



**TÉCNICO**  
LISBOA

**UNIVERSIDADE DE LISBOA  
INSTITUTO SUPERIOR TÉCNICO**

**Aero-thermal analysis and design of turbomachinery blades  
using multi-stage adjoint methods**

**Simão Santos Rodrigues**

**Supervisor: Doctor André Calado Marta**

**Thesis approved in public session to obtain the PhD Degree in**

**Aerospace Engineering**

**Jury final classification: Pass with Distinction**

**2019**





**TÉCNICO**  
LISBOA

**UNIVERSIDADE DE LISBOA**  
**INSTITUTO SUPERIOR TÉCNICO**

**Aero-thermal analysis and design of turbomachinery blades  
using multi-stage adjoint methods**

**Simão Santos Rodrigues**

**Supervisor: Doctor André Calado Marta**

**Thesis approved in public session to obtain the PhD Degree in  
Aerospace Engineering**

**Jury final classification: Pass with Distinction**

**Jury**

**Chairperson:** Doctor Hélder Carriço Rodrigues, Instituto Superior Técnico, Universidade de Lisboa

**Members of the Committee:**

Doctor José Arnaldo Pereira Leite Miranda Guedes, Instituto Superior Técnico,  
Universidade de Lisboa

Doctor João Eduardo de Barros Teixeira Borges, Instituto Superior Técnico, Universidade  
de Lisboa

Doctor Pedro Vieira Gamboa, Faculdade de Engenharia, Universidade da Beira Interior

Doctor Miguel Ângelo Rodrigues Silvestre, Faculdade de Engenharia, Universidade da  
Beira Interior

Doctor André Calado Marta, Instituto Superior Técnico, Universidade de Lisboa

Doctor Sriram Shankaran, GE Aviation, EUA

**Funding Institutions**

Fundação para a Ciência e a Tecnologia

**2019**



## **Acknowledgments**

I would like to thank my supervisor Professor André Calado Marta for providing the the opportunity to follow this PhD in this project and for his support and knowledgeable words of advice during these past years.

I would also like to thank General Electric for providing the opportunity to develop this work on their in-house solver and their funding of the first year of my studies through ID-MEC and Fundação para a Ciência e Tecnologia (FCT), Portugal, for the PhD scholarship SFRH/BD/97521/2013.

And lastly I would like to thank all my friends and family for their support and words of motivation throughout these years.



## Resumo

A utilização de ferramentas de alta fidelidade de mecânica de fluidos computacional na análise e design (projeto) de turbomáquinas tem vindo a aumentar nos últimos anos, resultado do aumento da capacidade computacional e melhorias nos métodos numéricos. Estas ferramentas são muito frequentemente utilizadas em ambientes de otimização numérica, onde os algoritmos de otimização baseados em gradientes são bastante comuns devido à sua eficiência. Nos casos em que os problemas de otimização contêm um elevado número de variáveis de design, como acontece em problemas típicos de design de turbomáquinas, o recurso ao método das variáveis adjuntas no cálculo dos gradientes prova-se bastante benéfico, uma vez que proporciona uma maneira de obter sensibilidade exatas de funções com um custo computacional quase independente do número de variáveis de design. A consideração do acoplamento entre os vários andares adjacentes de uma turbomáquina multi-andar na sua análise é de extrema importância, uma vez que a interação entre as várias pás dos diferentes andares, quer na direção do escoamento, quer na oposta, causa complexos fenómenos aerodinâmicos. Para a simulação de andares de turbomáquinas tendo em conta estes fenómenos de acoplamento, o tratamento de plano de mistura é dos métodos mais utilizados e é atualmente uma ferramenta standard em ambientes de design industrial. Esta tese apresenta a formulação e descrição da implementação da parte adjunta da interface de plano de mistura num código adjunto para aplicação em escoamentos em turbomáquinas, proporcionando assim a sua capacidade de contemplar o acoplamento dos vários andares na sua análise. O código adjunto é desenvolvido utilizando um método híbrido denominado adjoint em que as derivadas parciais do sistema de equações adjuntas são obtidas através da diferenciação automática do código direto. A implementação é verificada com aproximação de diferenças finitas, através da qual, é confirmada a correta implementação do código. O código adjunto é posteriormente utilizado na análise de sensibilidade de várias métricas de performance em relação a variáveis que definem a geometria das pás da turbomáquina e condições fronteira do escoamento, através do plano de mistura, evidenciando assim o acoplamento físico em turbomáquinas multi-andar.

**Palavras-chave:** Plano de mistura; Análise de sensibilidade; Diferenciação Automática; Variáveis Adjuntas Discretas; Otimização de forma



## Abstract

The use of high-fidelity computational fluid dynamics (CFD) tools in turbomachinery design has seen a continuous increase as a result of computational power growth and the improvement of numerical methods. These tools are often used in optimization environments, where gradient-based optimization algorithms are the most common due to their efficiency. In cases where the optimization contains a large number of design variables, as is often the case in turbomachinery design problems, the adjoint approach for calculating the gradients is beneficial, as it provides a way of obtaining exact function sensitivities with a computational cost that is nearly independent of the number of design variables. Taking the coupling between adjacent blade rows of a turbomachine in its analysis is of the utmost importance, as the interaction between the various rows both in the direction of the flow and in the opposite direction can cause complex phenomena that considering a single row will not capture. The most commonly used method to address these effects (i.e. coupling in the simulation of multiple rows) is the mixing-plane treatment, which has become a standard industrial tool in the design environment. This thesis presents the formulation and implementation of the adjoint counterpart of the mixing-plane interface in a legacy adjoint solver for sensitivity analysis of turbomachinery applications to handle multi-row problems. The solver is developed using the discrete ADjoint approach, where the partial derivatives required for the assembly of the adjoint system of equations are obtained using automatic differentiation tools. The differentiation is not performed in one go, but, instead, the individual routines that perform the various steps of the mixing-plane algorithm are differentiated and the final differentiated routine, corresponding to the differentiated mixing-plane is assembled by hand. The implementation is verified with finite-difference approximations and the sensitivity of several performance metrics relative to neighbor blade/hub row geometry and boundary conditions are shown to highlight the physical coupling in multi-row turbomachines.

**Keywords:** Mixing-plane; automatic differentiation; discrete adjoint; sensitivity analysis; shape optimization



# Contents

Acknowledgments . . . . .	i
Resumo . . . . .	iii
Abstract . . . . .	v
List of Tables . . . . .	xi
List of Figures . . . . .	xviii
Nomenclature . . . . .	xxi
Glossary . . . . .	xxiii
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Overview of Turbomachines . . . . .	4
1.2.1 Types of turbomachines . . . . .	5
1.2.2 Flowpath and Thermodynamics . . . . .	5
1.2.3 Secondary Flows . . . . .	7
1.3 Numerical Optimization . . . . .	8
1.4 Adjoint-based Sensitivity Analysis in Turbomachinery . . . . .	14
1.5 Contributions to the State-of-the-art . . . . .	17
1.6 Thesis Outline . . . . .	18
<b>2 Numerical Analysis of Turbomachinery</b>	<b>19</b>
2.1 Thermodynamic Relations . . . . .	19
2.2 Reynolds-Averaged Navier-Stokes Equations . . . . .	21
2.2.1 Discretization of the RANS Equations . . . . .	23
2.2.2 Physical Boundary Conditions . . . . .	25
2.2.3 Other Boundary Conditions . . . . .	29
2.3 Multi-row Turbomachinery Analysis . . . . .	31
2.3.1 Frozen Rotor . . . . .	32

2.3.2	Steady Mixing-plane . . . . .	32
2.3.3	Unsteady Sliding-mesh . . . . .	37
<b>3</b>	<b>Techniques of Sensitivity Analysis</b>	<b>39</b>
3.1	Finite Differences . . . . .	39
3.2	Complex Step . . . . .	41
3.3	Automatic Differentiation . . . . .	41
3.3.1	Forward and Reverse Modes . . . . .	42
3.3.2	Source Transformation and Operator Overloading . . . . .	43
3.3.3	Example using <i>Tapenade</i> . . . . .	44
3.4	Semi-analytic Methods . . . . .	46
3.4.1	Direct Method . . . . .	48
3.4.2	Adjoint Method . . . . .	48
3.5	Summary of the Various Approaches . . . . .	51
3.6	Hybrid ADjoint Approach . . . . .	52
<b>4</b>	<b>Derivation of the Adjoint of the Euler Equations</b>	<b>53</b>
4.1	Euler Equations and Adjoint Procedure . . . . .	53
4.2	Derivation of the Continuous Adjoint Equation . . . . .	54
4.3	Derivation of the Discrete Adjoint Equations . . . . .	59
<b>5</b>	<b>Adjoint Multi-row Turbomachinery Interface</b>	<b>65</b>
5.1	Adjoint Multi-row Formulation . . . . .	65
5.2	Implementation of the Adjoint Steady Mixing Plane Interface . . . . .	69
5.2.1	Rewriting of Original routines . . . . .	70
5.2.2	Automatic Differentiation of Rewritten Routines . . . . .	74
5.2.3	Hand-assembly of Differentiated Routines . . . . .	75
5.2.4	Integration with Single-stage Routines . . . . .	76
5.3	Overview of the Implementation Effort . . . . .	78
<b>6</b>	<b>Verification of the Implementation of the Adjoint Mixing-plane Interface</b>	<b>81</b>
6.1	Description of the Study Case . . . . .	81
6.2	Overview of the Metrics of Performance . . . . .	84
6.3	Overview of the Independent Variables . . . . .	86

6.3.1	Inlet and Exit Boundary Conditions . . . . .	86
6.3.2	Blade Shape and Hub Geometry . . . . .	87
6.4	Verification of the Rewritten Mixing-plane Interface . . . . .	88
6.4.1	Profile Computation . . . . .	88
6.4.2	Boundary Conditions Update . . . . .	90
6.5	Verification of the Adjoint Mixing-plane Interface . . . . .	90
6.5.1	Stencil of Dependency . . . . .	91
6.5.2	Local Sensitivity . . . . .	91
6.5.3	Inter-row Sensitivity . . . . .	92
6.6	Verification of the Adjoint-based Total Derivatives . . . . .	95
6.6.1	Sensitivity of Stator Pressure Loss . . . . .	97
6.6.2	Sensitivity of Rotor Pressure Ratio . . . . .	99
6.6.3	Sensitivity of Stage Pressure Ratio . . . . .	100
6.6.4	Sensitivity of Rotor Efficiency . . . . .	102
6.6.5	Sensitivity of Stage Efficiency . . . . .	104
6.6.6	Sensitivity of Exit Mass Flow . . . . .	105
6.6.7	Sensitivity of Rotor Blade Averaged Total Temperature . . . . .	106
6.7	Final Remarks . . . . .	107
<b>7</b>	<b>Sensitivity Analysis of a Stator/Rotor Stage of a Low Pressure Turbine</b>	<b>109</b>
7.1	Sensitivity to Boundary Conditions . . . . .	110
7.1.1	Rotor and Stage Efficiencies . . . . .	110
7.1.2	Rotor and Stage Pressure Ratio . . . . .	119
7.2	Sensitivity to Blade Shape and Hub Geometry . . . . .	122
7.2.1	Blade Shape . . . . .	122
7.2.2	Hub Geometry . . . . .	122
7.2.3	Improving Stage Efficiency with Endwall Contouring . . . . .	125
7.3	Thermal Sensitivity Analysis . . . . .	135
7.4	Final Remarks . . . . .	137
<b>8</b>	<b>Conclusions</b>	<b>139</b>
8.1	Achievements . . . . .	141
8.2	Future Work . . . . .	142



# List of Tables

1.1	Hypothetical savings of increasing fuel efficiency of a Boeing 757-200 on a fleet of 100 aircraft. . . . .	4
3.1	Comparison of discrete and continuous adjoint approaches. . . . .	50
3.2	Summary of various sensitivity analysis approaches. . . . .	51
6.1	Selection of test cases included in verification of total derivatives. . . . .	96
6.2	Comparison of computational requirements of direct and adjoint solvers (normalized by the direct solver). . . . .	97
7.1	Selection of test cases included in verification of total derivatives. . . . .	110
7.2	Hypothetical time requirements for equivalent of adjoint-based sensitivity analysis with finite-differences. . . . .	138



# List of Figures

1.1	Schematic of the various areas of application of turbomachinery. . . . .	2
1.2	Total global airline industry's fuel costs and net profits. . . . .	3
1.3	Classification of the turbomachines by the stream direction in relation to the axis of the shaft. . . . .	5
1.4	Schematic of simple turbomachine and respective thermodynamic cycle. . .	6
1.5	Schematic of a typical turbofan . . . . .	7
1.6	Representation of secondary flow phenomena in high-pressure compressor blades. . . . .	7
1.7	Schematic representation of the various categories of optimization algorithms.	9
1.8	Optimization history of a function with local and global minima. . . . .	10
1.9	Different methods for computing the gradient of a function. . . . .	12
1.10	Alternative approaches to obtain the discrete adjoint equations. . . . .	13
1.11	Representation of a typical adjoint-based sensitivity analysis framework. .	15
2.1	Non-isentropic compression and expansion processes. . . . .	21
2.2	Stencil of dependence of the residual of a single cell. . . . .	25
2.3	Sketch of cell-centered grid, with interior and auxiliary nodes. . . . .	25
2.4	Schematic of multiblock 1-2-1 boundary condition update. . . . .	30
2.5	Schematic of generalized multi-block boundary condition update. . . . .	30
2.6	Various approaches to the numerical simulation of a turbomachinery. . . .	31
2.7	Two-row computational domain using a single blade per row. . . . .	32
2.8	Schematic sketch of the flux balance algorithm . . . . .	33
2.9	Schematic representation of the mixing-plane interface steps. . . . .	37
3.1	Relative error of finite-difference approximations of the first derivative of a function with respect to the perturbation step. . . . .	40
3.2	Original Fortran subroutine to be differentiated. . . . .	44

3.3	Fortran subroutine automatically differentiated using the forward mode. . .	45
3.4	Fortran subroutine automatically differentiated using the reverse mode. . .	46
4.1	Schematic of an airfoil, domain of integration and respective boundaries. . .	55
5.1	Schematic of a multistage turbomachine modeled with a sequence of $N_r$ single blade passage rows. . . . .	66
5.2	Schematic of the original direct multi-row exchange algorithm . . . . .	71
5.3	Modification of profile data structure to apply automatic differentiation. . .	72
5.4	Schematic of the rewritten direct multi-row exchange algorithm. . . . .	74
5.5	Partial schematic of the manually assembled differentiated mixing-plane routine. . . . .	75
5.6	Simplified dataflow representation of the rewritten direct mixing-plane al- gorithm. . . . .	76
5.7	Stencil of influence of residual of a cell close to a mixing-plane boundary. .	77
5.8	Schematic representation of the assemble of the differentiated mixing-plane routine with the single-row differentiated routine. . . . .	78
5.9	Comparison of code metrics of the direct solver, adjoint solver before mixing- plane and adjoint solver after implementation of mixing-plane. . . . .	79
6.1	Representation of stator-rotor stage in study and respective computational mesh. . . . .	82
6.2	Division of the computational domains into various individual blocks. . . .	82
6.3	Schematic representation of the inlet and exit boundary surfaces of the stator-rotor stage domain.) . . . . .	83
6.4	Residual iteration history of flow solution convergence of stator/rotor stage simulation. . . . .	83
6.5	Residual iteration history of GMRES for adjoint solutions. . . . .	84
6.6	Normalized density field of converged direct solution of stator/rotor turbine stage. . . . .	85
6.7	Relative difference between averaged fluxes $\mathbf{F}_1$ to $\mathbf{F}_5$ computed with original and rewritten routines. . . . .	89
6.8	Relative difference between averaged fluxes $\mathbf{F}_6$ and $\mathbf{F}_6$ computed with original and rewritten routines. . . . .	89

6.9	Maximum relative difference between updated auxiliary cells values computed with original and rewritten routines. . . . .	90
6.10	Relative difference between updated auxiliary cells values computed with original and rewritten routines in row 1. . . . .	91
6.11	Sensitivity of $\mathbf{R}_{\text{rec}}$ of a single cell on the other interior cells in the same face $\partial\mathbf{R}_{\text{rec}}/\partial\mathbf{q}_{\text{local}}$ . . . . .	92
6.12	Comparison of mixing-plane local sensitivities of the residual of a cell at the exit of the stator computed with differentiated routines and FD approximations. . . . .	93
6.13	Comparison of mixing-plane local sensitivities of the residual of a cell at the inlet of the rotor computed with differentiated routines and FD approximations. . . . .	94
6.14	Maximum relative difference between inter-row sensitivities obtained with differentiated routines and second-order finite-difference approximations. . . . .	95
6.15	Adjoint-based sensitivity of stator pressure ratio $\pi_{\text{stator}}$ to exit static pressure $p_s^{\text{exit}}$ boundary condition. . . . .	98
6.16	Verification of adjoint-based sensitivities of stator pressure loss $\pi_{\text{stator}}$ to exit static pressure $p_s^{\text{exit}}$ boundary condition using FD. . . . .	98
6.17	Adjoint-based sensitivity of rotor pressure ratio $\pi_{\text{rotor}}$ to hub wall radial position $r_{\text{hub}}$ . . . . .	99
6.18	Verification of adjoint-based sensitivities of rotor pressure ratio $\pi_{\text{rotor}}$ to hub wall grid x-coordinates using FD. . . . .	100
6.19	Adjoint-based sensitivity of rotor pressure ratio $\pi_{\text{rotor}}$ inlet total enthalpy $h_T^{\text{inlet}}$ boundary condition. . . . .	100
6.20	Verification of adjoint-based sensitivities of rotor pressure ratio $\pi_{\text{rotor}}$ to inlet total enthalpy $h_T^{\text{inlet}}$ boundary condition using FD. . . . .	101
6.21	Adjoint-based sensitivity of stage total pressure ratio $\pi_{\text{stage}}$ to exit static pressure $p_s^{\text{exit}}$ boundary condition. . . . .	101
6.22	Verification of adjoint-based sensitivities of stage total pressure ratio $\pi_{\text{stage}}$ to exit static pressure $p_s^{\text{exit}}$ boundary condition using FD approximations. . . . .	102
6.23	Adjoint-based sensitivity of rotor efficiency $\eta_{\text{rotor}}$ to stator and rotor blade shapes in normal direction. . . . .	102

6.24	Verification of adjoint-based sensitivities of rotor efficiency to stator and rotor blade surface grid y-coordinates using FD. . . . .	103
6.25	Adjoint-based sensitivity of rotor efficiency $\eta_{\text{rotor}}$ to inlet tangential velocity $V_t^{\text{inlet}}$ boundary condition. . . . .	104
6.26	Verification of adjoint-based sensitivities of rotor efficiency $\eta_{\text{rotor}}$ to inlet tangential velocity $V_t^{\text{inlet}}$ boundary condition using FD. . . . .	104
6.27	Adjoint-based sensitivity of stage efficiency $\eta_{\text{stage}}$ to inlet total pressure $p_T^{\text{inlet}}$ boundary condition. . . . .	105
6.28	Verification of adjoint-based sensitivities of stage efficiency $\eta_{\text{stage}}$ to inlet total pressure $p_T^{\text{inlet}}$ boundary conditions using FD. . . . .	105
6.29	Adjoint-based sensitivity of outlet mass flow $\dot{m}^{\text{out}}$ to inlet total pressure $p_T^{\text{inlet}}$ boundary condition. . . . .	106
6.30	Verification of adjoint-based sensitivities of mass flow $\dot{m}^{\text{out}}$ to inlet total pressure $p_T^{\text{inlet}}$ boundary condition using FD. . . . .	106
6.31	Adjoint-based sensitivity of averaged total temperature of the rotor blade $(T_{Ta})_{\text{rotor}}$ to inlet total pressure $p_T^{\text{inlet}}$ boundary condition. . . . .	107
6.32	averaged total temperature of the rotor blade $(T_{Ta})_{\text{rotor}}$ to inlet total pressure $p_T^{\text{inlet}}$ boundary condition using FD. . . . .	107
7.1	Adjoint-based sensitivity of rotor and stage efficiencies to stage inlet total pressure $d\eta/dp_T^{\text{inlet}}$ . . . . .	112
7.2	Radial distribution of stage inlet total pressure and adjoint-based sensitivities of rotor and stage efficiencies to stage inlet total pressure profiles. . . . .	112
7.3	Adjoint-based sensitivity of rotor and stage efficiencies to respective inlet total pressure $d\eta/dp_T^{\text{inlet}}$ . . . . .	113
7.4	Adjoint-based sensitivity of rotor and stage efficiencies to stage inlet tangential velocity $d\eta/dV_t^{\text{inlet}}$ . . . . .	114
7.5	Comparison between adjoint-based sensitivity of stage and rotor efficiencies to inlet tangential velocity $\partial\eta_*/\partial V_t^{\text{inlet}}$ . . . . .	115
7.6	Adjoint-based sensitivity of rotor and stage efficiencies to stage inlet total enthalpy $d\eta/dh_T^{\text{inlet}}$ . . . . .	115
7.7	Comparison between adjoint-based sensitivity of stage and rotor efficiencies to inlet total enthalpy. . . . .	116

7.8	Adjoint-based sensitivity of rotor and stage efficiencies to respective inlet total enthalpy $d\eta/dh_T^{\text{inlet}}$ (normalized values). . . . .	117
7.9	Comparison between (normalized) adjoint-based sensitivity of stage and rotor efficiencies to inlet total enthalpy. . . . .	117
7.10	Adjoint-based sensitivities of stator and stage efficiencies ratio to stage exit static pressure, $d\eta/dp^{\text{exit}}$ . . . . .	118
7.11	Adjoint-based sensitivity of rotor and stage pressure ratios to stage inlet total pressure $d\pi/dp_T^{\text{inlet}}$ . . . . .	119
7.12	Adjoint-based sensitivity of rotor and stage pressure ratios to stage inlet tangential velocity $d\pi/dV_t^{\text{inlet}}$ . . . . .	120
7.13	Comparison between (normalized) adjoint-based sensitivity of stage and rotor pressure ratios to stage inlet tangential velocity. . . . .	121
7.14	Adjoint-based sensitivities of stator and stage pressure ratio to stage exit static pressure $d\pi/dp^{\text{exit}}$ . . . . .	121
7.15	Adjoint-based sensitivity of stage efficiency $\eta_{\text{stage}}$ to blade shape in normal direction. . . . .	123
7.16	Adjoint-based sensitivity of stage pressure ratio $\pi_{\text{stage}}$ to blade shape in normal direction. . . . .	124
7.17	Sensitivity of rotor and stage efficiencies to radial perturbation of hub mesh nodes $\partial\eta/\partial r_{\text{hub}}$ . . . . .	126
7.18	Sensitivity of rotor and stage pressure ratios to radial perturbation of hub mesh nodes $\partial\pi/\partial r_{\text{hub}}$ . . . . .	127
7.19	Hicks-Henne bumps imposed on the rotor hub. . . . .	128
7.20	Sensitivity of stage performance metrics to bumps height. . . . .	129
7.21	Variation of the stage performance metrics with the perturbation step parameter $\beta$ . . . . .	130
7.22	Radial displacement imposed on the rotor hub (in % of blade span). . . . .	131
7.23	Pressure field change due to the two bumps on the rotor hub at an XY plane centered at bump 1. . . . .	132
7.24	Pressure field change due to the two bumps on the rotor hub at an XY plane centered at bump 2. . . . .	133

7.25	Radial velocity change due to the two bumps in the rotor hub at an XY plane at 85% chord (normalized values). . . . .	134
7.26	Streamtraces along the rotor blade passage. . . . .	134
7.27	Radial velocity change due to the two bumps in the rotor hub at an XY plane behind blade (normalized values). . . . .	135
7.28	Adjoint-based sensitivity of rotor blade vane averaged total temperature $(T_{Ta})_{\text{rotor}}$ to stage inlet total pressure $p_T^{\text{inlet}}$ . . . . .	136
7.29	Adjoint-based sensitivity of rotor blade vane averaged total temperature $(T_{Ta})_{\text{rotor}}$ to stage inlet total enthalpy $h_T^{\text{inlet}}$ boundary condition. . . . .	136
7.30	Adjoint-based sensitivity of rotor blade vane averaged total temperature $(T_{Ta})_{\text{rotor}}$ to stage inlet tangential velocity $V_t^{\text{inlet}}$ boundary condition. . . . .	137

# Nomenclature

## Greek symbols

- $\alpha$  Vector of design variables.
- $\beta$  Engineering design parameter.
- $\gamma$  Heat capacity ratio.
- $\eta$  Efficiency.
- $\Lambda$  Continuous adjoint solution.
- $\lambda$  Bulk viscosity coefficient.
- $\mu$  Dynamic viscosity coefficient.
- $\mu_l$  Laminar viscosity coefficient.
- $\mu_t$  Eddy viscosity coefficient.
- $\pi$  Pressure ratio.
- $\rho$  Density.
- $\tau$  Viscous stresses.
- $\psi$  Discrete adjoint solution.
- $\omega$  Specific dissipation rate.

## Roman symbols

- $\mathcal{C}$  Optimization constraints.
- $c_p$  heat capacity at constant pressure.

$C_r$	Velocity radial direction cosine.
$c_v$	Heat capacity at constant volume.
$C_z$	Velocity Axial direction cosine.
$E$	Total energy.
$e$	Specific internal energy.
$\mathcal{F}$	Flux in computational space.
$\mathbf{f}$	Discretized flux.
$\mathbf{f}$	Flux vector.
$\mathbf{F}$	Mixing-plane flux.
$h$	Enthalpy.
$\mathcal{I}$	Objective function.
$k$	Turbulence kinematic energy.
$\dot{m}$	Mass flow.
$\mathbf{p}$	Mixing-plane profile.
$p$	Pressure.
$\mathbf{q}$	Vector of stage variables.
$\mathbf{R}$	Discrete residual.
$\mathcal{R}$	Residual of flow governing equations.
$R$	Ideal gas constant.
$s$	Entropy.
$T$	Temperature.
$\mathbf{U}$	Vector of boundary conditions.
$\mathbf{u}$	Velocity vector.

$V_t$  Tangential velocity.  
 $\mathbf{X}$  Vector of mesh grid coordinates.

### **Subscripts**

$c$  compressor.  
don Relative to the donor row.  
 $i, j, k$  Computational indexes.  
local Relative to the local row.  
 $m$  mixed.  
 $n$  Normal component.  
rec Relative to the receiver row.  
 $T$  Total quantity.  
 $Ta$  Total averaged quantity.  
 $t$  turbine.  
 $u$  unmixed.  
 $x, y, z$  Cartesian components.

### **Superscripts**

\* Updated state solution.



# Glossary

<b>AD</b>	Automatic Differentiation.
<b>CAD</b>	Computer Assisted Design
<b>CFD</b>	Computational Fluid Dynamics.
<b>CS</b>	Complex Step.
<b>FD</b>	Finite Differences.
<b>GB</b>	Gradient Based
<b>GF</b>	Gradient Free
<b>GMRES</b>	Generalized Minimal RESidual algorithm.
<b>IATA</b>	International Air Transport Association
<b>JST</b>	Jameson-Schmidt-Turel
<b>MPI</b>	Message Passing Interface
<b>NASA</b>	National Aeronautics and Space Administration
<b>PDE</b>	Partial Differential Equation
<b>RANS</b>	Reynolds-Averaged Navier-Stokes.
<b>SST</b>	Shear Stress Transport
<b>VKI</b>	von Karman Institute for Fluid Dynamics



# Chapter 1

## Introduction

This section will introduce the motivation for the work presented in this thesis, followed by an introduction to the various areas it encompasses. A review of the state-of-the-art in adjoint-based sensitivity analysis with applications to turbomachinery is presented next. Lastly, the expected contributions to the state-of-the-art resulting from the work here presented are described.

### 1.1 Motivation

Turbomachines transfer energy either to or from continuously flowing fluid by the dynamic action of one or more moving blade rows. Their name originates from the Latin word *turbo*, which means "that which spins". As schematically represented in figure 1.1, they are major components of aircraft, marine, space and land propulsion systems (blue); hydraulic, gas and steam turbines for energy production (green); industrial pipeline and processing equipment such as gas, petroleum, and water pumping plants (black); and a wide variety of other applications (e.g. heat-assist pumps, industrial compressors, refrigeration plants, etc).

After many decades of research in the field of jet propulsion, modern turbomachinery (and specially modern turbofan engines) are at a stage where it is very difficult to obtain further improvements. However, given the high number of turbomachine components in operation, particularly aircraft engines, a small increase in efficiency could result in a large global gain, being it in cost reduction or pollution reduction. Current predictions indicate an increase in the growth of these numbers.

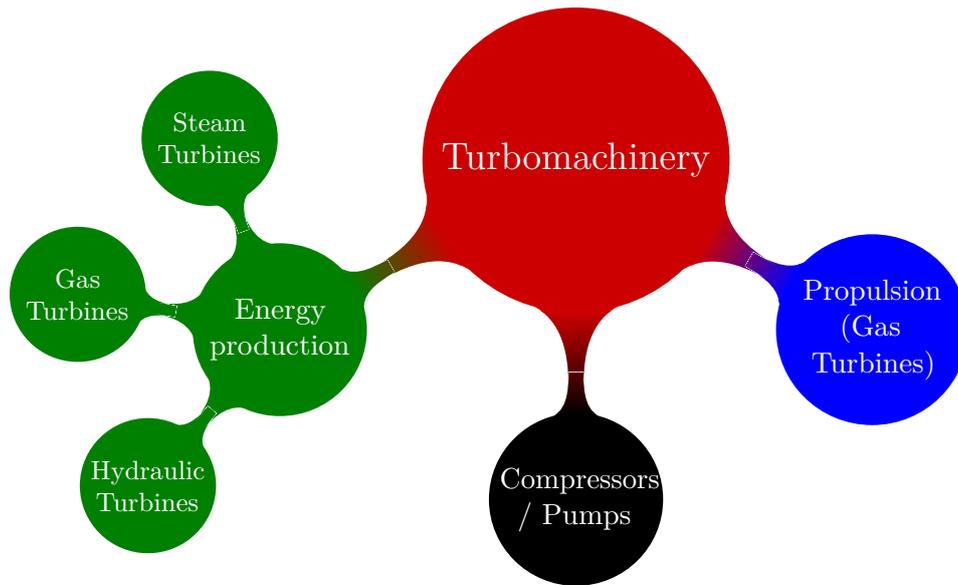


Figure 1.1: Schematic of the various areas of application of turbomachinery.

According to the International Air Transport Association's (IATA) 20-Year Air Passenger Forecast [1], 7.8 billion passengers are expected to travel in 2036, a near doubling of the 4 billion air travelers flown in 2017. The prediction is based on a 3.6% average Compound Annual Growth Rate noted in the release of the latest update to the association's 20-Year Air Passenger Forecast. According to Alexander de Juniac, CEO and Director General of IATA:

*"All indicators lead to growing demand for global connectivity. The world needs to prepare for a doubling of passengers in the next 20 years. It's fantastic news for innovation and prosperity, which is driven by air links. It is also a huge challenge for governments and industry to ensure we can successfully meet this essential demand".*

The report entitled "*Flightpath 2050 Europe's Vision for Aviation*" and published by the Advisory Council for Aeronautics Research in Europe [2] established a set of targets to be met by 2050 (relative to the capabilities of typical new aircraft in 2000). Amongst the targets are the following:

- Reduction of CO<sub>2</sub> by 75% per passenger kilometer;
- Reduction of NO<sub>x</sub> emissions by 80%;
- Reduction of perceived noise emission of flying aircraft by 65%.

As such, there is much interest and research being conducted on ways to make aircraft more efficient, not only to make air travel less expensive and more generalized, but also to comply to the stricter pollution rules.

With this continued increase in number of flights, so as the global consumption of fuel, with the total worldwide fuel consumption of commercial airlines in 2017 being around 90 billion ( $90 \times 10^9$ ) gallons [3]. The cost of fuel to an airline has a big impact on its profit margin, as it typically represents more than 20% of their total operating costs [3]. Figure 1.2 presents the global airline industry’s net profits and fuel costs, where an inverse trend between the two is visible. This relatively large fraction of the total expenses in fuel,

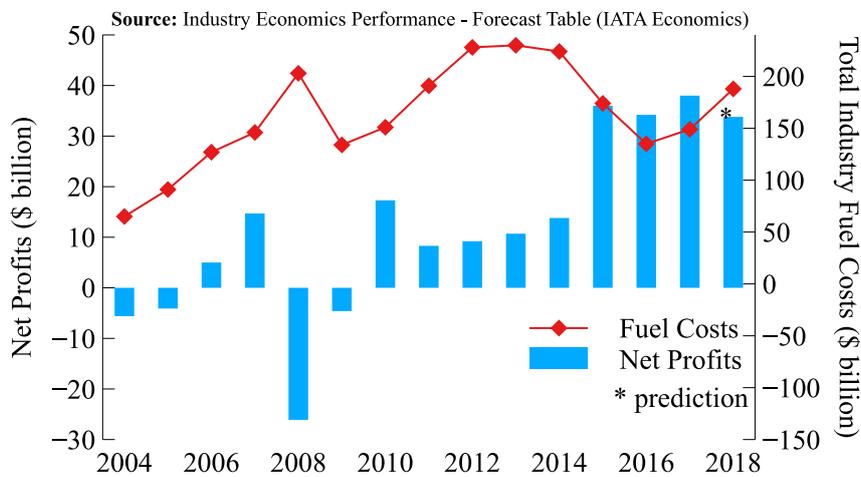


Figure 1.2: Total global airline industry’s fuel costs and net profits.

and particularly the differences of magnitude between fuel costs and net profits, makes the increase of efficiency (and consequent reduction of fuel costs) very desirable in economic terms. As an exercise, assuming a particular aircraft model, we can predict the economic impact that an increase of 1% in engine efficiency would have on a hypothetical airline. Assuming the operating costs of a Boeing 757-200, presented in table 1.1 for an average daily utilization of 11.3 block-hours, the savings obtained from the efficiency increase (represented by a decrease of 1% in fuel cost) would be of around \$2.2M annually, if we consider a fleet of 100 aircraft. Particularly for smaller airlines, where the low profit margins maintain the companies very close to the break-even point, such saving could potential help the company cross that threshold. Such increase in efficiency is not easily achievable, and requires increases in efficiency in many of the components of the engine. Nonetheless, this hypothetical results emphasizes the importance of improving the performance of turbomachines (jet engines, in this particular case) in terms of economical benefits.

