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

Adjoint-Based Aerodynamic Shape Optimization of Turbine Blade Airfoils using Customized Parameterization

Diploma Thesis

Theodoros Panagiotis Kokkos

Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2025

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Kyriakos C. Giannakoglou for his constant guidance and support throughout the entire course of my diploma thesis. His experience and approach to problem-solving as well as his willingness to help me at any time have truly been invaluable. I would also like to thank him for being the most inspiring figure during my studies and for making me, through his enlightening lectures, view engineering from a different perspective.

Special thanks are due to Dr. Xenofon Trompoukis for his inspiring guidance and support with every technical challenge I encountered throughout the course of this thesis. His profound knowledge, conveyed in a friendly and comprehensible manner, was invaluable, and I am deeply grateful for his mentorship. I would also like to thank all the members of the PCOpt Unit whom I had the pleasure of meeting and working with, for their readiness to assist me at any time.

Finally, and most importantly, I wish to express my heartfelt gratitude to my family for their unwavering support and the sacrifices they made throughout my studies, always supporting me. I am also thankful for the friends I have made along the way and the unforgettable memories we have created together.

National Technical University of Athens

School of Mechanical Engineering

Fluids Section

Lab. of Thermal Turbomachines

Parallel CFD & Optimization Unit

Adjoint-Based Aerodynamic Shape Optimization of Turbine Blade Airfoils using Customized Parameterization

Diploma Thesis

Theodoros Panagiotis Kokkos

Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2025

Abstract

Aerodynamic shape optimization (ShpO) plays a pivotal role in aerospace engineering design, enabling enhancements in performance metrics and the attainment of desired aerodynamic distributions. The objective of this diploma thesis is the development of a computational framework for the geometric parameterization and aerodynamic optimization of blade airfoil profiles. The blade geometry is parameterized using Bézier curves, constructed from design variables that represent physically meaningful blade characteristics such as chord length, stagger angle, metal inlet and outlet angles, leading and trailing edge radius and others. The blade is divided into four segments, allowing for localized geometric control. The suction and pressure sides are constructed based on a mean camber line, while leading and trailing edge segments are added at the front and rear, respectively. To ensure smooth and seamless transitions between the segments, C^2 continuity is enforced at the connections. Initially, an optimal fitting algorithm was developed in C++ using the steepest descent method to adjust the Bézier-based parameterization such that the generated geometry reproduces a given airfoil shape—performing inverse design. The fitting procedure is based on a least

squares error metric between the target and generated geometries. Following this stage, the Bezier parameterization was interfaced with the adjoint solver of the PUMA software, developed by the PCOpt/NTUA research unit, with the aim of solving shape optimization problems. Multiple studies have been conducted on the geometry of the two-dimensional rotor blade of a single-stage axial turbine, targeting aerodynamic performance objectives under a set of constraints. All aerodynamic optimizations were carried out using the continuous adjoint method, while in the case of the propoed parameterization, the chain rule was applied for the computation of sensitivity derivatives. Specifically, a C++ code was developed to compute the geometrical derivatives of each airfoil node with respect to the design variables. Then, using the chain rule, these derivatives were passed to the PUMA software to evaluate the sensitivities of the objective function. The aerodynamic ShpO was applied to minimize the total pressure losses of the airfoil representing the rotor of the single-stage turbine, subject to the constraints of prescribed flow turning angle, inlet capacity, and additional geometric limitations.

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

Αεροδυναμική Βελτιστοποίηση Μορφής Αεροτομών Πτερυγίων Στροβίλων με χρήση της Συζυγούς Μεθόδου και Προσαρμοσμένης Παραμετροποίησης

Διπλωματική Εργασία

Θεόδωρος Παναγιώτης Κόκκος

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, 2025

Περίληψη

Η αεροδυναμική βελτιστοποίηση μορφής (ΒΜ) διαδραματίζει καθοριστικό ρόλο στον σχεδιασμό αεροδιαστημικών εφαρμογών, καθώς επιτρέπει τη βελτίωση της συνολικής απόδοσης και την επίτευξη επιθυμητών αεροδυναμικών χαρακτηριστικών. Στόχος της διπλωματικής εργασίας είναι η ανάπτυξη ενός λογισμικού για τη γεωμετρική παραμετροποίηση και την αεροδυναμική ΒΜ αεροτομών (κυρίως πτερυγώσεων). Η γεωμετρία του πτερυγίου παραμετροποιείται μέσω καμπυλών Bézier, οι οποίες κατασκευάζονται από μεταβλητές σχεδιασμού που αντιστοιχούν σε φυσικά γεωμετρικά χαρακτηριστικά του πτερυγίου, όπως, μεταξυ άλλων, το μήκος χορδής, η γωνία κλίσης (stagger angle), οι γωνίες εισόδου και εξόδου μετάλλου, οι ακτίνες καμπυλότητας της ακμής πρόσπτωσης και εκφυγής. Το πτερύγιο χωρίζεται σε τέσσερα τμήματα, γεγονός που επιτρέπει τοπικό γεωμετρικό έλεγχο. Οι πλευρές υποπίεσης και υπερπίεσης κατασκευάζονται με βάση τη μέση γραμμή καμπυλότητας, ενώ προστίθενται τμήματα μορφής κυκλικού τόξου στην περιοχή γύρω από την ακμή πρόσπτωσης και εκφυγής. Για τη διασφάλιση ομαλών και συνεχών μεταβάσεων μεταξύ των τμημάτων,

επιβάλλεται συνέχεια τύπου C^2 στα σημεία σύνδεσης.

Αρχικά, αναπτύχθηκε σε γλώσσα προγραμματισμού C++ ένας αλγόριθμος βέλτιστης προσαρμογής με χρήση της μεθόδου της απότομης καθόδου, ώστε η παραγόμενη από την παραμετροποίηση, γεωμετρία να αποτυπώνει τη δοσμένη αεροτομή—υλοποιώντας κατ' αυτόν τον τρόπο την αντίστροφη σχεδίαση (inverse design). Η διαδικασία προσαρμογής βασίζεται στη μέθοδο ελαχίστων τετραγώνων, λαμβάνοντας υπόψη τη διαφορά μεταξύ της δοσμένης και της παραγόμενης γεωμετρίας. Στη συνέχεια, η παραμετροποίηση διασυνδέθηκε με τον συζυγή επιλυτή του λογισμικού PUMA, της ΜΠΤΡΒ/ΕΜΠ, με σκοπό την επίλυση προβλημάτων αεροδυναμικής ΒΜ.

Πραγματοποιήθηκαν πολλές μελέτες στη γεωμετρία του διδιάστατου πτερυγίου κινητής πτερύγωσης μονοβάθμιου αξονικού στροβίλου, με στόχο την επίτευξη καλύτερων αεροδυναμικών επιδόσεων υπό συγκεκριμένους περιορισμούς. Όλες οι αεροδυναμικές ΒΜ εκτελέστηκαν με χρήση της συνεχούς συζυγούς μεθόδου, ενώ στην περίπτωση της προτεινόμενης παραμετροποίησης εφαρμόστηκε ο κανόνας της αλυσίδας για τον υπολογισμό των παραγώγων ευαισθησίας. Πιο συγκεκριμένα, αναπτύχθηκε κώδικας σε γλώσσα προγραμματισμού C++ για τον υπολογισμό των γεωμετρικών παραγώγων κάθε κόμβου του προφίλ ως προς τις μεταβλητές σχεδιασμού. Έπειτα, με χρήση του κανόνα της αλυσίδας, οι παράγωγοι αυτοί μεταβιβάστηκαν στο λογισμικό PUMA για τον υπολογισμό των παραγώγων της αντικειμενικής συνάρτησης. Η αεροδυναμική ΒΜ εφαρμόστηκε για την ελαχιστοποίηση των ολικών απωλειών πίεσης της αεροτομής του πτερυγίου της κινητής πτερύγωσης του μονοβάθμιου στροβίλου, υπό τον περιορισμό προκαθορισμένης γωνίας εξόδου της ροής, παροχής στην είσοδο και πρόσθετων γεωμετρικών περιορισμών.

Abbreviations

CFD	Computational Fluid Dynamics
GPU	Graphics Processing Unit
NTUA	National Technical University of Athens
PCOpt	Parallel CFD and Optimization Unit
PUMA	Parallel Unstructured Multirow and Adjoint
MCL	Mean Camber Line
LE	Leading Edge
TE	Trailing Edge
SS	Suction Side
PS	Pressure Side
ShpO	Shape Optimization
w.r.t.	with respect to

ΕΜΠ	Εθνικό Μετσόβιο Πολυτεχνείο
ΜΠΥΡΒ	Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης
ΥΡΔ	Υπολογιστική Ρευστοδυναμική

Contents

Contents **viii**

1 Introduction **1**
1.1 Blade Airfoil Parameterization Methods 2
1.2 Aerodynamic Shape Optimization 5
1.3 Optimization Methods 6
1.4 The GPU-enabled CFD Solver PUMA 8
1.5 Thesis Outline 10

2 The Proposed Parameterization **12**
2.1 Mathematical Formulation 12
2.2 Camber Line Generation 14
2.3 Suction and Pressure Side Generation 16
2.4 Leading and Trailing Edge Modelling 20

3 Geometric Inverse Design Algorithm **25**
3.1 Formulation of the Best-Fit Loop 25
3.2 Initialization Techniques 30
3.3 Best-Fit Results 32

4 CFD Analysis - Validation **38**
4.1 Introduction 38
4.2 Mesh Generation 40
4.3 Results Validation 43

5 Integration of the Custom Parameterization into the ShpO Cycle **49**
5.1 ShpO Cycle 49

6 Aerodynamic ShpO Cases **53**
6.1 Case Description 53
6.2 Baseline Mesh 55

6.3 ShpO Results 57

7 Conclusions 69

7.1 Overview 69

7.2 Conclusions 70

7.3 Proposals for Future Work 70

Appendices 71

A Analytical derivative formulas 72

A.1 Derivatives of a Bézier curve 72

A.2 MCL partial derivatives w.r.t design variables 73

A.3 SS partial derivatives w.r.t design variables 77

A.4 PS partial derivatives w.r.t design variables 82

A.5 LE partial derivatives w.r.t design variables 88

A.6 TE partial derivatives w.r.t design variables 90

Bibliography 92

Chapter 1

Introduction

The design of turbine blades plays a pivotal role in determining the overall efficiency, reliability, and performance of turbomachinery. In gas turbines and steam turbines, aerodynamic optimization directly influences fuel consumption, power output, and operational lifespan. Traditionally, blade shapes have been derived from empirical correlations, iterative testing, and designer experience. However, the increasing demand for higher efficiency combined with the exponential growth of computational power of computers has led to the evolution of methods for blade design.

Among the various computational strategies, inverse design offers an effective approach in which the desired aerodynamic performance — such as pressure distribution or flow turning — is specified first, and the blade geometry is then determined to achieve it. This contrasts with the conventional trial-and-error forward design process and can significantly accelerate the development cycle [25], [35]. Complementary to this, adjoint-based aerodynamic shape optimization provides an efficient way to compute sensitivities of performance objectives with respect to numerous geometric parameters, enabling gradient-based optimization with a computational cost largely independent of the number of design variables [8], [13].

In this context, Bézier curve parameterization has emerged as a versatile tool for representing airfoil shapes ensuring smooth curvature and geometric flexibility. Its mathematical properties allow shape modifications while maintaining global smoothness, which is essential for capturing subtle aerodynamic effects in turbine cascades [11], [26].

The integration of inverse design and adjoint-based optimization, coupled with Bézier parameterization, offers a powerful framework for systematically improving 2D turbine blade performance. This diploma thesis applies such a framework to develop blade airfoil geometries that meet prescribed aerodynamic constraints while enhancing efficiency, thereby addressing modern challenges in turbomachinery design.

1.1 Blade Airfoil Parameterization Methods

The choice of parameterization technique significantly influences the optimization outcome, as it is involved both in the problem formulation and in the execution of the algorithm. Thus, it determines the discrepancy between the optimal and the computed shape, as well as the convergence rate. A geometry parameterization technique is suitable for integration within an aerodynamic optimization framework when it possesses some of the following characteristics [24]:

- The parameterization variables can be directly correlated with industrial design variables, e.g., thickness, twist of a wing or blade, or the mean camber line curvature of an airfoil, over which the designer maintains control.
- Updating parameters during each optimization cycle produces smooth geometric shapes. This is a critical factor when selecting parameterization techniques for gradient-based optimization methods, which require continuity of derivatives of surface nodes (design velocities) with respect to design variables.
- The analytical computation of these derivatives is feasible.
- The designer is afforded local control of the geometry, ensuring better handling and manipulation.
- The geometric deformation can be reconciled with the deformation imposed on the flow solution mesh.
- The time required for the accurate representation and parameterization of a given geometry is acceptable.
- The parameterization can employ variables introduced for geometry creation within CAD software.

Regarding airfoil parameterization, several methods have been developed. Below is a concise overview of the most common parameterization techniques:

1. Parameterization using CAD software

   Parameterization is based on geometric quantities imported from CAD software. In this approach, the sensitivity of the geometry to changes in design variables (design velocity) is computed using finite differences between the initial and deformed geometry nodes. This method has gained adoption in recent years because design velocity calculation via finite differences is independent of the parameterized geometry, with the direct advantage of no requirement for code reprogramming [33]. The implementation difficulty lies in the correspondence mapping between initial and final nodes, achieved under the assumption that geometry changes between iterations are small enough to allow projection of the new surface mesh onto the old one. This assumption imposes a small correction step size and consequently computational cost.

2. CST Parameterization

   The Class–Shape Transformation (CST) parameterization was proposed in and constitutes a method for representing 2D and 3D geometric shapes [16]. It expresses the geometry as the product of a class function, which defines the general geometric characteristics such as leading- and trailing-edge behavior, and a shape function, which provides flexibility to capture the detailed curvature of the side or surface. This method has also been applied within the context of the PCOpt/NTUA unit, as demonstrated in diploma thesis [20], where CST was employed for the parameterization and aerodynamic shape optimization of airfoil geometries.

3. PARSEC Parameterization

   The PARSEC parameterization specifically targets airfoil cases and uses a set of geometric parameters to describe fundamental airfoil characteristics, offering an approach with relatively few design variables. Various versions of PARSEC exist, with the most common one consisting of 11 design variables. The mathematical formulation of this method is detailed in [7], [18]. Furthermore, this parameterization has been utilized within the PCOpt/NTUA unit, as illustrated in diploma thesis [15], where it was applied for the parameterization and aerodynamic optimization of two-element airfoils.

4. Volumetric NURBS

   Volumetric B-Splines constitute an indirect technique for parameterizing curves and surfaces, with the additional capability to simultaneously morph both the geometry under study and the computational mesh nodes, which are also parameterized [32]. Control points are defined initially, which are subsequently displaced based on sensitivity derivatives to accordingly deform the geometry controlled by them. This technique is known as Free Form Deformation or Morphing Box Approach.

5. NURBS - Bézier Parameterization

   The NURBS parameterization is a widely used technique that offers a flexible way to define complex shapes [24]. Based on B-Spline curve theory, enables the user to manage the body directly. Moreover, NURBS provides smooth transitions between different shape regions and high accuracy in representation, which is why it has been widely adopted in many CAD packages.

   A special category within the NURBS family is the Bézier curve which are constructed using Bernstein polynomials. They consist of a set of control points that define the shape of the curve, with the curve smoothly interpolating between these points [12]. Unlike general NURBS, Bézier curves use uniform weights and a simpler knot structure, which makes them easier to implement and understand. They are well-suited for aerodynamic parameterization when the geometry is relatively simple or when a smooth, continuous curve needs to be constructed with relatively few parameters. One of the main advantages of Bézier curves in parameterization is their intuitive control: moving a control point reshapes the entire curve in a predictable way. This global influence simplifies shape design and optimization tasks by reducing complexity while still producing smooth curves. However, this same global effect can also be a drawback in some cases, as adjusting a single control point affects the entire curve, potentially limiting local control and making it difficult to accurately represent complex geometries. Whether this characteristic is advantageous or restrictive depends on how the parameterization is managed and applied. Additionally, Bézier curves are computationally efficient to evaluate and direct differentiatable, making them ideal for gradient-based optimization methods commonly used in aerodynamic shape design.

One of the objectives of this thesis is to present a custom Bézier-based parameterization method that achieves a balance between global and local control by generating the blade airfoil in distinct segments: camber line, leading edge region, trailing edge region, suction side, and pressure side. To this end, an inverse design code for airfoil shapes was developed in C++. The proposed parameterization is general and applicable to a wide range of airfoils, while being particularly well-suited for 2D turbine blades, where the design variables are directly correlated with industrial design parameters. Further details regarding the mathematical expression of the method, the role of the design variables and their contribution to the final airfoil geometry will be presented subsequently.

1.2 Aerodynamic Shape Optimization

The term aerodynamic shape optimization refers to the appropriate modification of the geometry of a solid body exposed to a fluid flow, such that a quantity called the Objective Function, F, be minimized or maximized. An example of an objective function is the force exerted by the fluid on the solid body in a specific direction, such as lift or drag. The geometry of the body under optimization is described by a set of variables $\vec{b}$, which are called design variables. The values of the design variables determine the geometry of the solid body. Therefore, the goal of the optimization is to find the values of the design variables $\vec{b}$ that minimize function F.

Aerodynamic shape optimization is typically performed by integrating computational fluid dynamics (CFD) with advanced optimization algorithms to identify the best geometric configuration for a given design problem. This process involves handling numerous design variables alongside predefined objectives and constraints, aiming to achieve improvements in aerodynamic performance metrics compared to an initial baseline. The scope of this research spans various industries, including aerospace, automotive, and wind energy.

Recent advancements in high-performance computing have significantly propelled aerodynamic optimization efforts, as CFD analyses are computationally intensive and can require extensive runtime, even when parallel processing techniques on cutting-edge hardware are employed. Additional difficulties arise from mesh dependency and the inherently nonlinear behavior of fluid flows. During iterative optimization, geometry modifications necessitate mesh regeneration or deformation, which can lead to inconsistent gradient calculations degrading solution accuracy. Moreover, the complex nonlinear dynamics of fluid phenomena—such as boundary layers and shock interactions—pose substantial challenges, as minor geometric alterations may cause sudden, large fluctuations in aerodynamic characteristics. This results in a design space with multiple local optima, complicating the search for a globally optimal solution.

1.3 Optimization Methods

Optimization problems can be classified according to various criteria, including the optimization methodology employed, the number of objective functions, and the presence or absence of constraints.

With respect to the number of objective functions, optimization problems are divided into Single Objective Optimization (SOO) problems, where a single performance metric is optimized, and Multi-Objective Optimization (MOO) problems, where two or more competing objectives must be simultaneously optimized [7].

Based on the optimization methodology, two primary categories are distinguished: stochastic methods and deterministic methods.

- **Stochastic or Gradient-Free Methods**

Stochastic, population-based optimization methods [7] constitute versatile algorithms applicable to a wide range of engineering problems. A representative class of stochastic methods is the Evolutionary Algorithms (EA). Their defining feature is the manipulation and evaluation of a population (population-based) of candidate solutions within each generation (iteration) mimicing the mechanisms of natural evolution [4]. These solutions are randomly generated and assessed according to the value of the objective function. A subset of solutions, termed parents, is promoted through a selection mechanism inspired by the principle of "survival of the fittest" (elitist tendency), wherein the best-performing individuals are preserved. Offspring solutions are subsequently produced through crossover operators, emulating heredity, and mutation operators, enabling the introduction of new traits. These offspring are evaluated in turn, and the process is iterated until convergence criteria are met the initialization, provided a sufficiently large number of evaluations is performed. As a result, they are inherently less prone to entrapment in local optima. However, their convergence generally requires a substantial number of objective function evaluations, which can lead to high computational cost. Moreover, the computational expense increases with the dimensionality of the design space, i.e. the number of design variables.

- **Deterministic or Gradient-Based Methods**

Conversely, deterministic optimization methods [19] rely on the computation or approximation of first (sensitivities or gradients) and ,sometimes, second-order derivatives (Hessian matrix) of the objective function, evaluated at the current design point. These derivatives are used to determine the search direction for subsequent iterations. Through an iterative procedure, deterministic methods progressively guide the solution by reducing the magnitude of these sensitivity derivatives at each cycle, with the ultimate goal of driving them to zero, thereby identifying an optimal solution.

Deterministic methods are more prone to becoming trapped in local extrema, and their convergence behavior is strongly dependent on the quality of the initial guess. Typically, they are employed primarily in single-objective optimization problems, where the objective function is scalar-valued. In multi-objective cases, the various objectives are commonly aggregated into a single scalar objective function through the application of user-defined weighting coefficients. Furthermore, deterministic approaches generally exhibit limited adaptability to arbitrary problem formulations in comparison to the stochastic ones, often requiring tailored implementation and increased programming effort to accommodate specific problem structures.

Despite these limitations, deterministic methods generally converge in significantly fewer function evaluations, which translates to considerably reduced computational cost. Additionally, their computational expense is often largely independent of the dimensionality of the design space, enabling efficient optimization of high-dimensional problems even when computational resources are limited. This characteristic makes deterministic methods particularly suitable for optimizing complex geometries described parametrically using analytic parametric representations.

Among the most commonly employed deterministic methods are the Steepest Descent method, which involves exact or approximate calculation of the first derivative, the Quasi-Newton method, which calculates the first derivative exactly while approximating the second derivative to reduce computational cost yet maintain rapid convergence and the Newton method, which relies on precise evaluations of both first and second derivatives.

Common derivative computation techniques include the Direct Differentiation Method (DDM), the Finite Difference Method, the Complex Step Method and both the Continuous and Discrete Adjoint Methods [7]. Except for the adjoint methods, the computational cost of these techniques typically scales linearly with the number of design variables. The adjoint method, however, is notably efficient: the computational cost of calculating the first derivative of the objective function is roughly equivalent to solving the primal problem twice. The selection of an appropriate optimization method and derivative computation technique depends on the specific characteristics of the problem at hand, and the engineer must carefully evaluate these factors to make informed decisions regarding the optimization strategy.

Lastly, optimization problems are broadly divided into constrained and unconstrained categories. In unconstrained optimization, no explicit restrictions are placed on the decision variables, which makes the problem simpler. These problems often arise directly in certain applications or as reformulations of constrained problems using penalty methods.. In contrast, constrained optimization involves explicit restrictions on variables, which can range from simple bounds to complex nonlinear inequalities that reflect real-world limitations or relationships. Properly handling these constraints is crucial since they define the feasible region where the optimal solution must lie.

1.4 The GPU-enabled CFD Solver PUMA

In order to predict the compressible flow field around the airfoils, the GPU-enabled flow solver PUMA, developed by the PCOpt/NTUA [29], will be employed in this thesis. PUMA numerically solves the Navier–Stokes equations, coupled with turbulence model equations, over a computational domain employing a vertex-centered finite volume formulation on unstructured meshes consisting of tetrahedra, pyramids, prisms, and hexahedra. Structured or matching block-structured grids, such as those generated in the present work, are treated internally as unstructured. The governing equations are predominantly hyperbolic in character, with elliptic terms playing a secondary role. Consequently, PUMA employs a time-marching technique, introducing a pseudo-time variable to iterate toward a steady-state solution. The spatial discretization is performed on finite volumes defined around each mesh vertex by connecting edge midpoints, face centers, and element centroids of the adjacent cells. Inviscid fluxes are computed using Roe's approximate Riemann solver [23], while viscous terms are discretized using a second-order central difference scheme involving all neighboring nodes.

The flow equations are numerically solved in a decoupled manner within each pseudo-time iteration at first, an iteration of the mean flow equations is performed, followed by one of the Spalart-Allmaras model. The adjoint equations are solved using a similar procedure [31]. PUMA incorporates steepest descent, BFGS, and damped BFGS optimization algorithms [14]. If optimization is disabled, the adjoint solver is launched only once to compute the sensitivity derivatives of the selected objective function for the baseline configuration, with no update to the design variables. The adjoint based sensitivity derivatives can be computed based on SI, FI or E-SI formulations [1]. SI formulation includes only surface integrals which makes derivative computation fast but less accurate compared to FI which includes domain-wide field integrals (both surface and volume integrals) that account for grid sensitivity terms such as, which describe how the grid adjusts when the design parameters change. These integrals demand a grid displacement model (GDM) solved via finite differences or analytic differentiation to properly capture those effects. As a result, FI is accurate, but scales poorly with mesh size because of the computational cost. E-SI combines the accuracy of FI with the efficiency of SI. By incorporating a Laplacian type GDM into the primal equations, and introducing additional adjoint fields, the method cancels out grid sensitivities within the domain. This reformulation results in sensitivity derivatives expressed strictly through surface integrals a reduced adjoint. It matches the accuracy of FI but at a significantly reduced computational cost. Mesh adaptation in PUMA can be performed as a standalone operation or during optimization or unsteady simulations with moving boundaries. Available morphing techniques include [31]

1. External (variant of the Inverse Distance Weighting) - each mesh node is linked with the closest wall node and a smoothing function (whose value is 1.0 on the wall and 0.0 far from it) is applied for computing the deformation field around the deformed surface
2. Volumetric NURBS
3. Laplace PDE morphing
4. Linear Spring Analogy
5. Torsional Spring Analogy
6. Elastic Medium (solid-mechanics-based deformation)
7. Inverse Distance Weighting (IDW)

8. Adjoint Laplace — Used in the E-SI formulation; solves the adjoint to Laplace's equation for the deformation after convergence of primal and adjoint fields, yielding purely surface-integral sensitivities.

The spring analogy treats mesh edges as linear springs with stiffness inversely proportional to their length. The elastic medium method models the mesh as a continuum elastic body, with material properties depending on geometric quantities such as cell volumes. PDE-based methods, including Laplace and elastic-medium morphers, are solved iteratively using Jacobi, Gauss–Seidel, GMRES, or QMRCGStab linear solvers.

The implementation of PUMA on GPUs provides a remarkable speed-up in comparison with CPU implemented software, reducing the time of a CFD analysis. The GPUs that were used are NVIDIA Tesla K40m for the CFD analysis and validation of the turbine stage and the NVIDIA Tesla V100 for the aerodynamic ShpO cases.

1.5 Thesis Outline

The outline of this diploma thesis is structured as follows

- **Chapter 2:** Describes the mathematical framework of the custom Bézier-based parameterization, including its structural components, an explanation of the design variables, and the formulation adopted, as well as their influence on the final airfoil geometry.
- **Chapter 3:** Analyzes the airfoil shape-fitting algorithm developed for the inverse design of airfoils in this work and presents its application to two-dimensional turbine blade cases.
- **Chapter 4:** Refers to the CFD simulations performed using PUMA and their validation against experimental data.
- **Chapter 5:** Details the modifications implemented in the PUMA software to communicate the Bézier parameterization into the optimization loop.
- **Chapter 6:** It illustrates the results of the aerodynamic ShpO of the airfoil geometry representing the rotor blade of a single-stage turbine for multiple cases, each subject to different geometric constraints but sharing the same objective and flow conditions.

- **Chapter 7:** An overview of the present diploma thesis, followed by conclusions drawn from the studies and suggestions for future work

Chapter 2

The Proposed Parameterization

In this work, the airfoil parameterization is based on a conceptual approach, wherein the aifoil geometry was represented through five distinct segments: the suction and pressure sides, defined relative to the mean camber line, and the leading and trailing edge regions, modeled as circular arcs. All segments were constructed using Bézier curves. This chapter presents the mathematical formulation of these curves and details the methodology employed for the construction of each airfoil segment, providing the foundation for the implementation of the corresponding C++ code.

2.1 Mathematical Formulation

Bézier curves were originally introduced by Paul de Casteljau in 1959 while working at Citroën, and later popularized by Pierre Bézier at Renault during the 1960s [22]. They provide a convenient and efficient means of representing smooth, arbitrary curves using a relatively small number of control points. This property makes them particularly valuable in computer-aided design and vector graphics, where they enable compact storage and scalable representation of curved paths [2].

A Bézier curve of degree n is defined as a parametric curve expressed as a linear combination of $n+1$ control points, denoted as $P_0, P_1, \ldots, P_n$. The mathematical formulation is based on the Bernstein polynomials, which serve as blending functions that guarantee smooth interpolation between the control points. The first and last points, P_0 and P_n, are referred to as *anchor points*, as they coincide with the endpoints of the curve. The intermediate points do not generally lie on the curve but instead define its overall shape. Together, the control points form a convex hull that encloses the curve.

The parametric form of a Bézier curve is expressed as a weighted sum of the control points:

$$\mathbf{B}(t) = \sum_{i=0}^{n} B_{i,n}(t)\, P_i, \quad t \in [0,1] \tag{2.1}$$

where

$$B_{i,n}(t) = \binom{n}{i} (1-t)^{n-i} t^i \tag{2.2}$$

are the Bernstein basis polynomials of degree n.

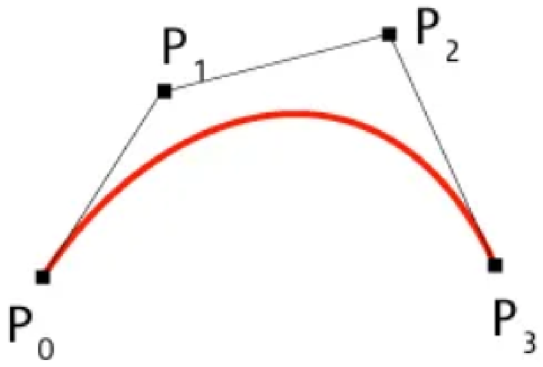


Figure 2.1: *Generation of the Bézier curve and the control polygon formed by the control points.*

Bézier curves possess several fundamental properties. Some of them are:

- **Endpoint interpolation:** the curve always passes through the first and last control points, i.e., $\mathbf{B}(0) = P_0$ and $\mathbf{B}(1) = P_n$.
- **Endpoint tangents:** the tangent direction at the beginning of the curve is determined by the line segment (P_0, P_1), while the tangent direction at the end of the curve is determined by the line segment

(P_{n-1}, P_n) [6]. Mathematically,

$$\mathbf{B}'(0) = n(P_1 - P_0), \quad \mathbf{B}'(1) = n(P_n - P_{n-1}).$$

- **Convex hull property:** the curve lies entirely within the convex hull of its control points, ensuring geometric stability.
- **Variation-diminishing property:** the curve oscillates no more than the polygon formed by its control points.
- **Affine invariance:** the curve preserves its shape under affine transformations such as translation, rotation, and scaling.
- **Continuity and smoothness:** the smoothness of the curve can be adjusted by modifying the positions of the control points.

Owing to these properties, Bézier curves offer an intuitive yet mathematically precise way to define smooth geometries. In aerodynamic design, they enable accurate control over the curvature of airfoils while maintaining a relatively low number of design parameters, making them well-suited for gradient-based aerodynamic optimization.

2.2 Camber Line Generation

The mean camber line $MCL(u)$ is generated using a cubic Bézier curve. Its four control points are determined from six design variables and two constants. The design variables correspond to typical blade airfoil design parameters, while the constants are the coordinates of the leading edge tip $(x_{\text{in}}, y_{\text{in}})$. The control points are defined as:

$$\mathbf{P}_0^c = \begin{bmatrix} x_{\text{in}} \\ y_{\text{in}} \end{bmatrix}, \tag{2.3}$$

$$\mathbf{P}_1^c = \mathbf{P}_0^c + d_{\text{in}} \begin{bmatrix} c\cos(\alpha_1') \\ c\sin(\alpha_1') \end{bmatrix}, \tag{2.4}$$

$$\mathbf{P}_2^c = \mathbf{P}_3^c - d_{\text{out}} \begin{bmatrix} c\cos(\alpha_2') \\ c\sin(\alpha_2') \end{bmatrix}, \tag{2.5}$$

$$\mathbf{P}_3^c = \mathbf{P}_0^c + \begin{bmatrix} c\cos(\zeta) \\ c\sin(\zeta) \end{bmatrix} \tag{2.6}$$

where ζ is the stagger angle, $c_{\text{ax}} = c\cos(\zeta)$ is the axial chord length, α_1' and α_2' are the inlet and outlet metal angles, and d_{in} and d_{out} are the inlet and outlet tangent proportions.

Table 2.1 summarizes the user-defined the design variables employed for the generation of the mean camber line, along with their corresponding symbols and definitions.

Variable name	**Symbol**
Chord length	c
Stagger angle	ζ
Inlet metal angle	α_1'
Exit metal angle	α_2'
Inlet tangent proportion	d_{in}
Outlet tangent proportion	d_{out}

Table 2.1: *Camber line design variables.*

This construction, as shown in Figure 1a, ensures that the blade airfoil achieves the specified axial chord length and that the slopes at the leading and trailing edges exactly match the prescribed metal angles, thanks to the endpoint tangency property of Bézier curves. Using physical design variables such as metal angles, stagger angle, and chord proportions provides the engineer with direct control over geometrically and aerodynamically meaningful parameters, enhancing both the intuitive understanding of the design and the ability to make informed trade-offs during optimization.

