



National Technical University of Athens

School of Mechanical Engineering

Lab. of Thermal Turbomachines

Parallel CFD & Optimization Unit

Constrained Shape Optimization of a Compressor Stator using Evolutionary Algorithms and Continuous Adjoint

Diploma Thesis

Alexios Tsachalidis

Supervisor:

Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2019

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my thesis advisor Professor, K. Giannakoglou of the School of Mechanical Engineering at National Technical University of Athens, who has been guiding me all the years of my studies. I would like to genuinely thank him for the willingness to help and guide me throughout the completion of this diploma thesis, for his precious advice, both technical and practical. For all this generosity, I consider myself lucky to have been taught by him.

I would also like to deeply thank Dr. Konstantinos Tsiakas, who dedicated many hours of his time, answering to my questions. His counsels and continuous support to my efforts were very important in successfully completing my diploma thesis.

I would like to genuinely thank Markella Zorba for her willingness to help, answer any questions I had and for the useful comments and remarks throughout the consummation of my thesis.

I would also like to thank Dr. Xenofon Trompoukis and Dr. Varvara Asouti who both guided me and answered my questions immediately, whenever I asked them.

The whole research team of the Parallel CFD & Optimization Unit of National Technical University of Athens is always eager to help when needed.

Finally, I must express my very profound gratitude to my parents and brother for providing me with continuous support and encouragement throughout my years of study and through the process of writing this diploma thesis. Thank you.

Author

Alexios Tsachalidis



National Technical University of Athens

School of Mechanical Engineering

Lab. of Thermal Turbomachines

Parallel CFD & Optimization Unit

Constrained Shape Optimization of a Compressor Stator using Evolutionary Algorithms and Continuous Adjoint

Diploma Thesis

Alexios Tsachalidis

Supervisor: Kyriakos C. Giannakoglou, Professor NTUA

In this diploma thesis, turbomachinery blade shape aerodynamic optimization methods are applied in the open case of TU Berlin's compressor stator. The purpose of this diploma thesis is to validate results computed by the PUMA CFD flow solver against experimental measurements performed at the TU Berlin and present solutions of the shape optimization performed by the Evolutionary Algorithm software EASY and the Continuous Adjoint Method. The optimization process is based on software entirely developed by PCOpt/NTUA, such as the CFD solver PUMA which runs on GPUs, the parameterization software GMTurbo which provides a complete representation of the blade, the Evolutionary Algorithm optimization software EASY, the Continuous Adjoint solver, the Reverse Parameterization Tool (RPT) that converts from a node-parameterized to an equivalent GMTurbo-parameterized blade and the surface Grid Adaptation Tool (GAT) which adjusts the initial surface CFD grid to the GMTurbo geometry, converting the GMTurbo-based blade representation to a CFD grid to perform CFD simulations. Next to them, codes for the purpose of handling the case's manufacturing constraints and the post processing are programmed.

At first, the Metamodel-Assisted Evolutionary Algorithm workflow is structured in order to optimize the shape of the blades of TU Berlin compressor stator. The optimization process is confined by manufacturing constraints. Shape optimization solutions for both the unconstrained and constrained cases are presented. The optimization provides satisfying improvements to the objective function values of the case by taking the computational budget allocated to this diploma thesis into consideration.

Then, the continuous adjoint method to the PUMA CFD software is used to provide the sensitivity map that shows the way to change the blade shape and to optimize the shape of the blades of TU Berlin compressor stator with significantly less computational cost compared to Evolutionary Algorithms.



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Μηχανολόγων Μηχανικών

Εργαστήριο Θερμικών Στροβιλομηχανών

Μονάδα Παράλληλης ΥΡΔ & Βελτιστοποίησης

Βελτιστοποίηση Μορφής της Σταθερής Πτερύγωσης Συμπιεστή με Περιορισμούς με χρήση Εξελικτικών Αλγορίθμων και τη Συνεχή Συζυγή Μέθοδο

Διπλωματική Εργασία

Αλέξιος Τσαχαλίδης

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Η διπλωματική αυτή εργασία αφορά την εφαρμογή μεθόδων αεροδυναμικής βελτιστοποίησης μορφής πτερυγίων στη σταθερή πτερύγωση συμπιεστή του Πολυτεχνείου του Βερολίνου. Σκοπός της εργασίας είναι η σύγκριση και πιστοποίηση των αποτελεσμάτων του επιλύτη ροής PUMA με τα πειραματικά δεδομένα που είναι διαθέσιμα από το Πολυτεχνείο του Βερολίνου και η παρουσίαση των αποτελεσμάτων βελτιστοποίησης μορφής που πραγματοποιήθηκε με χρήση Εξελικτικών Αλγορίθμων και συγκεκριμένα του λογισμικού EASY, αλλά και με τη Συνεχή Συζυγή μέθοδο. Για την πραγματοποίηση των παραπάνω, η διαδικασία βελτιστοποίησης βασίστηκε σε λογισμικό που αναπτύχθηκε πλήρως στη ΜΠΥΡ&Β/ΕΜΠ, όπως ο επιλύτης ροής PUMA που τρέχει σε Κάρτες Γραφικών, το λογισμικό GMTurbo που παρέχει δυνατότητα παραμετροποίησης πτερυγίου με χρήση περιορισμένου αριθμού μεταβλητών σχεδιασμού, το λογισμικό βελτιστοποίησης με χρήση Εξελικτικών Αλγορίθμων EASY, ο επιλύτης της Συνεχούς Συζυγούς μεθόδου, το λογισμικό αντίστροφης παραμετροποίησης RPT που μετατρέπει το πλέγμα σε GMTurbo μορφή και το λογισμικό προσαρμογής επιφανειακού πλέγματος GAT το οποίο δημιουργεί πλέγμα γύρω από κάθε νέα CAD γεωμετρία, με κατάλληλη παραμόρφωση του αρχικού, προκειμένου να πραγματοποιηθούν προλέξεις του πεδίου ροής. Η εργασία συμπεριλαμβάνει τον προγραμματισμό κωδίκων για την επιβολή των κατασκευαστικών περιορισμών του προβλήματος και την επεξεργασία των αποτελεσμάτων.

Η διαδικασία βελτιστοποίησης της μορφής των πτερυγίων της σταθερής πτερύγωσης εμπεριέχει απαιτητικούς κατασκευαστικούς περιορισμούς. Παρουσιάζονται αποτελέσματα της βελτιστοποίησης με Εξελικτικούς Αλγορίθμους, με και χωρίς να ληφθούν υπόψη οι περιορισμοί. Η διαδικασία βελτιώνει τις τιμές των συναρτήσεων στόχων λαμβάνοντας υπόψη και τους διαθέσιμους για τη διπλωματική εργασία υπολογιστικούς πόρους.

Τέλος, χρησιμοποιείται η συνεχής συζυγής μέθοδος για τη δημιουργία του χάρτη ευαισθησίας που παρέχει τον τρόπο που πρέπει να διαφοροποιηθεί η μορφή των πτερυγίων για την επίτευξη των στόχων του προβλήματος αλλά και για την βελτιστοποίηση της μορφής των πτερυγίων με σημαντικά λιγότερο υπολογιστικό κόστος σε σχέση με τους Εξελικτικούς Αλγορίθμους.

Acronyms

CFD	Computational Fluid Dynamics
NTUA	National Technical University of Athens
PCOpt	Parallel CFD & Optimization unit
TU	Technical University – Technische Universität
CAD	Computer-Aided Design
EA	Evolutionary Algorithm
MAEA	Metamodel-Assisted Evolutionary Algorithm
PUMA	Parallel Unstructured Multi-row Adjoint
EASY	Evolutionary Algorithms SYstem
GMTurbo	Geometry Modeler for Turbomachines
RPT	Reverse Parameterization Tool
GAT	Grid Adaptation Tool
LE	Leading Edge
TE	Trailing Edge
PS	Pressure Side
SS	Suction Side
VIGV	Variable Inlet Guide Vane
SST	Shear Stress Transport
FAE	Field Adjoint Equations
ABC	Adjoint Boundary Conditions

Contents

Contents	i
1 Introduction	1
1.1 Thermal Turbomachines	1
1.2 Parameterization and Optimization of Bladings	2
1.3 Stochastic and Deterministic Optimization Methods	4
1.4 Constrained and Unconstrained Optimization	5
1.5 Outline of this Diploma Thesis	6
2 The TU Berlin's Case	7
3 Governing Equations & CFD Solver	14
3.1 Governing Equations	14
3.2 The GPU-enabled CFD Solver PUMA	20
4 Blading Parameterization – GMTurbo, RPT & GAT	22
4.1 GMTurbo - Turbomachinery Blade Parameterization	22
4.2 Reverse Parameterization Tool (RPT)	29
4.3 Grid Adaptation Tool (GAT)	39
5 Optimization Methods	45
5.1 Evolutionary Algorithms – EASY Software	45
5.2 Gradient-Based Methods – Continuous Adjoint	49
6 Implementation of Constraints	51
6.1 Presentation of Manufacturing Constraints	51
6.2 Inclusion in the EA Optimization Process	53
7 CFD Analysis of TU Berlin's Case - Validation	57
7.1 Mach number	57
7.2 Outlet Whirl angle α	61
7.3 Outlet Pitch angle β	62
7.4 Outlet Total Pressure and Static Pressure	64
7.5 Total Pressure Loss Coefficient	66

8 Evolutionary Algorithm Optimization	68
8.1 Design Variables	68
8.2 Optimization using EASY	68
9 Unconstrained Gradient-Based Optimization	73
10 Overview and Conclusions.....	77
10.1 Overview	77
10.2 Proposals for Future Work	78
Conformal Mapping	79
Bibliography	82

Chapter 1

Introduction

1.1 Thermal Turbomachines

A thermal turbomachine is a device in which mechanical energy, in the form of shaft work, is transferred either to or from a continuously flowing fluid by the interaction with rotating blade rows. Interaction between the fluid and blades results in aerodynamic forces. A turbomachine produces change in total enthalpy of the fluid passing through it.

Thermal turbomachines are classified into two categories; compressors and turbines. From a different point of view, depending on the direction of the flow while entering, passing through and leaving the machine, there are:

- Axial turbomachines, where the flow direction is almost parallel to the shaft axis,
- Radial turbomachines, where the flow direction is almost vertical to the shaft axis,
- Mixed flow turbomachines, where the direction of the flow is neither purely axial nor purely radial.

Axial Compressor:

An axial compressor stage consists of a row of rotor blades followed by a row of stator blades and the working fluid traverses them without significant change in radius. The energy level of the fluid is increased by the action of the rotor blades, which exert torque on the fluid. A single stage axial compressor has a relatively low pressure ratio with a higher mass flow rate compared to a radial compressor. Flow enters axially and discharges almost axially. The blade passages diverge from inlet to exit and, hence, the flow decelerates. The compression of the fluid is done by first accelerating the fluid and then diffusing it to increase pressure. The fluid is accelerated by the row of rotating blades, the rotor, and diffused by the row of stationary blades, the stator. The diffusion in the stator converts the velocity increase gained in the rotor to increase in pressure.

A compressor usually consists of multiple stages. The air passes from one stage to the next with each stage raising the pressure. One additional row of fixed blades, the so-called inlet guide vanes, is frequently used at the compressor inlet to ensure that air enters the first-stage rotors at the desired angle. In addition to the stators, an additional diffuser at the exit of the compressor further diffuses the fluid and controls its velocity when entering the combustors.

A sketch of a multi-stage axial compressor is given in Fig. 1.1.

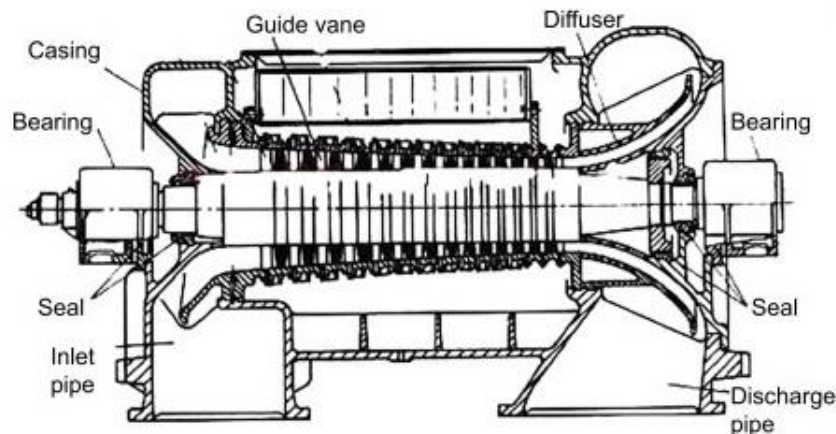


Figure 1.1: Sketch of an axial flow compressor, from [17].

1.2 Parameterization and Optimization of Bladings

Methods for the design and optimization of bladings, based on aerodynamic criteria, are a key research topic in turbomachines. In order to design and optimize the shape of a blading, geometrical representation is crucial. This is done by parameterization techniques. Parameterization is the process of composing a geometry according to an algorithm, by firstly determining a set of design variables that correspond to the input to this algorithm. Different sets of parameters produce different shapes. In order to parameterize the blading, CAD based methods are frequently used and this is the case in this diploma thesis, too. A blade is described with one or more NURBS surfaces controlled by their control points. Blade shapes can be modified by displacing the control points. The number of design variables is reduced in comparison with other methods, while the use of parametric geometry guarantees a smooth result. The method for parameterizing blade shapes used in this thesis is based on the parameterization software GMTurbo as described in detail in Section 4.1. In this thesis, the blading geometry is provided as a CFD grid, hence the Reverse Parameterization Tool (RPT), described in Section 4.2, is used to convert the CFD grid into a set of geometric parameters compatible with the GMTurbo input parameters. Such CAD-based methods have to be followed by a surface grid generator or morpher before generating the 3D CFD grid and running a CFD simulation. For this purpose, a surface Grid Adaptation Tool (GAT), described in Section 4.3, is used. The initial volume grid is deformed to fit the updated geometry using 2D and 3D spring analogy techniques. The method described before is included in an optimization loop as seen in Fig. 1.2. The optimization is carried out using stochastic methods such as Evolutionary Algorithms or gradient-based methods supported by the continuous adjoint method. These two main categories are presented in Section 1.3. The optimization methods used in this diploma thesis are described in detail in Sections 5.1 and 5.2.

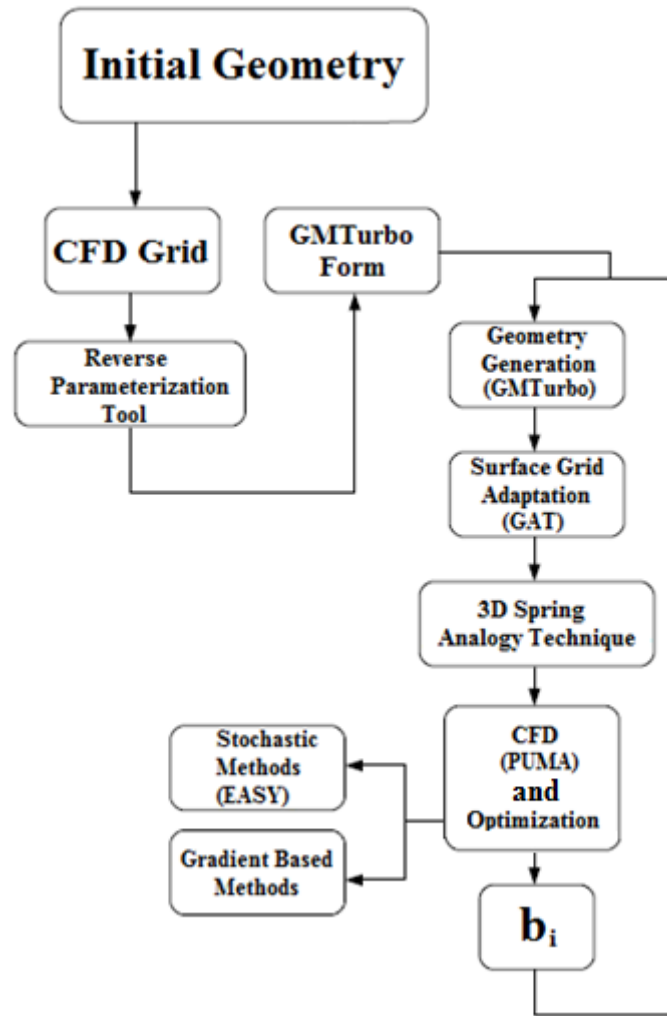


Figure 1.2: Shape optimization of a turbomachinery blade. Flowchart of the tasks that should be completed during the evaluation of a single individual, if stochastic optimization is performed, or an optimization cycle, if gradient-based methods are used. The initial geometry can be provided either in GMTurbo format or CFD grid format. In the common case in which the CFD grid is available, the pre-processing of the optimization includes the conversion of the geometry from grid format to the GMTurbo format.

1.3 Stochastic and Deterministic Optimization Methods

Optimization methods are classified into two major categories.

- Stochastic Optimization Methods

Stochastic optimization methods can be used in optimization problems of all kinds. Modern stochastic methods with basic representative the Evolutionary Algorithms choose randomized candidate solutions, by taking advantage of the evaluation of formerly computed solutions and the process of evolution, finally producing solutions that lead to convergence of the optimization process. Furthermore, the initial values of the variables in an EA do not affect the ability of finding the optimal solution. Hence, EAs are able to locate the global extremum of an objective function for every possible initialization. However, stochastic search methods require a large number of function evaluations. As a result, the computational cost of those methods scales with the number of design variables. For more information, the reader is referred to [11], [15].

- Deterministic or Gradient-Based Optimization Methods

The deterministic optimization methods use information on the direction of reduction or increase of the objective function by computing or approximating the first or the second order derivatives of the objective function with respect to the design variables. Solutions are updated until the convergence of the process to an extremum. Basic disadvantage of the deterministic methods is that the extremum in which the process will converge may be local, failing that way to spot the global minimum or maximum of the objective function. Contrary to the stochastic methods, the initialization affects the ability of finding the optimal solution. Furthermore, programming a method to efficiently compute the derivatives for the deterministic method is more time consuming and more difficult to reform for another objective function than the stochastic methods. However, gradient-based optimization can be used in order to have significantly fewer function calls, making this method less expensive than the stochastic methods. The computational cost of some of the gradient-based methods is independent to the number of design variables. This enables the optimization of complex geometries described parametrically using parametric analytic geometries such as Bézier surfaces. The most common deterministic methods are steepest descent, quasi-Newton and Newton methods. The derivatives of the objective function with respect to the design variables can be computed using various methods such as the continuous or discrete adjoint method. Further details are given in [12], [15].

The engineer is called to evaluate and decide on the optimization tools according to each case's characteristics.

1.4 Constrained and Unconstrained Optimization

Mathematically speaking, optimization is the minimization or maximization of a function subject to constraints. The following notation is used:

\vec{b} is the vector of design variables, also called unknowns or optimization parameters, F is the objective function that will be maximized or minimized and \vec{c} is the vector of constraints that solutions must satisfy. Both F and c are functions of \vec{b} . The number of components in \vec{c} is the number of constraints that have to be satisfied. The optimization problem can then be written as:

$$\min_{\vec{b} \in \mathbb{R}^n} F(\vec{b}) \quad \text{subject to} \quad \begin{cases} c_i(\vec{b}) = 0, & i \in \mathcal{E} \\ c_i(\vec{b}) \geq 0, & i \in I \end{cases} \quad (1.1)$$

where F and each c_i are scalar-valued functions of the variables \vec{b} and are given as equality and inequality constraints. I, \mathcal{E} are sets of indices.

Problems with the general form presented above can be classified according to the nature of the objective function and constraints, the number of variables, the smoothness of the functions in differentiable or non-differentiable and so on. Possibly the most important distinction is between problems that have constraints on variables and those that do not.

Unconstrained optimization problems arise directly in some applications. If there are natural constraints on the variables, it is sometimes safe to disregard them and to assume that they have no effect on the optimal solution. Unconstrained problems arise also as reformulations of constrained optimization problems, in which the constraints are replaced by penalization terms in the objective function to discourage constraint violations.

Constrained optimization problems may arise from models that include explicit constraints on the variables. These constraints may be simple bounds, more general linear constraints, or nonlinear inequalities that represent complex relationships among the variables. The TU Berlin stator case [16] studied in this diploma thesis possesses manufacturing constraints which are presented in Section 6.1. Results for the constrained and unconstrained optimization using Evolutionary Algorithms are presented in Section 8.2 and for the unconstrained gradient-based optimization in Chapter 9. More details on constrained and unconstrained optimization are given in [12].

1.5 Outline of this Diploma Thesis

The chapters of this diploma thesis are the following:

- Chapter 2: Presentation of the data and objectives of the TU Berlin stator case.
- Chapter 3: Presentation of the governing equations and of PUMA CFD solver.
- Chapter 4: Presentation of GMTurbo, RPT and GAT software used for blade parameterization and grid adaptation.
- Chapter 5: Presentation of the optimization methods used in this thesis. The Evolutionary Algorithm software EASY and the Continuous Adjoint method developed by the PCOpt/NTUA are presented.
- Chapter 6: Presentation of the methods for quantifying the manufacturing constraints of the problem and imposing them to the optimization process.
- Chapter 7: CFD Analysis of the TU Berlin stator case and validation of the CFD solver's results against experimental data.
- Chapter 8: Two shape optimizations are carried out using Evolutionary Algorithms with and without imposing manufacturing constraints.
- Chapter 9: The Continuous Adjoint method is used to provide the sensitivity map that shows potential changes in the shape of the blade and to optimize the blade shape without imposing manufacturing constraints.
- Chapter 10: Conclusions and a couple of proposals for future work.

Chapter 2

The TU Berlin's Case

The TurboLab Stator [16] is a stator in a measurement rig at the Technical University of Berlin, in the TurboLab at the Chair for Aero Engines. The initial stator geometry has been designed based on a representative stator geometry used in modern jet engine compressors. In this diploma thesis, the initial geometry, shown in Fig. 2.1 is optimized to reduce the total pressure losses and the flow angle deviation from the axial direction. Details of the CFD domain, flow boundary conditions, optimization requirements and manufacturing constraints are stated below.

CFD Domain:

Inner radius: 147.5 *mm*

Outer radius: 297.5 *mm*

Inlet axial position: -180 *mm*

Outlet axial position: 540 *mm*

The axial positioning of the blades is shown in Fig. 2.2. The zero position of the machine axis is defined by the LE as seen in Fig. 2.1.

Fluid Properties:

The working fluid is air, assumed as a perfect gas with the following properties:

- Ratio of specific heats $\gamma = 1.4$
- Gas constant $R = 287 \text{ J/kg} \cdot \text{K}$
- Dynamic viscosity $\mu = 1.785 \cdot 10^{-5} \text{ kg/m} \cdot \text{s}$

Optimization requirements:

Two optimization objectives are used, stated in points (1) and (2) below, leading to a two-objective optimization problem.

1) Minimization of the total pressure loss coefficient between the inlet to the CFD domain and its outlet, under the constraint of keeping the mass flow at $9.5 \pm 0.1 \text{ kg/s}$, full annulus. The total pressure loss coefficient is defined as:

$$\omega = \frac{p_{t1} - p_{t2}}{p_{t1} - p_1} \quad (2.1)$$

where p_{t1} is the inlet total pressure, p_{t2} is the outlet total pressure and p_1 is the inlet static pressure.

All p_* in Eq. 2.1 are mass-averaged values defined as:

$$p = \frac{\sum_A \dot{m}_i p_i}{\sum_A \dot{m}_i} \quad (2.2)$$

where \dot{m}_i is the mass flow that passes through the i^{th} element of the area A, which is the inlet or the outlet.

2) Minimization of the mass-averaged flow angle deviation from the axial direction at the CFD outlet. The mass-averaged exit angle is:

$$a_E = \sqrt{\frac{\sum_A \dot{m}_i a_i^2}{\sum_A \dot{m}_i}} \quad (2.3)$$

where a_i is the total flow angle at the i^{th} part of the CFD outlet defined as follows:

$$a = \cos^{-1} \left(\frac{\vec{v} \cdot \vec{i}}{|\vec{v}|} \right) = \cos^{-1} \left(\frac{w}{\sqrt{u^2 + v^2 + w^2}} \right) \quad (2.4)$$

where (u, v, w) are the three Cartesian velocity components.

The inlet whirl angle is given in radial distribution. The average value of inlet whirl angle is allowed to vary by $\pm 5^\circ$. Thus, three operating points are considered. The design point DP - OP1 with average value of 42° inlet whirl angle and two off - design points, OP2 with average value $42 + 5 = 47^\circ$ and OP3 with $42 - 5 = 37^\circ$.

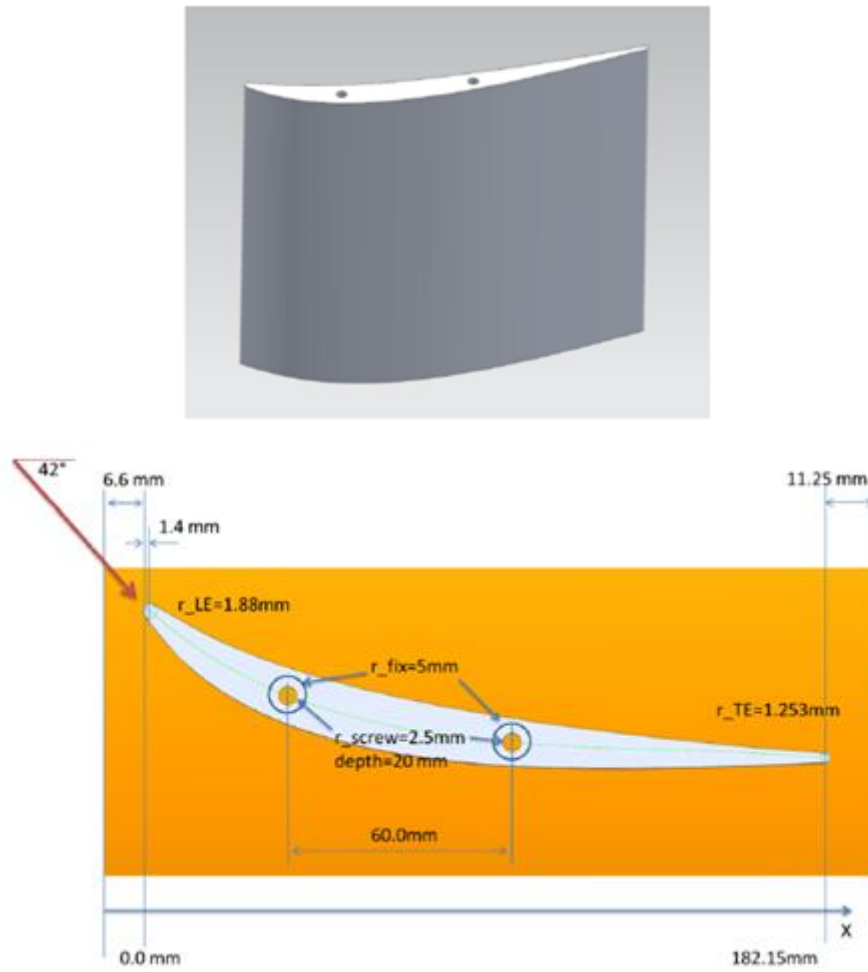


Figure 2.1: Initial blade geometry (top) and axial positioning (bottom), including the yellow domain for the end wall profiling and manufacturing constraints (bottom), from [16].

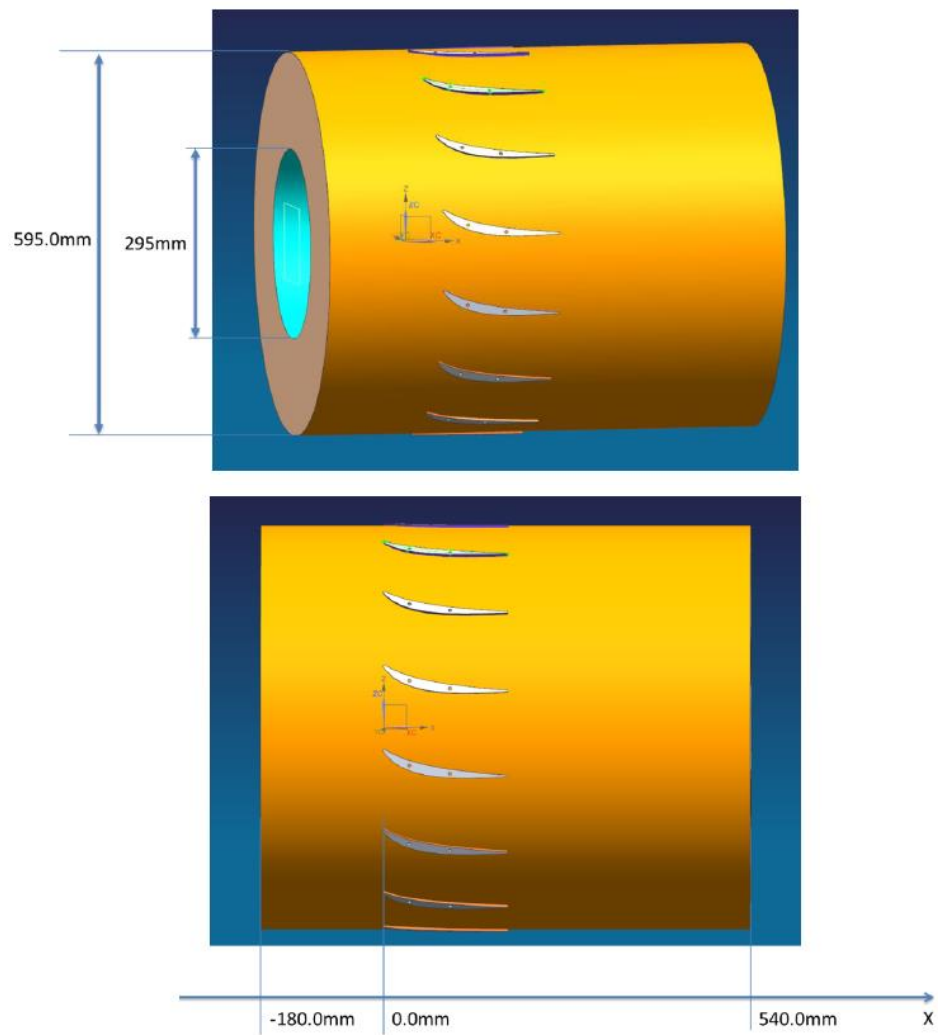


Figure 2.2: Axial positioning of the stator blades, from [16].

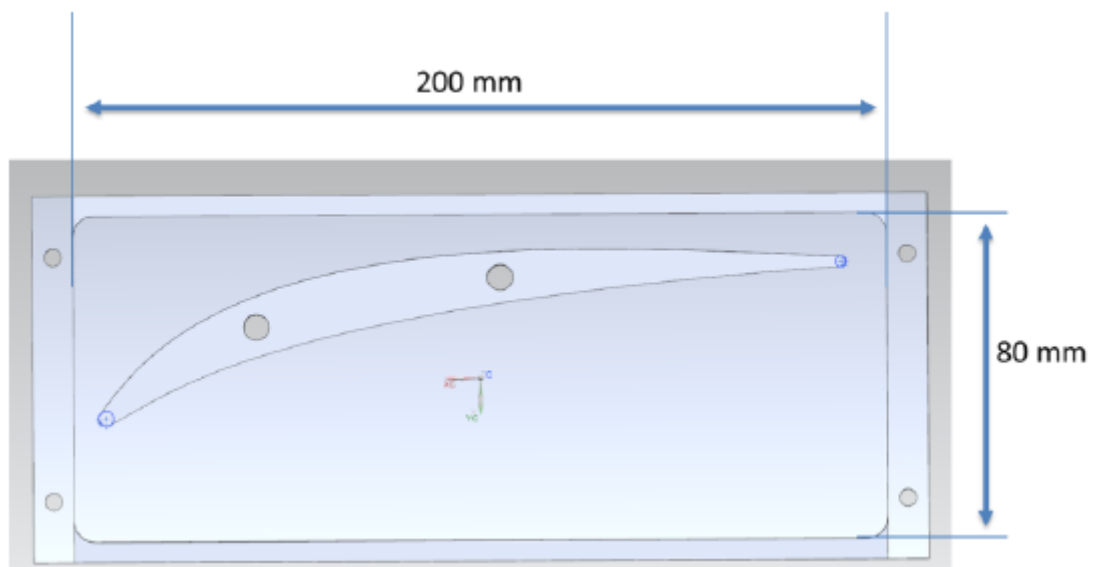


Figure 2.3: Manufacturing constraints for the casing fixture, from [16].

The assembly of the TU Berlin compressor stator is given in Fig 2.4.



Figure 2.4: Assembly of the TU Berlin stator, from [16].

As it can be seen in Fig 2.4, the hub and shroud are cylinders and their radii remain constant during the optimization.

Experimental Setup

The experimental setup of the TU Berlin stator is presented in Fig 2.5.

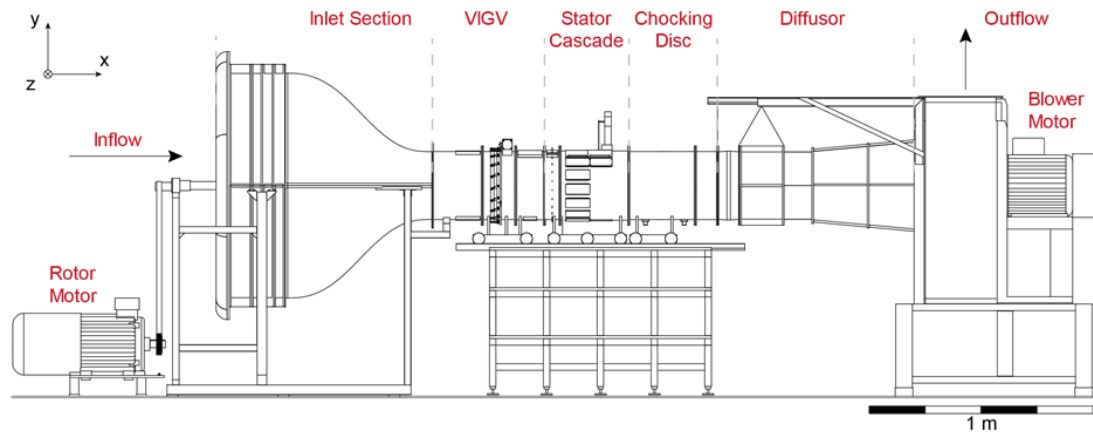


Figure 2.5: Experimental setup of the TU Berlin stator - 3D-annular low-speed compressor stator cascade, from [16].

The measurement section, seen in flow direction, is presented in Fig. 2.6.

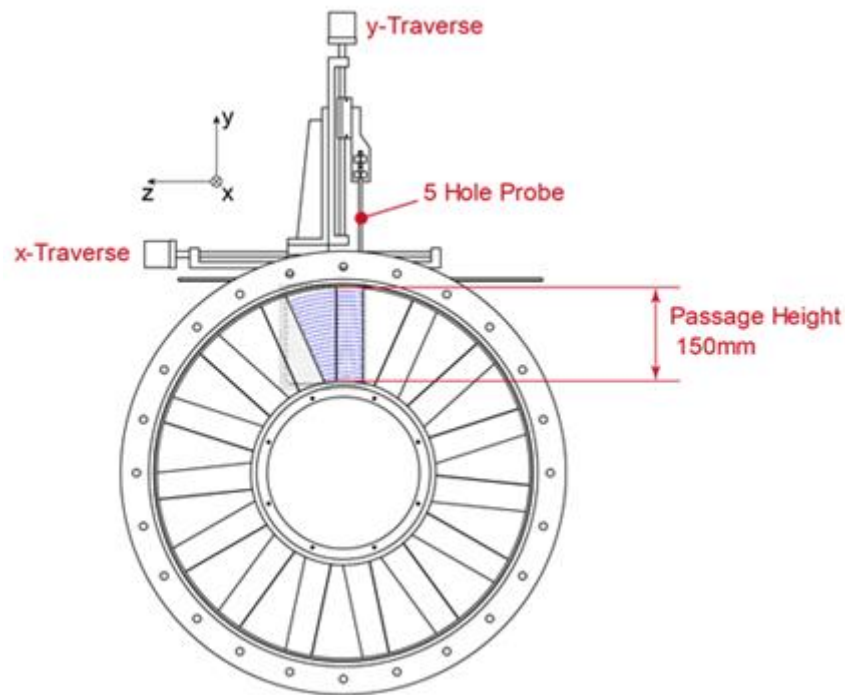


Figure 2.6: Measurement section, seen in flow direction with traversed grid, from [16].

The grid used in the measurement's procedure is given in Fig 2.7.

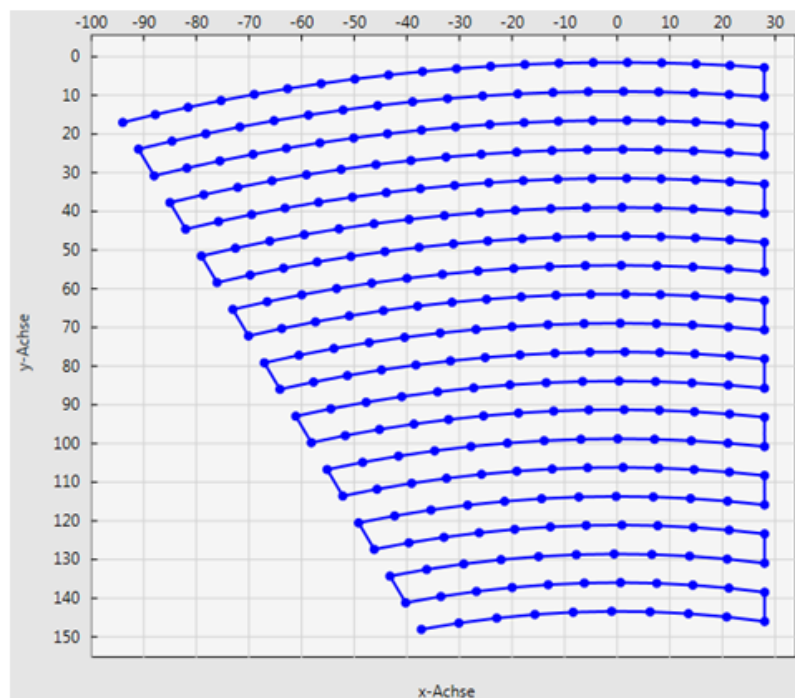


Figure 2.7: Measurement grid, from [16].

The same measurement grid, seen in Fig 2.7, was used for all inflow and wake measurements. For inflow measurements, the grid was mirrored at the y-axis.

The measurement position for both inflow and wake is given in Fig. 2.8.

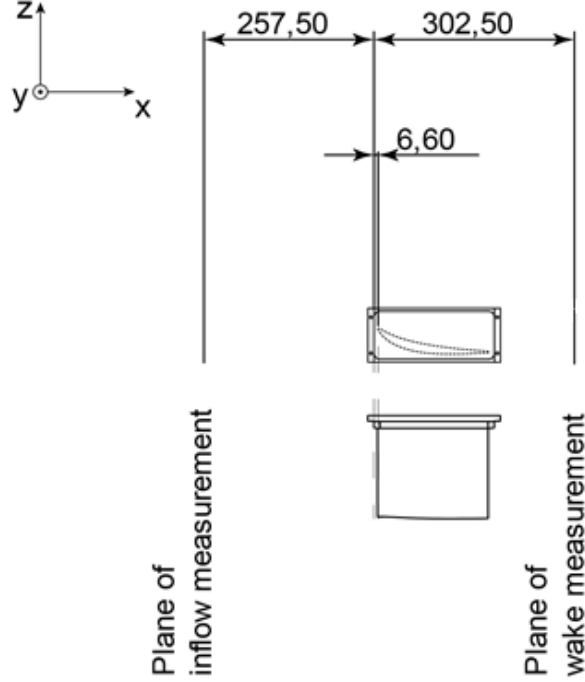


Figure 2.8: Measurement positions, from [16].