<b>Fuel Cost</b>	<b>Total Cost</b>	<b>Fuel Cost</b>	<b>Total Cost</b>	<b>Fuel Cost</b>	<b>Total Cost</b>
<b>per block hour</b>		<b>per day</b>		<b>per year</b>	
\$548.00	\$2,550.00	\$6,192.40	\$28,815.00	\$2,204,494.40	\$10,258,140.00
\$542.52	\$2,544.52	\$6,130.48	\$28,753.08	\$2,182,449.46	\$10,236,095.06
<b>Savings:</b>					\$22,044.94
(for 100 aircraft)					\$2,204,494.40

Table 1.1: Hypothetical savings of increasing fuel efficiency of a Boeing 757-200 on a fleet of 100 aircraft.

Additional future savings from increased efficiency are also expected when carbon taxes [4] become a reality since the carbon emissions are proportional to the jet fuel consumption, which is a particular type of hydrocarbon molecules. However, these are presently hard to quantify as the international legislation is still being outlined.

The demand for improvement on turbomachinery efficiency does not come only from the aeronautical industry but also from the energy production industry, where higher efficiencies are also required in order to comply to all the directives established for the future. The "*Roadmap on Turbomachinery Research 2014 – 2020*" by EUTurbines [5] states that more than 80% of the electricity generated worldwide is produced by gas and steam turbines being applied at both coal, nuclear, gas, biomass and solar-thermal power plants. The same report also mentions that the International Energy Agency estimates that in order to deliver a 50% of global emission reduction by 2050, 24% will need to come from end use fuel efficiency, 12% from end use electricity efficiency and 7% from power generation efficiency. Some measures to improve turbine efficiency are also mentioned: increasing turbine inlet temperature and compressor pressure ratio in parallel with cooling air reduction, more advanced aerodynamic concepts and loss reduction.

These demands and predictions highlight the necessity to use all the tools available to study and improve the performance and efficiency of turbomachines.

## 1.2 Overview of Turbomachines

As previously mentioned, turbomachines transfer energy either to or from a continuously flowing fluid by the dynamic action of one or more moving blade rows. Depending on the direction of the flow of energy, we can have a compressor (imparts energy to the fluid) or

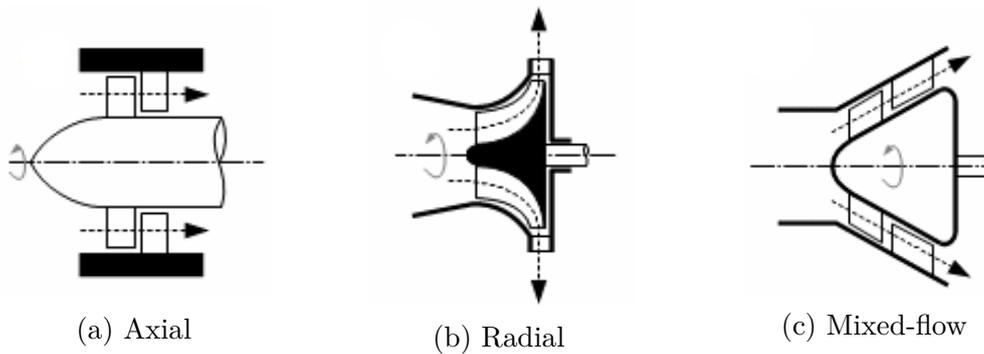


Figure 1.3: Classification of the turbomachines by the stream direction in relation to the axis of the shaft [6].

a turbine (removes energy from the fluid).

### 1.2.1 Types of turbomachines

Turbomachines can also be classified according to the path the flow follows, relative to axis of rotation, as it runs through it. They can be either *axial*, *radial* or *mixed-flow* (see figure 1.3). If the meridional flow path is axial, we have an axial turbomachine. If the flow path is predominantly radial, we have a radial (or centrifugal) turbomachine. The mixed-flow turbomachine has, like the name implies, a flow path that is partially axial and partially radial. The fluid that flows through a turbomachine also affects its denomination. As an example, a compressor of the axial type is called an *axial flow compressor*. If the same machine used liquid as the moving fluid, it would be called an *axial flow pump*.

### 1.2.2 Flowpath and Thermodynamics

The simplest gas turbine is composed of a compressor, a combustion chamber and a turbine. The thermodynamic processes that the fluid is subjected to as it goes through this simple gas turbine can be represented by the schematic and idealized thermodynamic cycle (also known as the Brayton cycle [7]) presented in figure 1.4. The compressor first imparts energy into the fluid by changing the stagnation (or total) enthalpy, kinetic energy and stagnation pressure of the fluid (1-2). In the combustion chamber, energy is added to the flow by means of heat transfer from the combustion of the fuel (2-3). The turbine then extracts energy from the fluid and typically transfers a part of it to the compressor by means of a rotating shaft (2-4). In a jet engine, this work performed on the rotating shaft

---

<sup>1</sup>Image by Duk [8]

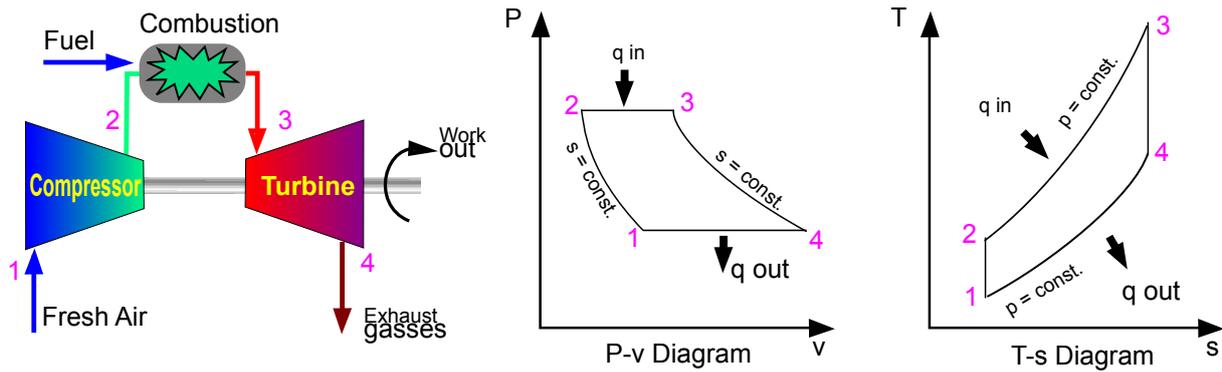


Figure 1.4: Schematic of simple turbomachine and respective thermodynamic cycle <sup>1</sup>.

is the minimal to keep the compressors and other rotating parts spinning. The energy that is not used for this leaves the turbine as kinetic energy (of the exhaust gases), producing thrust.

Most modern turbomachines are composed of a series of compressors and/or turbines. Each of these individual compressors or turbines is typically designated as a *stage* and it consists in two rows with blades on its circumference, one that rotates (rotor) and other that is fixed to the casing (stator). The stationary blades can also be called guide vanes and, when located at the beginning of a series of compressor stages or at the end of a series of turbine stages are denominated *inlet guide vanes* or *exit guide vanes*, respectively.

In figure 1.5 we can see a schematic representation of a typical turbofan, the most common type of commercial jet engines. Traveling downstream, starting from the inlet, the air is first accelerated by the fan, whose rotation is typically maintained by the low-pressure shaft. A great part of the flow, called bypass flow, is directed outside of the engine. The rest of the flow, called core flow, is then compressed by the low-pressure compressor (also rotating with the low-pressure shaft). The last step before entering the combustion chamber is another compression performed by the high-pressure compressor. In the combustion chamber, the flow is energized by the combustion of the fuel. The flow then leaves the combustion chamber into the high-pressure turbine, which expands the flow and, in doing so, providing work into the high-pressure shaft. Before leaving the engine through the nozzle, the low-pressure turbine stages further expand the flow, providing work to the low-pressure shaft.

<sup>2</sup>image by K. Aainsqatsi [9]

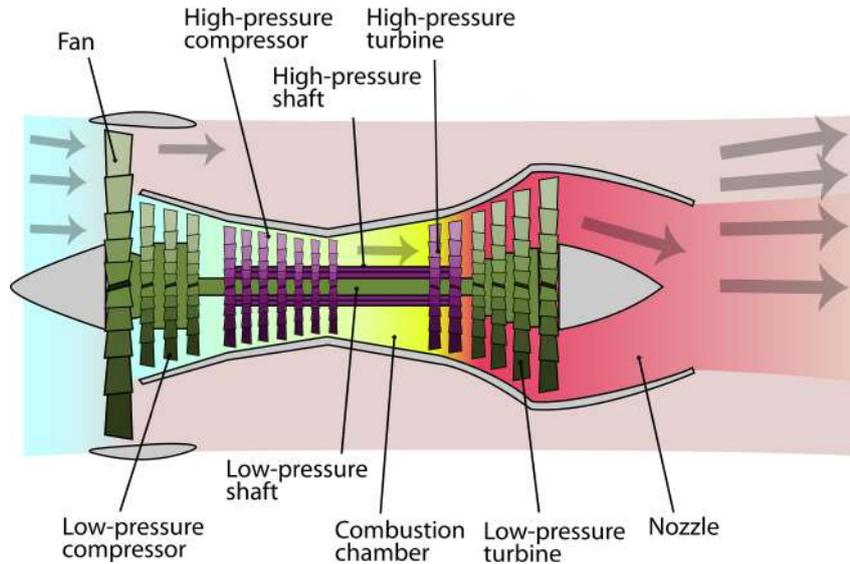


Figure 1.5: Schematic of a typical turbofan

2

### 1.2.3 Secondary Flows

The flow inside turbomachine components is highly three-dimensional and unsteady, particularly in high pressure compressors with high blade loading and small aspect ratio blades [10]. However, even in low pressure turbine components [11], the endwall (tip and hub) regions can be affected by 3D phenomena such as the tip clearance vortex or the horseshoe vortex, amongst others, that are responsible for the loss of efficiency. These phenomena developed in high-pressure compressor blades are represented in figure 1.6.

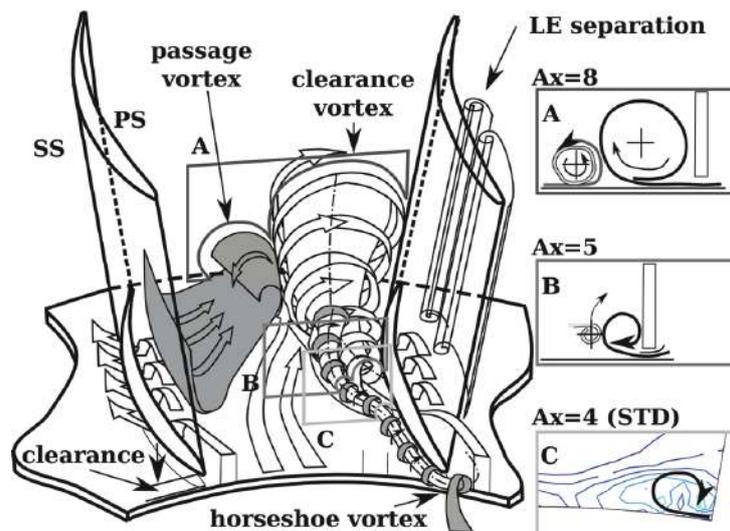


Figure 1.6: Representation of secondary flow phenomena in high-pressure compressor blades (figure by Beselt, Eck, and Peitsch [10]).

In steady-state simulations, albeit not capturing these phenomena in detail, the endwall

is still a very sensitive region, as it is where the two boundary layers, from the hub/casing and blade, merge.

As such, by properly controlling the flow in these critical regions, the losses can be minimized. Legacy designs have relied on the expertise of the designers to handle such challenges, but the usage of numerical design tools can open the way to further explore large sets of interacting design parameters whose simultaneous handling might proved too difficult if done manually.

### 1.3 Numerical Optimization

A field of research that has seen much growth in the past decades, partly due to the increasing available computational power is the use of external and internal flow simulations using high-fidelity Computational Fluid Dynamics (CFD) models. They have become a routine, with the emerging trend being to use optimization techniques as part of the design process, both in academia and industry [12]. Given the nature of the flow models, a numerical simulation may take hours or even days to complete, as larger and larger problems keep being considered. An optimization, which may require hundreds of function calls to find an optimum, may therefore lead to a prohibitive time requirement. The choice of optimization algorithm is therefore very important if results are to be obtained within a realistic timeframe.

In a typical CFD optimization problem, the objective is to minimize (or maximize) a performance metric (or a set of metrics) which are defined by some design variables and by the solution of a set of governing equations. This can be defined mathematically as

$$\begin{aligned}
 & \text{minimize } \mathcal{I}(\boldsymbol{\alpha}, \mathbf{q}(\boldsymbol{\alpha})) \\
 & \text{w.r.t. } \boldsymbol{\alpha}, \\
 & \text{subject to } \mathcal{C}(\boldsymbol{\alpha}, \mathbf{q}(\boldsymbol{\alpha})) = 0, \\
 & \mathcal{R}(\boldsymbol{\alpha}, \mathbf{q}(\boldsymbol{\alpha})) = 0,
 \end{aligned} \tag{1.1}$$

where  $\mathcal{I}$  is the set of objective functions (performance metric) to be minimized,  $\boldsymbol{\alpha}$  is the set of design variables,  $\mathbf{q}$  is the solution obtained from solving the governing equations represented by  $\mathcal{R}$ . A set of constrains is usually considered, and are here represented by  $\mathcal{C}$ . In turbomachinery, the typical performance metrics are efficiency, mass flow, pressure

ratio and the temperature at a certain region of the component in analysis, amongst others. The design variables usually consist in parameters that define the geometry of the blade or operating conditions. The optimization is typically constrained by certain geometry parameters or performance metrics such as the mass flow, which one might want to maintain, while improving other metrics.

When choosing a numerical optimization method to solve the problem above [13], one can select from two main categories: deterministic and heuristics (see figure 1.7). Heuristic

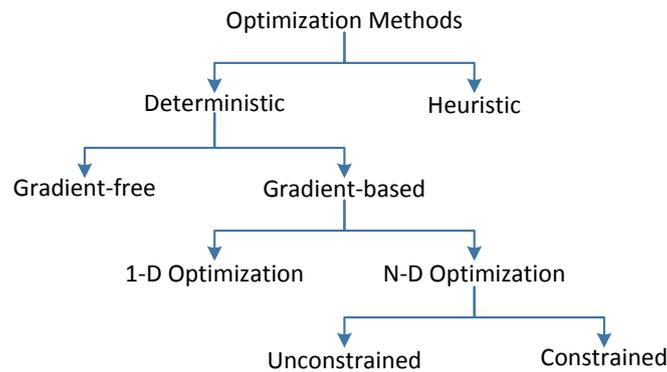


Figure 1.7: Schematic representation of the various categories of optimization algorithms.

methods make use of concepts found in nature to find the global optimal solution. This category includes methods such as Genetic Algorithms [14], Ant Colonies [15], Differential Evolution [16], Particle Swarm [17], etc. Deterministic methods take advantage of the analytical properties of the problem to generate a sequence of points that converge to a local optimal solution. These can be Gradient-Free (GF) and Gradient Based (GB). The first type of algorithms, like the name indicates, do not require the computation of the gradient (or Hessian), needing only, much like the heuristic methods, to evaluate the prescribed objective function a certain number of times. Examples of GF algorithms are the Simplex [18], Golden search [19] or the Hooke-Jeeves [20]. Also much like the heuristic methods, albeit being able to find (or approximate) the global minimum of a function, and being usually the better choice for discrete search or discontinuous/non-convex search spaces, they tend to require a very high number of those same function evaluations, which can lead to the prohibitive time requirements mentioned above. GB algorithms, on the other hand, use the information of the gradient or the Hessian of the objective function (and/or constraints) to chose the best direction in the design space

that leads to the optimal solution. This leads to much less function evaluations required to achieve the closest minimum. If the objective function is known to be smooth and the design variables are continuous, the GB algorithms are typically the best choice. The GB class contains *unconstrained* (Steepest Descent [21], Conjugate Gradient [22], quasi-Newton [23, 24], etc) and *constrained* (Feasible Direction [25], Reduced Gradient [26], SQP [13], etc) algorithms, depending or not if they deal with the constraints of the optimization problem. The difference in number of function calls between heuristic and GB algorithms is evidenced in figure 1.8, where the optimization history of a function containing a local and a global minimum is presented using a GB (figure 1.8a) and a Heuristic algorithm (figure 1.8b). While the GB algorithm was able to find the closest minimum with a relative low number of function evaluations, it could only find the minimum that was closest to the initial guess (this can be avoided to some point by the use of multiple initial guesses). The Heuristic method was able to find the global minimum without being stuck in the local one, evidencing that for global optimization of a function, it may be the best option. However, it required much more function evaluations to find the optimum than the GB algorithm, which means that these algorithms are in general restricted to higher fidelity computations, where the time requirements are higher. A conjugation of both algorithms

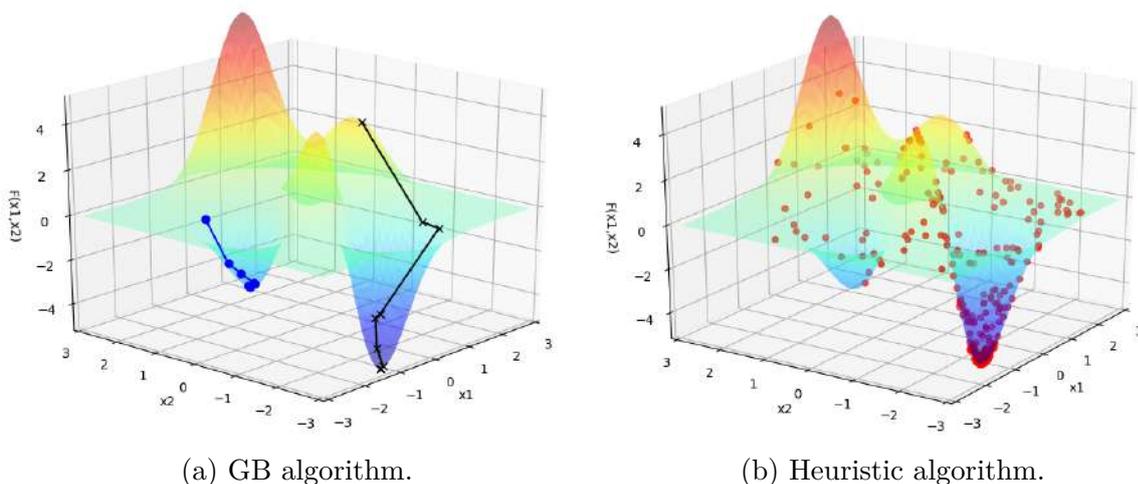


Figure 1.8: Optimization history of a function with local and global minima.

is a typical approach, where the design space is (roughly) globally optimized using an Heuristic method, with the global optimum return by the optimizer being used as the starting guess for the GB algorithm. Other method used with GB algorithms, to avoid being stuck on a local minimum is the use of multiple starting points distributed in the design space.

Typically, in the preliminary design phase of a turbomachine (or even an aircraft), low fidelity models (with low computational requirements) are used to allow the designer to experiment various configurations and parameters to achieve a first iteration of the design that is compliant with the requirements defined a-priori to be the goals for the turbomachine. In this phase, the use of heuristic and gradient-free methods is advantageous, as they allow a much broader search of the design space, and often the design variables considered at this initial phase are discrete. It is after the preliminary design phase that high-fidelity models (with high computational requirements) are used to fine tune the detailed designs to reduce unwanted behaviors of the flow that lead to performance losses. In this case, the efficiency of the GB algorithms provides a strong advantage and are typically the chosen type of optimization algorithms.

A GB algorithm for unconstrained optimization of a smooth and continuous function can be described as follows:

1. **Initial guess:** The iteration starts with number  $k = 0$  at a starting point  $\mathbf{x}_0$ ;
2. **Test for convergence:** If the convergence conditions are satisfied, the iterative procedure is stopped and  $\mathbf{x}_k$  is the optimal solution;
3. **Compute the search direction:** Compute the vector  $\mathbf{p}_k$ , which defines the direction in  $n$ -space where the algorithm will search for the minimum;
4. **Compute the step length:** Find a positive scalar,  $\alpha_k$ , such that  $\mathcal{I}(\mathbf{x}_k + \alpha_k \mathbf{p}_k) < \mathcal{I}(\mathbf{x}_k)$ ;
5. **Update the design variables:** Set  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$  and  $k = k + 1$ ; Go back to step 2.

Two subproblems can be identified in each iteration of this algorithm:

- Computing the search direction  $\mathbf{p}_k$ ;
- Finding the optimal step size  $\alpha_k$ .

One of the simplest GB algorithms, the Steepest Descent, uses the gradient of the function,  $\mathbf{g}_k = \nabla \mathcal{I}(\mathbf{x}_k)$ , as the search direction ( $\mathbf{p}_k = -\mathbf{g}_k / \|\mathbf{g}_k\|$ ). Others, such as the Conjugate Gradient or quasi-Newton algorithms, also use the previous history of the gradients to compute the search direction at each iteration. Regardless of the selected GB algorithm, they all require one computation of the gradient of the function per iteration. The second

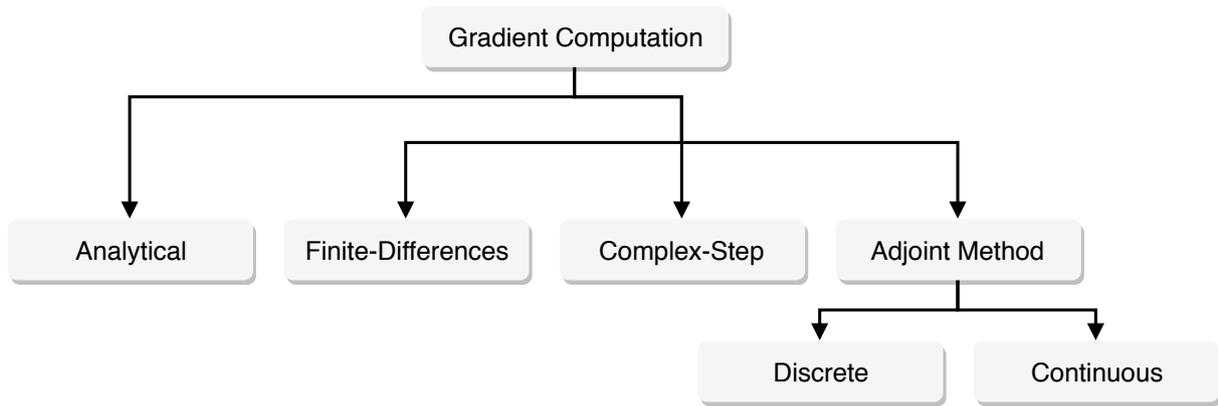


Figure 1.9: Different methods for computing the gradient of a function.

problem, finding the optimal step  $\alpha_k$  is an optimization problem on its own. The search direction obtained from the gradient is a *descent direction*, guaranteeing that  $\mathcal{I}$  can be reduced by stepping along that direction. Ideally a global minimum of  $\mathcal{I}(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$  would be found, with respect to  $\alpha_k$ , but this can greatly increase the computational (even finding the local minimum can be too computationally expensive) effort and usually an *inexact* line search is used to achieve adequate reductions of  $\mathcal{I}$  at reasonable computational costs.

The fact that GB methods use the gradient (also known as derivative or sensitivity) to move the solution towards the optimum in the design space, making them more efficient, requires attention to the method of computation of the gradient information known as sensitivity analysis methods. If the number of design variables is very large, the use of traditional methods such as the finite-differences approximation can lead to prohibitive computational time requirements. Since a turbomachinery optimization case can have up to thousands of design variables, efficient gradient computation methods are a necessity.

Figure 1.9 presents various methods for computing the gradient of a function. The first is the analytical methods. If the function to which the gradient is required can be analytically differentiated [27], then that is the most efficient choice, as it computes the derivative directly. This is however not the case for the majority of the cases, as typically the functions are based on solutions computed from the iterative convergence of non-linear systems of equations. The Finite-Difference (FD) [27] approximation method offers often a simple way of computing the derivative of a set of objective functions relative to an independent variable. As it is only able to produce the derivative to a single variable for each function call, its computational cost is proportional to the number of independent variables to analyze. This, allied to the fact that each function call may take several hours

(or even days), makes this approach unappealable. FD approximations also suffer from a high sensitivity to the perturbation step, which is subjected to errors due to subtractive cancellation. The Complex-Step (CS) method [28–30] is similar to the FD method, but instead of perturbing the selected independent variable with a real perturbation, it uses a complex perturbation. It does not suffer from subtractive cancellation and therefore much smaller steps can be used, avoiding the problem of sensitivity to the perturbation step inherent in the FD approximation. The computational cost of using such method is, like FD, proportional to the number of design variables, and as such its use in the numerical optimization of problems with a high number of design variables (as compared to the number of functions of interest) is limited.

The adjoint method is able to produce exact derivatives with a computational cost that is nearly independent of the number of design variables. It works by applying control theory to Partial Differential Equations (PDE) [31]. Depending on the approach taken, continuous or discrete, the control theory is either applied directly to the equations, resulting in adjoint equations which are then discretized or applied to already discretized PDEs (see figure 1.10). Both approaches have their advantages and disadvantages. While the continuous approach

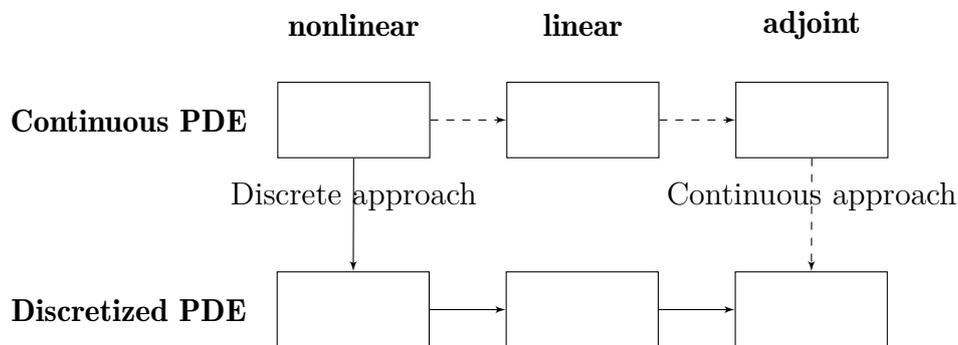


Figure 1.10: Alternative approaches to obtain the discrete adjoint equations.

can be simpler to implement, and typically requires less computational time and memory than a discrete implementation, the discrete approach is naturally consistent with the direct solver. This is highly advantageous when using the adjoint solver in gradient-based optimization, as it usually increases convergence. The discrete approach also allows the use of Automatic Differentiation (AD) tools [32, 33], in an approach that mixes both the advantages of the discrete adjoint with the fast development time allowed by automatic differentiation.

The adjoint method has a wide span of applications, ranging from oceanography and

geology to computer graphics simulation [34]. In CFD, due to the nature of the governing equations and size of the typical numerical problems, the adjoint method has seen an extensive range of applications in the previous decades. The introduction of the adjoint method to the field of fluid mechanics was done by Pironneau [35]. Jameson further extended the method to optimization of airfoil profiles [36] and wings [37]. More recently it has been used in solving multi-point aerodynamic shape [38, 39] and aero-structural [40, 41] optimization problems, magneto-hydrodynamic flow control [42] and turbomachinery blades [43]. It has recently also been used in flow visualization by capturing the relative importance of different flow regions with respect to a quantity of interest [44].

A typical adjoint-based optimization/analysis framework is represented in figure 1.11. First, using a set of parameters,  $\boldsymbol{\alpha}$ , that describe a geometry, a computational mesh,  $\mathbf{X}$ , is created. This mesh, along with a set of boundary conditions and operating conditions, is used in a flow solver to compute a converged solution,  $\mathbf{q}$ , to the partial differential equations governing the flow. The adjoint equations are then solved to obtain the adjoint solution  $\boldsymbol{\psi}$ , relative to a specific objective function  $\mathcal{I}$  (or performance metric). With the adjoint solution, the gradient of the selected metric of performance to the set of design parameters is easily computed. The adjoint-based gradient is often computed relative to the numerical mesh grid,  $\mathbf{X}$ . In that case, an extra step needs to be taken to compute the gradient relative to the design parameters,  $\boldsymbol{\alpha}$  by multiplying the adjoint-based gradients with the sensitivity of the mesh-grid to the design parameters (bottom equation of the gradient box in figure 1.11). This gradient can either be used by a numerical optimizer to change the design parameters and, following the previous steps iteratively, arrive to an optimal solution, or by the designer to gain insight of how the performance metric in study is influenced by the design parameters.

## 1.4 Adjoint-based Sensitivity Analysis in Turbomachinery

One of the earliest applications of the adjoint method to the optimization of turbomachinery was done by Yang and Liu [45], in 2003. Using the continuous approach, the adjoint-based sensitivity information was used in the inverse design of two-dimensional cascade blades in an inviscid flow environment. In 2004 Chung, Lee, and Martin [46] presented the discrete

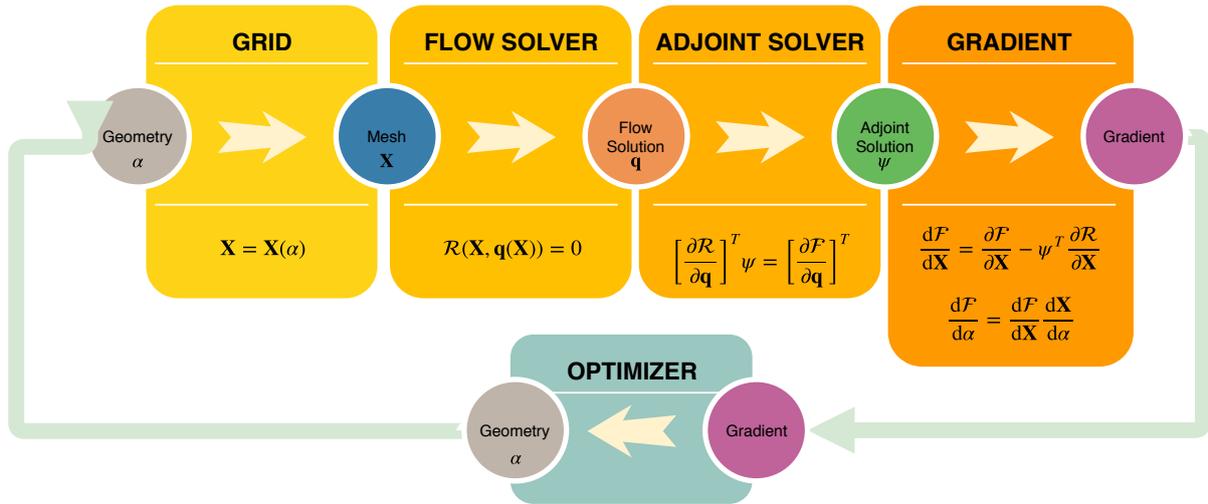


Figure 1.11: Representation of a typical adjoint-based sensitivity analysis framework.

adjoint formulation for a three-dimensional Euler solver and applied the adjoint solver to the inverse design of NASA’s Rotor 37 blade geometry. In 2005, Wu, Liu, and Tsai [47] used the continuous adjoint-based sensitivities on the constrained optimization of the VKI turbine stator, and on the Standard Configuration 4 turbine. In the same year, Arens, et al. [48] presented results on the optimization of turbine blades using the continuous approach on the 2D Euler equations. The continuous adjoint method was also used in the optimization of 2D shape of a turbine blade, combined with the quasi-Newton [49] and in compressor and turbine blade design for two- and three-dimensional flows, both inviscid and viscous, by Papadimitriou and Giannakoglou [50, 51]. The same authors later successfully minimized total pressure loss in both compressor and turbine cascades, while maintaining flow turning and blade thickness, using the continuous adjoint approach [52]. Luo, et al. [53] managed to apply the continuous viscous adjoint equations to the reduction of secondary loss of low-aspect-ratio turbine blades by using the gradient information to modify stagger angle, blade shape and endwall profile.

Marta, Shankaran, and Stein [54] firstly implemented the discrete adjoint solver for a legacy turbomachinery CFD solver using the ADjoint approach. The same authors tested an adjoint-based design framework on shape optimization of turbomachine blades using a set of Hicks-Henne bump functions superimposed on the baseline shape as design variables. The adjoint solution was also shown to provide insights into the nature of changes the designer could induce to cause improvement in the performance metric of interest [55]. The same adjoint solver was later used to compute sensitivity of aerothermal performance

parameters to blade geometry and inlet/exit boundary conditions of a single row of a compressor [56].

Mueller and Verstraete [57] used automatic differentiation to develop an adjoint-based framework that uses a tailored shape parameterization to satisfy geometric constraints due to mechanical and manufacturing requirements while maintaining the shape in a Computer Aided Design (CAD) representation.

While the previous mentioned works dealt with the optimization/sensitivity analysis of single rows of turbomachinery components. Turbomachines are typically composed of many rows, with each pair of stator/rotor rows defining a single stage. On a multi-stage turbomachine, as the flow is subsonic, there is a strong interaction between the various rows, both in upstream and downstream directions. The individual optimization of a single row would then most likely lead to a design that is less than optimal when combined with the rest of its rows.

Multistage turbomachinery numerical analysis had been performed for many years, with the steady mixing-plane [58, 59] approach often being the selected method for the coupling of the various rows.

Frey, Kersken, and Nurnberger [60], Wang and Li [61, 62] and Walther and Nadarajah [63, 64] presented adjoint solvers which allow multi-row optimization. Frey, Kersken, and Nurnberger used finite-differences to obtain the derivatives to set-up the discrete adjoint system of equations. Their selected mixing-plane approach used was based on Gile's exact two-dimensional non-reflecting boundary conditions [65]. Following the continuous approach, Wang, et al. [62] and Wang and He [66] developed a multi-row capable turbomachinery adjoint solver using a conservative adjoint mixing-plane approach, coupling averaged co-states based on one-dimensional characteristics. The solver was applied to the numerical optimization of various cases. Following the discrete approach, Walther and Nadarajah manually differentiated the discrete equations of a turbomachinery Reynolds Averaged Navier-Stokes (RANS) solver, proposing a framework for fully-automated constrained aerodynamic shape optimization. The coupling of the multiple rows was considered using the mixing-plane formulation. More recently, Backhaus, Engels-Putzka, and Frey [67] developed a multi-row capable adjoint solver using an operator-overloading AD tool to implement the adjoint solver.

## 1.5 Contributions to the State-of-the-art

This thesis describes the formulation, implementation and application of the adjoint of the mixing-plane interface of a legacy turbomachinery CFD solver. It follows the previous work of Marta and Shankaran [56] on the implementation of the discrete adjoint counterpart of a proprietary turbomachinery CFD solver, by using a source transformation AD tool on the direct routines. The improved adjoint solver is used to obtain sensitivity analysis of various functions of interest, such as pressure ratio, efficiency, mass flow and maximum or averaged total temperatures, to both the hub and blade shapes and to the inlet and exit boundary conditions of a stator-rotor turbomachinery stage.