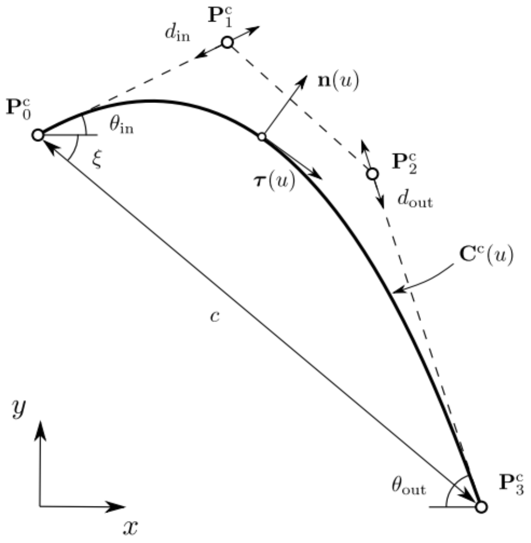

d_{in}
$\mathbf{P}_1^{\text{c}}$
$\mathbf{n}(u)$
$\mathbf{P}_0^{\text{c}}$
θ_{in}
ξ
$\boldsymbol{\tau}(u)$
$\mathbf{P}_2^{\text{c}}$
d_{out}
$\mathbf{C}^{\text{c}}(u)$
c
y
θ_{out}
$\mathbf{P}_3^{\text{c}}$
x

Figure 2.2: *Construction of the Mean Camber Line using a cubic Bézier curve with intuitive design variables.*

2.3 Suction and Pressure Side Generation

Given that the camber line has already been defined, the suction side (SS) and pressure side (PS) of the blade airfoil are defined as Bézier curves, with the number of control points specified by the user. The main idea is that the inner control points are generated by projecting a thickness distribution function (potentially different for the pressure and suction side) along the normal direction of the mean camber line. To ensure C^2 continuity at the junctions with the leading edge (LE) and trailing edge (TE), the first three and last three control points of each side are constrained accordingly. This feature is important for the aerodynamic design of turbomachinery blades because a sudden change in curvature could cause a spike in the surface pressure distribution or even a local separation bubble.

This thickness distribution is expressed as a fourth-degree polynomial in the parametric variable $t \in [0, 1]$. The use of t rather than the x coordinate is motivated by the fact that thickness is naturally defined along the camber line rather than in Cartesian space. Importantly, the suction and pressure sides are not required to share the same thickness distribution polynomial. This distinction is crucial in the case of turbine blade airfoils, where the pressure side is typically shorter and less curved than the suction side. The polynomial coefficients thus serve as design variables, enabling control over both the maximum thickness and its spatial distribution along the camber line.

The coordinates of the control points $\{P_i^{SS}\}$ and $\{P_i^{PS}\}$ are computed as follows:

$$\mathbf{P}_i^{SS} = \begin{cases} \mathbf{P}^{TE}_{n_b^{TE}-1}, & i = 0, \\ \dfrac{-\mathbf{P}^{TE}_{n_b^{TE}-2} + (N+1)\,\mathbf{P}^{TE}_{n_b^{TE}-1}}{N}, & i = 1, \\ \dfrac{\mathbf{P}^{TE}_{n_b^{TE}-1} - 2\mathbf{P}^{TE}_{n_b^{TE}-2} + \mathbf{P}^{TE}_{n_b^{TE}-3}}{N_2} + 2\mathbf{P}_1^{SS} - \mathbf{P}_0^{SS}, & i = 2, \\ \mathbf{MCL}(u_i) \pm \frac{1}{2}\, t(k_i)\, \vec{n}(u_i), & i = 3, \ldots, n_b^{SS} - 4, \\ \dfrac{\mathbf{P}_2^{LE} - 2\mathbf{P}_1^{LE} + \mathbf{P}_0^{LE}}{N_4} + 2\mathbf{P}^{SS}_{n_b^{SS}-2} - \mathbf{P}^{SS}_{n_b^{SS}-1}, & i = n_b^{SS} - 3, \\ \dfrac{-\mathbf{P}_1^{LE} + (N_3+1)\,\mathbf{P}_0^{LE}}{N_3}, & i = n_b^{SS} - 2, \\ \mathbf{P}_0^{LE}, & i = n_b^{SS} - 1 \end{cases} \tag{2.7}$$

$$\mathbf{P}_i^{PS} = \begin{cases} \mathbf{P}^{LE}_{n_b^{LE}-1}, & i = 0, \\ \dfrac{-\mathbf{P}^{LE}_{n_b^{LE}-2} + (N_5+1)\mathbf{P}^{LE}_{n_b^{LE}-1}}{N_5}, & i = 1, \\ \dfrac{\mathbf{P}^{LE}_{n_b^{LE}-1} - 2\mathbf{P}^{LE}_{n_b^{LE}-2} + \mathbf{P}^{LE}_{n_b^{LE}-3}}{N_6} + 2\mathbf{P}_1^{PS} - \mathbf{P}_0^{PS}, & i = 2, \\ \mathbf{MCL}(u_j) \pm \frac{1}{2}\, t_p(k_i)\, \vec{n}(u_j), & i = 3, \ldots, n_b^{PS} - 4, \\ \dfrac{\mathbf{P}_2^{TE} - 2\mathbf{P}_1^{TE} + \mathbf{P}_0^{TE}}{N_8} + 2\mathbf{P}^{PS}_{n_b^{PS}-2} - \mathbf{P}^{PS}_{n_b^{PS}-1}, & i = n_b^{PS} - 3, \\ \dfrac{-\mathbf{P}_1^{TE} + (N_7+1)\mathbf{P}_0^{TE}}{N_7}, & i = n_b^{PS} - 2, \\ \mathbf{P}_0^{TE}, & i = n_b^{PS} - 1 \end{cases} \tag{2.8}$$

where the ± symbol is included to indicate for whether the suction side lies above or below the mean camber line (the same convention applies for the pressure side). The k_i and u_i are the sampling indices expressed by the formulas:

$$k_i = \frac{i-2}{n_{bSS}-1}, \qquad u_i = \text{round}\left(k_i \cdot i_{\max}^{CL}\right) \tag{2.9}$$

$$k_j = \frac{j+1}{n_b^{PS}-1}, \qquad u_j = \text{round}\left(k_j \cdot i_{\max}^{CL}\right) \tag{2.10}$$

The thickness 4th-degree polynomial :

$$t(k_i) = a_0 + a_1 k_i + a_2 k_i^2 + a_3 k_i^3 + a_4 k_i^4 \tag{2.11}$$

The scaling factors for C^2 continuity:

$$N = \frac{n_{bTE}-1}{n_{bSS}-1}, \qquad N_2 = \frac{(n_{bTE}-1)(n_{bTE}-2)}{(n_{bSS}-1)(n_{bSS}-2)} \tag{2.12}$$

$$N_3 = \frac{n_{bLE}-1}{n_{bSS}-1}, \qquad N_4 = \frac{(n_{bLE}-1)(n_{bLE}-2)}{(n_{bSS}-1)(n_{bSS}-2)} \tag{2.13}$$

$$N_5 = \frac{n_b^{PS}-1}{n_b^{LE}-1}, \qquad N_6 = \frac{(n_b^{PS}-1)(n_b^{PS}-2)}{(n_b^{LE}-1)(n_b^{LE}-2)} \tag{2.14}$$

$$N_7 = \frac{n_b^{PS}-1}{n_b^{TE}-1}, \qquad N_8 = \frac{(n_b^{PS}-1)(n_b^{PS}-2)}{(n_b^{TE}-1)(n_b^{TE}-2)} \tag{2.15}$$

The unit vectors normal to the camber line $\mathbf{n}(u)$ are computed from the unit tangent vector $\mathbf{t}(u)$ according to

$$\mathbf{n}(u) = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} -t_y \\ t_x \end{bmatrix}, \quad \text{with} \quad \boldsymbol{t}(u) = \frac{\dot{\mathbf{MCL}}(u)}{\|\dot{\mathbf{MCL}}(u)\|} \tag{2.16}$$

where $\dot{\mathbf{MCL}}(u)$ is computed using analytical derivative formulas for Bézier curves. Further details are provided in Appendix A.

The required design variables for the generation of the pressure and suction side are presented in Table 2.2.

Variable name	**Symbol**
Suction-side thickness coefficients	$a_{0s}, a_{1s}, a_{2s}, a_{3s}, a_{4s}$
Pressure-side thickness coefficients	$a_{0p}, a_{1p}, a_{2p}, a_{3p}, a_{4p}$

Table 2.2: *Suction and pressure side design variables.*

Another user-defined factor that plays a crucial role in the thickness distribution along each side is the index sampling of the camber line. Since the generated camber line contains more actual points than the control points of the suction side (SS), an appropriate sampling technique must be selected. The simplest approach is uniform (equal) distribution. Equations (2.9) and (2.10) present an adjusted version of this method for the SS and PS respectivly (SS usually needs more support in the front while PS requirs the exact opposite). The adjustment arises from the fact that the first three and last three control points of each side are determined by the corresponding control points of the leading edge (LE) and trailing edge (TE), respectively, thereby enforcing C^2 continuity at these points. As a consequence, there is a concentration of control points near the LE and TE, leaving a relatively large gap among the inner control points. This not only limits the influence of the first and last control points—which primarily enforce continuity rather than actively shape the geometry but also places a greater burden on the inner control points to capture the most complex portions of the sides, such as the regions near maximum camber. To address this issue, the aforementioned distribution of sampling points is employed.

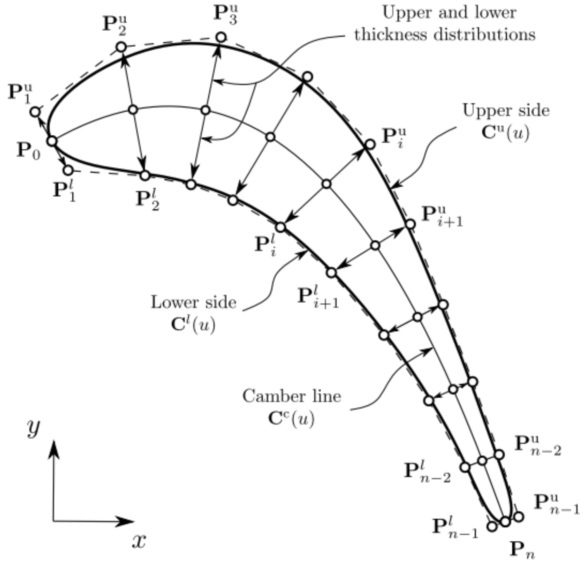

$\mathbf{P}_2^{\mathrm{u}}$
$\mathbf{P}_3^{\mathrm{u}}$
Upper and lower thickness distributions
$\mathbf{P}_1^{\mathrm{u}}$
Upper side $\mathbf{C}^{\mathrm{u}}(u)$
$\mathbf{P}_0$
$\mathbf{P}_i^{\mathrm{u}}$
$\mathbf{P}_1^l$
$\mathbf{P}_2^l$
$\mathbf{P}_{i+1}^{\mathrm{u}}$
$\mathbf{P}_i^l$
Lower side $\mathbf{C}^l(u)$
$\mathbf{P}_{i+1}^l$
Camber line $\mathbf{C}^{\mathrm{c}}(u)$
y
$\mathbf{P}_{n-2}^{\mathrm{u}}$
$\mathbf{P}_{n-2}^l$
$\mathbf{P}_{n-1}^{\mathrm{u}}$
x
$\mathbf{P}_{n-1}^l$
$\mathbf{P}_n$

Figure 2.3: *Construction of the Suction and Pressure Side of the airfoil using normal projection of the thickness distribution to the mean camber line.*

2.4 Leading and Trailing Edge Modelling

To complete the airfoil, the final step is the generation of the leading and trailing edge segments. It should be noted, in order to avoid confusion that, while the leading and trailing edges are typically considered as points, in this thesis the term will refer to a broader region that inlcudes these points.

Typically, these segments are constructed as curves—including only portions of the SS and PS—and are defined via a radius that implicitly controls the local curvature, ensuring G^2 continuity between the SS and PS. In this thesis, a different approach is adopted. Given that the leading and trailing edge shapes of airfoils (especially 2D turbine blades) are often elliptical and sometimes circular, the osculating circle method is employed.

The osculating circle of a curve C at a point P is the circle that shares both the tangent and the curvature of C at P. Just as the tangent line provides the best linear approximation of a curve at a point, the osculating circle provides the best circular approximation at that point [9].

To define this circular arc, a cubic Bézier curve is utilized. Its four control points are determined from five design variables, which correspond directly to physical parameters such as the arc radius, the center coordinates, and its angular span. The control points are then defined as follows:

$$\mathbf{P}_0^{LE} = \begin{bmatrix} x_{cLE} + r_{LE}\cos(\theta_0) \\ y_{cLE} + r_{LE}\sin(\theta_0) \end{bmatrix}, \tag{2.17}$$

$$\mathbf{P}_1^{LE} = \mathbf{P}_0^{LE} - \frac{4}{3} r_{LE} \tan\left(\frac{\theta_{LE}}{4}\right) \begin{bmatrix} \sin(\theta_0) \\ -\cos(\theta_0) \end{bmatrix}, \tag{2.18}$$

$$\mathbf{P}_3^{LE} = \begin{bmatrix} x_{cLE} + r_{LE}\cos(\theta_3) \\ y_{cLE} + r_{LE}\sin(\theta_3) \end{bmatrix}, \tag{2.19}$$

$$\mathbf{P}_2^{LE} = \mathbf{P}_3^{LE} + \frac{4}{3} r_{LE} \tan\left(\frac{\theta_{LE}}{4}\right) \begin{bmatrix} \sin(\theta_3) \\ -\cos(\theta_3) \end{bmatrix} \tag{2.20}$$

where $(x_{\text{cLE}}, y_{\text{cLE}})$ are the coordinates of the center of the leading edge circular arc, r_{LE} is the leading edge radius, θ_{0LE} and θ_{3LE} are the starting and ending angles of the leading edge arc, and $\theta_{\text{LE}} = \theta_{\text{3LE}} - \theta_{\text{0LE}}$ denotes the angular span of the arc. The same set of formulas is also associated with the trailing edge arc.

The design variables needed for the generation of the leading and trailing edge are listed in Table 2.3.

Variable name	**Symbol**
Circular arc center coordinates	$(x_{cLE}, y_{cLE}), (x_{cTE}, y_{cTE})$
Circular arc radius	r_{LE}, r_{TE}
Circular arc starting angles	$\theta_{0LE}, \theta_{0TE}$
Circular arc ending angles	$\theta_{3LE}, \theta_{3TE}$

Table 2.3: *Leading and trailing edge design variables.*

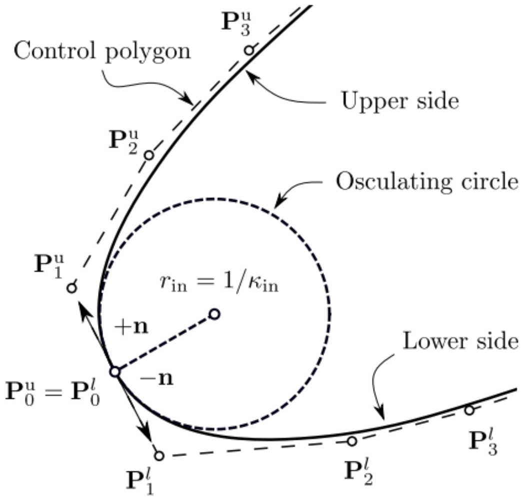

$\mathbf{P}_3^{\mathrm{u}}$
Control polygon
Upper side
$\mathbf{P}_2^{\mathrm{u}}$
Osculating circle
$\mathbf{P}_1^{\mathrm{u}}$
$r_{\mathrm{in}} = 1/\kappa_{\mathrm{in}}$
$+\mathbf{n}$
Lower side
$-\mathbf{n}$
$\mathbf{P}_0^{\mathrm{u}} = \mathbf{P}_0^{l}$
$\mathbf{P}_3^{l}$
$\mathbf{P}_2^{l}$
$\mathbf{P}_1^{l}$

Figure 2.4: *The osculating circle at the Leading Edge.*

The offset distance between the endpoint P_0 of a circular arc and the adjacent Bézier control point P_1 (and symmetrically between P_2 and P_3) is given by

$$d = \frac{4}{3} r \tan\left(\frac{\Delta\theta}{4}\right)$$

This expression arises from the requirement that a cubic Bézier curve approximating a circular arc must not only interpolate the endpoints of the arc, but also reproduce its local geometric properties. In particular, the Bézier curve is constrained to match both the tangent direction and the curvature of the circle at the endpoints.

The derivation of the above formula follows from a Taylor expansion of the circle and the cubic Bézier representation around $t = 0$ [5]. By equating the expansions up to second order, one ensures that the first and second derivatives coincide at the endpoints, thereby enforcing tangent and curvature consistency. This condition leads directly to the factor $\frac{4}{3}$ in the expression for d.

As a consequence, the resulting cubic Bézier curve osculates the circle at P_0 and P_3, meaning that it shares the same position, tangent, and curvature at these points. Such construction guarantees third-order accuracy in arc length. In contrast, omitting the $\frac{4}{3}$ factor would reduce the approximation to first-order accuracy, yielding only tangent agreement without curvature preservation.

As shown in Table 2.4, the relative error of this approximation is very small. For a quarter-circle ($\Delta\theta = \pi/2$), this approximation is highly accurate: the maximum radial error is approximately $0.00027\,r$ (i.e., roughly 0.027% of the radius). For larger arcs, it is common practice to subdivide the curve into multiple Bézier segments. For instance, a full circle is typically represented using four cubic Bézier curves, each spanning 90°. Consequently, this approach represents the best-practice method in computer-aided design (CAD) and computer graphics for approximating circular arcs with cubic Bézier curves.

Angle	Relative Error
$\pi/4$	4.2×10^{-6}
$\pi/2$	2.7×10^{-4}
π	1.8×10^{-2}
$3\pi/2$	2.8×10^{-1}

Table 2.4: *Relative error at different angles [3].*

The reason for adopting this approach, rather than the more straightforward one (i.e., simply joining the pressure and suction sides at the front), lies in the treatment of the leading edge (and similarly the trailing edge) as a curve rather than a sipmple point. Modeling it in this manner ensures infinite continuity at the internal points, thereby eliminating the risk of spikes in curvature—a critical consideration for aerodynamic design, as this region corresponds to the stagnation point. At the boundary points—specifically the junctions with the SS and PS—C^2 continuity can be enforced, ensuring a smooth transition. In contrast, the conventional approach not only places the junction point within the region of interest but also achieves only G^2 continuity between the SS and PS.

For clarity, it is important to distinguish between C^2 and G^2 continuity. Type C^2 continuity requires agreement of the second derivatives in addition to C^1 continuity. Because curvature depends on both the first and second derivatives, C^2 continuity implies that the curvature is continuous. On the other hand, G^2 continuity requires only that the directions of the second derivatives are identical, which is less stringent than C^2. By adjusting the parameterization of one entity, the geometric curvatures—independent of parameterization—can be made to agree.

Another motivation for this technique arises from the design considerations of turbine blades. Many blades incorporate complex cooling systems to withstand the high gas temperatures downstream of the combustion chamber. Designing these systems can present challenges both in terms of aerodynamic performance and manufacturability, particularly if the blade thickness in this region is very small. By employing the current approach, it is possible to optimize this critical section of the blade independently, while keeping the remainder of the geometry unchanged (as will be shown in Chapter 6) to avoid construction conflicts, or vice versa, something that is not achievable with the conventional method. This methodology provides substantial flexibility in controlling a small yet crucial region of the blade.

Overall, this custom Bézier-based parameterization offers a unique conceptual approach to airfoil generation, tailored for blade airfoils, by dividing the geometry into segments. A total of 26 design variables is used to describe the entire airfoil, providing a compact yet flexible representation. The Tables 2.1, 2.2, 2.3 highlight the intuitive nature of the design variables and demonstrate the extent to which the user can directly assign values to physical parameters. By segmenting the airfoil, this method also transforms the inherently global influence of Bézier curves into a more local control, as each segment is independent. This contrasts with other airfoil parameterization methods, in which the entire airfoil is described by one or two Bézier curves, making it difficult to capture complex geometries due to limited local control.

Chapter 3

Geometric Inverse Design Algorithm

The best-fit optimization developed in this work is a form of geometric inverse design. Instead of specifying aerodynamic targets, a reference airfoil geometry is provided, and the algorithm computes the physical design variables that best reproduce it. The method employs a best-fit loop between the given airfoil geometry and the one constructed using the custom Bézier-based parameterization. To this end, a custom C++ code was implemented, which will be presented in the following chapter, together with its mathematical formulation, the optimization and initialization procedures, as well as the resulting approximations. Such a tool is highly valuable for blade airfoil designers, as engineers are often required to improve an existing design with respect to a set of constraints. To achieve this improvement, the reference design must first be represented using an appropriate set of design variables, after which the combination of variables leading to an optimized design can be determined.

3.1 Formulation of the Best-Fit Loop

The objective of this loop is to determine the appropriate design vector $\vec{b}$ that, when applied using the aforementioned custom Bézier-based parameterization method, produces a blade airfoil that best approximates the given target geometry. Since the target blade airfoil is defined by a set of coordinates (x_i^T, y_i^T), a metric is required to quantify the deviation between these points and the corresponding points (x_i^B, y_i^B) of the generated airfoil. For this purpose, the squared Euclidean distance (L2 norm), commonly referred to as the *Sum of Squared Errors (SSE)*, is employed:

$$\mathrm{F} = \sum_{i=0}^{N} \left[\left(x_i^B - x_i^T\right)^2 + \left(y_i^B - y_i^T\right)^2 \right] \tag{3.1}$$

Due to the fact that the blade airfoil, as defined by the custom Bézier-based parameterization, is divided into four segments (leading edge, trailing edge, pressure side, and suction side), four individual metrics of the form given in Equation (3.1) are computed. The overall metric is then obtained as:

$$F = F_{SS} + F_{LE} + F_{PS} + F_{TE} \tag{3.2}$$

The design vector comprises a total of 25 design variables that must be determined in order for the parameterization to generate an airfoi:

$$\vec{b} = \begin{bmatrix} \text{chord} & \alpha_1' & \alpha_2' & d_{\text{in}} & d_{\text{out}} & x_{c,LE} & y_{c,LE} & r_{LE} & \theta_{0,LE} \\ \theta_{3,LE} & x_{c,TE} & y_{c,TE} & r_{TE} & \theta_{0,TE} & \theta_{3,TE} & a_{0s} & a_{1s} & a_{2s} \\ a_{3s} & a_{4s} & a_{0p} & a_{1p} & a_{2p} & a_{3p} & a_{4p} & & \end{bmatrix}^T \tag{3.3}$$

In other words, equation (3.2) represents the target-cost function that must be minimized for a given set of design variables (3.3) in order to achieve the best possible match between the two blade airfoils. To this end, a gradient-based optimization loop was developed in C++.

It should be emphasized that this optimization process differs fundamentally from the one performed using the PUMA software. The former aims to determine the geometry that best approximates the reference airfoil by means of the custom Bézier-based parameterization method, without the use of CFD analysis. In contrast, the latter identifies the geometry that best satisfies a prescribed cost function through a CFD-based optimization. For clarity, the first approach will henceforth be referred to as *airfoil best-fit optimization*, while the second will be referred to as *CFD optimization*. Importantly, the best-fit optimization provides a reliable initial solution, which can subsequently serve as the starting point for the CFD optimization.

The method chosen to minimize the cost function is the Steepest Descent method, which is mathematically expressed as:

$$\vec{b}_{k+1} = \vec{b}_k - \eta \left.\frac{dF}{d\vec{b}}\right|_{\vec{b}_k} \tag{3.4}$$

where $\vec{b}_k$ is the current vector of design variables, $\vec{b}_{k+1}$ is the updated vector after one step, F is the objective function, $\left.\frac{dF}{d\vec{b}}\right|_{\vec{b}_k}$ of the cost function with respect to the design variables, evaluated at the current iteration k (a vector pointing in the direction of steepest increase of F), and η is the step size or learning rate (a positive scalar that determines the step length along the direction of the negative gradient).

The gradient of the objective function with respect to the design variables is computed analytically to achieve the highest possible accuracy. This is feasible due to the availability of the explicit expression for the Bézier curves and it is formulated as:

$$\nabla F = \begin{bmatrix} \frac{\partial F}{\partial b_1} \\ \frac{\partial F}{\partial b_2} \\ \vdots \\ \frac{\partial F}{\partial b_M} \end{bmatrix} = 2 \sum_{i=0}^{N} \begin{bmatrix} (x_i^B - x_i^T)\frac{\partial x_i^B}{\partial b_1} + (y_i^B - y_i^T)\frac{\partial y_i^B}{\partial b_1} \\ (x_i^B - x_i^T)\frac{\partial x_i^B}{\partial b_2} + (y_i^B - y_i^T)\frac{\partial y_i^B}{\partial b_2} \\ \vdots \\ (x_i^B - x_i^T)\frac{\partial x_i^B}{\partial b_M} + (y_i^B - y_i^T)\frac{\partial y_i^B}{\partial b_M} \end{bmatrix} \tag{3.5}$$

Their evaluation is very fast, even for large matrices with many design variables, as they mostly consist of polynomial and trigonometric expressions that require minimal computational time. Again because the airfoil consists of four segments, the corresponding gradients must be computed individually for each segment. The overall gradient is then obtained as the summation of the four segment contributions as in equation (3.6)

$$\nabla F = \nabla F_{SS} + \nabla F_{LE} + \nabla F_{PS} + \nabla F_{TE} \tag{3.6}$$

At this point, it should be noted that the camber line does not contribute explicitly to the computation of the gradient. This is because it is not directly compared to any reference points, as the given blade airfoil coordinates describe only the airfoil geometry and not the camber line itself. However, as the camber line is coupled with the generation of the suction and pressure side control points, it contributes implicitly to the gradient. In fact, its contribution is significant, as it strongly influences the curvature and overall camber of the resulting airfoil.

The analytical expression of the term $\frac{\partial x_i^B}{\partial b_1}$ in equation (3.5) is derived from direct differentiation of the Bézier curve formulation. In other words, since the x-coordinate is represented by a Bézier curve, it can be expressed as:

$$X(t) = \sum_{i=0}^{n} B_{i,n}(t)\, P_i, \quad t \in [0,1] \tag{3.7}$$

so its partial derivative with respect to the design variables is obtained through the equation

$$\frac{\partial X(t)}{\partial b_j} = \sum_{i=0}^{n} B_{i,n}(t)\, \frac{\partial P_i}{\partial b_j}, \quad t \in [0,1] \tag{3.8}$$

This is the case to produce the partial derivatives of the Bézier curves with respect to the design variables for the camber line and the leading and trailing edge areas. When it comes to the suction and pressure side, distinction arises because of the coupling betwenn the inner control points of the first two and the produced points of the camber line as seen in equation (3.9) (for more details see equation (2.7) and (2.8)):

$$P_{x,i}^{SS,PS} = \mathbf{X}^{MCL}(u_i) \pm \tfrac{1}{2}\, t(k_i)\, \vec{n}(u_i), \quad i = 3, \ldots, n_b^{SS} - 4 \tag{3.9}$$

Due to this correlation, the chain rule must be applied, resulting in the following expression for the partial derivatives:

$$\frac{\partial X(t)^{SS,PS}}{\partial b_j} = \frac{\partial \mathbf{X}^{MCL}(u_i)}{\partial b_j} \pm \frac{1}{2} \left[\frac{\partial t(k_i)}{\partial b_j}\, \vec{n}(u_i) + t(k_i)\, \frac{\partial \vec{n}(u_i)}{\partial b_j} \right], \quad i = 3, \ldots, n_b^{SS} - 4 \tag{3.10}$$

The exact same principles apply for the y coordinate. Further details, including the explicit computation of the partial derivatives of each segment with respect to the design variables, are provided in Appendix A.

To sum up, the optimization loop for the best-fit procedure begins with the initialization of the design vector. A blade airfoil is then generated using the custom Bézier-based parameterization. The objective (cost) function is evaluated by comparing the generated airfoil with the target blade geometry and the analytical derivatives of the cost function with respect to the design variables are subsequently computed, after which the design vector is updated. This process is iteratively repeated until a specified convergence criterion is satisfied, either based on a maximum number of iterations or when the gradient norm approaches zero.. The flowchart of the best-fit algorithm for the inverse design is shown in Figure 3.1.

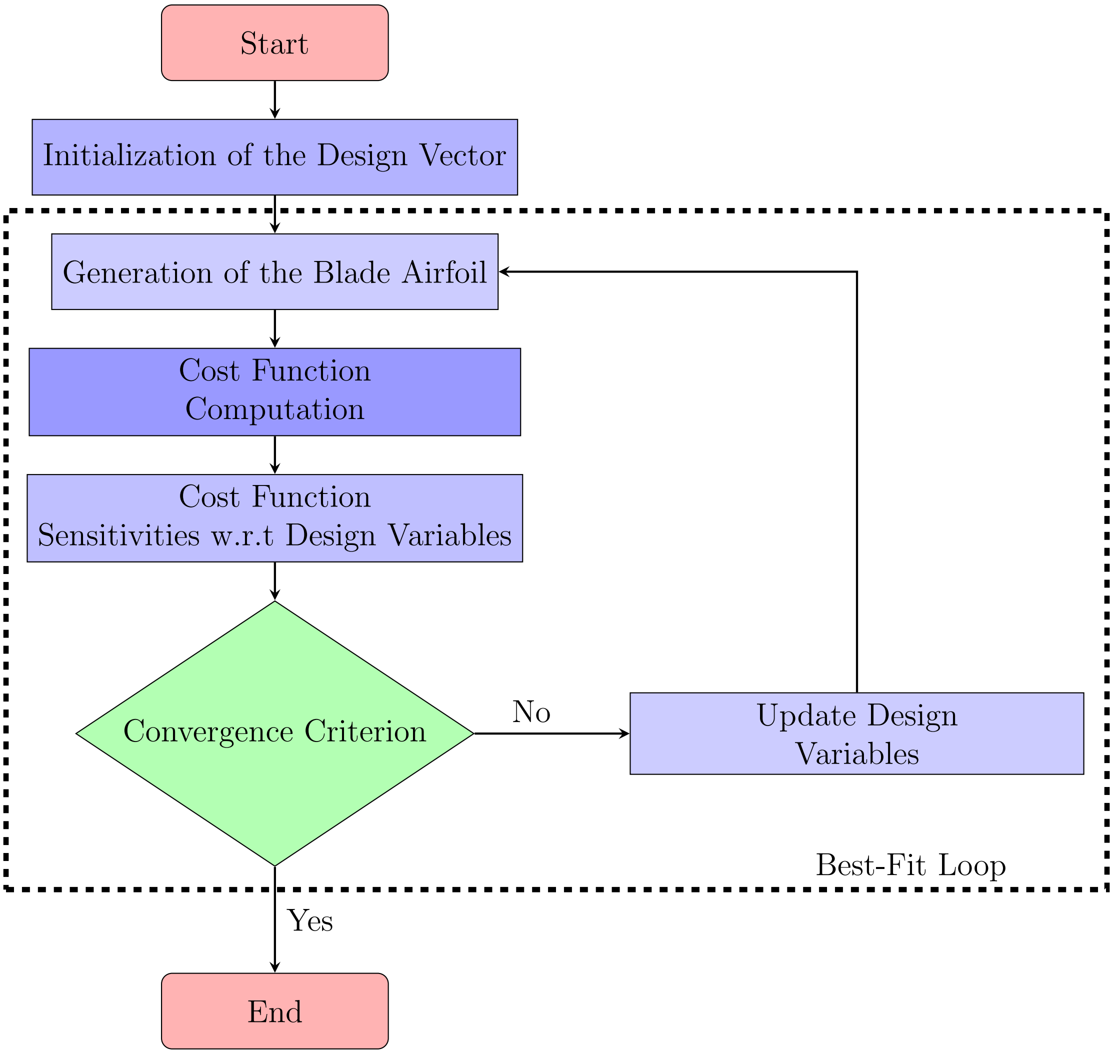

Start
Initialization of the Design Vector
Generation of the Blade Airfoil
Cost Function Computation
Cost Function Sensitivities w.r.t Design Variables
Convergence Criterion
No
Update Design Variables
Best-Fit Loop
Yes
End

Figure 3.1: *Flowchart of the Best-Fit optimization.*

3.2 Initialization Techniques

Given the fact that the design variables in the custom Bézier-based blade airfoil parameterization have pure physical meaning, selecting appropriate initial values for the best-fit optimization loop can be challenging. This task is further complicated by the fact that gradient-based algorithms, such as the Steepest Descent method employed in the loop, are highly sensitive to the initialization, which significantly influences both the convergence behavior and the local optimum reached. Therefore, it is critical to adopt robust initialization techniques. In this context, specific techniques for initializing the mean camber line and the circular arcs defining the leading and trailing edge regions are proposed.

Starting with the mean camber line, it serves a *baladeur* role during the best-fit loop, as it does not have a direct target for comparison and it implicitly, but strongly, affects the gradient. Moreover, as it is the stepping stone for the other segments to be built upon, it is important to provide an accurate initial shape. For a two-dimensional blade airfoil, the mean camber line, based on the classical definition, is the curve that lies halfway between the upper and lower surfaces at each chordwise location x, as seen in equation (3.11):

$$y_c(x) = \frac{y_u(x) + y_l(x)}{2} \tag{3.11}$$

where $y_c(x)$ = ordinate of the mean camber line at chordwise position x, $y_u(x)$ = ordinate of the upper surface at position x, $y_l(x)$ = ordinate of the lower surface at position x.

At first the user must split the target blade airfoil into an upper and lower surface and using the euation (3.11) to extract a mean camber line. After that a preliminary best-fit loop is performed between only the parameterized mean camber line and the extracted one. The outcome of this process provides a well-suited initialization set of values for the design variables of the mean camber line.

Moving to the initialization of the design variables of the circular arcs that represent the leading and trailing edge area, an algebraic circle fitting technique was implemented, known as *Kåsa method (1976)* [17], which will be explained in the following lines.

