National Technical University of Athens
School of Mechanical Engineering
Lab of Thermal Turbomachines
Parallel CFD & Optimization Unit

Application of the Intrusive Polynomial Chaos Expansion to Aerodynamic Optimization based on Continuous Adjoint

Diploma Thesis of

Kyriakos Dimitrios Kantarakias

Supervisor: K.C. Giannakoglou, Professor

Athens, September 2018

National Technical University of Athens
School of Mechanical Engineering
Fluid Mechanics Department
Lab of Thermal Turbomachines
Parallel CFD & Optimization Unit

Application of the Intrusive Polynomial Chaos Expansion to Aerodynamic Optimization based on Continuous Adjoint

Diploma Thesis

Kyriakos Dimitrios Kantarakias

Supervisor: K.C. Giannakoglou, Professor

Abstract

This thesis deals with Uncertainty Quantification (UQ) through the use of the intrusive Polynomial Chaos Expansion (iPCE), in a painless manner, for the purpose of applying it in aerodynamic shape optimization cases for robust design. The iPCE is generally considered as a computationally efficient method for UQ, in comparison with other UQ methods which are in use. However, it requires restructuring the original CFD code in the case without uncertainties, which makes its application specific to the governing equations and cumbersome in terms of the programming investment required for its development.

An approach for applying the iPCE without the additional programming investment is presented, by rendering the method independent of the PDE used. The method requires only minor software modifications to the original problem, while enjoying the benefits of the iPCE in terms of the computational efficiency. The proposed approach is shown to enjoy a lower memory footprint and computational cost than the iPCE method.

Even though the method is applicable to any PDE, applications are focused on the Navier – Stokes equations and their adjoint counterpart for optimization purposes. For the computation of the gradient of the objective function, the continuous adjoint

method is used, with the Enhanced Surface Integral formulation for the sensitivity derivatives. This method is, then, combined with the iPCE approach in order to conduct robust design in aerodynamic shape optimization.

The new, painless iPCE method is applied and validated in various cases for UQ and robust design purposes and it is shown to outperform the non–intrusive PCE in terms of computational cost while being straightforward in its application.

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Μηχανολογων Μηχανικων
Τομεας Ρευστων
Εργαστηριο Θερμικων Στροβιλομηχανων
Παραλληλης ΥΡΔ & Βελτιστοποιησης

Εφαρμογή του Επεμβατικού Αναπτύγματος Πολυωνυμικού Χάους στην Αεροδυναμική Βελτιστοποίηση Μορφής με τη Συνεχή Συζυγή Μέθοδο

Διπλωματική Εργασία

Κυριάκος Δημήτριος Κανταράκιας

Επιβλέπων: Κ.Γ. Γιαννάκογλου, Καθηγητής

Περίληψη

Το κύριο θέμα της διπλωματικής εργασίας είναι η ποσοτικοποίηση αβεβαιότητας με τη χρήση του επεμβατικού αναπτύγματος πολυωνυμικού χάους. Αυτό επιτυγχάνεται με την ανάπτυξη μιας άκοπης, από πλευράς ανάπτυξης λογισμικού, μεθόδου, ενώ οι εφαρμογές επικεντρώνονται στην αεροδυναμική βελτιστοποίηση μορφής με σκοπό τον στιβαρό σχεδιασμό. Το επεμβατικό ανάπτυγμα πολυωνυμικού χάους θεωρείται ευρέως ως μια αποδοτική, ως προς το υπολογιστικό κόστος, μέθοδος για ποσοτικοποίηση αβεβαιότητας, σε σύγκριση με άλλες μεθόδους που χρησιμοποιούνται. Ωστόσο, η μέθοδος απαιτεί εκτενή επένδυση σε προγραμματισμό, καθώς χρειάζεται επαναδόμηση του κώδικα αξιολόγησης υπολογιστικής ρευστοδυναμικής δίχως αβεβαιότητες, το οποίο την καθιστά δύσκολη ως προς την εφαρμογή.

Στη διπλωματική αυτή εργασία εισάγεται μια μέθοδος για την εφαρμογή του επεμβατικού πολυωνυμικού χάους δίχως τον επιπρόσθετο προγραμματισμό που συνήθως απαιτείται, καθιστώντας τη μέθοδο ανεξάρτητη της μερικής διαφορικής εξίσωσης που μελετάται. Η μέθοδος αυτή απαιτεί ελάχιστες προγραμματιστικές παρεμβάσεις στον αρχικό κώδικα ενώ απολαμβάνει τα οφέλη του επεμβατικού αναπτύγματος πολυωνυμικού χάους ως προς το υπολογιστικό κόστος. Παράλληλα, παρουσιάζεται νέα προσέγγιση στην ανάπτυξη των εξισώσεων επεμβατικού αναπτύγματος πολυωνυμικού χάους με

στόχο την μείωση των αναγκών μνήμης και του υπολογιστικού κόστους της μεθόδου.

Οι εφαρμογές επικεντρώνονται στις εξισώσεις Navier–Stokes και τις συζυγείς τους με σκοπό τη βελτιστοποίηση. Ο υπολογισμός των παραγώγων της αντικειμενικής συνάρτησης ως προς τις μεταβλητές σχεδιασμού γίνεται με τη συνεχή συζυγή μέθοδο, σύμφωνα με την ενισχυμένη διατύπωση που οδηγεί σε παραγώγους που εκφράζονται μόνο με επιφανειακά ολοκληρώματα. Παράλληλα, μελετάται η εφαρμογή της μεθόδου του επεμβατικού αναπτύγματος πολυωνυμικού χάους στη συνεχή συζυγή μέθοδο με σκοπό τον στιβαρό σχεδιασμό αεροδυναμικών μορφών.

Acknowledgments

I would like to thank professor Giannakoglou for his mentorship and guidance. He is the most important person in my education and has showed a rare, selfless willingness to pass on his knowledge. I would also like to thank Dr. Varvara Asouti for the excellent work she did during this thesis and the graceful patience she displayed towards my ingorance. No words can properly express the gratitude I feel towards my friend Michalis Chatzimanolakis, without which this project would not have been possible, so I will let the work speak for its self. I would also like to thank my friend Jason Tranos for his help with chaos theory.

Thank you for reading this.

Acronyms

CFD	Computational Fluid Dynamics
NTUA	National Technical University of Athens
PCopt	Parallel CFD & Optimization unit
PDE	Partial Differential Equation
PCE	Polynomial Chaos Expansion
UQ	Uncertainty Quantification
PDF	Probability Density Function
iPCE	Intrusive Polynomial Chaos Expansion
niPCE	non–Intrusive Polynomial Chaos Expansion
w.r.t.	with respect to
QoI	Quantity of Interest
GQ	Gauss Quadrature
SI	Surface Integral
FI	Field Integral
E–SI	Enhanced Surface Integral
LHS	Left–Hand–Side
RHS	Right–Hand–Side

Contents

Chapter 1

Introduction

1.1 Aerodynamic Optimization

One of the greatest achievements in modern CFD is having discovered a set of mathematical laws, in the form of some PDEs, such as the Navier–Stokes equations that describe the physics of fluid flows and developing powerful numerical techniques for their solution in a discretized domain. However, merely solving these equations is usually not enough for design purposes in large industrial problems. Modern CFD applications require the optimization of an aerodynamic object, which is a heavier problem computationally.
There are two main approaches regarding aerodynamic optimization. A common approach, which was the one employed in this thesis, is the deterministic or gradient–based optimization. These methods require the computation of the gradient of an objective function w.r.t. the design variables $\vec{b}$ that control the shape to be designed and use it for the maximization/minimization of the objective function, namely

$$\frac{\partial F}{\partial \vec{b}} = \left(\frac{\partial F}{\partial b_1}, \frac{\partial F}{\partial b_2}, \ldots, \frac{\partial F}{\partial b_N} \right) \tag{1.1}$$

For instance, it is of high technological interest to seek the optimal shape of a compressor cascade with maximum static pressure rise or the shape of the wing that minimizes the drag coefficient while maximizing the lift coefficient. These types of problems are known as optimal design or shape optimization problems. Alternatively, in inverse design problems, it is desirable to find the aerodynamic shape of an object with a pre – specified performance. An example of this would be seeking

the shape of an airfoil that has a desirable pressure distribution along its contour. In both cases, it is necessary to compute the gradient of the corresponding objective function w.r.t. the design variables $\vec{b}$ that control the aerodynamic shape.

Another approach to aerodynamic optimization is the stochastic optimization, usually in the form of Evolutionary Algorithms (EA), [8]. These methods don't require the computation of the gradient of the objective function and their formulation is derived mostly on a heuristic level. They require software that solves the field flow equations and produces values of the objective function for specific values of the design variables $\vec{b}$. The main idea is to mimic the process of the evolution of systems in biology, by generating populations (or generations) of aerodynamic shapes that are defined in the domain of the design variables $\vec{b}$, evaluating their objective function and ranking them accordingly. Then, the ones with the optimal behavior are kept, (along with a few with likely a non–optimal performance, selected at random) and are selected to mate and produce offspring through the application of evolution operators such as crossover and mutation. The convergence of this method can be accelerated with various techniques, such as the use of surrogate models [12].

Regarding differences between the two methods, deterministic optimization is generally known to converge much faster but may result to a local rather than the global optimal solution. Also, they usually require an extensive investment in method and software development, that is highly dependent on the type of PDEs used. On the other hand, it is heuristically known that stochastic optimization converges to the global optimum after an infinite number of evaluations, however it doesn't require additional software development and, in its standard form, can be considerably slower than its deterministic counterpart. In large scale problems this method can be prohibitively expensive in terms of its computational cost.

1.2 Methods for Computing the Gradient

The most commonly used methods for computing the gradient of eq. 1.1 are the following.

Finite Differences

In the Finite Differences (FD) method, the gradient of eq. 1.1 is computed by the following formula,

$$\frac{\partial F}{\partial \vec{b_i}} = \frac{F(b_1, b_2, \ldots, b_i + \epsilon, \ldots, b_N) - F(b_1, b_2, \ldots, b_i - \epsilon, \ldots, b_N)}{2\epsilon} \tag{1.2}$$

The FD method is simple in terms of its application, since it doesn't require changes in the evaluation software, which only needs to be called for different values of the design variables. However, for large, industrial scale problems, this method becomes computationally intractable, since the number of evaluations required for the computation of derivates increases linearly with the number of design variables.

Complex Variable Method

The Complex Variable (CV) method [24] is of high mathematical interest. Firstly, the objective function must be defined in the complex field. If that's the case, it's derivative w.r.t. $\vec{b}$ is given by the formula

$$\frac{\partial F}{\partial \vec{b_i}} = \lim_{\epsilon \to \mathbf{0}} \frac{imag(F(b_i + i\epsilon))}{\epsilon} \tag{1.3}$$

Notice that in this case, the computation of the derivative requires only one call of the objective function per design variable; this call, however, is more expensive since the code handles complex rather than real variables. Due to this, this method is superior to Finite Differences, where two calls per design variable are required. Another advantage is that the computed derivatives are very insensitive regarding the selection of the ϵ value. A disadvantage of the method is the extra programming it requires.

Automated Differentiation

The Automated Differentiation [4] method is a set of techniques that utilizes the fact that every numerical solver of a PDE, regardless of its mathematical complexity, executes sequentialy the elementary arithmetic operations of addition, subtraction, multiplication etc, as well as the elementary functions (exponential, sinus, cosinus etc). The main idea of the method is to apply the chain rule to each of these

operations sequentially by restructuring the original CFD code which, if done properly, allows for the computation of the derivative in a computationally efficient and precise manner.

The Adjoint Method

As it was previously mentioned, the main advantage of the adjoint method [6] is its independence for the number of the design variables that control the geometry of the problem, which is a crucial property in aerodynamic problems in which the design variables could be of the order of thousands or even millions. The adjoint method can be distinguished to two main approaches, the discrete adjoint, where the primal equations (such as the Navier–Stokes Equations) are discretized first and then differentiated, and the continuous adjoint, where the inverse procedure is followed, namely the equations are first differentiated and then discretized. The two methods but can produce differences in the values of the gradient, if the computational grid is not coarse enough. At the Parallel CFD & Optimization Unit of NTUA, a great part of the past and current research is directed towards the development of continuous adjoint methods, with emphasis being laid at its different formulations that result in various expressions for the Sensitivity Derivatives (SD), namely the derivatives of the objective function w.r.t. $\vec{b}$.
In the first formulation of the continuous adjoint published [6], named the Field Integral–FI Adjoint, the expressions of the SD include among other, volume integrals of the grid sensitivities. This adjoint formulation computes the exact values of the SDs, but it may have a high computational cost since they require the computation of grid sensitivities $\delta\vec{x}/\delta\vec{b}$. Their computation with finite differences leads to a linear scaling of the computational cost with the number of design variables [10], which is acceptable for a small number of uncertain variables but prohibitive as their number increases.
In the second formulation [19], named the Surface Integral–SI Adjoint (or reduced adjoint, in some publications), the expressions of the SD only includes surface integrals, which vastly reduces the computational cost, in comparison to the FI Adjoint while compromising the accuracy of the formulation.
At the Parallel CFD & Optimization Unit of NTUA, a new formulation named the Enhanced Surface Integral (E–SI) Adjoint [10] was developed that enjoys the computational cost of the SI with the accuracy of the FI adjoint, by avoiding the computation of $\delta\vec{x}/\delta\vec{b}$ by solving a new set of PDEs named adjoint grid displacement equations, that eliminated the field integrals of the FI formulation.

Each of the aforementioned methods, for the evaluation of each solution, requires

a grid generation or grid adaptation procedure that will affect the geometry at each optimization cycle. The simple approach of re-generating the grid at each optimization cycle can be prohibitively expensive and, as a result, methods for adapting the grid between the solutions of each cycle should be considered.

1.3 Uncertainty Quantification

In the early stages of computational methods, when computational power was limited both in terms of availability and capability, the solution of a PDE or a system of PDEs in some $2D$ spatial domain usually was sufficient for design purposes. However as the capabilities of computing systems grew, so did the demands of technological applications. CFD methods every year demand more complex models, finer grids and high order discretization schemes in order to better approximate the physical world and meet the demands of design applications. As these capabilities became larger and larger, deterministic optimization methods started to emerge that aimed at the maximization/minimization of some QoI, such as the total pressure loss coefficient of a compressor, through the computation of its derivatives.
One of the most recent developments are methods for the quantification of uncertainties that occur in real–world applications. At the earliest stages, UQ was considered computationaly expensive for CFD applications, since most techniques relied on some form of a stochastic sampling approach, such as the Monte–Carlo (MC) method [13]. These techniques usually require thousands of evaluations of the problem without uncertainties in order to make fair predictions regarding the mean value and the standard deviation of the QoI and in the modern CFD this is prohibitively expensive. However in the past few years, based on the early work done by Wiener [25] and generalized by Karniadakis and Xiu [28], methods that rely on the spectral representation of uncertain quantities in a stochastic orthonormal basis have started to emerge and are being generally referred to as the Polynomial Chaos Expansion (PCE). These methods vastly outperform stochastic sampling techniques in terms of computational cost however, in their standard form, they are much more complex mathematically and usually require, to varying degrees, a noticeable investment in terms of research and programming work. As a result, a lot of research has been directed towards that area, where this thesis is focusing on. On the next few pages, the most commonly implemented methods for UQ, along with their advantages and disadvantages are discussed.

Monte–Carlo and Stochastic Sampling

The simplest and most accurate method for computing statistical moments is the Monte–Carlo technique, since it is a direct implementation of the definitions of the statistical moments. It is based on stochastic sampling, by randomly choosing the stochastic inputs to the PDE in a manner that follows the assumed PDFs of the inputs, and solving them enough times so that an estimate can be made for the mean value and the standard deviation of the QoI. This method is accurate, since it is based directly on the definition of the statistical moments of a quantity, however, in its standard form, it is prohibitively expensive in heavy CFD applications, since it requires thousands of evaluations for a single uncertain variable.

In order to overcome this problem, attemps towards more efficient stochastic sampling techniques have been developed. The quasi–Monte Carlo method [18] uses quasi–random sequences of uncertain inputs, which greatly increases the convergence rate of the method, and therefore making it more affordable. Another approach is to have the stochastic samples satisfy constraints, which makes the method independent of the number of uncertain variables. This method is referred to as Latin Hypercube sampling [17] . However, even with those benefits, those methods are not cost effective enough for CFD purposes and they mostly find applications in other areas.

Stochastic Collocation

The computation of statistical moments in the Stochastic Collocation method is based on an interpolation scheme for the QoI, which is sampled on a set of nodes in the stochastic space of the uncertain variables, with a Lagrangian interpolation being the most commonly implemented [5],[3],[26]. The proper selection of nodes is crucial to lowering the computational cost while maintaining reasonable accuracy in the computation and several relevant methods have been suggested [5],[3].

Method of Moments

The main idea of the Method of Moments is to approximate the QoI with its Taylor expansion around its mean, w.r.t. to the uncertain variables [27], and to truncate this expansion. In its most commonly implemented version, the QoI is approximated from the moments of the expansion. This method uses a second–order truncation, which makes the method accurate for small variations in the stochastic variables, and it requires the computation of first and second–order derivatives of the QoI

w.r.t. the uncertain variables. Various alternatives have been deployed, such as a higher order truncation scheme in [20], which allows for the computation of larger values of uncertainties.

Polynomial Chaos Expansion

The PCE, which is the main subject of this thesis, is a method based on the spectral representation of uncertain quantities, through the use of polynomials that are orthogonal to the PDFs of the uncertain variables [21, 14, 2, 11]. This was originally developed in [25] for normal distributions and generalized in [28] for any PDF. This method is discussed in detail in this thesis, so it won't be analyzed in this section.

1.4 The Concept of Robust Design

Most modern design techniques that rely on optimization usually distinguish the variables that affect the primal problem (such as the Navier–Stokes) into two categories, the enviromental variables and the design variables. The design variables, denoted by $\vec{b}$, are controlled by the designer and their values in CFD applications usually controls the aerodynamic shape. The enviromental variables are usually quantities, such as the flow conditions, that the designer can't control. This distinction is crucial in uncertainty quantification, since in real world applications the effects of uncertainty appear in the enviromental variables of the problem, and it is desirable to quantify the effect of these uncertainties in the optimization process in order to select the proper set of design variables that takes into account the uncertainty of the enviromental variables. In this diploma thesis, the design variables include the boundary conditions of the PDEs, the various constants in turbulence models etc.
It is imporant to note that, in robust design, the goal isn't necessarily to find the global optimum of the QoI, but to find a point around which the value of the QoI isn't affected by slight changes in the values of the enviromental variables. For practical purposes, in the case of an optimization problem where the maximization of the lift coefficient C_L of an airfoil is desirable, a robust design process wouldn't necessarily seek the airfoil that produces maximum C_L for specific values of the Mach number and the inlet flow angle. It would seek for the airfoil that maintains a high C_L even if slight changes in the values of the boundary conditions occur.

In this thesis, robust design is based on the use of the PCE and new methods regarding its theory and implementation are developed and discussed.

1.5 Overview

Lets take a look at the contents of each chapter of this thesis.

In chapter 2 an overview of both the univariate and multivariate orthogonal polynomials, which is a necessary component of the PCE method is given. Also, an introduction to the PCE method is made, with an extended mention to the niPCE and iPCE variants, and several of their intricacies are discussed.

In chapter 3 the mathematics associated with the iPCE are reformulated in order to fit the non–intrusive operations required for the method. The Galerkin projection is generalized to an operator and used to prove that the homogeneity property of linear systems holds in the case where uncertainty is modeled with the PCE. With those tools, the iPCE equations for a general linear system are derived, and their Jacobian is proven to correspond to the exact Jacobian of the iPCE equations. Also, the stability properties of the iPCE and the niPCE are thoroughly discussed. It is proven that the iPCE equations are stable given that the background solver of their deterministic counterpart converges.

In chapter 4 a method for reducing the computational cost and the memory footprint of the method is developed, while detailed algorithms for actualizing it are provided.

In chapter 5 the UQ method introduced is validated against commonly used UQ methods, such as the Monte–Carlo and the niPCE. Useful conclusions are made for the accuracy, the stability and the computational cost of the method.

In chapter 6 the E–SI variant of the continuous adjoint 2D Euler equations is presented, as a necessary background solver for the robust design process to follow. The method is derived from the SI adjoint and is based on in–house developed software at the Parallel CFD & Optimization Unit.

In chapter 7 The continuous adjoint method is applied to the iPCE equations.

In chapter 8 A robust design test case is presented, in order to validate the method presented in chapter 7.

Chapter 2

Theory of the Polynomial Chaos Expansion

In this chapter the fundamental theory behind the PCE is discussed and a mathematical framework and research conducted for the development of this thesis is presented.

2.1 Orthogonal Polynomials

A fundamental building block of the PCE theory is the use of orthogonal polynomials as an orthonormal basis to a stochastic multi-dimensional space [25].

2.1.1 Univariate Orthogonal Polynomials

For a function $w(\xi)$ defined on l $[a, b]$, which for the purposes of this thesis will correspond to a probability density function, such as the normal or the uniform distribution. For two polynomials f and g, their inner product is defined by the integral $< f, g >= \int_a^b f(\xi)g(\xi)w(\xi)d\xi$ and is referred to as the inner product of f and g w.r.t. the function w.

In the use of the PCE, it is important for the polynomials defined to be orthogonal, which is defined in the following sense: the set of polynomials $p_n(\xi)$ with $n \in$

$[0, \ldots, \infty)$ is called orthogonal for $\xi \in [a, b]$ w.r.t. the weight function $w(\xi)$ if the following property holds,

$$\int_a^b p_n(\xi) p_m(\xi) w(\xi) d\xi = \delta_{mn} < p_n, p_n > \tag{2.1}$$

where δ_{mn} is Kronecker's delta. In orthonormal polynomials

$$< p_n, p_n >= 1, n \in 0, \ldots, \infty \tag{2.2}$$

2.1.2 Multivariate Orthogonal Polynomials

In cases where the stochastic space that models the uncertainty introduced to the PDEs is multidimentional, the corresponding basis used for the PCE is described by a multidimensional orthonormal stochastic polynomial basis with a well–defined tensor product [28]. This practically means that, for many uncertain variables, a multidinmensional polynomial basis must be defined, that properly describes the stochastic space in which the problem is solved. Such polynomials and their properties are discussed in this section.
Firstly, m sequences of univariate orthogonal polynomials $p^k \equiv \{p_n^k(\xi_k)\}_{n=0}^{\infty}, k = 1, \ldots, m$ are assumed, each of which is orthogonal to a weight function $w_k(\xi_k)$, where m corresponds to the dimensionality of the stochastic space. The tensor product of two such sequences $Q = \{q_n(\xi_1)\}_{n=0}^{\infty}$ and $W = \{w_n(\xi_2)\}_{n=0}^{\infty}$ can be defined in the following sense.

$$Q \otimes W = \{q_{n_1}(\xi_1) w_{n_2}(\xi_2)\}_{n_1,n_2=0}^{\infty} = \{q_0 w_0, q_1 w_0, q_0 w_1, q_1 w_1, q_2 w_0, q_0 w_2, \ldots\} \tag{2.3}$$

which is a widely used definition of the tensor product of such sequences. Through that, the following multidimensional sequence can be defined as the full factorial combination of m univariate orthogonal polynomials, namely,

$$Z = \{Z_n\}_{n=0}^{\infty} = \otimes_{k=1}^{m} Z^k = \{Z_{n_1}^1(\xi_1) Z_{n_2}^2(\xi_2) \ldots Z_{n_m}^m(\xi_m)\}_{n_1,n_2,\ldots,n_m=0}^{\infty}\} \tag{2.4}$$

An important property of such polynomials is their orthogonality expressed by inner

products, which in mathematical terms can be expressed as

$$< f, g >= \int_{\mathcal{E}} fgW d\xi \tag{2.5}$$

with $W = \prod_{j=1}^{m} w_j(\xi_j)$. The proof of this property is beyond the scope of this thesis. In practical terms, the above multidimensional formulation of an orthonormal basis is crucial for the application of the PCE. Firstly, in terms of uncertainty, one–dimensional problems are of great academic interest but of little practical interest, since the modelling of uncertainty usually involves a large number of uncertain variables, which necessitates the use of a multidimensional basis. Secondly, the aforementioned process for the production of a polynomial basis is flexible not only in terms of the dimensionality of the problem but also the Probability Density Function (PDF) that corresponds to each variable, since each PDF corresponds to a set of univariate orthogonal polynomials.

2.1.3 Commonly used Orthogonal Polynomials

In the following section various commonly used orthogonal polynomials and their properties are discussed.

Legendre Polynomials

The Legendre polynomials $L_n(\xi)$ correspond to the uniform distribution $w(\xi) = \frac{1}{2}$, with their orthogonality domain being $\xi \in (-1, 1)$. They can be produced, as most polynomial sequences, through a recursive formula, namely

$$L_{n+1}(\xi) = \frac{(2n+1)\xi L_n(\xi) - nL_{n-1}(\xi)}{(n+1)}, \qquad L_0(\xi) = 1 \tag{2.6}$$

Their inner product is analytically written as

$$\int_{-1}^{1} L_n(\xi) L_m(\xi) \frac{1}{2} d\xi = \frac{1}{2n+1} \delta_{mn} \tag{2.7}$$

Hermite Polynomials

The probabilistic Hermite polynomials $He_n(\xi)$ correspond to the most commonly found PDF, the normal distribution for which $w(\xi) = \frac{1}{2\pi}e^{-\xi^2/2}$, with their orthogonality domain being $\xi \in (-\infty, +\infty)$ and can be generated through the recursive formula

$$He_{n+1}(\xi) = \xi He_n(\xi) - nHe_{n-1}(\xi), \qquad He_0(\xi) = 1 \tag{2.8}$$

Their inner product is analytically written as

$$\int_{-\infty}^{+\infty} He_n(\xi)He_m(\xi)\frac{1}{2\pi}e^{-\xi^2/2}d\xi = n!\delta_{mn} \tag{2.9}$$

Laguerre Polynomials

The Laguerre polynomials $La_n(\xi)$ correspond to the exponential distribution for which $w(\xi) = e^{-\xi}$, with their orthogonality domain being $\xi \in (0, +\infty)$ and can be generated through the recursive formula

$$La_{n+1}(\xi) = \frac{(2n+1-\xi)La_n(\xi) - nLa_{n-1}(\xi)}{n+1} \tag{2.10}$$

Their inner product is analytically written as

$$\int_{-\infty}^{+\infty} He_n(\xi)He_m(\xi)\frac{1}{2\pi}e^{-\xi^2/2}d\xi = n!\delta_{mn} \tag{2.11}$$

2.2 Polynomial Chaos Expansion

The usefulnes of a multidimentional orthogonal polynomial basis can be seen in the following section, where the basic concept of the PCE is discussed. In the PCE, uncertainty is introduced through a vector ξ, each component of which corresponds to a PDF $w_i(\xi_i)$ with domain $\mathcal{E}_i$. The PDF and the domain of $\vec{\xi}$, are respectively calculated by the product $W = \prod_{i=1}^{m} w_i(\xi_i)$ and $\mathcal{E} = \prod_{i=1}^{m} \mathcal{E}_i$. The PCE of a scalar

quantity $\phi = \phi(\xi)$ is

$$\phi(\xi) = \sum_{j=0}^{\infty} \phi^j Y_j(\boldsymbol{\xi}) \tag{2.12}$$

Note that the polynomials Y_j, as it was previously discussed, are orthogonal to $W(\boldsymbol{\xi}) := \prod_{j=1}^{m} w_j(\xi_j)$ and constitute a stochastic orthonormal basis of $\mathcal{E}$. This expression can also be referred as the spectral representation of $\phi(\xi)$ in the stochastic space $\mathcal{E}$, and it was proven to converge with a truncation error $\epsilon \to 0$ as $j \to \infty$ [25]. The spectral coefficients of the expansion can be computed through the Galerkin Projection of ϕ onto Y_j, namely

$$\phi^j =< \phi(\xi), Y_j > \tag{2.13}$$

The Galerkin projection of any scalar function ϕ w.r.t the polynomial Y_j is defined as

$$< \phi(\xi), Y_j >= \int_{\mathcal{E}} \phi Y_j W d\boldsymbol{\xi} \tag{2.14}$$

A result of this representation is that the mean value and the standard deviation of the uncertain quantity $\phi(\xi)$ are given by the following expressions

$$\phi^0 = \int_{\mathcal{E}} \phi Y_0 W d\boldsymbol{\xi} = \mu_\phi \tag{2.15}$$

and for the standard deviation,

$$\begin{aligned} \sigma_\phi^2 = \int_{\mathcal{E}} (\phi - \mu_\phi)^2 W d\boldsymbol{\xi} = \int_{\mathcal{E}} \left(\sum_{j=0}^{\infty} \phi^j Y_j(\xi) - \phi^0 \right)^2 W d\boldsymbol{\xi} = \\ \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \phi^j \phi^k \int_{\mathcal{E}} Y_j(\xi) Y_k(\xi) W d\boldsymbol{\xi} = \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \phi^j \phi_{jk}^{\delta} < Y_j, Y_k > \end{aligned} \tag{2.16}$$

which simply yields

$$\sigma_\phi^2 = \sum_{j=1}^{\infty} (< Y_j, Y_j > \phi^j)^2 \tag{2.17}$$

To sum up, for orthonormal polynomials where $< Y_k, Y_k >= 1$. the mean value and standard deviation of $\phi(\xi)$ are, respectively,

$$\mu_\phi = \phi^0 \qquad \sigma_\phi^2 = \sum_{j=1}^{\infty} (\phi^j)^2$$

Further statistical moments, such as skewness, curtosis etc. can be calculated by similar formulas. Therefore, knowledge of the spectral coefficients is sufficient to fully determine the statistical behavior of a function of the uncertain variables $\boldsymbol{\xi}$.

2.3 Truncation, Chaos Order and Error

For practical purposes, the PCE expression of eq. 2.21 is truncated to a finite number of terms, denoted by $q+1$, which is determined by the dimension m of the stochastic space and the highest order C of polynomials Y_j kept in the expansion, in the following way

$$q + 1 = \frac{(m + C)!}{m!C!} \tag{2.18}$$

In the literature, C is commonly referred to as the chaos order of the expansion and is defined by the user. Through that truncation, for an application to a single PDE or a system of PDEs, a system of $q + 1$, coupled PDEs or $q + 1$ coupled systems of PDEs are produced, with its unkowns being the spectral coefficients ϕ^j of the flow field variables. This system will be referred to as the PCE equations. It is important to note that the stochastic nature of $\boldsymbol{\xi}$ is modelled through the basis polynomials [28] and thus the governing PCE equations that model uncertainties are in nature deterministic. As a result, discretizations in either space x and time t are carried out by conventional deterministic techniques without the need for stochastic modifications.

In [28], the truncation error of the expansion ϵ, which is defined as

$$\epsilon = \sum_{j=q+1}^{\infty} \phi^j Y_j(\boldsymbol{\xi}) \tag{2.19}$$

tends to zero, namely, $\epsilon \to 0$ as $q \to \infty$, for $q > 2$, although in the literature the approximation of ϕ for $q = 1$ is usually sufficient.

