von Karman Institute for Fluid Dynamics
Aeronautics and Aerospace Department

Universitat Politècnica de Catalunya
Escola Superior d'Enginyeries Industrial, Aeronàutica i Audiovisual de Terrassa

Efficient CAD based adjoint optimization of turbomachinery using an adaptive shape parameterization

Ismael Sanchez Torreguitart

April 2019

Thesis presented by Dipl.-Ing. Ismael Sanchez Torreguitart in order to obtain the degree of *Doctor en Enginyeria Tèrmica*, Universitat Politècnica de Catalunya, Barcelona, 23 May 2019.

Promoters:
Prof. Dr. Tom Verstraete (von Karman Institute for Fluid Dynamics, Belgium)
Prof. Dr. Assensi Oliva (Universitat Politècnica de Catalunya, Spain)
Prof. Dr. Carles David Perez Segarra (Universitat Politècnica de Catalunya, Spain)

Doctoral Committee:
Prof. Antonio Lecuona Newman, Dept. Ingeniería Térmica y de Fluidos, Universidad Carlos III de Madrid (UC3M)
Prof. Joaquim Rigola Serrano, Dept. Màquines i Motors Tèrmics, Universitat Politècnica de Catalunya (UPC)
Prof. Kyriakos C. Giannakoglou, Lab. of Thermal Turbomachines, Parallel CFD and Optimization Unit, National Technical University of Athens (NTU)

Confía de todo corazón en el Señor
y no en tu propia inteligencia.
Ten presente al Señor en todo lo que hagas
y él te llevará por el camino recto.
Proverbios 3:5-6

Acknowledgements

This work has been conducted within the IODA project (http://ioda.sems.qmul.ac.uk). The author would like to thank you the European Union for funding this work through the European Union HORIZON 2020 Framework Programme for Research and Innovation under Grant Agreement No. 642959.

I would like to thank Prof. Dr. Tom Verstraete, Prof. Dr. Assensi Oliva, Prof. Dr. Carles David Segarra and Dr. Lasse Mueller for their close supervision, for sharing their technical expertise with me and for reviewing the contents of this work. I would also acknowledge all the technical help and support from Mladen Banovic and the ADOL-C group from Paderborn University.

Also, I would like to thank God for giving me his strength to not give up during the hardest moments. Special thanks to my wife Mireia Pitarch Diago for her continuous support and encouragement, and to my family for their sacrifice all these years. Finally, I would like to give special thanks my grandparents and my aunty Jemima Sanchez, who is now in heaven, for their constant prayers. Last but not least, I would like to dedicate this thesis to my little brother Jeremi Sanchez Torreguitart, hoping that he will, one day, make many discoveries and contributions to science using all the gifts he has been given.

Abstract

Nowadays, the improvement of turbomachinery components within the gas turbine industry relies heavily on the use of high-fidelity optimization techniques and the user experience with a range of specialized sub-tools such as: the geometry modeler, the grid generator, the flow solver and the optimizer. Since the CFD analysis can take very long run times and often very rich design spaces are explored, the gradient-free optimization approaches can be extremely expensive for industrial applications. In this regard, the use of a gradient-based approaches, such as the adjoint CFD methods, is desirable, since they are computationally efficient specially for large number of design variables. The adjoint method is a popular technique to obtain the gradients of an objective function with respect to the grid coordinates at a relatively low cost. A common approach is to use the grid point coordinates as design variables, which offers a very rich design space but the connection to the CAD geometry is lost. This problem can be partially solved by transforming the optimal shape defined by grid points back to smooth CAD shape, but this step can take significant time and it is not guaranteed that the final approximated CAD shape will meet the design requirements and constraints. Therefore, the ideal would be to use the CAD geometry modeler within the optimization framework, since CAD is the industry adopted standard for the design of components. However, in the context of gradient-based optimization, this requires an additional sensitivity computation step. The partial derivative of the grid coordinates with respect to the CAD-based design variables, referred in this work as the grid sensitivities, will need to be computed as well. This can be done with finite differences, by perturbing the design variables to generate perturbed CAD models. However, some difficulties can arise if the CAD topology changes or the labels change when a design variable is updated, making this approach not feasible.

In this work, the difficulties of using CAD and adjoint based methods for the optimization of turbomachinery components are addressed and will be tested in the LS89 turbine cascade. First, the CAD kernel and the grid generator will be differentiated using Algorithmic Differentiation (AD), delivering the grid sensitivities. The use of AD to compute the grid sensitivities allows to circumvent the CAD topology and labelling problems that can arise when using finite differences and also allows to deliver accurate grid sensitivities. Secondly, the integration of the CAD in the design chain will be demonstrated in a single-point and multi-point aerodynamic optimization of the LS89 turbine cascade application. The performance gradients required by the optimizer will be computed by combining the sensitivities provided by the adjoint solver and the grid sensitivities computed by AD. Substantial aerodynamic improvements will be presented in this work for the optimization of the LS89 for both design point and off-design conditions. However, the aerodynamic improvements are bound to the CAD-based parameterization used throughout the optimization, and it could happen that the parameterization is not good enough to move the model boundary in the direction indicated by the adjoint sensitivities. To investigate this, a novel parametric effectiveness indicator

will be defined, which allows to rate the quality of a set of CAD based parameters to produce the shape change that the adjoint sensitivities dictate in a constrained or unconstrained problem. If, in the beginning of the optimization, the parametric effectiveness is high, it is expected to reach a final solution with increased performance. Finally, the use of adaptive shape parameterization or multilevel optimization, where the optimization starts with a small number of design variables (coarse parameterization) and ends with a large number (fine parameterization), will be investigated, since this can be an effective strategy to explore rich design spaces and to improve convergence rate, robustness and final solution of the optimization. For this purpose, the parametric effectiveness will be used in a novel multilevel shape refinement procedure to introduce the minimum amount of design variables required to modify the shape in the direction the adjoint sensitivities dictate. The novelty of the multilevel optimization strategy will lie in using the parametric effectiveness to refine the design space and create the best parameterization to be used during the optimization.

Contents

List of Figures

List of Tables

Nomenclature

Roman Symbols

a	Coefficienct of the Hicks and Henne sine "bump" functions
b	Coefficienct of the Hicks and Henne sine "bump" functions
c	chord length
c_{ax}	axial chord length
C_i	continuity of a curve or surface
C_b	base pressure coefficient
$C(u)$	parametric curve (e.g. B-spline curve)
$dJ/d\vec{X}$	performance sensitivities w.r.t. the volume grid
$dJ/d\vec{X}_S$	performance sensitivities w.r.t. the surface grid
$(dJ/d\vec{X}_S)_c$	constrained performance sensitivities w.r.t. the surface grid
$\partial R/\partial\vec{U}$	Jacobian of the residual of the CFD solver
$d\vec{U}/dt$	temporal derivative of the fluid system variables
$d\vec{U}/d\vec{X}$	change of the fluid system state when te grid changes
$f_j(x)$	Hicks and Henne sine "bump" functions
f_k	elementary functions (in Sec. 2.6.1)
g	pitch
g_1	inequality constraint
g_2	equality constraint
H	boundary layer shape factor
I_i	unit vector of of the ith column of an identity matrix
I	one of the indexes of a three dimensional structured grid (in Sec. 2.5.1), inverse of the jacobian (in Sec. 2.5.3)
J	objective function, index of a three dimensional structured grid (in Sec. 2.5.1)
J_{SP}	single point pseudo cost function
J_{MP}	multi point pseudo cost function
$\vec{J}$	vector of outputs of a computer program (in Sec. 2.6.1)
k_{PS}	stretching factor for the pressure side streamwise control point distribution
k_{SS}	stretching factor for the suction side streamwise control point distribution
K	index of a three dimensional structured grid (in Sec. 2.5.1)
m	total number of knots minus one (in Sec. 2.4)
$\dot{m}_{out}$	exit mass flow
M	Mach number
M_{ise}	isentropic Mach number

$M_{ise,2}$	downstream isentropic Mach number
n	total number of control points minus one (in Sec. 2.4)
N	Gram matrix (in Sec. 6.2.4)
$N_{i,p}$	B-spline basis functions of degree p
$\mathcal{O}$	truncation error
p	degree of a B-spline curve (in Sec. 2.4), static pressure (in 4.3)
p_2	outlet (downstream) static pressure
P	control function of the elliptical grid generation equations (in 2.5.1), total pressure (in Sec. 4.3)
P_{01}	inlet total pressure
P_{02}	outlet (downstream) total pressure
$P_{01} - P_{02}$	total pressure loss between inlet and outlet
$\vec{P}$	1-dimensional array of control points of a B-spline curve
q_X	downstream dynamic head at the plane X
q_2	outlet (downstream) dynamic head
Q	control function of the elliptical grid generation equations (in Sec. 2.5.1)
$\vec{R}$	non-linear residual of the CFD solver
R_{LE}	leading edge radius
R_{TE}	trailing edge radius
s	normalised arc-length control function used in TFI (in Sec. 2.5.1)
t	time (in Sec. 2.2.4), normalised arc-length control function used in TFI (in Sec. 2.5.1), throat height (in chapters 3, 4, 5, 6 and 7)
t^i_{PS}	blade i-th PS thickness
t^i_{SS}	blade i-th SS thickness
u^i_{PS}	blade i-th PS streamwise position of the control point
u^i_{SS}	blade i-th SS streamwise position of the control point
$\dot{t}_i$	derivative values of the intermediate variables of a computer program w.r.t. the inputs (in Sec. 2.6.1)
$\overline{\dot{t}}_i$	adjoint variable (in Sec. 2.6.1)
$\vec{t}$	intermediate variables of a computer program (in Sec. 2.6.1)
u	parametric length of a B-spline curve (in Sec. 2.4), blending formula used for TFI (in Sec. 2.5.1)
$\vec{u}$	knot vector of a B-spline curve
u_i	individual knot of a B-spline curve
U_∞	free stream velocity
$\vec{U}$	flow state variables
v	blending formula used for TFI (in Sec. 2.5.1)
$\vec{v}$	adjoint variable
$\overline{V_y}$	mass averaged axial velocity
$\overline{V_y}$	mass averaged vertical velocity

V_n	design velocities
$\vec{x}$	input to a computer program (in Sec. 2.6.1)
$\vec{X}$	grid points
$\vec{X}_S$	grid points on the shape surface
X_i	input to a computer program (in Sec. 2.6.1)
Z_w	Zweifel loading coefficient

Greek Symbols

$\vec{\alpha}$	design variables
α	scale metric factor of the grid generation coordinate transformation (in Sec. 2.5.3)
$dJ/d\vec{\alpha}$	performance sensitivities (w.r.t. the CAD parameters)
$d\vec{x}/d\vec{\alpha}$	geometrical sensitivities
$dM/d\alpha_i$	magnitude of the volume grid sensitivities
$d\vec{X}/d\vec{\alpha}$	volume grid sensitivities
$(d\vec{X}/d\alpha)_{best}$	best possible design movement (in Sec. 6.2.4)
$d\vec{X_S}/d\vec{\alpha}$	surface grid sensitivities
β	scale metric factor of the grid generation coordinate transformation (in Sec. 2.5.3)
β_{in}	blade inlet metal angle
β_{out}	blade outlet metal angle
δ	finite difference step size (in Sec. 2.2), boundary layer thickness (in Sec. 4.3.2)
δ^*	boundary layer thickness displacement
$\Delta J_{CAD,OBM}$	change in performance that can be achieved using existing parameterization, normalized w.r.t. the overall boundary movement.
$\Delta J_{ideal,OBM}$	change in performance that can be achieved if the model is not constrained by any parameterization and follows the shape changes dictated by the adjoint sensitivities, normalized w.r.t. the overall boundary movement. normalized w.r.t. the overall boundary movement.
ϵ	coordinate of the uniform orthogonal computational space $\epsilon\eta$ (in Sec. sec:ch2GridGeneration), blade rear suction side turning (in chapters 4, 5, 6 and 7)
$\epsilon_{xx}, \epsilon_{yy}$	second partial derivative of the computational space coordinate ϵ w.r.t to the physical space X and Y coordinates
η	coordinate of the uniform orthogonal computational space $\epsilon\eta$
η_{global}	parametric effectiveness
γ	scale metric factor of the grid generation coordinate transformation (in Sec. 2.5.3), blade stagger angle (in Sec. 3.2.1, chapters 4, 5, 6 and 7) step size in Sec. 6.3.1
ω	penalty coefficient
θ	boundary layer momentum thickness
φ_{PS}	blade pressure side trailing edge wedge angle

φ_{SS}	blade suction side trailing edge wedge angle
σ	blade solidity
ζ_2	total pressure loss coefficient
ζ_p	profile losses

Subscript

ise	isentropic
$ise, 2$	downstream isentropic
MP	multi point
op	operating point
out	outlet (exit)
PS	pressure side
ref	reference or baseline
SS	suction side
SP	single point
TE	trailing edge
01	total quantity at the inlet
2	static quantity at the outlet

Acronyms

AD	Algorithmic Differentiation
ADOL-C	Automatic Differentiation by OverLoading in C++
BFGS	Broyden-Fletcher-Goldfarb-Shanno
CAD	Computer Aided Design
CADO	Computer Aided Design and Optimization
CAE	Computer Aided Engineering
CFD	Computational Fluid Dynamics
DV	Design Variables
FD	Finite Differences
GMRES	Generalized Minimum Residual
KKT	Karush-Kuhn-Tucker
MUSCL	Monotonic Upwind Scheme for Conservation Laws
OBM	Overall Boundary Movement
PDEs	Partial Differential Equations
PE	Parametric Effectiveness
RANS	Reynolds Averaged Navier-Stokes
SNOPT	Sparse Nonlinear OPTimizer
SVD	Single Value Decomposition

STEP	Standard for the Exchange of Product model data
TFI	Transfinite Interpolation
VKI	von Karman Institute for Fluid Dynamics

Chapter 1.

Introduction

Abstract. This chapter serves to introduce the main issues which industry is currently facing in the context of aerodynamic shape optimization and to explain the contributions of this thesis to alleviate these problems. Finally, the outline of this thesis is presented.

1.1. Motivation

Many attempts have been made to improve the design of gas turbines over the last decades. The main efforts have been concentrated in improving the thermal and propulsive efficiencies, extending the life of the components, increasing the power and reducing the emissions. Most of the technological improvements have come with the computational advances. First, the use of high-fidelity analysis has deepened the understanding of the flow physics and this has allowed to make engineering judgements to further improve the designs. Also, the introduction of shape optimization in the engineering design framework has enabled to find non-intuitive and innovative geometries with better performance. Therefore, increasing the use of high-fidelity tools and shape optimization at the early stage of the design phase can significantly reduce the time required to find the optimal geometry and handing it over to manufacturing. However, nowadays the industry still faces some challenges associated to the use of shape optimization within the design phase. This thesis examines some of the main bottlenecks related to the use of gradient-based optimization techniques in an industrial design framework. The methodologies proposed in this thesis allow to integrate the Computer Aided Design (CAD) within the optimization and to improve the robustness, completeness and efficiency of the design system in order to explore high-dimensional design spaces at a relatively low computational cost.

In aerodynamic shape optimization, the problem typically consists in minimizing certain performance measure subject to certain manufacturing and aerodynamic constraints within limited computational resources. Typically, the grid node coordinates are used as design variables [3–5] and hence the link to the geometry defined in the CAD model is lost. This means that it is required to approximate the optimal shape defined by the cloud of points by a smooth CAD shape [6, 7] and this is a complex step that can take significant time and can impair the optimality of the final shape. Another drawback of allowing every grid point to move in each spatial direction during the optimization is that it is very difficult to satisfy any geometry or assembly constraints. For example, keeping the blade trailing edge thickness or the axial chord length fixed cannot be imposed directly on the grid nodes. Since the industrial standard for representing a shape is through CAD, this work examines integrating the CAD in a gradient-based shape optimization process. This introduces an additional bottleneck: the computation of the grid sensitivities, which represent the geometric derivative of the surface grid point coordinates w.r.t. CAD-based design parameters, is required. Typically, this can be done by using finite differences but this is restricted to problems where the topology (i.e., the number and arrangement of surfaces, edges and vertices over the model boundary) remains constant. Another approach is to use the design velocity approach [8] to compute these shape sensitivities, but this approach also introduces inaccuracies which originate from the surface mesh to mesh projection errors and the finite difference sources of error (i.e. the limited arithmetic precision and the truncation error). Both the finite difference approach and the design velocity approach can introduce large inaccuracies in the computation of the gradients if an appropriate step size is not chosen. This thesis addresses these issues and proposes a alternative strategy to compute the shape sensitivities, which consists in differentiating the CAD kernel and grid generator using algorithmic differentiation.

Since the full shape optimization problem is continuous but, in most cases, there is not any analytical solution, the problem is solved numerically by projecting it unto a finite set of design variables that govern the shape changes. Despite the integration of the CAD in the design and optimization framework can guarantee the satisfaction of manufacturing constraints and the optimality of the shape, it can also introduce other type of problems related to the efficiency of the optimization process. For example, too large number of design variables can increase significantly the time required to update the CAD model. Also, too few or too large number of design variables can lead to slow design improvement. Furthermore, since the choice of the CAD parameters constraints how the model boundary can change, a poor choice may prevent the exploration of innovative geometries. This thesis tackles these issues by developing two novel methodologies. First, a novel parametric effectiveness indicator is used to rate the quality of a set of CAD design variables to move the geometry boundary in the ideal direction dictated by the adjoint solution. The parametric effectiveness aids the designer to choose what is the best CAD-based parameterization to use for optimization. Secondly, this parametric effectiveness is used to develop a multilevel strategy in which the optimizer performs a rapid, low-dimensional exploration of the design space first and then progressively explores higher-dimensional design spaces that approximate the continuous problem. This allows to create the best CAD-based parameterization to use during the multilevel optimization and to find non-intuitive geometries. This progressive parameterization strategy results in a more autonomous, robust and efficient design methodology.

1.2. Contribution of this thesis

This work has been conducted within the Industrial Optimal Design using Adjoint CFD (IODA) project (http://ioda.sems.qmul.ac.uk) and funded by through the European Union HORIZON 2020 Framework Programme for Research and Innovation under Grant Agreement No. 642959. The central goal of this thesis is to propose efficient CAD-based and adjoint methodologies to tackle the main issues that the industry is currently facing in the context of aerodynamic shape optimization. The broad contributions of this thesis are:

- Differentiation of the CAD kernel and grid generator to deliver accurate sensitivities. This approach allows to: (1) circumvent the inaccuracies introduced by the design velocity approach and finite differences; (2) avoid CAD topology and labelling problems that can arise when using finite differences.

- Demonstration of the CAD integration in an optimization framework. The performance gradients propagated to the CAD design variables will be compared to finite differences and used in single point and multi-point optimizations of the LS89 to improve the aerodynamic performance of the turbine cascade at design point and off-design conditions respectively. The integration of the CAD in the optimization frameworks allows to: (1) guarantee geometry or assembly constraints; (2) produce smooth geometries; (3) deliver the optimal geometry in the industry accepted format for manufacturing (there is no longer the need to convert the optimal shape defined by a cloud of points to a CAD geometry).

- Definition of a novel parametric effectiveness indicator to rate how good a set of CAD-based parameters is to move the model boundary in the direction that the adjoint sentitivities dictate in a constrained optimization problem. The constraints can be of any type: aerodynamic, geometrical, structural, etc. A high parametric effectiveness is a key to the success of any optimization process. The parametric effectiveness can be used, at least, in the following three manners: (1) to select the best CAD-based parameterization to use for optimization; (2) to create a non-intuitive CAD-based parameterization that is ideal for moving the geometry boundary in the desired direction; (3) to select the most effective subset of parameters to use in an optimization.

- Definition of a novel multilevel shape refinement approach, which uses the parametric effectiveness for efficient CAD-based adjoint optimization. This strategy allows to create the best CAD-parameterization to use during the optimization and to find non-intuitive geometries in an efficient manner.

In this thesis, no contributions were made towards the development of the CFD solver, the ajoint solver and the optimizers, which were treated as black-boxes. The differentiation of the CAD kernel and grid generator was done using ADOL-C [9], an AD tool developed at the Department of Mathematics at the University of Paderborn that is written in C++ and uses overloaded operators and functions.

1.3. Outline of the thesis

The thesis is organized through the following chapters.

Chapter 1 introduces the main issues that the industry is currently facing in the context of aerodynamic shape optimization and to explain the contributions of this thesis to alleviate these problems.

Chapter 2 contains a literature review of the most widely used methods for gradient evaluation, shape parameterization and CAD-based optimization. The shortcomings of using finite differences or the design velocity approach for the computation of gradients in a CAD and adjoint based optimization framework are explained. It also describes in detail the main methods used in this thesis to generate the CAD geometry and the fluid mesh, to perform the flow analysis and to handle the flow sensitivity aspects. The benefits of using adaptive shape parameterization in an optimization framework, as opposed to a static parameterization, are also introduced.

Chapter 3 presents an alternative approach to finite differences and the design velocity approach to compute the grid sensitivities. The differentiation of the CAD kernel and grid generator is proposed in order to compute accurate sensitivities and to avoid the robustness issues that can arise with finite differences.

The integration of the CAD kernel and grid generator within a CFD adjoint-based optimization framework, by expressing the optimization problem through CAD parameters, is demonstrated by performing a single point optimization of the LS89 turbine cascade in chapter 4. Large aerodynamic improvements are reported at design point and the aerodynamic performance of the optimal design is analysed in detail. However, the performance of the optimal design deteriorates rapidly at higher outlet isentropic Mach numbers.

Chapter 5 builds up on the results presented in chapter 4 by extending the study to a multi-point optimization, in order to improve the off-design performance of the turbine cascade.

Chapter 6 defines a novel parametric effectiveness indicator, which can be used to rate the ability of a set of CAD-design variables to move the geometry boundary in the direction dictated by the adjoint solution. This allows to select and to create the best CAD-parameterization to use for optimization purposes. The parametric effectiveness deals with constraints of any type: geometric, structural, aerodynamic, etc.

Chapter 7 presents an efficient strategy that uses the parametric effectiveness to dynamically create the best CAD-based parameterization during the course of the optimization. The proposed method allows to preserve the richness of the design space by progressively refining the CAD model in a multi-level optimization framework.

Finally, the thesis terminates with the main conclusions and future work.

Chapter 2.

Optimization using CAD and adjoint based methods

In chapter 1, an introduction to the main problems that the industry is currently facing in the context of CAD-based optimization is presented and the main contributions of this thesis are outlined. A broader literature review is presented in this chapter, together with the description of the main methods used in this thesis.

Abstract. This chapter serves to present some of the most widely used methods in the literature for gradient evaluation, shape parameterization and CAD-based optimization. It also presents the main methods used in this thesis to generate the CAD geometry and the fluid mesh, to perform the flow analysis and to handle the flow sensitivity aspects. Finally, the benefits of using adaptive shape parameterization in an optimization framework, as opposed to a static parameterization, are also introduced.

2.1. Optimization methods

Optimization methods can be broadly distinguished in two main families: gradient-free (global) and gradient-based (local). Gradient-free methods only use the value of the objective function, whilst gradient-based methods require the computation of the gradient of the cost function and constraints. Each of these two main families have their own advantages and disadvantages. The computational resources for the global optimization algorithms, such as genetic algorithms or particle swarm methods, grows exponentially with the number of design variables and hence their use in industry chain design processes is sometimes restricted to very small test cases. The gradient-free methods are suitable for performing a global exploration of the design space using a population of individuals and identifying the region where the global minimum is located but this comes at the cost of low convergence rates and expensive computational costs [10, 11]. On the other hand, the use of local optimization algorithms, such as gradient based methods, can reduce significantly the computational effort, provided that the gradient is computed with the adjoint method, for which the cost is independent of the number of design variables [12]. However, the local optimization algorithms might find a local minimum in the design space instead of the global one.

In the context of this thesis, gradient-based approaches such as the adjoint CFD methods will be chosen for optimization purposes.

2.2. Literature review on gradient evaluation methods

The design of complex engineering systems and subsystems involves analysing solutions in an iterative manner by perturbing the design variables $\vec{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_n)$ of a baseline system. For gradient-based optimization strategies, the iterative process includes the generation of numerical solutions, determination of design sensitivities and the provision of these gradients to an optimizer which finds a new set of design variables by perturbing them in the appropriate direction. These perturbations are based on the performance sensitivities $dJ/d\vec{\alpha}$, which express the derivative of the objective function J (e.g. total pressure losses, entropy generation, maximum Von Misses stress, etc.) with respect to the design parameters. This is usually done by calculating the derivatives separately for the various disciplines used in the system analysis (e.g. grid generation, Computational Fluid Mechanics, Computational Solid Mechanics, etc.) and then combining them using the chain rule to obtain the total gradient of the system. There are different available techniques to compute the gradient of the objective function J with respect to the design variables . The derivatives may be obtained analytically, by differentiating the analysis code, or numerically, by using finite differences. The choice of the gradient evaluation technique is very important, since the speed of the gradient evaluation and the accuracy of the gradient can affect significantly the efficiency of the gradient based optimization algorithm. The advantages and downsides of these methods are presented in more detail in the following subsections.

2.2.1. Finite Difference Method

Using Finite Differences (FD) to compute the performance sensitivities $dJ/d\vec{\alpha}$ is the most simple method. One can use the forward (Eq. 2.1), backward (Eq. 2.2) or central difference (Eq. 2.3) schemes to compute the performance sensitivites.

$$\frac{dJ}{d\vec{\alpha}} = \frac{J(\alpha + \delta I_i) - J(\alpha)}{\delta} + \mathcal{O}(\delta) \tag{2.1}$$

$$\frac{dJ}{d\vec{\alpha}} = \frac{J(\alpha) - J(\alpha - \delta I_i)}{\delta} + \mathcal{O}(\delta) \tag{2.2}$$

$$\frac{dJ}{d\vec{\alpha}} = \frac{J(\alpha + \delta I_i) - J(\alpha - \delta I_i)}{2\delta} + \mathcal{O}(\delta^2) \tag{2.3}$$

where δ is a chosen step size and I_i is the unit vector of the ith column of an identity matrix. In order to compute $dJ/d\vec{\alpha}$ at each optimization iteration, it would be required to evaluate the cost function $J(\alpha)$ only once but the term $J(\alpha + \delta I_i)$ or $J(\alpha - \delta I_i)$ would need to be computed as many times as the number of design variables. Therefore, the cost of evaluating the gradient with FD at each optimization iteration is proportional to the number of design variables. This can represent a large computational effort and the designer might decide to explore more affordable design spaces by reducing the number of design variables. Secondly, using FD introduces inaccuracies in the gradients. The sources of the inaccuracies are two: the truncation error $\mathcal{O}$ and the subtractive cancellation error. The inaccuracies are very sensitive to the chosen step size and the step size needs to be relatively small to have a small truncation error. However, if the step size is too small, the numerical representation of the subtraction operation becomes less accurate due to the limited machine accuracy. There is an optimum step size which would yield the lowest inaccuracies but the designer does not know a priori what step size is the best for each design variable. In order to compute a reasonably accurate finite-difference gradient, it would be necessary to compute the gradient for a range of m step sizes and select the best step size. This would further increase the computational cost associated to the evaluation of the gradients for one optimization iteration, which would then be proportional to $(m \cdot n)$.

Despite the disadvantages of the FD method, it has been employed widely for gradient calculations mainly due to its simplicity of implementation. In [13] and [14] a closed-source commercial tool was used to represent the geometry of a 3D air duct of the automotive industry and the geometry of an axial compressor stator respectively. In both cases the geometry was optimized using CAD and adjoint based approaches, where the gradient of the grid coordinates with respect to the CAD parameters was calculated by finite-differences approximations. Chand et al. [15] optimized a turbine blade profile using the Navier-Stokes equations with the loss coefficient as the objective function and the mass flow rate, the blade loading and the cross sectional area as constraints. Due to the high computational cost of each optimization cycle, they limited the optimization to just a few design cycles. Kaplan [16] performed an inverse design optimization of a compressor blade to recover a prescribed pressure distribution using finite-difference sensitivities.

2.2.2. Complex Step Method

The Complex Step (CS) method [17, 18] uses the Taylor series expansion of the objective function, similarly to the FD method, but it uses an imaginary step size $i\delta$ instead of the real step size δ used by the FD method.

$$\frac{dJ}{d\vec{\alpha}} = \frac{Im(J(\alpha + i\delta I_i)))}{\delta} + \mathcal{O}(\delta^2) \tag{2.4}$$

where Im represents the imaginary part of the complex cost function, δ is a chosen step size and I_i is the unit vector of of the ith column of an identity matrix. More details of how Eq. 2.4 was derived and how it can be automatically implemented in an existing code can be found in [18]. Since the subtraction term $-J(\alpha)$ of Eq. 2.1 does no longer appear in Eq. 2.4, the round-off errors due the subtraction operation do no longer exist. This allows the designer to select a very small step size (e.g. $1.0e-16$) and compute the approximated gradients to machine accuracy. The relative error for a range of step sizes is compared for the FD and CS methods in [18] and [19], showing that as the step size is reduced the CS method produces more accurate results than the FD method. Therefore, the main advantages of the CS method is that: (1) it is more robust; (2) it does not require repeated calculations to find an optimum step size; (3) it can yield machine accurate gradients for very small step sizes. However, the main disadvantage is that computational cost associated to the gradient computation is therefore proportional to $n+1$.
A CAD integrated adjoint optimization is presented by Mueller [20] where the complex-step method is used to compute the gradients of the grid coordinates with respect to the CAD parameters. Anderson [21], Newman [22] and Vatsa [23] used the complex step method to compute the desrivatives of the lift and drag coefficients with respect to different design variables and compared the accuracy of the gradients with finite-difference results.

2.2.3. Algorithmic differentiation

Algorithmic Differentiation (AD) is a technique for computing analytic derivatives of programs [24]. The idea of AD is to differentiate analytically each elementary mathematical operation performed by the code and accumulate the derivatives by applying the chain rule in an automatic fashion. In this way, it is possible to build up the cost function or derivative of the output with respect to the input variable, which can then be used for an optimization algorithm. AD methods can be implemented by either source to source transformation or operator overloading. The former one consists in transferring the primal evaluation trace into a differentiated evaluation trace. The second is based on operator and function overloading.

One distinguishes between the forward and reverse mode of AD. In forward mode one is interested to know the derivative of the outputs depending on one setting of the input sensitivities, whereas in reverse mode one is interested in the derivative of all the inputs for one single output value. Both modes allows to accurately compute the sensitivities with the best possible machine accuracy. If the number of inputs to the analysis code is greater than the number of outputs, the reverse mode is preferred since it can provide the derivatives at a lower computational cost. However, if the number of outputs is greater than the number of inputs, the forward mode becomes the best choice. For both the forward and reverse modes, it is possible to compute the derivatives in scalar or vector mode. If the desire was to compute the derivatives for all input parameters in the forward scalar mode, it would be necessary to run as many simulations as the number of input parameter. In contrast, the forward vector mode would allow to obtain the gradients with only one run. In this one-run, you would perform n additional computations but the compiler can perform some optimization such that you can get some reduction in CPU cost. The same principle applies to the reverse mode. An introduction to the forward and reverse scalar modes of AD is presented in section 2.6.

The use and development of AD tools has increased significantly during the last decades. Some of the most famous AD tools available are: ADOL-C, Tapenade, AUTODIF, Adept, CoDiPack. There are many interesting applications like Medicine, Physics, Engineering, Chemistry, Mathematics, Bilogy, etc. in which AD tools have helped to improve complex systems. In the area of Aerodynamics, Albring et al. shows in [25, 26] how AD is used to implement an adjoint solver that enables the robust, accurate and efficient computation of gradients for any of the available state solver included in the open-source code CFD software SU2. A methodology for reducing the sonic boom under a supersonic aircraft is presented in [27], which combines hand-coding with AD in the reverse mode using the Tapenade tool.

2.2.4. Analytical differentiation

This method consists in hand-differentiating the main governing equations in order to compute the required gradients. The gradient of the cost function $J(\vec{U}, \vec{X}, \vec{\alpha})$ with respect to the design vector $\vec{\alpha}$ can be computed as follows:

$$\frac{dJ}{d\vec{\alpha}} = \frac{\partial J}{\partial \vec{\alpha}} + \frac{\partial J}{\partial \vec{U}} \frac{d\vec{U}}{d\vec{\alpha}} \tag{2.5}$$

where $\vec{U} = \vec{U}(\vec{X}, \vec{\alpha}) = (\rho, \rho\vec{V}, \rho e)^T$ is the flow variables vector in conservative form, $\vec{X} = \vec{X}(\vec{\alpha})$ are the grid point coordinates and $\vec{\alpha}$ are the design variables. The design variables can be geometrical parameters (e.g., inlet and outlet metal angles, leading edge and trailing edge radius, etc.) and boundary conditions (e.g., rotational speed of a turbine, the coolant inlet temperature of a blade cooling channel, etc.). However, in this thesis only geometrical parameters will be included in the design vector $\vec{\alpha}$. The terms $\partial J/\partial\vec{\alpha}$ and $\partial J/\partial\vec{U}$ appearing in Eq. 2.5 are relatively easy to compute since they can be calculated directly by hand-differentiating the functions. However, the total derivative $d\vec{U}/d\vec{\alpha}$ is the most computationally expensive term to evaluate since it requires solving the non linear or the linearised flow equations for each design variable. In order to better understand how this term is computed, a brief description is described herein. We assume that the steady-state of the Navier-Stokes governing equations are solved:

$$\vec{R}(\vec{X}, \vec{U}) = 0 \tag{2.6}$$

Eq. 2.6 relates the flow solution to the design variables and how the flow solution is altered when the design variables are changed. This relation must also hold for an infinitesimal change to a design variable:

$$\frac{d\vec{R}}{d\vec{\alpha}} = \frac{\partial\vec{R}}{\partial\vec{\alpha}} + \frac{\partial\vec{R}}{\partial\vec{U}}\frac{d\vec{U}}{d\vec{\alpha}} = 0 \tag{2.7}$$

The term $d\vec{U}/d\vec{\alpha}$ can be isolated from Eq. 2.7 as follows

$$\frac{d\vec{U}}{d\vec{\alpha}} = -\left(\frac{\partial\vec{R}}{\partial\vec{U}}\right)^{-1}\frac{\partial\vec{R}}{\partial\vec{\alpha}} \tag{2.8}$$

where the term $\partial\vec{R}/\partial\vec{U}$ is the Jacobian matrix and it is computed only once for each optimization iteration since it is independent of the design variables. However, the term $\partial\vec{R}/\partial\vec{X} \cdot \partial\vec{X}/\partial\vec{\alpha}$ is dependent on the design variables. Therefore, Eq. 2.8 has to be solved for each design variable separately. Finally, the term $d\vec{U}/d\vec{\alpha}$ is introduced in Eq. 2.5 in order to compute the gradient for each design variable. The main disadvantage of this method is that the computational cost is proportional to the number of design variables, similarly to the FD and CS methods.

Rabiega [28] used this method to optimize a centrifugal compressor blade in an Euler flow in order to reduce the mixing losses associated to the non-uniform velocity distribution at the blade outlet boundary. Pandya [29] differentiated analytically the Euler flow equations to compute the necessary gradients to maximize the lift to drag ratio of a 3D supersonic arrow wing configuration whilst mantaining aerodynamic and geometric constraints. Kim et al. [30] also used this method to compute sensitivity derivatives for the Navier-Stokes equations in 2D flow. They differentiated the $k-\omega$ turbulence model and therefore accounted for the variation of the turbulent viscosity with respect to the small perturbations of the design variables.

2.2.5. Adjoint method

The use and development of adjoint methods has been gaining momentum over the last decades in research and industrial design workflows [31–34], as an efficient way of computing gradients for a large number of design variables. Using control theory for optimum shape design of systems governed by elliptic equations was initiated by Pironneau [35]. Jameson pioneered the development and use of the adjoint method in aerodynamic optimization [36][37]. Considering that our design variables are the volumetric grid point coordinates (i.e., $\vec{\alpha} = \vec{X}$), the adjoint method provides the gradients of any cost function J (e.g. entropy generation) with respect to $\vec{X}$ (i.e. $dJ/d\vec{X}$). This gradient can be computed as follows.

$$\frac{dJ}{d\vec{X}} = \frac{\partial J}{\partial\vec{X}} + \frac{\partial J}{\partial\vec{U}}\frac{d\vec{U}}{d\vec{X}} \tag{2.9}$$

The first term $\partial J/\partial\vec{X}$ represents the change in the cost function due to changes in the grid. The second term represents the change in the cost function due to aerodynamic changes. The underlying theory of adjoint methods is well documented in the literature [38–43]. In order to better understand how this term is computed, a brief description of the adjoint technique is described herein. Assuming that the CFD solver has fully converged to steady state, then $\vec{R}(\vec{X}, \vec{U}) = 0$. Hence:

$$\frac{d\vec{R}}{d\vec{X}} = \frac{\partial \vec{R}}{\partial \vec{X}} + \frac{\partial \vec{R}}{\partial \vec{U}} \frac{d\vec{U}}{d\vec{X}} = 0 \tag{2.10}$$

The term $d\vec{U}/d\vec{X}$ can be isolated from Equation 2.10 as follows

$$\frac{d\vec{U}}{d\vec{X}} = -\left(\frac{\partial \vec{R}}{\partial \vec{U}}\right)^{-1} \frac{\partial \vec{R}}{\partial \vec{X}} \tag{2.11}$$

By introducing $d\vec{U}/d\vec{X}$ (Equation 2.11) in Equation 2.9, the $dJ/d\vec{X}$ gradient can be expressed as

$$\frac{dJ}{d\vec{X}} = \frac{\partial J}{\partial \vec{X}} - \left(\frac{\partial J}{\partial \vec{U}} \left(\frac{\partial \vec{R}}{\partial \vec{U}}\right)^{-1}\right) \frac{\partial \vec{R}}{\partial \vec{X}} \tag{2.12}$$

By introducing the adjoint variable $\vec{v}$ as

$$\vec{v}^T = \frac{\partial J}{\partial \vec{U}} \left(\frac{\partial \vec{R}}{\partial \vec{U}}\right)^{-1} \tag{2.13}$$

Equation 2.12 can be reformulated as

$$\frac{dJ}{d\vec{X}} = \frac{\partial J}{\partial \vec{X}} - \vec{v}^T \frac{\partial \vec{R}}{\partial \vec{X}} \tag{2.14}$$

As the adjoint variable $\vec{v}$ is still unknown, the adjoint solver needs to solve the following linear system

$$\left(\frac{\partial \vec{R}}{\partial \vec{U}}\right)^T \vec{v} = \left(\frac{\partial J}{\partial \vec{U}}\right)^T \tag{2.15}$$

where $\partial \vec{R}/\partial \vec{U}$ is the Jacobian matrix of the fluxes of the primal CFD solver. The term $\partial J/\partial \vec{U}$ can be computed by hand-differentiation.

In the literature, Eq. 2.15 is usually referred to as the adjoint equation and the adjoint variable $\vec{v}$ is also addressed as the Langrange multiplier. There are two different approaches in which the adjoint system of equations can be derived: the continuous or the discrete approach. The difference between the two approaches resides in the different treatment of the flow residual throughout the adjoint derivation procedure. The continuous adjoint method would be that one in which one differentiates the flow residual first and then discretizes the resultant expression. Conversevely, if one discretizes spatially the flow residual first and then differentiates the resultant expression, the result is called the discrete adjoint method. Thus, the discrete adjoint equation is obtained by differentiating a set of discrete field equations. For this reason, the resulting equation inherits the type of scheme used to solve the flow equations. The merits of the continuous and discrete adjoint approaches are discussed and compared for different wing configuration shape optimization problems with inviscid and viscous flows by Nadarajah [44]. The continuous and the discrete adjoint formulations are also presented and compared in [45].

With the adjoint method the total cost of the gradient computation for each optimization iteration would be equal to only one flow solution plus one adjoint solution per cost function and aerodynamic constraint. Since the total number of cost functions plus aerodynamic constraints is usually much lower than the number of design variables, the adjoint method allows to reduce the optimization computational cost over other gradient evaluation methods. Unfortunately, this superior advantage comes at the cost of an increase in the complexity of implementation.

Montanelli et al. [46] used the adjoint method to perform a multipoint optimization of a turbine cascade in which the total pressure loss was reduced under a mass flow rate constraint, by solving the Reynolds-averaged Navier-Stokes equations. Papadimitriou et al. [45] presents a method to compute the Hessian matrix of a functional in both discrete and continous forms, which was tested in an inverse design of a 2D duct. Papadimitriou et al. [47] used the continous form of the adjoint method to minimize the total pressure loss of two turbine and compressor cascades. He et al. [48, 49] used the adjoint method to optimize three different multi-stage compressors to improve the isentropic efficiency whilst keeping the total pressure ratio and cascade mass flow range as constraints.

2.3. Literature review on shape parameterization techniques

The ultimate aim of an optimization is to produce a geometry to achieve the design intent within a suitable time frame. These are some of the desirable properties in a good geometry parameterization:

1. It can be easily implemented and automated to reduce the setup time.
2. It can provide a compact set of design variables which will allow the optimizer to navigate faster through the design space.
3. It can maintain a smooth geometry that is practical to manufacture.
4. It allows to have local control on shape changes.
5. It provides a consistent geometry across multiple disciplines if the focus is to do MDO (Multi-Disciplinary Optimization).
6. It integrates well with the product development and life cycle management.
7. It has a direct connection to the CAD systems used for design.
8. It allows to compute efficiently and accurately grid and geometry sensitivities.

A detailed survey of the different shape parameterization techniques are given in [6, 50]. In general, the choice of shape parameterization greatly affects the performance (i.e. computational cost) and the accuracy of the shape optimization [51]. Several studies show the effects of different parameterization representations on shape optimization [51–53]. These techniques can be generally summarized in the following categories:

2.3.1. Node-based approach

The node-based parameterization is sometimes referred to as the "discrete approach" in the open literature [6, 54] and it is based on using the grid point coordinates X, Y, Z as design variables [3–5, 36, 55, 56]. Jameson et al. [36] used the grid points and a gradient smoothing algorithm to successfully improve aircraft wing performance. The benefits of this approach are that 1) it is relatively easy to implement, 2) there is a strong shape local control, 3) it offers a very rich design space since there is no shape limitation, 4) analytical sensitivities are available from the adjoint simulation at a relatively low cost. However, there are some drawbacks that need to be considered: (1) since the geometry changes do not have a limited form it is difficult to maintain smooth geometries and ensure that the geometry surface is C_1 and C_2 continuous; (2) the connection to the CAD geometry is lost and, since CAD is the industry adopted standard for the design of components, an additional step is required to transform the optimal shape defined by grid points back to smooth CAD shape (illustrated in Fig. 2.1), which can take significant time and it is not guaranteed that the final approximated CAD shape will meet the design requirements and constraints [6, 7]; (3) although there are several successful methods that can be used to convert a cloud of grid points to CAD [1, 57], the fitting error can impair the optimality of the shape; (4) there are too many design variables which can slow down the navigation of the optimizer through the design space; (5) the grids used amongst the different disciplines in MDO are inconsistent since they are parameterized separately; (6) for large meshes, the large number of design variables can limit the use of second order numerical optimization methods, such as Newton or quasi-Newton algorithms, due to the high cost associated to matrix operations [51].

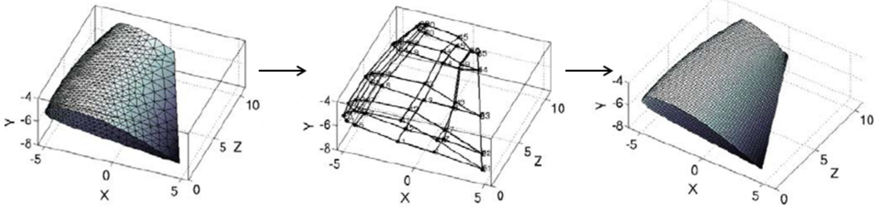


Figure 2.1.: Grid back to CAD conversion: NACA4415 wing surface grid (left) [1]; approximated NURBS control points (middle) [1]; approximated NURBS surface (right) [1].

2.3.2. Free-form deformation

Another way to parameterize the geometry is via the free-from deformation (FFD), which is a powerful tool originally used in the field of soft object animation (SOA) in computer graphics [58] for morphing images and deforming models [59, 60] that has proofed to be very useful for aerodynamic shape optimization applications [32, 61–63]. These algorithms treat the model as rubber or clay that can be bent, twisted, compressed, expanded, tapered, while retaining its topology. The main idea is to enclose the discrete geometry (i.e. defined within the grid) inside a box or any other shape and to link the deformations of the cube to the deformations of the geometry. The geometry grid points are related to the cube points via Bernstein polynomials. Therefore, the design variables are no longer the geometry grid points but the cube points instead. This type of parameterization is widely used in shape optimization for aerodynamics, as it presents the following advantages: (1) parameterization is consistent amongst different disciplines in MDO applications; (2) analytical grid and geometry sensitivities can be computed analytically; (3) complex geometries defined within the grid can be easily parameterized; (4) there is a strong local control; (5) it is possible to control the geometry smoothness; (6) it allows to have a compact set of design variables. There are, however, the following drawbacks to consider: (1) the shape changes are limited; (2) the link to the CAD is lost and hence it inherits the drawbacks (2) and (3) of the node-based approach described in 2.3.1.

2.3.3. Hicks-Henne functions

This is an alternative type of parameterization technique that was initially proposed by Hicks and Henne [64, 65], and has been widely used in aerodynamic shape optimization problems [15, 66–68] in the last decades. Hicks and Henne proposed a compact formulation to parameterize airfoil sections [64, 65]. The main idea is to modify the baseline geometry via adding shape functions (i.e. analytical functions) linearly to the baseline shape (see Eq. 2.16). The shape functions, which are described by $f(x)$ in Eq. 2.16, are often referred as Hicks and Henne sine "bump" functions [51]. These shape functions act as localized deformations and these are determined by the values of the participating coefficients associated to the shape functions. The bumps can either be applied to the discrete mesh or directly to the CAD geometry. In this approach, the coefficients of the shape functions are the design variables. When the coefficients are set to zero in the first iteration, there are no deformations to be applied and hence it gives the baseline geometry. The benefits of this approach are: (1) it allows to have a compact set of design variables; (2) it provides smooth geometries; (3) it supports local shape control; (4) the geometry and grid sensitivities can be computed analytically. Unfortunately, this approach presents the following main drawbacks: (1) the design variables do not have a physical sense for the designers; (2) parameterization is consistent amongst different disciplines in MDO applications; (3) the method is very effective for wing parameterization but it can be quite difficult to generalize for a complex geometry; (4) the connection to the CAD is also lost and hence it inherits the drawbacks (2) and (3) of the node-based approach described in 2.3.1.

$$\begin{aligned} y &= y_{baseline} + \sum_{i=1}^{N} \alpha_i f(x), \\ f(x) &= \left[sin\left(\pi x^{\frac{ln0.5}{lna}} \right) \right]^{b}, \\ \forall x \in [0,1], &\forall a \in [0,1], \forall b \in [2,10]. \end{aligned} \tag{2.16}$$

2.3.4. CAD-based

The geometry parameterization techniques discussed until now present at least one of the following drawacks: they loose the connection to the CAD (i.e, node-based approach and FFD) or they use design variables that don't have any physical sense for the designers (i.e., node-based approach, FFD and Hicks-Henne). These issues can be addressed by using a CAD-based parameterization. The main idea of the CAD-based approach is to use CAD representations, such as B-splines and NURBS, for shape optimization. This can be done either using a commercial CAD system package [13, 14] such as CATIA V5 or Siemens NX or using alternative developed libraries for the CAD geometry representation [20, 69]. A CAD-based parameterization has the main advantage that: (1) CAD-geometry can be defined by a set of engineering based design parameters that are relevant to different areas, including aerodynamic, structure, and system design; (2) parameterization is consistent between the different disciplines of MDO [70]; (3) smoothness can be controlled; (4) the degree of the polynomial curve can be defined by the user ; (5) it allows for local control; (6) the CAD models require a few design variables, which favours the use of Newton or quasi-Newton optimization methods; (7) ease of manipulation of very complex geometries, which reduces the efforts in research development for complex geometries. However, currently it is still a challenge to include the CAD in an optimization loop. Some of the main difficulties are: (1) CAD might be able to provide accurate and smooth geometries but it can also have unacceptable wiggles,free edges and gaps for updating or regenerating a CFD mesh [6]; (2) in gradient-based optimization, it is required to know the sensitivities of the mesh points with respect to the CAD-based design variables (referred throughout this thesis as the grid sensitivities) and this additional sensitivity computation step can be quite challenging (the different techniques to compute such gradients are described in Sec. 2.2); (3) the CAD surface topology can be changed when a design variable is updated and this can cause some problems when updating the CFD mesh or computing the grid sensitivities when computed by finite-differences. An alternative and promising method to compute surface sensitivities with respect to the design variables is by using the design velocity approach [8]. This method can be easily integrated to most industrial optimization workflows (regardless of CAD system) and is immune to the topology and labelling issues that were highlighted by other CAD based optimization processes [71, 72]. A brief description of the velocity design approach is presented in Sec. 2.9.

Despite the challenges of using a CAD system in a gradient-based optimization, researchers have successfully used them and achieved promising results. Two dimensional single- and multi-element airfoil configurations were optimized using a gradient-based Newton-Krylov algorithm and a CAD system, by parametrizing the airfoils with B-spline curves [73] and by using finite differences to compute the gradients. A CAD-based shape parameterization using CATIA V5 was used to perform a gradient-based optimization of an automotive car mirror for noise reduction [74]. In this case, the gradients were computed via the design velocity approach. A CAD-based parameterization using NURBS-patches was used to optimize a one-stage high pressure turbine using a gradient-based optimization algorithm [75]. The coordinates of the NURBS control points were used as design variables and the gradients of the surface points w.r.t. the control points were computed by calculating the shape derivatives of the NURBS surfaces.

In this thesis, the VKI in-house C++ based software CADO [69] is used for generating the geometry. CADO contains CAD libraries and also allows to export the geometry as a STEP file format if necessary.

2.4. B-spline curves

In the present work, the core of the CAD-based geometry modeller employed in this thesis uses B-spline curves to describe the geometry. A short overview regarding the definition of B-spline curves is given herein. A more detailed description on B-spline curves can be found in [76]. A B-spline curve is defined by:

- a 1-dimensional array of control points $\vec{P} = (P_0, P_1, .., P_n)$,
- a knot vector $\vec{u} = (u_0, u_1, ..., u_m)$, where each knot u_i must be defined in the parameter space of the curve, and $u_0 \leq u_1 \leq ... \leq u_m$

Once we define these elements, the B-spline curve is defined by Eq. 2.17, which gives the position along a curve as a function of the parametric length u. Typically, the parametric length u varies from [0.0,1.0].

$$C(u) = \sum_{i=1}^{n} N_{i,p}(u) P_i \tag{2.17}$$

where $n+1$ is the total number of control points, $m+1$ is the total number of knots and $N_{i,p}$ are the B-spline basis functions of degree p (the so-called polynomials). The degree p of the curve must satisfy the following relation:

$$m = n + p + 1 \tag{2.18}$$

A B-spline curve is defined with a series of basis functions. The knot vector determines where in the parameter space of the curve will these basis functions start and stop as the curve is drawn. They are called knots because these are the values in the parameter space where the basis functions are being tied together. For a non-decreasing set of knot vector the B-spline basis functions are recursively defined as:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{2.19}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{2.20}$$

Often, the first and last knot are repeated p+1 times to assure a clamped B-spline, which ensures that the begin and end points of the B-spline corresponds the begin and end control points (i.e. P_0 and P_n). For example, if we wanted to create a clamped B-spline curve of order $p = 4$ with knots at $u = 0.0, 0.25, 0.5, 0.75, 1.0$, we would require the knot vector to be $u = [0.0, 0.0, 0.0, 0.0, 0.0, 0.25, 0.5, 0.75, 1.0, 1.0, 1.0, 1.0, 1.0]$, hence $m = 12$. Figure 2.2 shows the B-spline basis functions for a B-spline of degree $p = 4$ and $m = 12$. The crosses indicate the position of the knots (i.e., the position on which the basis functions start or stop).

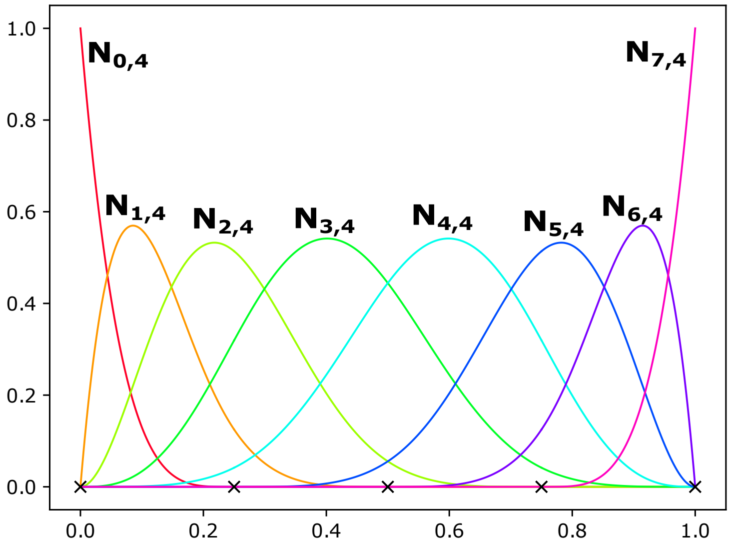


Figure 2.2.: B-spline basis functions for $p = 4$ and $m = 12$

Since the number of knots must satisfy the relation 2.18, in this example, the B-spline curve must have $n = 7$ (i.e. 8 control points in total). Given the position of these 8 control points $\vec{P} = (P_0, .., P_7)$, the B-spline curve can be established. This example B-spline curve is illustrated in Fig. 2.3. The B-spline curve is bounded by a control polygon which is formed by all the control points. Since this is a clamped B-spline, the first and last legs of the control polygon coincide with the tangential direction of the B-spline curve at starting and end points.

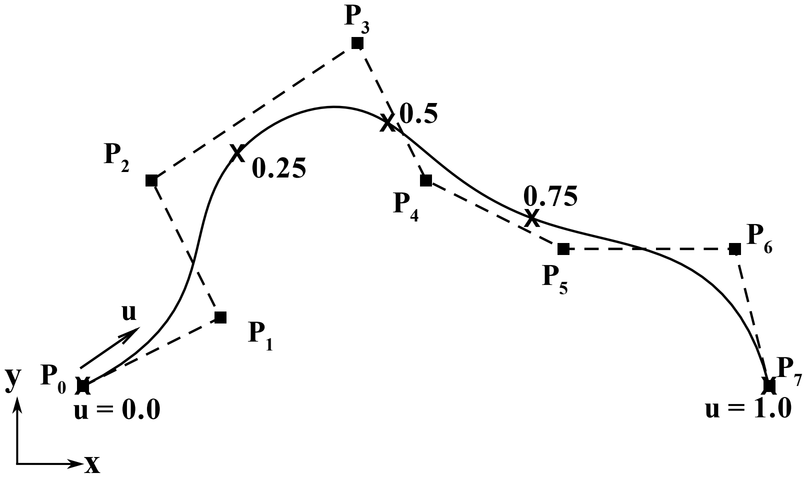


Figure 2.3.: B-spline curve with $p = 7$, $m = 12$ and $n = 7$

One of the main advantages of B-spline curves is that they are excellent for providing local shape control, since a change in the position of a control point P_i does not affect the entire curve but only the local zone of the curve where $u \in [u_i, u_{i+p+1}]$. For example, moving the position of the control point P_1 will only affect the curve where $u \in [u_1, u_6] = [0.0, 0.5]$. This is illustrated by Fig. 2.4 where

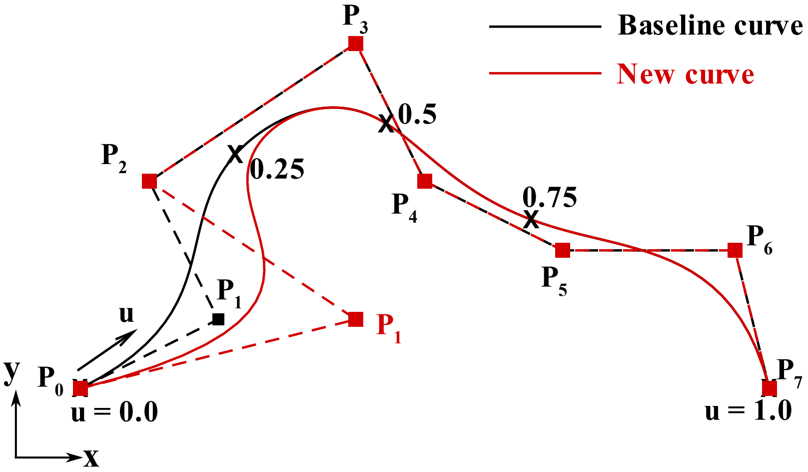


Figure 2.4.: B-spline curves before and after moving control point P_1

the new curve differs from the baseline curve only in this region. However, a shortcoming of the B-spline curve is that they cannot represent accurately implicit conic shapes, such as circles, ellipses or hyperbolas. This issue can be overcome by extending the dimensionality by one by introducing weights, yielding the so-called non-uniform rational B-spline (NURBS). A more detailed description on NURBS can be found in [76]. NURBS are not used within this thesis because there was no need to represent any implicit conic shape.

2.4.1. Knot insertion

Inserting a knot means to add a new knot into the existing knot vector of the B-spline curve without changing its shape. Since the fundamental equality relation shown in 2.18 must be ensured, adding a new knot means that the value of m increases by one and, consequently, either the number of control points or the degree of the curve must also be increased by one. Changing the degree of the curve due to the increase of the number of knots would change the shape of the curve globally and this is not desired. Therefore, inserting a knot means that an additional control point needs to be added. In fact, the old control points are removed and they are replaced by new ones which may be located at new positions. For example, Fig. 2.5 shows the results of inserting a knot at $u = 0.40$ to the baseline B-spline of Fig. 2.3. The new B-spline shape is identical to the baseline but has got 8 control points instead of 7 and the following knot vector: $u = [0.0, 0.0, 0.0, 0.0, 0.0, 0.25, 0.4, 0.5, 0.75, 1.0, 1.0, 1.0, 1.0, 1.0]$.

Since the knot insertion algorithm [76] allows to add additional control points without changing the shape of the curve, this can be used as many times as required during an optimization process to increase the number of control points (i.e., design variables). Therefore, it is possible to start with a small number of control points and end with a large number of them. This can result in a decrease of the number of optimization iterations whilst obtaining a better value for the objective function [77]. The knot insertion algorithm was used in [77, 78] to explore richer design spaces in a CAD-

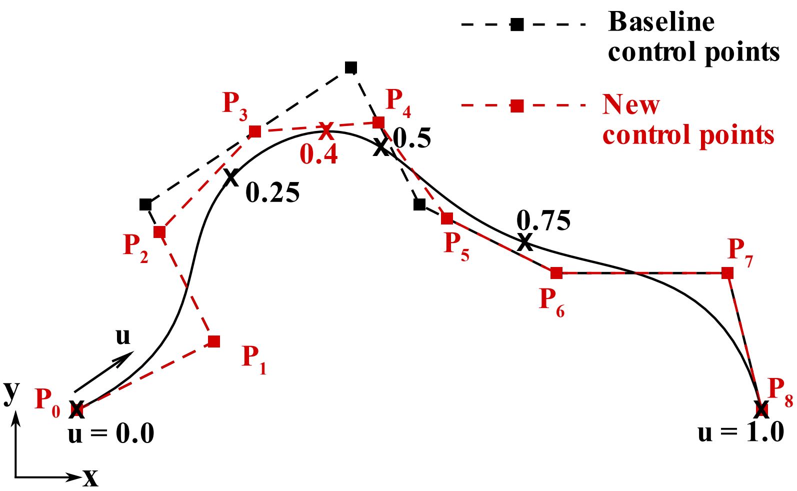


Figure 2.5.: B-spline curve before and after knot insertion at $u = 0.4$

based multilevel optimization framework. In this thesis, the knot insertion algorithm will be used in Sec. 7.2.2 to refine the design spaces in an adaptive shape parameterization optimization context.