The equation of a circle in Cartesian coordinates is given by:

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \tag{3.12}$$

where (x_c, y_c) denotes the center of the circle and r its radius. Expanding and rearranging terms, this can be written as:

$$x^2 + y^2 - 2x_c x - 2y_c y + (x_c^2 + y_c^2 - r^2) = 0 \tag{3.13}$$

Defining

$$A = -2x_c, \quad B = -2y_c, \quad C = x_c^2 + y_c^2 - r^2 \tag{3.14}$$

the circle equation becomes linear in the unknowns (A, B, C):

$$x^2 + y^2 + Ax + By + C = 0 \tag{3.15}$$

Given a set of n data points, the Kåsa method [17] determines the parameters A, B, C by minimizing the sum of squared algebraic distances:

$$\min_{A,B,C} \sum_{i=1}^{n} \left(x_i^2 + y_i^2 + Ax_i + By_i + C \right)^2 \tag{3.16}$$

The minimization leads to the following linear system:

$$\begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & n \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = - \begin{bmatrix} \sum x_i(x_i^2 + y_i^2) \\ \sum y_i(x_i^2 + y_i^2) \\ \sum (x_i^2 + y_i^2) \end{bmatrix} \tag{3.17}$$

Solving this system yields the coefficients A and B, from which the circle center is recovered as:

$$x_c = -\frac{A}{2}, \quad y_c = -\frac{B}{2} \tag{3.18}$$

Finally, the circle radius is obtained from:

$$r = \sqrt{x_c^2 + y_c^2 - C} \tag{3.19}$$

After fitting a circle with center (x_c, y_c) and radius r, any point (x_i, y_i) on or near the circle can be expressed in polar coordinates relative to the circle center:

$$x_i = x_c + r\cos\theta_i, \quad y_i = y_c + r\sin\theta_i \tag{3.20}$$

where θ_i is the angle of the point relative to the circle center, measured from the positive x-axis.

The angle θ_i for a given point (x_i, y_i) is computed using the function:

$$\theta_i = \arctan(y_i - y_c,\, x_i - x_c) \tag{3.21}$$

By substituting the coordinates of the first and last points of each circular arc into equation (3.21), the starting angle θ_0 and the ending angle θ_3 of the arc can be obtained.

The Kåsa method provides a robust initial estimate for the design variables of the circular arcs. However, it minimizes the algebraic distance rather than the true geometric distance between the points and the circle. This is why these design variables are recalculated during the best-fit optimization loop.

3.3 Best-Fit Results

As targets, the blade airfoils selected for this study correspond to the geometries of the stator and rotor blades of a single-stage turbine. In both cases, during the best-fit procedure, the uncoupled system of minimizing each target function (see Equation (3.2)) was solved. This means that when optimizing one pair of similarly parameterized surfaces, the other two surfaces were held fixed. In other words, after obtaining a suitable initialization for the mean camber line, the leading and trailing edges were approximated while keeping the suction and pressure sides frozen and then the process was reversed for the remaining sides of SS and PS.

Another thing that must be declared is how the difference in points distribution was managed. The given blade airfoil is typically provided with an arbitrary discretization, whereas the parameterized geometry is generated with a user-defined one (i.e. uniform distribution). A direct point-to-point comparison based on the index of each point would produce misleading results, as points from different physical regions of the airfoil could be incorrectly matched. To solve this issue, the *closest-neighbor approach* was employed. For every point on the generated geometry, the method searches the reference geometry and identifies the point that lies closest in space. The distance between these two points is then recorded as the local error. Repeating this procedure for all generated points and summing the results returns the cost function.

For the rotor blade airfoil approximation the total number of given points was 456 and the convergence of the best-fit loop for each side and for the total blade airfoil is presenteed in Figure 1.

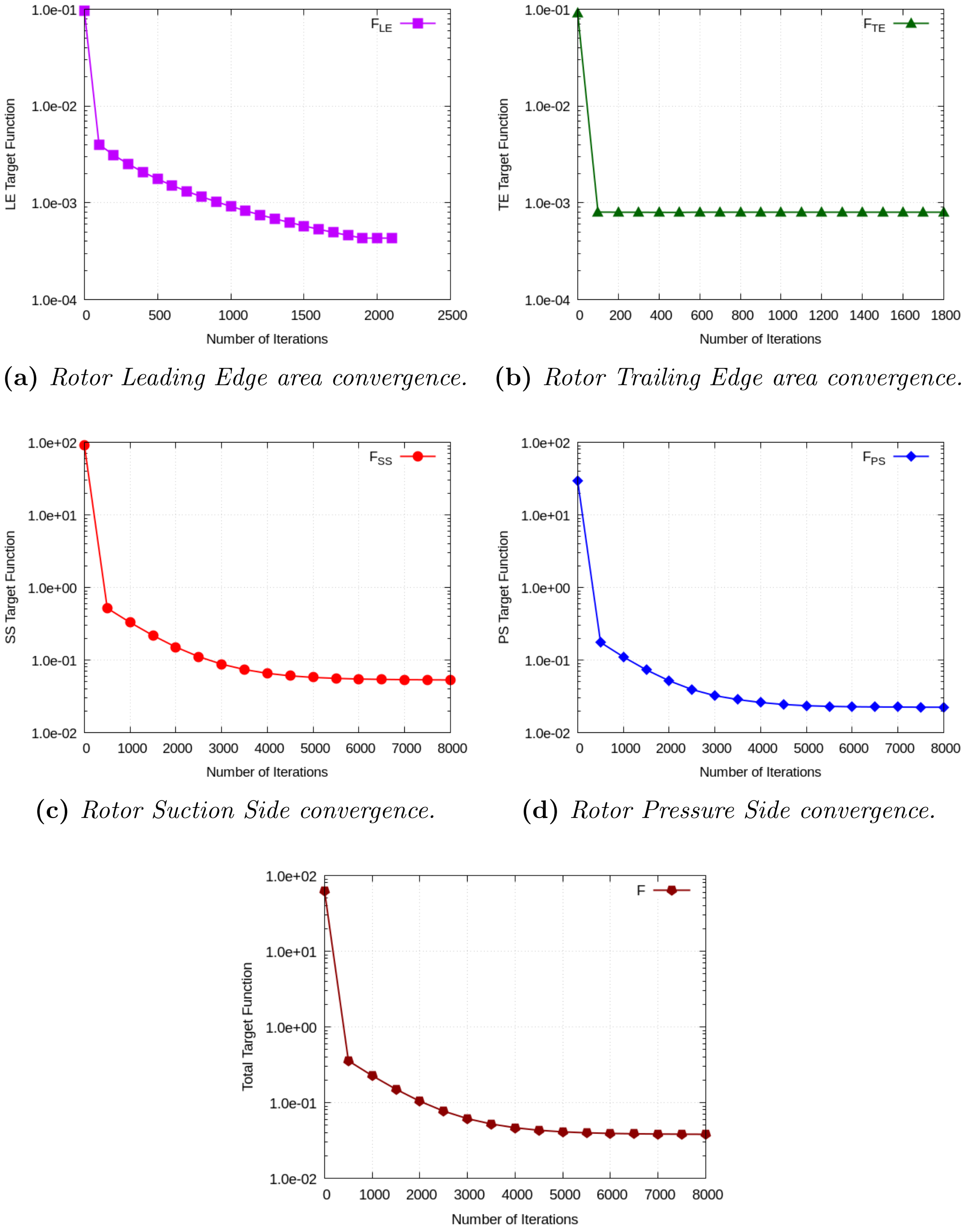
As shown in Figures 3.2a and 3.2b, the leading edge requires more iterations to converge compared to the trailing edge, but it achieves a smaller final error (4.30×10^{-4} versus 7.935×10^{-4}). Additionally, the convergence trajectories differ at the trailing edge, the error decreases sharply at the beginning, whereas at the leading edge, the decrease is smoother. These differences can be explained by the capability of the circular arc model to approximate these regions. As illustrated in Figure 3.3, the leading edge has a shape more like an elliptical arc rather than a circular one (unlike the trailing edge), which explains why convergence is slower at the start.

The suction and pressure sides present similar convergence paths (see Figures 3.2c and 3.2d), which validates that the proposed 4th-degree polynomial model for the thickness distribution is equally capable of representing both sides. The fact that the pressure side achieves a smaller error (2.236×10^{-2} versus 5.30×10^{-2} for the suction side) arises from its smaller camber, as it is obvious from Figure 3.3. The total error in approximating the rotor blade airfoil using the custom Bézier-based parameterization is 3.8×10^{-2}, demonstrating the accuracy of the model.

Table 3 presents a summary of the results obtained from the best-fit loop.

Side	**Number of points**	**Cost Function F**
LE	23	4.300×10^{-4}
TE	20	7.935×10^{-4}
SS	264	5.300×10^{-2}
PS	149	2.236×10^{-2}
Total	456	3.800×10^{-2}

Table 3.1: *Number of points and corresponding errors for each segment of the rotor blade airfoil and for the total.*



(a) *Rotor Leading Edge area convergence.*

(b) *Rotor Trailing Edge area convergence.*

(c) *Rotor Suction Side convergence.*

(d) *Rotor Pressure Side convergence.*

(e) *Rotor Cost function convergence.*

Figure 3.2: *Convergence of the objective function for each segment of the rotor blade airfoil and for the total blade airfoil.*

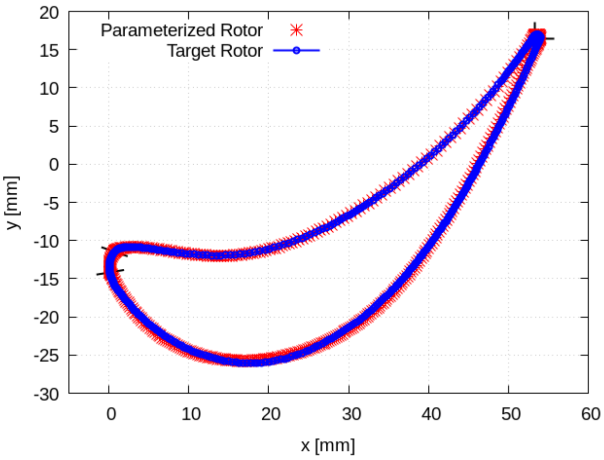


Figure 3.3: *Approximation of the rotor blade airfoil using the custom Bézier-based parameterization.*

The total number of given points for the stator inverse design procedure was 583. The convergence behavior of the best-fit loop for each segment, as well as for the complete blade airfoil, is illustrated in in Figure 3.4.

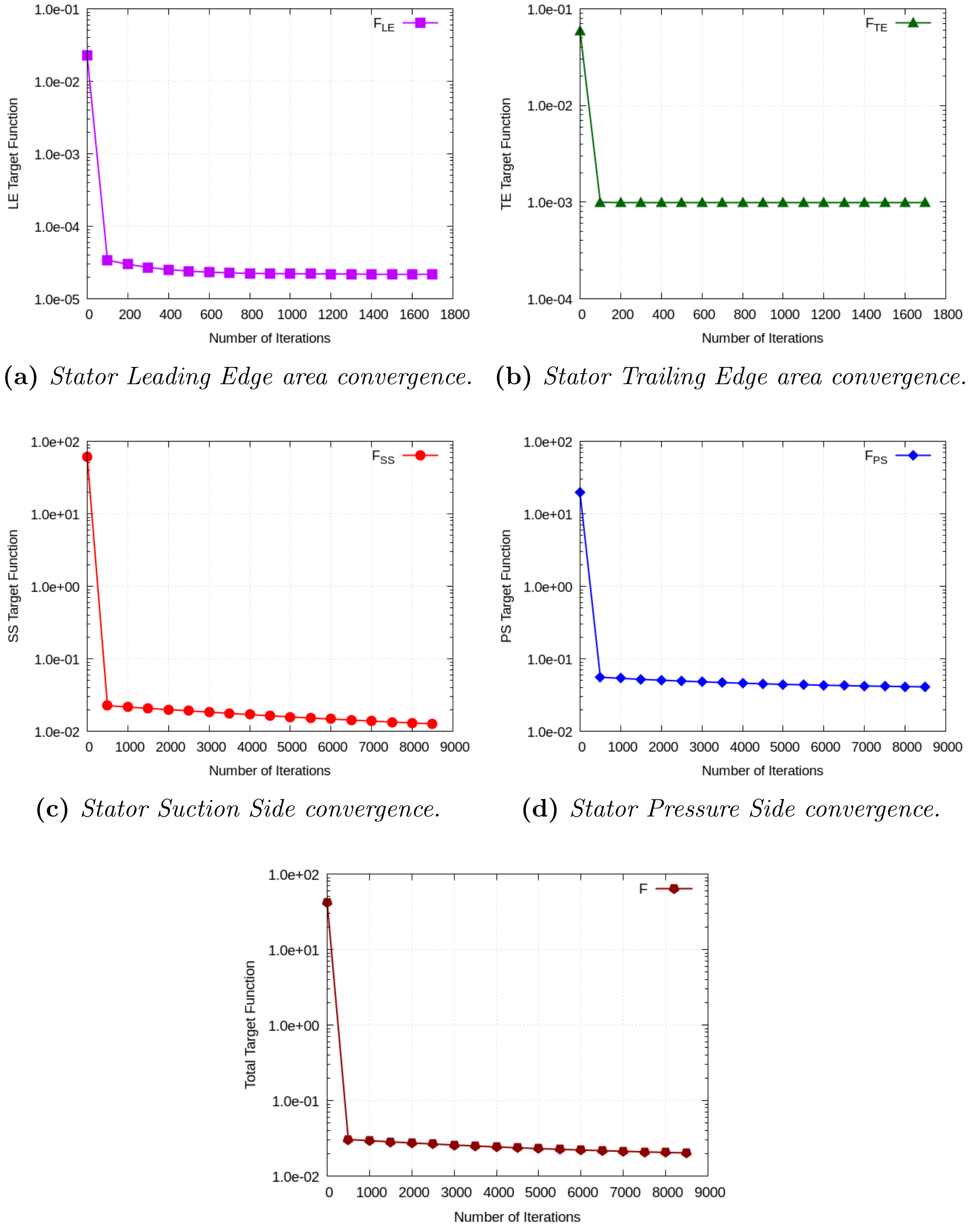
In this case, the convergence progress between the leading edge and trailing edge (Figures 3.4a and 3.4b) is more similar which indicates that the leading edge more closely resembles a circular arc, although it is still less circular than the trailing edge. Notably, again, the model achieves higher accuracy for the region with a more elliptical shape compared to the more circular region, with errors of 2.188×10^{-5} and 9.931×10^{-4}, respectively.

Regarding the suction and pressure side, their convergence behavior is largely similar (see Figures 3.4c and 3.4d), establishing that the 4th-degree polynomial model for the thickness distribution effectively represents both sides. This time, susction side results in a smaller error compared to the pressure side (1.265×10^{-2} versus 4.097×10^{-2} respectively), which can be justified by the fact that the curvature of the pressure side seems to slightly change near the chord midspan of the blade airfoil, as seen in Figure 3.5. Overall, the total error of the geometric invesre design procedure of the stator blade airfoil using the custom Bézier-based parameterization remains low, reflecting the accuracy of the proposed modeling approach.

The results produced by the best-fit loop are gathered in Table 3.2.

Side	Number of points	Cost Function F
LE	40	2.188×10^{-5}
TE	22	9.931×10^{-4}
SS	341	1.265×10^{-2}
PS	180	4.097×10^{-2}
Total	583	2.020×10^{-2}

Table 3.2: *Number of points and corresponding errors for each segment of the stator blade airfoil and for the total.*



(a) *Stator Leading Edge area convergence.*

(b) *Stator Trailing Edge area convergence.*

(c) *Stator Suction Side convergence.*

(d) *Stator Pressure Side convergence.*

(e) *Stator Cost function convergence.*

Figure 3.4: *Convergence of the objective function for each segment of the stator blade airfoil and for the total blade airfoil.*

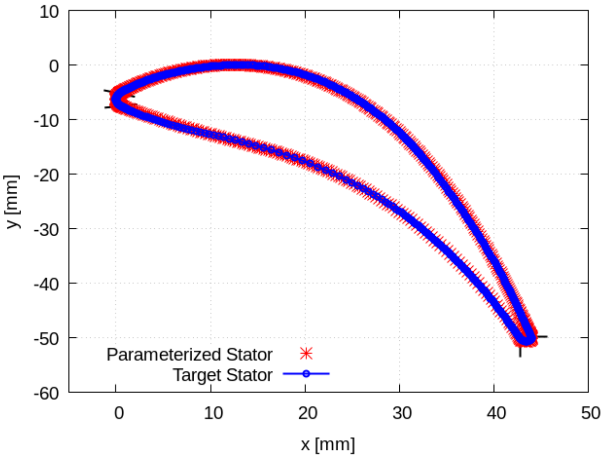


Figure 3.5: *Approximation of the stator blade airfoil using the custom Bézier-based parameterization.*

Overall, the accuracy of the custom Bézier-based blade airfoil parameterization model is sufficient to proceed with the fitted blade for aerodynamic shape optimization, where its advantage in imposing geometric constraints will be demonstrated. Another key benefit of this parameterization method is that, once the blade airfoil geometry has been obtained, the distribution of points along the airfoil can be re-adjusted. This capability is particularly important in optimization and, more generally, in CFD applications, where the quality of the mesh around the blade airfoil strongly depends on the number of points and their distances.

Chapter 4

CFD Analysis - Validation

PUMA is an in-house, GPU-enabled flow solver developed by the *PCOpt/NTUA* research group and has been extensively tested across a wide range of turbomachinery cases [29], [30]. Given the availability of experimental data for the single-stage turbine from which the stator and rotor blade airfoil geometries were derived, it was considered essential to perform a CFD analysis as a means of validating the numerical model of both the reference and the fitted stage geometries, prior to proceeding with the aerodynamic shape optimization. The comparison between numerical predictions and experimental measurements provides confidence in both the accuracy of the solver and the fidelity of the adopted CFD methodology. Therefore, this chapter presents the single-stage turbine configuration under study, together with the computational mesh and the details of the CFD setup. The analysis concludes with a validation against the experimental data, thereby establishing a reliable foundation for the aerodynamic shape optimization described in the subsequent chapters.

4.1 Introduction

The test case that was use is a single-stage axial turbine (in some experimental setups it is also referred to as a one-and-a-half-stage turbine, stator–rotor–stator [28]), which was developed and investigated around 1995 by the Institute of Jet Propulsion and Turbomachinery – IST of the Department of Mechanical Engineering at RWTH Aachen University, Germany [34], with the aim of studying three-dimensional unsteady flows in turbomachinery.

The turbine was constructed using blades without twist in both the stator and rotor, with a constant hub and tip diameter along the entire blade span. The stator employs a Traupel distribution, with its blades stacked radially, one on top of the other at the trailing edge, whereas the rotor was designed based on a modified model from the Von Karman Institute of the same period, with its blades positioned in a radial alignment relative to their center of gravity.

Experimental measurements were acquired at the midspan plane of the blades. Specifically, the measurement instruments were placed 8.8 mm downstream of each blade along the axial direction, with the stator positioned 15 mm upstream of the rotor. The main blade dimensions are provided in detail in Table 4.1, while a schematic representation of the turbine blade arrangement, including the relevant distances (in mm) and the corresponding velocity triangles, is shown in Figure 4.1.

Parameter	**Stator**	**Rotor**
Chord [mm]	62	60
Number of Blades	36	41
Blade Pitch (midspan) [mm]	47.6	41.8
Pitch-to-Chord Ratio	0.77	0.67
Rotational Speed [rpm]	–	3500

Table 4.1: *Main geometric and operational parameters of the stator and rotor blades.*

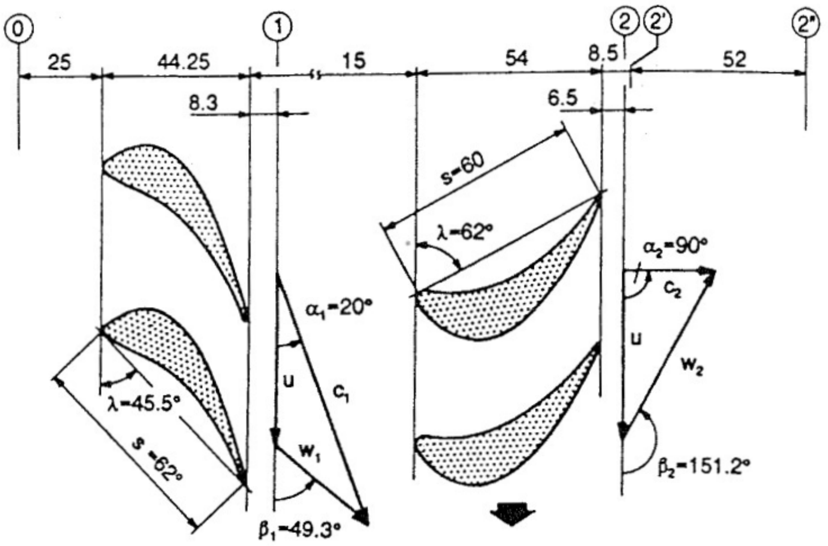


Figure 4.1: *Representation of the blading of the single-stage axial turbine. All the indicated distances and dimensions are given in mm [34].*

4.2 Mesh Generation

For the construction of all the meshes, the software *Fidelity Pointwise* (by Cadence) [21] was employed. A *block-structured mesh* was selected for both blade rows and particular emphasis was placed on achieving a high-fidelity, blade-centered, topology around the blades in order to ensure solver stability.

Intentionally, the rotor mesh is presented first, since it is more demanding to generate compared to the stator mesh. The main reason for this lies in the relatively high *stagger angle* (28°) of the rotor blade in combination with its high camber in the front. In practice, this corresponds to a pronounced curvature ("cambering") on the suction side, where the blade bends significantly and relatively sharply, complicating the mesh generation process in this region. The difficulty arises from the requirement that the mesh must accommodate this curvature while preserving its orthogonality as much as possible. Any excessive skewness would otherwise result in increased numerical dissipation and diffusion during the solution of the Navier–Stokes equations, thereby introducing errors in the results.

To address this issue, the mesh topology was carefully studied and divided into smaller blocks, allowing the preservation of orthogonality in regions of high curvature. The block separation lines (split lines) were placed after several trials, with the criterion being the minimization of skewness. Furthermore, near the blade walls, the mesh was refined. Using the *hyperbolic extrusion method*, a boundary-layer mesh was generated with a geometric growth rate of 1.1 and an initial wall-normal spacing of $\Delta s = 10^{-4}$ mm. This ensured improved resolution of near-wall flow features, minimizing dissipation and diffusion errors. Additionally, the *periodic boundaries* over suction and pressure sides were defined, along with the *inlet* and *outlet* flow regions, in accordance with the prescribed distances (see schematic in Figure 4.1).

The resulting mesh is shown in Figure 4.2. The two-dimensional grid consists of 31,354 nodes and is of high quality, with a *minimum included angle* of 34°. The minimum included angle is defined as the angle formed between one grid line along which a curvilinear coordinate (ξ_i) varies (while the other remains constant) and another grid line along which the second curvilinear coordinate (ξ_j) varies (with the first remaining constant).

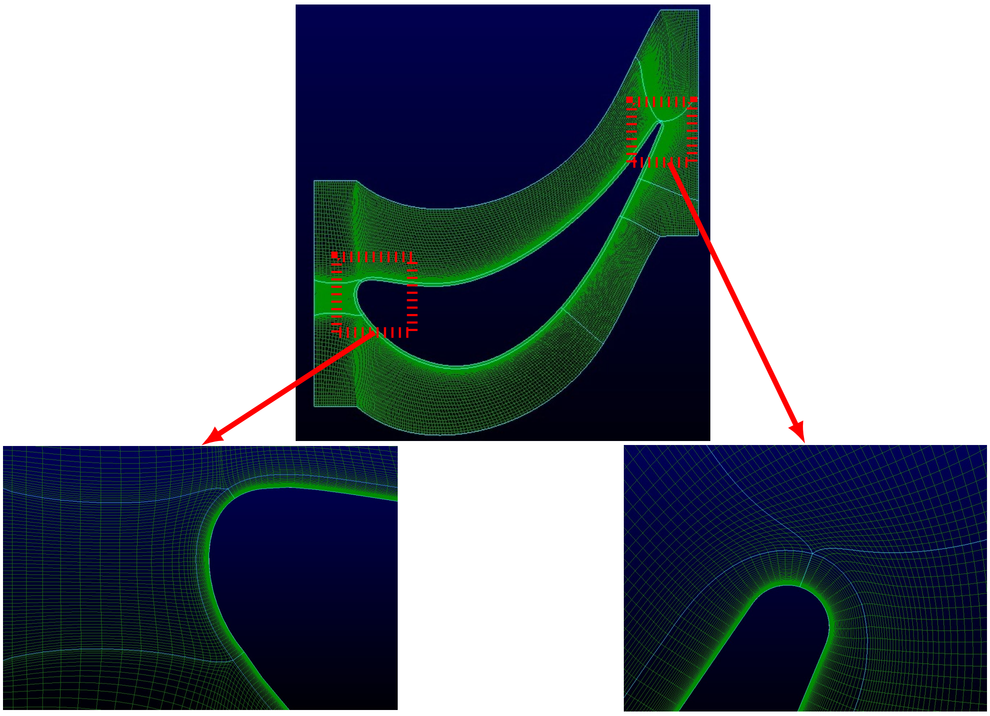


Figure 4.2: *The mesh of the rotor blade, generated using Fidelity Pointwise (top), with a zoomed-in view of the leading-edge region (bottom left) and the trailing-edge region (bottom right).*

Regarding the generation of the stator blade mesh, an entirely similar procedure to that of the rotor was followed. Initially, denser layers were created around the blade walls with the same growth rate (equal to 1.1) and the same initial spacing from the wall ($\Delta s = 10^{-4}$ mm), again using the hyperbolic extrusion method. In contrast to the rotor blade, the stator blade exhibits lower curvature-camber along its length. Nevertheless, at its trailing edge it becomes nearly vertical, maintaining a large inclination throughout its span (high stagger angle of 44.5°). This characteristic leads the structured mesh to distortion and skewness as it attempts to conform to the blade shape.

For this reason, as in the rotor case, the mesh topology was appropriately modified through the addition of split lines, in order to prevent distortion. Finally, the periodic flow boundaries were defined on both sides of the blade, along with the inlet and outlet regions, in accordance with the prescribed distances (see schematic representation in Figure 4.1).

The resulting mesh is shown in Figure 4.3. It consists of 46,600 nodes (in two dimensions) and is of high quality, with a minimum included angle of 28°.

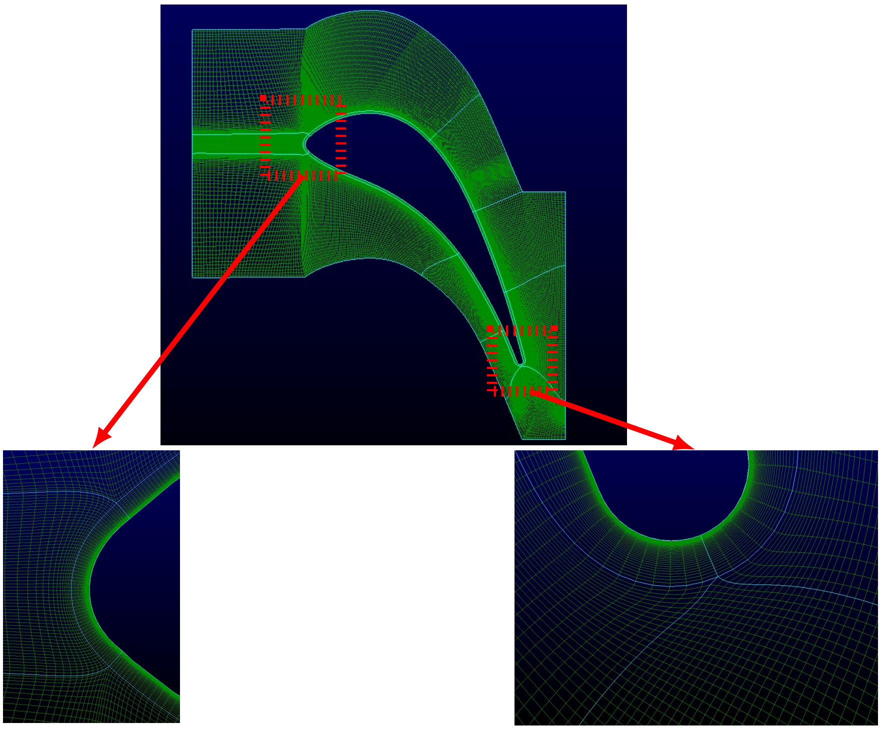


Figure 4.3: *The mesh of the stator blade, generated using Fidelity Pointwise (top), with a zoomed-in view of the leading-edge region (bottom left) and the trailing-edge region (bottom right).*

Property	**Value**
Gas constant R [J/kg·K]	287
Specific heat ratio γ	1.33
Dynamic viscosity μ [Pa·s]	1.716×10^{-5}

Table 4.2: *Gas properties.*

4.3 Results Validation

Having generated the two individual meshes of the single stage (stator and rotor blades), the PUMA software was employed to solve the fully resolved two-dimensional compressible RANS equations, along with the one equation low-Reynolds Spalart-Allmaras turbulence model. In the fluid properties section, the values presented in Table 4.2 were specified for the turbine gas.

In the flow model, parameters describing the flow characteristics were specified. In this application, the flow is turbulent, and the turbulence was modeled using the Spalart–Allmaras model. This is a one-equation turbulence model, with the low-Reynolds-number formulation applied without wall functions, since both of the generated meshes achieve $y^+ < 1$ (for these flow conditions). The model computes the turbulent viscosity μ_t at each iteration [27].

Based on experimental data, the inlet conditions specified total pressure, total temperature, and flow angles, while the Mach number was set as an extrapolated variable. The spatial distribution of these variables was chosen to be uniform across the inlet section, as shown in Table 4.3.

Variable	**Symbol**	**Value**
Total Pressure	p_t	158545.44 [Pa]
Total Temperature	T_t	309.12 [K]
Flow Angles	α, β	0
Viscocity Ratio	μ_r	20

Table 4.3: *Inlet boundary conditions for the CFD model.*

At the outlet, a fixed static pressure was imposed, while the static temperature and velocity components were treated as extrapolated variables. Similar to the inlet, their spatial distribution was defined as uniform. The conditions are gathered in Table 4.4.

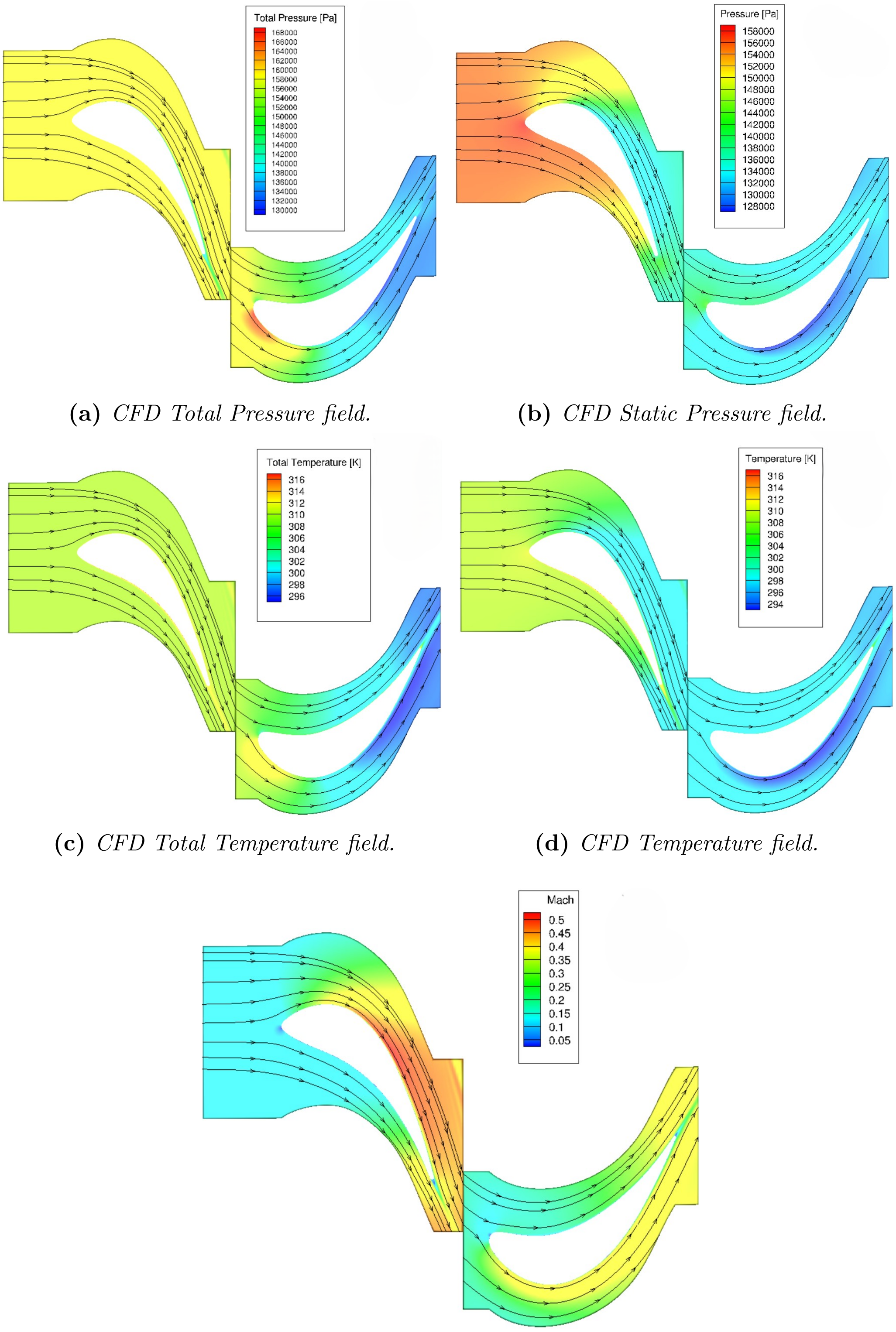
Variable	**Symbol**	**Value**
Pressure	p	130300 [Pa]

Table 4.4: *Outlet boundary conditions for the CFD model.*

The walls of both the stator and rotor blades were modeled as adiabatic. That is, during the solution, the code neglects heat transfer between the walls and the fluid. Additionally, a no-slip boundary condition was applied to the walls, enforcing zero relative velocity. It should be noted that, in the rotor, the velocity of interest is the relative velocity, whereas in the stator it is the absolute velocity V. Furthermore, to model the motion of the rotor blades, the rotor mesh was considered moving with a translational velocity of approximately 100m/s along the negative Y-axis, corresponding to the 3500rpm rotation in the 2D analysis.