2.4 Application to PDEs

In this section, various ways of applying the PCE to PDEs are discussed. The set of the PDEs under investigation is symbolically written in its discrete form as

$$\boldsymbol{R}(\vec{U}) = \mathbf{0} \tag{2.20}$$

with $\vec{U}$ denoting the field flow variables. This notation is used to indicate the flexibility of the PCE in terms of different sets of PDEs. A practical example would be in the case of aerodynamics, where eq. 2.20 is the Navier–Stokes equations and $\vec{U}$ the array with the conservative variables at each grid node.
In the deterministic version of the problem (i.e. in the absence of uncertainties), eq. 2.20 would be solved for the field variables $\vec{U}$ for a specific value of boundary conditions, and the QoI would be computed as some integral quantity of the entire flow field. In aerodynamics, a QoI can be the lift coefficient of an airfoil.
In the stochastic version of the problem, uncertainty would be introduced through the boundary conditions of the PDEs and/or other uncertain parameters such as gas constants, turbulence parameters etc, propagate through the entire flow field by means of the PDEs and introduce uncertainty into the QoI. The purpose of the PCE is to quantify and propagate this uncertainty.
A practical example of this would be the study of an airfoil, where the uncertain parameters are the infinite Mach number M_∞ and the infinite flow angle α_∞. In this case, the goal would be to quantify uncertainty in the lift coefficient C_L by computing not only its mean value but also its standard deviation and its PDF, that is skewed by the non–linearity of the problem.
There are two main ways of applying the PCE to such a problem, the non–intrusive PCE (niPCE) and the intrusive PCE (iPCE), to be discussed in the following sections.

2.4.1 Non–Intrusive Polynomial Chaos Expansion

The most commonly implemented method for uncertainty quantification in PDEs is the niPCE variant [16]. In this approach, the QoI, denoted by Q, is expanded as in eq. 2.21, namely

$$Q(\xi) = \sum_{j=0}^{q} Q^j Y_j(\boldsymbol{\xi}) \tag{2.21}$$

Note that the expansion is truncated, as determined by the chaos order. The polynomial basis is produced with the aforementioned means, with the univariate polynomials used corresponding to the PDFs of the uncertain variables vector ξ, and considered uncorrelated. The goal is to compute the spectral coefficients of this expansion. These are computed through Galerkin projections as in 2.14, namely

$$Q^j =< Q(\xi), Y_j >= \int_{\mathcal{E}} Q Y_j W d\boldsymbol{\xi} \tag{2.22}$$

Therefore, the niPCE requires the computation of the spectral coefficients of the QoI and, as a result, the integral $\int_{\mathcal{E}} Q Y_j W d\boldsymbol{\xi}$. This integral is approximated either through Gaussian Quadrature (GQ) or some form of sparce grid integration. Provided that the spectral coefficients have been computed, the mean value and the standard deviation of the QoI are given by

$$\mu_Q = Q^0 \qquad \sigma_Q^2 = \sum_{j=1}^{q} (Q^j)^2$$

The application of the niPCE involves the numerical computation of the integral of eq. 2.22. This is done by running the deterministic version of the solver at specific integration nodes $\boldsymbol{\xi}_i$, and computing the integral with the proper integration weights ω_i, namely

$$\int_{\mathcal{E}} Q Y_j W d\boldsymbol{\xi} = \sum_{i=1}^{a} \omega_i Q_(\boldsymbol{\xi}_i) Y_j(\boldsymbol{\xi}_i) \tag{2.23}$$

This approach greatly simplifies the application of the PCE to PDEs. Given a software capable of solving eq. 2.20 in the absence of uncertainties, UQ can be conducted by calling this software at the GQ nodes without any modifications. This practically means that the niPCE merely requires the programming of hard–coded

values of the GQ weights and nodes and a simple, managing software for calling the solver of eq. 2.20 at these nodes.
The simplicity of the niPCE method is a great advantage, since it doesn't require extensive software development, it doesn't depend on the type of PDE and is flexible in terms of the chaos order. However, the method suffers from the curse of dimensionality. The number of evaluation nodes required is given by $a = (C+1)^m$, which scales exponentialy with m. This is not an issue for a small number of uncertain variables, however in industrial applications, where the number of uncertain variables can range from thousands to even millions, this becomes computationally intractable very fast.

2.4.2 Intrusive Polynomial Chaos Expansion

Contrary to the niPCE, in the iPCE the governing PDEs are altered by introducing the PCE through the field flow variables. This, fundamental in its nature, difference, implies the need for reprogramming the PDE solver without uncertainties, so that it can handle the altered PDEs. Also, as it will be demonstrated later, this method produces results with a lower computational cost, but requires a possibly great investment in development.
Instead of expanding the QoI, the PCE is applied to the flow field variables at each node, in the following way

$$\vec{U} = \sum_{j=0}^{q} \vec{U}^j Y(\boldsymbol{\xi}) \tag{2.24}$$

In this variant, the unkowns of the problem are not the spectral coefficients of the QoI, but the spectral coefficients of the flow variables. This increases the memory requirements of the method which grow linearly with the number of uncertain variables.

In order to produce the iPCE, the field flow variables of the deterministic equations are expanded with the PCE, namely

$$\boldsymbol{R}\left(\sum_{j=0}^{q} \vec{U}^j Y(\boldsymbol{\xi})\right) = \mathbf{0} \tag{2.25}$$

The iPCE equations are derived by applying $q+1$ Galerkin projections at each node,

yielding the iPCE eqs.

$$\int_{\mathcal{E}} \boldsymbol{R}\left(\sum_{j=0}^{q} \vec{U}^{j} Y(\boldsymbol{\xi})\right) Y_k W d\boldsymbol{\xi} = \mathbf{0}\ , k = 0, \ldots, q \tag{2.26}$$

It is important to note that eq. 2.26 is merely a symbolic representation of the iPCE eqs. Their actual form depends heavily on the type of PDE, the chaos order and the number of uncertain variables and usually requires an extensive derivation by hand. The straigtforward implementation of eq. 2.26 is tedious and requires extensive mathematical development for any different chaos order and number of uncertain variables.
For the solution of eq. 2.26 an implicit (or point–implicit) method along with an iterative linear system solver can be used. This implicit approach requires the application of the PCE to the Jacobian of the PDE, along with the $\Delta\vec{U}$ vector and the RHS of the equation, namely

$$\left(\frac{\partial \boldsymbol{R}}{\partial \vec{U}}\right)_{old} \Delta\vec{U} = -\boldsymbol{R}_{old} \tag{2.27}$$

following by the updating formula

$$\vec{U}_{new} = \vec{U}_{old} + \Delta\vec{U} \tag{2.28}$$

which would require the formulation of the equations in their expanded form,

$$\left(\frac{\partial \boldsymbol{R}\left(\sum_{j=0}^{q} \vec{U}^{j} Y(\boldsymbol{\xi})\right)}{\partial \vec{U}}\right)_{old} \Delta\vec{U} = \left(-\boldsymbol{R}\left(\sum_{j=0}^{q} \vec{U}^{j} Y(\boldsymbol{\xi})\right)\right)_{old} \tag{2.29}$$

From now on, subscript *old* will be omitted. Also, eq. 2.30 is projected onto the stochastic basis Y, through the so–called Galerkin projections technique. Through that approach, eq. 2.30 is written as

$$\int_{\mathcal{E}} \left(\frac{\partial \boldsymbol{R}\left(\sum_{j=0}^{q} \vec{U}^{j} Y(\boldsymbol{\xi})\right)}{\partial \vec{U}} \Delta\vec{U}\right) Y_j W d\boldsymbol{\xi} = \int_{\mathcal{E}} -\boldsymbol{R}\left(\sum_{j=0}^{q} \vec{U}^{j} Y(\boldsymbol{\xi})\right) Y_j W d\boldsymbol{\xi} \tag{2.30}$$

Notice that the Galerkin projection is applied to the product $\frac{\partial \boldsymbol{R}}{\partial \vec{U}}\Delta\vec{U}$ and not separately to its components $\frac{\partial \boldsymbol{R}}{\partial \vec{U}}$ and $\Delta\vec{U}$. This doesn't allow for a straightforward implementation of an implicit method as in eq. 2.27 for the linearization and solution of eq. 2.26. Such a scheme would require the LHS of eq. 2.30, to be written as

$$\int_{\mathcal{E}} \left(\frac{\partial \boldsymbol{R}\left(\sum_{j=0}^{q} \vec{U}^j Y(\boldsymbol{\xi})\right)}{\partial \vec{U}} \Delta\vec{U} \right) Y_j W d\boldsymbol{\xi} = \int_{\mathcal{E}} \frac{\partial \boldsymbol{R}\left(\sum_{j=0}^{q} \vec{U}^j Y(\boldsymbol{\xi})\right)}{\partial \vec{U}} Y_j W d\boldsymbol{\xi} \Delta\vec{U}_j \tag{2.31}$$

with $\Delta\vec{U}_j$ being the spectral coefficients of $\Delta\vec{U}$ in eq. 2.31. It isn't straightforward that this property holds true, however it is crucial for the application of an implicit or point – implicit scheme. As a result, it is thoroughly discussed, proved and validated in this thesis. Most of the mathematical work done is developed so that this statement holds true.

2.5 On the Differences of the Two Variants

The two variants of the PCE, namely the niPCE and iPCE, have radical differences in terms of their application, adaptability and computational cost.
The niPCE method is superior to the iPCE in terms of simplicity, since it is relatively simple to apply. The niPCE doesn't require any changes in the deterministic version of the software, can easily be implemented for various chaos orders and numbers of uncertain variables. Also, it doesn't depend on the type of PDE used for uncertainty quantification. In constrast, the iPCE in its straightforward application, is a problem specific method that is strongly related to the intricacies of the PDE used. Extensive mathematical work is required for its application, along with a total restructuring of the deterministic solver. It isn't easily adaptable to various chaos orders and the number of uncertain variables and requires extensive development for modifications in the code. For example, if a 3D CFD software is using a specific turbulence model for the quantification of turbulence effects, the PCE of the model would have to be developed and programmed and, if a different turbulence model was desirable, that would take a similar, tedious effort for its development.

In terms of computational cost, the niPCE suffers greatly from the curse of dimensionality, as it was previously discussed, to the point where its application can even

become computationally intractable. This problem can be appeased with the use of sparse grid integration methods [22, 7]. In contrast, the iPCE is much more efficient, since as the number of uncertain variables grows, the corresponding linear system grows at a much slower rate and, as a result, significant gains are made in terms of efficiency.

In this thesis, a new method, [9], that enjoys the benefits of the iPCE in terms of its computational cost and the benefits of the niPCE in terms of its adaptability and programming simplicity is introduced. The method is formulated with a proper mathematical background and discussed in detail in section 3.

Chapter 3

Mathematical Formulation of the iPCE

In this chapter, a rigorous mathematical framework regarding the application of the iPCE is developed and discussed. Various useful definitions and proofs related to the iPCE are presented and the intricacies of their application are discussed.

3.1 The Galerkin Operator

The notion of a Galerkin projection of a scalar quantity can be extended to vectors and matrices, through a proper formulation of the Galerkin operator.
Firstly, the Galerkin projection of a scalar function $\phi(\boldsymbol{\xi})$ to the polynomial Y_j is defined as

$$\mathrm{G}^{\mathrm{q}}\left[\phi\right] =< \phi(\xi), Y_j >= \int_{\mathcal{E}} \phi Y_j W d\boldsymbol{\xi} \tag{3.1}$$

with q denoting the truncation order, and it is given by eq. 2.18. In our case, $\boldsymbol{\xi}$ is a vector of m stochastic variables with uncorrelated PDFs $w_k(\boldsymbol{\xi}_k)$, each in the appropriate domain $\mathcal{E}_k$, with their common domain being $\mathcal{E} = \prod_{k=1}^{m} \mathcal{E}_k$.
This definition can be extended to include the Galerkin projection of vectors. The Galerkin projection of order q of a vector $\vec{U}(\boldsymbol{\xi})=[U_1(\boldsymbol{\xi}), ..., U_n(\boldsymbol{\xi})]^T \in \mathbb{R}^n$ is defined

as

$$\mathrm{G^q}\left[\vec{U}\right] = [\vec{U}^0, \vec{U}^1, ..., \vec{U}^q]^T \tag{3.2}$$

with the block matrix k being denoted by $\vec{U}^k$=$[U_1^k, U_2^k, ..., U_n^k]^T \in \mathbb{R}^n,\ k = 0, \ldots, q$. Vector $\mathrm{G^q}\left[\vec{U}\right]$ will be referred to as the vector of order q, and is consisted of q such blocks. This definition is a structural reordering of the terms that result from the Galerkin projections of each of the individual components of the deterministic vector. Notice that, in the definition of the Galerkin projection of a vector in eq. 3.2, if $\vec{U}$ is a scalar quantity defined as a one–dimensional vector, then the definition trickles down to that of the Galerkin projection of a scalar, as in eq. 3.1.
A similar generalization can be made in the case of matrices. The Galerkin projection of order q of a square matrix A of dimension n with components A_{ij}=$A_{ij}(\boldsymbol{\xi})$ is defined as

$$G^q[A] = \begin{bmatrix} A^{00} & A^{01} & \ldots & A^{0q} \\ A^{10} & A^{11} & \ldots & A^{1q} \\ \vdots & \vdots & \vdots & \vdots \\ A^{q0} & A^{q1} & \ldots & A^{qq} \end{bmatrix} \tag{3.3}$$

where the (i, j) element of $A^{\lambda\mu} \in \mathbb{R}^{n\times n}$ is given by

$$A_{ij}^{\lambda\mu} = \int_{\mathcal{E}} A_{ij} Y_\lambda Y_\mu W d\boldsymbol{\xi} = \sum_{k=0}^{\infty} A_{ij}^k < Y_k, Y_\lambda, Y_\mu > \tag{3.4}$$

with $< Y_k, Y_\lambda, Y_\mu >= \int_{\mathcal{E}} Y_k Y_\lambda Y_\mu W d\boldsymbol{\xi}$.

This definition is also a structural reordering of terms resulting from the Galerkin projections of each of the individual components of the deterministic matrix. Notice that, in the definition of the Galerkin projection of a matrix in eq. 3.3, if $\vec{U}$ is a vector quantity defined as a matrix of dimension $n \times 1$, then the definition trickles down to that of the Galerkin projection of a vector, as in eq. 3.2.

3.2 The Homogeneity Property

These definitions were developed with the homogeneity property of eq. 2.31 in mind. Although the intricacies of the process are irrelevant to this thesis, a short proof of this property will be provided, in its most general form. The goal is to show that, for a matrix A and a vector $\vec{U}$, whose components are expanded using the PCE and truncated to q terms, the following property holds

$$\mathrm{G^q}\left[A\vec{U}\right] = \mathrm{G^q}\left[A\right]\mathrm{G^q}\left[\vec{U}\right] \tag{3.5}$$

For the purpose of proving the above property, all expressions are expanded using the PCE and truncated in the following manner,

$$A_{ij}{=}\sum_{k=0}^{q} A_{ij}^{k} Y_k(\boldsymbol{\xi}) \quad \text{and} \quad U_j{=}\sum_{k=0}^{q} U_j^{k} Y_k(\boldsymbol{\xi})$$

note that $i, j = 1, \ldots, n$. The PCE of the individual elements of the matrices are truncated to a finite number of terms. To prove it, let $\vec{f}{=}A\vec{U}$ or $f_i{=}A_{ij}U_j$. Then, for any $p, 0 \leq p \leq q$

$$f_i^p = (A_{ij}U_j)^p = \int_{\mathcal{E}} A_{ij}U_j Y_p W d\boldsymbol{\xi} = U_j^{\lambda}\int_{\mathcal{E}} A_{ij}Y_\lambda Y_p W d\boldsymbol{\xi} = U_j^{\lambda}A_{ij}^{\lambda p}$$

which is in fact the p^{th} element of $\mathrm{G^q}\left[A\right]\mathrm{G^q}\left[\vec{U}\right]$. This proves the general form of eq. 2.31, namely

$$G_i[\frac{\partial \boldsymbol{R}\left(\sum_{j=0}^{q}\vec{U}^j Y(\boldsymbol{\xi})\right)}{\partial \vec{U}}\Delta\vec{U}] = G_i[\frac{\partial \boldsymbol{R}\left(\sum_{j=0}^{q}\vec{U}^j Y(\boldsymbol{\xi})\right)}{\partial \vec{U}}]G_i[\Delta\vec{U}] \tag{3.6}$$

This property is critical for the application of a general and flexible iPCE method that isn't dependent on the type of PDEs solved. It allows for the discretization schemes used to be generalized in the stochastic version of the problem and also, allows for Galerkin projections to be made directly to the residuals of the discretized form of the equations. Namely, if the residual form of the PDE, in the form of eq. 2.25 can be written as a sum of products of matrices and vectors, it allows for the Galerkin projections made to trickle down to the individual matrices that the equation is consisted of.

3.3 Deriving the iPCE Equations

The aforementioned definitions and properties are given with the purpose of creating a framework that allows for the automation of the iPCE, without the need for the cumbersome research and programming work that is required for the straightforward application of the method. Again, the equations are symbolically written in their deterministic, discrete form as a function of their field flow variables $\vec{U}$ at each node of the solution grid,

$$\boldsymbol{R}(\vec{U}) = \mathbf{0} \tag{3.7}$$

A prerequisite for the application of the method described in this section is the existence of software for the solution of the deterministic PDEs (such as the Navier – Stokes equations; in this case, this software pre–exists and is made available in open–source for the purpose of this diploma thesis) that employs the following iterative scheme

$$\left(\frac{\partial \boldsymbol{R}}{\partial \vec{U}}\right)_{old} \Delta\vec{U} = -\boldsymbol{R}_{old} \tag{3.8}$$

Eq. 3.8 is solved for $\Delta\vec{U}$ followed by an updating step, $\vec{U}_{new} = \vec{U}_{old} + \Delta\vec{U}$ at each grid node. Then, the LHS and RHS are recomputed and the system is solved again, until convergence (sufficiently small value of $\|\boldsymbol{R}\|$) is reached.
As was the case in the standard iPCE, the field flow variables $\vec{U}$ are expanded using the PCE. Appropriate Galerkin projections are made to eq. 3.8 and with the use of the homogeneity property of eq. 3.6, the new stochastic iterative scheme for the solution of the iPCE equations is

$$\mathrm{G^q}\left[\frac{\partial \boldsymbol{R}}{\partial \vec{U}}\right] \mathrm{G^q}\left[\Delta\vec{U}\right] = -\,\mathrm{G^q}\left[\boldsymbol{R}\right] \tag{3.9}$$

For notation purposes, subscript *old* is ommited from eq. 3.9. Similarly to the case of the deterministic iterative scheme, eq. 3.9 is solved for the PCE spectral coefficients of the field flow variables, namely $\mathrm{G^q}\left[\Delta\vec{U}\right]$ followed by an updating step, $\mathrm{G^q}\left[\vec{U}_{new}\right] = \mathrm{G^q}\left[\vec{U}_{old}\right] + \mathrm{G^q}\left[\Delta\vec{U}\right]$ at each grid node.
Uncertainties are introduced through the boundary conditions of the problem by modeling them as a function of the vector of uncertain variables $\boldsymbol{\xi}$.
The RHS of eq. 3.9, namely $\mathrm{G^q}\left[\boldsymbol{R}\right]$ can be computed through a non–intrusive

aproach, by computing the value of the deterministic residual $\boldsymbol{R}$ at values of $\boldsymbol{\xi}$ that correspond to the quadrature nodes indicated by the stochastic polynomial tensor space Y. A similar approach can be used for the computation of the LHS $\mathrm{G}^{\mathrm{q}}\left[\frac{\partial \boldsymbol{R}}{\partial \vec{U}}\right]$ of eq. 3.9, by computing the values of the deterministic LHS, i.e. the Jacobian $\frac{\partial \boldsymbol{R}}{\partial \vec{U}}$ of the iterative scheme at specific values of $\boldsymbol{\xi}$ that correspond to the quadrature nodes used.
This approach is crucial for the automation of the iPCE, because it produces the iPCE equations without the cumbersome task of deriving them by hand and without the need for major reprogramming. The only evident prerequisite is the existence of software that solves the deterministic version of the PDEs through a linearization scheme in the form of eq. 3.8. This allows for the development of an iPCE method that is flexible in terms of the chaos order and the number of uncertain variables.

3.4 On the LHS and the System Jacobian

The method described in the previous section enjoys the advantages of flexibility and simplicity in terms of programming and application. In this section, it is proved that the method doesn't produce a different LHS than the straightforward iPCE does, by proving that the Jacobian produced by the method is the exact Jacobian of the iPCE equations. More specifically, it is proved that the application of the operator $\mathrm{G}^{\mathrm{q}}[]$ to the LHS of eq. 3.8, namely $\mathrm{G}^{\mathrm{q}}\left[\frac{\partial \boldsymbol{R}}{\partial \vec{U}}\right]$ is equal to the Jacobian that is the product of the derivative of the RHS $\mathrm{G}^{\mathrm{q}}[\boldsymbol{R}]$ w.r.t. the spectral coefficients $\mathrm{G}^{\mathrm{q}}\left[\vec{U}\right]$ of the spectral coefficients of the stochastic field flow variables, namely

$$\frac{\partial(\mathrm{G}^{\mathrm{q}}[\boldsymbol{R}])}{\partial(\mathrm{G}^{\mathrm{q}}\left[\vec{U}\right])} = G^{q}\left[\frac{\partial \boldsymbol{R}}{\partial \vec{U}}\right]$$

Recall the PCE of $\vec{U}$ as $\vec{U} = \vec{U}^{i} Y_i$ (the Einstein summation convention is implied for $i = 0, \ldots, q$), which yields $\frac{\partial \vec{U}}{\partial \vec{U}^{i}} = Y_i I$, with I the identity matrix. Therefore,

$$\frac{\partial \phi}{\partial U^{\lambda}} = Y_{\lambda} \frac{\partial \phi}{\partial \vec{U}} \tag{3.10}$$

for any scalar ϕ. Because of eq. 3.10, the (i,j) element of the (λ,μ) block of matrix $G^q\left[\frac{\partial \boldsymbol{R}}{\partial \vec{U}}\right]$ is

$$\left(\frac{\partial \boldsymbol{R}}{\partial \vec{U}}\right)_{ij}^{\lambda\mu} = \int_{\mathcal{E}} Y_\lambda Y_\mu \left(\frac{\partial \boldsymbol{R}}{\partial \vec{U}}\right)_{ij} W d\boldsymbol{\xi} = \int_{\mathcal{E}} Y_\mu \frac{\partial R_i}{\partial U_j^\lambda} W d\boldsymbol{\xi} = \left(\frac{\partial R_i}{\partial U_j^\lambda}\right)^\mu$$

which is equal to the corresponding element of $\frac{\partial(\mathrm{G}^{\mathrm{q}}[\boldsymbol{R}])}{\partial(\mathrm{G}^{\mathrm{q}}[\vec{U}])}$, namely $\frac{\partial R_i^\mu}{\partial U_j^\lambda}$.

Technically, the aforementioned relations allow for the solution of the iPCE equations as these result from the application of non–intrusive operators. Therefore, it requires minimal changes in the background software that solves the problem without uncertainties; it merely asks for a way to evaluate $\boldsymbol{R}$ and $\frac{\partial \boldsymbol{R}}{\partial \vec{U}}$ at some values of $\boldsymbol{\xi}$ or, equivalently, for some value–sets of boundary conditions and/or input parameters. We, thus, avoid the tedious mathematical development and extensive software programming associated with the application of the iPCE method. It is also flexible whenever the chaos order and/or the number of uncertain variables change; altering any of them solely requires the use of new GQ nodes and weights, in contrast to standard iPCE approaches calling for the mathematical development of the new governing PDEs and extensive re–programming. It is also important to note that the new system of equations described by eq. 3.9 is $q+1$ times larger than the original system.

3.5 Stability of the niPCE

Stability is an issue that can play a crucial role when it comes to its efficiency and accuracy. As it was previously mentioned, the main idea of the niPCE is to expand a QoI with the PCE and to compute the spectral coefficients of the expansion with appropriate Galerkin projections. Numerically, this is implemented through a numerical integration formula that requires the evaluation of $\boldsymbol{R}(\vec{U}(\boldsymbol{\xi}))$ at specific values $\boldsymbol{\xi} = \boldsymbol{\xi}_k$, with k denoting the quadrature nodes and, therefore breaking down the iPCE to a series of deterministic runs and computations of the stochastic phenomena at a post–processing level.
At this point, the stability of the code plays a crucial role, in the following sense. Depending on the type of PDE investigated and the numerical solver used and numerous other parameters, the solver that solves $\boldsymbol{R}(\vec{U}(\boldsymbol{\xi})) = 0$, for different values of $\boldsymbol{\xi}$ (and as a result for different values of the boundary conditions of the problem), must be a rigorous solver from a stability point of view. More specifically, the nu-

merical scheme used must be stable for $\boldsymbol{\xi} \in \mathcal{E}$, with $\mathcal{E} = \prod_{k=1}^{m} \mathcal{E}_k$ being the domain of the stochastic variable $\boldsymbol{\xi}$.

If the numerical scheme of the deterministic solver indeed happens to be stable for the entirety of the domain $\mathcal{E}$, then the niPCE method will lead to a succesful computation of the spectral coefficients of the QoI. However, if the numerical scheme is either unstable or conditionally stable for specific values of $\boldsymbol{\xi}$, then the two following cases are possible.
The most optimistic scenario is that the numerical scheme is conditionally stable and, therefore, a percentage of the evaluations of $\boldsymbol{R}(\vec{U}(\boldsymbol{\xi}))$ take longer than normal to complete, with the entire calculation of the niPCE being delayed. This case will be displayed later on in the application section of this thesis, and it absolutely possible to occur in CFD. As it is common in CFD solvers, certain evaluations require longer time than others and, therefore, may delay the entire computation.The least optimistic scenario is that the code becomes unstable in certain areas of domain $\boldsymbol{\xi}$, in which case the entire computation fails.

3.6 Stability of the iPCE

The stability of the numerical scheme for the solution of the iPCE equations is an important topic of discussion, since in the intrusive case, the Jacobian of the system is altered and a new Jacobian is produced. In this section, it is proved that the numerical stability of the iPCE method solely depends on the numerical stability of the iPCE solver. This practically means that when eq. 3.7 is numerically solved in its linearized form through a stable iterative scheme in the form of eq. 3.8, the corresponding numerical scheme of eq. 3.9 that solves for $\mathrm{G}^{\mathrm{q}}\left[\vec{U}(\boldsymbol{\xi})\right]$ is also stable.
In order to prove this property, it is assumed that the integrals in the involved Galerkin projections are computed by a numerical integration in through GQ, either full or sparse grid, that involves a set of quadrature weights $\mathcal{B}$ and the corresponding quadrature nodes $\mathcal{Q}$ that define c value–sets of the uncertain variables $\boldsymbol{\xi}$. These weights and nodes are denoted respectively by

$$\mathcal{B} := \{b_1, \ldots, b_c\} \ , \ \mathcal{Q} := \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_c\} \tag{3.11}$$

where $c = (C+1)^m$. It is also assumed that the chosen chaos order is such that $\vec{U}(\boldsymbol{\xi})$ is well approximated by the truncated expansion of $\vec{U}$ with q terms, namely,

$$\vec{U} = \sum_{i=0}^{q} \vec{U}^i Y_i(\boldsymbol{\xi}) \tag{3.12}$$

In what follows, $\mathrm{G^q}\left[\vec{U}^{(\kappa)}\right] \equiv [(\vec{U}^0)^{(\kappa)}, \ldots, (\vec{U}^q)^{(\kappa)}]^T$ denotes the PCE coefficient fields at the κ–th iteration of the iterative solver, eq. 3.9. It is also assumed that an existing solver of eq.3.7 is used c times, one for each $\boldsymbol{\xi} \in \mathcal{Q}$, in order to evaluate the value of $\boldsymbol{R}(\vec{U}(\boldsymbol{\xi}))$ at each integration node. Upon completion of the κ–th iteration of each run, $\vec{U}$ and $\boldsymbol{R}$ fields at each GQ node of $\mathcal{Q}$ are computed, and these are denoted by $\vec{U}'^{(\kappa)}(\boldsymbol{\xi}_i)$ and $\boldsymbol{R}'^{(\kappa)}(\boldsymbol{\xi}_i)$ respectively. As it was discussed in the previous chapter, the PCE coefficients of $\vec{U}$ and the iPCE equations residuals can be computed, in a non–intrusive manner, by the formulas

$$\begin{aligned} \left(\vec{U}'^g\right)^{(\kappa)} &= \sum_{i=1}^{c} b_i Y_g(\boldsymbol{\xi}_i) \vec{U}'^{(\kappa)}(\boldsymbol{\xi}_i) \\ \left(\boldsymbol{R}'^g\right)^{(\kappa)} &= \sum_{i=1}^{c} b_i Y_g(\boldsymbol{\xi}_i) \boldsymbol{R}'^{(\kappa)}(\boldsymbol{\xi}_i) \end{aligned} \tag{3.13}$$

In order to correlate the stability of the iPCE numerical scheme with the stability of the deterministic solver, it suffices to show that

$$(\vec{U}^g)^{(\kappa)} = (\vec{U}'^g)^{(\kappa)} \;\text{ implies }\; (\vec{U}^g)^{(\kappa+1)} = (\vec{U}'^g)^{(\kappa+1)},\; g = 0, \ldots, q \tag{3.14}$$

The solution of eq. 3.8 for each $\boldsymbol{\xi} \in \mathcal{Q}$ leads to

$$\vec{U}'^{(\kappa+1)}(\boldsymbol{\xi}_i) = \vec{U}'^{(\kappa)}(\boldsymbol{\xi}_i) - \mathcal{J}^{-1} \boldsymbol{R}'^{(\kappa)}(\boldsymbol{\xi}_i) \tag{3.15}$$

where $\mathcal{J} := \frac{\partial \boldsymbol{R}^{(\kappa)}}{\partial \vec{U}^{(\kappa)}}$ and $\boldsymbol{R}'^{(\kappa)}(\boldsymbol{\xi}_i)$ denotes the computed residuals at each Gaussian node. Assuming $\vec{U}_g^{(\kappa)} = \vec{U}_g'^{(\kappa)}$, which also results to

$$(R^g)^{(\kappa)} = (R'^g)^{(\kappa)},\; g = 0, \ldots, q \tag{3.16}$$

the application of eq. 3.15 for all $\boldsymbol{\xi} \in \mathcal{Q}$, by considering eq. 3.13, leads to

$$(\vec{U}'^{g})^{(\kappa+1)} = (\vec{U}^{g})^{(\kappa)} - \sum_{i=1}^{c} b_i Y_g(\boldsymbol{\xi}_i) \mathcal{J}^{-1}(\boldsymbol{\xi}_i) \boldsymbol{R}^{(\kappa)}(\boldsymbol{\xi}_i) \tag{3.17}$$

Moreover

$$\mathrm{G}^{\mathrm{q}}\left[\vec{U}^{(\kappa+1)}\right] = \mathrm{G}^{\mathrm{q}}\left[\vec{U}^{(\kappa)}\right] - \mathrm{G}^{\mathrm{q}}\left[\mathcal{J}^{-1}\right] \mathrm{G}^{\mathrm{q}}\left[\boldsymbol{R}^{(\kappa)}\right] \tag{3.18}$$

From eqs. 3.17 and 3.18, it can be seen that in order to prove that $(\vec{U}^{g})^{(\kappa+1)} = (\vec{U}'^{g})^{(\kappa+1)}$, $g = 0, \ldots, q$, it suffices to show that

$$\sum_{i=1}^{c} b_i Y_g(\boldsymbol{\xi}_i) \mathcal{J}^{-1}(\boldsymbol{\xi}_i) \boldsymbol{R}^{(\kappa)}(\boldsymbol{\xi}_i) = \mathrm{G}^{\mathrm{q}}\left[\mathcal{J}^{-1}\right] \mathrm{G}^{\mathrm{q}}\left[\boldsymbol{R}^{(\kappa)}\right]$$

Thus,

$$\begin{aligned}
&G^{q}[\mathcal{J}^{-1}] G^{q}[\boldsymbol{R}^{(\kappa)}]|_g = \sum_{k=0}^{q} (J^{-1})^{gk} (\boldsymbol{R}^{k})^{(\kappa)} = \\
&\sum_{i=1}^{c} b_i \mathcal{J}^{-1}(\boldsymbol{\xi}_i) Y_g(\boldsymbol{\xi}_i) \sum_{k=0}^{q} Y_k(\boldsymbol{\xi}_i) \sum_{j=1}^{c} b_j \boldsymbol{R}^{(\kappa)}(\boldsymbol{\xi}_j) Y_k(\boldsymbol{\xi}_j) = \\
&\sum_{i=1}^{c} b_i \mathcal{J}^{-1}(\boldsymbol{\xi}_i) Y_g(\boldsymbol{\xi}_i) \sum_{k=0}^{q} Y_k(\boldsymbol{\xi}_i) \boldsymbol{R}_k^{(\kappa)} = \sum_{i=1}^{c} b_i \mathcal{J}^{-1}(\boldsymbol{\xi}_i) Y_g(\boldsymbol{\xi}_i) \boldsymbol{R}^{(\kappa)}(\boldsymbol{\xi}_i)
\end{aligned}$$

Eq. 3.14 expresses the fact that the only prerequisite for the convergence of the iPCE equations is to use a solver for the problem without uncertainties that converges for all $\boldsymbol{\xi} \in \mathcal{Q}$.