2.4.2. Continuity of the curve

The continuity of a curve describes how smooth the curve is. Typically, a curve can be:

- C_0 continuous: when there is no break in the curve.
- C_1 continuous: when there is no kink in the curve because there is no change in the tangent along the curve.
- C_2 continuous: when there is no curvature change along the curve.

Inserting knots maintains the same curve and hence the same continuity. However, it is possible to get a C_n discontinuity in the B-spline curve if the knot vector contains a multiple knot that is repeated $p - n$ times and the control points are moved after knot insertion.

For aerodynamic applications, such as aircraft wing aerofoils or turbine profiles, special care needs to be taken for the geometry discontinuities. Typically, C2 continuity in the geometry is often desired on the suction side because a discontinuity on the curvature can cause strong velocity peaks.

2.5. Grid generation

Computational Fluid Dynamics (CFD) requires the spatial region of interest to be discretized. The spatial discretization process is typically referred to as grid generation and it requires a significant portion of the labour for each analysis. The rest of the labour is invested in the numerical solution and the post-processing of the results. The grid generation methodologies can be divided into two main categories: unstructured and structured. The unstructured grids consist of quadrilateral or triangular elements in 2D and hexahedral, tetrahedral, triangular prisms or pyramid elements in 3D. In contrast, the structured grids consist of quadrilateral and hexahedral elements in 2D and 3D respectively. The structured meshes are used in this thesis to provide fine control over the final quality of the mesh. Nevertheless, the work presented in this thesis could be equally applied to unstructured grids.

In the context of optimization, the grid generation process must be automatic (where the user expertise is not required to manually repair the geometry model or the grid itself) and must ensure high quality grids. Often the quality of the CFD solutions is dependent on the accuracy of the numerical discretization and the quality of the computational grid. The quality of structured grids is often described in terms of:

- smoothness: the gradient of grid cell size and shape should be minimized.
- orthogonality: grid lines should be as orthogonal to each other as possible.
- clustering: the grid points should be concentrated in those areas where high solution gradients are expected.

2.5.1. Structured grids

The quadrilateral cells of a two dimensional structured grid are arranged in a $I \times J$ array, whereas the hexahedral cells of a three dimensional structured grid are arranged in an $I \times J \times K$ array. Typically, for complex geometries the structured grids are implemented using a multi-block topology, in which the domain is subdivided into smaller domains called blocks.

In this thesis, the generation of structured grids consists in using the following methods:

- transfinite interpolation (TFI): to generate an initial mesh.
- elliptic partial differential equations (PDEs): to smooth the initial mesh in an iterative manner to produce a high quality grid.

These methods will be described in more detail below.

2.5.2. Transfinite interpolation

Imagine that the interior domain of the two dimensional square block shown in Fig. 2.6 is to be meshed using TFI. The block has got four edges or boundaries: the bottom ($\eta = 0$), the top ($\eta = 1$), the left ($\xi = 0$) and the right ($\xi = 1$). The first step is to create a distribution of points on these four boundaries. After this, the internal grid of the block is initialized using Transfinite Interpolation (TFI) equations 2.21 and 2.22 [79]. These equations allow to compute the internal grid given the coordinates of the boundaries for $\eta = 0, 1$ and $\xi = 0, 1$ (see Fig. 2.6).

$$\begin{aligned} X(\xi,\eta) &= (1-u)x(0,\eta) + ux(1,\eta) + (1-v)x(\xi,0) + vx(\xi,1) \\ &\quad - (1-u)(1-v)x(0,0) - u(1-s)x(1,0) - (1-u)vx(0,1) - uvx(1,1) \end{aligned} \tag{2.21}$$

$$\begin{aligned} Y(\xi,\eta) = {} & (1-u)y(0,\eta) + uy(1,\eta) + (1-v)y(\xi,0) + vy(\xi,1) \\ & - (1-u)(1-v)y(0,0) - u(1-s)y(1,0) - (1-u)vy(0,1) - uvy(1,1) \end{aligned} \tag{2.22}$$

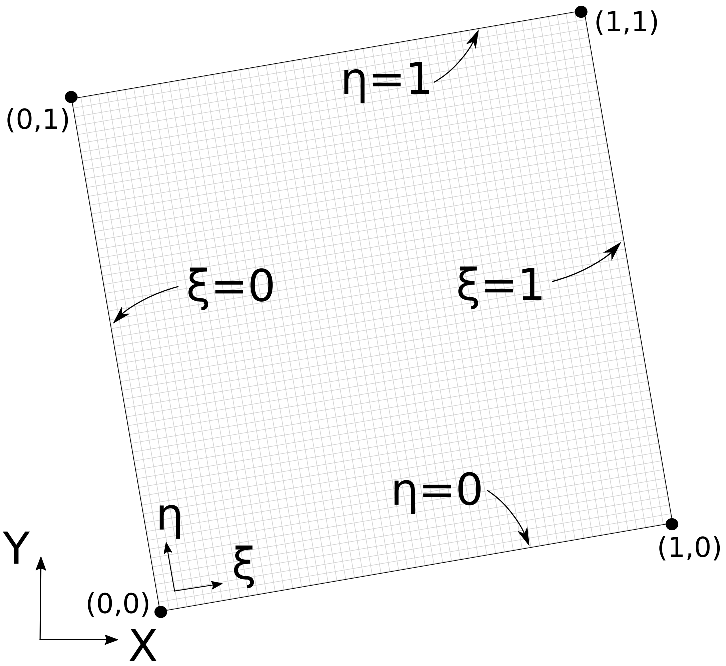


Figure 2.6.: Two dimensional block.

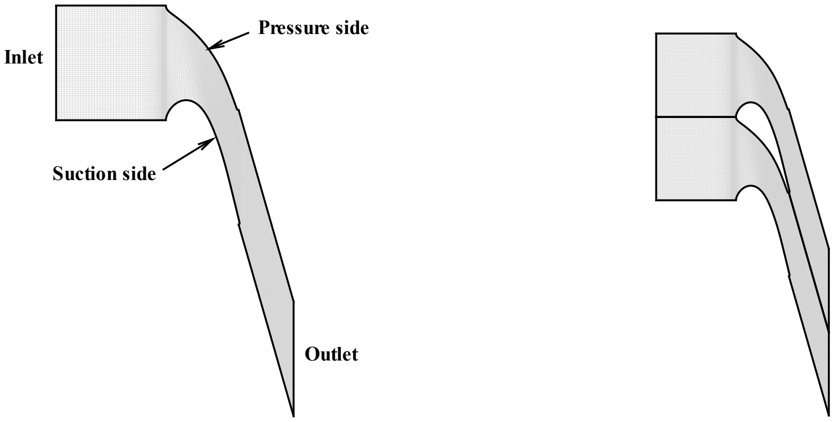
The parameters u and v from Eqns. 2.23 and 2.24 are blending formulas proposed by Soni [80] for the normalised arc-length control functions $s_1(\xi)$, $s_2(\xi)$, $t_1(\eta)$, $t_2(\eta)$ along the boundary edges. The normalised arc-length control functions intuitively express the position of each grid point along the edge as a percentage of the total length of the edge. In the present work, $s_1(\xi)$ and $s_2(\xi)$ are defined along the boundary edges spanning between $t_1(\eta = 0)$ and $t_2(\eta = 1)$ respectively, whereas $t_1(\eta)$ and $t_2(\eta)$ are defined along the boundary edges spanning between $s_1(\xi = 0)$ and $s_2(\xi = 1)$.

$$u(\xi,\eta) = \frac{(1-t_1(\eta))s_1(\xi) + t_1(\eta)s_2(\xi)}{1-(s_2(\xi)-s_1(\xi))(t_2(\eta)-t_1(\eta))} \tag{2.23}$$

$$v(\xi,\eta) = \frac{(1-s_1(\xi))t_1(\eta) + s_1(\xi)t_2(\eta)}{1-(t_2(\eta)-t_1(\eta))(s_2(\xi)-s_1(\xi))} \tag{2.24}$$

To create a single-block CFD mesh for a more complex geometry such as the highly staggered blade shown in Fig. 2.7b using TFI, the profile needs to be split into suction and pressure side in order to create the top and bottom boundaries of the single block (see Fig. 2.7a).

The speed and the ease of implementation are the two main strengths of the TFI methods. However, the TFI method does not guarantee that the grid lines will not intersect, fold or that the corners of the sides of the geometry will not be propagated to the interior of the mesh. For exam-



(a) One single block. **(b)** Two repeated single blocks.

Figure 2.7.: Single block mesh generated by TFI for a highly staggered blade.

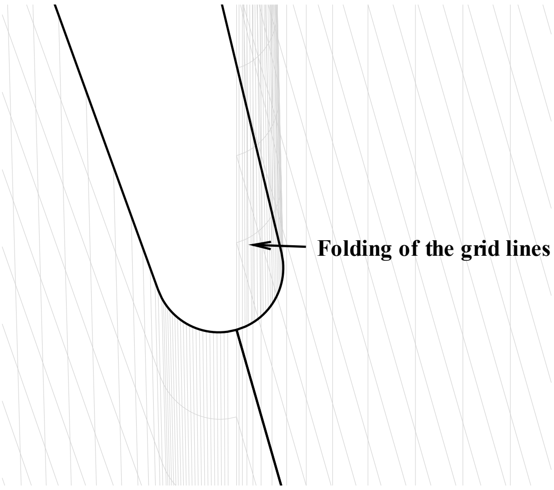


Figure 2.8.: Folding of grid lines around the trailing edge region.

ple, Fig. 2.8 shows that the grid lines are intersecting around the trailing edge region. Therefore, the quality of a mesh generated by TFI in terms of orthogonality, smoothness and clustering may be less than optimal and must be improved by, for example, the use of elliptic PDEs.

2.5.3. Elliptic PDEs

An elliptic PDE method is a boundary problem that involves solving the Poisson equations with control functions (right hand side terms), as shown in Eqns. 2.25-2.26. It was pioneered by Thompson et. al. [81] to generate structured grids. The control functions provide control over the grid's orthogonality, clustering and smoothness, since they have an influence on the grid points on the interior of the domain or near its boundaries. The main strength of the elliptic PDE methods is that it can generate high quality grids. An iterative procedure is required to solve the elliptic PDEs and it is repeated until certain convergence criteria is met.

$$\xi_{xx} + \xi_{yy} = P(\xi, \eta) \tag{2.25}$$

$$\eta_{xx} + \eta_{yy} = Q(\xi, \eta) \tag{2.26}$$

The Poisson equations shown in Eqns. 2.25-2.26 are formulated in the physical space XY. In the physical space, the grid is case dependent, curvilinear and typically body-fitted. In order to solve the Poisson equations, it is required to transform the grid point coordinates from the physical space to the computational space $\xi\eta$, where the grid is uniformly spaced and orthogonal. The inverse transformed system of equations then becomes [79]:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -I^2(Px_{\xi} + Qx_{\eta}) \tag{2.27}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -I^2(Py_{\xi} + Qy_{\eta}) \tag{2.28}$$

In Eqns. 2.27-2.28, the greek symbols α, β, γ represent the scale metric factors of the coordinate transformation and they are defined by $\alpha = x_{\eta}^2 + y_{\eta}^2$, $\beta = x_{\xi}x_{\eta} + y_{\xi}y_{\eta}$, $\gamma = x_{\xi}^2 + y_{\xi}^2$. The inverse of the determinant is given by $I = \frac{1}{D} = x_{\xi}y_{\eta} - x_{\eta}y_{\xi}$, which is often referred to as the Jacobian in grid generation.

2.5.3.1. Solution procedure

Suppose we want to solve the elliptic equations shown in Eqns. 2.27-2.28 in the stencil of grid nodes shown in Fig. 2.9. First, we set the distance between two adjacent grid points to be $\delta\xi = \delta\eta = 1$ in the computational space $\xi\eta$ and approximate the partial derivatives using Finite Differences (Eqns. 2.29-2.32). This allows us to convert the Eqns. 2.27-2.28 into the discretized system of Eqns. shown in 2.33 - 2.34, from which we can isolate the x_{ij} and y_{ij} terms (Eqns. 2.35 - 2.36). Finally, the grid coordinates are updated iteratively over the entire domain for $i = 0, .., I - 1$ and $j = 0, .., J - 1$ of the bock by means of a standard GS-(S)SOR (Gauss-Seidel Symmetric Successive Over-relaxation) approach (Eq. 2.37).

The approach to solve the elliptical equations in this work for a multiblock structured mesh is based on the approach presented here to solve the elliptical equations for the stencil shown in Fig. 2.9. The grid nodes of the interior domain of each block are updated first and the grid nodes at the interfaces of the blocks are updated next. The interface grid point coordinates are set to be the average of its surrounding grid node coordinates.

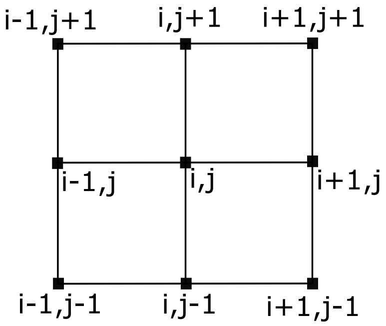


Figure 2.9.: Stencil of grid points.

$$
\begin{aligned}
x_\xi &= \frac{x_{i+1,j} - x_{i-1,j}}{2} \\
y_\xi &= \frac{y_{i+1,j} - y_{i-1,j}}{2}
\end{aligned}
\tag{2.29}
$$

$$
\begin{aligned}
x_\eta &= \frac{x_{i,j+1} - x_{i,j-1}}{2} \\
y_\eta &= \frac{y_{i,j+1} - y_{i,j-1}}{2}
\end{aligned}
\tag{2.30}
$$

$$
\begin{aligned}
x_{\xi\xi} &= x_{i+1,j} + x_{i-1,j} - 2x_{i,j} \\
y_{\xi\xi} &= y_{i+1,j} + y_{i-1,j} - 2y_{i,j}
\end{aligned}
\tag{2.31}
$$

$$
\begin{aligned}
x_{\eta\eta} &= x_{i,j+1} + x_{i,j-1} - 2x_{i,j} \\
y_{\eta\eta} &= y_{i,j+1} + y_{i,j-1} - 2y_{i,j}
\end{aligned}
\tag{2.32}
$$

$$
\begin{aligned}
\alpha(x_{i+1,j} + x_{i-1,j} - 2x_{i,j}) - \beta(\frac{x_{i+1,j+1} - x_{i+1,j-1}}{2} - \frac{x_{i-1,j+1} - x_{i-1,j-1}}{2}) \\
+\gamma(x_{i,j+1} + x_{i,j-1} - 2x_{i,j}) + I^2(P\frac{x_{i+1,j} - x_{i-1,j}}{2} + Q\frac{x_{i,j+1} - x_{i,j-1}}{2}) = 0
\end{aligned}
\tag{2.33}
$$

$$
\begin{aligned}
\alpha(y_{i+1,j} + y_{i-1,j} - 2y_{i,j}) - \beta(\frac{y_{i+1,j+1} - y_{i+1,j-1}}{2} - \frac{y_{i-1,j+1} - y_{i-1,j-1}}{2}) \\
+\gamma(y_{i,j+1} + y_{i,j-1} - 2y_{i,j}) + I^2(P\frac{y_{i+1,j} - y_{i-1,j}}{2} + Q\frac{y_{i,j+1} - y_{i,j-1}}{2}) = 0
\end{aligned}
\tag{2.34}
$$

$$\begin{aligned} x_{i,j} = \frac{1}{2(\alpha+\gamma)}\Big(&\alpha(x_{i+1,j} + x_{i-1,j}) \\ &+\gamma(x_{i,j+1} + x_{i,j-1}) \\ &-\frac{\beta}{2}(x_{i+1,j+1} - x_{i+1,j-1} - x_{i-1,j+1} + x_{i-1,j-1}) \\ &+I^2(P\frac{x_{i+1,j} - x_{i-1,j}}{2} + Q\frac{x_{i,j+1} - x_{i,j-1}}{2})\Big) \end{aligned} \tag{2.35}$$

$$\begin{aligned} y_{i,j} = \frac{1}{2(\alpha+\gamma)}\Big(&\alpha(y_{i+1,j} + y_{i-1,j}) \\ &+\gamma(y_{i,j+1} + y_{i,j-1}) \\ &-\frac{\beta}{2}(y_{i+1,j+1} - y_{i+1,j-1} - y_{i-1,j+1} + y_{i-1,j-1}) \\ &+I^2(P\frac{y_{i+1,j} - y_{i-1,j}}{2} + Q\frac{y_{i,j+1} - y_{i,j-1}}{2})\Big) \end{aligned} \tag{2.36}$$

$$\begin{aligned} x_{i,j}^{new} &= (1-r)x_{i,j}^{old} + rx_{i,j} \\ y_{i,j}^{new} &= (1-r)y_{i,j}^{old} + ry_{i,j} \end{aligned} \tag{2.37}$$

2.5.3.2. Control functions

Grid behaviour control is achieved through the introduction of the forcing function terms P and Q. Two different methods have been used in this thesis to set the source terms: the Laplace and the Steger and Sorenson [82]. The properties of each method are illustrated by showing their effect on the grid when applied to the initial mesh that was generated by TFI shown in Fig. 2.10.

The Laplace method consists in setting the source terms to zero. This results in a grid with smooth variation of cell size. This method is typically used in those regions of the domain where it is not required to have orthogonality and clustering of the grid points near the boundary.

The Steger and Sorenson method is an iterative scheme that adjusts the boundary forcing functions in order to meet a prescribed angle and spacing constraint at the boundary. The source terms are then interpolated into the domain based on their values at the boundaries. This method is typically used in the regions of the domain close to the profile, where a certain first cell thickness and orthogonality have to be imposed.

2.6. Algorithmic differentiation techniques

Algorithmic differentiation (AD), also known as computational differentiation or automatic differentiation, is a well known method that applies the chain rule to each operation in the computer

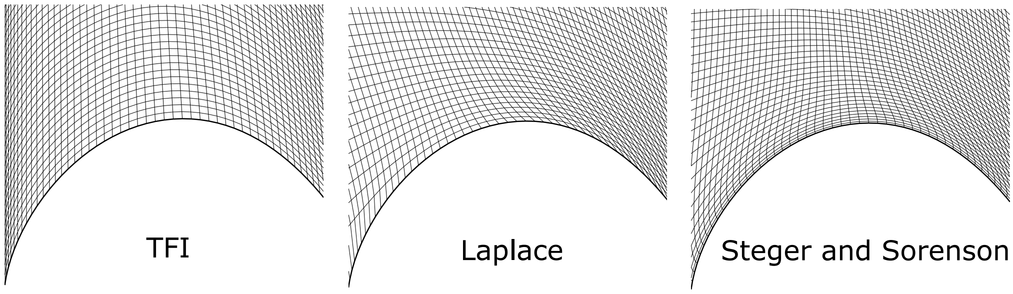


Figure 2.10.: Mesh before smoothing (left) and after smoothing with Laplace (middle) and Steger and Sorenson (right) control functions.

program flow systematically and obtains accurate gradients to machine accuracy. An introduction to the AD techniques is presented here, to show how the methodology works and how they are typically implemented. For a more detailed description of the different techniques for evaluating derivatives using AD the reader is referred to [24].

2.6.1. Forward and reverse modes

Consider a computer program that takes a given input $\vec{x}$ and does certain calculations to compute intermediate variables and a vector of outputs $\vec{J}$:

$$\begin{aligned}
&\vec{x} = \big(x_1, x_2, ..., x_n\big)\\
&\downarrow\\
&\vec{t} = \big(t_1, t_2, ..., t_p\big),\ p > m + n\\
&\downarrow\\
&\vec{J} = \big(J_1, J_2, ..., J_m\big)
\end{aligned} \tag{2.38}$$

This is also represented in Fig. 2.11. Each square box can be seen as a variable, or even a line of code or function in the program, depending on the scale considered but the principle remains the same. The value of a square box depends on each of its inputs, hence the partial derivative of the node with respect to its inputs is associated to the down-ward pointing edges of the graph. The intermediate variables $\vec{t}$ are related through a series of elementary functions f_k which can be either unary (e.g., operations like pow(), -(), sin(), cos(), etc.),

$$t_k = f_k\big(t_i\big) \tag{2.39}$$

or binary (e.g., operations like +, -, ., /),

$$t_k = f_k\big(t_i, t_j\big) \tag{2.40}$$

Now we want to make use of AD to compute these gradients. The derivatives given by the chain rule can be propagated in two directions: in forward (forward mode) or backward (reverse mode), as shown in Fig. 2.11.

In the forward mode, the derivatives are propagated throughout the computation using the chain rule. For each elementary operation the derivative $\partial t_k/\partial x_h$ can be computed as (e.g., for the binary operation)

$$\frac{\partial t_k}{\partial x_h} = \frac{\partial f_k}{\partial t_i}\frac{\partial t_i}{\partial x_h} + \frac{\partial f_k}{\partial t_j}\frac{\partial t_j}{\partial x_h} \tag{2.41}$$

and this requires that at the start of the computation the initial values of the derivatives $(\dot{x}_1, \dot{x}_2, ..., \dot{x}_n)$ need to be set. For example, if we require computing the derivative of the output J of the sample program in Fig. 2.11 with respect to the input X_1, we have to set the derivative $\dot{X}_1 = \partial X_1/\partial X_1$ to 1.0 and $\dot{X}_2 = \dot{X}_3 = 0.0$. In this example, to compute the derivative in the direction of each of the three inputs we need to run the tangent linear code three times. Finally, at the end of the derivative values

$$\dot{J}_s = \sum_{s=1}^{n} \frac{\partial J_s}{\partial x_h}\dot{x}_h \tag{2.42}$$

are known.

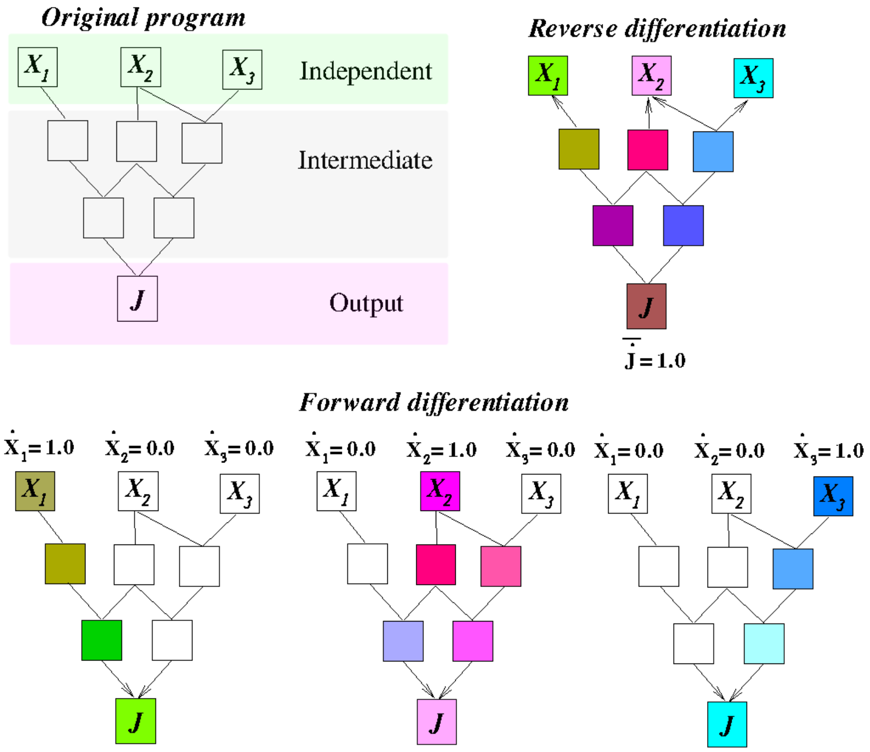


Figure 2.11.: Computational graphs for primal (top left), forward (bottom row) and reverse mode (top right).

A sample computer program is described below to show the practical implementation for the forward mode. Consider a program that performs the following series of operations and evaluates the function $J = f(x_1, x_2, x_3)$ at $(x_1, x_2, x_3) = \left(\frac{\pi}{2}, \frac{-\pi}{4}, \frac{3\pi}{4}\right)$, as shown in Tab. 2.1.

In the case where it is only required to know the derivative of the output w.r.t the first input $(\partial J/\partial x_1)$, then $\dot{x}_1 = 1.0, \dot{x}_2 = 0.0, \dot{x}_3 = 0.0$ needs to be given as an additional input to the function.

Table 2.1.: Original computer program

function $J = f(x_1, x_2, x_3)$	
$t_1 = x_1$	$= \frac{\pi}{2}$
$t_2 = x_2$	$= -\frac{\pi}{4}$
$t_3 = x_3$	$= \frac{3\pi}{4}$
$t_4 = t_1 + t_2$	$= \frac{\pi}{4}$
$t_5 = t_4 + t_3$	$= \pi$
$t_6 = \sin(t_5)$	$= 0.0$
$t_7 = t_1 \cdot t_6$	$= 0.0$
$J = t_7$	$= 0.0$
end	

The notation $\dot{t_i} = \frac{\partial t_i}{\partial x_1}$ is introduced for simplicity. At the start of the computation, $\dot{t_1} = \frac{\partial t_1}{\partial x_1} = 1.0$, $\dot{t_2} = \frac{\partial t_2}{\partial x_1} = 0.0$ and $\dot{t_3} = \frac{\partial t_3}{\partial x_1} = 0.0$, which yields the results shown in Tab. 2.2.

Table 2.2.: Forward mode differentiation of the computer program

function $(J, \dot{J}) = f(x_1, x_2, x_3, \dot{x_1}, \dot{x_2}, \dot{x_3})$	
$t_1 = x_1$	$= \frac{\pi}{2}$
$\dot{t_1} = \dot{x_1}$	$= 1.0$
$t_2 = x_2$	$= -\frac{\pi}{4}$
$\dot{t_2} = \dot{x_2}$	$= 0.0$
$t_3 = x_3$	$= \frac{3\pi}{4}$
$\dot{t_3} = \dot{x_3}$	$= 0.0$
$t_4 = t_1 + t_2$	$= \frac{\pi}{4}$
$\dot{t_4} = \dot{t_1} + \dot{t_2}$	$= 1.0$
$t_5 = t_4 + t_3$	$= \pi$
$\dot{t_5} = \dot{t_4} + \dot{t_3}$	$= 1.0$
$t_6 = \sin(t_5)$	$= 0.0$
$\dot{t_6} = \cos(t_5)$	$= -1.0$
$t_7 = t_1 \cdot t_6$	$= 0.0$
$\dot{t_7} = \dot{t_1} t_6 + t_1 \dot{t_6}$	$= -\frac{\pi}{2}$
$J = t_7$	$= 0.0$
$\dot{J} = \dot{t_7}$	$= -\frac{\pi}{2}$
end	

The same code in Tab. 2.2 can be used to evaluate $\partial J / \partial x_2$ and $\partial J / \partial x_3$ if we set $\dot{x_1} = 0.0, \dot{x_2} = 1.0, \dot{x_3} = 0.0$ and $\dot{x_1} = 0.0, \dot{x_2} = 0.0, \dot{x_3} = 1.0$ respectively.

The second mode of operation of AD is the reverse mode. In reverse mode, the derivatives $\partial J_s / \partial t_k$ for all intermediate variables are computed backwards or in reverse order. For example, for the elementary step in Eq. 2.40 the derivatives are propagated as:

$$\frac{\partial J}{\partial t_i} = \frac{\partial J}{\partial t_k} \frac{\partial t_k}{\partial t_i} \tag{2.43}$$

and

$$\frac{\partial J}{\partial t_j} = \frac{\partial J}{\partial t_k}\frac{\partial t_k}{\partial t_j} \tag{2.44}$$

At the start of the computation in reverse mode, the derivative of the output $\dot{J}_1, \dot{J}_2, ..., \dot{J}_m$ needs to be given additionally to the regular input values $\dot{x_1}, \dot{x_2}, ..., \dot{x_n}$. Seeding the derivative of one of the outputs to (1.0) and computing the chain rule for each elementary operation going backwards allows us to compute the derivatives of the single output with respect to all the intermediate variables and the inputs in a single reverse-mode computation. Finally, at the end of the computation the values of

$$\dot{x_k} = \sum_{s=1}^{m} \frac{\partial J_s}{\partial x_k}\dot{J_s} \tag{2.45}$$

are known.

In the case of one output only ($m = 1$) then the computation of $\dot{x_k}$ is given by $\frac{\partial J_1}{\partial x_k}\dot{y_1}$. The same sample program (Tab. 2.1) is considered to show a practical implementation for the reverse mode. A new variable called adjoint variable is introduced:

$$\overline{\dot{t_i}} = \frac{\partial J}{\partial t_i} \tag{2.46}$$

By definition, $\overline{\dot{J}} = 1.0$ at the start. The computation in reverse mode is shown in Tab. 2.3.

In the reverse mode, additional operations are needed for each line of code. For example, for the operation $t_7 = t_1 \cdot t_6$, two adjoint variables $(\overline{\dot{t_1}}, \overline{\dot{t_6}})$ need to be computed (because it is a binary operation):

$$\overline{\dot{t_1}} = \frac{\partial J}{\partial t_1} = \frac{\partial J}{\partial t_7}\frac{\partial t_7}{\partial t_1} = \frac{\partial J}{\partial t_7}\frac{\partial\big(t_1 \cdot t_6\big)}{\partial t_1} = \overline{\dot{J}} \cdot t_6 \tag{2.47}$$

$$\overline{\dot{t_6}} = \frac{\partial J}{\partial t_6} = \frac{\partial J}{\partial t_7}\frac{\partial t_7}{\partial t_6} = \frac{\partial J}{\partial t_7}\frac{\partial\big(t_1 \cdot t_6\big)}{\partial t_6} = \overline{\dot{J}} \cdot t_1 \tag{2.48}$$

whereas for the operation $t_6 = \sin(t_5)$ only one adjoint variable $(\overline{\dot{t_5}})$ needs to be computed, since it is a unary operation:

$$\overline{\dot{t_5}} = \frac{\partial J}{\partial t_5} = \frac{\partial J}{\partial t_6}\frac{\partial t_6}{\partial t_5} = \frac{\partial J}{\partial t_6}\frac{\partial\big(\sin(t_5)\big)}{\partial t_5} = \overline{\dot{t_6}} \cdot \cos(t_5) \tag{2.49}$$

For the evaluation of the derivatives in reverse mode, note that the temporary values of the intermediate variables t_i need to remain stored. This results in a large memory requirement for the reverse mode.

The main advantage of using AD for derivative computation is that the gradient can be computed to machine accuracy. Since the gradient computation speed is dependent on the number of design variables for the forward mode of AD, the forward mode is preferred when there are less number of inputs than outputs. If the number of inputs is greater than the number of outputs, the reverse mode is preferred, however with the drawback of a larger memory requirement.

Table 2.3.: Reverse mode differentiation of the computer program

function $(J, \overline{\dot{x}_1}, \overline{\dot{x}_2}, \overline{\dot{x}_3}) = f(x_1, x_2, x_3, \overline{\dot{J}})$	
$t_1 = x_1$	$= \frac{\pi}{2}$
$t_2 = x_2$	$= -\frac{\pi}{4}$
$t_3 = x_3$	$= \frac{3\pi}{4}$
$t_4 = t_1 + t_2$	$= \frac{\pi}{4}$
$t_5 = t_4 + t_3$	$= \pi$
$t_6 = \sin(t_5)$	$= 0.0$
$t_7 = t_1 \cdot t_6$	$= 0.0$
$J = t_7$	$= 0.0$
$\overline{\dot{t}_7} = \overline{\dot{J}}$	$= 1.0$
$\overline{\dot{t}_1} = \frac{\partial J}{t_1} = \frac{\partial J}{\partial t_7}\frac{\partial t_7}{\partial t_1} = t_6$	$= 0.0$
$\overline{\dot{t}_6} = \frac{\partial J}{t_6} = \frac{\partial J}{\partial t_7}\frac{\partial t_7}{\partial t_6} = t_1$	$= \frac{\pi}{2}$
$\overline{\dot{t}_5} = \frac{\partial J}{t_5} = \frac{\partial J}{\partial t_6}\frac{\partial t_6}{\partial t_5} = \overline{\dot{t}_6} \cdot \cos(t_5)$	$= -\frac{\pi}{2}$
$\overline{\dot{t}_4} = \frac{\partial J}{t_4} = \frac{\partial J}{\partial t_5}\frac{\partial t_5}{\partial t_4} = \overline{\dot{t}_5}$	$= -\frac{\pi}{2}$
$\overline{\dot{t}_3} = \frac{\partial J}{t_3} = \frac{\partial J}{\partial t_5}\frac{\partial t_5}{\partial t_3} = \overline{\dot{t}_5}$	$= -\frac{\pi}{2}$
$\overline{\dot{t}_1} = \frac{\partial J}{t_1} = \frac{\partial J}{\partial t_4}\frac{\partial t_4}{\partial t_1} + \overline{\dot{t}_1} = \overline{\dot{t}_4} + \overline{\dot{t}_1}$	$= -\frac{\pi}{2}$
$\overline{\dot{t}_2} = \frac{\partial J}{t_2} = \frac{\partial J}{\partial t_4}\frac{\partial t_4}{\partial t_2} = \overline{\dot{t}_4}$	$= -\frac{\pi}{2}$
$\overline{\dot{x}_3} = \frac{\partial J}{x_3} = \overline{\dot{t}_3} = x_1 \cdot \cos(x_1 + x_2 + x_3)$	$= -\frac{\pi}{2}$
$\overline{\dot{x}_2} = \frac{\partial J}{x_2} = \overline{\dot{t}_2} = x_1 \cdot \cos(x_1 + x_2 + x_3)$	$= -\frac{\pi}{2}$
$\overline{\dot{x}_1} = \frac{\partial J}{x_1} = \overline{\dot{t}_1} = x_1 \cdot \cos(x_1 + x_2 + x_3) + \sin(x_1 + x_2 + x_3)$	$= -\frac{\pi}{2}$
end	

2.6.2. AD implementation

The implementation of AD can be based on operator overloading (ADOL-C [83] for c/c++, ADF [84] for fortran) or source code transformation (ADIFOR [85] for fortran, ADIC [86] for c, TAPENADE [87] for c and fortran).

2.6.2.1. AD via operator overloading

Most modern programming languages like c++ offer operator overloading where it is possible to define specific datatypes and redefine the meaning of intrinsic operators such as addition, substraction, multiplication, etc. This can be used to implement the forward-differentiation of AD in a very simple way, following the steps below:

1. Creating a new data type called Real to represent floating point numbers, and overwriting it with a new data type that contains both the value and its derivative.
```
#if defined (ADForwardMode)
typedef adouble Real;
# else
typedef double Real;
# endif
```

When the code is to be executed using the forward-differentiation of AD, the user defined data type Real represents an adouble array, which contains two floating points: the value and its derivative (adouble = [value, derivative]).

2. Redefine the instrinsic operations using operator overloading. For example, the multiplication operator should be overloaded with the following function:
   ```
   multiplication(adouble a, adouble b){
       return [ a.value()·b.value(), a.derivative()·b.value() + a.value()·b.derivative() ] }
   ```
   Generally, the intrinsic operation functions are already redefined inside the AD classes.
3. Replace all the standard invocations of float or double by the new datatype Real.
4. Seed the input derivatives of the design variables.
5. Compile and run.

The use of AD in reverse mode via operator overloading is more complicated because, instead of computing the derivative at the same time than the primal value, we need to store the values of the primal and the history of operations on a tape. The overloaded multiplication function in reverse mode will become e.g.

```
multiplication(adouble a, adouble b){
    push2tape(mult, a, b);
    return [ a.value()·b.value() ] }
```

The reverse-differentiation of AD is implemented following the steps below:

1. Start recording the tape.
2. Mark the independent variables or inputs.
3. Mark the outputs.
4. End the recoding of the tape.
5. Seed the derivative of the output to 1.0.
6. Evaluate the tape, in order to accumulate the derivatives going backwards.

One disadvantage of this approach is that the tape size can become prohibitive for large industrial cases. However, the user can decide which part of the codes need to be recorded in the tape or not. For example, one might decide to stop taping for specific routines like linear solvers and then restart taping just afterwards. In such cases, deep code-insight and user manipulation is required to make sure that the final derivatives also take into account the operations that were not taped.

In this thesis, both forward and reverse modes of AD will be used to compute gradient information of a computer program. The forward and reverse mode of AD will be used mainly to differentiate the CAD kernel and grid generator in chapter 3. The resultant derivatives will be used in order to:

(1) compute the performance gradients required by the optimizer in chapters 4 - 7; (2) to compute the novel parametric effectiveness indicator defined in chapter 6 and used for adaptive shape optimization in chapter 7.

2.6.2.2. AD via Source Transformation

Although this technique is not used in this thesis, it will be briefly explained for completion. Source code transformation consists in using a transformation tool to generate a differentiated code at compilation time. This has the advantage that the code can be optimized by the compiler and the memory requirement can be kept at a small multiple of the primal code [88], which allows the application of AD to very large industrial codes. The disadvantage of source code transformation is that the implementation of the transformation tool is extremely complex and there are no tools available at the time for recent languages such as c++ with extensive pointer use. If at compile time it is not clear what value a pointer is pointed to, then the transformation tool is not able to know where to add the derivative to. Tools for Fortran and for c are available (such as TAPENADE [87], by Inria).

2.7. CFD and Adjoint solvers

All the numerical simulations in this thesis were performed with a Computer Aided Design and Optimization (CADO) [69] tool that contains both flow and adjoint solvers to manage the flow analysis and flow sensitivity aspects. The flow solver solves compressible RANS equations using a cell-centered finite-volume method on multiblock structured grids. The three dimensional compressible RANS equations are solved with an implicit time integration scheme accelerated by local time-stepping and multigrid. The fluid is considered as a calorically perfect gas and the eddy-viscosity hypothesis is used to account for the effect of turbulence. Convective fluxes are computed using Roe's approximate Riemann solver [89] with a MUSCL-type reconstruction [90] of primitive variables to attain second order accuracy. Oscillations near shocks are suppressed by a van-Albada type limiter [91]. The numerical dissipation of the scheme is controlled by the entropy correction by Harten and Hyman [92] for both linear and non-linear eigenvalues. Viscous fluxes are calculated with a central discretization scheme, while the Spalart-Allmaras turbulence model [93] is used for the turbulence closure problem. Boundary conditions are imposed weakly by utilizing the dummy cell concept [94].

The adjoint problem is derived from the primal governing equations via the discrete formulation. Only the mean-flow equations were adjoined and the turbulent quantities appearing therein were treated as constant. For one-equation turbulence model, this means that the turbulent eddy viscosity is assumed constant [12, 95].

The linear system of equations is solved by the Generalized Minimum Residual (GMRES) [96] algorithm. The flow solver and its discrete adjoint counterpart use the stabilization scheme described by [97].

2.8. Optimizers

Two different types of gradient-based optimizers are used in this thesis:

- SciPy L-BFGS-B [98]. The L-BFGS-B is a Quasi-Newton optimization algorithm available in the python SciPy software package [99]. It is a low memory consumption version of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method that approximates the inverse of the Hessian matrix based on the gradient changes with respect to the previous iteration. The appropriate step-size is determined by doing a line search. This algorithm is particularly well suited for unconstrained optimization problems with a large number of variables (there is no limit to the number of design variables). The use of the L-BFGS-B algorithm can also be extended constrained optimization problems if they are handled with the penalty method.

- SNOPT (Sparse Nonlinear OPTimizer). This is a software package used for solving large-scale optimization problems (constrained or unconstrained). It is based on a sparse Sequential Quadratic Programming (SQP) algorithm with limited-memory quasi-Newton approximations to the Hessian of Lagrangian. SNOPT is very effective to solve nonlinear optimization problems where functions or gradients are computationally expensive to evaluate. SNOPT ensures convergence from an arbitrary point, provided the functions are smooth (but they do not need to be convex). If necessary, SNOPT allows the nonlinear constraints to be violated but minimizes the sum of such violations. The version used in this thesis is SNOPT 7.6 (i.e., the trial version of SNOPT for research and academic purposes), hence problems are limited to 300 design variables and 300 constraints. A more detailed description of SNOPT is found in [100, 101].

Both the SciPy L-BFGS-B and SNOPT software packages can handle box constraints on variables of the type $\alpha_l < \alpha < \alpha_u$ where α_l and α_u are the lower and upper bounds of the design variable α. This avoids the optimizer from exploring unfeasible designs.

2.9. Design velocity approach

Using the CAD to represent the geometry in a gradient-based optimization framework requires to compute geometric sensitivities, which represent the surface grid point coordinates w.r.t. design parameters. These sensitivities can be approximated by finite differences using the design velocity approach [8]. The so-called design velocities are geometric sensitivities that represent the displacement normal to the geometry boundary of a point on the boundary due to a perturbation of a design parameter in the CAD model. Typically, a positive design velocity represents an outward movement of the boundary, whilst a negative is an inward movement [74]. The approach to compute the design velocities is based on a finite difference between the CAD's model shape before and after a perturbation of a CAD design parameter. It can work with any existing commercial CAD package and it is not affected by the changes in CAD model topology and face labelling. A brief description of this method is presented here. For a more detailed description, the reader is referred to [8].

In order to compute the design velocities associated to a CAD design parameter, the baseline and the perturbed CAD models are usually exported as STEP files. Then, the perturbed and unperturbed models are read in by Gmsh [102] or CADfix [103] and their grid generator is used to represent the CAD geometries using a surface mesh of triangular elements. The following step is to approximate the displacement of the model boundary due to a perturbation in the CAD design parameter. This is done by projecting the centroid of each surface mesh element in the unperturbed model onto the elements in the perturbed model. The question arises of which element in the perturbed model should be used for this projection. A multidimensional binary search tree known as KD-tree [104] and Barycentric coordinates [105] are used in this search to determine which perturbed element should be tested next for the projection and the process continues until a successful projection is found. In order to avoid projections onto the wrong face of the model, a successful projection is that one that ensures that the unperturbed boundary normal at the centroid to be similar to the perturbed boundary normal at the identified projected point. More details about this methodology are found in [8].

Since this approach is based on the computing shape differences between two surface meshes, it can introduce issues of surface to surface projection when computing the distances between the two surface meshes. Moreover, finite differences inaccuracies are inherited because of the truncation error. Therefore, an alternative to the design velocity approach is proposed in this thesis to address these shortcomings. The geometrical sensitivities will no longer be computed by finite differences but by applying algorithmic differentiation to the CAD kernel and grid generator. In chapter 3, the grid sensitivities obtained by Algorithmic Differentiation will be compared to the design velocities.

2.10. Parametric effectiveness

In an industrial optimization framework the geometry is initially created within the computer-aided design (CAD) environment and it is essential that the optimal geometry is also available in CAD format. One of the benefits of parameterizing the geometry in the CAD system is that higher level constraints can be imposed on the shape, allowing the optimized part to be manufactured. The disadvantage of using a CAD-based parameterization is that its success is mainly dependent on the expertise and skill of the user, since it is not obvious which parameters are the most effective to allow the optimizer to produce a shape change in the direction that the adjoint sensitivities dictate. In fact, a poor choice of CAD parameters will lead to a poor exploration of geometries, preventing the optimizer from exploring innovative geometries. For these reasons, it is vital to quantify how suitable is a set of CAD-based parameters for use in optimization.

In this context, Robinson et. al. [106] introduced the concept of parametric effectiveness (PE) of a CAD model (Eq. 2.52), in order to rate the quality of the parameters of a CAD-based model for use as optimization variables. The PE as defined by Robinson et. al. [106] is computed as the ratio between the maximum change in performance that can be achieved using existing parameterization (Eq. 2.50), to the maximum performance improvement that could be obtained if the model is not constrained by any parameterization and follows the shape changes dictated by the adjoint sensitivities (Eq.2.51).

$$\Delta J_{CAD,OBM} = -\sqrt{\frac{A}{\int_A \left(\sum_{i=1}^{n} \frac{dJ}{d\alpha_i} \frac{dX_S}{d\alpha_i} \right)^2 dA} \sum_{i=1}^{n} \left(\frac{dJ}{d\alpha_i} \right)^2} \tag{2.50}$$

$$\Delta J_{ideal,OBM} = -\sqrt{A \int_{A_j} \left(\frac{dJ}{dX_S} \right)^2 dA} \tag{2.51}$$

$$PE = \frac{\Delta J_{CAD,OBM}}{\Delta J_{ideal,OBM}} \tag{2.52}$$

The terms dJ/dX_S, $dX_S/d\alpha_i$ and $dJ/d\alpha_i$ represent the adjoint surface sensitivities, the surface grid sensitivities and the performance gradients respectively. The changes in performance that are used to compute the parametric effectiveness are normalized with respect to an overall boundary movement (OBM), which is defined as a unit root mean squared movement of the boundary. The PE can be used to select the parameters to localize the boundary movement in the regions of high surface adjoint sensitivity. The downside of the normalization with the overall boundary movement is that the benefits obtained in large boundary movements are removed, and those parameters α_i with high performance sensitivity are penalised if they produce a design movement in the regions of low surface adjoint sensitivity [106]. Moreover, the parametric effectiveness defined by Robinson et. al. [106] is mainly defined for optimization problems where the constraints are purely geometrical. Therefore, it is uncertain how to extend this methodology for constrained optimization problems where the constraints are aerodynamic or structural.

In this thesis, a novel indicator for parametric effectiveness will be defined and presented in chapter 6 in order to address the shortcomings mentioned before and to extend the use the parametric effectiveness to constraint optimization problems where the constraints can be of any type. The novel methodology will be used in chapter 7 for multilevel shape refinement to choose the best CAD-based parameterization during the coarse of the optimization.

2.11. Adaptive shape parameterization

In an industrial design workflow, very often the CAD models use a very large number of design variables to construct the geometry. For example, in [107] the parametric CAD model of a car mirror used 2925 feature parameters and the CPU time required to update the CAD geometry was nearly 3 hours, making the optimization of such geometry to be computationally expensive. Another downside of exploring rich design spaces with large number of CAD design variables is the convergence of the optimizer. A large number of design variables can lead to poor convergence rates for many optimization schemes [108]. Moreover, the large number of design variables can translate into a high cost associated to matrix operations, which can limit the use of Newton or quasi-Newton algorithms [51].

One solution to this problem would be to select a reduced set of CAD design parameters to use in the optimization and to freeze the rest of the parameters. However, this can hinder the optimizer from finding a superior design. This phenomena is illustrated by Fig. 2.12. Imagine that it is desired to find the maximum of the bivariate Gaussian distribution $J(\alpha_1, \alpha_2)$ shown at the left of Fig. 2.12. If the designer freezes the design variable α_2 by assuming that it has a constant value and the optimizer is only allowed to change the other design variable α_1, the objective space that the optimizer would see would no longer be the bivariate Gaussian distribution but that one shown at the right of Fig. 2.12. This example illustrates that the optimizer will find a worse optimum when using a reduced number of design variables from the whole set.

In this example, an alternative approach to the downselection of the design variables would be to start the optimization using one design variable (e.g., α_1) and then, at some point, to continue the optimization with two design variables (α_1 and α_2). This is the basic idea of an adaptive shape parameterization. The use adaptive shape parameterization or multilevel optimization, where the optimization starts with a small number of design variables (coarse parameterization) and ends with a large number (fine parameterization), can be an effective strategy to explore rich design spaces and to improve convergence rate, robustness and final solution of the optimization [77, 78, 108].

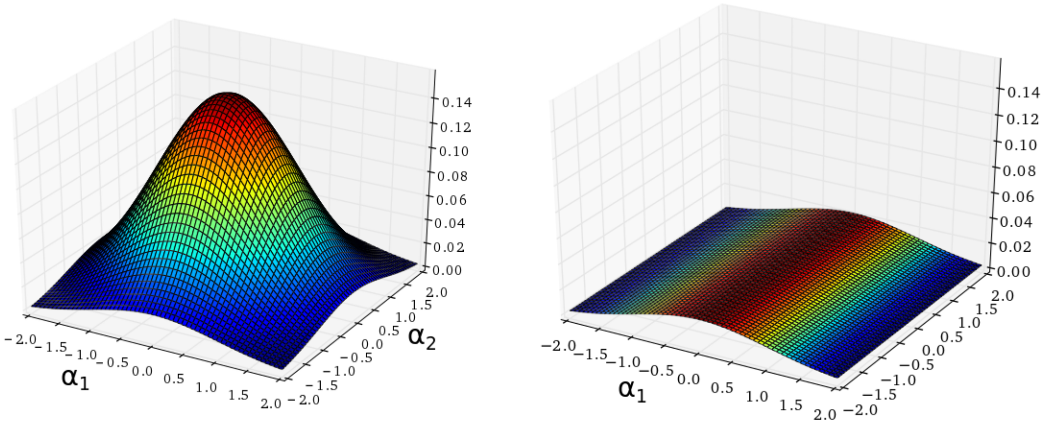


Figure 2.12.: Objective space when using: fine parameterization with two design variables (left), coarse parameterization with only one design variable (right)

In this thesis, the concept of adaptive shape parameterization will be used in chapter 7 in combination with the novel indicator for parametric effectiveness defined in chapter 6.

Chapter 3.

Differentiation of the CAD kernel and grid generator

In the previous chapter, the shortcomings of using finite differences or the design velocity approach for the computation of gradients in a CAD and adjoint based optimization framework are introduced. An alternative approach is presented in this chapter, which consists in using algorithmic differentiation techniques for the computation of accurate gradients. The main contents of this chapter have been published in [109].

Abstract. Although adjoint-based optimization methods have numerous advantages, e.g. the efficient computation of the gradient virtually independent of the number of design variables, the methods are not yet picked up largely by industry. One major bottleneck herein is that adjoint methods mainly work on deforming the CFD grid and as such loose the connection to CAD, the industry adopted standard for the design of components. After the optimization, a step is required which transforms the optimal shape, defined by grid points, back to a smooth CAD shape. This step is complicated, not easily automated and invariably impairs optimality of the shape, as CAD systems will only approximate the optimal shape. This chapter alleviates the problem by expressing the optimization problem through CAD parameters, rather than working directly on the CFD grid. This allows the optimal shape to remain defined within the CAD tool. The volume and surface grid sensitivities propagated to the CAD-based parameters are obtained through the differentiation of the CAD kernel and grid generation tools with algorithmic differentiation (AD). The in-house CAD kernel and grid generator have been differentiated in forward and reverse modes. The VKI LS89 axial turbine cascade is selected to demonstrate the methodology. The mid-span profile is parameterized by CAD based design parameters that have an important role in turbomachinery design practice. The geometrical as well as the volume grid sensitivities are obtained for each parameter, showing the influence of the design parameters on the profile and on the CFD grid coordinates. The geometrical and volume grid sensitivities obtained by AD are compared against second-order accurate central finite difference approximations through changing the perturbation step. The AD surface grid sensitivities are also compared to the sensitivities obtained by using the design velocity approach, which is a form of finite differences also. Finally, the performance of the AD differentiated sources is measured and compared to the original sources for the geometry and the mesh generation in terms of run time ratio, memory ratio and peak memory consumption.

3.1. Introduction

Aerodynamic shape optimization using low-cost, efficient and accurate computational methods is key for the aerospace industry, as it provides the opportunity to make significant design improvements at the early stage of the design chain.

For gradient-based optimization strategies, it is essential to compute the total gradient of the design chain, which is the derivative or gradient of the cost function with respect to a vector of design variables. This is usually done by calculating the derivatives separately for the various disciplines used in the design chain (e.g. grid generation, Computational Fluid Dynamics, Computational Solid Mechanics, etc.) and then combining them to obtain the total gradient. The derivatives can be obtained analytically, by finite differences or by differentiating the code(s). Adjoint-based optimization methods [36, 39, 110] have been introduced for more than two decades and have witnessed a tremendous improvement over time. They have become increasingly popular amongst the gradient-based optimization methods for its efficiency to compute the gradients at a cost that is independent of the number of design variables [12]. However, one major bottleneck herein is that adjoint methods mainly work on altering the shape by deforming the CFD grid [3–5, 36, 55, 56] and as such loose the connection to computer aided design (CAD), the industry adopted standard for the design of components. If the parameterization is done at the grid level, an additional step is required which transforms the optimal shape back to a smooth CAD shape. Many different procedures have been developed to approximate a given set of grid points with NURBS curves or surfaces [1, 57]. Despite the fact that these methods can be successfully employed to obtain a smooth CAD shape, the fitting error may invariably impair optimality of the shape as CAD systems will only approximate the optimal shape and it is not guaranteed that the final CAD shape will meet all the manufacturing design requirements and constraints.

Integrating the CAD in a gradient-based optimization framework introduces an additional bottleneck: the computation of the volume or the surface grid sensitivities. The volume and the surface grid sensitivities are defined as the derivative of the volume and the surface grid point coordinates w.r.t. the CAD design parameters respectively. In general, the volume grid points are referred in this thesis as the cell vertices. Any grid point in the interior domain of the mesh or in the boundaries are called volume grid points. In contrast, the surface grid points are considered to be only those vertices of the mesh that lay on the solid wall (i.e. the shape of the profile in discrete form). These sensitivities are often called shape sensitivities or mesh sensitivities in the literature. A number of approaches have been proposed in the literature for the computation of the surface grid sensitivities [8]. In general, most of the approaches use some form of finite differences and/or require the topology of the geometric model to remain constant (i.e., the same number and arrangement of surfaces, edges and vertices over the model boundary) after the model is perturbed [71, 111, 112]. But this constraint is difficult to enforce in practice. Kripac [113] computed these sensitivities by assigning tags or identities (IDs) to the topological entities of the unperturbed model. The entities with the same tag in the perturbed model were then used to compute these sensitivities. Techniques are described to deal with the situation where the IDs change after the CAD model is regenerated due to a parameter perturbation. An alternative approach to the naming problem is described in [114]. However, these techniques may not be robust enough where the CAD model changes significantly or where the order of constructions or the features change. Hence, Chen et. al. [115] concluded that the above mentioned problem remains unsolved. Agarwal et. al. [8] describe the design velocity method that allows to compute the shape sensitivities and to tackle possible changes in topology and the label renaming issues. However, since this approach computes the sensitivities by shape differences between two surface meshes, it can introduce surface to surface projection errors. Moreover, the accuracy of the gradients is dependent on the chosen step size for finite differences. A too large or too small step size will introduce inaccuracies due to the truncation error or limited arithmetic precision respectively. So far, the inaccuracies introduced by the design velocity approach have never been quantified by comparing to algorithmic differentiation (AD).

An alternative approach is presented in this chapter to compute the surface or volume grid sensitivities, which consists in differentiating the CAD kernel and grid generator using AD techniques. The use of AD allows to obtain accurate derivatives of the grid coordinates w.r.t. the CAD design variables. Therefore, the emphasis of this chapter is to:

1. Keep the shape defined within the CAD tool. The CAD parameters are directly used in defining the CAD geometry by means of Bézier and B-spline curves.
2. Apply AD to the in-house CAD and grid generation tools in both tapeless forward vector and in reverse modes. This allows to accurately predict the derivatives and circumvent the topology or renaming issues, the finite difference inaccuracies and also the projection errors of the design velocity approach.
3. Show the differences between the volume grid sensitivities computed by AD and by finite differences second-order central derivatives by tuning the step size.
4. Compare the approximated shape sensitivities by the design velocity approach with the accurate surface grid sensitivities computed by AD.
5. Compare the run time ratio, memory consumption ratio and peak memory consumption required to compute gradients w.r.t. CAD design variables when we use the differentiated sources by AD.

The work is carried out using a computer aided design and optimization tool for turbomachinery applications (CADO) [69], which has been used so far mainly in gradient free optimization methods, and hence the work presented herein represents the first steps to differentiate the entire CADO code to be used in gradient based optimization.

The LS89 [116–118] axial turbine cascade is selected as a demonstrator to test the methodology. It is a relatively modern high pressure turbine nozzle guide vane specifically designed for subsonic flows downstream of an aero-engine combustion chamber. The profile of the LS89 was designed and optimized in the two-dimensional space at the VKI for a 0.9 downstream isentropic Mach number by an inverse method described in [116]. It was tested at the VKI laboratories with the aim to use the experimental data as a benchmark to validate the VKI CFD prediction capabilities in the late 80's. The complete experimental results were published during the 1990 International Gas Turbine Conference held in Brussels [117]. The final report is the VKI Technical Note TN174 [118]. The LS89 has been tested for different Reynolds and Mach numbers in the VKI-CT2 test facility.

The remainder of the chapter is organised as follows. Section 3.2 presents the methodology to construct the geometry, the fluid computational grid and to compute the different derivatives. The results are shown in Sec. 3.3. Finally, the chapter terminates with the main conclusions.

3.2. Methodology

The design chain process shown in Fig. 3.1 starts typically from a set of input design parameters which allow the designer to build a CAD model, from which it is necessary to generate a computational fluid grid to perform CFD analysis. After obtaining a converged flow solution by the primal solver, the cost function is evaluated as a post-processing step. The differentiation of the design chain allows to compute the performance sensitivities w.r.t. the CAD parameters which are necessary for gradient based optimization algorithms. The following subsections present a description of the methods employed in this work for the geometry and grid generation steps and the calculation of the different derivatives.

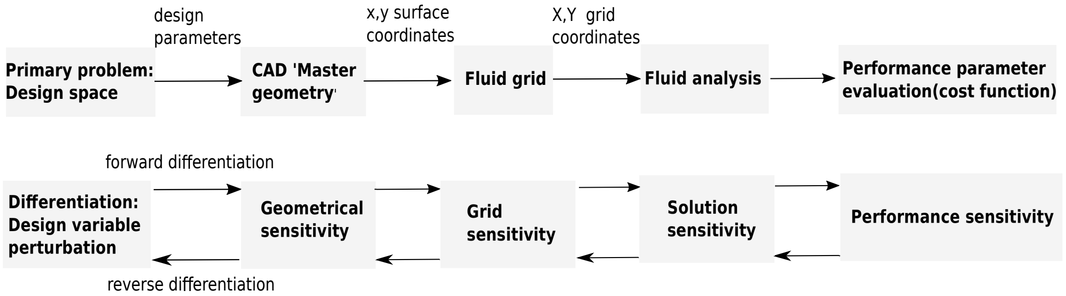


Figure 3.1.: Chain to compute the performance sensitivity.

