha_2$ [39], as well as the total pressure losses F_{p_t}. The increase of the flow turn, is accomplished through the maximization of the absolute value of θ. For this specific case, given that $\alpha_1 = -42 \ ^o$ and $|\theta| = |\alpha_1 - \alpha_2| = \alpha_2 + 42 \ ^o$, the maximization of $|\theta|$ is equivalent with the maximization of α_2. Therefore, the outlet velocity angle weight is defined as a negative value, so as to denote the desired maximization of it's stochastic distribution.

In a manner similar to the formulation used in Subsection 6.1.2, the mean value and standard deviation of F, assume the form

$$\mu_F = w_p\, \mu_{F_{Pt}} - (1 - w_p)\, \mu_{\alpha_2} \tag{6.15}$$

$$\sigma_F = w_p\, \sigma_{F_{Pt}} - (1 - w_p)\, \sigma_{\alpha_2} \tag{6.16}$$

Furthermore, the robustness metric is defined as

$$F_R = \mu_F + \kappa\, \sigma_F = w_p\, {F_{pt}}^{(robust)} - (1 - w_p)\, {\alpha_2}^{(robust)} \tag{6.17}$$

where the robust pressure losses (${F_{pt}}^{(robust)}$) and outlet velocity angle (${\alpha_2}^{(robust)}$) are formulated as

$${F_{pt}}^{(robust)} = \mu_{F_{Pt}} + \kappa\, \sigma_{F_{Pt}} \tag{6.18}$$

$${\alpha_2}^{(robust)} = \mu_{\alpha_2} + \kappa\, \sigma_{\alpha_2} \tag{6.19}$$

Accordingly, the derivatives of the mean vale and standard deviation of F w.r.t. the design variables become

$$\frac{\partial \mu_F}{\partial b_n} = w_p\, \frac{\partial \mu_{F_{Pt}}}{\partial b_n} - (1 - w_p)\, \frac{\partial \mu_{\alpha_2}}{\partial b_n} \tag{6.20}$$

$$\frac{\partial \sigma_F}{\partial b_n} = w_p\, \frac{\partial \sigma_{F_{Pt}}}{\partial b_n} - (1 - w_p)\, \frac{\partial \sigma_{\alpha_2}}{\partial b_n} \tag{6.21}$$

Finally, the robust SDs are defined as

$$\frac{\partial F_R}{\partial b_n} = \frac{\partial \mu_F}{\partial b_n} + \kappa\, \frac{\partial \sigma_F}{\partial b_n} = w_p\, \frac{\partial}{\partial b_n} {F_{pt}}^{(robust)} - (1 - w_p)\, \frac{\partial}{\partial b_n} {\alpha_2}^{(robust)} \tag{6.22}$$

where the derivatives of ${F_{pt}}^{(robust)}$ and ${\alpha_2}^{(robust)}$ assume the form

$$\frac{\partial}{\partial b_n} {F_{pt}}^{(robust)} = \frac{\partial \mu_{F_{Pt}}}{\partial b_n} + \kappa\, \frac{\partial \sigma_{F_{Pt}}}{\partial b_n} \tag{6.23}$$

$$\frac{\partial}{\partial b_n} {\alpha_2}^{(robust)} = \frac{\partial \mu_{\alpha_2}}{\partial b_n} + \kappa\, \frac{\partial \sigma_{\alpha_2}}{\partial b_n} \tag{6.24}$$

RDO is preformed for $\kappa = +1$ and six different weights: $w_p = 100, 95, 90, 75, 50, 0\,\%$, in order to produce a variety of robust stator airfoils, with a wide spectrum of different F_{p_t} to a_2 prioritization. Once again the optimization runs for 5 cycles. For the different values of w_p and for each cycle the robust F_{p_t} and a_2, as well as their relative difference ($\%\,\Delta$), are presented in Tables 6.6 and 6.7, respectively, while also in Figures 6.15 and 6.16.

$F_{p_t}^{(robust)}$ $[m^5/s^3]$ $(\%\,\Delta)$						
Cycle	$w_p = 100\,\%$	$w_p = 95\,\%$	$w_p = 90\,\%$	$w_p = 75\,\%$	$w_p = 50\,\%$	$w_p = 0\,\%$
1	0.109838 (0.00)	0.109838 (0.00)	0.109838 (0.00)	0.109838 (0.00)	0.109838 (0.00)	0.109838 (0.00)
2	0.108591 (1.14)	0.108457 (1.26)	0.108677 (1.06)	0.109157 (0.62)	0.109373 (0.42)	0.109500 (0.31)
3	0.107736 (1.91)	0.107527 (2.10)	0.107819 (1.84)	0.108612 (1.12)	0.109025 (0.74)	0.109270 (0.52)
4	0.107113 (2.48)	0.106854 (2.72)	0.107230 (2.37)	0.108178 (1.51)	0.108769 (0.97)	0.109129 (0.65)
5	0.106672 (2.88)	0.106403 (**3.13**)	0.106780 (2.78)	0.107888 (1.78)	0.108609 (1.12)	0.109085 (0.69)

Table 6.6: *TU Berlin stator airfoil RDO with manufacturing imperfections: $F_{p_t}^{(robust)}$ values for 5 optimization cycles, $\kappa = +1$ and six different weights.*

$\alpha_2^{(robust)}$ $[deg]$ $(\%\,\Delta)$						
Cycle	$w_p = 100\,\%$	$w_p = 95\,\%$	$w_p = 90\,\%$	$w_p = 75\,\%$	$w_p = 50\,\%$	$w_p = 0\,\%$
1	−1.29 (0.0)	−1.29 (0.0)	−1.29 (0.0)	−1.29 (0.0)	−1.29 (0.0)	−1.29 (0.0)
2	−1.00 (22.0)	−0.71 (44.8)	−0.62 (51.8)	−0.63 (50.6)	−0.64 (50.1)	−0.65 (49.8)
3	−0.76 (40.8)	−0.20 (84.2)	−0.02 (98.7)	−0.02 (98.7)	−0.02 (98.6)	−0.02 (98.5)
4	−0.54 (58.0)	0.25 (119.5)	0.55 (142.6)	0.58 (144.9)	0.59 (145.8)	0.60 (146.3)
5	−0.34 (73.8)	0.66 (151.4)	1.06 (182.5)	1.14 (188.6)	1.17 (191.4)	1.19 (**192.9**)

Table 6.7: *TU Berlin stator airfoil RDO with manufacturing imperfections: $\alpha_2^{(robust)}$ values for 5 optimization cycles, $\kappa = +1$ and six different weights.*

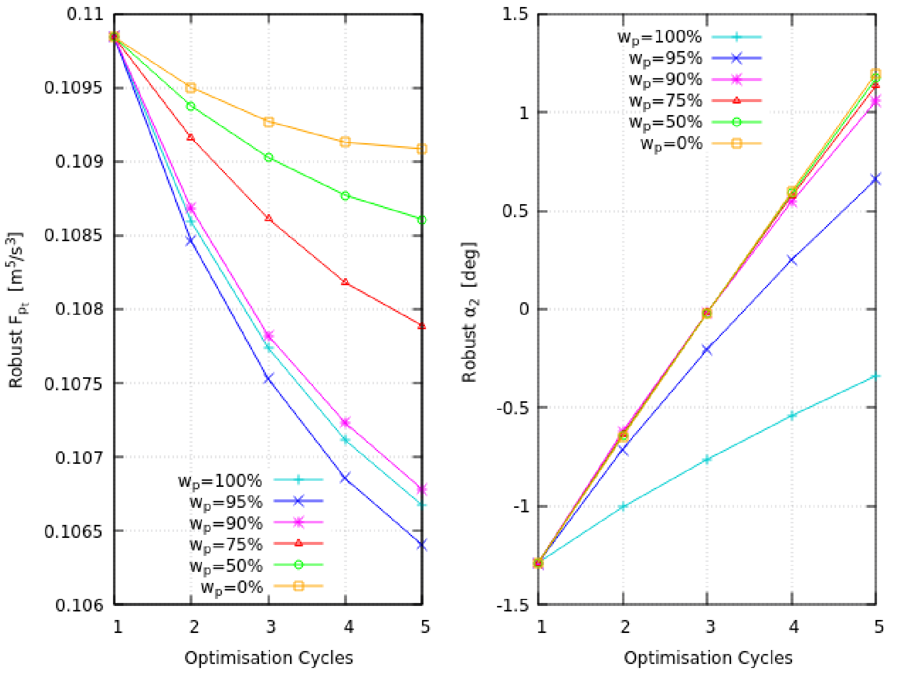


Figure 6.15: *TU Berlin stator airfoil RDO with manufacturing imperfections: $F_{p_t}^{(robust)}$ and $\alpha_2^{(robust)}$ values plotted for 5 optimization cycles, $\kappa = +1$ and six different weights.*

It is clear that, in contrast to Section 6.1, the selected objectives can be "combined", through the use of the weighted robustness metric function (eq. (6.17)), producing

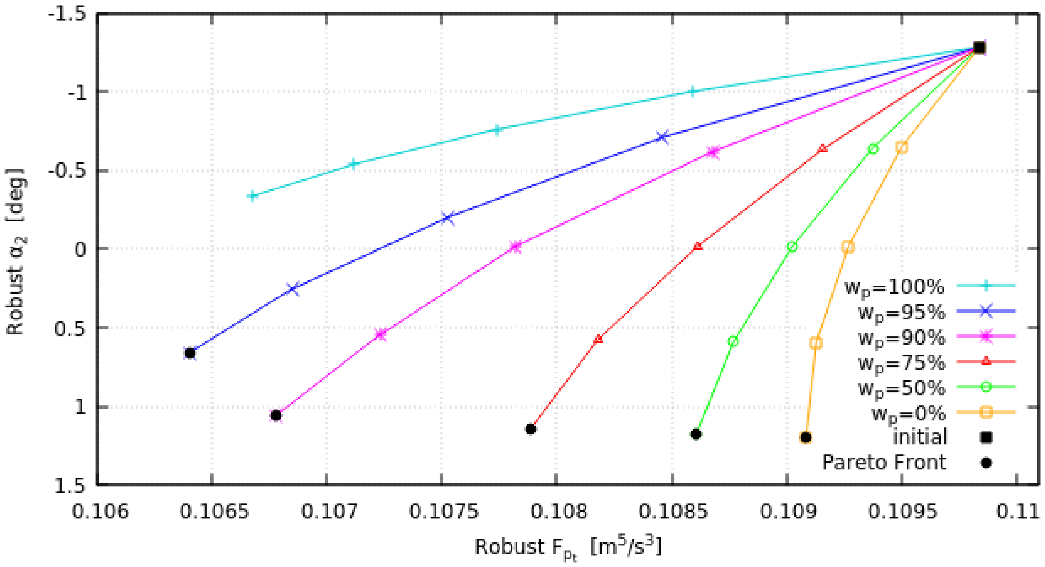


Figure 6.16: *TU Berlin stator airfoil RDO with manufacturing imperfections: $F_{p_t}^{(robust)}$ - $\alpha_2^{(robust)}$ plot for six different weights. All final robust solutions, except the one for $w_p = 100$ %, are dominant upon all others, constituting the Pareto front for this RDO case.*

robust “hybrid” stators that display enhanced results in both objectives. Specifically, for weights $w_p = 95,\ 90,\ 75,\ 50$ %, the final geometries display simultaneously the highly cambered trailing edge (which leads to the increase of α_2) as well as the reduced thickness (which lowers the total pressure losses). This is further supported, by the fact that the $w_p = 95$ % final robust blade sports the lowest robust pressure loss of all other robust blades, minimizing its value by 3.13 %, thus overcoming even the $\Delta F_{p_t} = 2.88$ % of the $w_p = 100$ % robust blade. The hybrid performance of the final robust blades can be observed in Figure 6.17, where the blade’s contours are displayed for the different values of w_p. The airfoils produced for weights 50 % $\leq w_p \leq$ 95 %, display both a reduced thickness, which lowers the pressure losses, as well as as greater camber near the trailing edge, which leads to a greater outlet flow angle. Besides, as detected in Figure 6.16, the final solution for the weights $w_p = 95,\ 90,\ 75,\ 50,\ 0$ %, generate robust F_{p_t} and a_2 that are dominant upon all other solutions, hence representing the Pareto front of this RDO case, w.r.t. the robust objectives.

Additionally, the mean value of F_{p_t} and α_2 are presented in Tables 6.8 and 6.9, respectively , while their standard deviation is displayed Tables 6.10 and 6.11.

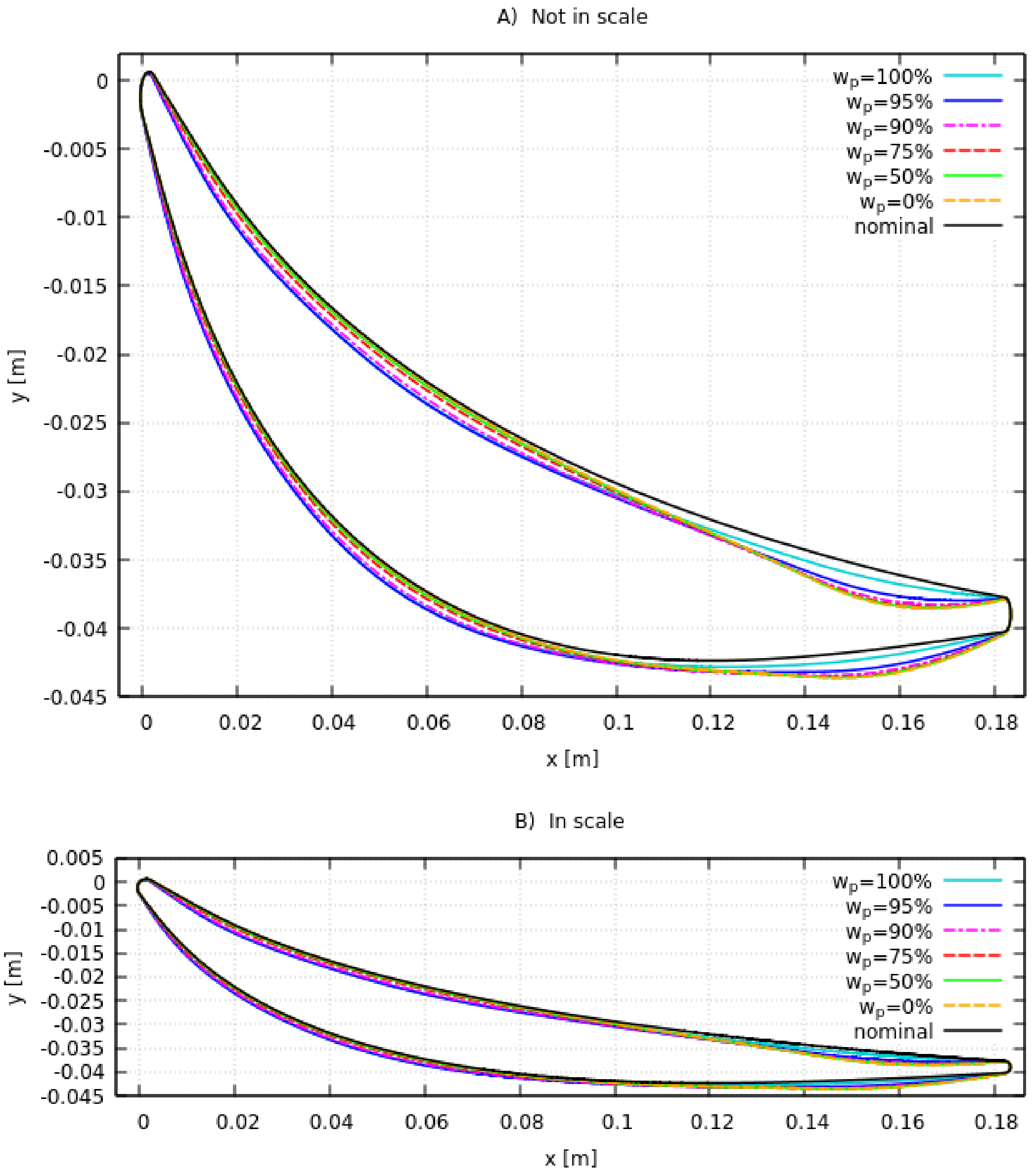


Figure 6.17: *TU Berlin stator airfoil RDO with manufacturing imperfections: the initial (black) and six robust stator blades after 5 optimization cycles, each for a different value of w_p and for $\kappa = +1$, visualized both not in scale (A) and in scale (B).*