Note that eqs. 3.12 and 3.16 are both approximate in the sense that they both underwent truncation. Therefore, it is expected that the results of the proposed iPCE formulation will tend to those of the corresponding niPCE (which, of course, also undergoes truncation), as the chaos order increases. It also implies that the iPCE and the niPCE converge to the same solution as the chaos order q increases.

Chapter 4

Numerical Solution the iPCE Equations

4.1 Comparison of the Computational Cost

In the previous chapter regarding the stability of the method, it was shown that the iPCE converges in as many iterations as the background solver. Therefore a theoretical comparison can be made between the cost of the iPCE and the niPCE, on a per iteration basis. The iPCE method, per iteration, includes the computation of the LHS and RHS terms of the iPCE equations, namely

$$G^q \left[\frac{\partial \boldsymbol{R}}{\partial \vec{U}} \right] \mathrm{G^q} \left[\Delta \vec{U} \right] = - \mathrm{G^q} \left[\boldsymbol{R} \right] \tag{4.1}$$

This is achieved through Gauss Quadrature based Galerkin projections of the Jacobians. After the computation, the cost of computing the residuals and numerically solving the resulting system, eq. 4.1 is also taken into account.
On the other hand, the niPCE method requires c distinct solutions of the background PDEs the cost of which on a per iteration basis includes c computations of the LHS and RHS terms of the deterministic equation

$$\frac{\partial \boldsymbol{R}}{\partial \vec{U}} \Delta \vec{U} = -\boldsymbol{R} \tag{4.2}$$

one for each $\boldsymbol{\xi} \in \mathcal{Q}$ and c solutions of the resulting systems, eq. 3.8, one for each $\boldsymbol{\xi} \in \mathcal{Q}$.

Up to this point in both the iPCE and the niPCE variants, the computational cost of forming the LHS and RHS terms within each iteration is considered, for practical purposes, to be equal in either method. This has a great practical importance in the proposed iPCE approach, where the residuals of the equations are computed in a non–intrusive manner, by computing the values of the RHS and the LHS at specific integration nodes.

4.2 Cost Reduction in the iPCE

The system of eq. 4.1 is $(q+1)^2$ times larger than its counterpart without uncertainties in 4.2, which makes it more expensive both computationally and from a memory consumption viewpoint. A method for appeasing these disadvantages can be formulated if eq. 4.1 is rewritten in block form, namely

$$\begin{bmatrix} \frac{\partial \boldsymbol{R}^{00}}{\partial \vec{U}} & \frac{\partial \boldsymbol{R}^{01}}{\partial \vec{U}} & \cdots & \frac{\partial \boldsymbol{R}^{0q}}{\partial \vec{U}} \\ \frac{\partial \boldsymbol{R}^{10}}{\partial \vec{U}} & \frac{\partial \boldsymbol{R}^{11}}{\partial \vec{U}} & \cdots & \frac{\partial \boldsymbol{R}^{1q}}{\partial \vec{U}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \boldsymbol{R}^{q0}}{\partial \vec{U}} & \frac{\partial \boldsymbol{R}^{q1}}{\partial \vec{U}} & \cdots & \frac{\partial \boldsymbol{R}^{qq}}{\partial \vec{U}} \end{bmatrix} \begin{bmatrix} \Delta \vec{U}^0 \\ \Delta \vec{U}^1 \\ \vdots \\ \Delta \vec{U}^q \end{bmatrix} = - \begin{bmatrix} \boldsymbol{R}^0 \\ \boldsymbol{R}^1 \\ \vdots \\ \boldsymbol{R}^q \end{bmatrix} \tag{4.3}$$

In industrial CFD applications, where the mere solution of eq. 4.2 can require extensive computational and storage power, the system of eq. 4.3 presents difficulties in both aspects. It is important to notice that the LHS of eq. 4.3 is heavily coupled, a problem which can be appeased if the first spectral coefficient of the field flow variables $\vec{U}^0$ is approximated by $\vec{U}(\boldsymbol{\xi} = \boldsymbol{\xi}_z)$, with $\boldsymbol{\xi}_z$ denoting being the root vector of the first order polynomials Y_1. This is done by running the software without uncertainties once. The error of this approximation is given by

$$\sum_{i=\frac{(m+2)!}{2!m!}}^{q} \vec{U}^i Y_i(\boldsymbol{\xi}_z) \tag{4.4}$$

with m denoting the number of uncertain variables. Now, assuming that $\vec{U}^0$ is well

approximated, eq. 4.2 can be written for $C = 1$

$$\frac{\partial \boldsymbol{R}^{\lambda\mu}}{\partial \vec{U}}_{ij} = \sum_{\rho=0}^{q_1} \frac{\partial \boldsymbol{R}^{\rho}}{\partial \vec{U}}_{ij} < Y_\rho, Y_\lambda, Y_\mu > \tag{4.5}$$

since $< Y_\rho, Y_\lambda, Y_\mu >= \delta_{0\rho}\delta_{\lambda\mu}$ for $1 \leq \lambda\ ,\ \mu \leq \frac{(m+1)!}{m!}$ and $0 \leq \rho \leq \frac{(m+1)!}{m!}$, eq. 4.5 is written as

$$\frac{\partial \boldsymbol{R}^{\lambda\mu}}{\partial \vec{U}}_{ij} = \delta_{\lambda\mu} \frac{\partial \boldsymbol{R}^{00}}{\partial \vec{U}}_{ij} \tag{4.6}$$

Eq. 4.6 eliminated all off–diagonal elements of the LHS of eq. 4.3 except for those in the first row and collumn. As a result, the LHS is rewritten as

$$\begin{bmatrix} \frac{\partial \boldsymbol{R}^{00}}{\partial \vec{U}} & \frac{\partial \boldsymbol{R}^{01}}{\partial \vec{U}} & \frac{\partial \boldsymbol{R}^{02}}{\partial \vec{U}} & \dots & \frac{\partial \boldsymbol{R}^{0q_1}}{\partial \vec{U}} \\ \frac{\partial \boldsymbol{R}^{10}}{\partial \vec{U}} & \frac{\partial \boldsymbol{R}^{00}}{\partial \vec{U}} & \mathbf{0} & \dots & \mathbf{0} \\ \frac{\partial \boldsymbol{R}^{20}}{\partial \vec{U}} & \mathbf{0} & \frac{\partial \boldsymbol{R}^{00}}{\partial \vec{U}} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \boldsymbol{R}^{q_1 0}}{\partial \vec{U}} & \mathbf{0} & \mathbf{0} & \dots & \frac{\partial \boldsymbol{R}^{00}}{\partial \vec{U}} \end{bmatrix} \begin{bmatrix} \Delta\vec{U}^0 \\ \Delta\vec{U}^1 \\ \Delta\vec{U}^2 \\ \vdots \\ \Delta\vec{U}^{q_1} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{R}^0 \\ \boldsymbol{R}^1 \\ \boldsymbol{R}^2 \\ \vdots \\ \boldsymbol{R}^{q_1} \end{bmatrix} \tag{4.7}$$

At this point, the appoximation that $\vec{U}^0 \approx \vec{U}(\boldsymbol{\xi} = \boldsymbol{\xi}_z)$ implies that $\Delta\vec{U}^0 \approx \mathbf{0}$. This allows for the first row and the first collumn of eq. 4.7 to be eliminated and therefore only keeping the diagonal blocks of the LHS. As a result, eq. 4.7 is rewritten as

$$\begin{bmatrix} \frac{\partial \boldsymbol{R}^{00}}{\partial \vec{U}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \frac{\partial \boldsymbol{R}^{00}}{\partial \vec{U}} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \frac{\partial \boldsymbol{R}^{00}}{\partial \vec{U}} \end{bmatrix} \begin{bmatrix} \Delta\vec{U}^1 \\ \Delta\vec{U}^2 \\ \vdots \\ \Delta\vec{U}^{q_1} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{R}^1 \\ \boldsymbol{R}^2 \\ \vdots \\ \boldsymbol{R}^{q_1} \end{bmatrix} \tag{4.8}$$

The system of eq. 4.8 is comprised of $\left(\frac{(m+1)!}{m!} + 1\right)$, decoupled, linear systems with LHS $\frac{\partial \boldsymbol{R}^{00}}{\partial \vec{U}}$ and a different RHS. Due to this, only one of these diagonal block matrices needs to be computed and stored per iteration, which greatly reduces the storage requiremets and the computational cost of the method. Also, the block $\frac{\partial \boldsymbol{R}^{0}}{\partial \vec{U}}_{ij}$ is well approximated by the solution of the problem without uncertainties, in the beginning of the previous iteration, and as a result reducing the computational cost of the method on a per iteration basis even further.
If a chaos order higher than $C = 1$ is desirable, the method is followed similarly with

the one described in the previous paragraph. The first $\frac{(m+C-1)!}{m!(C-1)!}$ spectral coefficients of the expansion are computed using the aforementioned process, while the next $\left(\frac{(m+C)!}{m!C!} - \frac{(m+C-1)!}{m!(C-1)!}\right)$ are computed with the use of eq. 4.3. In this case, even though the off-diagonal elements are not zero, because the first $\frac{(m+C-1)!}{m!(C-1)!}$ of $\vec{U}^i$ have already been computed, it holds that $\Delta\vec{U}^i \approx \mathbf{0}$. This practically means that the off–diagonal blocks of eq. 4.3 can be neglected for those high order terms. Also,

$$\frac{\partial \boldsymbol{R}^{kk}}{\partial \vec{U}_{ij}} = \frac{\partial \boldsymbol{R}^{\rho}}{\partial \vec{U}_{ij}} < Y_\rho, Y_k, Y_k > \approx \frac{\partial \boldsymbol{R}^{00}}{\partial \vec{U}_{ij}} \tag{4.9}$$

In eq. 4.9 no summation for k is implied. Also the blocks that correspond to the spectral coefficients of $C > 2$ can be neglected in comparison with $\frac{\partial \boldsymbol{R}^{00}}{\partial \vec{U}_{ij}}$. This means that the system for $C > 2$ trickles down to the one of eq. 4.8.
It is important to note that in the computation of the LHS of eq. 4.8, there is no need to re–compute the diagonal elements of the decoupled version at each iteration, since those remain constant and equal with each other. As a consequence, the memory requirements for the storage of the LHS of the iPCE equations are exactly the same with the storage requirements for the LHS in the absence of uncertainties. This also reduces the computational cost, merely by avoiding the part of the iteration where the LHS is built again.
This property is also convenient in terms of programming, since the block matrices involved in the system of eq. 4.8 are built similarly with the LHS of eq. 4.2. This means that for the solution of the iPCE equations, the same solver that was used in the problem without uncertainties can be used without any modifications.

4.3 Flow Model and Numerical Solution

In the applications to follow, the 3D Reynolds–Averaged Navier–Stokes (RANS) equations are written in vector form for compressible flows as

$$\frac{\partial \vec{U}}{\partial t} + \frac{\partial \vec{f}_i^{inv}}{\partial x_i} - \frac{\partial \vec{f}_i^{vis}}{\partial x_i} = \mathbf{0} \tag{4.10}$$

with the flow variable vector being denoted as $\vec{U} = (\rho, \rho\, \boldsymbol{u}, E_t)^T$, in which ρ is the density, $\boldsymbol{u} = [u_1, u_2, u_3]$ the velocity vector. $E_t = \frac{p}{\gamma - 1} - \frac{1}{2}\rho\, \boldsymbol{u}^2$ the per unit volume total energy. Also p denotes the pressure, while the inviscid and viscous fluxes are

given by

$$\vec{f}_i^{inv} = \begin{pmatrix} \rho u_i \\ \rho u_i \boldsymbol{u} + p\boldsymbol{\delta}_i \\ u_i(E_t + p) \end{pmatrix} \ , \quad \vec{f}_i^{vis} = \begin{pmatrix} 0 \\ \boldsymbol{\tau}_i \\ u_j \tau_{ij} + q_i \end{pmatrix} \tag{4.11}$$

where q_i the thermal flux components, $\boldsymbol{\tau}_i = [\tau_{i1}, \tau_{i2}, \tau_{i3}]^T$ are viscous and turbulent stresses respectively and $\boldsymbol{\delta}_i$ is the Kronecker symbol. '

In the following applications, the flow model used is the one described in this section. In order to simplify notation, the Galerkin operator will be denoted by its deterministic counterpart, namely below, A and $\vec{U}$ denote $G[A]$, and $G[\vec{U}]$ respectively. The homogeneity property of the iPCE equations implied by eq. 3.6 is crucial for the application of an upwind discretization scheme that is similar to the one used in the deterministic model. By integrating the equation

$$G^q \left[\frac{\partial \boldsymbol{R}}{\partial \vec{U}} \right] \mathrm{G^q} \left[\Delta \vec{U} \right] = - \mathrm{G^q} \left[\boldsymbol{R} \right] \tag{4.12}$$

over the finite volume Ω_P, the steady flow analysis yields

$$\Omega_P \left(\frac{\vec{U}_P^{\kappa+1} - \vec{U}_P^{\kappa}}{\Delta t_P} \right) + \sum_{Q \in nei(P)} \left[\vec{\Phi}_{PQ} \right] \partial \Omega_{PQ} = 0 \tag{4.13}$$

where κ denotes the pseudo–time step counter, $\vec{\Phi}_{PQ}$ is the inviscid numerical flux crossing the interface ($\partial\Omega_{PQ}$) between two adjacent finite volumes (pointing from P to Q); $nei(P)$ stands for the set of neighbouring finite volumes of node P

The inviscid fluxes are computed using the flux vector splitting technique [15], applied between P and Q, as follows

$$\vec{\Phi}_{PQ} = A^-_{PQ} \vec{U}^R_{PQ} + A^+_{PQ} \vec{U}^L_{PQ} \tag{4.14}$$

where $A_{PQ} = \frac{\partial(\vec{f}\vec{n})}{\partial \vec{U}} = A^+_{PQ} + A^-_{PQ}$ and A^+_{PQ}, A^-_{PQ} are defined using the positive and negative eigenvalues of the Jacobian matrix. For second–order spatial accuracy, $\vec{U}^L_{PQ}$ and $\vec{U}^R_{PQ}$ (where L and R denote the two states on both sides of the interface between Ω_P and Ω_Q) are computed from $\vec{U}_P$, $\vec{U}_Q$, $\nabla\vec{U}_P$ and $\nabla\vec{U}_Q$ as follows

$$\vec{U}^L_{PQ} = \vec{U}_P + \frac{1}{2} \, (\vec{PQ}) \cdot \nabla\vec{U}_P, \qquad \vec{U}^R_{PQ} = \vec{U}_Q - \frac{1}{2} \, (\vec{PQ}) \cdot \nabla\vec{U}_Q$$

The so–computed fluxes are limited using the van Leer-van Albada limiting function [15]. Spatial gradients are computed using the Green–Gauss integration formula.

The discretized eqs. 4.13 are solved at each pseudo–time step using the point–implicit Jacobi which is written as

$$\begin{aligned} D_P^{\kappa}\Delta\vec{U}_P^{\kappa+1,\nu} + \sum_{Q\in nei(P)} Z_Q^{\kappa}\Delta\vec{U}_Q^{\kappa+1,\nu} &= -\vec{R}_P^{\kappa,\nu} \\ \vec{U}_P^{\kappa+1} &= \Delta\vec{U}_P^{\kappa+1}+\vec{U}_P^{\kappa} \end{aligned} \tag{4.15}$$

where κ is the pseudo–time counter, ν the Jacobi internal iteration counter, D_P, Z_Q stand for the diagonal and non–diagonal matrices respectively and $\vec{R}_P^{\kappa,\nu}$ is the residual array. Each Jacobi iteration comprises one iteration to solve the equations corresponding to one of the statistical moments by freezing the other terms. This flow model described above is used in the following flow cases.

4.4 Turbulence Model

In this thesis, closure is defined by the state equation of perfect gases and the Spalart–Allmaras (SA) turbulence model, [23], which in its compressible form solves the following equation for $\tilde{\mu}$,

$$\begin{aligned} \frac{\partial(\rho\tilde{\mu})}{\partial t} + \frac{\partial(\rho u_i\tilde{\mu})}{\partial x_i} &= \frac{1}{\sigma}\left[\frac{\partial}{\partial x_i}\left((\mu+\tilde{\mu})\frac{\partial\tilde{\mu}}{\partial x_i}\right) + c_{b_2}\left(\frac{\partial\tilde{\mu}}{\partial x_i}\right)^2\right] + c_{b_1}(1-f_{t_2})\tilde{S}\rho\tilde{\mu} \\ &- (c_{w_1}f_w - \frac{c_{b_1}}{\kappa^2}f_{t_2})\left(\frac{\tilde{\mu}}{d}\right)^2 + \rho^2 f_{t_1}\Delta u^2 \end{aligned} \tag{4.16}$$

with

$$\begin{aligned} &\tilde{S} = |\,\boldsymbol{\omega}| + \frac{\tilde{\mu}}{y^2\kappa^2}f_{v_2}, \quad f_{v_2} = 1 - \frac{\chi}{1+\chi f_{v_1}} \\ &f_w = g\left(\frac{1+c_{w_3}^6}{g^6+c_{w_3}^6}\right)^{1/6}, \quad g = r + c_{w_2}(r^6 - r)\ , \quad r = \frac{\tilde{\mu}}{\tilde{S}\rho\kappa^2 y^2} \end{aligned}$$

Also note that μ the fluid's dynamic viscosity and that $\boldsymbol{\omega}$ is the vorticity vector defined as $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$. Also, y denotes the distance of a grid node from the closest

solid wall. The eddy viscocity is given by

$$\mu_t = \tilde{\mu} f_{v_1}, \quad f_{v_1} = \frac{\chi^3}{\chi^3 + c_{v_1}^3}, \quad \chi = \frac{\tilde{\mu}}{\mu} \tag{4.17}$$

Also, Δx is the grid spacing along the wall at this point. At the solid walls, $\tilde{\mu} = 0$ is imposed. The constants used in the turbulence model are assigned the following values

$$\sigma = \tfrac{2}{3}, \quad \kappa = 0.41, \quad Pr_t = 0.9, \quad c_{v_1} = 7.1, \quad c_{b_1} = 0.1355, \quad c_{b_2} = 0.622,$$
$$c_{w_1} = \tfrac{c_{b_1}}{\kappa^2} + \tfrac{1+c_{b_2}}{\sigma}, \quad c_{w_2} = 0.3\ , \ c_{w_3} = 2, \quad c_{t_1} = 1, \quad c_{t_2} = 2, \quad c_{t_3} = 1.1, \quad c_{t_4} = 2$$

Chapter 5

UQ Applications in Aerodynamics

In this chapter applications of the method presented for Uncertainty Quantification to aerodynamic problems are presented and compared with the niPCE.

5.1 Inviscid Flow in the SC10 Compressor Cascade

This case deals with a Standard Configuration 10 (SC10) cascade that is consisted of the NACA0006 profile , staggered at an angle of 45^o. Solidity is equal to 1 and the 2D profile is extruded spanwise to form a $3D$ cascade, since the background solver needs to read $3D$ domains. The flow around the cascade is inviscid and transonic, with the uncertainty being introduced through the boundary conditions of the cascade, namely the inlet angle $a_1 = 58^o$ and outlet isentropic Mach number $M_{2,is} = 0.4425$ with the QoI being the static pressure rise (p_2/p_1) between the inlet (1) and outlet (2). At the nominal point, $p_2/p_1 = 1.2967$. The outlet isentropic Mach number is normally distributed with $M_{2,is} \sim N(0.4425, 0.005)$ and the inlet angle is uniformly distributed, $a_1 \sim \mathcal{U}(57^o, 59^o)$.

The mean value and the standard deviation of the static pressure rise, table 5.1 have been computed by the niPCE and iPCE variants. The results of the two methods are in very good agreement. Regarding the computational cost comparison between the two methods, for $C = 1$, the proposed iPCE method outperforms the niPCE, being faster by $\sim 20\%$. In the $C = 2$ case, the iPCE outperforms the niPCE by $\sim 10\%$. As described in chapter 3, the problem without uncertainties is solved first and then

used to initialize the mean flow field $\vec{U}^0$ for the iPCE method as well as each run of the niPCE one, in order to allow for a fair comparison of the two methods.

For the iPCE, the resulting fields of the mean Mach number and standard deviation for $C=1$ can be seen in fig. 5.1. A shockwave appears in the mean Mach field, and a noticeable increase in the values of the standard deviation of the Mach number appears at the same position, which indicates that uncertainties in its position are caused by the uncertain flow conditions. Note that the fields resulting from the niPCE solver are identical to that of the iPCE, which is expected if the discussion in chapter 3 is recalled. For this reason, the corresponding figures are the same, as it can be seen in figures 5.1 and 5.2. The niPCE fields are computed at a post-processing phase and are presented, for comparison purposes, in figure 5.2, although this isn't a common practice in the niPCE method.

	iPCE	niPCE	iPCE	niPCE
	$C=1$		$C=2$	
μ_{p_2/p_1}	1.2933	1.2936	1.2938	1.2937
σ_{p_2/p_1}	0.0257	0.0254	0.0251	0.0253
time (sec)	*1*	*1.231*	*3.900*	*4.308*

Table 5.1: *Flow in the SC10 compressor cascade. Uncertain flow conditions. Mean value and standard deviation of the static pressure rise values, comparison of iPCE and niPCE results for $C=1$ and $C=1$ and computational cost.*

Notice that the iPCE is in agreement with the niPCE in terms of the mean value and the standard deviation, with the results being closer in the case of $C=2$, than in case $C=1$. However, the increase in computational time from $C=1$ to $C=2$ is small enough that the selection of $C=2$ isn't justified by the added computational cost. No Monte–Carlo comparisons were made, since the computational demands were too high.

5.2 Flow around a 2D Isolated Airfoil

This case deals with the turbulent flow around a 2D isolated airfoil, for which an unstructured grid of 11K nodes per slice is used; as in the previous case, this is extruded in the third direction. In the absence of uncertainties, the flow conditions are the infinite flow angle $a_\infty = 2^o$ the Mach number $M_\infty = 0.3$ and the Reynolds number based on the length of the chord $Re = 10^6$. The QoIs are the lift and drag coefficients of the airfoil. In this case, uncertainties are introduced through

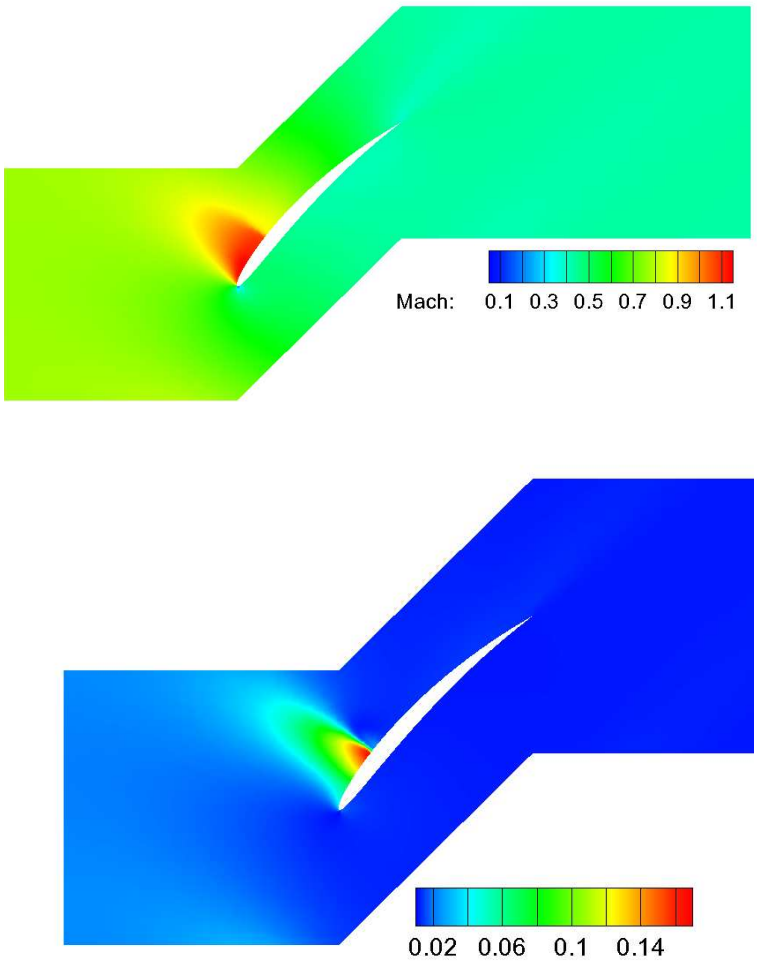


Figure 5.1: *Flow in the SC10 compressor cascade, with uncertain flow conditions. Mean field (left) and standard deviation (right) fields of the Mach number (iPCE, $C=1$). Comparison with niPCE in fig. 5.2*

the coefficients of the Spalart–Allmaras model, which are arbitrarily assumed to be uniformly distributed, centered at their values proposed in [23]. Arbitrarily, each coefficient is assumed to take on a maximum value of 105% and a minimum value of 95% of its mean value; the assumed distributions are

$$\kappa = \kappa^0 + 0.05\kappa^0\xi_2 \qquad \sigma = \sigma^0 + 0.05\sigma^0\xi_1$$
$$c_{v2} = c_{v2}^0 + 0.05c_{v2}^0\xi_4 \qquad c_{v1} = c_{v1}^0 + 0.05c_{v1}^0\xi_3$$
$$c_{b2} = c_{b2}^0 + 0.05c_{b2}^0\xi_6 \qquad c_{b1} = c_{b1}^0 + 0.05c_{b1}^0\xi_5$$
$$c_{w3} = c_{w3}^0 + 0.05c_{w3}^0\xi_8 \qquad c_{w2} = c_{w2}^0 + 0.05c_{w2}^0\xi_7$$

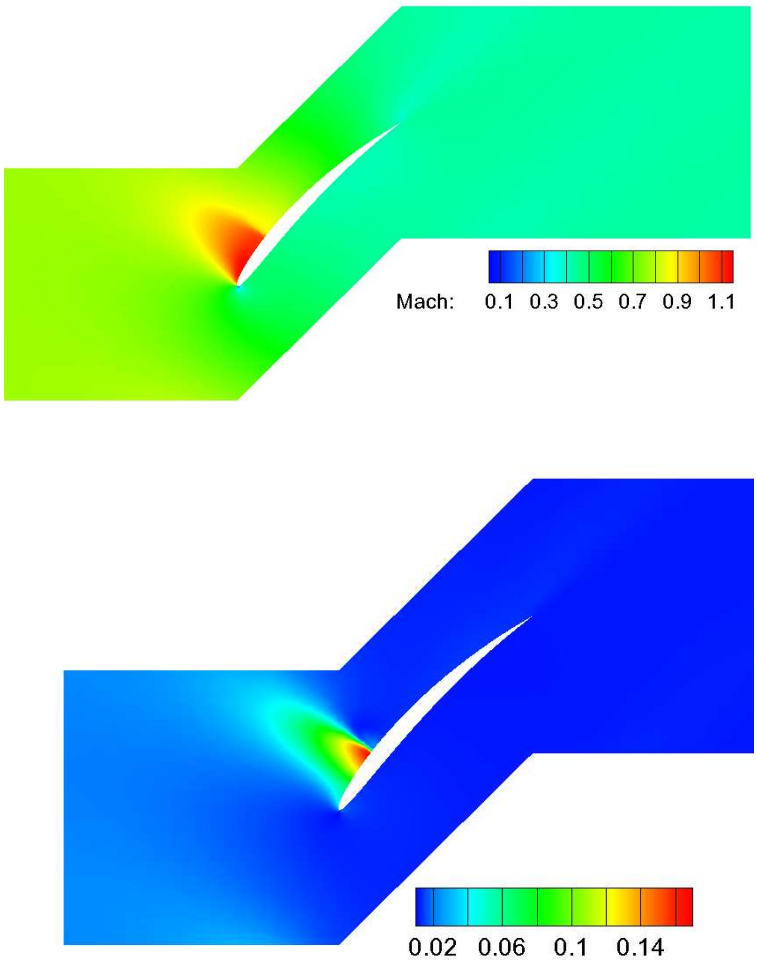


Figure 5.2: *Flow in the SC10 compressor cascade, with uncertain flow conditions. Mean field (left) and standard deviation (right) fields of the Mach number (niPCE, $C=1$). Comparison with iPCE in fig .5.1*

with $\xi_i \sim \mathcal{U}(-1,1), i=1,\ldots,8$ and $\sigma^0=2/3$, $\kappa^0=0.41$, $c_{v1}^0=7.1$, $c_{v2}^0=5$, $c_{b1}^0=0.1355$, $c_{b2}^0=0.622$, $c_{w2}^0=0.3$ and $c_{w3}^0=2$.

It is important to note that in this case where $m=8$, i.e. there are eight uncertain variables, the niPCE is asking for 256 evaluations of the flow equations (for $C=1$), for the complete grid of Gauss nodes. In this case, instead of the full GQ grid, sparse grids as in [22] are used. As a result 17 integration nodes are required and 96% less CFD evaluations are carried out for the niPCE. For a fair comparison, the iPCE method employed the same sparse grid to compute the residuals. In table 5.2, a comparison of the statistical moments as computed by the two variants, as well as their computational cost can be found .

