National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Parallel CFD & Optimization Unit

Computational Cost Reduction and Stabilization of Solvers of the Continuous Adjoint Method in Aerodynamic Shape Optimization, with or without Uncertainties

PhD Thesis

Themistoklis Skamagkis

Supervisor: Kyriakos C. Giannakoglou,
Professor NTUA

Athens, 2023

National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Parallel CFD & Optimization Unit

Computational Cost Reduction and Stabilization of Solvers of the Continuous Adjoint Method in Aerodynamic Shape Optimization, with or without Uncertainties

PhD Thesis

Themistoklis Skamagkis

Examination Committee:

1. K. Giannakoglou (Supervisor)*,
   Professor, NTUA, School of Mechanical Engineering
2. K. Mathioudakis*,
   Professor, NTUA, School of Mechanical Engineering
3. V. Riziotis*,
   Associate Professor, NTUA, School of Mechanical Engineering
4. S. Voutsinas,
   Professor, NTUA, School of Mechanical Engineering
5. D. Bouris,
   Professor, NTUA, School of Mechanical Engineering
6. I. Nikolos,
   Professor, TUC, School of Production Engineering and Management
7. G. Papadakis,
   Assistant Professor, NTUA, School of Naval Architecture and Marine Engineering

* Member of the Advisory Committee.

Athens, 2023

Abstract

The objective of this thesis is the development and assessment of techniques for the reduction of the computational cost of gradient-based aerodynamic shape optimization for flows that exhibit mild unsteadiness and/or in the presence of uncertainties. All these techniques are based on the continuous adjoint method and have been developed in the OpenFOAM environment, as a complement to the adjoint solver developed and made publicly available by the Parallel CFD & Optimization Unit of the National Technical University of Athens (PCOpt/NTUA). They are first validated in academic cases and are later used for aerodynamic shape optimization in industrial applications.

It is well known that adjoint-based shape optimization using unsteady solvers is costly and/or memory demanding. For this reason, it is sparingly used in real-world applications. When mild unsteadiness is present, steady primal and adjoint solvers can be used instead provided that convergence difficulties are properly treated. In this thesis and for this type of applications, the steady primal and the corresponding (continuous) adjoint solvers are both stabilized by implementing the Recursive Projection Method (RPM), a stabilization method for iterative procedures. Upon completion of the optimization using steady solvers, unsteady re-evaluations of the optimized solutions should normally be used to confirm a reduction in the time-averaged objective function. Thus, shape optimization costs less, even by an order of magnitude, than an optimization using unsteady adjoint. This approach is further tested in flows with vortex shedding wherein steady flow solutions are computed with the help of the RPM. Although vortex shedding problems can be optimized with steady solvers, optimization using unsteady adjoint cannot be avoided. However, the latter starts with a good initialization, computed by the steady solvers, and this reduces a lot the overall CPU cost.

In complex cases, the convergence difficulties of the primal and adjoint solvers are far more prevalent. For instance, the RPM may not necessarily ensure convergence of the adjoint solver on its own. This might be due to the lack of good grid quality, often in applications involving complex geometries (e.g. a car with wheels, side mirrors, surface details etc.), or the existence of many unstable modes causing rapid divergence of the solver. To overcome this difficulty, the controlled damping of the Adjoint Transposed-Convection (ATC) term in the adjoint momentum equations is occasionally employed along with the RPM. In the literature, the damping or, even, elimination of the ATC term is frequently used for the stabilization of continuous adjoint solvers. The effect of this treatment on the computed Sensitivity Derivatives (SDs) is studied in cases in which the RPM, on its own, makes the adjoint solver converge. Controlled/mild ATC damping (if needed), as applied in this thesis, proves to be harmless and greatly contributes to robustness. On the contrary, the

uncontrolled damping or, even, elimination of the ATC, an action that would have been required in the absence of the RPM, may become harmful. In addition, the RPM stabilization helps avoiding the solution of the adjoint equations based on averaged primal fields.

In the second part, this thesis tackles aerodynamic shape optimization in the presence of uncertainties. In such problems, Uncertainty Quantification (UQ) techniques are used to measure the effect of uncertain variables on a Quantity of Interest (QoI). The inclusion of UQ in a shape optimization loop may noticeably increase the computational cost. Two UQ methods, namely the First-Order Second-Moment (FOSM) and the Adjoint-assisted Polynomial Chaos Expansion (APCE) are studied. Both FOSM and APCE compute the first two statistical moments (mean value and standard deviation) of the QoI and the objective function to be minimized is their weighted sum. Gradient-based optimization with such an objective function requires mixed derivatives of the QoI with respect to the design and uncertain variables. However, in practice, only the projection of the matrix of mixed derivatives on some vectors needs to be computed, instead of the matrix itself. These projections are herein exploited in both UQ methods to reduce the CPU cost, yielding projected counterparts of these two methods. Their key point is that the cost of computing the projected matrix of mixed derivatives does not scale with the number of either the design or the uncertain variables.

In the projected FOSM (pFOSM), the cost per optimization cycle with gradient-based methods is 4 Equivalent Flow Solutions (EFS), i.e. as if the flow equations are solved 4 times. The cost of the projected APCE (pAPCE) is equal to 4L EFS (L is the number of collocation points needed in the regression). The gain in CPU cost is significant even if only a few uncertain variables are involved. The pAPCE method is more expensive than pFOSM but computes more accurate statistical moments. Over and above to some laminar flow problems, pFOSM and pAPCE are demonstrated in aerodynamic shape optimization problems with turbulent flows, solved using the Spalart-Allmaras model and its adjoint.

The industrial applications involve aerodynamic shape optimization of ground vehicles using steady solvers and the RPM and of an aircraft wing in the presence of flow uncertainties, to assess pFOSM and pAPCE.

Keywords: Aerodynamic Shape Optimization, Continuous Adjoint, Robust Design Optimization, Recursive Projection Method, Stability, Adjoint Transposed-Convection, Uncertainty Quantification, Polynomial Chaos Expansion, OpenFOAM

Περίληψη

Στη διδακτορική διατριβή αναπτύσσονται τεχνικές με σκοπό τη μείωση του υπολογιστικού κόστους αιτιοκρατικών μεθόδων βελτιστοποίησης μορφής στην αεροδυναμική, για ροές με ήπια χρονική αστάθεια ή/και με αβεβαιότητες. Όλες οι τεχνικές βασίζονται στη συνεχή συζυγή μέθοδο και αναπτύχθηκαν σε περιβάλλον OpenFOAM, συμπληρωματικά του επιλύτη συζυγών εξισώσεων ανοιχτής πρόσβασης ο οποίος έχει αναπτυχθεί από τη Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης του ΕΜΠ. Αρχικά, γίνεται πιστοποίηση των τεχνικών σε ακαδημαϊκές και ακολουθεί η χρήση τους σε βιομηχανικές εφαρμογές.

Η συζυγής μέθοδος για χρονικά μη-μόνιμους επιλύτες έχει μεγάλες απαιτήσεις σε υπολογιστικό κόστος ή/και αποθήκευση δεδομένων. Για αυτό και δεν χρησιμοποιείται συχνά στη βελτιστοποίηση μορφής σε βιομηχανικές εφαρμογές. Οι μόνιμοι επιλύτες πρωτευουσών και συζυγών εξισώσεων μπορούν, κατά περίπτωση, να χρησιμοποιηθούν όταν η ροή παρουσιάζει ήπια χρονική αστάθεια. Στη διατριβή αυτή εφαρμόζεται η Μέθοδος Αναδρομικών Προβολών (Recursive Projection Method, RPM) για την αντιμετώπιση των δυσκολιών σύγκλισης που παρουσιάζουν οι χρονικά μόνιμοι πρωτεύοντες και (συνεχείς) συζυγείς επιλύτες λόγω της χρονικής αστάθειας της ροής. Με το πέρας της βελτιστοποίησης με μόνιμους επιλύτες, οι βελτιστοποιημένες λύσεις επαναξιολογούνται από μη-μόνιμους επιλύτες πρωτεύοντος προβλήματος για να επιβεβαιωθεί η μείωση του χρονικού μέσου των συναρτήσεων στόχου. Έτσι, η βελτιστοποίηση γίνεται με υπολογιστικό κόστος ως και κατά μία τάξη μεγέθους χαμηλότερο από ότι με χρονικά μη-μόνιμους επιλύτες. Η τεχνική αυτή χρησιμοποιείται επίσης για βελτιστοποίηση μορφής σε ροές όπου εκλύονται στρόβιλοι. Παρότι μπορούν να υπολογιστούν βελτιωμένες λύσεις για τέτοιου είδους ροές με μόνιμους επιλύτες, δεν μπορεί εν τέλει να αποφευχθεί ένας βρόχος βελτιστοποίησης με μη-μόνιμους επιλύτες. Το συνολικό κόστος, όμως, μειώνεται χρησιμοποιώντας τη βελτιστοποιημένη λύση με μόνιμους επιλύτες ως αρχικοποίηση για τη βελτιστοποίηση με μη-μόνιμους.

Σε πιο πολύπλοκες εφαρμογές, οι δυσκολίες σύγκλισης των μόνιμων πρωτεύοντων και συζυγών επιλυτών, ακόμα και με την RPM, είναι πολύ πιο έντονες. Αιτία για αυτό μπορεί να είναι η ποιότητα του πλέγματος γύρω από περίπλοκες γεωμετρίες ή η ύπαρξη πληθώρας ιδιοσυχνοτήτων που προκαλούν τη γρήγορη απόκλιση του συζυγούς επιλύτη. Προκειμένου να αποφευχθούν τέτοια προβλήματα, χρησιμοποιείται (επιπλέον της RPM) ελεγχόμενη απόσβεση της Συζυγούς Ανάστροφης Συμμεταφοράς (Adjoint Transposed-Convection, ATC), ενός όρου που εμφανίζεται στις συνεχείς συζυγείς εξισώσεις ορμής. Η απόσβεση ή απαλοιφή αυτού του όρου χρησιμοποιείται συχνά στη βιβλιογραφία για τη σταθεροποίηση συνεχών συζυγών επιλυτών. Μελετάται, εδώ, η επίδραση που έχει αυτή η πρακτική στην ακρίβεια των παραγώγων ευαισθησίας σε περιπτώσεις όπου η RPM επιτυγχάνει από μόνη της τη σταθεροποίηση του συζυγούς επιλύτη. Η ελεγχόμενη απόσβεση της ATC (εφόσον απαιτείται) αποδεικνύεται πρακτικά

αβλαβής και συνεισφέρει σημαντικά στην ευστάθεια των συζυγών επιλυτών. Αντίθετα, η ανεξέλεγκτη απόσβεση του όρου απουσία της RPM είναι αρκετά συχνά επιζήμια. Επίσης, χάριν της RPM, αποφεύγεται η επίλυση συζυγών εξισώσεων βασισμένων στη μέση τιμή των μη-συγκλιμένων πρωτευόντων πεδίων.

Το δεύτερο σκέλος της διατριβής αφορά την αεροδυναμική βελτιστοποίηση μορφής υπό αβεβαιότητες. Σε τέτοια προβλήματα, προσδιορίζεται ποσοτικά η απόκριση μιας Ποσότητας Ενδιαφέροντος (Quantity of Interest, QoI) ως προς ένα σύνολο μεταβλητών αβεβαιότητας με τις λεγόμενες μεθόδους Ποσοτικοποίησης Αβεβαιότητας (Uncertainty Quantification, UQ). Αυτό αυξάνει σημαντικά το υπολογιστικό κόστος ενός βρόχου βελτιστοποίησης. Χρησιμοποιούνται οι μέθοδοι Πρώτης-Τάξης Δεύτερης-Ροπής (First-Order Second-Moment, FOSM) και Αναπτύγματος Πολυωνυμικού Χάους υποβοηθούμενου από τη Συζυγή Μέθοδο (Adjoint-assisted Polynomial Chaos Expansion, APCE) για τον υπολογισμό των δύο πρώτων στατιστικών ροπών (μέσης τιμής και τυπικής απόκλισης) μιας QoI. Το σταθμισμένο άθροισμα αυτών των ροπών χρησιμοποιείται ως συνάρτηση στόχος. Για μια τέτοια συνάρτηση, η βελτιστοποίηση με παραγώγους ευαισθησίας απαιτεί τις μικτές παραγώγους της QoI ως προς τις μεταβλητές σχεδιασμού και αβεβαιότητας. Καθώς, όμως, αρκεί μόνο ο υπολογισμός της προβολής του μητρώου των μικτών παραγώγων σε διανύσματα, αναπτύσσονται δύο μέθοδοι βελτιστοποίησης με προβολές, οι pFOSM και pAPCE, σε αντιστοιχία με τις δύο μεθόδους UQ που προαναφέρθηκαν. Βασικό κέρδος από τη χρήση των μεθόδων προβολής είναι πως το κόστος υπολογισμού του προβεβλημένου μητρώου μικτών παραγώγων δεν αυξάνεται με το πλήθος των μεταβλητών αβεβαιότητας ή σχεδιασμού.

Στην pFOSM, το κόστος ανά κύκλο βελτιστοποίησης είναι ίσο με 4 Υπολογιστικές Μονάδες (Equivalent Flow Solutions, EFS), ως δηλαδή να λύνεται το πρόβλημα ροής 4 φορές. Στην pAPCE, το κόστος είναι ίσο με $4L$ EFS (L είναι το πλήθος των σημείων δειγματοληψίας στην παλινδρόμηση). Το κέρδος είναι σημαντικό ακόμα και για μικρό πλήθος μεταβλητών αβεβαιότητας. Η pAPCE έχει μεγαλύτερο υπολογιστικό κόστος από την pFOSM, αλλά υπολογίζει στατιστικές ροπές με μεγαλύτερη ακρίβεια. Οι δύο μέθοδοι παρουσιάζονται σε προβλήματα αεροδυναμικής βελτιστοποίησης μορφής για στρωτές και τυρβώδεις ροές. Στις τελευταίες, η επίλυση γίνεται με χρήση του μοντέλου τύρβης των Spalart-Allmaras και της συζυγούς του εξίσωσης.

Οι βιομηχανικές εφαρμογές περιλαμβάνουν την αεροδυναμική βελτιστοποίηση μορφής οχημάτων επίγειας μεταφοράς με χρήση μόνιμων επιλυτών σε συνδυασμό με την RPM και μιας πτέρυγας αεροσκάφους παρουσία αβεβαιοτήτων ως προς τις συνθήκες ροής με χρήση των pFOSM και pAPCE.

Λέξεις κλειδιά: Αεροδυναμική Βελτιστοποίηση Μορφής, Συνεχής Συζυγής Μέθοδος, Στιβαρός Σχεδιασμός, Μέθοδος Αναδρομικών Προβολών, Ευστάθεια Εξισώσεων, Συζυγής Ανάστροφη Συμμεταφορά, Ποσοτικοποίηση Αβεβαιότητας, Ανάπτυγμα Πολυωνυμικού Χάους, OpenFOAM

Acknowledgments

I would like to thank all those who, in one way or another, contributed to the completion of this thesis.

First and foremost, I would like to sincerely thank my thesis advisor, Kyriakos Giannakoglou, Professor NTUA, for offering me the chance to work on this intriguing topic. I would also like to thank him for the diligent proofreading of all scientific texts as well as this one, the helpful and much needed criticism of my work and his commitment in time and effort towards guiding me in the right direction throughout the duration of this undertaking.

Also, I would like to thank the other members of the advisory committee, Vasilis Riziotis, Associate Professor NTUA, and Konstantinos Mathioudakis, Professor NTUA, for their suggestions and comments on the presentation of this thesis.

I would also like to express my deep gratitude towards the other members of the PCOpt/NTUA group for their friendship and support. Foremost, a sincere thank you goes to Dr. Vaggelis Papoutsis-Kiachagias for his patience and guidance during my familiarization with the adjoint method and the OpenFOAM code, the helpful tips provided at key points of this thesis and his friendship, all of which are much appreciated. I am grateful to Dr. Varvara Asouti for the administration of the computational infrastructure of PCOpt/NTUA which enabled the completion of this thesis. Special thanks to Dr. Yiannis Vrionis for the long-running friendly discussions, both scientific and not, and helpful tips. I would also like to thank Dr. Kontantinos Tsiakas, Dr. Konstantinos Samouchos, Dr. Morteza Monfaredi, Dr. Konstantinos Gkaragkounis, Dr. Dimitris Kapsoulis and PhD students Andreas Margetis, Christos Karliampas, Bhargav Chitneedi, Marina Kontou, Erfan Farhikhteh, Nikos Galanos and Andreas Chrysolouris for their friendship and for creating a pleasant working environment.

The first part of this thesis was funded by the Bayerische Motoren Werke (BMW) corporation. Additionally, the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT) funded research activities, part of which contributed to the second part of this thesis.

Finally, I am much obliged to my family and friends for the substantial moral support without which, the completion of this thesis would have felt impossible to do.

Acronyms

AD	Automatic Differentiation
APCE	Adjoint-assisted Polynomial Chaos Expansion
ATC	Adjoint Transposed-Convection
BC	boundary condition
BFGS	Broyden–Fletcher–Goldfarb–Shanno
CFD	Computational Fluid Dynamics
CP	control point
CV	Coefficient of Variation
DAE	Differential-Algebraic Equation
DD	Direct Differentiation
E-SI	Enhanced Surface Integral
EA	Evolutionary Algorithm
EFS	Equivalent Flow Solution
FAE	Field Adjoint Equation
FD	Finite Difference
FI	Field Integral
FOSM	First-Order Second-Moment
FPI	Fixed-point Iteration
GDM	Grid Displacement Model
iPCE	intrusive PCE
MC	Monte Carlo
MoM	Method of Moments

niPCE	non-intrusive PCE
NURBS	Non-uniform Rational B-Splines
pAPCE	projected APCE
PCE	Polynomial Chaos Expansion
PCOpt/NTUA	Parallel CFD & Optimization Unit of the National Technical University of Athens
PDE	Partial Differential Equation
PDF	Probability Density Function
pFOSM	projected FOSM
POD	Proper Orthogonal Decomposition
QoI	Quantity of Interest
r.h.s.	right-hand-side
RANS	Reynolds-Averaged Navier-Stokes
RDO	Robust Design Optimization
RPM	Recursive Projection Method
SD	Sensitivity Derivative
SI	Surface Integral
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
SOSM	Second-Order Second-Moment
SQP	Sequential Quadratic Programming
SVD	Singular Value Decomposition
UQ	Uncertainty Quantification
URANS	Unsteady Reynolds-Averaged Navier-Stokes (RANS)
VBS	Volumetric B-Splines
w.r.t.	with respect to

Contents

Chapter 1

Introduction

Optimization is an integral part of the design process and, following the advancements in technology and the increase in computational power of recent years, computer simulations have become a standard tool for aiding engineers towards optimizing designs across multiple fields. All of this is supported by the numerical methods that have been developed for solving the Partial Differential Equations (PDEs) that describe the physics of the problem at hand. In the field of fluid mechanics, after years of intensive research, Computational Fluid Dynamics (CFD) has become more competent and is nowadays routinely used to test new designs, enhance the engineer's understanding of the underlying physical phenomena and complement existing skills and approaches for design. This has been followed by the combination of control theory with CFD for inverse design and optimization of aerodynamic components. The attractiveness of simulation-driven design lies in the promise of lower cost compared to established design practices. However, significant challenges are still present and relate to the efficiency, robustness and fidelity of the numerical methods and models; for the methods to be of practical use, shorter turnaround times are important. This thesis contributes to the development of faster, more robust and efficient methods and practices for CFD-based optimization, even in the presence of uncertainties.

The three main components that drive an optimization are the objective function, the constraints and the design variables. To compute the objective function value, the so-called primal problem is solved, which, in mechanics, often involves the solution of a set of governing PDEs i.e. the primal equations. In CFD-based optimization, the primal equations correspond to the flow PDEs and the objective function is usually a surface or a field integral whose value depends on the so-computed flow variables, also referred to as the primal ones. It is usual for the constraint functions to be of this form as well. The design variables serve as an input to the primal problem. The goal of the optimization method is to compute the design variables that minimize or maximize the objective function while meeting the constraints. The same optimization

methods can be used either for minimization or maximization since maximization problems can be formulated as minimization ones, as adopted throughout this thesis for all optimization problems.

Optimization, within a field of mechanics, can be broadly classified in two categories, depending on whether the topology or the shape of a design component is being optimized. The design variables are essentially different between these two types. Topology optimization mostly involves the distribution of a material within the design space. A typical example from CFD could be a heat exchanger with predefined inlet and outlet sections that is initially regarded as a box filled with solid or fluid and, gradually, fluid or solid regions emerge and eventually form an interconnected system of ducts. Shape optimization on the other hand begins with a pre-defined connectivity pattern or follows from a topology optimization loop and seeks to find the optimal shape of the design. Recent advances even combine these two types of optimization in structural applications [17, 101]. This thesis deals only with shape optimization problems. However, the developed methods and tools can readily be extended to topology optimization, too.

In shape optimization, the choice of the design variables plays a crucial role. In case the surface is somehow parameterized, its shape is controlled by a few design variables. The parameterization of the surface offers ease of its manipulation and/or deformation, usually with guaranteed surface smoothness. Widespread, general-purpose techniques are Non-uniform Rational B-Splines (NURBS) [124] or free-form deformation with Volumetric B-Splines (VBS) [67, 88, 118] to name a few. Some techniques can be used to parameterize the volume as well and offer more flexibility in the manipulation of the computational grid. Other techniques may additionally include engineering information, such as the PARSEC parameterization, introduced by [144], which is customized for airfoils. The parameters of these techniques serve as the design variables which are modified to effectively control the shape. Each technique offers different advantages and disadvantages. A few non-restrictive guidelines are offered for determining what constitutes a successful technique in a comprehensive review on multidisciplinary shape optimization by [137]. In nodal-based optimization, the design variables are themselves the nodal coordinates of the surface and a richer design space can be explored because the surface nodes are controlled directly. However, it may be more difficult to impose geometric constraints and the independent displacement of each node may lead to irregular shapes. A regularization function is usually employed to ensure smoothness and manufacturability of the optimized shape. For more details on nodal-based shape optimization, [8] offers a review on latest developments for various applications including CFD. In most applications, surface morphing is usually accompanied by grid deformation and re-adjustment to the new surface instead of regenerating the grid.

Optimization is further classified into stochastic and deterministic,

depending on the strategy used to find the optimal values of the design variables [103]. Stochastic optimization incorporates randomness in order to heuristically find the optimum. Compared to deterministic methods, there is a greater chance at finding the global optimum even without running the search for an infinitely long time [152]. Stochastic optimization is frequently performed using population-based Evolutionary Algorithms (EAs) which mimic natural Darwinian evolution. A population of candidate solutions (i.e. sets of design variables) is initialized randomly and is improved iteratively, with each iteration representing a generation, following a "survival of the fittest" strategy. In each generation, the candidate solutions are evaluated, the best ones are withheld in the population, the worst ones are discarded and new ones are generated by combining features from the best ones through evolution operations such as crossover and mutation. The iterations finally converge to better solutions. Optimization problems with many objectives and constraints are easily handled by the EAs. On the other hand, in case the number of design variables is great, the EAs require many generations to find better solutions and this also increases the number of evaluations of candidate solutions. Thus, the optimization turnaround time can be high in case CFD is involved in these evaluations. Various techniques are used to reduce the turnaround time. Parallel EAs [146] take advantage of parallel computer architectures to perform the evaluations concurrently. Distributed EAs split the population in clusters, called demes or islands, and use variations of the evolution operations in each one (higher mutation probability, elitism etc). Hierarchical EAs comprise more than one level of distributed EAs and use a different evolution strategy or problem parameterization in each level [41]. Additionally, meta-models, produced with various techniques, are frequently used which operate as cheap surrogate functions that can significantly speed up the evaluation process. Numerous PhD theses [35, 62, 60, 9, 71, 77, 31, 61], completed at the Parallel CFD & Optimization Unit of the National Technical University of Athens (PCOpt/NTUA), made contributions to the above techniques and the development of the Evolutionary Algorithms SYstem (EASY) software. Stochastic methods are beyond the scope of this thesis and will not be addressed any further.

Deterministic refers to gradient-based optimization. The gradients of objective functions and constraints with respect to (w.r.t.) design variables, also known as Sensitivity Derivatives (SDs), are used to identify search directions in the design space and drive the optimization. By using a search direction instead of a stochastic strategy, the optimization loop may converge fast to a local optimum. The faster convergence is the main advantage of gradient-based methods over stochastic ones. Using gradient information, constraints are enforced efficiently. One limitation of gradient-based methods involves the computation of globally optimal solutions. Gradients provide only local information about the design space landscape and, as such, optimization

algorithms converge to solutions that are only locally optimal. To find a global optimum, gradient-based optimization loops may need to be performed from multiple starting points. Additionally, gradient-based methods may face difficulties in handling problems with discontinuities. An example for this can be found in [69] in which a constraint function was discontinuous close to points of interest and a blending function had to be used to overcome this problem. This thesis deals exclusively with gradient-based methods.

1.1 Methods for Computing Gradients

The key component of a gradient-based optimization loop is the computation of the gradients of objective functions and constraints. For an optimization software to be of practical of use, these gradients need to be computed with sufficient accuracy and at a low cost. Comprehensive reviews on methods for computing gradients in aerodynamic optimization are given by [89, 123]. This section reviews existing methods for computing gradients, finally leading up to the adjoint method.

The most straightforward way for computing SDs of a function is to use Finite Differences (FDs), wherein the design variables are perturbed using a small step size and the difference quotient yields the derivative. Well-known issues of this method are the truncation errors, caused by a big step size, and the round-off cancellation errors, caused by a smaller-than-needed step size. The step size is, in general, difficult to determine. It usually involves trial-and-error or a convergence study and a different step size might be needed for each design variable. Furthermore, to minimize cancellation errors, the primal equations' residuals need to be reduced by many orders of magnitude, something not always possible in more difficult-to-solve problems. An interesting result, [46], showed that, in more complex cases, two distinct derivative values may be computed with FDs if some conditions, such as the initial solution, are different between the two computations. An alternative, the complex step method, can avoid cancellation errors [6] by computing the FDs after perturbing the design variables in the complex plane. However, it requires serious investment in code development since the CFD code must be rewritten using complex variables. Most importantly, the computational cost of these two methods is proportional to the number of design variables and their use is prohibitive in practical applications.

Another method is Direct Differentiation (DD) with which the primal PDEs are differentiated w.r.t. the design variables and a series of new systems of equations are derived. These are solved for the derivatives of the primal fields w.r.t. the design variables. The objective function SDs are given from an expression that involves the aforementioned derivatives of primal variables. There are two approaches for this method, a continuous and a discrete one, depending on whether the primal equations are differentiated in continuous

form or after their discretization [123]. Irrespective of the approach followed, there is a system of equations that is as costly to solve as the primal problem for each design variable. Although this method avoids the errors of FDs, its cost is also proportional to the number of design variables. It might be preferred in case the objective and constraint functions are, in total, more than the design variables, although this is rarely the case in engineering applications.

In contrast, the adjoint method computes the required gradients at a cost that is (for all practical reasons) independent of the number of design variables [116]. It is essentially a method of Lagrange multipliers. The primal PDEs are the constraints and the Lagrangian function is formulated using these constraints and the adjoint variables which have the role of Lagrange multipliers. Much like in DD, there are two variants of the adjoint method, a continuous and a discrete one. In continuous adjoint, the Lagrangian function is developed using the continuous form of the primal equations' residuals and the adjoint PDEs are first derived and, then, discretized and numerically solved to compute the adjoint variables. In the discrete counterpart, the adjoint equations are directly derived from the Lagrangian developed using the discretized primal residuals. Discrete and continuous adjoint approaches have been compared in [95]. Examples of adjoint methods for aerodynamic optimization can be found in the literature [116, 89].

The gradient is always perpendicular to the iso-surfaces of the objective function in the design space and points in the direction of greatest increase whereas its negative in the direction of greatest decrease. An optimization algorithm uses the gradient to determine a search direction in the design space and take a step along this direction to update the design variables. This step reduces the value of the objective function. A simple method for unconstrained optimization, referred to as steepest descent, takes a step along the direction of the negative gradient. Well known issues of this method are its slow convergence, especially in the vicinity of the optimum. The step length is either user-defined or can be computed using line search methods which, however, require additional evaluations of the objective function. The slow convergence of steepest descent is due to the lack of curvature information in the scheme. To converge faster, Newton methods use the Hessian matrix, which includes curvature information in the form of second-order derivatives of the objective function. However, the computation of the Hessian matrix is rather costly. Methods to compute it, compared to existing ones, have been proposed in [111, 110, 109, 112, 119] and use a combination of adjoint and DD techniques. To avoid these computations, quasi-Newton methods such as the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [12] are employed which rely on approximations of the Hessian matrix instead. In the BFGS, the inverse of the Hessian matrix is approximated using gradient information from previous optimization cycles.

For constrained optimization, these methods are modified to use constraint gradients as well in order to determine a search direction within the feasible design space, allowed by the constraints. This is done by transforming the constrained optimization problem to an unconstrained one and minimize an augmented Lagrangian function defined through the use of Lagrange multipliers, one for each constraint function. Steepest descent methods are combined with a projection technique [133, 134] that uses the gradients of constraints in order to define a search direction that is tangent to the iso-surfaces of the constraint functions. An alternative that has much faster convergence and is more suitable for non-linear optimization problems relies on Sequential Quadratic Programming (SQP) techniques [103]. In the SQP, a quadratic model of the objective function is adopted, the constraints are linearized and a series of quadratic optimization sub-problems are solved using Newton's method. Practical applications of SQP use quasi-Newton approximations of the Hessian of the quadratic model. To handle inequality constraints, the SQP can be combined with an active-set or interior-point strategy. For more information, the reader is referred to [103]. Quasi-Newton methods have faster convergence rate but require the computation of accurate gradients and may fail in case the Hessian matrix is non-singular.

1.2 The Adjoint Method in Shape Optimization

In aerodynamic shape optimization, a continuous adjoint formulation was first devised for Stokes flow [125] and, later on, for inviscid incompressible flows [126]. The applications presented relied on the solution of the potential flow equations. In [53], a continuous adjoint formulations was derived for inviscid compressible flows that was suitable for CFD applications. In a series of publications that followed in later years, computational results using this method were presented. The shape optimization applications ranged from airfoils [128, 55] to wing-body airplane configurations [129] and full aircraft optimization [130, 127] using the compressible Euler equations solved on multi-block structured grids. The extension of the method for the full Navier-Stokes equations followed in [54]. All of these developments spearheaded the adjoint method and led to its widespread use in aerodynamic shape optimization. For incompressible laminar flows, a continuous adjoint formulation was presented by [7] wherein the primal and adjoint equations were solved on unstructured grids using artificial compressibility. The continuous adjoint method was extended to turbulent flows by the PCOpt/NTUA group, for the first time in the literature, in [169] using the Spalart-Allmaras turbulence model. The derivation of adjoint PDEs for more turbulence models followed [170, 120, 118, 65].

In earlier publications, the gradient expression comprised field integrals that, after discretization, contained the derivatives of grid coordinates w.r.t. design

variables, also known as grid sensitivities. These quantify the contribution of grid deformation to the objective function's value. To compute them, FDs are frequently used, the design variables are perturbed and the grid is deformed using the method or algorithm of choice. Thus, the computational cost scales with the number of design variables. This formulation is referred to in this thesis as the Field Integral (FI) adjoint. To overcome this cost, a new formulation based only on surface integrals was developed, first for structured grids [7, 56] and, later, for unstructured ones simultaneously by [57, 108]. This new formulation is referred to as the Surface Integral (SI) adjoint. The FI and SI formulations are mathematically equivalent. Although the new formulation was more efficient, second-order spatial derivatives of flow velocities had to be computed on the boundaries of the computational domain. In practice, these could be computed consistently on structured grids, as was the case in [7], however, on unstructured ones, the formulation was impractical. For this reason, the surface integral that involved grid sensitivities was largely omitted following a series of assumptions. This meant that the influence of grid deformation on the objective function was ignored. Due to this omission, discrepancies between the two formulations could be observed. This problem was more prevalent on coarser grids, as noted by a number of authors [7, 80, 81]. The FI formulation was more accurate than the SI one based on the previous simplifying assumptions. For the case of continuous adjoint, a solution was proposed by the PCOpt/NTUA group in [66] which developed the adjoint to the grid displacement model PDEs. By considering these adjoint equations, grid sensitivities were accounted for, at a cost that did not scale with the number of design variables and the accuracy of surface-based gradient expressions was recovered. The new formulation was referred to as the Enhanced Surface Integral (E-SI) formulation.

Discrete adjoint formulations can be derived with different ways. The earliest formulations followed "hand-differentiation" of the discrete primal residuals. In this way, the equations are adjoint to the discretized primal ones, discretization schemes are taken into account and the computed SDs are expected to be in very good agreement with FDs. In later years, to avoid the laborious task of "hand differentiation", discrete adjoint codes were frequently generated by Automatic Differentiation (AD) tools [43]. Such tools practically apply the chain rule to the series of arithmetic operations performed by the solution algorithm of the primal solver and automatically differentiate it. The summation of all parts of the chain rule yields the SDs. If the chain rule is applied in forward mode, the differentiated code corresponds to a discrete DD whereas the reverse mode corresponds to the discrete adjoint [89]. One technique for AD is *code transformation*, in which the chain rule is automatically differentiated and a new code is generated. Another technique uses *operator overloading*, applicable only to computer programming languages that support object-oriented programming, in which new data structures are used and the operations between them are

overloaded allowing the storage and accumulation of the partial derivatives of the chain rule automatically [100]. Modern AD methods can increase the efficiency through the use of *expression templates* [3]. Frequently, some approximations are made to simplify the development of discrete adjoint and reduce its memory overhead; their effect has been studied in [23]. For instance, it is not uncommon to eliminate (to some extent) terms from the adjoint equations that cause instability in real-world applications. The first discrete adjoint formulation using second-order spatial discretization schemes was developed for the first time by [24, 25], using "hand-differentiation" and was used for 2D and 3D applications. AD was used to develop discrete adjoint solvers for incompressible flows, solved using segregated [14, 135, 48] and coupled [27] pressure-based schemes.

Implementing an efficient and robust iterative scheme for the solution of the adjoint equations is important. In continuous adjoint, due to the similarity of primal and adjoint PDEs, iterative techniques used for solving the former can be used in solving the latter. In discrete adjoint, assuming that the formulation is consistent (i.e. all terms in the discrete primal residuals are differentiated without any simplifications), then, the Jacobian matrix in the adjoint system is the transpose of the Jacobian of the primal equations' residuals. As such, it inherits the eigenvalues of the latter. This is a welcome trait because it suggests that the adjoint iteration will be converging whenever the primal one does. However, a consistent differentiation by itself does not necessarily suffice for the adjoint solver to have the same asymptotic rate of convergence as the primal one [22]. To achieve this duality between the two solvers, the iterative scheme used by the primal solver must also be taken into account and this was first demonstrated in [34]. Duality-preserving solvers were developed by others as well, with either AD [3, 70] or the complex-step method in a Jacobian-free manner [48] or with graph-coloring to form the transposed Jacobian matrix [102].

In continuous adjoint, achieving duality is much more involved. Since the primal PDEs are differentiated in continuous form, the influence of the solution scheme employed to solve them is ignored. For example, the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) [122] family of algorithms is frequently used to solve the incompressible Navier-Stokes equations. The continuity equation is satisfied by deriving and solving a Poisson-type equation for the flow pressure. The derivation of the Poisson equation for pressure is carried out in a discrete sense. In continuous adjoint, it is the momentum and continuity equations that are differentiated and processed and the solution scheme is not taken into account. Despite this, practical experience has shown that even with continuous adjoint, the adjoint solver is much more likely to converge if based on a converged primal solution. On the other hand, divergence of the adjoint solver is a usual occurrence when the iterative scheme of the preceding primal solver stalls and residuals oscillate, such as in cases with flow

unsteadiness [22, 2]. This will become relevant in the following sections.

1.2.1 The Adjoint for Unsteady Flows

Despite its attractive features, in unsteady flows, the adjoint method faces a major setback. The unsteady adjoint equations need to be solved backwards in time and require the instantaneous primal fields to be available at each time-step of the adjoint solver. This noticeably increases storage requirements and/or CPU costs. For this reason, adjoint methods were developed for solution in the frequency domain [97, 98]. This is a viable approach in flows with periodicity or in case the unsteadiness is dominated by a handful of frequencies, however, for purely transient problems, time-domain solutions cannot be avoided.

Early applications used techniques and workarounds to reduce the memory overhead. In time-domain, continuous and discrete adjoints for inviscid flows were used in [96] for shape optimization whereas a discrete adjoint for viscous flows was developed and used in [136]. In the latter, a form of time-coarsening of the primal solution snapshots was used, whereby the storage of every other time-step solution was skipped, and the adjoint equations were solved using a larger time step. The effect of deforming grids from one time instant to the next was included in [84, 85] for fully coupled aeroelastic simulations in periodic flows for 2D applications and, later, extended for 3D in [90]. For these applications, the primal fields were stored on the hard disk and read to memory when it was time to be used by the adjoint solver. For real-world applications, if disk storage is not an option, the use of the check-pointing technique [44, 159] is a popular option. Instead of storing all snapshots, only specific instants are stored at a number of check-points. In order for the adjoint solver to march backwards in time, from one check-point to the next (located earlier in time), the intermediate primal fields are re-computed by integrating the primal PDEs forward in time from the earlier check-point to the current one. In this way, storage requirements are relaxed at the expense of extra CPU cost for the re-computation of the intermediate primal fields; therefore, it is up to the user to choose the number of check-points and control this trade-off. An example of this method can be found in [135] which used AD to construct an adjoint for unsteady incompressible flows with a Large Eddy Simulation code.

Other alternatives include the decomposition of the array of primal-field snapshots. For example, [160, 164] used Proper Orthogonal Decomposition (POD) with Singular Value Decomposition (SVD) or Gram-Schmidt orthogonalization and [18] used Principal Component Analysis to compress the data to be stored. These methods can be seen as reduced-order models for the accurate primal solution time-history. The PCOpt/NTUA has made a number of contributions to the development of such methods. In [153], the CPU cost of the POD was reduced by updating the decomposed matrices as the time-integration progresses with an incremental SVD. In [86], significant memory reductions were

achieved through the synergistic use of the ZFP library for compressing floating-point arrays and the incremental Proper Generalized Decomposition method, both of which were then combined with check-pointing in [87] for increased effectiveness.

1.2.1.1 Convergence Difficulties Due to Flow Unsteadiness

Although all of the aforementioned methods have effectively enabled the use of unsteady-based adjoint for a wider number of applications, in real-world engineering problems, the CPU cost of the optimization with unsteady solvers can still be prohibitively expensive [106]. This is true even by assuming that storage requirements have been overcome without extra CPU cost. For this reason, in cases exhibiting mild unsteadiness, the use of a steady-state solver is often preferred, especially when the flow in the sought optimal solution is expected to be less-time varying or even steady. The same holds for shapes generating vortical flow patterns, that are expected to be suppressed at the end of the optimization. This process is a common practice in industrial shape optimization in which a RANS equations' solver is used to compute quasi-steady solutions to a problem that is known to be unsteady [75], occasionally followed by the re-evaluation of the optimized solutions using an Unsteady RANS (URANS) equations' solver or any other high-fidelity simulation tool [115].

However, the use of steady-state solvers in cases with mild unsteadiness (or, even, cases with vortex shedding), often encountered in flows past bluff bodies, usually leads to significant convergence difficulties for both the primal and adjoint solvers. The steady-state flow solver may not converge and the adjoint solution is led to stalling or divergence [22, 2] as a consequence of the former. This dependence of the adjoint solver on the convergence of the primal one was discussed in the previous section as well. These implications are also reported in [74] which, in addition, demonstrated that the objective function value computed by averaging the results of an unsteady simulation might be far from that of a steady solution. The same holds for the adjoint-based sensitivities. Nevertheless, averaging the solutions of a steady solver over the last few iterations, using a steady adjoint on that field to compute the SDs, and, then, using these SDs in the optimization might result to a better value of the objective function that can be verified by re-evaluating it with an unsteady flow solver. This is a choice made purely with computational cost considerations in mind, at the user's responsibility.

Regarding continuous adjoint, in specific, convergence difficulties often manifest themselves through or are attributed to the so-called Adjoint Transposed-Convection (ATC) term [106] (see section 2.4.1), emerging in the adjoint momentum equations. For the latter, the most frequently used remedy is the damping of the ATC close to the wall, either totally or selectively, on a cell-by-cell basis, using an appropriate sensor [106, 63]. This can be seen as a

trade-off between accuracy and robustness of the adjoint method.

Convergence difficulties of the adjoint equations render the optimization procedure ineffective or, in some cases, even impossible. In this thesis, the Recursive Projection Method (RPM) [143], a stabilization method for iterative procedures, is used to deal with the aforementioned convergence difficulties. The RPM splits the solution space in two subspaces which contain the unstable and stable modes of the Jacobian matrix of the iterative scheme, the one being the orthogonal complement of the other. It, then, performs an additional Newton step within the unstable subspace while retaining the original iterative scheme on its complement. A precursor to the RPM was originally developed in [58], requiring though the Jacobian of the Fixed-point Iteration (FPI) scheme to be symmetric. Beyond that, in [83], a number of RPM variants were derived and a thorough analysis of the algorithm and its convergence properties was provided.

In the literature, the application of the RPM was successfully extended to the acceleration and stabilization of flow and adjoint solvers. In [42], it was used to accelerate a dual time-stepping Navier-Stokes code for compressible flows by improving the convergence of the pseudo-time iterations. In [59], the RPM contributed to the acceleration of a coupled pressure-based multigrid solver for the steady flow equations of incompressible fluids. It was also used for the stabilization of linearized compressible flow solvers [13] as well as the discrete adjoint to the Navier-Stokes equations for compressible fluids [22, 2].

The proper treatment of Differential-Algebraic Equations (DAEs) with the RPM was studied in the theses of [79, 154]. The algebraic constraints were firstly transformed to Ordinary Differential Equations before the RPM could be applied to find steady solutions of the system. An alternative approach for handling DAEs with the RPM was shown in [73] in which the constraints were treated as "very fast evolving" equations. The algebraic variables corresponding to these constraints could, thus, be treated by the RPM directly, without additional considerations. This is the approach followed in this thesis for these types of variables, such as the fluid volume fluxes passing through the faces of the finite volumes used by the SIMPLE algorithm.

An implementation of the RPM was first developed by the PCOpt/NTUA for the adjoint solver of the *adjointOptimisation* library in a previous PhD thesis [64]. It was used only to assist the convergence of adjoint solvers in academic cases and in the shape optimization of a ground vehicle. In all these cases, adjoint solvers used "pseudo-steady" primal solutions, averaged over the last iterations of the stalled primal solver. This thesis goes one step further by making use of both stabilized steady-state primal and adjoint solvers. Also, the algorithm was slightly modified to better suit the SIMPLE solution scheme and improvements were made to the algorithm that computes the unstable subspace basis. More information on this in section 3.1.

The RPM is well suited to problems such as those this thesis is dealing with

since it can handle instabilities due to flow unsteadiness. Furthermore, its implementation is non-intrusive in nature, treating the underlying iterator as a black box and leaving the outcome unaffected [143]. In view of the above, the RPM provides a means to potentially stabilize the adjoint equations without affecting the SDs. Using the RPM, an investigation is carried out regarding the use of the two techniques mentioned, the use of flow-field averaging and ATC damping, which overcome convergence issues by introducing assumptions or by tampering with the equations themselves.

1.3 Aerodynamic Shape Optimization Under Uncertainties

In practical applications, there is a degree of uncertainty related to the boundary conditions which might be beyond the designer's control. As a result, it might happen that whenever the optimization is performed for fixed boundary conditions, the optimized shapes might not perform well when these conditions change [69, 21]. Robust Design Optimization (RDO) methods solve optimization problems while accounting for uncertainties [11, 140].

A key part of RDO is the Uncertainty Quantification (UQ) of the model output (a.k.a. the Quantity of Interest (QoI)). In the absence of uncertainties, the QoI would act as the objective function. Following a probabilistic approach, uncertainties are incorporated into the model by associating a Probability Density Function (PDF) to each uncertain variable. The uncertainties are, then, propagated through the model computing the QoI. Therefore, in the presence of uncertainties, the QoI is treated as a statistical quantity and the optimization objective becomes a function of its statistical moments. Usually, the first two statistical moments, the mean value and standard deviation, are of interest. The objective function is frequently cast in the form of the weighted sum of these two statistical moments, see [119] for an example. Various UQ methods have been developed, including, though not limited to, the computationally exhaustive Monte Carlo (MC) method [26], the Method of Moments (MoM) [158, 47] and intrusive or non-intrusive Polynomial Chaos Expansion (PCE) techniques [161, 163]. More information can be found in [158], for a more general description of uncertainty analysis methods in CFD with applications. A recent review of RDO in aerodynamic applications can be found in [52]. An overview of UQ with PCE in CFD is given in [99] and more details on sampling techniques for non-intrusive PCE with least-squares in [45].

With the MC, the software that computes the QoI is treated as a "black-box". A great number of value-sets of uncertain variables are chosen, the MC replicates, and the QoI is sampled for these value-sets. The statistical moments are computed based on the results of this sampling. The MC is easy to implement and quantifies the effect of the joint PDF of the uncertain variables directly. Different uncertain variables may be governed by different types of PDFs but the MC may handle all

of them. The drawback is that, usually, a great number of QoI evaluations, each one corresponding to a call to the CFD solver, might be necessary for the accurate computation of the statistical moments. For this reason, alternative methods are sought that are more efficient.

A usual alternative which follows a more intrusive approach is to linearize the QoI about the mean values of the uncertain variables using a Taylor series expansion up to a certain order. The MoM is, then, used on the linearized quantity. Depending on the order of terms retained in the Taylor expansion, a variety of methods are derived, the most common among them are the First-Order Second-Moment (FOSM) and Second-Order Second-Moment (SOSM) methods. The latter includes second-order terms and offers higher accuracy; however, it requires more computations and is less straightforward to implement. Both involve the computation of derivatives of the QoI w.r.t. uncertain variables. If FOSM is combined with the adjoint method, the required gradients can be computed at minimal cost.

A popular approach to UQ is the PCE method. It relies on polynomial expansions to model the propagation of uncertainty from an input to the QoI. Its main advantage is the capability to model uncertainties up to an arbitrary order. It was introduced in [161] for Gaussian processes, using Hermite polynomials as basis functions, and generalized for non-Gaussian processes in [163]. The polynomial basis comprises different polynomials for each type of PDF which are chosen to be orthogonal to each other. Lately, PCE methods that handle arbitrary probability distributions without being restricted to parametric ones have been developed [104]. They are data-driven and can construct a polynomial basis on the fly, allowing the adaptation of the PCE to arbitrary distributions. UQ problems that are closer to real-world applications can be tackled in this way.

PCE methods can be either intrusive or non-intrusive. In intrusive PCE (iPCE), the governing PDEs are projected on an orthogonal polynomial basis and are, then, solved to compute PCE coefficients for the flow variables. In non-intrusive PCE, the QoI is expressed as a polynomial expansion of the uncertain variables and least-squares regression or quadrature rules are used to compute the polynomial coefficients; applications in fluid mechanics can be found in [51, 50]. Regression-based PCE may profit from the adjoint method which provides the derivatives of the QoI w.r.t. all the uncertain variables, at the cost of 1 Equivalent Flow Solution (EFS)[1], at each collocation point and, thus, additional equations to be satisfied in the regression system [132, 4, 32]. In this way, less collocation points over the uncertain space are needed and a great amount of local information per point fills this gap; the gain is a reduction in the number of EFS needed in UQ, compared to standard PCE methods.

In the literature, examples of RDO in aerodynamics can be found in recent

[1] EFS is the computational cost unit and 1 EFS corresponds to the solution of the flow equations.

years. In [167], a non-intrusive PCE (niPCE) method was combined with Particle Swarm Optimization to optimize the shape of an NLF airfoil by accounting for uncertainties in the Mach number. Sudden increases in drag, observed with fluctuations in the Mach number that were encountered in the shape optimized without accounting for uncertainties, were minimized. In [141], a series of NACA 4-digit airfoils were optimized using niPCE under geometric and farfield uncertainties. Adjoint methods have also been used with UQ in RDO. An example can be found in the work of [139] which used niPCE with the Karhunen–Loève expansion to model geometric uncertainties. In [142], niPCE was combined with an intrusive approach applied only for the quantities required by the adjoint to reduce the computational cost of the RDO.

Various contributions have been made by the PCOpt/NTUA to RDO based on PCE and/or the MoM. For PCE, in [113], the shape of two airfoils was optimized in the presence of uncertainties related to the flow conditions using an iPCE of the primal equations and the adjoint method. The continuous adjoint was formulated with two different ways, by developing either the adjoint to the iPCE of the primal PDEs or the iPCE to the adjoint PDEs without uncertainties. In [36], iPCE methods were used for shape optimization in Conjugate Heat Trasfer applications under environmental and manufacturing uncertainties. With respect to the MoM, in [119], a combination of DD and adjoint methods were used to perform shape optimization under uncertainties related to the boundary conditions using the SOSM method. To do so, third-order derivatives of the QoI w.r.t. to design and uncertain variables had to be computed. The combination of DD and adjoint has proved to be the most efficient way of computing them, thus far. In a recent paper by the PCOpt/NTUA, the projected FOSM (pFOSM) method [30], derived from the FOSM method, for solving optimization problems in fluid mechanics under uncertainties was presented, for laminar flows. In gradient-based optimization based on pFOSM, mixed derivatives (w.r.t. design and uncertain variables) of the QoI are involved. The pFOSM method profits from the computation of matrix (of these mixed derivatives) -vector products at the cost of 2 EFS only. The key point of pFOSM is that its cost per optimization cycle is no more than 4 EFS, irrespective of the number of uncertain variables. This cost corresponds to the solution of the primal and adjoint systems of equations (2 EFS) plus the solution of two systems that result from the so-called DD of the above equations w.r.t. the uncertain variables and their subsequent projection onto an appropriate vector (2 more EFS). A realistic assumption is made that the cost of solving either the adjoint or one system resulting from the DD of the primal or adjoint equations is 1 EFS.

The goal of this thesis is two-fold. The first goal is to develop the pFOSM method for turbulent flows, by extending the work done in [30]. The second and most important goal is to present a novel method based on the Adjoint-assisted Polynomial Chaos Expansion (APCE) in RDO exhibiting low computational cost,

for laminar and turbulent flows. In an APCE-based optimization, the objective function gradient involves mixed derivatives, since a quantity that includes derivatives of the QoI w.r.t. the uncertain variables must be differentiated w.r.t. the design variables. If computed, these mixed derivatives increase the cost of an RDO cycle based on APCE significantly, thus reducing the cost benefits of APCE in UQ. To this end, an RDO method is set-up herein, aptly named projected APCE (pAPCE), that makes use of the projected matrix of mixed derivatives of a QoI onto appropriate vectors, similarly to pFOSM, in order to minimize the weighted sum of its first two statistical moments. The advantages of pAPCE are that (a) it is based on a high-fidelity UQ method, (b) it uses a reduced number of EFS to compute the objective function in the presence of uncertainties, compared to standard regression-based PCE, and (c) it computes the gradient of this objective function w.r.t. the design variables by just doubling the cost of the UQ itself.

1.4 Adjoint Methods developed by the PCOpt/NTUA

The current work expands upon existing adjoint-based optimization methods developed by the PCOpt/NTUA throughout a number of PhD theses. Adjoint methods were developed either within the OpenFOAM [105] environment or in the in-house GPU-enabled PUMA CFD solver. Some of these methods are made publicly available in the *adjointOptimisation* library of OpenFOAM, via OpenCFD.

First developments were made in the thesis of Papadimitriou [107]. A continuous adjoint formulation was derived that was valid for any type of grid, for inviscid and viscous flows [108]. The sensitivity computations involved only boundary integrals for objective functions defined as surface or field integrals. The method was used for optimization of thermal turbomachinery cascades and external aerodynamic applications. Additionally, a discrete adjoint method was used to perform a posterriori analysis of objective functionals and perform grid adaptation to achieve pre-defined accuracy. Finally, gradients computed by the adjoint method were used, in addition to objective function values, to train an artificial neural network to be used as a surrogate evaluation model in an EA-based optimization.

The following theses developed continuous adjoint methods within the OpenFOAM environment. In Zymaris' thesis [168], continuous adjoint formulations were developed for flows solved using turbulence models for the first time in the literature. The development of adjoint PDEs involved the differentiation of this turbulence models w.r.t. design variables. This was done for the Spalart-Allmaras one-equation model in OpenFOAM and the applications involved cases in which the boundary layer was fully resolved to the wall [169]. On coarser grids, the use of wall functions introduces some challenges in the derivation of boundary conditions (BCs) with the continuous adjoint but these

were overcome through the use of the so-called adjoint wall function [170], a concept developed for the $k-\epsilon$ turbulence model. The frequently used frozen-turbulence assumption was scrutinized and its detrimental effects were highlighted. The developed methods were used in flow control optimization problems [171].

In Papoutsis-Kiachagias' thesis [114], the adjoint wall functions technique was extended to the Spalart-Allmaras model [118]. Additionally, the continuous adjoint to the Launder-Sharma low-Re $k-\epsilon$ turbulence model was derived [120]. Additionally, the adjoint to the Hamilton-Jacobi equation was developed to account for variations of the field of distances from nearest walls, used by the Spalart-Allmaras model. A continuous adjoint method was further developed for the RANS in the rotating reference frame for the design of hydraulic turbomachines [117]. Finally, combinations of DD and adjoint methods were used to compute third-order mixed derivatives w.r.t. design and environmental variables for shape optimization under uncertainties with the SOSM method [119]. Applications involved optimal flow control, topology optimization and, lastly, shape optimization of real car geometries.

In the PhD thesis of Kavvadias [64], the continuous adjoint to the $k-\omega$ SST turbulence model, with or without wall functions, was developed [65]. Additionally, the Grid Displacement Model (GDM) PDEs were considered in the Lagrangian formulation and adjoint counterparts were derived. It was assumed that grid displacement obey the Laplace equation. With this, a new sensitivities expression was derived that included only surface integrals without loss in accuracy and was a step-up from previous ones [66]. The new formulation accounted for the influence of grid deformation on the objective function without resorting to excessive computations of grid sensitivities throughout the whole domain. Additionally, the adjoint to the Multiple Reference Frame was developed for optimization of flows with rotor-stator interactions and a continuous adjoint for the URANS equations which made use of the check-pointing technique. Finally, the RPM was programmed and was used to stabilize an adjoint solver for shape optimization of a passenger car geometry.

In Gkaragkounis' thesis, the E-SI adjoint was further investigated by using different GDMs although a Laplace equation was used to develop the adjoint to the GDM PDEs. It was shown that this assumption can indeed be used without any serious harm to the computed SDs for a great number of GDMs [39]. A continuous adjoint method was developed for conjugate heat transfer problems [37]. Previous findings regarding the proper treatment of grid sensitivities were re-confirmed in these problems. A new adjoint-assisted technique for Pareto front tracing for multi-objective optimization problems was developed [38]. Lastly, shape optimization in the presence of environmental and manufacturing uncertainties was performed using an intrusive PCE method.

In the following theses, the developed adjoint methods were implemented in

PUMA. In Asouti's thesis [9], both the discrete and continuous adjoint were developed for low-Mach flows solved by preconditioning the compressible Navier-Stokes equations [10]. In Zervogiannis' thesis [165], a "hand-differentiated" discrete adjoint solver was developed for shape optimization in internal and external aerodynamics involving turbulent flows. Focus was laid on hybrid grids and methods for a posteriori analysis were improved to cover hybrid grids as well. Additionally, the method for computing first- and second-order derivatives of the objective function, presented in a series of publications by the PCOpt/NTUA [111, 110, 109, 112], was extended to turbulent flows and was combined with a quasi-Newton strategy for higher efficiency [166]. In Kontoleontos' thesis [71], the continuous adjoint method for incompressible flows was extended to heat transfer problems and the Spalart-Allmaras adjoint approach was developed for topology optimization problems [72]. In Tsiakas' thesis [149], the continuous adjoint method was developed for compressible flows. The formulation was based on both surface and field integrals and enhanced parameterization techniques were developed for turbomachines and transonic wings. In the thesis of Monfaredi [92], a continuous adjoint method for minimizing sound radiation modelled with the Ffwocs-Williams and Hawkings acoustic analogy was developed [93]. Aeroacoustic shape optimization was performed for airfoils, achieving omni–directional noise reduction, as well as for the geometry of an aero–engine intake. Additionally, adjoint methods based on the cut-cell technique were developed in the PhD thesis of Samouchos [138]. The method was developed on an in-house cut-cell CFD solver. It was extended in the thesis of Vrionis [157] for multiphase flows [155] and topology optimization [156].

1.5 Thesis Objectives and Outline

The thesis is developed along two main axes, expanding upon previous work on the continuous adjoint method; the first one is dealing with stability issues of flow and adjoint solvers caused by (usually mild) flow unsteadiness and the second one with the development of RDO techniques. The reduction in the computational cost of shape optimization, either with or without uncertainties, including problems with mild unsteadiness, is the main goal of the developed techniques. All methods were programmed in the OpenFOAM CFD toolbox.

In Chapter 2, the continuous adjoint method is presented for shape optimization governed by the RANS equations for turbulent flows modelled with the Spalart-Allmaras. This forms the basis for all the methods developed in the following chapters.

Chapter 3 includes a presentation of the RPM and the SIMPLE algorithm for the solution of incompressible flows. Some peculiarities of SIMPLE in relation to the RPM are discussed. Apart from tackling the stabilization problem itself,

the common practice of using steady solvers in flows that exhibit some level of unsteadiness, the feasibility of this approach and its cost-benefits are discussed in this chapter. The effects of flow-field averaging and ATC damping on the SDs are additionally evaluated. The proposed technique is demonstrated in 2D shape optimization problems. To help draw conclusions, the cases presented involve vortex shedding flows.

Chapter 4 presents the pFOSM and the newly developed pAPCE methods for shape optimization in the presence of uncertainties. Each of them can be seen as an extension of a UQ technique (FOSM or APCE) which are already in use and account for objective functions combining statistical moments of the QoI. Both are extended for turbulent flows with the inclusion of the Spalart-Allmaras model in the framework. A key point of the RDO framework presented in this thesis is the ability to compute the gradient of the RDO objective function at a cost that is almost equal to the cost of UQ. RDO problems for external and internal aerodynamics are solved using both methods.

The newly developed methods and proposed techniques are validated in academic cases and, then, used to solve a number of industrial applications. Chapter 5 demonstrates both the RPM and the RDO framework in industrial applications. These include aerodynamic shape optimization of ground vehicles using steady solvers and the RPM and of an aircraft wing, in the presence of farfield uncertainties, using pFOSM and pAPCE.

Chapter 2

The Continuous Adjoint Method

In this chapter, the continuous adjoint method for problems governed by the incompressible Reynolds-Averaged Navier-Stokes (RANS) equations is presented. The one-equation Spalart-Allmaras turbulence model [145] is used to effect closure in turbulent flows. The geometry and grid parameterization techniques, used in this thesis are presented in this chapter.

2.1 Flow Equations and the Primal Problem

The steady-state RANS equations for an incompressible fluid flow, with the Spalart-Allmaras model equation and a PDE for computing distances, are

$$\mathcal{R}^p = -\frac{\partial v_i}{\partial x_i} = 0 \tag{2.1a}$$

$$\mathcal{R}^v_i = v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} + \frac{\partial p}{\partial x_i} = 0 \ , \qquad i = 1, 2(, 3) \tag{2.1b}$$

$$\mathcal{R}^{\widetilde{\nu}} = v_i \frac{\partial \widetilde{\nu}}{\partial x_i} - \frac{\partial}{\partial x_i}\left[\frac{\nu + \widetilde{\nu}}{c_\sigma}\frac{\partial \widetilde{\nu}}{\partial x_i}\right] - \frac{c_{b_2}}{c_\sigma}\left(\frac{\partial \widetilde{\nu}}{\partial x_i}\right)^2 + \widetilde{\nu}\,(D - P) = 0 \tag{2.1c}$$

$$\mathcal{R}^\Delta = \frac{\partial}{\partial x_i}\left(\frac{\partial \Delta}{\partial x_i}\Delta\right) - \Delta\frac{\partial^2 \Delta}{\partial x_i^2} - 1 = 0 \tag{2.1d}$$

where v_i are the fluid velocity components, p the pressure divided by the fluid density, $\tau_{ij} = (\nu + \nu_t)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)$ the stresses and ν and ν_t are the bulk and eddy kinematic viscosity of the fluid, respectively. Equations 2.1a and 2.1b are the continuity and momentum equations, respectively, whereas eq. 2.1c is the Spalart-Allmaras turbulence model PDE which is solved for the model variable $\widetilde{\nu}$. This turbulence model makes use of the field of distances of cell-centers from the nearest solid walls, Δ, computed by solving the Hamilton-Jacobi equation [150], eq. 2.1d. P and D in eq. 2.1c are the turbulence production and destruction

terms, respectively, which are given by

$$
\begin{aligned}
&P=c_{b_1}\widetilde{S}\ ,\ D=\frac{c_{w_1}f_w\widetilde{\nu}}{\Delta^2}\ ,\ \widetilde{S}=\omega+\frac{f_{v_2}\widetilde{\nu}}{\kappa^2\Delta^2}\ ,\ f_{v_2}=1-\frac{\chi}{1+\chi f_{v_1}}\ ,\ f_{v_1}=\frac{\chi^3}{\chi^3+c_{v_1}^3}\\
&\chi=\frac{\widetilde{\nu}}{\nu}\ ,\ \omega=\|\boldsymbol{Y}\|\ ,\ Y_i=\epsilon_{ijk}\frac{\partial v_k}{\partial x_j}\ ,\ c_{w_1}=\frac{c_{b_1}}{\kappa^2}+\frac{1+c_{b2}}{c_\sigma}\\
&f_w=g\left(\frac{1+c_{w_3}^6}{g^6+c_{w_3}^6}\right)^{\frac{1}{6}}\ ,\ g=d+c_{w_2}(d^6-d)\ ,\ d=\frac{\widetilde{\nu}}{\widetilde{S}\kappa^2\Delta^2}
\end{aligned}
\tag{2.2}
$$

The eddy viscosity is computed by $\nu_t=\widetilde{\nu}f_{v_1}$. Other constants used are $c_\sigma=0.66666$, $\kappa=0.41$, $c_{b_1}=0.1355$, $c_{b2}=0.622$, $c_{w_2}=0.3$, $c_{w_3}=2$, $c_{v_1}=7.1$ and ϵ_{ijk} stands for the Levi-Civita permutation symbol. Equations 2.1a to 2.1d together constitute the primal equations.

In all problems to follow, the computational domain Ω is bounded by a possible combination of the following types of boundaries: inlet (S_I), outlet (S_O), solid walls (S_W) and symmetry planes (S_S), along which the following boundary conditions (BCs)

$$
\begin{aligned}
&S_I:\boldsymbol{v}=\boldsymbol{v}_I\ ,\ \frac{\partial p}{\partial n}=0\ ,\ \widetilde{\nu}=\widetilde{\nu}_I\ ,\ \frac{\partial\Delta}{\partial n}=1\\
&S_O:\frac{\partial v_i}{\partial n}=0\ ,\ p=0\ ,\ \frac{\partial\widetilde{\nu}}{\partial n}=0\ ,\ \frac{\partial\Delta}{\partial n}=1\\
&S_W:v_i=0\ ,\ \frac{\partial p}{\partial n}=0\ ,\ \widetilde{\nu}=0\ ,\ \Delta=0\\
&S_S:v_{\langle n\rangle}=0\ ,\ \frac{\partial v_{\langle t^l\rangle}}{\partial n}=0\text{ for }l=I,II\ ,\ \frac{\partial p}{\partial n}=0\ ,\ \frac{\partial\widetilde{\nu}}{\partial n}=0\ ,\ \frac{\partial\Delta}{\partial n}=0
\end{aligned}
\tag{2.3}
$$

are imposed. In Equation 2.3, $\partial()/\partial n$ is the normal derivative, in the direction of the surface normal unit vector $\boldsymbol{n}$ pointing outwards and $\boldsymbol{t}^I$ and $\boldsymbol{t}^{II}$ are two unit vectors, parallel to S. The first one, $\boldsymbol{t}^I$, is chosen arbitrarily and the second results from $\boldsymbol{t}^{II}=\boldsymbol{n}\times\boldsymbol{t}^I$. Also, $v_{\langle n\rangle}=v_in_i$ and $v_{\langle t^l\rangle}=v_it_i^l$.

2.2 Mathematical Background

The adjoint method computes the gradient of an objective function F w.r.t. a set of design variables b_n, $n\in[1,N]$, at a cost that does not scale with N. For the presentation of the continuous adjoint formulation, a distinction must be made between operators $\delta(.)/\delta b_n$ and $\partial(.)/\partial b_n$. For an arbitrary flow quantity $\Phi=\Phi(\boldsymbol{b},\boldsymbol{x}(\boldsymbol{b}))$, $\delta\Phi/\delta b_n$ represents the total derivative of Φ w.r.t. to $\boldsymbol{b}$ and stands for the total rate of the variation in Φ due to a variation in b_n. This variation in Φ is caused by the deformation of the domain boundary, in turn due to δb_n, which affects both the flow itself and the location which Φ corresponds to. On the other

hand, $\partial\Phi/\partial b_n$ is the partial derivative of Φ w.r.t. $\boldsymbol{b}$ and measures only the local rate of change of Φ due to a variation in b_n; it accounts only for changes in the flow caused by the boundary deformation, at a stationary point within the flow. These two derivatives are connected through

$$\frac{\delta\Phi}{\delta b_n}=\frac{\partial\Phi}{\partial b_n}+\frac{\partial\Phi}{\partial x_k}\frac{\delta x_k}{\delta b_n} \tag{2.4}$$

which is similar to the Lagrangian versus Eulerian description of derivatives of flow quantities w.r.t. time. From the discrete point of view, $\partial\Phi/\partial b_n$ does not account for the displacement of internal grid nodes, δx_k, caused by the boundary deformation. Partial derivatives w.r.t. $\boldsymbol{b}$ and spatial gradients of Φ permute

$$\frac{\partial}{\partial b_n}\left(\frac{\partial\Phi}{\partial x_j}\right)=\frac{\partial}{\partial x_j}\left(\frac{\partial\Phi}{\partial b_n}\right) \tag{2.5}$$

which is not the case for the total derivatives. The total derivatives of the spatial derivatives of Φ, after considering eqs. 2.4 and 2.5, are given by [108]

$$\frac{\delta}{\delta b_n}\left(\frac{\partial\Phi}{\partial x_j}\right)=\frac{\partial}{\partial x_j}\left(\frac{\delta\Phi}{\delta b_n}\right)-\frac{\partial\Phi}{\partial x_k}\frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right) \tag{2.6}$$

2.3 Definition and Differentiation of the Objective Function

Throughout this work, the objective function F is a surface integral over different boundaries. A generic expression is used

$$F=\int_S F_{S,i}n_i dS \tag{2.7}$$

Equation 2.7 is differentiated w.r.t. $\boldsymbol{b}$ and since the integrand $F_{S,i}$ in eq. 2.7 is considered to be a function of all primal variables, the chain rule is used leading to

$$\begin{aligned}\frac{\delta}{\delta b_n}\int_S F_{S,i}n_i dS=&\int_S\left(\frac{\partial F_{S,i}}{\partial v_k}\frac{\delta v_k}{\delta b_n}+\frac{\partial F_{S,i}}{\partial p}\frac{\delta p}{\delta b_n}+\frac{\partial F_{S,i}}{\partial \tau_{kj}}\frac{\delta \tau_{kj}}{\delta b_n}+\frac{\partial F_{S,i}}{\partial \widetilde{\nu}}\frac{\delta \widetilde{\nu}}{\delta b_n}+\frac{\partial F_{S,i}}{\partial \Delta}\frac{\delta \Delta}{\delta b_n}\right)n_i dS\\&+\int_S\frac{\partial F_{S,i}}{\partial x_k}n_i\frac{\delta x_k}{\delta b_n}dS+\int_S F_{S,i}\frac{\delta(n_i dS)}{\delta b_n}\end{aligned} \tag{2.8}$$

Stresses are considered as extra primal variables so as to allow the derivation of boundary conditions for the adjoint system of equations later on. The last surface integral expresses the contribution from the derivatives of the area/length of the boundary cell faces' surfaces to $\delta F/\delta b_n$. In eq. 2.8, the derivatives of the flow variables w.r.t. $\boldsymbol{b}$ are required to compute all integrals. The cost of computing them for all b_n is proportional to N and this is overcome by the adjoint method.

2.4 Lagrangian Function and its Differentiation

Starting point for the adjoint formulation is the definition of the Lagrangian function,

$$\mathcal{L}=F+\int_{\Omega}\left(u_i\mathcal{R}_i^{\boldsymbol{v}}+q\mathcal{R}^p+\widetilde{\nu}_\alpha\mathcal{R}^{\widetilde{\nu}}+\Delta_\alpha\mathcal{R}^{\Delta}\right)d\Omega \tag{2.9}$$

using the residuals of eqs. 2.1a to 2.1d and a set of Lagrange multipliers or adjoint variables' fields. The latter are u_i, q, $\widetilde{\nu}_\alpha$ and Δ_α which are the adjoint to v_i, p, $\widetilde{\nu}$ and Δ, respectively. The primal equations' residuals act as constraints since they must be satisfied in all optimization cycles. Because of this, the gradient of $\mathcal{L}$ is equal to that of F. Equation 2.9 is differentiated w.r.t. $\boldsymbol{b}$, yielding

$$\frac{\delta\mathcal{L}}{\delta b_n}=\frac{\delta F}{\delta b_n}+\int_{\Omega}\left(u_i\frac{\delta\mathcal{R}_i^{\boldsymbol{v}}}{\delta b_n}+q\frac{\delta\mathcal{R}^p}{\delta b_n}+\widetilde{\nu}_\alpha\frac{\delta\mathcal{R}^{\widetilde{\nu}}}{\delta b_n}+\Delta_\alpha\frac{\delta\mathcal{R}^{\Delta}}{\delta b_n}\right)d\Omega \tag{2.10}$$

and requires the differentiation of the primal equations' residuals.

2.4.1 Differentiation of the Mean-Flow Equations

For the differentiation of the following terms, within volume integrals that contain spatial gradients of primal variables, eq. 2.6 is used in addition to the Green-Gauss theorem. The differentiation of the continuity equation, eq. 2.1a, follows in detail and the rest of the terms are only briefly presented:

$$\begin{aligned}\int_{\Omega}q\frac{\delta\mathcal{R}^p}{\delta b_n}d\Omega&=-\int_{\Omega}q\frac{\delta}{\delta b_n}\left(\frac{\partial v_i}{\partial x_i}\right)d\Omega\\&=-\int_{\Omega}q\frac{\partial}{\partial x_i}\left(\frac{\delta v_i}{\delta b_n}\right)d\Omega+\int_{\Omega}q\frac{\partial v_i}{\partial x_k}\frac{\partial}{\partial x_i}\left(\frac{\delta x_k}{\delta b_n}\right)d\Omega\\&=-\int_{\Omega}\frac{\partial}{\partial x_i}\left(q\frac{\delta v_i}{\delta b_n}\right)d\Omega+\int_{\Omega}\frac{\partial q}{\partial x_i}\frac{\delta v_i}{\delta b_n}d\Omega+\int_{\Omega}q\frac{\partial v_i}{\partial x_k}\frac{\partial}{\partial x_i}\left(\frac{\delta x_k}{\delta b_n}\right)d\Omega\\&=-\int_{S}qn_i\frac{\delta v_i}{\delta b_n}dS+\int_{\Omega}\frac{\partial q}{\partial x_i}\frac{\delta v_i}{\delta b_n}d\Omega+\int_{\Omega}q\frac{\partial v_j}{\partial x_k}\frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right)d\Omega\end{aligned} \tag{2.11}$$

In eq. 2.11, the last term on the right-hand-side (r.h.s.) involves (in the discrete sense) the spatial gradient of grid sensitivities $(\delta x_k/\delta b_n)$. The differentiation of

eq. 2.1b yields

$$\begin{aligned}\int_\Omega u_i \frac{\delta \mathcal{R}_i^{\boldsymbol{v}}}{\delta b_n} d\Omega = & \int_\Omega \left[u_j \frac{\partial v_j}{\partial x_i} - v_j \frac{\partial u_i}{\partial x_j} - \frac{\partial \tau_{ij}^\alpha}{\partial x_j} \right] \frac{\delta v_i}{\delta b_n} d\Omega - \int_\Omega \frac{\partial u_i}{\partial x_i} \frac{\delta p}{\delta b_n} d\Omega \\ & + \int_\Omega f_{v_1} \frac{\partial u_i}{\partial x_j} \left[\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right] \frac{\delta \widetilde{\nu}}{\delta b_n} d\Omega \\ & + \int_S \left[u_i v_j n_j + \tau_{ij}^\alpha n_j \right] \frac{\delta v_i}{\delta b_n} dS - \int_S u_i n_j \frac{\delta \tau_{ij}}{\delta b_n} dS + \int_S u_i n_i \frac{\delta p}{\delta b_n} dS \\ & + \int_\Omega \left[-u_i v_j \frac{\partial v_i}{\partial x_k} + u_i \frac{\partial \tau_{ij}}{\partial x_k} - u_j \frac{\partial p}{\partial x_k} - \tau_{ij}^\alpha \frac{\partial v_i}{\partial x_k} \right] \frac{\partial}{\partial x_j} \left(\frac{\delta x_k}{\delta b_n} \right) d\Omega \end{aligned} \tag{2.12}$$

where $\tau_{ij}^\alpha = (\nu + \nu_t) \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right)$ are the adjoint stresses. The third integral on the r.h.s. of eq. 2.12 contributes to the adjoint to the Spalart-Allmaras model equation.

