

Adjoint-Based Inverse Design of Axial Com- pressor Airfoils

Development & Evaluation of a
New Design Method

C. S. E. F. Geuens

Master's thesis

Adjoint-Based Inverse Design of Axial Compressor Airfoils

Development & Evaluation of a New Design Method

by

C. S. E. F. Geuens

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended on Tuesday November 3, 2020 at 2:30 PM.

Supervisors:	Dr. ir. M. Pini, Prof. dr. ir. T. Verstraete,	TU Delft von Karman Institute
Thesis committee:	Prof. dr. ir. P. Colonna, Dr. R. P. Dwight,	TU Delft TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Direct design optimisation methods for turbomachinery blades have consistently gained in popularity over time. This has been accomplished by significant advances in optimisation algorithms, and notably the development of adjoint-based gradient computation, which provides objective sensitivity information at comparatively low computational cost. This enables the use of first-order optimisation methods for more complex design problems. On the other hand, interest in inverse design methods has steadily diminished over time. This is because these methods come with significant disadvantages or weaknesses, such as requiring strong simplification or only being narrowly applicable, e.g. incompressible, potential flow. However, by applying the advances in optimisation techniques to an inverse design strategy, a versatile and effective design tool can be created with functionalities not offered by direct design techniques.

In order to evaluate the feasibility and usefulness of such a method, an adjoint-based inverse design tool for 2D axial compressor blades was developed. The tool requires a baseline geometry and a target pressure or isentropic Mach distribution along the blade wall. It then analyses the flow field around the initial geometry and modifies autonomously the shape to reach closer to the desired target distribution. A discrete adjoint approach is used to compute the gradients of the objective with respect to each of the design variables. The objective and its gradients are used by a the first-order SQP optimisation algorithm to iterate the design until it matches the target distribution as closely as possible.

The newly developed design tool was used in an attempt of improving the performance of the NASA stage 35 stator blade. For comparison, also single- and multi-objective direct design optimisations were performed. It became clear that while significantly cheaper in computational cost and simpler in program complexity, the inverse design method cannot be used to reliably produce improved designs. The value of the resulting design is entirely dependent on the quality of the target distribution. It is up to the designer to create a target which is physically possible and leads to increased performance.

Still, inverse design does provide unique capabilities not offered by direct design. The tool was successfully used to retrieve the blade geometry used in a different work based on published isentropic Mach number distributions. The obtained geometry closely matched the actual geometry used in the source work. This demonstrates the accuracy of the method and the fulfilment of a use case not offered by other design methods.

Preface

This research project is a Master's thesis part of the curriculum at the TU Delft, but was executed at the von Karman Institute for Fluid Dynamics. Before the topic of the design method is opened, I would like to thank those which helped me during this project. First, my gratitude to the von Karman Institute for having me as a research student. Their facilities, tools, software and employees are what enabled me to work on this topic.

A special thanks to my supervisor at the VKI, Professor Tom Verstraete, who had great patience with me even from before the start of this project. This patience never subsided and he took his time to evaluate my work, give advice and share his insights and knowledge. I hope for him this collaboration was as pleasant as it was for me.

I would also like to extend my thanks to those other members of the VKI Turbomachinery and Propulsion Department who always explained to me how to use their work, helped me troubleshoot problems and pointed me to possible solutions. I thank Luca Zampini for his help in stopping the endless failures in the smoothing of the mesh. Thanks to Arnaud Châtel for making available the still in-development SQP optimisation algorithm and even fixing otherwise low-priority bugs so I could progress. Of course, I do not forget Tom De Bruyn for being my go-to advisor and teacher on all things CADO. Given that the entire project was based around CADO, this means a great deal.

Finally, I would like to thank my supervisor at the TU Delft, Dr. Matteo Pini, for making an exception and accepting me as a student doing the thesis externally. I hope the results do not disappoint and were worth the extra coordination.

*C. S. E. F. Geuens
Antwerp, September 2020*

Contents

Abstract	iii
Preface	v
Nomenclature	ix
List of Figures	xiii
List of Tables	xv
1 Introduction	1
2 State of Art	3
2.1 Adjoint	3
2.1.1 Historical Background	3
2.1.2 Mathematical Formulation	5
2.1.3 The Continuous and Discrete Adjoint Approaches	7
2.1.4 Other Adjoint Implementations	9
2.2 Inverse Design	9
2.2.1 Historical Overview	10
2.2.2 The Potential of Inverse Methods	11
2.3 An Adjoint-based Inverse Design Method	12
3 Methodology	15
3.1 Forward analysis	15
3.1.1 Shape Parametrisation	16
3.1.2 Topology	18
3.1.3 Mesh Generation	20
3.1.4 Flow Solution	22
3.2 Adjoint analysis	23
3.2.1 Inverse Design Objective	24
3.2.2 Adjoint Solution	26
3.2.3 Gradient Construction	29
3.3 Optimisation	30
3.4 Validation of the Inverse Method	30
4 Results and Discussion	33
4.1 Development of the Method	33
4.1.1 Gradient Verification	33
4.1.2 Verification of the Optimisation Framework	39
4.2 Loss Reductions of Existing Designs	42
4.2.1 Loss Reduction Through Inverse Design	42
4.2.2 Direct Design Optimisation	48
4.2.3 Assessment of the Inverse and Direct Methods	55
4.3 Unrealistic Targets	56
4.4 Reverse Engineering	59
5 Conclusions	63
Bibliography	65

Nomenclature

The symbols and abbreviations used throughout this document are listed below. Symbols written in regular font are fixed and those written in cursive are variable. This rule also extends to the subscripts. The symbols only used in the description of the dual problem are not included in this list. Those are mathematically defined when they appear.

Roman Symbols

Symbol	Definition
c	Chord length
i	Imaginary unit
i	incidence angle
i, j, k	Grid cell indices
J	Cost/Objective function
M	Mach number
N	Number of evaluation points
n	Evaluation point index
P	Pressure
R	Flow equations
S	Arc length
t	Blade thickness
U	State vector
V	Vertex
X	3D domain mesh
x, y, z	Cartesian coordinates
Y	Distribution value
y^+	Non-dimensional wall distance

Greek Symbols

Symbol	Definition
α	Design variable vector
α	Flow angle
β	Blade camber angle
γ	Heat capacity ratio
Δ	Difference
λ	Vector of Lagrange multipliers
ψ	Adjoint vector
ω	Loss coefficient

Subscripts

Subscript	Definition
1, 2	Vertex number
ax	Axial
Bnd	Boundary
c	Conservative
current	Current design
D1	First domain cell next to wall
FS	Free stream
<i>frac</i>	Fraction
G1	First ghost cell next to wall
in	Inlet
<i>is</i>	Isentropic
n	Negative incidence condition
<i>next</i>	Next cell
NS	Non-dimensional
o	On-design condition
out	Outlet
p	Positive incidence condition
p	Primitive
<i>prev</i>	Previous cell
PS	Pressure side
rot	Rotating frame of reference
SS	Suction side
<i>t</i>	Total
target	Target design
turb	Turbulent
<i>x, y, z</i>	Vertex cartesian coordinates

Abbreviations

Abbreviation	Definition
CAD	Computer aided design
CADO	Computer Aided Design and Optimization (software package)
CEV	Constant eddy viscosity
CFD	Computational fluid dynamics
CGNS	CFD General Notation System
CS	Complex step
GMRES	Generalized minimal residual method
LE	Leading edge
LES	Large eddy simulation
MDO	Multi-disciplinary design optimisation
MPI	Message Passing Interface
MUSCL	Monotonic upstream-centred scheme for conservation laws
NASA	National Aeronautics and Space Administration
RANS	Reynolds-averaged Navier-Stokes
PS	Pressure side
SA	Spalart-Allmaras
SS	Suction side
SLSQP	Sequential least squares programming
SQP	Sequential quadratic programming
TE	Trailing edge
VKI	von Karman Institute for Fluid Dynamics

List of Figures

2.1	Schematic overview of the inverse design tool logic.	13
3.1	Camber line angle design variables and distribution for NASA stage 35.	17
3.2	Thickness design variables and distribution for NASA stage 35.	17
3.3	NASA stage 35 stator blade geometry constructed by the tool.	18
3.4	Flow domain formed by the passage between two blades.	19
3.5	Flow domain surrounding a single blade.	19
3.6	Multiblock structure used in mesh generation.	20
3.7	NASA stage 35 2D grid before smoothing.	21
3.8	NASA stage 35 2D grid after smoothing.	22
4.1	Baseline and target geometry used for gradient verification.	34
4.2	Baseline and target isentropic Mach distribution used for gradient verification.	34
4.3	Comparison of gradients computed with finite differencing using a range of step sizes, the complex step method and the adjoint method.	35
4.4	Comparison of gradients computed with finite differencing using step size 10^{-8} , the complex step method and the adjoint method.	35
4.5	Flow field surrounding the baseline blade shape coloured by Mach number.	36
4.6	Updated baseline and target geometry.	37
4.7	Updated baseline and target isentropic Mach distribution.	37
4.8	Comparison of gradients computed with the complex step and adjoint methods using the updated designs.	38
4.9	Comparison of gradients computed with the complex step and adjoint methods with the CEV model active.	38
4.10	Convergence history of the program verification using both SQP and SLSQP.	39
4.11	Comparison of the resulting design geometries after 40 evaluations.	40
4.12	Comparison of the resulting distributions after 40 evaluations.	41
4.13	Optimised design geometry obtained with the SQP algorithm.	41
4.14	Optimised isentropic Mach distribution obtained with the SQP algorithm.	42
4.15	Isentropic Mach distribution corresponding to the NASA stage 35 blade.	43
4.16	Overview of the first two inverse design target distributions used.	43
4.17	Overview of last three the inverse design target distributions used.	44
4.18	Blade geometry resulting from inverse design with target 4.	45
4.19	Isentropic Mach distribution resulting from inverse design with target 4.	46
4.20	Blade geometry resulting from inverse design with target 5 with both adjoint- and CS-based gradient computations.	46
4.21	Isentropic Mach distribution resulting from inverse design with target 5 with both adjoint- and CS-based gradient computations.	47
4.22	Convergence history of the target 4 and both target 5 inverse design optimisations.	47
4.23	Loss bucket for the NASA stage 35 design and the baseline reference points for multi-point optimisation.	49
4.24	Comparison of objective and constraint function gradients computed with the complex step and adjoint methods.	49
4.25	Comparison of objective and constraint function gradients computed with the complex step and adjoint methods with the CEV model active.	50
4.26	Flow field surrounding the NASA stage 35 blade shape coloured by Mach number.	50
4.27	Flow field surrounding the NASA stage 35 blade shape coloured by Mach number calculated with the CEV model active.	51
4.28	Blade design resulting from the single-objective optimisation.	52

4.29	Blade design resulting from the first multi-point optimisation.	52
4.30	Thickness distributions of the baseline and multi-point optimised designs.	53
4.31	Blade design resulting from the restricted, second multi-point optimisation.. . . .	53
4.32	Convergence history of the single- and second multi-point optimisations.	54
4.33	Loss buckets corresponding to several designs obtained through inverse and direct methods.	55
4.34	Blade geometry resulting from bounded inverse design with an unrealistic target distribution.	56
4.35	Isentropic Mach distribution resulting from bounded inverse design with an unrealistic target distribution.	57
4.36	Blade geometry resulting from unbounded inverse design with an unrealistic target distribution.	57
4.37	Isentropic Mach distribution resulting from unbounded inverse design with an unrealistic target distribution.	58
4.38	Isentropic Mach distribution resulting from inverse design with target 3.	59
4.39	Target distributions for reverse engineering and the CFD and experimental data they are based on.	60
4.40	Blade geometry obtained with the CFD target distribution and the actual blade geometry.	60
4.41	Isentropic Mach distribution resulting from inverse design with the CFD target and the distribution corresponding to the actual geometry.	61
4.42	Blade geometry obtained with the experimental target distribution and the actual blade geometry.	61
4.43	Isentropic Mach distribution resulting from inverse design with the experimental target and the distribution corresponding to the actual geometry.	62

List of Tables

3.1	Initial set of design variables and the parameters they control.	16
3.2	Final set of design variables and the parameters they control.	17
3.3	Physical parameters of the CFD simulations.	23
4.1	Performance of the designs obtained with each of the inverse target distributions. . . .	44
4.2	Loss coefficients of the designs obtained through direct optimisation.	54

1

Introduction

Turbomachines are widely used for many different applications. Use cases include household appliances, aero-engines and power stations. Accordingly, the design of turbomachinery is a well researched field and many design methods have been proposed. The aim of most of these methods is to reach a turbomachinery design exhibiting the highest efficiency whilst satisfying all imposed requirements at acceptable computational cost.

Two categories of design methods can be recognised. The most straightforward option is to reach a design in a direct manner. In this approach, a design geometry is created, evaluated and possibly improved upon. This is a broad category of methods which often rely on empirical data, design charts, designer experience, or optimisation algorithms. An alternative approach is offered by inverse design strategies. In this case a desired quality or performance of the final design is imposed. The methods then aim to find a design geometry matching this requirement.

Both categories of methods have their strengths and weaknesses. For instance, direct methods may lead to poorer off-design performance than inverse methods unless special precautions are made. Generally, realistic designs are more reliably produced by inverse methods [1]. Direct design optimisations in particular are prone to reaching unrealistic results due to flow solver artefacts. In Chapter 2 an evaluation of inverse methods is made in further detail. The discussion is restricted to design methods suitable for 2D axial compressor blades. Conclusions from this analysis can then tentatively be expanded to the wider field of turbomachinery design.

A lot of work has gone in the development of direct optimisation methods. One way in which direct optimisation methods have been improved is by reducing their computational cost. This is done by using gradient-based optimisation. This reduces the number of design iterations and thus also flow evaluations. In return the computational cost per iteration is increased due to the necessity of gradient-computation. However, this cost has been strongly reduced by the development of adjoint methods, which are very cost-efficient in computing the gradient. These approaches have been improved over the years and became very potent tools in the direct optimisation toolbox.

Inverse optimisation methods have had an initial period of study and development, but have since been sidelined in favour these modern developments in direct optimisation techniques. Most of these initial inverse methods have several important drawbacks such as difficulty in implementation, prohibitive computational cost and excessive simplification leading to low accuracy. This resulted in them no longer presenting an interesting choice of design method.

However, the strong developments in direct design techniques can also be adapted and applied to inverse methods. This could circumvent some of the key disadvantages associated with inverse design. This option has received little attention so far. In this work, the developments in direct design optimisation, specifically in adjoint-based optimisation, are applied to an inverse method. This method can then be compared to a direct method where the same techniques are applied. This aim is precisely formulated in the research objective:

The research objective is to demonstrate the effectiveness and the application of the adjoint-based inverse design optimisation for 2D axial compressor blades by developing such a method, evaluating the difficulties and limitations of its implementation, validating the resulting designs and applying it to several illustrative problems.

Based on the research objective and a consideration of the necessary steps, research questions have also been formulated. The research questions guide the various steps of the project. When each of the questions can be answered, the research objective has also been completed. The questions are split into several subquestions, which together lead to the answer of the main question.

- I. How should the gradient of the target cost function be calculated?
 - (a) Which methods for gradient calculation are available?
 - (b) Under which conditions are each of these methods most advantageous?
- II. Which design requirements can be imposed on such an adjoint-based inverse design tool?
 - (a) How does the method perform under unrealistic design requirements?
 - (b) Is the original geometry regained after modification when the corresponding pressure distribution is imposed?
 - (c) How to impose geometric constraints and requirements?
- III. What can be concluded from the comparison between the adjoint-based inverse design tool and experimental results for similar cases?
 - (a) Which applications could benefit most from such adjoint-based methods?
 - (b) Does the method agree with experimental results?
 - (c) Does the tool save cost compared to other available methods?

The first question helps to determine which exact gradient-based method should be implemented and why. The first subquestion then asks which options exist, both in a general sense and in a narrow sense, within the adjoint formulation. The second aims to evaluate these alternatives. For the most part, these questions can be answered based on a literature study, but a comparison between different methods of gradient computation in a numerical experiment can provide additional insights.

The second question aims to find out the limits and behaviour of the method once implemented. Its first subquestion deals with the issue of demanding unrealistic requirements, either in the form of constraints or specified targets. The second steers towards validation of the method by checking if the tool converges to a blade geometry which is known to match the imposed condition. Another important aspect is that blade design is usually also subject to various constraints. Finding out how these can be imposed answers the third subquestion. Thus the complete answer to the second question serves as a way of verifying that the tool does reach the expected designs and can include relevant constraints when needed.

The last question serves to both validate the output of the tool and assess its performance. Therefore, the second and third subquestions ask for comparison with experimental results and other inverse design methods respectively. The first subquestion is a bit more general. It aims to assess the method by evaluating for which use cases the method is most suitable.

In this report, the current state of art on both adjoint-techniques and inverse design methods is first further investigated in Chapter 2. Based on this study, the steps in the development of the adjoint-based inverse design method are outlined. The methodology for the creation and evaluation of the design tool is explained in detail in Chapter 3. The results obtained with the tool are presented and discussed in Chapter 4. The performance of the method for several different cases is analysed in detail. The points made in these evaluations form the basis of the final assessment of the method. This assessment is presented in Chapter 5 and addresses the merits, shortcomings and use cases of the adjoint-based inverse design tool.

2

State of Art

The aim of this project is link two topics within the field of turbomachinery design. On one side there is the use of adjoint-based gradient calculation. This is a field currently experiencing a lot of research activity. On the other side is the topic of inverse design, which in contrast receives much less attention. However, the combination of these two topics might again lead to a relevant and useful design method. In order to set clear goals for the capabilities of such a method, a firm understanding of both topics is needed. This will highlight their strong and weak points and also will help in making motivated choices during the development of the method.

Both adjoint techniques and inverse design methods are discussed in this chapter based on the available literature. A historical overview and the current state of art are discussed. Section 2.1 introduces adjoint-based gradient computation. A comparison is also made between the two main formulations of the adjoint method, namely the continuous and discrete adjoint. Following this, an overview on inverse design methods for turbomachinery is given in Section 2.2. Several prominent methods are briefly evaluated by considering their features and limitations. Special attention is given to the advantages inverse design may offer over a direct design approach. Following from these discussions, a knowledge gap in linking these subjects is identified in Section 2.3. This is strongly linked to the previously defined research questions. The general steps to answering these questions are outlined.

2.1. Adjoint

Adjoint-based methods exist in many forms and are useful for a variety of engineering disciplines. Therefore a large body of research on adjoint methods can be found. First a historical overview of this research is given. The scope of the discussion is limited to the field of aerodynamics and focus is placed on turbomachinery. Three equivalent derivations of the adjoint equation are then presented, each approaching the idea of the adjoint from a different angle. Adjoint methods can further be divided into continuous and discrete approaches. Their differences, strengths and weaknesses are discussed. Some additional possible variations to the adjoint formulation are also discussed.

2.1.1. Historical Background

Even though they are not ubiquitous yet, there is a long history of research on the adjoint method. The concept was originally developed in the field of optimal control theory around 1960 [2]. The adjoint equation was derived from the Lagrangian function in an effort to accurately calculate the optimal trajectories of non-linear dynamic systems. The method also eventually found its way to other disciplines, with the field of structural mechanics being an early adopter [3]. It was Pironneau who first applied this idea to the field of fluid mechanics [4]. In his work the conditions for minimum drag of a body in Stokes flow were determined. This was later expanded to objects in higher Reynolds number flows as well [5].

Jameson later expanded on these results, introducing the adjoint method to aerodynamic design problems in 1988 [6]. Through subsequent work, he has become one of the leading researchers and developers of adjoint methods for aerodynamic design. Jameson et al. [7, 8] worked on both the

continuous and discrete adjoint implementations, but favoured the continuous method after evaluation of both [9]. Most of his work therefore was focused on the continuous adjoint.

As a result, the initial wave of research on the adjoint, spearheaded by Jameson, led to significant developments in continuous adjoint implementations. Jameson, together with various co-authors, advanced the adjoint method from potential flow to the Euler equations and the Navier-Stokes equations. The method was applied to a variety of cases, ranging from 2D airfoil shape optimisation on unstructured grids [10] to large, multi-objective optimisations of entire aircraft configurations [11, 12]. The method has been applied to inverse wing design problems as well [7]. Since then various researchers have applied and investigated the continuous adjoint method, also extending it to the field of turbomachinery design [13].

Despite these developments in the continuous adjoint method and its advantage of fast computation of sensitivity information, it would not be until more recently that there is significant academic and industrial interest in adjoint methods as a whole [14]. There are several reasons which could have led to this lack of interest. Firstly, adjoint computations are quite difficult to implement. The continuous adjoint equation needs to be derived by hand and is only valid for one specific objective function and set of flow equations. Furthermore, the boundary conditions are also tricky to formulate correctly. In case a discrete adjoint is used, access to the flow solver code is needed [3]. This establishes a large barrier to entry into the adjoint method.

Furthermore there is the issue of using high-fidelity CFD combined with gradient-based optimisation. For an effective optimisation method, reliable results and a fast computation of both the flow field and gradient are required, as multiple iterations are necessary. Until more recently the high computational cost of high-fidelity CFD and the difficult implementations of adjoint-based methods made them a suboptimal choice in many applications. Coupled with the difficulty in implementing the method, this led to little interest in its use [15]. However, in the last decade this trend has changed. Many research projects have since investigated and demonstrated the use of the adjoint-method in optimisation problems. The capabilities of the adjoint method have been proven multiple times for a variety of applications [10, 16–18].

It is notable that with the recent rise in interest, the focus has shifted from the continuous adjoint method to the discrete adjoint method. The differences between these two implementations are further discussed below. The discrete adjoint method itself is not new. Frank and Shubin highlighted a method they referred to as an ‘implicit gradient’ approach in the early 1990s [19]. This implementation amounted to what is essentially a discrete adjoint approach. In the same period Baysal and Eleshaky also demonstrated a discrete adjoint method for sensitivity analysis [20]. Both of these works used the Euler equations and found that this novel approach was indeed promising, but challenging in implementation as a separate adjoint solver needs to be created.

Further early work on the discrete adjoint was done by Elliott and Peraire [21] as well as Nielsen and Anderson [22]. In both works, the discrete adjoint method is implemented for airfoil design on unstructured grids using the Euler and Navier-Stokes equations respectively. Giles, much like Jameson, compared both the continuous and discrete adjoint methods. His opinion, in agreement with those of most researchers currently active, is that the discrete adjoint has the most promising properties for general applicability [15]. In 2001 Giles et al. have also further developed and demonstrated the discrete adjoint method for various implementations [23].

The difficulty in implementation can be reduced by employing algorithmic differentiation. This is sometimes also called automatic differentiation. This technique relies on the fact that any computer code is reducible to a series of simple mathematical operations. By finding the corresponding derivative computation for each operation, the computation of sensitivity information can be propagated through the program. This can be done in both a forward and reverse mode. Using this strategy, the discrete adjoint equations no longer require the difficult process of hand-differentiation [15, 24]. An early adopter of this technique is Mohammadi who demonstrated the relative ease of this method [25]. A notable issue are the large memory requirements of using this technique [26].

These initial forays into the discrete adjoint method were promising, but simultaneously much more complex design optimisation and inverse design problems were being solved using the more developed continuous adjoint method [11, 12]. Nevertheless, the difficulty in implementation of either methods has prevented the adjoint from becoming more widespread until recently. In the previous decade, research effort in the adjoint method has significantly increased. The aim of many has become to

develop effective adjoint methods which not only provide reliable results at low cost, but also are easy to implement and maintain. The inclusion of such methods in existing CFD packages has strongly lowered the barrier to entry. With a wider array of implementations becoming available, there now also exists the opportunity to quantitatively compare the various implementations as done by Kenway et al. [3]. By now projects routinely merely use the adjoint method as a tool instead of it being a subject of study.

2.1.2. Mathematical Formulation

As mentioned, initially the formulation of adjoint methods was based on the Lagrangian function. However, various interpretations are possible. To build a better understanding of the method, the adjoint equation is derived in three ways below. In the starting position there is a cost function $J(U, \alpha)$ which is to be minimised whilst satisfying the flow conservation equations $R(U, \alpha) = 0$. The state vector U contains the flow variables. The vector of design variables is represented as α . For gradient-based optimisation, the derivative of the cost function with respect to these design variables $dJ/d\alpha$ needs to be computed.

Lagrangian Function

The first approach, as outlined by Jameson [9], is based on the use of the Lagrangian function. The Lagrangian function is a linear combination of the cost function and flow equations as in equation 2.1 and can also be called the augmented cost function. The derivation below is performed following the notation defined by Verstraete [27].

$$\tilde{J}(U, \alpha) = J(U, \alpha) - \lambda^T \cdot R(U, \alpha) \quad (2.1)$$

In this equation, the vector λ contains Lagrange multipliers. This equation can then be differentiated to reach equation 2.2. The resulting equation can be further simplified to equation 2.3 by recognising that the flow equations evaluate to zero.

$$d\tilde{J} = \frac{\delta J}{\delta U} \cdot dU + \frac{\delta J}{\delta \alpha} \cdot d\alpha - \lambda^T \cdot \left(\frac{\delta R}{\delta U} \cdot dU + \frac{\delta R}{\delta \alpha} \cdot d\alpha \right) + d\lambda^T \cdot R \quad (2.2)$$

$$d\tilde{J} = \left(\frac{\delta J}{\delta U} - \lambda^T \cdot \frac{\delta R}{\delta U} \right) \cdot dU + \left(\frac{\delta J}{\delta \alpha} - \lambda^T \cdot \frac{\delta R}{\delta \alpha} \right) \cdot d\alpha \quad (2.3)$$

The computation of the flow differential dU is difficult to perform. Therefore the Lagrangian multipliers might be chosen to satisfy equation 2.4. This means equation 2.3 further simplifies to equation 2.5.

$$\frac{\delta J}{\delta U} - \lambda^T \cdot \frac{\delta R}{\delta U} = 0 \quad (2.4)$$

$$d\tilde{J} = \left(\frac{\delta J}{\delta \alpha} - \lambda^T \cdot \frac{\delta R}{\delta \alpha} \right) \cdot d\alpha \quad (2.5)$$

Finally, one can rewrite in order to find the gradient of the cost function with respect to the design variables, obtaining equation 2.6. Note that since the flow equation is zero, so is its total derivative. This means that the total derivative of the augmented cost function is equal to that of the cost function. Thus by solving equation 2.4, which is called the adjoint equation, one can subsequently solve for the desired sensitivity information with equation 2.6.

$$\frac{dJ}{d\alpha} = \frac{\delta J}{\delta \alpha} - \lambda^T \cdot \frac{\delta R}{\delta \alpha} \quad (2.6)$$

Duality

An alternative derivation is proposed by Giles and Pierce [15] by using the principle of duality. For the gradient computation, one can rephrase the derived objective and conservation equations to reach the dual problem. Starting from the gradient of the flow equations, given in equation 2.7, one can rewrite to equation 2.8. Again the notation from Verstraete [27] is used.

$$\frac{dR}{d\alpha} = \frac{\delta R}{\delta U} \cdot \frac{dU}{d\alpha} + \frac{\delta R}{\delta \alpha} = 0 \quad (2.7)$$

$$\frac{\delta R}{\delta U} \cdot \frac{dU}{d\alpha} = -\frac{\delta R}{\delta \alpha} \quad (2.8)$$

Similarly, for the objective function one can formulate equation 2.9.

$$\frac{dJ}{d\alpha} = \frac{\delta J}{\delta U} \cdot \frac{dU}{d\alpha} + \frac{\delta J}{\delta \alpha} \quad (2.9)$$

Equations 2.8 and 2.9 are rewritten into equations 2.10 and 2.11 respectively by substituting the shortened notation below.

$$A = \frac{\delta R}{\delta U}, w = \frac{dU}{d\alpha}, f = -\frac{\delta R}{\delta \alpha}, g^\top = \frac{\delta J}{\delta U}, h = \frac{\delta J}{\delta \alpha} \quad (2.10)$$

$$A \cdot w = f \quad (2.10)$$

$$\frac{dJ}{d\alpha} = g^\top \cdot w + h \quad (2.11)$$

This is the primal problem formulation, and would require one solution of equation 2.10 for every design variable since f is dependent on the design variables. The dual problem requires the solving a system depending on g^\top , seen in equation 2.12, which only needs to be solved once for every objective function. The terms f and g have been exchanged and the matrix A was transposed. This equation introduces the vector of adjoint variables v . The desired gradient can then be found by solving equation 2.13. Equation 2.12 is the adjoint equation. The cost of solving thus scales with the number of objective functions, but is independent of the number of design variables.

$$A^\top \cdot v = g \quad (2.12)$$

$$\frac{dJ}{d\alpha} = v^\top \cdot f + h \quad (2.13)$$

It is straightforward to prove that this is indeed the dual form of the initial formulation. By keeping in mind that $A \cdot w = f$ and the adjoint variables are selected so that $A^\top \cdot v = g$, it is clear that both formulations are equivalent.

$$v^\top \cdot f = v^\top \cdot A \cdot w = (A^\top \cdot v)^\top \cdot w = g^\top \cdot w$$

Substituting back the original terms, one finds the more familiar formulation for the adjoint and gradient equations in equations 2.14 and 2.15 respectively, where ψ is the vector of adjoint variables. The solution of the adjoint equation is independent of the design variables, and thus does not scale in cost with the design variables.

$$\frac{\delta R}{\delta U}^\top \cdot \psi = \frac{\delta J}{\delta U}^\top \quad (2.14)$$

$$\frac{dJ}{d\alpha} = \psi^\top \cdot -\frac{\delta R}{\delta \alpha} + \frac{\delta J}{\delta \alpha} \quad (2.15)$$

Separation of terms

Perhaps the most intuitive method for deriving the adjoint equation is given by Giannakoglou and Papadimitriou [28]. It is based on separating the terms with partial derivatives with respect to the state vector. This derivation is demonstrated clearly by Bradley [29] and Verstraete [27].

In equation 2.9 for the objective function gradient the most difficult term to compute is $dU/d\alpha$. It represents the sensitivity of the flow variables with respect to the design variables. The term is therefore not computed. It is also useful to note that the gradient of the flow equation is zero as shown in equation 2.7. This equation can be rewritten to equation 2.16. Substituting into equation 2.9, one then obtains equation 2.17.

$$\frac{dU}{d\alpha} = -\left(\frac{\delta R}{\delta U}\right)^{-1} \cdot \frac{\delta R}{\delta \alpha} \quad (2.16)$$

$$\frac{dJ}{d\alpha} = -\frac{\delta J}{\delta U} \cdot \left(\frac{\delta R}{\delta U}\right)^{-1} \cdot \frac{\delta R}{\delta \alpha} + \frac{\delta J}{\delta \alpha} \quad (2.17)$$

The first two factors are independent of the design variables α . Their product can be found by solving equation 2.18 for the adjoint variables ψ .

$$\left(\frac{\delta R}{\delta U}\right)^T \cdot \psi = -\left(\frac{\delta J}{\delta U}\right)^T \quad (2.18)$$

The remaining terms in equation 2.17 are not as costly to compute [29]. By solving the adjoint equation, one can subsequently find the gradient of the cost function with respect to each design variable. All three interpretations lead to exactly the same set of equations. The adjoint equations, one for each adjoint variable, are solved using an adjoint solver. This is analogous to the computation of the flow solution. The runtime of flow and adjoint solutions are similar.