As seen in Fig. 2.8, the plane of inlet measurement is 264.1 mm upstream of the LE of the blade which corresponds to 140% of chord length upstream of the LE and the plane of outlet - wake measurement is in 295.9 mm downstream of the LE which corresponds to 60% of chord length behind the TE. The validation of the CFD solver against the measurements is made in the positions presented in Fig. 2.8.

The experimental data include values of total pressure P_t in Pa, static pressure P_{st} in Pa, absolute velocity magnitude V in m/s , Mach number, whirl angle α in degrees, pitch angle β in degrees and the three Cartesian velocity components in m/s both over the inlet and outlet measurement planes, for all three inlet whirl angles. The comparison of the CFD results with the experimental data is made using mass-averaged values plotted in the radial section. Hence, a radial grid, as seen in Fig. 2.7, is created in order to interpolate the CFD results in radial sections.

Chapter 3

Governing Equations & CFD Solver

In this chapter, the flow equations with their boundary conditions that are solved in the TU Berlin stator case of this thesis, using the CFD software PUMA developed by the PCOpt/NTUA [2], [3], [5], are presented.

3.1 Governing Equations

Compressible Flow Equations

Let a coordinate system $O(x_1, x_2, x_3)$ be defined. The Reynolds-Averaged Navier-Stokes equations for the viscous flow of a compressible fluid through the stator are expressed as:

$$\frac{\partial U_n}{\partial t} + \frac{\partial f_{nk}^{inv}}{\partial x_k} - \frac{\partial f_{nk}^{vis}}{\partial x_k} = 0 \quad (3.1)$$

where U_n stands for the conservative flow variables namely $U_n = [\rho \rho u_1 \rho u_2 \rho u_3 \rho E]$, with ρ being the fluid density, u_m ($m = 1, 2, 3$) being the Cartesian velocity components with reference to the absolute/inertial frame of reference and E the energy per unit mass.

The inviscid fluxes f_{nk}^{inv} and viscous fluxes f_{nk}^{vis} are defined as:

$$f_{nk}^{inv} = \begin{bmatrix} \rho u_k \\ \rho u_1 u_k + p \delta_{1k} \\ \rho u_2 u_k + p \delta_{2k} \\ \rho u_3 u_k + p \delta_{3k} \\ \rho h_t u_k \end{bmatrix}, \quad f_{nk}^{vis} = \begin{bmatrix} 0 \\ \tau_{1k} \\ \tau_{2k} \\ \tau_{3k} \\ u_l \tau_{lk} + q_k \end{bmatrix} \quad (3.2)$$

where p stands for the static pressure, δ_{ij} is the Kronecker symbol, h_t is the total enthalpy and q_k is the heat flux.

The stress tensor is given by:

$$\tau_{km} = \frac{\mu + \mu_t}{Re} \left(\frac{\partial u_k}{\partial x_m} + \frac{\partial u_m}{\partial x_k} - \frac{2}{3} \delta_{km} \frac{\partial u_l}{\partial x_l} \right) \quad (3.3)$$

where μ_t the turbulent viscosity computed by the turbulence model, as described later in this chapter and Re the Reynolds number resulting from the non-dimensionalization of the equations.

Low Mach Number Preconditioning

The software for solving Eq. 3.1 is reliable for flows with Mach numbers approximately higher than 0.3. In various cases, a wide range of velocity values appears in the flow field which results in high and low values of Mach number. For instance, during the analysis of the flow around a blade, areas where the flow can be considered as incompressible may appear while, in other areas, compressibility phenomena are important. Hence, the algorithms for solving compressible flows when Mach number is close to zero appear to face convergence difficulties by even damaging the accuracy of the solution. The difficulty of numerically solving the equations is connected with the major difference between the velocity of the fluid and the velocity of the acoustic waves.

Aiming at possessing a single CFD analysis software for all flow speeds, the low-Mach number preconditioning is applied to an existing software for the solution of the compressible flow equations. It ensures faster convergence for very low Mach numbers, allows the solution even in cases in which the non-preconditioned equations fail and provides smooth solution at singular areas such as in the vicinity of the leading and trailing edges. Hence, Eq. 3.1 is altered and given by:

$$\Gamma_{nm}^{-1} \frac{\partial U_m}{\partial t} + \frac{\partial f_{nk}^{inv}}{\partial x_k} - \frac{\partial f_{nk}^{vis}}{\partial x_k} = 0 \quad (3.4)$$

where Γ_{nm} is the preconditioning matrix, given in Eq. 3.5, used to stabilize the system of PDEs and lead to a robust numerical solution, by giving appropriate values to the parameters α' and β' .

$$\Gamma_{nm} = \begin{bmatrix} 1 + \frac{1}{2} |\vec{u}|^2 \kappa & -\kappa u_1 & -\kappa u_2 & -\kappa u_3 & -\kappa \\ \frac{1}{2} |\vec{u}|^2 \kappa u_1 & 1 - \kappa u_1^2 & -\kappa u_1 u_2 & -\kappa u_1 u_3 & -\kappa u_1 \\ \frac{1}{2} |\vec{u}|^2 \kappa u_2 & -\kappa u_1 u_2 & 1 - \kappa u_2^2 & -\kappa u_2 u_3 & -\kappa u_2 \\ \frac{1}{2} |\vec{u}|^2 \kappa u_3 & -\kappa u_1 u_3 & -\kappa u_2 u_3 & 1 - \kappa u_3^2 & -\kappa u_3 \\ -\frac{1}{2} |\vec{u}|^2 [1 - \delta] & [1 - \delta] u_1 & [1 - \delta] u_2 & [1 - \delta] u_3 & -\delta \end{bmatrix} \quad (3.5)$$

where $\kappa = (\gamma - 1)\beta'$, $\delta = \alpha' + \frac{1}{2} |\vec{u}|^2 (\gamma - 1)\beta'$, $\alpha' = \min(1, M^2)$, $\beta' = \frac{-(1-\alpha')}{c^2}$, M is the Mach number and c is the speed of sound.

For further details on this topic, the reader is referred to [8] and [11].

Incompressible Flow Equations

In incompressible flows, the flow speed is insignificant compared with the speed of sound of the fluid medium, $V \ll c$. This results in:

$$M = \frac{V}{c} \ll 1 \quad (3.6)$$

For low values of the Mach number, high velocity variations are required to produce a variation in the density, thus the flow tends to be incompressible. Practically, the incompressible behavior of the flow occurs for $M < 0.3$ which corresponds to a change in density lower than 10%. In fluid mechanics, incompressible flow refers to a flow in which the material density ρ is constant in an infinitesimal volume that moves with the flow velocity, called fluid parcel. Incompressible flow does not imply that the fluid itself is incompressible. Even compressible fluids at low speeds can, to a good approximation, be modeled by an incompressible flow model.

Additionally, the flow is assumed to be isothermal and the last of Eqs. 3.1, the energy equation, is identically satisfied. Hence, the terms of Eq. 3.2 become:

$$f_{nk}^{inv} = \begin{bmatrix} u_k \\ u_1 u_k + \frac{p}{\rho} \delta_{1k} \\ u_2 u_k + \frac{p}{\rho} \delta_{2k} \\ u_3 u_k + \frac{p}{\rho} \delta_{3k} \end{bmatrix}, \quad f_{nk}^{vis} = \begin{bmatrix} 0 \\ \tau_{1k} \\ \tau_{2k} \\ \tau_{3k} \end{bmatrix} \quad (3.7)$$

The unknown flow variables are: $U_n = \left[\frac{p}{\rho} u_1 u_2 u_3 \right]$ where p/ρ is the kinematic pressure, if density is assumed constant and u_m ($m = 1,2,3$) are the velocity components.

The stress tensor is now expressed as:

$$\tau_{km} = \frac{\nu + \nu_t}{Re} \left(\frac{\partial u_k}{\partial x_m} + \frac{\partial u_m}{\partial x_k} \right) \quad (3.8)$$

since the velocity divergence vanishes due to the continuity equation. ν is the kinematic viscosity and ν_t the eddy viscosity. For further details, the reader is referred to [2] and [8].

The k- ω SST Turbulence Model

In this diploma thesis, all applications are simulated using the k- ω SST [9], [10] turbulence model, implemented in the PUMA solver. The PUMA solver also supports other turbulence models such as the Spalart-Allmaras [19]. The k- ω SST turbulence model is a two-equation eddy-viscosity model. The eddy viscosity is given by:

$$\nu_t = \frac{\mu_t}{\rho} = \frac{a_1 k}{\max(a_1 \omega, S F_2)} \quad (3.9)$$

where k is the turbulent kinetic energy:

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_j k)}{\partial x_j} = \bar{P}_k - b^* \rho k \omega + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_j} \right] \quad (3.10)$$

and ω is the specific dissipation rate:

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho u_j \omega)}{\partial x_j} = \frac{a}{\nu_t} \bar{P}_k - b \rho \omega^2 + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_j} \right] + 2(1 - F_1) \frac{\rho \sigma_{\omega 2}}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \quad (3.11)$$

Auxiliary relations are given below:

$$F_2 = \tanh \left[\left[\max \left(\frac{2\sqrt{k}}{b^* \omega y}, \frac{500\nu}{y^2 \omega} \right) \right]^2 \right] \quad (3.12)$$

A production limiter is used in the SST model to prevent the build-up of turbulence in stagnation regions:

$$\bar{P}_k = \min(P_k, 10b^* \rho k \omega) \quad (3.13)$$

$$P_k = \mu_t \frac{\partial u_i}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (3.14)$$

$$F_1 = \tanh \left\{ \left\{ \min \left[\max \left(\frac{\sqrt{k}}{b^* \omega y}, \frac{500\nu}{y^2 \omega} \right), \frac{4\rho \sigma_{\omega 2} k}{CD_{k\omega} y^2} \right] \right\}^4 \right\} \quad (3.15)$$

F_1 is equal to zero away from the surface and switches over to one inside the boundary layer.

$$CD_{k\omega} = \max \left(2\rho \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}, 10^{-10} \right) \quad (3.16)$$

where y is the distance from the field point to the nearest wall and S is the invariant measure of the strain rate.

Each of the constants is a blend of an inner, 1, and outer, 2, constant using:

$$\varphi = \varphi_1 F_1 + \varphi_2 (1 - F_1) \quad (3.17)$$

Coefficients are given bellow:

$$\begin{aligned} a_1 = \frac{5}{9}, \quad a_2 = 0.44, \quad b_1 = \frac{3}{40}, \quad b_2 = 0.0828, \quad b^* = \frac{9}{100} \\ \sigma_{k1} = 0.85, \quad \sigma_{k2} = 1, \quad \sigma_{\omega1} = 0.5, \quad \sigma_{\omega2} = 0.856 \end{aligned} \quad (3.18)$$

Boundary Conditions

In order to solve the system of PDEs of Eqs. 3.1, 3.11 and 3.12, appropriate boundary conditions should be defined and implemented. The GPU solver is equipped with a wide range of boundary condition options. In this section, the boundary conditions used in the numerical prediction of flows through stator blades are presented.

Along pairs of periodic boundaries, appropriate periodic conditions are imposed. In case of peripheral rows, two points are periodically paired if their projections on the meridional plane coincide and their circumferential position differs by the blade row pitch. Between paired points all scalar quantities are the same, while every vector and tensor quantity is rotated by the row pitch. Wall, inlet and outlet boundary conditions are analyzed separately as follows.

Wall Boundary Conditions:

The velocity is set equal to the wall velocity which is zero for a stator:

$$u_k = 0, k = 1, 2, 3 \quad (3.20)$$

The boundary wall conditions used for the turbulence model [9], [10] are given by:

$$\omega_{wall} = 10 \frac{6\nu}{b_1 y_w^2} \quad (3.21)$$

$$k_{wall} = 0 \quad (3.22)$$

where y_w is the distance between the wall and the first node off the wall.

Inlet Boundary Conditions:

At the inlet of the domain, total pressure P_t , total temperature T_t and inlet velocity direction are specified as radial distributions. The inlet velocity direction is given in terms of angles α and β in a cylindrical coordinate system. The inlet velocity components are written in the form:

$$u_1 = |V| \sin \alpha$$

$$u_2 = |V| \cos \alpha \sin \beta \quad (3.23)$$

$$u_3 = |V| \cos \alpha \cos \beta$$

Thus, angles α , β and the velocity magnitude V are specified by the user. u_1 is the radial velocity u_r , u_2 is the peripheral u_θ and u_3 the axial u_z . In the PUMA solver, the flow equations are solved with reference to the Cartesian coordinate system. Hence, the velocity components in cylindrical coordinates (r, θ, z) of Eq. 3.23 are transformed to Cartesian. Specifically, Finally, turbulence intensity $I = u'/U$ and viscosity ratio ν_t/ν should also be defined. In turbulence intensity I , u' is the root-mean-square of the velocity fluctuations given as $u' = \sqrt{\frac{1}{3}(u_x'^2 + u_y'^2 + u_z'^2)}$ and V is the mean velocity computed by the three mean velocity components as $U = \sqrt{u_x^2 + u_y^2 + u_z^2}$.

The inlet boundary conditions, used when solving the incompressible flow equations, distributed over the radial height are:

- Velocity magnitude V , in m/s , from the measurement data
- Whirl angle α , in degrees, from the measurement data
- Pitch angle β , in degrees, from the measurement data
- Turbulence intensity $I = u'/U = 0.05$ constant
- Viscosity ratio $\nu_t/\nu = 20$ constant as explained in Section 7.1

Angles α and β are defined as seen in Fig. 3.1.

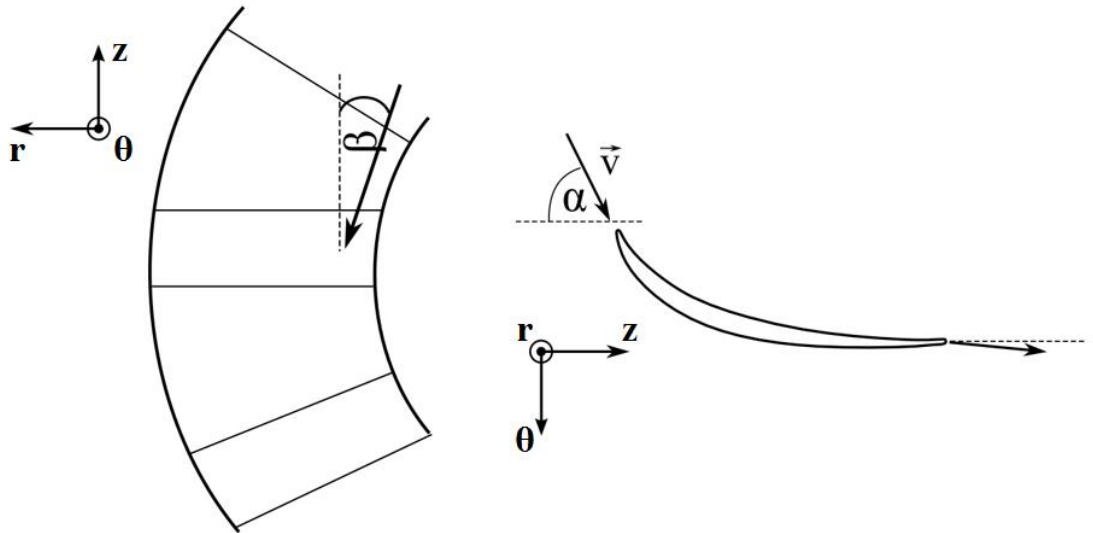


Figure 3.1: Definition of angles α and β .

The inlet boundary conditions of the turbulence model for both compressible and incompressible flow are:

$$k_{in} = \frac{3}{2} (Iu_{in})^2 \quad (3.24)$$

$$\omega_{in} = \frac{k_{in}^{1/2}}{c_\mu L} \quad (3.25)$$

where L is the chord length and $c_\mu = 0.09$ is a constant.

Outlet Boundary Conditions:

At the outlet, one quantity at each node must be specified and the remaining ones are extrapolated from the interior. The specified quantity is mass flow which is equal to 9.5 kg/s , full annulus. By taking the number of blades, $n = 15$, into consideration the outlet condition of mass flow of the CFD solver per passage is: $9.5/15 = 0.6333 \text{ kg/s}$. From the mass flow, the velocity of the fluid is calculated and set at each node.

3.2 The GPU-enabled CFD Solver PUMA

In order to predict the flow field in a turbomachinery blade passage, the compressible GPU enabled flow solver PUMA, developed by the PCOpt/NTUA is used [2]. This software numerically solves the Navier-Stokes equations along with the turbulence model equations in a computational domain, using the vertex-centered, finite volume method on unstructured grids consisting of tetrahedra, pyramids, prisms and hexahedra. Structured or matching block structured grids like the ones presented in this diploma thesis, are treated by PUMA as unstructured.

In the system of PDEs formed by Eqs. 3.1, 3.11 and 3.12, the hyperbolic character of the terms dominates over the elliptic. Consequently, these equations are solved using a time-marching technique. Hence, the pseudo-time step τ is defined. At each pseudo-time step the system of equations is linearized, discretized and solved for the correction of the field variables for the next pseudo-time step. In case of time-dependent problems, the time derivative remains in the initial equations and the pseudo-time derivative is artificially added. The system of equations is linearized and then marched in time, with intermediate pseudo-time steps. In case of the incompressible flow equations, time marching techniques are not directly applicable. Consequently, methods such as the artificial compressibility method can be used.

In order to solve the system of Eqs. 3.1, 3.11 and 3.12, discretization is applied. A finite volume is formed around each mesh node at real-time step by connecting the edge midpoints, face centers and element barycenters of the edges, faces and elements attached to this node, respectively. The discretization of the inviscid terms is made using Roe's approximate Riemann solver [18] while the viscous terms are discretized

using a 2nd order central difference scheme involving all neighbours. After discretizing all terms of Eqs. 3.1, 3.11 and 3.12, a new system of equations is created which is solved using the explicit multi-stage Runge-Kutta method with residual smoothing using the point-implicit Jacobi method which does not require synchronizations in each iteration when the solution of the system is parallelized on GPUs.

The implementation of PUMA on GPUs provides a remarkable speed-up in comparison with CPU implemented software, reducing the turnaround time of a CFD evaluation. Specifically, the implementation on GPUs is approximately 40 times faster than the CPUs, comparing one card against one core. The GPUs that were used are NVIDIA Tesla K40. For further details on the topic, the reader is referred to [2].

Chapter 4

Blading Parameterization – GMTurbo, RPT & GAT

4.1 GMTurbo - Turbomachinery Blade Parameterization

In order to design and optimize the shape of blading, accurate geometrical representation is required and it is done by parameterization. Parameterization is the process of composing geometry according to an algorithm, by firstly determining a set of design variables that correspond to the input to this algorithm. The blade parameterization this thesis is based upon is an intuitive method that exploits fundamental notions of turbomachinery to represent a blade. This method is developed and used by the PCOpt/NTUA in the GMTurbo parameterization software [2] and is presented in detail in this section. This software takes the necessary input values and returns the NURBS surface representations of the two sides of a 3D blade, PS and SS, in a neutral CAD format such as IGES, as seen in Fig. 4.1.



Figure 4.1: The GMTurbo generates the CAD compatible geometry of the blade.

The implementation of the geometric shapes needed throughout the parameterization is carried out using parametric NURBS curves and surfaces.

The first step of the parameterization procedure is to create the meridional contour of the turbomachine. For a blading that revolves around the z -axis, the meridional contour is an (r, z) projection of the axisymmetric parts of the blade, namely, the inlet and outlet planes, the hub and shroud and the LE and TE trace as the turbomachine revolves. These parts, being axisymmetric, exhibit symmetry around the z -axis, thus can fully be represented on the (r, z) plane and, then, by revolution by a certain angle θ around the z -axis, converted into (x, y, z) coordinates through:

$$(x, y, z) = (r \cos \theta, r \sin \theta, z) \quad (4.1)$$

Therefore, the first step is the definition of six meridional (r, z) NURBS curves. Namely, two meridional boundary curves, inlet and outlet, two meridional generatrices, hub and shroud and two meridional edge curves, LE and TE as shown in Fig. 4.2.

To make it possible to superpose information about the metal angles and thickness profiles, at different spanwise positions of the blade, projected streamlines are computed as a linear interpolation between the hub and shroud generatrices, defined in the previous step. It is noted that the word “streamwise” refers to a distribution of data from inlet to outlet, while spanwise refers to a distribution from hub to shroud. Therefore, the blade can be seen as a combination of streamwise and spanwise distributions of data. The blade meridional contour is given by:

$$h(u, v) = (r(u, v), z(u, v)) \quad (4.2)$$

where shroud, hub, inlet, outlet, LE and TE are given by $h(u, 0)$, $h(u, 1)$, $h(0, v)$, $h(1, v)$, $h(u_{LE}, v)$ and $h(u_{TE}, v)$, respectively.

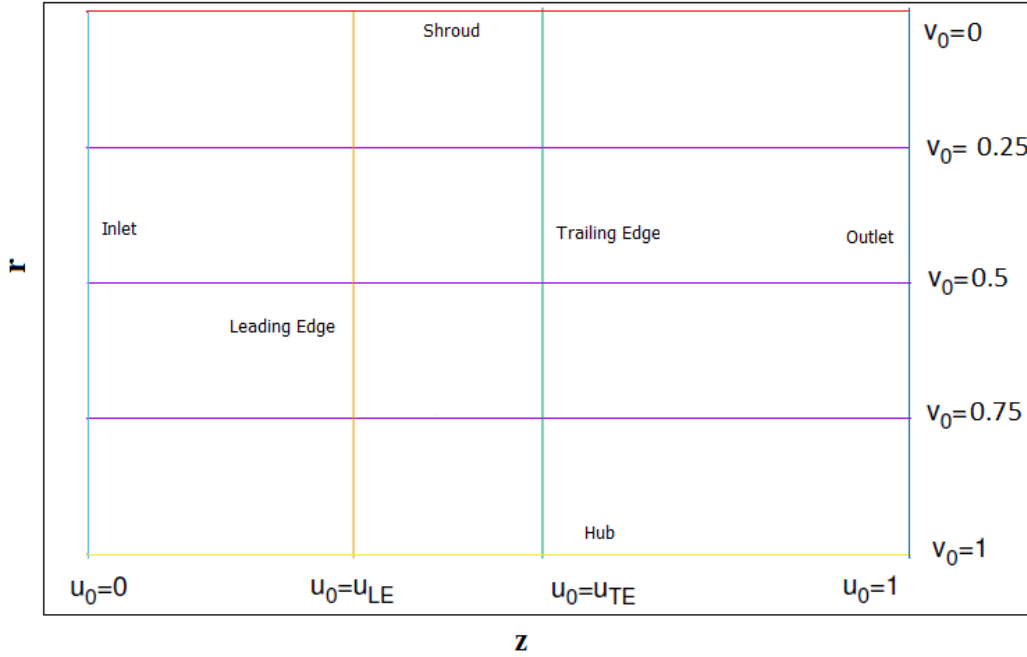


Figure 4.2: Meridional contour of the TU Berlin stator blade. The streamline projections - purple lines lie between hub and shroud.

Each projected streamline $h(u, v_0) = (r(u, v_0), z(u, v_0))$ corresponds to a revolved surface as shown in Fig. 4.3 by adding the angle θ using Eq. 4.1.

Conformal Mapping:

Every surface can be described by two parameters, using a transformation from 3D coordinates to a 2D plane. Hereupon, the conformal mapping is introduced. It is known that a 3D surface of revolution can be mapped onto the 2D (m, θ) plane through the transformation of:

$$\Phi(v_0) : (r(u, v_0)\cos\theta, r(u, v_0)\sin\theta, z(u, v_0)) \rightarrow (m(u, v_0), \theta) \quad (4.3)$$

where $m(u, v_0)$ is given by:

$$m(u, v_0) = \int_0^u \frac{\sqrt{r_u(t, v_0)^2 + r_v(t, v_0)^2}}{r(t, v_0)} dt \quad (4.4)$$

The mapping of Eq. 4.3 is conformal. For more information see Appendix A. The most important property of any conformal mapping is the angle preservation property. Conformal mappings preserve the magnitude and direction of the angle between two curves [4]. This property contributes to a better understanding of the parameterization, considering that the angles defined on the (m, θ) plane, are preserved on the (x, y, z) surface as well.

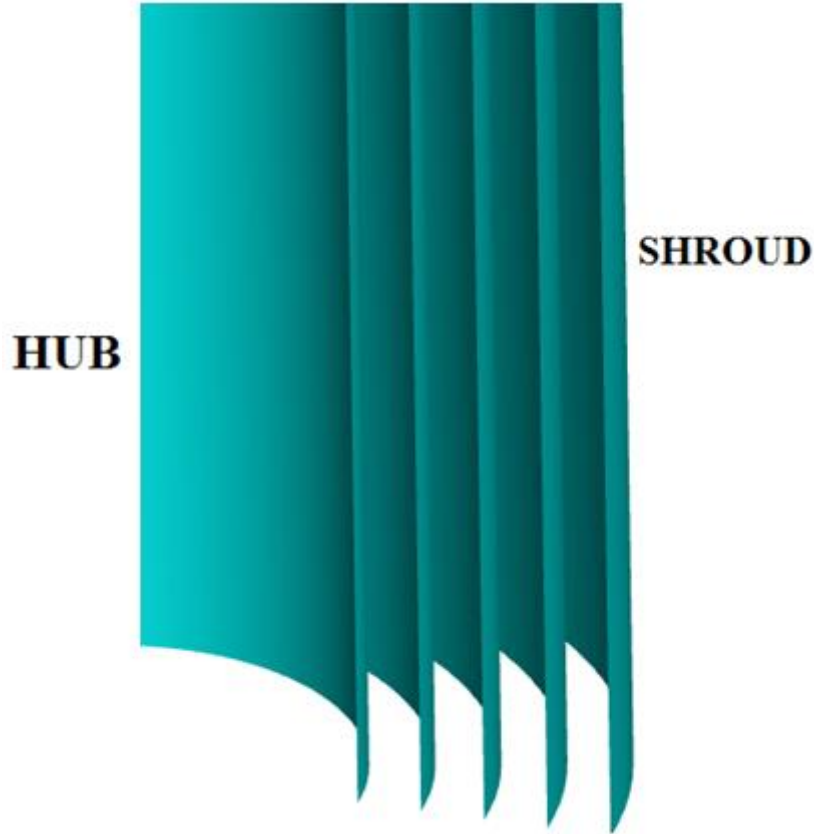


Figure 4.3: Surfaces of revolution of the TU Berlin stator that result from the (r, z) streamlines of Fig. 4.2.

Mean Camber Line Parameterization

At each spanwise position $v_0 \in [0, 1]$, a mean camber line is defined, to add information about the blade's metal angles. The mean camber line is chosen to be represented by a cubic Bézier curve on the (m, θ) plane. The four control points of the cubic Bézier curve are defined by six parameters that correspond to different angles at the LE and TE of the mean camber line, providing the designer with a better understanding of the mean camber line's slope along the arc length. These are:

- θ_{LE} and θ_{TE} are the peripheral positions of the LE and TE, respectively.
- β_{LE} and β_{TE} are the metal angles. These are the angles between the tangent to the mean camber line at the LE or TE and the meridional direction.
- δ_{LE} and δ_{TE} are the angles starting from the chord midpoint and intersection of the tangent lines in LE and TE respectively.

According to Fig. 4.4, the control points P_0, P_1, P_2 and P_3 are functions of $\theta_{LE}, \theta_{TE}, \beta_{LE}, \beta_{TE}, \delta_{LE}$ and δ_{TE} given by:

$$P_0 = (m_{LE}, \theta_{LE}) \quad (4.5)$$

$$P_3 = (m_{TE}, \theta_{TE}) \quad (4.6)$$

where $m_{LE} = m(u_{LE}, v_0)$ and $m_{TE} = m(u_{TE}, v_0)$ from Eq. 3.4.

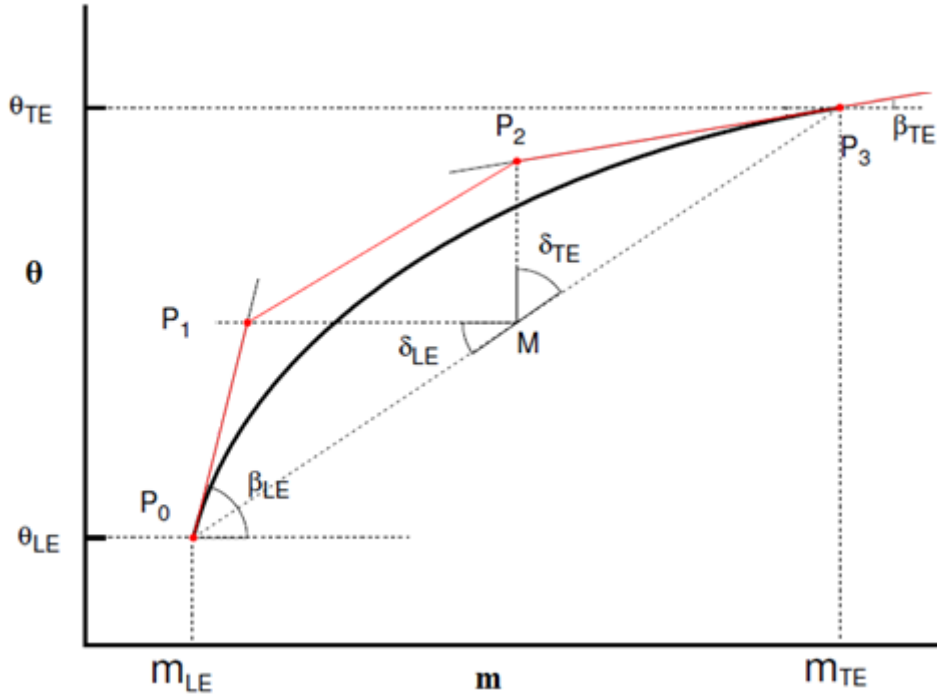


Figure 4.4: Definition of control points of a cubic Bézier mean camber line, from angles $\theta_{LE}, \theta_{TE}, \beta_{LE}, \beta_{TE}, \delta_{LE}$ and δ_{TE} .

Using the δ_{LE} and δ_{TE} angles, control points P_1 and P_2 are given as solution to the systems of the following two equations:

$$(P_1 - M) \cdot (P_0 - M) = \cos \delta_{LE} \|P_1 - M\| \cdot \|P_0 - M\| \quad (4.7)$$

$$(P_2 - M) \cdot (P_3 - M) = \cos \delta_{TE} \|P_2 - M\| \cdot \|P_3 - M\| \quad (4.8)$$

where M is the midpoint of the $P_0 P_3$ chord.

Through this cubic Bézier parameterization, spanwise distributions of θ_{LE} , θ_{TE} , β_{LE} , β_{TE} , δ_{LE} and δ_{TE} are defined, producing the mean camber line for each spanwise position as in Fig. 4.5.

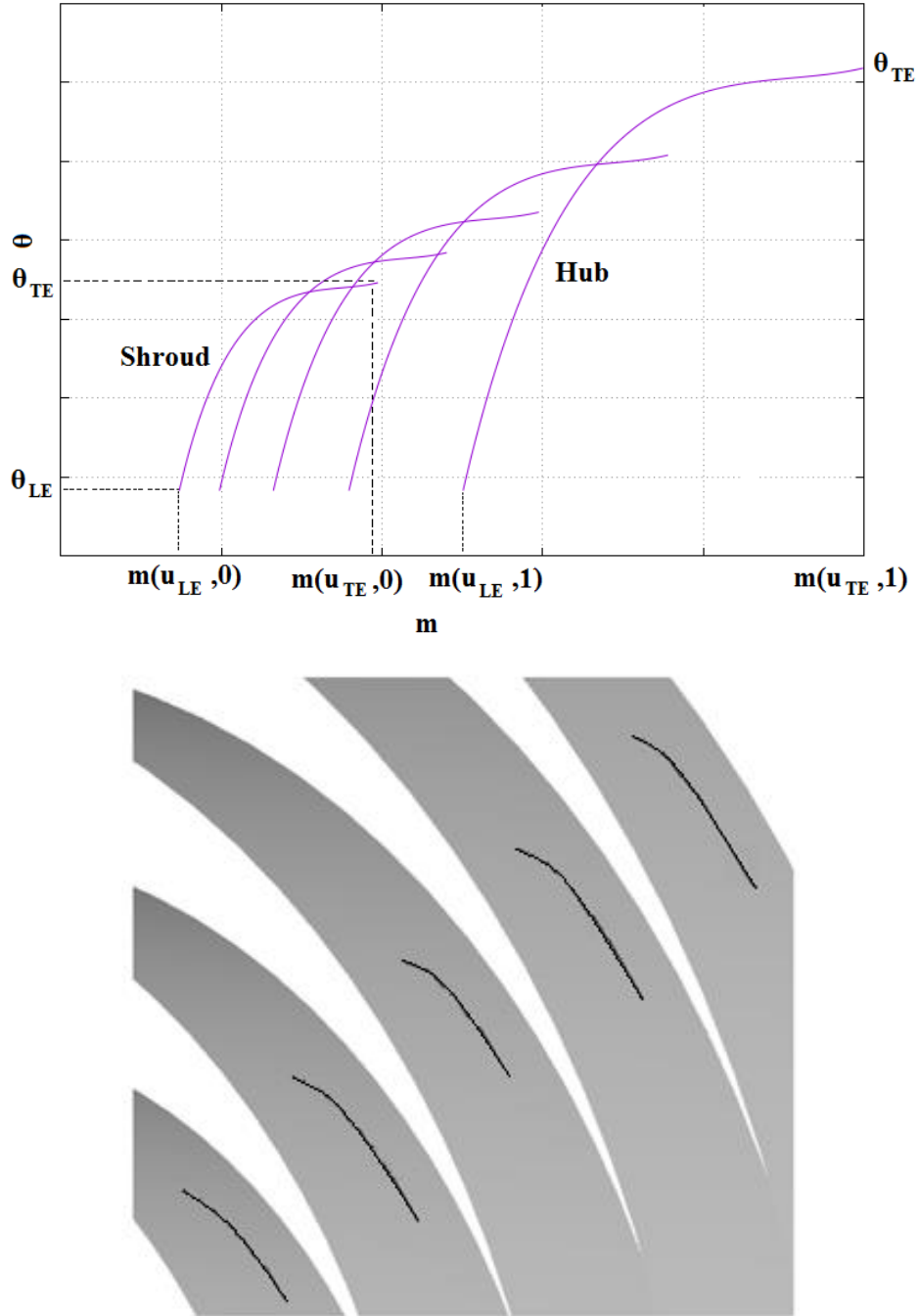


Figure 4.5: Spanwise mean camber lines of the TU Berlin stator, defined on the (m, θ) plane (top) and transformed to the 3D space through Eq. 4.3 (bottom). Each mean camber line lies on the corresponding surface of revolution.

Blade Thickness Profiles

By determining the mean camber line on each surface of revolution, a skeleton of the blade has been defined. The superposition of streamwise thickness profiles, along each spanwise position of the blade, creates the final blade. The thickness profile is imposed in two steps, to increase flexibility. First, the normalized thickness profile \hat{t} with respect to the normalized arc-length s of the mean camber line is defined separately for the pressure and suction sides. Then, a thickness factor t_f , that scales the thickness profiles, is specified for each profile, resulting to a thickness distribution at each spanwise position $v_0 \in [0,1]$ given by:

$$\begin{aligned} t^{PS}(s, v_0) &= \hat{t}^{PS}(s, v_0) t_f^{PS}(v_0) \\ t^{SS}(s, v_0) &= \hat{t}^{SS}(s, v_0) t_f^{SS}(v_0) \end{aligned} \quad (4.9)$$

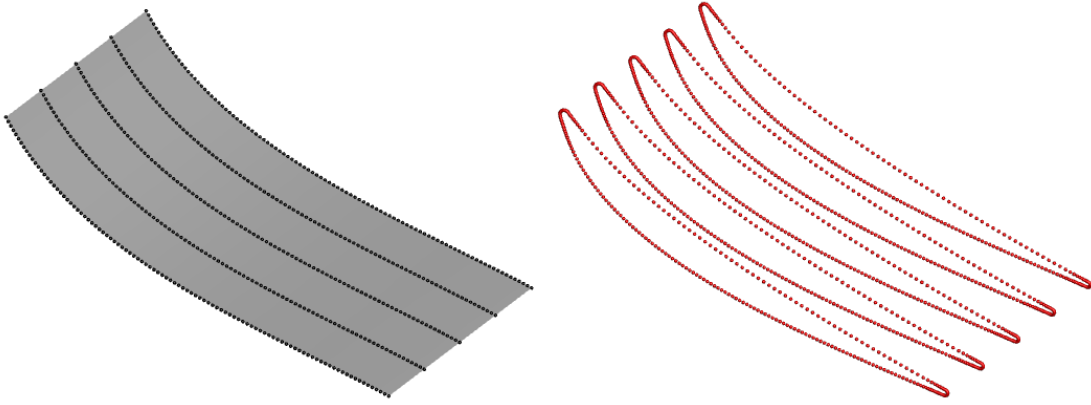


Figure 4.6: The blade of the TU Berlin stator without thickness - mean camber surface on the left, composed by interpolation of the mean camber lines in the 3D space. On the right, the thickness profiles are presented.

Having determined a mean camber line $\mu_{m\theta}$ for the spanwise section on the (m, θ) plane, the imposition of the thickness profiles requires the computation of the normal vector $\hat{n}_{m\theta}(s, v_0)$ at each normalized arc length s point of the mean camber line and the application of the equation to both PS and SS with the appropriate sign as seen bellow:

$$\begin{aligned} c_{m\theta}^{PS}(s, v_0) &= \mu_{m\theta}(s, v_0) \pm n_{m\theta}(s, v_0) \frac{t^{PS}(s, v_0)}{r^2(m(s, v_0))} \\ c_{m\theta}^{SS}(s, v_0) &= \mu_{m\theta}(s, v_0) \mp n_{m\theta}(s, v_0) \frac{t^{SS}(s, v_0)}{r^2(m(s, v_0))} \end{aligned} \quad (4.10)$$

where $r(m(s, v_0))$ is the corresponding radius of the (m, θ) point $\mu_{m\theta}(s, v_0)$ of the mean camber line, through $\Phi^{-1}(v_0)$ and is used to transform the length $t(s, v_0)$ of the 3D space to a length on the (m, θ) plane. For more information see Appendix A. These airfoil curves given in Eq. 4.10 are mapped back onto the (x, y, z) coordinates, through $\Phi^{-1}(v_0)$, to create the 3D skeleton of the blade seen in Fig. 4.6.

The final step is the skinning of the two sides, in order to create two NURBS surfaces, using an algorithm that passes a smooth surface through a set of curves, giving rise to the final 3D blade as in Fig. 4.7.