$\mu_{F_{Pt}}$ $[m^5/s^3]$						
Cycle	$w_p = 100\,\%$	$w_p = 95\,\%$	$w_p = 90\,\%$	$w_p = 75\,\%$	$w_p = 50\,\%$	$w_p = 0\,\%$
1	0.109338	0.109338	0.109338	0.109338	0.109338	0.109338
2	0.108207	0.108115	0.108336	0.108777	0.108977	0.109095
3	0.107422	0.107288	0.107585	0.108337	0.108721	0.108947
4	0.106862	0.106712	0.107091	0.107996	0.108552	0.108884
5	0.106447	0.106301	0.106721	0.107772	0.108458	0.108899

Table 6.8: *TU Berlin stator airfoil RDO with manufacturing imperfections: mean value of F_{p_t} for 5 optimization cycles, $\kappa = +1$ and six different weights.*

μ_{a_2} $[deg]$						
Cycle	$w_p = 100\,\%$	$w_p = 95\,\%$	$w_p = 90\,\%$	$w_p = 75\,\%$	$w_p = 50\,\%$	$w_p = 0\,\%$
1	-2.21	-2.21	-2.21	-2.21	-2.21	-2.21
2	-1.93	-1.63	-1.54	-1.56	-1.57	-1.57
3	-1.69	-1.13	-0.94	-0.94	-0.94	-0.94
4	-1.46	-0.67	-0.37	-0.34	-0.33	-0.32
5	-1.26	-0.26	0.14	0.22	0.26	0.28

Table 6.9: *TU Berlin stator airfoil RDO with manufacturing imperfections: mean value of α_2 for 5 optimization cycles, $\kappa = +1$ and six different weights.*

$\sigma_{F_{Pt}}$ $[m^5/s^3]$						
Cycle	$w_p = 100\,\%$	$w_p = 95\,\%$	$w_p = 90\,\%$	$w_p = 75\,\%$	$w_p = 50\,\%$	$w_p = 0\,\%$
1	$5.006 \cdot 10^{-4}$	$5.006 \cdot 10^{-4}$	$5.006 \cdot 10^{-4}$	$5.006 \cdot 10^{-4}$	$5.006 \cdot 10^{-4}$	$5.006 \cdot 10^{-4}$
2	$3.842 \cdot 10^{-4}$	$3.427 \cdot 10^{-4}$	$3.421 \cdot 10^{-4}$	$3.809 \cdot 10^{-4}$	$3.963 \cdot 10^{-4}$	$4.048 \cdot 10^{-4}$
3	$3.146 \cdot 10^{-4}$	$2.399 \cdot 10^{-4}$	$2.347 \cdot 10^{-4}$	$2.745 \cdot 10^{-4}$	$3.051 \cdot 10^{-4}$	$3.235 \cdot 10^{-4}$
4	$2.510 \cdot 10^{-4}$	$1.422 \cdot 10^{-4}$	$1.388 \cdot 10^{-4}$	$1.822 \cdot 10^{-4}$	$2.176 \cdot 10^{-4}$	$2.453 \cdot 10^{-4}$
5	$2.261 \cdot 10^{-4}$	$1.026 \cdot 10^{-4}$	$5.972 \cdot 10^{-5}$	$1.168 \cdot 10^{-4}$	$1.514 \cdot 10^{-4}$	$1.857 \cdot 10^{-4}$

Table 6.10: *TU Berlin stator airfoil RDO with manufacturing imperfections: standard deviation of F_{p_t} for 5 optimization cycles, $\kappa = +1$ and six different weights.*

σ_{a_2} $[deg]$						
Cycle	$w_p = 100\,\%$	$w_p = 95\,\%$	$w_p = 90\,\%$	$w_p = 75\,\%$	$w_p = 50\,\%$	$w_p = 0\,\%$
1	0.92122	0.92122	0.92122	0.92122	0.92122	0.92122
2	0.92517	0.92465	0.92425	0.92442	0.92435	0.92441
3	0.92499	0.92343	0.92277	0.92252	0.92267	0.92264
4	0.92429	0.92155	0.92029	0.91964	0.91931	0.91923
5	0.92387	0.92026	0.91776	0.91673	0.91627	0.91607

Table 6.11: *TU Berlin stator airfoil RDO with manufacturing imperfections: standard deviation of α_2 for 5 optimization cycles, $\kappa = +1$ and six different weights.*

In Tables 6.6 and 6.7, it is already established that the overall reduction of the robust pressure losses are relatively small when compared to the reduction of the robust outlet flow angle, given that for $w_p = 0\,\%$ it is increased by a tremendous 2.5^o after five cycles. Accordingly, for the other weights similar, yet lower, relative differences can be found. At first glance, this reveals that the outlet velocity angle is greatly affected by geometry changes.

The contents of Tables 6.8, 6.9, 6.10 and 6.11 are visualized, in total, in Figure 6.18. The data in Figure 6.18 provides even further insight in this matter. It can be deduced that the great increase of ${a_2}^{(robust)}$ is mainly due to the increase of the mean value of a_2, by a maximum of 2.5^o, approximately, for $w_p = 0\%$. Furthermore, the reduction of σ_{a_2}, if achieved, is seemingly small, with the maximum decrease being approximately 0.005^o, again, for $w_p = 0\,\%$. This signifies that while the maximization of the mean value of a_2 yields a significant increase, the decrease in its standard deviation is minute, even for w_p.

On the contrary, the reduction in the mean pressure losses generally small, following the same pattern with the respective robust value. Its highest reduction is 2.78 % achieved for $w_p = 95\,\%$. Still, the standard deviation of the total pressure losses, is greatly mitigated for every weight value, with the highest being 88.1% for $w_p = 90\%$.

Finally, in Figure 6.19 the relative deviation of F_{p_t} and the absolute deviation of α_2 are presented, for 32 KLT-generated imperfect renderings of the six final robust blades as well as for the initial. In Figure 6.19, one can distinguish that the outlet velocity a_2 displays a apparently high deviation. The maximum value of this absolute deviation is approximately $\Delta a_2 = \pm 1.5^o$, which is seemingly unchanged for different weight values. The maximum absolute deviation is 1.58^o and corresponds to the initial blade, while its value decreases with the decrease of w_p, with the lowest maximum absolute deviation being 1.53^o (1.6 % reduction) for $w_p = 0\,\%$. The relatively high deviation is expected, given that σ_{a_2} has, for the most part, the same order of magnitude with its respective mean value, as observed form Tables 6.11 and 6.9. This, also, denotes that the outlet flow angle is greatly affected by the manufacturing imperfection created though KLT. Moreover, the small reduction of the relative deviation of a_2 is also anticipated, due to the, aforementioned, low reduction of σ_{a_2}. On the other hand, when it comes to the total pressure losses F_{p_t}, a high decrease in its maximum relative deviation is expected, because of the high reduction of $\sigma_{F_{p_t}}$, previously, documented. The highest F_{p_t} maximum relative deviation is equal to 1.07 % can be found for the initial TU Berlin blade, while the lowest is equal to 0.32 % (70.4 % reduction) produced for $w_p = 90\,\%$, as expected. Otherwise, the overall influence of the KLT shape imperfections on the total pressure losses is relatively small, which is also concluded by the fact that the standard deviation of F_{p_t} is two orders of magnitude lower that its respective mean value, as displayed in Tables 6.10 and 6.8.

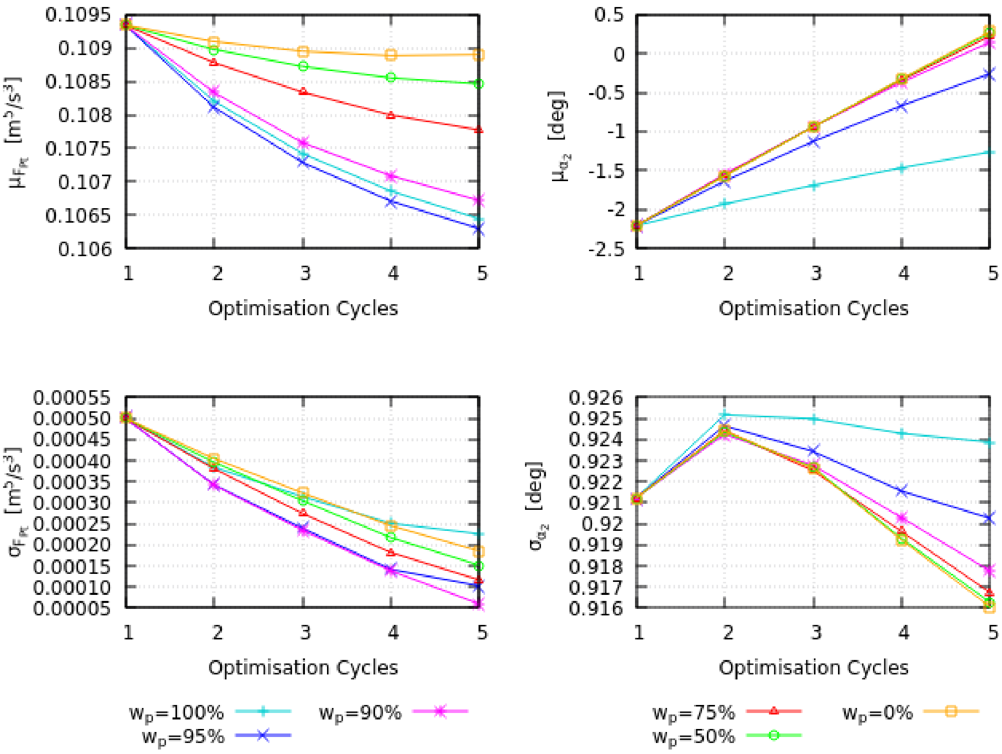


Figure 6.18: *TU Berlin stator airfoil RDO with manufacturing imperfections: mean value and standard deviation of F_{p_t} and α_2 plotted for six different weights.*

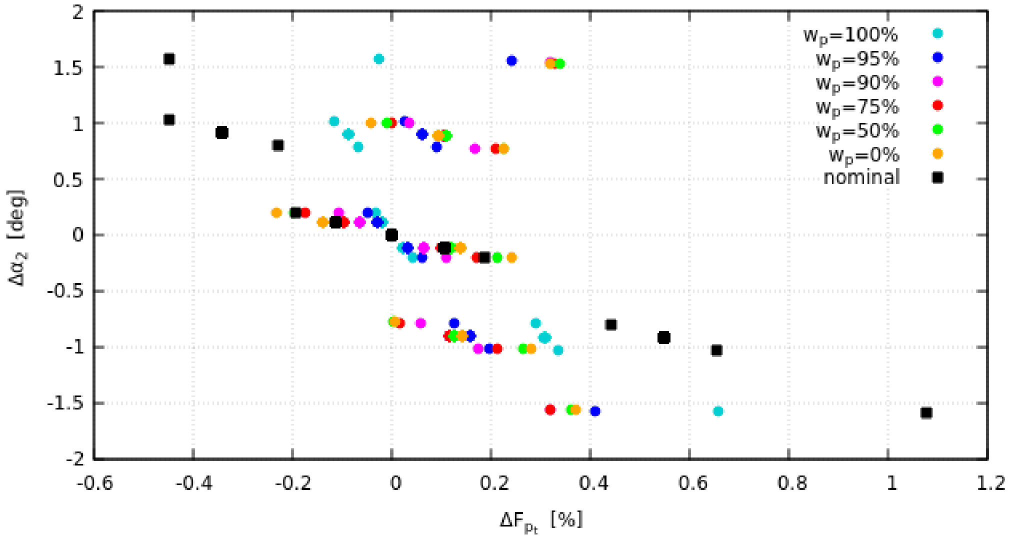


Figure 6.19: *TU Berlin stator airfoil RDO with manufacturing imperfections: absolute deviation $\Delta\alpha_2$ and relative deviation ΔF_{p_t} for the six final robust blades and initial stator blade.*

Chapter 7

Summary and Suggestions

7.1 Summary - Conclusion

In this diploma thesis, a KLT-implementing software is developed, in order to generate imperfect renderings of 2D geometries. This software alongside with OpenFOAM's *simpleFoam* and *adjointOptimisationFoam* solvers are integrated into an in-house code employing the niPCE method, so as to perform aerodynamic UQ and deterministic RDO, respectively, on 2D bodies with shape imperfections.

The niPCE theory, implemented into the according niPCE in-house software, is formulated in order to perform single and multi-dimensional UQ w.r.t. the user-defined QoI, for problems with stochastic input variables following normal distributions. Therefore, for the computation of the mean value and standard deviation of the selected QoI, GQ numerical integration with Hermite orthogonal polynomial is employed. Both Full and Smolyak Sparse grids of Gauss Nodes needed for the integration can be selected, the later proving especially useful for mitigating the method's computational cost of problems with a large number of uncertain variables ($M \geq 4$). Additionally, the niPCE code is, also, adjusted to conduct gradient-based RDO, by performing UQ to the SDs (computed though the adjoint solver), in order to produce the derivatives of the aforementioned statistical moments w.r.t. the design variables a.k.a. the robust SDs. Thus, the computation the robustness metric and its derivatives, a.k.a. the robust SD (in accordance with the DFSS approach), can be achieved.

The continuous adjoint method, developed by PCOpt/NTUA and employed in this thesis, is used to compute the SDs needed for the computation of the robust SDs. The Enhanced-SI formulation is used, which yields an accurate prediction of the SDs for a relatively lower computational cost, when compared with the alternative FI approach.