2.1.3. The Continuous and Discrete Adjoint Approaches

As previously touched upon, there are the continuous and discrete adjoint formulations. Despite being based on the same idea, these methods are quite different in practice. For the continuous case the adjoint equations are derived from the continuous flow equations. The flow and adjoint equations are separately discretised and solved. The discrete adjoint equations on the other hand are derived from the already discretised flow equations [28].

There is no general consensus on whether one method is better than the other. The optimal approach depends on the specifics of the problem, experience of the developers, etc. However, recent literature shows that the discrete approach is becoming more popular than the continuous alternative [3]. Four groups have conducted notable comparisons between the discrete and continuous adjoint. In historical order these are Nadarajah and Jameson [9], Giles and Pierce [15], Peter and Dwight [14], and Kenway et al. [3]. Their work is used as reference for the comparison between both approaches hereunder.

The adjoint is purely a method for computing gradient information. It should therefore be noted that adjoint-based optimisation shares the disadvantages common to all gradient-based optimisation methods. One such issue is that for an objective function with multiple minima, there is no guarantee that the global minimum will be reached. The design will only converge to the local minimum [27]. Furthermore, gradient-based methods are only appropriate with continuous design variables and objective functions. At discontinuities the gradient is not defined and the method will fail. This implies that design variables cannot be integer valued [15]. Finally, for noisy objective functions the gradient will not provide useful information [27].

General Limitations

There also are some limitations of the adjoint method common to both continuous and discrete implementations. Constraints are a hindrance to adjoint-based methods. Constraints can be added as penalty terms to the objective functions [30]. In this case, their effect is included in the computation of the objective gradient. This leads to reduced conditioning and more iterations required until convergence. An alternative is to separately impose the constraints. In this case, additional adjoint calculations are necessary, thus reducing the advantage in computational cost of adjoint-based methods [15]. Still, it is clear that an adjoint approach is advantageous if there are less constraints and objective functions than design variables.

Giles and Pierce also point out that for objectives of the least squares type, direct finite differencing can give an approximation of the Hessian matrix as well without increasing computational cost [15]. This is typically beneficial in inverse design problems where the objective is the least squares difference between a current and desired pressure or velocity distribution, as discussed in Chapter 3. By using an approximation of second order data, rapid design convergence can be reached. The adjoint approach does not provide any information with which the Hessian can be approximated. Quasi-Newton algorithms, which create an approximation of the Hessian, will require more iterations until convergence. Nevertheless, the lower cost per iteration for the adjoint method usually still results in reduced computational cost.

Comparison of the Continuous and Discrete Approaches

The continuous adjoint is derived from the continuous flow equations. The method thus provides the gradient of the continuous flow formulation. However, the optimisation algorithm solves a discrete

problem based on the numerical formulation in the flow solver. The computed gradient therefore does not exactly match the desired gradient [3]. The inaccuracy is larger for coarser meshes. The gradient is only consistent with the discrete problem for a sufficiently fine meshes and a shock-free flow [9]. Due to this inconsistency in gradients, the method may struggle to fully converge [15].

On the other hand, the discrete adjoint formulation is based on the discretised flow equations. The computed gradients are thus exactly consistent with the numerical problem [14]. This means that complete convergence to the solution of the discrete problem is possible. Furthermore, gradients computed using a discrete adjoint method are also consistent with those computed by finite difference [9]. This allows for easy verification of the adjoint implementation.

The continuous adjoint equation is difficult to obtain. It needs to be hand-derived and depends on the set of flow equations and objective function used. This process is tedious and error-prone [15]. Especially certain terms in the turbulence model are hard to derive. These are therefore often simplified, resulting in an additional error [14]. These simplifications may also lead to stability issues resulting in poor or no convergence of the adjoint solver [31]. The boundary conditions are also difficult to formulate. For some objective functions, the set of continuous adjoint results in an ill-posed boundary condition [14]. The restrictions this places on the objective functions and boundary conditions are complex [15]. These difficulties limit development to specialised and experienced groups and reduce the applicability of the method.

The discrete adjoint equations can also be hand-differentiated. However, the process can also be simplified by using algorithmic differentiation. This way the adjoint problem can be easily derived from existing code [3, 24, 26]. This means that equations for the partial derivative terms seen in equations 2.14 and 2.15 no longer need to be found by hand, but can be determined based on the already existing flow solver code [3]. Moreover, boundary conditions follow automatically from the adjoint formulation and are consistent [14].

The coding effort for the continuous adjoint is relatively simple. Less computation and memory are required since the Jacobian matrix is not required [14]. Furthermore, Pierce indicates that deriving a set of continuous adjoint equations is the best way to gain an understanding of the nature of adjoint solutions [15]. However, this is quite an esoteric argument and does not improve the results of the method. It should also be noted that the discretisation of the continuous adjoint is free from the discretisation of the flow equations. This can lead to simpler programming and the choice of adjoint solver. Nevertheless, Jameson suggests simply sticking to the same discretisation for both sets of equations [9]. This way, the same solver can be used and it is less likely there will be stability, accuracy or convergence issues with the iterative solver.

The discrete adjoint solver is generally adapted from the flow solver used. The steady-state equations for the flow and adjoint problems have the same eigenvalues. Both solutions will thus also converge to the steady-state solution at the same asymptotic rate [14]. The eigenvalues are based on the Jacobian matrix [15]. This matrix needs to be computed to solve the discrete adjoint equations. Pre-computing the Jacobian incurs significant computational and memory costs. Jacobian-free methods can be used to reduce these costs, but lead to stiffer problems [3]. In general, discrete methods have higher solution costs than continuous methods. Other notable drawbacks are the more difficult coding efforts and a lack of physical insight or meaning offered by the method [9]. Lastly, it is easier to generalise the discrete formulation to higher-order gradient computations [14, 28].

Overall, it can be noted that Jameson is a clear proponent of the continuous adjoint method. Giles and Pierce, who worked with both approaches, favour the discrete method and are joined by Peter and Dwight as well as Kenway et al. By investigating recent publications using either methods, it can be seen that there is a marked increase in the proportion of published works using the discrete adjoint. In this project, a discrete adjoint approach will be used as well. Based on the current state-of-the-art, it can be concluded that a discrete approach is preferable when developing effective, general-use adjoint methods. This is in agreement with recent trends in published research.

2.1.4. Other Adjoint Implementations

Less conventional or modified adjoint implementations have also been proposed. A short overview of some noteworthy examples is given. One such concept are hybrid schemes. Suggested by Nadarajah and Jameson [8], these combine the discrete and continuous adjoint on different parts of the domain. Due to the difficulty of constructing boundary conditions for continuous methods, it is proposed to use a discrete formulation on the boundary whilst using a continuous formulation inside the domain. Other hybrids are also possible, such as a continuous adjoint discretisation informed by the discrete adjoint or by combining a continuous adjoint for the mean-flow equations and a discrete adjoint for the turbulence equations. However, such implementations are generally not worth the effort [14].

Variations in the complete program logic are also possible. Typically, three iterative loops are present: the flow solver, the adjoint solver and the overarching optimisation loop. For each optimisation iteration, both other solvers need to fully converge. An alternative is the one-shot method, where all three loops converge simultaneously. This was initially demonstrated by Ta'asan et al. [32]. The solutions to all three systems of equations are obtained at the same time by solving the three systems of equations simultaneously. Such methods could greatly accelerate convergence. Various research projects have investigated the one-shot method [28, 33].

The adjoint method is not limited to first-order gradient computation. The Hessian matrix can also be computed. Giannakoglou and Papadimitriou have shown that the most efficient way to do this is by first calculating the first derivative in a direct manner and subsequently use an adjoint approach to calculate the second derivative [34]. The computational cost scales linearly with the number of design variables. Second-order optimisation methods can thus be used. However, due to the linearly scaling computational cost, a balance exists between the gain from using a higher-order method and the loss due to increased cost per iteration. In further work, the performance of this method is improved by computing the exact Hessian one time, after which it is updated with an approximate estimation each iteration [35].

So far, the discussion of adjoint methods has mainly been focused on time-averaged analysis. The method can nonetheless be extended to time-varying problems, such as when using an LES model. This comes at increased computational and memory costs since values are computed and stored at each time step. The adjoint solution is chaotic, so that small initial differences can lead to vastly different results down the line. No effective LES adjoint method has been developed as of yet [3]. Demonstrations have been made, but are limited in use due to the very large costs [36].

A simplification for periodic flows is to employ reduced-order models, such as the harmonic balance method. Broadly speaking, the flow and adjoint equations are mapped to the frequency domain using the Fourier transform. In this domain, both sets of equations are solved. Using certain sampling frequencies the inverse Fourier transform is then applied and time-dependent behaviour can be modelled. These sampling frequencies are determined from, for example, the blade passing frequency. This allows for some key unsteady behaviour to be taken into account [37].

Finally, it should be noted that the adjoint method for gradient calculation is not unique to the field of fluid mechanics. Having originally been developed in the field of control theory, the technique has spread to many different engineering disciplines. In most fields where sensitivity information is needed or optimisation is performed with a certain set of design variables, the adjoint method can be beneficial. Examples of such uses include complex heat transfer problems [38] and the design of microelectronics [39]. Multidisciplinary Design Optimisations have also been performed. Gradient information is found using adjoint information from multiple engineering disciplines. For example, Verstraete et al. have applied an MDO strategy to turbine design considering both aerodynamic and stress performance [16].

2.2. Inverse Design

There are many methods available for the design of turbomachinery components. Generally they can be divided in two categories. There are direct methods, which reach a design in a forward manner. A forward analysis is one where a design is specified and its performance is subsequently analysed. In the case of direct optimisation, the design is modified to maximise a measure of performance. Inverse methods, on the other hand, are design problems where a specific target is prescribed and the corresponding design is searched for. The choice of target can vary strongly. A typical example is the pressure distribution along the blade surface.

There is a great variety of inverse design methods, ranging in applicability, accuracy, level of detail and computational cost. Many inverse design methods rely on significant simplifications, such as using the potential flow model. However, inverse design methods can also be used in combination with high-fidelity CFD. Such a method is to be demonstrated in this research project. To put this into the right context, an historical overview of the various types inverse design methods is given. The possible advantages of high-fidelity inverse design methods over modern direct optimisation methods are then investigated.

2.2.1. Historical Overview

The inverse design methods can be further divided in subclasses. The first are the pure inverse methods, which find the design geometry based purely on the provided target. A second subclass are the iterative methods. These require a starting geometry as input, and iteratively approach the geometry by matching the specified target. Leonard [40] has provided a detailed assessment of many existing inverse methods according to this division. What follows is an overview of various methods. The use cases and restrictions of these methods are highlighted.

Historically, the pure inverse methods have been the first inverse methods design developed for turbomachinery design. These have been developed before computers were able to provide assistance in the design of components. An early, important contributor to the field of inverse design in turbomachinery is Lighthill. In the period immediately following the second world war, he worked on two pure inverse design methods for blades in a potential flow field. The first is the method of conformal transformations. The blade geometry and a velocity distribution are represented on two separate planes. A mapping between the two planes is formulated, and the velocity profile is integrated to reach a blade geometry [41]. However, several additional model parameters need to be specified for which it is difficult to give any physical justification.

A more developed but similar method which gained more popularity is the hodograph method [42]. This method applies a hodograph transformation, which maps values from the potential and stream function plane to the hodograph plane. This plane has as axes the flow velocity and flow direction. The inverse problem is formulated by specifying boundary conditions on the set of elliptic equations in the hodograph domain. Solving these equations and mapping back will then yield a geometry matching the imposed flow condition. The method has later been further developed by Bauer et al. [43]. This approach is quite cumbersome and requires the designer to have a fundamental mathematical understanding of the method. Furthermore, they are limited to two-dimensional and shock-free flows. There is no straightforward method for imposing geometric constraints either. Despite these shortcomings, the hodograph method is still sometimes used for simpler inverse design problems [44].

Many other pure inverse methods exist. Generally, these inverse methods are lacking in applicability for advanced or detailed design problems. This is due to the use of significant simplifications, such as by modelling the fluid with the potential flow equations and their restriction to 2D geometries [40]. On the other hand, iterative inverse methods are more commonly used and can be employed for more complex inverse design problems. Over time, a great variety of iterative methods have been developed. These techniques cover a large range in complexity, accuracy and computational cost. Hence, some of these techniques can be used in significantly more complex design problems than the pure inverse methods. A few notable examples of iterative inverse methods are discussed.

An early example of such a method is Martensen's method [45, 46]. A velocity distribution is imposed in a potential flow field. A vortex distribution on the camber line is created to modify the velocities on the blade wall. The vortex intensity is based on the difference between the imposed and current tangential velocities. The geometry is then modified to ensure that locally the normal velocity is zero. In this way, the desired design can iteratively be found. This idea has later been extended to Euler flows as well by Van den Braembussche et al. [47]. An issue with the method is that since a vortex has no effect on the normal velocity on its own location, the method can lead to oscillatory convergence behaviour.

Van den Braembussche also worked on a similar method using a transpiration model. The flow field around a given blade is calculated. The calculated and required pressure distributions will be different as long as the desired design is not found. A second calculation is performed with the desired pressure distribution imposed along the blade wall, where the blade walls are considered permeable. The blade wall is now modified to match the newly obtained streamlines, so that the flux through the

walls becomes zero. This is then repeated until convergence [48]. This method was extended to 3D problems by specifying the pressure at multiple sections along the blade span. The technique was also applied to viscous inverse design problems by accounting for the difference between the viscous and inviscid pressure field when imposing the required pressure distribution [49]. However, in this scenario the viscous method was unable to recover the exact target pressure distribution on the obtained blade design.

Another notable researcher of inverse methods is Zangeneh. He developed a method for the 3D inverse design of mixed and radial flow turbomachinery. For this method, a circumferentially averaged swirl velocity is imposed. The concept was initially formulated for 2D-problems by Hawthorne et al. [50, 51]. Based on the specified swirl velocity distribution, the flow field can be determined. The required blade shape is then found by imposing that there is no flow velocity normal to the blade surface [52]. The method has since been further extended for compressible, viscous flow problems for both axial and radial turbomachines, as well as multi-objective problems [53–55].

Many other iterative inverse design methods exist [40]. All of the previously discussed inverse methods create or modify the geometry based directly on the physics of the problem, for instance by looking at the tangential and normal velocities. However, these methods are no longer directly related to the physical problem if viscous, compressible flows are considered. Workarounds are needed to more accurately model the flow or to extend the methods to three-dimensional problems. Furthermore, it is difficult to impose geometrical or flow constraints on the resulting design using these methods. Each of these methods also generally require a specific and sometimes obscure target formulation, such as the circumferentially averaged swirl velocity.

These drawbacks can be mitigated by considering the inverse design problem as an optimisation problem. The optimisation objective is the difference between the current blade flow field and the prescribed target condition. The design is evaluated by resolving the flow field and calculating the objective function value. An optimisation algorithm is used to modify the design variables until no reductions in the objective function can be found. Geometry modifications are no longer based directly on physical considerations, but depend simply on the optimisation algorithm.

In such a method, the objective function can freely be chosen. Constraints can also be imposed by restricting the design space or by including them in the objective function. Moreover, depending on some of the specifics in implementation, these methods can give good solutions even if unfeasible design requirements are imposed. Instead of reaching the exact solution, the closest feasible result would then be obtained. Perhaps most importantly, these methods can be used even for complicated problems without workarounds, such as when using the Navier-Stokes equations.

It is for these reasons that optimisation is an essential element of modern inverse methods. It is easy to see the link with direct design optimisation. The only difference in implementation between the inverse and direct methods in this case is the choice of the objective function. The logical extension then is to use adjoint-based optimisation methods, given their advantages discussed in Section 2.1. Iollo et al. [56, 57] have demonstrated continuous adjoint-methods for the inverse aerodynamic design of turbomachinery components. In the current project a similar discrete-adjoint method will be developed and evaluated in more detail.

2.2.2. The Potential of Inverse Methods

It is now clear that inverse design has previously been of great academic interest, resulting in a variety of methods. However, in the last two decades this interest has markedly decreased. Instead, the focus has shifted to direct design optimisation methods. This can in large part be attributed to the strong increase in available computational power, meaning that high-fidelity CFD evaluations can now be used for the iterative design of complex components. This does not necessarily mean that inverse methods are obsolete. They may also benefit from the increased availability of CFD evaluations, especially when utilising an optimisation-based inverse design method. However, the question remains what inverse design method have to offer over direct design method. This question is investigated below.

Several advantages of inverse design can be identified. Direct design optimisation typically uses some measure of efficiency or loss as parameter to be maximised or minimised. This often leads to designs which are efficient on the design point, but have unacceptably poor performance in off-design conditions [58]. A solution is to use multi-point optimisation, but then multiple flow solutions are required to evaluate the design performance every iteration [59]. For gradient-based methods multiple

gradient-computations are then also needed.

Inverse design methods typically use a certain distribution of flow values such as static pressure or isentropic Mach number along the boundary as target. When good targets are selected, blade geometries produced in this way are more robust and can be expected to have good off-design performance as well. This is because the specified target is typically not as sensitive to small changes in the inlet conditions [54].

The industry standard is to use RANS flow solvers for optimisation problems. LES methods are still too computationally expensive for use in most design problems. Whilst RANS solvers give good results for design conditions, they are not as reliable in their prediction of off-design performance. This is because they struggle in predicting the boundary layer behaviour and stability. Phenomena such as flow separation are often poorly predicted. This can lead to designs which the flow solver predicts to be efficient, but in reality perform rather poorly [60]. The false predictions are an artefact of the Reynolds averaging and the turbulence models used.

By specifying certain flow conditions, one has much more control on the final behaviour and properties of the blade and flow field. For instance, the pressure distribution gives a strong indication of the blade performance. Such a distribution is also accurately predicted by RANS solvers [1]. Since the specification of a pressure distribution is representative of the flow field around the resulting blade, it is much easier to identify unrealistic designs and anticipate performance characteristics than if a more abstract metric such as a loss factor is used [61].

It should be noted, however, that the resulting design is only as good as the imposed pressure or velocity distribution. Thus, there is a certain level of experience and physical insight necessary in order to effectively use inverse methods [1]. To remedy this, some researchers suggest to cheaply find an optimal pressure distribution, which is then used as input for inverse design [62].

An inverse design method also does not need to be limited to design improvements. One could use such a method also to quickly reach designs with a desired behaviour, to be optimised later. They may also be suitable for the adapting of existing designs to changing flow conditions. Finally, inverse methods may also be used for the reproduction of blade geometries based on only performance data. For these reasons there is still strong industry interest in fast and accurate inverse design methods. Combined with the strengths of adjoint-based optimisation, a modern inverse design method may be a welcome addition to the toolkit of the turbomachinery designer.

2.3. An Adjoint-based Inverse Design Method

In Section 2.1 the adjoint approach has been discussed. Its advantages for the calculation of gradient information have become clear. It has also been concluded that the discrete form is a better option for the creation of a versatile and effective design tool. Section 2.2 provided an overview on several types of inverse methods and examined the relevance and use cases of inverse methods, particularly by comparison with direct optimisation methods.

From these considerations, a gap in the current body of research is identified. The advances in direct optimisation methods have scarcely been applied in iterative inverse design methods. This is despite the fact that an optimisation-based inverse design method circumvents most disadvantages typically associated with inverse design techniques. Due to the use of an adjoint-approach, such a method comes with acceptable computational cost as well. There have recently been some research projects demonstrating such an adjoint-based inverse design method, but these have typically used the continuous adjoint [13, 63] or have been simple implementations without evaluating the method in detail [64].

This further motivates the development of the adjoint-based inverse design method as previously defined by the research objective and research questions. A brief outline of the procedure followed to answer these questions and complete the project goal is sketched. Before starting development on the tool, a clear overview must be created of how the tool should be set-up. The gradient computation will be done using a discrete adjoint approach. Several other elements of the design tool also need to be developed. These are the parametrisation, mesh generation, flow solver and optimiser functions. Although these aspects are not the main focus of the project, informed choices and good implementation is still needed to reach a well-functioning design tool. Furthermore, a suitable cost function definition also must be formulated and necessary constraints should be considered.

Firstly a forward flow analysis is created. A blade geometry is constructed and the surrounding flow field resolved. Next, the inverse objective can be defined and the adjoint-based gradient computation implemented. This step computes the objective gradient with respect to the mesh coordinates. The gradients of these coordinates with respect to the design variables can be computed in a forward manner. The inverse design objective gradient with respect to the design variables can then be computed. Coupled with an optimiser, the design tool is complete. A schematic overview of the resulting program components is given in figure 2.1.

The proper functioning of the tool should be checked with numerical experiments. Finally, the answers to the remain research questions can be found by evaluating the tool in various scenarios. These experiences and results will provide the necessary answers and the arguments for the assessment of the tool. Those conclusions provide valuable insights useful for the further development of inverse design tools.

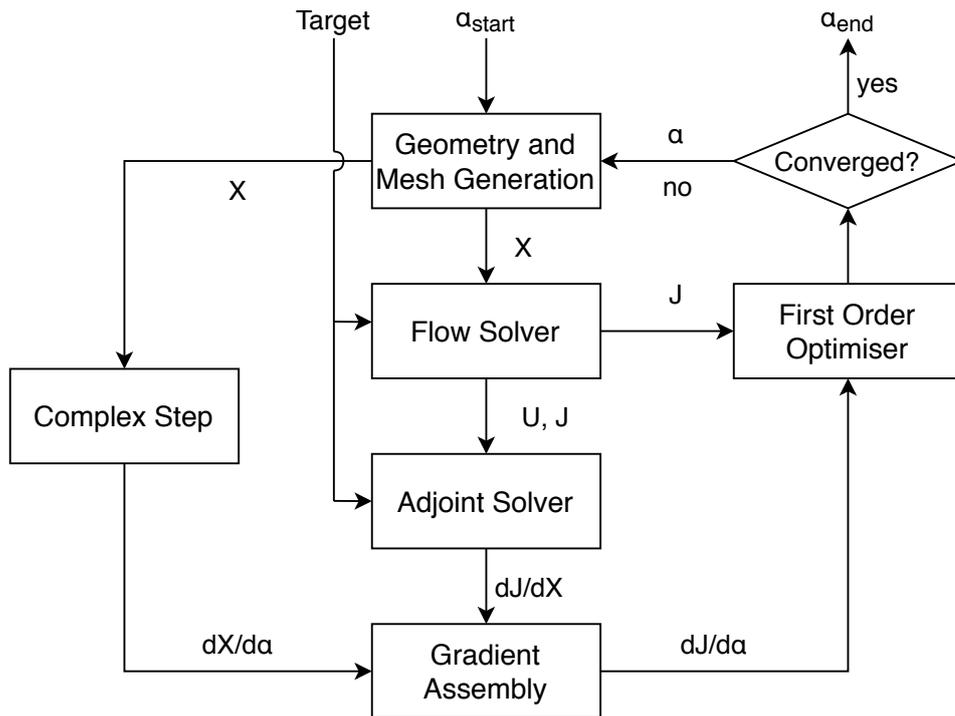


Figure 2.1: Schematic overview of the inverse design tool logic.

3

Methodology

Based on the knowledge gap identified in Chapter 2, clear goals for the tool to be developed have been defined. Next to the goals themselves, the context in which the tool is developed is also relevant. The project is supported by the von Karman Institute for Fluid Dynamics, and the tool will build on other programs previously developed there. These programs are bundled together as the CADO package, intended for the design and optimisation of turbomachinery components. CADO includes functions for geometric description of blades, mesh generation, flow analysis, discrete adjoint analysis and design optimisation. The code is written in C++ and programs are executed using Bash scripts. The programs are continuously under development. Functions within CADO often need to be adapted or customised for specific tasks. There is not yet an implementation of 2D axial compressor blades on which the inverse design tool can be based. The development of the inverse design program therefore starts here.

The development is split in four phases. In the first phase, the foundations of the tool are built, so that a generic compressor blade can be represented and analysed. This is called the forward analysis and its development is elaborated in Section 3.1. The second phase encompasses the core elements of the program, namely the formulation of an inverse design objective and its adjoint-based gradient computation. These steps are covered in Section 3.2. At this stage, the program is capable of calculating the gradients of the inverse design objective with respect to each of the design variables. This leaves gradient-based optimisation the final capability to be implemented. This process is explained in Section 3.3. The development phase also comprises the verification of the gradient computation and the optimisation framework. The use of the inverse design method can then be demonstrated and its performance assessed. The approach for these steps is documented in Section 3.4. When possible, these tasks were executed in parallel to the program development, so that necessary modification could be made in time.

3.1. Forward analysis

An essential feature of any design optimisation method is a simple way to represent and evaluate a design. This aspect and its constituent functions are referred to as the forward analysis in this work. The necessary functions for a forward analysis become clear when considering the starting point and desired result. The starting point is a set of design variables which represent a blade design. The desired outcome is a certain metric based on the flow solution around the blade.

The necessary functions logically follow. First, the blade geometry described by the design variables needs to be constructed. Around this blade, the relevant flow domain and its topology can then be defined. The scope is restricted to 2D blades. A 2D mesh can then be created for this domain. However, the CADO flow solver is restricted to 3D analyses. The 2D mesh is therefore simply extruded into the third dimension. Provided with the necessary boundary conditions and configuration, the flow solution can then be computed using the flow solver. Performance metrics such as the objective for optimisation are also computed in a post-processing step. The objective function for inverse design is introduced in Section 3.2, as it is unique to the inverse design task and has little meaning in other contexts. These steps are now discussed in more detail.

3.1.1. Shape Parametrisation

A key aspect of design optimisation is the selection of a suitable shape parametrisation. This parametrisation directly leads to the collection of design variables. Several aspects which indicate a good shape parametrisation are listed.

- The impact of each design variable should be clear to the designer.
- The number of design variables should be limited.
- The design variables should describe a continuous and smooth geometry.
- The design space should be maximised by the selection of variables.

Although the number of variables has little impact on the computational cost of the adjoint method, it should still be limited to maintain oversight. Furthermore, optimisers generally perform better with design variables which represent an important physical property, such as the blade angle or local thickness. A large selection of design variables will also result in a low impact of individual variables. The computed gradients may then be subject to a large amount of noise, reducing accuracy. A continuous description of the blade is also favoured in order to avoid inaccuracies based on format or discretisation. Still the variables should be selected so that they do not too strongly artificially limit the design space available to the optimiser.

The blade geometry is constructed in two steps. First a camber line is created. The camber line itself is computed from the blade angle distribution using equation 3.1. This equation gives the y -coordinate for x -coordinates varying from 0 to the axial chord length, c_{ax} . The camber line is computed from the blade angle curve $\beta(x)$, which represents the blade angle at every x -location. The blade angle distribution is described by a third degree Bezier curve with 6 control points. The coordinates of these control points are selected as design variables. The x -coordinates of these points are non-dimensionalised by dividing through the axial chord length and are fixed to zero and one for the first and last control points respectively.

$$y(x) = \int_0^{c_{ax}} \tan(\beta(x)) dx \quad (3.1)$$

Next, the blade thickness is applied to the camber-line to reach the blade geometry. The blade thickness distribution along the chord is described by a second Bezier curve. The thickness values are also non-dimensionalised with respect to the axial chord length. This distribution has 14 control points, so that fine modifications are possible. This is most critical near the leading and trailing edges where the thickness rapidly reaches zero. The y -value of the first and last control points are consequently fixed to zero.

The blade suction and pressure sides are generated by creating a list of 100 points offset with half the local thickness value along the camber line. The points are offset perpendicular to the camber line. These points are then interpolated with a fourth-degree B-spline for both the suction and pressure side curves. In order to ensure that the leading and trailing edges do not have a kink, G^1 -continuity is ensured by fixing the x -coordinates of the first two and last 2 coordinates to 0 and 1 respectively. Sharp edges on blades should be avoided for structural reasons.

The selected design variables are the control points for two Bezier curves. For some of the points, the coordinates need to be fixed. The axial chord-length is used to non-dimensionalise the geometry and needs to be specified for each design. This length itself is not a design variable since the input data for the inverse design problem is also non-dimensionalised. Another important value is the solidity. This value also needs to be provided for each case and is not a design variable. This leaves 32 variables for the optimisation algorithm to modify. An overview of this set is listed in table 3.1.

Table 3.1: Initial set of design variables and the parameters they control.

Variables	Associated parameter
1-4	Blade angle distribution control point x -coordinates
5-10	Blade angle distribution control point y -coordinates
11-20	Thickness distribution control point x -coordinates
21-32	Thickness distribution control point y -coordinates

The results of the gradient verification, discussed in Section 4.1.1, showed that the gradients with respect to the x -coordinates of the blade angle distribution control points are significantly smaller than other gradients. For simplicity, these x -coordinates are fixed and evenly spread, reducing the number of design variables to 28. An overview of the resulting final selection of design variables is given in table 3.2. The blade angle distributions and thickness distributions corresponding to the NASA stage 35 stator blade at mid-span are displayed in figures 3.1 and 3.2 respectively. The geometry described by these variables can be seen in figure 3.3. This specific blade was selected for numerical experiments, as motivated in Section 3.3.

Table 3.2: Final set of design variables and the parameters they control.

Variables	Associated parameter
1-6	Blade angle distribution control point y-coordinates
7-16	Thickness distribution control point x-coordinates
17-28	Thickness distribution control point y-coordinates

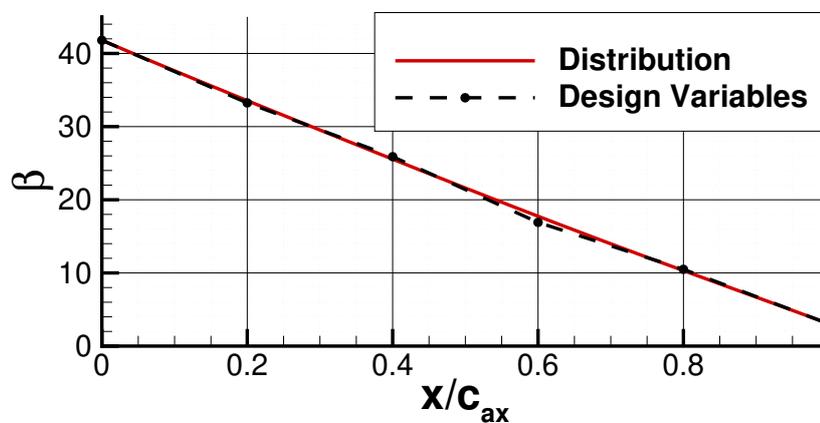


Figure 3.1: Camber line angle design variables and distribution for NASA stage 35.

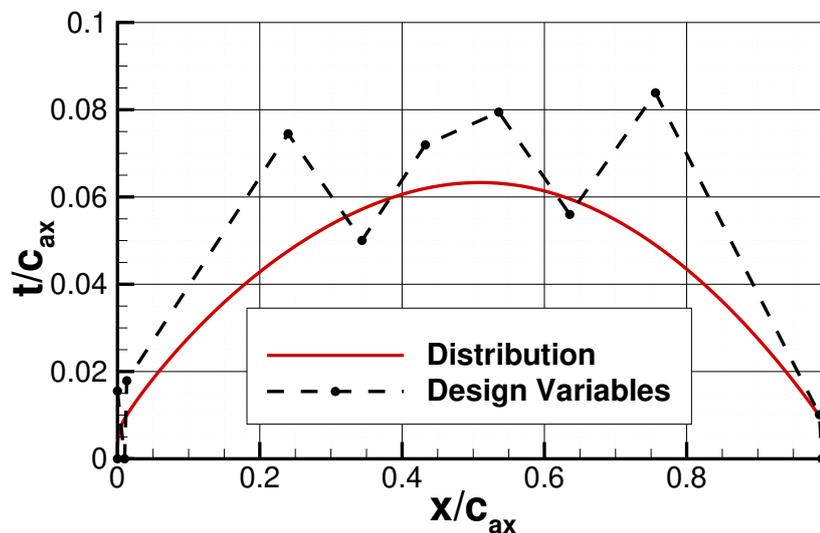


Figure 3.2: Thickness design variables and distribution for NASA stage 35.