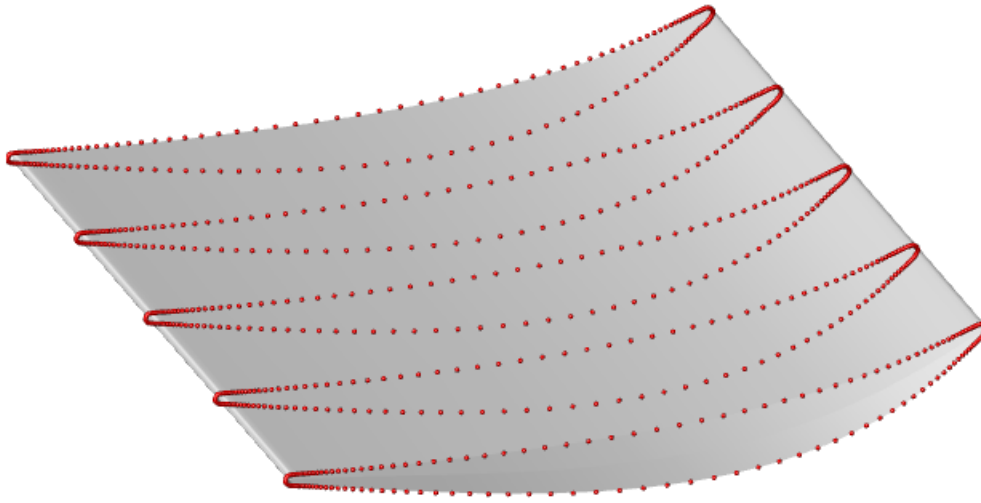


Figure 4.7: The final skinned blade surface.

The final collection of surfaces is exported in neutral CAD format such as IGES.

Leading and Trailing Edge Shapes

In GMTurbo parameterization, after having drawn the pressure and suction side, the shapes of the LE and TE must be defined. There are many available options as seen in Fig. 4.8.

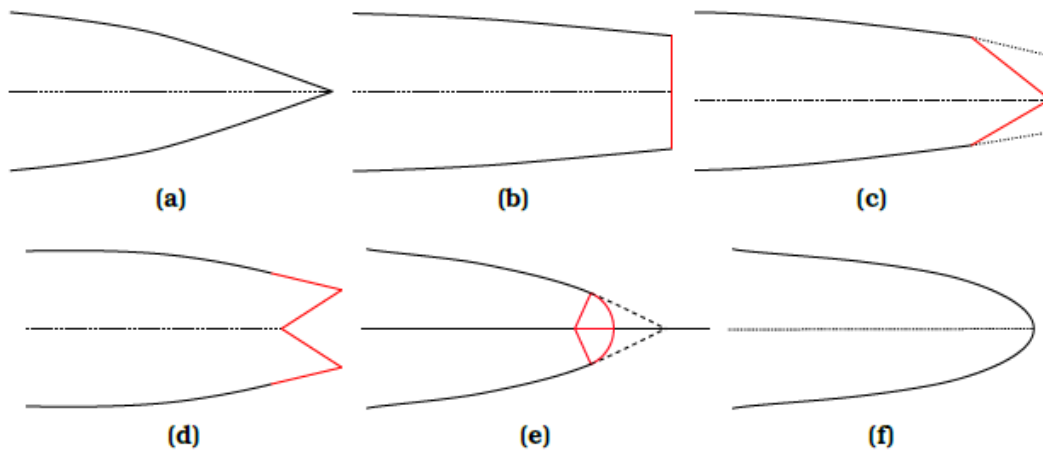


Figure 4.8: Different types of edges which can be produced either for the leading or the trailing edge of a blade. (a) Sharp edge (b) Blunt edge (c) Wedge type edge (d) Dovetail type edge (e) Circular arc edge is created by extending the blade sides and fitting a circle of specified radius between them. (f) General smooth edge, from [2].

4.2 Reverse Parameterization Tool (RPT)

A CFD grid is a common but unhandy form of representation of a blade's shape, since, in terms of design, it is not easy to modify the geometry given in grid form. It is thus essential that it is converted to a more useful CAD form, so that changes in the geometry can be made. In this diploma thesis, the CAD representation is the blade design parameterization GMTurbo. A software to transform a CFD grid into a GMTurbo compatible form is used in this thesis and presented in this chapter. This software was developed in [6]. The process is presented in Fig. 4.9.

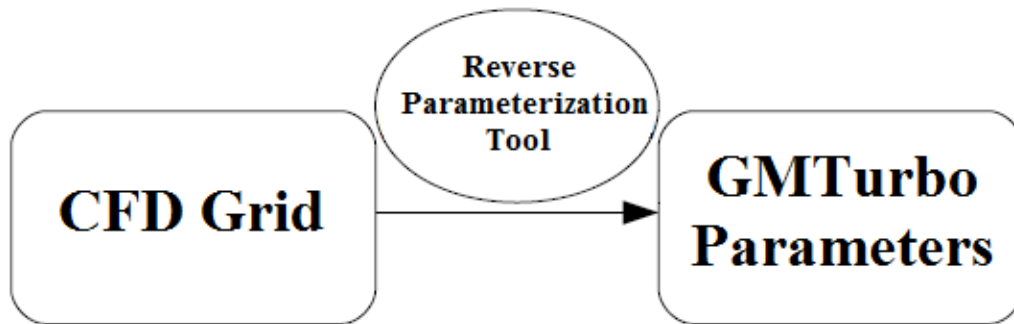


Figure 4.9: RPT uses information provided by the CFD Grid in order to compute the input parameters of GMTurbo.

A 3D CFD grid includes the following information:

- (x, y, z) nodal coordinates.
- The connectivity of these nodes.
- The boundary patches of the volume grid, namely surface patches formed by the nodes that have already been defined. These are the patches where the boundary conditions must be applied by the CFD solver.

The RPT is used to reparameterize structured surface grids, composed by quads. The boundary patches of these grids must be in a standardized format and contain the following:

- An Inlet and an Outlet Patch where the CFD solver applies the inlet and outlet boundary conditions.
- Wall Patches: solid boundaries where the wall boundary conditions are applied. These usually are hub and shroud patches, pressure and suction side patches. In turbomachinery CFD grids, wall patches might be rotating with the exception of the shroud that is usually stationary or the stationary blades.
- Periodic Patches. In case the CFD domain has a periodic repeating geometry and flow field.

The patches of the CFD grid, used in this thesis, can be seen in Fig. 4.10.

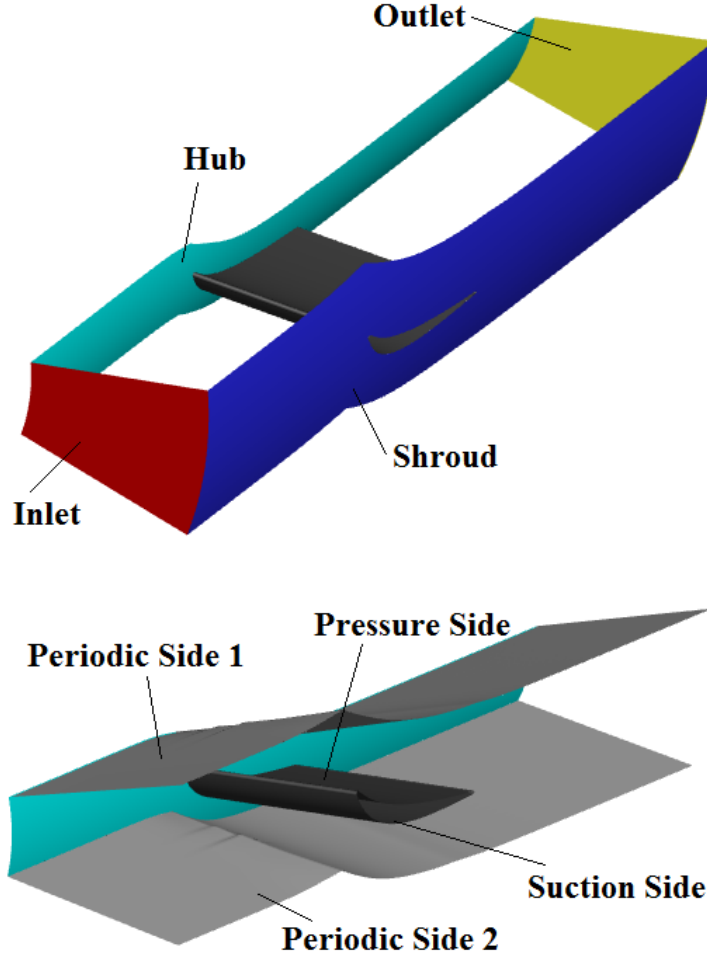


Figure 4.10: Patches of the CFD grid used for the analysis of the TU Berlin stator blading.

Starting with a turbomachinery surface grid like the one presented above, a back-to-CAD method, converting the CFD grid into a set of geometric parameters, compatible with the GMTurbo input parameters, is described below. The meridional contour, mean camber lines and thickness profiles of the existing grid are to be computed.

Meridional Contour of the Grid

The first step is the computation of the meridional contour. Having the surface grid of the meridional patches, hub, shroud, inlet and outlet, the (r, z) generatrices of each meridional surface are computed. The edge between the surface grid of a meridional patch with one of the periodic patches, as seen in Fig. 4.11 and Fig. 4.12, is a node representation of one generatrix of the meridional surface in (x, y, z) coordinates. Projecting this generatrix onto the (r, z) plane using

$$\begin{aligned} r &= \sqrt{x^2 + y^2} \\ z &= z \end{aligned} \tag{4.11}$$

produces the meridional projection of hub, shroud, inlet and outlet patches. Also, projecting the grid edges that correspond to the LE and TE onto the (r, z) plane, produces the meridional curves of the two edges. The hub and shroud generatrices are, then, approximated by a NURBS curve. The user defines the degree and the number of control points of the NURBS curve.

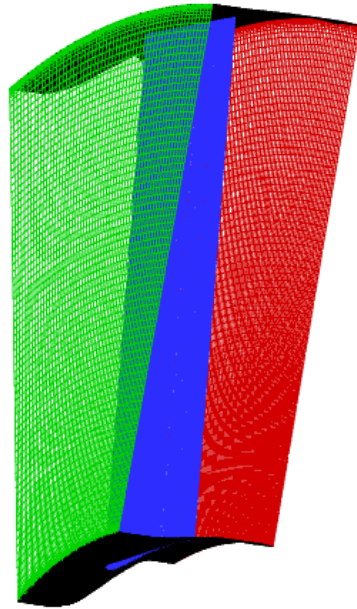


Figure 4.11: The CFD surface grid and the patches namely hub (black), shroud (black), inlet (red) and outlet (green) are shown.

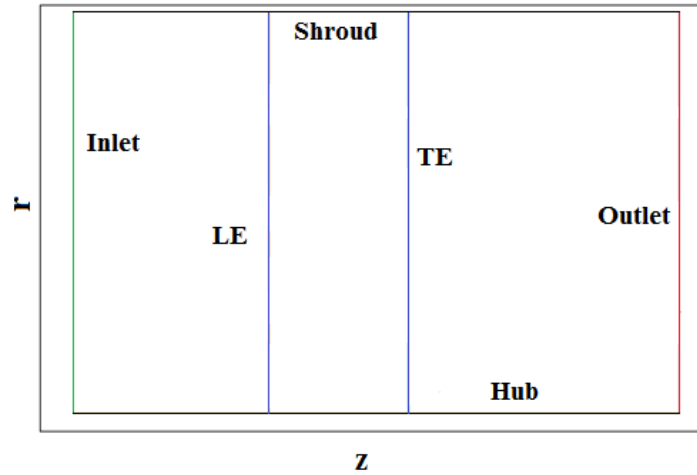


Figure 4.12: The meridional contour, as it results from the projection of the axisymmetric patch nodes and the edge nodes onto the (r, z) plane.

Next, a user-defined number of N' spanwise generatrices are generated. Having the NURBS curves of hub and shroud at the meridional plane, in (r, z) coordinates, in order to produce intermediate streamlines in NURBS representation, a linear interpolation of the control points is used. After defining the N' generatrices, the operations take place for each spanwise generatrix, thus the following are applied to each and every spanwise generatrix, in order to attain spanwise distributions of data.

Mean Camber Line and Thickness Data Computation

Based on the (r, z) NURBS curve of the N'^{th} generatrix, a revolved surface in the (x, y, z) space and the Φ transformation function to the (m, θ) plane are generated. A NURBS revolved surface that rotates around the z -axis to a certain angle θ , can be computed using a single (r, z) NURBS generatrix. Rotating this generatrix's control points at discrete angles from 0 to θ , leads to sets of control points of type (r, θ_i, z) . These control points generate a NURBS revolved surface with angle of revolution θ , the generatrix of which is the initial (r, z) generatrix. The Cartesian representation of the (r, z) surface points is given by:

$$(x, y, z) = (r \cos \theta, r \sin \theta, z) \quad (4.12)$$

Using the $N^{th}(r, z)$ generatrix a transformation function Φ that maps every (r, θ, z) point of the revolved surface to the (m, θ) plane can be computed.

The revolved surfaces of the spanwise generatrices, between hub and shroud and the grid of the blade that lies within hub and shroud and intersects the spanwise revolved surfaces can be seen in Fig. 4.13:

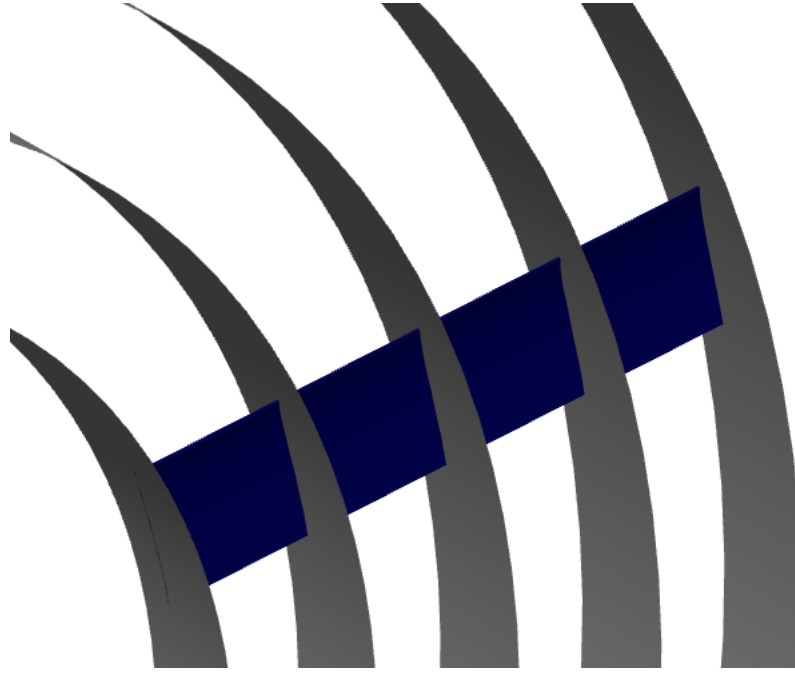


Figure 4.13: Revolved surfaces of the spanwise generatrices, between hub and shroud and grid of the blade within hub and shroud that intersects the revolved surfaces.

Pressure and Suction blade sides:

The intersection between the blade grid and the revolved surface produced results in the definition of the pressure and suction side curves that lie on the revolved surface. Additionally, the intersection points between a structured surface grid, consist of quads and the NURBS surface that intersects it, are found. These points lie on both the revolved surface and the blade, resulting to a set of points such as in Fig. 4.14. This method is described in detail in [6].

Having the blade airfoil (x, y, z) points in the 3D space and using the transformation function Φ , seen in Eq. 4.3, that has also been defined by knowing the (r, z) generatrix, the blade (x, y, z) points can be transformed to (m, θ) coordinates. The analysis changes from the 3D Cartesian space to the (m, θ) plane as seen in Fig. 4.15. The (m, θ) points of the two sides of the airfoil are approximated with two NURBS curves in order to achieve a continuous representation of the two sides. To do so, a NURBS constrained approximation algorithm was used, so that the resulting curves respect the continuity of first derivative at the LE. This method is described in detail in [6].

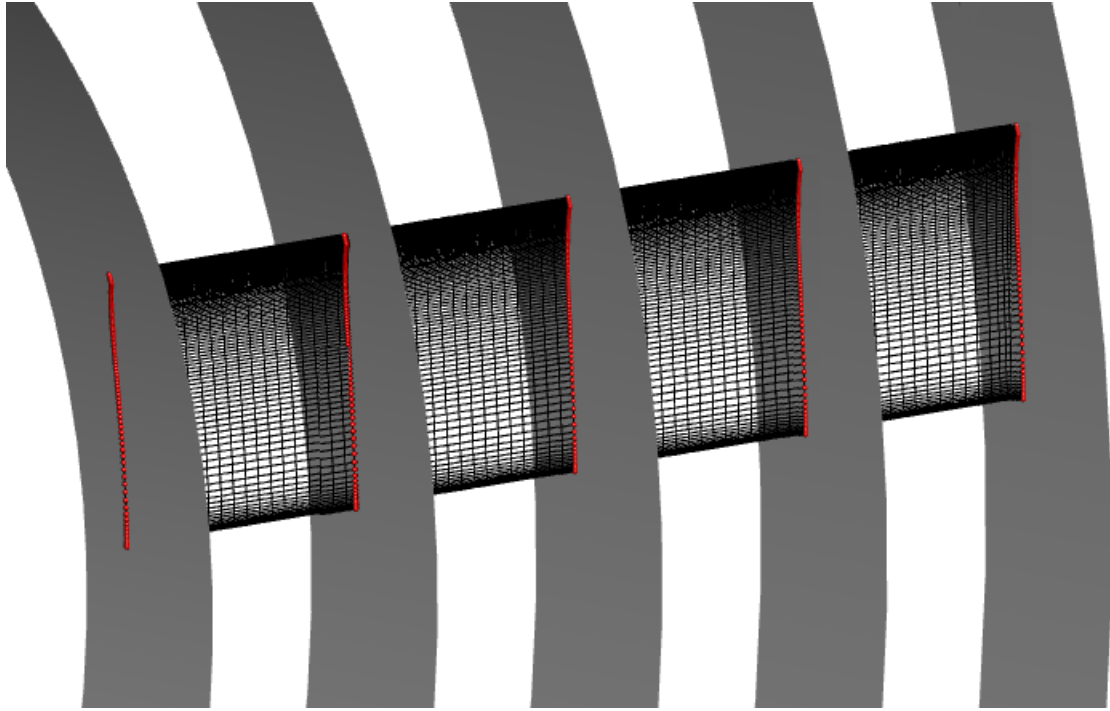


Figure 4.14: Discrete points (in red) of the blade grid that lie on the surfaces of revolution.

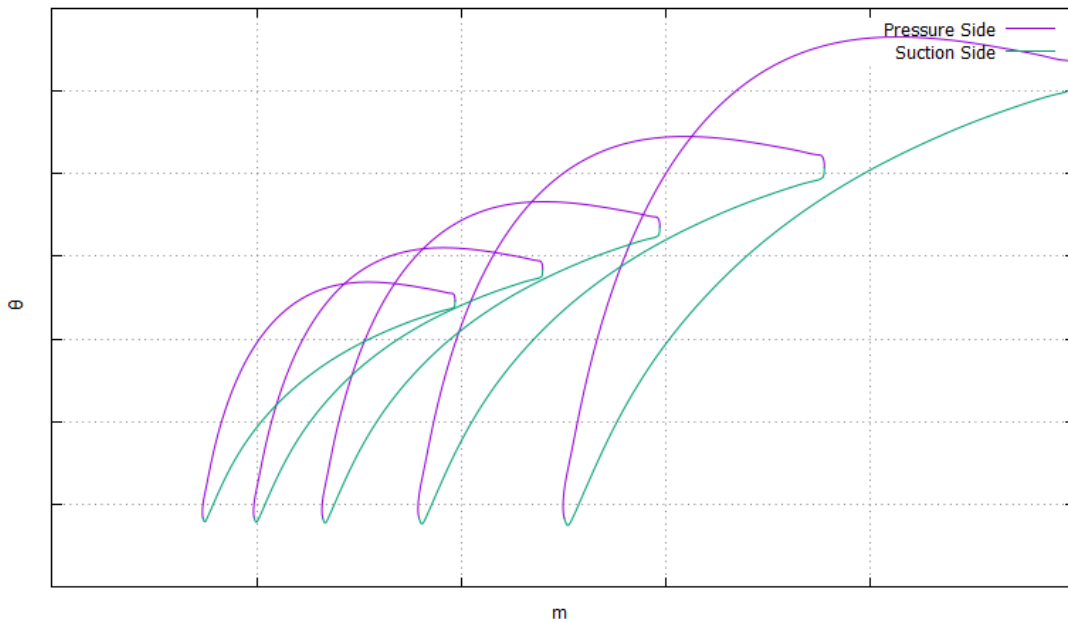


Figure 4.15: The points of Fig. 4.14 transformed into the (m, θ) plane and approximated by NURBS curves, creating spanwise airfoils on the (m, θ) plane.

Mean Camber Line:

The mean camber line of an existing blade airfoil, on a $2D$ plane, is computed as the exact mean camber line according to the following definition; *a line joining the leading and trailing edges of an airfoil equidistant from the upper and lower surfaces*. This method is described in detail in [6].

To produce a mean camber line compatible with a four point Bézier representation, the exact mean camber line points are approximated using a cubic Bézier curve that is referred to as the four point mean camber line. A cubic Bézier approximation technique enforced with constraints is used to approximate the exact mean camber line points, while preserving through constraints, the endpoints, LE and TE preservation and endpoint tangents such as the metal angles preservation of the airfoil. This method is described in detail in [6]. The exact and the cubic Bézier mean camber lines are given in Fig. 4.16.

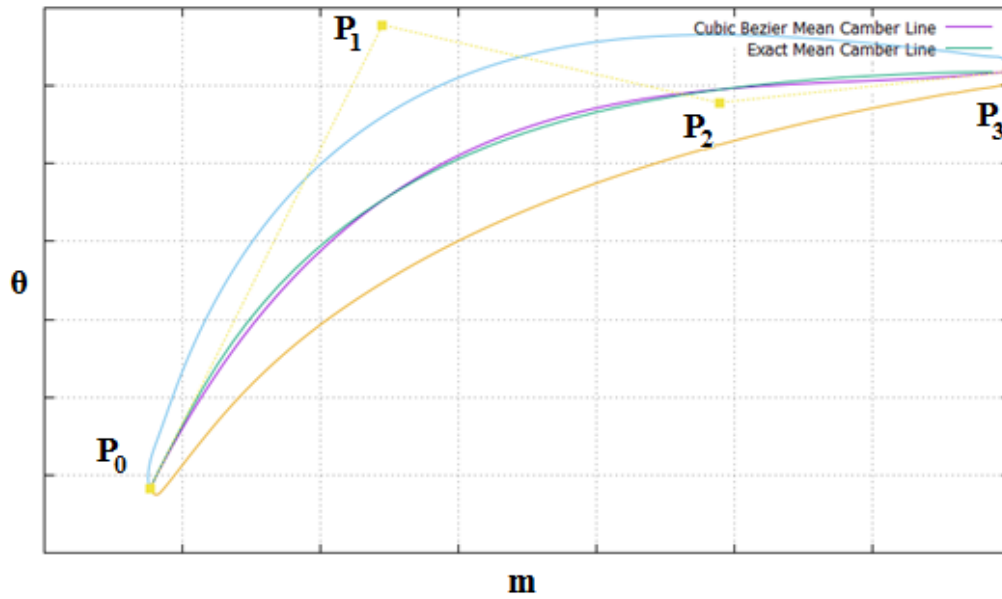


Figure 4.16: The exact (green) and cubic Bézier (purple) mean camber lines of the TU Berlin stator blades. The blade airfoil contour is also shown in light blue and yellow.

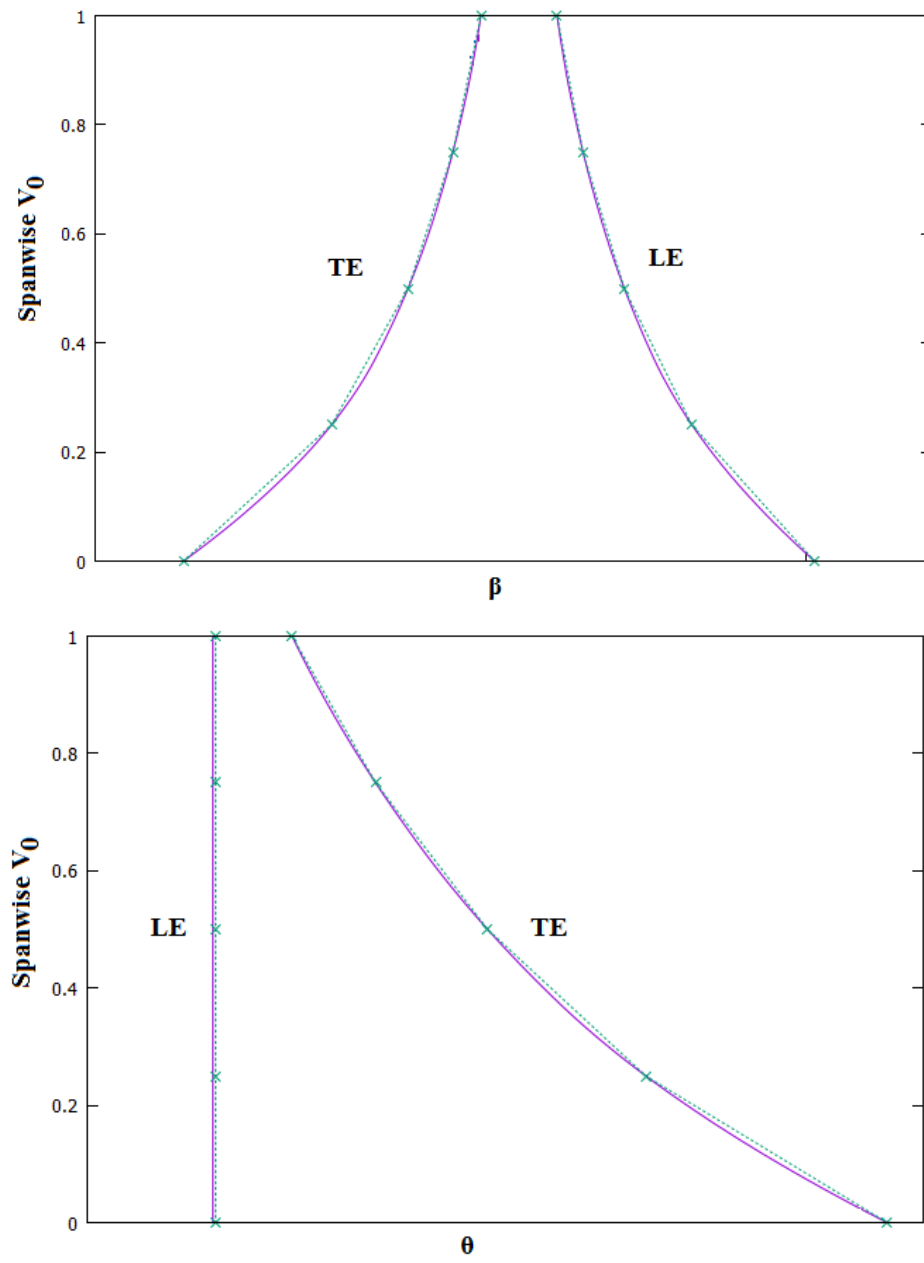
Having computed the cubic Bézier mean camber line, P_0, P_1, P_2 and P_3 , the angles θ , β and δ are computed using:

$$\theta_{LE} = P_{0,\theta}, \theta_{TE} = P_{3,\theta} \quad (4.13)$$

$$\beta_{LE} = \text{atan} \left(\frac{P_{1,\theta} - P_{0,\theta}}{P_{1,m} - P_{0,m}} \right), \beta_{TE} = \text{atan} \left(\frac{P_{2,\theta} - P_{3,\theta}}{P_{2,m} - P_{3,m}} \right) \quad (4.14)$$

$$\delta_{LE} = \text{acos} \left(\frac{\overrightarrow{MP_1} \cdot \overrightarrow{MP_0}}{|\overrightarrow{MP_1}| |\overrightarrow{MP_0}|} \right), \delta_{TE} = \text{acos} \left(\frac{\overrightarrow{MP_2} \cdot \overrightarrow{MP_3}}{|\overrightarrow{MP_2}| |\overrightarrow{MP_3}|} \right) \quad (4.15)$$

Equations 4.13, 4.14, 4.15 applied to each spanwise four point mean camber line, result to spanwise distributions of the angles β, θ and δ as in Fig. 4.17.



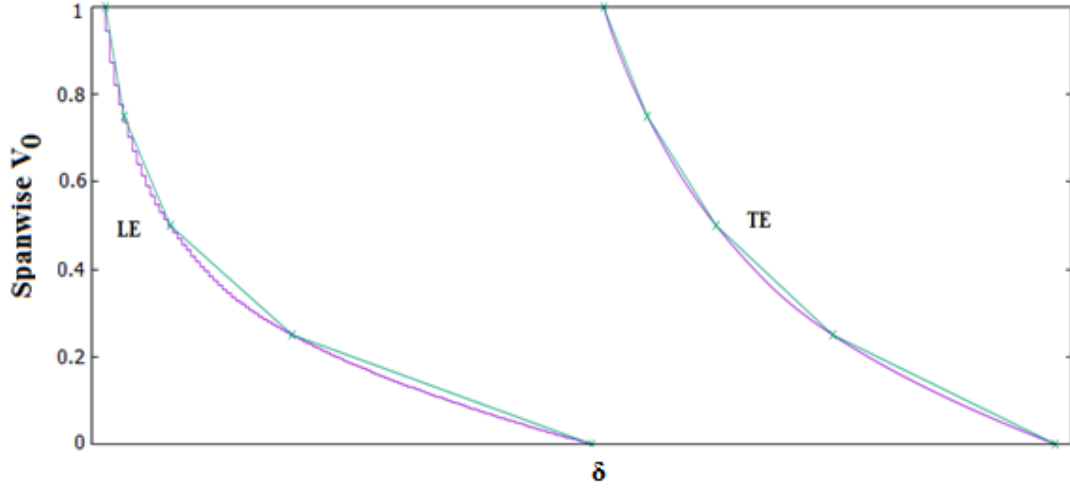


Figure 4.17: Spanwise distributions of the angles β , θ , and δ of the TU Berlin stator. The discrete green points are extracted from the surface grid, for every spanwise position. Then, a NURBS curve interpolation provides the continuous spanwise distribution of the angles (purple lines).

Thickness Profiles:

In order to complete the reparameterization of the blade grid, last step is to compute the thickness profiles that are a combination of the non-dimensional streamwise thickness profiles for each spanwise position and the spanwise thickness factor distribution. Having the PS and SS NURBS representations and the mean camber line cubic Bézier representation, the normal distances of the mean camber line to both sides are computed, resulting to the two thickness profiles $t(u)$ described in Eq. 4.9. Then, dividing each profile with its maximum value that is the thickness factor of the profile, $t_f(v)$ seen in Fig. 4.18, leads to the non-dimensional thickness profiles $\hat{t}(u)$ seen in Fig. 4.19.

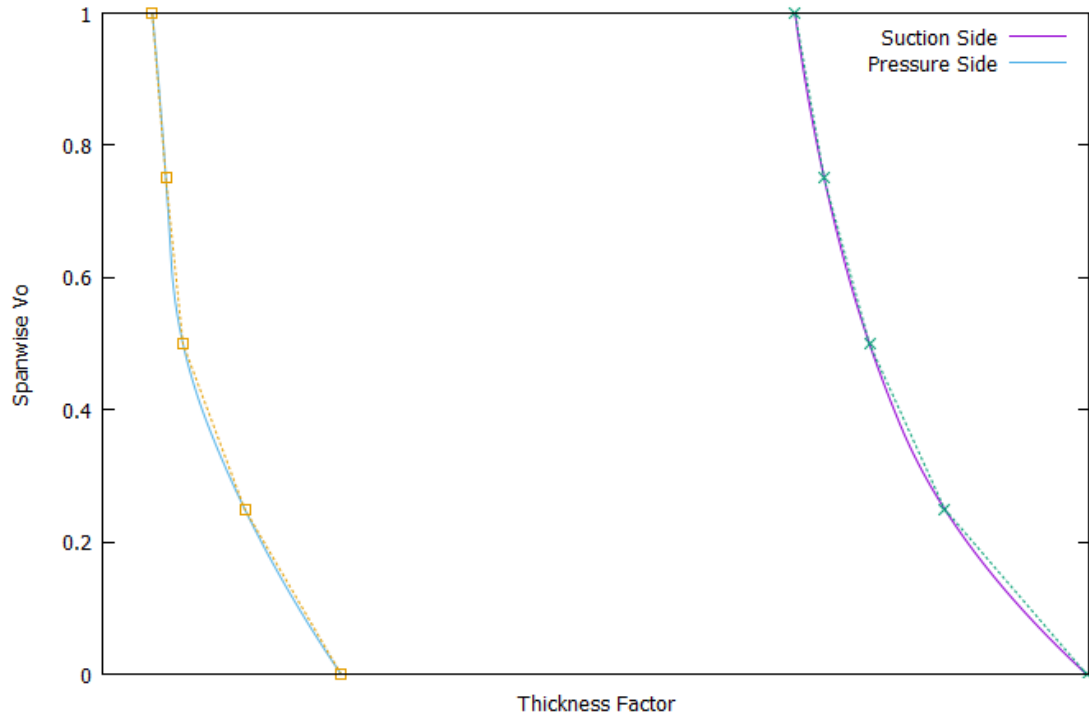


Figure 4.18: Spanwise thickness factor distributions of the TU Berlin stator blade.

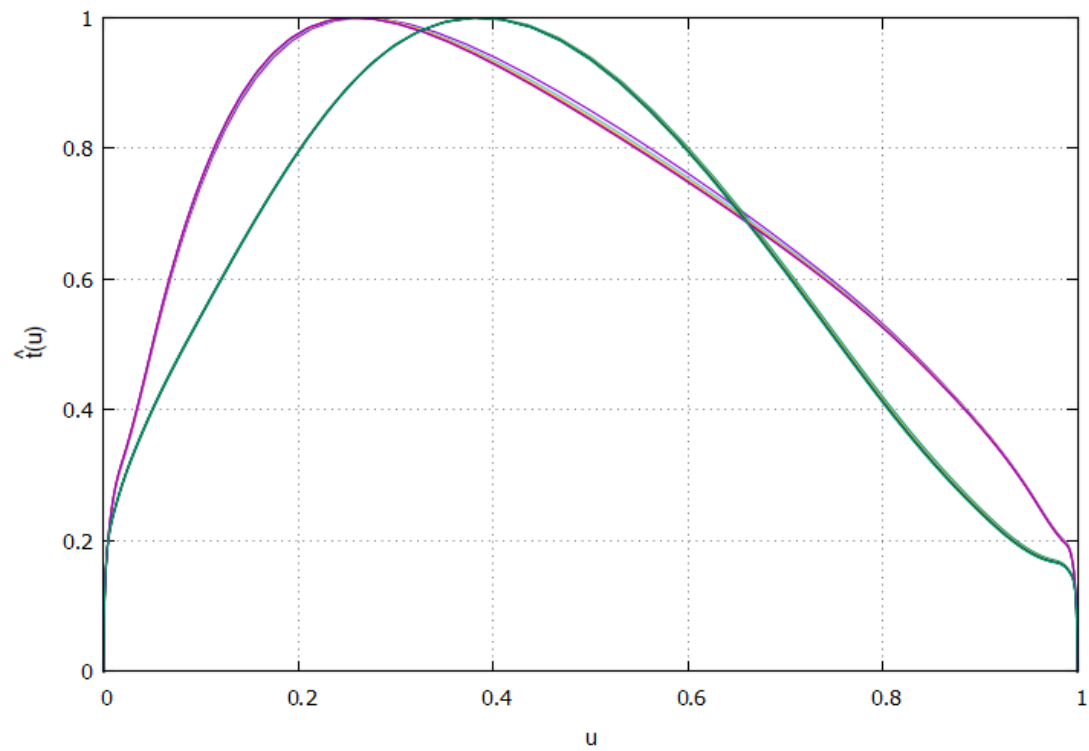


Figure 4.19: Streamwise thickness profiles of the TU Berlin stator blade.

Validation of the RPT

The RPT is validated in this section. The results demonstrate the level of accuracy in geometry terms. The parameterized basic geometry is studied and presented in order to highlight the capabilities of the RPT. The initial and reparameterized geometry are plotted together in Fig. 4.20.



Figure 4.20: Initial (grey) and reparameterized (red) geometry of the TU Berlin stator plotted together, using $N' = 5$ generatrices in the reparameterization.

In order to quantify the deviation between the two blades of Fig 4.20, the field of d is defined by:

$$\Delta = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \quad (4.16)$$

where $\Delta x, \Delta y, \Delta z$ is the x, y, z distance between initial nodal positions and the corresponding positions on the reparameterized blade. The results of such a comparison can be seen in Fig. 4.21. It is obvious that the accuracy of the reparameterization is strongly connected to the number of generatrices chosen for the reparameterization.

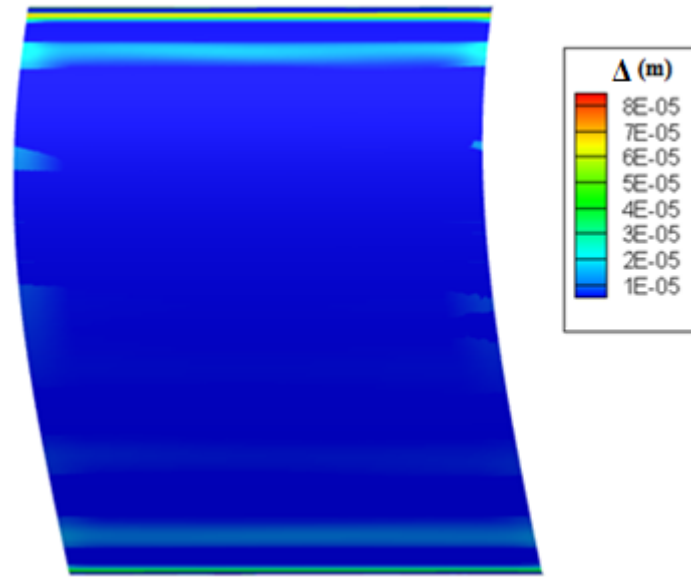


Figure 4.21: Deviation of the two blades in m , using $N' = 5$ generatrices in the reparameterization. The blade chord length is approximately $0.180\ m$ and it is repeated for comparison.

By comparing the two geometries, the initial and the reparameterized, as seen in Fig. 4.20 and 4.21 the conclusion is they differ by very little and this deviation can be reduced further by changing some parameters of the process.

4.3 Grid Adaptation Tool (GAT)

The back-to-CAD method presented in Section 4.2 generates a geometry that can be easily modified and optimized, by altering the CAD design variables of GMTurbo. However, through this transformation, the nodal representation is lost, when modifying the CAD parameters. To make it possible to perform CFD simulations on the modified CAD geometry a grid has to be generated around the geometry. To avoid mesh generation, the initial grid is exploited. A surface grid adaptation tool, GAT, developed in [6], that adapts the initial CFD grid to the reparameterized CAD geometry is used in this diploma thesis combined with the 2D and 3D Spring Analogy Morphers developed by the PCOpt/NTUA in [3], as seen in Fig. 4.22.