The computed PCE coefficients of the turbulence model coefficients are given in

	iPCE	niPCE
	$C=1$	
μ_{C_L}	0.095645	0.095642
σ_{C_L}	0.000074	0.000087
μ_{C_D}	0.029479	0.029481
σ_{C_D}	0.000031	0.000149
Time Units	*1*	*1.374*

Table 5.2: *Turbulent flow around an isolated airfoil, with eight uncertain turbulence model coefficients. Statistical moments of C_L and C_D computed using iPCE and niPCE for $C=1$ and computational cost.*

table 5.3. A comparison of the results of the iPCE and niPCE demonstrates the fact that the spectral coefficients of the C_L are, for numerical purposes equal. This isn't the case for the C_D PCE coefficients. Once again, the CFD runs for the niPCE didn't converge to the same extent as in the case of the iPCE code. For this reason, the niPCE failed to properly propagate the uncertainty introduced to the flow field by uncertain values in the turbulence model coefficients to the drag coefficient. This is an intrinsic problem of the niPCE method. If the solver without uncertainties fails to converge at some of the integration nodes, the method is made vulnerable to numerical instabilities and may be slow to converge and, sometimes, may diverge. On the other hand, as was shown in chapter 3, the iPCE method is stable provided the solver without uncertainties is stable at the nominal point. Here, their difference was even greater, since the iPCE was much more effective in modelling even these small changes that are by several orders of magnitude smaller than the mean value of the C_D. In this case, the iPCE equations converged smoothly and without any problems. In the C_L case, the differences are much less pronounced. This is mainly affected by the equations of the mean flow, the residuals of which are similarly small in both the iPCE and niPCE variants.

In this case, given that some niPCE runs failed to converge adequately, the comparison of the computational cost of the two methods demonstrates the superiority of the iPCE method in terms of convergence and stability. Thus, for a qualitatively fair comparison of the two methods, 7000 iterations were performed by both methods, which is the number of iterations the iPCE method required to converge. Even with this criterion, the latter is $\sim$30% faster.

	iPCE	niPCE		iPCE	niPCE
C_L^0	0.095645	0.095642	C_D^0	0.029480	0.029482
C_L^1	$-8.147 \cdot 10^{-6}$	$-8.223 \cdot 10^{-6}$	C_D^1	$6.13 \cdot 10^{-7}$	$2.99 \cdot 10^{-5}$
C_L^2	$5.457 \cdot 10^{-5}$	$5.643 \cdot 10^{-5}$	C_D^2	$2.46 \cdot 10^{-5}$	$1.01 \cdot 10^{-4}$
C_L^3	$4.951 \cdot 10^{-5}$	$6.581 \cdot 10^{-5}$	C_D^3	$-1.71 \cdot 10^{-5}$	$-8.92 \cdot 10^{-5}$
C_L^4	$7.806 \cdot 10^{-6}$	$7.757 \cdot 10^{-6}$	C_D^4	$-1.78 \cdot 10^{-6}$	$-8.59 \cdot 10^{-6}$
C_L^5	$3.588 \cdot 10^{-6}$	$3.356 \cdot 10^{-6}$	C_D^5	$9.23 \cdot 10^{-6}$	$5.48 \cdot 10^{-5}$
C_L^6	$2.758 \cdot 10^{-6}$	$2.782 \cdot 10^{-6}$	C_D^6	$8.80 \cdot 10^{-8}$	$1.76 \cdot 10^{-6}$
C_L^7	$-3.42 \cdot 10^{-7}$	$-3.82 \cdot 10^{-7}$	C_D^7	$1.82 \cdot 10^{-6}$	$8.44 \cdot 10^{-6}$
C_L^8	$-4.0 \cdot 10^{-8}$	$-4.3 \cdot 10^{-8}$	C_D^8	$7.0 \cdot 10^{-8}$	$3.26 \cdot 10^{-7}$

Table 5.3: *Turbulent flow around an isolated airfoil, $m = 8$. PCE coefficients of C_L and C_D are computed using iPCE and niPCE, for $C = 1$.*

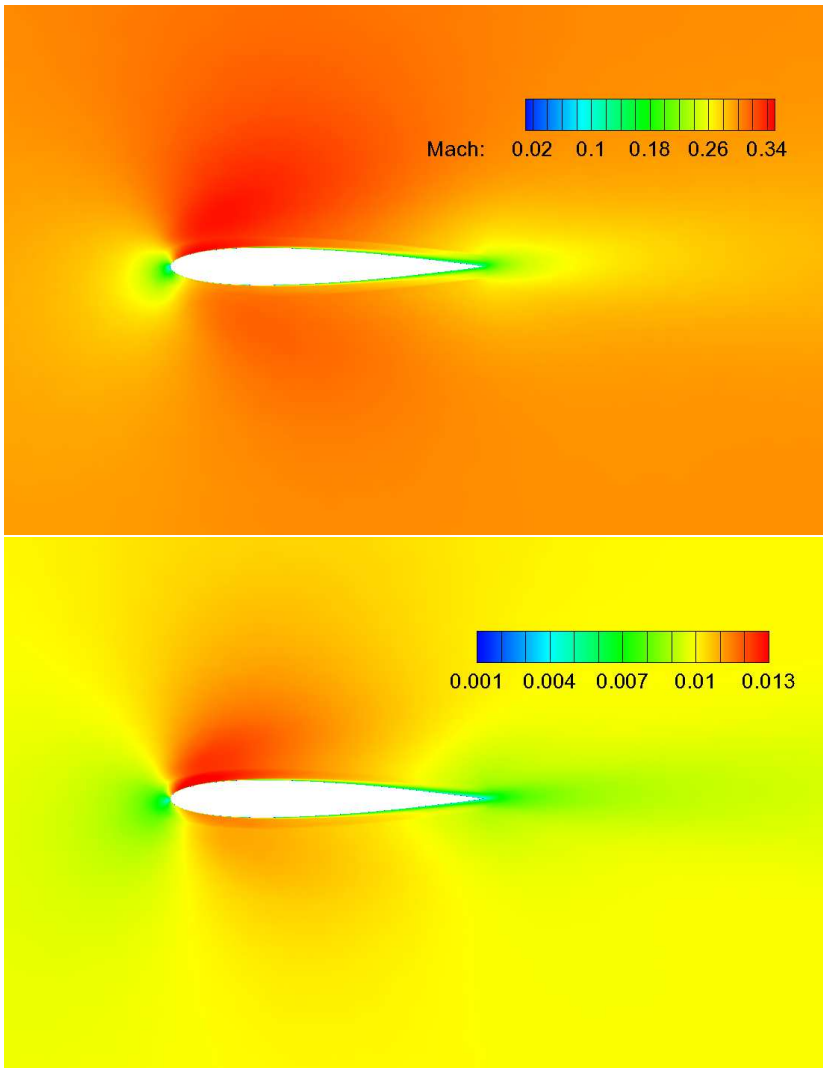


Figure 5.3: *Turbulent flow around the airfoil, with uncertain flow conditions. Computed mean (left) and standard deviation (right) fields of the Mach number (iPCE, C=1).*

Chapter 6

Continuous Adjoint with the E–SI Formulation

In this chapter, an overview of the Continuous Adjoint method is presented in the absence of uncertainties, for the 2D Euler equations. Their adjoint counterpart is developed for the Surface Integral (SI) formulation . This SI formulation is later on extended for the Enhanced SI (E–SI) method [10] for handling the sensitivity derivatives.

6.1 The Three Variants

In shape optimization problems, the continuous adjoint method can be formulated in three different ways, each of which leads to the same Field Adjoint Equation and Adjoint Boundary Conditions, but different expressions for the sensitivity derivatives of the objective function with respect to the design variables.

The first formulation, [6], referred to as the Field Integrals (FI) adjoint, leads to expressions for the sensitivity derivatives that are comprised of field integrals of the variations in the grid coordinates $\vec{x}$ w.r.t. the design variables $\vec{b}$. This formulation is expensive since it requires integrating over the entire field and also due to the need of computing $\delta\vec{x}$. This is commonly done using finite differences, which requires two grid displacement PDEs solutions per design variable. This leads to a cost that scales linearly with the number of design variables, that eventually dominates the

time needed for computing the sensitivity derivatives. This formulation is out of the scope of this thesis.

The second formulation developed, the Surface Integral (SI) adjoint, excluded the field integrals from the computation of the sensitivity derivatives by only involving surface integrals and, as a result, reduced the computational cost of the method. However, if the grid isn't sufficiently fine, the SI adjoint may lead to inaccuracies. This holds true in some cases where the grid is coarse enough to solve the flow equations with a high accuracy, which has been shown multiple times [10].

The third formulation [10], is based on the SI adjoint and introduces an alternative expression for the sensitivity derivatives by solving the adjoint grid displacement PDE, and introducing those terms into the augmented objective function. This method has been shown to enjoy the advantages of the FI in terms of its accuracy with a computational cost that is comparable to the SI formulation, and it is discussed thoroughly in this thesis.

Herein, existing software which implements the SI adjoint method is adapted to use the theory of the E–SI method. This was later on used as a basis code for the application of the iPCE to continuous adjoint. As a result, a similar process is followed, with the SI Adjoint for the Euler equations being presented first and, then, modified for the E–SI adjoint formulation.

6.2 SI Adjoint to the Euler Equations

First the SI Adjoint for the 2D Euler equations is presented. Even though the applications of this thesis are focused on the E–SI, the SI adjoint consitutes a good basis for the presentation of the E–SI adjoint.
It is assumed that it is desirable to optimize an aerodynamic object under a criterion, expressed mathematically as the minimization of a function F, which for a 2D aifoil could be the lift coefficient, namely

$$F = L \equiv \int_{S_W} p(n_2 \cos a_\infty - n_1 \sin a_\infty) dS \tag{6.1}$$

It is also assumed that the geometry is the result of a parameterization through any relevant technique (such as the Bezier–Bernstein Polynomials, b–Splines, NURBS etc). The parameterization determines the vector of design variables $\vec{b}$, of dimension

$N\left(\vec{b} \in \mathbb{R}^N\right)$ which practically means that the values of $\vec{b}$ define the geometry of the aerodynamic object in a unique manner. Function F can be expressed as a function of the design variables $\vec{b}$ and the field flow variables $\vec{U}$ in the following manner,

$$F = F(\vec{U}, \vec{b}) \tag{6.2}$$

For optimization purposes, the derivative of the objective function w.r.t. the design variables, namely $\frac{\delta F}{\delta \vec{b}}$, is of high interest and it can be written as

$$\delta F = \frac{\partial F}{\partial \vec{U}} \delta \vec{U} + \frac{\partial F}{\partial \vec{b}} \delta \vec{b} \tag{6.3}$$

For notation purposes, the operator $\frac{\delta}{\delta \vec{b}}[.]$ has been abbreviated to $\delta[.]$. The field flow equations can be written as

$$\boldsymbol{R} = \boldsymbol{R}(\vec{U}, \vec{b}) = 0, \text{ in } \Omega$$

The derivative of the field flow with respect to the design variables is similarly expressed as

$$\delta \boldsymbol{R} = \frac{\partial \boldsymbol{R}}{\partial \vec{U}} \delta \vec{U} + \frac{\partial \boldsymbol{R}}{\partial \vec{b}} \delta \vec{b} = 0 \tag{6.4}$$

Note that $\delta \boldsymbol{R} = 0$, since the field flow equations have been satisfied in the initial geometry as well as the geometry that has been affected by $\delta \vec{b}$. Therefore, eq. 6.4 can be multiplied with a vector $\boldsymbol{\Psi}$ and added to eq. 6.3. The result of that operation will be referred to as the augmented objective function, which is defined as

$$F_{aug} = F + \int_\Omega \boldsymbol{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_i} d\Omega = F + \int_\Omega \boldsymbol{\Psi}^T A_i \frac{\partial \vec{U}}{\partial x_i} d\Omega \tag{6.5}$$

The derivative of this expression w.r.t. the design variables is written, with the help of the Leibniz rule, as

$$\delta F_{aug} = \delta F + \underbrace{\int_\Omega \boldsymbol{\Psi}^T \frac{\partial}{\partial b} \left(\frac{\partial \vec{f_i}}{\partial x_i} \right) d\Omega}_{GG} + \underbrace{\int_S \boldsymbol{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_i} n_l \delta x_l dS}_{SD} \tag{6.6}$$

Term SD becomes a part of the sensitivity derivatives of the problem, while term GG can be expanded with the use of the Green–Gauss divergence theorem as

$$\int_{\Omega} \mathbf{\Psi}^T \frac{\partial}{\partial b}\left(\frac{\partial \vec{f_i}}{\partial x_i}\right) d\Omega = \underbrace{\int_S \mathbf{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_i} n_i dS}_{A} - \int_{\Omega} \frac{\partial \mathbf{\Psi}^T}{\partial x_i} \frac{\partial f_i}{\partial b} d\Omega \tag{6.7}$$

The surface integral A is written, along the solid wall, as

$$\begin{aligned}
A &= \int_{S_W} \mathbf{\Psi}^T \delta \vec{f_i} n_i dS - \int_{S_W} \mathbf{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_l} \delta x_l n_i dS = \\
&= \int_{S_W} \mathbf{\Psi}^T \delta(\vec{f_i} n_i dS) - \int_{S_W} \mathbf{\Psi}^T \vec{f_i} \delta(n_i dS) - \int_{S_W} \mathbf{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_l} \delta x_l n_i dS = \\
&= \int_{S_W} \mathbf{\Psi}_{i+1}^T \delta(p n_i dS) - \int_{S_W} \mathbf{\Psi}^T \delta(n_i dS) - \int_{S_W} \mathbf{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_l} \delta x_l n_i dS = \\
&= \int_{S_W} \mathbf{\Psi}_{i+1}^T n_i \delta p dS + \int_{S_W} \mathbf{\Psi}_{i+1}^T \delta(n_i dS) - \int_{S_W} \mathbf{\Psi}^T \delta(n_i dS) - \int_{S_W} \mathbf{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_l} \delta x_l n_i dS = \\
&= \int_{S_W} \mathbf{\Psi}_{i+1}^T n_i \delta p dS + \int_{S_W} (\mathbf{\Psi}_{i+1}^T p - \mathbf{\Psi}^T \vec{f_i}) \delta(n_i dS) - \int_{S_W} \mathbf{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_l} \delta x_l n_i dS
\end{aligned} \tag{6.8}$$

Therefore, F_{aug} can be written as

$$\begin{aligned}
\delta F_{aug} = \delta F + \int_{\Omega} \frac{\partial \mathbf{\Psi}^T}{\partial x_i} (\frac{\partial \vec{f_i}}{\partial b}) d\Omega + \int_{S_W} \mathbf{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_i} n_l \delta x_l dS + \\
\int_{S_W} \mathbf{\Psi}_{i+1}^T n_i \delta p dS + \int_{S_W} (\mathbf{\Psi}_{i+1}^T p - \mathbf{\Psi}^T f_i) \delta(n_i dS) - \int_{S_W} \mathbf{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_l} \delta x_l n_i dS
\end{aligned} \tag{6.9}$$

In case the objective function is given by eq. A.3, its variation δF is given by

$$\delta F = \delta F_{SD} + \int_{S_W} \delta p (n_2 \cos a_\infty - n_1 \sin a_\infty) dS \tag{6.10}$$

for a δF_{SD} equal to

$$\delta F_{SD} = \int_{S_W} p(\delta n_2 \cos a_\infty - \delta n_1 \sin a_\infty) dS + \int_{S_W} p(n_2 \cos a_\infty - n_1 \sin a_\infty) \delta(dS)$$

Note that so far, in eq. 6.5 no assumptions have been made regarding the vector $\mathbf{\Psi}$. The purpose of the selection of $\mathbf{\Psi}$ is to make the expression of the derivative of the augmented objective function δF_{aug} as shown in eq. 6.21 uncorrelated to the derivative of the flow variables w.r.t. the design variables $\vec{b}$. This is achieved by solving the Field Adjoint Equation (FAE), and satisfying its boundary conditions, namely

$$\begin{aligned} R_i^\Psi = A_i^T \frac{\partial \mathbf{\Psi}}{\partial x_i} = \mathbf{0} \ , \ \text{in } \Omega \\ n_2 \cos a_\infty - n_1 \sin a_\infty + \Psi_{i+1} n_i = 0 \ , \ \text{along } S \end{aligned} \tag{6.11}$$

In order to make the linearized version of the above system of equations more diagonally dominant, a pseudo–time term is added, with the final expression of the FAE being

$$\frac{\partial \mathbf{\Psi}}{\partial \tau} + A_i^T \frac{\partial \mathbf{\Psi}}{\partial x_i} = \mathbf{0} \ , \ \text{in } \Omega \tag{6.12}$$

Note that this term will only be added to the diagonal elements of the Jacobian $\frac{\partial R_i^\Psi}{\partial \vec{U}}$ and it will not affect the value of the residual R_i^Ψ. The final expression of the sensitivity derivatives is

$$\begin{aligned} \delta F_{aug} = \delta F + \int_{S_W} \mathbf{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_i} n_l \delta x_l dS + \int_{S_W} \mathbf{\Psi}_{i+1}^T n_i \delta p dS + \\ \int_{S_W} (\mathbf{\Psi}_{i+1}^T p - \mathbf{\Psi}^T f_i) \delta(n_i dS) - \int_{S_W} \mathbf{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_l} \delta x_l n_i dS + \int_{S_{I,O}} \mathbf{\Psi}^T \frac{\partial \vec{f_i}}{\partial b} n_l dS \end{aligned} \tag{6.13}$$

6.3 E–SI Adjoint to the Euler Equations

The SI formulation for the Euler equations at section 6.2 can be expanded to the E–SI formulation that was previously described.
If, between each optimization cycle, a Laplace type grid displacement system of PDEs (gdPDE) is solved as a grid displacement method. This is expressed mathematically as

$$\boldsymbol{R}_i^m = \frac{\partial^2 m_i}{\partial x_j^2} = 0 \tag{6.14}$$

with m_i denoting the grid displacements of the grid nodes between the optimization cycles. In the boundary, m_i denotes the displacement of the boundary nodes. In order to derive the adjoint gdPDE, eq. 6.14 must be included in F_{aug}, by adding the following term

$$F_{aug} = F + \int_\Omega \boldsymbol{\Psi}^T \frac{\partial \vec{f}_i}{\partial x_i} d\Omega = F + \int_\Omega \boldsymbol{\Psi}^T A_i \frac{\partial \vec{U}}{\partial x_i} d\Omega + \int_\Omega m_i^a \boldsymbol{R}_i^m d\Omega \tag{6.15}$$

The total derivative of eq. 6.17 is written as

$$\delta F_{aug} = \delta F + \int_\Omega \boldsymbol{\Psi}^T \frac{\partial}{\partial b}\left(\frac{\partial \vec{f}_i}{\partial x_i}\right) d\Omega + \int_S \boldsymbol{\Psi}^T \frac{\partial \vec{f}_i}{\partial x_i} n_l \delta x_l d\Omega + \underbrace{\delta \int_\Omega m_i^a \boldsymbol{R}_i^m d\Omega}_{B} \tag{6.16}$$

The first two terms of eq. 6.17 are expanded as in section 6.2. Term B is expanded by applying the Green – Gauss theorem twice

$$\begin{aligned} \delta \int_\Omega m_i^a \boldsymbol{R}_i^m d\Omega = \int_S m_i^a n_j \frac{\partial(\delta x_i)}{\partial x_j} dS - \int_{S_W} \frac{\partial m_i^a}{\partial x_j} n_j \delta x_i dS + \\ \int_\Omega \frac{\partial^2 m_i^a}{\partial x_j^2} \delta x_j d\Omega + \int_S m_i^a \boldsymbol{R}_i^m n_k \delta x_k dS \end{aligned} \tag{6.17}$$

At this point it is important to note that the extra term added to the objective

function only includes variations at the grid coordinates w.r.t. the design variables, which means that the FAE and their boundary conditions remain the same. The expression of the sensitivity derivatives is the same as with eq. 6.21 with the addition of terms resulting from eq. 6.17. The system of the adjoint gdPDE (agdPDE) is written as

$$\boldsymbol{R}_k^{m^a} = \frac{\partial^2 m_k^a}{\partial x_j^2} + \Psi_i \frac{\partial \boldsymbol{R}_i}{\partial x_k} = 0 \tag{6.18}$$

The boundary conditions are obtained by zeroing the coefficients of $\frac{\partial \delta x_k}{\partial x_j}$ along all boundaries which leads to the zero Dirichlet conditions,

$$m_k^a = 0 \tag{6.19}$$

After solving the system of the agdPDE, the extra term added to the sensitivity derivative of the objective function, according to the SI adjoint formulation, is

$$-\int_{S_W} \frac{\partial m_i^a}{\partial x_j} n_j \delta x_i dS \tag{6.20}$$

The FAE and the adjoint boundary conditions remain exactly the same, since no terms that include variations of the field flow variables w.r.t. the design variables have been added to the expression of δF_{aug} as a result of the E–SI. Thus, the final expression for the sensitivity derivatives becomes

$$\begin{aligned}
\delta F_{aug} = \delta F + \int_{S_W} \boldsymbol{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_i} n_l \delta x_l dS + \int_{S_W} \boldsymbol{\Psi}_{i+1}^T n_i \delta p dS + \\
\int_{S_W} (\boldsymbol{\Psi}_{i+1}^T p - \boldsymbol{\Psi}^T \vec{f_i}) \delta(n_i dS) - \int_{S_W} \boldsymbol{\Psi}^T \frac{\partial \vec{f_i}}{\partial x_l} \delta x_l n_i dS + \\
\int_{S_{I,O}} \boldsymbol{\Psi}^T \frac{\partial \vec{f_i}}{\partial b} n_l dS - \int_{S_W} \frac{\partial m_i^a}{\partial x_j} n_j \delta x_i dS
\end{aligned} \tag{6.21}$$

Regarding the computational cost of the E–SI adjoint, the optimization loop would require the extra cost of solving the system of agdPDE, which is negligible compared to solving the primal and adjoint problems, as is the computation of term 6.20.

6.4 Objective Functions

In this section, objective functions F which are used in this work are discussed.

6.4.1 Inverse Design Problems

In inverse design problems, the goal is to achieve a desired pressure distribution along the solid wall of the aerodynamic object. This is achieved by defining the objective function as

$$F = \frac{1}{2} \int_{S_w} (p - p_{tar})^2 \, dS \tag{6.22}$$

where S_w is the solid wall that is controlled by the design variables vector $\vec{b}$, p_{tar} is the desired target pressure distribution along S_w and p is the static pressure distribution of the current solution. The derivative of the objective function w.r.t. $\vec{b}$ is given by

$$\delta F = \int_{S_w} (p - p_{tar}) \, \delta p dS + \frac{1}{2} \int_{S_w} (p - p_{tar})^2 \, \delta(dS) \tag{6.23}$$

6.4.2 Minimization Problems

In the case of minimizing the drag of an isolated airfoil while the lift coefficient C_L should be kept as close as possible to a user defined value $C_{L_{tar}}$, the objective function is defined as

$$F = (c_l - c_{l_{tar}})^2 + \beta c_d^2 \tag{6.24}$$

where β is a user – defined weight that quantitatively relates the emphasis on the drag and the lift coefficients. In that case, the derivative of the objective function is

$$\delta F = 2(c_l - c_{l_{tar}}) \delta c_l + 2\beta c_d \delta c_d \tag{6.25}$$

with

$$c_{d,l} = \frac{1}{\varpi} \int_{S_w} (q_1 cosa_\infty + q_2 sina_\infty) \, dS \tag{6.26}$$

Note that a_∞ is the infinite flow angle, $\varpi = \frac{1}{2}\rho|\vec{u}_\infty^2|c$ with c denoting the chord

of the airfoil and $(q_1, q_2) = (f_1, f_2)$ for the C_D and $(q_1, q_2) = (f_2, -f_1)$ for the C_L, where f_1, f_2 are the forces on the airfoil at the x_1 and x_2 directions respectively. For inviscid flows, these forces are due to the pressure distribution around the airfoil, and as a result eq. 6.26 for the lift and drag coefficients is rewritten as

$$c_{d,l} = \frac{1}{\varpi} \int_{S_w} p \ (q_1 cosa_\infty + q_2 sina_\infty) \ dS \tag{6.27}$$

its derivative is

$$\begin{aligned} \delta c_{d,l} &= \frac{1}{\varpi} \int_{S_w} \delta p \ (q_1 cosa_\infty + q_2 sina_\infty) \ dS \\ &+ \frac{1}{\varpi} \int_{S_w} p \ cosa_\infty \ \delta(q_1 dS) + \frac{1}{\varpi} \int_{S_w} p \ sina_\infty \ \delta(q_2 dS) \end{aligned}$$

where $(q_1, q_2) = (n_1, n_2)$ for the C_D and $(q_1, q_2) = (t_1, t_2)$ for C_L, with $(t_1, t_2) = (n_2, -n_1)$ denoting the unit vector tangent to S_w.

6.5 Discretization and Numerical Solution

For the discretization of the continuous adjoint equations a finite volume technique with a cell – centered formulation is used, as in the primal problem. After integrating the adjoint eqs, the discrete form of the scheme is written as

$$\frac{\Omega_P}{\Delta t_P} \ \Delta \vec{\Psi}_P + \sum_{Q \in nei(P)} \{ \vec{\Phi}^{\Psi,inv}_{PQ} + \vec{\Phi}^{\Psi,vis}_{PQ} \} \ \Delta \partial\Omega = 0 \tag{6.28}$$

where P is a node at the control volume Ω in which the adjoint equations are integrated, and Q is a neighbouring node connected to P with a vertex. The inviscid flow vector for the adjoint equations $\vec{\Phi}^{\Psi,inv}_{PQ}$ is written using a Roe type discretization scheme similar to the one for the primal problem, and is written as

$$\vec{\Phi}^{\Psi,inv}_{PQ} = \frac{1}{2} \ (-A^T_P \vec{\Psi}_P - A^T_Q \vec{\Psi}_Q) \ - \ \frac{1}{2} \ |\tilde{A}^T_{PQ}| \ (\vec{\Psi}_Q - \vec{\Psi}_P) \tag{6.29}$$

The pseudo – time step used for the solution is the same as in the primal problem and the solution of the system is done with the same scheme as in the primal problem.

Chapter 7

iPCE to the Adjoint Euler Equations

In this chapter, the iPCE method described in chapter 3 is applied to the formulation of the adjoint presented in chapter 6.

7.1 E–SI Adjoint to the iPCE

For the solution of the primal problem, the Galerkin Operator $\mathrm{G}^{\mathrm{q}}[.]$ is applied directly to the Euler equations, [1] in the following manner,

$$\begin{aligned} \mathrm{G}^{\mathrm{q}}\left[\frac{\partial \vec{U}}{\partial \tau}+\frac{\partial \vec{f}_i}{\partial x_i}\right]=\mathrm{G}^{\mathrm{q}}\left[\frac{\partial \vec{U}}{\partial \tau}+A_i\,\frac{\partial \vec{U}}{\partial x_i}\right]=\mathbf{0}\ \text{, in } \Omega \\ \mathrm{G}^{\mathrm{q}}\left[u_i n_i\right]=\mathbf{0}\ \text{, in } S_W \end{aligned} \tag{7.1}$$

The solution of eq. 7.1 is necessary in order to obtain values for the spectral coefficients of the field flow variables $\vec{U}_i$, which are required for the adjoint method. Now, in the presence of uncertainties, the augmented objective function is defined as the weighted sum of the spectral coefficients of F, as follows

$$J=\sum_{j=0}^{q}\zeta_j F^j \tag{7.2}$$

The augmented QoI is similarly defined as

$$J_{aug} = J + \int_{\Omega} \mathrm{G}^{\mathrm{q}}\left[\boldsymbol{\Psi}\right]^{T} \mathrm{G}^{\mathrm{q}}\left[\frac{\partial \vec{f}_i}{\partial x_i}\right] d\Omega + \int_{\Omega} m_i^a \boldsymbol{R}_i^m d\Omega \tag{7.3}$$

Note that the Galerkin operator has not been applied to the term $\int_{\Omega} m_i^a \boldsymbol{R}_i^m d\Omega$, since uncertainties do not affect the geometry of the problem in any step of this process. The variation of J_{aug} is given by

$$\begin{aligned} \delta J_{aug} = \delta J &+ \int_{\Omega} \mathrm{G}^{\mathrm{q}}\left[\boldsymbol{\Psi}^{T}\right] \mathrm{G}^{\mathrm{q}}\left[\frac{\partial}{\partial b}\left(\frac{\partial \vec{f}_i}{\partial x_i}\right)\right] d\Omega \\ &+ \int_{S} \mathrm{G}^{\mathrm{q}}\left[\boldsymbol{\Psi}^{T}\right] \mathrm{G}^{\mathrm{q}}\left[\frac{\partial \vec{f}_i}{\partial x_i} n_l \delta x_l\right] dS \\ &+ \delta \int_{\Omega} m_i^a \boldsymbol{R}_i^m d\Omega \end{aligned} \tag{7.4}$$

with

$$\begin{aligned} &\delta J = \mathrm{G}^{\mathrm{q}}\left[\zeta\right]^{T} \mathrm{G}^{\mathrm{q}}\left[\delta F\right] \\ &= \mathrm{G}^{\mathrm{q}}\left[\zeta\right]^{T} \mathrm{G}^{\mathrm{q}}\left[\int_{S_W} \delta p (n_2 \cos a_\infty - n_1 \sin a_\infty) dS\right] + \mathrm{G}^{\mathrm{q}}\left[\zeta\right]^{T} \mathrm{G}^{\mathrm{q}}\left[\delta F_{SD}\right] \end{aligned} \tag{7.5}$$

Here, F_{SD} is defined as in eq. 6.10. Similarly to the development followed in chapter 6 for the E–SI adjoint, the first field integral of eq. 7.5 can be written, with the help of the Green–Gauss theorem, as

$$\begin{aligned} \int_{\Omega} \mathrm{G}^{\mathrm{q}}\left[\boldsymbol{\Psi}^{T}\right] \mathrm{G}^{\mathrm{q}}\left[\frac{\partial}{\partial b}\left(\frac{\partial \vec{f}_i}{\partial x_i}\right)\right] d\Omega &= \int_{S} \mathrm{G}^{\mathrm{q}}\left[\boldsymbol{\Psi}^{T}\right] \mathrm{G}^{\mathrm{q}}\left[\frac{\partial \vec{f}_i}{\partial x_i} n_i\right] dS \\ &- \int_{\Omega} \mathrm{G}^{\mathrm{q}}\left[\frac{\partial \boldsymbol{\Psi}^{T}}{\partial x_i}\right] \mathrm{G}^{\mathrm{q}}\left[\frac{\partial \vec{f}_i}{\partial b}\right] d\Omega \end{aligned} \tag{7.6}$$