To solve the flow on the two meshes and enable information transfer between them, the mixing plane technique [10] was used. This is one of several stator–rotor interaction methods available in PUMA, which allows the two meshes to communicate while maintaining their independence. In this case, the conservative flux formulation of was applied, whereby the integrated fluxes of mass, momentum, and energy at the stator outlet are imposed as uniform inlet conditions at the rotor inlet.

The simulations were executed on GPU processors, with the full-stage case requiring about 5 minutes for 3,000 iterations on an NVIDIA Tesla K40m. The resulting contours of static and total pressure, static and total temperature, and Mach number are shown in Figure 5.



(a) *CFD Total Pressure field.*

(b) *CFD Static Pressure field.*

(c) *CFD Total Temperature field.*

(d) *CFD Temperature field.*

(e) *CFD Mach number field.*

Figure 4.4: *Contours of flow field variables from CFD: Static and Total Pressure, Static and Total Temperature, and Mach Number distributions.*

Note: To avoid confusion, it should be emphasized that in Figure 5c, the flow field coloring for the stator corresponds to the Mach number calculated based on the absolute flow velocity, whereas for the rotor depicts the relative Mach number (M_w). This explains why, at the interface between the two, the coloring is not continuous, as it should based on the mixing plane technique and the use of area-averaged values with a uniform spatial distribution, which were applied in the flow solution.

Along the turbine stage, both total and static pressures decrease, as seen in Figures 5a and 4.4b, at different rates, due to flow acceleration, as expected, while no discontinuities are observed in the streamlines (such as abrupt distortions, sudden changes in flow angles, or shock waves). A comparison of these results with the corresponding experimental measurements at stations 0, 1, and 2 is summarized in Table 4. Subscript 0 denotes the stage inlet, subscript 1 corresponds to the stator outlet, and subscript 2 to the rotor outlet (see Figure 4.1). The x-axis (horizontal) represents the axial flow direction, aligned with the turbine longitudinal axis, whereas the y-axis (vertical) represents the circumferential direction. Subscript t is used to denote total quantities. Additionally, Table 4 contains the stage total pressure ratio as obtained from both the experimental measurements and the PUMA computations.

Quantity	Experimental [Pa]	PUMA Reference [Pa]	PUMA Fitted [Pa]	Percentage Deviation Reference [%]	Percentage Deviation Fitted [%]
p_{t0}	1.585×10^5	1.585×10^5	1.585×10^5	0	0
p_0	1.574×10^5	1.565×10^5	1.565×10^5	0.575	0.575
p_{t1}	1.521×10^5	1.580×10^5	1.584×10^5	-3.833	-3.977
p_1	1.338×10^5	1.381×10^5	1.386×10^5	-3.164	-3.463
p_{t2}	1.326×10^5	1.327×10^5	1.331×10^5	-0.093	-0.37
p_2	1.303×10^5	1.303×10^5	1.303×10^5	0	0
PR $= p_{t0}/p_{t2}$	1.1941	1.1945	1.1881	0.030	0.034

Table 4.5: *Comparison of experimental and numerical pressure values at different turbine stations for both the reference and the fitted stage geometry.*

As shown in Table 4, the PUMA solver predicts the flow pressures along the stage with high accuracy. More specifically, all deviations, except those at station 1, are below 1%. It should be emphasized that the total pressure at station 0 was imposed as an inlet boundary condition, while the static pressure at the outlet of the rotor (station 2) was prescribed as an outlet boundary condition (see Table 4.4). Regarding the outlet of the stator, PUMA slightly underestimates the losses in both total and static pressure. This, is most likely explained by the modeling assumption of adiabatic stator blade walls. Nevertheless, the error remains very small (less than 4%) for both the reference and the fitted geometry.

In Figures 5b and 4.4d, is clear that the computed flow fields are qualitatively consistent with physical expectations. Along the turbine stage, both total and static temperatures decrease, due to the work exchange with the moving rotor, and the flow acceleration. Table 4.6 presents a comparison between the experimental temperature measurements and the PUMA predictions at stations 0, 1, and 2.

Quantity	Experimental [K]	PUMA Reference [K]	PUMA Fitted [K]	Percentage Deviation Reference [%]	Percentage Deviation Fitted [%]
T_{t0}	309.12	309.12	309.12	0.000	0.000
T_0	308.44	308.14	308.14	0.0970	0.0973
T_{t1}	307.30	309.19	309.84	-0.615	-0.8265
T_1	296.26	299.05	299.38	-0.942	-1.05
T_{t2}	296.08	296.53	296.79	-0.152	-0.239
T_2	294.60	295.22	295.22	-0.210	-0.210

Table 4.6: *Comparison of experimental and numerical temperature values at different turbine stations for both the reference and the fitted stage geometry.*

PUMA predicts values very close to the measurements, with divergence of both total and static temperatures remaining almost below 1% at all stations. However, it is noteworthy that the increase in temperature deviation from station 0 to station 1 is larger than the corresponding increase observed for pressure. This is explained by the fact that temperature takes on greater values. Nonetheless, if someone examines the increase of each quantity from stator's inlet to outlet, then it is clear that the temperature miscalculation gets 10 times bigger (from 0,09% to 0,9%) while pressure's only increased by almost 6 times (from 0,58% to 3,16%) which reveals the truth. The error caused by the adiabatic walls modelling affects indirectly the pressure and directly the temperature as it should. However, the error between the two geometries is very small.

The Mach number values at the selected stations, based on both measurements and PUMA computations, are presented in Table 4.7.

Quantity	Experimental	PUMA Reference	PUMA Fitted	Percentage Deviation Reference [%]	Percentage Deviation Fitted [%]
M_0	0.1056	0.1096	0.1096	-3.803	-3.803
M_1	0.4315	0.4628	0.4672	-7.259	-8.274
M_{w1}	0.1869	0.1998	0.2018	-6.902	-7.972
M_{w2}	0.1584	0.2267	0.2267	-4.613	-4.613

Table 4.7: *Comparison of experimental and numerical Mach numbers at different turbine stations for both the reference and the fitted stage geometry.*

The Mach number within the stage is predicted by PUMA with a divergence from the experimental value of 7% according to Table 4.7. Furthermore, the flow is characterized as subsonic and compressible (maximum Mach number 0.5), without the presence of oblique or normal shock waves.

For the stator, the streamlines follow the absolute flow angles, whereas for the rotor, they follow the relative flow angles. The exact values of these angles, as computed by PUMA and as measured in the experimental setup, are presented in Table 4.8.

Quantity	Experimental [deg]	PUMA Reference [deg]	PUMA Fitted [deg]	Percentage Deviation Reference [%]	Percentage Deviation Fitted [%]
α_0	0.0	0.0	0.0	0.000	0.000
α_1	-20.0	-20.229	-20.229	-1.143	-1.143
β_1	-49.3	-49.100	-48.900	0.406	0.8114
α_2	-90.0	-89.915	-89.87	0.095	0.144
β_2	-151.2	-150.520	-150.20	0.450	0.6613

Table 4.8: *Comparison of experimental and numerical flow angles at different turbine stations for both the reference and the fitted stage geometry.*

From Table 4.8, is made understood that PUMA accurately predicts all flow angles at all considered stations, with the maximum deviation slightly above 1%. Furthermore, analyzing the deviations of the flow angles in comparison with those of other computed quantities, no significant increase is observed at station 1, unlike what was previously noted due to the modeling of the stator wall thermal behavior. This is reasonable, as the flow angles depend solely on kinematic factors (blade geometry, flow density) and are not influenced by thermodynamic effects (heat transfer). In addition, no significant differences are observed between the reference and fitted geometries.

Chapter 5

Integration of the Custom Parameterization into the ShpO Cycle

In order to perform aerodynamic ShpO of the best-fitted blade airfoil obtained from the inverse design algorithm, the custom Bézier-based parameterization, the CFD solver, and the adjoint solver PUMA must be integrated in the optimization process. In this chapter, the outline of the used optimization cycle along with some more specifications on each step will be adressed.

5.1 ShpO Cycle

The first step for the optimization circle to start is to apply the best-fit algorithm to the target blade airfoil in order to replicate the baseline geometry, as described in detail in Chapter 3. Subsequently, the computational mesh is generated, and the initial CFD simulation is performed. As previously mentioned, the so-called *primal problem* requires the solution of the 2D compressible RANS equations, coupled with the one-equation low-Reynolds Spalart–Allmaras turbulence model.

The next step is the computation of the sensitivity derivatives. The method employed for this purpose is the *continuous adjoint method*, as seen in [32], [31], an efficient numerical technique that evaluates the derivatives of the cost function with respect to the design variables at a computational cost independent of the number of such variables. This represents its greatest advantage and is the primary reason for its selection in this diploma thesis. In CFD, the adjoint method is particularly well-suited for aerodynamic shape optimization, where performance metrics depend explicitly on the geometry of the computational domain. When the number of geometric design variables is large, sensitivity methods, such as finite differences, become computationally prohibitive. The continuous adjoint method is based on the augmented function F_{aug} from the objective function F, through the following equation [7]:

$$F_{aug} = F + \int_{\Omega} \Psi_i R_i \, d\Omega \tag{5.1}$$

where F denotes the objective function, Ω is the flow domain, R_i are the residuals of the governing flow equations (RANS), and Ψ_i are the adjoint variables

Differentiation of equation (5.1) with respect to the design variables yields:

$$\frac{\delta F_{aug}}{\delta b_n} = \frac{\delta F}{\delta b_n} + \frac{\delta}{\delta b_n} \int_{\Omega} \Psi_i R_i \, d\Omega \tag{5.2}$$

where b_n are the design variables, $n = (1, 2, \ldots, N)$ is the index of the design variables, and N is the total number of design variables.

The procedures of solving both the primal and the continuous adjoint problem using the Field Integral adjoint formulation (FI) are carried out using the PUMA software. For the computation of the sensitivity derivatives, external communication between the PUMA and the custom code for calulating the geometrical grid sensitivites was required. More specifically, the sensitivity derivatives of the cost function F at the boundary nodes of the blade airfoil are computed using the chain rule, according to the equation:

$$\frac{\delta F}{\delta b_n} = \frac{\delta F}{\delta x_i}\frac{\delta x_i}{\delta b_n} + \frac{\delta F}{\delta y_i}\frac{\delta y_i}{\delta b_n} + \frac{\delta F}{\delta z_i}\frac{\delta z_i}{\delta b_n} \tag{5.3}$$

where $\frac{\delta F}{\delta x_i}$, $\frac{\delta F}{\delta y_i}$, and $\frac{\delta F}{\delta z_i}$ are the sensitivity derivatives of the cost function F

with respect to the coordinates x_i, y_i, and z_i of the i-th node of the blade airfoil, while $\frac{\delta x_i}{\delta b_n}$, $\frac{\delta y_i}{\delta b_n}$, and $\frac{\delta z_i}{\delta b_n}$ are the sensitivity derivatives of the coordinates of the i-th node of the blade airfoil with respect to the design variables, obtained from the differentiation of the custom Bézier-based parameterization.

As shown in equation (2), the computation of the sensitivity derivatives of the objective function F at the boundary nodes requires the corporation of the sensitivity derivatives of the custom Bézier-based parameterization into the PUMA software. For this purpose, a code named `GRID_SENSITIVITIES .exe` was developed in C++, which computes the required derivatives and provides them to PUMA via a properly formatted file, so that the sensitivity derivatives of F at the boundary nodes can be computed. For the remaining nodes, the PUMA software employs the Elastic Medium technique to transfer the sensitivity derivative information of the boundary nodes of the blade airfoil, to all other nodes.

Once the sensitivity derivatives of the cost function have been computed for all nodes, the following steps are performed: the update of the design variables, according to equation (3), the generation of the new blade airfoil, and the adaptation of the mesh to the updated geometry. At this stage, further external intervention in PUMA is required in order to prescribe the displacement of the boundary nodes of the new mesh. For this purpose, the same code, `GRID_SENSITIVITIES.exe`, is executed. It reads the updated design variables from an input file (`designVars.dat`), reconstructs the geometry of the current blade airfoil using the custom Bézier-based parameterization, and computes the deformations of the nodes between the baseline and the current blade airfoil configuration. These displacements correspond to the mesh deformations at the boundary nodes and are provided to PUMA via a properly formatted file. Finally, PUMA once again employs the Elastic Medium technique to propagate the boundary-node displacement information throughout the computational mesh.

$$\vec{b}_{k+1} = \vec{b}_k - B_k^{-1} \frac{dF}{d\vec{b}}\Big|_{\vec{b}_k} \tag{5.4}$$

where, $\vec{b}_k$ is the current vector of variables at iteration k, $\frac{dF}{d\vec{b}}\big|_{\vec{b}_k}$ is the gradient of the objective function evaluated at $\vec{b}_k$, and B_k^{-1} is the inverse of the Hessian approximation

The employed ShpO algorithm is a Quasi-Newton method for gradient-based constrained nonlinear optimization problems. Specifically, the Sequential

Least Squares Quadratic Programming (SLSQP) method is used, which applies the BFGS scheme to approximate the Hessian matrix and uses a line search strategy to determine the step length. This ensures that each iteration proceeds along a descent direction that reduces the objective function while simultaneously satisfying the imposed constraints.

The flowchart of the ShpO process is shown in Figure 5.1 providing a general overview of the optimization procedure.

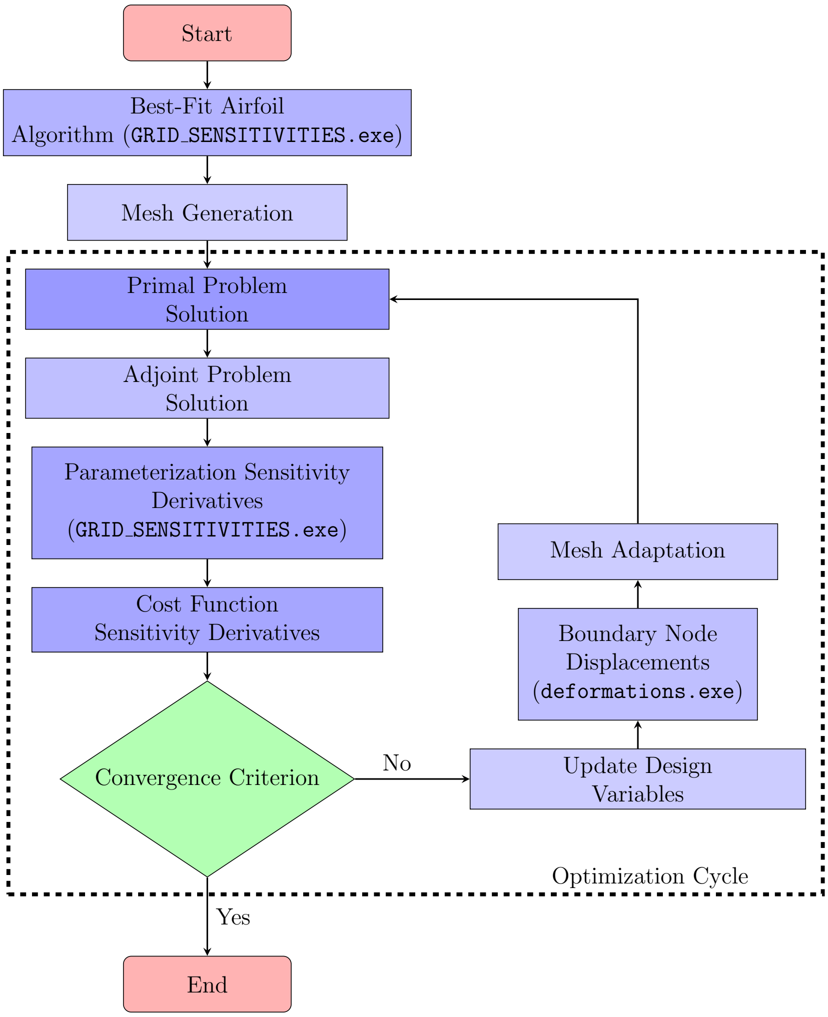

Start
Best-Fit Airfoil Algorithm (GRID_SENSITIVITIES.exe)
Mesh Generation
Primal Problem Solution
Adjoint Problem Solution
Parameterization Sensitivity Derivatives (GRID_SENSITIVITIES.exe)
Cost Function Sensitivity Derivatives
Convergence Criterion
No
Update Design Variables
Boundary Node Displacements (deformations.exe)
Mesh Adaptation
Optimization Cycle
Yes
End

Figure 5.1: *Flowchart of the Aerodynamic ShOp Airfoil Optimization Algorithm.*

Chapter 6

Aerodynamic ShpO Cases

Aerodynamic shape optimization (ShpO) of the airfoil representing the rotor blade in the single-stage turbine discussed in Chapter 4 was performed using a gradient-based algorithm. Specifically, the continuous adjoint method was employed through the PUMA software to minimize the mass-averaged total pressure losses in the airfoil cascade, subject to constraints on the exit flow angle and the inlet capacity across a set of test cases. This chapter provides a detailed discussion of the imposed flow and geometrical constraints for each case, presents the corresponding ShpO results, and offers a comparative analysis among the different cases.

6.1 Case Description

As previously mentioned, the target function to be minimized in the following aerodynamic ShpO cases is the mass-averaged total pressure loss. It quantifies irreversibilities and energy losses due to viscous and turbulent effects, boundary layers, shocks, secondary flows, and similar phenomena. Employing mass-averaging, ensures that the total pressure at each location is weighted by the local mass flux, so regions with higher flow contribute more to the average. Essentially, it represents the reduction in the fluid's ability to perform useful work as it passes through the component, accounting for the actual flow distribution. For a turbine rotor blade, it serves as a key performance indicator, since total pressure losses directly affect stage efficiency: higher losses reduce the energy extracted from the flow and degrade aerodynamic performance. Mathematically, it is defined as the difference between the mass-averaged total pressures at the inlet and outlet planes:

$$\Delta p_t^{\text{mass}} = \bar{p}_{t,\text{in}} - \bar{p}_{t,\text{out}} \tag{6.1}$$

where the mass-averaged total pressure at a plane is computed as

$$\bar{p}_t = \frac{\displaystyle\iint_A \rho\, V_n\, p_t\, dA}{\displaystyle\iint_A \rho\, V_n\, dA} \tag{6.2}$$

where, ρ is the local density, V_n is the velocity component normal to the surface, p_t is the local total pressure, and A is the surface over which the averaging is performed.

As far as the first constraint, the inlet capacity expresses the ability of a turbomachine blade row (e.g., a turbine rotor) to admit and process a given mass flow under specified inlet conditions. These conditions are determined by both geometric parameters (such as chord, pitch, and blade height) and flow parameters (including inlet flow angle, velocity, total pressure, and total temperature). For a turbine rotor blade, inlet capacity is particularly important in determining overall stage performance. It governs the amount of flow that the rotor can process for a given pressure ratio, directly influencing both efficiency and power output. At the same time, it defines whether the flow at the rotor throat becomes choked, establishing the upper limit on the mass flow that can enter the rotor. Furthermore, it must be carefully matched with the stator and with adjacent turbine stages to maintain proper flow alignment. Any mismatch in capacity can lead to increased aerodynamic losses, a reduction in efficiency, or even the onset of flow instabilities. Inlet capacity is defined in terms of the non-dimensional mass flow parameter, independently of the absolute operating conditions and directly comparable between different machines or operating points, as:

$$\text{Capacity} = \frac{\dot{m}\,\sqrt{T_t}}{p_t} \tag{6.3}$$

where, $\dot{m}$ is the inlet mass flow rate, T_t is the total (stagnation) temperature at the inlet, and p_t is the total (stagnation) pressure at the inlet.

The second constraint is the flow exit angle, which quantifies the flow deflection imposed by the blade geometry. In a stationary rotor blade, the exit angle represents the direction of the absolute velocity of the flow leaving the blade relative to the axial direction. It is a critical parameter for ensuring efficient flow redirection, minimizing aerodynamic losses, and properly matching the next stage. Misalignment of the flow can lead to increased incidence losses, reduced efficiency, and the development of secondary flows. Controlling the exit angle ensures that the flow leaves the blade with minimal swirl or separation, reducing total pressure losses and improving stage performance. It can be computed from the velocity components at the exit plane as:

$$\theta_{\text{exit}} = \text{atan}(V_a, U) \tag{6.4}$$

where V_a is the axial velocity component and U is the circumferential velocity component.

Both constraints are enforced as a double inequality with a small tolerance of $\pm 0.1\%$ of the initial value, rather than as strict equalities, to account for numerical errors and iterative convergence tolerances inherent in the computational method. This small margin ensures that the constraints are effectively satisfied while allowing the solver to converge reliably without being overly restrictive.

On top of these flow constraints, several geometrical ones were imposed. These constraints ensure that the generated blade geometry meets the required design and manufacturing standards, addressing the practical challenges that arise when constructing a blade airfoil. Further discussion of their purpose and implementation is provided in the corresponding chapter for each case study.

6.2 Baseline Mesh

The starting point for all aerodynamic ShpO cases was the best-fitted blade airfoil, hereafter referred to as the *baseline geometry*. In all cases, it will be represented with blue color as shown in Figure 6.1.

Accordingly, a new mesh was generated using an in-house meshing tool. The baseline mesh is hybrid consisting of 67,984 nodes, shown in Figure 6.2. In the near-wall region, structured layers with a step size of 10^{-3} mm (chord

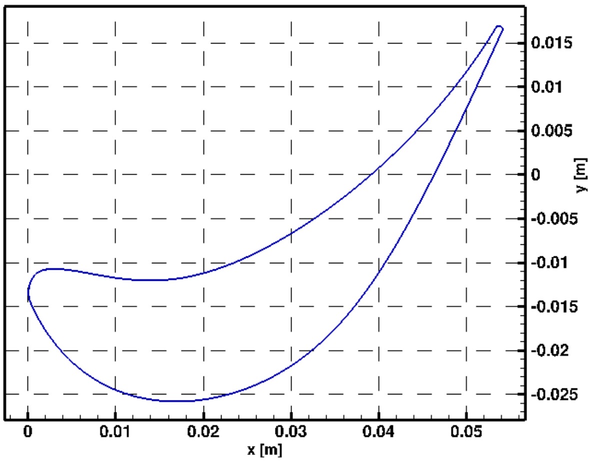


Figure 6.1: *Rotor airfoil baseline geometry.*

length 60 mm) were employed to adequately capture flow phenomena in this area, while the outer region of the mesh was unstructured and extended to the periodic boundary. The boundary conditions applied were identical to those described in Chapter 4, and the same inlet, outlet, and initialization flow variables were imposed as shown in Table 6.1

Category	**Variable**	**Symbol**	**Type / Units**	**Value**
Gas Properties	Gas constant	R	[J/kg·K]	287
	Specific heat ratio	γ	—	1.33
	Dynamic viscosity	μ	[Pa·s]	1.716×10^{-5}
Inlet	Total Pressure	p_t	Uniform [Pa]	1.586×10^{-5}
	Total Temperature	T_t	Uniform [K]	309.12
	Flow Angle	α	Scalar [°]	0
	Viscosity Ratio	μ_r	Scalar	20
Outlet	Pressure	p	Uniform [Pa]	1.303×10^{-5}

Table 6.1: *Gas properties and CFD boundary conditions.*

The resulting mesh, for these flow conditions, achieves a $y^+ < 1$, enabling the application of a low-Reynolds-number turbulence model without the use of wall functions.

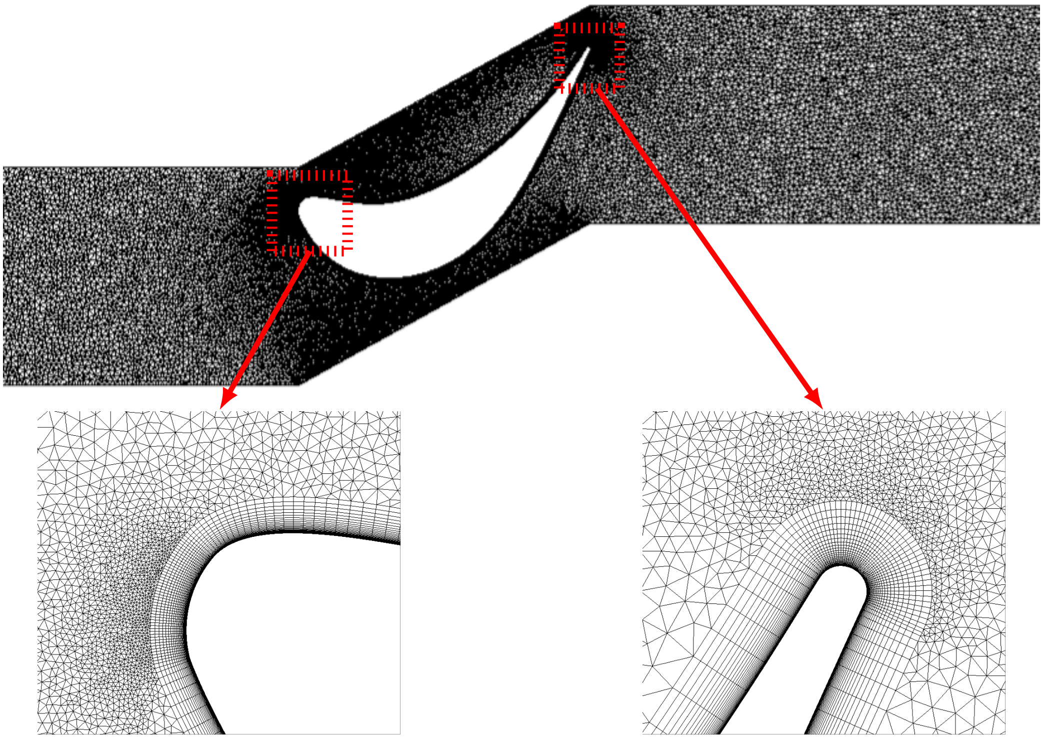


Figure 6.2: *The hybrid mesh of the stasonary rotor (top) with a zoomed-in view of the leading-edge region (bottom left) and the trailing-edge region (bottom right).*

A brief description of each case, along with its corresponding convergence figure, will be presented in the subsequent section, while the static pressure and Mach number fields will be also compiled and evaluated.

6.3 ShpO Results

In the first optimization, *Case A1*, the objective is to minimize the mass-averaged total pressure losses of the stationary rotor blade airfoil, subject to constraints on the exit flow angle and the inlet flow capacity. All design variables are permitted to vary freely within a margin of 10% of their initial values, and no additional geometrical constraints are imposed. The ShpO's convergence is shown in Figure 6.3a.

As the blade airfoil represents a stationary 2D rotor blade of a turbine stage, it is crucial to account for the high temperatures it experiences, particularly in the leading-edge region where the stagnation point forms. The primary solution to this issue is the implementation of an internal cooling system. However, reducing the leading-edge thickness to decrease total pressure losses simultaneously reduces the available area for cooling. For this reason, an explicit geometrical constraint was imposed in the *Case A2*, requiring the leading-edge region to remain identical to the initial configuration. This constraint was implemented through the custom Bézier-based parameterization by fixing the design variables that define the leading-edge region (x_{cLE}, y_{cLE}, r_{LE}, $\theta_{0,LE}$, and $\theta_{3,LE}$), while allowing the remaining variables to vary freely within the margin specified for Case A1.

Taking as a fact that the blade airfoil must be effectively cooled, a practical challenge in manufacturing lies in creating the internal air or fluid passages, particularly in regions where the available cross-sectional area becomes very small. One such critical region is the trailing edge, which was observed to thin out during the ShpO process aimed at reducing total pressure losses. If the trailing edge becomes excessively thin, the construction of internal cooling passages through casting becomes impractical. To address this, an ShpO with a frozen trailing edge was performed, designated as *Case A3*. In this case, only the corresponding variables x_{cTE}, y_{cTE}, r_{TE}, θ_{0TE}, and θ_{3TE} were kept constant, while the remaining variables were allowed to vary freely.

Allowing the trailing and/or leading edges to move results in a change in the airfoil chord, as the center points of the circular arcs representing these regions are free to move. Such changes can potentially impact several aspects of the rotor blade, including aerodynamic behavior (e.g., flow separation), manufacturability (e.g., blade interference), and overall performance (e.g., rotor solidity). For this reason, in this ShpO, *Case A4*, the leading and trailing edge regions were frozen by keeping x_{cLE}, y_{cLE}, r_{LE}, θ_{0LE}, θ_{3LE}, x_{cTE}, y_{cTE}, r_{TE}, θ_{0TE}, and θ_{3TE} constant. The remaining design variables were allowed to vary freely within the specified range.

Another critical parameter in the design of a blade airfoil, particularly a rotor blade, is its thickness. Blade thickness is directly related to the overall structural integrity, as it must withstand thermal, aerodynamic, and rotational loads. Therefore, maintaining the same maximum thickness during the design and ShpO of the airfoil is essential. Hence, in this ShpO, *Case A5*, the thickness of both the suction and pressure sides was kept constant, along with the leading and trailing edges. This was achieved by controlling the relevant design variables while allowing the camber line to vary freely. Consequently, only the variables $chord$, θ_{in}, θ_{out}, d_{in}, and d_{out} were allowed to change.

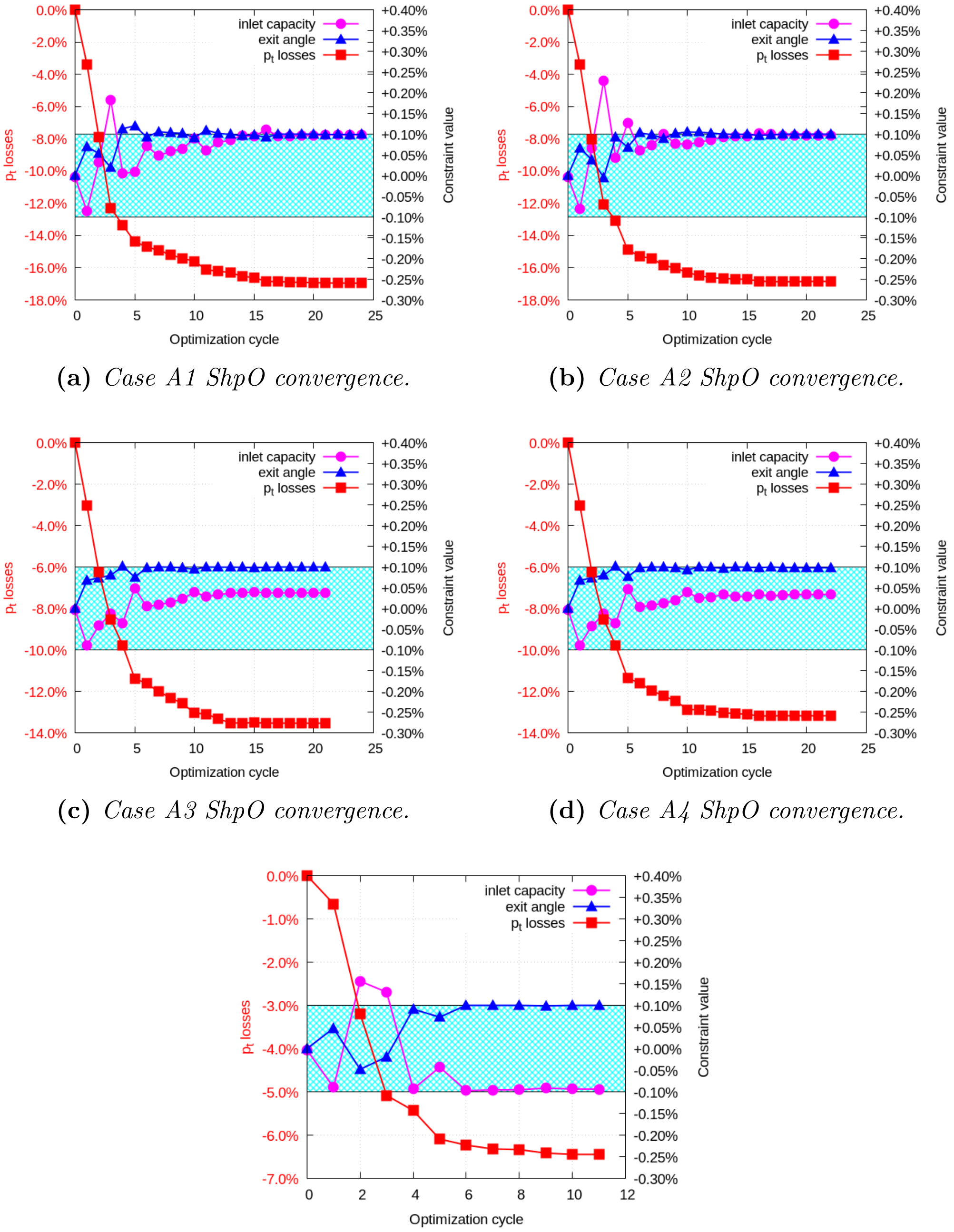
All of these geometrical constraints can be easily imposed thanks to the proposed custom Bézier-based parameterization and the way it generates the blade airfoil in distinct segments. By making minor modifications to the relevant code—specifically, by setting the appropriate design variables as constants—these constraints can be directly enforced. In contrast, with more conventional parameterization techniques, such as volumetric NURBS, direct imposition of geometric constraints is not feasible because they cannot be explicitly expressed. Instead, they must be applied indirectly as aerodynamic constraints, which not only requires more complex programming but also increases computational cost, as additional PDEs must be solved to satisfy the constraints.