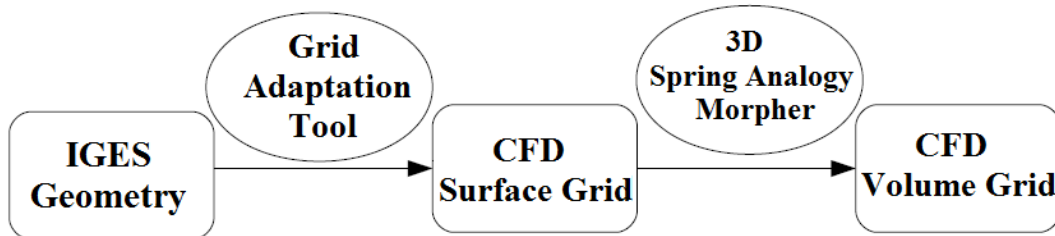


Figure 4.22: To create a grid around the reparameterized surface, GAT is used. This adapts the initial surface grid to the CAD geometry. Then, the volume grid is displaced to fit to the adapted surface grid, using the 3D spring analogy technique.

The GAT used in this thesis, adapts the initial surface grid to the CAD geometry. The method takes advantage of the availability of an initial grid that is taken for granted in this thesis to generate a new one around the CAD geometry. It is performed in two steps. First, the surface grid of the reparameterized wall patches is computed; by projecting the initial CFD surface nodes onto the reparameterized NURBS surfaces, for the various wall patches namely Hub, Shroud, Pressure Side and Suction Side as seen in Fig. 4.23. The structure and connectivity of the surface grid are maintained and only the coordinates of the nodes are adapted to the reparameterized and consequently slightly changed, surface shape. Then, the volume grid of the whole CFD domain is morphed with reference to the displacement of the surface grids computed in the previous step.

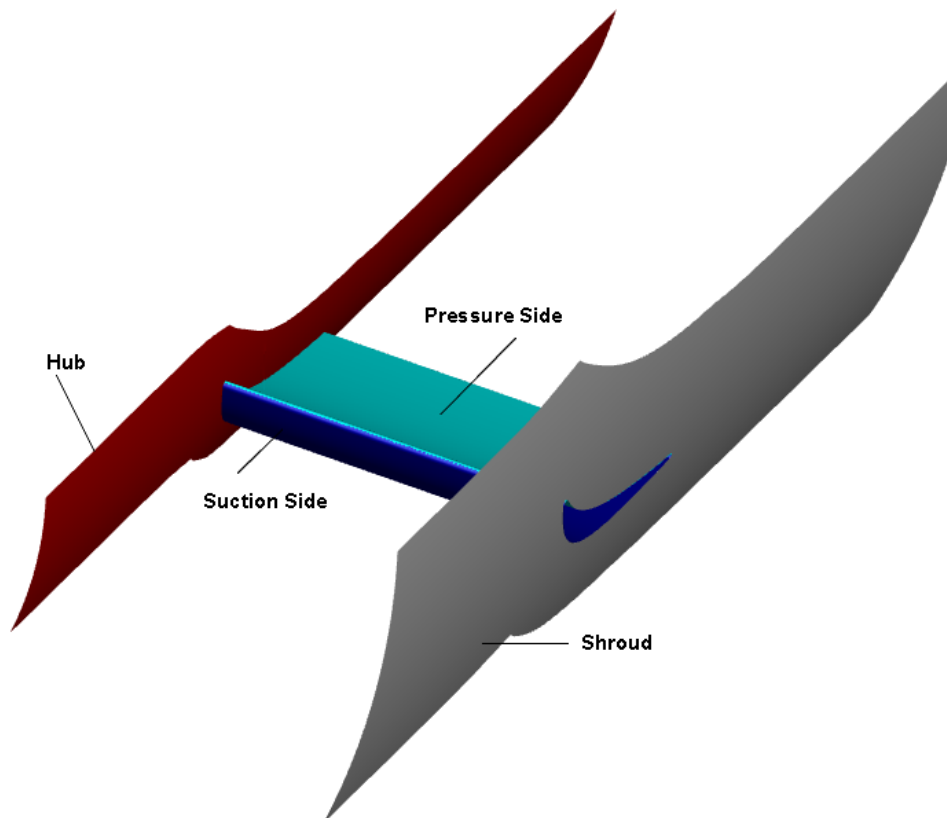


Figure 4.23: Wall patches of the blade grid of the TU Berlin stator.

Step 1: Surface Grid Adaptation

To compute the surface grid of the reparameterized geometry of each patch, the wall patch nodes of the initial grid are projected onto the corresponding NURBS surfaces, obtaining a $2D$ representation of each surface. Then, in $2D$ coordinates, the necessary morphing is performed and the morphed $2D$ grid is, then, transformed back to $3D$ coordinates.

GAT uses the initial mesh and the NURBS curves of the wall patches resulted from the GMTurbo, seen in Fig. 4.24, to create a new mesh with the same connectivity as the initial but node coordinates adapted to the parameterization surfaces.

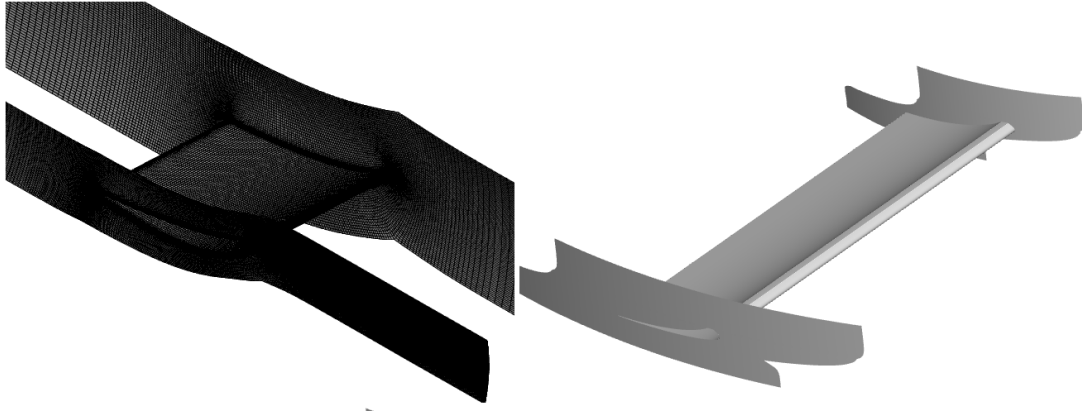


Figure 4.24: Initial mesh (left) and the NURBS curves of the wall patches resulted from the GMTurbo (right).

The patches are separated into two categories:

- In the first category, the PS and SS patches are NURBS surfaces, each point of which is represented by two parameters: (u, v) .
- The second category includes the patches Hub and Shroud that can be represented from a NURBS revolved surface each point of which can be represented by the NURBS surface (u, v) parameters or the (m, θ) parameters through the conformal mapping. It is preferable to use the (m, θ) representation instead of the (u, v) representation of the NURBS revolved surface, since the first preserves the periodicity of the nodes that belong to the periodic patches.

Two nodes in (x, y, z) coordinates, that have a periodic connection, have a specific angular θ pitch difference. This pitch is preserved when transforming into (m, θ) parameters, maintaining the periodicity of the nodes. Consequently, each one of the four wall patches corresponds to a parametric surface which, by definition, can be represented by two parameters, (u, v) or (m, θ) .

Projecting a single point onto a parametric surface, results to the closest to this (x, y, z) point that also belongs to the parametric surface. The latter can also be described with two parameters (u, v) , since it belongs to the parametric surface. Consequently, projecting all the nodes of a wall surface patch to its parametric surface produces a 2D grid of parameters seen in Fig. 4.25 and 4.26. Repeating the procedure for each of the four wall patches, four 2D grids of parameters are computed.

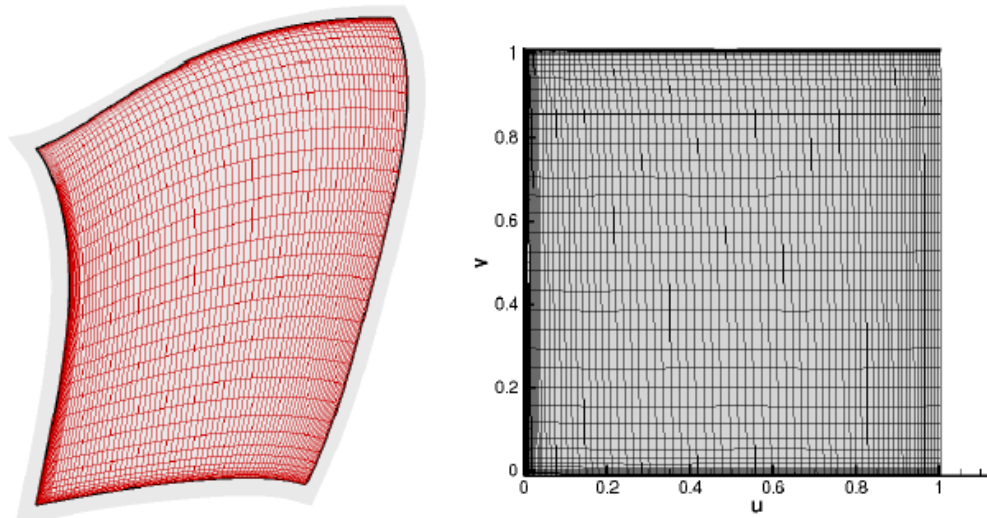


Figure 4.25: The projection of the surface grid onto the NURBS reparameterized surface on the left can be transformed into 2D (u, v) points, resulting to a 2D grid (right figure).

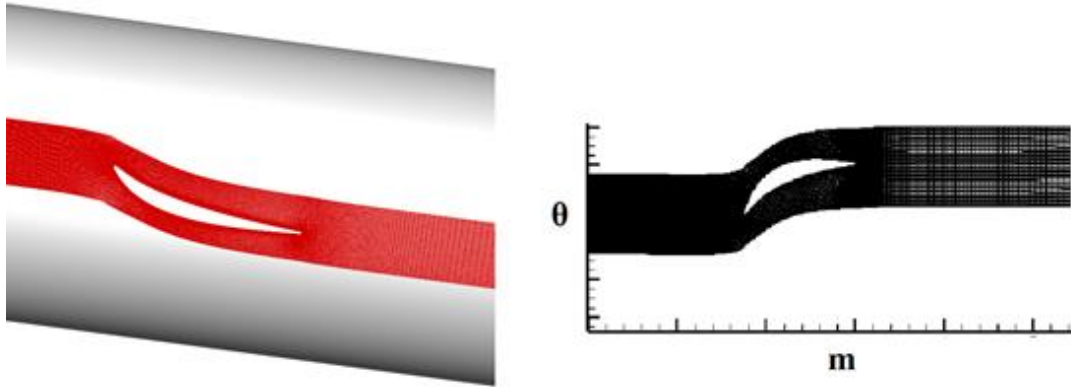


Figure 4.26: The projection of the surface grid onto the NURBS reparameterized revolved surface on the left can be transformed into 2D (m, θ) points, resulting to a 2D grid (right figure).

However, the edges of the grid are not projected onto the edges of the surface, as seen in Fig. 4.27. To force the surface grid to fit the NURBS surface, a deformation is applied to the 2D parameters of each wall patch computed earlier. The deformation is performed using the 2D Spring Analogy Technique. This software takes a 2D grid, computed earlier by projecting the surface nodes onto parametric surfaces and the 2D position of the edge nodes of the grid as input and distributes the internal nodes, with reference to the edge positions.

The position of the edge nodes can be found with the following technique. The edges of the surfaces are 3D NURBS curves provided by the parameterization, since they are the intersections of NURBS surfaces resulted from GMTurbo, as seen in Fig. 4.27. The edge nodes of the adapted grid must belong to those edge curves. Thus, the edge nodes of the initial grid are distributed onto the 3D edge curve, using the distance distribution they had in the initial grid. Since they belong to the 3D edge curve, they also belong to the wall NURBS surface, thus they are represented by two parameters

provided by the NURBS surface (u, v) or (m, θ) . These edge parameters are given as an input to the 2D spring analogy morpher. The morphing practically slides the nodes on the surface to make the surface grid fit the edges of the NURBS surface as seen in Fig. 4.28.

After morphing the 2D grid, the displaced parameters of the surface nodes are found. It is easy to go back to 3D, using the equations of the corresponding surface, $S(u, v)$ for the PS and SS and Φ^{-1} for hub and shroud, attaining the displaced (x, y, z) surface patches.

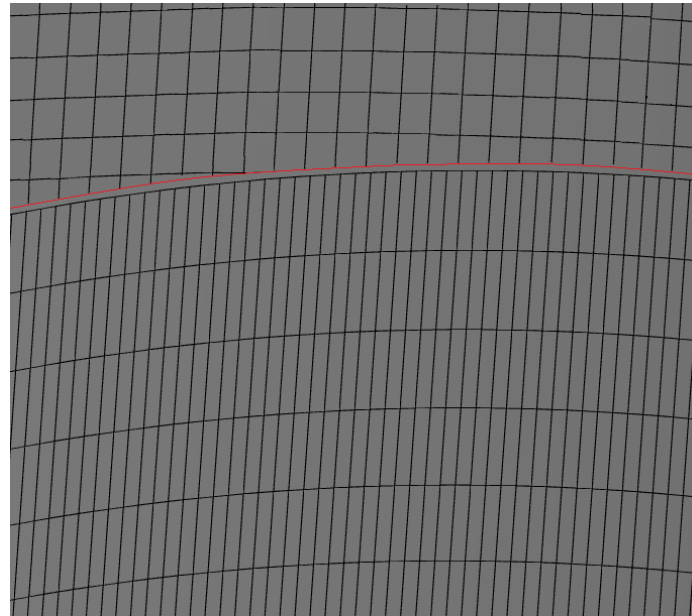


Figure 4.27: The projection of the mesh onto the NURBS surfaces is not exact, since the edge nodes are not projected onto the real edges of the surface. The need to displace the surface mesh to fit the edges of the parameterization (red line) comes up.

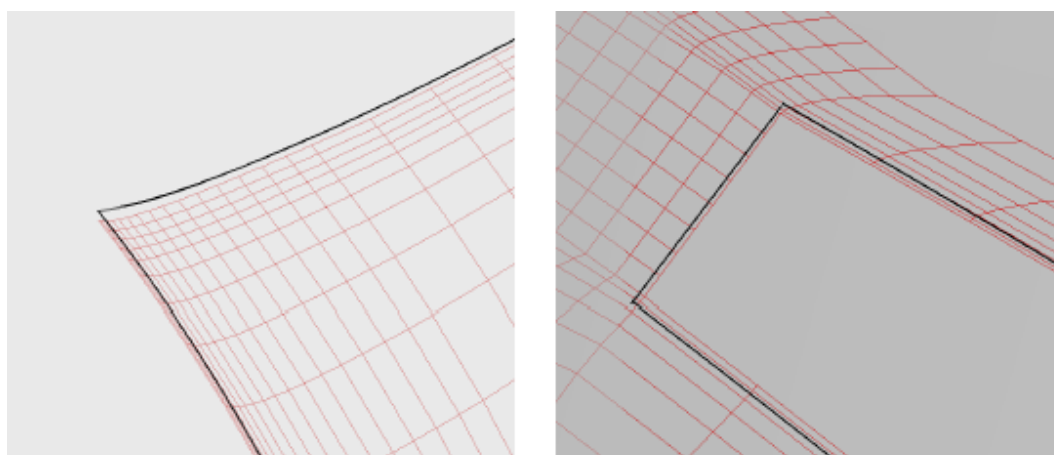


Figure 4.28: The edges according to the reparameterization (black curves) are different than the projections of the edges of the initial grid onto the reparameterized surfaces. Thus, a 2D spring analogy morphing takes place to displace the projected grid (red grid) with respect to the edge positions of the parameterization.

Step 2: Volume Grid Adaptation

Using the surface wall patches computed in the previous step, a deformation to the initial volume grid can be applied using the 3D spring analogy technique [3], to adapt the internal volume grid, with respect to the position of the surface patches. This deformation results to a volume grid of the reparameterized CAD blade. This grid has the same structure and connectivity as the initial one but is displaced in terms of coordinates as seen in Fig 4.29.

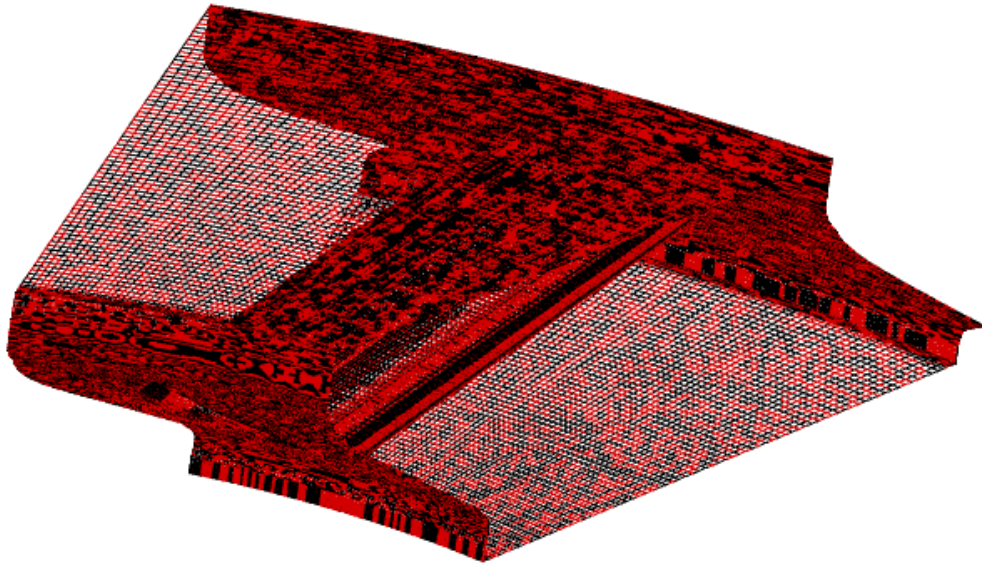


Figure 4.29: The resulting red grid is very close to the initial black of the TU Berlin case, depending on the user defined accuracy selected for the reparameterization that is, the number of generatrices and the number of control points in NURBS approximations.

GAT used in an EA Optimization

To generate a grid on the GMTurbo geometry, during an EA optimization, the method presented in Section 4.3 is used. However, a part of the method is executed only once, before the optimization begins, as a pre-processing step and the rest is integrated into the optimization workflow and creates the grid for every candidate geometry.

To be more specific, the first part of the method, the projection of the surface nodes onto the NURBS surfaces of the parameterization, is common for every candidate geometry, since it provides the connectivity of the various patches in 2D coordinates. Mapping back to the corresponding NURBS surface of the current EA evaluation, provides the new coordinates of the surface grid. Hence, the 2D parametric grid of each wall patch, seen in Fig. 4.23, is created once, at the beginning of the optimization. It is morphed for each candidate geometry, according to the edge positions of that geometry. The rest of the method follows and is integrated into the EA workflow to be repeated for every candidate geometry.

Chapter 5

Optimization Methods

5.1 Evolutionary Algorithms – EASY Software

Evolutionary algorithms (EA) are a class of search heuristics that can be applied to many different tasks providing a heuristic-based approach to solving problems. The premise of an EA is based on the process of natural selection. An EA contains four overall steps: initialization, selection, evolution operators, namely mutation and crossover, and termination. These steps each correspond, roughly, to a particular facet of natural selection, and provide easy ways to modularize implementations of this algorithm category. In an EA, fitter members will survive and proliferate, while unfit members will die off and will not contribute to the gene pool of further generations. EAs are controlled by a number of parameters which are crucial for the success and efficiency of the search. The processes of an EA can be seen in Fig. 5.1. More information is given in [11].

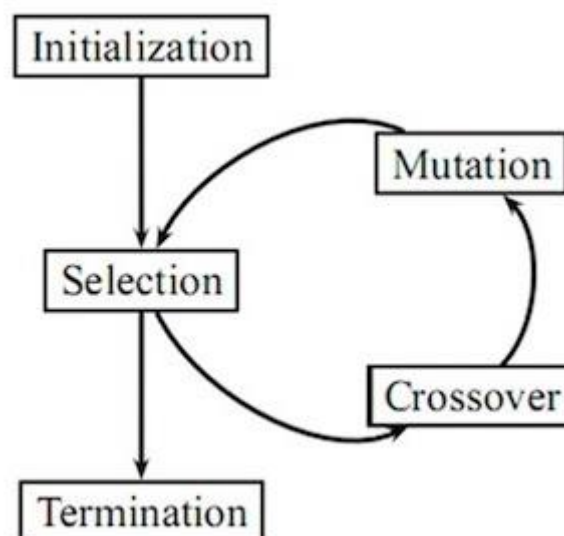


Figure 5.1 : Processes of an EA.

Initialization

In order to begin an initial population of solutions is created. The population contains an arbitrary number of possible solutions to the problem, called members. It is often being created randomly or, if some prior knowledge of the task is known, roughly centered on what is believed to be ideal. It is important that the population encompasses a wide range of solutions, because it essentially represents a gene pool. In order to explore many different possibilities over the course of the algorithm, many different genes should be present.

Parent Selection

Once a population of λ members is created, these members, called offspring, are evaluated according to a fitness function. A fitness function is a function that takes in the characteristics of a member and outputs a numerical representation of how viable of a solution it is. The members of the population that correspond to the best values of the fitness function are called parents and are given as μ . After selecting the top members, they are used to create the population of next generation.

Evolution Operators

This step includes two operators: crossover and mutation. Using the characteristics of the selected parents, new offspring are created that are a mixture of the parents' qualities. Furthermore, a new genetic material should be introduced into the generation. This step is mutation and it is done by changing a small portion of the offspring such that they no longer perfectly mirror subsets of the parents' genes. Mutation typically occurs probabilistically, that is the chance of a child receiving a mutation as well as the severity of the mutation is governed by a probability distribution.

Termination

Eventually, the algorithm must end. There are two cases in which this usually occurs: either the algorithm has reached some maximum runtime, or the algorithm has reached some threshold of performance. At this point a final solution is selected and returned.

Multiple Objective Optimization

EAs can also be extended to use many fitness functions. As a result, a set of optimal points is produced instead of a single optimal point. The set of optimal solutions is called the Pareto front and contains elements that are equally optimal in the sense that no solution dominates any other solution in the front. Then, the set is narrowed down to a single solution, based on the context of the problem by the decision maker. The Pareto front of an example case can be seen in Fig. 5.2.

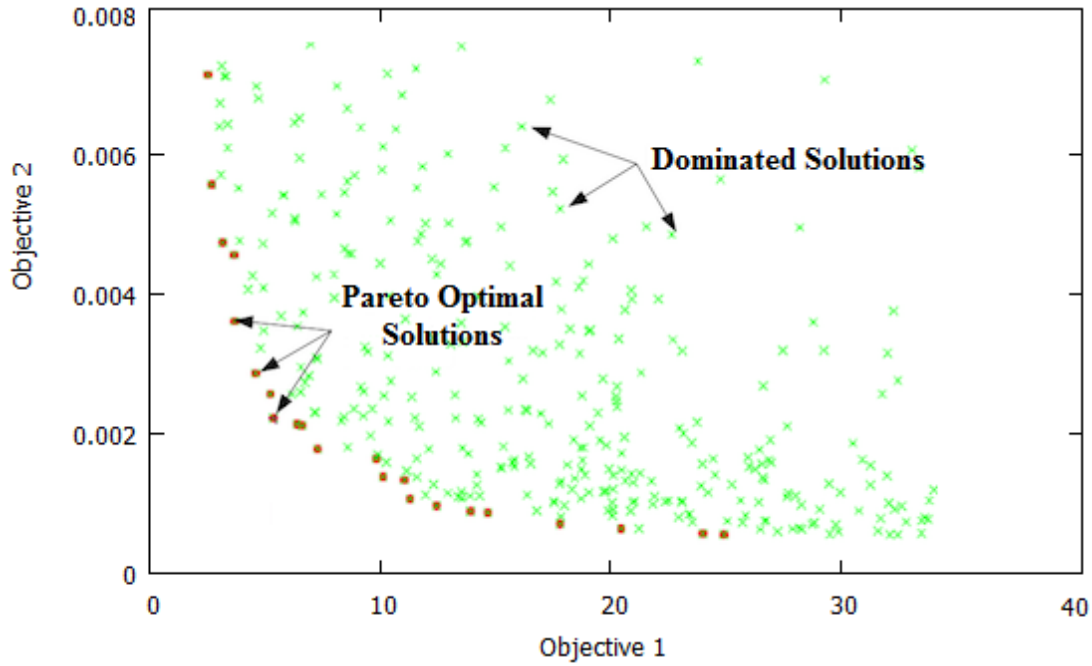


Figure 5.2: Pareto front of an example case.

Metamodel-Assisted Evolutionary Algorithms:

Optimization based on stochastic methods and, in particular, evolutionary algorithms are currently being used. Despite their advantages, all of the population-based search algorithms require excessive computational time due to the excessive number of candidate solutions which need to be evaluated through costly computational models. The reduction of this computing cost, so as to make stochastic optimization both efficient and effective is an area of active research. Emphasis is laid on techniques which rely approximation models which may substitute for the exact and costly evaluation tool. Possible solution to this problem is the use of Metamodel-Assisted Evolutionary Algorithms (MAEAs) or other methods. MAEAs rely on the smart management of calls to the exact evaluation tool and its metamodel during the evolution, leading to a considerable economy in computational cost. Metamodels serve as surrogate evaluation tools which allow for discerning an approximate ranking of all the population members within any generation. EAs may use metamodels trained on samples selected separately from the evolution, in this case, the metamodel should be updated regularly depending on the deviation between the objective function values computed on the metamodel and the exact tool. On the other hand, EAs assisted by on-line trained metamodels can be used. A locally valid metamodel is constructed on the fly for each new individual, by training it on previously evaluated neighboring individuals. Through the metamodel-based evaluation, a few promising members in each population are identified and only these are to be re-evaluated on the exact tool. This is referred to as the Inexact Pre-Evaluation (IPE) technique. Among the most frequently used metamodels, response surface methods, polynomial interpolation and various types of artificial neural networks, Gaussian processes can be found. For more information on MAEAs the reader is referred to [13], [14] and [24].

The EA software used in this diploma thesis is the EASY platform, developed by the PCOpt/NTUA. The evaluation code is a combination of the parameterization tool, the grid morpher, the CFD solver and the post processor code that computes the objective values from the results of the CFD software as presented in Fig. 1.2.

In this thesis, the geometry of the TU Berlin's compressor blade is given as a CFD grid. Before the optimization begins, the RPT generates the GMTurbo parameterization that describes the given node-based geometry. After specifying a set of the GMTurbo parameters as design variables, the optimization process starts. The sequence of tasks, of Fig. 1.2, is called for every candidate geometry. A surface grid displacement technique is also used in order to perform CFD simulations on the new, slightly different geometry. Finally, post-processors of the CFD results are programmed to compute the objective function values used by EASY.

EASY is a general purpose, high-fidelity software for the search of optimal solutions in single or multi objective problems and has been extensively used in engineering applications. It gives the freedom to the advanced user to specify every aspect of the evolution while it provides presets for the beginner. It supports single and multiple objective function approximation through Artificial Neural Networks for time consuming problems and a rich set of options that may define algorithms such as Genetic Algorithms or Evolution Strategies.

In this thesis, two metamodel-assisted optimization runs of the case presented in Section 8.2 are carried out using the same sets of design variables but in one case the constraints presented in Section 6.1 are taken into consideration while at the other the problem is unconstrained. Radial Basis Function networks are employed as metamodels. The MAEA is based on the Inexact Pre-Evaluation technique. Metamodels are built for each new candidate solution separately using neighbouring data collected during the preceding generations of the EA; therefore, the activation of the IPE is preceded by two generations based exclusively on the exact evaluation tool. After the first two generations have been completed and stored, in each generation the offspring population members are approximated based on a local metamodel and a few of the best among them, based on the metamodel prediction, are re-evaluated on the CFD tool. The metamodels used in this thesis are further accelerated by exploiting the Principal Component Analysis (PCA) of the elite members of the evolving population. PCA is used to guide the application of evolution operators and to train the metamodels faster than other methods. On-line trained metamodels are used. Finally, the optimizations carried out are two-objective, as it has already been presented in Chapter 2, hence, the way fitness score should be assigned to the individuals is depending on the each objective score. This is done using the Strength Pareto Evolutionary Algorithms (SPEA) 2, implemented in EASY. The fitness score depends on the number of individuals dominated and the number of individuals in a neighbourhood is taken into consideration during fitness score assignment to encourage Pareto front spreading on the objective's space.

5.2 Gradient-Based Methods – Continuous Adjoint

Stochastic search optimization methods, such as Evolutionary Algorithms, require a large number of function evaluations. The computational cost of those methods scales with the number of design parameters. On the contrary, gradient-based optimization can be used in order to have significantly fewer function calls making this method computationally less expensive. In the adjoint method, there is also another set of Partial Differential Equations, the adjoint equations and each adjoint variable corresponds to a variable of the flow field. Solving the adjoint equations has a comparable cost to solving the primal state equations and the gradients can be computed from the primal and adjoint fields with minor additional calculations compared to the CFD iterations. Practically, the gradient evaluation cost does not scale with the number of design parameters. Adjoint methods are divided into two major subcategories, the discrete and the continuous [15]. In the continuous approach, the adjoint equations are derived analytically from the flow equations and are then discretized and solved numerically. The adjoint equations are derived by adding the volume integral of the product of the adjoint variables with the state equations to the objective function. By formulating and using the Green-Gauss theorem, the final expression of the adjoint flow equations and the boundary conditions are determined [2].

In continuous adjoint the augmented objective function F_{aug} is introduced and defined as:

$$F_{aug} = F + \int_{\Omega} \Psi_n R_n d\Omega + \int_{\Omega} \tilde{v}_{\alpha} R_{\bar{\mu}} d\Omega \quad (5.1)$$

where Ψ_n ($n = 1, \dots, 5$) are the mean flow adjoint variables and \tilde{v}_{α} the adjoint turbulence model variable. F is the objective function and b_i ($i = 1, \dots, N$) the design variables. Both Ψ_n and \tilde{v}_{α} act as Lagrange multipliers since they multiply the equality constraints of the flow equations in the problem of minimizing F_{aug} . Upon convergence of the flow equations, $R_n = 0$, $R_{\bar{\mu}} = 0$ and $F_{aug} = F$. Consequently, the sensitivity derivatives can be computed from $\frac{\delta F_{aug}}{\delta b_i}$ as follows.

$$\frac{\delta F_{aug}}{\delta b_i} = \frac{\delta F}{\delta b_i} + \int_{\Omega} \Psi_n \frac{\partial R_n}{\partial b_i} d\Omega + \int_{\partial\Omega} \Psi_n R_n \frac{\delta x_k}{\delta b_i} n_k dS + \int_{\Omega} \tilde{v}_{\alpha} \frac{\partial R_{\bar{\mu}}}{\partial b_i} d\Omega + \int_{\partial\Omega} \tilde{v}_{\alpha} R_{\bar{\mu}} \frac{\delta x_k}{\delta b_i} n_k dS \quad (5.2)$$

Term $\frac{\delta F}{\delta b_i}$ can be developed after having defined the objective function F . After differentiating all terms resulting from the mean flow and turbulence model PDEs, during the mathematical development of $\frac{\delta F_{aug}}{\delta b_i}$, volume integrals containing the partial derivative $\frac{\partial Q_n}{\partial b_i}$, where Q_n ($n = 1, \dots, 6$) with $Q_m = U_m$ for ($m = 1, \dots, 5$) and $Q_6 = \tilde{v}$ are the flow variables, arise. These integrals will be collected to one and the factor multiplying the total derivative $\frac{\delta Q_n}{\delta b_i}$ will be set equal to zero. By doing so, a new set of PDEs, called field adjoint equations (FAE), arise. The total and the partial derivative are linked through:

$$\frac{\delta Q_n}{\delta b_i} = \frac{\partial Q_n}{\partial b_i} + \frac{\partial Q_n}{\partial x_k} \frac{\delta x_k}{\delta b_i} \quad (5.3)$$

From Eq. 5.3, it can be noticed that changing $\frac{\delta Q_n}{\delta b_i}$ with $\frac{\partial Q_n}{\partial b_i}$ and eliminating the factors multiplying $\frac{\partial Q_n}{\partial b_i}$ leads to the same FAE. Upon convergence of the FAE, the extra term, arising from the last term of Eq. 5.3, vanishes since $\frac{\partial Q_n}{\partial x_k} \frac{\delta x_k}{\delta b_i}$ is multiplied by the FAE themselves. The mean flow field adjoint equations are:

$$-A_{nmk} \frac{\partial \psi_n}{\partial x_k} - K_m + K_m^{SA} + B_m + B_m^{SA} = 0 \quad (5.4)$$

where the terms K_m and K_m^{SA} result from the differentiation of the mean-flow viscous terms and the differentiation of the turbulence model Spalart-Allmaras [19] which is used.

Satisfaction of the FAE leads to elimination of the volume integrals associated with high computational cost. Similar approach is followed for the surface integrals leading to the introduction of adjoint boundary conditions (ABC). However, for surface integrals the factors multiplying strictly the total derivatives of Q_n must be set equal to zero. By using the adjoint boundary conditions for the mean flow and the turbulence model, the system of Eq. 5.4 and the turbulence model adjoint equations can be solved. Finally, the second and the fourth integral of Eq. 5.2 do not contain any variation in flow quantities and consequently, contribute to the expression of sensitivity derivatives. For further details on this topic, the reader is referred to [1], [2], [20], [21], [22] and [23].

In this thesis, the continuous adjoint method, developed by the PCOpt/NTUA, is used to produce the surface sensitivity map of the blades, which provides the potential changes in the shape of the blade, and to optimize the blade shape without imposing manufacturing constraints. The continuous adjoint method is included in the GPU-enabled solver PUMA.

Chapter 6

Implementation of Constraints

6.1 Presentation of Manufacturing Constraints

The following manufacturing constraints are taken into account:

- The number of blades is fixed to $n = 15$.
- The axial chord of the blade should be kept constant in every spanwise position.
- The casing end wall cannot be changed.
- The thickness of the blade should satisfy the following conditions: 1) The minimum value for LE and TE circle radius is 1 mm . 2) The two holes for the fixture in the middle of the blade have a radius of 2.5 mm and a depth of 20 mm . The blade thickness at these positions has to accommodate a cylinder of material with a radius of 5 mm and a depth of 20 mm to allow cutting of the thread at both hub and casing. The two holes can be placed arbitrarily inside the profile shape, but should be at least 60 mm apart from each other. The above thickness conditions are given in Fig. 6.1 and 6.2.
- The blade has to be mountable on a plate of dimensions $200\text{ mm} \times 80\text{ mm}$ as a part of the cylindrical casing, shown in Fig 2.3. On the hub there is no location or dimensional constraint for the fixture.
- The reduction in the radius due to the hub contouring has to be 5 mm or less and the increase of the radius due to the hub contouring has to stay below 10 mm .

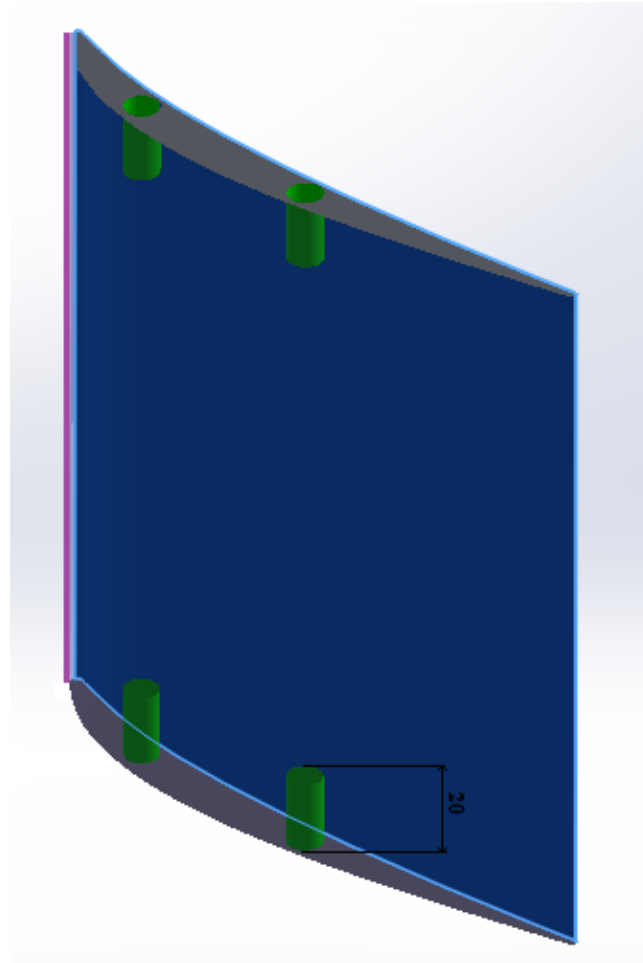


Figure 6.1: 3D view of the blade, depth of 20 mm will allow cutting of the thread at both hub and casing.

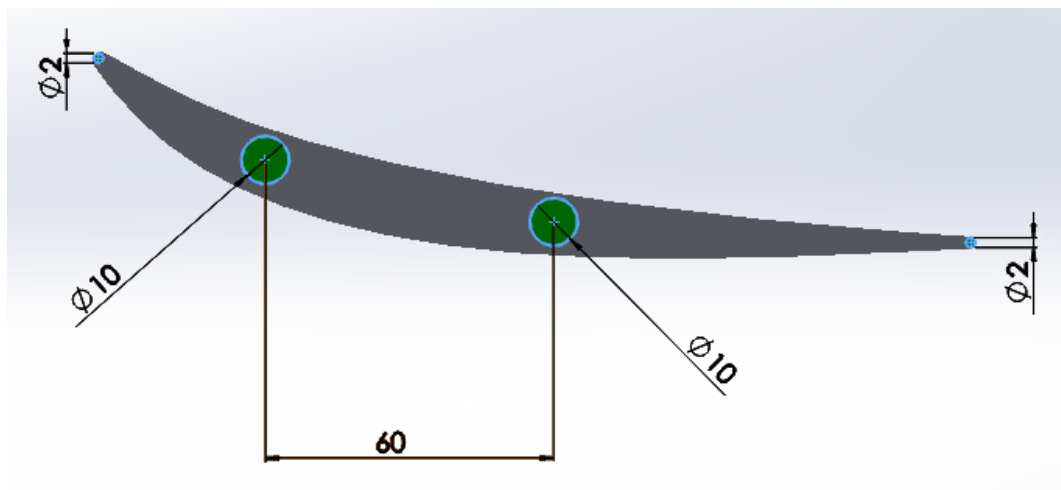


Figure 6.2: 2D view of the blade, minimum value for LE and TE circle radius is 1 mm, the blade thickness at the positions where the two holes in the middle of the blade will be created has to accommodate a cylinder of material with a radius of 5 mm. The two holes should be at least 60 mm apart from each other.