Therefore, it follows that

$$\delta J_{aug} = \delta J + \int_{\Omega} \frac{\partial \, \mathrm{G}^{\mathrm{q}}\left[\mathbf{\Psi}^T\right]}{\partial x_i} \mathrm{G}^{\mathrm{q}}\left[\frac{\partial \vec{f_i}}{\partial b}\right] d\Omega + \int_{S_W} \mathrm{G}^{\mathrm{q}}\left[\mathbf{\Psi}^T\right] \mathrm{G}^{\mathrm{q}}\left[\frac{\partial \vec{f_i}}{\partial x_i} n_l \delta x_l\right] dS +$$
$$\int_{S_W} \mathrm{G}^{\mathrm{q}}\left[\mathbf{\Psi}^T_{i+1}\right] \mathrm{G}^{\mathrm{q}}\left[n_i \delta p\right] dS + \int_{S_W} \left(\mathrm{G}^{\mathrm{q}}\left[\mathbf{\Psi}^T_{i+1}\right] \mathrm{G}^{\mathrm{q}}\left[p\right] - \mathrm{G}^{\mathrm{q}}\left[\mathbf{\Psi}^T\right] \mathrm{G}^{\mathrm{q}}\left[\vec{f_i}\right]\right) \mathrm{G}^{\mathrm{q}}\left[\delta(n_i dS)\right]$$
$$- \int_{S_W} \mathrm{G}^{\mathrm{q}}\left[\mathbf{\Psi}^T\right] \mathrm{G}^{\mathrm{q}}\left[\frac{\partial \vec{f_i}}{\partial x_l} \delta x_l n_i\right] dS + \int_{S_{I,O}} \mathrm{G}^{\mathrm{q}}\left[\mathbf{\Psi}^T\right] \mathrm{G}^{\mathrm{q}}\left[\frac{\partial \vec{f_i}}{\partial b} n_l\right] dS - \int_{S_W} \frac{\partial m_i^a}{\partial x_j} n_j \delta x_i dS \tag{7.7}$$

Similarly, the spectral coefficients of $\mathrm{G}^{\mathrm{q}}\left[\mathbf{\Psi}^T\right]$ are selected so that expression 7.8 is independent of variations of the spectral coefficients of the field flow variables w.r.t. the design variables. This is achieved by satisfying the iPCE FAE

$$\mathrm{G}^{\mathrm{q}}\left[\frac{\partial \mathbf{\Psi}}{\partial \tau} + A_i^T \frac{\partial \mathbf{\Psi}}{\partial x_i}\right] = \mathbf{0} \tag{7.8}$$

and its boundary conditions

$$\mathrm{G}^{\mathrm{q}}\left[\zeta(n_2 \cos a_\infty - n_1 \sin a_\infty) + \mathbf{\Psi}_{i+1} n_i\right] = \mathbf{0} \tag{7.9}$$

Note that in eq. 7.8, the pseudo–time term has been added to the FAE. The final expression of the sensitivity derivatives is

$$\delta J = \mathrm{G}^{\mathrm{q}}\left[\zeta\right]^T \mathrm{G}^{\mathrm{q}}\left[\delta F_{SD}\right] + \mathrm{G}^{\mathrm{q}}\left[\delta F^{\Psi}_{SD}\right]^T \mathrm{G}^{\mathrm{q}}\left[1\right] - \int_{S_W} \frac{\partial m_i^a}{\partial x_j} n_j \delta x_i dS \tag{7.10}$$

where

$$\mathrm{G}^{\mathrm{q}}\left[\delta F^{\Psi}_{SD}\right]^T \mathrm{G}^{\mathrm{q}}\left[1\right] = \int_{S_W} \left(\mathrm{G}^{\mathrm{q}}\left[\mathbf{\Psi}_{i+1}\right] \mathrm{G}^{\mathrm{q}}\left[p\right] - \mathrm{G}^{\mathrm{q}}\left[\mathbf{\Psi}\right]^T \mathrm{G}^{\mathrm{q}}\left[\vec{f_i}\right]\right) \delta(n_i dS) \tag{7.11}$$

7.2 Programming the E–SI iPCE

Programming the iPCE with the E–SI formulation of the Continuous Adjoint requires, at each optimization cycle, the solution of the primal PDE in its iPCE form in order to obtain the spectral coefficients of the field flow variables $\mathrm{G}^{\mathrm{q}}\left[\vec{U}\right]$, namely,

$$\mathrm{G}^{\mathrm{q}}\left[\boldsymbol{R}\right] = \mathbf{0}\ ,\ \text{in}\ \Omega \tag{7.12}$$

The solution of the FAE in its iPCE form is required to compute the adjoint field flow variables $\mathrm{G}^{\mathrm{q}}\left[\Psi\right]$, as in

$$\mathrm{G}^{\mathrm{q}}\left[\frac{\partial \boldsymbol{\Psi}}{\partial \tau} + A_i^T\ \frac{\partial \boldsymbol{\Psi}}{\partial x_i}\right] = \mathbf{0} \tag{7.13}$$

Then, the agdPDE must be solved for m_k^a,

$$\boldsymbol{R}_k^{m^a} = \frac{\partial^2 m_k^a}{\partial x_j^2} + \Psi_i\ \frac{\partial R_i}{\partial x_k} = 0 \tag{7.14}$$

Having obtained values for the spectral coefficients of the field flow variables $\mathrm{G}^{\mathrm{q}}\left[\vec{U}\right]$, the adjoint field variables $\mathrm{G}^{\mathrm{q}}\left[\Psi\right]$ and the adjoint grid displacements m_k^a, the sensitivity derivatives are computed through the formula

$$\delta J = \mathrm{G}^{\mathrm{q}}\left[\zeta\right]^T \mathrm{G}^{\mathrm{q}}\left[\delta F_{SD}\right] + \mathrm{G}^{\mathrm{q}}\left[\delta F_{SD}^{\Psi}\right]^T \mathrm{G}^{\mathrm{q}}\left[1\right] - \int_{S_W} \frac{\partial m_i^a}{\partial x_j} n_j \delta x_i dS \tag{7.15}$$

and the design variables are updated through the formula

$$\vec{b}^{new} = \vec{b}^{old} - \eta\left(\frac{dF}{d\vec{b}}\right) \tag{7.16}$$

There are numerical details for solving the Primal and the Adjoint equations summarized in the next few paragraphs.

Primal Problem

The system of the primal PDEs is solved in accordance with those mentioned in chapter 3. More specifically, a delta–formulation scheme is applied for the linearization of the iPCE equations, as in

$$\mathrm{G^q}\left[\frac{\partial \boldsymbol{R}}{\partial \vec{U}}\right]\mathrm{G^q}\left[\Delta\vec{U}\right] = -\,\mathrm{G^q}\left[\boldsymbol{R}\right] \tag{7.17}$$

The linear system of eq. 7.17 is solved for $\mathrm{G^q}\left[\Delta\vec{U}\right]$ with a Jacobi solver and the spectral coefficients of the field flow variables are updated at each iteration through the formula $\mathrm{G^q}\left[\vec{U}_{new}\right] = \mathrm{G^q}\left[\vec{U}_{old}\right] + \mathrm{G^q}\left[\Delta\vec{U}\right]$. This scheme is straightforward in its implementation and the Galerkin projections required in eq. 7.17 are found through the non – intrusive operations of chapter 3.

Adjoint Problem

The system of the iPCE FAEs is solved similarly to the system of the iPCE Primal equations, by employing the iterative scheme

$$\mathrm{G^q}\left[\frac{\partial \boldsymbol{R}^{adj}}{\partial \boldsymbol{\Psi}}\right]\mathrm{G^q}\left[\Delta\boldsymbol{\Psi}\right] = -\,\mathrm{G^q}\left[\boldsymbol{R}^{adj}\right] \tag{7.18}$$

Note that the pseudo–time term will be added to the LHS of eq. 7.18, as shown in section 6.5. Similarly to the solution of the iPCE Primal equations, the Galerkin projections required for the computing the components of eq. 7.18 are computed through non – intrusive operations. Also, as was the case in chapter 3, in the Jacobian $\mathrm{G^q}\left[\frac{\partial \boldsymbol{R}^{adj}}{\partial \boldsymbol{\Psi}}\right]$ of the iPCE FAE, only the diagonal blocks are kept in the solution, in order to reduce the memory requirements and the runtime of the solver.

Regarding the choice of the coefficients ζ_j, they must be selected by the user, depending on the desired outcome of the optimization. Since $E[\phi] = \phi^0$ and $Var[\phi] = \sum_{j=1}^{q}(\phi^j)^2$, the relative magnitude of the weights determines which spectral coefficient will be minimized with emphasis, and they must be selected with that feature in mind.

Chapter 8

Validation of the iPCE Adjoint

For the purposes of validating the iPCE adjoint method, an in–house code, developed at the Parallel CFD & Optimization Unit of NTUA, that materializes the SI–Adjoint is modified in order to employ the E–SI formulation. In this chapter, the E–SI adjoint is applied to the 2D Euler equations, in order to optimize the geometry of an airfoil. The parameterization used for the geometry of the airfoil is based on two Bezier curves, one for the pressure side and one for the suction side, as seen in figure 8.1. The coordinates of the control points of each curve are the design variables of the airfoil, with the first two and the last two control points being kept fixed.

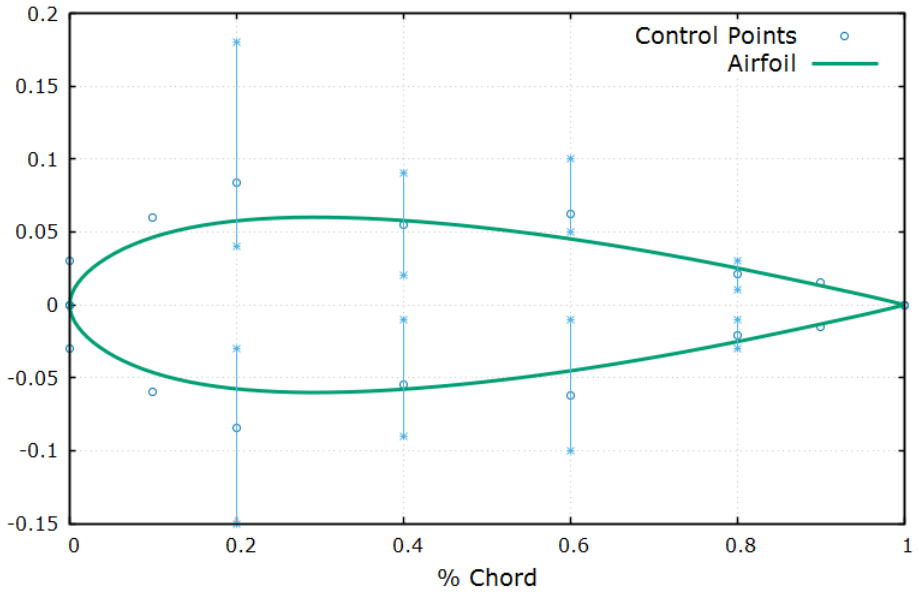


Figure 8.1: *Shape optimization of a 2D airfoil based on NACA0012. Initial airfoil geometry and control points used to create the two Bezier curves. Axes are not in scale.*

8.1 Optimization without Uncertainties

In the first case, an optimization without uncertainties is conducted for comparison purposes, with the objective function for minimization being the drag coefficient C_D of the airfoil, namely

$$F = C_D \tag{8.1}$$

The infinite flow conditions of the free stream flow of the airfoil are,

$$M_\infty = 0.55 \quad , \quad a_\infty = 2.5^o \quad , \quad Re = 5000 \tag{8.2}$$

20 optimization cycles were performed, as a basis for comparison. The resulting geometry yielded an airfoil with about 27% reduction in the drag coefficient, figure 8.2.

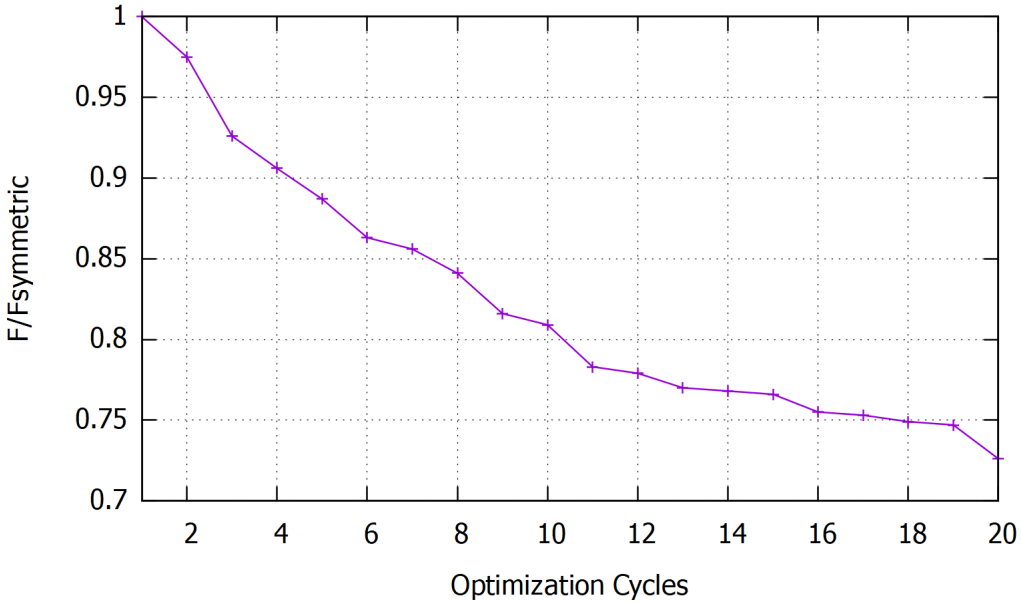


Figure 8.2: *Drag optimization without uncertainties of a 2D airfoil, history of the objective function in each optimization cycle.*

In figure 8.3 the Mach Number around the optimized airfoil without the presence of uncertainties is presented.

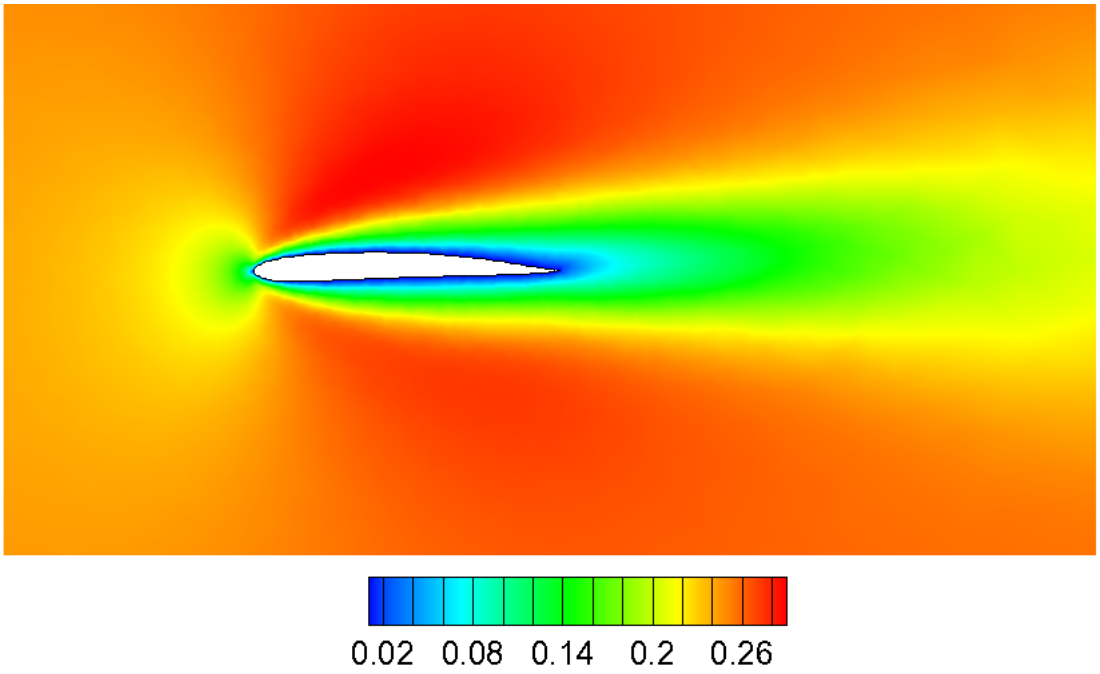


Figure 8.3: *Mach Number around the optimized airfoil geometry.*

8.2 Optimization under Uncertainties

Uncertainty is introduced through the boundary conditions of the airfoil, which follow the following PDFs,

$$M_\infty \sim \mathcal{N}(0.55, 0.05) \quad , \quad a_\infty \sim \mathcal{U}(2.0^o, 3.0^o) \quad , \quad Re \sim \mathcal{N}(5000, 200)$$

with the objective function being defined similarly as in chapter 7, namely

$$J = \sum_{j=0}^{q} \zeta_j F^j$$

The weights of the objective function are such so that the minimization of the spectral coefficients of the PCE is achieved, namely

$$\zeta_0 = -1 \ , \ \zeta_j = 3$$

These weights are by no means restrictive and depending on their values, emphasis is laid on the appropriate spectral coefficient of the objective function.

Three runs were conducted for comparison purposes, for three different values of C, namely $C = 1, 2, 3$, with q being $q = 3, 9, 19$ respectively in each case (see eq. 2.18). Recall that q is depends on the chaos order C selected and the number of uncertain variables. The geometry produced from the optimization without uncertainties was evaluated with uncertainty quantification, using the same boundary conditions used in the case without uncertainties. The results are summarized in table 8.1

	w/o UQ	with UQ, C=1	with UQ, C=2	with UQ, C=3
μ_{C_D}	$6.81 \cdot 10^{-2}$	$6.89 \cdot 10^{-2}$	$6.86 \cdot 10^{-2}$	$6.84 \cdot 10^{-2}$
σ_{C_D}	$1.17 \cdot 10^{-3}$	$1.09 \cdot 10^{-3}$	$1.05 \cdot 10^{-3}$	$1.02 \cdot 10^{-3}$

Table 8.1: *Comparison of optimization without uncertainties and under uncertainties, for* $C = 1, 2, 3$

As expected, the UQ that was applied on the geometry that resulted from the optimization without uncertainties results in lower mean value but has higher standard deviation of the drag coefficient compared to the mean value and standard deviation that resulted from the optimization under uncertainties. As we see, the chaos order affected the optimization results, by giving a smaller standard deviation as the chaos

order increased, for the same number of optimization cycles.
In figure 8.4, the airfoil produced from the optimization under uncertainty is presented and compared with the result of the optimization without uncertainties, as well as the original, symmetric airfoil.

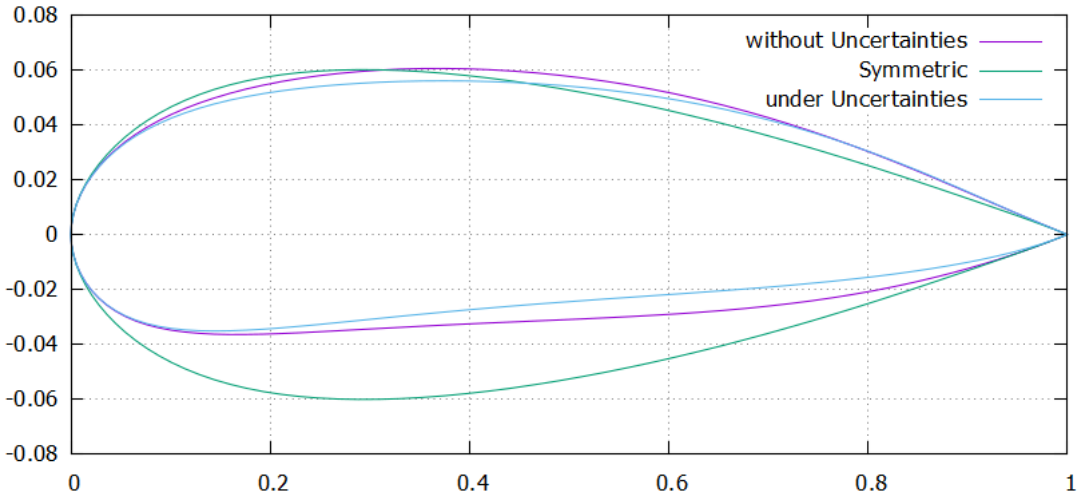


Figure 8.4: *Comparison of the airfoils produced by the optimization w/o uncertainties (purple), under uncertainties (blue) and the original airfoil (green)*

A visual comparison of the non – dimensional adjoint flow field between the case without uncertainties and the case of optimization under uncertainties is made in figure 8.5 . Notice that, consistently with the adjoint methodology, the direction of the adjoint flow runs opposite to the flow field, with the wake of the airfoil being is at its trailing edge.

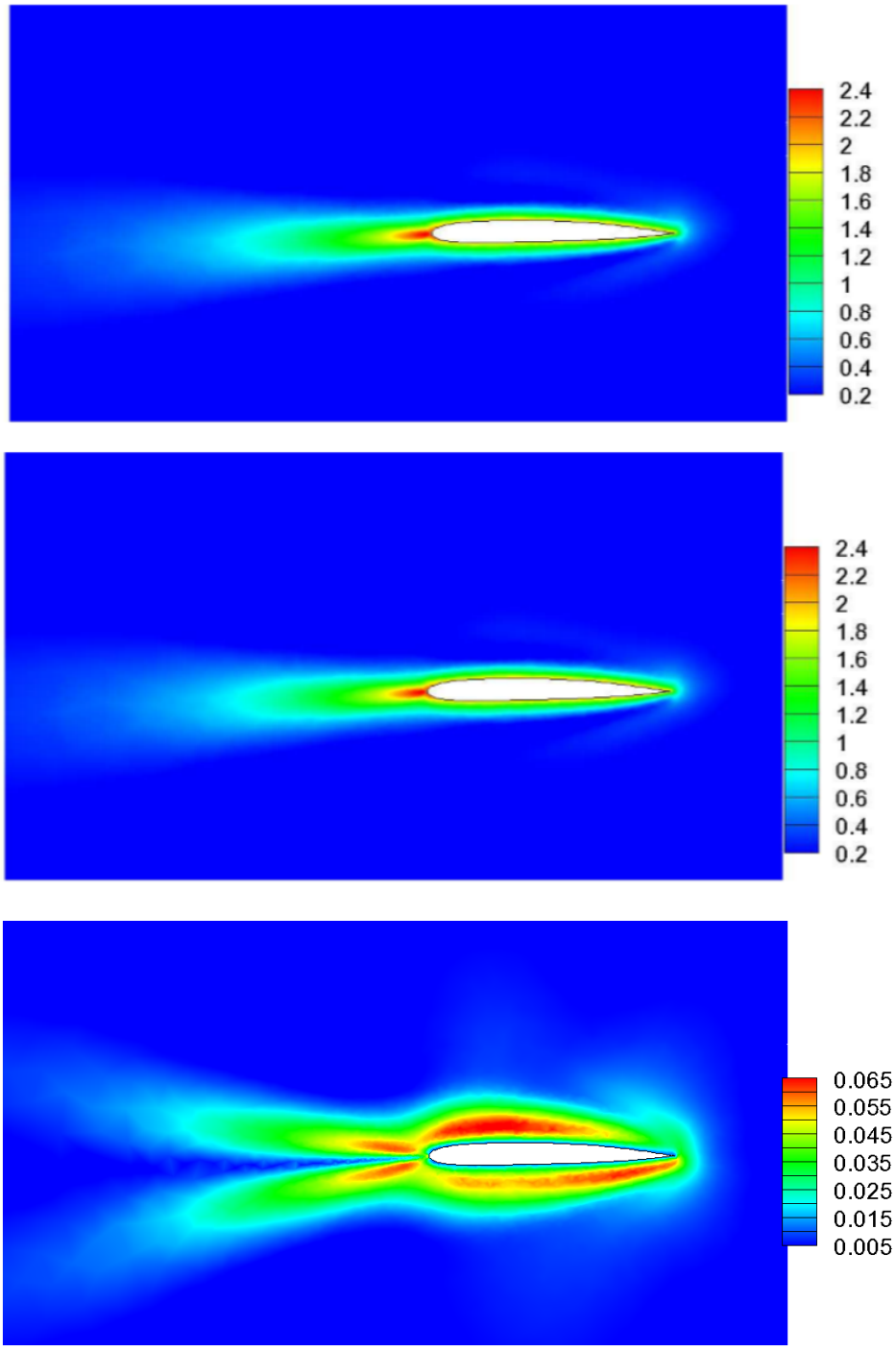


Figure 8.5: *Non–dimensional adjoint velocity field, comparison w/o uncertainties (top), and the mean value (middle) and standard deviation (bottom) of the adjoint velocity from the design under uncertainties for* $C = 3$

In figure 8.6 the same mean value fields of the non – dimensional adjoint velocity field, for $C = 1$ and $C = 2$ can be seen, for comparison purposes. The figures are presented in order to illustrate the similarity with the field resulted for $C = 3$, for which no visual difference is evedent.

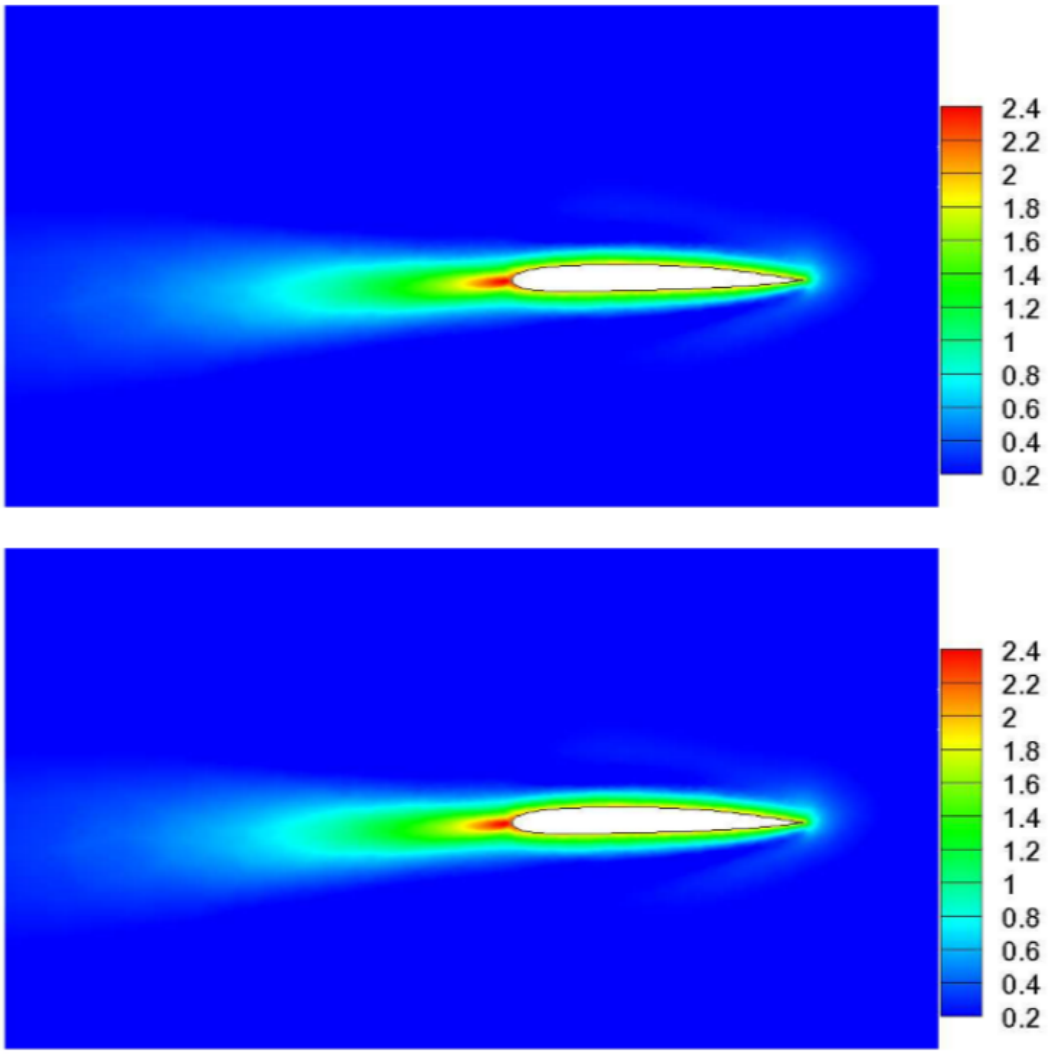


Figure 8.6: *Mean value of non–dimensional adjoint velocity field for $C = 1$ (top) and $C = 2$ (bottom)*

In figure 8.7 the mean value and the standard deviation of the Mach number around the optimized geometry for $C = 1$ is presented. Results for $C = 2$ are presented in figure 8.8. Notice that, in the Mach fields of the mean value and the standard deviation, no visual difference can be observed between $C = 1$ and $C = 2$.

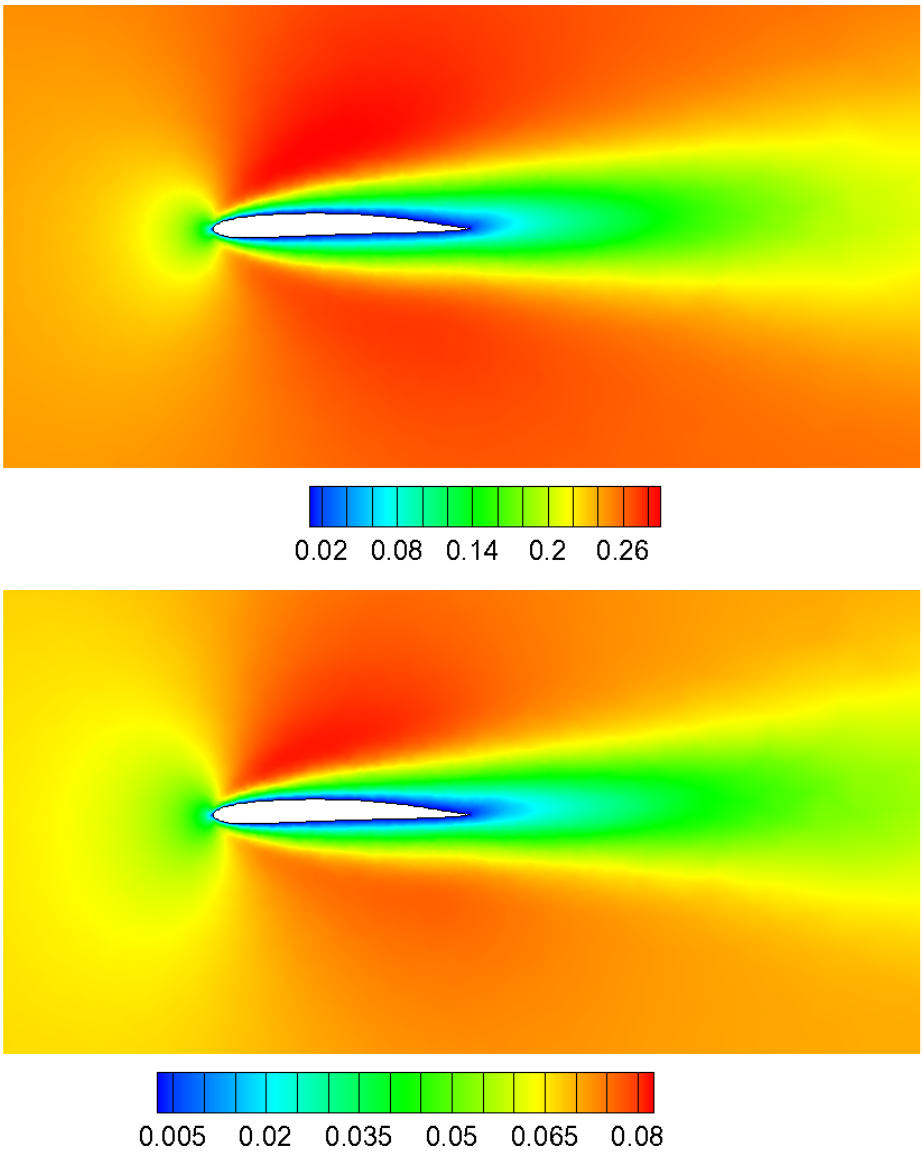


Figure 8.7: *Mean value (top) and standard deviation (bottom) of Mach number around the optimized airfoil geometry, for* $C = 1$.