2.4.2 Differentiation of the Turbulence Model and the Hamilton-Jacobi Equation

The differentiation of the Spalart-Allmaras equation's residual yields

$$\begin{aligned}\int_\Omega \widetilde{\nu}_\alpha \frac{\delta \mathcal{R}^{\widetilde{\nu}}}{\delta b_n} d\Omega = & \int_\Omega \Bigg[- v_i \frac{\partial \widetilde{\nu}_\alpha}{\partial x_i} - \frac{\partial}{\partial x_i} \left(\frac{\nu + \widetilde{\nu}}{c_\sigma} \frac{\partial \widetilde{\nu}_\alpha}{\partial x_i} \right) + \frac{1}{c_\sigma} \frac{\partial \widetilde{\nu}_\alpha}{\partial x_i} \frac{\partial \widetilde{\nu}}{\partial x_i} \\ & + \frac{2 c_{b2}}{c_\sigma} \frac{\partial}{\partial x_i} \left(\widetilde{\nu}_\alpha \frac{\partial \widetilde{\nu}}{\partial x_i} \right) + \widetilde{\nu}_\alpha \widetilde{\nu} C_{\widetilde{\nu}} + \widetilde{\nu}_\alpha (D - P) \Bigg] \frac{\delta \widetilde{\nu}}{\delta b_n} d\Omega \\ & + \int_\Omega \left[\widetilde{\nu}_\alpha \frac{\partial \widetilde{\nu}}{\partial x_i} - \frac{\partial}{\partial x_l} \left(\widetilde{\nu}_\alpha \widetilde{\nu} \frac{C_\omega}{\omega} Y_j \epsilon_{jli} \right) \right] \frac{\delta v_i}{\delta b_n} d\Omega + \int_\Omega \widetilde{\nu}_\alpha \widetilde{\nu} C_\Delta \frac{\delta \Delta}{\delta b_n} d\Omega \\ & + \int_S \left[\widetilde{\nu}_\alpha v_i + \frac{\nu + \widetilde{\nu}}{c_\sigma} \frac{\partial \widetilde{\nu}_\alpha}{\partial x_i} - \frac{\widetilde{\nu}_\alpha}{c_\sigma} \frac{\partial \widetilde{\nu}}{\partial x_i} - \widetilde{\nu}_\alpha \frac{2 c_{b_2}}{c_\sigma} \frac{\partial \widetilde{\nu}}{\partial x_i} \right] n_i \frac{\delta \widetilde{\nu}}{\delta b_n} dS \\ & - \int_S \frac{\widetilde{\nu}_\alpha (\nu + \widetilde{\nu})}{c_\sigma} n_i \frac{\delta}{\delta b_n} \left(\frac{\partial \widetilde{\nu}}{\partial x_i} \right) dS + \int_S \widetilde{\nu}_\alpha \widetilde{\nu} \frac{C_\omega}{\omega} Y_l \epsilon_{lji} n_j \frac{\delta v_i}{\delta b_n} dS \\ & + \int_\Omega \Bigg[- \widetilde{\nu}_\alpha v_j \frac{\partial \widetilde{\nu}}{\partial x_k} + \widetilde{\nu}_\alpha \frac{\partial}{\partial x_k} \left(\frac{\nu + \widetilde{\nu}}{c_\sigma} \frac{\partial \widetilde{\nu}}{\partial x_j} \right) - \frac{\nu + \widetilde{\nu}}{c_\sigma} \frac{\partial \widetilde{\nu}_\alpha}{\partial x_j} \frac{\partial \widetilde{\nu}}{\partial x_k} \\ & + \widetilde{\nu}_\alpha \frac{2 c_{b2}}{c_\sigma} \frac{\partial \widetilde{\nu}}{\partial x_j} \frac{\partial \widetilde{\nu}}{\partial x_k} - \widetilde{\nu}_\alpha \widetilde{\nu} \frac{C_\omega}{\omega} Y_l \epsilon_{lji} \frac{\partial v_i}{\partial x_k} \Bigg] \frac{\partial}{\partial x_j} \left(\frac{\delta x_k}{\delta b_n} \right) d\Omega \end{aligned} \tag{2.13}$$

where

$$
\begin{aligned}
C_{\widetilde{\nu}} &= \frac{C_\omega}{\kappa^2\Delta^2}\left(f_{v_2}+\widetilde{\nu}\frac{\partial f_{v_2}}{\partial\widetilde{\nu}}\right)+\frac{c_{w_1}d}{\widetilde{\nu}}C+\frac{c_{w_1}f_w}{\Delta^2}\\
C_\omega &= -\left(c_{b_1}+c_{w_1}\frac{d}{\widetilde{S}}C\right)\\
C_\Delta &= -\frac{2}{\Delta^3}\left[c_{w_1}\widetilde{\nu}f_w+c_{w_1}C\Delta^2 d+\frac{\widetilde{\nu}f_{v_2}}{\kappa^2}C_\omega\right]\\
C &= \frac{\widetilde{\nu}f_w}{\Delta^2}\,\frac{c_{w_3}^6}{g(g+c_{w_3}^6)}\left(1+6c_{w_2}d^5-c_{w_2}\right)\\
\frac{\partial f_{v_2}}{\partial\widetilde{\nu}} &= \frac{1}{\nu}\frac{3c_{v_1}^3\chi^4-(\chi^3+c_{v_1}^3)^2}{(1+\chi f_{v_1})^2\left(\chi^3+c_{v_1}^3\right)^2}
\end{aligned}
\tag{2.14}
$$

Finally, the term including the Hamilton-Jacobi equation residual yields

$$
\begin{aligned}
\int_\Omega \Delta_\alpha\frac{\delta\mathcal{R}^\Delta}{\delta b_n}d\Omega = &-\int_\Omega\frac{\partial}{\partial x_i}\left(2\Delta_\alpha\frac{\partial\Delta}{\partial x_i}\right)\frac{\delta\Delta}{\delta b_n}d\Omega-\int_\Omega 2\Delta_\alpha\frac{\partial\Delta}{\partial x_j}\frac{\partial\Delta}{\partial x_k}\frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right)d\Omega\\
&+\int_S 2\Delta_\alpha\frac{\partial\Delta}{\partial x_i}n_i\frac{\delta\Delta}{\delta b_n}dS
\end{aligned}
\tag{2.15}
$$

2.5 The Field Adjoint Equations

All terms resulting from the differentiation of eqs. 2.1a to 2.1d and 2.8 in the previous sections are re-arranged and aggregated into multipliers of the derivatives of flow variables and geometric quantities w.r.t. $\boldsymbol{b}$. Thus,

$$
\begin{aligned}
\frac{\delta\mathcal{L}}{\delta b_n} = &\int_\Omega\mathcal{R}^q\frac{\delta p}{\delta b_n}d\Omega+\int_\Omega\mathcal{R}_i^{\boldsymbol{u}}\frac{\delta v_i}{\delta b_n}d\Omega+\int_\Omega\mathcal{R}^{\widetilde{\nu_\alpha}}\frac{\delta\widetilde{\nu}}{\delta b_n}d\Omega+\int_\Omega\mathcal{R}^{\Delta_\alpha}\frac{\delta\Delta}{\delta b_n}d\Omega\\
&+\int_S\mathcal{BC}^p\frac{\delta p}{\delta b_n}dS+\int_S\mathcal{BC}_i^{\boldsymbol{v}}\frac{\delta v_i}{\delta b_n}dS+\int_S\mathcal{BC}_{ij}^\tau\frac{\delta\tau_{ij}}{\delta b_n}dS+\int_S\mathcal{BC}^{\widetilde{\nu}}\frac{\delta\widetilde{\nu}}{\delta b_n}dS+\int_S\mathcal{BC}^\Delta\frac{\delta\Delta}{\delta b_n}dS\\
&-\int_S\frac{\widetilde{\nu_\alpha}(\nu+\widetilde{\nu})}{c_\sigma}n_i\frac{\delta}{\delta b_n}\left(\frac{\partial\widetilde{\nu}}{\partial x_i}\right)dS+\int_S\frac{\partial F_{S,i}}{\partial x_k}n_i\frac{\delta x_k}{\delta b_n}dS+\int_S F_{S,i}\frac{\delta(n_i dS)}{\delta b_n}\\
&+\int_\Omega\left(\mathcal{A}_{jk}^L+\mathcal{A}_{jk}^T\right)\frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right)d\Omega
\end{aligned}
\tag{2.16}
$$

where

$$\mathcal{A}^L_{jk} = q\frac{\partial v_j}{\partial x_k} - u_i v_j \frac{\partial v_i}{\partial x_k} + u_i \frac{\partial \tau_{ij}}{\partial x_k} - \tau^{\alpha}_{ij}\frac{\partial v_i}{\partial x_k} - u_j \frac{\partial p}{\partial x_k} \tag{2.17a}$$

$$\begin{aligned}\mathcal{A}^T_{jk} = &-\widetilde{\nu_\alpha} v_j \frac{\partial \widetilde{\nu}}{\partial x_k} + \widetilde{\nu_\alpha}\frac{\partial}{\partial x_k}\left(\frac{\nu+\widetilde{\nu}}{c_\sigma}\frac{\partial \widetilde{\nu}}{\partial x_j}\right) - \frac{\nu+\widetilde{\nu}}{c_\sigma}\frac{\partial \widetilde{\nu_\alpha}}{\partial x_j}\frac{\partial \widetilde{\nu}}{\partial x_k} \\ &+\widetilde{\nu_\alpha}\frac{2c_{b2}}{c_\sigma}\frac{\partial \widetilde{\nu}}{\partial x_j}\frac{\partial \widetilde{\nu}}{\partial x_k} - \widetilde{\nu_\alpha}\widetilde{\nu}\frac{C_\omega}{\omega}Y_l \epsilon_{lji}\frac{\partial v_i}{\partial x_k} - 2\Delta_\alpha \frac{\partial \Delta}{\partial x_j}\frac{\partial \Delta}{\partial x_k}\end{aligned} \tag{2.17b}$$

In order to overcome the computation of the primal variables' derivatives w.r.t. $\boldsymbol{b}$, the following terms within the first four volume integrals of eq. 2.16 are set to zero,

$$\mathcal{R}^q = -\frac{\partial u_i}{\partial x_i} = 0 \tag{2.18a}$$

$$\mathcal{R}^{\boldsymbol{u}}_i = \underbrace{u_j \frac{\partial v_j}{\partial x_i}}_{\boldsymbol{ATC}} - v_j \frac{\partial u_i}{\partial x_j} - \frac{\partial \tau^{\alpha}_{ij}}{\partial x_j} + \frac{\partial q}{\partial x_i} + \widetilde{\nu_\alpha}\frac{\partial \widetilde{\nu}}{\partial x_i} - \frac{\partial}{\partial x_l}\left(\widetilde{\nu_\alpha}\widetilde{\nu}\frac{C_\omega}{\omega}Y_j \epsilon_{jli}\right) = 0\ , \quad i = 1, 2(, 3) \tag{2.18b}$$

$$\begin{aligned}\mathcal{R}^{\widetilde{\nu_\alpha}} = &-v_i \frac{\partial \widetilde{\nu_\alpha}}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\frac{\nu+\widetilde{\nu}}{c_\sigma}\frac{\partial \widetilde{\nu_\alpha}}{\partial x_i}\right) + \frac{1}{c_\sigma}\frac{\partial \widetilde{\nu_\alpha}}{\partial x_i}\frac{\partial \widetilde{\nu}}{\partial x_i} + \frac{2c_{b2}}{c_\sigma}\frac{\partial}{\partial x_i}\left(\widetilde{\nu_\alpha}\frac{\partial \widetilde{\nu}}{\partial x_i}\right) \\ &+\widetilde{\nu_\alpha}\widetilde{\nu}C_{\widetilde{\nu}} + f_{v_1}\frac{\partial u_j}{\partial x_i}\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j}\right) + \widetilde{\nu_\alpha}(D-P) = 0\end{aligned} \tag{2.18c}$$

$$\mathcal{R}^{\Delta_\alpha} = -2\frac{\partial}{\partial x_i}\left(\Delta_\alpha \frac{\partial \Delta}{\partial x_i}\right) + \widetilde{\nu_\alpha}\widetilde{\nu}C_\Delta = 0 \tag{2.18d}$$

giving rise to the Field Adjoint Equations (FAE) to be solved for u_i, q, $\widetilde{\nu_\alpha}$ and Δ_α. The term marked with ATC on the r.h.s. of eq. 2.18b is the so-called Adjoint Transposed-Convection (ATC) term, oftentimes responsible for most of the convergence issues encountered in the solution of the adjoint equations. A frequently used "remedy" for this is the damping of the ATC close to the wall using an appropriate sensor however, excessive damping is sometimes required to secure convergence and this can have a negative impact on the optimization. This issue is dealt with in Chapter 3. Terms within the first five surface integrals

of eq. 2.16 are

$$\mathcal{BC}^{p} = u_i n_i + \frac{\partial F_{S,i}}{\partial p} n_i \tag{2.19a}$$

$$\mathcal{BC}^{\boldsymbol{v}}_{i} = -q n_i + v_j u_i n_j + \tau^{\alpha}_{ij} n_j + \widetilde{\nu}_{\alpha} \widetilde{\nu} \frac{C_{\omega}}{\omega} Y_l \epsilon_{lji} n_j + \frac{\partial F_{S,j}}{\partial v_i} n_j \ , \quad i = 1, 2(, 3) \tag{2.19b}$$

$$\mathcal{BC}^{\tau}_{ij} = -u_i n_j + \frac{\partial F_{S,l}}{\partial \tau_{ij}} n_l \tag{2.19c}$$

$$\mathcal{BC}^{\widetilde{\nu}} = \widetilde{\nu}_{\alpha} v_i n_i + \frac{\nu + \widetilde{\nu}}{c_{\sigma}} \frac{\partial \widetilde{\nu}_{\alpha}}{\partial n} - \frac{\widetilde{\nu}_{\alpha}}{c_{\sigma}} \left(1 + 2 c_{b2}\right) \frac{\partial \widetilde{\nu}}{\partial n} \tag{2.19d}$$

$$\mathcal{BC}^{\Delta} = 2 \Delta_{\alpha} \frac{\partial \Delta}{\partial n} \tag{2.19e}$$

and are examined on a case-by-case basis for the S_O, S_I, S_W and S_S boundaries taking into account the BCs of the primal PDEs.

2.6 The Adjoint Boundary Conditions

The adjoint BCs are derived mainly by setting the multipliers of the primal variables' derivatives to zero, so as to eliminate the surface integrals in eq. 2.16 that contain them, and overcome the computation of these costly quantities, following a similar procedure as the derivation of the FAE. Along some boundaries, these surface integrals may vanish automatically due to the BCs of the flow equations or since derivatives of geometric quantities w.r.t. $\boldsymbol{b}$ over non-parameterized boundaries are zero. The procedure for deriving the adjoint BCs is thoroughly presented in [114, 64, 36]. Here, these expressions are briefly

presented below.

$$
\begin{aligned}
S_I : & \; u_{\langle n\rangle} = -\frac{\partial F_{S,i}}{\partial p} n_i, \quad u_{\langle t^l\rangle} = \frac{\partial F_{S,k}}{\partial \tau_{ij}} n_k t_i^l n_j + \frac{\partial F_{S,k}}{\partial \tau_{ij}} n_k t_j^l n_i \quad l = I, II \\
& \frac{\partial q}{\partial n} = 0, \quad \widetilde{\nu_\alpha} = 0, \quad \Delta_\alpha = 0 \\
S_O : & \; \frac{\partial u_{\langle n\rangle}}{\partial n} = 0, \quad v_{\langle n\rangle} u_{\langle t^l\rangle} + \tau_{ij}^\alpha t_i^l n_j + \widetilde{\nu_\alpha}\widetilde{\nu}\frac{C_\omega}{\omega} Y_k \epsilon_{kji} n_j t_i^l + \frac{\partial F_{S,j}}{\partial v_i} n_j t_i^l = 0 \quad l = I, II \\
& -q + v_{\langle n\rangle} u_{\langle n\rangle} + \tau_{ij}^\alpha n_i n_j + \widetilde{\nu_\alpha}\widetilde{\nu}\frac{C_\omega}{\omega} Y_l \epsilon_{lji} n_j n_i + \frac{\partial F_{S,j}}{\partial v_i} n_j n_i = 0 \\
& \widetilde{\nu_\alpha} v_{\langle n\rangle} + \frac{\nu + \widetilde{\nu}}{c_\sigma}\frac{\partial \widetilde{\nu_\alpha}}{\partial n} = 0, \quad \Delta_\alpha = 0 \\
S_W : & \; u_{\langle n\rangle} = -\frac{\partial F_{S,i}}{\partial p} n_i, \quad u_{\langle t^l\rangle} = \frac{\partial F_{S,k}}{\partial \tau_{ij}} n_k t_i^l n_j + \frac{\partial F_{S,k}}{\partial \tau_{ij}} n_k t_j^l n_i \quad l = I, II \\
& \frac{\partial q}{\partial n} = 0, \quad \widetilde{\nu_\alpha} = 0, \quad \frac{\partial \Delta_\alpha}{\partial n} = 0 \\
S_S : & \; u_{\langle n\rangle} = 0, \quad \frac{\partial u_{\langle t^l\rangle}}{\partial n} = 0 \quad l = I, II, \quad \frac{\partial q}{\partial n} = 0, \quad \frac{\partial \widetilde{\nu_\alpha}}{\partial n} = 0, \quad \frac{\partial \Delta_\alpha}{\partial n} = 0
\end{aligned}
\tag{2.20}
$$

For the adjoint velocity, BCs are expressed in terms of the normal component, $u_{\langle n\rangle} = u_i n_i$, and two tangential ones, $u_{\langle t^I\rangle} = u_i t_i^I$ and $u_{\langle t^{II}\rangle} = u_i t_i^{II}$, defined as in the BCs for $\boldsymbol{v}$ along S_S in Equation 2.3.

2.7 The FI Adjoint Formulation

Finally, after satisfying all BCs in eq. 2.20 and solving eqs. 2.18a to 2.18d, the SDs can be computed by

$$
\begin{aligned}
\left.\frac{\delta \mathcal{L}}{\delta b_n}\right|_{FI} \equiv \left.\frac{\delta F}{\delta b_n}\right|_{FI} = & \int_{S_{W_p}} \frac{\partial F_{S,i}}{\partial x_k} n_i \frac{\delta x_k}{\delta b_n} dS + \int_{S_{W_p}} F_{S,i} \frac{\delta (n_i dS)}{\delta b_n} + \int_\Omega \left(\mathcal{A}_{jk}^L + \mathcal{A}_{jk}^T\right) \frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right) d\Omega \\
& + \int_{S_{W_p}} \left(u_{\langle n\rangle} - \phi_{\langle n\rangle\langle n\rangle}\right) \tau_{ij} \frac{\delta (n_i n_j)}{\delta b_n} dS - \int_{S_{W_p}} \left(\phi_{\langle t^I\rangle\langle t^I\rangle}\right) \tau_{ij} \frac{\delta (t_i^I t_j^I)}{\delta b_n} dS \\
& - \int_{S_{W_p}} \left(\phi_{\langle t^{II}\rangle\langle t^{II}\rangle}\right) \tau_{ij} \frac{\delta (t_i^{II} t_j^{II})}{\delta b_n} dS - \int_{S_{W_p}} \left(\phi_{\langle t^{II}\rangle\langle t^I\rangle} + \phi_{\langle t^I\rangle\langle t^{II}\rangle}\right) \tau_{ij} \frac{\delta (t_i^{II} t_j^I)}{\delta b_n} dS
\end{aligned}
\tag{2.21}
$$

where $\phi_{\langle n\rangle\langle n\rangle} = \frac{\partial F_{S,k}}{\partial \tau_{ij}} n_i n_j n_k$, $\phi_{\langle t^I\rangle\langle t^{II}\rangle} = \frac{\partial F_{S,k}}{\partial \tau_{ij}} t_i^I t_j^{II} n_k$ and so forth.

Using the objective function's gradient, a correction is computed for $\boldsymbol{b}$, based on a variety of update methods available in the literature, and the parameterized/controlled boundary is displaced. The internal grid nodes are displaced using a Grid Displacement Model (GDM) and the optimization restarts.

GDMs could either be algebraic or based on solving PDEs for the displacement field.

The expression in eq. 2.21 involves derivatives of geometric quantities w.r.t. $\boldsymbol{b}$ which need be computed on the controlled/parameterized boundaries S_{W_p}. Most of these terms can be computed easily, most often through closed-form expressions resulting from the parameterization. The volume integral on the r.h.s. of eq. 2.21 involves the computation of grid sensitivities throughout the whole domain. Integrals that involve grid sensitivities quantify the sensitivity of the objective function to the displacement of the boundary and interior grid nodes. The cost of computing these volume integrals differs depending on the parameterization. It might be low if the derivatives are given by closed-form expressions, or high if FDs are used. In the latter case, the design variables are perturbed, one at a time, using a predetermined step size, the controlled boundary is displaced accordingly and the internal grid is adapted to the new boundaries using the GDM. After this, grid sensitivities are computed throughout the whole domain. Since this is done for each b_n separately, the cost scales with N. This formulation for the SDs that involves volume integrals of grid sensitivities is the so-called Field Integral (FI) adjoint, see [66].

2.8 The E-SI Adjoint

In order to overcome the computation of grid sensitivities throughout the whole domain, an alternative formulation for the SDs was first proposed in [66] by considering the adjoint to the GDM. For the internal grid nodes, it is assumed that their displacements m_i can be described by a set of Laplacian PDEs (GDM)

$$\mathcal{R}_i^{\boldsymbol{m}} = \frac{\partial^2 m_i}{\partial x_j^2} = 0 \ , \quad i = 1, 2(, 3) \tag{2.22}$$

The residual of eq. 2.22 is appended to the Lagrangian function,

$$\mathcal{L} = F + \int_\Omega \left(u_i \mathcal{R}_i^{\boldsymbol{v}} + q\mathcal{R}^p + \widetilde{\nu}_\alpha \mathcal{R}^{\widetilde{\nu}} + \Delta_\alpha \mathcal{R}^\Delta + m_i^\alpha \mathcal{R}_i^{\boldsymbol{m}} \right) d\Omega \tag{2.23}$$

where m_i^α are field components adjoint to the components of $\boldsymbol{m}$. Equation 2.23 is again differentiated. The procedure is the same as the one presented in section 2.2, for the residuals of eqs. 2.1a to 2.1d, and leads to eq. 2.21 but with the addition of terms that result from the differentiation of eq. 2.22. Following [36], these are

$$\int_\Omega m_i^\alpha \frac{\delta \mathcal{R}_i^{\boldsymbol{m}}}{\delta b_n} d\Omega = \int_S m_i^\alpha n_j \frac{\delta}{\delta b_n} \left(\frac{\partial m_i}{\partial x_j} \right) dS - \int_S \frac{\partial m_i^\alpha}{\partial x_j} n_j \frac{\delta x_i}{\delta b_n} dS + \int_\Omega \frac{\partial^2 m_k^\alpha}{\partial x_j^2} \frac{\delta x_k}{\delta b_n} d\Omega \tag{2.24}$$

The displacement field is written in terms of the current grid coordinates. The grid sensitivities refer to the current nodes that are displaced after updating $\boldsymbol{b}$. The Green-Gauss theorem is used on the following term (see eq. 2.21) to yield

$$\int_{\Omega}\left(\mathcal{A}^{L}_{jk}+\mathcal{A}^{T}_{jk}\right)\frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right)d\Omega=\int_{S}\left(\mathcal{A}^{L}_{jk}+\mathcal{A}^{T}_{jk}\right)n_j\frac{\delta x_k}{\delta b_n}dS-\int_{\Omega}\frac{\partial}{\partial x_j}\left(\mathcal{A}^{L}_{jk}+\mathcal{A}^{T}_{jk}\right)\frac{\delta x_k}{\delta b_n}d\Omega \tag{2.25}$$

By grouping together the multipliers of grid sensitivities within volume integrals in eqs. 2.24 and 2.25, the SDs expression becomes

$$\begin{aligned}\frac{\delta\mathcal{L}}{\delta b_n}=&\int_{\Omega}\mathcal{R}^{\boldsymbol{m}^{\alpha}}_{k}\frac{\delta x_k}{\delta b_n}d\Omega+\int_{S}m^{\alpha}_{i}n_j\frac{\delta}{\delta b_n}\left(\frac{\partial m_i}{\partial x_j}\right)dS+\int_{S_{W_p}}\frac{\partial F_{S,i}}{\partial x_k}n_i\frac{\delta x_k}{\delta b_n}dS+\int_{S_{W_p}}F_{S,i}\frac{\delta(n_i dS)}{\delta b_n}\\
&+\int_{S_{W_p}}\left(\mathcal{A}^{L}_{jk}+\mathcal{A}^{T}_{jk}\right)n_j\frac{\delta x_k}{\delta b_n}dS-\int_{S_{W_p}}\frac{\partial m^{\alpha}_{k}}{\partial x_j}n_j\frac{\delta x_k}{\delta b_n}dS\\
&+\int_{S_{W_p}}\left(u_{\langle n\rangle}-\phi_{\langle n\rangle\langle n\rangle}\right)\tau_{ij}\frac{\delta(n_i n_j)}{\delta b_n}dS-\int_{S_{W_p}}\left(\phi_{\langle t^I\rangle\langle t^I\rangle}\right)\tau_{ij}\frac{\delta(t^I_i t^I_j)}{\delta b_n}dS\\
&-\int_{S_{W_p}}\left(\phi_{\langle t^{II}\rangle\langle t^{II}\rangle}\right)\tau_{ij}\frac{\delta(t^{II}_i t^{II}_j)}{\delta b_n}dS-\int_{S_{W_p}}\left(\phi_{\langle t^{II}\rangle\langle t^{I}\rangle}+\phi_{\langle t^{I}\rangle\langle t^{II}\rangle}\right)\tau_{ij}\frac{\delta(t^{II}_i t^{I}_j)}{\delta b_n}dS\end{aligned} \tag{2.26}$$

In order to eliminate the first volume integral in eq. 2.26, the following term is set to zero, giving rise to the adjoint to the GDM field equation,

$$\mathcal{R}^{\boldsymbol{m}^{\alpha}}_{k}=\frac{\partial^2 m^{\alpha}_{k}}{\partial x_j^2}-\frac{\partial}{\partial x_j}\left(\mathcal{A}^{L}_{jk}+\mathcal{A}^{T}_{jk}\right)=0,\quad k=1,2(,3) \tag{2.27}$$

for which the corresponding BCs are

$$m^{\alpha}_{i}=0 \tag{2.28}$$

so as to eliminate the surface integral involving $\frac{\delta}{\delta b_n}\left(\frac{\partial m_i}{\partial x_j}\right)$ in eq. 2.26. After solving

eq. 2.27, in addition to eqs. 2.18a to 2.18d, a final SDs expression is derived

$$\begin{aligned}\left.\frac{\delta\mathcal{L}}{\delta b_n}\right|_{E-SI} \equiv \left.\frac{\delta F}{\delta b_n}\right|_{E-SI} &= \int_{S_{W_p}} \frac{\partial F_{S,i}}{\partial x_k} n_i \frac{\delta x_k}{\delta b_n} dS + \int_{S_{W_p}} F_{S,i} \frac{\delta(n_i dS)}{\delta b_n} \\ &+ \int_{S_{W_p}} \left(\mathcal{A}^L_{jk} + \mathcal{A}^T_{jk}\right) n_j \frac{\delta x_k}{\delta b_n} dS - \int_{S_{W_p}} \frac{\partial m^\alpha_k}{\partial x_j} n_j \frac{\delta x_k}{\delta b_n} dS \\ &+ \int_{S_{W_p}} \left(u_{\langle n\rangle} - \phi_{\langle n\rangle\langle n\rangle}\right) \tau_{ij} \frac{\delta(n_i n_j)}{\delta b_n} dS - \int_{S_{W_p}} \left(\phi_{\langle t^I\rangle\langle t^I\rangle}\right) \tau_{ij} \frac{\delta(t^I_i t^I_j)}{\delta b_n} dS \\ &- \int_{S_{W_p}} \left(\phi_{\langle t^{II}\rangle\langle t^{II}\rangle}\right) \tau_{ij} \frac{\delta(t^{II}_i t^{II}_j)}{\delta b_n} dS - \int_{S_{W_p}} \left(\phi_{\langle t^{II}\rangle\langle t^{I}\rangle} + \phi_{\langle t^{I}\rangle\langle t^{II}\rangle}\right) \tau_{ij} \frac{\delta(t^{II}_i t^{I}_j)}{\delta b_n} dS\end{aligned} \tag{2.29}$$

that uses only surface integrals of grid sensitivities. The Laplacian GDM of eq. 2.22 and its adjoint are assumed even in cases where the grid is displaced using a different GDM model. In [39], this was shown to be a harmless assumption. This alternative adjoint formulation is referred to as the Enhanced Surface Integral (E-SI) adjoint [66], a step-up from a previous formulation [114], based on surface integrals only, that however totally neglects integrals of grid sensitivities.

Depending on the parameterization, grid sensitivities may be computed throughout Ω using closed-form relations. In that case, eq. 2.21 becomes efficient to use. Otherwise, to avoid the costly computation of the volume integral in eq. 2.21, eq. 2.29 is used instead. This will become clear in the following section.

2.9 Parameterization Techniques

The parameterization techniques used in this thesis are described in the following sections.

2.9.1 Volumetric B-Splines

The surface to be optimized, as well as the part of the grid that surrounds or is contained within it, is parameterized and deformed with the help of Volumetric B-Splines (VBS). A number of control points (CPs) are arranged around the surface in a 3D lattice, the shape of which may not necessarily be a parallepiped. Let n_u, n_v and n_w be the number of CPs along each of the 3 main directions of the lattice. This lattice forms a morphing box (an example is shown in fig. 2.1) and for the grid and surface points contained within this box, the relationship between each grid point's coordinates and those of the CPs is given by

$$x_i(u,v,w) = U_{k,p_u}(u) V_{l,p_v}(v) W_{m,p_w}(w) X_i^{klm} \tag{2.30}$$

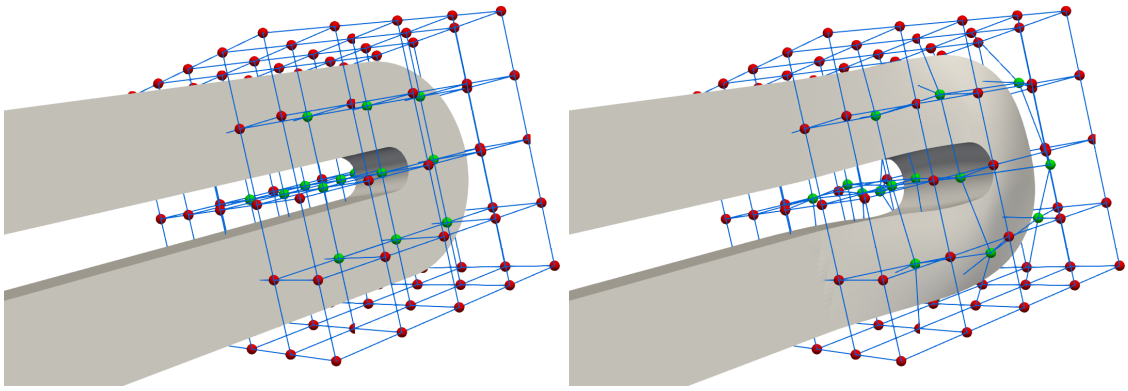


Figure 2.1: A VBS morphing box example. The U section of a duct is parameterized in 3D. The following coloring of CPs is adopted throughout the whole thesis: The CPs in red, populating the outer layers of the rectangular lattice, are not allowed to move throughout the optimization. The shape of the paremeterized surface changes by moving the green (inner) CPs. **Left**: The initial shape of the duct. **Right**: The one optimized for minimized total pressure losses between the inlet and outlet of the duct.

where u, v, w are parametric coordinates, U, V, W are a set of basis functions of polynomial degrees p_u, p_v, p_w (not necessarily the same for each basis function). X_i are the coordinates of the CPs and serve as the design variables. Indices k, l and m correspond to the location of each CP in the 3 principal directions spanned by the lattice, respectively, and together define the ID of each CP. The basis functions are defined using the Cox-de Boor formula [124], written here only for U

$$U_{k,0}(u) = \begin{cases} 1 & \text{if } t_i < u < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$
$$U_{k,p_u}(u) = \frac{u - t_i}{t_{i+p_u} - t_i} U_{k,p_u-1}(u) + \frac{t_{i+p_u+1} - u}{t_{i+p_u+1} - t_{i+p_u}} U_{k+1,p_u-1}(u) \tag{2.31}$$

where t_i are the $p_u + n_u + 1$ elements of a knot vector

$$\boldsymbol{t} = [\underbrace{0, ..., 0}_{p_u}, 0, \frac{1}{N_u}, ..., \frac{N_u - 1}{N_u}, 1, \underbrace{1, ..., 1}_{p_u}] \tag{2.32}$$

where $N_u = n_u - p_u + 1$. U, u, k are substituted with V, v, l or W, w, m in eqs. 2.31 and 2.32 to get the formula for V and W.

For a given set of basis functions and CPs, the parametric coordinates u, v, w satisfy

$$R_i^{VBS} = U_{k,p_u}(u) V_{l,p_v}(v) W_{m,p_w}(w) X_i^{klm} - x_i = 0 \tag{2.33}$$

This constitutes a 3×3 non-linear system which is solved for each grid node independently, using the Newton-Raphson method. The parametric coordinates

are thus computed and their values are kept constant during the optimization. Once the CPs are displaced, the coordinates of the new grid points can directly be computed using eq. 2.30. Since $X_i^{k,l,m}$ stand for the design variables b_n, the computation of grid sensitivities is straightforward: eq. 2.30 is differentiated w.r.t. b_n yielding $\frac{\partial x_k}{\partial b_n}$ for the grid points enclosed by the morphing box. This makes the use of the FI adjoint more efficient in case VBS parameterization is used.

In many problems, this thesis is dealing with, bound constraints are placed on the CPs of the morphing box, to prevent a CP from crossing the bisectors formed by this CP and its immediate neighbours:

$$\begin{aligned}
\frac{X_1^{k-1,l,m}+X_1^{k,l,m}}{2} &\leq X_1^{k,l,m} \leq \frac{X_1^{k,l,m}+X_1^{k+1,l,m}}{2} \\
\frac{X_2^{k,l-1,m}+X_2^{k,l,m}}{2} &\leq X_2^{k,l,m} \leq \frac{X_2^{k,l,m}+X_2^{k,l+1,m}}{2} \\
\frac{X_3^{k,l,m-1}+X_3^{k,l,m}}{2} &\leq X_3^{k,l,m} \leq \frac{X_3^{k,l,m}+X_3^{k,l,m+1}}{2}
\end{aligned} \tag{2.34}$$

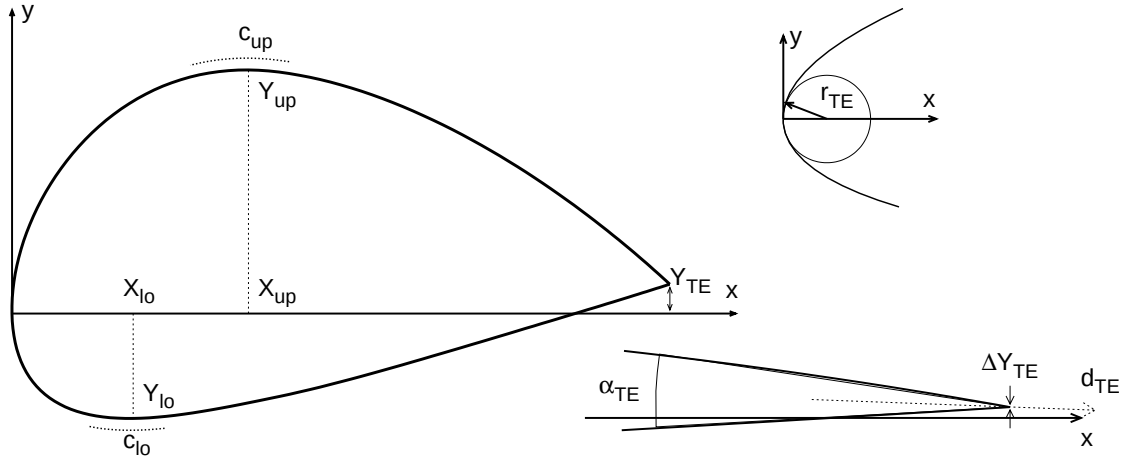
In case some of the CPs cross each other, it is possible that the deformed grid becomes invalid. The bound values are updated between consecutive optimization cycles once after the CPs have been moved.

2.9.2 PARSEC

PARSEC is a common method, developed by [144], for airfoil parameterization. The parameters are described in fig. 2.2. The y coordinate of any point on the upper (suction) or lower (pressure) sides of the airfoil, is given by

$$y_{up} = \sum_{i=1}^{6} \alpha_{i,up} x^{i-0.5} \tag{2.35}$$

$$y_{lo} = \sum_{i=1}^{6} \alpha_{i,lo} x^{i-0.5} \tag{2.36}$$



Parameter	Description
r_{LE}	LE radius
X_{up}	Upper crest x coordinate
Y_{up}	Upper crest y coordinate
c_{up}	Upper crest curvature
X_{lo}	Lower crest x coordinate
Y_{lo}	Lower crest y coordinate
c_{lo}	Lower crest curvature
Y_{TE}	TE offset from x axis
ΔY_{TE}	TE thickness
d_{TE}	TE direction
α_{TE}	TE wedge angle

Figure 2.2: PARSEC parameters reference and description. "TE" and "LE" are the trailing and leading edges, respectively.

where the coefficients $\alpha_{i,up}$ and $\alpha_{i,lo}$ depend on the PARSEC parameters and are found by solving the system

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ X_{up}^{\frac{1}{2}} & X_{up}^{\frac{3}{2}} & X_{up}^{\frac{5}{2}} & X_{up}^{\frac{7}{2}} & X_{up}^{\frac{9}{2}} & X_{up}^{\frac{11}{2}} \\ \frac{1}{2} & \frac{3}{2} & \frac{5}{2} & \frac{7}{2} & \frac{9}{2} & \frac{11}{2} \\ \frac{1}{2}X_{up}^{-\frac{1}{2}} & \frac{3}{2}X_{up}^{\frac{1}{2}} & \frac{5}{2}X_{up}^{\frac{3}{2}} & \frac{7}{2}X_{up}^{\frac{5}{2}} & \frac{9}{2}X_{up}^{\frac{7}{2}} & \frac{11}{2}X_{up}^{\frac{9}{2}} \\ -\frac{1}{4}X_{up}^{-\frac{3}{2}} & \frac{3}{4}X_{up}^{-\frac{1}{2}} & \frac{15}{4}X_{up}^{\frac{1}{2}} & \frac{15}{4}X_{up}^{\frac{1}{2}} & \frac{63}{4}X_{up}^{\frac{5}{2}} & \frac{99}{4}X_{up}^{\frac{7}{2}} \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_{1,up} \\ \alpha_{2,up} \\ \alpha_{3,up} \\ \alpha_{4,up} \\ \alpha_{5,up} \\ \alpha_{6,up} \end{bmatrix} = \begin{bmatrix} Y_{TE} + \frac{\Delta Y_{TE}}{2} \\ Y_{up} \\ tan(d_{TE} - \frac{\alpha_{TE}}{2}) \\ 0 \\ c_{up} \\ \sqrt{2r_{LE}} \end{bmatrix} \tag{2.37}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ X_{lo}^{\frac{1}{2}} & X_{lo}^{\frac{3}{2}} & X_{lo}^{\frac{5}{2}} & X_{lo}^{\frac{7}{2}} & X_{lo}^{\frac{9}{2}} & X_{lo}^{\frac{11}{2}} \\ \frac{1}{2} & \frac{3}{2} & \frac{5}{2} & \frac{7}{2} & \frac{9}{2} & \frac{11}{2} \\ \frac{1}{2}X_{lo}^{-\frac{1}{2}} & \frac{3}{2}X_{lo}^{\frac{1}{2}} & \frac{5}{2}X_{lo}^{\frac{3}{2}} & \frac{7}{2}X_{lo}^{\frac{5}{2}} & \frac{9}{2}X_{lo}^{\frac{7}{2}} & \frac{11}{2}X_{lo}^{\frac{9}{2}} \\ -\frac{1}{4}X_{lo}^{-\frac{3}{2}} & \frac{3}{4}X_{lo}^{-\frac{1}{2}} & \frac{15}{4}X_{lo}^{\frac{1}{2}} & \frac{15}{4}X_{lo}^{\frac{1}{2}} & \frac{63}{4}X_{lo}^{\frac{5}{2}} & \frac{99}{4}X_{lo}^{\frac{7}{2}} \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_{1,lo} \\ \alpha_{2,lo} \\ \alpha_{3,lo} \\ \alpha_{4,lo} \\ \alpha_{5,lo} \\ \alpha_{6,lo} \end{bmatrix} = \begin{bmatrix} Y_{TE} - \frac{\Delta Y_{TE}}{2} \\ Y_{lo} \\ tan(d_{TE} + \frac{\alpha_{TE}}{2}) \\ 0 \\ c_{lo} \\ \sqrt{2r_{LE}} \end{bmatrix} \tag{2.38}$$

In this thesis, the design variables are the coefficients $\alpha_{i,up}$ and $\alpha_{i,lo}$. In case the trailing edge is sharp and kept fixed throughout the optimization, such as in section 4.3.1, $\alpha_{1,up}$ and $\alpha_{1,lo}$ are equal to each other and only 10 out of 12 design variables are effectively used. Grid sensitivities are computed analytically, through eq. 2.35, only on the boundary. For computing them throughout the computational domain, as required by the FI adjoint, FDs are used. Each one of the PARSEC coefficients are perturbed by ε, the boundary and the grid are displaced using a GDM and grid sensitivities are computed for all grid nodes. The E-SI adjoint totally avoids this and is the method of choice with this parameterization technique.

Chapter 3

Stabilization of Steady Flow and Adjoint Solvers in the Presence of Unsteadiness

This chapter presents the Recursive Projection Method (RPM) which is used to stabilize the steady-state continuous adjoint and its corresponding primal solver, both built-in to OpenFOAM's *adjointOptimisation* library. In the cases involved, convergence difficulties of both solvers emerge due to flow unsteadiness.

First, it is investigated whether steady-state flow solvers and their adjoints, which are enabled to converge using the RPM, can indeed support shape optimization in flows that exhibit mild to moderate unsteadiness. To answer this question, the RPM stabilization is also used for optimization in cases that push the utilization of steady-state solvers to an extreme, such as in the case of vortex shedding; then, the so-optimized geometries are re-assessed using a URANS equations' solver. In such extreme cases, the optimized solutions, computed using steady solvers with the help of the RPM, are not expected to coincide with the local minima or extrema that would have been computed if an optimization using URANS was performed. For these cases, the optimized solutions, computed using steady solvers, are used as starting points for a URANS-based shape optimization loop which, as it will be seen in this chapter, allows the optimization to reach an optimized solution faster.

Finally, since the RPM allows for the stabilization of the adjoint equations without severing any terms, a reference adjoint solution can be computed. For the problems examined in this thesis, this reference solution is unattainable otherwise and offers a means of evaluating the effect of ATC treatments on the accuracy of SDs. Similarly, another widely used workaround for whenever the flow solver cannot fully converge, either due to unsteadiness or not, relies on averaging "pseudo-steady" flow fields and using them as input to the adjoint solver. The effect of this workaround is assessed by comparing optimized solutions computed using this technique and fully converged primal and adjoint solvers via the RPM.

The chapter is outlined as follows: The RPM is presented first and some practical aspects of its implementation are given. The SIMPLE algorithm, used

by OpenFOAM to solve the incompressible Navier-Stokes equations is, then, presented and is related to the way the RPM treats the divergence of iterative schemes. Then, the use of the RPM for the stabilization of steady primal and adjoint solvers is demonstrated in academic test-cases in order to showcase its capabilities and to assess the aforementioned treatments.

3.1 The Recursive Projection Method

The RPM was developed [143] for the purpose of treating the convergence difficulties of Fixed-point Iteration (FPI) schemes,

$$\boldsymbol{U}^{(n+1)} = \boldsymbol{G}(\boldsymbol{U}^{(n)}) \tag{3.1}$$

where $\boldsymbol{U} \in \mathbb{R}^N$ is the array of (primal or adjoint) unknowns and n the iteration counter. The convergence properties of such a scheme can be deduced from the magnitudes of the eigenvalues of the Jacobian matrix $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$. If all the eigenvalues lie within the unit disk, in the complex plane, the scheme is expected to converge to the solution (or fixed-point) $\boldsymbol{U}^*$. The opposite is expected to happen even if only the largest eigenvalue in magnitude exceeds unity [40]. In such a case and under certain conditions, the RPM may be able to force eq. 3.1 to converge to a solution. This is done by means of a stabilized scheme that incorporates a Newton iteration contained within the subspace formed out of the directions in which the solution diverges. Throughout this text, an important distinction is made whenever the evaluation of Jacobian matrices takes place at $\boldsymbol{U}^*$. This is because the mathematical tools used in the analysis of the convergence of operations such as eq. 3.1, as well as the properties of the stabilized scheme devised by the RPM, are valid only within an area of $\mathbb{R}^N$ that is (arbitrarily) close to $\boldsymbol{U}^*$. The "*" superscript is used to distinguish whenever claims involving the Jacobian matrices at $\boldsymbol{U}^*$ are made.

First, the method needs to approximate an orthonormal basis $V_p \in \mathbb{R}^{N \times N_p}$ for the subspace spanned by the N_p, in total, eigenvectors associated with the dominant eigenvalues of $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}^*$ i.e. the ones largest in magnitude. For the purpose of stabilization only, it is sufficient for V_p to contain only the diverging modes i.e. the ones associated with eigenvalues that are greater than unity. Two subspaces of $\mathbb{R}^N$ are defined: the invariant unstable subspace $\mathbb{P}$, for which V_p forms a basis, and its orthogonal complement $\mathbb{Q}$. The significance of the invariance property of $\mathbb{P}$ will become apparent in the following pages. For now, it is assumed that N_p is fixed and known although, within the RPM algorithm, V_p is formed incrementally during the solution of the primal/adjoint equations with N_p being initially zero and, then, growing as diverging and slowly decaying modes are gradually appended to V_p. What the method considers to be $\mathbb{P}$ and $\mathbb{Q}$ changes throughout the course of the solution. The diverging modes initially belong to $\mathbb{Q}$ but, as they are appended to V_p, $\mathbb{Q}$ shrinks whereas $\mathbb{P}$ grows instead.

Using V_p, projection matrices $P \in \mathbb{R}^{N\times N}$ and $Q \in \mathbb{R}^{N\times N}$ from $\mathbb{R}^N$ onto $\mathbb{P}$ and $\mathbb{Q}$, respectively, are defined through the following relations

$$P = V_p V_p^T, \quad Q = I - V_p V_p^T = V_q V_q^T, \quad V_q \in \mathbb{Q}^{N\times N_q} \tag{3.2}$$

where $N_q = N - N_p$, and have the following properties

$$\begin{aligned} &P = P^T, \quad P^\infty = P, \quad P + Q = I \\ &Q = Q^T, \quad Q^\infty = Q, \quad PQ = 0 \end{aligned} \tag{3.3}$$

The columns and rows of each one are orthogonal to the rows and columns of the other. Using P and Q, $\boldsymbol{U}$ is decomposed into

$$\boldsymbol{U} = \boldsymbol{U}_p + \boldsymbol{U}_q \tag{3.4}$$

where

$$\boldsymbol{U}_p = P\boldsymbol{U} \in \mathbb{P} \tag{3.5a}$$

$$\boldsymbol{U}_q = Q\boldsymbol{U} \in \mathbb{Q} \tag{3.5b}$$

namely the unstable and stable parts of the solution, respectively. P and Q are used on the r.h.s. of eq. 3.1 to yield $\boldsymbol{G}_p = P\boldsymbol{G}$ and $\boldsymbol{G}_q = Q\boldsymbol{G}$. Using this decomposition, the recursion of eq. 3.1 is split in two constituents,

$$\boldsymbol{U}_p^{(n+1)} = \boldsymbol{G}_p\left(\boldsymbol{U}_p^{(n)}, \boldsymbol{U}_q^{(n)}\right) \tag{3.6a}$$

$$\boldsymbol{U}_q^{(n+1)} = \boldsymbol{G}_q\left(\boldsymbol{U}_p^{(n)}, \boldsymbol{U}_q^{(n)}\right) \tag{3.6b}$$

The decomposition induced by eqs. 3.6a and 3.6b is the equivalent of eq. 3.1 and describes the evolution of the solution within the two subspaces. Both components, $\boldsymbol{U}_p$ and $\boldsymbol{U}_q$, are inputs to eqs. 3.6a and 3.6b but each one is now inspected separately. Equation 3.6a is responsible for the evolution of the unstable branch of the solution. If U_p is kept fixed, eq. 3.6b is able to converge. The practical outcome of this is that treatments can be applied separately. Starting from re-writing eq. 3.6a as $\boldsymbol{g}(\boldsymbol{U}_p, \boldsymbol{U}_q) = \boldsymbol{G}_p(\boldsymbol{U}_p, \boldsymbol{U}_q) - \boldsymbol{U}_p = \boldsymbol{0}$, a first order Taylor expansion is formed

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{U}_p, \boldsymbol{U}_q) &= \boldsymbol{g}(\boldsymbol{U}_p^{(n)}, \boldsymbol{U}_q) + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{U}_p}^{(n)} \Delta \boldsymbol{U}_p^{(n)} \\ &= \boldsymbol{G}_p(\boldsymbol{U}_p^{(n)}, \boldsymbol{U}_q) - \boldsymbol{U}_p^{(n)} - \left(I - \frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}^{(n)}\right)(\boldsymbol{U}_p^{(n+1)} - \boldsymbol{U}_p^{(n)}) = \boldsymbol{0} \end{aligned} \tag{3.7}$$

and the main idea behind the RPM is to substitute eq. 3.6a with a Newton-Raphson

step

$$\begin{aligned}\boldsymbol{U}_p^{(n+1)} &= \boldsymbol{N}(\boldsymbol{U}_p^{(n)}, \boldsymbol{U}_q^{(n)}) \\ &= \boldsymbol{U}_p^{(n)} + \left(I - \frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}^{(n)} \right)^{-1} \left(\boldsymbol{G}_p \left(\boldsymbol{U}_p^{(n)}, \boldsymbol{U}_q^{(n)} \right) - \boldsymbol{U}_p^{(n)} \right) \end{aligned} \tag{3.8}$$

Thus, a new stabilized scheme is derived with which eq. 3.8 is used to solve for $\boldsymbol{U}_p$ while the standard solver is retained for $\boldsymbol{U}_q$. In eq. 3.8, $\frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}$ is referred to as the restriction of $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$ to the subspace $\mathbb{P}$, and is equal to $P \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} P$. The rows and columns of this projected matrix belong to $\mathbb{P}$. It should not be viewed as the one holding the partial derivatives of $\boldsymbol{G}_p$ w.r.t. $\boldsymbol{U}_p$, derived using the chain rule, $\frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p} = P \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} \frac{\partial \boldsymbol{U}}{\partial \boldsymbol{U}_p}$ and differentiating $\boldsymbol{U}$ in eq. 3.4 w.r.t. to $\boldsymbol{U}_p$ since the latter yields $\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{U}_p} = I$. The derivatives in $\frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}$ are concerned with how the projection of $\boldsymbol{G}$ on $\mathbb{P}$ varies w.r.t. $\boldsymbol{U}$ only in directions within $\mathbb{P}$. This is more easily seen with $P \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} P = \frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}} V_p V_p^T$. The columns of $\frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}} V_p$ contain directional derivatives within $\mathbb{P}$ and V_p^T combines these columns to form $\frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}$. Variations in the components of $\boldsymbol{U}$ that are in $\mathbb{Q}$ vanish after the projection on V_p.

The original and stabilized schemes are now compared. The analysis of the convergence of the two schemes can provide insight only close to the solution $\boldsymbol{U}^*$. A first-order Taylor expansion of $\boldsymbol{G}$ about $\boldsymbol{U}^*$,

$$\boldsymbol{G}(\boldsymbol{U}^{(n)}) = \boldsymbol{G}(\boldsymbol{U}^*) + \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}^* (\boldsymbol{U}^{(n)} - \boldsymbol{U}^*) \tag{3.9}$$

recalling that $\boldsymbol{U}^* = \boldsymbol{G}(\boldsymbol{U}^*)$ at the solution, yields

$$\boldsymbol{U}^{(n+1)} - \boldsymbol{U}^* = \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}^* (\boldsymbol{U}^{(n)} - \boldsymbol{U}^*) \tag{3.10}$$

The following variables are defined to represent $\boldsymbol{U}_p$ and $\boldsymbol{U}_q$ in reference to the bases V_p and V_q, respectively,

$$\boldsymbol{\psi}_p = V_p^T \boldsymbol{U}, \qquad \boldsymbol{\psi}_p \in \mathbb{R}^{N_p} \tag{3.11}$$

$$\boldsymbol{\psi}_q = V_q^T \boldsymbol{U}, \qquad \boldsymbol{\psi}_q \in \mathbb{R}^{N_q} \tag{3.12}$$

from which $\boldsymbol{U}_p = V_p \boldsymbol{\psi}_p$ and $\boldsymbol{U}_q = V_q \boldsymbol{\psi}_q$ are retrieved. The properties in eq. 3.3 are

recalled and eq. 3.10 is left-multiplied by $\begin{bmatrix} V_p & V_q \end{bmatrix}^T$ to yield

$$\begin{aligned}
\begin{bmatrix} \Delta\boldsymbol{\psi}_p^{(n+1)} \\ \Delta\boldsymbol{\psi}_q^{(n+1)} \end{bmatrix} &= \begin{bmatrix} V_p & V_q \end{bmatrix}^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} \begin{bmatrix} V_p & V_q \end{bmatrix} \begin{bmatrix} V_p^T \\ V_q^T \end{bmatrix} \Delta\boldsymbol{U}^{(n)} \\
&= \begin{bmatrix} V_p^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_p & V_p^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_q \\ V_q^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_p & V_q^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_q \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\psi}_p^{(n)} \\ \Delta\boldsymbol{\psi}_q^{(n)} \end{bmatrix} \\
&= \begin{bmatrix} V_p^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_p & V_p^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_q \\ 0 & V_q^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_q \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\psi}_p^{(n)} \\ \Delta\boldsymbol{\psi}_q^{(n)} \end{bmatrix}
\end{aligned} \tag{3.13}$$

The top row corresponds to eq. 3.6a whereas the bottom one to eq. 3.6b. The term $V_q^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_p$ in eq. 3.13 vanishes due to $\mathbb{P}$ being an invariant subspace of $\frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}}$. This property dictates that any vector that belongs to $\mathbb{P}$ remains within this subspace after multiplication with $\frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}}$. Since each column vector of V_p (or P) resides in $\mathbb{P}$, $\frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_p$ also belongs to $\mathbb{P}$. Consequently, $V_q^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_p = 0$ (or $Q \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} P = 0$) due to the orthogonality of $\mathbb{Q}$ and $\mathbb{P}$. Subspaces formed out of eigenvectors of a matrix have this property by default and, as the diverging modes of $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$ are used in the construction of V_p, this property is ensured. This extends to generalized eigenvectors and Schur vectors of a matrix as well; the latter are used to construct V_p in the implementation of the RPM in this thesis. The elimination of the lower left block in eq. 3.13 leaves a block-diagonal system where the eigenvalues of $\frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}}$ are shared between the two diagonal blocks. Block $V_p^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_p$ is the representation of $\frac{\partial \boldsymbol{G}_p{}^*}{\partial \boldsymbol{U}_p}$ in $\mathbb{R}^{N_p \times N_p}$, contains the diverging modes of $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$ and is responsible for the divergence of $\boldsymbol{\psi}_p$ (or $\boldsymbol{U}_p$). On the other hand, the block $V_q^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_q$ contains the remaining eigenvalues and is the representation of $\frac{\partial \boldsymbol{G}_q{}^*}{\partial \boldsymbol{U}_q}$ in $\mathbb{R}^{N_q \times N_q}$. From eq. 3.13, it is apparent that keeping $\boldsymbol{\psi}_p$ fixed will cause $\boldsymbol{\psi}_q$ to converge.

Once eq. 3.8 is introduced, eq. 3.13 becomes

$$\begin{aligned}
\begin{bmatrix} \Delta\boldsymbol{\psi}_p^{(n+1)} \\ \Delta\boldsymbol{\psi}_q^{(n+1)} \end{bmatrix} &= \begin{bmatrix} I + (I - V_p^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_p)^{-1} (V_p^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_p - I) & (I - V_p^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_p)^{-1} V_p^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_q \\ 0 & V_q^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_q \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\psi}_p^{(n)} \\ \Delta\boldsymbol{\psi}_q^{(n)} \end{bmatrix} \\
&= \begin{bmatrix} 0 & (I - V_p^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_p)^{-1} V_p^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_q \\ 0 & V_q^T \frac{\partial \boldsymbol{G}^*}{\partial \boldsymbol{U}} V_q \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\psi}_p^{(n)} \\ \Delta\boldsymbol{\psi}_q^{(n)} \end{bmatrix}
\end{aligned} \tag{3.14}$$

The terms on the upper row are the projections of $\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{U}_p}^*$ and $\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{U}_q}^*$ on $\mathbb{R}^{N_p \times N_p}$ and $\mathbb{R}^{N_q \times N_p}$, respectively. The upper left block vanishes at $\boldsymbol{U}^*$ and the spectrum of the block matrix in eq. 3.14 is within the unit disk in the complex plane. From this

point, [143] shows that this result extends to a region of $\mathbb{R}^N$ close to $\boldsymbol{U}^*$, and in [83] the discussion is extended to include errors in the approximation to V_p and higher-order terms neglected in [143].

To arrive at this conclusion for the stabilized scheme, the assumption that $\mathbb{P}$ is an invariant subspace of $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}^*$ was essential. An additional assumption is that $\left(I - \frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}^*\right)$ has an inverse. This is not the case if an eigenvalue of $\frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}^*$ lies on the unit circle. This happens at so-called bifurcation points [16] where the steady state switches from stable to unstable or vice versa, such as when due to an increase in the Reynolds number, the flow around a cylinder switches from steady to periodic. Under these assumptions, eqs. 3.6b and 3.8 combined can converge to $\boldsymbol{U}^*$, even under circumstances in which the original scheme diverges.

Equation 3.8 is derived using an expansion of $\boldsymbol{g}$ in eq. 3.7 solely about $\boldsymbol{U}_p$. If an expansion about $\boldsymbol{U}_p$ and $\boldsymbol{U}_q$ is used instead,

$$\boldsymbol{U}_p^{(n+1)} = \boldsymbol{U}_p^{(n)} + \left(I - \frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}^{(n)}\right)^{-1} \left(\boldsymbol{G}_p\left(\boldsymbol{U}_p^{(n)}, \boldsymbol{U}_q^{(n)}\right) - \boldsymbol{U}_p^{(n)} + \frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_q}^{(n)} \Delta \boldsymbol{U}_q\right) \tag{3.15}$$

an equation to better estimate $\boldsymbol{U}_p^{(n+1)}$ can be derived. The last term on the r.h.s. of eq. 3.15, $\frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_q} \Delta \boldsymbol{U}_q$, is expensive to compute. The computation might be overcome by substituting the following FD scheme,

$$\frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_q} \Delta \boldsymbol{U}_q = \boldsymbol{G}_p(\boldsymbol{U}_p, \boldsymbol{U}_q + \Delta \boldsymbol{U}_q) - \boldsymbol{G}_p(\boldsymbol{U}_p, \boldsymbol{U}_q) \tag{3.16}$$

in eq. 3.15 and arriving at

$$\boldsymbol{U}_p^{(n+1)} = \boldsymbol{U}_p^{(n)} + \left(I - \frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}^{(n)}\right)^{-1} \left(\boldsymbol{G}_p(\boldsymbol{U}_p^{(n)}, \boldsymbol{U}_q^{(n)} + \Delta \boldsymbol{U}_q) - \boldsymbol{U}_p^{(n)}\right) \tag{3.17}$$

The term on the r.h.s. of eq. 3.17 is evaluated by keeping $\boldsymbol{U}_p$ fixed and performing iterations restricted inside $\mathbb{Q}$. In fact, the number of iterations that can be used might be greater than one, providing an updated estimate for $\boldsymbol{U}_q$ to be used by the $\mathbb{P}$-contained Newton iteration for improved convergence rate.

3.1.1 Cost Reduction of the Newton Iteration

In eq. 3.8, the inversion of $\left(I - \frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}\right)$ costs as much as performing the recursion in eq. 3.1. Because the Newton iteration, eq. 3.8, is contained within $\mathbb{P}$ and N_p is usually small compared to the size of the problem at hand, the cost of this inversion can be made negligible by performing all linear operations in $\mathbb{R}^{N_p}$ using

$\boldsymbol{\psi}_p$. Thus, eq. 3.8 becomes

$$\boldsymbol{\psi}_p^{(n+1)} = \boldsymbol{\psi}_p^{(n)} + (I - H_p)^{-1} \left(V_p^T \boldsymbol{G}\left(\boldsymbol{U}^{(n)}\right) - \boldsymbol{\psi}_p^{(n)} \right) \tag{3.18}$$

where

$$H_p = V_p^T \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} V_p, \;\; H_p \in \mathbb{R}^{N_p \times N_p} \tag{3.19}$$

Inverting $(I - H_p)$ in eq. 3.18 is trivial in case $N_p \ll N$. Once the new $\boldsymbol{\psi}_p$ iterate is computed, $\boldsymbol{U}_p$ is retrieved through $\boldsymbol{U}_p = V_p \boldsymbol{\psi}_p$. The computation of H_p is done in two steps. Directional derivatives of $\boldsymbol{G}$ w.r.t. $\boldsymbol{U}$ are computed first in a matrix-free fashion using first- or second-order accurate FDs,

$$\left(\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} V_p \right)_i = \frac{\boldsymbol{G}(\boldsymbol{U}^{(n)} + \varepsilon V_{p,i}) - \boldsymbol{G}(\boldsymbol{U}^{(n)})}{\varepsilon} \tag{3.20a}$$

$$\left(\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} V_p \right)_i = \frac{\boldsymbol{G}(\boldsymbol{U}^{(n)} + \varepsilon V_{p,i}) - \boldsymbol{G}(\boldsymbol{U}^{(n)} - \varepsilon V_{p,i})}{2\varepsilon} \tag{3.20b}$$

The resulting matrix $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} V_p$ is left-multiplied by V_p^T to yield H_p. Each time V_p is computed/updated, H_p must be computed anew and is used for the remainder of the solution until V_p changes again. An LU decomposition of $I - H_p$ is thus used to solve eq. 3.18. The step size ε is chosen following the formulas in [5]. The use of FDs cannot be circumvented because, as it will be seen in section 3.1.3, $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$ is not available and matrix-vector products cannot be computed through typical multiplication.

3.1.2 Basis Construction

Until this point, V_p was assumed to be known though, in practice, the sought eigenspace must be computed. There are numerous numerical techniques for approximating invariant subspaces of a matrix in the literature [40]. Most usually, a variant of the power method is used, wherein a series of matrix-vector products are formed between $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$ and a suitable initial estimation of V_p (e.g. in [83], subspace iterations are used to approximate V_p). After enough iterations, V_p converges to the dominant eigenspace of $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$. In [143], the computation of these matrix-vector products is avoided by using the $\boldsymbol{U}_q$ iterates. After a Taylor expansion of $\boldsymbol{G}_q$ around $\boldsymbol{U}_p$ and $\boldsymbol{U}_q$, with second-order terms neglected,

$$\begin{aligned} \boldsymbol{G}_q(\boldsymbol{U}_p^{(n-1)} + \Delta \boldsymbol{U}_q^{(n-1)}, \boldsymbol{U}_p^{(n-1)} + \Delta \boldsymbol{U}_p^{(n-1)}) - \boldsymbol{G}_q(\boldsymbol{U}_p^{(n-1)}, \boldsymbol{U}_p^{(n-1)}) \\ = \frac{\partial \boldsymbol{G}_q}{\partial \boldsymbol{U}_q} \Delta \boldsymbol{U}_q^{(n-1)} + \frac{\partial \boldsymbol{G}_q}{\partial \boldsymbol{U}_p} \Delta \boldsymbol{U}_p^{(n-1)} \end{aligned} \tag{3.21}$$

the second term on the r.h.s. of eq. 3.21 vanishes. By denoting

$$\begin{aligned}\Delta \boldsymbol{U}_q^{(n)} &= \boldsymbol{G}_q(\boldsymbol{U}_p^{(n-1)} + \Delta \boldsymbol{U}_q^{(n-1)}, \boldsymbol{U}_p^{(n-1)} + \Delta \boldsymbol{U}_p^{(n-1)}) - \boldsymbol{G}_q(\boldsymbol{U}_p^{(n-1)}, \boldsymbol{U}_p^{(n-1)}) \\ &= \boldsymbol{U}_q^{(n+1)} - \boldsymbol{U}_q^{(n)}\end{aligned} \tag{3.22}$$

the recursion of eq. 3.21 unrolls to

$$\Delta \boldsymbol{U}_q^{(n)} \approx \left(\frac{\partial \boldsymbol{G}_q}{\partial \boldsymbol{U}_q}\right)^n \Delta \boldsymbol{U}_q^{(0)} \tag{3.23}$$

Thus, $\Delta \boldsymbol{U}_q^{(n)}$ results from powers of matrix $\frac{\partial \boldsymbol{G}_q}{\partial \boldsymbol{U}_q}$ multiplying $\Delta \boldsymbol{U}_q^{(0)}$ and recursively approximates the dominant eigenvector of $\frac{\partial \boldsymbol{G}_q}{\partial \boldsymbol{U}_q}$. A number of consecutive $\boldsymbol{U}_q$ iterates is used to form the matrix $D \in \mathbb{R}^{N \times N_q}$ where

$$D = \left[\Delta \boldsymbol{U}_q^{(n)} \quad \Delta \boldsymbol{U}_q^{(n-1)} \quad \dots \quad \Delta \boldsymbol{U}_q^{(n-N_q)}\right] \tag{3.24}$$

D spans the dominant eigenspace of $\frac{\partial \boldsymbol{G}_q}{\partial \boldsymbol{U}_q}$ which corresponds to the eigenspace of $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$ that has not been included in $\mathbb{P}$. Before V_p is formed, $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$ and $\frac{\partial \boldsymbol{G}_q}{\partial \boldsymbol{U}_q}$ coincide. Once the subspace that contains diverging modes is approximated and completed, the algorithm locks onto slowly decaying ones. Letting the recursion in eq. 3.6b advance $\boldsymbol{U}_q$ and yield D as a by-product of the solution process itself is how the RPM avoids the computation of matrix-vector products in the construction of V_p.

Assuming convergence of $\Delta \boldsymbol{U}_q$ in eq. 3.23, the column vectors of D are not expected to be linearly independent. A Rank-revealing QR factorization [15] of D follows. The purpose of such a factorization is to reveal the rank of a matrix by producing an R factor that has diagonal elements in descending order of magnitude. A gap in magnitude between two consecutive diagonal elements emerges and indicates the point beyond which the orthogonalization produces almost zero vectors (in floating point arithmetic) due to linear dependency of the matrix's columns. This factorization is produced here using a row-oriented Modified Gram-Schmidt procedure with column pivoting [40]. In practice, the diagonal elements of the R factor are inspected and, if $R_{i,i} > \mathcal{C}_K R_{i+1,i+1}$, then the first i columns of the Q factor are appended to V_p. $\mathcal{C}_K$ is referred to as the Krylov acceptance criterion and is a user-defined parameter. Once the basis is updated, a number of smoothing iterations, based on the stabilized scheme, are performed before the check is performed again. This allows the recursion in eq. 3.23 to converge, otherwise new additions to V_p will most likely be inaccurate.

Solving the Eigenvalue Problem for H_p may provide some useful insight for refining V_p with a projection step, described in what follows. Each eigenvector/value pair $(\boldsymbol{z}, \lambda)$ of H_p corresponds to a pair $(V_p \boldsymbol{z}, \lambda)$ of $\frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}$. Instead of eigenvectors, Schur vectors $\boldsymbol{s}$ (for H_p) and $V_p \boldsymbol{s}$ (for $\frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}$) can be used due to

their increased numerical stability and lesser sensitivity to numerical errors [40]. The real Schur decomposition of H_p,

$$H_p = SYS^T \tag{3.25}$$

is computed where $S \in \mathbb{R}^{m\times m}$ is an orthogonal matrix whose column vectors are the $\boldsymbol{s}$. Equation 3.25 transforms H_p to a quasi-upper triangular block form Y, i.e. the Schur form of H_p, where 1-by-1 or 2-by-2 matrix blocks are produced for real or pairs of imaginary eigenvalues, respectively. The eigenvalues of these blocks are the same as the ones of H_p and can easily be computed. The decomposition in eq. 3.25 is performed using tools provided by OpenFOAM which make use of the QR algorithm, the implementation for which is loosely based on [162]. For each pair $(\boldsymbol{s}, \lambda)$, two criteria are used to refine V_p: First, the accuracy of V_p can be evaluated using the expression $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} V_p \boldsymbol{z} = \lambda V_p \boldsymbol{z}$. This checks whether V_p is invariant. If not, the basis suffers from low accuracy. This could be the case due to errors introduced from the FDs in eqs. 3.20 or because the sequence in eq. 3.23 might not have sufficiently converged. Secondly, it is of importance to keep the size of $\mathbb{P}$ as low as possible, not to damage the performance of the RPM. Therefore, it might be preferable to exclude modes that correspond to eigenvalues that do not cause divergence but were picked up by the algorithm nonetheless. The following checks

$$\left\| \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} V_p \boldsymbol{z} - \lambda V_p \boldsymbol{z} \right\| < \mathcal{C}_1, \quad \|\lambda\| > \mathcal{C}_2 \tag{3.26}$$

can be performed simultaneously, where $\mathcal{C}_{1,2}$ are user-defined thresholds. For the eigenpairs that do not meet these criteria, their corresponding Schur vectors are extracted from S. After this, with the remaining $\boldsymbol{N}_\lambda$ ones, $S \in \mathbb{R}^{m\times N_\lambda}$ and the remaining Schur vectors of $\frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}$ are used to substitute the basis by $V_p \leftarrow V_p S$.

A clear problem of the above process is the slow convergence of the power method in eq. 3.23. The rate of convergence of power iterations strongly relies on how well-separated the eigenvalues of $\frac{\partial \boldsymbol{G}_q}{\partial \boldsymbol{U}_q}$ are. This is clearly described in [40] (pp 357–363). When many diverging modes exist, the eigenvalues are usually clustered around the edges of the unit circle in the complex plane and the power method cannot provide an accurate representation of V_p, a situation reported in [2] with the RPM. The rate deteriorates even further if the initial V_p estimate is not "rich" in the directions of the dominant eigenmodes in $\mathbb{R}^N$. Apart from reduced solver performance, caused by a greater number of smoothing iterations that take place between consecutive basis updates, the slow convergence of eq. 3.23 may prevent the RPM from treating stability due to an inaccurate V_p. An alternative to eq. 3.23 is to use subspace iterations with projection, see Algorithm 1. In this case, the initial estimate for V_p can be the matrix D or a number of $\boldsymbol{U}$ snapshots. With this algorithm, a greater number of modes can be computed at once however, the iterations of Algorithm 1 can still be slow to converge for the same reasons as the ones in eq. 3.23. A more sophisticated

version of this algorithm, which used various techniques that accelerated its convergence, was the method of choice in [83]. However, in that case, the Jacobian matrix was available within the code. Since $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$ is never entirely available in the solution schemes employed in this thesis, matrix-vector products in Algorithm 1 need to be performed using FDs, with a significant toll on the performance. In some more difficult cases, a few subspace iterations were optionally used on the modes computed through eq. 3.21, to improve the accuracy of the basis and enhance the convergence of the stabilized scheme.

Algorithm 1 Subspace iterations with projection

1: Compute $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} V_p$
2: **repeat**
3: Orthonormalize $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$ with Modified Gram-Schmidt
4: $V_p \leftarrow \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} V_p$
5: $H_p \leftarrow V_p^T \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} V_p$
6: Compute $H_p = SYS^T$ and order eigenvalues in descending order
7: $V_p \leftarrow V_p S$, $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} V_p \leftarrow \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} V_p S$
8: **until** *convergence of all* N_λ λ

The overall RPM implementation is shown in Algorithm 2.

Algorithm 2 RPM

1: **repeat**
2: $\boldsymbol{U}^{\dagger} \leftarrow \boldsymbol{G}(\boldsymbol{U}^{(n)})$ (eq. 3.1)
3: RRQR of D (eq. 3.24)
4: **if** $R_{i,i} > \mathcal{C}_K R_{i+1,i+1}$ **then** update V_p
5: $N_p \leftarrow size(V_p)$
6: Compute $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} V_p$ (eqs. 3.20)
7: $H_p \leftarrow V_p^T \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} V_p$
8: $H_p = SYS^T$
9: $V_p \leftarrow V_p S$ based on criteria in eq. 3.26
10: **end if**
11: **if** $N_p > 0$ **then** use stabilized scheme:
12: $\boldsymbol{\psi}_p^{(n)} \leftarrow V_p^T \boldsymbol{U}^{(n)}$
13: $\boldsymbol{U}_q^{(n+1)} \leftarrow \boldsymbol{U}^{\dagger} - V_p V_p^T \boldsymbol{U}^{\dagger}$
14: **for** $l = 2, O_{RPM}$ **do**
15: $\boldsymbol{U}^{\dagger} \leftarrow \boldsymbol{G}(V_p \boldsymbol{\psi}_p^{(n)}, \boldsymbol{U}_q^{(n+1)})$
16: $\boldsymbol{U}_q^{(n+1)} \leftarrow \boldsymbol{U}^{\dagger} - V_p V_p^T \boldsymbol{U}^{\dagger}$
17: **end for**
18: $\boldsymbol{\psi}_p^{(n+1)} \leftarrow \boldsymbol{\psi}_p^{(n)} + (I - H_p)^{-1} \left(V_p^T \boldsymbol{U}^{\dagger} - \boldsymbol{\psi}_p^{(n)} \right)$
19: $\boldsymbol{U}_p^{(n+1)} \leftarrow V_p \boldsymbol{\psi}_p^{(n+1)}$
20: $\boldsymbol{U}^{(n+1)} \leftarrow \boldsymbol{U}_p^{(n+1)} + \boldsymbol{U}_q^{(n+1)}$
21: **else**
22: $\boldsymbol{U}^{(n+1)} \leftarrow \boldsymbol{U}^{\dagger}$
23: **end if**
24: $n = n + 1$
25: **until** *convergence*

3.1.3 The SIMPLE Algorithm and the RPM

The primal problem is governed by the incompressible Navier-Stokes equations. The main difficulty in solving these equations is the absence of an equation with p as its primary variable; eqs. 2.1b are solved for the components of $\boldsymbol{v}$ and eq. 2.1a serves as a kinematic constraint for $\boldsymbol{v}$. This constraint must be satisfied by computing an appropriate p field. To do this, the flow solver in OpenFOAM makes use of the SIMPLE algorithm [122] in which a type of Poisson equation is derived to be solved for p right after the solution of eqs. 2.1b and the computation of an intermediate velocity field. The new pressure field is used to correct the velocity field and satisfy eq. 2.1a. The adjoint equations use a similar algorithm. In this section, the SIMPLE algorithm employed by OpenFOAM is presented and brought into the form of eq. 3.1. With respect to the primal equations, the RPM is concerned only with eqs. 2.1a to 2.1c whereas eq. 2.1d is

solved at a pre-processing step, within each optimization cycle, and is completely decoupled from the rest.

The Navier-Stokes equations, in a fully discretized form, are

$$\begin{bmatrix} \mathcal{M} & -\mathcal{G} \\ \mathcal{D} & \emptyset \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{v}} \\ \overline{\boldsymbol{p}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ 0 \end{bmatrix}, \quad \overline{\boldsymbol{v}} = \begin{bmatrix} \overline{\boldsymbol{v}}_1 \\ \overline{\boldsymbol{v}}_2 \\ \overline{\boldsymbol{v}}_3 \end{bmatrix} \tag{3.27}$$

where $\mathcal{M}$ is a matrix that encompasses the discretization of the convective and diffusive terms in eqs. 2.1b, $\mathcal{G}$ and $\mathcal{D}$ are the discretization of the spatial gradient of p and the divergence operator for v, respectively. The r.h.s. $\boldsymbol{b}$ of the momentum equations encompasses terms that are treated explicitly as source terms, such as the non-orthogonal correction term of the velocity gradient or the transposed velocity gradient in the discretization of the shear stress tensor. The overbars in $\overline{\boldsymbol{v}}_{1-3}$ and $\overline{\boldsymbol{p}}$ denote the discretized fields of the velocity components and pressure. These are now all vectors whose components are stored at the cell centers of the finite volume elements. Due to the non-linear nature of eqs. 2.1b, the solution of eq. 3.27 is iterative. The linearization is performed about the current iterate which is used to construct $\mathcal{M}$. For example, the discretization of the convection term in eq. 2.1b over a finite volume V yields,

$$\int_V \frac{\partial v_j v_i}{\partial x_j} dV = \int_S v_j n_j v_i dS = \sum_{f=1}^{\#faces} \overline{\phi}_f^{(n)} \overline{v}_{i,f}^{(n+1)} \tag{3.28}$$

where $\overline{\phi}$ are the fluid volume fluxes, passing through and stored at each cell face f. $\overline{\phi}$ implicitly depends on $\overline{\boldsymbol{v}}$ and $\overline{\boldsymbol{p}}$ of the current iteration and the volume fluxes of the previous one. The summation of $\overline{\phi}$ over the faces should be equal to zero (to a tolerance) after solving the Poisson equation for $\overline{\boldsymbol{p}}$, thus satisfying the discrete form of the continuity equation. $\mathcal{M}$ is a lower block triangular, rather than block diagonal (one diagonal block for each $\overline{\boldsymbol{v}}$ coordinate component) matrix because once $\overline{\boldsymbol{v}}_1$ is computed, it is used in the computation of $\overline{\boldsymbol{v}}_2$ and both of them for $\overline{\boldsymbol{v}}_3$. The zero lower-right block on the left-hand-side of eq. 3.27 makes the iterative solution of this system difficult.

Strung together, all the unknown variables form $\boldsymbol{U} = \begin{bmatrix} \overline{\boldsymbol{v}}^T & \overline{\boldsymbol{p}}^T & \overline{\phi}^T \end{bmatrix}^T$ in eq. 3.1. Following the direct application of the RPM to DAEs in [73], the algebraic variable $\overline{\phi}$ is included in $\boldsymbol{U}$ as an additional unknown. If omitted from $\boldsymbol{U}$ and treated as a dependent variable, it should be adjusted consistently whenever the RPM manipulates $\boldsymbol{U}$, such as during the computation of FDs with eq. 3.20. With the approach followed in this thesis, this step is avoided. The discrete $\widetilde{\nu}$ field should be included here as well but, since turbulence model equations are solved after

the solution of eqs. 2.1a and 2.1b, it is omitted to simplify the analysis.

The momentum equations are solved using $\overline{\boldsymbol{p}}^{(n)}$ and $\overline{\phi}^{(n)}$ and the momentum predictor, $\overline{\boldsymbol{v}}^{\dagger}$, is computed. This intermediate velocity field satisfies the momentum but is not divergence-free. Let $\overline{\boldsymbol{v}}'$, $\overline{\boldsymbol{p}}'$ and $\overline{\phi}'$ be corrections that need to be added to $\overline{\boldsymbol{v}}^{\dagger}$, $\overline{\boldsymbol{p}}^{(n)}$ and $\overline{\phi}^{\dagger}$, respectively, in order for them to satisfy both the momentum and continuity equations,

$$\overline{\boldsymbol{v}}^{(n+1)} = \overline{\boldsymbol{v}}^{\dagger} + \overline{\boldsymbol{v}}'^{(n)} \tag{3.29a}$$

$$\overline{\boldsymbol{p}}^{(n+1)} = \overline{\boldsymbol{p}}^{(n)} + \overline{\boldsymbol{p}}'^{(n)} \tag{3.29b}$$

$$\overline{\phi}^{(n+1)} = \overline{\phi}^{\dagger} + \overline{\phi}'^{(n)} \tag{3.29c}$$

In the original SIMPLE algorithm, an equation is derived with which $\overline{\boldsymbol{p}}'$, as well as any other correction, are computed. Using eq. 3.29b, eq. 3.27 is written in terms of $\overline{\boldsymbol{v}}^{(n+1)}$ and $\overline{\boldsymbol{p}}'^{(n)}$,

$$\begin{bmatrix} \mathcal{M} & \emptyset \\ \mathcal{D} & -\mathcal{S} \end{bmatrix} \begin{bmatrix} \mathcal{I} & \mathcal{M}^{-1}\mathcal{G} \\ \emptyset & \mathcal{I} \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{v}}^{(n+1)} \\ \overline{\boldsymbol{p}}'^{(n)} \end{bmatrix} = \begin{bmatrix} \mathcal{G}\overline{\boldsymbol{p}}^{(n)} + \boldsymbol{b} \\ \emptyset \end{bmatrix} \tag{3.30}$$

where $\mathcal{S} = \mathcal{D}\mathcal{M}^{-1}\mathcal{G}$. In practice, the full computation of $\mathcal{S}$ is prohibitive. Various approximations can be used instead in SIMPLE and its variants such as the SIMPLEC algorithm. Equation 3.30 is essentially an LU decomposition of eq. 3.27, in which the lower block triangular system is solved first

$$\begin{bmatrix} \mathcal{M} & \emptyset \\ \mathcal{D} & -\mathcal{S} \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{v}}^{\dagger} \\ \overline{\boldsymbol{p}}'^{(n)} \end{bmatrix} = \begin{bmatrix} \mathcal{G}\overline{\boldsymbol{p}}^{(n)} + \boldsymbol{b} \\ \emptyset \end{bmatrix} \tag{3.31}$$

followed by the solution of the upper block triangular system

$$\begin{bmatrix} \mathcal{I} & \mathcal{M}^{-1}\mathcal{G} \\ \emptyset & \mathcal{I} \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{v}}^{(n+1)} \\ \overline{\boldsymbol{p}}'^{(n)} \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{v}}^{\dagger} \\ \overline{\boldsymbol{p}}'^{(n)} \end{bmatrix} \tag{3.32}$$

These two steps are described in more detail.

