# ADJOINT-BASED CONSTRAINED AERODYNAMIC SHAPE OPTIMIZATION FOR MULTISTAGE TURBOMACHINES

A Dissertation
Submitted to the Department of Mechanical Engineering
and the Office of Graduate and Postdoctoral Studies
of McGill University
in Partial Fulfillment of the Requirements
of the Degree of
Doctor of Philosophy

Benjamin Walther
March 2014

# Abstract

This work proposes a framework for fully-automatic gradient-based constrained aerodynamic shape optimization in a multistage turbomachinery environment. A turbomachinery solver which solves the Reynolds-averaged Navier-Stokes (RANS) equations to a steady-state in both rotating and stationary domains is developed. Characteristic-based inlet and outlet boundary conditions are imposed, while adjacent rotor and stator rows are coupled by mixing-plane interfaces. To allow for an efficient but accurate gradient calculation, the turbomachinery RANS solver is adjointed at a discrete level. The systematic approach for the development of the discrete adjoint solver is discussed. Special emphasis is put on the development of the turbomachinery specific features of the adjoint solver, i.e. on the derivation of flow-consistent adjoint inlet and outlet boundary conditions and, to allow for a concurrent rotor-stator optimization and stage coupling, on the development of an exact adjoint counterpart to the non-reflective, conservative mixing-plane formulation used in the flow solver. The adjoint solver is validated by comparing its sensitivities with finite-difference gradients obtained from the flow solver. A parallelized, automatic grid perturbation scheme utilizing radial basis functions, which is accurate and robust as well as able to handle complex multi-block grid configurations, is employed to calculate the gradient from the adjoint solution. A sequential quadratic programming algorithm is utilized to determine an improved blade shape based on the gradient information. The functionality of the proposed optimization method is demonstrated by the redesign of two different transonic compressor configurations. The design objective is to maximize the isentropic efficiency while constraining the mass flow rate and the total pressure ratio. The influence of the constraints on the design problem is investigated by comparing the results with those of an unconstrained optimization.

# Abrégé

Cette thèse présente un cadre algorithmique automatisé visant l'optimisation sous contraintes de turbomachines. Nous formulons un solveur aérodynamique stationnaire pour les équations de Navier-Stokes moyennées (RANS) en repère fixe et tournant. Des conditions limites basées sur la méthode de caractéristiques sont imposées en entrée et sortie de domaine, tandis que le couplage rotor-stator est introduit grâce à l'approche du plan de mélange. Les gradients nécessaires à la solution du problème d'optimisation sont obtenus par la méthode adjointe, implémentée sous forme discrète afin de garantir la précision du calcul des gradients. Nous discutons en détail la formulation systématique des équations adjointes discrètes. Nous mettons l'emphase sur les détails algorithmiques propres aux turbomachines, tels que les conditions limites adjointes d'entrée et de sorties ainsi que le plan de mélange adjoint non-réfléchissant. Nous présentons aussi un schéma de perturbation de maillage utilisant des fonctions de bases radiales (RBF) afin d'inclure l'influence des termes géométriques dans le calcul des gradients. La validation du solveur adjoint est démontrée en comparant les gradients calculés à ceux obtenus par la méthode des différences-finies. Finalement, ces gradients sont utilisé dans un algorithme quadratique séquentiel déterminant la forme d'aube optimale. Nous présentons deux exemples d'optimisation d'aube de compresseurs transsoniques maximisant l'efficacité isentropique tout en conservant de débit massique et la perte de charge totale. L'impact de ces contraintes est quantifié en comparant l'optimum atteint à la solution du problème d'optimisation non-contraint.

# Acknowledgements

First of all, I would like to express my utmost gratitude to my supervisor, Professor Siva Nadarajah, both for giving me the opportunity to carry out my doctoral research under his guidance and for his continuous encouragement and tremendous support throughout this work. His invaluable advice and stimulating suggestions helped me at all times and I am grateful for his patience and time. I also would like to thank the other members of my committee for their valuable suggestions and editorial help.

This research project has benefited greatly from the generous support of the Natural Sciences and Engineering Research Council of Canada (NSERC). In addition, I would like to thank MTU Aero Engines, Munich, Germany for providing the geometry and reference data of the single-stage compressor Darmstadt Rotor No. 1.

I would like to extend my gratitude to my colleagues at the Computational Aerodynamics Laboratory for their friendship and company as well as for the helpful conversations and suggestions on both technical and non-technical topics. I am especially indebted to Jean-Sébastien Cagnone for his constructive ideas and the numerous fruitful discussions from which this work greatly benefited. Furthermore, I would like to express special thanks to Vincent Poirier for sharing his thoughts and expertise on grid perturbation schemes, to Arash Mousavi for the valuable conversations on turbomachiney CFD, and to Peyman Khayatzadeh for his assistance and insight on turbulence modelling.

I am deeply grateful to my family, especially to my parents, for always believing in me and their continuous encouragement and support throughout the years. I also would like to thank all my friends who provided balance and diversion from work and made my time in Montreal truly memorable. Finally and in particular, I would like to thank Nita Perumal for her unconditional support and confidence in me.

# Nomenclature

This section lists the most important symbols and abbreviations used within this dissertation. All symbols are defined again within the thesis upon their first appearance. Some symbols represent more than just one quantity; to avoid confusion, these symbols are re-specified whenever their definition is not self-explanatory. To limit the length of this list, only the most important of the various symbols utilized to describe the turbulence model are itemized in the nomenclature.

Vector quantities can be expressed in a variety of coordinate systems. In this research, Cartesian, cylindrical, and, curvilinear coordinate formulations occur. Let $\mathbf{A}$ and $\mathbf{B}$ be two arbitrary vector quantities, where $\mathbf{A}$ takes on a velocity-vector-like form and $\mathbf{B}$ resembles the state vector of a Navier-Stokes-like equation system, then their entries shall be indexed as follows:

- in Cartesian coordinates: $\mathbf{A} = [\ a_{x_1},\ a_{x_2},\ a_{x_3}\ ]^T$ ,
- in cylindrical coordinates: $\mathbf{A} = [\ a_{x_1},\ a_{\varphi},\ a_r\ ]^T$ ,
- in curvilinear coordinates: $\mathbf{A} = [\ a_{\xi_1},\ a_{\xi_2}\ ,\ a_{\xi_3}\ ]^T$ ,

and

- in Cartesian coordinates: $\mathbf{B} = [\ b_1,\ b_{x_1},\ b_{x_2},\ b_{x_3},\ b_5\ ]^T$ ,
- in cylindrical coordinates: $\mathbf{B} = [\ b_1,\ b_{x_1},\ b_{\varphi},\ b_r,\ b_5\ ]^T$ ,
- in curvilinear coordinates: $\mathbf{B} = [\ b_1,\ b_{\xi_1},\ b_{\xi_2},\ b_{\xi_3},\ b_5\ ]^T$ .

For brevity, the notation $\mathbf{A} = [\ a_1,\ a_2,\ a_3\ ]^T$ and $\mathbf{B} = [\ b_1,\ b_2,\ b_3,\ b_4,\ b_5\ ]^T$ shall be adopted, whenever it is clear that the Cartesian coordinate system is used.

### Scalar Quantities

| | |
|---|---|
| $A$ | Cross section |
| $\mathrm{CD}_{k\omega}$ | Cross diffusion term SST turbulence model |
| $C_1$ | Sutherland constant |
| $C_2$ | Sutherland temperature |
| $c$ | Speed of sound |
| $c_p$ | Specific heat capacity |
| $c_j$ | Equality constraint |
| $\hat{c}_j$ | Inequality constraint |
| $c_{\dot{m}}$ | Constraint on mass flow rate |
| $c_{\bar{\pi}}$ | Constraint on mass-weighted total pressure ratio |
| $\mathcal{D}$ | Domain |
| $d$ | Distance parameter |
| $E$ | Specific total energy |
| $e$ | Specific internal energy |
| $h$ | Enthalpy |
| $I$ | Objective function |
| $J$ | Cell volume |
| $\mathcal{J}$ | Jacobian determinate of the transformation matrix $\boldsymbol{\mathcal{K}}$ |
| $k$ | Turbulent kinetic energy |
| $k$ | Thermal conductivity |
| $k_l$ | Thermal conductivity based on laminar viscosity |
| $k_t$ | Thermal conductivity based on turbulent viscosity |
| $L$ | Chord length |
| $L_{\mathrm{mix}}$ | Mixing length |
| $\mathcal{L}$ | Lagrangian function |
| $l_P$ | Pitch length |
| $M$ | Mach number |
| $\dot{m}$ | Mass flow rate |
| $N_s$ | Number of surface grid points |
| $N_v$ | Number of interior volume grid points |
| $N_{\mathrm{RBF}}$ | Number of RBF points |

| | |
|---|---|
| $\mathcal{N}_{ij}$ | Arc lengths ratio of algebraic grid perturbation scheme |
| $P_W$ | Shaft power |
| $P_\tau$ | Production term SST model |
| $Pr$ | Prandtl number |
| $Pr_l$ | Laminar Prandtl number |
| $Pr_t$ | Turbulent Prandtl number |
| $p$ | Pressure |
| $q_\infty$ | Prescribed boundary quantities |
| $R$ | Specific gas constant |
| $Re$ | Reynolds number |
| $Ri$ | Riemann invariant |
| $r$ | Radius |
| $r_{\text{SRAD}}$ | Support radius of RBF grid perturbation scheme |
| $S_{nm}$ | Cell face metrics |
| $s$ | Entropy |
| $T$ | Temperature |
| $t$ | Time |
| $U_n$ | Contravariant velocity |
| $u^+$ | Dimensionless velocity |
| $u_\tau$ | Friction velocity |
| $\mathcal{V}$ | Volume |
| $x_1, x_2, x_3$ | Cartesian coordinates |
| $x_1, \varphi, r$ | Cylindrical coordinates |
| $y^+$ | Dimensionless wall distance |
| $\alpha_k$ | Stage coefficients of the modified Runge-Kutta scheme |
| $\alpha_\varphi$ | Inflow angle in circumferential direction |
| $\alpha_r$ | Inflow angle in radial direction |
| $\beta_k$ | Blending coefficients of the modified Runge-Kutta scheme |
| $\Gamma$ | Freestream turbulence intensity |
| $\gamma$ | Ratio of specific heats |
| $\hat{\delta}_{ij}$ | Kronecker delta function |
| $\epsilon$ | Turbulent dissipation rate |

| | |
|---|---|
| $\epsilon$ | Smoothing coefficients for implicit residual smoothing |
| $\eta$ | Efficiency |
| $\lambda$ | Bulk (second) viscosity coefficient |
| $\lambda$ | Spectral radius |
| $\tilde{\lambda}$ | Scaled spectral radius |
| $\Lambda$ | Scaled spectral radius at a cell face |
| $\mu$ | Dynamic (first) viscosity coefficient |
| $\mu_l$ | Laminar viscosity coefficient |
| $\mu_t$ | Turbulent viscosity coefficient |
| $\nu$ | Kinematic viscosity |
| $\nu_t$ | Turbulent kinematic viscosity |
| $\nu^{(2)}$ | Pressure sensor for first-order difference of JST scheme |
| $\nu^{(4)}$ | Pressure sensor for third-order difference of JST scheme |
| $\pi$ | Total pressure ratio |
| $\rho$ | Density |
| $\xi_1, \xi_2, \xi_3$ | Curvilinear coordinates |
| $\tau_w$ | Wall shear stress |
| $\phi$ | Basis function of RBF grid perturbation scheme |
| $\Omega'$ | Vorticity magnitude |
| $\omega$ | Specific dissipation rate |

## Vector Quantities

| | |
|---|---|
| $\mathbf{A}$ | Convective flux Jacobian matrix in physical space |
| $\boldsymbol{\mathcal{A}}$ | Convective flux Jacobian matrix in computational space |
| $\mathbf{A}_{\text{RBF}}$ | RBF volume point basis function matrix |
| $\mathbf{C}$ | Perturbation matrix of algebraic grid perturbation scheme |
| $\delta\mathbf{C}$ | Characteristic variable vector |
| $\hat{\mathbf{e}}$ | Unit vector |
| $\mathbf{F}$ | Convective flux vector in physical space |
| $\mathbf{F_d}$ | Artificial dissipation flux vector in physical space |
| $\mathbf{F_v}$ | Viscous flux vector in physical space |
| $\bar{\mathbf{F}}$ | Averaged convective flux vector |

| | |
|---|---|
| $\boldsymbol{\mathcal{F}}$ | Convective flux vector in computational space |
| $\boldsymbol{\mathcal{F}}_{\mathbf{d}}$ | Artificial dissipation flux vector in computational space |
| $\boldsymbol{\mathcal{F}}_{\mathbf{v}}$ | Viscous flux vector in computational space |
| $\boldsymbol{\mathfrak{F}}$ | Discrete convective adjoint flux vector |
| $\boldsymbol{\mathfrak{F}}_{\mathbf{d}}$ | Discrete artificial dissipation adjoint flux vector |
| $\boldsymbol{\mathfrak{F}}_{\mathbf{v}}$ | Discrete viscous adjoint flux vector |
| $\mathbf{f}$ | Discretized convective flux vector |
| $\mathbf{f}_{\mathbf{d}}$ | Discretized artificial dissipation flux vector |
| $\mathbf{f}_{\mathbf{v}}$ | Discretized viscous flux vector |
| $\hat{\mathbf{I}}_h^H$ | Multigrid interpolation operator for the fine grid solution |
| $\mathbf{I}_h^H$ | Multigrid interpolation operator for the residual |
| $\mathbf{I}_H^h$ | Multigrid prolongation operator |
| $\boldsymbol{\mathcal{K}}$ | Transformation matrix physical/computational space |
| $\mathbf{M}$ | Transformation matrix primitive/conservative variables |
| $\mathbf{M}^{-1}$ | Transformation matrix conservative/primitive variables |
| $\mathbf{M}_{\text{RBF}}$ | RBF basis function matrix |
| $\mathbf{n}$ | Unit normal vector |
| $\mathbf{P}$ | Right eigenvector matrix of the convective flux Jacobian |
| $\mathbf{P}^{-1}$ | Left eigenvector matrix of the convective flux Jacobian |
| $\boldsymbol{\mathcal{P}}$ | Search direction |
| $\mathbf{Q}$ | Source term vector |
| $\mathbf{Q}_{\mathbf{H}}$ | Heat flux vector |
| $\boldsymbol{\mathcal{Q}}$ | Multigrid forcing function |
| $\mathbf{R}$ | Residual vector in physical space |
| $\mathbf{R}_{\mathbf{c}}$ | Residual vector comprised only of convective terms |
| $\mathbf{R}_{\mathbf{d}}$ | Residual vector comprised only of dissipative terms |
| $\boldsymbol{\mathcal{R}}$ | Residual vector in computational space |
| $\hat{\boldsymbol{\mathcal{R}}}$ | Residual vector excluding source term contribution |
| $\hat{\mathbf{t}}$ | Unit tangent vector |
| $\mathbf{U}$ | Absolute velocity vector |
| $\mathbf{V}$ | Relative velocity vector |
| $\mathbf{W}$ | Conservative variable vector |

| | |
|---|---|
| $\widetilde{\mathbf{W}}$ | Mixed-out conservative variable vector |
| $\delta\widetilde{\mathbf{W}}_{\mathbf{c}}$ | Boundary state correction for conservative variable vector |
| $\mathbf{W}_{\mathbf{P}}$ | Primitive variable vector |
| $\widetilde{\mathbf{W}}_{\mathbf{P}}$ | Mixed-out primitive variable vector |
| $\mathbf{X}$ | Cartesian coordinate vector |
| $\mathbf{X}_{\mathbf{C}}$ | Cylindrical coordinate vector |
| $\mathbf{X}_{\mathbf{s}}$ | Surface grid points design variables |
| $\mathbf{X}_{\mathbf{b}}$ | Hicks-Henne bump functions design variables |
| $\mathbf{X}_{\mathbf{v}}$ | Interior volume grid points |
| $\mathbf{X}_{\text{RBF}}$ | RBF points |
| $\delta\mathbf{X}_{\mathbf{sb}}$ | Grid sensitivity matrix surface grid points to Hicks-Henne bumps |
| $\delta\mathbf{X}_{\mathbf{vs}}$ | Grid sensitivity matrix volume grid points to surface grid points |
| $\Delta\mathbf{X}_{\mathbf{s}}$ | Displacement vector of surface grid points |
| $\Delta\mathbf{X}_{\mathbf{b}}$ | Displacement vector of Hicks-Henne bump functions |
| $\Delta\mathbf{X}_{\mathbf{v}}$ | Displacement vector of volume grid points |
| $\Delta\mathbf{X}_{\text{RBF}}$ | Displacement vector of RBF points |
| $\boldsymbol{\alpha}$ | Design variable vector |
| $\boldsymbol{\beta}$ | RBF coefficients |
| $\boldsymbol{\epsilon}$ | Slack variables |
| $\boldsymbol{\Lambda}$ | Diagonal matrix of the eigenvalues of the flux Jacobian matrix |
| $\boldsymbol{\tau}$ | Viscous stress tensor |
| $\boldsymbol{\xi}$ | Curvilinear coordinate vector |
| $\boldsymbol{\psi}$ | Adjoint variable vector |
| $\boldsymbol{\Omega}$ | Angular velocity vector |

## Subscripts

| | |
|---|---|
| 0 | Initial or reference quantity |
| B | Quantity at domain boundary |
| D | Domain cell quantity |
| $f$ | Freestream quantity |
| H | Halo cell quantity |
| Hub | Quantity located at the hub |

| | |
|---|---|
| $H$ | Coarse grid quantity |
| $h$ | Fine grid quantity |
| $i, j, k, l, m, n$ | Cell and summation indices |
| in | Inlet quantity |
| is | Isentropic quantity |
| out | Outlet quantity |
| $R$ | Rotor quantity |
| $S$ | Stator quantity |
| $t$ | Total quantity |
| rel | Relative quantity |
| wall | Quantity at a solid wall boundary |

**Superscripts**

| | |
|---|---|
| $^*$ | Dimensionless quantity |
| $^{*,**,***}$ | Smoothed residuals |
| avg | Averaged quantity |
| $k$ | $k$-th stage within the modified Runge-Kutta scheme |
| $n$ | Time level |
| new | Perturbed grid point |
| old | Unperturbed grid point |
| $T$ | Transpose |

**Abbreviations**

| | |
|---|---|
| ACARE | Advisory Council for Aeronautics Research in Europe |
| AD | Automatic differentiation |
| BFGS | Broyden-Fletcher-Goldfarb-Shanno |
| CEV | Constant eddy viscosity |
| CFD | Computational fluid dynamics |
| CFL | Courant-Friedrich-Lewy |
| ICAO | International Civil Aviation Organization |
| KKT | Karush-Kuhn-Tucker |
| LEAP | Leading Edge Aviation Propulsion |

| | |
|---|---|
| MPI | Message Passing Interface |
| NLFD | Nonlinear Frequency Domain |
| NSERC | National Sciences and Engineering Research Council of Canada |
| RANS | Reynolds-averaged Navier-Stokes |
| RBF | Radial basis function |
| SD | Steepest Descent |
| SNOPT | Sparse nonlinear optimizer |
| SQP | Sequential quadratic programming |
| SST | Shear-Stress Transport |
| VKI | Von-Kármán-Institute |

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

More than fifty years after the development of the Rolls-Royce Conway, the world's first production turbofan, and after decades of extensive research in the field of jet propulsion, modern turbofan engines have reached a stage where further improvements are increasingly hard to obtain. On the other hand, the aviation industry faces market demands which pose bigger challenges than ever before.

Over the past decades, the worldwide air traffic, both measured in passenger kilometres and in the active aircraft fleet, has been growing continuously. Although the future growth might temporarily stagnate due to economical, political or violent extraneous events, in the long term, air traffic is concordantly expected to continue its steady growth. In December 2012, the International Civil Aviation Organisation (ICAO) reported that global air transport moved some 2.9 billion people in 2012, while the United Nations agency said it expects the number to reach more than six billion by 2030, cf. [1]. At the same time, ICAO announced that the total scheduled passenger traffic grew by 5.5% in 2012, after an increase of 6.5% in 2011. These numbers are in line with the organisation's long term forecast, which expects an annual growth of 4-5% over the next twenty years. However, along this generally positive outlook, the aviation industry faces new challenges as well. In addition to reducing the fuel burn to the lowest possible level, attention is focusing more and more on topics such as the reduction of the pollutant emissions and the minimization of noise pollution. To lower the environmental pollution, various aerospace associations have progressively defined stricter standards and policies. For instance, in 2001 the

Advisory Council for Aeronautics Research in Europe (ACARE) published a "Vision 2020" report, cf. [2], in which the committee established the following targets to be met by the industry until 2020 (compared to the standard in 2001):

- the reduction of both the fuel consumption and the $CO_2$ emissions by 50%,
- to lower the $NO_x$ footprint by 80%,
- and to reduce the noise perception by 50%.

Furthermore, ACARE identified three main contributors for achieving the above targets; new aircraft, more efficient engines, and an improved air traffic management. For example, the advisory council suggested that by 2020 technology improvements and advanced engine design capabilities should result in more economical aircraft engines which emit 15-20% less $CO_2$ in comparison to the 2001 standard.

To satisfy the ambitious market demands for more economic and environmental-friendly jet engines, the industry faces the challenge to further advance the technically matured conventional turbofan configurations and, to meet the long term requirements, to eventually develop and realize new and innovative aircraft engine concepts. The latest generation of high-bypass turbofan engines, for instance the CFM International LEAP engine jointly developed by GE Aviation and Snecma or Pratt & Whitney's geared turbofan engine PW1000G, which the US-based engine manufacturer developed in collaboration with MTU Aero Engines, show remarkable improvements in fuel burn and pollutant emissions. Yet, the long term goals are still only fulfilled partially and, thus, the need for further advancements remains.

The recent design improvements can also be tributed to the development of new and more sophisticated design tools. Especially, the aerodynamic design strategy applied to compressor and turbine design benefited from the introduction of more advanced, numerical design techniques, which were gradually developed over the past decades. With the exponential growth of computational power and the extensive research in the field of computational fluid dynamics (CFD), numerical simulations have become more and more accurate and CFD is routinely employed in industry as an analysis tool within a turbomachine's standard aerodynamic design process. High-fidelity computational solutions have tremendously improved the understanding of the turbomachinery flow physics and offer valuable information that had to be obtained from expensive experiments before. As a result, the gas turbine efficiency could be improved considerably by considering a more realistic flow environment during the design process.

Further improvements in compressor or turbine efficiency demand a thorough understanding of their complex flow physics. These include three-dimensional flow phenomena, such as secondary flow effects or the interaction between tip leakage vortices and passage shocks potentially appearing in transonic rotor flows, as well as unsteady flow effects, for example caused by rotor-stator interactions in multi-row configurations. Following the conventional inverse design approach, in which the airfoil shape is determined based on a prescribed pressure distribution and flow field, an improved aerodynamic compressor or turbine design depends largely on the aerodynamicist's experience and knowledge. However, due to the growing number of aspects to be considered during the design process, the aerodynamicists face an incrementally more challenging design problem, which makes further improvements increasingly hard to obtain. Thus, the emerging trend is to use CFD not only for flow analysis but to also incorporate numerical optimization techniques into the design process. This direct design approach can guide the designer to find an improved design and provides a means to gain better insight into the design space of the underlying problem. As a consequence, automatic shape optimization has recently gained significant attention within the field of compressor and turbine blade design. However, considering that CFD flow simulations are still fairly time-consuming and can take several hours or even days, the question of how to build an efficient numerical optimization tool for automatic turbomachinery blade design arises naturally.

It is the motivation of this dissertation to find an answer to this question and to propose a framework for efficient, but accurate, automatic aerodynamic shape optimization in a turbomachinery environment. The development of such a design tool can help to further advance the aerodynamic designs of modern compressors and turbines and thus can contribute to the development of next-generation turbofan engines which meet the challenging future market demands.

The following introductory sections present the concept of numerical optimization applied to aerodynamic shape design, introduce the most common approaches, and elaborate on their advantages and disadvantages. Based on the outcome of the comparison, an optimization method is selected and, thereafter, its characteristics are discussed in more detail. Furthermore, a detailed review of the method's role in both academic and industrial research, from its first appearance to the current state-of-the-art, with a focus on its application to compressor and turbine blade design, is provided. Then, the framework of the proposed automatic design process is outlined. Afterwards, the motivation behind this work and the individual goals and contributions of this research are highlighted. Finally, the structure of the thesis as well as the content of the individual chapters is described.

## 1.1 Numerical Optimization for Turbomachines

Since its introduction into the field of gas turbine design to study turbomachinery aerodynamics, CFD has played an important role in the aerodynamic design process of compressor and turbine blades. However, CFD has mostly been employed as an analysis tool to evaluate the aerodynamic performance of certain compressor and turbine configurations or other aerodynamic devices. Only recently, more and more attempts have been made to utilize CFD as a direct design tool for aerodynamic shape optimization in a turbomachinery environment. The increase in computational power led to the advent of reasonable-priced high performance computing platforms, which together with the development of new and efficient analysis and design algorithms opened the door for numerical optimization using CFD.

Optimization is the minimization or maximization of a function subject to constraints on its variables. In the context of automatic aerodynamic shape optimization in turbomachinery this means: a numerical program tries to modify the geometry of a certain flow device, e.g. a stator blade, such that a user-defined performance parameter, e.g. the total pressure loss, is improved while satisfying the underlying flow governing equations, e.g. the Reynolds-averaged Navier-Stokes equations.

Generally speaking, numerical optimization methods can be divided into two main categories: non-gradient and gradient-based methods. Derivative-free optimization techniques, such as genetic algorithms or grid search and non-linear simplex methods, only require the evaluation of the prescribed objective function; no gradient calculation or Hessian estimate is needed. In theory, these methods are able to locate the global minimum within the defined design space. Yet, their applicability to complex design problems is limited due to the extremely long runtime and huge computational cost resulting from the large number of function evaluations that are typically necessary to converge to an optimum solution; especially when the evaluation of the objective function is costly. Gradient-based optimization methods instead are usually able to locate a local optimum within much fewer design iterations. However, since these techniques require an estimate of the objective function gradient with respect to the design variables, and possibly higher derivatives such as the Hessian, this second group of methods depends on the smoothness of the variation of the objective function and the ability to produce continuous derivatives. Once the gradient information is available it is used to determine a search direction suitable to optimize the design problem. Then, a step into this direction of improvement is taken and the procedure is repeated until a local optimum is found. The simplest gradient-based optimization method follows the approach of the steepest descent. The technique is characterized by a very low implementation cost and by the fact that it only requires the evaluation of the objective function gradient to determine the search direction. However, this popular method is relatively inefficient since it typically requires a considerable amount of design iterations to converge to an optimum solution. Quasi-Newton methods feature superior convergence properties and usually are more efficient. At the same time though, Quasi-Newton methods are computationally more expensive since they estimate the Hessian in order to find a better suited search direction.

The estimation of the objective function gradient can be an expensive task and often is the most time consuming portion of a gradient-based optimization algorithm. Consequently, the performance of a gradient-based optimization method largely depends on an efficient gradient calculation. Gradient information may be calculated from a variety of different approaches, cf. Figure 1.1. For instance, finite-difference offers a simple way to approximate the objective function derivatives. Each design variable is perturbed individually and a subsequent evaluation of the objective function then provides the sensitivity derivatives via finite-differences. Hence, the computational cost to obtain the gradient increases proportionally to the number of design variables. Clearly, this direct approach rapidly becomes inefficient for design problems where the number of design parameters is high and significantly larger than

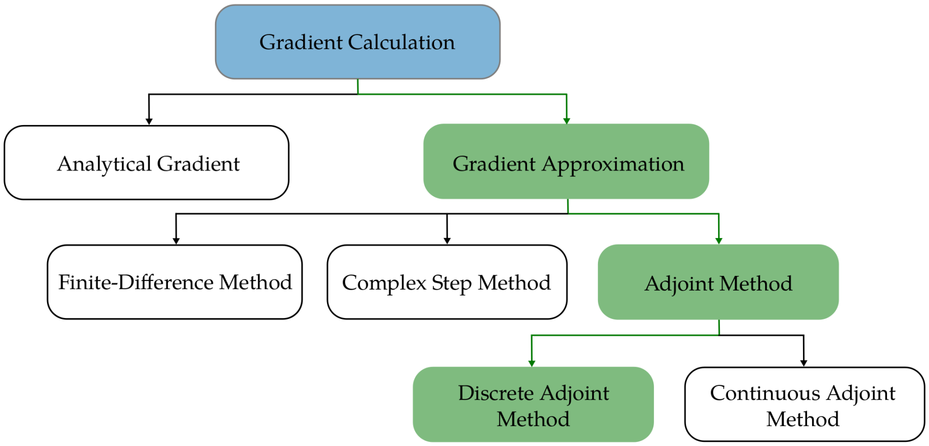

Figure 1.1: Gradient evaluation.

the number of functionals. Besides, the accuracy of the finite-difference gradient is a function of the step size used to perturb the design variables. The appropriate step length is not known *a priori* and thus has to be determined from a gradient study. A step length too large violates the finite-difference approximation and a step length too small may introduce significant subtractive cancellation errors. Utilizing the complex step method circumvents this additional second disadvantage, since in this case the gradient evaluation is absent of subtractive cancellation errors. Nonetheless, the computational cost is still proportional to the number of design variables since the complex step approach requires a separate function evaluation for each design variable perturbation as well.

Control theory, on the other hand, offers a means to calculate the objective function sensitivities nearly independent of the number of design variables. Following this so-called adjoint approach, the gradient is calculated indirectly by solving an adjoint system, which complexity is of the same order as that of the underlying flow governing equations. All sensitivities are obtained at the cost of one flow solution and one adjoint solution per function of interest. Thus, for cases where the underlying optimization problem involves a large number of design variables, adjoint-based gradient calculation is considerably more economical compared to the classical finite-difference based techniques, especially if the evaluation of the objective function requires the solution of an equation system as complex as the Euler or Navier-Stokes equations. Yet, this approach requires the development of a second solver to calculate the solution of the adjoint equations. This is not a trivial task and, thus, the cost involved in developing an adjoint-based optimization method is considerably larger than in developing a finite-difference based method.

Consequently, both derivative-free and gradient-based optimization techniques have contrasting advantages and disadvantages. There is not one particular optimization method that is generally superior to all the others. The key is rather to analyse the design problem beforehand and to choose a method that is appropriate for the specific mathematical model. For example, when developing a new aircraft engine, in the conceptual design stage the designers must first decide on the number and the type of engines based on the targeted aircraft. Then, the general characteristics of the aircraft engine, such as thrust, bypass ratio, and overall pressure ratio, are specified to meet the market requirements. The engine architecture is defined in terms of number of spools, the key identities of the individual components are determined, and important flow properties such as the total pressure and the total temperature of the flow are prescribed at certain stations. During this preliminary design stage, low-fidelity methods are often utilized to assess the designs. It is at this stage of the engine design process where derivative-free optimization methods can play an important role and may be able to reduce the turn-around time to come up with new designs. The computational cost of the analysis tools employed at this design stage is typically low. Hence, function evaluations are relatively cheap, which in turn drastically attenuates the disadvantage of applying an optimization technique that requires a considerable amount of function calls. Furthermore, gradient-free optimization algorithms handle both integer and continuous variables, while gradient-based methods only work with the latter. This makes derivative-free optimization methods well suited for preliminary design cases, since the capability of these algorithms to handle integer variables and their ability to cope with objective functions that do not have smooth derivatives provides the possibility to tackle design problems which involve integer or non-smooth variables. Furthermore, the ability to search for the global optimum is a feature that can be very beneficial within the preliminary design stage. On the other hand, at a more advanced design stage where the basic engine configuration is laid out and high-fidelity tools are utilized to refine the design, gradient-based optimization methods usually represent the better choice, since at this design level function evaluations tend to be costly and thus the number of function calls is to be minimized. For instance, the calculation of the three-dimensional, viscous flow field within a single-stage compressor or turbine, and thus the evaluation of the objective function, typically takes several hours for a steady-state solution or even days for an unsteady solution. Moreover, nearly all current turbomachinery design problems require a large number of design variables, generally in the hundreds or even thousands, to guarantee an adequate geometrical parametrization. Since finite-difference based gradient calculation involves an equivalent number of flow evaluations, this classical approach is inefficient as well. This leads to the adjoint approach. For aerodynamic shape optimization in a turbomachinery environment using a high-fidelity CFD solver, the adjoint method is considerably more economical and offers tremendous computational cost savings over alternative approaches.

Based on the findings of the comparison, for the optimization framework developed within this research a gradient-based optimization method is selected and preferred over a derivative-free technique. Furthermore, to guarantee an efficient gradient and accurate calculation it is decided to compute the objective function derivatives via the adjoint approach and to employ a Quasi-Newton method to determine the search direction and, thus, an improved blade design. The gradient estimation methodology utilized within the developed optimization framework is highlighted in Figure 1.1.

## 1.2 Adjoint-Based Optimization and its Application to Turbomachinery

The adjoint method, based on the mathematical theory for the control of systems governed by partial differential equations, cf. Lions [3], was introduced into the field of fluid mechanics by Pironneau [4] for solving elliptical problems in 1984 and extended by Jameson [5] to transonic flow problems in 1988. The adjoint equations can be obtained from two different approaches. In the continuous approach the adjoint equations are first derived from the governing equations and are discretized after, i.e. control theory is applied to the differential equations governing the flow. In particular, the objective function and the governing equations are linearized with respect to the flow variables and design parameters and their variations are then combined through the use of Lagrange multipliers, also called co-state or adjoint variables. Collecting the terms associated with the variation of the flow variables provides the continuous adjoint equation and its boundary conditions, while the terms associated with the design variable variations form the gradient equation. The discrete approach derives the adjoint system by applying control theory directly to the discretized flow equations, i.e. the discrete adjoint equation and the corresponding gradient expression are obtained from the linearization of the underlying flow solver. A numerical code based on the discrete adjoint approach typically requires more memory and CPU time than a code based on the continuous adjoint formulation. However, a discrete adjoint solver is naturally consistent with its flow solver, which is advantageous in gradient-based optimization since it usually leads to better convergence properties. A fully-linearized discrete adjoint solver produces the exact gradients of the discretized functionals and the adjoint derivatives are identical to those obtained from the flow solver. Another advantage of the discrete approach is that the adjoint boundary conditions are obtained straightforward by linearizing the flow boundary conditions, although their derivation can be challenging. For a more detailed analysis and comparison of the two different approaches, see Nadarajah and Jameson [6, 7].

Over the last twenty years, the adjoint approach has been applied extensively to external flow problems. In 1994 and 1995, Jameson [8, 9] introduced the continuous adjoint method to the two-dimensional and three-dimensional Euler equations and demonstrated its potential by successfully optimizing airfoils and wings. Since then, both the continuous and the discrete adjoint approaches have been investigated for various objective functions, cf. [10, 11, 12], and extended to increasingly complex problems, such as the optimization of complete aircraft configurations in viscous flow, cf. [13, 14], or airfoil design in an unsteady flow environment, cf. [15, 16].

Nonetheless, due to the complexity of formulating the adjoint approach and the challenging task of properly defining the adjoint boundary conditions for internal flow problems, the application of adjoint-based optimization to internal flows has been lagging far behind. Only within the last few years efforts have been made to apply the adjoint method to gradient-based optimization in a turbomachinery environment. For instance, in 2003, Yang et al. [17] developed the continuous adjoint approach for gradient calculation and applied it to the inverse design of two-dimensional cascade blades in an inviscid flow environment. One year later, Chung et al. [18] presented a discrete adjoint formulation for a three-dimensional Euler solver and tested it successfully for inverse design applied to Rotor 37. Furthermore, Wu et al. [19] calculated gradients based on the adjoint method and performed a constrained optimization of the VKI turbine stator in inviscid flow and the Standard Configuration 4 turbine stator in both laminar and turbulent flow. Papadimitriou and Giannakoglou [20, 21] presented continuous and discrete adjoint formulations with objective function derivatives based only on boundary integrals and applied them to compressor and turbine blade design in two- and three-dimensional inviscid and viscous flows. Papadimitriou and Giannakoglou [22] also introduced a total pressure loss objective function to the continuous adjoint approach, which then was successfully minimized in compressor and turbine cascades while maintaining the flow turning and the blade thickness. In 2008, Corral and Gisbert [23] developed a discrete adjoint solver to minimize the secondary flow losses in a three-dimensional low pressure turbine vane by redesigning the turbine hub end wall. Moreover, while Li et al. [24] utilized the continuous adjoint approach to perform aerodynamic shape optimization on a turbine cascade in two-dimensional turbulent flow, an adjoint-based optimization of a three-dimensional turbine stator in which the stagger angle was used as a design variable in addition to the profile shape was presented by Luo et al. [25] in 2011. Luo et al. [26] also applied the adjoint method to constrained multipoint optimization; they treated the constraints by a penalty function approach and redesigned Rotor 67 through blade profiling while considering three different compressor operation points. Finally, in 2010 and 2011, Mousavi and Nadarajah [27, 28] used the continuous adjoint approach to optimize the heat transfer in cooled gas turbine blades by defining the profile shape, the location of the cooling hole and the coolant injection angle as the design variables.

While in the last ten years the adjoint approach gained more and more attention within the turbomachinery community, nearly all design cases were restricted to the optimization of isolated stator or rotor configurations. Yet, similar to a flow mixing-plane calculation, cf. Denton [29], a concurrent optimization of all rows in a multistage compressor or turbine would lead to an optimized design reflecting the real flow environment to a higher degree than a redesign based on the isolated optimization of each blade. Combining all redesigned blades obtained from isolated optimizations would most likely lead to a multistage configuration that is less optimal than the redesign produced by a true multistage optimization, in which all blades are optimized simultaneously. Adjoint solvers that allow for a multi-row optimization were presented by Frey et al. [30], Wang and He [31, 32], and Walther and Nadarajah [33, 34]. Frey et al. [30] adjointed a conservative mixing-plane based on Giles' exact two-dimensional non-reflecting boundary conditions to obtain the equivalent discrete adjoint boundary formulation and performed a sensitivity analysis. Their adjoint gradients were in good agreement with the corresponding finite-difference derivatives; however, no optimization was performed. In 2010, Wang and He [31, 32] developed a conservative adjoint mixing-plane which couples averaged co-states based on the one-dimensional characteristics for their continuous adjoint solver. They applied their adjoint-based optimization method to different test cases in both two- and three-dimensional viscous flow including a single-stage transonic compressor and a three-stage industrial transonic compressor. For all their test cases they documented remarkable efficiency gains which demonstrated the potential of aerodynamic design optimization using the adjoint approach in a multistage environment. Their design objective was to minimize the entropy generation rate, while constraints were individually weighted and included as penalty terms into the objective function. Following this approach, the weights have to be determined by trial and error. Since different weights usually lead to different designs, the choice of the weights is crucial and can be a time-consuming task, as documented by Wu et al. [19] and Wang and He [32]. Walther and Nadarajah [33, 34] developed the discrete adjoint equations for a turbomachinery RANS solver and proposed a framework for fully-automatic constrained aerodynamic shape optimization in a multistage turbomachinery environment. They derived an exact adjoint counterpart to the mixing-plane formulation used in their flow solver and utilized a sequential quadratic programming algorithm, which enabled them to determine an improved blade design without having to introduce weights for each constraint. They applied the optimization method to several compressor configurations and presented redesigns which showed considerable efficiency improvements while satisfying the prescribed constraints.

## 1.3 Optimization Framework

Figure 1.2 summarises the design process of the optimization framework used within this research and compares the adjoint approach with the traditional finite-difference method. In an initial step, the design problem is specified, i.e. the geometry of the flow device and the surrounding domain are discretized, the design variables are defined, the design objective is selected, and the potential constraints are prescribed. Then, the flow solution is calculated for the baseline geometry. The next step requires the solution of the corresponding adjoint system. It is important to note that the adjoint equations have to be solved separately for the objective function and each constraint, i.e. $m + 1$ times where $m$ is the number of constraints. The adjoint solution then provides the information necessary to calculate the gradient for all design variables without having to re-evaluate the governing equations. However, the gradient calculation requires a separate perturbation of each design variable and, thus, of the geometry as well as of the grid. Once the gradient calculation is complete, the objective function derivatives and sensitivities of the constraints are provided to the optimizer. The optimization algorithm then determines the shape modifications based on the gradient information and, thus, proposes a new design. Afterwards, the flow field is calculated for the modified geometry and the objective function as well as the constraints are evaluated. In case the design objective is improved and the constraints are met, the new geometry is accepted and the design cycle is started again with solving the adjoint equations for the new design. This cycle is repeated until the optimization converges to an optimum solution. In case the design objective did not improve or constraints were violated, the shape modifications are discarded and the algorithm returns to the optimizer where a new line search is performed and a new shape is determined. This new geometry is then assessed again by calculating the flow field and evaluating the objective function and the constraints. This sub-cycle is repeated until a new geometry is found which improves the design objective and satisfies the constraints.

If the finite-difference method is used instead of the adjoint approach, the main structure will remain the same, as illustrated in Figure 1.2. However, the gradient calculation differs and requires a separate flow evaluation for each design variable perturbation, which makes the traditional approach inefficient and expensive.

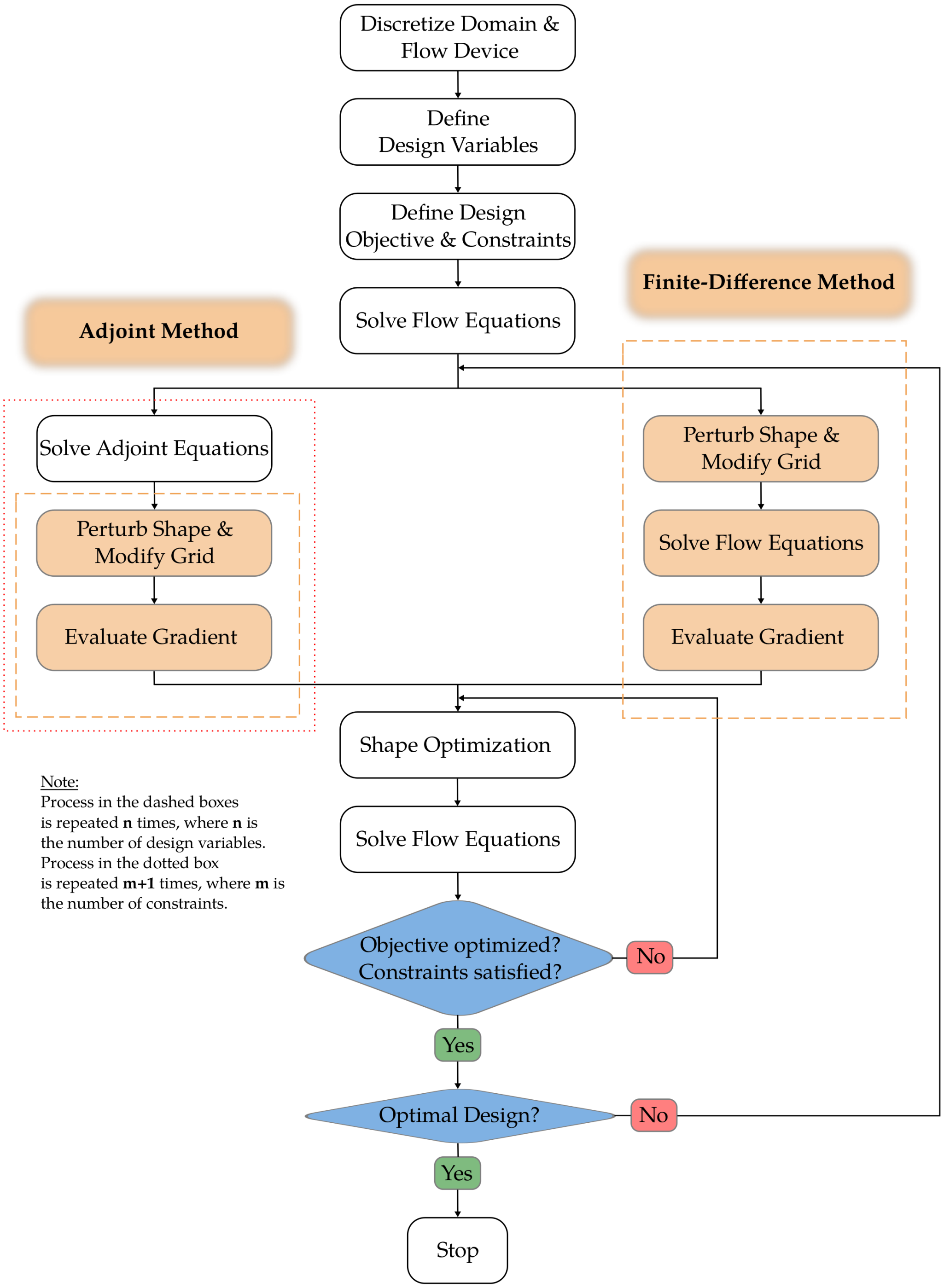
Discretize Domain & Flow Device
Define Design Variables
Define Design Objective & Constraints
Finite-Difference Method
Adjoint Method
Solve Flow Equations
Solve Adjoint Equations
Perturb Shape & Modify Grid
Evaluate Gradient
Perturb Shape & Modify Grid
Solve Flow Equations
Evaluate Gradient
Shape Optimization
Note:
Process in the dashed boxes is repeated **n** times, where **n** is the number of design variables.
Process in the dotted box is repeated **m+1** times, where **m** is the number of constraints.
Solve Flow Equations
Objective optimized? Constraints satisfied?
No
Yes
Optimal Design?
No
Yes
Stop

Figure 1.2: Design process of the proposed optimization method.

## 1.4 Research Objectives and Contributions

Numerical optimization, particularly the adjoint method, will gradually reform the aerodynamic design process and eventually replace the traditional inverse design approach with a multidisciplinary direct design strategy. In the future, CFD-based optimization tools will become an integrated part of the aerodynamic design process and will be used routinely just like CFD flow analysis tools are employed today.

### 1.4.1 Research Objectives

The main objective of this dissertation is to contribute to the recent research efforts made to apply gradient-based aerodynamic shape optimization methods to automatic compressor and turbine blade design and to advance the state-of-the-art in the field of adjoint-based gradient calculation for multistage turbomachinery flows. The motivation is to develop an efficient but accurate design tool that can help to improve the aerodynamic design of current turbomachinery configurations and at the same time provide the aerodynamicist with new design capabilities which will allow him to gain better insight into the underlying design problem and, thus, will help him to explore new approaches and designs.

The realization of this goal requires the development of an optimization framework which includes both a fast numerical algorithm for flow analysis and a rapid adjoint solver. Furthermore, the objective function gradients and the derivatives of potential constraints need to be calculated efficiently. Lastly, the optimization method should feature a rapid convergence to the optimum solution and provide a means to treat constraints with little effort. Thus, this research work involves several independent tasks and objectives.

**Flow Solver.** A flow solver is to be developed which solves the Reynolds-averaged Navier-Stokes (RANS) equations in a multistage turbomachinery environment. Hence, the numerical algorithm requires the ability to solve the governing equations in both rotating and stationary domains. Characteristic-based boundary conditions are to be imposed at inlet and outlet boundaries in order to avoid reflections which can compromise the solution accuracy within the domain. To obtain a steady-state multistage solution, adjacent rotor and stator rows have to be coupled by mixing-plane interfaces, which exchange averaged flow variables between two neighbouring domains such that the fluxes across a rotor-stator interface are conserved.

**Adjoint Solver.** To allow for an efficient but accurate gradient calculation, the turbomachinery RANS solver is to be adjointed at a discrete level. The systematic approach for the development of the discrete adjoint solver shall be discussed in detail. Special emphasis will be placed on the development of the turbomachinery specific features of the adjoint solver, i.e. on the derivation of flow-consistent adjoint inlet and outlet boundary conditions and, to allow for a concurrent rotor-stator optimization and stage coupling, on the development of an exact adjoint counterpart to the non-reflective, conservative mixing-plane formulation used in the flow solver. While the development of the discrete adjoint solver will follow a similar approach as described by Frey et al. [30], the treatment of the boundary fluxes in the underlying flow solver will differ, hence, different adjoint boundary conditions will be obtained. Furthermore, in contrast to Frey's adjoint solver, in which the numerical adjoint fluxes are approximated by finite-differences, the present optimization method shall be based on exact adjoint fluxes, which are to be obtained by manually differentiating the flux routines of the primal solver. Using finite-difference approximated adjoint fluxes circumvents the work involved in linearizing the fairly complex flux discretization; yet, it increases the computational cost and introduces an additional source of error. It is worthwhile mentioning automatic differentiation (AD) as an alternative approach to manual linearization. AD tools have been successfully applied to the adjoint approach, cf. [35], and can reduce the cost involved in developing the adjoint solver, yet, at the expense of higher memory requirements. Therefore, in this work manual differentiation is preferred. The dual adjoint solver shall be validated by comparing its sensitivities with finite-difference gradients obtained from the primal flow solver.

**Gradient Calculation.** The adjoint method provides a means to calculate gradients efficiently without having to re-evaluate the flow. However, the approach still requires the computational grid to be perturbed for each design variable. Manual grid perturbation defeats the purpose of an automatic optimization method with as little user interaction as possible. Thus, an automatic grid perturbation algorithm which is fast, accurate, and robust is to be implemented within this optimization framework. The grid movement scheme shall be able to handle complex multi-block grid configurations.

**Optimization Strategy.** Once the sensitivities of the objective functional and all constraints have been calculated with the help of the adjoint solver, a sequential-quadratic programming (SQP) algorithm shall be employed to find an optimized design. The SQP method offers an elegant way to treat constraints within an optimization problem by building and solving a quadratic sub-problem constructed of the objective and the linearized constraints and avoids the time-consuming task to determine penalty coefficients for the constraints, which is necessary with a penalty function approach, cf. [19, 32] The search direction will be determined by a Quasi-Newton method, while a general line search technique shall be used to find the appropriate step length into the direction of improvement.

**Application.** The functionality of the proposed optimization method is to be demonstrated. Two different transonic compressor configurations are to be redesigned considering both an inviscid and viscous flow environment. The objective shall be to maximize the isentropic efficiency while constraining the mass flow rate and the total pressure ratio. To investigate the influence of the constraints, an unconstrained optimization will be performed as well and the results shall be compared.

### 1.4.2 Contributions

The main contributions of this thesis towards aerodynamic shape optimization methods for automatic turbomachinery design may be summarized as follows:

- Development of the discrete adjoint equations of a turbomachinery RANS solver including a systematic approach for the derivation and efficient implementation of flow-consistent adjoint boundary conditions and exact adjoint mixing-plane formulations.

- A novel parallel grid deformation scheme based on a radial basis functions approach for highly-resolved complex three-dimensional viscous multi-block turbomachinery grids.

- Introduction of the SQP method into an adjoint-based optimization framework for multistage compressor design, which determines shape modifications that improve the design objective while satisfying the prescribed constraints.

## 1.5 Thesis Outline

The outline of the thesis follows the general optimization cycle described in subsection 1.3. First, chapter 2 presents the mathematical model employed within this research and introduces the equations on which basis the flow solver is developed and its solution is evaluated. The main features of the turbomachinery flow solver are described in chapter 3. It develops the non-dimensional form of the field equations, introduces some useful coordinate transformations, and presents the numerical discretization and solution methodology utilized to solve the RANS equations. Special focus is put on the development of the turbomachinery-specific boundary conditions. Chapter 4 then discusses the adjoint method in detail and elaborately describes the development of the adjoint equations for the tubomachinery RANS solver. This chapter especially expatiates on the derivation of the flow-consistent adjoint boundary conditions and mixing-plane formulation. The gradient calculation, including the choice of the design variables and two different grid perturbation schemes, as well as the optimization strategy, including the treatment of the constraints, are presented in chapter 5. Thereafter, the functionality of the proposed optimization method is demonstrated in chapter 6. First, the developed method is validated by investigating both the flow and adjoint solutions and by comparing the adjoint gradients with finite-difference sensitivities obtained from the flow solver. Then, two different compressor configurations are re-designed under various flow conditions. Finally, chapter 7 summarises the main outcomes of this dissertation and provides some ideas for future work.

# Chapter 2

# Governing Equations

This chapter presents the mathematical model employed within this research. The equations introduced in the following sections form the basis on which the flow solver is developed and its solution is evaluated. First, section 2.1 reviews the fundamental thermodynamic relations and assumptions applicable to a perfect gas. The mathematical form of the flow governing equations as well as their boundary conditions are presented, cf. sections 2.2 and 2.3. Thereupon, the concept of turbulence and the approach utilized to model the eddy viscosity is briefly discussed in section 2.4. Lastly, section 2.5 introduces some useful gasdynamic relations which are employed to evaluate the flow solution and to quantify the aerodynamic performance of a compressor.

## 2.1 Thermodynamics of a Perfect Gas

In pure aerodynamics, it is generally reasonable to assume that the working fluid behaves like a calorically perfect gas, i.e. the specific heat capacity $c_p$ is constant and does neither depend on the pressure nor on the temperature. For fluids with such properties the equation of state takes the form

$$p = \rho R T \ . \tag{2.1}$$

The ideal gas law (2.1) relates the pressure $p$ to the density $\rho$ and the temperature $T$ through the specific gas constant $R$ which, for air, takes the value $R =$

287.058 J/(kgK). Furthermore, the specific internal energy $e$ can be calculated from

$$e = \frac{1}{\gamma - 1}\frac{p}{\rho} \ , \tag{2.2}$$

where $\gamma$ is the ratio of specific heats defined as

$$\gamma = \frac{c_p}{c_p - R} \ . \tag{2.3}$$

The specific total energy $E$ is the sum of the internal energy and the kinetic energy, i.e.

$$E = e + \frac{|\mathbf{U}|^2}{2} \ , \tag{2.4}$$

where $\mathbf{U} = [u_1, u_2, u_3]^T$ is the absolute velocity vector and $|\mathbf{U}|^2 = u_1^2 + u_2^2 + u_3^2$. Combining Eqn. (2.2) with Eqn. (2.4) provides an alternative, convenient form of the equation of state,

$$p = (\gamma - 1)\, \rho \left[E - \frac{|U|^2}{2}\right] \ , \tag{2.5}$$

which expresses the pressure in terms of the conservative variables. Moreover, Fourier's law of heat conduction is usually used to calculate the heat transfer $\mathbf{Q_H}$ due to temperature gradients $\boldsymbol{\nabla} T$, i.e.

$$\mathbf{Q_H} = -k \boldsymbol{\nabla} T \ , \tag{2.6}$$

where $k$ denotes the thermal conductivity coefficient and which is obtained from

$$k = \frac{c_p \mu}{Pr} \ , \tag{2.7}$$

where $\mu$ is the dynamic viscosity coefficient and $Pr$ is the Prandtl number. While, for air, the Prandtl number is usually assumed to be constant, $Pr = 0.72$, the viscosity coefficient may be calculated from the Sutherland equation defined as

$$\mu = C_1 \frac{T^{\frac{3}{2}}}{T + C_2} \ , \tag{2.8}$$

where $C_1 = 1.458 \cdot 10^{-6}$ kg/(ms$\sqrt{\mathrm{K}}$) is the Sutherland constant and $C_2 = 110.4$ K represents the Sutherland temperature.

## 2.2 Three-dimensional Navier-Stokes Equations

The three-dimensional Navier-Stokes equations, coupled with the continuity and energy equation, are typically accepted to describe the viscous compressible flow within a turbomachinery environment and are used within this research as the underlying governing equations. These equations, based on the fundamental works of Navier [36], de Saint-Venant [37], and Stokes [38], describe the conservation of mass, momentum, and energy through a system of coupled differential equations. The Navier-Stokes equations can be written in a vector invariant form, i.e independent of the coordinate system. For turbomachinery applications most commonly a cylindrical or Cartesian coordinate formulation is adopted.

Multi-row turbomachinery applications involve rotating motions and require the Navier-Stokes equations to be solved simultaneously in multiple blade rows, some of which are stationary and some of which are rotating. Hence the question arises whether it is best to cast and solve the governing equations in an inertial frame of reference or in a rotating coordinate system. A flow solver utilizing a fixed frame formulation in a rotating domain requires the computational mesh to be physically moved, i.e. the metric terms associated with the mesh have to be recalculated after each time step. A rotating frame code, on the other hand, allows to freeze the grid motion and therefore usually represents the more convenient and computationally cheaper approach. Two different possibilities exist to cast the governing equations in a rotating frame of reference; either the relative or absolute velocities can be used as the dependent variables. Employing a relative velocity formulation is useful for problems in which the entire system is moving, e.g. the simulation of an isolated rotor row. However, for problems such as multi-row turbomachinery calculations, which involve interaction between stationary and rotating blade rows, the relative velocity approach would require a transformation of the dependent variable vector between fixed and rotating frames before information could be exchanged between adjacent blade rows. Furthermore, when the Navier-Stokes equations are solved in a rotating frame of reference, the acceleration of the fluid is augmented by additional terms that appear in the momentum equations. Employing the absolute velocity approach, the centripetal and Coriolis accelerations can be collapsed into a single term which yields to simpler expressions. Therefore, in this research a flow solver is developed that solves the Navier-Stokes equations in a rotating Cartesian coordinate system utilizing the absolute velocity formulation. The corresponding mathematical form of the Navier-Stokes equations is presented in the following. For a more detailed discussion and comparison of the different approaches in an inertial frame of reference and rotating frame of reference as well as a comparison between the absolute and relative formulations, the reader should refer to Chen et al. [39]

Using Einstein notation, i.e. summation over $i = 1, 2, 3$ is implied by a repeated index $i$, the three-dimensional compressible Navier-Stokes equations in differential, conservative form are given by

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} - \frac{\partial \mathbf{F}_{\mathbf{v}i}}{\partial x_i} = \mathbf{Q} \quad \text{in } \mathcal{D} \ , \tag{2.9}$$

where $x_i$ denotes the Cartesian coordinates, $t$ is the time, and $\mathcal{D}$ is the domain under consideration. The vector of conserved variables $\mathbf{W}$, the inviscid flux vector $\mathbf{F}$, and the viscous flux vector $\mathbf{F_v}$ are described respectively by

$$\mathbf{W} = \left\{ \begin{array}{c} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{array} \right\} , \ \mathbf{F}_i = \left\{ \begin{array}{c} \rho v_i \\ \rho u_1 v_i + p\hat{\delta}_{i1} \\ \rho u_2 v_i + p\hat{\delta}_{i2} \\ \rho u_3 v_i + p\hat{\delta}_{i3} \\ \rho E v_i + p u_i \end{array} \right\} , \ \mathbf{F}_{\mathbf{v}i} = \left\{ \begin{array}{c} 0 \\ \tau_{ij}\hat{\delta}_{i1} \\ \tau_{ij}\hat{\delta}_{i2} \\ \tau_{ij}\hat{\delta}_{i3} \\ u_j \tau_{ij} + k\frac{\partial T}{\partial x_i} \end{array} \right\} , \tag{2.10}$$

while the source term $\mathbf{Q}$ comprises all potential body forces. In the above definitions, $\hat{\delta}_{ij}$ is the Kronecker delta function, $\mathbf{U} = [u_1, u_2, u_3]^T$ is the absolute velocity vector, and $\mathbf{V} = [v_1, v_2, v_3]^T$ is the relative velocity vector. Furthermore, the absolute velocity is defined as the sum of the relative velocity and the entrainment velocity $\mathbf{\Omega} \times \mathbf{X}$, where $\mathbf{\Omega} = [\Omega_1, \Omega_2, \Omega_3]^T$ and $\mathbf{X} = [x_1, x_2, x_3]^T$. In a rotating rotor domain the source term $\mathbf{Q}$ accounts for the effects due to the Coriolis force and centrifugal force and reads as

$$\mathbf{Q} = \left\{ \begin{array}{c} 0 \\ -\rho(\Omega_2 u_3 - \Omega_3 u_2) \\ -\rho(\Omega_3 u_1 - \Omega_1 u_3) \\ -\rho(\Omega_1 u_2 - \Omega_2 u_1) \\ 0 \end{array} \right\} . \tag{2.11}$$

In a stationary stator domain the angular velocity $\mathbf{\Omega}$ is zero and, hence, the source term vanishes. In the present research work, the rotational axis of a rotor corresponds to the $x_1$-axis and the angular velocity is assumed to be constant, i.e.

$$\mathbf{V} = \left\{ \begin{array}{c} u_1 \\ u_2 + \Omega_1 x_3 \\ u_3 - \Omega_1 x_2 \end{array} \right\} \qquad \text{and} \qquad \mathbf{\Omega} \times \mathbf{X} = \left\{ \begin{array}{c} 0 \\ -\Omega_1 x_3 \\ \Omega_1 x_2 \end{array} \right\} . \tag{2.12}$$

Likewise, the source term simplifies to

$$\mathbf{Q} = \left\{ \begin{array}{c} 0 \\ 0 \\ \rho \Omega_1 u_3 \\ -\rho \Omega_1 u_2 \\ 0 \end{array} \right\} . \tag{2.13}$$

Assuming a Newtonian fluid, the applied shear varies linearly with the strain rate and the viscous stresses $\tau_{ij}$ may be written as

$$\tau_{ij} = \mu \left[ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] + \lambda \left[ \frac{\partial u_k}{\partial x_k} \right] \hat{\delta}_{ij} \ , \tag{2.14}$$

where $\mu$ is the dynamic (or first) viscosity coefficient and $\lambda$ stands for the bulk (or second) viscosity coefficient. The viscosity coefficients relate the momentum fluxes to the velocity gradients. Following Stokes' hypothesis the two coefficients are related through $\lambda = -2\mu/3$, i.e.

$$\tau_{ij} = \mu \left[ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \hat{\delta}_{ij} \right] \ . \tag{2.15}$$

The inviscid Euler equations are obtained from the Navier-Stokes equations by dropping the viscous flux gradient $\partial \mathbf{F}_{\mathbf{v}i} / \partial x_i$ in Eqn. (2.9).

Chapter 6 will present both three- and two-dimensional results. The corresponding two-dimensional formulation of the Navier-Stokes equations is obtained by neglecting the momentum equation for the $x_3$-direction and by setting the velocity components $u_3$ and $v_3$ to zero. Furthermore, in two dimensions, the rotor only per-forms are translational motion into the $x_2$-direction. Consequently, the additional rotational forces do not occur and the source term $\mathbf{Q}$ vanishes. The entrainment velocity simplifies to $[0, \Omega_2 r]^T$, where $r$ is a length scale specific to the investigated flow problem.

## 2.3 Boundary Conditions

At a solid wall, boundary conditions have to be applied to complete the mathematical model. In case of inviscid flow, the velocity at the wall must be tangent to the slope of the wall. This corresponds to a zero flux through the wall and thus

$$(\mathbf{U} \cdot \mathbf{n})_{\text{wall}} = 0 \ , \tag{2.16}$$

where $\mathbf{n} = [n_1, n_2, n_3]^T$ is the wall surface unit normal vector. For viscous problems, the no-injection and no-slip conditions are imposed and require an additional boundary condition to the one above,

$$(\mathbf{U} \cdot \hat{\mathbf{t}})_{\text{wall}} = 0 \ , \tag{2.17}$$

where $\hat{\mathbf{t}} = [\hat{t}_1, \hat{t}_2, \hat{t}_3]^T$ is the wall unit tangent vector. This effectively means that the velocity at the wall is zero. The above boundary conditions satisfy the momentum equation. For the case of the energy equation, either an adiabatic or isothermal boundary condition is required. In this work, an adiabatic boundary condition is employed and defined as

$$(\mathbf{Q_H} \cdot \mathbf{n})_{\text{wall}} = 0 \ . \tag{2.18}$$

This translates to a zero heat flux through the normal of the wall.

## 2.4 Turbulence Modelling

Despite the exponential growth of computational power, the onset of turbulence at high Reynolds numbers possesses a challenge to accurately solve the time-dependent Navier-Stokes equations introduced in section 2.2. The direct simulation of turbulent flows still continues to present a significant problem and, as of today, is only feasible for relatively simple flow problems at low Reynolds numbers. Instead, the effects of turbulence usually have to be accounted for in an approximate manner. As a consequence, a large variety of turbulence models were, and still are, developed, cf. Wilcox [40].

### 2.4.1 The Reynolds-Averaged Navier-Stokes Equations

In 1895, Reynolds presented a first approach for the approximate treatment of turbulent flows. Following this methodology, the flow variables are decomposed into a mean value and an instantaneous, fluctuating part. Applying the so-called Reynolds averaging procedure to the Navier-Stokes equations, cf. Eqns. (2.9) and (2.10), leads to the well-known Reynolds-averaged Navier-Stokes (RANS) equations. The RANS equations are formally identical to the Navier-Stokes equations except for an additional term which constitutes the so-called Reynolds-stress tensor and represents the transfer of momentum due to turbulent fluctuations. For a detailed derivation of the RANS equations and an elaborate discussion of the different Reynolds averaging techniques the reader should refer to Wilcox [40] and Blazek [41].

The Boussinesq hypothesis, on which the most common turbulence models are grounded, assumes that the turbulent shear stress is related linearly to the mean rate of strain, i.e. the Reynolds-stress tensor is calculated as the product of a proportionality factor, the so-called eddy viscosity $\mu_t$, and the mean strain rate. Applying the Boussinesq hypothesis to model the Reynolds-stress terms in the RANS equations, the first viscosity coefficient $\mu$ is replaced by the sum of a laminar and a turbulent component, i.e.

$$\mu = \mu_l + \mu_t \ . \tag{2.19}$$

Accordingly, the thermal conductivity coefficient $k$, cf. Eqn. (2.7), is substituted by

$$k = k_l + k_t = c_p \left( \frac{\mu_l}{Pr_l} + \frac{\mu_t}{Pr_t} \right) \ . \tag{2.20}$$

In Eqns. (2.19) and (2.20), the laminar viscosity coefficient $\mu_l$ is calculated from the Sutherland equation, cf. Eqn. (2.7), and the turbulent viscosity coefficient $\mu_t$ has to be determined by a turbulence model. Both, the laminar and turbulent Prandtl numbers are typically assumed to be constant and take the values $Pr_l = 0.72$ and $Pr_t = 0.9$.

### 2.4.2 The SST Turbulence Model

For engineering applications, the RANS equations and the Boussinesq hypothesis have become the basis for the simulation of turbulent flows and a large variety of turbulence closures were developed to calculate the eddy viscosity $\mu_t$. In this research, Menter's two-equation Shear-Stress Transport (SST) model [42, 43, 44] is used to estimate the turbulent viscosity coefficient. The SST turbulence model merges Wilcox' $k$-$\omega$ model [40, 45] with the standard high Reynolds Number $k$-$\epsilon$ model and seeks to combine the positive features of both turbulence models. In particular, the $k$-$\omega$ model is employed in the sublayer and logarithmic part of a boundary layer, while the $k$-$\epsilon$ model is utilized in the wake region of a boundary layer and in free shear flows. Because of the switch to the $k$-$\epsilon$ model in the outer wake regions and free shear layers, the SST model does not suffer from the deficiency of an undesirable freestream dependency as it is the case with the $k$-$\omega$ model. Furthermore, the eddy viscosity definition is modified in comparison to the original $k$-$\omega$ model such that it accounts for the effect of the transport of the principal turbulent shear stress. The SST eddy viscosity model is widely recognized to be superior to their alternatives, especially in the prediction of adverse pressure gradient flows. In the following, the main components and characteristics of the SST turbulence model are briefly discussed.

Following Menter [42], the conservative form of the SST model is defined as

$$\frac{\partial \rho k}{\partial t} = P_\tau - \beta^* \rho \omega k + \frac{\partial}{\partial x_j}\left[(\mu_l + \sigma_k \mu_t)\frac{\partial k}{\partial x_j}\right] , \tag{2.21}$$

$$\frac{\partial \rho \omega}{\partial t} = \frac{\gamma}{\nu_t} P_\tau - \beta \rho \omega^2 + \frac{\partial}{\partial x_j}\left[(\mu_l + \sigma_\omega \mu_t)\frac{\partial \omega}{\partial x_j}\right] + 2\rho(1 - F_1)\frac{1}{\rho}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} , \tag{2.22}$$

where the $k$-$\epsilon$ model has been transformed into a $k$-$\omega$-like formulation. Besides slightly different modelling constants and a revised eddy viscosity definition, the main difference compared to the standard $k$-$\omega$ model is an additional cross diffusion term which appears in the equation defining the specific dissipation rate $\omega$, cf. Eqn. (2.22). The production term $P_\tau$, originally defined as $P_\tau = \tau_{ij}(\partial u_i/\partial x_j)$, is approximated using a vorticity-based source term

$$P_\tau = \mu_t \Omega'^2 - \frac{2}{3}\rho k \delta_{ij}\frac{\partial u_i}{\partial x_j} \ . \tag{2.23}$$

The approximation (2.23) is commonly used because the vorticity magnitude $\Omega'$ is usually readily available in most RANS solvers. Furthermore, the vorticity source term is often nearly identical to the exact production term in boundary layer flows and the use of the vorticity term can avoid some numerical difficulties sometimes associated with the use of the exact source term. For more details the reader should refer to [43]. A production limiter,

$$P_\tau = \min\left(P_\tau, 20\beta^*\rho\omega k\right) \ , \tag{2.24}$$

is employed in the equation of the turbulent kinetic energy $k$ to prevent the build-up of turbulence in stagnation regions. The turbulent viscosity $\mu_t$ is computed from

$$\mu_t = \frac{\rho a_1 k}{\max(a_1\omega, \Omega' F_2)} \ , \tag{2.25}$$

where $F_2$ is a function that is one in boundary layer flows and zero in free shear layers. Specifically, $F_2$ is a hyperbolic tangent function defined as

$$F_2 = \tanh(\mathrm{arg}_2^2) \ , \tag{2.26}$$

with the argument

$$\mathrm{arg}_2 = \max\left(2\frac{\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{d^2\omega}\right) \ , \tag{2.27}$$

where

$$\nu = \frac{\mu_l}{\rho} \ , \tag{2.28}$$

and $d$ is the distance from the field point to the nearest no-slip boundary. Note, in Eqn. (2.22), the turbulent kinematic viscosity $\nu_t$ is obtained from

$$\nu_t = \frac{\mu_t}{\rho} \ . \tag{2.29}$$

Each constant of the SST model is a blend of the constants of the inner $k$-$\omega$ model and the outer $k$-$\epsilon$ model. In particular, if $\phi_1$ represents any constant in the original $k$-$\omega$ model, and $\phi_2$ any constant in the transformed $k$-$\epsilon$ model, then $\phi$, the corresponding constant of the SST model given by Eqns. (2.21) and (2.22) is computed via

$$\phi = F_1\phi_1 + (1 - F_1)\phi_2 \ , \tag{2.30}$$

where $F_1$ is a second hyperbolic tangent function defined as

$$F_1 = \tanh(\arg_1^4) \ , \tag{2.31}$$

with the argument

$$\arg_1 = \min\left[\max\left(\frac{\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{d^2\omega}\right), \frac{4\rho\sigma_{\omega_2}k}{\mathrm{CD}_{k\omega}d^2}\right] \ , \tag{2.32}$$

and $\mathrm{CD}_{k\omega}$, which represents the positive portion of the cross diffusion term of (2.22), being calculated from

$$\mathrm{CD}_{k\omega} = \max\left(2\rho\sigma_{\omega_2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial w}{\partial x_j}, 10^{-20}\right) \ . \tag{2.33}$$

The choice of the terms within Eqn. (2.32) is detailed in [42]; they are selected such that $F_1$ goes to zero near the boundary layer edge and equals to one in the sublayer.

Finally, the closing constants are defined as

$$\gamma_1 = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega_1}k^2}{\sqrt{\beta^*}}, \qquad \gamma_2 = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega_2}k^2}{\sqrt{\beta^*}}, \tag{2.34}$$

and

$$\begin{array}{lllllllll} \sigma_{k_1} & = & 0.85 \ , & \sigma_{\omega_1} & = & 0.5 \ , & \beta_1 & = & 0.075 \ , \\ \sigma_{k_2} & = & 1.0 \ , & \sigma_{\omega_2} & = & 0.856 \ , & \beta_2 & = & 0.0828 \ , \\ \beta^* & = & 0.09 \ , & \kappa & = & 0.41 \ , & a_1 & = & 0.31 \ . \end{array} \tag{2.35}$$

At a solid wall, the turbulent kinetic energy and dissipation rate are obtained from

$$k_{\text{wall}} = 0 \qquad \text{and} \qquad \omega_{\text{wall}} = \frac{60\nu}{\beta_1 d_{\text{wall}}^2} \ , \tag{2.36}$$

where $d_{\text{wall}}$ is the distance from the solid wall to the closest point away from the wall. Furthermore, for internal flow problems inflow conditions need to be defined. This work follows the suggestions from Moore and Moore [46] and Koubogiannis et al. [47] and the turbulent inflow quantities are calculated from

$$k_{\text{in}} = \frac{3}{2}(\Gamma u_{\text{in}})^2 \qquad \text{and} \qquad \omega_{\text{in}} = \frac{\sqrt{k_{\text{in}}}}{\beta^* L_{\text{mix}}} \ . \tag{2.37}$$

Here, $\Gamma$ is the freestream turbulence intensity which is defined to be 0.08165% and $u_{\text{in}}$ is the absolute inflow velocity. $L_{\text{mix}}$ is the mixing length which is set to 4% of the pitch length of the most upstream compressor row.

## 2.5 Turbomachinery Gasdynamics

The performance of a compressor is typically characterized by its total pressure ratio and an efficiency factor that indicates how much additional work is required compared to an ideal compressor. Furthermore, the rotational speed of the shaft and the work necessary to produce the pressure rise are important quantities to evaluate a compressor's overall performance.

Ideal compression, i.e. assuming that no heat is added to or extracted from the flow and that no energy transformations occur due to friction or dissipative effects, is an isentropic process. For the isentropic flow of a perfect gas a pressure increase can be related to a change in temperature through

$$\frac{T_3}{T_2} = \left(\frac{p_3}{p_2}\right)^{\frac{\gamma-1}{\gamma}} \ , \tag{2.38}$$

or in terms of total flow quantities through

$$\frac{T_{t,3}}{T_{t,2}} = \left(\frac{p_{t,3}}{p_{t,2}}\right)^{\frac{\gamma-1}{\gamma}} \ , \tag{2.39}$$

where, following typical labelling procedure for gas turbines, 2 represents the state before the compression and 3 the state after the compression. Since

$$h = c_p T \qquad \text{and} \qquad h_t = c_p T_t \ , \tag{2.40}$$

a pressure rise can only be realized with an increase in enthalpy $h$. Thus, a compressor must perform work on the fluid, which is achieved through the following process: The rotor reduces the relative kinetic head of the fluid and adds it to the absolute kinetic head, i.e. the relative velocity decreases in a rotor section while the absolute velocity component increases. In the downstream stator row the increased absolute kinetic head of the fluid is reduced again, which in turn leads to a pressure rise.

In reality, a compression is not isentropic; instead the entropy increases and, as displayed in Figure 2.1, additional work is needed to compensate for the inefficiency and to obtain the desired total pressure ratio. The shaft power necessary to obtain a specific change of enthalpy and ultimately a certain pressure rise is defined as

$$P_W = \dot{m} \cdot \Delta h_t = \dot{m} \cdot (h_{t,3} - h_{t,2}) = \dot{m} \cdot c_p (T_{t,3} - T_{t,2}) \ , \tag{2.41}$$

where $\dot{m}$ is the mass flow rate through the compressor

$$\dot{m} = \rho U_n A \ , \tag{2.42}$$

and $U_n = u_1 n_1 + u_2 n_2 + u_3 n_3$ is the contravariant velocity, i.e. the velocity normal to the cross section $A$. Hence, the compressor work per massflow is equal to the change in the specific enthalpy of the flow from the entrance to the exit of the compressor. Relating the real compressor work necessary to obtain the desired total pressure rise to the ideal case provides the isentropic efficiency

$$\eta_{\text{is}} = \frac{\Delta h_{t,\text{is}}}{\Delta h_t} = \frac{T_{t,3,\text{is}} - T_{t,2}}{T_{t,3} - T_{t,2}} \ . \tag{2.43}$$

Combining Eqn. (2.39) with Eqn. (2.43) finally yields

$$\eta_{\text{is}} = \frac{\pi^{\frac{\gamma-1}{\gamma}} T_{t,2} - T_{t,2}}{T_{t,3} - T_{t,2}} \ , \tag{2.44}$$

where $\pi$ is the total pressure ratio of the compressor, i.e.

$$\pi = \frac{p_{t,3}}{p_{t,2}} \ . \tag{2.45}$$

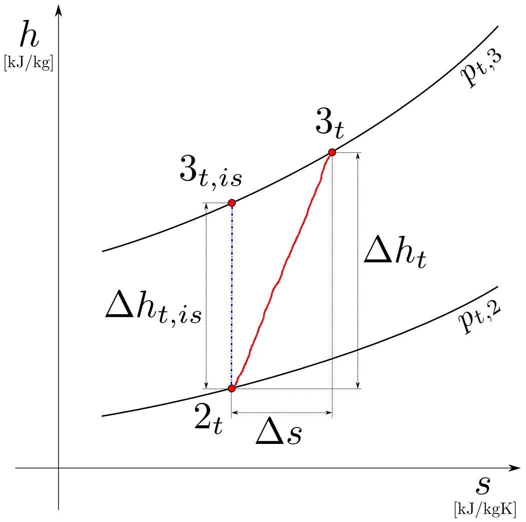

Figure 2.1: Isentropic compression process vs. actual compression process.

As it can be seen above, total flow quantities play an important role in evaluating the performance of a compressor or turbine. For flows at higher Mach numbers (typically $M > 0.3$) the Bernoulli principle, which in a simplified form states that the total pressure is the sum of the static pressure and a dynamic component, is invalid. Instead, total conditions can be obtained through the following relations. Considering adiabatic flow, the static and total enthalpy are related through

$$h_t = h + \frac{|\mathbf{U}|^2}{2} \ , \tag{2.46}$$

where again $|\mathbf{U}|^2 = u_1^2 + u_2^2 + u_3^2$. From Eqn.(2.40), the total and static temperature are related likewise,

$$T_t = T + \frac{|\mathbf{U}|^2}{2c_p} \ , \tag{2.47}$$

or can be expressed as a function of Mach number only,

$$\frac{T_t}{T} = 1 + \frac{\gamma - 1}{2} M^2 \ . \tag{2.48}$$

The local Mach number $M$ is defined as the ratio of the speed of the flow to the speed of sound in the surrounding medium, i.e.

$$M = \frac{|\mathbf{U}|}{c} \ , \tag{2.49}$$

where the speed of sound $c$ may be calculated from

$$c = \sqrt{\gamma R T} \ . \tag{2.50}$$

Considering isentropic flow, i.e. assuming that the flow can be brought to rest isentropically, total conditions for other flow quantities such as pressure and density can be defined through the isentropic relations and take a form similar to Eqn. (2.48). Substituting the absolute velocity $\mathbf{U}$ with the relative velocity $\mathbf{V}$ and using the relative Mach number

$$M_{\mathrm{rel}} = \frac{|\mathbf{V}|}{c} \ , \tag{2.51}$$

where $|\mathbf{V}| = \sqrt{v_1^2 + v_2^2 + v_3^2}$, yields the corresponding total quantities in a rotating frame of reference.