The KLT software is designed to generate stochastic perturbations on any 2D surface of any length and geometry, thus providing the ability to render imperfect recreations of a wide variety of shapes, such as airfoils or turbumachinery blades. The user can define the number M, as well as the mean value and standard deviation of the KLT uncertain variables, thus regulating the oscillation and the range of the KLT-generated stochastic perturbations, respectively. Moreover, after an application on a flat surface, it is observed the greater the number of M, the more oscillating are the yielded perturbations. From the application of the KLT software, on the NACA 0012 airfoil it is deduced that a middle ground solution for the number of uncertain variables, exhibiting both enough KLT imperfection complexity and a sustainable computational cost is $M = 5$.

Once the aforementioned integration is complete, the coupled software is, firstly, utilized to perform UQ on a NACA 0012 airfoil with manufacturing imperfections, regarding its aerodynamic coefficients (C_L, C_D). Five KLT shape uncertain variables ($M = 5$) and Smolyak Sparse grid GHQ integration are used, while a parametric analysis is held w.r.t. the chaos order k used, while all results are compared with corresponding results produced through the MC method for 5000 samples. From this analysis, it is concluded that the precision of niPCE-computed UQ for geometries with stochastic imperfections, is acceptable (mean relative error lower than 3 %) for lower chaos order values ($k \leq 3$), while the higher orders yield an unacceptable computational cost, even for Sparse niPCE integration grids. Secondly, UQ is executed on the E387 airfoil with manufacturing uncertainties, for the computation of the robust SDs, through the MC and niPCE methods, the later by utilizing both Full as well as Sparse grid integration. Both niPCE results for the robust SD are seemingly accurate, justifying the adoption of the Smolyak grid integration for the following RDO, given that it yields a reduced cost and practically the same accuracy when compared with the Full grid results.

Furthermore, in order to accomplish deterministic shape RDO, certain mesh displacement strategy must be employed and for that a the Volumentic B-Spline mesh paramaterization subroutine of PCOpt/NTUA's code *adjointOptimisationFoam*, is isolated and integrated into the niPCE-KLT-Adjoint coupled software. This way, after each optimization cycle, the pararemeterized mesh region is displaced according to the previously computed robust SDs and according to the *Steepest Descent* design variable renewal method.

The KLT-niPCE-Adjoint coupled software is then executed on the E387 airfoil and the TU Berlin compressor stator cascade, so as to perform RDO with shape uncertainties. The E387 airfoil, is subjected, initially, to single-objective robust drag minimization, i.e. the minimization of its drag stochastic performance. This RDO analysis is held for different values of the DFSS parameter κ. A conclusions drawn from this analysis, is that, after 5 RDO cycles, the mean value and standard deviation of the exerted drag can be reduced by 14 % and 35 %, respectively. Also, it is deduced that the results for the three values of κ are virtually indistinguishable, due to the fact that the derivatives of the standard deviation of the drag coefficient are negligible, when compared to its respective mean value. In addition, the airfoil is subjected to multi-objective RDO, namely through the minimization of a weighted objective function containing both the drag and lift coefficients. The analysis is held for three different sets of weights, each signifying a different lift to drag prioritization. The RDO process is successful in producing more balanced robust airfoils, with optimized results in both robust drag and lift objectives. Overall, though, it is perceived that the minimization of the stochastic variation of an airfoil's force coefficient is achievable, even though the KLT-modeled manufacturing uncertainties have a minor influence on the forces exerted on a airfoil, given that their respective maximum relative deviation is lower that 3 %.

Finally, multi-objective RDO with manufacturing uncertainties is held on the TU Berlin compressor stator cascade, w.r.t. to the total pressure losses and the outlet velocity angle. The two quantities are again arranged into a weighted objective function, opting to minimize the mean value of the pressure losses, maximizing the mean value of the outlet flow angle, while minimizing the standard deviation of both objectives. The RDO analysis is performed for six different weight values and the two objectives are proven to be compatible, yielding "hybrid" results sporting both the reduced thickness and highly cambered trailing edge of their respective single-objective final robust geometries. Ultimately, from this analysis, it is concluded that the outlet flow angle is greatly affected by the shape changes and therefore produces a relatively high standard deviation, with volatile responses to the KLT-generated uncertainties, which cannot be significantly reduced by the proposed RDO method. Therefore, while the maximization of its mean value is achievable by a maximum of approximately 2.5^o, the greatest reduction of its standard deviation achieved is 0.005^o. On the other hand, the total pressure losses exhibit a minor sensitivity to the shape uncertainties, similarly to the force coefficients, but their standard deviation can be significantly reduced by approximately 90 %, while their respective mean value is decreased by a maximum of approximately 3 %.

Overall, the proposed method yields the expected results concerning the execution of aerodynamic UQ and RDO of 2D geometries with shape uncertainties. The method has been verified on two isolated airfoils and a compressor stator cascade, for three distinct objective functions. Its computational cost is significant and it scales with the number of RDO cycles needed, but it is still lower than the expected cost (lower number of cycles and less evaluations per cycle) of other such methods, which employ stochastic RDO, i.e. *Evolutionary Algorithms*.

7.2 Suggestions for Future Research

The following proposition are made to fuel ideas for the further development of the this work, in the future:

1. In this work, only geometrical uncertain variables following normal stochastic distributions are taken into account and, thus, only their corresponding Hermite family of orthogonal polynomials are used for the numerical GQ integration of the niPCE coefficients. The *generalized Polynomial Chaos* theory, that can be found in [9], is suggested in order to widen the spectrum of different stochastic distributions that can be used as inputs.

2. The KLT-niPCE-Adjoint algorithm could be used to perform aerodynamic UQ and RDO with, both, flow and shape uncertainties. Such a computational process could evaluate and optimize the aerodynamic stochastic performance of geometries for any, user-defined, stochastic input.

3. The KLT shape imperfection model can further be expanded, in order to encompass also 3D geometries. According to the KLT theory, the generation of 3D stochastic perturbations, is feasible and this way the evaluation of manufacturing imperfections can be performed on complex 3D geometries.

4. The only design variable method, implemented in this thesis, is the *Steepest Descent* method. The reason behind this, is that this method provides a fairly easy formulation, requiring only 1st order sensitivity derivatives (robust or not) for the update of the design variables. Additionally this method can only cope with optimization with unconstrained objectives. It is suggested that the niPCE code, should be improved so as to to receive both 2nd order sensitivity derivatives as well as objective constraints and compute robust 2nd order derivatives and robust constraints, giving it thus the ability to work in conjunction with other design variables update methods, such as the *BFGS* or the *Constraint Projection* methods. Especially the *BFGS* method, with its high efficiency, could work perfectly alongside the coupled niPCE-KLT-Adjoint algorithm, for robust design, given that it often reaches an optimal solution a lot faster than other methods, when it is already established that each RDO cycle is quite costly.

5. Other UQ methods could be employed other than the niPCE, for the evaluation of stochastic QoI as well as their derivatives w.r.t. the design variables. Without changing the gradient-based method of optimization, in this case the continuous adjoint method, it is proposed to adopt another stochastic, e.g. the *intrusive* PCE method or the deterministic *Method of Moments* [19], [3], [4] could be implemented for the formulation of the statistical moments of the aerodynamic performance of a shape with KLT-produced geometrical uncertainties. Though the iPCE method lacks the generality of the niPCE, it makes up in higher accuracy and lower computational cost. Additionally, the *Method of Moments* could be utilized for UQ or RDO with KLT shape uncertainties, given that

this method would greatly reduce the computational cost, while it could be successfully coupled with 2nd order Quasi-Newton design variables update methods, such as the aforementioned *BFGS*, due the higher order derivatives formulation yielded from this method. Finally other niPCE variants found in [3] such as the niPCE-Regression or the niPCE-Regression-Adjoint could be utilized alongside the KLT model, for reasons of cost mitigation, especially given that elevated numbers M are employed.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Εργαστήριο Θερμικών Στροβιλομηχανών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής
& Βελτιστοποίησης

Μη-επεμβατικό Ανάπτυγμα Πολυωνυμικού Χάους για Αεροδυναμικό Στιβαρό Σχεδιασμό υπό Κατασκευαστικές Ατέλειες

Εκτενής Περίληψη Διπλωματικής Εργασίας

Σέργιος Βιλλέτ

Επιβλέπων
Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, Φεβρουάριος 2022

Εισαγωγή

Στην παρούσα διπλωματική εργασία αναπτύσσεται λογισμικό προσομοίωσης γεωμετρικών ατελειών κάνοντας χρήση του μετασχηματισμού *Karhunen-Loève Transform* (KLT). Το λογισμικό KLT μαζί με τον επιλύτη συνεχούς συζυγούς μεθόδου της ΜΠΤΥΡ&Β/ΕΜΠ, σε περιβάλλον OpenFOAM©, εντάσσονται σε προϋπάρχον λογισμικό *μη-επεμβατικού Αναπτύγματος Πολυωνυμικού Χάους* (niPCE), με τις κατάλληλες τροποποιήσεις για την εκτέλεση *Ποσοτικοποίησης Αβεβαιότητας* και αιτιοκρατικού *Στιβαρού Σχεδιασμού* σε διδιάστατες γεωμετρίες με κατασκευαστικές ατέλειες.

Η θεωρία niPCE αξιοποιείται για την αξιολόγηση της στοχαστικής αεροδυναμικής συμπεριφοράς ενός σώματος, υπολογίζοντας τη μέση τιμή και την τυπική απόκλιση κάποιας Ποσότητας Ενδιαφέροντος (QoI) $F(\vec{b},\vec{c})$, η οποία αποτελεί συνάρτηση τόσο των μεταβλητών σχεδιασμού ($b_n\ ,\ \ n=1,2,...,N$) όσο και των αβέβαιων μεταβλητών ($c_m\ ,\ \ m=1,2,...,M$). Επίσης ο αιτιοκρατικός Στιβαρός Σχεδιασμός για τη βελτιστοποίηση της Μετρικής Στιβαρότητας F_R (συνάρτηση-στόχος) απαιτεί την εύρεση των παραγώγων της ως προς τις μεταβλητές σχεδιασμού, δηλαδή των στιβαρών παραγώγων ευαισθησίας $\delta F_R/\delta b_n$, διαδικασία που, επίσης, υλοποιείται μέσω του niPCE. Η συνεχής συζυγής μέθοδος βελτιστοποίησης χρησιμοποιείται στον αιτιοκρατικό *Στιβαρό Σχεδιασμό* για την εύρεση των παραγώγων ευαισθησίας (SDs), δηλαδή των παραγώγων της συνάρτησης στόχου ως προς τις μεταβλητές σχεδιασμού που χρειάζονται για τον υπολογισμό των παραγώγων ευαισθησίας της μετρικής στιβαβότητας (robust SDs). Οι μεταβλητές σχεδιασμού αποτελούν τα ελεύθερα σημεία ελέγχου του παραμετροποιημένου πλέγματος μέσω Volumetric B-Splines. Τέλος η μέθοδος που υιοθετείται για την ανανέωση των σημείων ελέγχου είναι η μέθοδος της *Απότομης Καθόδου.*

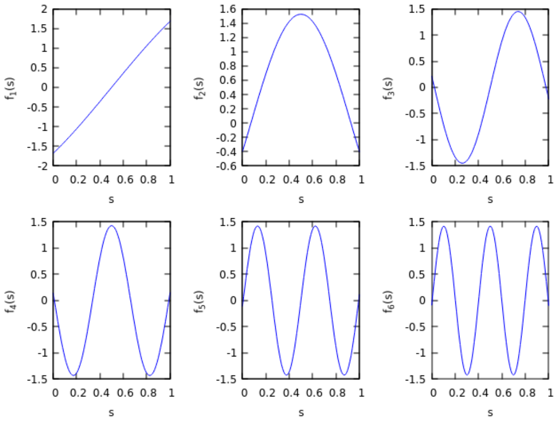
Μοντελοποίηση Κατασκευαστικών Ατελειών

Οι γεωμετρικές ατέλειες που μπορεί να προκύψουν στην επιφάνεια μηχανικών στοιχείων που αλληλεπιδρούν με ρευστά, είτε κατά την κατασκευή τους είτε κατά τη λειτουργία τους (π.χ. επικαθίσεις), μοντελοποιούνται μέσω του μετασχηματισμού KLT. Οι στοχαστικές αυτές διαταραχές παράγονται μέσω ενός πεπερασμένου αριθμού ιδιοτιμών λ_n και ιδιοσυναρτήσεων f_n, οι οποίες συσχετίζονται μέσω της ολοκληρωτικής εξίσωσης (7.1) που περιλαμβάνει τον εκθετικό *Πυρήνα Συνδιακύμανσης* (7.2)

$$\int_D C(s_1,s_2)f_n(s_2)ds_2=\lambda_n f_n(s_1) \tag{7.1}$$

$$C(s_1,s_2)=\sigma^2 e^{-\frac{|s_1-s_2|}{l}} \tag{7.2}$$

όπου s_1, $s_2\in D$, l το μήκος συσχέτισης των διαταραχών και σ η τυπική απόκλιση του πυρήνα. Στην εργασία αυτή, δεδομένου ότι δε δίνονται τα εμπειρικά δεδομένα ατελειών για 2Δ αεροδυναμικά σώματα, λ.χ. μεμονωμένες αεροτομές ή αεροτομές πτερυγώσεων, προτείνονται οι εξής παραδοχές: $D=[0,s_{max}]$ όπου s_{max} η περίμετρος του σώματος, $\sigma=1$ και $l=s_{max}$. Οι έξι πρώτες ιδιοσυναρτήσεις για τον εκθετικό πυρήνα παρατίθενται στο Σχήμα 7.1.



Σχήμα 7.1: *Πρώτες έξι ιδιοσυχνότητες $f_n(s)$ για τον εκθετικό πυρήνα* (7.2), *για* $s \in= [0,\ 1]$, $l = 1$ *και* $\sigma = 1$.

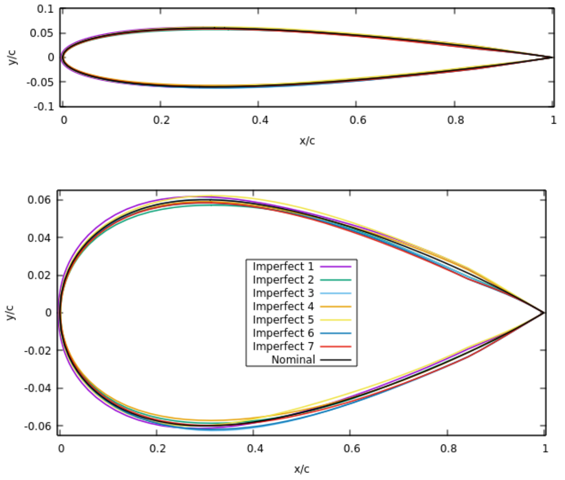
Λαμβάνοντας υπόψη τα παραπάνω, οι ιδιοτιμές και οι ιδιοσυναρτήσεις προκύπτουν από την επίλυση της εξίσωσης (7.1) και έτσι οι στοχαστικές διαταραχές υπερτίθενται στην αρχική γεωμετρία ($\bar{X}(s)$) κατά την κάθετη διεύθυνση σε σχέση με την επιφάνεια, ως εξής:

$$X(s,\vec{c}) = \bar{X}(s) + C_{Hann}(s)\sum_{n=1}^{M}\sqrt{\lambda_n}c_n f_n(s)\vec{n}(s) \tag{7.3}$$

όπου $s \in [0, s_{max}$ η εφαπτομενική συντεταγμένη στην επιφάνεια του διδιάστατου σώματος, c_n οι στοχαστικές μεταβλητές, M το πλήθος των συναρτήσεων βάσης και των στοχαστικών μεταβλητών, $\vec{n}(s)$ το κάθετο διάνυσμα και $C_{Hann}(s)$ μία συνάρτηση Hanning window που μπορεί να χρησιμοποιηθεί, προαιρετικά, για την απόσβεση της KLT διαταραχής κοντά στην ακμή εκφυγής, ώστε να παραμένει μυτερή.