A table summarizing all constraints and the key characteristics of each ShpO case is provided, Table 6.2:

Case	**Objective Function**	**Flow Constraints**	**Geometrical Constraints**
A1	Minimize Total Pressure losses	Inlet capacity and exit angle	-
A2	Same	Same	Frozen LE
A3			Frozen TE
A4			Frozen LE and TE
A5			Frozen LE, TE, SS, PS

Table 6.2: *Details of each adjoint ShpO case.*

To assess ShpO performance, Figure 6.3 depicts the convergence of the objective function for all five cases over the optimization cycle number. The evolution of the flow constraints, namely θ_{out} and *inlet capacity*, is also shown to evaluate constraint satisfaction.



(a) *Case A1 ShpO convergence.*

(b) *Case A2 ShpO convergence.*

(c) *Case A3 ShpO convergence.*

(d) *Case A4 ShpO convergence.*

(e) *Case A5 ShpO convergence.*

Figure 6.3: *Convergence of the objective function and evolution of constraints for the five ShpO cases.*

For the A1 case after 24 optimization cycles, the mass-averaged total pressure losses are reduced by 17%, reaching a local optimum with an inlet capacity and exit flow angle change of 0.1% (the absolute limit), as shown in Figure 6.3a.

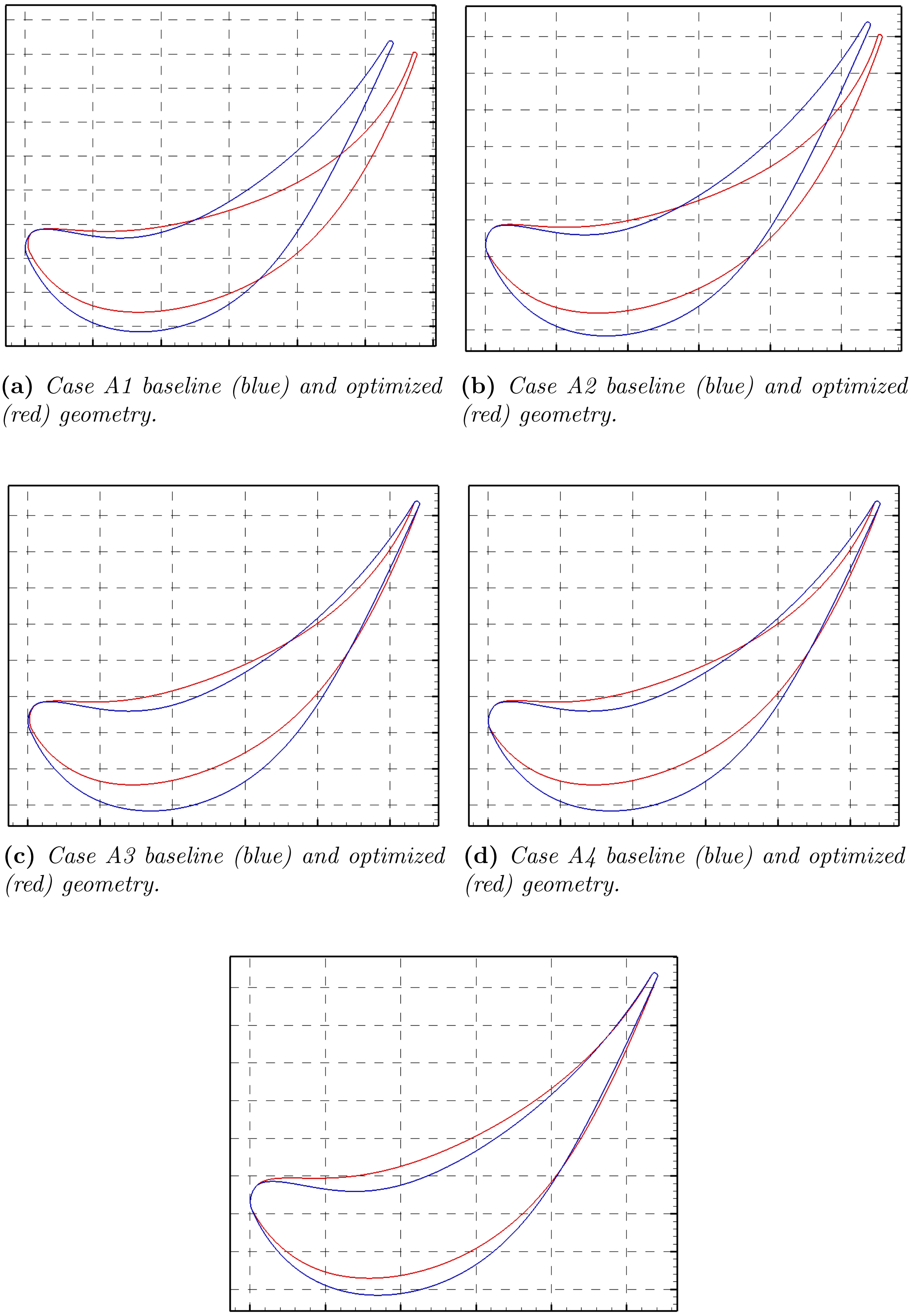
From Figure 6.3b it can be observed that after 23 optimization cycles, practically the same result as in Case A1 is achieved, with a reduction of 17% (difference smaller than 0.1%). This indicates that the contribution of the leading edge to the mass-averaged total pressure losses is negligible. The fact that two fewer cycles were required compared to Case A2 is most likely due to the reduced number of design variables, as five fewer variables needed to converge to their appropriate values for a local optimum to be reached.

As displayed in Figure 6.3c, after 21 optimization cycles, the total cost function was reduced by 13.6%, indicating that the trailing edge plays a significant role in generating losses (a 3.4% difference compared to Case A1). This outcome was expected, as the trailing edge is responsible for the wake it generates, which induces turbulence and energy dissipation in the downstream flow, thereby contributing significantly to total pressure losses

A reduction of 13.4% in the objective function after 22 optimization cycles was achieved in the case A4, as presented in Figure 6.3d . As expected, the leading edge contributes minimally to the generation of total pressure losses, which is evident from the difference in total pressure loss reduction between Cases A1 and A2. Consequently, the improvement observed in Case A4 relative to Case A3 is marginal, approximately 0.4%.

The result of the A5 case ShpO run is an important reduction in the total pressure losses, amounting to 6.5% after 11 optimization cycles, illustrated in Figure 6.3e . This improvement was achieved solely by modifying the camber line of the blade, which highlights its contribution to the cost function. The camber line influences the blade's pressure distribution and flow acceleration, thereby affecting boundary layer behavior and flow separation, which are key factors in total pressure losses. Also, noteworthy in A5 case is that the inlet capacity constraint shifted to the lower boundary, in contrast to all other cases. This behavior can be attributed to the fact that, in A5, the blade thickness remained constant, whereas in the other cases the blade became thinner, allowing a larger mass flow to pass through.

For clarity and comparison, Figure 6 presents the optimized airfoil shapes obtained for each case alongside the baseline configuration. This visualization highlights the geometric modifications introduced during the ShpO process. In addition, the Mach number and static pressure fields are presented for each ShpO case and the baseline configuration in Figure 6.5 and 6.6, respectively, to illustrate how geometric modifications affect local flow velocities, pressure distribution, and aerodynamic loading along the blade, thereby highlighting the mechanisms behind changes in total pressure losses.



(a) *Case A1 baseline (blue) and optimized (red) geometry.*

(b) *Case A2 baseline (blue) and optimized (red) geometry.*

(c) *Case A3 baseline (blue) and optimized (red) geometry.*

(d) *Case A4 baseline (blue) and optimized (red) geometry.*

(e) *Case A5 baseline (blue) and optimized (red) geometry.*

Figure 6.4: *Optimized airfoil geometries (red) for all cases compared with the baseline configuration (blue).*

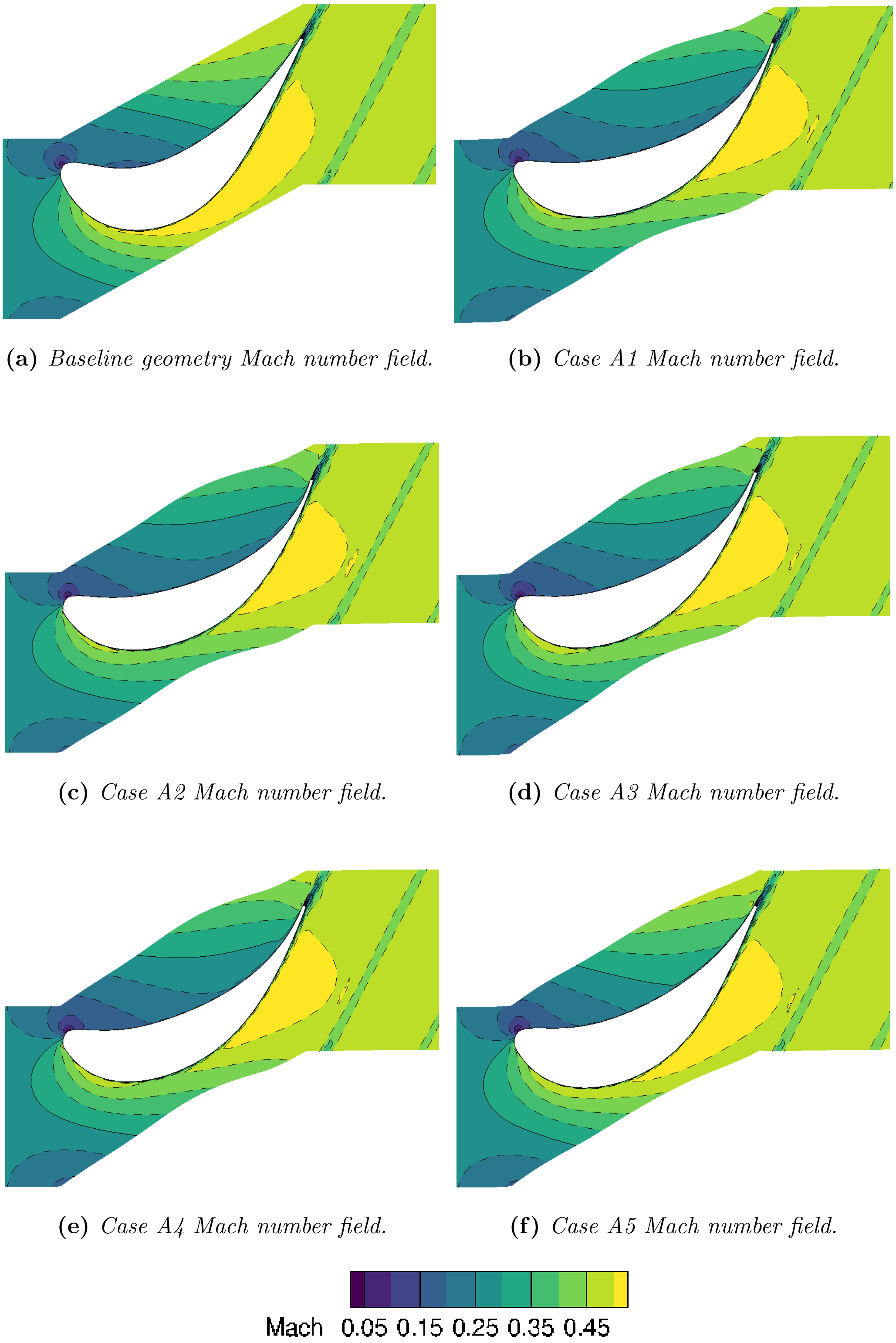
The Case A1 optimization run is indicative of the subsequent cases, as no geometrical constraints were imposed. This makes it possible to observe the natural trend of the blade airfoil towards an optimized solution, highlighting the regions with the greatest contribution to total pressure losses, intuitively corresponding to those that underwent the most significant modifications. The optimized airfoil exhibits minor changes in the first half of the chord, while the second half is heavily modified. In this latter region, the airfoil becomes thinner, particularly near the trailing edge, and presents a different stagger angle and camber, as illustrated in Figure 6a.

As expected, the optimized airfoil in Case A2 shows no substantial differences compared to Case A1, since the leading edge (which is frozen in this case) has only a minor contribution to the total pressure losses. What is noteworthy, however, is that because the leading edge cannot be modified, the thinning of the airfoil blade begins slightly earlier than in Case A1. In addition, the difference in stagger angle is smaller, as the airfoil cannot 'turn' as much in the first half of the chord as illustrated in Figure 6b.

The most pronounced differences relative to Case A1, though less substantial when compared to the baseline geometry, are observed in Case A3. Since the trailing edge—the primary contributor to total pressure losses—is frozen, the ShpO does not achieve as large a reduction as in the first ShpO case. This limitation is evident in Figure 6c, where the optimized airfoil shows only modest deviations from the baseline. The airfoil blade becomes thinner along most of its chord, with noticeable changes in camber, before thickening again near the trailing edge to satisfy the imposed constraint.

With the additional constraint of fixing both the leading and trailing edges, the blade airfoil in Case A4 closely resembles that of Case A3. As a result, the airfoil becomes slightly thinner toward the end compared to Case A3, before adjusting again to meet the constant trailing-edge constraint. Overall, it maintains a smoother camber along its entire length relative to the baseline geometry, as illustrated in Figure 6d.

Considering the geometrical constraints imposed in ShpO Case A5, the resulting airfoil maintains the same chord, leading and trailing edges, and thickness polynomial as the baseline, while achieving a reduction in mass-averaged total pressure losses. As illustrated in Figure 6e, the baseline and optimized airfoils appear very similar, yet the optimized configuration achieves a 6.5% improvement in the objective function. This improvement is primarily due to a more favorable redistribution of the camber, which is less aggressive near the leading edge and gradually converges to the baseline distribution toward the trailing edge.



(a) *Baseline geometry Mach number field.*

(b) *Case A1 Mach number field.*

(c) *Case A2 Mach number field.*

(d) *Case A3 Mach number field.*

(e) *Case A4 Mach number field.*

(f) *Case A5 Mach number field.*

(g) *Mach number field palette.*

Figure 6.5: *Mach number field around the optimized airfoils for the baseline geometry and the five ShpO cases.*

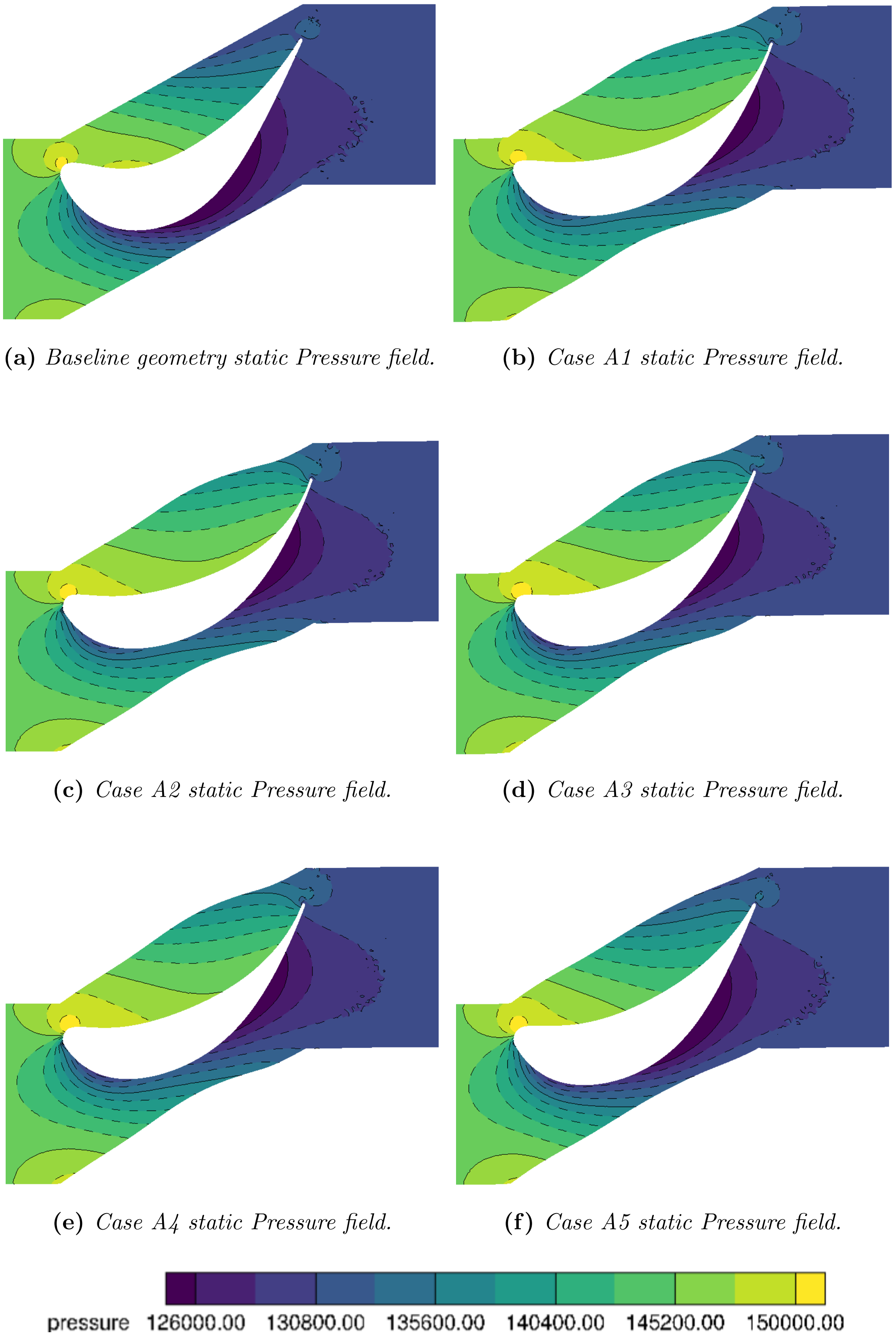
The Mach number contours for the baseline and optimized configurations (A1–A5) are bounded by $M \leq 0.55$. Consequently, the sharp gradients observed in the contour fields are regions of rapid subsonic deceleration associated with adverse pressure gradients (APGs). In such regimes, performance is governed by the management of surface diffusion, boundary-layer development, and wake mixing.

The baseline geometry, shown in Figure 6.5a, exhibits distinct flow features on the SS and PS. On the suction side, the flow experiences a strong acceleration followed by a pronounced adverse pressure gradient (APG), which is expected to promote boundary-layer thickening. On the pressure side, the flow accelerates near the trailing edge, resulting in steep velocity gradients and heightened sensitivity to downstream diffusion. The wake downstream of the trailing edge is relatively thick, as indicated by the extended low-Mach region and the elevated mixing losses, which are also verified by the ShpO results.

In Case A1, Figure 6.5b, the suction-side acceleration is smoother over the first half of the chord, resulting in a more concentrated high-Mach-number region, resembling a "bubble," toward the trailing edge.The diffusion process is distributed more evenly, resulting in a weaker adverse pressure gradient (APG) compared with the baseline. Consequently, the wake thickness is reduced, indicating lower total pressure loss. A similar behavior is observed in Case A2, Figure 6.5c, although the high-Mach-number region is slightly smaller, reflecting a reduction in acceleration and diffusion effectiveness relative to A1.

The suction-side Mach number distribution shifts slightly toward the leading edge, resulting in a more gradual and well-distributed high-Mach-number region in Case A3. Consequently, the adverse pressure gradient (APG) intensity is reduced, the boundary layer is less burdened, and the wake is thinner than in the baseline case, reflecting lower downstream mixing losses. A similar trend is observed in Figure 6.5e, Case A4, where the frozen leading edge produces slightly higher Mach numbers near the leading edge compared with Case A3.

Under all the additional geometrical constraints, Case A5 produces a blade airfoil whose suction-side Mach number field resembles the baseline pattern but exhibits a more favorable distribution. The smoother camber variations promote better acceleration and reduce the intensity of the (APG), as seen in Figure 6.5f.



(a) *Baseline geometry static Pressure field.*

(b) *Case A1 static Pressure field.*

(c) *Case A2 static Pressure field.*

(d) *Case A3 static Pressure field.*

(e) *Case A4 static Pressure field.*

(f) *Case A5 static Pressure field.*

(g) *Static Pressure field palette.*

Figure 6.6: *Static Pressure field around the optimized airfoils for the baseline geometry and the five ShpO cases.*

Total-pressure loss is typically correlated with three static-pressure features: the severity and streamwise extent of adverse pressure gradients, the abruptness of pressure recovery near the trailing edge, and the wake footprint, which can be judged by how far the low-pressure region extends downstream. Sharp suction peaks with rapid pressure recovery tend to thicken the boundary layer and may trigger separation, leading to irreversible energy losses. In contrast, a gentle recovery and a narrow, well-formed wake indicate lower dissipation and smaller total-pressure deficits. The static pressure contours presented in Figure 6.6 illustrate the distribution of pressure around the airfoil surfaces and within the near-field wake region. High-pressure zones, represented by yellow to green contours, are concentrated near the leading-edge stagnation region, while low-pressure zones dominate along the suction side. A general trend across the optimized cases is the smoothing of pressure gradients and the redistribution of the suction peak.

The baseline field shows a deep suction peak, visualized as a dark blue region in Figure 6.6a, just downstream of the leading edge and an abrupt recovery toward the trailing edge. Together with a relatively broad, uneven wake signature, these features point to significant viscous and mixing losses. The downstream wake region also displays notable pressure variations, suggesting the formation of drag-inducing vortices that contribute to total pressure losses.

For the Case A1, the suction peak develops more smoothly and is conveyed toward the back section of the airfoil, while the pressure recovery along the suction side occurs more gradually, as seen in Figure 6.6b. This distribution indicates improved flow attachment and a reduced likelihood of separation. Case A2, Figure 6.6c, exhibits very similar behavior, with a slightly stronger suction peak compared to Case A1.

Constrained by the constant trailing edge, in Case A3, Figure 6.6d, the suction peak shifts slightly toward the leading edge in an effort to distribute it more uniformly, while allowing a gradual pressure recovery. Case A4, illustrated in Figure 6.6e, exhibits higher suction than Case A3, yet maintains a gentle pressure recovery and a visibly narrower, coherent wake. This combination indicates reduced losses—higher than those of Case A3, but still below the baseline—representing a low-loss configuration that preserves a larger pressure differential.

Despite a shape very similar to the baseline, the airfoil in Case A5, Figure 6.6f, achieves a smoother and more gradual redistribution of static pressure compared to the baseline. This results in improved pressure recovery yielding lower total pressure losses.

Table 6 presents the values of the objective function and flow constraints for the baseline geometry and the optimized airfoil across all cases.

Case	Total Pressure Loss [Pa]			Inlet Capacity [$\times 10^{-7}$]			Exit Flow Angle [rad]		
	Baseline	Optimized	Δ[%]	Baseline	Optimized	Δ[%]	Baseline	Optimized	Δ[%]
A1	721.72	599.36	-17	5.280	5.285	-0.09	1.090	1.091	-0.09
A2	721.72	599.95	-17	5.280	5.285	-0.09	1.090	1.091	-0.09
A3	721.72	624.05	-13.6	5.280	5.282	-0.04	1.090	1.091	-0.09
A4	721.72	626.54	-13.4	5.280	5.282	-0.04	1.090	1.091	-0.09
A5	721.72	675.25	-6.5	5.280	5.275	-0.09	1.090	1.091	-0.09

Table 6.3: *Comparison of optimized results for all ShpO cases. Δ[%] indicates the percentage deviation relative to the baseline.*

Chapter 7

Conclusions

7.1 Overview

In this diploma thesis, the main objective was to propose a new, customized approach to airfoil parameterization, specialized for turbine blade airfoil geometries and to integrate it into the aerodynamic optimization process. The airfoil geometry was decomposed into five distinct segments: the mean camber line, suction and pressure sides, and the leading and trailing edges. Initially, the conceptual background of the custom Bézier-based airfoil parameterization method was presented, with emphasis on the mathematical formulation for generating each segment of the airfoil. Subsequently, an algorithm for best-fit approximation of airfoils was developed, enabling their representation through the proposed parameterization, and was tested on turbine blade airfoil geometries.

Furthermore, the validation of the CFD model for a single turbine stage, based on available experimental data, was conducted prior to performing aerodynamic shape optimization (ShpO) of the blade airfoil geometry representing the turbine rotor. The process of integrating the custom Bézier-based parameterization into the optimization cycle via the PUMA software was then described. Within this framework, multiple ShpO cases were conducted using the same objective function and flow constraints but with different geometrical constraints, highlighting the flexibility and unique capabilities of the proposed parameterization method in enforcing complex geometric requirements. The results were evaluated in terms of both the quality of the final solution and computational efficiency.

7.2 Conclusions

The main conclusions drawn from this diploma thesis are as follows:

- The proposed parameterization provides a smooth and flexible representation of the airfoil geometry while maintaining a clear link between the design variables and physical quantities such as camber, chord, thickness, and metal angles. Furthermore, explicit analytical gradients of the control points with respect to the design variables are available, enabling efficient application of gradient-based optimization methods.
- Its uniqueness arises from its distinctive conceptual approach, in which the airfoil is divided into five separate segments: the mean camber line, suction and pressure sides, and the leading and trailing edges.
- The proposed parameterization proved to be a powerful tool for describing airfoil geometry, offering high accuracy with a relatively small number of design variables, typically 25.
- Compared to other parameterization methods, it enables the direct enforcement of geometrical constraints without the need to solve additional equations, resulting in reduced complexity and computational cost. This capability is particularly important for turbine blades, where certain regions must remain unchanged during optimization due to cooling requirements, structural integrity, or other physical constraints.
- Following shape optimization, the resulting design vector can be easily understood, providing an intuitive understanding of the changes in physical quantities and informing future designs or optimization studies.

7.3 Proposals for Future Work

A natural follow up of the work carried out in this diploma thesis regarding the proposed parameterization method could include the following directions:

- Application of the custom Bézier-based parameterization to 3D problems: Extending the method to three-dimensional geometries could open new opportunities in the aerodynamic design of aerospace and automotive components, particularly in cases where complex geometrical constraints must be respected (e.g., maintaining the thickness distribution in a wing or airfoil section for structural integrity).

- Investigation of inverse design problems using the proposed parameterization: The application of this parameterization to inverse design problems, together with a comparative analysis against other parameterization methods, could establish it as a robust and efficient tool for inverse design.

- Combination of the proposed parameterization with other methods: Employing alternative parameterization techniques during the initial optimization cycles, followed by the proposed Bézier-based method for fine-tuning, could accelerate the optimization process while ensuring high-quality solutions.

Appendix A

Analytical derivative formulas

In the following appendices, the analytical derivative formulas for Bézier curves, as referenced in Chapter 2, as well as the explicit computation of the partial derivatives of each segment of the blade airfoil with respect to the design variables of the custom Bézier-based parameterization, as discussed in Chapter 3, are presented.

A.1 Derivatives of a Bézier curve

The computation of the unitary vector, described in equation (2.16) is based on the calculation of the first derivative of the mean camber line, expressed by Bézier curves. This is formed as follows:

Since the control points are independent of the variable u, computing the derivative curve $MCL'(u)$ reduces to the computation of the derivatives of $B_{n,i}(u)$'s. With some simple algebraic manipulations, we have the following result for $B'_{n,i}(u)$:

$$B'_{n,i}(u) = n\left[B_{n-1,i-1}(u) - B_{n-1,i}(u)\right], \quad i = 0, 1, \ldots, n \tag{A.1}$$

Then, computing the derivative of the curve $MCL(u)$ yields:

$$MCL'(u) = \sum_{i=0}^{n} P_i B'_{n,i}(u) \tag{A.2}$$

Let

$$Q_0 = n(P_1 - P_0), \quad Q_1 = n(P_2 - P_1), \quad \ldots, \quad Q_{n-1} = n(P_n - P_{n-1}) \tag{A.3}$$

Then the above equation reduces to the following:

$$MCL'(u) = \sum_{i=0}^{n-1} Q_i B_{n-1,i}(u) \tag{A.4}$$

A.2 MCL partial derivatives w.r.t design variables

The mean camber line partial derivatives w.r.t design variables

$$\frac{\partial x_B(t)}{\partial c} = B_1^3(t)\, d_{\text{in}} \cos\theta_{\text{in}} - B_2^3(t)\, d_{\text{out}} \cos\theta_{\text{out}} \tag{A.5}$$

$$\frac{\partial x_B(t)}{\partial \theta_{\text{in}}} = -B_1^3(t)\, d_{\text{in}}\, c\, \sin\theta_{\text{in}} \tag{A.6}$$

$$\frac{\partial x_B(t)}{\partial \theta_{\text{out}}} = B_2^3(t)\, d_{\text{out}}\, c\, \sin\theta_{\text{out}} \tag{A.7}$$

$$\frac{\partial x_B(t)}{\partial d_{\text{in}}} = B_1^3(t)\, c\, \cos\theta_{\text{in}} \tag{A.8}$$

$$\frac{\partial x_B(t)}{\partial d_{\text{out}}} = -B_2^3(t)\, c\, \cos\theta_{\text{out}} \tag{A.9}$$

$$\frac{\partial y_B(t)}{\partial c} = B_1^3(t)\, d_{\text{in}} \sin\theta_{\text{in}} - B_2^3(t)\, d_{\text{out}} \sin\theta_{\text{out}} \tag{A.10}$$

$$\frac{\partial y_B(t)}{\partial \theta_{\text{in}}} = B_1^3(t)\, d_{\text{in}}\, c\, \cos\theta_{\text{in}} \tag{A.11}$$

$$\frac{\partial y_B(t)}{\partial \theta_{\text{out}}} = -B_2^3(t)\, d_{\text{out}}\, c\, \cos\theta_{\text{out}} \tag{A.12}$$

$$\frac{\partial y_B(t)}{\partial d_{\text{in}}} = B_1^3(t)\, c\, \sin\theta_{\text{in}} \tag{A.13}$$

$$\frac{\partial y_B(t)}{\partial d_{\text{out}}} = -B_2^3(t)\, c\, \sin\theta_{\text{out}} \tag{A.14}$$

$$\frac{\partial \dot{x}_B(t)}{\partial c} = 3\Big(\dot{B}_0^2(t)\, d_{\text{in}} \cos\theta_{\text{in}} + \dot{B}_1^2(t)\,(-d_{\text{out}} \cos\theta_{\text{out}} - d_{\text{in}} \cos\theta_{\text{in}}) + \dot{B}_2^2(t)\, d_{\text{out}} \cos\theta_{\text{out}}\Big) \tag{A.15}$$

$$\frac{\partial \dot{x}_B(t)}{\partial \theta_{\text{in}}} = 3\, d_{\text{in}}\, c\, \sin\theta_{\text{in}}\, (\dot{B}_1^2(t) - \dot{B}_0^2(t)) \tag{A.16}$$

$$\frac{\partial \dot{x}_B(t)}{\partial \theta_{\text{out}}} = 3\, d_{\text{out}}\, c\, \sin\theta_{\text{out}}\, (\dot{B}_1^2(t) - \dot{B}_2^2(t)) \tag{A.17}$$

$$\frac{\partial \dot{x}_B(t)}{\partial d_{\text{in}}} = 3\, c\, \cos\theta_{\text{in}}\, (\dot{B}_0^2(t) - \dot{B}_1^2(t)) \tag{A.18}$$

$$\frac{\partial \dot{x}_B(t)}{\partial d_{\text{out}}} = 3\, c\, \cos\theta_{\text{out}}\, (\dot{B}_2^2(t) - \dot{B}_1^2(t)) \tag{A.19}$$

$$\frac{\partial \dot{y}_B(t)}{\partial c} = 3\Big(\dot{B}_0^2(t)\, d_{\text{in}} \sin\theta_{\text{in}} + \dot{B}_1^2(t)\, (-d_{\text{out}} \sin\theta_{\text{out}} - d_{\text{in}} \sin\theta_{\text{in}}) + \dot{B}_2^2(t)\, d_{\text{out}} \sin\theta_{\text{out}}\Big) \tag{A.20}$$

$$\frac{\partial \dot{y}_B(t)}{\partial \theta_{\text{in}}} = 3\, d_{\text{in}}\, c\, \cos\theta_{\text{in}}\, (\dot{B}_0^2(t) - \dot{B}_1^2(t)) \tag{A.21}$$

$$\frac{\partial \dot{y}_B(t)}{\partial \theta_{\text{out}}} = 3\, d_{\text{out}}\, c\, \cos\theta_{\text{out}}\, (\dot{B}_2^2(t) - \dot{B}_1^2(t)) \tag{A.22}$$

$$\frac{\partial \dot{y}_B(t)}{\partial d_{\text{in}}} = 3\, c\, \sin\theta_{\text{in}}\, (\dot{B}_0^2(t) - \dot{B}_1^2(t)) \tag{A.23}$$

$$\frac{\partial \dot{y}_B(t)}{\partial d_{\text{out}}} = 3\, c\, \sin\theta_{\text{out}}\, (\dot{B}_2^2(t) - \dot{B}_1^2(t)) \tag{A.24}$$

$$\text{denom}(t) = \dot{x}_B(t)^2 + \dot{y}_B(t)^2 \tag{A.25}$$

$$\frac{\partial n_x^{\mathrm{CL}}(t)}{\partial c} = -\frac{\partial \dot{y}_B(t)}{\partial c}\frac{1}{\sqrt{\mathrm{denom}(t)}} + \frac{y_{\mathrm{dev}}(t)\left(x_{\mathrm{dev}}(t)\frac{\partial \dot{x}_B(t)}{\partial c} + y_{\mathrm{dev}}(t)\frac{\partial \dot{y}_B(t)}{\partial c}\right)}{\left(\mathrm{denom}(t)\right)^{3/2}} \tag{A.26}$$

$$\frac{\partial n_x^{\mathrm{CL}}(t)}{\partial \theta_{\mathrm{in}}} = -\frac{\partial \dot{y}_B(t)}{\partial \theta_{\mathrm{in}}}\frac{1}{\sqrt{\mathrm{denom}(t)}} + \frac{y_{\mathrm{dev}}(t)\left(x_{\mathrm{dev}}(t)\frac{\partial \dot{x}_B(t)}{\partial \theta_{\mathrm{in}}} + y_{\mathrm{dev}}(t)\frac{\partial \dot{y}_B(t)}{\partial \theta_{\mathrm{in}}}\right)}{\left(\mathrm{denom}(t)\right)^{3/2}} \tag{A.27}$$