Mach number contour plots are often used to visualize the computed flow solution of a numerical simulation. In case of inviscid flow, e.g. if the Euler equations are considered, the Mach number distribution on a blade surface can provide valuable information to evaluate the flow field. However, for viscous flow, where the physical boundary conditions require the flow velocity to be zero on a solid wall, plotting the actual Mach number on a blade surface does not provide any useful information. Thus, the so-called isentropic Mach number is usually displayed instead. The isentropic Mach number is the Mach number that would be obtained on the blade surface if no losses would occur in the flow. It can be computed from the isentropic flow relations using the formula

$$M_{\mathrm{is}} = \sqrt{\left[\left(\frac{p_{t,f}}{p}\right)^{\frac{\gamma-1}{\gamma}} - 1\right] \frac{2}{\gamma - 1}} \ , \tag{2.52}$$

where $p_{t,f}$ is the total pressure in the freestream outside of the boundary layer. Using the relative total pressure, the isentropic Mach number in a rotating frame of reference is obtained.

# Chapter 3

# Flow Solver

This chapter describes the main features of the turbomachinery flow solver developed within this research. First, a non-dimensional form of the field equations, presented in section 2.2, is derived in section 3.1. While the RANS equations are solved in a Cartesian Coordinate system, it is more convenient and natural to apply boundary conditions in a turbomachinery environment using cylindrical coordinates. Section 3.2 discusses the mathematical relations between the two different coordinate systems. Finally, section 3.3 presents the numerical discretization and solution methodology utilized to solve the non-dimensionalized governing equations. Both, the spatial and temporal discretization schemes are discussed, the different types of discrete boundary conditions are presented, and the various convergence acceleration techniques utilized to speed up the convergence to steady-state are described.

## 3.1 Non-dimensionalized Navier-Stokes Equations

To reduce round-off errors due to the limited precision of computers, all physical quantities are normalized and non-dimensionalized such that they have the same order of magnitude. Generally, arbitrary reference quantities can be used to non-dimensionalize the governing equations. For external flow problems free-stream quantities are typically utilized. However, far-field conditions do not exist in turbomachinery flows, thus, different scaling parameters have to be employed. In this work, the following characteristic variables are used: The chord length $L$ [m] of the most upstream blade row at midspan, the inlet total temperature $T_{t,\mathrm{in}}$ [K], the inlet total pressure $p_{t,\mathrm{in}}$ [N/m$^2$], the specific gas constant $R$ [J/(kgK)] of the fluid, and the inlet viscosity $\mu_{t,\mathrm{in}}$ [m$^2$/s] calculated from the inlet total temperature. Utilizing these scales leads to the following dimensionless quantities, denoted with the superscript $^*$,

$$\begin{aligned}
x_i^* &= \frac{x_i}{L}\ , \quad t^* = \frac{\sqrt{RT_{t,\mathrm{in}}}}{L}t\ , \quad p^* = \frac{p}{p_{t,\mathrm{in}}}\ , \quad T^* = \frac{T}{T_{t,\mathrm{in}}}\ , \quad \rho^* = \frac{RT_{t,\mathrm{in}}}{p_{t,\mathrm{in}}}\rho\ , \\
u_i^* &= \frac{u_i}{\sqrt{RT_{t,\mathrm{in}}}}\ , \quad c^* = \frac{c}{\sqrt{RT_{t,\mathrm{in}}}}\ , \quad e^* = \frac{e}{RT_{t,\mathrm{in}}}\ , \quad E^* = \frac{E}{RT_{t,\mathrm{in}}}\ , \\
\mu^* &= \frac{\mu}{\mu_{t,\mathrm{in}}}\ , \quad k{=}\frac{k}{\mu_{t,\mathrm{in}}L}\ , \quad \Omega_i^* = \frac{L}{\sqrt{RT_{t,\mathrm{in}}}}\Omega_i\ , \quad R^* = 1\ , \quad c_p^* = \frac{\gamma}{\gamma-1}\ .
\end{aligned} \tag{3.1}$$

Substituting the dimensionless variables (3.1) into Eqns. (2.9)-(2.15) yields

$$\frac{p_{t,\mathrm{in}}}{\sqrt{RT_{t,\mathrm{in}}}L}\frac{\partial \mathbf{W}^*}{\partial t^*} + \frac{p_{t,\mathrm{in}}}{\sqrt{RT_{t,\mathrm{in}}}L}\frac{\partial \mathbf{F}_i^*}{\partial x_i^*} - \frac{\mu_{t,\mathrm{in}}}{L^2}\frac{\partial \mathbf{F}_{\mathbf{v}i}^*}{\partial x_i^*} = \frac{p_{t,\mathrm{in}}}{\sqrt{RT_{t,\mathrm{in}}}L}\mathbf{Q}^*\ , \tag{3.2}$$

where $\mathbf{W}^*$, $\mathbf{F}_i^*$, $\mathbf{F}_{\mathbf{v}i}^*$, and $\mathbf{Q}^*$ are the state vector, the inviscid and viscous flux vectors, and the source term in non-dimensional form, respectively. Multiplying Eqn. (3.2) with $\sqrt{RT_{t,\mathrm{in}}}L/p_{t,\mathrm{in}}$ provides

$$\frac{\partial \mathbf{W}^*}{\partial t^*} + \frac{\partial \mathbf{F}_i^*}{\partial x_i^*} - \frac{\mu_{t,\mathrm{in}}\sqrt{RT_{t,\mathrm{in}}}}{p_{t,\mathrm{in}}L}\frac{\partial \mathbf{F}_{\mathbf{v}i}^*}{\partial x_i^*} = \mathbf{Q}^*\ . \tag{3.3}$$

Equation (3.3) represents a set of differential equations which is identical to that of the dimensional Navier-Stokes equations. The only difference is an additional term which acts as a scaling factor for the viscous fluxes and is a function of inlet quantities only. The scaling factor can be further transformed using the equation of state, the definition of the Reynolds number $Re = \rho U_n L/\mu$, and the isentropic relations. After some algebra the final form of the non-dimensionalized Navier-Stokes equations is obtained,

$$\frac{\partial \mathbf{W}^*}{\partial t^*} + \frac{\partial \mathbf{F}_i^*}{\partial x_i^*} - \frac{\gamma M_\mathrm{in}}{Re_\mathrm{in}}\frac{\mu_{t,\mathrm{in}}}{\mu_\mathrm{in}}\left(1 + \frac{\gamma-1}{2}M_\mathrm{in}^2\right)^{\frac{1+\gamma}{2(1-\gamma)}}\frac{\partial \mathbf{F}_{\mathbf{v}i}^*}{\partial x_i^*} = \mathbf{Q}^*\ . \tag{3.4}$$

For the sake of brevity, the remainder of this paper will drop the asterisks and reference is henceforth made to non-dimensional variables only.

## 3.2 Coordinate Transformation

The three-dimensional non-dimensionalized Navier-Stokes equations are solved in a Cartesian coordinate system. However, in a turbomachinery flow environment it is more convenient to apply certain boundary conditions, cf. subsection 3.3.3, using cylindrical coordinates. Therefore, at these computational boundaries the variables are transferred between the two different coordinate systems according to their mathematical relation. With reference to Figure 3.1, where both coordinate systems are right-handed if thought of as $(x_1, x_2, x_3)$ and $(x_1, \varphi, r)$, the following relations apply,

$$x_2 = r\cos\varphi \ , \tag{3.5}$$

$$x_3 = r\sin\varphi \ , \tag{3.6}$$

$$r = \sqrt{x_2^2 + x_3^2} \ , \tag{3.7}$$

$$\varphi = \arctan\left(\frac{x_3}{x_2}\right) \ . \tag{3.8}$$

Any vector quantity can be transferred from a Cartesian coordinate formulation to

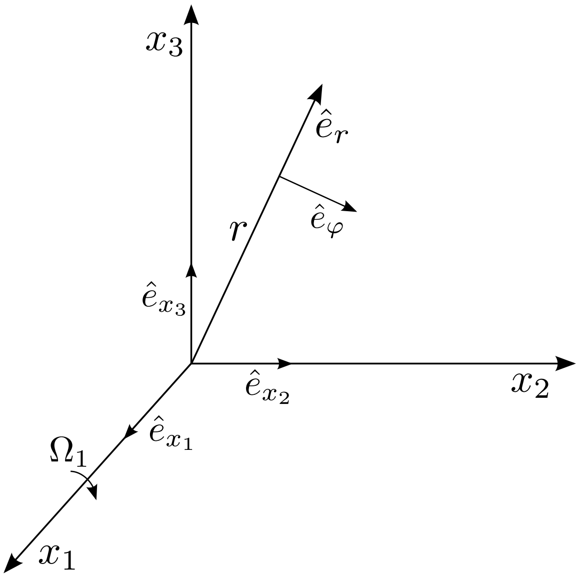

Figure 3.1: Cartesian and cylindrical coordinate systems employed within the flow solver.

the cylindrical coordinate system and vice versa through

$$\hat{e}_{\varphi} = -\hat{e}_{x_2}\sin\varphi + \hat{e}_{x_3}\cos\varphi \ , \tag{3.9}$$

$$\hat{e}_{r} = \hat{e}_{x_2}\cos\varphi + \hat{e}_{x_3}\sin\varphi \ , \tag{3.10}$$

$$\hat{e}_{x_2} = \hat{e}_{r}\cos\varphi - \hat{e}_{x_\varphi}\sin\varphi \ , \tag{3.11}$$

$$\hat{e}_{x_3} = \hat{e}_{r}\sin\varphi + \hat{e}_{x_\varphi}\cos\varphi \ . \tag{3.12}$$

Moreover, in cylindrical coordinates the absolute velocity $\mathbf{U} = [u_{x_1}, u_\varphi, u_r]^T$ is related to the relative velocity $\mathbf{V} = [v_{x_1}, v_\varphi, v_r]^T$ through

$$v_{x_1} = u_{x_1} \ , \tag{3.13}$$

$$v_{\varphi} = u_{\varphi} + \Omega_1 r \ , \tag{3.14}$$

$$v_{r} = u_{r} \ . \tag{3.15}$$

The two inflow angles $\alpha_\varphi$ and $\alpha_r$ prescribed at an inlet boundary, cf. subsection 3.3.3, are defined in Figure 3.2. With respect to the cylindrical coordinate system the following relations apply to any vector,

$$u_{x_1} = |\mathbf{U}|\cos\alpha_r\cos\alpha_\varphi \ , \tag{3.16}$$

$$u_{\varphi} = |\mathbf{U}|\cos\alpha_r\sin\alpha_\varphi \ , \tag{3.17}$$

$$u_{r} = |\mathbf{U}|\sin\alpha_r \ , \tag{3.18}$$

where again $|\mathbf{U}| = \sqrt{u_1^2 + u_2^2 + u_3^2}$,

## 3.3 Numerical Discretization

Except for some simplified flow problems, it generally is not possible to solve the Navier-Stokes equations analytically due to the non-linear character of the coupled system of partial differential equations. Instead the solution usually has to be approximated in an iterative manner. A vast number of solution strategies exist; most of which follow a similar approach. First, the physical domain is discretized by divid-

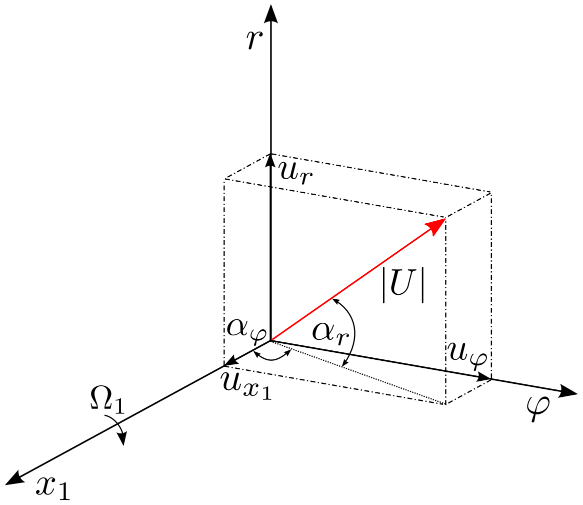

Figure 3.2: Prescribed inflow angles.

ing it into a number of geometrical elements. The discrete elements form a grid and the dependant variables of the governing equations are then calculated iteratively at discrete grid locations to approximate the continuous flow solution. Grid generation has become an important field of research since the accuracy of a flow solution substantially depends on the quality of the underlying grid. A large number of different grid generation strategies exist and the choice of the type of grid is usually based on the complexity of the geometry and the physical character of the flow problem under consideration. For example, the type of geometrical elements can differ; in two-dimensional problems typically quadrilaterals or triangles are chosen and three-dimensional grids are often built of tetrahedra or hexahedra. In addition, the spatial arrangement of the individual elements and the choice of indexing between these elements can vary, which leads to structured and unstructured grids. Furthermore, the nature of the flow usually determines certain grid properties such as local resolution levels. Viscous flow problems require highly-resolved grids close to the surface of the airfoil so that the boundary layer and its interaction with shocks can be resolved accurately. This typically leads to grids with large aspect ratios, which may pose a challenge for the flow solver to fully-converge the field equations. In an inviscid flow environment, large gradients do not occur in the vicinity of the surface of the airfoil, except for shock waves. Consequently, a more uniform and regular grid is sufficient to provide accurate numerical approximations. A more detailed discussion on grid generation techniques, including important quality requirements, can be found in [48]. After a grid has been generated, the governing equations are discretized. Most numerical methods developed for the solution of the Navier-Stokes equations employ the so-called method of lines. This approach allows for a separate discretization of the temporal and spatial derivatives, i.e. numerical approximations of different levels of accuracy can be used in space and time. Spatial discretization schemes can be divided into three main categories: finite-difference, finite-volume, and finite-element methods. Temporal discretization schemes are typically classified into explicit and implicit time integration methods. The choice of the right time-stepping scheme also depends on the nature of the flow problem. While unsteady flows require a time-accurate simulation, in steady-state calculations time accuracy is not a concern and the solution accuracy is only determined by the approximation order of the spatial discretization, which in turn opens up the possibility to utilize certain techniques to speed up the convergence process. For more detailed information on the various numerical discretization schemes, the reader should refer to [41] and [49].

In this research, curvilinear, structured multi-block grids are used to discretize the physical domain under consideration. The linear address space and straight-forward indexing of structured grids greatly simplifies the evaluation of flux gradients and boundary conditions and, thus, are preferred over unstructured grids. The multi-block approach reduces the challenge of generating high-quality grids for complex geometries. Following this strategy the physical space is first divided into a number of topologically simpler parts which are then meshed individually. A parallelized multi-block solver is used to solve the three-dimensional compressible RANS equations on the point-to-point matched multi-block grid system. Message Passing Interface (MPI) standard is used for communication between processors during the flow calculations. In accordance with the method of lines, Eqn. (2.9) can be rewritten in semi-discrete form as

$$\mathcal{V}\frac{\partial \mathbf{W}}{\partial t} + \mathbf{R}(\mathbf{W}) = 0 \ , \tag{3.19}$$

which is now a system of coupled, ordinary differential equations in time. $\mathcal{V}$ is the volume and $\mathbf{R}(\mathbf{W})$ is the residual containing the convective and dissipative flux gradients, the source term as well as all boundary conditions. Based on Eqn. (3.19) the flow governing equations are spatially discretized in a cell-centered finite volume method framework by using a second-order central-difference scheme for the convective and viscous flux gradients coupled with a blended first- and third-order artificial dissipation scheme [50]. Time integration is achieved by using a five-stage modified Runge-Kutta method. Multigrid and local time-stepping techniques as well as implicit residual smoothing are employed to speed up the convergence of the solution process to steady-state. The next subsections describe the numerical discretization schemes, the artificial dissipation scheme, and the convergence acceleration techniques used in this work in more detail.

### 3.3.1 Spatial Discretization

This subsection deals with the discretization of the spatial derivatives of the governing equations, i.e. with the numerical approximation of the residual term $\mathbf{R}(\mathbf{W})$ introduced in Eqn. (3.19). It describes the finite-volume methodology utilized in this work, the numerical approximation of the individual flux terms, and the discretization of the source term.

#### Finite-Volume Technique

The finite-volume method is preferred over the classical finite-difference formulation because it allows for an arbitrary grid. The method preserves the conservation laws at a discrete level and, thus, ensures the global conservation of mass, momentum, and energy. Following the finite-volume approach, hexahedral control volumes are defined based on the grid in order to evaluate the flux gradients and the source term. This flow solver employs a cell-centered discretization scheme, hence, the control volumes coincide with the cells of the grid. All flow variables are stored at the cell centers and the source term is calculated using these quantities. The flux gradients, however, are evaluated at the six faces of each hexahedra. Since the flow variables are not readily available at the flux faces, either the flow quantities or the fluxes have to be interpolated to the cell boundaries.

When using a body-fitted, structured grid, it is useful to introduce a mapping function between the Cartesian coordinate system $(x_1, x_2, x_3)$ and the curvilinear coordinates $(\xi_1, \xi_2, \xi_3)$ describing the grid. Figure **??** illustrates the control volume of a specific cell $(i,j,k)$ in physical and computational space; the $\xi_1$-, $\xi_2$-, and $\xi_3$-coordinates correspond to the index-directions $i$, $j$, and $k$, respectively. The red point is located at the centroid of the cell, while the six blue points represent the six flux faces. The transformation from the physical to the computational space is then defined by the metrics

$$\mathcal{K}_{nm} = \left[\frac{\partial x_n}{\partial \xi_m}\right] \ , \quad \mathcal{J} = \det(\mathcal{K}) \ , \quad \mathcal{K}^{-1}_{nm} = \left[\frac{\partial \xi_n}{\partial x_m}\right] \ , \tag{3.20}$$

for $n, m = 1, 2, 3$. Applying the mapping to the Navier-Stokes equations provides the field equations in computational space, where they read as

$$\frac{\partial(\mathcal{J}\mathbf{W})}{\partial t} + \frac{\partial \boldsymbol{\mathcal{F}}_n}{\partial \xi_n} - \frac{\partial \boldsymbol{\mathcal{F}}_{\mathbf{v}n}}{\partial \xi_n} = \mathcal{J}\mathbf{Q} \quad \text{in } \mathcal{D} \ . \tag{3.21}$$

The inviscid and viscous flux gradients are now defined with respect to the computational cell faces by

$$\boldsymbol{\mathcal{F}}_n = S_{nm}\mathbf{F}_m \qquad \text{and} \qquad \boldsymbol{\mathcal{F}}_{\mathbf{v}n} = S_{nm}\mathbf{F}_{\mathbf{v}m} \ , \tag{3.22}$$

where $S_{nm} = \mathcal{J}\mathcal{K}^{-1}_{nm}$ represents the projection of the $\xi_n$ cell face along the $x_m$ axis. The face vector normals are obtained from

$$n_{S_{nm}} = \frac{S_{nm}}{|S_{nm}|} \qquad \text{with} \qquad |S_{nm}| = \sqrt{S^2_{n1} + S^2_{n2} + S^2_{n3}} \ . \tag{3.23}$$

Applying the same metric transformation to the semi-discrete form of the Navier-Stokes equations, cf. Eqn. (3.19), the ordinary differential equation can be re-written for a specific cell $(i,j,k)$ as

$$\frac{\partial(\mathcal{J}\mathbf{W})_{i,j,k}}{\partial t} + \boldsymbol{\mathcal{R}}(\mathbf{W})_{i,j,k} = 0 \ , \tag{3.24}$$

where all terms comprised in the residual $\boldsymbol{\mathcal{R}}(\mathbf{W})$ are now defined in the computational space. Treating the source term separately,

$$\boldsymbol{\mathcal{R}}(\mathbf{W})_{i,j,k} = \hat{\boldsymbol{\mathcal{R}}}(\mathbf{W})_{i,j,k} + (\mathcal{J}\mathbf{Q})_{i,j,k} \ , \tag{3.25}$$

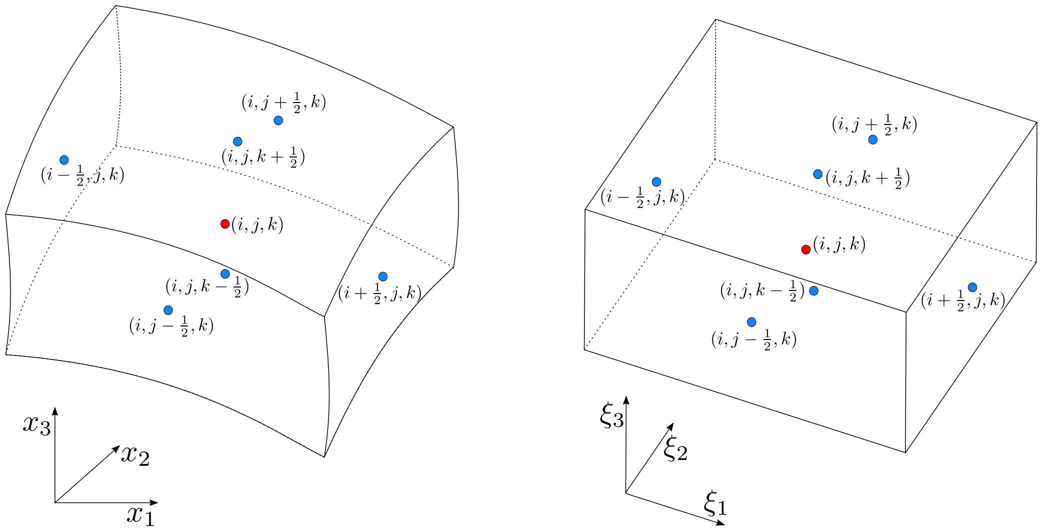

Figure 3.3: Control volume for cell $(i, j, k)$ in the physical domain (left) and the computational domain (right).

and expanding the remaining residual terms into its individual flux contributions yields

$$\hat{\boldsymbol{\mathcal{R}}}(\mathbf{W})_{i,j,k} = \frac{\partial \boldsymbol{\mathcal{F}}_1}{\partial \xi_1} + \frac{\partial \boldsymbol{\mathcal{F}}_2}{\partial \xi_2} + \frac{\partial \boldsymbol{\mathcal{F}}_3}{\partial \xi_3} - \frac{\partial \boldsymbol{\mathcal{F}}_{\mathbf{v}1}}{\partial \xi_1} - \frac{\partial \boldsymbol{\mathcal{F}}_{\mathbf{v}2}}{\partial \xi_2} - \frac{\partial \boldsymbol{\mathcal{F}}_{\mathbf{v}3}}{\partial \xi_3} \ , \tag{3.26}$$

where each partial derivative accounts for the net flux into one of the three computational directions. Using a second-order central-difference scheme to discretize both the convective and viscous flux gradients, e.g.

$$\frac{\partial \boldsymbol{\mathcal{F}}_1}{\partial \xi_1} = \mathbf{f}_{i+\frac{1}{2},j,k} - \mathbf{f}_{i-\frac{1}{2},j,k} \ , \tag{3.27}$$

where the $\pm\frac{1}{2}$ notation indicates that the quantity is defined at a flux face, provides the discrete form

$$\begin{aligned} \hat{\boldsymbol{\mathcal{R}}}(\mathbf{W})_{i,j,k} = \ & \mathbf{f}_{i+\frac{1}{2},j,k} - \mathbf{f}_{i-\frac{1}{2},j,k} + \mathbf{f}_{i,j+\frac{1}{2},k} - \mathbf{f}_{i,j-\frac{1}{2},k} + \mathbf{f}_{i,j,k+\frac{1}{2}} - \mathbf{f}_{i,j,k-\frac{1}{2}} \\ & - \mathbf{f_v}_{i+\frac{1}{2},j,k} + \mathbf{f_v}_{i-\frac{1}{2},j,k} - \mathbf{f_v}_{i,j+\frac{1}{2},k} + \mathbf{f_v}_{i,j-\frac{1}{2},k} - \mathbf{f_v}_{i,j,k+\frac{1}{2}} + \mathbf{f_v}_{i,j,k-\frac{1}{2}} \ , \end{aligned} \tag{3.28}$$

with $\mathbf{f}$ and $\mathbf{f_v}$ being the convective and viscous fluxes at the six faces of cell $(i, j, k)$.

Second-order accurate central-difference schemes allow for an odd-even decoupling of the numerical solution and are prone to generate oscillations and overshoots at shock discontinuities. For stability reasons it is therefore necessary to add additional dissipative terms to the numerical scheme to damp these high frequency errors. Thus, Eqn. (3.28) is expanded to

$$\begin{aligned}\hat{\boldsymbol{\mathcal{R}}}(\mathbf{W})_{i,j,k} = \quad & \mathbf{f}_{i+\frac{1}{2},j,k} - \mathbf{f}_{i-\frac{1}{2},j,k} + \mathbf{f}_{i,j+\frac{1}{2},k} - \mathbf{f}_{i,j-\frac{1}{2},k} + \mathbf{f}_{i,j,k+\frac{1}{2}} - \mathbf{f}_{i,j,k-\frac{1}{2}} \\ & - \mathbf{f}_{\mathbf{v}_{i+\frac{1}{2},j,k}} + \mathbf{f}_{\mathbf{v}_{i-\frac{1}{2},j,k}} - \mathbf{f}_{\mathbf{v}_{i,j+\frac{1}{2},k}} + \mathbf{f}_{\mathbf{v}_{i,j-\frac{1}{2},k}} - \mathbf{f}_{\mathbf{v}_{i,j,k+\frac{1}{2}}} + \mathbf{f}_{\mathbf{v}_{i,j,k-\frac{1}{2}}} \\ & - \mathbf{f}_{\mathbf{d}_{i+\frac{1}{2},j,k}} + \mathbf{f}_{\mathbf{d}_{i-\frac{1}{2},j,k}} - \mathbf{f}_{\mathbf{d}_{i,j+\frac{1}{2},k}} + \mathbf{f}_{\mathbf{d}_{i,j-\frac{1}{2},k}} - \mathbf{f}_{\mathbf{d}_{i,j,k+\frac{1}{2}}} + \mathbf{f}_{\mathbf{d}_{i,j,k-\frac{1}{2}}} \ .\end{aligned} \tag{3.29}$$

where $\mathbf{f_d}$ represents the added artificial dissipation term. A more detailed discussion of the concept of artificial dissipation and the scheme employed within this work is presented in subsection 3.3.2.

### Discretization of the Convective Fluxes

The cell-centered flow variables are viewed as cell averages and arithmetic averaging is used to approximate the convective fluxes $\mathbf{f}$ at the cell boundaries. Thus, the convective flux at a face has equal contributions from the two cells sharing the face,

$$\mathbf{f}_{i+\frac{1}{2},j,k} = \frac{1}{2}\left(\mathbf{f}_{i+1,j,k} + \mathbf{f}_{i,j,k}\right) \ . \tag{3.30}$$

The fluxes $i-\frac{1}{2}$, $j \pm \frac{1}{2}$, and $k \pm \frac{1}{2}$ are calculated accordingly. This produces a three-point stencil for each flux gradient and a seven-point stencil for the entire convective flux contribution to the residual of a particular cell, cf. Figure 3.4. In particular, the residual $\boldsymbol{\mathcal{R}}(\mathbf{W})_{i,j,k}$ is a function of the convective fluxes calculated at the cells $(i,j,k)$, $(i \pm 1, j, k)$, $(i, j \pm 1, k)$, and $(i, j, k \pm 1)$. In Eqn. (3.30) both fluxes, $\mathbf{f}_{i+1,j,k}$ and $\mathbf{f}_{i,j,k}$, are evaluated using the metrics of the flux face they have in common, i.e.

$$\mathbf{f}_{i\pm\frac{i}{2},j,k} = \frac{1}{2}\left(S_{1m_{i\pm\frac{1}{2},j,k}}\mathbf{F}_{m_{i\pm1,j,k}} + S_{1m_{i\pm\frac{1}{2},j,k}}\mathbf{F}_{m_{i,j,k}}\right) \ , \tag{3.31}$$

$$\mathbf{f}_{i,j\pm\frac{i}{2},k} = \frac{1}{2}\left(S_{2m_{i,j\pm\frac{1}{2},k}}\mathbf{F}_{m_{i,j\pm1,k}} + S_{2m_{i,j\pm\frac{1}{2},k}}\mathbf{F}_{m_{i,j,k}}\right) \ , \tag{3.32}$$

$$\mathbf{f}_{i,j,k\pm\frac{i}{2}} = \frac{1}{2}\left(S_{3m_{i,j,k\pm\frac{1}{2}}}\mathbf{F}_{m_{i,j,k\pm1}} + S_{3m_{i,j,k\pm\frac{1}{2}}}\mathbf{F}_{m_{i,j,k}}\right) \ , \tag{3.33}$$

where summation over $m = 1, 2, 3$ is implied by a repeated index $m$.

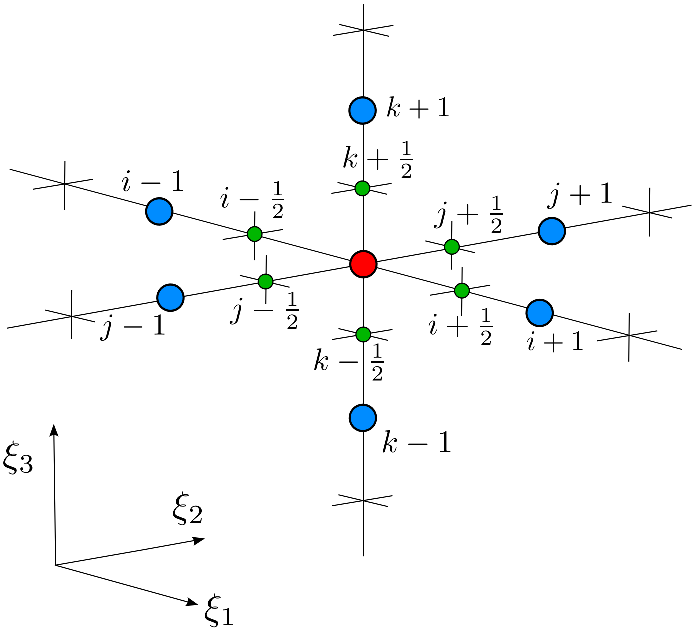

Figure 3.4: Stencil of the spatial discretization scheme for the convective flux gradient.

### Discretization of the Viscous Fluxes

The numerical evaluation of the viscous fluxes $\mathbf{f_v}$ at the cell boundaries is a more complex task due to the necessity of approximating the partial derivatives of the velocities appearing in the viscous stress tensor, cf. Eqn. (2.14), and the temperature gradients introduced to the energy equation by Fourier's law of heat conduction, cf. Eqn. (2.6). A second-order accurate discretization of the viscous fluxes is obtained from the following procedure: As defined in Eqn. (3.22), the viscous boundary flux $\mathbf{f_v}$ in computational space is calculated from the physical boundary flux $\mathbf{F_v}$ through the transformation

$$\mathbf{f_v}_{i\pm\frac{i}{2},j,k} = S_{1m_{i\pm\frac{1}{2},j,k}} \mathbf{F_v}_{m_{i\pm\frac{i}{2},j,k}} \ , \tag{3.34}$$

$$\mathbf{f_v}_{i,j\pm\frac{i}{2},k} = S_{2m_{i,j\pm\frac{1}{2},k}} \mathbf{F_v}_{m_{i,j\pm\frac{i}{2},k}} \ , \tag{3.35}$$

$$\mathbf{f_v}_{i,j,k\pm\frac{i}{2}} = S_{3m_{i,j,k\pm\frac{1}{2}}} \mathbf{F_v}_{m_{i,j,k\pm\frac{i}{2}}} \ , \tag{3.36}$$

where $m$ again implies summation over $m = 1, 2, 3$. The physical flux itself has to be interpolated to the face center. It is computed at the mid-point of a cell face as the arithmetic average of the flux quantities defined at the four vertexes of the face. Thus, the flux vector at the face $(i+\frac{1}{2}, j, k)$ expands to

$$\begin{aligned}\mathbf{F_v}_{m_{i+\frac{1}{2},j,k}} = \frac{1}{4}( & \mathbf{F_v}_{m_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}} + \mathbf{F_v}_{m_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}} \\ & + \mathbf{F_v}_{m_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}} + \mathbf{F_v}_{m_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}} ) \ . \end{aligned} \tag{3.37}$$

From Eqns. (2.10) and (2.15) it follows that the viscous flux at the vertex $(i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2})$ can be written as

$$\mathbf{F_v}m_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \left\{ \begin{array}{c} 0 \\ \tau_{mn}\hat{\hat{\delta}}_{n1} \\ \tau_{mn}\hat{\hat{\delta}}_{n2} \\ \tau_{mn}\hat{\hat{\delta}}_{n3} \\ u_n\tau_{mn} + k\frac{\delta T}{\delta x_m} \end{array} \right\}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \tag{3.38}$$

with the viscous stress tensor

$$\tau_{mn_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}} = \mu_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \left[ \frac{\partial u_m}{\partial x_n} + \frac{\partial u_n}{\partial x_m} - \frac{2}{3}\frac{\partial u_l}{\partial x_l}\hat{\delta}_{mn} \right]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} . \tag{3.39}$$

In Eqns. (3.38) and (3.39) the $(i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2})$ subscript is placed outside the brackets to avoid repeating the subscript for every term within the brackets. To approximate the stress tensor and the heat flux components of the viscous flux at the cell vertex, auxiliary control volumes are formed for each vertex by joining the cell centers of the eight cells that share the same vertex. As illustrated in Figure 3.5, the auxiliary control volume for the vertex $(i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2})$ is consequently specified by the eight cell centers $(i,j,k)$, $(i+1,j,k)$, $(i,j+1,k)$, $(i+1,j+1,k)$, $(i,j,k+1)$, $(i+1,j,k+1)$, $(i,j+1,k+1)$, and $(i+1,j+1,k+1)$. The velocity components $u$, the coefficient of thermal conductivity $k$, and the temperature $T$ at a cell vertex are obtained from averaging the cell center values of the eight cells forming the auxiliary control volume. The first viscosity coefficient $\mu$, which is a combination of the laminar and turbulent viscosity, i.e.

$$\mu_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = (\mu_l + \mu_t)_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \ , \tag{3.40}$$

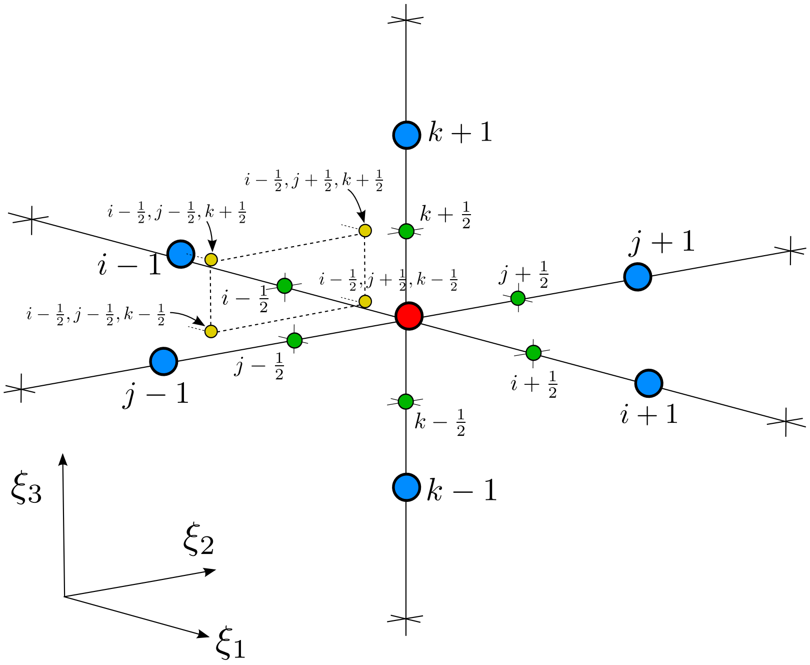
$k+1$
$i-\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2}$
$i-\frac{1}{2}, j-\frac{1}{2}, k+\frac{1}{2}$
$k+\frac{1}{2}$
$j+1$
$i-1$
$i-\frac{1}{2}, j+\frac{1}{2}, k-\frac{1}{2}$
$j+\frac{1}{2}$
$i-\frac{1}{2}$
$i-\frac{1}{2}, j-\frac{1}{2}, k-\frac{1}{2}$
$j-\frac{1}{2}$
$i+\frac{1}{2}$
$j-1$
$i+1$
$k-\frac{1}{2}$
$\xi_3$
$k-1$
$\xi_2$
$\xi_1$

Figure 3.5: Stencil of the spatial discretization scheme for the viscous flux gradient.

is estimated by applying the same averaging technique. Hence, any of these quantities is calculated as

$$\begin{aligned} T_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \frac{1}{8}( & \quad T_{i,j,k} \quad + T_{i+1,j,k} \quad + T_{i,j+1,k} \quad + T_{i+1,j+1,k} \\ & + T_{i,j,k+1} + T_{i+1,j,k+1} + T_{i,j+1,k+1} + T_{i+1,j+1,k+1}) \ . \end{aligned} \tag{3.41}$$

The velocity and temperature gradients at the vertexes are calculated by a transformation to the computational coordinates. For example, the gradient of the velocity component $u_m$ into the Cartesian direction $x_n$ is defined as

$$\left[\frac{\partial u_m}{\partial x_n}\right]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \frac{1}{J_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}} \left[\frac{\partial \hat{u}_{m_{1n}}}{\partial \xi_1} + \frac{\partial \hat{u}_{m_{2n}}}{\partial \xi_2} + \frac{\partial \hat{u}_{m_{3n}}}{\partial \xi_3}\right]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \ , \tag{3.42}$$

where $J$ is the volume of the auxiliary control volume, which is approximated by averaging the volumes of the eight cells that share the vertex using Eqn. (3.41). The velocity gradient in the computational domain, for example $\partial \hat{u}_{m_{1n}}/\partial \xi_1$, can be calculated by taking an average of the velocity differences in the $\xi_1$-direction, i.e.

$$\begin{aligned}\left[\frac{\partial \hat{u}_{m_{1n}}}{\partial \xi_1}\right]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} &= \frac{\left(\hat{u}_{m_{1n,i+1,j,k}} - \hat{u}_{m_{1n,i,j,k}}\right) + \left(\hat{u}_{m_{1n,i+1,j+1,k+1}} - \hat{u}_{m_{1n,i,j+1,k+1}}\right)}{4} \\ &+ \frac{\left(\hat{u}_{m_{1n,i+1,j+1,k}} - \hat{u}_{m_{1n,i,j+1,k}}\right) + \left(\hat{u}_{m_{1n,i+1,j,k+1}} - \hat{u}_{m_{1n,i,j,k+1}}\right)}{4} \ , \end{aligned} \tag{3.43}$$

where $\hat{u}$ is the velocity multiplied with the cell metrics interpolated to the cell center by averaging the two corresponding face metrics

$$\hat{u}_{m_{1n,i,j,k}} = \frac{S_{1n_{i+\frac{1}{2},j,k}} + S_{1n_{i-\frac{1}{2},j,k}}}{2} \ u_{m_{i,j,k}} \ . \tag{3.44}$$

The formulation described to compute the viscous flux, guaranties conservation and produces a second-order accurate algorithm. The stencil of the numerical scheme extends over twenty-seven cells.

#### Discretization of the Source Term

While both, the convective and viscous flux gradients are evaluated at the cell faces, the source term $\mathbf{Q}$ is calculated at the cell center. Since the flow variables are readily available at the centroid of the control volume, its evaluation is straight forward. From Eqn. (3.25) it follows that the residual $\boldsymbol{\mathcal{R}}(\mathbf{W})_{i,j,k}$ has the source term contribution $(J\mathbf{Q})_{i,j,k}$, where $J$ is the volume of the cell $(i, j, k)$.

### 3.3.2 Artificial Dissipation

Considering a simplified one-dimensional inviscid flow problem such as depicted in Figure 3.6, where a flux only occurs in the $i$-direction, the Cartesian Coordinate system coincides with the computational coordinates, and the face vectors are assumed to be all of size one, then the second-order central-difference scheme employed to approximate the convective fluxes, cf. Eqns. (3.38) and (3.39), reads as

$$\mathbf{f}_{i+\frac{1}{2}} = \frac{1}{2}\left(\mathbf{F}_{i+1} + \mathbf{F}_i\right) \ , \tag{3.45}$$

and provides the residual contribution

$$\mathbf{R}(\mathbf{W})_i = \mathbf{f}_{i+\frac{1}{2}} - \mathbf{f}_{i-\frac{1}{2}} = \frac{1}{2}\left(\mathbf{F}_{i+1} + \mathbf{F}_i - \mathbf{F}_i - \mathbf{F}_{i-1}\right) \ , \tag{3.46}$$

which reduces to

$$\mathbf{R}(\mathbf{W})_i = \mathbf{f}_{i+\frac{1}{2}} - \mathbf{f}_{i-\frac{1}{2}} = \frac{1}{2}\left(\mathbf{F}_{i+1} - \mathbf{F}_{i-1}\right) \ . \tag{3.47}$$

Equation (3.47) illustrates that the residual for cell $i$ is only a function of the flow variables in the two neighbouring cells $(i+1)$ and $(i-1)$, but is independent of its current solution at cell $i$. A solution that takes on the value 1 at all odd cells and the value $-1$ at all even cells, would satisfy the flow problem in the steady-state limit. This phenomenon is called odd-even decoupling. Oscillations resulting from discontinuities in the flow solution or introduced by round-off errors are not damped out and would make the numerical scheme unstable if it is not further modified.

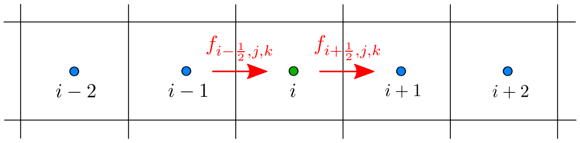

Figure 3.6: Uniform grid for the simplified one-dimensional flow problem.

The idea behind the concept of artificial dissipation is to add additional difference operators to the numerical scheme which couple the equation, suppress the tendency for odd-even point oscillations, and increase the dissipation in the vicinity of discontinuities to limit overshoots near shock waves. The artificial dissipation scheme used within this research is a blend of adaptive second- and fourth-order differences which result from the sum of additional first- and third-order fluxes. This scalar dissipation scheme was first introduced by Jameson, Schmidt and Turkel [50], hence the name JST dissipation scheme, and is defined as

$$\mathbf{F_d} = \epsilon^{(2)} \Delta x^3 \frac{\lambda}{p} \left| \frac{\partial^2 p}{\partial x^2} \right| \frac{\partial \mathbf{W}}{\partial x} - \epsilon^{(4)} \Delta x^3 \lambda \frac{\partial^3 \mathbf{W}}{\partial x^3} \ , \tag{3.48}$$

where $\epsilon^{(2)}$ and $\epsilon^{(4)}$ are adjustable constants and $\lambda$ is the spectral radius of the flux Jacobian matrices. The discrete formulation of the artificial dissipation flux may be written as

$$\mathbf{f}_{\mathbf{d}_{i+\frac{1}{2},j,k}} = \mathbf{f}^{(2)}_{\mathbf{d}_{i+\frac{1}{2},j,k}} - \mathbf{f}^{(4)}_{\mathbf{d}_{i+\frac{1}{2},j,k}} \ , \tag{3.49}$$

with

$$\mathbf{f}^{(2)}_{\mathbf{d}_{i+\frac{1}{2},j,k}} = \nu^{(2)}_{i+\frac{1}{2},j,k}\Lambda_{i+\frac{1}{2},j,k}(\mathbf{W}_{i+1,j,k} - \mathbf{W}_{i,j,k}) \ , \tag{3.50}$$

$$\mathbf{f}^{(4)}_{\mathbf{d}_{i+\frac{1}{2},j,k}} = \nu^{(4)}_{i+\frac{1}{2},j,k}\Lambda_{i+\frac{1}{2},j,k}(\mathbf{W}_{i+2,j,k} - 3\mathbf{W}_{i+1,j,k} + 3\mathbf{W}_{i,j,k} - \mathbf{W}_{i-1,j,k}) \ . \tag{3.51}$$

The dissipative fluxes at the cell faces $i-\frac{1}{2}$, $j\pm\frac{1}{2}$, and $k\pm\frac{1}{2}$ are calculated accordingly.

In Eqn. (3.50), $\mathbf{d}^{(2)}$ is a first-order flux term that is proportional to the normalized second difference of pressure. The term $\nu^{(2)}$ acts as a pressure sensor,

$$\nu^{(2)}_{i+\frac{1}{2},j,k} = \epsilon^{(2)}\max(\sigma_{i,j,k}, \sigma_{i+1,j,k}) \ , \tag{3.52}$$

where

$$\sigma_{i,j,k} = \frac{|p_{i+1,j,k} - p_{i,j,k} + p_{i-1,j,k}|}{p_{i+1,j,k} + p_{i,j,k} + p_{i-1,j,k}} \ , \tag{3.53}$$

and $\epsilon^{(2)} = 1$. In smooth flow regions the pressure gradient term becomes negligible, thus, $\mathbf{d}^{(2)}$ is inactive and the artificial dissipation is reduced to the lowest possible level; only the third-order flux will be active and provide background dissipation. In flow regions with large pressure gradients however, e.g. close to a shock wave, the first-order flux is activated. The introduction of the first-order dissipative flux term adds an upwind biasing to the second-order central-difference scheme and allows for the capture of pressure discontinuities.

In Eqns. (3.50) and (3.51), $\Lambda_{i+\frac{1}{2},j,k}$ is a scaled spectral radius term interpolated to the cell face through

$$\Lambda_{i+\frac{1}{2},j,k} = \frac{1}{2}\left(\tilde{\lambda}_{\xi_1,i+1,j,k} + \tilde{\lambda}_{\xi_1,i,j,k}\right) \ , \tag{3.54}$$

where $\tilde{\lambda}_{\xi}$ is the scaled spectral radius of the flux Jacobian in the $\xi_1$-direction of the computational domain. The spectral radii are scaled to improve the convergence properties of the numerical scheme on stretched meshes for viscous flows, cf. [51, 52], using the formula

$$\tilde{\lambda}_{\xi_1} = \left[1 + \left(\frac{\tilde{\lambda}_{\xi_2}}{\tilde{\lambda}_{\xi_1}}\right)^{\sigma} + \left(\frac{\tilde{\lambda}_{\xi_3}}{\tilde{\lambda}_{\xi_1}}\right)^{\sigma}\right] \tilde{\lambda}_{\xi_1} \ . \tag{3.55}$$

with $\sigma = 2/3$. The spectral radius is the largest eigenvalue of the flux Jacobian matrix and is given by

$$\tilde{\lambda}_{\xi_m} = |S_{m1}v_1 + S_{m2}v_2 + S_{m3}v_3| + c(S_{m1}^2 + S_{m2}^2 + S_{m3}^2) \ , \tag{3.56}$$

where $v_l$ is the relative velocity and $c$ is the speed of sound. Multiplying the first-order dissipative flux with the spectral radius helps controlling the magnitude of the dissipative term, which is especially important in viscous shear layers. The magnitude of the dissipative term has to be considerably smaller than the magnitude of the viscous flux gradient, otherwise the dissipative terms would corrupt the quality of the flow solution. In the boundary layer of viscous flows the flow velocity tangential to the surface is slowly reduced to zero; the spectral radius scaling ensures that the artificial dissipation is reduced simultaneously.

In Eqn. (3.51), another pressure-based sensor $\nu^{(4)}$ is used to switch off the fourth-order differences at shocks and in flow regions of high pressure gradients, where they would introduce oscillations. The sensor is defined as

$$\nu^{(4)}_{i+\frac{1}{2},j,k} = \max(0, \epsilon^{(4)} - \nu^{(2)}_{i+\frac{1}{2},j,k}) \ , \tag{3.57}$$

where $\epsilon^{(4)}$ typically is set to 1/32, which guarantees that $\epsilon^{(4)} - \nu^{(2)} < 0$ in the vicinity of flow discontinuities.

The stencil of this compact artificial dissipation scheme spans over two cells in each direction, therefore, requiring information from a total of thirteen cells.

### 3.3.3 Discrete Boundary Conditions

Any numerical simulation can only consider a certain part of the real physical domain. Truncating the domain leads to artificial boundaries in addition to the natural solid wall boundaries. At these boundaries the numerical scheme must be modified and numerical boundary conditions have to be applied such that the flow physics are reflected adequately. The numerical treatment of the boundary conditions requires a particular care, since an improper implementation can result in an inaccurate simulation of the real system. Furthermore, the convergence properties of the numerical scheme as well as its stability can be negatively influenced.

The following types of boundary conditions are generally encountered in the numerical solution of the Euler and the Navier-Stokes equations in a turbomachinery environment:

- solid wall boundaries,
- inflow and outflow boundaries,
- periodic boundaries,
- boundaries between blocks,
- rotor-stator interfaces.

The numerical treatment of these boundary conditions, which is based on the concept of halo cells, is described in the following. For clarity, the subscript “D” shall indicate a domain cell, the subscript “H” shall represent a halo cell, and the subscript “B” shall determine the domain boundary which coincides with the cell face shared by a domain and halo cell.

### Solid Wall Boundary Conditions

For inviscid flow, the physical model requires the velocity at a wall to be tangent to the slope of the wall. This corresponds to a zero flux through the wall. In a viscous flow environment, the no-injection and no-slip conditions require the tangent velocity components to be zero as well. Thus, the convective terms in all flux directions vanish at solid walls. In computational space, assuming that the wall surface lies in the $\xi_1\xi_2$-plane, cf. Figure 3.7, this leads to the following physical requirements:

$$S_{n1}u_{1,\mathrm{B}} + S_{n2}u_{2,\mathrm{B}} + S_{n3}u_{3,\mathrm{B}} = 0 \tag{3.58}$$

at a stationary wall and

$$S_{n1}u_{1,\mathrm{B}} + S_{n2}(u_{2,\mathrm{B}} + \Omega_1 x_3) + S_{n3}(u_{3,\mathrm{B}} - \Omega_1 x_2) = 0 \tag{3.59}$$

at a rotating wall, where $n = 3$ in case of inviscid flow and $n = 1, 2, 3$ for viscous flow. Accordingly, at a wall boundary the convective flux reduces to

$$\boldsymbol{\mathcal{F}}_{n,\mathrm{B}} = \left\{ \begin{array}{ll} \rho(S_{n2}\Omega_1 x_3 - S_{n3}\Omega_1 x_2) & \\ \rho(S_{n2}\Omega_1 x_3 - S_{n3}\Omega_1 x_2)u_1 & +S_{n1}p \\ \rho(S_{n2}\Omega_1 x_3 - S_{n3}\Omega_1 x_2)u_2 & +S_{n2}p \\ \rho(S_{n2}\Omega_1 x_3 - S_{n3}\Omega_1 x_2)u_3 & +S_{n3}p \\ \rho(S_{n2}\Omega_1 x_3 - S_{n3}\Omega_1 x_2)E & \end{array} \right\}_{\mathrm{B}} \tag{3.60}$$

in case of a stationary surface. If the domain is non-rotating and an absolute frame of reference formulation is employed, i.e. $\Omega_1 = 0$, then Eqn. (3.60) further simplifies to

$$\boldsymbol{\mathcal{F}}_{n,\mathrm{B}} = \left\{ \begin{array}{c} 0 \\ S_{n1}p \\ S_{n2}p \\ S_{n3}p \\ 0 \end{array} \right\}_{\mathrm{B}} , \tag{3.61}$$

i.e. only the pressure is required at the wall. At a rotating wall the convective flux is given by

$$\boldsymbol{\mathcal{F}}_{n,\mathrm{B}} = \left\{ \begin{array}{c} 0 \\ S_{n1}p \\ S_{n2}p \\ S_{n3}p \\ p(S_{n1}u_1 + S_{n2}u_2 + S_{n3}u_3) \end{array} \right\}_{\mathrm{B}} . \tag{3.62}$$

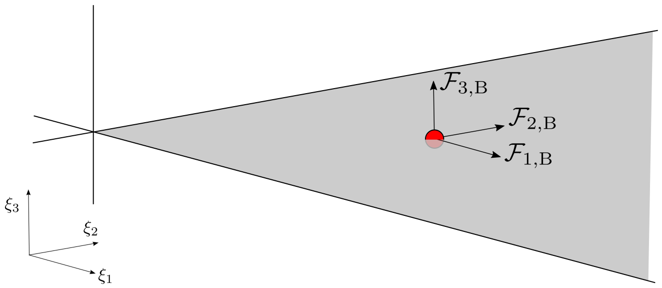
$\mathcal{F}_{3,\mathrm{B}}$
$\mathcal{F}_{2,\mathrm{B}}$
$\mathcal{F}_{1,\mathrm{B}}$
$\xi_3$
$\xi_2$
$\xi_1$

Figure 3.7: Solid wall boundary in computational space in the $\xi_1\xi_2$-plane.

Again, in Eqns. (3.60)-(3.62) $n = 3$ in case of an inviscid flow environment and $n = 1, 2, 3$ for viscous flow. All three wall boundary types are encountered in this turbomachinery flow solver. For instance, in a rotating rotor domain the convective flux at the stationary casing is described by Eqn. (3.60), at the boundary of a rotating rotor blade it is given by Eqn. (3.62), and at the stationary stator blade, which is located within the non-rotating stator domain, the convective flux is described by Eqn. (3.61).

Halo cells are introduced and the halo state vectors are specified such that the physical correct boundary fluxes at a wall are obtained directly from the cell-centered scheme utilized within this flow solver. Since the scheme estimates the convective fluxes at a cell face by averaging the fluxes calculated from the state solutions of the two cells sharing the boundary, the halo states

$$w_{1,\mathrm{H}} = w_{1,\mathrm{D}} \ , \tag{3.63a}$$

$$w_{2,\mathrm{H}} = w_{2,\mathrm{D}} - 2n_1[w_{2,\mathrm{D}}n_1 + w_{3,\mathrm{D}}n_2 + w_{4,\mathrm{D}}n_3 + w_{1,\mathrm{D}}\Omega_1(x_3n_2 - x_2n_3)] \ , \tag{3.63b}$$

$$w_{3,\mathrm{H}} = w_{3,\mathrm{D}} - 2n_2[w_{2,\mathrm{D}}n_1 + w_{3,\mathrm{D}}n_2 + w_{4,\mathrm{D}}n_3 + w_{1,\mathrm{D}}\Omega_1(x_3n_2 - x_2n_3)] \ , \tag{3.63c}$$

$$w_{4,\mathrm{H}} = w_{4,\mathrm{D}} - 2n_3[w_{2,\mathrm{D}}n_1 + w_{3,\mathrm{D}}n_2 + w_{4,\mathrm{D}}n_3 + w_{1,\mathrm{D}}\Omega_1(x_3n_2 - x_2n_3)] \ , \tag{3.63d}$$

$$w_{5,\mathrm{H}} = w_{5,\mathrm{D}} \ , \tag{3.63e}$$

provide the correct boundary fluxes for inviscid flow, while the halo states

$$w_{1,\mathrm{H}} = \phantom{-}w_{1,\mathrm{D}} \ , \tag{3.64a}$$

$$w_{2,\mathrm{H}} = -w_{2,\mathrm{D}} \ , \tag{3.64b}$$

$$w_{3,\mathrm{H}} = -w_{3,\mathrm{D}} - 2w_{1,\mathrm{D}}\Omega_1 x_3 \ , \tag{3.64c}$$

$$w_{4,\mathrm{H}} = -w_{4,\mathrm{D}} + 2w_{1,\mathrm{D}}\Omega_1 x_2 \ , \tag{3.64d}$$

$$w_{5,\mathrm{H}} = \phantom{-}w_{5,\mathrm{D}} \ , \tag{3.64e}$$

satisfy the physical model for viscous flow. The boundary states $\mathbf{W}_\mathrm{B}$ can be obtained by averaging the conserved variables stored in the halo cells $\mathbf{W}_\mathrm{H}$ and the states of the domain cells next to a wall $\mathbf{W}_\mathrm{D}$,

$$\mathbf{W}_\mathrm{B} = \frac{1}{2}\left(\mathbf{W}_\mathrm{D} + \mathbf{W}_\mathrm{H}\right) \ , \tag{3.65}$$

which yields

$$w_{1,\mathrm{B}} = w_{1,\mathrm{D}} \ , \tag{3.66a}$$

$$w_{2,\mathrm{B}} = w_{2,\mathrm{D}} - n_1[w_{2,\mathrm{D}}n_1 + w_{3,\mathrm{D}}n_2 + w_{4,\mathrm{D}}n_3 + w_{1,\mathrm{D}}\Omega_1(x_3n_2 - x_2n_3)] \ , \tag{3.66b}$$

$$w_{3,\mathrm{B}} = w_{3,\mathrm{D}} - n_2[w_{2,\mathrm{D}}n_1 + w_{3,\mathrm{D}}n_2 + w_{4,\mathrm{D}}n_3 + w_{1,\mathrm{D}}\Omega_1(x_3n_2 - x_2n_3)] \ , \tag{3.66c}$$

$$w_{4,\mathrm{B}} = w_{4,\mathrm{D}} - n_3[w_{2,\mathrm{D}}n_1 + w_{3,\mathrm{D}}n_2 + w_{4,\mathrm{D}}n_3 + w_{1,\mathrm{D}}\Omega_1(x_3n_2 - x_2n_3)] \ , \tag{3.66d}$$

$$w_{5,\mathrm{B}} = w_{5,\mathrm{D}} \ , \tag{3.66e}$$

for inviscid flow and

$$w_{1,\mathrm{B}} = \quad w_{1,\mathrm{D}} \ , \tag{3.67a}$$

$$w_{2,\mathrm{B}} = \quad 0 \ , \tag{3.67b}$$

$$w_{3,\mathrm{B}} = -w_{1,\mathrm{D}}\Omega_1 x_3 \ , \tag{3.67c}$$

$$w_{4,\mathrm{B}} = \quad w_{1,\mathrm{D}}\Omega_1 x_2 \ , \tag{3.67d}$$

$$w_{5,\mathrm{B}} = \quad w_{5,\mathrm{D}} \ , \tag{3.67e}$$

for viscous flow.

To complete the solid wall boundary conditions, the pressure has to be determined at a wall. As for the density and total energy, a simple zeroth-order extrapolation through the wall is used within this work. Thus, the pressure of the domain cell is copied to the halo cell which corresponds to a zero pressure gradient at the wall,

$$p_\mathrm{B} = p_\mathrm{H} = p_\mathrm{D} \ . \tag{3.68}$$

### Inlet and Outlet Boundary Conditions

At the inlet and outlet of the computational domain, characteristic-based boundary conditions using Riemann invariants are imposed. The method extrapolates the outgoing waves based on interior information and calculates incoming waves using the prescribed freestream conditions. To simplify the treatment of inlet and outlet boundaries, the inlet and outlet planes are defined such that they coincide with the $\xi_2\xi_3$-plane. Thus, the axial velocity component $u_{1,\mathrm{B}}$ is normal to the domain boundaries, while the components $u_{2,\mathrm{B}}$ and $u_{3,\mathrm{B}}$ are tangent to the inlet and outlet planes.

Assuming subsonic inflow, there are four incoming characteristics and one outgoing characteristic at the inlet. Hence, four quantities have to be prescribed. These are the total pressure $p_{t,\mathrm{in}}$, the total temperature $T_{t,\mathrm{in}}$ and the circumferential and radial inflow angles $\alpha_\varphi$ and $\alpha_r$. The outgoing Riemann invariant

$$Ri = u_{1,\mathrm{D}} - \frac{2c_\mathrm{D}}{\gamma - 1} \ , \tag{3.69}$$

is used to express the velocity normal to the boundary $u_{1,\mathrm{in}}$ as a function of the speed of sound at the boundary and some interior quantities,

$$u_{1,\mathrm{in}} = \frac{2c_\mathrm{in}}{\gamma - 1} + Ri \ . \tag{3.70}$$

Combining Eqns. (2.48)-(2.50) with Eqns. (3.16)-(3.18), the total temperature at the inlet can be written as

$$T_{t,\mathrm{in}} = \frac{c_\mathrm{in}^2}{\gamma R}\left[1 + \frac{\gamma - 1}{2}\frac{u_{1,\mathrm{in}}^2}{c_\mathrm{in}^2}\left(1 + \tan^2\alpha_\varphi + \frac{\tan^2\alpha_r}{\cos^2\alpha_\varphi}\right)\right] \ . \tag{3.71}$$

From Eqn. (3.70) follows

$$u_{1,\mathrm{in}}^2 = \frac{4c_\mathrm{in}^2}{(\gamma-1)^2} + \frac{4c_\mathrm{in}}{(\gamma-1)}Ri + Ri^2 \ , \tag{3.72}$$

which is used with Eqn. (3.71) to form

$$T_{t,\mathrm{in}} = \frac{c_\mathrm{in}^2}{\gamma R} + \frac{\gamma - 1}{2\gamma R}\left(\frac{4c_\mathrm{in}^2}{(\gamma-1)^2} + \frac{4c_\mathrm{in}}{(\gamma-1)}Ri + Ri^2\right)\alpha_x \ , \tag{3.73}$$

where

$$\alpha_x = 1 + \tan^2\alpha_\varphi + \frac{\tan^2\alpha_r}{\cos^2\alpha_\varphi} \ . \tag{3.74}$$

Rearranging Eqn. (3.73) yields a quadratic equation which solves for the speed of sound at the inlet boundary,

$$c_\mathrm{in} = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \ , \tag{3.75}$$

with

$$A = \frac{2}{\gamma - 1} + \frac{1}{\alpha_x} \ , \qquad B = 2Ri \ , \qquad \text{and} \qquad C = \frac{\gamma - 1}{2} Ri^2 - \frac{\gamma R T_{t,\text{in}}}{\alpha_x} \ . \tag{3.76}$$

The remaining boundary quantities can then be determined from the prescribed inlet quantities, the isentropic relations, and the equation of state. In particular,

$$\begin{aligned} T_{\text{in}} &= \frac{c_{\text{in}}^2}{\gamma R} \ , & p_{\text{in}} &= p_{t,\text{in}} \left( \frac{T_{\text{in}}}{T_{t,\text{in}}} \right)^{\frac{\gamma}{\gamma-1}} \ , \\ \rho_{\text{in}} &= \frac{p_{\text{in}}}{R T_{\text{in}}} \ , & |\mathbf{U}|_{\text{in}} &= \sqrt{2 c_p (T_{t,\text{in}} - T_{\text{in}})} \ . \end{aligned} \tag{3.77}$$

The inlet velocity can be decomposed into its individual components using Eqns. (3.11), (3.12), and (3.16)-(3.18). This completes the calculation of the primitive variables

$$\mathbf{W}_{\mathbf{P},\text{in}} = \left[ \rho, \ u_1, \ u_2, \ u_3, \ p \right]_{\text{in}}^T \ , \tag{3.78}$$

at the boundary. The conservative variable vector at the inlet boundary may now be obtained from the transformation

$$\mathbf{W}_{\text{in}} = \mathbf{M}_{\text{in}} \mathbf{W}_{\mathbf{P},\text{in}} \ , \tag{3.79}$$

where $\mathbf{M}_{\text{in}}$ is a transformation operator relating the flow variables to the primitive variables; cf. appendix A.1 for a detailed definition of the transformation matrix $\mathbf{M}$.

At a subsonic outlet boundary, there are four outgoing characteristics and one incoming characteristic. Hence, one quantity, namely the static pressure $p_{\text{out}}$, is specified. In case the outlet follows a stator row, the exit pressure is chosen to be constant across the entire outlet plane. However, if the blade row upstream the outlet is rotating, a radial equilibrium outlet boundary condition is applied. In this case, the specified exit pressure is only applied at the hub, i.e. at the position of minimum radius. The static exit pressure across the remaining outlet plane is calculated from the assumption that the radial velocity is negligible, so that the pressure gradient is given by

$$\frac{\partial p}{\partial r} = \frac{\rho u_\varphi^2}{r} \ . \tag{3.80}$$

Thus, the exit pressure is obtained from

$$p_{\text{out}} = p_{\text{out, Hub}} + \left( \frac{r}{r_{\text{Hub}}} - 1 \right) \rho_{\text{D}} u_{\varphi,\text{D}}^2 \ . \tag{3.81}$$

This outlet boundary condition satisfies the requirement that the outflow is in radial equilibrium. The entropy is then extrapolated from the interior domain, i.e.

$$s_{\text{out}} = \frac{p_{\text{out}}}{\rho_{\text{out}}^{\gamma}} = \frac{p_{\text{D}}}{\rho_{\text{D}}^{\gamma}} \ , \tag{3.82}$$

which provides the density at the boundary,

$$\rho_{\text{out}} = \rho_{\text{D}} \left( \frac{p_{\text{out}}}{p_{\text{D}}} \right)^{\frac{1}{\gamma}} \ . \tag{3.83}$$

Thereafter, the speed of sound can be determined from

$$c_{\text{out}} = \sqrt{\gamma \frac{p_{\text{out}}}{\rho_{\text{out}}}} \ . \tag{3.84}$$

Together with the information obtained from the outgoing Riemann invariant

$$Ri = u_{1,\text{D}} + \frac{2 c_{\text{D}}}{\gamma - 1} \ , \tag{3.85}$$

this allows for the calculation of the velocity normal to the outlet boundary

$$u_{1,\text{out}} = Ri - \frac{2 c_{\text{out}}}{\gamma - 1} \ . \tag{3.86}$$

The velocity components tangential to the outlet plane are extrapolated from the interior domain,

$$u_{2,\text{out}} = u_{2,\text{D}} \ , \qquad u_{3,\text{out}} = u_{3,\text{D}} \ , \tag{3.87}$$

while the temperature can be calculated as in Eqn. (3.77). Lastly, the outlet boundary states $\mathbf{W}_{\text{out}}$ are constructed from the the primitive variables $\mathbf{W}_{\mathbf{P},\text{out}}$ through the identical transformation as at the inlet boundary, i.e.

$$\mathbf{W}_{\text{out}} = \mathbf{M}_{\text{out}} \mathbf{W}_{\mathbf{P},\text{out}} \ , \tag{3.88}$$

where $\mathbf{M}_{\mathrm{out}}$ is defined as in Eqn. (A.3).

### Periodic Boundary Conditions

Flows through axial compressors and turbines are usually assumed to be periodic in the circumferential direction. This assumption greatly reduces the computational cost since in this case it is sufficient to only model one blade-to-blade section instead of the entire annulus.

The concept of halo cells simplifies the treatment of periodic boundaries. Scalar quantities, such as the density and pressure, are simply copied from the domain cell next to a periodic boundary into the halo cell of the other corresponding periodic boundary. The treatment of vector quantities, e.g. the velocity, depends on the type of periodicity. In case of translational periodicity the vector quantities can be copied just like the scalar quantities. In case of rotational periodicity the rotation of the coordinate system has to be accounted for. Figure 3.8 illustrates the circumferential periodicity typically encountered in turbomachinery flows as well as within this work. The state of the domain cell (red-colored) marked by the circle (or square) defines the accordingly marked boundary state (blue-colored). However, here the Cartesian velocity components $u_2$ and $u_3$ cannot be directly exchanged between two periodic boundaries. Instead, they are first transferred into the cylindrical components $u_\varphi$ and $u_r$ using Eqns. (3.9) and (3.10). Afterwards, the vector quantities are exchanged in their cylindrical form and are then transferred back into Cartesian components, cf. (3.11) and (3.12). This procedure yields the halo states

$$w_{1,\mathrm{H,II}} = w_{1,\mathrm{D,I}} \ , \tag{3.89a}$$

$$w_{2,\mathrm{H,II}} = w_{2,\mathrm{D,I}} \ , \tag{3.89b}$$

$$\begin{aligned} w_{3,\mathrm{H,II}} = & \left(w_{3,\mathrm{D,I}} \cos\varphi_{,\mathrm{I}} + w_{4,\mathrm{D,I}} \sin\varphi_{,\mathrm{I}}\right) \cos\varphi_{,\mathrm{II}} \\ & + \left(w_{3,\mathrm{D,I}} \sin\varphi_{,\mathrm{I}} - w_{4,\mathrm{D,I}} \cos\varphi_{,\mathrm{I}}\right) \sin\varphi_{,\mathrm{II}} \ , \end{aligned} \tag{3.89c}$$

$$\begin{aligned} w_{4,\mathrm{H,II}} = & \left(w_{3,\mathrm{D,I}} \cos\varphi_{,\mathrm{I}} + w_{4,\mathrm{D,I}} \sin\varphi_{,\mathrm{I}}\right) \sin\varphi_{,\mathrm{II}} \\ & - \left(w_{3,\mathrm{D,I}} \sin\varphi_{,\mathrm{I}} - w_{4,\mathrm{D,I}} \cos\varphi_{,\mathrm{I}}\right) \cos\varphi_{,\mathrm{II}} \ , \end{aligned} \tag{3.89d}$$

$$w_{5,\mathrm{H,II}} = w_{5,\mathrm{D,I}} \ , \tag{3.89e}$$

and ensures that the fluxes at two corresponding periodic boundaries are identical. The states at a periodic boundary may be obtained from Eqn. (3.65).

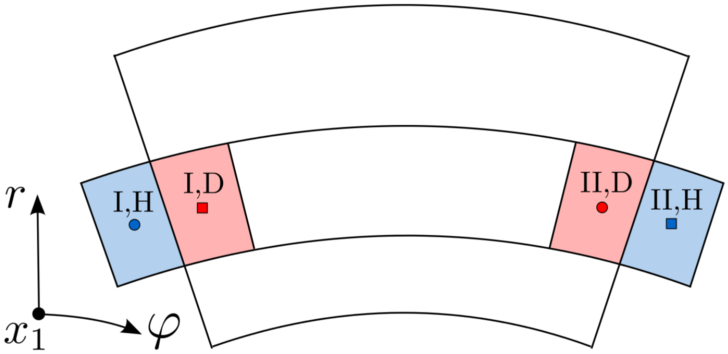

Figure 3.8: Periodicity in circumferential coordinate direction.

### Block-to-Block Boundaries

The multi-block approach greatly simplifies the generation of high-quality grids for complex geometries. Furthermore, the computational time can be reduced dramatically if the blocks are distributed to different processors and, thus, multiple processors are used to solve the governing equations in the computational domain. However, it also complicates the numerical solution process as information has to be exchanged between neighbouring blocks since the temporal development of the solution in a particular block depends on the solution in its neighbouring blocks.

In this work, the multi-block approach is implemented utilizing the concept of halo cells. Furthermore, the flow solver requires the grid lines of two adjacent blocks to match at a block-to-block interface. Every block is extended by two additional levels of cells at its six boundaries. The halo cells of a particular block possess the same geometric properties as the first two domain cells of the neighbouring block and thus two adjacent blocks overlap each other by two cell levels. The flow variables are then exchanged between two neighbouring blocks after every stage of the Runge-Kutta time-stepping scheme, cf. subsection 3.3.4, by copying the domain states of one block to the corresponding overlapping halo cells of the adjacent block. This boundary treatment allows the standard spatial discretization scheme to be extended until the block interfaces and guarantees that the boundary flux is equivalent to one that would be obtained on a single-block grid. In case two neighbouring blocks are solved on different processors, MPI standard is used for communication between the two processors.

### Rotor-Stator Interface

To allow for a concurrent rotor-stator and hence multi-row flow simulation, it is necessary to couple a rotational rotor domain with its neighbouring stationary stator domain during the solution process of the governing equations. In this flow solver, rotor-stator interfaces are treated based on Chima's steady averaging-plane approach for multistage turbomachinery flow analysis [53]. A mixing-plane interface is shown schematically in Figure 3.9 at a constant radial section, where the rotor and stator domains have been displaced axially for clarity. Similarly to the inflow and outflow boundaries, the orientation of the blade row interface is always chosen to be normal to the $x_1$-axis, i.e. only fluxes into the axial direction have to be considered at the mixing-plane interface. To further simplify the treatment, the boundary conditions are developed in a cylindrical coordinate system.

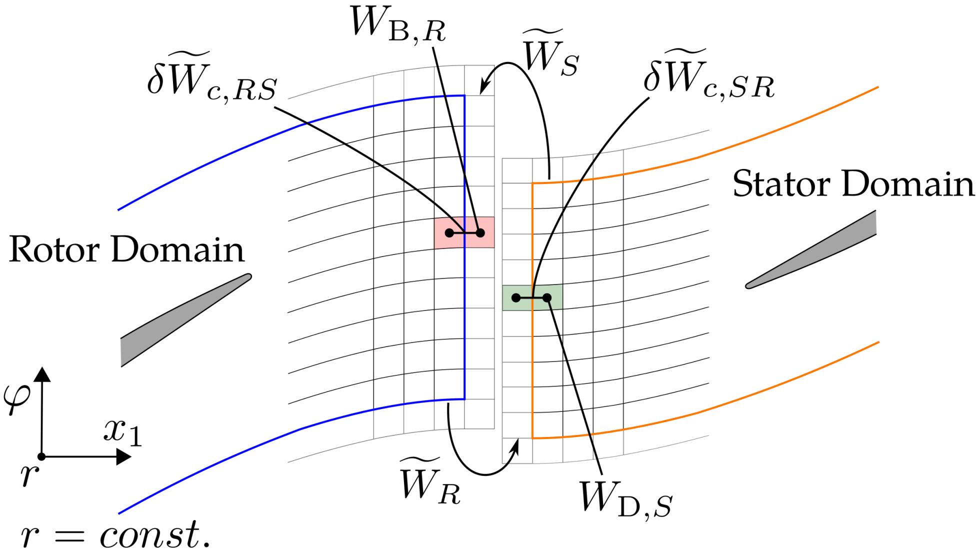

Figure 3.9: Schematic mixing-plane interface at constant radius.

There are two fundamental requirements on a rotor-stator interface; the conservation of the fluxes across the mixing-plane and a non-reflective behaviour at the domain interface to allow for a close spacing between adjacent blade rows. For any averaging-plane method, averaged flow properties must be transferred between the blade rows at the domain interfaces. Following Chima's approach, mixed-out circumferential averages of the primitive variables are calculated from all cells at a constant radius on both sides of the interface and are used as boundary values in the neighbouring domain. The differences between the local domain states next to the boundary and the boundary states are then treated as characteristic perturbations and Giles' 1D characteristic theory [54] is applied to drive the difference to zero.

The definition of mixed-out variables is based on the assumption that sufficiently far upstream or downstream of the boundary under consideration the flow field is circumferentially uniform and that the flux based on the uniform primitive variables is equal to the average flux at the boundary. Following Wyss et al. [55], the mixed-out primitive variables $\widetilde{\mathbf{W}}_{\mathbf{P}}$ are calculated as

$$\widetilde{p} = \frac{1}{\gamma+1}\left(\bar{F}_{x_1} + \sqrt{\bar{F}_{x_1}^2 + (\gamma^2-1)(\bar{F}_{x_1}^2 + \bar{F}_{\varphi}^2 + \bar{F}_r^2 - 2\bar{F}_1^2\bar{F}_5^2)}\right) , \tag{3.90}$$

$$\widetilde{u}_{x_1} = \frac{\bar{F}_{x_1} - \widetilde{p}}{\bar{F}_1} , \tag{3.91}$$

$$\widetilde{u}_{\varphi} = \frac{\bar{F}_{\varphi}}{\bar{F}_1} , \tag{3.92}$$

$$\widetilde{u}_{r} = \frac{\bar{F}_{r}}{\bar{F}_1} , \tag{3.93}$$

$$\widetilde{\rho} = \frac{\bar{F}_1}{\widetilde{u}_{x_1}} , \tag{3.94}$$

with the averaged fluxes

$$\bar{\mathbf{F}} = \frac{1}{l_P}\int_0^{l_P} \mathbf{F}\,\mathrm{d}A \qquad \text{at} \quad r = \text{const.} , \tag{3.95}$$

where $l_P$ is either the rotor or stator pitch area, $\mathrm{d}A$ is the face normal of each cell, and $r$ is the radius. Hence, the discrete version of the integral in Eqn. (3.95) represents a summation over all domain cells next to a rotor-stator interface at constant radius into the circumferential direction. The flux vector in Eqn. (3.95) is formulated in cylindrical coordinates and defined as

$$\mathbf{F} = \left\{ \begin{array}{ll} \rho u_{x_1} & \\ \rho u_{x_1} u_{x_1} & +p \\ \rho u_{x_1} u_{\varphi} & \\ \rho u_{x_1} u_r & \\ \rho u_{x_1} E & +p u_{x_1} \end{array} \right\} . \tag{3.96}$$

Based on the mixed-out primitive variables (3.90)-(3.94), averaged values of all other flow quantities can be defined. The mixed-out conservative variable vector $\widetilde{\mathbf{W}} = [\widetilde{w}_1, \widetilde{w}_{x_1}, \widetilde{w}_{\varphi}, \widetilde{w}_r, \widetilde{w}_5]^T$ is then copied into the adjacent mixing-plane domain where it serves as the boundary state. The average Mach number is then checked in the neighbouring domain to determine whether the flow is subsonic or supersonic and characteristic boundary conditions are utilized to correct the boundary states. Thus, the mixing-plane boundary condition may be written as

$$\mathbf{W}_{\mathrm{B},R} = \widetilde{\mathbf{W}}_S + \delta\widetilde{\mathbf{W}}_{\mathbf{c},RS} \ , \tag{3.97}$$

$$\mathbf{W}_{\mathrm{B},S} = \widetilde{\mathbf{W}}_R + \delta\widetilde{\mathbf{W}}_{\mathbf{c},SR} \ , \tag{3.98}$$

where the subscripts $R$ and $S$ indicate values obtained from or calculated for the rotor and stator domain, respectively. Giles' 1D characteristic theory provides the boundary state correction $\delta\widetilde{\mathbf{W}}_{\mathbf{c}}$. For example, the boundary correction for a rotor domain is calculated from

$$\delta\widetilde{\mathbf{W}}_{\mathbf{c},RS} = \widetilde{\mathbf{P}}\delta\mathbf{C}_{RS} \ , \tag{3.99}$$

where $\delta\mathbf{C} = [\delta c_1, \delta c_{x_1}, \delta c_{\varphi}, \delta c_r, c_5]^T$ is the vector of characteristic variables and $\widetilde{\mathbf{P}}$ is the right eigenvector matrix of the convective flux Jacobian which is obtained from diagonalizing the Jacobian system. For a more detailed discussion of the diagonalization procedure and the transformation between conservative and characteristic variables the reader should refer to appendix A.3. Here, the right eigenvector matrix is defined as specified in Eqn. (A.23), but built of mixed-out variables. For subsonic flow at an inlet boundary, i.e. the downstream domain of a rotor-stator interface, the four incoming characteristics $\delta c_1$, $\delta c_{x_1}$, $\delta c_{\varphi}$, and $\delta c_r$ are set to zero. In case of supersonic flow the fifth characteristic $\delta c_5$ is incoming and, hence, null as well. For subsonic flow at an exit boundary, i.e. the upstream domain of a rotor-stator interface, only the fifth characteristic $\delta c_5$ is incoming and thus is discarded, while for supersonic flow all characteristics are outgoing and have to be extrapolated from the interior domain. The outgoing characteristics can be determined from

$$\delta \mathbf{C}_{RS} = \widetilde{\mathbf{P}}^{-1} \delta \mathbf{W}_{RS} \ , \tag{3.100}$$

where $\widetilde{\mathbf{P}}^{-1}$ is the left eigenvector matrix of the convective flux Jacobian as specified in Eqn. (A.22) and again is constructed using mixed-out quantities. In Eqn. (3.100), $\delta \mathbf{W}_{RS}$ is the difference between the local domain value next to the mixing-plane interface and the boundary state, i.e.

$$\delta \mathbf{W}_{RS} = \left\{ \begin{array}{ll} \rho_{\mathrm{D},R} & -\widetilde{\rho}_S \\ \rho_{\mathrm{D},R} u_{x_1,D,R} & -\widetilde{\rho}_S \widetilde{u}_{x_1,S} \\ \rho_{\mathrm{D},R} u_{\varphi,D,R} & -\widetilde{\rho}_S \widetilde{u}_{\varphi,S} \\ \rho_{\mathrm{D},R} u_{r,D,R} & -\widetilde{\rho}_S \widetilde{u}_{r,S} \\ \rho_{\mathrm{D},R} E_{D,R} & -\widetilde{\rho}_S \widetilde{E}_S \end{array} \right\} \ . \tag{3.101}$$

The boundary values for the stator domain are obtained accordingly by substituting $R$ with $S$ and vice versa. During the convergence of the governing equations this procedure leads to equal mixed-out primitive variables on both sides of the mixing-plane interface, which guarantees the conservation of the circumstantially averaged fluxes.