3.2.1. Constructing the LS89 profile with Bézier and B-spline curves

The construction of the turbine profile is based on the parameterization described in [119]. The turbine profile, which is shown in Fig. 3.2, is defined herein with three independent curves, two B-spline curves for the suction side (SS) and pressure side (PS), and a circular arc at the trailing edge (TE) to close the profile. In order to construct the profile, first a camber line is constructed, which is used to define the position of the control points of the SS and PS B-spline curves relative to the camber line. The camber line is defined by an inlet blade metal angle β_{in}, an exit blade metal angle β_{out}, an axial chord length c_{ax} and a stagger angle γ. The position of the leading edge (LE) and TE is defined by the axial chord length and the stagger angle. By defining an inlet and outlet lines respecting the inlet and outlet blade metal angles, one can intersect the lines and define the point P_{mid} shown in Fig. 3.2.

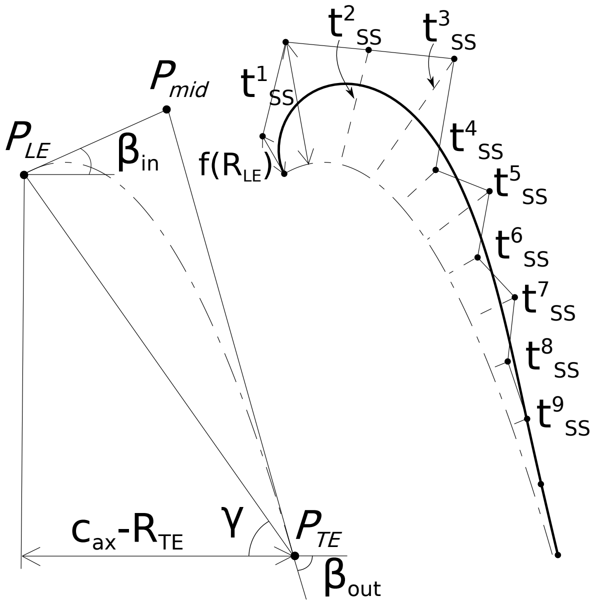


Figure 3.2.: Construction of the suction side by a B-spline curve.

The points P_{LE}, P_{mid}, P_{TE} define the control points of the 2nd order Bézier curve describing the camber line. By specifying a stretch factor and a number of points, the camber line is divided into a number of intervals. For each point obtained on the camber line (except the first and last two points) a normal distance is specified by the designer to obtain the corresponding control point of the SS and PS B-spline curves. The normal distance of the first control point relative to the camber line is a function of the LE radius, in order to guarantee G_2 geometric continuity (i.e. equal curvature) between the SS and PS B-splines at the stagnation point of the leading edge. The normal distance of the last control point relative to the camber line is equal to the TE radius. For the second last point, the normal distance is computed by specifying a wedge angle. Usually more control points are placed on the SS than on the PS because the highly curved shape of the SS is deemed to play a bigger role than the PS during the aerodynamic shape optimization. Finally, the shape of the cascade is fixed after specifying the pitch, which can be computed for a given solidity. The solidity plays an important role in turbomachinery design practice, and this justifies the use of this parameter as an independent variable.

The LS89 profile geometry comes first from point data [118] that has been fitted with the parameterization method described before. The cascade geometry is shown in Fig. 3.3 and uses 24 parameters (see Table 3.1 and 3.2). The position of the B-spline control points of the suction side and pressure side is shown in Fig. 3.3.

3.2.2. Generation of block-structured smooth grids

The generation of a boundary conformal curvilinear grid for the turbomachinery flow geometry of this study was achieved by subdividing the domain into multiple structured grid blocks which are

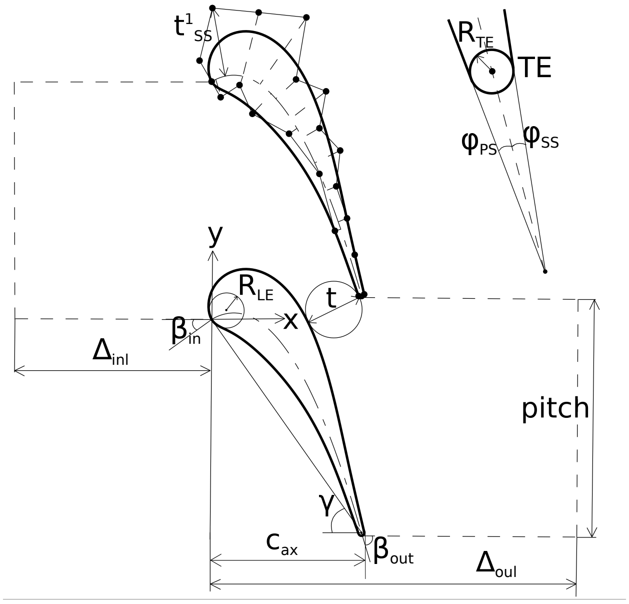
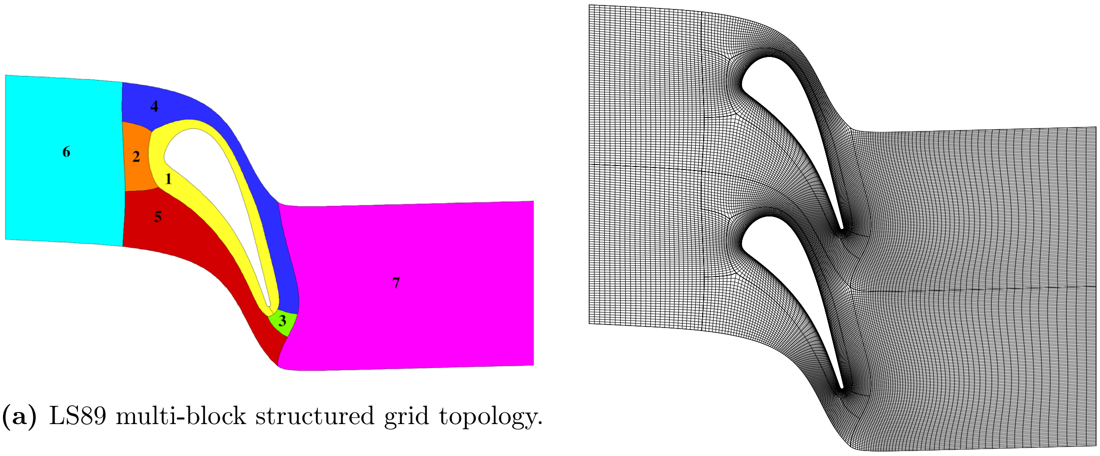


Figure 3.3.: LS89 parameterization.

independent from each other but share a common interface. A single block approach for a relatively highly staggered profile, such as the LS89 turbine profile, would result in a poor quality grid due to the high skewness of the cells. Therefore, a multi-block approach was preferred because it allows to have more detailed control of the element size and shape. Figure 3.4a shows the selected multi-block structured topology for the LS89 CFD domain. One O-grid block is placed around the profile and there are six additional H-grid blocks, four of them distributed around the profile and the remaining two being used as the inlet and outlet blocks. The fluid mesh has $300,000$ grid points approximately.



(a) LS89 multi-block structured grid topology.

(b) LS89 mesh topology and final smoothed grid.

Figure 3.4.: *LS89 mesh topology and final smoothed grid*

The first step in the grid generation process consist of generating an initial grid on the edges of the different blocks. The internal grid is subsequently initialized using Transfinite Interpolation (TFI) equations from the boundary mesh points, which are described by the Eqns. 2.21 and 2.22 [79]. The generation of an algebraic grid for each block is efficient and easy to implement but there is no

Design parameters	Acronyms	Value
Solidity	σ	1.118
Axial Chord Length	c_{ax}	36.955 mm
Stagger Angle	γ	54.9°
LE Radius	R_{LE}	4.126 mm
TE Radius	R_{TE}	0.710 mm
TE Wedge Angle SS	φ_{SS}	4.0°
TE Wedge Angle PS	φ_{PS}	2.5°
Inlet Angle	β_{in}	0.0°
Outlet Angle	β_{out}	74.0°
Stretching Factors PS	k_{PS}	1.175
PS Thickness (x4)	t^1_{PS}, ..., t^4_{PS}	see Table 3.2
Stretching Factors SS	k_{SS}	1.1
SS Thickness (x9)	t^1_{SS}, ..., t^9_{SS}	see Table 3.2

Table 3.1.: *Design parameters used to define the LS89 blade profile*

B-spline Control Point Index j	t^j_{SS} [mm]	t^j_{PS} [mm]
1	16.750	2.250
2	15.900	6.650
3	19.690	2.300
4	6.750	0.040
5	10.750	
6	4.750	
7	6.850	
8	2.565	
9	2.295	

Table 3.2.: *B-spline control point normal distances relative to the camber line*

guarantee that the grid lines will not intersect, fold (see Fig. 2.8), or that the corners from the side will not be propagated inside the domain. Given the difficulty to generate successful smooth grids using TFI equations, it is convenient to smooth them afterwards by solving a partial differential equation (PDE) of some type (elliptic, hyperbolic, parabolic). The elliptic PDE is commonly used for grid generation for ducted flows. The elliptic grid generation equations shown in Eqns. 2.25-2.26 were pioneered by Thompson et. al. [81] and are employed in this work. Grid behaviour control is achieved through the introduction of forcing function terms in the manner of Steger and Sorenson [82] in the O-grid block. The Steger and Sorenson method is an iterative forcing function approach that adjusts the boundary forcing functions to meet a prescribed angle and spacing constraint at the boundary. The forcing function is then interpolated into the domain based on the boundary values of the source terms. In all the H-grid blocks, the source terms have been set to zero and hence the elliptical equations being solved are the so called Laplace equations. A more detailed description about the grid generation procedure is presented in Sec. 2.5.1.

3.2.3. Geometrical and grid sensitivities using algorithmic differentiation

In this work, the geometrical sensitivites are the derivatives of the geometry boundary coordinates x-y w.r.t. the CAD parameters. The grid sensitivities are the derivatives of the CFD grid coordinates X-Y w.r.t. the CAD parameters. A distinction is made between the volume and the surface grid sensitivities, because the volume grid is referred to the full grid and the surface grid is referred to the grid points on the shape surface.

The geometrical and grid sensitivities have been computed by applying algorithmic differentiation (AD) to the CAD kernel and grid generator. AD is a technique for computing analytic derivatives of programs [24]. The idea of AD is to differentiate analytically each elementary mathematical operation performed by the code and accumulating the derivatives by applying the chain rule in an automatic fashion. In this way, it is possible to build up the cost function or derivative of the output w.r.t. the input variable, which can then be used for an optimization algorithm.

One distinguishes between the forward and reverse mode of AD. In forward mode one is interested to know the derivative of the outputs depending on the setting of the input sensitivities, whereas in reverse mode one is interested in the derivative of all inputs for the number of outputs. Both modes allows to accurately compute the sensitivities with the best possible machine accuracy. Also, it is possible to use either the scalar or the vector mode. In the forward scalar mode, it is required to execute the code as many times as the number of input variables. Alternatively the forward vector mode allows to compute the derivatives in one run of the differentiated code. In this run, the primal (undifferentiated) part is only computed once and the derivative (differentiated) part is computed within a loop such that compiler can optimize by using vectorization. On the other hand, in the reverse scalar mode you need as many runs as the number of output variables and in reverse vector mode you only require running the differentiated code once at the expenses of using more resources. The reverse mode always requires storing the intermediate operations and the variable values throughout the evaluation of the code in a trace or tape, which can result in a large memory requirement. An introduction to the AD techniques is presented in Sec. 2.6. For a more detailed description of the different techniques for evaluating derivatives using AD the reader is referred to [24].

The AD tool used for the differentiation of the CAD kernel and grid generator is called ADOL-C [9] and is developed at the Department of Mathematics at the University of Paderborn. ADOL-C is written in C++ and uses overloaded operators and functions. The ADOL-C derivative evaluation routines can be called from C, C++, Fortran and any other language that can be linked with C [9]. First, the geometrical and grid sensitivities are calculated in tapeless forward vector mode and linked with the volumetric adjoint sensitivities to give the performance sensitivities. Also, the trace reverse mode of ADOL-C is exploited to compute the performance sensitivities.

3.2.4. Performance tests

To measure the performance of using AD to compute the derivatives of the cost function w.r.t. the CAD design parameters, the following cost functions are defined:

- The throat distance J_t is computed after the creation of the CAD geometry. This is computed as the minimum distance between the suction side and the trailing edge of the adjacent blade (this is referred as t in Fig. 3.3).
- The average X coordinate J_X of the fluid grid is computed after the creation of the CFD grid. This objective function does not have any physical meaning but it is useful for computational performance test purposes.

Three different derivative computation methods are presented to measure the performance of computing the derivatives of the cost functions J w.r.t the CAD parameters $dJ/d\vec{\alpha}$:

- Using finite differences. The derivatives $dJ_t/d\vec{\alpha}$, $dJ_X/d\vec{\alpha}$ are computed by the first order backward scheme by perturbing each CAD design variable by a negative step size and computing the gradient as the change in the cost function divided by the perturbation step.
- Using the AD forward scalar and vector modes. The derivatives $dJ_t/d\vec{\alpha}$ and $dJ_X/d\vec{\alpha}$ are the outputs of running the differentiated code for the geometry and mesh creation respectively.
- using the reverse scalar mode of AD. The derivatives $dJ_t/d\vec{\alpha}$ and $dJ_X/d\vec{\alpha}$ are the outputs of running the differentiated code fo the geometry and mesh creation respectively, seeding the output with 1.0 and evaluating the trace in reverse fashion.

The parameterization used for the performance tests is similar to that one shown in Fig. 3.3, but it allows to introduce more control points in the suction side and pressure side B-spline curves by using the knot insertion algorithm [76] (see Sec. 2.4.1). This allows introducing more design variables in the parameterization and testing the performance of the differentiated sources when we use a large number of design variables.

3.3. Results

The geometrical and the grid sensitivities are obtained for all the design parameters, showing the influence that each design parameter has on the aerofoil surface (x,y) coordinates and the CFD grid (X,Y) coordinates respectively. The discussion herein is limited to the parameters t^1_{SS}, R_{LE}, c_{ax} (see Table 3.1) for the sake of simplicity. In the following subsections, the AD geometrical sensitivities $d\vec{x}/d\vec{\alpha}$, which are defined as the change in the (x,y) turbine profile coordinates with respect to a design parameter change, will be discussed first. The volume grid sensitivities $d\vec{X}/d\vec{\alpha}$, which are defined as the change in the (X,Y) grid coordinates with respect to a design parameter change will be discussed next. Third, the geometrical and volume grid sensitivities obtained by

AD will be compared against FD. Also, the AD surface grid sensitivities will be compared to the approximated sensitivities by the design velocity approach. Finally, a summary of the run time, run time ratio, peak memory consumption requirements will be given for the different approaches presented to compute the gradients of each cost function J (i.e., J_t and J_X) w.r.t. the CAD design variables.

3.3.1. AD geometrical sensitivities

Figures 3.5, 3.6 and 3.7 show the magnitude and direction of the geometrical sensitivities obtained for the t^1_{SS}, R_{LE}, c_{ax} design parameters, respectively. Different scale factors are used to display the length of the vectors.

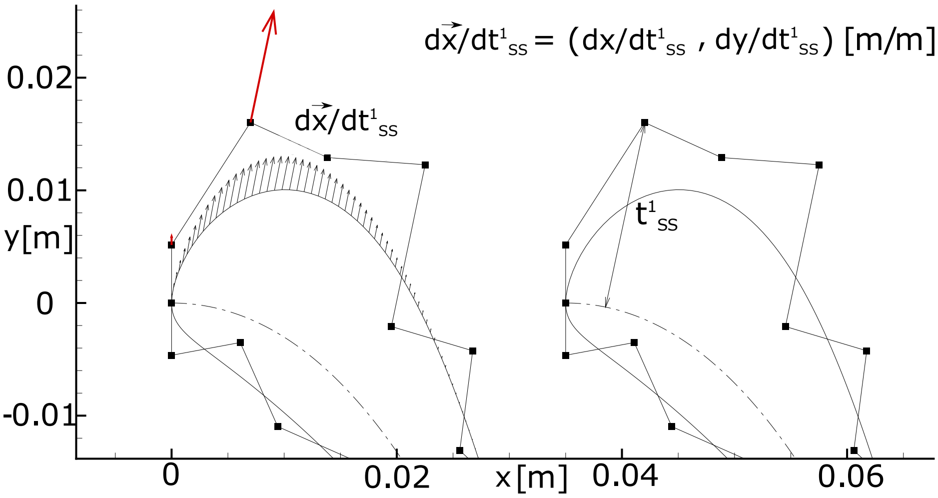


Figure 3.5.: Geometrical sensitivity for t^1_{SS}, Vector Scale Factor = 0.01.

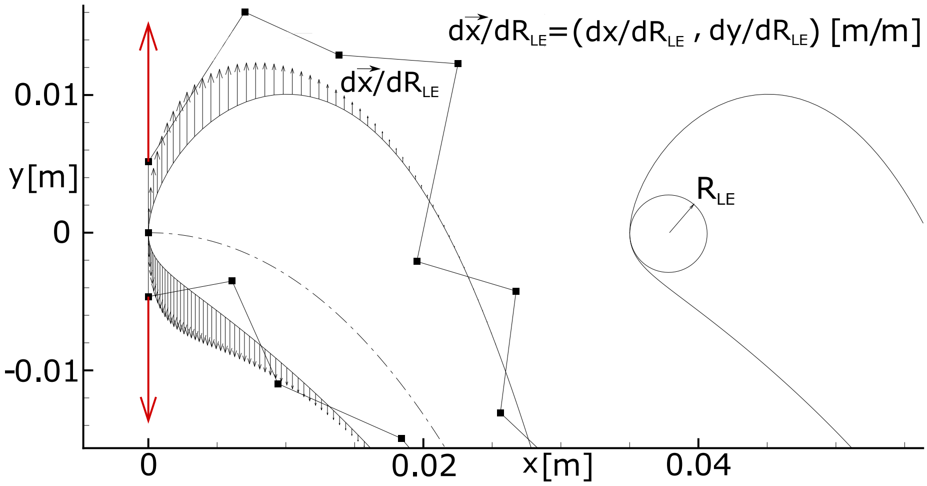


Figure 3.6.: Geometrical sensitivity for R_{LE}, Vector Scale Factor = 0.0016.

Figure 3.5 clearly shows that an infinitesimally small perturbation of t^1_{SS} causes the movement of the points along the suction side shoulder in the normal direction in which the suction side first

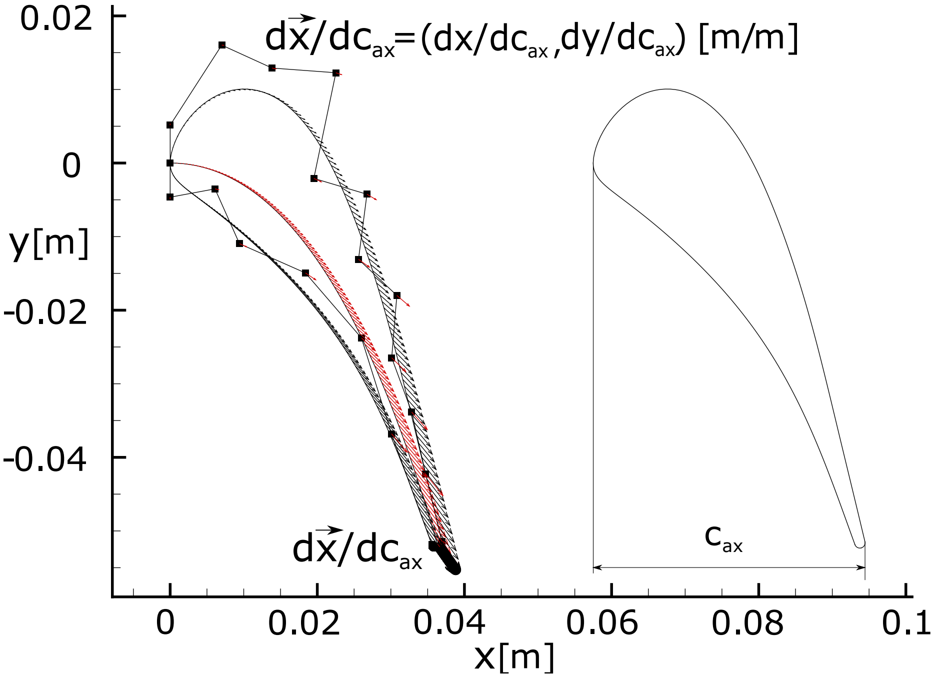


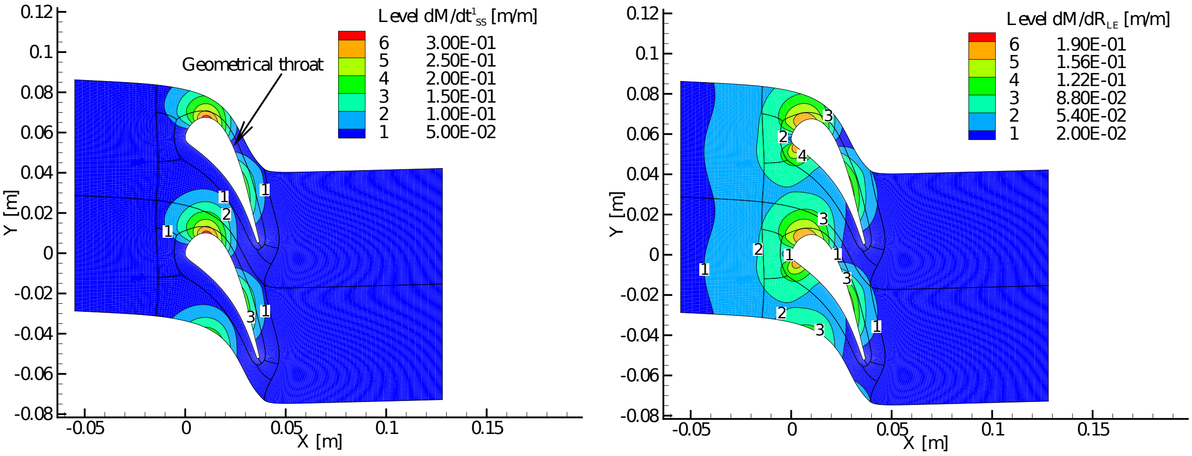
Figure 3.7.: Geometrical sensitivity for c_{ax}, Vector Scale Factor = 0.0025.

B-spline control point position is specified relative to the camber line. A similar perturbation in the R_{LE} forces the first control point of the suction and pressure side to move away from the camber line because their position is a function of the LE radius, making the points along the LE region to move in the y direction accordingly as displayed in Fig. 3.6. Figure 3.7 shows that a perturbation in the c_{ax} causes the points along the camber line, SS and PS curves to move away from the stagnation point at the LE. Although the scale factor does not allow to show the magnitude of all the sensitivities, all the control points (except the first of the SS and PS curves) sensitivity vectors point away from the stagnation point because the TE point (P_{TE} from Fig. 3.2) also moves away in both x and y directions. The TE point movement is not in the x direction only because an increase of axial chord by virtue of a constant stagger angle requires the TE point (and the camber line) to move in the (x,y) space.

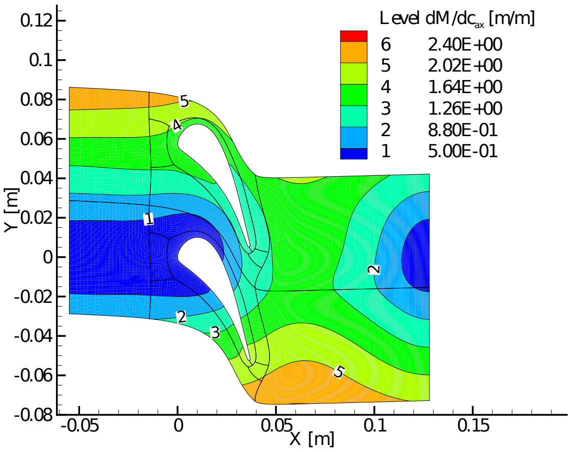
3.3.2. AD volume grid sensitivities

Figure 3.8 shows the magnitude of the volume grid sensitivities $dM/d\alpha_i$ (Eq. 3.1) obtained for the t^1_{SS}, R_{LE}, c_{ax} design parameters, respectively. In general, a perturbation in the design parameter can cause two type of perturbations in the CFD grid point coordinates. First, the grid point distribution along the edge of the curve itself can change due to the fact that the normalised arc-length functions used to distribute points in the edge undergo some changes. This type of perturbation that moves the surface grid points tangentially to the surface can also propagate into the interior domain of the grid but it has a tendency to decay rapidly. Secondly, the grid points coordinates in the fluid domain change as well according to the overall grid point field movement sensed by them. The source of this perturbation is stronger than for the first type and originates in the region where the geometrical sensitivities are not zero.

$$\frac{dM}{d\alpha_i} = \sqrt{\left(\frac{dX}{d\alpha_i}\right)^2 + \left(\frac{dY}{d\alpha_i}\right)^2} \tag{3.1}$$



(a) Magnitude of t^1_{SS} AD volume grid sensitivities (b) Magnitude of R_{LE} AD volume grid sensitivities



(c) Magnitude of c_{ax} AD volume grid sensitivities

Figure 3.8.: Magnitude of the t^1_{SS}, R_{LE} and c_{ax} AD volume grid sensitivities.

Figure 3.8a clearly shows that a small perturbation in the t^1_{SS} causes the grid points in the fluid domain above the SS shoulder to move and this movement also propagates into the PS fluid domain through the periodic interface. The rear suction side grid points are also affected due to the fact that the arc-length distribution changes. The geometrical throat is used herein to split the SS curve into two edges to have more control of the grid point distribution along the SS curve. This means that the geometrical throat on the SS (see Fig. 3.8a) is the end of the SS pre-throat and the beginning of the SS post-throat edges, which are using different arc-length distributions. Therefore, it is expected that a perturbation in t^1_{SS} causes a grid point position variation along the SS pre-throat and post-throat edges because the length of these edges changes. However, as the geometrical throat (X,Y) coordinates remain fixed, the grid points in the vicinity of the SS geometrical throat do not sense any perturbation.

Figure 3.8b shows that the biggest influence of R_{LE} is in the vicinity of the largest geometrical sensitivities. The perturbation on the SS also propagates into the PS interior domain through the periodic interface. Also, the changes in the normalised arc-length functions for the PS and SS edges result into grid point movements in both PS and SS curves. As the SS is more curved than the PS, it is expected that a perturbation in the R_{LE} changes the total length of the SS curve more than for the PS. This is the reason why the R_{LE} volume grid sensitivities at the rear SS are bigger than the rear PS. The grid point displacements along the SS and PS also propagate into the interior of the fluid domain but decay more rapidly than in those that propagate from the LE region. The minimum sensitivity areas around the profile correspond to those grid points that are fixed, like the stagnation point, the geometrical throat position, and the SS and PS end points towards the TE. In these points, the arc-length functions do not change because they take the values zero or one, depending on the orientation in which the grid point distribution has been applied to the edges.

Figure 3.8c shows a non-periodic character. As can be seen, the volume grid sensitivity field is not repeated passage after passage. This is because the solidity, as an independent parameter, remains constant under a change of the axial chord length, resulting in a proportional change in pitch. Identical points on the opposite side from the periodic boundary will experience a jump in the y-component of the grid sensitivity, proportional to the change in pitch. In the figure, the jump value has been added to the top CFD domain such that a continuous field is visualized. The choice to use solidity as an independent parameter is inspired by its importance in classical turbomachinery design practice.

3.3.3. Comparison to finite differences

The geometrical and volume grid sensitivities obtained by AD are compared against FD second-order central derivatives by tuning the step size. Figures 3.9a and 3.9b show the maximum error in magnitude between the FD approximated and the exact AD sensitivities for the t^1_{SS}, R_{LE} and c_{ax} design parameters. The error is very sensitive to the chosen step size and the step size needs to be relatively small to have a small error. Figures 3.9a and 3.9b show that the error in both cases reduces with order two as the step size is reduced up to a critical step size, as to be expected for a central second-order FD scheme. If the step size is chosen too small below this critical step size, the numerical representation of the FD approximation becomes unstable and more error-prone due to the limited machine accuracy. The error increases with order one approximately if the step size is reduced below the critical step size. Figures 3.9c, 3.9d and 3.9e show the grid sensitivity error for the t^1_{SS}, R_{LE} and c_{ax} design parameters if the step size is 1mm. It has to be noted that the maximum grid sensitivity error shown in Fig. 3.9d is relatively large, as it is only two orders of magnitude lower than the maximum volume grid sensitivities shown in Fig. 3.9d, and this can eventually result in wrong gradient information.

3.3.4. Comparison to design velocities

A comparison between the surface grid sensitivities obtained by AD and the design velocities computed using the method described in [8] follows. Figures 3.10, 3.11 and 3.12 show the surface grid sensitivities computed by AD evaluated at the solid wall boundary on the left and the design velocities V_n on the right. The AD surface grid sensitivities capture both the tangential and the normal components of the design movement produced by an infinitesimal perturbation of the design variable (e.g., Fig. 3.10 left), whereas the design velocities only capture the normal component to the boundary (e.g., Fig. 3.10 right). Table 3.3 shows that the AD surface grid sensitivities and the design velocities are significantly different, the largest root mean square average error between the design velocities and the surface grid sensitivities is of the order of 49.5% for the c_{ax} parameter. However, if we project the AD surface grid sensitivities to the normals of the boundary (i.e., AD_n), the design velocities can match reasonably well the normal component of the surface grid sensitivities, provided that an appropriate step size is chosen for the computation of the design velocities.

Design parameter	$\mathrm{err}(V_n - AD)$	$\mathrm{err}(V_n - AD_n)$
t^1_{SS}	14.24%	0.16%
R_{LE}	19.20 %	0.68%
c_{ax}	49.47%	2.94%

Table 3.3.: *Error between design velocities and AD surface grid sensitivities*

3.3.5. Performance tests

The performance of the differentiated sources are analysed and compared to the original sources. The time required for creating the CAD geometry and evaluating the throat distance J_t and the corresponding derivatives $dJ_t/d\vec{\alpha}$ is measured and summarised in Table 3.4 and 3.5 for $p = 1$ and $p = 205$ directions respectively. The number of directions p represent the number of design variables $\vec{\alpha}$ for which we want to compute the corresponding derivatives. Also, the time required to generate the fluid mesh and evaluate the average X coordinate J_X and the corresponding derivatives $dJ_X/d\vec{\alpha}$ is measured and summarised in Table 3.6 and 3.7 for $p = 1$ and $p = 205$ directions respectively. The averaged timings and run time ratios are based on 15 measurements.

	Original CAD sources	Traceless forward	Trace-based reverse
Avg. time [ms]	0.398	1.570	10.916
Run time ratio	-	3.94	27.37

Table 3.4.: *Timings with original and differentiated geometry creation sources with number of directions $p = 1$ (scalar mode). The results are based on 15 measurements.*

	Original CAD sources	Traceless forward	Trace-based reverse
Avg. time [ms]	0.398	26.98	10.895
Run time ratio	-	67.64	27.32

Table 3.5.: *Timings with original and differentiated geometry creation sources with number of directions $p = 205$. The results are based on 15 measurements.*

According to the theory [24], the run time ratio between the derivative computation using AD and the function (primal) evaluation should be between $[1 + p, 1 + 1.5p]$ and $[1 + 2q, 1.5 + 2.5q]$ for

	Original mesh sources	Traceless forward	Trace-based reverse
Avg. time [s]	1.77	88.36	256.00
Run time ratio	-	49.92	144.64

Table 3.6.: *Timings with original and differentiated fluid mesh creation sources with number of directions* $p = 1$ *(scalar mode). The results are based on 15 measurements.*

	Original mesh sources	Traceless forward	Trace-based reverse
Avg. time [s]	1.77	393.39	255.60
Run time ratio	-	222.26	144.41

Table 3.7.: *Timings with original and differentiated fluid mesh creation sources with number of directions* $p = 205$*. The results are based on 15 measurements.*

the forward and reverse modes of AD respectively, where p and q is the number of directions and of adjoints respectively. In this work, $q = 1$ and hence the expected run time ratio range in reverse mode is $[3, 4]$.

The results shown in Table 3.4 and 3.5 show that the run time ratios for the geometry creation using traceless forward mode of AD either agree with the theoretical bounds (e.g., $p = 1$) or are below the theoretical bounds (e.g., $p = 205$). The compiler optimization could be a reason for this good run time ratio. Tables 3.6 and 3.7 show that the run time ratio using the traceless forward mode of AD is higher than for the geometry creation sources. This slow-down could be due to the fact that the mesh creation involves initializing and destroying many objects during the large smoothing iterative procedure and the compiler is not able to optimize so well this part of the code.

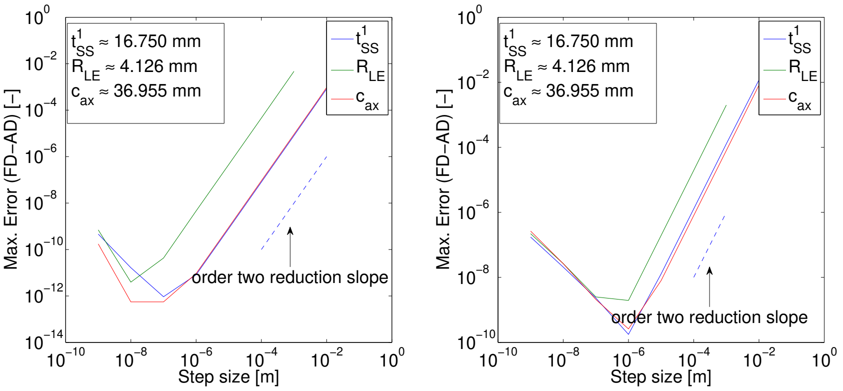
The results shown in Tables 3.4, 3.5, 3.6 and 3.7 show that the run time ratios when using the trace reverse mode of AD are well above the theoretical values for both the geometry and mesh creation sources. The iterative procedures present in the geometry and mesh creation could be the reason for this. In the geometry creation code, there are certain point inversion iterative procedures, whereas in the mesh creation sources there is a large iterative procedure to smooth the grid. Large iterative procedures slow down the performance of differentiated sources with trace mode of AD, because more intermediate variables and operations need to be traced. When the memory required to trace this operations exceeds the memory available in the RAM, the trace is then written into the hard-disk, which further slows down the taping process. Figure 3.13 shows the run time ratio required to run the mesh sources with traceless mode and trace reverse mode of AD for $p = 1$ as the number of mesh smoothing iterations is increased. The larger number of iterations, the higher is the slow down.

An overview of all the run time ratios and memory ratios evaluated in the range of $p = 1$ to $p = 205$ directions is shown in Fig. 3.14a and 3.14b respectively for the geometry creation sources. The memory ratio is defined as the ratio between the peak memory consumption of the differentiated sources and the primal sources. The finite difference approach requires using the same memory usage than the primal. The computation of the gradients for the geometry creation (i.e., $dJ_t/d\vec{\alpha}$) is faster using AD than the finite differences approach especially for large number of directions, but requires more memory.

The run time ratios and memory ratios evaluated in the range of $p = 1$ to $p = 205$ directions for the mesh creation sources is shown in Fig. 3.15a and 3.15b respectively. In this case, the computation of the gradients for the mesh creation (i.e., $dJ_X/d\vec{\alpha}$) is faster using trace reverse mode of AD than the finite differences approach for $p \geq 145$ directions, but it requires using more memory.

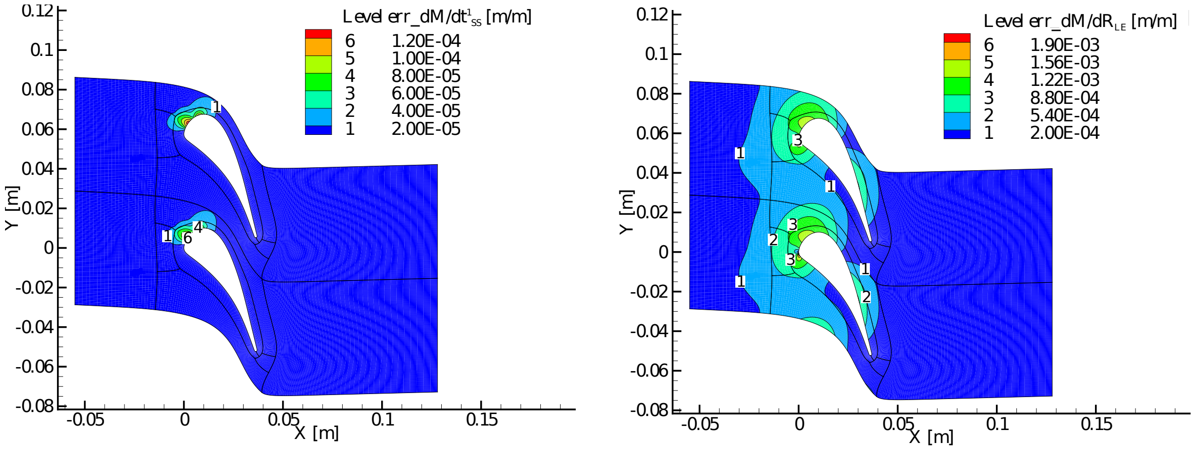
3.4. Conclusions

This chapter presents a methodology to integrate the CAD kernel and grid generation tools within a CFD adjoint-based optimization framework by expressing the optimization problem through CAD parameters, rather than working directly on the CFD grid. The advantage of including the CAD model in the design system is that higher level constraints can be imposed on the shape, allowing the optimized model or component to be manufactured. This requires the computation of the grid sensitivities, which can be done by differentiating the CAD kernel and grid generator with algorithmic differentiation (AD). The use of AD allows computing the grid sensitivities to machine accuracy and avoid the limited arithmetic precision and the truncation error of finite differences. The differentiation is done in traceless forward and trace reverse modes of AD and is tested in the construction of the LS89 turbine cascade geometry and fluid mesh construction. The two dimensional profile, constructed by means of Bézier and B-spline curves, is parametrized by 24 design parameters. The geometrical as well as grid sensitivities are obtained for each parameter, showing the influence that some design parameters have on the profile and on the CFD grid coordinates. The geometrical and volume grid sensitivities obtained by AD are compared against finite differences second order central derivatives by tuning the step size. The error introduced by FD is very sensitive to the chosen step size and its magnitude can be relatively close to the sensitivity value itself if the step size is not chosen appropriately. Therefore, applying AD to the CAD kernel and grid generator allows to obtain accurate sensitivities with the best machine accuracy. The design velocities are compared to the AD surface grid sensitivities. There is a relatively large error introduced by the design velocities. Nevertheless, the sources of error of the design velocities (i.e., truncation error, limited arithmetic precision and mesh surface to surface projection errors) can be reduced by choosing an appropriate step size for the computation of the design velocities. The design velocities can match reasonably well the normal component of the AD surface grid sensitivities if the step size is appropriately chosen. The computation of gradients for the geometry creation or the fluid mesh creation is generally faster using the trace reverse mode of AD than finite differences, specially when using a large number of directions, but it requires more memory. The cost of computing the performance sensitivities using the AD reverse mode is almost independent of the number of design variables, whereas for the AD traceless forward mode the cost is proportional to the number of design variables. Optimization studies would be beneficial to demonstrate the integration of the CAD in the optimization framework.



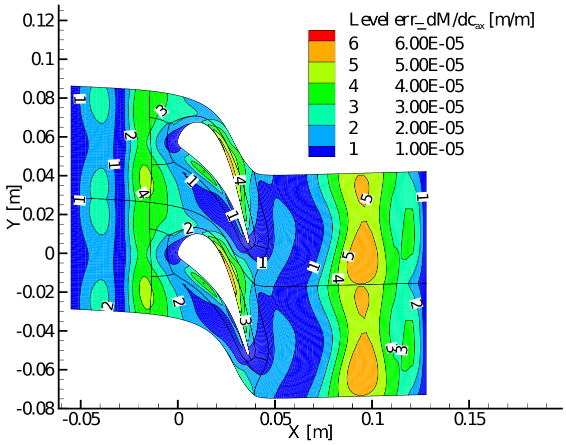
(a) Geometrical sensitivity error

(b) Grid sensitivity error



(c) t$^1_{SS}$ volume grid sensitivity error, step size = 1E-03.

(d) R$_{LE}$ volume grid sensitivity error, step size = 1E-03



(e) c$_{ax}$ volume grid sensitivity error, step size = 1E-03.

Figure 3.9.: Max error between FD and AD (a-b), volume grid sensitivity error for t^1_{SS}, R_{LE} and c_{ax} (c-e).

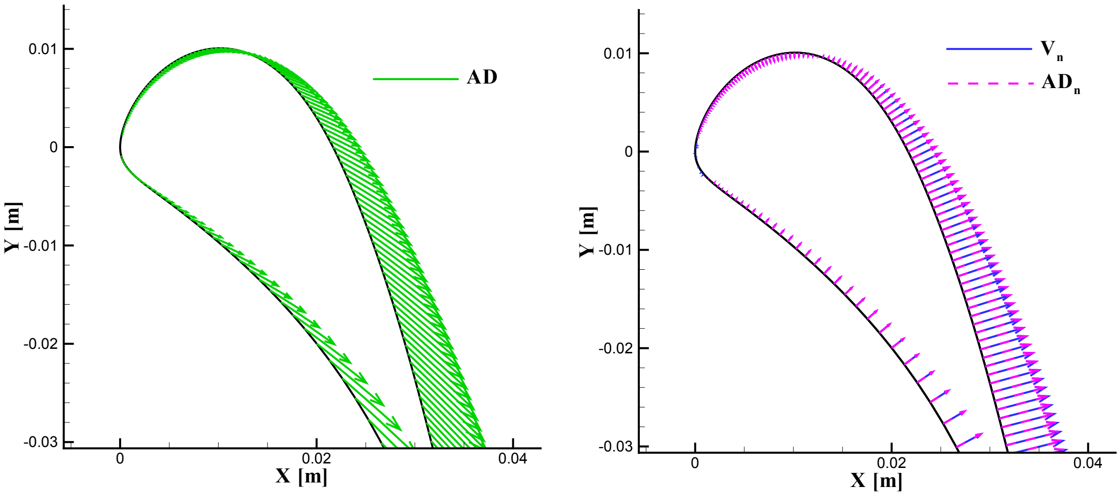


Figure 3.10.: c_{ax} AD surface grid sensitivities (left), c_{ax} design velocities vs. AD surface grid sensitivities projected to normals (right).

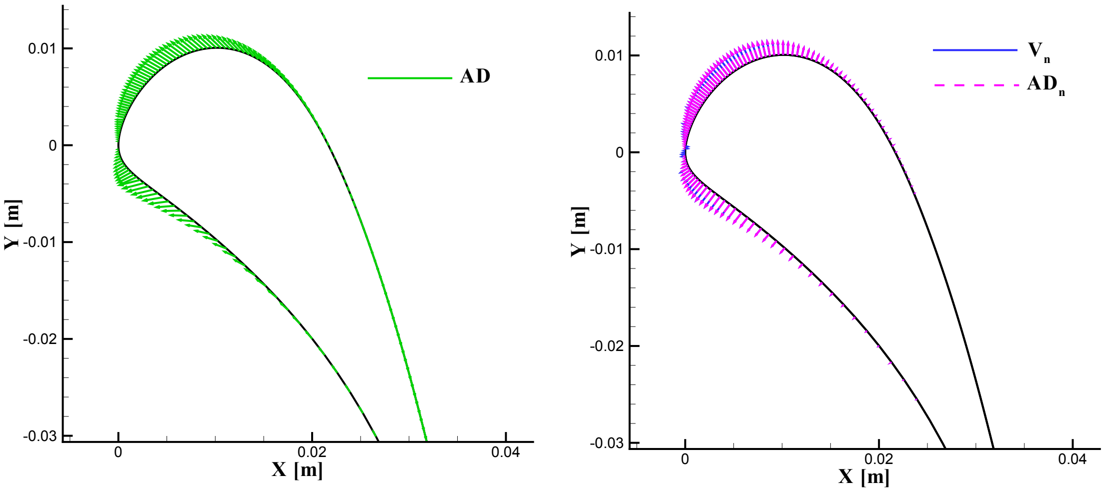


Figure 3.11.: R_{LE} AD surface grid sensitivities (left), R_{LE} design velocities vs. AD surface grid sensitivities projected to normals (right).

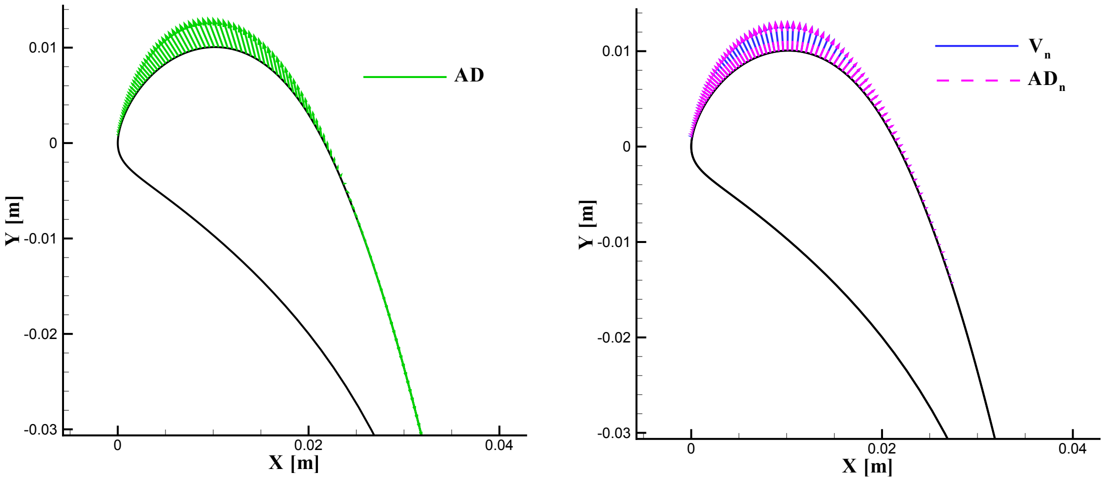


Figure 3.12.: t^1_{SS} AD surface grid sensitivities (left), t^1_{SS} design velocities vs. AD surface grid sensitivities projected to normals (right).

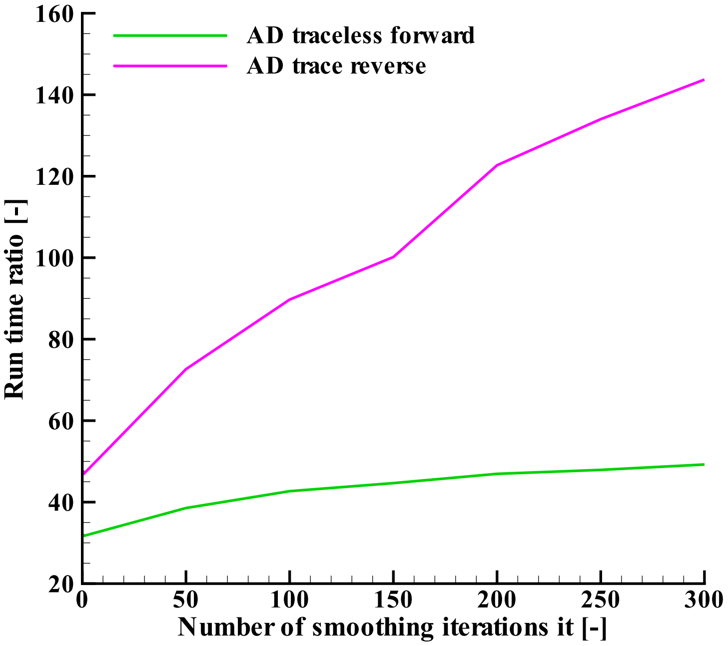
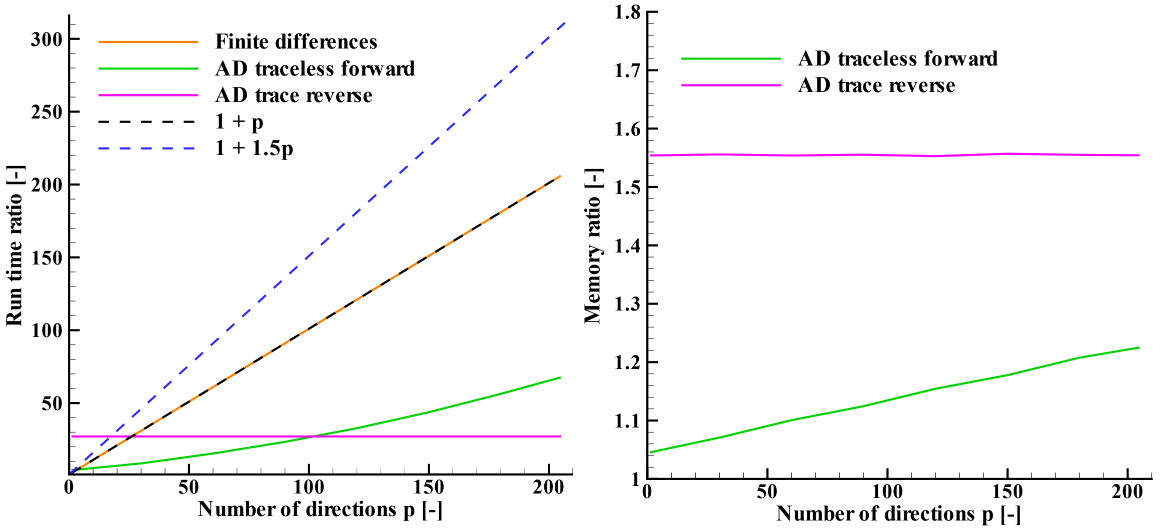


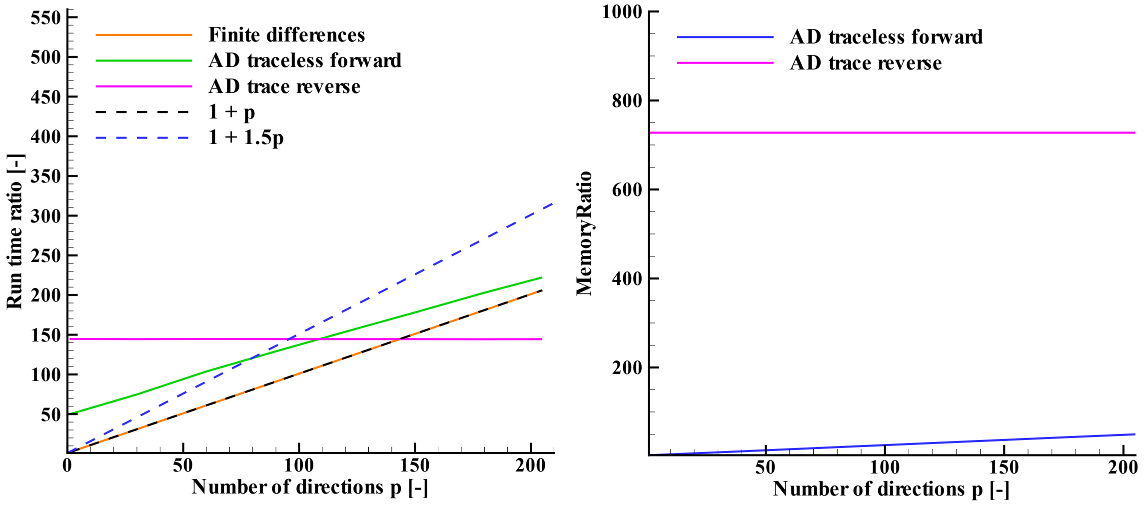
Figure 3.13.: Run time ratio for the creation of the LS89 fluid mesh and evaluation of $dJ_X/d\vec{\alpha}$ gradients with number of directions $p = 1$ vs. the number of smoothing iterations involved during the fluid mesh creation. The results are based on 15 measurements.



(a) Run time ratio (geometry sources)

(b) Memory ratio (geometry sources)

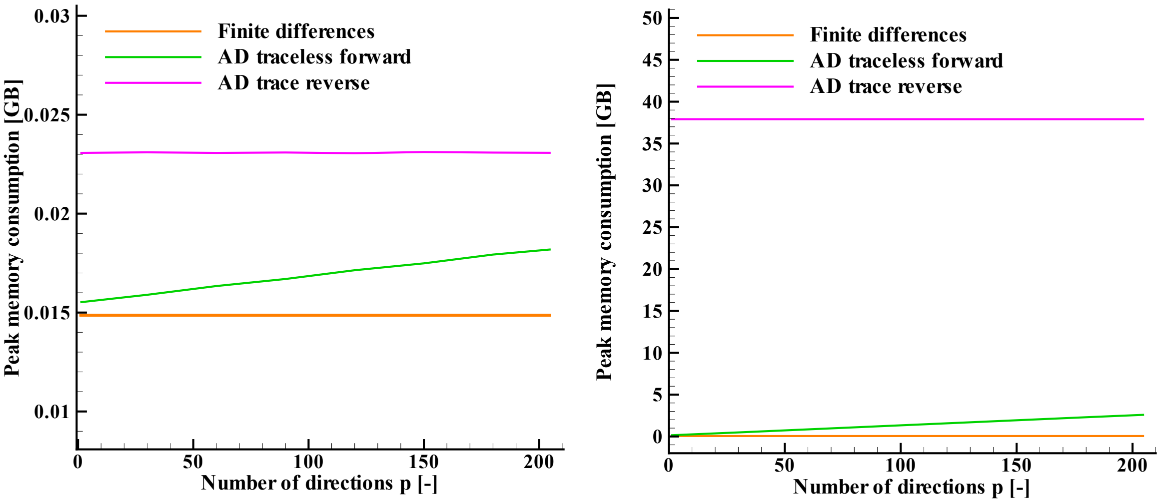
Figure 3.14.: Summary of run time ratio (left) and memory ratio (right) for the creation of the LS89 geometry and evaluation of the $dJ_t/d\vec{\alpha}$ gradients. The results are based on 15 measurements.



(a) Run time ratio (mesh sources)

(b) Memory ratio (mesh sources)

Figure 3.15.: Summary of run time ratio (a) and memory ratio (b) for the creation of the LS89 fluid mesh and evaluation of $dJ_X/d\vec{\alpha}$ gradients. The results are based on 15 measurements.



(a) Geometry sources memory consumption (b) Mesh sources memory consumption

Figure 3.16.: Peak memory consumption for the derivative computation using the geometry sources (a) and the mesh sources (b). The results are based on 15 measurements.

Chapter 4.

Optimization of the LS89 turbine cascade at design point

In the previous chapter, a methodology is presented to integrate the CAD kernel and grid generator within a CFD adjoint-based optimization framework by expressing the optimization problem through CAD parameters, rather than working directly on the CFD grid. In this chapter, the integration of the CAD in the optimization framework is demonstrated by performing a single point optimization of the LS89 turbine cascade. The main contents of this chapter have been published in [120] and they have also been accepted for publication in the *International Journal of Turbomachinery, Propulsion and Power* [121].

Abstract. The LS89 high pressure axial turbine vane was originally designed and optimized for a downstream isentropic Mach number of 0.9. This profile has been widely used for CFD validation in the open literature. This chapter presents a sound methodology to design and optimize the LS89 using Computer Aided Design (CAD) at design conditions. The novelty of the study resides in the parameterization of design space, which is done at the CAD level, and the detailed analysis of the aerodynamic performance of the optimized design. Higher level constraints are imposed on the shape, such as the trailing edge thickness, the axial chord length, and G_2 geometric continuity between the suction side and pressure side at the leading edge. The gradients used for the optimization are obtained by applying Algorithmic Differentiation to both the CAD kernel and grid generator and the discrete adjoint method to the CFD solver. A reduction of almost 12% entropy generation is achieved, which is equivalent to a 16% total pressure loss reduction. The entropy generation is reduced whilst keeping the exit flow angle as constraint, which is enforced via the penalty formulation. The resulting unconstrained optimization problem is solved by the L-BFGS-B algorithm. The flow is governed by the Reynolds-averaged Navier-Stokes equations and the one-equation Spalart-Allmaras turbulence model. The optimal profile is compared and benchmarked against the baseline case.

4.1. Introduction

Designing and optimizing a turbine vane is a complex iterative design process that can take significant time and effort. The use of relatively low-cost numerical shape optimization methods has become more popular and has been widely used in turbomachinery applications. In particular, the adjoint method enables the efficient computation of the sensitivities required by gradient-based optimizers, at a cost independent of the number of design variables [12].

In this chapter, it is proposed to use a CAD-based adjoint-based optimization method for the LS89 [116, 118] high pressure axial turbine nozzle guide vane profile. The LS89 was originally designed and optimized at the Von Karman Institute for Fluid Dynamics for a subsonic isentropic outlet Mach number of 0.9 by the inverse method [116], which is an iterative design method based on both potential and Euler type solvers that uses the difference between the calculated velocity distribution and the required one to modify the profile geometry. Montanelli et. al. [46] performed both single-point and multi-point optimizations on the LS89 to reduce the total pressure losses by constraining the outlet mass flow, whilst keeping the leading and trailing edge geometries and the profile thicknesses fixed. The L-BFGS-B algorithm was used to explore different geometries with a wing parameterization model. They solved the Reynolds-Averaged Navier-Stokes (RANS) equations and the one-equation Spalart-Allmaras turbulence model. The gradients were computed assuming that the eddy viscosity and thermal conductivity are constant. A total pressure loss reduction below 1% was achieved for the nominal case (outlet $M_{ise} = 0.927$). The question arises whether the LS89 is indeed optimal, and cannot be optimized any further. This chapter addresses this and presents an adjoint optimization of the turbine profile by using a novel CAD-based approach in which:

1. The optimal shape remains defined within the CAD tool. The optimization problem herein is expressed by CAD parameters that are directly used in defining the CAD geometry by means of Bézier and B-spline curves.

2. The in-house CAD and grid generation tools are automatically differentiated in forward mode to obtain the exact derivatives of the grid coordinates with respect to the CAD-based design parameters. This allows to accurately predict the sensitivities and circumvent the errors introduced by finite differences.

3. Higher level constraints are imposed on the shape, such as the trailing edge thickness, the axial chord length and the C_2 geometric continuity between the suction side and pressure side at the leading edge.

4. The exit flow angle is considered as an aerodynamic constraint and needs to be kept above a reference value.

A computer aided design and optimization tool for turbomachinery applications (CADO) [69] is used throughout this chapter.

4.2. Methodology

The following subsections present a brief description of the methods employed in this chapter for the geometry parameterization within the CAD, the grid generation, the flow solution, the flow sensitivity aspects and the gradient computation.

4.2.1. CAD based parameterization

The construction of the turbine profile shown in Fig. 3.3 and 3.2, is based on the parameterization described by [119]. The profile is defined by a set of CAD-based or engineering based design parameters that are relevant to the aerodynamic performance (e.g. solidity, stagger angle, etc.) and the manufacturing requirements (e.g. axial chord length, trailing edge radius). First, a camber line is constructed. The points P_{LE}, P_{mid}, P_{TE} define the control points of the 2nd order Bézier curve describing the camber line. The camber line is used to define the position of the control points of the suction side (SS) and pressure side (PS) B-spline curves relative to it. The profile is constructed by two B-spline curves for the SS and PS and a circular arc at the TE to close the profile. The normal distance of the first control point relative to the camber line is calculated in such a way that G_2 geometric continuity (i.e. equal curvature) is maintained between the SS and PS B-splines at the leading edge. Finally, the distance between adjacent blades in the cascade is fixed after specifying the pitch, which can be computed for a given solidity.