$$\frac{\partial n_x^{\mathrm{CL}}(t)}{\partial \theta_{\mathrm{out}}} = -\frac{\partial \dot{y}_B(t)}{\partial \theta_{\mathrm{out}}}\frac{1}{\sqrt{\mathrm{denom}(t)}} + \frac{y_{\mathrm{dev}}(t)\left(x_{\mathrm{dev}}(t)\frac{\partial \dot{x}_B(t)}{\partial \theta_{\mathrm{out}}} + y_{\mathrm{dev}}(t)\frac{\partial \dot{y}_B(t)}{\partial \theta_{\mathrm{out}}}\right)}{\left(\mathrm{denom}(t)\right)^{3/2}} \tag{A.28}$$

$$\frac{\partial n_x^{\mathrm{CL}}(t)}{\partial d_{\mathrm{in}}} = -\frac{\partial \dot{y}_B(t)}{\partial d_{\mathrm{in}}}\frac{1}{\sqrt{\mathrm{denom}(t)}} + \frac{y_{\mathrm{dev}}(t)\left(x_{\mathrm{dev}}(t)\frac{\partial \dot{x}_B(t)}{\partial d_{\mathrm{in}}} + y_{\mathrm{dev}}(t)\frac{\partial \dot{y}_B(t)}{\partial d_{\mathrm{in}}}\right)}{\left(\mathrm{denom}(t)\right)^{3/2}} \tag{A.29}$$

$$\frac{\partial n_x^{\mathrm{CL}}(t)}{\partial d_{\mathrm{out}}} = -\frac{\partial \dot{y}_B(t)}{\partial d_{\mathrm{out}}}\frac{1}{\sqrt{\mathrm{denom}(t)}} + \frac{y_{\mathrm{dev}}(t)\left(x_{\mathrm{dev}}(t)\frac{\partial \dot{x}_B(t)}{\partial d_{\mathrm{out}}} + y_{\mathrm{dev}}(t)\frac{\partial \dot{y}_B(t)}{\partial d_{\mathrm{out}}}\right)}{\left(\mathrm{denom}(t)\right)^{3/2}} \tag{A.30}$$

$$\frac{\partial n_y^{\mathrm{CL}}(t)}{\partial c} = \frac{\partial \dot{x}_B(t)}{\partial c}\frac{1}{\sqrt{\mathrm{denom}(t)}} - \frac{x_{\mathrm{dev}}(t)\left(x_{\mathrm{dev}}(t)\frac{\partial \dot{x}_B(t)}{\partial c} + y_{\mathrm{dev}}(t)\frac{\partial \dot{y}_B(t)}{\partial c}\right)}{\left(\mathrm{denom}(t)\right)^{3/2}} \tag{A.31}$$

$$\frac{\partial n_y^{\mathrm{CL}}(t)}{\partial \theta_{\mathrm{in}}} = \frac{\partial \dot{x}_B(t)}{\partial \theta_{\mathrm{in}}}\frac{1}{\sqrt{\mathrm{denom}(t)}} - \frac{x_{\mathrm{dev}}(t)\left(x_{\mathrm{dev}}(t)\frac{\partial \dot{x}_B(t)}{\partial \theta_{\mathrm{in}}} + y_{\mathrm{dev}}(t)\frac{\partial \dot{y}_B(t)}{\partial \theta_{\mathrm{in}}}\right)}{\left(\mathrm{denom}(t)\right)^{3/2}} \tag{A.32}$$

$$\frac{\partial n_y^{\mathrm{CL}}(t)}{\partial \theta_{\mathrm{out}}} = \frac{\partial \dot{x}_B(t)}{\partial \theta_{\mathrm{out}}}\frac{1}{\sqrt{\mathrm{denom}(t)}} - \frac{x_{\mathrm{dev}}(t)\left(x_{\mathrm{dev}}(t)\frac{\partial \dot{x}_B(t)}{\partial \theta_{\mathrm{out}}} + y_{\mathrm{dev}}(t)\frac{\partial \dot{y}_B(t)}{\partial \theta_{\mathrm{out}}}\right)}{\left(\mathrm{denom}(t)\right)^{3/2}} \tag{A.33}$$

$$\frac{\partial n_y^{\mathrm{CL}}(t)}{\partial d_{\mathrm{in}}} = \frac{\partial \dot{x}_B(t)}{\partial d_{\mathrm{in}}}\frac{1}{\sqrt{\mathrm{denom}(t)}} - \frac{x_{\mathrm{dev}}(t)\left(x_{\mathrm{dev}}(t)\frac{\partial \dot{x}_B(t)}{\partial d_{\mathrm{in}}} + y_{\mathrm{dev}}(t)\frac{\partial \dot{y}_B(t)}{\partial d_{\mathrm{in}}}\right)}{\left(\mathrm{denom}(t)\right)^{3/2}} \tag{A.34}$$

$$\frac{\partial n_y^{\mathrm{CL}}(t)}{\partial d_{\mathrm{out}}} = \frac{\partial \dot{x}_B(t)}{\partial d_{\mathrm{out}}}\frac{1}{\sqrt{\mathrm{denom}(t)}} - \frac{x_{\mathrm{dev}}(t)\left(x_{\mathrm{dev}}(t)\frac{\partial \dot{x}_B(t)}{\partial d_{\mathrm{out}}} + y_{\mathrm{dev}}(t)\frac{\partial \dot{y}_B(t)}{\partial d_{\mathrm{out}}}\right)}{\left(\mathrm{denom}(t)\right)^{3/2}} \tag{A.35}$$

A.3 SS partial derivatives w.r.t design variables

The suction side partial derivatives w.r.t design variables.

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial c} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(\frac{\partial x_b}{\partial c} - \frac{\partial n_x^{\mathrm{CL}}}{\partial c}\frac{t_{\mathrm{suction}}}{2}\Big) \tag{A.36}$$

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial \theta_{\mathrm{in}}} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(\frac{\partial x_b}{\partial \theta_{\mathrm{in}}} - \frac{\partial n_x^{\mathrm{CL}}}{\partial \theta_{\mathrm{in}}}\frac{t_{\mathrm{suction}}}{2}\Big) \tag{A.37}$$

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial \theta_{\mathrm{out}}} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(\frac{\partial x_b}{\partial \theta_{\mathrm{out}}} - \frac{\partial n_x^{\mathrm{CL}}}{\partial \theta_{\mathrm{out}}}\frac{t_{\mathrm{suction}}}{2}\Big) \tag{A.38}$$

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial d_{\mathrm{in}}} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(\frac{\partial x_b}{\partial d_{\mathrm{in}}} - \frac{\partial n_x^{\mathrm{CL}}}{\partial d_{\mathrm{in}}}\frac{t_{\mathrm{suction}}}{2}\Big) \tag{A.39}$$

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial d_{\mathrm{out}}} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(\frac{\partial x_b}{\partial d_{\mathrm{out}}} - \frac{\partial n_x^{\mathrm{CL}}}{\partial d_{\mathrm{out}}}\frac{t_{\mathrm{suction}}}{2}\Big) \tag{A.40}$$

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial a_{0\mathrm{s}}} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(-\frac{1}{2}n_x\Big) \tag{A.41}$$

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial a_{1\mathrm{s}}} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(-\frac{1}{2}n_x k_i\Big) \tag{A.42}$$

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial a_{2\mathrm{s}}} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(-\frac{1}{2}n_x k_i^2\Big) \tag{A.43}$$

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial a_{3\mathrm{s}}} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(-\frac{1}{2}n_x k_i^3\Big) \tag{A.44}$$

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial a_{4\mathrm{s}}} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(-\frac{1}{2}n_x k_i^4\Big) \tag{A.45}$$

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial x_{c\mathrm{LE}}} = B^{n_{\mathrm{SS}}-1}_{n_{\mathrm{SS}}-3}(t) + B^{n_{\mathrm{SS}}-1}_{n_{\mathrm{SS}}-2}(t) + B^{n_{\mathrm{SS}}-1}_{n_{\mathrm{SS}}-1}(t) \tag{A.46}$$

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial r_{\mathrm{LE}}} = B^{n_{\mathrm{SS}}-1}_{n_{\mathrm{SS}}-1}(t)\cos\theta_0 + B^{n_{\mathrm{SS}}-1}_{n_{\mathrm{SS}}-2}(t)\Big(\cos\theta_0 + \frac{4}{3N_3}\tan\frac{\theta_{\mathrm{LE}}}{4}\sin\theta_0\Big) \tag{A.47}$$

$$\begin{aligned}\frac{\partial x_{\mathrm{SS}}(t)}{\partial \theta_{0\mathrm{LE}}} &= B^{n_{\mathrm{SS}}-1}_{n_{\mathrm{SS}}-1}(t)(-r_{\mathrm{LE}}\sin\theta_{0\mathrm{LE}}) \\ &\quad + B^{n_{\mathrm{SS}}-1}_{n_{\mathrm{SS}}-2}(t)\Big(-r_{\mathrm{LE}}\sin\theta_{0\mathrm{LE}} - \frac{1}{3N_3}r_{\mathrm{LE}}\sec^2\Big(\tfrac{\theta_{\mathrm{LE}}}{4}\Big)\sin\theta_{0\mathrm{LE}} \\ &\quad + \frac{4}{3N_3}r_{\mathrm{LE}}\tan\frac{\theta_{\mathrm{LE}}}{4}\cos\theta_{0\mathrm{LE}}\Big)\end{aligned} \tag{A.48}$$

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial \theta_{3\mathrm{LE}}} = B^{n_{\mathrm{SS}}-1}_{n_{\mathrm{SS}}-2}(t)\frac{1}{3N_3}r_{\mathrm{LE}}\sec^2\Big(\tfrac{\theta_{\mathrm{LE}}}{4}\Big)\sin\theta_{0\mathrm{LE}} \tag{A.49}$$

$$\frac{\partial x_{\mathrm{SS}}(t)}{\partial x_{c\mathrm{TE}}} = B^{n_{\mathrm{SS}}-1}_{0}(t) + B^{n_{\mathrm{SS}}-1}_{1}(t) + B^{n_{\mathrm{SS}}-1}_{2}(t) \tag{A.50}$$

$$\begin{aligned}\frac{\partial x_{\mathrm{SS}}(t)}{\partial r_{\mathrm{TE}}} &= B^{n_{\mathrm{SS}}-1}_{2}(t)\Big(\frac{\cos\theta_{0\mathrm{TE}} - \cos\theta_{3\mathrm{TE}} - \tan\frac{\theta_{\mathrm{TE}}}{4}(\frac{8}{3}\sin\theta_{3\mathrm{TE}} + \frac{4}{3}\sin\theta_{0\mathrm{TE}})}{N_2} \\ &\quad + \cos\theta_{3\mathrm{TE}} - \frac{8}{3N}\tan\frac{\theta_{\mathrm{TE}}}{4}\sin\theta_{3\mathrm{TE}}\Big) \\ &\quad + B^{n_{\mathrm{SS}}-1}_{1}(t)\Big(\cos\theta_{3\mathrm{TE}} - \frac{4}{3N}\tan\frac{\theta_{\mathrm{TE}}}{4}\sin\theta_{3\mathrm{TE}}\Big) + B^{n_{\mathrm{SS}}-1}_{0}(t)\cos\theta_{3\mathrm{TE}}\end{aligned} \tag{A.51}$$

$$\begin{aligned}\frac{\partial x_{\mathrm{SS}}(t)}{\partial \theta_{0\mathrm{TE}}} &= B^{n_{\mathrm{SS}}-1}_{1}(t)\left[\frac{1}{3N}r_{\mathrm{TE}}\sec^2\Big(\frac{\theta_{\mathrm{TE}}}{4}\Big)\sin\theta_{3\mathrm{TE}}\right] \\ &\quad + B^{n_{\mathrm{SS}}-1}_{2}(t)\left[\frac{-r_{\mathrm{TE}}\sin\theta_{0\mathrm{TE}} + \frac{1}{4}r_{\mathrm{TE}}\sec^2\Big(\frac{\theta_{\mathrm{TE}}}{4}\Big)\Big(\frac{8}{3}\sin\theta_{3\mathrm{TE}} + \frac{4}{3}\sin\theta_{0\mathrm{TE}}\Big)}{N_2}\right. \\ &\quad \left. - \frac{\frac{4}{3}r_{\mathrm{TE}}\tan\Big(\frac{\theta_{\mathrm{TE}}}{4}\Big)\cos\theta_{0\mathrm{TE}}}{N_2} + \frac{2}{3N}r_{\mathrm{TE}}\sec^2\Big(\frac{\theta_{\mathrm{TE}}}{4}\Big)\sin\theta_{3\mathrm{TE}}\right] \\ &\quad + B^{n_{\mathrm{SS}}-1}_{0}(t)\Big[-r_{\mathrm{TE}}\sin\theta_{0\mathrm{TE}}\Big]\end{aligned} \tag{A.52}$$

$$\begin{aligned}\frac{\partial x_{\text{SS}}(t)}{\partial \theta_{3\text{TE}}} = B_2^{n_{\text{SS}}-1}(t)\Bigg[&\frac{r_{\text{TE}}\sin\theta_3 - \frac{1}{4}r_{\text{TE}}\text{sec\_term\_sq\_TE}(\frac{8}{3}\sin\theta_3 + \frac{4}{3}\sin\theta_0) - \frac{8}{3}r_{\text{TE}}\tan\frac{\theta_{\text{TE}}}{4}\cos\theta_3}{N_2} \\ &- \frac{2}{3N}r_{\text{TE}}\text{sec\_term\_sq\_TE}\sin\theta_3 - r_{\text{TE}}\sin\theta_3\Bigg] \\ &+ B_1^{n_{\text{SS}}-1}(t)\Big[-\frac{1}{3N}r_{\text{TE}}\text{sec\_term\_sq\_TE}\sin\theta_3 - \frac{4}{3N}r_{\text{TE}}\tan\frac{\theta_{\text{TE}}}{4}\cos\theta_3 - r_{\text{TE}}\sin\theta_3\Big] \\ &+ B_0^{n_{\text{SS}}-1}(t)(-r_{\text{TE}}\sin\theta_3)\end{aligned} \tag{A.53}$$

$$\begin{aligned}\frac{\partial y_{\text{SS}}(t)}{\partial \theta_0^{\text{TE}}} = B_2^{n_{\text{SS}}-1}(t)\Bigg[&\frac{r_{\text{TE}}\cos\theta_0 - \frac{1}{4}r_{\text{TE}}\sec^2\Big(\frac{\theta_{\text{TE}}}{4}\Big)\Big(\frac{8}{3}\cos\theta_3 + \frac{4}{3}\cos\theta_0\Big)}{N_2} \\ &- \frac{\frac{4}{3}r_{\text{TE}}\tan\Big(\frac{\theta_{\text{TE}}}{4}\Big)\sin\theta_0}{N_2} - \frac{2}{3N}\,r_{\text{TE}}\sec^2\Big(\frac{\theta_{\text{TE}}}{4}\Big)\cos\theta_3\Bigg] \\ &+ B_1^{n_{\text{SS}}-1}(t)\Big[-\tfrac{1}{3N}r_{\text{TE}}\sec^2\Big(\frac{\theta_{\text{TE}}}{4}\Big)\cos\theta_3\Big]\end{aligned} \tag{A.54}$$

$$\begin{aligned}\frac{\partial y_{\text{SS}}(t)}{\partial \theta_3^{\text{TE}}} = B_2^{n_{\text{SS}}-1}(t)\Bigg[&\frac{-r_{\text{TE}}\cos\theta_3 + \frac{1}{4}r_{\text{TE}}\sec^2\Big(\frac{\theta_{\text{TE}}}{4}\Big)\Big(\frac{8}{3}\cos\theta_3 + \frac{4}{3}\cos\theta_0\Big) - \frac{8}{3}r_{\text{TE}}\tan\Big(\frac{\theta_{\text{TE}}}{4}\Big)\sin\theta_3}{N_2} \\ &+ \tfrac{2}{3N}r_{\text{TE}}\sec^2\Big(\frac{\theta_{\text{TE}}}{4}\Big)\cos\theta_3 - \tfrac{8}{3N}r_{\text{TE}}\tan\Big(\frac{\theta_{\text{TE}}}{4}\Big)\sin\theta_3 + r_{\text{TE}}\cos\theta_3\Bigg] \\ &+ B_1^{n_{\text{SS}}-1}(t)\Big[\tfrac{1}{3N}r_{\text{TE}}\sec^2\Big(\frac{\theta_{\text{TE}}}{4}\Big)\cos\theta_3 - \tfrac{4}{3N}r_{\text{TE}}\tan\Big(\frac{\theta_{\text{TE}}}{4}\Big)\sin\theta_3 + r_{\text{TE}}\cos\theta_3\Big] \\ &+ B_0^{n_{\text{SS}}-1}(t)\,r_{\text{TE}}\cos\theta_3\end{aligned} \tag{A.55}$$

$$\frac{\partial y_{\mathrm{SS}}(t)}{\partial y_c} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(\frac{\partial y_b}{\partial y_c} - \frac{\partial n_y^{\mathrm{CL}}}{\partial y_c}\frac{t_{\mathrm{suction}}}{2}\Big) \tag{A.56}$$

$$\frac{\partial y_{\mathrm{SS}}(t)}{\partial \theta_{\mathrm{In}}} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(\frac{\partial y_b}{\partial \theta_{\mathrm{In}}} - \frac{\partial n_y^{\mathrm{CL}}}{\partial \theta_{\mathrm{In}}}\frac{t_{\mathrm{suction}}}{2}\Big) \tag{A.57}$$

$$\frac{\partial y_{\mathrm{SS}}(t)}{\partial \theta_{\mathrm{Out}}} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(\frac{\partial y_b}{\partial \theta_{\mathrm{Out}}} - \frac{\partial n_y^{\mathrm{CL}}}{\partial \theta_{\mathrm{Out}}}\frac{t_{\mathrm{suction}}}{2}\Big) \tag{A.58}$$

$$\frac{\partial y_{\mathrm{SS}}(t)}{\partial d_{\mathrm{In}}} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(\frac{\partial y_b}{\partial d_{\mathrm{In}}} - \frac{\partial n_y^{\mathrm{CL}}}{\partial d_{\mathrm{In}}}\frac{t_{\mathrm{suction}}}{2}\Big) \tag{A.59}$$

$$\frac{\partial y_{\mathrm{SS}}(t)}{\partial d_{\mathrm{Out}}} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(\frac{\partial y_b}{\partial d_{\mathrm{Out}}} - \frac{\partial n_y^{\mathrm{CL}}}{\partial d_{\mathrm{Out}}}\frac{t_{\mathrm{suction}}}{2}\Big) \tag{A.60}$$

$$\frac{\partial y_{\mathrm{SS}}(t)}{\partial a_0^s} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(-\frac{1}{2}n_y\Big) \tag{A.61}$$

$$\frac{\partial y_{\mathrm{SS}}(t)}{\partial a_1^s} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(-\frac{1}{2}n_y\,\xi\Big) \tag{A.62}$$

$$\frac{\partial y_{\mathrm{SS}}(t)}{\partial a_2^s} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(-\frac{1}{2}n_y\,\xi^2\Big) \tag{A.63}$$

$$\frac{\partial y_{\mathrm{SS}}(t)}{\partial a_3^s} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(-\frac{1}{2}n_y\,\xi^3\Big) \tag{A.64}$$

$$\frac{\partial y_{\mathrm{SS}}(t)}{\partial a_4^s} = \sum_{j=0}^{n_{\mathrm{SS}}-1} B_{n_{\mathrm{SS}}-1-j}^{n_{\mathrm{SS}}-1}(t)\Big(-\frac{1}{2}n_y\,\xi^4\Big) \tag{A.65}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{SS}}(t)}{\partial \theta_0^{\mathrm{LE}}} &= B^{n_{\mathrm{SS}}-1}_{n_{\mathrm{SS}}-1}(t) r_{\mathrm{LE}} \cos\theta_0 \\ &+ B^{n_{\mathrm{SS}}-1}_{n_{\mathrm{SS}}-2}(t)\left[r_{\mathrm{LE}}\cos\theta_0 + \frac{1}{3N_3} r_{\mathrm{LE}} \mathrm{sec\_term\_sq\_LE} \cos\theta_0 + \frac{4}{3N_3} r_{\mathrm{LE}} \tan\frac{\theta_{\mathrm{LE}}}{4}\sin\theta_0\right]\end{aligned} \tag{A.66}$$

$$\frac{\partial y_{\mathrm{SS}}(t)}{\partial \theta_3^{\mathrm{LE}}} = B^{n_{\mathrm{SS}}-1}_{n_{\mathrm{SS}}-2}(t)\Big(-\frac{1}{3N_3} r_{\mathrm{LE}} \mathrm{sec\_term\_sq\_LE} \cos\theta_0\Big) \tag{A.67}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{SS}}(t)}{\partial r_{\mathrm{TE}}} &= B^{n_{\mathrm{SS}}-1}_{2}(t)\left[\frac{\sin\theta_0 - \sin\theta_3 + \tan\frac{\theta_{\mathrm{TE}}}{4}\big(\frac{8}{3}\cos\theta_3 + \frac{4}{3}\cos\theta_0\big)}{N_2} + \sin\theta_3 + \frac{8}{3N}\tan\frac{\theta_{\mathrm{TE}}}{4}\cos\theta_3\right] \\ &+ B^{n_{\mathrm{SS}}-1}_{1}(t)\Big(\sin\theta_3 + \frac{4}{3N}\tan\frac{\theta_{\mathrm{TE}}}{4}\cos\theta_3\Big) + B^{n_{\mathrm{SS}}-1}_{0}(t)\sin\theta_3\end{aligned} \tag{A.68}$$

A.4 PS partial derivatives w.r.t design variables

The pressure side partial derivatives w.r.t design variables.

$$\frac{\partial x_{\mathrm{PS}}(t)}{\partial x_c} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{\partial x_b}{\partial x_c} + \frac{\partial n_x^{\mathrm{CL}}}{\partial x_c}\frac{t_{\mathrm{pressure}}}{2}\Big) \tag{A.69}$$

$$\frac{\partial x_{\mathrm{PS}}(t)}{\partial \theta_{\mathrm{In}}} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{\partial x_b}{\partial \theta_{\mathrm{In}}} + \frac{\partial n_x^{\mathrm{CL}}}{\partial \theta_{\mathrm{In}}}\frac{t_{\mathrm{pressure}}}{2}\Big) \tag{A.70}$$

$$\frac{\partial x_{\mathrm{PS}}(t)}{\partial \theta_{\mathrm{Out}}} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{\partial x_b}{\partial \theta_{\mathrm{Out}}} + \frac{\partial n_x^{\mathrm{CL}}}{\partial \theta_{\mathrm{Out}}}\frac{t_{\mathrm{pressure}}}{2}\Big) \tag{A.71}$$

$$\frac{\partial x_{\mathrm{PS}}(t)}{\partial d_{\mathrm{In}}} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{\partial x_b}{\partial d_{\mathrm{In}}} + \frac{\partial n_x^{\mathrm{CL}}}{\partial d_{\mathrm{In}}}\frac{t_{\mathrm{pressure}}}{2}\Big) \tag{A.72}$$

$$\frac{\partial x_{\mathrm{PS}}(t)}{\partial d_{\mathrm{Out}}} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{\partial x_b}{\partial d_{\mathrm{Out}}} + \frac{\partial n_x^{\mathrm{CL}}}{\partial d_{\mathrm{Out}}}\frac{t_{\mathrm{pressure}}}{2}\Big) \tag{A.73}$$

$$\frac{\partial x_{\mathrm{PS}}(t)}{\partial a_0^p} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{1}{2}n_x\Big) \tag{A.74}$$

$$\frac{\partial x_{\mathrm{PS}}(t)}{\partial a_1^p} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{1}{2}n_x\,\xi_{\mathrm{PS}}\Big) \tag{A.75}$$

$$\frac{\partial x_{\mathrm{PS}}(t)}{\partial a_2^p} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{1}{2}n_x\,\xi_{\mathrm{PS}}^2\Big) \tag{A.76}$$

$$\frac{\partial x_{\mathrm{PS}}(t)}{\partial a_3^p} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{1}{2}n_x\,\xi_{\mathrm{PS}}^3\Big) \tag{A.77}$$

$$\frac{\partial x_{\mathrm{PS}}(t)}{\partial a_4^p} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{1}{2}n_x\,\xi_{\mathrm{PS}}^4\Big) \tag{A.78}$$

$$\frac{\partial x_{\mathrm{PS}}(t)}{\partial x_c^{\mathrm{LE}}} = B_0^2(t) + B_1^2(t) + B_2^2(t) \tag{A.79}$$

$$\begin{aligned}\frac{\partial x_{\mathrm{PS}}(t)}{\partial r_{\mathrm{LE}}} &= B_0^2(t)\cos\theta_3 + B_1^2(t)\Big(\cos\theta_3 - \frac{4}{3N_5}\tan\frac{\theta_{\mathrm{LE}}}{4}\sin\theta_3\Big) \\ &\quad + B_2^2(t)\left[\frac{\cos\theta_0 - \cos\theta_3 - \tan\frac{\theta_{\mathrm{LE}}}{4}\big(\frac{8}{3}\sin\theta_3 + \frac{4}{3}\sin\theta_0\big)}{N_6} + \cos\theta_3 - \frac{8}{3N_5}\tan\frac{\theta_{\mathrm{LE}}}{4}\sin\theta_3\right]\end{aligned} \tag{A.80}$$

$$\begin{aligned}\frac{\partial x_{\mathrm{PS}}(t)}{\partial \theta_0^{\mathrm{LE}}} &= B_1^2(t)\frac{1}{3N_5} r_{\mathrm{LE}}\mathrm{sec\_term\_sq\_LE}\sin\theta_3 \\ &\quad + B_2^2(t)\left[\frac{-r_{\mathrm{LE}}\sin\theta_0 + \frac{1}{4}r_{\mathrm{LE}}\mathrm{sec\_term\_sq\_LE}\big(\frac{8}{3}\sin\theta_3 + \frac{4}{3}\sin\theta_0\big) - \frac{4}{3}r_{\mathrm{LE}}\tan\frac{\theta_{\mathrm{LE}}}{4}\cos\theta_0}{N_6}\right. \\ &\quad \left. + \frac{2}{3N_5} r_{\mathrm{LE}}\mathrm{sec\_term\_sq\_LE}\sin\theta_3\right]\end{aligned} \tag{A.81}$$

$$\begin{aligned}\frac{\partial x_{\mathrm{PS}}(t)}{\partial \theta_3^{\mathrm{LE}}} &= B_0^2(t)(-r_{\mathrm{LE}}\sin\theta_3) + B_1^2(t)\left[- r_{\mathrm{LE}}\sin\theta_3 - \frac{1}{3N_5} r_{\mathrm{LE}}\mathrm{sec\_term\_sq\_LE}\sin\theta_3\right. \\ &\quad \left. - \frac{4}{3N_5} r_{\mathrm{LE}}\tan\frac{\theta_{\mathrm{LE}}}{4}\cos\theta_3\right] \\ &\quad + B_2^2(t)\left[\frac{r_{\mathrm{LE}}\sin\theta_3 - \frac{1}{4}r_{\mathrm{LE}}\mathrm{sec\_term\_sq\_LE}\big(\frac{8}{3}\sin\theta_3 + \frac{4}{3}\sin\theta_0\big) - \frac{8}{3}r_{\mathrm{LE}}\tan\frac{\theta_{\mathrm{LE}}}{4}\cos\theta_3}{N_6}\right. \\ &\quad \left. - r_{\mathrm{LE}}\sin\theta_3 - \frac{2}{3N_5} r_{\mathrm{LE}}\mathrm{sec\_term\_sq\_LE}\sin\theta_3 - \frac{8}{3N_5} r_{\mathrm{LE}}\tan\frac{\theta_{\mathrm{LE}}}{4}\cos\theta_3\right]\end{aligned} \tag{A.82}$$

$$\frac{\partial x_{\mathrm{PS}}(t)}{\partial x_c^{\mathrm{TE}}} = B_{n_{\mathrm{PS}}-3}^2(t) + B_{n_{\mathrm{PS}}-2}^2(t) + B_{n_{\mathrm{PS}}-1}^2(t) \tag{A.83}$$

$$\begin{aligned}\frac{\partial x_{\mathrm{PS}}(t)}{\partial r_{\mathrm{TE}}} &= B_{n_{\mathrm{PS}}-3}^2(t)\left[\frac{\cos\theta_3 - \cos\theta_0 + \tan\frac{\theta_{\mathrm{TE}}}{4}\big(\frac{4}{3}\sin\theta_3 + \frac{8}{3}\sin\theta_0\big)}{N_8} + \cos\theta_0 + \frac{8}{3N_7}\tan\frac{\theta_{\mathrm{TE}}}{4}\sin\theta_0\right] \\ &\quad + B_{n_{\mathrm{PS}}-2}^2(t)\Big(\cos\theta_0 + \frac{4}{3N_7}\tan\frac{\theta_{\mathrm{TE}}}{4}\sin\theta_0\Big) + B_{n_{\mathrm{PS}}-1}^2(t)\cos\theta_0\end{aligned} \tag{A.84}$$

$$\begin{aligned}\frac{\partial x_{\mathrm{PS}}(t)}{\partial \theta_0^{\mathrm{TE}}} &= B^2_{n_{\mathrm{PS}}-3}(t)\Bigg[\frac{r_{\mathrm{TE}}\sin\theta_0 - \frac{1}{4}r_{\mathrm{TE}}\sec^2\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\left(\frac{4}{3}\sin\theta_3 + \frac{8}{3}\sin\theta_0\right) + \frac{8}{3}r_{\mathrm{TE}}\tan\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\cos\theta_0}{N_8} \\ &\quad \frac{-r_{\mathrm{TE}}\sin\theta_0}{N_8} - \tfrac{2}{3N_7}r_{\mathrm{TE}}\sec^2\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\sin\theta_0 + \tfrac{8}{3N_7}r_{\mathrm{TE}}\tan\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\cos\theta_0\Bigg] \\ &\quad + B^2_{n_{\mathrm{PS}}-2}(t)\Big[- \tfrac{1}{3N_7}r_{\mathrm{TE}}\sec^2\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\sin\theta_0 + \tfrac{4}{3N_7}r_{\mathrm{TE}}\tan\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\cos\theta_0 - r_{\mathrm{TE}}\sin\theta_0\Big] \\ &\quad + B^2_{n_{\mathrm{PS}}-1}(t)(-r_{\mathrm{TE}}\sin\theta_0)\end{aligned} \tag{A.85}$$

$$\begin{aligned}\frac{\partial x_{\mathrm{PS}}(t)}{\partial \theta_3^{\mathrm{TE}}} &= B^2_{n_{\mathrm{PS}}-3}(t)\Bigg[\frac{-r_{\mathrm{TE}}\sin\theta_3 + \frac{1}{4}r_{\mathrm{TE}}\sec^2\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\left(\frac{4}{3}\sin\theta_3 + \frac{8}{3}\sin\theta_0\right) + \frac{4}{3}r_{\mathrm{TE}}\tan\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\cos\theta_3}{N_8} \\ &\quad + \tfrac{2}{3N_7}r_{\mathrm{TE}}\sec^2\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\sin\theta_0\Bigg] \\ &\quad + B^2_{n_{\mathrm{PS}}-2}(t)\Big[\tfrac{1}{3N_7}r_{\mathrm{TE}}\sec^2\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\sin\theta_0\Big]\end{aligned} \tag{A.86}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{PS}}(t)}{\partial \theta_0^{\mathrm{TE}}} &= B^2_{n_{\mathrm{PS}}-3}(t)\Bigg[\frac{-r_{\mathrm{TE}}\cos\theta_0 + \frac{1}{4}r_{\mathrm{TE}}\sec^2\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\left(\frac{8}{3}\cos\theta_0 + \frac{4}{3}\cos\theta_3\right) + \frac{8}{3}r_{\mathrm{TE}}\tan\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\sin\theta_0}{N_8} \\ &\quad \frac{+r_{\mathrm{TE}}\cos\theta_0}{N_8} + \tfrac{2}{3N_7}r_{\mathrm{TE}}\sec^2\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\cos\theta_0 + \tfrac{8}{3N_7}r_{\mathrm{TE}}\tan\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\sin\theta_0 + r_{\mathrm{TE}}\cos\theta_0\Bigg] \\ &\quad + B^2_{n_{\mathrm{PS}}-2}(t)\Big[\tfrac{1}{3N_7}r_{\mathrm{TE}}\sec^2\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\cos\theta_0 + \tfrac{4}{3N_7}r_{\mathrm{TE}}\tan\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\sin\theta_0 + r_{\mathrm{TE}}\cos\theta_0\Big] \\ &\quad + B^2_{n_{\mathrm{PS}}-1}(t)\Big[r_{\mathrm{TE}}\cos\theta_0\Big]\end{aligned} \tag{A.87}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{PS}}(t)}{\partial \theta_3^{\mathrm{TE}}} &= B^2_{n_{\mathrm{PS}}-3}(t)\Bigg[\frac{r_{\mathrm{TE}}\cos\theta_3 - \frac{1}{4}r_{\mathrm{TE}}\sec^2\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\big(\frac{8}{3}\cos\theta_0+\frac{4}{3}\cos\theta_3\big)+\frac{4}{3}r_{\mathrm{TE}}\tan\left(\frac{\theta_{\mathrm{TE}}}{4}\right)\sin\theta_3}{N_8}\\ &\quad - \tfrac{2}{3N_7}r_{\mathrm{TE}}\sec^2\left(\tfrac{\theta_{\mathrm{TE}}}{4}\right)\cos\theta_0\Bigg]\\ &\quad + B^2_{n_{\mathrm{PS}}-1}(t)\Big[-\tfrac{1}{3N_7}r_{\mathrm{TE}}\sec^2\left(\tfrac{\theta_{\mathrm{TE}}}{4}\right)\cos\theta_0\Big]\end{aligned}\tag{A.88}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{PS}}(t)}{\partial \theta_0^{\mathrm{LE}}} &= B^2_1(t)\Big(-\frac{1}{3N_5}r_{\mathrm{LE}}\sec^2\Big(\frac{\theta_{\mathrm{LE}}}{4}\Big)\cos\theta_3\Big)\\ &\quad + B^2_2(t)\Bigg[\frac{r_{\mathrm{LE}}\cos\theta_0 + r_{\mathrm{LE}}\sec^2\left(\frac{\theta_{\mathrm{LE}}}{4}\right)\Big(-\frac{1}{4}\big(\frac{8}{3}\cos\theta_3+\frac{4}{3}\cos\theta_0\big)\Big)-\frac{4}{3}r_{\mathrm{LE}}\tan\left(\frac{\theta_{\mathrm{LE}}}{4}\right)\sin\theta_0}{N_6}\\ &\quad - \frac{2}{3N_5}r_{\mathrm{LE}}\sec^2\Big(\frac{\theta_{\mathrm{LE}}}{4}\Big)\cos\theta_3\Bigg]\end{aligned}\tag{A.89}$$