With the adjoint mixing-plane interface, the adjoint solver will be able to obtain sensitivity analysis of multi-row turbomachinery cases of study, where the appropriate coupling among the several blade rows is taken into account. In addition, the functions of interest will be extended to represent not only single blade row metrics but also full stage metrics (with multiple rows).

Summarizing, the work presented in this document introduced the following contributions to the state-of-the-art:

- A formulation for handling coupled adjoint systems of equations in multi-row turbomachinery simulations;
- A detailed description of how to handle the differentiation of complex industrial codes, with features that are difficult (or impossible) to differentiate in the process of automatic differentiation;
- An improved adjoint solver, capable of handling multiple blade rows in the sensitivity analysis of arbitrary functions of interest with regard to boundary conditions or mesh geometry.

The implemented adjoint-mixing plane will provide the users with a large set of sensitivity information of multi-row turbomachinery problems, which will allow the study of new geometries and/or operating conditions that will hopefully translate into more efficient and better performing turbomachines.

## 1.6 Thesis Outline

This document began with an introduction to the motivation behind the work here presented, followed by an introduction to the subject of numerical optimization, particularly in turbomachinery. A brief description of the adjoint method was also presented together with a review of the various contributions to the subject in the past.

The description of the theoretical background for CFD analysis of multi-row turbomachine components is contained in **Chapter 2**.

**Chapter 3** presents a detailed description of the various sensitivity analysis techniques, with their advantages and disadvantages, and practical examples of application. The reasoning for choosing a hybrid technique on the development of the work described in this document is also highlighted.

**Chapter 4** presents the derivation of the continuous and discrete adjoint equations for the 2D Euler equations, with the purpose to illustrate the adjoint formulation of flow governing partial differential equations, that constitute the core of the adjoint solver used in the simulations presented in this work.

The core of the present document is included in **Chapter 5** in which the adjoint formulation of the mixing-plane algorithm is presented. It contains a detailed description of all the process, from the conceptual formulation to the numerical implementation, with emphasis on the required modifications to the original code of the legacy CFD solver on which the adjoint mixing-plane interface was implemented to allow for AD of the rewritten code.

Following the numerical implementation, **Chapter 6** presents the numerical verification of the implementation by comparing the adjoint-based sensitivities of various performance metrics to an assortment of independent parameters with FD approximations.

In **Chapter 7**, the adjoint-based sensitivity analysis results of a multi-row stator/rotor low pressure turbine stage is presented. The chapter is divided into two parts, sensitivity to boundary conditions, sensitivity to computational mesh.

The document ends with final conclusions and description of possible future work that could be done to further improve the delivered capabilities in **Chapter 8**.

# Chapter 2

## Numerical Analysis of Turbomachinery

The numerical analysis of turbomachinery can range from simple one-dimensional analysis of the thermodynamic cycle to complex three-dimensional modeling of the flow, heat transfer and chemical reactions that occur through the various components of a turbomachine. In this chapter, the basic thermodynamic relations are reviewed first and then the simplified Euler equations in two dimensions are introduced followed by the Reynolds Averaged Navier-Stokes equations.

### 2.1 Thermodynamic Relations

The thermodynamic processes that the flow experiences as it goes through a gas turbine can be represented by the Joule-Brayton cycle, assuming a continuous flow and that the heat input takes place at constant pressure. This cycle, which was introduced in the previous chapter and is schematically represented in figure 1.4, assumes flow behaves like a calorically perfect gas with constant specific heat coefficient at constant pressure  $c_p$  and constant volume  $c_v$  and constant specific heat ratio  $\gamma = c_p/c_v$ . This is a reasonable assumption for aerodynamic problems, and, as such, it is maintained throughout this thesis.

For an ideal gas, the relation between pressure  $p$ , density  $\rho$  and temperature  $T$ , is given by the specific gas constant,  $R$ , as

$$p = \rho RT . \tag{2.1}$$

Specific internal energy can be defined as a function of density, pressure and  $\gamma$  as

$$e = \frac{1}{\gamma - 1} \frac{p}{\rho}. \quad (2.2)$$

Since total specific energy  $E$  is the sum of specific internal energy and kinetic energy, it yields

$$E = e + \frac{1}{2} |\mathbf{u}|^2, \quad (2.3)$$

where  $\mathbf{u}$  is the absolute velocity vector and  $|\mathbf{u}|^2 = (u_x^2 + u_y^2 + u_z^2)$ .

Total enthalpy can be defined as

$$h_T = h + \frac{|\mathbf{u}|^2}{2}. \quad (2.4)$$

Since for an ideal gas total enthalpy can be related to total temperature as  $h_T = c_p T_T$ , it yields

$$T_T = T + \frac{|\mathbf{u}|^2}{2c_p}, \quad (2.5)$$

which can be expressed as a function of density, pressure and velocity as

$$T_T = c_p \frac{p}{\rho R} + \frac{1}{2} \frac{|\mathbf{u}|^2}{c_p}. \quad (2.6)$$

The increase of pressure  $p$  and total pressure  $p_T$  in the compressor, for an ideal (isentropic) gas can be related to the increase of temperature  $T$  and total temperature  $T_T$  by

$$\frac{p_2}{p_1} = \left( \frac{T_2}{T_1} \right)^{\frac{\gamma}{\gamma-1}}; \quad \frac{p_{T2}}{p_{T1}} = \left( \frac{T_{T2}}{T_{T1}} \right)^{\frac{\gamma}{\gamma-1}}. \quad (2.7a, 2.7b)$$

Similarly, for an ideal isentropic expansion in the turbine, we have

$$\frac{p_4}{p_3} = \left( \frac{T_4}{T_3} \right)^{\frac{\gamma}{\gamma-1}}; \quad \frac{p_{T4}}{p_{T3}} = \left( \frac{T_{T4}}{T_{T3}} \right)^{\frac{\gamma}{\gamma-1}}. \quad (2.8a, 2.8b)$$

In reality, the compression and expansion processes taking place in a real gas turbine deviate from the ideal cycle represented in figure 1.4. A turbomachine compressor or turbine is often characterized by its total pressure ratio and efficiency, indicating its deviation from the ideal cycle (dotted lines in figure 2.1). The total pressure ratio is the

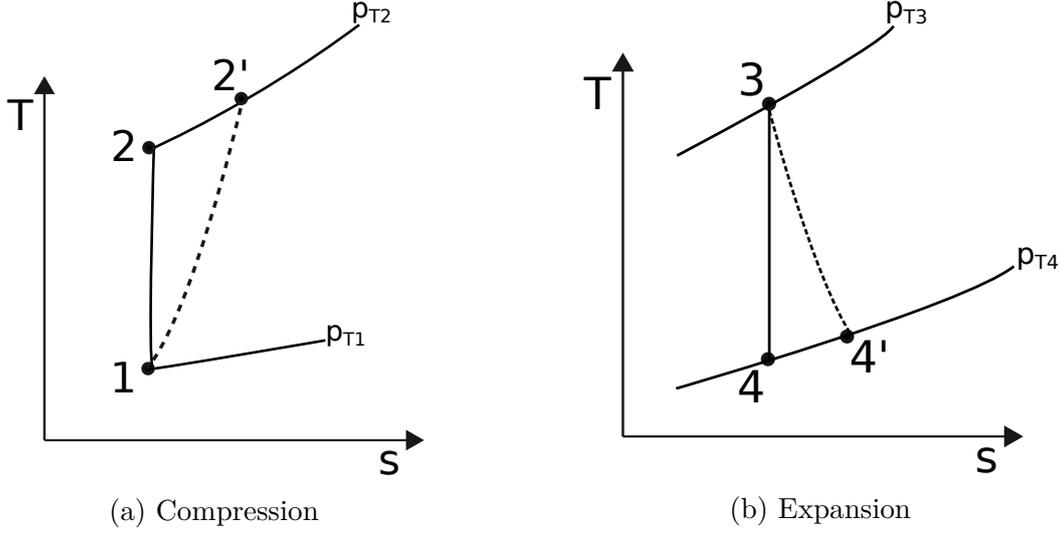


Figure 2.1: Non-isentropic compression and expansion processes.

ratio between the total pressure  $p_T$  before and after the compression/expansion, given by

$$\pi_c = \frac{p_{T2}}{p_{T1}} \quad \text{and} \quad \pi_t = \frac{p_{T4}}{p_{T3}}, \quad (2.9a, 2.9b)$$

where the subscripts  $c$  and  $t$  indicate the compressor and the turbine, respectively. The efficiency is the ratio of work for the ideal case versus the real process, which is given by

$$\eta_c = \frac{h_{T2s} - h_{T1}}{h_{T2} - h_{T1}} \quad \text{and} \quad \eta_t = \frac{h_{T3} - h_{T4}}{h_{T3} - h_{T4s}}, \quad (2.10a, 2.10b)$$

for the case of compressor and turbine, respectively. Inserting equations (2.7a, 2.7b) and (2.8a, 2.8b) into the previous expressions yields

$$\eta_c = \frac{\left(\frac{p_{T2}}{p_{T1}}\right)^{\frac{\gamma-1}{\gamma}} - 1}{\left(\frac{T_{T2}}{T_{T1}}\right) - 1} \quad \text{and} \quad \eta_t = \frac{\left(\frac{T_{T4}}{T_{T3}}\right) - 1}{\left(\frac{p_{T4}}{p_{T3}}\right)^{\frac{\gamma-1}{\gamma}} - 1}. \quad (2.11a, 2.11b)$$

## 2.2 Reynolds-Averaged Navier-Stokes Equations

The Navier-Stokes equations [68, 69], in conservative form, can be written as

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}_i}{\partial x_i} - \frac{\partial \mathbf{f}_{vi}}{\partial x_i} = \mathbf{Q}, \quad (2.12)$$

where  $\mathbf{q}$ ,  $\mathbf{f}$  and  $\mathbf{f}_v$  are the vectors of state variables, inviscid, and viscous fluxes, respectively, defined as

$$\mathbf{q} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{pmatrix}, \mathbf{f}_i = \begin{pmatrix} \rho u_i \\ \rho u_1 u_i + p \delta_{i1} \\ \rho u_2 u_i + p \delta_{i2} \\ \rho u_3 u_i + p \delta_{i3} \\ \rho E u_i + p u_i \end{pmatrix} \text{ and } \mathbf{f}_{v_i} = \begin{pmatrix} 0 \\ \tau_{ij} \delta_{j1} \\ \tau_{ij} \delta_{j2} \\ \tau_{ij} \delta_{j3} \\ u_j \tau_{ij} + q_i \end{pmatrix}, \quad (2.13)$$

and source term  $\mathbf{Q}$  represents all the potential body forces where  $\rho$  is the flow density,  $u_i$  is the mean velocity in direction  $i$ ,  $E$  is the total energy,  $\tau_{ij}$  are the viscous stresses and  $q_i$  is the heat flux and the source term  $\mathbf{Q}$  represents all potential body forces. If the viscous fluxes  $\mathbf{f}_v$  are dropped, we are left with the Euler equations.

Viscous stresses  $\tau_{ij}$  can be written, assuming a Newtonian fluid where shear stress varies linearly with strain rate, as

$$\tau_{ij} = \mu \left[ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] + \lambda \left[ \frac{\partial u_k}{\partial x_k} \right] \delta_{ij}, \quad (2.14)$$

where  $\mu$  is the dynamic viscosity coefficient and  $\lambda$  the bulk viscosity coefficient. Following Stokes' hypothesis, relating the two coefficients through  $\lambda = -2\mu/3$ , leads to

$$\tau_{ij} = \mu \left[ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right]. \quad (2.15)$$

The computation of high Reynolds viscous turbulent flows can be cumbersome, as a very fine resolution on the computational mesh is required to account for the turbulence effects. The RANS equations [70], as the name implies, are obtained by applying the so-called Reynolds averaging procedure to the Navier-Stokes equations. This averaging consists in decomposing the flow variables into a mean value and an instantaneous fluctuating part. This introduces an additional term to the equations, the so-called Reynolds-stress tensor, which has to be computed somehow to achieve closure of the equations. The Reynolds-stress tensor is typically approximated with models such as the one-equation Spallart-Almaras [71] or the two-equation models  $k$ - $\epsilon$ ,  $k$ - $\omega$  [72, 73] or SST [74]. These models are based on the Boussinesq approximation, which assumes that the turbulent

shear stress is linearly related to the mean rate of strain, meaning that the Reynolds-stress tensor can be calculated as the sum of a proportionality factor  $\mu_t$ , called the eddy viscosity, and the mean strain rate. Following this hypothesis, the previously introduced viscosity coefficient  $\mu$  is replaced by the sum of a laminar and a turbulent component, as

$$\mu = \mu_l + \mu_t. \quad (2.16)$$

The turbulence model used in the present work solves for the turbulence kinematic energy  $k$  and the specific rate of dissipation  $\omega$  using the  $k$ - $\omega$  model. This model introduces two new equations to the system:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho k u_j) = \tau_{ij} \frac{\partial u_i}{\partial x_j} - \beta_k \rho k \omega + \frac{\partial}{\partial x_j} \left[ (\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_j} \right] \quad (2.17)$$

and

$$\frac{\partial}{\partial t}(\rho \omega) + \frac{\partial}{\partial x_j}(\rho \omega u_j) = \alpha \frac{\omega}{k} \tau_{ij} \frac{\partial u_i}{\partial x_j} - \beta_\omega \rho \omega^2 + \frac{\partial}{\partial x_j} \left[ (\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_j} \right], \quad (2.18)$$

where  $\alpha$ ,  $\beta$ ,  $\sigma_k$  and  $\sigma_\omega$  are closure coefficients and the eddy viscosity  $\mu_t$  is obtained from

$$\mu_t = \frac{\rho k}{\omega}. \quad (2.19)$$

The reader is advised to check references [73, 75] for further details on this model.

## 2.2.1 Discretization of the RANS Equations

In their discretized form, the RANS equations can be written as

$$\frac{\partial \mathbf{q}}{\partial t} + \mathbf{R}_{ijk} = 0 \quad (2.20)$$

where the residual term  $\mathbf{R}$  encompasses all fluxes and source terms and the triad  $ijk$  represents the three directions of the computational mesh. A multi-block structured cell centered finite-volume scheme with second-order central differences is used to discretize the equations [76]. This means that the variables are assumed to be known at the center of each cell, and fluxes across the boundaries of each cell are evaluated as the average of the values in the cells on either side of the face. The second-order central-difference scheme

results in the following expression for both convective and viscous flux gradients,

$$\frac{\partial \mathbf{f}}{\partial x_1} = \frac{\mathbf{f}_{i+\frac{1}{2},j,k} - \mathbf{f}_{i-\frac{1}{2},j,k}}{\Delta x_1}, \quad (2.21)$$

where  $\mathbf{f}_{i\pm\frac{1}{2},j,k}$  represents the fluxes at the faces of the cell in direction 1 and  $\Delta x_1$  the distance between the central nodes at each face. The same occurs for the other directions. The fluxes across boundaries of each cell are evaluated as the average of the values in the cells on either side of the face. For face  $(i + \frac{1}{2}, j, k)$  this results in

$$\mathbf{f}_{i+\frac{1}{2},j,k} = \frac{1}{2} (\mathbf{f}_{i+1,j,k} + \mathbf{f}_{i,j,k}). \quad (2.22)$$

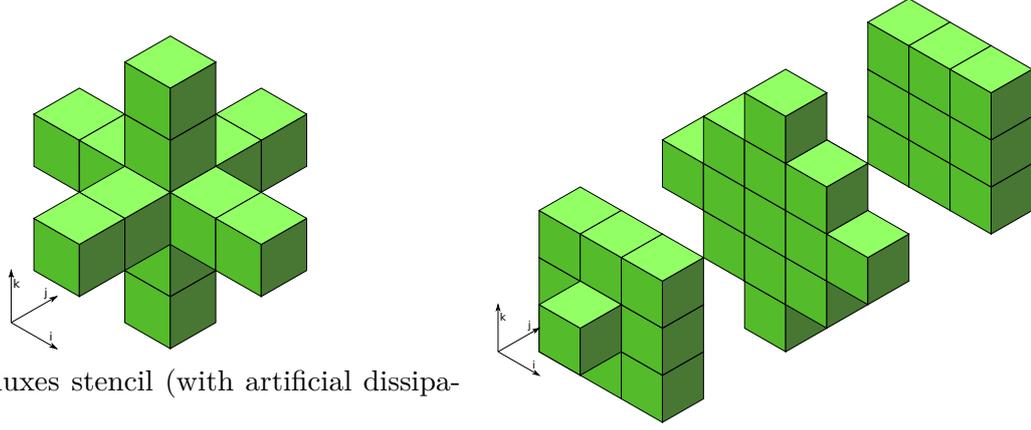
The fluxes across faces  $(i - \frac{1}{2}, j, k)$ ,  $(i, j \pm \frac{1}{2}, k)$  and  $(i, j, k \pm \frac{1}{2})$  are calculated accordingly. This discretization results in a three-point stencil for each flux gradient and a seven-point stencil for the entire convective flux contribution to the residual of the cell.

Artificial dissipation is used to suppress the tendency for odd and even point decoupling, and to prevent the appearance of wiggles in the regions containing severe pressure gradients in the neighborhood of shock waves or stagnation points [77]. It is introduced as a dissipative flux operator  $\mathbf{f}_d$ , which comes from a blend of second and fourth-order differences with coefficients which depend on the local pressure gradient. This is the so called Jameson-Schmidt-Turel (JST) scalar dissipation scheme. The stencil of this artificial dissipation scheme spans over two cells in each direction, therefore, requiring information from a total of thirteen cells (see figure 2.2a).

Viscous fluxes at the cell boundaries are obtained from a second-order discretization as well. This results in a stencil of 27 cells (see figure 2.2b). The discrete residual  $\mathbf{R}_{ijk}$  of a single cell can now be written as

$$\mathbf{R}(\mathbf{q})_{ijk} = \mathbf{h}_{i+\frac{1}{2},j,k} - \mathbf{h}_{i-\frac{1}{2},j,k} + \mathbf{h}_{i,j+\frac{1}{2},k} - \mathbf{h}_{i,j-\frac{1}{2},k} + \mathbf{h}_{i,j,k+\frac{1}{2}} - \mathbf{h}_{i,j,k-\frac{1}{2}}, \quad (2.23)$$

where  $\mathbf{h}$  is the sum of the various discretized fluxes (convective, viscous, and due to artificial



(a) Euler fluxes stencil (with artificial dissipation).

(b) RANS fluxes stencil.

Figure 2.2: Stencil of dependence of the residual of a single cell.

dissipation),  $h_{ijk} = f_{ijk} - f_{v_{ijk}} - f_{d_{ijk}}$ , yielding

$$\begin{aligned}
 \mathbf{R}(\mathbf{q})_{ijk} = & f_{i+\frac{1}{2},j,k} - f_{i-\frac{1}{2},j,k} + f_{i,j+\frac{1}{2},k} - f_{i,j-\frac{1}{2},k} + f_{i,j,k+\frac{1}{2}} - f_{i,j,k-\frac{1}{2}} \\
 & - f_{v_{i+\frac{1}{2},j,k}} + f_{v_{i-\frac{1}{2},j,k}} - f_{v_{i,j+\frac{1}{2},k}} + f_{v_{i,j-\frac{1}{2},k}} - f_{v_{i,j,k+\frac{1}{2}}} + f_{v_{i,j,k-\frac{1}{2}}} \\
 & - f_{d_{i+\frac{1}{2},j,k}} + f_{d_{i-\frac{1}{2},j,k}} - f_{d_{i,j+\frac{1}{2},k}} + f_{d_{i,j-\frac{1}{2},k}} - f_{d_{i,j,k+\frac{1}{2}}} + f_{d_{i,j,k-\frac{1}{2}}}.
 \end{aligned} \tag{2.24}$$

A multi-step Runge-Kutta explicit time marching scheme [77] with convergence acceleration via local time steps, residual averaging and V-/W-cycle multigrid [78, 79] is used to obtain the converged flow solution.

## 2.2.2 Physical Boundary Conditions

In a cell-centered scheme, the boundary conditions are applied by giving appropriate values to the "auxiliary" (or ghost) cells that surround the computational grid block (see figure 2.3).

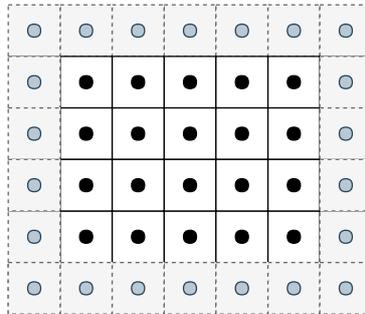


Figure 2.3: Sketch of cell-centered grid, with interior (full black) and auxiliary nodes (light blue).

Physical boundary conditions include the inlet, exit and wall boundary conditions, which come from the modeling of physical phenomenon, such as the non-slip condition of a wall, or the imposition of the flow that enters or exits the domain.

## Wall

In a solid wall boundary, the process of updating the auxiliary cell values to impose the zero flux condition and the no-slip condition (if required) can be described as follows.

The static pressure at the auxiliary cell is first computed from the pressure imposed at the wall surface as

$$(p_s)_{\text{aux}} = p_s^{\text{wall}} - (p_s)_{\text{int}} \quad (2.25)$$

The density at the auxiliary cell is then obtained from the wall temperature as

$$\rho_{\text{aux}} = \frac{(p_s)_{\text{aux}}}{T_{\text{aux}} R}, \quad (2.26)$$

where the auxiliary pressure is computed from the wall temperature  $T_{\text{wall}}$ , obtained from the wall integration or wall function solution for adiabatic or isothermal flows, and interior cell temperature as

$$T_{\text{aux}} = 2T_{\text{wall}} - T_{\text{int}}. \quad (2.27)$$

The internal temperature  $T_{\text{int}}$  is computed from the internal static pressure  $p_s$  and density  $\rho$  as

$$T_{\text{int}} = \frac{(p_s)_{\text{int}}}{\rho_{\text{int}} R}. \quad (2.28)$$

The momentum quantities at the auxiliary cell are obtained from extrapolation/reflection from the interior cell as

$$(\rho u_i)_{\text{aux}} = (\rho_{\text{int}} + \rho_{\text{aux}})(V_i^{\text{wall}}) - (\rho u_i)_{\text{int}}, \quad (2.29)$$

where  $V_i^{\text{wall}}$  is also obtained from either wall integration or wall function solutions. The momentum quantities are then corrected according to

$$(\rho u_i)_{\text{aux}} = (\rho u_i)_{\text{aux}} - \rho u_n (n_w)_i, \quad (2.30)$$

where  $\rho u_n$  is the momentum in the normal direction given by

$$\rho u_n = \sum_{i=1}^3 ((\rho u_i)_t + (\rho u_i)_{\text{int}}) n_{wi} - (\rho_{\text{aux}} + \rho_{\text{int}}) (u_n)_w, \quad (2.31)$$

with  $(u_n)_w$  being the whirl velocity contribution to the normal velocity at the wall. The energy equation quantities are computed as

$$(\rho E)_{\text{aux}} = \frac{1}{\gamma - 1} (p_s)_{\text{aux}} + \frac{1}{2} \frac{\sum_{i=1}^3 (\rho u_i)_{\text{aux}}^2}{\rho_{\text{aux}}}. \quad (2.32)$$

## Inlet

The vector of inlet boundary conditions  $\mathbf{U}^{\text{inlet}}$  is defined as

$$\mathbf{U}^{\text{inlet}} = \{p_T^{\text{inlet}}, h_T^{\text{inlet}}, V_t^{\text{inlet}}, C_r^{\text{inlet}}, C_z^{\text{inlet}}\} \quad (2.33)$$

where  $p_T$  is the total pressure,  $h_T$  is the total enthalpy,  $V_t$  is the tangential velocity,  $C_r$  and  $C_z$  are the direction cosines. Each of the five terms is of the size corresponding to the number of ghost (or auxiliary) cells that discretize the inlet and exit boundary faces. When turbulence models are in use, two extra quantities are introduced to the boundary conditions vector,  $k^{\text{inlet}}$  and  $\omega^{\text{inlet}}$ .

The inlet boundary condition assumed in the present work imposes the various quantities at the ghost cells, starting with total pressure, as

$$p_{\text{aux}} = \min(p_{\text{int}}, p_T^{\text{inlet}}). \quad (2.34)$$

The density imposed at the auxiliary cell is computed from total pressure  $p_T^{\text{inlet}}$  and total enthalpy  $h_T^{\text{inlet}}$  as

$$\rho_{\text{aux}} = \rho_T \left( \frac{p_{\text{aux}}}{p_T^{\text{inlet}}} \right)^{1/\gamma}; \quad \rho_T = \frac{\gamma}{\gamma - 1} \frac{p_T^{\text{inlet}}}{h_T^{\text{inlet}}}. \quad (2.35a, 2.35b)$$

The momentum equation variables are computed as

$$(\rho u_x)_{\text{aux}} = \rho_{\text{aux}} [V_r \cos(\theta) - V_t \sin(\theta)], \quad (2.36)$$

$$(\rho u_y)_{\text{aux}} = \rho_{\text{aux}} [V_r \sin(\theta) + V_t \cos(\theta)] \quad (2.37)$$

and

$$(\rho u_z)_{\text{aux}} = \rho_{\text{aux}} V_z, \quad (2.38)$$

where the velocities in radial, tangential and axial directions,  $V_r$ ,  $V_t$  and  $V_z$ , respectively, are computed as

$$V_r = V_{rz} C_r^{\text{inlet}}; \quad V_t = V_t^{\text{inlet}}; \quad V_z = V_{rz} C_z^{\text{inlet}}, \quad (2.39)$$

where

$$V_{rz} = \sqrt{V^2 - V_t^{\text{inlet}^2}}; \quad V^2 = 2(h_T^{\text{inlet}} - h^{\text{inlet}}); \quad h^{\text{inlet}} = \frac{\gamma}{\gamma - 1} \frac{p_{\text{aux}}}{\rho_{\text{aux}}}. \quad (2.40 \text{ a-c})$$

The conserved total energy at the auxiliary cell is computed as

$$(\rho E)_{\text{aux}} = \frac{1}{\gamma - 1} p_{\text{aux}} + \frac{1}{2} \rho_{\text{aux}} V^2, \quad (2.41)$$

and, lastly, the turbulence quantities are computed from the imposed b.c. values as

$$(\rho k)_{\text{aux}} = \rho_{\text{aux}} k^{\text{inlet}} \quad (\rho \omega)_{\text{aux}} = \rho_{\text{aux}} \omega^{\text{inlet}}. \quad (2.42)$$

## Exit

In turbomachinery subsonic flow, static pressure is usually prescribed at the outlet.

$$p_{\text{aux}} = \mathbf{U}^{\text{exit}} = p^{\text{exit}} \quad (2.43)$$

The density is obtained from the imposed pressure in the auxiliary cell as

$$\rho_{\text{aux}} = \frac{1}{\gamma - 1} \frac{p_{\text{aux}}}{e_{\text{aux}}}; \quad e_{\text{aux}} = E_{\text{aux}} - \frac{1}{2} V_{\text{aux}}^2; \quad E_{\text{aux}} = E_{\text{int}} = (\rho E)_{\text{int}} / \rho_{\text{int}}. \quad (2.44 \text{ a-c})$$

The momentum flow variables in the auxiliary cells are obtained from extrapolation of the states at the interior cells as

$$(\rho u_i)_{\text{aux}} = \rho_{\text{aux}} (u_i)_{\text{aux}}, \quad (2.45)$$

with

$$(u_i)_{\text{aux}} = (u_i)_{\text{int}} = (\rho u_i)_{\text{int}} / \rho_{\text{int}} . \quad (2.46)$$

The conserved energy variable is computed as

$$(\rho E)_{\text{aux}} = \rho_{\text{aux}} \left( e_{\text{aux}} + \frac{1}{2} V_{\text{aux}}^2 \right); \quad V_{\text{aux}}^2 = \sum_{i=1}^3 (u_i)_{\text{aux}}^2 , \quad (2.47\text{a}, 2.47\text{b})$$

and the turbulence quantities are computed as

$$(\rho k)_{\text{aux}} = \rho_{\text{aux}} (\rho k)_{\text{int}} / \rho_{\text{int}}; \quad (\rho \omega)_{\text{aux}} = \rho_{\text{aux}} (\rho \omega)_{\text{int}} / \rho_{\text{int}} . \quad (2.48\text{a}, 2.48\text{b})$$

### 2.2.3 Other Boundary Conditions

This subsection describes the non-physical boundary conditions that are relevant to this work. These boundary conditions come from the simplification and the numerical discretization of the problem.

#### Periodic

Periodic boundary conditions are used to reduce the computational domain and, thus, the solution cost, in problems with geometrical periodicity with the assumption of flow periodicity as well. In the case of an axial turbomachine, this domain can be a single blade row passage, making a pitchwise slice of the row.

Periodic boundary conditions are imposed by copying the interior cells at the periodic boundary into the auxiliary cells of the corresponding periodic boundary. For steady simulations there is no need for extra treatment of this boundary conditions besides the coordinate transformation. For unsteady simulations, the time the flow takes to travel through the computational domain must be taken into account by means of a phase-lag boundary condition.

#### Multi-block

Another type of non-physical boundary condition is the multi-block boundary condition. As the computational domain is subdivided in various blocks, which act as independent computational domains to a certain extent, there must exist an exchange of information between them, to correctly compute the flow solution across the various blocks.

Each block contains halo cells, that can be used to impose physical boundary conditions, if that is the case, to impose periodic boundary conditions, or the multi-block boundary conditions being described. These are imposed by exchanging information of the two interior cell layers close to the boundary of one block to the two auxiliary cell layers of an adjacent block, and vice versa.

For blocks with meshes consistent with each other, this is a straightforward process, as represented by figure 2.4. This type of connectivity between blocks is usually referred to as 1-2-1.

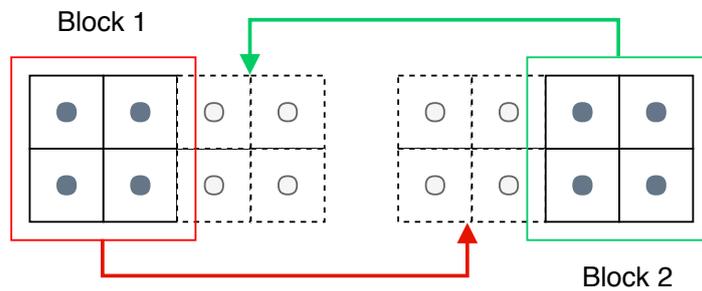


Figure 2.4: Schematic of multiblock 1-2-1 boundary condition update.

If the meshes of the blocks are not consistent, then, the so called generalized multi-block boundary condition is used, on which the auxiliary cells of one block are still updated from the interior cells of the adjacent blocks, but in this case, an interpolation must be performed, to accommodate for the relative difference between the location of the cells of each block.

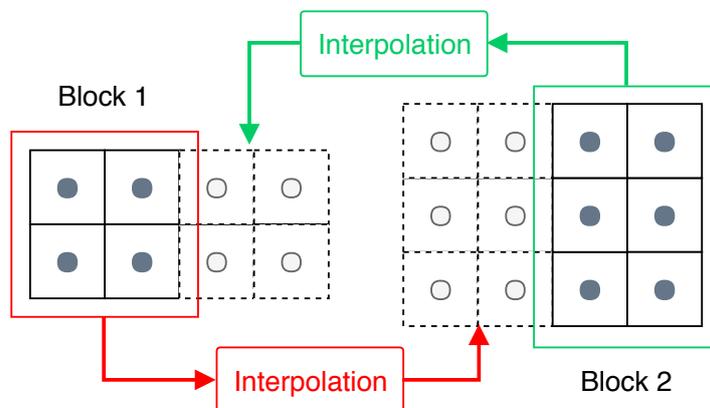


Figure 2.5: Schematic of generalized multi-block boundary condition update.

## 2.3 Multi-row Turbomachinery Analysis

Simulating multi-row turbomachinery components can be achieved in many ways of varying complexity and fidelity levels. Figure 2.6 schematically presents a series of approaches to the numerical integration of fluid zones with single and multiple rotating frames.

For single rotating frames, (SRF) the simulation is straightforward. An angular speed is given and introduced into the equations by means of a frame transformation. The simulation is limited to one rotating row and can only be performed in the absence of stators or volutes.

To simulate multiple rotating frames (or rotating and non-rotating) together, different approaches can be taken, of various complexities. For steady-state simulations they are the Frozen Rotor and Mixing-plane, while for unsteady simulations we have the sliding mesh. These three approaches are presented in the following subsections, with emphasis on the mixing-plane approach, to frame its use in the present work.

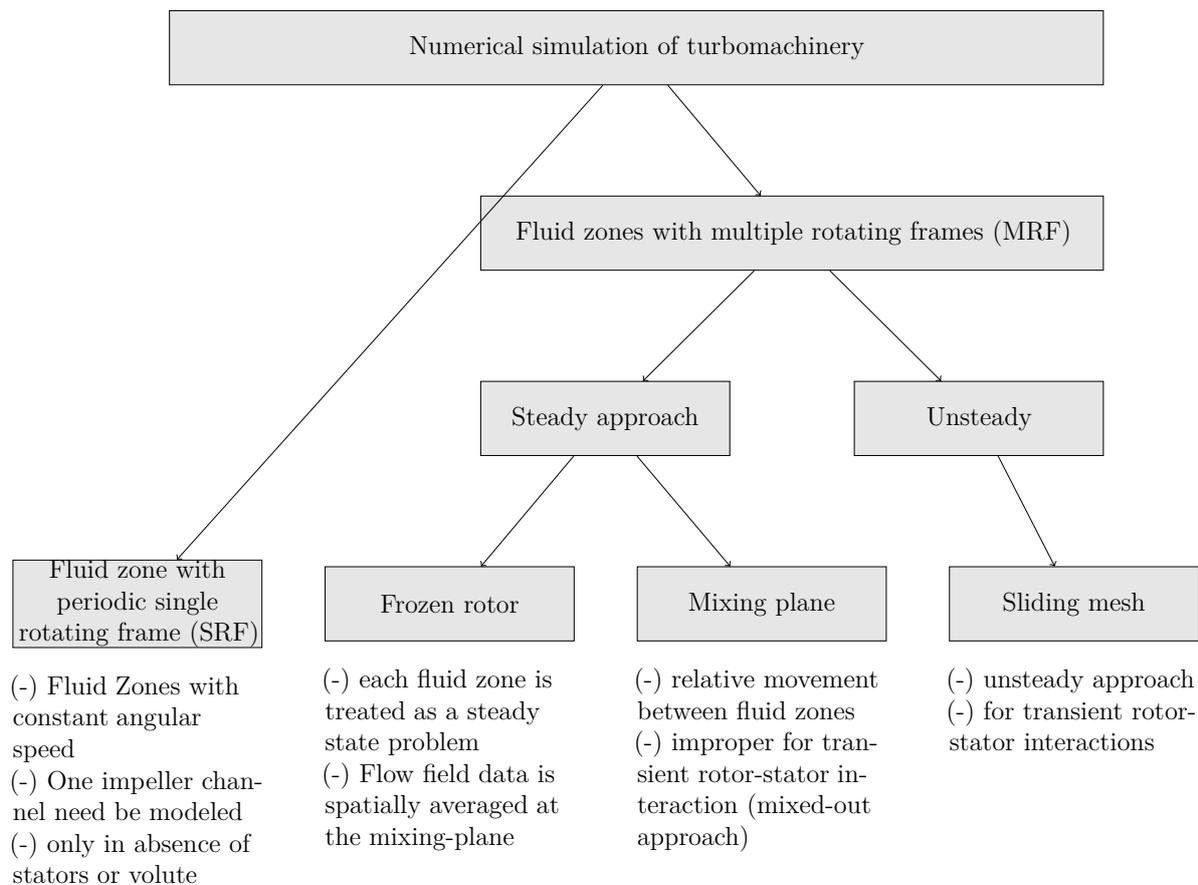


Figure 2.6: Various approaches to the numerical simulation of a turbomachinery (adapted from [80]).