Παρατηρείται ότι όσο μεγαλύτερο είναι το πλήθος των ιδιοσυναρτήσεων M τόσο μεγαλύτερες και πιο πολύπλοκες είναι οι παραγόμενες διαταραχές, ενώ τόσο μεγαλύτερο είναι το υπολογιστικό κόστος κατά την Ποσοτικοποίηση Αβεβαιότητας ή το Στιβαρό Σχεδιασμό. Τελικά επιλέγεται $M = 5$, καθώς αποτελεί μία μέση λύση μεταξύ της πολυπλοκότητας των ατελειών και του υπολογιστικού κόστους, ενώ ταυτόχονα ικανοποιεί το κριτήριο (;;) για ένα πλήθος αεροτομών: $\frac{\sum_{n=1}^{M}\lambda_n}{\sum_n \lambda_n} \geq 0.96$.

Τελικά επτά αποδόσεις της αεροτομής NACA 0012 με γεωμετρικές ατέλειες, παρουσιάζονται στο Σχήμα 7.2, για $M = 5$ και για τη συνάρτηση Hanning window ενεργοποιημένη στο 15 % της χορδής, παράγοντας επτά διαφορετικά διανύσματα αβέβαιων μεταβλητών $\vec{xi_n}$ μέσω γεννήτριας τυχαίων αριθμών (RNG).



Σχήμα 7.2: *Η αρχική αεροτομή NACA 0012 (μαύρο) καθώς και επτά αποδόσεις της με ατέλειες (άλλα χρώματα), παραγόμενες μέσω KLT, τόσο σε πραγματική (πάνω) όσο και σε μη πραγματική κλίμακα (κάτω).*

Μη-επεμβατικό Ανάπτυγμα Πολυωνυμικού Χάους

Σύμφωνα με τη θεωρία του Πολυωνυμικού Χάους, μία Ποσότητα Ενδιαφέροντος $F = F(\vec{b},\vec{c})$ προσεγγίζεται ως γραμμικός συνδυασμός ορθοκανονικών πολυδιάστατων πολυνύμων $\widetilde{\psi}$, ως εξής:

$$F(\vec{b},\vec{c}) \approxeq \sum_{i=0}^{N_{cut}} a_i \widetilde{\psi}_i(\vec{c}) \ , \quad N_{cut} = \left(\begin{array}{c} k_{max} + M \\ k_{max} \end{array} \right) = \frac{(k+M)!}{k!M!} \tag{7.4}$$

Η μέση τιμή και η τυπική απόκλιση της F προσεγγίζονται ως αθροίσματα των συντελεστών a_i του γραμμικού συνδυασμού της εξίσωσης (7.4)

$$\mu_F \approxeq a_0 \ \ \& \ \ \sigma_F \approxeq \sqrt{\sum_{i=1}^{N_{cut}} a_i^2} \tag{7.5}$$

όπου k η τάξη πολυωνυμικού χάους και $M = dim(\vec{x})$ ο αριθμός των αβέβαιων μεταβλητών.

Υπολογίζοντας τις παραπάνω στατιστικές ροπές, μπορεί να υπολογιστεί η Μετρική Στιβαρότητας F_R, ενώ για τις παραγώγους της μετρικής απαιτούνται και οι παράγωγοι των στατιστικών ροπών:

$$F_R = \mu_F + \kappa\, \sigma_F \ \Rightarrow \ \frac{\partial F_R}{\partial b_n} = \frac{\partial \mu_F}{\partial b_n} + \kappa \, \frac{\partial \sigma_F}{\partial b_n} \tag{7.6}$$

όπου κ η παράμετρος DFSS, της οποίας το πρόσημο δείχνει το κατά πόσο είναι απαισιόδοξη ή αισιόδοξη η προσέγγιση του σχεδιασμού, ενώ η απόλυτη τιμή της φανερώνει τη βαρύτητα της τυπικής απόκλισης ως προς τη μέση τιμή. Οι παράγωγοι των στατιστικών ροπών προσδιορίζονται παραγωγίζοντας τις εξισώσεις (7.5) ως προς b_n:

$$\frac{\partial \mu_F}{\partial b_n} \approxeq \frac{\partial a_0}{\partial b_n} \quad \& \quad \frac{\partial \sigma_F}{\partial b_n} \approxeq \frac{\sum_{i=1}^{N_{cut}} a_i \frac{\partial a_i}{\partial b_n}}{\sqrt{\sum_{i=1}^{N_{cut}} a_i^2}} = \frac{1}{\sigma_F} \sum_{i=1}^{N_{cut}} a_i \frac{\partial a_i}{\partial b_n} \tag{7.7}$$

Για κάθε τύπο κατανομής που μπορεί να ακολουθούν οι στοχαστικές μεταβλητές c_l, αντιστοιχούν συγκεκριμένες οικογένειες ορθογώνιων πολυωνύμων. Στην περίπτωση της κανονικής κατανομής την οποία ακολουθούν όλες οι στοχαστικές μεταβλητές σε αυτήν την εργασία ($c_l \in \mathcal{N}(\mu_l, \sigma_l^2)$) αντιστοιχούν τα πιθανοτικά πολυώνυμα Hermite ($He_i(c_l)$), με πεδίο ορισμού $D = [-\infty, +\infty]$ και συνάρτηση πυκνότητας πιθανότητας (PDF) $w(c_l) = \frac{1}{\sigma_l \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{c_l - \mu_l}{\sigma_l}\right)^2}$.

Τα πολυδιάστατα πιθανοτικά ορθοκανονικά πολυώνυμα ορίζονται ως το γινόμενο των αντίστοιχων μονοδιάστατων πολυωνύμων $\widetilde{\psi}_i(\vec{c}) = \prod_{l=1}^{M} \widetilde{He}_{i_l}\left(\frac{c_l - \mu_l}{\sigma_l}\right)$. Η PDF των $\widetilde{\psi}_i$, ομοίως, ορίζεται ως το γινόμενο των PDF των αντίστοιχων μονοδιάστατων πολυωνύμων: $W(\vec{c}) = \prod_{l=1}^{M} w(c_l)$. Έτσι οι συντελεστές niPCE a_i ορίζονται ως εξής:

$$a_i = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} F(\vec{b}, c_1, ..., c_M) \widetilde{\psi}_i(z_1, ..., z_M) W(c_1, ..., c_M)\, dc_1 \cdots dc_M \tag{7.8}$$

όπου $z_l = \frac{c_l - \mu_l}{\sigma_l}$, $l = 1, 2, ..., M$, ενώ οι τιμές της Ποσότητας Ενδιαφέροντος F απαιτούν την επίλυση του πρωτεύοντος προβλήματος (πρόβλημα ροής).

Αντίστοιχα, οι παράγωγοι των a_i ως προς τις μεταβλητές σχεδιασμού, προκύπτουν από την παραγώγηση της σχέσης (7.8)

$$\frac{\partial a_i}{\partial b_n} = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \frac{\partial}{\partial b_n} F(\vec{b}, c_1, ..., c_M) \widetilde{\psi}_i(z_1, ..., z_M) W(c_1, ..., c_M)\, dc_1 \cdots dc_M \tag{7.9}$$

όπου οι παράγωγοι ευαισθησίας $\partial F / \partial b_n$ υπολογίζονται με την επίλυση του συζυγούς προβλήματος.

Ο υπολογισμός των ολοκληρωμάτων (7.8) και (7.9) γίνεται με αριθμητική ολοκλήρωση Gauss Hermite. Η ολοκλήρωση μπορεί να γίνει είτε για πλήρες πλέγμα κόμβων Gauss είτε για *αραιά πλέγματα Smolyak*. Στην πρώτη περίπτωση οι τιμές των κόμβων είναι ίσες με τις ρίζες των $\widetilde{\psi}_i$, ενώ το πλήθος τους ισούται με $(k+1)^M$, που είναι ταυτόσημο με το κόστος της μεθόδου (μετρούμενο με μονάδα το κόστος μίας αξιολόγησης), αφού ισούται με τον αριθμό των αξιολογήσεων που απαιτούνται για το υπολογισμό των στατιστικών μεγεθών και των παραγώγων τους. Η εκθετικη αύξηση του κόστους για μεγάλους αριθμούς αβέβαιων μεταβλητών, αντιμετωπίζεται με τη χρήση πλεγμάτων ολοκλήρωσης Smolyak, που απαιτεί αισθητά μικρότερους αριθμούς αξιολογήσεων για $M \geq 4$, σε βάρος της ακρίβειας των αποτελεσμάτων. Στον Πίνακα 7.1 παρατίθενται οι διαφορές στο πλήθος των αξιολογήσεων για τα δύο πλέγματα κόμβων Gauss.

Πλήρες Πλέγμα / Πλέγμα Smolyak						
	M					
k	1	2	3	4	5	6
0	1/1	1/1	1/1	1/1	1/1	1/1
1	2/3	4/5	8/7	16/9	32/11	64/13
2	3/5	9/13	27/25	81/41	243/61	729/85
3	4/9	16/29	64/69	256/137	1024/241	4096/389
4	5/17	25/65	125/177	625/401	3125/801	15625/1457
5	6/33	36/145	216/441	1296/1105	7776/2433	46656/4865

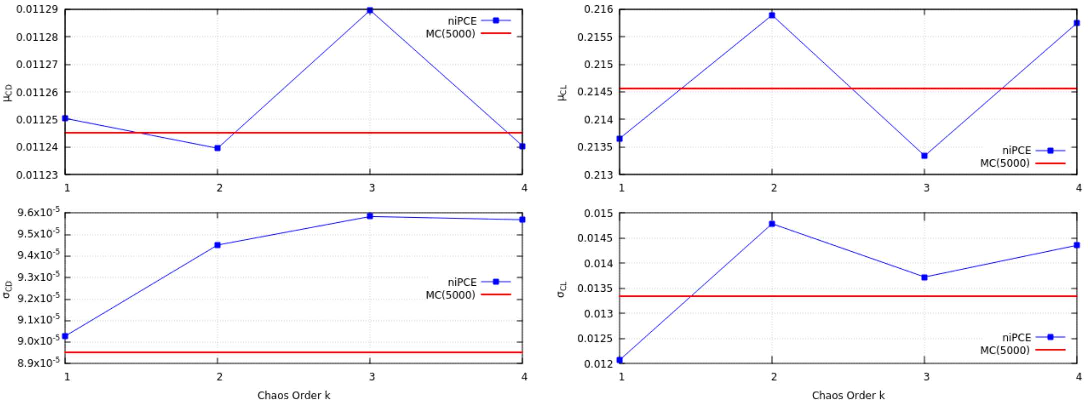
Πίνακας 7.1: *Κόμβοι Gauss, άρα και πλήθος αξιολογήσεων, για τον υπολογισμό των συντελεστών niPCE, τόσο για πλήρη όσο και για Smolyak πλέγματα ολοκλήρωσης.*

Κατά την Ποσοτικοποίηση Αβεβαιότητας ή το Στιβαρό Σχεδιασμό υπό κατασκευαστικές ατέλειες, οι κόμβοι Gauss για το εκάστοτε πλέγμα ολοκλήρωσης αποτελούν τις τιμές των αβέβαιων μεταβλητών c_n που εισάγονται στο μοντέλο KLT για την παραγωγή γεωμετρικής ατέλειας σε 2Δ σώμα. Οι στατιστικές ροπές των αβέβαιων μεταβλητών $\mu_l, \ \sigma_l \ \forall \ l = 1, 2, ..., M$ καθορίζονται από τον χρήστη.

Ο κώδικας niPCE που είχε αναπτυχθεί από την ΜΠΥΡ&Β/ΕΜΠ, επεκτάθηκε ώστε να μπορεί να υπολογίσει τις παραγώγους των στατιστικών μεγεθών της F και κατ΄ επέκταση των στιβαρών παραγώγων ευαισθησίας για δεδομένο κ, λαμβάνοντας τιμές για τις παραγώγους ευαισθησίας από εξωτερικό επιλύτη. Στην περίπτωση αυτή χρησιμοποιείται ο συνεχής συζυγής επιλύτης την ΜΠΥΡ&Β/ΕΜΠ, σε περιβάλλον OpenFOAM©.

Ποσοτικοποίηση Αβεβαιότητας υπό Κατασκευασικές Ατέλειες

Διεξάγεται Ποσοτικοποίηση Αβεβαιότητας των αεροδυναμικών συντελεστών (C_D, C_L) μεμονωμένης αεροτομής NACA 0012 με κατασκευαστικές ατέλειες. Η ροή θεωρείται ασυμπίεστη και χρονικά μόνιμη για $Re_c = 2,600,000$ και γωνία πρόσπτωσης $a = 2^o$ και επιλύεται σε δομημένο υπολογιστικό πλέγμα 160×135 τετραεδρικών κελιών, ενώ χρησιμοποιείται το μοντέλο τύρβης Spalart–Allmaras . Ο αριθμός των KLT στοχαστικών μεταβλητών είναι $M = 5$ και οι στοχαστικές τους ροπές $\mu_l = 0$, $\sigma_l = 2 \cdot 10^{-3} \, m \ \forall \ l = 1, 2, ..., 5$, ενώ πραγματοποιείται παραμετρική μελέτη ως προς την τάξη χάους k και σύγκριση των αποτελεσμάτων με τη μέθοδο *Monte Carlo* για 5000 δείγματα. Η πιστοποίηση των αποτελεσμάτων για τις δύο μεθόδους περιγράφεται στο Σχήμα 7.3. Στο Σχήμα 7.3 παρατηρείται ότι, στην περίπτωση αυτή, αύξηση της τάξης χάους δεν συνεπάγεται την αύξηση ακρίβειας. Οι πιο ακριβείς λύσεις παράγονται, κατά μέσο όρο, για $k = 1$ και $k = 3$, όμως αυτό είναι συγκυριακό και δεν μπορεί να γενικευτεί.