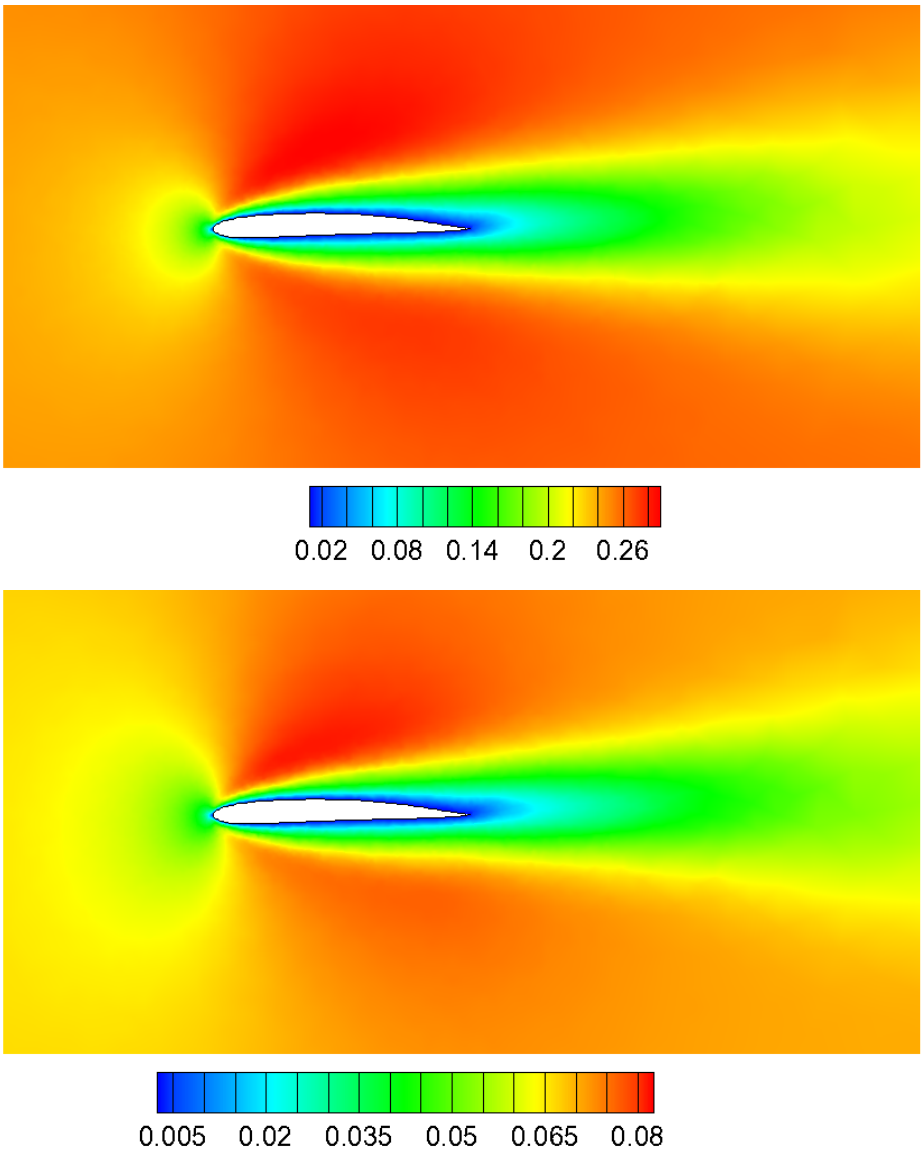


Figure 8.8: *Mean value (top) and standard deviation (bottom) of Mach number around the optimized airfoil geometry, for* $C = 2$.

Chapter 9

Overview, Conclusions and Future Research

9.1 Overview

In this diploma thesis, a method for applying the intrusive PCE method was introduced and mathematically formulated. The method is based on non–intrusive operations and as a result it is independent of the governing PDE used, while enjoying the benefits of the intrusive PCE in terms of computational cost. Tthe method is based on non–intrusive operations for producing the iPCE equations, therefore rendering its application painless, both in terms of the mathematical development and programming investment required. Practically, the method doesn't require significant changes in the software without uncertainties and, in terms of applicability, it is similarly simple to the niPCE. This was made possible by establishing a proper mathematical framework for the method, that generalized the notion of the Galerkin projection in order to prove the homogeneity property, produce the numerical solution scheme of the iPCE equations and to establish their stability properties. In terms of stability, the presented iPCE method is found to be stable, provided the stability of its counterpart without uncertainties, and, as a result, it is regarded superior to the niPCE.

In order to test the validity of this method, a RANS solver for unstructured grids was modified in order to employ this method. This software which is an in–house one,

developed at the Parallel CFD & Optimization Unit. For validation purposes, this is later applied to various 2D and 3D test cases, for viscous and turbulent flows and the results are compared with other UQ techniques, such as the non–intrusive PCE and the Monte–Carlo method. The results indicate that this method accurately predicts the statistical moments of commonly used QoIs, while it outperforms its non–intrusive counterpart in terms of computational cost, with the difference expected to rise as the number of uncertain variables increases. More specificaly, for a small number of uncertain variables, the iPCE is shown to be between 10% and 40% faster than the niPCE, for a number of uncertain variables $m \leq 8$. Since the two variants, with the painless method presented in this thesis, require a similarly minimal programming investment for their development, the iPCE method is regarded as superior to the niPCE in terms of computational cost.
Having obtained a valid method for UQ, this is applied for optimization purposes with the goal of conducting robust design with the use of the E–SI continuous adjoint formulation. Firstly, a software developed by the PCOpt/NTUA that employed the SI adoint is modified in order to handle the E–SI adjoint formulation. With this software as a basis, the iPCE method developed in this thesis is applied for the computation of the sensitivity derivatives of a QoI. This approach produced the set of the continuous adjoint iPCE equations for the 2D Euler equations, as well as explicit expressions for the sensitity derivatives that is consistent with the iPCE methodology. The E–SI software without uncerainties is, then, modified in order to employ this method. The resulting software was used in order to conduct robust design to a 2D airfoil.

9.2 Conclusions

Basic conclusions derived from this thesis are:

1. The iPCE, as formulated by this, painless approach, can be easily applied to any type of PDE.
2. The iPCE method can be applied in a painless manner. This is an important aspect of this method, since it solves its biggest drawback i.e. the painful and expensive, in terms of programming investment, implementation.
3. The method presented can be implemented with memory requirements that are very close to the requirements of the CFD solver without uncertainties, which is crucial since memory footprint of UQ methods can be prohibitive in large problems.

4. The solver produced through the painless iPCE method has good stability properties that are superior to the stability properties of the niPCE variant. The solver of the iPCE equations is stable, given the stability of the solver without uncertainties.

5. For a number of uncertain variables $m < 8$, the iPCE method was shown to be up to 40% faster than the niPCE. This difference is expected to rise as the number of uncertain variables increases.

6. This method can be extended to the Continuous Adjoint, requiring minimal mathematical and programming work

7. For the extension to the Continuous Adjoint, minimal interventions are required to the software without uncertainties.

9.3 Potential Future Research

There are various areas where further research could potentially provide fruitful results. The continuous adjoint method for the iPCE equations can be applied to a 3D problem. This is true for both the Euler equations, as well as the Navier–Stokes equations. Also, it is of high mathematical and technological interest to formulate the adjoint if there is uncertainty in the geometry of the aerodynamic object under investigation.

Appendix A

The 2D Euler Equations

The 2D Euler equations for a steady state flow can be written as

$$\begin{aligned} \frac{\partial \vec{f_i}}{\partial x_i} = A_i \, \frac{\partial \vec{U}}{\partial x_i} &= \mathbf{0} \ , \text{in } \Omega \\ u_i n_i &= 0 \ , \text{along } S_W \\ \vec{U} &= \vec{U}_\infty \ , \text{along } S_\infty \end{aligned} \tag{A.1}$$

with the Jacobians $A_i = \frac{\partial \vec{f_i}}{\partial \vec{U}}$ being

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2}[(\gamma-3)u_1^2 + (\gamma-1)u_2^2] & (3-\gamma)u_1 & (1-\gamma)u_2 & \gamma-1 \\ -u_1 u_2 & u_2 & u_1 & 0 \\ -\gamma u_1 \frac{E_t}{\rho} + (\gamma-1)u_1 \ \boldsymbol{u}^2 & \gamma\frac{E_t}{\rho} - \frac{\gamma-1}{2}(u_2^2+3u_1^2) & (1-\gamma)u_1 u_2 & \gamma u_1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -u_1 u_2 & u_2 & u_1 & 0 \\ \frac{1}{2}[(\gamma-3)u_2^2 + (\gamma-1)u_1^2] & (1-\gamma)u_1 & (3-\gamma)u_2 & \gamma-1 \\ -\gamma u_2 \frac{E_t}{\rho} + (\gamma-1)u_2 \ \boldsymbol{u}^2 & (1-\gamma)u_1 u_2 & \gamma\frac{E_t}{\rho} - \frac{\gamma-1}{2}(u_1^2+3u_2^2) & \gamma u_2 \end{bmatrix} \tag{A.2}$$

with S_W denoting the solid wall boundary, $\boldsymbol{n} = [n_1, n_2]^T$ the unit normal vector and $\vec{U}_\infty$ being the infinite flow conditions in S_∞. In the case of an airfoil with an

infinite flow angle a_∞, the objective function will be the lift of the airfoil, namely

$$F = L \equiv \int_{S_W} p(n_2 \cos a_\infty - n_1 \sin a_\infty) dS \tag{A.3}$$

These equations will constitute the basis for the development of the continuous adjoint, and will be referred to as the primal equations.

Appendix B

Orthonormal Polynomials through the Gram–Schmidt orthonormalization

In order to produce a series of orthonormal polynomials, a zero order polynomial p_0 is arbitrarily chosen. Then, each polynomial of the sequence can be obtained recursively, using the formula

$$p_k(\xi) = \xi^k - \sum_{j=1}^{k-1} \frac{< \xi^k, p_j >_w}{< p_j, p_j >_w} p_j(\xi) \tag{B.1}$$

It is easy to see that eq.B.1 defines a polynomial p_k that is orthogonal to all $p_j, \quad j = 0, \ldots, k-1$, since

$$< p_k, p_i >_w = < \xi^k, p_i >_w - \sum_{j=1}^{k-1} \frac{< \xi^k, p_j >_w}{< p_j, p_j >_w} \delta_{ij} = 0$$

.

Also, note that $degree[p_n] = n$ implies that the polynomials generated this way are linearly independant and hence form a basis of $\mathbb{R}$.

Appendix C

Eigen–Decomposition

The implementation of the FVS scheme requires the solution of the corresponding eigenproblem. Starting point is the characteristic equations $\Lambda n_i \frac{\partial \vec{W}}{\partial x_i} = \vec{0}$ where $\vec{W} = [w^A \; w^B \;\; w^C \; w^D \; w^E]^T$ are the characteristic variables, $\Lambda = diag(u^{(n)}, u^{(n)}, u^{(n)}, u^{(n)} + c, u^{(n)} - c)$, $u^{(n)} = u_i n_i$ and c is the speed of sound. By applying the iPCE followed by Galerkin projections, we get

$$G[\Lambda] n_i \frac{\partial G[\vec{W}]}{\partial x_i} = \vec{0} \tag{C.1}$$

Eqs. C.1 can also be written as

$$\begin{bmatrix} d_j G_j[Y_0 Y_0] & d_j G_j[Y_0 Y_1] & \dots & d_j G_j[Y_0 Y_q] \\ d_j G_j[Y_1 Y_0] & d_j G_j[Y_1 Y_1] & \dots & d_j G_j[Y_1 Y_q] \\ \vdots & \vdots & \vdots & \vdots \\ d_j G_j[Y_q Y_0] & d_j G_j[Y_q Y_1] & \dots & d_j G_j[Y_q Y_q] \end{bmatrix} \begin{bmatrix} \vec{W}_0 \\ \vec{W}_1 \\ \vdots \\ \vec{W}_q \end{bmatrix} = \vec{0} \tag{C.2}$$

where

$$d_j = diag(u_j^{(n)}, u_j^{(n)}, u_j^{(n)}, u_j^{(n)} + c_j, u_j^{(n)} - c_j)$$

$$\vec{W}_i = [w_i^A, w_i^B, w_i^C, w_i^D, w_i^E]^T$$

and $u_j^{(n)}$, w_j^A, c_j denote the j–th term of the PCE of $\vec{u} \cdot \vec{n}$, w^A and of c, respectively. Note that $G_k[Y_i Y_j]$ is equal to $< Y_i, Y_j, Y_k > = \int_{\mathcal{E}} Y_i Y_j Y_k w d\vec{\xi}$.
The spectral components of c_j can be found through the Galerkin projections of the

expression of c, written as a function of the conservative variables, as follows

$$c_j = G_j[c] = \int_{\mathcal{E}} \sqrt{\frac{\gamma R}{c_v \rho_i Y_i}\left(E_{t_k} Y_k - \frac{(\rho_i u_i Y_i)(\rho_k u_k Y_k)}{2\rho_l Y_l}\right)} w d\vec{\xi} \tag{C.3}$$

In eq. C.3, γ is the specific heat ratio, R the specific gas constant and c_v the specific heat capacity at constant volume. Note that a similar procedure can be used for the PCE of any quantity expressed in terms of the conservative variables. By re–ordering eqs. C.2, we get

$$diag[(Z(u^{(n)}), Z(u^{(n)}), Z(u^{(n)}), Z(u^{(n)} + c), Z(u^{(n)} - c)]\hat{w} = \vec{0} \tag{C.4}$$

where

$$\begin{aligned} \hat{w} &= [w_0^A \dots w_p^A w_0^B \dots w_p^B w_0^C \dots w_p^C w_0^D \dots w_p^D w_0^E \dots w_p^E]^T \\ Z(\lambda) &= \begin{bmatrix} \lambda_j G_j[Y_0 Y_0] & \lambda_j G_j[Y_0 Y_1] & \dots & \lambda_j G_j[Y_0 Y_q] \\ \lambda_j G_j[Y_1 Y_0] & \lambda_j G_j[Y_1 Y_1] & \dots & \lambda_j G_j[Y_1 Y_q] \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_j G_j[Y_q Y_0] & \lambda_j G_j[Y_q Y_1] & \dots & \lambda_j G_j[Y_q Y_q] \end{bmatrix} \end{aligned} \tag{C.5}$$

Thus, the solution of an eigenproblem corresponding to a $(5\times(q+1))\times(5\times(q+1))$ matrix is now reduced to one corresponding to the $(q+1)\times(q+1)$ matrix Z. The diagonalization of Z yields the desired eigenvalues and eigenvectors of $G[A_i]$.

References

[1] M.E. Chatzimanolakis, K.-D Kantarakias, V.G. Asouti, and K.C Giannakoglou. Setting up the intrusive polynomial chaos method for uncertainty quanti cation and adjoint-based optimization in compressible fluid flows. In *Tenth International Conference on Computational Fluid Dynamics (ICCFD10)*, Barcelona, Spain, July 9-13 2018.

[2] B.J. Debusschere, H.N. Najm, P.P. Pébray, O.M. Knio, R.G. Ghanem, and O.P. Le Maître. Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM Journal of Scientific Computing*, 26(2):698–719, 2004.

[3] MS Eldred. Recent advances in non-intrusive polynomial chaos and stochastic collocation methods for uncertainty analysis and design. In *50th AIAA/ASME/ ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2009.

[4] David A. Fournier, Hans J. Skaug, Johnoel Ancheta, James Ianelli, Arni Magnusson, Mark N. Maunder, Anders Nielsen, and John Sibert. Ad model builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. *Optimization Methods and Software*, 27(2):233–249, 2012.

[5] B. Ganapathysubramanian and N. Zabaras. Sparse grid collocation schemes for stochastic natural convection problems. *Journal of Computational Physics*, 225(1):652–685, 2007.

[6] M. B. Giles and N A. Pierce. Adjoint equations in cfd - duality, boundary conditions and solution behaviour. *AIAA Journal*, 97, 2000.

[7] V. Kaarnioja. Smolyak quadrature. Master's thesis. University of Helsinki, Department of Mathematics and Statistics, 2013.

[8] I.C. Kampolis, D.I. Papadimitriou, and K.C. Giannakoglou. Evolutionary optimization using a new radial basis function network and the adjoint formulation. *Inverse Problems in Science and Engineering*, 14(4):397–410, 2006.

[9] K.-D. Kantarakias, M.E. Chatzimanolakis, V.G. Asouti, and K.C. Giannakoglou. On the development of the 3D Euler equations using intrusive pce for uncertainty quantification. In *2nd ECCOMAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2017)*, Rhodes Island, Greece, June 15-17 2017.

[10] 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. *Journal of Computational Physics*, 301:1–18, 2015.

[11] O.M. Knio and O.P. Le Maître. Uncertainty propagation in CFD using polynomial chaos decomposition. *Fluid Dynamics Research*, 38:616–640, 2006.

[12] E.A. Kontoleontos, K.C. Giannakoglou, and D.G Koubogiannis. Robust design of compressor cascade airfoils using evolutionary algorithms and surrogate models. In *1st International Conference on Experiments/Process/System Modelling/Simulation/Optimization*, Athens, Greece, July 6-9 2005.

[13] B. Lapeyre. Introduction to monte–carlo methods. Halmstad University, Lecture Notes, 2007.

[14] O.P. Le Maître, O.M. Knio, H.N. Najm, and R.G. Ghanem. A stochastic projection method for fluid flow I. Basic formulation. *Journal of Computational Physics*, 173(2):481–511, 2001.

[15] B. van Leer. Flux vector splitting for the euler equations. In *8th International Conference on Numerical Methods in Fluid Dynamics*, Aachen, Germany, June 28 - July 2 1982.

[16] A.G. Liatsikouras, V.G. Asouti, K.C. Giannakoglou, G. Pierrot, and M. Megahed. Aerodynamic shape optimization under flow uncertainties using non-intrusive polynomial chaos and evolutionary algorithms. In *2nd ECCOMAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2017)*, Rhodes Island, Greece, June 15-17 2017.

[17] M.D. McKay. Latin hypercube sampling as a tool in uncertainty analysis of computer models. In *24th Conference on Winter Simulation, WSC '92*, New York, NY, USA, June 7-10 1992.

[18] W. Morokoff and R. Caflisch. Quasi-Monte Carlo integration. *Journal of Computational Physics*, 122(2):218–230, 1995.

[19] D.I. Papadimitriou and K.C. Giannakoglou. A continuous adjoint method with objective function derivatives based on boundary integrals for inviscid and viscous flows. *Computers & Fluids*, 36(2):325–341, 2007.

[20] E.M. Papoutsis-Kiachagias, Papadimitriou D.I, and K.C. Giannakoglou. Robust design in aerodynamics using 3rd-order sensitivity analysis based on discrete adjoint. application to quasi-1d flows. *International Journal for Numerical Methods in Fluids*, 69(3):691–709, 2012.

[21] M.P. Pettersson, G. Iaccarino, and J. Nordström. *Polynomial Chaos Methods for Hyperbolic Partial Differential Equations*. Springer International Publishing, Switzerland, 2015.

[22] S. Smolyak. Quadrature and interpolation formulas on tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR*, 148(5):1042–1045, 1963.

[23] P. Spalart and S. Allmaras. A one–equation turbulence model for aerodynamic flows. *La Recherche Aerospatiale*, 1:5–21, 1994.

[24] W. Squire and G. Trapp. Using complex variables to estimate derivatives of real functions. *SIAM Review*, 40(1):110–112, 1998.

[25] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60:897–936, 1938.

[26] D. Xiu. Fast numerical methods for stochastic computations: a review. *Communications in Computational Physics*, 5(2-4):242–272, 2009.

[27] D. Xiu. *Numerical Methods for Stochastic Computations: A Spectral Approach*. Princeton University Press, 2010.

[28] D. Xiu and G.E Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187:37–67, 2003.

Εθνικο Μετσοβειο Πολυτεχνειο
Σχολη Μηχανολογων Μηχανικων
Τομεας Ρευστων
Εργαστηριο Θερμικων Στροβιλομηχανων
Παραλληλης ΥΡΔ & Βελτιστοποιησης

Εφαρμογή του Επεμβατικού Αναπτύγματος Πολυωνυμικού Χάους στην Αεροδυναμική Βελτιστοποίηση Μορφής με τη Συνεχή Συζυγή Μέθοδο

Εκτενής Ελληνική Περίληψη Διπλωματικής Εργασίας

Κυριάκος Δημήτριος Κανταράκιας

Επιβλέπων Καθηγητής: Κ.Γ. Γιαννάκογλου

Αθήνα, Σεπτέμβριος 2018

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

1.1 Αεροδυναμική Βελτιστοποίηση

Στον σύγχρονο τομέα της Υπολογιστικής Ρευστοδυναμικής (ΥΡΔ), ένα μεγάλο τμήμα της έρευνας στρέφεται γύρω από το αντικείμενο της βελτιστοποίησης με σκοπό το σχεδιασμό αεροδυναμικών μορφών (λ.χ. αεροτομών, πτερυγίων στροβιλομηχανών) ώστε να αποδίδουν καλύτερα με βάση τις επιλεγείς συναρτήσεις–στόχους (λ.χ. συντελεστής άνωσης, συντελεστής αντίστασης κλπ).

Στην αεροδυναμική βελτιστοποίηση ο στόχος είναι να ανιχνευτεί, με οικονομικό τρόπο, ο χώρος τον υποψήφιων λύσεων – σχημάτων που μπορεί να λάβει το υπό εξέταση αεροδυναμικό σώμα, και ο εντοπισμός αυτής με τις καλύτερες ιδιότητες. Αυτή η απαίτηση υποδηλώνει την ανάγκη καθορισμού ενός ή περισσοτέρων συναρτήσεων–στόχων οι οποίες πρέπει να ελαχιστοποιηθούν, ικανοποιώντας ταυτόχρονα ένα σύνολο περιορισμών του προβλήματος. Κατά τη διαδικασία ανίχνευσης του χώρου των υποψήφιων λύσεων είναι απαραίτητη η υποστήριξη της διαδικασίας από ένα υπολογιστικό εργαλείο (λογισμικό αξιολόγησης), το οποίο λύνει τις εξισώσεις ροής και να αξιολογεί τις υποψήφιες λύσεις ως προς τους στόχους που τέθηκαν.

Στην αεροδυναμική, τίθεται συχνά το πρόβλημα της βελτιστοποίησης μορφής, που στοχεύει στον προσδιορισμό της γεωμετρίας ενός αεροδυναμικού σχήματος το οποίο θα έχει βέλτιστη συμπεριφορά ως προς κάποια συνήθη αεροδυναμική ιδιότητα (λ.χ. συντελεστής άνωσης, αντίστασης κλπ). Για παράδειγμα, στο πρόβλημα της βελτιστοποίησης μιας αεροτομής, είναι επιθυμητή η εύρεση της αεροτομής που, για δεδομένο συντελεστή άνωσης, έχει τον ελάχιστο δυνατό συντελεστή αντίστασης.

Αναπόσπαστο τμήμα της βελτιστοποίησης, είναι η διακριτοποίηση του χωρίου υπολογισμού μέσω κατάλληλου υπολογιστικού πλέγματος, και η επακόλουθη διακριτοποίηση των Μερικών Διαφορικών Εξισώσεων (ΜΔΕ) του προβλήματος και η αναγωγή τους σε ένα γραμμικοποιημένο πρόβλημα, το οποίο μπορεί να επιλυθεί με γνωστές μαθηματικές μεθόδους.

Όσον αφορά τον τρόπο βελτιστοποίησης, αυτή μπορεί να γίνει με χρήση στοχαστικών μεθόδων όπως οι εξελικτικοί αλγόριθμοι [1, 2, 3, 4, 5]. Τέτοιοι αλγόριθμοι αξιολογούν πληθυσμούς λύσεων, τους κατατάσσουν με βάση την τιμή της συνάρτησης–στόχος και

παράγουν νέους πληθυσμούς λύσεων ως γόνους των προηγούμενων πληθυσμών, με τις συνήθεις διαδικασίες διασταύρωσης και μετάλλαξης.

Εν αντιθέσει, ένα εναλλακτικό σύνολο μεθόδων στηρίζεται στον υπολογισμό της παραγώγου της αντικειμενικής συνάρτησης ως προς τις μεταβλητές σχεδιασμού. Τέτοιες μέθοδοι ονομάζονται αιτιοκρατικές και αυτή που χρησιμοποιείται σε αυτήν τη διπλωματική εργασία είναι η συζυγής (adjoint) μέθοδος [6, 7].

1.2 Αβεβαιότητα και Στιβαρός Σχεδιασμός

Οι σύγχρονοι κώδικες ΥΡΔ έχουν την ικανότητα να προλέγουν ροές με μεγάλη ακρίβεια, σε περιπτώσεις όπου η αβεβαιότητα που υπάρχει στον φυσικό κόσμο αμελείται. Ωστόσο, σε πολλές περιπτώσεις προβλημάτων, η επίδραση που έχει η στοχαστικότητα των αβέβαιων παραμέτρων ενός προβλήματος στην απόδοση ενός συστήματος είναι αρκετά μεγάλη ώστε να μην μπορεί να αμεληθεί. Για παράδειγμα, μια μικρή αλλαγή της γωνίας της ροής στην είσοδο ενός συμπιεστή μπορεί να προκαλέσει μεγάλη μεταβολή στην απόδοσή του.

Για αυτόν τον λόγο, τα τελευταία χρόνια, αρκετή έρευνα έχει στραφεί προς τη μελέτη και την ανάπτυξη μεθόδων για την ποσοτικοποίηση της επίδρασης της αβεβαιότητας του φυσικού κόσμο στα αεροδυναμικά συστήματα. Ο στόχος τέτοιων μεθόδων είναι, δεδομένης της στατιστικής συμπεριφοράς των αβέβαιων παραμέτρων των ροών, να ποσοτικοποιηθεί η επίδρασή τους σε κάποια ποσότητα ενδιαφέροντος, όπως λ.χ. ο συντελεστής άνωσης μιας αεροτομής. Στην εργασία αναπτύσσεται μια μέθοδος βασισμένη στο Polynomial Chaos Expansion (PCE), στην επεμβατική της (intrusive) εκδοχή. Η intrusive PCE (iPCE) μέθοδος για την Ποσοτικοποίηση Αβεβαιότητας (Uncertainty Quantification ή UQ) είναι αποδοτική, από πλευράς υπολογιστικού κόστους, σε σύγκριση με άλλες μεθόδους για UQ, αλλά είναι εξαιρετικά δύσκολη στην εφαρμογή της. Στην διπλωματική αυτή εργασία, εισάγεται μια νέα εκδοχή της μεθόδου iPCE, όπου εφαρμόζει την μέθοδο με έναν άκοπο (Painless) τρόπο.

Η ύπαρξη αυτών των αβεβαιοτήτων, υποδηλώνει την ανάγκη για την ανάπτυξη διαδικασιών Στιβαρού Σχεδιασμού. Τέτοιες διαδικασίες στοχεύουν στον σχεδιασμό συστημάτων με απόδοση η οποία δεν μεταβάλλεται σε σημαντικό βαθμό όταν οι συνθήκες περιβάλλοντος του συστήματος αποκλίνουν από το σημείο σχεδιασμού του, ακόμα και αν έχουν ελαφρώς χειρότερη απόδοση στο ονομαστικό σημείο σχεδιασμού. Ο σκοπός της διπλωματικής εργασίας είναι η ανάπτυξη μεθόδων για στιβαρό σχεδιασμό, ή για βελτιστοποίηση υπό αβεβαιότητες.

Κεφάλαιο 2

Ορθογώνια Πολυώνυμα και PCE

Στο κεφάλαιο αυτό συζητούνται οι ιδιότητες των ορθογωνίων πολυωνύμων, καθώς και η εφαρμογή τους στη μέθοδο του PCE.

2.1 Μονοδιάστατα Ορθογώνια Πολυώνυμα

Ένα σύνολο πολυωνύμων $\{p_n(\xi)\}_{n=0}^{\infty}$ καλείται ορθογώνιο ως προς μια συνάρτηση $w(\xi)$ που ορίζεται στο διάστημα (a, b) αν ισχύει η παρακάτω ιδιότητα,

$$\langle p_n, p_m \rangle \equiv \int_a^b p_n(\xi) p_m(\xi) w(\xi) d\xi = \delta_{mn} \langle p_n, p_n \rangle \tag{2.1}$$

στο διάστημα (a, b) όπου δ_{mn} είναι το δέλτα του Kronecker. Στην περίπτωση όπου $< p_n, p_n >= 1$ για κάθε N, τότε τα πολυώνυμα λέγονται ορθοκανονικά.

2.2 Πολυδιάστατα Ορθογώνια Πολυώνυμα

Ο ορισμός των πολυδιάστατων ορθογώνιων πολυωνύμων βασίζεται ένα σύνολο m μονοδιάστατων ορθογωνίων πολυωνύμων $p^k \equiv \{p_n^k(\xi_k)\}_{n=0}^{\infty}, k = 1, \ldots, m$, το καθένα από τα οποία είναι ορθογώνια ως προς μια συνάρτηση $w_k(\xi_k)$.

Συνεπώς, μπορεί να οριστεί η ακόλουθη σειρά πολυωνύμων m μεταβλητών ως το ακόλουθο τανυστικό γινόμενο

$$Y \equiv \{Y_n\}_{n=0}^{\infty} := \otimes_{k=1}^{m} p^k = \{p_{n_1}^1(\xi_1) p_{n_2}^2(\xi_2) \ldots p_{n_m}^m(\xi_m)\}_{n_1, n_2, \ldots, n_m = 0}^{\infty} \tag{2.2}$$

Με βάση αυτήν την ιδιότητα, αυτά τα πολυώνυμα είναι ορθογώνια, με το εσωτερικό

γινόμενο να ορίζεται ως

$$< f, g >_W = \int_{\mathcal{E}} f g W d\xi_1 \dots d\xi_m \quad , \quad W := \prod_{j=1}^{m} w_j(\xi_j) \tag{2.3}$$

2.3 Η μεθοδολογία του PCE

Σε αυτήν την ενότητα συζητείται η έννοια του PCE, που είναι βασικό προαπαιτούμενο για την ποσοτικοποίηση της αβεβαιότητας. Πιο συγκεκριμένα, για μία συνάρτηση $\phi = \phi(\boldsymbol{\xi})$, με το $\boldsymbol{\xi}$ να έχει διάσταση m, το PCE του $\phi(\boldsymbol{\xi})$ ορίζεται ως η σειρά

$$\phi(\xi) = \sum_{j=0}^{\infty} \phi^j Y_j(\boldsymbol{\xi}) \tag{2.4}$$

Τα πολυώνυμα Y_j, όπως αναφέρθηκε και στην προηγούμενη ενότητα, είναι ορθογώνια ως προς τη συνάρτηση $W(\boldsymbol{\xi}) := \prod_{j=1}^{m} w_j(\xi_j)$. Στη μέθοδο του PCE σκοπός είναι ο υπολογισμός των συνελεστών της ανωτέρω σειράς, οι οποίοι υπολογίζονται από τη σχέση

$$\phi^j = \langle \phi(\xi), Y_j \rangle \tag{2.5}$$

Αν η συνάρτηση $W(\boldsymbol{\xi}) := \prod_{j=1}^{m} w_j(\xi_j)$ αποτελείται από συναρτήσεις πυκνότητας πιθανότητας μιας μεταβλητής, τότε οι συντελεστές του PCE του $\phi(\boldsymbol{\xi})$ ικανοποιούν τις σχέσεις

$$\begin{aligned} E[\phi] &\equiv \mu_\phi = \phi^0 \\ Var[\phi] \equiv \sigma_\phi^2 &= \sum_{j=1}^{\infty} (< Y_j, Y_j > \phi^j)^2 \end{aligned} \tag{2.6}$$

Δηλαδή η μέση τιμή και η τυπική απόκλιση της συνάρτησης $\phi(\boldsymbol{\xi})$ δίνεται από αλγεβρικές συναρτήσεις των συντελεστών του PCE της ίδιας της συνάρτησης. Προφανώς το PCE πρέπει να αποκοπεί για αριθμητικούς σκοπούς σε έναν πεπερασμένο αριθμό όρων.