### 2.3.1 Frozen Rotor

The simplest way to obtain a steady multi-row solution is the so called *frozen rotor* method. It is widely available and utilized [81–83] and works by fixing the relative position between the rotating and stationary parts. The rotating effects on the rotating sections are obtained with a frame transformation. It is directly applied to a multiple passage domain and it does not involve any truncation at the rotor-stator interface. As such, all spatial non-uniformities (wake and pressure waves) are able to pass through the interface accurately in a fully conservative manner. This steady state approach produces no transient effects so, rotating wakes, secondary flows, leading edge pressure increases, amongst others, will always stay in the same positions.

The frozen rotor approach is typically used to produce flow-fields to be used as starting point for unsteady sliding-mesh simulations [84].

### 2.3.2 Steady Mixing-plane

The mixing-plane method was first introduced by Denton [85] and has since become the industry standard for multi-row simulations. It is used with steady state simulations and it requires only a single blade per row, as illustrated in figure 2.7 for single stator/rotor stage. Between each blade passage, the flow properties are circumferentially averaged in the

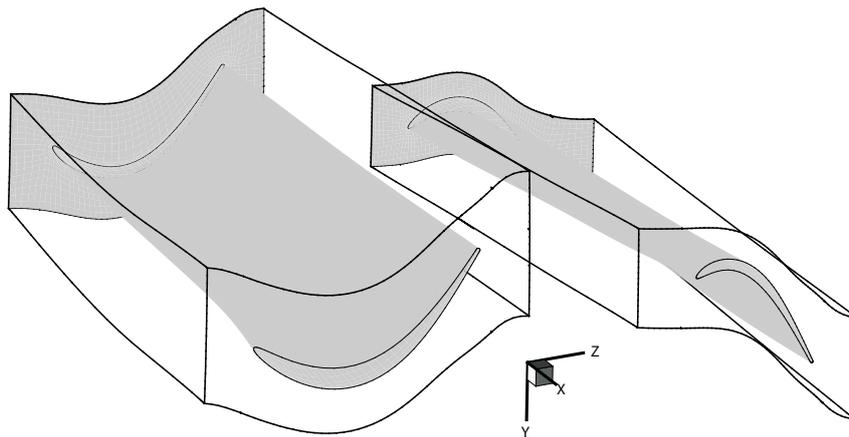


Figure 2.7: Two-row computational domain using a single blade per row.

so-called mixing-plane interface and exchanged between adjacent computational domains. The example is made for the case of an axial turbomachine but it can be easily extended to radial and mixed configurations.

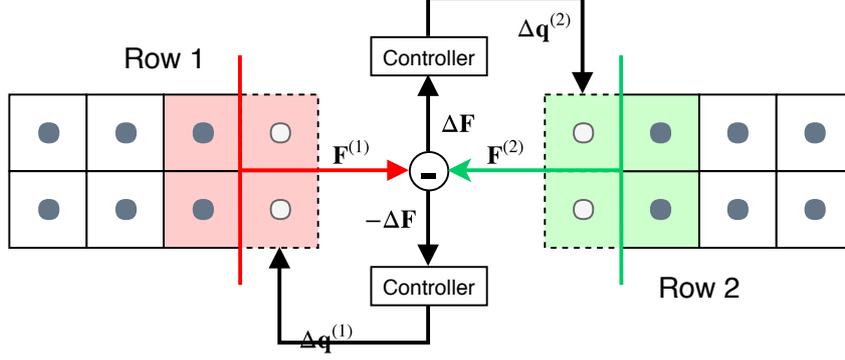


Figure 2.8: Schematic sketch of the flux balance algorithm [86].

Holmes [86] describes a mixing plane algorithm that achieves several key goals, including complete flux conservation at the interface, robustness, indifference to local flow direction and non-reflectivity. It consists in using a control-theory based flux balance algorithm to drive the differences between the fluxes in the two faces to zero, by updating the conserved variables in the ghost cells with a value based on the flux differences. To assure maximum non-reflectivity in the interface, the method uses the two dimensional approach of Giles [59]. The algorithm is schematically represented in figure 2.8. To obtain the quantity  $\Delta \mathbf{q}_{\text{local}}^{(i)}$  to be added to the auxiliary cell, and starting from the set of conserved variables, the fluxes across the interface are first defined as

$$\mathbf{F}_m = \{\rho u_n, \rho u_n u_s, \rho u_n u_\theta, \rho u_n^2 + p, \rho u_n h_T\}^T, \quad (2.49)$$

or

$$\mathbf{F}_u = \{\rho u_n, \rho u_n u_s, \rho u_n u_\theta, \rho u_n s, \rho u_n h_T\}^T, \quad (2.50)$$

where  $s$  is the entropy, the subscripts  $m$  and  $u$  indicate "mixed-out" or "unmixed" fluxes and  $u_s$ ,  $u_\theta$  and  $u_n$  are the velocity in spanwise, pitchwise and normal directions, respectively. As stated by Holmes [86], the first will have a higher average entropy than the average entropy of the flow before pitchwise averaging, while the latter will have the same entropy as the mass flux averaged entropy of the flow before pitchwise averaging. The difference  $\Delta \mathbf{F}$  is simply the difference across the interface,

$$\Delta \mathbf{F} = \mathbf{F}^{(1)} - \mathbf{F}^{(2)}. \quad (2.51)$$

This difference is treated as a small perturbation and is first translated into a perturbation

of the primitive variables,  $\Delta \mathbf{P}$  from the multiplication of the fluxes difference with  $\mathbf{A}^{-1}$ , as

$$\Delta \mathbf{P} = \mathbf{A}^{-1} \cdot \Delta \mathbf{F}, \quad (2.52)$$

with  $\mathbf{A}$  being given by

$$A_m = \begin{bmatrix} u_n & 0 & 0 & \rho & 0 \\ u_n u_s & \rho u_n & 0 & \rho u_s & 0 \\ u_n u_\theta & 0 & \rho u_n & \rho u_\theta & 0 \\ v_n^2 & 0 & 0 & 2\rho u_n & 1 \\ u_n h_T + \rho u_n \left. \frac{\partial h}{\partial \rho} \right|_p & \rho u_n u_s & \rho u_n u_\theta & \rho h_T + \rho u_n^2 & \rho u_n \left. \frac{\partial h}{\partial p} \right|_\rho \end{bmatrix}, \quad (2.53)$$

or

$$A_u = \begin{bmatrix} u_n & 0 & 0 & \rho & 0 \\ u_n u_s & \rho u_n & 0 & \rho u_s & 0 \\ u_n u_\theta & 0 & \rho u_n & \rho u_\theta & 0 \\ u_n s + \rho u_n \left. \frac{\partial s}{\partial \rho} \right|_p & 0 & 0 & \rho s & \rho u_n \left. \frac{\partial s}{\partial p} \right|_\rho \\ u_n h_T + \rho u_n \left. \frac{\partial h}{\partial \rho} \right|_p & \rho u_n u_s & \rho u_n u_\theta & \rho h_T + \rho u_n^2 & \rho u_n \left. \frac{\partial h}{\partial p} \right|_\rho \end{bmatrix}, \quad (2.54)$$

depending on either mixed-out or unmixed fluxes are being used, respectively. The transformation of primitive perturbations into one-dimensional characteristics,

$$\Delta \boldsymbol{\lambda} = \{\Delta p - c^2 \Delta \rho, \rho c \Delta u_s, \rho c \Delta u_\theta, \Delta p + \rho c \Delta u_n, \Delta p - \rho c \Delta u_n\}^T \quad (2.55)$$

is given by

$$\Delta \boldsymbol{\lambda} = \mathbf{B} \cdot \Delta \mathbf{P}, \quad (2.56)$$

with  $\mathbf{B}$  being defined as

$$\mathbf{B} = \begin{bmatrix} -c^2 & 0 & 0 & 0 & 1 \\ 0 & \rho c & 0 & 0 & 0 \\ 0 & 0 & \rho c & 0 & 0 \\ 0 & \rho c & 0 & \rho c & 1 \\ 0 & \rho c & 0 & -\rho c & -1 \end{bmatrix}, \quad (2.57)$$

where  $c$  is the local speed of sound. The characteristic perturbations are then segregated into two parts,  $\Delta\boldsymbol{\lambda}^{(1)}$  and  $\Delta\boldsymbol{\lambda}^{(2)}$ , depending on the sign of the normal velocity and on the Mach number normal to the interface. If the local direction of the flow is in the downstream direction (1 to 2),  $\Delta\boldsymbol{\lambda}^{(1)}$  will consist only of the forth element of  $\Delta\boldsymbol{\lambda}$  (the upstream acoustic characteristics). The part  $\Delta\boldsymbol{\lambda}^{(2)}$  will have the remaining characteristics, with the forth being zero. The selection of the characteristics is given by

$$\Delta\boldsymbol{\lambda}^{(i)} = \mathbf{D}^{(i)} \Delta \cdot \boldsymbol{\lambda}, \quad (2.58)$$

with  $\mathbf{D}$  being a diagonal matrix with diagonal entries of  $\{0, 0, 0, 1, 0\}$  or  $\{1, 1, 1, 0, 1\}$ . The transformation of the characteristic perturbations on each side of the interface into conserved quantities is performed by first converting them into primitive perturbations, as

$$\Delta\mathbf{P}^{(i)} = \mathbf{B}^{-1} \cdot \Delta\boldsymbol{\lambda}^{(i)}, \quad (2.59)$$

and finally by converting the primitive perturbations to perturbations in the conserved variables, as

$$\Delta\mathbf{q}^{(i)} = \mathbf{C} \cdot \Delta\mathbf{P}^{(i)}, \quad (2.60)$$

with  $\mathbf{C}$  being defined as

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u_s & \rho & 0 & 0 & 0 \\ u_\theta & 0 & \rho & 0 & 0 \\ u_n & 0 & 0 & \rho & 0 \\ E + \rho \left. \frac{\partial e}{\partial \rho} \right|_p & \rho u_s & \rho u_\theta & \rho u_n & \rho \left. \frac{\partial e}{\partial p} \right|_\rho \end{bmatrix}. \quad (2.61)$$

The whole process can then be written as

$$\Delta \mathbf{q}^{(i)} = \mathbf{C} \cdot \mathbf{B}^{-1} \cdot \mathbf{D}^{(i)} \cdot \mathbf{B} \cdot \mathbf{A}^{-1} \cdot \Delta \mathbf{F}. \quad (2.62)$$

The overall procedure (schematically represented in figure 2.9) can be condensed in the following five steps:

1. Compute the fluxes profiles  $\mathbf{p}$  from conserved quantities at the mixing-plane face and create a local profile by averaging them at each spanwise position,

$$\mathbf{p}_{\text{local},j} = f(\tilde{\mathbf{q}}_j); \quad (2.63)$$

2. Communicate the local radial profiles  $\mathbf{p}_{\text{local}}$  of averaged quantities as donor profiles  $\mathbf{p}_{\text{don}}$  between blade rows,

$$\mathbf{p}_{\text{local}} \rightarrow \mathbf{p}_{\text{don}}; \quad (2.64)$$

3. Interpolate the received  $\mathbf{p}_{\text{rec}}$  profiles to match local cell distribution,

$$\mathbf{p}_{\text{rec}} = f(\mathbf{p}_{\text{don}}, \mathbf{p}_{\text{local}}); \quad (2.65)$$

4. Compute differences in fluxes and state variables  $\mathbf{p}_{\text{rec}}^*$  between the interpolated and local profiles,

$$\mathbf{p}_{\text{rec}}^* = f(\mathbf{p}_{\text{rec}}, \mathbf{p}_{\text{local}}); \quad (2.66)$$

5. Compute the variation in the conserved variables  $\mathbf{q}^*$  to be applied to the auxiliary

cells, from the flux differences (as given by equation (2.62)) and update them,

$$\mathbf{q}_{\text{local}}^* = f(\mathbf{p}_{\text{rec}}^*, \mathbf{q}_{\text{local}}). \quad (2.67)$$

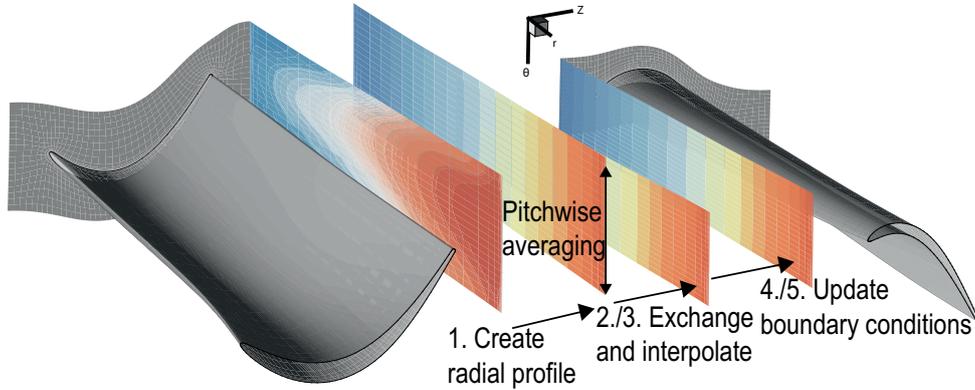


Figure 2.9: Schematic representation of the mixing-plane interface steps.

### 2.3.3 Unsteady Sliding-mesh

With increased complexity compared to the steady mixing-plane, the sliding-mesh method is able to capture the unsteady interaction between consecutive rows of a turbomachine [87–91].

The first description of sliding meshes was done by Rai [92, 93] and, much like the previously described mixing-plane, the flow fields of multiple blade rows are computed simultaneously. With the sliding-mesh approach, the simulations are run in fully unsteady mode and rotating effects are included into the simulation by moving the meshes of adjacent rows relative to each other at every time step. This leads to non-matching cells at the interface between the two meshes (moving and non-moving), which introduces a problem of interpolation. At the interface between two blade rows in relative motion, the relative grid must lie on a common surface of revolution. In cell-centered discretization, the interpolation transfers the information from the interior cells on one side of the interface to the auxiliary cells on the other. The interpolation scheme should guarantee a complete balance of fluxes across the interface, which can be assured by interpolating not only flow variables across the interface but also fluxes [94].

Typically, to avoid instabilities due to resonance between different rings, the number of stator vanes and rotor blades in a turbomachine does not have a common denominator.

This is known as *detuning* [95]. A full unsteady sliding-mesh computation would therefore need the simulation of the complete wheel, including all stator vanes and rotor blades. This is often not possible due to increased computational requirements, so the number of vanes and rotor blades is reduced by finding a denominator that is almost common, and scaling the geometry circumferentially. This might introduce some cases that require special treatment during the iterative process, due to the meshes on either side of the interface not being aligned at a given time step. For the case of equal angular width meshes, this can be done by simply replicating the mesh, since the problem is periodic. When non-equal angular width meshes are used, a phase-lag storage scheme must be used [96].

# Chapter 3

## Techniques of Sensitivity Analysis

As previously mentioned in the first chapter, a numerical optimization problem, when solved using GB algorithms will require the gradients of the objective functions and constraints relative to the design variables, in order to select the direction on which it will advance to reach the (local) optimal solution. Given a set of functions (or vector-valued function)  $\mathbf{f}$  which depend on a set of independent variables  $\boldsymbol{\alpha}$ , we are then interested in calculating the Jacobian  $d\mathbf{f}/d\boldsymbol{\alpha}$ , which can be expanded as

$$\frac{d\mathbf{f}}{d\boldsymbol{\alpha}} = \begin{bmatrix} \frac{df_1}{d\alpha_1} & \dots & \frac{df_1}{d\alpha_{N_\alpha}} \\ \vdots & \ddots & \vdots \\ \frac{df_{N_f}}{d\alpha_1} & \dots & \frac{df_{N_f}}{d\alpha_{N_\alpha}} \end{bmatrix}, \quad (3.1)$$

with size  $N_f \times N_\alpha$ . This chapter presents various methods of computing the entries of this matrix and discusses their advantages and disadvantages. At the end of the chapter, the method selected to continue with the proposed work is presented and its choice supported.

### 3.1 Finite Differences

The finite differences method is the most widely used for calculating derivatives. Due to its simplicity, it can be easily implemented, even when using black-box computational models. The finite differences formulas are obtained from the combination of Taylor series expansions [97, 98] and the resulting expressions can be of arbitrary order of accuracy.

The second-order central difference of the first derivative of  $f$  is given as

$$\frac{df}{d\alpha_j} = \frac{\mathbf{f}(\boldsymbol{\alpha} + h\mathbf{e}_j) - \mathbf{f}(\boldsymbol{\alpha} - h\mathbf{e}_j)}{2h} + \mathcal{O}(h^2), \quad (3.2)$$

where  $h$  is the step size and  $\mathcal{O}(h^2)$  represents the truncation error, which is proportional to the largest term of the remainder. For this case it reduces with the square of  $h$ .

Each evaluation of the derivative results in a column of the Jacobian in equation (3.1), thus, the cost of computing the full Jacobian with this method is proportional to the number of design variables  $N_\alpha$ .

With finite differences, reducing  $h$  only reduces the truncation error down to a certain value, from which the error does in fact increase, due to subtractive cancellation error [99]. This error appears for small values of the perturbation step, which lead to both perturb terms of equation (3.2) being nearly identical ( $\mathbf{f}(\boldsymbol{\alpha} + h\mathbf{e}_j) \simeq \mathbf{f}(\boldsymbol{\alpha} - h\mathbf{e}_j)$ ), yielding a zero valued derivative. Such behavior can be observed in figure 3.1, where the relative error of two FD approximations of different order of the first derivative of a selected function is presented. The different slopes of the two lines reveal the order of the two approximations, with the red line (second-order) having a slope of -2 and the blue line (fourth-order) having a slope of -4. Starting at  $h = 10^{-5}$ , the error of the second order approximation starts to increase. The fourth-order derivative reaches lower errors for larger  $h$  but the error starts to rise sooner. Due to this increase in the dominance of the subtractive cancellation error for small values of  $h$ , a thorough analysis of the optimal perturbation step must be performed to remain out of that problematic region.

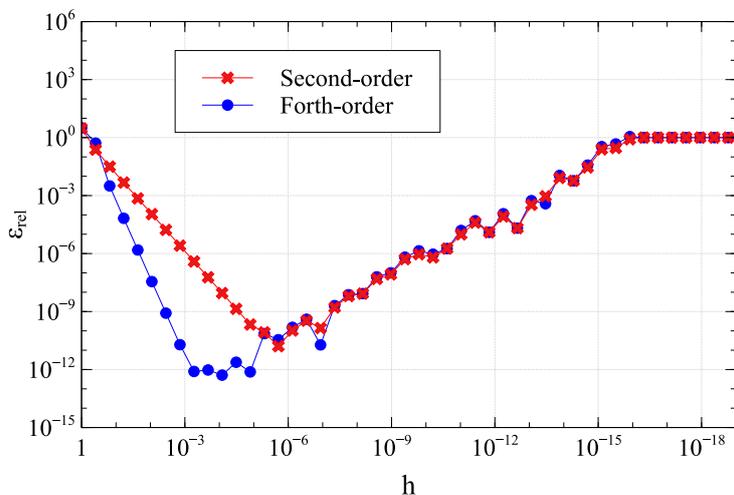


Figure 3.1: Relative error of finite-difference approximations of the first derivative of a function with respect to the perturbation step.

## 3.2 Complex Step

The complex step approximation uses complex variables to compute the derivatives of real functions [28]. Like FD, the complex-step derivative approximation can also be derived using a Taylor series expansion, using an imaginary step  $ih$  instead of the real step  $h$ . Expanding  $\mathbf{f}$  in a Taylor series around a real point  $\alpha$  leads to

$$\mathbf{f}(\alpha + ih\mathbf{e}_j) = \mathbf{f}(\alpha) + ih \frac{\partial \mathbf{f}}{\partial \alpha_j} - \frac{h^2}{2} \frac{\partial^2 \mathbf{f}}{\partial \alpha_j^2} - \frac{ih^3}{6} \frac{\partial^3 \mathbf{f}}{\partial \alpha_j^3} + \dots \quad (3.3)$$

The first derivative of  $\mathbf{f}$  is then obtained by taking the imaginary part of both sides of the previous equation and dividing it by  $h$ , yielding

$$\frac{d\mathbf{f}}{d\alpha_j} = \frac{\text{Im}[\mathbf{f}(\alpha + ih\mathbf{e}_j)]}{h} + \mathcal{O}(h^2), \quad (3.4)$$

which approximates the derivative with order  $\mathcal{O}(h^2)$ .

Unlike the finite-difference formula, there is no subtraction in the complex-step derivative approximation and the only source of numerical error is the truncation error. One can then reduce  $h$  until the truncation error is of the same order as the numerical precision of the evaluation of  $\mathbf{f}$ . Each evaluation of the derivative results in a column of the Jacobian in equation (3.1), thus, the cost of computing the full Jacobian with this method is proportional to the number of design variables  $N_\alpha$ . This method also requires access to the source code of the computational model to modify all real variables and computations with complex ones and therefore cannot be applied to black-box models.

## 3.3 Automatic Differentiation

Automatic differentiation – also known as *algorithmic* or *computational differentiation* – applies the chain rule to computer programs to obtain derivatives of their outputs based on their inputs [100].

Any computational algorithm consists in a sequence of operations that can be expressed in the form

$$t_i = f_i(t_1, t_2, \dots, t_{i-1}), \quad i = n + 1, n + 2, \dots, m, \quad (3.5)$$

where each function  $f_i$  can be either a unary or binary operation,  $t_1, t_2, \dots, t_n$  are the

independent variables and  $t_{n+1}, t_{n+2}, \dots, t_m$  are the dependent variables. By applying the chain rule, the derivative of  $t_i$  with respect to  $t_j$  is given by

$$\frac{\partial t_i}{\partial t_j} = \sum_{k=1}^{i-1} \frac{\partial f_i}{\partial t_k} \frac{\partial t_k}{\partial t_j} \quad j = 1, 2, \dots, n. \quad (3.6)$$

### 3.3.1 Forward and Reverse Modes

Automatic differentiation can operate in two different modes – *forward* (*tangent* or *direct*) and *reverse* (*backwards* or *adjoint*). The forward mode propagates the required sensitivity data along with the solution, as it is computed. The reverse mode requires the function to be computed first-hand, with the intermediate values stored. These intermediate values are then used by the reversed code to compute the sensitivities.

If we look at equation (3.6), the forward mode works by selecting one index  $j$ , keeping it fixed and then working our way forward in the index  $i$  until we arrive to the desired derivative. A function differentiated using the direct mode produces, with each run, a column of the Jacobian matrix.

The reverse mode, on the other hand, works by fixing  $i$  and advancing backward in the index  $j$  until reaching the independent variable. A function differentiated using the reverse mode produces, with each run, a whole row of the Jacobian matrix.

Lets consider the subroutine

```
subroutine compute_bc( $\downarrow$  $\mathbf{q}$ ,  $\downarrow$  $\mathbf{p}$ ,  $\mathbf{q}^*$ ),
 $\downarrow$ 
```

written with Fortan syntax, representing the mixing-plane algorithm step that updates the boundary conditions. The subroutine would take as inputs the state solution array to be updated  $\mathbf{q}$ , the profile with the information from the adjacent row  $\mathbf{p}$  and would output the updated state solution  $\mathbf{q}^*$ . The subroutine obtained from AD using the forward mode would appear as

```
subroutine compute_bc_d( $\downarrow$  $\mathbf{q}$ ,  $\downarrow$  $\mathbf{q}_d$ ,  $\downarrow$  $\mathbf{p}$ ,  $\downarrow$  $\mathbf{p}_d$ ,  $\mathbf{q}^*$ ,  $\mathbf{q}_d^*$ ).
 $\downarrow$   $\downarrow$ 
```

This new subroutine would now take as inputs, besides the original  $\mathbf{q}$  and  $\mathbf{p}$ , the so called seeds  $\mathbf{p}_d$  and  $\mathbf{q}_d$ . The derivative of the full array  $\mathbf{q}^*$  with respect to a combination of

selected inputs is given by this subroutine as

$$\mathbf{q}_d^* = \frac{\partial \mathbf{q}^*}{\partial \mathbf{q}} \mathbf{q}_d + \frac{\partial \mathbf{q}^*}{\partial \mathbf{p}} \mathbf{p}_d. \quad (3.7)$$

Differentiating the original subroutine using the reverse mode would produce

```
subroutine compute_bc_b( $\mathbf{q}$ ,  $\mathbf{q}_b$ ,  $\mathbf{p}$ ,  $\mathbf{p}_b$ ,  $\mathbf{q}^*$ ,  $\mathbf{q}_b^*$ ),
```

where  $\mathbf{q}_b$  and  $\mathbf{p}_b$  are the output derivatives and  $\mathbf{q}_b^*$  is the seed. The output derivatives are computed in the differentiated subroutine as

$$\mathbf{q}_b = \left( \frac{\partial \mathbf{q}^*}{\partial \mathbf{q}} \right)^T \mathbf{q}_b^*, \quad \mathbf{p}_b = \left( \frac{\partial \mathbf{q}^*}{\partial \mathbf{p}} \right)^T \mathbf{q}_b^*. \quad (3.7 \text{ a,b})$$

### 3.3.2 Source Transformation and Operator Overloading

Automatic differentiation can be implemented by two methods: *source code transformation* and *operator overloading*.

The first method involves the original code being processed by a parser which introduces additional lines corresponding to all the derivative calculations while generating the differentiated version of the source code. This method produces code that is practically unreadable, which might constitute an implementation disadvantage, as it becomes impractical to debug the new version of the code. Despite that and being necessary to rerun the parser every time the original code is changed, this method has the advantage of yielding considerably faster code. The use of automated scripts can also help preparing the code for being parsed, making the whole differentiation process much more streamlined. This approach is sometimes an imposition of the language in which the original code is developed. Some examples of AD tools that use the source transformation method to differentiate Fortran code are ADIFOR [101], TAF [102], OpenAD/F [103], and Tapenade [104].

The second method defines a new user-defined type, which is used instead of the real type, including both value of the original variable and the derivative. This requires the source code to be written in a language that supports derived data types, such as Fortran 90 or C++. This method also requires that all the intrinsic operations and functions are

redefined for the new data type, so that these functions also compute the derivative as well as the real value. Albeit usually resulting in a very elegant implementation, with little changes to the original code, the differentiated version of the code generated using this method is usually less efficient than that obtained with source code transformation. Tools such as ADOL-C [105], Adept [106], CodiPack [107] use the operator overloading method to differentiate code.

### 3.3.3 Example using *Tapenade*

An example of how Tapenade can be used to differentiate a Fortran routine is presented below. Figure 3.2 presents the routine `ubdate_bc` that is to be differentiated. This routine was written as an overly simplified version of how the boundary conditions are updated by the mixing-plane algorithm. It can be seen from the source code that the subroutine receives an averaged profile and from its values it updates the boundary conditions. Figures 3.3 and 3.4 present the source code of the routines differentiated in the forward and reverse modes, respectively.

```

1      subroutine update_bc(n1, n2, bc_old, profile, bc_new)
2          implicit none
3
4          integer, intent(in)  :: n1, n2
5          real(r8), intent(in) :: bc_old(n1,n2)
6          real(r8), intent(in) :: profile(n2)
7          real(r8), intent(out) :: bc_new(n1,n2)
8
9          integer :: i1, i2
10
11         do i1 = 1, n1
12             do i2 = 1, n2
13                 bc_new(i1,i2) = bc_old(i1,i2)**2
14             &                + 0.75*profile(i2)
15             end do
16         end do
17
18     end subroutine update_bc

```

Figure 3.2: Original Fortran subroutine to be differentiated.

Looking at the routine differentiated in forward mode, `update_bc_d`, we can see that it takes two new data variables as inputs and a new variable as output, as described in subsection 3.3.1. The two new input variables, `bc_oldd` and `profiled`, are used as seeds

to compute the derivative of `bc_old` with respect to `bc_old` and `profile`, respectively, as stated by equation (3.7).

```

1      SUBROUTINE UPDATE_BC_D(n1,      n2,
2      &                      bc_old,  bc_oldd,
3      &                      profile, profiled,
4      &                      bc_new,  bc_newd )
5      IMPLICIT NONE
6  C
7      INTEGER, INTENT(IN) :: n1, n2
8      REAL*(r8), INTENT(IN) :: bc_old(n1, n2)
9      REAL*(r8), INTENT(IN) :: bc_oldd(n1, n2)
10     REAL*(r8), INTENT(IN) :: profile(n2)
11     REAL*(r8), INTENT(IN) :: profiled(n2)
12     REAL*(r8), INTENT(OUT) :: bc_new(n1, n2)
13     REAL*(r8), INTENT(OUT) :: bc_newd(n1, n2)
14  C
15     INTEGER i1, i2
16     INTEGER ii2
17     INTEGER ii1
18     DO ii1=1,n2
19         DO ii2=1,n1
20             bc_newd(ii2, ii1) = 0.0
21         ENDDO
22     ENDDO
23     DO i1=1,n1
24         DO i2=1,n2
25             bc_newd(i1, i2) = bc_oldd(i1, i2) +
26 &                          0.75*2*profile(i2)*profiled(i2)
27             bc_new(i1, i2) = bc_old(i1, i2) + 0.75*profile(i2)**2
28         ENDDO
29     ENDDO
30     END

```

Figure 3.3: Fortran subroutine automatically differentiated using the forward mode.

The routine differentiated in backward mode, `update_bc_b`, also has three new variables in its arguments, one input `bc_newb` and two outputs `bc_olddb` and `profileb`. The new input variable is the seed in this case, whose values, as seen from figure 3.4, are used as weights for the output derivatives. This is extremely useful if one wants to create a chain of various differentiated routines, which is then made by using the resulting derivative of a previous routine as the seed of the next differentiated routine.

```

1      SUBROUTINE UPDATE_BC_B(n1, n2, bc_old, bc_oldb, profile,
2      &                        profileb, bc_new, bc_newb)
3      IMPLICIT NONE
4  C
5      INTEGER, INTENT(IN) :: n1, n2
6      REAL*(r8), INTENT(IN) :: bc_old(n1, n2)
7      REAL*(r8) bc_oldb(n1, n2)
8      REAL*(r8), INTENT(IN) :: profile(n2)
9      REAL*(r8) profileb(n2)
10     REAL*(r8) bc_new(n1, n2)
11     REAL*(r8) bc_newb(n1, n2)
12  C
13     INTEGER i1, i2
14     INTEGER ii1
15     INTEGER ii2
16     DO ii1=1,n2
17         DO ii2=1,n1
18             bc_oldb(ii2, ii1) = 0.0
19         ENDDO
20     ENDDO
21     DO ii1=1,n2
22         profileb(ii1) = 0.0
23     ENDDO
24     DO i1=n1,1,-1
25         DO i2=n2,1,-1
26             bc_oldb(i1, i2) = bc_oldb(i1, i2) + bc_newb(i1, i2)
27             profileb(i2) = profileb(i2) +
28 &                        0.75*2*profile(i2)*bc_newb(i1, i2)
29             bc_newb(i1, i2) = 0.0
30         ENDDO
31     ENDDO
32     END

```

Figure 3.4: Fortran subroutine automatically differentiated using the reverse mode.

## 3.4 Semi-analytic Methods

Typically, the objective functions introduced in the beginning of this chapter do not depend uniquely on a set of design variables, but also on the physical state of the system  $\mathbf{q}$ , which usually also depends on those design variables,

$$\mathbf{f} = \mathbf{f}(\boldsymbol{\alpha}, \mathbf{q}(\boldsymbol{\alpha})). \quad (3.8)$$

Applying the chain rule of differentiation, the total sensitivity of  $\mathbf{f}$  with respect to the design variables is given as

$$\frac{d\mathbf{f}}{d\boldsymbol{\alpha}} = \frac{\partial\mathbf{f}}{\partial\boldsymbol{\alpha}} + \frac{\partial\mathbf{f}}{\partial\mathbf{q}} \frac{d\mathbf{q}}{d\boldsymbol{\alpha}}, \quad (3.9)$$

with the size of the sensitivity matrices being

$$\frac{\partial\mathbf{f}}{\partial\boldsymbol{\alpha}} \quad (N_f \times N_\alpha), \quad \frac{\partial\mathbf{f}}{\partial\mathbf{q}} \quad (N_f \times N_q), \quad \frac{d\mathbf{q}}{d\boldsymbol{\alpha}} \quad (N_q \times N_\alpha), \quad (3.10)$$

where  $N_f$  is the number of functions of interest,  $N_\alpha$  the number of design variables and  $N_q$  is the size of the state vector, which, for the solution of a large, three-dimensional problem involving a system of conservation laws, can be very large. The size of the state vector will depend on both the number of computational cells,  $N_c$ , and governing equations,  $N_v$ , as  $N_q = N_v \times N_c$ .