Σχήμα 7.3: *Αεροτομή NACA 0012 με κατασκευαστικές ατέλειες: Ποσοτικοποίηση Αβεβαιότητας C_D, C_L για niPCE με τάξη χάους k από 1 έως 4 και για Monte Carlo(5000).*

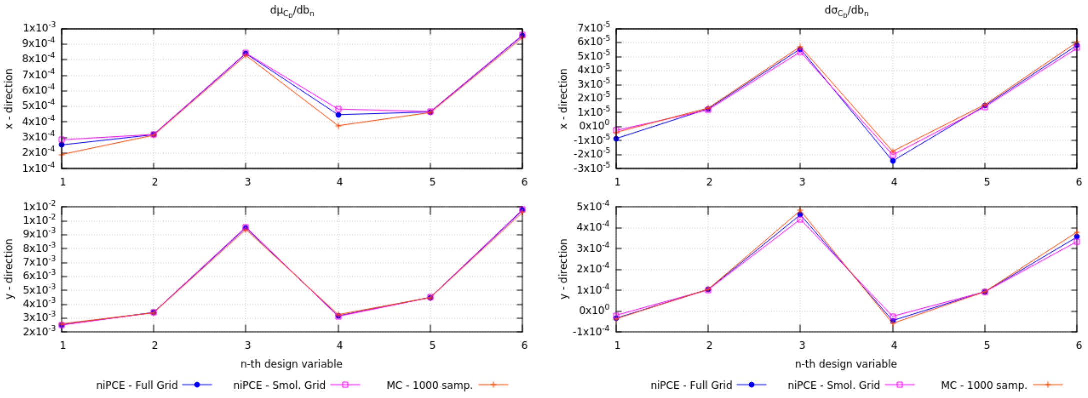
Για τις προσομοιώσεις που ακολουθούν επιλέγεται $k = 1$, διότι παράγει ένα σχετικά ακριβή υπολογισμό των στατιστικών ροπών για το χαμηλότερο υπολογιστικό κόστος, καθώς το κύριο μέλημα της εργασίας δεν είναι η υψηλή ακρίβεια αλλά η διερεύνηση και πιστοποίηση του μοντέλου κατασκευστικών ατελειών για το Στιβαρό Σχεδιασμό.

Στη συνέχεια, πραγματοποιείται Ποσοτικοποίηση Αβεβαιότητας ως προς τις παραγώγους των στατιστικών ροπών όπως ορίζονται στη σχέση (7.7). Η Ποσότητα Ενδιαφέροντος είναι το C_D και η αεροτομή με γεωμετρικές ατέλειες που μελετάται είναι η E387. Οι ροϊκές συνθήκες είναι $Re_c = 2,600,000$ και $a = 2^o$, ενώ το υπολογιστικό πλέγμα είναι δομημένο, με 160×135 τετραεδρικά κελιά. Οι παράγωγοι υπολογίζονται σε πλέγμα 3×2 ελεύθερων σημείων ελέγχου, για $k = 1$ και $M = 5$ τόσο για πλήρη πλέγματα όσο και για πλέγματα Smolyak, ενώ τα αποτελέσματα συγκρίνονται με της *Monte Carlo*, για 1000 δείγματα. Η πιστοποίηση των παραγώγων παρατίθεται στο Σχήμα 7.4.

Στο Σχήμα 7.4 φαίνεται πως η προτεινόμενη μέθοδος, με τις υιοθετούμενες ρυθμίσεις, προλέγει ικανοποιητικά τις παραγώγους των στατιστικών ροπών μιας Ποσότητας Ενδιαφέροντος. Τα αποτελέσματα για αραιά πλέγματα ολοκλήρωσης Smolyak είναι πρακτικά εξίσου ακριβή με τα αποτελέσματα για πλήρη πλέγματα ολοκλήρωσης, για τάξη χάους $k = 1$. Έτσι δικαιολογείται η επιλογή αξιοποίησης πλεγμάτων Smolyak για το σχεδιασμό υπό κατασκευασικές ατέλειες.

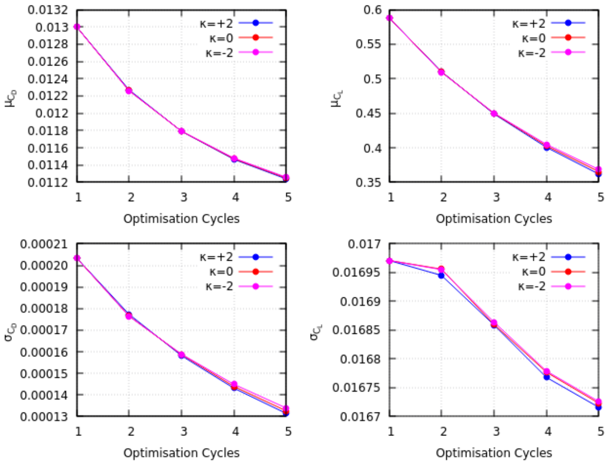
Σχεδιασμός στην περίπτωση Κατασκευαστικών Ατελειών

Αρχικά πραγματοποιείται μονοκριτηριακός Στιβαρός Σχεδιασμός με στόχο τη βελτιστοποίηση της στοχαστικής συμπεριφοράς της οπισθέλκουσας μεμονωμένης αεροτομής E387 για 5 κύκλους βελτιστοποίησης. Διατηρούνται οι ίδιες ροϊκές συνθήκες, το ίδιο υπολογιστικό πλέγμα και οι ίδιες ρυθμίσεις για το μοντέλο KLT με την παραπάνω ανάλυση. Επιλέγονται τρείς τιμές της παραμέτρου DFSS: $\kappa = -2, 0, +2$. Οι στατιστι-



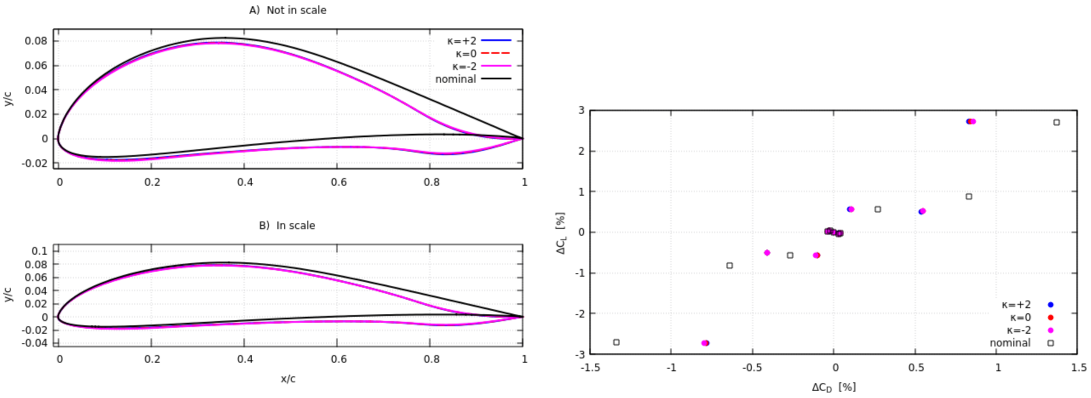
Σχήμα 7.4: *Αεροτομή E387 με κατασκευαστικές ατέλειες: Ποσοτικοποίηση αβεβαιότητας $\partial\mu_{C_D}/\partial b_n$ και $\partial\sigma_{C_D}/\partial b_n$ για niPCE($k=1$) και για Monte Carlo(1000).*

κές ροπές του C_D που προκύπτουν σε κάθε κύκλο παρατίθενται στο Σχήμα 7.5.



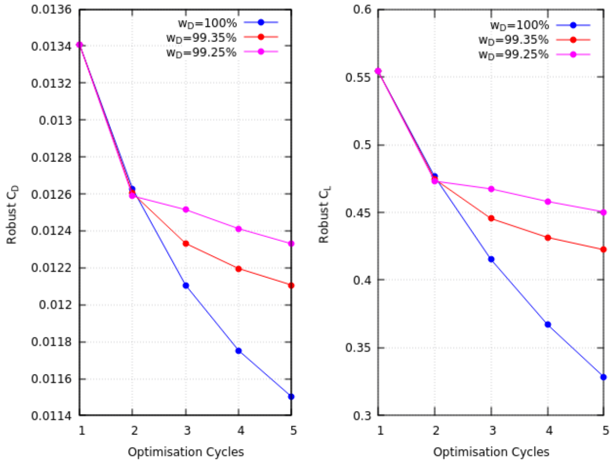
Σχήμα 7.5: *Μονοκριτηριακός σχεδιασμός αεροτομής E387 με κατασκευαστικές ατέλειες: στατιστικές ροπές C_D, C_L για 5 κύκλους βελτιστοποίησης και $\kappa = -2, 0, +2$.*

Επίσης οι τελικές στιβαρές γεωμετρίες καθώς και η απόκλιση των C_D, C_L από την αντίστοιχη μέση τιμή τους, αποτυπώνονται στο Σχήμα 7.6. Συνολικά από το Σχήμα 7.5 βγαίνει το συμπέρασμα ότι η προτεινόμενη μέθοδος επιτυγχάνει μείωση τόσο της μέσης τιμής του C_D (κατά 13.5 %) όσο και της τυπικής απόκλισής του (κατά 34.8 %). Το γεγονός αυτό παρατηρείται και από τη μείωση της μέγιστης σχετικής απόκλισης του C_D στο Σχήμα 7.6. Επιπλέον, από τα Σχήματα 7.5 και 7.6 προκύπτει ότι τα αποτελέσματα δεν επηρεάζονται ιδιαίτερα από τη μεταβολή του κ. Αυτό δεν μπορεί να γενικευτεί, αλλά αιτιολογείται δεδομένου ότι οι παράγωγοι του C_D είναι τουλάχιστον δύο τάξεις μεγέθους μικρότερες από τις αντίστοιχες παραγώγους της μέσης τιμής του, όπως φαίνεται στο Σχήμα 7.4.



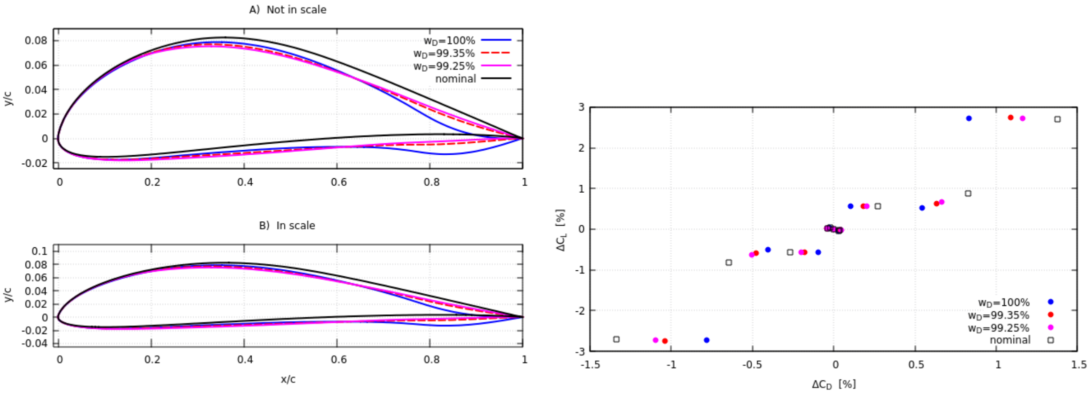
Σχήμα 7.6: *Μονοκριτηριακός σχεδιασμός αεροτομής E387 με κατασκευαστικές ατέλειες: η αρχική και οι τελικές στιβαρές γεωμετρίες (αριστερά) και η σχετική απόκλιση των C_D, C_L από την αντίστοιχη μέση τιμή (δεξιά).*

Στη συνέχεια εκτελείται στην ίδια αεροτομή, δικριτηριακός Στιβαρός Σχεδιασμός ως προς τη σταθμισμένη συνάρτηση-στόχο $w_D C_D - (1 - w_D) C_L$ (που απαιτεί ένα συζυγή επιλύτη αντί για δύο, για κάθε στόχο), αποσκοπώντας στη βελτιστοποίηση της στοχαστικής συμπεριφοράς της άνωσης και της οπισθέλκουσας, για 5 κύκλους βελτιστοποίησης. Για τις ίδιες συνθήκες, επιλέγονται $\kappa = +2$ και τρεις τιμές βαρών: $w_D = 100,\ 99.35,\ 99.25\ \%$. Στο Σχήμα 7.7 παρουσιάζονται οι στιβαροί αεροδυναμικοί συντελεστές που ορίζονται ως εξής: $C_D^{(robust)} = \mu_{C_D} + \kappa\, \sigma_{C_D}$ και $C_L^{(robust)} = \mu_{C_L} + \kappa\, \sigma_{C_L}$.



Σχήμα 7.7: *Δικριτηριακός σχεδιασμός αεροτομής E387 με κατασκευαστικές ατέλειες: στιβαροί συντελεστές C_D, C_L για 5 κύκλους βελτιστοποίησης και τρεις συνδυασμούς βαρών.*

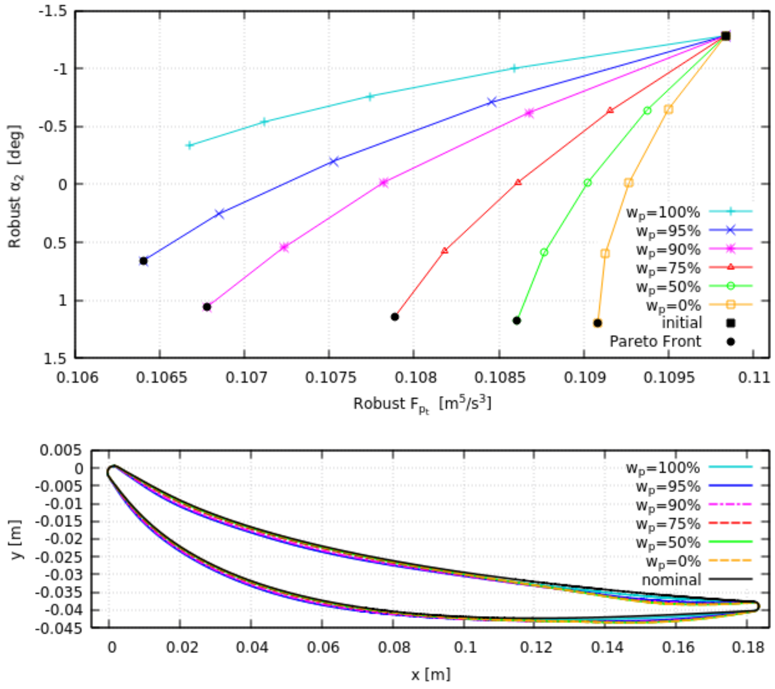
Οι τελικές στιβαρές γεωμετρίες και η απόκλιση των C_D, C_L από την αντίστοιχη μέση τιμή αποτυπώνονται στο Σχήμα 7.8. Παρατηρείται ότι η χρήση σταθμισμένης συνάρτησης-στόχου δύναται να ανακόψει τη δραστική πτώση του ${C_L}^{(robust)}$, από 40.7% για $w_D = 100$ % σε 18.7 % για $w_D = 99.25$ %. Βέβαια, οι δύο στόχοι είναι αντικρουόμενοι, καθώς όποια μεταβολή στα βάρη που προκαλεί μείωση της οπισθέλκουσας προκαλεί αύξηση της άνωσης και αντίστροφα. Το γεγονός αυτό παρατηρείται στις διαφορές μεταξύ των στιβαρών αεροτομών στο Σχήμα 7.8. Στο Σχήμα 7.8 παρατηρείται αύξηση της μέγιστης σχετικής απόκλισης του C_D με την αύξηση της τιμής του w_D, η οποία όμως παραμένει πάντοτε μικρότερη από την αντίστοιχη απόκλιση της αρχικής αεροτομής. Η ουσιαστικά αμετάβλητη μέγιστη σχετική απόκλιση του C_L, είναι αναμενόμενη λόγω της επιλογής μεγάλων τιμών για τα βάρη w_D, που φανερώνει την ιεράρχηση της οπισθέλκουσας κατά τη σχεδίαση.