### 3.3.4 Temporal Discretization

This subsection deals with the discretization of the temporal derivatives of the governing equations, i.e. with the numerical approximation of the term $\partial \mathbf{W} / \partial t$ described in Eqn. (3.19).

This work employs an explicit multistage time integration scheme and utilizes Jameson's modified Runge-Kutta approach, cf. [50]. Multistage schemes advance the solution in a number of so-called stages between the current time $n$ and the next time level $n+1$. Jameson's modified Runge-Kutta scheme reads as

$$\begin{aligned}
\mathbf{W}^{(0)} &= \mathbf{W}^{(n)} \\
\mathbf{W}^{(k)} &= \mathbf{W}^{(0)} - \alpha_k \Delta t \mathbf{R}(\mathbf{W}^{(k-1)}) \quad \text{for } k = 1, 2, \ldots, m \\
\mathbf{W}^{(n+1)} &= \mathbf{W}^{(m)} \ ,
\end{aligned} \tag{3.102}$$

where $m$ is the total number of stages and the residual $\mathbf{R}(\mathbf{W}^{(k-1)})$ is evaluated with the solution $\mathbf{W}^{(k-1)}$ of the previous stage. The residual at each stage of the scheme is defined as

$$\mathbf{R}^{(k)} = \mathbf{R}_{\mathbf{c}}^{(k)} + \mathbf{R}_{\mathbf{d}}^{(k)} \ , \tag{3.103}$$

with

$$\mathbf{R}_{\mathbf{c}}^{(k)} = \mathbf{R}_{\mathbf{c}}(\mathbf{W}^{(k)}) \ , \tag{3.104}$$

$$\mathbf{R}_{\mathbf{d}}^{(k)} = \beta_k \mathbf{R}_{\mathbf{d}}(\mathbf{W}^{(k)}) + (1-\beta_k)\mathbf{R}_{\mathbf{d}}^{(k-1)} \ , \tag{3.105}$$

where $\mathbf{R}_{\mathbf{c}}^{(k)}$ includes the residual contributions from the convective flux gradient and the source term, while $\mathbf{R}_{\mathbf{d}}^{(k)}$ is composed of the residual contributions from the viscous flux gradient and the artificial dissipation. From Eqns. (3.102) and (3.104) it can be seen that only the zeroth solution and the latest residual need to be stored, which reduces the memory requirements compared to the classical Runge-Kutta schemes. The stage coefficients $\alpha_k$ and the blending coefficients $\beta_k$ are optimized to maximize the time step and to improve the numerical stability for the selected spatial discretization method. Research has shown, cf. [52], that it is not necessary to evaluate the dissipative fluxes at each stage and that the computational cost can be reduced without comprising the stability of the time stepping scheme by only evaluating the dissipative operators at some stages. In this work, a five stage Runge-Kutta scheme is employed where the dissipative fluxes are evaluated at stages (1), (3), and (5). The coefficients for the scheme are

$$\alpha_1 = 0.25 \ , \quad \alpha_2 = 0.1667 \ , \quad \alpha_3 = 0.375 \ , \quad \alpha_4 = 0.5 \ , \quad \alpha_5 = 1.0 \ , \tag{3.106}$$

$$\beta_1 = 1.0 \ , \quad \beta_2 = 0.0 \ , \quad \beta_3 = 0.56 \ , \quad \beta_4 = 0.0 \ , \quad \beta_5 = 0.44 \ . \tag{3.107}$$

### 3.3.5 Convergence Acceleration

Despite the continuous growth in computational power, CFD simulations are still fairly time-consuming and can take up to several hours or even days. To minimize the amount of time needed to obtain a converged solution, many numerical tools have been developed in order to accelerate the solution of the governing equations to steady-state. In this subsection, the convergence acceleration techniques utilized within this work are briefly discussed.

**Local Time Stepping**

The convergence of a time integration method typically can be accelerated by increasing the time step. However, for every numerical scheme, especially if an explicit time integration technique is utilized, a maximum step size exists after which the scheme becomes unstable. The maximum local time step can be determined by the Courant-Friedrich-Lewy (CFL) condition. It states that the domain of dependence of the numerical method must at least contain the domain of dependence of the partial differential equation. This means, the time step of the numerical scheme should not be larger than the time required to transport information across the stencil of the spatial discretization scheme.

Considering an inviscid flow environment, the time step limit for a specific control volume $(i,j,k)$ can be estimated from the relation

$$\Delta t_{i,j,k} = \mathrm{CFL}\frac{\mathcal{V}_{i,j,k}}{(\tilde{\lambda}_{\xi_1} + \tilde{\lambda}_{\xi_2} + \tilde{\lambda}_{\xi_3})_{i,j,k}} \ , \tag{3.108}$$

where $\mathcal{V}$ is the volume and $\tilde{\lambda}_{\xi_m}$ are the spectral radii of the convective flux Jacobian matrices calculated from Eqn. (3.56). The maximum CFL number generally depends both on the selected time stepping scheme and the spatial discretization. Here it is a function of the stability region of the modified Runge-Kutta scheme, cf. subsection 3.3.4. From the CFL condition and Eqn. (3.108) it follows that the time step limit decreases with the local grid resolution. A conservative approach, which also produces a time-accurate solution, would be to calculate the smallest maximum time step in the entire computational domain and use this fixed time step for the time integration of the discretized governing equations in every control volume. For flow problems in which the physical domain can be spatially discretized using a grid with cells of similar size this approach is reasonable. However, for viscous flow problems, which require highly resolved and stretched grids close to a wall, the fixed time stepping approach becomes impractical since it would require an excessive amount of iterations. For stationary flow problems in which the transient solution is of no importance, the solution process to steady state can be strongly accelerated by integrating the discretized governing equation in time using the largest possible time step for each control volume instead. Employing a variable time stepping approach, the state vector in each cell is advanced using a local time step which only satisfies the stability limit for that cell. This approach greatly accelerates the convergence of the governing equations but leads to a temporally inaccurate transient solution. Since in this work only steady-state calculations are performed, this is irrelevant and the local time stepping technique can be applied.

In a viscous flow environment, the time step calculated from Eqn. (3.108) has to be augmented to include the maximum eigenvalues of the diffusion operator. The reader should refer to [56] and [52] for a detailed discussion of the maximum time step estimate for viscous flows.

### Implicit Residual Smoothing

While the local time stepping approach speeds up the time integration to steady state by increasing the time step, implicit residual smoothing accelerates the convergences by enabling the selection of a higher CFL number. The general idea is to replace the residual of a specific cell with a weighted average of the surrounding residuals. The residual smoothing technique adds an implicit character to the explicit Runge-Kutta scheme and improves the damping of high frequency error components of the residual. In three dimensions it can be formulated as

$$-\epsilon_i \mathbf{R}(\mathbf{W})^{*}_{i-1,j,k} + (1+2\epsilon_i)\mathbf{R}(\mathbf{W})^{*}_{i,j,k} - \epsilon_i \mathbf{R}(\mathbf{W})^{*}_{i+1,j,k} = \mathbf{R}(\mathbf{W})_{i,j,k}\ , \quad (3.109)$$

$$-\epsilon_j \mathbf{R}(\mathbf{W})^{**}_{i,j-1,k} + (1+2\epsilon_j)\mathbf{R}(\mathbf{W})^{**}_{i,j,k} - \epsilon_j \mathbf{R}(\mathbf{W})^{**}_{i,j+1,k} = \mathbf{R}(\mathbf{W})_{i,j,k}\ , \quad (3.110)$$

$$-\epsilon_k \mathbf{R}(\mathbf{W})^{***}_{i,j,k-1} + (1+2\epsilon_k)\mathbf{R}(\mathbf{W})^{***}_{i,j,k} - \epsilon_k \mathbf{R}(\mathbf{W})^{***}_{i,j,k+1} = \mathbf{R}(\mathbf{W})_{i,j,k}\ , \quad (3.111)$$

which represents a second-order central-difference operator. $\mathbf{R}(\mathbf{W})^*$, $\mathbf{R}(\mathbf{W})^{**}$, and $\mathbf{R}(\mathbf{W})^{***}$ denote the smoothed residuals in $i$-, $j$-, and $k$-directions, respectively. The implicit equation system (3.109)-(3.111) is solved for the smoothed residuals in each coordinate direction using a tridiagonal solver. The smoothing coefficients $\epsilon_i$, $\epsilon_j$, and $\epsilon_k$ are functions of the spectral radii of the convective flux Jacobian. A complete discussion of the stability character and overall benefit of this acceleration method is provided by Jameson and Baker in [57] and Turkel et al. in [58].

### Multigrid

The multigrid method is one of the most powerful and most effective convergence techniques. The basic idea behind the multigrid approach is to successively coarsen the grid, to advance the solution on these auxiliary grids, and to calculate a correction factor which can be used to drive the solution on the original grid faster to steady state. On the coarser auxiliary grids larger time steps can be employed, while at the same time the numerical effort is reduced. Information is exchanged more rapidly and travels faster through the computational domain. In consequence, disturbances are more quickly expelled through the outer boundary. Moreover, the low frequency errors on the fine grid become high-frequency errors on a coarser grid. Since most numerical schemes reduce high-frequency errors efficiently, but take a long time to damp low-frequency errors, the multigrid technique helps to reduce all error components quickly, which accelerates the convergence significantly. On structured grids, successively coarser grids can easily be generated by simply selecting every other grid point in each direction to obtain the next coarser level. This procedure results in the agglomeration of eight grid cells to form one coarser grid cell. The multigrid method is tributed to Brandt [59], who developed it for elliptic partial differential equations and Jameson [60], who introduced the concept of multigrid to the hyperbolic Euler equations. In the following, the major steps of the multigrid strategy are briefly explained.

Based on the latest flow solution $\mathbf{W}_h^{(n)}$, where $h$ denotes the fine grid, a new residual term $\mathbf{R}(\mathbf{W}_h^{(n)})_h$ can be evaluated. In a first step, both the fine grid solution and the residual are transferred to the coarse grid by

$$\mathbf{W}_H^{(0)} = \hat{\mathbf{I}}_h^H \mathbf{W}_h^{(n)} \qquad \text{and} \qquad \mathbf{R}(\mathbf{W}_h^{(n)})_H = \mathbf{I}_h^H \mathbf{R}(\mathbf{W}_h^{(n)})_h \ , \tag{3.112}$$

where $H$ denotes the coarser grid, $\hat{\mathbf{I}}_h^H$ is an appropriate interpolation operator for the fine grid solution, and $\mathbf{I}_h^H$ is the residual transfer operator. For the solution vector, the cell-centered scheme used within this work employs a volume-weighted averaging to the control volumes that will form the new cell on the coarser level, i.e.

$$\mathbf{W}_H^{(0)} = \frac{\sum \mathcal{V}_{i,j,k} \mathbf{W}_{h_{i,j,k}}^{(n)}}{\sum \mathcal{V}_{i,j,k}} \ , \tag{3.113}$$

where $\sum$ implies summation over the eight cells $(i,j,k)$, $(i+1,j,k)$, $(i,j+1,k)$, $(i+1,j+1,k)$, $(i,j,k+1)$, $(i+1,j,k+1)$, $(i,j+1,k+1)$, and $(i+1,j+1,k+1)$. To ensure that the low-frequency errors of the fine grid are smoothed on the coarse grid, a conservative transfer operator $\mathbf{I}_h^H$ has to be used for the residual interpolation as well. In case of a cell-centered scheme the coarse grid residual is obtained by simply adding up all the fine grid residuals that form the new coarse grid cell, i.e.

$$\mathbf{R}(\mathbf{W}_h^{(n)})_H = \sum \mathbf{R}(\mathbf{W}_h^{(n)})_h \ , \tag{3.114}$$

where $\sum$ implies summation over the same eight cells specified for Eqn. (3.113). To retain the order of accuracy of the fine grid solution on the coarse grid, a forcing function $\boldsymbol{\mathcal{Q}}$ has to be defined and added to the coarse grid residual. It is defined as the difference between the transferred residual and the residual calculated from the initial solution on the coarse grid,

$$\boldsymbol{\mathcal{Q}}_H = \mathbf{R}(\mathbf{W}_h^{(n)})_H - \mathbf{R}(\mathbf{W}_H^{(0)})_H \ , \tag{3.115}$$

and ensures that the solution vector on the coarse grid is primarily driven by the transferred fine grid residuals during the first modified Runge-Kutta step. Hence, the time stepping scheme on the coarse grid may be written as

$$
\begin{aligned}
\mathbf{W}_H^{(0)} &= \mathbf{W}_H^{(n)} \\
\mathbf{W}_H^{(k)} &= \mathbf{W}_H^{(0)} - \alpha_k \Delta t_H [\mathbf{R}(\mathbf{W}_H^{(k-1)})_H + \boldsymbol{\mathcal{Q}}_H] \quad \text{for } k = 1, 2, \ldots, m \qquad (3.116) \\
\mathbf{W}_H^{(n+1)} &= \mathbf{W}_H^{(m)} \ .
\end{aligned}
$$

The second step of the multigrid scheme represents the calculation of a new solution on the coarse grid, i.e. Eqn. (3.116) is advanced in time by performing one or several iterations. Within this flow solver, only one iteration is performed on the coarser grids. The first two steps are repeated until the coarsest grid level is reached. Then the solution is interpolated back from the coarsest grid to the next finer level. First, the coarse grid correction is computed. Then the correction is interpolated to the finer grid. Hence, the new solution on the finer grid can be written as

$$
\mathbf{W}_h^{(n+1),\text{new}} = \mathbf{W}_h^{(n+1)} + \mathbf{I}_H^h (\mathbf{W}_H^{(n+1)} - \mathbf{W}_H^0) \ , \qquad (3.117)
$$

where $\mathbf{I}_H^h$ denotes the prolongation operator. Once the solution on the finest multigrid level is updated, the multigrid cycle is restarted. Alternative strategies for traversing the grids can be derived; in this research so-called W-cycles are employed.

# Chapter 4

# Discrete Adjoint Solver

This chapter develops the discrete adjoint equations for the turbomachinery RANS solver described in chapter 3. First, the general adjoint principle is discussed in an algebraic approach, cf. section 4.1. Afterwards, the systematic approach for the development of the discrete adjoint solver is presented. The discrete adjoint equations are derived by applying control theory directly to the set of discrete field equations, cf. section 4.2. Special emphasis is put on the development of the turbomachinery specific features of the adjoint solver, i.e. on the derivation of flow-consistent adjoint boundary conditions and, to allow for a concurrent rotor-stator optimization and stage coupling, on the development of an exact adjoint counterpart to the non-reflective, conservative mixing-plane formulation used in the flow solver, cf. section 4.3. Finally, the discretization and linearization of the various objective functions employed within this work is described in section 4.4.

## 4.1 Adjoint Principle

A gradient-based optimization method uses the gradient of an objective function with respect to the design variables $\boldsymbol{\alpha}$ to determine a search direction suitable to optimize the design problem. The gradient of a scalar objective function $I = I(\mathbf{W}, \boldsymbol{\alpha})$ may be written as

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\partial I}{\partial \boldsymbol{\alpha}} + \frac{\partial I}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial \boldsymbol{\alpha}} \ , \tag{4.1}$$

and can be numerically estimated via finite-differences. Once the sensitivities $\delta\mathbf{W} = \partial\mathbf{W}/\partial\boldsymbol{\alpha}$ are calculated, this direct approach allows for the evaluation of many functionals in a short period of time. However, for a large number of design variables, the calculation of $\delta\mathbf{W}$ becomes an expensive procedure since it requires the evaluation of the governing equations for each design variable perturbation. The adjoint approach eliminates the dependence of the gradient on the flow variable sensitivity and is considerably more efficient if the number of design variables exceeds the number of objective functions. This is the case for nearly all current turbomachinery design problems, since usually a large number of design variables are required for an adequate geometrical parametrization. The adjoint equations are obtained from the following procedure. First, the variation of the steady-state governing equations,

$$\frac{\partial\mathbf{R}}{\partial\boldsymbol{\alpha}} + \frac{\partial\mathbf{R}}{\partial\mathbf{W}}\frac{\partial\mathbf{W}}{\partial\boldsymbol{\alpha}} = 0 \ , \tag{4.2}$$

is introduced, where $\mathbf{R} = \mathbf{R}(\mathbf{W}, \boldsymbol{\alpha})$ is the residual vector defined as in Eqn. 3.19. Second, multiplying Eqn. (4.2) by the transpose of a Lagrange multiplier $\boldsymbol{\psi}$ and subtracting the variation of the governing equations from the variation of the objective function (4.1) produces the augmented Lagrangian function

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\partial I}{\partial\boldsymbol{\alpha}} + \frac{\partial I}{\partial\mathbf{W}}\frac{\partial\mathbf{W}}{\partial\boldsymbol{\alpha}} - \boldsymbol{\psi}^T\left[\frac{\partial\mathbf{R}}{\partial\boldsymbol{\alpha}} + \frac{\partial\mathbf{R}}{\partial\mathbf{W}}\frac{\partial\mathbf{W}}{\partial\boldsymbol{\alpha}}\right] \ . \tag{4.3}$$

Regrouping Eqn. (4.3) yields

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\partial I}{\partial\boldsymbol{\alpha}} - \boldsymbol{\psi}^T\frac{\partial\mathbf{R}}{\partial\boldsymbol{\alpha}} + \left[\frac{\partial I}{\partial\mathbf{W}} - \boldsymbol{\psi}^T\frac{\partial\mathbf{R}}{\partial\mathbf{W}}\right]\frac{\partial\mathbf{W}}{\partial\boldsymbol{\alpha}} \ . \tag{4.4}$$

Since $\boldsymbol{\psi}$ is an arbitrary differentiable function, it may be chosen in such a way that the terms within the brackets of Eqn. (4.4) cancel each other and, thus, the gradient of the objective function can be evaluated directly from the geometrical variations without having to re-compute the flow variable sensitivities. This procedure determines the adjoint equation, which can be expressed, both for an algebraic equation and for a system of partial differential equations, as

$$\left[\frac{\partial\mathbf{R}}{\partial\mathbf{W}}\right]^T\boldsymbol{\psi} = \left[\frac{\partial I}{\partial\mathbf{W}}\right]^T . \tag{4.5}$$

The adjoint vector $\boldsymbol{\psi}$ defines the solution to the adjoint system and corresponds to the state vector $\mathbf{W}$ of the flow governing equations. The flux-like term $[\partial\mathbf{R}/\partial\mathbf{W}]^T\boldsymbol{\psi}$ represents the adjoint counterpart to the flux gradients appearing in the field equations, while the linearized objective function $[\partial I/\partial\mathbf{W}]^T$ on the right hand side of Eqn. (4.5) acts like a source term and drives the adjoint solution. The adjoint system (4.5) can be solved similarly to the flow governing equations in a time-marching manner by adding a pseudo-time derivative term $\partial\boldsymbol{\psi}/\partial t$ to the equation. The complexity of the system is of the same order as that of the underlying flow governing equations and, thus, the computational cost to produce the adjoint solution is comparable to the cost of finding the flow solution. Identical to the flow equations, multigrid, local time-stepping, and implicit residual smoothing can be employed to accelerate the convergence to steady state. After a solution to the adjoint system has been calculated, the gradient of the objective function with respect to the design variables is obtained from the simplified gradient expression

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\partial I}{\partial\boldsymbol{\alpha}} - \boldsymbol{\psi}^T\frac{\partial\mathbf{R}}{\partial\boldsymbol{\alpha}} \; . \tag{4.6}$$

The gradient formulation (4.6) is free of the flow variable sensitivities $\delta\mathbf{W}$, i.e. the gradient can be calculated without having to re-compute the flow solution and is obtained at the cost of $n$ grid perturbations for $n$ design variables. Note however, the adjoint equation, i.e. Eqn. (4.5), provides information for the sensitivity calculation of one objective functional only. Thus, each function of interest requires a separate adjoint calculation.

## 4.2 Discrete Adjoint Equations

This section discusses the derivation of the discrete adjoint field equations for the turbomachinery RANS solver presented in chapter 3. The crucial step in discretizing the adjoint equations, i.e. Eqn. (4.5), is the derivation and implementation of the left-hand-side, i.e. to obtain the exact linearization of the residual with respect to the flow variables. Since turbomachinery specific functionals typically depend only on the state at certain boundary planes, the linearized objective function, i.e. the right hand side of Eqn. (4.5), usually acts as a boundary source term. Its derivation will be discussed in a separate section along with the adjoint boundary conditions.

The discrete form of the adjoint equations, cf. Eqn. (4.5), for the steady-state solution of the discrete field equations defined in computational space (3.24), i.e. $\boldsymbol{\mathcal{R}}(\mathbf{W}) = 0$, is given by

$$\sum_i \sum_j \sum_k \left[\frac{\partial \boldsymbol{\mathcal{R}}}{\partial \mathbf{W}}\right]^T_{i,j,k} \boldsymbol{\psi}_{i,j,k} = \text{R.H.S.} \;, \tag{4.7}$$

where $\sum_i$, $\sum_j$, and $\sum_k$ implies summation over all domain cells in the $i$-, $j$-, and $k$-directions. Expanding the linearized residual provides

$$\sum_i \sum_j \sum_k \left[\frac{\partial \boldsymbol{\mathcal{F}}_n}{\partial \mathbf{W}} - \frac{\partial \boldsymbol{\mathcal{F}}_{\mathbf{v}n}}{\partial \mathbf{W}} - \frac{\partial \boldsymbol{\mathcal{F}}_{\mathbf{d}n}}{\partial \mathbf{W}} - \frac{\partial (\mathcal{J}\mathbf{Q})}{\partial \mathbf{W}}\right]^T_{i,j,k} \boldsymbol{\psi}_{i,j,k} = \text{R.H.S.} \;, \tag{4.8}$$

for $n = 1, 2, 3$ and where $\mathcal{J}\mathbf{Q}$ is the source term contribution from potential body forces, here the Coriolis force as well as the centrifugal force, and $\boldsymbol{\mathcal{F}}$, $\boldsymbol{\mathcal{F}}_{\mathbf{v}}$, and $\boldsymbol{\mathcal{F}}_{\mathbf{d}}$ are the discretized convective, viscous, and artificial dissipation fluxes respectively. The final set of discrete adjoint equations for a specific cell $(i, j, k)$ is obtained by further expanding the linearized discrete fluxes for each cell and collecting every term that is a function of the state vector at cell $(i, j, k)$. Thus, the resulting equations depend on the details of the numerical scheme used to solve the flow governing equations. In the underlying numerical scheme, cf. subsection 3.3.1, the convective flux at a cell face is calculated by taking the average of the flux contributions from the two cells next to the face. Hence, the calculation of the total convective flux balance for a certain cell requires information from six adjacent cells. The viscous flux through a cell face is calculated by taking the average of the fluxes computed at the face vertexes, where the vertex flux is calculated from the flow variables in the eight surrounding cells. Thus, the viscous flux is a function of its own and the twenty-six surrounding cells. The stencil of the artificial dissipative scheme spans over two cells in each direction, therefore, requiring information from a total of thirteen cells.

This illustrates that a full linearization of the discretized RANS equations proves to be a daunting task, due to the need to keep track of the contributions from numerous terms and cells. The development cost of the method increases rapidly with the order and size of the stencil of the numerical scheme. To reduce the complexity of deriving and implementing the adjoint equations, the turbulence model has not been linearized within this research. Instead, the adjoint solver takes advantage of the so-called constant eddy viscosity (CEV) approximation, which assumes that the variation of the turbulent viscosity is negligible. For the validity of the CEV approximation, see Marta and Shankaran [61].

The following four subsections illustrate the development of the individual flux and source term contributions to the discrete adjoint equations for the spatial discretization scheme employed in the turbomachinery RANS solver. First, the residual of each flux gradient is expanded and its individual terms are linearized. For a particular cell $(i, j, k)$ the variation of the residual is given by

$$\delta\boldsymbol{\mathcal{R}}_{i,j,k} = \left[\delta\boldsymbol{\mathcal{F}}_n - \delta\boldsymbol{\mathcal{F}}_{\mathbf{v}n} - \delta\boldsymbol{\mathcal{F}}_{\mathbf{d}n} - \delta(\mathcal{J}\mathbf{Q})\right]_{i,j,k} \ , \tag{4.9}$$

with the abbreviations

$$\delta\boldsymbol{\mathcal{F}} = \frac{\partial\boldsymbol{\mathcal{F}}}{\partial\mathbf{W}}\delta\mathbf{W} \ , \qquad \delta\boldsymbol{\mathcal{F}}_{\mathbf{v}} = \frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{v}}}{\partial\mathbf{W}}\delta\mathbf{W} \ , \tag{4.10}$$

$$\delta\boldsymbol{\mathcal{F}}_{\mathbf{d}} = \frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\delta\mathbf{W} \ , \qquad \delta(\mathcal{J}\mathbf{Q}) = \frac{\partial(\mathcal{J}\mathbf{Q})}{\partial\mathbf{W}}\delta\mathbf{W} \ . \tag{4.11}$$

Then, the individual adjoint contributions are derived and the adjoint fluxes are constructed by collecting the relevant terms.

### 4.2.1 Contribution from the Convective Fluxes

This subsection develops the contribution from the convective flux gradients of the governing equations to the discrete adjoint equations. All other flux contributions as well as the source term contributions are temporarily neglected.

From Eqn. (3.29) it follows that, for a particular cell $(i, j, k)$, the convective flux gradient produces the residual contributions

$$\boldsymbol{\mathcal{F}}_{i,j,k} = \mathbf{f}_{i+\frac{1}{2},j,k} - \mathbf{f}_{i-\frac{1}{2},j,k} + \mathbf{f}_{i,j+\frac{1}{2},k} - \mathbf{f}_{i,j-\frac{1}{2},k} + \mathbf{f}_{i,j,k+\frac{1}{2}} - \mathbf{f}_{i,j,k-\frac{1}{2}} \ . \tag{4.12}$$

Linearizing the convective terms consequently provides the sensitivities

$$\delta\boldsymbol{\mathcal{F}}_{i,j,k} = \delta\mathbf{f}_{i+\frac{1}{2},j,k} - \delta\mathbf{f}_{i-\frac{1}{2},j,k} + \delta\mathbf{f}_{i,j+\frac{1}{2},k} - \delta\mathbf{f}_{i,j-\frac{1}{2},k} + \delta\mathbf{f}_{i,j,k+\frac{1}{2}} - \delta\mathbf{f}_{i,j,k-\frac{1}{2}} \ . \tag{4.13}$$

The utilized second-order central-difference scheme averages the fluxes at the cell boundaries before the flux gradients are computed, cf. Eqn. (3.30). Thus, the variations of the convective fluxes computed at the cell faces are given by

$$\begin{aligned}
\delta \mathbf{f}_{i\pm\frac{1}{2},j,k} &= \frac{1}{2}\left(\delta \mathbf{f}_{i\pm1,j,k} + \delta \mathbf{f}_{i,j,k}\right) \ , && (4.14)\\
\delta \mathbf{f}_{i,j\pm\frac{1}{2},k} &= \frac{1}{2}\left(\delta \mathbf{f}_{i,j\pm1,k} + \delta \mathbf{f}_{i,j,k}\right) \ , && (4.15)\\
\delta \mathbf{f}_{i,j,k\pm\frac{1}{2}} &= \frac{1}{2}\left(\delta \mathbf{f}_{i,j,k\pm1} + \delta \mathbf{f}_{i,j,k}\right) \ . && (4.16)
\end{aligned}$$

Introducing the metric terms, the convective flux variations can be expressed as a function of the fluxes defined in the physical space, cf. Eqns. (3.22) and (3.31)-(3.33), and may be written as

$$\begin{aligned}
\delta \mathbf{f}_{i\pm\frac{1}{2},j,k} &= \frac{1}{2}\left[\delta\left(S_{1m_{i\pm\frac{1}{2},j,k}} \mathbf{F}_{m_{i\pm1,j,k}}\right) + \delta\left(S_{1m_{i\pm\frac{1}{2},j,k}} \mathbf{F}_{m_{i,j,k}}\right)\right] \ , && (4.17)\\
\delta \mathbf{f}_{i,j\pm\frac{1}{2},k} &= \frac{1}{2}\left[\delta\left(S_{2m_{i,j\pm\frac{1}{2},k}} \mathbf{F}_{m_{i,j\pm1,k}}\right) + \delta\left(S_{2m_{i,j\pm\frac{1}{2},k}} \mathbf{F}_{m_{i,j,k}}\right)\right] \ , && (4.18)\\
\delta \mathbf{f}_{i,j,k+\frac{1}{2}} &= \frac{1}{2}\left[\delta\left(S_{3m_{i,j,k\pm\frac{1}{2}}} \mathbf{F}_{m_{i,j,k\pm1}}\right) + \delta\left(S_{3m_{i,j,k\pm\frac{1}{2}}} \mathbf{F}_{m_{i,j,k}}\right)\right] \ , && (4.19)
\end{aligned}$$

where summation over $m = 1, 2, 3$ is implied by a repeated index $m$. Expanding Eqns. (4.17)-(4.19) yields

$$\begin{aligned}
\delta \mathbf{f}_{i\pm\frac{1}{2},j,k} &= \frac{1}{2}\Big[\delta S_{1m_{i\pm\frac{1}{2},j,k}} \mathbf{F}_{m_{i\pm1,j,k}} + \delta S_{1m_{i\pm\frac{1}{2},j,k}} \mathbf{F}_{m_{i,j,k}} \\
&\quad + S_{1m_{i\pm\frac{1}{2},j,k}} \delta\mathbf{F}_{m_{i\pm1,j,k}} + S_{1m_{i\pm\frac{1}{2},j,k}} \delta\mathbf{F}_{m_{i,j,k}}\Big] \ , && (4.20)\\
\delta \mathbf{f}_{i,j\pm\frac{1}{2},k} &= \frac{1}{2}\Big[\delta S_{2m_{i,j\pm\frac{1}{2},k}} \mathbf{F}_{m_{i,j\pm1,k}} + \delta S_{2m_{i,j\pm\frac{1}{2},k}} \mathbf{F}_{m_{i,j,k}} \\
&\quad + S_{2m_{i,j\pm\frac{1}{2},k}} \delta\mathbf{F}_{m_{i,j\pm1,k}} + S_{2m_{i,j\pm\frac{1}{2},k}} \delta\mathbf{F}_{m_{i,j,k}}\Big] \ , && (4.21)\\
\delta \mathbf{f}_{i,j,k+\frac{1}{2}} &= \frac{1}{2}\Big[\delta S_{3m_{i,j,k\pm\frac{1}{2}}} \mathbf{F}_{m_{i,j,k\pm1}} + \delta S_{3m_{i,j,k\pm\frac{1}{2}}} \mathbf{F}_{m_{i,j,k}} \\
&\quad + S_{3m_{i,j,k\pm\frac{1}{2}}} \delta\mathbf{F}_{m_{i,j,k\pm1}} + S_{3m_{i,j,k\pm\frac{1}{2}}} \delta\mathbf{F}_{m_{i,j,k}}\Big] \ . && (4.22)
\end{aligned}$$

Utilizing the chain rule, these expressions can be further modified to produce

$$\begin{aligned}
\delta \mathbf{f}_{i\pm\frac{1}{2},j,k} &= \frac{1}{2}\Big[\delta S_{1m_{i\pm\frac{1}{2},j,k}}\mathbf{F}_{m_{i\pm1,j,k}} + \delta S_{1m_{i\pm\frac{1}{2},j,k}}\mathbf{F}_{m_{i,j,k}} \\
&\quad + S_{1m_{i\pm\frac{1}{2},j,k}}(\mathbf{A}_m\delta\mathbf{W})_{i\pm1,j,k} + S_{1m_{i\pm\frac{1}{2},j,k}}(\mathbf{A}_m\delta\mathbf{W})_{i,j,k}\Big]\ , \qquad (4.23) \\
\delta \mathbf{f}_{i,j\pm\frac{1}{2},k} &= \frac{1}{2}\Big[\delta S_{2m_{i,j\pm\frac{1}{2},k}}\mathbf{F}_{m_{i,j\pm1,k}} + \delta S_{2m_{i,j\pm\frac{1}{2},k}}\mathbf{F}_{m_{i,j,k}} \\
&\quad + S_{2m_{i,j\pm\frac{1}{2},k}}(\mathbf{A}_m\delta\mathbf{W})_{i,j\pm1,k} + S_{2m_{i,j\pm\frac{1}{2},k}}(\mathbf{A}_m\delta\mathbf{W})_{i,j,k}\Big]\ , \qquad (4.24) \\
\delta \mathbf{f}_{i,j,k+\frac{1}{2}} &= \frac{1}{2}\Big[\delta S_{3m_{i,j,k\pm\frac{1}{2}}}\mathbf{F}_{m_{i,j,k\pm1}} + \delta S_{3m_{i,j,k\pm\frac{1}{2}}}\mathbf{F}_{m_{i,j,k}} \\
&\quad + S_{3m_{i,j,k\pm\frac{1}{2}}}(\mathbf{A}_m\delta\mathbf{W})_{i,j,k\pm1} + S_{3m_{i,j,k\pm\frac{1}{2}}}(\mathbf{A}_m\delta\mathbf{W})_{i,j,k}\Big]\ , \qquad (4.25)
\end{aligned}$$

where $\mathbf{A}_m$ are the convective flux Jacobians $\partial\mathbf{F}_m/\partial\mathbf{W}$ as defined in the appendix A.2.1. Hence, the linearization of the convective flux gradient at cell $(i,j,k)$ produces the seven flow variable sensitivities $\delta\mathbf{W}_{i,j,k}$, $\delta\mathbf{W}_{i\pm1,j,k}$, $\delta\mathbf{W}_{i,j\pm1,k}$, and $\delta\mathbf{W}_{i,j,k\pm1}$. Substituting Eqns. (4.23)-(4.25) into Eqn. (4.13), discarding the terms which include metric variations $\delta S$ since these terms contribute to the gradient and not to the adjoint equations, and reordering yields

$$\begin{aligned}
\delta\boldsymbol{\mathcal{F}}_{i,j,k} &= \frac{1}{2}\Big[\Big(\ S_{1m_{i+\frac{1}{2},j,k}} - S_{1m_{i-\frac{1}{2},j,k}} + S_{2m_{i,j+\frac{1}{2},k}} \\
&\quad - S_{2m_{i,j-\frac{1}{2},k}} + S_{3m_{i,j,k+\frac{1}{2}}} - S_{3m_{i,j,k-\frac{1}{2}}}\ \Big)\,\mathbf{A}_{m,i,j,k}\delta\mathbf{W}_{i,j,k} \\
&\quad + S_{1m_{i+\frac{1}{2},j,k}}\mathbf{A}_{m,i+1,j,k}\delta\mathbf{W}_{i+1,j,k} - S_{1m_{i-\frac{1}{2},j,k}}\mathbf{A}_{m,i-1,j,k}\delta\mathbf{W}_{i-1,j,k} \\
&\quad + S_{2m_{i,j+\frac{1}{2},k}}\mathbf{A}_{m,i,j+1,k}\delta\mathbf{W}_{i,j+1,k} - S_{2m_{i,j-\frac{1}{2},k}}\mathbf{A}_{m,i,j-1,k}\delta\mathbf{W}_{i,j-1,k} \\
&\quad + S_{3m_{i,j,k+\frac{1}{2}}}\mathbf{A}_{m,i,j,k+1}\delta\mathbf{W}_{i,j,k+1} - S_{3m_{i,j,k-\frac{1}{2}}}\mathbf{A}_{m,i,j,k-1}\delta\mathbf{W}_{i,j,k-1}\Big]\ . (4.26)
\end{aligned}$$

As stated above, the discrete adjoint equations for a specific cell $(i,j,k)$ are obtained by expanding the linearized residual for each cell in the computational domain and by collecting every term that is a function of the state vector at cell $(i,j,k)$. Consequently, the linearized convective flux gradient at cell $(i,j,k)$ contributes to the adjoint fluxes of the cells $(i,j,k)$, $(i\pm1,j,k)$, $(i,j\pm1,k)$, and $(i,j,k\pm1)$, while the adjoint flux $\mathfrak{F}$ at cell $(i, j, k)$ has contributions from the convective flux gradients $\mathcal{F}$ of the same seven cells, i.e.

$$\begin{aligned}
\mathfrak{F}(\boldsymbol{\psi})_{i,j,k} \;=\;& \left[\frac{\partial \boldsymbol{\mathcal{F}}_m}{\partial \mathbf{W}}\right]^T_{i+1,j,k} \boldsymbol{\psi}_{i+1,j,k} + \left[\frac{\partial \boldsymbol{\mathcal{F}}_m}{\partial \mathbf{W}}\right]^T_{i,j+1,k} \boldsymbol{\psi}_{i,j+1,k} + \left[\frac{\partial \boldsymbol{\mathcal{F}}_m}{\partial \mathbf{W}}\right]^T_{i,j,k+1} \boldsymbol{\psi}_{i,j,k+1} \\
& + \left[\frac{\partial \boldsymbol{\mathcal{F}}_m}{\partial \mathbf{W}}\right]^T_{i-1,j,k} \boldsymbol{\psi}_{i-1,j,k} + \left[\frac{\partial \boldsymbol{\mathcal{F}}_m}{\partial \mathbf{W}}\right]^T_{i,j-1,k} \boldsymbol{\psi}_{i,j-1,k} + \left[\frac{\partial \boldsymbol{\mathcal{F}}_m}{\partial \mathbf{W}}\right]^T_{i,j,k-1} \boldsymbol{\psi}_{i,j,k-1} \\
& + \left[\frac{\partial \boldsymbol{\mathcal{F}}_m}{\partial \mathbf{W}}\right]^T_{i,j,k} \boldsymbol{\psi}_{i,j,k} \;. \qquad (4.27)
\end{aligned}$$

Substituting Eqn. (4.26) into each term $\partial\boldsymbol{\mathcal{F}}_m/\partial\mathbf{W}$ of Eqn. (4.27) and keeping only the terms that are functions of the variation $\delta\mathbf{W}_{i,j,k}$ provides the contribution from the convective fluxes to the discrete adjoint equations

$$\begin{aligned}
\mathfrak{F}(\boldsymbol{\psi})_{i,j,k} \;=\;& \frac{1}{2}\Big[ -S_{1m_{i+\frac{1}{2},j,k}} \mathbf{A}^T_{m,i,j,k} \boldsymbol{\psi}_{i+1,j,k} - S_{2m_{i,j+\frac{1}{2},k}} \mathbf{A}^T_{m,i,j,k} \boldsymbol{\psi}_{i,j+1,k} \\
& -S_{3m_{i,j,k+\frac{1}{2}}} \mathbf{A}^T_{m,i,j,k} \boldsymbol{\psi}_{i,j,k+1} + S_{1m_{i-\frac{1}{2},j,k}} \mathbf{A}^T_{m,i,j,k} \boldsymbol{\psi}_{i-1,j,k} \\
& +S_{2m_{i,j-\frac{1}{2},k}} \mathbf{A}^T_{m,i,j,k} \boldsymbol{\psi}_{i,j-1,k} + S_{3m_{i,j,k-\frac{1}{2}}} \mathbf{A}^T_{m,i,j,k} \boldsymbol{\psi}_{i,j,k-1} \\
& +\left(\, S_{1m_{i+\frac{1}{2},j,k}} - S_{1m_{i-\frac{1}{2},j,k}} + S_{2m_{i,j+\frac{1}{2},k}} \right. \\
& \left. -S_{2m_{i,j-\frac{1}{2},k}} + S_{3m_{i,j,k+\frac{1}{2}}} - S_{3m_{i,j,k-\frac{1}{2}}} \,\right) \mathbf{A}_{m,i,j,k} \;. \qquad (4.28)
\end{aligned}$$

Rearranging Eqn.(4.28) yields a more convenient formulation of the adjoint fluxes,

$$\begin{aligned}
\mathfrak{F}(\boldsymbol{\psi})_{i,j,k} \;=\;& -\frac{1}{2}\Big[ \quad S_{1m_{i+\frac{1}{2},j,k}} \mathbf{A}^T_{m,i,j,k} (\boldsymbol{\psi}_{i+1,j,k} - \boldsymbol{\psi}_{i,j,k}) \\
& -S_{1m_{i-\frac{1}{2},j,k}} \mathbf{A}^T_{m,i,j,k} (\boldsymbol{\psi}_{i-1,j,k} - \boldsymbol{\psi}_{i,j,k}) \\
& +S_{2m_{i,j+\frac{1}{2},k}} \mathbf{A}^T_{m,i,j,k} (\boldsymbol{\psi}_{i,j+1,k} - \boldsymbol{\psi}_{i,j,k}) \\
& -S_{2m_{i,j-\frac{1}{2},k}} \mathbf{A}^T_{m,i,j,k} (\boldsymbol{\psi}_{i,j-1,k} - \boldsymbol{\psi}_{i,j,k}) \\
& +S_{3m_{i,j,k+\frac{1}{2}}} \mathbf{A}^T_{m,i,j,k} (\boldsymbol{\psi}_{i,j,k+1} - \boldsymbol{\psi}_{i,j,k}) \\
& -S_{3m_{i,j,k-\frac{1}{2}}} \mathbf{A}^T_{m,i,j,k} (\boldsymbol{\psi}_{i,j,k-1} - \boldsymbol{\psi}_{i,j,k}) \Big] \;. \qquad (4.29)
\end{aligned}$$

Thus, the convective discrete adjoint flux is built of the transpose of the local convective flux Jacobian multiplied with the fluxes produced by the co-state differences between the neighbouring and current cells, which are evaluated at the corresponding cell faces and therefore multiplied with the appropriate face metrics.

### 4.2.2 Contribution from the Viscous Fluxes

This subsection deals with the contribution from the viscous flux gradients of the Navier-Stokes equations to the discrete adjoint equations. All other flux contributions as well as the source term contributions are temporarily neglected. The procedure is similar to the derivation of the discrete convective adjoint fluxes illustrated in the previous subsection; however, the task of producing the viscous counterpart is more complex due to the larger stencil of the numerical scheme originating from the necessity of approximating the partial derivatives of velocity appearing in the viscous stress tensor and the temperature gradients present in the energy equation.

From Eqn. (3.29) it follows that, for a particular cell $(i, j, k)$, the viscous flux gradient produces the residual contributions

$$\boldsymbol{\mathcal{F}}_{\mathbf{v}_{i,j,k}} = -\mathbf{f}_{\mathbf{v}_{i+\frac{1}{2},j,k}} + \mathbf{f}_{\mathbf{v}_{i-\frac{1}{2},j,k}} - \mathbf{f}_{\mathbf{v}_{i,j+\frac{1}{2},k}} + \mathbf{f}_{\mathbf{v}_{i,j-\frac{1}{2},k}} - \mathbf{f}_{\mathbf{v}_{i,j,k+\frac{1}{2}}} + \mathbf{f}_{\mathbf{v}_{i,j,k-\frac{1}{2}}} \ . \tag{4.30}$$

Linearizing the viscous terms consequently provides the sensitivities

$$\boldsymbol{\mathcal{F}}_{\mathbf{v}_{i,j,k}} = -\delta\mathbf{f}_{\mathbf{v}_{i+\frac{1}{2},j,k}} + \delta\mathbf{f}_{\mathbf{v}_{i-\frac{1}{2},j,k}} - \delta\mathbf{f}_{\mathbf{v}_{i,j+\frac{1}{2},k}} + \delta\mathbf{f}_{\mathbf{v}_{i,j-\frac{1}{2},k}} - \delta\mathbf{f}_{\mathbf{v}_{i,j,k+\frac{1}{2}}} + \delta\mathbf{f}_{\mathbf{v}_{i,j,k-\frac{1}{2}}} \ . \tag{4.31}$$

The variations of the viscous fluxes at the cell faces can be calculated from the physical boundary flux sensitivities through the transformation

$$\delta\mathbf{f}_{\mathbf{v}_{i\pm\frac{1}{2},j,k}} = \delta\big(S_{1m_{i\pm\frac{1}{2},j,k}}\mathbf{F}_{\mathbf{v}_{m_{i\pm\frac{1}{2},j,k}}}\big) \ , \tag{4.32}$$

$$\delta\mathbf{f}_{\mathbf{v}_{i,j\pm\frac{1}{2},k}} = \delta\big(S_{2m_{i,j\pm\frac{1}{2},k}}\mathbf{F}_{\mathbf{v}_{m_{i,j\pm\frac{1}{2},k}}}\big) \ , \tag{4.33}$$

$$\delta\mathbf{f}_{\mathbf{v}_{i,j,k+\frac{1}{2}}} = \delta\big(S_{3m_{i,j,k\pm\frac{1}{2}}}\mathbf{F}_{\mathbf{v}_{m_{i,j,k\pm\frac{1}{2}}}}\big) \ , \tag{4.34}$$

where summation over $m = 1, 2, 3$ is implied by a repeated index $m$. Expanding

Eqns. (4.32)-(4.34) by the chain rule yields

$$
\begin{aligned}
\delta \mathbf{f}_{\mathbf{v}_{i\pm\frac{1}{2},j,k}} &= \delta S_{1m_{i\pm\frac{1}{2},j,k}} \mathbf{F}_{\mathbf{v}_{m_{i\pm\frac{1}{2},j,k}}} + S_{1m_{i\pm\frac{1}{2},j,k}} \delta \mathbf{F}_{\mathbf{v}_{m_{i\pm\frac{1}{2},j,k}}} \ , && (4.35) \\
\delta \mathbf{f}_{\mathbf{v}_{i,j\pm\frac{1}{2},k}} &= \delta S_{2m_{i,j\pm\frac{1}{2},k}} \mathbf{F}_{\mathbf{v}_{m_{i,j\pm\frac{1}{2},k}}} + S_{2m_{i,j\pm\frac{1}{2},k}} \delta \mathbf{F}_{\mathbf{v}_{m_{i,j\pm\frac{1}{2},k}}} \ , && (4.36) \\
\delta \mathbf{f}_{\mathbf{v}_{i,j,k+\frac{1}{2}}} &= \delta S_{3m_{i,j,k\pm\frac{1}{2}}} \mathbf{F}_{\mathbf{v}_{m_{i,j,k\pm\frac{1}{2}}}} + S_{3m_{i,j,k\pm\frac{1}{2}}} \delta \mathbf{F}_{\mathbf{v}_{m_{i,j,k\pm\frac{1}{2}}}} \ , && (4.37)
\end{aligned}
$$

and, thus, produces terms that are multiplied to metric variations and terms that are functions of the flow variable sensitivities. The metric variations will be ignored in the remainder of this section since these terms are attributed to the gradient.

The second-order accurate discretization scheme utilized to calculate the viscous fluxes estimates the viscous fluxes at the cell faces by averaging the flux quantities defined at the four vertexes of the face. Thus, the variation of the viscous flux vector at cell face $(i+\frac{1}{2},j,k)$ expands to

$$
\begin{aligned}
\delta \mathbf{F}_{\mathbf{v}_{m_{i+\frac{1}{2},j,k}}} = \frac{1}{4}\Big( & \delta \mathbf{F}_{\mathbf{v}_{m_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}}} + \delta \mathbf{F}_{\mathbf{v}_{m_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}}} \\
& + \delta \mathbf{F}_{\mathbf{v}_{m_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}}} + \delta \mathbf{F}_{\mathbf{v}_{m_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}}} \Big) \ . \qquad (4.38)
\end{aligned}
$$

The variations at the other cell faces take on similar expressions. From Eqns. (2.10) and (2.15) it follows that the sensitivity of the viscous flux at the vertex $(i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2})$ can be written as

$$
\delta \mathbf{F}_{\mathbf{v}_{m_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}}} = \left\{ \begin{array}{c} 0 \\ \delta\tau_{mn}\hat{\delta}_{n1} \\ \delta\tau_{mn}\hat{\delta}_{n2} \\ \delta\tau_{mn}\hat{\delta}a_{n3} \\ \delta u_n \tau_{mn} + u_n \delta\tau_{mn} + \delta k \frac{\delta T}{\delta x_m} + k \delta \frac{\delta T}{\delta x_m} \end{array} \right\}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \ , \qquad (4.39)
$$

with $n = 1, 2, 3$ and the Kronecker delta function $\hat{\delta}$. The variation of the viscous fluxes at all other cell vertexes are defined in a similar fashion. The further derivation of the discrete viscous adjoint fluxes is split up into two sections, which develop the contributions from the momentum and energy equations separately.

### Contribution from the Momentum Equation

The contribution from the momentum equation to the discrete viscous adjoint fluxes arise from the variation of the stress tensor terms $\delta\tau_{mn}$. The linearized stress tensor terms defined in Eqn. (4.39) expands to

$$\delta\tau_{mn_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}} = \left\{\mu\left[\delta\left(\frac{\partial u_m}{\partial x_n}\right) + \delta\left(\frac{\partial u_n}{\partial x_m}\right) + \frac{2}{3}\delta\left(\frac{\partial u_l}{\partial x_l}\right)\hat{\delta}_{mn}\right]\right\}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} . \quad (4.40)$$

Note, the viscosity coefficient $\mu$ is treated as a constant and hence its variation is zero. The stress tensor is a function of the primitive variables $\mathbf{W_P} = [\rho, u_1, u_2, u_3, p]^T$ and not directly of the flow variables $\mathbf{W} = [\rho, \rho u_1, \rho u_2, \rho u_3, \rho E]^T$. Therefore, it is convenient to first calculate the sensitivities and adjoint contributions with respect to the primitive variables. From Eqn. (4.40) it can be observed that the variation of the stress tensor produces the primitive variable sensitivities $\delta u_1$, $\delta u_2$, and $\delta u_3$. At the end of the derivation the viscous adjoint fluxes are then multiplied with a transformation operator to express the adjoint fluxes in terms of the flow variable sensitivities such that the discrete viscous adjoint fluxes can be added to the discrete convective and artificial dissipation fluxes, i.e.

$$\delta\boldsymbol{\mathcal{F}}_{\mathbf{v}} = \frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{v}}}{\partial\mathbf{W}}\delta\mathbf{W} = \frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{v}}}{\partial\mathbf{W_P}}\frac{\partial\mathbf{W_P}}{\partial\mathbf{W}}\delta\mathbf{W} = \frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{v}}}{\partial\mathbf{W_P}}\mathbf{M}^{-1}\delta\mathbf{W} \ , \quad (4.41)$$

where the transformation matrix $\mathbf{M}^{-1} = \partial\mathbf{W_P}/\partial\mathbf{W}$ is provided in the appendix, cf. Eqn. (A.4).

With the definition of the velocity gradients, cf. Eqn. (3.42), Eqn. (4.40) may be written as

$$\begin{aligned}\delta\tau_{mn_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}} = \Bigg\{\mu\Bigg[ & \left(\frac{1}{J}\left[\delta\frac{\partial u^*_{m_{1n}}}{\partial\xi_1} + \delta\frac{\partial u^*_{m_{2n}}}{\partial\xi_2} + \delta\frac{\partial u^*_{m_{3n}}}{\partial\xi_3}\right]\right) \\ + & \left(\frac{1}{J}\left[\delta\frac{\partial u^*_{n_{1m}}}{\partial\xi_1} + \delta\frac{\partial u^*_{n_{2m}}}{\partial\xi_2} + \delta\frac{\partial u^*_{n_{3m}}}{\partial\xi_3}\right]\right) \\ + & \frac{2}{3}\left(\frac{1}{J}\left[\delta\frac{\partial u^*_{l_{1l}}}{\partial\xi_1} + \delta\frac{\partial u^*_{l_{2l}}}{\partial\xi_2} + \delta\frac{\partial u^*_{l_{3l}}}{\partial\xi_3}\right]\right)\hat{\delta}_{mn}\Bigg]\Bigg\}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \end{aligned} \quad (4.42)$$

where $J$ is the volume of the auxiliary control volume, which is approximated by averaging the volumes of the eight cells that share the vertex, cf. subsection 3.3.1. Following Eqn.(3.43), the variation of the velocity gradient $\delta(\partial u^*_{m_{1n}}/\partial\xi_1)$ defined in the computational domain is given by

$$\begin{aligned}\delta\left[\frac{\partial u^*_{m_{1n}}}{\partial\xi_1}\right]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} &= \frac{1}{4}\big(\delta u^*_{m_{1n,i+1,j,k}} - \delta u^*_{m_{1n,i,j,k}} + \delta u^*_{m_{1n,i+1,j+1,k+1}} - \delta u^*_{m_{1n,i,j+1,k+1}} \\ &\quad + \delta u^*_{m_{1n,i+1,j+1,k}} - \delta u^*_{m_{1n,i,j+1,k}} + \delta u^*_{m_{1n,i+1,j,k+1}} - \delta u^*_{m_{1n,i,j,k+1}}\big)\ , \qquad (4.43)\end{aligned}$$

The linearization of the other velocity gradient terms in Eqn. (4.42) take on similar expressions. In Eqn. (4.43), $\delta u^*$ are the sensitivities of the velocities multiplied with the cell metrics interpolated to the cell center by averaging the two corresponding face metrics; for example

$$\delta u^*_{m_{1n,i,j,k}} = \frac{\delta S_{1n_{i+\frac{1}{2},j,k}} + \delta S_{1n_{i-\frac{1}{2},j,k}}}{2}\, u_{m_{i,j,k}} + \frac{S_{1n_{i+\frac{1}{2},j,k}} + S_{1n_{i-\frac{1}{2},j,k}}}{2}\, \delta u_{m_{i,j,k}}\ . \qquad (4.44)$$

Hence, the variation of the stress tensor generates additional metric variations which contribute to the gradient formulation.

The derivation shows that the stress tensor sensitivity $\delta\tau_{mn_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}}$ produces velocity variations in eight cells; namely $(i,j,k)$, $(i{+}1,j,k)$, $(i,j{+}1,k)$, $(i,j,k{+}1)$, $(i{+}1,j{+}1,k)$, $(i{+}1,j,k{+}1)$, $(i,j{+}1,k{+}1)$, and $(i{+}1,j{+}1,k{+}1)$. From Eqns. (4.38) and (4.31) it follows that the contribution from the momentum equation spans over the entire stencil of the second-order accurate spatial discretization scheme. Therefore, the linearized viscous flux gradient $\delta\boldsymbol{\mathcal{F}}_{\mathbf{v}_{i,j,k}}$ produces flow variable sensitivities in its own and the twenty-six surrounding cells. The momentum equation contributes terms multiplied with the velocity variations $\delta u_m$ at each of these cells. Accordingly, the discrete viscous adjoint flux $\mathfrak{F}_{\mathbf{v}}$ at cell $(i,j,k)$ has contributions from the viscous flux gradients $\boldsymbol{\mathcal{F}}_{\mathbf{v}}$ of its own and the twenty-six surrounding cells, i.e.

$$\begin{aligned}\mathfrak{F}_{\mathbf{v}}(\boldsymbol{\psi})_{i,j,k} \;&=\; \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i+1,j+1,k} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i+1,j,k+1} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i-1,j,k+1} \\ +\;& \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i,j+1,k} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i,j+1,k+1} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i+1,j-1,k-1} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i+1,j,k-1} \\ +\;& \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i,j,k+1} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i-1,j-1,k} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i-1,j+1,k-1} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i,j+1,k-1}\end{aligned}$$

$$
\begin{aligned}
&+ \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i-1,j,k} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i-1,j,k-1} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i-1,j-1,k+1} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i,j-1,k+1}\\
&+ \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i,j-1,k} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i,j-1,k-1} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i+1,j+1,k-1} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i+1,j+1,k+1}\\
&+ \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i,j,k-1} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi} i\right]_{i+1,j-1,k} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i+1,j-1,k+1} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i-1,j-1,k-1}\\
&+ \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i+1,j,k} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i-1,j+1,k} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i-1,j+1,k+1} + \left[\delta\boldsymbol{\mathcal{F}}^T_{\mathbf{v}_m}\boldsymbol{\psi}\right]_{i,j,k}, \quad (4.45)
\end{aligned}
$$

with the abbreviation $\delta\boldsymbol{\mathcal{F}}_{\mathbf{v}_m} = (\partial\boldsymbol{\mathcal{F}}_{\mathbf{v}_m}\partial\mathbf{W_P})\mathbf{M}^{-1}$. Substituting Eqn. (4.42) with Eqn. (4.43) and Eqn. (4.44) into Eqn. (4.45) and collecting all the terms that are multiplied with the variations $\delta u_{m_{i,j,k}}$ provides the discrete adjoint flux due to the momentum equation with respect to the primitive variable variations. Multiplication with the transformation matrix $\mathbf{M}^{-1}$ finally provides the momentum flux contribution in terms of the flow variable sensitivities. For a more detailed discussion of the construction of the discrete adjoint momentum flux the reader should refer to Nadarajah [56].

### Contribution from the Energy Equation

The contribution from the energy equation to the discrete viscous adjoint fluxes can be expressed as

$$\delta F_{vm_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}} = \left(\delta u_n \tau_{mn} + u_n \delta\tau_{mn} + k\delta\frac{\partial T}{\partial x_m}\right)_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} . \quad (4.46)$$

The coefficient of thermal conductivity $k$ is treated as a constant and hence its variation is zero. Following Eqn. (4.46), the contribution from the energy equation can be divided into three parts: the variation of the stress tensors $\delta\tau_{mn}$, the contribution from the velocity variations $\delta u_n$, and the variation of the temperature gradients $\delta(\partial T/\partial x_m)$. As done for the development of the contribution from the momentum equation, it is convenient to first calculate the sensitivities and adjoint contributions with respect to the primitive variables. From Eqn. (4.46) it can be observed that the variation of the energy equation produces the primitive variable sensitivities $\delta u_1$, $\delta u_2$, $\delta u_3$, and $\delta T$. Utilizing the ideal gas law the temperature can be expressed as a function of density and pressure; hence, the temperature variations can easily be transferred into variations of density and pressure.

The variation of the stress tensor terms from the energy equation produces primitive variable variations similar to the terms obtained in the momentum equation. The only difference is that the energy equation stress tensor sensitivities are multiplied with the velocities $u_n$. However, this has no impact on the development of the adjoint contributions. Thus, the linearization of the stress tensor terms is not further discussed; the reader may refer to the previous section for a detailed derivation of the stress tensor sensitivities.

The second contribution originates from the variation of the velocity components at the face vertexes. The numerical scheme estimates these velocities by averaging the velocity values of the eight cells that share the same vertex, cf. Eqn. (3.41). Accordingly, the variation of the vertex velocity expands to

$$\begin{aligned}\delta u_{n,i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \frac{1}{8}(\ & \delta u_{n,i,j,k} + \delta u_{n,i+1,j,k} + \delta u_{n,i,j+1,k} + \delta u_{n,i+1,j+1,k} + \\ & \delta u_{n,i,j,k+1} + \delta u_{n,i+1,j,k+1} + \delta u_{n,i,j+1,k+1} + \delta u_{n,i+1,j+1,k+1}). \quad (4.47)\end{aligned}$$

The velocity sensitivities at the other face vertexes are defined in a similar manner.

The third contribution to the discrete viscous adjoint fluxes is provided by the variation of the temperature gradient. Utilizing the ideal gas law (2.1), the temperature gradient can be expressed in terms of the primitive variables, pressure and density,

$$\frac{\partial T}{\partial x_m} = \frac{\partial}{\partial x_m}\left(\frac{p}{\rho}\right) \ . \qquad (4.48)$$

Thus, the variation of the temperature gradient is given by

$$\left[k\delta\frac{\partial T}{\partial x_m}\right]_{n,i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \left[k\delta\frac{\partial}{\partial x_m}\left(\frac{p}{\rho}\right)\right]_{n,i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} , \qquad (4.49)$$

which can be expressed in terms of a pressure and a density variation,

$$\left[k\delta\frac{\partial T}{\partial x_m}\right]_{n,i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \left[k\frac{\partial}{\partial x_m}\left(\frac{1}{\rho}\delta p - \frac{p}{\rho^2}\delta\rho\right)\right]_{n,i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} . \qquad (4.50)$$

The modified temperature gradient in Eqn. (4.50) is treated in a fashion similar to the velocity gradients in the momentum equation, cf. Eqns. (3.42)-(3.44).

The final discrete adjoint flux contribution from the energy equation is obtained through the following steps: First, substituting the stress tensor variations derived in the previous section, the velocity variations (4.47), and the temperature variations (4.50) into Eqn. (4.46) completes the linearization of the flux from the energy equation. Second, substituting the linearized energy flux into Eqn. (4.45) and again keeping only the terms that are functions of the primitive variable variations at cell $(i, j, k)$ provides the discrete adjoint energy flux. Third, multiplication with the transformation matrix $\mathbf{M}^{-1}$ yields the adjoint energy flux in terms of the flow variable sensitivities. A more elaborate derivation of the discrete adjoint energy fluxes can be found in Nadarajah [56].

### 4.2.3 Contribution from the Source Term

This subsection develops the contribution from the source term to the discrete adjoint equations. All flux contributions are temporarily neglected.

The source term only occurs and contributes to the adjoint equations when the governing equations are formulated in a rotating frame of reference. From Eqn. (3.25) it follows that, for a particular cell $(i, j, k)$, the source term, which accounts for the effects due to the Coriolis force and centrifugal force, produces the residual contribution $(\mathcal{J}\mathbf{Q})_{i,j,k}$, where $J$ is the volume of the cell and $\mathbf{Q}$ is obtained from Eqn. (2.11). Thus, the linearization of the source term is straight forward to derive;

$$\delta(\mathcal{J}\mathbf{Q})_{i,j,k} = \left(\frac{\partial(\mathcal{J}\mathbf{Q})}{\partial\mathbf{W}}\delta\mathbf{W}\right)_{i,j,k} = \left(\mathcal{J}\frac{\partial\mathbf{Q}}{\partial\mathbf{W}}\delta\mathbf{W}\right)_{i,j,k} , \tag{4.51}$$

where

$$\frac{\partial\mathbf{Q}}{\partial\mathbf{W}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ u_2 & 0 & 0 & \Omega_1 & 0 \\ u_3 & 0 & -\Omega_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} . \tag{4.52}$$

Accordingly, the adjoint contribution from the source term is simply given by

$$\left(\frac{\partial\mathbf{Q}}{\partial\mathbf{W}}\right)^T \boldsymbol{\psi} = [\ 0,\ 0, -\Omega_1\psi_4,\ \Omega_1\psi_3,\ 0\ ]^T . \tag{4.53}$$

### 4.2.4 Contribution from the Artificial Dissipation Fluxes

This subsection develops the contribution from the artificial dissipation scheme to the discrete adjoint equations. All other flux contributions as well as the source term contributions are temporarily neglected.

From Eqn. (3.29) it follows that, for a particular cell $(i, j, k)$, the artificial dissipation flux produces the residual contributions

$$\boldsymbol{\mathcal{F}}_{\mathbf{d}_{i,j,k}} = -\mathbf{f}_{\mathbf{d}_{i+\frac{1}{2},j,k}} + \mathbf{f}_{\mathbf{d}_{i-\frac{1}{2},j,k}} - \mathbf{f}_{\mathbf{d}_{i,j+\frac{1}{2},k}} + \mathbf{f}_{\mathbf{d}_{i,j-\frac{1}{2},k}} - \mathbf{f}_{\mathbf{d}_{i,j,k+\frac{1}{2}}} + \mathbf{f}_{\mathbf{d}_{i,j,k-\frac{1}{2}}} \ . \tag{4.54}$$

Linearizing the artificial dissipation terms consequently provides the sensitivities

$$\delta\boldsymbol{\mathcal{F}}_{\mathbf{d}_{i,j,k}} = -\delta\mathbf{f}_{\mathbf{d}_{i+\frac{1}{2},j,k}} + \delta\mathbf{f}_{\mathbf{d}_{i-\frac{1}{2},j,k}} - \delta\mathbf{f}_{\mathbf{d}_{i,j+\frac{1}{2},k}} + \delta\mathbf{f}_{\mathbf{d}_{i,j-\frac{1}{2},k}} - \delta\mathbf{f}_{\mathbf{d}_{i,j,k+\frac{1}{2}}} + \delta\mathbf{f}_{\mathbf{d}_{i,j,k-\frac{1}{2}}} \ . \tag{4.55}$$

The artificial dissipation scheme employed in the flow solver is a blend of adaptive second- and fourth-order differences, cf. subsection 3.3.2. The variation of the discrete artificial dissipation flux may be written as

$$\delta\mathbf{f}_{\mathbf{d}_{i+\frac{1}{2},j,k}} = \delta\mathbf{f}^{(2)}_{\mathbf{d}_{i+\frac{1}{2},j,k}} - \delta\mathbf{f}^{(4)}_{\mathbf{d}_{i+\frac{1}{2},j,k}} \ , \tag{4.56}$$

with

$$\delta\mathbf{f}^{(2)}_{\mathbf{d}_{i+\frac{1}{2},j,k}} = \nu^{(2)}_{i+\frac{1}{2},j,k}\Lambda_{i+\frac{1}{2},j,k}(\delta\mathbf{W}_{i+1,j,k} - \delta\mathbf{W}_{i,j,k}) \ , \tag{4.57}$$

$$\delta\mathbf{f}^{(4)}_{\mathbf{d}_{i+\frac{1}{2},j,k}} = \nu^{(4)}_{i+\frac{1}{2},j,k}\Lambda_{i+\frac{1}{2},j,k}(\delta\mathbf{W}_{i+2,j,k} - 3\delta\mathbf{W}_{i+1,j,k} + 3\delta\mathbf{W}_{i,j,k} - \delta\mathbf{W}_{i-1,j,k}) \ , \tag{4.58}$$

where the variations of the pressure sensor terms $\nu^{(2)}$ and $\nu^{(4)}$ as well as the sensitivities of the spectral radius $\Lambda$ are neglected. Although the sensor terms and the spectral radii are functions of the state vector and thus will generally contribute to the adjoint equation, in this work these terms are treated as constants and their variations are ignored since the magnitude of the dissipative terms is lower than those of the convective and viscous fluxes. The dissipative flux sensitivities at the cell faces $i-\frac{1}{2}$, $j\pm\frac{1}{2}$, and $k\pm\frac{1}{2}$ are calculated similarly as the sensitivity at $i+\frac{1}{2}$ in Eqn. (4.58).

The stencil of the artificial dissipation scheme spans over two cells in each direction, therefore, requiring information from a total of thirteen cells. Accordingly, the linearized artificial dissipation flux $\delta\boldsymbol{\mathcal{F}}_{\mathbf{d}}$ at cell $(i,j,k)$ contributes to the adjoint fluxes of the cells $(i,j,k)$, $(i\pm1,j,k)$, $(i\pm2,j,k)$, $(i,j\pm1,k)$, $(i,j\pm2,k)$, $(i,j,k\pm1)$, and $(i,j,k\pm2)$, while the corresponding dissipative adjoint flux $\boldsymbol{\mathfrak{F}}_{\mathbf{d}}$ at cell $(i,j,k)$ has contributions from the artificial dissipation scheme of the same thirteen cells, i.e.

$$\begin{aligned}
\boldsymbol{\mathfrak{F}}_{\mathbf{d}}(\boldsymbol{\psi})_{i,j,k} \;=\;& \left[\frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\right]^T_{i-2,j,k}\boldsymbol{\psi}_{i-2,j,k} + \left[\frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\right]^T_{i-1,j,k}\boldsymbol{\psi}_{i-1,j,k} + \left[\frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\right]^T_{i,j,k}\boldsymbol{\psi}_{i,j,k} \\
&+ \left[\frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\right]^T_{i+1,j,k}\boldsymbol{\psi}_{i+1,j,k} + \left[\frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\right]^T_{i+2,j,k}\boldsymbol{\psi}_{i+2,j,k} + \left[\frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\right]^T_{i,j-2,k}\boldsymbol{\psi}_{i,j-2,k} \\
&+ \left[\frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\right]^T_{i,j-1,k}\boldsymbol{\psi}_{i,j-1,k} + \left[\frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\right]^T_{i,j+1,k}\boldsymbol{\psi}_{i,j+1,k} + \left[\frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\right]^T_{i,j+2,k}\boldsymbol{\psi}_{i,j+2,k} \\
&+ \left[\frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\right]^T_{i,j,k-2}\boldsymbol{\psi}_{i,j,k-2} + \left[\frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\right]^T_{i,j,k-1}\boldsymbol{\psi}_{i,j,k-1} + \left[\frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\right]^T_{i,j,k+1}\boldsymbol{\psi}_{i,j,k+1} \\
&+ \left[\frac{\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}}{\partial\mathbf{W}}\right]^T_{i,j,k+2}\boldsymbol{\psi}_{i,j,k+2}\;. \qquad (4.59)
\end{aligned}$$

Substituting Eqns. (4.56)-(4.58) into each term $\partial\boldsymbol{\mathcal{F}}_{\mathbf{d}}/\partial\mathbf{W}$ of Eqn. (4.59) and discarding all terms that are not functions of the state variable sensitivity $\delta\mathbf{W}_{i,j,k}$ provides the contribution from the artificial dissipation flux to the discrete adjoint equations

$$\begin{aligned}
\boldsymbol{\mathfrak{F}}_{\mathbf{d}}(\boldsymbol{\psi})_{i,j,k} \;=\;& \left[\nu^{(4)}_{\Lambda,i-\frac{3}{2},j,k}\right]^T\boldsymbol{\psi}_{i-2,j,k} - \left[\nu^{(2)}_{\Lambda,i-\frac{1}{2},j,k} + 3\nu^{(4)}_{\Lambda,i-\frac{1}{2},j,k} + \nu^{(4)}_{\Lambda,i-\frac{3}{2},j,k}\right]^T\boldsymbol{\psi}_{i-1,j,k} \\
&+ \left[\nu^{(4)}_{\Lambda,i+\frac{3}{2},j,k}\right]^T\boldsymbol{\psi}_{i+2,j,k} - \left[\nu^{(2)}_{\Lambda,i+\frac{1}{2},j,k} + 3\nu^{(4)}_{\Lambda,i+\frac{1}{2},j,k} + \nu^{(4)}_{\Lambda,i+\frac{3}{2},j,k}\right]^T\boldsymbol{\psi}_{i+1,j,k} \\
&+ \left[\nu^{(4)}_{\Lambda,i,j-\frac{3}{2},k}\right]^T\boldsymbol{\psi}_{i,j-2,k} - \left[\nu^{(2)}_{\Lambda,i,j-\frac{1}{2},k} + 3\nu^{(4)}_{\Lambda,i,j-\frac{1}{2},k} + \nu^{(4)}_{\Lambda,i,j-\frac{3}{2},k}\right]^T\boldsymbol{\psi}_{i,j-1,k} \\
&+ \left[\nu^{(4)}_{\Lambda,i,j+\frac{3}{2},k}\right]^T\boldsymbol{\psi}_{i,j+2,k} - \left[\nu^{(2)}_{\Lambda,i,j+\frac{1}{2},k} + 3\nu^{(4)}_{\Lambda,i,j+\frac{1}{2},k} + \nu^{(4)}_{\Lambda,i,j+\frac{3}{2},k}\right]^T\boldsymbol{\psi}_{i,j+1,k} \\
&+ \left[\nu^{(4)}_{\Lambda,i,j,k-\frac{3}{2}}\right]^T\boldsymbol{\psi}_{i,j,k-2} - \left[\nu^{(2)}_{\Lambda,i,j,k-\frac{1}{2}} + 3\nu^{(4)}_{\Lambda,i,j,k-\frac{1}{2}} + \nu^{(4)}_{\Lambda,i,j,k-\frac{3}{2}}\right]^T\boldsymbol{\psi}_{i,j,k-1} \\
&+ \left[\nu^{(4)}_{\Lambda,i,j,k+\frac{3}{2}}\right]^T\boldsymbol{\psi}_{i,j,k+2} - \left[\nu^{(2)}_{\Lambda,i,j,k+\frac{1}{2}} + 3\nu^{(4)}_{\Lambda,i,j,k+\frac{1}{2}} + \nu^{(4)}_{\Lambda,i,j,k+\frac{3}{2}}\right]^T\boldsymbol{\psi}_{i,j,k+1} \\
&+ \left[\nu^{(2)}_{\Lambda,i+\frac{1}{2},j,k} + 3\nu^{(4)}_{\Lambda,i+\frac{1}{2},j,k} + \nu^{(2)}_{\Lambda,i-\frac{1}{2},j,k} + 3\nu^{(4)}_{\Lambda,i-\frac{1}{2},j,k}\right]^T\boldsymbol{\psi}_{i,j,k}\;, \qquad (4.60)
\end{aligned}$$

with the abbreviations $\nu_\Lambda^{(2)} = \nu^{(2)}\Lambda$ and $\nu_\Lambda^{(4)} = \nu^{(4)}\Lambda$. Regrouping the terms in Eqn. (4.60) yields the final discrete adjoint artificial dissipation flux

$$\begin{aligned}
\mathfrak{F}_{\mathbf{d}}(\boldsymbol{\psi})_{i,j,k} \;=\;& \left[\nu^{(4)}_{\Lambda,i+\frac{3}{2},j,k}\right]^T (\boldsymbol{\psi}_{i+2,j,k} - \boldsymbol{\psi}_{i+1,j,k}) + \left[\nu^{(4)}_{\Lambda,i-\frac{3}{2},j,k}\right]^T (\boldsymbol{\psi}_{i-2,j,k} - \boldsymbol{\psi}_{i-1,j,k}) \\
&- \left[\nu^{(2)}_{\Lambda,i+\frac{1}{2},j,k} + 3\nu^{(4)}_{\Lambda,i+\frac{1}{2},j,k}\right]^T (\boldsymbol{\psi}_{i+1,j,k} - \boldsymbol{\psi}_{i,j,k}) \\
&- \left[\nu^{(2)}_{\Lambda,i-\frac{1}{2},j,k} + 3\nu^{(4)}_{\Lambda,i-\frac{1}{2},j,k}\right]^T (\boldsymbol{\psi}_{i-1,j,k} - \boldsymbol{\psi}_{i,j,k}) \\
&+ \left[\nu^{(4)}_{\Lambda,i,j+\frac{3}{2},k}\right]^T (\boldsymbol{\psi}_{i,j+2,k} - \boldsymbol{\psi}_{i,j+1,k}) + \left[\nu^{(4)}_{\Lambda,i,j-\frac{3}{2},k}\right]^T (\boldsymbol{\psi}_{i,j-2,k} - \boldsymbol{\psi}_{i,j-1,k}) \\
&- \left[\nu^{(2)}_{\Lambda,i,j+\frac{1}{2},k} + 3\nu^{(4)}_{\Lambda,i,j+\frac{1}{2},k}\right]^T (\boldsymbol{\psi}_{i,j+1,k} - \boldsymbol{\psi}_{i,j,k}) \\
&- \left[\nu^{(2)}_{\Lambda,i,j-\frac{1}{2},k} + 3\nu^{(4)}_{\Lambda,i,j-\frac{1}{2},k}\right]^T (\boldsymbol{\psi}_{i,j-1,k} - \boldsymbol{\psi}_{i,j,k}) \\
&+ \left[\nu^{(4)}_{\Lambda,i,j,k+\frac{3}{2}}\right]^T (\boldsymbol{\psi}_{i,j,k+2} - \boldsymbol{\psi}_{i,j,k+1}) + \left[\nu^{(4)}_{\Lambda,i,j,k-\frac{3}{2}}\right]^T (\boldsymbol{\psi}_{i,j,k-2} - \boldsymbol{\psi}_{i,j,k-1}) \\
&- \left[\nu^{(2)}_{\Lambda,i,j,k+\frac{1}{2}} + 3\nu^{(4)}_{\Lambda,i,j,k+\frac{1}{2}}\right]^T (\boldsymbol{\psi}_{i,j,k+1} - \boldsymbol{\psi}_{i,j,k}) \\
&- \left[\nu^{(2)}_{\Lambda,i,j,k-\frac{1}{2}} + 3\nu^{(4)}_{\Lambda,i,j,k-\frac{1}{2}}\right]^T (\boldsymbol{\psi}_{i,j,k-1} - \boldsymbol{\psi}_{i,j,k}) \;. \qquad (4.61)
\end{aligned}$$

## 4.3 Discrete Adjoint Boundary Conditions

If fully linearized, the discrete adjoint equations derived in the previous section represent the exact counterpart to the discretized field equations. However, to obtain physically correct boundary fluxes, the standard numerical scheme used in the turbomachinery RANS solver is modified at the boundaries of a computational domain, cf. subsection 3.3.3. First, boundary conditions are applied, which define the state vector at the boundary $\mathbf{W}_\mathrm{B}$ as a function of the flow variables in the computational domain $\mathbf{W}_\mathrm{D}$ and, at the inlet and outlet boundary, as a function of some prescribed constant quantities $q_\infty$, thus $\mathbf{W}_\mathrm{B} = \mathbf{W}_\mathrm{B}(\mathbf{W}_\mathrm{D}, q_\infty)$. Second, the evaluation of the convective and viscous fluxes using the boundary states $\mathbf{W}_\mathrm{B}$ then provides the physically correct boundary flux $\boldsymbol{\mathcal{F}}_\mathrm{B}$, hence $\boldsymbol{\mathcal{F}}_\mathrm{B} = \boldsymbol{\mathcal{F}}_\mathrm{B}(\mathbf{W}_\mathrm{D}, q_\infty)$. Third, at boundary faces the artificial dissipation flux is set to zero.