Let us also consider the system of governing equations represented by the residual

$$\mathcal{R}(\boldsymbol{\alpha}, \mathbf{q}(\boldsymbol{\alpha})) = 0, \quad (3.11)$$

where the first instance of  $\boldsymbol{\alpha}$  indicates that the residual may depend explicitly on the design variables. In the same way as for the function of interest  $\mathbf{f}$ , the total derivative of  $\mathcal{R}$  is given by

$$\frac{d\mathcal{R}}{d\boldsymbol{\alpha}} = \frac{\partial\mathcal{R}}{\partial\boldsymbol{\alpha}} + \frac{\partial\mathcal{R}}{\partial\mathbf{q}} \frac{d\mathbf{q}}{d\boldsymbol{\alpha}} = 0, \quad (3.12)$$

which, after rearranging, provides an expression for computing the total sensitivity of the state variables with respect of the design variables  $d\mathbf{q}/d\boldsymbol{\alpha}$ , given by

$$\frac{\partial\mathcal{R}}{\partial\mathbf{q}} \frac{d\mathbf{q}}{d\boldsymbol{\alpha}} = -\frac{\partial\mathcal{R}}{\partial\boldsymbol{\alpha}}, \quad (3.13)$$

with the size of the sensitivity matrices being

$$\frac{\partial\mathcal{R}}{\partial\mathbf{q}} \quad (N_q \times N_q), \quad \frac{\partial\mathcal{R}}{\partial\boldsymbol{\alpha}} \quad (N_q \times N_\alpha). \quad (3.14)$$

Solving equation (3.13) for  $d\mathbf{q}/d\boldsymbol{\alpha}$  and substituting the result into equation (3.9) yields

$$\frac{d\mathbf{f}}{d\boldsymbol{\alpha}} = \frac{\partial\mathbf{f}}{\partial\boldsymbol{\alpha}} - \frac{\partial\mathbf{f}}{\partial\mathbf{q}} \left[ \frac{\partial\mathcal{R}}{\partial\mathbf{q}} \right]^{-1} \frac{\partial\mathcal{R}}{\partial\boldsymbol{\alpha}}. \quad (3.15)$$

From this expression, there are two methods to evaluate the total derivative  $d\mathbf{f}/d\boldsymbol{\alpha}$ , the direct and the adjoint. Each one presents advantages and disadvantages, depending on the size of the problem (number of functions of interest,  $N_f$ , and number of design variables,  $N_\alpha$ ).

### 3.4.1 Direct Method

In the direct method, the result of solving equation (3.13) for  $d\mathbf{q}/d\boldsymbol{\alpha}$  is substituted in equation (3.9), resulting in the direct sensitivity equations, given by

$$\frac{d\mathbf{f}}{d\boldsymbol{\alpha}} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\alpha}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \frac{d\mathbf{q}}{d\boldsymbol{\alpha}}, \quad (3.16)$$

$$\text{such that} \quad \frac{\partial \mathcal{R}}{\partial \mathbf{q}} \frac{d\mathbf{q}}{d\boldsymbol{\alpha}} = -\frac{\partial \mathcal{R}}{\partial \boldsymbol{\alpha}}.$$

From equation (3.13), it can be seen that each design variable  $\alpha_i$  will require one solve of the system of equations, making the computational cost of this approach proportional to the number of design variables,  $N_\alpha$ .

### 3.4.2 Adjoint Method

A different method can be taken for computing the total sensitivity  $d\mathbf{f}/d\boldsymbol{\alpha}$  by defining an auxiliary vector  $\boldsymbol{\psi}$  as

$$\boldsymbol{\psi}^T = \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \left[ \frac{\partial \mathcal{R}}{\partial \mathbf{q}} \right]^{-1} \quad (N_f \times N_q), \quad (3.17)$$

which can be rearranged into

$$\left[ \frac{\partial \mathcal{R}}{\partial \mathbf{q}} \right]^T \boldsymbol{\psi} = \left[ \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \right]^T. \quad (3.18)$$

The auxiliary vector  $\boldsymbol{\psi}$  is usually called the *adjoint vector* and, substituting it into equation (3.15) leads to the adjoint sensitivity equations - or the *dual problem* - given as

$$\frac{d\mathbf{f}}{d\boldsymbol{\alpha}} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\alpha}} - \boldsymbol{\psi}^T \frac{\partial \mathcal{R}}{\partial \boldsymbol{\alpha}}, \quad (3.19)$$

$$\text{such that} \quad \left[ \frac{\partial \mathcal{R}}{\partial \mathbf{q}} \right]^T \boldsymbol{\psi} = \left[ \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \right]^T.$$

Contrary to the direct method, the adjoint method does not require solving a system of equations  $N_\alpha$  times. Instead, it is solved  $N_f$  times, one for each function of interest  $f$ . This highlights the impact the choice of the method (direct vs. adjoint) to computing (3.15) has on the cost of the sensitivity analysis. For problems with a larger set of design variables than of functions of interest ( $N_\alpha \gg N_{\mathcal{I}}$ ) the adjoint method would be the most efficient, while for problems where the number of functions of interest greatly surpasses the number of design variables ( $N_{\mathcal{I}} \gg N_\alpha$ ), the direct method is the obvious choice.

### Discrete and continuous approaches

There are two ways of obtaining the adjoint equations of a system of PDE's, which differ by the order in which the discretization and linearisation are performed, as illustrated in figure 1.10. Following the dotted line in the figure, the *continuous adjoint* approach starts by linearising the non-linear PDE's followed by the forming of the adjoint equations, with the last step consisting in the discretization of these continuous adjoint equations. Following the solid line, the *discrete adjoint* approach first discretizes the non-linear PDE's which are then linearised and transposed.

Both approaches produce a set of discrete adjoint equations that, in theory, should be consistent and converge to the correct analytic value of the gradient of the function of interest in the limit of infinite grid resolution and given that the solutions are sufficiently smooth (e.g. no shocks) [108].

The choice of using the discrete or the continuous approach of the adjoint problem is not straightforward as both present their advantages and disadvantages [108–110].

One of the advantages of the discrete approach formulation is that it can be applied to any set of governing equations and, since the adjoint equations are derived from the discretized form of the flow governing equations, produces gradients consistent with the flow solver. This is an important feature in optimization, as inconsistencies may not allow the numerical optimizer to converge to a local minimum, due to the numerical gradient not being zero at that location. Regarding the functions of interest, the discrete approach presents the advantage over the continuous approach of being able to treat *arbitrary* functions. The latter can only treat specific forms of integral functions [37, 111]. Another advantage of the discrete formulation is the seamlessness with which the boundary conditions are handled, since the adjoint solver is derived from the discretized

flow residual equations that already implement them. In terms of implementation, the discrete approach presents a very interesting feature, which is allowing the use of AD tools in its derivation [33, 112, 113]. Using these tools to obtain the differentiated form of the discretized governing equations necessary to assemble the adjoint system of equations allows for a considerable reduction in development time.

The differentiation of the continuous governing equations is usually a more involved process than differentiating the discrete equations and a number of shortcuts must be included for complicated governing equations, such as the RANS equations with turbulence models [110]. In this case, the viscous effects might not be modeled in their entirety, and even if they are, the flow is usually assumed to be laminar and no turbulence model is used [37], with the viscosity and heat transfer ratio assumed to be independent from the flow, and kept constant when deriving the adjoint equations. The biggest advantage of the continuous approach is the reduced memory requirements of the adjoint solver, which are at the same level as the flow solver.

A summary of the advantages and disadvantages of both approaches is presented in table 3.1.

Discrete	Continuous
Advantages	
<ul style="list-style-type: none"> <li>• Applied to any set of equations</li> <li>• Gradient consistent with flow solver</li> <li>• Handles arbitrary functions of interest</li> <li>• Seamless boundary condition treatment</li> <li>• Can be derived using AD</li> </ul>	<ul style="list-style-type: none"> <li>• Reduced memory requirements</li> <li>• Clearer physical significance of adjoint variables and role of b.c.'s</li> </ul>
Disadvantages	
<ul style="list-style-type: none"> <li>• Increased memory requirements</li> </ul>	<ul style="list-style-type: none"> <li>• More involved derivation</li> <li>• Only allows function of interest in integral form</li> </ul>

Table 3.1: Comparison of discrete and continuous adjoint approaches.

### 3.5 Summary of the Various Approaches

Having completed the description of various approaches for the analysis of sensitivity of functions of interest, we now summarize their various features, advantages and disadvantages. Section 3.5 presents a qualitative prediction of computational cost, ease of implementation, fidelity and memory usage of the various methods, assuming they are to be used to compute the sensitivities that were mentioned in the first chapter of this document – sensitivities in multi-row turbomachinery simulations – which must be computed for a large number of design variables ( $N_\alpha \gg N_{\mathcal{I}}$ ), using relatively low CPU time and for general functions of interest. The first three methods presented in the table can be excluded

	CPU time	Ease of implementation	Fidelity	Memory usage
FD	High	Very Easy	Moderate	Low
CS	High	Moderate	High	Low
AD (fwd)	High	Complex	High	Low
AD (bwd)	Medium	Complex	High	High
Direct	High	Complex	High	Low
Adjoint (cont.)	Low	Complex	High	Low
Adjoint (disc. )	Low	Complex	High	Moderate

Table 3.2: Summary of various sensitivity analysis approaches.

due to their high CPU time requirements, as one of the goals of the sensitivities that we want to obtain is to be used in numerical optimization environments. Using AD for the whole solver (or mixing-plane interface) would prove to be impossible, as the complexity of the direct solver would make it impossible (and the code generated by the AD would probably be extremely inefficient). Using the semi-analytic direct approach would present the same problem of the first three methods that were excluded – high computational time requirements. We are then left with the adjoint approach, either continuous or discrete. The various advantages of the discrete approach, highlighted in table 3.1 led to the choice of the discrete adjoint method, particularly the option of deriving the discrete adjoint equations using AD tools. As such, the selected sensitivity analysis method was an hybrid, merging the discrete adjoint with automatic differentiation, which will be presented in the following section.

## 3.6 Hybrid ADjoint Approach

As mentioned in the previous section, the discrete adjoint approach allows the use of AD tools in the derivation of the adjoint equations [33, 112]. The hybrid ADjoint approach consists exactly in that. The total derivative is computed with the previously described adjoint method, and the partial derivatives are computed by routines obtained from automatic differentiation, as indicated in equation (3.20).

$$\frac{d\mathbf{f}}{d\boldsymbol{\alpha}} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\alpha}} - \boldsymbol{\psi}^T \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}}, \quad \text{such that} \quad \begin{bmatrix} \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \end{bmatrix}^T \boldsymbol{\psi} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \end{bmatrix}^T. \quad (3.20)$$

AD

This concept merges the advantages of the discrete adjoint approach with the reduced development time that automatic differentiation typically allows. The development of the solver using this approach consists in first defining the residual  $\mathbf{R}$  and functions of interest  $\mathbf{f}$  in terms of computer subroutines, that take as input the flow state solution  $\mathbf{q}$  and design variables  $\boldsymbol{\alpha}$  and output either the residual of a computational cell or the function of interest (or set of functions) and then applying an AD tool to those subroutines, thus obtaining differentiated routines that compute the required terms for the adjoint system of equations of equation (3.20). The hybrid ADjoint approach encompasses three major advantages: being largely automatic, being exactly consistent and generic [114].

- *Largely automatic*: Given the source code of the solver to be adjointed, the AD tool creates the code that produces the necessary terms of the discrete adjoint formulation;
- *Exactly consistent*: The process of automatic differentiation allows the exact treatment of arbitrarily complex expressions, and as such, the sensitivities produced by the differentiated code are perfectly consistent with those that would be obtained with the exact numerical differentiation of the original solver;
- *Generic*: A new formulation of the governing equations can be easily adjointed using this method or even a new set of governing equations.

# Chapter 4

## Derivation of the Adjoint of the Euler Equations

In this chapter, the derivation of the adjoint of the 2D Euler equations is made using both the continuous and discrete approaches. This serves the purpose of illustrating the adjoint formulation of flow governing PDE's, that constitute the core of the adjoint solver used in the simulations presented in this work. To this, a simplified form of the governing equations and the general procedure to derive the adjoint equations are first introduced, and then the adjoint formulation is derived using the two possible approaches. The chapter concludes with the proof that, in the limit, both the continuous and discrete adjoint approaches lead to the same adjoint equation

### 4.1 Euler Equations and Adjoint Procedure

Recalling the Navier-Stokes equations from Chapter 2, the Euler equations can be obtained by not taking the viscous fluxes into account, thus obtaining

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}_i}{\partial x_i} = 0. \quad (4.1)$$

The derivation of the adjoint equations is simplified by mapping the solution to a fixed computational domain with coordinates  $\xi_1$  and  $\xi_2$ , where

$$K_{n,m} = \left[ \frac{\partial x_n}{\partial \xi_m} \right] = T^{-1}, \quad K_{n,m}^{-1} = \left[ \frac{\partial \xi_n}{\partial x_m} \right] = T, \quad (4.2 \text{ a,b})$$

and

$$J = \det(K) = V, \quad S = JK^{-1}. \quad (4.3 \text{ a,b})$$

The elements of  $S$  are the cofactors of  $K$ , which, in a finite volume discretization are just the face areas of the computational cells projected in the  $x_i$  directions.

The general procedure to obtain the adjoint of a system of PDE's is described as [115]:

1. Derive the first variation of the flux gradient;
2. Multiply it by the Lagrange multiplier  $\phi$  and integrate over the domain  $\mathcal{D}$ ;
3. Subtract the integral obtained from the second step from the variation of the interest function;
4. Perform integration by parts to isolate the variation of the state vector terms,  $\delta \mathbf{q}$ , from the variation of the shape function,  $\delta f$ .

This procedure is the same for both the continuous and the discrete approaches.

## 4.2 Derivation of the Continuous Adjoint Equation

The Euler equations for steady state can be expressed as

$$\frac{\partial \mathcal{F}_k}{\partial \xi_k} = 0, \quad (4.4)$$

whose weak form, using the arbitrarily differentiable test vector  $\phi \neq 0$ , is

$$\mathcal{R} = \int_{\mathcal{D}} \phi^T \frac{\partial \mathcal{F}_k}{\partial \xi_k} d\mathcal{D} = 0, \quad (4.5)$$

where  $\mathcal{D}$  is the domain of integration. Since from differential calculus  $\int a \delta b = \int \delta(ab) - \int (\delta a)b$  [27], it yields

$$\int_{\mathcal{D}} \frac{\partial}{\partial \xi_k} (\phi^T \mathcal{F}_k) d\mathcal{D} - \int_{\mathcal{D}} \frac{\partial \phi^T}{\partial \xi_k} \mathcal{F}_k d\mathcal{D} = 0. \quad (4.6)$$

Using the divergence theorem [27] on the first term of the previous equation leads to

$$\int_{\mathcal{D}} \frac{\partial \phi^T}{\partial \xi_k} \mathcal{F}_k d\mathcal{D} - \int_{\mathcal{B}} n_k \phi^T \mathcal{F}_k d\mathcal{B}, = 0 \quad (4.7)$$

where  $n_k$  is the outward normal at the boundary  $\mathcal{B}$  (in the computational domain). The weak form of equation (4.7), for  $\delta \mathbf{q}$  is

$$\delta \mathcal{R} = \int_{\mathcal{D}} \frac{\partial \phi^T}{\partial \xi_k} \delta \mathcal{F}_k \, d\mathcal{D} - \int_{\mathcal{B}} (n_k \phi^T \delta \mathcal{F}_k) \, d\mathcal{B} = 0, \quad (4.8)$$

where

$$\delta \mathcal{F}_k = \mathbf{C}_k \delta \mathbf{q} + \delta S_{kl} \mathbf{f}_l, \quad (4.9)$$

which should hold for any differentiable test function  $\phi$ . The Jacobian matrices  $\mathbf{A}_k$  and  $\mathbf{C}_k$  are defined as

$$\mathbf{A}_k = \frac{\partial \mathbf{f}_k}{\partial \mathbf{q}}, \quad \mathbf{C}_k = S_{kl} \mathbf{A}_l. \quad (4.10 \text{ a,b})$$

Lets assume that we wish to control the pressure at the surface of an airfoil, like the one represented in figure 4.1 by varying its shape.

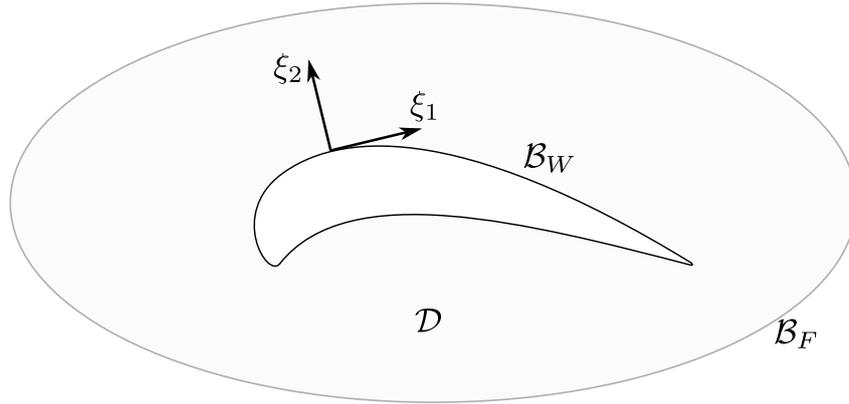


Figure 4.1: Schematic of an airfoil, domain of integration and respective boundaries.

By retaining a fixed computational domain, the variations in the shape will result in a corresponding variation in the mapping derivatives defined by  $K$ . Introducing the interest function

$$\mathcal{I} = \frac{1}{2} \int_{\mathcal{B}_W} (p - p_d)^2 \, ds, \quad (4.11)$$

where  $p_d$  is de desired pressure,  $ds$  represents the infinitesimal length on the airfoil surface and  $\mathcal{B}_W$  represents the surface of the airfoil. This is a typical function of interest in the case of inverse design, where the end goal is to achieve a specific pressure distribution over the airfoil surface [49]. This is attained with the minimization of  $\mathcal{I}$ . A variation in  $\mathcal{I}$  due

to a variation in the control function  $ds$  can be expressed as

$$\delta\mathcal{I} = \int_{\mathcal{B}_W} (p - p_d) \delta p \, ds + \frac{1}{2} \int_{\mathcal{B}_W} (p - p_d)^2 \delta(ds). \quad (4.12)$$

The pressure  $p$  depends on  $\mathbf{q}$  through the governing equations, therefore the variation  $\delta p$  is determined from the variation  $\delta\mathbf{q}$ . Since  $\delta\mathcal{R} = 0$ , we can sum equation (4.8) to equation (4.12), choosing the test function to be the Lagrange multiplier  $\mathbf{\Lambda}$ , yielding

$$\begin{aligned} \delta\mathcal{I} = & \int_{\mathcal{B}_W} (p - p_d) \delta p \, ds + \frac{1}{2} \int_{\mathcal{B}_W} (p - p_d)^2 \delta(ds) \\ & - \int_{\mathcal{D}} \frac{\partial \mathbf{\Lambda}^T}{\partial \xi_k} \delta \mathcal{F}_k \, d\mathcal{D} + \int_{\mathcal{B}} n_k \mathbf{\Lambda}^T \delta \mathcal{F}_k \, d\mathcal{B} \end{aligned} \quad (4.13)$$

The previous equation can be expanded as

$$\begin{aligned} \delta\mathcal{I} = & \int_{\mathcal{B}_W} (p - p_d) \delta p \, ds + \frac{1}{2} \int_{\mathcal{B}_W} (p - p_d)^2 \delta(ds) \\ & - \int_{\mathcal{D}} \frac{\partial \mathbf{\Lambda}^T}{\partial \xi_k} (\mathbf{C}_k \delta \mathbf{q} + \delta S_{kl} \mathbf{f}_l) \, d\mathcal{D} + \int_{\mathcal{B}} n_k \mathbf{\Lambda}^T \delta \mathcal{F}_k \, d\mathcal{B}. \end{aligned} \quad (4.14)$$

By selecting  $\mathbf{\Lambda}$  to be such that

$$- \frac{\partial \mathbf{\Lambda}^T}{\partial \xi_k} \mathbf{C}_k = 0 \quad \text{in} \quad \mathcal{D}, \quad (4.15)$$

equation (4.14) becomes

$$\begin{aligned} \delta\mathcal{I} = & \int_{\mathcal{B}_W} (p - p_d) \delta p \, ds + \frac{1}{2} \int_{\mathcal{B}_W} (p - p_d)^2 \delta(ds) \\ & - \int_{\mathcal{D}} \frac{\partial \mathbf{\Lambda}^T}{\partial \xi_k} \delta S_{kl} \mathbf{f}_l \, d\mathcal{D} + \int_{\mathcal{B}} n_k \mathbf{\Lambda}^T \delta \mathcal{F}_k \, d\mathcal{B}, \end{aligned} \quad (4.16)$$

which no longer depends explicitly on  $\delta\mathbf{q}$ . Equation (4.15) is the continuous adjoint equation and it can be driven to a converged steady-state solution by using pseudo time  $\tau$ , as

$$\frac{\partial \mathbf{\Lambda}}{\partial \tau} - \mathbf{C}_k^T \frac{\partial \mathbf{\Lambda}}{\partial \xi_k} = 0 \quad \text{in} \quad \mathcal{D}. \quad (4.17)$$

The variation  $\delta\mathcal{I}$  still has dependency on  $\delta\mathbf{q}$  through  $\delta p$  and  $\delta \mathcal{F}_k$ . To deal with this, the surface interval (last term of equation (4.13)) can first be split into the integral over the

wall (airfoil surface) and over the far field, as

$$\int_{\mathcal{B}} n_k \Lambda^T \delta \mathcal{F}_k \, d\mathcal{B} = \underbrace{\int_{\mathcal{B}_W} n_k \Lambda^T \delta \mathcal{F}_k \, d\mathcal{B}_W}_{\text{Wall}} + \underbrace{\int_{\mathcal{B}_F} n_k \Lambda^T \delta \mathcal{F}_k \, d\mathcal{B}_F}_{\text{Far Field}}. \quad (4.18)$$

As represented in figure 4.1,  $\xi_2$  is tangent to the surface of the airfoil, which can be represented by  $\xi_2 = 0$ , leading to the flow tangency condition of  $u_2 = 0$  in  $\mathcal{B}_W$  and the unit normal  $n_W = [0 \ 1]^T$ . By expanding the surface integral over the wall, and applying the described boundary conditions, we obtain

$$\begin{aligned} \int_{\mathcal{B}_W} n_k \Lambda^T \delta \mathcal{F}_k \, d\mathcal{B}_W &= \int_{\mathcal{B}_W} \Lambda^T \delta \mathcal{F}_2 \, d\xi_1 \\ &= \int_{\mathcal{B}_W} \Lambda^T \begin{bmatrix} 0 \\ S_{21} \delta p \\ S_{22} \delta p \\ 0 \end{bmatrix} d\xi_1 + \int_{\mathcal{B}_W} \Lambda^T \begin{bmatrix} 0 \\ \delta S_{21} p \\ \delta S_{22} p \\ 0 \end{bmatrix} d\xi_1. \end{aligned} \quad (4.19)$$

By letting  $\Lambda$  satisfy the boundary condition

$$\Lambda_j n_j = p - p_d \quad \text{on} \quad \mathcal{B}_W, \quad (4.20)$$

where

$$n_j = \frac{S_{2j}}{\sqrt{S_{2j} S_{2j}}}, \quad (4.21)$$

the first term of the RHS of equation (4.19) cancels out with the first term of equation (4.16).

The surface integral over the outer boundary (far-field) can be expanded as

$$\int_{\mathcal{B}_F} n_k \Lambda^T \delta \mathcal{F}_k \, d\mathcal{B}_F = \int_{\mathcal{B}_F} n_k \Lambda^T (\mathbf{C}_k \delta \mathbf{q} + \delta S_{kl} \mathbf{f}_l) \, d\mathcal{B}_F. \quad (4.22)$$

At this outer boundary, incoming characteristics for  $\Lambda$  correspond to outgoing characteristics for  $\delta \mathbf{q}$ . Consequently, the boundary condition for  $\Lambda$  can be chosen such that

$$n_k \Lambda^T \mathbf{C}_k \delta \mathbf{q} = 0 \quad \text{on} \quad \mathcal{B}_F, \quad (4.23)$$

thus canceling out the term  $\mathbf{C}_k \delta \mathbf{q}$  in equation (4.22). If the coordinate transformation is such that  $\delta S$  is negligible in the far-field, then the term  $\delta S_{kl} \mathbf{f}_l$  is also zero and equation (4.13) becomes

$$\delta \mathcal{I} = \frac{1}{2} \int_{\mathcal{B}_W} (p - p_d)^2 \delta(ds) - \int_{\mathcal{D}} \frac{\partial \mathbf{\Lambda}^T}{\partial \xi_k} \delta S_{kl} \mathbf{f}_l d\mathcal{D} - \int_{\mathcal{B}_W} (\delta S_{21} \mathbf{\Lambda}_2 + \delta S_{22} \mathbf{\Lambda}_3) p d\xi_1. \quad (4.24)$$

The first term of the RHS of equation (4.24) is a function of the direct flow solution,  $p$ , and the last term, of the adjoint solution  $\mathbf{\Lambda}$ . The second term is a function of both solutions ( $\mathcal{I}$  and  $\mathbf{\Lambda}$ ).

## Numerical Discretization

The adjoint equation given by equation (4.17) can be expanded for a two-dimensional problem as

$$\frac{\partial \mathbf{\Lambda}}{\partial \tau} - \mathbf{C}_1^T \frac{\partial \mathbf{\Lambda}}{\partial \xi} - \mathbf{C}_2^T \frac{\partial \mathbf{\Lambda}}{\partial \eta} = 0, \quad (4.25)$$

with  $\xi = \xi_1$  and  $\eta = \xi_2$ . Discretizing the convective adjoint flux using a second-order central spatial discretization leads to

$$V \frac{\partial \mathbf{\Lambda}_{i,j}}{\partial \tau} = \frac{1}{2} \left[ \mathbf{C}_1^T (\mathbf{\Lambda}_{i+1,j} - \mathbf{\Lambda}_{i-1,j}) + \mathbf{C}_2^T (\mathbf{\Lambda}_{i,j+1} - \mathbf{\Lambda}_{i,j-1}) \right], \quad (4.26)$$

where  $V$  is the cell area and  $(i, j)$  corresponds to the computational indexes in the  $(\xi, \eta)$  directions, respectively. From equation (4.10 a,b), the Jacobian fluxes can be expanded as

$$\mathbf{C}_{1,i,j}^T = S_{11,i,j} \mathbf{A}_{1,i,j}^T + S_{12,i,j} \mathbf{A}_{2,i,j}^T, \quad (4.27a)$$

and

$$\mathbf{C}_{2,i,j}^T = S_{21,i,j} \mathbf{A}_{2,i,j}^T + S_{22,i,j} \mathbf{A}_{2,i,j}^T, \quad (4.27b)$$

where

$$S_{11,i,j} = \frac{1}{2} \left( S_{11,i+\frac{1}{2},j} + S_{11,i-\frac{1}{2},j} \right), \quad S_{12,i,j} = \frac{1}{2} \left( S_{12,i+\frac{1}{2},j} + S_{12,i-\frac{1}{2},j} \right), \quad (4.28 \text{ a,b})$$

and

$$\mathbf{A}_{1,i,j}^T = \left[ \frac{\partial \mathbf{f}_1}{\partial \mathbf{q}} \right]_{i,j}^T, \quad \mathbf{A}_{2,i,j}^T = \left[ \frac{\partial \mathbf{f}_2}{\partial \mathbf{q}} \right]_{i,j}^T. \quad (4.29 \text{ a,b})$$

Reducing the number of subscripts and simplifying the notation by defining the Euler Jacobian matrices as

$$\widehat{\mathbf{A}}_{i,j}^T = \mathbf{C}_{1,i,j}^T, \quad \widehat{\mathbf{B}}_{i,j}^T = \mathbf{C}_{2,i,j}^T, \quad \mathbf{A}_{i,j}^T = \mathbf{A}_{1,i,j}^T, \quad \mathbf{B}_{i,j}^T = \mathbf{A}_{2,i,j}^T, \quad (4.30 \text{ a-d})$$

yields the discretized continuous adjoint residual, written as

$$\mathbf{R}(\boldsymbol{\Lambda})_{i,j} = \frac{1}{2} \left[ \widehat{\mathbf{A}}_{i,j}^T (\boldsymbol{\Lambda}_{i+1,j} - \boldsymbol{\Lambda}_{i-1,j}) + \widehat{\mathbf{B}}_{i,j}^T (\boldsymbol{\Lambda}_{i,j+1} - \boldsymbol{\Lambda}_{i,j-1}) \right]. \quad (4.31)$$

### 4.3 Derivation of the Discrete Adjoint Equations

The discrete adjoint equations are obtained by applying control theory directly to the set of discrete field equations, following the same step sequence as in the continuous approach.

The equations resulting in the discrete approach will depend on the scheme used to discretize the flow equations and so will their complexity. For the derivation of the discrete adjoint equations, the previously described cell-centered, second order spacial discretization is used and only the convective flux gradients are considered for simplicity.

We first start by defining the variation of the discrete residual as

$$\delta \mathbf{R}(\mathbf{q})_{ij} = \delta \mathbf{h}_{i+\frac{1}{2},j} - \delta \mathbf{h}_{i-\frac{1}{2},j} + \delta \mathbf{h}_{i,j+\frac{1}{2}} - \delta \mathbf{h}_{i,j-\frac{1}{2}} \quad (4.32)$$

with

$$\delta \mathbf{h}_{i\pm\frac{1}{2},j} = \delta \mathbf{f}_{1,i\pm\frac{1}{2},j} \quad \text{and} \quad \delta \mathbf{h}_{i,j\pm\frac{1}{2}} = \delta \mathbf{f}_{2,i,j\pm\frac{1}{2}}, \quad (4.33 \text{ a,b})$$

where  $\mathbf{f}_1$  and  $\mathbf{f}_2$  are the convective flux gradients in the two directions and their variation  $\delta \mathbf{f}$  being the abbreviation of

$$\delta \mathbf{f} = \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \delta \mathbf{q}. \quad (4.34)$$

The next step consists in pre-multiplying the variation of the discrete residual  $\mathbf{R}$  by the Lagrange multiplier  $\boldsymbol{\psi}$  and sum the product over the computation domain, thus producing

$$\sum_{i=1}^{N_\xi} \sum_{j=1}^{N_\eta} \boldsymbol{\psi}_{ij}^T \delta \mathbf{R}(\mathbf{q})_{ij} = 0, \quad (4.35)$$

where  $N_\xi$  and  $N_\eta$  are the number of computational cells along each direction.

The third step consists in adding equation (4.35) to the variation of the discrete function of interest,  $\delta\mathcal{I}_c$ , yielding

$$\delta\mathcal{I} = \delta\mathcal{I}_c + \sum_{i=1}^{N_\xi} \sum_{j=1}^{N_\eta} \boldsymbol{\psi}_{ij}^T \delta\mathbf{R}(\mathbf{q})_{ij}, \quad (4.36)$$

where  $\mathcal{I}_c$  represents the discretized function of interest. The continuous adjoint partial differential system of equations is formulated employing integration by parts. In the discrete approach, the discrete counterpart to integration by parts, the summation by parts, is used instead.

The final set of discrete adjoint equations is produced by expanding  $\delta\mathbf{R}(\mathbf{q})$  for  $(i, j)$  and the four adjacent cells, which is then multiplied by the Lagrange multiplier,  $\boldsymbol{\psi}_{i,j}$ . The last step consists in collecting any term that is multiplied by  $\delta\mathbf{q}_{i,j}$ .