6.2 Inclusion in the EA Optimization Process

The fixed number of blades, $n = 15$, is fulfilled by the outlet boundary condition of the flow solver. The stator is adjusted to achieve a mass flow of 9.5 kg/s , full annulus. By taking the number of blades into consideration, the outlet condition of mass flow of the flow solver is: $9.5/15 = 0.6333 \text{ kg/s}$

The axial chord of the blade is kept constant by the parameterization of the GMTurbo which enables the user to fulfill such conditions.

The casing end wall will not be changed during the optimization process.

The minimum value for LE and TE circle radius of 1 mm is also fulfilled by the GMTurbo parameterization which enables the user to define the shape of the LE and the TE as presented in Section 4.1. In this thesis the circular arc edge is used where the two sides are linked with a circular arc with specified radius. The generated airfoil sides are extended and, then, a part of them is truncated in the process of modifying the edge shape, fitting a circle of specified radius between them. The radius of the arc is specified as a spanwise distribution.

The constraint that demands the blade thickness, at the positions where the holes will be created, to accommodate a cylinder of material with a radius of 5 mm and a depth of 20 mm and distance between the two holes at least 60 mm is fulfilled using the process described below:

The range of the thickness factors for both PS and SS is defined in order to fit holes with at least 5 mm radius as explained in Section 8.1.

Additionally, a code is programmed that creates a grid inside the blade. Then it creates straight lines in the direction of the LE in the whole surface of the blade that start from both the hub and shroud as seen in Fig. 6.3 and 6.4. LE direction is defined as the line that connects the tip of the LE at the hub with the tip of the LE at the shroud and can be seen as a purple line in Fig. 6.1.

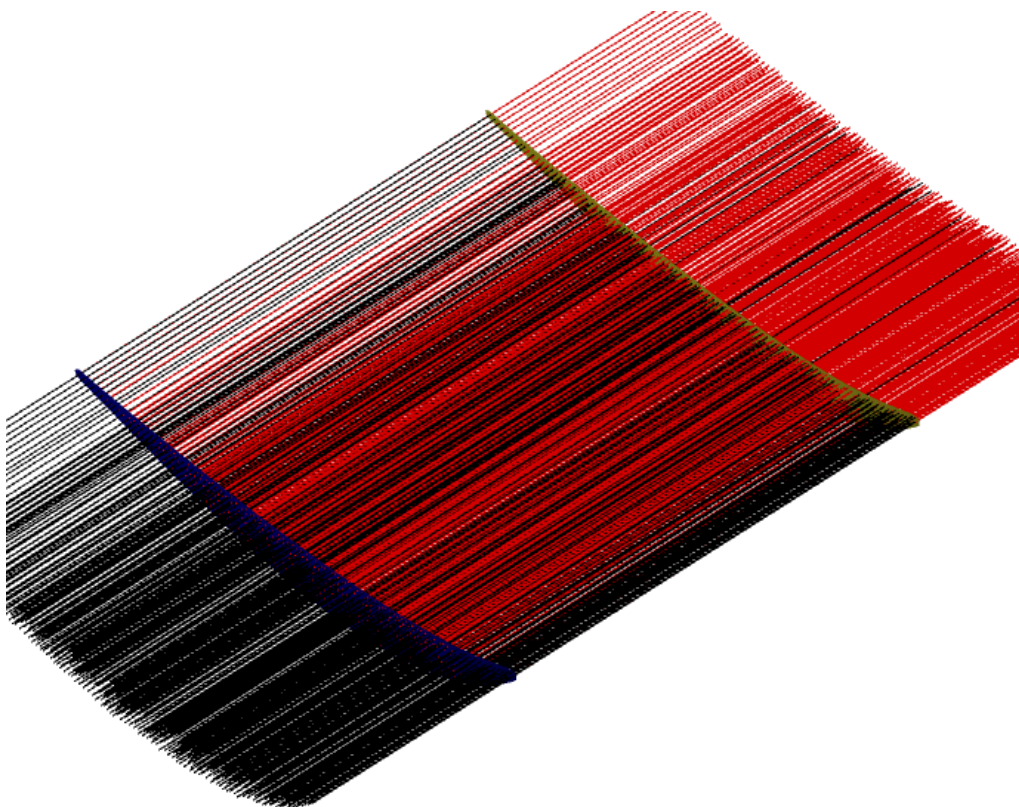


Figure 6.3: The red lines start from the hub (blue shade) towards the shroud (yellow shade) and the black lines start from the shroud towards the hub.

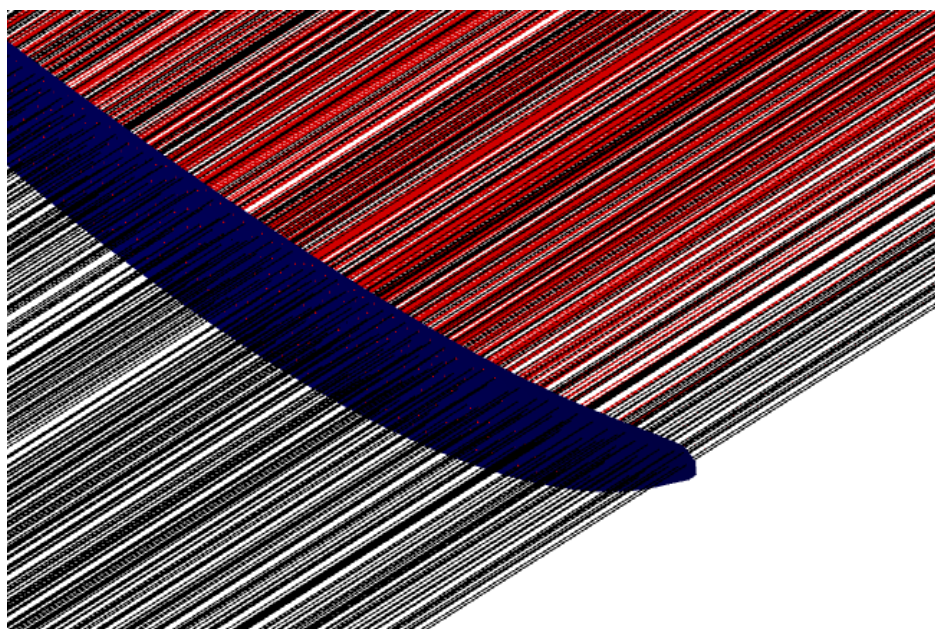


Figure 6.4: The lines intersect the surface.

These lines that intersect the blade have different lengths and constitute the depth of the holes measured either from the hub or the shroud. As a result, different areas of the blade can be intersected by lines of different lengths, that is, depths of holes as seen in Fig. 6.5.

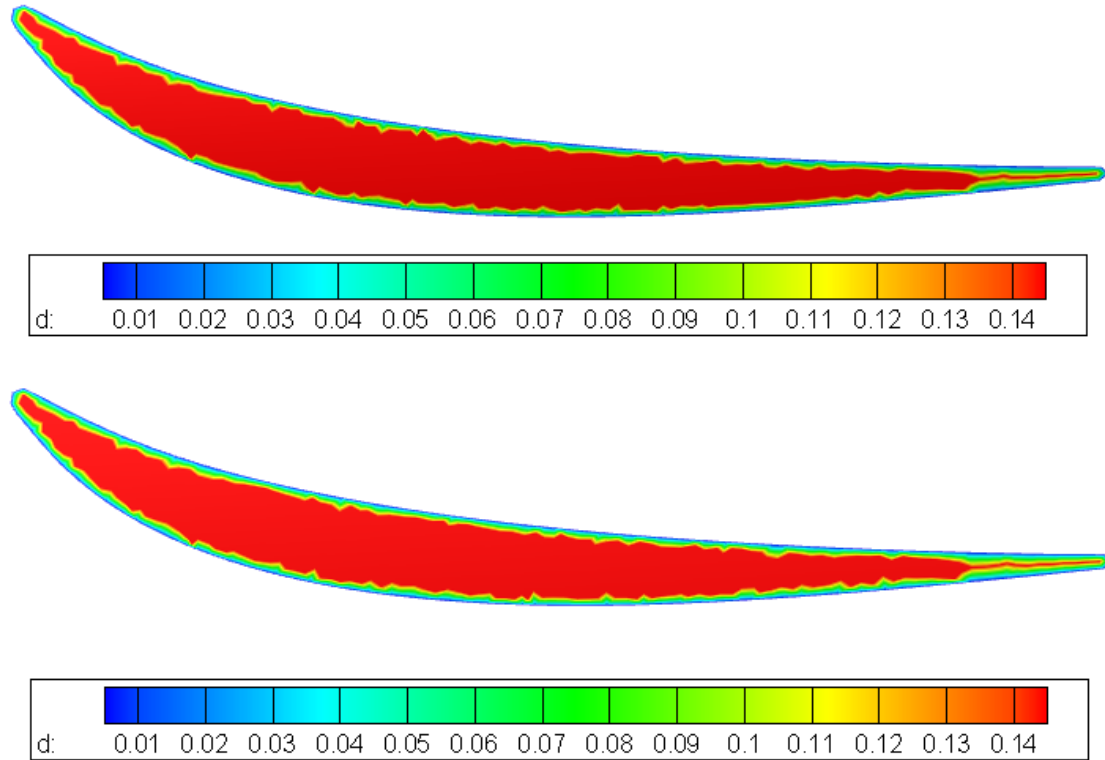


Figure 6.5: Isoareas of the depth of holes d in m for hub (top) and shroud (bottom).

From Fig. 6.5 can be seen that for 20 mm depth for holes the surface that can be created is surrounded by the light blue area. The area of the appropriate depth is extracted in (m, θ) coordinates consisting of isolines for the mean camber line, PS and SS. These points are interpolated by spline curves in order to extract 300 points for each curve. Furthermore, the coordinates are converted to (x, y, z) coordinates using Eq. A.11 of Appendix A where $r(u) = r_{HUB} = 0.1475\ m$ for the hub and $r(u) = r_{SHROUD} = 0.2975\ m$ and the distance between the points of PS and SS is calculated as seen in Fig. 6.6 for the initial geometry of the hub. The s direction is the direction of the mean camber line. This distance has to be at least 10.5 mm as a safety factor of 5 % is applied.

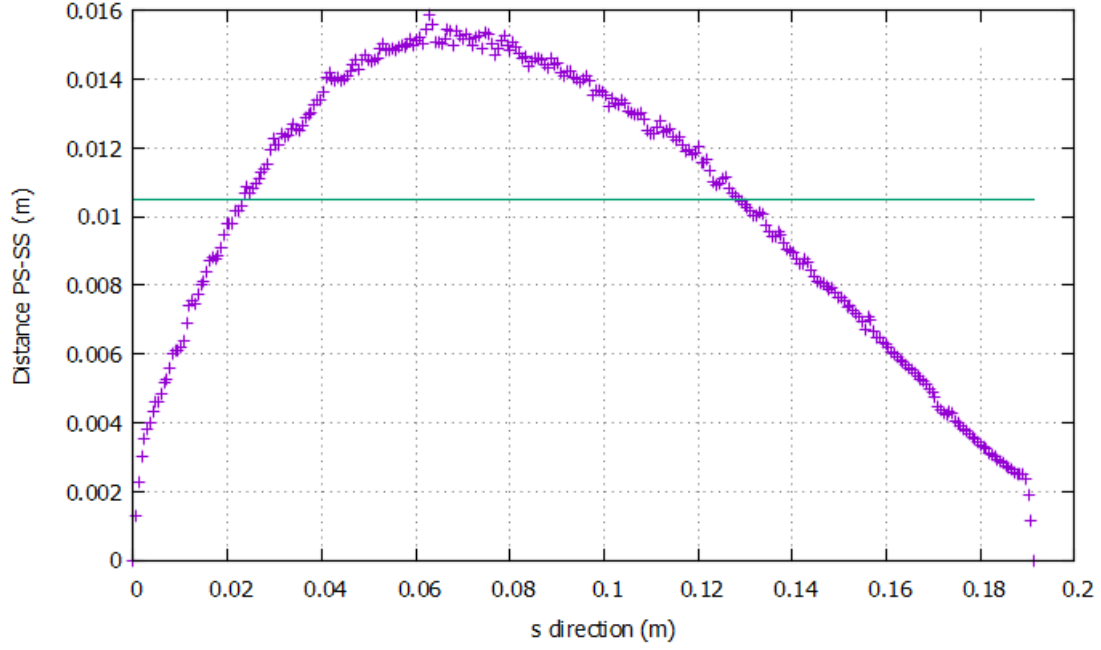


Figure 6.6: Distance between PS and SS for the initial geometry of hub (purple points) and limit of 10,5 *mm* (green line) defined by the constraint.

Finally, the distance between the second and the second to last point of the Distance PS-SS (*m*) - *s* direction (*m*) diagram, that fulfills the distance of 10.5 *mm* constraint, is calculated and has to be at least 60 *mm*.

The constraint that demands the blade to be mountable on a plate of dimensions 200 *mm* × 80 *mm* as a part of the cylindrical casing is fulfilled when defining the range of the *z* variable for both LE and TE and the range of θ_{LE} , θ_{TE} , β_{LE} , β_{TE} , δ_{LE} and δ_{TE} as defined in Section 8.1.

The radius of the hub is not changed since hub contouring is not used. The radius of the hub is defined by the GMTurbo parameterization which enables controlling of the hub generatrix.

Using the techniques described above, all manufacturing constraints of the case are taken into consideration.

Chapter 7

CFD Analysis of TU Berlin's Case - Validation

In order to analyze the CFD results and validate them against the experimental data, post processing codes have been programmed and used.

7.1 Mach number

The absolute Mach number is given as follows:

$$M = \frac{V}{c} \quad (7.1)$$

where c is the speed of sound in the medium and V is the velocity of the flow.

A comparison of the inlet Mach number computed by the PUMA solver for compressible flows without preconditioning and experimental data, for all three inlet whirl angles, 42° , 47° and 37° , is given in Fig. 7.1. The radius r in mm , spanning from hub to shroud, is given in the horizontal axis. The inlet whirl angle will be given as α_{in} .

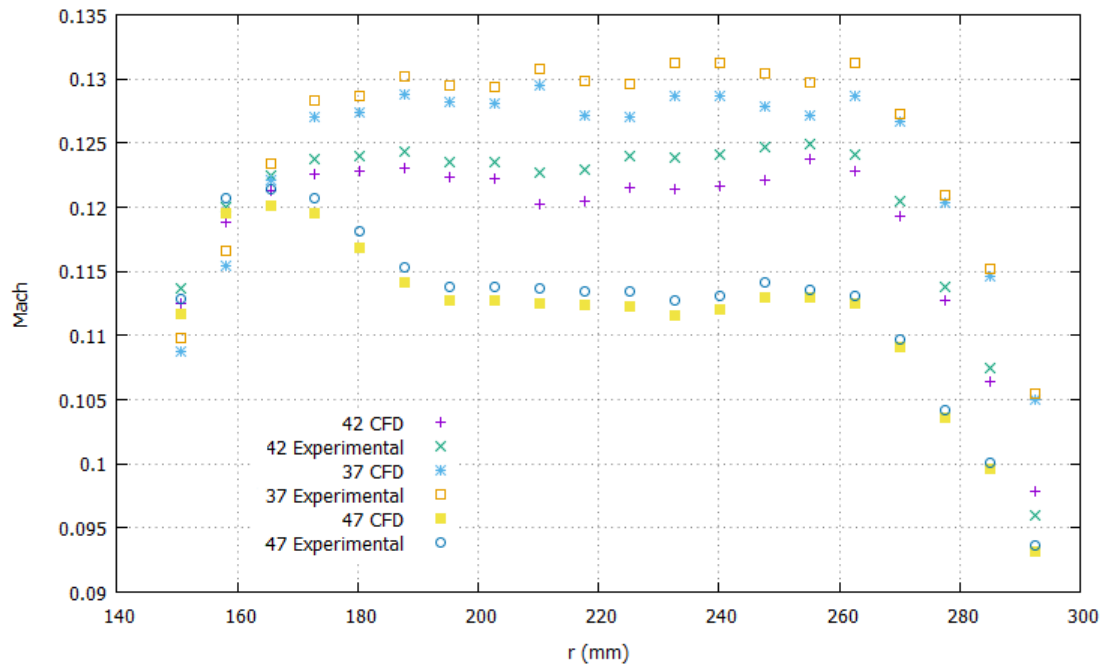


Figure 7.1: Comparison of CFD and experimental inlet Mach number, from [16], for 42° , 47° and 37° inlet whirl angle.

A comparison of the inlet Mach number for $\alpha_{in} = 42^\circ$ computed by the PUMA solver for compressible flows, with and without low Mach preconditioning, is given in Fig 7.2.

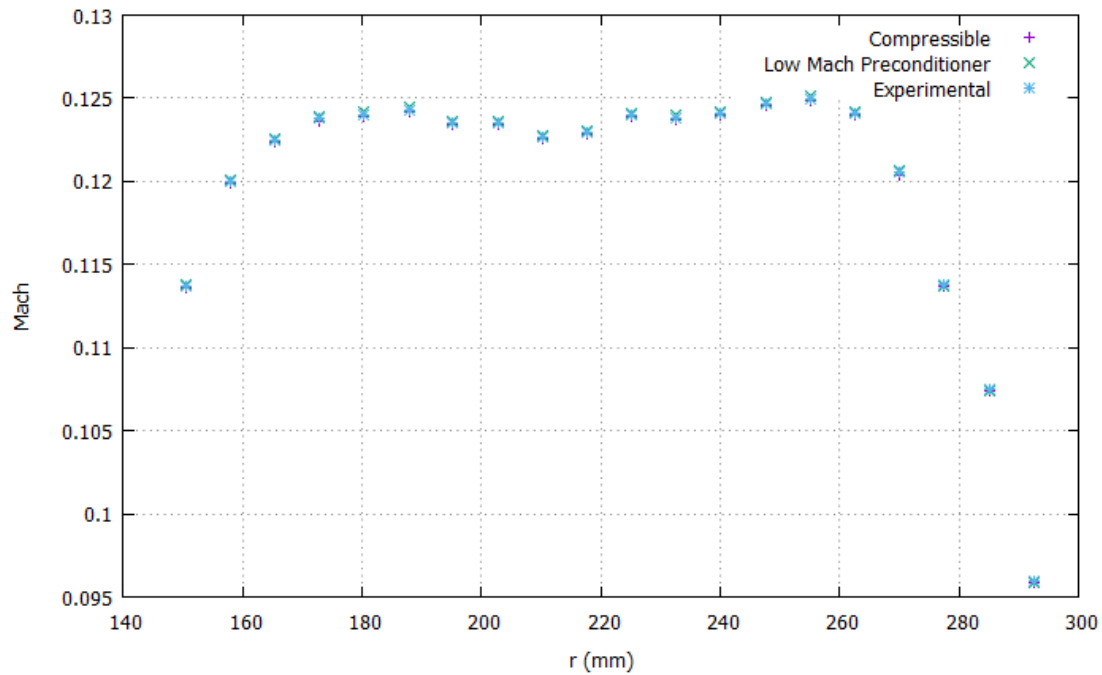


Figure 7.2: Comparison of inlet Mach number for $\alpha_{in} = 42^\circ$ as computed by the PUMA code for compressible flows, with and without low Mach preconditioning.

A comparison of the outlet Mach number computed by the PUMA solver for compressible flows without preconditioning and experimental data for all three inlet whirl angles, 42° , 47° and 37° , is given in Fig. 7.3.

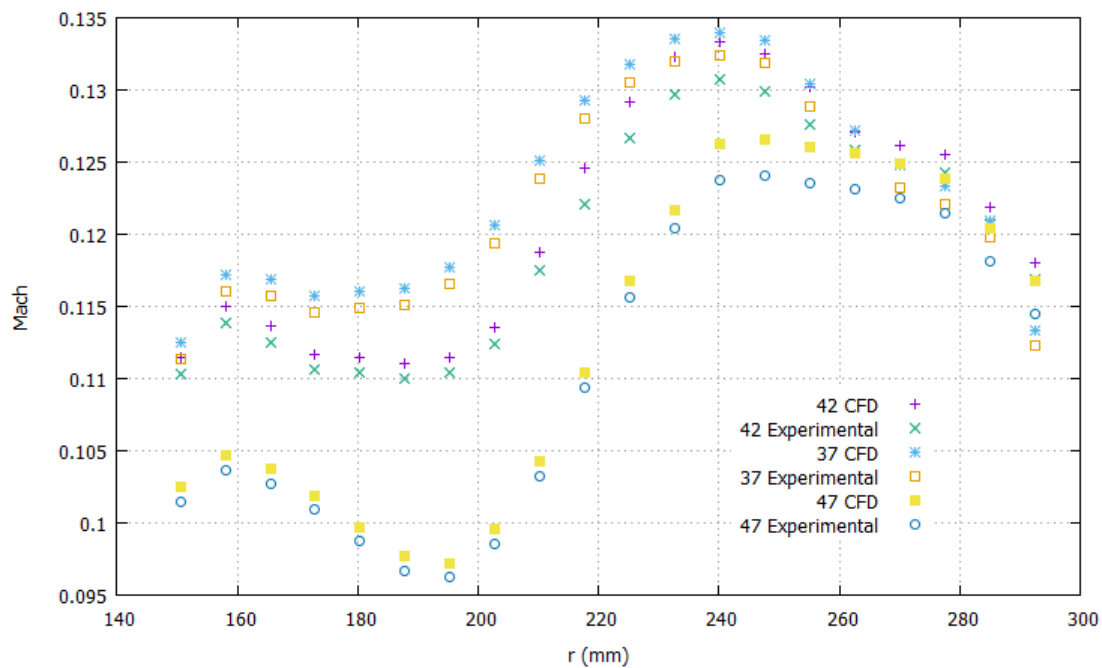


Figure 7.3: Comparison of CFD and experimental outlet Mach number, from [16], for 42° , 47° and 37° inlet whirl angle.

A comparison of the outlet Mach number for $\alpha_{in} = 42^\circ$ computed by the PUMA solver for compressible flows, with and without low Mach preconditioning, is given in Fig. 7.4.

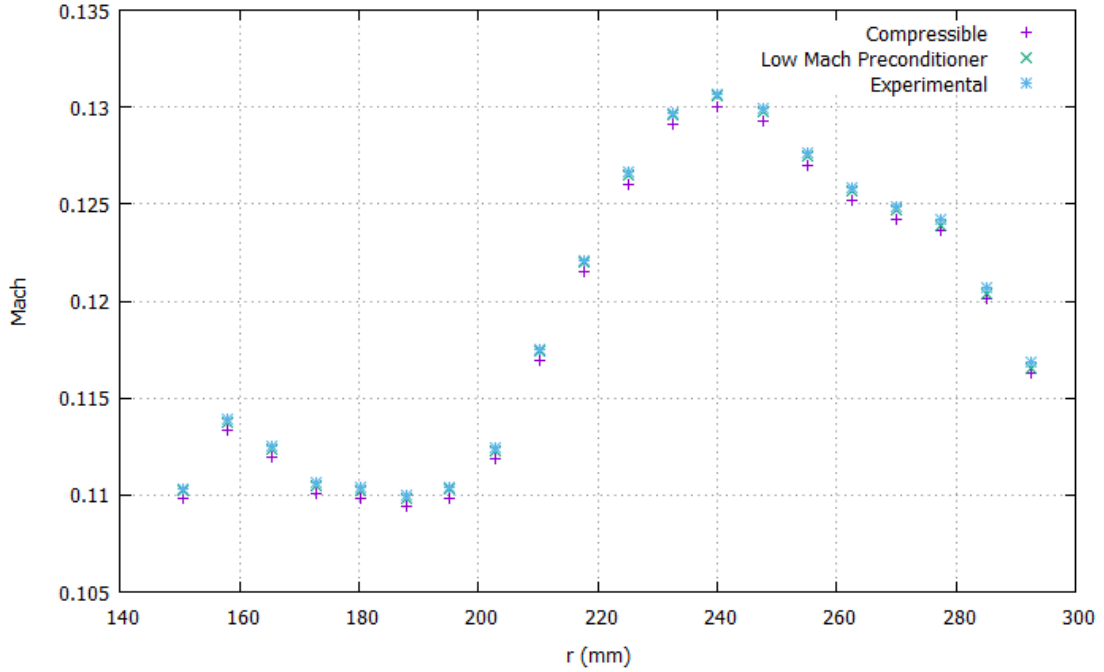


Figure 7.4: Comparison of outlet Mach number for $\alpha_{in} = 42^\circ$ as computed by the PUMA code for compressible flows, with and without low Mach preconditioning.

In Fig. 7.1 for the inlet and Fig. 7.3 for the outlet, it can be seen that the CFD results from the PUMA compressible flow solver are very close to the measurements with a relative error less than 1 % for all three inlet whirl angles. The relative error is defined as:

$$\varepsilon_r = \frac{\text{Experimental} - \text{CFD}}{\text{Experimental}} \cdot 100\% \quad (7.2)$$

It can be seen that in all three inlet whirl angle cases, Mach number is lower than 0.135 for both inlet and outlet. From Figs. 7.2 and 7.4, it can be seen that the values of the Mach number are very close compared the two CFD methods of compressible flow, with and without low Mach preconditioning and the experimental results in account of the fact that the Mach number of the flow is low. As a result, low Mach preconditioning can be used and the flow can be sufficiently approximated by an incompressible flow model as it will be presented in figures for whirl and pitch.

From Fig. 7.1 and 7.3 it can be seen that, in all three inlet whirl angle cases, the Mach number is decreased in low and high radius, that is near hub and shroud due to the flow deceleration that occurs as a result of the hub and shroud boundary layer and the flow mixing, as seen in Fig. 7.5. Furthermore, as the inlet whirl angle increases, Mach number decreases at the same radius as a result of the lower velocity.

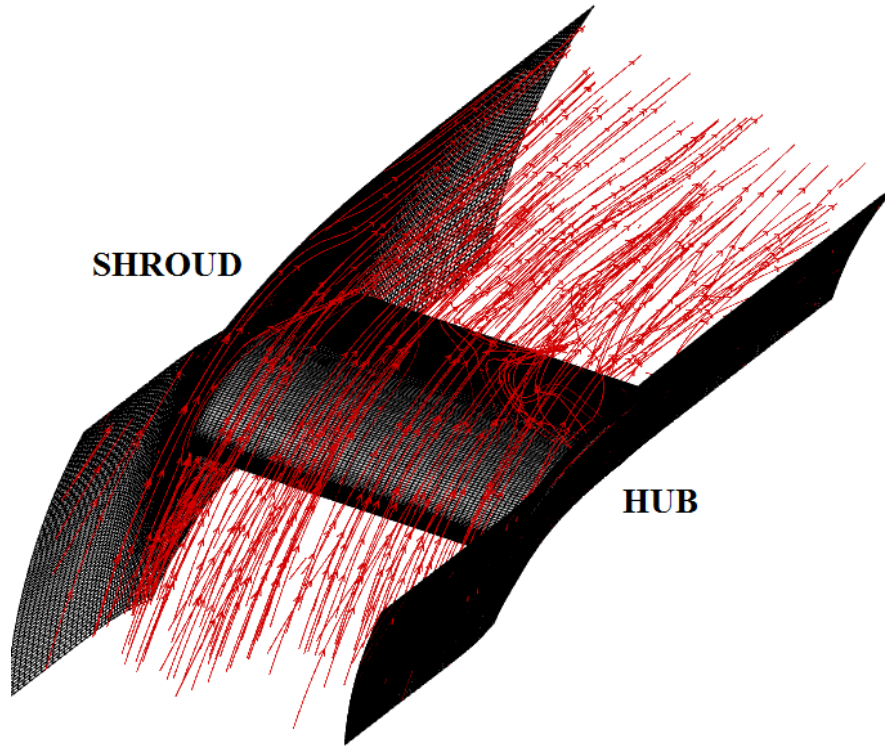


Figure 7.5: Stream traces of flow.

A comparison of the outlet Mach number for $\alpha_{in} = 42^\circ$ computed using the code for compressible flows, without preconditioning, for different values of viscosity ratio, is made in Fig. 7.6.

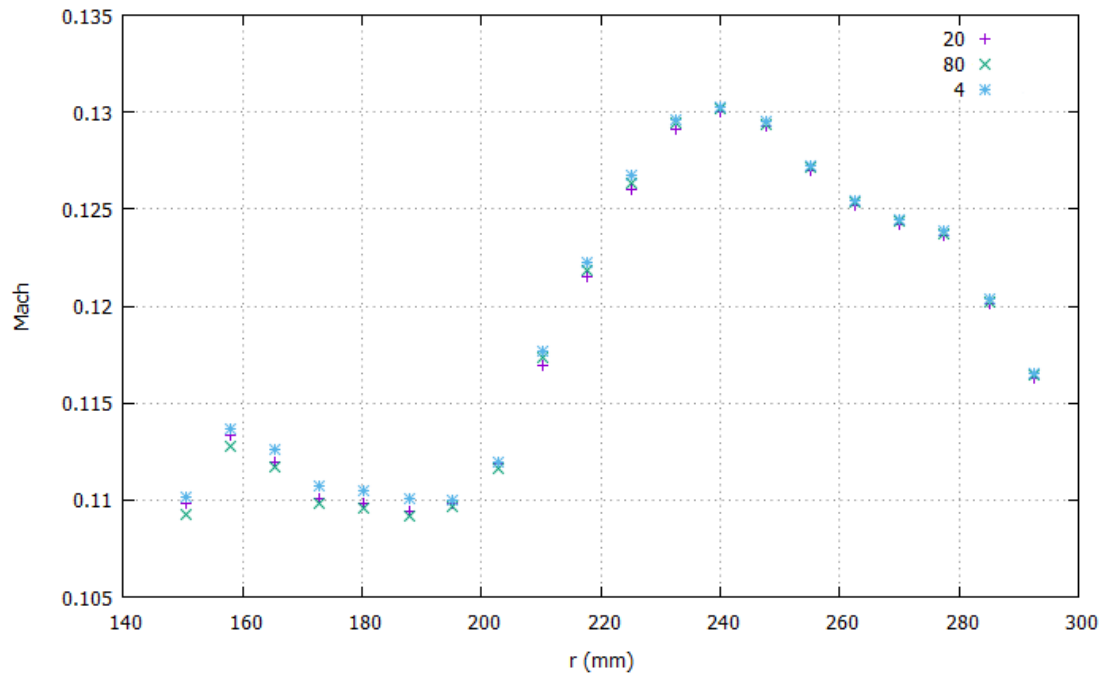


Figure 7.6: Comparison of the outlet Mach number for $\alpha_{in} = 42^\circ$ using a compressible flow model, for different values of viscosity ratio, at the inlet.

The value of viscosity ratio ν_t/ν that is needed for the inlet boundary conditions for the k- ω SST turbulence model is unknown, hence, a comparison of the outlet Mach number is made for three values of viscosity ratios as seen in Fig. 7.6. The minor differences of Mach number values between the three viscosity ratio scenarios lead to the conclusion that the case is almost independent from the inlet viscosity ratio. Viscosity ratio equal to 20 is used in all CFD cases.

7.2 Outlet Whirl angle α

The task of the stator is to turn the incoming flow with a whirl angle of $42 \pm 5^\circ$ into axial flow that is, a flow that has an outlet whirl angle value very close to 0° . The whirl angle is calculated through:

$$\alpha = \cos^{-1} \left(\frac{\vec{v} \cdot \vec{i}}{|\vec{v}|} \right) = \cos^{-1} \left(\frac{u}{\sqrt{u^2 + v^2 + w^2}} \right) \quad (7.3)$$

where (u, v, w) are the three Cartesian velocity components.

A comparison of the outlet whirl angle computed by the PUMA solver for compressible flows without preconditioning and experimental data for all three inlet whirl angles, 42° , 47° and 37° , is given in Fig. 7.7.

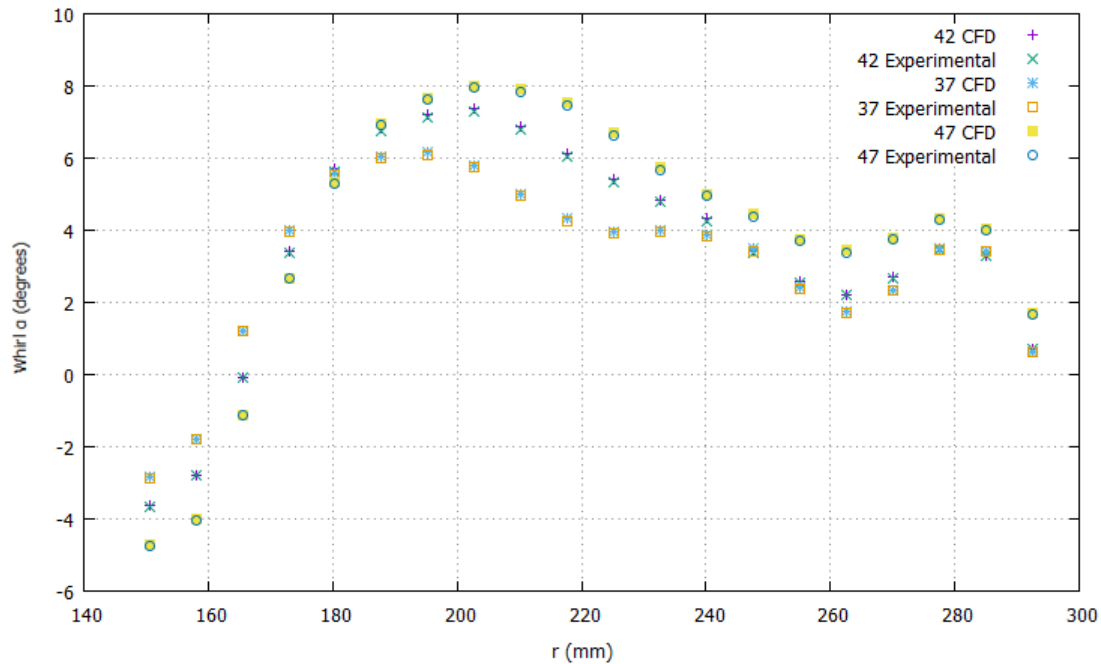


Figure 7.7: Comparison of CFD and experimental outlet whirl angle α , from [16], for 42° , 47° and 37° inlet whirl angle.

A comparison of the outlet whirl angle for $\alpha_{in} = 42^\circ$ computed by the PUMA solver for compressible flows, with and without low Mach number preconditioning, and the corresponding code for incompressible flow is given in Fig. 7.8.

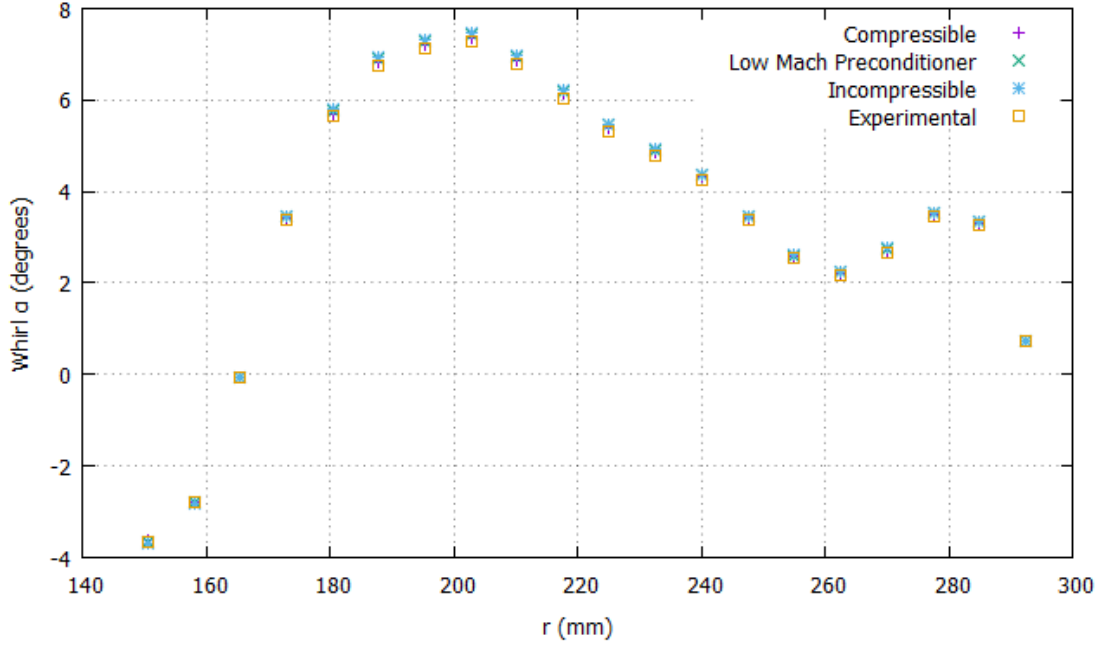


Figure 7.8: Comparison of outlet whirl angle α for $\alpha_{in} = 42^\circ$ as computed by the PUMA code for compressible flows, with and without low Mach number preconditioning, and the incompressible variant of the same code.

From Fig. 7.8, it can be seen that the outlet whirl angle distributions are very close comparing the three methods and the experimental results in account of the fact that the Mach number of the flow is low as it has already been explained. From Fig. 7.7 it can be seen that the CFD results from the PUMA Compressible flow solver are very close to the experimental results with relative error less than 1 % for all three inlet whirl angles. It can also be seen that the outlet whirl angle increases as the inlet whirl angle increases.

The outlet whirl angle varies from $(-4^\circ, 8^\circ)$ for $\alpha_{in} = 42^\circ$, $(-5^\circ, 8^\circ]$ for $\alpha_{in} = 47^\circ$ and $(-3^\circ, 6^\circ]$ for $\alpha_{in} = 37^\circ$ as the radius increases. The flow needs to be as close to axial as possible which is one of the objectives of the optimization.

7.3 Outlet Pitch angle β

The pitch angle is calculated through:

$$\beta = \tan^{-1} \left(\frac{\text{Radial Velocity}}{\text{Axial Velocity}} \right) = \tan^{-1} \left(\frac{u_r}{u_a} \right) \quad (7.4)$$

Comparison of the outlet pitch angle computed by the PUMA solver for compressible flows without preconditioning and experimental data for all three inlet whirl angles, 42° , 47° and 37° , is given in Fig. 7.9.

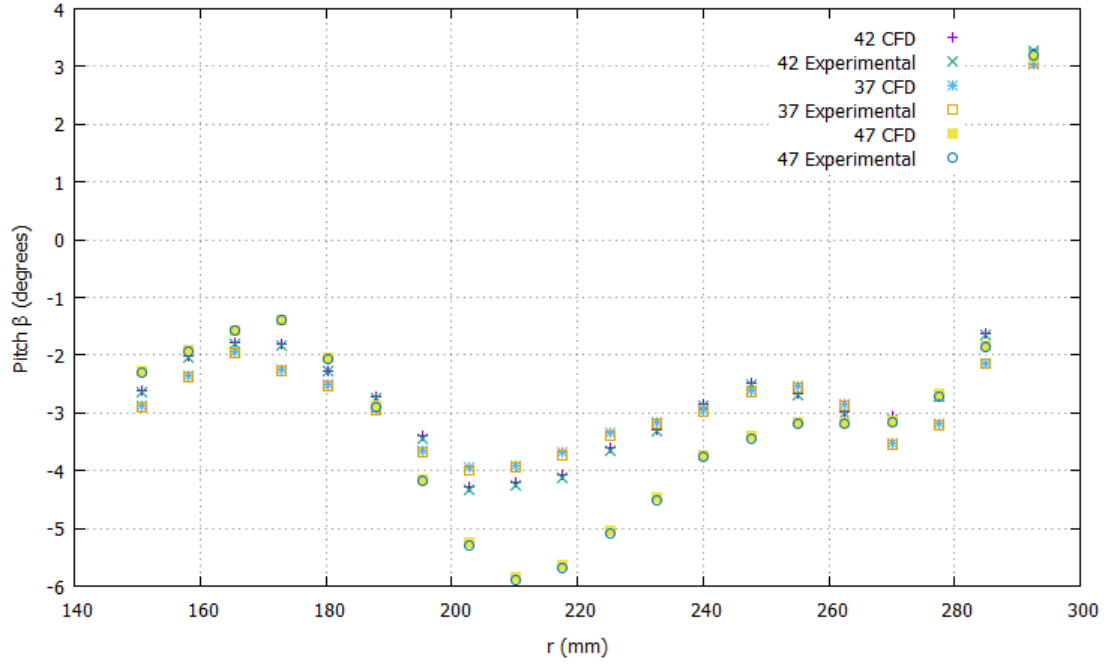


Figure 7.9: Comparison of CFD and experimental outlet pitch angle β , from [16], for 42°, 47° and 37° inlet whirl angle.