First, the momentum equations are written by splitting $\mathcal{M}$ in a diagonal component $\mathcal{A}$ and denoting

$$\boldsymbol{H}(\overline{\boldsymbol{v}}) = (\mathcal{A} - \mathcal{M})\overline{\boldsymbol{v}} \tag{3.33}$$

the off-diagonal part multiplying $\overline{\boldsymbol{v}}$. Using eq. 3.33, the discretized momentum equations are

$$\mathcal{A}\overline{\boldsymbol{v}} - \boldsymbol{H}(\overline{\boldsymbol{v}}) = \mathcal{G}\overline{\boldsymbol{p}} + \boldsymbol{b} \tag{3.34}$$

and $\overline{\boldsymbol{v}}$ is expressed as

$$\overline{\boldsymbol{v}} = \mathcal{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{v}}) - \mathcal{A}^{-1}\mathcal{G}\overline{\boldsymbol{p}} + \mathcal{A}^{-1}\boldsymbol{b} \tag{3.35}$$

At each iteration of the flow solver, the momentum equations are solved for $\overline{\boldsymbol{v}}^{\dagger}$ first

$$\mathcal{M}\overline{\boldsymbol{v}}^{\dagger} = -\mathcal{G}\overline{\boldsymbol{p}}^{(n)} + \boldsymbol{b}^{(n)} \tag{3.36}$$

Obviously, $\overline{\boldsymbol{v}}^{\dagger} = \overline{\boldsymbol{v}}^{\dagger}(\overline{\boldsymbol{v}}^{(n)}, \overline{\boldsymbol{p}}^{(n)}, \overline{\boldsymbol{\phi}}^{(n)}) \equiv \overline{\boldsymbol{v}}^{\dagger}(\boldsymbol{U}^{(n)})$ where the old iterates $\overline{\boldsymbol{v}}^{(n)}$ and $\overline{\boldsymbol{\phi}}^{(n)}$ contribute to terms in $\mathcal{M}$ or $\boldsymbol{b}$. In order to derive a relation between $\overline{\boldsymbol{v}}'$ and $\overline{\boldsymbol{p}}'$, eq. 3.35 is written for $\overline{\boldsymbol{v}}^{\dagger}$ and $\overline{\boldsymbol{v}}^{(n+1)}$,

$$\overline{\boldsymbol{v}}^{\dagger} = \mathcal{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{v}}^{\dagger}) - \mathcal{A}^{-1}\mathcal{G}\overline{\boldsymbol{p}}^{(n)} + \boldsymbol{b}^{(n)} \tag{3.37a}$$

$$\overline{\boldsymbol{v}}^{(n+1)} = \mathcal{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{v}}^{(n+1)}) - \mathcal{A}^{-1}\mathcal{G}\overline{\boldsymbol{p}}^{(n+1)} + \boldsymbol{b}^{(n+1)} \tag{3.37b}$$

and eq. 3.37a is subtracted from eq. 3.37b. An assumption is made that $\mathcal{M}$, $\mathcal{A}$ and $\boldsymbol{b}$ do not change significantly from iteration to iteration. Additionally, the difference between $\boldsymbol{H}(\overline{\boldsymbol{v}}^{(n+1)})$ and $\boldsymbol{H}(\overline{\boldsymbol{v}}^{\dagger})$ is assumed to be negligible compared to the pressure correction term. The result is

$$\overline{\boldsymbol{v}}' = -\mathcal{A}^{-1}\mathcal{G}\overline{\boldsymbol{p}}' \tag{3.38}$$

Next, the divergence operator is applied to eq. 3.29c requiring the net summation of all fluxes passing from the faces of a cell to be zero, $\mathcal{D}\overline{\boldsymbol{\phi}}^{(n+1)} = 0$, in order to satisfy the continuity equation. The substitution of eqs. 3.37a and 3.38 in the divergence of eq. 3.29c leads to

$$\begin{aligned}
\mathcal{D}\overline{\boldsymbol{\phi}}^{(n+1)} &= \mathcal{D}\overline{\boldsymbol{\phi}}^{\dagger} + \mathcal{D}\overline{\boldsymbol{\phi}}'^{(n)} \\
&= \mathcal{D}\mathcal{T}\overline{\boldsymbol{v}}^{\dagger} + \mathcal{D}\mathcal{T}\overline{\boldsymbol{v}}'^{(n)} \\
&= \mathcal{D}\mathcal{T}\mathcal{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{v}}^{\dagger}) - \mathcal{D}\mathcal{T}\mathcal{A}^{-1}\mathcal{G}\overline{\boldsymbol{p}}^{(n)} - \mathcal{D}\mathcal{T}\mathcal{A}^{-1}\mathcal{G}\overline{\boldsymbol{p}}'^{(n)} \\
&= \mathcal{D}\mathcal{T}\mathcal{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{v}}^{\dagger}) - \mathcal{D}\mathcal{T}\mathcal{A}^{-1}\mathcal{G}\overline{\boldsymbol{p}}^{(n+1)}
\end{aligned} \tag{3.39}$$

where $\mathcal{T}$ is a matrix representing the act of interpolating variables stored at the cell centers to faces. The equation to be solved for $\overline{\boldsymbol{p}}^{(n+1)}$ is

$$\widehat{\mathcal{S}}\,\overline{\boldsymbol{p}}^{(n+1)} = \mathcal{D}\mathcal{T}\mathcal{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{v}}^{\dagger}) \tag{3.40}$$

where $\widehat{\mathcal{S}} = \mathcal{D}\mathcal{T}\mathcal{A}^{-1}\mathcal{G}$ and the inverse of $\mathcal{M}$ has been approximated by $\mathcal{A}^{-1}$. This is the discrete form of the Poisson equation derived for p.[1] $\mathcal{T}$ can be chosen so as to implement the Rhie-Chow interpolation [131] and couple the $\overline{\boldsymbol{p}}$ values of cells

[1]The semi-discretized form would lead to a Poisson-type equation: $\nabla\cdot\left(\frac{1}{\alpha_P}\nabla p\right)$ where α_P denotes the diagonal coefficients of $\mathcal{M}$ for each cell-center P.

straddling a face. More details on how this is implemented within OpenFOAM can be found in [94]. The computation of $\overline{\boldsymbol{p}}^{(n+1)}$ is followed by the update of the velocity field

$$\overline{\boldsymbol{v}}^{(n+1)} = \mathcal{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{v}}^{\dagger}) - \mathcal{A}^{-1}\mathcal{G}\overline{\boldsymbol{p}}^{(n+1)} \tag{3.41}$$

and the volume fluxes

$$\overline{\phi}^{(n+1)} = \mathcal{T}\mathcal{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{v}}^{\dagger}) - \mathcal{T}\mathcal{A}^{-1}\mathcal{G}\overline{\boldsymbol{p}}^{(n+1)} \tag{3.42}$$

Once this is done, $\mathcal{M}$ is assembled again and the iteration begins anew. Until convergence is achieved, all the linear systems are solved using lenient tolerances. Strict tolerances decrease performance and are of no use because upon convergence, all equations will be satisfied either way. An exception is made whenever directional FDs are computed using eqs. 3.20.

The update of all variables can be written explicitly in terms of $\boldsymbol{U}^{(n)}$. To do this, $\overline{\boldsymbol{v}}^{\dagger} = \overline{\boldsymbol{v}}^{\dagger}(\boldsymbol{U}^{(n)})$ is used and all the $(n+1)$ iterates are substituted in eqs. 3.40 to 3.42, leading to

$$\overline{\boldsymbol{v}}^{(n+1)} = \mathcal{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{v}}^{\dagger}) - \mathcal{A}^{-1}\mathcal{G}\widehat{\mathcal{S}}^{-1}\mathcal{D}\mathcal{T}\mathcal{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{v}}^{\dagger}) \qquad = \boldsymbol{G}_1(\boldsymbol{U}^{(n)}) \tag{3.43a}$$

$$\overline{\boldsymbol{p}}^{(n+1)} = \widehat{\mathcal{S}}^{-1}\mathcal{D}\mathcal{T}\mathcal{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{v}}^{\dagger}) \qquad = \boldsymbol{G}_2(\boldsymbol{U}^{(n)}) \tag{3.43b}$$

$$\overline{\phi}^{(n+1)} = \mathcal{T}\mathcal{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{v}}^{\dagger}) - \mathcal{T}\mathcal{A}^{-1}\mathcal{G}\widehat{\mathcal{S}}^{-1}\mathcal{D}\mathcal{T}\mathcal{A}^{-1}\boldsymbol{H}(\overline{\boldsymbol{v}}^{\dagger}) \qquad = \boldsymbol{G}_3(\boldsymbol{U}^{(n)}) \tag{3.43c}$$

Equations 3.43a to 3.43c bring the SIMPLE algorithm into the form of eq. 3.1 with $\boldsymbol{G} = \begin{bmatrix}\boldsymbol{G}_1^T & \boldsymbol{G}_2^T & \boldsymbol{G}_3^T\end{bmatrix}^T$. Any particular approximations made by different implementations of the algorithm can be included by modifying each one of the matrices involved in eqs. 3.43. In practice, the inverse matrices in eqs. 3.43 cannot be computed and the solution procedure follows the segregated approach. For this reason, the Jacobian matrix $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$ is not available and the FD schemes in eqs. 3.20 are used to compute H_p. A similar algorithm is used for the solution of the adjoint equations and its analysis leads to similar considerations.

The RPM has been developed as a wrapper around the solvers of the *adjointOptimisation* library of OpenFOAM. Depending on the case, the RPM could be used in the numerical solution of the primal and/or adjoint PDEs, excluding eqs. 2.1d and 2.18d which are decoupled from the rest and, usually, their solution does not face convergence issues.

3.1.4 Flow-Field Averaging and Damping of the ATC Term

In complex applications, either exhibiting unsteadiness or not, the RPM may not always be sufficient to stabilize the primal or the adjoint solver on its own. The main cause for this is the difficulty in forming accurate approximations to V_p. Apart from the problem of clustered eigenvalues mentioned in section 3.1.2, the

mode-tracking algorithm of the RPM in eq. 3.21 relies on the evolution of the equations to gradually form V_p. If the eigenvalues causing divergence are too high in number or too large in value, which is often encountered in the solution of the adjoint equations in complex cases, the equations diverge rapidly leaving no room for the RPM to stabilize them. Also, due to the non-linearity of the flow equations, $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$ changes from one iteration to the next, thus, upon stagnation of the flow equations' residuals, eq. 3.21 may never adequately converge due to an ever-changing matrix on its r.h.s..

Whenever the flow equations cannot converge, despite the assistance of the RPM, averaging over a user-defined number of iterations gives the "pseudo-steady" primal fields about which the adjoint equations are linearized. This is considered to yield a better solution estimate overall, rather than just using the last snapshot of the primal solver. This use of either an averaged flow field or the last solution snapshot of the non-converged flow solver breaks the original assumption made in eq. 2.9 regarding the flow equations' residuals.

In case divergence occurs in the solution of the adjoint equations, despite the use of the RPM, the ATC term requires damping. This is implemented by using a field, $L = L(\boldsymbol{x})$, that multiplies the ATC term in eq. 2.18b. To cope with divergence issues, L is set to 0 at the first cell off the solid boundaries, whereas $L = 1$ everywhere else. Then, a few smoothing iterations create a smooth transition from 0 to 1. Smoothing is done by interpolating the values of L from the cell centers to the faces of finite volumes and, then, assigning the average of face values to the cell centers. 2 to 5 smoothing iterations are typically enough, meaning that only the first 2 to 5 rows of cells off the walls are affected. This treatment secures enough iterations for the RPM to form the unstable subspace. The role of the RPM still remains pivotal because, without it, in cases exhibiting unsteadiness, excessive ATC damping would be required in order to stabilize the steady adjoint equations.

The impact of flow averaging and ATC damping on the SDs is assessed in the cases that follow.

3.2 Stabilization of Steady Solvers for 2D Shape Optimization

Shape optimization was performed for two cases involving 2D vortex shedding flows. Flow unsteadiness hindered the convergence of the steady primal and adjoint solvers and, in both cases, the RPM was used to stabilize them. In these cases, from a physical standpoint, the periodic flow is the one naturally occurring. The steady flow field on the other hand, described by the steady-state Navier-Stokes equations and obtained numerically with the help of the RPM, represents an equilibrium state of the flow. Such a steady-state model proves to have great value in supporting an optimization loop at a lower cost.

3.2.1 Shape Optimization of a 2D Cylinder

The flow around a circular cylinder undergoes a Hopf bifurcation and switches from steady to periodic at $Re \approx 47$ [33]. The first case is concerned with the minimization of the drag coefficient (C_D) of such a cylinder at $Re = 140$ which corresponds to a periodic flow. Formally, in order to optimize the cylinder shape by adequately modelling the physics of the flow, it would be necessary to use an unsteady flow solver, average C_D over the (generally, unknown) period of vortex shedding and use an unsteady adjoint solver. Since, though, the optimization method is expected to minimize the body's frontal area by squeezing it in the transversal to the flow direction, in order to suppress vortex shedding, after the first few cycles, the flow around the modified shape will become steady, justifying the use of steady (primal and adjoint) solvers.

The flow and adjoint equations were solved on a grid with 46080 quadrilateral elements; a VBS morphing box was used for the parameterization, both shown in fig. 3.1. An equality constraint on the body area/volume V (to retain its initial

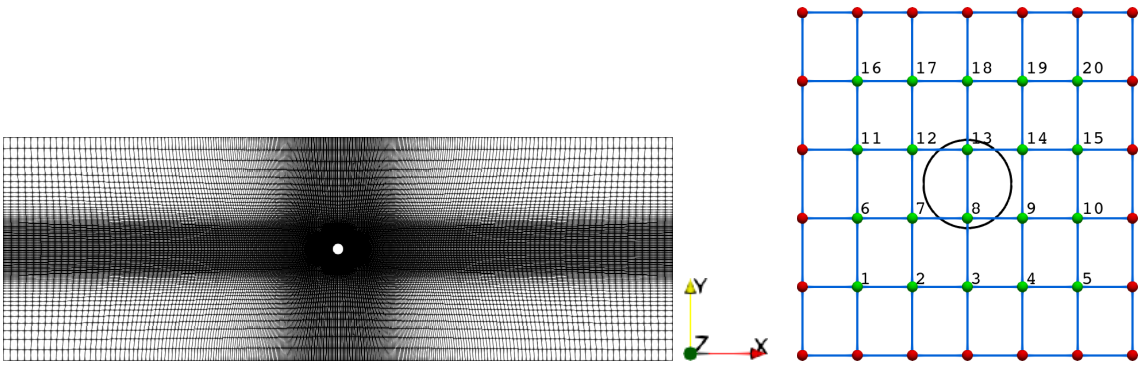


Figure 3.1: Cylinder Optimization. **Left**: Computational grid. **Right**: 11×10 VBS morphing box for the parameterization of the geometry and enclosed grid. CPs in red remained fixed throughout the optimization whereas green ones were allowed to move.

value $V^{\oplus}$) was imposed and the SQP technique of [103] was used for updating the design variables. The values of C_D w.r.t. Re, computed using the steady solver are shown in fig. 3.2 and are compared with reference values from the literature, in the $Re \in [20, 200]$ range. The steady solution underestimated C_D which was affected by unsteadiness past a certain Re number. After $Re \approx 130$, solving the flow equations around the initial geometry using a steady solver without the RPM resulted to the system of equations reaching limit cycle oscillations.

Using the RPM, convergence was made possible at $Re = 140$ and the so-computed flow fields were used to solve the adjoint equations which also failed to converge without the RPM (fig. 3.3). The setup for the RPM varied between the two solvers. In the solution of the primal equations, the RPM was activated after the first 500 iterations and the update of the V_p basis with new

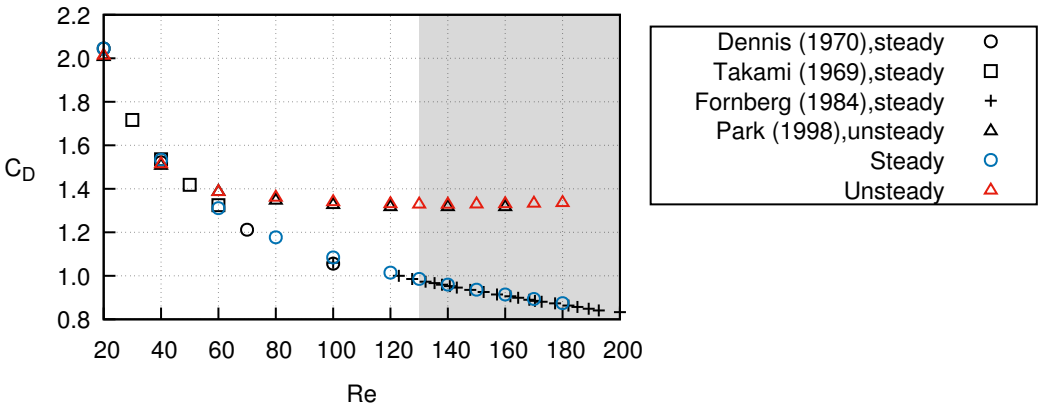


Figure 3.2: Cylinder Optimization. C_D of the circular cylinder versus Re. Reference C_D values were compiled using tabulated data from [20, 147, 29] and are the result of steady solutions obtained with various methods. Values beyond $Re = 120$ were digitized from [29]. The time-averaged C_D values were digitized from [121]. The grey-shaded region of the plot corresponds to the Re range in which the RPM had to be used in order for the SIMPLE solver to converge.

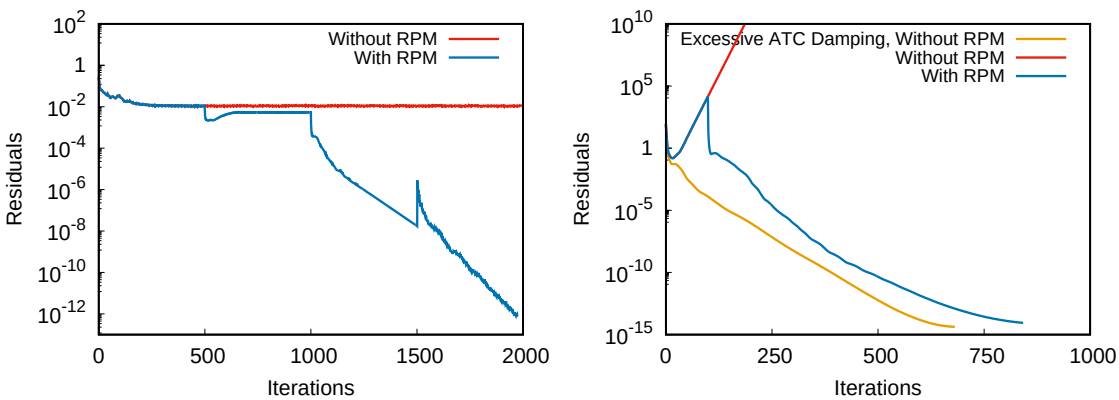


Figure 3.3: Cylinder Optimization. **Left**: Residuals of the primal equations. **Right**: Residuals of the adjoint equations, solved after having stabilized the primal ones by means of the RPM. The geometric mean of the residuals of all variables is shown in all plots.

modes was performed every 500 iterations. In the adjoint solution, this was activated after the first 100 iterations and basis update occurred every 1000 iterations. The size of D in eq. 3.24 was 5 and the 6 latest $\boldsymbol{U}_q$ iterates were used to estimate the basis for $\mathbb{P}$ in each case, according to the method described in section 3.1.2. During the primal solution, the RPM identified 2 modes responsible for divergence and 3 stable, yet slowly decaying, modes which were appended to V_p, improving the convergence rate for the latter part of the solution; for the adjoint problem, 2 diverging modes were used by the RPM to form V_p.

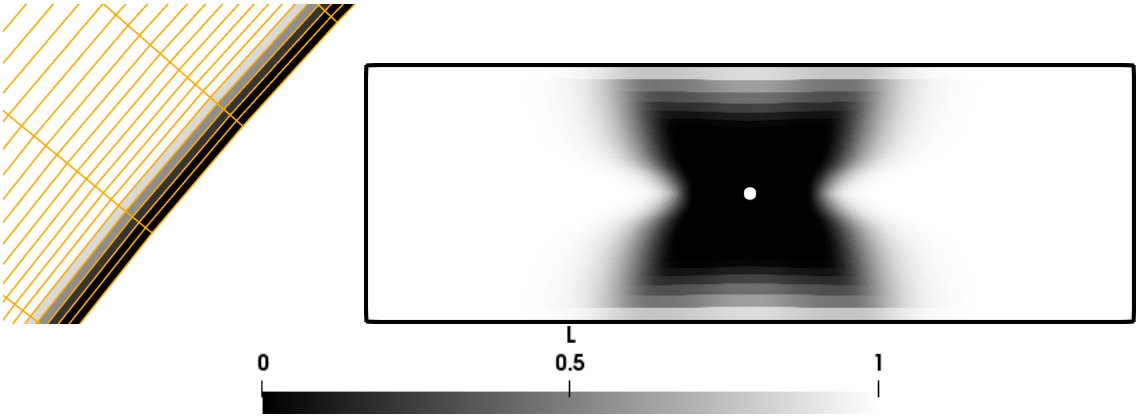


Figure 3.4: Cylinder Optimization. Regions (non whitish) where ATC damping was applied. **Left**: Within the span of 5 cell rows from the solid wall (which proved enough for convergence only in the presence of the RPM). **Right**: Almost throughout the whole domain in order to stabilize the adjoint equations without the RPM. L becomes zero at the first row of cells from the solid wall eliminating the ATC and gradually increases to one further within the domain.

After being able to get converged solutions to both the primal and adjoint equations on the starting geometry, despite vortex shedding, an optimization loop was initiated. Although, in this case, convergence of the adjoint equations was achieved solely through the use of the RPM, there are examples (see section 5.2) in which the RPM could not provide stability on its own, so ATC damping was additionally needed. Therefore, it was of interest to assess the impact of this damping in this case, where the RPM could ensure convergence. Damping was performed by means of 5 smoothing iterations and the resulting L multiplier field can be seen in fig. 3.4. SDs computed using the adjoint method were compared to FDs, fig. 3.5, and errors in the adjoint SDs are quantified in Table 3.1. The adjoint sensitivities were in good agreement with FDs, with or without damping the ATC and this indicates that mild ATC damping is acceptable. Note that, without using the RPM, an excessive amount of 500 smoothing iterations was required, leading to significantly different SDs, even with opposite signs for some of the CPs, (see fig. 3.5 for the SDs & fig. 3.4 for L field). This difference in the number of smoothing iterations highlights the difference between mild and excessive ATC damping and demonstrates the extent which the ATC should be eliminated to, in order to get converged results without the use of the RPM (which depends on the case).

Shape optimization proceeded using the RPM for both the primal and adjoint sets of equations. After 20 optimization cycles, C_D was decreased by 39.8% and the volume constraint was eventually satisfied ($1 > V/V_{\oplus} > 0.9999$), fig. 3.6. Around the initial shape, the time-averaged value of C_D of the unsteady CFD solution was 1.345 whereas the one computed with the RPM-assisted steady

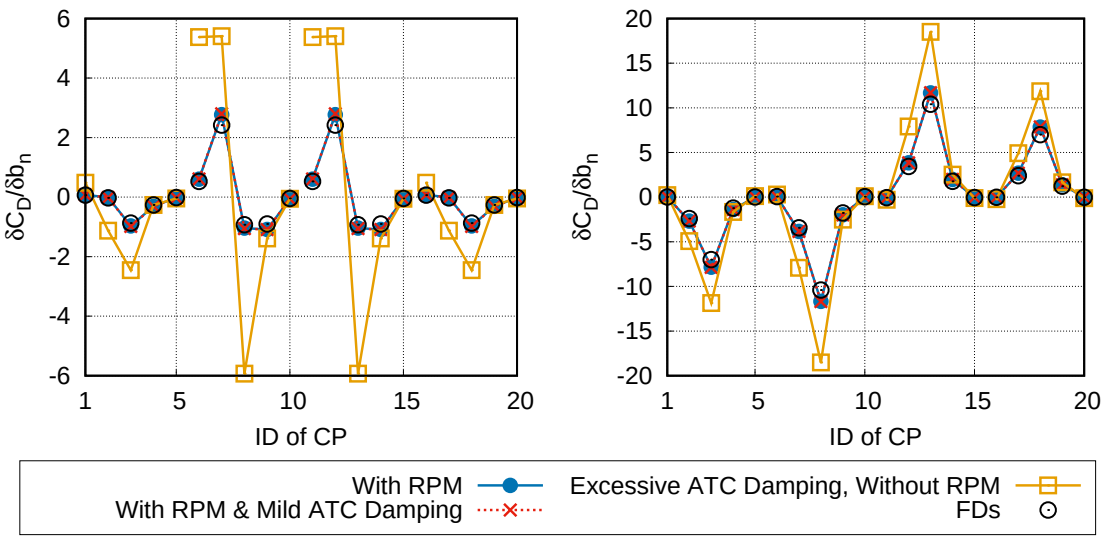


Figure 3.5: Cylinder Optimization. C_D SDs w.r.t. the x (left) and y (right) coordinates of the VBS CPs. The RPM stabilized the primal solvers in all cases, even for FD computations. The adjoint equations were stabilized by the RPM except for the case in which excessive ATC damping was used. Mild and excessive ATC damping were caused by means of 5 and 500 smoothing iterations on L, respectively (see fig. 3.4). All SDs plotted in this figure made use of steady primal (and adjoint) solvers. For FDs, a step size $\varepsilon = 10^{-6}$ was used.

Table 3.1: Cylinder Optimization. Adjoint SDs and FDs (same parametric study as in fig. 3.5). Differences are expressed as $\frac{\|\boldsymbol{SD}_{Adj} - \boldsymbol{SD}_{FD}\|}{\|\boldsymbol{SD}_{FD}\|} 100\%$ as well as by the angles between the gradient vectors in degrees. Mild ATC damping is harmless whereas excessive one introduces significant errors.

	With RPM	With RPM & Mild ATC Damping	Excessive ATC Damping, Without RPM
Difference	12%	13%	97%
Angle (°)	0.4	0.4	15

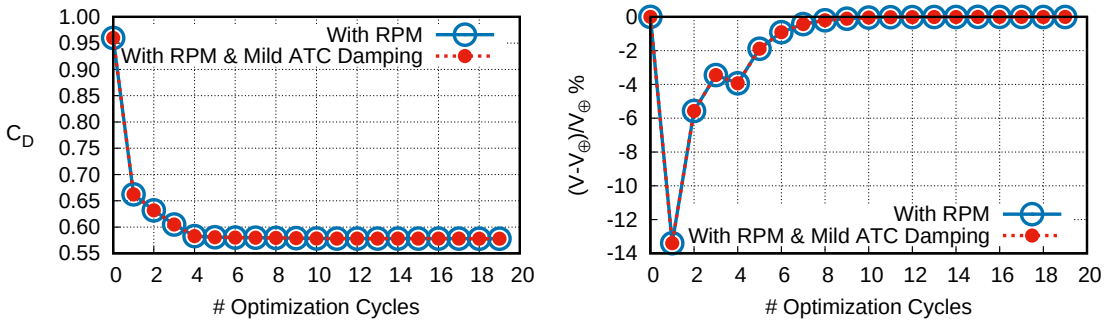


Figure 3.6: Cylinder Optimization. Evolution of the objective function and constraint. **Left**: C_D. **Right**: Volume constraint expressed as percentage difference from the target value.

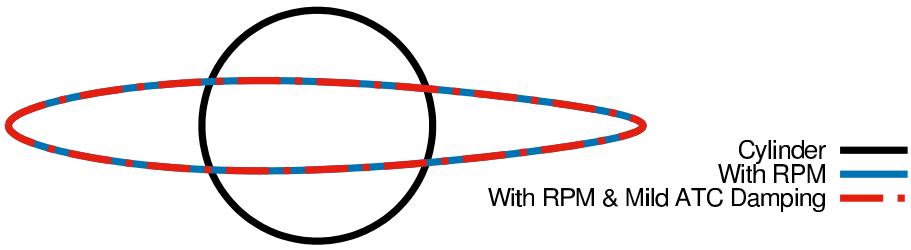


Figure 3.7: Cylinder Optimization. Geometries before and after the C_D minimization using the full ATC model and the RPM and mild ATC damping and the RPM.

solver was 0.96; this was a significant difference which vanished in the optimized solution, as the flow became steady, with $C_D = 0.58$. The same optimization was performed using the RPM and mild ATC damping. The objective function value was reduced by the same degree and mild ATC damping led to indiscernible differences in the optimized shapes, fig. 3.7. The flow fields around the initial and optimized shapes are plotted in fig. 3.8.

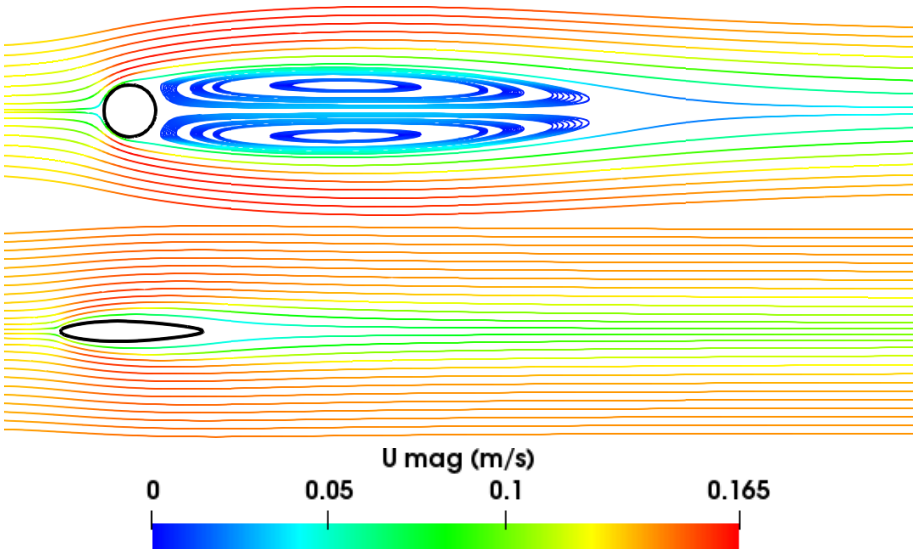


Figure 3.8: Cylinder Optimization. **Top**: Steady flow streamlines around the initial shape (cylinder), computed using the RPM. **Bottom**: Streamlines around the optimized shape with suppressed vortex shedding.

3.2.2 Shape Optimization of an Isolated Airfoil

This section is dealing with the optimization of an isolated NACA0012 airfoil for min. C_D and max. C_L. The laminar flow around this airfoil at Re=1000 has been studied in the literature at various farfield flow angles [78, 76]. By increasing the angle, the flow switches from steady to unsteady, the onset of unsteadiness being at $\alpha_\infty \approx 8^o$. From this point onward, vortices begin shedding in the wake of the airfoil.

First, a comparative study between the computed objective function values of the steady and unsteady simulations on the NACA0012 airfoil was performed. Computations were carried out in the range of $\alpha_\infty \in [0^o, 20^o]$. C_D and C_L were computed using both a steady and an unsteady solver, the latter followed by time-averaging, fig. 3.9. Since the time period of the coefficients' oscillations changes with α_∞, the time series was filtered using a Hanning window in order to correctly compute the time-averaged values. Beyond a certain α_∞, the RPM was necessary for the stabilization of the steady flow solver. For $\alpha_\infty > 8^o$, the time-averaged values of the coefficients were greater than those computed by the stabilized steady solver.

The airfoil was optimized at $\alpha_\infty = 20^o$. The optimization was purposefully performed at such a high α_∞ in order to generate unsteadiness for assessing the capabilities and the extent of the use of the RPM. With α_∞=20°, the primal solver definitely required RPM stabilization to converge. Compared to the previous case, herein the flow remained unsteady even after the optimization and this made the use of the RPM even more necessary. The CPs of the VBS morphing

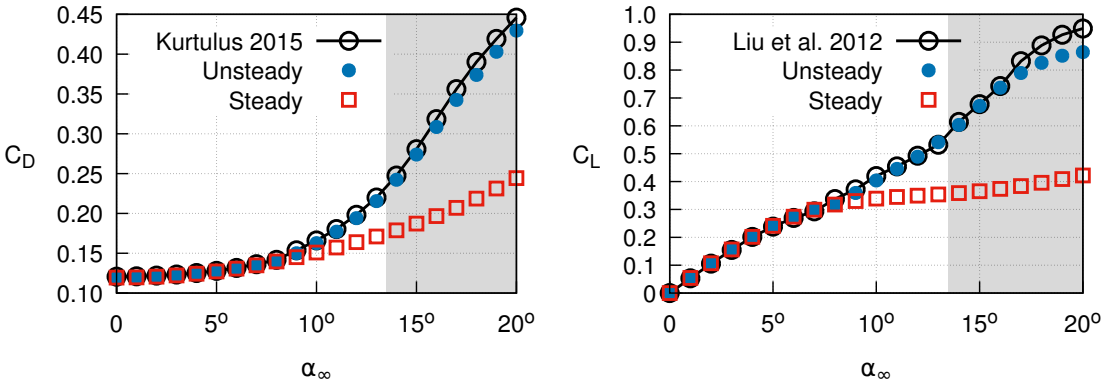


Figure 3.9: NACA0012 airfoil. Aerodynamic coefficients computed by the unsteady flow solver and the RPM-assisted steady flow solver. **Left**: C_D. **Right**: C_L. Reference points were digitized from [78, 76].

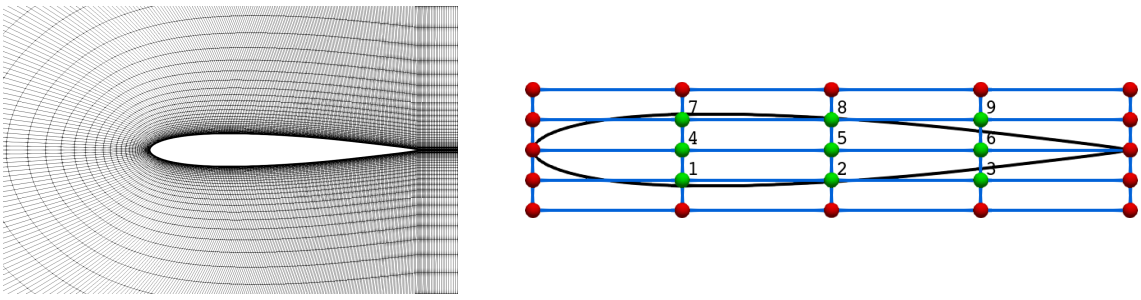


Figure 3.10: NACA0012 airfoil. **Left**: Computational grid close to the airfoil. **Right**: 5×5 VBS morphing box parameterizing the airfoil geometry and enclosed grid. CPs in red remained fixed throughout the optimization whereas green ones could be displaced.

box parameterizing the airfoil contour and part of the grid close to it are shown in fig. 3.10. Constraints were imposed, keeping the cross-sectional area of the airfoil V within $\pm 0.1\%$ of its initial value $V^{\oplus}$ throughout the optimization, in addition to the bound constraints of eq. 2.34 on the morphing box CPs. These constraints were imposed using the interior-point SQP method [103].

The RPM stabilized the flow equations' solver and, the adjoint solvers, based on the so-computed steady flow for both targets C_D and C_L, could not converge without the RPM. Using the RPM, the adjoint solvers were able to converge without damping the ATC for both quantities. The convergence of all these solvers is shown in fig. 3.11. The RPM was activated after the first 1000 iterations in the primal solution and after 150 iterations in each adjoint run; 15 and 5 solution snapshots were used for the flow and adjoint solvers, respectively, in order to construct the V_p basis. Divergence of the flow equations was due to 2 diverging modes; 4 diverging modes were identified during the solution of both systems of adjoint equations

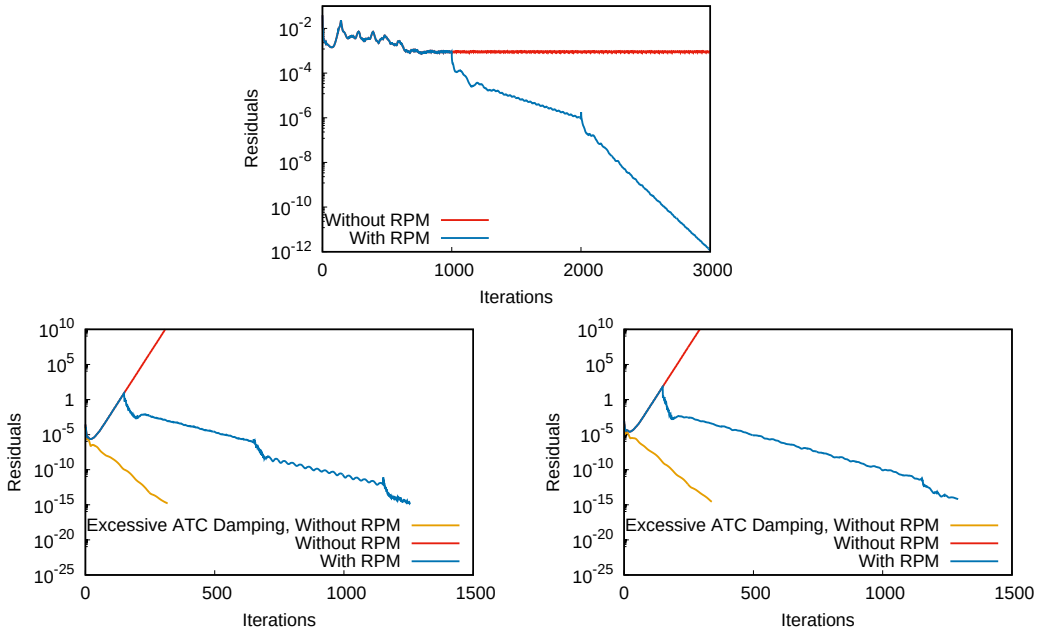


Figure 3.11: NACA0012 airfoil. Geometric mean of residuals. **Top**: Residuals of the primal equations. **Bottom**: Residuals of the adjoint equations for C_D (left) and C_L (right). All adjoints were based on the flow fields obtained using the RPM stabilization.

and, additionally, the RPM picked up a few stable modes as well which enhanced the convergence rate. Overall, for the adjoint equations, the size of V_p ranged from 2 to 4, in all optimization cycles.

The same study on the effect of ATC damping was carried out, as in section 3.2.1, which re-confirmed previous findings and conclusions, see fig. 3.12 and Table 3.2. Indeed, mild ATC damping does not affect the accuracy of the adjoint sensitivities. The field of L, indicating the regions of ATC damping, is shown in fig. 3.13. Additionally, SDs computed using the "pseudo-steady" averaged flow field are also plotted. The SDs computed using the latter were very close to the accurate ones.

Shape optimization proceeded for both objectives at $\alpha_\infty = 20^o$. The RPM was first used both for the primal and the adjoint solvers, without any damping for the ATC. An optimization loop was also performed for each objective using the mild ATC damping technique in conjunction with the RPM for the solution of the adjoint equations. Additionally, to assess the effect of the "pseudo-steady" averaging of flow fields used in other cases, the optimization runs were performed using this technique as well. An averaged objective function value was, thus, computed (F_{avg}) and the RPM was used only to get a converged adjoint solution. Upon completion of all optimization loops, results were re-evaluated using the unsteady flow solver. The time-dependent aerodynamic coefficients were averaged over $\sim$10 vortex shedding periods, for each new design. The

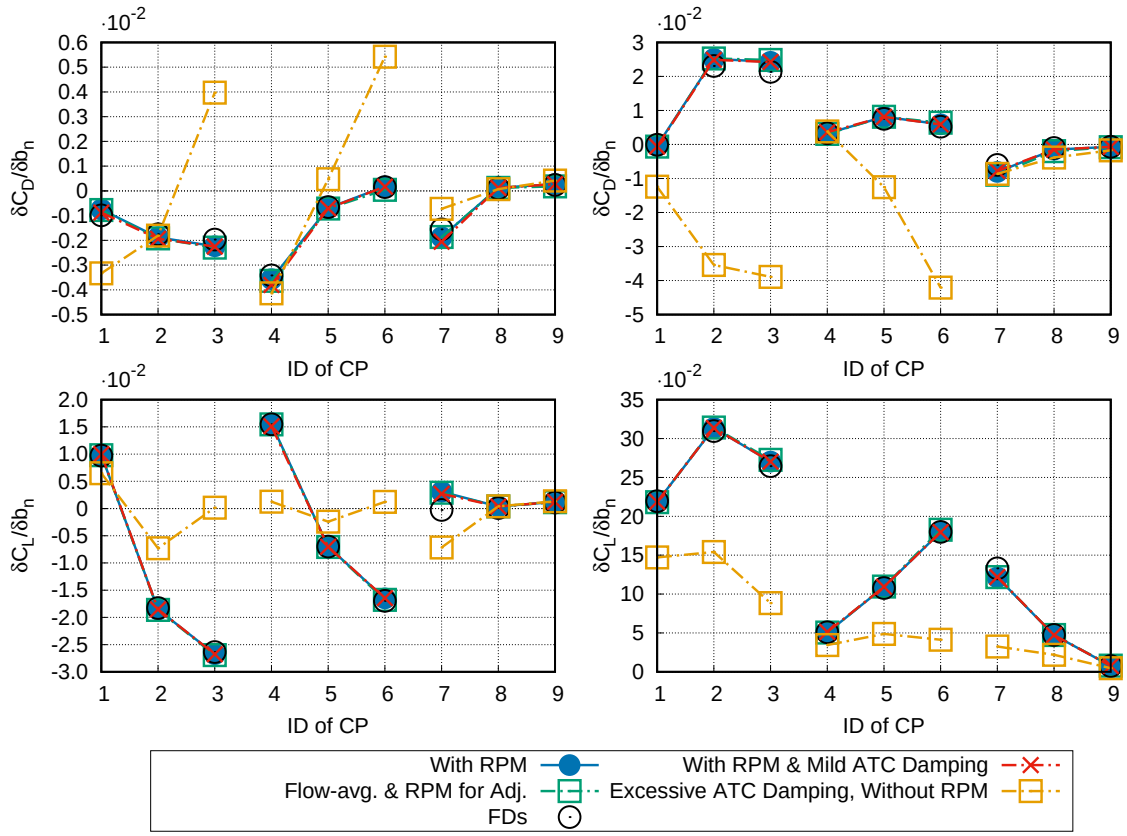


Figure 3.12: NACA0012 airfoil. **Top**: C_D SDs w.r.t. to the x (left) and y (right) coordinates of the CPs. **Bottom**: C_L SDs. 5 and 300 smoothing iterations were used in the cases of mild and excessive ATC damping, respectively. "Flow-avg." indicates SDs computed by averaging the flow fields of the last 1000 iterations of the steady flow solver, followed by an adjoint solution the divergence of which was treated with the RPM.

Table 3.2: NACA0012 airfoil. Adjoint SDs and FDs (as in fig. 3.12) for C_D and C_L. The SD error due to excessive ATC damping was much higher than in the cylinder case and the same holds for the direction to which the SDs point to in the design space.

	With RPM	With RPM & Mild ATC Damping	Flow-avg. and RPM for Adjoint	Excessive ATC Damping, Without RPM
		C_D		
Difference	13%	12%	16%	295%
Angle (°)	2.7	2.5	4.1	141.5
		C_L		
Difference	2.7%	2.6%	2.9%	42.7%
Angle (°)	1.5	1.4	1.6	18.8

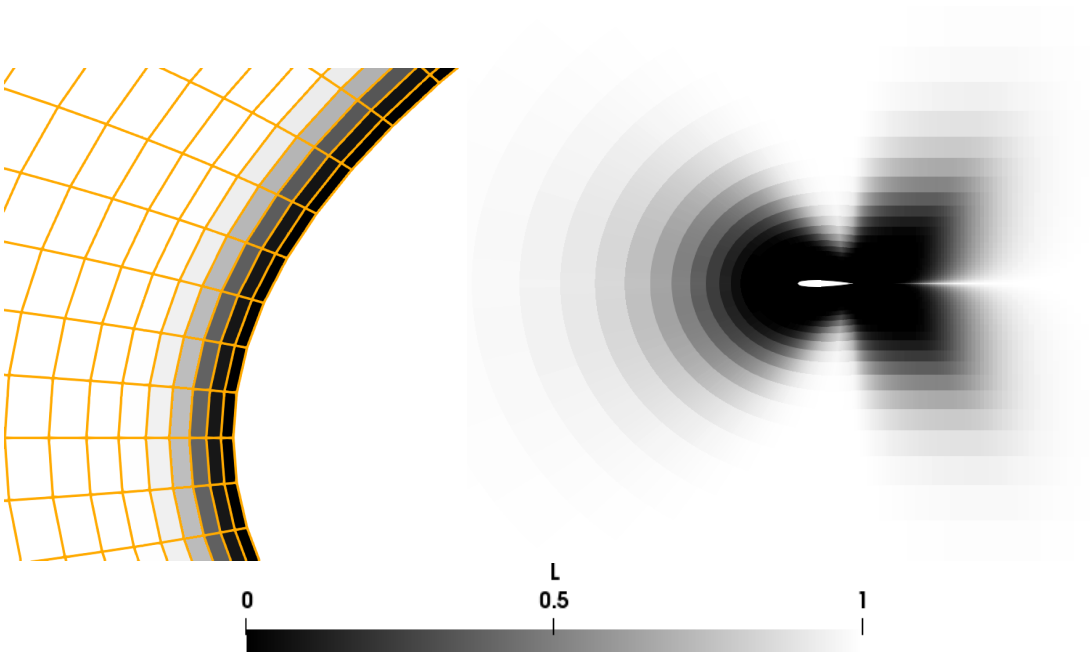


Figure 3.13: NACA0012 airfoil. Regions where ATC damping was applied. **Left**: Mild ATC damping within the span of 5 cell rows close to the wall. **Right**: Excessive ATC damping, throughout most of the domain in order to stabilize the adjoint equations without the RPM.

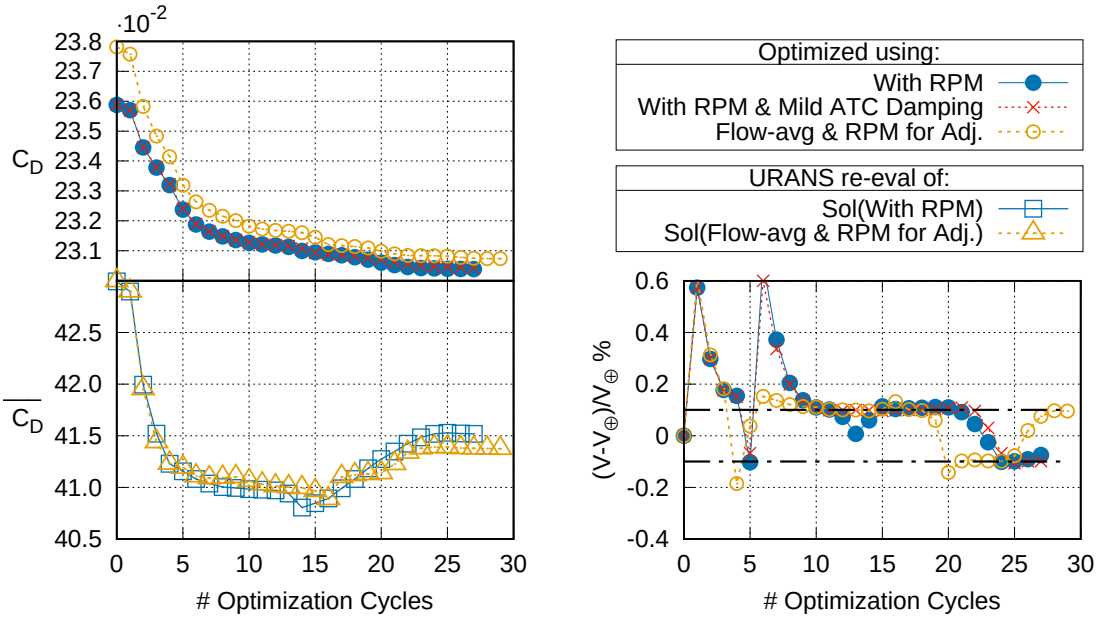


Figure 3.14: NACA0012 airfoil. Convergence of the optimization loop for min. C_D. **Top-Left**: C_D or $C_{D,\text{avg}}$. **Bottom-Left**: $\overline{C_D}$ re-evaluated at the end of each cycle using URANS. **Bottom-Right**: Volume constraint. Black dashed lines indicate the $\pm 0.1\%$ range. Sol("technique") in the legend key stands for the optimized solution computed using "technique".

vortex shedding period was different in each design. Instead of re-adjusting the time-averaging window for each design, a Hanning window was used to minimize the influence of the time series tail.

For min. C_D, the convergence of the objective functions, as well as the time-averaged ones, is plotted in fig. 3.14; C_D was reduced by 2.6% and $\overline{C_D}$ by 7%. Results were then compared with an unsteady-based shape optimization, shown in fig. 3.15. Despite the reduction achieved in $\overline{C_D}$ with the RANS-based optimization, there was still room for improvement, possible only with an unsteady adjoint. However, computational time was saved by starting the unsteady-based optimization loop from the one optimized with the RPM-assisted steady solver. In that case, for the sake of comparison, the optimization was started using the final position of the CPs and the airfoils were not parameterized anew. Following this approach, there was a ~40% cost reduction, assuming that the cost of the initialization based on steady runs is negligible compared to even a single unsteady computation. In this case, the steady optimization loop using the full ATC model and the RPM for both solvers was concluded within less than one CPU hour whereas one URANS-based optimization cycle required approximately 2 CPU hours, both running on 4 processors (AMD© EPYC 7452 32-Core Processor). The final shapes are compared in fig. 3.16.

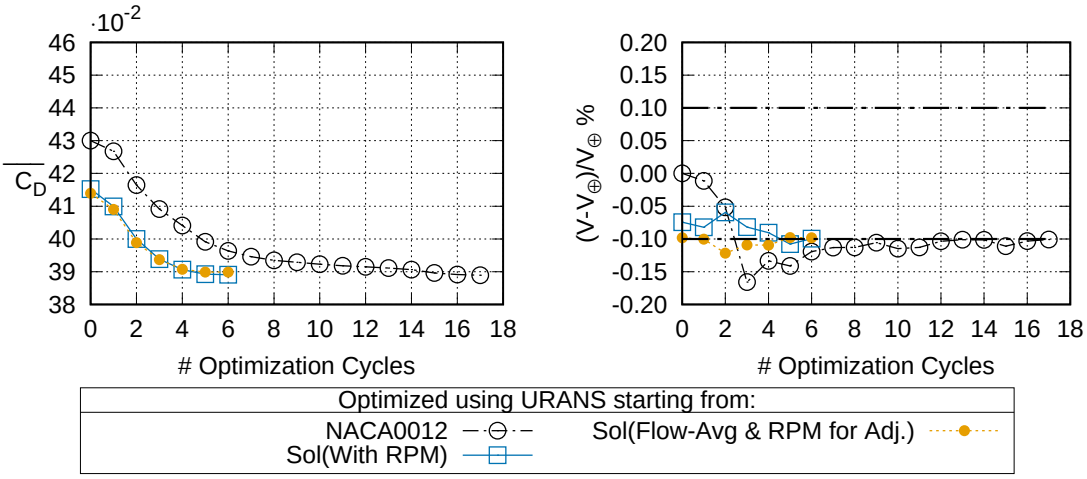


Figure 3.15: NACA0012 airfoil. Convergence of the optimization loop for min. $\overline{C_D}$. **Left**: $\overline{C_D}$. **Right**: Volume constraint within the acceptance tolerance. All optimizations start from different airfoils, namely the NACA0012 airfoil and the ones at the end of the optimization runs shown in fig. 3.14.

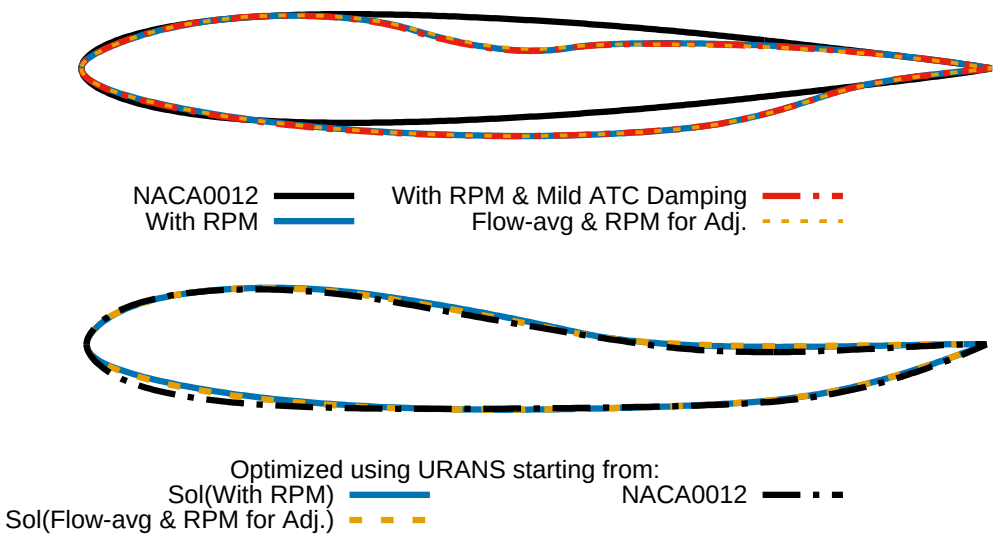


Figure 3.16: NACA0012 airfoil. **Top**: Airfoils optimized for min. C_D using steady-state solvers (fig. 3.14). **Bottom**: Airfoils optimized for min. $\overline{C_D}$ using unsteady solvers (fig. 3.15).

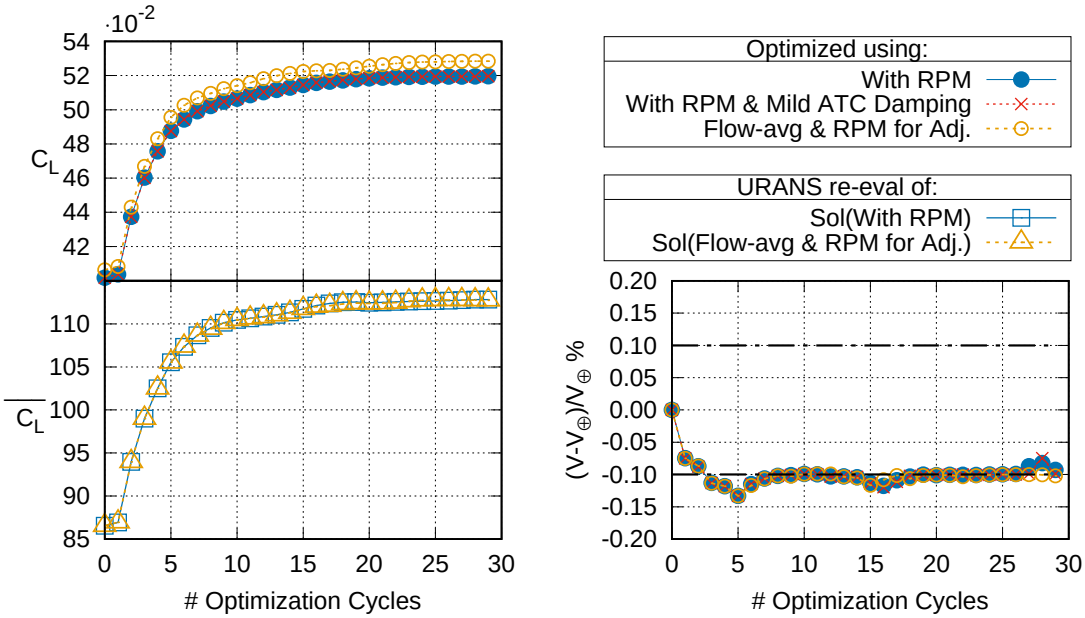


Figure 3.17: NACA0012 airfoil. Convergence of the optimization loop for max. C_L. **Top-Left**: C_L or $C_{L,\text{avg}}$. **Bottom-Left**: $\overline{C_L}$ re-evaluated at the end of each cycle using URANS. **Bottom-Right**: Volume constraint within the acceptance tolerance.

For max. C_L, the convergence of C_L and $\overline{C_L}$ is plotted in fig. 3.17; C_L was increased by 28.9% and $\overline{C_L}$ by 30.5%. Time-averaged objectives were indeed increased despite the use of steady solvers for the optimization. In contrast to the optimization for min. C_D, the performance of the optimized airfoil was very close to the one computed by a URANS-based shape optimization, shown in fig. 3.18.

In contrast to the previous case in section 3.2.1, vortex shedding was present around the optimized shapes and unsteady re-evaluations of the latter were performed. Even by optimizing the shape using steady solvers, the unsteady re-evaluations revealed a significant gain in the time-averaged objective function value. During the final stages of the optimization for min. C_D, using the steady solver, minor reductions in C_D took place; however, the unsteady re-evaluations showed that $\overline{C_D}$ could not further be improved and took on a higher value instead. This is the usual limitation of all optimization methods that rely upon a lower-accuracy model to perform the search and, at the end, re-evaluate the optimal solution with a high-fidelity model. Since the gap between the two models is usually not negligible, small refinements cannot always be reconfirmed by the higher-fidelity model. To improve the time-averaged objective functions further, a URANS-based shape optimization had to be performed. However, even for a purely unsteady case such as this one, there was a significant cost reduction by using steady solvers to compute a good starting point for the

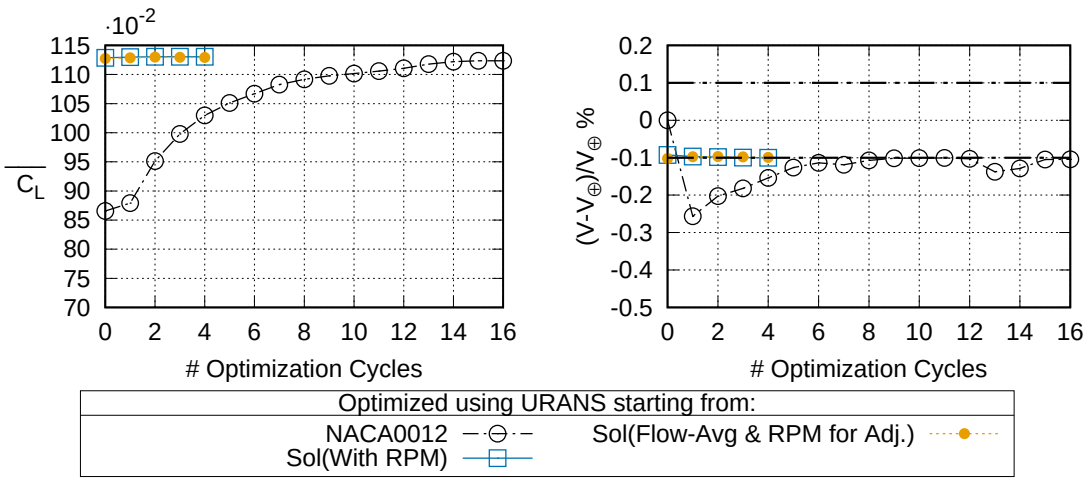


Figure 3.18: NACA0012 airfoil. Convergence of the optimization loop for max. $\overline{C_L}$. **Left**: $\overline{C_L}$. **Right**: Volume constraint within acceptance tolerance.

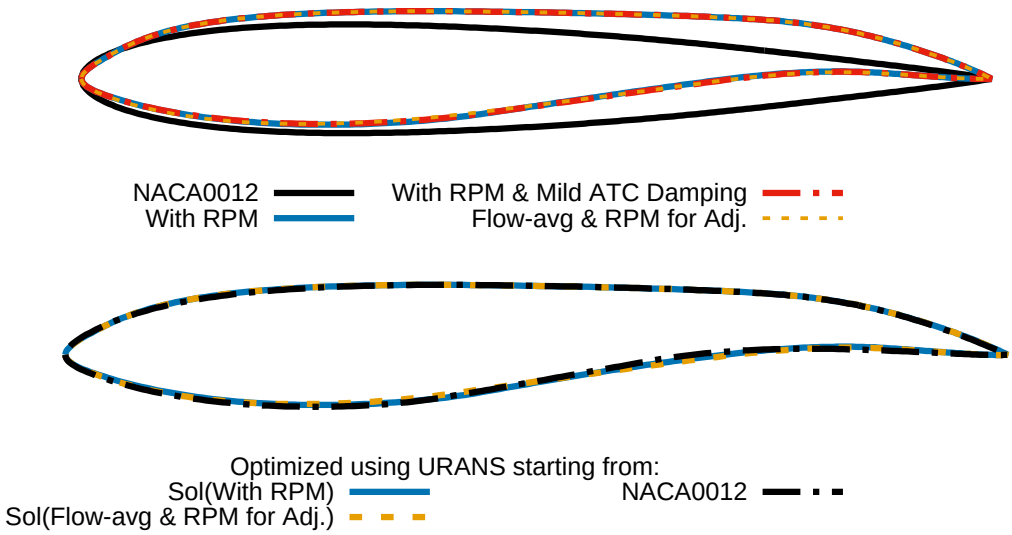


Figure 3.19: NACA0012 airfoil. **Top**: Airfoils optimized for max. C_D using steady-state solvers (fig. 3.17). **Bottom**: Airfoils optimized for max. $\overline{C_L}$ using unsteady solvers (fig. 3.18).

URANS-based shape optimization.

Chapter 4

Shape Optimization in the Presence of Uncertainties

In a Robust Design Optimization (RDO) problem, there are uncertain variables $\boldsymbol{c} \in \mathbb{R}^M$ which, alongside $\boldsymbol{b}$, serve as input to the model. Their effect is propagated to the Quantity of Interest (QoI) which is now treated as a statistical quantity. Usually, the RDO deals with the minimization/maximization of the expression

$$J = \mu_F + w\sigma_F \tag{4.1}$$

where μ_F and σ_F are the mean value and standard deviation of the QoI F, respectively, and w weights the two statistical moments.

The role of an Uncertainty Quantification (UQ) method is to compute μ_F and σ_F. In the problems this thesis is dealing with, a probabilistic approach is adopted where all c_m are assumed to follow a certain (known) type of probability distribution, and are uncorrelated.

4.1 Uncertainty Quantification Methods

4.1.1 The Method of Moments

In the Method of Moments (MoM), the QoI is expanded around the mean values $\boldsymbol{\mu}$ of $\boldsymbol{c}$,

$$F(\boldsymbol{\mu} + \Delta\boldsymbol{c}) = F(\boldsymbol{\mu}) + \left.\frac{\delta F}{\delta c_m}\right|_{\boldsymbol{c}=\boldsymbol{\mu}} \Delta c_m + O\left(\Delta\boldsymbol{c}^2\right) \tag{4.2}$$

Retaining only first-order terms in eq. 4.2 gives rise to a First-Order Second-Moment (FOSM) method, according to which the mean value of F is

given by

$$\begin{aligned}\mu_F^{(FOSM)} &= \int_{-\infty}^{+\infty} F(\boldsymbol{c}) h_c(\boldsymbol{c}) d\boldsymbol{c} \\ &= \int_{-\infty}^{+\infty} \left[F(\boldsymbol{\mu}) + \left.\frac{\delta F}{\delta c_m}\right|_{\boldsymbol{\mu}} (c_m - \mu_m) h_c(\boldsymbol{c}) \right] h_c(\boldsymbol{c}) d\boldsymbol{c} \\ &= F(\boldsymbol{\mu}) \int_{-\infty}^{+\infty} h_c(\boldsymbol{c}) d\boldsymbol{c} + \left.\frac{\delta F}{\delta c_m}\right|_{\boldsymbol{\mu}} \int_{-\infty}^{+\infty} (c_m - \mu_m) h_c(\boldsymbol{c}) d\boldsymbol{c} \end{aligned} \tag{4.3}$$

where $h_c(\boldsymbol{c})$ is the PDF of $\boldsymbol{c}$. The variance of F is equal to

$$\begin{aligned}\left(\sigma_F^{(FOSM)}\right)^2 &= \int_{-\infty}^{+\infty} \left(F(\boldsymbol{c}) - \mu_F\right)^2 h_c(\boldsymbol{c}) d\boldsymbol{c} \\ &= \int_{-\infty}^{+\infty} F(\boldsymbol{c})^2 h_c(\boldsymbol{c}) d\boldsymbol{c} - 2\mu_F \int_{-\infty}^{+\infty} F(\boldsymbol{c}) h_c(\boldsymbol{c}) d\boldsymbol{c} + \mu_F^2 \int_{-\infty}^{+\infty} F(\boldsymbol{c}) h_c(\boldsymbol{c}) d\boldsymbol{c} \\ &= F(\boldsymbol{\mu})^2 \int_{-\infty}^{+\infty} h_c(\boldsymbol{c}) d\boldsymbol{c} + 2F(\boldsymbol{\mu}) \left.\frac{\delta F}{\delta c_m}\right|_{\boldsymbol{\mu}} \int_{-\infty}^{+\infty} (c_m - \mu_m) h_c(\boldsymbol{c}) d\boldsymbol{c} \\ &+ \left[\frac{\delta F}{\delta c_m}\frac{\delta F}{\delta c_j}\right]_{\boldsymbol{\mu}} \int_{-\infty}^{+\infty} (c_m - \mu_m)(c_j - \mu_j) h_c(\boldsymbol{c}) d\boldsymbol{c} - 2\mu_F F(\boldsymbol{\mu}) \int_{-\infty}^{+\infty} h_c(\boldsymbol{c}) d\boldsymbol{c} \\ &- 2\mu_F \left.\frac{\delta F}{\delta c_m}\right|_{\boldsymbol{\mu}} \int_{-\infty}^{+\infty} (c_m - \mu_m) h_c(\boldsymbol{c}) d\boldsymbol{c} + \mu_F^2 \int_{-\infty}^{+\infty} h_c(\boldsymbol{c}) d\boldsymbol{c} \end{aligned} \tag{4.4}$$

For symmetric PDFs, such as those used in this thesis, the first integral on the r.h.s. of eq. 4.3 is equal to 1 whereas the second one vanishes. Also, the first, fourth and the last integral on the r.h.s. of eq. 4.4 cancel out whereas the second and fifth ones vanish. Thus, in FOSM, the statistical moments of the QoI

$$\mu_F^{(FOSM)} = F(\boldsymbol{\mu}) \tag{4.5a}$$

$$\sigma_F^{(FOSM)} = \sqrt{\left.\frac{\delta F}{\delta c_m}\right|_{\boldsymbol{\mu}} \left.\frac{\delta F}{\delta c_j}\right|_{\boldsymbol{\mu}} \mathcal{K}_{mj}} \tag{4.5b}$$

where F and $\frac{\delta F}{\delta c_m}$ are computed at the mean values of c_m and

$$\mathcal{K}_{mj} = \int_{-\infty}^{+\infty} (c_m - \mu_m)(c_j - \mu_j) h_c(\boldsymbol{c}) d\boldsymbol{c} \tag{4.6}$$

is the covariance matrix of $\boldsymbol{c}$ and is diagonal in case c_m are uncorrelated or symmetric in the more general case. The diagonal elements depend on the type of the probability distribution. For the Gauss normal distribution, each m-th diagonal element is equal to σ_m^2. For bounded PDFs, the integrals in eqs. 4.3

and 4.4 are accounted for within the corresponding bounds, instead of $[-\infty, +\infty]$.

To compute σ_F with eq. 4.5b, $\frac{\delta F}{\delta c_m}$ are required. These are computed efficiently by the adjoint method making the cost of UQ with FOSM, per optimization cycle, to be equal to 2 EFS.

4.1.2 The Polynomial Chaos Expansion

According to the PCE method, for any candidate shape (i.e. a different value-set of $\boldsymbol{b}$), the QoI is approximated by

$$F(\boldsymbol{b}, \boldsymbol{c}(\boldsymbol{\zeta})) \approx \sum_{q=0}^{Q-1} \alpha_q \Psi_q(\boldsymbol{\zeta}) \tag{4.7}$$

where α_q are the PCE coefficients, Ψ_q are multivariate orthogonal polynomials and $\boldsymbol{\zeta}$ are standardized random variables for which $\boldsymbol{c} = \boldsymbol{c}(\boldsymbol{\zeta})$, depending on the parameterization of the probability distribution. The basis $\boldsymbol{\Psi}$ comprises multivariate polynomials of zeroth up to a user-defined degree, O_{PCE}, referred to as the PCE order. Each multivariate polynomial in the basis is a product of M univariate ones. For each c_m, there is a sequence of $O_{PCE} + 1$ orthogonal univariate polynomials, one for each degree. These are chosen from the Askey scheme depending on the type of probability distribution [163]. For each polynomial degree, from zero till O_{PCE}, univariate polynomials from each sequence are combined so that their products yield a set of multivariate polynomials of that degree. For each degree, many multivariate polynomials, orthogonal to each other, may result from these combinations, depending on M. Thus, from the theory of combinatorics, the number of terms retained on the r.h.s. of eq. 4.7 is $Q = \frac{(O_{PCE}+M)!}{O_{PCE}!M!}$. To accurately model the effects of higher order terms in the model computing F, a higher O_{PCE} is used. For this value-set of b_n, the statistical moments of F result from [163]

$$\mu_F^{(PCE)} = \alpha_0 \tag{4.8a}$$

$$\sigma_F^{(PCE)} = \sqrt{\sum_{q=1}^{Q-1} \alpha_q^2 \|\Psi_q\|^2} \tag{4.8b}$$

where

$$\|\Psi_i\|^2 = \int_{-\infty}^{+\infty} \Psi_j \Psi_j \delta_{ij} h_c(\boldsymbol{\zeta}) d\boldsymbol{\zeta} \tag{4.9}$$

is the h_c-norm of the orthogonal polynomials.

To compute the coefficients α_q, in standard regression-based PCE, after evaluating F at L value-sets of c_m (collocation points), the system $A\boldsymbol{\alpha} = \boldsymbol{f}$, with

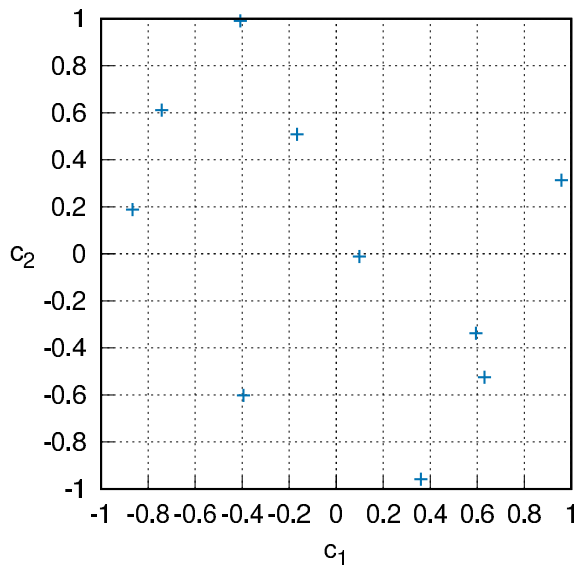


Figure 4.1: An example of Latin Hypercube Sampling for 2 uncertain variables for a uniform probability distribution. 10 collocation points are used. Each direction was split into 10 segments and within any of the so-defined sub-areas, one point at most was randomly chosen. The user only defines the number of collocation points.

$A \in \mathbb{R}^{L \times Q}$ and $\boldsymbol{f} \in \mathbb{R}^{L}$, is assembled. In more detail, the system has the following form

$$\underbrace{\begin{bmatrix} \Psi_0(\boldsymbol{\zeta}^{(1)}) & \dots & \Psi_{Q-1}(\boldsymbol{\zeta}^{(1)}) \\ \vdots & \ddots & \vdots \\ \Psi_0(\boldsymbol{\zeta}^{(L)}) & \dots & \Psi_{Q-1}(\boldsymbol{\zeta}^{(L)}) \end{bmatrix}}_{A} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{Q-1} \end{bmatrix} = \underbrace{\begin{bmatrix} F(\boldsymbol{c}^{(1)}) \\ \vdots \\ F(\boldsymbol{c}^{(L)}) \end{bmatrix}}_{\boldsymbol{f}} \tag{4.10}$$

The L collocation points are generated using a sampling technique. Latin Hypercube Sampling [91] is used in this thesis. This sampling technique operates by splitting the cumulative PDF into equiprobable segments and randomly instantiating each uncertain variable within each segment. In 2D, the set of uncertain variables forms a latin square i.e. a square whose elements appear only once in each row and column. Random permutations are applied to the elements in the set so as to re-combine them as in fig. 4.1. Alternative methods for choosing collocation points exist and could substitute this method.

In case one has access to the adjoint method with which the derivatives of F w.r.t. c_m can be computed at the cost of 1 EFS, irrespective of M, extra data per collocation point may become available in the form of these derivatives. With more data, the total number of collocation points can be reduced. Since the additional data are computed at the cost of 1 EFS per collocation point, the total cost of

UQ is also reduced. This is the main idea the Adjoint-assisted Polynomial Chaos Expansion (APCE) method is based upon. Starting from eq. 4.7, M additional equations per collocation point are derived in the form of

$$\frac{\delta F}{\delta c_m}\frac{\partial c_m}{\partial \zeta_j} \approx \sum_{q=0}^{Q-1} \alpha_q \frac{\partial \Psi_q}{\partial \zeta_j}(\boldsymbol{\zeta}),\ j = 1, M \tag{4.11}$$

Using eq. 4.11 at L collocation points leads to the formation of $C\boldsymbol{\alpha} = \boldsymbol{g}$ with $C \in \mathbb{R}^{LM\times Q}$ and $\boldsymbol{g} \in \mathbb{R}^{LM}$, written in an expanded form as

$$\underbrace{\begin{bmatrix} \frac{\partial \Psi_0}{\partial \zeta_1}(\boldsymbol{\zeta}^{(1)}) & \dots & \frac{\partial \Psi_{Q-1}}{\partial \zeta_1}(\boldsymbol{\zeta}^{(1)}) \\ \vdots & \ddots & \vdots \\ \frac{\partial \Psi_0}{\partial \zeta_M}(\boldsymbol{\zeta}^{(1)}) & \dots & \frac{\partial \Psi_{Q-1}}{\partial \zeta_M}(\boldsymbol{\zeta}^{(1)}) \\ \vdots & \ddots & \vdots \\ \frac{\partial \Psi_0}{\partial \zeta_1}(\boldsymbol{\zeta}^{(L)}) & \dots & \frac{\partial \Psi_{Q-1}}{\partial \zeta_1}(\boldsymbol{\zeta}^{(L)}) \\ \vdots & \ddots & \vdots \\ \frac{\partial \Psi_0}{\partial \zeta_M}(\boldsymbol{\zeta}^{(L)}) & \dots & \frac{\partial \Psi_{Q-1}}{\partial \zeta_M}(\boldsymbol{\zeta}^{(L)}) \end{bmatrix}}_{C} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{Q-1} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial c_m}{\partial \zeta_1}\frac{\delta F}{\delta c_m}(\boldsymbol{c}^{(1)}) \\ \vdots \\ \frac{\partial c_m}{\partial \zeta_M}\frac{\delta F}{\delta c_m}(\boldsymbol{c}^{(1)}) \\ \vdots \\ \frac{\partial c_m}{\partial \zeta_1}\frac{\delta F}{\delta c_m}(\boldsymbol{c}^{(L)}) \\ \vdots \\ \frac{\partial c_m}{\partial \zeta_M}\frac{\delta F}{\delta c_m}(\boldsymbol{c}^{(L)}) \end{bmatrix}}_{\boldsymbol{g}} \tag{4.12}$$

The combination of eqs. 4.10 and 4.12 leads to the system

$$D\boldsymbol{\alpha} = \boldsymbol{\beta} \tag{4.13}$$

where $D \in \mathbb{R}^{L(M+1)\times Q}$ and $\boldsymbol{\beta} \in \mathbb{R}^{L(M+1)}$. Matrix D and vector $\boldsymbol{\beta}$ are formed by stacking the matrices A and C and vectors $\boldsymbol{f}$ and $\boldsymbol{g}$ from eqs. 4.10 and 4.12, so that

$$D = \begin{bmatrix} A \\ C \end{bmatrix} \quad \text{and} \quad \boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{bmatrix} \tag{4.14}$$

For known PDFs, all the elements of A and C are available from closed-form relations of orthogonal polynomials and their derivatives.

The system of eq. 4.13 is solved using least squares. The gradients of F w.r.t. c_m, required on the r.h.s. of eq. 4.13, are computed using the (continuous) adjoint method. For each value-set of c_m, $1+M$ equations are assembled into the above system at the cost of 2 EFS. The cost of UQ with APCE is equal to $2L$. Compared to standard PCE, eq. 4.10, a smaller L is used by including M additional data per collocation point. The higher the M value, the higher the gain from including the adjoint method into the APCE framework. This will be elaborated further in section 4.2.3.

The least-squares solution is given by

$$\begin{aligned}\boldsymbol{\alpha} = G\boldsymbol{\beta} &= (D^T D)^{-1} D^T \boldsymbol{\beta} \\ &= \left(\begin{bmatrix} A^T & C^T \end{bmatrix} \begin{bmatrix} A \\ C \end{bmatrix} \right)^{-1} \begin{bmatrix} A^T & C^T \end{bmatrix} \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{bmatrix}\end{aligned} \tag{4.15}$$

Both the FOSM and APCE methods require the values of F and $\delta F/\delta c_m$, though at different points in the uncertain space.

An example is given to show the gain from APCE compared to standard PCE. Consider the polynomial function,

$$f(x_1, x_2, x_3) = x_1^4 + 4x_1x_2^2 + x_3^3 + 2x_3^2 + 3x_2 + x_1 + 1 \tag{4.16}$$

and uncertain variables: $x_1 \sim \mathcal{N}(1, 2)$, $x_2 \sim \mathcal{N}(2, 0.5)$, $x_3 \sim \mathcal{N}(0.5, 1)$. To relate these results to CFD problems, assume that 1 time unit (1 EFS in case this was a CFD problem) is required to compute f and 1 time unit is spent by the adjoint method to compute all 3 components of the gradient of f w.r.t. x_1, x_2 and x_3. Using $O_{PCE} = 4$, the α_q and Ψ_q of the PCE expansion in eq. 4.7 can accurately reconstruct f, since this is a 4th-order polynomial. Thus, the statistical moments of f can be computed up to machine accuracy, provided that enough collocation points are used. By increasing L (and, thus, the cost), μ_f and σ_f, computed using standard PCE or APCE, converge to their analytical values, as shown in fig. 4.2. The takeaway from fig. 4.2 is that APCE requires fewer time units than standard PCE to accurately compute μ_f and σ_f by using fewer collocation points.

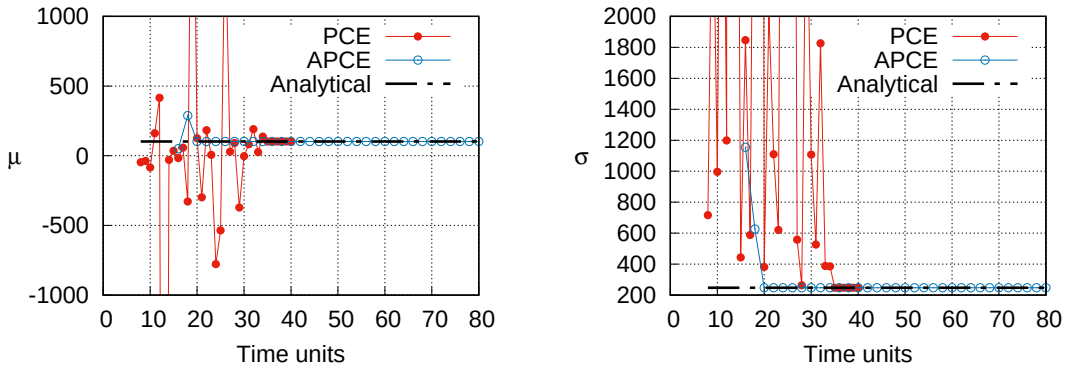


Figure 4.2: The statistical moments μ_f and σ_f of f, eq. 4.16, computed with PCE and APCE. The x axis corresponds to time units (equivalent to EFS in CFD).

4.2 Gradient-based Optimization under Uncertainties

For gradient-based optimization, $\frac{\delta J}{\delta b_n}$ needs to be computed. Equation 4.1 is thus differentiated w.r.t. b_n

$$\frac{\delta J}{\delta b_n} = \frac{\delta \mu_F}{\delta b_n} + w\frac{\delta \sigma_F}{\delta b_n} \tag{4.17}$$

In the case of FOSM, by substituting eqs. 4.5 in eq. 4.17 (and dropping the $\boldsymbol{\mu}$ subscript for simplicity), one gets

$$\begin{aligned}
\frac{\delta J}{\delta b_n}^{(FOSM)} &= \frac{\delta F}{\delta b_n} + w\frac{\delta}{\delta b_n}\left(\sqrt{\frac{\delta F}{\delta c_m}\frac{\delta F}{\delta c_j}\mathcal{K}_{mj}}\right) \\
&= \frac{\delta F}{\delta b_n} + \frac{w}{2\sqrt{\frac{\delta F}{\delta c_m}\frac{\delta F}{\delta c_j}\mathcal{K}_{mj}}}\frac{\delta}{\delta b_n}\left(\frac{\delta F}{\delta c_m}\frac{\delta F}{\delta c_j}\mathcal{K}_{mj}\right) \\
&= \frac{\delta F}{\delta b_n} + \frac{w}{2\sigma_F^{(FOSM)}}\left[\left(\frac{\delta^2 F}{\delta c_m \delta b_n}\frac{\delta F}{\delta c_j} + \frac{\delta F}{\delta c_m}\frac{\delta^2 F}{\delta c_j \delta b_n}\right)\mathcal{K}_{mj}\right] \\
&= \frac{\delta F}{\delta b_n} + \frac{w}{\sigma_F^{(FOSM)}}\frac{\delta^2 F}{\delta c_m \delta b_n}\frac{\delta F}{\delta c_j}\mathcal{K}_{mj} \\
&= \frac{\delta F}{\delta b_n} + \frac{w}{\sigma_F^{(FOSM)}} z_m \frac{\delta^2 F}{\delta c_m \delta b_n}
\end{aligned} \tag{4.18}$$

where

$$z_m^{(FOSM)} = \mathcal{K}_{mj}\frac{\delta F}{\delta c_j} \tag{4.19}$$

According to eq. 4.18, $\frac{\delta F}{\delta b_n}$ and $\frac{\delta^2 F}{\delta c_m \delta b_n}$ need to be computed in addition to $\frac{\delta F}{\delta c_m}$ which is involved in $\sigma_F^{(FOSM)}$. If all components in $\frac{\delta^2 F}{\delta c_m \delta b_n}$ were to be computed, using a combination of Direct Differentiation (DD) and adjoint techniques, the cost of the computation would scale with the minimum of M and N [119], which is usually equal to M in typical RDO problems in fluid mechanics (since $N \gg M$, in general). In [30], instead of $\frac{\delta^2 F}{\delta c_m \delta b_n}$, $z_m \frac{\delta^2 F}{\delta c_m \delta b_n}$ was computed directly at a cost that does not scale with M. The method was named projected FOSM (pFOSM) as it computes the projected matrix of mixed derivatives. This projection technique is described in section 4.2.2.