$$\frac{\partial y_{\mathrm{PS}}(t)}{\partial y_c} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{\partial y_b}{\partial y_c} + \frac{\partial n_y^{\mathrm{CL}}}{\partial y_c}\frac{t_{\mathrm{pressure}}}{2}\Big) \tag{A.90}$$

$$\frac{\partial y_{\mathrm{PS}}(t)}{\partial \theta_{\mathrm{In}}} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{\partial y_b}{\partial \theta_{\mathrm{In}}} + \frac{\partial n_y^{\mathrm{CL}}}{\partial \theta_{\mathrm{In}}}\frac{t_{\mathrm{pressure}}}{2}\Big) \tag{A.91}$$

$$\frac{\partial y_{\mathrm{PS}}(t)}{\partial \theta_{\mathrm{Out}}} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{\partial y_b}{\partial \theta_{\mathrm{Out}}} + \frac{\partial n_y^{\mathrm{CL}}}{\partial \theta_{\mathrm{Out}}}\frac{t_{\mathrm{pressure}}}{2}\Big) \tag{A.92}$$

$$\frac{\partial y_{\mathrm{PS}}(t)}{\partial d_{\mathrm{In}}} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{\partial y_b}{\partial d_{\mathrm{In}}} + \frac{\partial n_y^{\mathrm{CL}}}{\partial d_{\mathrm{In}}}\frac{t_{\mathrm{pressure}}}{2}\Big) \tag{A.93}$$

$$\frac{\partial y_{\mathrm{PS}}(t)}{\partial d_{\mathrm{Out}}} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{\partial y_b}{\partial d_{\mathrm{Out}}} + \frac{\partial n_y^{\mathrm{CL}}}{\partial d_{\mathrm{Out}}}\frac{t_{\mathrm{pressure}}}{2}\Big) \tag{A.94}$$

$$\frac{\partial y_{\mathrm{PS}}(t)}{\partial a_0^p} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{1}{2}n_y\Big) \tag{A.95}$$

$$\frac{\partial y_{\mathrm{PS}}(t)}{\partial a_1^p} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{1}{2}n_y\,\xi_{\mathrm{PS}}\Big) \tag{A.96}$$

$$\frac{\partial y_{\mathrm{PS}}(t)}{\partial a_2^p} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{1}{2}n_y\,\xi_{\mathrm{PS}}^2\Big) \tag{A.97}$$

$$\frac{\partial y_{\mathrm{PS}}(t)}{\partial a_3^p} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{1}{2}n_y\,\xi_{\mathrm{PS}}^3\Big) \tag{A.98}$$

$$\frac{\partial y_{\mathrm{PS}}(t)}{\partial a_4^p} = \sum_{j=0}^{n_{\mathrm{PS}}-1} B_j^{n_{\mathrm{PS}}-1}(t)\Big(\frac{1}{2}n_y\,\xi_{\mathrm{PS}}^4\Big) \tag{A.99}$$

$$\frac{\partial y_{\mathrm{PS}}(t)}{\partial y_c^{\mathrm{LE}}} = B_0^2(t) + B_1^2(t) + B_2^2(t) \tag{A.100}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{PS}}(t)}{\partial r_{\mathrm{LE}}} &= B_0^2(t)\sin\theta_3 + B_1^2(t)\Big(\sin\theta_3 + \frac{4}{3N_5}\tan\frac{\theta_{\mathrm{LE}}}{4}\cos\theta_3\Big) \\ &+ B_2^2(t)\left[\frac{\sin\theta_0 - \sin\theta_3 + \tan\frac{\theta_{\mathrm{LE}}}{4}\left(\frac{8}{3}\cos\theta_3 + \frac{4}{3}\cos\theta_0\right)}{N_6} + \sin\theta_3 + \frac{8}{3N_5}\tan\frac{\theta_{\mathrm{LE}}}{4}\cos\theta_3\right]\end{aligned} \tag{A.101}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{PS}}(t)}{\partial \theta_3^{\mathrm{LE}}} &= B_0^2(t) r_{\mathrm{LE}}\cos\theta_3 + B_1^2(t)\Big(r_{\mathrm{LE}}\cos\theta_3 + \frac{1}{3N_5} r_{\mathrm{LE}}\mathrm{sec\_term\_sq\_LE}\cos\theta_3 \\ &- \frac{4}{3N_5} r_{\mathrm{LE}}\tan\frac{\theta_{\mathrm{LE}}}{4}\sin\theta_3\Big) \\ &+ B_2^2(t)\left[\frac{-r_{\mathrm{LE}}\cos\theta_3 + \frac{1}{4} r_{\mathrm{LE}}\mathrm{sec\_term\_sq\_LE}\left(\frac{8}{3}\cos\theta_3 + \frac{4}{3}\cos\theta_0\right) - \frac{8}{3} r_{\mathrm{LE}}\tan\frac{\theta_{\mathrm{LE}}}{4}\sin\theta_3}{N_6}\right. \\ &\left. + \frac{2}{3N_5} r_{\mathrm{LE}}\mathrm{sec\_term\_sq\_LE}\cos\theta_3 - \frac{8}{3N_5} r_{\mathrm{LE}}\tan\frac{\theta_{\mathrm{LE}}}{4}\sin\theta_3 + r_{\mathrm{LE}}\cos\theta_3\right]\end{aligned} \tag{A.102}$$

$$\frac{\partial y_{\mathrm{PS}}(t)}{\partial y_c^{\mathrm{TE}}} = B_{n_{\mathrm{PS}}-3}^2(t) + B_{n_{\mathrm{PS}}-2}^2(t) + B_{n_{\mathrm{PS}}-1}^2(t) \tag{A.103}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{PS}}(t)}{\partial r_{\mathrm{TE}}} &= B_{n_{\mathrm{PS}}-3}^2(t)\left[\frac{\sin\theta_3 - \sin\theta_0 - \tan\frac{\theta_{\mathrm{TE}}}{4}\left(\frac{8}{3}\cos\theta_0 + \frac{4}{3}\cos\theta_3\right)}{N_8} + \sin\theta_0 - \frac{8}{3N_7}\tan\frac{\theta_{\mathrm{TE}}}{4}\cos\theta_0\right] \\ &+ B_{n_{\mathrm{PS}}-2}^2(t)\Big(\sin\theta_0 - \frac{4}{3N_7}\tan\frac{\theta_{\mathrm{TE}}}{4}\cos\theta_0\Big) + B_{n_{\mathrm{PS}}-1}^2(t)\sin\theta_0\end{aligned} \tag{A.104}$$

A.5 LE partial derivatives w.r.t design variables

The leading edge partial derivatives w.r.t design variables.

$$\text{sec\_term\_sq\_LE} = \frac{1}{\cos^2\left(\frac{\theta_{\text{LE}}}{4}\right)} \tag{A.105}$$

$$\frac{\partial x_{\text{LE}}(t)}{\partial x_{c,\text{LE}}} = B_0^3(t) + B_1^3(t) + B_2^3(t) + B_3^3(t) \tag{A.106}$$

$$\begin{aligned}\frac{\partial x_{\text{LE}}(t)}{\partial r_{\text{LE}}} &= B_0^3(t)\cos\theta_0 + B_1^3(t)\Big(\cos\theta_0 - \frac{4}{3}\tan\frac{\theta_{\text{LE}}}{4}\sin\theta_0\Big)\\ &+ B_2^3(t)\Big(\cos\theta_3 + \frac{4}{3}\tan\frac{\theta_{\text{LE}}}{4}\sin\theta_3\Big) + B_3^3(t)\cos\theta_3\end{aligned} \tag{A.107}$$

$$\begin{aligned}\frac{\partial x_{\text{LE}}(t)}{\partial \theta_0} &= B_0^3(t)(-r_{\text{LE}}\sin\theta_0) + B_1^3(t)\Big(-r_{\text{LE}}\sin\theta_0 + \frac{1}{3}r_{\text{LE}}\text{sec\_term\_sq\_LE}\sin\theta_0\\ &- \frac{4}{3}r_{\text{LE}}\tan\frac{\theta_{\text{LE}}}{4}\cos\theta_0\Big) - B_2^3(t)\frac{1}{3}r_{\text{LE}}\text{sec\_term\_sq\_LE}\sin\theta_3\end{aligned} \tag{A.108}$$

$$\begin{aligned}\frac{\partial x_{\text{LE}}(t)}{\partial \theta_3} &= B_1^3(t)\Big(-\frac{1}{3}r_{\text{LE}}\text{sec\_term\_sq\_LE}\sin\theta_0\Big)\\ &+ B_2^3(t)\Big(-r_{\text{LE}}\sin\theta_3 + \frac{1}{3}r_{\text{LE}}\text{sec\_term\_sq\_LE}\sin\theta_3\\ &+ \frac{4}{3}r_{\text{LE}}\tan\frac{\theta_{\text{LE}}}{4}\cos\theta_3\Big) + B_3^3(t)(-r_{\text{LE}}\sin\theta_3)\end{aligned} \tag{A.109}$$

$$\frac{\partial y_{\mathrm{LE}}(t)}{\partial y_{c,\mathrm{LE}}} = B_0^3(t) + B_1^3(t) + B_2^3(t) + B_3^3(t) \tag{A.110}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{LE}}(t)}{\partial r_{\mathrm{LE}}} &= B_0^3(t)\sin\theta_0 + B_1^3(t)\Big(\sin\theta_0 + \frac{4}{3}\tan\frac{\theta_{\mathrm{LE}}}{4}\cos\theta_0\Big) \\ &\quad + B_2^3(t)\Big(\sin\theta_3 - \frac{4}{3}\tan\frac{\theta_{\mathrm{LE}}}{4}\cos\theta_3\Big) + B_3^3(t)\sin\theta_3\end{aligned} \tag{A.111}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{LE}}(t)}{\partial \theta_0} &= B_0^3(t)(r_{\mathrm{LE}}\cos\theta_0) + B_1^3(t)\Big(r_{\mathrm{LE}}\cos\theta_0 - \frac{1}{3}r_{\mathrm{LE}}\text{sec\_term\_sq\_LE}\cos\theta_0 \\ &\quad - \frac{4}{3}r_{\mathrm{LE}}\tan\frac{\theta_{\mathrm{LE}}}{4}\sin\theta_0\Big) + B_2^3(t)\frac{1}{3}r_{\mathrm{LE}}\text{sec\_term\_sq\_LE}\cos\theta_3\end{aligned} \tag{A.112}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{LE}}(t)}{\partial \theta_3} &= B_1^3(t)\frac{1}{3}r_{\mathrm{LE}}\text{sec\_term\_sq\_LE}\cos\theta_0 + B_2^3(t)\Big(r_{\mathrm{LE}}\cos\theta_3 - \frac{1}{3}r_{\mathrm{LE}}\text{sec\_term\_sq\_LE}\cos\theta_3 \\ &\quad + \frac{4}{3}r_{\mathrm{LE}}\tan\frac{\theta_{\mathrm{LE}}}{4}\sin\theta_3\Big) + B_3^3(t)(r_{\mathrm{LE}}\cos\theta_3)\end{aligned} \tag{A.113}$$

A.6 TE partial derivatives w.r.t design variables

The trailing edge partial derivatives w.r.t design variables.

$$\text{sec\_term\_sq\_TE} = \frac{1}{\cos^2\left(\frac{\theta_{\mathrm{TE}}}{4}\right)} \tag{A.114}$$

$$\frac{\partial x_{\mathrm{TE}}(t)}{\partial x_{c,\mathrm{TE}}} = B_0^3(t) + B_1^3(t) + B_2^3(t) + B_3^3(t) \tag{A.115}$$

$$\begin{aligned}\frac{\partial x_{\mathrm{TE}}(t)}{\partial r_{\mathrm{TE}}} &= B_0^3(t)\cos\theta_0 + B_1^3(t)\Big(\cos\theta_0 - \frac{4}{3}\tan\frac{\theta_{\mathrm{TE}}}{4}\sin\theta_0\Big) \\ &\quad + B_2^3(t)\Big(\cos\theta_3 + \frac{4}{3}\tan\frac{\theta_{\mathrm{TE}}}{4}\sin\theta_3\Big) + B_3^3(t)\cos\theta_3\end{aligned} \tag{A.116}$$

$$\begin{aligned}\frac{\partial x_{\mathrm{TE}}(t)}{\partial \theta_0} &= B_0^3(t)(-r_{\mathrm{TE}}\sin\theta_0) + B_1^3(t)\Big(- r_{\mathrm{TE}}\sin\theta_0 + \frac{1}{3}r_{\mathrm{TE}}\text{sec\_term\_sq\_TE}\sin\theta_0 \\ &\quad - \frac{4}{3}r_{\mathrm{TE}}\tan\frac{\theta_{\mathrm{TE}}}{4}\cos\theta_0\Big) - B_2^3(t)\frac{1}{3}r_{\mathrm{TE}}\text{sec\_term\_sq\_TE}\sin\theta_3\end{aligned} \tag{A.117}$$

$$\begin{aligned}\frac{\partial x_{\mathrm{TE}}(t)}{\partial \theta_3} &= B_1^3(t)\Big(- \frac{1}{3}r_{\mathrm{TE}}\text{sec\_term\_sq\_TE}\sin\theta_0\Big) \\ &\quad + B_2^3(t)\Big(- r_{\mathrm{TE}}\sin\theta_3 + \frac{1}{3}r_{\mathrm{TE}}\text{sec\_term\_sq\_TE}\sin\theta_3 \\ &\quad + \frac{4}{3}r_{\mathrm{TE}}\tan\frac{\theta_{\mathrm{TE}}}{4}\cos\theta_3\Big) + B_3^3(t)(-r_{\mathrm{TE}}\sin\theta_3)\end{aligned} \tag{A.118}$$

$$\frac{\partial y_{\mathrm{TE}}(t)}{\partial y_{c,\mathrm{TE}}} = B_0^3(t) + B_1^3(t) + B_2^3(t) + B_3^3(t) \tag{A.119}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{TE}}(t)}{\partial r_{\mathrm{TE}}} &= B_0^3(t)\sin\theta_0 + B_1^3(t)\Big(\sin\theta_0 + \frac{4}{3}\tan\frac{\theta_{\mathrm{TE}}}{4}\cos\theta_0\Big) \\ &\quad + B_2^3(t)\Big(\sin\theta_3 - \frac{4}{3}\tan\frac{\theta_{\mathrm{TE}}}{4}\cos\theta_3\Big) + B_3^3(t)\sin\theta_3\end{aligned} \tag{A.120}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{TE}}(t)}{\partial \theta_0} &= B_0^3(t)(r_{\mathrm{TE}}\cos\theta_0) + B_1^3(t)\Big(r_{\mathrm{TE}}\cos\theta_0 - \frac{1}{3}r_{\mathrm{TE}}\text{sec\_term\_sq\_TE}\cos\theta_0 \\ &\quad - \frac{4}{3}r_{\mathrm{TE}}\tan\frac{\theta_{\mathrm{TE}}}{4}\sin\theta_0\Big) + B_2^3(t)\frac{1}{3}r_{\mathrm{TE}}\text{sec\_term\_sq\_TE}\cos\theta_3\end{aligned} \tag{A.121}$$

$$\begin{aligned}\frac{\partial y_{\mathrm{TE}}(t)}{\partial \theta_3} &= B_1^3(t)\frac{1}{3}r_{\mathrm{TE}}\text{sec\_term\_sq\_TE}\cos\theta_0 + B_2^3(t)\Big(r_{\mathrm{TE}}\cos\theta_3 - \frac{1}{3}r_{\mathrm{TE}}\text{sec\_term\_sq\_TE}\cos\theta_3 \\ &\quad + \frac{4}{3}r_{\mathrm{TE}}\tan\frac{\theta_{\mathrm{TE}}}{4}\sin\theta_3\Big) + B_3^3(t)(r_{\mathrm{TE}}\cos\theta_3)\end{aligned} \tag{A.122}$$

Bibliography

[1] Asouti, V., Trompoukis, X., Kampolis, I., Giannakoglou, K.: Unsteady CFD computations using vertex–centered finite volumes for unstructured grids on Graphics Processing Units. International Journal for Numerical Methods in Fluids **67**(2), 232–246 (May 2011)

[2] Cridge, E.: Approximating arcs using cubic bézier curves. `http://ecridge.com` (Jun 2015), [Online; accessed 16-Aug-2025]

[3] Cridge, E.: Approximating arcs using cubic bézier curves. `https://ecridge.com` (June 2015), accessed: 2025-08-18

[4] Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer (2015)

[5] Farin, G.: Curves and Surfaces for CAGD: A Practical Guide. Morgan Kaufmann (2002)

[6] Giannakoglou, K.C., Anagnostopoulos, I., Bergeles, G.: Arithmetic Analysis for Engineers. National Technical University of Athens, School of Mechanical Engineering, 3rd edn. (2003)

[7] Giannakoglou, K.C.: Optimization Methods in Aerodynamics (2006)

[8] Giles, M., Pierce, N.: An introduction to the adjoint approach to design. Flow, Turbulence and Combustion (2000)

[9] Gray, A.: Modern Differential Geometry of Curves and Surfaces. CRC Press, 2nd edn. (1997)

[10] Hanimann, L., Mangani, L., Casartelli, E., Gröschel, E., Fischer, M.: Concept of mixing plane averaging. `https://www.researchgate.net/publication/310617437_Concept_of_Mixing_Plane_Averaging` (2014), accessed: 2025-09-13

[11] Hicks, R.M., Henne, P.A.: Wing design by numerical optimization. Journal of Aircraft **15**(7), 407–412 (1978). https://doi.org/10.2514/3.58379

[12] Jaiswal, S.: Shape parameterization of airfoil shapes using bezier curves. In: Bajpai, R., Chandrasekhar, U. (eds.) Innovative Design and Development Practices in Aerospace and Automotive Engineering, pp. 79–85. Springer (2016). https://doi.org/10.1007/978-981-10-1771-1_13

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

[14] Kampolis, I., Trompoukis, X., Asouti, V., Giannakoglou, K.: CFD–based analysis and two–level aerodynamic optimization on Graphics Processing Units. Computer Methods in Applied Mechanics and Engineering **199**(9–12), 712–722 (2010)

[15] Klavdianos, G.G.: Parameterization of Two-Element Airfoils, Implemented in Aerodynamic Shape Optimization. Diploma thesis, National Technical University of Athens, School of Mechanical Engineering, Fluids Department, Parallel CFD & Optimization Unit, Athens (2025)

[16] Kulfan, B.M., Bussoletti, J.E.: Fundamental parametric geometry representations for aircraft component shapes. In: 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2006)

[17] Kåsa, I.: A circle fitting procedure and its error analysis. IEEE Transactions on Instrumentation and Measurement **IM-25**(1), 8–14 (1976)

[18] Montano, A.A., Coello Coello, C.A.: Evolutionary algorithms applied to multiobjective aerodynamic shape optimization. In: Studies in Computational Intelligence (2011)

[19] Nocedal, J., Wright, S.: Numerical Optimization. Springer, New York (1999)

[20] Papadopoulos, F.: Airfoil Parameterization Using Class–Shape Transformation Functions: Applications in Aerodynamic Shape Optimization. Diploma thesis, National Technical University of Athens, School of Mechanical Engineering, Laboratory of Thermal Turbomachines, Athens (2025)

[21] Pointwise, Inc.: Fidelity Pointwise User Guide (2025), version X.X, `https://www.pointwise.com`

[22] Riškus, A.: Approximation of a cubic Bézier curve by circular arcs and vice versa. Information Technology and Control **35**(4), 346–352 (2006)

[23] Roe, P.L.: Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. Journal of Computational Physics **43**(2), 357–372 (1981)

[24] Samareh, J.A.: A survey of shape parameterization techniques. Tech. Rep. NASA/CP-1999-209136, NASA Langley Research Center (1999)

[25] Savio, L., et al.: Time domain modeling of propeller forces due to ventilation in static and dynamic conditions. Journal of Ship Research **61**(4), 198–213 (2017). https://doi.org/10.5957/JOSR.2017.61.4.198

[26] Sederberg, T., Parry, S.: Free-form deformation of solid geometric models. In: Proceedings of SIGGRAPH. pp. 151–160. ACM Press, New York (1986). https://doi.org/10.1145/15886.15903

[27] Spalart, P., Allmaras, S.: A one-equation turbulence model for aerodynamic flows. AIAA Paper 1992-439, 30th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, USA (January 6–9 1992)

[28] Stephan, B., Gallus, H., Niehuis, R.: Experimental investigations of tip clearance flow and its influence on secondary flows in a 1-1/2 stage axial turbine. In: Proceedings of ASME Turbo Expo 2000. No. 2000-GT-613, Munich, Germany (May 8-11 2000)

[29] Trompoukis, X.: Numerical Solution of Aerodynamic-Aeroelasticity Problems in Graphics Card Processors. Ph.D. thesis, National Technical University of Athens, Laboratory Of Thermal Turbomachines, Athens, Greece (2012)

[30] Trompoukis, X., Asouti, V., Kampolis, I., Giannakoglou, K.: CUDA implementation of vertex–centered, finite volume CFD methods on unstructured grids with flow control applications. In: GPU Computing Gems, chap. 17, pp. 207–224. Addison-Wesley Professional (2011)

[31] Tsiakas, K.: Development of Shape Parameterization Techniques, a Flow Solver and its Adjoint, for Optimization on GPUs: Turbomachinery and External Aerodynamics Applications. Ph.D. thesis, National Technical University of Athens, Laboratory Of Thermal Turbomachines, Athens, Greece (2019)

[32] Tsiakas, K.T., Trompoukis, X.S., Asouti, V.G., Giannakoglou, K.C.: Shape optimization using the continuous adjoint method and volumetric nurbs on a many gpu system. In: 8th GRACM International Congress on Computational Mechanics. Volos, Greece (July 2015)

[33] Vasilopoulos, I., Agarwal, D., Meyer, M., Robinson, T., Armstrong, C.: Linking parametric cad with adjoint surface sensitivities. In: European Congress on Computational Methods in Applied Sciences and Engineering (2016)

[34] Walraevens, R., Gallus, H.: Ercoftac sig on 3d turbomachinery flow prediction: Testcase 6 – 1-1/2 stage axial flow turbine. Tech. rep., Institut für Strahlantriebe und Turboarbeitsmaschinen, RWTH Aachen, University of Technology, Templergraben 55, D-52062 Aachen, Germany (nd)

[35] Zangeneh, M.: A compressible three-dimensional design method for radial and mixed flow turbomachinery blades. International Journal for Numerical Methods in Fluids **13**(5), 599–624 (1991). https://doi.org/10.1002/fld.1650130502

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

Αεροδυναμική Βελτιστοποίηση Μορφής Αεροτομών Πτερυγίων Στροβίλων με χρήση της Συζυγούς Μεθόδου και Προσαρμοσμένης Παραμετροποίησης

Διπλωματική Εργασία - Εκτενής Περίληψη στα Ελληνικά

Θεόδωρος Παναγιώτης Κόκκος

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

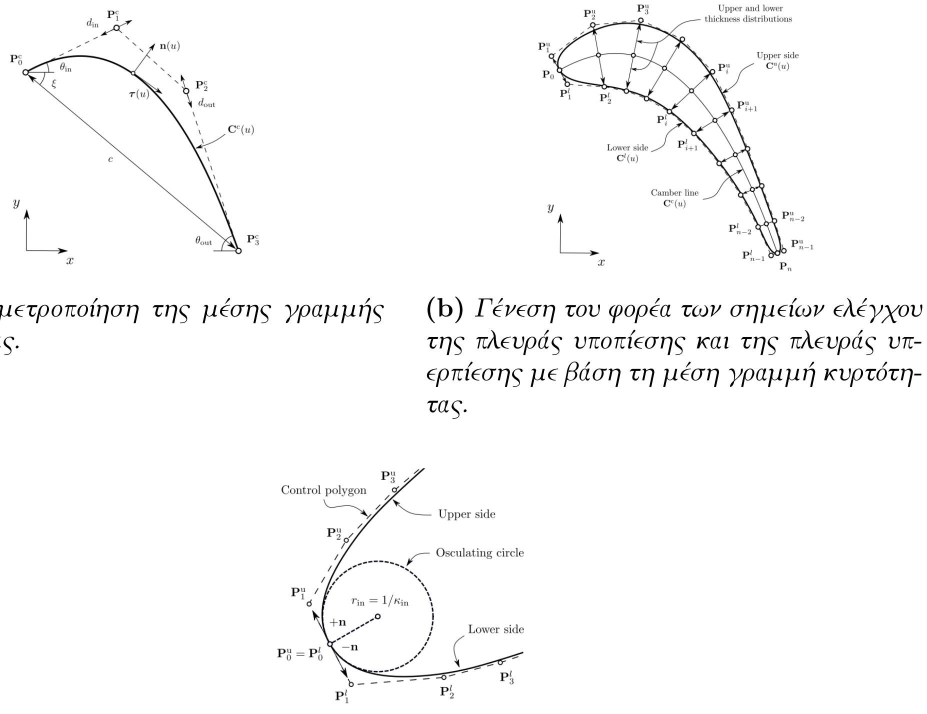
Αθήνα, 2025

Εισαγωγή

Ο αεροδυναμικός σχεδιασμός των πτερυγίων ενός στροβίλου είναι κρίσιμος για την απόδοση, την αποτελεσματικότητα και την αξιοπιστία του. Οι παραδοσιακές μέθοδοι σχεδιασμού βασίζονται σε εμπειρικές μεθόδους και επαναληπτικές δοκιμές γεγονός που τις καθιστά χρονοβόρες. Παράλληλα, η αυξανόμενη ζήτηση για υψηλότερη απόδοση, σε συνδυασμό με την πρόοδο της υπολογιστικής ισχύος, έχει καταστήσει δυνατές πιο προηγμένες προσεγγίσεις σχεδιασμού. Επιπλέον, η βελτιστοποίηση με βάση τη συζυγή μέθοδο συμπληρώνει αυτήν την προσέγγιση, επιτρέποντας τον αποτελεσματικό υπολογισμό των παραγώγων ευαισθησίας των στόχων απόδοσης ως προς πολλές μεταβλητές σχεδιασμού, με χαμηλό συγκρτικά με άλλες μεθόδους υπολογιστικό κόστος. Ακόμη, η παραμετροποίηση με καμπύλες Bézier παρέχει ευέλικτη και ομαλή αναπαράσταση του σχήματος των αεροτομών, επιτρέποντας ακριβείς τροποποιήσεις ενώ διατηρεί τη γεωμετρική συνέχεια. Η ενσωμάτωση του αντίστροφου γεωμετρικού σχεδιασμού, της βελτιστοποίησης μορφής (ΒΜ) με τη συζυγή μέθοδο και της παραμετροποίησης Bézier προσφέρουν ένα πλαίσιο για τη βελτίωση της απόδοσης αεροτομών πτερυγίων στροβίλου. Η διπλωματική εργασία εφαρμόζει αυτό το πλαίσιο αυτό και προτείνει μία νέα μέθοδο παραμετροποίησης αεροτομών, με χρήση των καμπυλών Bézier, ενισχύοντας ταυτόχρονα την αποδοτικότητα στη ΒΜ και αντιμετωπίζοντας τις σύγχρονες σχεδιαστικές-γεωμετρικές προκλήσεις που μπορεί να προκύψουν κατά την εφαρμογή της.

Η Προτεινόμενη Παραμετροποίηση

Στην εργασία αυτή, η παραμετροποίηση της αεροτομής βασίζεται σε μια σχεδιαστική προσέγγιση, σύμφωνα με την οποία η γεωμετρία της αεροτομής αναπαρίσταται μέσω πέντε διακριτών τμημάτων, εν ονόματι την πλευρά υποπίεσης (suction side) και υπερπίεσης (pressure side), ορισμένες σε σχέση με τη μέση γραμμή κυρτότητας (mean camber line), και τις περιοχές της ακμής πρόσπτωσης (leading edge) και της ακμής εκφυγής (trailing edge), οι οποίες μοντελοποιούνται ως κυκλικά τόξα. Όλα τα τμήματα κατασκευάζονται με χρήση καμπυλών Bézier και απαιτούν τον ορισμό, συνολικά, 26 μεταβλητών σχεδιασμού με άμεση φυσική σημασία. Το κεφάλαιο αυτό παρουσιάζει τη διαμόρφωση αυτών των καμπυλών και περιγράφει τις μεταβλητές σχεδιασμού που απαιτεί καθεμιά, παρέχοντας τη βάση για την υλοποίηση του αντίστοιχου κώδικα σε C++.


d_{in}
$\mathbf{P}_1^{\text{c}}$
$\mathbf{n}(u)$
$\mathbf{P}_0^{\text{c}}$
θ_{in}
ξ
$\boldsymbol{\tau}(u)$
$\mathbf{P}_2^{\text{c}}$
d_{out}
$\mathbf{C}^{\text{c}}(u)$
c
y
x
θ_{out}
$\mathbf{P}_3^{\text{c}}$
$\mathbf{P}_2^{\text{u}}$
$\mathbf{P}_3^{\text{u}}$
Upper and lower thickness distributions
$\mathbf{P}_1^{\text{u}}$
$\mathbf{P}_0$
$\mathbf{P}_1^{l}$
$\mathbf{P}_2^{l}$
Upper side $\mathbf{C}^{\text{u}}(u)$
$\mathbf{P}_i^{\text{u}}$
$\mathbf{P}_{i+1}^{\text{u}}$
$\mathbf{P}_i^{l}$
Lower side $\mathbf{C}^{l}(u)$
$\mathbf{P}_{i+1}^{l}$
Camber line $\mathbf{C}^{\text{c}}(u)$
$\mathbf{P}_{n-2}^{l}$
$\mathbf{P}_{n-2}^{\text{u}}$
$\mathbf{P}_{n-1}^{\text{u}}$
$\mathbf{P}_{n-1}^{l}$
$\mathbf{P}_n$
y
x
Control polygon
$\mathbf{P}_3^{\text{u}}$
Upper side
$\mathbf{P}_2^{\text{u}}$
Osculating circle
$\mathbf{P}_1^{\text{u}}$
$r_{\text{in}} = 1/\kappa_{\text{in}}$
$+\mathbf{n}$
Lower side
$\mathbf{P}_0^{\text{u}} = \mathbf{P}_0^{l}$
$-\mathbf{n}$
$\mathbf{P}_1^{l}$
$\mathbf{P}_2^{l}$
$\mathbf{P}_3^{l}$

(a) *Παραμετροποίηση της μέσης γραμμής κυρτότητας.*

(b) *Γένεση του φορέα των σημείων ελέγχου της πλευράς υποπίεσης και της πλευράς υπερπίεσης με βάση τη μέση γραμμή κυρτότητας.*

(c) *Η μοντελοποίηση των περιοχών γύρω από το σημείο πρόσπτωσης και το σημείο εκφυγής με κυκλικά τόξα.*

Σχήμα 1: *Γένεση με ξεχωριστή παραμετροποίηση των τμημάτων της αεροτομής.*

Τα κυκλικά τόξα ενώνονται με την πλευρά υποπίεσης και την πλευρά υπερπίεσης με συνέχεια πρώτης και δεύτερης παραγώγου (συνέχεια τύπου C^2), ενώ οι 26 μεταβλητές σχεδιασμού που πρέπει να προσδιοριστούν ώστε η παραμετροποίηση να παράξει μία αεροτομή παρουσιάζονται στον Πίνακα 1. Τονίζεται ότι σε όλες τις περιπτώσεις οι μεταβλητές αυτές προσδιορίζουν τη θέση των σημείων ελέγχου (control points) των αντίστοιχων καμπυλών Bézier.