4.2.2. Optimization

The purpose of the optimization algorithm (Figure 4.1) is to reduce the entropy generation J_1 (Eq. 4.1) below the baseline value of $J_{1,ref} = 826.73\ Pa \cdot kg^{-1} \cdot m^3$ whilst maintaining the exit flow angle J_2 (Eq. 4.2) above or equal to the baseline value of $J_{2,ref} = 74.83°$, by modifying the design vector $\vec{\alpha}$. The flow state $\vec{U}$ and the design vector $\vec{\alpha}$ are coupled via the primal flow governing equations. The constrained optimization problem is handled with the penalty method. The penalty term becomes active only when the exit flow angle is below the target value. The $J_{2,flag}$ parameter is either set to 1.0 or 0.0 in order to activate or deactivate the penalty term respectively. The weighed cost function of the single point J_{SP} optimization problem is given by Eq. 4.3. The limited memory version of the Broyden-Fletcher-Goldfarb-Shanno algorithm with bounds (L-BFGS-B) [98] available in the python SciPy package [99] is used to find the new design vector.

$$J_1(\vec{\alpha}, \vec{U}) = \frac{\int_{out} p\rho^{1-\gamma} V_x dy}{\dot{m}_{out}} \tag{4.1}$$

$$J_2(\vec{\alpha}, \vec{U}) = atan\left(\frac{\overline{V_y}}{\overline{V_x}}\right)_{out} \tag{4.2}$$

$$J_{SP}(\vec{\alpha}, \vec{U}) = \frac{J_1}{J_{1,ref}} + J_{2,flag} J_{2,coef} \left(\frac{J_2}{J_{2,ref}} - 1\right)^2 \tag{4.3}$$

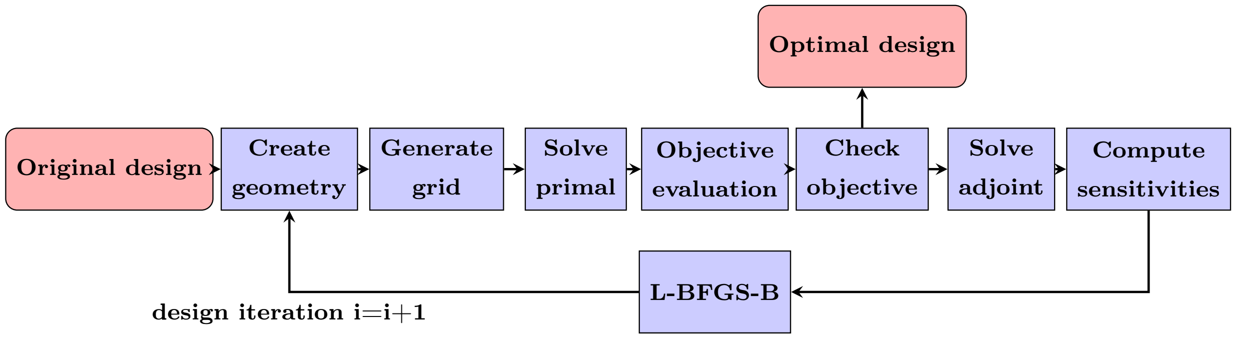


Figure 4.1.: *Optimization flow chart*

4.2.3. Grid generation

A multi-block structured grid is generated for every optimization iteration. A mesh-independence study was carried out in order to select an appropriate mesh for the optimization. The chosen fluid mesh has $300,000$ grid points approximately. One O-grid block is placed around the profile and there are six additional H-grid blocks, four of them distributed around the profile and the remaining two being used as the inlet and outlet blocks. The grid generation method used is described in more detail in Sec. 2.5.

4.2.4. Flow and Adjoint solvers

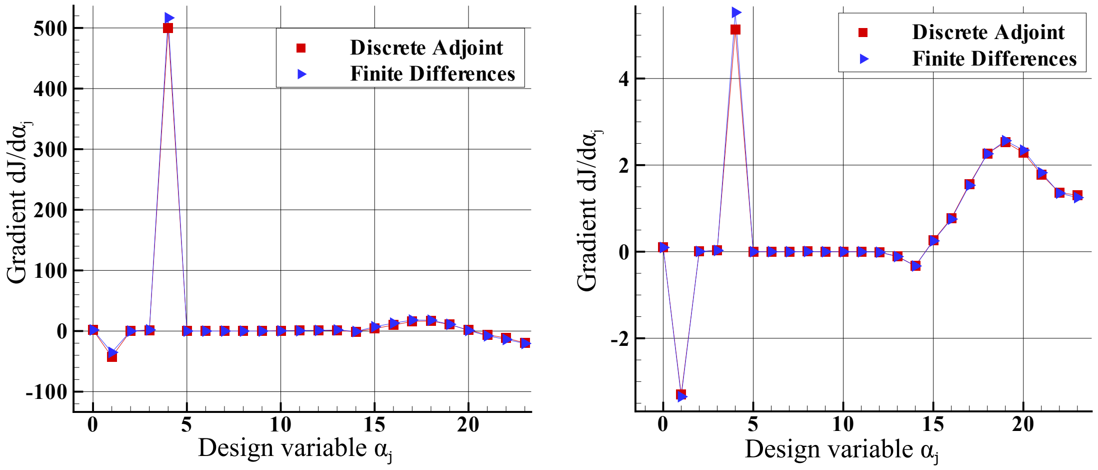
The flow solver employs a cell-centered finite volume discretization on multiblock structured grids. The three dimensional compressible RANS equations are solved with an implicit time integration scheme accelerated by local time-stepping and multigrid. The Spalart-Allmaras turbulence model [93] is used for the turbulence closure problem assuming fully turbulent flow from the inlet ($Re_{inlet} \approx 2 \cdot 10^5$). The boundary conditions are summarized in Table 4.1 and are imposed weakly by utilizing the dummy cell concept [94]. A constant total pressure, total temperature and inlet flow angle are imposed at the inlet and a constant static pressure is imposed at the outlet. The hand derived discrete adjoint solver uses constant eddy viscosity assumption. Further details about the flow and adjoint solvers are described in Sec. 2.7.

Inlet	Outlet	Units
$P_{01} = 147500.0$	$p_2 = 87000.0$	[Pa]
$T_{01} = 420.0$		[K]
$atan\left(\frac{\overline{V_y}}{\overline{V_x}}\right) = 0.0$		[deg]

Table 4.1.: *Boundary conditions*

4.2.5. Gradient computation

The results of differentiating the CAD kernel and grid generation tools using the AD tool ADOL-C [9] are shown in chapter 3. In the work herein, the optimization algorithm computes the volume grid sensitivities $d\vec{X}/d\vec{\alpha}$ for every optimization step, using the forward vector mode as opposed to the forward scalar or reverse modes as it allows to compute $d\vec{X}/d\vec{\alpha}$ in one single evaluation of the primal at a relatively low cost.



(a) Entropy generation (J_1). **(b)** Exit flow angle (J_2).

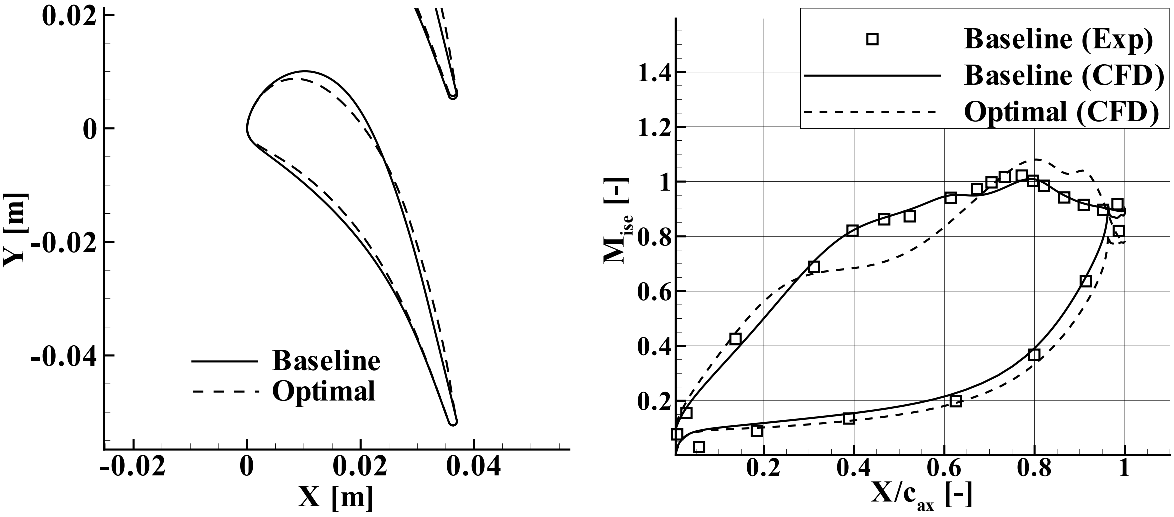
Figure 4.2.: Adjoint vs finite difference gradients.

The performance sensitivities w.r.t the CAD design variables $dJ/d\vec{\alpha}$ for each cost function J_1 or J_2 are computed by doing a scalar product of the performance sensitivities w.r.t. the volume grid $dJ/d\vec{X}$ (ie. the derivative of the cost function with respect to the grid coordinates $\vec{X}$) with the volume grid sensitivities $d\vec{X}/d\vec{\alpha}$. A suitable step-size for each design variable was selected to compute the gradients with finite differences and compare them against the gradients obtained by the discrete adjoint. Figure 4.2a and 4.2b show a good agreement with finite differences for both the entropy generation and exit flow angle. The largest performance sensitivities w.r.t. the CAD design variables shown in these figures correspond to the design variables $\alpha_j = 1$ (c_{ax}) and 4 (R_{TE}), which do not change during the optimization because they are considered as manufacturing constraints.

4.3. Results

Figure 4.3a shows significant shape differences between the baseline and the optimal shape, the later being a more aft-loaded profile than the baseline design. The main geometry differences between the optimal and baseline profiles are summarised in Table 4.2. The isentropic Mach number distributions are shown in Figure 4.3b, which shows a fairly good agreement between the M_{ise} predicted by CFD and the experimental data (MUR45 test condition, with downstream M_{ise} = 0.875, see [118]) for the baseline geometry. The density, static pressure, temperature, Mach and entropy generation contours for the baseline and optimal shapes are shown in Sec. B.1 and B.2.1 respectively.

The L-BFGS-B algorithm converges within 20 optimization iterations. More cycles were performed but no further decrease in the objective function was obtained. Despite the fact that the mass flow for the optimal geometry has increased slightly by 1.2%, the entropy generation is reduced by 11.6%, whilst maintaining the exit flow angle within ±0.1% of the baseline value. As the aft-loaded profile has significant rear suction side curvature, the flow is rapidly accelerated towards a higher peak Mach number than the baseline design. This is followed by a deceleration from X/c_{ax}



(a) Baseline and optimal profiles.

(b) Isentropic Mach number comparison.

Figure 4.3.: Baseline and optimal profiles and isentropic Mach number comparison.

	Acronyms	Units	Baseline	Optimal (Variation)
Rear suction side turning	ϵ_{SS}	[deg]	11.6	18.37 (6.76°)
Pressure side trailing edge wedge angle	φ_{PS}	[deg]	2.500	2.506 (0.006°)
Suction side trailing edge wedge angle	φ_{SS}	[deg]	4.000	4.001 (0.001°)
Chord	c	[mm]	64.310	64.293 (0.03%)
Stagger angle	γ	[deg]	54.925	54.914 (−0.011°)
Inlet metal angle	β_{in}	[deg]	0.0	0.001 (0.0010°)
Outlet metal angle	β_{out}	[deg]	74.000	74.006 (0.006°)
Leading edge radius	R_{LE}	[mm]	4.126	3.959 (-4.06%)
Solidity	σ	[-]	1.118	1.108 (-0.93%)
Pitch	g	[mm]	57.500	58.021 (0.91%)
Pitch/chord	g/c	[-]	0.894	0.902 (0.93%)
Trailing edge thickness/throat height	$2R_{TE}$/t	[-]	0.09133	0.08644 (-5.35%)

Table 4.2.: *Geometry parameters*

$= 0.8$ to 0.88 and a small acceleration from $X/c_{ax} = 0.88$ to 0.91. At $X/c_{ax} = 0.91$ the flow is rapidly decelerated again and leaves the trailing edge at a significantly lower exit isentropic Mach number. The strong deceleration towards the end of the suction side suggests that the boundary layer will be thicker and one would expect higher profile losses.

The total pressure loss reduction between the inlet and the fully mixed-out flow at the outlet plane, expressed via $P_{01} - P_{02}$ or by the ζ_2 coefficient (see Eq. 4.4), is of the order of 16.3%. The downstream total pressure loss profile, at the plane X/c_{ax} = 1.433, is shown in Fig. 4.4. In the vertical axis the total pressure loss $P_{01} - P_{0X}$ between the inlet and a downstream plane at x/c_{ax} = 1.433 is expressed as a a percentage of the downstream dynamic head q_X at the X/c_{ax} = 1.433 plane. This figure shows that the total pressure loss generated in the wake is reduced significantly.

$$\zeta_2 = 1 - \frac{1 - (\frac{p_2}{P_{02}})^{\frac{\gamma-1}{\gamma}}}{1 - (\frac{p_2}{P_{01}})^{\frac{\gamma-1}{\gamma}}} \tag{4.4}$$

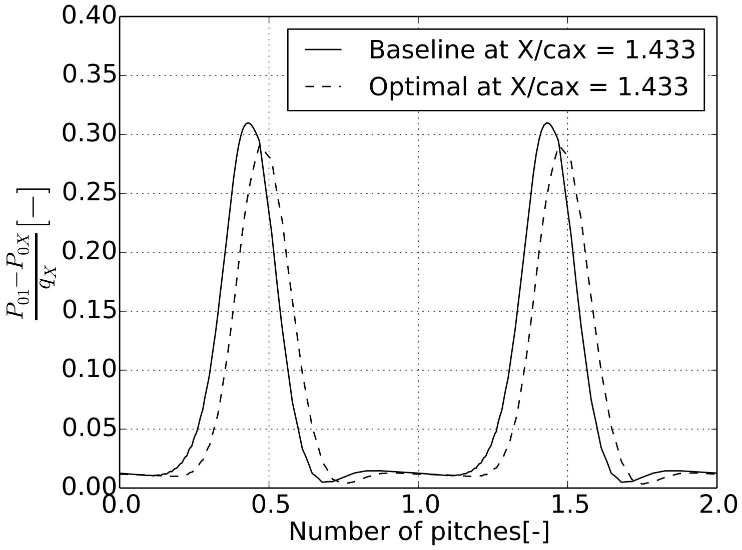


Figure 4.4.: *Total pressure loss downstream at* $X/c_{ax} = 1.433$

In order to understand why the optimal shape is more efficient than the baseline, the following subsections will look into the loading, boundary layer parameters, base pressure and profile losses for each profile.

4.3.1. Zweifel loading coefficient

The total loading, which can be expressed as the integral of the $p_{PS}-p_{SS}$ over the axial chord length, has increased by 1% for the optimal shape as a result of the 1% increase in the pitch and having the same exit flow angle. The Zweifel loading coefficient Z_w [122] is a measure of how efficiently the profile is loaded. An efficient loaded profile would be one in which the pressure is equal to the stagnation pressure over the pressure side and the velocity on the suction side is constant and equal to the downstream velocity. The Zweifel's design rule says that loss is minimized for $0.8 < Z_w < 1$. Equation 4.5 was used to calculate the Zweifel loading coefficient, which increased from 0.636 for the baseline to 0.646 for the optimal.

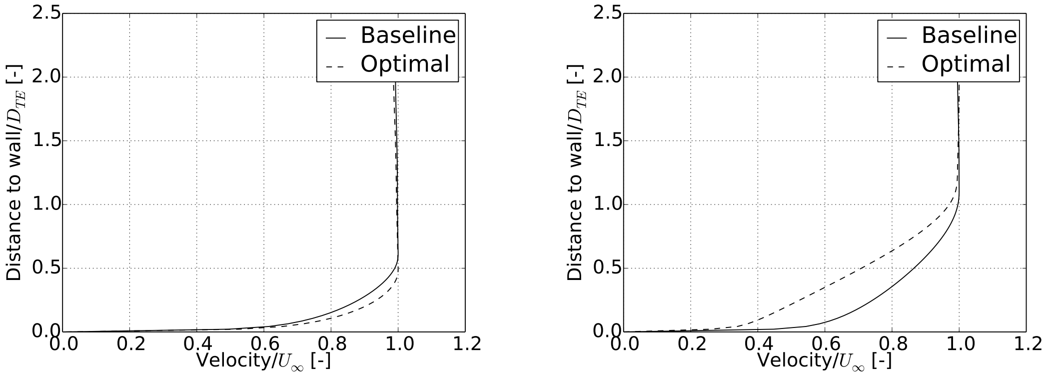
$$Z_w = \frac{\int_0^1 (p_{PS} - p_{SS}) \frac{dx}{c_{ax}}}{(P_{01} - p_2) c_{ax}} \tag{4.5}$$

4.3.2. Boundary layer parameters

The state of the boundary layer close to the trailing edge is deemed to have an impact on the profile losses (see Equation 4.6). Table 4.3 compares the boundary layer parameters between the baseline and optimal shapes for two locations on the suction side: at the peak M_{ise} location and very close to the TE (approximately at $x = 0.03m$ and $x = 0.037m$ respectively, see Figure 4.3b). The latter location is defined herein as the point of the suction side slightly upstream of the TE circle. Some of the boundary layer parameters shown in Table 4.3 are non-dimensionalized with the trailing edge thickness $D_{TE} = 2R_{TE}$. The boundary layer velocity profiles are shown in Figures 4.5a and 4.5b for both locations, showing that there is no boundary layer separation.

At the peak M_{ise} location, the boundary layer thickness δ for the optimal design is smaller than for the baseline, as expected due to the higher acceleration. When the flow decelerates from the peak M_{ise} location to the point just upstream of the trailing edge circle, the boundary layer thickness almost doubles for the baseline and triples for the optimal profile. The resulting δ of the optimal shape is 18% thicker than the baseline. The shape factor H is higher for the optimal than for the baseline design in both locations, which means that the optimal profile has stronger adverse pressure gradient. The shape factor for both designs are between the typical laminar Blasius boundary layer value ($H = 2.59$) and the turbulent values ($H = 1.3 - 1.5$).

From a boundary layer perspective, the optimal design would have higher profile losses due to the higher thickness displacement and momentum thickness. However, there is at least one more factor to be considered when determining the profile losses: the base pressure.



(a) Suction side at peak Mach number location. **(b)** Suction side close to the TE.

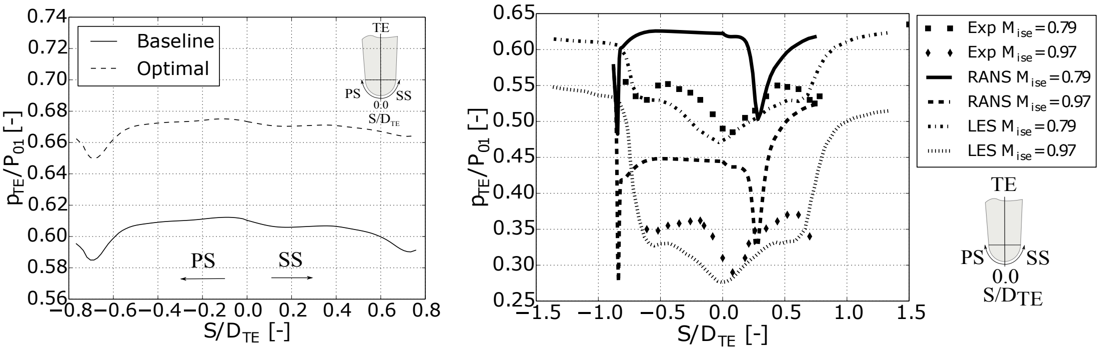
Figure 4.5.: Boundary layer profiles on the suction side.

	Units	Baseline at $M_{ise,max}$	Optimal at $M_{ise,max}$	Baseline at $X_{SS,TE}$	Optimal at $X_{SS,TE}$
δ/D_{TE}	[-]	0.501	0.390	0.939	1.11
δ^*/D_{TE}	[-]	0.104	0.079	0.206	0.346
θ/D_{TE}	[-]	0.0557	0.0405	0.1186	0.1780
H	[-]	1.86	1.96	1.74	1.95

Table 4.3.: *Boundary layer parameters*

Base pressure

The base pressure, which is the static pressure at the mid point of the trailing edge is known to play an important role in the profile losses. Higher base pressures contribute to the reduction of the profile losses. The trailing edge pressure distribution shown in Figure 4.6a shows that the optimal shape has got 10% higher base pressure. It is generally thought that higher values of boundary layer thickness result in higher values of base pressure [123]. Also, higher base pressures are expected of an aft-loaded profile with significant rear suction side curvature and higher values of unguided or uncovered turning [124]. The angle between tangents to blade suction side in the throat and at the TE increased by 6.76 degrees when going from the baseline to the aft-loaded optimal profile. According to the base pressure correlation given by [124], the optimal profile is expected to have 3% higher base pressure than the baseline. The percentage increase predicted by the correlation is mainly due to the higher suction side rear curvature because the trailing edge wedge angle difference between the baseline and optimal profiles is negligible. However, [124] showed that, for 80% of the tested blades, there was a discrepancy between the predicted base pressures (using the correlation) and the measured ones of the order of +/-5% of the downstream pressure. Although there are no measurements available for the optimal profile yet, by taking this error into account, the minimum and maximum values for the base pressure of the optimal profile can be estimated to be 5% lower and 16.7% higher than the baseline respectively. Despite the 10% higher base pressure predicted by RANS falls inside this range, and is in line with what would be expected of an aft-loaded profile with significantly higher rear suction side curvature [124], only unsteady analysis and experimental tests will be able to provide enough evidence to support these results.



(a) Optimal vs. Baseline trailing edge pressure distributions.

(b) Trailing edge pressure distribution with the blade experimentally tested by [125]. LES results are from [126].

Figure 4.6.: Trailing edge (TE) pressure distributions over the pressure side (PS) and suction side (SS). S is the arc-length distance from the TE mid-point and D_{TE} is the trailing edge thickness ($D_{TE} = 2R_{TE}$).

Predicting the absolute values of the base pressure correctly can be very difficult due to the highly complex nature of the flow in the trailing edge. Vagnoli et. al. [126] showed that steady-state simulations usually predict the wrong shape or distribution of pressure around the trailing edge, whereas LES or URANS capture much better the shape and the pressure values when compared to experimental data. The pressure distribution shown in Figure 4.6a is a nearly-uniform pressure around the trailing edge, which is suspected to be different to the real pressure distribution profile. However, it is believed that the base pressure delta difference between two steady-state simulations, one for the optimal and the other for the baseline profiles, is a fairly good estimate of the real difference. The validity of this assumption has been investigated by performing two RANS simulations for the turbine profile that was tested by [125] at $M_{ise} = 0.79$ and 0.97, and comparing the base pressure delta between these two operating conditions against the experimental data and the LES results performed by [126] (Figure 4.6b). The base pressure delta, which is taken at $S/D_{TE} = 0$ from Figure 4.6b, is 0.18 for both the experiments and RANS and 0.20 for LES. This confirms that, despite the limitations of CFD in predicting the base pressure averaged values, RANS is able to capture very well the relative differences in base pressure between two operating conditions.

4.3.3. Profile losses

The profile losses for incompressible flow can be related to the base pressure coefficient and the boundary layer parameters [127], as shown in Equation 4.6. The relative effect of the base pressure coefficient, which is computed with Eq. 4.7, is increased in compressible flow [128]. For blades operating in the transonic range from 0.8 to 1.2, the major source of loss is the trailing edge loss at these speeds, and the base pressure plays a dominant role in determining the loss [123]. The profile loss split is shown in Table 4.4. The higher δ^* and θ values of the optimal shape contribute to increasing the profile loss. However, the reduction in profile losses due to the 302.2% increase in the trailing edge loss term dominates and explains the overall 30% reduction in profile losses. However, if the base pressure increase was equal or below 6.54%, the optimal shape would have similar or higher profile losses than the baseline, respectively. The question arises whether using RANS for the optimization of an axial turbine operating at transonic speeds is a valid approach, since the base pressure plays such a dominant effect on the profile losses for such speeds, and the flow field around the TE of a transonic turbine blade is too complex to be captured by RANS. Future LES simulations and or experimental test are going to be necessary to support or reject the aerodynamic improvements reported herein for the optimal shape.

$$\zeta_p = \frac{-C_b D_{TE}}{t} + 2\frac{\theta_{SS} + \theta_{PS}}{t} + \left(\frac{\delta^*_{SS} + \delta^*_{PS} + D_{TE}}{t}\right)^2 \tag{4.6}$$

$$C_b = \frac{p_b - p_2}{\frac{1}{2}\rho V_2^2} \tag{4.7}$$

4.3.4. Off-Design Performance

The 16% reduction in total pressure loss shown in Table 4.5 was achieved for an outlet isentropic Mach number of 0.9 while keeping the axial chord length and trailing edge radius fixed and exit flow angle above 74.83°. However, the performance of the optimal profile is deemed to deteriorate significantly for higher outlet isentropic Mach numbers. Figure 4.7 shows the total pressure loss coefficient for different outlet isentropic Mach numbers for the baseline and optimal profiles. The total pressure

	Units	Baseline	Optimal	Variation
$\frac{-C_b D_{TE}}{t}$	[-]	-0.0016	-0.0065	302.2%
$2\frac{\theta_{SS}+\theta_{PS}}{t}$	[-]	0.0066	0.0095	43.5%
$\left(\frac{\delta^*_{SS}+\delta^*_{PS}+D_{TE}}{t}\right)^2$	[-]	0.0010	0.0012	21%
ζ_p	[-]	0.0060	0.0042	-30%

Table 4.4.: *Profile loss contribution for the baseline and optimal*

	Acronyms	Units	Baseline	Optimal	Variation
Entropy generation	J_1	[Pa/(kg/m^3)]	826.7	731.0	-11.6%
Exit flow angle	J_2	[deg]	74.89	74.89	-0.01%
Mass flow	$\dot{m}$	[kg/s]	0.008	0.009	+1.2%
Total pressure losses	$P_{01} - P_{02}$	[kPa]	2.47	2.07	-16.3%
Downstream loss coeff	ζ_2	[-]	0.02986	0.02500	-16.3%
Profile losses	ζ_p	[-]	0.0060	0.0042	-30%
Zweifel coefficient	Z_w	[-]	0.67	0.81	+21.2%
Solidity	σ	[-]	1.118	1.108	-0.9%
Pitch	g	[m]	0.0575	0.0580	+0.9%

Table 4.5.: *Comparison between the baseline and optimal*

loss in Fig. 4.7 is defined as the total pressure difference between the inlet and outlet divided by the dynamic head at the outlet plane. At off-design conditions, the aerodynamic improvements of the optimal design are reduced as the downstream isentropic Mach number is increased. Beyond $M_{ise,2} = 0.94$, the baseline would have lower total pressure losses than the optimal profile. A detailed aerodynamic performance study of the baseline and optimal geometries at off-design conditions is outside the scope of this chapter but can be found in the chapter 5, where the performance of the LS89 is significantly improved at off-design conditions.

4.3.5. Comparison with previous studies

Previous optimization studies use different approaches, settings, boundary conditions, etc., which makes it very hard to compare this work with others. For example, the main differences with the study performed by Montanelli et al. [46] are the following:

1. Flow conditions. The single point optimization performed by Montanelli et al. [46] was performed at the MUR235 conditions reported in [118], which has slightly different flow conditions than those used in this study (donwnstream $M_{ise} = 0.927$ and 0.90, for MUR235 and this study respectively). They also performed five single point optimizations at higher and lower pressure ratios but the total pressure loss reduction was below 3% in the best scenario. However, it is suspected that the difference in flow conditions might not be the main reason to justify the differences in performance improvements observed between the optimization study in [46] and the current study.

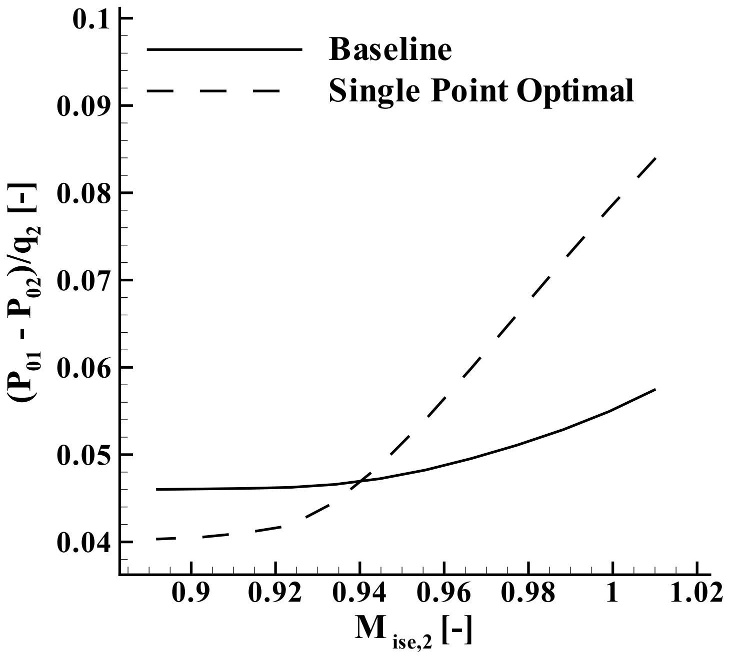


Figure 4.7.: *Total pressure loss coefficient for different downstream* $M_{ise,2}$.

2. Optimization problem definition. Some differences are expected due to the different statement of the optimization problem. Montanelli et al. [46] reduced the total pressure losses whilst keeping the outlet mass flow as an aerodynamic constraint (to avoid reducing the entropy due to a mass flow reduction). Also, it was assumed that the exit flow angle would remain unchanged by freezing the trailing edge geometry. In the present study the entropy generation is reduced whilst keeping the exit flow angle as a flow constraint. Therefore, the objective space is different between the two studies.

3. CAD-parameterization and design space. The design space is mainly defined by all the possible shapes that can be created by the chosen CAD-parameterization. Both [46] and the current study use different parameterizations of the LS89 but they use a similar number of active design variables during the optimization. However, in the current study the trailing edge thickness and the axial chord length are kept fixed as manufacturing constraints, whereas [46] kept also frozen the profile thicknesses and the leading edge geometry, in order to satisfy possible structural constraints and avoid G_2 discontinuities at the LE, respectively. This means that the CAD-parameterization used herein should have a richer design space than the one used by [46].

4. Grid update. The mesh deformation approach to update the grid for a new set of design variables was used in [46]. This approach computes a surface deformation field by subtracting the initial surface to the updated surface and propagates it inside the volume grid. Typically, the weakness of the mesh deformation approach is that it can fail to produce a good mesh for large changes in the CAD-geometry. In contrast, re-meshing is the approach employed herein to update the grid for every new set of design variables. The L-BFGS-B algorithm requires

the user to provide certain bounds for the design variables. This stops the algorithm from exploring non-possible designs which could make the mesh generation/deformation step fail inside the optimization procedure. Although the bounds have not been reported by neither [46] nor this study, it is suspected that the mesh deformation approach might be more sensitive to fail for large geometry variations than the mesh generation approach. There is the possibility that the bounds given to the L-BFGS-B algorithm by [46] might have been more restrictive than those used herein and the optimizer might have explored a smaller design space.

4.4. Conclusions

This chapter presents a single point optimization of the LS89 axial turbine cascade vane profile for the design downstream isentropic Mach number of 0.9. The parameterization is done at the CAD level, which allows to impose higher level constraints on the shape, such as the axial chord length, the trailing edge radius, and the C_2 geometric continuity between the suction side and pressure side at the leading edge. Additionally, the adjoint sensitivities are filtered out and only smooth shapes are produced. The use of algorithmic differentiation for the CAD kernel and grid generator allows computing the grid sensitivities to machine accuracy, avoiding the limited arithmetic precision and the truncation error of finite differences. The optimization results show that the total pressure losses and entropy generation can be reduced by 16% and nearly 12% respectively by going from a front to a rear loaded profile and by keeping the exit flow angle fixed. Despite the negative effect of the increase in boundary layer thickness displacement and momentum thickness on the profile losses, the base pressure coefficient plays a dominant role herein and reduces the profile losses by 30%. The substantial aerodynamic improvements reported herein have not been observed previously, probably due to the fact that the current CAD-based parameterization allowed the exploration of a richer design space. The relative delta in base pressure predicted by the steady simulation between the optimal and baseline profiles is deemed to be representative of the real value, as RANS tends to capture well the relative differences between two designs. However, unsteady studies or experimental investigations would be beneficial in order to confirm these findings. Finally, multi-point optimization studies would also be beneficial to improve performance of the turbine cascade at off-design conditions.

Chapter 5.

Multipoint optimization of the LS89 turbine cascade

5.1. Abstract

In the previous chapter, the integration of the CAD in a CFD adjoint optimization framework is demonstrated by performing a single point optimization of the LS89 turbine cascade. Large aerodynamic improvements were reported at design point, but the performance of the turbine cascade deteriorates significantly for higher outlet isentropic Mach numbers. This chapter builds up on the results presented by the previous chapter by extending the study to a multi-point optimization, in order to improve the off-design performance of the turbine cascade. The main contents of this chapter have been published in [129] and in the Springer book [130].

Abstract. A computer-aided design (CAD) and adjoint based multipoint optimization of the LS89 high pressure axial turbine vane is presented in this chapter. The aim is to reduce the entropy generation at both subsonic and transonic flow conditions by means of employing CAD and adjoint based methods during the optimization process. The performance metrics at design and off-design conditions are grouped into a single objective function using equal weights. A steady-state Reynolds-Averaged density based Navier-Stokes solver and the one-equation Spalart-Allmaras turbulence model are used to predict the losses. The entropy generation is reduced whilst keeping the trailing edge thickness and the axial chord length as manufacturing constraints and the exit flow angle as a flow constraint, which is enforced via the penalty formulation. The resulting unconstrained optimization problem is solved by a L-BFGS-B algorithm. At every optimization iteration a new profile is constructed using B-splines and the grid is rebuilt by elliptic grid generation. The gradients used for the optimization are obtained via a novel approach in which both the CAD kernel and grid generation are differentiated using Algorithmic Differentiation techniques. The sensitivities of the objective function with respect to the grid coordinates are computed by a hand-derived adjoint solver. The off-design performance of the LS89 is significantly improved and the optimal geometry is analyzed in more detail.

5.2. Introduction

A CAD and adjoint based approach was used in [46] to perform a single and multi-point optimization of the LS89 to reduce the total pressure losses by constraining the outlet mass flow. However, the shape changes were constrained by keeping the leading and trailing edge geometries and the profile thicknesses fixed. Too many geometry constraints can reduce the ability of the optimizer to find superior designs. In chapter 4, substantial aerodynamic improvements were achieved at design point by means of using a different parameterization and possibly richer design space because the profile thicknesses and the leading edge geometry were free to change. A 16 % reduction in total pressure loss was achieved whilst keeping the axial chord length, trailing edge radius and exit flow angle fixed. However, the performance of the optimal profile deteriorates significantly at higher outlet isentropic Mach numbers. Given that a turbomachine typically operates over a range of different aerodynamic conditions, this chapter aims at optimizing the performance of the LS89 axial turbine profile at off-design conditions. The design optimization is carried out by using a CAD and adjoint based approach. In this chapter, a combination of forward AD and hand-derived reverse differentiation is used for the grid generation and the flow solver respectively.

5.3. Methodology

The same optimization process used in chapter 4 (see Fig. 4.1) is used in this chapter to optimize the turbine cascade at off-design conditions. The difference lies in the computation of the cost function and the gradients. Some of the individual components of the optimization flow chart will be discussed in more detail in this section.

The CAD-based parameterization and the geometry construction is explained in Sec. 4.2.1. The grid generation process and the details of the flow and adjoint solvers can be found in Sec. 4.2.3 and 4.2.4 respectively. The boundary conditions are summarized in Table 5.1, which shows two additional operating points than in Table 4.1 because, in this chapter, it is desired to optimize the performance of the LS89 at a wider range of operating conditions. A constant total pressure, total temperature and inlet flow angle are imposed at the inlet and a constant static pressure is imposed at the outlet.

Inlet	Outlet $op = 1$	Outlet $op = 2$	Outlet $op = 3$	Units
$P_{01} = 147500.0$	$p_2 = 87000.0$	$p_2 = 82050.0$	$p_2 = 77000.0$	[Pa]
$T_{01} = 420.0$				[K]
$atan\left(\frac{V_y}{V_x}\right) = 0.0$				[deg]

Table 5.1.: *Boundary conditions for the different operating points (op)*

After solving the primal solver, a performance metric that is proportional to the entropy generation J_1 and the exit flow angle J_2 at the outlet of the domain can be computed for each operating condition with expressions 5.1 and 5.2 respectively.

$$J_1 = \frac{\int_{out} p\rho^{1-\gamma} V_x dy}{\dot{m}_{out}} \tag{5.1}$$

$$J_2 = atan\left(\frac{\overline{V_y}}{\overline{V_x}}\right)_{out} \tag{5.2}$$

After non-dimensionalizing them with $J_{1,ref} = 826.73\ Pa \cdot kg^{-1} \cdot m^3$ and $J_{2,ref} = 74.83°$ they are combined into a pseudo cost function via the penalty term method (expression 5.3), which transforms the constrained problem in an unconstrained optimization formulation.

$$J_{MP} = \sum_{op=1}^{3} \frac{1}{3}\left(\frac{J_{1_{op}}}{J_{1,ref}}\right) + \omega\left(\frac{J_{2_{op3}}}{J_{2,ref}} - 1\right)^2 \tag{5.3}$$

The left term of the pseudo cost function is a weighted average of the non-dimensional entropy generation, which is computed with equal weights for the following operating points (op): $M_{ise,2} = 0.9$ (op=1), $M_{ise,2} = 0.955$ (op=2) and $M_{ise,2} = 1.01$ (op=3). By grouping the performance metrics at design and off-design conditions into one single objective function, the multipoint objective optimization can be treated as a single objective one. The right term of expression 5.3 is the penalty term, which becomes larger the more the exit flow angle deviates from the target value. In order to reduce the computational effort, the penalty term is computed only for the 1.01 Mach transonic operating point, which is the most challenging operating point when it comes to satisfy the aerodynamic constraint. It is assumed that the flow turning at $M_{ise,2} = 1.01$ is always smaller than at $M_{ise,2} = 0.9$, which is later validated on the baseline and optimized profiles. The ω is the penalty coefficient and was selected by trial and error.

After evaluating the cost function, it is necessary to compute the gradients. The adjoint solver computes the gradients of the cost function J (e.g., entropy generation or exit flow angle) with respect to the volume grid point coordinates $\vec{X}$ (i.e., $dJ/d\vec{X}$). Similarly to the flow solver, the adjoint solver uses the same stabilization JT-KIRK scheme of [97]. The hand derived discrete adjoint solver assumes that the eddy viscosity does not change with geometry variations (frozen turbulence). The computation of $dJ/d\vec{X}$ is explained in more detail in Sec. 2.2.5.

Next, the performance sensitivities w.r.t the CAD parameters ($dJ/d\vec{\alpha}$) for each cost function J are computed by a scalar product of the performance sensitivities w.r.t the volume grid ($dJ/d\vec{X}$) with the volume grid sensitivities ($d\vec{X}/d\vec{\alpha}$) as follows:

$$\frac{dJ}{d\vec{\alpha}} = \frac{dJ}{d\vec{X}}\frac{d\vec{X}}{d\vec{\alpha}}. \tag{5.4}$$

In this work, the AD tool ADOL-C is used to compute the volume grid sensitivities $d\vec{X}/d\vec{\alpha}$

in one single evaluation of the primal at a relatively low cost by using the forward vector mode approach. The $dJ/d\vec{\alpha}$ gradients were also computed by finite differences and compared against the ones obtained with the method described in this study, showing a reasonably good agreement (Figures 4.2a and 4.2b).

Finally, the gradients are given to the Quasi-Newton L-BFGS-B algorithm [98], which is available in the python SciPy package [99], and it is used to find the new design vector.

5.4. Results

The optimizer performed a total number of 30 iterations, from which 12 of them were line search iterations in order to find an appropriate step size. Figure 5.1 shows the evolution of the non-dimensional cost functions during the optimization process after excluding the line search iterations. The green line represents the evolution of the first term of the pseudo cost function (expression 5.3), which has been reduced by 6.4% whilst satisfying the exit flow angle aerodynamic constraint (i.e. $J_{2_{op3}}/J_{2,ref} \geqslant 1.0$), which is indicated by the line with discrete points. The evolution of the non-dimensional entropy generation for each operating point is also shown on Fig. 5.1. The optimizer was able to improve the aerodynamic performance of the profile for each operating point, but the largest improvements were made for the operating point with a downstream isentropic Mach number of 1.01.

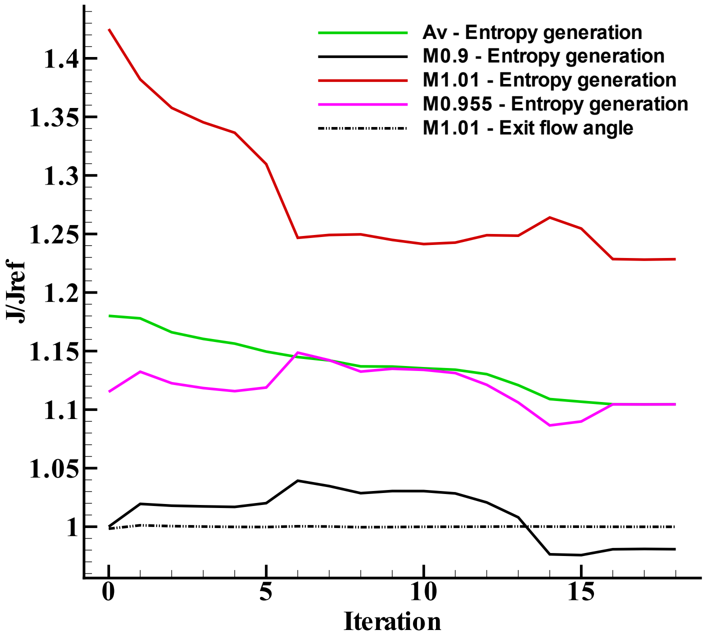


Figure 5.1.: *Cost functions evolution*

Figure 5.2 compares the baseline and the multipoint optimal profiles and Table 5.2 summarizes the main geometrical changes. The multipoint optimization results show that the differences between the baseline and the optimal shapes are much smaller than those observed for the single point optimization in the previous chapter (see Fig. 4.3a and Tab. 4.2). For instance, the pitch of the single point optimal profile was increased by 0.91 % but only 0.32% for the multipoint optimal profile. The leading edge radius was reduced by 4.06% for the single point optimal shape but only 0.71% for the multipoint optimal shape. Also, the rear suction side turning, which is defined here as the angle between a line tangent to the geometric throat and a line tangent to the start of the trailing edge circle at the suction side, increased by 6.76% for the single point optimal shape but as little as 0.35% for the multipoint optimal shape. However, the relatively small geometry changes have quite a substantial effect on the aerodynamic performance as it will be shown in the following sections.

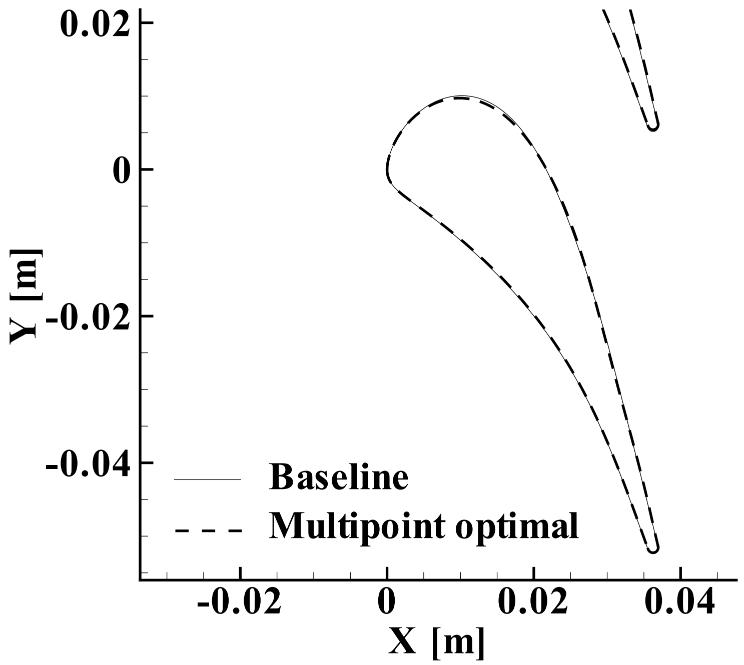


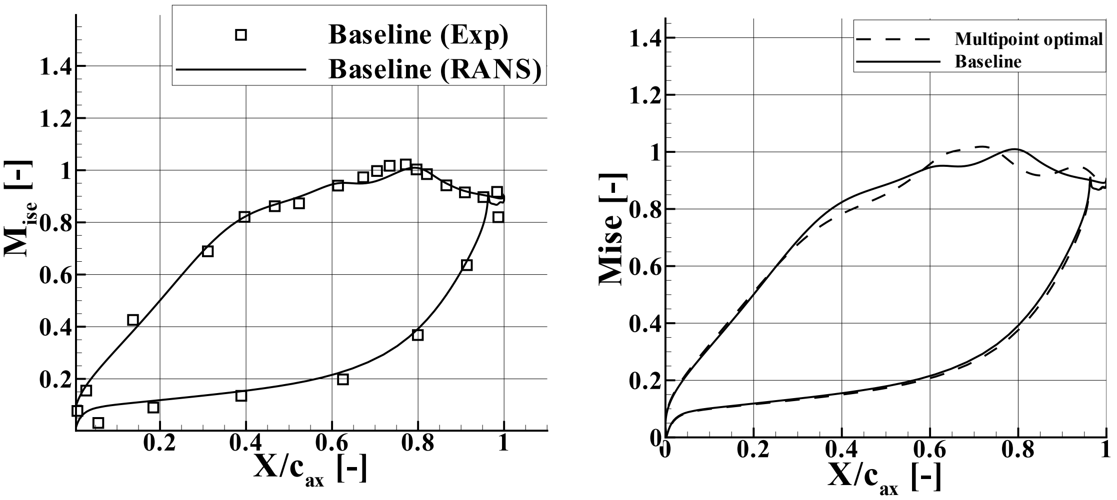
Figure 5.2.: *Geometry comparison of the baseline and multipoint optimal profiles*

	Acronyms	Units	Baseline	Multipoint optimal	Variation
Rear suction side turning	ϵ_{SS}	[deg]	11.6	13.0	1.4°
Pressure side trailing edge wedge angle	φ_{PS}	[deg]	2.500	2.498	−0.0022 °
Suction side trailing edge wedge angle	φ_{SS}	[deg]	4.000	4.002	0.002 °
Chord	c	[mm]	64.310	64.254	-0.09 %
Stagger angle	γ	[deg]	54.925	54.890	−0.035 °
Inlet metal angle	β_{in}	[deg]	0.0	0.0032	0.0032 °
Outlet metal angle	β_{out}	[deg]	74.000	73.981	−0.0190 °
Leading edge radius	R_{LE}	[mm]	4.126	4.097	-0.71 %
Solidity	σ	[-]	1.118	1.114	-0.40 %
Pitch	g	[mm]	57.500	57.683	0.32 %
Pitch/chord	g/c	[-]	0.894	0.898	0.41 %
Trailing edge thickness/throat height	D_{TE}/t	[-]	0.09133	0.09131	-0.02%

Table 5.2.: *Comparison of the main geometrical changes between the baseline and multipoint optimal profiles*

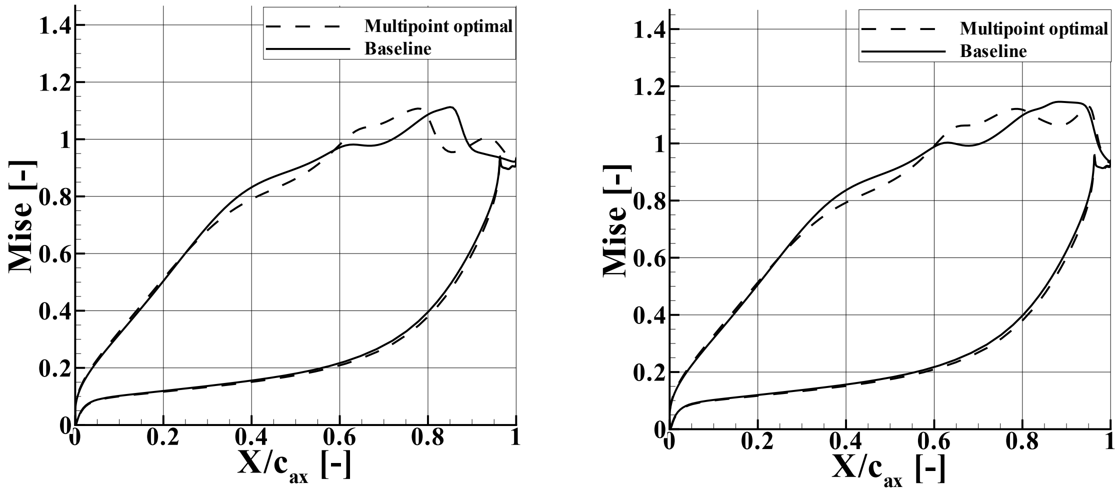
5.4.1. Isentropic blade Mach number comparison

Figure 5.3a shows a reasonably good agreement between the CFD prediction of the baseline isentropic blade Mach number and the experimental data at MUR45 test conditions [118]. Figures 5.3b, 5.3c and 5.3d compare the isentropic Mach number distribution of the baseline and the multipoint optimal profiles for a downstream isentropic Mach number of 0.9, 0.955 and 1.01 respectively. Figure 5.3d shows that a shock is located approximately at X/c_{ax} = 0.95 for the baseline. The optimizer was able to reduce slightly the suction side peak Mach number from 1.145 to 1.129 and moved the peak Mach number location from X/c_{ax} = 0.9 to X/c_{ax} = 0.95. The density, static pressure, temperature, Mach and entropy generation contours for the baseline and optimal shapes are shown in Sec. B.1 and B.2.2 respectively.



(a) Baseline RANS results vs experimental data (MUR45)

(b) Isentropic Mach number for $M_{ise,2}$ = 0.9



(c) Isentropic Mach number for $M_{ise,2}$ = 0.955

(d) Isentropic Mach number for $M_{ise,2}$ = 1.01

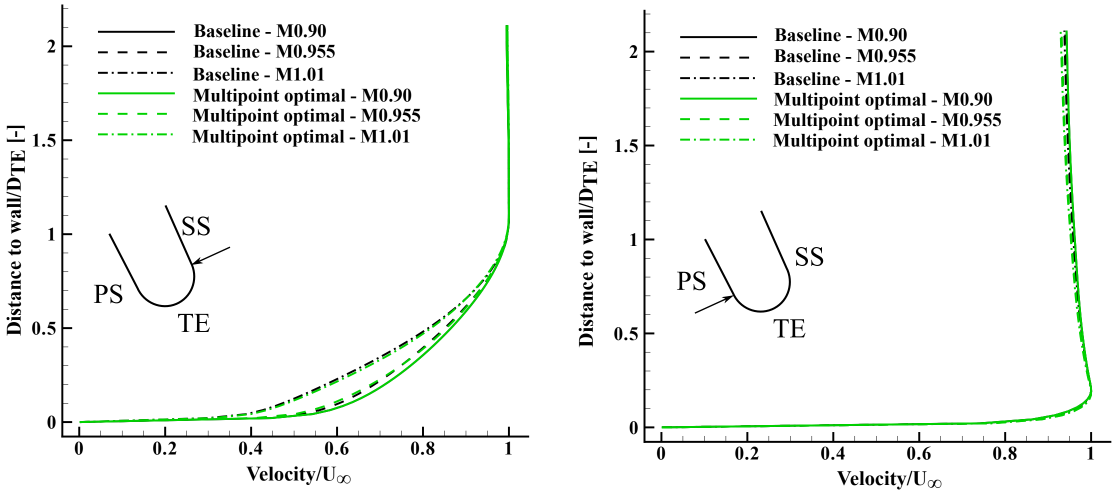
Figure 5.3.: *CFD validation (a) and isentropic Mach number plots for downstream* $M_{ise,2}$ = 0.9 *(b),* $M_{ise,2}$ = 0.955 *(c),* $M_{ise,2}$ = 1.01 *(d)*

5.4.2. Boundary layers

The boundary layer velocity profiles are shown in Figs. 5.4a and 5.4b for the baseline and multipoint profiles at different downstream isentropic Mach numbers at a location very close to the TE for the suction side ($x = 0.037m$) and the pressure side ($x = 0.036m$) respectively. The latter locations are defined as the points slightly upstream of the TE circle at the suction side and pressure side, as indicated by the arrows in Figs. 5.4a and 5.4b. The boundary layer profiles show that there is no boundary layer separation.

The boundary layer displacement thickness, momentum thickness and shape factor for different Mach numbers are shown in Figs. 5.5a -5.5d at the same locations for the suction side and pressure side. The multipoint optimal profile has smaller boundary layer displacement thickness and momentum thickness than the baseline at the pressure side location for the entire Mach range (see Figs. 5.5b and 5.5d), which is good for low profile losses. At the suction side these parameters are above or below the baseline parameters depending on the Mach number. In general, the boundary layer displacement thickness and momentum thickness of the multipoint optimal design are lower than the baseline for Mach numbers above 0.96 approximately.

Figures 5.5e and 5.5f show the shape factors at the start of the trailing edge on the suction side and pressure side respectively. A higher shape factor means that there is a stronger adverse pressure gradient. The shape factor for the baseline design at both the suction side and pressure side locations is in between the typical laminar Blasius boundary layer value ($H = 2.59$) and the turbulent values ($H = 1.3 - 1.5$). The multipoint optimal design has also a shape factor in between the laminar and the turbulent values, except for the pressure side where the shape factor is above the typical Blasius boundary layer value.



(a) Suction side close to the TE

(b) Pressure side close to the TE

Figure 5.4.: *Boundary layer profiles close to the trailing edge*

5.4.3. Losses

The main mechanisms of entropy creation in the LS89 turbine cascade are the viscous friction in the boundary layers and shear layers. The profile losses are usually taken to be the losses generated at the blade boundary layers and the loss arising at the trailing edge. Figures 5.7a - 5.7d show the entropy generated close to the trailing edge w.r.t the entropy at the inlet boundary for the baseline and multipoint optimal designs at different Mach numbers. A comparison between Fig. 5.7a and Fig. 5.7b or between Fig. 5.7c and Fig. 5.7d shows that the multipoint optimal design generates lower losses than the baseline design, specially in the region where the suction side and the pressure side boundary layers mix downstream of the trailing edge.

For incompressible flow, Denton [127] expressed the profile losses by Eq. 5.5, which relates the base pressure coefficient, the boundary layer parameters, the throat distance and the trailing edge thickness. The base pressure coefficient is computed with Eq. 5.6, and its effect is increased in compressible flow [128].

$$\zeta_p = \frac{-C_b D_{TE}}{t} + 2\frac{\theta_{SS} + \theta_{PS}}{t} + \left(\frac{\delta^*_{SS} + \delta^*_{PS} + D_{TE}}{t}\right)^2 \tag{5.5}$$

$$C_b = \frac{p_b - p_2}{\frac{1}{2}\rho V_2^2} \tag{5.6}$$

The first term of Eq. 5.5 accounts for the downstream mixing losses due to the differences between the average pressure across the channel exit p_2 and the pressure at the base of the trailing edge p_b. The second term of Eq. 5.5 represents the mixed out loss of the boundary layers on the blade surface just before the trailing edge for a zero trailing edge thickness. The third term of Eq. 5.5 represents the losses arising from a sudden expansion at the trailing edge due to the combined blockage of the trailing edge and the boundary layers. Figures 5.6a-c show the magnitude of each term as a function of the downstream isentropic Mach number. The mixed out loss of the boundary layers plays a dominant role in the profile losses (Fig. 5.6b), followed by the downstream mixing loss (Fig. 5.6a) and the trailing edge blockage loss (Fig. 5.5c). Figure 5.6d shows that the profile losses of the multipoint optimal design are lower than the baseline for the entire range of Mach numbers analysed in this chapter. The base pressure of the baseline and optimal designs is higher than the outlet static pressure (i.e., $p_b > p_2$), which is good for low profile losses. However, the greatest reduction in profile losses does not come from the downstream mixing losses. The multipoint optimal design has lower profile losses than the baseline for the entire Mach range mainly due to the lower mixed out loss of the boundary layers (Fig. 5.6b) and the lower trailing edge blockage loss (Fig. 5.6c).

Table 5.3 shows a summary of the main changes in entropy generation, exit flow angle, mass flow, and total pressure losses ($P_{01} - P_{02}$), for the three selected operating points. The largest aerodynamic improvements were achieved for the transonic operating point where the total pressure loss reduction was of the order of 14.06 %. The mass flow changes are kept within a reasonable limit and the exit flow angle is in all the three operating points above the baseline value.

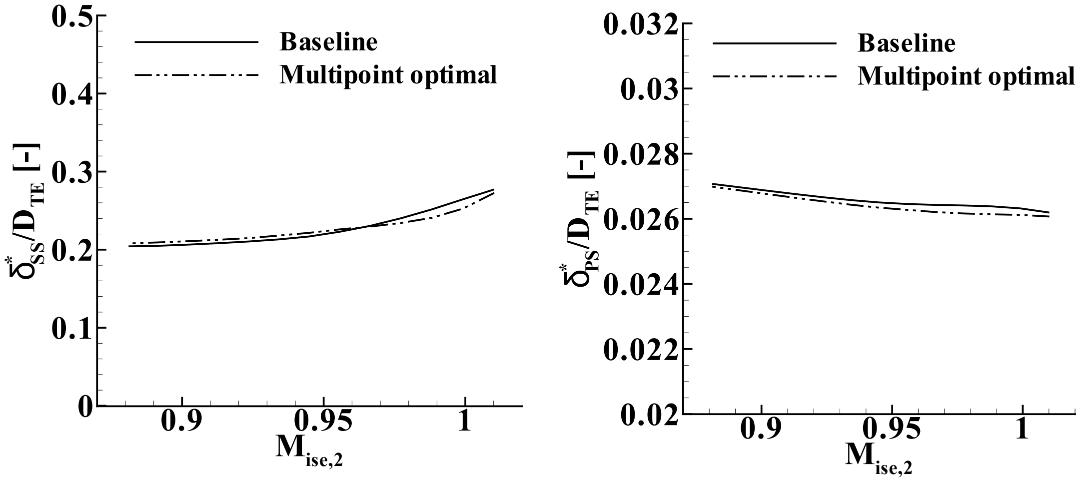
Figure 5.8 shows the total pressure loss coefficient at off-design conditions for the baseline, single point (from chapter 4) and multipoint optimal profiles, which is defined as the total pressure difference between the inlet and outlet divided with the dynamic head at the outlet plane (i.e. $q_2 = P_{02} - p_2$). The total pressure loss coefficient shown in Fig. 5.8 includes the irreversibilities in a compressible flow, such as the appearance of shock waves at high downstream isentropic Mach numbers. The single point optimal profile targets only low losses at the $M_{ise,2} = 0.9$ operating point. At off-design conditions, the aerodynamic performance of the single point optimal profile is reduced as the downstream isentropic Mach number is increased. Below $M_{ise,2} = 0.94$, the single point optimal profile would yield the lowest total pressure losses. However, beyond $M_{ise,2} = 0.94$ the performance deteriorates rapidly, and the baseline would have lower total pressure losses than the single point optimal profile. In contrast, the multipoint optimal profile has lower total pressure losses than the baseline at off-design conditions for the whole Mach number range analysed in the present study.