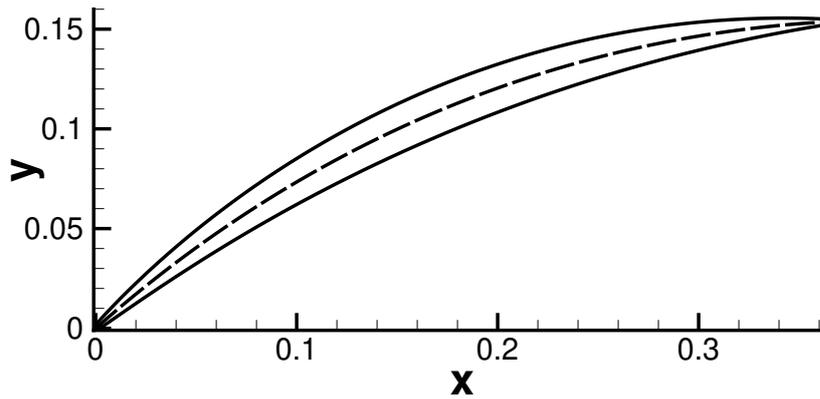


Figure 3.3: NASA stage 35 stator blade geometry constructed by the tool.

3.1.2. Topology

For aerodynamic analysis, the selection of the flow domain surrounding the body of interest is also important. In the case of a turbomachinery blade, the relevant flow domain can be represented in several ways. The 2D blade geometry described previously can be seen as a slice of a three-dimensional compressor along the blade-to-blade plane. Only one blade row is considered. This leaves a series of identical blades repeating in the circumferential direction. Since the blades are identical and repeat periodically, only the flow around a single blade needs to be represented, reducing computational cost.

Thus only a single, 2D blade needs to be represented. The domain can be defined by a single blade passage, delimited by the pressure side of one blade and the suction side of the next. Alternatively, the area directly surrounding a single blade can be represented. The two options can be compared in figures 3.4 and 3.5. Since the inverse design target to be formulated relates to the continuous distribution of a parameter around the blade surface, it makes sense to represent the flow domain with a continuous blade as well. The flow domain surrounds the blade and extends to half the pitch both above and below the blade. The curvature of the edges of the domain follows the shape of the blade camber line.

This leaves the inlet and outlet areas to be defined. The inlet area does not need to be large, as there are only small changes in the flow field until the blade is encountered. Vertically, the inlet area is as tall as the pitch, centred on the leading edge. The inlet boundary is placed 20% of the axial chord length in front of the leading edge, so that upstream flow effects can be accounted for. The outlet vertical size is again one pitch-length and is centred on the trailing edge. In order to allow for the wake to develop and mixing losses to be taken into account when evaluating the blade, the outlet boundary is placed one axial chord-length downstream from the trailing edge.

A multiblock structured grid will be created. Dividing the domain in smaller blocks allows for faster mesh generation and fast, accurate flow and adjoint solutions [65]. A multiblock grid is used so that the grid pattern can be changed where advantageous. It also provides for simple control of the distribution of cells throughout the domain. The domain is divided into seven blocks. For each block the grid pattern, cell numbers and distribution on the boundaries and connectivity to other blocks is specified. For most of the flow domain a H-grid pattern is the most straightforward choice.

The first block is given a constant offset from the blade surface, and thus surrounds the blade. A second and third block are placed directly upstream and downstream of the first block respectively. The positioning of these blocks allows for a locally finer cell spacing at the blade leading and trailing edges. A fourth block is placed above the three previous blocks. This block connects to the top boundary. Similarly, a fifth block covers the bottom segment. Due to limitations in the CADO-software, only one block can be located on both the inlet and outlet boundaries. Therefore a sixth and seventh block are defined to connect to those boundaries respectively. The lay-out of the block structure is presented in figure 3.6. Note that the seventh block does stretch out beyond the image, but was cropped for better overview.

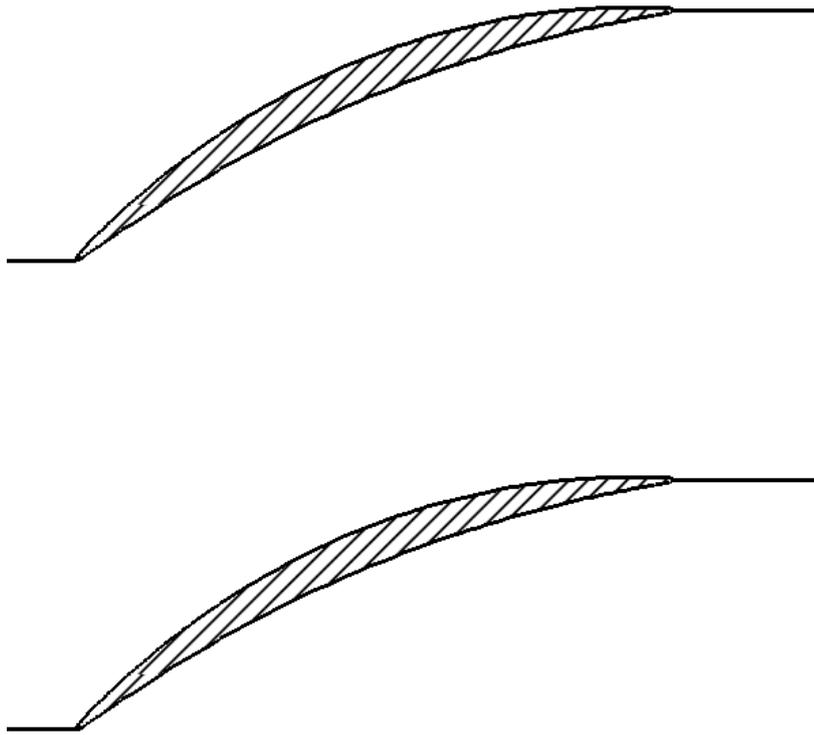


Figure 3.4: Flow domain formed by the passage between two blades.

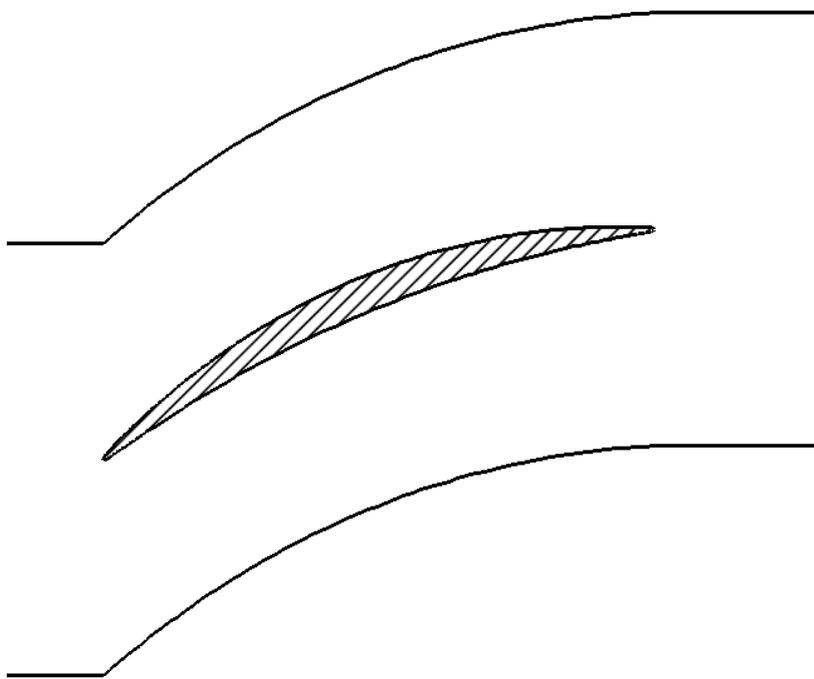


Figure 3.5: Flow domain surrounding a single blade.

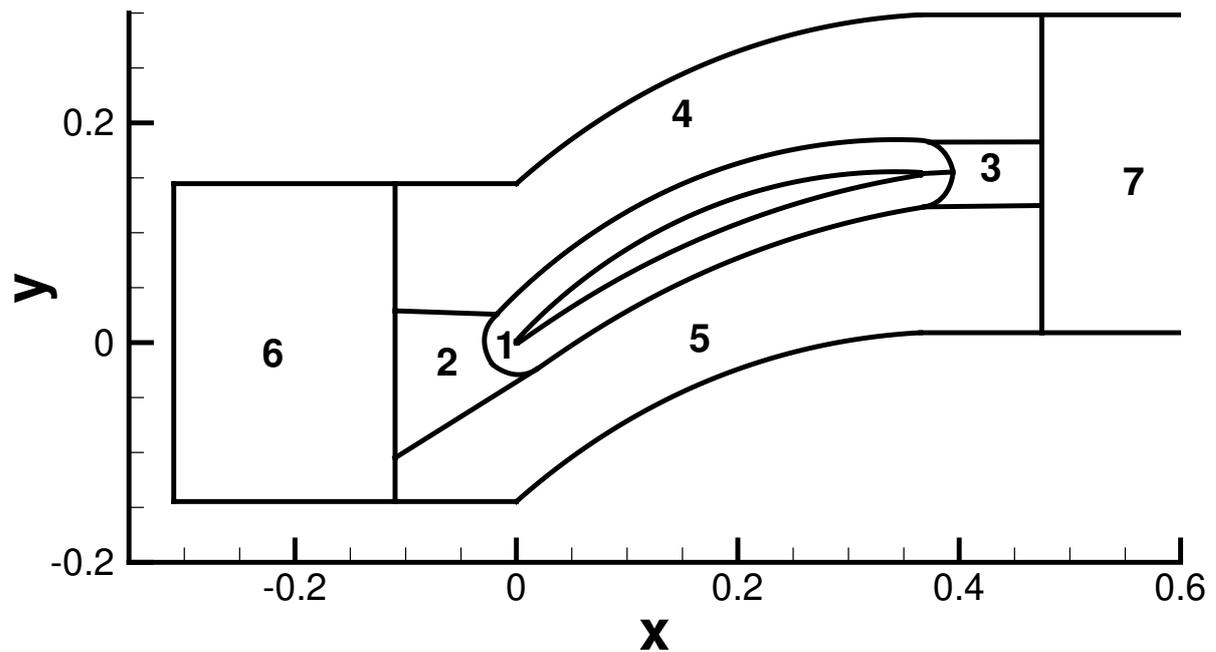


Figure 3.6: Multiblock structure used in mesh generation.

3.1.3. Mesh Generation

The geometries and connectivity of the blocks need to be defined so that the grid generator and later the CFD solver can access the neighbouring cells across block boundaries. An O-grid pattern is used in the first block in order to better discretise the rounded leading and trailing edges of the blade. The remaining blocks use a simple H-grid pattern. In order to generate an initial mesh, the number of cells on each of the block boundaries is specified. Naturally, the number of cells specified on each boundary needs to be consistent. Each grid point connects to four cells, and each cell has four grid points, with one exception. The grid points on the corner of block one and any two other blocks are adjacent to 5 cells, due to the change of grid pattern. The simplest way to keep the cell numbers consistent is to keep the number of cells on the left and right borders and top and bottom borders of each block equal. More complex approaches are possible, but since the geometries considered are relatively simple, this simple approach will yield meshes of sufficient quality.

Not only does the number of cells on each boundary need to be specified, but also their distributions along those edges. For most edges a uniform distribution is appropriate, so that each cell has the same width. Near the blade more care is required. The leading and trailing edges are regions of high curvature and fine concentration of cells along the blade surface is needed in these regions. Towards the middle of the blade, curvature is lower and cells can be stretched out more. A stretching function is used on the blade SS and PS curves, resulting in the desired refinement where necessary. A one-sided stretching is also applied on the boundaries between block 1 and blocks 2 and 3, so that the refinement is applied directly upstream of the leading edge and directly downstream of the trailing edge. It is also important to control the height of the cells at the blade wall. In order to be able to capture turbulent phenomena near the wall, the first cell layers must not be too tall. The height of the first cell is specified to reach a y^+ value between 1 and 5. An initial estimate for this height is used, which can later be updated based on the flow solution if needed. The remaining lines of the O-grid are also spaced using the one-sided stretching.

Special care must also be taken at the top and bottom boundaries. Only a single blade is represented in the model, but in practice this blade would repeat periodically. The domain was selected so that it ends halfway to the blades above and below. These boundaries are periodic, meaning that values at the top should match those at the bottom of the domain. The grid lines also need to match along these boundaries. The edges at the top are selected as master and the bottom counterparts as slave, so that the locations of the grid lines along the bottom always follow from those at the top.

Based on these specifications, the grid construction tool in CADO is able to automatically generate a lattice. This lattice follows the specified cell numbers, connections, thicknesses and stretching. However, it is quite rectilinear and block boundaries are easily recognisable. Such a mesh is not suitable for CFD-analysis. Therefore, a grid-smoothing algorithm is applied. The grid smoother solves the Poisson equation to reach a smoothed grid, similar to the description of electric field and potential lines. The source term is needed to create local refinement and is formulated using the method developed by Steger and Sorenson [66]. This significantly improves mesh quality. The resulting meshes are suitable for CFD analysis. To ensure good quality, some criteria can be checked. These are the conventional values of cell aspect ratio, presence of negative cell volumes, cell angles and the expansion ratio between cells. The same mesh on the NASA 35 blade before and after smoothing can be seen in figures 3.7 and 3.8 respectively.

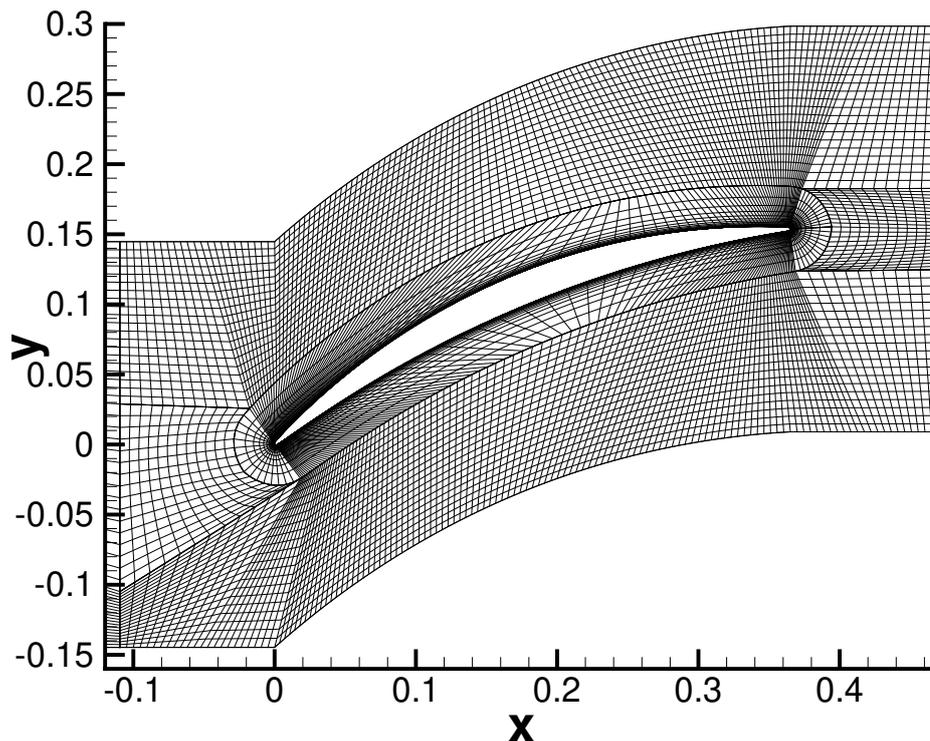


Figure 3.7: NASA stage 35 2D grid before smoothing.

A mesh of good quality has now been created, but it remains two-dimensional. The CFD solver in CADO is only capable of 3D flow solutions. The mesh is converted to three dimensional space by extruding the 2D mesh to create two thin cell layers in the new z -direction. The new 3D cells are regrouped in their respective blocks. The connectivity between the blocks needs to be defined again by specifying the transformation matrix along each boundary. These matrices contain the operation needed to transfer between the cell indices of one block to the other along the boundary.

Cells are given a three-dimensional index value (i,j,k) in each block. The cell with index $(1,1,1)$ is at the lowest x,y and z -values of the block. Adjacent cell i, j or k index values increase by one depending if they are positioned at higher x, y , or z -coordinates respectively. Hence the cell index system is right-handed like the geometrical coordinate system. For the first block surrounding the blade, the cell with the lowest index is at the trailing edge. The i -values increase radially and the j -values increase along the blade surface in counterclockwise direction until the trailing edge is reached again. Since all blocks follow a same right-handed coordinate system, the transformation matrices are straightforward. The connectivity with block 1 is a bit more involved due to the different grid-pattern, but can easily be recognised and specified. The transformation matrices are also specified for the periodic boundaries at the top and bottom of the domain, since they are seen as adjacent by the solver.

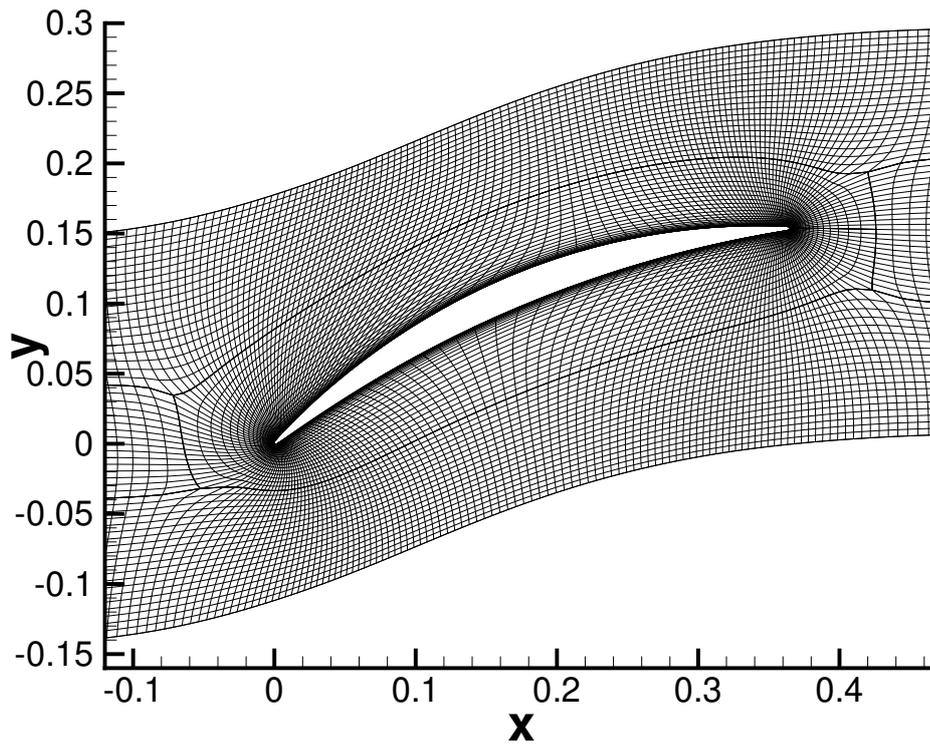


Figure 3.8: NASA stage 35 2D grid after smoothing.

Finally the boundaries conditions are specified on the remaining faces which are not adjacent to another block. The front face of block 6 is specified as the inlet and the back face of block 7 as the outlet. All blocks also have boundaries on the planes with the lowest and highest z -coordinates. These are designated as symmetry planes. No effects of hub or shroud are to be accounted for. The mesh is saved in the CGNS-format and is ready to be used by a flow solver.

3.1.4. Flow Solution

The last step of the forward analysis is acquiring the flow solution. The mesh and boundaries have already been defined, but the flow solver still needs to be configured. The flow solver used is part of the CADO software package. This solver makes use of two levels of abstraction to approximate the physical reality. These are the physical and numerical models. The blade optimisation will only be based on time-averaged phenomena since no large flow separations or other unsteady effects should be present near the design points. The RANS equations will therefore be used to model the flow. The set of equations is completed using the Spalart-Allmaras turbulence model. This model introduces only one additional equation, limiting the computational cost. The model was developed for wing profiles, but is considered suitable for 2D turbomachinery blades as well and is quite robust.

The fluid surrounding the blade is air. It is represented as a calorically perfect gas with a specific heat ratio of 1.4 and specific gas constant of 287.0 J/kg·K. Some physical conditions at the boundaries also need to be specified. At the inlet the total pressure, total temperature and vector of the incoming velocity are defined. A Riemann boundary condition is used. This implies a circumferentially homogeneous flow, representing the average flow state at the stage inlet. The static pressure at the outlet is also specified. The specific values depend based on the case analysed. Flow solutions need an initial state to start the computation. If not provided in the form of a previous flow solution, a standard uniform flow field is prescribed. This can be modified by the designer to match more closely the expected result. Note that the flow solution is performed in the relative frame of reference. Consequently the relative inlet flow angle and total quantities need to be used for a rotor stage. An overview of the required physical parameters is given in table 3.3. The assigned values may vary depending on the case considered.

Table 3.3: Physical parameters of the CFD simulations.

Parameter	Units
Specific heat ratio	-
Specific gas constant	J/kg·K
Inlet total pressure	Pa
Inlet total temperature	K
Inlet axial velocity component	-
Inlet meridional velocity component	-
Outlet pressure	Pa
Initial domain pressure	Pa
Initial domain temperature	K
Initial domain axial velocity	m/s
Initial domain meridional velocity	m/s

The numerical modelling covers the representation of the continuous flow equations in a discrete manner. This requires both spatial and temporal discretisation. A Roe-type upwind scheme is used for the spatial discretisation. MUSCL extrapolation is used to reach second order accuracy. Temporal discretisation is done using an implicit Runge-Kutta scheme. The scheme is accelerated using local time-stepping. The system of equations is solved using a GMRES linear solver. To speed up the computation of a flow solution, the flow equations are solved in several blocks in parallel using MPI. The blocks are divided between processors based on the number of cells in each block.

When starting a new flow analysis, the solver is set to first initialise the solution on an automatically generated coarser grid. After 100 iterations the solver uses a multigrid cycle between the fine and coarse grid. The solver terminates when the maximum number of 1000 iterations is reached or the convergence criterion is satisfied. This occurs when the relative density residual is less than 10^{-14} . Convergence so close to machine accuracy is possible due to the strong stabilising behaviour of the implicit time discretisation method. This suppresses the more typical limit cycle oscillations caused by unsteady phenomena in the flow. This strong convergence in turn improves the stability of the adjoint solution process.

The configuration of the flow solver is stored in a text document together with settings for the parametrisation, topology and meshing steps. These functions are all linked in an encompassing axial compressor analysis program. This program can be called and is provided the configuration file so that it executes each of the steps in order. The resulting flow solution is stored in the CGNS-file and can be analysed with data visualisation and processing software such as Tecplot. Some initial post-processing is also automatically provided upon completion of the flow solution. A report is created containing the mass-averaged flow conditions and velocities at the inlet and outlet. Some performance metrics are also computed, namely the static and total pressure and temperature ratios as well as the total pressure loss coefficient ω , defined by equation 3.2.

$$\omega = \frac{P_{t,in} - P_{t,out}}{P_{t,in} - P_{in}} \quad (3.2)$$

3.2. Adjoint analysis

The set-up of the program so far has been rather straightforward and generic. The tool is now capable of interpreting a set of design variables and computing the flow field surrounding the corresponding geometry. The next task is for the program to compute the desired objective function value and find its gradients with respect to all the design variables. The gradients are computed using a discrete adjoint approach. The formulation of the objective function for inverse design is discussed below, followed by the implementation of the adjoint computation. The adjoint solver computes the gradient of the objectives with respect to the grid point coordinates. To reach the objective gradients with respect to the design variables, the gradients of these grid points with respect to the design variables need to be separately computed in a final gradient construction step.

3.2.1. Inverse Design Objective

Inverse design means that a certain target is specified to which the optimiser tries to converge. The inverse design target should lead to a unique blade geometry and contain information on the expected performance of the corresponding design. Two conventional ways of evaluating 2D blade designs are the pressure and isentropic Mach number distributions along the blade surface. These distributions contain a lot of information on the blade performance, such as the flow turning achieved and the state of the boundary layer. An experienced designer can see ways in which modification to these distribution would lead to improved blade performance. Both are therefore a suitable choice as inverse design target. The isentropic Mach number and pressure distributions are closely linked through the isentropic flow equations, making the inclusion of both options simple. Based on these target distributions, a scalar objective value needs to be computed. This value represents how close the current design matches specified target and is the value minimised during optimisation.

The selection of the inverse design objective and target distribution type are included in the previously discussed configuration file. The target distribution itself is separately provided to the program. This file contains the coordinates of the distribution along both the SS and PS edges of a two-dimensional blade. The x -values range from zero to one and represent the non-dimensional position along the blade surface from LE to TE. The y -values contain the pressure value or isentropic Mach number value at this location. The number and distribution of points in the target is free for the designer to choose, but should be sufficient to accurately represent the desired distribution when linearly interpolated. It is advantageous to sample the target distribution points with a cosine distribution along the x -coordinate, since the curvature of the distribution is typically larger near the leading and trailing edges of the blade.

The corresponding distribution of the current blade design also needs to be fetched to compute the objective. This is the pressure or isentropic Mach number distribution of the design that has previously been evaluated using the flow solver. The required information is saved in a CGNS-file. The flow solver saves the flow properties, including pressure, in the centre of every cell. The values of the first cells along the boundary do not exactly match the values on the boundary itself. The local pressure value in the centre of a boundary face between two cells can be approximated by averaging the values of the adjacent cells. Despite the flow domain ending at the blade wall, there are still cells on both sides of this boundary. Cells outside the flow domain are called ghost cells. These are created so that the solver can apply differencing and interpolation schemes at the boundary. In a stationary domain, the velocity in the ghost cells usually opposes that of the adjacent real cell, but has equal density and pressure. Furthermore, the blade flow solution is computed on a 3D mesh. To return to a 2-dimensional distribution, the pressure values are averaged along the z -direction as well.

The pressure at a certain location along the boundary is thus calculated with equation 3.3. $P(i, j, k)$ is the pressure of the cell with indices i, j and k . The cells along the blade wall all have index i_{\min} and the ghost cells have index $i_{\min}-1$. Since there are only two cells in the z -direction, the k -indices are always either k_{\min} or k_{\max} . The j -index ranges from its minimum to its maximum along the blade wall.

$$P(j) = 0.25 \cdot (P(i_{\min}, j, k_{\min}) + P(i_{\min} - 1, j, k_{\min}) + P(i_{\min}, j, k_{\max}) + P(i_{\min} - 1, j, k_{\max})) \quad (3.3)$$

Next to the pressure value, the location of each sample along the blade surface should also be recorded. For each cell index j , the distance between the corners adjacent to the wall is computed from the grid point coordinates. This is done on the plane between the k_{\min} and k_{\max} cell layers. This distance approximates the arc length of the blade surface covered by that cell, $\Delta S(j)$. The sum of all these values gives the total arc length of the blade edge S_{tot} . The location $S(j)$ of each pressure value $P(j)$ is computed with equation 3.4. The non-dimensionalised length $S_{\text{ND}}(j)$ can then be computed by dividing with the total arc length. The pressure at the TE itself has not been evaluated, but is required since it is both the start and end point. The value is estimated by averaging the pressure of the first and last segments.

$$S(j) = \left(\sum_{n=j_{\min}}^{j-1} \Delta S(n) \right) + 0.5 \Delta S(j) \quad (3.4)$$

The flow solver also stores the Mach number in each cell. However, for viscous flows the velocity, and thus also the Mach number, is 0 at the wall. Instead, the Mach number just outside the boundary layer can be computed. It is assumed that the local pressure is the same as the pressure in the boundary

layer on the wall. Using the isentropic relation in equation 3.5, the corresponding Mach number can then be derived. This assumes an isentropic process for the flow between the inlet plane and the local point outside the boundary layer. The total inlet pressure $P_{t,FS}$ is required for this computation. Due to numerical inaccuracies it could be that the local pressure is stored as being slightly higher than the total inlet pressure near the LE. In this case the Mach number is capped off as being zero.

$$M_{is}(j) = \sqrt{\left[\left(\frac{P_{t,FS}}{P(j)} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right] \cdot \frac{2}{\gamma-1}} \quad (3.5)$$

The distribution of the current design needs to be split between the pressure and suction sides, just like the target distribution is. The vertex corresponding to the leading edge is not stored in the flow solution. However, the geometry was constructed so that the leading edge is always located in the origin. The sample location closest to the geometrical coordinates (0,0,0) indicates where the leading edge is passed. All values up until and including this point are reversed in order and stored as the suction side values. The values from this point onward are stored as the pressure side values. The S_{ND} -values, which represent the location along the surface, are rescaled to range from 0 to 1 for both series.

After these steps, the distribution of the current design and the target distribution have been formulated in the same way and are of the same type. The objective to be calculated is the difference between the two distributions. Analytically, this can be expressed as:

$$J = \int_0^1 (Y_{target}(S_{ND}) - Y_{current}(S_{ND}))^2 dS_{ND}. \quad (3.6)$$

The integral is discretised by computing the difference on a number of evaluation points. The number of points describing the current distribution is free to pick by the designer. For the current design distribution, this depends on the number of cells on the blade wall. To ensure a consistent calculation of the difference between both distributions, the number of points where the difference is evaluated is independently defined.

These points are spread on a cosine-distribution along the S_{ND} -axis between 0 and 1. There is a higher concentration of points at the LE and TE where distributions are changing more rapidly. The integral is calculated using the midpoint rule. Each of the evaluation points n has an associated S_{ND} -fraction S_{frac} . This fraction indicates the range of S_{ND} -values where point n is the closest evaluation point. This fraction gives less weight to the more closely spaced points, so that each segment of the distributions has equal weight. The fractions are computed with equation 3.7. For the first and last point, the fraction is simply half the distance to the next and previous point respectively.

$$S_{frac}(n) = \frac{S_{ND}(n+1) - S_{ND}(n-1)}{2} \quad (3.7)$$

At each evaluation point n the distribution Y -values are linearly interpolated based on the distances along the x -axis between the evaluation point and the surrounding points describing the distributions using equation 3.8. Indices j_{prev} and j_{next} respectively indicate the previous and next point defining the distribution with respect to the evaluation point. The interpolation is done on both the current design and target distributions. Note that the Y -value could represent either pressure values or isentropic Mach numbers, depending on the type of target distribution.

$$Y(n) = \frac{S_{ND}(n) - S_{ND}(j_{prev})}{S_{ND}(j_{next}) - S_{ND}(j_{prev})} \cdot [(Y(j_{next}) - Y(j_{prev}))] + Y(j_{prev}) \quad (3.8)$$

The difference between the interpolated values for both distributions is computed on the SS and PS curves. These differences are squared, multiplied with their respective S_{ND} -fractions and multiplied with 1000. All the resulting values are summed up to obtain the inverse design objective value. The multiplication by 1000 simply serves to reach a typical objective value between 1 and 10 for most distributions. The objective equation function is finally expressed as:

$$J = \sum_{n=1}^N [\Delta Y_{SS}(n)^2 \cdot S_{frac}(n) \cdot 1000 + \Delta Y_{PS}(n)^2 \cdot S_{frac}(n) \cdot 1000] \quad (3.9)$$

where N is the total number of evaluation points and ΔY represents the difference between the target and current design distribution Y -values. As indicated, the differences are computed on both the SS and PS. When solving an inverse design problem, the objective values are stored in the automatically generated post-processing report together with the other performance metrics.