On the other hand, in PCE, by substituting eqs. 4.8 into eq. 4.17 one gets

$$\begin{aligned}\frac{\delta J}{\delta b_n}^{(PCE)} &= \frac{\delta \alpha_0}{\delta b_n} + w\frac{\delta}{\delta b_n}\left(\sqrt{\sum_{q=1}^{Q-1}\alpha_q^2\|\Psi_q\|^2}\right)\\ &= \frac{\delta \alpha_0}{\delta b_n} + \frac{w}{2\sqrt{\sum_{q=1}^{Q-1}\alpha_q^2\|\Psi_q\|^2}}\frac{\delta}{\delta b_n}\left(\sum_{q=1}^{Q-1}\alpha_q^2\|\Psi_q\|^2\right)\\ &= \frac{\delta \alpha_0}{\delta b_n} + \frac{w}{2\sigma_F^{(PCE)}}\sum_{q=1}^{Q-1}\left(2\alpha_q\|\Psi_q\|^2\frac{\delta\alpha_q}{\delta b_n}\right)\\ &= \frac{\delta \alpha_0}{\delta b_n} + w\sum_{q=1}^{Q-1}\left(\frac{\alpha_q\|\Psi_q\|^2}{\sigma_F^{(PCE)}}\frac{\delta\alpha_q}{\delta b_n}\right)\end{aligned} \tag{4.20}$$

The unknown quantities $\frac{\delta\alpha_q}{\delta b_n}$ can be computed by differentiating eq. 4.15 w.r.t. b_n, yielding

$$\begin{aligned}\frac{\delta\boldsymbol{\alpha}}{\delta b_n} = G\frac{\delta\boldsymbol{\beta}}{\delta b_n} &= \left(D^T D\right)^{-1} D^T\frac{\delta\boldsymbol{\beta}}{\delta b_n}\\ &= \left(\begin{bmatrix}A^T & C^T\end{bmatrix}\begin{bmatrix}A\\ C\end{bmatrix}\right)^{-1}\begin{bmatrix}A^T & C^T\end{bmatrix}\begin{bmatrix}\frac{\delta\boldsymbol{f}}{\delta b_n}\\ \frac{\delta\boldsymbol{g}}{\delta b_n}\end{bmatrix}\end{aligned} \tag{4.21}$$

where

$$\frac{\delta\boldsymbol{f}}{\delta b_n} = \begin{bmatrix}\frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(1)})\\ \vdots\\ \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(L)})\end{bmatrix}, \quad \frac{\delta\boldsymbol{g}}{\delta b_n} = \begin{bmatrix}\frac{\partial c_m}{\partial\zeta_1}\frac{\delta^2 F}{\delta c_m\delta b_n}(\boldsymbol{c}^{(1)})\\ \vdots\\ \frac{\partial c_m}{\partial\zeta_M}\frac{\delta^2 F}{\delta c_m\delta b_n}(\boldsymbol{c}^{(1)})\\ \vdots\\ \frac{\partial c_m}{\partial\zeta_1}\frac{\delta^2 F}{\delta c_m\delta b_n}(\boldsymbol{c}^{(L)})\\ \vdots\\ \frac{\partial c_m}{\partial\zeta_M}\frac{\delta^2 F}{\delta c_m\delta b_n}(\boldsymbol{c}^{(L)})\end{bmatrix} \tag{4.22}$$

To compute $\frac{\delta\alpha_q}{\delta b_n}$, both $\frac{\delta F}{\delta b_n}$ and $\frac{\delta^2 F}{\delta c_m\delta b_n}$ are required for each value-set of c_m. In PCE, the C and $\frac{\delta\boldsymbol{g}}{\delta b_n}$ blocks are absent from eq. 4.21 and, thus, only $\frac{\delta F}{\delta b_n}$ need be computed for each value-set of c_m. In APCE however, derivatives $\frac{\delta F}{\delta c_m}$ and $\frac{\delta^2 F}{\delta c_m\delta b_n}$ are additionally needed for each value-set of c_m, for the current value-set of b_n, in order to compute $\frac{\delta J}{\delta b_n}$.

For $\frac{\delta F}{\delta b_n}$, the adjoint method is used, as in chapter 2. In section 4.2.1,

adjustments are made to the adjoint formulation of chapter 2 for the computation of $\frac{\delta F}{\delta c_m}$. Regarding the appearance of $\frac{\delta^2 F}{\delta c_m \delta b_n}$ in eq. 4.21, the projection technique is used to overcome the computation of the mixed derivatives matrix. To this end, eqs. 4.20 and 4.21 are developed further in section 4.2.2 so as to derive expressions for $z_m \frac{\delta^2 F}{\delta c_m \delta b_n}$ products for each collocation point, much like the ones in eq. 4.18. The proposed projected APCE (pAPCE) method benefits from similar matrix-vector products as pFOSM. Note that the projection vector z_m is different between pFOSM and pAPCE.

4.2.1 Computation of First-order Derivatives

The first-order derivatives of F w.r.t. both b_n and c_m are computed by solving the same adjoint equations, with the same boundary conditions at the cost of 1 EFS per collocation point. The method was presented in chapter 2 for the SDs of F w.r.t. $\boldsymbol{b}$. For the SDs of F w.r.t. $\boldsymbol{c}$, eq. 2.7 is differentiated w.r.t. c_m,

$$\frac{\delta F}{\delta c_m} = \int_S \left(\frac{\partial F_{S,i}}{\partial v_k} \frac{\delta v_k}{\delta c_m} + \frac{\partial F_{S,i}}{\partial p} \frac{\delta p}{\delta c_m} + \frac{\partial F_{S,i}}{\partial \tau_{kj}} \frac{\delta \tau_{kj}}{\delta c_m} + \frac{\partial F_{S,i}}{\partial \widetilde{\nu}} \frac{\delta \widetilde{\nu}}{\delta c_m} + \frac{\partial F_{S,i}}{\partial c_m} \right) n_i dS \tag{4.23}$$

The Lagrangian in eq. 2.9 is differentiated w.r.t. c_m and the procedure described in section 2.4 is again followed, leading to

$$\begin{aligned} \frac{\delta L}{\delta c_m} = & \int_\Omega \mathcal{R}^q \frac{\delta p}{\delta c_m} d\Omega + \int_\Omega \mathcal{R}_i^{\boldsymbol{u}} \frac{\delta v_i}{\delta c_m} d\Omega + \int_\Omega \mathcal{R}^{\widetilde{\nu_\alpha}} \frac{\delta \widetilde{\nu}}{\delta c_m} d\Omega \\ & + \int_S \mathcal{BC}^p \frac{\delta p}{\delta c_m} dS + \int_S \mathcal{BC}_i^{\boldsymbol{v}} \frac{\delta v_i}{\delta c_m} dS + \int_S \mathcal{BC}_{ij}^\tau \frac{\delta \tau_{ij}}{\delta c_m} dS + \int_S \mathcal{BC}^{\widetilde{\nu}} \frac{\delta \widetilde{\nu}}{\delta c_m} dS \\ & + \int_S \frac{\partial F_{S,i}}{\partial c_m} n_i dS - \int_S \frac{\widetilde{\nu}_\alpha (\nu + \widetilde{\nu})}{c_\sigma} n_i \frac{\delta}{\delta c_m} \left(\frac{\partial \widetilde{\nu}}{\partial x_i} \right) dS \end{aligned} \tag{4.24}$$

The derivatives of Δ w.r.t. c_m are absent since, in the present cases, $\boldsymbol{c}$ does not affect the shape of any boundary. After the elimination of the volume integrals in eq. 4.24, through the solution of the adjoint equations, this expression becomes

$$\begin{aligned} \frac{\delta L}{\delta c_m} = & \int_S \mathcal{BC}^p \frac{\delta p}{\delta c_m} dS + \int_S \mathcal{BC}_i^{\boldsymbol{v}} \frac{\delta v_i}{\delta c_m} dS + \int_S \mathcal{BC}_{ij}^\tau \frac{\delta \tau_{ij}}{\delta c_m} dS + \int_S \mathcal{BC}^{\widetilde{\nu}} \frac{\delta \widetilde{\nu}}{\delta c_m} dS \\ & + \int_S \frac{\partial F_{S,i}}{\partial c_m} n_i dS - \int_S \frac{\widetilde{\nu}_\alpha (\nu + \widetilde{\nu})}{c_\sigma} n_i \frac{\delta}{\delta c_m} \left(\frac{\partial \widetilde{\nu}}{\partial x_i} \right) dS \end{aligned} \tag{4.25}$$

The $\mathcal{BC}$ terms on the r.h.s. of eq. 4.25 are the same ones as in eqs. 2.19a to 2.19d and the same adjoint BCs are used as the ones in eqs. 2.20 for the solution of the adjoint equations, derived in section 2.6. Through these BCs, the surface integrals in eq. 4.25 are eliminated across boundaries where the derivatives of the flow variables w.r.t. c_m cannot be computed. For the remaining integrals,

these either vanish (in case the flow equations' BCs are not affected by c_m) or (across boundaries where the relation between the flow and uncertain variables is known) these are directly computed. Expressions for particular QoIs and types of uncertain variables are given in Appendix A.

4.2.2 The Projection Technique

The projection technique which overcomes the computation of the mixed derivatives' matrix is presented in this section. Since, in this thesis, the uncertain variables do not affect any geometrical quantity, the differentiation of eq. 2.21 w.r.t. c_m, yields

$$\begin{aligned}\left.\frac{\delta^2 F}{\delta c_m \delta b_n}\right|_{FI} &= \int_{S_{W_p}} \frac{\delta}{\delta c_m}\left(\frac{\partial F_{S,i}}{\partial x_k}\right) n_i \frac{\delta x_k}{\delta b_n} dS + \int_{S_{W_p}} \frac{\delta F_{S,i}}{\delta c_m} \frac{\delta(n_i dS)}{\delta b_n} \\ &+ \int_{\Omega} \left(\frac{\delta \mathcal{A}^L_{jk}}{\delta c_m} + \frac{\delta \mathcal{A}^T_{jk}}{\delta c_m}\right) \frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right) d\Omega \\ &+ \int_{S_{W_p}} \frac{\delta}{\delta c_m}\left[\left(u_{\langle n\rangle} - \phi_{\langle n\rangle\langle n\rangle}\right)\tau_{ij}\right] \frac{\delta(n_i n_j)}{\delta b_n} dS \\ &- \int_{S_{W_p}} \frac{\delta}{\delta c_m}\left(\phi_{\langle t^I\rangle\langle t^I\rangle}\tau_{ij}\right) \frac{\delta(t^I_i t^I_j)}{\delta b_n} dS - \int_{S_{W_p}} \frac{\delta}{\delta c_m}\left(\phi_{\langle t^{II}\rangle\langle t^{II}\rangle}\tau_{ij}\right) \frac{\delta(t^{II}_i t^{II}_j)}{\delta b_n} dS \\ &- \int_{S_{W_p}} \frac{\delta}{\delta c_m}\left[\left(\phi_{\langle t^{II}\rangle\langle t^I\rangle} + \phi_{\langle t^I\rangle\langle t^{II}\rangle}\right)\tau_{ij}\right] \frac{\delta(t^{II}_i t^I_j)}{\delta b_n} dS \end{aligned} \tag{4.26}$$

for the FI adjoint, and

$$\begin{aligned}\left.\frac{\delta^2 F}{\delta c_m \delta b_n}\right|_{E-SI} &= \int_{S_{W_p}} \frac{\delta}{\delta c_m}\left(\frac{\partial F_{S,i}}{\partial x_k}\right) n_i \frac{\delta x_k}{\delta b_n} dS + \int_{S_{W_p}} \frac{\delta F_{S,i}}{\delta c_m} \frac{\delta(n_i dS)}{\delta b_n} \\ &+ \int_{S_{W_p}} \left(\frac{\delta \mathcal{A}^L_{jk}}{\delta c_m} + \frac{\delta \mathcal{A}^T_{jk}}{\delta c_m}\right) n_j \frac{\delta x_k}{\delta b_n} dS - \int_{S_{W_p}} \frac{\delta}{\delta c_m}\left(\frac{\partial m^\alpha_k}{\partial x_j}\right) n_j \frac{\delta x_k}{\delta b_n} dS \\ &+ \int_{S_{W_p}} \frac{\delta}{\delta c_m}\left[\left(u_{\langle n\rangle} - \phi_{\langle n\rangle\langle n\rangle}\right)\tau_{ij}\right] \frac{\delta(n_i n_j)}{\delta b_n} dS \\ &- \int_{S_{W_p}} \frac{\delta}{\delta c_m}\left(\phi_{\langle t^I\rangle\langle t^I\rangle}\tau_{ij}\right) \frac{\delta(t^I_i t^I_j)}{\delta b_n} dS - \int_{S_{W_p}} \frac{\delta}{\delta c_m}\left(\phi_{\langle t^{II}\rangle\langle t^{II}\rangle}\tau_{ij}\right) \frac{\delta(t^{II}_i t^{II}_j)}{\delta b_n} dS \\ &- \int_{S_{W_p}} \frac{\delta}{\delta c_m}\left[\left(\phi_{\langle t^{II}\rangle\langle t^I\rangle} + \phi_{\langle t^I\rangle\langle t^{II}\rangle}\right)\tau_{ij}\right] \frac{\delta(t^{II}_i t^I_j)}{\delta b_n} dS \end{aligned} \tag{4.27}$$

for the E-SI adjoint. In eqs. 4.26 and 4.27, the following terms result from the differentiation of eqs. 2.17a and 2.17b w.r.t. c_m

$$\begin{aligned}\frac{\delta \mathcal{A}^L_{jk}}{\delta c_m} =& \frac{\delta q}{\delta c_m}\frac{\partial v_j}{\partial x_k} + q\frac{\partial}{\partial x_k}\left(\frac{\delta v_j}{\delta c_m}\right) - v_j\frac{\delta u_i}{\delta c_m}\frac{\partial v_i}{\partial x_k} - u_i\frac{\partial v_i}{\partial x_k}\frac{\delta v_j}{\delta c_m} - u_i v_j\frac{\partial}{\partial x_k}\left(\frac{\delta v_i}{\delta c_m}\right)\\ &-\frac{\delta \tau^\alpha_{ij}}{\delta c_m}\frac{\partial v_i}{\partial x_k} - \tau^\alpha_{ij}\frac{\partial}{\partial x_k}\left(\frac{\delta v_i}{\delta c_m}\right) + \frac{\delta u_i}{\delta c_m}\frac{\partial \tau_{ij}}{\partial x_k} + u_i\frac{\partial}{\partial x_k}\left(\frac{\delta \tau_{ij}}{\delta c_m}\right)\\ &-\frac{\delta u_j}{\delta c_m}\frac{\partial p}{\partial x_k} - u_j\frac{\partial}{\partial x_k}\left(\frac{\delta p}{\delta c_m}\right)\end{aligned} \tag{4.28a}$$

$$\begin{aligned}\frac{\delta \mathcal{A}^T_{jk}}{\delta c_m} =& -\frac{\delta \widetilde{\nu_\alpha}}{\delta c_m} v_j\frac{\partial \widetilde{\nu}}{\partial x_k} - \widetilde{\nu_\alpha}\frac{\delta v_j}{\delta c_m}\frac{\partial \widetilde{\nu}}{\partial x_k} - \widetilde{\nu_\alpha} v_j\frac{\partial}{\partial x_k}\left(\frac{\delta \widetilde{\nu}}{\delta c_m}\right) + \frac{\delta \widetilde{\nu_\alpha}}{\delta c_m}\frac{\partial}{\partial x_k}\left(\frac{\nu+\widetilde{\nu}}{\sigma}\frac{\partial \widetilde{\nu}}{\partial x_j}\right)\\ &+\widetilde{\nu_\alpha}\frac{\partial}{\partial x_k}\left[\frac{1}{\sigma}\left(\frac{\delta \nu}{\delta c_m} + \frac{\delta \widetilde{\nu}}{\delta c_m}\right)\frac{\partial \widetilde{\nu}}{\partial x_j}\right] + \widetilde{\nu_\alpha}\frac{\partial}{\partial x_k}\left[\frac{\nu+\widetilde{\nu}}{\sigma}\frac{\partial}{\partial x_j}\left(\frac{\delta \widetilde{\nu}}{\delta c_m}\right)\right]\\ &-\frac{1}{\sigma}\left(\frac{\delta \nu}{\delta c_m} + \frac{\delta \widetilde{\nu}}{\delta c_m}\right)\frac{\partial \widetilde{\nu}}{\partial x_j}\frac{\partial \widetilde{\nu}}{\partial x_k} - \frac{\nu+\widetilde{\nu}}{\sigma}\frac{\partial}{\partial x_j}\left(\frac{\delta \widetilde{\nu_\alpha}}{\delta c_m}\right)\frac{\partial \widetilde{\nu}}{\partial x_k} - \frac{\nu+\widetilde{\nu}}{\sigma}\frac{\partial \widetilde{\nu_\alpha}}{\partial x_j}\frac{\partial}{\partial x_k}\left(\frac{\delta \widetilde{\nu}}{\delta c_m}\right)\\ &+\frac{2c_{b_2}}{\sigma}\frac{\delta \widetilde{\nu_\alpha}}{\delta c_m}\frac{\partial \widetilde{\nu}}{\partial x_j}\frac{\partial \widetilde{\nu}}{\partial x_k} + \frac{2c_{b_2}\widetilde{\nu_\alpha}}{\sigma}\frac{\partial}{\partial x_j}\left(\frac{\delta \widetilde{\nu}}{\delta c_m}\right)\frac{\partial \widetilde{\nu}}{\partial x_k} + \frac{2c_{b_2}\widetilde{\nu_\alpha}}{\sigma}\frac{\partial \widetilde{\nu}}{\partial x_j}\frac{\partial}{\partial x_k}\left(\frac{\delta \widetilde{\nu}}{\delta c_m}\right)\\ &-\frac{1}{\omega}\left[\left(\frac{\delta \widetilde{\nu_\alpha}}{\delta c_m}\widetilde{\nu} + \widetilde{\nu_\alpha}\frac{\delta \widetilde{\nu}}{\delta c_m}\right)C_\omega Y_q + \widetilde{\nu_\alpha}\widetilde{\nu}\left(Y_q\frac{\delta C_\omega}{\delta c_m} + C_\omega\frac{\delta Y_q}{\delta c_m} - \frac{C_\omega Y_q}{\omega^2}Y_l\frac{\delta Y_l}{\delta c_m}\right)\right]\epsilon_{qji}\frac{\partial v_i}{\partial x_k}\\ &-\widetilde{\nu_\alpha}\widetilde{\nu}\frac{C_\omega}{\omega}Y_q\epsilon_{qji}\frac{\partial}{\partial x_k}\left(\frac{\delta v_i}{\delta c_m}\right) - 2\frac{\delta \Delta_\alpha}{\delta c_m}\frac{\partial \Delta}{\partial x_j}\frac{\partial \Delta}{\partial x_j}\end{aligned} \tag{4.28b}$$

Equations 4.26 and 4.27 involve $\frac{\delta v_i}{\delta c_m}$, $\frac{\delta p}{\delta c_m}$, $\frac{\delta \widetilde{\nu}}{\delta c_m}$, $\frac{\delta u_i}{\delta c_m}$, $\frac{\delta q}{\delta c_m}$, $\frac{\delta \widetilde{\nu_\alpha}}{\delta c_m}$, $\frac{\delta \Delta_\alpha}{\delta c_m}$. The above-mentioned derivatives can be computed by differentiating the primal and adjoint equations w.r.t. c_m and solving the so-derived DD systems [30]. For each of the primal and adjoint systems of equations, M new systems are formulated in total, one for each c_m component.

Putting this into the context of (standard) APCE and recalling from eq. 4.22 that $\frac{\delta^2 F}{\delta c_m \delta b_n}$ are needed at the L collocation points, a cost of $2ML$ EFS would be required to compute them, over and above to the $2L$ EFS needed to compute $\frac{\delta F}{\delta b_n}$ and $\frac{\delta F}{\delta c_m}$. To obtain $\frac{\delta J}{\delta b_n}$, the total cost sums up to $(2+2M)L$ EFS. Inspired by pFOSM, this dependency on M is eliminated by computing projections onto suitable vectors, i.e. $z_m\frac{\delta^2 F}{\delta c_m \delta b_n}$, instead of $\frac{\delta^2 F}{\delta c_m \delta b_n}$. Indeed, the product of G and $\frac{\delta \boldsymbol{\beta}}{\delta b_n}$, after regrouping

the terms on the r.h.s. of eq. 4.21, yields

$$
\begin{aligned}
\frac{\delta \boldsymbol{\alpha}}{\delta b_n} = G\frac{\delta \boldsymbol{\beta}}{\delta b_n} &= \left(\begin{bmatrix} A^T & C^T \end{bmatrix}\begin{bmatrix} A \\ C \end{bmatrix}\right)^{-1}\begin{bmatrix} A^T & C^T \end{bmatrix}\begin{bmatrix} \frac{\delta \boldsymbol{f}}{\delta b_n} \\ \frac{\delta \boldsymbol{g}}{\delta b_n} \end{bmatrix} \\
&= \begin{bmatrix} \gamma_0^{(1)} & \cdots & \gamma_0^{(L)} & \kappa_{0,j}^{(1)} & \cdots & \kappa_{0,j}^{(L)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \gamma_{Q-1}^{(1)} & \cdots & \gamma_{Q-1}^{(L)} & \kappa_{Q-1,j}^{(1)} & \cdots & \kappa_{Q-1,j}^{(L)} \end{bmatrix} \begin{bmatrix} \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(1)}) \\ \vdots \\ \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(L)}) \\ \frac{\partial c_m}{\partial \zeta_j}\frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(1)}) \\ \vdots \\ \frac{\partial c_m}{\partial \zeta_j}\frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(L)}) \end{bmatrix} \\
&= \begin{bmatrix} \gamma_0^{(1)} & \cdots & \gamma_0^{(L)} \\ \vdots & \ddots & \vdots \\ \gamma_{Q-1}^{(1)} & \cdots & \gamma_{Q-1}^{(L)} \end{bmatrix} \begin{bmatrix} \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(1)}) \\ \vdots \\ \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(L)}) \end{bmatrix} + \begin{bmatrix} \kappa_{0,j}^{(1)} & \cdots & \kappa_{0,j}^{(L)} \\ \vdots & \ddots & \vdots \\ \kappa_{Q-1,j}^{(1)} & \cdots & \kappa_{Q-1,j}^{(L)} \end{bmatrix} \begin{bmatrix} \frac{\partial c_m}{\partial \zeta_j}\frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(1)}) \\ \vdots \\ \frac{\partial c_m}{\partial \zeta_j}\frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(L)}) \end{bmatrix} \\
&= \begin{bmatrix} \sum_{l=1}^{L} \gamma_0^{(l)} F(\boldsymbol{c}^{(l)}) + \sum_{l=1}^{L} \kappa_{0,j}^{(l)} \frac{\partial c_m}{\partial \zeta_j}\frac{\delta F}{\delta c_m}(\boldsymbol{c}^{(l)}) \\ \vdots \\ \sum_{l=1}^{L} \gamma_{Q-1}^{(l)} F(\boldsymbol{c}^{(l)}) + \sum_{l=1}^{L} \kappa_{Q-1,j}^{(l)} \frac{\partial c_m}{\partial \zeta_j}\frac{\delta F}{\delta c_m}(\boldsymbol{c}^{(l)}) \end{bmatrix}
\end{aligned}
\tag{4.29}
$$

Here, $\gamma_q^{(l)}$ and $\kappa_{qj}^{(l)}$ represent blocks of G that multiply $\frac{\delta F}{\delta b_n}\left(\boldsymbol{c}^{(l)}\right)$ and $\frac{\partial c_m}{\partial \zeta_j}\frac{\delta^2 F}{\delta c_m \delta b_n}\left(\boldsymbol{c}^{(l)}\right)$, respectively. Equation 4.29 can, then, be written as

$$
\frac{\delta \alpha_q}{\delta b_n} = \sum_{l=1}^{L}\left[\gamma_q^{(l)}\frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(l)}) + \kappa_{q,j}^{(l)}\frac{\partial c_m}{\partial \zeta_j}\frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(l)})\right] \tag{4.30}
$$

The mixed derivatives' matrix $\frac{\delta^2 F}{\delta c_m \delta b_n}$ is projected onto $\kappa_{qj}\frac{\partial c_m}{\partial \zeta_j}$ which differ for each value-set of $\boldsymbol{c}$ and each α_q component. Therefore, though the dependency on M is eliminated, the cost of computing $\kappa_{qj}\frac{\partial c_m}{\partial \zeta_j}\frac{\delta^2 F}{\delta c_m \delta b_n}$ still scales with Q. By substituting

eq. 4.30 into eq. 4.20, we get

$$
\begin{aligned}
\frac{\delta J}{\delta b_n} &= \frac{\delta \alpha_0}{\delta b_n} + \sum_{q=1}^{Q-1} \left(\frac{w\alpha_q \|\Psi_q\|^2}{\sigma_F^{(PCE)}} \frac{\delta \alpha_q}{\delta b_n} \right) \\
&= \sum_{l=1}^{L} \left[\gamma_0^{(l)} \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(l)}) + \kappa_{0,j}^{(l)} \frac{\partial c_m}{\partial \zeta_j} \frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(l)}) \right] \\
&+ \sum_{q=1}^{Q-1} \left(\frac{w\alpha_q \|\Psi_q\|^2}{\sigma_F^{(PCE)}} \sum_{l=1}^{L} \left[\gamma_q^{(l)} \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(l)}) + \kappa_{q,j}^{(l)} \frac{\partial c_m}{\partial \zeta_j} \frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(l)}) \right] \right) \\
&= \sum_{l=1}^{L} \left[\gamma_0^{(l)} \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(l)}) + \kappa_{0,j}^{(l)} \frac{\partial c_m}{\partial \zeta_j} \frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(l)}) \right] \\
&+ \sum_{l=1}^{L} \left[\sum_{q=1}^{Q-1} \left(\frac{w\alpha_q \|\Psi_q\|^2}{\sigma_F^{(PCE)}} \left\{ \gamma_q^{(l)} \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(l)}) + \kappa_{q,j}^{(l)} \frac{\partial c_m}{\partial \zeta_j} \frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(l)}) \right\} \right) \right] \\
&= \sum_{l=1}^{L} \left[\gamma_0^{(l)} \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(l)}) \right] + \sum_{l=1}^{L} \left[\kappa_{0,j}^{(l)} \frac{\partial c_m}{\partial \zeta_j} \frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(l)}) \right] \\
&+ \sum_{l=1}^{L} \left[\sum_{q=1}^{Q-1} \left(\frac{w\alpha_q \|\Psi_q\|^2}{\sigma_F^{(PCE)}} \gamma_q^{(l)} \right) \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(l)}) \right] + \sum_{l=1}^{L} \left[\sum_{q=1}^{Q-1} \left(\frac{w\alpha_q \|\Psi_q\|^2}{\sigma_F^{(PCE)}} \kappa_{q,j}^{(l)} \right) \frac{\partial c_m}{\partial \zeta_j} \frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(l)}) \right] \\
&= \sum_{l=1}^{L} \left[\gamma_0^{(l)} \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(l)}) + \sum_{q=1}^{Q-1} \left(\frac{w\alpha_q \|\Psi_q\|^2}{\sigma_F^{(PCE)}} \gamma_q^{(l)} \right) \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(l)}) \right] \\
&+ \sum_{l=1}^{L} \left[\kappa_{0,j}^{(l)} \frac{\partial c_m}{\partial \zeta_j} \frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(l)}) + \sum_{q=1}^{Q-1} \left(\frac{w\alpha_q \|\Psi_q\|^2}{\sigma_F^{(PCE)}} \kappa_{q,j}^{(l)} \right) \frac{\partial c_m}{\partial \zeta_j} \frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(l)}) \right] \\
&= \sum_{l=1}^{L} \left[\left\{ \gamma_0^{(l)} + \sum_{q=1}^{Q-1} \left(\frac{w\alpha_q \|\Psi_q\|^2}{\sigma_F^{(PCE)}} \gamma_q^{(l)} \right) \right\} \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(l)}) \right] \\
&+ \sum_{l=1}^{L} \left[\left\{ \kappa_{0,j}^{(l)} + \sum_{q=1}^{Q-1} \left(\frac{w\alpha_q \|\Psi_q\|^2}{\sigma_F^{(PCE)}} \kappa_{q,j}^{(l)} \right) \right\} \frac{\partial c_m}{\partial \zeta_j} \frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(l)}) \right] \\
&= \sum_{l=1}^{L} k^{(l)} \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(l)}) + \sum_{l=1}^{L} z_m^{(l)} \frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(l)}) \qquad (4.31)
\end{aligned}
$$

where $k^{(l)}$ and $z_m^{(l)}$ are quantities and projection vectors' components (different

from the one used in pFOSM), respectively, and read

$$k^{(l)} = \left\{ \gamma_0^{(l)} + \sum_{q=1}^{Q-1} \left(\frac{w\alpha_q \|\Psi_q\|^2}{\sigma_F^{(PCE)}} \gamma_q^{(l)} \right) \right\} \tag{4.32a}$$

$$z_m^{(l)} = \left\{ \kappa_{0,j}^{(l)} + \sum_{q=1}^{Q-1} \left(\frac{w\alpha_q \|\Psi_q\|^2}{\sigma_F^{(PCE)}} \kappa_{q,j}^{(l)} \right) \right\} \frac{\partial c_m}{\partial \zeta_j} \tag{4.32b}$$

In eq. 4.31, $\frac{\delta^2 F}{\delta c_m \delta b_n}$ are projected onto z_m and $z_m \frac{\delta^2 F}{\delta c_m \delta b_n}$ can be computed at a cost that is independent of both M and Q, in much the same way as in pFOSM [30].

Guided by eq. 4.31, if $\widehat{\psi} = z_m \frac{\delta \psi}{\delta c_m}$, the projected mixed derivatives' matrix becomes

$$\begin{aligned} z_m \frac{\delta^2 F}{\delta c_m \delta b_n}\bigg|_{FI} = & \int_{S_{W_p}} \frac{\partial \widehat{F}_{S,i}}{\partial x_k} n_i \frac{\delta x_k}{\delta b_n} dS + \int_{S_{W_p}} \widehat{F}_{S,i} \frac{\delta(n_i dS)}{\delta b_n} + \int_\Omega \left(\widehat{\mathcal{A}}_{jk}^L + \widehat{\mathcal{A}}_{jk}^T \right) \frac{\partial}{\partial x_j} \left(\frac{\delta x_k}{\delta b_n} \right) d\Omega \\ & + \int_{S_{W_p}} \left[\left(\widehat{u}_{\langle n \rangle} - \widehat{\phi}_{\langle n \rangle \langle n \rangle} \right) \tau_{ij} + \left(u_{\langle n \rangle} - \phi_{\langle n \rangle \langle n \rangle} \right) \widehat{\tau}_{ij} \right] \frac{\delta(n_i n_j)}{\delta b_n} dS \\ & - \int_{S_{W_p}} \left(\widehat{\phi}_{\langle t^I \rangle \langle t^I \rangle} \tau_{ij} + \phi_{\langle t^I \rangle \langle t^I \rangle} \widehat{\tau}_{ij} \right) \frac{\delta(t_i^I t_j^I)}{\delta b_n} dS \\ & - \int_{S_{W_p}} \left(\widehat{\phi}_{\langle t^{II} \rangle \langle t^{II} \rangle} \tau_{ij} + \phi_{\langle t^{II} \rangle \langle t^{II} \rangle} \widehat{\tau}_{ij} \right) \frac{\delta(t_i^{II} t_j^{II})}{\delta b_n} dS \\ & - \int_{S_{W_p}} \left[\left(\widehat{\phi}_{\langle t^{II} \rangle \langle t^I \rangle} + \widehat{\phi}_{\langle t^I \rangle \langle t^{II} \rangle} \right) \tau_{ij} + \left(\phi_{\langle t^{II} \rangle \langle t^I \rangle} + \phi_{\langle t^I \rangle \langle t^{II} \rangle} \right) \widehat{\tau}_{ij} \right] \frac{\delta(t_i^{II} t_j^I)}{\delta b_n} dS \end{aligned} \tag{4.33}$$

for the FI adjoint, and

$$
\begin{aligned}
z_m \frac{\delta^2 F}{\delta c_m \delta b_n}\bigg|_{E-SI} = & \int_{S_{W_p}} \frac{\partial \widehat{F}_{S,i}}{\partial x_k} n_i \frac{\delta x_k}{\delta b_n} dS + \int_{S_{W_p}} \widehat{F}_{S,i} \frac{\delta(n_i dS)}{\delta b_n} \\
& + \int_{S_{W_p}} \left(\widehat{\mathcal{A}}^L_{jk} + \widehat{\mathcal{A}}^T_{jk}\right) n_j \frac{\delta x_k}{\delta b_n} dS - \int_{S_{W_p}} \frac{\partial \widehat{m}^\alpha_k}{\partial x_j} n_j \frac{\delta x_k}{\delta b_n} dS \\
& + \int_{S_{W_p}} \left[\left(\widehat{u}_{\langle n\rangle} - \widehat{\phi}_{\langle n\rangle\langle n\rangle}\right)\tau_{ij} + \left(u_{\langle n\rangle} - \phi_{\langle n\rangle\langle n\rangle}\right)\widehat{\tau}_{ij}\right] \frac{\delta(n_i n_j)}{\delta b_n} dS \\
& - \int_{S_{W_p}} \left(\widehat{\phi}_{\langle t^I\rangle\langle t^I\rangle}\tau_{ij} + \phi_{\langle t^I\rangle\langle t^I\rangle}\widehat{\tau}_{ij}\right) \frac{\delta(t^I_i t^I_j)}{\delta b_n} dS \\
& - \int_{S_{W_p}} \left(\widehat{\phi}_{\langle t^{II}\rangle\langle t^{II}\rangle}\tau_{ij} + \phi_{\langle t^{II}\rangle\langle t^{II}\rangle}\widehat{\tau}_{ij}\right) \frac{\delta(t^{II}_i t^{II}_j)}{\delta b_n} dS \\
& - \int_{S_{W_p}} \left[\left(\widehat{\phi}_{\langle t^{II}\rangle\langle t^I\rangle} + \widehat{\phi}_{\langle t^I\rangle\langle t^{II}\rangle}\right)\tau_{ij} + \left(\phi_{\langle t^{II}\rangle\langle t^I\rangle} + \phi_{\langle t^I\rangle\langle t^{II}\rangle}\right)\widehat{\tau}_{ij}\right] \frac{\delta(t^{II}_i t^I_j)}{\delta b_n} dS
\end{aligned}
\tag{4.34}
$$

for the E-SI adjoint, wherein $\widehat{\mathcal{A}}^L_{jk}$ and $\widehat{\mathcal{A}}^T_{jk}$ depend on $\widehat{v}_i$, $\widehat{p}$, $\widehat{\tau}_{ij}$, $\widehat{\widetilde{\nu}}$, $\widehat{u}_i$, $\widehat{q}$, $\widehat{\widetilde{\nu}}_\alpha$, $\widehat{\tau}^\alpha_{ij}$, $\widehat{\Delta}_\alpha$. Applying the operator $z_m \frac{\delta}{\delta c_m}(.)$ to eqs. 2.1a to 2.1c and 2.18a to 2.18c yields the systems of *projected* DD primal and adjoint equations to be solved for the projections of the derivatives of the primal and adjoint variables. The system of projected DD primal equations (excluding the projected DD Hamilton-Jacobi equation which is not required since, herein, $\frac{\delta\Delta}{\delta c_m} = 0$) is

$$
\widehat{\mathcal{R}}^p = -\frac{\partial \widehat{v}_j}{\partial x_j} = 0 \tag{4.35a}
$$

$$
\widehat{\mathcal{R}}^v_i = \widehat{v}_j \frac{\partial v_i}{\partial x_j} + v_j \frac{\partial \widehat{v}_i}{\partial x_j} - \frac{\partial \widehat{\tau}_{ij}}{\partial x_j} + \frac{\partial \widehat{p}}{\partial x_i} = 0 \ , \qquad i = 1, 2 \tag{4.35b}
$$

$$
\begin{aligned}
\widehat{\mathcal{R}}^{\widetilde{\nu}} = & \widehat{v}_j \frac{\partial \widetilde{\nu}}{\partial x_j} + v_j \frac{\partial \widehat{\widetilde{\nu}}}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\frac{\nu + \widetilde{\nu}}{\sigma}\frac{\partial \widehat{\widetilde{\nu}}}{\partial x_j}\right] - \frac{\partial}{\partial x_j}\left[\frac{\widehat{\nu} + \widehat{\widetilde{\nu}}}{\sigma}\frac{\partial \widetilde{\nu}}{\partial x_j}\right] \\
& - \frac{2c_{b_2}}{\sigma}\frac{\partial \widetilde{\nu}}{\partial x_j}\frac{\partial \widehat{\widetilde{\nu}}}{\partial x_j} + (D - P)\widehat{\widetilde{\nu}} + (\widehat{D} - \widehat{P})\widetilde{\nu} = 0
\end{aligned}
\tag{4.35c}
$$

where the differentiation of the terms in eq. 2.2 results to

$$\widehat{P}=c_{b_1}\widehat{\widetilde{S}}\ ,\ \widehat{D}=\frac{c_{w_1}}{\Delta^2}\left(\widetilde{\nu}\widehat{f}_w+\widehat{\widetilde{\nu}}f_w\right)\ ,\ \widehat{f}_w=\frac{c_{w_3}^6}{g^6+c_{w_3}^6}\left(\frac{c_{w_3}^6+1}{g^6+c_{w_3}^6}\right)^{\frac{1}{6}}\widehat{g}$$

$$\widehat{g}=\left(1+6c_{w_2}d^5-c_{w_2}\right)\widehat{d}\ ,\ \widehat{d}=\frac{1}{\widetilde{S}\kappa^2\Delta^2}\left(\widehat{\widetilde{\nu}}-\frac{\widetilde{\nu}}{\widetilde{S}}\widehat{\widetilde{S}}\right)\ ,\ \widehat{\widetilde{S}}=\widehat{\omega}+\frac{1}{\kappa^2\Delta^2}\left(\widehat{f}_{v_2}\widetilde{\nu}+f_{v_2}\widehat{\widetilde{\nu}}\right)$$

$$\widehat{f}_{v_2}=\frac{1}{\left(1+\chi f_{v_1}\right)^2}\left(\chi^2\widehat{f}_{v_1}-\widehat{\chi}\right)\ ,\ \widehat{f}_{v_1}=3c_{v_1}^3\left(\frac{\chi}{\chi^3+c_{v_1}^3}\right)^2$$

$$\widehat{\chi}=\frac{1}{\nu}\widehat{\widetilde{\nu}}-\frac{\widetilde{\nu}}{\nu^2}\widehat{\nu}\ ,\ \widehat{\omega}=\frac{Y_i}{\omega}\widehat{Y}_i\ ,\ \widehat{Y}_i=\epsilon_{ijk}\frac{\partial\widehat{v}_k}{\partial x_j}$$

with BCs from the differentiation of eq. 2.3 w.r.t. c_m and the subsequent projection on z_m

$$\begin{aligned}
S_I &: \widehat{\boldsymbol{v}}=\widehat{v}_I,\quad \frac{\partial\widehat{p}}{\partial n}=0,\quad \widehat{\widetilde{\nu}}=\widehat{\widetilde{\nu}}_I\\
S_O &: \frac{\partial\widehat{v}_i}{\partial n}=0,\quad \widehat{p}=0,\quad \frac{\partial\widehat{\widetilde{\nu}}}{\partial n}=0\\
S_W &: \widehat{v}_i=0,\quad \frac{\partial\widehat{p}}{\partial n}=0,\quad \widehat{\widetilde{\nu}}=0\\
S_S &: \widehat{v}_{\langle n\rangle}=0,\quad \frac{\partial\widehat{v}_{\langle t^l\rangle}}{\partial n}=0 \text{ for } l=I,II,\quad \frac{\partial\widehat{p}}{\partial n}=0,\quad \frac{\partial\widehat{\widetilde{\nu}}}{\partial n}=0
\end{aligned}\tag{4.36}$$

The system of projected DD adjoint equations is

$$\widehat{\mathcal{R}}^{q} = -\frac{\partial \widehat{u}_j}{\partial x_j} = 0 \tag{4.37a}$$

$$\begin{aligned}
\widehat{\mathcal{R}}^{u}_{i} = {} & \widehat{u}_j \frac{\partial v_j}{\partial x_i} + u_j \frac{\partial \widehat{v}_j}{\partial x_i} - \widehat{v}_j \frac{\partial \widehat{u}_i}{\partial x_j} - v_j \frac{\partial \widehat{u}_i}{\partial x_j} - \frac{\partial \widehat{\tau}^{\alpha}_{ij}}{\partial x_j} + \frac{\partial \widehat{q}}{\partial x_i} + \widehat{\widetilde{\nu}}_{\alpha} \frac{\partial \widetilde{\nu}}{\partial x_i} + \widetilde{\nu}_{\alpha} \frac{\partial \widehat{\widetilde{\nu}}}{\partial x_i} \\
& - \frac{\partial}{\partial x_l} \left(\frac{1}{\omega} \left[\left(\widehat{\widetilde{\nu}}_{\alpha} \widetilde{\nu} + \widetilde{\nu}_{\alpha} \widehat{\widetilde{\nu}} \right) C_{\omega} + \widetilde{\nu}_{\alpha} \widetilde{\nu} \left(\widehat{C}_{\omega} - \frac{C_{\omega} \widehat{\omega}}{\omega} \right) \right] \epsilon_{qjk} \frac{\partial v_k}{\partial x_j} \epsilon_{qli} \right) \\
& - \frac{\partial}{\partial x_l} \left(\widetilde{\nu}_{\alpha} \widetilde{\nu} \frac{C_{\omega}}{\omega} \epsilon_{qjk} \frac{\partial \widehat{v}_k}{\partial x_j} \epsilon_{qli} \right) = 0 \quad i = 1, 2(, 3)
\end{aligned} \tag{4.37b}$$

$$\begin{aligned}
\widehat{\mathcal{R}}^{\widetilde{\nu}_{\alpha}} = {} & -\widehat{v}_j \frac{\partial \widetilde{\nu}_{\alpha}}{\partial x_j} - v_j \frac{\partial \widehat{\widetilde{\nu}}_{\alpha}}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(\frac{\widehat{\nu} + \widehat{\widetilde{\nu}}}{\sigma} \right) \frac{\partial \widetilde{\nu}_{\alpha}}{\partial x_j} \right] - \frac{\partial}{\partial x_j} \left[\left(\frac{\nu + \widetilde{\nu}}{\sigma} \right) \frac{\partial \widehat{\widetilde{\nu}}_{\alpha}}{\partial x_j} \right] \\
& + \frac{1}{\sigma} \frac{\partial \widehat{\widetilde{\nu}}_{\alpha}}{\partial x_j} \frac{\partial \widetilde{\nu}}{\partial x_j} + \frac{1}{\sigma} \frac{\partial \widetilde{\nu}_{\alpha}}{\partial x_j} \frac{\partial \widehat{\widetilde{\nu}}}{\partial x_j} + \frac{2 c_{b_2}}{\sigma} \frac{\partial}{\partial x_j} \left(\widehat{\widetilde{\nu}}_{\alpha} \frac{\partial \widetilde{\nu}}{\partial x_j} + \widetilde{\nu}_{\alpha} \frac{\partial \widehat{\widetilde{\nu}}}{\partial x_j} \right) \\
& + \left(\widehat{\widetilde{\nu}}_{\alpha} \widetilde{\nu} + \widetilde{\nu}_{\alpha} \widehat{\widetilde{\nu}} \right) C_{\widetilde{\nu}} + \widetilde{\nu}_{\alpha} \widetilde{\nu} \widehat{C}_{\widetilde{\nu}} \\
& + \frac{\partial \widehat{\nu}_t}{\partial \widetilde{\nu}} \frac{\partial u_i}{\partial x_j} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + \frac{\partial \nu_t}{\partial \widetilde{\nu}} \frac{\partial \widehat{u}_i}{\partial x_j} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \\
& + \frac{\partial \nu_t}{\partial \widetilde{\nu}} \frac{\partial u_i}{\partial x_j} \left(\frac{\partial \widehat{v}_i}{\partial x_j} + \frac{\partial \widehat{v}_j}{\partial x_i} \right) + (\widehat{D} - \widehat{P}) \widetilde{\nu}_{\alpha} + (D - P) \widehat{\widetilde{\nu}}_{\alpha} = 0
\end{aligned} \tag{4.37c}$$

$$\widehat{\mathcal{R}}^{\Delta_{\alpha}} = -2 \frac{\partial}{\partial x_j} \left(\widehat{\Delta}_{\alpha} \frac{\partial \Delta}{\partial x_j} \right) + \left(\widehat{\widetilde{\nu}} \widetilde{\nu}_{\alpha} + \widetilde{\nu} \widehat{\widetilde{\nu}}_{\alpha} \right) C_{\Delta} + \widetilde{\nu} \widetilde{\nu}_{\alpha} \widehat{C}_{\Delta} = 0 \tag{4.37d}$$

and the terms resulting from the differentiation of the terms in eq. 2.14

$$
\begin{aligned}
\widehat{C}_{\widetilde{\nu}} = & \frac{\widehat{C}_\omega}{\kappa^2\Delta^2}\left(f_{v_2}+\widetilde{\nu}\frac{\partial f_{v_2}}{\partial\widetilde{\nu}}\right)+\frac{C_\omega}{\kappa^2\Delta^2}\left[\widehat{f}_{v_2}+\widehat{\widetilde{\nu}}\frac{\partial f_{v_2}}{\partial\widetilde{\nu}}+z_m\frac{\delta}{\delta c_m}\left(\frac{\partial f_{v_2}}{\partial\widetilde{\nu}}\right)\widetilde{\nu}\right] \\
& +c_{w_1}\left(\frac{\widehat{d}}{\widetilde{\nu}}C+\frac{d}{\widetilde{\nu}}\widehat{C}-\frac{d}{\widetilde{\nu}^2}C\widehat{\widetilde{\nu}}\right)+\frac{c_{w_1}\widehat{f}_w}{\Delta^2} \\
\widehat{C}_\omega = & -c_{w_1}\left(\frac{\widehat{d}}{\widetilde{S}}C+\frac{d}{\widetilde{S}}\widehat{C}-\frac{d}{\widetilde{S}^2}C\widehat{\widetilde{S}}\right) \\
\widehat{C}_\Delta = & -\frac{2}{\Delta^3}\left[c_{w_1}\left(\widehat{\widetilde{\nu}}f_w+\widetilde{\nu}\widehat{f}_w\right)+c_{w_1}\Delta^2\left(\widehat{d}C+d\widehat{C}\right)\right] \\
& -\frac{2}{\Delta^3}\left[\frac{1}{\kappa^2}\left(\widehat{\widetilde{\nu}}f_{v_2}C_\omega+\widetilde{\nu}\widehat{f}_{v_2}C_\omega+\widetilde{\nu}f_{v_2}\widehat{C}_\omega\right)\right] \\
\widehat{C} = & \frac{1}{\Delta^2}\frac{c_{w_3}^6}{g(g+c_{w_3}^6)}\left(\widehat{\widetilde{\nu}}f_w+\widetilde{\nu}\widehat{f}_w-\widetilde{\nu}f_w\frac{\widehat{g}(2g+c_{w_3}^6)}{g(g+c_{w_3}^6)}\right)\left(1+6c_{w_2}d^5-c_{w_2}\right) \\
& +\frac{\widetilde{\nu}f_w}{\Delta^2}\frac{c_{w_3}^6}{g(g+c_{w_3}^6)}30c_{w_2}d^4\widehat{d} \\
z_m\frac{\delta}{\delta c_m}\left(\frac{\partial f_{v_2}}{\partial\widetilde{\nu}}\right) = & -\frac{\widehat{\nu}}{\nu}\frac{\partial f_{v_2}}{\partial\widetilde{\nu}}+\frac{6}{\nu}\frac{2c_{v_1}^3\chi^3-\left(\chi^3+c_{v_1}^3\right)\chi^2}{\left(1+\chi f_{v_1}\right)^2\left(\chi^3+c_{v_1}^3\right)^2}\widehat{\chi} \\
& -\frac{2}{\nu}\frac{3c_{v_1}^3\chi^4-\left(\chi^3+c_{v_1}^3\right)^2}{\left(1+\chi f_{v_1}\right)^2\left(\chi^3+c_{v_1}^3\right)^2}\left[\frac{\widehat{\chi}f_{v_1}+\chi\widehat{f}_{v_1}}{1+\chi f_{v_1}}+\frac{3\chi^2}{\chi^3+c_{v_1}^3}\widehat{\chi}\right]
\end{aligned}
$$

with BCs resulting from the differentiation of eq. 2.20

$$
\begin{aligned}
S_I : \widehat{u}_{\langle n\rangle} &= -\frac{\partial \widehat{F}_{S,i}}{\partial p} n_i, \quad \widehat{u}_{\langle t^l\rangle} = \frac{\partial \widehat{F}_{S,k}}{\partial \tau_{ij}} n_k t_i^l n_j + \frac{\partial \widehat{F}_{S,k}}{\partial \tau_{ij}} n_k t_j^l n_i \quad l = I, II \\
&\frac{\partial \widehat{q}}{\partial n} = 0, \quad \widehat{\widetilde{\nu}}_\alpha = 0, \quad \widehat{\Delta}_\alpha = 0 \\
S_O : v_{\langle n\rangle}\widehat{u}_{\langle t^l\rangle} + u_{\langle t^l\rangle}\widehat{v}_{\langle n\rangle} + \widehat{\tau}_{ij}^{\alpha} n_j t_i - &\left(\widetilde{\nu}_\alpha \widetilde{\nu} \frac{C_\omega}{\omega} \epsilon_{qjk} \frac{\partial \widehat{v}_k}{\partial x_j} \epsilon_{qli}\right) + \frac{\partial \widehat{F}_{S,j}}{\partial v_i} n_j t_i^l \\
&- \frac{1}{\omega}\left[\left(\widehat{\widetilde{\nu}}_\alpha \widetilde{\nu} + \widetilde{\nu}_\alpha \widehat{\widetilde{\nu}}\right) C_\omega + \widetilde{\nu}_\alpha \widetilde{\nu}\left(\widehat{C}_\omega - \frac{C_\omega \widehat{\omega}}{\omega}\right)\right] \epsilon_{qjk} \frac{\partial v_k}{\partial x_j} \epsilon_{qli} = 0, \quad l = I, II \\
&\frac{\partial \widehat{u}_{\langle n\rangle}}{\partial n} = 0, \quad \widehat{q} = v_{\langle n\rangle}\widehat{u}_{\langle n\rangle} + u_{\langle n\rangle}\widehat{v}_{\langle n\rangle} + 2\nu \frac{\partial \widehat{v}_{\langle n\rangle}}{\partial n} + \frac{\partial \widehat{F}_{S,j}}{\partial v_i} n_j n_i \\
&\widehat{\widetilde{\nu}}_\alpha v_{\langle n\rangle} + \widetilde{\nu}_\alpha \widehat{v}_{\langle n\rangle} + \left(\frac{\widehat{\nu} + \widehat{\widetilde{\nu}}}{\sigma}\right) \frac{\partial \widetilde{\nu}_\alpha}{\partial n} + \left(\frac{\nu + \widetilde{\nu}}{\sigma}\right) \frac{\partial \widehat{\widetilde{\nu}}_\alpha}{\partial n} = 0, \quad \widehat{\Delta}_\alpha = 0 \\
S_W : \widehat{u}_{\langle n\rangle} &= -\frac{\partial \widehat{F}_{S,i}}{\partial p} n_i, \quad \widehat{u}_{\langle t^l\rangle} = \frac{\partial \widehat{F}_{S,k}}{\partial \tau_{ij}} n_k t_i^l n_j + \frac{\partial \widehat{F}_{S,k}}{\partial \tau_{ij}} n_k t_j^l n_i \quad l = I, II \\
&\frac{\partial \widehat{q}}{\partial n} = 0, \quad \widehat{\widetilde{\nu}}_\alpha = 0, \quad \frac{\partial \widehat{\Delta}_\alpha}{\partial n} = 0 \\
S_S : \widehat{u}_{\langle n\rangle} &= 0, \quad \frac{\partial \widehat{u}_{\langle t^l\rangle}}{\partial n} = 0 \quad l = I, II, \quad \frac{\partial \widehat{q}}{\partial n} = 0, \quad \frac{\partial \widehat{\widetilde{\nu}}_\alpha}{\partial n} = 0, \quad \frac{\partial \widehat{\Delta}_\alpha}{\partial n} = 0
\end{aligned}
\tag{4.38}
$$

Equations 4.35a to 4.35c are solved for $\widehat{v}_i$, $\widehat{p}$ and $\widehat{\widetilde{\nu}}$ whereas eqs. 4.37a to 4.37d for $\widehat{u}_i$, $\widehat{q}$, $\widehat{\widetilde{\nu}}_\alpha$ and $\widehat{\Delta}_\alpha$ with $\widehat{\tau}_{ij} = (\nu + \nu_t)\left(\frac{\partial \widehat{v}_i}{\partial x_j} + \frac{\partial \widehat{v}_j}{\partial x_i}\right)$ and $\widehat{\tau}_{ij}^{\alpha} = (\nu + \nu_t)\left(\frac{\partial \widehat{u}_i}{\partial x_j} + \frac{\partial \widehat{u}_j}{\partial x_i}\right)$. The additional terms in eqs. 4.35a to 4.35c and 4.37a to 4.37d, compared to the corresponding ones in [30], originate from the differentiation of the terms in the Spalart-Allmaras model, its adjoint and the adjoint to the Hamilton-Jacobi equation. Through this inclusion, both pAPCE and pFOSM can be used for turbulent flows resolved via this turbulence model. In the E-SI adjoint, the following equation is additionally solved

$$
\widehat{\mathcal{R}}_k^{m^\alpha} = \frac{\partial^2 \widehat{m}_k^\alpha}{\partial x_j^2} - \frac{\partial}{\partial x_j}\left(\widehat{\mathcal{A}}_{jk}^L + \widehat{\mathcal{A}}_{jk}^T\right) \tag{4.39}
$$

with BCs

$$
\widehat{m}_i^\alpha = 0 \tag{4.40}
$$

along all boundaries.

4.2.3 Overview of the Gradient-based RDO Using pFOSM and pAPCE

To assess the cost of performing the RDO with the pFOSM and pAPCE methods, the following must be taken into account.

Firstly, the mathematical development of previous sections assumes that only one adjoint system of equations needs to be solved which, as it will be seen in the following sections, is not the case when additional flow-related constraints are imposed. If n_α adjoint problems are to be solved, the cost per cycle of pFOSM is equal to $1+3n_\alpha$ EFS (1 primal, n_α adjoint, $2n_\alpha$ DD-then-projected primal and adjoint systems of equations).

In the case of pAPCE, the cost per cycle is equal to $(1+3n_\alpha)L$, and L should be at least such that Q equations are assembled for the solution of eq. 4.15. Since Q increases with M and O_{PCE}, so does L and, hence, the cost of performing the RDO with pAPCE. Usually, in regression-based PCE, an oversampling ratio r_{os} is chosen so that $r_{os}Q$ equations become available for least-squares regression [45, 51, 132, 4, 32, 148, 167]. This ratio may vary depending on the case; for example [51, 167] suggest $r_{os} \geq 2$ to achieve acceptable accuracy, although an even lower ratio might yield acceptable results. Consequently, without gradient information, in standard PCE, $L = r_{os}\frac{(O_{PCE}+M)!}{O_{PCE}!M!}$, whereas with gradient information in the APCE, $L = \lceil \frac{r_{os}}{M+1}\frac{(O_{PCE}+M)!}{O_{PCE}!M!} \rceil$ [1] and, as such, the APCE reduces the total number of EFS required for UQ. Thus far, the drawback of its use in the RDO was the emergence of the mixed derivatives, the computational cost of which was alleviated by using the pAPCE and making the cost of computing $z_m \frac{\delta^2 F}{\delta c_m \delta b_n}$ independent of M and N, for each collocation point. Without the projection technique, the cost of using APCE in the RDO would have been equal to $(1+n_\alpha)(1+M)L$ EFS.

Evidently, the greater the value of M, the greater the gain from the projection technique in either method. An example for this is shown in fig. 4.3.

The algorithms for pFOSM and pAPCE are illustrated below:

[1] $\lceil . \rceil$ denotes the ceiling operation

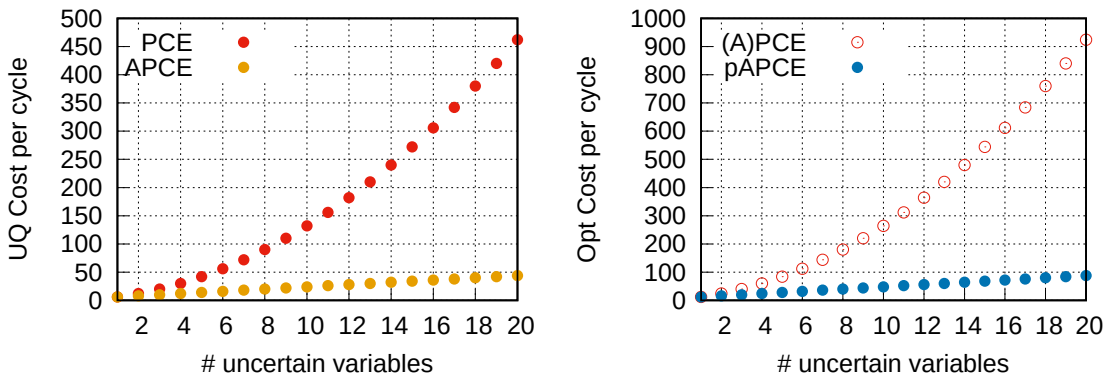


Figure 4.3: Left: UQ cost per cycle with standard PCE and APCE as a function of M. **Right**: RDO cost with standard PCE, APCE without the projection technique implemented in pAPCE. The y-axis is measured in EFS and the cost scales with M. These costs correspond to $r_{os} = 5$ for the least-squares system computing the PCE coefficients, $O_{PCE} = 2$ and $n_\alpha = 1$. Without the projection of pAPCE, the UQ cost-benefits of APCE are counteracted in RDO. The assumption made is that the number of uncertain variables is greater than that of the QoIs.

Algorithm 3 pFOSM

Initialize b_n
repeat
Solve Primal equations (1 EFS)
Solve Adjoint equations (n_α EFS) and compute $\frac{\delta F}{\delta b_n}$, $\frac{\delta F}{\delta c_m}$
Compute μ_F and σ_F via eqs. 4.5 and z_m in eq. 4.18
Solve DD Primal equations (n_α EFS)
Solve DD Adjoint equations (n_α EFS)
Compute $z_m \frac{\delta^2 F}{\delta c_m \delta b_n}$, eqs. 4.33
Compute $\frac{\delta J}{\delta b_n}$, eq. 4.18
Update b_n
until convergence

Algorithm 4 pAPCE

Initialize b_n and L sets of c_m
repeat
for $l \leftarrow 1, L$ **do**
    Solve Primal equations (1 EFS)
end for
for $l \leftarrow 1, L$ **do**
    Solve Adjoint equations (n_α EFS) and compute $\frac{\delta F}{\delta b_n}(l), \frac{\delta F}{\delta c_m}(l)$
end for
Solve system 4.13 and compute μ_F and σ_F via eqs. 4.8 and $z_m^{(l)}$ in eq. 4.31
for $l \leftarrow 1, L$ **do**
    Solve DD Primal equations (n_α EFS)
    Solve DD Adjoint equations (n_α EFS)
    Compute $z_m \frac{\delta^2 F}{\delta c_m \delta b_n}(l)$, eqs. 4.33
end for
Compute $\frac{\delta J}{\delta b_n}$, eq. 4.20
Update b_n
until convergence

4.3 Robust Design Optimization Applications

A series of shape optimization problems, including uncertainties, are solved using both pFOSM and pAPCE. Some aspects of both methods are demonstrated and discussed. Validations of the methods are shown for computing SDs for all parameterization techniques used.

First, airfoil shape optimization problems are solved with C_D as the QoI. In all cases, uncertainties are related to the farfield flow velocity magnitude v_∞ and angle α_∞ and both follow normal distributions. Additional constraints are imposed on μ_{C_L} and the cross-sectional area of the airfoil V. In one case, the mean value of the pitching moment coefficient (μ_{C_M}) is also constrained. Target values for these constraints are denoted by the superscript $^\oplus$.

The first case deals with the use of the developed pAPCE method. Some aspects of its use, such as the choice of O_{PCE} and L are discussed and a parametric study for both is carried out. The value of L directly affects the cost of (p)APCE. Code verification for QoI SDs w.r.t. c_m and statistical moments' SDs w.r.t. b_n follows for turbulent flows. Additionally, it is also of interest to assess the effect of the projected mixed derivatives' matrices in eq. 4.31. As it will be seen, these matrices and the first-order derivatives do not contribute equally to the SDs of the statistical moments, depending on the statistical moment at hand or the QoI. In case their contribution is small, compared to that of the first-order derivatives, they could be omitted. Since a significant portion of the RDO cost with pAPCE comes from their computation, this omission can significantly reduce this cost. Finally, the RDO results are compared with the outcome of an optimization without uncertainties which is performed at $v_\infty = \mu_{v_\infty}$ and $\alpha_\infty = \mu_{\alpha_\infty}$, referred to, from now on, as the "nominal conditions".

The next cases compare pFOSM and pAPCE. For this purpose, the statistical moments, computed with the FOSM and APCE methods, are compared with MC using an adequate number of replicates (which is, thus, expensive), decided based on parametric studies which are omitted. Then, the results of the optimization under uncertainties performed with both pFOSM and pAPCE are presented and the accuracy and computational cost of the two methods are discussed.

The last case demonstrates the use of the projection technique in a problem with $M > 2$. It shows the shape optimization of a duct with many inlet sections. A probability distribution is assigned to the velocity magnitude (practically, the Reynolds number) of each inlet section.

4.3.1 Shape Optimization of an Isolated Airfoil Using pAPCE

The starting shape was the NACA0012 airfoil. Its contour was parameterized using PARSEC-11 (see section 2.9.2). With a fixed trailing edge position, this problem had $N = 10$ design variables. The mean values and standard deviations of α_∞ and v_∞ were $\mu_{\alpha_\infty} = 4^\circ, \sigma_{\alpha_\infty} = 0.6^\circ$ and $\mu_{v_\infty} = 60\ m/s, \sigma_{v_\infty} = 10\ m/s$, respectively. A high

value of σ_{v_∞} was chosen in order to make this case study more challenging since the SD of C_D w.r.t. to α_∞ was much greater than the one w.r.t. v_∞ and UQ would otherwise be dominated by it. A grid of about 43,000 cells, with structured-like quadrilateral elements close to the airfoil and triangular ones elsewhere, was used. Equality constraints were imposed on μ_{C_L}, V and μ_{C_M}. In this case, the constraint targets $\mu^{\oplus}_{C_L}$ and $V^{\oplus}$ were computed on the starting airfoil whereas $\mu^{\oplus}_{C_M} = 0$.

First, the statistical moments of the 3 aerodynamic coefficients were computed using APCE with O_{PCE} in the range of 1 to 5 and L up to 50; results are shown in fig. 4.4. By increasing L, the statistical moments stabilized around a certain value and, as L increased beyond a certain point, only small deviations were observed within a small margin for all O_{PCE}. Beyond $L = 10$, no significant gain was observed. For $L = 10$, the PCE coefficients are shown in Table 4.1. The mean value is equal to α_0 and the remaining coefficients contribute to the standard deviation. There were non-negligible PCE coefficients that emerged by setting $O_{PCE} = 2$. Beyond that point, for higher O_{PCE}, the magnitudes of the newly added coefficients in the PCE decayed by almost one order of magnitude. In this case, values of $O_{PCE} = 2$ and $L = 10$ were decided for the RDO, since no significant gain was observed with higher O_{PCE} or L values.

Table 4.1: RDO Case 1. PCE coefficients computed with $L = 10$. As O_{PCE} increases, more coefficients are involved in the expansion (eq. 4.7) but their values gradually decay. α_0 is equal to the mean value and the standard deviation involves all the rest.

	$O_{PCE}=0$	$O_{PCE}=1$		$O_{PCE}=2$			$O_{PCE}=3$			
PCE Coefficients:	α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9
C_D	0.0133	6.7e-4	-3.4e-4	7.3e-5	-6.5e-6	5.6e-5	1.5e-6	-3.2e-6	1.6e-6	-1.1e-5
C_L	0.4063	0.0601	0.0012	-2.4e-4	2.5e-4	-2.0e-4	-6.0e-5	4.3e-5	-6.3e-5	3.8e-5
C_M	-0.0029	-5.4e-4	1.3e-4	-1.9e-5	3.4e-5	-1.9e-5	-6.7e-6	-2.4e-6	-1.4e-5	1.7e-6

The derivatives of C_D, C_L and C_M w.r.t. b_n and c_m, computed with the adjoint method, are shown in fig. 4.5 and are compared with FDs. The same comparison for the derivatives of the statistical moments, computed with pAPCE, is shown in fig. 4.6. All derivatives in figs. 4.5 and 4.6 are in good agreement with FDs. It was also of interest to assess the effect of the mixed derivatives in eq. 4.31. Their omission did not affect the SDs of the mean values of all coefficients; however, this was not true when J included σ_F. For the purpose of imposing constraints on μ_{C_L} and μ_{C_M} only, the computation of the projected matrix of mixed derivatives, and, thus the solution of the projected-then-differentiated systems was avoided for these QoIs. The same was done for the RDO with $J = \mu_{C_D}$.

First, an optimization run was performed for the minimization of C_D while imposing constraints on C_L and C_M instead of their statistical moments, all computed at the "nominal conditions". Three optimizations under uncertainties

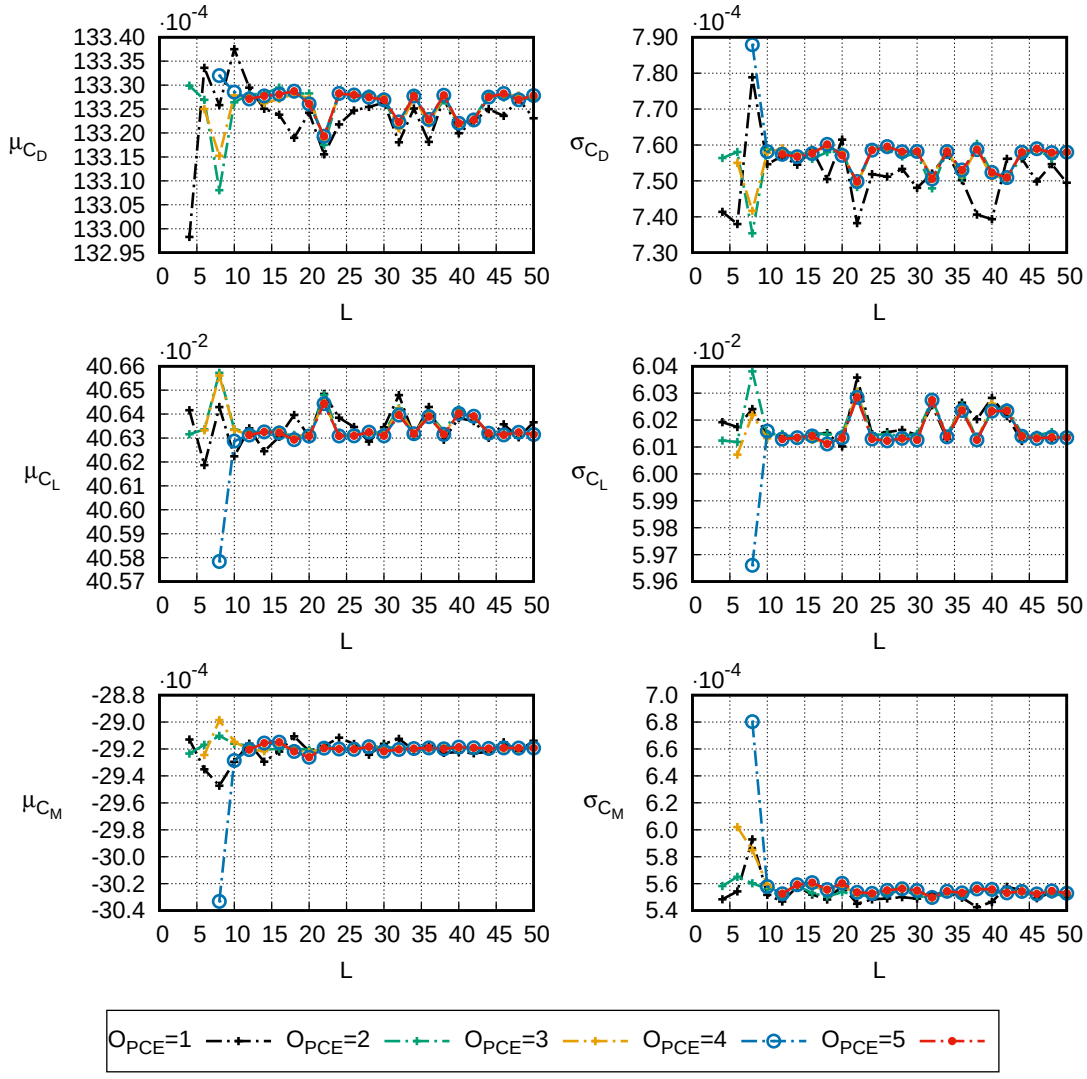


Figure 4.4: RDO Case 1. Statistical moments of C_D (top), C_L (middle) and C_M (bottom) computed using APCE for $L \in [4, 50]$ and $O_{PCE} \in [1, 5]$. **Left**: Mean values. **Right**: Standard deviations.

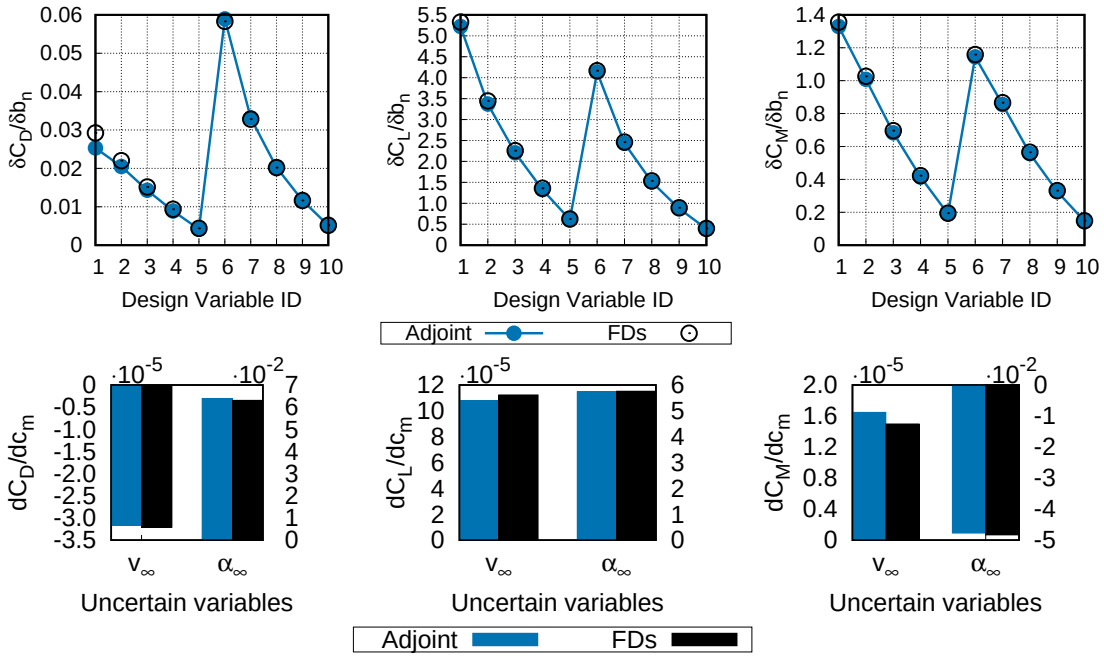


Figure 4.5: RDO Case 1. **Top**: Adjoint SDs w.r.t. b_n. **Bottom**: Adjoint SDs w.r.t. c_m. Derivatives w.r.t. α_∞ are shown on the right y-axis. **Left**: C_D. **Middle**: C_L. **Right**: C_M. Adjoint SDs are compared with reference FDs. In FD computations, each b_n or c_m was perturbed by $\varepsilon = 10^{-6}$.

followed, one with $w = 0$ and the others with $w = 3$ and $w = 6$. The values of all QoIs are shown in fig. 4.7. After each optimization run, MC simulations with 2000 samples were performed to validate the solutions of pAPCE. The optimization at the "nominal conditions" decreased C_D by 0.89% while keeping C_L and V constant and practically setting C_M to zero. Table 4.2 shows the statistical moments of the three aerodynamic coefficients computed for the starting shape and the three optimized airfoils. Per cent reductions/increases in μ_{C_D} and σ_{C_D} are also provided. The MC results re-confirmed that the statistical moments were indeed reduced/increased, albeit by a slightly different factor between each run. All constraints were satisfied within acceptable engineering thresholds. PDFs of all aerodynamic coefficients, computed through MC, are shown in fig. 4.8. The shift of the median of the PDF of C_M around 0 is evident from the plot. The probability of lower C_D was also slightly increased.

Between the two uncertain variables, the QoIs were more sensitive to α_∞. Two plots of C_L and C_M versus C_D, are shown in fig. 4.9, corresponding to the range of α_∞ from 3° to 5°. C_D decreased close to the "nominal conditions" but, as α_∞ increased, the performance deteriorated, especially for the airfoil optimized for $J = \mu_{C_D}$.

The optimized airfoils for $J = C_D$ and $J = \mu_{C_D}$ were very close to each other

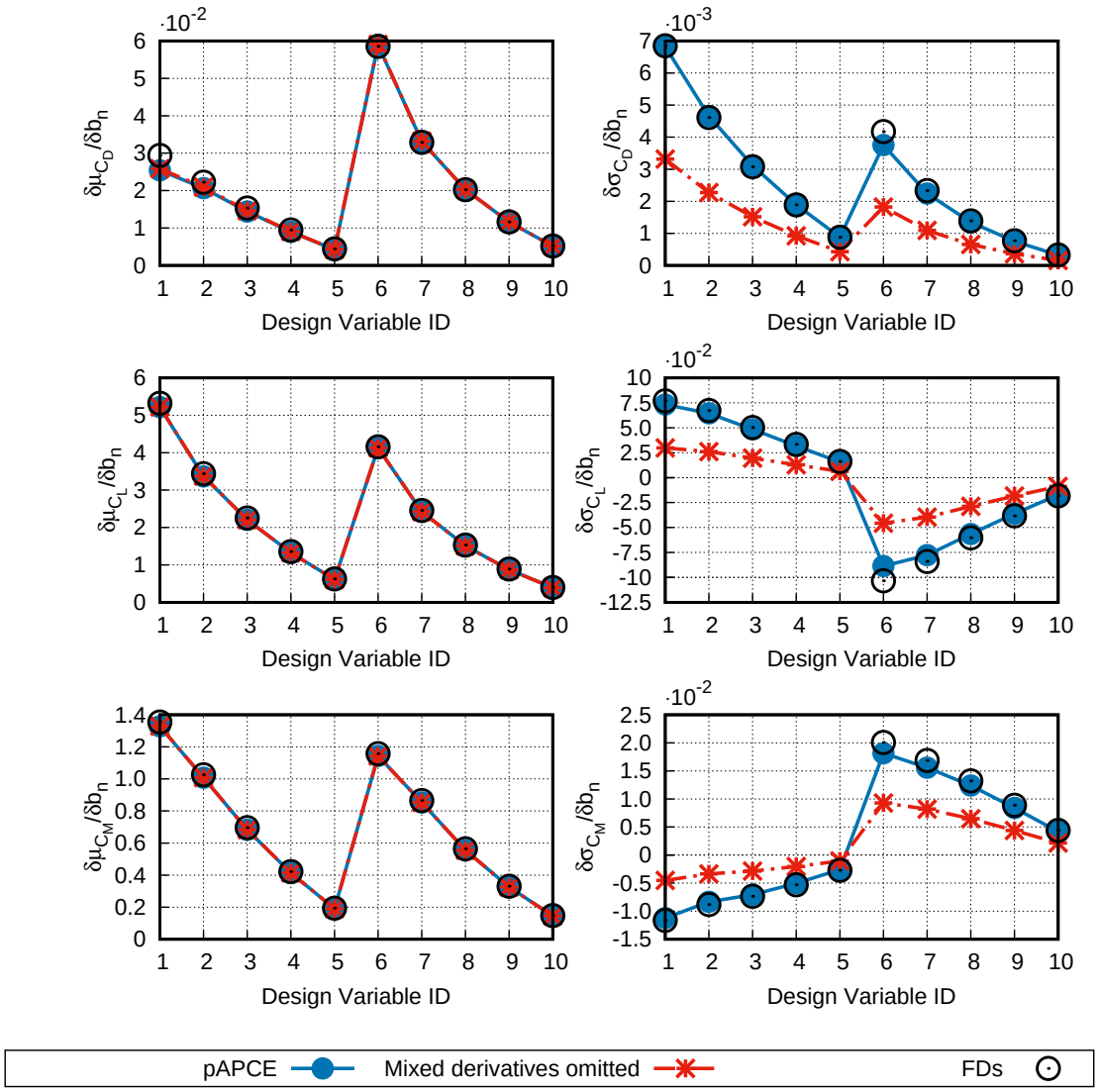


Figure 4.6: RDO Case 1. Derivatives w.r.t. b_n of statistical moments computed with pAPCE for 3 aerodynamic coefficients and verification with reference FDs. In the latter, the FDs perturbed each design variable by $\varepsilon = 10^{-6}$ and solved the UQ problem. **Top**: C_D. **Middle**: C_L. **Bottom**: C_M. **Left**: Mean values. **Right**: Standard deviations.