Operating point	$M_{ise,2} = 0.9$	$M_{ise,2} = 0.955$	$M_{ise,2} = 1.01$
Entropy generation variation	-1.92 %	-0.96%	-13.79%
Exit flow angle variation	0.05 %	0.16 %	0.17 %
Mass flow variation	0.12 %	-0.36 %	-0.05 %
Total pressure losses variation	-1.98 %	-5.94 %	-14.06 %

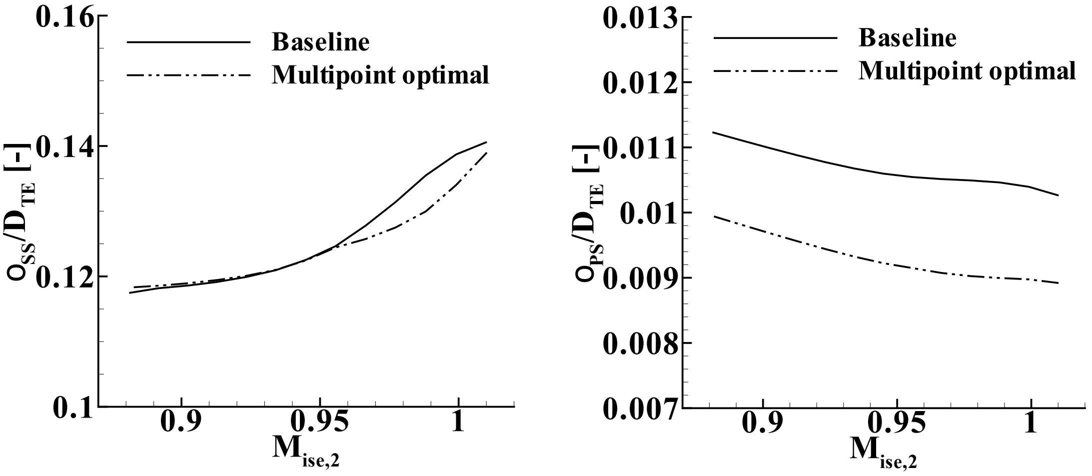
Table 5.3.: *Changes in performance at the different operating points*

5.5. Conclusions

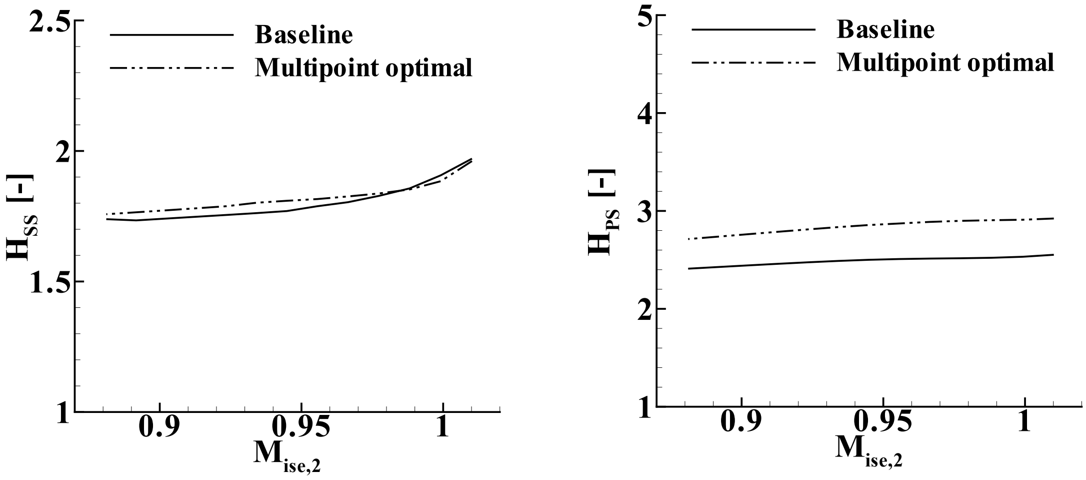
This chapter presents a multipoint optimization of the LS89 axial turbine vane profile for a downstream isentropic Mach number of 0.9 (design point), 0.955 and 1.01. By keeping the CAD representation in the optimization loop, it is not necessary to convert the optimal grid back to a smooth CAD shape and it is possible to impose higher level constraints on the shape, such as axial chord length, trailing edge radius and second order derivative continuity at the leading edge. The off-design performance of the LS89 was significantly improved. The largest aerodynamic improvements were achieved at the transonic operating point where the total pressure losses were reduced by 14% whilst keeping the exit flow angle fixed. A comparison between the multipoint and the single point optimal designs shows that the blade with a lower rear suction side turning (i.e. the multipoint optimal shape) can generate lower losses at transonic operating conditions than the blade with higher rear suction side curvature (i.e. single point optimal profile). However, the single point optimal design can generate lower losses if it operates at subsonic downstream isentropic Mach below to 0.935.



(a) Displacement thickness at SS close to TE (b) Displacement thickness at PS close to TE

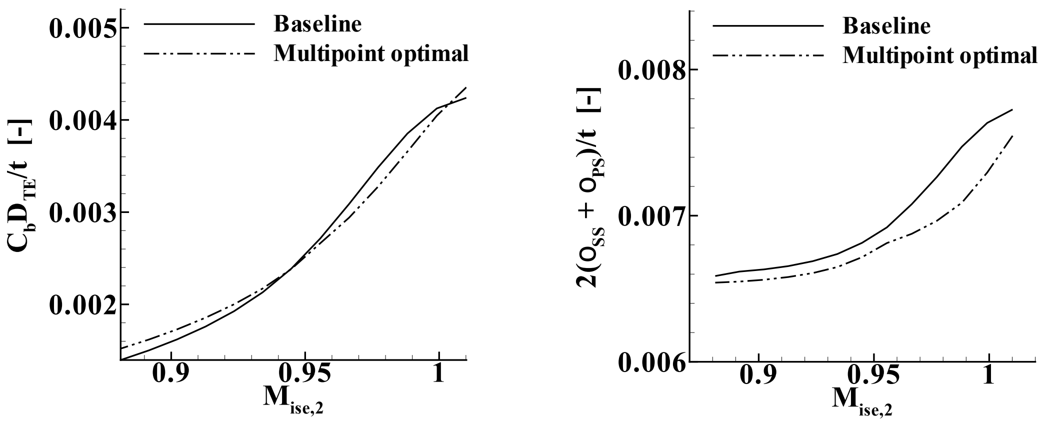


(c) Momentum thickness at SS close to TE (d) Momentum thickness at PS close to TE



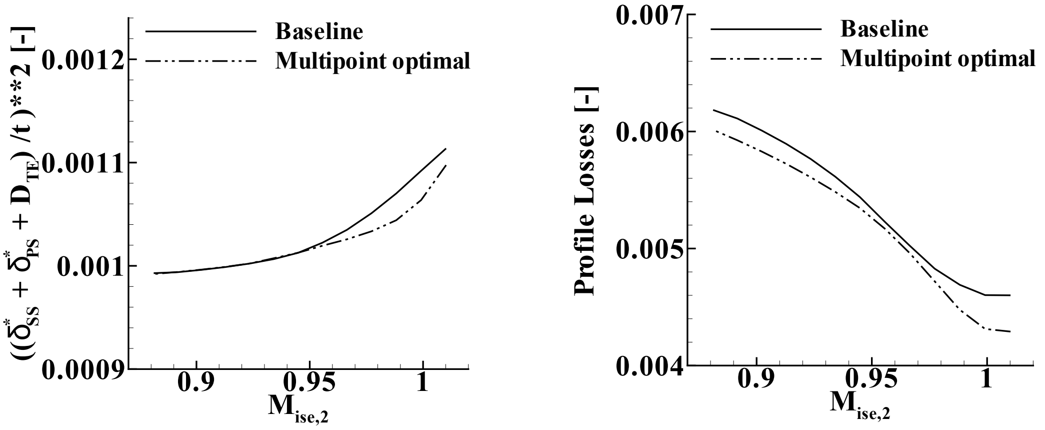
(e) Shape factor at SS close to TE (f) Shape factor at PS close to TE

Figure 5.5.: *Boundary layer parameters for different downstream isentropic Mach numbers*



(a) Downstream mixing losses

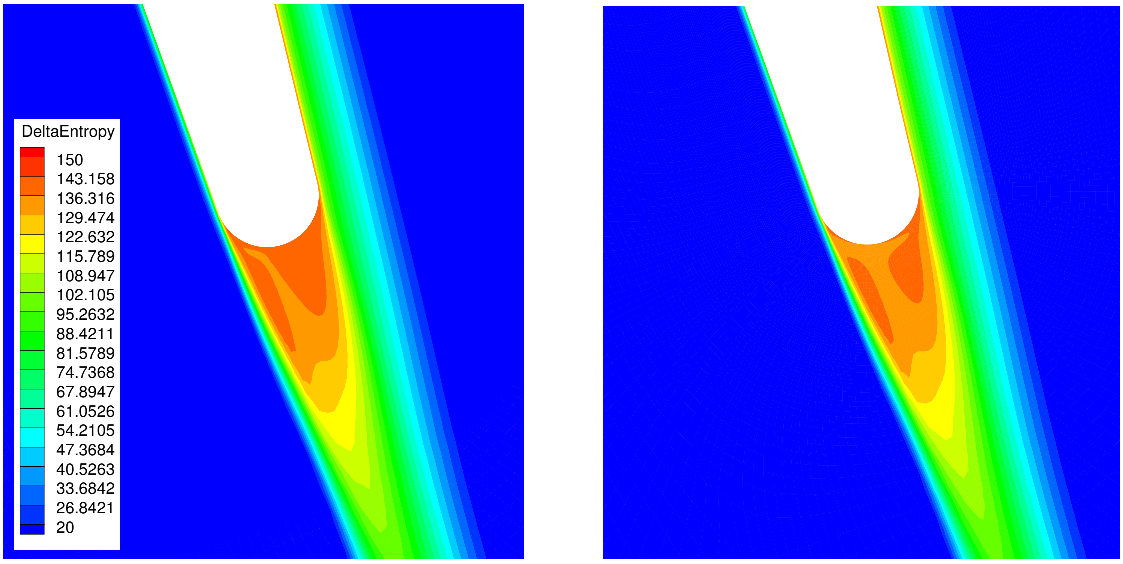
(b) Mixed out loss of the boundary layers



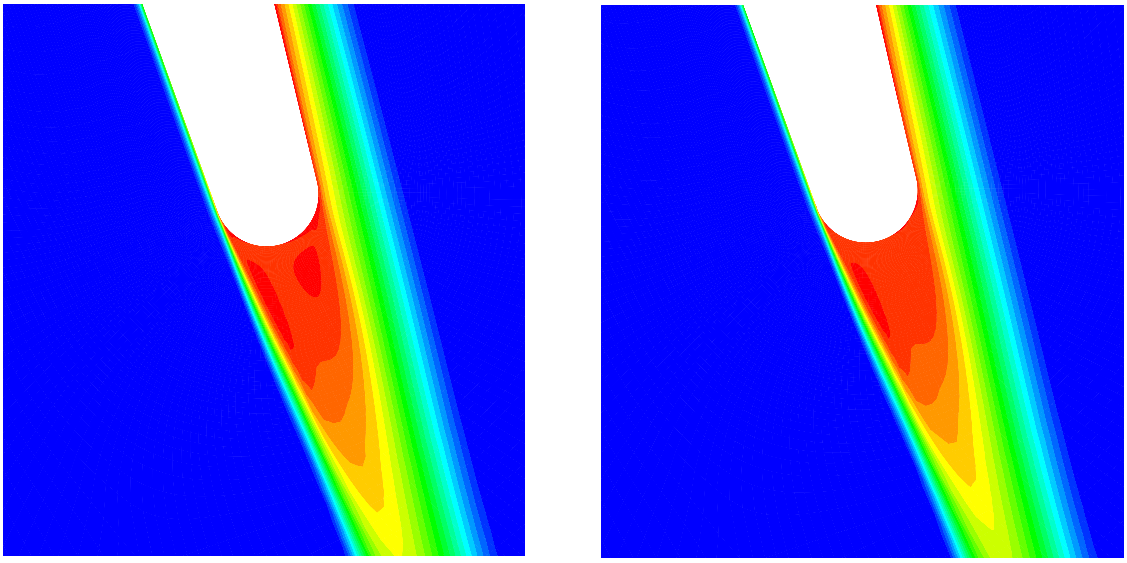
(c) Trailing edge blockage loss

(d) Profile losses

Figure 5.6.: *Contributions to the profile losses (a-c) and profile losses as a function of the downstream isentropic Mach number*



(a) Baseline at $M_{ise,2} = 0.90$ (b) Multipoint optimal at $M_{ise,2} = 0.90$



(c) Baseline at $M_{ise,2} = 1.01$ (d) Multipoint optimal at $M_{ise,2} = 1.01$

Figure 5.7.: *Entropy generation contour at the trailing edge for the baseline and multipoint optimal design for* $M_{ise,2} = 0.90$ *and* $M_{ise,2} = 1.01$

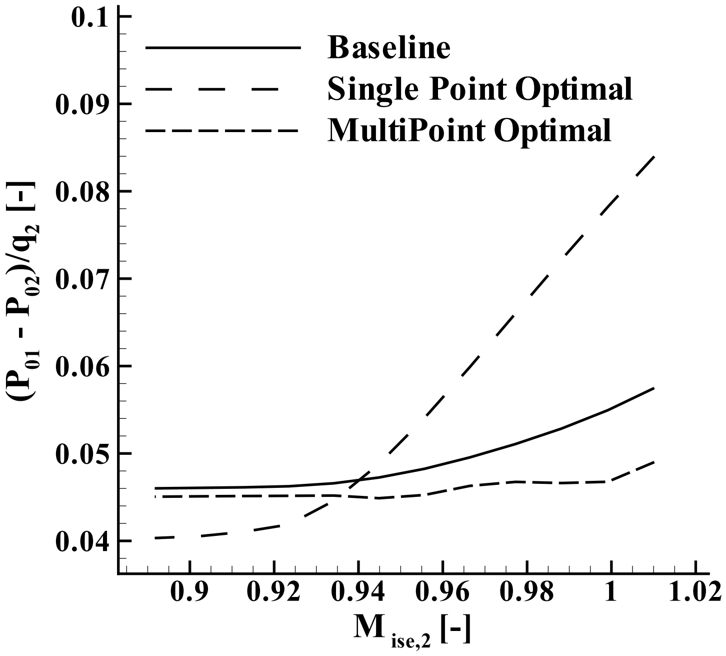


Figure 5.8.: *Variation of the total pressure loss in function of the downstream isentropic Mach number*

Chapter 6.

A novel definition of parametric effectiveness for CAD-based optimization

In chapters 4 and 5 the CAD is integrated in a CFD and adjoint based optimization framework to improve the LS89 turbine cascade aerodynamic performance at design and off-design conditions respectively. However, the question arises whether the parameterization defined at the CAD level is fit for the optimization, or larger aerodynamic gains could be achieved by changing the parameterization. In this chapter, a novel parametric effectiveness indicator is defined, which aims to rate the quality of the CAD-based parameterization for optimization purposes.

Abstract. This chapter presents a new methodology to rate a CAD-based parameterization for optimization. CAD-based approaches have the advantage to represent the optimal shape in the industry accepted format for manufacturing. However, the success of a CAD-based optimization framework highly depends on the expertise and skill of the designer, and often it is not obvious which parameters should be chosen for the optimization to move the model boundary in the direction indicated by the adjoint sensitivities. Furthermore, the choice of the CAD parameters constraints how the model boundary can change, and a poor choice may prevent the exploration of innovative geometries. For this purpose, the parametric effectiveness is defined to rate the CAD-based parameterization in terms of its ability to produce a shape change that is aligned with the adjoint sensitivities. The parametric effectiveness takes into account the non-orthogonality of the design variables and uses accurate geometric sensitivities provided by algorithmic differentiation. The methodology is extended to constrained optimization problems where the constraints can be of any type: geometrical, aerodynamic, structural, etc. The method will be tested to rate different CAD-based parameterizations of the LS89 turbine cascade and to select four of them for optimization at design point. Large aerodynamic improvements are achieved whilst keeping the aerodynamic constraints satisfied (e.g. exit flow angle and/or mass flow). The trailing edge and axial chord length are kept as manufacturing constraints. The flow is governed by the Reynolds-averaged Navier-Stokes equations and the one-equation Spalart-Allmaras turbulence model.

6.1. Introduction

A product design in an industrial framework typically starts by creating a computer-aided design (CAD) geometry and by iteratively changing it until the optimal geometry is found and delivered to manufacturing. Two categories of optimization algorithms can be used in order to find the best performing geometry: gradient-free or gradient based. Gradient-free approaches require the standard output of the computer aided engineering (CAE) packages used in the design chain, such as the total pressure losses or the von misses maximum stresses. Despite these methods can find the global optimum in the design space, gradient-free approaches lack speed of convergence to the optimum. Since the CFD analysis can take very long run times and often very rich design spaces are needed, the gradient-free approach can become extremely expensive for industrial applications. In this regard, the use of gradient-based approaches is desirable, since they are computationally efficient specially for large number of design variables. However, this requires computing the gradients of the objective function with respect to each design variable. Typically, these gradients can be computed using finite-differences by computing the objective function change due to the change of a parameter. However, the cost of computing these derivatives is proportional to the number of design variables and this can become impractical for test cases with very large number of design variables. This issue can be resolved by using CFD adjoint methods [31–34] as they can compute gradients with respect to a large number of design variables at near-constant computational cost and very similar to the primal flow computation.

Choosing a suitable parameterization is important for at least two reasons. First, it can reduce the post-processing efforts to deliver an optimal shape feasible for manufacturing. Also, since the choice of the CAD parameters constraints how the model boundary can change, a poor choice may prevent the exploration of innovative geometries. In general, the parameterization of the geometry can be done at either the grid level or within the computer-aided design (CAD) environment. In the former case, a common approach is to use the grid point coordinates as design variables [3–5], which offers a very rich design space but the connection to the CAD geometry is lost and it can be very complex and time consuming to transform the optimal shape defined by grid points back to smooth CAD shape [6, 7]. Also, using the grid coordinates as design variables can be quite challenging in a multi-disciplinary design optimization where different meshes might be required by each CAE package. Alternatively, the advantage of parameterizing the CAD model in the CAD system is that higher level constraints can be imposed on the shape, allowing the optimized part to be manufactured. The downside of using CAD model parameters is that its success highly depends on the expertise and skill of the user, and often it is not obvious from the parameterization which parameters should be chosen for the optimization that would allow to achieve the desired shape change that the adjoint sensitivities dictate. It is therefore of vital importance to quantify how suitable is a set of CAD-based design parameters for use in optimization.

Robinson et. al. [106] introduced the concept of parametric effectiveness of a CAD model, in order to rate the quality of the parameters of a CAD-based model for use as optimization variables. The parametric effectiveness as defined by Robinson et. al. [106] is computed as the ratio between the maximum change in performance that can be achieved using existing parameterization $\Delta J_{CAD,OBM}$ (Eq. 6.6), to the maximum performance improvement that could be obtained if the model is not constrained by any parameterization and follows the shape changes dictated by the adjoint sensitivities $\Delta J_{ideal,OBM}$ (Eq. 6.5). The authors in [106] normalize the changes in performance that are used to compute the parametric effectiveness with respect to an overall boundary movement (OBM), which is defined as a unit root mean squared movement of the boundary. This ensures that the parameters moving an area of low sensitivity by a large amount will not be favoured compared to the parameters causing a small localised movement in the areas of high sensitivity, even if they have a high performance sensitivity $dJ/d\alpha$.

In this chapter, a novel methodology is presented to compute the parametric effectiveness without any constraint on the boundary movement and the direction of gradient as used in [106]. The exact grid sensitivities are calculated by differentiating the CAD kernel and grid generator using Algorithmic Differentiation (AD), and is developed to deal with constrained optimization problems. Different CAD parameterizations of the LS89 turbine cascade [117, 118] will be rated using the parametric effectiveness defined in this work and in [106]. Finally, the LS89 will be optimized using different parameterizations and the chosen parameterization will remain fixed during the optimization.

6.2. Methodology

6.2.1. Adjoint methods

The adjoint method has been extensively developed in the last two decades and has been used as an efficient tool for computing performance gradients for a large number of design variables [31–34]. The underlying theory of adjoint methods is presented in Sec. 2.2.5.

6.2.2. Algorithmic differentiation of CAD kernel and grid generator

The application of the adjoint method to the CFD solver allows an efficient computation of $dJ/d\vec{X}$, the sensitivity of the objective function with respect to grid point movement. In a CAD based approach, however, the grid movement is controlled by a limited set of design variables $\vec{\alpha}$; this is more restrictive than if the grid points on the shape surface are allowed to be changed independently. In the following we denote $\vec{X_S}$ as the grid points on the shape surface, which are a subset of the full grid $\vec{X}$. We distinguish two parameterization approaches:

- a node based approach, where each surface grid point $\vec{X_S}$ is a design variable
- a CAD based approach, where for less CAD parameters $\vec{\alpha}$ control the grid point movement $\vec{X_S}$

The full grid $\vec{X}$ is constructed based on the grid boundary $\vec{X_S}$ through, in this work, applying an elliptic smoothing algorithm [79]. This defines the relation $\vec{X}(\vec{X_S})$ and can be differentiated to obtain $d\vec{X}/d\vec{X}_S$, or in other words the sensitivity of internal mesh points to a boundary grid point movement. These sensitivities allow computing $dJ/d\vec{X}_S$ by

$$\frac{dJ}{d\vec{X}_S} = \frac{dJ}{d\vec{X}}\frac{d\vec{X}}{d\vec{X}_S} \tag{6.1}$$

The projection step from the performance sensitivities w.r.t. the volume grid $dJ/d\vec{X}$ to the surface grid $dJ/d\vec{X}_S$ is done by applying Algorithmic Differentiation (AD) to the grid generation in reverse mode. In this work, the reverse mode of AD is used to record the grid generation operations from the surface mesh coordinates $\vec{X}_S$ to the volume mesh coordinates $\vec{X}$ in a tape. Therefore, the term $dJ/d\vec{X}_S$ is obtained by evaluating the tape in reverse mode whilst seeding the outputs $\vec{X}$ with the term $dJ/d\vec{X}$.

For the node based approach, $dJ/d\vec{X}_S$ is the end station. For a CAD based approach, however, the performance sensitivities w.r.t. the surface grid $dJ/d\vec{X}_S$ need one step of further projection to obtain $dJ/d\vec{\alpha}$ (Eq. 6.2). For this the relation $\vec{X}_S(\vec{\alpha})$, or the definition of the surface grid nodes based on the CAD parameters $\vec{\alpha}$, needs to be differentiated leading to $d\vec{X}_S/d\vec{\alpha}$. This is typically done by using finite differences [15, 16]. However, the accuracy of the gradients is dependent upon the chosen step-size for each design variable. Moreover, it is not guaranteed that the topology of the CAD and the grid will remain the same after perturbing the design variable.

$$\frac{dJ}{d\vec{\alpha}} = \frac{dJ}{d\vec{X}_S}\frac{d\vec{X}_S}{d\vec{\alpha}} \tag{6.2}$$

6.2.3. Constrained optimization problem

A simple optimization problem with constraints can be represented as

$$\begin{aligned} min \quad & J(\vec{\alpha}), \\ w.r.t \quad & \vec{\alpha}, \\ s.t. \quad g_1(\vec{\alpha},\vec{U}) & > 0, \\ g_2(\vec{\alpha},\vec{U}) & = 0. \end{aligned} \tag{6.3}$$

In a constrained optimization problem, the motive is not only to increase the performance but also to meet the constraints imposed by the system. When the constraints are not satisfied, an efficient way to compute the constrained performance sensitivities w.r.t the surface grid $\left(dJ/d\vec{X}_S\right)_c$ is required. These are the sensitivities that allow a reduction of the objective function without violating the constraints. The solution proposed here is based on the principle of the projected gradient method, which is illustrated in Fig. 6.1. The grey area of Fig. 6.1 represents the design space in which the constraint is satisfied and the border of the grey area represents the constraint hyperplane surface.

Suppose that the vectors $\vec{\nabla J} = dJ/d\vec{X}_S$ and $\vec{\nabla g_1} = dg_1/d\vec{X}_S$ contain the performance sensitivities for the objective function and the constraint respectively w.r.t. the surface grid. The sensitivity vector $\vec{\nabla J}$ can be expressed as a sum of two vectors $\nabla\vec{J}_{c1}$ and $\nabla\vec{J}_{u1}$, where $\nabla\vec{J}_{u1}$ is the component

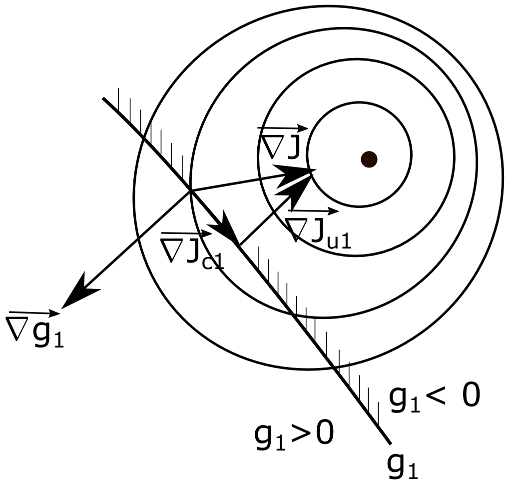


Figure 6.1.: Principle of the projected gradient method.

of $\nabla\vec{J}$ in the direction of the constraint $\nabla\vec{g}_1$ and $\nabla\vec{J}_{c1}$ the component orthogonal to the constraint $\nabla\vec{g}_1$. $\nabla\vec{J}_{c1}$ is thus the portion of the gradient which satisfies the constraints and can be computed as follows

$$\nabla\vec{J}_{c1} = \nabla\vec{J} - \nabla\vec{J}_{u1} = \nabla\vec{J} - \frac{\nabla\vec{J} \cdot \nabla\vec{g}_1}{||\nabla\vec{g}_1||^2}\nabla\vec{g}_1 \tag{6.4}$$

$\nabla\vec{J}_{c1}$ thus contains the sensitivities which improve the objective function without violating the constraints. For a constrained optimization problem with multiple constraints, we need to project the sensitivities to the hyperplane orthogonal to all the constraint gradients, as shown by the algorithm 1.

Algorithm 1: Projected gradient method to compute the constrained performance sensitivities w.r.t. the surface grid

```
Data: Performance sensitivities for the objective function $\nabla\vec{J}$ and the constraints $\nabla\vec{g}_i$ for
      $i = 1, ..., n_c$ w.r.t. the surface grid, where $n_c$ is the total number of constraints
Result: Constrained performance sensitivities w.r.t the surface grid $\nabla\vec{J}_c$
Initialize performance sensitivities w.r.t the surface grid $\nabla\vec{J}_c = \nabla\vec{J}$ ;
for $i = 1; i \leq n_c; i++$ do
    Update constrained performance sensitivities w.r.t the surface grid $\nabla\vec{J}_c = \nabla\vec{J}_c - \frac{\nabla\vec{J}_c \cdot \nabla\vec{g}_i}{||\nabla\vec{g}_i||^2}\nabla\vec{g}_i$
end
Return $\nabla\vec{J}_c$ ;
```

6.2.4. Parametric effectiveness

In a grid-based parameterization every grid coordinate is allowed to move during the course of the optimization. However, in a CAD-based parameterization there is an additional constraint that imposes the surface grid nodes to lie on the bounding surfaces of the CAD model. Imagine that we have a set of CAD design variables α_1, α_2, ..., α_n. The design movements or shape modes that can be achieved for each design variable are dictated by $d\vec{X}/d\alpha_1$, $d\vec{X}/d\alpha_2$, ..., $d\vec{X}/d\alpha_n$ as shown in Fig. 6.2.

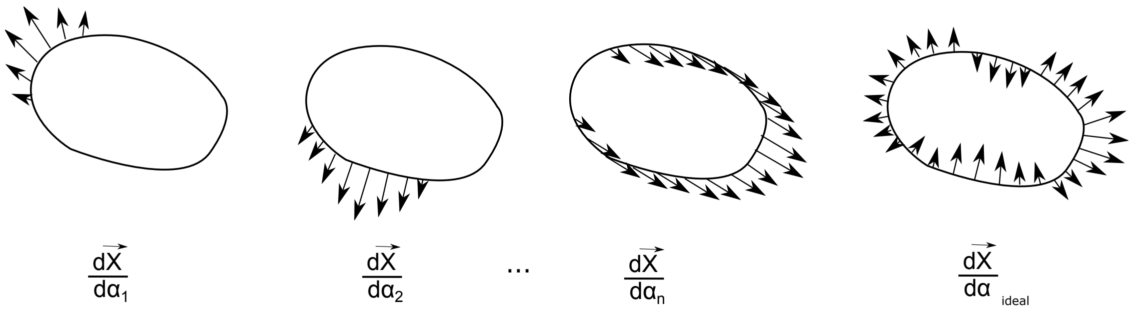


Figure 6.2.: Individual shape modes and ideal design movement.

In order to achieve the maximum gain/reduction in the objective function J, the surface grid points are required to be moved in the ideal direction given by the performance sensitivities w.r.t. the surface grid $(dJ/d\vec{X_S})$. However, when moving from a grid-based to a CAD-based parameterization it may not be possible to move every single grid point in the exact desired direction as dictated by $(dJ/d\vec{X_S})$. The need to rate the CAD-based parameterization, in terms of its ability to produce a shape change governed by what the adjoint sensitivities dictate, gave birth to the definition of parametric effectiveness by Robinson et. al. [106]. The parametric effectiveness was proposed by Robinson et. al. [106] as the ratio of the change in performance achieved by perturbing all the parameters in the steepest descent direction (i.e. $\Delta J_{CAD,OBM}$), subject to the constraint of an overall unit root-mean-squared boundary movement, to the maximum performance improvement that could be obtained if the model is free to follow the shape changes dictated by the adjoint sensitivities subject to the OBM constraint (i.e. $\Delta J_{ideal,OBM}$). The parametric effectiveness ranged from 0 to 1, where a higher value (~ 1) indicated that the model parameters are capable to change the shape close to that predicted by the performance sensitivities w.r.t. the surface grid and vice-versa. The detailed mathematical derivation about the methodology can be found in [106], and only a summary of the important equation is presented here.

The optimum performance change ($\Delta J_{ideal,OBM}$) per unit of root-mean-square design velocity, for a model which is not constrained by any parameterization can be computed as

$$\Delta J_{ideal,OBM} = -\sqrt{A \int_A \phi^2 dA} \tag{6.5}$$

where A is the total surface area of the model over the discretized facets and ϕ are the performance sensitivities w.r.t. the surface grid per unit of area for an objective function J.

The assumption that the optimum performance using the CAD parameterization can be achieved by perturbing the parameters in the steepest descent direction [106] allows to compute the optimum performance change as

$$\Delta J_{CAD,OBM} = -\sqrt{\frac{A}{\int_A \left(\sum_{i=1}^{n} S_i V_{ni}\right)^2 dA} \sum_{i=1}^{n} \left(S_i\right)^2} \tag{6.6}$$

where S_i are the performance sensitivities w.r.t. the CAD parameters and V_{ni} are the design velocities. The parametric effectiveness (PE) defined by Robinson et. al. [106] was then calculated as

$$PE = \frac{\Delta J_{CAD,OBM}}{\Delta J_{ideal,OBM}} \tag{6.7}$$

Using the constraint on boundary movement, authors in [106] were able to penalise the parameters which moved a larger part of boundary in the low sensitivity regions. Later in [107], authors showed that using this measure of PE it was possible to localize the boundary movement in the areas of high sensitivities.

An alternative definition of parametric effectiveness is proposed in this thesis, which removes the OBM constraint. Also, the parametric effectiveness is extended to constrained optimization problems where the constraints can be of any type: geometrical, aerodynamic, structural, etc. The new definition of parametric effectiveness is illustrated in Fig. 6.3, which shows the shape modes ($d\vec{X}/d\alpha_1$, $d\vec{X}/d\alpha_2$) of a two-design variable CAD-based parameterization. Ideally, the design movement would be the direction dictated by the constrained performance sensitivities w.r.t. the surface grid $(dJ/d\vec{X}_S)_c$. However, all the possible design movements that can be achieved due to the constraints imposed by the CAD-based parameterization on the boundary are contained in a hyperplane (grey area of Fig. 6.3). From all the possible design movements contained within the hyperplane defined by the basis ($d\vec{X}/d\alpha_1$, $d\vec{X}/d\alpha_2$, ..., $d\vec{X}/d\alpha_n$), the best possible design movement $(d\vec{X}/d\alpha)_{best}$ that a CAD-based parameterization can produce due its constraints is the projection of the constrained performance sensitivities w.r.t. the surface grid on the hyperplane. $(d\vec{X}/d\alpha)_{best}$ can be expressed as a linear combination of grid sensitivities which are multiplied by individual scalars λ_i (see Eq. 6.8). Therefore, we need to solve Eq. 6.9 for the λ_i that yield a design movement as close as possible to the ideal one, which is dictated by $(dJ/d\vec{X}_S)_c$. In this regard, the new parametric effectiveness can be defined as an indicator that rates the ability of a set of CAD parameters to produce the shape change in the direction that $(dJ/d\vec{X}_S)_c$ dictate. The parametric effectiveness ranges between zero and one. If the value of parametric effectiveness is one, this means that the CAD parameterization is capable to move every grid point on the boundary in the ideal direction and thus $\left(dJ/d\vec{X}_S\right)_c$ is contained in the hyperplane spanned by $d\vec{X}/d\alpha_1$, $d\vec{X}/d\alpha_2$, ..., $d\vec{X}/d\alpha_n$.

$$\left(\frac{d\vec{X}}{d\alpha}\right)_{best} = \sum_{i=1}^{n} \lambda_i \frac{d\vec{X}}{d\alpha_i} \tag{6.8}$$

$$\sum_{i=1}^{n} \lambda_i \frac{d\vec{X}}{d\alpha_i} = \left(\frac{dJ}{d\vec{X}_S}\right)_c \tag{6.9}$$

Eq. 6.9 can be re-written as Eq. 6.10, where A is a matrix that contains the grid sensitivities $d\vec{X}/d\alpha_i$, $\vec{\lambda}$ is a vector containing the λ_i's and $\vec{b}$ is a vector containing the constrained performance sensitivities w.r.t. the surface grid $(dJ/d\vec{X}_S)_c$.

$$A\vec{\lambda} = \vec{b} \tag{6.10}$$

Since the size of A is $m \times n$ with $m > n$, the system of equations expressed in Eq. 6.10 is over-

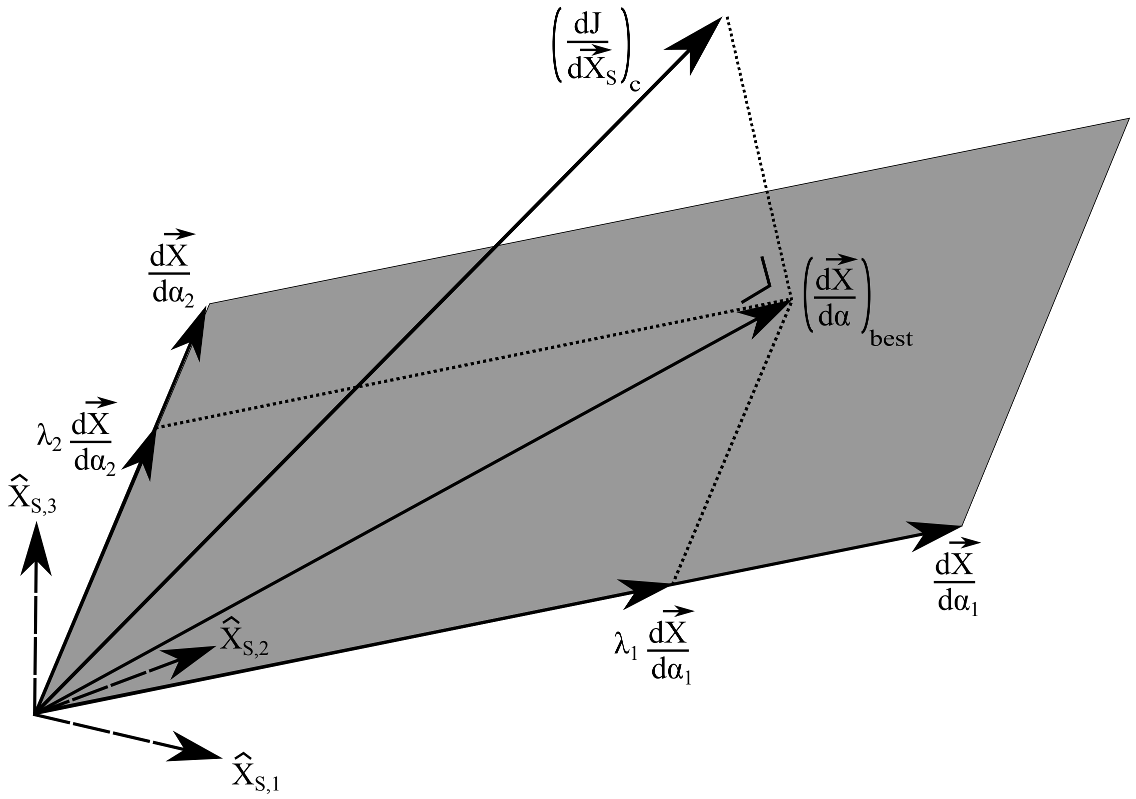


Figure 6.3.: Best possible design movement due to the constraints imposed by a CAD-based parameterization that uses two design variables (α_1, α_2)

determined, and can be solved using least-square approximation with the smallest norm of the residual $\vec{r} = A\vec{\lambda} - \vec{b}$. The required solution $\vec{\lambda}$ can be computed as

$$A^T A\vec{\lambda} = A^T\vec{b} \tag{6.11}$$

$$\vec{\lambda} = (A^T A)^{-1} A^T\vec{b} \tag{6.12}$$

Fig. 6.2 shows that perturbing each design variable α_i produces a particular design movement on the surface. Two variables that produce similar design movements on the surface cannot be considered independent from each other, since the shape modes they produce are not orthogonal. In the method presented above the cross-dependency terms $d\vec{X}_S/d\alpha_i \cdot d\vec{X}_S/d\alpha_j$ appear inside the off-diagonal terms of the matrix $A^T A$. If $d\vec{X}_S/d\alpha_i \cdot d\vec{X}_S/d\alpha_j \neq 0$, the two basis $d\vec{X}_S/d\alpha_i$ and $d\vec{X}_S/d\alpha_j$ are not orthogonal to each other.

The square matrix $N = A^T A$, which is often called the Gram matrix (or Grammian) [131], has size $n \times n$ and its inverse can be computed using Single Value Decomposition (SVD). In this work, a two-sided Jacobi R-SVD decomposition [132] is used to compute the inverse of the matrix $A^T A$ and solve Eq. 6.12.

After computing the least-square solution $\vec{\lambda}$, the parametric effectiveness can be computed as the scalar product of the best possible movement and the adjoint vectors, divided by each of its norms

(Eq. 6.13). If the best possible movement $A\vec{\lambda}$ is equal to $\vec{b}$, the parametric effectiveness would be unity, in which case the CAD parameterization would provide performance improvement which will be similar to grid-based parameterization.

$$\eta_{global} = \frac{abs((A\vec{\lambda}) \cdot \vec{b})}{||A\vec{\lambda}|| \cdot ||\vec{b}||} \tag{6.13}$$

Equation 6.13 can be used to compute the parametric effectiveness of a group of n CAD design variables, in terms of their ability to move the shape boundary according to what the adjoint sensitivities dictate. However, this equation can also be used to find out which parameters are individually the most effective ones. The only difference in the procedure described above (Equations 6.8 - 6.13) is that, in the particular case where $n = 1$, the matrix A has size $m \times 1$ and the vector $\vec{\lambda}$ becomes a scalar. Therefore, the individual parametric effectiveness of a CAD parameter is computed with Eq. 6.13 but with only one active design variable ($n = 1$).

6.2.5. CAD-based parameterization

The LS89 baseline geometry shown in Fig. 6.4 has a constant cross-section in the spanwise direction of the vane. The parameterization of LS89 used in this chapter (Fig. 6.5) is similar to that one used in Sec. 3.2.1 but with additional degrees of freedom (the position of the individual control points). The control points distribution in the streamwise direction (i.e., u^i_{SS} and u^i_{PS}) defines the position of the control point in the tangent direction to the camber line and the control point thickness distribution (i.e., t^i_{SS} and t^i_{PS}) defines it in the normal direction.

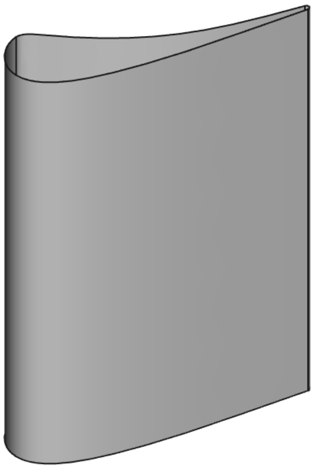


Figure 6.4.: LS89 vane geometry.

6.2.6. Sets, subsets and most effective susbsets

A set is the group of design variables (DV) contained by the CAD-based parameterization. In contrast, a subset is defined as a part of a set of design variables. In this chapter, three different sets are created: a set of 35DV and two sets of 43DV (named SS9PS9 and SS14PS4). Also, many different

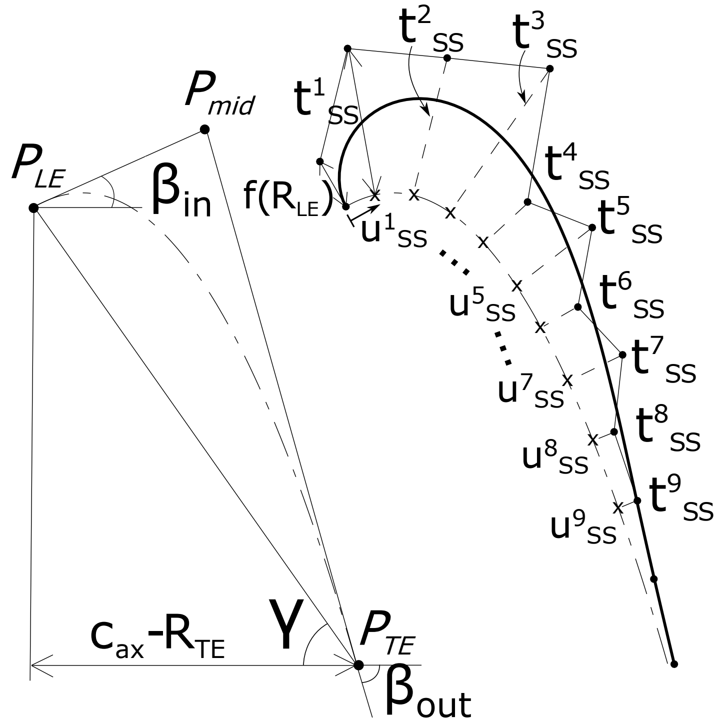


Figure 6.5.: Construction of the Suction Side by a B-spline curve using the streamwise distribution ${u_{SS}}^i$ of the control points.

subsets of DV are created from the before mentioned sets. The different groups of design variables will be rated by the parametric effectiveness later in order to show which set or subsets are more fit for optimization purposes, and this will be later demonstrated by performing an optimization using the different sets and subsets.

Five different subsets of the full set of 35DV are created by selecting which design variables are considered to be active during the optimization. Table 6.1 shows which parameters are active in each subset. Subset 5 contains the lowest number of DV (i.e., 7DV).Subset 4 contains all the active DV of subset 5 plus the pressure side thicknesses (i.e. 11DV). Subset 3 contains all the active DV of subset 5 plus the suction side thicknesses (i.e. 16DV). Subset 2 contains the DV of subset 5 plus both the pressure and suction side thicknesses (i.e. 20DV). In subset 1, all the DV of the full set are active except the axial chord length and the trailing edge radius (i.e. 33DV). The relationship between the different subsets is illustrated in Fig. 6.8, as it shows which subsets include the design variables of other subsets. The subset that includes all the other subsets is subset 1, as illustrated in Fig. 6.8. The full set of 35DV contains all the design variables of subset 1 plus the axial chord length and the trailing edge radius.

The two sets of 43DV (SS14PS4 and SS9PS9) are shown in Figs. 6.6 and 6.7. The SS14PS4 and SS9PS9 CAD-based parameterizations were created after inserting 5 knots equally spaced on the suction side and the pressure side B-spline curves respectively, by using the knot insertion algorithm described in [76]. In this way, the set SS14PS4 has 14 suction side thicknesses and 4 on the pressure side, whereas the set SS9PS9 contains 9 thicknesses on both the suction and pressure sides. Fig.

6.9 shows that the only active parameters that are shared between the set of 35DV and the sets of 43DV are those parameters contained in subset 5.

Assuming that the objective space is convex, a final shape with worse performance is expected to be found by the optimizer if we use a subset of CAD parameters instead of the full set (as shown in [107] for the S-bend duct optimization). This is because the optimizer explores a richer design space when using the full parametric CAD model during the optimization. However, using a subset of CAD design variables for the optimization has at least the following advantages: (1) the update of the CAD model during the course of the optimization can be faster [107]; (2) the optimizer converges faster to the optimum shape when using a coarse design space than when using a very fine design space [108]. For these reasons, sometimes it is desired to reduce the number of design variables to be used in the optimization. However, the designer does not always know a priori which design variables should be excluded from the optimization. In this context, the parametric effectiveness can also be used to indicate which are the most effective CAD parameters for optimization, in terms of their ability to move the boundary in the direction that the adjoint solution dictates. Finding the most effective subset of a set of parameters involves following two steps: (1) compute the individual parametric effectiveness of each CAD parameter α_i inside the set using Eq. 6.13 (with $n = 1$); (2) group all the CAD parameters with an individual parametric effectiveness above 1% as the most effective subset. Therefore, the most effective subset is defined as the group of CAD based parameters from a set that have an individual parametric effectiveness above 1%. The most effective subsets of the set SS14PS4 is illustrated in Fig 6.10.

Design parameter	Full set of 35DV	Subset 1 (33DV)	Subset 2 (20DV)	Subset 3 (16DV)	Subset 4 (11DV)	Subset 5 (7DV)
Solidity (σ)	1	1	1	1	1	1
Stagger angle (γ)	1	1	1	1	1	1
Leading edge radius (R_{LE})	1	1	1	1	1	1
Trailing edge wedge angle SS (φ_{SS})	1	1	1	1	1	1
Trailing edge wedge angle PS (φ_{PS})	1	1	1	1	1	1
Inlet metal angle (β_{in})	1	1	1	1	1	1
Outlet metal angle (β_{out})	1	1	1	1	1	1
PS thicknesses (t^1_{PS}, ..., t^4_{PS})	1	1	1	0	1	0
SS thicknesses (t^1_{SS}, ..., t^9_{SS})	1	1	1	1	0	0
PS streamwise position (u^1_{PS}, ..., u^4_{PS})	1	1	0	0	0	0
SS streamwise position (u^1_{SS}, ..., u^9_{SS})	1	1	0	0	0	0
Trailing edge radius (R_{TE})	1	0	0	0	0	0
Axial chord length (c_{ax})	1	0	0	0	0	0

Table 6.1.: *Active CAD parameters in each set and subset. The active ones are represented by* 1*'s and the non-active by* 0*'s.*

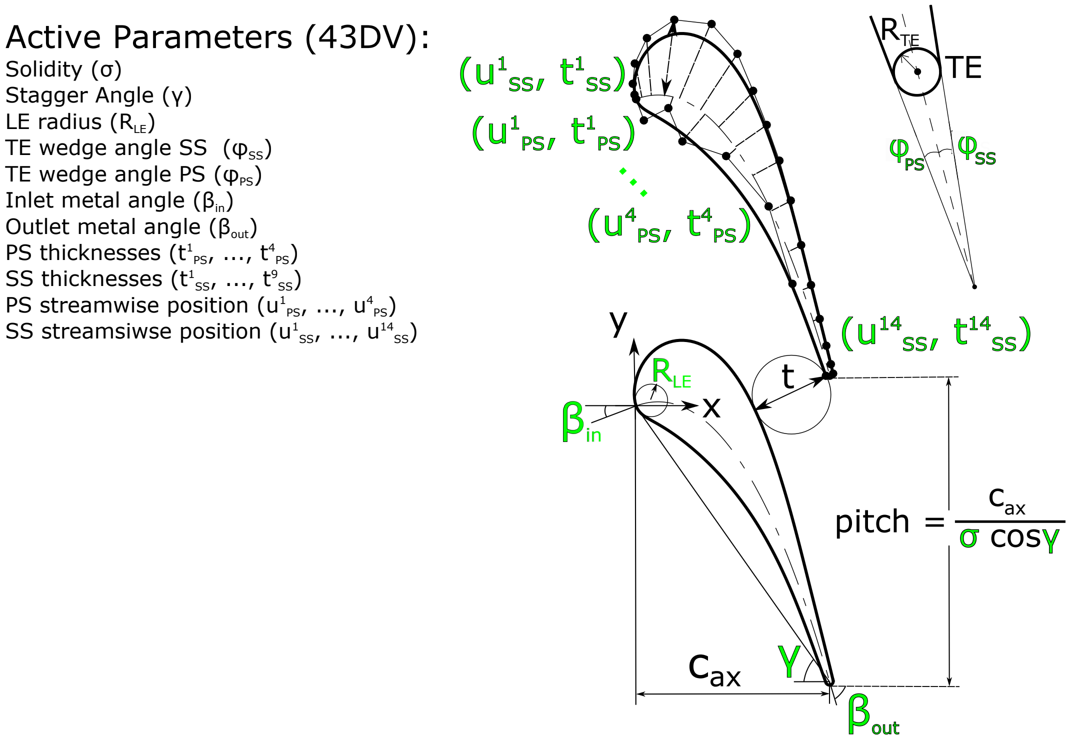


Figure 6.6.: Set SS14PS4 of CAD-based design parameters.

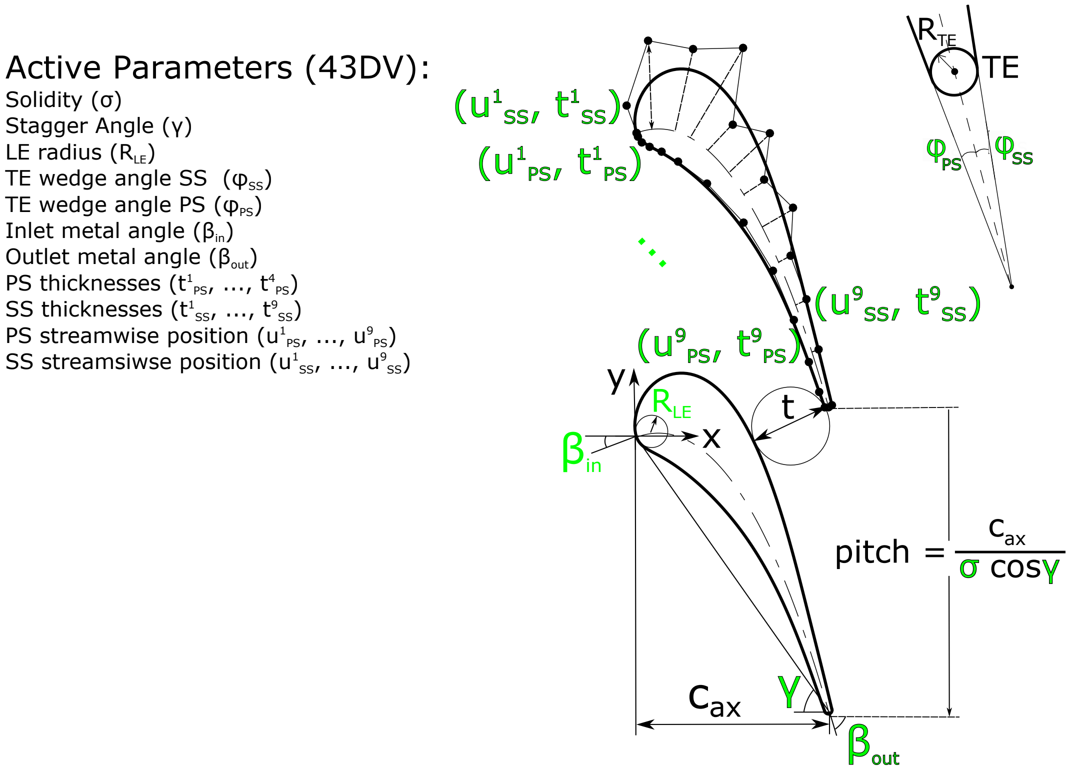


Figure 6.7.: Set SS9PS9 of CAD-based design parameters.

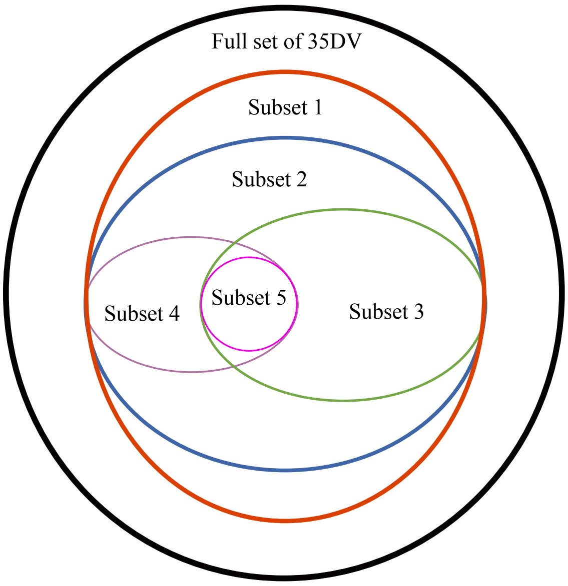


Figure 6.8.: Circles showing which subsets include the design variables of other subsets.

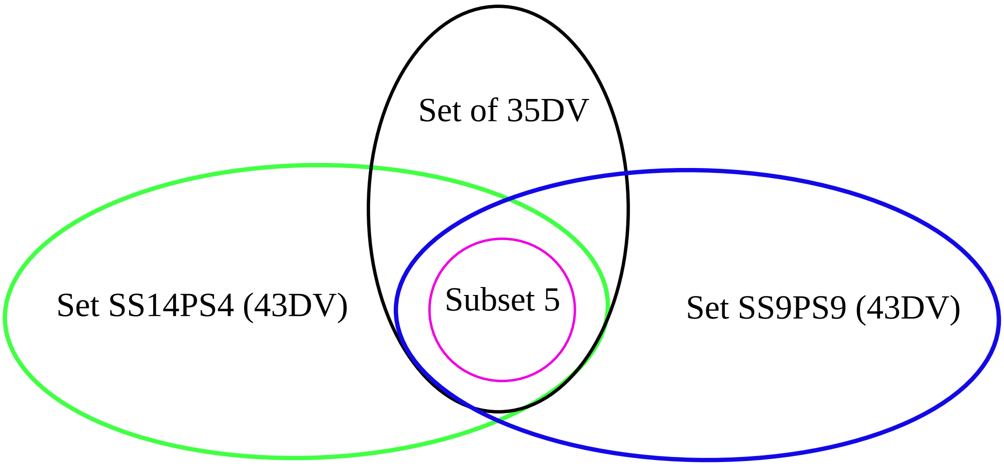


Figure 6.9.: Circles showing which subset of design variables is shared between the sets of 35DV and 43DV.

6.2.7. Optimization

In this chapter, some of the different sets and subsets of design parameters shown in Sec. 6.2.6 are used for optimization. The Sparse Nonlinear OPTimizer (SNOPT) version 7.6 is used, which is a software package specially useful for solving large-scale optimization problems (constrained or unconstrained) that is based on a sparse Sequential Quadratic Programming (SQP) algorithm with limited-memory quasi-Newton approximations to the Hessian of the Lagrangian. A non-derivative line search technique is chosen. A more detailed description of SNOPT is found in [100, 101].

A mesh-independence study was carried out at the beginning to find the most appropriate mesh settings to build a high quality multi-block structured grid. A new grid is built using elliptical equations during the coarse of optimization using the initial mesh settings. The grid generation methodology is described in more detail in Sec. 2.5.1.

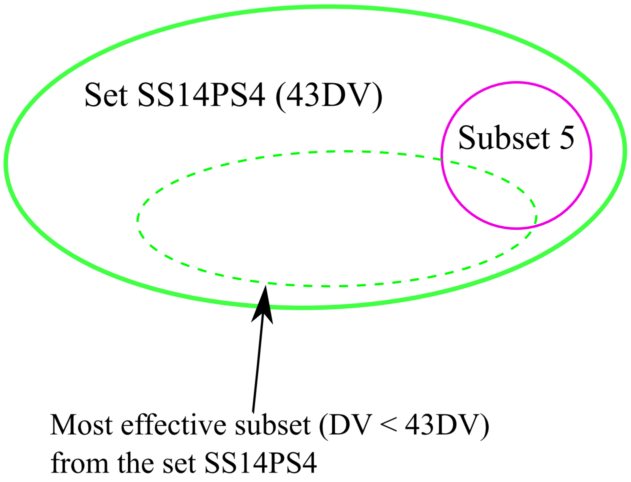


Figure 6.10.: Circles showing the set SS14PS4, subset 5 and the most effective subset of the set SS14PS4.