3.2.2. Adjoint Solution

Now the inverse design objective has been defined, its gradient computation can be developed as well. A discrete adjoint approach is used. This means that the adjoint equations are formulated based on the discretised flow and objective equations. This process is relatively straightforward. The steps in computing the objective are followed in reverse order and differentiated at every step. An adjoint solver has already been developed for CADO in this way. The computations unique to the inverse design objective still need to be derived. Computing the adjoint solution yields the objective gradient with respect to each of the grid point coordinates dJ/dX . Thus at this stage, the solution of the equation 3.10 is computed.

$$\frac{dJ}{dX} = \psi^T \cdot \frac{\delta R}{\delta X} + \frac{\delta J}{\delta X} \quad (3.10)$$

As outlined in Chapter 2, this equation is derived from the more straightforward formulation in equation 3.11. The set of adjoint variables ψ is determined by the iterative adjoint solver using equation 3.12. The factors $\delta R/\delta U$ and $\delta R/\delta X$ represent the change in the residuals with respect to the flow properties and mesh respectively. The first of these is the Jacobian matrix and is computed when starting the adjoint solver. The second is computed after the adjoint solution has been obtained and the objective gradient is to be calculated.

$$\frac{dJ}{dX} = \frac{\delta J}{\delta U} \cdot \frac{dU}{dX} + \frac{\delta J}{\delta X} \quad (3.11)$$

$$\left(\frac{\delta R}{\delta U} \right)^T \cdot \psi = - \left(\frac{\delta J}{\delta U} \right)^T \quad (3.12)$$

Two terms remain in the equations to be computed. Both depend on the objective function and thus need to be derived for the new inverse design objective. The first is the term $\delta J/\delta U$ in equation 3.12 represents the direct sensitivity of the objective function with respect to the flow solution. Since the objective is calculated at the blade wall, only the flow properties in the cells along the blade wall will contribute to this term. Similarly, the second term $\delta J/\delta X$ in equation 3.10 portrays the direct objective function grid sensitivity and is only dependent on the coordinates of the cells along the blade wall.

For the calculation of the inverse design objective, only the pressure values and coordinates of the cells along the wall were used as design-specific input. Thus the discrete differentiation will yield both of the needed terms. The following notation is used for the adjoint values:

$$\bar{X} = \frac{\delta J}{\delta X}.$$

The next relation is useful to keep in mind when following the derivations underneath. It indicates that the adjoint value of variable Y on which the variable X is dependent can be found by multiplying the adjoint value of variable X with the partial derivative of variable X with respect to variable Y . This is simply an application of the chain rule and forms the basis for the entire reverse differentiation process.

$$\bar{Y} = \frac{\delta J}{\delta Y} = \frac{\delta J}{\delta X} \cdot \frac{\delta X}{\delta Y} = \bar{X} \cdot \frac{\delta X}{\delta Y}$$

Reverse Differentiation

The inverse design objective calculation is hand-differentiated in reverse mode. It can be seen that by definition $\bar{J} = 1$. First, equation 3.9 is differentiated with respect to the difference in Y -value between current design and target distributions. These operations are executed in parallel for both the SS and PS values but are illustrated for the SS only:

$$\overline{\Delta Y_{SS}(n)} = \bar{J} \cdot 2\Delta Y_{SS}(n) \cdot S_{frac}(n) \cdot 1000$$

The difference in Y -value is simply found by subtracting the current design value from the target value:

$$\overline{Y_{current,SS}(n)} = -\overline{\Delta Y_{SS}(n)}$$

The Y_{current} value was linearly interpolated using equation 3.8. This equation can then be derived as follows:

$$\begin{aligned} \overline{S_{\text{ND}}(j_{\text{next}})}_+ &= \overline{Y_{\text{current,SS}}(n)} \cdot (Y(j_{\text{next}}) - Y(j_{\text{prev}})) \cdot \frac{S_{\text{ND}}(j_{\text{prev}}) - S_{\text{ND}}(n)}{[S_{\text{ND}}(j_{\text{next}}) - S_{\text{ND}}(j_{\text{prev}})]^2} \\ \overline{S_{\text{ND}}(j_{\text{prev}})}_+ &= \overline{Y_{\text{current,SS}}(n)} \cdot (Y(j_{\text{next}}) - Y(j_{\text{prev}})) \cdot \\ &\quad \left[\frac{S_{\text{ND}}(n) - S_{\text{ND}}(j_{\text{prev}})}{(S_{\text{ND}}(j_{\text{next}}) - S_{\text{ND}}(j_{\text{prev}}))^2} - \frac{1}{S_{\text{ND}}(j_{\text{next}}) - S_{\text{ND}}(j_{\text{prev}})} \right] \\ \overline{Y(j_{\text{next}})}_+ &= \overline{Y_{\text{current,SS}}(n)} \cdot \frac{S_{\text{ND}}(n) - S_{\text{ND}}(j_{\text{prev}})}{S_{\text{ND}}(j_{\text{next}}) - S_{\text{ND}}(j_{\text{prev}})} \\ \overline{Y(j_{\text{prev}})}_+ &= \overline{Y_{\text{current,SS}}(n)} \cdot \left[1 - \frac{S_{\text{ND}}(n) - S_{\text{ND}}(j_{\text{prev}})}{S_{\text{ND}}(j_{\text{next}}) - S_{\text{ND}}(j_{\text{prev}})} \right] \end{aligned}$$

As some sample points may be referred to more than once, the various contributions are added to each other using the addition assignment operator whilst repeating for every evaluation point n . This results in lists of adjoint values for the coordinates of the SS and PS current design distributions. These lists are re-ordered and scaled back into a single list corresponding to the initially obtained distribution, so that it covers the entire blade in counterclockwise direction starting from the trailing edge. If the distribution contains the isentropic Mach numbers, the wall pressure adjoint values are obtained by deriving equation 3.5 with respect to the wall pressure as shown below. If at a location the Mach number was capped off to zero, the corresponding pressure adjoint value is set to zero.

$$\begin{aligned} \overline{P(j)} &= \overline{Y(j)} \cdot \frac{-1}{\sqrt{\left[\left(\frac{P_{t,\text{FS}}}{P(j)} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right] \cdot \frac{2}{\gamma-1}}} \cdot \left(\frac{P_{t,\text{FS}}}{P(j)} \right)^{\frac{\gamma-1}{\gamma}-1} \cdot \frac{P_{t,\text{FS}}}{\gamma \cdot P(j)^2} \\ &= -\overline{Y(j)} \cdot \frac{P_{t,\text{FS}} \cdot \left(\frac{P_{t,\text{FS}}}{P(j)} \right)^{\frac{-1}{\gamma}}}{\sqrt{2\gamma \cdot P(j)^2 \left[\left(\frac{P_{t,\text{FS}}}{P(j)} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right] / (\gamma-1)}} \end{aligned}$$

Now a single list has been obtained containing the adjoint values for the pressure value $P(j)$ and non-dimensional location $S_{\text{ND}}(j)$ of each cell along the blade boundary. The adjoint of the dimensional positions along the blade is found by dividing by the total arc-length of the blade edge, S_{tot} . The total arc-length itself also has an adjoint value, which is added to by the computation of the non-dimensional arc length at every point. The computations below are executed for every index j .

$$\begin{aligned} \overline{S(j)} &= \overline{S_{\text{ND}}(j)} / S_{\text{tot}} \\ \overline{S_{\text{tot}}}_+ &= \overline{S_{\text{ND}}(j)} \cdot (-S(j)) / S_{\text{tot}}^2 \end{aligned}$$

The pressure value at the trailing edge was computed as the average between the first and last cell. The TE pressure adjoint values therefore also need to be factored in the pressure adjoint values of the first and last cells. The adjoint value of its dimensional position also contributes to the adjoint value of the total edge arc-length. The pressure values on the boundary were averaged between the two cell layers at k_{min} and k_{max} . Hence, the boundary pressure adjoint values for the cells in each layer can simply be found by equally dividing the averaged boundary pressure adjoint values.

The final step in the reverse differentiation of the inverse design objective is to return from positions based on arc length to the vertex coordinates. Each cell covers a segment of the total arc length $\Delta S(j)$,

the adjoint value of which consists of contributions by the local position and the total arc length adjoint values. At every cell the local position was computed from the total arc length up until the previous cell and adding half the local increment as seen in equation 3.4. To find the adjoint value of the segment lengths, this loop is executed in reverse and the total arc-length adjoint value is updated on every step accordingly. The computation below is thusly executed for every index j , starting from j_{\max} .

$$\begin{aligned}\overline{\Delta S(j)} &= \overline{S(j)}/2 + \overline{S_{tot}} \\ \overline{S_{tot}+} &= \overline{S(j)}\end{aligned}$$

These segment lengths were in turn calculated based on the distance between the grid points on the boundary and on the plane between the two cell layers. This step simply uses the formula for the distance between two points in 3D Cartesian space. That formula can be derived with respect to the x , y and z coordinates of both vertices. The vertex common with the cell at one lower j -value is called vertex 1. The vertex in common with the cell at one higher j -value is named vertex 2. The adjoint value of the x -coordinate of this second vertex for a cell with index j can be calculated using:

$$\overline{V_{2,x}(j)+} = \overline{\Delta S(j)} \cdot \frac{1}{\overline{\Delta S(j)}} \cdot (V_{2,x}(j) - V_{1,x}(j))$$

where the subscripts indicate which vertex and which coordinate is evaluated. The computations for $\overline{V_{2,y}(j)}$ and $\overline{V_{2,z}(j)}$ are analogous. The computations of the adjoint values of vertex 1 follow a similar equation, but with the order of the coordinates in the last terms switched. Since vertex 1 for a cell with index j is the same as vertex 2 corresponding to the cell with index $(j - 1)$, only the values for vertex 2 are stored. The adjoint values of vertex 1 coordinates are added to $\overline{V_2(j - 1)}$. These computations are again repeated for all indices j .

Flow and Grid Sensitivities

With this the reverse derivation is complete and the gradients of the objective with respect to the boundary pressure values as well as the grid coordinates on the boundary have been computed. This information can then be used to compute the objective flow sensitivity, which is needed for solving the adjoint equation (3.12). The blade wall pressure is the only flow quantity directly impacting the objective and is provided by the reverse differentiation. However, the wall pressure is determined by the pressure of the adjacent domain and ghost cells. In turn, the ghost cell pressure is determined by the adjacent domain cell pressure as well. Furthermore, the CADO solvers do not compute directly with pressure and velocity values. Instead these values, named primitives, are converted to and from conservative values used for the conservation equations. Thus to compute the flow sensitivity term, the adjoint value of the boundary pressure value needs to be converted to the conservative adjoint values of the first domain cell. The conversion is done using the following equation:

$$\frac{\delta J}{\delta U_{c,D1}} = \frac{\delta J}{\delta U_{p,Bnd}} \cdot \left(\frac{\delta U_{p,Bnd}}{\delta U_{p,D1}} + \frac{\delta U_{p,Bnd}}{\delta U_{p,G1}} \cdot \frac{dU_{p,G1}}{dU_{p,D1}} \right) \cdot \frac{dU_p}{dU_c} \quad (3.13)$$

The subscripts signal whether the flow quantities are in primitive or conservative forms and to which location they correspond. The first factor is provided by the reverse differentiation of the inverse design objective. The last factor converts between primitive and conservative values and is simply given by the definition of the conservative values in CADO. The terms between brackets also turn out to be simple. The boundary pressure was calculated by averaging the ghost and domain cell pressure values, so the boundary pressure gradients are 0.5 with respect to both the ghost and domain cell pressure values. The dependency of the ghost cell on the domain cell is available in the program, and simply turns out to be one-to-one for the pressure value. Thus the entire group between brackets simplifies to 1 for the pressure values. No other primitive flow quantities have a contribution.

The objective flow sensitivity is now in the correct form for solving the adjoint equations. The iterative adjoint solver can thus be started and the adjoint values ψ are computed. This process does not depend any further on the objective specifically. After the adjoint solver converges and the adjoint values have been computed throughout the flow field, the objective gradient with respect to the grid coordinates can be computed using equation 3.10. Here the gradient computation in CADO makes

one important simplification. The computation of the partial derivative of the turbulence equation with respect to the grid, $\delta R_{\text{turb}}/\delta X$, has not yet been implemented. The effect of changes in the grid on the turbulent quantities, which in turn affects the objective, is therefore not accounted for. In general, this effect is expected to be small compared to the overall objective gradient. This means that the gradients should still be of sufficient accuracy. A slightly offset gradient should still lead to successful optimisation. However, the accuracy of the gradients and performance of the optimiser needs to be verified.

One more term is needed to solve equation 3.10, namely the grid sensitivity term $\delta J/\delta X$. The reverse differentiation of the objective already provides the required information. The adjoint values of the grid coordinates on the blade wall are directly computed. These adjoint values only correspond assigned to the grid points on the plane between the two cell-layers at index k_{min} and k_{max} . The grid coordinates on other planes were not used to compute the objective value.

For a rotating blade, the grid point coordinates also have an additional impact on the objective value. Due to the rotational velocity varying along the blade span, the flow quantities along the blade boundary have a direct dependence on the grid coordinates. The objective value is in turn also dependent on the values on the boundary. Since this effect does not impact the flow residuals, this gradient has not been accounted for in the flow sensitivity term. Instead, this effect is part of the direct objective grid dependency as shown in equation 3.14. This computation is included to be consistent within CADO and for future compatibility with 3D blade designs. Nevertheless, the current analysis is restricted to 2D blade segments, and thus span-wise rotational effects will not impact the objective gradients.

$$\left(\frac{\delta J}{\delta X}\right)_{\text{rot}} = \frac{\delta J}{\delta X} + \left(\frac{\delta J}{\delta U_{p,\text{Bnd}}} \cdot \frac{\delta U_{p,\text{Bnd}}}{\delta X}\right) \quad (3.14)$$

3.2.3. Gradient Construction

With all of the terms now known, equation 3.10 can be solved. This yields the gradient of the objective with respect to each of the grid coordinates for all grid points in the mesh. To reach the gradient of the objective with respect to the design variables, multiplication with the gradient of each coordinate with respect to the design variables is required, shown in equation 3.15.

$$\frac{dJ}{d\alpha} = \frac{dJ}{dX} \cdot \frac{dX}{d\alpha} \quad (3.15)$$

These mesh gradients can be computed by using an adjoint strategy again or by calculating the gradients in a forward manner. However, the grid smoothing algorithm used in CADO has not yet been derived. This makes an adjoint computation impossible. The gradients are therefore computed using the complex step method. This method is similar to finite differencing, with the difference that a complex-valued perturbation is applied. The gradient can then be computed with second order accuracy and numerical stability issues are avoided [27]. Values and operations in CADO are implemented in such a way that a switch from real to complex values is possible without changing the code. This makes the use of the complex step approach in CADO as simple as assigning a complex variable to a design variable in the design configuration file. When generating the mesh, the program will now not only output the mesh itself. It will also compute the gradient of the grid point coordinates with respect to this design variable. This is simply done by dividing the imaginary value of the coordinates with the step size.

The complex step computations can be executed in parallel with the computation of the dJ/dX factor. A complex step computation takes roughly twice as long as its equivalent real-valued operation. However, the mesh generation step is significantly faster than the adjoint-solution. The complex step evaluations for each of the design variables can therefore all be completed during the adjoint solution process without slowing down the program. Finally, the objective sensitivity with respect to each design variable can be computed using equation 3.15. With this, the gradient computation is complete.

Before starting any optimisations using these gradients, the quality of these gradient computations has been verified by comparing with objective gradients computed completely using the complex step and finite difference methods. The finite difference computations are executed on a range of step sizes. This is done because the gradient accuracy will increase with a reducing step size, until numerical instability due to truncation and round-off errors again result in poorer gradient predictions. The step-size giving the most accurate gradient can thus be identified through comparison with the complex

step and adjoint gradients. If both finite difference and complex step are consistent, this indicates that they provide a good reference to evaluate the gradients computed using the adjoint method.

3.3. Optimisation

With the direct and adjoint analysis functions complete, blade geometries can be constructed, evaluated and the objective gradients computed. A gradient-based optimisation algorithm can then be used to modify the design so that the objective value is minimised. Many first-order optimisation methods exist, each with their advantages and disadvantages. A very simple method is the steepest descent method, which simply changes the design by taking a step opposing the local gradient. However this method tends to converge slowly. Other methods, such as those called quasi-Newton methods, are able to better use the available information gathered over different iterations, resulting in faster convergence.

One effective optimisation algorithm is SQP. It is suitable used for non-linear, constrained optimisation problems, making it suitable for a wide array of engineering problems. Two SQP-type algorithms have been implemented in CADO. The first is the freely available and open-source SLSQP algorithm developed by Kraft [67]. This algorithm is commonly used in various software packages, such as SciPy and NLOpt. Due to experiences of poor performance of this algorithm for some design problems, development of a proprietary SQP algorithm within CADO is also ongoing. Both algorithms are available for use in the inverse design optimisation problem. A comparison of the performance of both algorithms is presented in Chapter 4.

Both algorithms evaluate the objective for a design and then compute the gradients to inform the next design iteration. A line search is then performed along the gradient direction to find a step size so that the objective value is reduced. The implementation of SLSQP in CADO is quite limited. One of the issues is that the algorithm also computes the gradients on every step of the line search. This only slightly speeds up the one-dimensional line search problem whilst significantly increasing the computational effort required. This results in a significant waste of computational effort. The SQP method does not have this issue and is also simpler to configure and use. Given that performance for both methods is similar, the SLSQP is not used for further inverse design optimisations.

If no constraints are used, the program requires a dummy constraint which is always satisfied and does not produce gradient values. For genuine constraints a separate adjoint solution needs to be computed since each adjoint solution only provides the gradient with respect to a single function. The adjoint computations for several functions have already been included in CADO. Since the SQP optimiser is still in development, convergence criteria are not yet implemented. The program will only cease after a specified number of maximum iterations or when manually terminated.

3.4. Validation of the Inverse Method

Several tests validating the performance of the tool were performed. These serve to confirming correct functioning the method and evaluating its use cases. Each of the four tests executed is shortly discussed below.

Verification

To ensure a the proper functioning of the tool, a dedicated test was developed for which the exact solution is known, thus allowing to measure concisely the performance of the tool. This test uses as target distribution the Mach number of a defined profile, for which all design variables are known. The profile is then arbitrarily modified to generate an initial starting point for the inverse design. The success of the inverse methodology is measured by the difference between the solution proposed by the inverse method and the known profile. For a proper functioning inverse method, the known profile should be found back precisely.

Loss Reduction

Further validation of the tool was performed by comparing the performance with direct design methods. The aim here is to check whether an inverse design tool, by specifying a good target Mach number at the design point, is able to find a profile that achieves good performance over a wide operating range. This ability is compared to a direct method, where two options are compared. In a first attempt, only the performance in the design point is considered, which would correspond to a similar computation

cost as the inverse method. A second method considers the performance in various operating points as objective, hence requiring more computational resources than the inverse tool.

Three operating points are considered in the multi-point method. The operational range is defined by changing the incidence angle until the loss coefficient is doubled with respect to the design loss coefficient of the baseline design. The loss coefficients at design conditions ω_o , negative incidence ω_n and positive incidence ω_p are combined using equation 3.16.

$$J = 0.5\omega_o + 0.25\omega_n + 0.25\omega_p \quad (3.16)$$

Proper use of constraints is important in direct shape optimisation. If, for instance, no constraint is imposed on the flow turning, the optimiser may simply reduce blade loading to minimise loss. This does not provide useful information. A flow turning constraint is imposed, so that the outlet flow angle must not be larger than for the original design, ensuring sufficient blade loading. The objective and constraint functions had already been implemented in CADO. During optimisation, the objective and constraint values are normalised with respect to the baseline design.

A suitable baseline to optimise from is needed. The NASA stage 35 stator blade geometry at mid-span was selected. This is a highly loaded blade and is intended to be used in the first stage in a compressor. The blade has been designed for low loss and high stage pressure ratio [68]. This blade needs to be represented in the previously defined parametrisation. This is done using a new tool which calculates the distance between the currently described and desired blade PS and SS curves at a large number of points. Using the same optimisation algorithm as the inverse design tool and complex step gradient computations, this total distance is minimised and the desired blade description is obtained.

A second tool is needed for creating target distributions for inverse design. The designer needs to modify the existing distribution so that increased performance is expected. This modification can be done by specifying a set of points which the distribution should follow. A designer needs to specify a few points, which the tool then interpolates with a B-spline curve. This results in a smooth, continuous distribution which can then be used as target distribution.

Unrealistic Targets

The shaping of target distributions is left completely to the designer. It is therefore possible that the demanded profiles delivering these Mach number distributions do not exist. An unrealistic isentropic Mach number can, for instance, be created by specifying excessively low values along the entire distribution. Since the averaged Mach number in the blade passage cannot be lower than the inlet Mach number, such a distribution is not possible. To evaluate how well the method can deal with these targets, its behaviour when unrealistic distributions are specified is analysed.

Reverse Engineering

A second possible use case of the inverse method is the reverse engineering of a blade geometry based on only a pressure or velocity distribution. It is common practice in scientific publications to not publish the shape of the blades used, since these are valuable intellectual property. However, their corresponding pressure or velocity distributions are published after being non-dimensionalised. If enough other information can be obtained or approximated so that suitable boundary conditions can be formulated, it is possible for the blade shape to be recovered using the inverse design tool.

This possibility was demonstrated by generating a blade design based on the data published in the PhD thesis written by Sans [69]. The data in this thesis is publicly available, but the blade geometry used is not. However, since this thesis was executed at the VKI, the geometry was available for this project and can thus be compared with blade shapes obtained through inverse design. The results of these numerical experiments are presented in the following chapter.

4

Results and Discussion

A crucial step in the method development is ensuring its correct functioning, assessing its performance and identifying the use cases. In this chapter the performance and results produced by the program during each of the previously described tests are documented and discussed. In Section 4.1 the tool development process is reviewed. This includes the gradient and optimisation verification steps.

Following this, several use cases and scenarios were examined. The use of the inverse tool for design improvement is assessed in Section 4.2. Conventionally, loss reduction is achieved through direct design optimisation. For comparison, said optimisations were also carried out. Since the inverse design is based on the target distribution, it is important to create good quality and realistic target distributions. The outcomes of specifying unrealistic targets are examined in Section 4.3. A second use case was also evaluated, namely the reproduction of blade geometries based purely on published data. The findings thereof are discussed in Section 4.4.

4.1. Development of the Method

The development of the inverse design tool itself was relatively straightforward. Most program elements are not unique to the use of an inverse design objective. This meant that most critical and complex programs, namely the flow solver, adjoint solver and optimisation algorithm, did not need to be adapted for the new objective. The geometry and mesh generation functions had to be newly created solely because no functions suitable for 2D axial compressors were available in CADO.

Only the objective evaluation function and its partial derivatives are unique to the inverse design approach. While not easy to include in CADO, these functions do not incur unreasonably high development costs. Especially for developers already familiar with the CADO functions and structure, these elements can be completed within days. The implemented inverse design objective function is not limited to axial compressor blade profiles, but can be used for any two-dimensional blade design problem. It can also be extended to cover three-dimensional problems if a suitable target formulation is selected.

4.1.1. Gradient Verification

For gradient-based optimisation to be successful, it has to be ensured that the computed gradient information is sufficiently accurate. The adjoint solver in CADO has previously been confirmed to produce correct gradient information during the development of other projects. However, because a new objective has been introduced for the inverse process, the verification process needs to be re-initiated. This verification was done by comparing the gradient information computed using the adjoint method with the results of both finite difference and complex step gradient computations. The finite difference computations were repeated for several step sizes.

The gradient verification is performed on an arbitrarily defined compressor profile and an arbitrary velocity target distribution. The profile and target are chosen to be representative for common axial compressor profiles. The baseline and target geometries and corresponding velocity distributions can be seen in figures 4.1 and 4.2 respectively. The selected distribution was of the Mach number type since this covers one more step in gradient computation.

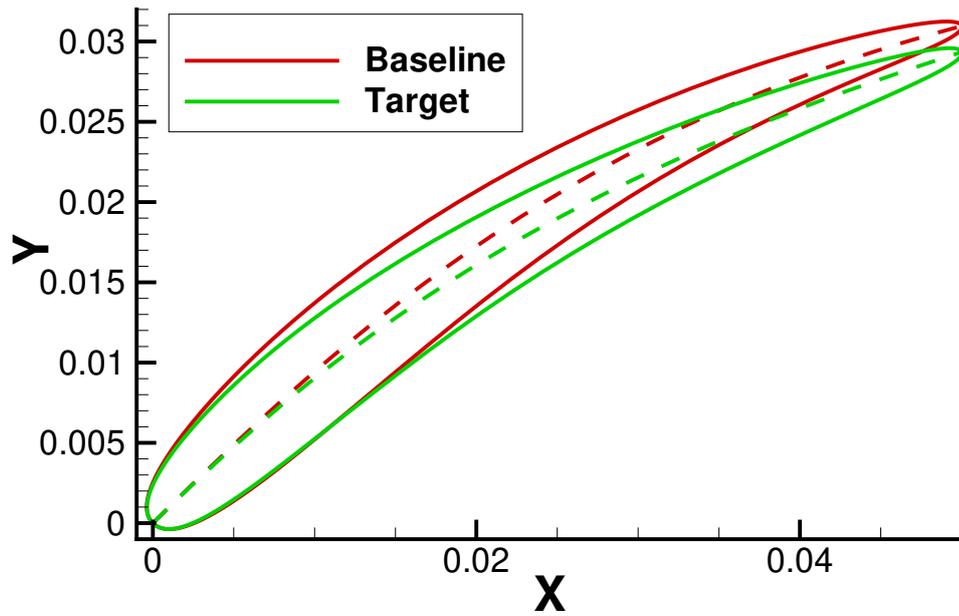


Figure 4.1: Baseline and target geometry used for gradient verification.

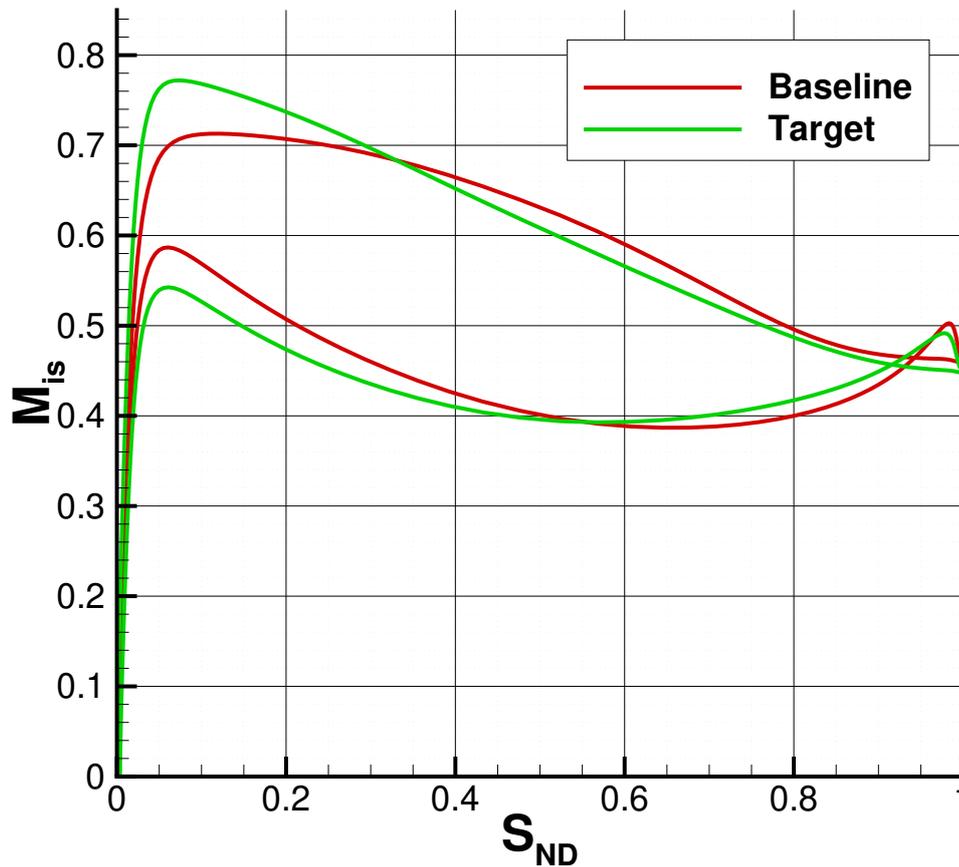


Figure 4.2: Baseline and target isentropic Mach distribution used for gradient verification.

With the starting blade design and target distribution known, the gradients of the inverse design objective with respect to each of the variables was computed. The finite difference computations are repeated with varying step sizes, ranging between 10^{-2} to 10^{-12} . The complex step computations use a step size of $10^{-12}i$. The gradients produced by each of the methods can be seen in figure 4.3.

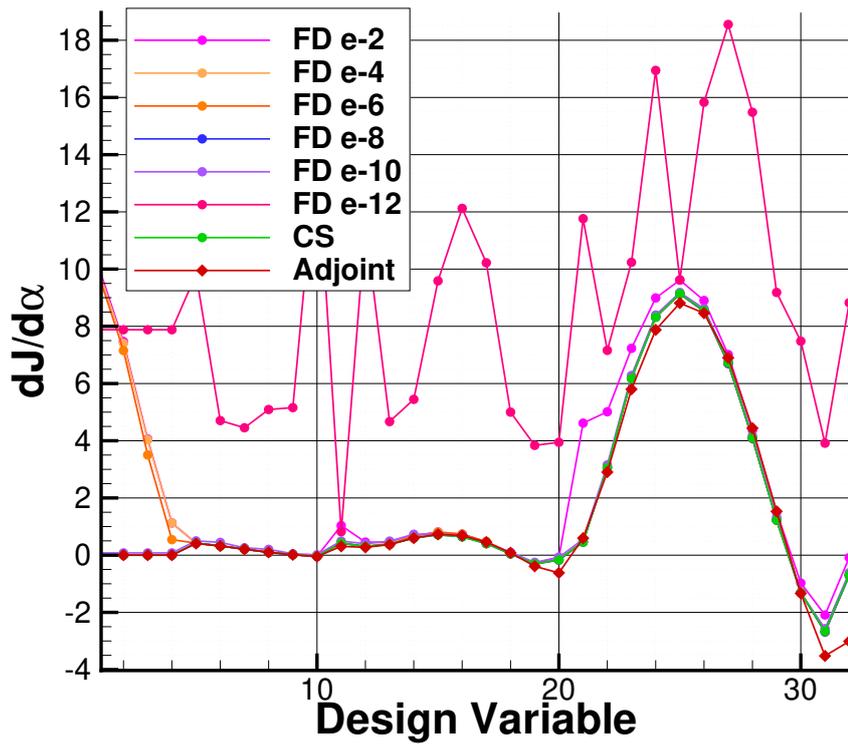


Figure 4.3: Comparison of gradients computed with finite differencing using a range of step sizes, the complex step method and the adjoint method.