2.4 Non–Intrusive PCE

Στο non–intrusive PCE (niPCE), η ποσότητα ενδιαφέροντος της οποίας η στατιστική συμπεριφορά πρέπει να μελετηθεί γράφεται ως

$$F = \sum_{j=0}^{\infty} F^j Y_j(\boldsymbol{\xi}) \tag{2.7}$$

όπου με βάση τα όσα αναφέρθηκαν στην προηγούμενη ενότητα, η μέση τιμή και η τυπική απόκλιση αυτής της ποσότητας δίνονται αντίστοιχα από τις σχέσεις

$$E[F] = F^0$$
$$Var[F] = \sum_{j=1}^{\infty} (< Y_j, Y_j > F^j)^2 \tag{2.8}$$

Το παραπάνω ανάπτυγμα πρέπει, για πρακτικούς λόγους υπολογισμού, να αποκοπεί σε πεπερασμένο αριθμό $q+1$ όρων. Αυτός ο αριθμός εξαρτάται από την επιλογή μιας τάξης χάους C, η οποία καθορίζεται από τον χρήστη ανάλογα με την επιθυμητή ακρίβεια της μεθόδου και τη διαθέσιμη υπολογιστική ισχύ και δίνεται από

$$q + 1 = \frac{(C + m)!}{C!m!} \tag{2.9}$$

Με βάση αυτά, ισχύει ότι

$$F = \sum_{j=0}^{q} F^j Y_j(\boldsymbol{\xi}) \tag{2.10}$$

Άρα, οι συντελεστές προς υπολογισμό δίνονται από τη σχέση

$$F^j \equiv \langle F, Y_j \rangle \equiv \int_{\mathcal{E}} F Y_j W d\boldsymbol{\xi} \quad , j = 0, \ldots, q \tag{2.11}$$

αυτό επιτυγχάνεται με αριθμητική ολοκλήρωση, όπως λ.χ. η ολοκλήρωση κατά Gauss. Αυτή η μέθοδος βασίζεται στην κλήση της F στους κόμβους της αριθμητικής ολοκλήρωσης, το οποίο γίνεται με επιμέρους τρεξίματα του λογισμικού αξιολόγησης δίχως παρεμβολή σε αυτό.

2.5 Intrusive PCE

Στο iPCE (iPCE) γράφουμε

$$\boldsymbol{U} = \sum_{j=0}^{q} \boldsymbol{U}^j Y(\boldsymbol{\xi}) \tag{2.12}$$

Στην περίπτωση αυτή, οι φασματικοί συντελεστες $\boldsymbol{U}^j \ , \ j = 0, \ldots, q$ είναι οι άγνωστοι ως προς τους οποίους θα λυθεί το πρόβλημα. Με βάση αυτήν τη λογική, η έκφραση του $\boldsymbol{U}$ μπαίνει κατευθείαν στην εξίσωση $\boldsymbol{R}(\boldsymbol{U}) = \boldsymbol{0}$, όπου είναι η εξίσωση του προβλήματος όταν δεν μοντελοποιούνται οι αβεβαιότητες,

$$\boldsymbol{R}\left(\sum_{j=0}^{q} \boldsymbol{U}^j Y(\boldsymbol{\xi})\right) = \boldsymbol{0} \tag{2.13}$$

Με χρήση των προβολών κατά Galerkin, προκύπτουν οι επεμβατικές εξισώσεις, οι οποίες γράφονται απλά ως

$$\int_{\mathcal{E}} \boldsymbol{R}\left(\sum_{j=0}^{q} \boldsymbol{U}^{j} Y(\boldsymbol{\xi})\right) Y_k W d\boldsymbol{\xi} = \mathbf{0}\ , k = 0, \ldots, q \tag{2.14}$$

Με αριθμητική επίλυση των εξισώσεων αυτών, βρίσκονται οι φασματικοί συντελεστές του PCE των ροικών μεταβλητών $\boldsymbol{U}$. Για να γίνει αυτό, απαιτούνται εκτενείς αλλαγές στο λογισμικό αξιολόγησης.

Κεφάλαιο 3

Νέα Προσέγγιση στο iPCE

Σε αυτό το κεφάλαιο συζητείται μια νέα προσέγγιση στην εφαρμογή του PCE. Η μέθοδος που παρατίθεται αναπτύχθηκε για τους σκοπούς της διπλωματικής εργασίας και έχει το πλεονέκτημα ότι εφαρμόζει το iPCE με τη χρήση non–intrusive πράξεων. Αυτό απλοποιεί σημαντικά την εφαρμογή της μεθόδου, καθιστώντας την άκοπη στην εφαρμογή της.

3.1 Βασικό Μαθηματικό Υπόβαθρο

Στα παρακάτω, γίνεται η θεώρηση πως η αβεβαιότητα εισάγεται μέσω ενός διανύσματος αβέβαιων μεταβλητών $\boldsymbol{\xi} \in \mathbb{R}^m$, κάθε μια από τις οποίες ακολουθεί μια συκεκριμένη συνάρτηση πυκνότητας πιθανότητας $w_k(\xi_k)$. Τα πολυώνυμα $Y = \{Y_n\}_{n=0}^{\infty}$ που θα αναφέρονται είναι ορθογώνια ως προς την συνάρτηση $W = \prod_{j=1}^{m} w_j$, με βάση τα όσα αναφέρθηκαν στην προηγούμενη ενότητα για τα ορθογώνια πολυώνυμα.

Με βάση αυτά, μπορεί να οριστεί η προβολή Galerkin ενός βαθμωτού μεγέθους, ως ακόλουθη ποσότητα

$$\phi^j := \int_{\mathcal{E}} \phi Y_j W d\boldsymbol{\xi} \tag{3.1}$$

Αυτός ο ορισμός μπορεί να γενικευθεί από τα βαθμωτά μεγέθη σε διανύσματα και σε πίνακες, με σκοπό τη χρήση τους σε υπολογιστικές μεθόδους. Πιο συγκεκριμένα, για ένα διάνυσμα $\boldsymbol{U}(\boldsymbol{\xi}) = [U_1(\boldsymbol{\xi}), ..., U_n(\boldsymbol{\xi})]^T$, η προβολή Galerkin ορίζεται ως

$$\mathrm{G}^{\mathrm{q}}[\boldsymbol{U}] := [\boldsymbol{U}^0, \boldsymbol{U}^1, ..., \boldsymbol{U}^q]^T \tag{3.2}$$

όπου το $\boldsymbol{U}^k = [U_1^k, U_2^k, ..., U_n^k]^T,\ k = 0, \ldots, q.$

Αξίζει να σημειωθεί πως στον παραπάνω ορισμό, αν $\phi = \phi(\boldsymbol{\xi})$, τότε $\mathrm{G}^{\mathrm{q}}[\phi] = [\phi^0, \ldots, \phi^q]^T$. Αυτό σημαίνει πως η περίπτωση μιας βαθμωτής ποσότητας είναι υποπερίπτωση του ορισμού για τα διανύσματα. Η ίδια διαδικασία μπορεί να ακολουθηθεί και στην περίπτωση

ενός πίνακα $A \in \mathbb{R}^{n\times n}$ του οποίου τα στοιχεία είναι A_{ij}=$A_{ij}(\boldsymbol{\xi})$. Τότε, ομοίως με πριν, η προβολή Galerkin ορίζεται ως το παρακάτω μητρώο.

$$\mathrm{G}^{\mathrm{q}}[A] = \begin{bmatrix} A^{00} & A^{01} & \dots & A^{0q} \\ A^{10} & A^{11} & \dots & A^{1q} \\ \vdots & \vdots & \vdots & \vdots \\ A^{q0} & A^{q1} & \dots & A^{qq} \end{bmatrix} \tag{3.3}$$

όπου το (i,j) στοιχείο του $A^{\lambda\mu} \in \mathbb{R}^{n\times n}$ δίνεται από τη σχέση

$$A_{ij}^{\lambda\mu} := \int_{\mathcal{E}} A_{ij} Y_\lambda Y_\mu W d\boldsymbol{\xi} = \sum_{\rho=0}^{\infty} A_{ij}^{\rho} < Y_\rho, Y_\lambda, Y_\mu > \tag{3.4}$$

όπου $< Y_\rho, Y_\lambda, Y_\mu >:= \int_{\mathcal{E}} Y_\rho Y_\lambda Y_\mu W d\boldsymbol{\xi}$.

Στην περίπτωση όπου όλες οι αβέβαιες ποσότητες έχουν αποκοπεί σε $q+1$ όρους, τότε μπορεί να αποδειχτεί ότι για ένα πίνακα A και ένα διάνυσμα $\boldsymbol{U}$ ισχύει η ομοιογένεια πρώτου βαθμού για τον τελεστή Galerkin, όπως έχει ήδη οριστεί. Δηλαδή

$$A_{ij}=\sum_{k=0}^{q} A_{ij}^{k} Y_k(\boldsymbol{\xi}) \quad \text{ανδ} \quad U_j=\sum_{k=0}^{q} U_j^k Y_k(\boldsymbol{\xi}) \quad \text{ωιτη} \quad i,j=1,\dots,n$$

τότε αποδεικνύεται ότι

$$\mathrm{G}^{\mathrm{q}}[A\boldsymbol{U}] = \mathrm{G}^{\mathrm{q}}[A]\,\mathrm{G}^{\mathrm{q}}[\boldsymbol{U}] \tag{3.5}$$

Απόδειξη. Έστω $\boldsymbol{f}=A\boldsymbol{U}$ ή $f_i=A_{ij}U_j$. Τότε, για κάθε $0 \le p \le q$

$$f_i^p = (A_{ij}U_j)^p \equiv \int_{\mathcal{E}} A_{ij}U_j Y_p W d\boldsymbol{\xi} = U_j^\rho \int_{\mathcal{E}} A_{ij} Y_\rho Y_p W d\boldsymbol{\xi} = U_j^\rho A_{ij}^{\rho p}$$

που είναι το p στοιχείο του $\mathrm{G}^{\mathrm{q}}[A]\,\mathrm{G}^{\mathrm{q}}[\boldsymbol{U}]$. □

Η ιδιότητα της ομοιογένειας είναι πολύ σημαντική για την εφαρμογή της μεθόδου του PCE σε αριθμητικά σχήματα επίλυσης Μερικών Διαφορικών Εξισώσεων. Αυτή η χρησιμότητα φαίνεται στην επόμενη ενότητα.

3.2 Αριθμητική Επίλυση των Επεμβατικών Εξισώσεων

Για ένα πρόβλημα το οποίο περιγράφεται, στη διακριτή του μορφή από ΜΔΕ, οι οποίες γράφονται συμβολικά ως

$$\boldsymbol{R}(\boldsymbol{U}) = \boldsymbol{0} \tag{3.6}$$

Πρέπει να χρησιμοποιηθεί ένα σχήμα γραμμικοποίησης για τη λύση τους. Το σύστημα 3.6 επιλύεται με το παρακάτω επαναληπτικό σχήμα

$$\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right)_{old} \Delta \boldsymbol{U} = -\left(\boldsymbol{R}\right)_{old} \quad , \quad \Delta \boldsymbol{U} = \boldsymbol{U}_{new} - \boldsymbol{U}_{old} \tag{3.7}$$

Αυτό το γραμμικό σύστημα λύνεται με αγνώστους τα $\Delta \boldsymbol{U}$ και ακολουθείται από την ακόλουθη ενημέρωση

$$\boldsymbol{U}_{new} = \boldsymbol{U}_{old} + \Delta \boldsymbol{U} \tag{3.8}$$

των τιμών του $\boldsymbol{U}$ σε κάθε κόμβο του πλέγματος. Σε κάθε λύση, το σύστημα υπολογίζεται ξανά και επιλύεται, έως ότου το $\boldsymbol{R}$ να λάβει επαρκώς μικρές τιμές.

Σε αυτήν τη διαδικασία, μπορεί να εφαρμοστεί κατευθείαν ο τελεστής Γκαλέρκιν, με τον ακόλουθο τρόπο.

$$\mathrm{G}^{\mathrm{q}}\left[\boldsymbol{R}\right] = \boldsymbol{0} \tag{3.9}$$

Η εφαρμογή του τελεστή στο αριθμητικό σχήμα επίλυσης δίνει ακολούθως

$$\mathrm{G}^{\mathrm{q}}\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right] \mathrm{G}^{\mathrm{q}}\left[\Delta \boldsymbol{U}\right] = -\,\mathrm{G}^{\mathrm{q}}\left[\boldsymbol{R}\right] \tag{3.10}$$

Αυτή η ιδιότητα ισχύει επειδή, όπως αποδείχτηκε στην ιδιότητα της ομοιογένειας, μπορούμε να γράψουμε ότι

$$G_i[\,\frac{\partial \boldsymbol{R}\left(\sum_{j=0}^{q} \boldsymbol{U}^j Y(\boldsymbol{\xi})\right)}{\partial \boldsymbol{U}} \Delta \boldsymbol{U}] = G_i[\,\frac{\partial \boldsymbol{R}\left(\sum_{j=0}^{q} \boldsymbol{U}^j Y(\boldsymbol{\xi})\right)}{\partial \boldsymbol{U}}] G_i[\Delta \boldsymbol{U}] \tag{3.11}$$

3.3 Εξοικονόμηση Μνήμης και Υπολογιστικού Κόστους

Ένας εναλλακτικός τρόπος επίλυσης των επεμβατικών εξισώσεων, με σκοπό την εξοικονόμηση μνήμης και υπολογιστικού κόστους, είναι να γίνει η προσέγγιση

$$\boldsymbol{U}(\boldsymbol{\xi}_z) - \boldsymbol{U}^0 = \sum_{i=1}^{q_1} \boldsymbol{U}^i Y_i(\boldsymbol{\xi}_z) + \sum_{i=q_1+1}^{\infty} \boldsymbol{U}^i Y_i(\boldsymbol{\xi}_z) = \sum_{i=q_1+1}^{\infty} \boldsymbol{U}^i Y_i(\boldsymbol{\xi} = \boldsymbol{\xi}_z) \tag{3.12}$$

Δηλαδή το πεδίο των φασματικών συντελεστών $\boldsymbol{U}_0$ να προσεγγιστεί από μια λύση του συστήματος δίχως αβεβαιότητες. Αυτό είναι χρήσιμο, αν η εξ. 3.10 ξαναγραφτεί στη

γενική της μορφή ως

$$\begin{bmatrix} \frac{\partial \boldsymbol{R}^{00}}{\partial \boldsymbol{U}} & \frac{\partial \boldsymbol{R}^{01}}{\partial \boldsymbol{U}} & \cdots & \frac{\partial \boldsymbol{R}^{0q}}{\partial \boldsymbol{U}} \\ \frac{\partial \boldsymbol{R}^{10}}{\partial \boldsymbol{U}} & \frac{\partial \boldsymbol{R}^{11}}{\partial \boldsymbol{U}} & \cdots & \frac{\partial \boldsymbol{R}^{1q}}{\partial \boldsymbol{U}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \boldsymbol{R}^{q0}}{\partial \boldsymbol{U}} & \frac{\partial \boldsymbol{R}^{q1}}{\partial \boldsymbol{U}} & \cdots & \frac{\partial \boldsymbol{R}^{qq}}{\partial \boldsymbol{U}} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{U}^0 \\ \Delta \boldsymbol{U}^1 \\ \vdots \\ \Delta \boldsymbol{U}^q \end{bmatrix} = - \begin{bmatrix} \boldsymbol{R}^0 \\ \boldsymbol{R}^1 \\ \vdots \\ \boldsymbol{R}^q \end{bmatrix} \tag{3.13}$$

Όμως, για $C = 1$ ισχύει ότι

$$\frac{\partial \boldsymbol{R}^{\lambda\mu}}{\partial \boldsymbol{U}}_{ij} = \sum_{\rho=0}^{q_1} \frac{\partial \boldsymbol{R}^{\rho}}{\partial \boldsymbol{U}}_{ij} < Y_\rho, Y_\lambda, Y_\mu >= \delta_{\lambda\mu} \mathcal{J}_{ij}^{00} \tag{3.14}$$

Άρα το σύστημα 3.13 γράφεται ως

$$\begin{bmatrix} \frac{\partial \boldsymbol{R}^{00}}{\partial \boldsymbol{U}}_{ij} & \frac{\partial \boldsymbol{R}^{01}}{\partial \boldsymbol{U}} & \frac{\partial \boldsymbol{R}^{02}}{\partial \boldsymbol{U}} & \cdots & \frac{\partial \boldsymbol{R}^{0q_1}}{\partial \boldsymbol{U}} \\ \frac{\partial \boldsymbol{R}^{10}}{\partial \boldsymbol{U}} & \frac{\partial \boldsymbol{R}^{00}}{\partial \boldsymbol{U}}_{ij} & \mathbf{0} & \cdots & \mathbf{0} \\ \frac{\partial \boldsymbol{R}^{20}}{\partial \boldsymbol{U}} & \mathbf{0} & \frac{\partial \boldsymbol{R}^{00}}{\partial \boldsymbol{U}}_{ij} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \boldsymbol{R}^{q_1 0}}{\partial \boldsymbol{U}} & \mathbf{0} & \mathbf{0} & \cdots & \frac{\partial \boldsymbol{R}^{00}}{\partial \boldsymbol{U}}_{ij} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{U}^0 \\ \Delta \boldsymbol{U}^1 \\ \Delta \boldsymbol{U}^2 \\ \vdots \\ \Delta \boldsymbol{U}^{q_1} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{R}^0 \\ \boldsymbol{R}^1 \\ \boldsymbol{R}^2 \\ \vdots \\ \boldsymbol{R}^{q_1} \end{bmatrix} \tag{3.15}$$

Λόγω της προσέγγισης στην εξίσωση 3.12, τα μη–διαγώνια στοιχεία του συστήματος 3.15 μπορούν να αμεληθούν, και να αποθηκευτούν μόνο τα διαγώνια στοιχεία του συστήματος.

3.4 Ευστάθεια Επεμβατικών Εξισώσεων

Εϊναι ενδιαφέρον να αποδειχτεί ότι αν η εξίσωση 3.6 γραμμικοποιηθεί με ένα σχήμα της μορφής 3.7, και το σχήμα επίλυσης που προκύψει είναι ευσταθές, τότε και το αντίστοιχο αριθμητικό σχήμα της εξίσωσης 3.10 είναι ευσταθές. Για να γίνει αυτό, θεωρούμε ότι τα ολοκληρώματα του τελεστή Galerkin υπολογίζονται αριθμητικά μέσω Gauss Quadrature (GQ), με βάρη $\mathcal{B}$ στους κόμβους $\mathcal{Q}$, για αβέβαιες μεταβλητές $\boldsymbol{\xi}$.

$$\mathcal{B} := \{b_1, \ldots, b_c\} \ , \ \mathcal{Q} := \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_c\} \tag{3.16}$$

όπου $c = (C+1)^m$. Επίσης θεωρείται πως η τάξη χάους αρκεί ώστε η το $\boldsymbol{U}(\boldsymbol{\xi})$ να έχει προσεγγιστεί επαρκώς από το αποκομμένο ανάπτυγμα του $\boldsymbol{U}$ με q όρους, δηλαδή

$$\boldsymbol{U} = \sum_{i=0}^{q} \boldsymbol{U}^i p_i(\boldsymbol{\xi}) \tag{3.17}$$

Σε όσα ακολουθούν $\mathrm{G^q}\left[\boldsymbol{U}^{(\kappa)}\right] \equiv [(\boldsymbol{U}^0)^{(\kappa)}, \ldots, (\boldsymbol{U}^q)^{(\kappa)}]^T$θα συμβολίζει τους φασματικούς συντελεστές του PCE στην κ–ιοστή επανάληψη του επιλύτη της εξίσωσης 3.10.

Επίσης υποθέτουμε ότι ο επιλύτης της εξίσωσης 3.6 χρησιμοποιείται c φορές, μια για κάθε $\boldsymbol{\xi} \in \mathcal{Q}$ ώστε να αξιολογήσει το $\boldsymbol{R}(\boldsymbol{U}(\boldsymbol{\xi}))$ σε κάθε κόμβο. Οι φασματικοί συντελεστές του $\boldsymbol{U}$ υπολογίζονται ως εξής

$$\begin{aligned}(\boldsymbol{U}'^g)^{(\kappa)} &= \sum_{i=1}^{c} b_i p_g(\boldsymbol{\xi}_i)\boldsymbol{U}'^{(\kappa)}(\boldsymbol{\xi}_i)\\ (\boldsymbol{R}'^g)^{(\kappa)} &= \sum_{i=1}^{c} b_i p_g(\boldsymbol{\xi}_i)\boldsymbol{R}'^{(\kappa)}(\boldsymbol{\xi}_i)\end{aligned} \tag{3.18}$$

Αρκεί να δείξουμε ότι

$$(\boldsymbol{U}^g)^{(\kappa)} = (\boldsymbol{U}'^g)^{(\kappa)} \;\text{ υπονοεί }\; (\boldsymbol{U}^g)^{(\kappa+1)} = (\boldsymbol{U}'^g)^{(\kappa+1)},\; g = 0,\ldots,q \tag{3.19}$$

Η λύση της εξίσωσης 3.7 για κάθε $\boldsymbol{\xi} \in \mathcal{Q}$ δίνει

$$\boldsymbol{U}'^{(\kappa+1)}(\boldsymbol{\xi}_i) = \boldsymbol{U}'^{(\kappa)}(\boldsymbol{\xi}_i) - \mathcal{J}^{-1}\boldsymbol{R}'^{(\kappa)}(\boldsymbol{\xi}_i) \tag{3.20}$$

όπου $\mathcal{J} := \frac{\partial \boldsymbol{R}^{(\kappa)}}{\partial \boldsymbol{U}^{(\kappa)}}$ και $\boldsymbol{R}'^{(\kappa)}(\boldsymbol{\xi}_i)$ συμβολίζουν το υπολογιζόμενο υπόλοιπο της εξίσωσης σε κάθε κόμβο Gauss. Υποθέτοντας ότι $\boldsymbol{U}_g^{(\kappa)} = \boldsymbol{U}_g'^{(\kappa)}$, παίρνουμε ότι

$$(R^g)^{(\kappa)} = (R'^g)^{(\kappa)},\; g = 0,\ldots,q \tag{3.21}$$

και, συνεπώς,

$$(\boldsymbol{U}'^g)^{(\kappa+1)} = (\boldsymbol{U}^g)^{(\kappa)} - \sum_{i=1}^{c} b_i Y_g(\boldsymbol{\xi}_i)\mathcal{J}^{-1}(\boldsymbol{\xi}_i)\boldsymbol{R}^{(\kappa)}(\boldsymbol{\xi}_i) \tag{3.22}$$

Επίσης

$$\mathrm{G^q}\left[\boldsymbol{U}^{(\kappa+1)}\right] = \mathrm{G^q}\left[\boldsymbol{U}^{(\kappa)}\right] - \mathrm{G^q}\left[\mathcal{J}^{-1}\right]\mathrm{G^q}\left[\boldsymbol{R}^{(\kappa)}\right] \tag{3.23}$$

Για την ευστάθεια του σχήματος, αρκεί να δείξουμε ότι

$$\sum_{i=1}^{c} b_i Y_g(\boldsymbol{\xi}_i)\mathcal{J}^{-1}(\boldsymbol{\xi}_i)\boldsymbol{R}^{(\kappa)}(\boldsymbol{\xi}_i) = \mathrm{G^q}\left[\mathcal{J}^{-1}\right]\mathrm{G^q}\left[\boldsymbol{R}^{(\kappa)}\right]$$

και συνεπώς

$$\begin{aligned}&G^q[\mathcal{J}^{-1}]G^q[\boldsymbol{R}^{(\kappa)}]|_g = \sum_{k=0}^{q}(J^{-1})^{gk}(\boldsymbol{R}^k)^{(\kappa)} =\\ &\sum_{i=1}^{c} b_i\mathcal{J}^{-1}(\boldsymbol{\xi}_i)Y_g(\boldsymbol{\xi}_i)\sum_{k=0}^{q}Y_k(\boldsymbol{\xi}_i)\sum_{j=1}^{c} b_j\boldsymbol{R}^{(\kappa)}(\boldsymbol{\xi}_j)Y_k(\boldsymbol{\xi}_j) =\\ &\sum_{i=1}^{c} b_i\mathcal{J}^{-1}(\boldsymbol{\xi}_i)Y_g(\boldsymbol{\xi}_i)\sum_{k=0}^{q}Y_k(\boldsymbol{\xi}_i)\boldsymbol{R}_k^{(\kappa)} = \sum_{i=1}^{c} b_i\mathcal{J}^{-1}(\boldsymbol{\xi}_i)Y_g(\boldsymbol{\xi}_i)\boldsymbol{R}^{(\kappa)}(\boldsymbol{\xi}_i)\end{aligned}$$

Κεφάλαιο 4

Εφαρμογή της Μεθόδου iPCE

Σε αυτό το κεφάλαιο, η μέθοδος iPCE εφαρμόζεται στην ατριβή τριδιάστατη ροή γύρω από μια πτέρυγα συμπιεστή τύπου NACA0010. Η ροή είναι διηχητική. Στην περίπτωση δίχως αβεβαιότητες, η γωνία πρόσπτωσης είναι $a_1 = 58^o$ με ισεντροπικό αριθμό Mach στην έξοδο να είναι $M_{2,is} = 0.4425$. Η ποσότητα ενδιαφέροντος είναι η αύξηση της στατικής πίεσης (p_2/p_1) μεταξύ της εισόδου (1) και της εξόδου (2). Για την περίπτωση του ονομαστικού σημείου λειτουργίας, $p_2/p_1 = 1.2967$. Οι αβεβαιότητες εισάγονται μέσω των οριακών συνθηκών, όπου $M_{2,is} \sim N(0.4425, 0.005)$ και $a_1 \sim \mathcal{U}(57^o, 59^o)$.

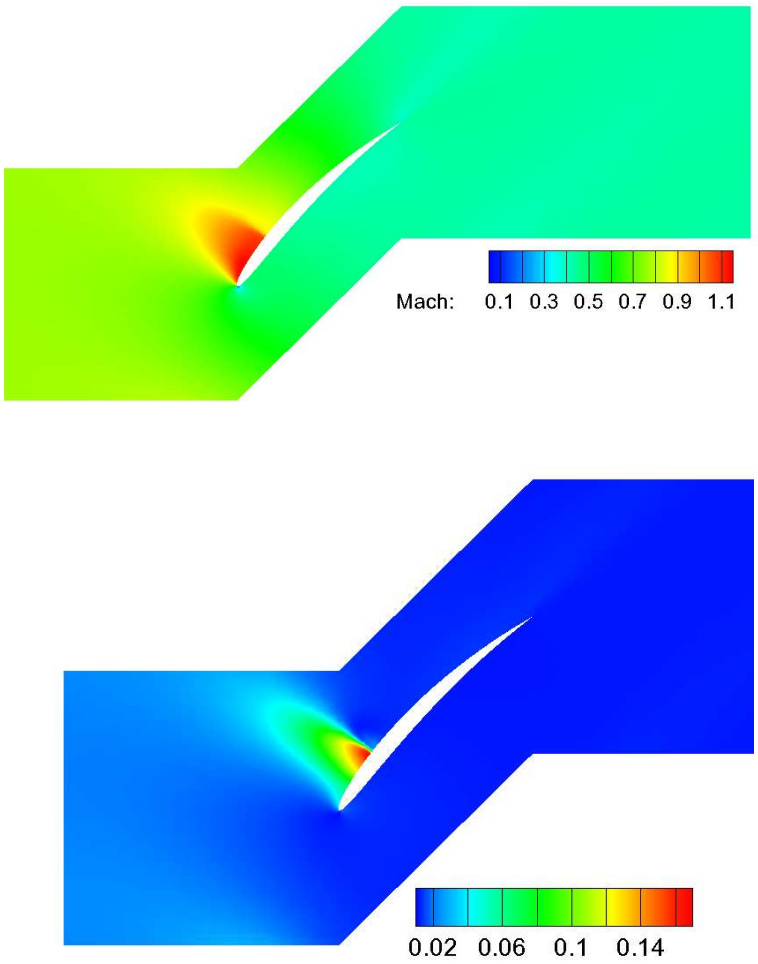
Τα αποτελέσματα φαίνονται στον πίνακα 4.1, όπου γίνεται και σύγκριση με τη μέθοδο του niPCE

Στο σχήμα 4.1 φαίνεται η μέση τιμή του αριθμού Mach .

	iPCE	niPCE	iPCE	niPCE
	$C=1$		$C=2$	
μ_{p_2/p_1}	1.2933	1.2936	1.2938	1.2937
σ_{p_2/p_1}	0.0257	0.0254	0.0251	0.0253
time (sec)	*2600*	*3200*	*10140*	*11200*

Πίνακας 4.1: *Ροή στον συμπιεστή, με αβέβαιες οριακές συνθήκες. Στατιστικές ροπές της αύξησης στατικής πίεσης, υπολογιζόμενες με το iPCE και το niPCE και σύγκριση κόστους*

Τα αποτελέσματα των δύο μεθόδων μοιάζουν να συμφωνούν μεταξύ τους.