The compressible RANS equations were used to solve the flow field. The Spalart-Allmaras turbulence model [93] is used for the turbulence closure problem assuming fully turbulent flow with Reynolds number at the inlet as $Re_{inlet} \approx 2 \cdot 10^5$. More details of the flow solver are found in Sec. 2.7. The total pressure and the static pressure are specified at the inlet and outlet of the domain for the operating condition considered (i.e. an outlet isentropic Mach number of $M_{ise,2} = 0.90$). The boundary conditions are summarized in Table 6.2. After solving the primal solver, the three objective functions are evaluated at the outlet of the domain: a performance metric that is proportional to the entropy generation J_1 (Eq. 6.14), the exit flow angle J_2 (Eq. 6.15) and the outlet mass flow J_3 (Eq. 6.16). Two types of constrained optimization are considered in this chapter. In the first one, the objective is to reduce the J_1 whilst keeping J_2 above 74.83°. In the second one, the same criteria are considered but in addition the exit mass flow J_3 variation should not be greater than ($\pm 0.12\%$) of the baseline mass flow $8.4595 \cdot 10^{-3} kg/s$.

$$J_1 = \frac{\int_{out} p\rho^{1-\gamma} V_x dy}{\dot{m}_{out}} \tag{6.14}$$

$$J_2 = atan\left(\frac{\overline{V_y}}{\overline{V_x}}\right)_{out} \tag{6.15}$$

$$J_3 = \dot{m}_{out} \tag{6.16}$$

Inlet	Outlet	Units
$P_{01} = 147500.0$	$p_2 = 87000.0$	[Pa]
$T_{01} = 420.0$		[K]
$atan\left(\frac{V_y}{V_x}\right) = 0.0$		[deg]

Table 6.2.: *CFD Boundary conditions of the LS89 at an outlet isentropic Mach number of* $M_{ise,2} = 0.90$

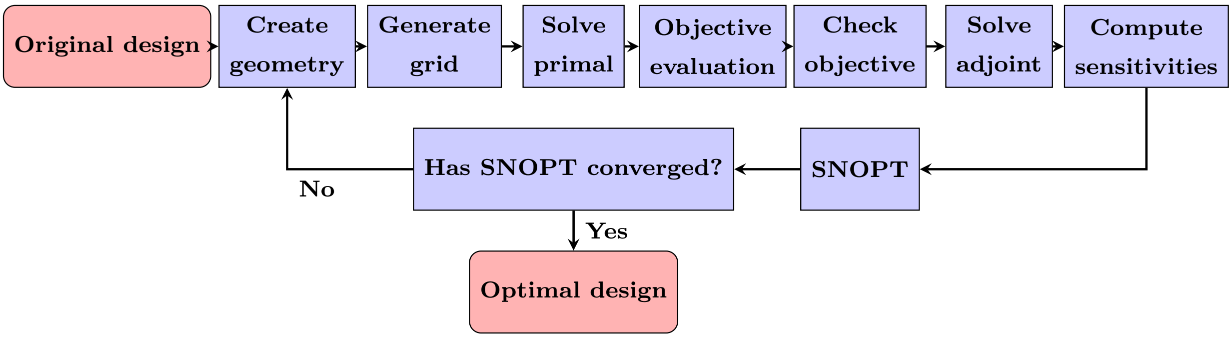


Figure 6.11.: *Optimization flow chart*

In the optimization process shown in Fig. 6.11, the performance sensitivities w.r.t CAD parameters $dJ/d\vec{\alpha}$ are computed using Eq. 6.2. The performance sensitivities for the cost function and constraint w.r.t. the volume grid $dJ/d\vec{X}$ are provided by the hand-derived discrete adjoint solver, whereas the grid sensitivities $d\vec{X}/d\vec{\alpha}$ are provided by the AD tool ADOL-C [9], which is applied to the CAD kernel and grid generator. The computation of $d\vec{X}/d\vec{\alpha}$ for every optimization step is done using the forward vector mode of AD, which allows to obtain the derivatives w.r.t. to the design variables in one single evaluation of the differentiated code at a relatively low cost.

The convergence of SNOPT is achieved when one of the following criteria is satisfied: (1) the Karush-Kuhn-Tucker (KKT) first-order optimality condition [100], satisfying a minor feasibility tolerance of 10^{-6}; (2) the optimizer being unable to improve the objective function; (3) when we have reached the maximum number of major iterations.

6.3. Results

In this section a comparison is made between the parametric effectiveness (PE) defined in this thesis and that one defined by [106]. After, the PE defined in this chapter is computed for different sets and subsets of design variables for different types of constrained optimizations. Finally, some optimizations are performed using parameterizations with different initial PE and the results are compared and benchmarked.

6.3.1. Parametric effectiveness methodology comparison

The idea behind the computation of parametric effectiveness is to rate the ability of a set of parameters to produce a design movement that is aligned with the one dictated by the performance sensitivities w.r.t. the surface grid. In the methodology described in this chapter, the parametric effectiveness is unity if the best possible design movement that can be achieved with the set of CAD based parameters matches the performance sensitivities w.r.t. the surface grid. The method described by [106] is different to the method described in this chapter, mainly because in [106] the authors assume that the design parameters are perturbed in the steepest descent direction and the constraint of overall boundary movement (OBM) is imposed.

In this section, the exact surface sensitivities provided by AD were used for both methodologies, in order to exclude any possible differences originating from design velocities. The right hand side picture of Fig. 6.12 shows the displacement dX that every grid node would move if we allow every grid node to move in the direction dictated by the performance sensitivities w.r.t. the surface grid by an arbitrary step size γ. The step size was chosen such that it normalizes the shape movement to unity (i.e., $\gamma = 1/\|dJ/d\vec{X}_S\|$). This displacement field or shape change is also represented by the graph on the left of Fig. 6.12 by the black line. If the CAD parameterization allows every grid node to displace by the amount indicated by the black line, the parametric efficiency would be unity. The red and green lines represent the assumed best possible shape change that can be achieved with the existing set of CAD parameters, with the methodology defined by [106] and in this chapter respectively to compute parametric effectiveness. The green line matches better the black line because the approach defined by [106] unfairly penalises some CAD parameters when computing the parametric effectiveness due to the OBM constraint and the choice of the steepest descent direction in parameter space.

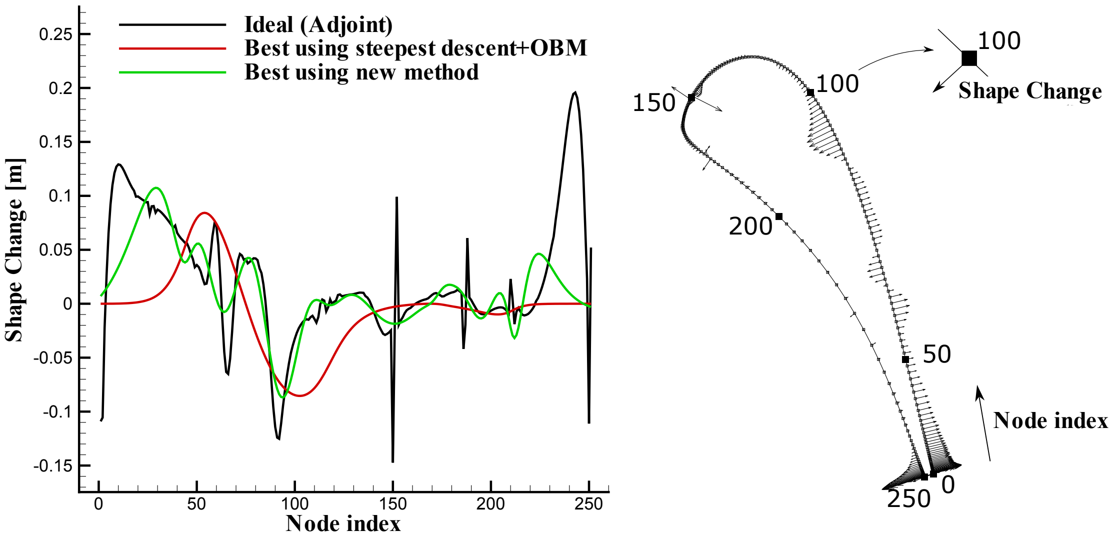


Figure 6.12.: Shape change comparison (left) and adjoint surface sensitivities (right).

Suppose the objective of the unconstrained problem is to reduce the entropy generation and the designer would like to know which CAD-based parameterization from Table 6.1 is the best one to use for the optimization. The parametric effectiveness can be used to rate the different parameterizations and to aid the designer to choose the parameterization that would be most effective for the optimization. Table 6.3 compares the parametric effectiveness between the different parameterizations.

	η_{global}	$PE(V_n)$	$PE(AD)$
Set SS9PS9 (43DV)	73.58 %	9.87 %	9.84%
Set SS14PS4 (43DV)	68.53 %	10.60 %	15.93%
Subset 1 (33DV)	64.18 %	3.07 %	9.77%
Subset 2 (20DV)	52.07 %	9.21 %	9.76%
Subset 3 (16DV)	49.96 %	9.48 %	9.77%
Subset 4 (11DV)	44.78 %	0.46 %	0.52%
Subset 5 (7DV)	40.06 %	0.80 %	0.75%

Table 6.3.: *Parametric effectiveness comparison between different CAD-based parameterizations for an unconstrained problem where the objective is to reduce the entropy generation*

The first column of Table 6.3 shows the parametric effectiveness η_{global} defined in this chapter (Eq. 6.13). The second column of Table 6.3 shows the parametric effectiveness using the methodology described in [106] in combination with the design velocity approach. Finally, the third column of Table 6.3 shows the parametric effectiveness using the methodology described in [106] with the AD grid sensitivities instead of the design velocities. The $PE(AD)$ should give more accurate results than the $PE(V_n)$ because the AD grid sensitivities are accurate to machine accuracy, whereas the design velocities include finite difference inaccuracies and surface to surface mesh projection errors. According to the results shown in the first column of Table 6.3, set SS9PS9 is the best performing set to produce a design movement aligned with the adjoint sensitivities. However, the PE shown in the second and third columns suggests that set SS14PS4 would be the most performing one. If we compare the η_{global} of subsets 1 to 5, subset 1 is predicted to be the best performing subset, because it contains all the design parameters of all the other subsets. In contrast, the $PE(V_n)$ suggests that subset 3 would be the most performing one instead. However, if PE is computed using the AD grid sensitivities instead of the design velocities, subset 1 and 3 would be equally performing, followed by subset 2. The differences between the third and the second column for subsets 1 and 3 show that the sources of inaccuracies present in $PE(V_n)$ can have a large impact on the final value of the PE. These inaccuracies can bias the designer to choose a subset that seems to have high PE but in reality has a lower PE when the inaccuracies are removed.

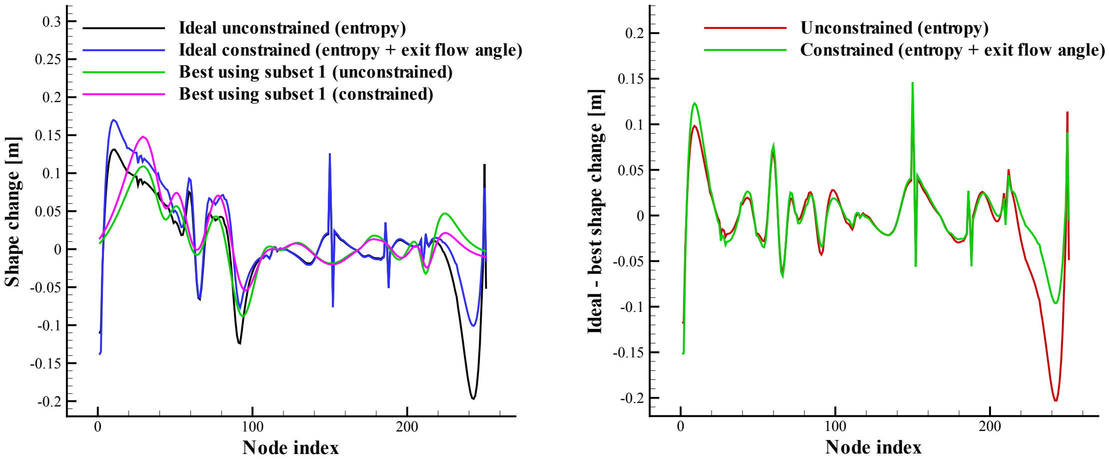
6.3.2. Parametric effectiveness in an aerodynamically constrained problem

The parametric effectiveness shown in Table 6.3 is computed for an unconstrained problem where we want to reduce the entropy generation. However, if we launch an unconstrained optimization we would find out that the optimizer tries to reduce the turning of the vane until there is no flow turning and hence the flow is axial at the outlet. The design practice in a high pressure turbine is to design the vanes and blades with non-zero flow turning. Therefore, in order to optimize the LS89 it is desired to include, at least, one aerodynamic constraint: the exit flow angle. Also, if we allow the pitch (or the solidity) of the LS89 linear cascade to change during the optimization, we would see that the optimizer reduces the losses and increases the pitch to increase the mass flow in order to maintain the same total to static pressure ratio. However, in a real scenario, the pitch or the number of blades is typically fixed and if we kept this design variable constant during the optimization the mass flow variation would be negligible. Alternatively, keeping the outlet mass flow constant can also be achieved if we allow the pitch to be changed during the optimization but we impose the outlet mass flow as an aerodynamic constraint, as well as the exit flow angle. The question arises whether the rating of the different CAD-based parameterizations, in terms of parametric effectiveness, for an aerodynamically constrained problem will be the same or similar to the rating obtained for an unconstrained problem (Table 6.3).

	η_{global,J_1}	η_{global,J_1J_2}	$\eta_{global,J_1J_2J_3}$
Set SS9PS9 (43DV)	73.58 %	80.03 %	73.64 %
Set SS14PS4 (43DV)	68.53 %	80.51 %	62.83 %
Subset 1 (33DV)	64.18 %	76.26 %	61.33 %
Subset 2 (20DV)	52.07 %	65.49 %	60.24 %
Subset 3 (16DV)	49.96 %	62.38 %	53.23 %
Subset 4 (11DV)	44.78 %	56.86 %	45.09 %
Subset 5 (7DV)	40.06 %	54.82 %	40.98 %

Table 6.4.: *Parametric effectiveness comparison between the unconstrained and constrained optimization problems. The effect of the entropy generation* J_1 *is captured in* η_{global,J_1}. η_{global,J_1J_2} *captures the effects of both* J_1 *and the exit flow angle* J_2. $\eta_{global,J_1J_2J_3}$ *captures the effects of* J_1, J_2 *and the outlet mass flow* J_3.

Table 6.4 shows the parametric effectiveness for the unconstrained problem (η_{global,J_1}), the constrained problem with the exit flow angle (J_2) only constraint (η_{global,J_1J_2}) and the constrained problem with both the exit flow angle and outlet mass flow (J_3) constraints ($\eta_{global,J_1J_2J_3}$). The second column shows that, when we include the exit flow angle as an aerodynamic constraint, set SS14PS4 is the most performing set instead of set SS9PS9. The black and blue lines in Fig. 6.13a show the ideal shape change for the unconstrained and constrained problems respectively, with the exit flow angle only constraint in the constrained problem. The addition of the exit flow angle constraint changes the performance sensitivities w.r.t. the surface grid by: (1) increasing their magnitude on the suction side close to the trailing edge (node index 5 - 50 in Fig. 6.13a); (2) reducing their magnitude on the pressure side close to the trailing edge (node index 220 - 250 in Fig. 6.13a). The green and pink lines in Fig. 6.13a show the best shape change using subset 1 for the unconstrained and constrained problems respectively. The mismatch between the ideal and the best shape change using subset 1 is represented in Fig. 6.13b. The mismatch is reduced at the node index 220 - 250 for the constrained problem with the exit flow angle only constraint, which explains why the parametric effectiveness shown in the second column of Table 6.4 is, on average, 11.9% higher than in the first column. The maximum increase in parametric effectiveness is for subset 5, where η_{global,J_1J_2} is 14.8% higher than η_{global,J_1}. The shape modes responsible for this increase in parametric effectiveness are those produced by the following CAD parameters: the trailing edge wedge angle at the pressure side, the outlet metal angle and the stagger angle. Finally, the parametric effectiveness shown in the third column of Table 6.4 considers both the exit flow angle and the outlet mass flow as aerodynamic constraints. The values of $\eta_{global,J_1J_2J_3}$ are relatively similar to those predicted for the unconstrained problem. This is because the constrained ideal shape change with the two aerodynamic constraints is very similar to the unconstrained shape change for the analysed baseline geometry.



(a) Shape change comparison between the unconstrained and constrained problem with the exit flow angle as constraint

(b) Shape change mismatch between the ideal and the best shape change using subset 1

Figure 6.13.: Shape change for the unconstrained and the constrained problem with the exit flow angle (left) and mismatch between ideal and best shape change using subset 1 (right).

6.3.3. Optimization with one aerodynamic constraint

The two sets of 43DV (SS14PS4 and SS9PS9), subset 1 (33DV) and subset 3 (16DV) were selected for optimization with the exit flow angle as an aerodynamic constraint. Figure 6.14 shows the evolution of the non-dimensional cost functions for each set/subset and excludes the non-derivative line searches. The convergence of the optimizer was achieved in all four cases. The optimization using subset 3 was the fastest to converge, probably due to the fact that it uses lower number of DV than subset 1 and sets SS14PS4 and SS9PS9.

Table 6.5 shows that the optimizer was able to reduce the entropy generation term J_1 by more than 20% in all cases whilst keeping the exit flow angle within reasonable limits.

	Set SS14PS4 optimal	Set SS9PS9 optimal	Subset 1 optimal	Subset 3 optimal
Entropy generation variation	-23.46 %	-22.19%	-20.59 %	-20.43%
Exit flow angle variation	-0.02%	-0.03%	-0.01 %	-0.03 %
Mass flow variation	50.58%	44.0%	30.69 %	31.19 %
Total pressure losses variation	-23.2%	-23.00%	-20.71 %	-20.71 %

Table 6.5.: *Changes in performance relative to the baseline, for set SS14PS4 and SS9PS9 and subsets 1 and 3 optimal profiles. Results are taken from the optimizations where the exit flow angle is treated as an aerodynamic constraint.*

Figure 6.15 shows the isentropic blade Mach number distribution for the baseline and optimal

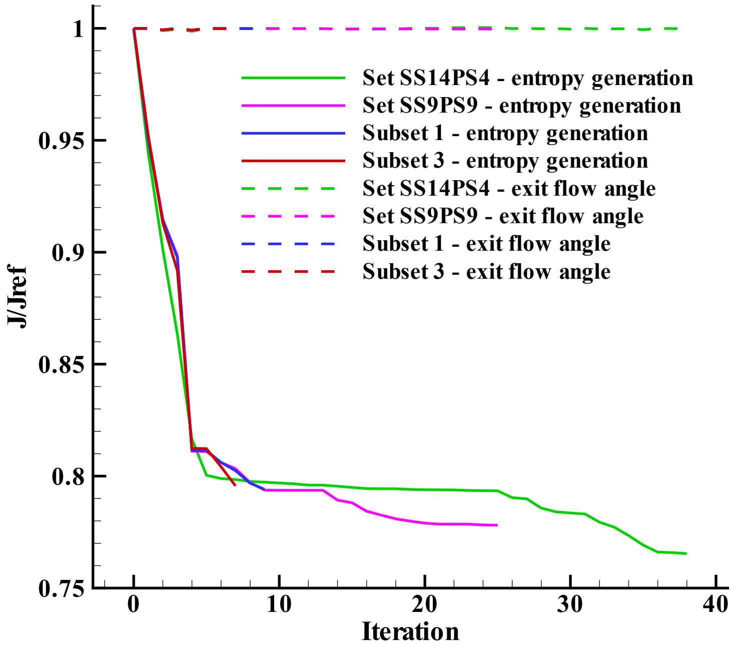


Figure 6.14.: Cost function evolution during the optimization of the LS89 with the exit flow angle treated as an aerodynamic constraint for Sets SS14PS4 and SS9PS9 and Subsets 1 and 3.

designs on the left and the cascade geometries on the right. The optimal geometry for subset 1 and 3 are very similar to each other. However, Table 6.6 shows that there are slight geometrical differences between these two optimal geometries. The density, static pressure, temperature, Mach and entropy generation contours for the baseline and optimal shapes (using set SS9PS9, set SS14PS4, subsets 1 and 3) are shown in Sec. B.1, B.2.3, B.2.4, B.2.5 and B.2.6 respectively.

	Units	Baseline values	Set SS14PS4 change	Set SS9PS9 change	Subset 1 change	Subset 3 change
Rear suction side turning (ϵ)	[deg]	11.6	+14.00°	+11.96°	+10.19°	+9.81°
Pressure side trailing edge wedge angle (φ_{PS})	[deg]	2.500	+0.372 °	+0.355 °	+0.375 °	+0.368 °
Suction side trailing edge wedge angle (φ_{SS})	[deg]	4.000	−0.585 °	+0.422 °	+0.051 °	+0.041 °
Chord (c)	[mm]	64.310	+27.29%	21.60%	+10.56%	+10.89%
Stagger angle (γ)	[deg]	54.925	+8.24 °	+6.87 °	+3.76 °	+3.86 °
Inlet metal angle (β_{in})	[deg]	0.0	+0.41 °	+0.331 °	+0.028 °	+0.025 °
Outlet metal angle (β_{out})	[deg]	74.000	+4.50 °	+4.13 °	+1.79 °	+1.60 °
Leading edge radius (R_{LE})	[mm]	4.126	+15.0%	-5.89%	+0.33%	+0.39%
Solidity (σ)	[-]	1.118	-15.0%	-15.0%	-15.0%	-15.0%
Pitch (g)	[mm]	57.500	+49.75%	43.06%	+30.07%	+30.46%
Pitch/chord (g/c)	[-]	0.894	+17.65%	+17.65%	+17.65%	+17.65%
Trailing edge thickness/throat height	[-]	0.09133	-34.94%	-32.42%	-24.56%	-24.67%

Table 6.6.: *Main geometry characteristics changes of the optimal designs w.r.t the baseline design. Results are taken from the optimizations where the exit flow angle is treated as an aerodynamic constraint.*

Table 6.7 shows that larger entropy reductions were achieved with those CAD-based parameterizations that had a higher initial parametric effectiveness.

6.3.4. Optimization with two aerodynamic constraints

In the previous section, the set SS14PS4 was the most effective CAD-based parameterization to use amongst the other groups shown in Table 6.7, as it allowed to find a final shape with better

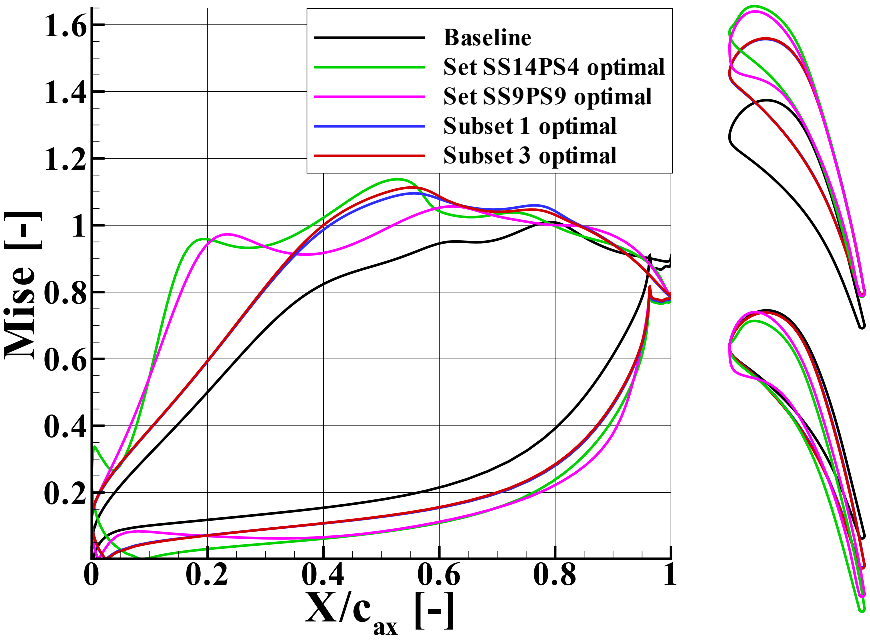


Figure 6.15.: Isentropic Mach number (left) and cascade geometry (right) between the baseline and the optimal shapes for the sets SS14PS4 and SS9PS9 and the subsets 1 and 3. Results are taken from the optimizations where the exit flow angle is treated as an aerodynamic constraint.

	η_{global,J_1J_2}	Entropy generation variation
Set SS14PS4 (43DV)	80.51 %	-23.46 %
Set SS9PS9 (43DV)	80.03 %	-22.19 %
Subset 1 (33DV)	76.26 %	-20.59 %
Subset 3 (16DV)	62.38 %	-20.43 %

Table 6.7.: *Parametric effectiveness and entropy generation variation of the optimal design w.r.t. the baseline. Results are taken from the optimizations that have the exit flow angle as an aerodynamic constraint.*

performance than those obtained with the other sets and subsets. However, the $\eta_{global,J_1J_2J_3}$ in Table 6.4 shows that the set SS14PS4 is no longer the most effective CAD-based parameterization when we consider both the exit flow angle and the outlet mass flow as aerodynamic constraints. When both constraints are considered, the set SS9PS9 becomes the most effective one instead. In this section, two optimizations were launched using the sets SS14PS4 and SS9PS9, to confirm which of the two sets is more fit for the optimization of the LS89 where the two aerodynamic constraints are considered.

The evolution of the non-dimensional cost functions during the optimization, excluding the non-derivative line searches, is shown in Fig. 6.16 for each set. The convergence of the optimizer was achieved for both sets.

Table 6.8 shows that the optimizer was able to reduce the entropy generation term J_1 by 15.13% and 14.34% for sets SS9PS9 and SS14PS4 respectively, whilst keeping the exit flow angle and mass flow within reasonable limits.

Figure 6.17 shows the isentropic blade Mach number distribution for the baseline and optimal

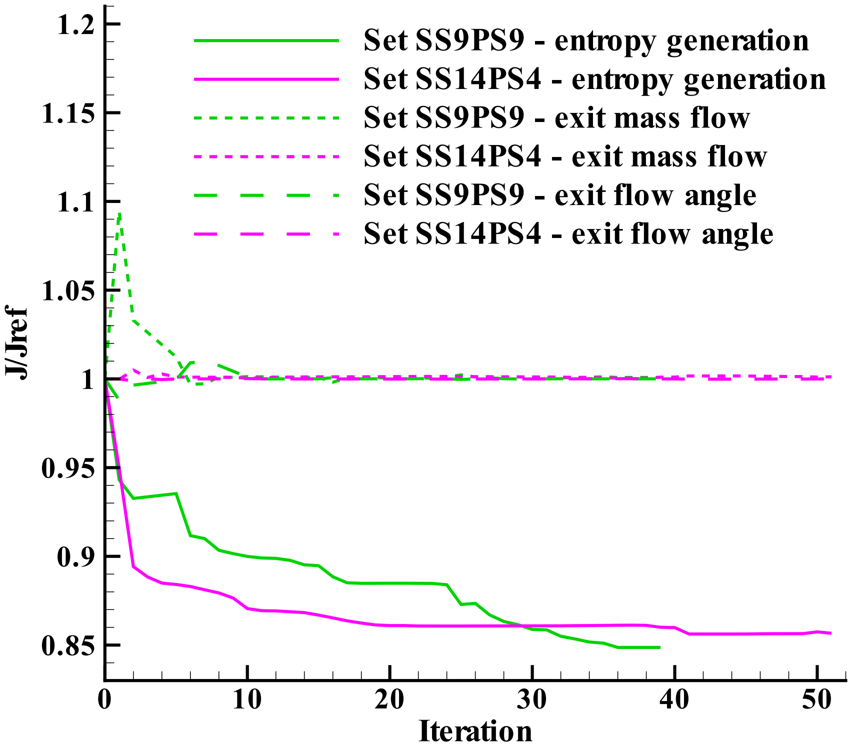


Figure 6.16.: Cost function evolution during the optimization of the LS89 using different sets SS9PS9 and SS14PS4 of design variables. The exit flow angle and the exit mass flow as treated as aerodynamic constraints.

	Set SS9PS9 optimal	Set SS14PS4 optimal
Entropy generation variation	-15.13 %	-14.34%
Exit flow angle variation	0.002 %	-0.006 %
Mass flow variation	0.081 %	0.118 %
Total pressure losses variation	-14.49 %	-14.09 %

Table 6.8.: *Changes in performance relative to the baseline, for sets SS9PS9 and SS14PS4 optimal profiles. Results are taken from the optimizations where the exit flow angle and the exit mass flow are treated as aerodynamic constraints.*

designs on the left and the cascade geometries on the right. Table 6.9 shows the main geometrical differences between the optimal and baseline geometries. The density, static pressure, temperature, Mach and entropy generation contours for the baseline and optimal shapes (using sets SS9PS9 and SS14PS4) are shown in Sec. B.1, B.3.1 and B.3.2 respectively.

Table 6.10 shows that largest entropy reduction was achieved with the set SS9PS9, the CAD-based parameterizations with the highest parametric effectiveness $\eta_{global,J_1J_2J_3}$ at the beginning of the optimization. This confirms that the set SS14PS4 was indeed less effective for the optimization with the two aerodynamic constraints than the set SS9PS9. Therefore, a set of CAD parameters can be more effective for one type of constrained optimization and be less effective for another type where other aerodynamic constraints are considered. The results shown in Table 6.10 demonstrate again that the CAD-based parameterization with the highest parametric effectiveness should be

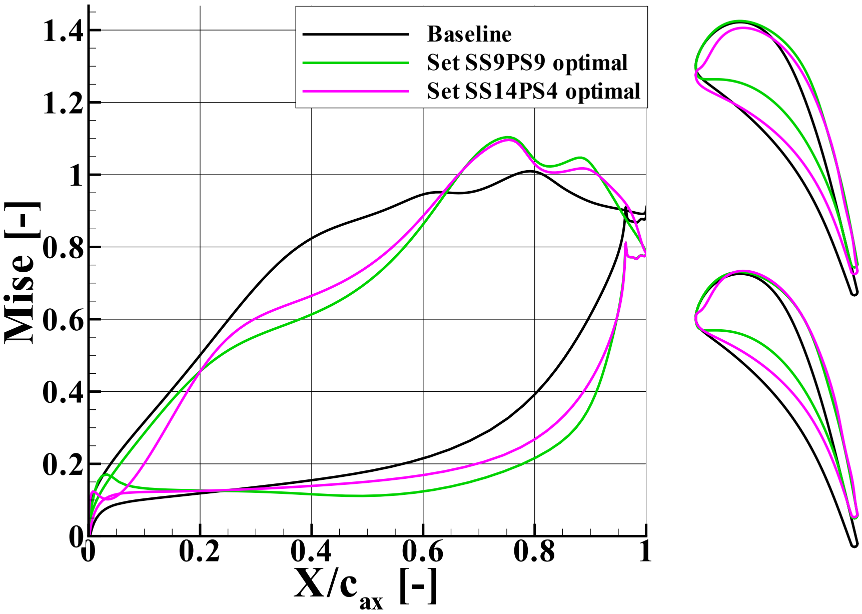


Figure 6.17.: Isentropic Mach number (left) and cascade geometry (right) between the baseline and the optimal shapes for sets SS9PS9 and SS14PS4. Results are taken from the optimizations where the exit flow angle and the outlet mass flow are treated as aerodynamic constraints.

chosen to be used for optimization. This is because a CAD-based parameterization with a high parametric effectiveness is more able to produce the boundary changes that the adjoint solution dictates and hence larger performance improvements are expected to be found by the optimizer.

6.3.5. Optimization with two aerodynamic constraints using the most effective subset

Figure 6.16 shows a very slow design improvement after 20 iterations when the optimizer uses the set SS14PS4 (43DV). From iteration 20 to 51, the optimizer is only able to further reduce the entropy generation by 0.33%. In order to speed the optimization process it is suggested in this section

	Units	Baseline	Set SS9PS9 optimal	Set SS14PS4 optimal	Variation
Rear suction side turning (ϵ_{SS})	[deg]	11.6	25.486	24.227	13.87 °/12.61 °
Pressure side trailing edge wedge angle (φ_{PS})	[deg]	2.500	2.875	2.772	0.375 °/0.272 °
Suction side trailing edge wedge angle (φ_{SS})	[deg]	4.000	4.600	4.047	0.600 °/0.047 °
Chord (c)	[mm]	64.310	58.993	58.768	−8.27%/ − 8.62%
Stagger angle (γ)	[deg]	54.925	51.212	51.036	−3.71 °/ − 3.89 °
Inlet metal angle (β_{in})	[deg]	0.0	-0.477	-0.124	−0.477 °/ − 0.124 °
Outlet metal angle (β_{out})	[deg]	74.000	77.479	76.753	3.479°/2.753°
Leading edge radius (R_{LE})	[mm]	4.126	3.507	4.207	−15.0%/1.94%
Solidity (σ)	[-]	1.118	1.028	1.024	−8.04%/ − 8.41%
Pitch (g)	[mm]	57.500	57.360	57.372	−0.24%/ − 0.22%
Pitch/chord (g/c)	[-]	0.894	0.972	0.976	8.75%/9.19%
Trailing edge thickness/throat height	[-]	0.09133	0.08775	0.08709	−3.92%/ − 4.65%

Table 6.9.: *Comparison of the main geometry characteristics between the baseline and optimal profiles for sets SS9PS9 and SS14PS4. Results are taken from the optimizations where the exit flow angle and the exit mass flow are treated as aerodynamic constraints.*

	$\eta_{global,J_1J_2J_3}$	Entropy generation variation
Set SS9PS9 (43DV)	73.64 %	-15.13 %
Set SS14PS4 (43DV)	62.83 %	-14.34 %

Table 6.10.: *Parametric effectiveness and entropy generation variation of the optimal design w.r.t. the baseline. Results are taken from the optimizations that have the exit flow angle and the exit mass flow as aerodynamic constraints.*

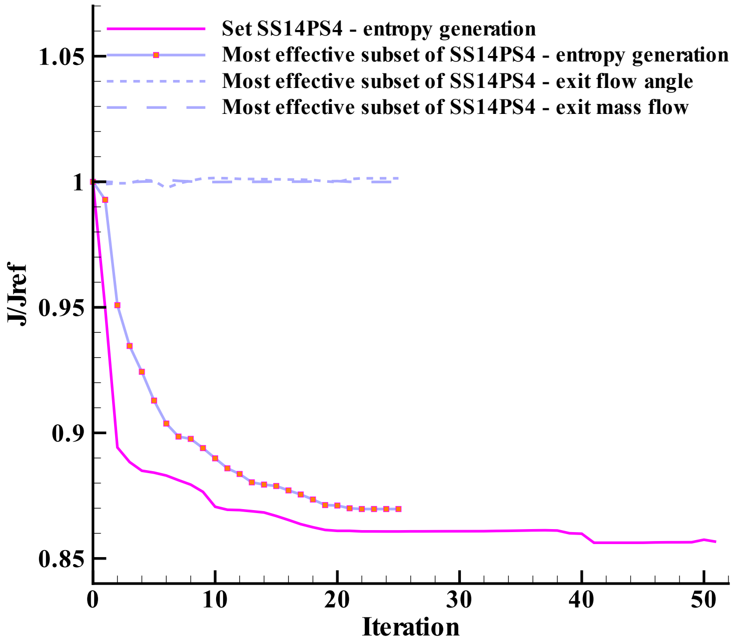


Figure 6.18.: Cost function evolution during the optimization of the LS89 with the exit flow angle and exit mass flow treated as aerodynamic constraints, using set SS14PS4 and its most effective subset.

to use the most effective subset of the set SS14PS4, which contains smaller number of design variables (i.e. 28DV). Table 6.11 shows the individual parametric effectiveness of each parameter in set SS14PS4, for a contrained optimization where the exit flow angle and exit mass flow are treated as aerodynamic constraints. The last column of Table 6.11 highlights that only those parameters with an individual parametric effectiveness above 1% were included in the most effective subset. The parametric effectiveness of the set SS14PS4 and of the most effective subset is 62.83% and 56.15% respectively (see Table 6.12) at the beginning of the optimization. The most effective subset has 6.68% lower parametric effectiveness than set SS14PS4, due to the fact that it contains 15 parameters less.

Figure 6.18 shows the evolution of the non-dimensional cost functions during the optimization, excluding the non-derivative line searches, for the set SS14PS4 and its most effective subset. Using the most effective subset, with a reasonable smaller number of design parameters, resulted in a time saving of approximately 8.3 days (26 iterations less) to complete the optimization on a XeonE5-2690 2.60GHz workstation with 264GB RAM. However, the reduction in the gain in performance is 1.31% (see Table 6.12).

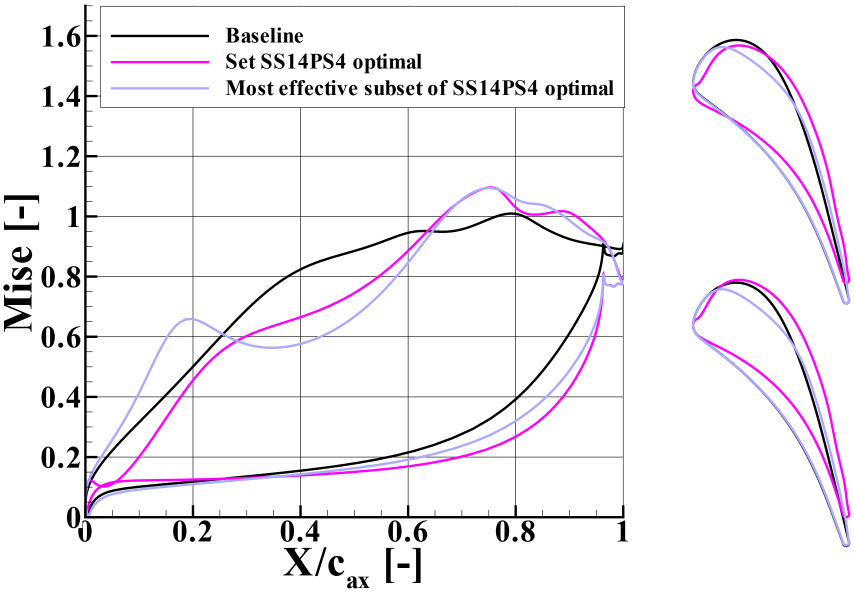


Figure 6.19.: Isentropic Mach number (left) and cascade geometry (right) between the baseline and the optimal shapes for the set SS14PS4 and its most effective subset. Results are taken from the optimizations where the exit flow angle and exit mass flow are treated as aerodynamic constraints.

The isentropic blade Mach number distribution and the cascade geometry are shown in Fig. 6.19 for the baseline and the optimal designs found with set SS14PS4 and its most effective subset. The main geometrical differences between the baseline and the optimal design found with the most effective subset of the set SS14PS4 are shown in Table 6.13. The density, static pressure, temperature, Mach and entropy generation contours for the baseline and optimal shapes (using set SS14PS4 and the most effective subset of the set SS14PS4) are shown in Sec. B.1, B.3.2 and B.3.3 respectively.

6.4. Conclusions

In this chapter, a new definition of parametric effectiveness is proposed to rate the ability of a set of CAD-based design variables to produce a shape change dictated by the adjoint sensitivities. The parametric effectiveness takes into account the fact that the design variables can be non-orthogonal to each other. The developed method is demonstrated for constrained optimization problems, by using the projected gradient method. The AD tool is used to obtain exact surface sensitivities by differentiating the CAD kernel and grid generator. Two types of constrained optimizations were carried out. In the first one, the objective was to reduce the entropy generation whilst keeping the exit flow angle above a target value. The second optimization problem consisted in reducing the entropy generation whilst keeping the exit flow angle above a certain value and the exit mass flow in between a lower and upper limit. The parametric effectiveness was used to rate different groups of CAD-based design variables and some were selected for optimization. Seven optimizations were performed in total, showing that the largest reductions in entropy were achieved when using those groups of CAD-based design variables that exhibited the highest parametric effectiveness at the start of the optimization. Therefore, the parametric effectiveness can be used to assist the designer to decide which is the best CAD-based parameterization to use for optimization. Additionally, the parametric effectiveness can also be used to rate the individual effectiveness of each CAD parameter, in order to exclude the least performing ones from the optimization. Considerable time savings can be achieved by using the most effective subset of parameters for optimization, with a reasonable smaller number of design parameters, at the expenses of a smaller reduction in the gain in performance. In this chapter, the parametric effectiveness was only evaluated at the start of the optimization process to decide which group of CAD parameters was more fit for optimization. However, it is possible that another group of design variables becomes more effective than the chosen one in later stages of the optimization. For this reason, using parametric effectiveness in adaptive shape parameterization is deemed to be a promising area of future research.

	$\eta_{global,J_1J_2J_3}$	include in most effective subset
SS thickness t_{SS}^7	39.28%	Yes
Outlet metal angle β_{out}	32.74%	Yes
SS streamwise position u_{SS}^6	32.74%	Yes
SS streamwise position u_{SS}^7	24.55%	Yes
Stagger angle γ	22.73%	Yes
SS thickness t_{SS}^8	22.66 %	Yes
SS thickness t_{SS}^6	22.20 %	Yes
SS thickness t_{SS}^{14}	18.74 %	Yes
SS thickness t_{SS}^9	18.12 %	Yes
SS thickness t_{SS}^{10}	16.70 %	Yes
SS thickness t_{SS}^{13}	14.79 %	Yes
SS streamwise position u_{SS}^{13}	14.64 %	Yes
SS streamwise position u_{SS}^8	14.59%	Yes
SS streamwise position u_{SS}^9	14.46%	Yes
TE wege angle SS ψ_{SS}	14.15%	Yes
SS streamwise position u_{SS}^{10}	13.57 %	Yes
SS thickness t_{SS}^{11}	12.79 %	Yes
SS thickness t_{SS}^{12}	11.88 %	Yes
SS thickness t_{SS}^5	11.59 %	Yes
SS streamwise position u_{SS}^{12}	11.47%	Yes
SS streamwise position u_{SS}^5	11.17%	Yes
SS streamwise position u_{SS}^{11}	10.86%	Yes
SS streamwise position u_{SS}^{14}	8.07%	Yes
TE wege angle PS ψ_{PS}	6.56%	Yes
SS streamwise position u_{SS}^4	4.70%	Yes
SS thickness t_{SS}^4	3.55%	Yes
Inlet metal angle β_{in}	1.94%	Yes
PS thickness t_{PS}^4	1.61%	Yes
PS streamwise position u_{PS}^4	0.58%	No
SS streamwise position u_{SS}^3	0.53%	No
PS streamwise position u_{PS}^3	0.35%	No
SS streamwise position u_{SS}^1	0.35%	No
SS thickness t_{SS}^3	0.34 %	No
SS thickness t_{SS}^1	0.25 %	No
SS streamwise position u_{SS}^2	0.15%	No
PS streamwise position u_{PS}^1	0.14%	No
PS thickness t_{PS}^2	0.11%	No
PS thickness t_{PS}^1	0.06%	No
SS thickness t_{SS}^2	0.06 %	No
PS thickness t_{PS}^3	0.03%	No
PS streamwise position u_{PS}^2	0.03%	No
LE radius R_{LE}	0.02%	No
Solidity σ	0.01%	No

Table 6.11.: *Individual parametric effectiveness of each parameter in the set SS14PS4. The results shown are obtained by considering that the CAD-based parameterization only contains only one parameter at the time. The exit flow angle and exit mass flow are considered as aerodynamic constraints.*

	$\eta_{global,J_1J_2J_3}$	Entropy generation variation	Optimization iterations
Set SS14PS4 (43DV)	62.83 %	-14.34 %	51
Most effective subset of SS14PS4 (28DV)	56.15%	-13.03%	25

Table 6.12.: *Parametric effectiveness, entropy generation variation and iterations to complete the optimization using set SS14PS4 and its most effective subset. Results are taken from the optimizations where the exit flow angle and the exit mass flow are treated as aerodynamic constraints.*

	Units	Baseline	Most effective subset of SS14PS4 optimal	Variation
Rear suction side turning (ϵ_{SS})	[deg]	11.6	21.29	9.67 °
Pressure side trailing edge wedge angle (φ_{PS})	[deg]	2.500	2.875	0.375 °
Suction side trailing edge wedge angle (φ_{SS})	[deg]	4.000	3.586	−0.413 °
Chord (c)	[mm]	64.310	64.183	-0.20%
Stagger angle (γ)	[deg]	54.925	54.845	−0.15 °
Inlet metal angle (β_{in})	[deg]	0.0	0.470	0.470 °
Outlet metal angle (β_{out})	[deg]	74.000	74.105	0.105 °
Leading edge radius (R_{LE})	[mm]	4.126	4.126	0.0%
Solidity (σ)	[-]	1.118	1.118	0.0%
Pitch (g)	[mm]	57.500	57.386	-0.20%
Pitch/chord (g/c)	[-]	0.894	0.894	0.0%
Trailing edge thickness/throat height	[-]	0.09133	0.08805	−3.59%

Table 6.13.: *Comparison of the main geometry characteristics between the baseline and optimal profiles for the most effective subset of the set SS14PS4. Results are taken from the optimizations where the exit flow angle and the exit mass flow are treated as aerodynamic constraints.*

Chapter 7.

Efficient CAD-based adjoint optimization using parametric effectiveness for multilevel shape refinement

In chapter 6, a novel methodology is presented to compute the parametric effectiveness without any constraint on the boundary movement and the direction of gradient, and is developed to deal with constrained optimization problems. In general, a high parametric effectiveness at the beginning of the optimization allowed the optimizer to converge to a better solution. However, using a static or fixed parameterization during the course of the optimization may not be the most efficient strategy, specially when it is desired to explore rich design spaces whilst maintaining a good convergence rate of the optimizer. This chapter addresses this and presents a novel multilevel optimization strategy.

Abstract. This chapter presents an efficient optimization strategy, in which the design variables are defined within a CAD system and the optimizer explores higher-dimensional design spaces by using a multi-level shape refinement approach. CAD based approaches have the advantage to represent the optimal shape in the industry accepted format for manufacturing, however they restrict the design space compared to node based approaches where each single mesh point is free to move. In order to preserve the rich design space of the node based approach, we develop an automatic framework which refines the CAD model and creates the best parameterization to be used during the optimization. For this purpose, parametric effectiveness is computed to rate the CAD-based parameterization in terms of its ability to produce the design movement dictated by the adjoint sensitivities. The novelty lies in the definition of the parametric effectiveness and using it to refine the CAD model parameterization in a multilevel optimization process. The parametric effectiveness takes into account the non-orthogonality of the design variables and uses accurate geometric sensitivities provided by algorithmic differentiation. The method is tested on the LS89 turbine cascade and large aerodynamic improvements in the entropy generation are achieved whilst keeping the exit flow angle fixed. The trailing edge and axial chord length are kept fixed as manufacturing constraints. The flow is governed by the Reynolds-averaged Navier-Stokes equations and the one-equation Spalart-Allmaras turbulence model.

7.1. Introduction

The success of a CAD-based optimization framework highly depends on the expertise and skill of the designer, and often it is not obvious which parameters should be chosen for the optimization to achieve the desired shape change dictated by the adjoint sensitivities. Furthermore, the choice of the CAD parameters constrains how the model boundary can change, and a poor choice may prevent the exploration of innovative geometries. It is therefore of vital importance to quantify how suitable is a set of CAD-based design parameters for use in optimization.

In this context, adaptive shape parameterization or multilevel optimization [77, 78, 108], where the optimization starts with a small number of design variables (coarse parameterization) and ends with a large number (fine parameterization), can be an effective strategy to improve the convergence rate, robustness and final solution of an optimization. The exploration of richer design spaces in a CAD-based multilevel optimization framework which uses B-splines can be done using the knot insertion algorithm [77, 78], since it allows to introduce new control points (i.e. design variables) while maintaining the shape. In practice, it is unclear how many new design variables should be introduced at every new design space or how good is every new parameterization to produce the shape changes that are required by the adjoint sensitivities. In this context, the parametric effectiveness can be used to find the appropiate parameterization to use in each new dimensional design space and make the optimization a more autonomous, robust and efficient design strategy. In this chapter, the methodology presented in chapter 6 to compute the parametric effectiveness is used in a multilevel optimization of LS89 turbine cascade [117, 118], where the CAD parameterization is refined during the course of the optimization.

The remainder of the chapter is organised as follows. Section 7.2 describes the proposed adaptive shape parameterization which is assisted by the parametric effectiveness. The optimization results are presented in Sec. 7.3. Finally, the chapter terminates with the main conclusions.

7.2. Methodology

Throughout this chapter, the parametric effectiveness is used in the context of adaptive shape parameterization or multilevel optimization. The idea is that we start the optimization with a coarse design space with a low number of CAD design variables and automatically transition to finer parameterization design spaces at strategic moments in order to find better optima. When refinement of the design space is needed, we insert new CAD features in the CAD model in a systematic manner and use the global parametric effectiveness to choose the best CAD-based parameterization in order to continue with the optimization.

7.2.1. Optimization

The same parameterization of the LS89 from Sec. 6.2.5 is used in this chapter. The same methods and CFD boundary conditions described in Sec. 6.2.7 for the optimization of the LS89 at design point using one aerodynamic constraint are used in this chapter.

The convergence of SNOPT is achieved when one of the following criteria is satisfied: (1) the Karush-Kuhn-Tucker (KKT) first-order optimality condition [100], satisfying a minor feasibility tolerance of 10^{-6}; (2) the optimizer being unable to improve further the objective function; (3) when we have reached the maximum number of major iterations ($N_{max} = 50$).

In this chapter, a multi-level optimization process is developed (Fig. 7.1). Here, once the first level of SNOPT is converged, an additional level of optimization is performed by refining the design space and increasing the number of design variables. At this stage, SNOPT is restarted with the new set of parameters. The multilevel optimization algorithm stops when one of the following criteria is satisfied:

1. Convergence condition 1: when the number of design variables is equal to the maximum number allowed by the user.
2. Convergence condition 2: when both the number of control points on the pressure side and suction side B-Splines reaches the maximum number allowed by the user.
3. Convergence condition 3: when the number of refinement cycles that we have explored is equal to the maximum number specified by the user.

For the convergence condition 1 the maximum number of design variables can be used for instances where the optimization algorithm has a limit on the number of design variables used. The convergence condition 2 is used to limit the maximum number of B-Spline control points to the number of mesh points used to discretize the PS ($PSCntrlPnts_{max} = 85$) and SS ($SSCntrlPnts_{max} = 171$) of LS89 geometry for the CFD analysis. This is essential to have sufficiently fine mesh to capture the shape changes introduced by the parametric perturbations. The convergence condition 3 is simply added to specify the maximum number of refinement cycles that we want to perform during the multilevel optimization, it was set to 30.

7.2.2. Refinement strategy

The initial LS89 geometry has B-spline curves with multiple knots at the start and end of the parametric space. The pressure and suction side has curves of the order of 12 and 7 for the PS and SS respectively. The knot insertion algorithm [76] is used to refine the design space since it allows to add additional control points (i.e. design variables) without changing the shape of the curve.

The refinement procedure, which is illustrated in Fig. 7.2, adds one additional control point at the pressure and suction sides (if the number of control points does not exceed the maximum allowed for each side) and computes the gain in parametric effectiveness due to the introduction of the new control points. The procedure keeps on inserting knots on the pressure and suction side B-spline curves in an iterative fashion until one of the following criteria is met:

- the gain in effectiveness between the last two refinements is smaller than a certain threshold.
- the number of control points on the pressure and suction side B-spline curves has reached the maximum limit.

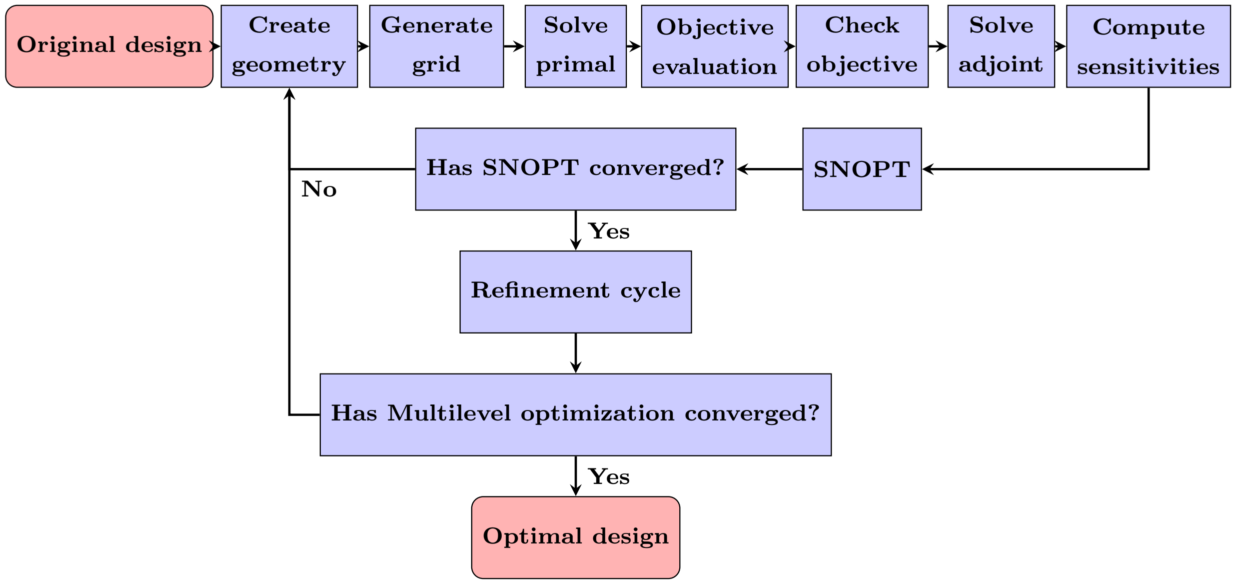


Figure 7.1.: *Multilevel optimization flow chart*

This method allows to increase the parametric effectiveness iteratively until very little gains are observed by the addition of control points. This guarantees that we do not introduce an excessive number of design variables while refining the design space.

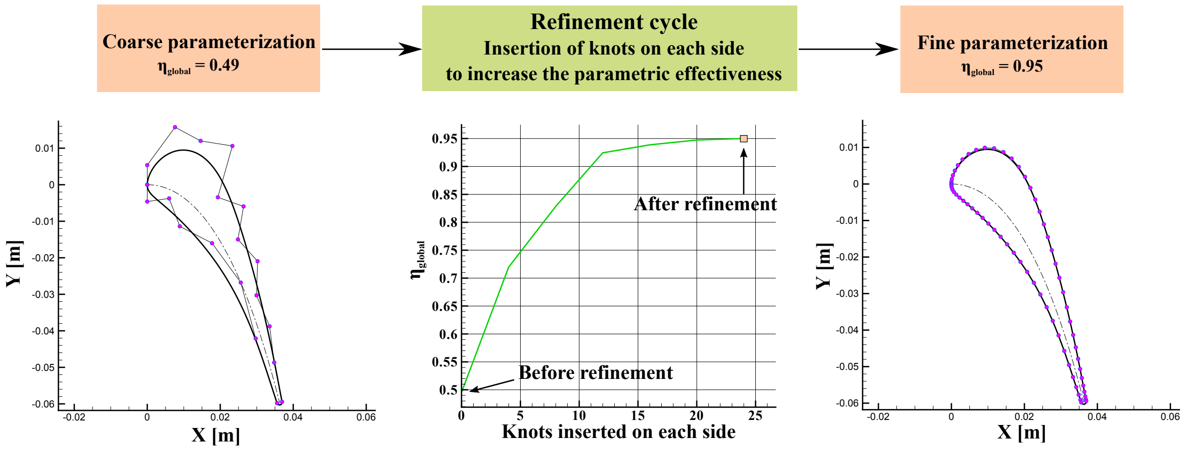
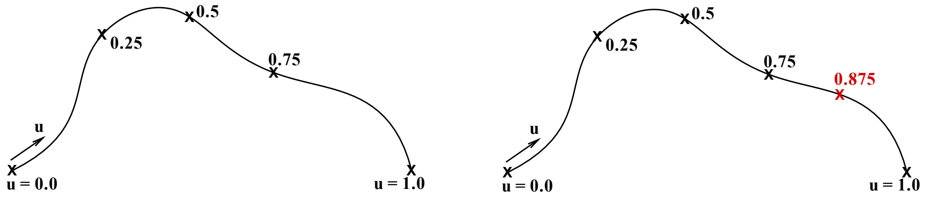


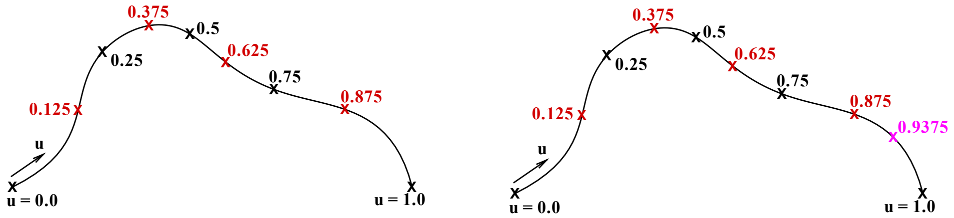
Figure 7.2.: Refinement cycle of the CAD-based parameterization.

The strategy to insert knots in a B-spline curve during the refinement cycle consists in inserting the new knots in between the last two already existing knots, by placing the new knot in the middle. This strategy is illustrated in Fig. 7.3. Suppose that we have the B-spline curve shown in Fig. 7.3a, which has four knots equally spaced, as indicated by the black crosses. If we want to insert a new knot to the B-spline curve of Fig. 7.3a, the new knot should be placed in between the knot at 0.75 and 1.0, as indicated by the red cross in Fig. 7.3b. Figure 7.3c shows the insertion of 4 knots (i.e. red crosses), which are inserted in between the already existing ones. Therefore, the strategy consists in inserting knots progressively from the end ($u = 1.0$) to the beginning ($u = 0.0$) of the knot vector. However, if the knots have already been inserted through the full span of the knot vector and it is required to insert additional ones, then the additional knots are inserted into the knot



(a) Initial knot positions

(b) Knot positions after inserting 1 knot



(c) Knot positions after inserting 4 knots

(d) Knot positions after inserting 5 knots

Figure 7.3.: Strategy to insert knots in the B-spline curves during the refinement cycle.

vector by starting from the end of the knot vector again. This is illustrated in Fig. 7.3d, where the position of the 5th new knot (i.e. the pink cross) is in between the last two knots (i.e. $u = 0.875$ and $u = 1.0$).

7.3. Results

Figure 7.4 shows the evolution of the non-dimensional cost functions during the single level and multilevel optimizations. Note that the functional evaluations during the line search are not shown. The solid and dashed lines represent the evolution of the entropy generation and the exit flow angle respectively. The multilevel optimization started from the coarsest level with 33 design variables (DV) and the design space was progressively refined to contain 97, 137, 193 and 273 DV. The arrows and the numbers on top of the green line indicate the moments at which the design space refinement occurred and the number of design variables for the refined design spaces. The single level optimization was performed after the multi-level optimization had converged using the same 273 DV that were used in the multi-level optimization finest level from the start.

For both single level and multilevel approaches, the optimizer was able to improve significantly the entropy generation of the turbine cascade whilst keeping the exit flow angle within a reasonable limit of ±0.5% (see Table 7.1).

After SNOPT finds the optimal shape on the coarsest level (33 DV), it is necessary to proceed to refine the design space. Figure 7.5 shows the results of performing the design space refinement from the coarsest design space (33 DV) to the next design space (97 DV). The left figure monitors the evolution of the parametric effectiveness as the knots are inserted on the pressure and suction side B-spline curves simultaneously. The parametric effectiveness before the refinement was 0.49

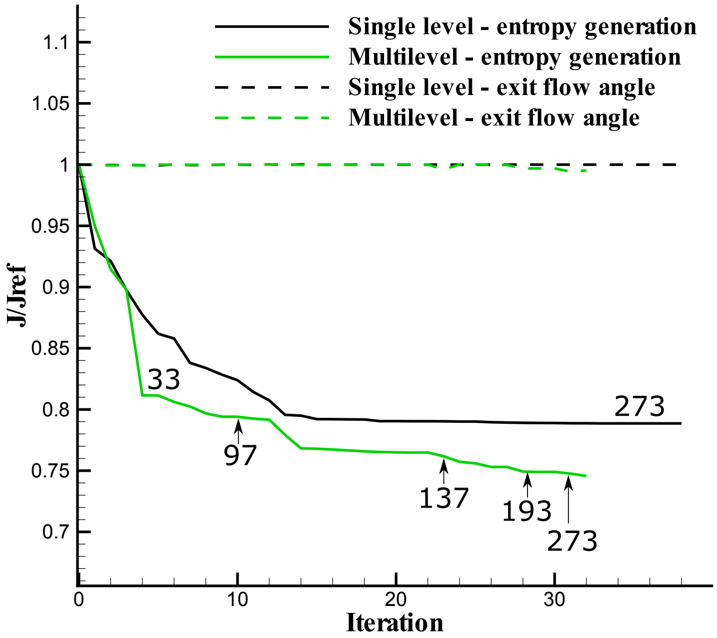


Figure 7.4.: Cost function evolution during the optimization. Numbers represent the number of design variables at each optimization phase.