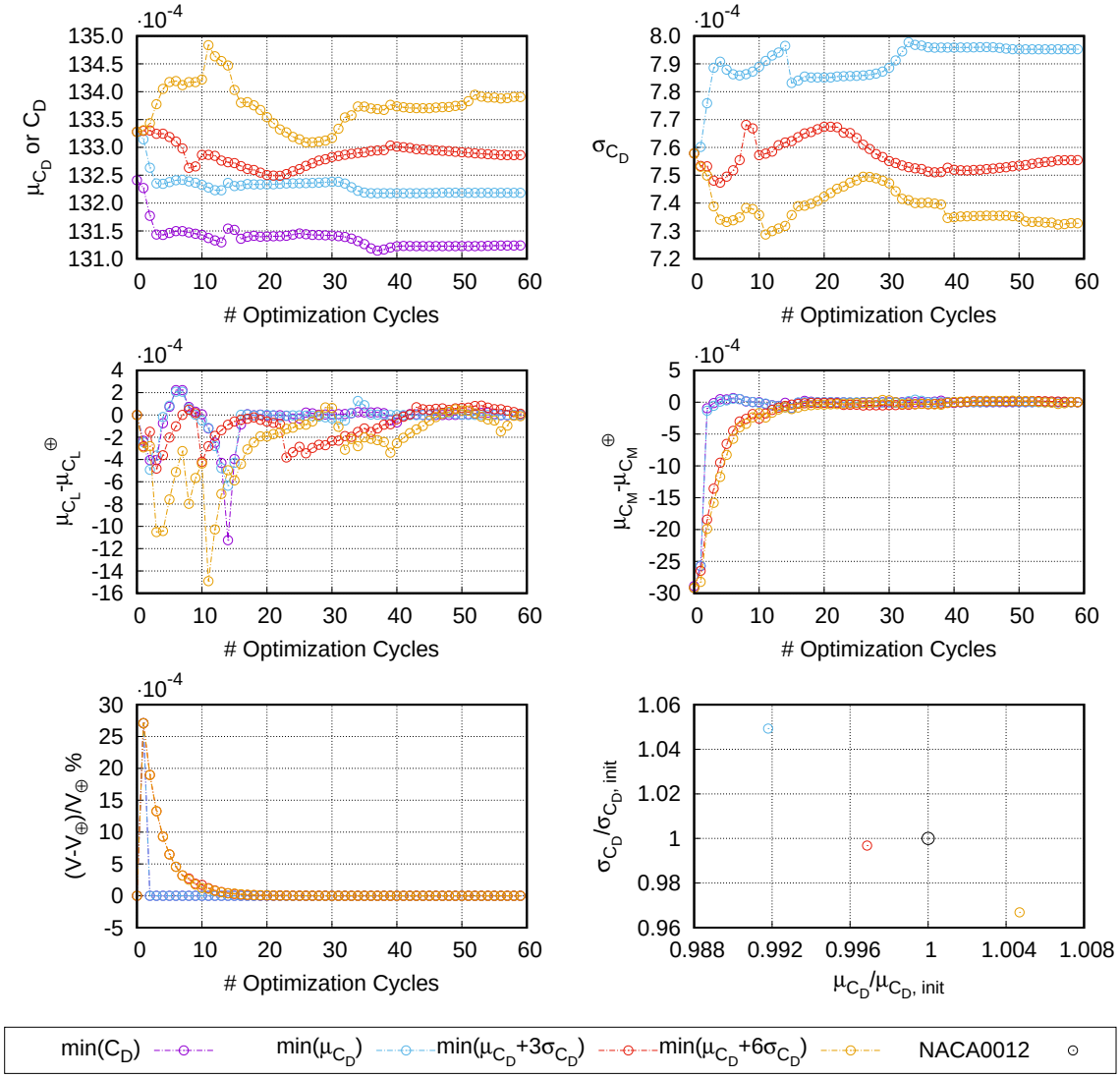


Figure 4.7: RDO Case 1. Evolution of statistical moments of C_D and constraints for airfoils optimized for "min(J)". **Top-Left**: μ_{C_D} or C_D depending on the optimization objective. **Top-Right**: σ_{C_D}. **Middle**: Difference of μ_{C_L} (left) and μ_{C_M} (right) from target. For min. C_D optimization, the values and targets correspond to C_L and C_M. **Bottom-Left**: V constraint. **Bottom-Right**: Pareto front of non-dominated solutions in the μ_{C_D} and σ_{C_D} space (non-dimensional values).

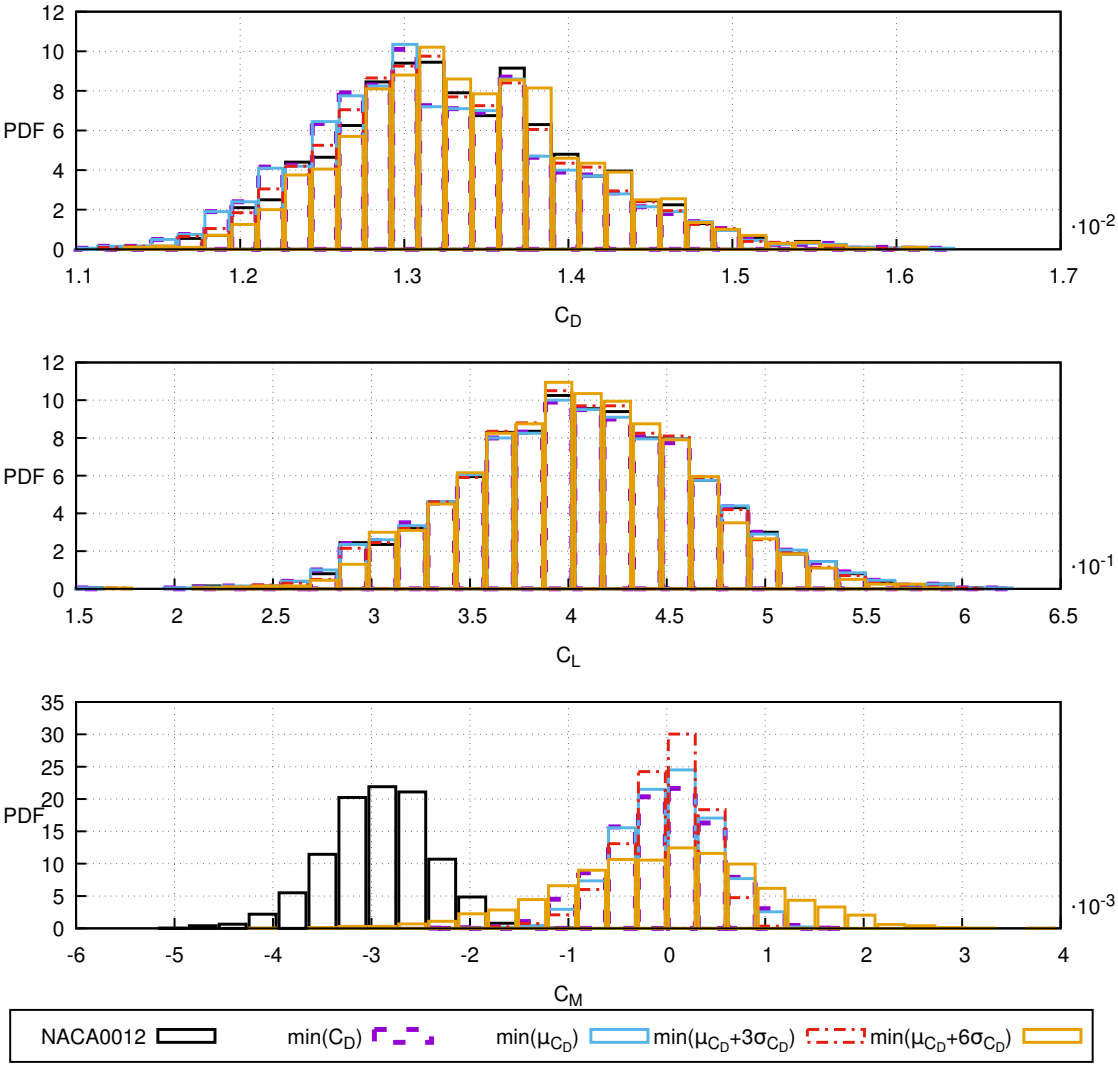


Figure 4.8: RDO Case 1. PDFs of all 3 aerodynamic coefficients computed using MC for the NACA0012 airfoil and the 4 ones optimized for "min(J)". **Top**: C_D. **Middle**: C_L. **Bottom**: C_M.

Table 4.2: RDO Case 1. Statistical moments of C_D, C_L and C_M for the starting and 3 optimized airfoils. The "min(J)" on the left column indicates the objective function for which the three airfoils were optimized. Statistical moments were computed with MC and APCE. The per-cent change in the values of these moments due to the optimization is shown below, measured both w.r.t. to MC and APCE.

	μ_{C_D} values		σ_{C_D} values		μ_{C_L} values		σ_{C_L} values		μ_{C_M} values		σ_{C_M} values	
Geometries evaluated	MC	APCE	MC	APCE	MC	APCE	MC	APCE	MC	APCE	MC	APCE
NACA0012	0.01334	0.01333	7.40e-4	7.58e-4	0.40717	0.40633	0.06063	0.06013	-2.93e-3	-2.91e-3	5.31e-4	5.59e-4
min(μ_{C_D})	0.01323	0.01322	7.74e-4	7.95e-4	0.40719	0.40633	0.06190	0.06137	<|1e-4|	<|1e-4|	4.88e-4	4.34e-4
min($\mu_{C_D} + 3\sigma_{C_D}$)	0.01330	0.01329	7.36e-4	7.55e-4	0.40714	0.40635	0.05851	0.05796	<|1e-4|	<|1e-4|	4.33e-4	4.67e-4
min($\mu_{C_D} + 6\sigma_{C_D}$)	0.01340	0.01339	7.14e-4	7.33e-4	0.40709	0.40632	0.05624	0.05570	<|1e-4|	<|1e-4|	1.03e-3	1.08e-3
	μ_{C_D} % change		σ_{C_D} % change									
min(μ_{C_D})	-0.82	-0.82	4.59	4.88								
min($\mu_{C_D} + 3\sigma_{C_D}$)	-0.29	-0.30	-0.54	-0.39								
min($\mu_{C_D} + 6\sigma_{C_D}$)	0.44	0.45	-3.51	-3.29								

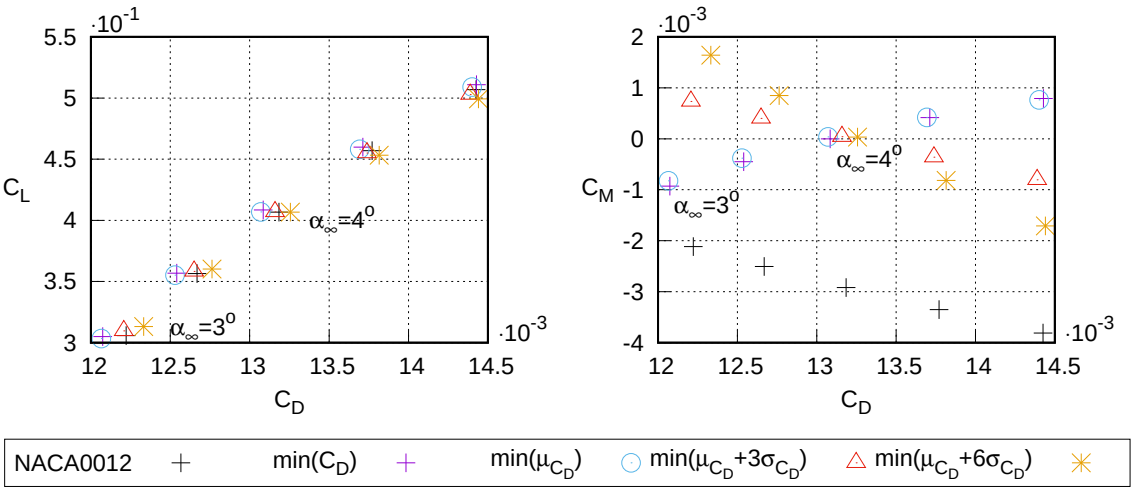


Figure 4.9: RDO Case 1. **Left**: C_L versus C_D. **Right**: C_M versus C_D. The points were computed over the $\alpha_\infty \in [3, 5]^\circ$ range, with a 0.5° step, for the NACA0012 airfoil and the 4 ones optimized for "min(J)".

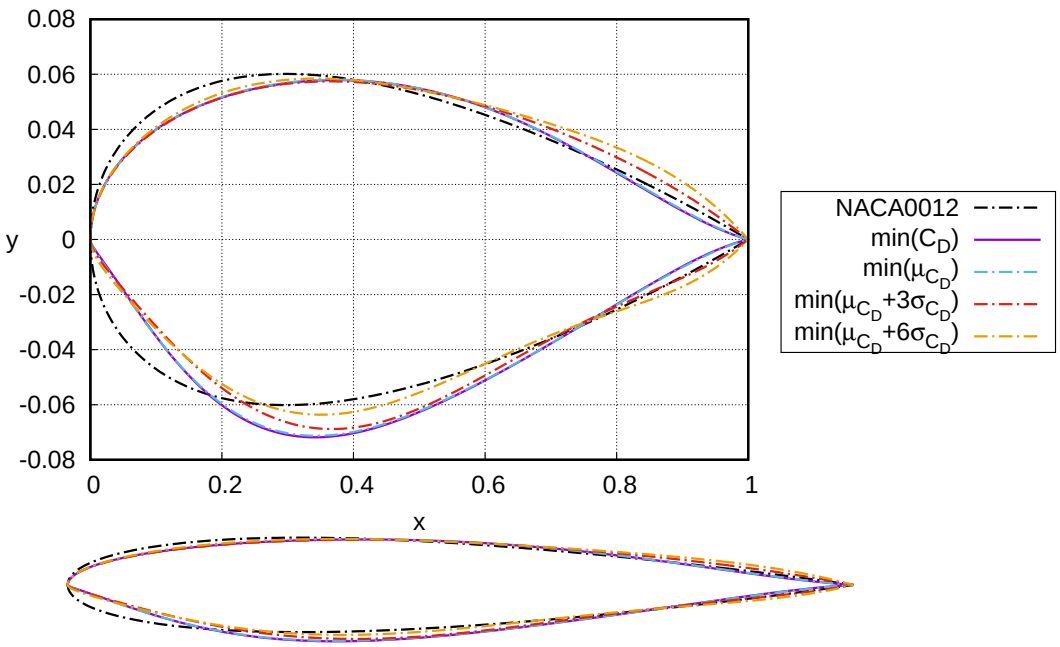


Figure 4.10: RDO Case 1. NACA0012 and optimized airfoil contours with different objective functions, identified through the legend key on the right by "min(J)". **Top**: Airfoils not in scale. **Bottom**: In scale.

(fig. 4.10). This was expected since the derivatives for the two objectives were also very close, figs. 4.5 and 4.6. All airfoils were slightly thinner close to the leading edge and thicker along their middle section. The optimized airfoils for $J = \mu_{C_D} + 3\sigma_{C_D}$ and $J = \mu_{C_D} + 6\sigma_{C_D}$ had a progressively thicker trailing edge wedge. The pressure and skin friction coefficients' plots are shown in fig. 4.11. The coefficients were computed at the "nominal conditions". All airfoils had a milder positive pressure gradient along their suction side but skin friction deviated from one another along the last 20% of the chord. Overall, the optimization prioritized the creation of an airfoil with zero C_M and lower C_D.

Regarding the cost of the constrained optimization with uncertainties, with $O_{PCE} = 2$ and $M = 2$, 10 PCE coefficients need be computed. APCE with $L = 10$ corresponds to an oversampling ratio of $r_{os} = 3$ due to the use of adjoint gradients. To achieve the same ratio, standard regression-based PCE (which excludes gradient information) would have required $L = 30$. With $L = 30$, standard PCE would have required 120 EFS for the RDO (1 primal and 3 adjoint systems of equations at 30 collocation points). With the pAPCE method, the cost is equal to 100 EFS per cycle (solution of the primal and 3 adjoint systems of equations plus 3 DD-then-projected primal and adjoint systems for each one of the 10 collocation points). Without the projection technique, the cost would have been equal to 120 EFS. By omitting the computation of the projected matrix of mixed derivatives of μ_{C_L} and μ_{C_M} for imposing the relevant constraints, the cost

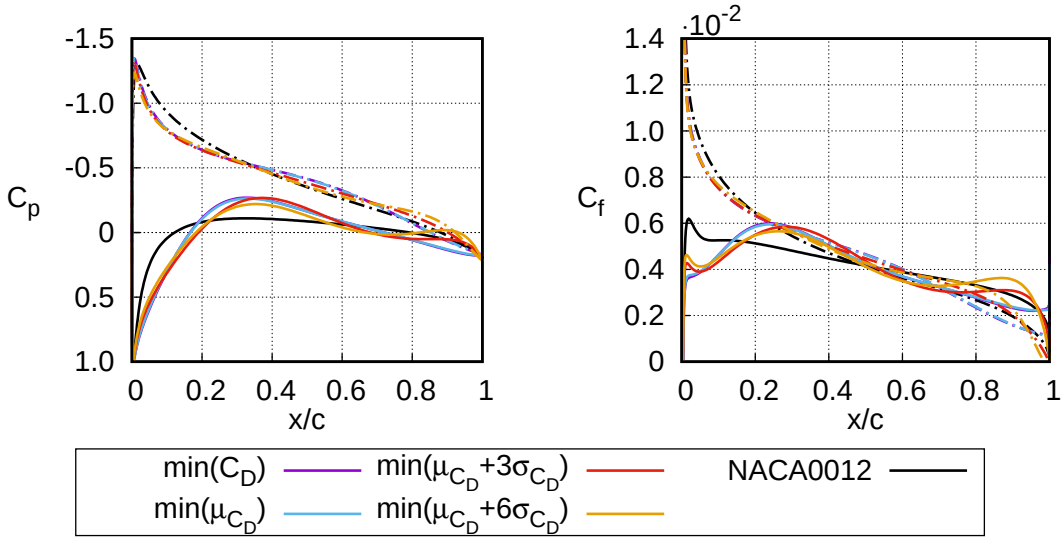


Figure 4.11: RDO Case 1. **Left**: Pressure coefficient. **Right**: Skin-friction coefficient. The coefficients were computed along the two sides of the NACA0012 airfoil and those optimized for "min(J)". Continuous and dashed lines correspond to the pressure and suction sides, respectively.

of the RDO with pAPCE was 60 EFS and, for the case of $J = \mu_{C_D}$, the cost became 40 EFS.

4.3.2 A Comparison of pFOSM and pAPCE

Two airfoil shape optimization problems were solved using both pFOSM and pAPCE. The case published in [30], demonstrating the use of pFOSM in a laminar flow was revisited (with some changes, though) and, then, a turbulent flow case followed. VBS parameterized the airfoil contours and deformed the CFD grids during the optimization. Inequality constraints were imposed on μ_{C_L} and V requiring that they should be greater than or equal to their initial values $\mu_{C_L}^{\oplus}$ and $V^{\oplus}$ (this differs from the equality constraint used in [30]). Additionally, the bounding constraints in eq. 2.34 were applied on the x and y coordinates of the CPs of the VBS morphing boxes. The design variables were updated using the SQP method incorporating an interior-point method strategy [103] for handling inequalities.

4.3.2.1 Shape Optimization in Laminar Flows

The airfoil shape in fig. 4.12, studied also in [30], was optimized by minimizing μ_{C_D} and σ_{C_D}, this time by using and comparing both the pAPCE and pFOSM methods. The chord-based Reynolds number was $Re = 3000$ (calculated using

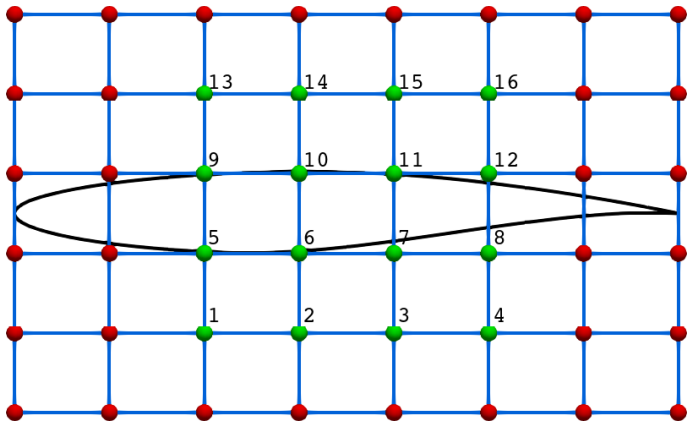


Figure 4.12: RDO Case 2. Airfoil optimized using pFOSM and pAPCE and VBS morphing box. Red/green CPs remained frozen/active throughout the optimization. The $x-y$ coordinates of the internal (green) CPs act as the $N = 16 \times 2 = 32$ design variables.

μ_{v_∞}) and the flow was simulated as a laminar one using a hybrid grid of about 54000 quadrilateral (structured-like layers around the airfoil contour) and triangular elements (elsewhere). As for the uncertain variables, their mean values were $\mu_{v_\infty} = 1\ m/s$, $\mu_{\alpha_\infty} = 4^\circ$ and their standard deviations were $\sigma_{v_\infty} = 0.15\ m/s$, $\sigma_{\alpha_\infty} = 0.5^\circ$, both following normal distributions. Mesh and Reynolds number were the same with [30]. Differences existed in the imposed constraints on μ_{C_L} and V and, also, the displacements of the CPs were constrained which was not the case in [30]. Over and above, the statistical moments of the uncertain variables took on different values and the VBS morphing box was different. Due to all these differences, the pFOSM solution was re-computed and its results, included in this thesis, were different than those of [30].

UQ results computed with FOSM, APCE for $O_{PCE} \in [1,5]$ and $L \in [4:20]$ and MC are presented in fig. 4.13. With $O_{PCE} \geq 2$, the error in both μ_{C_D} and σ_{C_D}, based on the results of MC, seems to be sufficiently low for most of the values of L. Therefore, it was decided to use $O_{PCE} = 2$ and $L = 7$ for the optimization, since no significant gain in accuracy was observed for higher O_{PCE} or L values. APCE computed statistical moments (the standard deviation, in particular) more accurately than FOSM. According to [82], accuracy issues in FOSM (and, thus, pFOSM) are expected when the ratio of standard deviation to the mean value of the uncertain variables, a.k.a. Coefficient of Variation (CV), is high. In the examined case, $CV_{v_\infty} = 15\%$ and $CV_{\alpha_\infty} = 12.5\%$. By switching to a case with higher CV values, namely $CV_{v_\infty} = 30\%$ and $CV_{\alpha_\infty} = 40\%$, and re-running the UQ, fig. 4.13 (bottom), both FOSM and APCE became less accurate; the drop in accuracy was more visible in FOSM. With the exception of σ_{C_D} computations with higher CVs, the APCE results were close to or within the confidence interval bounds computed

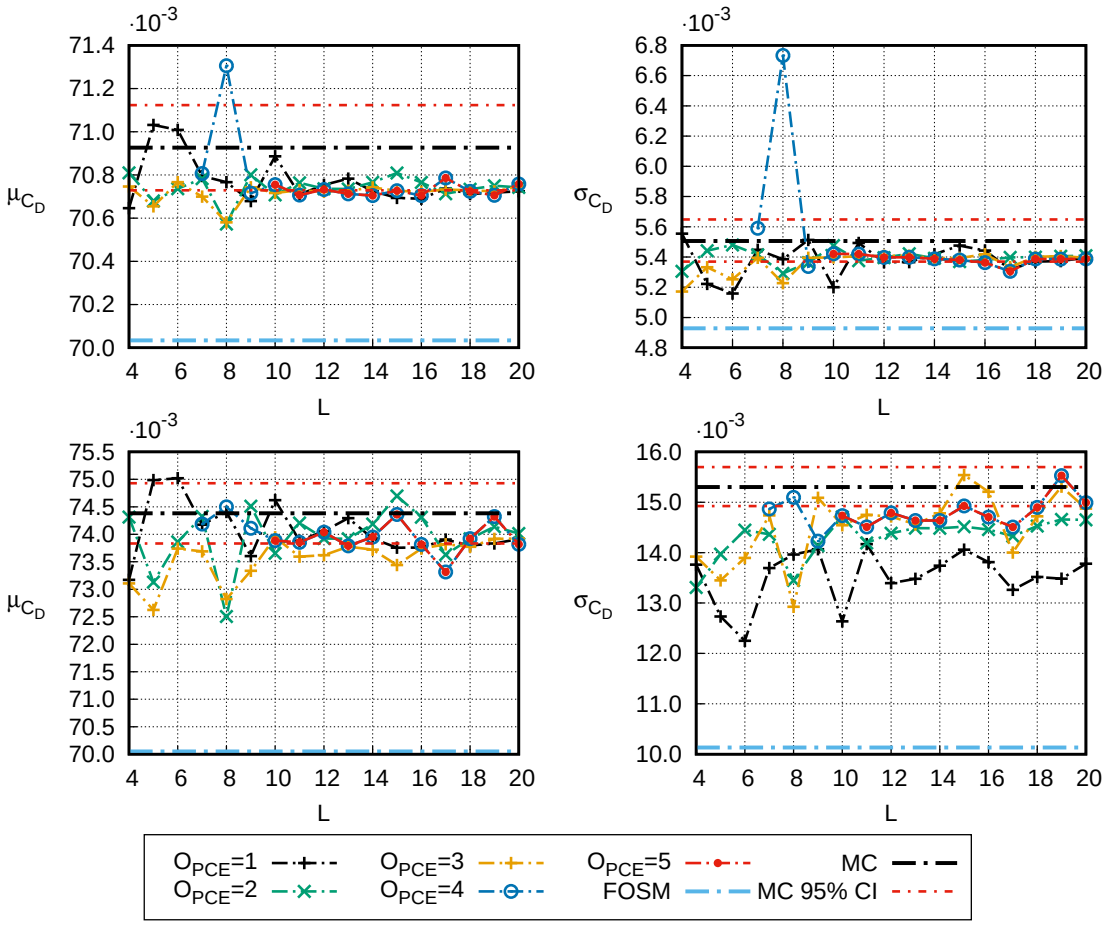


Figure 4.13: RDO Case 2. μ_{C_D} and σ_{C_D} computed using APCE for various values of L (in the range from 4 to 20) and O_{PCE} (from 1 to 5) and FOSM. **Top**: With $CV_{v_\infty} = 15\%$, $CV_{\alpha_\infty} = 12.5\%$. **Bottom**: $CV_{v_\infty} = 30\%$, $CV_{\alpha_\infty} = 40\%$. **Left**: Mean values. **Right**: Standard deviations. MC simulations were performed using 3000 replicates. The lower and upper 95% confidence interval bounds, for both μ_{C_D} and σ_{C_D} (marked as CI) computed with MC are included.

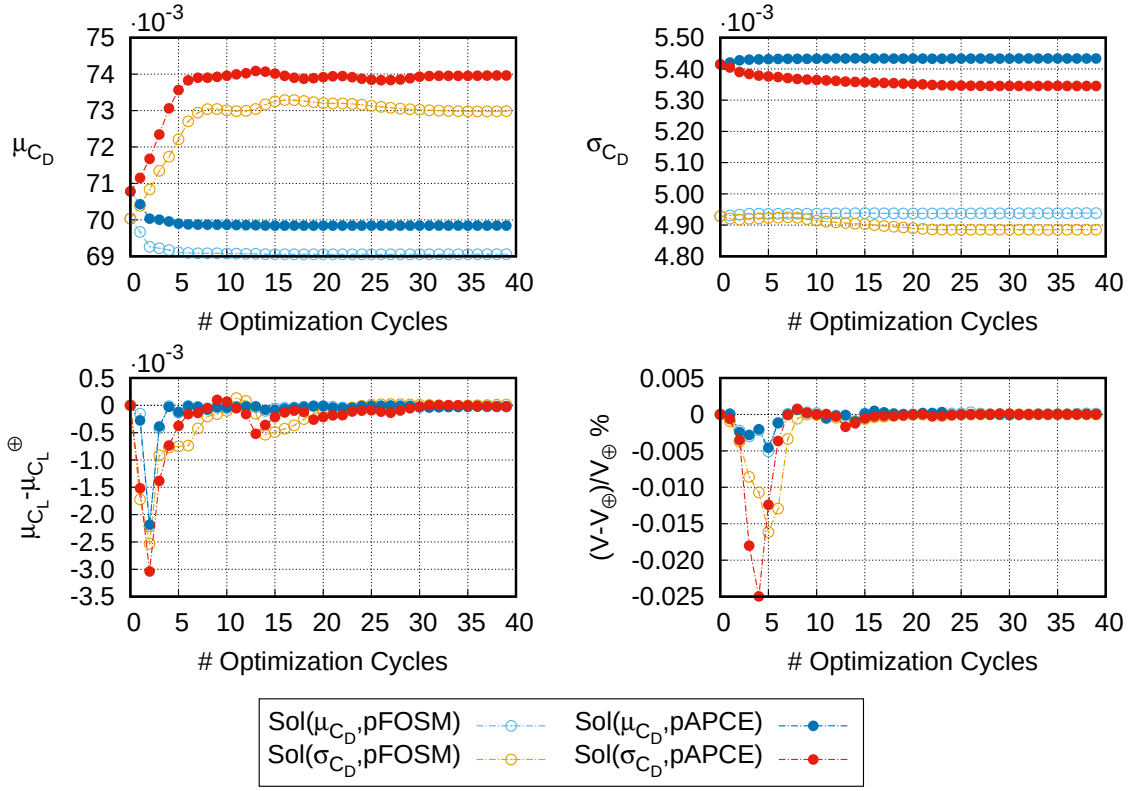


Figure 4.14: RDO Case 2. Convergence of objective functions and constraints in the RDO with pFOSM and pAPCE (O_{PCE}=2, $L=7$) for three different J. **Top-Left**: μ_{C_D}. **Top-Right**: σ_{C_D} **Bottom-Left**: μ_{C_L} constraint. **Bottom-Right**: V constraint. "Sol(J,method)" stands for the solution/airfoil of an optimization targeting min. J, performed using "method".

by the MC, in contrast to the FOSM results which laid beyond the interval bounds. Compared to FOSM, a SOSM MoM could provide more accurate results; however, a gradient-based optimization relying on SOSM requires the computation of third-order derivatives of the QoI which greatly increases the complexity of the method.

Two optimization runs under uncertainties were performed with $CV_{v_\infty} = 15\%$ and $CV_{\alpha_\infty} = 12.5\%$ using both pFOSM and pAPCE ($O_{PCE}=2$, $L=7$). The first run aimed at minimizing only μ_{C_D} and the second only σ_{C_D}. Although minimizing only σ_{C_D} may have little practical benefits, runs with $J=\mu_{C_D}$ or $J=\sigma_{C_D}$ help to draw useful conclusions. Convergence plots of the runs are shown in fig. 4.14, the optimized airfoils are illustrated in fig. 4.15, whereas fig. 4.16 shows the pressure and skin friction coefficients for the initial and optimized airfoils.

With $J = \mu_{C_D}$, the curvature of the airfoil suction side became smaller and a decrease in C_f was observed along that side, fig. 4.16. At the same time, pressure dropped more along the suction side. With $J = \sigma_{C_D}$, σ_{C_D} was slightly reduced but μ_{C_D} increased instead. In order for σ_{C_D} to be reduced, the airfoil thickness

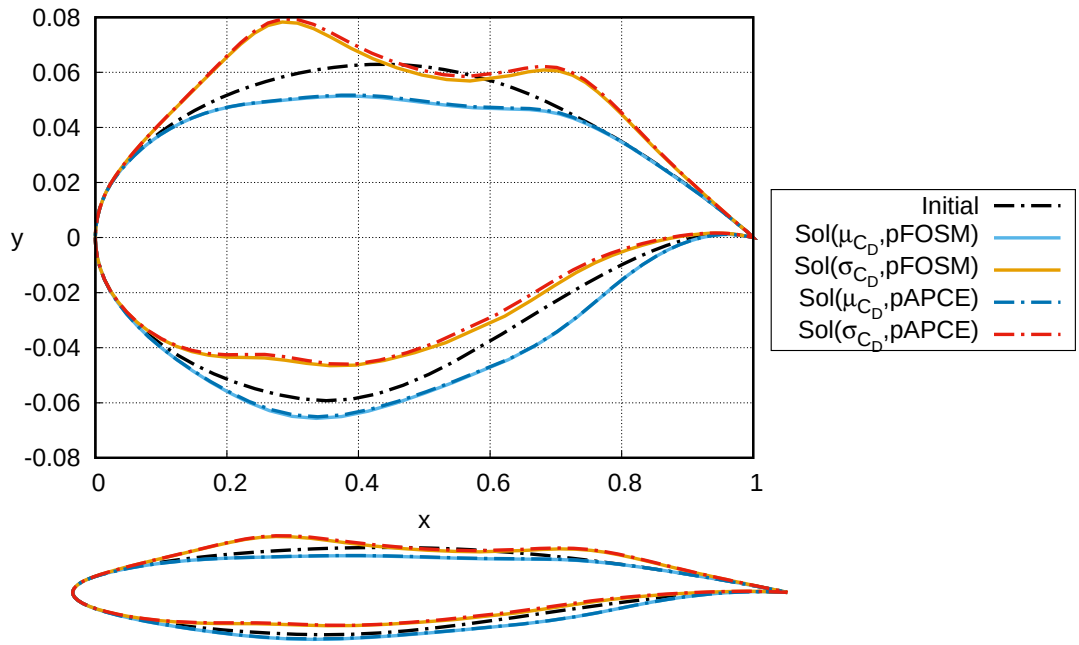


Figure 4.15: RDO Case 2. Initial and optimized airfoils for $J = \mu_{C_D}$ and $J = \sigma_{C_D}$ using the pFOSM and pAPCE (O_{PCE}=2, $L = 7$) methods. **Top**: Airfoils not in scale. **Bottom**: In scale.

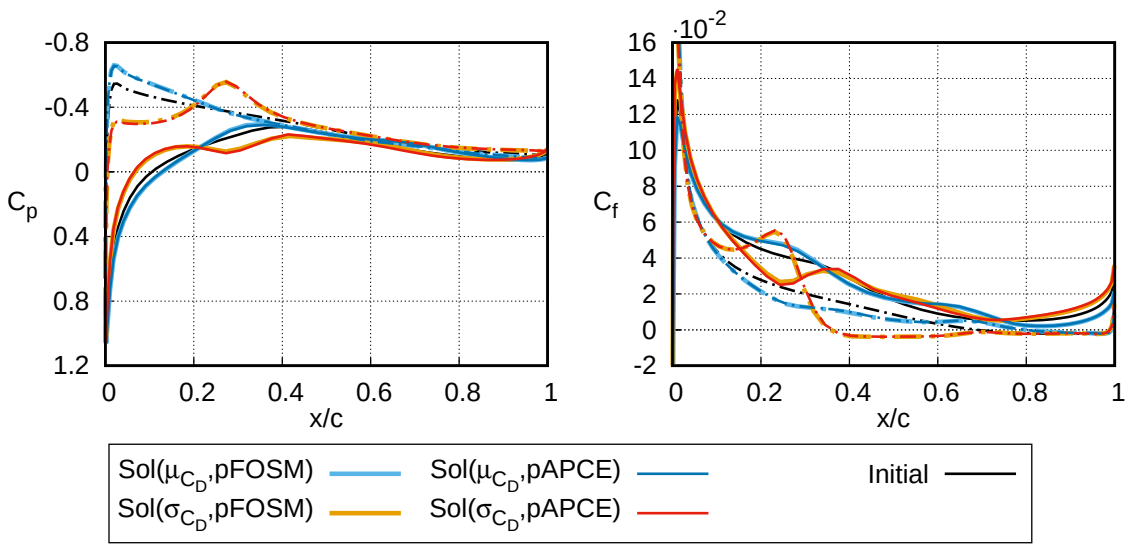


Figure 4.16: RDO Case 2. **Left**: C_p. **Right**: C_f. The coefficients were computed for the initial and optimized with pFOSM and pAPCE airfoils, for two different J. Continuous/dashed lines correspond to the pressure/suction sides of the airfoils, respectively. Curves for the airfoils optimized with both methods and for each J practically coincide.

increased retarding the formation or minimizing the effect of the adverse pressure gradient on the suction side, fig. 4.16. The increased thickness and camber also made the airfoil less sensitive to changes in α_∞. The increase in μ_{C_D} might be due to the flow separation occurring close to 40% of the chord on the optimized airfoil, fig. 4.16.

Re-evaluating the statistical moments of C_D of all optimized shapes using MC, Table 4.3, re-confirmed the reduction in the objective functions' values. Using the moments computed by MC as reference values, APCE was consistently more accurate than FOSM which underestimated σ_{C_D} both for the initial and optimized geometries. Additionally, the per-cent changes estimated by APCE were slightly closer to MC than FOSM. Nevertheless, the shapes and, thus, the performances of the airfoils optimized using pFOSM and pAPCE were very close to each other. Figure 4.17 shows the PDF of C_D and C_L, computed by MC, on the initial and all optimized airfoils.

Finally, fig. 4.18 shows the relation between C_D and C_L and both uncertain variables around their nominal values and within the $[-2\sigma : 2\sigma]$ range for the initial shape (figures corresponding to the optimized shapes show similar trends and are omitted). Linear fitting was performed for all curves. Although the coefficients of determination, in fig. 4.18, show good correlation for the linear model, especially for C_L and α_∞, it can visually be deduced that the curves do not exhibit a perfectly linear behavior. This apparently small deviation from linearity could perhaps explain the differences in accuracy observed between FOSM, which uses a linear approximation for the QoI and retains only first-order terms, and APCE with $O_{PCE}=2$.

For the cost of the optimization using the pFOSM method, this was equal to 5 EFS per cycle; for the more accurate pAPCE (O_{PCE}=2, L=7), this went up to 49 EFS. The cost of the latter includes the solution of the adjoint system computing the gradient of C_L, as one of the constraints involved μ_{C_L}, and the solution of the projected DD primal and adjoint systems that computed the derivatives of μ_{C_L} w.r.t. b_n. To compute the derivatives of μ_{C_L} w.r.t. b_n, in pFOSM, only one additional adjoint system must be solved (4+1=5 EFS). If $J=\mu_{C_D}$, the pFOSM optimization is essentially the same as an optimization at the mean flow conditions that does not account for uncertainties, see eq. 4.18, and its cost per optimization cycle is only 3 EFS (one primal and two adjoint systems' solutions). However, with the pAPCE method, the derivatives of either μ_{C_D} or μ_{C_L} w.r.t. b_n require $z_m \frac{\delta^2 F}{\delta c_m \delta b_n}$ and the projected DD primal and adjoint systems still need to be solved. This makes the pFOSM method faster than pAPCE. Nevertheless, the additional computational cost of the latter compensates for the increased UQ accuracy, based on the results shown in fig. 4.13, Table 4.3 and the discussion above. On the other hand, the re-evaluations with the pAPCE and MC methods of all the airfoils optimized using the pFOSM in Table 4.3 indicate that, despite its lower accuracy, the quality of the outcome of the optimization is re-confirmed by UQ methods of higher fidelity.

Table 4.3: RDO Case 2. Statistical moments of C_D for the initial and 6 optimized airfoils. All moments were re-evaluated using the MC (3000 replicates), FOSM and APCE, irrespective of which of them was used in the optimization. In the second half of the table, the corresponding error, computed using the MC as reference, and value changes w.r.t. the initial airfoil are shown. Figures in bold denote that optimization and re-evaluation used the exact same method.

	μ_{C_D} values			σ_{C_D} values		
Geometries evaluated	MC	FOSM	APCE	MC	FOSM	APCE
Initial	0.07093	0.07003	0.07078	5.506e-3	4.929e-3	5.415e-3
Sol(μ_{C_D},pFOSM)	0.06997	**0.06906**	0.06983	5.531e-3	4.939e-3	5.434e-3
Sol(μ_{C_D},pAPCE)	0.06998	0.06907	**0.06984**	5.531e-3	4.938e-3	5.433e-3
Sol(σ_{C_D},pFOSM)	0.07388	0.07298	0.07373	5.395e-3	**4.886e-3**	5.341e-3
Sol(σ_{C_D},pAPCE)	0.07412	0.07322	0.07396	5.396e-3	4.892e-3	**5.346e-3**
	μ_{C_D} % errors (w.r.t. MC)			σ_{C_D} % errors (w.r.t. MC)		
Initial		1.26	0.21		10.47	1.65
Sol(μ_{C_D},pFOSM)		**1.30**	0.20		10.71	1.76
Sol(μ_{C_D},pAPCE)		1.30	**0.20**		10.72	1.76
Sol(σ_{C_D},pFOSM)		1.21	0.20		**9.43**	0.98
Sol(σ_{C_D},pAPCE)		1.21	0.21		9.34	**0.93**
	μ_{C_D} % change			σ_{C_D} % change		
Sol(μ_{C_D},pFOSM)	-1.35	**-1.38**	-1.34	0.45	0.20	0.35
Sol(μ_{C_D},pAPCE)	-1.33	-1.37	**-1.32**	0.45	0.18	0.33
Sol(σ_{C_D},pFOSM)	4.15	4.21	4.16	-2.01	**-0.87**	-1.36
Sol(σ_{C_D},pAPCE)	4.49	4.55	4.49	-1.99	-0.75	**-1.27**
	μ_{C_L} values			σ_{C_L} values		
Initial	0.15825	0.15788	0.15758	0.02568	0.02396	0.02461
Sol(μ_{C_D},pFOSM)	0.15792	**0.15786**	0.15724	0.02468	0.02332	0.02363
Sol(μ_{C_D},pAPCE)	0.15824	0.15816	**0.15756**	0.02481	0.02343	0.02376
Sol(σ_{C_D},pFOSM)	0.15647	0.15790	0.15595	0.02064	**0.01975**	0.02075
Sol(σ_{C_D},pAPCE)	0.15807	0.15943	0.15756	0.02080	0.01993	**0.02090**

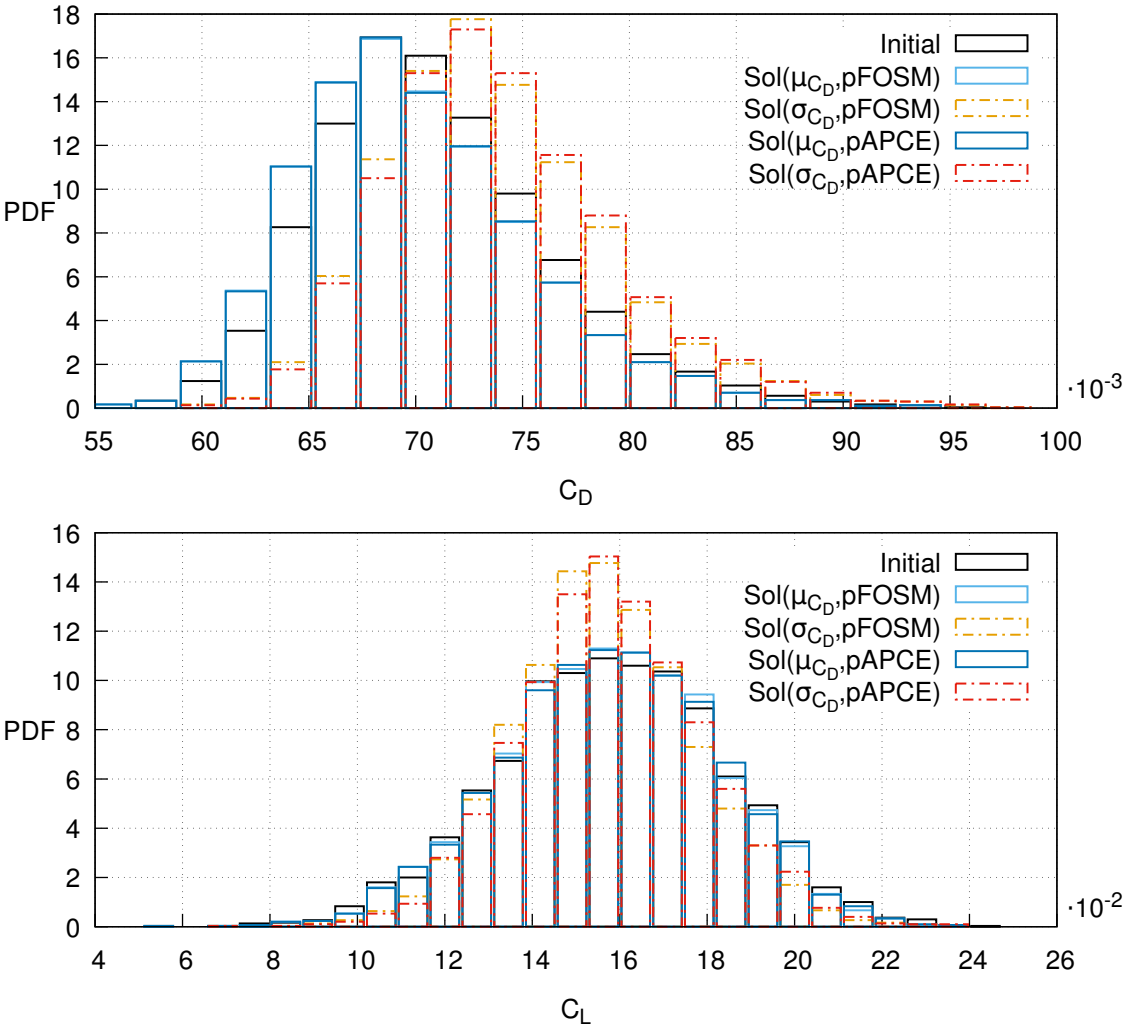


Figure 4.17: RDO Case 2. PDFs of C_D and C_L computed by MC. **Top**: C_D **Bottom**: C_L. The distributions were computed on the initial airfoil and the ones optimized with pFOSM and pAPCE for $J = \mu_{C_D}$ and $J = \sigma_{C_D}$, respectively, using 3000 replicates.

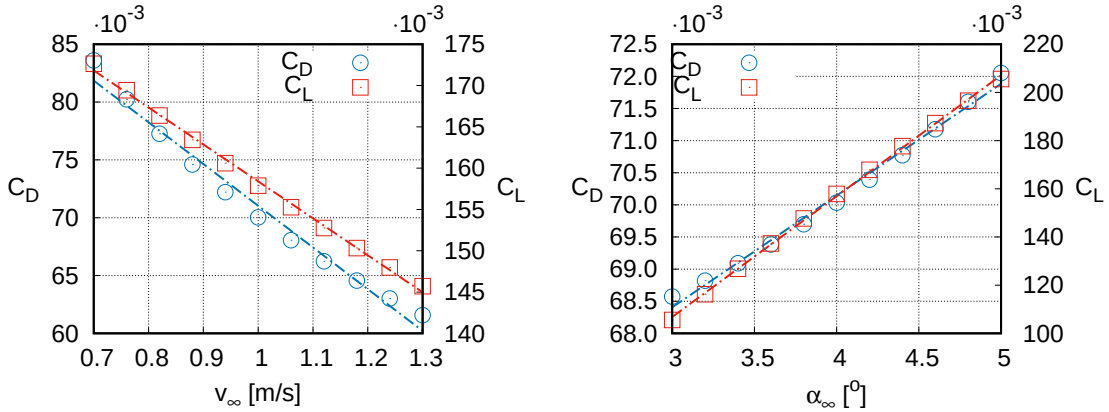


Figure 4.18: RDO Case 2. Variations of C_D and C_L around the nominal values of the uncertain variables for the initial shape computed within the $[-2\sigma : 2\sigma]$ interval. **Left**: v_∞. **Right**: α_∞. Linear regression is included; the coefficient of determination R is greater than 0.982 for all lines.

4.3.2.2 Shape Optimization of the Fauvel 14% Airfoil

The extension of pFOSM and pAPCE to turbulent flows is demonstrated in the shape optimization of the isolated Fauvel 14% thickness airfoil [151] in a turbulent flow with $Re{=}10^6$, based on μ_{v_∞}. The grid consisted of approximately 48000 quadrilateral and triangular cells. The airfoil contour and the VBS morphing box are shown in fig. 4.19. The mean values of the uncertain variables were $\mu_{v_\infty} = 15\ m/s$ and $\mu_{\alpha_\infty} = 8^\circ$ and their standard deviations were $\sigma_{v_\infty} = 0.75\ m/s$ and $\sigma_{\alpha_\infty} = 0.8^\circ$, respectively. Following an analysis similar to that of section 4.3.1, it was chosen that $O_{PCE}{=}2$ and $L{=}8$.

The SDs of the statistical moments of C_D and C_L are shown in figs. 4.20 and 4.21, where comparisons are made between adjoint SDs and FDs. These are the derivatives of eq. 4.1 and are computed by setting $J = \mu_{C_D}$ and $J = \sigma_{C_D}$. For each perturbation of the CPs in FD, μ_{C_D} and σ_{C_D} were computed by solving the UQ problem either with the FOSM or APCE method (eq. 4.13). The omission of the mixed derivatives from the gradient led to similar considerations as in

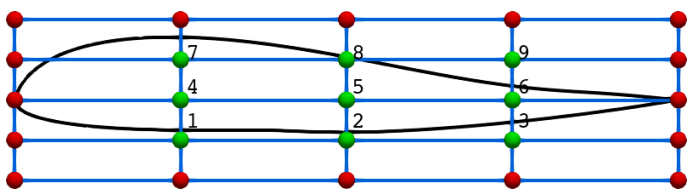


Figure 4.19: RDO Case 3. Fauvel 14% airfoil contour and 5×5 VBS morphing box. Red/green CPs remained frozen/active throughout the optimization. The $x - y$ coordinates of the internal (green) CPs act as the $N{=}9 \times 2{=}18$ design variables.

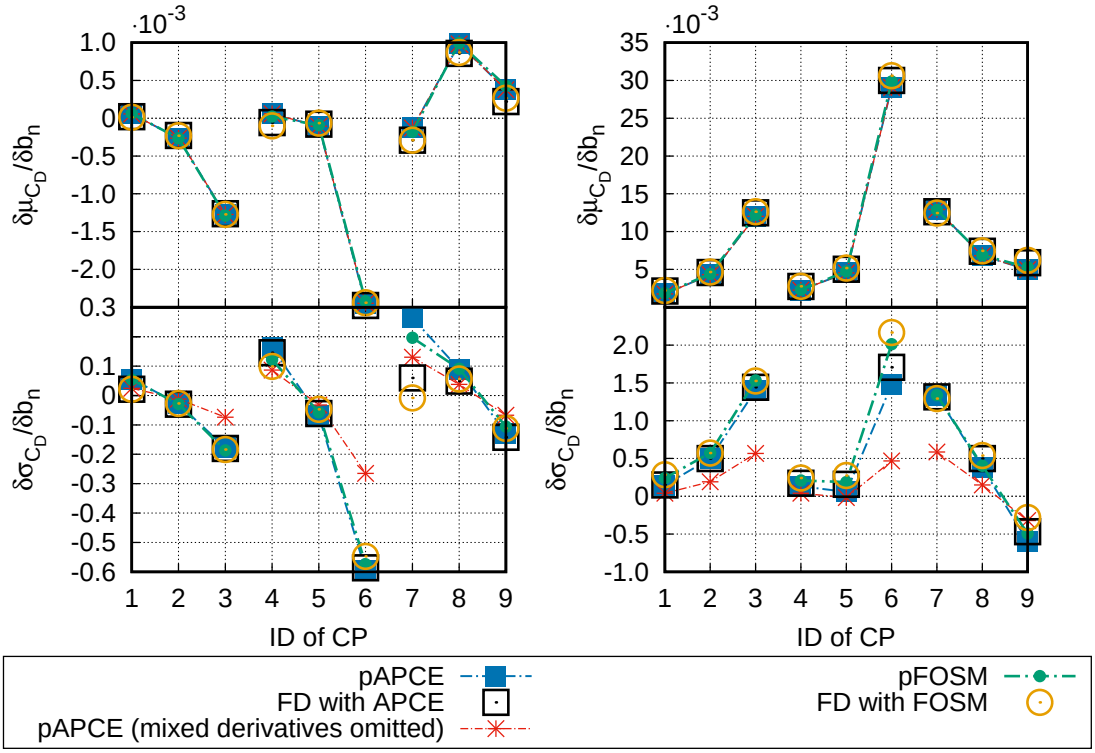


Figure 4.20: RDO Case 3. Derivatives of statistical moments of C_D w.r.t. b_n i.e. the x- and y-coordinates of the free-to-move CPs (numbered in fig. 4.19). **Left**: For the x coordinate. **Right**: For the y coordinate. **Top**: μ_{C_D} **Bottom**: σ_{C_D}. Plots on the left and right correspond to x and y components of CPs, respectively. A step value equal to $10^{-6}m$ was used in FD computations.

section 4.3.1 for pAPCE, and alleviated the cost of computing the gradients of μ_{C_L} and μ_{C_D}.

Three shape optimization runs under uncertainties were performed, with min. μ_{C_D}, $\mu_{C_D} + 2\sigma_{C_D}$ and $\mu_{C_D} + 6\sigma_{C_D}$ as the target, in each case. Shape optimization results can be seen in fig. 4.22. Additionally, UQ for C_D was performed using APCE ($O_{PCE} = 2$, $L = 8$), FOSM and MC, for the initial and all optimized shapes. The results of UQ, along with the reductions achieved through the optimization, are shown in Table 4.4. MC re-confirmed that the objective functions were indeed reduced in all cases and, once again, the computations were accurate enough for μ_{C_D} with both methods; however, it was re-confirmed that APCE is more accurate in computing σ_{C_D}. For min. $\mu_{C_D} + 6\sigma_{C_D}$, the optimization with pAPCE achieved a slightly greater reduction in both μ_{C_D} and σ_{C_D} than pFOSM. This greater reduction could also be seen when the result was re-evaluated with all methods shown in Table 4.4. This was not observed in the case of the other objectives, however, as seen in fig. 4.22. This might be due to the optimization not having fully "converged", in contrast to the

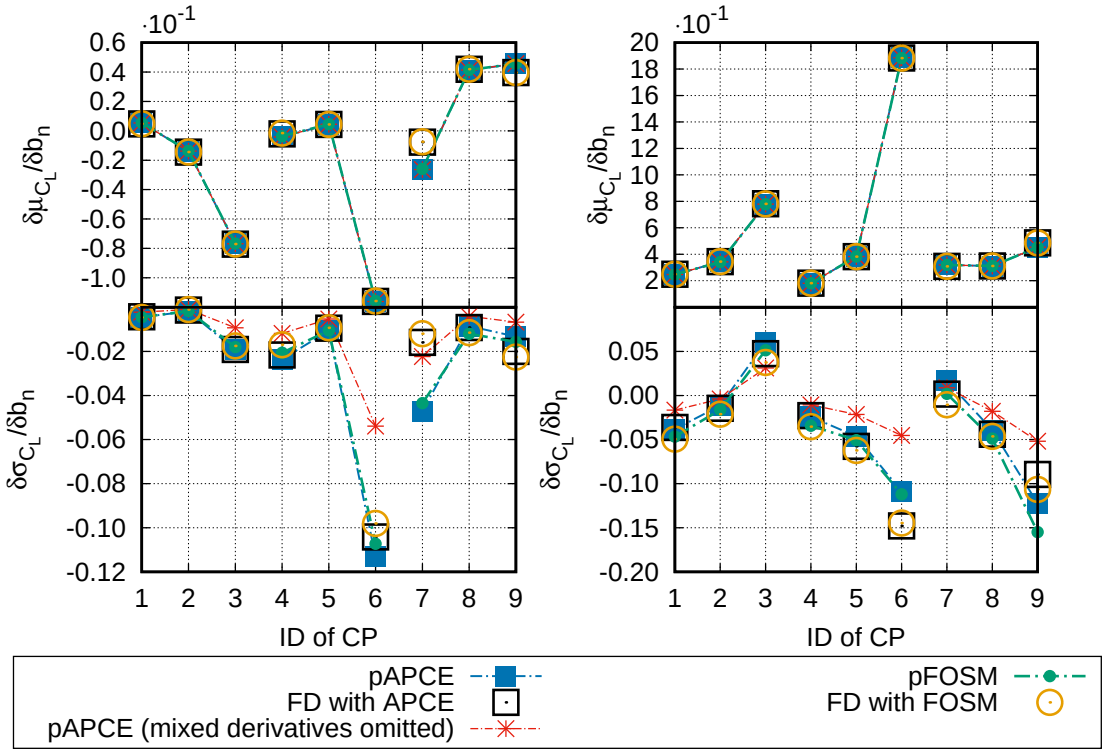


Figure 4.21: RDO Case 3. Derivatives statistical moments of C_L w.r.t. b_n and FDs computed using $\varepsilon = 10^{-6}$ m. **Left**: SDs w.r.t. the x coordinate. **Right**: SDs w.r.t. the y coordinate. **Top**: μ_{C_L}. **Bottom**: σ_{C_L}.

run for min. $\mu_{C_D} + 6\sigma_{C_D}$ which required approximately 160 optimization cycles to complete. Figure 4.23 shows the initial and optimized shapes. In the cases with $J = \mu_{C_D} + 6\sigma_{C_D}$, the two methods led to visibly different optimized shapes. Figure 4.24 shows the pressure and skin friction coefficients of the initial and optimized airfoils. The decrease in C_f after $x/c \approx 0.65$, observed on the initial airfoil, was not that great over the optimized airfoils, the only exception being the one optimized for $J = \mu_{C_D} + 6\sigma_{C_D}$ over which the flow separated after $x/c \approx 0.8$.

In fig. 4.25, the PDFs of both aerodynamic coefficients are shown. The simultaneous reductions in μ_{C_D} and σ_{C_D} were visible in all cases except for the one with $J = \mu_{C_D} + 6\sigma_{C_D}$ where μ_{C_D} was increased. With $J = \mu_{C_D}$, the likelihood of lower C_D values increased, compared to the initial PDF, and the right tail of the distribution was severed, thus reducing σ_{C_D}. The latter also occured with $J = \mu_{C_D} + 6\sigma_{C_D}$ but the "centroid" of the PDF was shifted towards higher values.

Some conclusions can be drawn from the work presented in this section. The pFOSM method requires less EFS for shape optimization than pAPCE. On the other hand, pAPCE is able to compute statistical moments more accurately. To gain the most out of pAPCE, a study should be performed to determine adequate values for L and O_{PCE} and ensure a good response from the PCE. The computation

Table 4.4: RDO Case 3. Statistical moments of C_D for the initial and 4 optimized airfoils. All moments are re-evaluated using MC (2000 replicates), FOSM and APCE, irrespective of which of them was used in the optimization. The corresponding error, computed via comparisons with MC (reference), and value changes w.r.t. the initial airfoil are shown below them. Figures in bold denote that the same method is used in both the optimization and the re-evaluation.

	μ_{C_D}			σ_{C_D}		
Geometries	MC	FOSM	APCE	MC	FOSM	APCE
Initial	0.02197	0.02185	0.02183	1.937e-3	2.059e-3	1.985e-3
Sol(μ_{C_D},pFOSM)	0.02161	**0.02152**	0.02148	1.746e-3	1.884e-3	1.804e-3
Sol(μ_{C_D},pAPCE)	0.02164	0.02155	**0.02151**	1.746e-3	1.884e-3	1.804e-3
Sol($\mu_{C_D} + 2\sigma_{C_D}$,pFOSM)	0.02182	**0.02173**	0.02169	1.713e-3	**1.848e-3**	1.769e-3
Sol($\mu_{C_D} + 2\sigma_{C_D}$,pAPCE)	0.02189	0.02180	**0.02175**	1.722e-3	1.858e-3	**1.778e-3**
Sol($\mu_{C_D} + 6\sigma_{C_D}$,pFOSM)	0.02285	**0.02280**	0.02272	1.377e-3	**1.496e-3**	1.444e-3
Sol($\mu_{C_D} + 6\sigma_{C_D}$,pAPCE)	0.02308	0.02304	**0.02295**	1.327e-3	1.455e-3	**1.395e-3**
	μ_{C_D} % errors (w.r.t. MC)			σ_{C_D} % errors (w.r.t. MC)		
Initial		0.54	0.63		6.29	2.47
Sol(μ_{C_D},pFOSM)		**0.41**	0.60		7.90	3.32
Sol(μ_{C_D},pAPCE)		0.41	**0.60**		7.90	3.32
Sol($\mu_{C_D} + 2\sigma_{C_D}$,pFOSM)		**0.41**	0.59		**7.88**	3.26
Sol($\mu_{C_D} + 2\sigma_{C_D}$,pAPCE)		0.41	**0.63**		7.89	**3.25**
Sol($\mu_{C_D} + 6\sigma_{C_D}$,pFOSM)		**0.21**	0.56		**8.64**	4.86
Sol($\mu_{C_D} + 6\sigma_{C_D}$,pAPCE)		0.17	**0.56**		9.64	**5.12**
	μ_{C_D} % change			σ_{C_D} % change		
Sol(μ_{C_D},pFOSM)	-1.63	**-1.51**	-1.60	-9.86	-8.49	-9.11
Sol(μ_{C_D},pAPCE)	-1.50	-1.37	**-1.46**	-9.86	-8.49	-9.11
Sol($\mu_{C_D} + 2\sigma_{C_D}$,pFOSM)	-0.68	**-0.54**	-0.64	-11.56	**-10.24**	-10.88
Sol($\mu_{C_D} + 2\sigma_{C_D}$,pAPCE)	-0.36	-0.22	**-0.36**	-11.09	-9.76	**-10.42**
Sol($\mu_{C_D} + 6\sigma_{C_D}$,pFOSM)	4.00	**4.34**	4.07	-28.91	**-27.34**	-27.25
Sol($\mu_{C_D} + 6\sigma_{C_D}$,pAPCE)	5.05	5.44	**5.13**	-31.49	-29.33	**-29.72**

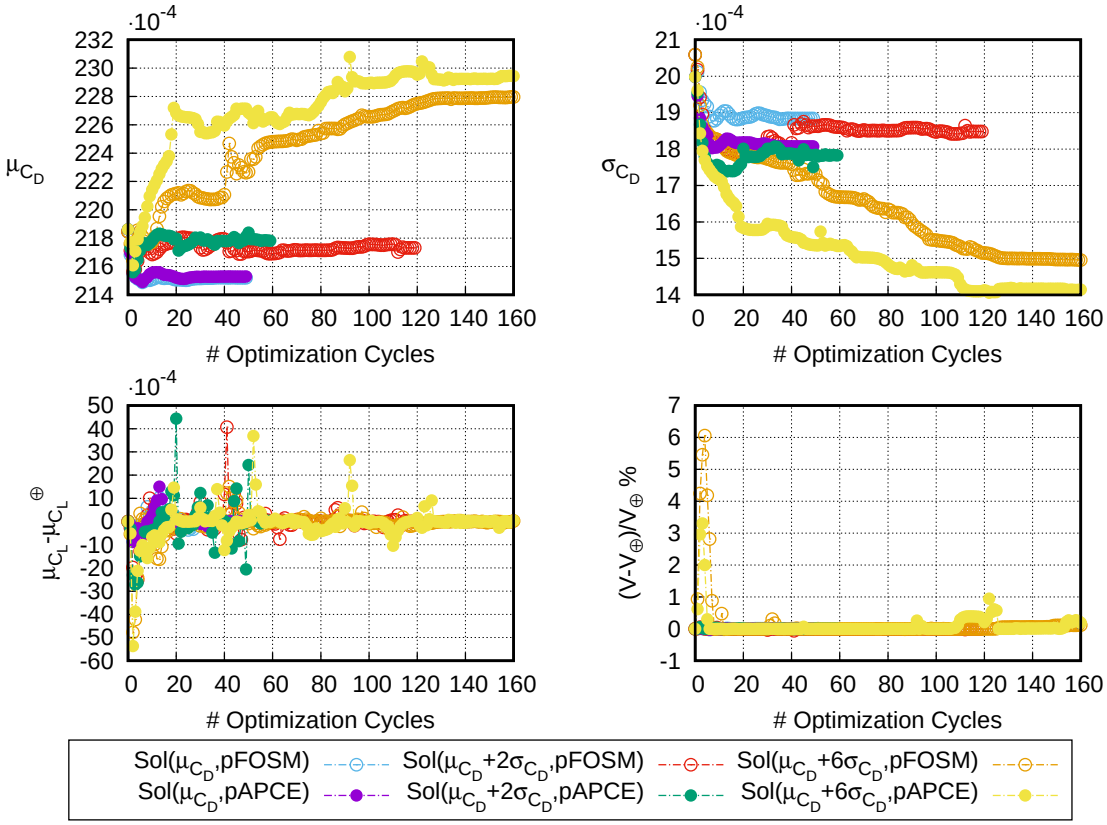


Figure 4.22: RDO Case 3. Convergence of the optimization loops performed with pFOSM and pAPCE ($O_{PCE}=2$ and $L=8$). **Top-Left**: μ_{C_D}. **Top-Right**: σ_{C_D}. **Bottom-Left**: μ_{C_L} constraint. **Bottom-Right**: V constraints. Both inequality constraints were finally met.

of unnecessary PCE coefficients can be avoided by using an accurate enough PCE of lower order. The use of approximate gradients with pAPCE also reduces the CPU cost and this is especially beneficial in case the QoIs are more than the uncertain variables. For the aerodynamic optimization cases dealt with in this chapter, the results of pFOSM and pAPCE were very close to each other which was most likely due to the prevalence of lower order terms in the FOSM and PCE approximations. As seen from the PDF plots, the changes in the airfoil shapes had a noticeable effect on the PDF of the aerodynamic coefficients. The cost reduction gained by the use of projections was up to 33%, depending on the case, even with two uncertain variables.

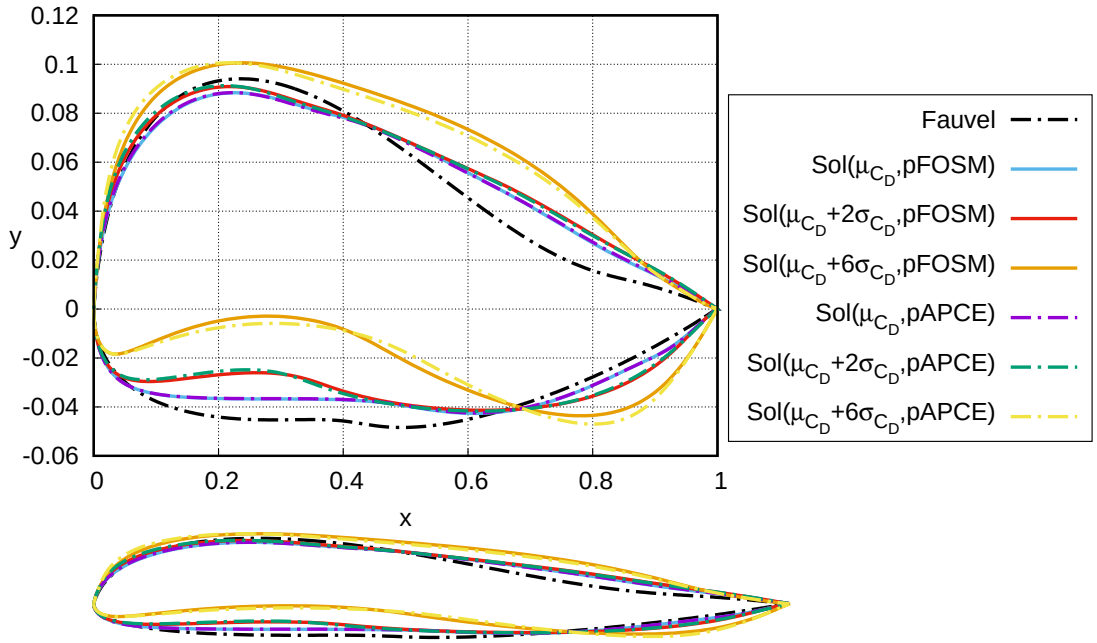


Figure 4.23: RDO Case 3. Initial airfoil contours and optimized ones using pFOSM and pAPCE (O_{PCE}=4 and $L=8$), for $J=\mu_{C_D}$, $J=\mu_{C_D}+2\sigma_{C_D}$ and $J=\mu_{C_D}+6\sigma_{C_D}$. **Top**: Airfoils not in scale. **Bottom**: In scale.

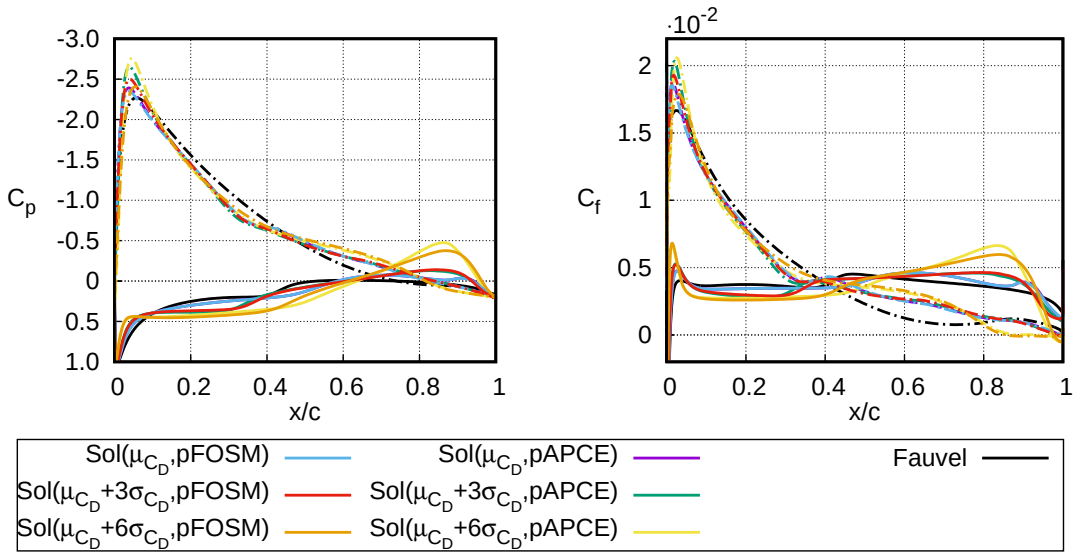


Figure 4.24: RDO Case 3. **Left**: C_p. **Right**: C_f. The two coefficients were computed for the initial and optimized with pFOSM and pAPCE airfoils, for all optimization targets. Continuous/dashed lines correspond to the pressure/suction sides, respectively.

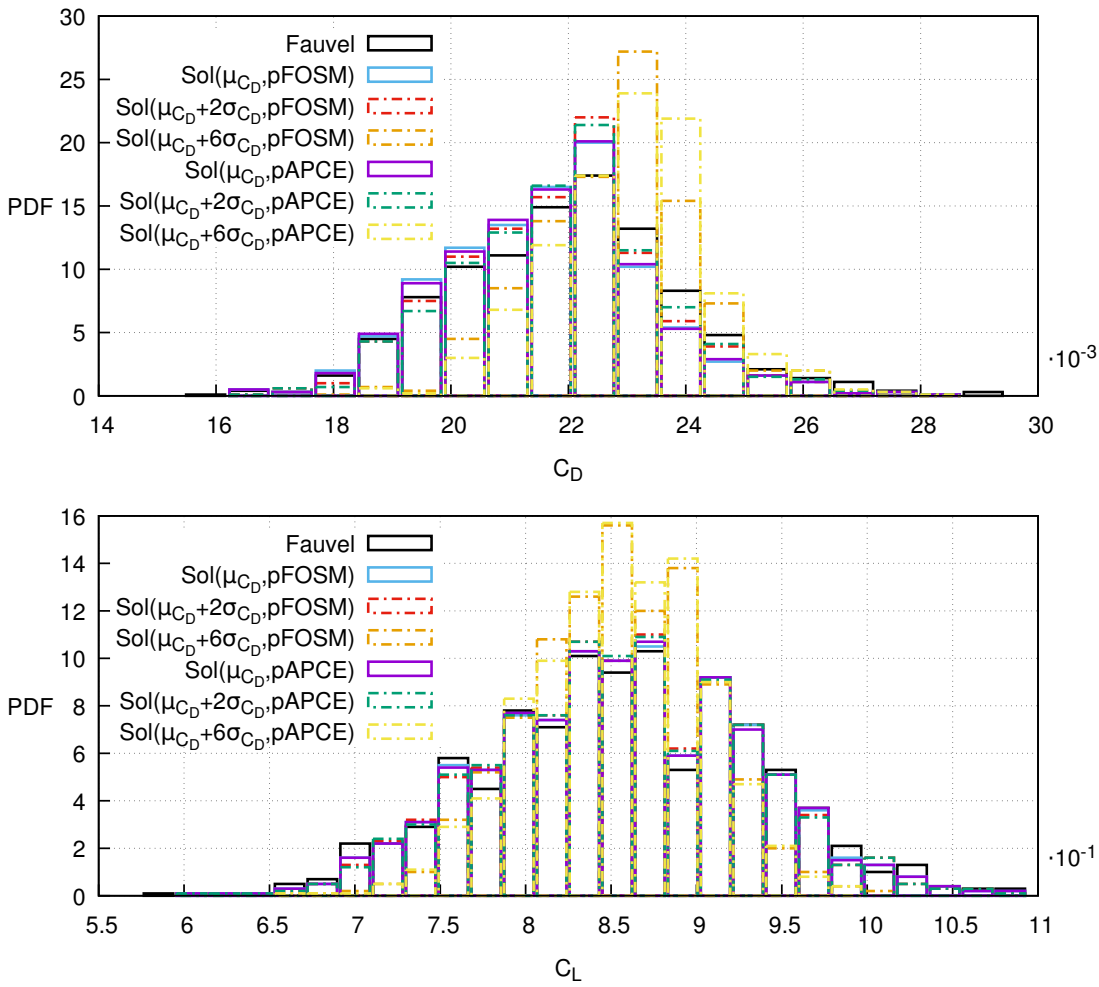


Figure 4.25: RDO Case 3. PDFs computed with the MC using 2000 replicates. **Top**:C_D. **Bottom**: C_L. The distributions were computed for the initial airfoil and all optimized airfoils.

4.3.3 Scalability of the Projection Technique

In this section, a shape optimization problem with 5 uncertain variables was set up and solved with pAPCE. The shape of a network comprising ducted sections and a middle-section compartment, shown in fig. 4.26, were optimized for min. total pressure losses, P_t, between the five inlets leading to the middle compartment and the outlet. The walls of some of the ducts and the middle compartment were parameterized using NURBS curves, see fig. 4.26. The uncertainty pertained to the flow velocity magnitude at the five inlets. The Reynolds numbers for all inlet sections correspond to laminar flows. At S_{I_1} and S_{I_2}, $\mu_{v_I} = 5 \cdot 10^{-3} m/s$ and $\sigma_{v_I} = 10^{-3} m/s$. At S_{I_3}, $\mu_{v_I} = 4 \cdot 10^{-3} m/s$ and $\sigma_{v_I} = 5 \cdot 10^{-4} m/s$. At S_{I_4} and S_{I_5}, $\mu_{v_I} = 4 \cdot 10^{-3} m/s$ and $\sigma_{v_I} = 10^{-3} m/s$.

The optimization was performed using pAPCE for min. $J = \mu_{P_t} + 6\sigma_{P_t}$. It was chosen that $O_{PCE} = 2$ and $L = 6$. SDs w.r.t. the coordinates of NURBS CPs are shown in fig. 4.27. The reduction in the statistical moments is shown in fig. 4.28. Both moments were reduced by 33% to 36%. The optimized shapes are shown in fig. 4.29. Ducted sections were inflated causing flow deceleration (not shown) in the optimized shape. The total pressure difference between the outlet and inlets,

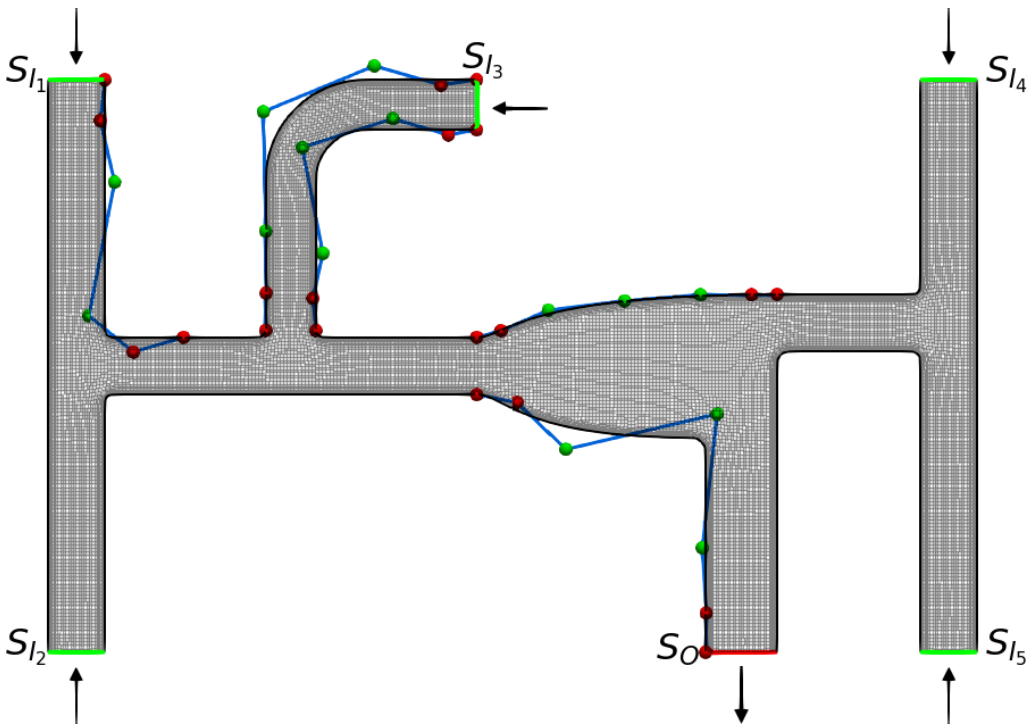


Figure 4.26: RDO Case 4. A network of ducts with five inlet sections and one outlet. Green/red patches correspond to the inlets/outlet. A few of the walls of the ducts and the compartment were parameterized with NURBS curves. Green CPs of the NURBS were allowed to be displaced during the optimization, red ones remained fixed.

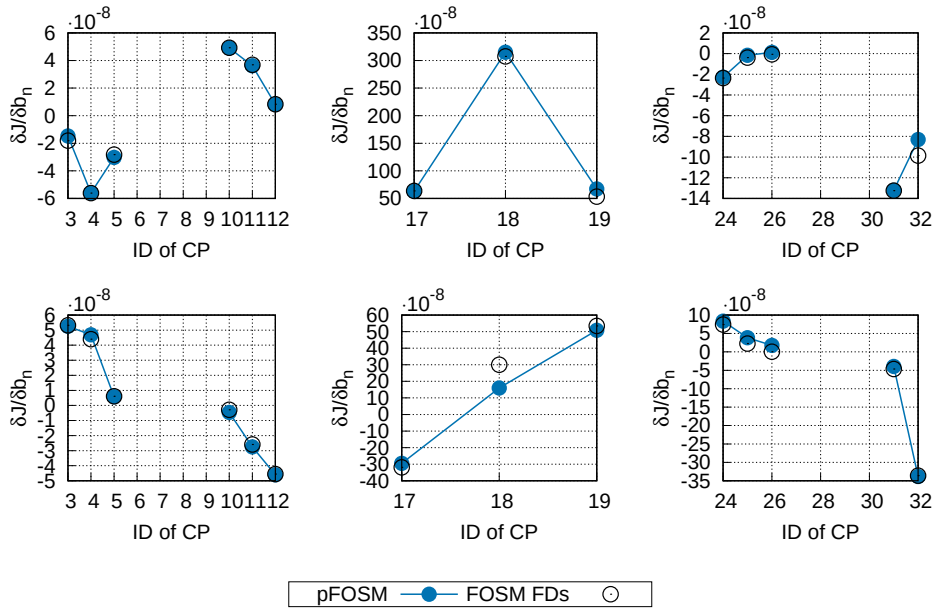


Figure 4.27: RDO Case 4. SDs w.r.t. the x and y coordinates of the NURBS CPs for $J = \mu_{P_t} + 6\sigma_{P_t}$. **Top**: x coordinate. **Bottom**: y coordinate. Each row corresponds to a different curve. Only active CPs are shown.

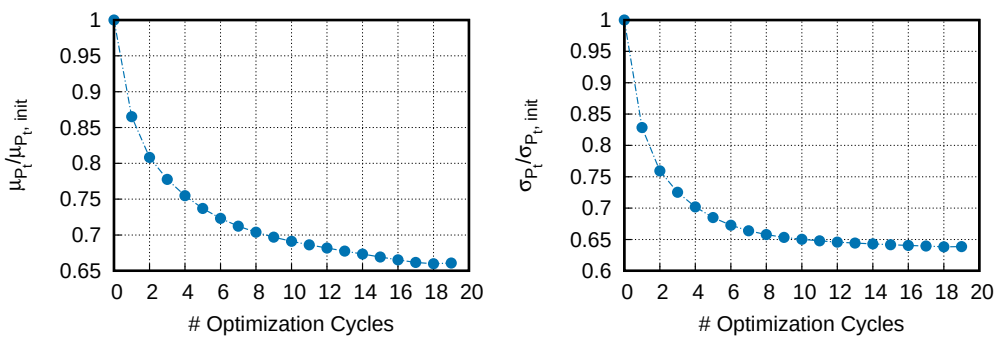


Figure 4.28: RDO Case 4. Convergence of the optimization loop with pAPCE. **Left**: μ_{P_t}. **Right**: σ_{P_t}. All values are normalized w.r.t. their initial ones.

computed at the mean values of the uncertain variables, was visibly decreased.