A comparison of the outlet pitch angle for $\alpha_{in} = 42^\circ$ computed by the PUMA solver for compressible flows, with and without low Mach number preconditioning, and incompressible flow is given in Fig. 7.10.

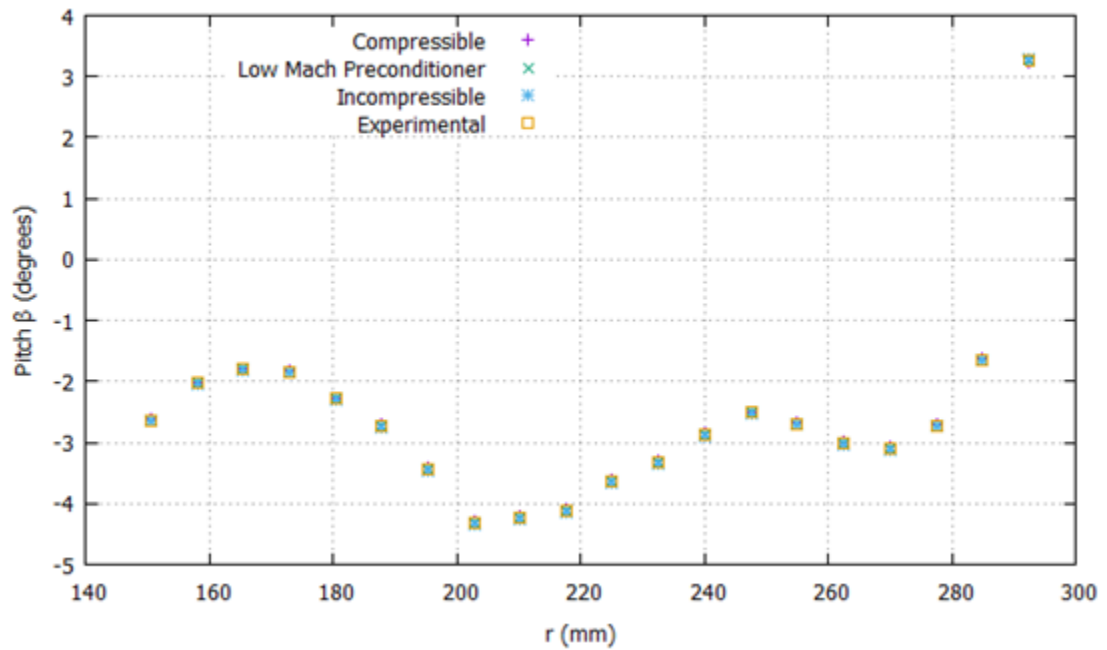


Figure 7.10: Comparison of outlet pitch angle β for $\alpha_{in} = 42^\circ$ as computed by the PUMA code for compressible flows, with and without low Mach number preconditioning, and the incompressible variant of the same code.

In Fig. 7.9 it can be seen that the CFD results from the PUMA compressible flow solver are very close to the experimental results with relative error less than 1 % for all three inlet whirl angles. The outlet pitch angle varies from $(-5^\circ, 3^\circ]$ for $\alpha_{in} = 42^\circ$, $(-6^\circ, 3^\circ]$ for $\alpha_{in} = 47^\circ$ and $(-4^\circ, 3^\circ]$ for $\alpha_{in} = 37^\circ$ as the radius increases. The pitch angle is almost constant as the radius increases but in high values of radius, near shroud, a dramatic increase in pitch angle appears as a result of the secondary flow effects.

7.4 Outlet Total Pressure and Static Pressure

The experimental data for the outlet pressure must be corrected, according to the corresponding ambient pressure value, using the following equation:

$$P_t^{corr} = P_t \cdot \frac{P_{amb}^{in}}{P_{amb}^{out}} \quad (7.5)$$

The ambient pressure for each case is:

Inlet -5° : 101290 Pa

Outlet -5° : 101570 Pa

Inlet 0° : 101240 Pa

Outlet 0° : 101530 Pa

Inlet $+5^\circ$: 101150 Pa

Outlet $+5^\circ$: 101570 Pa

A comparison of the outlet total pressure computed by the PUMA solver for compressible flows without preconditioning and experimental data for all three inlet whirl angles, 42° , 47° and 37° , is given in Fig. 7.11.

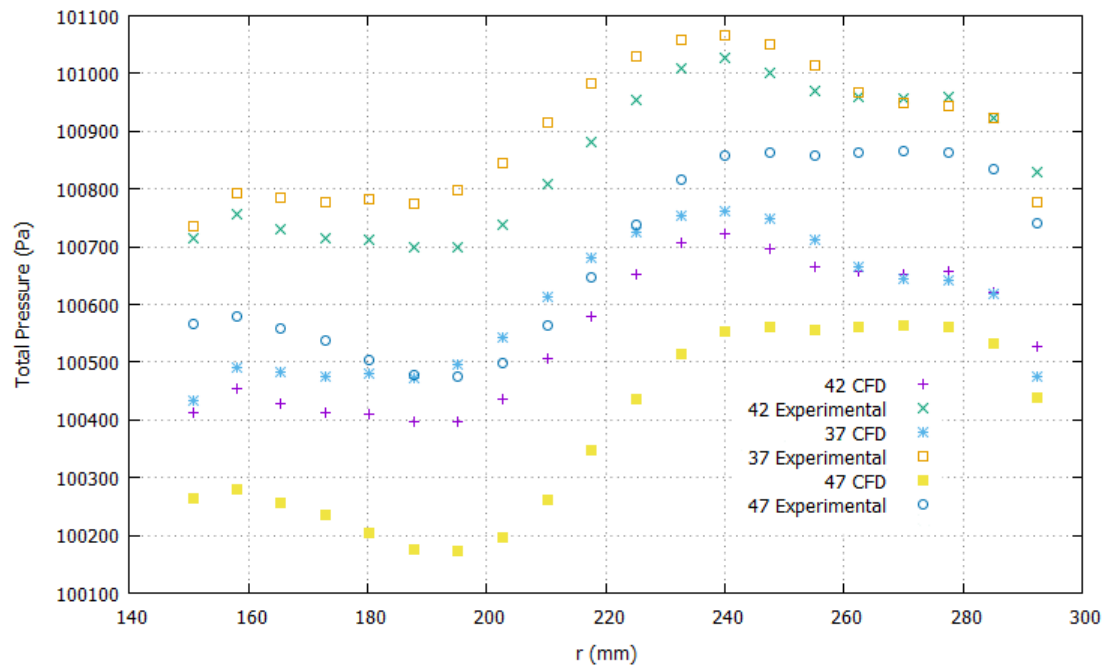


Figure 7.11: Comparison of CFD and experimental outlet P_t , from [16], for 42° , 47° and 37° inlet whirl angle.

A comparison of the outlet total pressure for $\alpha_{in} = 42^\circ$ computed by the PUMA solver for compressible flows, with and without low Mach number preconditioning, is given in Fig. 7.12.

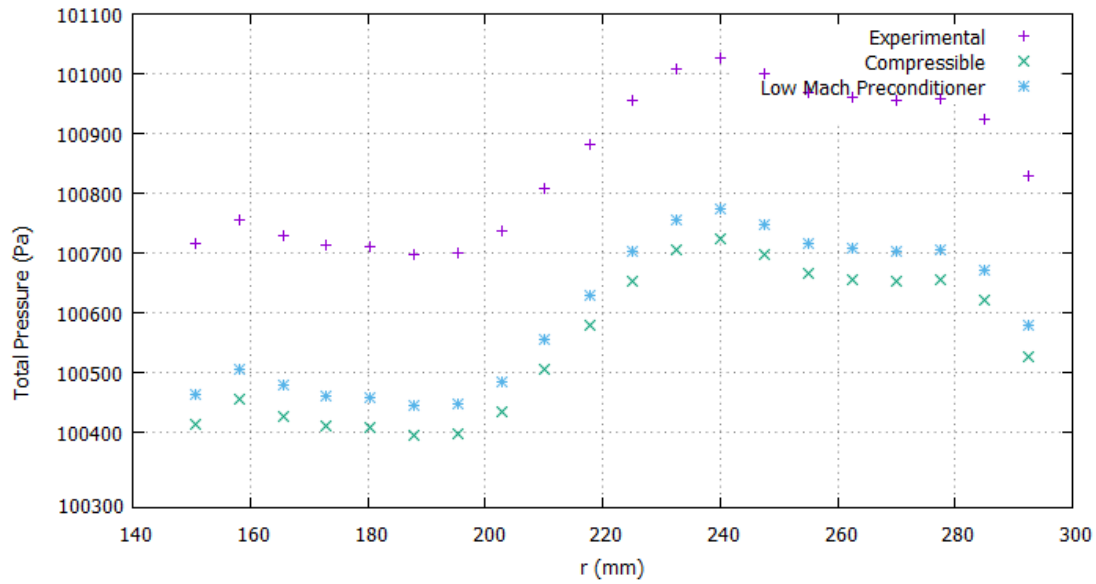


Figure 7.12: Comparison of outlet P_t for $\alpha_{in} = 42^\circ$ as computed by the PUMA code for compressible flows, with and without low Mach number preconditioning.

In Fig. 7.11 it can be seen that the CFD results from the PUMA compressible flow solver are very close to the experimental results, with relative error of 0.5 % to 1 %, for all three inlet whirl angles.

A comparison of the outlet static pressure computed by the PUMA solver for compressible flows without preconditioning and experimental data for all three inlet whirl angles, 42° , 47° and 37° , is given in Fig. 7.13.

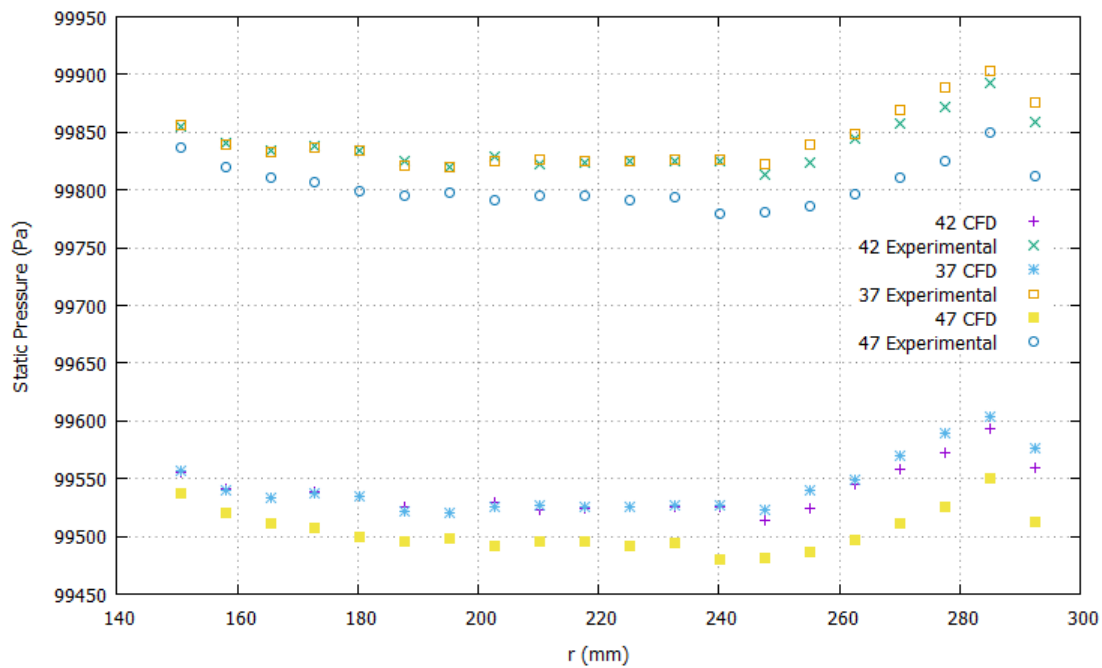


Figure 7.13: Comparison of CFD and experimental outlet P_{st} , from [16], for 42° , 47° and 37° inlet whirl angle.

A comparison of the outlet static pressure for $\alpha_{in} = 42^\circ$ computed by the PUMA solver for compressible flow, with and without low Mach number preconditioning, is given in Fig. 7.14.

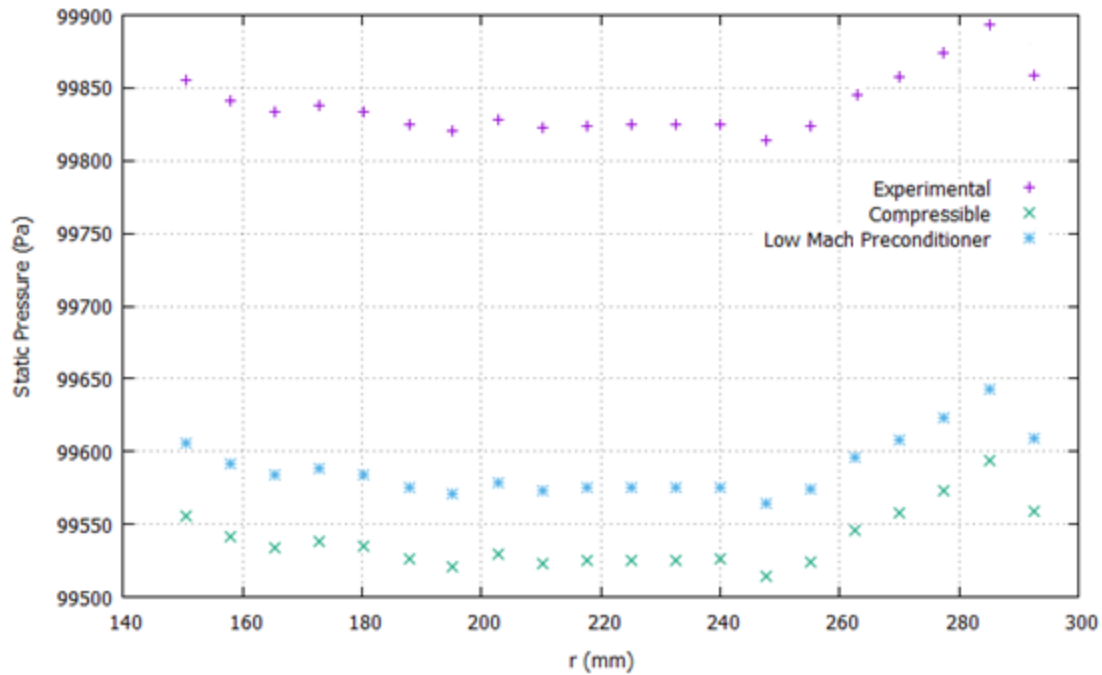


Figure 7.14: Comparison of outlet P_{st} for $\alpha_{in} = 42^\circ$ as computed by the PUMA code for compressible flows, with and without low Mach number preconditioning.

7.5 Total Pressure Loss Coefficient

Total pressure loss coefficient is a dimensionless quantity. In 3D geometries, it usually needs to be computed along isospan blade sections, from hub to shroud. It is given by Eq. 2.1. A comparison of the total pressure loss coefficient computed by the PUMA solver for compressible flows without preconditioning and experimental data for all three inlet whirl angles, 42° , 47° and 37° , is given in Fig. 7.15.

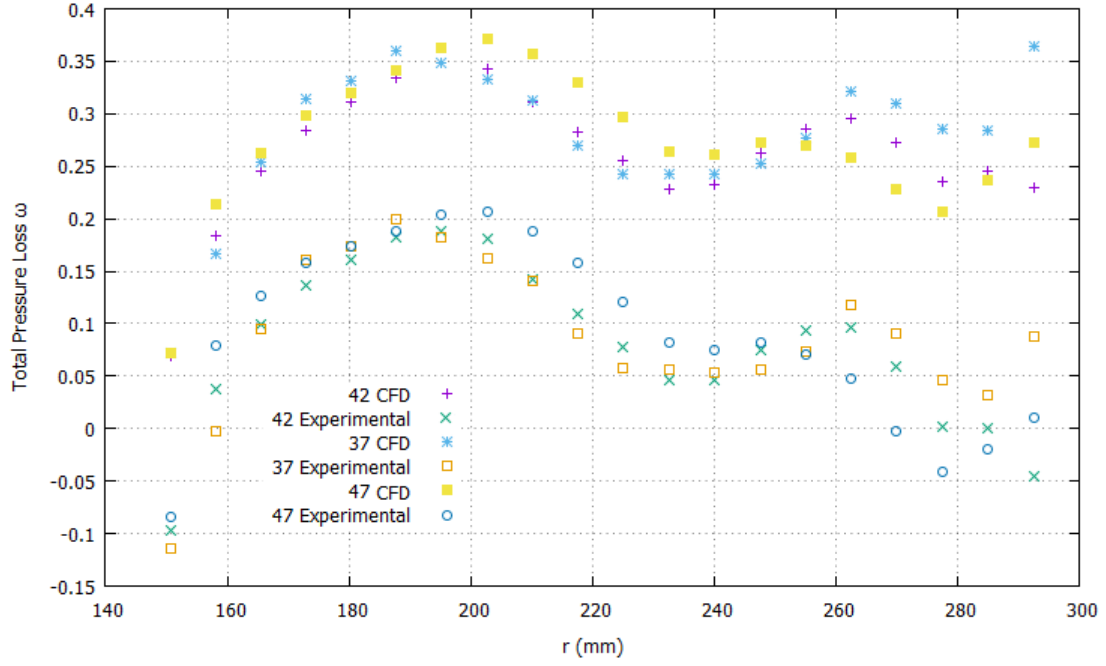


Figure 7.15: Comparison of CFD and experimental ω , from [16], for 42° , 47° and 37° inlet whirl angle.

A comparison of ω for $\alpha_{in} = 42^\circ$ computed by the PUMA solver for compressible flow, with and without low Mach number preconditioning, is given in Fig. 7.16.

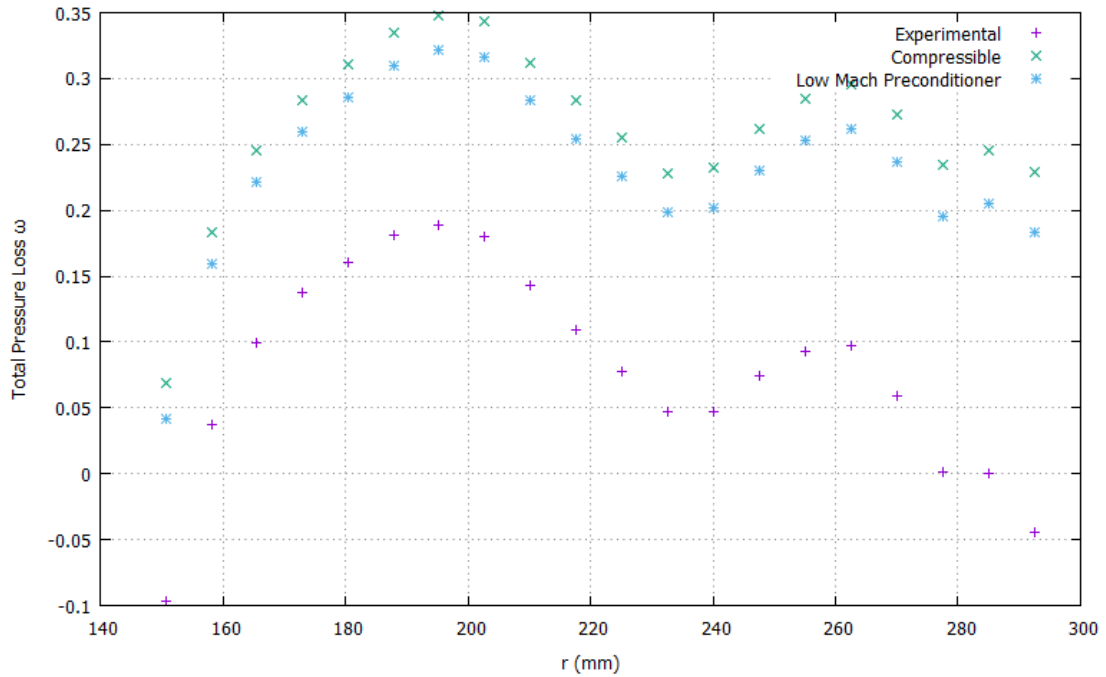


Figure 7.16: Comparison of ω for $\alpha_{in} = 42^\circ$ as computed by the PUMA code for compressible flows, with and without low Mach number preconditioning.

The total pressure loss coefficient varies from $(-0.05, 0.2)$ for $\alpha_{in} = 42^\circ$, $(-0.05, 0.2]$ for $\alpha_{in} = 47^\circ$ and $(-0.15, 0.2]$ for $\alpha_{in} = 37^\circ$ as the radius increases. The total pressure loss coefficient has to be minimized which is one of the objectives of the optimization.

Chapter 8

Evolutionary Algorithm Optimization

8.1 Design Variables

When using the RPT, presented in Section 4.2, to create the input file for the GMTurbo, patches such as the hub and shroud, the surface of which is a revolved surface around the z -axis, are given as meridional generatrices, inlet and outlet are given as meridional boundary curves, LE and TE are given as meridional edge curves and variables such as θ_{LE} , θ_{TE} , β_{LE} , β_{TE} , δ_{LE} , δ_{TE} and thickness factors for both PS and SS are given in spanwise distribution, using NURBS curves. The control points of those NURBS curves are the design variables to be determined during the optimization process. The design variables are:

- The LE and TE are given in (r, z) and three out of the seven r variables of the NURBS curve for both LE and TE are used as design variables.
- θ_{LE} , θ_{TE} , β_{LE} , β_{TE} , δ_{LE} , δ_{TE} and thickness factors for both pressure and suction side are given in spanwise distribution with five points for every NURBS curve. All five points for θ_{LE} , θ_{TE} , β_{LE} , β_{TE} , δ_{LE} , δ_{TE} and thickness factors are used as design variables.

The total number of design variables is 46.

8.2 Optimization using EASY

In this section, two optimization runs of the TU Berlin's stator case are carried out using the same sets of design variables. In one case, constraints are taken into consideration while in the other the problem is unconstrained. The objective functions, in all runs, are total pressure loss coefficient ω and the flow angle deviation at the CFD outlet from the axial direction. Both must be minimized. A $(\mu, \lambda) = (8, 12)$ EA with 8 parents and 12 offspring, assisted by metamodels performing inexact pre-evaluation of the offspring population, is used. A termination criterion of 100 evaluations on the CFD was set. Each evaluation has a total duration of approximately 4 hours and the CFD solver runs on one NVIDIA Tesla K40 GPU. The basic parameters of the metamodel-assisted optimization using EASY can be seen in Table 1.

TABLE 1 EASY Settings	
Number of Demes	1
Coding	Real
Parent population size μ	8
Offspring population size λ	12
Evaluations before termination	100
Parents that create one offspring	3
Elite population size	20
Elite individuals to force as new offspring	1
Probability to select an elite	0.08
Tournament size	3
Tournament probability	0.85
Crossover probability	0.9
Mutation probability	0.05
Metamodel type	RBF
Minimum data base size for IPE	20
Exact evaluations per generation (IPE) min-max	2-5

The front of non-dominated solutions is given in Fig. 8.1 for the unconstrained case and in Fig. 8.2 for the constrained case.

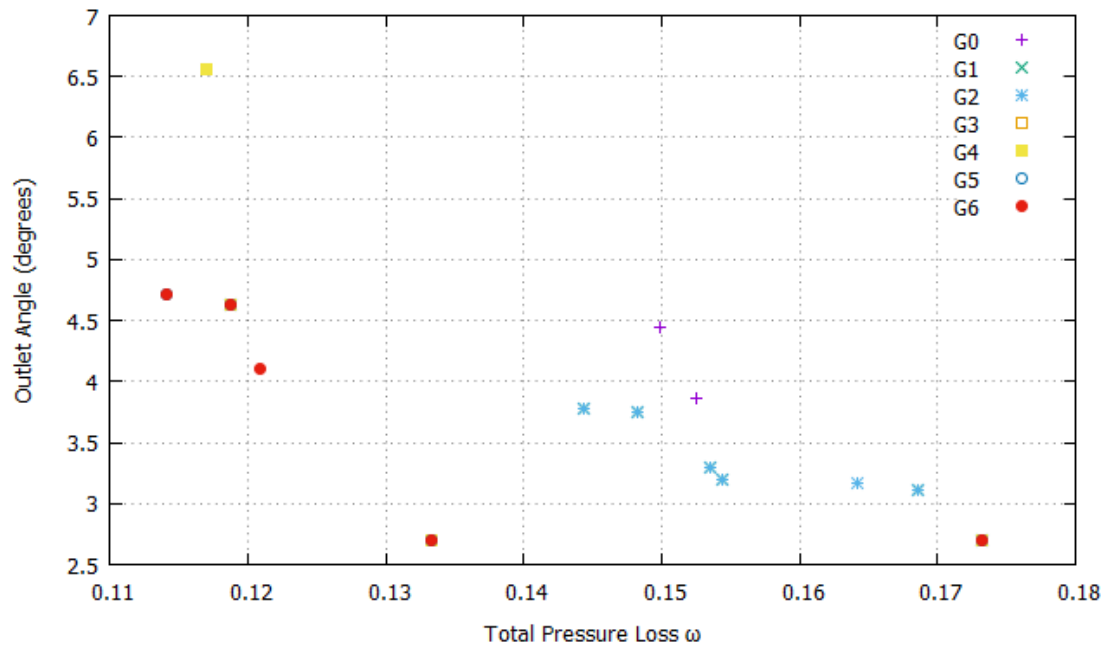


Figure 8.1 : Front of non-dominated solutions of different generations for the unconstrained case.

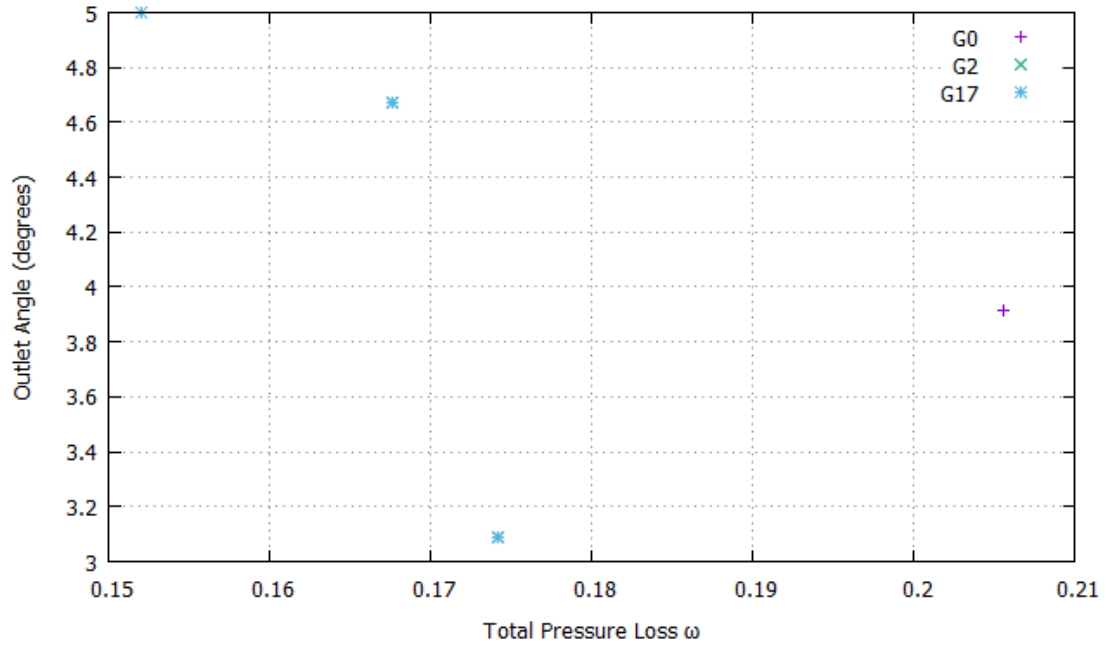


Figure 8.2: Front of non-dominated solutions of different generations for the constrained case.

The Pareto fronts for the constrained and the unconstrained optimization using MAEAs are given in Fig. 8.3. The values of the objective functions for the basic, initial, geometry can also be seen in Fig. 8.3. All the optimization runs are for the case of 42° inlet whirl angle.

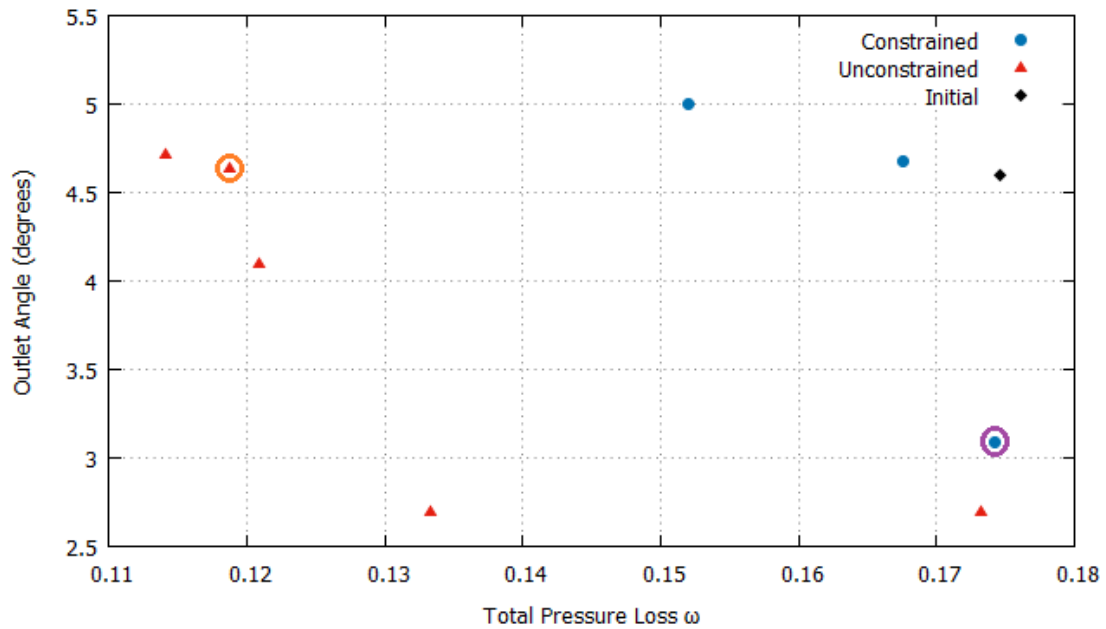


Figure 8.3: Pareto front for unconstrained and constrained case.

As far as the unconstrained case is concerned, some members of the Pareto front result in reduction of one of the objective functions while the other is increased. However, one possible solution manages to minimize both the objective functions

significantly. Similar results appear in the constrained case. The possible solution in purple circle, as seen in Fig. 8.3 achieves slight decrease of the total pressure loss coefficient and major decrease of the outlet angle. Specifically, the total pressure loss coefficient is decreased by 0.25 % and the outlet angle by 1.5° which corresponds to 32.8 %. The improvement i of the objectives is defined as:

$$i(\%) = \frac{\text{Optimized} - \text{Initial}}{\text{Initial}} \cdot 100\% \quad (8.1)$$

The resulting optimized geometry compared to the baseline for the case without manufacturing constraints can be seen in Fig. 8.4 and for the constrained optimization in Fig. 8.5.

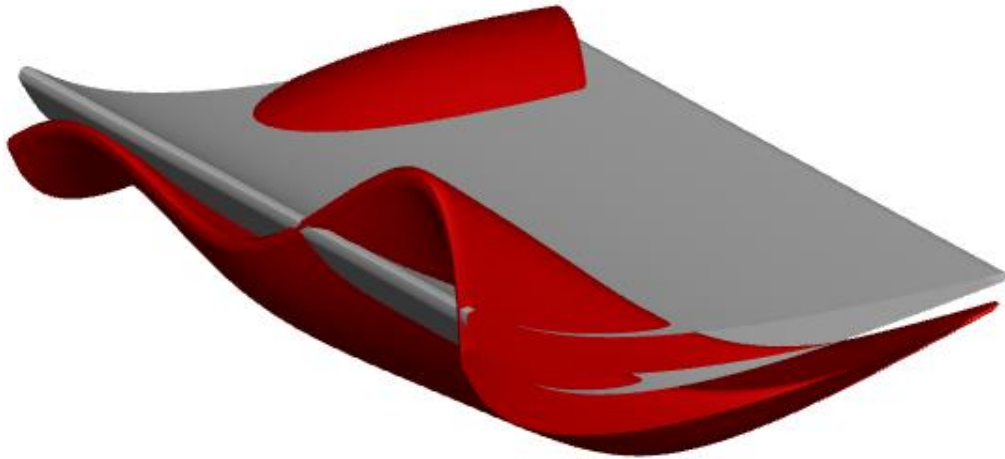


Figure 8.4: Initial (grey) and optimized without manufacturing constraints (red) shape of blade.

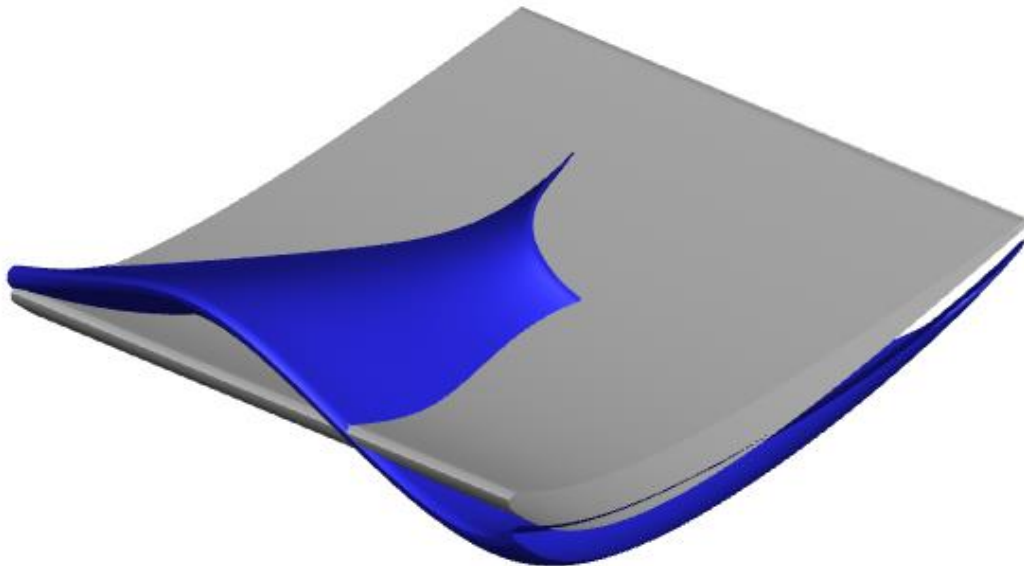


Figure 8.5: Initial (grey) and optimized with manufacturing constraints (blue) shape of blade.

The optimized blade shape is significantly deformed, in comparison with the initial shape, for both cases, as seen in Figs. 8.4 and 8.5. The optimized shape for the constrained case corresponds to the purple circle while the optimized shape for the unconstrained case corresponds to the orange circle as seen in Fig. 8.3.

The comparison between the constrained and unconstrained optimized blade shapes can be seen in Fig. 8.6.

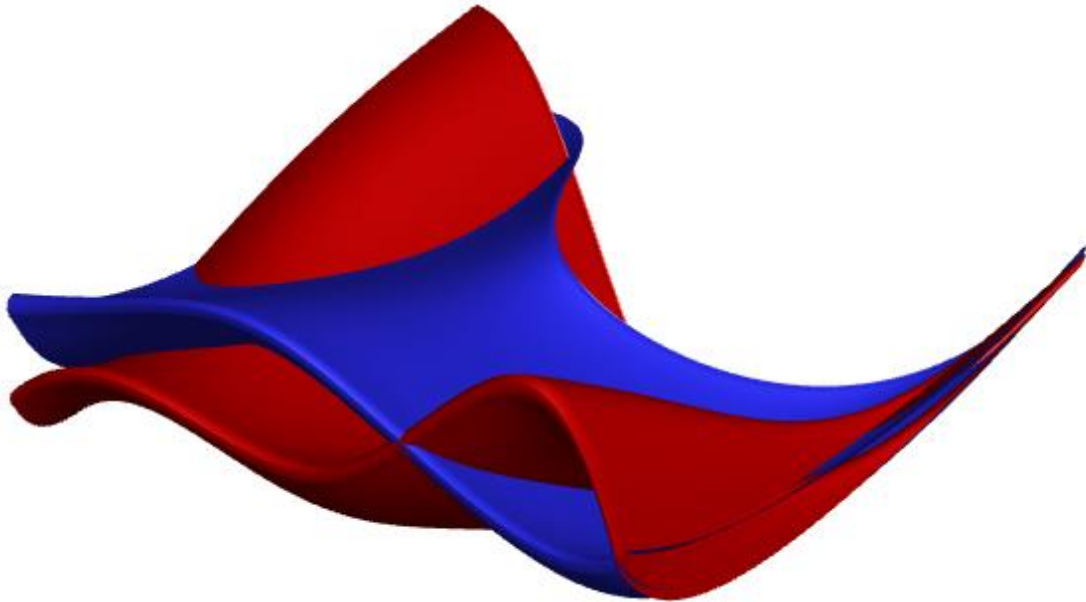


Figure 8.6: Optimized without constraints (red) and optimized with manufacturing constraints (blue) shape of blade.

The optimized geometries appear to have a spanwise displacement because the design variables are spanwise parameters of the blade. Any manufacturing constraint can be applied to the geometry shape by bounding the shape deformation that EASY can handle. The great value of the GMTurbo parameterization software, within an optimization, is highlighted here. The design variables have a physical sense that can provide the designer with the ability to roughly predict the displacements the blade will undergo during the stochastic optimization and, thus, chose to optimize those variables that are significant for the application.

Chapter 9

Unconstrained Gradient Based Optimization

In this chapter, optimization runs using gradient-based optimization [1] of the TU Berlin's stator case are carried out using the set of design variables presented in Section 8.1 while the manufacturing constraints are not taken into account. The objective function is the total pressure loss coefficient ω to be minimized. The gradient-based method that is used is the steepest descent method where the design variables update is given by:

$$b_n^{i+1} = b_n^i - \eta \frac{\delta F}{\delta b_n^i} \quad (9.1)$$

where i is the counter for the optimization cycles.