As we are dealing with the Euler equations, the only contribution that must be considered is the contribution from the convective terms and artificial dissipation. The latter are not considered for simplicity, but its handling can be found in [115, 116]. The variation of the convective flux computed at the cell faces can be written as

$$\delta\mathbf{f}_{i\pm\frac{1}{2},j} = \frac{1}{2} (\delta\mathbf{f}_{i\pm 1,j} + \delta\mathbf{f}_{i,j}) \quad (4.37a)$$

and

$$\delta\mathbf{f}_{i,j\pm\frac{1}{2}} = \frac{1}{2} (\delta\mathbf{f}_{i,j\pm 1} + \delta\mathbf{f}_{i,j}). \quad (4.37b)$$

These convective flux variations can also be expressed as the function of the fluxes defined in the physical space,  $\mathbf{f}$ , yielding

$$\delta\mathbf{f}_{i\pm\frac{1}{2},j} = \frac{1}{2} \left[ \delta \left( S_{1m_{i\pm\frac{1}{2},j}} \mathbf{f}_{m_{i\pm 1,j}} \right) + \delta \left( S_{1m_{i\pm\frac{1}{2},j}} \mathbf{f}_{m_{i,j}} \right) \right] \quad (4.38a)$$

and

$$\delta\mathbf{f}_{i,j\pm\frac{1}{2}} = \frac{1}{2} \left[ \delta \left( S_{2m_{i,j\pm\frac{1}{2}}} \mathbf{f}_{m_{i,j\pm 1}} \right) + \delta \left( S_{2m_{i,j\pm\frac{1}{2}}} \mathbf{f}_{m_{i,j}} \right) \right]. \quad (4.38b)$$

The previous equations can be expanded as

$$\begin{aligned} \delta \mathbf{f}_{i \pm \frac{1}{2}, j} = & \frac{1}{2} \left[ \delta S_{1m_{i \pm \frac{1}{2}, j}} \mathbf{f}_{m_{i \pm 1, j}} + \delta S_{1m_{i \pm \frac{1}{2}, j}} \mathbf{f}_{m_{i, j}} \right. \\ & \left. + S_{1m_{i \pm \frac{1}{2}, j}} \delta \mathbf{f}_{m_{i \pm 1, j}} + S_{1m_{i \pm \frac{1}{2}, j}} \delta \mathbf{f}_{m_{i, j}} \right] \end{aligned} \quad (4.39a)$$

and

$$\begin{aligned} \delta \mathbf{f}_{i, j \pm \frac{1}{2}} = & \frac{1}{2} \left[ \delta S_{2m_{i, j \pm \frac{1}{2}}} \mathbf{f}_{m_{i, j \pm 1}} + \delta S_{2m_{i, j \pm \frac{1}{2}}} \mathbf{f}_{m_{i, j}} \right. \\ & \left. + S_{2m_{i, j \pm \frac{1}{2}}} \delta \mathbf{f}_{m_{i, j \pm 1}} + S_{2m_{i, j \pm \frac{1}{2}}} \delta \mathbf{f}_{m_{i, j}} \right], \end{aligned} \quad (4.39b)$$

which becomes

$$\begin{aligned} \delta \mathbf{f}_{i \pm \frac{1}{2}, j} = & \frac{1}{2} \left[ \delta S_{1m_{i \pm \frac{1}{2}, j}} \mathbf{f}_{m_{i \pm 1, j}} + \delta S_{1m_{i \pm \frac{1}{2}, j}} \mathbf{f}_{m_{i, j}} \right. \\ & \left. + S_{1m_{i \pm \frac{1}{2}, j}} \left( \frac{\partial \mathbf{f}_m}{\partial \mathbf{q}} \delta \mathbf{q} \right)_{i \pm 1, j} + S_{1m_{i \pm \frac{1}{2}, j}} \left( \frac{\partial \mathbf{f}_m}{\partial \mathbf{q}} \delta \mathbf{q} \right)_{i, j} \right] \end{aligned} \quad (4.40a)$$

and

$$\begin{aligned} \delta \mathbf{f}_{i, j \pm \frac{1}{2}} = & \frac{1}{2} \left[ \delta S_{2m_{i, j \pm \frac{1}{2}}} \mathbf{f}_{m_{i, j \pm 1}} + \delta S_{2m_{i, j \pm \frac{1}{2}}} \mathbf{f}_{m_{i, j}} \right. \\ & \left. + S_{2m_{i, j \pm \frac{1}{2}}} \left( \frac{\partial \mathbf{f}_m}{\partial \mathbf{q}} \delta \mathbf{q} \right)_{i, j \pm 1} + S_{2m_{i, j \pm \frac{1}{2}}} \left( \frac{\partial \mathbf{f}_m}{\partial \mathbf{q}} \delta \mathbf{q} \right)_{i, j} \right]. \end{aligned} \quad (4.40b)$$

Substituting equation (4.40) into equation (4.32) yields, after discarding the terms including the variation  $\delta S$  which does not contribute to the adjoint equations,

$$\begin{aligned} \delta \mathbf{R}(\mathbf{q})_{i, j} = & \frac{1}{2} \left[ \left( S_{1m_{i + \frac{1}{2}, j}} - S_{1m_{i - \frac{1}{2}, j}} + S_{2m_{i, j + \frac{1}{2}}} - S_{2m_{i, j - \frac{1}{2}}} \right) \frac{\partial \mathbf{f}_{m_{i, j}}}{\partial \mathbf{q}} \delta \mathbf{q}_{i, j} \right. \\ & + S_{1m_{i + \frac{1}{2}, j}} \frac{\partial \mathbf{f}_{m_{i+1, j}}}{\partial \mathbf{q}} \delta \mathbf{q}_{i+1, j} - S_{1m_{i - \frac{1}{2}, j}} \frac{\partial \mathbf{f}_{m_{i-1, j}}}{\partial \mathbf{q}} \delta \mathbf{q}_{i-1, j} \\ & \left. + S_{2m_{i, j + \frac{1}{2}}} \frac{\partial \mathbf{f}_{m_{i, j+1}}}{\partial \mathbf{q}} \delta \mathbf{q}_{i, j+1} - S_{2m_{i, j - \frac{1}{2}}} \frac{\partial \mathbf{f}_{m_{i, j-1}}}{\partial \mathbf{q}} \delta \mathbf{q}_{i, j-1} \right]. \end{aligned} \quad (4.41)$$

From the previous equation, we can see that the linearized residual has contributions from all four adjacent cells,  $(i \pm 1, j)$  and  $(i, j \pm 1)$ . Therefore, the adjoint residual will only have contributions from the same four cells. The sum over the entire domain of the product of the residual by the transpose of the Lagrange multiplier vector, after collecting the terms

with  $\delta \mathbf{q}_{i,j}$  and transposing equation [115], results in

$$\begin{aligned} \mathbf{R}(\boldsymbol{\psi})_{i,j} = & -\frac{1}{2} \left[ S_{1m_{i+\frac{1}{2},j}} \mathbf{A}_{m_{i,j}}^T (\boldsymbol{\psi}_{i+1,j} - \boldsymbol{\psi}_{i,j}) - S_{1m_{i-\frac{1}{2},j}} \mathbf{A}_{m_{i,j}}^T (\boldsymbol{\psi}_{i-1,j} - \boldsymbol{\psi}_{i,j}) \right. \\ & \left. + S_{2m_{i,j+\frac{1}{2}}} \mathbf{A}_{m_{i,j}}^T (\boldsymbol{\psi}_{i,j+1} - \boldsymbol{\psi}_{i,j}) - S_{2m_{i,j-\frac{1}{2}}} \mathbf{A}_{m_{i,j}}^T (\boldsymbol{\psi}_{i,j-1} - \boldsymbol{\psi}_{i,j}) \right]. \end{aligned} \quad (4.42)$$

Redefining  $\widehat{\mathbf{A}}^T$  and  $\widehat{\mathbf{B}}^T$  as

$$\widehat{\mathbf{A}}^T = S_{11_{i+\frac{1}{2},j}} \mathbf{A}_{ij}^T + S_{12_{i+\frac{1}{2},j}} \mathbf{B}_{ij}^T \quad \text{and} \quad \widehat{\mathbf{B}}^T = S_{21_{i+\frac{1}{2},j}} \mathbf{A}_{ij}^T + S_{22_{i+\frac{1}{2},j}} \mathbf{B}_{ij}^T, \quad (4.43)$$

yields

$$\begin{aligned} \mathbf{R}_{i,j}(\boldsymbol{\psi}) = & \frac{1}{2} \left[ \widehat{\mathbf{A}}^T_{i-\frac{1}{2},j} (\boldsymbol{\psi}_{ij} - \boldsymbol{\psi}_{i-1,j}) + \widehat{\mathbf{A}}^T_{i+\frac{1}{2},j} (\boldsymbol{\psi}_{i+1,j} - \boldsymbol{\psi}_{ij}) \right. \\ & \left. \widehat{\mathbf{B}}^T_{i,j+\frac{1}{2}} (\boldsymbol{\psi}_{i,j+1} - \boldsymbol{\psi}_{i,j}) + \widehat{\mathbf{B}}^T_{i,j-\frac{1}{2}} (\boldsymbol{\psi}_{i,j} - \boldsymbol{\psi}_{i,j-1}) \right]. \end{aligned} \quad (4.44)$$

Equation (4.44) is identical to the discretized continuous adjoint equation (4.31) derived in the previous section, which illustrates the similarity between the discretization of the continuous and discrete convective fluxes. They do however differ in the manner in which the metrics are calculated at each cell. While in the discretization of the continuous equations, the metrics across the cell faces are averaged for each cell in each direction, it is not so in the calculation of the discrete adjoint flux.

A further analysis can be done to compare the continuous with the discrete equations, by assuming an infinitely discretized mesh. In this limit that the mesh cell size reduces to zero, the adjoint convective flux can be written as

$$\lim_{\Delta \xi \rightarrow 0, \Delta \eta \rightarrow 0} \mathbf{R}(\boldsymbol{\psi}) = \frac{1}{2} \left[ \widehat{\mathbf{A}}^T_{ij} (\boldsymbol{\psi}_{i+1,j} - \boldsymbol{\psi}_{i-1,j}) + \widehat{\mathbf{B}}^T_{ij} (\boldsymbol{\psi}_{i,j+1} - \boldsymbol{\psi}_{i,j-1}) \right]. \quad (4.45)$$

The second-order central difference of the Lagrange multipliers can then be reduced to

$$\lim_{\Delta \xi \rightarrow 0} \left[ \frac{\boldsymbol{\psi}_{i+1,j} - \boldsymbol{\psi}_{i-1,j}}{2} \right] = \frac{\partial \boldsymbol{\psi}}{\partial \xi}, \quad (4.46)$$

which allows the adjoint convective flux term to be written in continuous form as

$$\lim_{\Delta \xi \rightarrow 0, \Delta \eta \rightarrow 0} \mathbf{R}(\mathbf{q}) = \widehat{\mathbf{A}}^T \frac{\partial \boldsymbol{\psi}}{\partial \xi} + \widehat{\mathbf{B}}^T \frac{\partial \boldsymbol{\psi}}{\partial \eta}. \quad (4.47)$$

If the same notation as in equation (4.30 a-d) is used, the continuous form of the discrete adjoint convective flux can be expressed as

$$\mathbf{R}(\boldsymbol{\psi}) = \widehat{\mathbf{A}}^T \frac{\partial \boldsymbol{\psi}}{\partial \xi} + \widehat{\mathbf{B}}^T \frac{\partial \boldsymbol{\psi}}{\partial \eta} \quad (4.48a)$$

$$= \mathbf{C}_1^T \frac{\partial \boldsymbol{\psi}}{\partial \xi} + \mathbf{C}_2^T \frac{\partial \boldsymbol{\psi}}{\partial \eta} \quad (4.48b)$$

$$= \mathbf{C}_k^T \frac{\partial \boldsymbol{\psi}}{\partial \xi_k}, \quad (4.48c)$$

which is identical to the continuous adjoint equation.



# Chapter 5

## Adjoint Multi-row Turbomachinery

### Interface

Chapter 2 described various approaches to handle the interaction between adjacent rows in a turbomachinery CFD simulation. As previously mentioned, the goal of the work presented in this document was to extend the capability of a legacy adjoint solver to compute sensitivity information for multi-row problems, which are typical in turbomachinery designs.

The direct flow solver TACOMA [94, 96], on which the existing adjoint solver was implemented [113], uses the mixing-plane approach described in chapter 2 to couple multiple rows in steady state simulations. As such, the adjoint multi-row interface that was implemented and which formulation and implementation is presented in this chapter is based on the same mixing-plane algorithm.

### 5.1 Adjoint Multi-row Formulation

If we do not take into account the interaction between rows, for the  $i^{\text{th}}$  of  $N_r$  blade rows (as represented in figure 5.1), we have  $N_r$  independent systems of adjoint equations, one for each blade row,

$$\left[ \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right]_{i,i}^T \boldsymbol{\psi}_i = \left[ \frac{\partial \mathcal{I}}{\partial \mathbf{q}} \right]_i^T, \quad i = 1, \dots, N_r. \quad (5.1)$$

However, since we are interested in taking into account the coupling among rows, that are found to be of utmost importance in multi-row turbomachines, then the dependence of the residual of cells of one row on cells of other rows must be considered, resulting in

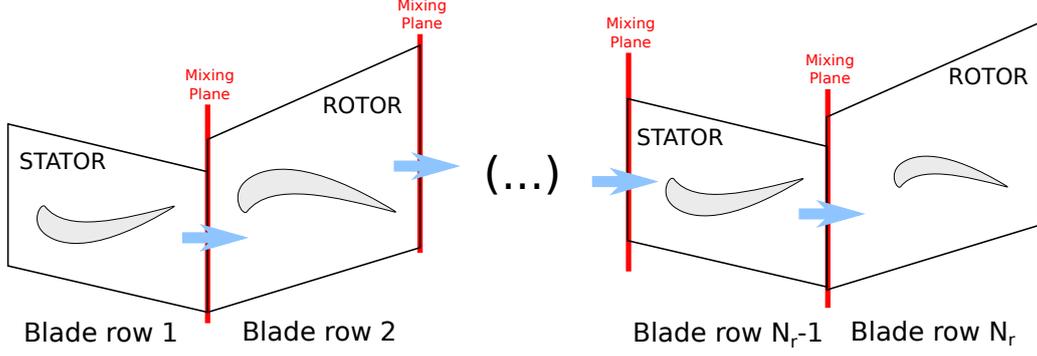


Figure 5.1: Schematic of a multistage turbomachine modeled with a sequence of  $N_r$  single blade passage rows.

the coupled system of equations given as

$$\begin{bmatrix} \left[ \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right]_{1,1}^T & \cdots & \left[ \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right]_{1,N_r}^T \\ \vdots & \ddots & \vdots \\ \left[ \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right]_{N_r,1}^T & \cdots & \left[ \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right]_{N_r,N_r}^T \end{bmatrix} \begin{Bmatrix} \psi_1 \\ \vdots \\ \psi_{N_r} \end{Bmatrix} = \begin{bmatrix} \left[ \frac{\partial \mathcal{I}}{\partial \mathbf{q}} \right]_i^T \\ \vdots \\ \left[ \frac{\partial \mathcal{I}}{\partial \mathbf{q}} \right]_{N_r}^T \end{bmatrix}$$

where the term  $[\partial \mathbf{R} / \partial \mathbf{q}]_{i,j}$  represents the influence of row  $j$  in the residual of row  $i$ . As each row only influences its neighbors (adjacent rows), we have that

$$\left[ \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right]_{i,j} = 0, \quad i-1 > j > i+1. \quad (5.2)$$

Therefore, if we want to find the adjoint solution in a global computational domain composed as various row sub domains, we have to solve the single-row solutions with additional coupling terms, given by the adjoint of the coupling procedure used in the direct solver.

Assuming a simplified case of only two adjacent blade rows, such as a single stage comprised of stator-rotor, the coupled adjoint system reduces to

$$\begin{bmatrix} \left[ \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right]_{1,1}^T & \left[ \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right]_{1,2}^T \\ \left[ \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right]_{2,1}^T & \left[ \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right]_{2,2}^T \end{bmatrix} \begin{Bmatrix} \psi_1 \\ \psi_2 \end{Bmatrix} = \begin{bmatrix} \left[ \frac{\partial \mathcal{I}}{\partial \mathbf{q}} \right]_1^T \\ \left[ \frac{\partial \mathcal{I}}{\partial \mathbf{q}} \right]_2^T \end{bmatrix}. \quad (5.3)$$

The mixing-plane algorithm works in both directions, with every row being simultaneously a donor and a receiver. For simplicity of the following derivation, we shall consider only one direction of influence and, as such we can define one row as the donor and other as

the receiver,

$$\frac{\partial \mathbf{R}_2}{\partial \mathbf{q}_1} \equiv \frac{\partial \mathbf{R}_{\text{rec}}}{\partial \mathbf{q}_{\text{don}}}. \quad (5.4)$$

The chain rule can be applied in the computation of the coupling non-zero off-diagonal terms to distinguish the *single-row* term  $\partial \mathbf{R}_{\text{rec}} / \partial \mathbf{q}_{\text{local}}^*$  from a term that represents the influence of the state solution of the adjacent domain,  $\mathbf{q}_{\text{don}}$  on the updated state solution  $\mathbf{q}_{\text{local}}^*$ , thus obtaining

$$\frac{\partial \mathbf{R}_{\text{rec}}}{\partial \mathbf{q}_{\text{don}}} = \frac{\partial \mathbf{R}_{\text{rec}}}{\partial \mathbf{q}_{\text{local}}^*} \frac{d\mathbf{q}_{\text{local}}^*}{d\mathbf{q}_{\text{don}}}. \quad (5.5)$$

Recalling the mixing-plane algorithm described in subsection 2.3.2, the updated state solution can be represented as a function of the various terms computed during the multi-row exchange yielding

$$\mathbf{q}_{\text{local}}^* = f\left(\mathbf{p}_{\text{rec}}^*(\mathbf{p}_{\text{don}}(\mathbf{q}_{\text{don}}), \mathbf{p}_{\text{local}}(\mathbf{q}_{\text{local}})), \mathbf{q}_{\text{local}}\right). \quad (5.6)$$

An expression for the multi-row coupling term can be obtained by differentiating each of the terms identified above and applying the chain rule, thus obtaining

$$\frac{d\mathbf{q}_{\text{local}}^*}{d\mathbf{q}_{\text{don}}} = \frac{\partial \mathbf{q}_{\text{local}}^*}{\partial \mathbf{p}_{\text{rec}}^*} \frac{\partial \mathbf{p}_{\text{rec}}^*}{\partial \mathbf{p}_{\text{don}}} \frac{\partial \mathbf{p}_{\text{don}}}{\partial \mathbf{q}_{\text{don}}}. \quad (5.7)$$

The previous expression regards only the dependence on the cells across the multi-row interface. However, there is also a new dependency on the cells of the local face due to the multi-row boundary condition, yielding

$$\frac{\partial \mathbf{R}_{\text{rec}}}{\partial \mathbf{q}_{\text{local}}} = \frac{\partial \mathbf{R}_{\text{rec}}}{\partial \mathbf{q}_{\text{local}}^*} \frac{d\mathbf{q}_{\text{local}}^*}{d\mathbf{q}_{\text{local}}}, \quad (5.8)$$

where the term  $d\mathbf{q}_{\text{local}}^*/d\mathbf{q}_{\text{local}}$  being given by

$$\frac{d\mathbf{q}_{\text{local}}^*}{d\mathbf{q}_{\text{local}}} = \frac{\partial \mathbf{q}_{\text{local}}^*}{\partial \mathbf{p}_{\text{rec}}^*} \frac{\partial \mathbf{p}_{\text{rec}}^*}{\partial \mathbf{p}_{\text{local}}} \frac{\partial \mathbf{p}_{\text{local}}}{\partial \mathbf{q}_{\text{local}}} + \frac{\partial \mathbf{q}_{\text{local}}^*}{\partial \mathbf{q}_{\text{local}}}. \quad (5.9)$$

The first term of the RHS of equation (5.9) increases the stencil of the residual calculation to cover at least a whole row of cells in the radial position of each cell of the single stage stencil that belongs to the multi-row interface. The second term comes from the non-reflectivity boundary conditions and also increases the stencil of influence to cover a certain number of radial rows of the mixing-plane face.

To obtain the total derivative given by equation (3.19) it is also necessary to compute the term  $\partial \mathbf{R}/\partial \boldsymbol{\alpha}$  with the coupling taken into account. Similarly to  $\partial \mathbf{R}/\partial \mathbf{q}$ , for multi-row simulations, the adjoint system of equations becomes a coupled system of equations given as

$$\left[ \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}} \right]_{\text{multi-row}} = \begin{bmatrix} \left[ \frac{\partial \mathbf{R}_1}{\partial \boldsymbol{\alpha}_1} \right] & \cdots & \left[ \frac{\partial \mathbf{R}_1}{\partial \boldsymbol{\alpha}_{N_r}} \right] \\ \vdots & \ddots & \vdots \\ \left[ \frac{\partial \mathbf{R}_{N_r}}{\partial \boldsymbol{\alpha}_1} \right] & \cdots & \left[ \frac{\partial \mathbf{R}_{N_r}}{\partial \boldsymbol{\alpha}_{N_r}} \right] \end{bmatrix}, \quad (5.10)$$

where the diagonal submatrices  $\partial \mathbf{R}_i/\partial \boldsymbol{\alpha}_i$  represent the individual single-row systems of equations and the off-diagonal submatrices introduce the coupling between rows.

The present work assumes inlet and outlet boundary conditions  $\mathbf{U}$  and computational grid coordinates  $\mathbf{X}$  as possible design variables  $\boldsymbol{\alpha}$ , representing operating conditions and blade/hub shape, respectively.

For the case of boundary conditions  $\mathbf{U}$ , since the inlet and outlet surfaces are either the first inlet or last outlet of the coupled domains, there the off-diagonal terms of  $\partial \mathbf{R}/\partial \mathbf{U}$  are zero, as there is no imposition of traditional boundary conditions in the mixing-plane interface.

There is, however, a dependence of the updated state on the grid coordinates  $\mathbf{X}$  of the adjacent domain. Therefore, it is necessary to take the multi-row coupling into account in its computation, if the grid coordinates are chosen as the design variables. In this case, the coupling terms in  $\partial \mathbf{R}/\partial \mathbf{X}$  are given by

$$\frac{\partial \mathbf{R}_{\text{rec}}}{\partial \mathbf{X}_{\text{don}}} = \frac{\partial \mathbf{R}_{\text{rec}}}{\partial \mathbf{q}_{\text{local}}^*} \frac{d\mathbf{q}_{\text{local}}^*}{d\mathbf{X}_{\text{don}}} \quad (5.11)$$

and

$$\frac{\partial \mathbf{R}_{\text{rec}}}{\partial \mathbf{X}_{\text{local}}} = \frac{\partial \mathbf{R}_{\text{rec}}}{\partial \mathbf{q}_{\text{local}}^*} \frac{d\mathbf{q}_{\text{local}}^*}{d\mathbf{X}_{\text{local}}}, \quad (5.12)$$

where the terms  $d\mathbf{q}_{\text{local}}^*/d\mathbf{X}_{\text{don}}$  and  $d\mathbf{q}_{\text{local}}^*/d\mathbf{X}_{\text{local}}$  reflect the dependency of the local state solution on the computational grid coordinates of the donor and local cells, respectively. Similarly to equations (5.7) and (5.9), this coupling terms can be obtained using the chain rule for derivatives as

$$\frac{d\mathbf{q}_{\text{local}}^*}{d\mathbf{X}_{\text{don}}} = \frac{\partial \mathbf{q}_{\text{local}}^*}{\partial \mathbf{p}_{\text{rec}}^*} \frac{\partial \mathbf{p}_{\text{rec}}^*}{\partial \mathbf{p}_{\text{don}}} \frac{\partial \mathbf{p}_{\text{don}}}{\partial \mathbf{X}_{\text{don}}}, \quad (5.13)$$

and

$$\frac{d\mathbf{q}_{\text{local}}^*}{d\mathbf{X}_{\text{local}}} = \frac{\partial \mathbf{q}_{\text{local}}^*}{\partial \mathbf{p}_{\text{rec}}^*} \frac{\partial \mathbf{p}_{\text{rec}}^*}{\partial \mathbf{p}_{\text{local}}} \frac{\partial \mathbf{p}_{\text{local}}}{\partial \mathbf{X}_{\text{local}}} + \frac{\partial \mathbf{q}_{\text{local}}^*}{\partial \mathbf{X}_{\text{local}}}. \quad (5.14)$$

## 5.2 Implementation of the Adjoint Steady Mixing Plane Interface

Having defined the necessary terms to compute the off-diagonal terms of the multi-row adjoint system of section 5.1, the ideal procedure with Automatic Differentiation (AD) would be to give the direct solver code responsible for the mixing-plane interface and obtain differentiated code that would produce those terms. It is however not that straightforward. The complexity of the code, MPI communications and use of dynamic memory allocation, amongst others, require a special treatment of the code before the AD tools can be used.

As such, the implementation of the previously described adjoint steady mixing-plane interface consisted in four steps:

1. Rewrite the original routines into routines that can easily be differentiated using AD tools;
2. Differentiate a set of routines to obtain the various terms of the right-hand side of equations (5.7) and (5.13);
3. Assemble the routines in order to obtain the left-hand side term of equations (5.7), (5.9), (5.13) and (5.14);
4. Implement the calculations of the previous step in the assembling of the global matrix in equations (5.1) and (5.10).

A detailed description of each individual step is presented in the following subsections 5.2.1 to 5.2.4.

Step 4, albeit being the last step in the implementation of the adjoint mixing-plane, was performed, in a way, before all the other steps, to assure that the modifications, differentiation and hand-assembly of the differentiated routines would result in a routine consistent with the existent adjoint solver and could be correctly introduced in the chain of derivatives of the adjoint solver. In the original iterative procedure, the mixing-plane exchange occurs after a certain number of time steps (here considered in every time step,

for simplicity), as represented by algorithm 1. The implementation of the adjoint solver is

---

**Algorithm 1:** Original simplified iterative cycle.

---

```
1 initialization;  
2 while not converged do  
3   | while simple cycle do  
4   |   | compute residual;  
5   |   | time step;  
6   |   | update boundary conditions;  
7   | end  
8   | mixing-plane exchange;  
9   | check convergence;  
10 end
```

---

based on the fact that at convergence the updates of the solution are negligible, and, as such, the iterative loop can be removed to represent the last iterative step, as represented in algorithm 2. As such, one possible approach would be to implement the mixing-plane

---

**Algorithm 2:** Simplification of the last step of the convergence iteration.

---

```
1 initialization;  
2 mixing-plane exchange;  
3 update boundary conditions;  
4 compute residual;
```

---

boundary condition update inside the rewritten residual computation routines and feed the updated routine to the AD tool. Another approach would be to keep the adjoint mixing-plane as a separate routine which would be integrated with the single-row routine at the assembly of the left-hand side of the adjoint system of section 5.1. This approach also presents a closer similarity to the original structure of the code, and, as such, it was the selected approach.

## 5.2.1 Rewriting of Original routines

The first step comes from the way the original flow solver was implemented, which makes use of modification to data structures contained in modules and, as such, are not explicitly stated in the arguments of the various routines. These data structures were also implemented making use of dynamic allocation of arrays in such a way that it was not possible to use them directly with the AD tool Tapenade. The use of a custom MPI communication interface also lead to the decision of removing all the MPI communications from the

routines to be differentiated and implementing them by hand in the adjoined version of the main mixing-plane routine.

The main original mixing-plane routine, `ms_exchange`, is schematically represented in figure 5.2. Inside this main routine, the routine `ms_compute_profile` does the com-

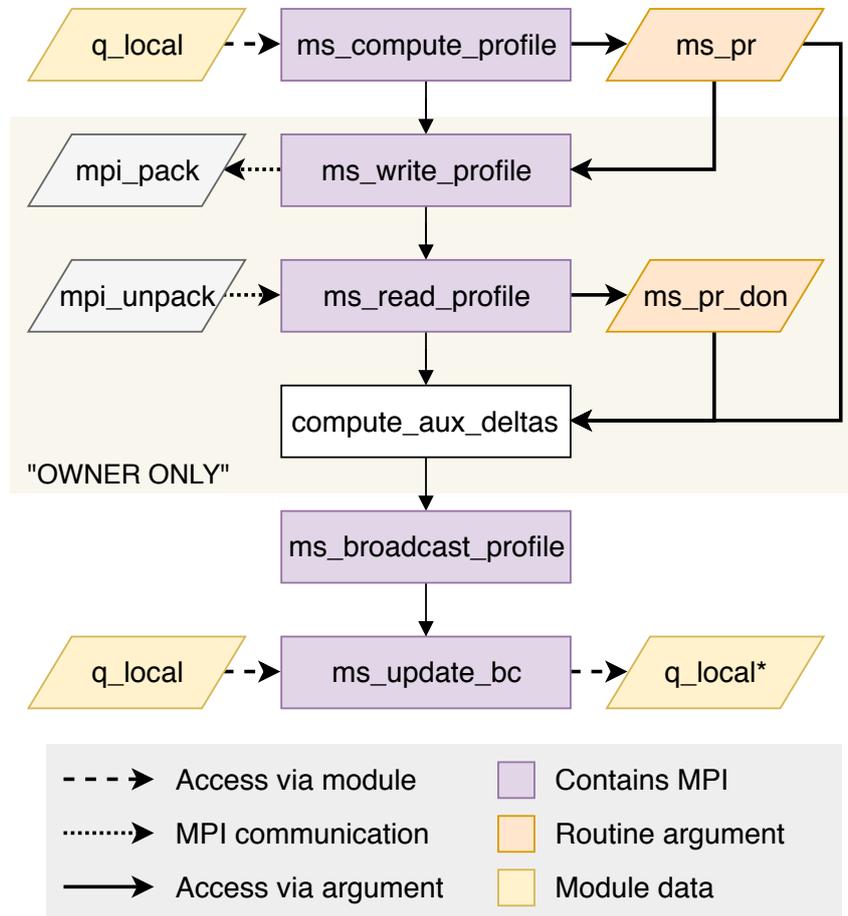


Figure 5.2: Schematic of the original direct multi-row exchange algorithm `ms_exchange`.

putation of the mixing-plane profiles, `ms_write_profile` and `ms_read_profile` performs the exchange of profiles between adjacent rows (and their interpolation upon receive), `compute_aux_deltas` computes the difference between the fluxes of the two (local and received) profiles, `ms_broadcast_profile` broadcasts the profiles and differences to all interested profiles, which then perform the boundary conditions update in routine `ms_update_bc`. In order to maintain consistency with the original code, the rewriting and hand-assembly of the differentiated routines were performed while trying to keep the dataflow as close to the original as possible.

The remaining of this subsection will present the description of how the routines of the original mixing-plane interface were rewritten and assembled into the rewritten main

mixing-plane routine `adj_ms_exchange`, represented in figure 5.4. The description is divided into four steps of the rewriting process, namely 1) memory allocation management, where a description of how the custom data structures of the original code were handled/rewritten to allow for differentiation; 2) profile computation; 3) profile exchange and interpolation and 4) boundary conditions update. These last three steps describe how the routines of each of those steps of the mixing-plane algorithm were dealt with, to allow for the correct differentiation using the AD tool.

### Memory Allocation Management

Starting from the top of figure 5.2, the computational mesh and state solution data at the mixing-plane interface is loaded from its original data structure into a set of arrays containing the information required by the routines. This is like the approach taken in the development of the adjoint solver but, in this case, the stencil is the inlet/exit face located at the mixing-plane interface. Routine `adj_ms_preprocess` deals with all the copying of the information into the "adjoint" structures. The data structure containing the profile information was also modified, as illustrated in figure 5.3. The original profile structures contained a series of allocatable arrays. Instead, the modified structure consists of an array of structures with fixed values arrays. As an example, with this modification `ms_pr%val1(k)` becomes `prAdj(k)%val1`. Special attention was also given to dynamic

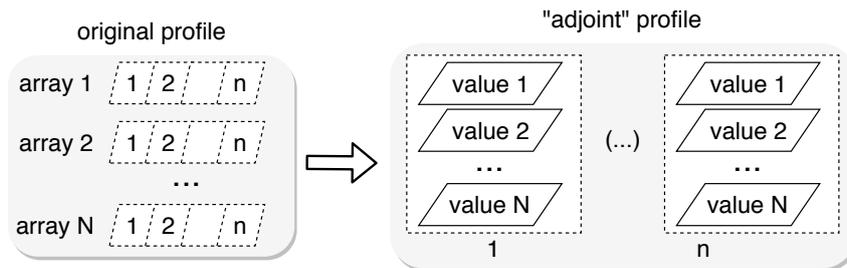


Figure 5.3: Modification of profile data structure to apply automatic differentiation.

memory allocation inside the routines to be differentiated, as in some cases they led to an incorrect differentiation.

### Profile Computation

This routine, responsible for the computation of the profile, was split into three routines, `adj_ms_comp_prof_1`, `adj_ms_accumulate` and `adj_ms_comp_prof_2`. This split-

ting came as a necessity to remove the MPI communication due to the accumulation at the owner to finish the normalization of the profiles. The local spanwise averaging is performed in `adj_ms_comp_prof_1`, the MPI gathering of the local averages into the owner of the profile is performed by `adj_ms_accumulate` and the final normalization is performed by `adj_ms_comp_prof_2`. Only `adj_ms_comp_prof_1` and `adj_ms_comp_prof_2` were differentiated using AD.

## Profile Exchange and Interpolation

The routines responsible for the exchange of profiles between owners of adjacent profiles `ms_write_profile` and `ms_read_profile` were rewritten to use the modified data structures instead of the original ones, resulting in the routines `adj_ms_write_profile` and `adj_ms_read_profile`. One major change was the removal of the interpolation routine, originally called inside the `ms_read_profile` into an independent routine, `adj_ms_interp`, to allow its differentiation. The routine `compute_aux_deltas` dealt with the computation of the difference between the fluxes of the local and adjacent row, as well as all the mixing-plane computations described in subsection 2.3.2. This routine was also modified to accommodate the rewritten data structures, as well as certain details to achieve the correct differentiation using the AD tool. This rewritten routine `adj_ms_aux_deltas` was merged with `adj_ms_interp`, as indicated in figure 5.4, as a way to simplify the hand assembly of the differentiated routines.

## Boundary Conditions Update

The last step of the mixing-plane algorithm is the update of the boundary conditions using the local information and the profile obtained from both local and received profiles. The routine responsible for this, `ms_update_bc`, also contains MPI communications. They occur due to a second accumulation for the computation of the non-reflectivity Giles boundary conditions. As stated before, all MPI operations were to be removed from routines that were to be subjected to the automatic differentiation procedure. In this case, the solution found was to accumulate the full information of the whole mixing-plane face and use it as input for the routine. In terms of Fortran data structures, the routines responsible for the non-reflectivity originally used user-defined type structures implemented in a similar way to the previously described profiles data structures (with dynamically allocated arrays/vectors).

New structures were created following the same approach described above for the profiles, to allow automatic differentiation.

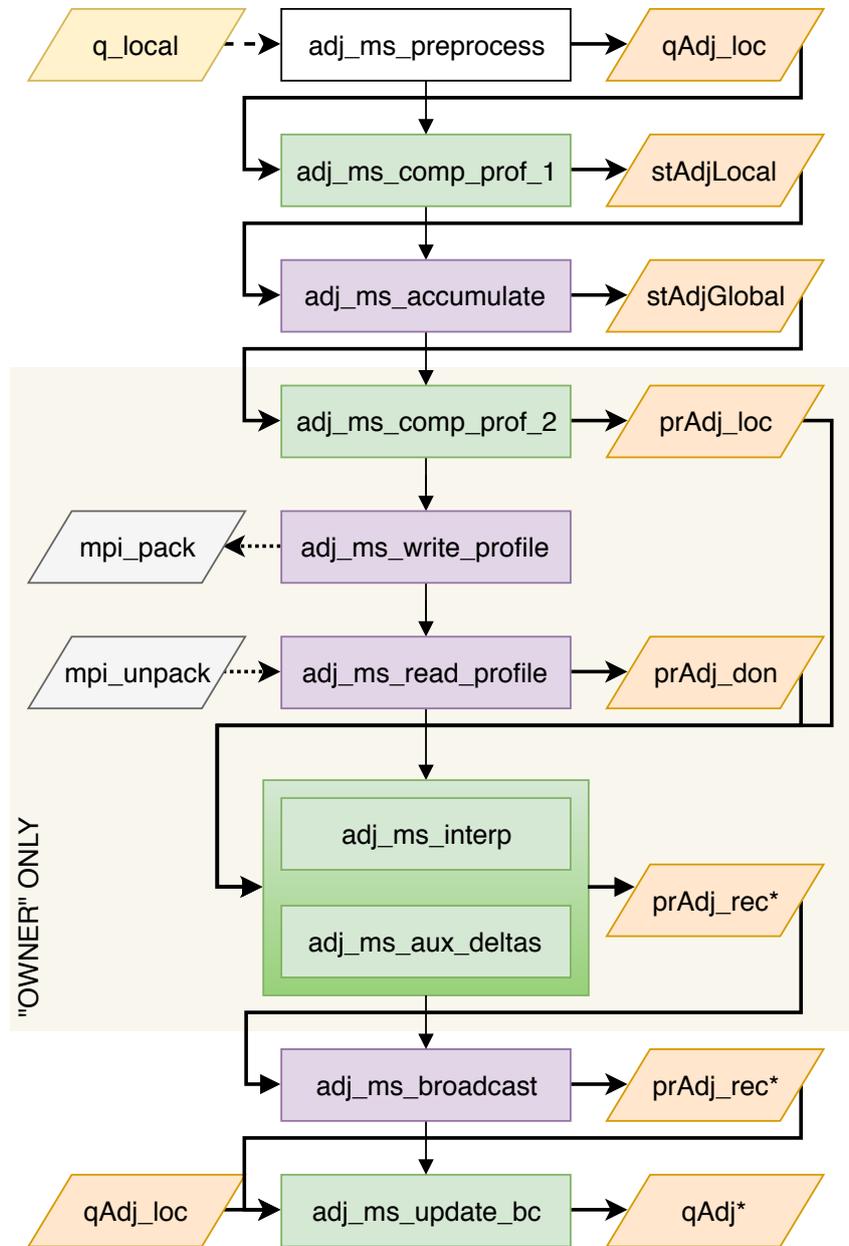


Figure 5.4: Schematic of the rewritten direct multi-row exchange algorithm.