With 5 uncertain variables and $O_{PCE} = 2$, there exist 21 PCE coefficients to be computed. For shape optimization with pAPCE, using $L = 6$, only 24 EFS per cycle were needed. Without the projection technique, the cost per cycle would have been equal to 72 EFS. The profit from using adjoint gradients in UQ with APCE, compared to standard PCE, is shown in fig. 4.30. Statistical moments computed with MC were used as the reference values. At least 21 EFS would have to be spared by standard PCE to accurately compute both statistical moments due to the existence of the 21 PCE coefficients. With APCE, these moments were

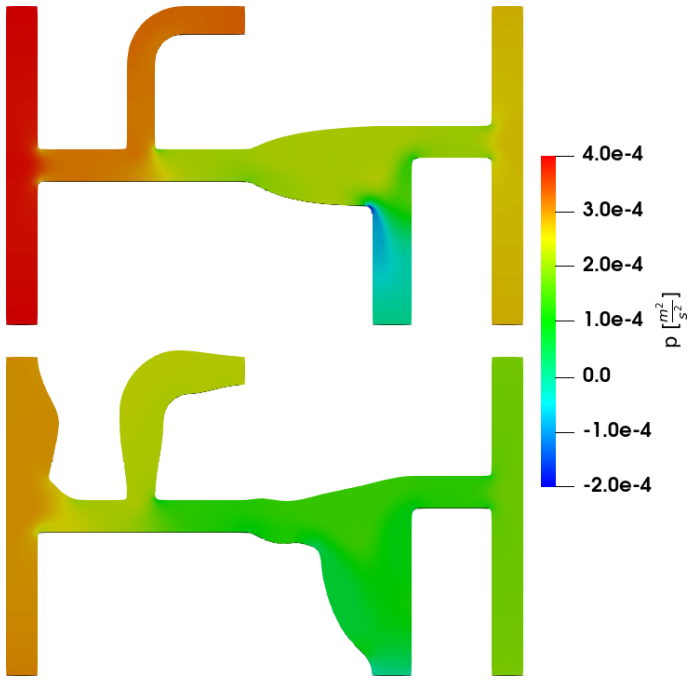


Figure 4.29: RDO Case 4. **Top**: Initial shape. **Bottom**: Optimized shape for $J = \mu_{P_t} + 6\sigma_{P_t}$. The total pressure field, computed at the mean values of the uncertain variables, is shown for both shapes.

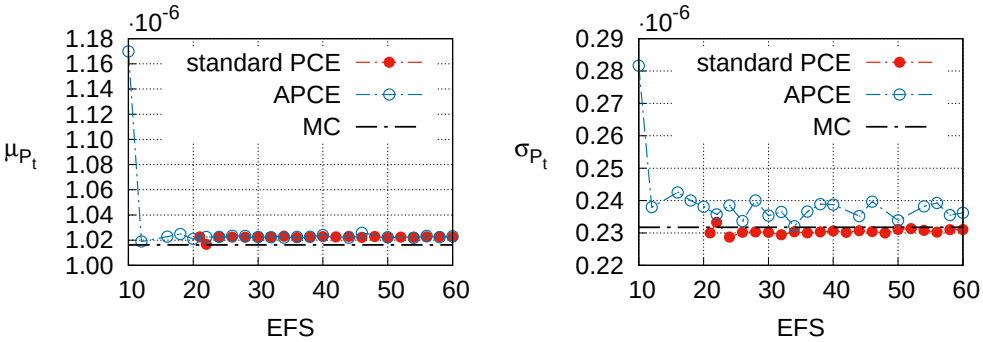


Figure 4.30: RDO Case 4. Statistical moments of P_t versus the EFS required to compute them with standard PCE and APCE. Values are compared with MC (5000 replicates). **Left**: μ_{P_t}. **Right**: σ_{P_t}.

computed using only 12 EFS with some loss in accuracy in σ_{P_t} (approximately 3%), w.r.t. the results of the MC.

An optimization without uncertainties was also carried out. The initial geometry, the one optimized with pAPCE and the one optimized without uncertainties were re-evaluated using the MC. Based on these re-evaluations, both optimization runs reduced μ_{P_t} by ∼33% and σ_{P_t} was reduced by 35.6% using pAPCE and by 34.1% in the optimization without uncertainties.

Chapter 5

Industrial Applications

This chapter presents the use of the methods developed in chapters 3 and 4 to industrial applications. In sections 5.1 and 5.2, the RPM is used to assist convergence of steady flow and adjoint solvers for shape optimization in automotive applications followed by re-evaluations of the optimized solutions with URANS solvers. In section 5.3, pFOSM and pAPCE are used to optimize a wing with symmetric cross section and minimize C_D through a distribution of a twist angle along its span.

5.1 Shape Optimization of a Car Spoiler and Diffuser

The spoiler and diffuser sections of a car [116] were optimized for min. C_D. The flow had a $Re \approx 8.8 \cdot 10^6$, based on the length of the car. The grid consisted of $1.2 \cdot 10^6$ cells and two VBS morphing boxes, shown in fig. 5.1, were used for the parameterization of the spoiler and diffuser of the car. The vehicle speed was equal to $33m/s$. The BCs imposed on the wheels modeled their rotation by assigning an angular velocity which, upon contact with the ground, caused the peripheral velocity at the surface of the wheels to be equal to the speed of the car. The design variables were updated using the conjugate gradient method [28].

The flow equations' residuals, shown in fig. 5.2, initially stagnated. The RPM assisted the solver in reducing the flow equations' residuals for up to 3 orders of magnitude from the point where stagnation began. Beyond this point, the residuals and the objective function still oscillated but to a much smaller extent. The objective function was converged up to the 5th significant digit. The RPM was activated after 2000 iterations and 400 iterations were allowed to pass before the addition of new modes. Based on the so-computed primal fields, the adjoint equations diverged and were stabilized by combining the RPM with two smoothing iterations on L for damping the ATC. For the adjoint solution, the RPM was activated after 600 iterations and 500 iterations were used in between consecutive basis augmentations. 15 flow/adjoint solution snapshots were used in both cases for the construction of D (eq. 3.24).

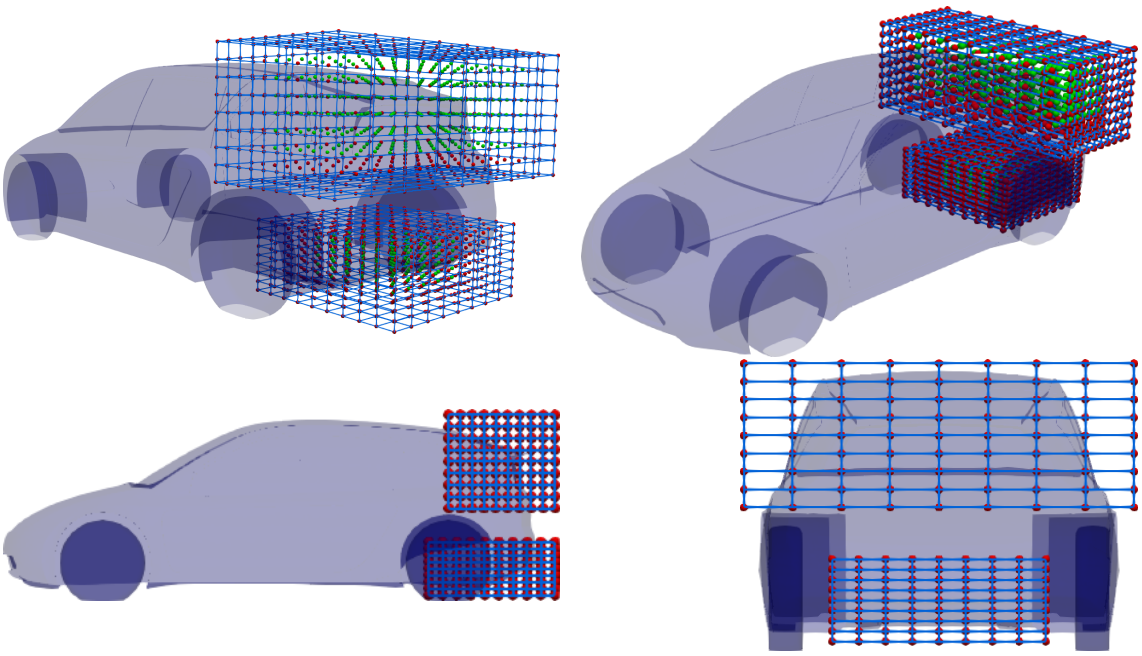


Figure 5.1: Shape Optimization of a Car. Views of the car geometry. Two $10 \times 9 \times 9$ VBS morphing boxes were used to parameterize the spoiler area and the diffuser of the car.

With this setup, the optimization began and after 11 cycles, C_D was reduced by 4.8%, shown in fig. 5.3. Beyond this point, the displacement of the morphing boxes' CPs created an invalid grid and the optimization loop could not proceed any further. The resulting shape is shown in fig. 5.4. The diffuser section was pulled outwards whereas the side section of the spoiler was pushed inwards and the top side downwards. Re-evaluations using URANS were performed for 3 of the 11 geometries. A time-integration window of 2 sec. was used for the computation of $\overline{C_D}$. Although the RANS-based solution greatly underestimated the value of the objective function, the achieved reduction in $\overline{C_D}$ was 4.9%, which was not that far from the value computed by the RANS-based solver.

Regarding the cost, 11 optimization cycles required approximately 4.5 hours on 156 processors (Intel© Xeon© CPU E5-2620 v2, 2.10GHz) using the RPM and steady solvers. Indicatively, with this particular time-integration window, the URANS flow solver required approximately 15 hours to complete a single flow analysis on the same hardware.

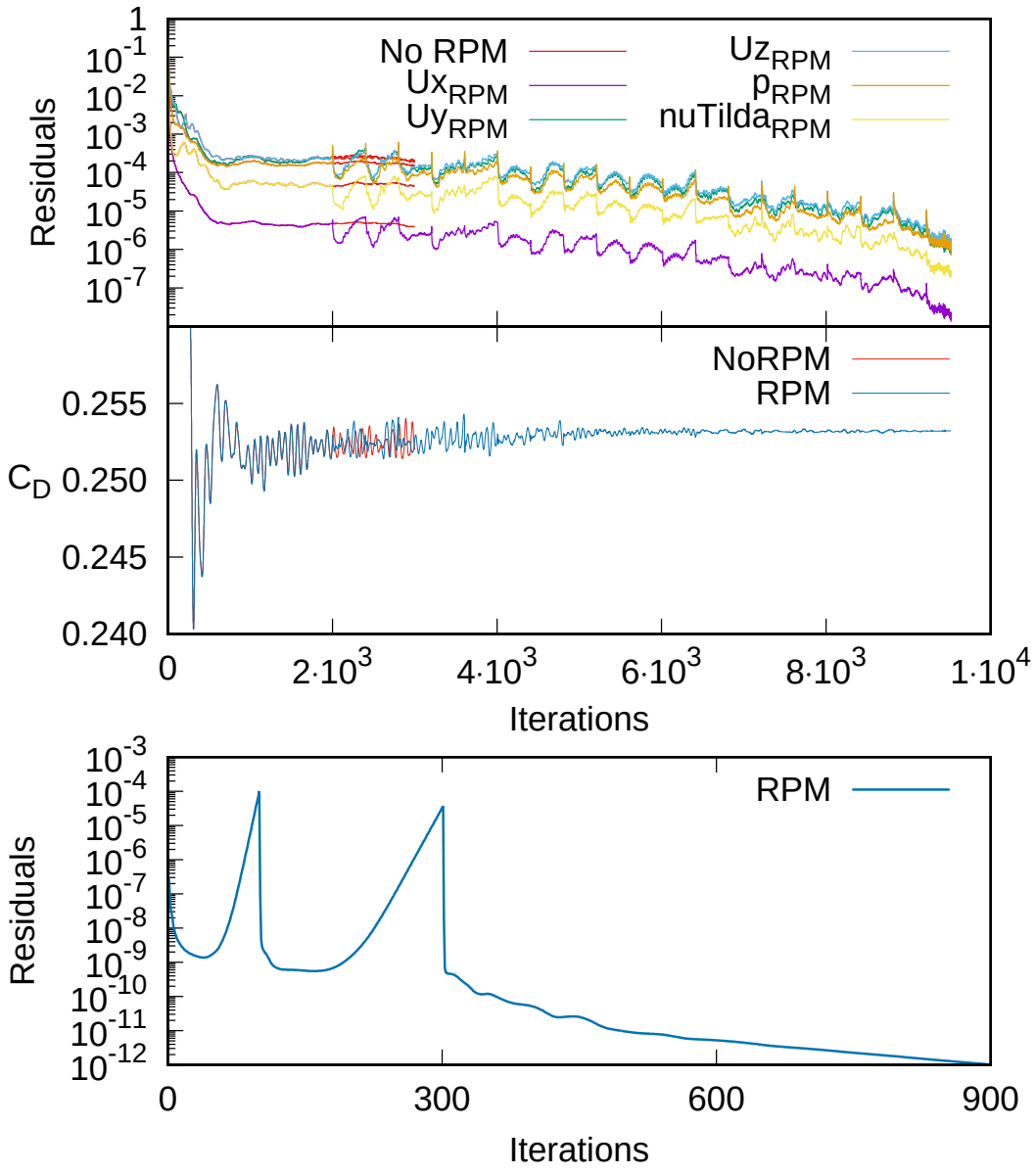


Figure 5.2: Shape Optimization of a Car. **Top**: Flow equations' residuals. **Middle**: Convergence of C_D after the use of the RPM. C_D still oscillates but only within a small margin (5th significant digit). **Bottom**: Geometric mean of the adjoint equations' residuals, solved with the RPM and mild ATC damping. The adjoint solution is based on the RPM-stabilized flow-fields.

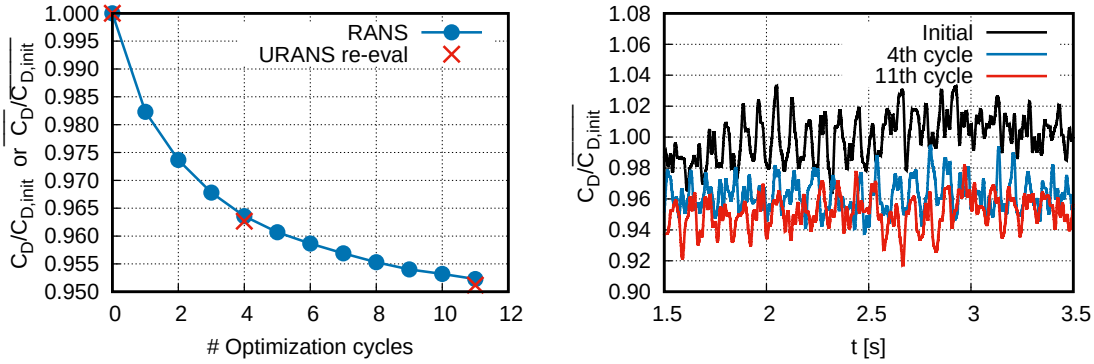


Figure 5.3: Shape Optimization of a Car. **Left**: C_D minimization within 11 optimization cycles and $\overline{C_D}$ from URANS re-evaluations of the initial design and after the end of the 4th and 11th optimization cycles. **Right**: Instantaneous C_D within the integration time-window for the URANS re-evaluation. All values were normalized w.r.t. the ones of the initial design.

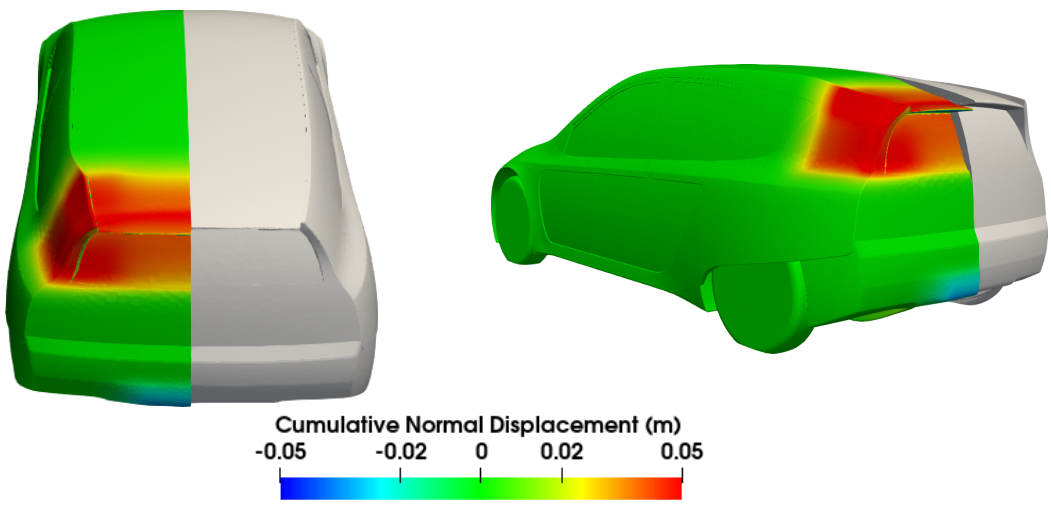


Figure 5.4: Shape Optimization of a Car. Split comparison between the initial and optimized exterior of the car along the plane aligned with the streamwise direction of the flow. The initial is portrayed in grey whereas the optimized is colored by the cumulative normal displacement field where positive and negative displacement values indicate inwards and outwards displacement of the car surface, respectively.

5.2 Shape Optimization of a Motorbike Fairing

In this section, the fairing of a motorbike was optimized for min. $C_D A$ (A is the motorbike frontal surface) computed over the entire vehicle. The flow had a $Re \approx 2.6 \cdot 10^6$, based on the length of the motorbike. The grid consisted of $14 \cdot 10^6$ cells and a $7 \times 7 \times 7$ VBS morphing box was used for the parameterization of the fairing and the enclosed grid around it; this is shown in fig. 5.5 along with the overall geometry.

The complexity of the geometry and of the generated grid, as well as the unsteadiness of the flow, made it difficult for the flow solver to converge even with the RPM. The workaround was to average the "pseudo-steady" flow fields which represented a better approximation to the flow, compared to just using one instantaneous flow solution about which the adjoint equations were linearized. The flow equations were solved for a total number of 2000 iterations and the flow fields were averaged over the last 500. These averaged fields were used as the "pseudo-steady" flow solution the adjoint solver was based on; the adjoint solver diverged too, as a result of flow unsteadiness combined with large values of the ATC term close to the motorbike surface. Divergence occurred rapidly, during the very first iterations of the adjoint solver, and the RPM was unable to form the required unstable subspace. To assure convergence, the RPM was combined with mild ATC damping. Based on the assessments carried out in sections 3.2.1 and 3.2.2, such an approach seems to be minimally invasive, w.r.t. the accuracy of the SDs. Although mild ATC damping could not achieve stability on its own, this treatment secured a window of iterations for the RPM which was enough to stabilize the adjoint solver. The regions where the ATC was damped can be seen in fig. 5.6.

Using both the RPM and the mild damping of the ATC, an optimization was

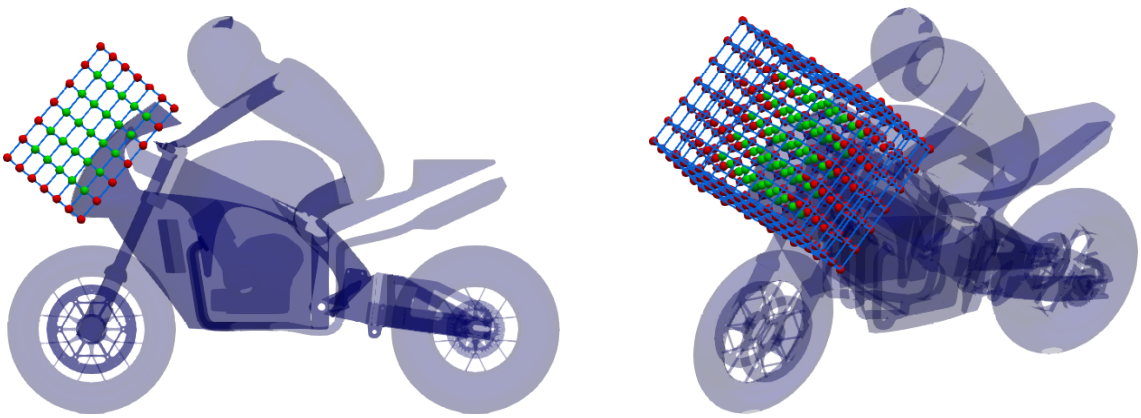


Figure 5.5: Motorbike case. Geometry and VBS morphing box parameterizing the surface points of the fairing and the enclosed grid. Only the green CPs were allowed to move during the optimization.

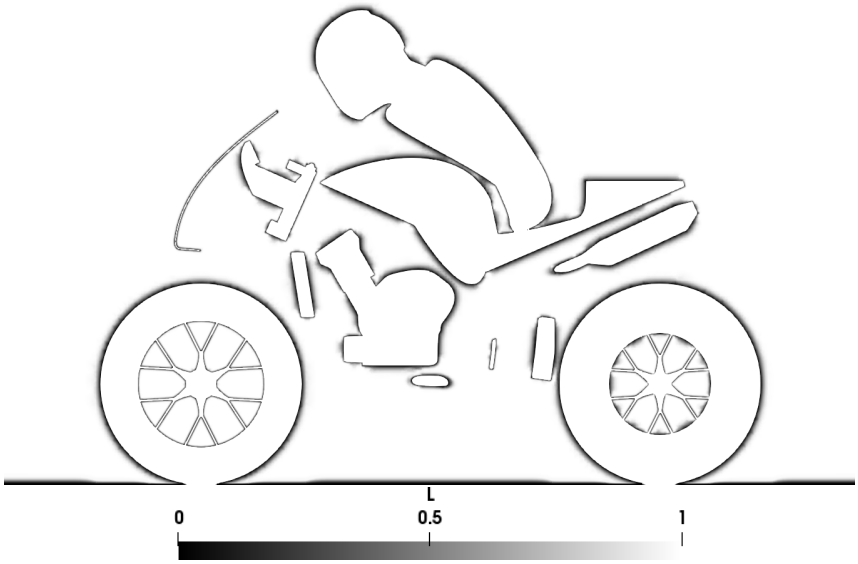


Figure 5.6: Motorbike case. Regions of ATC damping. The value of L, plotted along the symmetry plane of the motorbike, ranges from 0 to 1; 5 smoothing iterations were used. The shaded regions correspond to cells where the ATC was damped.

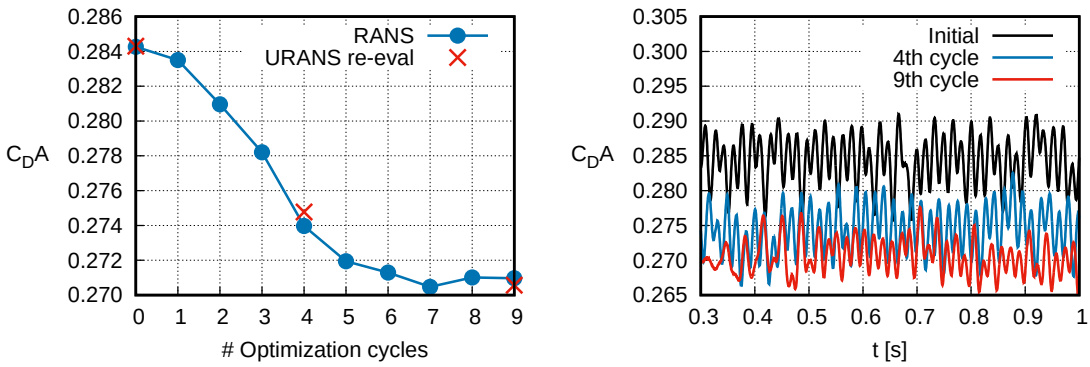


Figure 5.7: Motorbike case. **Left**: Convergence of C_DA during the shape optimization of the fairing within 9 optimization cycles. $\overline{C_DA}$ from re-evaluations of the initial, an intermediate (4th cycles) and the optimized (9th cycle) geometries are also plotted. Both C_DA and $\overline{C_DA}$ were reduced. **Right**: C_DA time history from the 3 URANS re-evaluations mentioned above.

conducted using the conjugate gradient method. After 9 optimization cycles, C_DA decreased by 4.6%, fig. 5.7, exclusively through modifications in the fairing shape, made visible in fig. 5.8. The RPM was activated after 1000 iterations and the check for updating the basis occurred every 500 iterations using 15 solution snapshots all the while. The size of V_p ranged from 1 to 3, except for one cycle where the basis increased to 7. The objective function was re-evaluated using an unsteady solver for 1 sec. and C_DA was averaged over a period of 0.7 sec., which proved

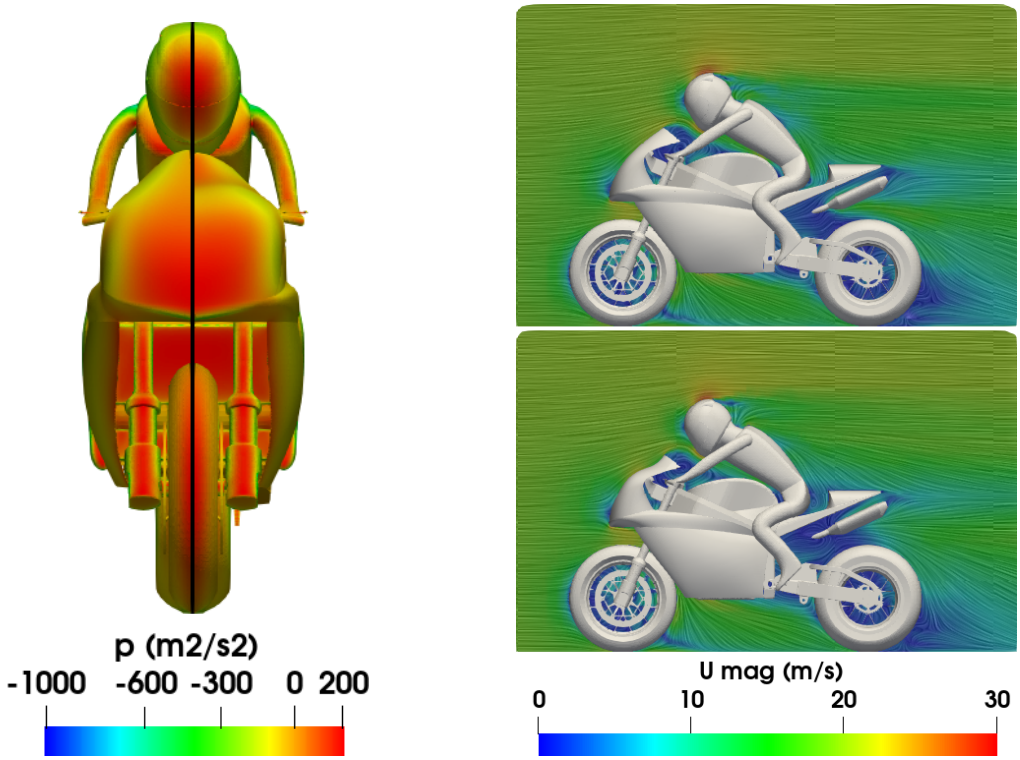


Figure 5.8: Motorbike case. **Left**: Pressure distribution on the initial (left) and optimized (right) motorbikes. Frontal view. **Right**: Mean flow velocity magnitude on the symmetry plane, normal to the spanwise direction of the flow, around the initial (top) and optimized (bottom) geometries.

to be long enough for the statistical convergence of $\overline{C_D A}$. Figure 5.7 shows that both solvers gave a very close estimate of the objective function's value. $\overline{C_D A}$ was indeed decreased by a similar factor as $C_D A$.

In this case, the adjoint solver, enhanced by a combination of the RPM and mild ATC damping, supported shape optimization with steady solvers at a cost and memory requirements which were much lower than the ones of the unsteady solvers. Indicatively, 9 cycles required approximately 20 hours on 156 processors (Intel© Xeon© CPU E5-2620 v2, 2.10GHz). This was, approximately, also the cost of a single unsteady optimization cycle, based on the compressed full storage of the entire flow series, using the scheme presented in [86].

So, in these two cases (sections 5.1 and 5.2) which represent best-case scenarios with an almost zero overhead for the unsteady adjoint solver, the CPU cost of a single steady and unsteady optimization cycle differ by one order of magnitude. Nevertheless, the outcome of this comparison may change significantly depending on how storage requirements of the unsteady adjoint solver are handled (e.g. use of check-pointing, compression algorithms [86, 87], reduced-order models, etc), on the duration of integration of the flow equations

and the number of modes identified by the RPM. Regarding the latter, the extra cost of the RPM implementation heavily depends upon the size of the V_p basis which is involved in the matrix-vector products that take place within each iteration of the steady solver. With the implementation of the RPM used in this work, the increase in CPU time required to perform each solver iteration may range between 10% and 40%, depending on the size of V_p which, in this case, ranged from 3 to 7 vectors. However, an increase in size is often accompanied by an increased convergence rate for the solver and, overall, this is an affordable cost to pay in order to get an otherwise unattainable solution.

5.3 Wing Shape Optimization

This stands for one of the AIAA Aircraft Design Optimization Discussion Group (ADODG) cases (namely Case 3 [49]). These cases were developed as benchmarks for the validation and comparison of shape optimization tools. The objective, in this case, is the reduction of C_D only through modifications of the twist distribution of the wing about the axis passing through the trailing edge, under constraints on C_L and, sometimes, C_M as well. The twist angle is distributed across the wing along the spanwise direction. Although originally developed as an application for inviscid transonic flow, the case was studied under different conditions in [70] and the flow was modelled as an incompressible one, for the purpose of comparing two discrete adjoint solvers, namely *ADflow* [1] and *DAFoam* (OpenFOAM-based) [19]. Here, the optimization was carried out under the assumption that v_∞ and α_∞ followed normal distributions with $\mu_{v_\infty} = 50m/s$, $\sigma_{v_\infty} = 10m/s$ and $\mu_{\alpha_\infty} = 5°$, $\sigma_{\alpha_\infty} = 1°$. Both pFOSM and pAPCE were used to do this.

The grid file used to produce the results for this thesis was drawn from the database in [68]. The geometry was created by extruding the NACA0012 cross sectional area along the spanwise direction and adding a tip at the end. The case had a $Re = 1.1 \cdot 10^6$ and $A_c = 0.27$ (chord length squared times the spanwise length without the wing-tip [49]). VBS were used to parameterize the wing surface, the CPs of which can be seen in fig. 5.9 along with an outline view of the computational grid. Each plane of CPs was twisted about the axis that coincides with the leading edge of the wing. By varying the twist angle of each plane, the twist was distributed along the wing span.

SDs of statistical moments of C_D and C_L w.r.t. b_n are shown in fig. 5.10. Based on these results, in the case of pAPCE and for the purpose of imposing constraints on μ_{C_L}, as well as for μ_{C_D} minimization only, mixed derivatives (eq. 4.20) were neglected, saving significant computational time. Slight differences were expected in the optimized wings between the two methods due to the differences between the SDs of σ_{C_D}.

RDO was carried out using both pFOSM and pAPCE using values of w (eq. 4.1)

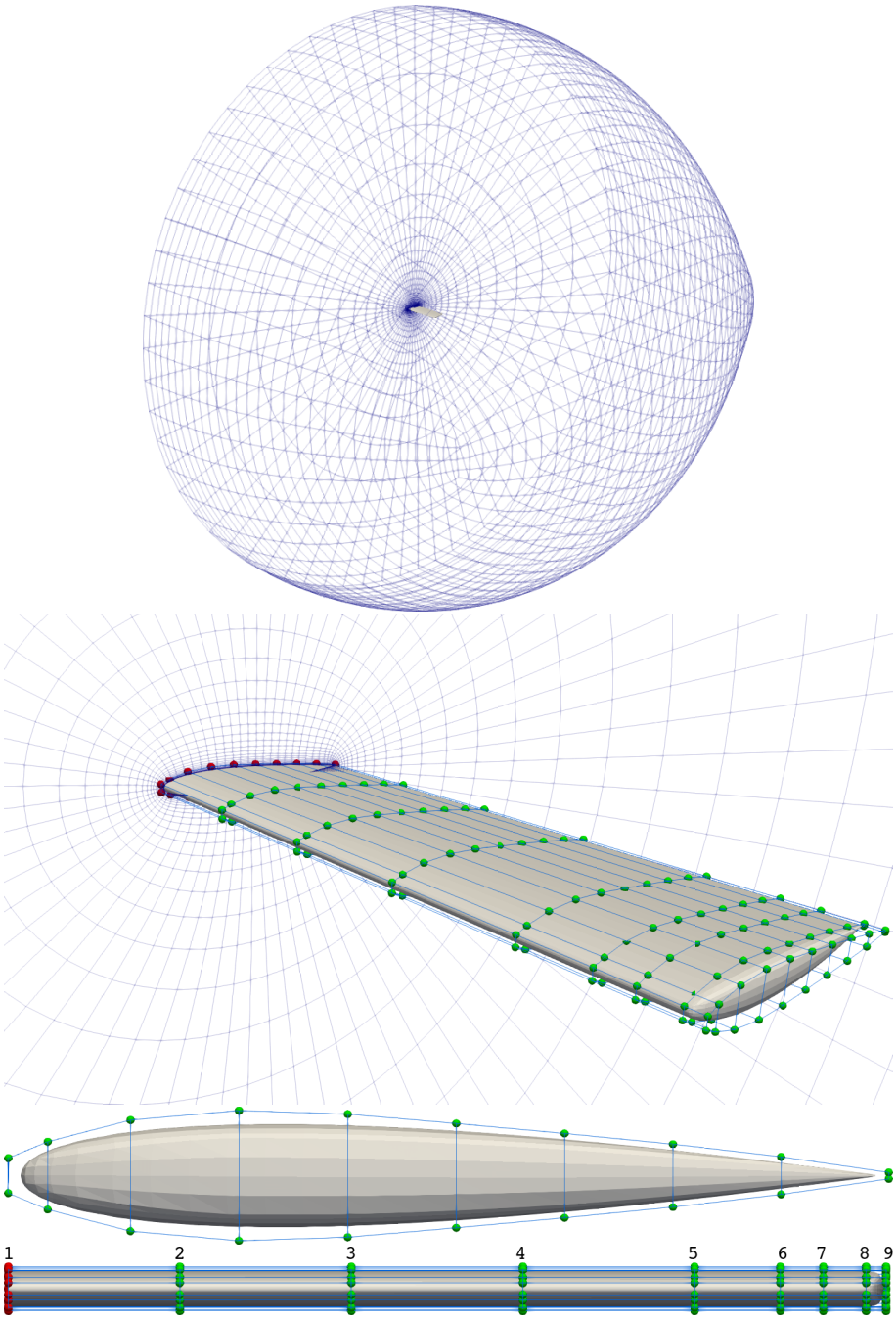


Figure 5.9: ADODG3 Wing Case. **Top**: Overview of the wing and grid. **Bottom**: VBS morphing box enclosing the wing to be optimized. Views are: the wing at an arbitrary angle (top), the wing tip (middle) and the front of the wing. Each plane of CPs, perpendicular to the spanwise direction, rotated rigidly about the axis passing through the leading edge of the wing. Green CPs were the ones to be twisted about whereas red ones were situated exactly on the symmetry patch of the grid and remained fixed throughout the optimization.

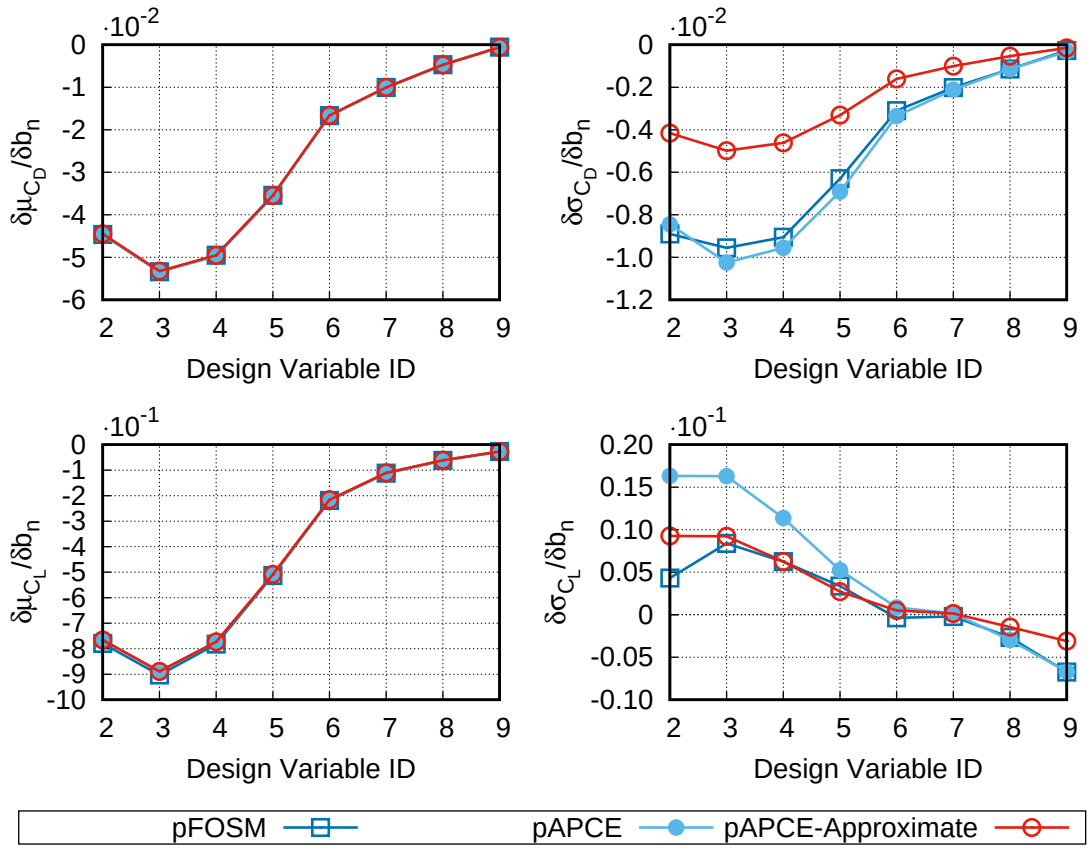


Figure 5.10: ADODG3 Wing Case. SDs of statistical moments of C_D and C_L w.r.t. b_n i.e. the twist angles of the planes of CPs (fig. 5.9). **Top**: C_D. **Bottom**: C_L. **Left**: Mean values. **Right**: Standard deviations.

from 0 to 6 with a step of 1. With pAPCE, $O_{PCE}=3$ and $L = 8$. The SQP method was used to impose an equality constraint on μ_{C_L} and $\mu^{\oplus}_{C_L}$ was chosen equal to the initial value. The convergence of the optimization loop, only for the cases with $w = 0$, $w = 3$ and $w = 6$, is shown in fig. 5.11. The optimization runs required between 30 and 60 optimization cycles to complete. The μ_{C_L} constraint was satisfied within an acceptable $\pm 10^{-4}$ threshold. The two RDO methods yield different estimated values, both for μ_{C_D} and σ_{C_D} (also for μ_{C_L}). The optimized wings are shown in fig. 5.12. All the optimized solutions, with both pFOSM and pAPCE, were re-evaluated with MC using 1000 replicates. A Pareto front was formed in the $\mu_{C_D} - \sigma_{C_D}$ space, shown in fig. 5.13. The optimized solutions computed with pAPCE slightly dominated the ones computed with pFOSM within a selected region of the front. More points would be required to draw a conclusion for the rest of it. The cost of the optimization was equal to 5 EFS with pFOSM and 40 EFS with pAPCE.

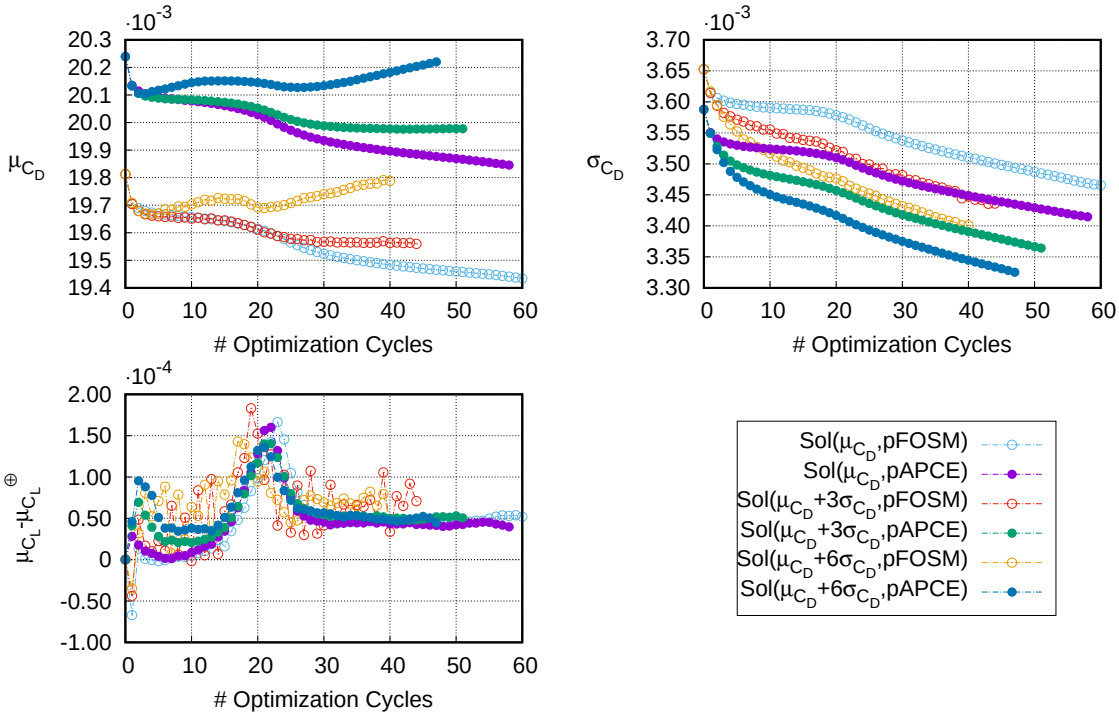


Figure 5.11: ADODG3 Wing Case. Shape optimization loop convergence. **Top-Left**: μ_{C_D}. **Top-Right**: σ_{C_D}. **Bottom-Left**: μ_{C_L}. The convergence shown corresponds to optimization runs performed using pFOSM and pAPCE for $J = \mu_{C_D}$, $J = \mu_{C_D} + 3\sigma_{C_D}$ and $J = \mu_{C_D} + 6\sigma_{C_D}$.

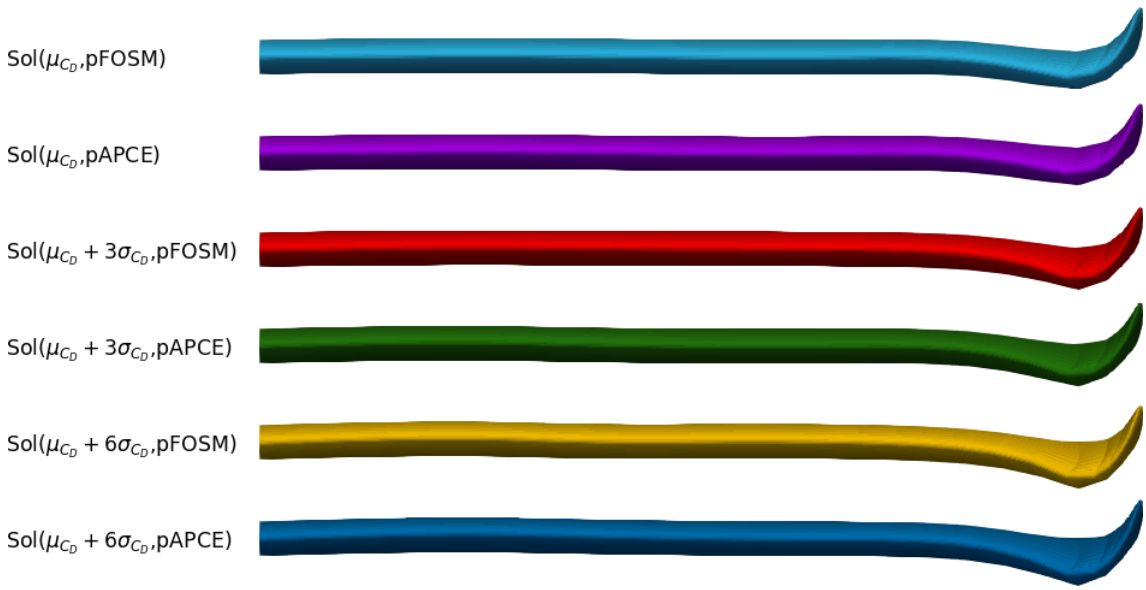


Figure 5.12: ADODG3 Wing Case. Frontal views of the optimized wings computed using pFOSM and pAPCE with $J = \mu_{C_D}$ and $J = \mu_{C_D} + 3\sigma C_D$.

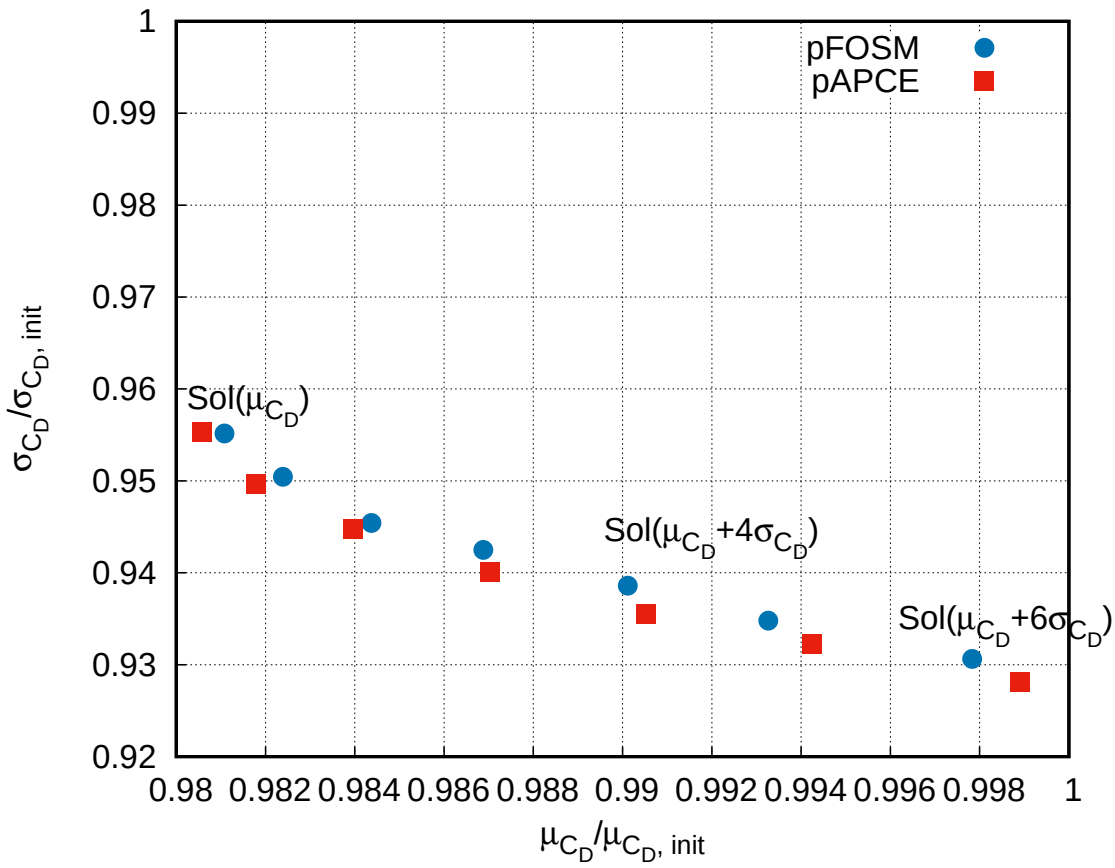


Figure 5.13: ADODG3 Wing Case. Pareto front of solutions in the μ_{C_D} and σ_{C_D} space. The two fronts were computed with pFOSM and pAPCE but the points shown were all re-evaluated with the MC (1000 replicates).

Chapter 6

Closure

This PhD thesis expands previous work in the PCOpt Unit of NTUA on aerodynamic shape optimization based on the continuous adjoint method. The contributions pertain to two distinct areas:

- The treatment of convergence difficulties of steady primal (flow) and (continuous) adjoint solvers, caused by flow unsteadiness.
- Cost reduction of shape optimization in the presence of uncertainties.

Common theme among these two areas is the focus on the reduction of the computational cost of the optimization. The methods are not restricted to shape optimization and can very well be applied to other areas of CFD-based optimization, such as topology optimization. Although the two subjects are presented independently, they can also be used synergistically, depending on the application. For each one of the two areas, a recapitulation of the major steps taken into this thesis follows, followed by conclusions, findings and discussion.

6.1 Conclusions, Findings and Discussion

6.1.1 Stabilization of Steady Primal and Adjoint Solvers

The first section of the thesis dealt with the convergence difficulties of steady primal and adjoint solvers, commonly encountered in gradient-based aerodynamic shape optimization problems in which the flow exhibits unsteadiness. In case the unsteadiness in these problems is mild, steady solvers can be used to avoid the large computational overhead associated with unsteady ones in optimization loops. A stabilization method, called the Recursive Projection Method (RPM), was used to make the steady primal and adjoint solvers converge. Additionally, two existing and widely used remedies for overcoming the aforementioned convergence difficulties were sometimes combined with the RPM for increased efficiency and robustness. The first one involves the solution of the adjoint equations based on “pseudo-steady” primal

fields, averaged over a number of iterations of a stalled primal solver. The second one involves the damping of the ATC, a term that appears in the adjoint momentum equations and is a frequent cause of divergence of continuous adjoint solvers. On their own, these remedies may damage the accuracy of the SDs. This impact was assessed with the help of the RPM, thanks to which "correct" reference steady solutions were made available for both primal and adjoint solvers. Following this assessment, some guidelines were derived.

The RPM can be used to assist the convergence of primal solvers in aerodynamic shape optimization with flows that exhibit mild unsteadiness and, even, vortex shedding. The implementation of the RPM ensures the minimization of the residuals of the primal PDEs and, thus, flow-field averaging can be avoided. In addition, adjoint solvers can also be stabilized by the RPM and this enables their use in the optimization loop. From a purely physical standpoint, in optimization problems such as the ones examined in this thesis, the objective is the time-averaged aerodynamic force acting on a surface within the flow. By using steady solvers, stabilized with the RPM, the turnaround time for this kind of optimization problems can become one order of magnitude less than that of URANS-based optimization, as shown in sections 5.1 and 5.2. Instead of the time-averaged aerodynamic coefficient, the optimization minimizes the coefficient computed with the stabilized steady primal solver. Although the values of these two objective functions may vary, the re-evaluation of the optimized solutions with an unsteady primal solver can verify the optimization outcome and the reduction of the time-averaged objective function. Most importantly, the objective function reduction percentages computed by the steady and unsteady solvers can be very close to each other. Even when unsteady-based adjoint cannot be avoided, shape optimization with steady solvers can significantly reduce the CPU time by providing a better initialization for the URANS-based optimization loop. This is observed in optimization of vortex-shedding flows in section 3.2.2. Although the use of steady solvers in such cases seems counter-intuitive, the proposed methodology operates well in case the objective function is defined using time-averaged quantities.

In applications involving complex geometries, such as in section 5.2 (fairing, wheels, exhaust etc), it might be difficult to generate a good-quality grid and, also, maintain it throughout the optimization. On top of this, flow unsteadiness may be caused by an excessive number of unstable modes that are difficult to track and handle with the RPM. Together, these issues amplify the convergence difficulties of steady solvers. A major problem in such complex cases is that the adjoint solver may diverge rapidly, before the RPM can do anything to stabilize it. This problem can be overcome by assisting the RPM with a controlled damping of the ATC term. According to the findings of this thesis, the use of mild ATC damping, in synergy with the RPM, increases the robustness of the adjoint solver. This approach is not to be confused with the previous strategy, the uncontrollable damping of the

ATC, which has a detrimental effect on the SDs. The role of the RPM still remains pivotal because, without it, an excessive ATC damping would have been required to stabilize the adjoint solvers with unforseen consequences on the accuracy of the computed gradients. Similarly, thanks to the RPM, the use of "pseudo-steady" primal solutions is largely avoided although, in the cases studied in this thesis, this technique did not cause any significant damage to the optimization.

6.1.2 Shape Optimization in the Presence of Uncertainties

Setting aside the issues of solver stabilization, the second section of this thesis dealt with shape optimization using gradient-based methods in the presence of uncertainties. The cost of this optimization can increase dramatically as more uncertain variables become involved. A framework was developed to reduce this cost. Within this framework, two distinct methods were presented for the minimization of objective functions expressed as a weighted sum of the two statistical moments, the mean value and standard deviation, of a QoI. These methods were treated as extensions of existing UQ techniques. Gradient-based optimization relying on these two UQ methods involved the computation of a matrix of mixed derivatives of QoIs w.r.t. design and uncertain variables. Their costly computation was overcome by computing projections of this matrix on vectors instead and this projection technique was the main idea behind the two methods. This technique was facilitated by the solution of two systems of PDEs resulting from the differentiation of the primal and adjoint PDEs w.r.t. uncertain variables and their subsequent projection on appropriate vectors. Both methods were used to optimize the shape of aerodynamic bodies while taking uncertainties related to the farfield boundary conditions into account.

The first method is called pFOSM and was derived from a first-order Method of Moments. It uses the (continuous) adjoint method to compute the two statistical moments of the QoI at a low cost. In pFOSM, the optimization can be performed by solving only four systems of PDEs at a cost that is equivalent to solving four times the flow equations, per optimization cycle. As seen in section 4.2.3, the cost of computing the projected matrix of mixed derivatives and, thus, the cost of pFOSM, is independent of the number of design and uncertain variables.

The second method, called pAPCE, is based on Polynomial Chaos Expansions. It makes use of the continuous adjoint method to reduce the number of collocation points involved in regression-based PCE. Compared to standard PCE, fewer systems of PDEs need to be solved. Additionally, thanks to the projection technique, the optimization cost of pAPCE is equivalent to solving four systems of flow PDEs times the number of collocation points, per optimization cycle. For each collocation point, the projected matrix of mixed derivatives is computed at a cost that is independent of the number of design and uncertain variables. A significant cost reduction can be achieved if approximate sensitivities are computed by totally omitting the mixed derivatives,

irrespective of the projection. In sections 4.3.1 and 4.3.2.2, this is seen to be harmless in case only gradients of the mean value of a QoI are required, such as when imposing constraints on the mean values of aerodynamic forces and moments. Thus, in case working with approximate derivatives is not an issue, mixed derivatives can be ignored for higher efficiency. In both methods, flow-related constraints should be accounted for in the costs mentioned above. In turbulent flows, the adjoint to the turbulence model equation(s) are involved in the developed methods; none of them made the "frozen" turbulence assumption.

Both pFOSM and pAPCE can benefit a lot from the use of projections. The key point of the developed framework is the ability to compute the gradient of objective functions at a cost that is almost equal to the cost of UQ. Overall, pAPCE is more costly than pFOSM but at the same time more accurate in the computation of statistical moments. It is therefore up to the designer to choose between the pFOSM for efficiency or pAPCE for accuracy.

6.2 Novel Contributions

- The computational cost reduction of shape optimization for incompressible flows with mild unsteadiness through the treatment of convergence difficulties of the steady primal and adjoint solvers by the RPM. It is also seen how the cost of an unsteady-based optimization loop for a vortex shedding flow can be reduced if a preceding optimization loop is performed to compute a better initialization point using the proposed approach.

- The assessment of widely used remedies for overcoming the convergence difficulties of primal and adjoint solvers, such as flow-field averaging and ATC damping, w.r.t. the accuracy of the SDs. Thanks to the RPM stabilization (which gave reference solutions to compare with), practical guidelines are derived: Only a mild ATC damping, effective only in synergy with the RPM, is harmless whereas excessive damping should be avoided. Flow-field averaging does not seem to have such a negative effect but can totally be avoided in many cases with the RPM.

- The development and presentation, for the first time in the literature, of a new method, referred to as pAPCE, shape optimization under uncertainties with lower cost than standard regression-based non-intrusive PCE. The new method allows the computation of the mean value and standard deviation of a QoI at a reduced cost, compared to standard regression-based PCE, thanks to the inclusion of gradient information without though suffering a costly penalty in the computation of statistical moments' sensitivities. A combination of adjoint and direct differentiation along with a projection technique were used to achieve this outcome.

- The extension of both pAPCE and pFOSM methods for use in turbulent flows, modelled using the Spalart-Allmaras turbulence model. To this end, the Spalart-Allmaras model PDE and its adjoint were differentiated w.r.t. uncertain variables (related to farfield flow conditions).

6.3 Publications and Conference Presentations

Journal Publications:

- T. Skamagkis, E.M. Papoutsis-Kiachagias, K.C. Giannakoglou. "On the Stabilization of Steady Continuous Adjoint Solvers in the Presence of Unsteadiness, in Shape Optimization". International Journal of Numerical Methods in Fluids 2021; 93(8):2677-2693

- Th. Skamagkis, E.M. Papoutsis-Kiachagias, K. Giannakoglou. "CFD-based shape optimization under uncertainties using the Adjoint-assisted Polynomial Chaos Expansion and projected derivatives". Computers & Fluids 2022; 241:105458

Conference Presentations:

- T. Skamagkis, A.-S.I. Margetis, E.M. Papoutsis-Kiachagias and K.C. Giannakoglou. "On the efficiency and robustness of the adjoint method: Applications in steady and unsteady shape optimization in fluid mechanics". 8th OpenFOAM Conference, Digital Event, October 13-15, 2020.

- E.M. Papoutsis-Kiachagias, M. Farhikhteh, T. Skamagkis and K.C. Giannakoglou. "Aerodynamic Shape Optimization of the MEXICO Wind Turbine Blade using the Continuous Adjoint Method". EUROGEN 2021, 14th International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control, online from Athens, Greece, June 28-30, 2021.

- E.M. Papoutsis-Kiachagias, K.T. Gkaragkounis, A.-S.I. Margetis, T. Skamagkis, V.G. Asouti and K.C. Giannakoglou. "*adjointOptimisationFoam*: An OpenFOAM-based Framework for Adjoint-assisted Optimization". EUROGEN 2021, 14th International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control, online from Athens, Greece, June 28-30, 2021.

- Th. Skamagkis, E.M. Papoustis-Kiachagias, K.C. Giannakoglou. "Adjoint-Based Aerodynamic Shape Optimization for Turbulent Flows in the Presence of Uncertainties Using the Method of Moments and

Projections". 17th OpenFOAM Workshop, Cambridge, UK, 11-14 July, 2022.

- Th. Skamagkis, E.M. Papoutsis-Kiachagias, K.C. Giannakoglou. "Aerodynamic Shape Optimization Under Uncertainties using Adjoint-assisted PCE and Projections." ASMO-UK 12 / ASMO-Europe 1 / ISSMO Conference on Engineering Design Optimization, Leeds, UK, 18-19 July, 2022.

6.4 Future Work

For the RPM:

- Further automate the choice of parameters such as the Krylov criterion.
- Assess the combination of the RPM with block-coupled primal or block-coupled adjoint solvers for enhanced stability and ease of computing the unstable subspace basis through direct use of the Jacobian matrix.

For the projection technique:

- Extension of the projection method to SOSM where higher-order terms appear.
- Investigate whether pAPCE can benefit from better sampling techniques.
- Combine the projection technique with variants of APCE available in the literature.
- Extend the projection technique for other turbulence models the adjoint of which are available in the *adjointOptimisation* library or exist in-house.

Appendix A

Adjoint Boundary Conditions for Specific Objective Functions

This appendix gives the forms of the adjoint BCs, the SDs in eqs. 2.21, 2.29 and 4.23 and the projected matrices of mixed derivatives in eqs. 4.26 and 4.27 for the objective functions (or QoIs) utilized in this thesis.

A.1 Force

In the external aerodynamics where the objective function is the aerodynamic coefficient of the force exerted on S_W, this is equal to

$$F = \int_{S_W} \left(p\delta_i^j - \tau_{ij}\right) n_j r_i dS \tag{A.1}$$

where $\boldsymbol{n}$ is the normal to the surface unit vector pointing outwards of the fluid domain and $\boldsymbol{r} = \widetilde{\boldsymbol{r}}\frac{1}{0.5v_\infty^2 A_c}$, where $\widetilde{\boldsymbol{r}}$ is a unit vector aligned with the force component which is of interest; A_c denotes a reference area or length, depending on the dimensions of the problem. For airfoils, this is equal to the length of the airfoil chord. The adjoint BCs are

$$\begin{aligned}
&S_I : u_i = 0, \quad \frac{\partial q}{\partial n} = 0, \quad \widetilde{\nu_\alpha} = 0, \quad \Delta_\alpha = 0 \\
&S_O : \frac{\partial u_{\langle n\rangle}}{\partial n} = 0, \quad v_{\langle n\rangle} u_{\langle t^l\rangle} + \tau_{ij}^\alpha t_i^l n_j + \widetilde{\nu_\alpha}\widetilde{\nu}\frac{C_\omega}{\omega} Y_k \epsilon_{kji} n_j t_i^l = 0 \quad l = I, II \\
&\qquad -q + v_{\langle n\rangle} u_{\langle n\rangle} + \tau_{ij}^\alpha n_i n_j + \widetilde{\nu_\alpha}\widetilde{\nu}\frac{C_\omega}{\omega} Y_l \epsilon_{lji} n_j n_i = 0, \quad \widetilde{\nu_\alpha} v_{\langle n\rangle} + \frac{\nu + \widetilde{\nu}}{c_\sigma}\frac{\partial \widetilde{\nu_\alpha}}{\partial n} = 0, \quad \Delta_\alpha = 0 \\
&S_W : u_i = -r_i, \quad \frac{\partial q}{\partial n} = 0, \quad \widetilde{\nu_\alpha} = 0, \quad \frac{\partial \Delta_\alpha}{\partial n} = 0 \\
&S_S : u_{\langle n\rangle} = 0, \quad \frac{\partial u_{\langle t^l\rangle}}{\partial n} = 0 \quad l = I, II, \quad \frac{\partial q}{\partial n} = 0, \quad \frac{\partial \widetilde{\nu_\alpha}}{\partial n} = 0, \quad \frac{\partial \Delta_\alpha}{\partial n} = 0
\end{aligned} \tag{A.2}$$

The SDs expressions become,

$$\left.\frac{\delta F}{\delta b_n}\right|_{FI} = \int_{S_{W_p}} \left(p\delta_i^j - \tau_{ij}\right) r_i \frac{\delta(n_j dS)}{\delta b_n} + \int_{\Omega} \left(\mathcal{A}_{jk}^L + \mathcal{A}_{jk}^T\right) \frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right) d\Omega \tag{A.3}$$

for the FI, and

$$\left.\frac{\delta F}{\delta b_n}\right|_{E-SI} = \int_{S_{W_p}} \left(p\delta_i^j - \tau_{ij}\right) r_i \frac{\delta(n_j dS)}{\delta b_n} + \int_{S_{W_p}} \left(\mathcal{A}_{jk}^L + \mathcal{A}_{jk}^T\right) n_j \frac{\delta x_k}{\delta b_n} dS - \int_{S_{W_p}} \frac{\partial m_k^\alpha}{\partial x_j} n_j \frac{\delta x_k}{\delta b_n} dS \tag{A.4}$$

for the E-SI adjoint. With the farfield-flow angle and velocity magnitude as the uncertain variables, the sensitivities are equal to

$$\frac{\delta F}{\delta c_m} = \int_{S_W} \left(p\delta_i^j - \tau_{ij}\right) n_j \frac{\partial r_i}{\partial c_m} dS + \int_{S_I} \left(-q n_i + v_j u_i n_j + \tau_{ij}^\alpha n_j + \widetilde{\nu}_\alpha \widetilde{\nu} \frac{C_\omega}{\omega} Y_q \epsilon_{qji} n_j\right) \frac{\delta v_i}{\delta c_m} dS \tag{A.5}$$

where

$$\frac{\partial r_i}{\partial v_\infty} = -\frac{\widetilde{r}_i}{0.25 v_\infty^3 A_c}\ ,\ \frac{\partial \boldsymbol{r}}{\partial \alpha_\infty} = [-r_2 \quad r_1]^T \tag{A.6}$$

and

$$\left.\frac{\partial \boldsymbol{v}}{\partial v_\infty}\right|_{S_I} = [cos(\alpha_\infty) \quad sin(\alpha_\infty)]^T, \quad \left.\frac{\partial \boldsymbol{v}}{\partial \alpha_\infty}\right|_{S_I} = v_\infty[-sin(\alpha_\infty) \quad cos(\alpha_\infty)]^T \tag{A.7}$$

The projected mixed derivatives matrix is equal to

$$\begin{aligned} z_m \left.\frac{\delta^2 F}{\delta c_m \delta b_n}\right|_{FI} = & \int_{S_{W_p}} \left[\left(\widehat{p}\delta_i^j - \widehat{\tau}_{ij}\right) r_i + \left(p\delta_i^j - \tau_{ij}\right) \widehat{r}_i\right] \frac{\delta(n_j dS)}{\delta b_n} dS \\ & + \int_{\Omega} \left(\widehat{\mathcal{A}}_{jk}^L + \widehat{\mathcal{A}}_{jk}^T\right) \frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right) d\Omega \end{aligned} \tag{A.8}$$

for the FI adjoint and

$$\begin{aligned} z_m \left.\frac{\delta^2 F}{\delta c_m \delta b_n}\right|_{E-SI} = & \int_{S_{W_p}} \left[\left(\widehat{p}\delta_i^j - \widehat{\tau}_{ij}\right) r_i + \left(p\delta_i^j - \tau_{ij}\right) \widehat{r}_i\right] \frac{\delta(n_j dS)}{\delta b_n} dS \\ & + \int_{S_{W_p}} \left(\widehat{\mathcal{A}}_{jk}^L + \widehat{\mathcal{A}}_{jk}^T\right) n_j \frac{\delta x_k}{\delta b_n} dS - \int_{S_{W_p}} \frac{\partial \widehat{m}_k^\alpha}{\partial x_j} n_j \frac{\delta x_k}{\delta b_n} dS \end{aligned} \tag{A.9}$$

for the E-SI adjoint.

A.2 Moment

The moment coefficient, in external aerodynamics, is equal to

$$F = \int_{S_W} \left(p\delta_i^j - \tau_{ij}\right) n_j \left(x_k - \overline{x}_k\right) r_l \epsilon_{ikl} dS \tag{A.10}$$

where $\boldsymbol{r} = \widetilde{\boldsymbol{r}} \frac{1}{0.5 v_\infty^2 A_c l_c}$ with $\widetilde{\boldsymbol{r}}$ a unit vector corresponding to the rotation axis passing from $\overline{x}_k$ and l_c denotes a reference length. The adjoint BCs are the same as in eq. A.1 with the exception of

$$S_W : u_i = -\left(x_k - \overline{x}_k\right) r_l \epsilon_{ikl} \tag{A.11}$$

The SDs expressions become,

$$\begin{aligned} \left.\frac{\delta F}{\delta b_n}\right|_{FI} &= \int_{S_{W_p}} \left(p\delta_i^j - \tau_{ij}\right) n_j r_l \epsilon_{ikl} \frac{\delta x_k}{\delta b_n} dS + \int_{S_{W_p}} \left(p\delta_i^j - \tau_{ij}\right) \left(x_k - \overline{x}_k\right) r_l \epsilon_{ikl} \frac{\delta (n_j dS)}{\delta b_n} \\ &+ \int_\Omega \left(\mathcal{A}_{jk}^L + \mathcal{A}_{jk}^T\right) \frac{\partial}{\partial x_j} \left(\frac{\delta x_k}{\delta b_n}\right) d\Omega \end{aligned} \tag{A.12}$$

for the FI, and

$$\begin{aligned} \left.\frac{\delta F}{\delta b_n}\right|_{E-SI} &= \int_{S_{W_p}} \left(p\delta_i^j - \tau_{ij}\right) n_j r_l \epsilon_{ikl} \frac{\delta x_k}{\delta b_n} dS + \int_{S_{W_p}} \left(p\delta_i^j - \tau_{ij}\right) \left(x_k - \overline{x}_k\right) r_l \epsilon_{ikl} \frac{\delta (n_j dS)}{\delta b_n} \\ &+ \int_{S_{W_p}} \left(\mathcal{A}_{jk}^L + \mathcal{A}_{jk}^T\right) n_j \frac{\delta x_k}{\delta b_n} dS - \int_{S_{W_p}} \frac{\partial m_k^\alpha}{\partial x_j} n_j \frac{\delta x_k}{\delta b_n} dS \end{aligned} \tag{A.13}$$

for the E-SI adjoint. With the farfield-flow angle and velocity magnitude as the uncertain variables, the sensitivities are equal to

$$\begin{aligned} \frac{\delta F}{\delta c_m} &= \int_{S_W} \left(p\delta_i^j - \tau_{ij}\right) n_j \left(x_k - \overline{x}_k\right) \epsilon_{ikl} \frac{\partial r_l}{\partial c_m} dS \\ &+ \int_{S_I} \left(-q n_i + v_j u_i n_j + \tau_{ij}^\alpha n_j + \widetilde{\nu}_\alpha \widetilde{\nu} \frac{C_\omega}{\omega} Y_q \epsilon_{qji} n_j\right) \frac{\delta v_i}{\delta c_m} dS \end{aligned} \tag{A.14}$$

where

$$\frac{\partial r_i}{\partial v_\infty} = -\frac{\widetilde{r}_i}{0.25 v_\infty^3 A_c l_c}, \quad \frac{\partial \boldsymbol{r}}{\partial \alpha_\infty} = 0 \tag{A.15}$$

The projected mixed derivatives matrix is equal to

$$
\begin{aligned}
z_m \frac{\delta^2 F}{\delta c_m \delta b_n}\bigg|_{FI} = & \int_{S_{W_p}} \left[\left(\widehat{p}\delta_i^j - \widehat{\tau}_{ij}\right) r_l + \left(p\delta_i^j - \tau_{ij}\right) \widehat{r}_l \right] n_j \epsilon_{ikl} \frac{\delta x_k}{\delta b_n} dS \\
& + \int_{S_{W_p}} \left[\left(\widehat{p}\delta_i^j - \widehat{\tau}_{ij}\right) r_l + \left(p\delta_i^j - \tau_{ij}\right) \widehat{r}_l \right] \left(x_k - \overline{x}_k\right) \epsilon_{ikl} \frac{\delta(n_j dS)}{\delta b_n} dS \\
& + \int_{\Omega} \left(\widehat{\mathcal{A}}_{jk}^L + \widehat{\mathcal{A}}_{jk}^T\right) \frac{\partial}{\partial x_j} \left(\frac{\delta x_k}{\delta b_n}\right) d\Omega
\end{aligned} \tag{A.16}
$$

for the FI adjoint and

$$
\begin{aligned}
z_m \frac{\delta^2 F}{\delta c_m \delta b_n}\bigg|_{E-SI} = & \int_{S_{W_p}} \left[\left(\widehat{p}\delta_i^j - \widehat{\tau}_{ij}\right) r_l + \left(p\delta_i^j - \tau_{ij}\right) \widehat{r}_l \right] n_j \epsilon_{ikl} \frac{\delta x_k}{\delta b_n} dS \\
& + \int_{S_{W_p}} \left[\left(\widehat{p}\delta_i^j - \widehat{\tau}_{ij}\right) r_l + \left(p\delta_i^j - \tau_{ij}\right) \widehat{r}_l \right] \left(x_k - \overline{x}_k\right) \epsilon_{ikl} \frac{\delta(n_j dS)}{\delta b_n} dS \\
& + \int_{S_{W_p}} \left(\widehat{\mathcal{A}}_{jk}^L + \widehat{\mathcal{A}}_{jk}^T\right) n_j \frac{\delta x_k}{\delta b_n} dS - \int_{S_{W_p}} \frac{\partial \widehat{m}_k^\alpha}{\partial x_j} n_j \frac{\delta x_k}{\delta b_n} dS
\end{aligned} \tag{A.17}
$$

for the E-SI adjoint.

A.3 Total Pressure Losses

The total pressure losses, expressing the loss of power in the flow as it moves from the inlets to the outlets of the flow domain, is equal to

$$
F = -\int_{S_I} \left(p + \frac{1}{2} v_k^2\right) v_j n_j dS - \int_{S_O} \left(p + \frac{1}{2} v_k^2\right) v_j n_j dS \tag{A.18}
$$

The adjoint BCs are

$$
\begin{aligned}
S_I &: u_{\langle n\rangle}=v_{\langle n\rangle},\quad u_{\langle t^l\rangle}=0,\quad \frac{\partial q}{\partial n}=0,\quad \widetilde{\nu_\alpha}=0,\quad \Delta_\alpha=0\\
S_O &: \frac{\partial u_{\langle n\rangle}}{\partial n}=0,\quad v_{\langle n\rangle}u_{\langle t^l\rangle}+\tau^\alpha_{ij}t^l_i n_j+\widetilde{\nu_\alpha}\widetilde{\nu}\frac{C_\omega}{\omega}Y_k\epsilon_{kji}n_j t^l_i-v_{\langle n\rangle}v_{\langle t^l\rangle}=0\quad l=I,II\\
&\quad -q+v_{\langle n\rangle}u_{\langle n\rangle}+\tau^\alpha_{ij}n_i n_j+\widetilde{\nu_\alpha}\widetilde{\nu}\frac{C_\omega}{\omega}Y_l\epsilon_{lji}n_j n_i-p-\frac{1}{2}v_k^2-v^2_{\langle n\rangle}=0,\\
&\quad \widetilde{\nu_\alpha}v_{\langle n\rangle}+\frac{\nu+\widetilde{\nu}}{c_\sigma}\frac{\partial\widetilde{\nu_\alpha}}{\partial n}=0,\quad \Delta_\alpha=0\\
S_W &: u_i=0,\quad \frac{\partial q}{\partial n}=0,\quad \widetilde{\nu_\alpha}=0,\quad \frac{\partial\Delta_\alpha}{\partial n}=0\\
S_S &: u_{\langle n\rangle}=0,\quad \frac{\partial u_{\langle t^l\rangle}}{\partial n}=0\quad l=I,II,\quad \frac{\partial q}{\partial n}=0,\quad \frac{\partial\widetilde{\nu_\alpha}}{\partial n}=0,\quad \frac{\partial\Delta_\alpha}{\partial n}=0
\end{aligned}
\tag{A.19}
$$

The SDs expressions become,

$$
\left.\frac{\delta F}{\delta b_n}\right|_{FI}=\int_\Omega\left(\mathcal{A}^L_{jk}+\mathcal{A}^T_{jk}\right)\frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right)d\Omega \tag{A.20}
$$

for the FI, and

$$
\left.\frac{\delta F}{\delta b_n}\right|_{E-SI}=\int_{S_{W_p}}\left(\mathcal{A}^L_{jk}+\mathcal{A}^T_{jk}\right)n_j\frac{\delta x_k}{\delta b_n}dS-\int_{S_{W_p}}\frac{\partial m^\alpha_k}{\partial x_j}n_j\frac{\delta x_k}{\delta b_n}dS \tag{A.21}
$$

for the E-SI adjoint. With the inlet velocity magnitude as the uncertain variable, the sensitivities are equal to

$$
\frac{\delta F}{\delta c_m}=\int_{S_I}\left(-qn_i+v_ju_in_j+\tau^\alpha_{ij}n_j+\widetilde{\nu_\alpha}\widetilde{\nu}\frac{C_\omega}{\omega}Y_q\epsilon_{qji}n_j-pn_i-\frac{1}{2}v_k^2n_i-v_{\langle n\rangle}v_i\right)\frac{\delta v_i}{\delta c_m}dS \tag{A.22}
$$

The projected mixed derivatives matrix is equal to

$$
z_m\left.\frac{\delta^2F}{\delta c_m\delta b_n}\right|_{FI}=\int_\Omega\left(\widehat{\mathcal{A}}^L_{jk}+\widehat{\mathcal{A}}^T_{jk}\right)\frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right)d\Omega \tag{A.23}
$$

for the FI adjoint and

$$
z_m\left.\frac{\delta^2F}{\delta c_m\delta b_n}\right|_{E-SI}=\int_{S_{W_p}}\left(\widehat{\mathcal{A}}^L_{jk}+\widehat{\mathcal{A}}^T_{jk}\right)n_j\frac{\delta x_k}{\delta b_n}dS-\int_{S_{W_p}}\frac{\partial\widehat{m}^\alpha_k}{\partial x_j}n_j\frac{\delta x_k}{\delta b_n}dS \tag{A.24}
$$

for the E-SI adjoint.

Bibliography

[1] ADflow software. https://github.com/mdolab/adflow.

[2] T. Albring, T. Dick, and N. Gauger. Assessment of the Recursive Projection Method for the stabilization of discrete adjoint solvers. In *Proceedings of the 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA 2017-3664, Denver, Colorado, 2017.

[3] T. A. Albring, M. Sagebaum, and N. R. Gauger. Development of a consistent discrete adjoint solver in an evolving aerodynamic design framework. In *Proceedings of the 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA 2015-3240, Dallas, Texas, 2015.

[4] A. K. Alekseev, I. M. Navon, and M. E. Zelentsov. The estimation of functional uncertainty using polynomial chaos and adjoint equations. *International Journal for Numerical Methods in Fluids*, 67(3):328–341, 2011.

[5] H. An, J. Wen, and T. Feng. On finite difference approximation of a matrix-vector product in the Jacobian-free Newton–Krylov method. *Journal of Computational and Applied Mathematics*, 236(6):1399–1409, 2011.

[6] W. K. Anderson, J. C. Newman, D. L. Whitfield, L. David, and E. J. Nielsen. Sensitivity analysis for Navier-Stokes equations on unstructured meshes using complex variables. *AIAA Journal*, 39(1):56–63, 2001.

[7] W. K. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Computers & Fluids*, 28(4):443–480, 1999.

[8] I. Antonau, S. Warnakulasuriya, K. U. Bletzinger, F. M. Bluhm, M. Hojjat, and R. Wüchner. Latest developments in node-based shape optimization using vertex morphing parameterization. *Structural and Multidisciplinary Optimization*, 65, 198, 2022.

[9] V. G. Asouti. *Aerodynamic Analysis and Design Methods at High and Low Speed Flows, on Multiprocessor Platforms*. PhD thesis, National Technical University of Athens, 2009.

[10] V. G. Asouti, A. S. Zymaris, D. I. Papadimitriou, and K. C. Giannakoglou. Continuous and discrete adjoint approaches for aerodynamic shape optimization with low Mach number preconditioning. *International Journal for Numerical Methods in Fluids*, 57(10):1485–1504, 2008.

[11] H.-G. Beyer and B. Sendhoff. Robust optimization – a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33):3190–3218, 2007.

[12] C. G. Broyden. The convergence of a class of double-rank minimization algorithms 1. General considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970.

[13] M. Campobasso and M. Giles. Stabilizing linear harmonic flow solvers for turbomachinery aeroelasicity with complex iterative algorithms. *AIAA Journal*, 44(5):1048–1059, 2006.

[14] A. Carnarius, F. Thiele, E. Oezkaya, and N. R. Gauger. Adjoint approaches for optimal flow control. In *Proceedings of the 5th Flow Control Conference*, AIAA 2010-5088, Chicago, Illinois, 2010.

[15] T. F. Chan. Rank revealing QR factorizations. *Linear Algebra and its Applications*, 88–89:67–82, 1987.

[16] T. F. C. Chan and H. B. Keller. Arc-length continuation and multigrid techniques for nonlinear elliptic eigenvalue problems. *SIAM Journal on Scientific and Statistical Computing*, 3(2):173–194, 1982.

[17] A. N. Christiansen, J. A. Bærentzen, M. Nobel-Jørgensen, N. Aage, and O. Sigmund. Combined shape and topology optimization of 3D structures. *Computers & Graphics*, 46:25–35, 2015.

[18] E. C. Cyr, J. N. Shadid, and T. Wildey. Towards efficient backward-in-time adjoint computations using data compression techniques. *Computer Methods in Applied Mechanics and Engineering*, 288:24–44, 2015.