The sensitivity derivatives are computed using the chain rule as follows:

$$\frac{\delta F}{\delta b_n} = \frac{\delta F}{\delta x_k^m} \cdot \frac{\delta x_k^m}{\delta b_n} \quad (9.2)$$

where F is the objective function, b_n $n = 1, \dots, N$ are the design variables.

$\frac{\delta F}{\delta x_k^m}$ is computed using the continuous adjoint method. In Fig. 9.1, the sensitivity map is given using the adjoint method in comparison with the result of the unconstrained EASY optimization for minimum of total pressure loss coefficient.

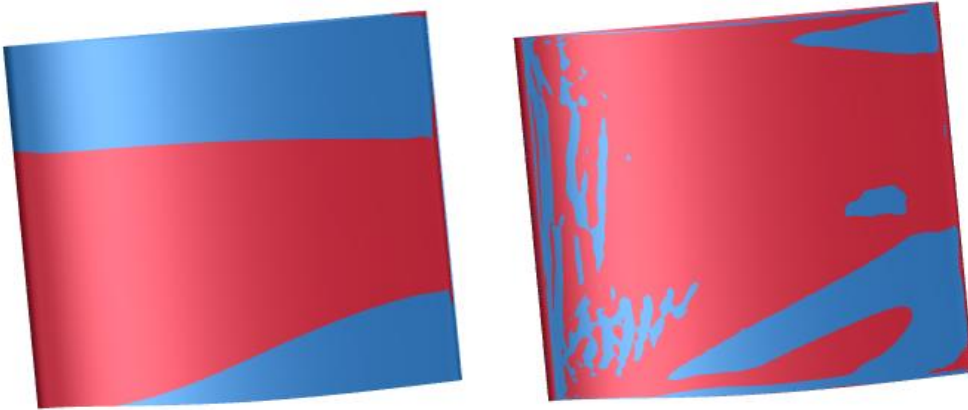


Figure 9.1: Optimized solution computed by the EA without constraints (left) and sensitivity map (right) using adjoint. The red color represents pulling the blade surface vertically while the blue color represents pushing the blade surface vertically to improve the objective function value.

The two shapes have many similarities which indicate that the adjoint method can be used to propose changes to alternate the shape of the geometry with significantly less computational budget. The differences between the two shapes are due to the parameterization method used in EA optimization which smoothens the shape of the geometry.

$\frac{\delta x_k^m}{\delta b_n}$ are the derivatives of grid nodal positions with reference to the design variables and are calculated from the differentiation of the parameterization model by creating N geometries, where $N = 46$ is the number of design variables, which differ by a small quantity, $\varepsilon = 10^{-8}$, from the initial.

The derivatives are computed using finite differences by the following equation:

$$\frac{\delta x_k^m}{\delta b_n} = \frac{x_k^m(b_1, b_2, \dots, b_n + \varepsilon, \dots, b_N) - x_k^m(b_1, b_2, \dots, b_n, \dots, b_N)}{\varepsilon} \quad (9.3)$$

where $k = 1, 2, 3$ and $x_1 = x$, $x_2 = y$, $x_3 = z$, $n = 1, \dots, N$ and $m = 1, \dots, M$ are the number of the nodes

The values of the objective function ω that is minimized using the steepest descent method for unconstrained optimization are given in Fig 9.2. The horizontal axis corresponds to optimization cycles. The initial solution is presented in cycle 0 and is also given in Fig. 9.2.

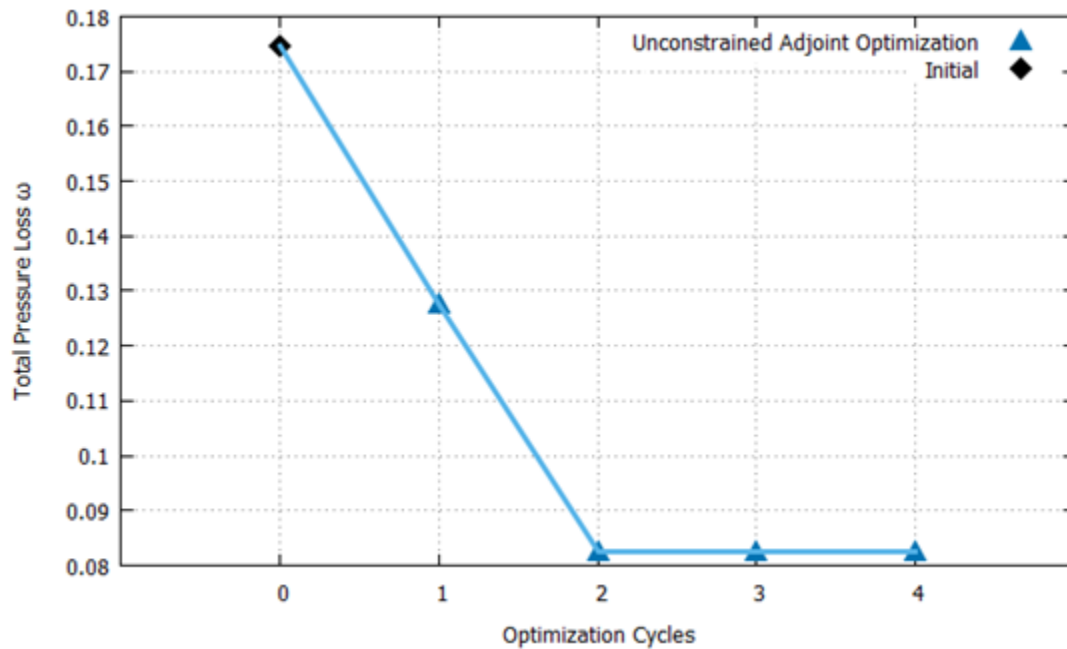


Figure 9.2: Total pressure loss coefficient evolution for unconstrained continuous adjoint optimization. ω is decreased to 0.0825 which corresponds to 52.8 %. The improvement of the objective is defined by Eq. 8.1.

The convergence history of the primal equations for optimization cycle 2 is given in Fig. 9.3.

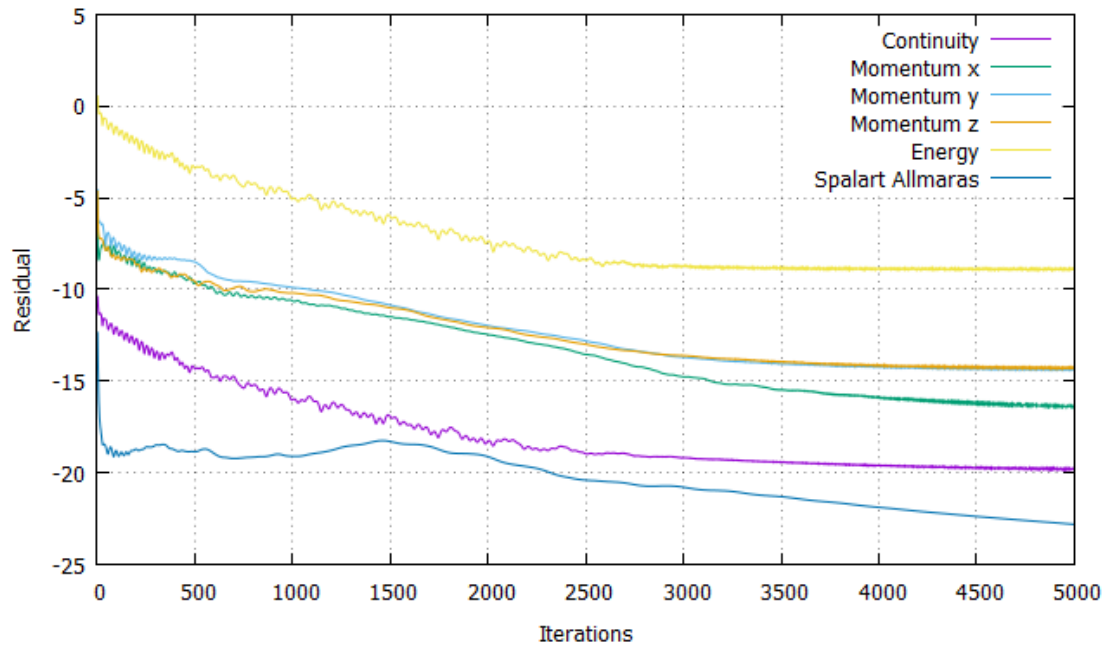


Figure 9.3: Convergence history of the primal equations for cycle 2. The horizontal axis corresponds to the primal solver iterations.

The convergence history of the objective function for optimization cycle 2 is given in Fig. 9.4. It can be seen that the solver has successfully converged.

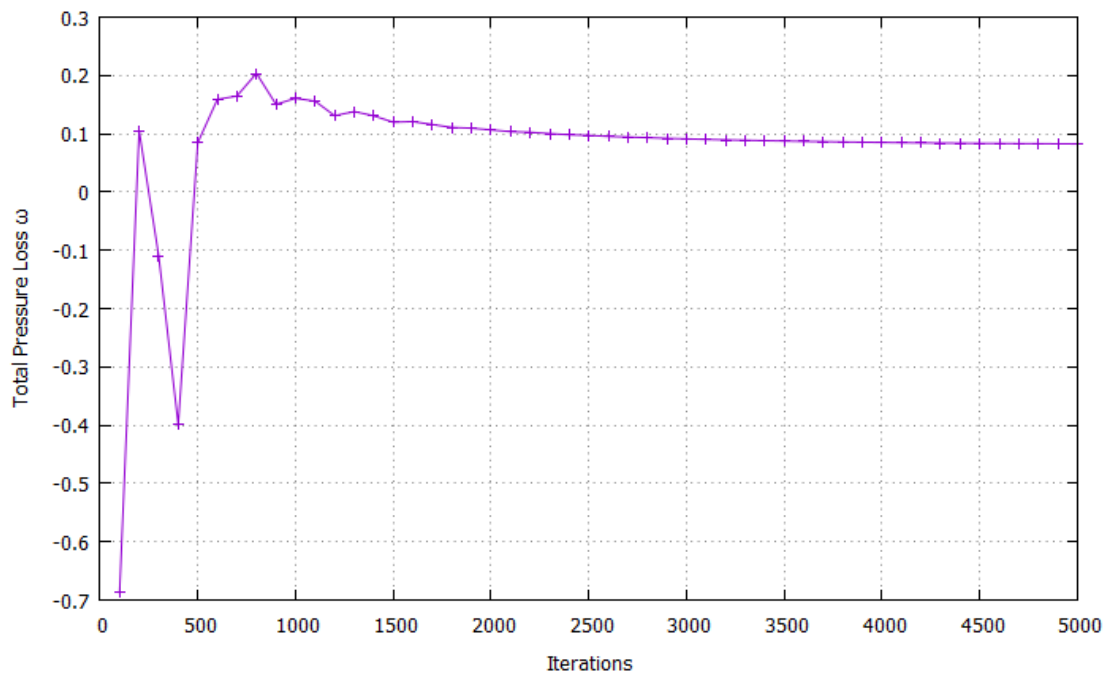


Figure 9.4: Convergence history of the objective function for cycle 2. The horizontal axis corresponds to the primal solver iterations.

As it has already been stated in Section 3.2, the system of equations is solved using the explicit multi-stage Runge-Kutta method with residual smoothing using the point-implicit Jacobi method.

The resulting optimized geometry compared to the baseline, for 46 design variables, for the case without manufacturing constraints is presented in Fig. 9.5.

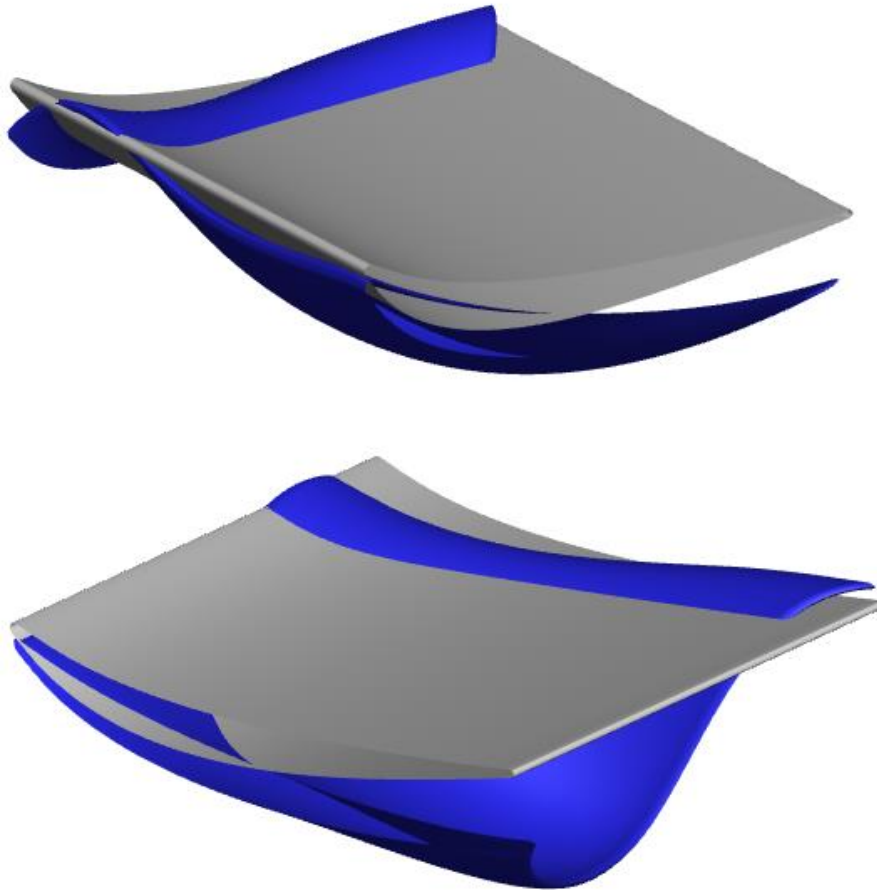


Figure 9.5: Initial (grey) and optimized without manufacturing constraints (blue) shape of blade.

By comparing the results of the gradient-based optimization with the unconstrained EA optimization, can be seen that the gradient-based reduces the objective function ω even more than the Pareto point of the unconstrained case with the minimum ω . As it can be seen from Fig. 8.3 this point corresponds to $\omega = 0.114$, yielding 34.5% reduction in comparison with the initial, while in the adjoint optimization ω is decreased to 0.0825 which corresponds to 52.8 %. The computational cost of the adjoint optimization is significantly less than the EA optimization. However, the EA is used for two-objective optimization, while the adjoint optimization is single-objective. It is possible that the adjoint method does not give that low value of the objective function if used for the same two-objective optimization.

Chapter 10

Overview and Conclusions

10.1 Overview

The work presented in this diploma thesis concerned the application of shape optimization methods to the open case of the TU Berlin's compressor stator. Specifically, the following tasks were implemented:

- Validation of the PUMA CFD solver with experimental data published by TU Berlin. Comparison of the experimental data with the results produced from the versions of the PUMA CFD solver for compressible flow, with and without low Mach number preconditioning, and for incompressible flow. The results demonstrate that the PUMA solver efficiently handles the case and that the flow can be approximated as incompressible. By approximating the flow as incompressible, the solving time is reduced for 4 hours to 3.5.
- Application of the RPT as a node-to-CAD method is used to transform the blade geometry, given in CFD grid format, into a GMTurbo format. Validation of the RPT was also performed. Comparison between the initial and the reparameterized geometry, show that the shapes are very close, thus the reparameterization is accurate enough.
- Application of the GMTurbo parameterization software, which is ideal for use in optimization workflows, since the parameters it uses are few and have physical meaning. Thus, optimization that can be carried out using GMTurbo is expected to work.
- Application of the GAT, to adapt the initial CFD grid to the GMTurbo surfaces using the 2D and 3D Spring Analogy Morphers. Hence, CFD simulation can be performed on the GMTurbo parameterized geometry. The validation of the GAT in terms of geometry and CFD results shows that the grid adaptation is accurate and can be integrated in the optimization workflow.
- Programming and application of codes in order to handle the manufacturing constraints of the case. The tough constraints of the case were sufficiently handled and applied in the optimization.
- Application of shape optimization using the Evolutionary Algorithms software EASY. The results of the optimization were satisfying taking into account the computational budget available for this thesis. The results would be even better for both the constrained and unconstrained cases if more computational means were used. The EA optimization, for the constrained case, lasted about

one week and resulted in reduction of total pressure loss coefficient by 0.25 % and the outlet angle by 1.5° which corresponds to 32.8 %.

- Application of the continuous adjoint method in order to produce the sensitivity map that shows how shape can be changed to optimize the geometry. The representation of the sensitivity map was accurate and slightly different by the EA optimized blade. As far as the optimization is concerned, ω is decreased to 0.0825 which corresponds to 52.8 %. The results of the adjoint optimization are better than the unconstrained EA optimization. The computational cost for the four optimization cycles is approximately 24 hours, significantly less than the cost of the EA. However, the EA is used for two-objective optimization, while the adjoint optimization is single-objective. As a result, it is not guaranteed that the adjoint optimization will provide as good results for the two-objective optimization.

10.2 Proposals for Future Work

As far as ideas for future work are concerned, the following are proposed:

- Implementation of a grid generating tool to create grid different from the one used in this diploma thesis in order to test the independency of the results from the size, the type and the quality of the grid.
- Use of higher computational budget for the EASY optimization and comparison with the results presented in this thesis.
- Application of two-objective gradient-based optimization, with and without taking the manufacturing constraints into consideration, and comparison with the results produced by the EA as presented in this thesis.
- Handling of the constraints using Sequential Quadratic Programming or other methods. Some of the TU Berlin case's constraints are non-linear. Hence, SQP method which is one of the most effective methods for nonlinearly constrained optimization can be used to handle the constraints with low computational cost and high accuracy.

Appendix A

Conformal Mapping

For a mapping of a surface $S(u, v)$ to a surface $P(u, v)$ through the transformation function $\Phi : S(u, v) \rightarrow P(u, v)$ to be conformal, it is sufficient to prove that a function $c(u, v) > 0$ that satisfies Eq. A.1 exists.

$$\bar{g}_{ij}(u, v) = c(u, v)g_{ij}(u, v), \text{ for } i, j = 1, 2 \quad (\text{A.1})$$

where g_{ij} and \bar{g}_{ij} are the coefficients of the first fundamental form of $S(u, v)$ and $P(u, v)$ [4]. The most important property of every conformal mapping is the angle preservation property. *Conformal mappings preserve the magnitude and direction of the angle between two curves.* [4]

The mapping of a surface of revolution $S(u, \theta) = (r(u)\cos\theta, r(u)\sin\theta, z(u))$ to a (m, θ) plane is given by the transformation:

$$\Phi: (r(u)\cos\theta, r(u)\sin\theta, z(u)) \rightarrow (m(u), \theta) \quad (\text{A.2})$$

where $m(u)$ is given by:

$$m(u) = \int_0^u \frac{\sqrt{r_u(t)^2 + z_u(t)^2}}{r(t)} \quad (\text{A.3})$$

The mapping of Eq. A.2 is conformal according to Eq A.1.

For a point s that lies on the revolved surface $S(u, v) = (r(u)\cos\theta, r(u)\sin\theta, z(u))$ the partial derivatives with reference to u and θ are given by:

$$\begin{aligned} \frac{\partial s}{\partial u} &= \left(\frac{\partial r(u)}{\partial u} \cos \theta, \frac{\partial r(u)}{\partial u} \sin \theta, \frac{\partial z(u)}{\partial u} \right) \\ \frac{\partial s}{\partial \theta} &= (-r(u) \sin \theta, r(u) \cos \theta, 0) \end{aligned} \quad (\text{A.4})$$

and for a p point on the (m, θ) plane $p(u, \theta) = (m, \theta)$ the partial derivatives with reference to u and θ are given by:

$$\begin{aligned} \frac{\partial p}{\partial u} &= \left(\int_0^u \frac{\sqrt{r_u(t)^2 + z_u(t)^2}}{r(t)}, 0 \right) \\ \frac{\partial p}{\partial \theta} &= (0, 1) \end{aligned} \quad (\text{A.5})$$

while the first fundamental coefficients of s are given by:

$$\begin{aligned} E_s &= \frac{\partial s}{\partial u} \cdot \frac{\partial s}{\partial u} = \left(\frac{\partial r(u)}{\partial u} \right)^2 + \left(\frac{\partial z(u)}{\partial u} \right)^2 \\ F_s &= \frac{\partial s}{\partial u} \cdot \frac{\partial s}{\partial \theta} = 0 \\ G_s &= \frac{\partial s}{\partial \theta} \cdot \frac{\partial s}{\partial \theta} = r^2(u) \end{aligned} \quad (\text{A.6})$$

while the first fundamental form coefficients of p , by:

$$\begin{aligned} E_p &= \frac{\partial p}{\partial u} \cdot \frac{\partial p}{\partial u} = \frac{\left(\frac{\partial r(u)}{\partial u} \right)^2 + \left(\frac{\partial z(u)}{\partial u} \right)^2}{r^2(u)} \\ F_p &= \frac{\partial p}{\partial u} \cdot \frac{\partial p}{\partial \theta} = 0 \\ G_p &= \frac{\partial p}{\partial \theta} \cdot \frac{\partial p}{\partial \theta} = 1 \end{aligned} \quad (\text{A.7})$$

from Eqs. A.6 and A.7 the relation between the coefficients is given by:

$$\begin{aligned} E_p &= \frac{E_s}{r^2(u)} \\ F_p &= \frac{F_s}{r^2(u)} \\ G_p &= \frac{G_s}{r^2(u)} \end{aligned} \quad (\text{A.8})$$

That proves that the mapping of Eq. A.2 is conformal with conformal factor $\frac{1}{r^2(u)}$.

The square of the distance between two points s and $s + ds$ on surface S is given by the first fundamental form of s seen in the following equation:

$$I_s = l_s^2 = ds \cdot ds = E_s du^2 + 2F_s du d\theta + G_s d\theta^2 \quad (\text{A.9})$$

and the square of the distance between two points p and $p + dp$ on the (m, θ) plane is given by the first fundamental form of p seen in the following equation:

$$I_p = l_p^2 = dp \cdot dp = E_p du^2 + 2F_p du d\theta + G_p d\theta^2 \quad (\text{A.10})$$

Combining Eq. A.9 and Eq. A.10, is easily proved that the relationship between the magnitudes on the two surface representations $S(u, \theta)$ and (m, θ) is given by:

$$l_s = r(u) l_p \quad (\text{A.11})$$

Consequently, the mapping used to transform a revolved surface into a (m, θ) plane, given by Eq. A.2, being conformal, preserves angles and scales the magnitudes by a factor of $r(u)$ as seen in Eq. A.11.

Bibliography

- [1] Tsiakas K.T., Gagliardi F., Trompoukis X.S. and Giannakoglou K.C.: *Shape Optimization of Turbomachinery Rows using a Parametric Blade Modeller and the Continuous Adjoint Method running on GPUs*. 2016.
- [2] Tsiakas K.: *Development of Shape Parameterization Techniques, a Flow Solver and its Adjoint, for Optimization on GPUs. Turbomachinery and External Aerodynamics Applications*. PhD thesis, Laboratory of Thermal Turbomachines, NTUA, Athens, 2019.
- [3] Trompoukis X.: *Numerical solution of aerodynamic-aeroelastic problems on GPUs*. PhD thesis, Laboratory of Thermal Turbomachines, NTUA, Athens, 2012.
- [4] Rossgatterer M., Jüttler B., Kapl M. and Della Vecchia G.: *Medial design of blades for hydroelectric turbines and ship propellers*. Computers & Graphics, 36(5): 434-444, 2012.
- [5] Kampolis I.: *Parallel, Multilevel Algorithms for the Aerodynamic Optimization In Turbomachines*. PhD thesis, Laboratory of Thermal Turbomachines, NTUA, Athens, 2009.
- [6] Zormpa M.: *Programming of a "Back-to-CAD" Tool for Use in the Analysis and Optimization of Turbomachinery Rows. Industrial Applications*. Diploma Thesis, Laboratory of Thermal Turbomachines, NTUA, 2018.
- [7] Chorin A.: *A numerical method for solving incompressible viscous flow problems*. Journal of Computational Physics, 2(1): 12-26, 1967.
- [8] Turkel E.: *Preconditioned methods for solving the incompressible and low speed compressible equations*. Journal of Computational Physics, 72: 277-298, 1987.
- [9] Menter F. R.: *Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications*. AIAA Journal, 32 (8): 1598-1605, 1994.
- [10] Menter F. R., Kuntz M., and Langtry R.: *Ten Years of Industrial Experience with the SST Turbulence Model*. Turbulence, Heat and Mass Transfer 4, 625 – 632, 2003.
- [11] Asouti V.: *Aerodynamic Analysis and Design Methods at High and Low Speed Flows on Multiprocessing Platforms*. PhD thesis, Laboratory of Thermal Turbomachines, NTUA, Athens, 2009.
- [12] Nocedal J. and Wright S. J.: *Numerical Optimization*. Springer-Verlag, New York, 1999.

- [13] Giannakoglou K. C.: *Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence*. Progress in Aerospace Sciences, 38(1): 43–76, 2002.
- [14] Kampolis I.C., Zymaris A.S., Asouti V.G. and Giannakoglou K.C.: *Multilevel Optimization Strategies Based on Metamodel-Assisted Evolutionary Algorithms, for Computationally Expensive Problems*. IEEE Congress on Evolutionary Computation, Singapore, 2007.
- [15] Γιαννάκογλου Κ.Χ.: *Μέθοδοι Βελτιστοποίησης στην Αεροδυναμική*. Εθνικό Μετσόβιο Πολυτεχνείο, 2006.
- [16] TU Berlin Stator testcase:
www.aboutflow.sems.qmul.ac.uk/events/munich2016/benchmark/testcase3
- [17] Bahadori A.: *Natural Gas Processing, Technology and Engineering Design*. Gulf Professional Publishing, 2014.
- [18] Roe P.: *Approximate Riemann Solvers, Parameter Vectors and Difference Schemes*. Journal of Computational Physics, 43(2): 357-372, 1981.
- [19] Spalart P. and Allmaras S.: *A one-equation turbulence model for aerodynamic flows*. AIAA Paper 1992-439, 30th Aerospace Sciences Meeting and Exhibit, Nevada, USA, 1992.
- [20] Trompoukis X.S., Tsiakas K.T., Ghavami Nejad M., Asouti V.G. and Giannakoglou K.C.: *The Continuous Adjoint Method on Graphics Processing Units for Compressible Flows*. OPT-i, International Conference on Engineering and Applied Sciences Optimization, Greece, June 4-6, 2014.
- [21] Tsiakas K.T., Trompoukis X.S., Asouti V.G., Ghavami Nejad M. and Giannakoglou K.C.: *Shape Optimization Using the Continuous Adjoint Method and Volumetric NURBS on a Many GPU System*. 8th GRACM International Congress on Computational Mechanics, Greece, July 12-15, 2015.
- [22] Kapsoulis D., Tsiakas K., Trompoukis X., Asouti V.G. and Giannakoglou K.C.: *PCA-Assisted Hybrid Algorithm Combining EAs and Adjoint Methods for CFD-based Optimization*. Applied Soft Computing, 73: 520-529, 2018.
- [23] Kavvadias I.S., Papoutsis-Kiachagias E.M. and Giannakoglou K.C.: *On the Proper Treatment of Grid Sensitivities in Continuous Adjoint Methods for Shape Optimization*. Journal of Computational Physics, 301: 1-18, 2015.
- [24] Asouti V.G., Kyriacou S.A. and Giannakoglou K.C.: *PCA-Enhanced Metamodel-Assisted Evolutionary Algorithms for Aerodynamic Optimization*. 11th World Congress on Computational Mechanics (WCCM XI), Barcelona, Spain, July 21-25, 2014.



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Μηχανολόγων Μηχανικών

Εργαστήριο Θερμικών Στροβιλομηχανών

Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής
& Βελτιστοποίησης

Βελτιστοποίηση Μορφής της Σταθερής Πτερύγωσης Συμπιεστή με Περιορισμούς με χρήση Εξελικτικών Αλγορίθμων και τη Συνεχή Συζυγή Μέθοδο

Διπλωματική Εργασία

Αλέξιος Τσαχαλίδης

Επιβλέπων:

Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, 2019

Εισαγωγή

Ο αεροδυναμικός σχεδιασμός και η βελτιστοποίηση πτερυγώσεων στροβιλομηχανών αποτελεί αντικείμενο έρευνας μεγάλης σημασίας. Η βελτιστοποίηση της μορφής του πτερυγίου είναι καίρια για τις επιδόσεις του.

Σκοπός της διπλωματικής αυτής εργασίας είναι η εφαρμογή μεθόδων βελτιστοποίησης μορφής πτερυγίων στροβιλομηχανών στη σταθερή πτερύγωση συμπιεστή του Πολυτεχνείου του Βερολίνου, με χρήση Εξελικτικών Αλγορίθμων, και συγκεκριμένα του λογισμικού EASY, αλλά και με τη χρήση μεθόδου κλίσεων όπως η Συνεχής Συζυγής μέθοδος, τα οποία αναπτύχθηκαν στην ΜΠΥ&Β/ΕΜΠ, και η παρουσίαση των αποτελεσμάτων. Παράλληλα, πραγματοποιείται σύγκριση και επαλήθευση των αποτελεσμάτων του επιλύτη ροής PUMA με τα πειραματικά δεδομένα που παρέχονται από το Πολυτεχνείο του Βερολίνου.

Σταθερή Πτερύγωση Συμπιεστή Πολυτεχνείου Βερολίνου

Η αρχική γεωμετρία δίνεται στο Σχ. 1. Ορίζονται τρία σημεία λειτουργίας της μηχανής: το σημείο σχεδιασμού με 42° περιφερειακή γωνία στην είσοδο και δύο άλλα σημεία με αντίστοιχη γωνία 47° και 37° . Οι δύο στόχοι βελτιστοποίησης παρουσιάζονται στη συνέχεια:

1) Ελαχιστοποίηση του συντελεστή απωλειών ολικής πίεσης μεταξύ εισόδου και εξόδου του χωρίου ΥΡΔ. Ο συντελεστής απωλειών ολικής πίεσης ορίζεται αδιάστατα ως εξής:

$$\omega = \frac{p_{t1} - p_{t2}}{p_{t1} - p_1} \quad (1)$$

όπου p_{t1} είναι η ολική πίεση εισόδου, p_{t2} η ολική πίεση εξόδου και p_1 η στατική πίεση εισόδου όλες μαζικά ολοκληρωμένες στην εκάστοτε θέση.

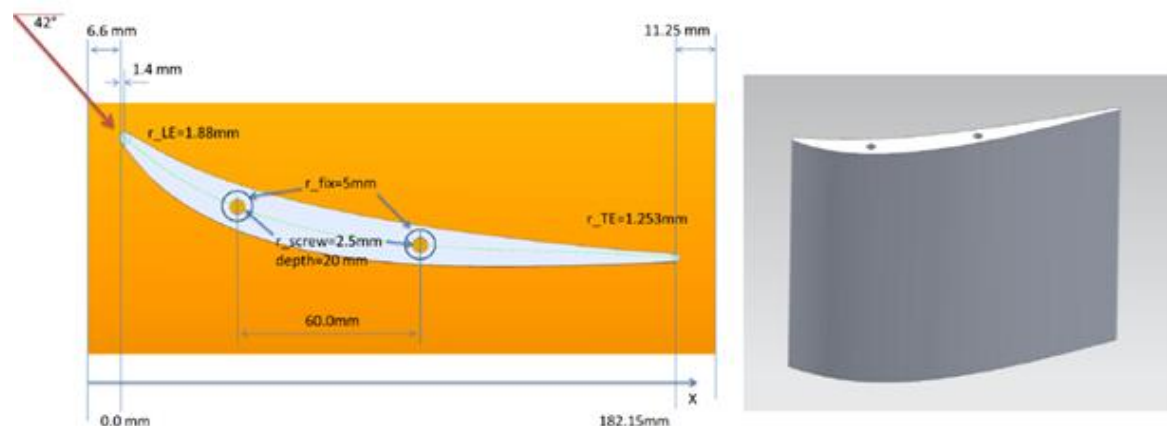
2) Ελαχιστοποίηση της απόκλισης της γωνίας εξόδου της ροής από την αξονική διεύθυνση. Η γωνία εξόδου ορίζεται ως:

$$a_E = \sqrt{\frac{\sum_A \dot{m}_i a_i^2}{\sum_A \dot{m}_i}} \quad (2)$$

όπου κάθε γωνία a_i δίνεται ως:

$$a = \cos^{-1} \left(\frac{\vec{V} \cdot \vec{i}}{|\vec{V}|} \right) = \cos^{-1} \left(\frac{w}{\sqrt{u^2 + v^2 + w^2}} \right) \quad (3)$$

όπου (u, v, w) είναι οι τρεις Καρτεσιανές συνιστώσες της ταχύτητας



Σχήμα 1: Αρχική γεωμετρία και διαστάσεις πτερυγίων σταθερής πτερύγωσης συμπιεστή του πολυτεχνείου του Βερολίνου

Επιλύτης Ροής PUMA

Για τη μελέτη της ροής γύρω από τη σταθερή πτερύγωση χρησιμοποιείται ο επιλύτης ροής PUMA που τρέχει σε κάρτες γραφικών. Το λογισμικό αυτό επιλύει αριθμητικά τις εξισώσεις Navier-Stokes μαζί με τις εξισώσεις του μοντέλου τύρβης k- ω SST, που χρησιμοποιείται στην εργασία αυτή, σε ένα υπολογιστικό χωρίο με χρήση κεντροκομβικής μεθόδου πεπερασμένων όγκων σε μη-δομημένα πλέγματα. Ο επιλύτης PUMA χειρίζεται το δομημένο πλέγμα της εργασίας σαν μη-δομημένο. Η εφαρμογή σε κάρτες γραφικών προσδίδει σημαντική αύξηση της ταχύτητας εκτέλεσης των υπολογισμών.

Οι εξισώσεις που επιλύονται δίνονται στις Εξ. 4, 6, και 7.

$$\Gamma_{nm}^{-1} \frac{\partial U_m}{\partial t} + \frac{\partial f_{nk}^{inv}}{\partial x_k} - \frac{\partial f_{nk}^{vis}}{\partial x_k} = 0 \quad (4)$$

Γ_{nm} , είναι το μητρώο προσταθεροποίησης και δίνεται ως εξής:

$$\Gamma_{nm} = \begin{bmatrix} 1 + \frac{1}{2} |\vec{u}|^2 \kappa & -\kappa u_1 & -\kappa u_2 & -\kappa u_3 & -\kappa \\ \frac{1}{2} |\vec{u}|^2 \kappa u_1 & 1 - \kappa u_1^2 & -\kappa u_1 u_2 & -\kappa u_1 u_3 & -\kappa u_1 \\ \frac{1}{2} |\vec{u}|^2 \kappa u_2 & -\kappa u_1 u_2 & 1 - \kappa u_2^2 & -\kappa u_2 u_3 & -\kappa u_2 \\ \frac{1}{2} |\vec{u}|^2 \kappa u_3 & -\kappa u_1 u_3 & -\kappa u_2 u_3 & 1 - \kappa u_3^2 & -\kappa u_3 \\ -\frac{1}{2} |\vec{u}|^2 [1 - \delta] & [1 - \delta] u_1 & [1 - \delta] u_2 & [1 - \delta] u_3 & -\delta \end{bmatrix} \quad (5)$$

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_j k)}{\partial x_j} = \bar{P}_k - \beta^* \rho k \omega + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_j} \right] \quad (6)$$

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho u_j \omega)}{\partial x_j} = \frac{a}{v_t} \bar{P}_k - \beta \rho \omega^2 + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_j} \right] + 2(1 - F_1) \frac{\rho \sigma_{\omega 2}}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \quad (7)$$

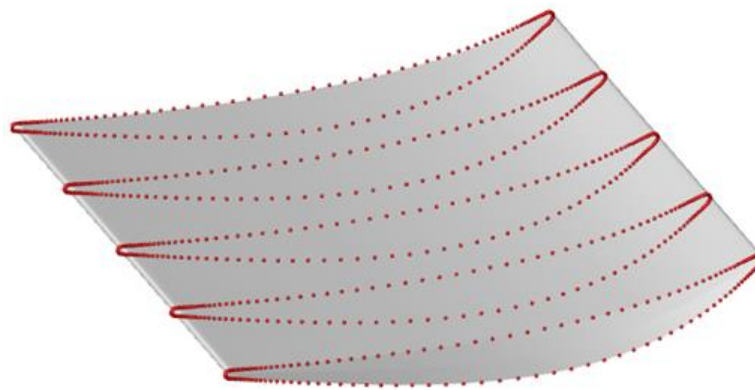
Παραμετροποίηση των πτερυγίων

Προκειμένου να σχεδιαστεί και να βελτιστοποιηθεί η μορφή του πτερυγίου απαιτείται ακριβής γεωμετρική αναπαράσταση. Αυτό γίνεται με χρήση κατάλληλης παραμετροποίησης. Στην παραμετροποίηση πρέπει να χρησιμοποιούνται παράμετροι, η φύση των οποίων σχετίζεται άμεσα με τη θεωρία των στροβιλομηχανών όπως η μεσημβρινή τομή, οι γωνίες μετάλλου και η κατανομή πάχους. Η παραμετροποίηση που θα χρησιμοποιηθεί γίνεται με χρήση του λογισμικού GMTurbo που έχει αναπτυχθεί από τη ΜΠΥΡ&Β/ΕΜΠ.

GMTurbo

Το λογισμικό GMTurbo λαμβάνει τις απαραίτητες μεταβλητές εισόδου προκειμένου να δημιουργήσει την επιφάνεια των δύο πλευρών, υπερπίεσης και υποπίεσης, ενός 3D πτερυγίου με χρήση αναπαράστασης NURBS, σε μορφή συμβατή με λογισμικά CAD. Η διαδικασία που ακολουθείται περιγράφεται στη συνέχεια.

Αρχικά, ορίζονται οι NURBS καμπύλες που περιγράφουν τις μεσημβρινές των τοιχωμάτων ποδός και κεφαλής, εισόδου, εξόδου και των ακμών πρόσπτωσης και εκφυγής του πτερυγίου. Στη συνέχεια, δημιουργούνται οι ενδιάμεσες καμπύλες στο μεσημβρινό επίπεδο οι οποίες αποτελούν εκ περιστροφής επιφάνειες που μετασχηματίζονται σύμμορφα. Στη συνέχεια, υπολογίζεται η μέση γραμμή κυρτότητας με μία κυβική καμπύλη Bézier. Έχοντας ορίσει μία μέση γραμμή κυρτότητας σε κάθε εκ περιστροφής επιφάνεια, έχει δημιουργηθεί το πτερύγιο μηδενικού πάχους. Γίνεται υπέρθεση κατανομών πάχους στις δύο πλευρές υπερπίεσης και υποπίεσης, κατά μήκος του πτερυγίου. Τέλος, πραγματοποιείται παρεμβολή με δύο NURBS επιφάνειες των αεροτομών κατά μήκος του πτερυγίου για την ολοκλήρωση της κατασκευής του πτερυγίου όπως φαίνεται στο Σχ. 2.