Σχήμα 7.8: *Δικριτηριακός σχεδιασμός αεροτομής E387 με κατασκευαστικές ατέλειες: η αρχική και οι τελικές στιβαρές γεωμετρίες (αριστερά) και η σχετική απόκλιση των C_D, C_L από την αντίστοιχη μέση τιμή (δεξιά).*

Τέλος, διεξάγεται δικριτηριακός στιβαρός σχεδιασμός 2Δ σταθερής πτερύγωσης TU Berlin για 5 κύκλους βελτιστοποίησης. Σκοπός είναι η απόλυτη αύξηση γωνίας απόκλισης της ροής $\theta = a_1 - a_2$ και η ταυτόχρονη μείωση των απωλειών ολικής πίεσης F_{pt}. Δεδομένου ότι η γωνία εισόδου είναι σταθερή και ίση με $a_1 = -42^o$ αρκεί να μεγιστοποιηθεί η γωνία εξόδου a_2 και η συνάρτηση-στόχος ορίζεται $w_p F_{pt} - (1 - w_p) a_2$. Οι ροϊκές συνθήκες είναι $U_\infty = 48 m/s$ και $Re_c = 672,000$, ενώ η χορδή της αεροτομής είναι ίση με $0.1876\, m$. Η ροή που θεωρείται μόνιμη και ασυμπίεστη επιλύεται σε 2Δ μη δομημένο πλέγμα 80228 κελιών, με περιοδικά όρια. Οι στατιστικές ροπές των 5 αβέβαιων μεταβλητών του μοντέλου KLT ορίζονται $\mu_l = 0$, $\sigma_l = 2 \cdot 10^{-3}\, m \;\forall\; l = 1, 2, ..., 5$. Επίσης επιλέγονται $\kappa = +2$ και έξι τιμές βαρών $w_p = 100,\ 95,\ 90,\ 75,\ 50,\ 0$ %. Στο Σχήμα 7.9 αποτυπώνονται η στιβαρή γωνία απόκλισης $({\alpha_2}^{(robust)} = \mu_{\alpha_2} + \kappa\, \sigma_{\alpha_2})$ και οι στιβαρές απώλειες ολικής πίεσης $({F_{pt}}^{(robust)} = \mu_{F_{Pt}} + \kappa\, \sigma_{F_{Pt}})$ για τους 5 κύκλους.

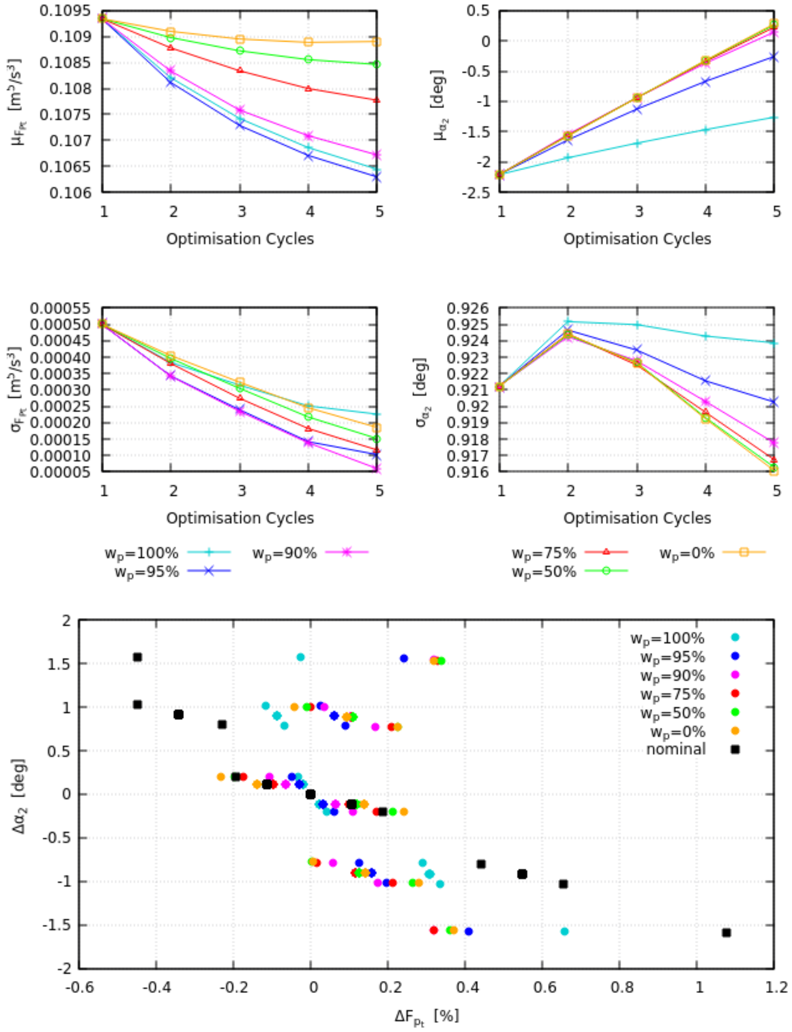
Στο Σχήμα 7.9 παρατηρείται ότι οι δύο στόχοι είναι μη αντικρουόμενοι, καθώς οι τελικές τιμές των στιβαρών στόχων για όλα τα βάρη, εκτός από το $w_p = 100$%, αποτελούν το Μέτωπο Pareto των κυρίαρχων λύσεων. Έτσι, μετά από 5 κύκλους, η μεγαλύτερη μείωση των στιβαρών απωλειών ολικής πίεσης εμφανίζεται για $w_p = 95$ % και είναι



Σχήμα 7.9: *Δικριτηριακός σχεδιασμός 2Δ σταθερής πτερύγωσης TU Berlin με κατασκευαστικές ατέλειες: απώλειες ολικής πίεσης $F_{pt}^{(robust)}$ και γωνία εξόδου $\alpha_2^{(robust)}$ (πάνω) και οι τελικές στιβαρές γεωμετρίες σε πραγματική κλίμακα (κάτω).*

ίση με 3.1 %, ενώ η μέγιστη αύξηση της στιβαρής γωνίας εξόδου είναι 192.9 % για το βάρος $w_p = 0\,\%$. Αντίστοιχα, το γεγονός ότι οι στόχοι είναι μη αντικρουόμενοι φανερώνεται και από τις ‘υβριδικές’ τελικές στιβαρές αεροτομές, που συνδυάζουν τόσο το μειωμένο πάχος όσο και την αύξηση της καμπυλότητας στην ακμή εκφυγής, για τα βάρη $w_p = 95,\ 90,\ 75,\ 50;\%$.

Στο Σχήμα 7.10 παρατίθενται οι υπολογιζόμενες στατιστικές ροπές και η συσχέτιση των αποκλίσεων των a_2 και F_{p_t}. Το συμπέρασμα του Σχήματος είναι ότι επιτυγχάνεται δραστική μείωση της διασποράς και της τυπικής απόκλισης των απωλειών ολικής πίεσης (μέγιστη κατά 88.1 % για $w_p = 90\,\%$), ενώ η μείωση της τυπικής απόκλισης της γωνίας εξόδου είναι πολύ μικρότερη, όπου επιτυγχάνεται (μέγιστη μείωση 0.005^o για $w_p = 0\,\%$). Αντίθετα η αύξηση της μέσης τιμής της a_2 είναι ιδιαίτερα μεγάλη, με τη μέγιστη να είναι ίση με περίπου 2.5^o. Τα ίδια συμπεράσματα προκύπτουν και από τις τιμές των αποκλίσεων. Η μικρή πτώση της σ_{a_2} φανερώνεται και από τη μηδαμινή μείωση της αντίστοιχης μέγιστης απόλυτης απόκλισης. Αντίθετα η πτώση της μέγιστης σχετικής απόκλισης της F_{p_t} είναι αισθητά μεγάλυτερη για όλα τα βάρη, με τη μέγιστη να ισούται με 70.4 % για $w_p = 90\,\%$.



Σχήμα 7.10: *Δικριτηριακός σχεδιασμός 2Δ σταθερής πτερύγωσης TU Berlin με κατασκευαστικές ατέλειες: στατιστικές ροπές των F_{pt}, a_2 (πάνω) και συσχέτιση της απόλυτης απόκλισης της a_2 με τη σχετική απόκλιση των F_{pt} (κάτω).*

Συμπεράσματα

Η προτεινόμενη μέθοδος αιτιοκρατικού στιβαρού σχεδιασμού υπό κατασκευαστικές ατέλειες που συνδυάζει τις μεθόδους KLT, niPCE και τη συζυγή μέθοδο. Έχει ποσοτικά αποτελέσματα που ποιοτικά είναι τα αναμενόμενα, όσον αφορά τη βελτιστοποίηση της στοχαστικής αεροδυναμικής συμπεριφοράς ενός 2Δ σώματος με γεωμετρικές ατέλειες. Η μέθοδος πιστοποιήθηκε τόσο για μεμονωμένες αεροτομές όσο και για αεροτομές σταθερών πτερυγώσεων, για τρεις διαφορετικές συναρτήσεις-στόχους. Η μέθοδος έχει σημαντικό υπολογιστικό κόστος που πολλαπλασιάζεται ανά κύκλο βελτιστοποίησης, το οποίο όμως είναι αισθητά μικρότερο σε σχέση με άλλες προσπάθειες που πραγματοποιούν στοχαστικό Στιβαρό Σχεδιασμό.

Appendix A

OpenFOAM CFD Validation

In this section of the Appendix, the validation of the CFD method used to numerically solve the primal problem, as presented in Section 2.1, takes place. Meaning, that the the results produced by the CFD analysis, through OpenFOAM are compared with corresponding experimental results found in literature.

In this thesis, three distinct primal problems are solved, both with and without the inclusion of KLT-generated manufacturing imperfections (explored thoroughly in Chapter 4), the later integrated within UQ and RDO loops (more information in Section 5.1). These three CFD cases are mainly distinguished by their varying initial shape, two of them being airfoils: the symmetric NACA 0012 (Subsection 5.2.2) as well as the E387 (Subsection 5.3.1) and the third being the TU Berlin compressor stator cascade (Subsection 6.2.2). Nevertheless, only the first two airfoil cases are verified, given that they have been extensively subjected to wind tunnel testing and their aerodynamic performance is well documented through the years. The compressors cascade is a simplified 2D model of the TU Berlin axial compressor stator blade. The 3D CFD simulation of the stator has been executed and verified in [40]. The cascade has been extracted from these 3D simulations, so as to be used as a simplified model for the purposes of this work. Up until now, no experimental data for this specific cascade are found in literature.

Airfoil: NACA 0012

The NACA 0012 airfoil experimental data used to validate the accuracy of the OpenFOAM fluid solver can be found in [41], and for now on is referred to as *Landson et al.* experiments, for short. The flow solver configurations, computational mesh, turbulence model and boundary conditions are exactly the same with the ones described in Subsection 5.2.1, only changing the far-field velocity to $U_\infty = 60\ m/s$, in order to achieve the flow conditions used in the wind tunnel experiments: $Mach = 0.15$ and $Re_c = 6 \cdot 10^6$.

For these flow conditions the OpenFOAM's *simpleFoam* incompressible flow solver is used, to compute the flow field around the airfoil for three of the same far-field velocity angles of attack (AoA) used by *Landson et al.*: $AoA = [0.01, 2.15, 4.11]$ *degrees*. The reason for which these AoA are selected is to validate the solver's accuracy close to the AoA= 2 *deg* used in the UQ analysis that takes place in Subsection 5.2.3.

The C_D and C_L coefficients as computed both through OpenFOAM and the *Landson et al.* experiments, for the aforementioned AoA, are displayed in Table A.1.

	Landson et al.		*OpenFOAM*	
AoA [deg]	C_L	C_D	C_L	C_D
-0.01	-0.0122	0.00804	-0.012716	0.007876
2.15	0.2236	0.00823	0.228503	0.008392
4.11	0.4397	0.00879	0.435971	0.009771

Table A.1: *NACA 0012 airfoil: aerodynamic coefficients for* $Re_c = 6,000,000$. *Comparison between the OpenFOAM-computed and Landson et al. experimental results.*

The same comparison is also visualized through the Figure A.1, where the polar and $C_L - AoA$ diagrams for the NACA 0012 airfoil. Meanwhile the relative error of the CFD results when compared to the corresponding experimental results are displayed in Figure A.2.

Overall, the results of the OpenFOAM's solver configurations produce valid results, especially in for AoA close to zero. As expected the C_L results are fairly precise, given that the pressure field is more or less easily computed accurately for all AoA, keeping in mind the dominant component of the lift force exerted on an airfoil is caused by the pressure difference between its suction and pressure sides. This result also backed by the CFD-computed pressure coefficient (C_p) distribution on to the airfoil's surface is relatively accurate as well when compared with the corresponding experimental distribution, as plotted in Figure A.3.

Nevertheless, even the more complex, due to the its viscous component, C_D, requiring proper treatment (as featured in Subsection 2.1.3) and denser meshing near the airfoil surface, is computed with relative accuracy. According to Figure A.2, the lowest relative error is identified for AoA= 2.15 *deg*, justifying the 2 *deg* AoA used in the C_L, C_D UQ analysis, executed in Subsection 5.2.1, for the same airfoil. The slightly higher relative error appearing for AoA= 4.11 *deg*, indicates that a denser mesh might be required, given that greater velocity gradients as well as some unsteady phenomena may start to appear.