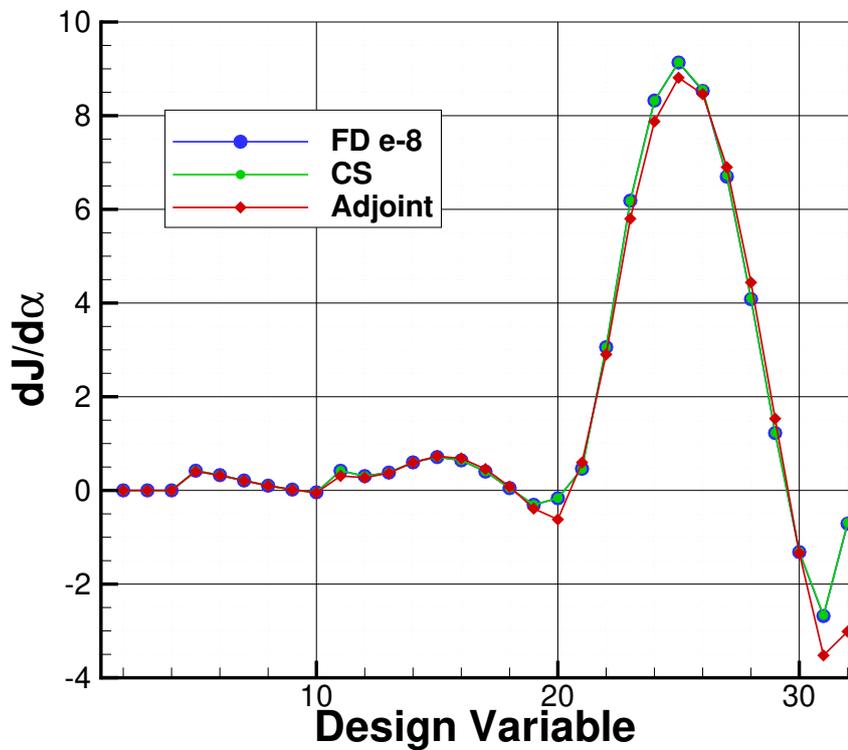


Figure 4.4: Comparison of gradients computed with finite differencing using step size 10^{-8} , the complex step method and the adjoint method.

For clearer comparison, the gradients are again displayed in figure 4.4 with only the most accurate set of finite difference gradients. Due to the strong overlap with the complex step results, the finite difference gradients are marked marked in bold. Note that the initial set of 32 design variables listed in table 3.1 was used during this phase.

There is a clear match between the complex step and finite difference results. The agreement is strongest at a finite difference step size of 10^{-8} . Both methods both rely on only the forward analysis code to compute the gradients. Therefore it was as expected that the two methods are exactly consistent. This indicates that the complex step method can be used as accurate reference case to evaluate the adjoint results against. The finite difference gradients are therefore no longer computed in following evaluations, reducing the computational burden.

The first 4 variables also have near-zero gradient values. Therefore these values can be fixed to reduce complexity. These parameters determine the x-coordinates of the blade angle distribution control points. By manipulating the respective y-coordinates, the camber line can still be modified as required. The reduced set of design variables was listed in table 3.2.

The adjoint result does not fully match the other methods. The discrepancy is larger for variables determining the blade shape near the trailing edge. Due to the drastically different calculations for the adjoint gradients, some differences can be expected. One source of error is that the effect of the change in turbulent quantities on the objective value is not accounted for. The flow field around the blade shown in figure 4.5 is investigated to see what causes the discrepancy.

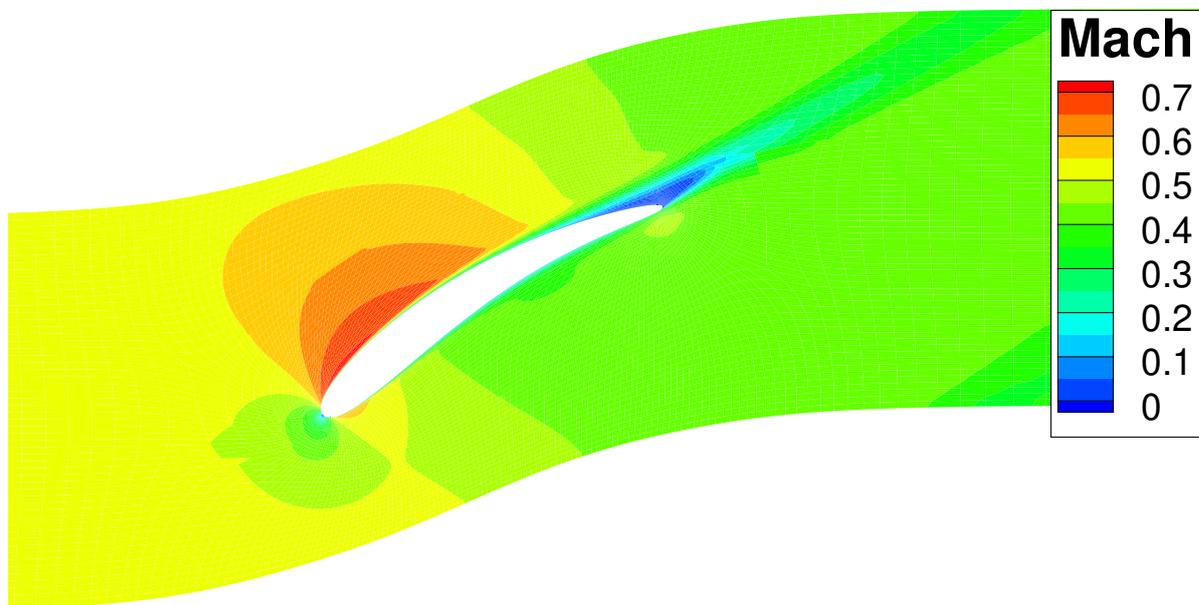


Figure 4.5: Flow field surrounding the baseline blade shape coloured by Mach number.

The flow field indicates that the selected baseline may not be a suitable or representative case. A separation bubble at the TE can be seen. The location of this bubble coincides with the design variables with the largest gradient discrepancies. Since the presence of the separation bubble is strongly impacted by the turbulent quantities, it is possible that the incomplete adjoint gradient computation leads to these differences.

New baseline and target cases are defined where no separation occurs. Figures 4.6 and 4.7 show the new blade designs and their corresponding velocity distributions respectively. The gradients for each of the design variables in this set-up are compared in figure 4.8.

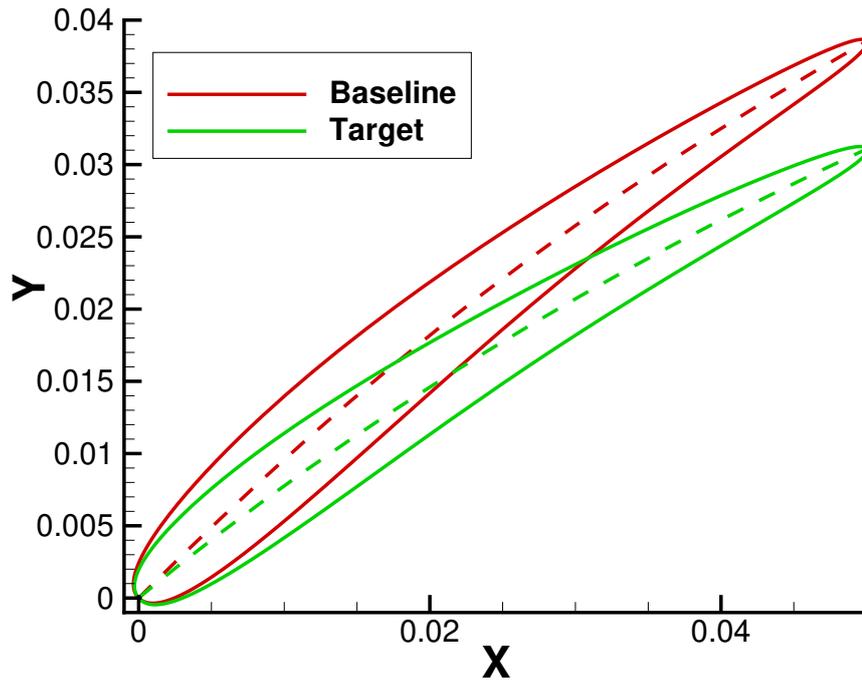


Figure 4.6: Updated baseline and target geometry.

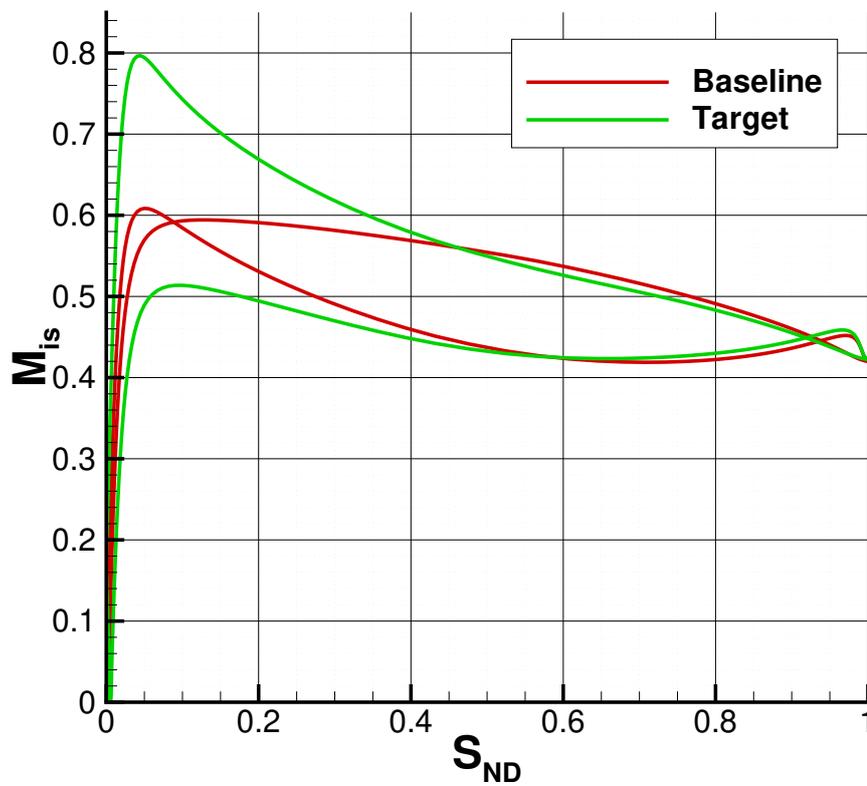


Figure 4.7: Updated baseline and target isentropic Mach distribution.

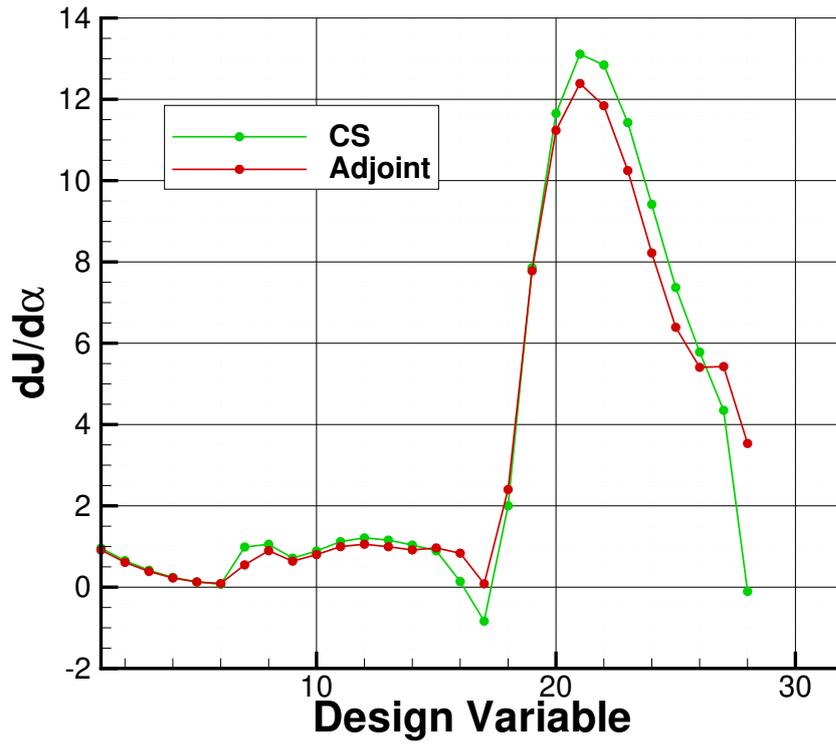


Figure 4.8: Comparison of gradients computed with the complex step and adjoint methods using the updated designs.

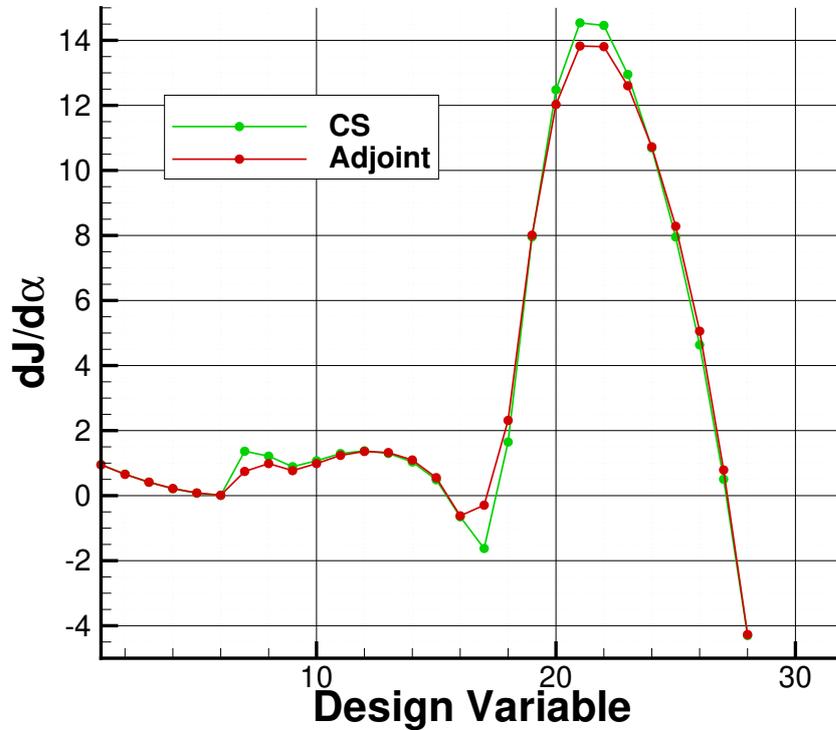


Figure 4.9: Comparison of gradients computed with the complex step and adjoint methods with the CEV model active.

There is still a notable difference between the two methods over the range of design variables. It was previously mentioned that the adjoint code in CADO does not take into account the sensitivity of the turbulence quantities with respect to the grid. To test the hypothesis that these discrepancies are caused by the omission of the turbulence grid sensitivity, the comparison was repeated a third time with the CEV model active. Thus there no values are omitted when solving the adjoint problem. Figure 4.9 displays the gradients obtained when using the CEV model.

The resulting gradients are now for the most part consistent. This confirms the suspected cause for the discrepancies. However, the CEV model should not be used during optimisation, since the flow solution and gradients computed in this way are further removed from reality. The completion of the adjoint solver to fully account for the adjoint turbulence values was outside of the scope of this project. The level of accuracy reached is sufficient for gradient-based methods since the size of the offset should decrease in tandem with the overall gradients as the optimisation converges.

4.1.2. Verification of the Optimisation Framework

The correct functioning of the entire optimisation tool is also verified. The inverse methodology is tested using the profiles in figure 4.6. The green target profile defines the target Mach number distribution, while the red baseline profile is used as an initial guess. The aim is thus to recover the green profile after the optimisation. Successful convergence also confirms sufficient accuracy of the gradients. The tests were performed with both the SQP and SLSQP optimisation algorithms. In figure 4.10 the evolution of the cost function for both optimisers is shown as a function of the number of evaluations. In the definition of the evaluation number, due to the fact that one adjoint simulation costs approximately the same as one flow simulation, 2 units are considered each time a gradient evaluation is performed. Failed evaluations are indicated with a objective values J of 10^2 .

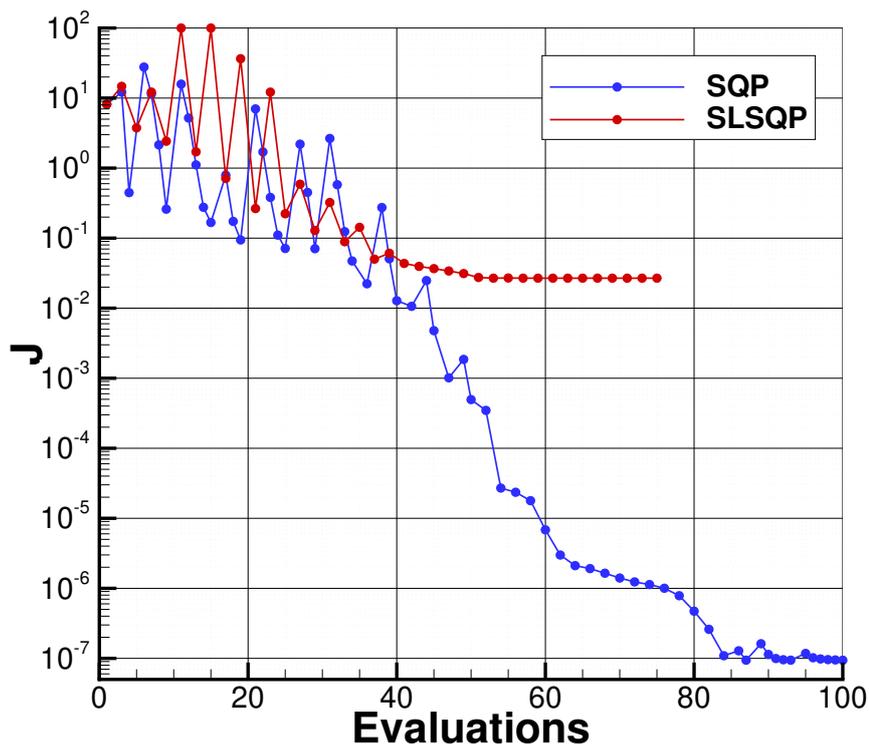


Figure 4.10: Convergence history of the program verification using both SQP and SLSQP.

The SQP algorithm converged at a faster rate. The zig-zag pattern in their optimisation histories indicate that both methods may benefit from reducing the initial step-size taken when starting a line search step. This is possible for the SQP method and was slightly fine-tuned during subsequent optimisations. However this is not possible when using SLSQP as it is implemented in CADO. As the design converges and the design space is mapped better, this effect reduces and fewer line search steps are needed.

The SLSQP algorithm failed to reach the optimum, unlike the SQP algorithm which recovered the target blade shape. The designs resulting from both methods are displayed in figure 4.11. The baseline and target designs are also indicated. The corresponding isentropic Mach number distributions can be seen in figure 4.12. The design results are those after 39 evaluations for the SLSQP method and 40 evaluations for the SQP method, since the former did not yield significant improvements after this point. This way, the comparison can be done at similar computational cost.

The SLSQP method got stuck at the design space boundary as the blade thickness near the trailing edge was increased. The optimiser may have gotten on this wrong path due to the large initial step sizes. The SQP method was able to reach significantly lower residuals, eventually nearly exactly matching the target blade shape and distribution after 87 evaluations. The converged results can be seen in figures 4.13 and 4.14. This confirms the correct function of the optimisation framework. It also indicates the sufficient accuracy of the gradient computations. The SQP algorithm was used for all subsequent optimisations since it performs better and is easier to use in CADO. Furthermore, experience in the use of SQP provided useful feedback for the further development of the program.

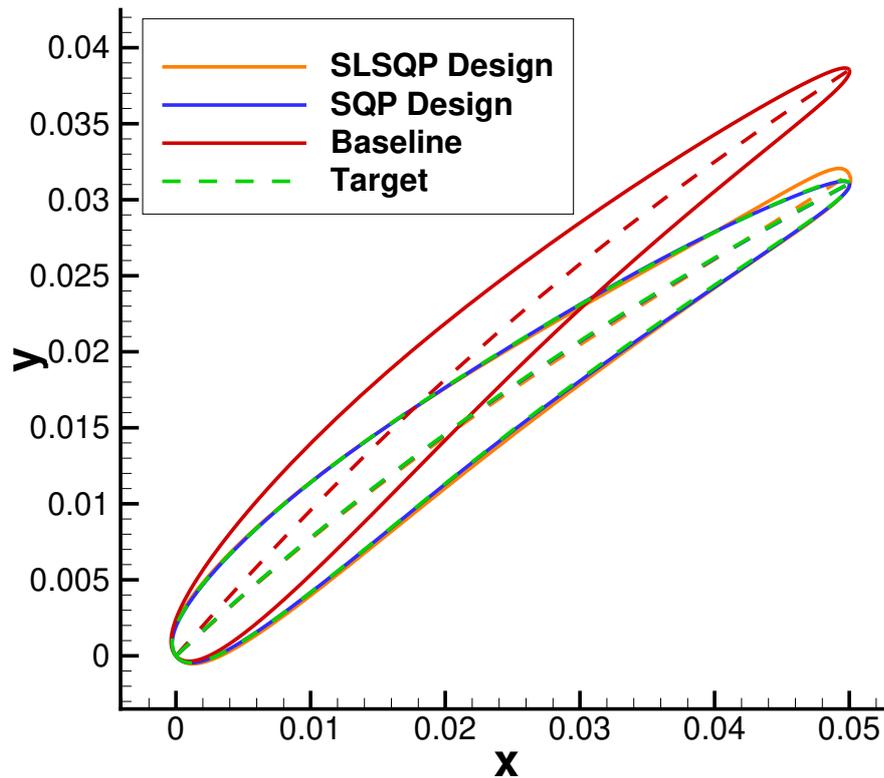


Figure 4.11: Comparison of the resulting design geometries after 40 evaluations.

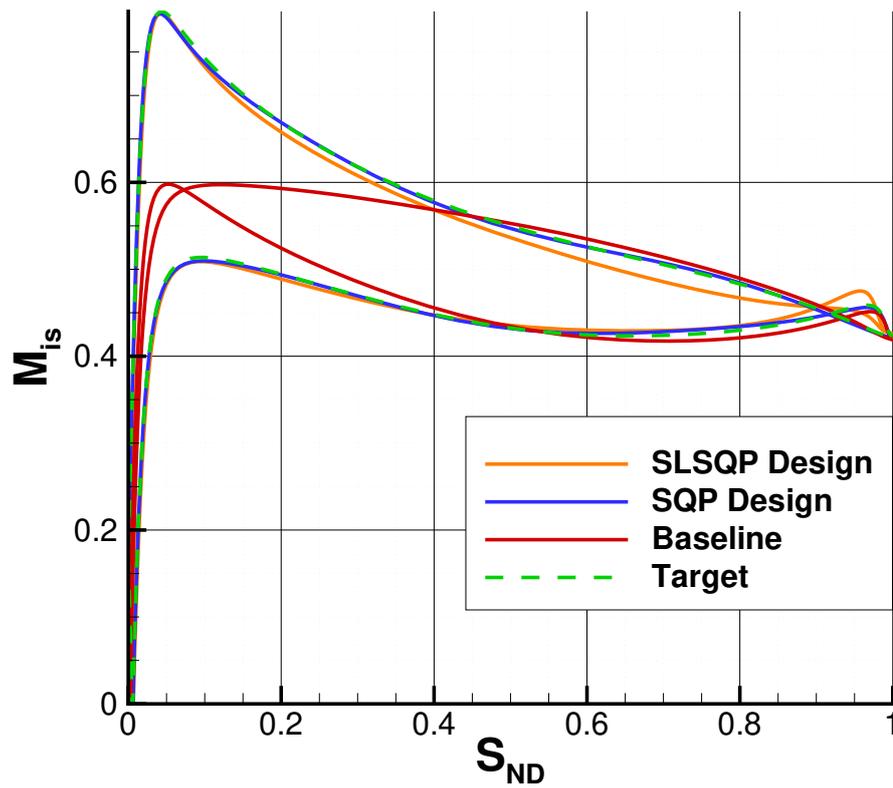


Figure 4.12: Comparison of the resulting distributions after 40 evaluations.

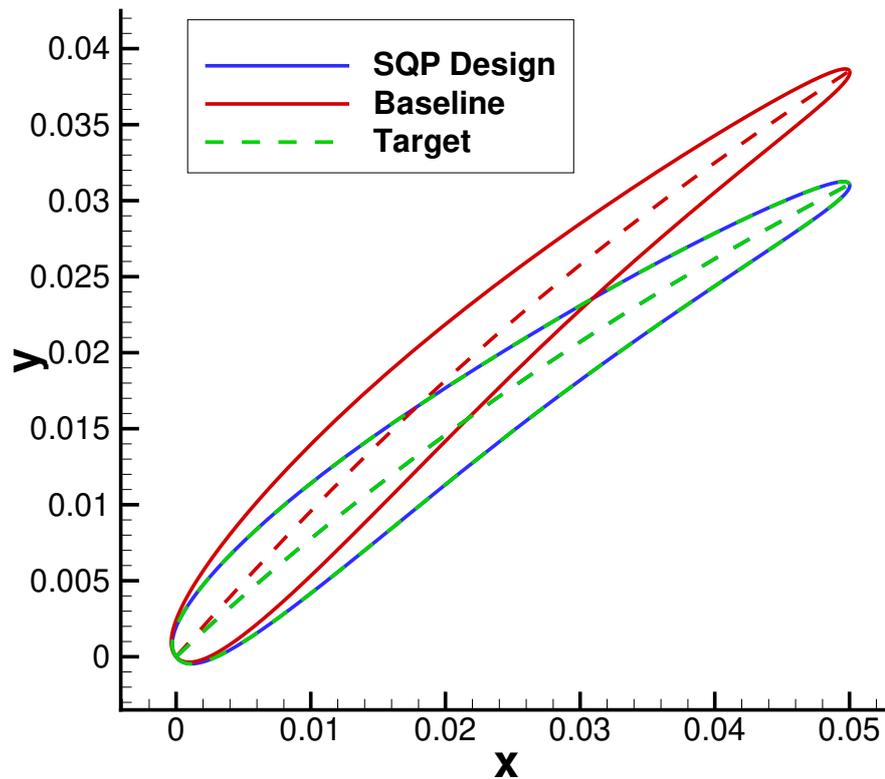


Figure 4.13: Optimised design geometry obtained with the SQP algorithm.

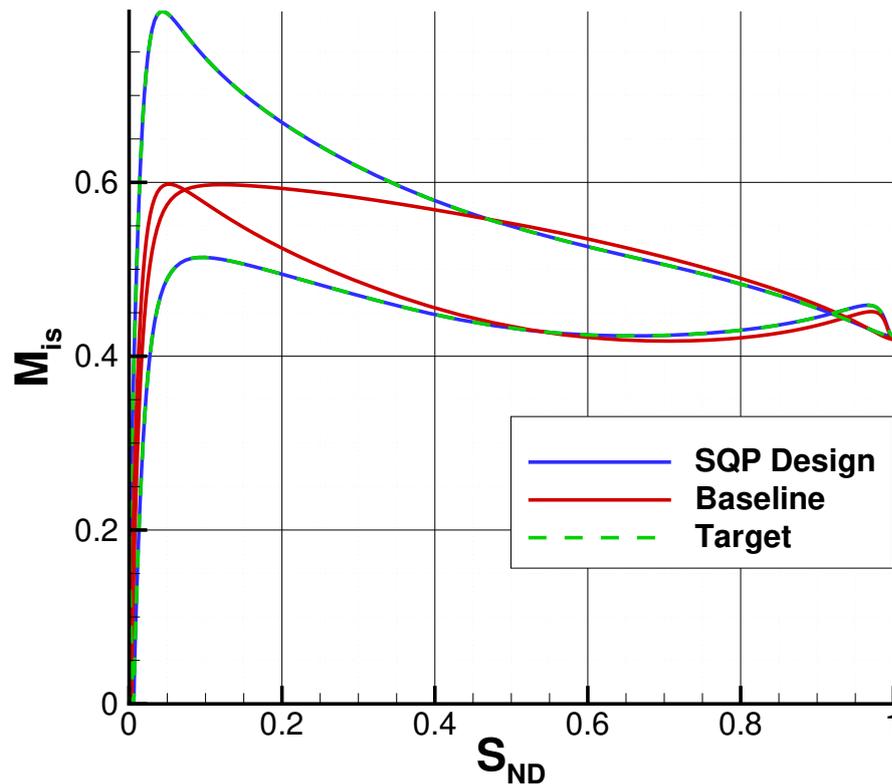


Figure 4.14: Optimised isentropic Mach distribution obtained with the SQP algorithm.

4.2. Loss Reductions of Existing Designs

The inverse method was used in attempts to improve the performance of an existing compressor blade design. The selected baseline design was the NASA stage 35 stator blade at mid-span. The inlet and outlet conditions of the blade row have been identified from the blade performance evaluation [70]. This is a carefully designed blade and already has a good performance. However, further performance gain in the current set-up is possible. Attempts at improving the design were made both with inverse design and direct optimisation strategies.

4.2.1. Loss Reduction Through Inverse Design

A target distribution needs to be specified for the inverse design strategy. An initial flow solution yields the velocity distribution for the baseline geometry in figure 4.15. This isentropic Mach number distribution at design conditions was modified by selecting a series of points on both the suction and pressure side curves. By translating and subsequently interpolating these points, a new target distribution is generated. The modifications are made based purely on designer judgement. However, the formulation of a good target distribution proved difficult. As such, the process is quite iterative.

To reduce the losses, in some attempts the entire shape of the distribution was modified. Other trials involve simple offsets, aiming at maintaining the initial amount of flow turning. An overview of the different targets used can be seen in figures 4.16 and 4.17. The subdivisions for the second target indicate small modifications to reach the correct amount of flow turning. Only for the first target an outlet flow angle constraint was additionally imposed. This was not done for the remaining targets to reduce computational cost.

In table 4.1, the performance of each of the resulting designs is listed. The loss coefficients at design conditions ω_0 are listed together with the outlet flow angles α_{out} . Additionally, the table also introduces the loss coefficients at off-design conditions ω_n and ω_p . These will be discussed in Section 4.2.2.

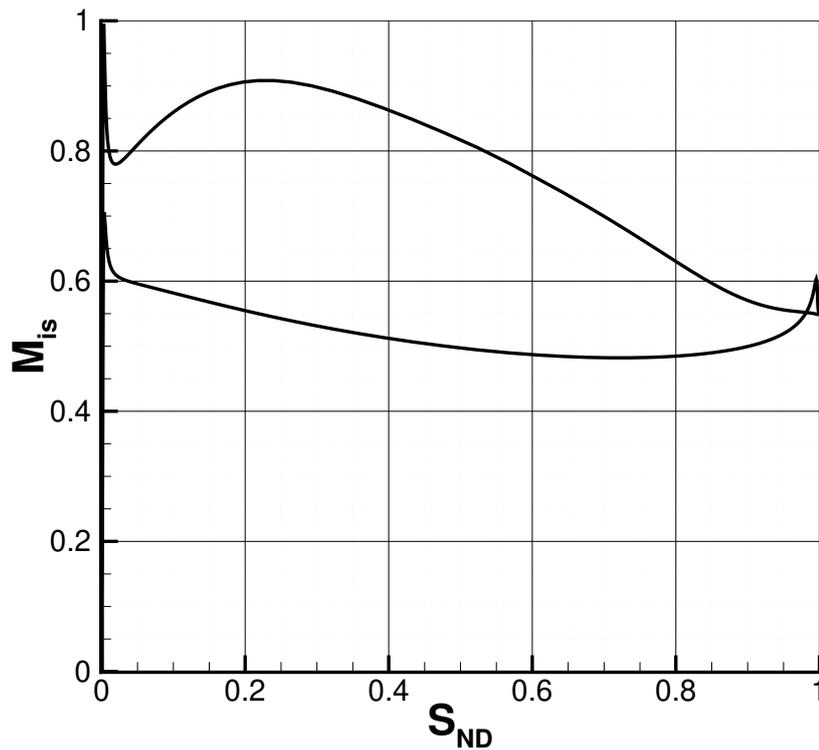


Figure 4.15: Isentropic Mach distribution corresponding to the NASA stage 35 blade.