Recall, to obtain the exact discrete adjoint flux of a particular domain cell $(i,j,k)$, it is necessary to fully linearize the residual and to collect all terms that are functions of the state vector at cell $(i,j,k)$. If the discretized boundary flux is a function of the state at cell $(i,j,k)$, then the linearized boundary flux $\partial\boldsymbol{\mathcal{F}}_{\mathrm{B}}/\partial\mathbf{W}_{\mathrm{D}}$ consequently contributes towards the total adjoint flux of cell $(i,j,k)$ and produces the adjoint boundary flux $[\partial\boldsymbol{\mathcal{F}}_{\mathrm{B}}/\partial\mathbf{W}_{\mathrm{D}}]^T\boldsymbol{\psi}_{\mathrm{D}}$. It is the author's experience that the inclusion of the adjoint boundary fluxes are vital and that an inaccurate implementation of the adjoint boundary terms can significantly corrupt the adjoint solution and, thus, the objective function gradient. The crucial step in calculating the adjoint boundary flux is the exact linearization of the boundary flux with respect to the states in the domain it depends on. Using the chain rule, the linearized boundary flux can be expanded into

$$\frac{\partial\boldsymbol{\mathcal{F}}_{\mathrm{B}}}{\partial\mathbf{W}_{\mathrm{D}}}=\frac{\partial\boldsymbol{\mathcal{F}}_{\mathrm{B}}}{\partial\mathbf{W}_{\mathrm{B}}}\frac{\partial\mathbf{W}_{\mathrm{B}}}{\partial\mathbf{W}_{\mathrm{D}}} \tag{4.62}$$

and, thus, can be written as the product of the flux Jacobian matrix $\mathbf{A}_{\mathrm{B}}=\partial\boldsymbol{\mathcal{F}}_{\mathrm{B}}/\partial\mathbf{W}_{\mathrm{B}}$, which is evaluated using the boundary state $\mathbf{W}_{\mathrm{B}}$, and the transformation matrix

$$\boldsymbol{\mathcal{T}}_{\mathrm{BD}}=\frac{\partial\mathbf{W}_{\mathrm{B}}}{\partial\mathbf{W}_{\mathrm{D}}}, \tag{4.63}$$

which represents the linearized flow boundary condition. While the formulation of the flux Jacobian is identical for any type of boundary condition, the transformation matrix $\boldsymbol{\mathcal{T}}_{\mathrm{BD}}$ is boundary-type-specific. To simplify its derivation, it is desirable to construct the transformation matrix as a composition of several matrices and elementary transformations.

The remaining part of this section presents the development of the transformation matrices and, hence, flow-consistent adjoint boundary fluxes for the various discrete boundary conditions employed in the turbomachinery RANS solver.

### 4.3.1 Solid Wall Boundary Conditions

In this subsection the transformation matrices for the inviscid and viscous wall boundary conditions are derived.

In both cases it is straightforward to write the state vector at the boundary as a function of the flow variables in the domain. The transformation matrix is the

Jacobian matrix of this function. For an inviscid wall with slip boundary condition and a zeroth-order pressure extrapolation, cf. Eqn. (3.66), the transformation matrix is given by

$$\boldsymbol{\mathcal{T}}_{\mathrm{BD}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -n_1\Omega_1(x_3n_2 - x_2n_3) & 1 - n_1n_1 & -\,n_1n_2 & -\,n_1n_3 & 0 \\ -n_2\Omega_1(x_3n_2 - x_2n_3) & -\,n_2n_1 & 1 - n_2n_2 & -\,n_2n_3 & 0 \\ -n_3\Omega_1(x_3n_2 - x_2n_3) & -\,n_3n_1 & -\,n_3n_2 & 1 - n_3n_3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{4.64}$$

For an adiabatic, viscous wall with no-slip condition and a zeroth-order pressure extrapolation, cf. Eqn. (3.67), the transformation matrix reads as

$$\boldsymbol{\mathcal{T}}_{\mathrm{BD}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\Omega_1x_3 & 0 & 0 & 0 & 0 \\ \Omega_1x_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{4.65}$$

Multiplying the inviscid or viscous transformation matrix with the flux Jacobian evaluated at the boundary, taking the transpose of the product, and then multiplying the resulting matrix with the adjoint solution in the domain provides the flow-consistent adjoint boundary flux for a solid wall boundary.

### 4.3.2 Inlet and Outlet Boundary Conditions

To obtain physically correct boundary fluxes at an inlet or outlet, characteristic-based boundary conditions using Riemann invariants are applied in the flow solver, cf. subsection 3.3.3. The corresponding transformation matrix is obtained by linearizing these imposed flow boundary conditions. Since all boundary quantities are expressed in terms of the domain states through local relations only, the derivation of $\boldsymbol{\mathcal{T}}_{\mathrm{BD}}$ is lengthy but relatively simple.

At a subsonic inlet, the outgoing Riemann invariant $Ri$, the prescribed total temperature $T_{t,\mathrm{in}}$, and the circumferential and radial inflow angles $\alpha_\varphi$ and $\alpha_r$ are combined to form a quadratic equation which solves for the speed of sound $c_{\mathrm{in}}$. The remaining boundary quantities are then determined from the prescribed inlet quantities, the isentropic relations, and the equation of state. To obtain the transformation matrix $\boldsymbol{\mathcal{T}}_{\mathrm{BD}}$ at the inlet boundary, first the outgoing Riemann invariant, cf. Eqn. (3.69), is linearized, providing

$$\delta Ri = \frac{\partial Ri}{\partial \mathbf{W}_{\mathrm{D}}} = \left[\begin{array}{ccl} -\dfrac{u_1}{\rho} & + & \dfrac{c}{p}\left(E - |U|^2\right) \\ \dfrac{1}{\rho} & + & \dfrac{c}{p} u_1 \\ & & \dfrac{c}{p} u_2 \\ & & \dfrac{c}{p} u_3 \\ & - & \dfrac{c}{p} \end{array}\right]_{\mathrm{D}} , \tag{4.66}$$

where all quantities are calculated from the domain states. Afterwards, the variation of the speed of sound $\delta c_{\mathrm{in}}$, cf. Eqns. (3.75) and (3.76), is calculated utilizing the chain rule, i.e.

$$\delta c_{\mathrm{in}} = \frac{\partial c_{\mathrm{in}}}{\partial \mathbf{W}_{\mathrm{D}}} = \frac{\partial c_{\mathrm{in}}}{\partial Ri} \delta Ri \ , \tag{4.67}$$

with

$$\frac{\partial c_{\mathrm{in}}}{\partial Ri} = \frac{-1 - \dfrac{(\gamma-1)Ri}{2\alpha_x}\left[\sqrt{\dfrac{\gamma R T_{t,\mathrm{in}}}{\alpha_x}\left(\dfrac{2}{\gamma-1} + \dfrac{1}{\alpha_x}\right) - \dfrac{(\gamma-1)Ri^2}{2\alpha_x}}\right]^{-1}}{\dfrac{2}{\gamma-1} + \dfrac{1}{\alpha_x}} , \tag{4.68}$$

where

$$\alpha_x = 1 + \tan^2 \alpha_\varphi + \frac{\tan^2 \alpha_r}{\cos^2 \alpha_\varphi} \ . \tag{4.69}$$

With this information and the relations of Eqn. (3.77), the variation of the inlet temperature $T_{\mathrm{in}}$ and the linearized inlet velocity $|U|_{\mathrm{in}}$ are straightforward to obtain, namely

$$\delta T_{\mathrm{in}} = \frac{\partial T_{\mathrm{in}}}{\partial \mathbf{W}_{\mathrm{D}}} = \frac{\partial T_{\mathrm{in}}}{\partial c_{\mathrm{in}}} \delta c_{\mathrm{in}} \ , \tag{4.70}$$

where

$$\frac{\partial T_{\text{in}}}{\partial c_{\text{in}}} = \frac{2c_{\text{in}}}{\gamma R} \ , \tag{4.71}$$

and

$$\delta|\mathbf{U}|_{\text{in}} = \frac{\partial|\mathbf{U}|_{\text{in}}}{\partial \mathbf{W}_{\text{D}}} = \frac{\partial|\mathbf{U}|_{\text{in}}}{\partial T_{\text{in}}}\delta T_{\text{in}} \ , \tag{4.72}$$

with

$$\frac{\partial|\mathbf{U}|_{\text{in}}}{\partial T_{\text{in}}} = -\sqrt{\frac{c_p}{2(T_{t,\text{in}} - T_{\text{in}})}} \ . \tag{4.73}$$

Utilizing the chain rule again provides the variations of the individual velocity components

$$\delta u_{1,\text{in}} = \frac{\partial u_{1,\text{in}}}{\partial \mathbf{W}_{\text{D}}} = \frac{\partial u_{1,\text{in}}}{\partial|\mathbf{U}|_{\text{in}}}\delta|\mathbf{U}|_{\text{in}} = \cos\alpha_r \cos\alpha_\varphi \delta|\mathbf{U}|_{\text{in}} \ , \tag{4.74}$$

$$\delta u_{2,\text{in}} = \frac{\partial u_{2,\text{in}}}{\partial \mathbf{W}_{\text{D}}} = \frac{\partial u_{2,\text{in}}}{\partial|\mathbf{U}|_{\text{in}}}\delta|\mathbf{U}|_{\text{in}} = (\sin\alpha_r \cos\varphi - \cos\alpha_r \sin\alpha_\varphi \sin\varphi)\delta|\mathbf{U}|_{\text{in}} \ , \tag{4.75}$$

$$\delta u_{3,\text{in}} = \frac{\partial u_{3,\text{in}}}{\partial \mathbf{W}_{\text{D}}} = \frac{\partial u_{3,\text{in}}}{\partial|\mathbf{U}|_{\text{in}}}\delta|\mathbf{U}|_{\text{in}} = (\sin\alpha_r \sin\varphi + \cos\alpha_r \sin\alpha_\varphi \cos\varphi)\delta|\mathbf{U}|_{\text{in}} \ , \tag{4.76}$$

as well as the sensitivity of the inlet pressure

$$\delta p_{\text{in}} = \frac{\partial p_{\text{in}}}{\partial \mathbf{W}_{\text{D}}} = \frac{\partial p_{\text{in}}}{\partial T_{\text{in}}}\delta T_{\text{in}} = \frac{\gamma}{\gamma - 1}\frac{p_{t,\text{in}}}{T_{t,\text{in}}}\left(\frac{T_{\text{in}}}{T_{t,\text{in}}}\right)^{\frac{1}{\gamma-1}}\delta T_{\text{in}} \ , \tag{4.77}$$

and the density variation

$$\delta\rho_{\text{in}} = \frac{\partial \rho_{\text{in}}}{\partial \mathbf{W}_{\text{D}}} = \frac{\partial \rho_{\text{in}}}{\partial p_{\text{in}}}\delta p_{\text{in}} + \frac{\partial \rho_{\text{in}}}{\partial T_{\text{in}}}\delta T_{\text{in}} = \frac{1}{RT_{\text{in}}}\delta p_{\text{in}} - \frac{p_{\text{in}}}{RT_{\text{in}}^2}\delta T_{\text{in}} \ . \tag{4.78}$$

This completes the linearization of the primitive variables $\partial\mathbf{W_P}$ at the boundary with respect to the domain states, $\partial\mathbf{W}_{\mathbf{P},\text{B}}/\partial\mathbf{W}_{\text{D}}$. The transformation matrix can now be constructed through the matrix multiplication

$$\boldsymbol{\mathcal{T}}_{\text{BD}} = \frac{\partial \mathbf{W}_{\text{B}}}{\partial \mathbf{W}_{\mathbf{P},\text{B}}}\frac{\partial \mathbf{W}_{\mathbf{P},\text{B}}}{\partial \mathbf{W}_{\text{D}}} \tag{4.79}$$

where $\partial\mathbf{W}_{\text{B}}/\partial\mathbf{W}_{\mathbf{P},\text{B}}$ is the transformation matrix $\mathbf{M}$ provided in Eqn. (A.3), i.e.

$$\delta w_{1,\text{B}} = \delta\rho_{\text{in}} \ , \tag{4.80a}$$

$$\delta w_{2,\mathrm{B}} = \rho_{\mathrm{in}} \delta u_{1,\mathrm{in}} + u_{1,\mathrm{in}} \delta w_{1,\mathrm{B}} \ , \tag{4.80b}$$

$$\delta w_{3,\mathrm{B}} = \rho_{\mathrm{in}} \delta u_{2,\mathrm{in}} + u_{2,\mathrm{in}} \delta w_{2,\mathrm{B}} \ , \tag{4.80c}$$

$$\delta w_{4,\mathrm{B}} = \rho_{\mathrm{in}} \delta u_{3,\mathrm{in}} + u_{3,\mathrm{in}} \delta w_{3,\mathrm{B}} \ , \tag{4.80d}$$

$$\delta w_{5,\mathrm{B}} = E_{\mathrm{in}} \delta \rho_{\mathrm{in}} + \rho_{\mathrm{in}} \left[ \frac{R}{\gamma - 1} \delta T_{\mathrm{in}} + u_{1,\mathrm{in}} \delta u_{1,\mathrm{in}} + u_{2,\mathrm{in}} \delta u_{2,\mathrm{in}} + u_{3,\mathrm{in}} \delta u_{3,\mathrm{in}} \right] \ . \tag{4.80e}$$

At a subsonic outlet, the entropy is extrapolated from the interior domain, which provides the density at the boundary. Together with the information obtained from the outgoing Riemann invariant and the prescribed static pressure, this allows for a calculation of the missing flow variables. The transformation matrix is constructed in a similar fashion as at the inlet. First, the outgoing Riemann invariant, cf. Eqn. (3.85), and the density formulation, cf. Eqn. (3.83), are linearized, providing

$$\delta Ri = \frac{\partial Ri}{\partial \mathbf{W}_{\mathrm{D}}} = \left[ \begin{array}{ccl} -\frac{u_1}{\rho} & - & \frac{c}{p} \left( E - |\mathbf{U}|^2 \right) \\ \frac{1}{\rho} & - & \frac{c}{p} u_1 \\ & - & \frac{c}{p} u_2 \\ & - & \frac{c}{p} u_3 \\ & & \frac{c}{p} \end{array} \right]_{\mathrm{D}} \ , \tag{4.81}$$

and

$$\delta \rho_{\mathrm{out}} = \frac{\partial \rho_{\mathrm{out}}}{\partial \mathbf{W}_{\mathrm{D}}} = \left( \frac{p_{\mathrm{out}}}{p_{\mathrm{D}}} \right)^{\frac{1}{\gamma}} \delta \rho_{\mathrm{D}} - \frac{\rho_{\mathrm{D}}}{\gamma p_{\mathrm{out}}} \left( \frac{p_{\mathrm{out}}}{p_{\mathrm{D}}} \right)^{\frac{\gamma+1}{\gamma}} \left[ \delta p_{\mathrm{D}} - \frac{p_{\mathrm{D}}}{p_{\mathrm{out}}} \delta p_{\mathrm{rad}} \right] \ , \tag{4.82}$$

where

$$\delta \rho_{\mathrm{D}} = \frac{\partial \rho_{\mathrm{D}}}{\partial \mathbf{W}_{\mathrm{D}}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}_{\mathrm{D}} \quad \text{and} \quad \delta p_{\mathrm{D}} = \frac{\partial p_{\mathrm{D}}}{\partial \mathbf{W}_{\mathrm{D}}} = (\gamma - 1) \begin{bmatrix} \frac{|\mathbf{U}|^2}{2} \\ -u_1 \\ -u_2 \\ -u_3 \\ 1 \end{bmatrix}_{\mathrm{D}} \ , \tag{4.83}$$

and $\delta p_{\text{rad}}$ is either null or, in case the radial equilibrium boundary condition is applied,

$$\delta p_{\text{rad}} = \frac{\partial p_{\text{rad}}}{\partial \mathbf{W}_{\text{D}}} = \left(\frac{r}{r_{\text{Hub}}} - 1\right) \begin{bmatrix} u_\varphi^2 \\ 0 \\ -\ 2u_\varphi \sin\varphi \\ 2u_\varphi \cos\varphi \\ 0 \end{bmatrix}_{\text{D}} . \tag{4.84}$$

Thereafter, the variation of the speed of sound can be determined

$$\delta c_{\text{out}} = \frac{\partial c_{\text{out}}}{\partial \mathbf{W}_{\text{D}}} = \frac{1}{2}\sqrt{\frac{\gamma}{\rho_{\text{out}} p_{\text{out}}}} \left[-\frac{p_{\text{out}}}{\rho_{\text{out}}}\delta\rho_{\text{out}} + \delta p_{\text{rad}}\right] , \tag{4.85}$$

and the individual velocity components can be linearized. The sensitivity of the velocity component normal to the outlet boundary may be calculated from

$$\delta u_{1,\text{out}} = \frac{\partial u_{1,\text{out}}}{\partial \mathbf{W}_{\text{D}}} = \delta Ri - \frac{2}{\gamma - 1}\delta c_{\text{out}} , \tag{4.86}$$

while the velocity components tangential to the outlet plane are straightforward to obtain since they are simply extrapolated from the interior domain, i.e.

$$\delta u_{2,\text{out}} = \frac{\partial u_{2,\text{out}}}{\partial \mathbf{W}_{\text{D}}} = \frac{1}{\rho} \begin{bmatrix} -u_2 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}_{\text{D}} \quad \text{and} \quad \delta u_{3,\text{out}} = \frac{\partial u_{3,\text{out}}}{\partial \mathbf{W}_{\text{D}}} = \frac{1}{\rho} \begin{bmatrix} -u_3 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}_{\text{D}} . \tag{4.87}$$

The temperature variation is obtained from the relation described in Eqn. (4.70). Finally, $\boldsymbol{\mathcal{T}}_{\text{BD}}$ can be constructed through the same matrix multiplication employed at the inlet boundary, cf. Eqns. (4.79) and (4.80).

In case of supersonic inflow, all characteristics are incoming, thus, the boundary states are calculated from the prescribed inlet quantities only. Consequently, the transformation matrix equals zero. At a supersonic outlet boundary, where all characteristics are outgoing, the transformation matrix $\boldsymbol{\mathcal{T}}_{\text{BD}}$ equals the identity matrix.

### 4.3.3 Periodic Boundary Condition

The discrete adjoint periodic boundary conditions are straightforward to implement. In case of translational periodicity a periodic boundary is equivalent to a block-to-block boundary and both scalar and vector quantities are simply copied from the domain cell next to a periodic boundary into the halo cell of the other corresponding periodic boundary, cf. subsection 3.3.3. The adjoint variables are treated accordingly. In case of rotational periodicity the rotation of the coordinate system has to be accounted for and the transfer of the adjoint variables follows the procedure described in Eqn. (3.89), thus

$$\begin{aligned}
\psi_{1,\mathrm{H,II}} =& \quad \psi_{1,\mathrm{D,I}} \ , && (4.88a)\\
\psi_{2,\mathrm{H,II}} =& \quad \psi_{2,\mathrm{D,I}} \ , && (4.88b)\\
\psi_{3,\mathrm{H,II}} =& \quad (\psi_{3,\mathrm{D,I}} \cos\varphi_\mathrm{I} + \psi_{4,\mathrm{D,I}} \sin\varphi_\mathrm{I}) \cos\varphi_\mathrm{II} \\
& + (\psi_{3\mathrm{D,I}} \sin\varphi_\mathrm{I} - \psi_{4,\mathrm{D,I}} \cos\varphi_\mathrm{I}) \sin\varphi_\mathrm{II} \ , && (4.88c)\\
\psi_{4,\mathrm{H,II}} =& \quad (\psi_{3,\mathrm{D,I}} \cos\varphi_\mathrm{I} + \psi_{4,\mathrm{D,I}} \sin\varphi_\mathrm{I}) \sin\varphi_\mathrm{II} \\
& - (\psi_{3,\mathrm{D,I}} \sin\varphi_\mathrm{I} - \psi_{4,\mathrm{D,I}} \cos\varphi_\mathrm{I}) \cos\varphi_\mathrm{II} \ , && (4.88d)\\
\psi_{5,\mathrm{H,II}} =& \quad \psi_{5,\mathrm{D,I}} \ , && (4.88e)
\end{aligned}$$

where the subscripts $H$, $D$, $I$, and $II$ correspond to the cell indexing used in Figure 3.8. This procedure ensures that the adjoint fluxes at two corresponding periodic boundaries are identical.

### 4.3.4 Block-to-Block Boundaries

The treatment of the adjoint variables at a block-to-block boundary is identical to the treatment of the flow variables at a block-to-block interface, cf. subsection 3.3.3. The blocks are extended by two additional levels of cells at each boundary face. These halo cells possess the same geometric properties as the first two domain cells of the neighbouring block, which ensures that the domain and halo cells perfectly overlap. After every stage of the Runge-Kutta time-stepping scheme, the adjoint variables are then exchanged between two neighbouring blocks by copying the domain states of one block into the overlapping halo cells of the adjacent block. This boundary treatment ensures that the correct adjoint boundary fluxes are obtained at a block-to-block interface. Similarly as in the flow solver, the adjoint solver uses MPI standard for communication between multiple processors.

### 4.3.5 Rotor-Stator Interface

This subsection discusses the development of the exact adjoint boundary condition for Chima's steady averaging-plane approach [53] described in subsection 3.3.3. Following this approach, mixed-out circumferential averages of the primitive variables are first calculated from all cells at a constant radius on both sides of an interface and are used as boundary values in the neighbouring domain. The differences between the local domain values next to the boundary and the boundary values are then treated as characteristic perturbations and Giles' 1D characteristic theory [54] is applied to drive the differences to zero. Hence, the mixing-plane boundary condition involves nonlocal quantities which make a boundary state depend on many interior cells. The state vector at a boundary is a function of the state vector in its adjacent domain cell and the mixed-out averages from the neighbouring domain. Figure 4.1 shows a schematic mixing-plane interface and illustrates the dependency of the boundary states. The state in the red rotor boundary cell is a function of all red domain cells. Thus,

$$\mathbf{W}_{\mathrm{B},R} = \mathbf{W}_{\mathrm{B},R}\Big(\mathbf{W}_{\mathrm{D},R}, \widetilde{\mathbf{W}}_S\Big) = \mathbf{W}_{\mathrm{B},R}\Big(\mathbf{W}_{\mathrm{D},R}, \sum_j \mathbf{W}_{\mathrm{D},S,j}\Big) \ , \tag{4.89}$$

where $R$ and $S$ represent respectively rotor and stator quantities and $\widetilde{\mathbf{W}}$ is the state vector built of mixed-out primitive variables. $\sum_j$ implies summation over all cells located at the same radial section of the mixing-plane interface in the neighbouring domain. Accordingly, the state in the green stator boundary cell depends on all green domain cells, hence

$$\mathbf{W}_{\mathrm{B},S} = \mathbf{W}_{\mathrm{B},S}\Big(\mathbf{W}_{\mathrm{D},S}, \widetilde{\mathbf{W}}_R\Big) = \mathbf{W}_{\mathrm{B},S}\Big(\mathbf{W}_{\mathrm{D},S}, \sum_j \mathbf{W}_{\mathrm{D},R,j}\Big) \ . \tag{4.90}$$

From Eqns. (4.89) and (4.90) it follows that the adjoint flux for a domain cell next to a rotor-stator interface has adjoint boundary flux contributions from both sides of

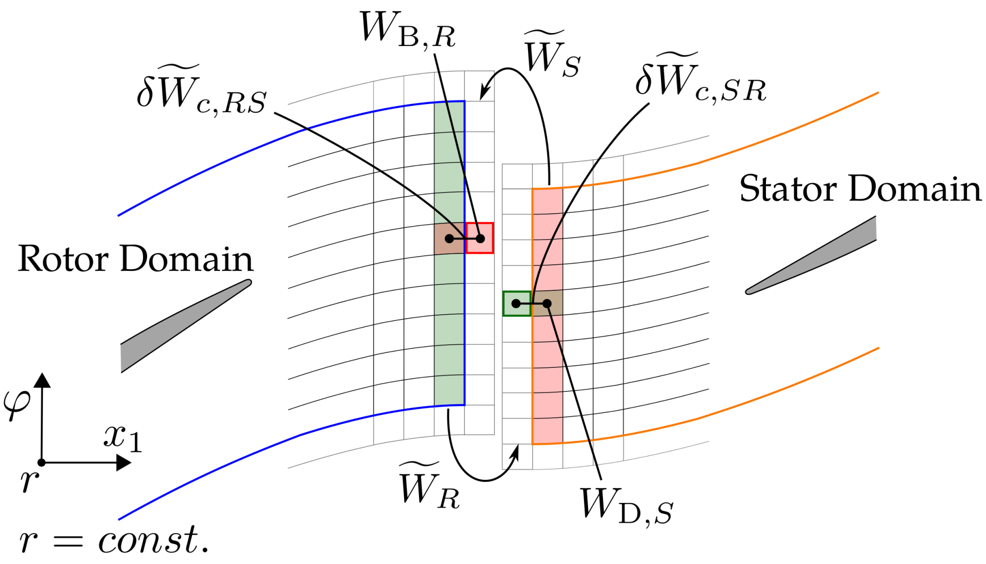

Figure 4.1: Dependency of the boundary states at a mixing-plane interface.

the mixing-plane, namely

$$\left(\mathbf{A}_{\mathrm{B},R}\frac{\partial \mathbf{W}_{\mathrm{B},R}}{\partial \mathbf{W}_{\mathrm{D},R}}\right)^T \boldsymbol{\psi}_{\mathrm{D},R} + \sum_j \left(\mathbf{A}_{\mathrm{B},S,j}\frac{\partial \mathbf{W}_{\mathrm{B},S,j}}{\partial \mathbf{W}_{\mathrm{D},R}}\right)^T \boldsymbol{\psi}_{\mathrm{D},S,j} \tag{4.91}$$

for a domain cell on the rotor side and

$$\left(\mathbf{A}_{\mathrm{B},S}\frac{\partial \mathbf{W}_{\mathrm{B},S}}{\partial \mathbf{W}_{\mathrm{D},S}}\right)^T \boldsymbol{\psi}_{\mathrm{D},S} + \sum_j \left(\mathbf{A}_{\mathrm{B},R,j}\frac{\partial \mathbf{W}_{\mathrm{B},R,j}}{\partial \mathbf{W}_{\mathrm{D},S}}\right)^T \boldsymbol{\psi}_{\mathrm{D},R,j} \tag{4.92}$$

in case of a domain cell on the stator side. The various boundary states contributing to the adjoint flux of both a rotor and stator domain cell are visualized in Figure 4.2. The red-colored boundary states contribute to the adjoint flux of the red domain cell, while the green domain cell has contributions from all green-colored boundary states. In Eqns. (4.91) and (4.92), $\mathbf{A}_{\mathrm{B},R}$ and $\mathbf{A}_{\mathrm{B},S}$ are the flux Jacobians evaluated at the boundary, while

$$\boldsymbol{\mathcal{T}}_{\mathrm{BD},R} = \frac{\partial \mathbf{W}_{\mathrm{B},R}}{\partial \mathbf{W}_{\mathrm{D},R}} \quad \text{and} \quad \boldsymbol{\mathcal{T}}_{\mathrm{BD},SR,j} = \frac{\partial \mathbf{W}_{\mathrm{B},S,j}}{\partial \mathbf{W}_{\mathrm{D},R}}\ , \tag{4.93}$$

$$\boldsymbol{\mathcal{T}}_{\mathrm{BD},S} = \frac{\partial \mathbf{W}_{\mathrm{B},S}}{\partial \mathbf{W}_{\mathrm{D},S}} \quad \text{and} \quad \boldsymbol{\mathcal{T}}_{\mathrm{BD},RS,j} = \frac{\partial \mathbf{W}_{\mathrm{B},R,j}}{\partial \mathbf{W}_{\mathrm{D},S}}\ , \tag{4.94}$$

are the transformation matrices to be determined. Chima's averaging plane approach calculates the boundary states from

$$\mathbf{W}_{\mathrm{B},R} = \widetilde{\mathbf{W}}_S + \delta\widetilde{\mathbf{W}}_{\mathbf{c},RS} \ , \tag{4.95}$$

$$\mathbf{W}_{\mathrm{B},S} = \widetilde{\mathbf{W}}_R + \delta\widetilde{\mathbf{W}}_{\mathbf{c},SR} \ , \tag{4.96}$$

where $\widetilde{\mathbf{W}}$ is again the averaged conservative variable vector obtained from the mixed-out primitive variables and the boundary state correction $\delta\widetilde{\mathbf{W}}_{\mathbf{c}}$ is obtained through Giles' 1D characteristic theory. Linearizing Eqns. (4.95) and (4.96) provides the corresponding transformation matrices. In particular, the boundary contribution to the adjoint flux of a rotor domain cell next to a mixing-plane interface is specified through the transformation matrices

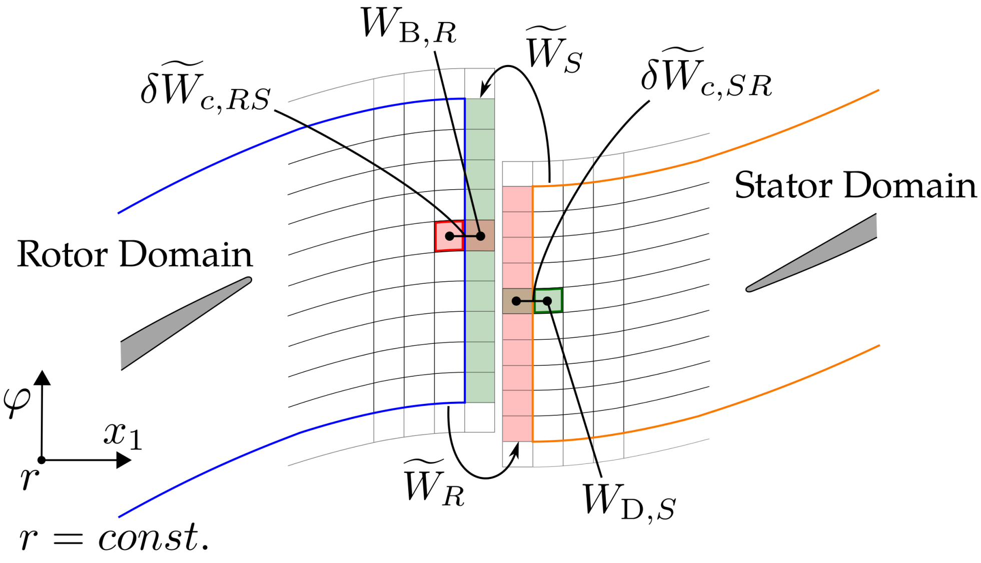

Figure 4.2: Boundary states contributing to the adjoint flux of a cell next to a mixing-plane interface.

$$\boldsymbol{\mathcal{T}}_{\mathrm{BD},R} = \frac{\partial\delta\widetilde{\mathbf{W}}_{\mathbf{c},RS}}{\partial\mathbf{W}_{\mathrm{D},R}} \qquad \text{and} \qquad \boldsymbol{\mathcal{T}}_{\mathrm{BD},SR,j} = \frac{\partial(\widetilde{\mathbf{W}}_{R,j} + \delta\widetilde{\mathbf{W}}_{\mathbf{c},SR,j})}{\partial\mathbf{W}_{\mathrm{D},R}} \ , \tag{4.97}$$

where the term $\widetilde{\mathbf{W}}_S$ appearing in Eqn. (4.95) could be dropped since it does not directly depend on the states in the rotor domain. The transformation matrices defining the adjoint boundary flux contribution to the adjoint flux of a stator domain cell next to a rotor-stator interface are given by

$$\boldsymbol{\mathcal{T}}_{\mathrm{BD},S} = \frac{\partial \delta \widetilde{\mathbf{W}}_{\mathbf{c},SR}}{\partial \mathbf{W}_{\mathrm{D},S}} \ , \qquad \text{and} \qquad \boldsymbol{\mathcal{T}}_{\mathrm{BD},RS,j} = \frac{\partial (\widetilde{\mathbf{W}}_{S,j} + \delta \widetilde{\mathbf{W}}_{\mathbf{c},RS,j})}{\partial \mathbf{W}_{\mathrm{D},S}} \ , \tag{4.98}$$

where here the term $\widetilde{\mathbf{W}}_R$ could be dropped since the variation $\partial \widetilde{\mathbf{W}}_R / \partial \mathbf{W}_{\mathrm{D},S}$ equals zero. All four terms $\delta \widetilde{\mathbf{W}}_{\mathbf{c},RS}$, $\widetilde{\mathbf{W}}_R$, $\delta \widetilde{\mathbf{W}}_{\mathbf{c},SR}$, and $\widetilde{\mathbf{W}}_S$ appearing in Eqns. (4.97) and (4.98) are functions of the mixed-out quantities. Hence, the calculation of the transformation matrices involves the linearization of the equations defining the mixed-out primitive variables. Furthermore, to obtain the linearization of $\delta \widetilde{\mathbf{W}}_{\mathbf{c},RS}$ and $\delta \widetilde{\mathbf{W}}_{\mathbf{c},SR}$ it is necessary to differentiate the 1D characteristics.

The further development of the adjoint boundary flux contribution is split up into two sections, in which the linearization of the individual terms will be discussed separately. First, the differentiation of the mixed-out flow variables $\widetilde{\mathbf{W}}_R$ and $\widetilde{\mathbf{W}}_S$ will be presented in more detail. Then, the linearization of Giles' 1D characteristics $\delta \widetilde{\mathbf{W}}_{\mathbf{c},RS}$ and $\delta \widetilde{\mathbf{W}}_{\mathbf{c},SR}$ will be derived.

### Linearization of the mixed-out flow variables

For the sake of brevity, in this subsection the indices associating quantities to a specific rotor or stator domain cell will be dropped. Henceforth, the linearization of the vector of mixed-out flow variables $\delta \widetilde{\mathbf{W}}$ represents both the variations $\partial \widetilde{\mathbf{W}}_{R,j} / \partial \mathbf{W}_{\mathrm{D},R}$ and $\partial \widetilde{\mathbf{W}}_{S,j} / \partial \mathbf{W}_{\mathrm{D},S}$.

Utilizing the chain rule, the linearization of the mixed-out flow variables can be expanded to produce

$$\delta \widetilde{\mathbf{W}} = \frac{\partial \widetilde{\mathbf{W}}}{\partial \mathbf{W}_{\mathrm{D}}} = \frac{\partial \widetilde{\mathbf{W}}}{\partial \bar{\mathbf{F}}} \frac{\partial \bar{\mathbf{F}}}{\partial \mathbf{W}_{\mathrm{D}}} \ , \tag{4.99}$$

where $\bar{\mathbf{F}}$ are the averaged fluxes obtained from either the rotor or stator domain cells located next to a rotor-stator interface, cf. Eqn. (3.95). With Eqn. (3.96), which formulates the flux vector in a cylindrical coordinate system, the linearization of the averaged fluxes is obtained straightforward from

$$\delta \bar{\mathbf{F}} = \frac{\partial \bar{\mathbf{F}}}{\partial \mathbf{W}_{\mathrm{D}}} = \frac{1}{l_P} \frac{\partial \mathbf{F}_{\mathrm{D}}}{\partial \mathbf{W}_{\mathrm{D}}} \ , \tag{4.100}$$

where $l_P$ is either the rotor or stator pitch area and $\partial\mathbf{F}_{\mathrm{D}}/\partial\mathbf{W}_{\mathrm{D}}$ represents the flux Jacobian expressed in cylindrical coordinates and defined as in the appendix A.2.2. The variation $\partial\widetilde{\mathbf{W}}/\partial\bar{\mathbf{F}}$ is derived by first defining the mixed-out flow variables $\widetilde{\mathbf{W}} = [\widetilde{w}_1, \widetilde{w}_{x_1}, \widetilde{w}_\varphi, \widetilde{w}_r, \widetilde{w}_5]^T$ in terms of the averaged fluxes,

$$\widetilde{\mathbf{W}} = \left[\frac{\bar{F}_1^2}{\bar{F}_{x_1} - \widetilde{p}},\ \bar{F}_1,\ \frac{\bar{F}_1\bar{F}_\varphi}{\bar{F}_{x_1} - \widetilde{p}},\ \frac{\bar{F}_1\bar{F}_r}{\bar{F}_{x_1} - \widetilde{p}},\ \frac{\widetilde{p}}{\gamma - 1} + \frac{\bar{F}_{x_1} - \widetilde{p}}{2} + \frac{\bar{F}_\varphi^2 + \bar{F}_r^2}{2(\bar{F}_{x_1} - \widetilde{p})}\right]^T , \tag{4.101}$$

and then linearizing the averaged conservative variable vector with respect to the averaged fluxes to provide

$$\frac{\partial\delta\widetilde{\mathbf{W}}}{\partial\bar{\mathbf{F}}} = \begin{bmatrix} \bar{F}_1^2 B_1 + 2D_1 & 1 & \bar{F}_1\bar{F}_\varphi B_1 + D_3 & \bar{F}_1\bar{F}_r B_1 + D_4 & \frac{1}{2}(D_5 B_1 + T_1) \\ \bar{F}_1^2 B_2 - D_1^2 & 0 & \bar{F}_1\bar{F}_\varphi B_2 - D_1 D_3 & \bar{F}_1\bar{F}_r B_2 - D_1 D_4 & \frac{1}{2}(D_5 B_2 + T_2 - D_5 D^2 + 1) \\ \bar{F}_1^2 B_3 & 0 & \bar{F}_1\bar{F}_\varphi B_3 + D_1 & \bar{F}_1\bar{F}_r B_3 & \frac{1}{2}(D_5 B_3 + T_3 + 2D_3) \\ \bar{F}_1^2 B_4 & 0 & \bar{F}_1\bar{F}_\varphi B_4 & \bar{F}_1\bar{F}_r B_4 + D_1 & \frac{1}{2}(D_5 B_4 + T_4 + 2D_4) \\ \bar{F}_1^2 B_5 & 0 & \bar{F}_1\bar{F}_\varphi B_5 & \bar{F}_1\bar{F}_r B_5 & \frac{1}{2}(D_5 B_5 + T_5) \end{bmatrix} \tag{4.102}$$

with the abbreviations

$$D = (\bar{F}_{x_1} - \widetilde{p})^{-1} , \qquad D_1 = \bar{F}_1 D , \qquad D_2 = \bar{F}_{x_1} D , \tag{4.103}$$

$$D_3 = \bar{F}_\varphi D , \qquad D_4 = \bar{F}_r D , \qquad D_5 = (\bar{F}_\varphi^2 + \bar{F}_r^2) , \tag{4.104}$$

and

$$B_i = D^2 \frac{\partial\widetilde{p}}{\partial\bar{F}_i} , \qquad T_i = \frac{3-\gamma}{\gamma - 1}\frac{\partial\widetilde{p}}{\partial\bar{F}_i} , \tag{4.105}$$

where the linearization of the mixed-out pressure is given by

$$\frac{\partial\widetilde{p}}{\partial\bar{\mathbf{F}}} = \frac{\gamma^2 - 1}{(\gamma + 1)\bar{F}_x} \begin{bmatrix} -\bar{F}_5 & \\ \dfrac{\bar{F}_x}{\gamma^2 - 1} & +\dfrac{\gamma^2\bar{F}_{x_1}}{\gamma^2 - 1} \\ \bar{F}_\varphi & \\ \bar{F}_r & \\ -\bar{F}_1 & \end{bmatrix} , \tag{4.106}$$

with

$$\bar{F}_x = \sqrt{\bar{F}_{x_1}^2 + (\gamma^2 - 1)(\bar{F}_{x_1}^2 + \bar{F}_{\varphi}^2 + \bar{F}_r^2 - 2\bar{F}_{x_1}\bar{F}_5)}\ . \tag{4.107}$$

This completes the linearization of the mixed-out flow variables.

### Linearization of Giles' 1D characteristics

The calculation of the boundary contribution from Chima's steady averaging plane to the adjoint flux of a domain cell next to a rotor-stator interface requires the linearization of Giles' 1D characteristics. In the following, the derivation focuses on the development of the adjoint boundary flux from the mixing-plane interface on the rotor side. The stator side adjoint flux contribution is obtained by substituting the subscript $R$ with $S$ and vice versa. From Eqn. (4.97) it follows that the characteristic contribution to the adjoint boundary flux on the rotor side is specified through the transformation matrices

$$\frac{\partial \delta \widetilde{\mathbf{W}}_{\mathbf{c},RS}}{\partial \mathbf{W}_{\mathrm{D},R}} \qquad \text{and} \qquad \frac{\partial \delta \widetilde{\mathbf{W}}_{\mathbf{c},SR}}{\partial \mathbf{W}_{\mathrm{D},R}}\ , \tag{4.108}$$

where the index $j$, which indicates summation over all cells in the neighbouring domain at the same radial section, has been neglected for the sake of brevity.

The transformation matrix $\partial \delta \widetilde{\mathbf{W}}_{\mathbf{c},RS} / \partial \mathbf{W}_{\mathrm{D},R}$ can be expanded to produce

$$\frac{\partial \delta \widetilde{\mathbf{W}}_{\mathbf{c},RS}}{\partial \mathbf{W}_{\mathrm{D},R}} = \frac{\partial (\widetilde{\mathbf{P}} \delta \mathbf{C}_{RS})}{\partial \mathbf{W}_{\mathrm{D},R}}\ , \tag{4.109}$$

where $\delta \mathbf{C} = [\delta c_1, \delta c_{x_1}, \delta c_{\varphi}, \delta c_r, \delta c_5]^T$ is the vector of characteristic variables and $\widetilde{\mathbf{P}}$ is the right eigenvector matrix of the convective flux Jacobian built of the mixed-out variables from the stator domain. Since $\widetilde{\mathbf{P}}$ does not depend on the state vector $\mathbf{W}_{\mathrm{D},R}$, Eqn. (4.109) can be simplified to

$$\frac{\partial \delta \widetilde{\mathbf{W}}_{\mathbf{c},RS}}{\partial \mathbf{W}_{\mathrm{D},R}} = \widetilde{\mathbf{P}} \frac{\partial \delta \mathbf{C}_{RS}}{\partial \mathbf{W}_{\mathrm{D},R}}\ , \tag{4.110}$$

with the characteristic variables $\delta \mathbf{C}_{RS}$ defined as in Eqn. (3.100). While the left eigenvector matrix $\widetilde{\mathbf{P}}^{-1}$ is independent of the rotor flow variables as well, the linearization of the vector of flow variable perturbations, cf. Eqn. (3.101), yields the identity matrix. Thus, at an inlet boundary with supersonic inflow, where all five characteristics are incoming, the transformation matrix $\partial\delta\widetilde{\mathbf{W}}_{\mathbf{c},RS}/\partial\mathbf{W}_{\mathrm{D},R}$ equals to zero. At a supersonic outlet boundary, where all characteristics are outgoing, the corresponding transformation matrix simplifies to the identity matrix. For subsonic inflow, the linearized 1D characteristics take on the expression

$$\frac{\partial\delta\widetilde{\mathbf{W}}_{\mathbf{c},RS}}{\partial\widetilde{\mathbf{W}}_{\mathbf{P},R}} = \begin{bmatrix} B_{11} & B_{21} & B_{31} & B_{41} & B_{51} \\ B_{12} & B_{22} & B_{32} & B_{42} & B_{52} \\ B_{13} & B_{23} & B_{33} & B_{43} & B_{53} \\ B_{14} & B_{24} & B_{34} & B_{44} & B_{54} \\ B_{15} & B_{25} & B_{35} & B_{45} & B_{55} \end{bmatrix} , \tag{4.111}$$

where $B_{ij} = T_i Z_j$ with $T_i$ and $Z_j$ being defined as follows

$$T_1 = C_1 \, , \qquad Z_1 = \tfrac{1}{2}C_2|\widetilde{\mathbf{U}}|^2 + \widetilde{u}_{x_1}\widetilde{c} \, , \tag{4.112}$$

$$T_2 = C_1(\widetilde{u}_{x_1} - \widetilde{c}) \, , \qquad Z_2 = -\, C_2\widetilde{u}_{x_1} - \widetilde{c} \, , \tag{4.113}$$

$$T_3 = C_1\widetilde{u}_{\varphi} \, , \qquad Z_3 = -\, C_2\widetilde{u}_{\varphi} \, , \tag{4.114}$$

$$T_4 = C_1\widetilde{u}_{r} \, , \qquad Z_4 = -\, C_2\widetilde{u}_{r} \, , \tag{4.115}$$

$$T_5 = \tfrac{1}{2}\big(C_1|\widetilde{\mathbf{U}}|^2 - \widetilde{u}_{x_1}\widetilde{c}^{-1} + C_2^{-1}\big) \, , \qquad Z_5 = C_2 \, , \tag{4.116}$$

$$C_1 = (2\widetilde{c}^2)^{-1} \, , \qquad C_2 = \gamma - 1 \, . \tag{4.117}$$

At a subsonic outlet boundary the transformation matrix may be written as

$$\frac{\partial\delta\widetilde{\mathbf{W}}_{\mathbf{c},RS}}{\partial\widetilde{\mathbf{W}}_{\mathbf{P},R}} = \begin{bmatrix} B_{11} & B_{21} & B_{31} - \widetilde{u}_{\varphi} & B_{41} - \widetilde{u}_{r} & B_{51} - (\widetilde{u}_{\varphi}^2 + \widetilde{u}_{r}^2) \\ B_{12} & B_{22} & B_{32} & B_{42} & B_{52} \\ B_{13} & B_{23} & B_{33} + 1 & B_{43} & B_{53} + \widetilde{u}_{\varphi} \\ B_{14} & B_{24} & B_{34} & B_{44} + 1 & B_{54} + \widetilde{u}_{r} \\ B_{15} & B_{25} & B_{35} & B_{45} & B_{55} \end{bmatrix} , \tag{4.118}$$

where $B_{ij} = T_{1i}Z_{1j} + T_{2i}Z_{2j}$ with

$$\begin{aligned}
T_{11} &= 1 \ , & T_{21} &= C_1 \ , & &(4.119)\\
T_{12} &= \widetilde{u}_{x_1} \ , & T_{22} &= C_1(\widetilde{u}_{x_1} + \widetilde{c}) \ , & &(4.120)\\
T_{13} &= \widetilde{u}_{\varphi} \ , & T_{23} &= C_1\widetilde{u}_{\varphi} \ , & &(4.121)\\
T_{14} &= \widetilde{u}_{r} \ , & T_{24} &= C_1\widetilde{u}_{r} \ , & &(4.122)\\
T_{15} &= \tfrac{1}{2}|\widetilde{\mathbf{U}}|^2 \ , & T_{25} &= \tfrac{1}{2}\big(C_1|\widetilde{\mathbf{U}}|^2 + \widetilde{u}_{x_1}\widetilde{c}^{-1} + C_2^{-1}\big) \ , & &(4.123)\\
Z_{11} &= 1 - C_2C_1|\widetilde{\mathbf{U}}|^2 \ , & Z_{21} &= \tfrac{1}{2}C_2|\widetilde{\mathbf{U}}|^2 - \widetilde{u}_{x_1}\widetilde{c} \ , & &(4.124)\\
Z_{12} &= 2C_2C_1\widetilde{u}_{x_1} \ , & Z_{22} &= -C_2\widetilde{u}_{x_1} + \widetilde{c} \ , & &(4.125)\\
Z_{13} &= 2C_2C_1\widetilde{u}_{\varphi} \ , & Z_{23} &= -C_2\widetilde{u}_{\varphi} \ , & &(4.126)\\
Z_{14} &= 2C_2C_1\widetilde{u}_{r} \ , & Z_{24} &= -C_2\widetilde{u}_{r} \ , & &(4.127)\\
Z_{15} &= -2C_2C_1 \ , & Z_{25} &= C_2 \ , & &(4.128)\\
C_1 &= (2\widetilde{c}^2)^{-1} \ , & C_2 &= \gamma - 1 \ . & &(4.129)
\end{aligned}$$

Utilizing the chain rule greatly simplifies the derivation of the second characteristic boundary contribution $\partial\delta\widetilde{\mathbf{W}}_{\mathbf{c},SR}/\partial\mathbf{W}_{\mathrm{D},R}$ of Eqn. (4.108). The transformation matrix can be expanded to produce

$$\frac{\partial\delta\widetilde{\mathbf{W}}_{\mathbf{c},SR}}{\partial\mathbf{W}_{\mathrm{D},R}} = \frac{\partial\delta\widetilde{\mathbf{W}}_{\mathbf{c},SR}}{\partial\widetilde{\mathbf{W}}_{\mathbf{P},R}}\frac{\partial\widetilde{\mathbf{W}}_{\mathbf{P},R}}{\partial\widetilde{\mathbf{W}}_{R}}\frac{\partial\widetilde{\mathbf{W}}_{R}}{\partial\bar{\mathbf{F}}_{R}}\frac{\partial\bar{\mathbf{F}}_{R}}{\partial\mathbf{W}_{\mathrm{D},R}} \ , \qquad (4.130)$$

where the last two terms $\partial\widetilde{\mathbf{W}}_R/\partial\bar{\mathbf{F}}_R$ and $\partial\widetilde{\mathbf{F}}_R/\partial\mathbf{W}_{\mathrm{D},R}$ are readily available since they were already derived during the discussion of the linearization of the mixed-out flow variables, cf. subsection 4.3.5. Moreover, the term $\partial\widetilde{\mathbf{W}}_{\mathbf{P},R}/\partial\widetilde{\mathbf{W}}_R$ simply is the primitive transformation matrix $\mathbf{M}^{-1}$, cf. Eqn. (A.4); this time evaluated using mixed-out quantities. Thus, the only term missing to complete the derivation is the first term of Eqn. (4.109), i.e. the 1D characteristics $\delta\widetilde{\mathbf{W}}_{\mathbf{c},SR}$ differentiated with respect to the mixed-out primitive variables $\widetilde{\mathbf{W}}_{\mathbf{P},R}$. Using Eqn. (3.99) this term can be expanded to produce

$$\frac{\partial \delta \widetilde{\mathbf{W}}_{\mathbf{c},SR}}{\partial \mathbf{W}_{\mathbf{P},R}} = \frac{\partial (\widetilde{\mathbf{P}} \delta \mathbf{C}_{SR})}{\partial \mathbf{W}_{\mathbf{P},R}} \ , \tag{4.131}$$

where both the characteristics $\delta\mathbf{C}_{SR}$ and the right eigenvector matrix $\widetilde{\mathbf{P}}$ are functions of the mixed-out primitive variables from the rotor domain. Similarly, the left eigenvector matrix $\widetilde{\mathbf{P}}^{-1}$ and the vector defining the jump perturbation $\delta\mathbf{W}_{SR}$, cf. Eqns.(3.100) and (3.101), both comprise mixed-out primitive variables from the rotor domain. The linearization of Eqn. (3.101) yields the negative identity matrix. For supersonic flow at the inlet boundary the transformation matrix of the characteristics equals zero again, while at a supersonic outlet boundary the transformation matrix simplifies to

$$\frac{\partial \delta \widetilde{\mathbf{W}}_{\mathbf{c},SR}}{\partial \mathbf{W}_{\mathbf{P},R}} = \widetilde{\mathbf{M}} \ , \tag{4.132}$$

where $\widetilde{\mathbf{M}}$ is defined as in Eqn. (A.3) and evaluated using mixed-out rotor quantities. Developing the transformation matrices for a subsonic inflow or outflow boundary is more tedious, due to the numerous terms that have to be carried forward during the derivation. At a subsonic inlet the linearization yields

$$\frac{\partial \delta \widetilde{\mathbf{W}}_{\mathbf{P},SR}}{\partial \widetilde{\mathbf{W}}_{\mathbf{P},R}} = \begin{bmatrix} B_{11} + C_3 \dfrac{T_1}{\widetilde{\rho}} & B_{21} + C_3 \dfrac{D_2}{\widetilde{\rho}} & B_{31} + C_3 \dfrac{T_3}{\widetilde{\rho}} & B_{41} + C_3 \dfrac{T_4}{\widetilde{\rho}} & B_{51} + C_3 \dfrac{D_5}{2\widetilde{\rho}} \\ B_{12} & B_{22} + C_3 T_1 & B_{32} & B_{42} & B_{52} + C_3 T_2 \\ B_{13} & B_{23} & B_{33} + C_3 T_1 & B_{43} & B_{53} + C_3 T_3 \\ B_{14} & B_{24} & B_{34} & B_{44} + C_3 T_1 & B_{54} + C_3 T_4 \\ B_{15} - C_3 \dfrac{T_1}{\widetilde{p}} & B_{25} - C_3 \dfrac{D_2}{\widetilde{p}} & B_{35} - C_3 \dfrac{T_3}{\widetilde{p}} & B_{45} - C_3 \dfrac{T_4}{\widetilde{p}} & B_{55} - C_3 \dfrac{D_5}{2\widetilde{p}} \end{bmatrix} \tag{4.133}$$

with $B_{ij} = T_i Z_j$, where $T_i$ and $Z_j$ take on the expressions

$$T_1 \ = \ C_1 \ , \qquad Z_1 \ = \ \rho \widetilde{c} (\widetilde{u}_{x_1} - u_{x_1})(2\widetilde{\rho})^{-1} \ , \tag{4.134}$$

$$T_2 \ = \ C_1 (\widetilde{u}_{x_1} - \widetilde{c}) \ , \qquad Z_2 \ = \ C_2 \rho^2 \widetilde{c} (\widetilde{u}_{x_1} - u_{x_1}) \ , \tag{4.135}$$

$$T_3 \ = \ C_1 \widetilde{u}_{\varphi} \ , \qquad Z_3 \ = \ C_2 \rho (\widetilde{u}_{\varphi} - u_{\varphi}) \ , \tag{4.136}$$

$$T_4 \ = \ C_1 \widetilde{u}_r \ , \qquad Z_4 \ = \ C_2 \rho (\widetilde{u}_r - u_r) \ , \tag{4.137}$$

$$T_5 = \tfrac{1}{2}\big(C_1|\widetilde{\mathbf{U}}|^2 - \widetilde{u}_{x_1}\widetilde{c}^{-1} + C_2^{-1}\big)\ , \quad Z_5 = \rho\widetilde{c}(\widetilde{u}_{x_1} - u_{x_1})(2\widetilde{p})^{-1} - 1\ , \tag{4.138}$$

$$D_2 = C_1\widetilde{u}_{x_1} - (4\widetilde{c})^{-1}\ , \quad D_5 = C_1(|\widetilde{\mathbf{U}}|^2 - \widetilde{u}_{x_1}\widetilde{c})\ , \tag{4.139}$$

$$C_1 = (2\widetilde{c}^2)^{-1}\ , \quad C_3 = \rho\widetilde{c}(\widetilde{u}_{x_1} - u_{x_1}) + \rho C_2 C_4 - \widetilde{p}\ , \tag{4.140}$$

$$C_2 = \gamma - 1\ , \quad C_4 = \tfrac{1}{2}|\widetilde{\mathbf{U}}|^2 - \widetilde{u}_{x_1}u_{x_1} - \widetilde{u}_{\varphi}u_{\varphi} + \widetilde{u}_r u_r + E. \tag{4.141}$$

At a subsonic outlet boundary the transformation matrix may be written as

$$\frac{\partial\delta\widetilde{\mathbf{W}}_{\mathbf{c},RS}}{\partial\widetilde{\mathbf{W}}_{\mathbf{P},R}} = \begin{bmatrix} B_{11} + C_6\frac{T_{21}}{\widetilde{\rho}} & B_{12} + C_6\frac{D_2}{\widetilde{\rho}} & B_{13} + C_6\frac{T_{23}}{\widetilde{\rho}} & B_{14} + C_6\frac{T_{24}}{\widetilde{\rho}} & B_{15} + C_6\frac{D_5}{\widetilde{2\rho}} \\ B_{21} & B_{22} + L_1 & B_{23} & B_{24} & B_{25} + C_6T_{22} + C_5\widetilde{u}_{x_1} \\ B_{31} & B_{32} & B_{33} + L_2 & B_{34} & B_{35} + C_6T_{23} - C_5\widetilde{u}_{\varphi} - L_3 \\ B_{41} & B_{42} & B_{43} & B_{44} + L_2 & B_{45} + C_6T_{24} - C_5\widetilde{u}_r - L_4 \\ B_{51} - C_6\frac{T_{21}}{\widetilde{p}} & B_{52} - C_6\frac{D_2}{\widetilde{p}} & B_{53} - C_6\frac{T_{23}}{\widetilde{p}} & B_{54} - C_6\frac{T_{24}}{\widetilde{p}} & B_{55} - C_6\frac{D_5}{\widetilde{2p}} \end{bmatrix} \tag{4.142}$$

where $B_{ij} = T_{1i}Z_{1j} + T_{2i}Z_{2j}$ and

$$T_{11} = 1\ , \quad T_{21} = C_1\ , \tag{4.143}$$

$$T_{12} = \widetilde{u}_{x_1}\ , \quad T_{22} = C_1(\widetilde{u}_{x_1} + \widetilde{c})\ , \tag{4.144}$$

$$T_{13} = \widetilde{u}_{\varphi}\ , \quad T_{23} = C_1\widetilde{u}_{\varphi}\ , \tag{4.145}$$

$$T_{14} = \widetilde{u}_r\ , \quad T_{24} = C_1\widetilde{u}_r\ , \tag{4.146}$$

$$T_{15} = \tfrac{1}{2}|\widetilde{\mathbf{U}}|^2\ , \quad T_{25} = \tfrac{1}{2}\big(C_1|\widetilde{\mathbf{U}}|^2 + \widetilde{u}_{x_1}\widetilde{c}^{-1} + C_2^{-1}\big)\ , \tag{4.147}$$

$$Z_{11} = -\rho C_2 C_1|\widetilde{\mathbf{U}}|^2\ , \quad Z_{21} = \tfrac{1}{2}C_2|\widetilde{\mathbf{U}}|^2 - \widetilde{u}_{x_1}\widetilde{c}\ , \tag{4.148}$$

$$Z_{12} = 2C_2C_1\widetilde{u}_{x_1}\ , \quad Z_{22} = -C_2\widetilde{u}_{x_1} + \widetilde{c}\ , \tag{4.149}$$

$$Z_{13} = 2C_2C_1\widetilde{u}_{\varphi}\ , \quad Z_{23} = -C_2\widetilde{u}_{\varphi}\ , \tag{4.150}$$

$$Z_{14} = 2C_2C_1\widetilde{u}_r\ , \quad Z_{24} = -C_2\widetilde{u}_r\ , \tag{4.151}$$

$$Z_{15} = -2C_2C_1\ , \quad Z_{25} = C_2\ , \tag{4.152}$$

$$D_2 = C_1\widetilde{u}_{x_1} + (4\widetilde{c})^{-1}\ , \quad D_5 = C_1(|\widetilde{\mathbf{U}}|^2 - \widetilde{u}_{x_1}\widetilde{c})\ , \tag{4.153}$$

$$C_1 = (2\widetilde{c}^2)^{-1}\ , \quad C_4 = \tfrac{1}{2}|\widetilde{\mathbf{U}}|^2 - \widetilde{u}_{x_1}u_{x_1} - \widetilde{u}_{\varphi}u_{\varphi} - \widetilde{u}_r u_r + E, \tag{4.154}$$

$$C_2 = \gamma - 1 \ , \qquad C_5 = \rho - \widetilde{\rho} - 2C_1(\rho C_2 C_4 - \widetilde{p}) \ , \tag{4.155}$$

$$C_3 = \gamma C_2^{-1} \ , \qquad C_6 = -\rho\widetilde{c}(\widetilde{u}_{x_1} - u_{x_1}) + \rho C_2 C_4 - \widetilde{p} \ , \tag{4.156}$$

$$L_1 = C_6 C_1 + C_5 \ , \qquad L_3 = \rho(2\widetilde{u}_\varphi - u_\varphi) \ , \tag{4.157}$$

$$L_2 = L_1 - \rho \ , \qquad L_4 = \rho(2\widetilde{u}_r - u_r) \ . \tag{4.158}$$

This completes the development of the adjoint boundary flux contribution from the the rotor-stator interface. The derived adjoint mixing-plane boundary condition is consistent with the averaging-plane utilized in the flow solver. Note, the adjoint boundary flux is formulated in a cylindrical coordinate system, while the adjoint field equations are solved in Cartesian coordinates. Thus, once all individual mixing-plane contributions are calculated and exchanged between the blade rows, the adjoint boundary flux has to be transferred into a Cartesian coordinate formulation before it can be added to the adjoint field equation.

## 4.4 Linearization of the Discrete Objective Function

The right hand side of the adjoint equation (4.7) is obtained through linearizing the discretized objective function $I$. Turbomachinery specific functionals are typically evaluated at the boundaries of the computational domain, most frequently at the inlet and exit plane. Hence, the linearized objective function is given by

$$\frac{\partial I}{\partial \mathbf{W}_D} = \frac{\partial I}{\partial \mathbf{W}_B} \boldsymbol{\mathcal{T}}_{BD} \ , \tag{4.159}$$

where $\boldsymbol{\mathcal{T}}_{BD} = \partial \mathbf{W}_B / \partial \mathbf{W}_D$ is the boundary-type-specific transformation matrix discussed in section 4.3. However, it is now multiplied with the linearization of the objective function with respect to the boundary state. Since the functionals are usually integrated and either area-averaged, mass-averaged, or mixed-out quantities, $\partial I / \partial \mathbf{W}_B$ depends on the underlying averaging technique. For instance, let $\bar{q}$ be a mass-averaged and normalized scalar quantity defined as

$$\bar{q} = \frac{1}{q_0} \frac{\sum_i \left[q(\rho U_n A)\right]_i}{\sum_i \left[\rho U_n A\right]_i} \ , \tag{4.160}$$

where $U_n$ is the velocity component normal to the boundary, $A$ is the surface, and $q_0$ is the initial mass-averaged quantity. Then, its linearization is given by

$$\frac{\partial \bar{q}}{\partial \mathbf{W}_{B,i}} = \frac{1}{q_0 \sum_i \left[\rho U_n A\right]_i} \left[ (\rho U_n A)_i \frac{\partial q_i}{\partial \mathbf{W}_B} + (q - \hat{q}) \frac{\partial (\rho U_n A)_i}{\partial \mathbf{W}_B} \right] \ , \tag{4.161}$$

where

$$\hat{q} = \frac{\sum_i \left[q(\rho U_n A)\right]_i}{\sum_i \left[\rho U_n A\right]_i} \ . \tag{4.162}$$

Note, while different averaging techniques might lead to nearly identical function values, the linearized function might differ noticeably and, thus, might affect the adjoint solution.

The design objective for the test cases presented in this work is to minimize the entropy generation rate, which is equivalent to maximizing the isentropic efficiency but results in simpler sensitivity expressions. The discrete objective function is defined as the difference between the normalized mass-averaged entropy at the outlet and inlet

$$I = \frac{1}{s_0 \cdot \dot{m}_{\text{out}}} \sum_{B_{\text{out}}} \left[s(\rho U_n A)\right]_i - \frac{1}{s_0 \cdot \dot{m}_{\text{in}}} \sum_{B_{\text{in}}} \left[s(\rho U_n A)\right]_i \ , \tag{4.163}$$

where $s_0$ is the initial entropy at the outlet and $s = p/\rho^\gamma$. Furthermore, constraints are prescribed on the mass flow rate $\dot{m}$ as well as on the mass-weighted total pressure ratio $\bar{\pi}$ through the expressions

$$c_{\dot{m}} = \frac{1}{2} \left[ \frac{\dot{m}}{\dot{m}_0} - 1 \right]^2 \qquad \text{and} \qquad c_{\bar{\pi}} = \frac{1}{2} \left[ \frac{\bar{\pi}}{\bar{\pi}_0} - 1 \right]^2 \ , \tag{4.164}$$

where $\dot{m}_0$ is the initial mass flow rate defined as in Eqn. (2.42) and $\bar{\pi}_0$ is the initial ratio between the mass-weighted total pressure at the outlet and inlet, cf. Eqn. (2.45). The linearization of the entropy objective function (4.163) is obtained from Eqn. (4.161), where

$$\frac{\partial s}{\partial \mathbf{W}_B} = \frac{1}{\rho^\gamma}\begin{bmatrix} -\dfrac{\gamma p}{\rho} + (\gamma-1)\dfrac{|\mathbf{U}|^2}{2} \\ -(\gamma-1)u_1 \\ -(\gamma-1)u_2 \\ -(\gamma-1)u_3 \\ \gamma-1 \end{bmatrix} \tag{4.165}$$

and

$$\frac{\partial(\rho u_n A)}{\partial \mathbf{W}_B} = \begin{bmatrix} 0 \\ n_1 A \\ n_2 A \\ n_3 A \\ 0 \end{bmatrix}. \tag{4.166}$$

The linearized constraints take on the expressions

$$\frac{\partial c_{\dot m}}{\partial \mathbf{W}_B} = \frac{1}{\dot m_0}\left[\frac{\dot m}{\dot m_0} - 1\right]\frac{\partial \dot m}{\partial \mathbf{W}_B} \qquad \text{and} \qquad \frac{\partial c_{\bar\pi}}{\partial \mathbf{W}_B} = \frac{1}{\bar\pi_0}\left[\frac{\bar\pi}{\bar\pi_0} - 1\right]\frac{\partial \bar\pi}{\partial \mathbf{W}_B}\,, \tag{4.167}$$

where the variation of the mass flow rate $\partial\dot m/\partial\mathbf{W}_B$ is defined as in Eqn. (4.166) and the linearization of the mass-weighted total pressure ratio $\partial\bar\pi/\partial\mathbf{W}_B$ may be calculated from Eqn. (4.161). The inlet total pressure is prescribed and, thus, does not change between subsequent design iterations. Therefore, it can be treated as a constant and only the total pressure at the exit boundary has to be linearized. The variation of the total pressure with respect to the conserved variables yields the expression

$$\frac{\partial p_t}{\partial \mathbf{W}_B} = \left[1 + \frac{\gamma-1}{2c^2}|\mathbf{U}|^2\right]^{\frac{1}{\gamma-1}} \begin{bmatrix} \dfrac{|\mathbf{U}|^2}{2}(\gamma-2) - \dfrac{(\gamma-1)}{4c^2}|\mathbf{U}|^4 \\ -u_1(\gamma-2) + \dfrac{(\gamma-1)}{2c^2}|\mathbf{U}|^2 u_1 \\ -u_2(\gamma-2) + \dfrac{(\gamma-1)}{2c^2}|\mathbf{U}|^2 u_2 \\ -u_3(\gamma-2) + \dfrac{(\gamma-1)}{2c^2}|\mathbf{U}|^2 u_3 \\ (\gamma-1) - \dfrac{(\gamma-1)}{2c^2}|\mathbf{U}|^2 \end{bmatrix}. \tag{4.168}$$

# Chapter 5

# Gradient Calculation and Optimization Strategy

This chapter discusses the gradient calculation for the adjoint approach and presents the optimization strategy employed to determine an improved design. Section 5.1 introduces the different kinds of design variables utilized within this work, elaborates on their advantages and disadvantages, and develops the gradient formulation for the different choices of design variables. Furthermore, the two different grid movement schemes used to perturb the computational grid are introduced. Section 5.2 presents the optimization algorithm used to determine the shape modifications and illustrates the treatment of the constraints within the framework of a sequential-quadratic programming algorithm.

## 5.1 Gradient Calculation

Once the adjoint system is solved and the co-state solution $\boldsymbol{\psi}$ is available, the objective function gradient can be calculated through the simplified expression

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\partial I}{\partial \boldsymbol{\alpha}} - \boldsymbol{\psi}^T \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}} \ , \tag{5.1}$$

which is free of the flow variable sensitivities $\delta\mathbf{W}$. Thus, the gradient can be evaluated without having to re-compute the flow solution; cf. section 4.1 for the derivation of Eqn. (5.1). The partial derivative $\partial I/\partial\boldsymbol{\alpha}$ is zero for functionals which are only evaluated at the inlet or exit plane of the computational domain, since these boundaries are fixed. This is the case for the design objective and constraints employed within this research, cf. section 4.4; hence the gradient expression further simplifies to

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{\alpha}} = -\boldsymbol{\psi}^T \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}} \ . \tag{5.2}$$

In this work, the remaining partial derivative $\partial \mathbf{R}/\partial \boldsymbol{\alpha}$ is approximated through finite-differences, i.e. the residual is evaluated for the initial and perturbed design variables. Simple first-order forward differences are used. Since each variable has to be perturbed individually, $n$ grid perturbations are necessary for $n$ design parameters and, thus, the cost of the gradient evaluation is proportional to the number of design variables. Nonetheless, gradient calculation through Eqn. (5.1) or Eqn. (5.2) is dramatically faster than through the forward approach defined in Eqn. (4.1).

In the following two subsections the choice of the design variables is discussed and the approach utilized to perturb the computational grid is presented.

### 5.1.1 Design variables

The choice of the design variables is one of the most crucial steps in any optimization procedure. The success of an optimization does not only depend on the selected objective function but also on the choice of the right design variables. At the same time, it is important to deliberate and understand the relation between the design objective and a potential design variable. In case the objective and the design variables are only loosely coupled the effectiveness of the optimization might suffer. On the other hand, if the design objective is extremely sensitive to the occurring changes in the design variables or if their relation possesses a highly non-linear character, then the robustness of the optimization will most likely be negatively affected. Ideally, the selected design variables have a large influence on the design objective, while at the same time they provide a design space that guarantees a robust design problem. Furthermore, the choice of the design variables often depends on the stage of the design process at which the optimization problem is formulated. For instance, during the design of a new gas turbine a design task could be to increase the total pressure built-up throughout the compressor. In an earlier, more conceptual design stage, the number of compressor stages could serve as the design variables. At a later stage, where the conceptual design is mostly frozen and the number of compressor stages cannot be modified, the number of blades defined within each blade row could be used as the design variables. Even later, towards the final stages of the design process where the blade count is decided on as well, the blade geometry itself could serve as the design parameter. Apart from constraints related to the current design stage, geometrical constraints may be present due to requisites posed by other disciplines. For example, an aerodynamic shape optimization should not violate any structural requirements. Moreover, from a computational perspective, the available resources and time as well as the computational cost typically influence and limit the choice of the design variables.

In this research, design problems at earlier conceptual design stages are not investigated. Instead, the optimization problems posed are to redesign some given compressor configurations, which serve as the baseline design. The blade geometries are chosen to be the design variables, thus, the optimizer tries to meet the design objective by modifying the shape of the blades. In a computational domain, this naturally translates to changing the surface grid points. However, there exist several other ways to modify the shape during a design process as well; e.g. surface perturbations using Hicks-Henne bump functions, B-spline control points, etc. The following two sub-sections describe the two different types of design variables used within this work.

### Surface Grid Points

Using the surface grid points as design variables, i.e. $\boldsymbol{\alpha} = \mathbf{X_s}$, ensures that there is no restriction on the achievable geometry. Equation (5.2) becomes

$$\frac{\mathrm{d}I}{\mathrm{d}\mathbf{X_s}} = -\boldsymbol{\psi}^T \frac{\partial \mathbf{R}}{\partial \mathbf{X_s}} \ , \tag{5.3}$$

where $\mathbf{X_s}$ represents the grid nodes selected as design variables. As stated above, in this research the partial derivative $\partial\mathbf{R}/\partial\mathbf{X_s}$ is approximated via finite-differences. However, the gradient could also be calculated analytically by utilizing the chain rule, i.e.

$$\frac{\partial \mathbf{R}}{\partial \mathbf{X_s}} = \frac{\partial \mathbf{R}}{\partial \mathbf{X_v}} \frac{\partial \mathbf{X_v}}{\partial \mathbf{X_s}} \ , \tag{5.4}$$

where $\mathbf{X_v}$ are the interior volume points and the grid sensitivity $\partial \mathbf{X_v} / \partial \mathbf{X_s}$ is a function of the employed grid perturbation scheme, cf. subsection 5.1.2. If every surface grid point is used, then the complete design space, including the solution that achieves the global minimum, is attainable. However, the independent displacement of a single point violates the assumption that the surface geometry is continuous. Yet, this is irrelevant for gradient calculation using the adjoint approach since the evaluation of the adjoint sensitivity $\boldsymbol{\psi}^T(\partial \mathbf{R} / \partial \mathbf{X_s})$ does not require a re-calculation of the flow field for the perturbed geometry. In addition, point-wise gradients are typically highly non-smooth, which, throughout the optimization, would gradually lead to non-smooth geometry profiles. Therefore, it is necessary to either smooth the calculated gradient or the shape modifications determined from each gradient calculation. Within this optimization framework the modified geometry is smoothed after every design cycle; a second-order implicit smoothing technique is utilized to obtain an optimized blade shape without discontinuities.

For the two-dimensional design cases presented in the subsections 6.2.5, 6.2.6, and 6.3.5, the surface grid points are used as the design variables. However, for the gradient validation conducted in the subsections 6.2.4 and 6.3.4, so-called Hicks-Henne bump functions are selected as design variables instead. This is necessary since, as mentioned above, point-wise gradients cannot be used with the finite-difference method. Perturbing the geometry with the smooth Hicks-Henne bump functions ensures that the perturbed geometry remains smooth and, thus, enables the re-calculation of the flow solution to obtain finite-difference gradients.

### Hicks-Henne Bump Functions

Complex three-dimensional geometries require highly-resolved grids. When using the surface grid points as the design variables this usually leads to a very large number of design variables; more than 10,000 parameters would be common. Since the gradient calculation using the adjoint approach does not require a re-evaluation of the flow equations for each design variable perturbation, it is generally feasible to use surface grid points as design variables even for such highly-resolved grids, whereas the cost would be prohibitive if the gradients were computed by the classical finite-difference method. Nevertheless, the computational cost of the gradient calculation still scales with the number of design variables, since each design variable has to be perturbed individually. Computationally cheap grid perturbation schemes, such as algebraic algorithms, are often tailored to specific grid configurations and their implementation rapidly becomes complex when employing them to more general multi-block solvers such as that one used within this work. Other mesh movement schemes that are better suited for multi-block solvers are usually more costly, as shown in the next subsection. Thus, despite the remarkable advantages of the adjoint approach, there is still a need to keep the number of design variables low.

Therefore, in the three-dimensional design cases presented in the sections 6.3.6 and 6.3.7, Hicks-Henne bump functions [62] are utilized as design variables instead of the surface grid points, thus $\boldsymbol{\alpha} = \mathbf{X_b}$. Using these functions, the number of design variables can be reduced to less than 1,000 parameters. Another advantage of employing these smooth functions is that they ensure that the perturbed geometry remains smooth both during the gradient calculation and throughout the optimization. Hence, no additional gradient or geometry smoothing has to be performed. In this research work, the Hicks-Henne bumps are defined through a modified sine function given by

$$\Delta X_{s,i} = \Delta X_{b,j} \left[ \sin\left( \pi \hat{x}_{1,i}^{\frac{\log 0.5}{\log t_{l,1,j}}} \right) \right]^{t_{w,1,j}} \left[ \sin\left( \pi \hat{x}_{2,i}^{\frac{\log 0.5}{\log t_{l,2,j}}} \right) \right]^{t_{w,2,j}} . \tag{5.5}$$

The bump function (5.5) determines the amount of perturbation $\Delta X_{s,i}$ normal to the surface of the $i$-th surface grid point $X_{s,i}$ that is located in the vicinity of the $j$-th bump $X_{b,j}$ and the parameters provide the flexibility to place the bump at strategic points where a redesign is preferred while leaving other parts of the blade intact. For example, the maximum bump magnitude is provided by the coefficient $\Delta X_{b,j}$ and its location is defined by the two parameters $t_{l,1,j}$ and $t_{l,2,j}$, which specify the reduced coordinates in chord-wise and radial direction respectively. The coefficients $t_{w,1,j}$ and $t_{w,2,j}$ control the width of the $j$-th bump. The bigger the value of $t_w$, the more local the bump function will be. $\hat{x}_{1,i}$ and $\hat{x}_{2,i}$ define the location of a specific surface grid point $\Delta X_{s,i}$ in reduced coordinates. Here, the reduced coordinate $\hat{x}_{1,i}$ is defined along the grid line of the surface point running into the chord-wise direction and $\hat{x}_{2,i}$ represents the reduced coordinate of the second grid line running into the radial direction. The values of $\hat{x}_1$ are 0 and 1 at the leading edge and trailing edge respectively, while $\hat{x}_2$ takes on the values 0 at the hub section and 1 at the casing section.