## 5.2.2 Automatic Differentiation of Rewritten Routines

Having dealt with all the modifications necessary for the AD tool to differentiate the routines, the tool Tapenade was then used. The routines were differentiated in the reverse

mode, resulting in routines that, given a seed  $\mathbf{x}_b$  and inputs  $\mathbf{x}$  and  $\mathbf{y}$  would produce

$$\mathbf{x}_b = \left( \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \right)^T \mathbf{y}_b. \quad (5.15)$$

The selection of the reverse mode, despite the fact that the mixing-plane algorithm has a number of inputs and outputs in the same order of magnitude (the number of cells in the adjacent faces), was to later assemble the mixing-plane with the differentiated residual routines as stated in equation (5.5).

### 5.2.3 Hand-assembly of Differentiated Routines

As described in the previous subsection, and in section 3.3 where the AD approach for sensitivity analysis was presented, the routines generated by the AD tool will receive as input a seed that will define the weights of the sum of the derivatives output by the differentiated routine. With this, the multiplication of the various terms of the chain rule of equations (5.7) and (5.9) is done by using the derivatives produced by the previous routine as the seed to the next routine, as represented in figure 5.5.

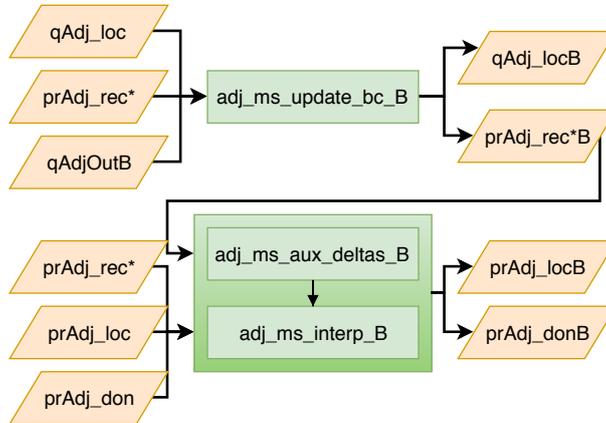


Figure 5.5: Partial schematic of the manually assembled differentiated mixing-plane routine `adj_ms_exchange_B`.

This step required special attention regarding the data flow of the derivatives. As seen in figure 5.6, the local face of a process with interest in the mixing-plane data exchange will send its averaged quantities to the owner of the profile, which in turn, after completing the creation of the profile, will proceed to exchange profiles with the owner of the adjacent face. After computing the deltas of the fluxes, the owner will broadcast these differences profile to all interested processes. We therefore have a dependency of each process with

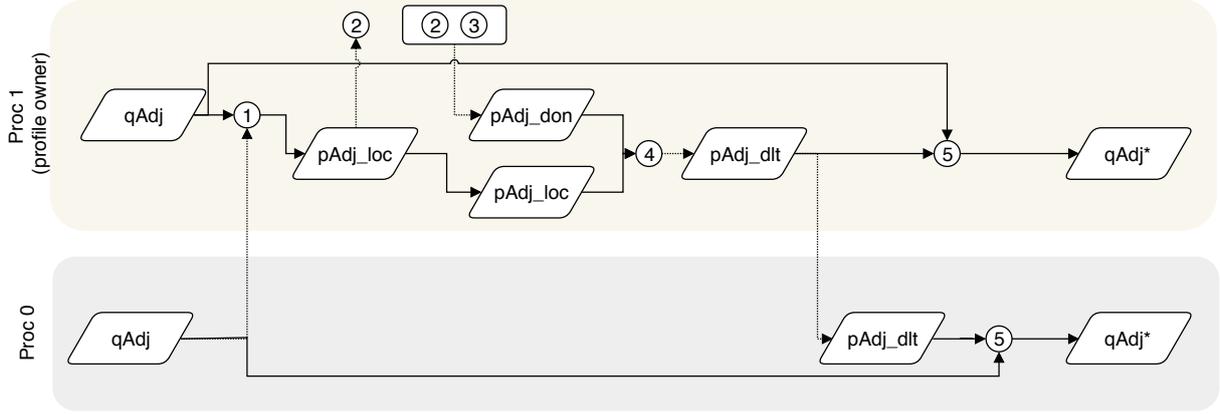


Figure 5.6: Simplified dataflow representation of the rewritten direct mixing-plane algorithm.

interest in the mixing-plane exchange on every process with interest in the mixing-plane exchange in the neighbor domain. The approach taken for this is described in algorithm 3. Once again, assuming just one direction of the exchange, where one row is the receiver and other the donor, each process computes its own derivatives relative to steps 5, 4 and 3 and then sends the derivatives  $\partial \mathbf{q}_{\text{local}}^* / \partial \mathbf{p}_{\text{don}}$  back to the owner of the profile in the other row (the donor row). The owner then computes the derivatives relative to the third part of step 1 (normalization), broadcasts the derivatives to all interested processes in the same row, which then compute the derivatives relative do the first part of step 1,  $\partial \mathbf{p}_{\text{don}} / \partial \mathbf{q}_{\text{don}}$ .

This communication approach is not optimal, but it was found to be the only consistent with the direct solver. It creates a bottleneck of information, as the owners of the profiles must receive the derivatives from each process. In the direct solver, this is not a problem since it is only performed once (one accumulation, one exchange and one broadcast) per exchange. The dataflow of the derivatives is much more complex as each process depends on all the other processes. This also imposes a synchronization between the processes interested in the mixing-plane exchange. If two adjacent faces belonging to a mixing-plane interface have levels of discretization, i.e. different number of cells, the processes of one of the rows will have to run the differentiated exchange routine just so they can compute their terms of the derivative to send to the other row.

## 5.2.4 Integration with Single-stage Routines

With the assembled differentiated routine producing  $\partial \mathbf{q}_{\text{local}}^* / \partial \mathbf{q}_{\text{don}}$  and  $\partial \mathbf{q}_{\text{local}}^* / \partial \mathbf{q}_{\text{local}}$ , the next step is to integrate the coupling term with the existent single-row differentiated

---

**Algorithm 3:** Adjoint mixing-plane exchange algorithm.

---

```

1 call adj_ms_compute_profile_1;
2 call adj_ms_accumulate;
3 if (ms_owner) then
4   call adj_ms_compute_profile_2;
5   call adj_ms_write_profile;
6   call adj_ms_read_profile;
7   call adj_ms_interp_and_deltas;
8 call adj_ms_broadcast;
9 call adj_ms_update_bc_B;
10 call adj_ms_interp_and_deltas_B;
11 call adj_ms_write_profile_B;
12 for ip ← to nproc_nbr do
13   if (ms_owner) then
14     call read_profile_B;
15     call adj_ms_compute_profile_2_B;
16   call adj_ms_accumulate_B;
17   call adj_ms_compute_profile_1_B;
18   save derivatives locally;
19 send derivatives back to interested processes;

```

---

residual routine to insert the terms into  $\partial \mathbf{R} / \partial \mathbf{q}$ .

The partial derivatives  $\partial \mathbf{R}_i / \partial \mathbf{q}$  computed with the single-row differentiated routine `adjoint_residual_B` are first converted into the adjoint multi-row stencil, which is then used as seed for the hand-assembled differentiated mixing-plane routine `adj_ms_exchange_B`. This only happens in the case the stencil of influence of the residual of that cell contains at least one cell in the mixing-plane interface (see figure 5.7). The derivatives  $(\partial \mathbf{R}_i / \partial \mathbf{q})_j$

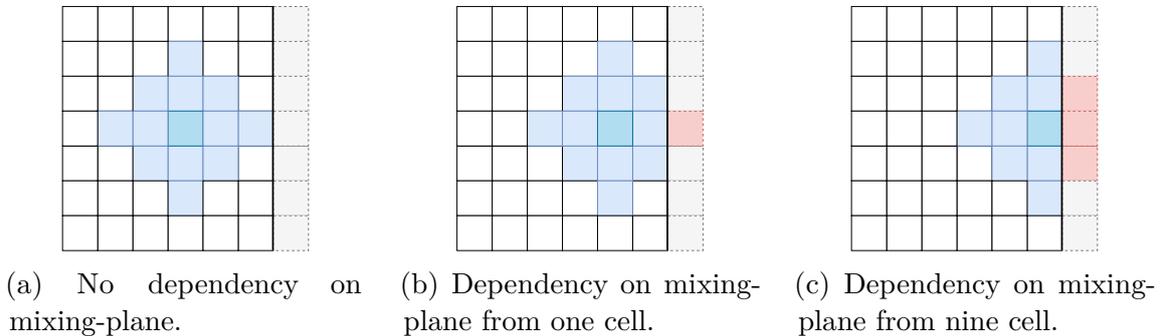


Figure 5.7: Stencil of influence of residual of a cell close to a mixing-plane boundary. (blue: local residual stencil; red: local residual stencil on mixing-plane surface)

indicated in figure 5.8 as blue boxes (where each "box" corresponds to the derivatives relative to each process on the adjacent row with interest in the mixing-plane interface)

are then inserted into the global matrix  $\partial \mathbf{R} / \partial \mathbf{q}$  in section 5.1.

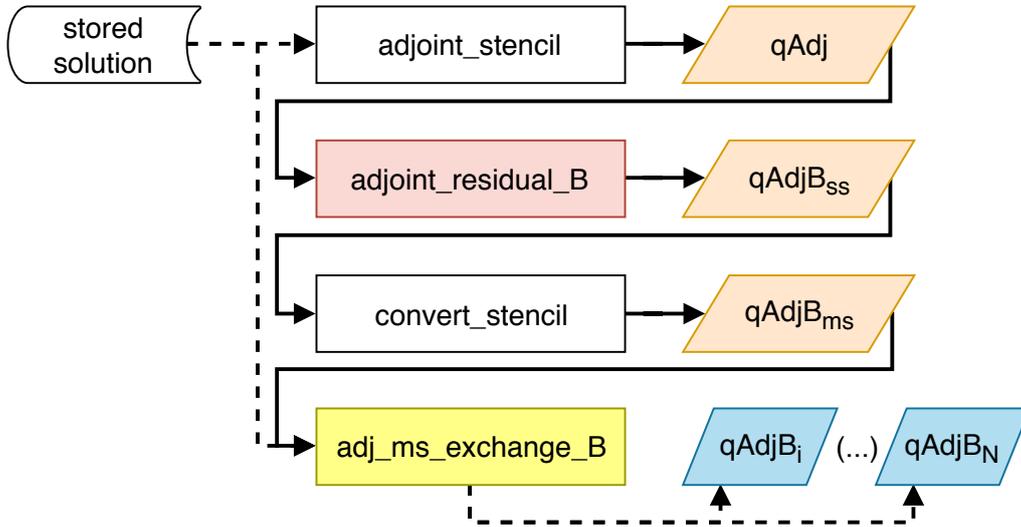


Figure 5.8: Schematic representation of the assemble of the differentiated mixing-plane routine with the single-row differentiated routine.

### 5.3 Overview of the Implementation Effort

The implementation of the adjoint mixing-plane described in the previous section resulted in a considerable increment to the source code of the adjoint solver.

A comparison of these metrics with the legacy and enhanced adjoint solver is depicted in figure 5.9, where these two metrics are plotted. The implementation of the adjoint mixing-plane increase the code size of the the adjoint solver with 420 new routines and about 55,000 new lines of code, from which 120 are the rewritten and differentiated routines and 300 vary between the routines directly involved in the hand assembly differentiated routines, such as communication routines and data structures handling and routines used for debugging of the code during its development.

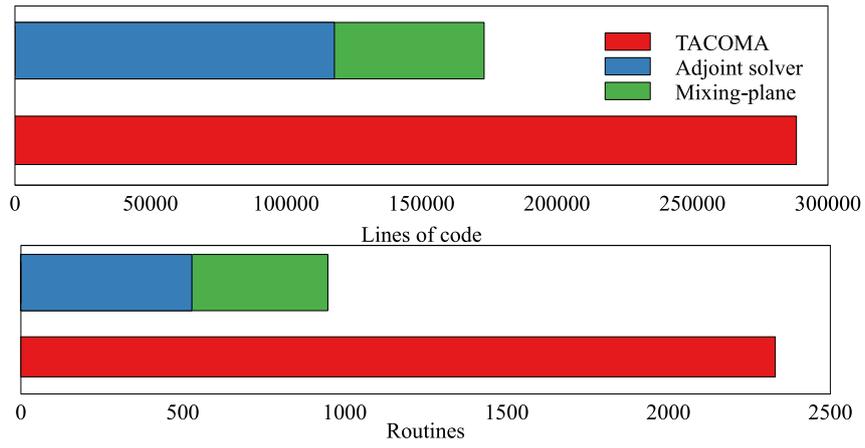


Figure 5.9: Comparison of code metrics of the direct solver, adjoint solver before mixing-plane and adjoint solver after implementation of mixing-plane.



# Chapter 6

## Verification of the Implementation of the Adjoint Mixing-plane Interface

The verification of the implementation of the adjoint mixing-plane interface is here presented following the same three steps of its development: rewriting, differentiation and hand-assembling, integration.

In the first step, the profiles computed with the original and rewritten routines are compared. Then, the verification of the differentiation and hand-assembling of the mixing-plane exchange routines is here presented as a comparison of  $\partial \mathbf{q}_{\text{local}}^* / \partial \mathbf{q}_{\text{don}}$  computed both with the differentiated routine and with finite-differences approximation. The last step, the integration of the adjoint mixing-plane into the existing adjoint solver, is also verified against finite-difference approximations.

### 6.1 Description of the Study Case

The verification is performed with the analysis of a stator/rotor stage of a low pressure turbine of a commercial jet engine. The low pressure turbine stator/rotor stage is modeled by two domains coupled with the mixing-plane interface. Each of the domains is discretized with an O-H grid with a total of 90,750 cells amongst the two domains. The full stage and computational mesh are represented in figure 6.1. The division of the computational domains into various blocks is presented in figure 6.2. The first row has a total of 15 blocks and the second is divided into 14 blocks.

The inlet boundary conditions prescribed are absolute tangential velocity and pressure

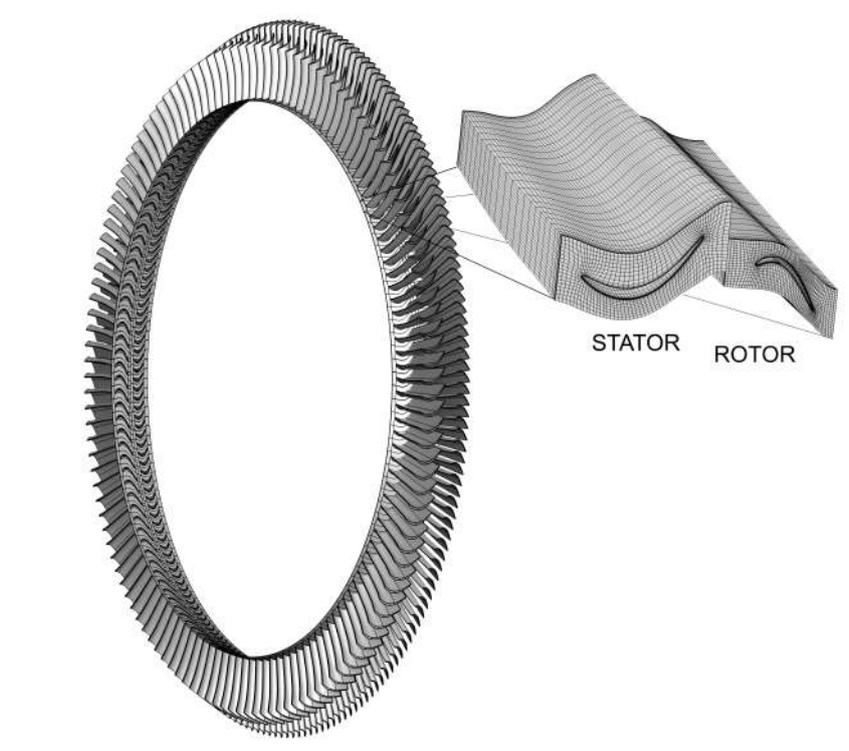


Figure 6.1: Representation of stator-rotor stage in study and respective computational mesh.

extrapolated from the interior. The exit static pressure is held fixed [56]. All solid walls are considered impermeable with no-slip condition. Wall-functions [117] are used to model near-wall viscous boundary layer. The remaining faces are either block-to-block interfaces or periodic. Between the two domains, the boundary conditions are updated with the mixing-plane algorithm with exchange of boundary fluxes. The inlet, outlet and mixing-plane surfaces are represented in figure 6.3.

The flow solution was converged to a relative averaged residual of the continuity

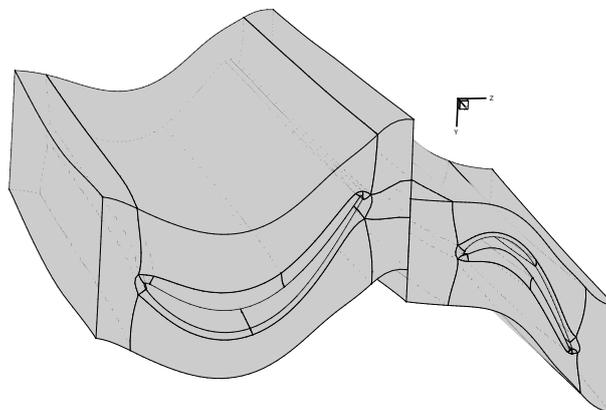


Figure 6.2: Division of the computational domains into various individual blocks.

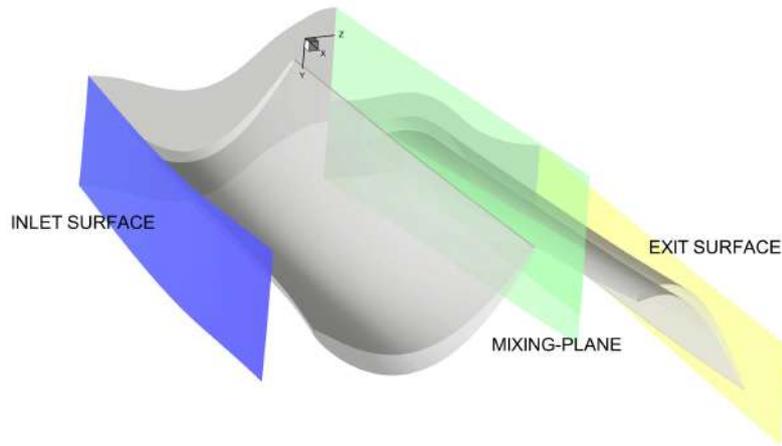


Figure 6.3: Schematic representation of the inlet and exit boundary surfaces of the stator-rotor stage domain.)

equation of  $10^{-6}$  or less, as shown in figure 6.4. For the adjoint solutions, the convergence

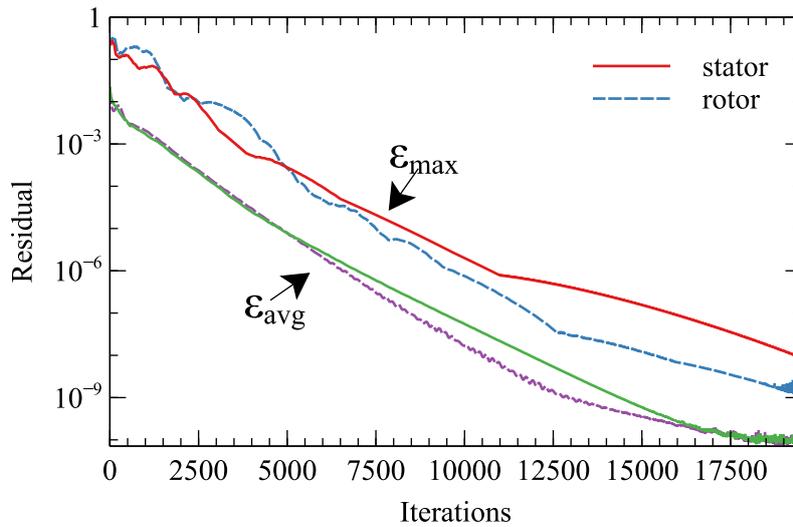


Figure 6.4: Residual iteration history of flow solution convergence of stator/rotor stage simulation.

criterion was a relative difference in the magnitude of the residual between iterations of  $10^{-9}$ . The history of the residual during the GMRES iterations is presented in figure 6.5. The restart of the GMRES was set at 75 iterations, which is noticeable in the residual history from the sharp bend on the residual curve for all metrics, with a slight increase of the residual.

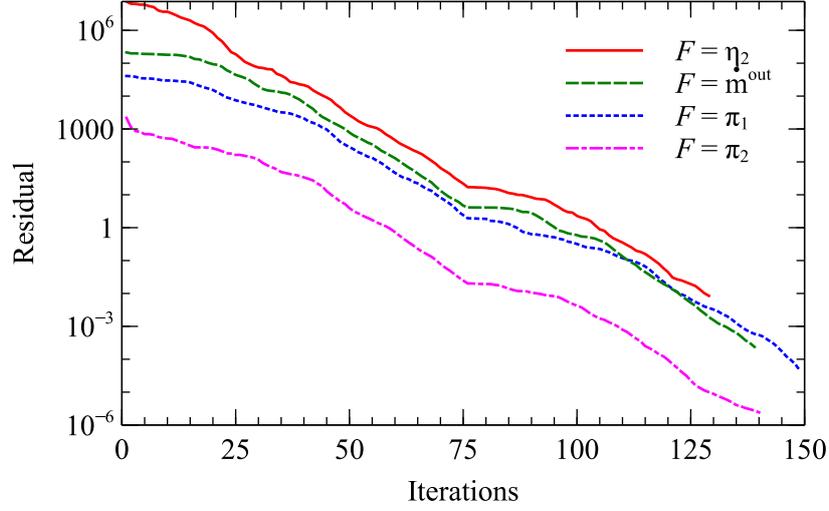


Figure 6.5: Residual iteration history of GMRES for adjoint solutions.

## 6.2 Overview of the Metrics of Performance

The performance metrics used as objective functions in the present verification are exit mass flow ( $\dot{m}^{\text{out}}$ ), rotor isentropic efficiency ( $\eta_{\text{rotor}}$ ), stage isentropic efficiency ( $\eta_{\text{stage}}$ ), stator pressure loss ( $\pi_{\text{stator}}$ ), rotor total pressure ratio ( $\pi_{\text{rotor}}$ ), stage total pressure ratio ( $\pi_{\text{stage}}$ ) and rotor blade area averaged total temperature  $(T_{Ta})_{\text{rotor}}$ . Mass flow considered is the mass flow going out of the stage, computed at the exit of the rotor, and is given by the sum of the fluxes across the exit of the stage as

$$\dot{m}^{\text{out}} = N_B \sum^{\text{exit}} \mathbf{f}_i, \quad (6.1)$$

where  $\mathbf{f}_i$  is the momentum flux across the exit of the stage and  $N_B$  the number of blades of the rotor row. Recalling the expressions for pressure ratio and efficiency introduced in section 2.1, rotor isentropic efficiency is computed as

$$\eta_{\text{rotor}} = \frac{(T_{Ta}^{\text{exit}} / T_{Ta}^{\text{mix}}) - 1}{(p_{Ta}^{\text{exit}} / p_{Ta}^{\text{mix}})^{(\gamma-1)/\gamma} - 1}, \quad (6.2)$$

and stage isentropic efficiency as

$$\eta_{\text{stage}} = \frac{(T_{Ta}^{\text{exit}} / T_{Ta}^{\text{inlet}}) - 1}{(p_{Ta}^{\text{exit}} / p_{Ta}^{\text{inlet}})^{(\gamma-1)/\gamma} - 1}, \quad (6.3)$$

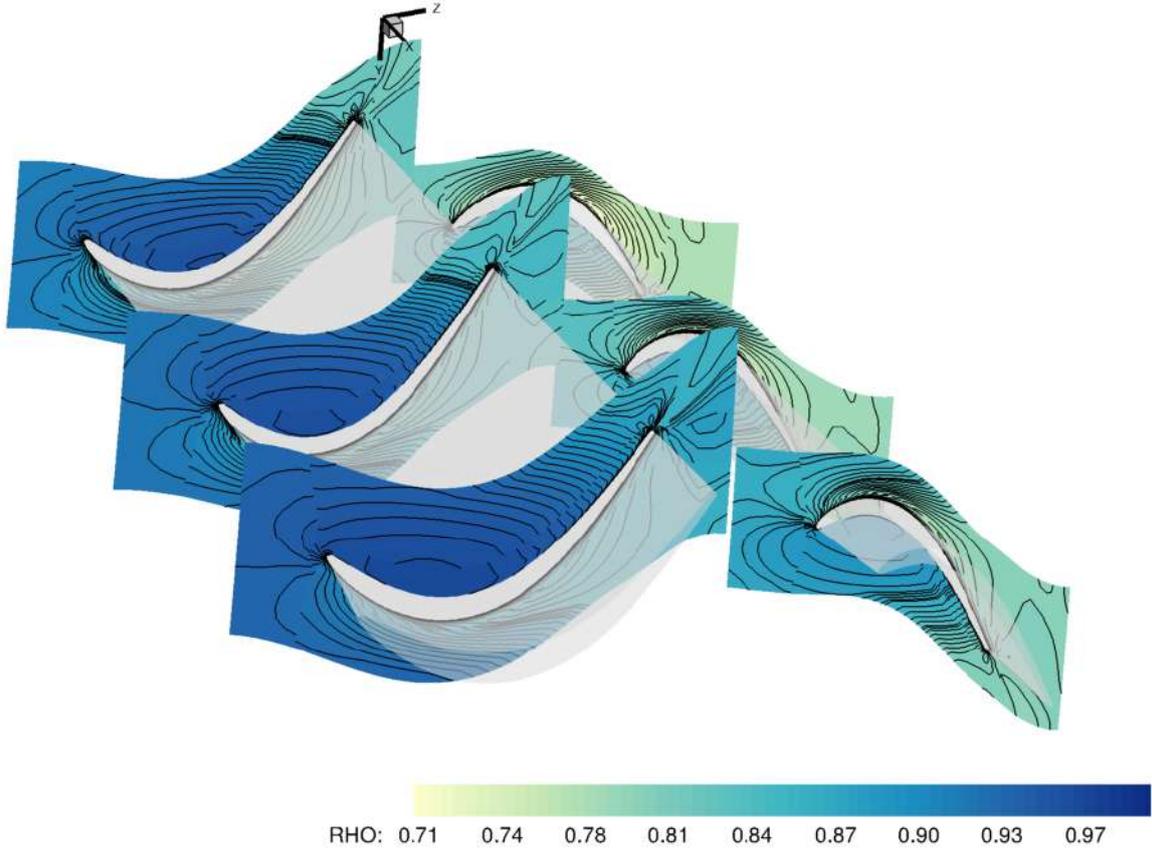


Figure 6.6: Normalized density field of converged direct solution of stator/rotor turbine stage.

where the subscript  $Ta$  indicates enthalpy averaged total quantities for the case of pressure, and mass averaged total quantities for the case of temperature. The superscripts *inlet* and *exit* indicate a quantity at the inlet and exit of the stage, and the superscript *mix* indicates a quantity at the inlet/exit belonging to the mixing-plane interface, as represented by figure 6.3.

The total pressure loss of the stator is given as

$$\pi_{\text{stator}} = \frac{p_{Ta}^{\text{inlet}} - p_{Ta}^{\text{mix}}}{p_{Ta}^{\text{inlet}}} \times 100, \quad (6.4)$$

while the pressure ratios of the rotor and of the whole stage are computed as

$$\pi_{\text{rotor}} = \frac{p_{Ta}^{\text{exit}}}{p_{Ta}^{\text{mix}}}; \quad \pi_{\text{stage}} = \frac{p_{Ta}^{\text{exit}}}{p_{Ta}^{\text{inlet}}}. \quad (6.4 \text{ a,b})$$

The area averaged total temperature at the rotor blade is computed as

$$(T_{Ta})_{\text{rotor}} = \frac{\sum_{i=1}^{(N_c)_{\text{rotor}}} a_i (T_T)_i}{\sum_{i=1}^{(N_c)_{\text{rotor}}} a_i}, \quad (6.5)$$

where  $(N_c)_{\text{rotor}}$  is the total number of cells in the rotor blade surface,  $a_i$  is the computational cell face area and  $(T_T)_i$  is the temperature at the wall of the individual cell  $i$ .

## 6.3 Overview of the Independent Variables

The adjoint-based sensitivity of the metrics of performance described in the previous section can be computed to an arbitrary quantity and variety of independent variables (or design variables) from equation (3.19), as long as the terms  $\partial\mathcal{I}/\partial\boldsymbol{\alpha}$  and  $\partial\mathbf{R}/\partial\boldsymbol{\alpha}$  can be computed, either directly or by using the chain rule of derivatives. The following subsections briefly describe the independent variables assumed for the remaining of this document.

### 6.3.1 Inlet and Exit Boundary Conditions

The first set of independent variables considered are the inlet and exit boundary conditions, imposed at the inlet and exit of the complete domain in analysis. For multi-rows they are the boundary conditions imposed at the inlet of the first row and at the exit of the last row, that is, at the inlet of the stator and at the exit of the rotor, for the present case in study (see figure 6.3). Recalling the boundary conditions defined in Chapter 2, the complete vector of inlet/exit boundary conditions information is given by

$$\mathbf{U} = \left\{ p_T^{\text{inlet}}, h_T^{\text{inlet}}, V_t^{\text{inlet}}, C_r^{\text{inlet}}, C_z^{\text{inlet}}, p^{\text{exit}} \right\}. \quad (6.6)$$

The adjoint-based sensitivity information of aerodynamic metrics of performance can be used by the designer in a variety of ways, such as 1) directly used to manually tweak the flow [56], 2) incorporated in an automatic gradient-based optimization design framework [118], or 3) used for uncertainty quantification in robust design [119].

### 6.3.2 Blade Shape and Hub Geometry

The second set of independent variables considered are the blade shape and hub/casing geometry, directly defined by the computational mesh grid  $\mathbf{X}$ . While being the foundation for obtaining *higher-end* sensitivities of geometry parameters, the sensitivity of the performance metric to mesh grid  $d\mathcal{I}/d\mathbf{X}$  does not provide information that can be easily analyzed as is. As such, in the present document, the sensitivities of the various metrics to mesh grid are processed with some transformations of variables to allow a better analysis of the the adjoint-based results.

The first transformation is the projection of the sensitivities onto the outer normal of the surface of the blade, evaluated as

$$\frac{d\mathcal{I}}{dn} = \frac{d\mathcal{I}}{dx} \frac{dx}{dn} + \frac{d\mathcal{I}}{dy} \frac{dy}{dn} + \frac{d\mathcal{I}}{dz} \frac{dz}{dn} = \frac{d\mathcal{I}}{dx} n_x + \frac{d\mathcal{I}}{dy} n_y + \frac{d\mathcal{I}}{dz} n_z, \quad (6.7)$$

where the surface outer normal unit vector is given by  $\mathbf{n} = (n_x, n_y, n_z)$ . This allows to better visually assess this geometric sensitivity. The second transformation is performed to assess the influence of hub and casing geometries on the performance metric and it is defined as

$$\frac{d\mathcal{I}}{dr} = \frac{d\mathcal{I}}{dx} \frac{dx}{dr} + \frac{d\mathcal{I}}{dy} \frac{dy}{dr} = \frac{d\mathcal{I}}{dx} \cos(\theta) + \frac{d\mathcal{I}}{dy} \sin(\theta), \quad (6.8)$$

where  $\theta$  is the tangential angle in cylindrical coordinates measured from the x to the y axis. This sensitivity information can be extremely important in hub and/or casing shape optimization processes, often referred as *endwall contouring*, which can lead to significant performance improvement of the turbomachine by significantly impacting the secondary flows [53, 120–122].

A designer (or an optimizer) would not be interested in perturbing the mesh directly, but in changing a set of design parameters  $\beta$  that would represent the geometry/deformation by some method of parameterization, which would introduce some smoothing to the perturbation and thus smoothing unwanted oscillations. Examples of these engineering significant parameters  $\beta$  are blade stagger or blade angle. The sensitivity information of the performance metrics to the design parameters  $\beta$  would be obtained using the adjoint-based

sensitivity information to the mesh as

$$\frac{d\mathcal{I}}{d\boldsymbol{\beta}} = \frac{d\mathcal{I}}{d\mathbf{x}} \frac{d\mathbf{x}}{d\boldsymbol{\beta}} + \frac{d\mathcal{I}}{d\mathbf{y}} \frac{d\mathbf{y}}{d\boldsymbol{\beta}} + \frac{d\mathcal{I}}{d\mathbf{z}} \frac{d\mathbf{z}}{d\boldsymbol{\beta}}. \quad (6.9)$$

Depending on the tool (or set of tools) used to generate the flow grid mesh  $(\mathbf{x}, \mathbf{y}, \mathbf{z})$  from the geometry parameterization  $\boldsymbol{\beta}$ , the sensitivities  $d\mathbf{x}/d\boldsymbol{\beta}$ ,  $d\mathbf{y}/d\boldsymbol{\beta}$  and  $d\mathbf{z}/d\boldsymbol{\beta}$  can be obtained by either analytic methods (if source code is available) or by FD approximations (if a blackbox is used).

## 6.4 Verification of the Rewritten Mixing-plane Interface

The first step of the verification was performed after the routines responsible for the mixing-plane exchange were rewritten. Those followed the steps illustrated in figure 2.9. Before introducing them into the AD tools, we needed to be certain that they were producing boundary updates consistent with the original routines.

### 6.4.1 Profile Computation

The comparison of the profiles computed both with the original and the rewritten routines is presented in figure 6.7, where the relative difference of radial profiles of the first five averaged flux quantities  $\mathbf{F}_1$  to  $\mathbf{F}_5$  are plotted. It can be seen from the comparison that the rewritten profile computation routines produce profiles that are nearly identical, with maximum differences in the order of machine precision. A similar comparison is presented in figure 6.8, for the turbulence fluxes  $\mathbf{F}_6$  and  $\mathbf{F}_7$ . The relative differences for this case, albeit larger than the ones in the previous figure, are still very small, with maximum relative differences of 0.0005%, and thus allows the conclusion that the profile computed with the rewritten routines is numerically identical to the original profile. The profile structures of the code contain a larger set of variables than the variables here presented. The verification of those variables is not presented here, although a full verification of every variable was performed, obtaining similar agreement between the profiles produced by the original and rewritten routines.