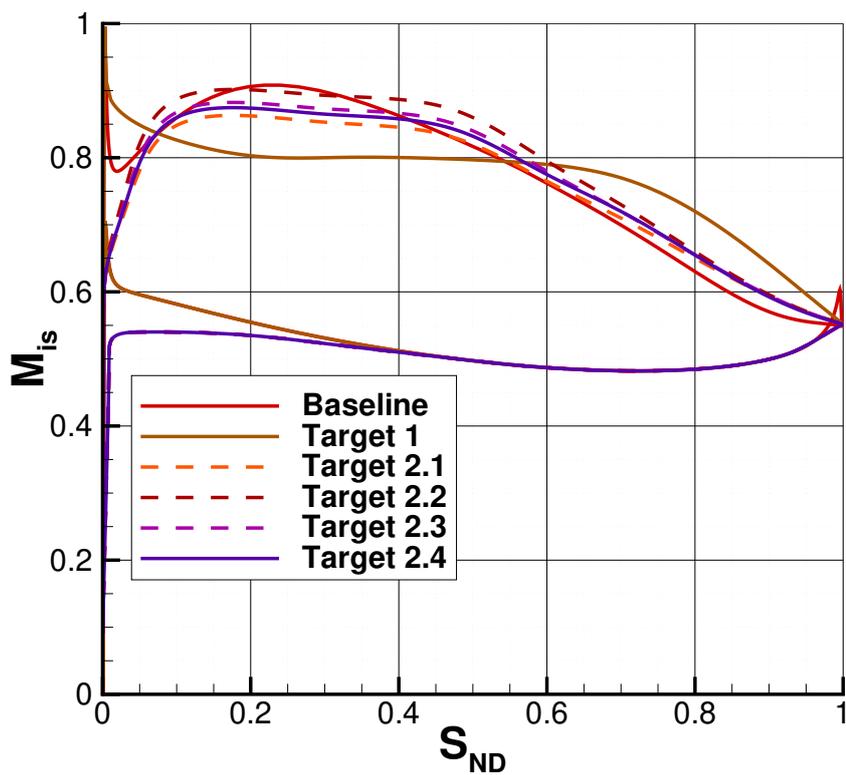


Figure 4.16: Overview of the first two inverse design target distributions used.

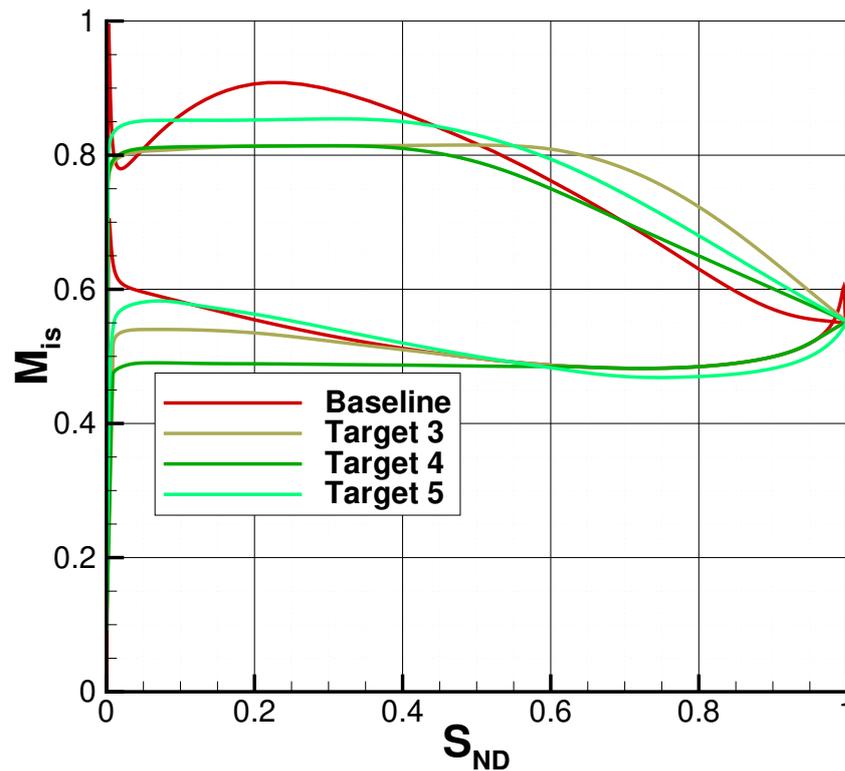


Figure 4.17: Overview of last three the inverse design target distributions used.

Table 4.1: Performance of the designs obtained with each of the inverse target distributions.

Design	ω_o [-]	ω_n [-]	ω_p [-]	α_{out} [°]
NASA stage 35	0.02408	0.04770	0.04769	13.447
Target 1	0.02730	0.05859	0.05558	13.661
Target 2.1	0.02263	0.04701	0.04329	15.088
Target 2.2	0.03137	0.04969	0.05268	12.116
Target 2.3	0.02504	0.05047	0.04856	12.813
Target 2.4	0.02450	0.05068	0.04689	13.561
Target 3	0.02907	0.06044	0.05456	12.691
Target 4	0.01988	0.05032	0.05055	14.115
Target 5	0.02483	0.02839	0.05235	15.533
Target 5 CS	0.02573	0.04268	0.05200	12.202

New target distributions are created based on lessons learned from previous attempts. Resulting from these experiences some rules of thumb were formulated for the further modification of the distributions:

- keep the area between the the SS and PS curve constant,
- restrict deceleration zone on the suction side to the rear of the blade,
- limit the slope of the velocity profile on the suction side.

The area between both curves should not be changed as this conserves blade loading and thus also the flow turning. Fair comparison in blade performance based on loss can only be made if the designs maintain the same amount of flow turning. Since a large part of the loss is associated with deceleration on the suction-side, the deceleration zone is kept to a minimum. However the zone should not be too small since strong deceleration will lead to separation and additional loss.

It is clear that the design of a target distribution benefits strongly from designer experience. Several iterations were needed before a gain in performance was reached. The first target simply reduces the deceleration on the suction-side by flattening the curve. However, strong deceleration at the trailing edge results in flow separation and additional loss.

In the second target, the peak at the leading edge is removed, so that the resulting design would be at the optimal incidence condition. To maintain flow turning, the pressure side curve was slightly lowered and the peak Mach number on the suction side increased. This value is modified in several steps to reach the desired flow turning. Nevertheless, the resulting design still did not notably improve performance. Firstly, nearly the entire suction-side experiences deceleration, resulting in a thick boundary layer. Secondly, the acceleration near the leading edge now is too low, resulting in a very sharp leading edge. This results in poor off-design performance.

As a result, in the third target a flatter profile was specified. This profile is similar to the first but with the velocity peak near the leading edge removed. However, just like this first design, strong deceleration near the TE results in flow separation. Furthermore, excessive flow turning was reached, again resulting in an increased loss.

Only after the fourth target a notable gain in performance was reached, but only for design conditions. This was done by reducing the slope of the velocity near the SS trailing edge. The pressure side velocity was lowered further to reach a nearly flat profile on this side as well in an attempt to maintain flow turning. The resulting design does not fully match the prescribed target, as seen in figure 4.19. The discrepancy is stronger towards both the LE and TE. The resulting design, shown in figure 4.18, has very sharp edges. This results in increased design condition performance, but worse performance in the off-design conditions. Such a design is not feasible due to structural concerns.

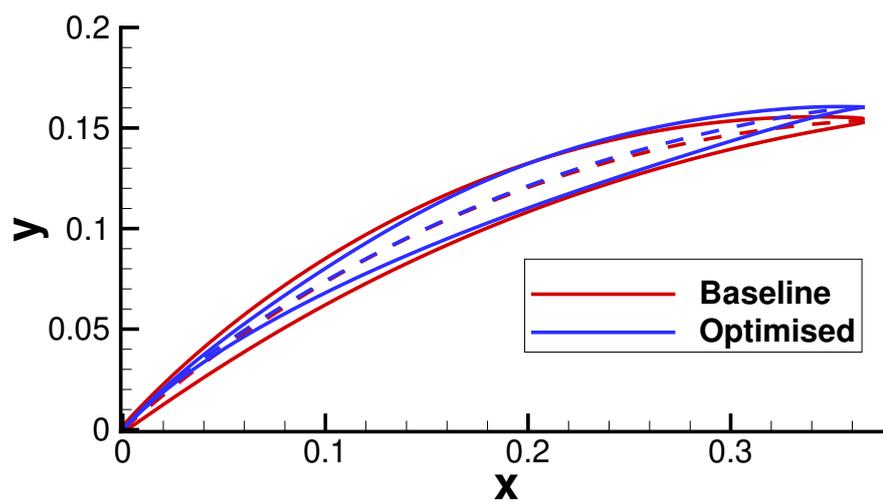


Figure 4.18: Blade geometry resulting from inverse design with target 4.

A final, fifth target was created by increasing the suction side velocity to reach additional turning. The pressure side curve was then shaped so that for any location along the blade, the average of the SS and PS velocity is similar to that of the baseline design. The idea was that this would result in a blade with reasonable thickness distribution along the entire chord and similar flow turning.

However, just like before, the inverse design method did not properly converge to the desired target. In order to investigate whether this is due to the limited gradient accuracy, the same inverse design was repeated with complex step gradient computations. The blade designs obtained with both optimisations using target 5 are compared in figure 4.20. Their corresponding isentropic Mach number distributions are shown together with the target distribution in figure 4.21. The results of the optimisation with complex step gradient computations are included in table 4.1 and separately indicated. The optimisation histories of the inverse design optimisations with target 4 and both with target 5 are presented in figure 4.22.

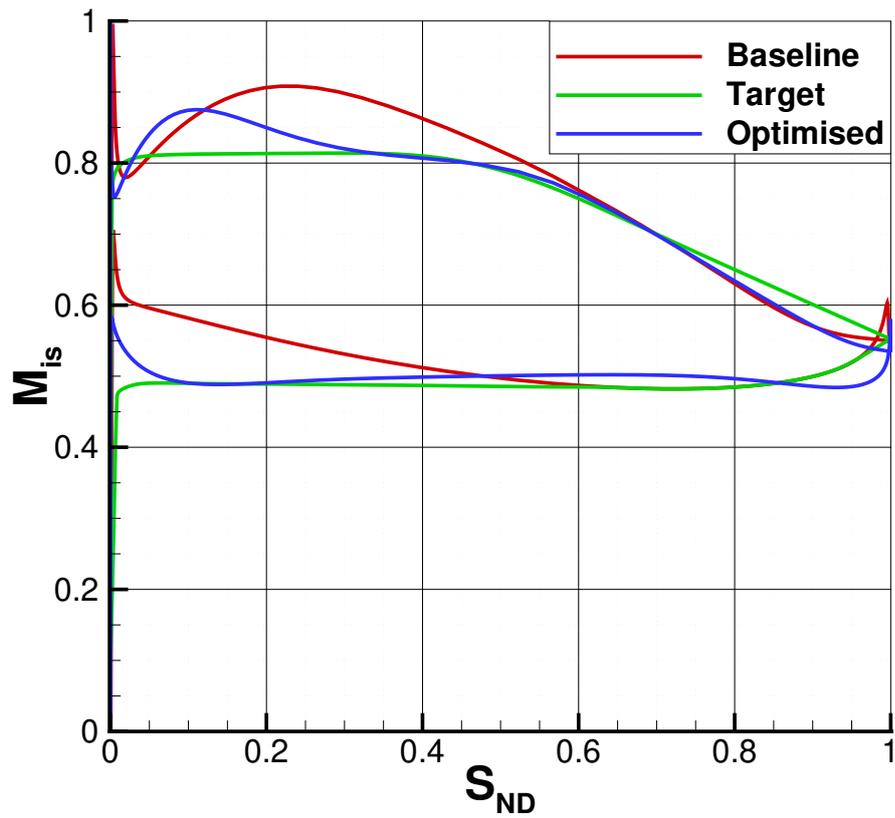


Figure 4.19: Isentropic Mach distribution resulting from inverse design with target 4.

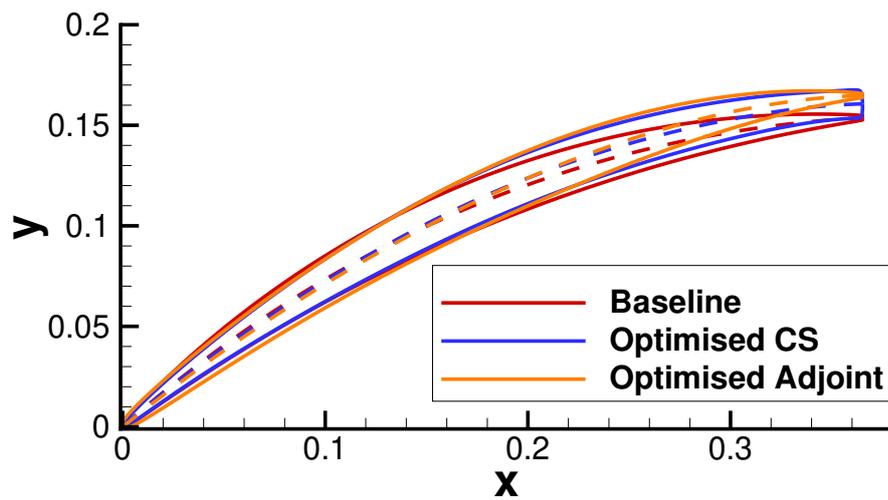


Figure 4.20: Blade geometry resulting from inverse design with target 5 with both adjoint- and CS-based gradient computations.

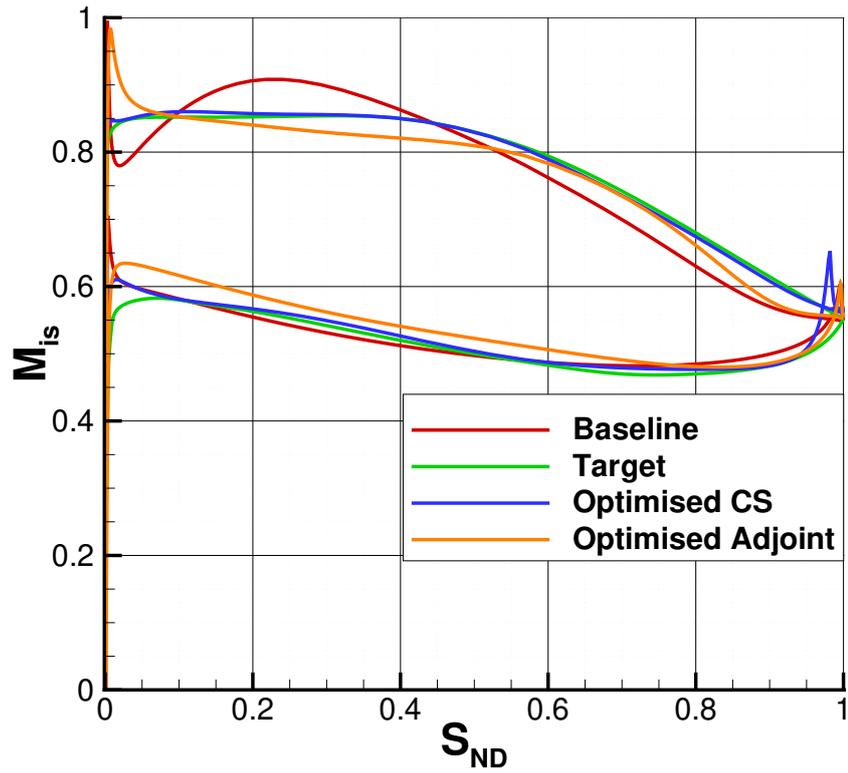


Figure 4.21: Isentropic Mach distribution resulting from inverse design with target 5 with both adjoint- and CS-based gradient computations.

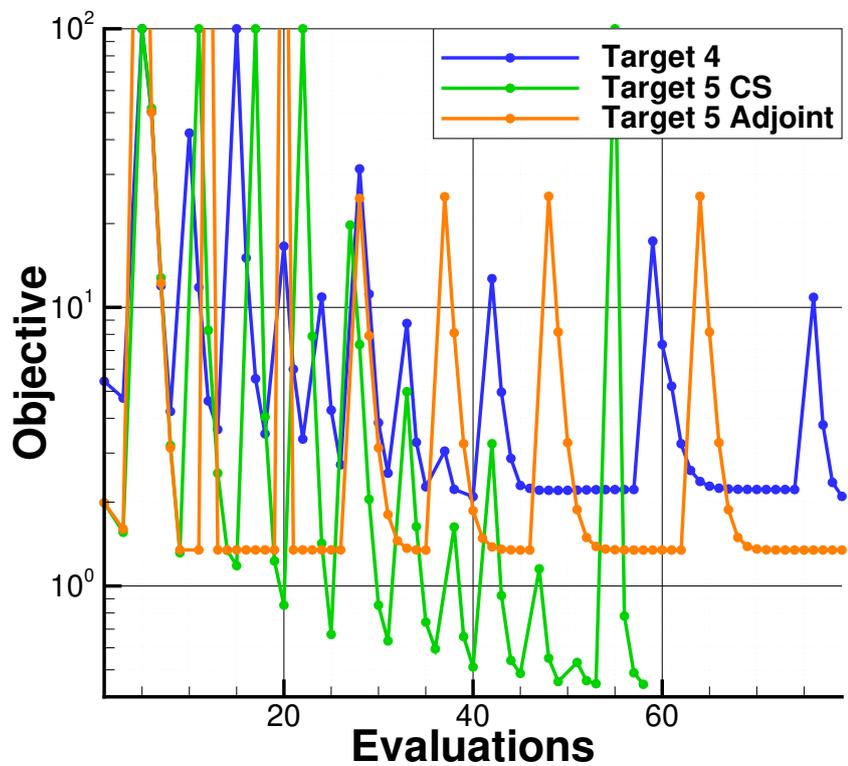


Figure 4.22: Convergence history of the target 4 and both target 5 inverse design optimisations.

The design obtained with adjoint-based gradient calculation has a strong velocity peak at the leading edge. This indicates that the design may perform better under different incidence angles. This is supported by the fact that the negative incidence losses for this blade are very low. The optimum inlet angle got shifted to lower inlet angles. The entirety of the SS also experiences deceleration, leading to a continuous build-up of the boundary layer and an increased loss.

It can be seen that the design obtained with CS closely matches the target, except at the TE. The blade has a blunt trailing edge. This results in a thick trailing edge wake and increased mixing losses. The blade design therefore also has poorer performance. This is similar to the result of the SLSQP verification step where the optimised blade also had a thick trailing edge. This issue is likely linked to the shape of the target distribution, but the exact cause is not clear.

The method showed good accuracy on most of the obtained results. The distribution of the optimised blade generally quite closely matches the specified target distribution. Nevertheless the number of iterations until convergence is increased by the limited accuracy of the gradients. For a more effective tool it would still be advantageous to use a complete gradient computation.

In some optimisations the tool did not fully converge. The optimisation with target 5 using complex step was able to reach a design significantly closer to the prescribed target distribution. It also converged notably faster than the adjoint-based optimisations. For this comparison, the cost of gradient evaluation by both methods is assumed to be equal to a single evaluation. This shows that reduced gradient accuracy leads to poorer optimiser performance and can in some cases prevent from convergence being reached. A similar phenomenon was indicated by Giles [15] who noted the same issues due to inaccuracy of continuous adjoint gradients. For some other cases the solver was not able to exactly match the specified target distribution. This is attributed to limitations to the shape freedom imposed by the chosen parametrisation method.

These observations provide an answer to the first research question. CADO can be used to accurately compute gradients using finite difference, complex step or discrete adjoint methods. Of these methods, the discrete adjoint method is the cheapest method for any problem with more design variables than objective and constraint functions. However, the adjoint solver is not complete and may not produce accurate gradients for functions sensitive to the local turbulent behaviour, such as those measured at the blade wall. In those cases the complex step method is the best alternative, as it provides accurate gradient computations without suffering from numerical instability. Thus there is no search for the optimal step size, as is necessary for finite difference computations. The inclusion of the turbulence grid sensitivity term in the adjoint solution is being worked on.

4.2.2. Direct Design Optimisation

It is more conventional to improve performance through direct optimisation. The inverse methodology was therefore compared to both single- and multi-point optimisations. The loss coefficient was used as performance metric to minimise. For the multi-point optimisation, the performance value is the weighted sum of the design loss coefficient and the coefficients at the two limiting off-design conditions at varying incidence angles i . The loss bucket for the NASA stage 35 blade is visualised in figure 4.23. The operational range and baseline performance values for the multi-point optimisation is also indicated. The design incidence angle is -1.6° with respect to the blade metal angle [68]. The off-design conditions are at incidence angles -7.95° and 6.35° .

Neither the multi-point optimisation nor the single-objective optimisations improved the objective values or satisfied the outlet flow angle constraint. The optimisations were repeated with entropy generation as objective, since it has been used for previous projects. However, still the optimisations did not succeed. Therefore the gradients with respect to a couple of design variables of the entropy generation and outlet flow angle objectives are compared with gradients computed using the complex step. The resulting sets of gradients can be seen in figure 4.24.

Strong differences can be seen between the adjoint and complex step gradients. Again, the CEV model was then used to see if the inaccuracy is caused by the simplification in the adjoint gradient computation. The resulting sets of gradients can be seen in 4.25. It can be seen that the accuracy of the adjoint gradient computation is significantly improved when the CEV model is used.

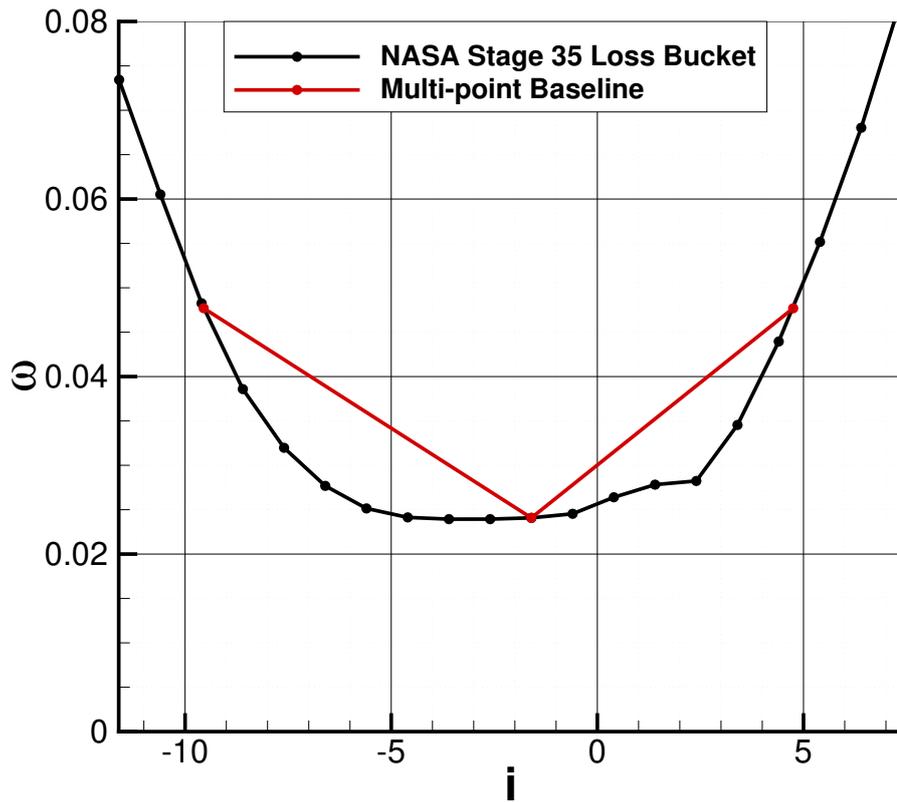


Figure 4.23: Loss bucket for the NASA stage 35 design and the baseline reference points for multi-point optimisation.

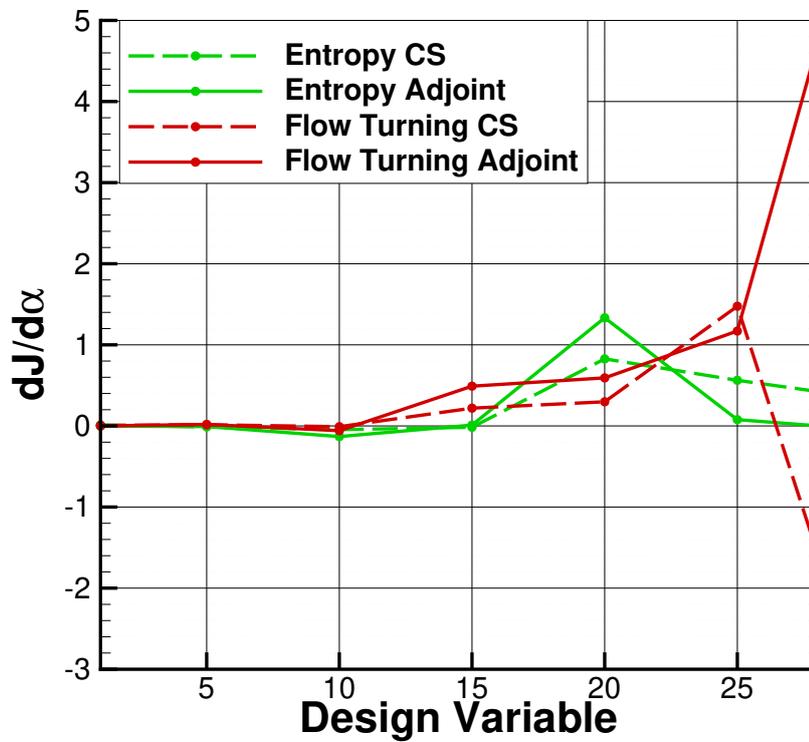


Figure 4.24: Comparison of objective and constraint function gradients computed with the complex step and adjoint methods.

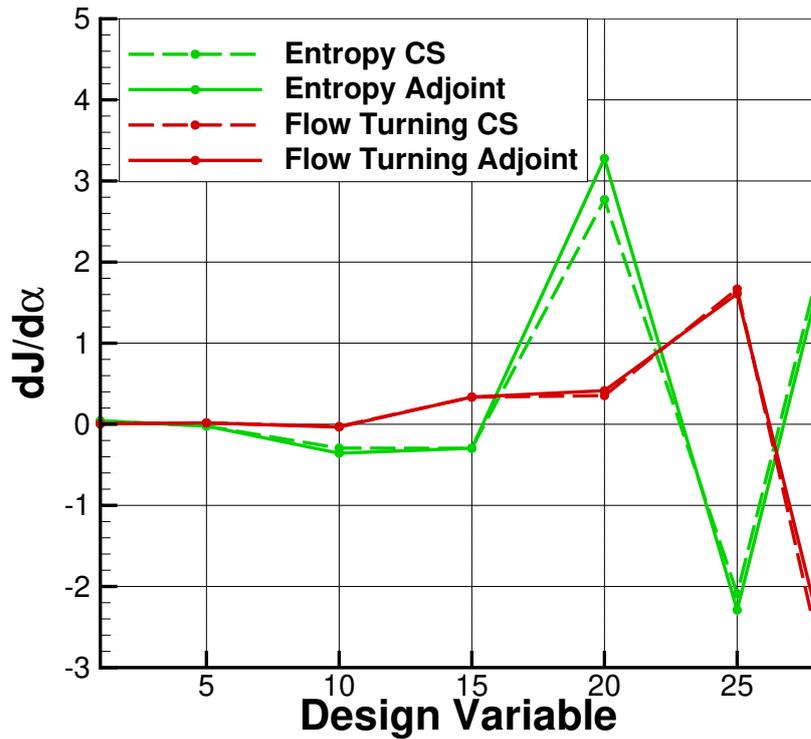


Figure 4.25: Comparison of objective and constraint function gradients computed with the complex step and adjoint methods with the CEV model active.

This indicates that the inaccuracies can once again be attributed to omission of the turbulence grid sensitivity in the adjoint gradient computation. However, in this case the gradients are not sufficiently accurate for optimisation. This was also an issue for previous design problems using these objectives. However, those were all for turbine blades and thus operate under favourable pressure gradients. Turbulence does play a role in turbine blade design, but some impactful phenomena, such as flow separation, are generally not expected.

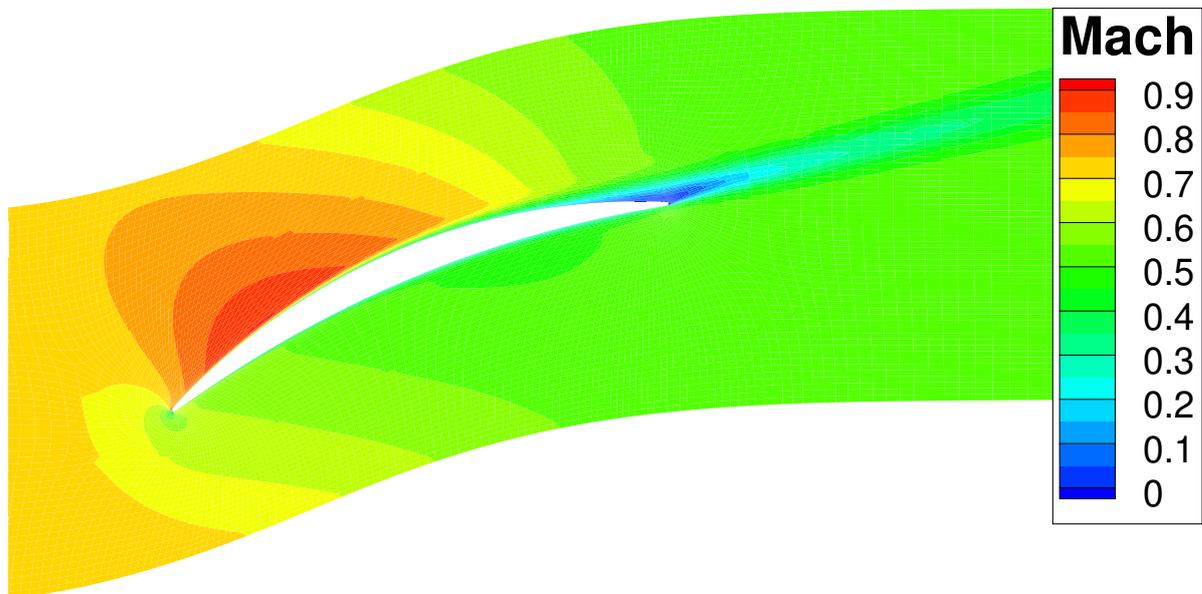


Figure 4.26: Flow field surrounding the NASA stage 35 blade shape coloured by Mach number.