Σχήμα 2: Τελικό πτερύγιο σταθερής πτερύγωσης συμπίεστη του πολυτεχνείου του Βερολίνου

Λογισμικό Αντίστροφης Παραμετροποίησης

Προκειμένου να πραγματοποιηθεί η παραμετροποίηση μέσω GMTurbo πρέπει πρώτα να χρησιμοποιηθεί το Λογισμικό Αντίστροφης Παραμετροποίησης RPT το οποίο μετατρέπει ένα πλέγμα ΥΡΔ σε παραμέτρους εισόδου συμβατές με το λογισμικό GMTurbo. Η μέθοδος που ακολουθείται περιγράφεται στη συνέχεια.

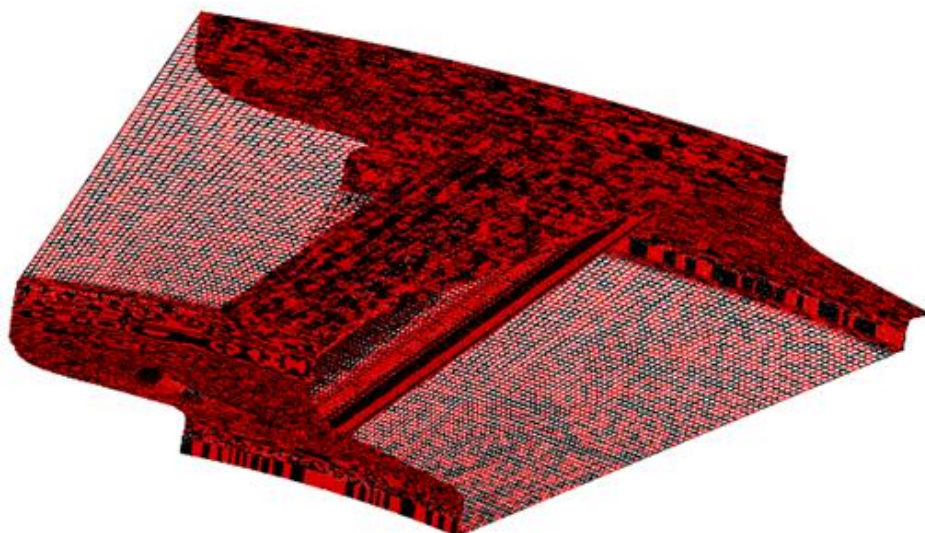
Σε πρώτη φάση, υπολογίζεται η μεσημβρινή τομή του πλέγματος. Στη συνέχεια, γίνεται υπολογισμός των πλευρών υπερπίεσης και υποπίεσης και της μέσης γραμμής κυρτότητας με προσέγγιση σημείων από μία κυβική καμπύλη Bézier. Τέλος, υπολογίζεται η κατανομή πάχους κατά μήκος της χορδής του πτερυγίου.

Λογισμικό Προσαρμογής Πλέγματος GAT

Προκειμένου να ακολουθήσει ανάλυση ΥΡΔ με χρήση κάποιου κώδικα, απαιτείται η προσαρμογή πλέγματος γύρω από την παραγόμενη από τον GMTurbo γεωμετρία. Αυτό γίνεται με χρήση του Λογισμικού Προσαρμογής Πλέγματος, GAT, με το οποίο το αρχικό πλέγμα προσαρμόζεται στη νέα γεωμετρία με χρήση του 2D και 3D Λογισμικού Παραμόρφωσης Πλέγματος με τη μέθοδο των Γραμμικών Ελατηρίων που έχει αναπτυχθεί από την ΜΠΥΡ&Β/ΕΜΠ. Η προσαρμογή πλέγματος πραγματοποιείται, αρχικά, δημιουργώντας επιφανειακό πλέγμα πάνω στην GMTurbo γεωμετρία και στη συνέχεια, παραμορφώνοντας το αρχικό ογκικό πλέγμα με βάση το επιφανειακό.

Για τη δημιουργία επιφανειακού πλέγματος πάνω στη GMTurbo γεωμετρία, προσαρμόζεται το αρχικό επιφανειακό πλέγμα πάνω στα στερεά όρια της γεωμετρίας. Τα στερεά όρια είναι τα εξής: 1) Πλευρά υπερπίεσης, 2) Πλευρά υποπίεσης, 3) Κέλυφος ποδός, 4) Κέλυφος κεφαλής. Για καθένα ακολουθείται η εξής διαδικασία:

Το αρχικό επιφανειακό πλέγμα προβάλλεται στην NURBS επιφάνεια που προκύπτει από το λογισμικό GMTurbo. Ωστόσο, από αυτή την προβολή, τα όρια του πλέγματος δεν ταυτίζονται με τα όρια της επιφάνειας. Για την επίλυση αυτού του προβλήματος, πραγματοποιείται παραμόρφωση του προβεβλημένου πλέγματος ώστε να εφαρμόζει στα όρια της επιφάνειας με τη χρήση της μεθόδου γραμμικών ελατηρίων. Αφού υπολογιστούν τα έγκυρα επιφανειακά πλέγματα πάνω στα στερεά όρια, μετατοπίζεται το εσωτερικό πλέγμα με τη χρήση ενός κώδικα παραμόρφωσης 3D πλέγματος με τη μέθοδο των γραμμικών ελατηρίων. Έτσι, προκύπτει ένα έγκυρο ογκικό πλέγμα. Στο Σχ. 3 δίνεται το τελικό επιφανειακό πλέγμα όλων των στερεών ορίων που βρίσκεται πάνω στη GMTurbo γεωμετρία.



Σχήμα 3: Τελικό (κόκκινο) πλέγμα και αρχικό (μαύρο)

Ποσοτικοποίηση των Περιορισμών

Ο σταθερός αριθμός των πτερυγίων $n = 15$, λαμβάνεται υπόψη στην επιβολή της συνθήκης εξόδου της ροής μάζας η οποία είναι ίση με την συνολική μάζα εξόδου διαιρεμένη με τον αριθμό των πτερυγίων: $9.5/15 = 0.6333 \text{ kg/s}$

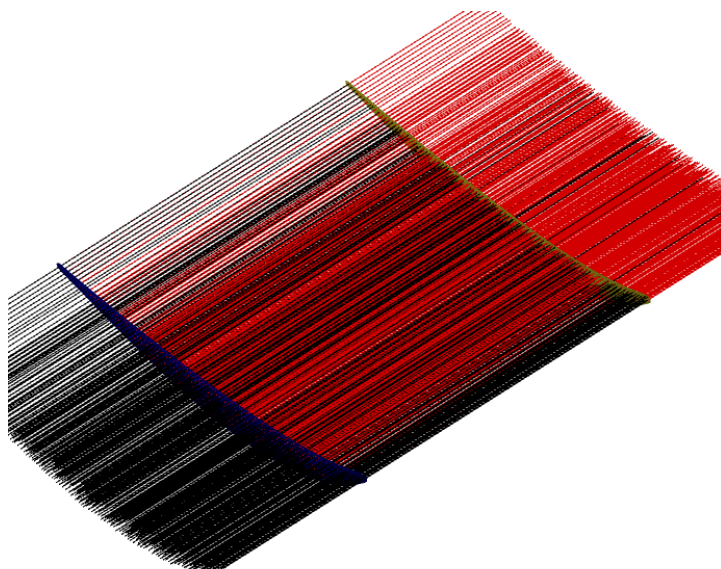
Η αξονική χορδή του πτερυγίου διατηρείται σταθερή μέσω της παραμετροποίησης του GMTurbo που επιτρέπει τέτοιες επιλογές.

Η ελάχιστη τιμή ακτίνας 1 mm για την ακμή προσβολής και την ακμή εκφυγής του πτερυγίου πληρείται επίσης μέσω της GMTurbo παραμετροποίησης που δίνει τη δυνατότητα στον χρήστη να καθορίσει το σχήμα των δύο ακμών.

Ο περιορισμός που απαιτεί, στις θέσεις κατασκευής των δύο οπών, το πάχος υλικού να επιτρέπει την ύπαρξη κυλίνδρου ακτίνας 5 mm και βάθους 20 mm ώστε να μπορεί

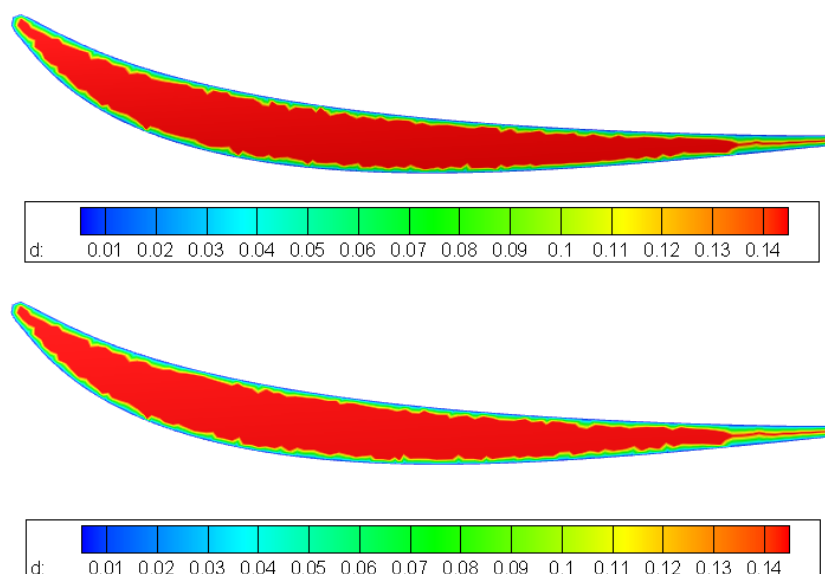
να κατασκευαστεί σπείρωμα αντιμετωπίζεται μέσω της διαδικασίας που περιγράφεται στην συνέχεια, πληρώντας παράλληλα και τον περιορισμό της απόστασης των δύο οπών που πρέπει να είναι μεγαλύτερη από 60 mm.

Αρχικά, το εύρος του συντελεστή πάχους για τις πλευρές υπερπίεσης και υποπίεσης καθορίζεται ώστε να επιτρέπει την ύπαρξη πάχους μεγαλύτερο από 5mm. Επιπλέον, κατασκευάζεται πλέγμα εσωτερικά του πτερυγίου και κατασκευάζονται ευθείες γραμμές, στη διεύθυνση της ακμής προσβολής, σε όλη την επιφάνεια του πτερυγίου οι οποίες ξεκινούν τόσο από το κέλυφος ποδός όσο και από το κέλυφος κεφαλής όπως φαίνεται στο Σχ. 4.



Σχήμα 4: Οι κόκκινες γραμμές ξεκινούν από το κέλυφος ποδός (μπλε επιφάνεια) με κατεύθυνση προς το κέλυφος κεφαλής (κίτρινη επιφάνεια) και οι μαύρες γραμμές ξεκινούν από το κέλυφος κεφαλής με κατεύθυνση προς το κέλυφος ποδός.

Οι γραμμές αυτές οι οποίες τέμνουν το πτερύγιο έχουν διαφορετικά μήκη τα οποία αποτελούν το βάθος των οπών που δύναται να κατασκευαστούν ξεκινώντας τόσο από το κέλυφος ποδός όσο και από το κέλυφος κεφαλής. Συνεπώς, διαφορετικές περιοχές του πτερυγίου τέμνονται από διαφορετικά μήκη γραμμών δηλαδή διαφορετικά βάθη οπών, όπως φαίνεται στο Σχ. 5.



Σχήμα 5: Ισοεπιφάνειες του βάθους d σε m από το κέλυφος ποδός (πάνω σχήμα) και από το κέλυφος κεφαλής (κάτω σχήμα).

Από το Σχ. 5 προκύπτει ότι το βάθος των 20 mm μπορεί να κατασκευαστεί στην περιοχή που περικλύεται από την γαλάζια γραμμή. Η επιφάνεια που επιτρέπει το ζητούμενο βάθος εξάγεται σε (m, θ) συντεταγμένες που αποτελούν ισογραμμές της μέσης γραμμής κυρτότητας και των πλευρών υπερπίεσης και υποπίεσης. Τα σημεία αυτά παρεμβάλλονται με καμπύλες SPLINE. Οι συντεταγμένες μετατρέπονται στο επίπεδο (x, y, z) και υπολογίζεται η απόσταση μεταξύ των σημείων της πλευράς υπερπίεσης και της πλευράς υποπίεσης. Η απόσταση αυτή πρέπει να είναι τουλάχιστον 10.5 mm καθώς επιβάλλεται συντελεστής ασφαλείας 5% . Τέλος, η απόσταση μεταξύ του δεύτερου και του προτελευταίου σημείου, επί του πτερυγίου που πληρεί τον περιορισμό των 10.5 mm υπολογίζεται και πρέπει να είναι μεγαλύτερη από 60 mm .

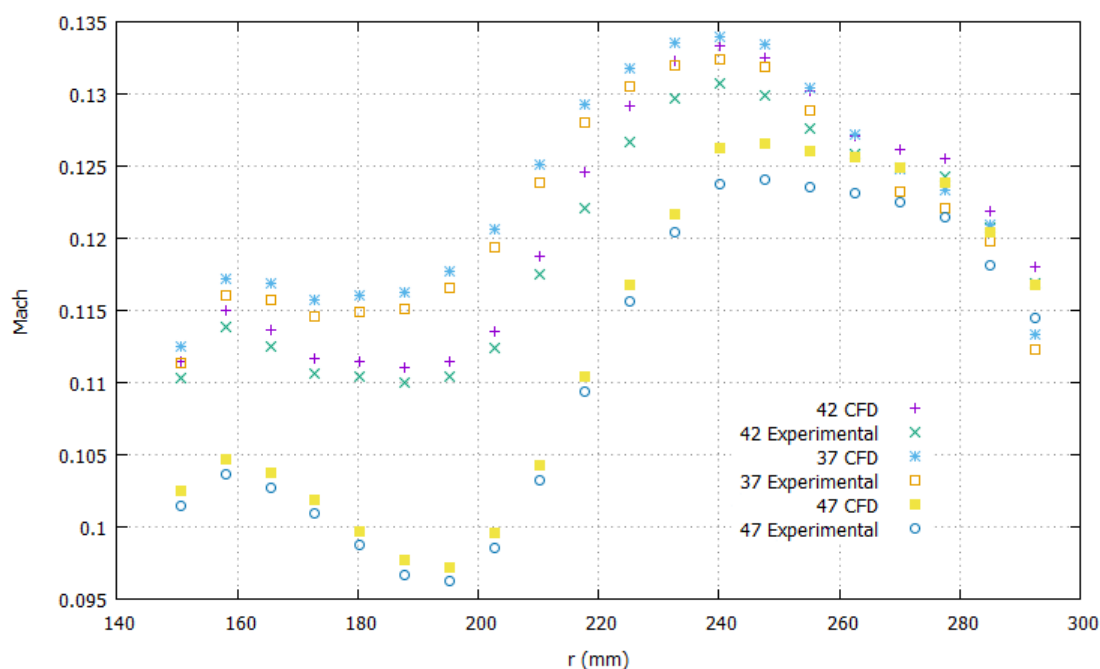
Ο περιορισμός που απαιτεί το πτερύγιο πρέπει να μπορεί να χωρέσει σε πλάκα διαστάσεων $200\text{ mm} \times 80\text{ mm}$ πληρείται με τον καθορισμό του εύρους μεταβολής της θέσης των ακμών εκφυγής και προσβολής κατά την z διεύθυνση και του εύρους των θ_{LE} , θ_{TE} , β_{LE} , β_{TE} , δ_{LE} και δ_{TE}

Τέλος, δεν υπάρχει μεταβολή της θέσης του κελύφους ποδός κατά την ακτινική διεύθυνση πληρώντας τον αντίστοιχο περιορισμό. Η ακτίνα του κελύφους ποδός ελέγχεται από τη GMTurbo παραμετροποίηση μέσω της γενέτειρας του hub.

Ανάλυση με λογισμικό ΥΡΔ

Πραγματοποιήθηκε ανάλυση της ροής με χρήση του επιλύτη PUMA προκειμένου να επαληθευτούν τα αποτελέσματα από τα πειραματικά δεδομένα. Η ανάλυση γίνεται με μοντέλα συμπίεστης ροής, με και χωρίς χρήση προσταθεροποίησης χαμηλών αριθμών Mach, και ασυμπίεστης ροής. Στο Σχ. 6 δίνεται ο Mach εξόδου, από τον ΥΡΔ κώδικα, για τις τρεις διαφορετικές περιπτώσεις γωνίας, μαζί με τα αντίστοιχα πειραματικά

δεδομένα. Στον οριζόντιο άξονα, δίνεται η ακτίνα r σε mm η οποία αυξάνεται από το κέλυφος ποδός στο κέλυφος κεφαλής.



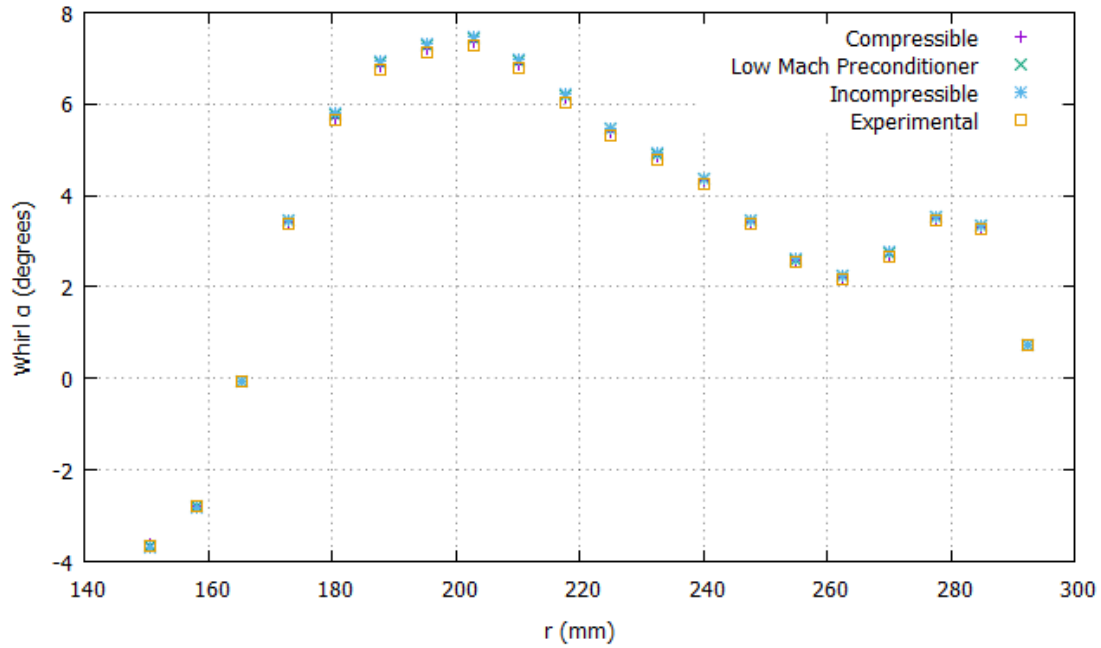
Σχήμα 6: Σύγκριση Mach εξόδου για κώδικα ΥΡΔ και πειραματικά δεδομένα για 42°, 37° και 47° γωνία εισόδου

Από το Σχ. 6 προκύπτει ότι τα αποτελέσματα του κώδικα ΥΡΔ συμπιεστής ροής είναι πολύ κοντά στα πειραματικά με σχετικό σφάλμα μικρότερο από 1 %. Το σχετικό σφάλμα ορίζεται μέσω της:

$$\varepsilon_r = \frac{\text{Πειραματικά} - \text{ΥΡΔ}}{\text{Πειραματικά}} \cdot 100 \% \quad (8)$$

Προκύπτει επίσης ότι για όλες τις τιμές γωνίας εισόδου ο αριθμός Mach είναι μικρότερος από 0.135 για τις διάφορες τιμές της ακτίνας. Συνεπώς, είναι δυνατόν να χρησιμοποιηθεί προσταθεροποίηση χαμηλών αριθμών Mach ακόμα και να προσεγγισθεί η ροή από ασυμπίεστο κώδικα όπως φαίνεται και στο Σχ. 7 για την γωνία α .

Στο Σχ. 7 δίνεται η γωνία α εξόδου για 42° γωνία εισόδου για κώδικα ΥΡΔ συμπιεστής και ασυμπίεστης ροής και με χρήση προσταθεροποίησης χαμηλών αριθμών Mach:



Σχήμα 7: Γωνία α εξόδου για 42° γωνία εισόδου για κώδικα ΥΡΔ συμπιεστής ροής με και χωρίς χρήση προσταθεροποίησης χαμηλών Mach και για ασυμπίεστη ροή

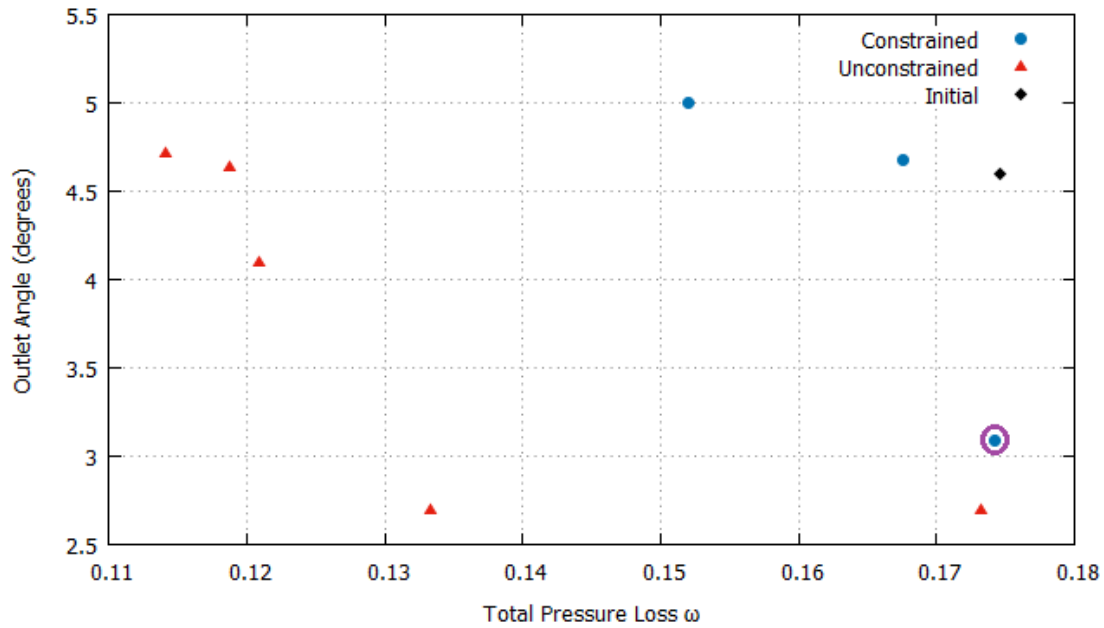
Βελτιστοποίηση με Εξελικτικούς Αλγορίθμους

Οι μεταβλητές σχεδιασμού προκύπτουν ως εξής:

- Η ακμή προσβολής και η ακμή εκφυγής δίνονται σε (r, z) . Τρεις από τις επτά μεταβλητές r των καμπυλών NURBS τόσο για την ακμή προσβολής όσο και για την ακμή εκφυγής χρησιμοποιούνται ως μεταβλητές σχεδιασμού.
- Οι μεταβλητές θ_{LE} , θ_{TE} , β_{LE} , β_{TE} , δ_{LE} , δ_{TE} και συντελεστής πάχους για τις πλευρές υπερπίεσης και υποπίεσης δίνονται σε κατά μήκος του πτερυγίου κατανομή με πέντε σημεία για κάθε καμπύλη NURBS. Όλα τα σημεία θ_{LE} , θ_{TE} , β_{LE} , β_{TE} , δ_{LE} , δ_{TE} και συντελεστή πάχους χρησιμοποιούνται ως μεταβλητές σχεδιασμού.

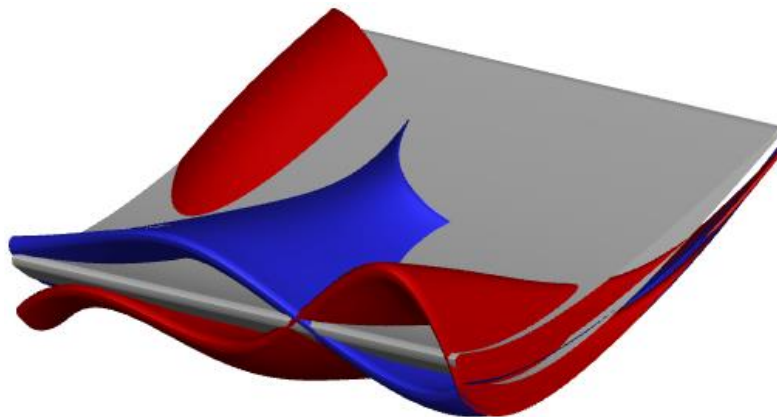
Ο συνολικός αριθμός των μεταβλητών σχεδιασμού είναι 46.

Το μέτωπο Pareto για το πρόβλημα με και χωρίς περιορισμούς δίνεται στο Σχ. 8. Επίσης, δίνονται και οι αντίστοιχες τιμές για την αρχική γεωμετρία. Η βελτιστοποίηση αφορά το σημείο λειτουργίας με γωνία εισόδου 42° .



Σχήμα 8: Μέτωπο Pareto για βελτιστοποίηση με και χωρίς περιορισμούς

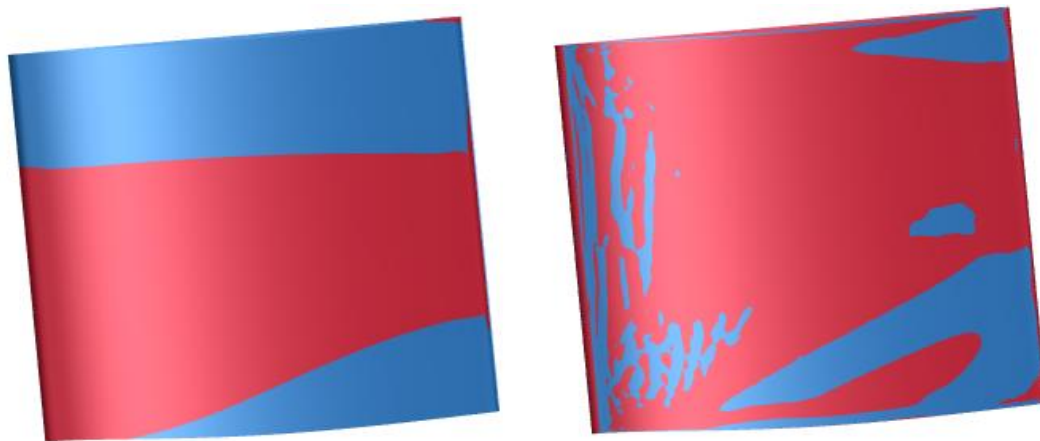
Η τελική λύση για το πρόβλημα με περιορισμούς, κυκλωμένη με μωβ στο Σχ. 8 επιτυγχάνει μείωση 0.26 % στον συντελεστή απωλειών ολικής πίεσης και μείωση 1.5° που αντιστοιχεί σε 32.8 % στην γωνία εξόδου. Η τελική βελτιστοποιημένη γεωμετρία δίνεται στο Σχ. 9.



Σχήμα 9: Αρχική γεωμετρία (γκρι), βελτιστοποιημένη χωρίς περιορισμούς (κόκκινη) και βελτιστοποιημένη με περιορισμούς (μπλε) γεωμετρία του πτερυγίου

Βελτιστοποίηση με Συζυγή Μέθοδο

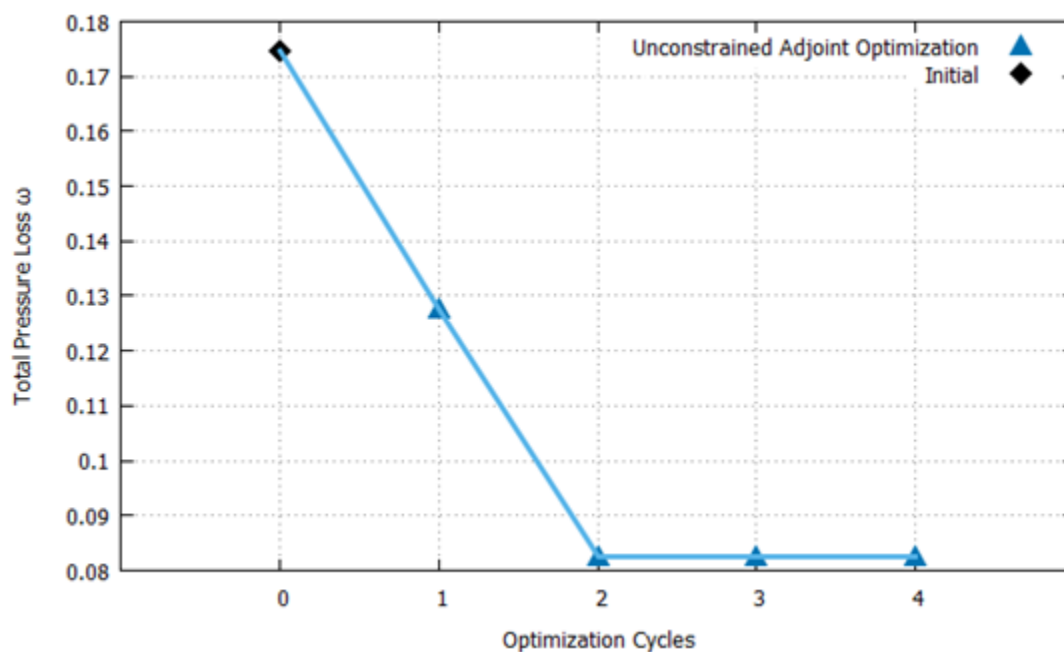
Η συζυγής μέθοδος που αναπτύχθηκε στη ΜΠΥΡ&Β/ΕΜΠ χρησιμοποιείται για τη δημιουργία του χάρτη ευαισθησίας που παρέχει τον τρόπο που πρέπει να διαφοροποιηθεί η μορφή των πτερυγίων για την επίτευξη των στόχων του προβλήματος αλλά και για την βελτιστοποίηση μορφής των πτερυγίων με σημαντικά λιγότερο υπολογιστικό κόστος σε σχέση με τη χρήση εξελικτικών αλγορίθμων. Στο Σχ. 10 δίνεται ο χάρτης ευαισθησίας με αντικειμενική συνάρτηση τον συντελεστή απωλειών ολικής πίεσης.



Σχήμα 10: Βελτιστοποιημένη χωρίς περιορισμούς με χρήση EA (αριστερά) και χάρτης ευαισθησίας (δεξιά) με χρήση συζυγούς μεθόδου. Το κόκκινο χρώμα αναπαριστά τράβηγμα της γεωμετρίας προς τα έξω ενώ το μπλε σπρώξιμο προς τα μέσα προκειμένου να επιτευχθεί η μείωση του συντελεστή απωλειών ολικής πίεσης.

Η διαφοροποίηση μεταξύ των δύο σχημάτων οφείλεται στην μέθοδο παραμετροποίησης που χρησιμοποιείται στην βελτιστοποίηση με EA η οποία προκαλεί ομαλοποίηση του σχήματος του πτερυγίου.

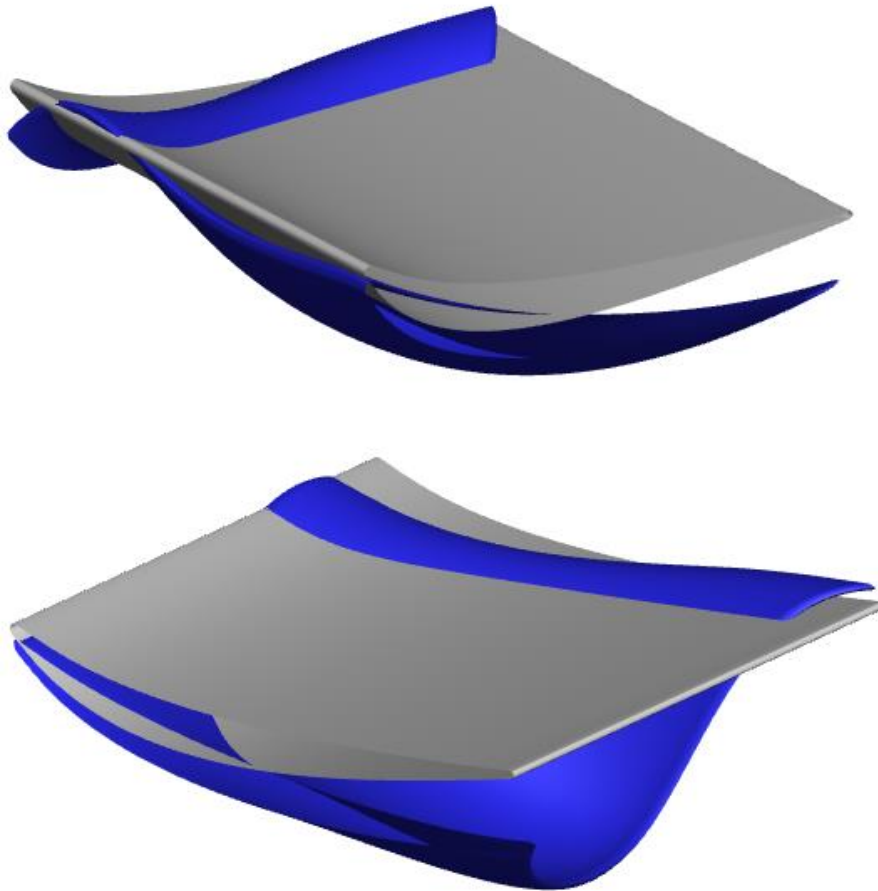
Στο Σχ. 11 δίνονται οι τιμές της συνάρτησης στόχου ω η οποία ελαχιστοποιείται με χρήση της μεθόδου απότομης καθόδου. Στη βελτιστοποίηση δεν λαμβάνονται υπόψη οι κατασκευαστικοί περιορισμοί. Στον οριζόντιο άξονα δίνονται οι κύκλοι βελτιστοποίησης.



Σχήμα 11: Συντελεστής απωλειών ολικής πίεσης ω για τους διάφορους κύκλους βελτιστοποίησης χωρίς κατασκευαστικούς περιορισμούς

Ο συντελεστής απωλειών ολικής πίεσης ω μειώνεται σε 0.0825 το οποίο αντιστοιχεί σε 52.8 % μείωση σε σχέση με την αρχική τιμή.

Η τελική γεωμετρία μετά τη βελτιστοποίηση δίνεται στο Σχ. 12.



Σχήμα 12: Αρχική γεωμετρία (γκρι), βελτιστοποιημένη χωρίς περιορισμούς (μπλε) γεωμετρία του πτερυγίου

Σύνοψη

Στη διπλωματική εργασία πραγματοποιήθηκε υπολογιστική μελέτη και βελτιστοποίηση μορφής των πτερυγίων της σταθερής πτερύγωσης συμπίεστή του Πολυτεχνείου του Βερολίνου. Συγκεκριμένα, υλοποιήθηκαν τα ακόλουθα:

- Επαλήθευση των αποτελεσμάτων του επιλύτη ΥΡΔ PUMA μέσω των πειραματικών δεδομένων που παρέχονται από το Πολυτεχνείο του Βερολίνου. Επίσης, έγινε σύγκριση των πειραματικών δεδομένων με τα αποτελέσματα που προέκυψαν από τις διάφορες εκδοχές του επιλύτη PUMA οι οποίες αφορούν συμπίεστή και ασυμπίεστη ροή και χρήση προσταθεροποίησης χαμηλών αριθμών Mach. Προκύπτει ότι ο επιλύτης PUMA δίνει αποτελέσματα εξαιρετικά κοντά στα πειραματικά. Επίσης, η ροή μπορεί να

προσεγγισθεί ικανοποιητικά από τον ασυμπίεστο κώδικα. Η χρήση του ασυμπίεστου κώδικα μειώνει τον χρόνο επίλυσης της ροής από 4 σε 3.5 ώρες.

- Εφαρμογή του Λογισμικού Αντίστροφης Παραμετροποίησης για τη μετατροπή της γεωμετρίας του πτερυγίου, η οποία δίνεται μορφή πλέγματος ΥΡΔ, σε συμβατή με το λογισμικό GMTurbo μορφή, του λογισμικού GMTurbo για την κατασκευή του πτερυγίου και του Λογισμικού Προσαρμογής Πλέγματος για τη δημιουργία πλέγματος γύρω από την GMTurbo γεωμετρία με προσαρμογή του αρχικού. Η παραμετροποιημένη γεωμετρία σχεδόν ταυτίζεται με την αρχική. Όμοια εικόνα προκύπτει και από την σύγκριση του αρχικού πλέγματος με αυτό που προκύπτει από το Λογισμικό Προσαρμογής Πλέγματος. Συνεπώς, τα παραπάνω λογισμικά μπορούν να ενταχθούν στην βελτιστοποίηση.
- Εφαρμογή και προγραμματισμός κώδικα για την ποσοτικοποίηση των κατασκευαστικών περιορισμών του προβλήματος. Οι περιορισμοί αντιμετωπίζονται αποτελεσματικά και εντάσσονται στη βελτιστοποίηση.
- Εφαρμογή βελτιστοποίησης μορφής με χρήση του λογισμικού εξελικτικών αλγορίθμων EASY. Τα αποτελέσματα της βελτιστοποίησης είναι αρκετά ικανοποιητικά λαμβάνοντας υπόψη τους διαθέσιμους για αυτήν την διπλωματική υπολογιστικούς πόρους και μπορούν να βελτιωθούν περαιτέρω αν χρησιμοποιηθούν περισσότεροι υπολογιστικοί πόροι. Η βελτιστοποίηση με EA διήρκεσε περίπου μία εβδομάδα καταλήγοντας σε μείωση 0.25 % του συντελεστή απωλειών ολικής πίεσης και 1.5° που αντιστοιχεί σε 32.8 % στην γωνία εξόδου, για την περίπτωση που λαμβάνονται υπόψη οι περιορισμοί.
- Εφαρμογή της συζυγούς μεθόδου για τη δημιουργία του χάρτη ευαισθησίας που δείχνει τον δρόμο για την αλλαγή του σχήματος του πτερυγίου για την βελτιστοποίηση της γεωμετρίας αλλά και για τη βελτιστοποίηση της μορφής του με χαμηλό υπολογιστικό κόστος. Τα αποτελέσματα του χάρτη ευαισθησίας ήταν αρκετά κοντά στο τελικό, βελτιστοποιημένο με EA, πτερύγιο. Όσον αφορά την βελτιστοποίηση, ο συντελεστής ω μειώνεται κατά 52.8 %. Τα αποτελέσματα της βελτιστοποίησης χωρίς περιορισμούς με χρήση της συζυγούς μεθόδου είναι καλύτερα από τα αποτελέσματα της βελτιστοποίησης με EA χωρίς περιορισμούς. Παράλληλα, το υπολογιστικό κόστος για τους τέσσερις κύκλους βελτιστοποίησης ήταν περίπου 20 ώρες, εξαιρετικά χαμηλότερο από αυτό των EA. Ωστόσο, οι EA χρησιμοποιήθηκαν για βελτιστοποίηση δύο στόχων ενώ η συζυγής μέθοδος για βελτιστοποίηση ενός στόχου. Συνεπώς, δεν είναι βέβαιο ότι η βελτιστοποίηση με συζυγή μέθοδο θα δώσει καλύτερα αποτελέσματα από τον EA για βελτιστοποίηση δύο στόχων.