If geometric parameters not handled directly by the solver are chosen as design variables, the chain rule has to be applied to express the gradient of the objective function with respect to these design variables. Thus, for Hicks-Henne bump functions the gradient formulation (5.2) becomes

$$\frac{\mathrm{d}I}{\mathrm{d}\mathbf{X_b}} = -\boldsymbol{\psi}^T \frac{\partial \mathbf{R}}{\partial \mathbf{X_b}} \ , \tag{5.6}$$

where now the sensitivity $\partial\mathbf{R}/\partial\mathbf{X_b}$ is approximated via finite-differences. The residual perturbation may be also calculated analytically from

$$\frac{\partial \mathbf{R}}{\partial \mathbf{X_b}} = \frac{\partial \mathbf{R}}{\partial \mathbf{X_v}} \frac{\partial \mathbf{X_v}}{\partial \mathbf{X_s}} \frac{\partial \mathbf{X_s}}{\partial \mathbf{X_b}} \ . \tag{5.7}$$

In Eqn. (5.7), the grid sensitivity $\partial\mathbf{X_v}/\partial\mathbf{X_s}$ again depends on the underlying grid perturbation scheme, while the second partial derivative $\partial\mathbf{X_s}/\partial\mathbf{X_b}$ is a function of the chosen design variables. In case of Hicks-Henne bump functions it can be obtained with little effort; from Eqn. (5.5) it follows

$$\frac{\partial X_{s,i}}{\partial X_{b,j}} = \left[\sin\left(\pi \hat{x}_{1,i}^{\frac{\log 0.5}{\log t_{l,1,j}}}\right)\right]^{t_{w,1,j}} \left[\sin\left(\pi \hat{x}_{2,i}^{\frac{\log 0.5}{\log t_{l,2,j}}}\right)\right]^{t_{w,2,j}} \ . \tag{5.8}$$

### 5.1.2 Grid Perturbation

Grid deformation schemes play an important role in numerical aerodynamic shape optimization. As the shape of the body at hand changes, the computational grid must adapt to conform to the deformed geometry. Thus, grid perturbation is required during the gradient calculation and at the end of each design cycle to update the current design. Re-generating each of these grids manually defeats the purpose of an efficient optimization framework with as little user interaction as possible. Re-generating the grids through automatic grid generation can be a challenge to implement since it may still require adjustments from a user to ensure adequate quality. Furthermore, if the re-generated grid is comprised of a different distribution or number of grid cells, an interpolation scheme would also be needed to interpolate data from the previous grid. Alternatively, the current grid can be deformed based on the displacements of the surface nodes. This approach is favourable in the sense that it can be implemented as an automated process while maintaining a consistent grid connectivity and grid sensitivity. Thus, this grid deformation approach is preferred and implemented within this optimization framework. Consequently, the gradient expressions (5.2) may be rewritten in a general form, as

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{\alpha}} = -\boldsymbol{\psi}^T \frac{\partial \mathbf{R}}{\partial \mathbf{X_v}} \frac{\partial \mathbf{X_v}}{\partial \mathbf{X_s}} \frac{\partial \mathbf{X_s}}{\partial \boldsymbol{\alpha}} \ , \tag{5.9}$$

where $\boldsymbol{\alpha}$ are the design points and represent either the surface grid points $\mathbf{X_s}$ or Hicks-Henne bump functions $\mathbf{X_b}$. $\mathbf{X_v}$ are again the interior volume points; their location is a function of the surface grid points, which in turn may depend on some other control points such as Hicks-Henne bump functions. As already stated above, the grid sensitivity

$$\delta \mathbf{X_{vs}} = \frac{\partial \mathbf{X_v}}{\partial \mathbf{X_s}} \tag{5.10}$$

is a function of the underlying grid movement scheme. In case an automatic grid generator is used and the grid sensitivities cannot be determined analytically, the partial derivative $\partial \mathbf{X_v}/\partial \mathbf{X_s}$ has to be estimated via finite-differences.

Design cases which produce large shape changes demand robust grid deformation schemes which are typically computationally expensive. Many different grid movement schemes have been devised over the past couple of decades, each with its own advantages and disadvantages; algebraic grid movement schemes [63, 64], schemes utilizing Laplace smoothing [65], linear spring analogy methods [66, 67], grid movers based on the equations of linear elasticity [68, 69], schemes developed for B-spline surface control volumes [70], and grid deformation schemes employing radial basis functions [71, 72, 73, 74], just to name a few. In this work two different grid movement schemes are employed, both of which are discussed in the following two subsections. First, an algebraic grid deformation approach, which is used for two-dimensional design optimization, is discussed. Afterwards, a radial basis function grid movement scheme, which is more suited for complex multi-block grid configurations and which is utilized within three-dimensional shape optimization, is presented.

### Algebraic Grid Perturbation

On structured grids, algebraic grid movement schemes usually are inexpensive and relatively easy to implement. The algebraic scheme employed within the two-dimensional optimization framework developed in this research is based on a grid perturbation method introduced by Jameson [63], which modifies the current location of the volume grid points based on the known surface grid point perturbations. The method propagates the displacement of the surface nodes into the interior domain along the respective normal grid lines. First, the arc length between a surface point and the grid point at the edge of the computational domain is calculated along the grid line that connects both points. Then the grid points at each location along this grid line are attenuated proportional to their arc length distance from the surface point and the total arc length between the surface and the computational domain boundary. For a single-block grid, the algorithm can be described as

$$x_{1,i,j}^{\mathrm{new}} = x_{1,i,j}^{\mathrm{old}} + C_{ij}\big(x_{1,i,1}^{\mathrm{new}} - x_{1,i,1}^{\mathrm{old}}\big) \quad \text{for} \quad j = 2, \dots, j_{\max} \ , \tag{5.11}$$

$$x_{2,i,j}^{\mathrm{new}} = x_{2,i,j}^{\mathrm{old}} + C_{ij}\big(x_{2,i,1}^{\mathrm{new}} - x_{2,i,1}^{\mathrm{old}}\big) \quad \text{for} \quad j = 2, \dots, j_{\max} \ , \tag{5.12}$$

where the grid is assumed to be oriented such that a point $(i, 1)$ is located on the surface and a node $(i, j_{\max})$ represents a point at the boundary of the computational domain. $i$ is the current grid index which runs along the geometry and the $ij$-th entry of the matrix $\mathbf{C}$ can be defined as

$$C_{ij} = 1 - (3 - 2\mathcal{N}_{ij})\mathcal{N}_{ij}^2 \ , \tag{5.13}$$

where $\mathcal{N}_{ij}$ is the ratio of the arc length from the surface to the current grid point and the total arc length from the surface to the edge of the computational domain along the grid point, i.e.

$$\mathcal{N}_{ij} = \frac{\sum_{l=2}^{j} \sqrt{(x_{1,i,l} - x_{1,i,l-1})^2 + (x_{2,i,l} - x_{2,i,l-1})^2}}{\sum_{l=2}^{j_{\max}} \sqrt{(x_{1,i,l} - x_{1,i,l-1})^2 + (x_{2,i,l} - x_{2,i,l-1})^2}} \ . \tag{5.14}$$

The grid sensitivities $\partial \mathbf{X_v}/\partial \mathbf{X_s}$ for this algebraic deformation scheme are obtained straightforward from linearizing Eqn. (5.12), thus,

$$\delta \mathbf{X_{vs}} = \mathbf{C}\ . \tag{5.15}$$

This simple grid perturbation scheme has been found to be very robust, yet, it possesses some major drawbacks as well. The simplicity of the method lies in the effortlessness of producing new grid point locations along a grid line. However, this advantage disappears in case of unstructured grids due to the lack of a continuous grid line extending from the surface to the boundary of the computational domain. Even for multi-block solvers, which typically have to handle various different grid configurations, this efficient property of the grid perturbation method is mostly removed. Implementing the algebraic deformation scheme in a more general way, increases the computational cost of the method and specifically the complexity of the implementation. For example, various blocks with differently oriented coordinate systems might exist between a surface point and the corresponding boundary point. To be applicable to the two-dimensional multi-block solver used within this research, the algorithm described above is adjusted. First, the matching pair of surface/far-field grid points are determined with the help of the block connectivity information. Then, the interior grid points located at the faces of each block are modified; the block face deformations are obtained from expressions similar to those presented above. At this step, special care is required to ensure that the face and vertex points which share various blocks are perturbed identically on each of these blocks. Afterwards, the interior block points are perturbed based on the modifications of the face nodes.

### Grid Perturbation Based on Radial Basis Functions

Extending the algebraic grid perturbation scheme presented in the previous subsection to three dimensions, while keeping it general enough to handle complex three-dimensional multi-block grids, would further increase the complexity of the implementation and the computational cost. Thus, the algebraic scheme is discarded in favour of a more general grid movement scheme based on radial basis functions.

Radial basis functions (RBF) are used in many applications to interpolate scattered data, for a grid perturbation scheme the interpolation function may be described as

$$\Delta X_{v,n} = \sum_{i=1}^{N_{\text{RBF}}} \beta_i \phi\big(||X_{v,n} - X_{\text{RBF},i}||\big) \ , \tag{5.16}$$

where $\Delta X_{v,n}$ is the interpolated displacement of the $n$-th volume mesh point $X_{v,n}$, $N_{\text{RBF}}$ is the number of RBF points, $\beta_i$ are coefficients that are solved to satisfy the condition that the interpolation function recovers the known displacement at the RBF points $X_{\text{RBF},i}$, $\phi$ is the basis function, and $||X_{v,n} - X_{\text{RBF},i}||$ is the Euclidean distance between point $X_{v,n}$ and the RBF point $X_{\text{RBF},i}$ defined as

$$||X_{v,n} - X_{\text{RBF},i}|| = \sqrt{(x_{1,v,n} - x_{1,\text{RBF},i})^2 + (x_{2,v,n} - x_{2,\text{RBF},i})^2 + (x_{3,v,n} - x_{3,\text{RBF},i})^2} \ . \tag{5.17}$$

Wendland's C2 basis function with support radius is adopted in this work because it provides a good combination of mesh quality and matrix conditioning. It is given by

$$\phi(\xi) \ = \ (1-\xi)^4(4\xi+1) \qquad \text{for} \quad \xi < 1.0 \ , \tag{5.18}$$

$$\phi(\xi) \ = \ 0 \qquad \text{for} \quad \xi \geq 1.0 \ , \tag{5.19}$$

where $\xi = ||X_{v,n} - X_{\text{RBF},i}||/r_{\text{SRAD}}$ with $r_{\text{SRAD}}$ being the support radius. Any grid point outside this region is not displaced.

The RBF interpolation function (5.16) can be conveniently expressed through matrices when considering the entire set of grid points. First, the RBF coefficients $\boldsymbol{\beta}$ must be solved for the coordinates $x_1$, $x_2$, and $x_3$ such that the known displacements of the RBF points $\Delta\mathbf{X}_{\text{RBF}}$ are recovered, i.e.

$$\Delta\mathbf{X}_{\text{RBF}} = \mathbf{M}_{\text{RBF}}\boldsymbol{\beta} \ . \tag{5.20}$$

The displacements of the volume grid points $\Delta\mathbf{X_v}$ can then be expressed as

$$\Delta\mathbf{X_v} = \mathbf{A}_{\text{RBF}}\boldsymbol{\beta} = \mathbf{A}_{\text{RBF}}\mathbf{M}_{\text{RBF}}^{-1}\Delta\mathbf{X}_{\text{RBF}} \ . \tag{5.21}$$

The matrices are defined as

$$\mathbf{M}_{\text{RBF}} = \begin{bmatrix} \phi_{r1,r1} & \phi_{r1,r2} & \cdots & \phi_{1,N_{\text{RBF}}} \\ \phi_{r2,r1} & \phi_{r2,r2} & \cdots & \phi_{r2,N_{\text{RBF}}} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{N_{\text{RBF}},r1} & \phi_{N_{\text{RBF}},r2} & \cdots & \phi_{N_{\text{RBF}},N_{\text{RBF}}} \end{bmatrix}_{N_{\text{RBF}} \times N_{\text{RBF}}} , \tag{5.22}$$

and

$$\mathbf{A}_{\text{RBF}} = \begin{bmatrix} \phi_{v1,r1} & \phi_{v1,r2} & \cdots & \phi_{v1,N_{\text{RBF}}} \\ \phi_{v2,r1} & \phi_{v2,r2} & \cdots & \phi_{v2,N_{\text{RBF}}} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{N_{\text{RBF}},r1} & \phi_{N_{\text{RBF}}r2} & \cdots & \phi_{N_{\text{RBF}},N_{\text{RBF}}} \end{bmatrix}_{N_{\text{v}} \times N_{\text{RBF}}} . \tag{5.23}$$

Here, $\phi_{rn,rj}$ represents $\phi(||X_{\text{RBF},n} - X_{\text{RBF},j}||/r_{\text{SRAD}})$, i.e. the basis function between the RBF point $X_{\text{RBF},n}$ and the RBF point $X_{\text{RBF},j}$, while $\phi_{vn,rj}$ stands for the basis function between the volume point $X_{v,n}$ and the RBF point $X_{\text{RBF},j}$, cf. Eqns. (5.16) and (5.17). $N_{\text{v}}$ is the number of volume points and $\Delta\mathbf{X}_{\text{RBF}}$ is the vector of known RBF point displacements. An RBF interpolation function must be solved for every coordinate direction $x_1$, $x_2$, and $x_3$.

If the RBF-based grid deformation scheme is utilized for grid perturbation during the gradient calculation, then formulation (5.9) expands to

$$\frac{\mathrm{d}I}{\mathrm{d}\boldsymbol{\alpha}} = -\boldsymbol{\psi}^T \frac{\partial \mathbf{R}}{\partial \mathbf{X_v}} \frac{\partial \mathbf{X_v}}{\partial \mathbf{X}_{\text{RBF}}} \frac{\partial \mathbf{X}_{\text{RBF}}}{\partial \mathbf{X_s}} \frac{\partial \mathbf{X_s}}{\partial \boldsymbol{\alpha}} , \tag{5.24}$$

where in the above equation, $\partial\mathbf{X_v}/\partial\mathbf{X}_{\text{RBF}}$ represents the RBF grid sensitivity, while $\partial\mathbf{X}_{\text{RBF}}/\partial\mathbf{X_s}$ is a $N_{\text{RBF}} \times N_s$ matrix which maps the designated RBF surface grid points to all grid points on the surface. In block form this second sensitivity can be represented as

$$\frac{\partial \mathbf{X}_{\text{RBF}}}{\partial \mathbf{X_s}} = \begin{bmatrix} \mathbf{I}_{\text{RBF}} \\ 0 \end{bmatrix} , \tag{5.25}$$

where $\mathbf{I}_{\text{RBF}}$ is an identity matrix of size $N_{\text{RBF}} \times N_{\text{RBF}}$. Linearizing Eqns. (5.20) and

(5.21) provides the RBF grid sensitivity as

$$\frac{\partial \mathbf{X_v}}{\partial \mathbf{X}_{\text{RBF}}} = \mathbf{A}_{\text{RBF}} \mathbf{M}_{\text{RBF}}^{-1} \ . \tag{5.26}$$

Figure 5.1 illustrates the typical multi-block grid configuration used to spatially discretize the various blade rows investigated within this research. The task of an optimization would be to modify the blade geometry (color: red), and thus the blade surface points, such that the design objective is met. At the same time, both the hub geometry and the casing (color: blue) are frozen and are not part of the design space. The remaining outer boundaries of the computational domain are made of an inlet and outlet plane (color: green) and two corresponding periodic boundaries (color: orange, for clarity only one periodic boundary is depicted). Accordingly, all grid points on the blade surface can be defined as RBF points and their displacements become the RBF displacements. As the grid is deformed, it has to be ensured that not only the hub and casing but also all other outer boundaries are not perturbed. Consequently, all grid points located on these planes can be added to the RBF matrix as well, with their displacements forced to zero. Thus, in its full and most simple form, the RBF grid deformation scheme assumes every blade surface grid point and every point located on the other boundaries of the computational domain to be an RBF point. The known displacements of every blade surface grid point are then accurately recovered, while the scheme would prevent the hub geometry and the casing as well as the outer planes from being deformed. In this form, the deformation scheme is robust and the grid quality is well maintained. Unfortunately, when applied to highly-resolved three-dimensional grids, the method becomes expensive and requires a vast amount of memory, since a geometry defined by many boundary grid points would imply an equally large amount of RBF points. Adding all these points to the RBF matrix $\mathbf{M}_{\text{RBF}}$, the system of equations simply becomes too expensive to solve in a reasonable amount of time. Speeding up the RBF grid movement scheme can be achieved by reducing the number of RBF points and, hence, by neglecting some boundary grid points. However, at the same time this deteriorates the accuracy and robustness of the deformation scheme. Thus, a compromise between computational cost and grid quality has to be made and the trade-off has to be investigated carefully.

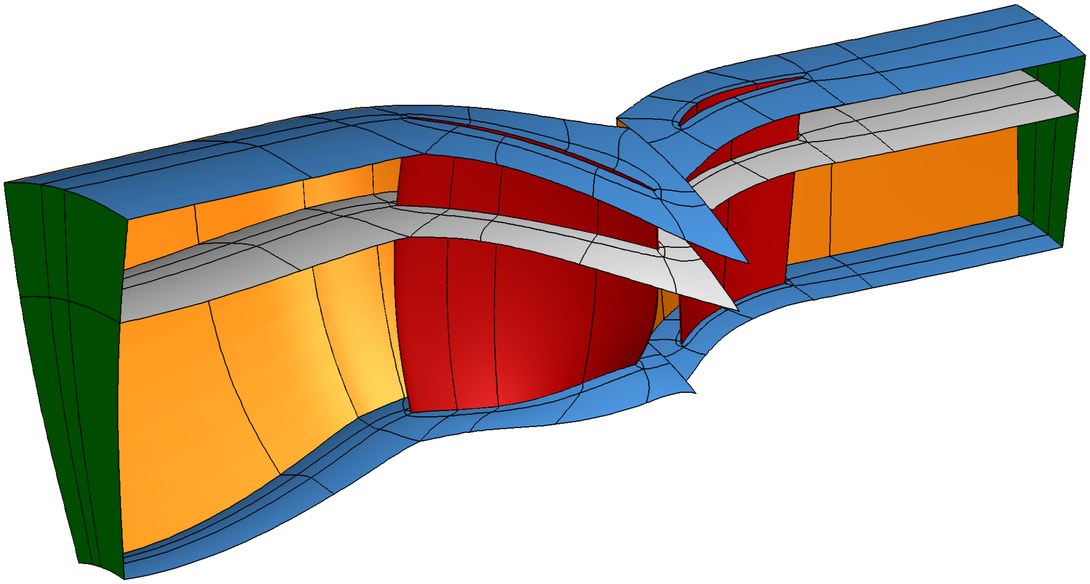

Figure 5.1: Multi-block grid structure for a single-stage transonic compressor.

In this research work, several measures are undertaken to reduce the computational cost to a reasonable level while compromising the quality of the grid perturbation as little as possible. As it can be seen from Figure 5.1, both periodic boundaries and especially the inlet and outlet planes are located relatively far from the blade surface that is redesigned. Changes in the blade geometry are propagated into the interior domain through the RBF scheme but also attenuate more and more as further they travel. Accordingly, blade shape modifications produce only very small and almost negligible grid perturbations close to the outer boundaries. Therefore, it is not necessary to include every outer boundary point as an RBF point. Investigations have shown that it is sufficient to only include the four vertex points of each inlet or outlet plane into the RBF matrix. Furthermore, at a periodic boundary it usually is enough to only load every eighty to twelfth grid point in each coordinate direction as an RBF point. The displacements of the excluded outer boundary grid points are simply forced to zero. Although they do not provide any information to the deformation scheme, the grid quality remains high and the introduced inaccuracies are negligible. Moreover, tests have shown that it usually is not necessary to define every surface point of the modified geometry as an RBF point. Loading only every third blade surface point into the RBF matrix produces grids with comparable qualities but at considerable less cost. On viscous grids the grid quality suffers noticeably if even fewer blade surface points are included into the deformation scheme. However, on inviscid grids the number of RBF points can be reduced even further by loading only every fourth or fifth blade surface perturbation. Inviscid problems do not require such highly-resolved grids near a solid body. Hence, the distance from the surface to the first interior grid point is much larger and consequently the aspect ratio is considerably smaller than it is for viscous grids. This poses less strict quality requirements on the grid deformation scheme. Similarly, the number of RBF points included from the surfaces defining the hub and casing geometry can be reduced; as larger the distance to the modified blade, as fewer points will be loaded into the RBF matrix.

However, the grid movement scheme can not only be accelerated by reducing the size of the RBF matrix $\mathbf{M}_{\text{RBF}}$, but also by minimizing the size of the volume point matrix $\mathbf{A}_{\text{RBF}}$ and by parallelizing the computations. Therefore, the RBF grid deformation scheme is split into two stages. On the first stage, only the interior domain points defining the faces of each block are perturbed based on the information provided from the reduced RBF matrix. This operation is performed on the root processor. Once the skeleton of the multi-block grid configuration is modified, cf. Figure 5.2, the perturbations are communicated to each processor via MPI. Afterwards, the block interior points are perturbed based on its block face deformations calculated on the first stage. Hence, on this second stage, the face perturbations become the RBF points which define a new RBF matrix. Thus, all information required is locally available and provided from the block to be perturbed. Consequently, the second stage can be parallelized and every processor only perturbs the blocks it is assigned to from the pre-processor load balancer. The RBF matrix of the second stage can be reduced as well by only loading every third or fourth perturbed face point into the new RBF matrix.

This completes the RBF deformation scheme. Figures 5.2 and 5.3 show an example perturbation and illustrate the two stages of the grid movement scheme. A bump is placed on the rotor suction side at 70% span and 50% of the rotor chord length; the grey-colored plane in Figure 5.1 locates the 70% span section within the compressor configuration. For demonstration purposes the size of the bump ampli-

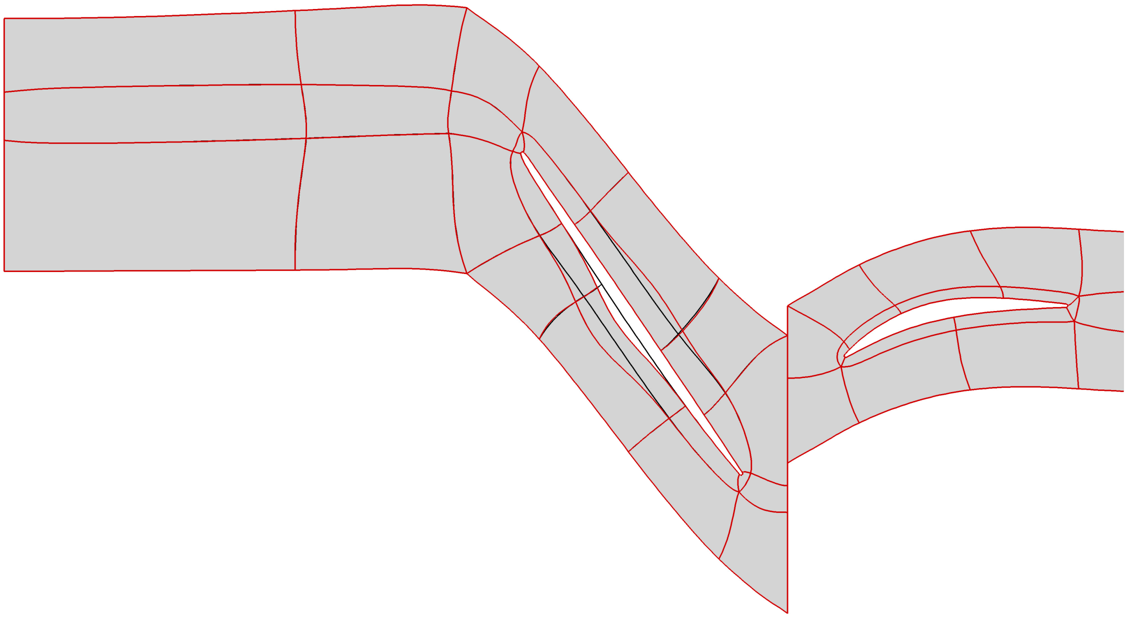

Figure 5.2: Perturbed skeleton after the first stage of the RBF grid perturbation scheme.

tude is exaggerated. The blade geometry usually experiences smaller modifications throughout an optimization. Figure 5.2 depicts the multi-block configuration after the first stage of the grid perturbation scheme, i.e. the skeleton is already adjusted (color: red) while the interior grid points are not yet modified, and compares it with the baseline skeleton (color: black). The final modified grid is illustrated in Figure 5.3. The enlargements demonstrate that the grid quality of the perturbed grid remains high despite the neglected RBF points. However, it is important to note, that the measures described to speed up the deformation scheme cannot be generalized and would have to be adjusted on different grid configurations.

## 5.2 Optimization Strategy

Once the objective function gradient is obtained, one can generally employ any optimization method to find an improved airfoil shape. Experience has shown that even a simple optimization method, such as the Steepest Descent (SD) method in combination with a fixed step length, is able to provide reasonable results for unconstrained

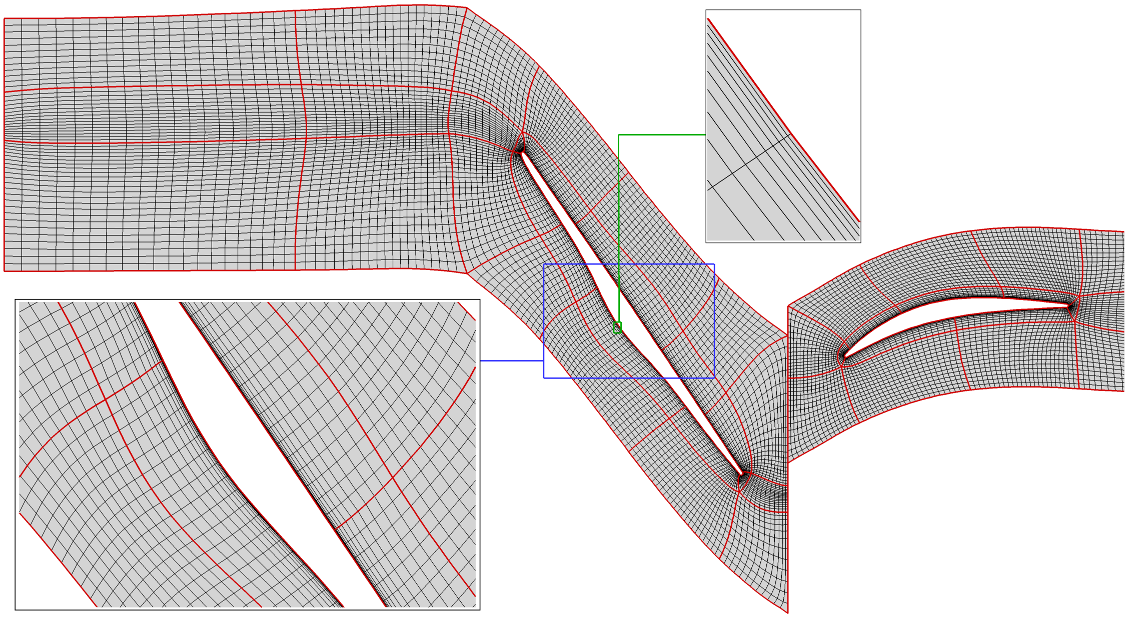

Figure 5.3: Perturbed grid after completion of the RBF grid perturbation scheme.

optimization problems. However, difficulties occur when the SD method is applied to constrained or multi-objective design problems. In such a case, the objective function is usually defined as a weighted sum of the constraints and the main objective. The weights have to be determined individually for each constraint and every design problem by trial and error which can be a time-consuming task, cf. [19, 32].

A sequential quadratic programming (SQP) algorithm instead offers a more elegant way to treat constraints in a design problem. Optimization within an SQP framework avoids the need to determine the penalty coefficients by including the constraints directly into the design problem. Considering a general optimization problem

$$\begin{aligned} \min \quad & I(\boldsymbol{\alpha}) \quad \text{w.r.t.} \quad \boldsymbol{\alpha} \in \mathbb{R}^n \\ \text{subject to} \quad & c_j(\boldsymbol{\alpha}) = 0, \quad j = 1, \ldots, m_1 \\ & \hat{c}_j(\boldsymbol{\alpha}) \geq 0, \quad j = 1, \ldots, m_2 \end{aligned} \tag{5.27}$$

where $I(\boldsymbol{\alpha})$ is the objective function to be minimized, $c_j(\boldsymbol{\alpha})$ and $\hat{c}_j(\boldsymbol{\alpha})$ represent the equality and inequality constraints respectively, and $\boldsymbol{\alpha}$ are the design variables, the

SQP algorithm determines an appropriate search direction by the following procedure. First, inequality constraints can be converted into equalities by introducing a set of slack variables $\boldsymbol{\epsilon}$, i.e. $\hat{c}_j(\boldsymbol{\alpha}) - \epsilon_j = c_j(\boldsymbol{\alpha}) = 0$. For example, a linear inequality constraint $l_1 \leq \hat{c}_1(\boldsymbol{\alpha}) \leq l_2$ is replaced by $\hat{c}_1(\boldsymbol{\alpha}) - \epsilon_1 = 0$ together with the bounded slack $l_1 \leq \epsilon_1 \leq l_2$. Second, a Lagrangian function $\boldsymbol{\mathcal{L}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = I(\boldsymbol{\alpha}) - \boldsymbol{\lambda}^T \mathbf{c}(\boldsymbol{\alpha})$ is defined. Third, taking the derivative of the Lagrangian function yields the first-order Karush-Kuhn-Tucker (KKT) conditions

$$\mathbf{F}_{\mathcal{L}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}} \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} \end{pmatrix} = \begin{pmatrix} \nabla I(\boldsymbol{\alpha}) - \mathbf{A}(\boldsymbol{\alpha})^T \boldsymbol{\lambda} \\ \mathbf{c}(\boldsymbol{\alpha}) \end{pmatrix} = 0, \tag{5.28}$$

where $\mathbf{A}(\boldsymbol{\alpha})^T = \nabla \mathbf{c}(\boldsymbol{\alpha})$. Taking again the derivative provides

$$\mathbf{F}'_{\mathcal{L}}(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = \begin{pmatrix} \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\alpha}} & \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\lambda}} \\ \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\lambda} \partial \boldsymbol{\alpha}} & \frac{\partial^2 L}{\partial \lambda \partial \boldsymbol{\lambda}} \end{pmatrix} = \begin{pmatrix} \nabla^2_{\boldsymbol{\alpha\alpha}} \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\lambda}) & -\mathbf{A}(\boldsymbol{\alpha})^T \\ \mathbf{A}(\boldsymbol{\alpha}) & 0 \end{pmatrix}. \tag{5.29}$$

Then, the search direction $\boldsymbol{\mathcal{P}}$ is obtained by applying Newton's method to the KKT conditions, i.e. $\boldsymbol{\mathcal{P}} = -\mathbf{F_L}(\boldsymbol{\alpha}, \boldsymbol{\lambda}) \cdot \mathbf{F'_L}(\boldsymbol{\alpha}, \boldsymbol{\lambda})$, where $\boldsymbol{\mathcal{P}} = [p_{\boldsymbol{\alpha}}, p_{\boldsymbol{\lambda}}]^T$.

In this research, SNOPT [75], a general-purpose software package for constrained optimization embedded into an SQP framework, is used to determine an improved airfoil shape. Slack variables are used to convert inequality constraints to equalities. The search direction is determined by the Quasi-Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, while a general line search technique is used to find an appropriate step length, cf. Gill et al. [76] for more details. The gradients of the objective function and each constraint have to be provided to SNOPT. Hence, $m+1$ adjoint solutions, where $m = m_1 + m_2$ is the number of all equality and inequality constraints in the design problem, are necessary to determine the new search direction.

# Chapter 6

# Results

In this chapter, the functionality of the proposed optimization method is demonstrated. First, the flow solver is validated against experimental data available for some classical flow problems, cf. section 6.1. In the remainder of this chapter, two different transonic axial flow compressor configurations are then optimized, where both configurations are redesigned considering various flow conditions and dissimilar geometrical set-ups, cf. sections 6.2 and 6.3. In each of these two sections, first the respective design case is introduced. Then, the flow and adjoint solutions are discussed and evaluated. Next, the adjoint solver and gradient calculation routines are validated by comparing the adjoint sensitivities of various functionals with finite-difference gradients obtained from the primal flow solver. Lastly, several unconstrained and constrained optimization cases are discussed, in which the respective test configuration is redesigned in order to increase the isentropic efficiency.

It is worthwhile to mention that although the proposed optimization framework is only applied to compressor configurations, it can readily be applied to multi-row turbines as well.

## 6.1 Flow Solver Validation

Before moving towards the actual design cases, in this preliminary section the flow solver is validated. To do so, numerical results obtained for three classical flow problems are presented and compared against experimental data.

### 6.1.1 Flat Plate

First, the implemented SST turbulence model is validated. Therefore, the incompressible turbulent flow over a flat plate, as described in [77], is investigated. The flow conditions are as follows: the freestream Mach number is $M = 0.2$, the freestream static temperature takes on the value $T = 294,44K$, the freestream static pressure is given by $101.35kPa$, and the angle of attack is $\alpha = 0.0°$. Furthermore, the length of the flat plate is 5.09 meters. This particular flow problem was first investigated experimentally by Wieghardt and Tillmann [78]; their measured data serves at the reference for this validation case.

A Cartesian grid is used to model the flow problem numerically, cf. Figure 6.1. 111 grid points are distributed in the axial direction and 81 grid points are used to discretize the domain normal to the viscous wall. The grid is clustered in the streamwise direction near the leading edge of the plate, which is located at $x = 0.0$, to resolve the locally occurring high flow gradients accurately. At the same time the gird is clustered normal to the plate to ensure that the boundary layer is sufficiently resolved but that an unnecessary amount of cells in the freestream region is avoided. The non-dimensional wall distance is $y^+_{\text{avg}} < 1.0$. To obtain a uniform velocity profile at the leading edge location of the plate, the fourteen upstream grid points at $h = 0.0$ are treated as an inviscid wall. As depicted in Figure 6.2, both the flow solver and the turbulence model show a slow convergence rate, which is due to the incompressible nature of the low Mach number flow and since neither preconditioning nor multigrid is employed to accelerate the solution process to steady-state. Although after 50,000 iterations the flow solution is only converged by a few orders, the calculated flow field is already sufficiently accurate, cf. Figures 6.3 and 6.4. In these plots, the

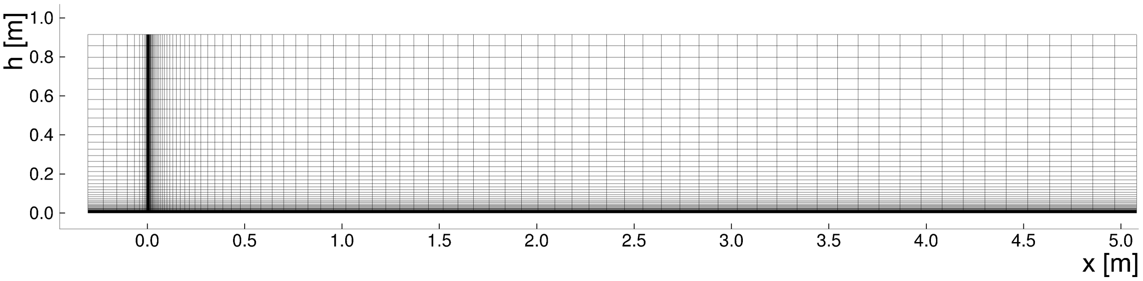

Figure 6.1: Computational grid for the flat plate test case.

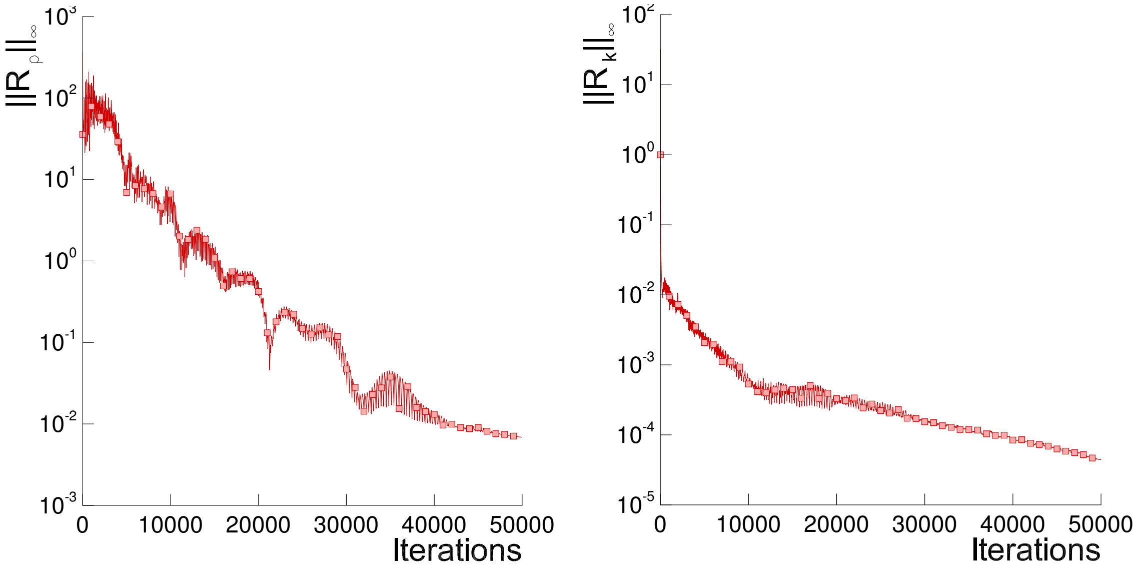

Figure 6.2: Convergence of maximum density residual (left) and maximum turbulent kinetic energy residual (right), flat plate.

predicted velocity profiles are compared with the experimentally measured data from Wieghardt and Tillmann [78] at several axial locations; the dimensionless velocity $u^+$, defined as $u^+ = u/u_\tau$ with the friction velocity $u_\tau = \sqrt{\tau_w/\rho}$ and $\tau_w$ being the wall shear stress, is plotted against the dimensionless wall distance $y^+ = u_\tau y/\nu$, where $y$

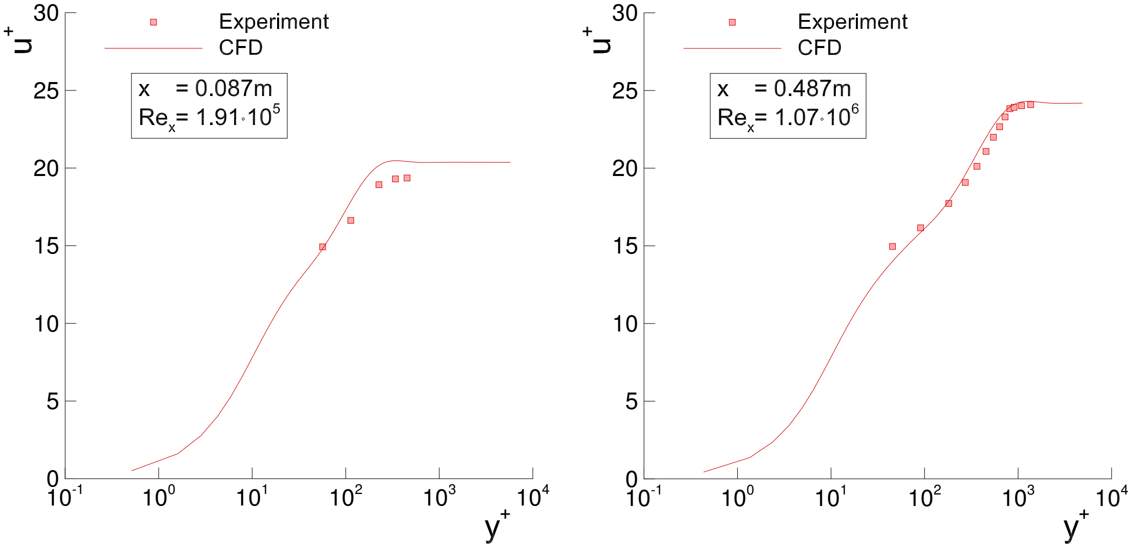

Figure 6.3: Velocity profiles along the flat plate.

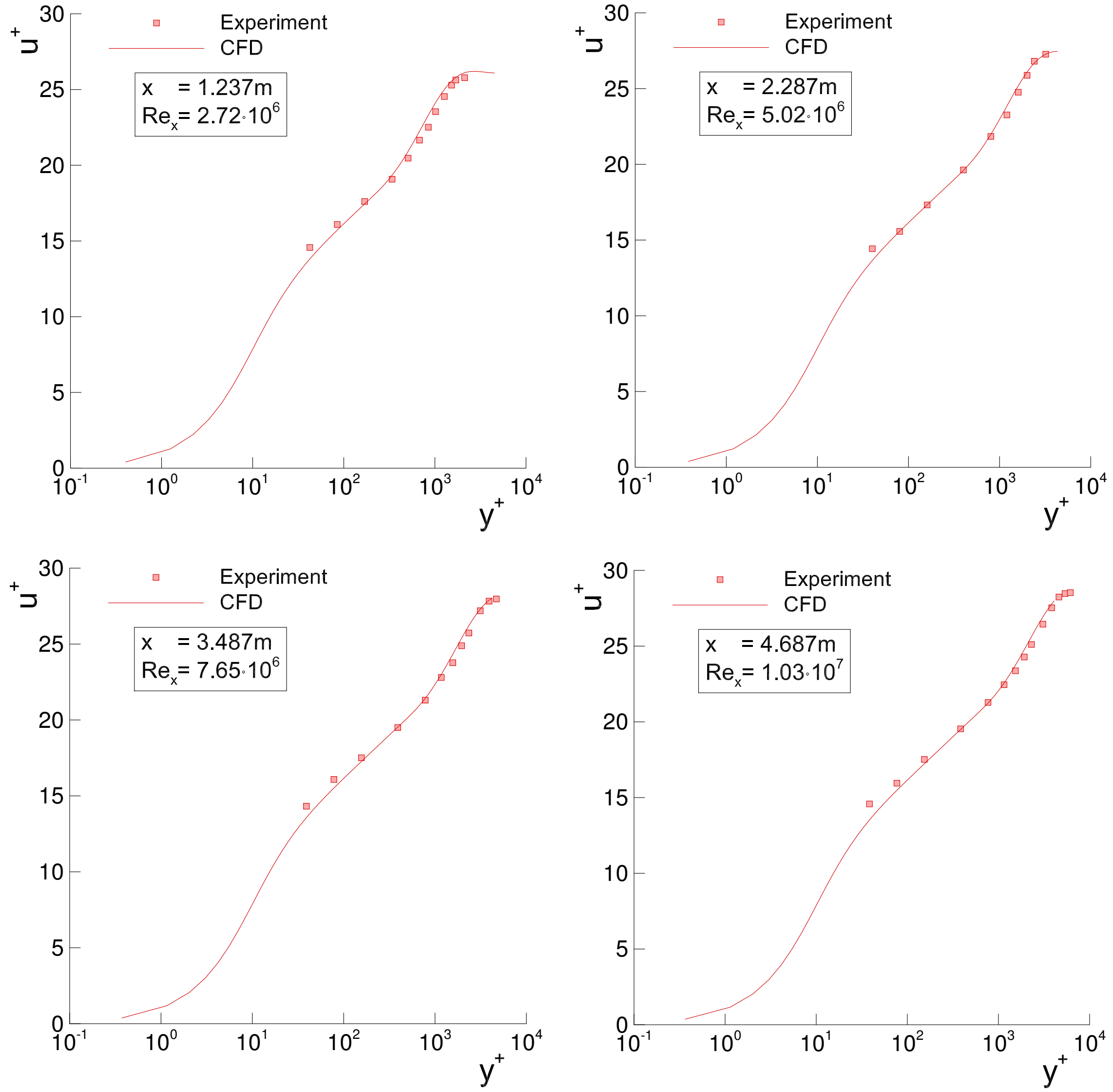

Figure 6.4: Velocity profiles along the flat plate.

is the distance to the wall. It can be observed that the numerical solution is in good agreement with the experiment and that the turbulence model correctly predicts the growth of the boundary layer. Hence, the results of this validation case demonstrate that the flow solver and the SST model accurately calculate turbulent boundary layer flows.

### 6.1.2 Sajben Transonic Diffuser

The Sajben transonic diffuser serves as the second validation case. The configuration was introduced by Sajben and co-workers, who studied the set-up experimentally in an effort to investigate the impact of self-excited unsteady fluctuations at inlets on the transonic flow system within a diffuser typical for supersonic aircraft propulsion systems, cf. [79, 80, 81, 82]. The study produced a vast amount of experimental data at a variety of flow conditions, including static wall pressure distributions and velocity profiles at numerous streamwise locations, such that the Sajben diffuser became a popular test case for the verification and validation of CFD codes investigating internal compressible flows, cf. [83].

The computational grid for the two-dimensional convergent-divergent Sajben diffuser is shown in Figure 6.5 (top); 81 grid points are used to discretize the diffuser in the axial direction and 51 grid points are distributed in the radial direction. Here $x$ is the axial coordinate and $h^*$ is the height of the throat. A more detailed description of the geometry is provided in [83]. The flow field is assumed to be fully-turbulent with an inlet Reynolds number of $Re_{\text{in}} = 7.4 \cdot 10^5$. The wall spacing is chosen such that the turbulent boundary layer is resolved to $y^+_{\text{avg}} = 0.9$. The inlet and outlet boundary conditions for the Sajben diffuser weak shock case are provided in Table 6.1; both upper and lower viscous walls are assumed to be adiabatic.

The prescribed pressure ratio $p_{s,\text{out}}/p_{t,\text{in}} = 0.82$ results in the transonic flow field depicted in Figure 6.5 (bottom). The subsonic flow at the inlet, $M_{\text{in}} = 0.46$, accelerates through the convergent section of the configuration, reaching sonic speed at the throat $x/h^* = 0.0$, and accelerates further to reach a peak Mach number of approximately $M_{\text{max}} = 1.3$ just before a shock terminates the supersonic zone at

| | |
|---|---|
| $T_{t,\text{in}}$ [K] | 277.78 |
| $p_{t,\text{in}}$ [kPa] | 135.00 |
| $p_{s,\text{out}}$ [kPa] | 110.66 |
| $\alpha_{\text{in}}$ [°] | 0.00 |

Table 6.1: Inlet and outlet boundary conditions for the Sajben transonic diffuser weak shock case.

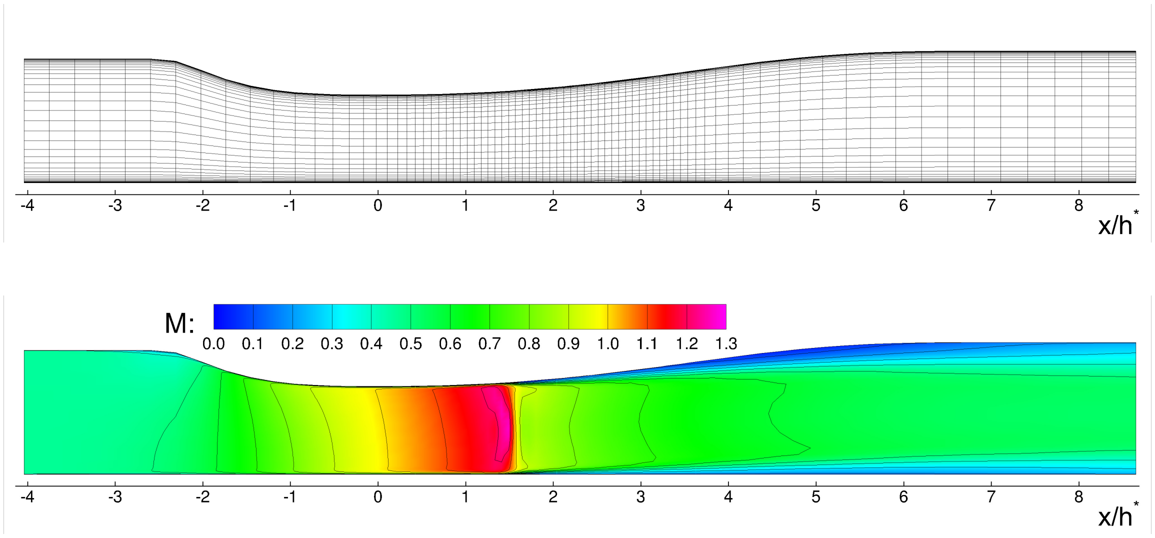

Figure 6.5: Computational grid (top) and Mach number contours (bottom) for the Sajben transonic diffuser weak shock case.

the beginning of the divergent section. From there on, the flow continues to decelerate subsonically until the exit of the diffuser, where an averaged Mach number of $M_{\text{out}} = 0.48$ can be measured. The subsonic flow region downstream the shock contains rapidly growing boundary layers; however, the shock is not strong enough to induce flow separation at the upper wall and the flow remains attached throughout the divergent section of the nozzle. The convergence of the flow solver, represented by the decrease in the $\text{L}_\infty$-norm of the density residual, and the SST turbulence model, illustrated by the reduction of the $\text{L}_\infty$-norm of the residual of the turbulent kinetic energy, are illustrated in Figure 6.6. Figure 6.7 compares the computed static pressure distributions along the top and bottom surfaces of the diffuser with the experimentally measured values. The numerical results compare very well with the experimental data; both the predicted shock location and strength match the experiment almost perfectly. Figure 6.8 plots the calculated velocity profiles at four different axial locations $x/h^*$ downstream of the shock against the experimental measurements. The core velocity and the velocity profile near the lower wall predicted by the flow solver are in high agreement with the experimental data. Only further downstream in the near-wall region of the upper surface a discrepancy between the numerical solution and the experiment can be identified. Similar observations were

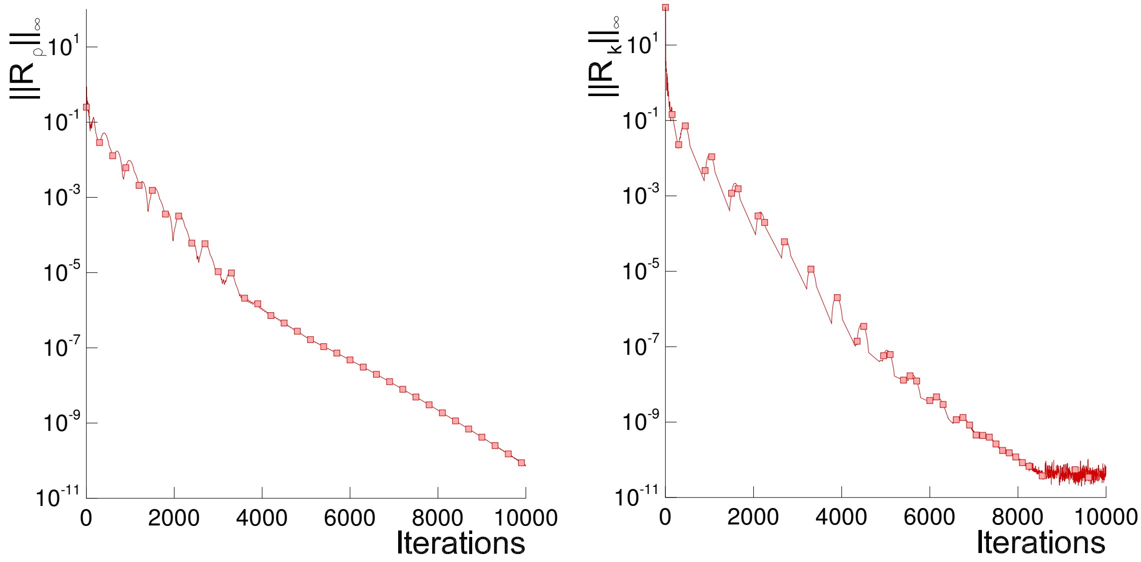

Figure 6.6: Convergence of maximum density residual (left) and maximum turbulent kinetic energy residual (right), Sajben transonic diffuser weak shock case.

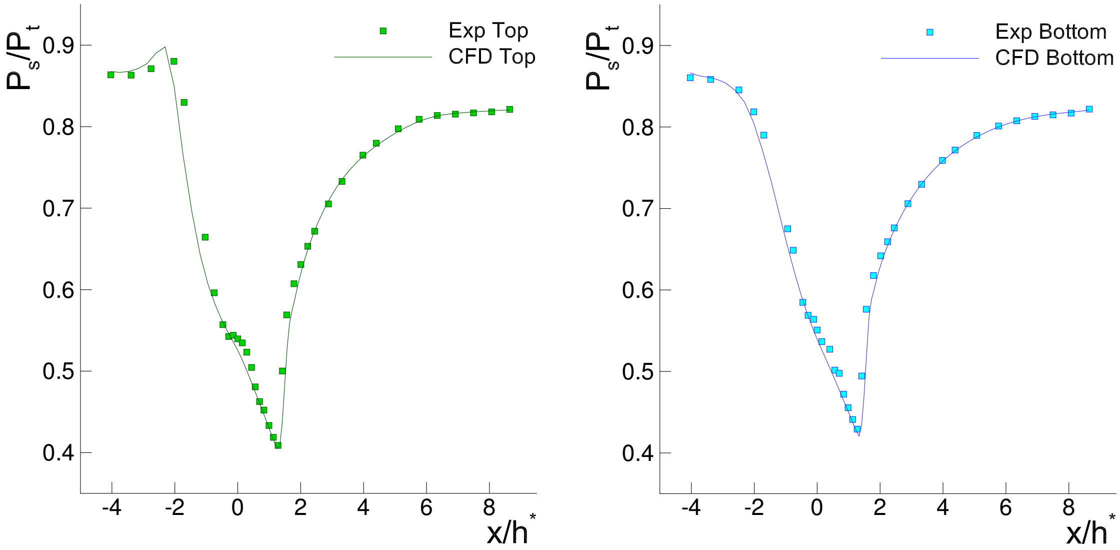

Figure 6.7: Surface pressure distribution along the upper (left) and lower (right) wall, Sajben transonic diffuser weak shock case.

made by other authors including Yoder, cf. [84], who investigated different turbulence models and concluded that the numerical results obtained with the $k$-$\epsilon$ turbulence model more accurately match the experimental data near the upper surface, whilst the SST model better predicts the velocity data near the lower surface of the diffuser.

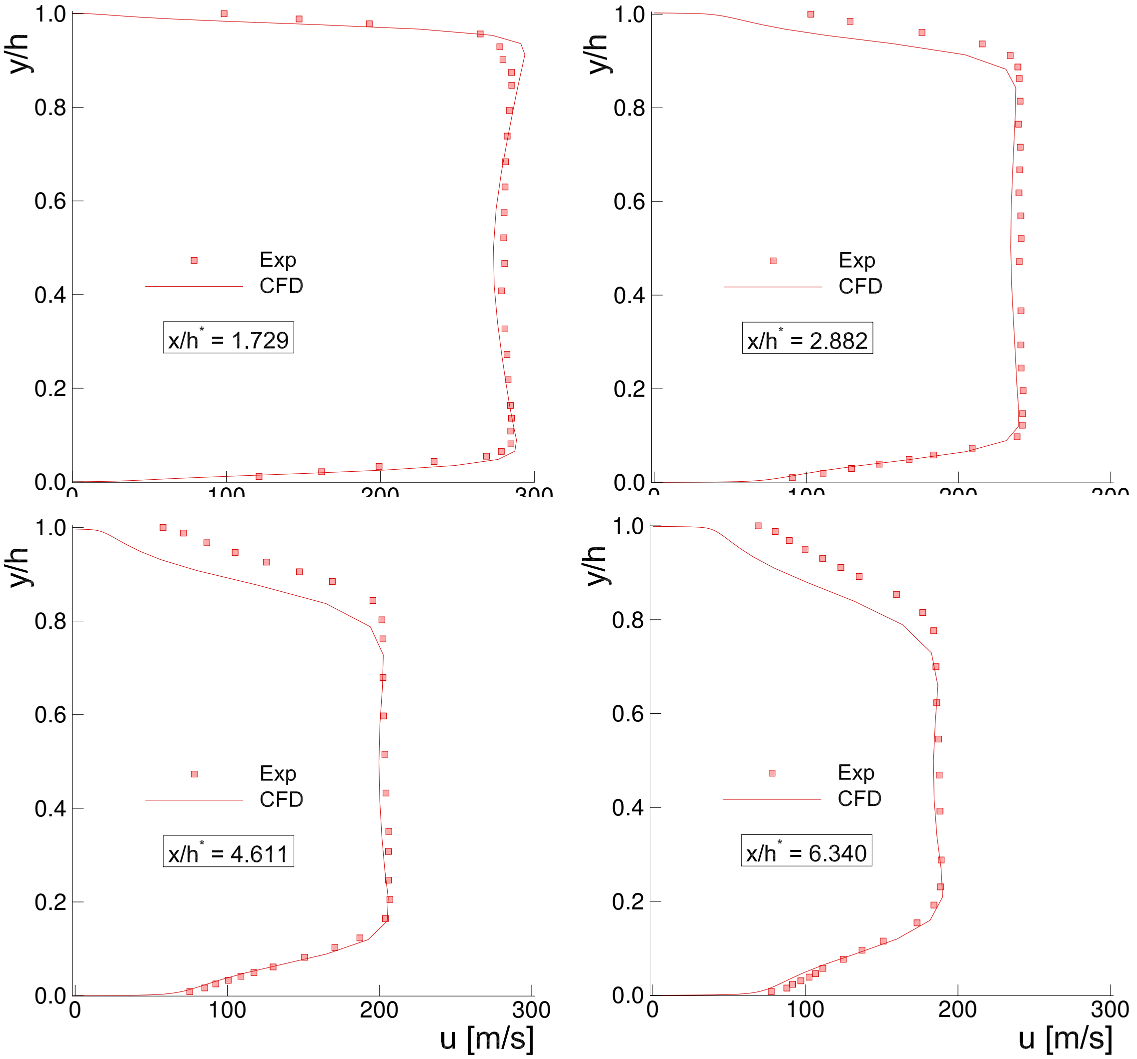

Figure 6.8: Velocity profiles at four axial locations, Sajben transonic diffuser weak shock case.

### 6.1.3 Mark II Turbine Vane

The Mark II turbine vane serves as the third validation case. This configuration was first investigated by Nealy et al. [85], who experimentally studied the heat transfer distribution over the surfaces of highly-loaded turbine nozzle guide vanes under various flow conditions. Nealy and co-workers varied aerodynamic parameters such as the exit Mach number, the Reynolds number, the turbulence intensity, and the wall-to-gas temperature ratio to obtain data over a wide range of gas turbine engine operation points. The detailed measurements were made publicly available with the intend to serve as verification data for numerical simulations. As a result, the internally cooled Mark II turbine vane became a well-established test case which is particularly used to validate conjugate heat transfer models. A detailed description of the configuration can be found in [86].

Figure 6.9 (left) depicts the block-structured grid of the two-dimensional cascade. The entire computational domain consists of 12,200 grid points, where the blade surface is represented by 177 grid points and the channel section between two adjacent vanes is discretized by 97 grid points. The turbulent boundary layer is accurately resolved with a non-dimensional wall distance of $y^+_{\mathrm{avg}} = 0.1$. In the presented validation study, the Mark II cascade is investigated for the aerodynamic set-up described in [86] as case 4311; Table 6.2 provides the corresponding inlet and outlet boundary conditions. Furthermore, the viscous blade surface is assumed to be adiabatic, i.e. this validation study is purely aerodynamic and not the subject of heat transfer investigations. With an inlet Reynolds number of $Re_{\mathrm{in}} = 4.5 \cdot 10^5$, the flow field is fully-turbulent.

Figure 6.9 (right) shows the Mach number contours for the calculated flow solution. Despite a relatively large pressure ratio of $p_{s,\mathrm{out}}/p_{t,\mathrm{in}} = 0.60$, the inflow velocity is low and results in an average Mach number of approximately $M_{\mathrm{in}} = 0.2$. Starting from the leading edge stagnation point, the flow accelerates along the highly-curved blade suction side to supersonic speeds until a strong shock, located at approximately 40% axial chord length, abruptly decelerates the flow to subsonic regimes. The strong adverse pressure gradient and the high blade curvature cause the flow to separate.

| | |
|---|---|
| $T_{t,\mathrm{in}}$ [K] | 803.00 |
| $p_{t,\mathrm{in}}$ [kPa] | 276.47 |
| $p_{s,\mathrm{out}}$ [kPa] | 165.88 |
| $\alpha_{\mathrm{in}}$ [°] | 0.00 |

Table 6.2: Inlet and outlet boundary conditions for the Mark II turbine vane case 4311.

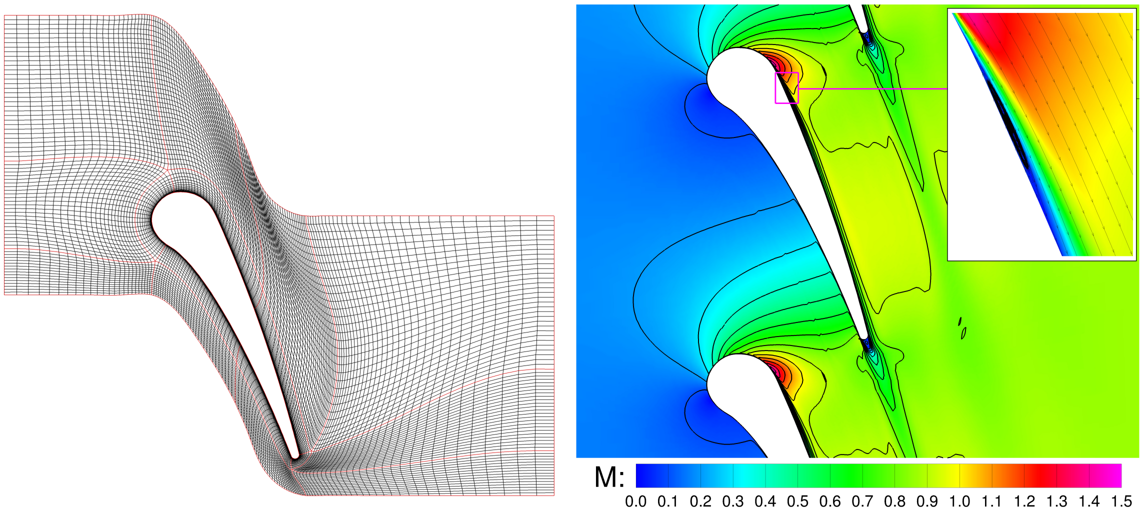

Figure 6.9: Computational grid (left) and Mach number contours (right) for the Mark II turbine vane.

However, as observed in the enlargement of Figure 6.9 (right), the flow reattaches after a short distance. Downstream of the shock, the flow remains subsonic until the exit, where a Mach number of $M_{\text{out}} = 0.9$ is reached. Both, the flow solver and the turbulence model converge by more than nine orders within less than 1500 multigrid cycles, cf. Figure 6.10. Comparing the calculated surface pressure distribution along

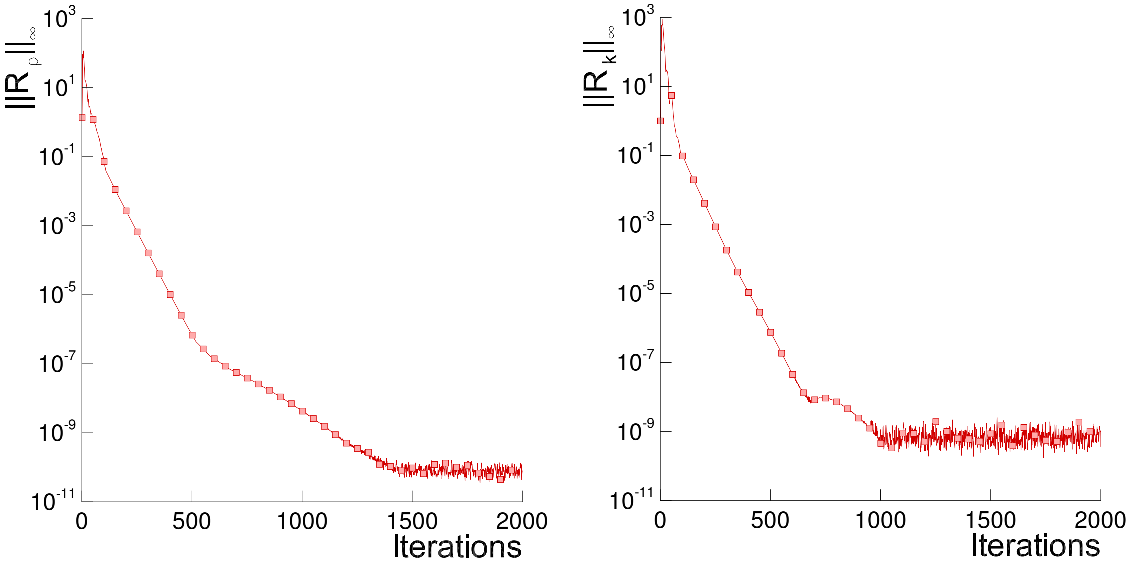

Figure 6.10: Convergence of maximum density residual (left) and maximum turbulent kinetic energy residual (right), Mark II turbine vane.

the vane's suction and pressure side with the experimental data shows that the flow solver predicts the flow field very accurately, cf. Figure 6.11. The shock location as well as the pressure recovery in the aft part of the blade are captured and agree well with the measurements.

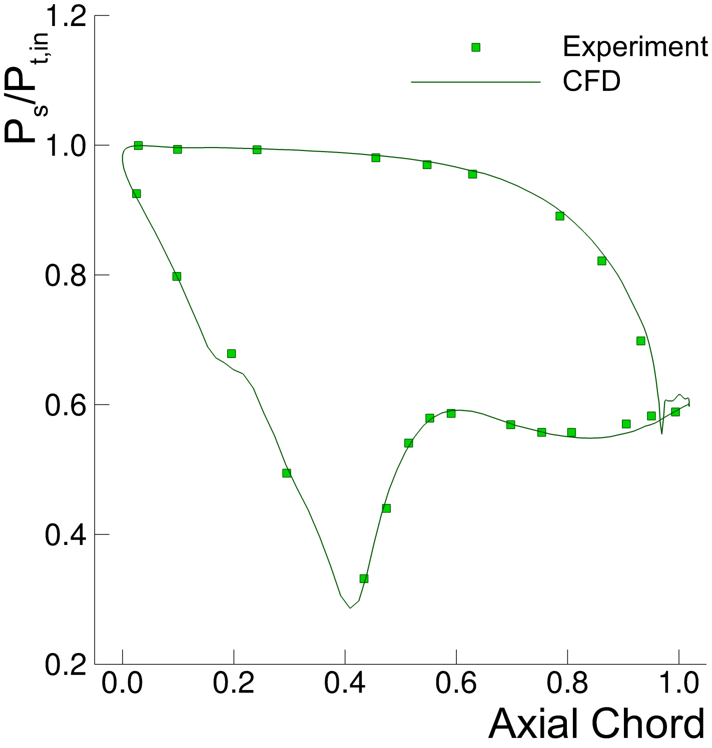

Figure 6.11: Surface pressure distribution over the Mark II turbine vane.

## 6.2 Configuration D

In this section, the proposed optimization framework is applied to the two-dimensional 2.5-stage Configuration D compressor considering a fully-turbulent flow environment. After introducing the test case in subsection 6.2.1, a grid study is performed to determine the appropriate grid dimensions and evaluate the flow solution, cf. subsection 6.2.2. Then, in subsections 6.2.3 and 6.2.4, the adjoint solution is discussed and the adjoint sensitivities are validated. Finally, two optimization cases, an unconstrained redesign as well as a constrained optimization of the multistage compressor, are presented and compared with each other in subsections 6.2.5 and 6.2.6.

### 6.2.1 Test Case

The first test case is a 2.5-stage transonic axial flow compressor known as Configuration D. This two-dimensional model compressor, designed and used by Hall and co-workers to investigate unsteady flows in multistage turbomachines, cf. [87] and [88], consists of three rotor and two stator blades (R1-S1-R2-S2-R3), each made of NACA four digit series airfoils. The geometric and aerodynamic design parameters of the Configuration D compressor are provided in Table 6.3. All lengths are non-dimensionalized by the chord length of rotor R2. The blade rows are closely spaced, as it is typical in modern compressors, cf. axial gab in Table 6.3. The 2.5-stage model compressor is operated at a total pressure ratio of $\pi = 3.0$, the translational rotor speed in the $x_2$-direction is defined as $\Omega_{x_2} = 1.18$, and the respective radial length scale is $r = 1.0$. In this research work, a fully-turbulent viscous flow environment is investigated, while Hall and co-workers examined an inviscid flow field. Yet, the design goal is still to match the overall total pressure ratio. Accordingly, some other aerodynamic design parameters, e.g. the pressure ratio $p_{s,\mathrm{out}}/p_{t,\mathrm{in}}$, have to be slightly adjusted, which in turn leads to slightly different flow quantities at the five flow stations. Figure 6.12 depicts the block-structured grid of the multistage compressor and shows the different flow stations at which the aerodynamic parameters are evaluated. Station 0 is located at the inlet plane, station 5 represents the outlet, and stations 2, 3, and 4 are located at the center of the axial gab, where the adjacent blade rows are coupled by a mixing-plane interface. O-grids are used to discretize the domain around the blade surface. H-topology grids are utilized to model the remaining passage. The final grid dimensions are determined by an elaborate grid study; its results are discussed in the following subsection 6.2.2.

### 6.2.2 Flow Solution

The grid dimensions of the structured multiblock grid for the Configuration D model compressor, depicted in Figure 6.12, are determined by a grid study, in which the numerical solutions obtained on grids with different dimensions are compared. Starting from the coarsest grid, the grid dimensions are successively doubled until a grid-independent flow solution is established.

| Row or flow station | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Number of blades | — | 26 | 32 | 40 | 50 | 62 |
| NACA 4-digit airfoil | — | (3.5)506 | (4.5)506 | (4.5)506 | (4.5)506 | (4.5)506 |
| Chord | — | 1.539 | 1.25 | 1.0 | 0.8 | 0.645 |
| Blade-to-blade gap | — | 1.231 | 1.0 | 0.8 | 0.64 | 0.516 |
| Axial gap | — | 0.31 | 0.25 | 0.20 | 0.16 | — |
| Stagger angle | — | -44.0 | 43.0 | -49.5 | 52.0 | -55.0 |
| Abs. flow angle | 28.6 | 55.6 | 38.2 | 63.1 | 48.4 | 68.4 |
| Rel. flow angle | -53.3 | -41.4 | -60.4 | -46.4 | -66.1 | -55.0 |
| Abs. Mach number | 0.626 | 0.747 | 0.483 | 0.679 | 0.404 | 0.619 |
| Rel. Mach number | 0.919 | 0.563 | 0.769 | 0.446 | 0.660 | 0.379 |
| Static pressure | 0.768 | 1.049 | 1.263 | 1.608 | 1.935 | 2.320 |
| Abs. total pressure | 1.000 | 1.519 | 1.482 | 2.191 | 2.165 | 3.005 |
| Rel. total pressure | 1.325 | 1.302 | 1.866 | 1.847 | 2.592 | 2.562 |

Table 6.3: Geometric and aerodynamic design parameters of the 2.5-stage Configuration D compressor.

CFD flow solvers often are not able to cope with multistage turbomachinery flow problems, if the flow field in the computational domain, which consists of several stationary and fast-translating blade rows, is not initialized adequately. To avoid numerical stability problems, in this work the following initialization strategy is applied: First, a meaningful starting solution is established by solving the inviscid Euler equations and progressively increasing the back pressure and translational velocity until the desired exit pressure and the required rotor entrainment velocity are obtained. Only then the viscous terms are included. Furthermore, for particularly challenging flow problems it is advantageous to first calculate a meaningful flow field by employing a first-order artificial dissipation scheme, before switching to a more accurate but often less stable second-order accurate numerical scheme.

Starting from a first-order viscous flow solution, the numerical code reduces the maximum density residual by eleven to twelve orders and a converged second-order turbulent flow solution for the Configuration D test case is obtained within less than 2500 multigrid cycles on each grid investigated, cf. Figure 6.13 (left). Figure 6.13

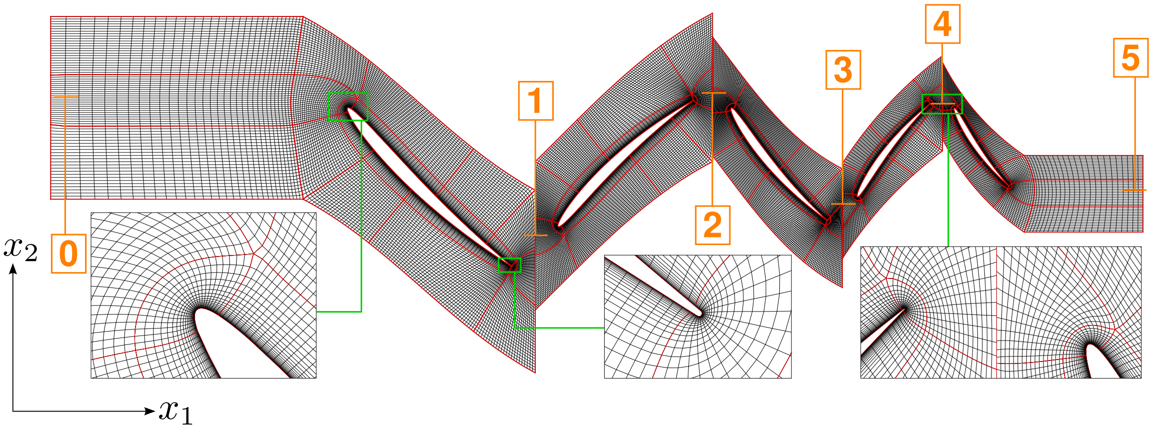

Figure 6.12: Computational grid and flow stations for the 2.5-stage Configuration D compressor.

(right) illustrates that the SST turbulence model converges at a similar rate. As depicted in the figure, the maximum turbulent kinetic energy $k$ residual also drops by elven to twelve orders within the same number of multigrid cycles. The specific dissipation rate $\omega$ possesses similar convergence properties. Figures 6.14 and 6.15 (left) show the calculated flow field near the Rotor 1 blade, while Figure 6.15 (right) illustrates the corresponding non-dimensionalized surface pressure distribution of Rotor 1

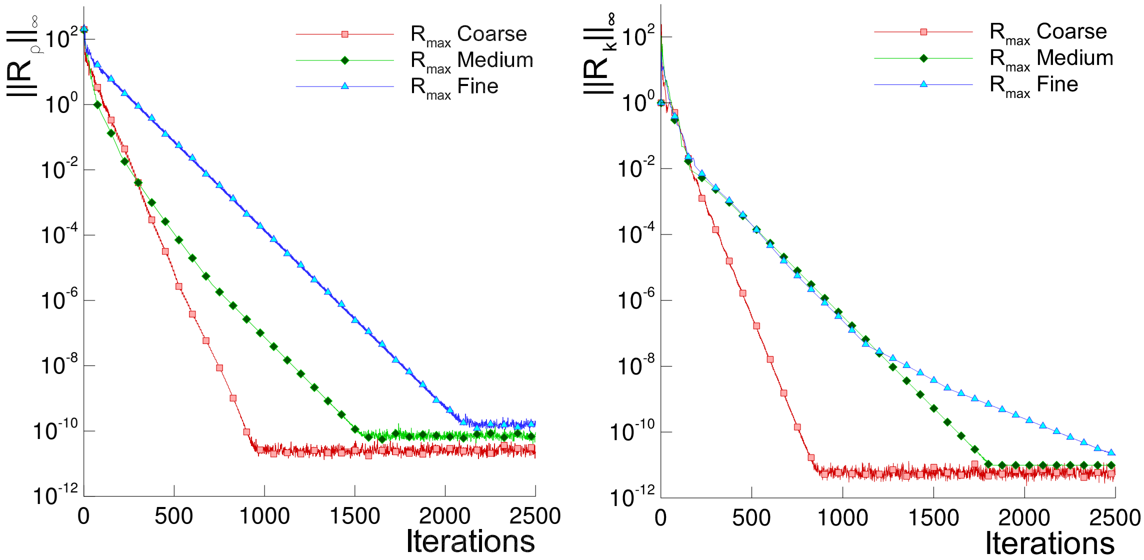

Figure 6.13: Convergence of maximum density residual (left) and maximum turbulent kinetic energy residual (right) after initialization with first-order solution.

for the various grid resolutions investigated. All numerical results predict a shock at approximately 40% of the suction side chord of Rotor 1; however, on the coarsest grid the shock is slightly smeared. Furthermore, compared to the medium and fine grid distributions, the static pressure provided from the coarse grid solution is slightly lower on both the suction and pressure side. On the other hand, the numerical solution obtained on the medium grid is almost identical to the fine grid solution; not only for the Rotor 1 domain but also for the downstream blade rows S1-R2-S2-R3 where the flow remains subsonic and the numerical solution is less critical to the grid resolution. A comparison of the main aerodynamic design parameters, cf. Table 6.4, confirms that the medium grid resolution is sufficiently fine. The differences in the total pressure and temperature ratio as well as the change in the mass flow rate (normalized with the mass flow rate of the fine grid solution) between the medium and the finest grid are negligible. Therefore, the medium grid is selected and used for the optimization cases presented in the following subsections. The final grid dimensions are summarized in Table 6.5. To resolve the boundary layer accurately the wall-spacing is chosen such that $y^+_{\mathrm{avg}} = 1.4$.

| Grid | Coarse | Medium | Fine |
|---|---|---|---|
| $\pi$ | 3.014 | 3.010 | 3.011 |
| $T_{t,\mathrm{out}}/T_{t,\mathrm{in}}$ | 1.389 | 1.388 | 1.388 |
| $\dot{m}/\dot{m}_{\mathrm{fine}}$ | 1.002 | 0.999 | 1.000 |

Table 6.4: Design parameters for different grids, Configuration D.

| Row | R1 | S1 | R2 | S2 | S3 |
|---|---|---|---|---|---|
| Grid points | 15325 | 12850 | 12850 | 7634 | 8237 |
| Surface grid points | 225 | 225 | 225 | 161 | 145 |
| Blade-to-blade passage | 97 | 97 | 97 | 81 | 81 |

Table 6.5: Grid dimensions of the selected medium grid, Configuration D.

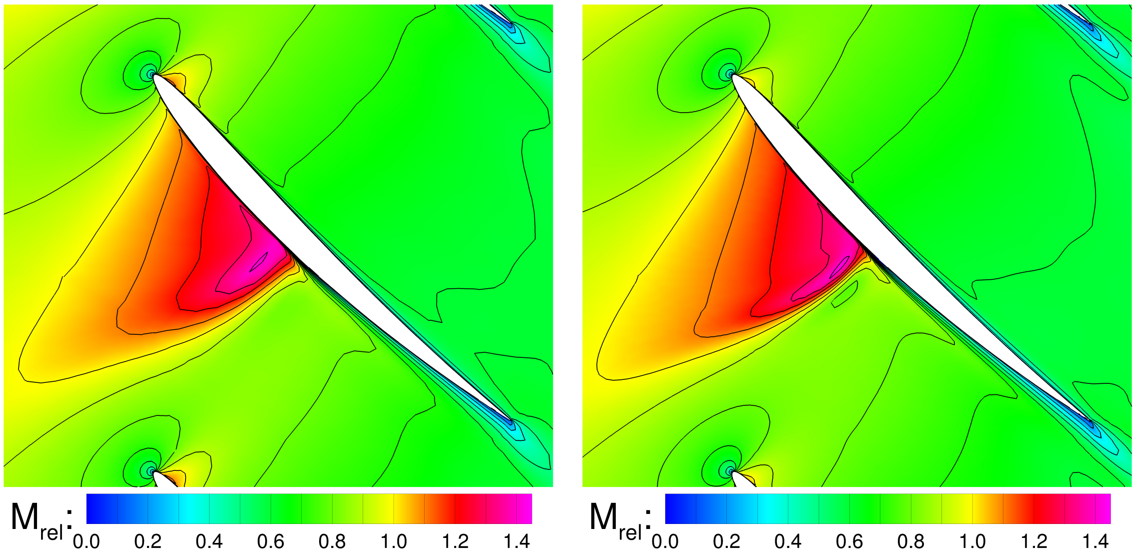

Figure 6.14: Relative Mach number contours for the coarse grid (left) and medium grid (right), Configuration D, Rotor 1.