Σχήμα 4.1: *Ροή στον συμπιεστή, με αβέβαιες οριακές συνθήκες. Μέση τιμή (πάνω) και τυπική απόκλιση (κάτω) του πεδίου του αριθμού Mach*

Κεφάλαιο 5

Συζυγής Μέθοδος στις εξισώσεις Euler

Σε αυτό το κεφάλαιο παρουσιάζονται οι συζυγείς εξισώσεις των εξισώσεων Euler και εφαρμόζεται σε αυτές το iPCE

5.1 Συνεχής Συζυγής Μέθοδος Επιφανειακού Ολοκληρώματος για την Ατριβή Ροή

Για τις εξισώσεις Euler, η επαυξημένη συνάρτηση ορίζεται ως

$$F_{aug} = F + \int_{\Omega} \mathbf{\Psi}^T \frac{\partial \boldsymbol{f}_i}{\partial x_i} d\Omega = F + \int_{\Omega} \mathbf{\Psi}^T A_i \frac{\partial \boldsymbol{U}}{\partial x_i} d\Omega \tag{5.1}$$

Η παράγωγός της ως προς τις μεταβλητές σχεδιασμού δίνεται

$$\delta F_{aug} = \delta F + \int_{\Omega} \mathbf{\Psi}^T \frac{\partial}{\partial b}(\frac{\partial f_i}{\partial x_i}) d\Omega + \int_{S} \mathbf{\Psi}^T \frac{\partial \boldsymbol{f}_i}{\partial x_i} n_l \delta x_l d\Omega \tag{5.2}$$

με το δF να δίνεται από

$$\delta F = \ \delta F_{SD} + \int_{S_W} \delta p (n_2 \cos a_\infty - n_1 \sin a_\infty) dS \tag{5.3}$$

$$\delta F_{SD} = \int_{S_W} p(\delta n_2 \cos a_\infty - \delta n_1 \sin a_\infty) dS + \int_{S_W} p(n_2 \cos a_\infty - n_1 \sin a_\infty) \delta(dS) \tag{5.4}$$

Ο όρος $\int_\Omega \mathbf{\Psi}^T \frac{\partial}{\partial b}(\frac{\partial f_i}{\partial x_i})d\Omega$ γράφεται ως

$$\int_\Omega \mathbf{\Psi}^T \frac{\partial}{\partial b}(\frac{\partial f_i}{\partial x_i})d\Omega = \underbrace{\int_S \mathbf{\Psi}^T \frac{\partial \boldsymbol{f}_i}{\partial x_i} n_i dS}_{A} - \int_\Omega \frac{\partial \mathbf{\Psi}^T}{\partial x_i} \frac{\partial f_i}{\partial b} d\Omega \tag{5.5}$$

Με το επιφανειακό ολοκλήρωμα A να γράφεται ως

$$A = \int_{S_W} \mathbf{\Psi}_{i+1}^T n_i \delta p dS + \int_{S_W} (\mathbf{\Psi}_{i+1}^T p - \mathbf{\Psi}^T f_i)\delta(n_i dS) - \int_{S_W} \mathbf{\Psi}^T \frac{\partial f_i}{\partial x_l} \delta x_l n_i dS \tag{5.6}$$

Συνεπώς, η επαυξημένη συνάρτηση δF_{aug}

$$\begin{aligned}\delta F_{aug} = \delta F + \int_\Omega \frac{\partial \mathbf{\Psi}^T}{\partial x_i}(\frac{\partial f_i}{\partial b})d\Omega + \int_{S_W} \mathbf{\Psi}^T \frac{\partial \boldsymbol{f}_i}{\partial x_i} n_l \delta x_l dS + \\ \int_{S_W} \mathbf{\Psi}_{i+1}^T n_i \delta p dS + \int_{S_W} (\mathbf{\Psi}_{i+1}^T p - \mathbf{\Psi}^T f_i)\delta(n_i dS) - \int_{S_W} \mathbf{\Psi}^T \frac{\partial f_i}{\partial x_l} \delta x_l n_i dS\end{aligned} \tag{5.7}$$

Με την πεδιακή συζυγή εξίσωση

$$\begin{aligned} A_i^T \frac{\partial \mathbf{\Psi}}{\partial x_i} = \mathbf{0} \ , \ \text{στο} \Omega \\ n_2 \cos a_\infty - n_1 \sin a_\infty + \Psi_{i+1} n_i = 0 \ , \ \text{στο} \ S \end{aligned} \tag{5.8}$$

Η τελική έκφραση των παραγώγων ευαισθησίας είναι

$$\begin{aligned}\delta F_{aug} = \delta F + \int_{S_W} \mathbf{\Psi}^T \frac{\partial \boldsymbol{f}_i}{\partial x_i} n_l \delta x_l dS + \int_{S_W} \mathbf{\Psi}_{i+1}^T n_i \delta p dS + \\ \int_{S_W} (\mathbf{\Psi}_{i+1}^T p - \mathbf{\Psi}^T f_i)\delta(n_i dS) - \int_{S_W} \mathbf{\Psi}^T \frac{\partial f_i}{\partial x_l} \delta x_l n_i dS\end{aligned} \tag{5.9}$$

5.2 Συνεχής Συζυγής Ενισχυμένη Μέθοδος Επιφανειακού Ολοκληρώματος για την Ατριβή Ροή

Για να παραχθεί η διατύπωση ενισχυμένη διατύπωση για την συνεχή συζυγή μέθοδο, αρχικά υποτίθεται ότι μεταξύ κάθε κύκλου βελτιστοποίησης λύνεται ένα σύνολο Λα-

πλασιανών ΜΔΕ για την μετατόπιση του πλέγματος, το οποίο εκφράζεται μαθηματικά ως

$$\boldsymbol{R}_i^m = \frac{\partial^2 m_i}{\partial x_j^2} = 0 \tag{5.10}$$

Το ολοκλήρωμα στον όγκο του όρου της εξίσωσης 5.10, πολλαπλασιαζόμενο με την αντίστοιχη συζυγή ποσότητα, προστίθεται στην επαυξημένη αντικειμενική συνάρτηση

$$F_{aug} = F + \int_\Omega \boldsymbol{\Psi}^T \frac{\partial f_i}{\partial x_i} d\Omega = F + \int_\Omega \boldsymbol{\Psi}^T A_i \frac{\partial \boldsymbol{U}}{\partial x_i} d\Omega + \int_\Omega m_i^a R_i^m d\Omega \tag{5.11}$$

και παραγωγίζεται

$$\delta F_{aug} = \delta F + \int_\Omega \boldsymbol{\Psi}^T \frac{\partial}{\partial b}\left(\frac{\partial f_i}{\partial x_i}\right) d\Omega + \int_S \boldsymbol{\Psi}^T \frac{\partial f_i}{\partial x_i} n_l \delta x_l d\Omega + \underbrace{\delta \int_\Omega m_i^a R_i^m d\Omega}_{B} \tag{5.12}$$

Για τον επιπλέον όρο B χρησιμοποιείται το θεώρημα του Gauss δύο φορές,

$$\begin{aligned} \delta \int_\Omega m_i^a R_i^m d\Omega = \int_S m_i^a n_j \frac{\partial(\delta x_i)}{\partial x_j} dS - \int_{S_W} \frac{\partial m_i^a}{\partial x_j} n_j \delta x_i dS + \\ \int_\Omega \frac{\partial^2 m_i^a}{\partial x_j^2} \delta x_j d\Omega + \int_S m_i^a R_i^m n_k \delta x_k dS \end{aligned} \tag{5.13}$$

Για τον υπολογισμό του επιπρόσθετου όρου της εξ. 5.13, απαιτείται η λύση της συζυγούς πεδιακής εξίσωσης μετατόπισης πλέγματος, η οποία εκφράζεται ως

$$\boldsymbol{R}_k^{m^a} = \frac{\partial^2 m_k^a}{\partial x_j^2} + \Psi_i \frac{\partial R_i}{\partial x_k} = 0 \tag{5.14}$$

Επιλύοντας την εξίσωση 5.14, η παράγωγος της επαυξημένης αντικειμενικής συνάρτησης γράφεται ως

$$\begin{aligned} \delta F_{aug} = \delta F + \int_{S_W} \boldsymbol{\Psi}^T \frac{\partial f_i}{\partial x_i} n_l \delta x_l dS + \int_{S_W} \boldsymbol{\Psi}_{i+1}^T n_i \delta p dS + \\ \int_{S_W} (\boldsymbol{\Psi}_{i+1}^T p - \boldsymbol{\Psi}^T f_i) \delta(n_i dS) - \int_{S_W} \boldsymbol{\Psi}^T \frac{\partial f_i}{\partial x_l} \delta x_l n_i dS - \int_{S_W} \frac{\partial m_i^a}{\partial x_j} n_j \delta x_i dS \end{aligned} \tag{5.15}$$

5.3 PCE στις Συζυγείς Εξισώσεις Euler

Για να παραχθούν οι επεμβατικές εξισώσεις ατριβούς ροής, ο τελεστής $\mathrm{G}^{\mathrm{q}}[.]$ εφαρμόζεται κατευθείαν στις εξισώσεις ροής, με τον ακόλουθο τρόπο

$$\begin{aligned} \mathrm{G}^{\mathrm{q}}\left[\frac{\partial f_i}{\partial x_i}\right] = \mathrm{G}^{\mathrm{q}}\left[A_i \, \frac{\partial \boldsymbol{U}}{\partial x_i}\right] = \mathbf{0} \ , \text{ στο } \Omega \\ \mathrm{G}^{\mathrm{q}}\left[u_i n_i\right] = \mathbf{0} \ , \text{ στο } S_W \\ \mathrm{G}^{\mathrm{q}}\left[\boldsymbol{U}\right] = \mathrm{G}^{\mathrm{q}}\left[\boldsymbol{U}_\infty\right] \ , \text{ στο } S_\infty \end{aligned} \tag{5.16}$$

Η αντικειμενική συνάρτηση ορίζεται ως

$$J = \sum_{j=0}^{q} \zeta_j F^j \tag{5.17}$$

Με την επαυξημένη αντικειμενική συνάρτηση να δίνεται από

$$J_{aug} = J + \int_\Omega \mathrm{G}^{\mathrm{q}}\left[\boldsymbol{\Psi}\right]^T \mathrm{G}^{\mathrm{q}}\left[\frac{\partial f_i}{\partial x_i}\right] d\Omega + \int_\Omega m_i^a R_i^m d\Omega \tag{5.18}$$

Με την παράγωγό της να είναι

$$\begin{aligned} \delta J_{aug} = \delta J + \int_\Omega \mathrm{G}^{\mathrm{q}}\left[\boldsymbol{\Psi}^T\right] \mathrm{G}^{\mathrm{q}}\left[\frac{\partial}{\partial b}\left(\frac{\partial f_i}{\partial x_i}\right)\right] d\Omega \\ + \int_{S_W} \mathrm{G}^{\mathrm{q}}\left[\boldsymbol{\Psi}^T\right] \mathrm{G}^{\mathrm{q}}\left[\frac{\partial f_i}{\partial x_i} n_l \delta x_l\right] d\Omega \\ + \delta \int_\Omega m_i^a R_i^m d\Omega \end{aligned} \tag{5.19}$$

όπου ισχύει ότι

$$\begin{aligned} &\delta J = \mathrm{G}^{\mathrm{q}}\left[\zeta\right]^T \mathrm{G}^{\mathrm{q}}\left[\delta F\right] \\ &= \mathrm{G}^{\mathrm{q}}\left[\zeta\right]^T \mathrm{G}^{\mathrm{q}}\left[\int_{S_W} \delta p (n_2 \cos a_\infty - n_1 \sin a_\infty) dS\right] + \mathrm{G}^{\mathrm{q}}\left[\zeta\right]^T \mathrm{G}^{\mathrm{q}}\left[\delta F_{SD}\right] \end{aligned} \tag{5.20}$$

Συνεπώς, προκύπτει ότι

$$\delta J_{aug} = \delta J + \int_{\Omega} \frac{\partial \, \mathrm{G^q}\left[\mathbf{\Psi}^T\right]}{\partial x_i} \mathrm{G^q}\left[\frac{\partial f_i}{\partial b}\right] d\Omega + \int_{S_W} \mathrm{G^q}\left[\mathbf{\Psi}^T\right] \mathrm{G^q}\left[\frac{\partial f_i}{\partial x_i} n_l \delta x_l\right] dS +$$
$$\int_{S_W} \mathrm{G^q}\left[\mathbf{\Psi}_{i+1}^T\right] \mathrm{G^q}\left[n_i \delta p\right] dS + \int_{S_W} \left(\mathrm{G^q}\left[\mathbf{\Psi}_{i+1}^T\right] \mathrm{G^q}\left[p\right] - \mathrm{G^q}\left[\mathbf{\Psi}^T\right] \mathrm{G^q}\left[f_i\right]\right) \mathrm{G^q}\left[\delta(n_i dS)\right]$$
$$- \int_{S_W} \mathrm{G^q}\left[\mathbf{\Psi}^T\right] \mathrm{G^q}\left[\frac{\partial f_i}{\partial x_l} \delta x_l n_i\right] dS - \int_{S_W} \frac{\partial m_i^a}{\partial x_j} n_j \delta x_i dS \tag{5.21}$$

Άρα, η συζυγής πεδιακή εξίσωση γράφεται, με τη βοήθεια του τελεστή Galerkin,

$$\mathrm{G^q}\left[A_i^T \frac{\partial \mathbf{\Psi}}{\partial x_i}\right] = \mathbf{0} \tag{5.22}$$

με τις οριακές της συνθήκες να γράφονται ως

$$\mathrm{G^q}\left[\zeta(n_2 \cos a_\infty - n_1 \sin a_\infty) + \Psi_{i+1} n_i\right] = \mathbf{0} \tag{5.23}$$

Η τελική έκφραση των παραγώγων ευαισθησίας είναι

$$\delta J = \mathrm{G^q}\left[\zeta\right]^T \mathrm{G^q}\left[\delta F_{SD}\right] + \mathrm{G^q}\left[\delta F_{SD}^{\Psi}\right]^T \mathrm{G^q}\left[1\right] - \int_{S_W} \frac{\partial m_i^a}{\partial x_j} n_j \delta x_i dS \tag{5.24}$$

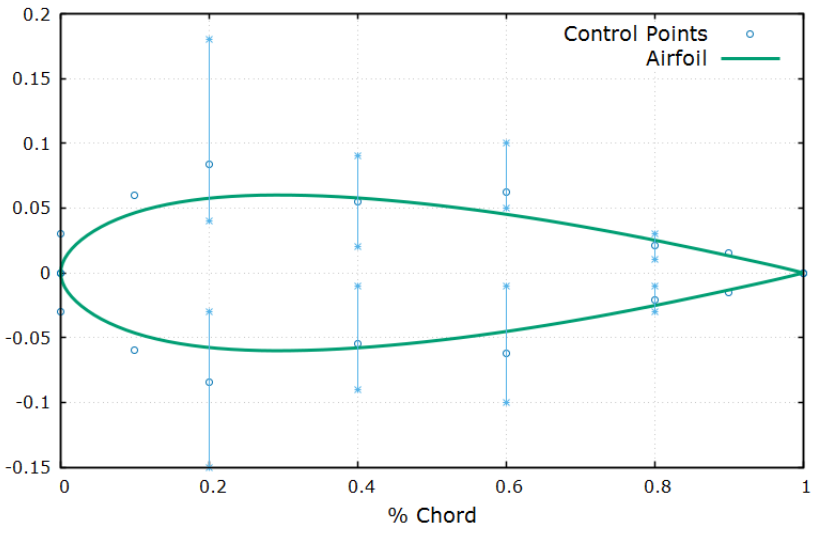
όπου

$$\mathrm{G^q}\left[\delta F_{SD}^{\Psi}\right]^T \mathrm{G^q}\left[1\right] = \int_{S_W} \left(\mathrm{G^q}\left[\Psi_{i+1}\right] \mathrm{G^q}\left[p\right] - \mathrm{G^q}\left[\mathbf{\Psi}\right]^T \mathrm{G^q}\left[f_i\right]\right) \delta(n_i dS) \tag{5.25}$$

Κεφάλαιο 6

Αριθμητικές Εφαρμογές

Σε αυτό το κεφάλαιο παρουσιάζονται αριθμητικές εφαρμογές της ενισχυμένης μεθόδου επιφανειακού ολοκληρώματος.



Σχήμα 6.1: *Βελτιστοποίηση μορφής αεροτομής. Αρχική αεροτομή και σημεία ελέγχου Bezier.*

6.1 Βελτιστοποίηση δίχως αβεβαιότητες

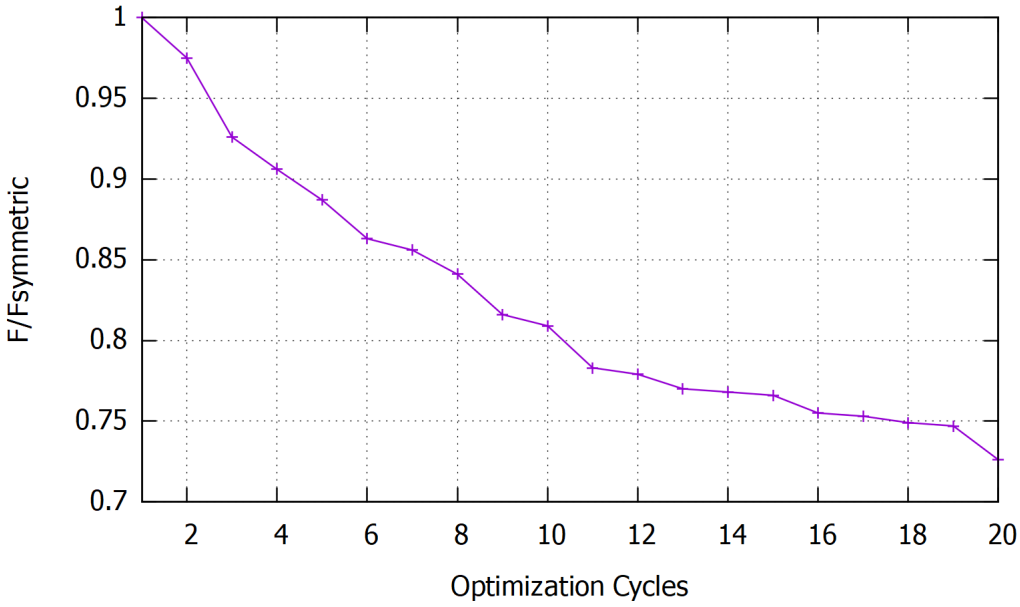
Στην πρώτη περίπτωση, βελτιστοποιείται αεροτομή με σκοπό την ελαχιστοποίηση του συντελεστή οπισθέλκουσας C_D

$$F = C_D \tag{6.1}$$

με οριακές συνθήκες

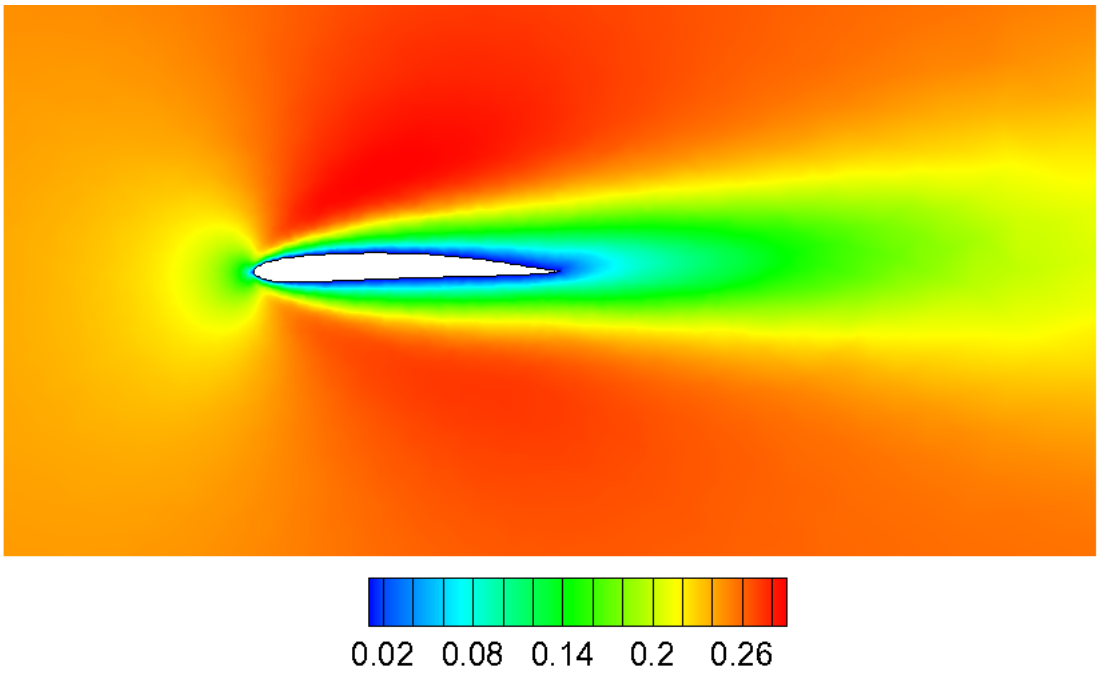
$$M_\infty = 0.55 \quad , \quad a_\infty = 2.5^o \quad , \quad Re = 5000 \tag{6.2}$$

Τα αποτελέσματα παρουσιάζονται συνοπτικά στο σχήμα 6.2



Σχήμα 6.2: *Ελαχιστοποίηση συντελεστή οπισθέλκουσας. Πορεία της αντικειμενικής συνάρτησης.*

Στο σχήμα 6.3 παρουσιάζεται ο αριθμός Mach γύρω από την αεροτομή.



Σχήμα 6.3: *Ελαχιστοποίηση συντελεστή οπισθέλκουσας. Αριθμός Mach γύρω από την αεροτομή.*

6.2 Βελτιστοποίηση υπό αβεβαιότητες

Σε αυτήν την περίπτωση, η αβεβαιότητα εισάγεται μέσω των οριακών συνθηκών,

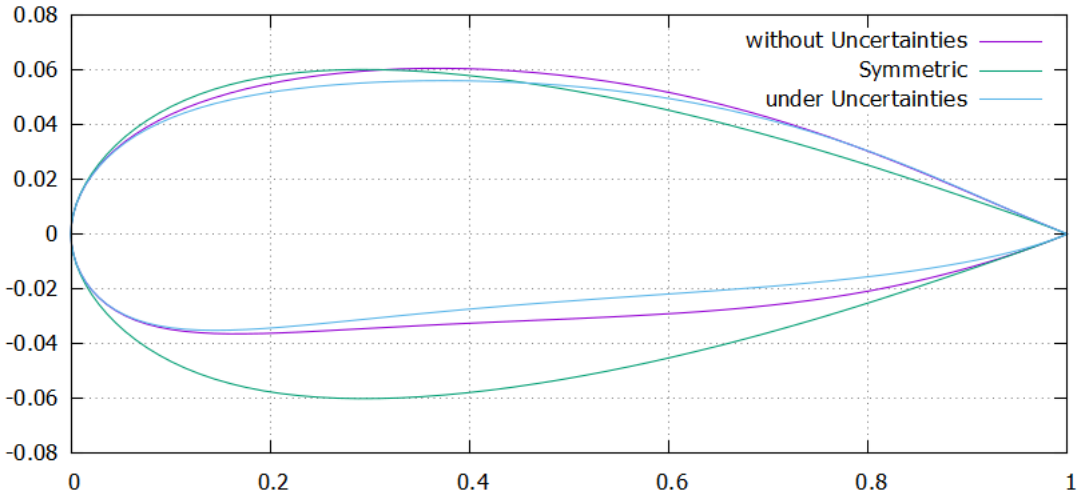
$$M_\infty \sim \mathcal{N}(0.55, 0.05) \quad , \quad a_\infty \sim \mathcal{U}(2.0^o, 3.0^o) \quad , \quad Re \sim \mathcal{N}(5000, 200)$$

Τα αποτελέσματα παρουσιάζονται συνοπτικά στον πίνακα 6.1

	χωρίς ΠΑ	με ΠΑ, $C = 1$	με ΠΑ, $C = 2$	με ΠΑ, $C = 3$
μ_{C_D}	$6.81 \cdot 10^{-2}$	$6.89 \cdot 10^{-2}$	$6.86 \cdot 10^{-2}$	$6.84 \cdot 10^{-2}$
σ_{C_D}	$1.17 \cdot 10^{-3}$	$1.09 \cdot 10^{-3}$	$1.05 \cdot 10^{-3}$	$1.02 \cdot 10^{-3}$

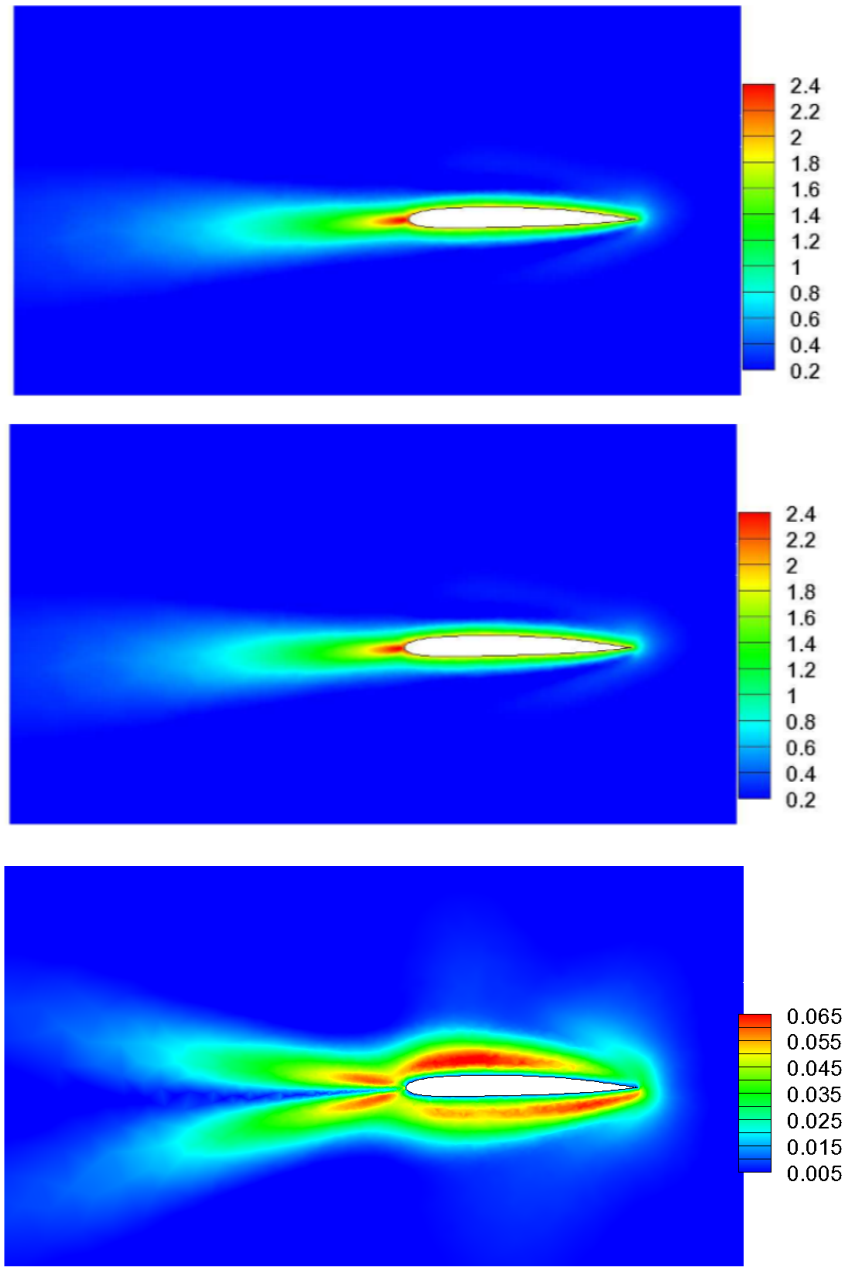
Πίνακας 6.1: *Μέση τιμή και τυπική απόκλιση του συντελεστή οπισθέλκουσας για* $C = 1, 2, 3$

Στο σχήμα 6.4, παρουσιάζεται η αρχική αεροτομή σε σύγκριση με το σχήμα που προέκυψε από τη βελτιστοποίηση



Σχήμα 6.4: *Σύγκριση αεροτομών, δίχως αβεβαιότητα (μοβ), υπό αβεβαιότητα (μπλε) και αρχική (πράσινο)*

Ενώ στο παρακάτω σχήμα παρουσιάζεται η αδιάστατη συζυγής ταχύτητα της αεροτομής, όπως προέκυψε από τη βελτιστοποίηση



Σχήμα 6.5: *Αδιάστατη συζυγής ταχύτητα, σύγκριση δίχως αβεβαιότητες (πάνω), μέση τιμή (μέση) και τυπική απόκλιση (κάτω) για* $C = 3$

Κεφάλαιο 7

Συμπεράσματα

Κάποια βασικά συμπεράσματα που μπορούν να εξαχθούν είναι τα ακόλουθα

1. Εισάχθηκε μια νέα, άκοπη μέθοδος, η οποία είναι ανεξάρτητη της ΜΔΕ που μελετάται.
2. Η μέδοδος αυτή μπορεί να εφαρμοστεί άκοπα, δίχως εκτενείς παρεμβάσεις στον πηγαίο κώδικα που λύνει το πρόβλημα δίχως αβεβαιότητες. Αυτό είναι σημαντικό καθώς μειώνεται ο χρόνος ανάπτυξης και προγραμματισμού που απαιτείται για την εφαρμογή της.
3. Η μέθοδος μπορεί να εφαρμοστεί με απαιτήσεις μνήμης οι οποίες είναι συγκρίσιμες με αυτές του κώδικα δίχως αβεβαιότητες. Αυτό μειώνει σημαντικά τις ανάγκες μνήμης του κώδικα, οι οποίες είναι συνήθως πολύ σημαντικές σε μεθόδους UQ
4. Αποδείχτηκε ότι το iPCE είναι ευσταθές, αν το αριθμητικό σχήμα του κώδικα δίχως αβεβαιότητες στον οποίον στηρίζεται είναι ευσταθές.
5. Για αριθμό αβέβαιων μεταβλητών $m < 8$, το iPCE ήταν από 10% μέχρι και 40% ταχύτερο από την αντίστοιχη niPCE μέθοδο.
6. Η μέθοδος iPCE , όπως εφαρμόστηκε, μπορεί να επεκταθεί για χρήση στη συνεχή συζυγή μέθοδο, με σκοπό τη βελτιστοποίηση. Για αυτό, απαιτούνται ελάχιστες παρεμβάσεις στον αρχικό κώδικα, και η μέθοδος είναι εξίσου άκοπη στην εφαρμογή της

References

[1] A.G. Liatsikouras, V.G. Asouti, K.C. Giannakoglou, G. Pierrot, and M. Megahed. Aerodynamic shape optimization under flow uncertainties using non-intrusive polynomial chaos and evolutionary algorithms. In *2nd ECCOMAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2017)*, Rhodes Island, Greece, June 15-17 2017.

[2] R. Duvigneau. Aerodynamic shape optimization with uncertain operating conditions using metamodels. RR-6143, INRIA, 2007.

[3] D. Lee, L. Gonzalez, J. Periaux, and K. Srinivas. Robust design in aerodynamics using 3rd-order sensitivity analysis based on discrete adjoint. application to quasi-1d flows. *Computers and Fluids*, 37:565–583, 2008.

[4] J. Vigouroux, L. Deshayes, S. Foufou, J. Filliben, L. Welsch, and M. Donmez. Robust design of an evolutionary algorithm for machining optimization problems. *Journal of Computing and Information Science in Engineering*, 2016.

[5] S. Ho and Y. Shiyou. A fast robust optimization methodology based on polynomial chaos and evolutionary algorithm for inverse problems. *IEEE Transactions on Magnetics*, 48:259–262, ???

[6] K.C. Giannakoglou, D.I. Papadimitriou, E.M. Papoutsis-Kiachagias, and I. Kavvadias. Adjoint methods for shape optimization and robust design in fluid mechanics. In *OPT-i, International Conference on Engineering and Applied Sciences Optimization*, Kos Island, Greece, June 4-6 2014.

[7] K.C. Giannakoglou, D. Papadimitriou, E Papoutsis-Kiachagias, and I.S. Kavvadias. Adjoint methods for shape optimization and robust design in fluid mechanics. pages 2252–2265, 01 2014.