[19] DAFoam software. https://github.com/mdolab/dafoam.

[20] S. C. R. Dennis and G. Chang. Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100. *Journal of Fluid Mechanics*, 42(3):471–489, 1970.

[21] M. Drela. Pros and cons of airfoil optimization. In D. A. Caughey and M. M. Hafez, editors, *Frontiers of Computational Fluid Dynamics*, pages 363–381. World Scientific: Hackensack, NJ, 1998.

[22] R. Dwight and D. Vollmer. Efficient algorithms for solution of the adjoint compressible Navier-Stokes equations with applications. In *Proceedings of the ONERA-DLR Aerospace Symposium (ODAS)*, Toulouse, France, 2006.

[23] R. P. Dwight and J. Brezillon. Effect of approximations of the discrete adjoint on gradient-based optimization. *AIAA Journal*, 44(12):3022–3031, 2006.

[24] J. Elliott and J. Peraire. Aerodynamic design using unstructured meshes. In *Proceedings of the Fluid Dynamics Conference*, AIAA 1996-1941, New Orleans, LA, 1996.

[25] J. Elliott and J. Peraire. Practical three-dimensional aerodynamic design and optimization using unstructured meshes. *AIAA Journal*, 35(9):1479–1485, 1997.

[26] G. Fishman. *Monte Carlo: Concepts, Algorithms and Applications*. Springer, New York, NY, 1st edition, 1996.

[27] B. Fleischli, L. Mangani, A. Del Rio, and E. Casartelli. A discrete adjoint method for pressure-based algorithms. *Computers & Fluids*, 227, 105037, 2021.

[28] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, 1964.

[29] B. Fornberg. Steady viscous flow past a circular cylinder up to Reynolds number 600. *Journal of Computational Physics*, 61(2):297–320, 1985.

[30] K. B. Fragkos, E. M. Papoutsis-Kiachagias, and K. C. Giannakoglou. pFOSM: an efficient algorithm for aerodynamic robust design based on continuous adjoint and matrix-vector products. *Computers and Fluids*, 181:57–66, 2019.

[31] C. Georgopoulou. *Optimization Techniques for Commiting Combined Cycle Power Plants and Designing their Components*. PhD thesis, National Technical University of Athens, 2009.

[32] T. Ghisu, D. I. Lopez, P. Seshadri, and S. Shahpar. Gradient-enhanced least-square polynomial chaos expansions for uncertainty quantification and robust optimization. In *Proceedings of the AIAA Aviation 2021 Forum*, AIAA 2021-3073, Virtual Event, 2021.

[33] F. Gianetti and P. Luchini. Structural sensitivity of the first instability of the cylinder wake. *Journal of Fluid Mechanics*, 581:167–197, 2007.

[34] M. Giles, M. Duta, and J.-D. Müller. Adjoint code developments using the exact discrete approach. In *Proceedings of the 15th AIAA Computational Fluid Dynamics Conference*, AIAA 2001-2596, Reston, Virigina, 2001.

[35] A. Giotis. *Application of Evolutionary Algorithms, Computational Intelligence and Advanced Computational Fluid Dynamics Techniques to the Optimization-Inverse Design of Turbomachinery Cascades, using Parallel Processing*. PhD thesis, National Technical University of Athens, 2003.

[36] K. Gkaragkounis. *The Continuous Adjoint Method in Aerodynamic and Conjugate Heat Transfer Shape Optimization, for Turbulent Flows*. PhD thesis, National Technical University of Athens, 2020.

[37] K. T. Gkaragkounis, E. M. Papoutsis-Kiachagias, and K. C. Giannakoglou. The continuous adjoint method for shape optimization in conjugate heat transfer problems with turbulent incompressible flows. *Applied Thermal Engineering*, 140:351–362, 2018.

[38] K. T. Gkaragkounis, E. M. Papoutsis-Kiachagias, and K. C. Giannakoglou. Adjoint-assisted Pareto front tracing in aerodynamic and conjugate heat transfer shape optimization. *Computers & Fluids*, 214, 104753, 2021.

[39] K. T. Gkaragkounis, E. M. Papoutsis-Kiachagias, A. G. Tsolovikos, and K. C. Giannakoglou. Effect of grid displacement models on sensitivity derivatives computed by the continuous adjoint method in aerodynamic and conjugate heat transfer shape optimization. *Engineering Optimization*, 53(8):1398–1416, 2021.

[40] G. H. Golub and C. F. van Loan. *Matrix Computations*. JHU Press, Baltimore, 4th edition, 2013.

[41] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan, J. Zhang, Y. Li, Q. Zhang, and J.-J. Li. Distributed evolutionary algorithms and their models: A survey of the state-of-the-art. *Applied Soft Computing*, 34:286–300, 2015.

[42] S. Gortz and J. Moller. Evaluation of the Recursive Projection Method for efficient unsteady turbulent CFD simulations. In *Proceedings of the 24th International Congress of the Aeronautical Sciences*, Yokohama, Japan, 2004.

[43] A. Griewank. *Evaluating Derivatives*. Society for Industrial and Applied Mathematics, Philadelfia, 2000.

[44] A. Griewank and A. Walther. Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software*, 26(1):19–45, 2000.

[45] M. Hadigol and A. Doostan. Least squares polynomial chaos expansion: a review of sampling strategies. *Computer Methods in Applied Mechanics and Engineering*, 332:382–407, 2018.

[46] R. T. Haftka. Sensitivity calculations for iteratively solved problems. *International Journal for Numerical Methods in Engineering*, 21(8):1535–1546, 1985.

[47] A. Haldar and S. Mahadevan. *Probability, Reliability and Statistical Methods in Engineering Design*. John Wiley & Sons, New York/Chichester, UK, 1st edition, 2000, pages 195–218.

[48] P. He, C.A. Mader, J.R.R.A. Martins, and K.J. Maki. An aerodynamic design optimization framework using a discrete adjoint approach with OpenFOAM. *Computers & Fluids*, 168:285–303, 2018.

[49] J.E. Hicken. Aerodynamic Design Optimization Workshop: Twist Optimization Case. Rensselaer Polytechnic Institute, Dept. of Mechanical, Aerospace, and Nuclear Eng., Troy, NY, United States, 12180. https://sites.google.com/view/mcgill-computational-aerogroup/adodg.

[50] S. Hosder, R. Walters, and M. Balch. Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables. In *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA 2007-1939, Honolulu, HI, 2007.

[51] S. Hosder, R. Walters, and R. Perez. A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations. In *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2006-891, Reno, Nevada, 2006.

[52] Z. Huan, G. Zhenghong, X. Fang, and Z. Yidian. Review of robust aerodynamic design optimization for air vehicles. *Archives of Computational Methods in Engineering*, 26(3):685–732, 2019.

[53] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.

[54] A. Jameson, L. Martinelli, and N. Pierce. Optimum aerodynamic design using the Navier-Stokes equations. *Thoretical and Computational Fluid Dynamics*, 10:213–237, 1998.

[55] A. Jameson and J. Reuther. Control theory based airfoil design using the Euler equations. In *Proceedings of the 5th Symposium on Multidisciplinary Analysis and Optimization*, AIAA 1994-4272, Panama City Beach, FL, USA, 1994.

[56] A. Jameson and S. Kim. Reduction of the adjoint gradient formula in the continuous limit. In *Proceedings of the 41st Aerospace Sciences Meeting and Exhibit*, AIAA 2003-40, Reno, Nevada, 2003.

[57] A. Jameson, S. Shankaran, and L. Martinelli. Continuous adjoint method for unstructured grids. *AIAA Journal*, 46(5):1226–1239, 2008.

[58] H. Jarausch and W. Mackens. Numerical treatment of bifurcation branches by adaptive condensation. *Numerical Methods for Bifurcation Problems. International Series of Numerical Mathematics*, 70, 1984.

[59] A. Jemcov, J. Maruszewski, and H. Jasak. Acceleration and stabilization of algebraic multigrid solver applied to incompressible flow problems. *18th AIAA Computational Fluid Dynamics Conference, Miami, Florida*, 2007.

[60] I. Kampolis. *Parallel, Multilevel Algorithms for the Aerodynamic Optimization in Turbomachines*. PhD thesis, National Technical University of Athens, 2009.

[61] D. Kapsoulis. *Low-Cost Metamodel-Assisted Evolutionary Algorithms with Applications in Shape Optimization in Fluid Dynamics*. PhD thesis, National Technical University of Athens, 2019.

[62] M. Karakasis. *Hierarchical Distributed Evolutionary Algorithms and Computational Intelligence in Aerodynamic Shape Optimization, on Multiprocessing Systems*. PhD thesis, National Technical University of Athens, 2006.

[63] G. Karpouzas, E. M. Papoutsis-Kiachagias, T. Schumacher, E. de Villiers, K. C. Giannakoglou, and C. Othmer. Adjoint optimization for vehicle external aerodynamics. *International Journal of Automotive Engineering*, 7(1):1–7, 2016.

[64] I. S. Kavvadias. *Continuous Adjoint Methods for Steady and Unsteady Turbulent flows with Emphasis on the Accuracy of Sensitivity Derivatives*. PhD thesis, National Technical University of Athens, 2015.

[65] I. S. Kavvadias, E. M. Papoutsis-Kiachagias, G. Dimitrakopoulos, and K. C. Giannakoglou. The continuous adjoint approach to the $k-\omega$ SST turbulence model with applications in shape optimization. *Engineering Optimization*, 47(11):1523–1542, 2015.

[66] 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.

[67] G. Kenway, G. Kennedy, and J. R. R. A. Martins. A CAD-free approach to high-fidelity aerostructural optimization. In *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, AIAA 2010-9231, Fort Worth, Texas, United States, 2010.

[68] G. Kenway, C. Mader, P. He, and J. R. R. A. Martins. CFD discrete adjoint benchmarks, 2019. url: http://dx.doi.org/10.17632/w4hsj8wzm8.2. Mendeley Data, V2.

[69] G. Kenway and J. R. R. A. Martins. Aerodynamic shape optimization of the CRM configuration including buffet-onset conditions. In *Proceedings of the 54th AIAA Aerospace Sciences Meeting*, AIAA 2016-1294, San Diego, California, USA, 2016.

[70] G. K. W. Kenway, C. A. Mader, P. He, and J. R. R. A. Martins. Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences*, 110, 100542, 2019.

[71] E. Kontoleontos. *Designing Thermo-Fluid Systems using Gradient-based Optimization Methods and Evolutionary Algorithms*. PhD thesis, National Technical University of Athens, 2012.

[72] E. A. Kontoleontos, E. M. Papoutsis-Kiachagias, A. S. Zymaris, D. I. Papadimitriou, and K. C. Giannakoglou. Adjoint-based constrained topology optimization for viscous flows, including heat transfer. *Engineering Optimization*, 45(8):941–961, 2013.

[73] E. D. Koronaki, A. G. Boudouvis, and I. G. Kevrekidis. Enabling stability analysis of tubular reactor models using PDE/PDAE integrators. *Computers & Chemical Engineering*, 27(7):951–964, 2003.

[74] J. A. Krakos and D. L. Darmofal. Effect of small-scale output unsteadiness on adjoint-based sensitivity. *AIAA Journal*, 48:2611–2623, 2010.

[75] J. Kröger, N. Kühl, and T. Rung. Adjoint volume-of-fluid approaches for the hydrodynamic optimisation of ships. *Ship Technology Research*, 65(1):47–68, 2018.

[76] D. Kurtulus. On the unsteady behavior of the flow around NACA0012 airfoil with steady external conditions at Re=1000. *International Journal of Micro Air Vehicles*, 7:301–326, 2015.

[77] S. Kyriacou. *Evolutionary Algorithm-based Design-Optimization Methods in Turbomachinery*. PhD thesis, National Technical University of Athens, 2013.

[78] Y. Liu, K. Li, J. Zhang, H. Wang, and L. Liu. Numerical bifurcation analysis of static stall of airfoil and dynamic stall under unsteady perturbation. *Communications in Nonlinear Science and Numerical Simulation*, 17:3427–3434, 2012.

[79] P. Love. *Bifurcations in Kolmogorov and Taylor-Vortex Flows*. PhD thesis, California Institute of Technology, 1999.

[80] C. Lozano. Discrete surprises in the computation of sensitivities from boundary integrals in the continuous adjoint approach to inviscid aerodynamic shape optimization. *Computers & Fluids*, 56:118–127, 2012.

[81] C. Lozano, E. Andrés, M. Martín, and P. Bitrián. Domain versus boundary computation of flow sensitivities with the continuous adjoint method for aerodynamic shape optimization problems. *International Journal for Numerical Methods in Fluids*, 70(10):1305–1323, 2012.

[82] H. Luc. Free-form airfoil shape optimization under uncertainty using maximum expected value and second-order second-moment strategies. Technical report ICASE, 2001-18/NASA CR 2001-211020, NASA Langley Research Center, Hampton, VA, 2001.

[83] K. Lust. *Numerical Bifurcation Analysis of Periodic Solutions of Partial Differential Equations*. PhD thesis, Department of Computer Science, K. U. Leuven, 1997.

[84] K. Mani and D. J. Mavriplis. Unsteady discrete adjoint formulation for two-dimensional flow problems with deforming meshes. *AIAA Journal*, 46(6):1351–1364, 2008.

[85] K. Mani and D. J. Mavriplis. Adjoint-based sensitivity formulation for fully coupled unsteady aeroelasticity problems. *AIAA Journal*, 47(8):1902–1915, 2009.

[86] A. I. Margetis, E. M. Papoutsis-Kiachagias, and K. C. Giannakoglou. Lossy compression techniques supporting unsteady adjoint on 2D/3D unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 387, 114152, 2021.

[87] A. I. Margetis, E. M. Papoutsis-Kiachagias, and K. C. Giannakoglou. Reducing memory requirements of unsteady adjoint by synergistically using check-pointing and compression. *Numerical Methods in Fluids*:1–21, 2022.

[88] J. M. Martin, E. Andres, C. Lozano, and E. Valero. Volumetric B-Splines shape parametrization for aerodynamic shape design. *Aerospace Science and Technology*, 37:26–36, 2014.

[89] J. R. R. A. Martins and J. T. Hwang. Review and unification of methods for computing derivatives of multidisciplinary computational models. *AIAA Journal*, 51(11):2582–2599, 2013.

[90] D. Mavriplis. Solution of the unsteady discrete adjoint for three-dimensional problems on dynamically deforming unstructured meshes. In *Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2008-0727, Reno, NV, 2008.

[91] M. McKay, R. Beckman, and W. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[92] M. Monfaredi. *Continuous Adjoint-based Aeroacoustic Shape Optimization by Coupling the Ffowcs-Williams Hawkings Analogy with URANS equations*. PhD thesis, National Technical University of Athens, 2021.

[93] M. Monfaredi, X. Trompoukis, K. Tsiakas, and K. Giannakoglou. Unsteady continuous adjoint to URANS coupled with FW-H analogy for aeroacoustic shape optimization. *Computers & Fluids*, 230, 105136, 2021.

[94] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics*. Springer International Publishing, Cham, 1st edition, 2016.

[95] S. Nadarajah and A. Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. In *Proceedings of the 38th Aerospace Sciences Meeting and Exhibit*, AIAA 2000-667, Reno, NV, 2000.

[96] S. Nadarajah and A. Jameson. Optimal control of unsteady flows using a time accurate method. In *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA 2002–5436, Atlanta, Georgia, 2002.

[97] S. Nadarajah, M. McMullen, and A. Jameson. Optimal control of unsteady flows using time accurate and non-linear frequency domain methods. In *Proceedings of the 33rd AIAA Fluid Dynamics Conference and Exhibit*, AIAA 2003-3875, Orlando, Florida, 2003.

[98] S. K. Nadarajah and A. Jameson. Optimum shape design for unsteady three-dimensional viscous flows using a nonlinear frequency-domain method. *Journal of Aircraft*, 44(5):1513–1527, 2007.

[99] H. Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 41(1):35–52, 2009.

[100] U. Naumann. *The Art of Differentiating Computer Programs*. Society for Industrial and Applied Mathematics, Philadelfia, PA, 2011.

[101] T. T. Nguyen, J. A. Bærentzen, and O. Sigmund. Efficient hybrid topology and shape optimization combining implicit and explicit design representations. *Structural and Multidisciplinary Optimization*, 62:1061–1069, 2020.

[102] E. J. Nielsen and W. L. Kleb. Efficient construction of discrete adjoint operators on unstructured grids using complex variables. *AIAA Journal*, 44(4):827–836, 2006.

[103] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, NY, 1st edition, 1999.

[104] S. Oladyshkin and W. Nowak. Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliability Engineering & System Safety*, 106:179–190, 2012.

[105] OpenFOAM software. www.openfoam.com.

[106] C. Othmer. Adjoint methods for car aerodynamics. *Journal of Mathematics in Industry*, 4(6), 2014.

[107] D. I. Papadimitriou. *Adjoint Formulations for the Analysis and Design of Turbomachinery Cascades and Optimal Grid Adaptation using Aposteriori Error Analysis*. PhD thesis, National Technical University of Athens, 2007.

[108] 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.

[109] D. I. Papadimitriou and K. C. Giannakoglou. Aerodynamic shape optimization using first and second order adjoint and direct approaches. *Archives of Computational Methods in Engineering*, 15(4):447–488, 2008.

[110] D. I. Papadimitriou and K. C. Giannakoglou. Computation of the Hessian matrix in aerodynamic inverse design using continuous adjoint formulations. *Computers & Fluids*, 37(8):1029–1039, 2008.

[111] D. I. Papadimitriou and K. C. Giannakoglou. Direct, adjoint and mixed approaches for the computation of Hessian in airfoil design problems. *International Journal for Numerical Methods in Fluids*, 56(10):1929–1943, 2008.

[112] D. I. Papadimitriou and K. C. Giannakoglou. The continuous direct-adjoint approach for second order sensitivities in viscous aerodynamic inverse design problems. *Computers & Fluids*, 38(8):1539–1548, 2009.

[113] A. K. Papageorgiou, E. M. Papoutsis-Kiachagias, and K. C. Giannakoglou. Development and assessment of an intrusive polynomial chaos expansion-based continuous adjoint method for shape optimization under uncertainties. *International Journal for Numerical Methods in Fluids*, 94(1):59–75, 2022.

[114] E. M. Papoutsis-Kiachagias. *Adjoint methods for turbulent flows, applied to shape or topology optimization and robust design*. PhD thesis, National Technical University of Athens, 2010.

[115] E. M. Papoutsis-Kiachagias, V. G. Asouti, K. C. Giannakoglou, K. Gkagkas, S. Shimokawa, and E. Itakura. Multi-point aerodynamic shape optimization of cars based on continuous adjoint. *Structural and Multidisciplinary Optimization*, 59(2):675–694, 2019.

[116] E. M. Papoutsis-Kiachagias and K. C. Giannakoglou. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: industrial applications. *Archives in Computational Methods in Engineering*, 32(2):255–299, 2016.

[117] E. M. Papoutsis-Kiachagias, S. Kyriacou, and K. C. Giannakoglou. The continuous adjoint method for the design of hydraulic turbomachines. *Computer Methods in Applied Mechanics and Engineering*, 278:621–639, 2014.

[118] E. M. Papoutsis-Kiachagias, N. Magoulas, J. Mueller, C. Othmer, and K. C. Giannakoglou. Noise reduction in car aerodynamics using a surrogate objective function and the continuous adjoint method with wall functions. *Computers & Fluids*, 122:223–232, 2015.

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

[120] E. M. Papoutsis-Kiachagias, A. S. Zymaris, I. S. Kavvadias, D. I. Papadimitriou, and K. C. Giannakoglou. The continuous adjoint approach to the $k - \epsilon$ turbulence model for shape optimization and optimal active control of turbulent flows. *Engineering Optimization*, 47(3):370–389, 2015.

[121] J. Park, K. Kwon, and H. Choi. Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160. *KSME International Journal*, 12(6):1200–1205, 1998.

[122] S. V. Patankar and D. B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15(10):1787–1806, 1972.

[123] J. E. V. Peter and R. P. Dwight. Numerical sensitivity analysis for aerodynamic optimization: a survey of approaches. *Computers & Fluids*, 39(3):373–391, 2010.

[124] L. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, New York, 2nd edition, 1997.

[125] O. Pironneau. On optimum profiles in stokes flow. *Journal of Fluid Mechanics*, 59(1):117–128, 1973.

[126] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(1):97–110, 1974.

[127] J. Reuther, J. Alonso, M. Rimlinger, and A. Jameson. Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers. *Computers & Fluids*, 28(4):675–700, 1999.

[128] J. Reuther and A. Jameson. Control theory based airfoil design for potential flow and a finite volume discretization. In *Proceedings of the 32nd Aerospace Sciences Meeting and Exhibit*, AIAA 1994-499, Reno, Nevada, 1994.

[129] J. Reuther and A. Jameson. Aerodynamic shape optimization of wing and wing-body configurations using control theory. In *Proceedings of the 33rd Aerospace Sciences Meeting and Exhibit*, AIAA 1995-123, Reno, Nevada, 1995.

[130] J. Reuther, A. Jameson, J. Farmer, L. Martinelli, and D. Saunders. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. In *Proceedings of the 34th Aerospace Sciences Meeting and Exhibit*, AIAA 1996-94, Reno, Nevada, 1996.

[131] C. M. Rhie and W. L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21(11):1525–1532, 1983.

[132] O. Roderick, M. Anitescu, and P. Fischer. Polynomial regression approaches using derivative information for uncertainty quantification. *Nuclear Science and Engineering*, 164(2):122–139, 2010.

[133] J. B. Rosen. The gradient projection method for nonlinear programming. Part I. Linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):181–217, 1960.

[134] J. B. Rosen. The gradient projection method for nonlinear programming. Part II. Nonlinear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):514–532, 1961.

[135] R. Roth and S. Ulbrich. A discrete adjoint approach for the optimization of unsteady turbulent flows. *Flow, Turbulence and Combustion*, 90(4), 2013.

[136] M. Rumpfkeil and D. Zingg. A general framework for the optimal control of unsteady flows with applications. In *Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2007–1128, Reno, NV, 2007.

[137] A. J. Samareh. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA Journal*, 39(5):877–884, 2001.

[138] K. Samouchos. *The Cut-Cell Method for the Prediction of 2D/3D Flows in Complex Geometries and the Adjoint-Based Shape Optimization*. PhD thesis, National Technical University of Athens, 2022.

[139] C. Schillings, S. Schmidt, and V. Schulz. Efficient shape optimization for certain and uncertain aerodynamic design. *Computers & Fluids*, 46(1):78–87, 2011.

[140] G. I. Schuëller and H. A. Jensen. Computational methods in optimization considering uncertainties - An overview. *Computer Methods in Applied Mechanics and Engineering*, 198(1):2–13, 2008.

[141] H. Shah, S. Hosder, S. Koziel, Y. A. Tesfahunegn, and L. Leifsson. Multi-fidelity robust aerodynamic design optimization under mixed uncertainty. *Aerospace Science and Technology*, 45:17–29, 2015.

[142] S. Shankaran and A. Jameson. Robust optimal control using polynomial chaos and adjoints for systems with uncertain inputs. In *Proceedings of the 20th AIAA Computational Fluid Dynamics Conference*, AIAA 2011-3069, Honolulu, Hawaii, 2011.

[143] G. M. Shroff and H. Keller. Stabilization of unstable procedures: the Recursive Projection Method. *SIAM Journal of Numerical Analysis*, 30(4):1099–1120, 1993.

[144] H. Sobieczky. Parametric airfoils and wings. In K. Fujii and G. S. Dulikravich, editors, *Recent Development of Aerodynamic Design Methodologies. Notes on Numerical Fluid Mechanics (NNFM)*. Volume 65, pages 71–88. Vieweg+Teubner Verlag, 1998.

[145] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *Proceedings of the 30th Aerospace Sciences Meeting and Exhibit*, AIAA Paper 92-0439, Reno, Nevada, USA, 1992.

[146] D. Sudholt. Parallel evolutionary algorithms. In J. Kacprzyk and W. Pedrycz, editors, *Springer Handbook of Computational Intelligence. Springer Handbooks*, pages 929–959. Springer, Berlin, Heidelberg, 2015.

[147] H. Takami and H. B. Keller. Steady two-dimensional viscous flow of an incompressible fluid past a circular cylinder. *The Physics of Fluids*, 12(12):II–51–II–56, 1969.

[148] M. Thapa, S. B. Mulani, and R. W. Walters. Adaptive weighted least-squares polynomial chaos expansion with basis adaptivity and sequential adaptive sampling. *Computer Methods in Applied Mechanics and Engineering*, 360, 112759, 2020.

[149] K. Tsiakas. *Development of shape parameterization techniques, a flow solver and its adjoint, for optimization on GPUs. Turbomachinery and external aerodynamics applications*. PhD thesis, National Technical University of Athens, 2019.

[150] P. G. Tucker. Differential equation-based wall distance computation for DES and RANS. *Journal of Computational Physics*, 190:229–248, 2003.

[151] 'UIUC Airfoil Data Site'. University of Illinois, Department of Aerospace Engineering. https://m-selig.ae.illinois.edu/ads/coord_database.html. (accessed: 21.09.2021).

[152] G. Venter and J. Sobieszczanski-Sobieski. Particle swarm optimization. In *Proceedings of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA 2002-1235, pages 1–9, Denver, Colorado, 2002.

[153] C. Vezyris, E. M. Papoutsis-Kiachagias, and K. C. Giannakoglou. On the incremental singular value decomposition method to support unsteady adjoint based optimization. *Numerical Methods in Fluids*, 91(7):315–331, 2019.

[154] H. von Sosen. *Part II: The Recursive Projection Method applied to differential-algebraic equations and incompressible fluid mechanics*. PhD thesis, California Institue of Technology, Pasadena, California, 1994.

[155] P. Y. Vrionis, K. D. Samouchos, and K. C. Giannakoglou. The continuous adjoint cut-cell method for shape optimization in cavitating flows. *Computers & Fluids*, 224, 104974, 2021.

[156] P. Y. Vrionis, K. D. Samouchos, and K. C. Giannakoglou. Topology optimization in fluid mechanics using continuous adjoint and the cut-cell method. *Computers & Mathematics with Applications*, 97:286–297, 2021.

[157] P. Y. Vrionis. *Shape and topology optimization using the cut-cell method and its continuous adjoint for single- and two- phase turbulent flows, in a multiprocessor environment*. PhD thesis, National Technical University of Athens, 2022.

[158] R. Walters and L. Huyse. Uncertainty Analysis for Fluid Mechanics with Applications. Technical report ICASE 2002-1, NASA/CR-2002-211449, NASA Langley Research Center, Hampton, VA, 2002.

[159] A. Walther and A. Griewank. Advantages of binomial checkpointing for memory-reduced adjoint calculations. In M. Feistauer, V. Dolejší, P. Knobloch, and K. Najzar, editors, *Numerical Mathematics and Advanced Applications*, pages 834–843. Springer, Berlin, Heidelberg, 2004.

[160] S. Walton, O. Hassan, and K. Morgan. Reduced order modelling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions. *Applied Mathematical Modelling*, 37(20):8930–8945, 2013.

[161] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 187(1):137–167, 1938.

[162] J. H. Wilkinson and C. Reinsch. *Handbook for Automatic Computation: Volume II: Linear Algebra*. Springer-Verlag Berlin Heidelberg, New York, 1st edition, 1971.

[163] D. Xiu and G. Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187(1):137–167, 2003.

[164] L. Yang and S. Nadarajah. Data compression algorithms for adjoint based sensitivity studies of unsteady flows. In *Proceedings of the ASME 2018 5th Joint US-European Fluids Engineering Division Summer Meeting. Volume 2: Development and Applications in Computational Fluid Dynamics; Industrial and Environmental Applications of Fluid Mechanics; Fluid Measurement and Instrumentation; Cavitation and Phase Change*, Montreal, Quebec, Canada, 2018.

[165] T. Zervogiannis. *Optimization Methods in Aerodynamics and Turbomachinery based on the Adjoint Technique, Hybrid Grids and the Exact Hessian Matrix*. PhD thesis, National Technical University of Athens, 2011.

[166] T. Zervogiannis, D. I. Papadimitriou, and K. C. Giannakoglou. Total pressure losses minimization in turbomachinery cascades using the exact Hessian. *Computer Methods in Applied Mechanics and Engineering*, 199(41):2697–2708, 2010.

[167] H. Zhao, Z. Gao, Y. Gao, and C. Wang. Effective robust design of high lift NLF airfoil under multi-parameter uncertainty. *Aerospace Science and Technology*, 68:530–542, 2017.

[168] A. S. Zymaris. *Adjoint methods for the design of shapes with optimal aerodynamic performance in laminar and turbulent flows*. PhD thesis, National Technical University of Athens, 2010.

[169] A. S. Zymaris, D. I. Papadimitriou, K. C. Giannakoglou, and C. Othmer. Continuous adjoint approach to the Spalart-Allmaras turbulence model for incompressible flows. *Computers & Fluids*, 38(8):1528–1538, 2009.

[170] A. S. Zymaris, D. I. Papadimitriou, K. C. Giannakoglou, and C. Othmer. Adjoint wall functions: A new concept for use in aerodynamic shape optimization. *Journal of Computational Physics*, 229(13):5228–5245, 2010.

[171] A. S. Zymaris, D. I. Papadimitriou, E. M. Papoutsis-Kiachagias, K. C. Giannakoglou, and C. Othmer. The continuous adjoint method as a guide for the design of flow control systems based on jets. *Engineering Computations: International Journal for Computer-Aided Engineering and Software*, 30(4):494–520, 2013.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστομηχανικής & Βελτιστοποίησης

Μείωση Υπολογιστικού Κόστους και Σταθεροποίηση των Επιλυτών της Συνεχούς Συζυγούς Μεθόδου στη Βελτιστοποίηση Μορφής στην Αεροδυναμική, με η χωρίς Αβεβαιότητες

Διδακτορική Διατριβή

Θεμιστοκλής Σκαμάγκης

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου,
Καθηγητής ΕΜΠ

Αθήνα, 2023

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστομηχανικής & Βελτιστοποίησης

Μείωση Υπολογιστικού Κόστους και Σταθεροποίηση των Επιλυτών της Συνεχούς Συζυγούς Μεθόδου στη Βελτιστοποίηση Μορφής στην Αεροδυναμική, με η χωρίς Αβεβαιότητες

Διδακτορική Διατριβή

Θεμιστοκλής Σκαμάγκης

Εξεταστική Επιτροπή:

1. Κ. Γιαννάκογλου (Επιβλέπων)*,
   Καθηγητής, ΕΜΠ, Σχολή Μηχανολόγων Μηχανικών
2. Κ. Μαθιουδάκης*,
   Καθηγητής, ΕΜΠ, Σχολή Μηχανολόγων Μηχανικών
3. Β. Ριζιώτης*,
   Αναπληρωτής Καθηγητής, ΕΜΠ, Σχολή Μηχανολόγων Μηχανικών
4. Σ. Βουτσινάς,
   Καθηγητής, ΕΜΠ, Σχολή Μηχανολόγων Μηχανικών
5. Δ. Μπούρης,
   Καθηγητής, ΕΜΠ, Σχολή Μηχανολόγων Μηχανικών
6. Ι. Νικολός,
   Καθηγητής, ΠΚ, Σχολή Μηχανικών Παραγωγής και Διοίκησης
7. Γ. Παπαδάκης,
   Επίκουρος Καθηγητής, ΕΜΠ, Σχολή Ναυπηγών Μηχανολόγων Μηχανικών

* Μέλος της Συμβουλευτικής Επιτροπής.

Αθήνα, 2023

Περίληψη

Στη διδακτορική διατριβή αναπτύσσονται τεχνικές με σκοπό τη μείωση του υπολογιστικού κόστους αιτιοκρατικών μεθόδων βελτιστοποίησης μορφής στην αεροδυναμική, για ροές με ήπια χρονική αστάθεια ή/και με αβεβαιότητες. Όλες οι τεχνικές βασίζονται στη συνεχή συζυγή μέθοδο και αναπτύχθηκαν σε περιβάλλον OpenFOAM, συμπληρωματικά του επιλύτη συζυγών εξισώσεων ανοιχτής πρόσβασης ο οποίος έχει αναπτυχθεί από τη Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης του ΕΜΠ. Αρχικά, γίνεται πιστοποίηση των τεχνικών σε ακαδημαϊκές και ακολουθεί η χρήση τους σε βιομηχανικές εφαρμογές.

Η συζυγής μέθοδος για χρονικά μη-μόνιμους επιλύτες έχει μεγάλες απαιτήσεις σε υπολογιστικό κόστος ή/και αποθήκευση δεδομένων. Για αυτό και δεν χρησιμοποιείται συχνά στη βελτιστοποίηση μορφής σε βιομηχανικές εφαρμογές. Οι μόνιμοι επιλύτες πρωτευουσών και συζυγών εξισώσεων μπορούν, κατά περίπτωση, να χρησιμοποιηθούν όταν η ροή παρουσιάζει ήπια χρονική αστάθεια. Στη διατριβή αυτή εφαρμόζεται η Μέθοδος Αναδρομικών Προβολών (Recursive Projection Method, RPM) για την αντιμετώπιση των δυσκολιών σύγκλισης που παρουσιάζουν οι χρονικά μόνιμοι πρωτεύοντες και (συνεχείς) συζυγείς επιλύτες λόγω της χρονικής αστάθειας της ροής. Με το πέρας της βελτιστοποίησης με μόνιμους επιλύτες, οι βελτιστοποιημένες λύσεις επαναξιολογούνται από μη-μόνιμους επιλύτες πρωτεύοντος προβλήματος για να επιβεβαιωθεί η μείωση του χρονικού μέσου των συναρτήσεων στόχου. Έτσι, η βελτιστοποίηση γίνεται με υπολογιστικό κόστος ως και κατά μία τάξη μεγέθους χαμηλότερο από ότι με χρονικά μη-μόνιμους επιλύτες. Η τεχνική αυτή χρησιμοποιείται επίσης για βελτιστοποίηση μορφής σε ροές όπου εκλύονται στρόβιλοι. Παρότι μπορούν να υπολογιστούν βελτιωμένες λύσεις για τέτοιου είδους ροές με μόνιμους επιλύτες, δεν μπορεί εν τέλει να αποφευχθεί ένας βρόχος βελτιστοποίησης με μη-μόνιμους επιλύτες. Το συνολικό κόστος, όμως, μειώνεται χρησιμοποιώντας τη βελτιστοποιημένη λύση με μόνιμους επιλύτες ως αρχικοποίηση για τη βελτιστοποίηση με μη-μόνιμους.

Σε πιο πολύπλοκες εφαρμογές, οι δυσκολίες σύγκλισης των μόνιμων πρωτεύοντων και συζυγών επιλυτών, ακόμα και με την RPM, είναι πολύ πιο έντονες. Αιτία για αυτό μπορεί να είναι η ποιότητα του πλέγματος γύρω από περίπλοκες γεωμετρίες ή η ύπαρξη πληθώρας ιδιοσυχνοτήτων που προκαλούν τη γρήγορη απόκλιση του συζυγούς επιλύτη. Προκειμένου να αποφευχθούν τέτοια προβλήματα, χρησιμοποιείται (επιπλέον της RPM) ελεγχόμενη απόσβεση της Συζυγούς Ανάστροφης Συμμεταφοράς (Adjoint Transposed-Convection, ATC), ενός όρου που εμφανίζεται στις συνεχείς συζυγείς εξισώσεις ορμής. Η απόσβεση ή απαλοιφή αυτού του όρου χρησιμοποιείται συχνά στη βιβλιογραφία για τη σταθεροποίηση συνεχών συζυγών επιλυτών. Μελετάται, εδώ, η επίδραση που έχει αυτή η πρακτική στην ακρίβεια των παραγώγων ευαισθησίας σε περιπτώσεις όπου η RPM επιτυγχάνει από μόνη της τη σταθεροποίηση του συζυγούς επιλύτη. Η ελεγχόμενη απόσβεση της ATC (εφόσον απαιτείται) αποδεικνύεται πρακτικά

αβλαβής και συνεισφέρει σημαντικά στην ευστάθεια των συζυγών επιλυτών. Αντίθετα, η ανεξέλεγκτη απόσβεση του όρου απουσία της RPM είναι αρκετά συχνά επιζήμια. Επίσης, χάριν της RPM, αποφεύγεται η επίλυση συζυγών εξισώσεων βασισμένων στη μέση τιμή των μη-συγκλιμένων πρωτευόντων πεδίων.

Το δεύτερο σκέλος της διατριβής αφορά την αεροδυναμική βελτιστοποίηση μορφής υπό αβεβαιότητες. Σε τέτοια προβλήματα, προσδιορίζεται ποσοτικά η απόκριση μιας Ποσότητας Ενδιαφέροντος (Quantity of Interest, QoI) ως προς ένα σύνολο μεταβλητών αβεβαιότητας με τις λεγόμενες μεθόδους Ποσοτικοποίησης Αβεβαιότητας (Uncertainty Quantification, UQ). Αυτό αυξάνει σημαντικά το υπολογιστικό κόστος ενός βρόχου βελτιστοποίησης. Χρησιμοποιούνται οι μέθοδοι Πρώτης-Τάξης Δεύτερης-Ροπής (First-Order Second-Moment, FOSM) και Αναπτύγματος Πολυωνυμικού Χάους υποβοηθούμενου από τη Συζυγή Μέθοδο (Adjoint-assisted Polynomial Chaos Expansion, APCE) για τον υπολογισμό των δύο πρώτων στατιστικών ροπών (μέσης τιμής και τυπικής απόκλισης) μιας QoI. Το σταθμισμένο άθροισμα αυτών των ροπών χρησιμοποιείται ως συνάρτηση στόχος. Για μια τέτοια συνάρτηση, η βελτιστοποίηση με παραγώγους ευαισθησίας απαιτεί τις μικτές παραγώγους της QoI ως προς τις μεταβλητές σχεδιασμού και αβεβαιότητας. Καθώς, όμως, αρκεί μόνο ο υπολογισμός της προβολής του μητρώου των μικτών παραγώγων σε διανύσματα, αναπτύσσονται δύο μέθοδοι βελτιστοποίησης με προβολές, οι pFOSM και pAPCE, σε αντιστοιχία με τις δύο μεθόδους UQ που προαναφέρθηκαν. Βασικό κέρδος από τη χρήση των μεθόδων προβολής είναι πως το κόστος υπολογισμού του προβεβλημένου μητρώου μικτών παραγώγων δεν αυξάνεται με το πλήθος των μεταβλητών αβεβαιότητας ή σχεδιασμού.

Στην pFOSM, το κόστος ανά κύκλο βελτιστοποίησης είναι ίσο με 4 Υπολογιστικές Μονάδες (Equivalent Flow Solutions, EFS), ως δηλαδή να λύνεται το πρόβλημα ροής 4 φορές. Στην pAPCE, το κόστος είναι ίσο με $4L$ EFS (L είναι το πλήθος των σημείων δειγματοληψίας στην παλινδρόμηση). Το κέρδος είναι σημαντικό ακόμα και για μικρό πλήθος μεταβλητών αβεβαιότητας. Η pAPCE έχει μεγαλύτερο υπολογιστικό κόστος από την pFOSM, αλλά υπολογίζει στατιστικές ροπές με μεγαλύτερη ακρίβεια. Οι δύο μέθοδοι παρουσιάζονται σε προβλήματα αεροδυναμικής βελτιστοποίησης μορφής για στρωτές και τυρβώδεις ροές. Στις τελευταίες, η επίλυση γίνεται με χρήση του μοντέλου τύρβης των Spalart-Allmaras και της συζυγούς του εξίσωσης.

Οι βιομηχανικές εφαρμογές περιλαμβάνουν την αεροδυναμική βελτιστοποίηση μορφής οχημάτων επίγειας μεταφοράς με χρήση μόνιμων επιλυτών σε συνδυασμό με την RPM και μιας πτέρυγας αεροσκάφους παρουσία αβεβαιοτήτων ως προς τις συνθήκες ροής με χρήση των pFOSM και pAPCE.

Λέξεις κλειδιά: Αεροδυναμική Βελτιστοποίηση Μορφής, Συνεχής Συζυγής Μέθοδος, Στιβαρός Σχεδιασμός, Μέθοδος Αναδρομικών Προβολών, Ευστάθεια Εξισώσεων, Συζυγής Ανάστροφη Συμμεταφορά, Ποσοτικοποίηση Αβεβαιότητας, Ανάπτυγμα Πολυωνυμικού Χάους, OpenFOAM

Ακρωνύμια

APCE	Ανάπτυγμα Πολυωνυμικού Χάους υποβοηθούμενου από τη Συζυγή Μέθοδο
ATC	Συζυγής Ανάστροφη Συμμεταφορά
BC	Οριακή Συνθήκη
CFD	Υπολογιστική Ρευστοδυναμική
CPs	Σημεία Ελέγχου
EFS	Υπολογιστική Μονάδα
FDs	Πεπερασμένες Διαφορές
FOSM	Πρώτης-Τάξης Δεύτερης-Ροπής
pAPCE	προβεβλημένη APCE
PCE	Ανάπτυγμα Πολυωνυμικού Χάους
PDE	Μερική Διαφορική Εξίσωση
pFOSM	προβεβλημένη FOSM
QoI	Ποσότητα Ενδιαφέροντος
RANS	Reynolds-Averaged Navier-Stokes
RPM	Μέθοδος Αναδρομικών Προβολών
SDs	Παράγωγοι Ευαισθησίας
UQ	Ποσοτικοποίηση Αβεβαιοτήτων
URANS	Unsteady RANS
VBS	Ογκικές B-Splines

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

Η βελτιστοποίηση με μεθόδους CFD είναι πλέον αναπόσπαστο τμήμα της διαδικασίας σχεδιασμού σχετικών βιομηχανικών εφαρμογών. Συνδυάζει αριθμητικές μεθόδους επίλυσης των εξισώσεων ροής με μεθόδους βελτιστοποίησης. Παρά την ευρεία χρήση αυτών των μεθόδων στον σχεδιασμό, εξακολουθούν να υπάρχουν σημαντικές προκλήσεις, οι οποίες σχετίζονται με την αποτελεσματικότητα, την ευστάθεια και την ακρίβεια των αριθμητικών μεθόδων και μοντέλων. Προκειμένου οι μέθοδοι να είναι χρηστικές, είναι σημαντική η επίτευξη μικρότερων χρόνων διεκπεραίωσης των βελτιστοποιήσεων. Αυτή η διατριβή συμβάλλει στην ανάπτυξη ταχύτερων, ευσταθών και αποτελεσματικών μεθόδων και πρακτικών για βελτιστοποίηση με βοήθεια της CFD, ακόμη και υπό αβεβαιότητες.

Στη βελτιστοποίηση, διακρίνονται τρία κύρια στοιχεία: η συνάρτηση στόχου, οι περιορισμοί και οι μεταβλητές σχεδιασμού. Για τον υπολογισμό της τιμής της συνάρτησης στόχου, επιλύονται οι εξισώσεις ροής, ή αλλιώς οι πρωτεύουσες PDEs, με χρήση αριθμητικών μεθόδων. Οι μέθοδοι που αναπτύχθηκαν στη διατριβή εντάσσονται στις αιτιοκρατικές μεθόδους βελτιστοποίησης, οι οποίες χρησιμοποιούν τις SDs των συναρτήσεων στόχου και περιορισμών. Υπάρχουν διάφορες μέθοδοι που χρησιμοποιούνται για τον υπολογισμό των SDs [13, 21], ανάμεσα τους η συζυγής μέθοδος, το κόστος της οποίας είναι ανεξάρτητο του πλήθους των μεταβλητών σχεδιασμού [19]. Με τη συζυγή μέθοδο, αναπτύσσεται και, κατόπιν, επιλύεται ένα σύστημα συζυγών εξισώσεων για τις συζυγείς μεταβλητές. Το κόστος επίλυσης αυτού του συστήματος είναι ισοδύναμο με το κόστος επίλυσης των πρωτευουσών εξισώσεων και θα ποσοτικοποιείται ως μία EFS. Παραδείγματα συζυγών μεθόδων για αεροδυναμική βελτιστοποίηση μπορούν να βρεθούν στη βιβλιογραφία [19, 13].

Η διατριβή ασχολείται μόνο με προβλήματα βελτιστοποίησης μορφής, παρότι οι μέθοδοι που αναπτύχθηκαν είναι γενικότερης χρήσης. Χρησιμοποιούνται τεχνικές παραμετροποίησης μιας επιφάνειας, όπως είναι οι NURBS, [22] ή και του όγκου ταυτοχρόνως, όπως οι VBS [7, 12, 20]. Κατά τη βελτιστοποίηση μορφής μιας αεροτομής, χρησιμοποιείται επίσης και η τεχνική PARSEC [25], της οποίας οι παράμετροι περικλείουν χαρακτηριστικές πληροφορίες για τη γεωμετρία μιας

αεροτομής.

1.1 Στόχοι και Δομή της Διατριβής

Η διατριβή αναπτύσσεται σε δύο σκέλη, επεκτείνοντας προηγούμενες διδακτορικές διατριβές [32, 17, 6, 2] πάνω στη συνεχή συζυγή μέθοδο. Το πρώτο σκέλος ασχολείται με την αντιμετώπιση των δυσκολιών σύγκλισης των χρονικά μόνιμων πρωτεύοντων και των αντίστοιχων συνεχών συζυγών επιλυτών σε περιπτώσεις όπου αυτές προκαλούνται από (συνήθως ήπια) χρονική αστάθεια της ροής. Οι επιλύτες αυτοί είναι μέρος της *adjointOptimisation* βιβλιοθήκης που αναπτύχθηκε από την PCOpt/NTUA και είναι ελεύθερης χρήσης. Το δεύτερο σκέλος αφορά την ανάπτυξη τεχνικών βελτιστοποίησης υπό αβεβαιότητες. Η μείωση του υπολογιστικού κόστους της βελτιστοποίησης μορφής, με ή χωρίς αβεβαιότητες, συμπεριλαμβανομένων προβλημάτων με ήπια χρονική αστάθεια, είναι ο κύριος στόχος των τεχνικών που αναπτύχθηκαν. Όλες οι μέθοδοι προγραμματίστηκαν στην εργαλειοθήκη CFD του OpenFOAM [16].

Στο Κεφάλαιο 2, παρουσιάζεται η συνεχής συζυγής μέθοδος για προβλήματα βελτιστοποίησης μορφής που διέπονται από τις μόνιμες RANS εξισώσεις. Η ανάλυση της τύρβης γίνεται με χρήση του μοντέλου τύρβης των Spalart-Allmaras. Το κεφάλαιο αυτό αποτελεί τη βάση για όλες τις μεθόδους που αναπτύσσονται στα επόμενα.

Το Κεφάλαιο 3 περιλαμβάνει μια παρουσίαση του αλγορίθμου της Μεθόδου Αναδρομικών Προβολών (Recursive Projection Method, RPM) που χρησιμοποιείται για τη σταθεροποίηση των μόνιμων πρωτεύοντων και συζυγών επιλυτών. Πέρα από το ζήτημα των δυσκολιών σύγκλισης που παρουσιάζουν οι μόνιμοι επιλύτες κατά την επίλυση χρονικά μη-μόνιμων ροών, τίθεται το ερώτημα του κατά πόσο η κοινή πρακτική της χρήσης μόνιμων επιλυτών για βελτιστοποίηση σε τέτοιες ροές είναι αποτελεσματική. Εξετάζονται η σκοπιμότητα αυτής της προσέγγισης και τα οικονομικά οφέλη από τη χρήση της. Επιπλέον, αξιολογείται η επίδραση που έχει η απόσβεση της ATC στις SDs με τη βοήθεια της RPM. Η προτεινόμενη τεχνική παρουσιάζεται σε 2D προβλήματα βελτιστοποίησης μορφής. Για την εξαγωγή συμπερασμάτων, παρουσιάζονται προβλήματα με ροές στις οποίες εκλύονται στρόβιλοι.

Το Κεφάλαιο 4 παρουσιάζει τις μεθόδους pFOSM και pAPCE για βελτιστοποίηση σχήματος παρουσία αβεβαιοτήτων. Καθεμιά από αυτές μπορεί να θεωρηθεί επέκταση μιας υπάρχουσας μεθόδου UQ. Συγκεκριμένα, χρησιμοποιούνται οι μέθοδοι FOSM και APCE. Η βελτιστοποίηση γίνεται για συναρτήσεις στόχου που συνδυάζουν τις στατιστικές ροπές μιας QoI. Και οι δύο επεκτείνονται για τυρβώδεις ροές με τη συμπερίληψη του μοντέλου Spalart-Allmaras στο πλαίσιο. Το βασικό χαρακτηριστικό των pFOSM και pAPCE είναι η δυνατότητα υπολογισμού των SDs της συνάρτησης στόχου με κόστος σχεδόν ίσο με το κόστος της UQ χάριν της χρήσης μιας τεχνικής προβολής.

Παρουσιάζονται προβλήματα αεροδυναμικής βελτιστοποίησης υπό αβεβαιότητες και με τις δύο μεθόδους.

Στο Κεφάλαιο 5 παρουσιάζονται η χρήση της RPM και των pFOSM και pAPCE σε βιομηχανικές εφαρμογές. Σε αυτές περιλαμβάνονται η αεροδυναμική βελτιστοποίηση της μορφής οχημάτων εδάφους, χρησιμοποιώντας σταθεροποιημένους μόνιμους επιλύτες και την RPM, και μιας πτέρυγας αεροσκάφους, παρουσία αβεβαιοτήτων, χρησιμοποιώντας τις pFOSM και pAPCE.

Κεφάλαιο 2

Η Συνεχής Συζυγής Μέθοδος

Σε αυτό το κεφάλαιο, παρουσιάζεται η συνεχής συζυγής μέθοδος για προβλήματα που διέπονται από τις ασυμπίεστες RANS εξισώσεις. Οι χρονικά μόνιμες RANS εξισώσεις, μαζί με την εξίσωση μοντέλου Spalart-Allmaras [26] και μια PDE για τον υπολογισμό αποστάσεων, είναι

$$\mathcal{R}^p = -\frac{\partial v_i}{\partial x_i} = 0 \tag{2.1a}$$

$$\mathcal{R}_i^v = v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} + \frac{\partial p}{\partial x_i} = 0 \ , \qquad i = 1, 2(, 3) \tag{2.1b}$$

$$\mathcal{R}^{\widetilde{\nu}} = v_i \frac{\partial \widetilde{\nu}}{\partial x_i} - \frac{\partial}{\partial x_i}\left[\frac{\nu + \widetilde{\nu}}{c_\sigma}\frac{\partial \widetilde{\nu}}{\partial x_i}\right] - \frac{c_{b_2}}{c_\sigma}\left(\frac{\partial \widetilde{\nu}}{\partial x_i}\right)^2 + \widetilde{\nu}\,(D - P) = 0 \tag{2.1c}$$

$$\mathcal{R}^{\Delta} = \frac{\partial}{\partial x_i}\left(\frac{\partial \Delta}{\partial x_i}\Delta\right) - \Delta\frac{\partial^2 \Delta}{\partial x_i^2} - 1 = 0 \tag{2.1d}$$

όπου v_i είναι οι συνιστώσες της ταχύτητας του ρευστού, p η πίεση διαιρεμένη με την πυκνότητα του ρευστού, $\tau_{ij} = (\nu + \nu_t)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)$ οι τάσεις και ν και ν_t είναι το κινηματικό ροϊκό και τυρβώδες ιξώδες του ρευστού, αντίστοιχα. Οι εξισώσεις 2.1a και 2.1b είναι οι εξισώσεις συνέχειας και ορμής, αντίστοιχα, ενώ η εξ. 2.1c είναι η PDE του μοντέλου τύρβης των Spalart-Allmaras που επιλύεται για τη μεταβλητή $\widetilde{\nu}$. Το μοντέλο αυτό χρησιμοποιεί το πεδίο των αποστάσεων των κέντρων των υπολογιστικών κελιών από τα πλησιέστερα στερεά τοιχώματα, Δ, που υπολογίζεται με την επίλυση της εξίσωσης Hamilton-Jacobi [27], 2.1d. Στην εξ. 2.1c, οι P και D είναι όροι παραγωγής και καταστροφής τύρβης, αντίστοιχα, που δίνονται στο [24]. Το τυρβώδες ιξώδες υπολογίζεται από τη σχέση $\nu_t = \widetilde{\nu} f_{v_1}$. Οι υπόλοιπες σταθερές

είναι $c_\sigma = 0.66666$, $\kappa = 0.41$, $c_{b_1} = 0.1355$, $c_{b2} = 0.622$, $c_{w_2} = 0.3$, $c_{w_3} = 2$, $c_{v_1} = 7.1$ και ϵ_{ijk} είναι το σύμβολο εναλλαγής Levi-Civita. Όλες μαζί, οι εξισώσεις 2.1 αποτελούν τις πρωτεύουσες εξισώσεις.

Σε όλα τα προβλήματα που ακολουθούν, το υπολογιστικό χωρίο Ω περικλείεται από τους εξής τύπους ορίων: είσοδοι (S_I), έξοδοι (S_O), στερεά τοιχώματα (S_W) και επίπεδα συμμετρίας (S_S), για τα οποία ισχύουν οι εξής BCs

$$
\begin{aligned}
S_I &: \boldsymbol{v} = \boldsymbol{v}_I \ , \ \frac{\partial p}{\partial n} = 0 \ , \ \widetilde{\nu} = \widetilde{\nu}_I \ , \ \frac{\partial \Delta}{\partial n} = 1 \\
S_O &: \frac{\partial v_i}{\partial n} = 0 \ , \ p = 0 \ , \ \frac{\partial \widetilde{\nu}}{\partial n} = 0 \ , \ \frac{\partial \Delta}{\partial n} = 1 \\
S_W &: v_i = 0 \ , \ \frac{\partial p}{\partial n} = 0 \ , \ \widetilde{\nu} = 0 \ , \ \Delta = 0 \\
S_S &: v_{\langle n \rangle} = 0 \ , \ \frac{\partial v_{\langle t^l \rangle}}{\partial n} = 0 \text{ for } l = I, II \ , \ \frac{\partial p}{\partial n} = 0 \ , \ \frac{\partial \widetilde{\nu}}{\partial n} = 0 \ , \ \frac{\partial \Delta}{\partial n} = 0
\end{aligned}
\tag{2.2}
$$

Στην εξ. 2.2, η παράγωγος $\partial()/\partial n$ υπολογίζεται κατά τη κατεύθυνση του μοναδιαίου κάθετου στην επιφάνεια του ορίου διανύσματος $\boldsymbol{n}$, το οποίο είναι προσανατολισμένο προς το εξωτερικό του χωρίου. Τα $\boldsymbol{t}^I$ και $\boldsymbol{t}^{II}$ είναι δύο μοναδιαία διανύσματα, παράλληλα με το S. Το πρώτο, $\boldsymbol{t}^I$, ορίζεται αυθέραιτα και το δεύτερο προκύπτει από $\boldsymbol{t}^{II} = \boldsymbol{n} \times \boldsymbol{t}^I$. Επίσης, $v_{\langle n \rangle} = v_i n_i$ και $v_{\langle t^l \rangle} = v_i t_i^l$.

Για την ανάπτυξη της συνεχούς συζυγούς μεθόδου, ορίζεται πρώτα η συνάρτηση Lagrange,

$$
\mathcal{L} = F + \int_\Omega \left(u_i \mathcal{R}_i^{\boldsymbol{v}} + q \mathcal{R}^p + \widetilde{\nu}_\alpha \mathcal{R}^{\widetilde{\nu}} + \Delta_\alpha \mathcal{R}^\Delta \right) d\Omega \tag{2.3}
$$

χρησιμοποιώντας τα υπόλοιπα των εξισώσεων 2.1 και ένα σύνολο πολλαπλασιαστών Lagrange ή αλλιώς συζυγών μεταβλητών. Αυτές είναι οι u_i, q, $\widetilde{\nu}_\alpha$ και Δ_α, κατά αντιστοιχία με τις πρωτεύουσες μεταβλητές v_i, p, $\widetilde{\nu}$ και Δ. Επειδή τα υπόλοιπα των πρωτευουσών εξισώσεων τείνουν στο μηδέν, οι SDs της $\mathcal{L}$ ως προς ένα σύνολο μεταβλητών σχεδιασμού b_n, με $n \in [1, N]$, ισούνται με αυτές της F. Η εξ. 2.3 διαφορίζεται ως προς τις b_n και δίνει

$$
\frac{\delta \mathcal{L}}{\delta b_n} = \frac{\delta F}{\delta b_n} + \int_\Omega \left(u_i \frac{\delta \mathcal{R}_i^{\boldsymbol{v}}}{\delta b_n} + q \frac{\delta \mathcal{R}^p}{\delta b_n} + \widetilde{\nu}_\alpha \frac{\delta \mathcal{R}^{\widetilde{\nu}}}{\delta b_n} + \Delta_\alpha \frac{\delta \mathcal{R}^\Delta}{\delta b_n} \right) d\Omega \tag{2.4}
$$

και ο υπολογισμός της απαιτεί τη διαφόριση των πρωτευουσών εξισώσεων.

Κάνοντας το ανάπτυγμα για τις παραγώγους των υπόλοιπων των πρωτευουσών

PDEs, όπως και στις διατριβές [32, 17, 6, 2], εξάγωνται οι συζυγείς πεδιακές PDEs,

$$\mathcal{R}^q = -\frac{\partial u_i}{\partial x_i} = 0 \tag{2.5a}$$

$$\mathcal{R}_i^{\boldsymbol{u}} = \underbrace{u_j \frac{\partial v_j}{\partial x_i}}_{\boldsymbol{ATC}} - v_j \frac{\partial u_i}{\partial x_j} - \frac{\partial \tau_{ij}^{\alpha}}{\partial x_j} + \frac{\partial q}{\partial x_i} + \widetilde{\nu}_\alpha \frac{\partial \widetilde{\nu}}{\partial x_i} - \frac{\partial}{\partial x_l}\left(\widetilde{\nu}_\alpha \widetilde{\nu} \frac{C_\omega}{\omega} Y_j \epsilon_{jli}\right) = 0 \ , \quad i = 1, 2(, 3) \tag{2.5b}$$

$$\begin{aligned}\mathcal{R}^{\widetilde{\nu}_\alpha} = & -v_i \frac{\partial \widetilde{\nu}_\alpha}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\frac{\nu + \widetilde{\nu}}{c_\sigma} \frac{\partial \widetilde{\nu}_\alpha}{\partial x_i}\right) + \frac{1}{c_\sigma} \frac{\partial \widetilde{\nu}_\alpha}{\partial x_i} \frac{\partial \widetilde{\nu}}{\partial x_i} + \frac{2c_{b2}}{c_\sigma} \frac{\partial}{\partial x_i}\left(\widetilde{\nu}_\alpha \frac{\partial \widetilde{\nu}}{\partial x_i}\right) \\ & + \widetilde{\nu}_\alpha \widetilde{\nu} C_{\widetilde{\nu}} + f_{v_1} \frac{\partial u_j}{\partial x_i}\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j}\right) + \widetilde{\nu}_\alpha (D - P) = 0\end{aligned} \tag{2.5c}$$

$$\mathcal{R}^{\Delta_\alpha} = -2 \frac{\partial}{\partial x_i}\left(\Delta_\alpha \frac{\partial \Delta}{\partial x_i}\right) + \widetilde{\nu}_\alpha \widetilde{\nu} C_\Delta = 0 \tag{2.5d}$$

οι οποίες επιλύονται για τις u_i, q, $\widetilde{\nu}_\alpha$ και Δ_α. Ο όρος με σήμανση ATC στο δεξί μέλος της εξ. 2.5b είναι η επονομαζόμενη Συζυγής Ανάστροφη Συμμεταφορά (Adjoint Transposed-Convection, ATC) που, συχνά, ευθύνεται για τα περισσότερα προβλήματα σύγκλισης που παρουσιάζονται κατά την επίλυση των συζυγών εξισώσεων. Στις εξισώσεις 2.5, οι όροι $C_{\widetilde{\nu}}$, C_ω και C_Δ και οι οριακές συνθήκες μπορούν να βρεθούν στο [19]. Λύνοντας τις εξισώσεις 2.5, οι SDs της F μπορούν να υπολογιστούν από την ακόλουθη σχέση

$$\begin{aligned}\left.\frac{\delta \mathcal{L}}{\delta b_n}\right|_{FI} \equiv \left.\frac{\delta F}{\delta b_n}\right|_{FI} = & \int_{S_{W_p}} \frac{\partial F_{S,i}}{\partial x_k} n_i \frac{\delta x_k}{\delta b_n} dS + \int_{S_{W_p}} F_{S,i} \frac{\delta (n_i dS)}{\delta b_n} + \int_\Omega (\mathcal{A}_{jk}^L + \mathcal{A}_{jk}^T) \frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right) d\Omega \\ & + \int_{S_{W_p}} \left(u_{\langle n \rangle} - \phi_{\langle n \rangle \langle n \rangle}\right) \tau_{ij} \frac{\delta (n_i n_j)}{\delta b_n} dS - \int_{S_{W_p}} \left(\phi_{\langle t^I \rangle \langle t^I \rangle}\right) \tau_{ij} \frac{\delta (t_i^I t_j^I)}{\delta b_n} dS \\ & - \int_{S_{W_p}} \left(\phi_{\langle t^{II} \rangle \langle t^{II} \rangle}\right) \tau_{ij} \frac{\delta (t_i^{II} t_j^{II})}{\delta b_n} dS - \int_{S_{W_p}} \left(\phi_{\langle t^{II} \rangle \langle t^I \rangle} + \phi_{\langle t^I \rangle \langle t^{II} \rangle}\right) \tau_{ij} \frac{\delta (t_i^{II} t_j^I)}{\delta b_n} dS\end{aligned} \tag{2.6}$$

όπου $\phi_{\langle n \rangle \langle n \rangle} = \frac{\partial F_{S,k}}{\partial \tau_{ij}} n_i n_j n_k$, $\phi_{\langle t^I \rangle \langle t^{II} \rangle} = \frac{\partial F_{S,k}}{\partial \tau_{ij}} t_i^I t_j^{II} n_k$ και ούτω καθεξής. Επίσης, οι όροι

$\mathcal{A}^L_{jk}$ και $\mathcal{A}^T_{jk}$ δίνονται στο [18].

Κεφάλαιο 3

Σταθεροποίηση Μόνιμων Επιλυτών Παρουσία Χρονικής Αστάθειας της Ροής

Αυτό το κεφάλαιο παρουσιάζει την RPM [23] με την οποία αντιμετωπίζονται οι δυσκολίες σύγκλισης των χρονικά μόνιμων επιλυτών ροϊκών και συνεχών συζυγών εξισώσεων, οι οποίες εμφανίζονται σε ροές με (ήπια) χρονική αστάθεια.

Διερευνάται το κατά πόσο οι μόνιμοι επιλύτες ροής και οι αντίστοιχοι συζυγείς, οι οποίοι έχουν σταθεροποιηθεί με την RPM, μπορούν πράγματι να υποστηρίξουν τη βελτιστοποίηση μορφής σε ροές με ήπια έως μέτρια χρονική αστάθεια. Η χρήση τους σε τέτοιες περιπτώσεις γίνεται καθαρά για λόγους εξοικονόμησης κόστους. Για την εξαγωγή συμπερασμάτων, χρησιμοποιούνται παραδείγματα βελτιστοποίησης σε ροές στις οποίες εκλύονται στρόβιλοι πίσω από στερεά σώματα. Οι βελτιστοποιημένες με μόνιμους επιλύτες γεωμετρίες επαναξιολογούνται χρησιμοποιώντας επιλύτες URANS εξισώσεων.

Τέλος, δεδομένου ότι η RPM επιτρέπει τη σταθεροποίηση των συζυγών επιλυτών χωρίς την απαλοιφή όρων, μπορεί να υπολογιστεί μια συζυγής λύση αναφοράς. Για τα προβλήματα που εξετάζονται σε αυτήν τη διατριβή, αυτή η λύση είναι αδύνατο να υπολογιστεί χωρίς την RPM. Η λύση αναφοράς αποτελεί ένα μέσο αξιολόγησης της επίδρασης των τεχνικών της απόσβεσης της ATC και της χρήσης των πρωτεύοντων "μέσων πεδίων" στην ακρίβεια των SDs. Η επίδραση των τεχνικών αυτών αξιολογείται συγκρίνοντας τις SDs που υπολογίζονται με τη βοήθεια αυτών των τεχνικών έναντι των SDs που υπολογίστηκαν με τις λύσεις αναφοράς. Αυτή η αξιολόγηση θα δώσει χρήσιμα πορίσματα για τον χειρισμό δυσκολότερων προβλημάτων (βλ. 5.2) όπου η RPM θα συνδυαστεί με ήπια απόσβεση της ATC για αυξημένη ευστάθεια.

3.1 Η Μέθοδος Αναδρομικών Προβολών

Η RPM αναπτύχθηκε από τους [23] και είναι μια μέθοδος σταθεροποίησης επαναληπτικών σχημάτων της μορφής

$$\boldsymbol{U}^{(n+1)} = \boldsymbol{G}(\boldsymbol{U}^{(n)}) \tag{3.1}$$

όπου $\boldsymbol{U} \in \mathbb{R}^N$ είναι ένα διάνυσμα των (πρωτεύοντων ή συζυγών) αγνώστων στην n-οστή επανάληψη. Το κατά πόσο θα συγκλίνει ένα τέτοιο επαναληπτικό σχήμα εξαρτάται από τις ιδιοτιμές του Ιακωβιανού μητρώου $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$. Εάν όλες οι ιδιοτιμές έχουν τιμή μικρότερης της μονάδας, το σχήμα αναμένεται να συγκλίνει στη λύση $\boldsymbol{U}^*$. Το αντίθετο αναμένεται να συμβεί ακόμη και αν μόνο η μεγαλύτερη ιδιοτιμή σε μέγεθος υπερβαίνει τη μονάδα [3]. Σε μια τέτοια περίπτωση και υπό ορισμένες συνθήκες, η RPM μπορεί να εξαναγκάσει το σχήμα της εξ. 3.1 να συγκλίνει σε μια λύση.

Αρχικά, η μέθοδος πρέπει να προσεγγίσει μια ορθοκανονική βάση $V_p \in \mathbb{R}^{N \times N_p}$ για τον υποχώρο που ορίζουν τα N_p ιδιοδιανύσματα που σχετίζονται με τις ιδιοτιμές της $\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}}$ που είναι μεγαλύτερες από τη μονάδα. Ορίζονται δύο υποχώροι του $\mathbb{R}^N$: ο αναλλοίωτος ασταθής υποχώρος $\mathbb{P}$, για τον οποίο η V_p αποτελεί βάση και το ορθογώνιο συμπλήρωμά του $\mathbb{Q}$. Χρησιμοποιώντας τη V_p, ορίζονται οι πίνακες προβολής από τον $\mathbb{R}^N$ στους $\mathbb{P}$ και $\mathbb{Q}$, αντίστοιχα, μέσω των ακόλουθων σχέσεων

$$P = V_p V_p^T, \quad Q = I - V_p V_p^T = V_q V_q^T, \quad V_q \in \mathbb{Q}^{N \times N_q} \tag{3.2}$$

όπου $N_q = N - N_p$. Οι στήλες και οι γραμμές του P είναι ορθογώνιες προς τις γραμμές και τις στήλες του Q. Το N_p είναι αρχικά μηδέν και, στη συνέχεια, αυξάνεται καθώς τα ιδιοδιανύσματα που προκαλούν απόκλιση ή/και αργή σύγκλιση προσαρτώνται σταδιακά στη V_p. Τα ιδιοδιανύσματα που προκαλούν απόκλιση ανήκουν αρχικά στον $\mathbb{Q}$, αλλά, καθώς προστίθενται στη V_p, ο $\mathbb{Q}$ συρρικνώνεται ενώ ο $\mathbb{P}$ αυξάνεται.

Η $\boldsymbol{U}$ αναλύεται σε $\boldsymbol{U} = \boldsymbol{U}_p + \boldsymbol{U}_q$ όπου $\boldsymbol{U}_p = P\boldsymbol{U} \in \mathbb{P}$ και $\boldsymbol{U}_q = Q\boldsymbol{U} \in \mathbb{Q}$ είναι οι ασταθείς και ευσταθείς συνιστώσες της λύσης, αντίστοιχα. Οι P και Q εφαρμόζονται και στο δεξί μέλος της εξ. 3.1 για να δώσουν $\boldsymbol{G}_p = P\boldsymbol{G}$ και $\boldsymbol{G}_q = Q\boldsymbol{G}$. Με αυτόν τον διαχωρισμό, το επαναληπτικό σχήμα της εξ. 3.1 γράφεται σε δύο συνιστώσες,

$$\boldsymbol{U}_p^{(n+1)} = \boldsymbol{G}_p\left(\boldsymbol{U}_p^{(n)}, \boldsymbol{U}_q^{(n)}\right) \tag{3.3a}$$

$$\boldsymbol{U}_q^{(n+1)} = \boldsymbol{G}_q\left(\boldsymbol{U}_p^{(n)}, \boldsymbol{U}_q^{(n)}\right) \tag{3.3b}$$

Η εξ. 3.3a ευθύνεται για την απόκλιση. Η βασική ιδέα πίσω από την RPM είναι η αντικατάσταση της εξ. 3.3a με ένα βήμα Newton-Raphson,

$$\boldsymbol{U}_p^{(n+1)} = \boldsymbol{U}_p^{(n)} + \left(I - \frac{\partial \boldsymbol{G}_p}{\partial \boldsymbol{U}_p}^{(n)}\right)^{-1} \left(\boldsymbol{G}_p\left(\boldsymbol{U}_p^{(n)}, \boldsymbol{U}_q^{(n)}\right) - \boldsymbol{U}_p^{(n)}\right) \tag{3.4}$$

Συνδυάζοντας τις εξισώσεις 3.4 και 3.3b προκύπτει ένα σταθεροποιημένο επαναληπτικό σχήμα με το οποίο η εξ. 3.4 επιλύεται για την $\boldsymbol{U}_p$ ενώ η εξ. 3.3b χρησιμοποιείται για την $\boldsymbol{U}_q$. Με αυτό το σχήμα, η RPM μπορεί να βοηθήσει τη σύγκλιση επιλυτών που αποκλίνουν. Σημαντικό ρόλο για την επίτευξη αυτού του σκοπού έχει ο ακριβής υπολογισμός της βάσης V_p. Περισσότερες πληροφορίες για τις προϋποθέσεις σύγκλισης του σταθεροποιημένου επαναληπτικού σχήματος δίνονται στις [23, 10]. Το Newton βήμα μπορεί να εκτελεστεί σε πολύ χαμηλό

κόστος, υπό την προϋπόθεση η διάσταση του $\mathbb{P}$ να είναι σχετικά μικρή, με τον τρόπο που παρουσιάζεται από τους [23].

3.2 Βελτιστοποίηση Μορφής Μεμονωμένης Αεροτομής

Εδώ γίνεται βελτιστοποίηση της μεμονωμένης NACA0012 αεροτομής με σκοπό τη μείωση του C_D και αύξηση του C_L. Η στρωτή ροή γύρω από αυτή την αεροτομή για $Re = 1000$ έχει μελετηθεί στη βιβλιογραφία [9, 8], σε διάφορες γωνίες της αδιατάρακτης ροής, α_∞. Για γωνίες μεγαλύτερες από $\alpha_\infty \approx 8^o$, εμφανίζεται έντονη αποκόλληση, προκαλείται έκλυση στροβίλων και η ροή γίνεται μη-μόνιμη. Σκόπιμα, η βελτιστοποίηση γίνεται για $\alpha_\infty = 20^o$. Η παραμετροποίηση της αεροτομής έγινε με VBS και υπάρχει ένας περιορισμός που απαγορεύει τα CPs να καβαλήσουν. Επίσης, εφαρμόστηκε ένας περιορισμός στο εμβαδόν της κάθετης επιφάνειας της αεροτομής V, διατηρώντας την εντός $\pm 0.1\%$ της αρχικής της τιμής $V^{\oplus}$ και η βελτιστοποίηση έγινε με τη μέθοδο ISQP [15].

Η RPM πέτυχε τη σταθεροποίηση του μόνιμου επιλύτη ροής και, έπειτα, των μόνιμων επιλυτών συζυγών εξισώσεων για τα C_D και C_L. Αρχικά, μελετήθηκε η επίδραση της απόσβεσης της ATC και της χρήσης των πρωτεύοντων "μέσων πεδίων" κατά την επίλυση των συζυγών εξισώσεων στην ακρίβεια των SDs. Η διάκριση ανάμεσα στην ήπια και ανεξέλεγκτη απόσβεση της ATC φαίνεται στο σχήμα 3.1. Με ήπια απόσβεση της ATC, η χρήση της RPM είναι απαραίτητη για τη σύγκλιση των συζυγών επιλυτών. Η ανεξέλεγκτη απόσβεση της ATC είναι όσο χρειάζονται οι συζυγείς επιλύτες προκειμένου να συγκλίνουν χωρίς τη βοήθεια της RPM. Οι SDs φαίνονται στο σχήμα 3.2. Η ήπια απόσβεση της ATC δεν έχει αρνητική επίδραση στην ακρίβεια των SDs. Επίσης, χρησιμοποιώντας πρωτεύοντα "μέσα πεδία", οι SDs είναι πολύ κοντά στις σωστές τιμές. Αντίστοιχα συμπεράσματα προκύπτουν και σε άλλες περιπτώσεις αλλά και για τις SDs του C_L σε αυτή την περίπτωση οι οποίες, όμως, παραλείπονται.

Η σύγκλιση του C_D κατά τη βελτιστοποίηση, η οποία πραγματοποιήθηκε είτε χρησιμοποιώντας μόνο την RPM είτε σε συνδυασμό με τις υπόλοιπες τεχνικές, φαίνεται στο σχήμα 3.3. Το C_D μειώθηκε κατά 2.6%. Με το τέλος της βελτιστοποίησης, όλες οι λύσεις επαναξιολογήθηκαν από έναν μη-μόνιμο επιλύτη, ο οποίος φανέρωσε μείωση του $\overline{C_D}$ κατά 7%. Η βελτιστοποίηση πραγματοποιήθηκε και με μη-μόνιμους επιλύτες, τα αποτελέσματα της οποίας φαίνονται στο σχήμα 3.4. Χρησιμοποιώντας ως σημείο εκκίνησης τη βελτιστοποιημένη λύση με μόνιμους επιλύτες και την RPM, η βελτιστοποίηση με μη-μόνιμους εκτελέστηκε γρηγορότερα. Σε αυτήν την περίπτωση, η βελτιστοποίηση με μη-μόνιμους επιλύτες δεν μπορούσε εν τέλει να αποφευχθεί. Όμως, το κόστος της βελτιστοποίησης με μη-μόνιμους μειώθηκε σημαντικά χρησιμοποιώντας τη λύση της βελτιστοποίησης με μόνιμους ως σημείο εκκίνησης. Αυτό το κόστος μειώθηκε κατά 40% ακολουθώντας αυτήν την πρακτική, λογαριάζοντας ότι το κόστος βελτιστοποίησης με μόνιμους επιλύτες έναντι αυτού με μη-μόνιμους είναι συγκριτικά αμελητέο. Αντίστοιχα

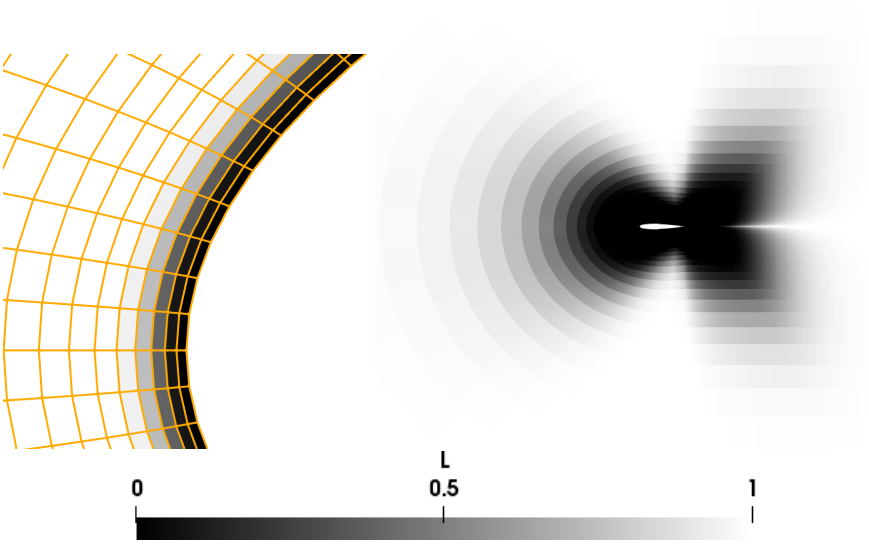


Figure 3.1: NACA0012 αεροτομή. Περιοχές απόσβεσης της ATC. **Αριστερά**: Ήπια απόσβεση της ATC στα 5 πρώτα κελιά μπροστά από τον τοίχο. **Δεξιά**: Ανεξέλεγκτη απόσβεση της ATC προκειμένου να επιτευχθεί σταθεροποίηση των συζυγών επιλυτών χωρίς τη βοήθεια της RPM.

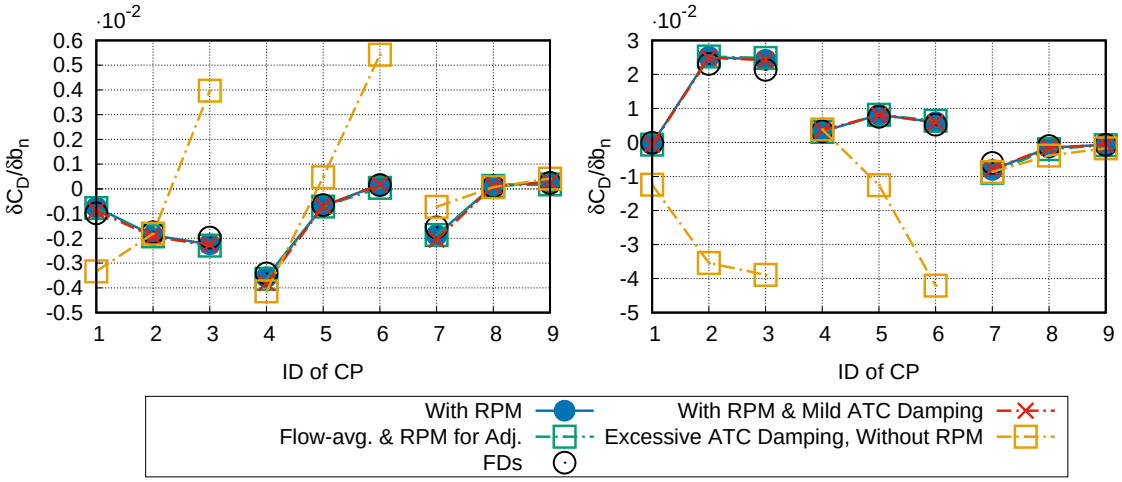


Figure 3.2: NACA0012 αεροτομή. SDs του C_D ως προς τις x (αριστερά) και y (δεξιά) συντεταγμένες των CPs. Ο δείκτης "Flow-avg" υποδηλώνει τις SDs που υπολογίστηκαν χρησιμοποιώντας τη μέση τιμή των 1000 τελευταίων πεδίων του μη-συγκλιμένου πρωτεύοντος επιλύτη (χωρίς δλδ. τη βοήθεια της RPM) ενώ ο αντίστοιχος συζυγής επιλύτης συνέκλινε με τη βοήθεια της RPM.