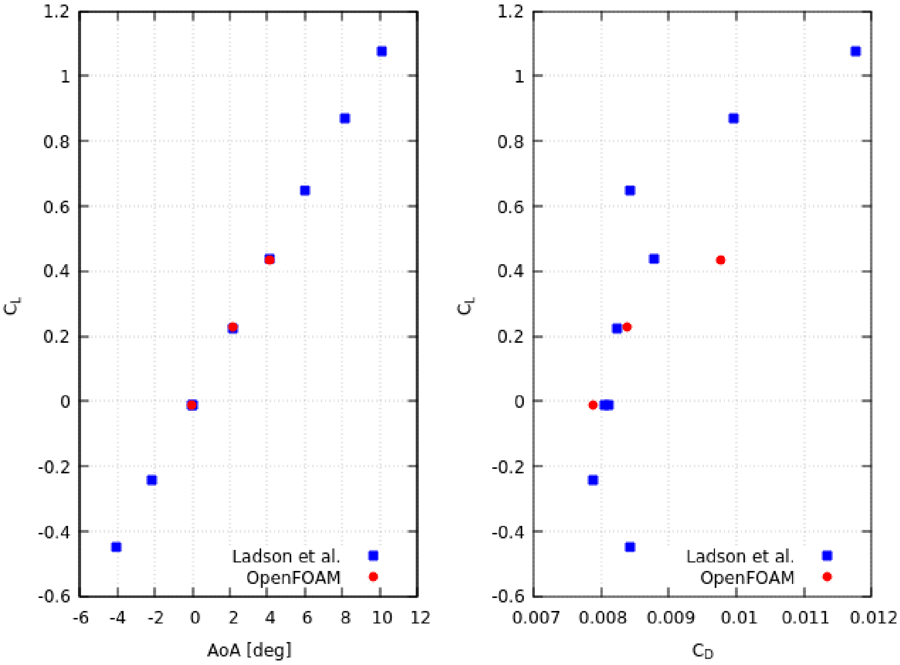


Figure A.1: *NACA 0012 airfoil: C_L − C_D polar diagram and C_L − AoA diagram for $Re_c = 6,000,000$. The Ladson et al. results are included for all AoA used in the experiments.*

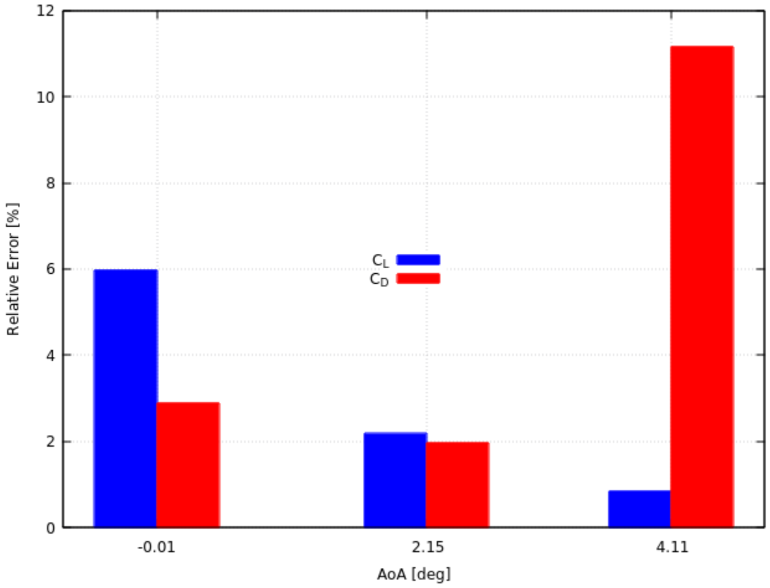


Figure A.2: *NACA 0012 airfoil: relative error of the OpenFOAM-generated aerodynamic coefficients w.r.t. the results of the Ladson et al. experiments for $Re_c = 6,000,000$.*

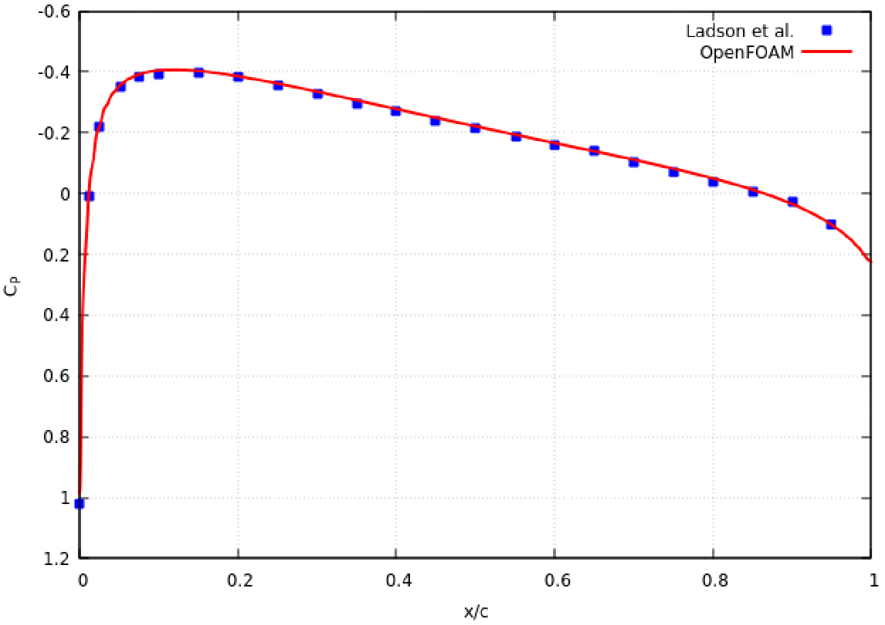


Figure A.3: *NACA 0012 airfoil: pressure coefficient distribution for AoA*$=0$ *deg and* $Re_c = 6,000,000$ *from both the OpenFOAM analysis and the Ladson et al. experiments.*

Airfoil: E387

The experimental data of the aerodynamic performance of the E387 airfoil are extracted from the [42]. Specifically the Spring 1997, J.Robertson data set are used and are referred to as *Robertson* experimental results. The same *simpleFoam* solver configuration and mesh are implemented as those mentioned in Subsection 5.3.1. Only the far-field velocity is modified to $U_\infty = 2\ m/s$, to attain the Reynolds number $Re_c = 200,000$ of the *Robertson* experiments.

The flow field around the airfoil is computed for five different AoA $= [0, 1, 2, 3, 4]$ *degs* and the results are given in Table A.2, in comparison to the *Robertson* results.

The same comparison is also visualized through Figure A.4, where the polar and $C_L - AoA$ diagrams for the E387 airfoil and through Figure A.5, where the relative error of the CFD results, when compared to the corresponding experimental results, are displayed.

	Robertson		*OpenFOAM*	
AoA [*deg*]	C_L	C_D	C_L	C_D
0	0.371	0.0110	0.372208	0.011169
1	0.477	0.0118	0.479919	0.011952
2	0.585	0.0128	0.587641	0.012983
3	0.703	0.0139	0.694792	0.014277
4	0.799	0.0148	0.801176	0.015822

Table A.2: *E387 airfoil: aerodynamic coefficients for* $Re_c = 200,000$. *Comparison between the OpenFOAM-computed and Robertson experimental results.*

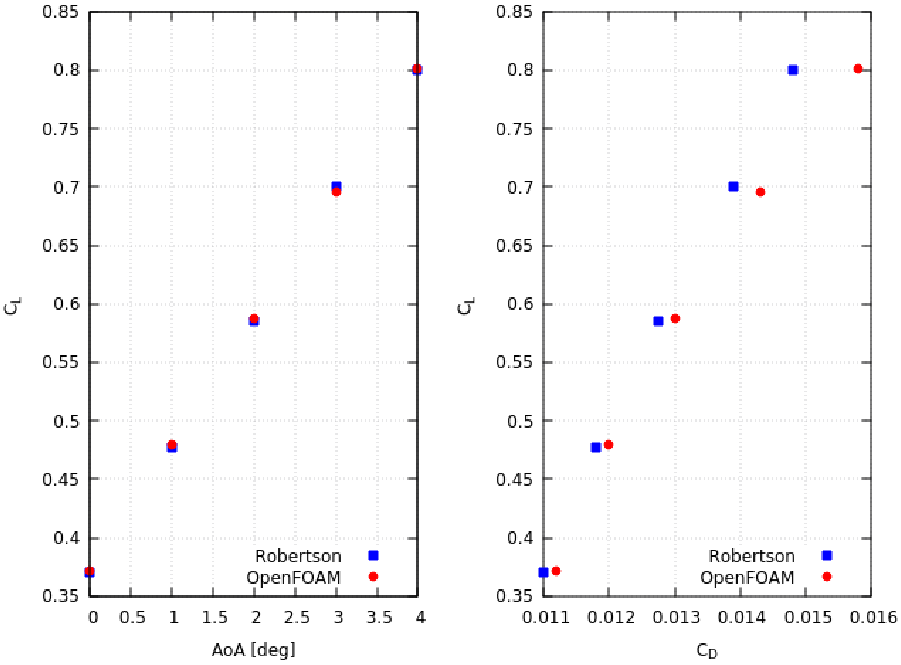


Figure A.4: *E387 airfoil:* $C_L - C_D$ *polar diagram and* C_L − *AoA diagram for* $Re_c = 200,000$ *from both the OpenFOAM analysis and the Robertson experiments.*

The conclusions drawn from the comparison of the results are very similar to those described in Subsection A of the Appendix. The CFD accuracy is greater the for AoA close to zero, while the lowest error for the drag coefficient is found for AoA=1 *deg*. Generally, the C_L CFD-generated prediction is more precise, while for AoA=4 *deg* the C_D error is the highest, indicating the need for a denser mesh near the airfoil. Overall, the OpenFOAM solver configuration is proven to produce relatively valid results, justifying its integration in UQ and RDO loops, as in Subsection 5.3.2 and Section 6.1, respectively.

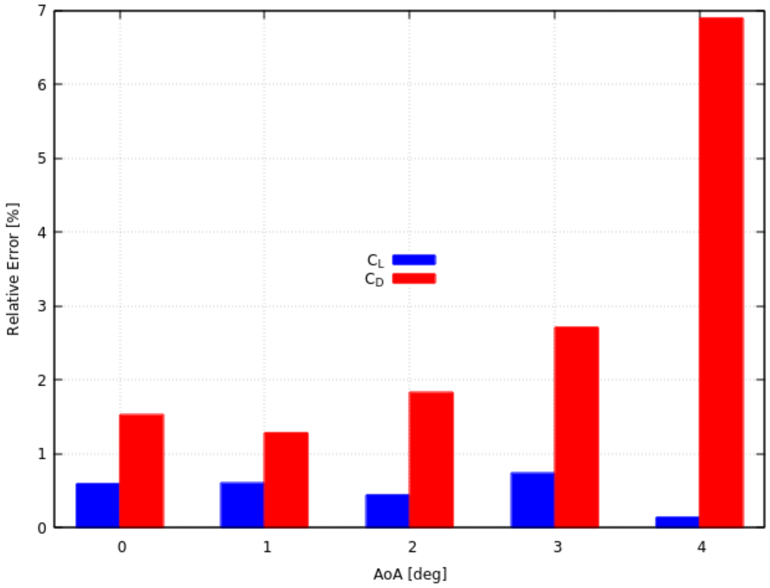


Figure A.5: *E387 airfoil: relative error of the OpenFOAM-generated aerodynamic coefficients w.r.t. the results of the Robertson experiments for* $Re_c = 200,000$.

Unfortunately, no C_p distributions can be found in [42], in order for such a comparison to be held.

Appendix B

Hermite Polynomials

The Hermite orthogonal polynomials are used for the niPCE method for uncertain variables following normal distributions. Two different categories of Hermite polynomials exist: the probabilists' Hermite polynomials He_n used in statistics and the physicists' Hermite polynomials H_n more often employed in different scientific domains related to physics. The main difference between the two is based on the formulation of their respective weight function. Yet both are defined into the same domain $D = [-\infty, +\infty]$.

The generalized formula for generating Hermite polynomials of degree n, is formulated as

$$H_n(x) = (-1)^n w^{-1}(x) \frac{d^n w(x)}{dx^n} \tag{B.1}$$

where $w(x)$ the weight function.

Weight Functions

The probabilists' polynomials weight function is defined as

$$w_{He}(x) = e^{-\frac{x^2}{2}} \tag{B.2}$$

while the physicists' polynomials weight function is defined as

$$w_H(x) = e^{-x^2} \tag{B.3}$$

Polynomial Formula

By integrating the weight function from eqs. (B.2) and (B.3) into the generalized formula in eq. (B.1), the formulas for each of the two categories, respectively, is produced.

A probabilists' polynomial of n degree is generated by

$$He_n(x) = (-1)^n e^{\frac{x^2}{2}} \, \frac{d^n e^{-\frac{x^2}{2}}}{dx^n} = \left(x - \frac{d}{dx}()\right)^n \cdot 1 \tag{B.4}$$

while physicists' polynomials of n degree are generated by

$$H_n(x) = (-1)^n e^{x^2} \, \frac{d^n e^{-x^2}}{dx^n} = \left(2x - \frac{d}{dx}()\right)^n \cdot 1 \tag{B.5}$$

The two definitions are not identical. This is backed by the fact that the first can be produced through the latter (as well as the opposite), by making use of the transform

$$He_n(x) = 2^{-\frac{n}{2}} \, H_n(\frac{x}{\sqrt{2}}) \iff H_n(x) = 2^{\frac{n}{2}} \, He_n(x\sqrt{2}) \tag{B.6}$$

Recurring Formula

Similarly, from definitions in eqs. (B.4) and (B.5), respectively, the recurring formulas for the polynomials is defined as

$$He_{n+1}(x) = x \, He_n(x) - n \, He_{n-1}(x) \tag{B.7}$$

$$H_{n+1}(x) = 2x \, H_n(x) - 2n \, H_{n-1}(x) \tag{B.8}$$

while their respective recurring derivatives w.r.t. to $x \in [-\infty, +\infty]$

$$He'_n(x) = \frac{dHe_n(x)}{dx} = n \, He_{n-1}(x) \tag{B.9}$$

$$H'_n(x) = \frac{dH_n(x)}{dx} = 2n \, H_{n-1}(x) \tag{B.10}$$

In addition, in a Hermite polynomial of n degree, the coefficient A_n of the term to the power of n is defined, though the use of the recurring formulas, as follows

$$A_n^{(He)}(x) = 1 \quad , \quad A_n^{(H)}(x) = 2^n \tag{B.11}$$

Orthogonality

Both of the two categories of Hermite polynomials are orthogonal w.r.t. their corresponding weight function $w(x)$, as follows

$$\langle He_n(x), He_m(x) \rangle_w = \int_{-\infty}^{+\infty} He_n(x) He_m(x) e^{-\frac{x^2}{2}} dx = n!\sqrt{2\pi}\delta_n^m \tag{B.12}$$

$$\langle H_n(x), H_m(x) \rangle_w = \int_{-\infty}^{+\infty} H_n(x) H_m(x) e^{-x^2} dx = 2^n n! \sqrt{\pi} \delta_n^m \tag{B.13}$$

where δ_n^m the Kronecker Delta.

From eqs. (B.12) and (B.13) the w-norms, for the two categories, are formulated

$$\left\langle He_n^2(x) \right\rangle_w = ||He_n||_w^2 = \gamma_n^{(He)} = n!\sqrt{2\pi} \quad , \quad \left\langle H_n^2(x) \right\rangle_w = ||H_n||_w^2 = \gamma_n^{(H)} = 2^n n! \sqrt{\pi} \tag{B.14}$$

Gauss Hermite Quadrature weights

The weights used in GQ integration are generally defined by the expression

$$\omega_j = -\frac{A_{n+1}}{A_n} \cdot \frac{\gamma_n}{\phi_{n+1}(x_j)\, \phi_n'(x_j)} \tag{B.15}$$

where ϕ_n a n degree polynomial belonging to a certain orthogonal family and x_j the roots of the aforementioned polynomial.