	Single level optimal	Multilevel optimal
Entropy generation variation	-21.14 %	-25.43%
Exit flow angle variation	-0.0 %	-0.48 %
Mass flow variation	31.89 %	63.29 %
Total pressure losses variation	-22.34 %	-26.55 %

Table 7.1.: *Changes in performance relative to the baseline*

and after inserting 16 knots on each side, the effectiveness increases to 0.94. The right figure of Fig. 7.5 shows the boundary movement for the parameters before and after the shape refinement, and can be seen that refining the design space enabled the shape change to match more closely the performance sensitivities w.r.t. the surface grid (green line).

Figure 7.6 shows the comparison between the baseline geometry and the single and multi-level optimization results. The left figure shows the isentropic Mach number distribution for the baseline and optimal profiles for the downstream isentropic Mach number of 0.9. For both single level and multilevel optimal profiles, the flow is rapidly accelerated towards a higher Mach number than the baseline design. This is followed by deceleration until the flow exits the trailing edge at a lower Mach number ($X/c_{ax} = 1$). The figure on the right shows the baseline and optimal cascades profiles and Table 7.2 shows a summary of the cascade geometry parameter changes. The largest improvements were made for the multilevel strategy, which also showed a much higher convergence rate than the single level case. The use of a large number of design variables in the single level case led to poor convergence and a lower optimum than the multilevel case. This has also been observed in [108] for the inverse design optimizations of NACA4410 and ONERA M6.

Typically, for blades operating in the transonic regime from $M = 0.8$ to 1.2 the major source of loss

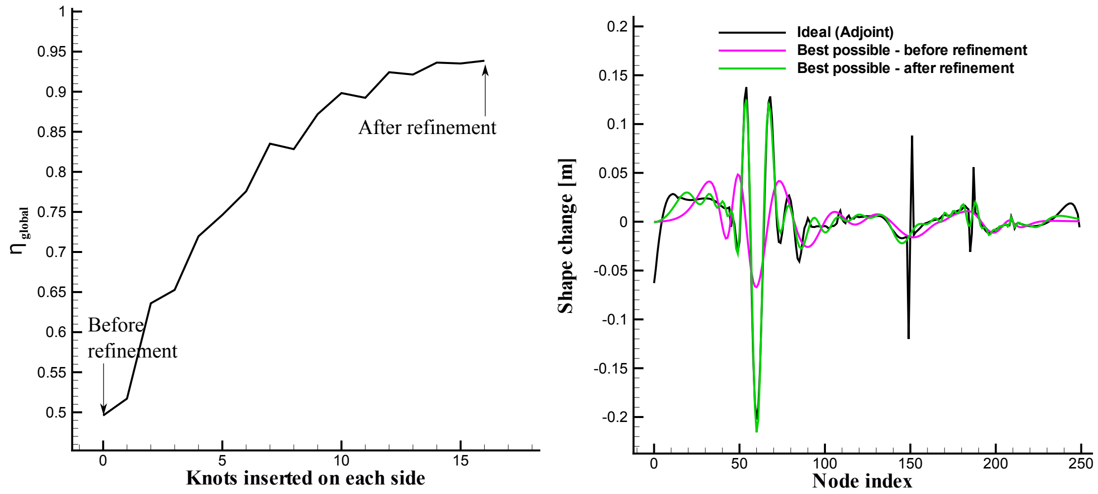


Figure 7.5.: Global parametric effectiveness as we insert knots on the PS and SS B-spline curves simultaneously to refine the design space (left), and shape change comparison before and after refinement (right).

	Units	Baseline	Single level optimal	Multilevel optimal	Variation
Rear suction side turning (ϵ_{SS})	[deg]	11.6	20.5	25.2	8.9 °/13.6 °
Pressure side trailing edge wedge angle (φ_{PS})	[deg]	2.500	2.508	2.884	0.008 °/0.384 °
Suction side trailing edge wedge angle (φ_{SS})	[deg]	4.000	4.043	4.088	0.043 °/0.088 °
Chord (c)	[mm]	64.310	71.761	81.688	11.59% / 27%
Stagger angle (γ)	[deg]	54.925	59.004	63.102	4.079 °/8.177 °
Inlet metal angle (β_{in})	[deg]	0.0	0.063	0.037	0.063 °/0.037 °
Outlet metal angle (β_{out})	[deg]	74.000	75.270	75.544	1.27 °/1.544 °
Leading edge radius (R_{LE})	[mm]	4.126	4.127	4.141	0.01% / 0.4%
Solidity (σ)	[-]	1.118	0.951	0.894	-15.0% / -20.0%
Pitch (g)	[mm]	57.500	75.484	91.324	31.28% / 58.8%
Pitch/chord (g/c)	[-]	0.894	1.052	1.118	17.65% / 25.0%

Table 7.2.: *Comparison of the main geometry characteristics between the baseline and optimal profiles*

is the trailing edge loss [120, 123]. The lower Mach number at the trailing edge region contributes to a lower entropy generation, as shown by Fig. 7.7. The optimal geometry obtained as the result of multi-level optimization has the lowest entropy generation at the trailing edge, owing to the lowest Mach number at the trailing edge (see Fig. 7.6). The density, static pressure, temperature, Mach and entropy generation contours for the baseline and optimal shapes (using the single level and the multilevel approaches) are shown in Sec. B.1, B.2.7 and B.2.8 respectively.

7.3.1. Alternative refinement strategy

The refinement procedure presented in Sec. 7.2.2 consists in adding new control points by inserting knots to the B-spline curve at chosen positions. Although inserting new knots at the chosen positions tends to increase the parametric effectiveness, the parametric effectiveness could be further increased if the location of the control points (or the knots introduced) is optimized. For this purpose, Châtel et. al. [2] proposed an improved refinement procedure in which the control point position is optimized using the parametric effectiveness defined in Sec. 6.2.4. In this section, a summary of the methodology and the results presented in [2] are presented in order to further

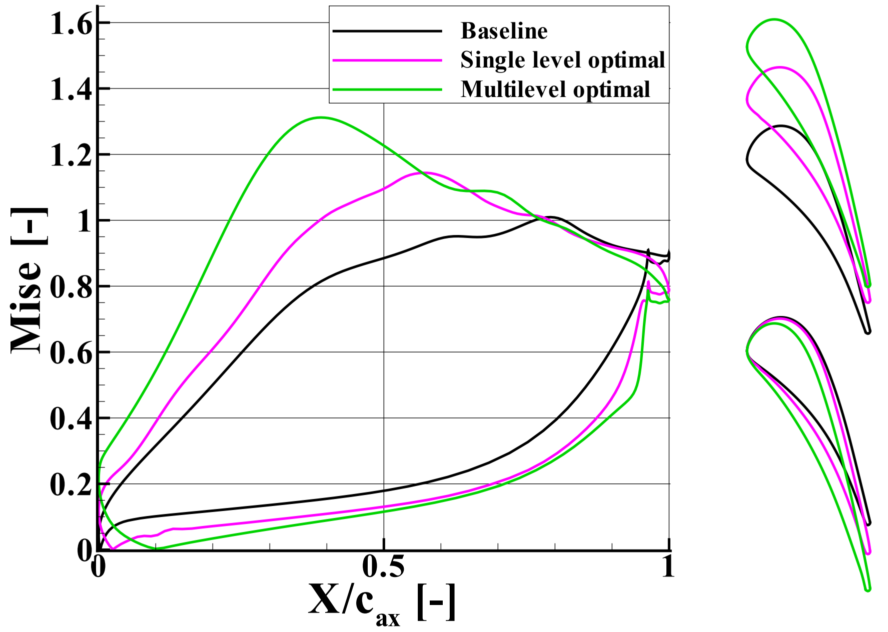


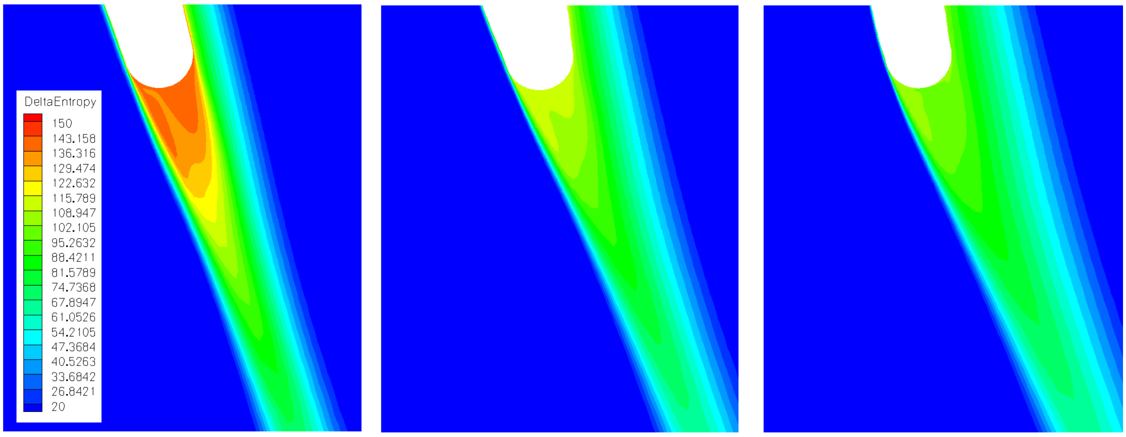
Figure 7.6.: Isentropic Mach number (left) and cascade geometry (right) comparison between the baseline, single level optimal and multilevel optimal.

demonstrate the advantages of using parametric effectiveness in a multilevel optimization.

The refinement procedure proposed in [2] is an improved refinement strategy that makes use of a classical evolutionary algorithm, called Differential Evolution [133, 134], in order to maximise the parametric effectiveness. Therefore, the increase in parametric effectiveness does not only come from the introduction of new control points but also from the optimization of the location of the control points.

The multilevel optimizations performed in [2] used the simplest steepest descent algorithm [135] combined with: (1) a line search to determine the appropriate step size [136]; (2) the projected gradient method [137] to handle the aerodynamic constraints . The latter allows to walk in the direction perpendicular to the constraints whilst reducing the objective.

Figure 7.8 shows the evolution of the objective (i.e. entropy generation) and constraint (i.e. exit flow angle) during the multilevel optimization. Before starting the optimization, the coarse CAD-based parameterization has a parametric effectiveness (PE) of 70%. By performing the first level optimization using this parameterization the optimizer is able to reduce the objective by approximately 10%. At the end of the first level optimization, the PE of the new design is equal to 9%, which means that the CAD-based parameterization does no longer allow to efficiently move the boundary in the direction dictated by the adjoint solution. Therefore, the CAD-based parameterization needs to be refined in order to start the optimization from a second level. Three control points are inserted in order to increase the PE. Using the Differential Evolution algorithm to optimize the position of the control points, the PE is increased from 9% to 45%. The optimized CAD-based parameterization is shown in the left part of Figure 7.9. Alternatively, the PE is increased from 9%



(a) Baseline (b) Single level optimal (c) Multilevel optimal

Figure 7.7.: Entropy generation contour at the trailing edge for the baseline, single level optimal and multilevel optimal.

to 20% if the knot positions are chosen randomly. The right part of Figure 7.9 shows the refined parameterization where the position of the knots has been arbritarily chosen. Figure 7.10 compares the boundary movement dictated by the adjoint solution with the boundary movement that can be achieved by the refined CAD-based parameterizations. The graph shows that the parameterization with $PE = 45\%$ is able to match better the boundary change dictated by the adjoint solution than the parameterization with $PE = 20\%$. In order to demonstrate that a higher PE allows to find a better final shape at the end of the optimization, a second level optimization is performed; the first one using the optimal CAD-based parameterization ($PE = 45\%$), and the second one using the same number of parameters but characterizaed by a lower PE ($PE = 20\%$). The results shown in 7.8 for the second level of the multilevel optimization show that using the parameterization characterized by a higher PE gives rise to a larger reduction in the entropy generation than the parameterization characterized by the lower PE. This demonstrates once more the benefits of using the parametric effectiveness in shape refinement for efficient multilevel CAD-based optimization.

7.4. Conclusions

In this chapter, an efficient CAD based optimization strategy is presented in which the optimizer explores richer design spaces by autonomously creating the best parameterization to use at each time. For this purpose, the parametric effectiveness is computed to rate the ability of a set of CAD design parameters to produce the design shape change dictated by the adjoint sensitivities. Exact surface sensitivities are obtained by applying algorithmic differentiation to the CAD kernel and grid generator. The parametric effectiveness was then used for the progressive exploration of higher-dimensional design spaces in order to make autonomous decisions and creates the best parameterization to be used during the refinement of the design space. It was shown that the multilevel optimization was more efficient and was able to reach a better optimum than the single level optimization, even though they used the same 273 design variables at the end of the multilevel

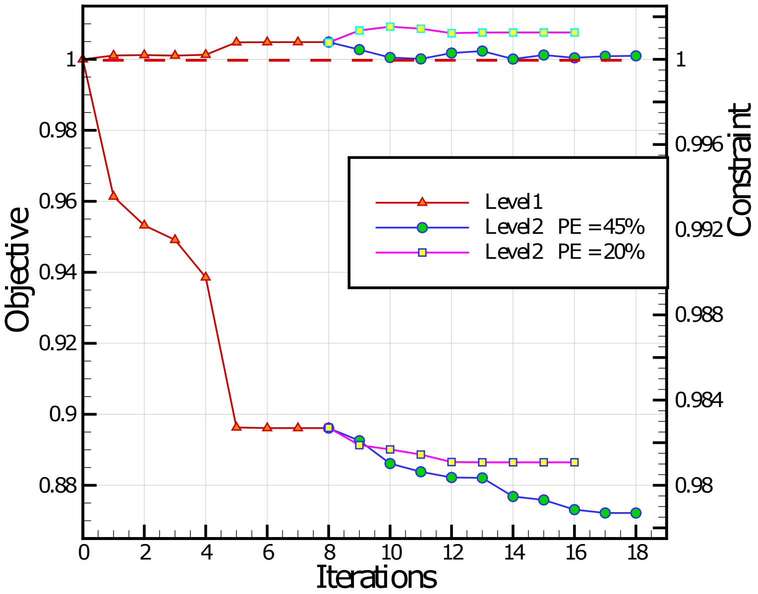


Figure 7.8.: Multilevel optimization results by inserting 3 control points on the second level. The parametric effectiveness (PE) of 45% and 20% correspond to the optimized and non-optimized CAD-based parameterizations respectively [2].

optimization. Also, two alternative approaches are presented to refine the CAD-based parameterization. In the first one, the parametric effectiveness is increased by inserting control points to the pressure and suction side B-spline curves. In the second one, the parametric effectiveness is increased by the insertion of the control points and the optimization of its position by using the Differential Evolution algorithm. The second refinement strategy allows to increase the parametric effectiveness to higher values than the first refinement strategy. The optimizer is able to further reduce the objective when using both refinement strategies, but larger reductions are obtained with the second refinement strategy. Though the method presented in this chapter was applied mainly to gradient-based optimization strategies, the benefits of using the parametric effectiveness in combination with the hybrid methods, which combine gradient-based and gradient-free techniques, will be an area of future research.

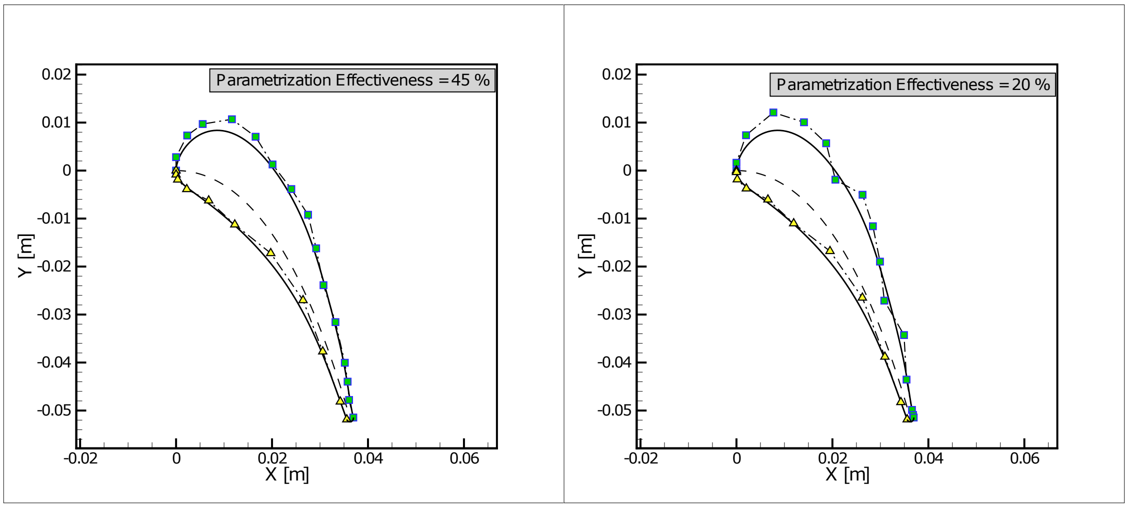


Figure 7.9.: Insertion of 3 knots on the suction side and 3 knots on the pressure side. Comparison between the optimized parametrization (left) and a random parametrization (right) [2].

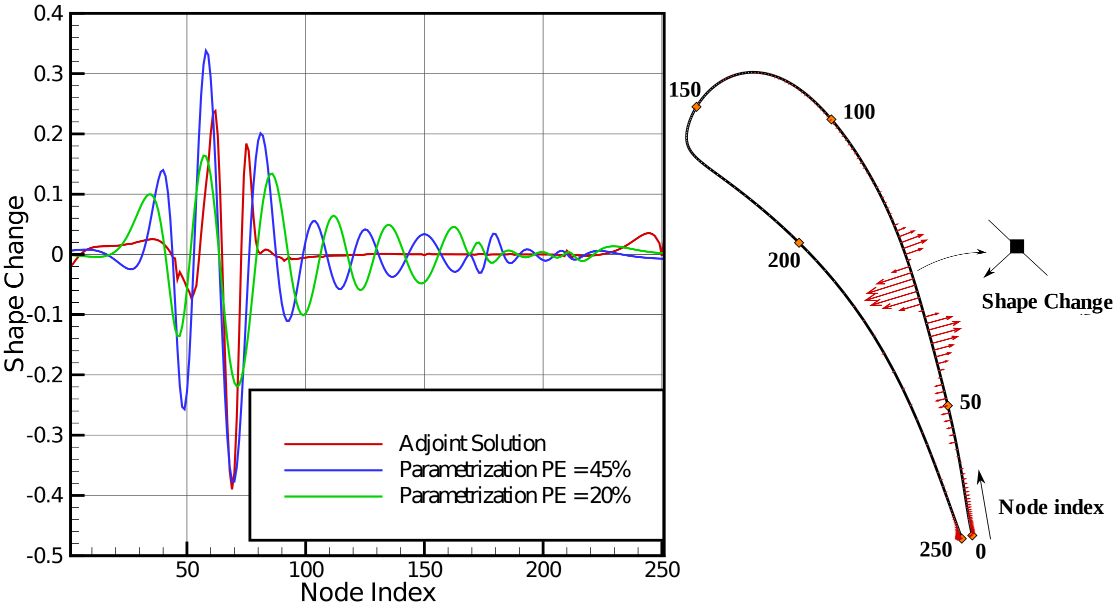


Figure 7.10.: Comparison between the ideal shape change (Adjoint Solution) and the best possible shape movements feasible with the 2 parametrizations (left), adjoint surface sensitivities (right) [2].

Chapter 8.

Conclusions

Although adjoint-based optimization methods have numerous advantages, specially the efficient computation of the gradient virtually independent of the number of design variables, the methods are not yet picked up largely by industry. One major bottleneck herein is that adjoint methods mainly work on deforming the CFD grid and as such loose the connection to CAD, the industry adopted standard for the design of components. In this thesis, a methodology is presented to integrate the CAD kernel and grid generation tools within a CFD adjoint-based optimization framework by expressing the optimization problem through CAD parameters, rather than working directly on the CFD grid. The advantage of including the CAD model in the design system is that higher level constraints can be imposed on the shape, allowing the optimized model or component to be manufactured. However, this requires an additional sensitivity computation step: the computation of the grid sensitivities. An alternative approach to finite differences (FD) or the design velocity approach is proposed in chapter 3, which consists in differentiating the CAD kernel and grid generator with algorithmic differentiation (AD). The computation of gradients for the geometry creation or the fluid mesh creation of turbine cascades is generally faster using the trace reverse mode of AD than FD, specially when using a large number of directions, but it requires more memory. The comparison between the sensitivities obtained by AD and FD shows for the LS89 turbine cascade that the error introduced by FD is very sensitive to the chosen step size and its magnitude can be relatively close to the sensitivity value itself if the step size is not chosen appropriately. The use of AD allows computing the grid sensitivities to machine accuracy and avoids the limited arithmetic precision, the truncation, and some robustness issues that can arise with FD. Also, the AD surface grid sensitivities were compared to the design velocities, showing that there are large differences between them. The largest root mean square average error between the design velocities and the AD surface grid sensitivities is of the order of 49.5%. However, the design velocities can match reasonably well the normal component of the AD surface grid sensitivities if the step size is appropriately chosen (2.9% of error). Therefore, the differentiation of these CAD kernel and grid generation using algorithmic differentiation is preferred to finite differences or the design velocity approach when the CAD kernel and the grid generator primal source code is available to the designer. If the geometry and mesh creation is performed by external software and the designer has no access to the primal sources, the design velocity approach is still more robust than finite differences and it can approximate relatively well the shape sensitivities in the normal direction to the model boundary.

The integration of the CAD kernel and grid generator within a CFD-adjoint-based optimization framework is demonstrated in chapters 4 and 5, where the aerodynamic performance of the LS89 turbine cascade is significantly improved at design and off-design conditions. A single point optimization of the LS89 axial turbine cascade vane profile for the design downstream isentropic Mach number of 0.9 is shown in chapter 4. By keeping the CAD representation in the optimization loop, it is not necessary to convert the optimal grid back to a smooth CAD shape, and it is possible to impose manufacturing constraints on the shape, such as axial chord length, trailing edge radius and second order derivative continuity at the leading edge. Additionally, the adjoint sensitivities are filtered out and only smooth shapes are produced. The optimization results show that the total pressure losses and entropy generation can be reduced by 16% and nearly 12% respectively by going from a front to a rear loaded profile and by keeping the exit flow angle fixed. Despite the increase in mass flow, loading, boundary layer thickness displacement and momentum thickness, the base pressure coefficient plays a dominant role and reduces the profile losses by 30%. In chapter 5, a multipoint optimization of the LS89 axial turbine vane profile for a downstream isentropic Mach number of 0.9 (design point), 0.955 and 1.01 is presented. The largest aerodynamic improvements were achieved at the transonic operating point where the total pressure losses were reduced by 14% whilst keeping the exit flow angle fixed. A comparison between the single point and multipoint optimal designs show that the blade with lower rear suction side turning (i.e. multipoint optimal profile) can generate lower losses at transonic operating conditions than a blade with higher rear suction side turning (i.e. single point optimal profile). However, the latter can generate lower losses if it operates at subsonic downstream isentropic Mach below to 0.935 approximately.

CAD based approaches have the advantage to represent the optimal shape in the industry accepted format for manufacturing. However, the success of a CAD-based optimization framework highly depends on the expertise and skill of the designer, and often it is not obvious which parameters should be chosen for the optimization to move the model boundary in the direction indicated by the adjoint sensitivities. Furthermore, the choice of the CAD parameters constraints how the model boundary can change, and a poor choice may prevent the exploration of innovative geometries. For this purpose, a new methodology is presented in chapter 6 to rate the quality of the CAD-based parameterization for optimization. The parametric effectiveness rates a group of CAD-based design variables in terms of its ability to produce a shape change that is aligned with the adjoint sensitivities. This measure takes into account the non-orthogonality of the design variables and uses accurate geometric sensitivities provided by algorithmic differentiation. The methodology is extended to constrained optimization problems where the constraints can be of any type: geometrical, aerodynamic, structural, etc. The method is tested to rate different CAD-based parameterizations of the LS89 turbine cascade and to select four of them for optimization at design point. Two types of constrained optimizations were considered and seven aerodynamically constrained optimizations were performed. The optimization results showed that the largest reductions in entropy were achieved when using those groups of CAD-based design variables that exhibited the highest parametric effectiveness at the start of the optimization. Therefore, the parametric effectiveness is a good metric to be used to choose the best CAD-based parameterization for optimization. Additionally, the parametric effectiveness can also be used to rate the individual effectiveness of each CAD parameter, in order to exclude the least performing ones from the optimization. Considerable time savings can be achieved by using the most effective subset of parameters for optimization, with a reasonable smaller number of design parameters, at the expenses of a smaller reduction in the gain in performance. In chapter 6 the parametric effectiveness was only evaluated at the start of the optimization process to decide which group of CAD parameters was more fit for optimization. However, it is possible that another group of design variables becomes more effective than the chosen one in later stages of the optimization.

Another downside of CAD-based approaches is that they restrict the design space compared to node based approaches where each single mesh point is free to move. In order to preserve the rich design space of the node based approach, an efficient CAD based optimization strategy is presented in chapter 7 in which the optimizer explores richer design spaces by autonomously creating the best parameterization to use at each time. The novelty of this strategy lies in the definition of the parametric effectiveness and using it to refine the CAD model parameterization in a multilevel optimization process. In a multilevel optimization where the optimization starts with a small number of design variables and ends with a large number, it is unclear how many new design variables should be introduced at every new design space, or how good is every new parameterization to produce the shape changes that are required by the adjoint sensitivities. In this context, the parametric effectiveness can be used to create the appropiate parameterization to use in each new dimensional design space and make the optimization a more autonomous, robust and efficient design strategy. It was shown that the multilevel optimization was more efficient and was able to reach a better optimum than the single level optimization, even though they used the same 273 design variables at the end of the multilevel optimization. Also, two alternative approaches were presented to refine the CAD-based parameterization. In the first one, the parametric effectiveness is increased by inserting control points to the pressure and suction side B-spline curves. In the second one, the parametric effectiveness is increased by the insertion of the control points and the optimization of its position by using the Differential Evolution algorithm. The second refinement strategy allows to increase the parametric effectiveness to higher values than the first refinement strategy. The optimizer is able to further reduce the objective when using both refinement strategies, but larger reductions are obtained with the second refinement strategy. Though the method presented was applied mainly to gradient-based optimization strategies, the benefits of using the parametric effectiveness in combination with the hybrid methods, which combine gradient-based and gradient-free techniques, will be an area of future research.

Chapter 9.

Future work

In this thesis, it is shown that the use of the reverse mode of algorithmic differentiation for the computation of gradients is more accurate and more efficient than using finite differences, specially for a large number of directions. However, using the reverse mode requires much more memory than the primal and this requirement could become a bottleneck for larger test cases. When the size of the tape cannot fit in the RAM of the working station, the tape needs to be written into the hard disk and this can slow down significantly the computation of the gradients. To avoid these problems, future efforts should focus in differentiating the mesh sources using an AD technique called check pointing, which allows to reduce the memory requirements at the expenses of an increase of the computational time. By using check pointing, it would also be possible to make sure that the tape data fits in the RAM and hence avoid writing out the tape into the hard disk, making the computation of the gradients more efficient. The structured meshes are used in this thesis to provide fine control over the final quality of the mesh. Nevertheless, the work presented in this thesis could be equally applied to unstructured grids. This would require the differentiation of the unstructured grid generation.

The optimization results presented in chapters 4 and 5 show that large aerodynamic improvements can be achieved for the LS89 turbine cascade at design point and off-design respectively by reducing the downstream mixing losses and the mixed out loss of the boundary layers. However, the question arises whether using RANS to predict the flow field around the trailing edge of a transonic turbine is a valid approach, since the base pressure has a relatively big influence on the profile losses. Future unsteady simulations and/or experimental test are going to be necessary to support or reject the aerodynamic improvements for the single point and multipoint optimal designs.

The parametric effectiveness was used in chapter 6 to rate the ability of groups of CAD parameters to move the shape boundary according to what the adjoint solution dictates. This allows to find the most effective parameters and also to select the best CAD-based parameterization for optimization. A high parametric effectiveness at the beginning of the optimization resulted in larger reductions in the objective. Also, it was shown that using the most effective subset of parameters instead of the full set allowed the optimizer to converge faster. Future research should explore the benefits of using the most effective subset of parameters instead of the full set in gradient-free techniques or hybrid methods, where the cost savings are expected to be quite substantial.

In chapter 7, the parametric effectiveness was used to refine the CAD model parameterization in a multilevel optimization process. The benefits of using the parametric effectiveness in combination with the hybrid methods, which combine gradient-based and gradient-free techniques, in a multilevel shape refinement will be an area of future research. Also, two alternative procedures were presented to refine the CAD-based parameterization. The first one consists in introducing new control points by assuming the positions of the knots to insert. The second strategy consists in introducing control points and then optimizing its positions by finding the optimal positions of the knots to insert. The second refinement strategy gave rise to a higher parametric effectiveness and larger reductions in the objective. However, the use of Differential Evolution is limited to the use of a small number of design variables, as the computational effort to optimize the position of the control points can become prohibitive if a large number of variables are considered. For this reason, future efforts should investigate the replacement of the Differential Evolution algorithm by an efficient gradient-based algorithm, specially if the objective space does not present multiple minima. However, if the parametric effectiveness objective space presents multiple minima, an efficient hybrid method should be used instead. In both cases, this would require computing the gradients of the parametric effectiveness w.r.t. the position of the inserted knots, which can be done by making use of algorithmic differentiation. In addition, the use of parametric effectiveness in a multilevel optimization was exploited at design point. Future research should investigate the use of parametric effectiveness to create the best CAD model parameterization in a multipoint optimization.

This thesis includes many ideas and pieces of works that should be evaluated in a wide range of problems, extending this work to geometries other than the LS89. Though the thesis exclusively focused in turbomachinery, it is a matter of a different implementation of the main ideas presented in order to generalize the proposed methods in other domains of applications, such as external aerodynamics.

Finally, the proposed methods-tools could be plugged into any other adjoint based optimization method, in order to make better AD tools and more efficient adjoint methods.

Appendix A.

Main publications in the context of this thesis

International Journal papers:

- Ismael Sanchez Torreguitart, Tom Verstraete, and Lasse Mueller. Optimization of the ls89 axial turbine profile using a cad and adjoint based approach. *International Journal of Turbomachinery Propulsion and Power, 3(3):20, 2018*, DOI: 10.3390/ijtpp3030020.
- Ismael Sanchez Torreguitart, Tom Verstraete, Dheeraj Agarwal, Trevor R. Robinson, and Cecil G. Armstrong. Efficient cad-based adjoint optimization using parametric effectiveness for multilevel shape refinement. 2018, *submitted to an international journal.*

Chapters in books

- Ismael Sanchez Torreguitart, Tom Verstraete, and Lasse Mueller. CAD and Adjoint based Multipoint Optimization of an Axial Turbine Profile. Springer, 2017, DOI: 10.1007/978-3-319-89890-2-3.

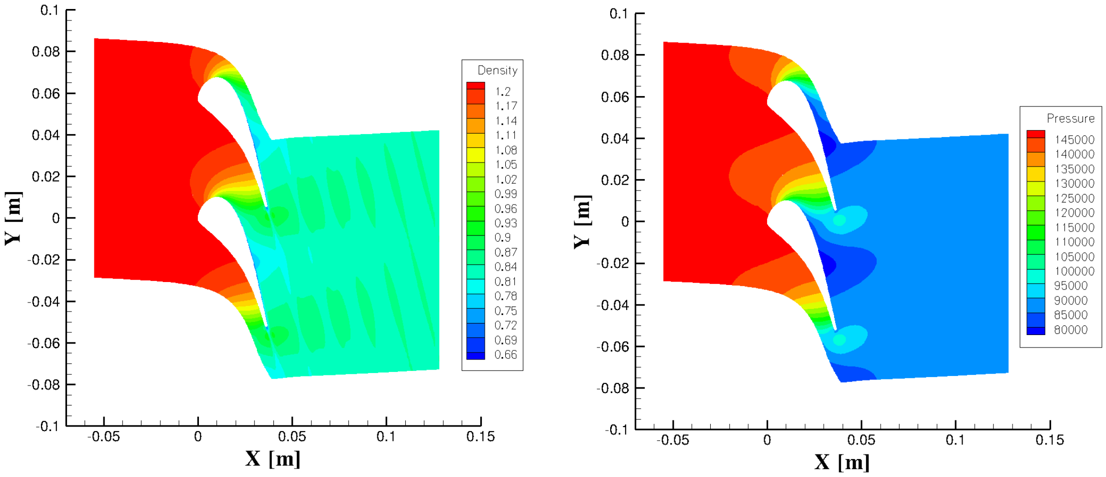
International conference papers:

- Ismael Sanchez Torreguitart, Tom Verstraete, and Lasse Mueller. Cad kernel and grid generation algorithmic differentiation for turbomachinery adjoint optimization. In *7th European Congress on Computational Methods in Applied Sciences and Engineering, Hersonissos, Crete, Greece, June, pages 5-10, 2016*, DOI: 10.7712/100016.2077.6621.
- Ismael Sanchez Torreguitart, Tom Verstraete, and Lasse Mueller. Optimization of the ls89 axial turbine profile using a cad and adjoint based approach. In *12th European Conference on Turbomachinery Fluid dynamics & Thermodynamics.* European Turbomachinery Society, 2017.
- Ismael Sanchez Torreguitart, Tom Verstraete, and Lasse Mueller. Cad and adjoint based multipoint optimization of an axial turbine profile. In *International Conference on Evolutionary and Deterministic Methods for Design Optimization and Control with Applications to Industrial and Societal Problems.* EUROGEN, 2017.

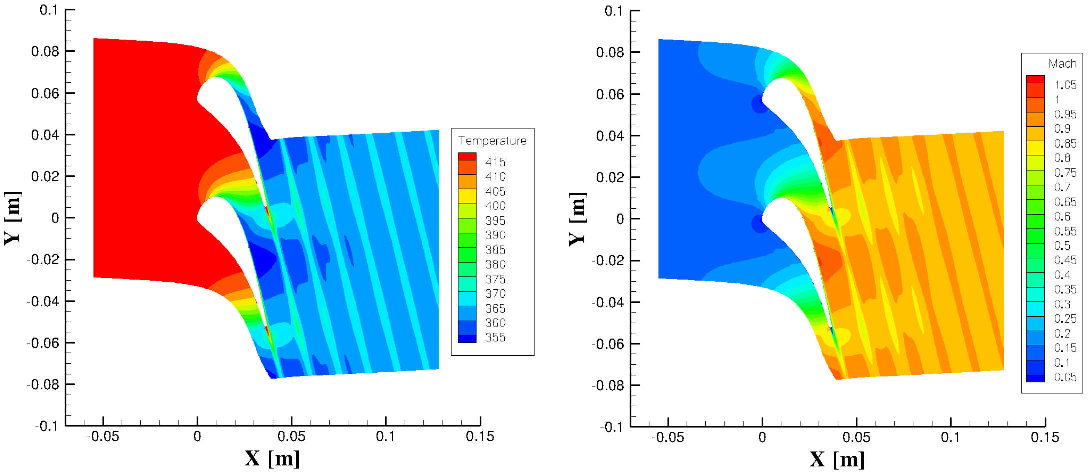
Appendix B.

Summary of the flow results

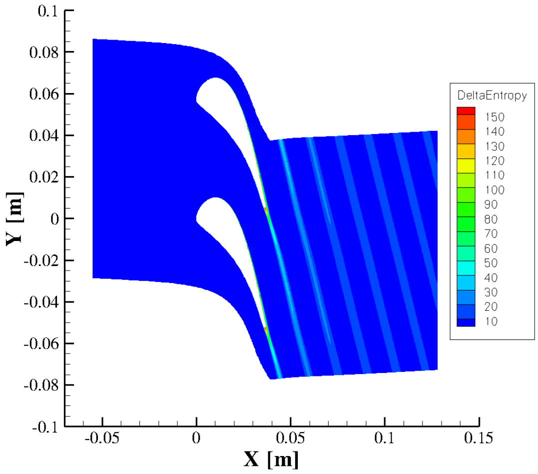
B.1. Baseline shape



(a) Density [kg m^{-3}] (b) Static pressure [Pa]

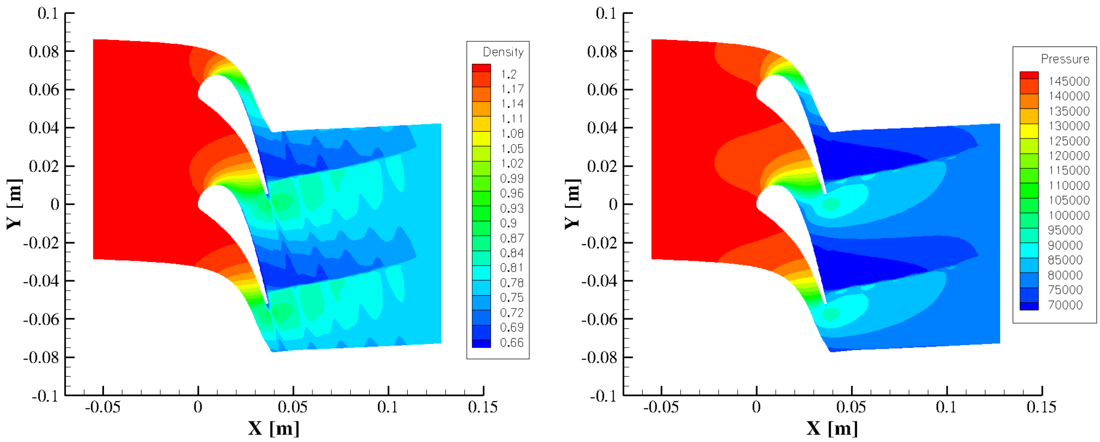


(c) Static temperature [K] (d) Mach [-]



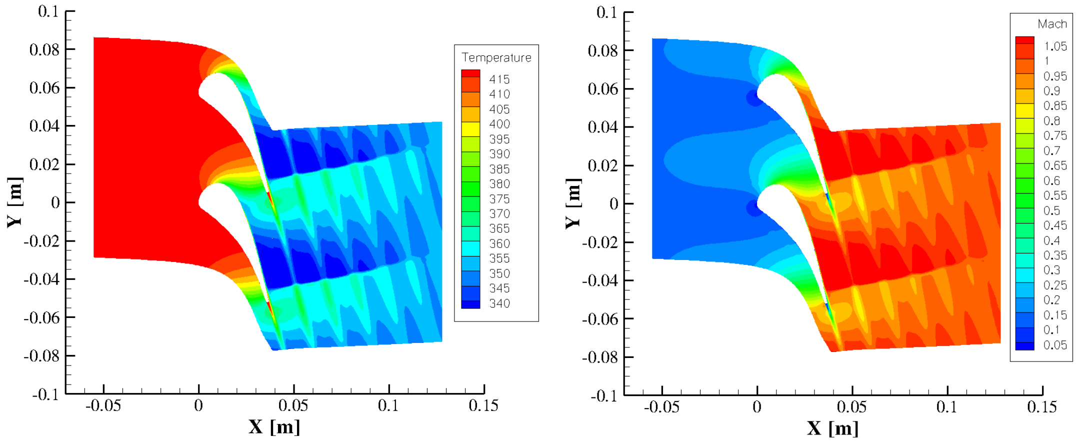
(e) Entropy delta [Pa m^3 kg^{-1}]

Figure B.1.: Flow results of the LS89 baseline shape for an outlet isentropic Mach number $M_{ise,2}$ = 0.90 (SI units)



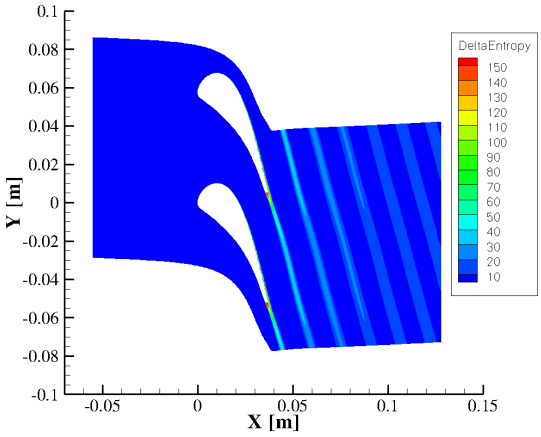
(a) Density [kg m^{-3}]

(b) Static pressure [Pa]



(c) Static temperature [K]

(d) Mach [-]

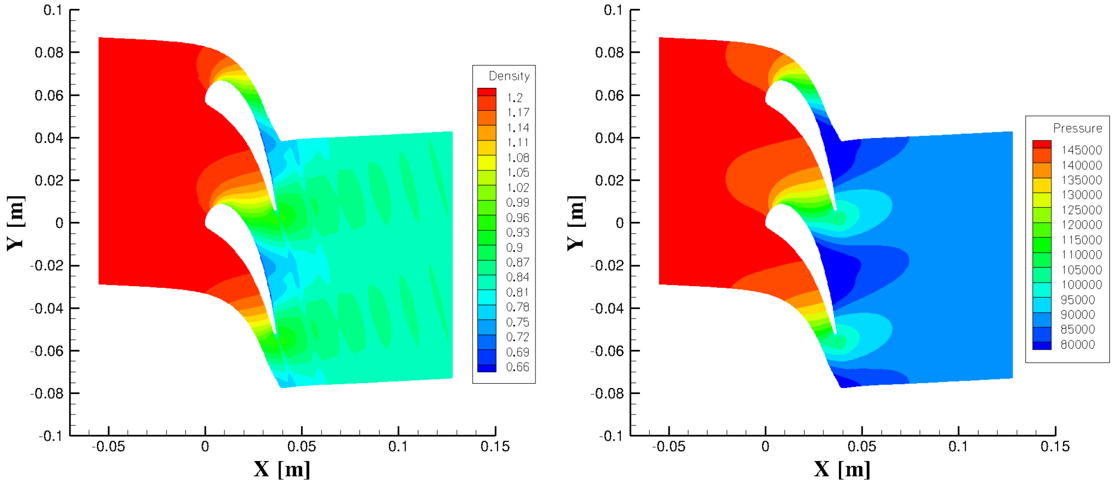


(e) Entropy delta [Pa m^3 kg^{-1}]

Figure B.2.: Flow results of the LS89 baseline shape for an outlet isentropic Mach number $M_{ise,2}$ = 1.01 (SI units)

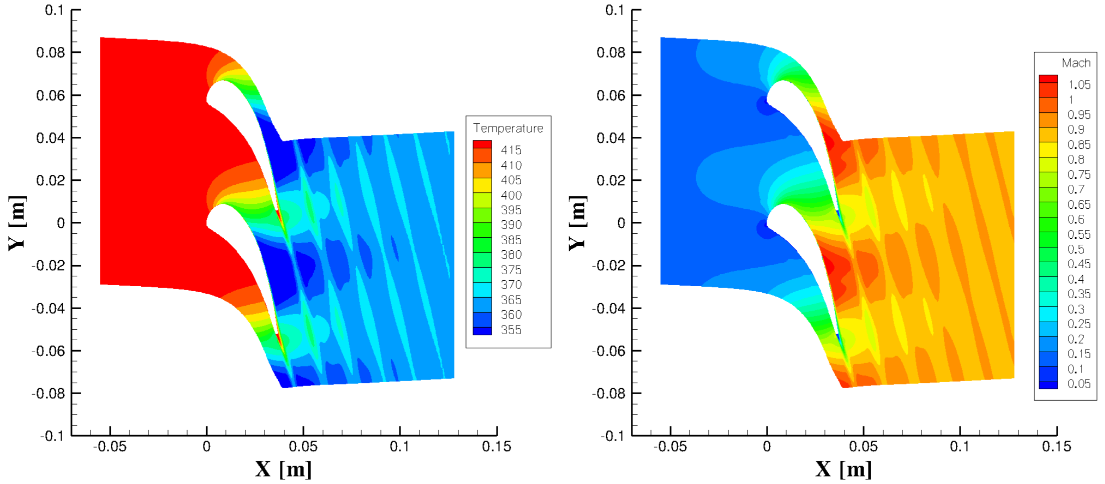
B.2. Optimal shapes with the same exit flow angle as the baseline

B.2.1. Single point optimal shape from chapter 4



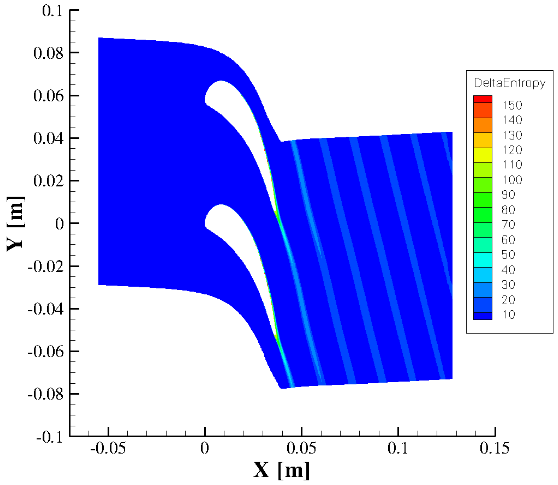
(a) Density [kg m^{-3}]

(b) Static pressure [Pa]



(c) Static temperature [K]

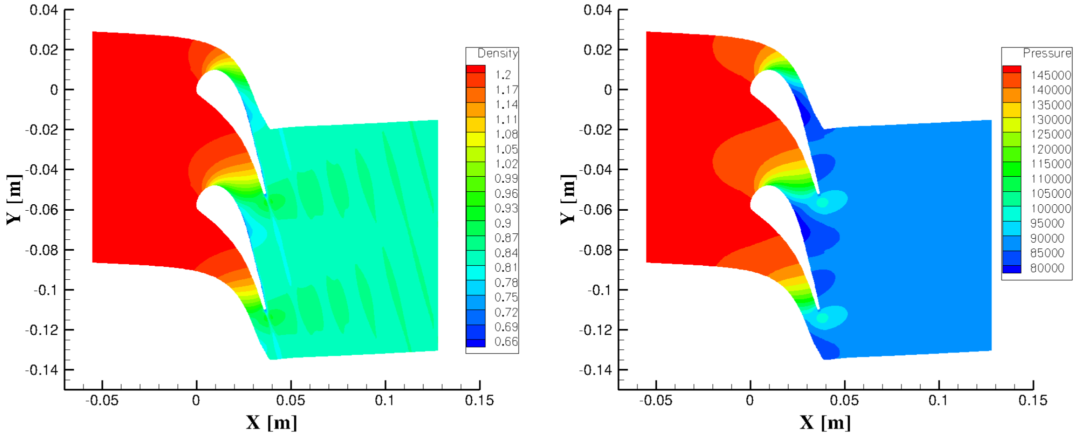
(d) Mach [-]



(e) Entropy delta [Pa m^3 kg^{-1}]

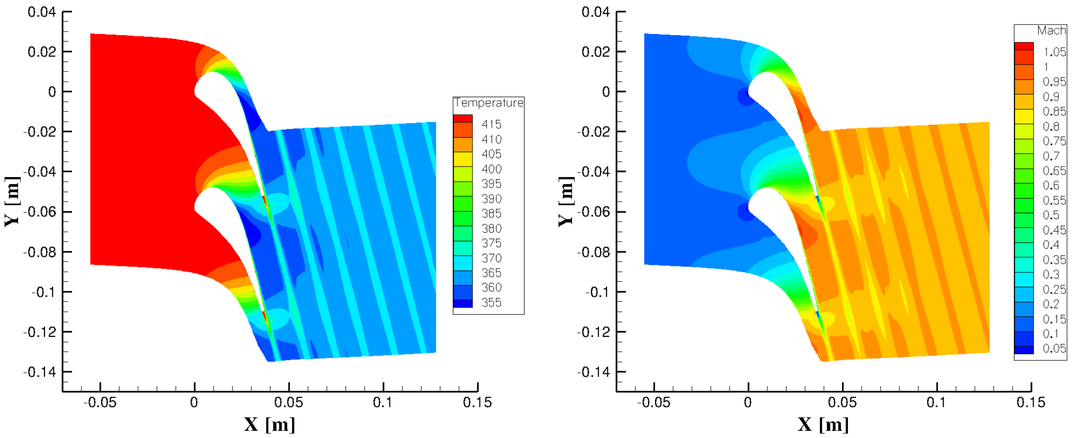
Figure B.3.: Flow results of the single point optimal shape at an outlet isentropic Mach number $M_{ise,2} = 0.90$

B.2.2. Multipoint optimal shape from chapter 5



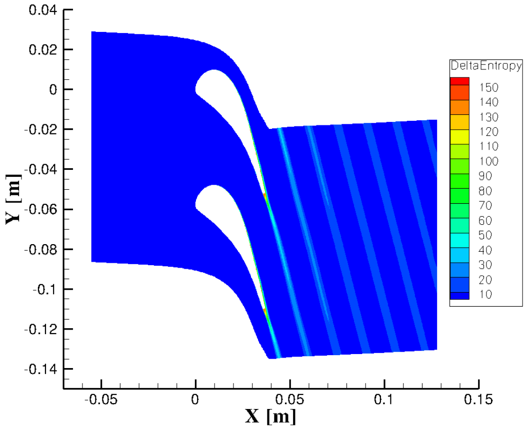
(a) Density [kg m^{-3}]

(b) Static pressure [Pa]



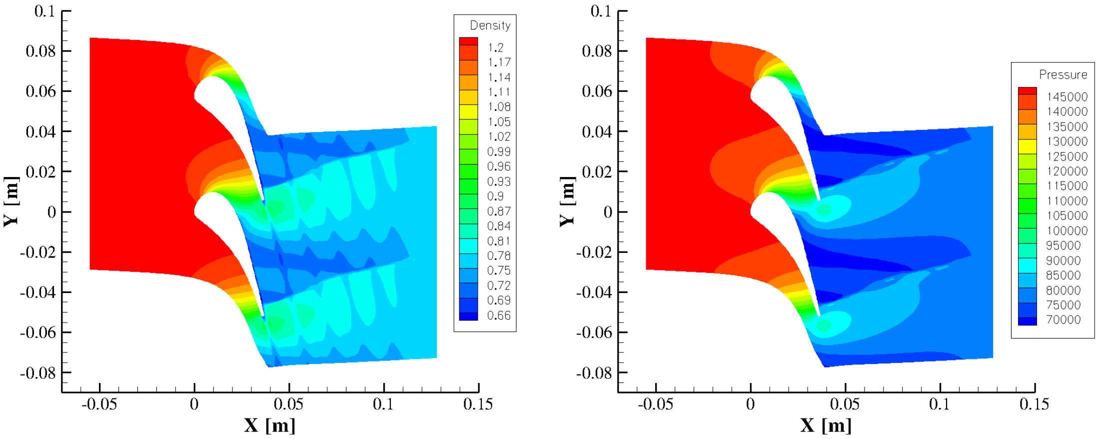
(c) Static temperature [K]

(d) Mach [-]



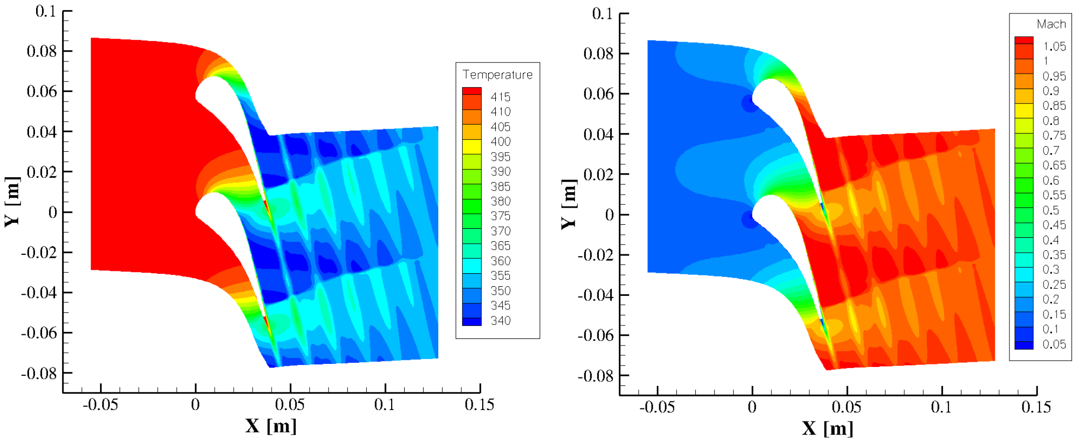
(e) Entropy delta [Pa m^3 kg^{-1}]

Figure B.4.: Flow results of the multipoint optimal shape for an outlet isentropic Mach number $M_{ise,2} = 0.90$ (SI units)



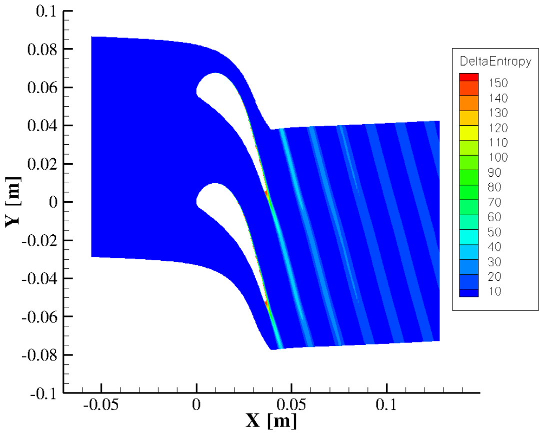
(a) Density [kg m^{-3}]

(b) Static pressure [Pa]



(c) Static temperature [K]

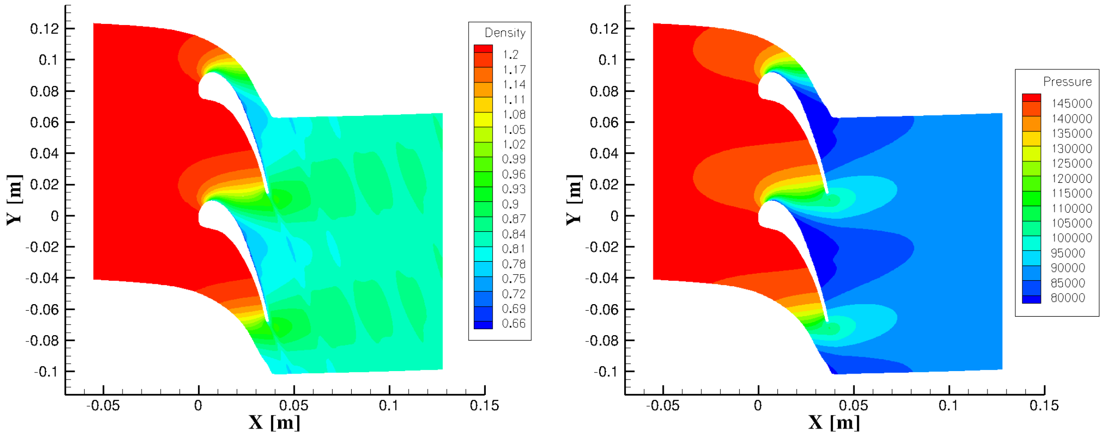
(d) Mach [-]



(e) Entropy delta [Pa m^3 kg^{-1}]

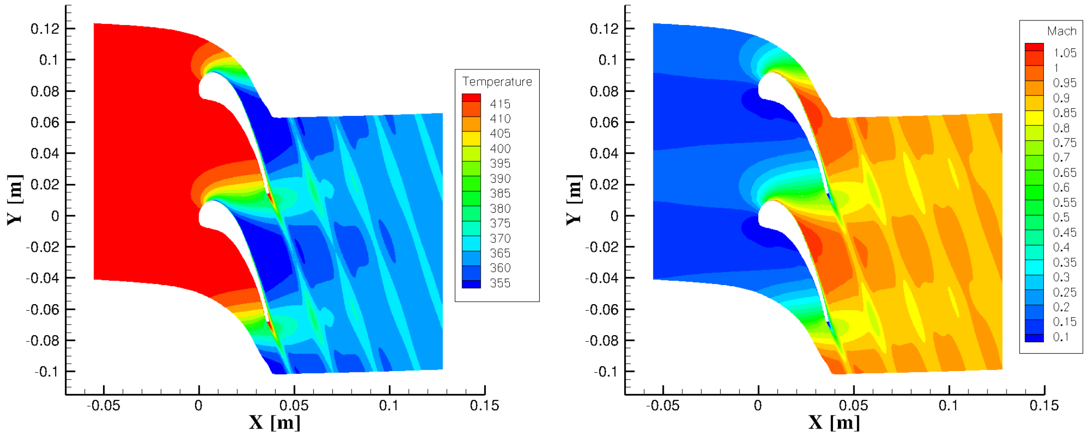
Figure B.5.: Flow results of the multipoint optimal shape for an outlet isentropic Mach number $M_{ise,2} = 1.01$ (SI units)

B.2.3. Single point optimal shape from chapter 6 using set SS9PS9



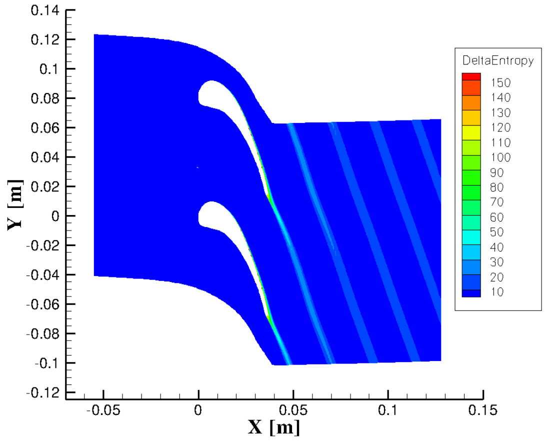
(a) Density [kg m^{-3}]

(b) Static pressure [Pa]



(c) Static temperature [K]

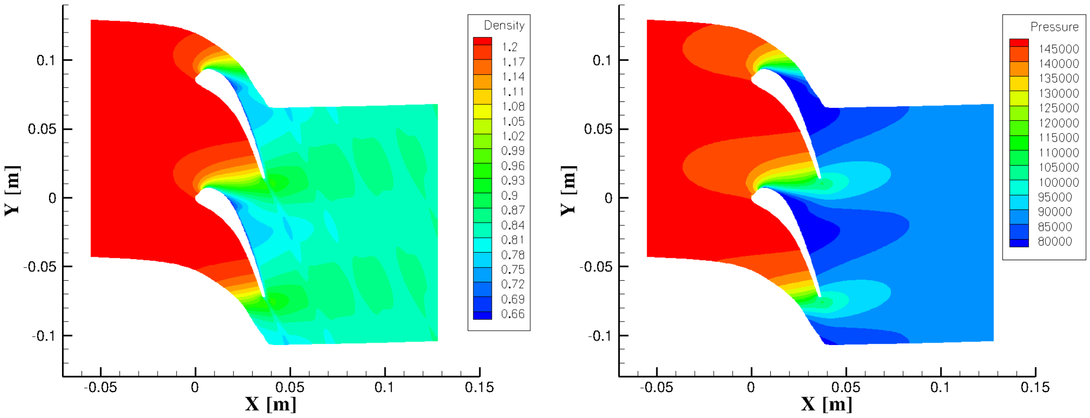
(d) Mach [-]