Όνομα μεταβλητής	Σύμβολο
Κυκλικά τόξα	
Συντεταγμένες κέντρων κυκλικών τόξων	$(x_{cLE}, y_{cLE}), (x_{cTE}, y_{cTE})$
Ακτίνα κυκλικών τόξων	r_{LE}, r_{TE}
Γωνίες εκκίνησης κυκλικών τόξων	$\theta_{0LE}, \theta_{0TE}$
Γωνίες λήξης κυκλικών τόξων	$\theta_{3LE}, \theta_{3TE}$
Πλευράς υποπίεσης και υπερπίεσης	
Συντελεστές πάχους πλευράς αναρρόφησης	$a_{0s}, a_{1s}, a_{2s}, a_{3s}, a_{4s}$
Συντελεστές πάχους πλευράς πίεσης	$a_{0p}, a_{1p}, a_{2p}, a_{3p}, a_{4p}$
Μέση γραμμή καμπυλότητας	
Μήκος χορδής	c
Γωνία κλίσης	ζ
Γωνία μετάλλου εισόδου	α'_1
Γωνία μετάλλου εξόδου	α'_2
Λόγος εφαπτομένης εισόδου	d_{in}
Λόγος εφαπτομένης εξόδου	d_{out}

Πίνακας 1: *Μεταβλητές σχεδιασμού για το σημείο πρόσπτωσης/εκφυγής, τις πλευρές υποπίεσης/υπερπίεσης και τη μέση γραμμή καμπυλότητας.*

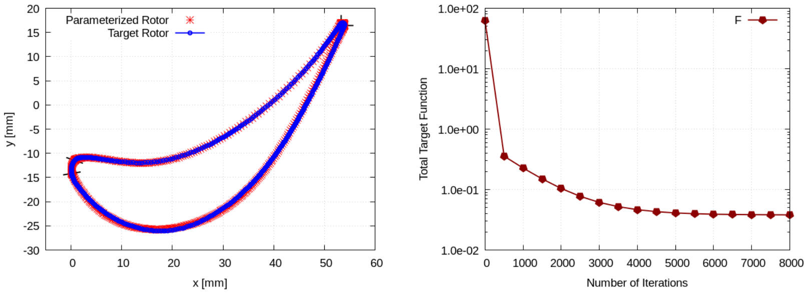
Αλγόριθμος Αντίστροφου Γεωμετρικού Σχεδιασμού

Η βέλτιστη προσαρμογή (best-fit optimization) που αναπτύχθηκε στην παρούσα εργασία αποτελεί μία μορφή αντίστροφου γεωμετρικού σχεδιασμού. Αντί της προδιαγραφής αεροδυναμικών στόχων, παρέχεται μία γεωμετρία αεροτομής, ως αναφορά, και ο αλγόριθμος υπολογίζει τις φυσικές μεταβλητές σχεδιασμού που την αναπαράγουν με τον καλύτερο δυνατό τρόπο. Η μέθοδος βασίζεται σε έναν βρόχο βέλτιστης προσαρμογής μεταξύ της δοθείσας γεωμετρίας της αεροτομής και αυτής που κατασκευάζεται με χρήση της προσαρμοσμένης παραμετροποίησης. Για τον σκοπό αυτό, αναπτύχθηκε κώδικας σε γλώσσα C++, ο οποίος παρουσιάζεται κατωτέρω.

Για την επίτευξη του αντίστροφου γεωμετρικού σχεδιασμού ορίστηκε μία συνάρτηση κόστους, προς ελαχιστοποίηση, η οποία αποτελεί την ευκλίδεια γεωμετρική απόσταση των σημειών της παραμετροποιημένης αεροτομής με τα αντίστοιχα της δοσμένης, υψομένη στο τετράγωνο, όπως φαίνεται στην Εξίσωση (1). Η μέθοδος που χρησιμοποιήθηκε είναι αυτή της απότομης καθόδου και επειδή όλα τα τμήματα της αεροτομής περιγράφονται με τις καμπύλες Bézier ο υπολογισμός του διανύσματος κλίσης (grad) πραγματοποιείται αναλυτικά.

$$\mathrm{F} = \sum_{i=0}^{N} \left[\left(x_i^B - x_i^T\right)^2 + \left(y_i^B - y_i^T\right)^2 \right] \tag{1}$$

Ως αεροτομές στόχοι τέθηκαν οι γεωμετρίες των πτερυγίων σταθερής και κινητής πτερύγωσης ενός μονοβάθμιου αξονικού στροβίλου ο οποίος ήταν διαθέσιμος στο εργαστήριο από προηγούμενες μελέτες. Τα αποτελέσματα του αντίστροφου γεωμετρικού σχεδιασμού για τις περιπτώσεις αυτές, παρουσιάζονται κατωτέρω.



(a) *Προσέγγιση της αεροτομής του πτερυγίου κινητής πτερύγωσης με χρήση της προσαρμοσμένης παραμετροποίησης.*

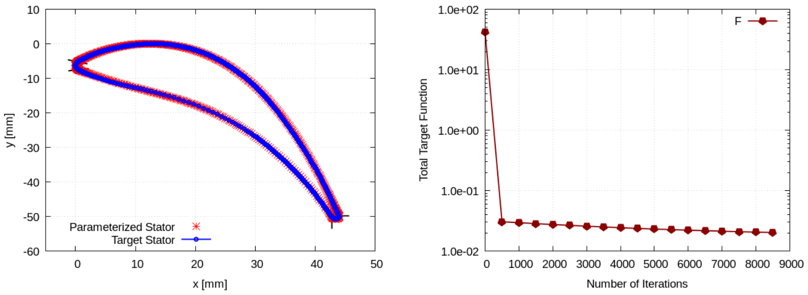
(b) *Κινητή Πτερύγωση: Σύγκλιση της συνάρτησης κόστους.*

Σχήμα 2: *Κινητή Πτερύγωση: Γεωμετρική προσέγγιση της αεροτομής και σύγκλιση της συνάρτησης κόστους.*

Τμήμα	**Αριθμός σημείων**	**Συνάρτηση κόστους**
LE	23	4.300×10^{-4}
TE	20	7.935×10^{-4}
SS	264	5.300×10^{-2}
PS	149	2.236×10^{-2}
Σύνολο	456	3.800×10^{-2}

Πίνακας 2: *Κινητή Πτερύγωση: Αριθμός σημείων και αντίστοιχα σφάλματα για κάθε τμήμα της αεροτομής.*

Το τμήμα γύρω από το σημείο πρόσπτωσης απαιτεί περισσότερες επαναλήψεις για να συγκλίνει σε σύγκριση με το αντίστοιχο εκφυγής, αλλά επιτυγχάνει μικρότερο τελικό σφάλμα ($4.30{\times}10^{-4}$ έναντι $7.935{\times}10^{-4}$). Αυτό οφείλεται στο ότι η περιοχή γύρω από το σημείο πρόσπτωσης έχει σχήμα που μοιάζει περισσότερο με ελλειπτικό τόξο παρά με κυκλικό, σε αντίθεση με το σημείο εκφυγής, όπως απεικονίζεται στο Σχήμα 2α. Αντίθετα, οι πλευρές υποπίεσης και υπερπίεσης παρουσιάζουν παρόμοιες πορείες σύγκλισης και επιτυγχάνουν τελικές τιμές της συνάρτησης κόστους της τάξης του 10^{-2}. Αντίστοιχα αποτελέσματα παρουσιάζονται και για τη γεωμετρία του πτερυγίου σταθερής πτερύγωσης.



(a) *Προσέγγιση της αεροτομής του πτερυγίου σταθερής πτερύγωσης με χρήση της προσαρμοσμένης παραμετροποίησης.*

(b) *Σταθερή Πτερύγωση: Σύγκλιση της συνάρτησης κόστους.*

Σχήμα 3: *Σταθερή Πτερύγωση: Γεωμετρική προσέγγιση της αεροτομής και σύγκλιση της συνάρτησης κόστους.*

Τμήμα	**Αριθμός σημείων**	**Συνάρτηση κόστους**
LE	40	2.188×10^{-5}
TE	22	9.931×10^{-4}
SS	341	1.265×10^{-2}
PS	180	4.097×10^{-2}
Σύνολο	456	3.800×10^{-2}

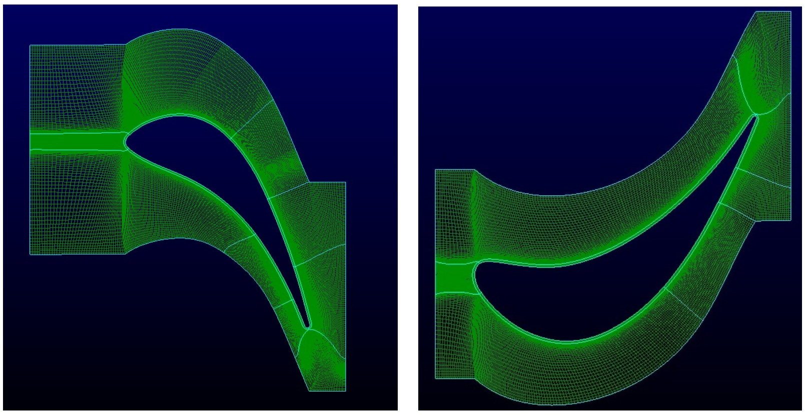
Πίνακας 3: *Σταθερή Πτερύγωση: Αριθμός σημείων και αντίστοιχα σφάλματα για κάθε τμήμα της αεροτομής.*

Σε αυτήν την περίπτωση, η σύγκλιση του τμήματος γύρω από το σημείο πρόσπτωσης και το σημείο εκφυγής είναι παρόμοια, γεγονός που υποδεικνύει ότι το πρώτο προσομοιάζει περισσότερο με κυκλικό τόξο. Η ακρίβεια που σημειώνεται στα τμήματα αυτά είναι της τάξης του 10^{-5} και 10^{-4} , αντίστοιχα. Όσον αφορά την πλευρά υποπίεσης και την πλευρά υπερπίεσης, επιτυγχάνουν τελική τιμή της συνάρτησης κόστους της τάξης του 10^{-2}. Συνολικά, το σφάλμα της διαδικασίας αντίστροφου γεωμετρικού σχεδιασμού με χρήση της προτεινόμενης παραμετροποίησης παραμένει χαμηλό, αντικατοπτρίζοντας την ακρίβεια της μεθοδολογίας.

Πιστοποίηση των αποτελεσμάτων του Αντίστροφου Γεωμετρικού Σχεδιασμού με χρήση του PUMA

Δεδομένης της διαθεσιμότητας πειραματικών δεδομένων για τον μονοβάθμιο αξονικό στρόβιλο από τον οποίο προήλθαν οι γεωμετρίες των αεροτομών σταθερής και κινητής πτερύγωσης, κρίθηκε απαραίτητη η εκτέλεση μίας CFD ανάλυσης ως μέσο επικύρωσης του μοντέλου τόσο της αρχικής γεωμετρίας όσο και αυτής που προέκυψε από τον αλγόριθμο αντίστροφου γεωμετρικού σχεδιασμού, πριν τη διαδικασία της αεροδυναμικής ΒΜ. Όλες οι αναλύσεις πραγματοποιήθηκαν με χρήση του επιλύτη PUMA ο οποίος αναπτύχθηκε στο Εργαστήριο Θερμικών Στροβιλομηχανών του ΕΜΠ από τη ΜΠΥΡΒ.

Για την γένεση όλων των πλεγμάτων έγινε χρήση του λογισμικού Fidelity Pointwise (by Cadense), ενώ ο τύπος τους είναι multiblock δομημένα πλέγματα. Ιδιαίτερη έμφαση δόθηκε στην τοπολογία γύρω από τα τοιχώματα με σκοπό την καλύτερη δυνατή αποτύπωση της ροής στις περιοχές αυτές. Ως οριακές συνθήκες στην είσοδο τέθηκαν τα ολικά μεγέθη της πίεσης και της θερμοκρασίας καθώς και η γωνία της ροής, ενώ στην λεξοδο τέθηκε η στατική πίεση. Το μοντέλο τύρβης που επιλύθηκε είναι το μίας εξίσωσης Spalart–Allmaras χωρίς τη χρήση συναρτήσεων τοίχου μιας και για τις συγκεκριμένες συνθήκες ροής τα πλέγματα φέρουν $y^+ < 1$. Τα τοιχώματα μοντελοποιήθηκαν ως αδιαβατικά, ενώ για την αλληλεπίδραση της σταθερής με την κινητή πτερύγωση επιλέχθηκε, η τεχνική του επιπέδου ανάμιξης (mixing plane technique).



(a) *Πλέγμα σταθερής πτερύγωσης.* (b) *Πλέγμα κινητής πτερύγωσης.*

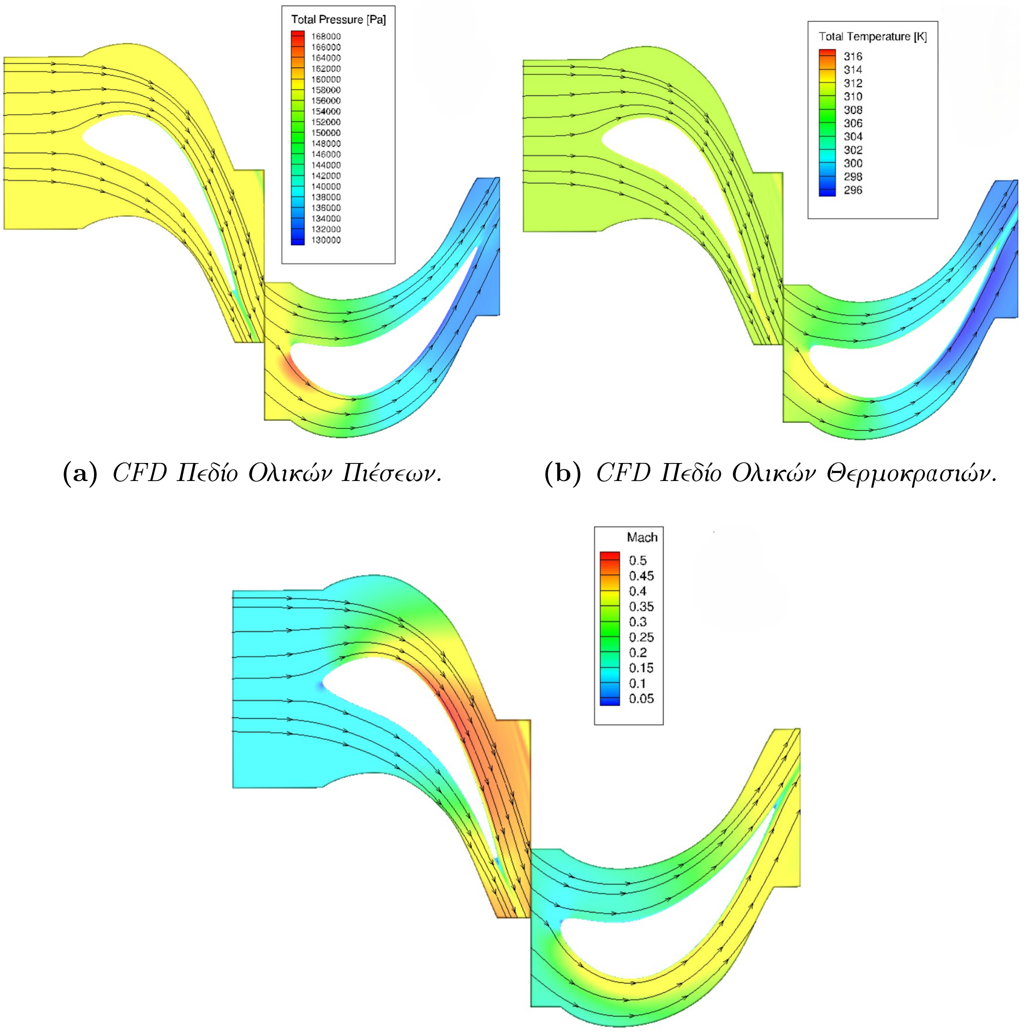
Σχήμα 4: *Τα πλέγματα της σταθερής και της κινητής πτερύγωσης του μονοβάθμιου αξονικού στροβίλου.*

Παρακάτω παρουσιάζονται ενδεικτικά τα πεδία των ολικών θερμοκρασιών, πιέσεων και του αριθμού Mach καθώς και ένας συγκεντρωτικός πίνακας μεταξύ των πειραματικών δεδομένων και των αποτελεσμάτων του PUMA για τις δύο περιπτώσεις (γεωμετρία αναφοράς και προσαρμοσμένη).

Μεγεθος	Πειραματικό [Pa]	PUMA Αναφορά [Pa]	PUMA Προσαρμοσμένη [Pa]	Ποσοστιαία Απόκλιση Αναφορά [%]	Ποσοστιαία Απόκλιση Προσαρμοσμένη [%]
p_{t0}	1.585×10^5	1.585×10^5	1.585×10^5	0	0
p_0	1.574×10^5	1.565×10^5	1.565×10^5	-0.575	-0.575
p_{t1}	1.521×10^5	1.580×10^5	1.584×10^5	3.833	3.977
p_1	1.338×10^5	1.381×10^5	1.386×10^5	3.164	3.463
p_{t2}	1.326×10^5	1.327×10^5	1.331×10^5	0.093	0.37
p_2	1.303×10^5	1.303×10^5	1.303×10^5	0	0
$\Pi\mathrm{P} = p_{t0}/p_{t2}$	1.1941	1.1945	1.1881	0.030	0.034

Πίνακας 4: *Σύγκριση των πειραματικών και αριθμητικών τιμών της πίεσης σε διάφορες θέσεις του στροβίλου για την αρχική γεωμετρία και την προσαρμοσμένη.*

Όπως φαίνεται στον Πίνακα 4, ο PUMA προβλέπει τις πιέσεις ροής κατά μήκος του σταδίου με υψηλή ακρίβεια. Πιο συγκεκριμένα, όλες οι αποκλίσεις, εκτός από αυτές στη θέση 1, είναι κάτω από 1%. Όσον αφορά την έξοδο του στάτορα, ο PUMA υποεκτιμά ελαφρώς τις απώλειες τόσο στην ολική όσο και στη στατική πίεση. Αυτό πιθανότατα εξηγείται από την υπόθεση μοντελοποίησης ότι τα τοιχώματα των πτερυγίων της σταθερής πτερύγωσης είναι αδιαβατικά. Επιπλέον, η απόκλιση μεταξύ της αρχικής γεωμετρίας (αναφορά) και της προσαρμοσμένης είναι ιδιαίτερα μικρή.



(a) *CFD Πεδίο Ολικών Πιέσεων.*

(b) *CFD Πεδίο Ολικών Θερμοκρασιών.*

(c) *CFD Πεδίο Αριθμού Mach.*

Σχήμα 5: *Τα πεδία των ολικών πιέσεων και θερμοκρασιών καθώς και του αριθμού Mach όπως αυτά προέκυψαν από την CFD ανάλυση.*

Ενσωμάτωση της Προσαρμοσμένης Παραμετροποίησης στην Αεροδυναμική ΒΜ

Το πρώτο βήμα για την έναρξη του κύκλου βελτιστοποίησης είναι η εφαρμογή του αλγορίθμου αντίστροφου γεωμετρικού σχεδιασμού, προκειμένου να προσδιοριστεί η αρχική γεωμετρία αναφοράς. Στη συνέχεια, δημιουργείται το υπολογιστικό πλέγμα και πραγματοποιείται η αρχική προσομοίωση CFD. Το επόμενο βήμα είναι ο υπολογισμός των παραγώγων ευαισθησίας. Η μέθοδος που χρησιμοποιείται για το σκοπό αυτό είναι η συνεχής μέθοδος, μια αποδοτική αριθμητική τεχνική που αξιολογεί τις παραγώγους της συνάρτησης κόστους ως προς τις μεταβλητές σχεδιασμού με υπολογιστικό κόστος ανεξάρτητο από τον αριθμό αυτών των μεταβλητών. Οι διαδικασίες επίλυσης τόσο του πρωτεύοντος όσο και του συνεχούς αδθοιντ προβλήματος πραγματοποιούνται χρησιμοποιώντας το λογισμικό PUMA. Οι παράγωγοι ευαισθησίας της συνάρτησης κόστους στους κόμβους ορίων της αεροτομής υπολογίζονται χρησιμοποιώντας τον κανόνα αλυσίδας, σύμφωνα με την εξίσωση:

$$\frac{\delta F}{\delta b_n} = \frac{\delta F}{\delta x_i}\frac{\delta x_i}{\delta b_n} + \frac{\delta F}{\delta y_i}\frac{\delta y_i}{\delta b_n} + \frac{\delta F}{\delta z_i}\frac{\delta z_i}{\delta b_n} \tag{2}$$

Για τον σκοπό αυτό, αναπτύχθηκε ένας κώδικας με την ονομασία `GRID_SENSITIVITIES.exe` σε C++, ο οποίος υπολογίζει τις απαιτούμενες παραγώγους και τις επικοινωνεί στον PUMA μέσω ενός κατάλληλα μορφοποιημένου αρχείου, ώστε να μπορούν να υπολογιστούν οι παράγωγοι ευαισθησίας της συνάρτησης κόστους στους κόμβους της αεροτομής. Για τους υπόλοιπους κόμβους, το λογισμικό PUMA χρησιμοποιεί την τεχνική του Ελαστικού Μέσου για να μεταφέρει τις πληροφορίες των παραγώγων ευαισθησίας των κόμβων ορίων της αεροτομής σε όλους τους υπόλοιπους.

Μόλις υπολογιστούν οι παράγωγοι ευαισθησίας της συνάρτησης κόστους για όλους τους κόμβους, εκτελείται η ενημέρωση των μεταβλητών σχεδιασμού, η δημιουργία της νέας αεροτομής και η προσαρμογή του πλέγματος στη νέα γεωμετρία. Σε αυτό το στάδιο απαιτείται περαιτέρω εξωτερική παρέμβαση στον PUMA προκειμένου να υπολογιστεί η μετατόπιση των κόμβων ορίων του νέου πλέγματος. Για τον σκοπό αυτό εκτελείται ξανά ο ίδιος κώδικας, `GRID_SENSITIVITIES.exe` . Ο κώδικας αυτός διαβάζει τις ενημερωμένες μεταβλητές σχεδιασμού από ένα αρχείο εισόδου (designVars.dat), ανακατασκευάζει τη γεωμετρία της τρέχουσας αεροτομής χρησιμοποιώντας την προσαρμοσμένη παραμετροποίηση υπολογίζει τις παραμορφώσεις των κόμβων μεταξύ της αρχικής και της τρέχουσας διαμόρφωσης της αεροτομής.

$$\vec{b}_{k+1} = \vec{b}_k - B_k^{-1} \left.\frac{dF}{d\vec{b}}\right|_{\vec{b}_k} \tag{3}$$

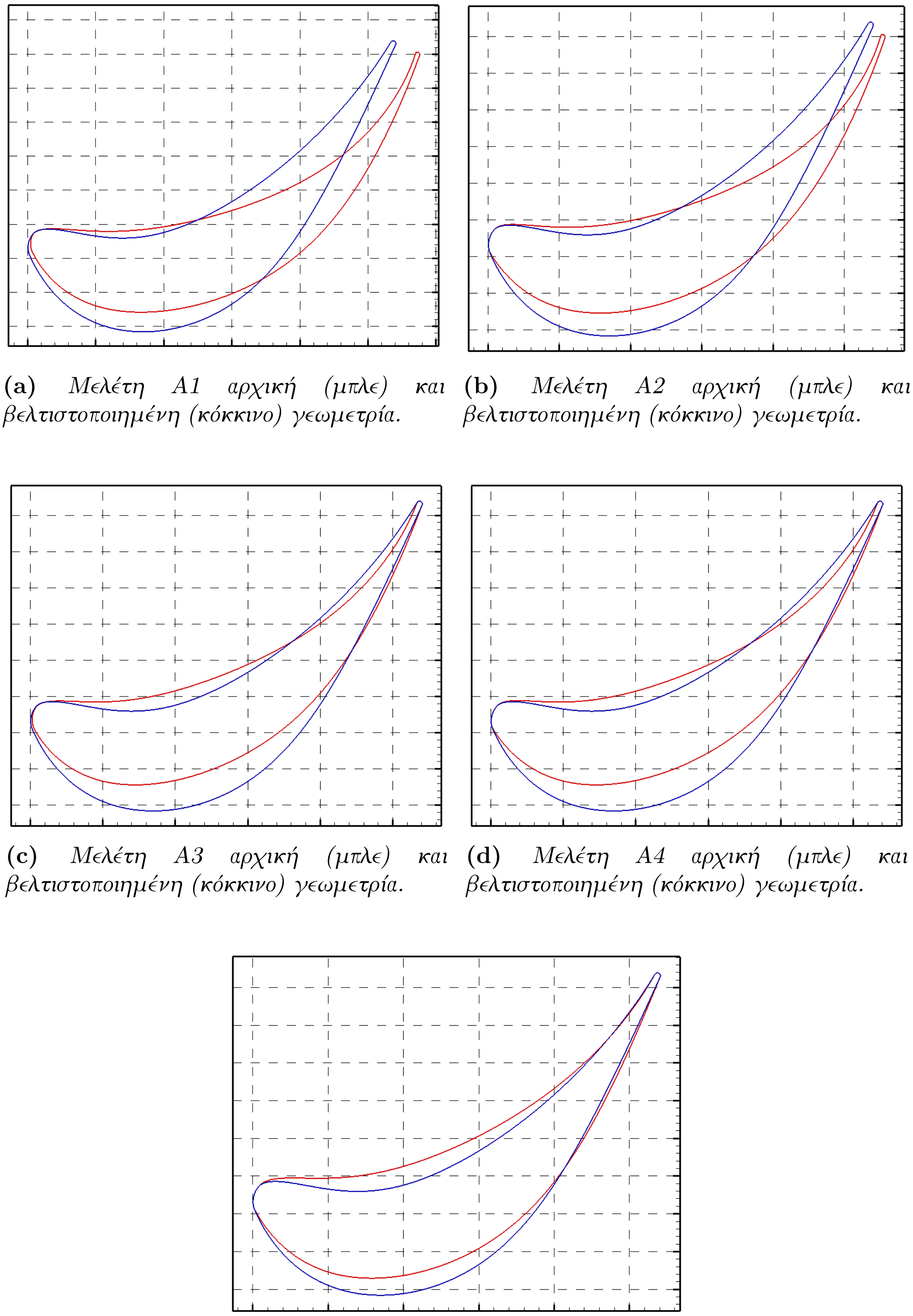
Ο χρησιμοποιούμενος αλγόριθμος BM είναι μια μέθοδος Quasi-Newton για προβλήματα μη γραμμικής βελτιστοποίησης με περιορισμούς. Συγκεκριμένα, χρησιμοποιείται η μέθοδος Sequential Least Squares Quadratic Programming (SLSQP), η οποία εφαρμόζει το σχήμα BFGS για την προσέγγιση των δευτέρων παραγώγων.

Μελέτες Αεροδυναμικής ΒΜ

Σημείο έναρξης της αεροδυναμικής ΒΜ αποτέλεσε η γεωμετρία της αεροτομής των πτερυγίων κινητής πτερύγωσης του μονοβάθμιου αξονικού στροβίλου όπως αυτή προσσεγγίστηκε από τον αλγόριθμο αντίστροφου γεωμετρικού σχεδιασμού μιας και αυτή δεν διαφέρει σημαντικά με την αρχική, αεροδυναμικά και γεωμετρικά, όπως αποδείχθηκε. Για τη βελτιστοποίηση χρησιμοποιήθηκε η συνεχής συζυγής μέθοδος μέσω του λογισμικού PUMA για την ελαχιστοποίηση των απωλειών ολικής πίεσης, υπό τον περιορισμό η ανηγμένη παροχή καθώς και η γωνία εξόδου της ροής να μην μεταβληθιύν περισσότερο από 0.1%. Επιπλέον, σε κάθε μελέτη επιβλήθηκαν διαφορετικοί γεωμετρικοί περιορισμοί όπς περιγράφονται στον Πίνακα 5. Τόσο τα γεωμετρικά όσο και τα αριθμητικά αποτελέσματα παρουσιάζονται στον Πίνακα 5.

Μελέτη	**Αντικειμενική Συνάρτηση**	**Περιορισμοί Ροής**	**Γεωμετρικοί Περιορισμοί**
A1	Ελαχιστοποίηση Απωλειών Ολικής Πίεσης	Ανηγμένη Παροχή & Γωνία Εξόδου	-
A2			Σταθερό LE
A3	Ίδιο	Ίδιο	Σταθερό TE
A4			Σταθερό LE, TE
A5			Σταθερό LE, TE, SS, PS

Πίνακας 5: *Τα χαρακτηριστικά των διαφορετικών μελετών αεροδυναμικής ΒΜ.*



(a) *Μελέτη A1 αρχική (μπλε) και βελτιστοποιημένη (κόκκινο) γεωμετρία.*

(b) *Μελέτη A2 αρχική (μπλε) και βελτιστοποιημένη (κόκκινο) γεωμετρία.*

(c) *Μελέτη A3 αρχική (μπλε) και βελτιστοποιημένη (κόκκινο) γεωμετρία.*

(d) *Μελέτη A4 αρχική (μπλε) και βελτιστοποιημένη (κόκκινο) γεωμετρία.*

(e) *Μελέτη A5 αρχική (μπλε) και βελτιστοποιημένη (κόκκινο) γεωμετρία.*

Σχήμα 6: *Βελτιστοποιημένες αεροτομές (κόκκινο) για όλες τις μελέτες συγκριτικά με την αρχική γεωμετρία (μπλε).*

Μελέτη	Απώλειες Ολικής Πίεσης [Pa]			Ανηγμένη Παροχή [$\times 10^{-7}$]			Γωνία Εξόδου Ροής [rad]		
	Αρχική	Βελτ/νη	Δ[%]	Αρχική	Βελτ/νη	Δ[%]	Αρχική	Βελτ/νη	Δ[%]
A1	721.72	599.36	-17	5.280	5.285	-0.09	1.090	1.091	-0.09
A2	721.72	599.95	-17	5.280	5.285	-0.09	1.090	1.091	-0.09
A3	721.72	624.05	-13.6	5.280	5.282	-0.04	1.090	1.091	-0.09
A4	721.72	626.54	-13.4	5.280	5.282	-0.04	1.090	1.091	-0.09
A5	721.72	675.25	-6.5	5.280	5.275	-0.09	1.090	1.091	-0.09

Πίνακας 6: *Σύγκριση των αποτελεσμάτων της αεροδυναμικής ΒΜ. Το Δ[%] συμβολίζει την ποσοστιαία απόκλιση από την αρχική γεωμετρία.*

Συμπεράσματα και Προτάσεις για Μελλοντική Μελέτη

Τα κύρια συμπεράσματα που προκύπτουν από την παρούσα διπλωματική εργασία είναι τα εξής:

- Η προτεινόμενη μεθοδολογία παραμετροποίησης παρέχει μια ομαλή αναπαράσταση της γεωμετρίας της αεροτομής, διατηρώντας ταυτόχρονα σαφή σύνδεση μεταξύ των 26 μεταβλητών σχεδιασμού και των φυσικών μεγεθών, όπως η καμπυλότητα, η χορδή, το πάχος και οι γωνίες μετάλλου. Επιπλέον, είναι διαθέσιμες ρητές αναλυτικές παραγώγοι των σημείων ελέγχου ως προς τις μεταβλητές σχεδιασμού, επιτρέποντας την αποτελεσματική εφαρμογή αιτιοκρατικών μεθόδων βελτιστοποίησης.

- Η μοναδικότητά της προκύπτει από τη διακριτική σχεδιαστική προσέγγιση, κατά την οποία η αεροτομή χωρίζεται σε πέντε διακριτά τμήματα: τη μέση γραμμή καμπυλότητας, τις πλευρές υποπίεσης και υπερπίεσης, καθώς και τις περιοχές γύρω από τα σημεία πρόσπτωσης και εκφυγής.

- Σε σύγκριση με άλλες μεθόδους παραμετροποίησης, επιτρέπει την άμεση επιβολή γεωμετρικών περιορισμών χωρίς την ανάγκη επίλυσης πρόσθετων μερικών διαφορικών εξισώσεων, με αποτέλεσμα τη μείωση της πολυπλοκότητας και του υπολογιστικού κόστους. Αυτή η δυνατότητα είναι ιδιαίτερα σημαντική για τα πτερύγια στροβίλων, στα οποία ορισμένες περιοχές πρέπει να παραμείνουν αμετάβλητες κατά τη διάρκεια της βελτιστοποίησης λόγω απαιτήσεων ψύξης, δομικής ακεραιότητας ή άλλων φυσικών περιορισμών.

- Μετά τη βελτιστοποίηση σχήματος, η προκύπτουσα αεροτομή μπορεί να γίνει εύκολα κατανοητή, παρέχοντας μια διαισθητική κατανόηση των αλλαγών στα φυσικά μεγέθη. Με αυτόν τον τρόπο δίνεται στον μηχανικό η κατεύθυνση για μελλοντικά σχέδια ή μελέτες βελτιστοποίησης.

Μια φυσική συνέχεια της εργασίας σχετικά με την προτεινόμενη μέθοδο παραμετροποίησης θα μπορούσε να περιλαμβάνει τις ακόλουθες κατευθύνσεις:

- Εφαρμογή της προσαρμοσμένης παραμετροποίησης σε τριδιάστατα προβλήματα: Η επέκταση της μεθόδου σε τρισδιάστατες γεωμετρίες θα μπορούσε να ανοίξει νέες δυνατότητες στο αεροδυναμικό σχεδιασμό, ιδιαίτερα σε περιπτώσεις όπου πρέπει να τηρούνται πολύπλοκοι γεωμετρικοί περιορισμοί (π.χ. διατήρηση της κατανομής πάχους σε μία πτέρυγα ή τμήμα αεροτομής για τη δομική ακεραιότητα).
- Μελέτη προβλημάτων αντίστροφου αεροδυναμικού σχεδιασμού χρησιμοποιώντας την προτεινόμενη παραμετροποίηση: Η εφαρμογή αυτής της παραμετροποίησης σε προβλήματα αντίστροφου αεροδυναμικού σχεδιασμού, σε συνδυασμό με συγκριτική ανάλυση έναντι άλλων μεθόδων παραμετροποίησης, θα μπορούσε να την καθιερώσει ως ένα αξιόπιστο και αποτελεσματικό εργαλείο για τον αντίστροφο αεροδυναμικό σχεδιασμό.
- Συνδυασμός της προτεινόμενης παραμετροποίησης με άλλες μεθόδους: Η χρήση εναλλακτικών τεχνικών παραμετροποίησης κατά τους αρχικούς κύκλους βελτιστοποίησης, ακολουθούμενη από την προτεινόμενη μέθοδο για λεπτομερή ρύθμιση, θα μπορούσε να επιταχύνει τη διαδικασία βελτιστοποίησης, διασφαλίζοντας ταυτόχρονα ακόμη υψηλότερης ποιότητας λύσεις.