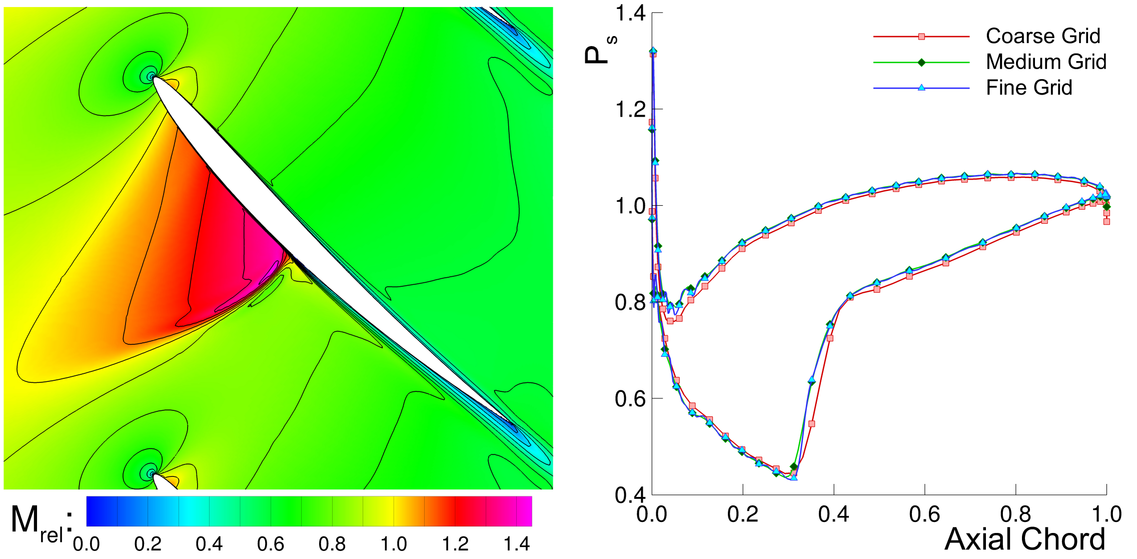

Figure 6.15: Relative Mach number contours for the fine grid (left) and surface pressure distribution for different grid dimensions (right), Configuration D, Rotor 1.

### 6.2.3 Adjoint Solution

After having selected the grid dimensions such that it is guaranteed that the converged flow solution is sufficiently accurate and grid independent, the adjoint equations can be solved based on the information provided by the calculated flow field.

The final solution of the adjoint system naturally depends on its own convergence level, but is also a function of the flow solution. Hence, the accuracy of the adjoint solution can be sensitive to the convergence level of the flow governing equations. Both, an inadequately converged flow or adjoint solution may impact the gradient accuracy in a negative manner and deteriorate the quality of the optimization. On the other hand, converging both the flow and adjoint systems to a level far beyond a necessary level, considerably increases the computational cost of the optimization and, thus, should be avoided. As a consequence, an initial convergence study is performed in which the convergence levels of both the flow and adjoint equations required for an accurate gradient calculation but efficient optimization are determined.

In the following, the first design cycle of the optimization cases presented in subsections 6.2.5 and 6.2.6 is investigated. The adjoint equations are solved for the entropy functional, which serves as the design objective in both optimization cases. Four test cases are considered. Table 6.6 summarized the number of multigrid cycles performed to solve the flow and adjoint equations and provides the final convergence level for each calculation. Figure 6.16 (left) illustrates the convergence history of the maximum density residual for the Configuration D compressor; the convergence of the adjoint equations is shown in Figure 6.16 (right). It can be seen that a fully-converged flow solution is obtained after approximately 1000 multigrid cycles. Although the underlying grid is identical to the medium-sized grid selected in subsection 6.2.2, the convergence now is slightly faster, which is due to an improved flow initialization. Furthermore, it can be observed that considerably more multigrid cycles are necessary to converge the adjoint system to a similar level as the flow equations. The reasons for this are twofold. First, the adjoint solution is calculated without an appropriate

| Case | Flow Cycles | Adjoint Cycles | $\|\|\mathrm{R}_\rho\|\|_\infty$ | $\|\|\mathrm{R}_{\psi_1}\|\|_\infty$ |
|---|---|---|---|---|
| C1 | 1200 | 12000 | $5.24\mathrm{e}^{-11}$ | $1.10\mathrm{e}^{-10}$ |
| C2 | 600 | 6000 | $5.77\mathrm{e}^{-8}$ | $1.27\mathrm{e}^{-6}$ |
| C3 | 300 | 3000 | $1.63\mathrm{e}^{-5}$ | $1.54\mathrm{e}^{-2}$ |
| C4 | 150 | 1500 | $1.22\mathrm{e}^{-3}$ | $2.60\mathrm{e}^{+0}$ |

Table 6.6: Test cases investigated in the convergence study, Configuration D.

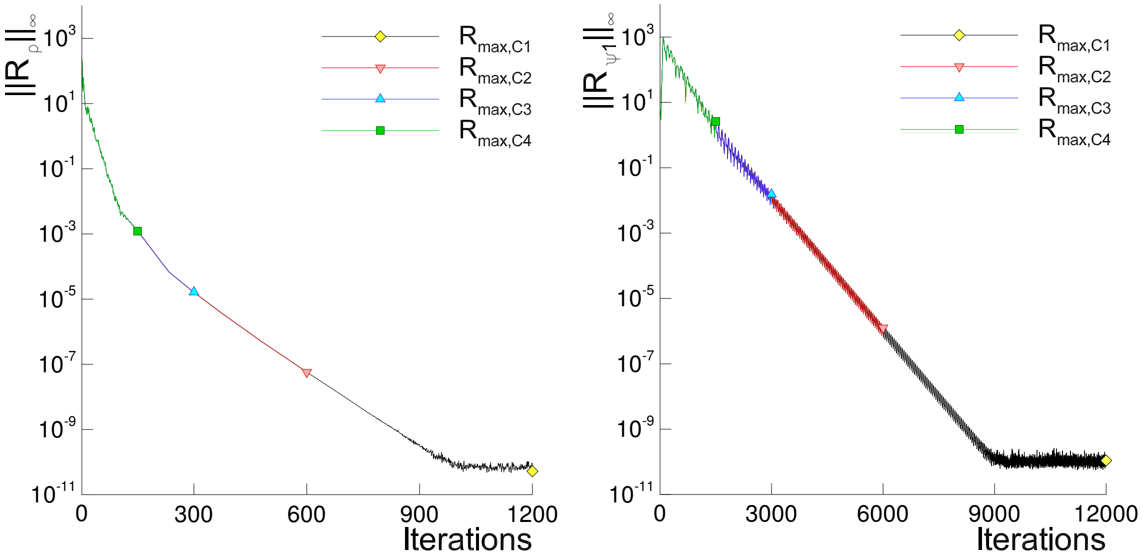

Figure 6.16: Convergence level of maximum density residual (left) and maximum co-state $\psi_1$-residual (right) for various multigrid cycles, Configuration D.

initialization. Second, for complex viscous flow problems, the adjoint system tends to become stiff and the CFL number as well as the number of multigrid levels employed have to be reduced, compared to the values used within the flow solver, to guarantee numerical stability.

Depending on the test case, the adjoint solutions are calculated based on flow solutions which were converged to different levels. Despite this, the convergence curves of the various adjoint calculations match almost perfectly, which suggests that converging the flow equations further than in the test cases C4 and C3 does not further improve the accuracy of the gradient calculation. At the same time, the number of iterations used to solve the adjoint equations differs, cf. Table 6.6. Figure 6.17, which shows the difference in the co-state solution $\psi_4$ between the fully-converged test case C1 and the partially-converged test cases C4 (left) and C3 (right), indicates that converging the adjoint equations beyond the levels of test case C3 does not alter the adjoint solution significantly. Table 6.7 confirms these findings and depicts the maximum error in the adjoint solution $\Delta\boldsymbol{\psi}$ of the cases C2, C3, and C4 compared to the fully-converged test case C1. Note, the differences originate from converging both the flow and adjoint equations only partially. For comparison purposes, the minimum and maximum values of the adjoint solution of case C1 are

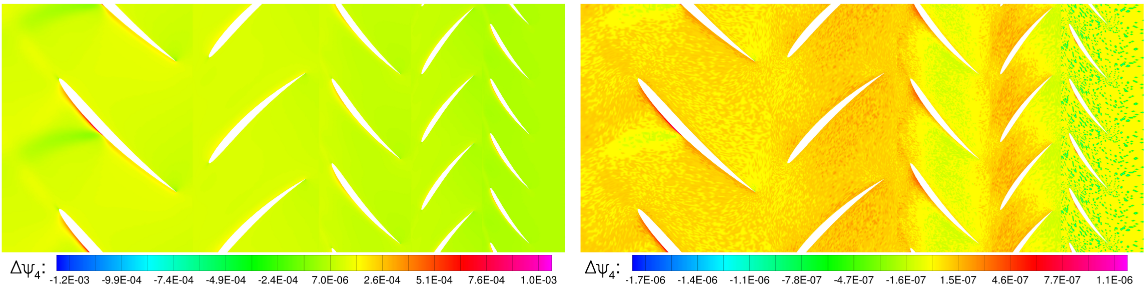

Figure 6.17: Error in adjoint solution $\psi_4$ for test case C4 (left) and test case C3 (right), $\Delta\psi_4 = \psi_{4,\mathrm{C1}} - \psi_{4,\mathrm{C4/C3}}$, Configuration D.

provided as well. To ultimately determine the impact on the gradient accuracy, the adjoint sensitivities are calculated for each test case. Table 6.8 summarizes these results and shows the maximum error in the gradient for all partially-converged test cases. For the cases C2 and C3, the gradient error is at least four to five orders smaller than the gradient value itself. Hence, the convergence study demonstrates that for these test cases, the inaccuracies introduced by converging both the flow and adjoint equations only partially, are beyond significance for the calculation of accurate adjoint sensitivities.

Consequently, the results suggest that the convergence levels obtained in the test case C3 lead to a sufficiently accurate flow and adjoint solution. However, during the optimization the blade geometries change and the rate of convergence varies. Thus, to ensure that the equations are converged adequately at every design cycle, the multigrid cycle set-up employed in test case C2 is applied in the design cases presented in subsections 6.2.5 and 6.2.6. This means, the robustness of the optimization is

| $i$ | $\psi_{i,\mathrm{C1}}$ | $\lvert\Delta\psi_{i,\mathrm{C4}}\rvert$ | $\lvert\Delta\psi_{i,\mathrm{C3}}\rvert$ | $\lvert\Delta\psi_{i,\mathrm{C2}}\rvert$ |
|---|---|---|---|---|
| 1 | 0.34 … 3.22 | $3.02\mathrm{e}^{-2}$ | $3.91\mathrm{e}^{-5}$ | $1.03\mathrm{e}^{-7}$ |
| 2 | -1.58 … 1.44 | $1.14\mathrm{e}^{-2}$ | $1.56\mathrm{e}^{-5}$ | $1.04\mathrm{e}^{-7}$ |
| 3 | -0.87 … 3.59 | $2.77\mathrm{e}^{-2}$ | $3.58\mathrm{e}^{-5}$ | $9.54\mathrm{e}^{-7}$ |
| 4 | -0.80 … -0.10 | $1.37\mathrm{e}^{-3}$ | $1.85\mathrm{e}^{-6}$ | $1.19\mathrm{e}^{-7}$ |

Table 6.7: Minimum and maximum value of adjoint solution for test case C1 and error of test cases C2, C3, and C4, Configuration D.

| Row | $G_{C1}$ | $\lvert\Delta G_{C4}\rvert$ | $\lvert\Delta G_{C3}\rvert$ | $\lvert\Delta G_{C2}\rvert$ |
|---|---|---|---|---|
| R1 | $-7.87e^{-2}$ ... $4.95e^{-2}$ | $1.04e^{-3}$ | $1.40e^{-6}$ | $1.00e^{-11}$ |
| S1 | $-2.17e^{-2}$ ... $1.57e^{-2}$ | $6.56e^{-4}$ | $8.40e^{-7}$ | $1.00e^{-11}$ |
| R2 | $-3.49e^{-2}$ ... $1.69e^{-2}$ | $3.84e^{-4}$ | $2.10e^{-6}$ | $1.00e^{-10}$ |
| S2 | $-3.55e^{-2}$ ... $2.78e^{-2}$ | $4.84e^{-4}$ | $3.50e^{-7}$ | $1.00e^{-10}$ |
| R3 | $-4.55e^{-2}$ ... $3.17e^{-2}$ | $2.79e^{-4}$ | $5.70e^{-7}$ | $1.00e^{-10}$ |

Table 6.8: Minimum and maximum gradient for test case C1 and error of test cases C2, C3, and C4, $\Delta G = G_{C1} - G_{C4/C3/C2}$, Configuration D.

increased at the expense of a higher computational cost.

The convergence study performed in this subsection determines the required number of flow and adjoint iterations to ensure that the error in the adjoint solution, compared to the fully-converged calculations, is negligible. However the study does not verify the correctness of the adjoint solution itself. Its actual correctness is verified in the gradient study presented in the following subsection 6.2.4.

### 6.2.4 Gradient Validation

Before presenting the redesign of the Configuration D model compressor, the adjoint solver as well as the implemented gradient calculation routines are validated. This is done by comparing the sensitivities obtained from the dual adjoint solver with finite-difference gradients calculated from the primal flow solver. The finite-difference gradients are obtained from a simple first-order forward difference approximation, i.e. the function of interest is evaluated both for the unperturbed design and based on the solution for the perturbed geometry. Next, the difference between the two function values is calculated and divided by the perturbation step, which then provides the corresponding finite-difference approximation of the design variable sensitivity. To ensure that the perturbation step does not affect the accuracy of the finite-difference gradient, the appropriate step size is determined from a separate gradient study.

The main motivation of this gradient validation is to validate the turbomachinery specific features of the adjoint solver, i.e. to verify the correctness of the implemented adjoint wall, inlet and outlet boundary conditions, as well as to ensure the flow-consistency of the derived adjoint mixing-plane formulation and the linearization of the various functionals. Since the characteristic-based inlet and outlet boundary conditions and the flow treatment at a mixing-plane interface are derived from the linearized Euler equations, the corresponding adjoint transformation matrices $\boldsymbol{\mathcal{T}}_{\mathrm{BD}}$ are identical for both inviscid and viscous flows. Similarly, the linearized objective function does not depend on the flow physics. Only, the wall boundary condition changes, thus, the adjoint transformation matrix at a wall depends on the flow governing equations. However, as illustrated in subsection 4.3.1, the linearization of the wall boundary condition is straightforward for both inviscid and viscous flows. Hence, in this section the adjoint boundary conditions and the developed adjoint mixing-plane formulation are validated based on the inviscid Euler equations.

**Test Case I: R1**

The isolated Rotor 1 of the Configuration D model compressor serves as the first two-dimensional test case. The inviscid grid is of the same multi-block topology as the viscous grid shown in Figure 6.12; however, the computational domain is extended at the second flow station to push the outlet further downstream. Figure 6.18 (left) depicts the flow solution for this test case. The back pressure is chosen such that the rotor is operated within a transonic flow regime. A supersonic flow region develops in the front part of the rotor suction side, which is terminated by a strong shock before the rotor passage entry. Figure 6.18 (right) shows the $\psi_4$-contours of the adjoint solution, i.e. the co-state variable corresponding to the energy equation, for the entropy functional defined in Eqn. (4.163). For the gradient evaluation, 78 Hicks-Henne bump functions are equally distributed along the blade; 39 design variables on the suction side and 39 bumps on the pressure side. The calculated adjoint sensitivities are compared with the finite-difference gradients in Figure 6.19. It can be observed that the adjoint sensitivities agree very well with the gradient obtained from the forward approach. Note, the discontinuity at the 39$^{\mathrm{th}}$ design variable, which is located at the leading edge (LE), origins from the fact that the Hicke-Henne bump function is not defined at the leading edge point; cf. Eqn. (5.5) where $t_{l,1}$ and $t_{l,2}$ cannot take on the values 0 (trailing edge) and 1 (leading edge).

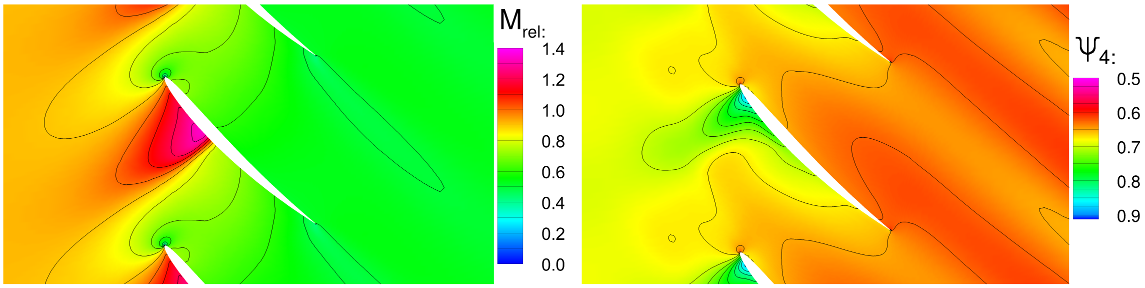

Figure 6.18: Relative Mach number contours (left) and $\psi_4$-contours (right), Configuration D, test case I: R1, objective function: entropy generation rate.

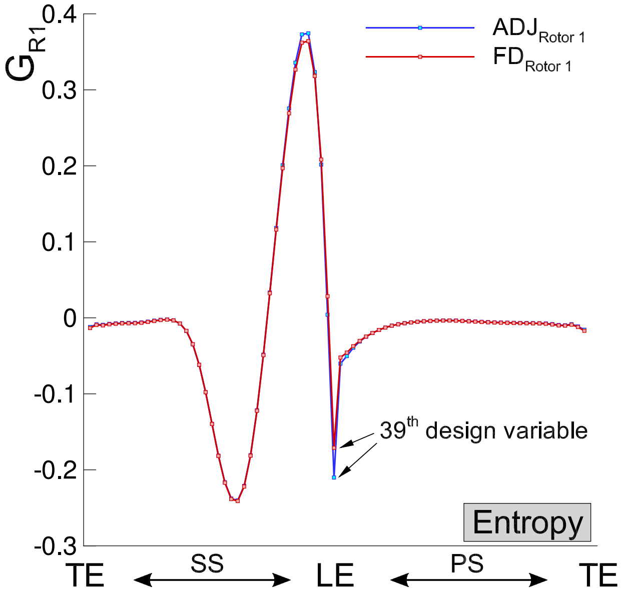

Figure 6.19: Gradient comparison, Configuration D, test case I: R1, objective function: entropy generation rate.

### Test Case II: R1-S1-R2

A second test case, which consists of the first three rows of the Configuration D model compressor, is used to verify the accuracy of the adjoint mixing-plane boundary condition within a two-dimensional flow environment. Two rotor-stator interfaces appear within this test case; the first one couples Rotor 1 with the downstream Stator 1 and the second mixing-plane exchanges the flow variables between the Stator 1 and Rotor 2 domains. Figure 6.20 illustrates the $\psi_1$-contours of the adjoint solution, i.e. the co-state variable corresponding to the continuity equation, calculated for the total pressure ratio functional, cf. Eqns. (4.167) and (4.168). The plot clearly depicts that the path of information propagation in the adjoint system is opposite to that of the flow governing equations. A wake develops at the Rotor 2 leading edge and propagates into the physical upstream direction. The wake is mixed-out at the upstream adjoint mixing-plane and a jump occurs in the local adjoint values across the domain interface. The adjoint gradients of the three blades are compared with the corresponding finite-difference gradients in Figures 6.21 to 6.23. Again, 78 Hicks-Henne functions are used as design variables and the bumps are equidistantly distributed along each blade surface. The plots show that the adjoint sensitivities of all blades are in good agreement with the finite-difference gradients. Since the objective function is active at the exit plane only, this test case particularly verifies that the derived adjoint mixing-plane formulation passes the objective function information correctly through the averaging-plane interfaces.

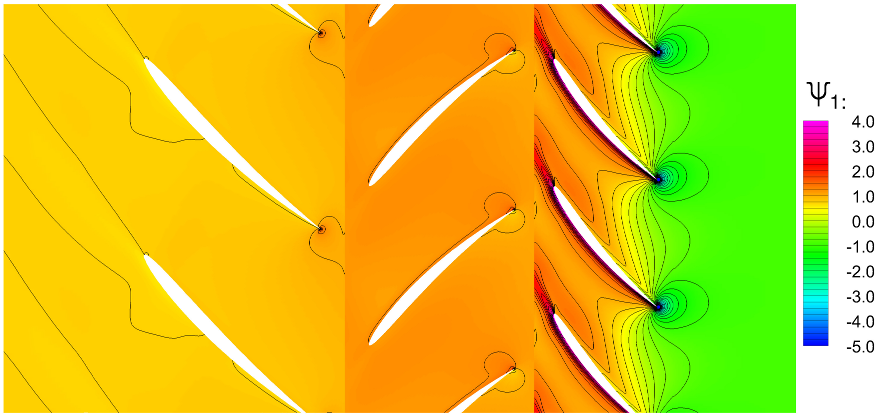

Figure 6.20: $\psi_1$-contours, Configuration D, test case II: R1-S1-R2, objective function: total pressure ratio.

Figure 6.22 suggests that there is a larger discrepancy between the adjoint and finite-difference gradients in the vicinity of the stator leading edge. However, the amplitudes of the stator sensitivities are much smaller than those of the rotor sensitivities and the absolute differences are of the same order as they are for the rotor gradients. Furthermore, the discrepancy is restricted to the near leading edge region, where the smoothness of the Hicks-Henne bumps is compromised by the function's leading edge discontinuity and, thus, the accuracy of the finite-difference gradients is questionable in itself. As can be seen in subsection 6.3.4, the adjoint and finite-difference gradients generally conform well in the blade leading edge region.

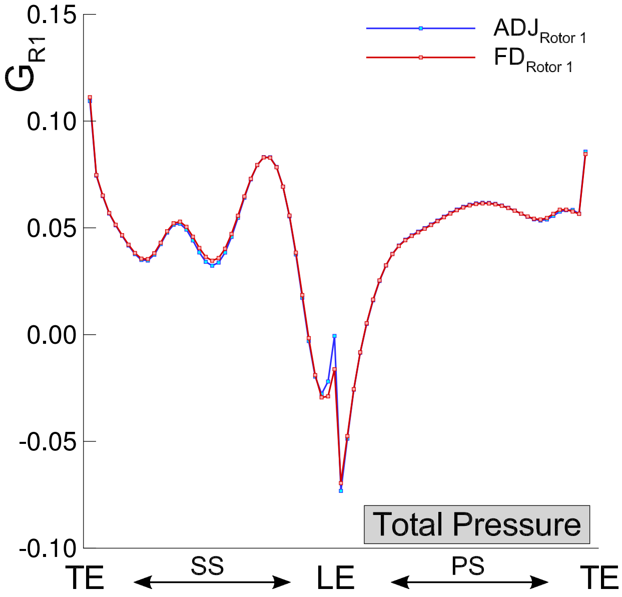

Figure 6.21: Gradient comparison Rotor 1, Configuration D, test case II: R1-S1-R2, objective function: total pressure ratio.

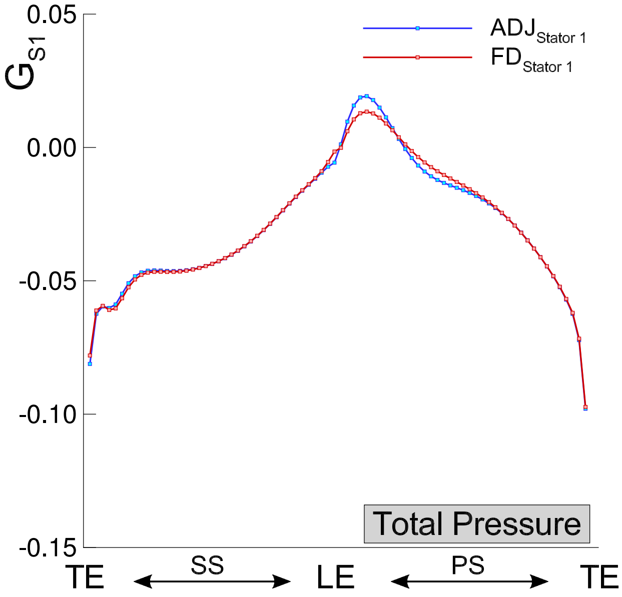

Figure 6.22: Gradient comparison Stator 1, Configuration D, test case II: R1-S1-R2, objective function: total pressure ratio.

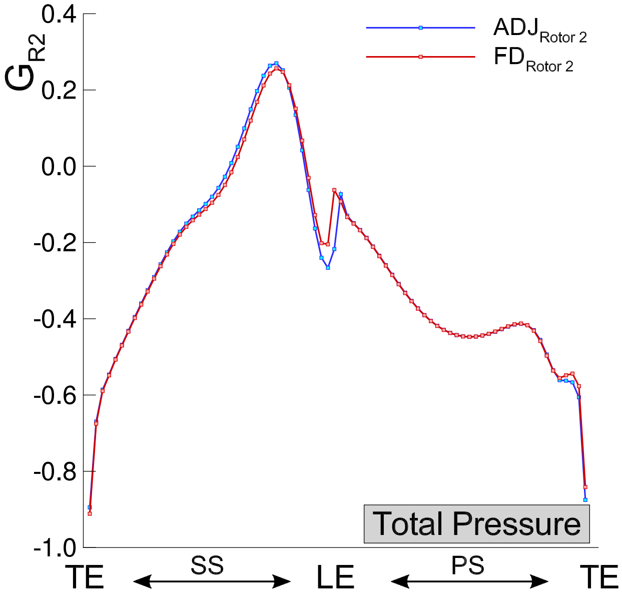

Figure 6.23: Gradient comparison Rotor 2, Configuration D, test case II: R1-S1-R2, objective function: total pressure ratio.

### 6.2.5 Unconstrained Optimization

First, an unconstrained optimization case is investigated. The objective is to reduce the entropy generation rate, which is equivalent to increasing the isentropic efficiency. After solving the flow equations and the adjoint system as described in chapters 3 and 4, the objective function gradient is efficiently calculated using the approach presented in section 5.1. The gradient is then provided to the optimizer, which determines the blade shape modifications through the SQP algorithm outlined in section 5.2. For all two-dimensional design cases the surface grid points are used as design variables, which leads to highly non-smooth gradients. Although generally it is not required, the calculated gradient is slightly smoothed by a second-order implicit smoothing technique before providing it to SNOPT. The smoothing acts as a preconditioner and allows for the calculation of much larger steps, and thus accelerates the convergence. However, too large a smoothing parameter degrades the gradient information and can result in the calculation of a suboptimal search direction, which then negatively affects the optimization. Accordingly, the amount of smoothing is kept as low as possible to keep the gradient information as accurate as possible. As a consequence, the shape modifications calculated by SNOPT are non-smooth as well and have to be smoothed to obtain an optimized blade geometry without kinks. The same implicit smoothing technique as for the gradient smoothing is employed; however, this time a large smoothing coefficient is used. Furthermore, geometrical constraints are assigned to the rotor and stator leading and trailing edge grid points to preserve the original blade radii. No maximum thickness constraint is prescribed to the blade profiles. At each design cycle, the computational domain is first initialized with the flow solution at the previous design cycle and, then, 600 multigrid cycles are performed to evaluate the new flow field. Depending on the blade shapes, the density residual drops by six to eight orders, which has been shown to be sufficient for optimization, cf. subsection 6.2.3. The adjoint system is converged to a similar level.

Figure 6.24 depicts the flow solution obtained for the Configuration D baseline design. The compressor operates at a total pressure ratio of $\pi = 3.01$ and has a baseline efficiency of $\eta_{\mathrm{is}} = 95.31\%$. The relative Mach number contour plot illustrates that the flow accelerates to supersonic speeds in the front region of the Rotor 1 suction side, before a shock reduces the flow velocity to subsonic regimes. According to the numerical solution, the flow separates after the steady shock front,

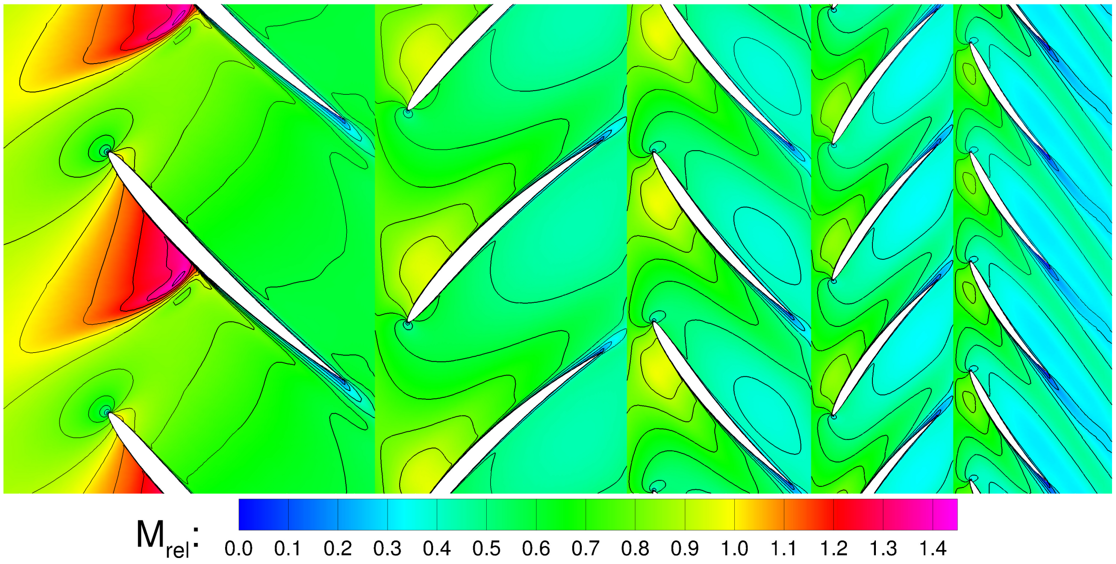

Figure 6.24: Relative Mach number contours, Configuration D, baseline design, $\eta_{\mathrm{is}} = 95.31\%$.

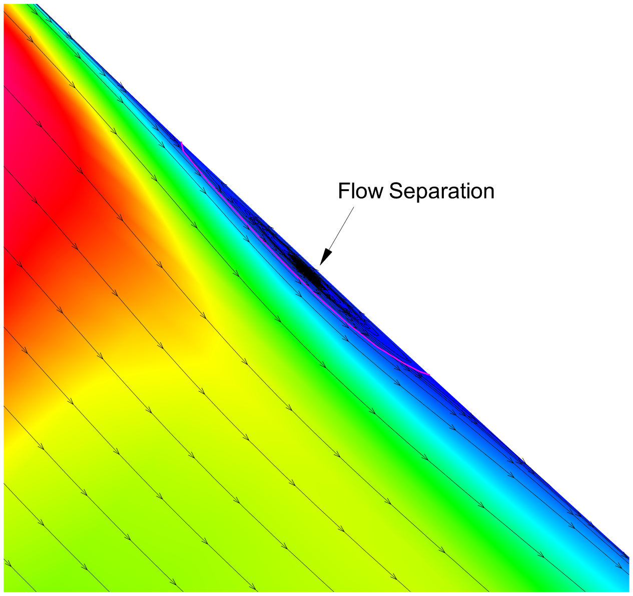

Figure 6.25: Boundary-layer flow after Rotor 1 shock: flow separation for baseline design, Configuration D.

cf. Figure 6.25. Yet, despite the strong shock, the expansion of the separation bubble is restricted to the vicinity of the rotor suction side surface and the flow re-attaches almost immediately. It should be stressed, that no experimental results are available and, thus, the existence of the separation bubble predicted by the turbulence model, cannot be verified. In the downstream stator and rotor rows the flow does not exceed $M_{\text{rel}} = 1.0$ and remains attached to the blade profiles. At the four rotor-stator interfaces, the local flow solution is discontinuous; note the mismatching contour lines at the averaging plane interfaces in Figure 6.24. Nonetheless, the circumferentially averaged fluxes are conserved across each mixing-plane interface.

Figure 6.26 displays the corresponding adjoint solution for the Configuration D baseline design, which shows the co-state solution $\psi_1$ corresponding to the continuity equation for the entropy functional. The plot confirms a phenomenon already observed in Figure 6.20; information within the adjoint system propagates in the direction opposite to the path it travels in the flow governing equations. At every blade leading edge a wake develops and propagates into the physically upstream direction until it is mixed-out at a rotor-stator interface, where the adjoint solution is discontinuous due to the nature of the mixing-plane boundary.

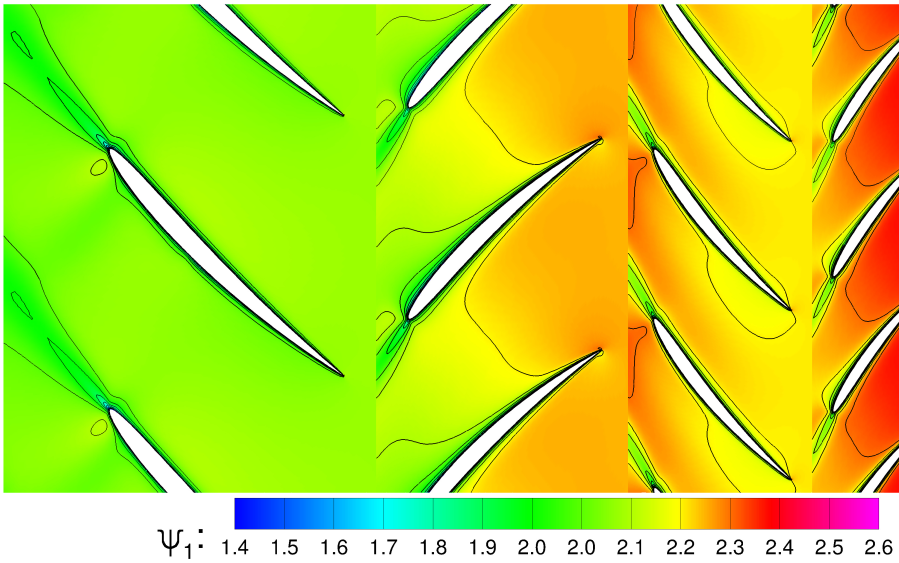

Figure 6.26: $\psi_1$-contours, Configuration D, baseline design, objective function: entropy generation rate.

Figure 6.27 depicts the gradient distribution for the Rotor 1 and Stator 1 blades at the first design iteration. The design variables within the red-dashed boxes represent the leading edge (LE) and trailing edge (TE) points which are geometrically constrained and remain unchanged throughout the optimization. Clearly, the gradient amplitudes of the Rotor 1 design points are larger than those of the Stator 1 design parameters, which indicates that the rotor geometry has more influence on the design objective than the stator. This becomes evident when considering the reasons for the entropy increase. While in the stator domain the entropy generation is rather small and due to viscous effects only, the main source for the entropy increase throughout the first compressor stage is the strong rotor shock and the accompanying shock/boundary-layer interaction. Consequently, the highest gradient values occur in the near-shock and leading edge regions. The gradient amplitudes of the downstream rotor and stator blades are of the same magnitude as the Stator 1 amplitudes since in these blade rows the flow remains subsonic as well, cf. Figure 6.28.

The final optimized compressor geometry, obtained after eleven design cycles, is compared with the baseline design in Figure 6.29. The Rotor 1 geometry experiences the largest profile changes, where the suction side is modified noticeably. While the suction side curvature is reduced in the supersonic flow region, it is increased in the

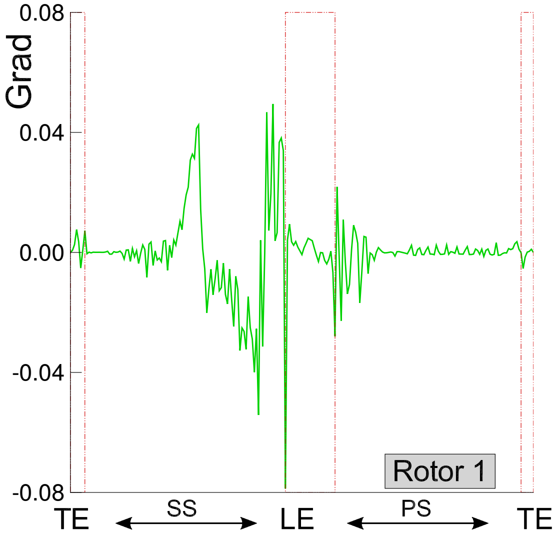

Figure 6.27: Gradient for Rotor 1 (left) and Stator 1 (right), Configuration D, baseline design, objective function: entropy generation rate.

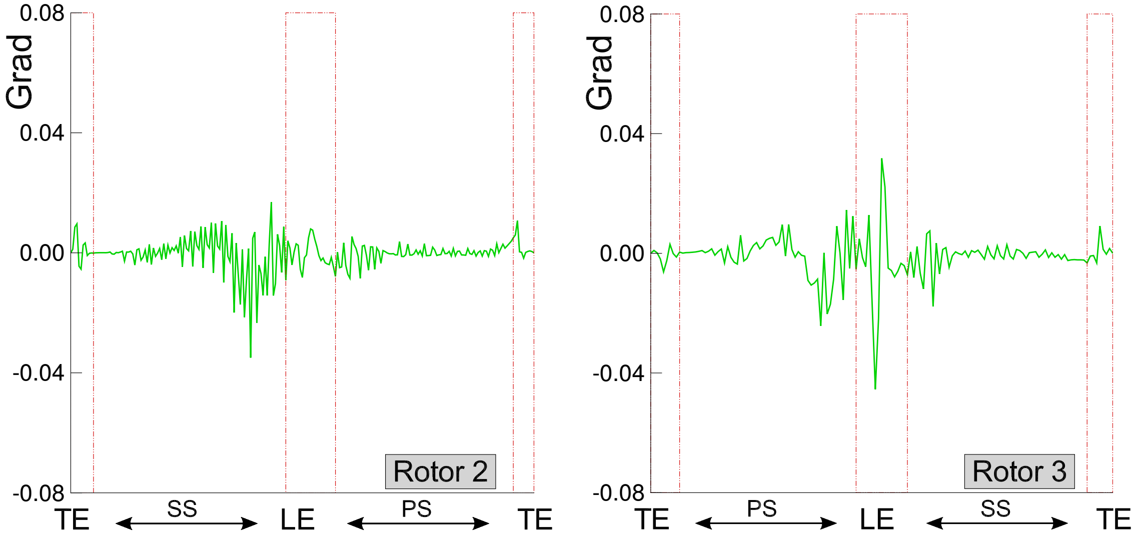

Figure 6.28: Gradient for Rotor 2 (left) and Rotor 3 (right), Configuration D, baseline design, objective function: entropy generation rate.

subsonic flow region after the shock. Rotor 2 and Rotor 3 undergo similar, but less distinct shape modifications. The stator geometries experience only slight modifications, with primarily a reduction in the curvature on the forward suction side. In the redesigned Rotor 1, the flow acceleration in the supersonic region is consequently

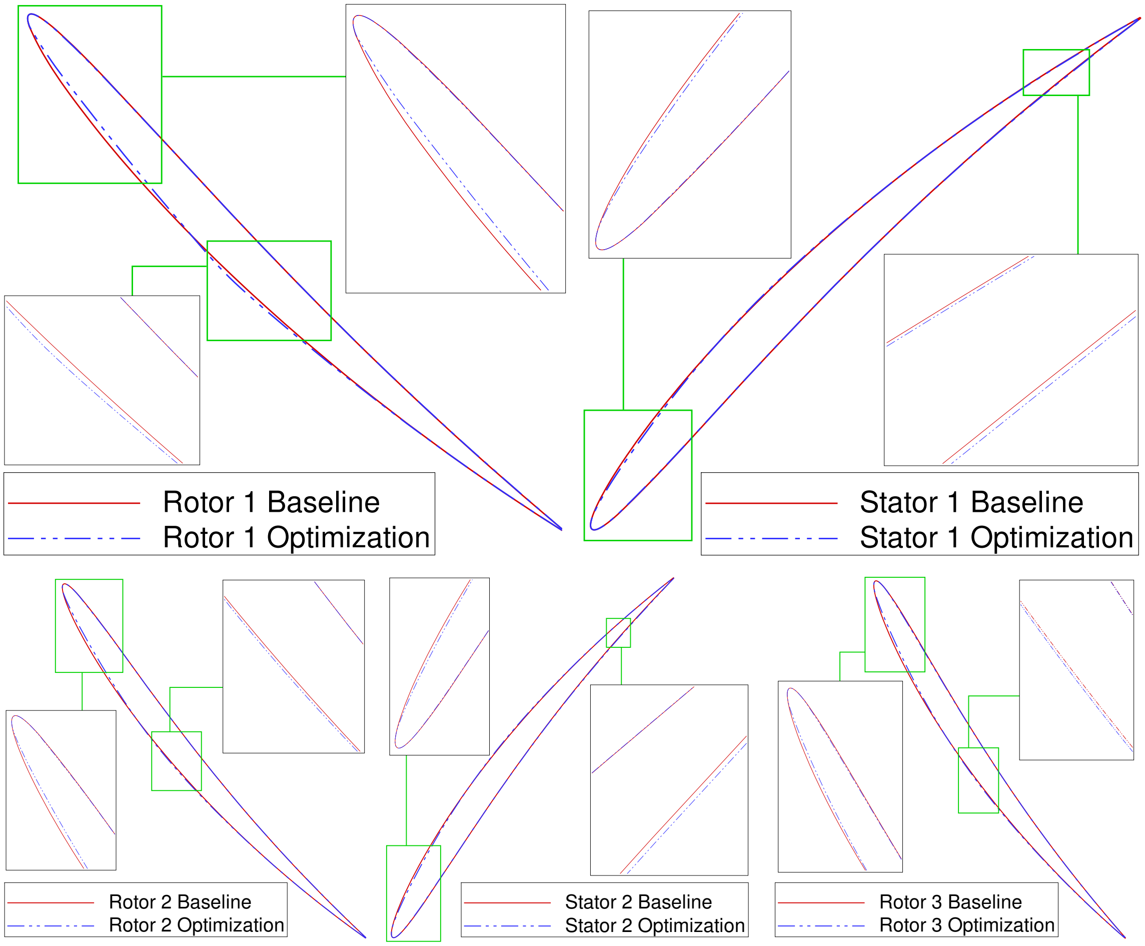

Figure 6.29: Shape modification Configuration D, unconstrained optimization.

slowed down leading to a weaker shock, cf. Figure 6.30. The weaker shock/boundary-layer interaction results in a thinner boundary-layer and the flow remains attached in the optimized design, cf. Figure 6.32. Comparing the entropy field of the baseline design with that of the optimized design, cf. Figure 6.31, where $\delta s = s_0 - s_{\text{opt}}$ and hence $\delta s > 0$ indicates a lower entropy level in the optimized design, shows that the weaker shock in turn leads to a reduced entropy generation after the shock and in the downstream boundary-layer of Rotor 1. The slight entropy increase in the optimized design (blue coloured region) compared to the baseline design ($\delta s < 0$) just upstream of the shock is due to a change in the shock location. As depicted in Figure 6.33 (left), the overall entropy generation rate drops during the optimization by 23.1% and the isentropic efficiency increases by 1.04% to 96.35%. Furthermore, Figure 6.33 (left)

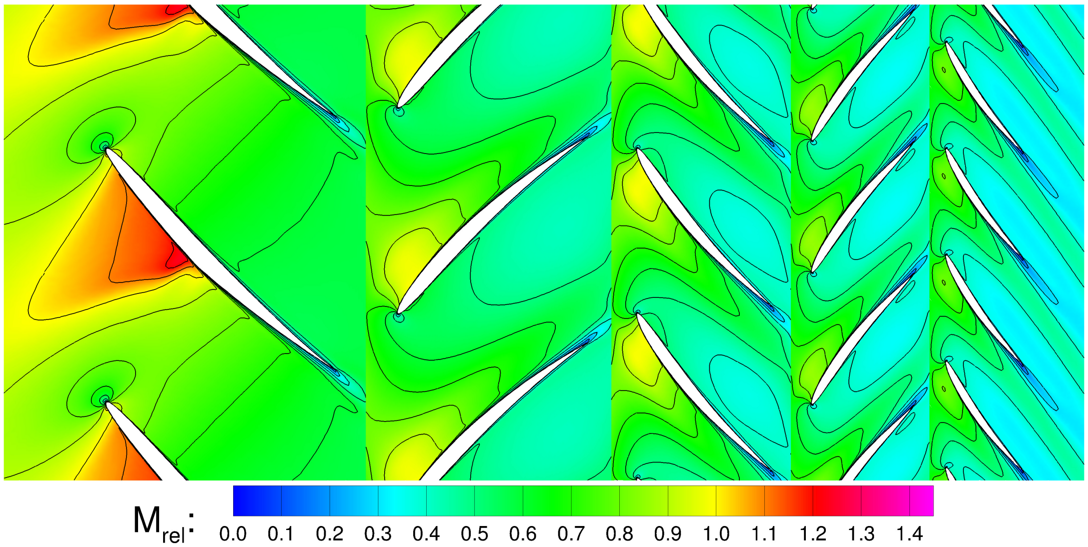

Figure 6.30: Relative Mach number contours, Configuration D, optimized design, unconstrained optimization, $\eta_{\text{is}} = 96.35\%$.

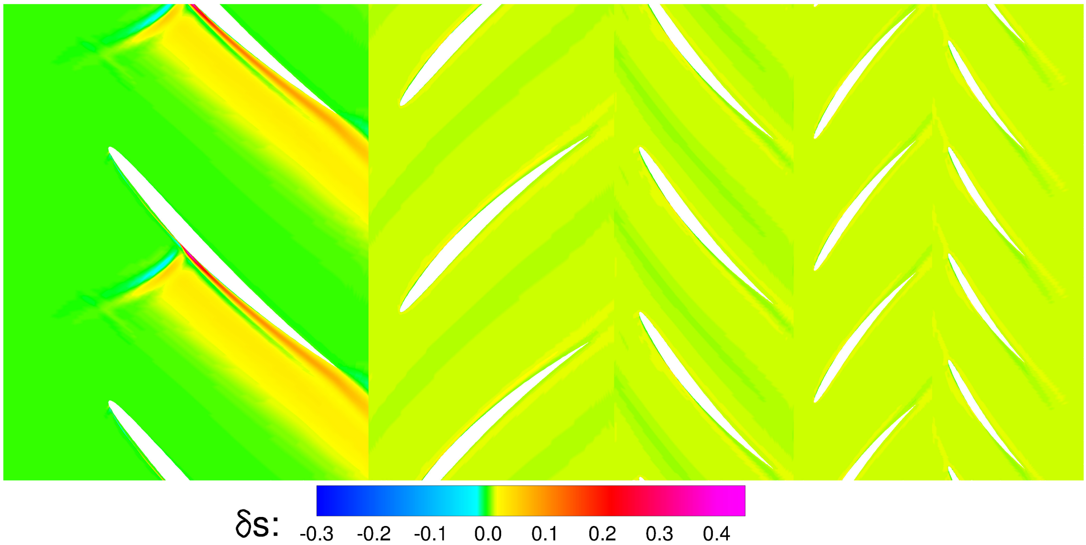

Figure 6.31: Difference in entropy field, $\delta s = s_0 - s_{\text{opt}}$, Configuration D, unconstrained optimization.

illustrates that the optimization terminates after 18 design iterations. However, the final geometry providing the largest entropy decrease, is already obtained after eleven design cycles; only design iterations marked by coloured symbols represent a design improvement over the previous geometry. The last seven design iterations marked by uncoloured symbols indicate additional line searches in which a design proposed

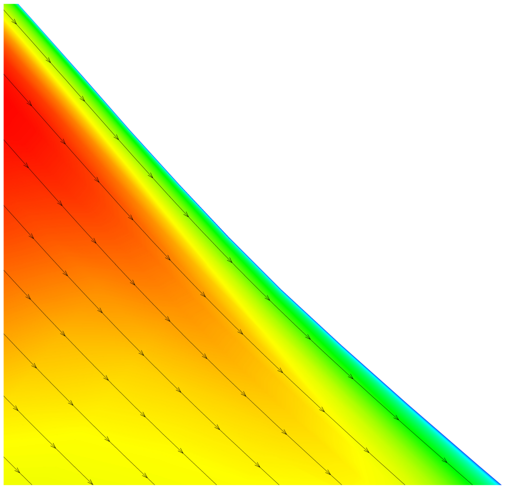

Figure 6.32: Boundary-layer flow after Rotor 1 shock: attached flow for optimized design, Configuration D.

by the optimizer did not lead to an improvement in the objective function; SNOPT aborts the optimization after unsuccessful function evaluations. As it can be seen from Figure 6.33 (right), both the mass flow rate as well as the total pressure ratio change during the unconstrained optimization by 0.67% and 0.17% respectively.

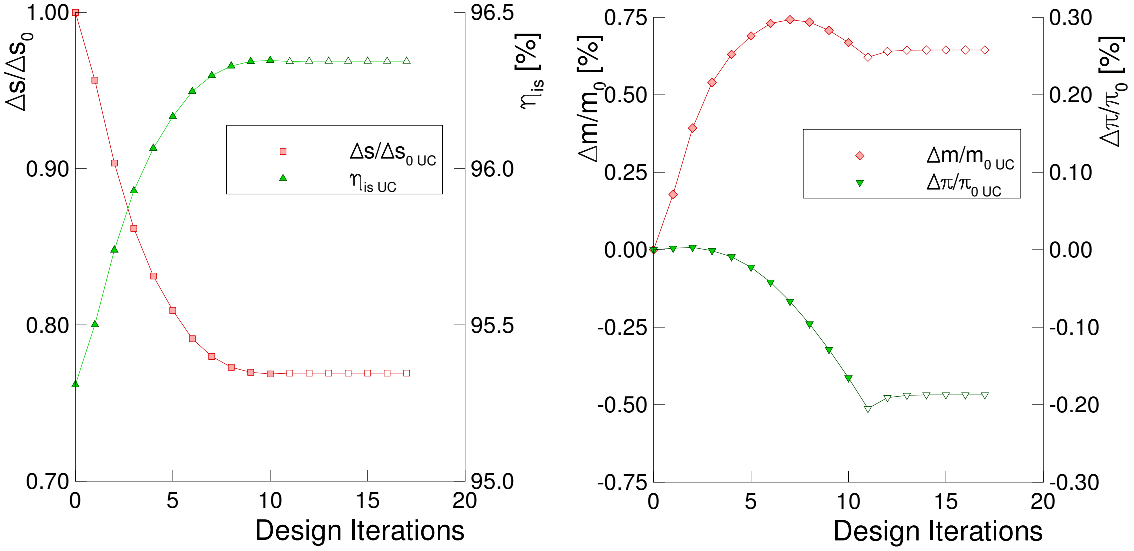

Figure 6.33: Normalized objective function and isentropic efficiency (left), change in mass flow rate and total pressure ratio (right), Configuration D, unconstrained optimization.

### 6.2.6 Constrained Optimization

An improved compressor efficiency of a few percentage points already leads to significant fuel burn savings. Thus, the unconstrained optimization of the transonic compressor stage discussed in subsection 6.2.5 shows a remarkable efficiency gain. Yet, it also reveals that other aerodynamic performance measures change during the design process. However, in most cases at least some of these measures are crucial and are not allowed to change to keep the optimization problem well-posed, e.g. the blade loading due to structural-mechanical reasons or the mass flow rate as it is a global rather than an engine-component-restricted design parameter. Therefore, it usually is necessary to constrain some aerodynamic performance measures during the optimization. Accordingly, in this subsection the compressor efficiency is maximized while constraining the mass flow rate and the total pressure ratio. The bounds on the constraints are defined such that the mass flow rate can vary within ±0.25% and the total pressure ratio is allowed to change by ±0.02% of its original value. The unconstrained optimization presented in the previous subsection produces mass flow rate variations and total pressure ratio changes that exceed the limits of the constrained design case by several fold; thus, this constrained optimization case represents a rather challenging design problem.

The SQP formulation described in section 5.2 enforces the specified constraints through the following procedure. For instance, if the mass flow rate is allowed to vary within ±0.25%, then an inequality constraint given by $-0.25\% \leq c_{\dot{m}} \leq +0.25\%$ is obtained. Slack variables are introduced and the inequality constraint is transformed into an equality constraint $c_{\dot{m}} - \epsilon = 0$, where $-0.25\% \leq \epsilon \leq +0.25\%$. The SQP algorithm then generates a sequence of iterates that satisfies the linear constraints and converge to a point that satisfies the nonlinear constraints (i.e. the slack variables stay within the specified bounds) and the first-order condition for optimality.

The final design of the constrained optimization case is reached after 32 design cycles; or 56 design iterations including the additional line search iterations. Thus, overall the optimization requires 56 flow evaluations and 168 adjoint calculations. Figure 6.34 (left) compares the baseline surface pressure distribution of Rotor 1 with the surface pressure distributions obtained for the unconstrained and constrained Rotor 1 redesigns and documents a reduced flow acceleration in the supersonic flow region and correspondingly a weaker shock on both optimized blades. The optimized Rotor 1 geometries are illustrated in Figure 6.34 (right). Similar to the unconstrained optimization case, the shape modifications of the Rotor 1 geometry exceed those of the other blade geometries. However, compared to the unconstrained redesign the rotor blade curvature is reduced less in the constrained optimization case. The constraints prevent larger profile modifications. The relative Mach number contour and the entropy comparison plots, cf. Figures 6.35 and 6.36, confirm that the redesign leads to a weaker shock and show a thinner boundary layer on the Rotor 1 suction side, which in turn reduces the generation of entropy. Yet, it also can be observed that the shock occurring in the final design of the constrained optimization case is stronger than the Rotor 1 shock observed in the unconstrained redesign.

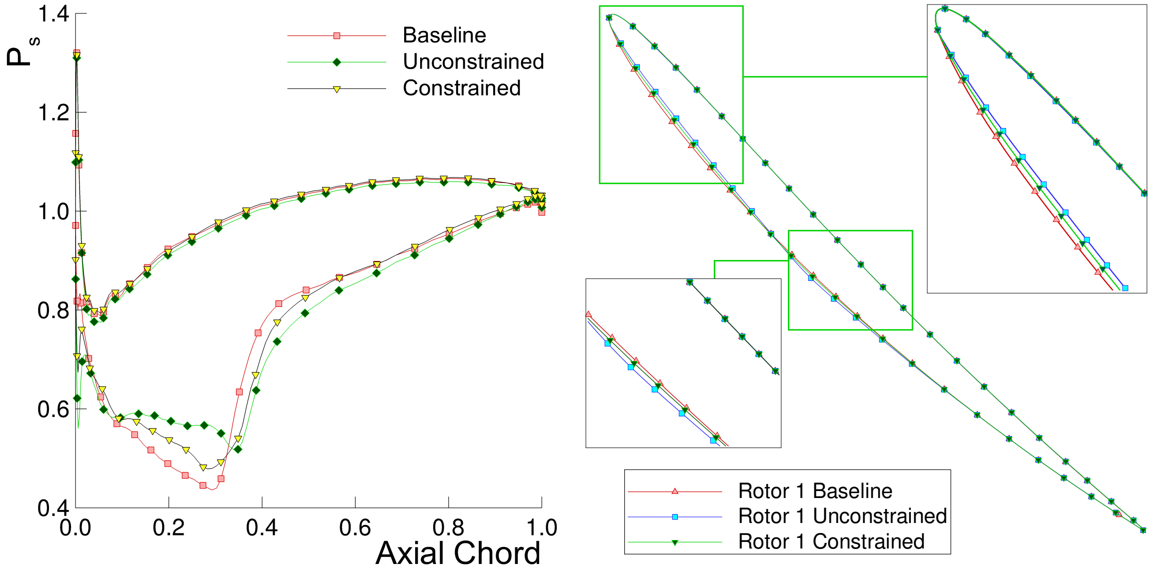

Figure 6.34: Surface pressure distribution (left) and shape modification (right) for Rotor 1, Configuration D, baseline design vs. constrained optimization cases.

Figure 6.37 (left) depicts the convergence of the objective function and the increase in the isentropic efficiency throughout the constrained optimization and compares the results against the unconstrained redesign. Iterations marked by coloured symbols again represent a new design cycle, i.e. a modified geometry that yields an improvement in the objective function while simultaneously satisfying the constraints, whereas iterations marked by uncoloured symbols indicate unsuccessful line searches

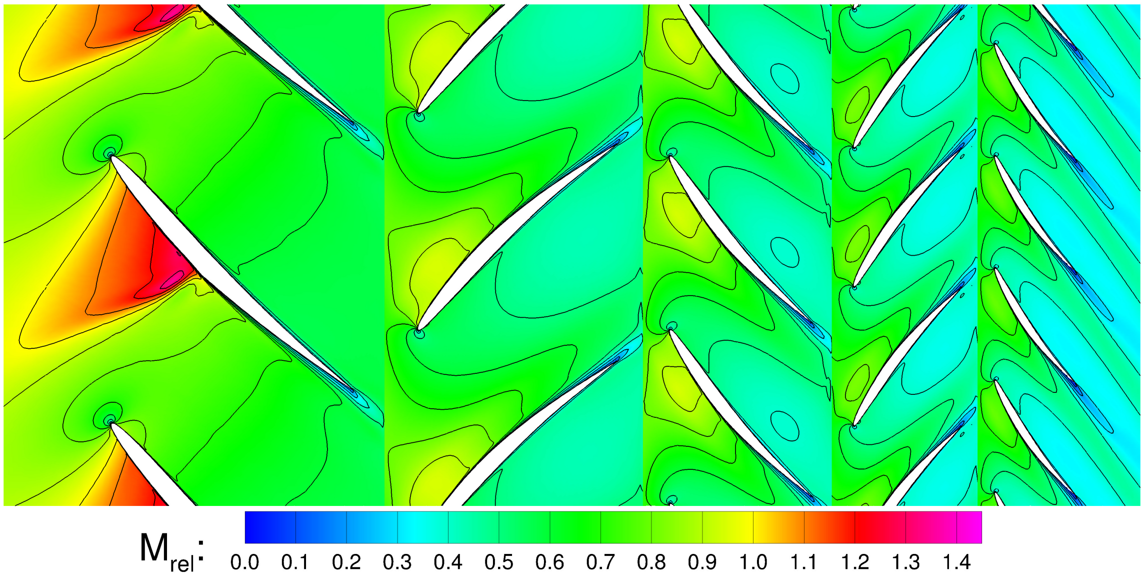

Figure 6.35: Relative Mtheseach number contours, Configuration D, optimized design, constrained optimization, $\eta_{\text{is}} = 96.01\%$.

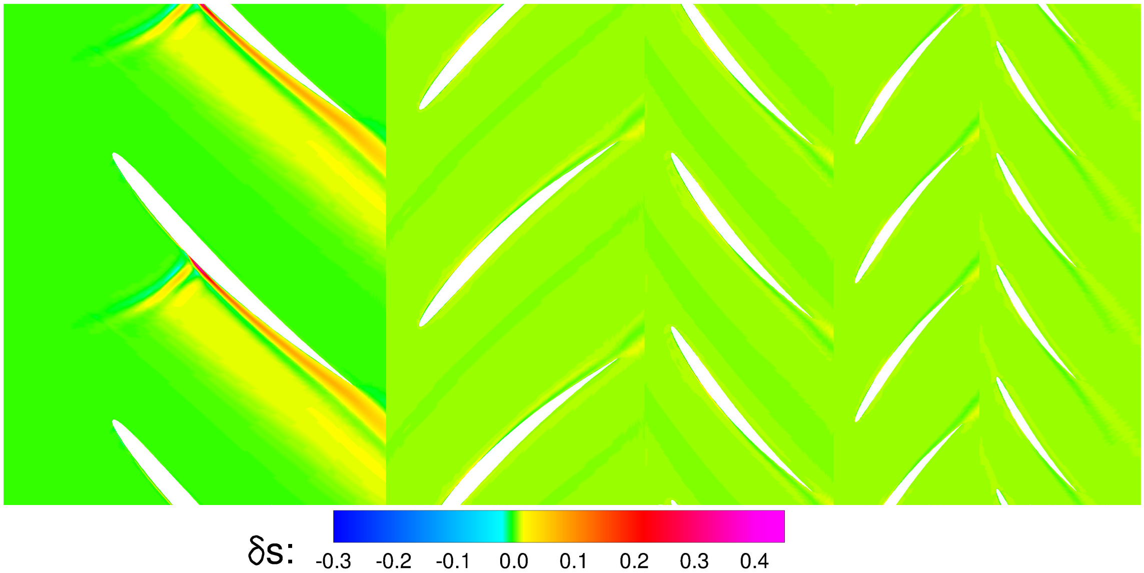

Figure 6.36: Difference in entropy field, $\delta s = s_0 - s_{\text{opt}}$, Configuration D, constrained optimization.

in which either the design objective did not improve or constraints were violated. Figure 6.37 (right), in which the dashed red lines define the bounds on the constraints, displays that the constraints are satisfied at each design cycle. It is worthwhile to mention, that it is the total pressure ratio constraint which prevents the numerical program from finding a more efficient compressor design. The optimization eventually terminates, when the optimizer is unable to find a feasible solution where there is a step length into the search direction that guarantees an improvement in the objective function while satisfying the constraints. In the constrained design case, the objective function is reduced by 16.7%, which in turn leads to an efficiency increase to 96.01% (+0.70%). The total pressure ratio increases by only 0.019%, while the mass flow rate increases by 0.11%.

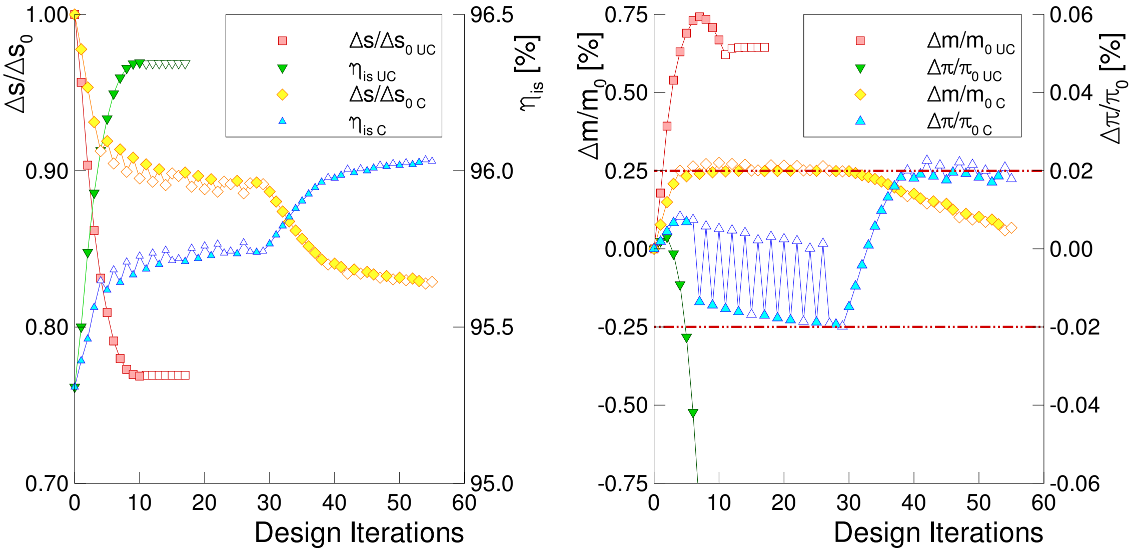

Figure 6.37: Normalized objective function and isentropic efficiency (left), change in mass flow rate and total pressure ratio (right), Configuration D, constrained optimization.

Figure 6.37 (right) also indicates that in the constrained optimization case the search direction changes during the redesign. During the first 29 design iterations the total pressure ratio drops and the mass flow rate increases until both constraints are active. At the 30th design iteration, SNOPT changes the path of the optimization and in subsequent design iterations the total pressure ratio increases while the mass flow rate starts to decrease, while a precipitous improvement in the entropy reduction and isentropic efficiency increase is observed in Figure 6.37 (left). A comparison of the Rotor 1 shape at the 30th design iteration with the final optimized design, cf. Figure 6.38, provides a geometrical illustration of the sudden change in the search direction. While the suction side curvature is constantly reduced in the supersonic flow region of Rotor 1 from the first until the last design cycle, the aft part of the blade remains nearly unchanged until the 30th design iteration. At this juncture, the profile curvature is increased in the mid and trailing edge region of the airfoil, which leads to higher flow turning and consequently increases the work and the total pressure ratio as depicted in Figure 6.37 (right). Thus, the constrained optimization case demonstrates the effectiveness of employing a sequential quadratic programming algorithm within the proposed adjoint-based optimization framework, where the evaluation of the gradient of the constraints via the adjoint approach reduces the total computational cost, but most importantly the SQP method provides a means to find feasible optimum solutions which may not be realizable through a penalty function approach.

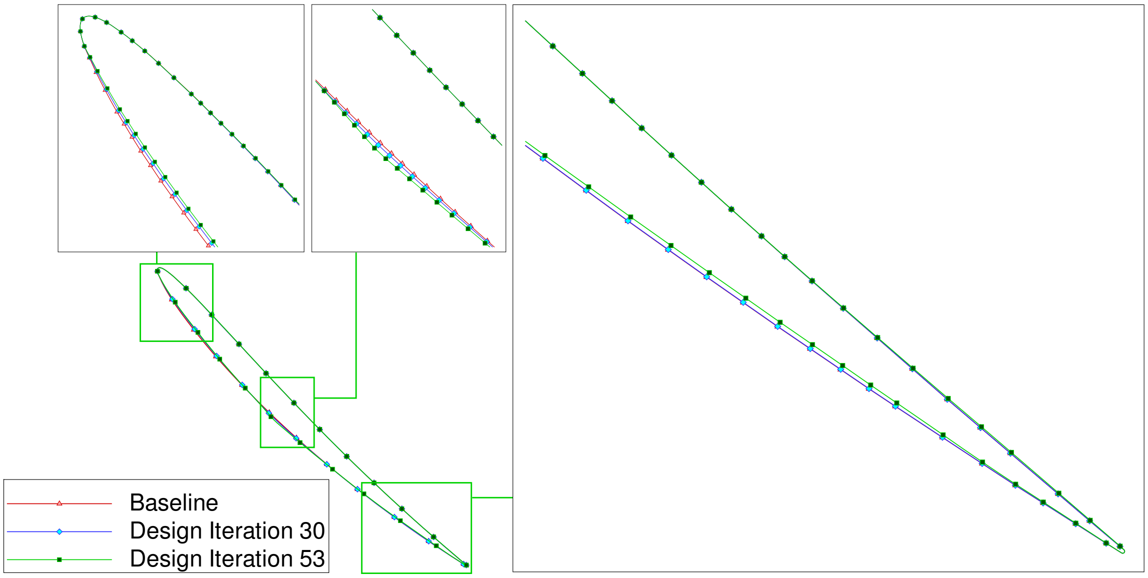

Figure 6.38: Shape modification Rotor 1 throughout the constrained optimization of Configuration D.

A comparison of the aerodynamic performance measures, total pressure ratio $\pi$, work $h_t$, and isentropic efficiency $\eta_{\text{is}}$ between the optimized rotor designs and the corresponding baseline rotors provides another interesting insight into the physical changes occurring during the optimization. First, from Table 6.9 it can be seen that the design objective, to maximize the compressor's isentropic efficiency, is achieved by increasing the isentropic efficiency of each individual rotor. The greatest performance

| | $\Delta\pi$ [%] | $\Delta h_t$ [%] | $\Delta\eta_{\text{is}}$ [%] |
|---|---|---|---|
| Rotor 1, Unconstrained | $+0.1044$ | $-2.7100$ | $+2.9179$ |
| Rotor 2, Unconstrained | $-0.0658$ | $-1.3741$ | $+0.0500$ |
| Rotor 3, Unconstrained | $-0.0365$ | $-0.9024$ | $+0.0636$ |
| Rotor 1, Constrained | $+0.4778$ | $-0.8388$ | $+2.0841$ |
| Rotor 2, Constrained | $-0.2461$ | $-1.5270$ | $+0.0523$ |
| Rotor 3, Constrained | $-0.2823$ | $-1.4547$ | $+0.0381$ |
| Rotor 1, Constrained 30th | $+0.1183$ | $-0.9310$ | $+1.2081$ |
| Rotor 2, Constrained 30th | $-0.0591$ | $-0.6246$ | $+0.0465$ |
| Rotor 3, Constrained 30th | $-0.1170$ | $-0.7101$ | $+0.0016$ |

Table 6.9: Comparison of the aerodynamic performance of each optimized rotor design of Configuration D.

improvements are obtained for Rotor 1 ($\Delta\eta_{\text{is}} = +2.08\%$ and $+2.92\%$), while the efficiency increase observed for the Rotor 2 and Rotor 3 rows is considerably smaller ($\Delta\eta_{\text{is}} \leq +0.06\%$). This is in agreement with the observations made earlier. The efficiency benefit of the optimized Rotor 1 row is mainly due to the weaker shock and the resulting thinner boundary layer. Since the entropy production within the Rotor 2 and Rotor 3 rows is due to viscous effects only and hence at a much lower level, the efficiency improvement is consequently smaller as well. Second, a comparison of the rotor work, $h_t = \Omega_2 r \cdot (u_{2,\text{out}} - u_{2,\text{in}})$, reveals that the work decreases in each rotor row. In the unconstrained test case, the work performed by the optimized Rotor 1 is 2.71% less than the work done by the corresponding baseline rotor. The work of the optimized Rotor 2 and Rotor 3 rows drops by 1.37% and 0.90% respectively. Accordingly, the total pressure ratio drops in both the second and third rotor rows. However, Table 6.9 depicts that, despite a work decrease compared to the baseline design, the total pressure ratio increases for the Rotor 1 row. The reduced shock losses outweigh the effects of the work decrease and are responsible for the increased total pressure ratio. Third, comparing the different optimized designs with each other shows that the aerodynamic performance of the rotor rows of the constrained design case differs. In the constrained design case, the work load for Rotor 1 is initially reduced as observed in the unconstrained design case. At the 30th design cycle, the work performed in the first transonic rotor row is 0.93% less than in the baseline configuration. However, after the change in the search direction that is documented in Figure 6.37 (right), the work load increases again due to the increase in profile curvature and higher flow turning, cf. also Figure 6.38. The final design shows the Rotor 1 work load is 0.84% less than that of the baseline design and, hence, is 0.09% higher than at the 30th design cycle and noticeably higher than in the final design of the unconstrained case. Similarly, the total pressure ratio of Rotor 1 is considerably increased after the change in the search direction and therefore is much higher than in both the baseline and the unconstrained optimized solution. At the same time, the drop in the total pressure ratio for Rotor 2 and Rotor 3 is larger than in the unconstrained design case.

Table 6.10 summarizes the aerodynamic performance parameters of the multistage compressor obtained for the constrained optimization case and compares them with the values from the baseline design and the unconstrained optimization. Despite the quite strict constraints on the total pressure ratio and the mass flow rate, the constrained optimization case leads to a considerable efficiency improvement. Although a small efficiency deficit exists compared to the unconstrained optimization, the constrained redesign ensures that other crucial aerodynamic parameters do not change during the optimization process, which is not necessarily the case for the unconstrained optimization.

| | $\eta_{\mathrm{is}}$ [%] | $\pi$ | $\frac{\dot{m}_{\mathrm{opt}}}{\dot{m}_0}$ |
|---|---|---|---|
| Baseline | 95.31 | 3.00997 | 1.0000 |
| Unconstrained | 96.35 | 3.00485 | 1.0067 |
| Constrained | 96.01 | 3.01054 | 1.0011 |

Table 6.10: Performance comparison between the original and optimized multistage compressor designs, Configuration D.

## 6.3 Darmstadt Rotor No. 1

In this section, the single-stage axial flow compressor Darmstadt Rotor No. 1 is investigated and redesigned considering various geometrical and physical settings. Subsection 6.3.2 discusses the results of an initial grid study, verifies the numerical solution by comparing it against experimental and computational data published by various authors, and determines the final grid dimensions. The subsections 6.3.3 and 6.3.4 show the adjoint solution and validate the adjoint gradients by comparing them with sensitivities obtained from finite-differences. Afterwards, several numerical optimization cases are presented. First, a two-dimensional section slice of the transonic compressor configuration is redesigned considering a fully-turbulent viscous flow environment, cf. subsection 6.3.5. Then, the developed optimization framework is applied to the entire three-dimensional Darmstadt Rotor No. 1; thus, the complete domain between the hub and casing is considered. However, for the sake of simplicity the rotor gap is removed and the blade is extended until the casing. The results of a constrained optimization considering inviscid flow are discussed in subsection 6.3.6. The redesign obtained from an optimization assuming a fully-turbulent flow field is presented in subsection 6.3.7.

### 6.3.1 Test Case

The second test case is a three-dimensional single-stage transonic axial flow compressor, commonly known as Darmstadt Rotor No. 1, which was developed at MTU Aero Engines, Munich, Germany and extensively tested at the Darmstadt University of Technology, Germany. The original rotor was designed as a bladed disk made of titanium with conventionally stacked blade sections, cf. Figure 6.39. From the mid 1990s to the early 2000s, the transonic compressor was representative of a front stage of contemporary high pressure compressors. Its main aerodynamic design parameters are given in Table 6.11. For a detailed description of the compressor rig and the experiments the reader should refer to Blaha et al. [89]; a cross section of the test rig is depicted in Figure 6.40. Numerical investigations were presented by Fritsch et al. [90] and Höger et al. [91] To simplify the computational model, in this research work the rotor gap is removed and the rotor blade is extended until the casing. Fig-

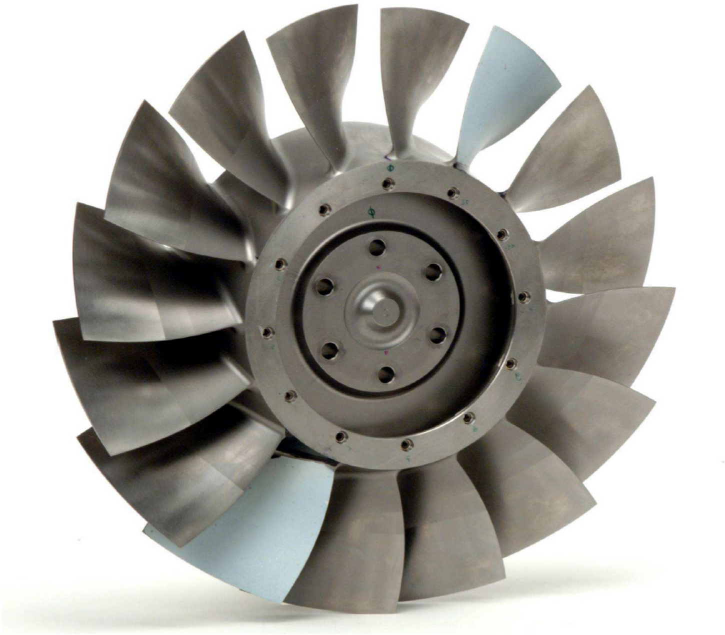

Figure 6.39: Transonic compressor blade disk rotor made of titanium.

| | |
|---|---|
| Number of rotor blades | 16 |
| Number of stator blades | 29 |
| Rotational speed | 20,000 rpm |
| Total pressure ratio | 1.5 |
| Mass flow rate (corrected) | 16.0 kg/s |
| Rotor tip speed | 398 m/s |
| Inlet Rel. Mach number (tip) | 1.35 |
| Inlet Rel. Mach number (hub) | 0.70 |
| Rotor aspect ratio | 0.88 |
| Stator aspect ratio | 1.50 |
| Rotor solidity (hub/mid/tip) | 1.9/1.5/1.2 |
| Stator solidity (hub/mid/tip) | 2.0/1.6/1.3 |
| Annulus diameter | 0.38 m |

Table 6.11: Design parameters Darmstadt Rotor No. 1.

ure 6.41 depicts a section slice through the three-dimensional block-structured grid at 50% span. Similarly as for the Configuration D test case, O-grids are used to discretize the domain in the vicinity of a blade surface, while H-topology grids are utilized to model the remaining passage. A dimensionless wall distance of $y^+_{\mathrm{avg}} = 1.0$ guarantees that the viscous boundary layer effects are modelled accurately. The grid independence of the flow solution is verified by a second grid study, cf. section 6.3. The numerical results are further verified by comparing them with experimental data.

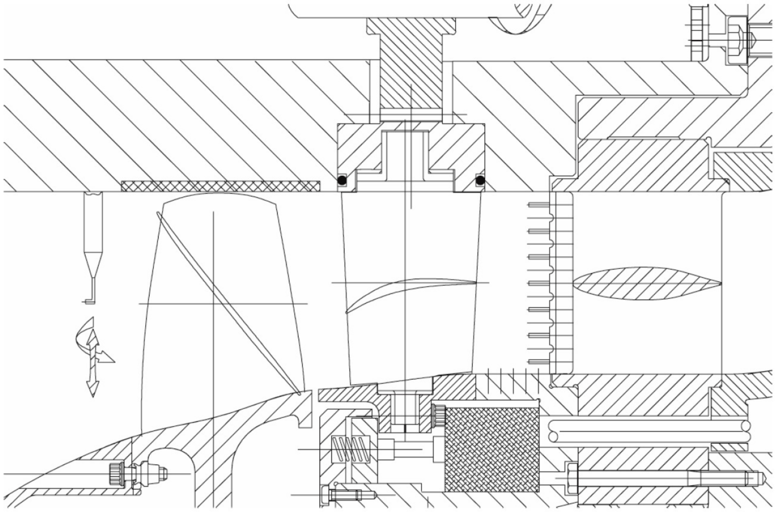

Figure 6.40: Cross section of Darmstadt compressor test rig.