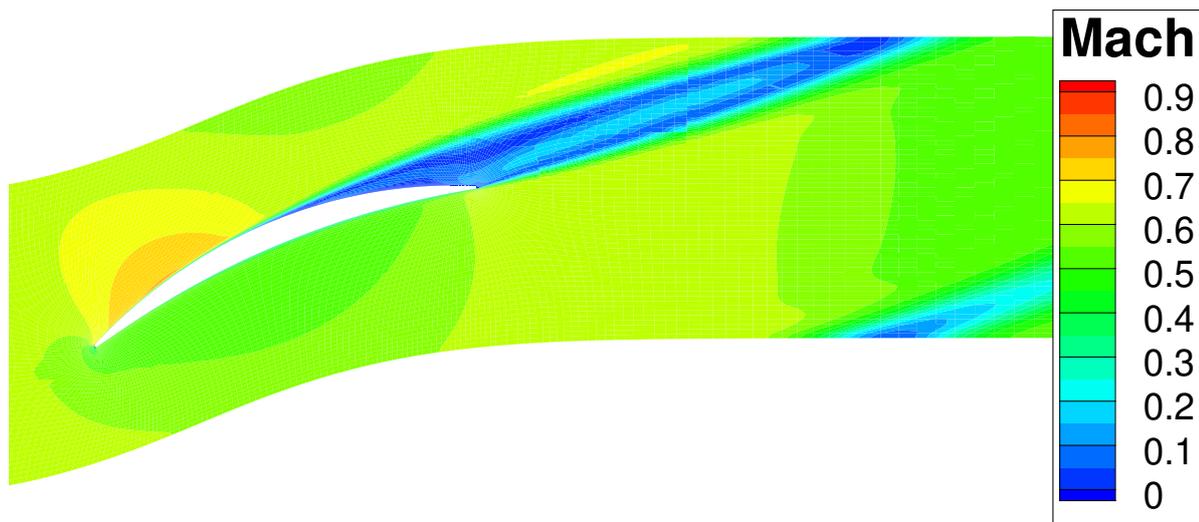


Figure 4.27: Flow field surrounding the NASA stage 35 blade shape coloured by Mach number calculated with the CEV model active.

The cause of the high sensitivity to the turbulent quantities becomes clear when comparing the flow field around the blade when analysed with the SA and CEV models, shown in figures 4.26 and 4.27 respectively. When using the former, the flow remains attached to the blade until very close to the trailing edge. Only a small wake is created and modest losses are to be expected. With the CEV model, the flow separates from the suction side halfway the blade, resulting in a separation bubble of several chord lengths. This leads to significantly increased losses. Furthermore, the flow does not follow the blade curvature, and little flow turning is achieved.

It can be seen that the selected turbulence model has a big impact on the final flow field and the performance of the blade. It can therefore also be accepted that the sensitivity of both the outlet entropy and flow angle functions with respect to the turbulent quantities is not insignificant. The adjoint calculation in CADO is incomplete and lacks a calculation of the turbulence grid sensitivities. This means that the contribution of the turbulence quantities to the objective grid sensitivity is not included. The resulting objective gradients are no longer sufficiently accurate for optimisation assignments.

The derivation of the turbulence model needed to complete the adjoint gradient calculation is outside the scope of this project. Instead, the direct optimisations are executed using complex step gradient computations. Such computations incur significantly increased computational costs, but for comparison with other results the number of evaluations is still computed as if the adjoint method was used. Thus the computational cost if an adjoint method were to be used is assessed whilst obtaining accurate gradient information.

The loss coefficients at the three operating points considered for the resulting designs of each of these direct optimisations are collected in table 4.2 at the end of this section. For comparison, the loss coefficients of the designs obtained through inverse design are also displayed. The coefficients obtained with the original design are also included.

The following optimisations did produce designs with significantly reduced losses. As could be expected, the single-objective optimisation did this by reducing the blade thickness as allowed by the domain bounds, as seen in figure 4.28. The resulting blade has a uniform thickness of 1% of the axial chord length. Such a design works well at the design condition, but experiences flow separation in off-design conditions. Interestingly, the loss coefficient at the negative incidence condition actually decreased. Under these conditions, separation occurs on the pressure side at the leading edge of the blade. The resulting separation bubble creates a more rounded geometry which the incoming flow can follow, leading to lower loss than expected.

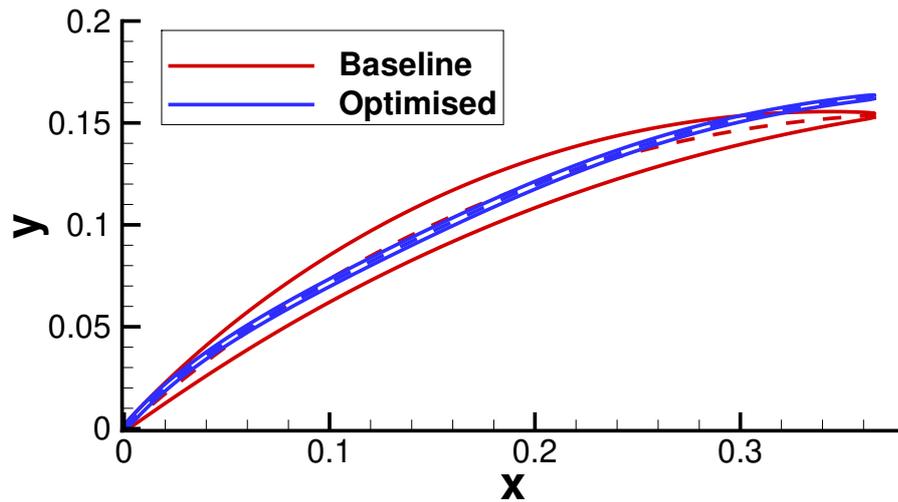


Figure 4.28: Blade design resulting from the single-objective optimisation.

The multi-point optimisation also was able to improve the design. This resulting blade is displayed in figure 4.29. It also has reduced thickness towards the trailing edge of the blade. Both methods converge towards thin trailing edges since this results in smaller wake and shedding losses. The thickness near the leading edge was slightly increased in order to improve the off-design performance. Significant gain was made on the negative incidence loss in exchange for a slight increase in positive incidence loss.

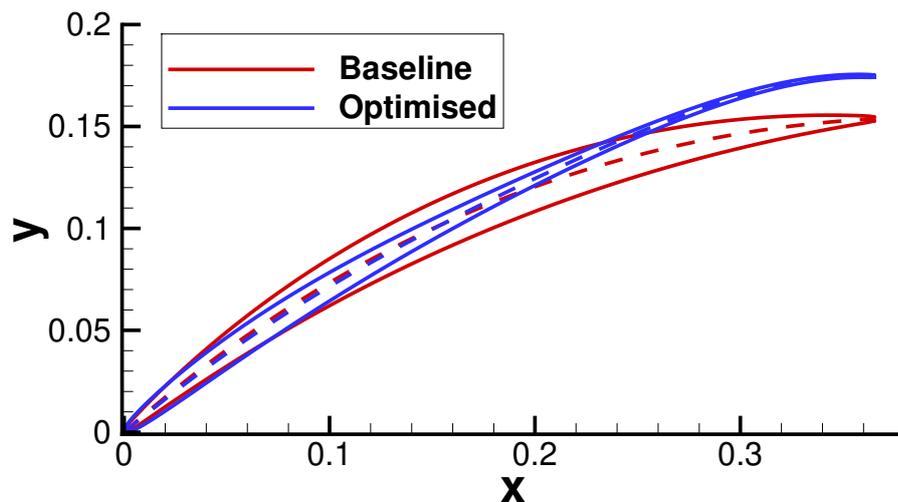


Figure 4.29: Blade design resulting from the first multi-point optimisation.

Both optimisations resulted in significant improvements in aerodynamic performance, but neither blades are feasible designs as they would not be able to cope with significant structural loads. Several methods exist to remedy this issue. The most thorough method is to perform MDO by including a structural analysis component in the design evaluation and constraining the design with a set maximum allowable stress. However, this greatly increases the complexity of the program, the computational cost and is ill-suited to two-dimensional design problems. A simpler alternative is to include a geometric constraint, such as imposing a minimum cross-sectional area. Nonetheless, this method does not guarantee a structurally sound blade design and again increases complexity.

Instead of adding more constraints, a new, restricted multi-point optimisation was performed. The number of design variables was reduced by fixing most of the coordinates of the thickness distribution control points. The thickness distribution described by the design variables of the baseline and multi-point optimised design are compared in figure 4.30. Only the coordinates of the first 3 control points

remained variable as for those points the local thickness was increased during the first multi-point optimisation. At the remaining points the thickness was decreased, potentially leading to structurally infeasible designs. In the restricted optimisation only 11 design variables remained.

The second multi-point optimisation results in a feasible design and improves the performance of the entire blade. The thickness distribution is mostly fixed, so the resulting design is very similar to the original design, as can be seen in figure 4.31. The optimised blade is more strongly curved near the LE. As with the previous optimisation, the design and negative incidence loss is reduced, but the positive incidence loss is slightly increased. This can be understood by investigating the blade design. The increased curvature means that the blade is a smaller obstacle for more axial flows, whilst increasing the blockage for more tangential flows. From these optimisations it seems that it is significantly easier to reduce losses at lower incidence angles than higher incidence angles.

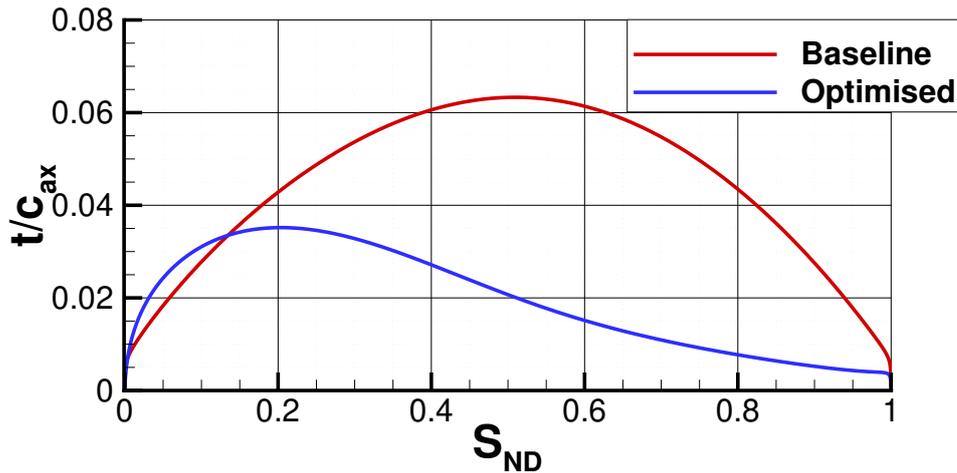


Figure 4.30: Thickness distributions of the baseline and multi-point optimised designs.

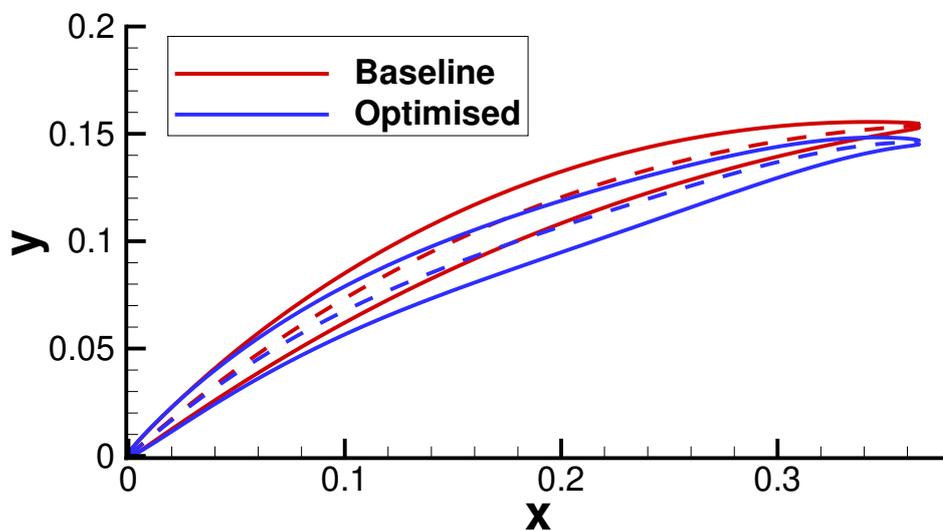


Figure 4.31: Blade design resulting from the restricted, second multi-point optimisation..

The computational cost of both direct optimisation methods also are compared in figure 4.32. As before, the number of evaluations is simply the sum of flow evaluations and gradient evaluations. The significantly increased cost of gradient computations due to the use of complex step gradient computation was not taken into account. The comparison thus represents the hypothetical case where all of the objectives can use accurate adjoint-based gradient calculations. As before, objective values of 10^2 indicate failed evaluations.

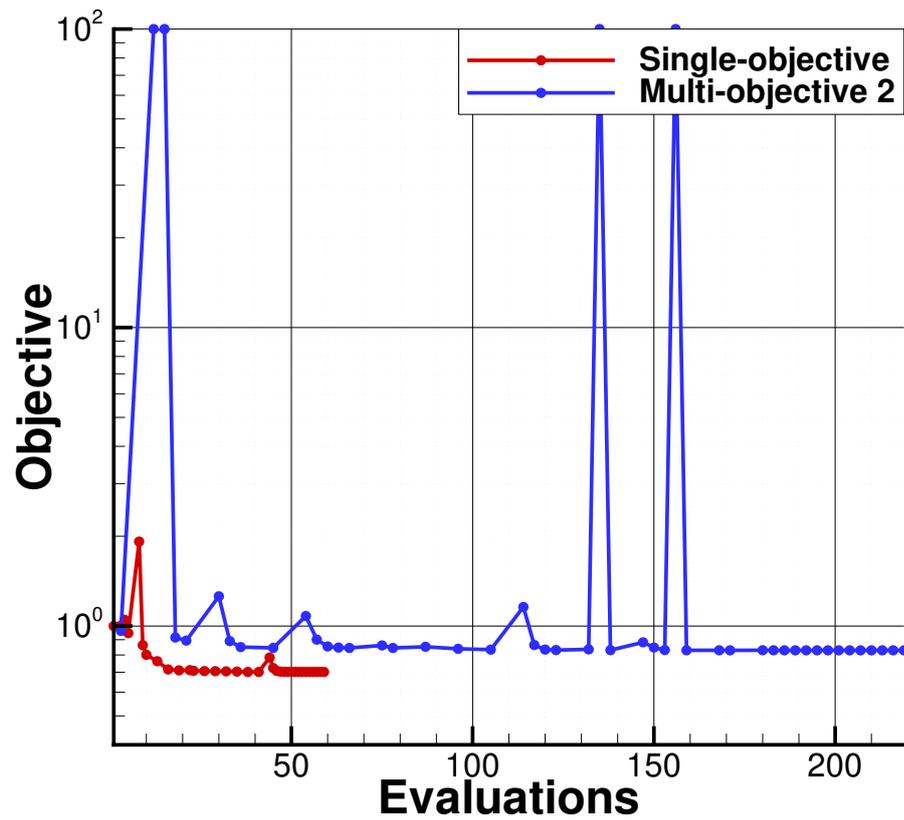


Figure 4.32: Convergence history of the single- and second multi-point optimisations.

Table 4.2: Loss coefficients of the designs obtained through direct optimisation.

Design	ω_o [-]	ω_n [-]	ω_p [-]
NASA stage 35	0.02408	0.04770	0.04769
Single-objective	0.01645	0.04272	0.06585
Multi-point 1	0.01826	0.02022	0.05625
Multi-point 2	0.02085	0.02239	0.05343
Target 1	0.02730	0.05859	0.05558
Target 2.1	0.02263	0.04701	0.04329
Target 2.2	0.03137	0.04969	0.05268
Target 2.3	0.02504	0.05047	0.04856
Target 2.4	0.02450	0.05068	0.04689
Target 3	0.02907	0.06044	0.05456
Target 4	0.01988	0.05032	0.05055
Target 5	0.02483	0.02839	0.05235
Target 5 CS	0.02573	0.04268	0.05200

To visualise the data in table 4.2, the performance over the entire operational range of some of the obtained designs is shown in figure 4.33.

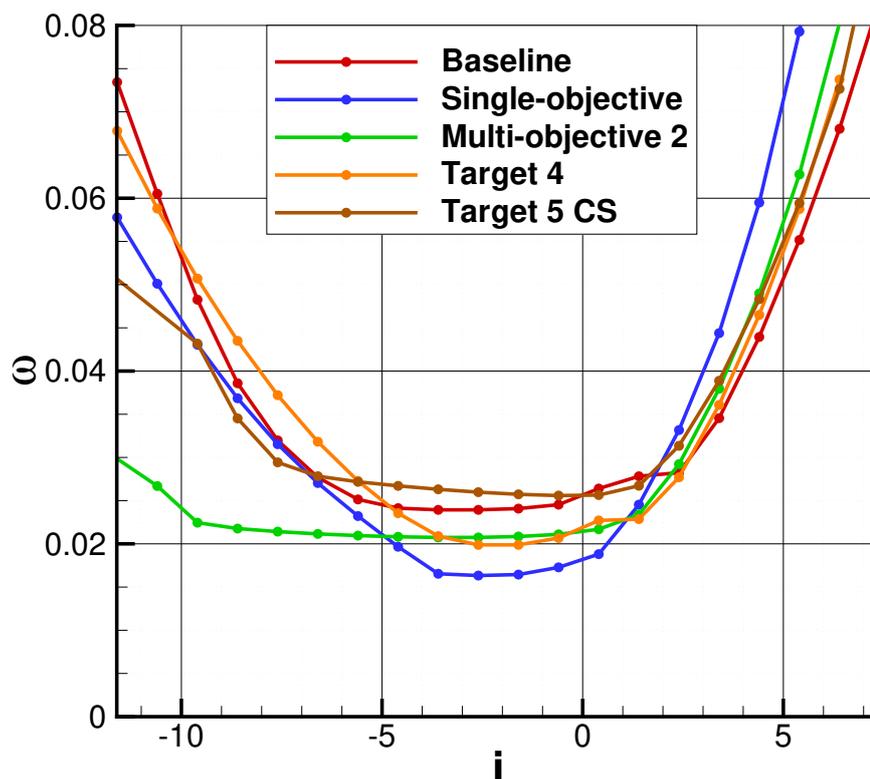


Figure 4.33: Loss buckets corresponding to several designs obtained through inverse and direct methods.

4.2.3. Assessment of the Inverse and Direct Methods

None of the designs obtained through the inverse method can be considered as an improvement to the original NASA 35 blade. Several factors complicated the creation of improved target distributions and the acquisition of the corresponding blade designs. Most important is the inaccuracy in the gradient information, preventing proper convergence to the target for both the fourth and fifth targets. For the other targets, convergence was slowed down due to this inaccuracy. If the gradient accuracy was increased to the level of complex step gradient computation, significantly faster and stronger convergence is achieved.

The performance is entirely dependent on the quality of the target provided. Good understanding of various tweaks to the target distribution is essential. As a result, the method becomes more useful for more experienced designers. The five different distributions evaluated were created by an inexperienced designer and thus did not succeed in improving the design. Furthermore, it is not trivial to accurately reach a certain amount of flow turning, adding to the iterative nature of the inverse design process. A flow turning constraint can be used, but this increases the otherwise low computational cost of the method.

Direct optimisation methods did produce designs with improved design and off-design performance, but come with a different set of problems. The objective and constraint functions used proved to be more sensitive to the inaccuracies in the adjoint-based gradient computation. The complex step method had to be used, resulting in significantly increased computational cost. The resulting designs were not unexpected, but did not provide any useful information without adding additional constraints or significantly reducing the design space.

A direct optimisation method needs at least two constraints for it to be useful, such as outlet flow angle and blade cross-sectional area. Without such constraints, the optimiser will tend to unload the blade and minimise the blade thickness to reduce wake losses. This adds to the program complexity and the computational cost of the solution with respect to the inverse design method, as seen from figures 4.22 and 4.32. Furthermore, single-objective optimisations will improve the design performance, but may narrow the loss bucket as observed in figure 4.33. If this is unacceptable, multi-point optimisation is required. This comes at the expense of further increased computational cost.

From these results it is clear that for gradient-based methods in general it is crucial to have accurate gradient computations. Otherwise the optimiser may fail to reach the objective or converge slowly. Once such issues are resolved, direct optimisation remains the preferred method for the improvement of blade designs. Despite the lower computational cost, the inverse method is not suitable for this task. This is because of the difficulty of further improving an already well-performing blade design based on the isentropic Mach number or pressure distributions.

4.3. Unrealistic Targets

It can easily happen for an inexperienced designer to accidentally specify an infeasible target distribution. The tool will still function, but is unable to reach the desired result. Ideally, the tool would reach the closest feasible result. However, the resulting behaviour has not been confirmed. It is characterised so that a designer can recognise when an unrealistic target is specified.

Unrealistic target distributions were deliberately tested. This was done by creating a velocity distribution with the averaged Mach number between SS and PS below the inlet Mach number. This is unrealistic since the presence of the blade restricts the flow, increasing its velocity. To reach velocities lower than the inlet velocity near the leading edge, the blade would have to have negative thickness. The inverse design was repeated without bounds on the thickness distribution to see if such negative volumes would be obtained.

From the result of the bounded design, seen in 4.34, it can be seen that the solver behaves as expected. The velocity distribution, displayed in figure 4.35, follows the target distribution as close as possible, but remains quite far removed from the target. The thickness of the blade is minimised to lower the Mach number. Some flow turning is done as this reduces the tangential velocity component, thus further reducing the Mach number. An interesting feature is the reversed curvature near the trailing edge. This is linked to the specified strong deceleration on the suction side.

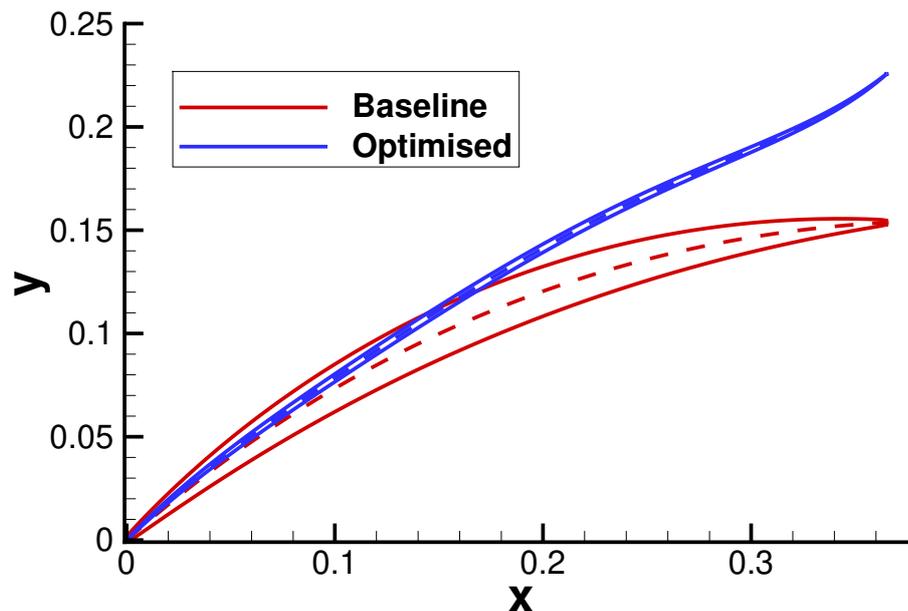


Figure 4.34: Blade geometry resulting from bounded inverse design with an unrealistic target distribution.

The unbounded design is shown in figure 4.36 and the corresponding distribution in 4.37. It does not represent a physically possible solution. The suction and pressure side curves cross over. This means that, as expected, the blade volume has become negative in order to further reduce the velocity. The solver is not able to find a shape following the specified target distribution.

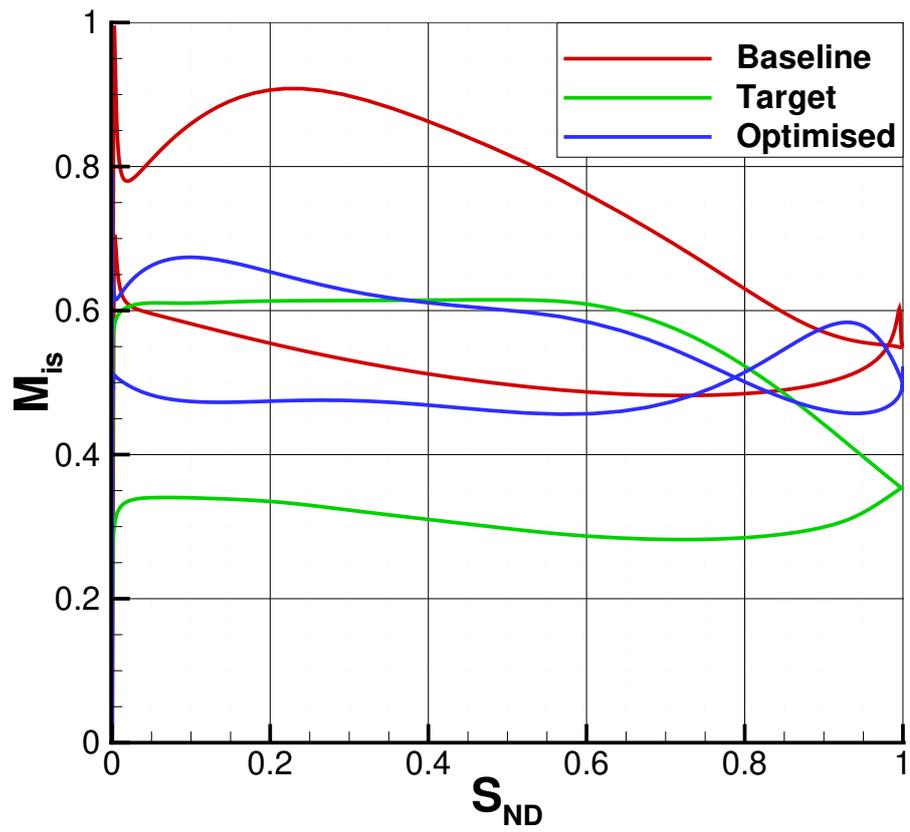


Figure 4.35: Isentropic Mach distribution resulting from bounded inverse design with an unrealistic target distribution.

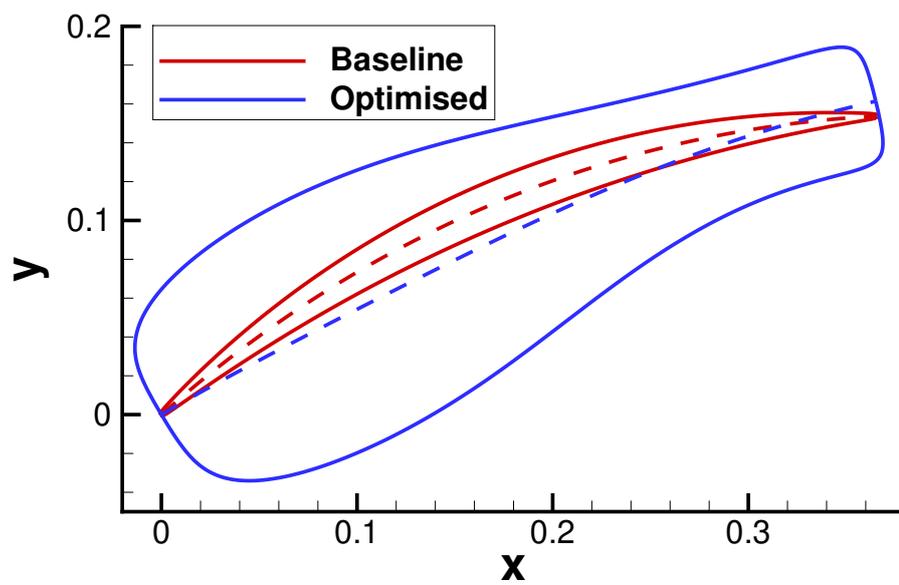


Figure 4.36: Blade geometry resulting from unbounded inverse design with an unrealistic target distribution.

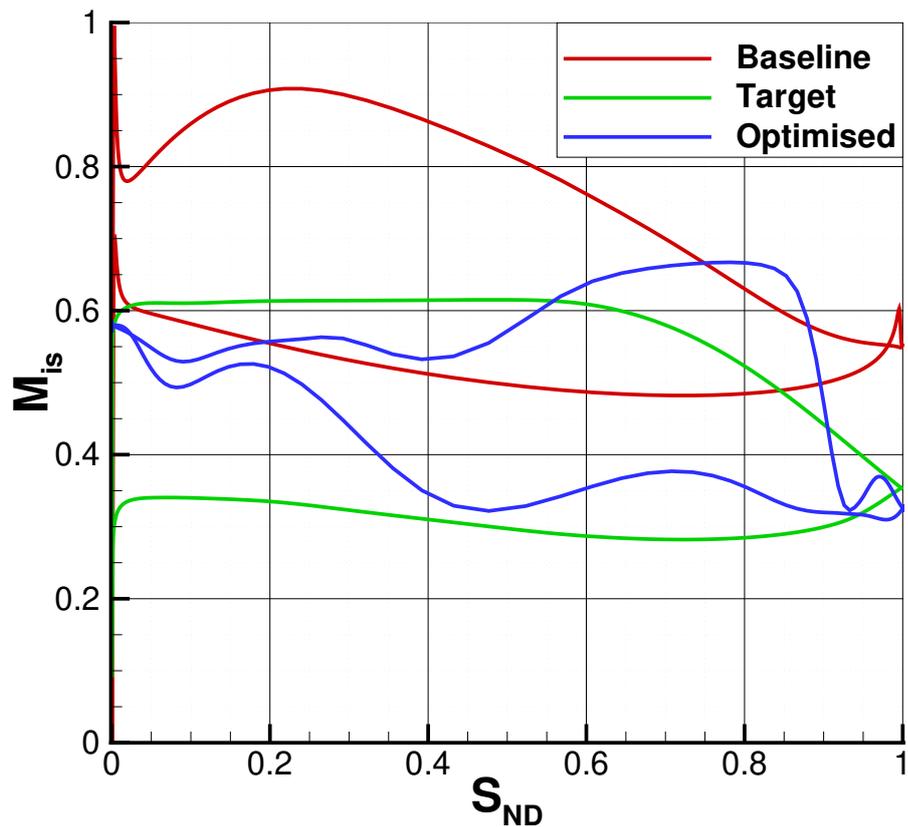


Figure 4.37: Isentropic Mach distribution resulting from unbounded inverse design with an unrealistic target distribution.

Another, less extreme, example of an unrealistic target distribution was previously inadvertently specified in target 3. This target is used in an attempt to improve the performance of the NASA stage 35 blade. The target and distribution corresponding to the resulting design can be seen in figure 4.38. The deceleration on the suction side is too strong and cannot be followed by the flow. The optimiser attempts to find the closest feasible velocity distribution, in this case with a separation zone near the trailing edge.

When creating target distributions, it is important to consider the physical meaning of the specified distributions. Even if the distribution is not at all possible, the optimiser will still function. It is important to set up proper bounds so that such unrealistic results are prevented. A new rule for target distribution creation is needed. For the target to be physical, the average Mach number between the PS and SS near the leading edge of the blade needs to be higher than the inlet Mach number. Still, it is a beneficial property that the inverse method will converge to the closest realistic result when an unrealistic target distribution is specified. Typically, the differences would be smaller than for these extreme cases. The differences can then provide useful insight for the improvement of the target distribution, such as was the case for inverse design target 3.