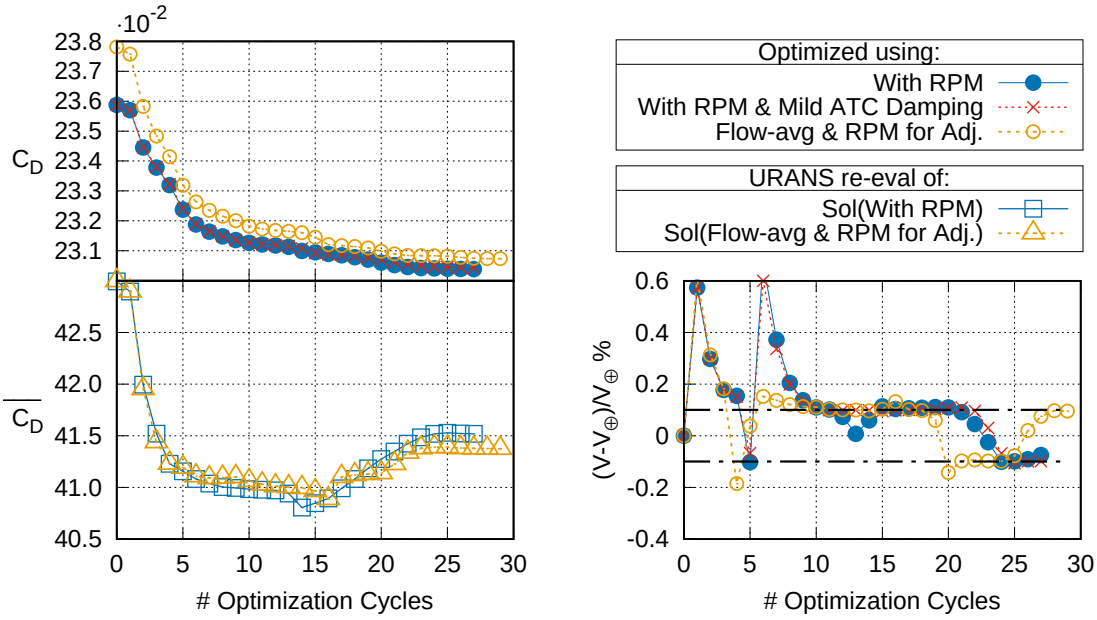


Figure 3.3: NACA0012 αεροτομή. Σύγκλιση βρόχου βελτιστοποίησης για μείωση του C_D. **Άνω-Αριστερά**: C_D ή $C_{D,\text{avg}}$ (υπολογισμένο με πρωτεύοντα "μέσα πεδία"). **Κάτω-Αριστερά**: Επαναξιολόγηση του $\overline{C_D}$ για κάθε κύκλο βελτιστοποίησης με έναν URANS επιλύτη. **Κάτω-Δεξιά**: Περιορισμός στο V. Το Sol("τεχνική") στη λεζάντα επεξήγησης υποδηλώνει ότι η λύση υπολογίστηκε χρησιμοποιώντας την εκάστοτε "τεχνική".

συμπεράσματα προκύπτουν και για τη βελτιστοποίηση με στόχο την αύξηση του C_L.

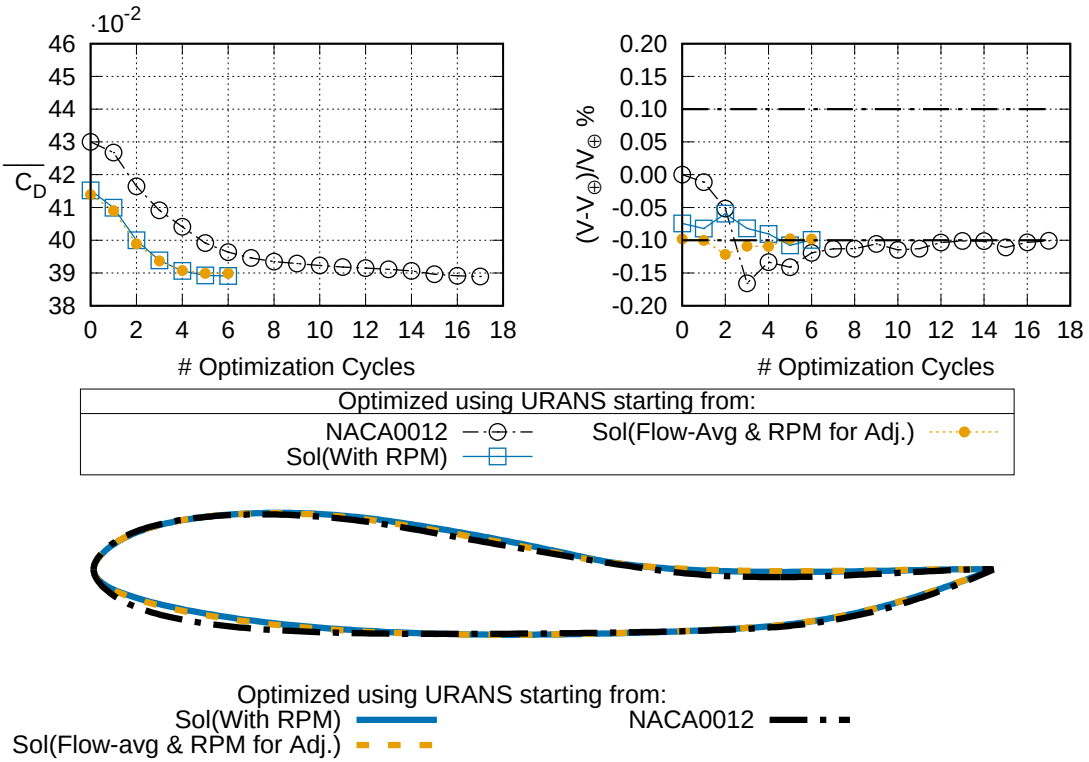


Figure 3.4: NACA0012 αεροτομή. Βρόχος βελτιστοποίησης με μη-μόνιμους επιλύτες για μείωση του $\overline{C_D}$. **Άνω-Αριστερά**: $\overline{C_D}$. **Άνω-Δεξιά**: Περιορισμός στο V. Η βελτιστοποίηση ξεκινάει κάθε φορά από άλλη αεροτομή. Αυτές είναι η (αρχική) NACA0012 και αυτές που υπολογίστηκαν στο τέλος των βρόχων βελτιστοποίησης που φαίνονται στο σχήμα 3.3. **Κάτω**: Βελτιστοποιημένες γεωμετρίες.

Κεφάλαιο 4

Βελτιστοποίηση Μορφής υπό Αβεβαιότητες

Σε προβλήματα βελτιστοποίησης μορφής υπό αβεβαιότητες, ορίζεται ένα σύνολο μεταβλητών αβεβαιότητας, οι $\boldsymbol{c} \in \mathbb{R}^M$. Η QoI είναι πλέον μία στοχαστική ποσότητα, έχει μέση τιμή μ_F και τυπική απόκλιση σ_F, και η συνάρτηση στόχου είναι πλέον η

παρακάτω

$$J = \mu_F + w\sigma_F \tag{4.1}$$

όπου w ένα βάρος που σταθμίζει τις δύο στατιστικές ροπές. Υπεισέρχονται μέθοδοι UQ των οποίων ο σκοπός είναι ο υπολογισμός των μ_F και σ_F. Σε αυτήν τη διατριβή, οι c_m σχετίζονται με τις συνθήκες ροής, ακολουθούν κανονική κατανομή και δεν υπάρχει συσχέτιση μεταξύ τους. Χρησιμοποιούνται οι μέθοδοι FOSM και APCE.

Κατά τη μέθοδο FOSM [29, 4], οι στατιστικές ροπές δίνονται από

$$\mu_F^{(FOSM)} = F(\boldsymbol{\mu}) \quad \text{και} \quad \sigma_F^{(FOSM)} = \sqrt{\left.\frac{\delta F}{\delta c_m}\right|_{\boldsymbol{\mu}} \left.\frac{\delta F}{\delta c_j}\right|_{\boldsymbol{\mu}} \mathcal{K}_{mj}} \tag{4.2}$$

όπου η F και οι $\frac{\delta F}{\delta c_m}$ υπολογίζονται στις μέσες τιμές των c_m και

$$\mathcal{K}_{mj} = \int_{-\infty}^{+\infty} (c_m - \mu_m)(c_j - \mu_j) h_c(\boldsymbol{c}) d\boldsymbol{c} \tag{4.3}$$

είναι το μητρώο συνδιασποράς των $\boldsymbol{c}$, το οποίο είναι διαγώνιο σε περίπτωση όπου οι c_m δεν συσχετίζονται μεταξύ τους. Οι $\frac{\delta F}{\delta c_m}$ υπολογίζονται με τη συνεχή συζυγή μέθοδο σε κόστος μίας EFS επιπλέον της επίλυσης των εξισώσεων ροής. Προκειμένου να γίνει βελτιστοποίηση με αυτήν τη μέθοδο, χρειάζονται οι $\frac{\delta J}{\delta b_n}$ που δίνονται από την παρακάτω σχέση

$$\frac{\delta J}{\delta b_n} = \frac{\delta F}{\delta b_n} + \frac{w}{\sigma_F^{(FOSM)}} z_m \frac{\delta^2 F}{\delta c_m \delta b_n} \quad \text{όπου} \quad z_m^{(FOSM)} = \mathcal{K}_{mj} \frac{\delta F}{\delta c_j} \tag{4.4}$$

Παρατηρείται ότι εδώ αρκεί μόνο ο υπολογισμός της προβολής του μητρώου $\frac{\delta^2 F}{\delta c_m \delta b_n}$ επάνω σε ένα διάνυσμα z_m. Ο υπολογισμός του μητρώου μπορεί να αποφευχθεί υπολογίζοντας αυτήν την προβολή. Αυτή είναι η βάση της pFOSM [1], η οποία συνδυάζει την τεχνική προβολής με τη FOSM.

Με τη μέθοδο μη-παρεμβατικού Αναπτύγματος Πολυωνυμικού Χάους (Polynomial Chaos Expansion, PCE) [30, 31], η F αναπτύσσεται ως

$$F(\boldsymbol{b}, \boldsymbol{c}(\boldsymbol{\zeta})) \approx \sum_{q=0}^{Q-1} \alpha_q \Psi_q(\boldsymbol{\zeta}) \tag{4.5}$$

όπου α_q είναι οι συντελεστές του PCE, Ψ_q είναι πολυδιάστατα ορθογώνια πολυώνυμα και $\boldsymbol{\zeta}$ είναι τυποποιημένες τυχαίες μεταβλητές για τις οποίες ισχύει $\boldsymbol{c} = \boldsymbol{c}(\boldsymbol{\zeta})$, ανάλογα με τον τύπο της κατανομής πιθανότητας. Το πλήθος των συντελεστών α_q είναι $Q =$

$\frac{(O_{PCE}+M)!}{O_{PCE}!M!}$. Οι στατιστικές ροπές δίνονται από

$$\mu_F^{(PCE)} = \alpha_0 \quad \text{και} \quad \sigma_F^{(PCE)} = \sqrt{\sum_{q=1}^{Q-1} \alpha_q^2 \|\Psi_q\|^2} \tag{4.6}$$

όπου

$$\|\Psi_i\|^2 = \int_{-\infty}^{+\infty} \Psi_j \Psi_j \delta_{ij} h_c(\boldsymbol{\zeta}) d\boldsymbol{\zeta} \tag{4.7}$$

είναι η κατά-h_c νόρμα των ορθογώνιων πολυωνύμων.

Για τον υπολογισμό των α_q, χρησιμοποιείται παλινδρόμηση με ελάχιστα τετράγωνα, έχοντας πρώτα υπολογίσει τις τιμές της F σε L σημεία δειγματοληψίας, αλλάζοντας τις τιμές των c_m με βάση την τεχνική Latin Hypercube Sampling [14], σε αυτήν τη διατριβή. Με τη συζυγή μέθοδο, που υπολογίζει τις $\frac{\delta F}{\delta c_m}$ σε κόστος μιας EFS, ανεξάρτητα του πληθους M, είναι δυνατή η συλλογή περισσότερης πληροφορίας ανά σημείο δειγματοληψίας. Με M επιπλέον εξισώσεις για το πρόβλημα ελαχίστων τετραγώνων, το πλήθος L μπορεί να μειωθεί. Αυτή είναι η βασική ιδέα πίσω από τη μέθοδο APCE. Διαμορφώνεται λοιπόν το σύστημα $D\boldsymbol{\alpha} = \boldsymbol{\beta}$, με $D \in \mathbb{R}^{L(M+1)\times Q}$ και $\boldsymbol{\beta} \in \mathbb{R}^{L(M+1)}$, το οποίο σε αναπτυγμένη μορφή είναι

$$\underbrace{\begin{bmatrix} \Psi_0(\boldsymbol{\zeta}^{(1)}) & \dots & \Psi_{Q-1}(\boldsymbol{\zeta}^{(1)}) \\ \vdots & \ddots & \vdots \\ \Psi_0(\boldsymbol{\zeta}^{(L)}) & \dots & \Psi_{Q-1}(\boldsymbol{\zeta}^{(L)}) \\ \frac{\partial \Psi_0}{\partial \zeta_1}(\boldsymbol{\zeta}^{(1)}) & \dots & \frac{\partial \Psi_{Q-1}}{\partial \zeta_1}(\boldsymbol{\zeta}^{(1)}) \\ \vdots & \ddots & \vdots \\ \frac{\partial \Psi_0}{\partial \zeta_M}(\boldsymbol{\zeta}^{(1)}) & \dots & \frac{\partial \Psi_{Q-1}}{\partial \zeta_M}(\boldsymbol{\zeta}^{(1)}) \\ \vdots & \ddots & \vdots \\ \frac{\partial \Psi_0}{\partial \zeta_1}(\boldsymbol{\zeta}^{(L)}) & \dots & \frac{\partial \Psi_{Q-1}}{\partial \zeta_1}(\boldsymbol{\zeta}^{(L)}) \\ \vdots & \ddots & \vdots \\ \frac{\partial \Psi_0}{\partial \zeta_M}(\boldsymbol{\zeta}^{(L)}) & \dots & \frac{\partial \Psi_{Q-1}}{\partial \zeta_M}(\boldsymbol{\zeta}^{(L)}) \end{bmatrix}}_{D} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{Q-1} \end{bmatrix} = \underbrace{\begin{bmatrix} F(\boldsymbol{c}^{(1)}) \\ \vdots \\ F(\boldsymbol{c}^{(L)}) \\ \frac{\partial c_m}{\partial \zeta_1} \frac{\delta F}{\delta c_m}(\boldsymbol{c}^{(1)}) \\ \vdots \\ \frac{\partial c_m}{\partial \zeta_M} \frac{\delta F}{\delta c_m}(\boldsymbol{c}^{(1)}) \\ \vdots \\ \frac{\partial c_m}{\partial \zeta_1} \frac{\delta F}{\delta c_m}(\boldsymbol{c}^{(L)}) \\ \vdots \\ \frac{\partial c_m}{\partial \zeta_M} \frac{\delta F}{\delta c_m}(\boldsymbol{c}^{(L)}) \end{bmatrix}}_{\boldsymbol{\beta}} \tag{4.8}$$

και επιλύεται με ελάχιστα τετράγωνα.

Για τη βελτιστοποίηση, χρειάζεται ο υπολογισμός της παρακάτω ποσότητας

$$\frac{\delta J}{\delta b_n} = \frac{\delta \alpha_0}{\delta b_n} + \sum_{q=1}^{Q-1} \left(\frac{w \alpha_q \|\Psi_q\|^2}{\sigma_F^{(PCE)}} \frac{\delta \alpha_q}{\delta b_n} \right) \tag{4.9}$$

Οι ποσότητες $\frac{\delta\alpha_q}{\delta b_n} = \left(D^T D\right)^{-1} D^T \frac{\delta\boldsymbol{\beta}}{\delta b_n}$, προκύπτουν διαφορίζοντας τη λύση του συστήματος της εξ. 4.8. Κάνοντας το ανάπτυγμα, προκύπτει τελικά η παρακάτω έκφραση

$$\frac{\delta J}{\delta b_n} = \sum_{l=1}^{L} k^{(l)} \frac{\delta F}{\delta b_n}(\boldsymbol{c}^{(l)}) + \sum_{l=1}^{L} z_m^{(l)} \frac{\delta^2 F}{\delta c_m \delta b_n}(\boldsymbol{c}^{(l)}) \tag{4.10}$$

όπου τα $k^{(l)}$ και $z_m^{(l)}$ είναι ποσότητες και συνιστώσες διανυσμάτων προβολής (διαφορετικές από αυτές της pFOSM), αντίστοιχα, και εμπεριέχονται μέσα στον πίνακα $\left(D^T D\right)^{-1} D^T$.

4.1 Η Τεχνική Προβολής

Για τον υπολογισμό του προβεβλημένου μητρώου στις εξισώσεις 4.4 και 4.10, διαφορίζεται η εξ. 2.6 ως προς c_m και εκτελείται η προβολή πάνω στο z_m. Εάν $\widehat{\psi} = z_m \frac{\delta\psi}{\delta c_m}$, συμβολίζεται μια προβεβλημένη παράγωγος μιας ποσότητας, τότε η έκφραση για το προβεβλημένο μητρώο μικτών παραγώγων δίνεται από

$$\begin{aligned}
z_m \frac{\delta^2 F}{\delta c_m \delta b_n}\bigg|_{FI} = & \int_{S_{W_p}} \frac{\partial \widehat{F}_{S,i}}{\partial x_k} n_i \frac{\delta x_k}{\delta b_n} dS + \int_{S_{W_p}} \widehat{F}_{S,i} \frac{\delta(n_i dS)}{\delta b_n} + \int_{\Omega} \left(\widehat{\mathcal{A}}^L_{jk} + \widehat{\mathcal{A}}^T_{jk}\right) \frac{\partial}{\partial x_j}\left(\frac{\delta x_k}{\delta b_n}\right) d\Omega \\
& + \int_{S_{W_p}} \left[\left(\widehat{u}_{\langle n\rangle} - \widehat{\phi}_{\langle n\rangle\langle n\rangle}\right)\tau_{ij} + \left(u_{\langle n\rangle} - \phi_{\langle n\rangle\langle n\rangle}\right)\widehat{\tau}_{ij}\right] \frac{\delta(n_i n_j)}{\delta b_n} dS \\
& - \int_{S_{W_p}} \left(\widehat{\phi}_{\langle t^I\rangle\langle t^I\rangle}\tau_{ij} + \phi_{\langle t^I\rangle\langle t^I\rangle}\widehat{\tau}_{ij}\right) \frac{\delta(t_i^I t_j^I)}{\delta b_n} dS \\
& - \int_{S_{W_p}} \left(\widehat{\phi}_{\langle t^{II}\rangle\langle t^{II}\rangle}\tau_{ij} + \phi_{\langle t^{II}\rangle\langle t^{II}\rangle}\widehat{\tau}_{ij}\right) \frac{\delta(t_i^{II} t_j^{II})}{\delta b_n} dS \\
& - \int_{S_{W_p}} \left[\left(\widehat{\phi}_{\langle t^{II}\rangle\langle t^I\rangle} + \widehat{\phi}_{\langle t^I\rangle\langle t^{II}\rangle}\right)\tau_{ij} + \left(\phi_{\langle t^{II}\rangle\langle t^I\rangle} + \phi_{\langle t^I\rangle\langle t^{II}\rangle}\right)\widehat{\tau}_{ij}\right] \frac{\delta(t_i^{II} t_j^I)}{\delta b_n} dS
\end{aligned} \tag{4.11}$$

όπου οι όροι $\widehat{\mathcal{A}}^L_{jk}$ και $\widehat{\mathcal{A}}^T_{jk}$ εξαρτώνται από τις μεταβλητές $\widehat{v}_i$, $\widehat{p}$, $\widehat{\tau}_{ij}$, $\widehat{\widetilde{\nu}}$, $\widehat{u}_i$, $\widehat{q}$, $\widehat{\widetilde{\nu}}_\alpha$, $\widehat{\tau}^\alpha_{ij}$, $\widehat{\Delta}_\alpha$. Για τον υπολογισμό τους, εξάγονται PDEs διαφορίζοντας τις εξισώσεις 2.1, οι

οποίες δίνουν

$$\widehat{\mathcal{R}}^{p}=-\frac{\partial\widehat{v}_j}{\partial x_j}=0 \tag{4.12a}$$

$$\widehat{\mathcal{R}}^{v}_{i}=\widehat{v}_j\frac{\partial v_i}{\partial x_j}+v_j\frac{\partial\widehat{v}_i}{\partial x_j}-\frac{\partial\widehat{\tau}_{ij}}{\partial x_j}+\frac{\partial\widehat{p}}{\partial x_i}=0\ , \qquad i=1,2 \tag{4.12b}$$

$$\begin{aligned}\widehat{\mathcal{R}}^{\widetilde{\nu}}=&\widehat{v}_j\frac{\partial\widetilde{\nu}}{\partial x_j}+v_j\frac{\partial\widehat{\widetilde{\nu}}}{\partial x_j}-\frac{\partial}{\partial x_j}\left[\frac{\nu+\widetilde{\nu}}{\sigma}\frac{\partial\widehat{\widetilde{\nu}}}{\partial x_j}\right]-\frac{\partial}{\partial x_j}\left[\frac{\widehat{\nu}+\widehat{\widetilde{\nu}}}{\sigma}\frac{\partial\widetilde{\nu}}{\partial x_j}\right]\\&-\frac{2c_{b_2}}{\sigma}\frac{\partial\widetilde{\nu}}{\partial x_j}\frac{\partial\widehat{\widetilde{\nu}}}{\partial x_j}+(D-P)\widehat{\widetilde{\nu}}+(\widehat{D}-\widehat{P})\widetilde{\nu}=0\end{aligned} \tag{4.12c}$$

και 2.5 οι οποίες δίνουν

$$\widehat{\mathcal{R}}^{q}=-\frac{\partial\widehat{u}_j}{\partial x_j}=0 \tag{4.13a}$$

$$\begin{aligned}\widehat{\mathcal{R}}^{u}_{i}=&\widehat{u}_j\frac{\partial v_j}{\partial x_i}+u_j\frac{\partial\widehat{v}_j}{\partial x_i}-\widehat{v}_j\frac{\partial\widehat{u}_i}{\partial x_j}-v_j\frac{\partial\widehat{u}_i}{\partial x_j}-\frac{\partial\widehat{\tau}^{\alpha}_{ij}}{\partial x_j}+\frac{\partial\widehat{q}}{\partial x_i}+\widehat{\widetilde{\nu}}_\alpha\frac{\partial\widetilde{\nu}}{\partial x_i}+\widetilde{\nu}_\alpha\frac{\partial\widehat{\widetilde{\nu}}}{\partial x_i}\\&-\frac{\partial}{\partial x_l}\left(\frac{1}{\omega}\left[\left(\widehat{\widetilde{\nu}}_\alpha\widetilde{\nu}+\widetilde{\nu}_\alpha\widehat{\widetilde{\nu}}\right)C_\omega+\widetilde{\nu}_\alpha\widetilde{\nu}\left(\widehat{C}_\omega-\frac{C_\omega\widehat{\omega}}{\omega}\right)\right]\epsilon_{qjk}\frac{\partial v_k}{\partial x_j}\epsilon_{qli}\right)\\&-\frac{\partial}{\partial x_l}\left(\widetilde{\nu}_\alpha\widetilde{\nu}\frac{C_\omega}{\omega}\epsilon_{qjk}\frac{\partial\widehat{v}_k}{\partial x_j}\epsilon_{qli}\right)=0\quad i=1,2(,3)\end{aligned} \tag{4.13b}$$

$$\begin{aligned}\widehat{\mathcal{R}}^{\widetilde{\nu}_\alpha}=&-\widehat{v}_j\frac{\partial\widetilde{\nu}_\alpha}{\partial x_j}-v_j\frac{\partial\widehat{\widetilde{\nu}}_\alpha}{\partial x_j}-\frac{\partial}{\partial x_j}\left[\left(\frac{\widehat{\nu}+\widehat{\widetilde{\nu}}}{\sigma}\right)\frac{\partial\widetilde{\nu}_\alpha}{\partial x_j}\right]-\frac{\partial}{\partial x_j}\left[\left(\frac{\nu+\widetilde{\nu}}{\sigma}\right)\frac{\partial\widehat{\widetilde{\nu}}_\alpha}{\partial x_j}\right]\\&+\frac{1}{\sigma}\frac{\partial\widehat{\widetilde{\nu}}_\alpha}{\partial x_j}\frac{\partial\widetilde{\nu}}{\partial x_j}+\frac{1}{\sigma}\frac{\partial\widetilde{\nu}_\alpha}{\partial x_j}\frac{\partial\widehat{\widetilde{\nu}}}{\partial x_j}+\frac{2c_{b_2}}{\sigma}\frac{\partial}{\partial x_j}\left(\widehat{\widetilde{\nu}}_\alpha\frac{\partial\widetilde{\nu}}{\partial x_j}+\widetilde{\nu}_\alpha\frac{\partial\widehat{\widetilde{\nu}}}{\partial x_j}\right)\\&+\left(\widehat{\widetilde{\nu}}_\alpha\widetilde{\nu}+\widetilde{\nu}_\alpha\widehat{\widetilde{\nu}}\right)C_{\widetilde{\nu}}+\widetilde{\nu}_\alpha\widetilde{\nu}\widehat{C}_{\widetilde{\nu}}\\&+\frac{\partial\widehat{\nu}_t}{\partial\widetilde{\nu}}\frac{\partial u_i}{\partial x_j}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)+\frac{\partial\nu_t}{\partial\widetilde{\nu}}\frac{\partial\widehat{u}_i}{\partial x_j}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)\\&+\frac{\partial\nu_t}{\partial\widetilde{\nu}}\frac{\partial u_i}{\partial x_j}\left(\frac{\partial\widehat{v}_i}{\partial x_j}+\frac{\partial\widehat{v}_j}{\partial x_i}\right)+(\widehat{D}-\widehat{P})\widetilde{\nu}_\alpha+(D-P)\widehat{\widetilde{\nu}}_\alpha=0\end{aligned} \tag{4.13c}$$

$$\widehat{\mathcal{R}}^{\Delta_\alpha}=-2\frac{\partial}{\partial x_j}\left(\widehat{\Delta}_\alpha\frac{\partial\Delta}{\partial x_j}\right)+\left(\widehat{\widetilde{\nu}}\widetilde{\nu}_\alpha+\widetilde{\nu}\widehat{\widetilde{\nu}}_\alpha\right)C_\Delta+\widetilde{\nu}\widetilde{\nu}_\alpha\widehat{C}_\Delta=0 \tag{4.13d}$$

Όλοι οι υπόλοιποι όροι στις εξισώσεις 4.12 και 4.13 και οι αντίστοιχες BCs τους μπορούν να βρεθούν στο [24].

Το κόστος ανά κύκλο βελτιστοποίησης των pFOSM και pAPCE αναλύεται στο [24]. Υποθέτοντας ότι υπάρχουν n_α συστήματα συζυγών εξισώσεων που πρέπει να επιλυθούν, το κόστος ανά κύκλο βελτιστοποίησης με την pFOSM είναι ίσο με $1+$

$3n_\alpha$ EFS (1 πρωτεύον πρόβλημα, n_α συστήματα συζυγών εξισώσεων, $2n_\alpha$ συστήματα διαφορισμένων και έπειτα προβεβλημένων πρωτευουσών και συζυγών εξισώσεων). Με την pAPCE, το κόστος είναι ίσο με $(1+3n_\alpha)L$.

4.2 Βελτιστοποίηση Μορφής της Μεμονωμένης Fauvel 14% Αεροτομής υπό Αβεβαιότητες

Εδώ παρουσιάζεται η επέκταση των pFOSM και pAPCE σε τυρβώδεις ροές. Γίνεται βελτιστοποίηση μορφής της μεμονωμένης Fauvel 14% αεροτομής [28] για μείωση των στατιστικών ροπών του C_D υπό V και μ_{C_L} μεγαλύτερα ή ίσα των αρχικών τους τιμών. Οι μεταβλητές αβεβαιότητας ήταν το μέτρο της ταχύτητας v_∞ και η γωνία α_∞ της αδιατάρακτης ροής. Η ροή είχε $Re = 10^6$, βάσει της μ_{v_∞} και το πλέγμα αποτελούνταν από περίπου 48000 κελιά. Η αεροτομή παραμετροποιήθηκε με VBS, το κουτί παραμετροποίησης φαίνεται στο σχήμα fig. 4.1. Για τις τιμές αβεβαιότητας, $\mu_{v_\infty} = 15\ m/s$ και $\mu_{\alpha_\infty} = 8^\circ$ και οι τυπικές αποκλίσεις $\sigma_{v_\infty} = 0.75\ m/s$ και $\sigma_{\alpha_\infty} = 0.8^\circ$, αντίστοιχα. Ακολουθώντας μια παραμετρική μελέτη όμοια με αυτήν στο [24], επιλέχθηκαν οι παρακάτω παράμετροι για την pAPCE, $O_{PCE}=2$ και $L=8$.

Στο σχήμα 4.2, φαίνονται οι SDs των στατιστικών ροπών του C_D υπολογισμένες με τις pFOSM και pAPCE. Αυτές συγκρίνονται με Πεπερασμένες Διαφορές (Finite Differences, FDs) και βρίσκονται σε καλή συμφωνία. Οι μεταβλητές σχεδιασμού είναι οι συντεταγμένες των CPs των VBS. Για κάθε αυξομείωση των CPs στις FDs, επιλύεται ένα πρόβλημα UQ είτε με τη FOSM είτε με την APCE. Το σχήμα 4.2 δείχνει και SDs υπολογισμένες με την pAPCE αγνοώντας όμως πλήρως τις μικτές παραγώγους στην εξ. 4.10. Όπως φαίνεται, αυτή η παράλειψη είναι αβλαβής όταν υπολογίζονται SDs της μέσης τιμής των QoIs ενώ ταυτόχρονα μειώνει το κόστος υπολογισμού SDs της μέσης τιμής με την pAPCE στο μισό. Παρόμοια συμπεράσματα προκύπτουν και για τις SDs άλλων αεροδυναμικών συντελεστών οι οποίες όμως παραλείπονται.

Πραγματοποιήθηκαν 3 κύκλοι βελτιστοποίησης με στόχο τη μείωση των μ_{C_D}, $\mu_{C_D} + 2\sigma_{C_D}$ και $\mu_{C_D} + 6\sigma_{C_D}$, σε κάθε περίπτωση. Τα αποτελέσματα φαίνονται στο σχήμα 4.3. Οι βελτιστοποιημένες αεροτομές φαίνονται στο σχήμα 4.4. Για την περίπτωση όπου $J = \mu_{C_D} + 6\sigma_{C_D}$, οι δύο μέθοδοι παράγουν διαφορετικά σχήματα.

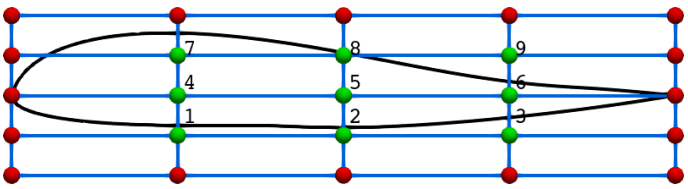


Figure 4.1: Fauvel 14% αεροτομή. Περίγραμμα της αεροτομής και το 5×5 κουτί παραμετροποίησης VBS. Κόκκινα/πράσινα CPss μένουν σταθερά/κινούνται κατά τη βελτιστοποίηση.

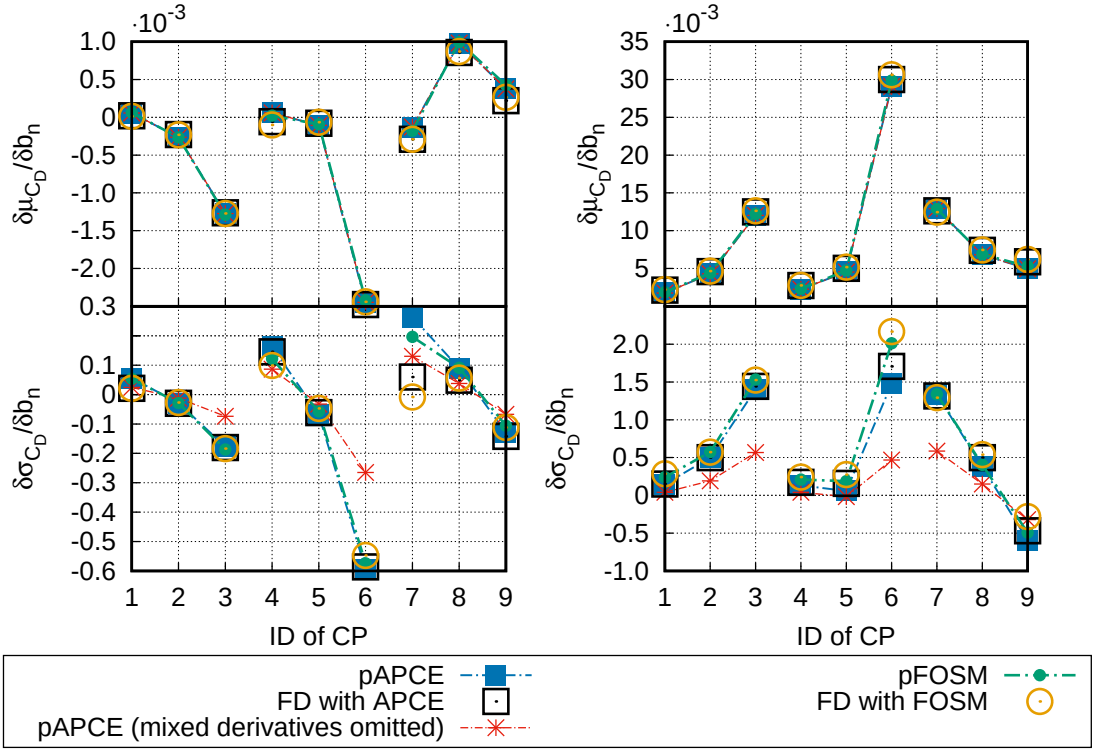


Figure 4.2: Fauvel 14% αεροτομή. SDs των στατιστικών ροπών του C_D ως προς τις x (αριστερά) και y (δεξιά) συντεταγμένες των CPs (αριθμημένα στο σχήμα fig. 4.1). **Άνω**: μ_{C_D}. **Κάτω**: σ_{C_D}. Το μέγεθος του βήματος κατά τις FDs ήταν ίσο με $10^{-6}m$.

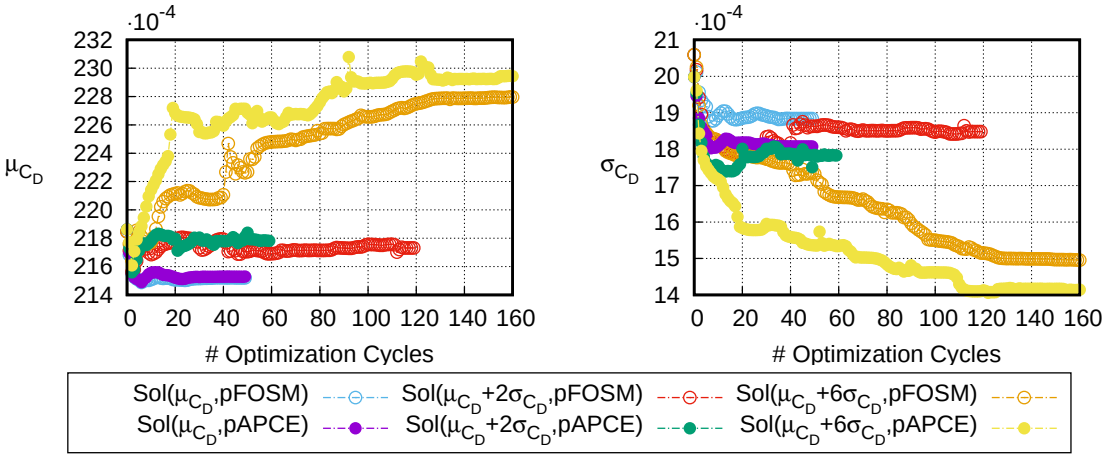


Figure 4.3: Fauvel 14% αεροτομή. Πορεία των βελτιστοποιήσεων με τις pFOSM και pAPCE ($O_{PCE}=2$ and $L=8$). **Αριστερά**: μ_{C_D}. **Δεξιά**: σ_{C_D}.

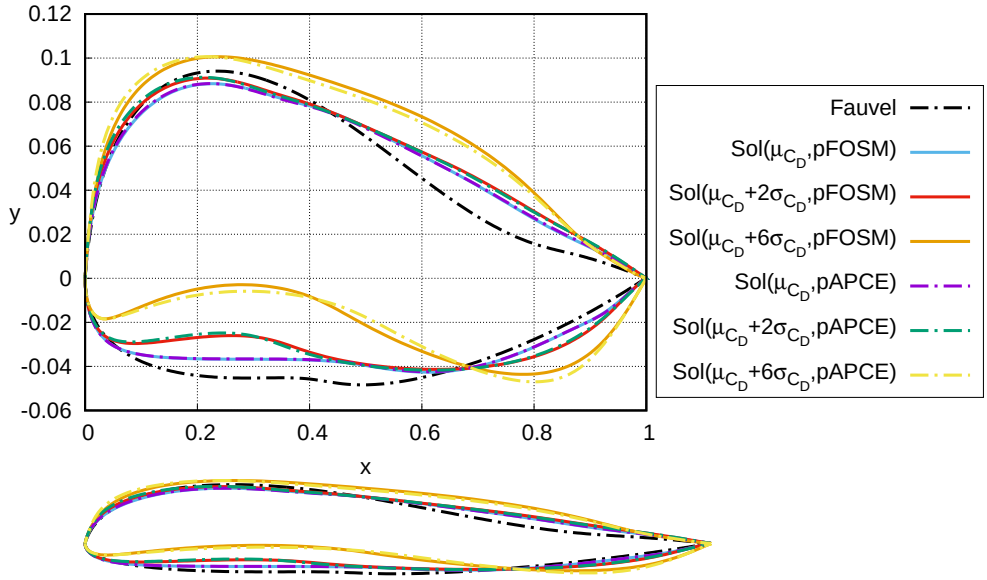


Figure 4.4: Fauvel 14% αεροτομή. Περιγράμματα της αρχικής και των βελτιστοποιημένων αεροτομών με τις pFOSM and pAPCE ($O_{PCE} = 4$ και $L = 8$) με συναρτήσεις στόχου $J = \mu_{C_D}$, $J = \mu_{C_D} + 2\sigma_{C_D}$ και $J = \mu_{C_D} + 6\sigma_{C_D}$.

Κεφάλαιο 5

Βιομηχανικές Εφαρμογές

Το κεφάλαιο αυτό παρουσιάζει την εφαρμογή των μεθόδων που αναπτύχθηκαν στα Κεφάλαια 3 και 4 σε εφαρμογές βιομηχανικού τύπου. Στις ενότητες 5.1 και 5.2, η RPM χρησιμοποιείται για να βοηθήσει τη σύγκλιση πρωτεύοντων και συζυγών επιλυτών στη βελτιστοποίηση μορφής σε εφαρμογές για την αυτοκινητοβιομηχανία. Ακολουθούν επαναξιολογήσεις των λύσεων με URANS επιλύτες. Στην ενότητα 5.3, οι pFOSM and pAPCE χρησιμοποιούνται για βελτιστοποίηση μιας πτέρυγας με συμμετρικη διατομή με σκοπό τη μείωση του C_D κατανέμοντας τη στρέψη στην πτέρυγα κατά το εκπέτασμά της.

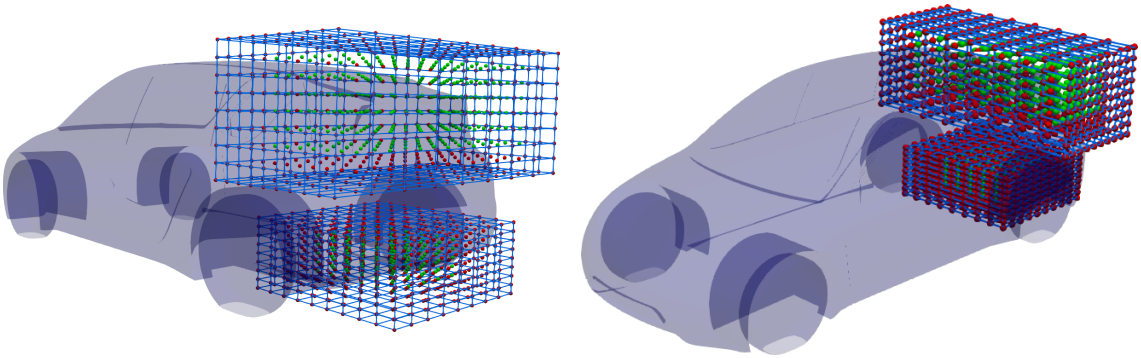


Figure 5.1: Βελτιστοποίηση μορφής επιβατικού οχήματος. Γεωμετρία του οχήματος και 2 κουτιά παραμετροποίησης $10 \times 9 \times 9$ VBS. Τα πράσινα CPs κινούνται ενώ τα κόκκινα μένουν σταθερά.

5.1 Βελτιστοποίηση Μορφής Επιβατικού Οχήματος

Το όχημα που φαίνεται στο σχήμα fig. 5.1 παραμετροποιήθηκε με VBSs και βελτιστοποιήθηκε για μείωση του C_D. Είναι $Re \approx 8.8 \cdot 10^6$, βάσει του μήκους του οχήματος, το πλέγμα περιείχε $1.2 \cdot 10^6$ κελιά και η ταχύτητα του οχήματος ήταν ίση με $33m/s$.

Η RPM σταθεροποίησε τους επιλύτες πρωτευουσών και συζυγών εξισώσεων, οι οποίοι έως πρότινως απέκλιναν. Για τον συζυγή επιλύτη, χρησιμοποιήθηκε συνδυαστικά ήπια απόσβεση της ATC, επηρεάζοντας μόνο τα πρώτα 2 κελιά μπροστά από τα στερεά τοιχώματα. Μέσα σε 11 κύκλους βελτιστοποίησης, το C_D μειώθηκε κατά 4.8%, όπως φαίνεται στο σχήμα 5.2. Για τις 3 από τις 11 γεωμετρίες, έγινε επαναξιολόγηση των λύσεων με URANS επιλύτες, οι οποίοι φανέρωσαν μείωση του $\overline{C_D}$ κατά 4.9%, πολύ κοντά στην εκτίμηση που έδωσε ο σταθεροποιημένος μόνιμος επιλύτης. Οι βελτιστοποιημένες μορφές φαίνονται στο σχήμα 5.3. Για 11 κύκλους βελτιστοποίησης, χρειάστηκαν 4.5 ώρες σε 156 επεξεργαστές (Intel© Xeon© CPU E5-2620 v2, 2.10GHz) με σταθεροποιημένους από την RPM μόνιμους επιλύτες. Ενδεικτικά, για το παράθυρο ολοκλήρωσης που φαίνεται στο σχήμα 5.2, χρειάστηκαν 15 ώρες προκειμένου να γίνει μια φορά η επίλυση των URANS εξισώσεων στους ίδιους επεξεργαστές (πλήθος και τύπος).

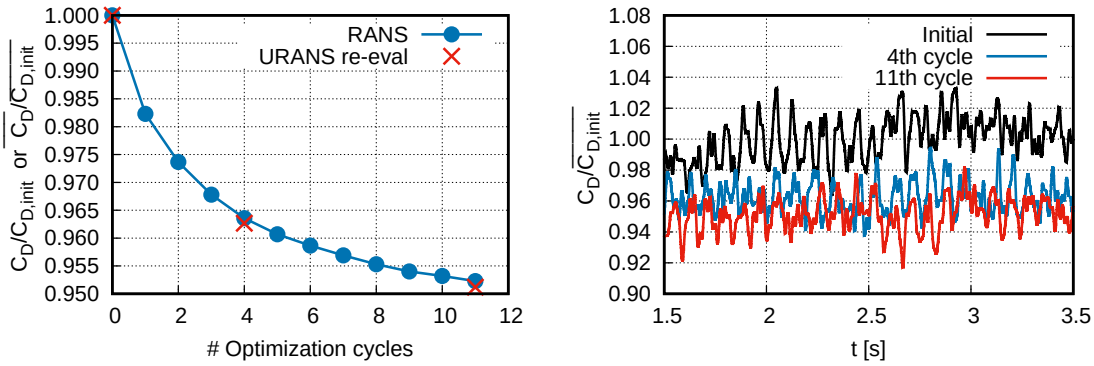


Figure 5.2: Βελτιστοποίηση μορφής επιβατικού οχήματος. **Αριστερά**: Μείωση του C_D μέσα σε 11 κύκλους βελτιστοποίησης και $\overline{C_D}$ υπολογισμένος με URANS επιλύτες. **Δεξιά**: Στιγμιαίο C_D εντός του παραθύρου ολοκλήρωσης των URANS επαναξιολογήσεων. Όλες οι τιμές είναι διαιρεμένες με τις αντίστοιχες αρχικές.

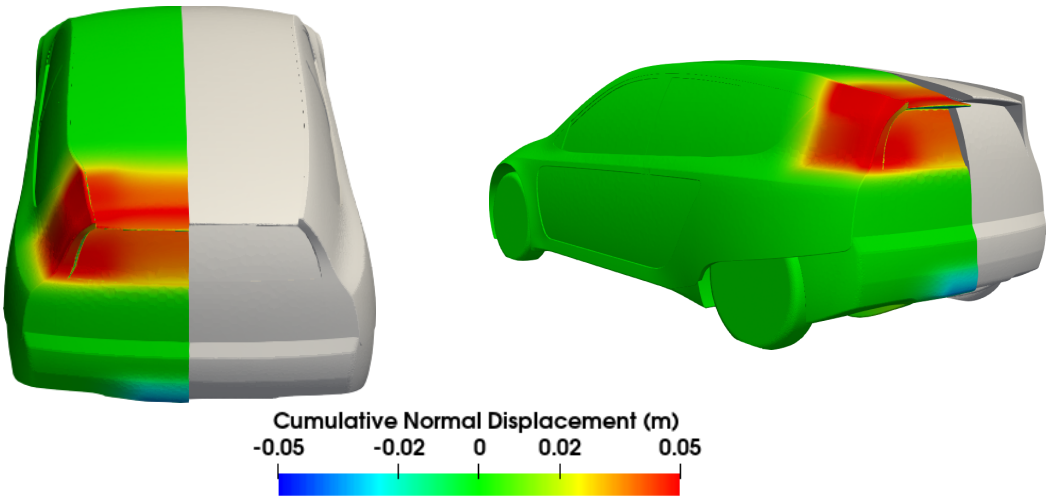


Figure 5.3: Βελτιστοποίηση μορφής επιβατικού οχήματος. Σύγκριση μεταξύ της αρχικής (γκρι) και τελικής γεωμετρίας του οχήματος. Η τομή γίνεται στον άξονα συμμετρίας. Στο βελτιστοποιημένο σχήμα φαίνονται οι μετατοπίσεις των κόμβων του επιφανειακού πλέγματος, με κόκκινο προς το εσωτερικό, με μπλε προς το εξωτερικό του οχήματος.

5.2 Βελτιστοποίηση Μορφής Ανεμοθώρακα Μοτοσυκλέτας

Πραγματοποιείται βελτιστοποίηση μορφής του ανεμοθώρακα μιας μοτοσυκλέτας για μείωση του $C_D A$ (όπου A είναι η εμπρόσθια επιφάνεια) της μοτοσυκλέτας. Ο αριθμός $Re \approx 2.6 \cdot 10^6$, το πλέγμα αποτελούνταν από $14 \cdot 10^6$ κελιά και χρησιμοποιήθηκε ένα 7×7×7 κουτί παραμετροποίησης με VBS για την παραμετροποίηση του ανεμοθώρακα, σχήμα 5.4.

Η περιπλοκότητα της γεωμετρίας και της ροής έκανε δύσκολη τη σύγκλιση του πρωτεύοντος επιλύτη, ακόμη και με την RPM. Για αυτόν τον λόγο, υπολογίστηκαν οι μέσες τιμές των πεδίων του μετρίως συγκλιμένου πρωτεύοντος επιλύτη και χρησιμοποιήθηκαν για την επίλυση των συζυγών εξισώσεων. Ο συζυγής επιλύτης σταθεροποιήθηκε με χρήση της RPM και της ήπιας απόσβεσης της ATC, κατά την οποία επηρεάστηκαν μόνο τα 5 πρώτα κελιά μπροστά από το στερεό τοίχωμα. Έτσι, το $C_D A$ μειώθηκε κατά 4.6%, σχήμα 5.5 μετά από 9 κύκλους βελτιστοποίησης. Οι αλλαγές στη γεωμετρία φαίνονται στο σχήμα 5.6.

Οι 9 κύκλοι βελτιστοποίησης, χρησιμοποιώντας την RPM και ήπια απόσβεση της ATC, χρειάστηκαν 20 ώρες σε 156 επεξεργαστές (Intel© Xeon© CPU E5-2620 v2, 2.10GHz). Αυτό ήταν περίπου το κόστος εκτέλεσης ενός μόνο κύκλου βελτιστοποίησης χρησιμοποιώντας μη-μόνιμους επιλύτες, υποθέτοντας πως όλα τα πρωτεύοντα πεδία αποθηκεύονταν στη μνήμη, χωρίς επιπλέον κόστος, σύμφωνα με τον τρόπο που παρουσιάστηκε στους [11]. Και στις δύο περιπτώσεις των ενοτήτων 5.2 και 5.1, το υπολογιστικό κόστος βελτιστοποίησης με μόνιμους επιλύτες, σταθεροποιημένους με την RPM, ήταν τουλάχιστον μία τάξη μεγέθους μικρότερο από ότι θα απαιτούσε η χρήση μη-μόνιμων επιλυτών.

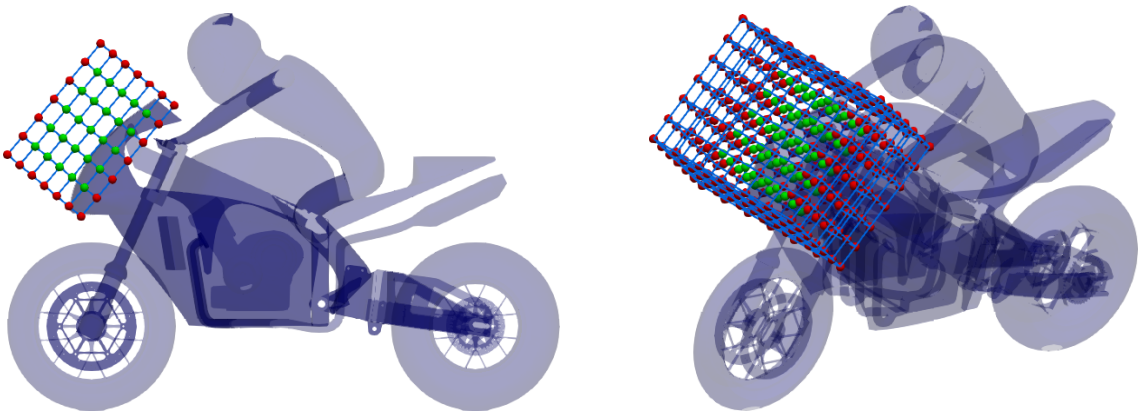


Figure 5.4: Βελτιστοποίηση μορφής ανεμοθώρακα μοτοσυκλέτας. Γεωμετρία και κουτί παραμετροποίησης. Μόνο τα πράσινα CPs μετακινούνταν κατά τη βελτιστοποίηση.

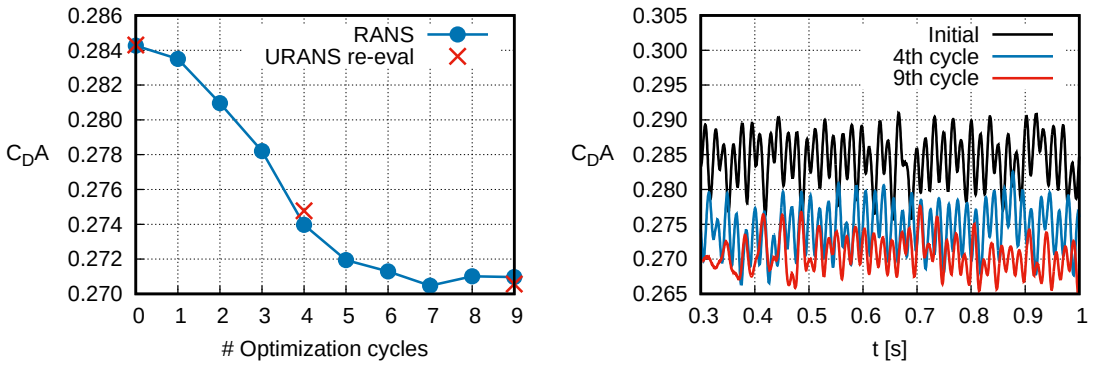


Figure 5.5: Βελτιστοποίηση μορφής ανεμοθώρακα μοτοσυκλέτας. **Αριστερά**: Σύγκλιση του $C_D A$ κατά τη βελτιστοποίηση. Το $\overline{C_D A}$ επαναξιολογήθηκε από μη-μόνιμους επιλύτες στις αρχικές γεωμετρίες και αυτές του 4ου και 9ου κύκλου βελτιστοποίησης. **Δεξιά**: Διάγραμμα χρονοσειράς του $C_D A$ εντός του παραθύρου ολοκλήρωσης κατά τις 3 επαναξιολογήσεις με URANS επιλύτες.

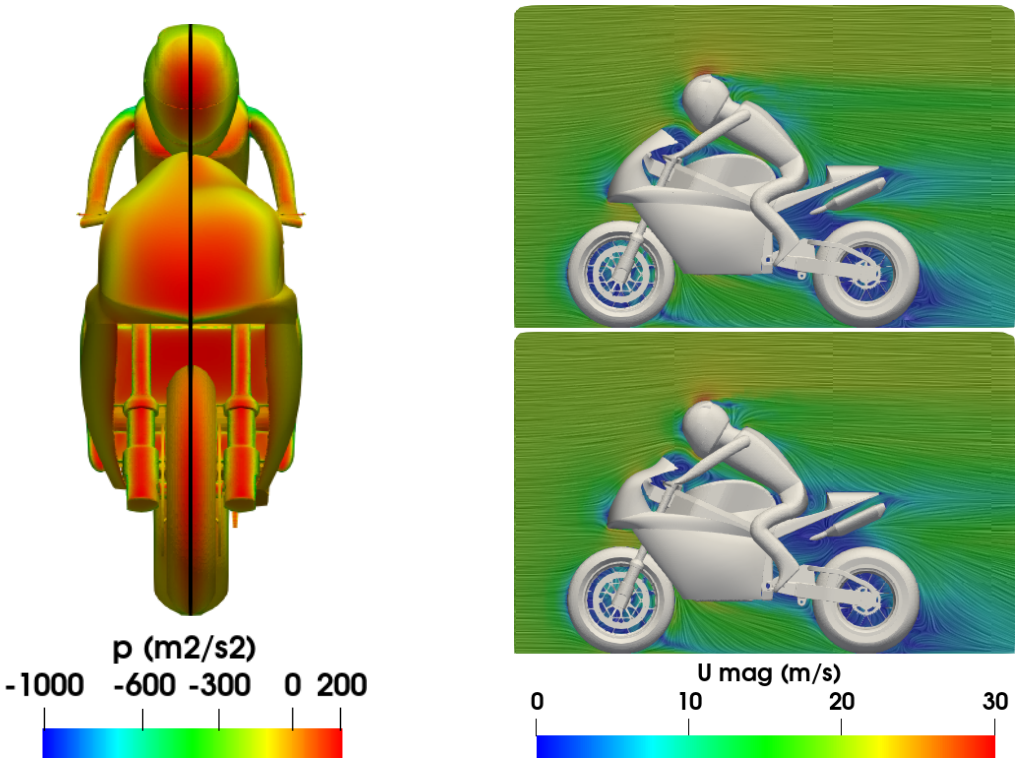


Figure 5.6: Βελτιστοποίηση μορφής ανεμοθώρακα μοτοσυκλέτας. **Αριστερά**: Κατανομή πίεσης στην εμπρόσθια όψη της αρχικής (αριστερά) και βελτιστοποιημένης (δεξιά) μοτοσυκλέτας. **Δεξιά**: Μέση τιμή της ταχύτητας ροής στο επίπεδο συμμετρίας της γεωμετρίας γύρω από την αρχική και βελτιστοποιημένη γεωμετρία.

5.3 Βελτιστοποίηση Μορφής Πτέρυγας υπό Αβεβαιότητες

Αυτή η εφαρμογή βασίζεται σε μία από τα παραδείγματα του AIAA Aircraft Design Optimization Discussion Group (ADODG) (η περίπτωση 3 [5]), τα οποία αναπτύχθηκαν με σκοπό τη δημιουργία προβλημάτων αναφοράς για κώδικες βελτιστοποίησης. Στόχος είναι η μείωση του C_D αποκλειστικά μέσω της κατανομής της στρέψης της πτέρυγας γύρω από άξονα που διέρχεται από την ακμή εκφυγής. Εδώ, η βελτιστοποίηση γίνεται υποθέτοντας πως οι μεταβλητές v_∞ και α_∞ ακολουθούν κανονικές κατανομές με $\mu_{v_\infty} = 50m/s$, $\sigma_{v_\infty} = 10m/s$ και $\mu_{\alpha_\infty} = 5°$, $\sigma_{\alpha_\infty} = 1°$. Χρησιμοποιήθηκαν οι pFOSM και pAPCE για τη βελτιστοποίηση.

Η γεωμετρία και τα CPs του κουτιού παραμετροποίησης φαίνονται στο σχήμα 5.7. Είναι $Re = 1.1 \cdot 10^6$ και η επιφάνεια αδιαστατοποίησης της οπισθέλκουσας για τον υπολογισμό του C_D ήταν $A_c = 0.27$ [5].

Η βελτιστοποίηση μορφής με τις pFOSM και pAPCE έγινε για τιμές w (εξ. 4.1) από 0 έως 6 με βήμα 1. Για την pAPCE, $O_{PCE} = 3$ και $L = 8$. Χρησιμοποιήθηκε η μέθοδος Sequential Quadratic Programming [15] για την εφαρμογή ενός περιορισμού στην τιμή του μ_{C_L}, η οποία τέθηκε ίση με την τιμή του στην αρχική πτέρυγα. Στο σχήμα 5.8 φαίνονται οι βελτιστοποιημένες και με τις δύο μεθόδους πτέρυγες για w ίσο με 0, 3 και 6. Για όλες τις βελτιστοποιημένες λύσεις, έγινε αξιολόγηση τους με τη μέθοδο Monte Carlo χρησιμοποιώντας 1000 δείγματα. Τα

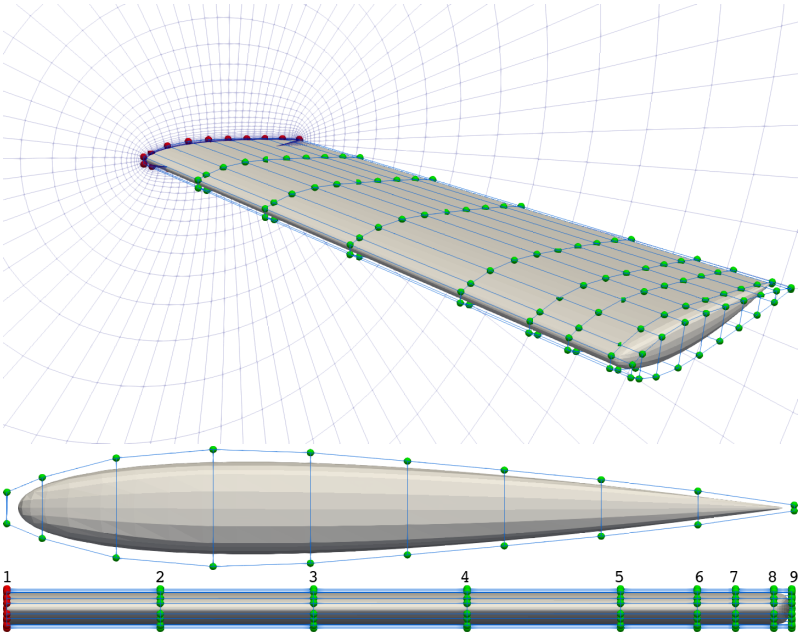


Figure 5.7: Βελτιστοποίηση μορφής της ADODG3 πτέρυγας. Κουτί παραμετροποίησης με VBS και όψεις της πτέρυγας. Κάθε επίπεδο από CPs περιστρέφεται γύρω από άξονα που διέρχεται από την ακμή εκφυγής της πτέρυγας. Τα κόκκινα CPs που βρίσκονται επάνω στο επίπεδο συμμετρίας παραμένουν ακίνητα.

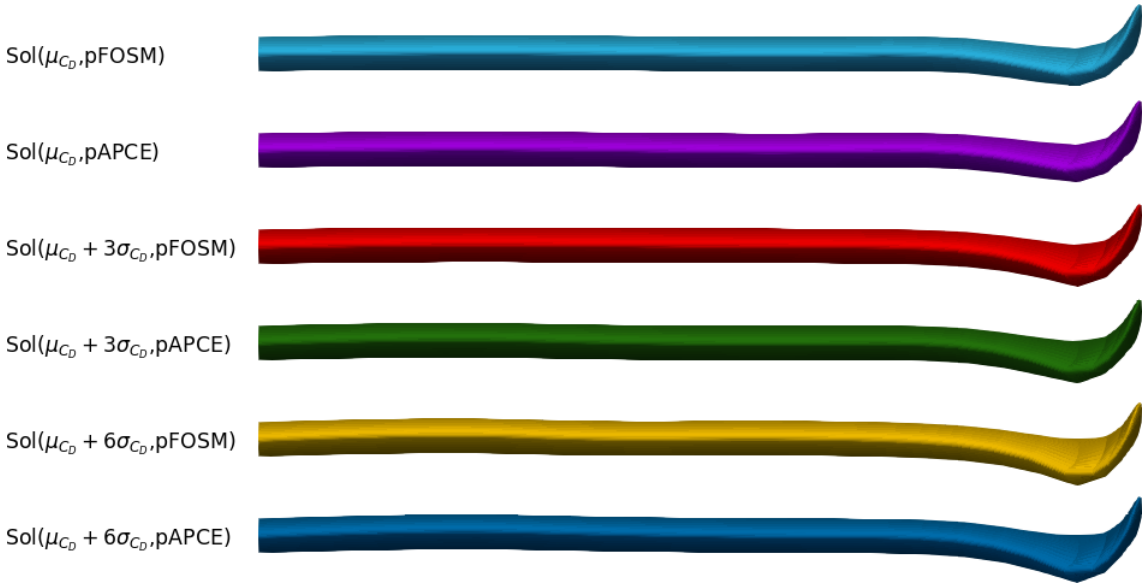


Figure 5.8: Βελτιστοποίηση μορφής της ADODG3 πτέρυγας. Εμπρόσθια όψη των βελτιστοποιημένων πτερυγίων υπολογισμένων με τις pFOSM και pAPCE με συναρτήσεις στόχου $J = \mu_{C_D}$ και $J = \mu_{C_D} + 3\sigma C_D$.

αποτελέσματα φαίνονται στο σχήμα 5.9 όπου σχηματίζουν ένα μέτωπο Pareto στο επίπεδο $\mu_{C_D} - \sigma_{C_D}$. Το μέτωπο που παράγει η pAPCE είναι ευρύτερο και δύο από τις λύσεις κυριαρχούν επί των αντίστοιχων που παρήγαγε η pFOSM. Θα χρειάζονταν περισσότερα σημεία προκειμένου να εξαχθούν συμπεράσματα για όλες τις περιοχές του μετώπου. Το κόστος βελτιστοποίησης ήταν ίσο με 5 EFS με την pFOSM και 40 EFS με την pAPCE.

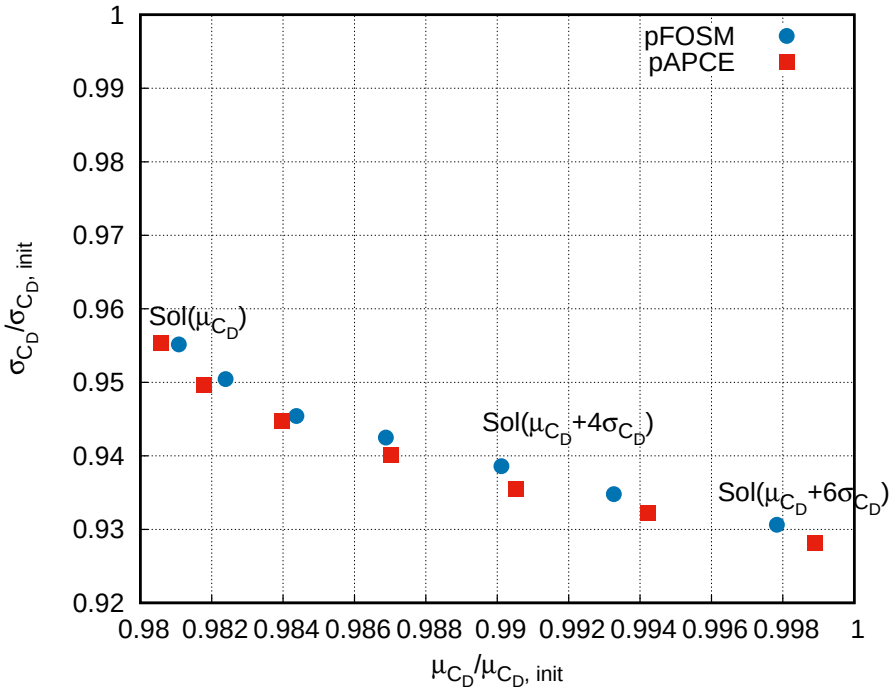


Figure 5.9: Βελτιστοποίηση μορφής της ADODG3 πτέρυγας. Μέτωπο Pareto των βελτιστοποιημένων λύσεων στο επίπεδο $\mu_{C_D} - \sigma_{C_D}$. Τα δύο μέτωπα υπολογίστηκαν με τις pFOSM και pAPCE όμως τα σημεία στο διάγραμμα επαναξιολογήθηκαν με τη μέθοδο Monte Carlo (1000 δείγματα).

Κεφάλαιο 6

Σύνοψη-Συμπεράσματα

Η διατριβή επεκτείνει προηγούμενες μεθόδους αεροδυναμικής βελτιστοποίησης μορφής που στηρίζονται στη συνεχή συζυγή μέθοδο. Οι συνεισφορές αφορούν δύο διακριτούς τομείς:

- Την αντιμετώπιση δυσκολιών σύγκλισης των μόνιμων πρωτεύοντων (ροϊκών) και (συνεχών) συζυγών επιλυτών, οι οποίες προκαλούνται από τη χρονική αστάθεια της ροής.

- Τη βελτιστοποίηση υπό αβεβαιότητες.

Οι μέθοδοι που αναπτύχθηκαν έχουν ως κοινό παρονομαστή τη μείωση του υπολογιστικού κόστους της βελτιστοποίησης. Οι μέθοδοι είναι γενικής χρήσης, δεν περιορίζονται σε προβλήματα βελτιστοποίησης μορφής και μπορούν κάλλιστα να χρησιμοποιηθούν σε άλλους τομείς βελτιστοποίησης με τη CFD, όπως, βελτιστοποίηση τοπολογίας. Επίσης, παρότι η παρουσίαση των μεθόδων έγινε ξεχωριστά, οι μέθοδοι μπορούν να συνδυαστούν.

6.1 Στοιχεία Πρωτοτυπίας

- Η μείωση του υπολογιστικού κόστους της βελτιστοποίησης μορφής για ασυμπίεστες ροές που παρουσιάζουν ήπια χρονική αστάθεια μέσω της αντιμετώπισης των δυσκολιών σύγκλισης των μόνιμων πρωτευουσών και συζυγών εξισώσεων με χρήση της RPM. Επίσης, δείχνεται πώς μπορεί να μειωθεί το κόστος ενός βρόχου βελτιστοποίησης, βασισμένου σε χρονικά μη-μόνιμους επιλύτες, σε ροές όπου εκλύονται στρόβιλοι, αν προηγηθεί βελτιστοποίηση με σταθεροποιημένους μόνιμους επιλύτες.

- Έγινε αξιολόγηση διαδεδομένων τεχνικών που χρησιμοποιούνται για την υπέρβαση των δυσκολιών σύγκλισης των πρωτεύοντων και συζυγών επιλυτών ως προς την ακρίβεια των SDs. Αυτές είναι η χρήση πρωτεύοντων "μέσων πεδίων" και η απόσβεση της ATC. Προέκυψαν τα εξής πρακτικά πορίσματα από τη χρήση της RPM: Η ήπια απόσβεση της ATC, η οποία είναι αποτελεσματική μόνο σε συνέργεια με την RPM, είναι αβλαβής, ενώ η ανεξέλεγκτη απόσβεση πρέπει να αποφεύγεται. Η χρήση των πρωτεύοντων "μέσων πεδίων" δεν φαίνεται να έχει τόσο αρνητικό αποτέλεσμα, όμως, μπορεί να αποφευχθεί εντελώς σε πολλές περιπτώσεις με την RPM.

- Η ανάπτυξη και παρουσίαση, για πρώτη φορά στη βιβλιογραφία, μιας νέας μεθόδου, που ονομάζεται pAPCE, για βελτιστοποίηση μορφής υπό αβεβαιότητες με χαμηλότερο κόστος συγκρινόμενη με το κλασικό μη-παρεμβατικό PCE που βασίζεται σε παλινδρόμηση. Το χαμηλότερο κόστος επιτυγχάνεται χάρη στη συμπερίληψη SDs της QoI ως προς τις μεταβλητές αβεβαιότητας, υπολογισμένες με τη συζυγή μέθοδο, στην παλινδρόμιση, δίχως όμως να γίνεται δαπανηρός ο υπολογισμός των SDs των στατιστικών ροπών. Αυτό επιτυγχάνεται με τον συνδυασμό της συζυγούς μεθόδου και της ευθείας διαφόρισης με μια τεχνική προβολής.

- Η επέκταση των δύο μεθόδων pAPCE και pFOSM σε τυρβώδεις ροές, οι οποίες επιλύονται με χρήση του μοντέλου τύρβης των Spalart-Allmaras. Για τον σκοπό αυτό, τόσο η PDE του μοντέλου όσο και η συζυγής της διαφορίστηκαν ως προς τις μεταβλητές αβεβαιότητας (που σχετίζονται με τις συνθήκες ροής).

6.2 Δημοσιεύσεις και Παρουσιάσεις σε Διεθνή Επιστημονικά Περιοδικά και Συνέδρια

Δημοσιεύσεις σε επιστημονικά περιοδικά:

- T. Skamagkis, E.M. Papoutsis-Kiachagias, K.C. Giannakoglou. "On the Stabilization of Steady Continuous Adjoint Solvers in the Presence of Unsteadiness, in Shape Optimization". International Journal of Numerical Methods in Fluids 2021; 93(8):2677-2693

- Th. Skamagkis, E.M. Papoutsis-Kiachagias, K. Giannakoglou. "CFD-based shape optimization under uncertainties using the Adjoint-assisted Polynomial Chaos Expansion and projected derivatives". Computers & Fluids 2022; 241:105458

Παρουσιάσεις σε συνέδρια:

- T. Skamagkis, A.-S.I. Margetis, E.M. Papoutsis-Kiachagias and K.C. Giannakoglou. "On the efficiency and robustness of the adjoint method: Applications in steady and unsteady shape optimization in fluid mechanics". 8th OpenFOAM Conference, Digital Event, October 13-15, 2020.

- E.M. Papoutsis-Kiachagias, M. Farhikhteh, T. Skamagkis and K.C. Giannakoglou. "Aerodynamic Shape Optimization of the MEXICO Wind Turbine Blade using the Continuous Adjoint Method". EUROGEN 2021, 14th International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control, online from Athens, Greece, June 28-30, 2021.

- E.M. Papoutsis-Kiachagias, K.T. Gkaragkounis, A.-S.I. Margetis, T. Skamagkis, V.G. Asouti and K.C. Giannakoglou. "*adjointOptimisationFoam*: An OpenFOAM-based Framework for Adjoint-assisted Optimization". EUROGEN 2021, 14th International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control, online from Athens, Greece, June 28-30, 2021.

- Th. Skamagkis, E.M. Papoustis-Kiachagias, K.C. Giannakoglou. "Adjoint-Based Aerodynamic Shape Optimization for Turbulent Flows in the Presence of Uncertainties Using the Method of Moments and Projections". 17th OpenFOAM Workshop, Cambridge, UK, 11-14 July, 2022.

- Th. Skamagkis, E.M. Papoutsis-Kiachagias, K.C. Giannakoglou. "Aerodynamic Shape Optimization Under Uncertainties using

Adjoint-assisted PCE and Projections." ASMO-UK 12 / ASMO-Europe 1 / ISSMO Conference on Engineering Design Optimization, Leeds, UK, 18-19 July, 2022.

6.3 Προτάσεις για Μελλοντική Εργασία

Για την RPM:

- Περαιτέρω αυτοματοποίηση της επιλογής παραμέτρων.
- Συνδυασμός της RPM με έναν συζευγμένο πρωτεύοντα ή συζευγμένο συζυγή επιλύτη για αυξημένη ευστάθεια και ευκολία στον υπολογισμό του ασταθούς υποχώρου με απευθείας χρήση του Ιακωβιανού μητρώου.

Για την τεχνική προβολής:

- Συνδυασμός της τεχνικής προβολής με τη μέθοδο δεύτερων ροπών με ακρίβεια δεύτερης τάξης (Second-order second-moment, SOSM) όπου εμφανίζονται όροι ανώτερης τάξης.
- Διερεύνηση του κατά πόσο η μέθοδος pAPCE μπορεί να επωφεληθεί από τη χρήση καλύτερων τεχνικών δειγματοληψίας.
- Συνδυασμός της τεχνικής προβολής με παραλλαγές της pAPCE που είναι διαθέσιμες στη βιβλιογραφία.
- Επέκταση της τεχνικής προβολής για χρήση με άλλα μοντέλα τύρβης οι συζυγείς επιλύτες των οποίων είναι διαθέσιμοι, είτε στην ανοιχτής πρόσβασης είτε στην οικεία, *adjointOptimisation* βιβλιοθήκη.

Βιβλιογραφία

[1] K. B. Fragkos, E. M. Papoutsis-Kiachagias, and K. C. Giannakoglou. pFOSM: an efficient algorithm for aerodynamic robust design based on continuous adjoint and matrix-vector products. *Computers and Fluids*, 181:57–66, 2019.

[2] K. Gkaragkounis. *The Continuous Adjoint Method in Aerodynamic and Conjugate Heat Transfer Shape Optimization, for Turbulent Flows*. PhD thesis, National Technical University of Athens, 2020.

[3] G. H. Golub and C. F. van Loan. *Matrix Computations*. JHU Press, Baltimore, 4th edition, 2013.

[4] A. Haldar and S. Mahadevan. *Probability, Reliability and Statistical Methods in Engineering Design*. John Wiley & Sons, New York/Chichester, UK, 1st edition, 2000, pages 195–218.

[5] J. E. Hicken. Aerodynamic Design Optimization Workshop: Twist Optimization Case. Rensselaer Polytechnic Institute, Dept. of Mechanical, Aerospace, and Nuclear Eng., Troy, NY, United States, 12180. https://sites.google.com/view/mcgill-computational-aerogroup/adodg.

[6] I. S. Kavvadias. *Continuous Adjoint Methods for Steady and Unsteady Turbulent flows with Emphasis on the Accuracy of Sensitivity Derivatives*. PhD thesis, National Technical University of Athens, 2015.

[7] G. Kenway, G. Kennedy, and J. R. R. A. Martins. A CAD-free approach to high-fidelity aerostructural optimization. In *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, AIAA 2010-9231, Fort Worth, Texas, United States, 2010.

[8] D. Kurtulus. On the unsteady behavior of the flow around NACA0012 airfoil with steady external conditions at Re=1000. *International Journal of Micro Air Vehicles*, 7:301–326, 2015.

[9] Y. Liu, K. Li, J. Zhang, H. Wang, and L. Liu. Numerical bifurcation analysis of static stall of airfoil and dynamic stall under unsteady perturbation. *Communications in Nonlinear Science and Numerical Simulation*, 17:3427–3434, 2012.

[10] K. Lust. *Numerical Bifurcation Analysis of Periodic Solutions of Partial Differential Equations*. PhD thesis, Department of Computer Science, K. U. Leuven, 1997.

[11] A. I. Margetis, E. M. Papoutsis-Kiachagias, and K. C. Giannakoglou. Lossy compression techniques supporting unsteady adjoint on 2D/3D unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 387, 114152, 2021.

[12] J. M. Martin, E. Andres, C. Lozano, and E. Valero. Volumetric B-Splines shape parametrization for aerodynamic shape design. *Aerospace Science and Technology*, 37:26–36, 2014.

[13] J. R. R. A. Martins and J. T. Hwang. Review and unification of methods for computing derivatives of multidisciplinary computational models. *AIAA Journal*, 51(11):2582–2599, 2013.

[14] M. McKay, R. Beckman, and W. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[15] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, NY, 1st edition, 1999.

[16] OpenFOAM software. www.openfoam.com.

[17] E. M. Papoutsis-Kiachagias. *Adjoint methods for turbulent flows, applied to shape or topology optimization and robust design*. PhD thesis, National Technical University of Athens, 2010.

[18] E. M. Papoutsis-Kiachagias, V. G. Asouti, K. C. Giannakoglou, K. Gkagkas, S. Shimokawa, and E. Itakura. Multi-point aerodynamic shape optimization of cars based on continuous adjoint. *Structural and Multidisciplinary Optimization*, 59(2):675–694, 2019.

[19] E. M. Papoutsis-Kiachagias and K. C. Giannakoglou. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: industrial applications. *Archives in Computational Methods in Engineering*, 32(2):255–299, 2016.

[20] E. M. Papoutsis-Kiachagias, N. Magoulas, J. Mueller, C. Othmer, and K. C. Giannakoglou. Noise reduction in car aerodynamics using a surrogate objective function and the continuous adjoint method with wall functions. *Computers & Fluids*, 122:223–232, 2015.

[21] J. E. V. Peter and R. P. Dwight. Numerical sensitivity analysis for aerodynamic optimization: a survey of approaches. *Computers & Fluids*, 39(3):373–391, 2010.

[22] L. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, New York, 2nd edition, 1997.

[23] G. M. Shroff and H. Keller. Stabilization of unstable procedures: the Recursive Projection Method. *SIAM Journal of Numerical Analysis*, 30(4):1099–1120, 1993.

[24] T. Skamagkis, E. M. Papoutsis-Kiachagias, and K. C. Giannakoglou. CFD-based shape optimization under uncertainties using the adjoint-assisted polynomial chaos expansion and projected derivatives. *Computers & Fluids*, 241, 105458, 2022.

[25] H. Sobieczky. Parametric airfoils and wings. In K. Fujii and G. S. Dulikravich, editors, *Recent Development of Aerodynamic Design Methodologies. Notes on Numerical Fluid Mechanics (NNFM)*. Volume 65, pages 71–88. Vieweg+Teubner Verlag, 1998.

[26] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *Proceedings of the 30th Aerospace Sciences Meeting and Exhibit*, AIAA Paper 92-0439, Reno, Nevada, USA, 1992.

[27] P. G. Tucker. Differential equation-based wall distance computation for DES and RANS. *Journal of Computational Physics*, 190:229–248, 2003.

[28] 'UIUC Airfoil Data Site'. University of Illinois, Department of Aerospace Engineering. https://m-selig.ae.illinois.edu/ads/coord_database.html. (accessed: 21.09.2021).

[29] R. Walters and L. Huyse. Uncertainty Analysis for Fluid Mechanics with Applications. Technical report ICASE 2002-1, NASA/CR-2002-211449, NASA Langley Research Center, Hampton, VA, 2002.

[30] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 187(1):137–167, 1938.

[31] D. Xiu and G. Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187(1):137–167, 2003.

[32] A. S. Zymaris. *Adjoint methods for the design of shapes with optimal aerodynamic performance in laminar and turbulent flows*. PhD thesis, National Technical University of Athens, 2010.