### 6.3.2 Flow Solution

The appropriate grid dimensions of the structured multiblock grid for the Darmstadt Rotor No. 1 compressor, depicted in Figure 6.41, are determined by a grid study similar to one performed for the two-dimensional Configuration D compressor. The numerical solutions are obtained on two grids of the same topology but different number of grid points are compared. Starting from the coarser grid, the grid dimensions are doubled in axial, circumferential, and radial directions to obtain the second grid. The dimensions of the finer grid are as follows: The rotor domain consists of 1,129,285 grid points, while 827,424 grid points are used to discretize the stator

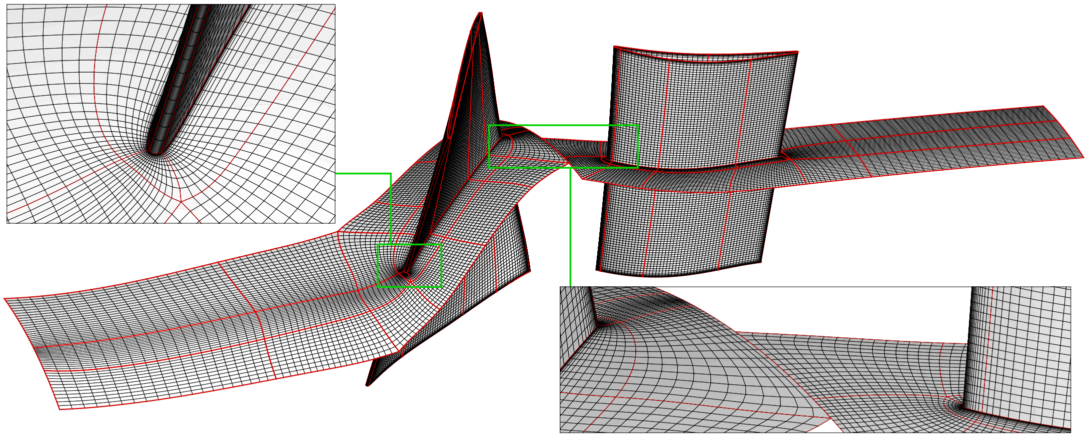

Figure 6.41: Computational grid for Darmstadt Rotor No. 1, section slice at 50% span.

domain. The rotor and stator blade surfaces are represented by 197 and 145 grid points respectively, 65 grid points are used to discretize the channel section between two adjacent rotor or stator blades, and 117 grid points are distributed in the radial direction. To resolve the boundary layer accurately, the wall-spacing is chosen such that $y^+_{\mathrm{avg}} = 1.0$.

The same initialization strategy as for the Configuration D test case is employed. Before solving for the viscous Navier-Stokes equations, a meaningful starting solution is established by solving the inviscid Euler equations and progressively increasing the back pressure and rotational velocity until the desired exit pressure and the required rotor entrainment velocity are obtained. Figure 6.42 illustrates the convergence of the maximum density residual for the flow calculations on both grids. The two computations are started from a previously calculated viscous flow field that was only converged by approximately three orders. Within less than 5000 multigrid cycles, the flow solver then fully converges the governing equations on the coarser grid, whereas on the fine grid the density residual is reduced by six orders.

A comparison of the main aerodynamic design parameters, cf. Table 6.12, confirms that the resolution of the fine grid is sufficient. In fact, even the coarser grid provides reasonable results; the differences in the total pressure and temperature ratio as well as the change in the isentropic efficiency and the mass flow rate between the coarse and the fine grids are negligible. Yet, some slight local differences in the flow

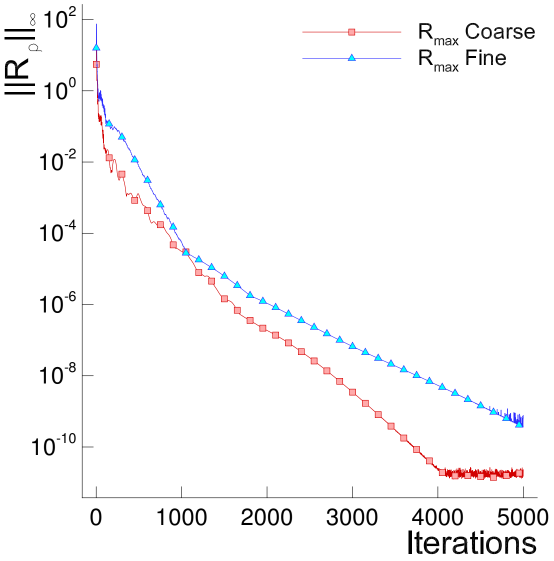

Figure 6.42: Convergence of maximum density residual, Darmstadt Rotor No. 1

solution, e.g. in the location and clarity of the shock, can be observed when comparing the flow field predicted on the coarser grid with the flow solution calculated on the finer grid. The compressor stage pressure ratios are also in good agreement with the numerical and experimental results provided by Fritsch et al. [90] and Höger et al. [91] for the compressor design point. The isentropic efficiency values are over-predicted; however, this was to be expected since in the presented research work the rotor gab is not modelled. Based on these observations, the fine grid is selected and used for the optimization cases presented in the following subsections.

In addition to the viscous grid, a grid for inviscid flow is generated. Due to the absence of a boundary layer in inviscid flow, it is not necessary to resolve the

| Design parameter | $\pi$ | $T_{t,\text{out}}/T_{t,\text{in}}$ | $\eta_{\text{is}}$ [%] | $\dot{m}/\dot{m}_{\text{fine}}$ |
|---|---|---|---|---|
| Coarse Grid | 1.505 | 1.138 | 89.43 | 0.995 |
| Fine Grid | 1.504 | 1.138 | 89.49 | 1.000 |
| Numerical [90] | 1.491 | — | 88.3 | — |
| Experiment [90] | 1.483 | — | 86.7 | — |

Table 6.12: Design parameter comparison, Darmstadt Rotor No. 1.

spatial domain in the vicinity of a blade as detailed as in viscous flow problems. Consequently, the number of grid points can be reduced. For the inviscid grid, 460,265 grid points are used to discretize the rotor domain and the stator domain consists of 367,900 grid points. As in the viscous grid, the rotor and stator blade surfaces are represented by 197 and 145 grid points respectively; however, only 49 grid points are used to discretize the channel section between two adjacent rotor or stator blades and only 65 grid points are placed in the radial direction.

### 6.3.3 Adjoint Solution

The convergence levels of both the flow and adjoint equations required for an accurate gradient calculation are determined from the same convergence study described in subsection 6.2.3. To reduce the number of multigrid cycles necessary to converge the equations to a sufficient level, both the flow solution and the adjoint solution are initialized with approximate solutions obtained from a previous numerical calculation. In the following, the results for the entropy functional are presented exemplary, i.e. the study in which the adjoint equations are solved for the main design objective of the optimization cases. The conclusions that can be drawn from the investigation of other functionals such as the mass flow and total pressure ratio constraints are similar to those presented below.

Two cases are considered, both of which are summarized in Table 6.13. The convergence of the flow governing equations is depicted in Figure 6.43 (left), whereas the convergence of the adjoint equations is illustrated in Figure 6.43 (right). The results resemble those of the Configuration D compressor; the convergence curves of the different adjoint calculations match almost perfectly, despite the fact that the convergence level of the underlying flow solutions is different. Table 6.14 shows the

| Case | Flow Cycles | Adjoint Cycles | $\|\|\mathrm{R}_\rho\|\|_\infty$ | $\|\|\mathrm{R}_{\psi_1}\|\|_\infty$ |
|---|---|---|---|---|
| C1 | 5000 | 5000 | $7.39\mathrm{e}^{-10}$ | $1.43\mathrm{e}^{-11}$ |
| C2 | 2500 | 2500 | $1.48\mathrm{e}^{-7}$ | $4.54\mathrm{e}^{-7}$ |

Table 6.13: Test cases investigated in the convergence study, Darmstadt Rotor No. 1.

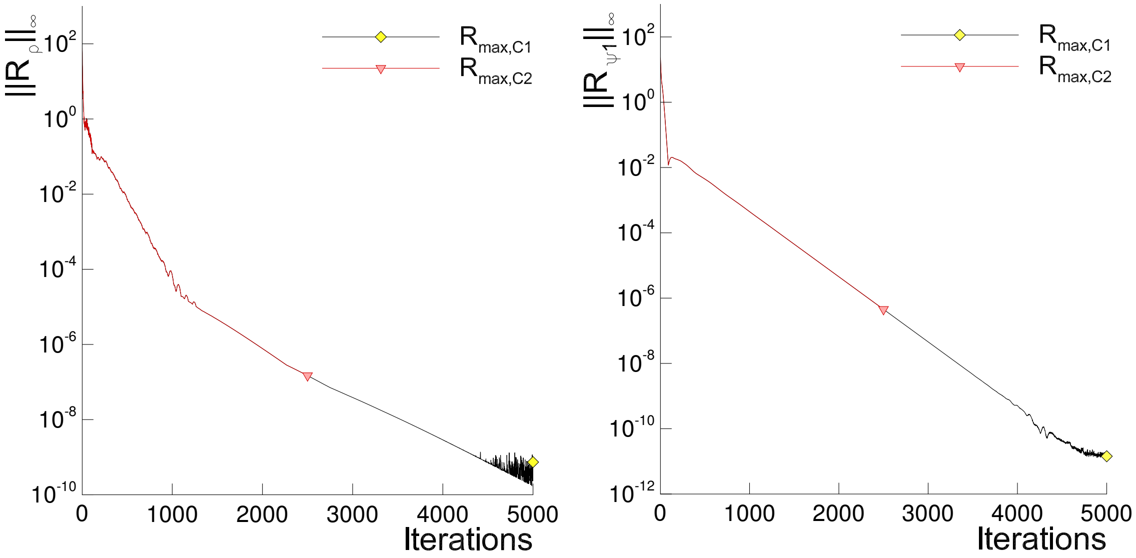

Figure 6.43: Convergence level of maximum density residual (left) and maximum co-state $\psi_1$-residual (right) for various multigrid cycles, Darmstadt Rotor No. 1.

maximum difference between the partially-converged adjoint solution of test case C2 and the fully-converged solution of test case C1. The impact on the gradient accuracy is depicted in Table 6.15. It can be observed that the differences between the adjoint sensitivities of the two test cases are negligible for both the rotor and stator blades.

The results of the convergence study demonstrate that the convergence levels obtained in the test case C2 lead to a sufficiently accurate flow and adjoint solution and confirm that generally it is not necessary to converge either equations by more

| $i$ | $\psi_{i,\mathrm{C1}}$ | $\lvert\Delta\psi_{i,\mathrm{C2}}\rvert$ |
|---|---|---|
| 1 | -1.85 … 6.01 | 1.43e$^{-6}$ |
| 2 | -0.78 … 4.32 | 1.07e$^{-6}$ |
| 3 | -1.81 … 2.45 | 1.19e$^{-7}$ |
| 4 | -1.17 … 1.12 | 1.04e$^{-7}$ |
| 5 | -1.64 … 0.21 | 1.07e$^{-6}$ |

Table 6.14: Minimum and maximum value of adjoint solution for test case C1 and error of test cases C2, Darmstadt Rotor No. 1.

| Row | $G_{C1}$ | $\lvert\Delta G_{C2}\rvert$ |
|---|---|---|
| Rotor | $-4.48e^{-2}$ … $5.82e^{-2}$ | $1.00e^{-10}$ |
| Stator | $-2.08e^{-2}$ … $2.30e^{-2}$ | $1.00e^{-11}$ |

Table 6.15: Minimum and maximum gradient for test case C1 and error of test case C2, $\Delta G_{C2} = G_{C1} - G_{C2}$, Darmstadt Rotor No. 1.

than five to six orders to obtain accurate gradient information. For the design cases presented in the following, the number of flow and adjoint iterations are chosen accordingly.

### 6.3.4 Gradient Validation

To further strengthen the confidence in the accurateness of the developed adjoint boundary conditions, the adjoint solver and gradient routines are validated for the second compressor configuration as well. Two different set-ups of the Darmstadt Rotor No. 1, for which the adjoint sensitivities are compared with finite-difference gradients, are considered. The finite-difference gradients are calculated as described in subsection 6.2.4; the adjoint boundary conditions and the mixing-plane formulation are again validated based on the inviscid Euler equations.

**Test Case I: 2D section at 55% span**

First, a two-dimensional flow environment is assumed and only the section slice at 55% span is analysed. The inviscid flow solution, depicted in Figure 6.44 (left), is calculated on a two-dimensional multiblock grid which possesses the same topology as the 50% span cut shown in Figure 6.41. Similar as in the previous test cases, a supersonic flow region develops in the front part of the rotor suction side and a strong shock, located just in front of the rotor passage, abruptly decelerates the flow to subsonic regimes. The rotor passage flow itself, as well as the flow in the downstream stator domain, remains subsonic. Figure 6.44 (right) visualizes the adjoint solution for the co-state variable $\psi_1$, i.e. the adjoint variable corresponding to the continuity equation, for the mass flow rate functional. Again, 78 Hicks-Henne bump functions

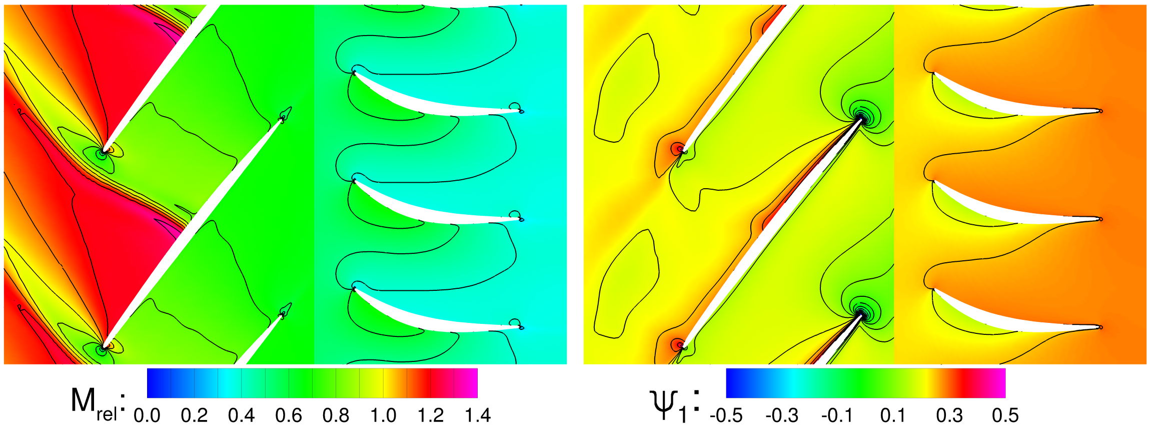

Figure 6.44: Relative Mach number contours (left) and $\psi_1$-contours (right), Darmstadt Rotor No. 1, test case I: 2D section at 55% span, functional: mass flow rate.

are defined as design variables and are equidistantly distributed on both the rotor and stator blades. Figures 6.45 to 6.47 compare the adjoint sensitivities with the finite-difference gradients for different functions of interest. For all three functionals the adjoint sensitivities match the gradients obtained from the forward approach very well and, thus, confirm that the adjoint boundary conditions and the adjoint mixing-plane formulation are flow-consistent and accurate.

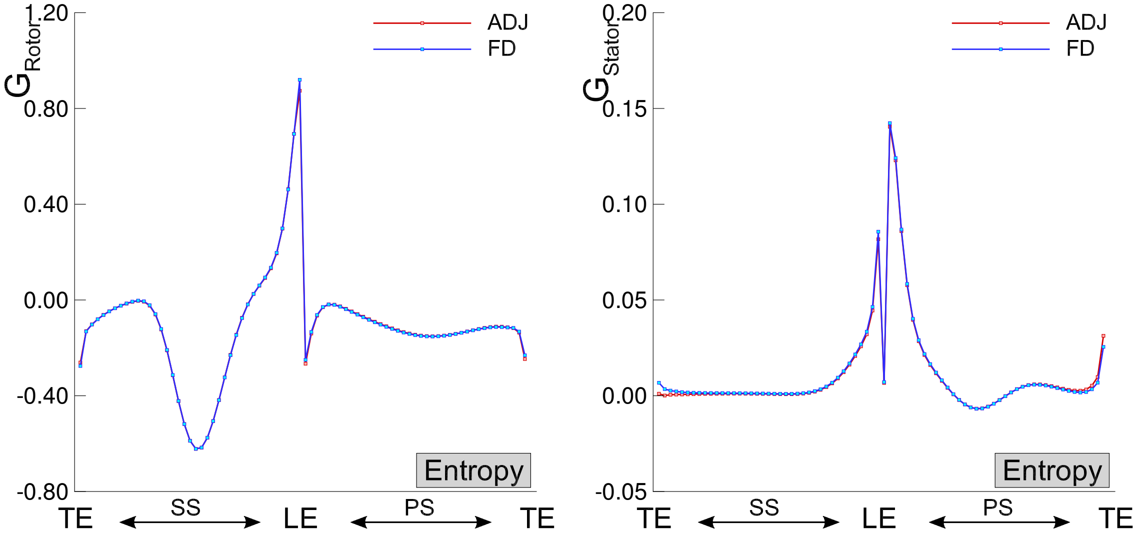

Figure 6.45: Gradient comparison rotor (left) and stator (right), Darmstadt Rotor No. 1, test case I: 2D section at 55% span, functional: entropy generation rate.

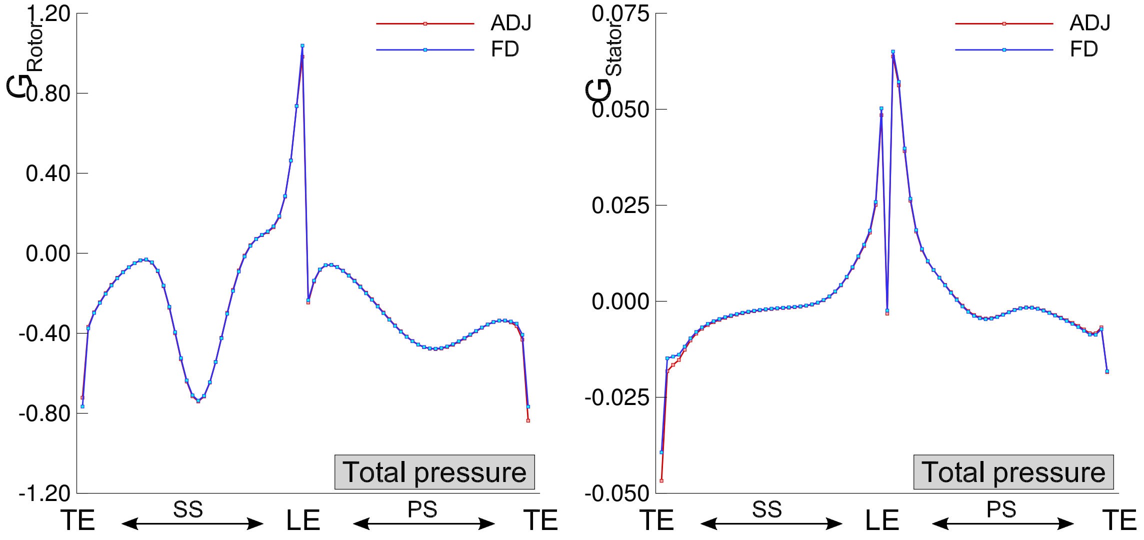

Figure 6.46: Gradient comparison rotor (left) and stator (right), Darmstadt Rotor No. 1, test case I: 2D section at 55% span, objective function: total pressure ratio.

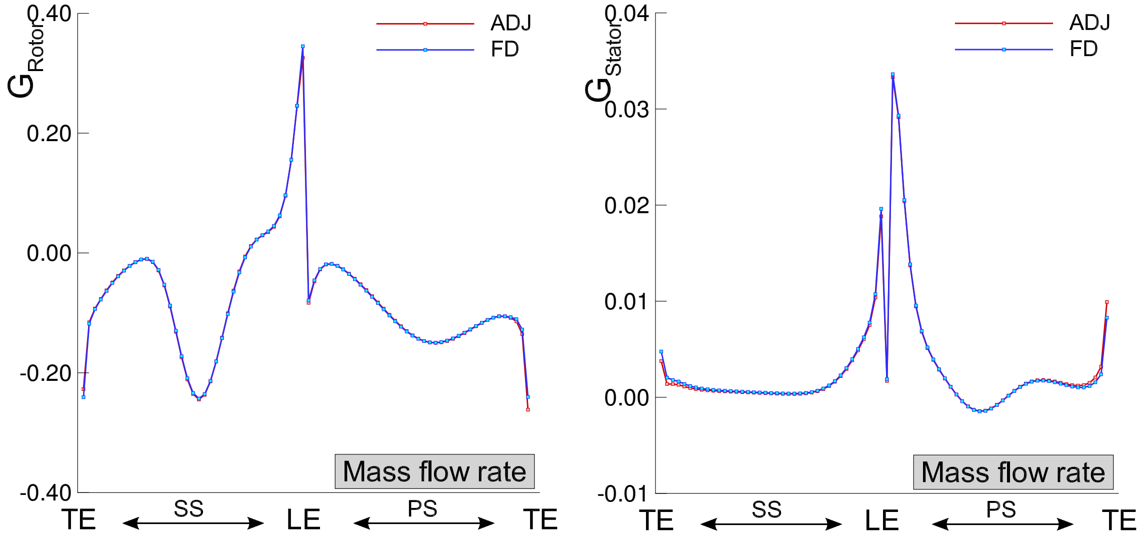

Figure 6.47: Gradient comparison rotor (left) and stator (right), Darmstadt Rotor No. 1, test case I: 2D section at 55% span, objective function: mass flow rate.

### Test Case II: 3D Configuration

To verify the adjoint boundary conditions and particularly the adjoint mixing-plane formulation in a three-dimensional flow environment, the entire Darmstadt Rotor

No. 1 compressor is investigated, i.e. in this last gradient verification test case the complete domain between the hub and casing is considered. The flow solution is visualized in Figure 6.48. From the relative Mach number contour plot it can be observed, that at 50% span the shock structure extends into the rotor passage, where it stretches across the entire cross section. Thus, at the midspan section the passage is chocked. Figure 6.49 displays the $\psi_2$-contours of the adjoint solution, i.e. the co-state variable corresponding to the momentum equation in the $x_1$-direction, for the entropy generation rate as the objective function. In this test case, 81 Hicks-Henne bump functions are equidistantly distributed on each blade at 50% span; i.e. 41 design variables are defined on the midspan section between leading (LE) and trailing edge (TE) on each suction and pressure side. Figure 6.50 compares the

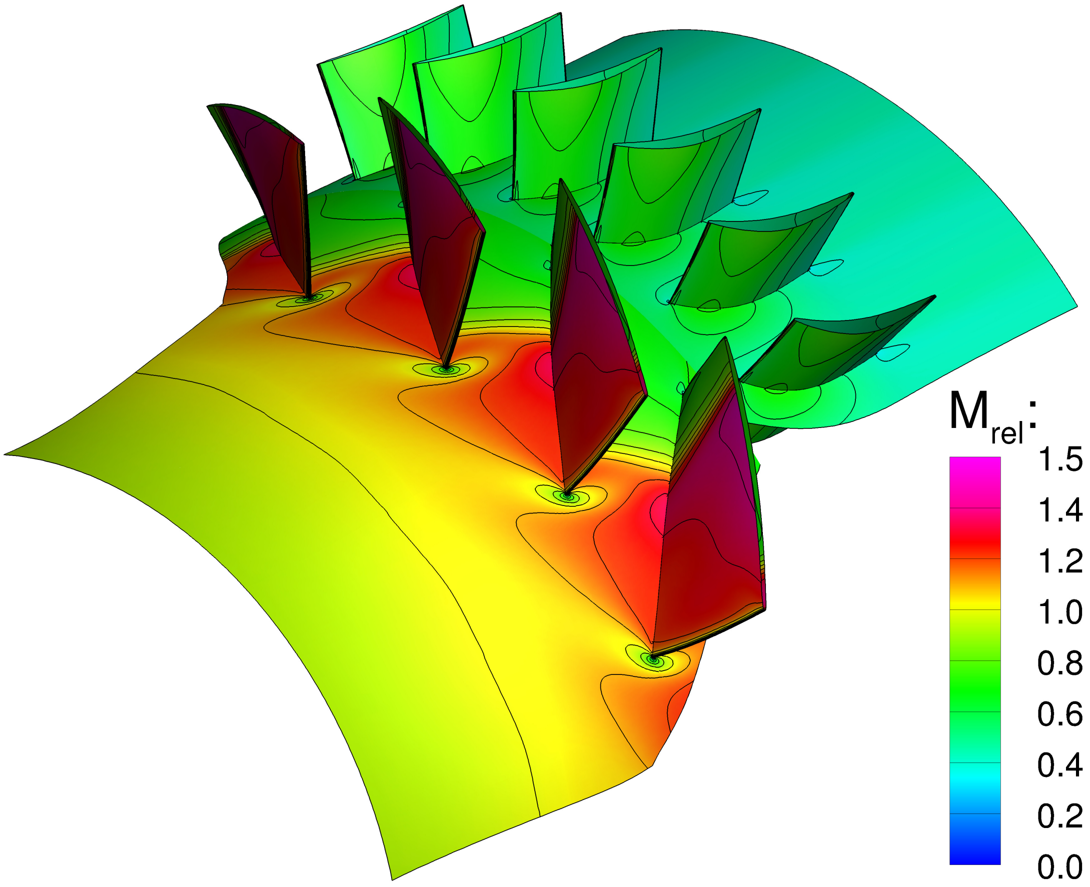

Figure 6.48: Relative Mach number contours, Darmstadt Rotor No. 1, test case II: 3D configuration, plotted section slice: 50% span.

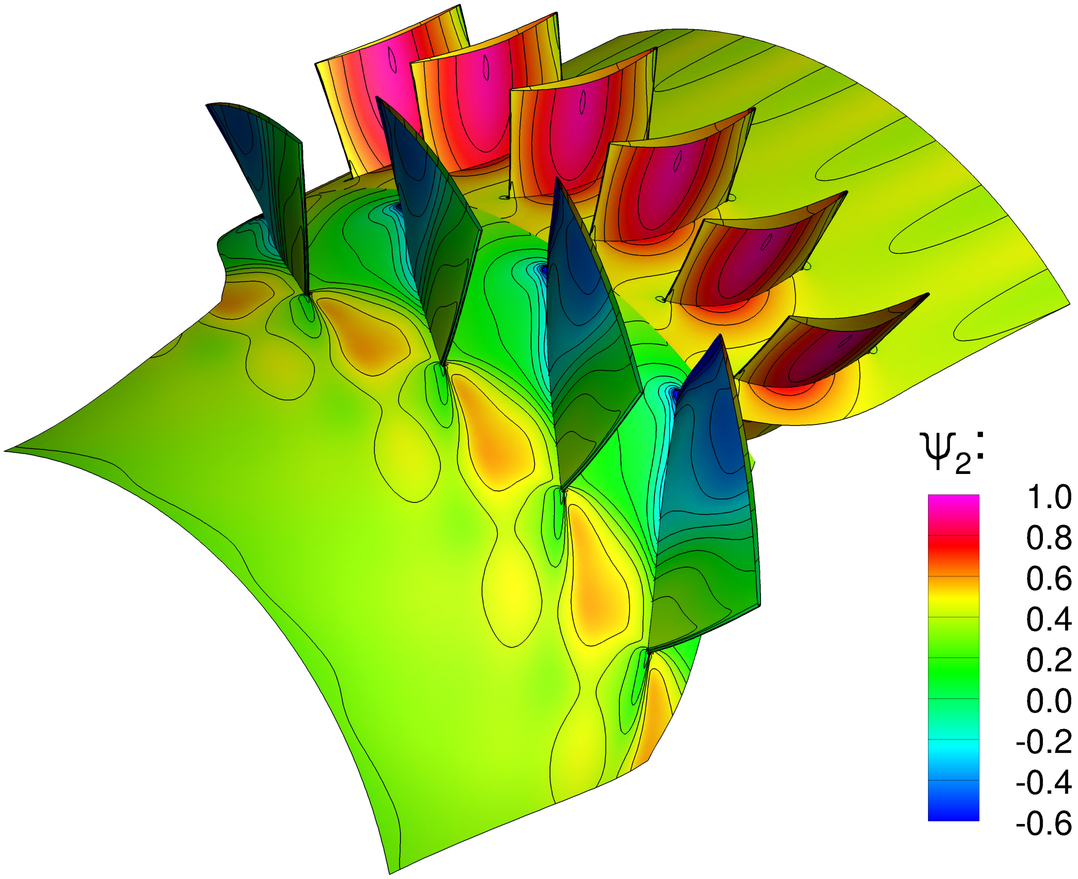

Figure 6.49: $\psi_2$-contours, Darmstadt Rotor No. 1, test case II: 3D configuration, objective function: entropy generation rate, plotted section slice: 50% span.

adjoint sensitivities with the finite-difference gradient for the entropy functional; the vertical axis on the left plots the rotor gradients while the sensitivities of the stator blade are scaled by the vertical axis on the right. The gradient comparison for the total pressure ratio functional is depicted in Figure 6.51. The adjoint sensitivities match the finite-difference gradients very well and the plots verify the accuracy of the three-dimensional adjoint solver. The results particularly confirm that the three-dimensional adjoint boundary conditions are flow-consistent as well.

This completes the gradient study. The remaining sections of this chapter present several optimization cases in which both compressor configurations are redesigned.

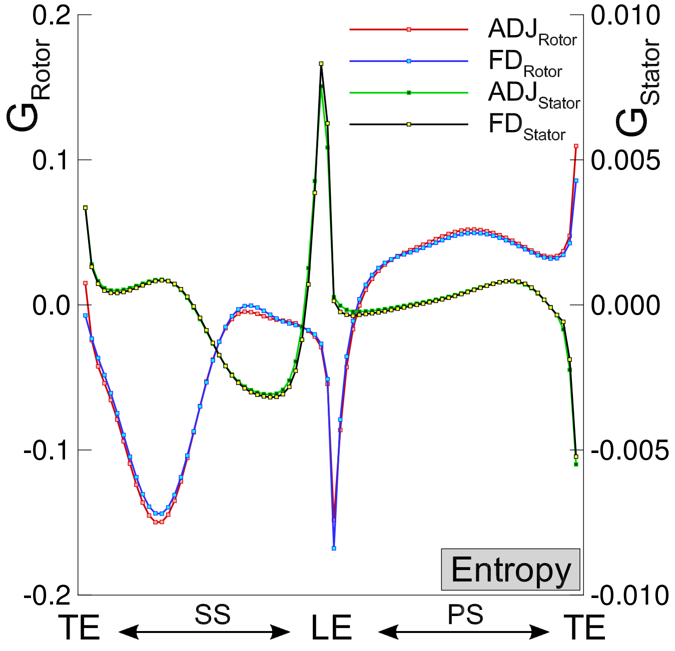

Figure 6.50: Gradient comparison rotor and stator, Darmstadt Rotor No. 1, test case II: 3D configuration, objective function: entropy generation rate.

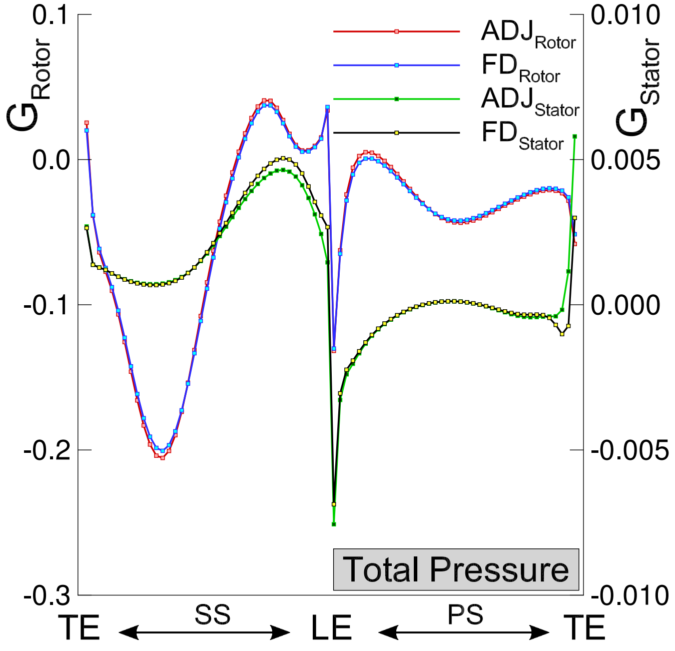

Figure 6.51: Gradient comparison rotor and stator, Darmstadt Rotor No. 1, test case II: 3D configuration, objective function: total pressure ratio.

### 6.3.5 Optimization in Two-Dimensional Viscous Flow

Before redesigning the entire Darmstadt Rotor No. 1, first a two-dimensional section slice of the transonic single-stage compressor is examined and optimized in isolation. Since three-dimensional flow effects typically are less dominant in the midspan region, the compressor geometry is extracted and investigated at 55% span. Figure 6.52 depicts the block-structured grid of the section slice in more detail. The rotor domain consists of 9529 grid points, while 7021 points are used to discretize the stator domain. The rotor and stator blade surfaces are represented by 197 and 145 grid points respectively, 65 grid points are used to discretize the channel section between two adjacent rotor or stator blades.

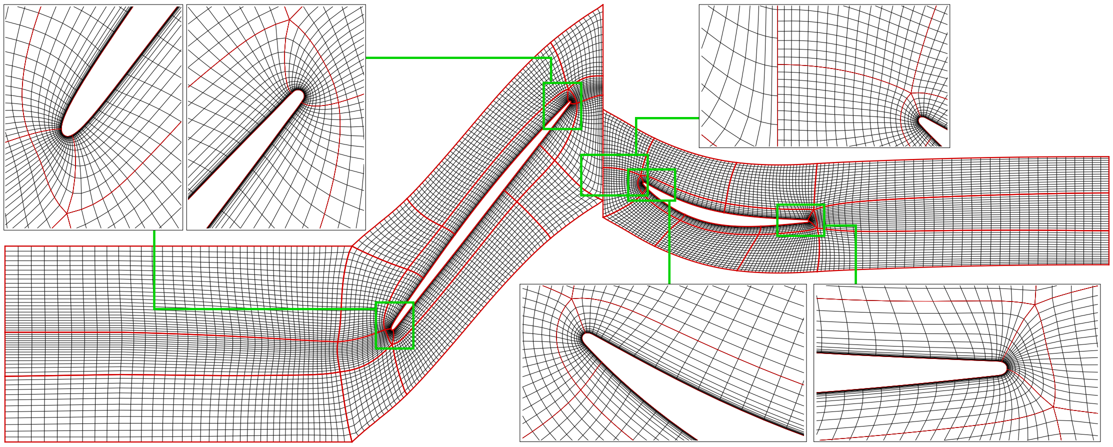

Figure 6.52: Computational grid for the 55% span section of Darmstadt Rotor No. 1.

In the following, the two-dimensional section slice of the transonic compressor is redesigned while considering a fully-turbulent viscous flow environment. Figure 6.53 depicts the relative Mach number contours for the baseline design and illustrates that the flow field in the rotor domain is highly transonic. The inlet Mach number is supersonic, $M_{\mathrm{rel,in}} = 1.13$, and a detached bow shock forms upstream of the rotor leading edge. The supersonic flow region extends until the rotor suction side and is terminated by a strong normal shock located at approximately 50% chord just in front of the rotor passage entry. The isentropic efficiency obtained from the numerical solution is 90.92% and the total pressure ratio of the baseline design is 1.495. The objective of this study is again to increase the isentropic efficiency by reducing the

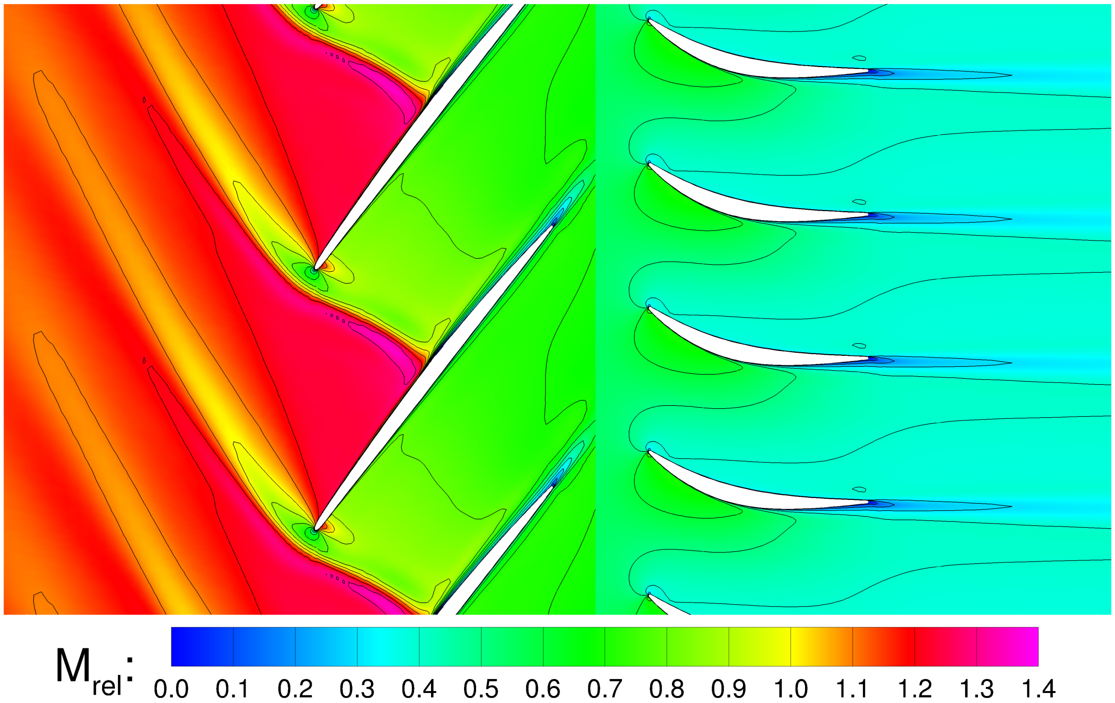

Figure 6.53: Relative Mach number contours, Darmstadt Rotor No. 1, 55% span section, baseline design, $\eta_{\text{is}} = 90.92\%$.

entropy generation throughout the single-stage compressor. Similarly as for the Configuration D compressor, two optimization cases are investigated; an unconstrained optimization and a constrained design case, in which the efficiency of the Darmstadt Rotor No. 1 is maximized while constraining the mass flow rate ($\pm 0.1\%$) and the total pressure ratio ($\pm 0.02\%$). The surface grid points are utilized as design variables; however, the original blade radii are preserved by assigning geometrical constraints to the rotor and stator leading and trailing edge points. Furthermore, no specific maximum thickness constraints are prescribed to the blades. To speed up the convergence process, at each design cycle, the computational domain is initialized with the previous flow solution and, then, 1500 multigrid cycles are performed to evaluate the flow field. Depending on the airfoil shapes, the residual typically drops by eight to nine orders. As shown in the convergence studies presented in subsections 6.2.3 and 6.3.3, it is generally sufficient to converge the flow equations by four to five orders; however, to increase the robustness of the optimization and to ensure an accurate adjoint solution as well as gradient calculation, the number of iterations is chosen to be higher than usually necessary. The adjoint system is initialized accordingly and is converged to a similar level as the flow equations.

Figure 6.54 illustrates the adjoint solution $\boldsymbol{\psi}_1$, i.e. the co-state variable corresponding to the continuity equation, for the entropy generation rate at the first design iteration. The corresponding gradient distribution is depicted in Figure 6.55, where the design variables within the red dashed boxes are the geometrically constrained leading edge (LE) and trailing edge (TE) points. Comparing both the rotor and stator gradients, one observes that the largest gradient amplitudes occur on the rotor suction side (SS) near the location of the shock. This reconfirms that the rotor suction side geometry is highly sensitive to the design objective, while, as indicated by the small gradient amplitudes associated with the stator design points, the stator geometry only has a minor influence on the objective functional. These observations confirm the conclusions that can be drawn from the flow solution, cf. Figure 6.53, which suggests that the entropy production in the stator domain is negligible compared to the losses and entropy rise generated by the rotor shock.

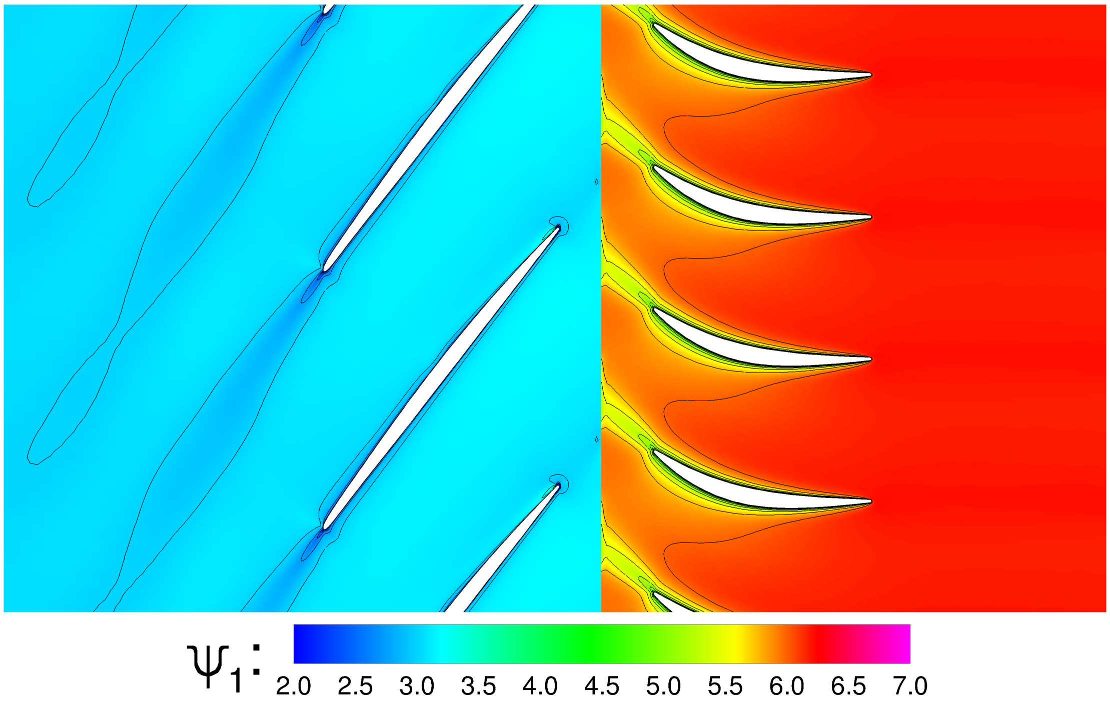

Figure 6.54: $\psi_1$-contours, Darmstadt Rotor No. 1, 55% span section, baseline design, objective function: entropy generation rate.

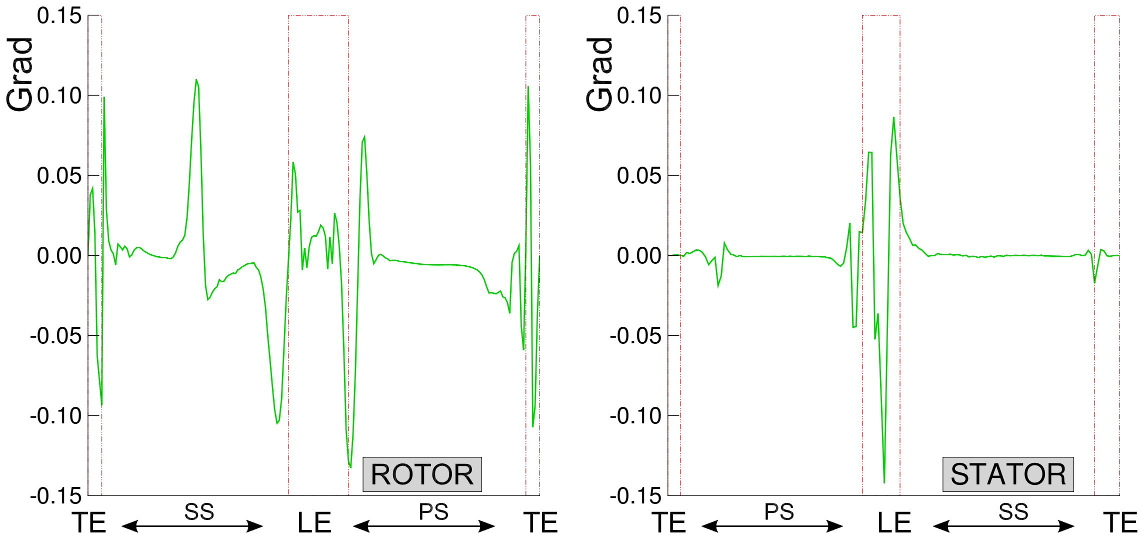

Figure 6.55: Gradient for rotor (left) and stator (right), Darmstadt Rotor No. 1, 55% span section, baseline design, objective function: entropy generation rate.

The unconstrained optimization requires nine design cycles to reach an optimum design (27 function evaluations including all additional line searches), whereas the constrained optimization converges to the final solution within 22 design cycles (overall cost: 69 flow evaluations and 207 adjoint calculations). The final blade geometries of both optimization cases are compared with the baseline design in Figure 6.56. Throughout the unconstrained optimization, the stator geometry remains nearly unchanged. Likewise, the rotor pressure side is modified only slightly. As suggested by the gradient plots, the largest profile modifications occur on the rotor suction side; the profile curvature is reduced in the supersonic flow region but increased in the subsonic flow region after the shock. Similarly as in the unconstrained optimization, in the constrained design case the shape modifications of the rotor geometry exceed those of the stator geometry. However, in the constrained optimization the stator shape is modified and both the suction and pressure sides are pushed downwards in the aft part of the airfoil. Furthermore, the rotor blade remains thicker in the supersonic flow region and the pressure side is now adjusted noticeably, especially towards the rotor trailing edge. Moreover, it can be observed that the maximum thickness of the optimized rotor blades is located at different chord lengths. In the constrained design case the maximum rotor thickness is reached further upstream.

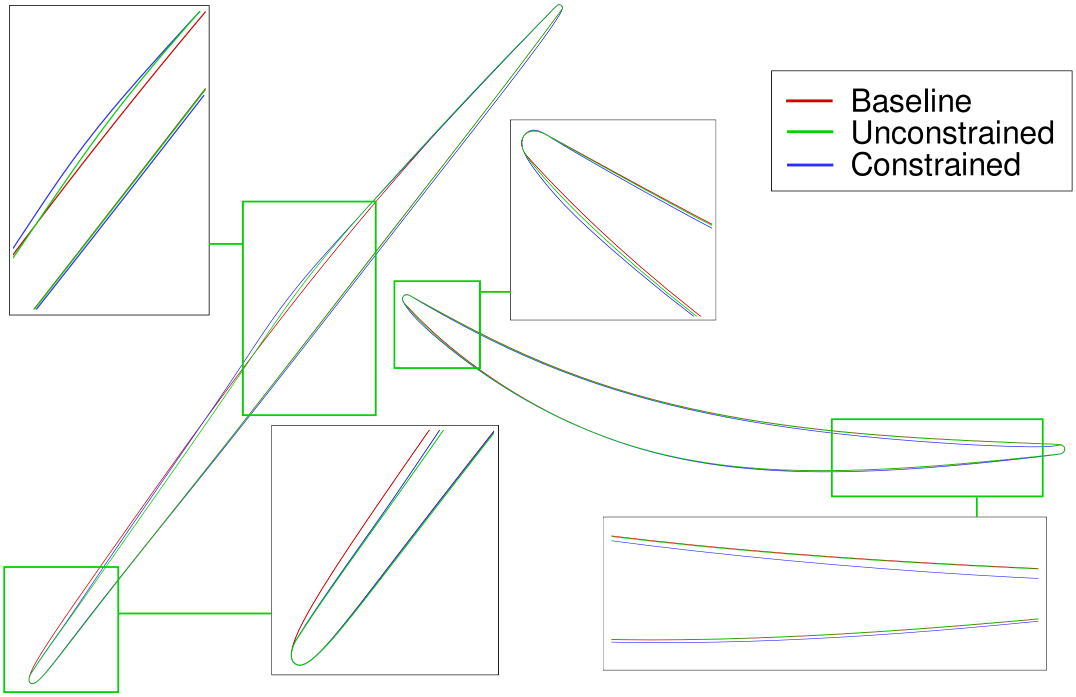

Figure 6.56: Shape modification Darmstadt Rotor No. 1, 55% span section.

As discussed further below, these additional shape modifications can be attributed to the constraints and their gradients, which control the mass flow rate as well as the total pressure ratio and ensure that the constraints stay within their prescribed bounds.

The shape modifications of both the unconstrained and constrained optimization cases lead to a reduced flow acceleration in the supersonic flow region and yield optimized designs which produce a weaker and more oblique shock than the baseline geometry, cf. Figures 6.57 (left) and 6.58 (left). The entropy contour plots 6.57 (right) and 6.58 (right), where $\delta s = s_0 - s_{\text{opt}}$ and hence $\delta s > 0$ indicates a lower entropy level in the redesigns, show that the shock structures of the optimized designs generate less entropy than the normal shock occurring in the baseline design. Comparing both optimized designs with each other, the entropy plots illustrate as well that the unconstrained optimization reduces the entropy generation further; especially in the boundary-layer downstream of the rotor shock, where the entropy level is lower in the absence of constraints. In the constrained design case, the mass flow and total pressure ratio constraints prevent the realization of profile modifications which would

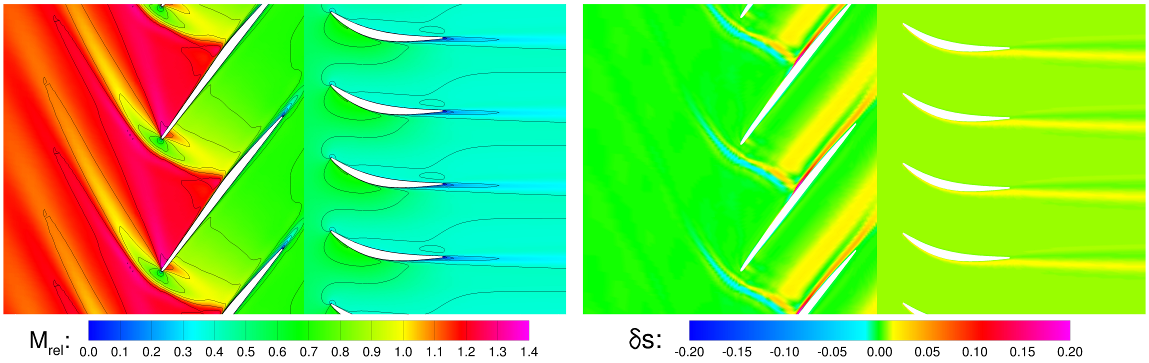

Figure 6.57: Relative Mach number contours (left) and difference in entropy (right), Darmstadt Rotor No. 1, 55% span section, unconstrained optimization, $\eta_{\text{is}} = 92.86\%$.

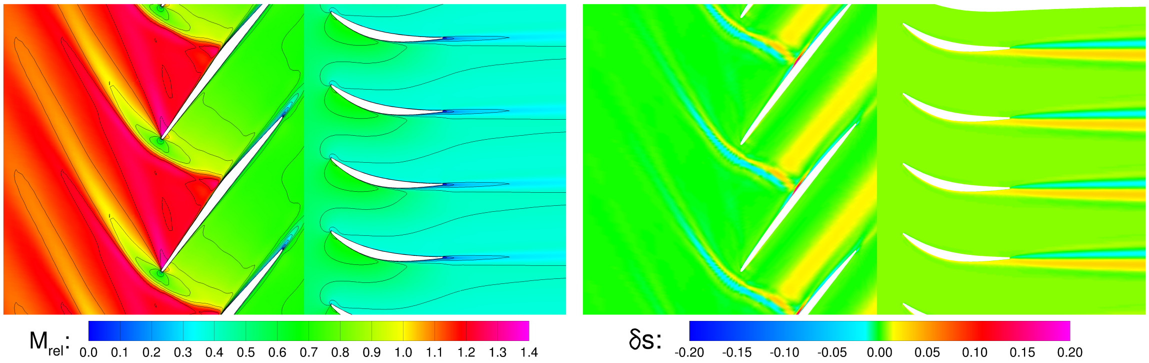

Figure 6.58: Relative Mach number contours (left) and difference in entropy (right), Darmstadt Rotor No. 1, 55% span section, constrained optimization $\eta_{\text{is}} = 92.46\%$.

enable a further reduction of the shock strength to the level obtained from the unconstrained optimization. The stronger shock/boundary-layer interaction generates higher losses, which manifest themselves in the higher entropy level observed within the boundary-layer. At the same time, in the unconstrained optimization the rotor shape modifications only aim to reduce the entropy generation. However, as a side effect they also cause an increase in the mass flow rate and the total pressure ratio, cf. Figure 6.59 (right) further below. In the constrained design case the rotor suction side is modified to reduce the shock strength as well, but also such that the constraints stay within their prescribed bounds. The aforementioned upstream-shift of the maximum rotor thickness and the profile modifications on the rotor pressure side control the mass flux and ensure that it is maintained throughout the optimization.

Figure 6.59 (left), which displays the convergence of the objective function throughout the optimization, confirms these observations. During the unconstrained optimization, the overall entropy generation rate drops by 22.5%, while the constrained design case reduces the objective functional by 18.4%. Accordingly, the isentropic efficiency increases to 92.86% (+1.94%) for the unconstrained optimization and to 92.46% (+1.54%) for the constrained design case. Thus, out of both optimization cases, the unconstrained redesign yields a slightly better efficiency (+0.40%); however, at the same time both the mass flow rate and the total pressure ratio increase by 1.48% and 0.26% respectively and, thus, experience changes that exceed the prescribed bounds of the constrained optimization by severalfold. Figure 6.59 (right) displays the changes in the mass flow rate and the total pressure ratio throughout the redesign; the dashed red lines define the bounds on the constraints. For the constrained optimization, the constraints stay within their prescribed bounds at each design cycle; in case a constraint is violated, the solution is not accepted and a new line search is performed. The final optimized design shows a mass flow rate increase of only 0.03% and a negligible total pressure ratio deficit of 0.02% compared to the

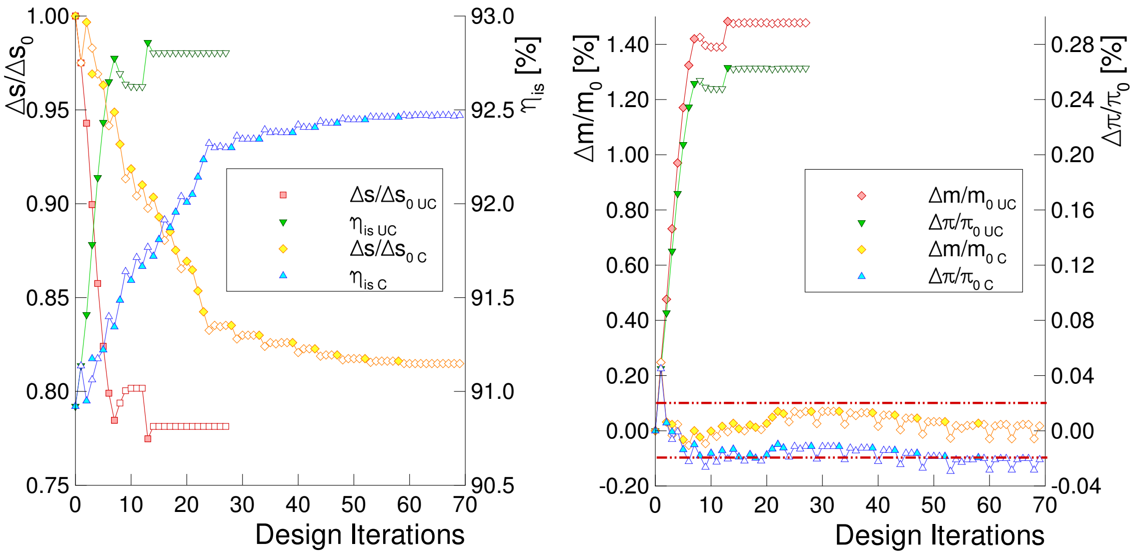

Figure 6.59: Normalized objective function and isentropic efficiency (left), change in mass flow rate and total pressure ratio (right), constrained optimization, Darmstadt Rotor No. 1, 55% span section.

baseline design. The constrained optimization terminates after the 22$^{\text{nd}}$ design cycle, where the total pressure ratio constraint prevents the numerical program from finding a more efficient compressor design. Table 6.16 summarizes the aerodynamic performance parameters of the single-stage transonic compressor obtained for the constrained optimization case and compares them with the values from the baseline design and the unconstrained optimization.

| | $\eta_{\text{is}}$ [%] | $\pi$ | $\frac{\dot{m}_{\text{opt}}}{\dot{m}_0}$ |
|---|---|---|---|
| Baseline | 90.92 | 1.49491 | 1.0000 |
| Unconstrained | 92.86 | 1.49884 | 1.0148 |
| Constrained | 92.46 | 1.49462 | 1.0003 |

Table 6.16: Performance comparison between the original and optimized compressor designs, Darmstadt Rotor No. 1, 55% span section.

### 6.3.6 Optimization in Three-Dimensional Inviscid Flow

After having discussed two two-dimensional design cases, in this subsection a three-dimensional optimization is presented in which the entire Darmstadt Rotor No. 1 is redesigned while considering an inviscid flow field. The objective of the optimization remains the same, i.e. to increase the stage efficiency by minimizing the entropy generation throughout the single-stage compressor. The baseline design, cf. Figure 6.60 (left) which shows the flow solution at midspan, is operated at a total pressure ratio of 1.63 and the isentropic efficiency obtained from the numerical solution is 88.78%. The difference between the numerical total pressure ratio and the experimental value, cf. subsection 6.3.2, is mainly due to the fact that in the current CFD simulation both the viscous effects as well as the rotor gab are neglected. While an initial unconstrained optimization of the transonic compressor stage results in a remarkable efficiency gain, it also shows that other aerodynamic performance measures, including the mass flow rate and the total pressure ratio, change during the design process. Consequently, a second constrained optimization is performed, in which the compressor efficiency is maximized while restricting the change in the mass flow rate to

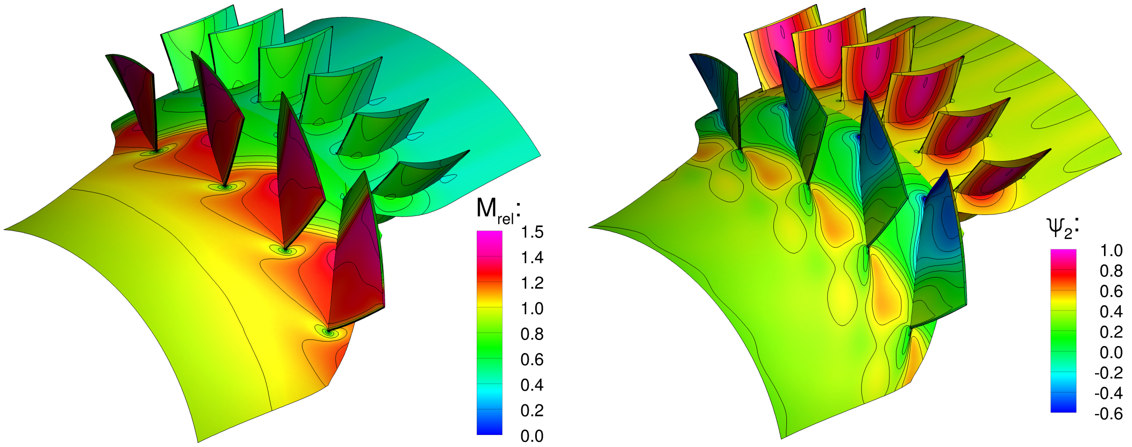

Figure 6.60: Relative Mach number contours (left) and adjoint $\psi_2$-contours (right) for Darmstadt Rotor No. 1, inviscid design case, plotted section slice: 50% span, functional: entropy generation rate.

±0.25% and the variation of the total pressure ratio to ±0.10%.

In the optimization case presented below, 81 Hicks-Henne bump functions are evenly distributed on each suction and pressure side; thus, overall 324 design variables are utilized to control the blade shapes. Nine radial sections are defined between the hub and tip of each blade (from 10% span to 90% span) and at every radial section nine Hicks-Henne bump functions are placed between the trailing edge and leading edge (from 10% to 90% of the chord length). No specific geometrical constraints are prescribed to the rotor and stator blades. However, the selected locations of the bump functions ensure that the trailing and leading edge radii as well as the near hub and tip sections of the blades are maintained throughout the optimization. Furthermore, the hub geometry and the compressor casing are not modified during the redesign. To reduce the number of iterations necessary to sufficiently converge the flow equations and the adjoint system, at each design cycle the computational domain is initialized with the flow and adjoint solutions from the previous design iteration. Following this procedure, only 1200 multigrid cycles are required to reduce the density residual and to converge the adjoint equations by six to eight orders.

A physical interpretation of the adjoint equation helps understand the adjoint solution depicted in Figure 6.60 (right) and ultimately the geometrical changes occurring during the redesign. From Eqn. (4.5) it can be observed that the product of the adjoint co-state vector $\boldsymbol{\psi}$ and the variation of the flow residual $[\partial \mathbf{R}/\partial \mathbf{W}]^T$

determines the change to the objective function. On the discrete level, the adjoint solution in the computational cell $(i, j, k)$ represents a vector counterpart to the flow solution in that particular cell. For instance, as stated by Shankaran [92], the adjoint co-state $\psi_1$ corresponds to the density and can be interpreted as the change required to be induced to the mass flux to cause a change in the objective function of interest. Similarly, the adjoint co-states $\psi_2$ to $\psi_4$ represent the counterparts to the momentum quantities $\rho u_1$, $\rho u_2$, and $\rho u_3$ and signify the change required in the momentum flux to change the objective function. In particular, a positive adjoint solution $\psi_1$ (or $\psi_2$ to $\psi_4$ or $\psi_5$) indicates that the density flux (or momentum flux or energy flux) should be increased to meet the design objective. Accordingly, a negative co-state solution suggests a decrease in the corresponding primal flux. The adjoint solution for the entropy generation rate functional, cf. Figure 6.60 (right), shows that the sign of the co-state variable $\psi_2$ changes on the rotor suction side from a positive value in the supersonic flow region to a negative value in the subsonic flow region. Hence, the adjoint solution indicates that an increase in the $x_1$-momentum flux in the forward part and a decrease in the $x_1$-momentum flux in the aft part of the rotor suction side will improve the design objective. Combining this information with the gradient trends depicted in Figure 6.61, which illustrates the rotor and stator gradients at 70% span, confirms that the objective function is particularly sensitive to geometry variations on the rotor suction side and suggests that a reduced profile curvature in the front part of the rotor suction side will reduce the entropy generation. From the gradient distributions it further can be observed that, similar to the two-dimensional design cases, the gradient amplitudes of the rotor design points are larger than those of the stator design points. The highest gradient values occur again in the near-shock region on the rotor suction side, which attests that the rotor geometry has more influence on the design objective than the stator shape. This is obvious since in the stator domain the entropy generation is due to dissipative effects only. In fact, due to the assumption of an inviscid flow environment and the fact that no shock losses occur in the stator domain, the entropy increase within the stator row is a result of the artificial dissipation introduced by the numerical scheme. The rotor shock is responsible for most of the entropy produced within the compressor stage.

The final optimized compressor stage is obtained after twelve design cycles; the

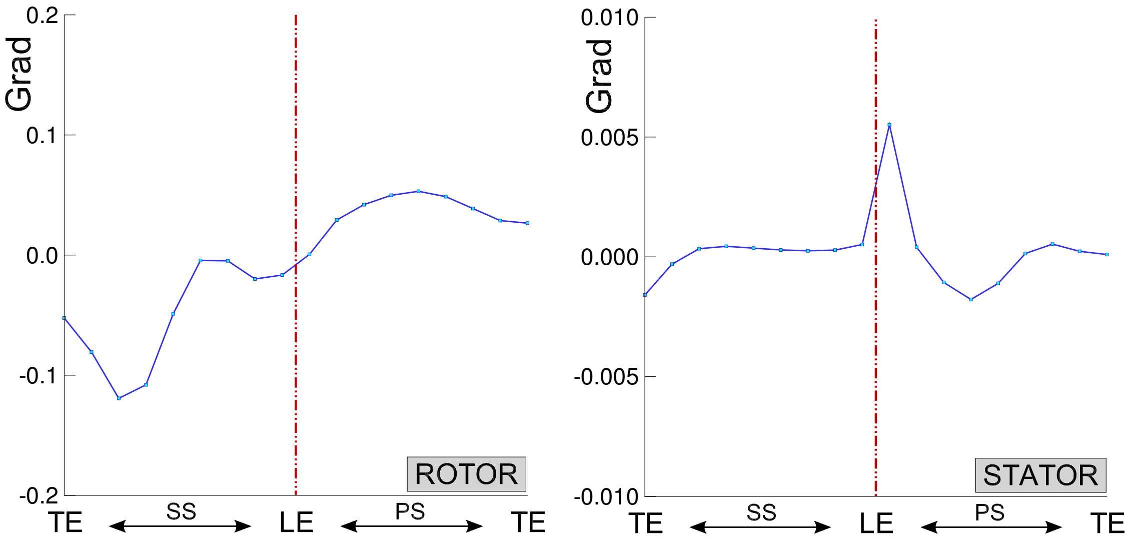

Figure 6.61: Rotor (left) and stator (right) gradient at 70% span, Darmstadt Rotor No. 1, inviscid design case, baseline design, functional: entropy generation rate.

overall cost of the constrained optimization, which includes additional line searches, amounts to fifteen flow and 45 adjoint evaluations. The optimization terminates when the optimizer is unable to find a solution that improves the objective function without violating the total pressure ratio constraint. Figure 6.62 compares the geometry of the redesigned blades with the baseline geometry at 25% span, 50% span, and 75%

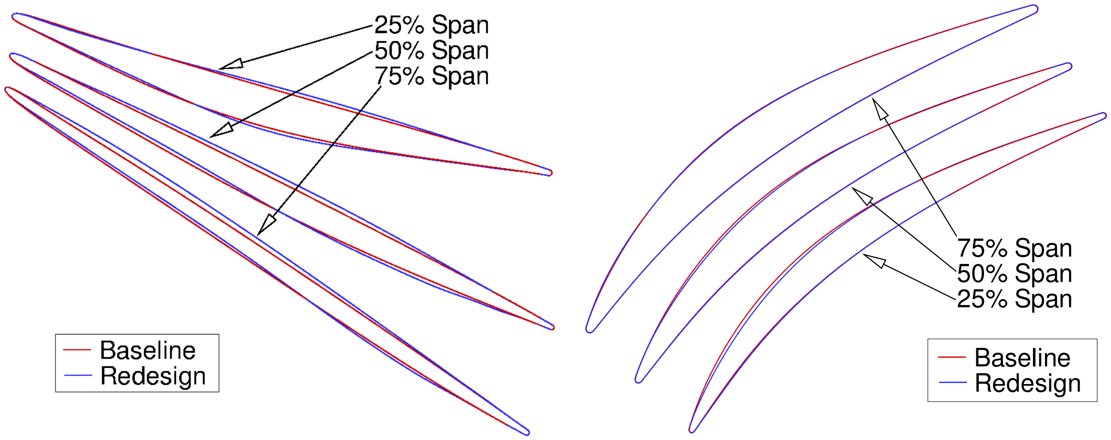

Figure 6.62: Shape modification of rotor (left) and stator (right), Darmstadt Rotor No. 1, inviscid design case, constrained optimization.

span. For clarity the radial sections have been displaced. As indicated by the gradient distributions, the stator geometry undergoes only small modifications. In the front part the suction side curvature is slightly reduced, especially towards the hub region. The rotor blade experiences larger profile changes, both the suction and pressure sides are modified noticeably. As expected, the suction side profile curvature is reduced in the supersonic flow region and increased in the subsonic flow region after the shock. The rotor geometry modifications grow larger towards the tip region, which is a result of the increasing shock strength in the radial direction. In the redesign, the flow acceleration in the supersonic region is slowed down again, leading to a weaker shock, cf. Figure 6.63. Furthermore, as illustrated in Figure 6.63 (left), in the baseline design the supersonic flow region extends into the rotor passage and at both the 50% span and 75% span sections the shock stretches from the rotor suction side to the pressure side of the neighbouring blade. The sections are chocked. In the redesign, the shock location is pushed upstream and out of the rotor passage; the flow within the rotor passage remains subsonic, cf. Figure 6.63 (right). Figure 6.64, which depicts the relative Mach number distribution on the rotor blade, confirms these observations and shows that the flow on the rotor pressure side remains subsonic over the entire span.

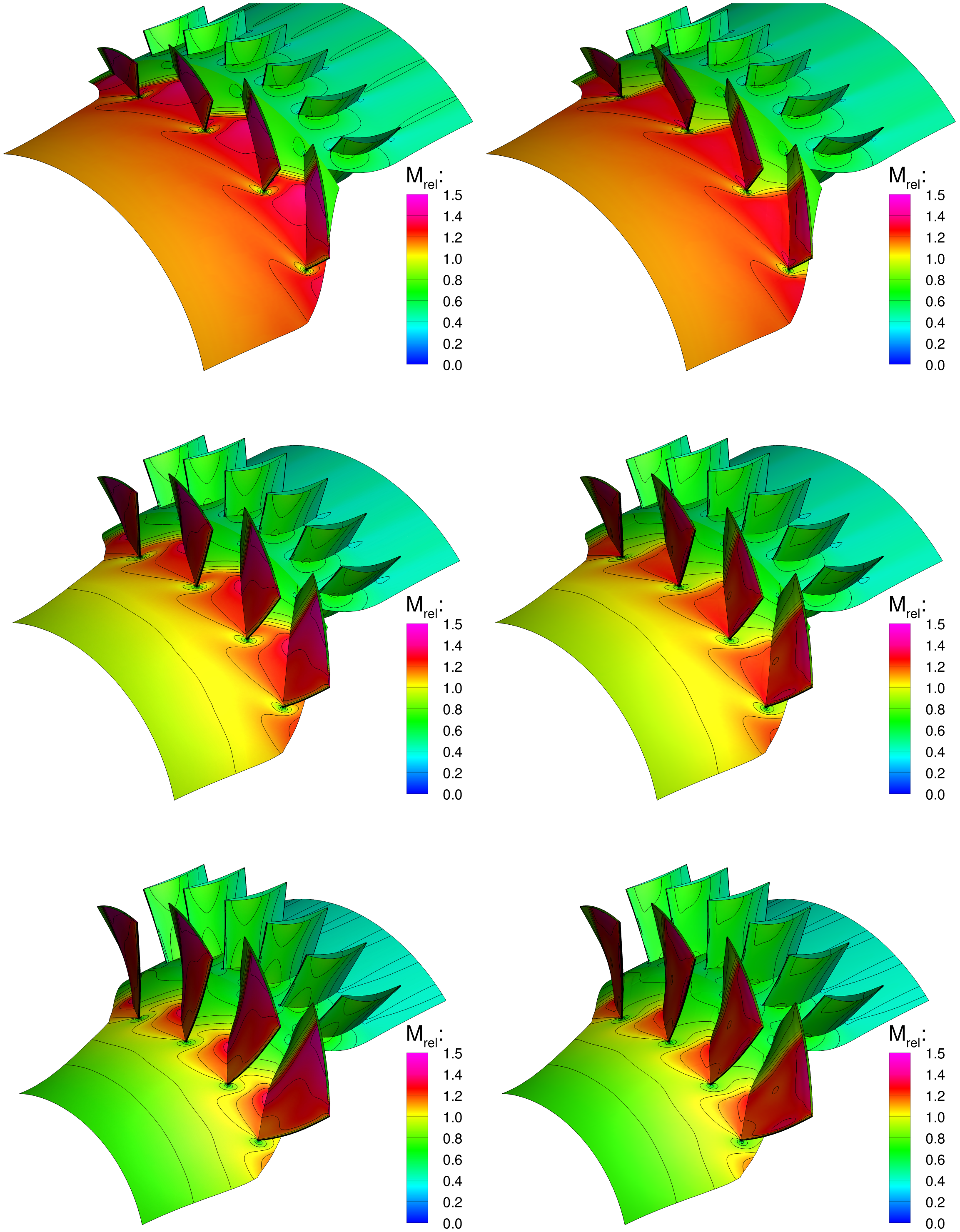

Figure 6.63: Relative Mach number contours of Darmstadt Rotor No. 1 baseline design (left) and redesign (right) at 75% span (top), 50% span (center), and 25% span (bottom), inviscid design case, constrained optimization.

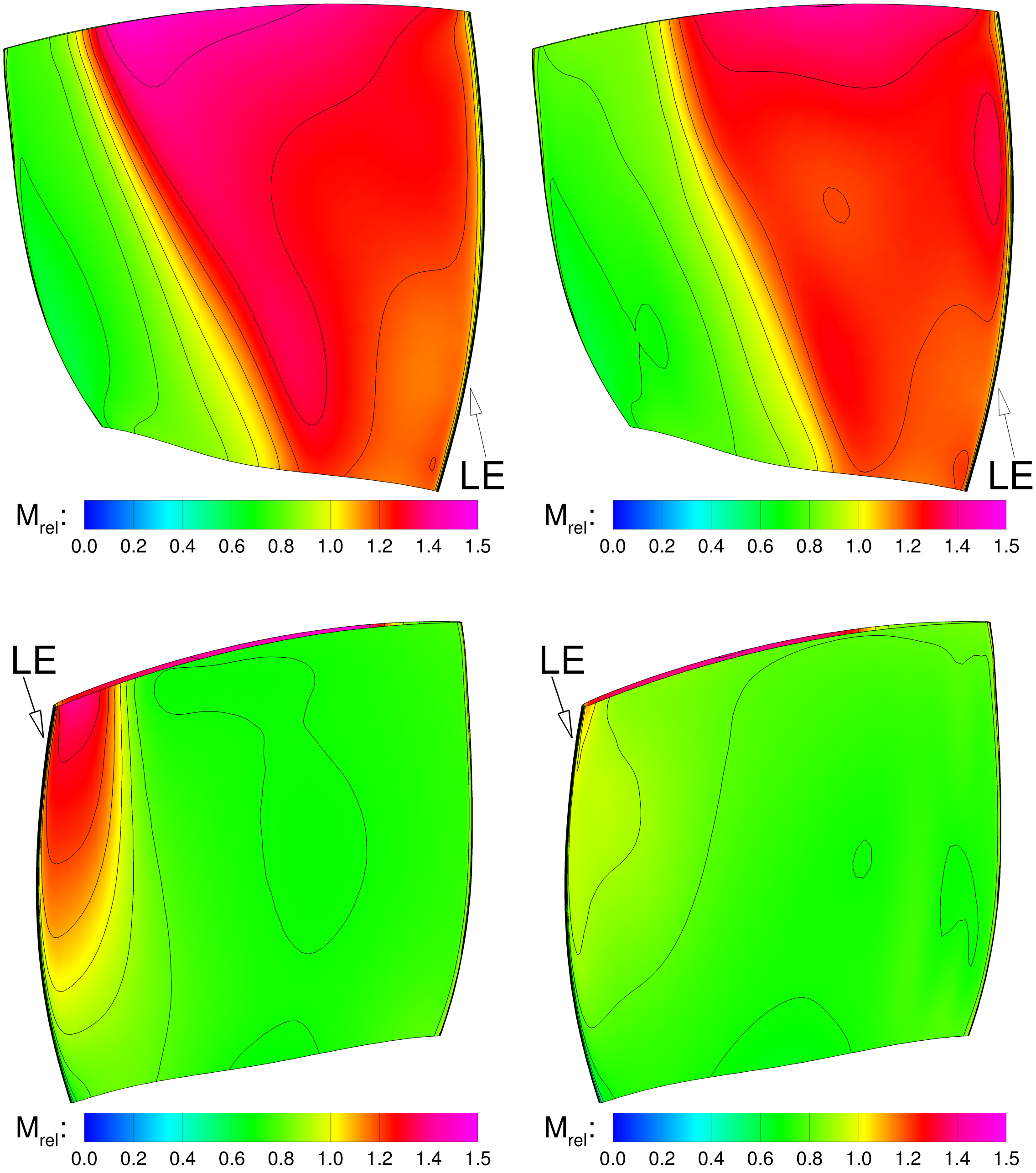

Figure 6.64: Relative Mach number distribution on the rotor suction side (top) and rotor pressure side (bottom) of the baseline design (left) and redesign (right), inviscid design case, constrained optimization.

The weaker rotor shock in the redesign leads to a smaller entropy rise across the shock and within the entire compressor stage. Throughout the constrained optimization, the overall entropy generation rate drops by 19.9% and the isentropic efficiency increases by 2.02%, cf. Figure 6.65 (left). Figure 6.65 (left) also depicts the convergence of the objective function and the increase of the isentropic efficiency for the aforementioned unconstrained optimization case. In the absence of constraints, the final design is reached after 22 design cycles (23 flow and adjoint evaluations including additional line searches), which provides an efficiency increase of 2.61%. However, at the same time both the mass flow rate and the total pressure ratio increase by 1.55% and 0.27% respectively and, thus, exceed the bounds prescribed on the constrained optimization. Figure 6.65 (right) monitors the changes of the constraints throughout the redesign, where the dashed red lines define the bounds on the constraints. In the case of the constrained optimization, a design proposed by the optimizer is automatically rejected if it does not improve the design objective or violates a constraint (design iterations marked by uncoloured symbols represent such unsuccessful line searches) and new step lengths are tested for the given search direction until a feasible solution is found. After the fifteenth design iteration, the optimizer cannot find an improved design without surpassing the lower bound on the total pressure ratio and consequently the optimization terminates. The final optimized design shows

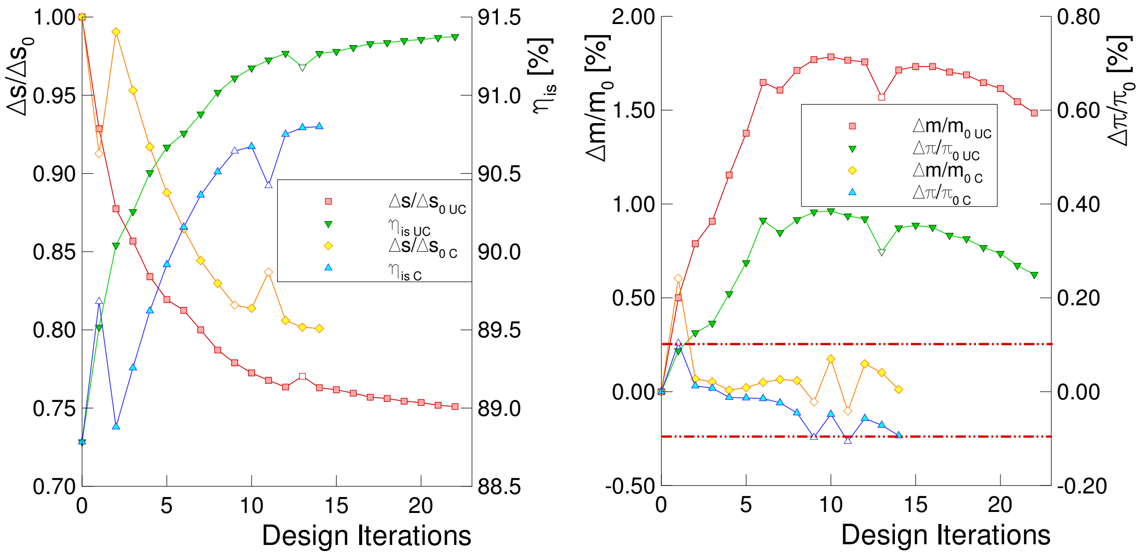

Figure 6.65: Normalized objective function and isentropic efficiency (left), change in mass flow rate and total pressure ratio (right), Darmstadt Rotor No. 1, inviscid design case.

| | $\eta_{\mathrm{is}}$ [%] | $\pi$ | $\frac{\dot{m}_{\mathrm{opt}}}{\dot{m}_0}$ |
|---|---|---|---|
| Baseline | 88.78 | 1.63190 | 1.0000 |
| Unconstrained | 91.39 | 1.63659 | 1.0155 |
| Constrained | 90.80 | 1.63038 | 1.0001 |

Table 6.17: Performance comparison between the original and optimized compressor designs, Darmstadt Rotor No. 1, inviscid design case.

a total pressure ratio deficit of only 0.09% and a negligible change in the mass flow rate of 0.01% compared to the baseline design.