(e) Entropy delta [Pa m^3 kg^{-1}]

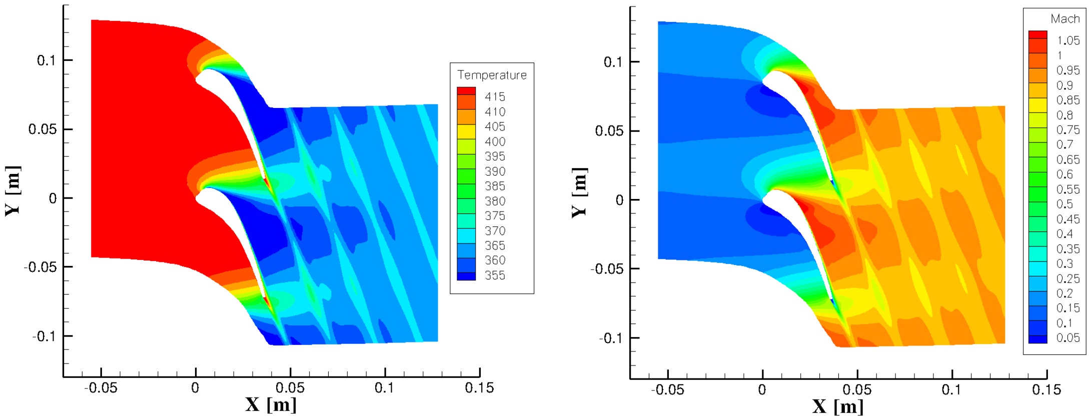
Figure B.6.: Flow results of the single point optimal shape using set SS9PS9 for an outlet isentropic Mach number $M_{ise,2} = 0.90$ (SI units)

B.2.4. Single point optimal shape from chapter 6 using set SS14PS4



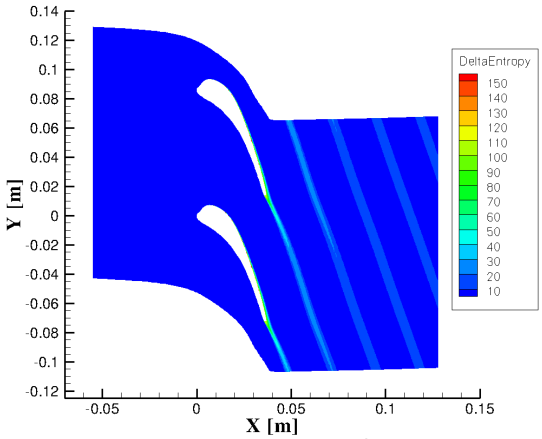
(a) Density [kg m^{-3}]

(b) Static pressure [Pa]



(c) Static temperature [K]

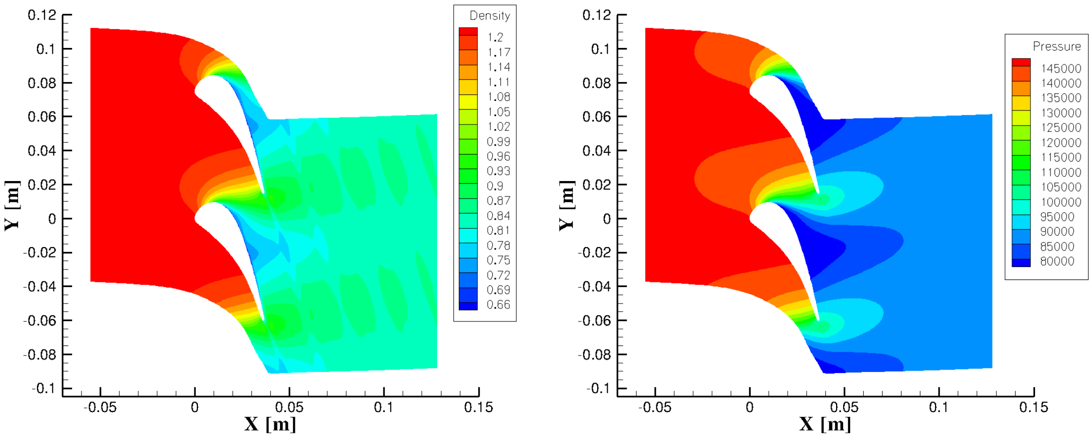
(d) Mach [-]



(e) Entropy delta [Pa m^3 kg^{-1}]

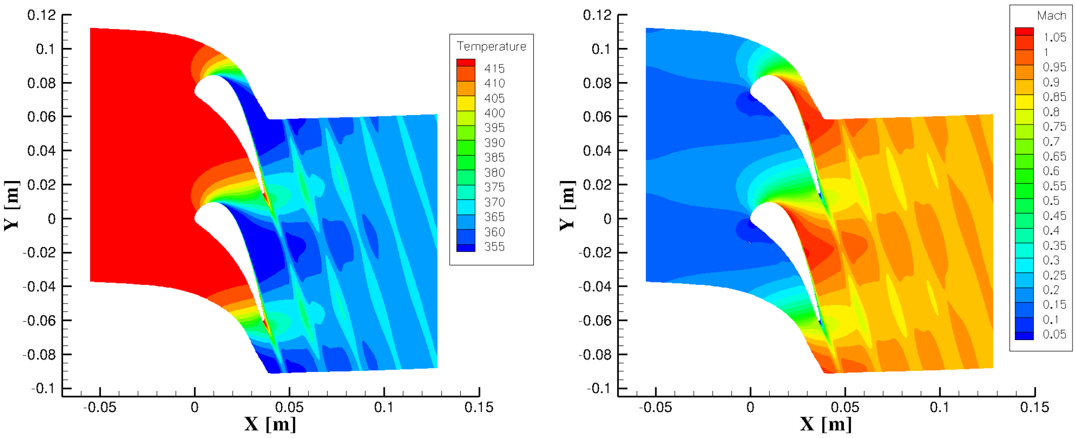
Figure B.7.: Flow results of the single point optimal shape using set SS14PS4 for an outlet isentropic Mach number $M_{ise,2} = 0.90$ (SI units)

B.2.5. Single point optimal shape from chapter 6 using subset 1



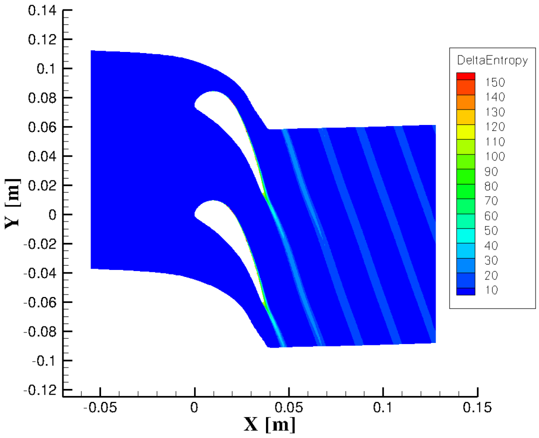
(a) Density [kg m^{-3}]

(b) Static pressure [Pa]



(c) Static temperature [K]

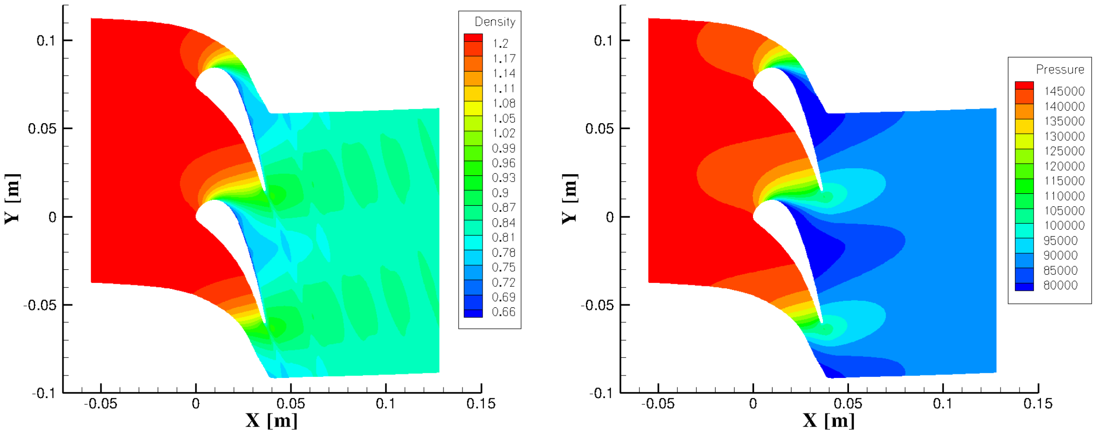
(d) Mach [-]



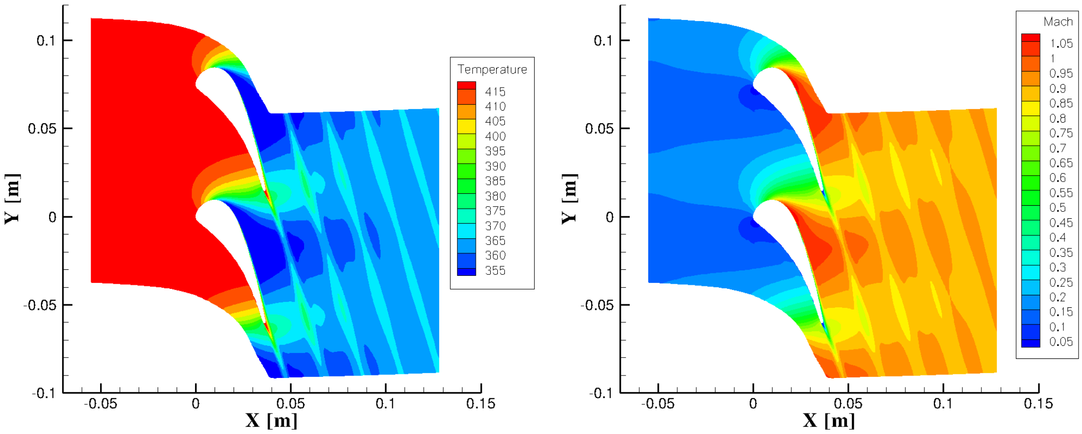
(e) Entropy delta [Pa m^3 kg^{-1}]

Figure B.8.: Flow results of the single point optimal shape using subset 1 for an outlet isentropic Mach number $M_{ise,2} = 0.90$ (SI units)

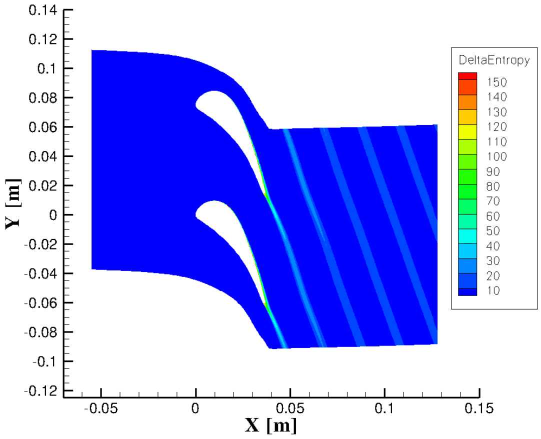
B.2.6. Single point optimal shape from chapter 6 using subset 3



(a) Density [kg m^{-3}] (b) Static pressure [Pa]



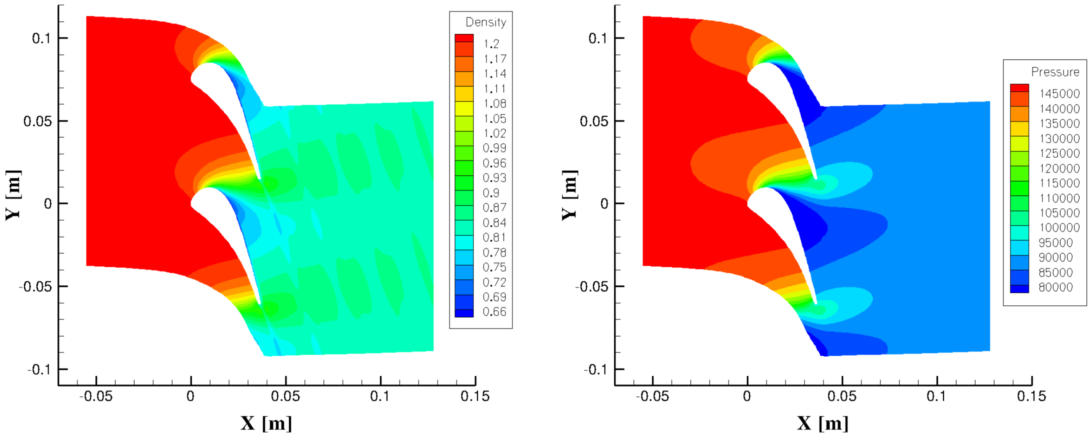
(c) Static temperature [K] (d) Mach [-]



(e) Entropy delta [Pa m^3 kg^{-1}]

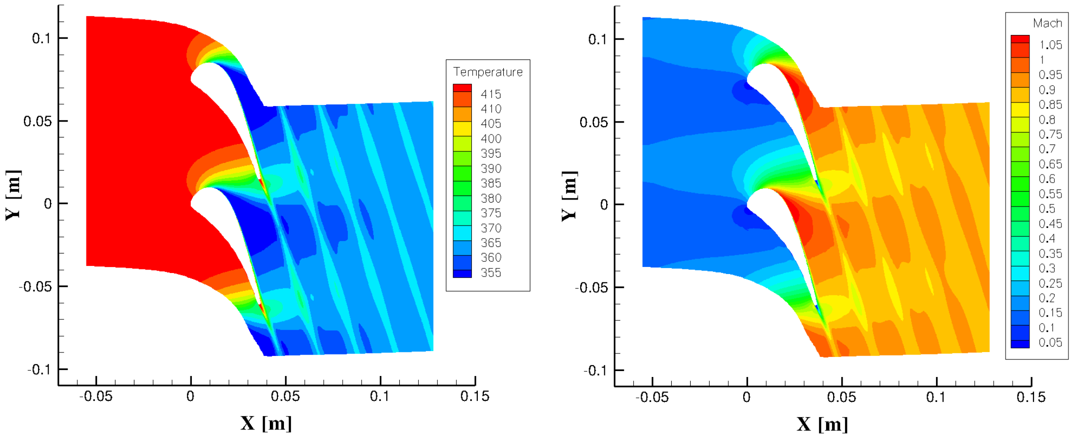
Figure B.9.: Flow results of the single point optimal shape using subset 3 for an outlet isentropic Mach number $M_{ise,2} = 0.90$ (SI units)

B.2.7. Single level and single point optimal shape from chapter 7



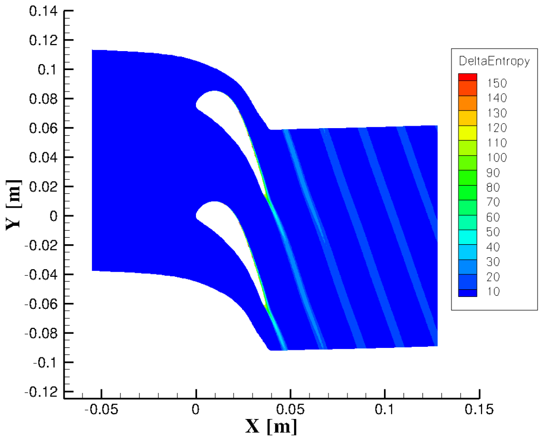
(a) Density [kg m^{-3}]

(b) Static pressure [Pa]



(c) Static temperature [K]

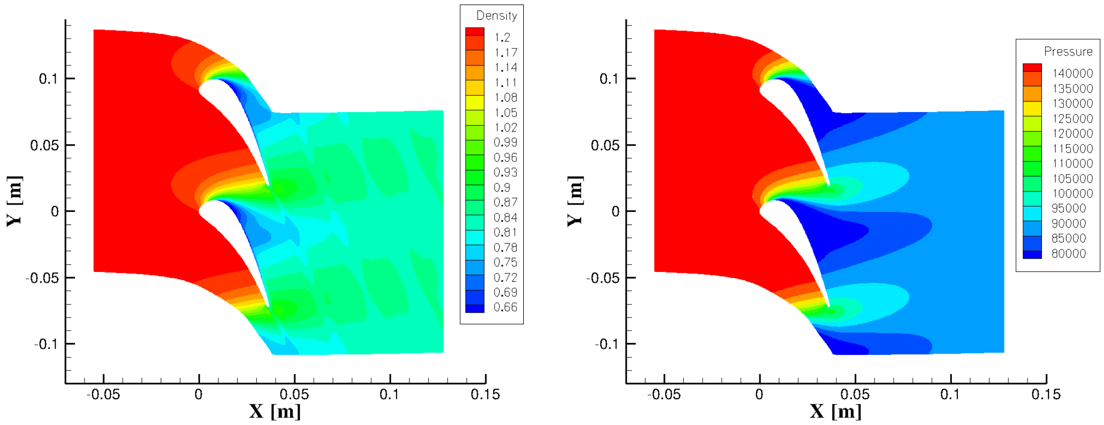
(d) Mach [-]



(e) Entropy delta [Pa m^3 kg^{-1}]

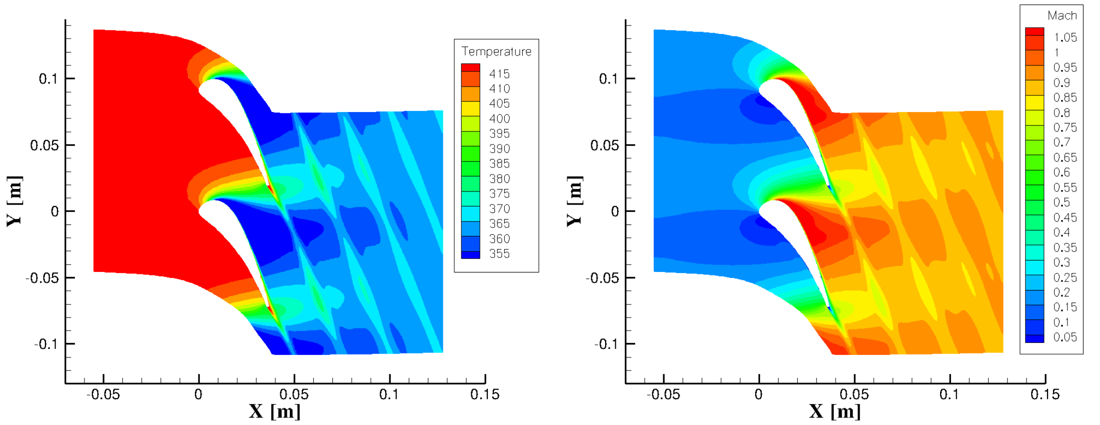
Figure B.10.: Flow results of the single level and single point optimal shape for an outlet isentropic Mach number $M_{ise,2}$ = 0.90 (SI units)

B.2.8. Multilevel and single point optimal shape from chapter 7



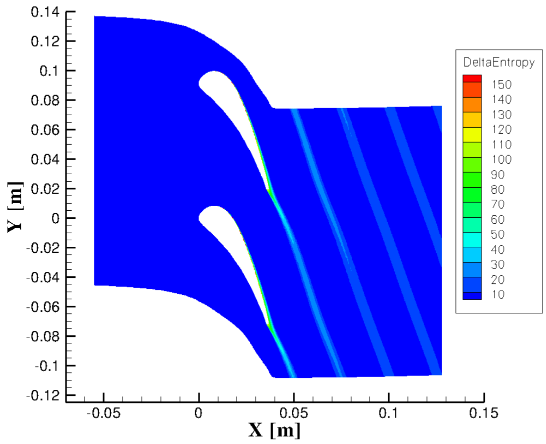
(a) Density [kg m^{-3}]

(b) Static pressure [Pa]



(c) Static temperature [K]

(d) Mach [-]

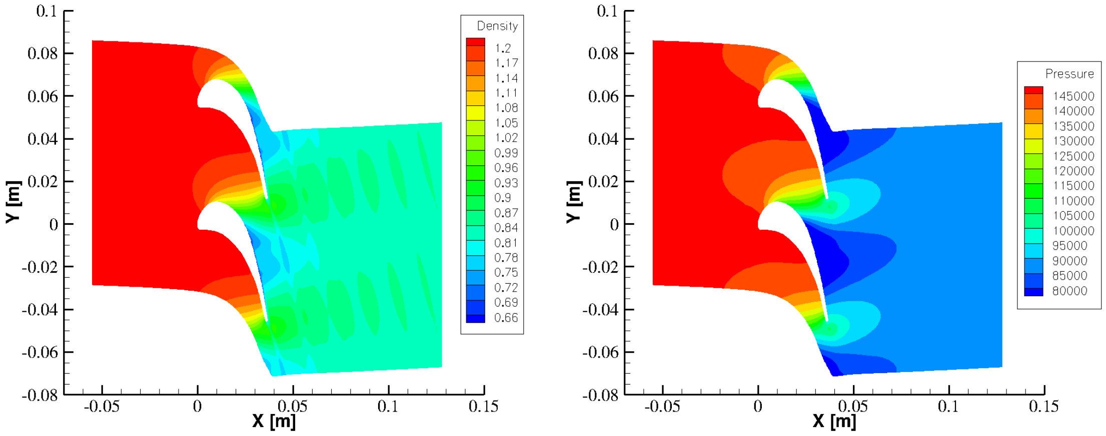


(e) Entropy delta [Pa m^3 kg^{-1}]

Figure B.11.: Flow results of the multilevel and single point optimal shape for an outlet isentropic Mach number $M_{ise,2}$ = 0.90 (SI units)

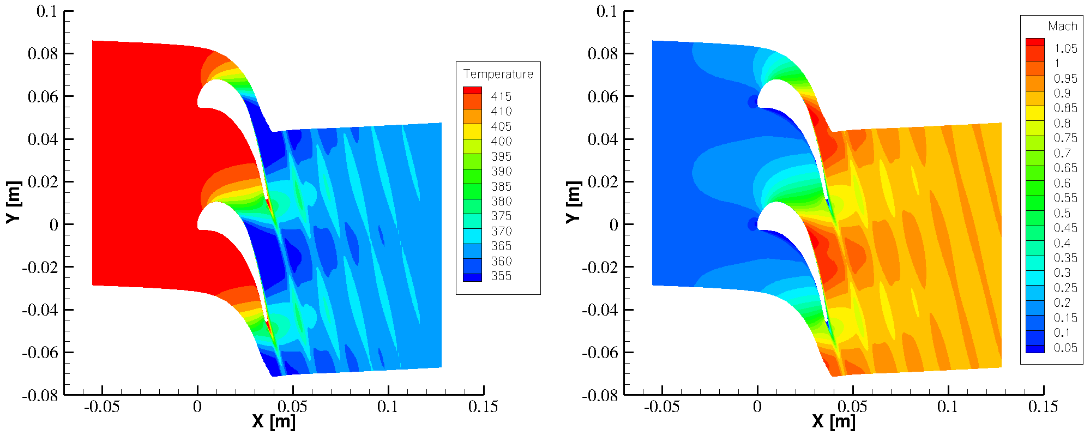
B.3. Optimal shapes with the same exit flow angle and mass flow as the baseline

B.3.1. Single point optimal shape from chapter 6 using set SS9PS9



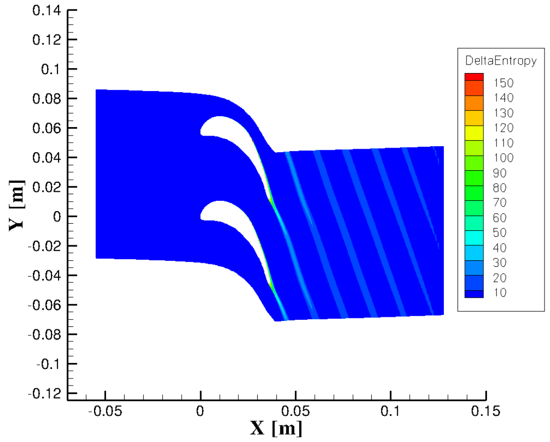
(a) Density [kg m^{-3}]

(b) Static pressure [Pa]



(c) Static temperature [K]

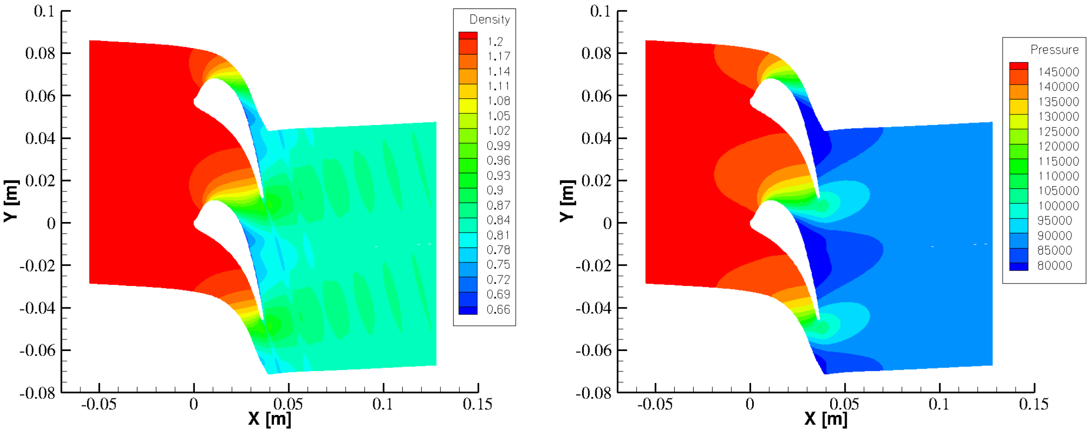
(d) Mach [-]



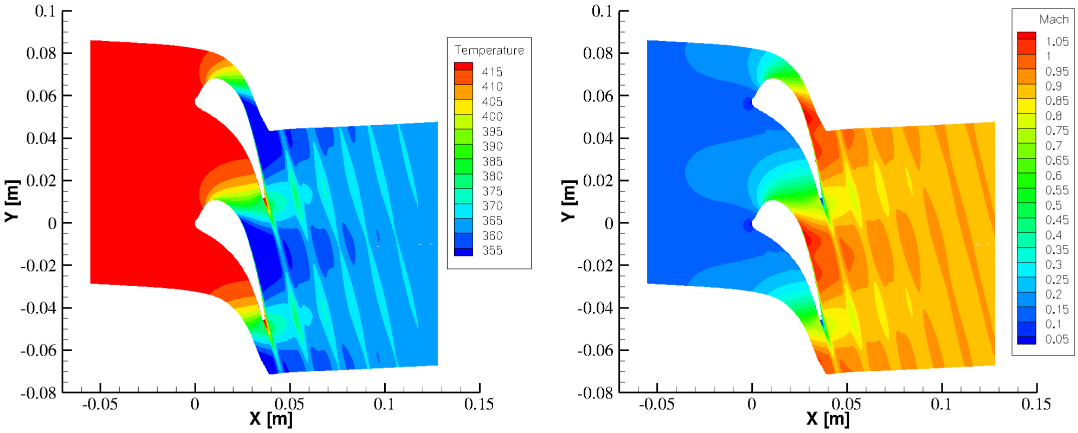
(e) Entropy delta [Pa m^3 kg^{-1}]

Figure B.12.: Flow results of the single point optimal shape using set SS9PS9 for an outlet isentropic Mach number $M_{ise,2} = 0.90$ (SI units). The exit flow angle and mass flow are approximately the same than the baseline.

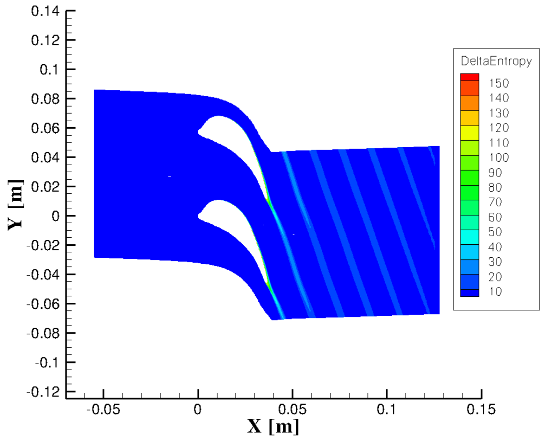
B.3.2. Single point optimal shape from chapter 6 using set SS14PS4



(a) Density [kg m^{-3}] (b) Static pressure [Pa]



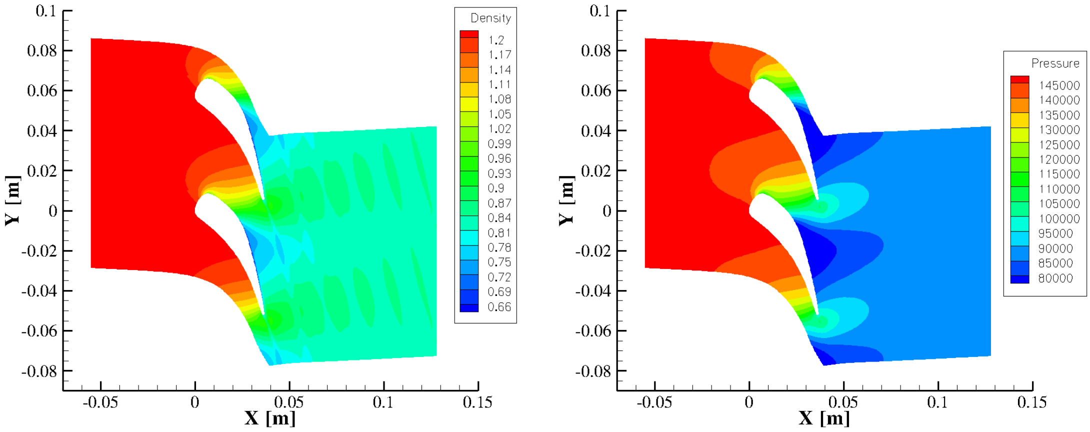
(c) Static temperature [K] (d) Mach [-]



(e) Entropy delta [Pa m^3 kg^{-1}]

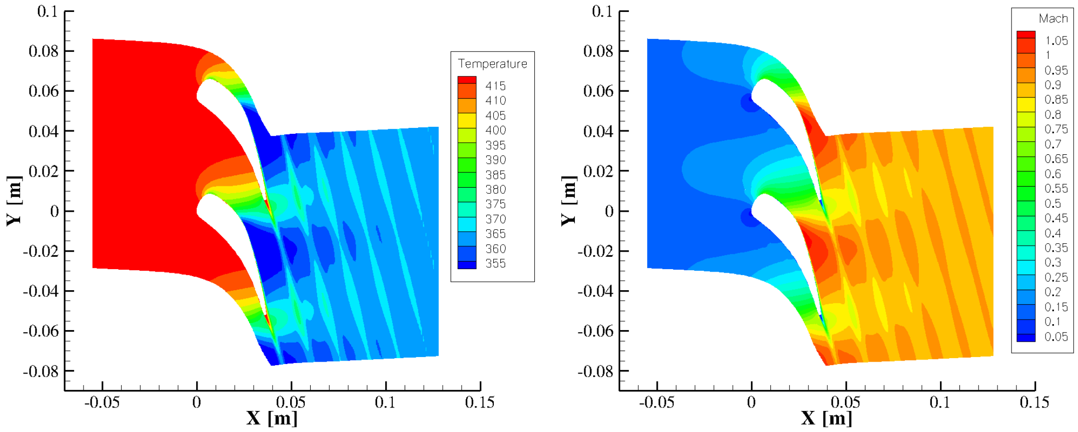
Figure B.13.: Flow results of the single point optimal shape using set SS14PS4 for an outlet isentropic Mach number $M_{ise,2} = 0.90$ (SI units). The exit flow angle and mass flow are approximately the same than the baseline.

B.3.3. Single point optimal shape from chapter 6 using the most effective subset of the set SS14PS4



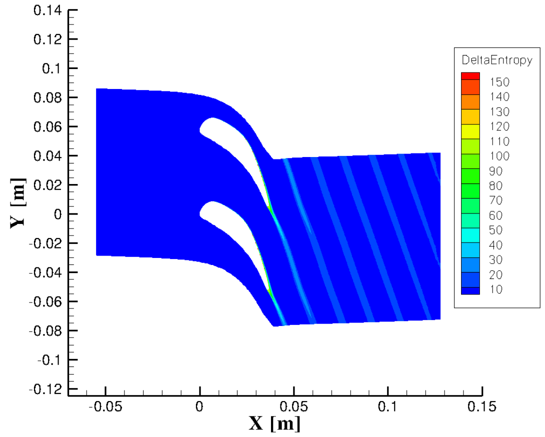
(a) Density [kg m^{-3}]

(b) Static pressure [Pa]



(c) Static temperature [K]

(d) Mach [-]



(e) Entropy delta [Pa m^3 kg^{-1}]

Figure B.14.: Flow results of the single point optimal shape using the most effective subset of the set SS14PS4 for an outlet isentropic Mach number $M_{ise,2} = 0.90$ (SI units). The exit flow angle and mass flow are approximately the same than the baseline.

Bibliography

[1] Gerrit Becker, Michael Schäfer, and Antony Jameson. An advanced nurbs fitting procedure for post-processing of grid-based shape optimizations. In *49th AIAA Aerospace sciences meeting, Orlando, FL, USA*, 2011. xi, 15, 40

[2] Arnaud Châtel, Ismael Sanchez Torreguitart, and Tom Verstraete. Using a novel pameterization effectiveness for efficient multilevel cad-based adjoint constrained optimization strategies. 2018, submitted to Advances in Engineering Software Journal. xiv, 123, 124, 126, 127

[3] Antony Jameson. Efficient aerodynamic shape optimization. *AIAA paper*, 4369:2004, 2004. 2, 15, 40, 90

[4] Majid Hojjat, Electra Stavropoulou, and Kai-Uwe Bletzinger. The vertex morphing method for node-based shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 268:494–513, 2014.

[5] Electra Stavropoulou, Majid Hojjat, and Kai-Uwe Bletzinger. In-plane mesh regularization for node-based shape optimization problems. *Computer Methods in Applied Mechanics and Engineering*, 275:39–54, 2014. 2, 15, 40, 90

[6] Jamshid A Samareh. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA journal*, 39(5):877–884, 2001. 2, 14, 15, 17, 90

[7] Vincent Braibant and Claude Fleury. Shape optimal design using b-splines. *Computer methods in applied mechanics and engineering*, 44(3):247–267, 1984. 2, 15, 90

[8] Dheeraj Agarwal, Trevor T Robinson, Cecil G Armstrong, Simão Marques, Ilias Vasilopoulos, and Marcus Meyer. Parametric design velocity computation for cad-based design optimization using adjoint methods. *Engineering with Computers*, pages 1–15, 2017. 2, 17, 35, 36, 40, 41, 51

[9] Andrea Walther and Andreas Griewank. A package for the automatic differentiation of algorithms written in c/c+. *URL https://projects. coin-or. org/ADOL-C*, 2014. 4, 46, 64, 103

[10] Tarek A El-Mihoub, Adrian A Hopgood, Lars Nolle, and Alan Battersby. Hybrid genetic algorithms: A review. *Engineering Letters*, 13(2):124–137, 2006. 8

[11] Vincent Kelner, Florin Capitanescu, Olivier Léonard, and Louis Wehenkel. A hybrid optimization technique coupling an evolutionary and a local search algorithm. *Journal of Computational and Applied Mathematics*, 215(2):448–456, 2008. 8

[12] Jacques EV Peter and Richard P Dwight. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. *Computers & Fluids*, 39(3):373–391, 2010. 8, 34, 40, 62

[13] Trevor T Robinson, Cecil G Armstrong, Hung Soon Chua, Carsten Othmer, and Thorsten Grahs. Optimizing parameterized cad geometries using sensitivities based on adjoint functions. *Computer-Aided Design and Applications*, 9(3):253–268, 2012. 9, 17

[14] Ilias Vasilopoulos, Peter Flassig, and Marcus Meyer. Cad-based aerodynamic optimization of a compressor stator using conventional and adjoint-driven approaches. In *ASME Turbo Expo 2017: Turbomachinery Technical Conference and Exposition*, pages V02CT47A004–V02CT47A004. American Society of Mechanical Engineers, 2017. 9, 17

[15] KK Chand and KD Lee. Turbomachinery blade optimization using the navier-stokes equations. *AIAA Paper*, pages 98–0933, 1997. 9, 16, 92

[16] B Kaplan and S Eyi. Inverse design of compressor cascades. *AIAA paper*, 387:2001, 2001. 9, 92

[17] James N Lyness and Cleve B Moler. Numerical differentiation of analytic functions. *SIAM Journal on Numerical Analysis*, 4(2):202–210, 1967. 10

[18] Joaquim RRA Martins, Peter Sturdza, and Juan J Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software (TOMS)*, 29(3):245–262, 2003. 10

[19] Siva Nadarajah and Antony Jameson. Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. *AIAA paper*, 2530:2001, 2001. 10

[20] Lasse Mueller and Tom Verstraete. Cad integrated multipoint adjoint-based optimization of a turbocharger radial turbine. *International Journal of Turbomachinery, Propulsion and Power*, 2(3):14, 2017. 10, 17

[21] W Kyle Anderson, James C Newman, David L Whitfield, and Eric J Nielsen. Sensitivity analysis for navier-stokes equations on unstructured meshes using complex variables. *AIAA journal*, 39(1):56–63, 2001. 10

[22] James C Newman, W Kyle Anderson, and David L Whitfield. Multidisciplinary sensitivity derivatives using complex variables. *Mississippi State University Publication, MSSU-EIRS-ERC-98-08*, 1998. 10

[23] Veer N Vatsa. Computation of sensitivity derivatives of navier–stokes equations using complex variables. *Advances in Engineering Software*, 31(8):655–659, 2000. 10

[24] Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation.* Siam, 2008. 10, 28, 46, 52

[25] T. Albring, B.Y. Zhou, N.R. Gauger, and M. Sagebaum. An aerodynamic design framework based on algorithmic differentiation. *ERCOFTAC Bulletin*, 102:10–16, 2015. 11

[26] T. Albring, M. Sagebaum, and N.R. Gauger. Development of a consistent discrete adjoint solver in an evolving aerodynamic design framework. *AIAA 2015-3240*, 2015. 11

[27] Laurent Hascoët, Mariano Vázquez, and Alain Dervieux. Automatic differentiation for optimum design, applied to sonic boom reduction. In V. Kumar, M. L. Gavrilova, C. J. K. Tan, and P. L'Ecuyer, editors, *Computational Science and Its Applications – ICCSA 2003, Proceedings of the International Conference on Computational Science and its Applications, Montreal, Canada, May 18–21, 2003. Part II*, volume 2668 of *Lecture Notes in Computer Science*, pages 85–94, Berlin, 2003. Springer. 11

[28] Marek Rabiega. Shape optimization of the turbomachine channel by a gradient methodâĂŤaccuracy improvement. *Journal of Thermal Science*, 12(3):239–244, 2003. 12

[29] Mohagna J Pandya and Oktay Baysal. Gradient-based aerodynamic shape optimization using alternating direction implicit method. *Journal of aircraft*, 34(3):346–352, 1997. 12

[30] Ch Kim, Chongam Kim, and Oh Hyun Rho. Sensitivity analysis for the navier-stokes equations with two-equation turbulence models. *AIAA journal*, 39(5):838–845, 2001. 12

[31] James Reuther, Juan Jose Alonso, Mark J Rimlinger, and Antony Jameson. Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers. *Computers & fluids*, 28(4-5):675–700, 1999. 12, 90, 91

[32] Shahrokh Shahpar and Stefano Caloni. Adjoint optimisation of a high-pressure turbine stage for a lean-burn combustion system. *Proceeding of 10th ETC, Lappeenranta, Finland*, 2013. 15

[33] Mihai C Duta, Shahrokh Shahpar, and Michael B Giles. Turbomachinery design optimization using automatic differentiated adjoint code. In *ASME Turbo Expo 2007: Power for Land, Sea, and Air*, pages 1435–1444. American Society of Mechanical Engineers, 2007.

[34] J Brezillon and NR Gauger. 2d and 3d aerodynamic shape optimisation using the adjoint approach. *Aerospace Science and Technology*, 8(8):715–727, 2004. 12, 90, 91

[35] O Pironneau. Optimal shape design for elliptic systems. 1984. 12

[36] Antony Jameson. Aerodynamic design via control theory. *Journal of scientific computing*, 3(3):233–260, 1988. 12, 15, 40

[37] Antony Jameson. Optimum aerodynamic design using cfd and control theory. *AIAA paper*, 1729:124–131, 1995. 12

[38] Charles A Mader, JR RA Martins, Juan J Alonso, and E Van Der Weide. Adjoint: An approach for the rapid development of discrete adjoint solvers. *AIAA journal*, 46(4):863–873, 2008. 12

[39] Michael B Giles and Niles A Pierce. An introduction to the adjoint approach to design. *Flow, turbulence and combustion*, 65(3-4):393–415, 2000. 40

[40] Michael B Giles, Mihai C Duta, Jens-Dominik M-uacute, ller, and Niles A Pierce. Algorithm developments for discrete adjoint methods. *AIAA journal*, 41(2):198–205, 2003.

[41] Antony Jameson. Aerodynamic shape optimization using the adjoint method. *Lectures at the Von Karman Institute, Brussels*, 2003.

[42] W Kyle Anderson and V Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Computers & Fluids*, 28(4-5):443–480, 1999.

[43] Rolf Roth and Stefan Ulbrich. A discrete adjoint approach for the optimization of unsteady turbulent flows. *Flow, turbulence and combustion*, 90(4):763–783, 2013. 12

[44] Siva Kumaran Nadarajah. *The discrete adjoint approach to aerodynamic shape optimization.* PhD thesis, stanford university Stanford, USA, 2003. 13

[45] Kyriakos C Giannakoglou and Dimitrios I Papadimitriou. Adjoint methods for shape optimization. *Optimization and computational fluid dynamics*, pages 79–108, 2008. 13, 14

[46] Hadrien Montanelli, Marc Montagnac, and François Gallard. Gradient span analysis method: application to the multipoint aerodynamic shape optimization of a turbine cascade. *Journal of turbomachinery*, 137(9):091006, 2015. 14, 62, 71, 72, 73, 76

[47] DI Papadimitriou and KC Giannakoglou. A continuous adjoint method with objective function derivatives based on boundary integrals, for inviscid and viscous flows. *Computers & Fluids*, 36(2):325–341, 2007. 14

[48] DX Wang and L He. Adjoint aerodynamic design optimization for blades in multistage turbomachinesâĂŤpart i: Methodology and verification. *Journal of Turbomachinery*, 132(2):021011, 2010. 14

[49] DX Wang, Li He, YS Li, and RG Wells. Adjoint aerodynamic design optimization for blades in multistage turbomachinesâĂŤpart ii: Validation and application. *Journal of Turbomachinery*, 132(2):021012, 2010. 14

[50] S Shahpar. Automatic aerodynamic design optimisation of turbomachinery componentsâĂŤan industrial perspective. *Optimisation Methods & Tools for Multicriteria/Multidisciplinary Design*, pages 1–40, 2004. 14

[51] Patrice Castonguay and Siva K Nadarajah. Effect of shape parameterization on aerodynamic shape optimization. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, volume 59. AIAA Reno, NV, 2007. 14, 15, 16, 38

[52] Song Wenbin and Andrew J Keane. A study of shape parameterisation methods for airfoil optimisation. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Paper AIAA*, pages 2004–4482.

[53] H-Y Wu, Shuchi Yang, Feng Liu, and H-M Tsai. Comparison of three geometric representations of airfoils for aerodynamic optimization. 2003. 14

[54] Jamshid A Samareh. Status and future of geometry modeling and grid generation for design and optimization. *Journal of Aircraft*, 36(1), 1999. 15

[55] Antony Jameson. Automatic design of transonic airfoils to reduce the shock induced pressure drag. In *Proceedings of the 31st Israel annual conference on aviation and aeronautics, Tel Aviv*, pages 5–17, 1990. 15, 40

[56] Antony Jameson. Optimum aerodynamic design via boundary control. 1994. 15, 40

[57] Nallig Leal, Esmeide Leal, and John William Branch. Simple method for constructing nurbs surfaces from unorganized points. In *Proceedings of the 19th international meshing roundtable*, pages 161–175. Springer, 2010. 15, 40

[58] Alan H Watt and Mark Watt. *Advanced animation and rendering techniques.* ACM press New York, NY, USA:, 1992. 15

[59] Alan H Barr. Global and local deformations of solid primitives. In *ACM Siggraph Computer Graphics*, volume 18, pages 21–30. ACM, 1984. 15

[60] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. *ACM SIGGRAPH computer graphics*, 20(4):151–160, 1986. 15

[61] Régis Duvigneau. *Adaptive parameterization using free-form deformation for aerodynamic shape optimization.* PhD thesis, INRIA, 2006. 15

[62] Multidisciplinary Optimization Branch. Aerodynamic shape optimization based on free-form deformation. 2004.

[63] Gaetan KW Kenway and Joaquim RRA Martins. Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration. *Journal of Aircraft*, 2014. 15

[64] Raymond Morton Hicks, Earll M Murman, and Garret N Vanderplaats. An assessment of airfoil design by numerical optimization. 1974. 16

[65] Raymond M Hicks and Preston A Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407–412, 1978. 16

[66] Siva Nadarajah and Antony Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. *AIAA paper*, 667(2000):3–17, 2000. 16

[67] Sangho Kim, Juan J Alonso, and Antony Jameson. Two-dimensional high-lift aerodynamic optimization using the continuous adjoint method. *AIAA paper*, 4741(8), 2000.

[68] James Reuther, Antony Jameson, James Farmer, Luigi Martinelli, and David Saunders. Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. *AIAA paper*, 94, 1996. 16

[69] Tom Verstraete. Cado: A computer aided design and optimization tool for turbomachinery applications. In *2nd Int. Conf. on Engineering Optimization, Lisbon, Portugal, September*, pages 6–9, 2010. 17, 34, 41, 62

[70] Arno Ronzheimer, Franz Josef Natterer, and Joël Brezillon. Aircraft wing optimization using high fidelity closely coupled cfd and csm methods. In *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, page 9078, 2010. 17

[71] Edwin Hardee, Kuang-Hua Chang, Jian Tu, Kyung K Choi, Iulian Grindeanu, and Xiaoming Yu. A cad-based design parameterization for shape optimization of elastic solids. *Advances in Engineering Software*, 30(3):185–199, 1999. 17, 40

[72] Jiri Kripac. A mechanism for persistently naming topological entities in history-based parametric solid models. *Computer-Aided Design*, 29(2):113–122, 1997. 17

[73] Marian Nemec, David W Zingg, and Thomas H Pulliam. Multipoint and multi-objective aerodynamic shape optimization. *AIAA journal*, 42(6):1057–1065, 2004. 17

[74] Dheeraj Agarwal, Christos Kapellos, Trevor R. Robinson, and Cecil G. Armstrong. Using parametric effectiveness for efficient cad-based automotive design driven by adjoint sensitivity maps. In *Eurogen 2017, Madrid, Spain, September*, 2017. 17, 36

[75] Shenren Xu, David Radford, Marcus Meyer, and Jens-Dominik Müller. Cad-based adjoint shape optimisation of a one-stage turbine with geometric constraints. In *ASME Turbo Expo 2015: Turbine Technical Conference and Exposition*, pages V02CT45A006–V02CT45A006. American Society of Mechanical Engineers, 2015. 17

[76] Les Piegl and Wayne Tiller. *The NURBS book*. Springer Science & Business Media, 2012. 18, 20, 47, 98, 119

[77] PA Sherar, CP Thompson, B Xu, and B Zhong. An optimization method based on b-spline shape functions & the knot insertion algorithm. In *World congress on engineering*, pages 862–866, 2007. 20, 38, 118

[78] Xiaocong Han and David W Zingg. An adaptive geometry parametrization for aerodynamic shape optimization. *Optimization and Engineering*, 15(1):69–91, 2014. 20, 38, 118

[79] Joe F Thompson, Bharat K Soni, and Nigel P Weatherill. *Handbook of grid generation*. CRC press, 1998. 22, 25, 44, 91

[80] B SONI. Two-and three-dimensional grid generation for internal flow applications of computational fluid dynamics. In *7th Computational Physics Conference*, page 1526, 1985. 23

[81] Joe F Thompson, Frank C Thames, and C Wayne Mastin. Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. *Journal of computational physics*, 15(3):299–319, 1974. 24, 45

[82] JL Steger and RL Sorenson. Automatic mesh-point clustering near a boundary in grid generation with elliptic partial differential equations. *Journal of computational physics*, 33(3):405–410, 1979. 27, 45

[83] A. Walther and A. Griewank. Getting started with adol-c. In U. Naumann and O. Schenk, editors, *Combinatorial Scientific Computing*, chapter 7, pages 181–202. Chapman-Hall CRC Computational Science, 2012. 32

[84] C. W. Straka. Adf95: Tool for automatic differentiation of a fortran code designed for large numbers of independent variables. *Computer Physics Communications*, 168:123–139, 2005. 32

[85] Christian H. Bischof, Alan Carle, Peyvand Khademi, and Andrew Mauer. ADIFOR 2.0: Automatic differentiation of Fortran 77 programs. *IEEE Computational Science & Engineering*, 3(3):18–32, 1996. 32

[86] Christian H. Bischof, Lucas Roh, and Andrew Mauer. ADIC — An extensible automatic differentiation tool for ANSI-C. *Software–Practice and Experience*, 27(12):1427–1456, 1997. 32

[87] L. Hascoët and V. Pascual. The Tapenade automatic differentiation tool: Principles, model, and specification. *ACM Transactions on Mathematical Software*, 39(3):20:1–20:43, 2013. 32, 34

[88] J-D Müller and P Cusdin. On the performance of discrete adjoint cfd codes using automatic differentiation. *International journal for numerical methods in fluids*, 47(8-9):939–945, 2005. 34

[89] Roe P.L. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43:357–372, 1981. 34

[90] Bram Van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *Journal of computational physics*, 32(1):101–136, 1979. 34

[91] Venkatakrishnan V. On the accuracy of limiters and convergence to steady state solutions. In *AIAA conference*, number 93-0880, 1993. 34

[92] Harten A., Hyman J.M. Self-adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of computational physics*, 50(2):235–269, 2 May 1983. 34

[93] Allmaras S.R.; Johnson F.T.; Spalart P.R. Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model. In *ICCFD7-1902, 7th International conference on computational fluid dynamics*, 2012. 34, 64, 102

[94] Blazek J. *Computational fluid dynamics: principles and applications.* Elsevier Science Ltd, Amsterdam, 2nd edition, 2001. 34, 64

[95] Antony Jameson, L Martinelli, and NA Pierce. Optimum aerodynamic design using the navier–stokes equations. *Theoretical and computational fluid dynamics*, 10(1-4):213–237, 1998. 34

[96] Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986. 34

[97] Shenren Xu, David Radford, Marcus Meyer, and Jens-Dominik Müller. Stabilisation of discrete steady adjoint solvers. *Journal of computational physics*, 299:175–195, 2015. 35, 77

[98] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)*, 23(4):550–560, 1997. 35, 63, 78

[99] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: open source scientific tools for Python, 2001. [Online; accessed 2017-02-08]. 35, 63, 78

[100] Philip E. Gill, Walter Murray, Michael A. Saunders, and Elizabeth Wong. User's guide for SNOPT 7.6: Software for large-scale nonlinear programming. Center for Computational Mathematics Report CCoM 17-1, Department of Mathematics, University of California, San Diego, La Jolla, CA, 2017. 35, 101, 103, 119

[101] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Rev.*, 47:99–131, 2005. 35, 101

[102] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009. 36

[103] Cadfix. `https://www.iti-global.com/cadfix`. Accessed: 2018-03-09. 36

[104] Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977. 36

[105] Christer Ericson. *Real-time collision detection*. CRC Press, 2004. 36

[106] Trevor T Robinson, Cecil G Armstrong, and Hung Soon Chua. Determining the parametric effectiveness of a cad model. *Engineering with Computers*, 29(1):111–126, 2013. 37, 90, 91, 94, 95, 103, 104, 105

[107] Dheeraj Agarwal, Christos Kapellos, Trevor R. Robinson, and Cecil G. Armstrong. Using parametric effectiveness for efficient cad-based adjoint optimization. 2018. 37, 95, 99

[108] DA Masters, NJ Taylor, TCS Rendall, and CB Allen. Multilevel subdivision parameterization scheme for aerodynamic shape optimization. *AIAA Journal*, pages 1–16, 2017. 38, 99, 118, 122

[109] Ismael Sanchez Torreguitart, Tom Verstraete, and Lasse Mueller. Cad kernel and grid generation algorithmic differentiation for turbomachinery adjoint optimization. In *7th European Congress on Computational Methods in Applied Sciences and Engineering, Hersonissos, Crete, Greece, June*, pages 5–10, 2016, DOI: 10.7712/100016.2077.6621. 39

[110] Olivier Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(01):97–110, 1974. 40

[111] Shanshin Chen and DA Torterelli. Three-dimensional shape optimization with variational geometry. *Structural optimization*, 13(2-3):81–94, 1997. 40

[112] Anh H Truong, David W Zingg, and Robert Haimes. Surface mesh movement algorithm for computer-aided-design-based aerodynamic shape optimization. *AIAA Journal*, 54(2):542–556, 2016. 40

[113] Jiri Kripac. A mechanism for persistently naming topological entities in history-based parametric solid models. In *Proceedings of the third ACM symposium on Solid modeling and applications*, pages 21–30. ACM, 1995. 40

[114] Srinivas Raghothama and Vadim Shapiro. Boundary representation deformation in parametric solid modeling. *ACM Transactions on Graphics (TOG)*, 17(4):259–286, 1998. 41

[115] Jiaqin Chen, Michael Freytag, and Vadim Shapiro. Shape sensitivity of constructively represented geometric models. *Computer Aided Geometric Design*, 25(7):470–488, 2008. 41

[116] RA Van den Braembussche, Olivier Léonard, and Lotvi Nekmouche. Subsonic and transonic blade design by means of analysis codes. *Computational methods for aerodynamic design (inverse) and optimization, AGARD CP*, 463, 1990. 41, 62

[117] Tony Arts and Muriel Lambert De Rouvroit. Aero-thermal performance of a two dimensional highly loaded transonic turbine nozzle guide vane: A test case for inviscid and viscous flow computations. In *ASME 1990 International Gas Turbine and Aeroengine Congress and Exposition*, pages V001T01A106–V001T01A106. American Society of Mechanical Engineers, 1990. 42, 91, 118

[118] Tony Arts, M Lambert De Rouvroit, and AW Rutherford. Aero-thermal investigation of a highly loaded transonic linear turbine guide vane cascade. A test case for inviscid and viscous flow computations. *NASA STI/Recon technical report N*, 91:23437, 1990. 41, 42, 43, 62, 65, 71, 80, 91, 118

[119] Stephane Pierret. *Designing turbomachinery blades by means of the function approximation concept based on artificial neural network, genetic algorithm, and the Navier-Stokes equations.* PhD thesis, Von Karman Institute for Fluid Dynamics, 12 1999. 42, 63

[120] Ismael Sanchez Torreguitart, Tom Verstraete, and Lasse Mueller. Optimization of the ls89 axial turbine profile using a cad and adjoint based approach. In *12 th European Conference on Turbomachinery Fluid dynamics & Thermodynamics*. EUROPEAN TURBOMACHINERY SOCIETY, 2017. 61, 123

[121] Ismael Sanchez Torreguitart, Tom Verstraete, and Lasse Mueller. Optimization of the ls89 axial turbine profile using a cad and adjoint based approach. *International Journal of Turbomachinery Propulsion and Power*, 3(3):20, 2018, DOI: 10.3390/ijtpp3030020. 61

[122] O Zweifel. The spacing of turbo-machine blading, especially with large angular deflection. *Brown Boveri Rev.*, 32(12):436–444, 1945. 67

[123] L Xu and JD Denton. The base pressure and loss of a family of four turbine blades. *ASME J. Turbomach*, 110(1):9–17, 1988. 69, 70, 123

[124] CH Sieverding, M Stanislas, and J Snoek. The base pressure problem in transonic turbine cascades. In *ASME 1979 International gas turbine conference and exhibit and solar energy conference*, pages V01BT02A019–V01BT02A019. American society of mechanical engineers, 1979. 69

[125] Claus H Sieverding, Hugues Richard, and Jean-Michel Desse. Turbine blade trailing edge flow characteristics at high subsonic outlet mach number. *Journal of turbomachinery*, 125(2):298–309, 2003. 69, 70

[126] S Vagnoli, T Verstraete, B Mateos, and CH Sieverding. Prediction of the unsteady turbine trailing edge wake flow characteristics and comparison with experimental data. *Proceedings of the institution of mechanical engineers, Part A: Journal of power and energy*, page 0957650915592074, 2015. 69, 70

[127] J Do Denton. Loss mechanisms in turbomachines. In *ASME 1993 International gas turbine and aeroengine congress and exposition*, pages V002T14A001–V002T14A001. American society of mechanical engineers, 1993. 70, 82

[128] D Corriveau and SA Sjolander. Influence of loading distribution on the performance of transonic high pressure turbine blades. In *ASME Turbo expo 2003, collocated with the 2003 international joint power generation conference*, pages 125–135. American society of mechanical engineers, 2003. 70, 82

[129] Ismael Sanchez Torreguitart, Tom Verstraete, and Lasse Mueller. Cad and adjoint based multipoint optimization of an axial turbine profile. In *International Conference on Evolutionary and Deterministic Methods for Design Optimization and Control with Applications to Industrial and Societal Problems.* EUROGEN, 2017. 75

[130] Ismael Sanchez Torreguitart, Tom Verstraete, and Lasse Mueller. *CAD and Adjoint based Multipoint Optimization of an Axial Turbine Profile.* Springer, 2017, DOI: 10.1007/978-3-319-89890-2-3. 75

[131] Edwin KP Chong and Stanislaw H Zak. *An introduction to optimization*, volume 76. John Wiley & Sons, 2013. 96

[132] Jacovisvd class from eigen libraries. `https://eigen.tuxfamily.org/dox/classEigen_1_1JacobiSVD.html`. Accessed: 2018-04-02. 96

[133] R. M. Storn K. Price and J. A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization.* Springer, Germany, 2005. 124

[134] K. Price and N. Storn. Differential Evolution. *Dr. Dobb's Journal*, pages 18–24, 1997. 124

[135] G. N. Vanderplaats. *Numerical Optimization Techniques for Engineering Design.* Vanderplaats Research & Development, Inc, New York, Third Edition 1999. 124

[136] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer, Germany, Second Edition 2006. 124

[137] A. Chatel, T. Verstraete, Gergory Coussement, and L. Mueller. Single-point optimization of the ls89 turbine cascade using a hybrid algorithm. In *ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition*, Oslo, Norway, June 2018. 124