Meanwhile, the transform between the orthogonal polynomials ϕ of degrees $n+1$ and n is defined as

$$\phi_{n+1}(x_j) = -\frac{A_{n+1} A_{n-1}}{A_n^2} \cdot \frac{\gamma_n}{\gamma_{n-1}} \cdot \phi_{n-1}(x_j) \tag{B.16}$$

By including eq. (B.16) in eq. (B.15), the weights are formulated as

$$\omega_j = -\frac{A_n}{A_{n-1}} \cdot \frac{\gamma_{n-1}}{\phi_{n-1}(x_j)\, \phi_n'(x_j)} \tag{B.17}$$

When implementing the Gauss Hermite Quadrature, the weights are defined with Hermite polynomials, coefficient and w-norms. Therefore, when probabilists' Hermite polynomials are used, according to eqs. (B.4), (B.9), (B.11) and (B.14), the weights from eq. (B.17) become

$$\omega_j = \frac{(n-1)!\, \sqrt{2\pi}}{He_{n-1}(x_j)\, He_n'(x_j)} = \frac{n!\, \sqrt{2\pi}}{n^2\, He_{n-1}^2(x_j)} \tag{B.18}$$

If physicists' Hermite polynomials are utilized, eq. (B.17) alongside with eqs. (B.5), (B.10), (B.11) and (B.14) yields

$$\omega_j = \frac{2^n\, (n-1)!\, \sqrt{\pi}}{H_{n-1}(x_j)\, H_n'(x_j)} = \frac{2^{n-1}\, n!\, \sqrt{\pi}}{n^2\, H_{n-1}^2(x_j)} \tag{B.19}$$

To conclude, during the application of GHQ for the numerical integration of the niPCE coefficients, in this thesis, the symbol n designating the polynomial degree is replaced by $k+1$ (given that $n = k+1$), where k the niPCE chaos order.

Polynomial examples

According to the polynomials' formula in eqs. (B.4) and (B.5), respectively, the ten first Hermite polynomials are produced and displayed.

Probabilists' Hermite polynomial examples:

$$\begin{aligned}
He_0(x) &= 1\\
He_1(x) &= x\\
He_2(x) &= x^2 - 1\\
He_3(x) &= x^3 - 3x\\
He_4(x) &= x^4 - 6x^2 + 3\\
He_5(x) &= x^5 - 10x^3 + 15x\\
He_6(x) &= x^6 - 15x^4 + 45x^2 - 15\\
He_7(x) &= x^7 - 21x^5 + 105x^3 - 105x\\
He_8(x) &= x^8 - 28x^6 + 210x^4 - 420x^2 + 105\\
He_9(x) &= x^9 - 36x^7 + 378x^5 - 1260x^3 + 945x
\end{aligned}$$

Physicists' Hermite polynomial examples:

$$\begin{aligned}
H_0(x) &= 1\\
H_1(x) &= 2x\\
H_2(x) &= 4x^2 - 4\\
H_3(x) &= 8x^3 - 12x\\
H_4(x) &= 16x^4 - 48x^2 + 12\\
H_5(x) &= 32x^5 - 160x^3 + 120x\\
H_6(x) &= 64x^6 - 480x^4 + 720x^2 - 120\\
H_7(x) &= 128x^7 - 1344x^5 + 3360x^3 - 1680x\\
H_8(x) &= 256x^8 - 3584x^6 + 13440x^4 - 13440x^2 + 1680\\
H_9(x) &= 512x^9 - 9216x^7 + 48384x^5 - 80640x^3 + 30240x
\end{aligned}$$

GHQ roots and weights

The Gauss Hermite Quadrature integration (explored in Subesections 3.3.5 and 3.4.5), used in this thesis for the computation of the niPCE coefficients a_i, requires the roots and the weights of Hermite polynomials. In this thesis, only canonical probabilists' polynomials $\widetilde{He}$ (as formulated in (3.33)) are employed for the GHQ integration. Yet, for showcasing reasons, the weights and roots of both default and canonical probabilists' Hermite polynomials are presented in Table B.1.

	$He_n(x)$		$\widetilde{He}_n(x)$	
n	*Roots* x_j	*Weights* ω_j	*Roots* x_j	*Weights* ω_j
1	0	2.506628275	0	1
2	1	1.25331414	1	0.5
	-1	1.25331414	-1	0.5
3	1.732050808	0.417771379	1.732050808	0.1666666667
	0	1.671085516	0	0.6666666667
	-1.732050808	0.417771379	-1.732050808	0.1666666667
4	2.334414218	0.11499371	2.334414218	0.04587585477
	0.7419637843	1.13832042	0.7419637843	0.4541241452
	-0.7419637843	1.13832042	-0.7419637843	0.4541241452
	-2.334414218	0.11499371	-2.334414218	0.04587585477
5	2.856970014	0.028218146	2.856970014	0.01125741133
	1.35562618	0.55666179	1.35562618	0.222075922
	0	1.336868413	0	0.5333333333
	-1.35562618	0.55666179	-1.35562618	0.222075922
	-2.856970014	0.028218146	-2.856970014	0.01125741133
6	3.324257434	0.0064064014	3.324257434	0.002555784402
	1.889175878	0.22212673	1.889175878	0.08861574604
	0.6167065902	1.02478100	0.6167065902	0.4088284696
	-0.6167065902	1.02478100	-0.6167065902	0.4088284696
	-1.889175878	0.22212673	-1.889175878	0.08861574604
	-3.324257434	0.0064064014	-3.324257434	0.002555784402

Table B.1: *GHQ Integration: Roots and weights of probabilists' $He_n(x)$ as well as canonical probabilists' polynomials $\widetilde{He}_n(x)$ for their degrees n spaning from 1 to 6.*

Bibliography

[1] Κ. Χ. Γιαννάκογλου: *Μέθοδοι Βελτιστοποίησης στην Αεροδυναμική*. Πανεπιστημιακές Εκδόσεις Ε.Μ.Π., Αθήνα, 2006.

[2] K. C. Giannakoglou, D. I. Papadimitriou, E. M. Papoutsis-Kiachagias, and C. Othmer: *Adjoint methods in CFD-based optimization - Gradient computation & beyond.* ECCOMAS 2012–European Congress on Computational Methods in Applied Sciences and Engineering, 2012.

[3] E. M. Papoutsis-Kiachagias, V. G. Asouti, and K. C. Giannakoglou: *Assessment of variants of the Method of Moments and Polynomial Chaos approaches to aerodynamic Uncertainty Quantification.* UNCECOMP 2021, 4th ECCOMAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering, 2021.

[4] K. B. Fragkos, E. M. Papoutsis-Kiachagias, and K. C. Giannakoglou: *pFOSM: An efficient algorithm for aerodynamic robust design based on continuous adjoint and matrix-vector products.* Computers & Fluids, 181:57–66, 2019.

[5] M. Martinelli and D.I. Duvigneau: *Comparison of second-order derivatives and metamodel-based Monte-Carlo approaches to estimate statistics for robust design of a transonic wing.* 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2008.

[6] A. B. Owen: *Monte Carlo theory, methods and examples.* Stanford University, 2013.

[7] D. Xiu and G. E. Karniadakis: *The Wiener-Askey polynomial chaos for stochastic differential equations.* SIAM Journal on Scientific Computing, 24(4):619–644, 2002.

[8] S. Shankaran and A. Jameson: *Robust optimal control using polynomial chaos and adjoints for systems with uncertain inputs.* 20th AIAA Computational Fluid Dynamics Conference, 2011.

[9] D. Xiu and G. E. Karniadakis: *Modeling uncertainty in flow simulations via generalized polynomial chaos.* Journal of Computational Physics, 187:137–167, 2003.

[10] S. K. Nadarajah and A. Jameson: *A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization.* AIAA Paper 2000-0667, AIAA 38th. Aerospace Sciences Meeting and Exhibit, Reno, NV, 2000.

[11] K. Shimoyama, A. Oyama, and K. Fujii: *Development of Multi-Objective Six Sigma Approach for Robust Design Optimization.* Journal of Aerospace Computing, Information, and Communication, 5(8):215–233, 2008.

[12] G. Tenant: *Six Sigma: SPC and TQM in Manufacturing and Services.* Gower Publishing, Farnham, UK, 2001.

[13] *Test Case 3: TU Berlin TurboLab Stator.* `https://aboutflow.sems.qmul.ac.uk/events/munich2016/benchmark/testcase3/`.

[14] Κ. Χ. Γιαννάκογλου: *Συνεκτικές Ροές στις Στροβιλομηχανές.* Πανεπιστημιακές Εκδόσεις Ε.Μ.Π., Αθήνα, 2006.

[15] Σ. Τσαγγάρης: *Μηχανική των Ρευστών.* Συμεών, Αθήνα, 2005.

[16] P. Spalart and S. Allmaras: *A one-equation turbulence model for aerodynamic flows.* AIAA Paper 1992-439, 30th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, USA, January 6–9 1992.

[17] F. Moukalled, M. Darwish, and L. Mangani: *The Finite Volume Method in Computational Fluid Dynamics. An Advanced Introduction with OpenFOAMand MATLAB.* Springer, 2016.

[18] D. B. Spalding: *A single formula for the law of the wall.* Journal of Applied Mechanics, 28:455–457, 1961.

[19] E. M. Papoutsis-Kiachagias: *Adjoint methods for turbulent flows, applied to shape and topology optimization and robust design.* PhD thesis, Lab. of Thermal Turbomachines, N.T.U.A., Athens, 2013.

[20] K. C. Giannakoglou, E. M. Papoutsis-Kiachagias, and I. S. Kavvadias: *On the proper treatment of grid sensitivities in continuous adjoint methods for shape optimization.* Journal of Computational Physics, volume 301 p. 1-18, 2015.

[21] K. C. Giannakoglou, E. M. Papoutsis-Kiachagias, I. S. Kavvadias, and K. T. Gkaragkounis: *Continuous Adjoint in Shape & Topology Optimization - Recent Developments & Applications.* Conference: Seminar on Adjoint CFD Methods in Industry and Research, At Wiesbaden, Germany, 2016.

[22] K. C. Giannakoglou, E. M. Papoutsis-Kiachagias, I. S. Kavvadias, and K. Th. Gkaragkounis: *Continuous Adjoint in Shape & Topology Optimization - Recent Developments & Applications.* Conference: Seminar on Adjoint CFD Methods in Industry and Research, At Wiesbaden, Germany, 2016.

[23] I. S. Kavvadias, E. M. Papoutsis-Kiachagias, and K. C. Giannakoglou: *On the proper treatment of grid sensitivities in continuous adjoint methods for shape optimization*, volume 301. Journal of Computational Physics, 2015.

[24] A. S. Zymaris, D. I. Papadimitriou, K. C. Giannakoglou, and C. Othmer: *Continuous adjoint approach to the Spalart-Allmaras turbulence model for incompressible flows.* Computers & Fluids, 38(8):1528–1538, 2009.

[25] E. M. Papoutsis-Kiachagias and K. C. Giannakoglou: *Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications.* Springer, 2014.

[26] A. S. Zymaris, D. I. Papadimitriou, K. C. Giannakoglou, and C. Othmer: *Adjoint wall-functions: A new concept for use in aerodynamic shape optimization.* Journal of Computational Physics, 229(13):5228–5245, 2010.

[27] E.M. Papoutsis-Kiachagias, V.G. Asouti, K.C. Giannakoglou, K. Gkagkas, S. Shimokawa, and E. Itakura: *Multi-point Aerodynamic Shape Optimization of Cars Based On Continuous Adjoint.* Structural and Multidisciplinary Optimization, 59(2):675–694, 2019.

[28] E. M. Papoutsis–Kiachagias and K. C. Giannakoglou: *A parameterization and mesh movement strategy based on volumetric B-splines. Applications to shape optimization.* NTUA/PCOpt/2015/01 REPORT, 2015.

[29] S. V. Patankar: *Numerical Heat Transfer and Fluid Flow.* McGraw-Hill Book Company, New York, USA, 1980.

[30] W. Wiener: *The homogeneous chaos.* American Journal of Mathematics, 60(4):897–926, 1938.

[31] W. Askey and J. Wilson: *Memoirs of the American Mathematical Society: Some basic hypergeometric polynomials that generalize Jacobi polynomials*, volume 54. AMS, Providense, Rhode Island, USA, March 2001.

[32] Κ. Χ. Γιαννάκογλου, Ι. Αναγνωστάκης, και Γ. Μπεργελές: *Αριθμητική Ανάλυση για Μηχανικούς.* Πανεπιστημιακές Εκδόσεις Ε.Μ.Π., Αθήνα, 2003.

[33] K. L. Judd, L. Maliar, S. Maliar, and R. Valero: *Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain.* Journal of Economic Dynamics & Control, pages 44–92 and 123, 2004.

[34] D. R. Burt: *Spectral Methods in Gaussian Process Approximations.* Master's Dissertation. Department of Engineering, University of Cambridge, 2018.

[35] R. G. Ghanem and P. D. Spanos: *Stochastic Finite Elements: A Spectral Approach.* Springer-Verlag, 1991.

[36] H. L. Van Trees: *Detection, Estimation and Modulation Theory: Part I. Detection, Estimation and Linear Modulation Theory.* John Wiley & Sons, 2001.

[37] O. P. Le Maitre and O. M. Knio: *Spectral Methods for Uncertainty Quantification With Applications to Computational Fluid Dynamics*, volume 718. Springer, 2010.

[38] W. Verwoerd and D.Kulasiri: *Solving the Eigenvalue Problem for a Covariance Kernel with Variable Correlation Length.* Centre for Computing and Biometrics Research Report, pages 1–12, 1999.

[39] Κ. Δ. Παπαηλιού, Κ. Μ. Μαθιουδάκης, και Κ. Χ. Γιαννάκογλου: *Εισαγωγή στις Στροβιλομηχανές.* Πανεπιστημιακές Εκδόσεις Ε.Μ.Π., Αθήνα, 2000.

[40] I. C. Vasilopoulos: *CAD-based and CAD-free Aerodynamic Shape Optimization of Turbomachinery Blade Rows using the Adjoint Method.* PhD thesis, Lab. of Thermal Turbomachines, N.T.U.A., Athens, 2020.

[41] A. S. Ladson, C. L. Hill and W. G. Johnson Jr.: *Pressure distributions from high reynolds number transonic tests of an naca 0012 airfoil in the langley 0.3-meter transonic cryogenic tunnel.* NASA Technical Memorandum, 100526, December 1987.

[42] C. A. Lyon, A. P. Broeren, P. Giguère, A. Gopalarathnam, and M. S. Selig: *Summary of Low-Speed Airfoil Data*, volume 3. SoarTech Publications, Virginia Beach, Virginia 23451, USA, 1997.