The results of the optimization of the Darmstadt Rotor No. 1 in an inviscid flow field are summarized in Table 6.17. Despite the relative strict constraints on the total pressure ratio and the mass flow rate, the constrained optimization case leads to considerable efficiency improvements and, at the same time, ensures that the constrained aerodynamic parameters do not change during the optimization process.

### 6.3.7 Optimization in Three-Dimensional Viscous Flow

In this last optimization case, the entire single-stage transonic compressor is redesigned while considering a fully-turbulent viscous flow environment. The design objective is again to increase the stage efficiency by minimizing the entropy generation rate; however, this time no additional constraints are prescribed. Figure 6.66 depicts the relative Mach number contours for the baseline design at 50% span and displays the isentropic Mach number distribution, cf. Eqn. (2.52), on the blade surfaces. The isentropic efficiency obtained from the numerical solution is 89.49% and the flow solver predicts a total pressure ratio of 1.50. The stage pressure ratio is in good agreement with the numerical and experimental results provided by Fritsch et al. [90] and Höger et al. [91] for the compressor design point ($\pi_{\mathrm{exp}} = 1.48$ and $\pi_{\mathrm{num}} = 1.49$). The isentropic efficiency ($\eta_{\mathrm{is,exp}} = 86.7\%$ and $\eta_{\mathrm{is,num}} = 88.3\%$) is overpredicted, which was to be expected since in the presented research work the rotor gab is not modelled.

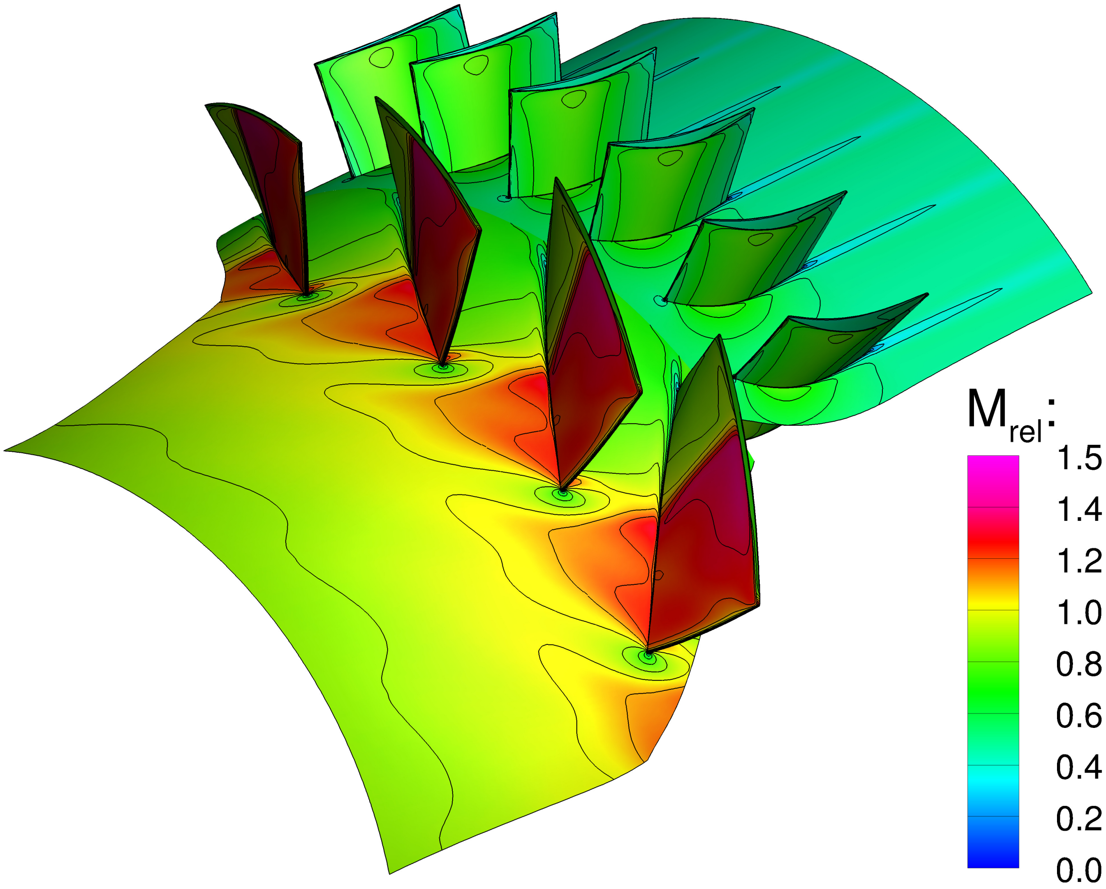

Figure 6.66: Relative Mach number contours, Darmstadt Rotor No. 1, viscous design case, plotted section slice: 50% span, baseline design, $\eta_{\mathrm{is}} = 89.49\%$.

Again, Hicks-Henne functions are utilized as design variables and 81 bumps are equidistantly distributed on each suction and pressure side. The unconstrained optimization terminates after eighteen design iterations, which includes one unsuccessful line search. The rotor geometry experiences shape modifications which are generally similar to those observed in the inviscid design case, cf. Figure 6.67. Since the main reason for the entropy increase within the rotor row is the strong shock and, thus, is due to an inviscid physical phenomenon, this is not surprising. However, while the stator geometry remained nearly unchanged throughout the inviscid optimization, in the present design case the stator blade is modified noticeably. In the viscous design case, the entropy generation in the stator domain is not only caused by the artificial dissipation introduced by the numerical scheme, but also due to the dissipative

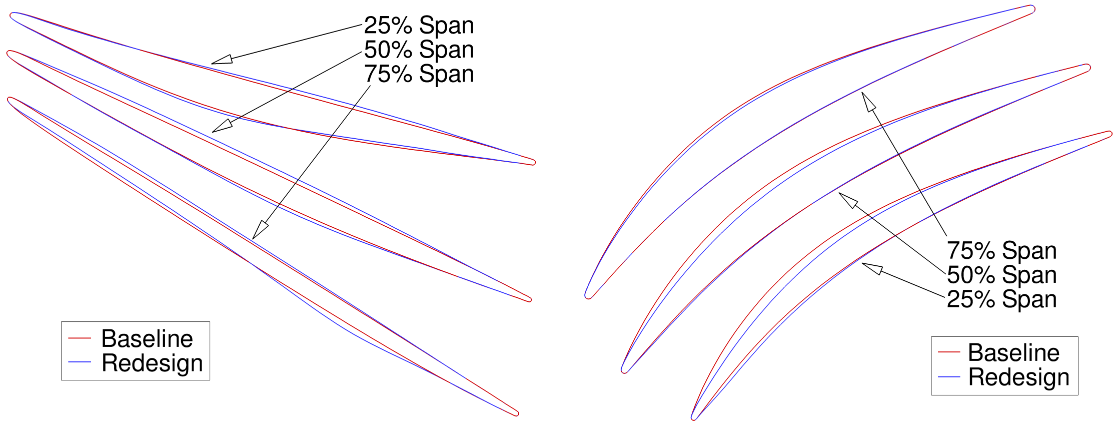

Figure 6.67: Shape modification of rotor (left) and stator (right), Darmstadt Rotor No. 1, viscous design case, constrained optimization.

nature of the Navier-Stokes equations. Viscous effects, such as the stator boundary-layer and the downstream wake, make the stator geometry more sensitive to the design objective. Especially towards the hub region, where the flow turning is the highest in order to maximize the work output of the stage, the optimizer reduces the profile thickness and the suction side curvature of the stator blade significantly. This reduces the flow acceleration in the front part as well as the deceleration in the aft part of the stator suction side and consequently produces a more uniform flow field, which in turn leads to a thinner boundary-layer and reduced entropy generation.

The rotor shape modifications lead to a weaker and more oblique shock, which in turn attenuates the entropy generation across the shock discontinuity and in the downstream boundary-layer. Comparing the flow field of the optimized design with that of the baseline design, cf. Figure 6.68, illustrates that particularly in the near tip region, i.e. in the area where the highest relative flow velocities occur, the shock strength could be reduced considerably. Furthermore, the shock location is pushed upstream and the flow acceleration from the rotor leading edge stagnation point towards the pressure side is reduced. Accordingly, the supersonic flow region which occurs on the rotor pressure side in the vicinity of the leading edge diminishes noticeably throughout the optimization, cf. Figure 6.69, although it is not completely eliminated as in the inviscid design case.

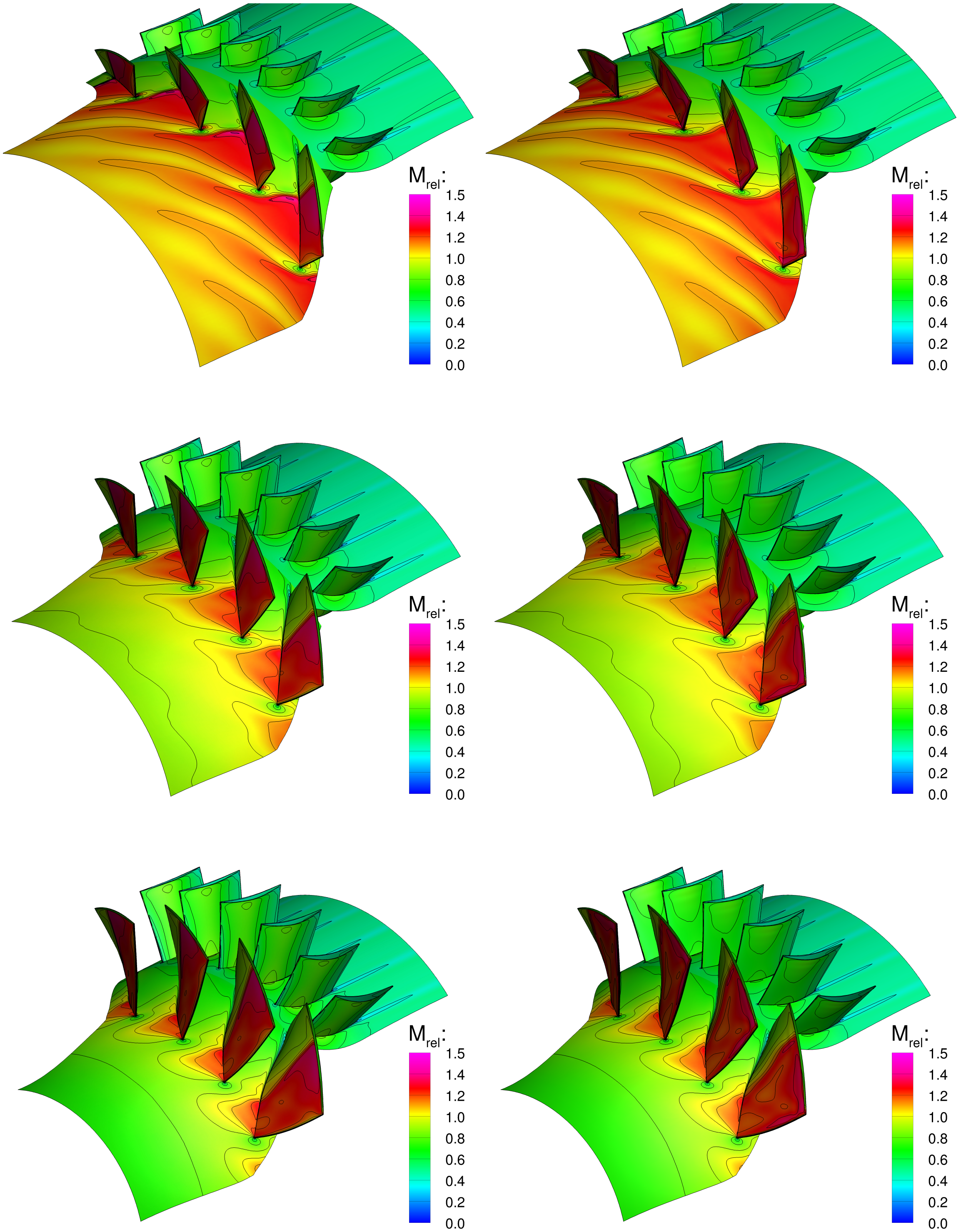

Figure 6.68: Relative Mach number contours of Darmstadt Rotor No. 1 baseline design (left) and redesign (right) at 75% span (top), 50% span (center), and 25% span (bottom), viscous design case, constrained optimization.

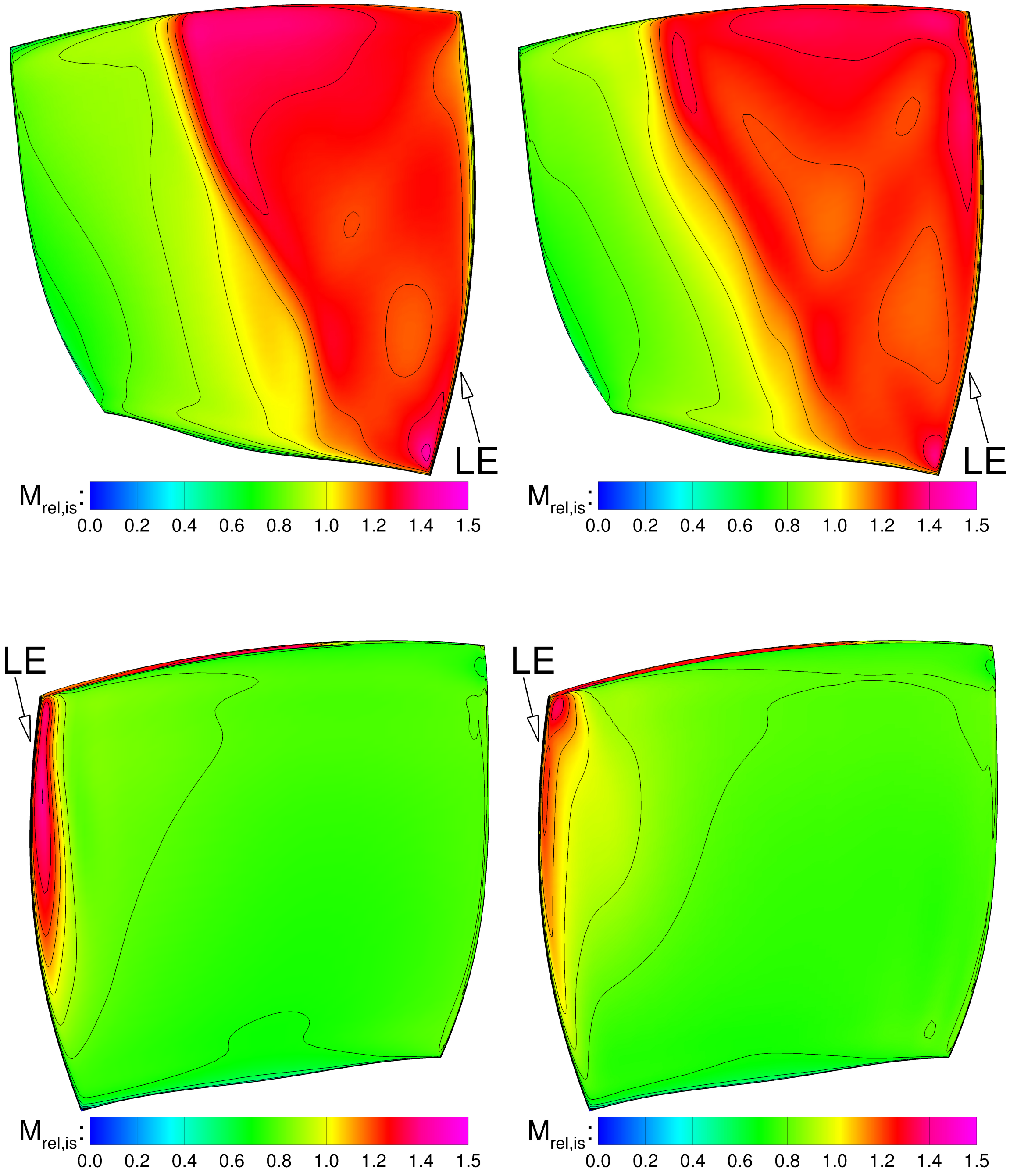

Figure 6.69: Isentropic Mach number distribution on the rotor suction side (top) and rotor pressure side (bottom) of the baseline design (left) and redesign (right), viscous case, constrained optimization.

Figure 6.70 shows the entropy at the exit plane downstream the stator for both the baseline (top) and optimized (bottom) design. The redesign leads to a reduced entropy level in several regions across the outflow plane; the most important ones are highlighted by dashed boxes. Particularly noticeable is the entropy reduction in the stator wake between 20% and 70% span as well at the lower entropy levels close to the hub and tip. Since the hub geometry and the casing are not modified during the optimization, these improvements are due to the changes in the rotor and stator geometry only.

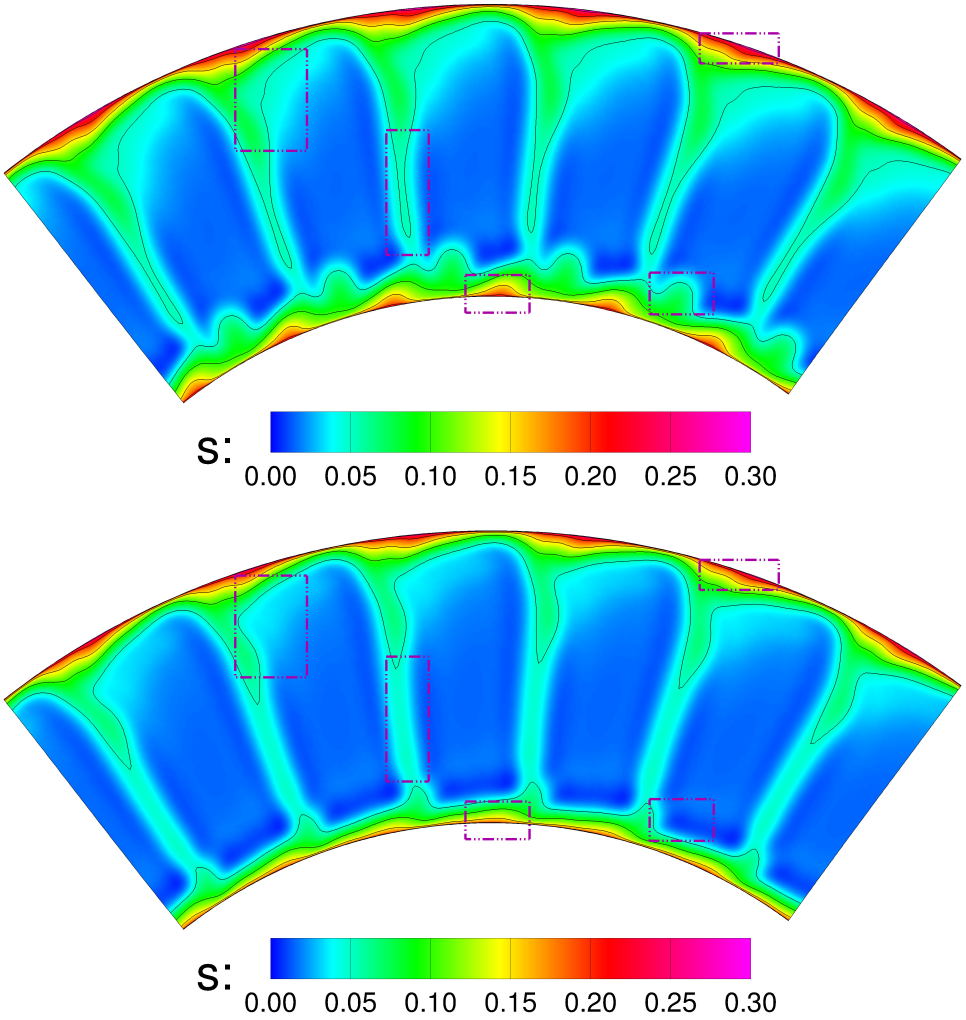

Figure 6.70: Entropy contours baseline design (top) and optimized design (bottom) at the compressor exit plane, Darmstadt Rotor No. 1, viscous design case, constrained optimization.

As depicted in Figure 6.71 (left), the optimization reduces the objective function by 17.2%, which leads to an isentropic efficiency increase by 1.62%. Although no constraints were prescribed on the mass flow rate and total pressure ratio, in this particular design case both measures do not change considerably. Figure 6.71 (right) displays the changes these two aerodynamic parameters experience during the redesign. For illustration purposes, the bounds on the constraints applied in the previous inviscid optimization case are shown as well. It can be observed, that both parameters change considerably less than observed in the inviscid optimization. At every design cycle the mass flow rate stays within ±0.50% of the reference mass flow rate, while the total pressure ratio never exceeds a change of ±0.10%. The final design possesses a total pressure ratio deficit of 0.07% and the mass flow rate differs by 0.13% compared to the baseline design.

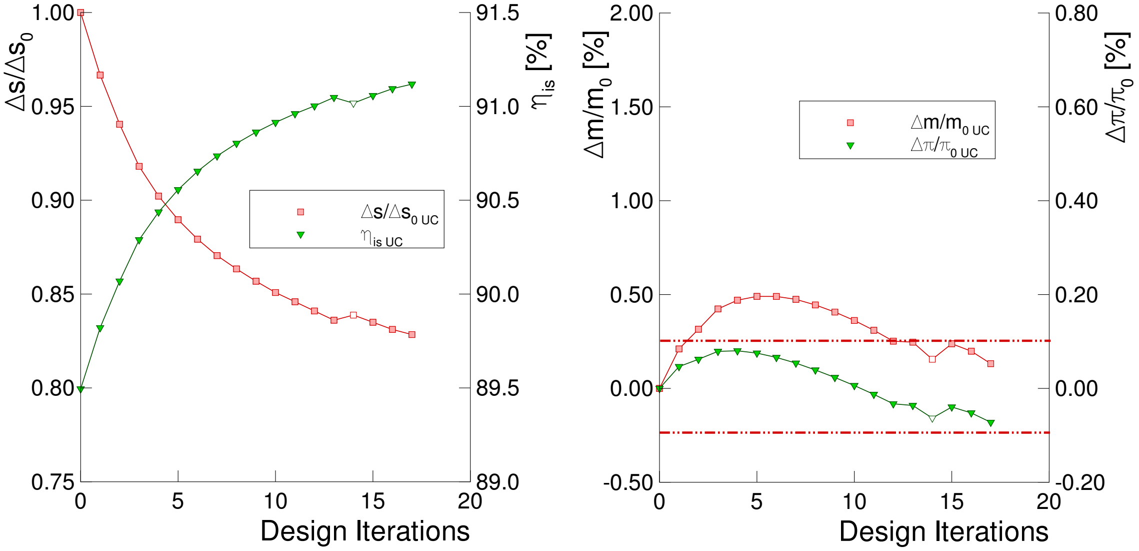

Figure 6.71: Normalized objective function and isentropic efficiency (left), change in mass flow rate and total pressure ratio (right), Darmstadt Rotor No. 1, viscous design case.

Due to the various viscous wall boundaries, which demand a highly resolved grid in the vicinity of the rotor and stator blades as well as near the hub geometry and the casing, the three-dimensional multi-block grid of this last design case poses the most demanding challenges on the RBF-based grid deformation scheme utilized to modify the computational grid throughout the optimization such that it conforms to the continuously deformed blade geometries. Nevertheless, as illustrated in Figure 6.72, the grid for the final optimized compressor configuration, which is obtained after seventeen grid deformations, shows no noticeable quality deteriorations compared to the grid of the baseline design. Figure 6.73 depicts a cut through the final multiblock configuration extracted at 75% span and compares its skeleton (color: red) with the skeleton of the baseline grid (color: blue). The geometry modifications and the resulting deformations of the skeleton are clearly visible. The corresponding grid including the perturbed interior grid points for the optimized design is illustrated in Figure 6.74. The enlargements demonstrate that the grid quality remains remarkably high, even within the boundary-layer where the grid is highly stretched. The RBF scheme propagates the geometry deformation into the domain, where the calculated

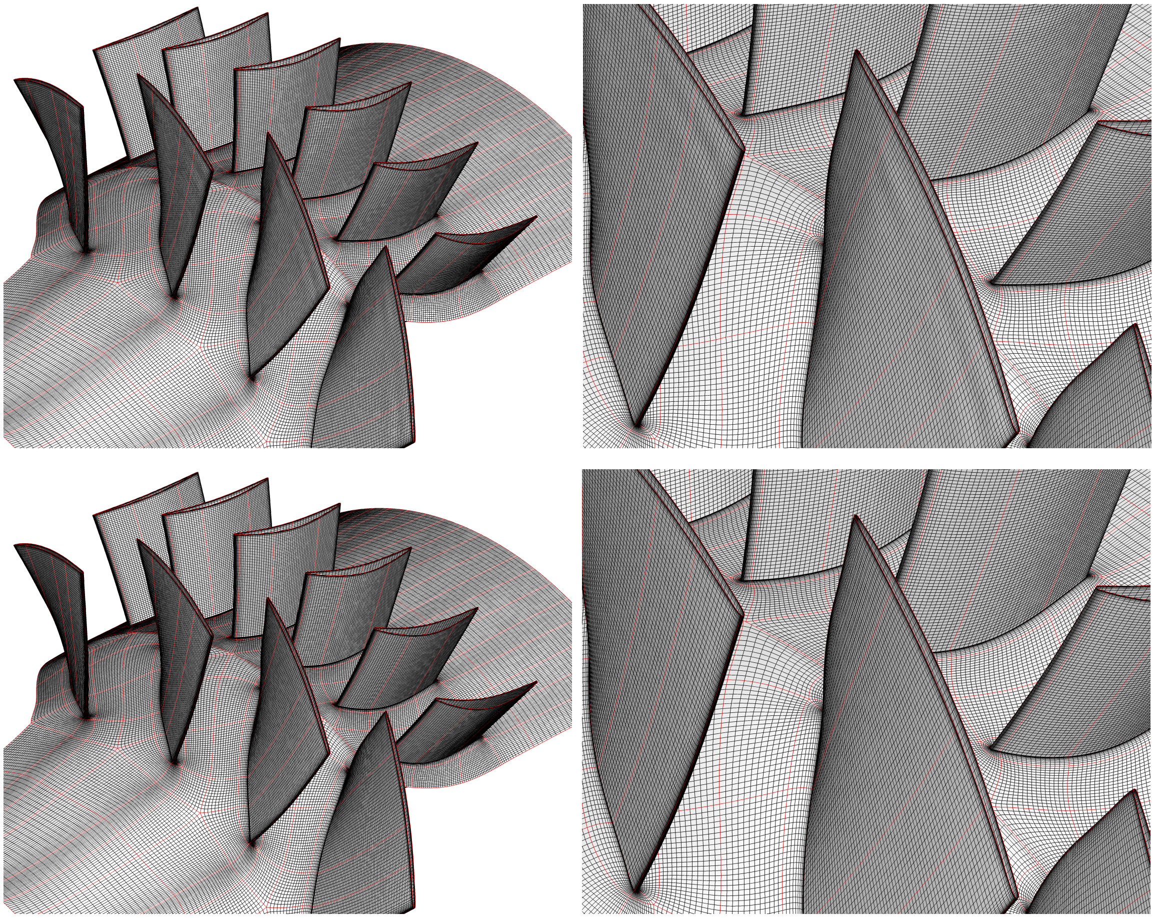

Figure 6.72: Final grid after optimization (top) and baseline grid before optimization (bottom), 33% span cut, Darmstadt Rotor No. 1, viscous design case.

grid point modifications slowly attenuate and eventually tend to zero well before the outer boundaries of the computational domain are reached, cf. subsection 5.1.2 for more details. Furthermore, the interior grid point adjustments are calculated such that the grid remains smooth and quality parameters such as the aspect ratio, orthogonality, expansion factor, and skewness are maintained. Thus, the developed RBF-based grid perturbation scheme ensures that the grid quality remains high, even after a large number of consecutive grid perturbations for complex three-dimensional viscous multi-block grid configurations. Moreover, its novel two-stage approach, in which first the skeleton is deformed on the root processor and then the remaining interior grid points are adjusted in parallel, as well as the various measures implemented to speed up and reduce the memory footprint of the algorithm enable the calculation of high-quality grid deformations in a timely manner. In this particular case investigated here, 48 cores are used at the second stage of the deformation scheme to move the interior grid points; the time of one grid perturbation amounts to approximately 0.15% of the overall cost of one design cycle (which here includes one flow and adjoint solution, the gradient calculation, and the final grid perturbation) or to approximately 0.5% of the time required to solve the flow governing equations alone.

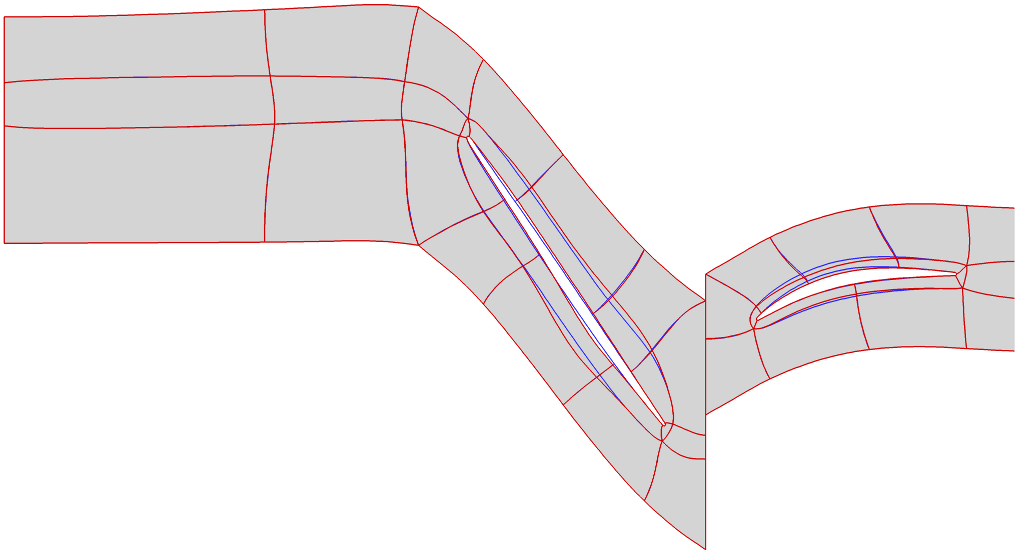

Figure 6.73: Perturbed skeleton after the first stage of the RBF grid perturbation scheme 75% span cut, Darmstadt Rotor No. 1, viscous design case.

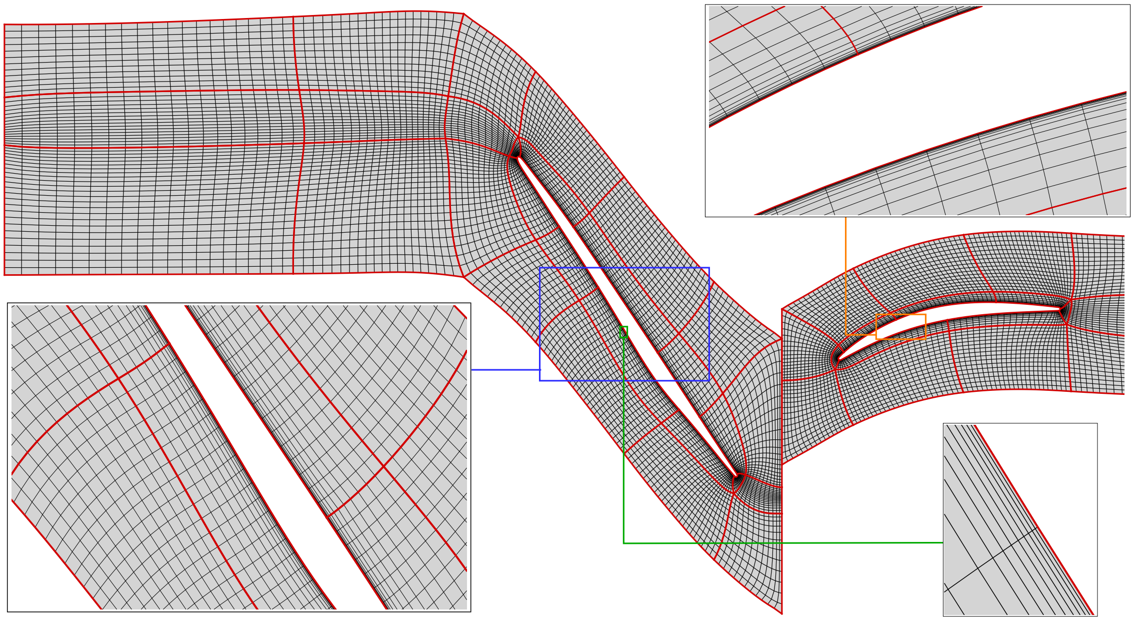

Figure 6.74: Perturbed grid after completion of the RBF grid perturbation scheme, 75% span cut, Darmstadt Rotor No. 1, viscous design case.

# Chapter 7

# Conclusion

This chapter presents the main conclusions which can be drawn from this research. It highlights the contributions and key aspects of the proposed optimization framework and lists several topics which remain open for future work.

## 7.1 Adjoint Solver for Turbomachinery Flow

This dissertation presents the development of the discrete adjoint counterpart of a turbomachinery RANS solver using manual differentiation. The thesis discusses the derivation and accurate linearization of the flow boundary conditions, including the implemented mixing-plane formulation, as well as the resulting adjoint boundary fluxes in detail. The development of an exact adjoint mixing-plane allows for a concurrent rotor-stator optimization and, hence, is of particular importance when applying an adjoint-based aerodynamic shape optimization method to multistage turbomachinery design.

The research reveals that even a slightly inaccurate implementation of the adjoint boundary terms can adversely effect the accuracy of the adjoint solution and, thus, the objective function gradient. An inaccurate gradient information in turn may affect the convergence of the optimization algorithm and ultimately the quality of the optimized design. It is concluded and should be stressed that the inclusion and the correctness of the adjoint boundary fluxes is vital and a prerequisite for an accurate optimization.

Against this background, a major advantage of the discrete adjoint approach is that the adjoint boundary conditions, are obtained straightforward by linearizing the discretized flow boundary conditions. Yet, as illustrated in this thesis, their derivation can be challenging, especially if a boundary condition involves non-local quantities. This, for instance, is the case for the mixing-plane formulation, where averaged quantities are transferred between adjacent blade rows.

It may be summarized that the crucial step in calculating any adjoint boundary flux is the exact linearization of the flow boundary flux with respect to the states in the domain it depends on. The research addresses this aspect and demonstrates that, to simplify the development of the flow-consistent adjoint fluxes for the various boundaries (e.g. solid walls, inlet and outlet interfaces, mixing-plane), it is advisable to construct the different adjoint fluxes from a composition of several matrices and elementary transformations. It is shown that this approach considerably reduces the development cost, since in this case the implementation of the different adjoint boundary conditions largely boils down to selecting and combining the appropriate adjoint matrices and transformations. As depicted in this work, the adjoint boundary fluxes produced by the objective functional and potential constraints can be constructed in a similar fashion and by reusing the same matrices and elementary transformations. Consequently, this approach provides a means to easily implement new objective functionals and minimizes the development work involved in adding additional constraints to the optimization framework.

Gradient verification studies are performed for various set-ups of each investigated compressor and show that the adjoint sensitivities are in very high agreement with the finite-difference gradients predicted by the flow solver. Thus, the results of these sensitivity analyses verify both the accuracy of the developed adjoint solver as well as the correctness of the gradient calculation routines. Furthermore, the adjoint equations and their solutions are interpreted and put into physical context; the analysis leads to various conclusions. First, the adjoint sensitivities, defined as the product of the adjoint co-state vector $\boldsymbol{\psi}$ and the variation of the flow residual $[\partial \mathbf{R}/\partial \mathbf{W}]^T$, determine the geometrical changes occurring during an optimization and, thus, ultimately the change to the objective function. Second, on the discrete level, the adjoint solution in a certain computational cell represents a vector counter-part to the flow solution in that same cell. Consequently, the adjoint co-state vector corresponds to the flow variable vector and signifies the change required to be induced to the primal flux to cause a change in the function of interest. Third, this particularly means that a positive adjoint solution indicates that the primal flux should be increased to meet the design objective and a negative adjoint solution suggests a decrease in the corresponding primal flux.

## 7.2 Grid Perturbation and Design Variables

To fully benefit from the advantages of the adjoint approach, it is important to employ a robust grid perturbation scheme for turbomachinery grids which allows for an efficient but accurate gradient calculation. At the same time, the scheme must produce grids without deteriorating quality, even after a repeated number of perturbation cycles. Moreover, the choice of the design variables is crucial and can largely influence the outcome and success of an optimization. In this work, several grid deformation schemes and different kinds of design variables are studied and the following are the conclusions of this research.

An algebraic grid deformation scheme is employed for all two-dimensional design cases and is found to be relatively inexpensive and very robust. Even after a large number of design iterations and grid perturbations the grid quality remains high and both flow and adjoint solvers converge well. Yet, it is realized that the computational cost and particularly the complexity of the development increase considerably when implementing an algebraic perturbation scheme in a general way such that it can be applied to multi-block grids. To make things worse, the limited generality of algebraic schemes pose severe restrictions on their applicability to a variety of multi-block grid configurations, especially those that are typically used for the calculation of turbomachinery flows. For instance, O-grid configurations, as they are utilized within this work and which are commonly used in turbomachinery CFD, do not provide continuous grid lines from the blade surface to the boundary for the entire grid. Unfortunately, this is a requirement for algebraic perturbation schemes and, thus, it is infeasible to deform certain parts of such a multi-block grid when employing an algebraic approach. Consequently, the algebraic perturbation scheme is discarded in favour of a more general grid movement scheme based on radial basis functions (RBF). The RBF-based perturbation scheme is applied to all three-dimensional design cases and stands out for its generality as well as straightforward applicability to grids of various complexity. Investigations demonstrate that the standard version of the scheme is extremely robust and maintains the grid quality, even for large geometrical perturbations. However, it also is observed that the RBF scheme is expensive and requires a vast amount of memory; up to the point that the standard scheme becomes too expensive to use with highly-resolved viscous grids such as that of the Darmstadt Rotor No. 1 test case. Two novel contributions are presented in this thesis that reduce the memory footprint and consequently lower the computational cost of the RBF-based deformation scheme. First, through a significant reduction of the RBF points by selecting only a subset of the surface points. Second, by splitting the deformation process into two stages, which allows for a partial parallelization of the grid perturbation scheme. These measures reduce the cost of the deformation scheme to a reasonable level without compromising the grid quality and make it applicable to highly-resolved viscous three-dimensional turbomachinery multi-block grids.

The surface grid points serve as design variables for all two-dimensional design cases and the optimization results show that the choice of the surface grid points leads to a wide design space and provides sufficient flexibility to obtain a design with considerable performance improvements. Furthermore, it is found that it is necessary to manually restrict the design space by prescribing geometrical constraints on the leading and trailing edge points of the blades to avoid the generation of unrealistic blade shapes. It requires some tedious trial and error to determine the ideal range to be restricted and how to treat the subsequent grid surface points such that the blade profile remains smooth. Hicks-Henne bump functions are used as design variables for all three-dimensional design cases instead; there are two primary advantages over grid points. First, there is no need to separately smooth the shape modifications. Second, their convenient formulation provides a means to easily modify and distribute the bumps such that no additional geometrical constraints have to be prescribed. Due to these advantages, and despite the fact that choosing Hicks-Henne bump functions over surface grid points slightly restricts the design space and the attainable geometry, it is concluded that Hicks-Henne bump functions represent the favourable choice of design variables for the problems investigated in this work.

## 7.3 Constrained Compressor Optimization

The proposed optimization method is used to redesign two different transonic axial flow compressor configurations while considering various flow conditions and different geometrical set-ups. The following are the primary conclusions from the results of the numerous optimization cases.

Most unconstrained optimization cases yield a remarkable increase in isentropic compressor efficiency while allowing other crucial design parameters such as the mass flow rate and the total pressure ratio to change drastically. The constrained redesigns often produce a comparable compressor efficiency improvement to the unconstrained cases, while maintaining both the mass flow rate and total pressure ratio within tight bounds. This demonstrates that the sequential quadratic programming (SQP) algorithm introduced into the developed adjoint-based optimization framework, provides a means to control the variation of crucial aerodynamic measures with little effort. Since the bounds of the constraints can readily be modified, the design problem can easily be further restricted or relaxed to better understand the relationship between the constraints and the objective function. In particular, the SQP method avoids the time-consuming task of determining appropriate constraint penalty coefficients by trial and error, which typically is required when employing a weighting function approach in which an aggregate function is defined as the sum of the main objective and the constraints. Furthermore, the design cases show that optimization within a SQP framework guarantees that the constraints stay within their prescribed bounds at every design cycle. In contrast, when using a penalty function approach, bounding each constraint by manually prescribing penalty coefficients can result into an expensive trial and error series of simulations. To avoid ill-conditioning of the optimization problem, the penalty coefficients are gradually increased, thus further increasing the computational expense. Finally, as demonstrated by the constrained optimization of the Configuration D compressor, the active set strategies employed in the SQP algorithm, i.e. the treatment of the constrained problems through Lagrangian function formulations, allow for a change in the search direction to further improve the design objective. Thus, optimization within a SQP framework provides a means to find feasible optimum solutions which may not be realizable through a weighting function approach.

The various successful redesigns validate the potential of the developed optimization framework. The benefits of the optimization method are not limited to the generation of optimized geometrical designs. The analysis of the optimization, i.e. the information provided by the adjoint sensitivities for the objective function and constraints, the convergence history of the optimization, and the interpretation of the calculated blade shape modifications, prove to be extremely valuable as well. The proposed optimization method may serve as a direct design tool, but also represents an important analysis tool which can be used to investigate the underlying design space. Thus, it can help the designer to better understand the design problem at hand and may ultimately assist him in further advancing the aerodynamic designs of modern compressor and turbine configurations.

## 7.4 Future Work

This dissertation developed a fully-automatic adjoint-based optimization method for constrained aerodynamic shape optimization in a viscous multistage turbomachinery environment. The general functionality and potential of the method was demonstrated by redesigning two different compressor configurations. At the same time, the developed optimization framework possesses several limitations and the following future work could greatly improve its potential and further advance the state-of-the-art in automatic shape optimization methods for turbomachinery design.

- **Implementation of additional design variables.** Representing the blade geometry and controlling the shape during an optimization via B-splines would further ease the handling of geometrical restrictions. Although Hicks-Henne bump functions generally yield smooth profile modifications as well, these “artificial” design variables cannot be used to parametrize the actual blade geometry. Furthermore, Hicks-Henne design variables perform rather poor in the vicinity of blade leading and trailing edges; a deficit that would be overcome by employing a B-spline parametrization. Moreover, adding design variables such as blade twist and sweep, and utilizing them in combination with the surface design variables, would enhance the design flexibility, extend the design space, and lead to larger design improvements.

- **Optimization of hub geometry and casing.** The aerodynamic performance of a compressor or turbine is not only determined by the rotor and stator blade designs but also by the hub geometry as well as the casing. Especially in the near hub and tip region the flow field is noticeably affected by the hub geometry and the casing. Thus, extending the current optimization framework such that it allows for a concurrent redesign of the blades, hub geometry, and casing would increase the optimization potential considerably.

- **Implementation of new functionals and application to more complex design cases.** Adding additional objective functions and constraints would further increase the applicability of the optimization method and would allow to tie the optimization to various new design aspects. For instance, the optimization method could be applied to high-lift airfoil design. Considering a given multistage turbine, the number of blades per row would be reduced and the remaining blades could then be redesigned such that the loss increase is minimized while enforcing the total pressure ratio and work output of the original turbine configuration. The overall blade loading and its distribution represents an important aerodynamic design parameter, especially for high-lift airfoil design. It could be introduced into the design problem either as a constraint or as a design objective to obtain a specific blade loading or to study the effects of front- and aft-loading. The complexity of the design cases could also be increased by including the turbine cavities in order to investigate secondary flow phenomena or the rotor gab in a transonic compressor to study near-tip designs and to minimize the losses due to the interaction of the tip-leakage vortex with the passage shock.

- **Multi-point optimization and application to robust design.** Multi-point optimization capabilities could be added to the present optimization framework with little effort. This extension would allow to improve a compressor or turbine design not only for a certain operating point (e.g. peak efficiency), but would enable the inclusion of additional operation points (e.g. off-design), therefore leading to a more robust design and annihilating the risk of improving the configuration for a specific operating point while deteriorating the overall operability of the compressor or turbine. Additionally, the optimization method could then be applied to design tasks such as improving a fan design with respect to inflow-angle deviations or increasing the robustness of a compressor with respect to pressure ratio variations.

- **Unsteady multistage optimization.** Utilizing a mixing-plane approach, the proposed optimization method provides the capability to carry out aerodynamic shape optimization in a steady-state multistage turbomachinery environment. This framework represents a tremendous improvement over an isolated redesign of each individual row; yet, it naturally would lack accuracy in comparison to an unsteady multistage optimization method. Extending the current optimization framework to unsteady flows would require the development of both a fast numerical algorithm for unsteady flow analysis and a rapid unsteady adjoint solver. The nonlinear frequency domain (NLFD) approach, which represents a spectral-like scheme that recasts the unsteady governing equations in the temporal domain into a stationary equation system in the frequency domain using Fast Fourier Transform, provides a technique to solve periodic flow problems eight to ten times faster than other time-accurate schemes, cf. McMullen [93]. Therefore, the scheme is especially suited for turbomachinery flows. The extension of the current framework for unsteady multistage optimization employing an unsteady NLFD-based adjoint solver would represent a major advancement in the field of aerodynamic shape optimization for turbomachines.

- **Advancement of grid perturbation schemes.** Grid perturbation is a crucial part of an automatic optimization method. Despite the considerable improvements made in this research, the computational cost of a flexible deformation scheme for complex three-dimensional viscous multi-block grids is by no means negligible and it is important to further advance the schemes currently available to obtain an even more efficient optimization tool.

# Appendix A

For convenience the appendix lists some common transformation matrices and operators related to the Euler and Navier-Stokes equations. These matrices are utilized both within the flow and adjoint solver, in particular when constructing the flow and adjoint boundary conditions.

## A.1 Primitive variable transformation

The conservative variables, defined as

$$\mathbf{W} = \left[\rho,\ \rho u_1,\ \rho u_2,\ \rho u_3,\ \rho E\right]^T , \tag{A.1}$$

can be related to the primitive variables $\mathbf{W_P}$, given by

$$\mathbf{W_P} = \left[\rho,\ u_1,\ u_2,\ u_3,\ p\right]^T , \tag{A.2}$$

through the transformation matrix

$$\mathbf{M} = \frac{\partial \mathbf{W}}{\partial \mathbf{W_P}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u_1 & \rho & 0 & 0 & 0 \\ u_2 & 0 & \rho & 0 & 0 \\ u_3 & 0 & 0 & \rho & 0 \\ \frac{|\mathbf{U}|^2}{2} & \rho u_1 & \rho u_2 & \rho u_3 & \frac{1}{\gamma - 1} \end{bmatrix}, \tag{A.3}$$

and its inverse

$$M^{-1} = \frac{\partial \mathbf{W_P}}{\partial \mathbf{W}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -\dfrac{u_1}{\rho} & \dfrac{1}{\rho} & 0 & 0 & 0 \\ -\dfrac{u_2}{\rho} & 0 & \dfrac{1}{\rho} & 0 & 0 \\ -\dfrac{u_3}{\rho} & 0 & 0 & \dfrac{1}{\rho} & 0 \\ \dfrac{\Upsilon}{2}|\mathbf{U}|^2 & -\Upsilon u_1 & -\Upsilon u_2 & -\Upsilon u_3 & \Upsilon \end{bmatrix}, \tag{A.4}$$

where $\Upsilon = \gamma - 1$.

## A.2 Convective flux Jacobian matrices for conservative variables

The convective flux Jacobian matrices for conservative variables are obtained from linearizing the convective fluxes with respect to the conserved variables, i.e.

$$\mathbf{A}_m = \frac{\delta \mathbf{F}_m}{\delta \mathbf{W}} \ . \tag{A.5}$$

### A.2.1 Cartesian Coordinate Formulation

Formulated in a Cartesian coordinate system, the convective flux Jacobian matrices are defined as

$$\mathbf{A}_m = \left[ \begin{array}{ccc} U_{\text{rot}} & n_1 & n_2 \\ n_1 \dfrac{\Upsilon}{2}|\mathbf{U}|^2 - u_1 U_n & U_n + U_{\text{rot}} - (\Upsilon - 1) n_1 u_1 & n_2 u_1 - \Upsilon n_1 u_2 \\ n_2 \dfrac{\Upsilon}{2}|\mathbf{U}|^2 - u_2 U_n & n_1 u_2 - \Upsilon n_2 u_1 & U_n + U_{\text{rot}} - (\Upsilon - 1) n_2 u_2 \\ n_3 \dfrac{\Upsilon}{2}|\mathbf{U}|^2 - u_3 U_n & n_1 u_3 - \Upsilon n_3 u_1 & n_2 u_3 - \Upsilon n_3 u_2 \\ -\gamma U_n E + \Upsilon U_n |\mathbf{U}|^2 & n_1 \gamma E - \Upsilon u_1 U_n & n_2 \gamma E - \Upsilon u_2 U_n \end{array} \right.$$

$$\left.\begin{array}{cc} n_3 & 0 \\ n_3u_1 - \Upsilon n_1 u_3 & \Upsilon n_1 \\ n_3u_2 - \Upsilon n_2 u_3 & \Upsilon_1 n_2 \\ U_n + U_{\mathrm{rot}} - (\Upsilon - 1) n_3 u_3 & \Upsilon n_3 \\ n_3 \gamma E - \Upsilon u_3 U_n & \gamma U_n + U_{\mathrm{rot}} \end{array}\right] , \tag{A.6}$$

with the abbreviations

$$U_n = u_1n_1 + u_2n_2 + u_3n_3 \ , \qquad U_{\mathrm{rot}} = \Omega_1(n_2x_3 - n_3x_2) \ , \tag{A.7}$$

$$|\mathbf{U}|^2 = u_1^2 + u_2^2 + u_3^2 \ , \qquad \Upsilon = \gamma - 1 \ . \tag{A.8}$$

The individual Jacobian matrices for each Cartesian direction are obtained from Eqn. (A.6) by setting

$$x_1\text{-direction} \rightarrow m = 1 \rightarrow n_1 = 1,\ n_2 = 0,\ n_3 = 0 \ , \tag{A.9}$$

$$x_2\text{-direction} \rightarrow m = 2 \rightarrow n_1 = 0,\ n_2 = 1,\ n_3 = 0 \ , \tag{A.10}$$

$$x_3\text{-direction} \rightarrow m = 3 \rightarrow n_1 = 0,\ n_2 = 0,\ n_3 = 1 \ . \tag{A.11}$$

In computational space, the convective flux Jacobians are calculated from the mapped fluxes, i.e.

$$\boldsymbol{\mathcal{A}}_n = \frac{\partial(S_{nm}\mathbf{F}_m)}{\partial \mathbf{W}} \ , \tag{A.12}$$

for $n, m = 1, 2, 3$.

### A.2.2 Cylindrical Coordinate Formulation

The cylindrical coordinate formulation of the convective flux Jacobian matrices is nearly identical to the Cartesian coordinate expression. Substituting the velocity components yields

$$
\mathbf{A}_m = \begin{bmatrix}
U_{\mathrm{rot}} & n_{x_1} & n_\varphi & n_r & 0 \\
n_{x_1}\dfrac{\Upsilon}{2}|\mathbf{U}|^2 - u_{x_1}U_n & U_n + U_{\mathrm{rot}} - (\Upsilon - 1)n_{x_1}u_{x_1} & n_\varphi u_{x_1} - \Upsilon n_{x_1}u_\varphi & n_r u_{x_1} - \Upsilon n_{x_1} u_r & \Upsilon n_{x_1} \\
n_\varphi\dfrac{\Upsilon}{2}|\mathbf{U}|^2 - u_\varphi U_n & n_{x_1}u_\varphi - \Upsilon n_\varphi u_{x_1} & U_n + U_{\mathrm{rot}} - (\Upsilon - 1)n_\varphi u_\varphi & n_r u_\varphi - \Upsilon n_\varphi u_r & \Upsilon_{x_1} n_\varphi \\
n_r\dfrac{\Upsilon}{2}|\mathbf{U}|^2 - u_r U_n & n_{x_1}u_r - \Upsilon n_r u_{x_1} & n_\varphi u_r - \Upsilon n_r u_\varphi & U_n + U_{\mathrm{rot}} - (\Upsilon - 1)n_r u_r & \Upsilon n_r \\
-\gamma U_n E + \Upsilon U_n|\mathbf{U}|^2 & n_{x_1}\gamma E - \Upsilon u_{x_1}U_n & n_\varphi\gamma E - \Upsilon u_\varphi U_n & n_r\gamma E - \Upsilon u_r U_n & \gamma U_n + U_{\mathrm{rot}}
\end{bmatrix} , \qquad \text{(A.13)}
$$

with abbreviations slightly different to those used in the formulation for the Cartesian coordinate system,

$$
U_n = u_{x_1}n_{x_1} + u_\varphi n_\varphi + u_r n_r \ , \qquad U_{\mathrm{rot}} = \Omega_1 n_\varphi r \ , \qquad \text{(A.14)}
$$

$$
|\mathbf{U}|^2 = u_{x_1}^2 + u_\varphi^2 + u_r^2 \ , \qquad \Upsilon = \gamma - 1 \ . \qquad \text{(A.15)}
$$

The Jacobian matrices for each direction of the cylindrical coordinate system are obtained from Eqn. (A.13) by setting

$$
\begin{array}{lllll}
x_{x_1}\text{-direction} & \rightarrow & m = 1 & \rightarrow & n_{x_1} = 1,\ n_\varphi = 0,\ n_r = 0 \ , \qquad \text{(A.16)} \\
x_\varphi\text{-direction} & \rightarrow & m = 2 & \rightarrow & n_{x_1} = 0,\ n_\varphi = 1,\ n_r = 0 \ , \qquad \text{(A.17)} \\
x_r\text{-direction} & \rightarrow & m = 3 & \rightarrow & n_{x_1} = 0,\ n_\varphi = 0,\ n_r = 1 \ . \qquad \text{(A.18)}
\end{array}
$$

## A.3 Diagonalization of the Convective flux Jacobian matrices

The convective flux Jacobian matrices can be diagonalized by applying the transformation

$$\mathbf{\Lambda}_m = \mathbf{P}_m^{-1}\mathbf{A}_m\mathbf{P}_m \ , \tag{A.19}$$

where $\mathbf{\Lambda}_m$ is the diagonal matrix of the eigenvalues of the Jacobian matrix $\mathbf{A}_m$. Furthermore, $\mathbf{P}_m^{-1}$ denotes the matrix of left eigenvectors, while $\mathbf{P}_m$ contains the right eigenvectors of the convective flux Jacobian matrix. The left and right eigenvector matrices also act as a transformation operator between the conservative $\mathbf{W}$ and characteristic $\delta\mathbf{C}$ variables; in particular

$$\delta\mathbf{C} = \mathbf{P}^{-1}\mathbf{W} \ , \tag{A.20}$$

and

$$\mathbf{W} = \mathbf{P}\delta\mathbf{C} \ . \tag{A.21}$$

The diagonlization of the convective flux Jacobian represents a decomposition into the different characteristic waves of the system, where the right eigenvectors can be viewed as the waves, the characteristic variables as the wave amplitudes, and the eigenvalues as the associated wave speeds.

The characteristic theory is applied in both the flow and adjoint mixing-plane formulation, cf. subsections 3.3.3 and 4.3.5. At a rotor-stator interface the equations are formulated in cylindrical coordinates and the boundary plane is always normal to the $x_1$-direction, thus, only the flux into the $x_1$-direction has to be considered. In this case $m = 1$ and the left eigenvector matrix takes the form

$$\mathbf{P}^{-1} = \begin{bmatrix} 1 - \Upsilon\dfrac{|\mathbf{U}|^2}{2c^2} & \Upsilon\dfrac{u_{x_1}}{c^2} & \Upsilon\dfrac{u_\varphi}{c^2} & \Upsilon\dfrac{u_r}{c^2} & -\Upsilon\dfrac{1}{c^2} \\ -u_{x_1}c + \Upsilon\dfrac{|\mathbf{U}|^2}{2} & c - \Upsilon u_{x_1} & -\Upsilon u_\varphi & -\Upsilon u_r & \Upsilon \\ -\dfrac{u_\varphi}{\rho} & 0 & \dfrac{1}{\rho} & 0 & 0 \\ -\dfrac{u_r}{\rho} & 0 & 0 & \dfrac{1}{\rho} & 0 \\ u_{x_1}c + \Upsilon\dfrac{|\mathbf{U}|^2}{2} & -c - \Upsilon u_{x_1} & -\Upsilon u_\varphi & -\Upsilon u_r & \Upsilon \end{bmatrix}, \tag{A.22}$$

where $\Upsilon = \gamma - 1$. The right eigenvector matrix of the convective flux Jacobian is given by

$$\mathbf{P} = \begin{bmatrix} 1 & \dfrac{1}{2c^2} & 0 & 0 & \dfrac{1}{2c^2} \\ u_{x_1} & \dfrac{u_{x_1} + c}{2c^2} & 0 & 0 & \dfrac{u_{x_1} - c}{2c^2} \\ u_\varphi & \dfrac{u_r}{2c^2} & \rho & 0 & \dfrac{u_\varphi}{2c^2} \\ u_r & \dfrac{u_\varphi}{2c^2} & 0 & \rho & \dfrac{u_r}{2c^2} \\ \dfrac{|\mathbf{U}|^2}{2} & \dfrac{1}{2c^2}\left(H + u_{x_1}c\right) & \rho u_\varphi & \rho u_r & \dfrac{1}{2c^2}\left(H - u_{x_1}c\right) \end{bmatrix}, \tag{A.23}$$

with the enthalpy $H = E + p/\rho$ and the eigenvalues of the system are structured as follows

$$\mathbf{\Lambda}_1 = \begin{bmatrix} u_{x_1} + U_{\text{rot}} & 0 & 0 & 0 & 0 \\ 0 & u_{x_1} + U_{\text{rot}} + c & 0 & 0 & 0 \\ 0 & 0 & u_{x_1} + U_{\text{rot}} & 0 & 0 \\ 0 & 0 & 0 & u_{x_1} + U_{\text{rot}} & 0 \\ 0 & 0 & 0 & 0 & u_{x_1} + U_{\text{rot}} - c \end{bmatrix}. \tag{A.24}$$

# Bibliography

[1] International Civil Aviation Organization, 2012. ICAO News Release COM 30/12. `http://www.icao.int/Newsroom/News%20Doc%202012/COM.30.12.EN.pdf`. last checked: 2014-03-02.

[2] Advisory Council for Aeronautics Research in Europe, 2001. European Aeronautics: A Vision for 2020. `http://www.acare4europe.org/sites/acare4europe.org/files/document/Vision%202020_0.pdf`. last checked: 2014-03-02.

[3] Lions, J., 1971. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York.

[4] Pironneau, O., 1984. *Optimal Shape Design for Elliptic Systems*. Springer-Verlag, New York.

[5] Jameson, A., 1988. "Aerodynamic Design via Control Theory". *Journal of Scientific Computing,* ***3***(3), pp. 233–260.

[6] Nadarajah, S., and Jameson, A., 2000. "A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization". In 38th AIAA Aerospace Sciences Meeting, AIAA Paper 2000-667.

[7] Nadarajah, S., and Jameson, A., 2001. "Studies of the Continuous and Discrete Adjoint Approaches to Viscous Automatic Aerodynamic Optimization". In 15th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2001-2530.

[8] Jameson, A., and Reuther, J., 1994. "Control Theory Based Airfoil Design Using the Euler Equations". In 5th AIAA/USAF/NASA Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper 1994-4272.

[9] Jameson, A., 1995. "Optimum Aerodynamic Design Using CFD and Control Theory". In AIAA 12th Computational Fluid Dynamics Conference, AIAA Paper 1995-1729.

[10] Jameson, A., Pierce, N., and Martinelli, L., 1997. "Optimum Aerodynamic Design Using the Navier-Stokes Equations". In 35th Aerospace Sciences Meeting and Exhibit, AIAA Paper 1997-0101.

[11] Nadarajah, S., and Jameson, A., 2002. "Sonic Boom Reduction Using an Adjoint Method for Wing-Body Configurations in Supersonic Flow". In 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper 2002-5547.

[12] Giles, M., and Pierce, N., 2000. "An Introduction to the Adjoint Approach to Design". *Flow, Turbulence and Combustion,* ***65***, pp. 393–415.

[13] Jameson, A., Shankaran, S., Martinelli, L., and Haimes, B., 2004. "Aerodynamic Shape Optimization of Complete Aircraft Configurations Using Unstructured Grids". In 42nd AIAA Aerospace Sciences Meeting, AIAA Paper 2004-533.

[14] Kim, S., Alonso, J., and Jameson, A., 2004. "Multi-Element High-Lift Configuration Design Optimization Using Viscous Continuous Adjoint Method". *Journal of Aircraft,* ***41***(5), pp. 1082–1097.

[15] Nadarajah, S., and Jameson, A., 2007. "Optimum Shape Design for Unsteady Flows with Time-Accurate Continuous and Discrete Adjoint Methods". *AIAA Journal,* ***45***(7), pp. 1478–1491.

[16] Nadarajah, S., and Jameson, A., 2006. "Optimum Shape Design for Unsteady Three-Dimensional Viscous Flows Using a Non-Linear Frequency Domain Method". In 4th AIAA Applied Aerodynamics Conference, AIAA Paper 2006-3455.

[17] Yang, S., Wu, H., Liu, F., and Tasi, H., 2003. "Aerodynamic Design of Cascades by Using an Adjoint Equation Method". In 41st AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2003-1068.

[18] Chung, J., Martin, G., and Lee, K., 2004. "Aerodynamic Design of 3D Compressor Blade Using an Adjoint Method". In 42nd AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2004-27.

[19] Wu, H., Liu, F., and Tasi, H., 2005. "Aerodynamic Design of Turbine Blades Using an Adjoint Equation Method". In 43rd AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2005-1006.

[20] Papadimitriou, D., and Giannakoglou, K., 2007. "A Continuous Adjoint Method with Objective Function Derivatives Based on Boundary Integrals for Inviscid and Viscous Flows". *Journal of Computer and Fluids,* ***36***(2), pp. 325–341.

[21] Papadimitriou, D., and Giannakoglou, K., 2006. "A Continuous Adjoint Method for the Minimization of Losses in Cascade Viscous Flows". In 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2006-49.

[22] Papadimitriou, D., and Giannakoglou, K., 2007. "Total Pressure Loss Minimization in Turbomachinery Cascades Using a New Continuous Adjoint Formulation". *Journal of Power and Energy,* ***221***, pp. 865–872.

[23] Corral, R., and Gisbert, F., 2008. "Profiled End Wall Design Using an Adjoint Navier-Stokes Solver". *Journal of Turbomachinery,* ***130***(2).

[24] Li, H., Song, L., Li, Y., and Feng, Z., 2011. "2D Viscous Aerodynamic Shape Design Optimization for Turbine Blades Based on Adjoint Method". *Journal of Turbomachinery,* ***133***(3).

[25] Luo, J., Xiong, J., Liu, F., and McBean, I., 2011. "Three-Dimensional Aerodynamic Design Optimization of a Turbine Blade by Using an Adjoint Method". *Journal of Turbomachinery,* ***133***(1).

[26] Luo, J., Zhou, C., and Liu, F., 2013. "Multipoint Design Optimization of a Transonic Compressor Blade by Using an Adjoint Method". *Journal of Turbomachinery,* ***136***(5).

[27] Mousavi, A., and Nadarajah, S., 2010. "Heat Transfer Optimization of Gas Turbine Blades Using an Adjoint Approach". In 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, AIAA Paper 2010-9048.

[28] Mousavi, A., and Nadarajah, S., 2011. "Adjoint-Based Multidisciplinary Design Optimization of Cooled Gas Turbine Blades". In 49th AIAA Aerospace Sciences Meeting, AIAA Paper 2011-1131.

[29] Denton, J., 1992. "The Calculation of Three-Dimensional Viscous Flow Through Multistage Turbomachines". *Journal of Turbomachinery,* ***114***(1).

[30] Frey, C., Kersken, H.-K., and Nürnberger, D., 2009. "The Discrete Adjoint of a Turbomachinery RANS Solver". In ASME Turbo Expo 2009: Power for Land, Sea and Air, ASME Paper 2009-GT-59062.

[31] Wang, D., and He, L., 2010. "Adjoint Aerodynamic Design Optimization for Blades in Multistage Turbomachines: Part I: Methodology and Verification". *Journal of Turbomachinery,* ***132***(2).

[32] Wang, D., and He, L., 2010. "Adjoint Aerodynamic Design Optimization for Blades in Multistage Turbomachines: Part II: Validation and Application". *Journal of Turbomachinery,* ***132***(2).

[33] Walther, B., and Nadarajah, S., 2013. "Constrained Adjoint-Based Aerodynamic Shape Optimization of a Single Stage Transonic Compressor". *Journal of Turbomachinery,* ***135***(2).

[34] Walther, B., and Nadarajah, S., 2014. "An Adjoint-Based Optimization Method for Constrained Aerodynamic Shape Design of Three-Dimensional Blades in Multi-Row Turbomachinery Configurations". In ASME Turbo Expo 2014: Power for Land, Sea and Air, ASME Paper 2014-GT-26604.

[35] Marta, A., der Weide, E. V., Alonso, J., Mader, C., and Martins, J., 2007. "A Method for the Development of Discrete Adjoint Solvers Using Automatic Differentiation Tools". *International Journal of Computational Fluid Dynamics,* ***21***, pp. 307–327.

[36] Navier, C. L. M. H., 1823. "Mémoire sur les Lois du Mouvement des Fluides". *Mém. Acad. Roy. Sci.,* ***6***, pp. 389–416.

[37] de Saint-Venant, B., 1843. “Mémoire sur la Dynamique des Fluides”. *Comp. Rendus,* ***17***, pp. 1240–1243.

[38] Stokes, J. L., 1846. “On the Theories of the Internal Friction of Fluids in Motion”. *Transactions of the Cambridge Philosophical Society,* ***8***, pp. 287–305.

[39] Chen, J., Ghosh, A., Sreenvias, K., and Whitfield, D., 1997. “Comparison of Computations Using Navier-Stokes Equations in Rotating and Fixed Coordinates for Flow Through Turbomachinery”. In 35th AIAA Aerospace Sciences Meeting, AIAA Paper 1997-878.

[40] Wilcox, D. C., 1993. *Turbulence Modeling for CFD.* Griffin.

[41] Blazek, J., 2001. *Computational Fluid Dynamics: Principles and Applications.* Elsevier.

[42] Menter, F. R., 1994. “Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications”. *AIAA Journal,* ***32***(8), pp. 1598–1605.

[43] Menter, F. R., 1982. “Improved Two-Equation $k$-$\omega$ Turbulence Models for Aerodynamic Flows”. In NASA TM 103975.

[44] Spalart, P. R., and Rumsey, C. L., 2007. “Effective Inflow Conditions for Turbulence Models in Aerodynamic Calculations”. *AIAA Journal,* ***45***(10), pp. 2544–2553.

[45] Wilcox, D. C., 1988. “Reassessement of the Scale-Determining Equation for Advanced Turbulence Models”. *AIAA Journal,* ***26***(11), pp. 1299–1310.

[46] Moore, J. G., and Moore, J., 1999. “Realizability in Turbulence Modelling for Turbomachinery CFD”. In ASME Turbo Expo 1999: Power for Land, Sea and Air, ASME Paper 1999-GT-24.

[47] Koubogiannis, D. G., Athanasiadis, A. N., and Giannakoglou, K. C., 2003. “One and Two-Equation Turbulence Models for the Prediction of Complex Cascade Flows Using Unstructured Grids”. *Computers & Fluids,* ***32***, pp. 403–430.

[48] Thompson, J. E., Warsi, Z. U. A., and Mastin, C. W., 1997. *Numerical Grid Generation.* Elsevier.

[49] Hirsch, C., 1990. *Numerical Computation of Internal and External Fows*, Vol. 2. John Wiley & Sons.

[50] Jameson, A., Schmidt, W., and Turkel, E., 1981. "Numerical Solutions of the Euler Equations by Finite Volume Methods with Runge-Kutta Time-Stepping Schemes". In 14th AIAA Fluid and Plasma Dynamic Conference, AIAA Paper 1981-1259.

[51] Martinelli, L., and Jameson, A., 1988. "Validation of a Multigrid Method for the Reynolds Averaged Equations". In 26th AIAA Aerospace Sciences Meeting, AIAA Paper 1988-0414.

[52] Martinelli, L., 1987. "Calculations of Viscous Flow with a Multigrid Method". PhD thesis, Departement of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, Oct.

[53] Chima, R., 1998. "Calculation of Multistage Turbomachinery Using Steady Characteristic Boundary Conditions". In 36th AIAA Aerospace Sciences Meeting, AIAA Paper 1998-0968.

[54] Giles, M., 1990. "Nonrefelecting Boundary Conditions for Euler Equation Calculations". *AIAA Journal,* ***28***, pp. 2050–2058.

[55] Wyss, M., Chima, R., and Tweedt, D., 1993. "Averaging Techniques for Steady and Unsteady Calculations of a Transonic Fan Stage". In NASA TM 16231.

[56] Nadarajah, S., 2003. *The Discrete Adjoint Approach to Aerodynamic Shape Optimization.* PhD thesis, Stanford University, Dept. of Aeronautics and Astronautics.

[57] Jameson, A., and Baker, T., 1983. "Solution of the Euler Equations for Complex Configurations". In 6th AIAA Computational Fluid Dynamics Conference, AIAA Paper 83-1929.

[58] Turkel, E., Swanson, R., Vatsa, V., and White, J., 1991. “Multigrid for Hypersonic Viscous Two- and Three-Dimensional Flows”. In 10th AIAA Computational Fluid Dynamics Conference, AIAA Paper 91-1572.

[59] Brandt, A., 1977. “Multi-Level Adaptive Solutions to Boundary-Value Problems”. *Mathematics of Computations,* ***31***, pp. 333–390.

[60] Jameson, A., 1983. “Solution of the Euler Equations for Two-Dimensional Transoinc Flow by a Multigrid Method”. *Applied Mathematics and Computation,* ***13***, pp. 327–356.

[61] Marta, A., and Shankaran, S., 2013. “On the Handling of Turbulence Equations in RANS Adjoint Solvers”. *Computers & Fluids,* ***74***, pp. 102–113.

[62] Hicks, R., and Henne, P., 1978. “Wing Design by Numerical Optimization”. *Journal of Aircraft,* ***15***, pp. 407–412.

[63] Jameson, A., 1990. “Automatic Design of Transonic Airfoils to Reduce the Shock Inducded Pressure Drag”. In 31st Israel Annual Conference on Aviation and Aeronautics.

[64] Burgreen, G. W., and Baysal, O., 1996. “Three-Dimensional Aerodynamic Shape Optimization Using Discrete Sensitivity Analysis”. *AIAA Journal,* ***34***(9), pp. 1761–1770.

[65] Batina, J., 1991. “Unsteady Euler Algorithm With Unstructured Dynamic Mesh for Complex-Aircraft Aerodynamic Analysis”. *AIAA Journal,* ***29***(3), pp. 327–333.

[66] Batina, J., 1990. “Unsteady Euler Flow Solutions Using Unstructured Dynamic Meshes”. *AIAA Journal,* ***28***(8), pp. 1381–1388.

[67] Farhat, C., Degand, C., Koobus, B., and Lesoinne, M., 1998. “Torsional Springs for Two-Dimensional Dynamic Unstructured Fluid Meshes”. *Computer Methods in Applied Mechanics and Engineering,* ***163***(1-4), pp. 231–245.

[68] Johnson, A., and Tezduyar, T., 1994. “Mesh Update Strategies in Parallel Finite Element Computations of Flow Problems With Moving Boundaries and

Interfaces". *Computer Methods in Applied Mechanics and Engineering,* ***119***(1-2), pp. 73–94.

[69] Truong, A., Oldfield, C., and Zingg, D., 2008. "Mesh Movement for a Discrete-Ajoint Newton-Krylov Algorithm for Aerodynamic Optimization". *AIAA Journal,* ***46***(7), pp. 1965–1704.

[70] Hicken, J., and Zingg, D., 2010. "Aerodynamic Optimization Algorithm for Integrated Geometry Paramtrization and Mesh Movement". *AIAA Journal,* ***48***(2), pp. 401–413.

[71] de Boer, A., van der Schoot, M., and Bijl, H., 2007. "Mesh Deformation Based on Radial Basis Function Interpolation". *Computers and Structures,* ***85***, pp. 784–795.

[72] Rendall, T., and Allen, C., 2010. "Parallel Efficient Mesh Motion Using Radial Basis Functions With Application to Multi-Bladed Rotors". *Journal of Numerical Methods in Engineering,* ***81***(1), pp. 89–105.

[73] Jakobsson, S., and Amoignon, O., 2007. "Mesh Deformation Using Radial Basis Functions for Gradient-Based Aerodynamic Shape Optimization". *Computers and Fluids,* ***36***(6), pp. 1119–1136.

[74] Poirier, V., and Nadarajah, S., 2012. "Efficient RBF Deformation Within an Adjoint-Based Aerodynamic Optimization Framework". In 50th AIAA Aerospace Sciences Meeting, AIAA Paper 2012-0059.

[75] Gill, P., Murray, W., and Saunders, M., 2008. *User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming.* Manual.

[76] Gill, P., Murray, W., and Saunders, M., 2005. "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization". *SIAM Review,* ***47***(1), pp. 99–131.

[77] NPARC Alliance CFD Verification and Validation Archive, 2011. Turbulent Flat Plate. `http://http://www.grc.nasa.gov/WWW/wind/valid/fpturb/fpturb.html`. last checked: 2014-03-22.

[78] Wieghardt, K., and Tillmann, W., 1951. *On the Turbulent Friction Layer for Rising Pressure*. NASA TM-1314.

[79] Bogar, T. J., Sajben, M., and Kroutil, J. C., 1983. "Characteristic Frequencies of Transonic Diffuser Flow Oscillations". *AIAA Journal,* ***21***(9), pp. 1232–1240.

[80] Sajben, M., Bogar, T. J., and Kroutil, J. C., 1984. "Forced Oscillation Experiments in Supercritical Diffuser Flows". *AIAA Journal,* ***22***(4), pp. 465–474.

[81] Salmon, J. T., Bogar, T. J., and Sajben, M., 1983. "Laser Doppler Velocimeter Measurements in Unsteady Seperated Transonic Diffuser Flows". *AIAA Journal,* ***21***(12), pp. 1690–1697.

[82] Hasieh, T., Bogar, T. J., and Coakley, T. J., 1987. "Numerical Simulation and Comparison with Experiment for Self-Excited Oscillations in a Diffuser Flow". *AIAA Journal,* ***25***(7), pp. 936–943.

[83] NPARC Alliance CFD Verification and Validation Archive, 2011. Sajben Transonic Diffuser. `http://http://www.grc.nasa.gov/WWW/wind/valid/transdif/transdif.html`. last checked: 2014-03-20.

[84] D. Yoder, 2008. Sajben Transonic Diffuser: Study 2. `http://www.grc.nasa.gov/WWW/wind/valid/transdif/transdif02/transdif02.html`. last checked: 2014-03-20.

[85] Nealy, D. A., Gladden, H., Mihelic, M. S., and Hylton, L. D., 1984. "Measurments of Heat Transfer Distribution Over the Surfaces of Highly Loaded Turbine Nozzle Guide Vanes". *Journal of Engineering for Gas Turbines and Power,* ***106***(1), pp. 149–158.

[86] National Aeronautics and Space Administration, 1999. Mark II Turbine Vane. `http://www.grc.nasa.gov/WWW/Acoustics/code/adpac/sample/markii/`. last checked: 2014-03-20.

[87] Silkowski, P., and Hall, K., 1998. "A Coupled Mode Analysis of Unsteady Multistage Flows in Turbomachinery". *Journal of Turbomachinery,* ***120***(3), pp. 410–421.

[88] Ekicic, K., and Hall, K., 2007. "Nonlinear Analysis of Unsteady Flows in Multistage Turbomachines Using Harmonic Balance". *AIAA Journal,* ***45***(5), pp. 1047–1057.

[89] Blaha, C., Hennecke, D., Fritsch, G., Höger, M., and Beversdorff, M., 1997. "Laser-2-Focus Measurements in a Transonic Compressor Blisk-Rotor and Comparison with 3D Numerical Simulations". In 13th International Symposium Air Breathing Engines, ISABE Paper 1997-706.

[90] Fritsch, G., Höger, M., Blaha, C., and Bauer, D., 1998. "Viscous 3D Compressor Simulations on Parallel Architectures". In International Gas Turbine & Aeroengine Congress, ASME Paper 98-GT-261.

[91] Höger, M., Fritsch, G., and Bauer, D., 1997. "Numerical Simulation of the Shock-Tip Leakage Vortex Interaction in a HPC Front Stage". In 33rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference, AIAA Paper 1997-2876.

[92] Shankaran, S., Marta, A., Venugopal, P., Barr, B., and Wang, Q., 2012. "Interpretation of Adjoint Solutions for Turbomachinery Flows". In ASME Turbo Expo 2012: Power for Land, Sea and Air, ASME Paper 12-GT-69588.

[93] McMullen, M., Jameson, A., and Alonso, J., 2002. "Application of a Non-Linear Frequency Domain Solver to the Euler and Navier-Stokes Equations". In 40th Aerospace Sciences Meeting and Exhibit, AIAA Paper 2002-0120.