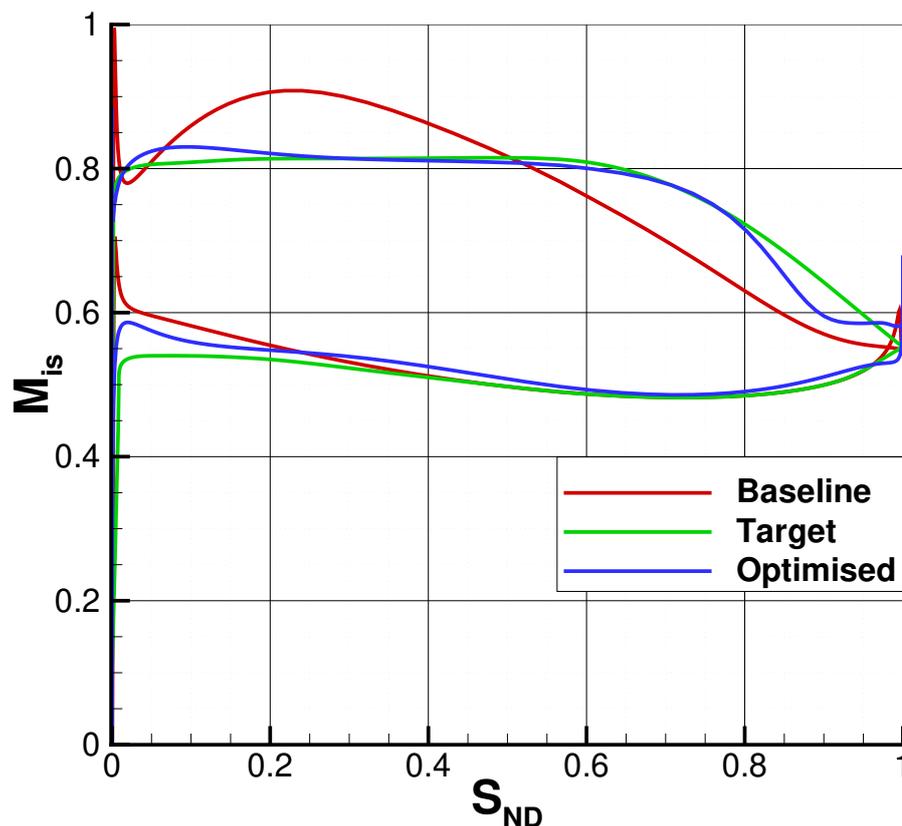


Figure 4.38: Isentropic Mach distribution resulting from inverse design with target 3.

4.4. Reverse Engineering

It was found that the inverse design method is not a suitable technique for improving existing blade designs. An alternative use case is the reproduction of a blade geometry based on published velocity or pressure distributions. This option is demonstrated using data in the PhD thesis of Sans [69].

This work contains the isentropic Mach number distribution obtained both through experiment and numerical simulation. The chord length, solidity, inlet Mach number and flow angle are also disclosed. From the experimental set-up, it can be deduced that the inlet total temperature and outlet static pressure are 293 K and 101325 Pa respectively. The required inlet total pressure was then varied in simulations so that the known inlet Mach number of 0.6 is reached. The blade geometry itself is not published, but is available from the VKI directly.

The experimental velocity distribution is only described on the suction-side. Furthermore, both distributions range between chord-wise positions 20% and 80% of the chord length. Both of the sets are used to construct completed target distributions for the inverse design tool, seen in figure 4.39. The original, limited data sets are indicated in bold to distinguish them from the constructed target distributions. The pressure side curve is based on the CFD data for both target distributions. The distributions based on experimental data are labelled with 'Exp'.

It is notable that the two distributions are quite different. Since the actual blade geometry used in the PhD thesis is known, a flow evaluation with this geometry and the identified boundary conditions could also be executed. The geometries resulting from inverse design are compared to the actual design.

The target distributions are both used in an attempt to recover the actual blade design. The NASA stage 35 stator blade geometry was again used as baseline. The designs obtained with both targets are presented in figures 4.40 and 4.42. The respective surface isentropic Mach distributions are seen in figures 4.41 and 4.43. The distributions are compared with the isentropic Mach number distribution corresponding to the actual geometry used by Sans.

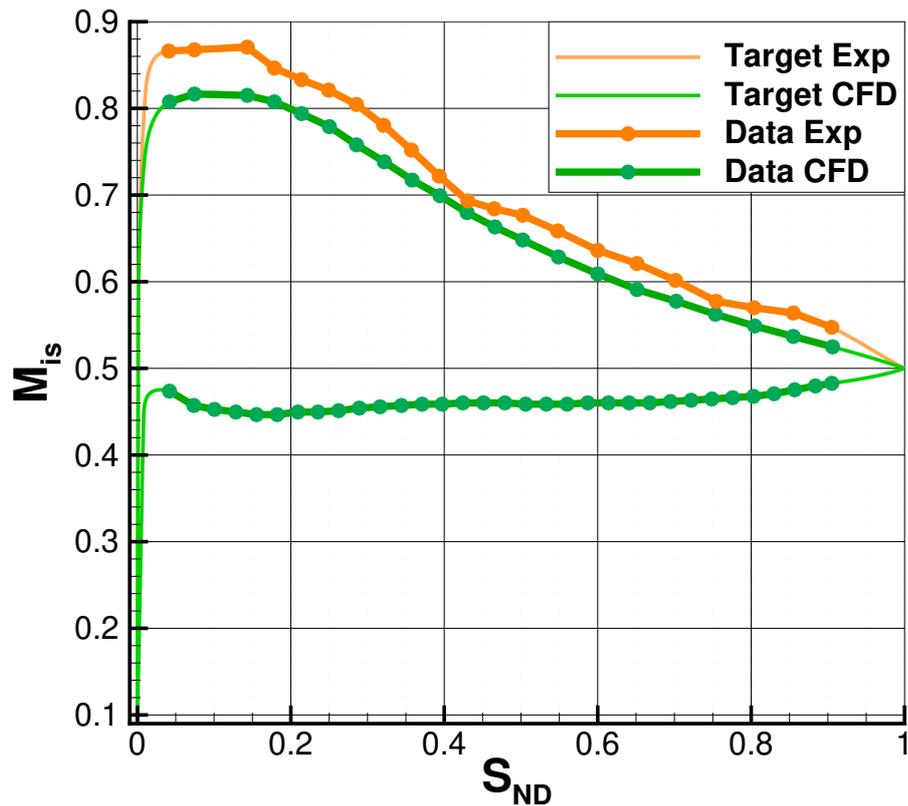


Figure 4.39: Target distributions for reverse engineering and the CFD and experimental data they are based on.

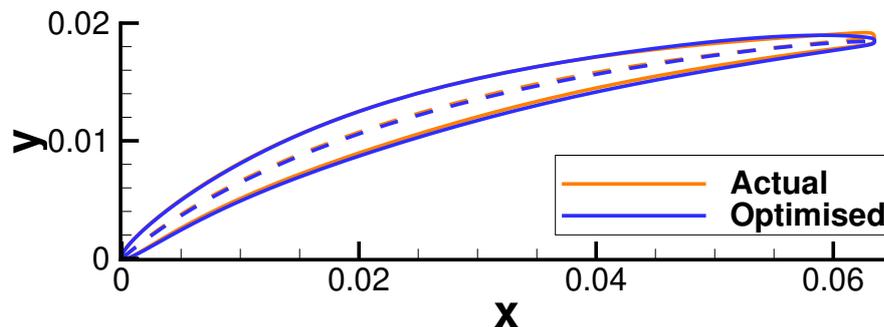


Figure 4.40: Blade geometry obtained with the CFD target distribution and the actual blade geometry.

The first target distribution was based on the published CFD data. A different flow solver, turbulence model and discretisation schemes were used in the generation of the source data. Despite this, the generated target distribution matches closely to the distribution obtained by evaluating the flow around the actual geometry. This indicates that the selected boundary conditions were correct.

In turn, the target was also matched very closely by the inverse design process. The accuracy of this result is surprising given the previous issues with converging exactly to the specified target distributions. This shows that a realistic target distribution is very important for the convergence of the inverse design method. The corresponding blade geometry nearly exactly matches the actual geometry.

The second target distribution was based on experimental data. The experimental distribution reported was not smooth. The target distribution also had not been smoothed. No experimental data for the pressure-side was available, so the same PS curve was used as for the first target. It can be seen that this distribution does not match well with the performance of the actual geometry. The parametrisation used cannot properly represent small oscillations in the blade shape. Because of this the inverse method was not able to create a design matching this target.

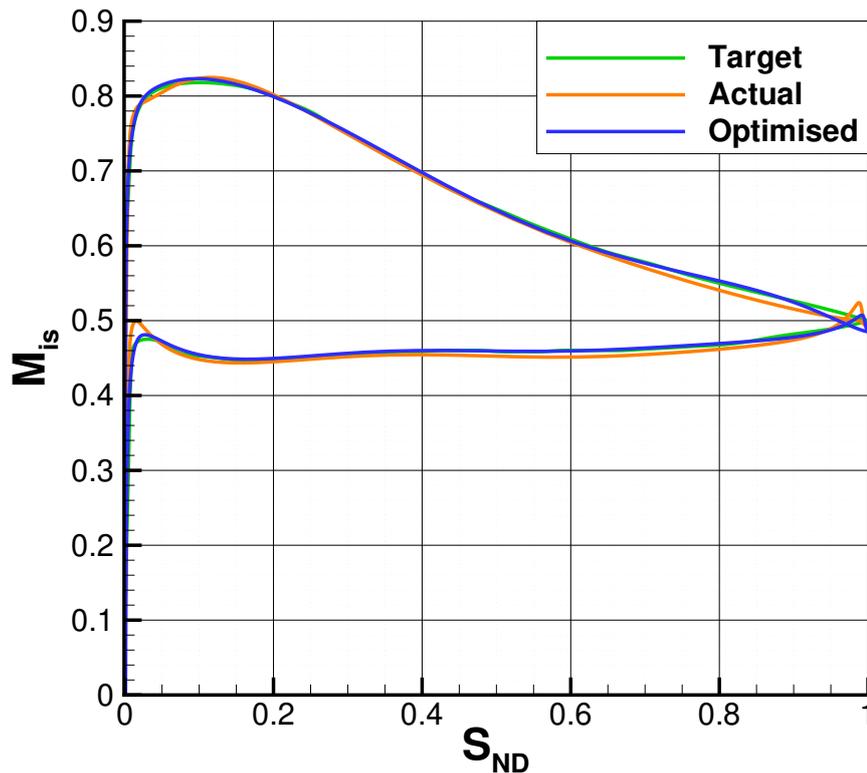


Figure 4.41: Isentropic Mach distribution resulting from inverse design with the CFD target and the distribution corresponding to the actual geometry.

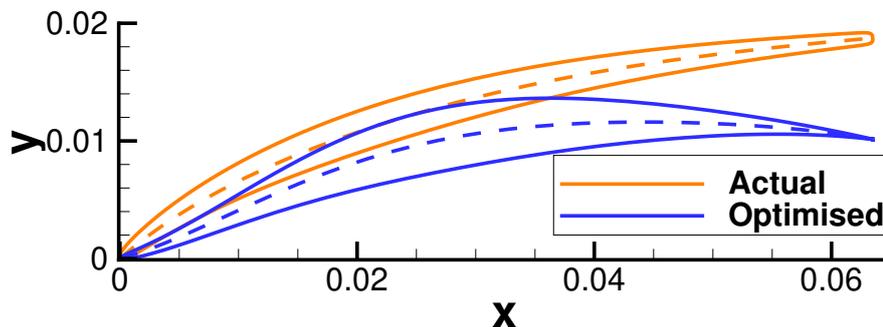


Figure 4.42: Blade geometry obtained with the experimental target distribution and the actual blade geometry.

Smoothing the experimental data may have yielded better results. Furthermore, the experimental SS velocities were higher than for CFD result. Since the same PS curve was used, this results in higher blade loading. A second modification would have been needed by changing the velocities on the pressure side to maintain similar blade loading. These issues with the data set make it unsuitable for inverse design.

It has been demonstrated that the inverse design method is not suitable for the improvement of existing designs, but can be very effective in reproducing blade geometry based purely on the published data. This is a very interesting capability with which blade designs, which are often considered intellectual property, can be retrieved based on freely available published documents.

Several other use cases are also proposed which can be investigated in following work. Since the inverse design method is significantly faster than direct optimisation methods, it could be used to create an initial design for a new situation where no suitable baseline exists yet. This design can then be optimised with direct methods. This may be computationally cheaper than starting from an arbitrary design, especially for stiff optimisation problems which converge slowly.

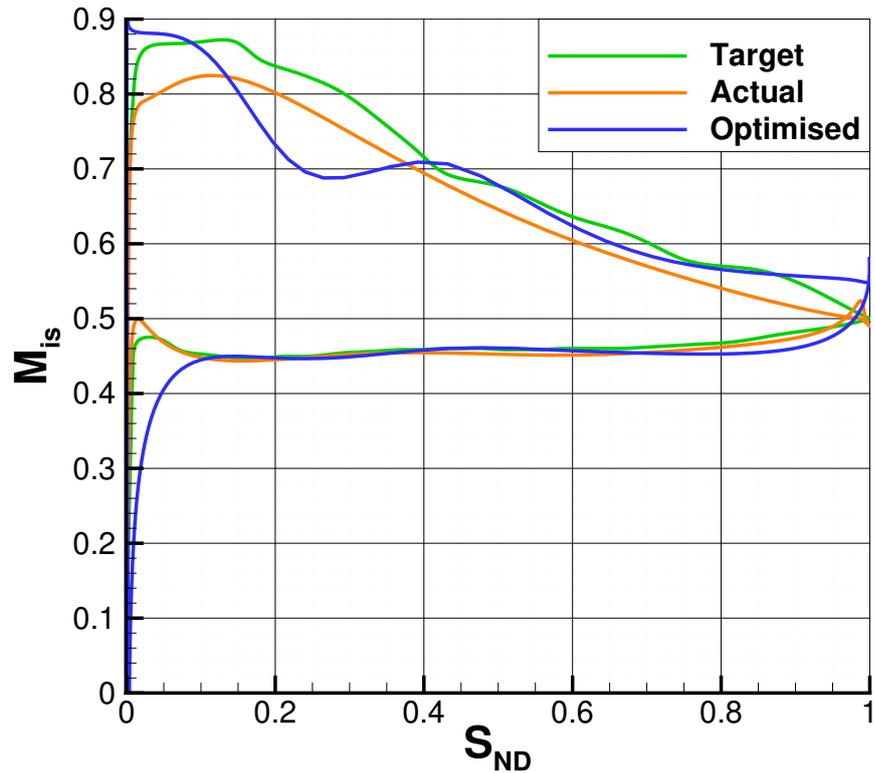


Figure 4.43: Isentropic Mach distribution resulting from inverse design with the experimental target and the distribution corresponding to the actual geometry.

Another possible use case is the adaptation of a previously optimised design to changing inlet conditions. During system design processes, the boundary conditions between elements often shift. Instead of repeating costly direct optimisations at every design iteration, an inverse design could quickly adapt the design to the latest requirements. The method could also be effective for improving the velocity profiles on transonic blades experiencing shock. By specifying a smooth deceleration, inverse design can produce a shock-free blade.

5

Conclusions

Inverse design methods are often considered outdated for detailed turbomachinery blade design. However, these methods do have some advantages over the more conventional direct design optimisation methods. Several typical shortcomings of inverse design may be avoided by adopting an optimisation strategy. Specifically, the use of adjoint-based optimisation could lead to an accurate, fast and flexible inverse design tool.

Such a tool for the inverse design of two-dimensional axial compressor blades has been developed. The tool takes as input a pressure or isentropic Mach distribution along the blade surface. It then produces the blade geometry which best matches this target distribution. The accuracy and performance of the tool was assessed and some use cases investigated. The results of these efforts are synthesised to answer the posed research questions and are addressed below.

- I. (a) Within CADO the gradients can be computed using finite difference, complex step or discrete adjoint approaches. For the latter the function of interest needs to be reverse differentiated.
(b) The adjoint solution is simplified. It's accuracy decreases with increased importance of turbulent behaviour. In those cases the complex step method is the best option.
- II. (a) Even for unrealistic target distributions the tool will try to approach this target. This can lead to unusable results. Therefore the target must be carefully crafted to be physically possible and the design space properly bounded.
(b) Also for feasible target distributions, the tool was not always able to exactly match the target. The convergence was curbed by the selection of the design parameters which is quite limited and only suitable for smooth blade designs.
(c) Other requirements, such as flow turning or cross-sectional area, can be imposed as separate constraint functions. The SQP optimisation algorithm will then take those into account. However, this results in increased computational cost and may impede convergence to the specified target.
- III. (a) The method is not suitable for the improvement of existing designs via loss reduction, but was very successful in the reverse engineering of a blade design based on published data. Other possible uses include making blades shock-free, generating initial designs for a new scenario and adapting designs to changing boundary conditions. These have not been demonstrated.
(b) No experiments could be carried out due to restricted availability of the needed facilities, nor was suitable experimental data available. However, the method was found to be in agreement with numerical results from different sources.
(c) Since no constraints need to be used, inverse design is significantly cheaper than direct design optimisation. The single-objective optimisation performed only has slightly increased costs, but needs additional constraints to produce usable results.

The development of the adjoint-based inverse method for 2D axial compressor blades was successful. The resulting tool is integrated in CADO and can be used for other projects. The assessment of its performance and use cases yielded mixed results. The adjoint calculation does not deliver accurate gradient values due to omitting the full differentiation of the turbulence model, resulting in poor convergence in some cases. Furthermore, the performance of the output design depends entirely on the quality of the targets specified.

The method thus does not provide a convenient method for design improvement, especially in the hands of a beginning designer. On the other hand it is able to effectively reproduce blade geometries based on published pressure or isentropic Mach number distributions. Other use cases can also be envisioned. Some recommendations for future work are given.

- It is crucial to complete the adjoint gradient computation. This will result in a more accurate, faster and more reliable method.
- Accuracy can further be increased by improving or substituting the parametrisation so that more shapes can be represented.
- Other potential use cases, such as creating shock-free transonic blades, should be investigated. Confirming for which problems the method provides a practical solution will expand the toolkit for turbomachinery designers.

Bibliography

- [1] M. Zhang and A. W. Rizzi, *Assessment of inverse and direct methods for airfoil and wing design*, in *Simulation-Driven Modeling and Optimization* (Springer, 2016) pp. 75–109.
- [2] A. E. Bryson, *Optimal control-1950 to 1985*, IEEE Control Systems Magazine 16, 26 (1996).
- [3] G. K. Kenway, C. A. Mader, P. He, and J. R. Martins, *Effective adjoint approaches for computational fluid dynamics*, Progress in Aerospace Sciences , 100542 (2019).
- [4] O. Pironneau, *On optimum profiles in stokes flow*, Journal of Fluid Mechanics 59, 117 (1973).
- [5] O. Pironneau, *On optimum design in fluid mechanics*, Journal of Fluid Mechanics 64, 97 (1974).
- [6] A. Jameson, *Aerodynamic design via control theory*, Journal of scientific computing 3, 233 (1988).
- [7] A. Jameson, L. Martinelli, and N. Pierce, *Optimum aerodynamic design using the Navier–Stokes equations*, Theoretical and computational fluid dynamics 10, 213 (1998).
- [8] S. Nadarajah and A. Jameson, *Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization*, in *15th AIAA Computational Fluid Dynamics Conference* (2001) p. 2530.
- [9] S. Nadarajah and A. Jameson, *A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization*, in *38th Aerospace Sciences Meeting and Exhibit* (2000) p. 667.
- [10] W. K. Anderson and V. Venkatakrishnan, *Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation*, Computers & Fluids 28, 443 (1999).
- [11] J. J. Reuther, A. Jameson, J. J. Alonso, M. J. Rimlinger, and D. Saunders, *Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 1*, Journal of aircraft 36, 51 (1999).
- [12] J. J. Reuther, A. Jameson, J. J. Alonso, M. J. Rimlinger, and D. Saunders, *Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 2*, Journal of aircraft 36, 61 (1999).
- [13] D. Wang and Y. Li, *3D direct and inverse design using ns equations and the adjoint method for turbine blades*, in *ASME Turbo Expo 2010: Power for Land, Sea, and Air* (American Society of Mechanical Engineers Digital Collection, 2010) pp. 537–545.
- [14] J. E. Peter and R. P. Dwight, *Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches*, Computers & Fluids 39, 373 (2010).
- [15] M. B. Giles and N. A. Pierce, *An introduction to the adjoint approach to design*, Flow, turbulence and combustion 65, 393 (2000).
- [16] T. Verstraete, L. Müller, and J.-D. Müller, *Multidisciplinary adjoint optimization of trubomachinery components including aerodynamic and stress performance*, in *35th AIAA Applied Aerodynamics Conference* (2017) p. 4083.
- [17] M. Pini, G. Persico, and V. Dossena, *Robust adjoint-based shape optimization of supersonic turbomachinery cascades*, in *ASME Turbo Expo 2014: Turbine Technical Conference and Exposition* (American Society of Mechanical Engineers Digital Collection, 2014).
- [18] M. Pini, P. Cinnella, et al., *Hybrid adjoint-based robust optimization approach for fluid-dynamics problems*, in *15th AIAA Non-Deterministic Approaches Conference* (2013).

- [19] P. D. Frank and G. R. Shubin, *A comparison of optimization-based approaches for a model computational aerodynamics design problem*, Journal of Computational Physics 98, 74 (1992).
- [20] O. Baysal and M. E. Eleshaky, *Aerodynamic sensitivity analysis methods for the compressible euler equations*, Journal of Fluids Engineering 113, 681 (1991).
- [21] J. Elliott and J. Peraire, *Practical three-dimensional aerodynamic design and optimization using unstructured meshes*, AIAA journal 35, 1479 (1997).
- [22] E. J. Nielsen and W. K. Anderson, *Aerodynamic design optimization on unstructured meshes using the navier-stokes equations*, AIAA journal 37, 1411 (1999).
- [23] M. B. Giles, M. C. Duta, J.-D. Muller, and N. A. Pierce, *Algorithm developments for discrete adjoint methods*, AIAA journal 41, 198 (2003).
- [24] M. B. Giles and E. Süli, *Adjoint methods for pdes: a posteriori error analysis and postprocessing by duality*, Acta numerica 11, 145 (2002).
- [25] B. Mohammadi, *Optimal shape design, reverse mode of automatic differentiation and turbulence*, in *35th Aerospace Sciences Meeting and Exhibit* (1997) p. 99.
- [26] M. B. Giles, *On the iterative solution of adjoint equations*, in *Automatic differentiation of algorithms* (Springer, 2002) pp. 145–151.
- [27] T. Verstraete, *Introduction to optimization and multidisciplinary design*, Lecture Notes (2016), von Karman Institute.
- [28] K. C. Giannakoglou and D. I. Papadimitriou, *Adjoint methods for shape optimization*, in *Optimization and computational fluid dynamics* (Springer, 2008) pp. 79–108.
- [29] A. M. Bradley, *PDE-constrained optimization and the adjoint method*, Tech. Rep. (Stanford University, 2013).
- [30] M. Nemec, D. W. Zingg, and T. H. Pulliam, *Multipoint and multi-objective aerodynamic shape optimization*, AIAA journal 42, 1057 (2004).
- [31] M. Schramm, B. Stoevesandt, and J. Peinke, *Optimization of airfoils using the adjoint approach and the influence of adjoint turbulent viscosity*, Computation 6, 5 (2018).
- [32] S. Ta'asan, G. Kuruwila, and M. Salas, *Aerodynamic design and optimization in one shot*, in *30th aerospace sciences meeting and exhibit* (1992).
- [33] S. Ta'asan, G. Kuruwila, and M. Salas, *Airfoil design and optimization by the one-shot method*, in *33d aerospace sciences meeting and exhibit* (1995).
- [34] D. Papadimitriou and K. Giannakoglou, *Direct, adjoint and mixed approaches for the computation of hessian in airfoil design problems*, International journal for numerical methods in fluids 56, 1929 (2008).
- [35] D. Papadimitriou and K. Giannakoglou, *The continuous direct-adjoint approach for second order sensitivities in viscous aerodynamic inverse design problems*, Computers & fluids 38, 1539 (2009).
- [36] D. Mavriplis, *Solution of the unsteady discrete adjoint for three-dimensional problems on dynamically deforming unstructured meshes*, in *46th AIAA Aerospace Sciences Meeting and Exhibit* (2008) p. 727.
- [37] A. Rubino, M. Pini, P. Colonna, T. Albring, S. Nimmagadda, T. Economou, and J. Alonso, *Adjoint-based fluid dynamic design optimization in quasi-periodic unsteady flow problems using a harmonic balance method*, Journal of Computational Physics 372, 220 (2018).
- [38] Y. Jarny, M. Ozisik, and J. Bardou, *A general optimization method using adjoint equation for solving multidimensional inverse heat conduction*, International journal of heat and mass transfer 34, 2911 (1991).

- [39] D. I. Karatzidis, T. V. Yioultsis, and T. D. Tsiboukis, *Gradient-based adjoint-variable optimization of broadband microstrip antennas with mixed-order prism macroelements*, AEU-International Journal of Electronics and Communications 62, 401 (2008).
- [40] O. Léonard, *Conception et développement d'une méthode inverse de type Euler et application à la génération de grilles d'aubes transsoniques*, Ph.D. thesis, von Karman Institute (1992).
- [41] M. J. Lighthill, *A mathematical method of cascade design* (HM Stationery Office, 1945).
- [42] M. J. Lighthill, *The hodograph transformation in trans-sonic flow*, Proceedings of the Royal Society of London 191, 323 (1947).
- [43] F. Bauer, P. Garabedian, and D. Korn, *A Theory of Supercritical Wing Sections, with Computer Programs and Examples: With Computer Programs and Examples* (Springer, 1972).
- [44] A. Chikitkin, M. Petrov, R. Dushkov, and E. Shifrin, *Aerodynamic design of a laval nozzle for real gas using hodograph method*, Aerospace 5, 96 (2018).
- [45] E. Martensen, *Berechnung der druckverteilung an gitterprofilen in ebener potentialströmung mit einer fredholmschen integralgleichung*, Archive for Rational Mechanics and Analysis 3, 235 (1959).
- [46] K. Murugesan and J. Raily, *Pure design method for aerofoils in cascade*, Journal of Mechanical Engineering Science 11, 454 (1969).
- [47] R. Van den Braembussche, O. Léonard, and L. Nekmouche, *Subsonic and transonic blade design by means of analysis codes*, AGARD Computational methods for aerodynamic design (inverse) and optimization 463 (1989).
- [48] A. Demeulenaere and R. Van den Braembussche, *Three-dimensional inverse method for turbomachinery blading design*, Journal of Turbomachinery 120, 247 (1998).
- [49] L. de Vito, R. Van den Braembussche, and H. Deconinck, *A novel two-dimensional viscous inverse design method for turbomachinery blading*, J. Turbomach. 125, 310 (2003).
- [50] W. Hawthorne, C. Wang, C. Tan, and J. McCune, *Theory of blade design for large deflections: Part I—two-dimensional cascade*, Journal of Engineering for Gas Turbines and Power 106, 346 (1984).
- [51] W. Hawthorne, C. Wang, C. Tan, and J. McCune, *Theory of blade design for large deflections: Part II—annular cascades*, Journal of Engineering for Gas Turbines and Power 106, 354 (1984).
- [52] M. Zangeneh, *A compressible three-dimensional design method for radial and mixed flow turbomachinery blades*, International Journal for Numerical Methods in Fluids 13, 599 (1991).
- [53] K. Yiu and M. Zangeneh, *A 3D automatic optimization strategy for design of centrifugal compressor impeller blades*, in *ASME 1998 International Gas Turbine and Aeroengine Congress and Exhibition* (American Society of Mechanical Engineers Digital Collection, 1998).
- [54] D. Bonaiuti and M. Zangeneh, *On the coupling of inverse design and optimization techniques for the multiobjective, multipoint design of turbomachinery blades*, Journal of Turbomachinery 131 (2009).
- [55] L. Zhang, G. Davila, and M. Zangeneh, *Multi-objective optimization of a high specific speed centrifugal volute pump using 3D inverse design coupled with CFD simulations*, in *8th Joint Fluids Engineering Conference* (American Society of Mechanical Engineers Digital Collection, 2019).
- [56] M. Ferlauto, A. Iollo, and L. Zannetti, *Optimal inverse method for turbomachinery design*, in *Proceedings of the ECCOMAS* (2000).
- [57] A. Iollo, M. Ferlauto, and L. Zannetti, *An aerodynamic optimization method based on the inverse problem adjoint equations*, Journal of Computational Physics 173, 87 (2001).

- [58] I. Sanchez Torreguitart, T. Verstraete, and L. Müller, *Optimization of the LS89 axial turbine profile using a CAD and adjoint based approach*, International Journal of Turbomachinery, Propulsion and Power 3, 20 (2018).
- [59] I. Sanchez Torreguitart, T. Verstraete, and L. Müller, *CAD and adjoint based multipoint optimization of an axial turbine profile*, in *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems* (Springer, 2019) pp. 35–46.
- [60] M. Schobeiri and S. Abdelfattah, *On the reliability of RANS and URANS numerical results for high-pressure turbine simulations: a benchmark experimental and numerical study on performance and interstage flow behavior of high-pressure turbines at design and off-design conditions using two different turbine designs*, Journal of turbomachinery 135 (2013).
- [61] M. Zangeneh, A. Goto, and H. Harada, *On the role of three-dimensional inverse design methods in turbomachinery shape optimization*, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 213, 27 (1999).
- [62] Y. Zhu, Y. Ju, and C. Zhang, *An experience-independent inverse design optimization method of compressor cascade airfoil*, Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy 233, 431 (2019).
- [63] B. Ziegler, *Adjoint method-based inverse design of transonic compressor cascade with boundary layer control*, Progress in Computational Fluid Dynamics, an International Journal 17, 335 (2017).
- [64] J. Lu, C. Zhang, and Z. Feng, *Aerodynamic optimization and inverse design of 2D and 3D turbine cascades using the discrete adjoint method*, in *ASME Turbo Expo 2013: Turbine Technical Conference and Exposition* (American Society of Mechanical Engineers Digital Collection, 2013).
- [65] Z. Ali and P. G. Tucker, *Multiblock structured mesh generation for turbomachinery flows*, in *Proceedings of the 22nd International Meshing Roundtable* (Springer, 2014) pp. 165–182.
- [66] J. Steger and R. Sorenson, *Automatic mesh-point clustering near a boundary in grid generation with elliptic partial differential equations*, Journal of Computational Physics (1979).
- [67] D. Kraft et al., *A software package for sequential quadratic programming*, DFVLR Obersfaffehofen (1988).
- [68] L. Reid and R. D. Moore, *Design and overall performance of four highly loaded, high speed inlet stages for an advanced high-pressure-ratio core compressor*, NTRS (1978).
- [69] J. Sans, *Experimental and numerical investigations of solidity effects on compressor performance and stability*, Ph.D. thesis, UCL-Université Catholique de Louvain (2015).
- [70] L. Reid, *Performance of single-stage axial-flow transonic compressor with rotor and stator aspect ratios of 1.19 and 1.26, respectively, and with design pressure ration of 1.82*, NTRS (1978).