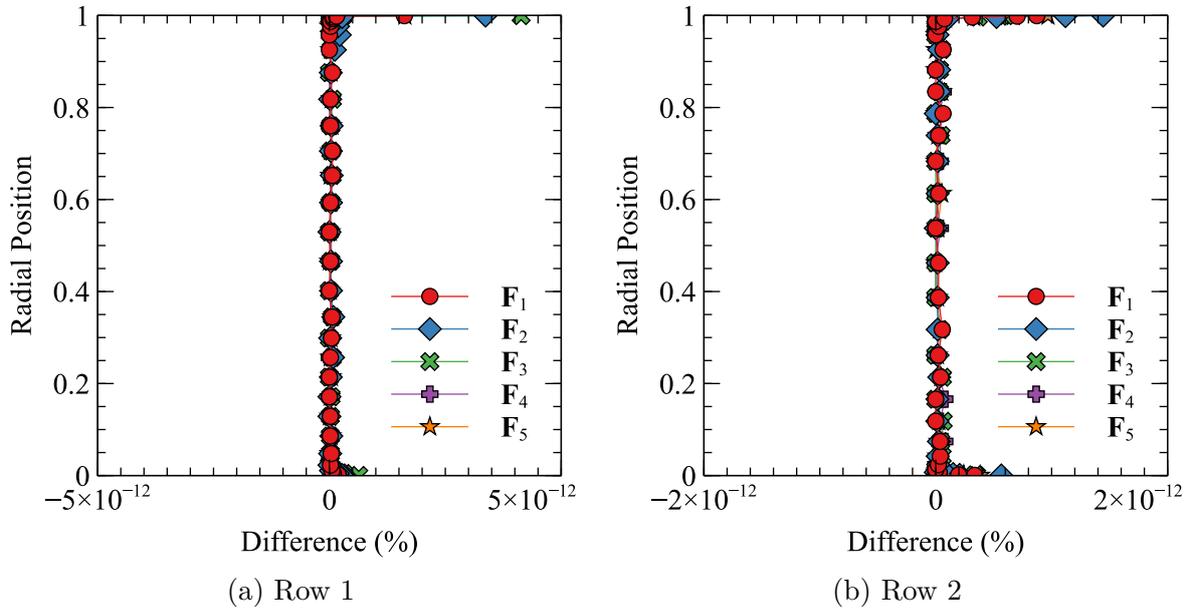


Figure 6.7: Relative difference between averaged fluxes  $F_1$  to  $F_5$  computed with original and rewritten routines.

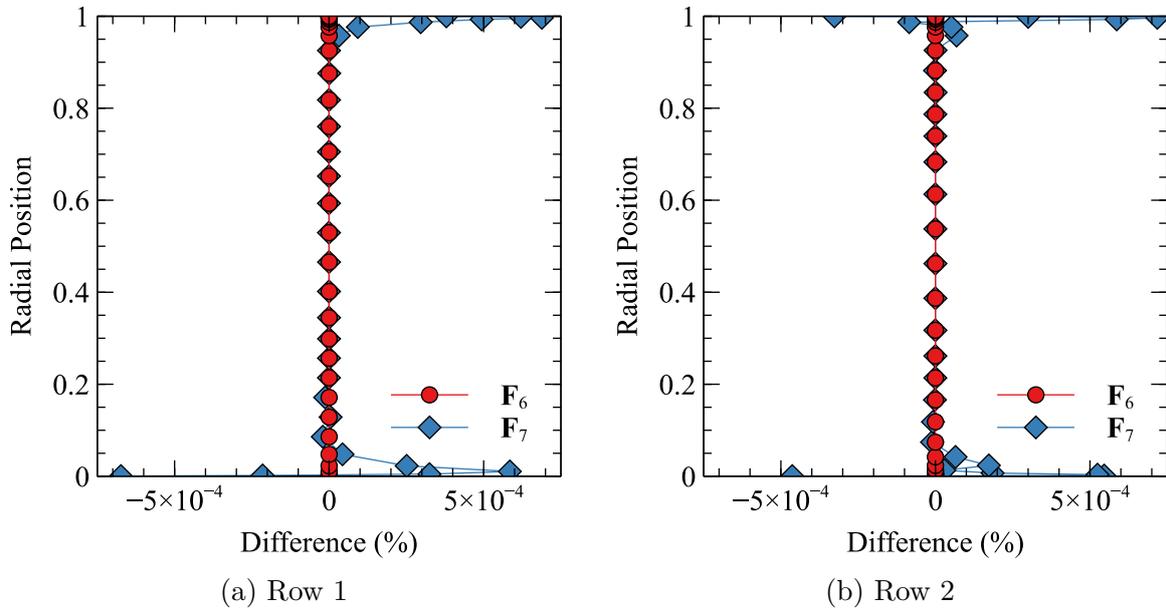


Figure 6.8: Relative difference between averaged fluxes  $F_6$  and  $F_7$  computed with original and rewritten routines.

## 6.4.2 Boundary Conditions Update

Following the verification of the rewritten routines responsible for the computation of the profiles, the results of the verification of the rewritten boundary update routines is now presented in figure 6.9, where the relative difference between the boundary conditions updated with the original and the rewritten routines are shown. Each column represents the value of the maximum difference between the updated auxiliary cell value computed with the original and rewritten routines. A maximum relative difference of about 0.1% is obtained for  $q_5$  in row 1, with all the other relative differences being below 0.001%. This larger difference, when compared to the other values is localized, as can be seen from

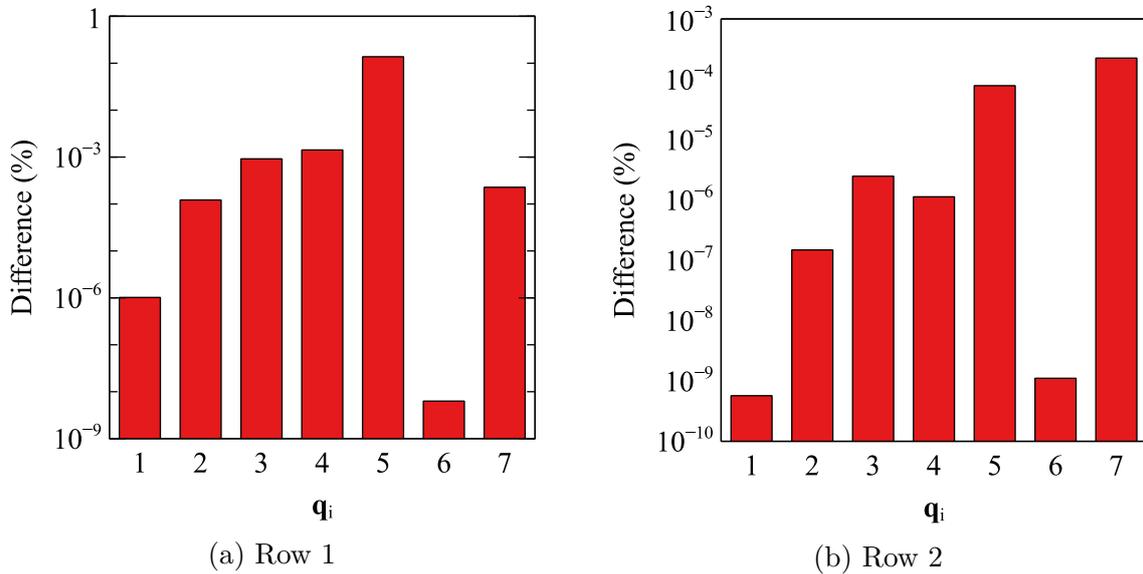


Figure 6.9: Maximum relative difference between updated auxiliary cells values computed with original and rewritten routines.

figure 6.10, where the relative difference between the values of  $q_5$  produced by the original and rewrite routines is presented at the exit face of row 1 and as will be observed from the following sections, will not significantly impact the correctness of the adjoint mixing-plane coupling terms and adjoint-based total derivatives.

## 6.5 Verification of the Adjoint Mixing-plane Interface

With assurance that the rewritten routines produce results with differences from the original routines that are within machine precision, the next step is the verification of the coupling terms  $\partial \mathbf{R}_{\text{rec}} / \partial \mathbf{q}_{\text{don}}$  and  $\partial \mathbf{R}_{\text{rec}} / \partial \mathbf{q}_{\text{local}}$ , used in equations (5.5) and (5.8).

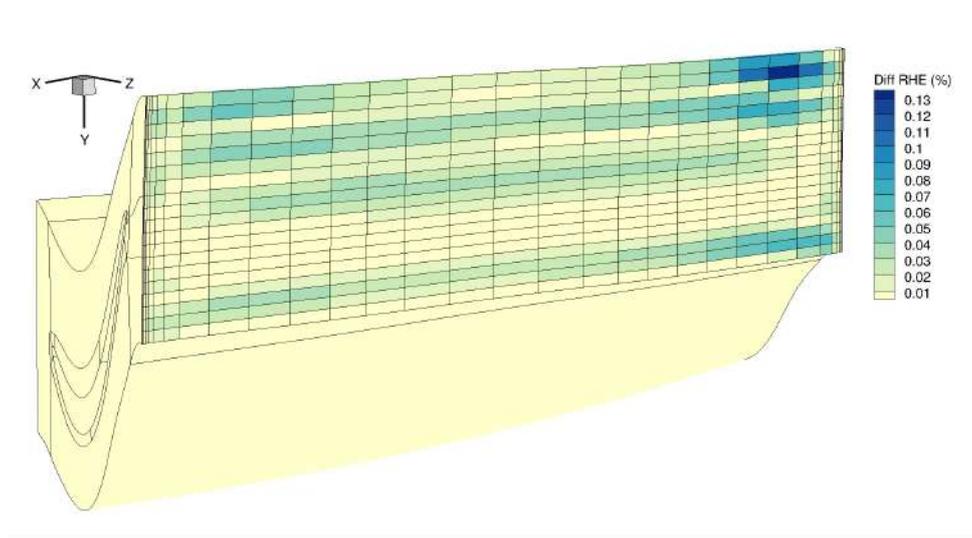


Figure 6.10: Relative difference between updated auxiliary cells values computed with original and rewritten routines in row 1.

### 6.5.1 Stencil of Dependency

The first step of this verification was to see if the stencil of influence of a selected cell was correct. This can be observed in figure 6.11, where the derivatives  $\partial \mathbf{R}_1 / \partial \mathbf{q}_1$  are presented for an arbitrary cell in the exit face of the stator computational domain. The dependency of the residual of cell (dark blue in the figure) on the whole row of cells in the same radial position is clearly visible. This is the expected stencil of influence, as the boundary conditions are updated from the difference of averaged fluxes, computed from the whole row of cells at a specific radial position. A dependency on the two cells in each azimuthal position is also visible, directly resultant from the local (single row) residual stencil. One thing that should be mentioned is the lack of dependency on the corner cells, as previously mentioned and represented in figure 5.7c, which is due to the dissipation near the boundaries becoming of second-order.

### 6.5.2 Local Sensitivity

To verify the intra-row sensitivities  $\partial \mathbf{R}_{\text{rec}} / \partial \mathbf{q}_{\text{local}}$ , one cell located at the stator face belonging to the mixing-plane and the sensitivity of the residual to the interior local cells was compared with second-order finite-difference approximations of the same sensitivity.

Figure 6.12 presents the comparison of the sensitivities of the the residual of the continuity equation  $\mathbf{R}_1$  of the cell highlighted in figure 6.11, on the face of the mixing-

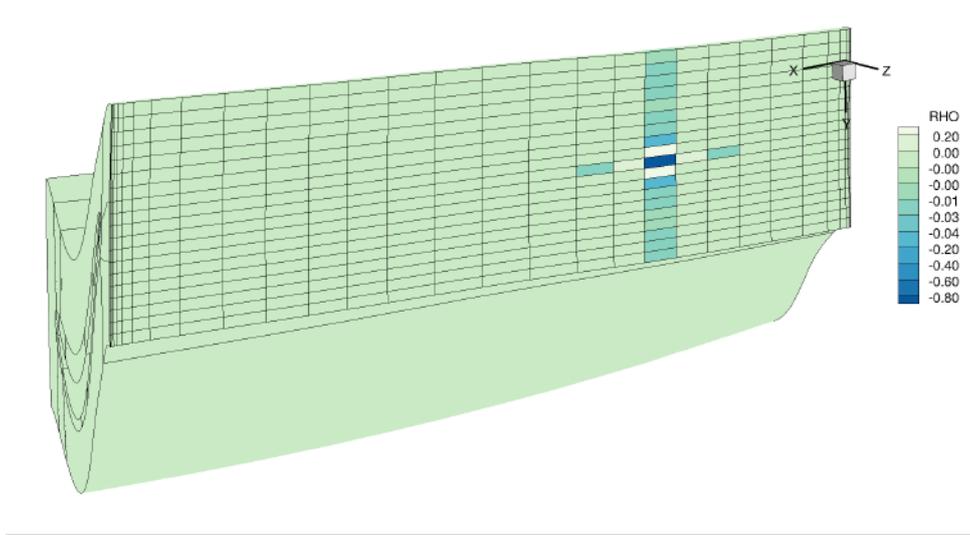


Figure 6.11: Sensitivity of  $\mathbf{R}_{\text{rec}}$  of a single cell on the other interior cells in the same face  $\partial\mathbf{R}_{\text{rec}}/\partial\mathbf{q}_{\text{local}}$  (normalized values).

plane of the first row, on the interior cells of the same face, located in the same radial position as the selected cell, computed with differentiated routines and second-order finite-differences (with a relative perturbation step of 0.01%). The values of the derivatives computed with both methods are very close and always with same sign, albeit the large relative differences that can be seen in the figure. This is a typical problem when dealing with relative differences, which, for small values close to zero, can translate into very large values.

An equivalent comparison for a cell located at the inlet face of the second row is presented in figure 6.13. In this case, the errors are very small, even for the smaller valued derivatives, showing excellent agreement with the finite-difference approximations.

Similar verification was performed for the residuals of the other equations, achieving good agreement with finite-difference approximation.

### 6.5.3 Inter-row Sensitivity

To verify the inter-row sensitivities  $\partial\mathbf{R}_{\text{rec}}/\partial\mathbf{q}_{\text{don}}$ , a set of cells in each face was selected and their sensitivity to the face of the other row compared to second-order finite-difference approximations was analyzed. Figure 6.14 presents the maximum relative difference of  $\partial(\mathbf{R}_i)_{\text{rec}}/\partial\mathbf{q}_{\text{don}}$  for 6 control points (or cells). The first three ( $CP_{1-3}$ ) are located at the exit of the stator, and the other three ( $CP_{4-6}$ ) located at the inlet of the rotor. Good agreement is visible with both methods, with maximum relative differences of less than 4%.

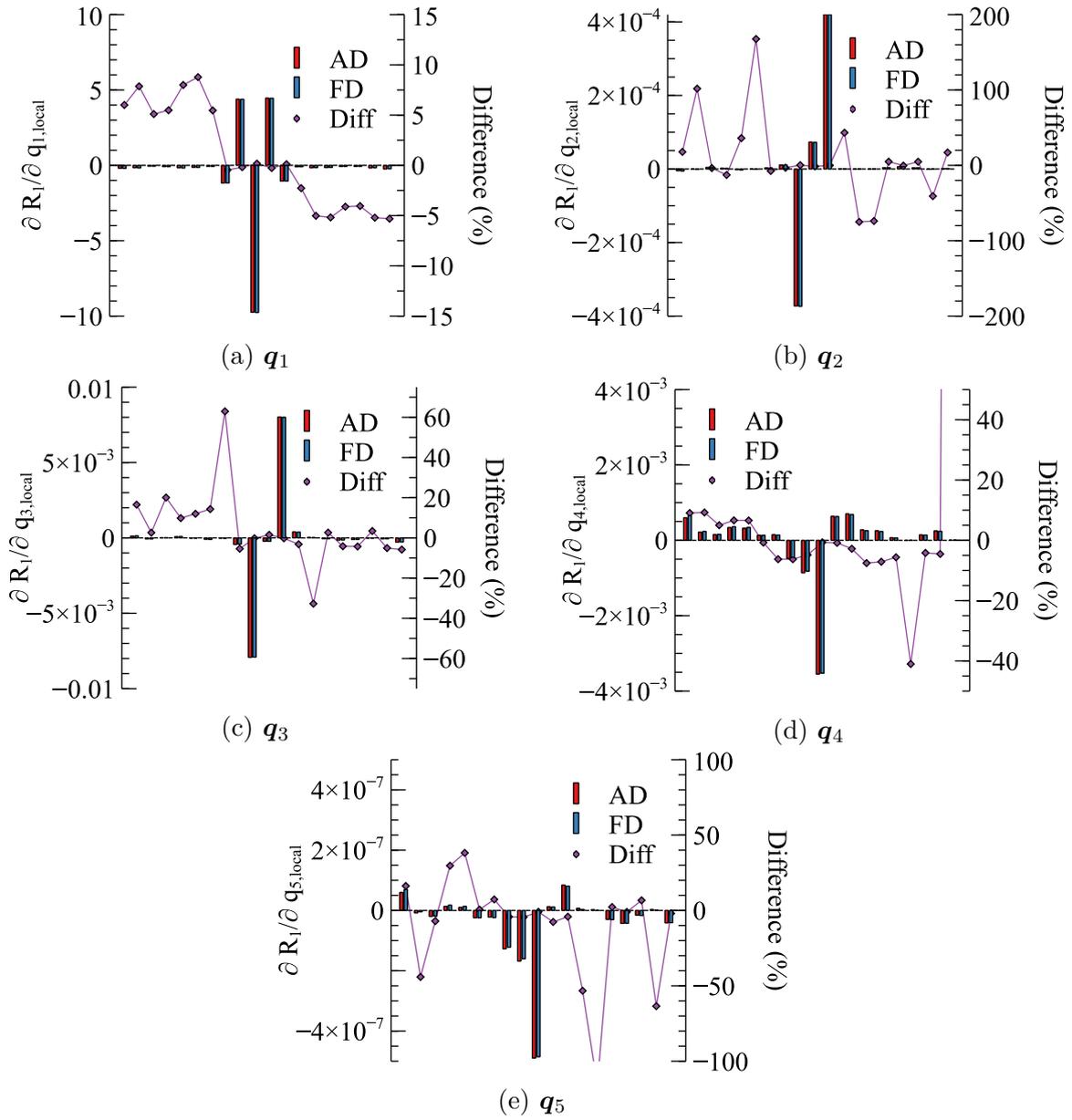


Figure 6.12: Comparison of mixing-plane local sensitivities of the residual of a cell at the exit of the stator computed with differentiated routines and FD approximations.

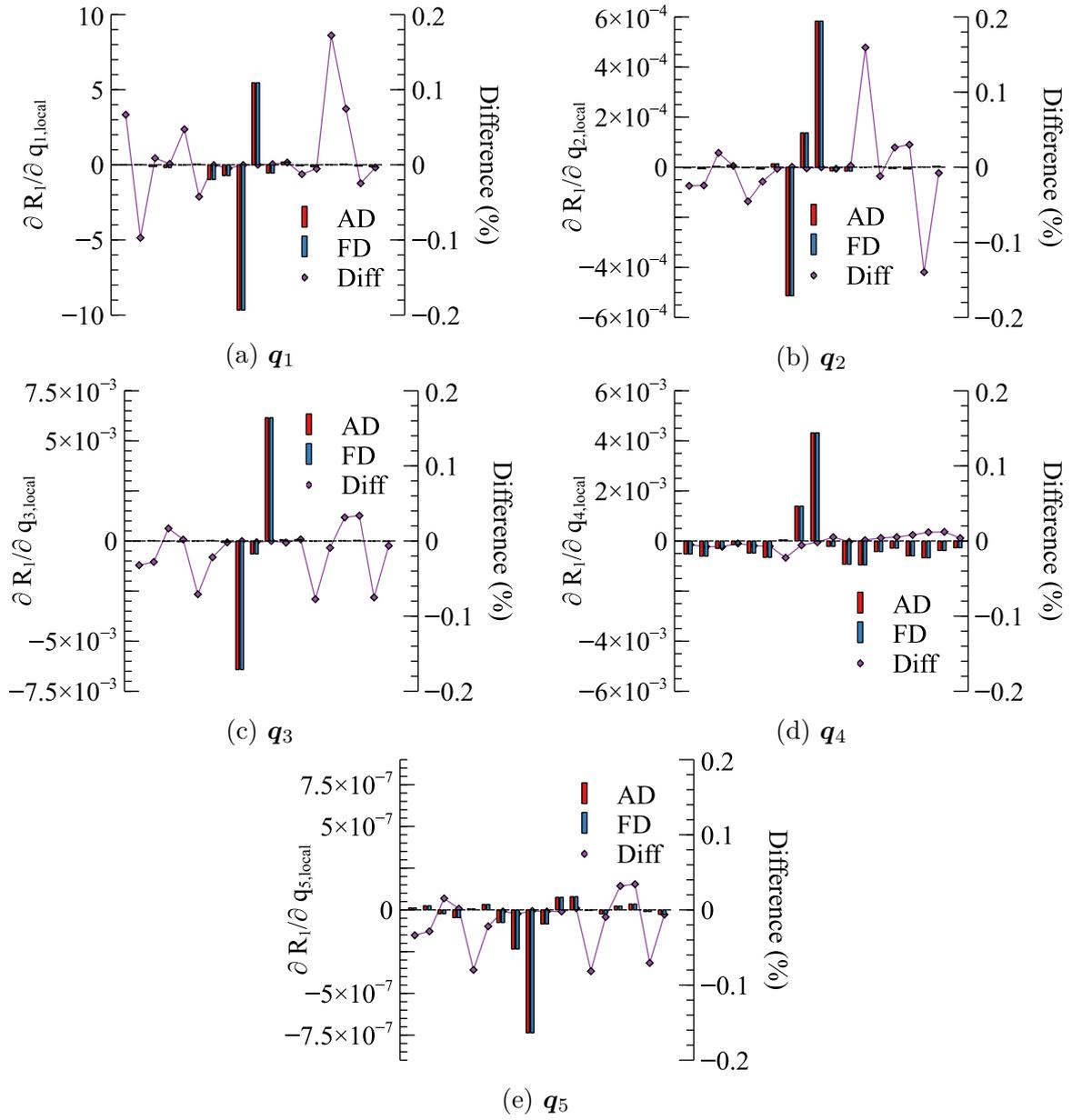


Figure 6.13: Comparison of mixing-plane local sensitivities of the residual of a cell at the inlet of the rotor computed with differentiated routines and FD approximations.

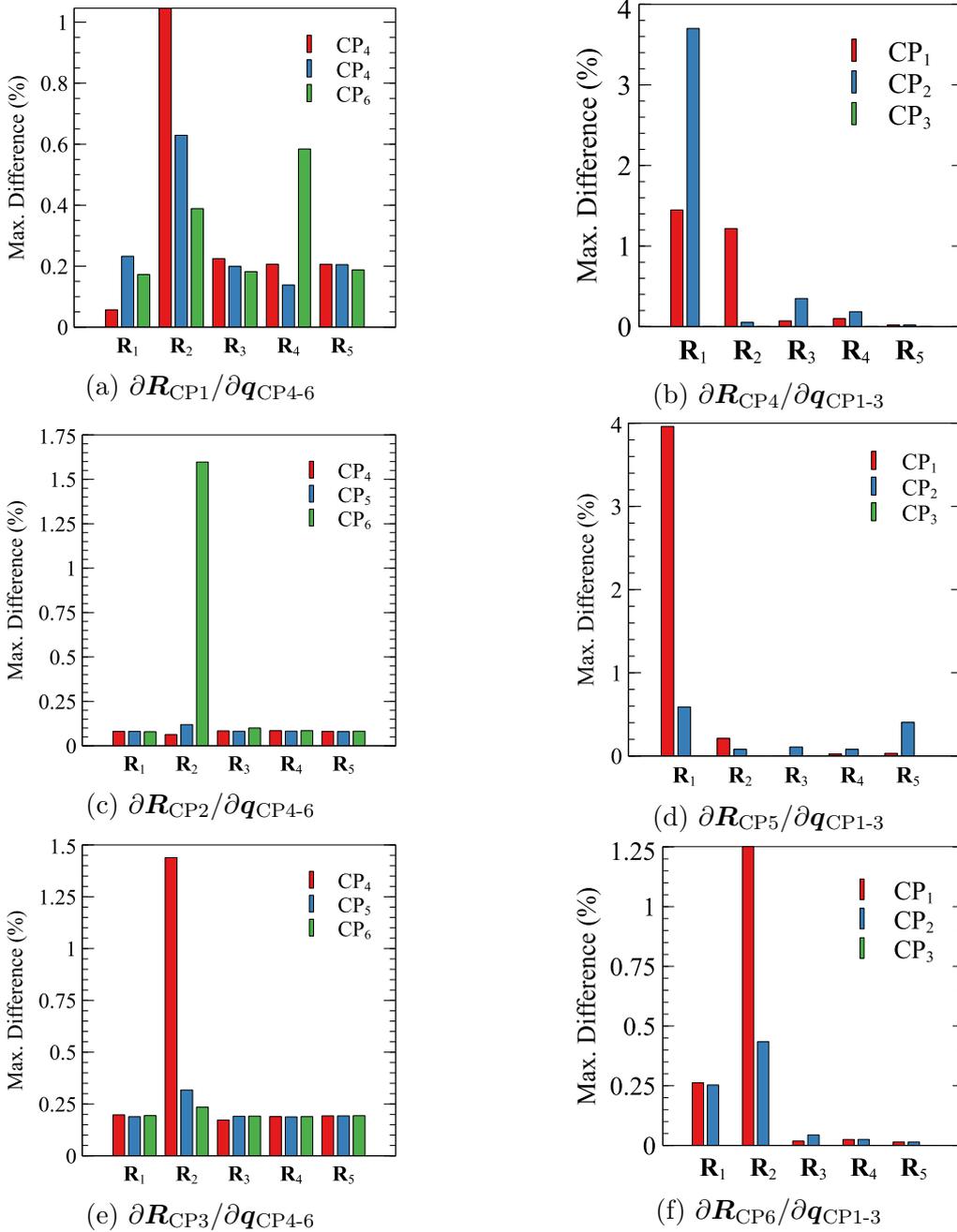


Figure 6.14: Maximum relative difference between inter-row sensitivities obtained with differentiated routines and second-order finite-difference approximations.

## 6.6 Verification of the Adjoint-based Total Derivatives

Following the successful verification of the correct rewriting, differentiation and hand-assembling of the mixing-plane routines, the next step is verifying the final objective of the implementation of the adjoint mixing-plane interface: the total sensitivity analysis.

In this section, the adjoint-based sensitivity of a selection of the six metrics of performance introduced in section 6.2 to various independent variables introduced in section 6.3 are compared to first-order FD-based sensitivities, as

$$\frac{d\mathcal{I}}{d\alpha_i} = \frac{\mathcal{I}(\alpha_i + h) - \mathcal{I}(\alpha_i)}{h}. \quad (6.10)$$

Table 6.1 presents a summary of the various combinations of metrics/design variables here presented. For these verifications, four nodes were randomly selected at the inlet and

$\frac{d\mathcal{I}}{d\alpha}$	$\pi_{\text{stator}}$	$\pi_{\text{rotor}}$	$\pi_{\text{stage}}$	$\eta_{\text{rotor}}$	$\eta_{\text{stage}}$	$\dot{m}^{\text{out}}$	$(I_{Ta})_{\text{rotor}}$
$h_T^{\text{inlet}}$	-	✓	✓	-	-	-	-
$p_T^{\text{inlet}}$	-	-	-	-	✓	✓	✓
$V_t^{\text{inlet}}$	-	-	-	✓	-	-	-
$\phi^{\text{inlet}}$	-	-	-	-	-	-	-
$p_s^{\text{exit}}$	✓	-	✓	-	-	-	-
$r_{\text{hub}}$	-	✓	-	-	-	-	-
$n_{\text{blade}}$	-	-	-	✓	-	-	-

Table 6.1: Selection of test cases included in verification of total derivatives.

exit boundary faces of the stator-rotor stage, as well as four at the hub of the stator and four at the surface of the stator blade (two on the pressure and two on the suction side), leading to a total of 16 control points. The flow solver was run to convergence for every perturbation imposed on the design variable defined at each control node. The results here presented would in theory, assuming the optimal perturbation step was known for each case, lead to 24 extra flow solver runs (six selected design variables times 4 control points). In practice, the number of runs was much higher. As the FD approximations are highly sensitive to the perturbation step size, a manual search had to be performed to obtain the optimal step for each control node to obtain a good trade-off between truncation error due to large step sizes and subtractive cancellation due to too small perturbations. This led to a minimum of five perturbed solver runs to obtain an adequate perturbation step, and in many cases, much more were required. This highlights the advantage of the adjoint method over finite-differences, for cases as the one in analysis, where the number of design variables largely surpasses the number of functions of interest.

Still, in order to truly access this advantage, one needs to know the computational requirements of the adjoint solver.

A comparison of the computational requirements of the direct and adjoint solvers is presented in table 6.2. While the CPU time requirements of computing both solutions is approximately the same, the memory requirements of the adjoint solver increase tenfold compared to the direct solver. This is a direct consequence of the solution method chosen for the adjoint system of equations, that includes a full matrix storage and the iterative GMRES method. This effect could be considerably mitigated if matrix-free algorithms [123] or pseudo-time marching Runge-Kutta methods [57] were employed instead.

The table also presents a detailed description of the memory and CPU time required by each of the processes of computing the adjoint solution, namely the 1) preprocessing, 2) assembling of the system of equations, 3) computing the solution with the GMRES solver, 4) assembling the matrices/vectors to compute the total derivatives, 5) computing the total derivatives and 6) Output of the solutions to files. As observed, the assembly of the matrices and the solution of the adjoint system of equations take the bulk of the required CPU time and memory, being roughly the same for each part.

	CPU time	Memory
Direct	1	1
Adjoint	$\sim 1$	$\sim 10$
Preprocess <sup>(1)</sup>	0.3%	-
Assemble $\frac{\partial \mathbf{R}}{\partial \mathbf{q}}, \frac{\partial \mathcal{I}}{\partial \mathbf{q}}$ <sup>(2)</sup>	51.1%	$\sim 50\%$
GMRES Solver <sup>(3)</sup>	42.2%	$\sim 50\%$
Assemble $\frac{\partial \mathbf{R}}{\partial \mathbf{U}}, \frac{\partial \mathcal{I}}{\partial \mathbf{U}}$ <sup>(4)</sup>	6.2%	-
Compute Sensitivity <sup>(5)</sup>	0.1%	-
I/O <sup>(6)</sup>	0.1%	-

Table 6.2: Comparison of computational requirements of direct and adjoint solvers (normalized by the direct solver).

### 6.6.1 Sensitivity of Stator Pressure Loss

The normalized adjoint-based sensitivity of stator pressure loss  $\pi_{\text{stator}}$  to exit static pressure  $p_s^{\text{exit}}$  boundary condition of the rotor domain can be observed in figure 6.15, where the contour plot is shown for the rotor exit surface. It is worth noting that the sensitivity is always negative, implying that stator pressure ratio would decrease with an increase of the static exit pressure, as expected. This effect is largest at the exit section midspan, away

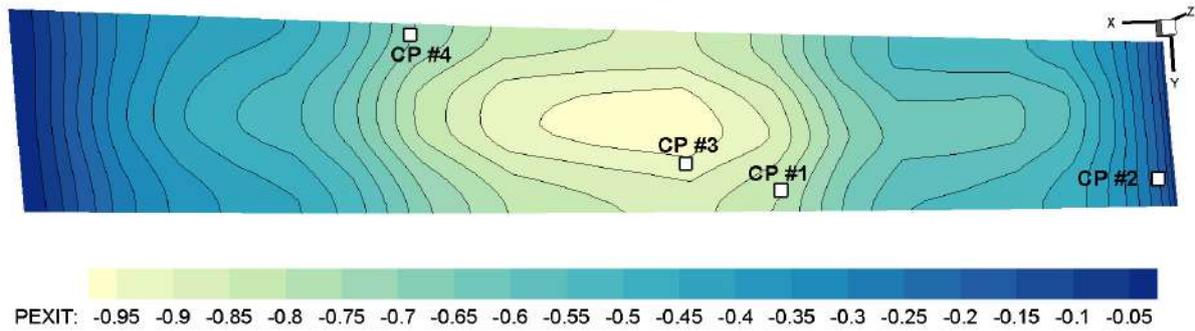


Figure 6.15: Adjoint-based sensitivity of stator pressure ratio  $\pi_{\text{stator}}$  to exit static pressure  $p_s^{\text{exit}}$  boundary condition (normalized values).

from the hub and casing walls and aligned with the rotor wake. This probably means that the effect is amplified when reducing the blockage effect of the blade.

The results of the verification of the adjoint-based sensitivity of the four control nodes identified in figure 6.15 are presented in figure 6.16, where, for each control node, the normalized adjoint- and FD-based sensitivities are plotted (black and white bars, respectively), along with the relative difference between the two (gray bar). The sensitivities

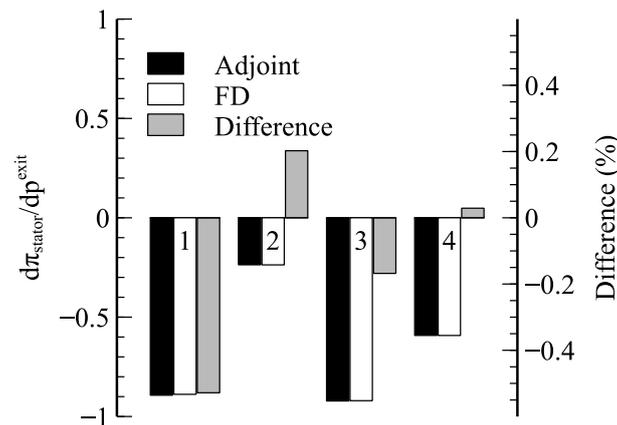


Figure 6.16: Verification of adjoint-based sensitivities of stator pressure loss  $\pi_{\text{stator}}$  to exit static pressure  $p_s^{\text{exit}}$  boundary condition using FD (normalized values).

computed by both methods differ by less than 0.5%, which attests the correct numerical implementation of the adjoint mixing-plane interface.

## 6.6.2 Sensitivity of Rotor Pressure Ratio

Figure 6.17 presents the normalized sensitivity of the pressure ratio of the rotor  $\pi_{\text{rotor}}$  to the radial position of the hub wall  $r_{\text{hub}}$ . This sensitivity, when positive, indicates that moving the grid node into the positive radial position (towards the casing, or the reader) will produce an increase in  $\pi_{\text{rotor}}$ . It can then be inferred from the figure that the rotor pressure ratio can be increased by contouring the hub wall in different ways: making humps on the positive derivative regions and/or making recessions at the negative derivative regions. The multi-row coupling manifests itself in this example since there is a clear effect of stator hub endwall contouring on the rotor pressure ratio. Some oscillations, visible near the inlet and outlet of the stage, may result from the pointwise perturbation of the mesh and from non-reflectivity only being enforced at the mixing plane and not at the inlet and exit of the stage.

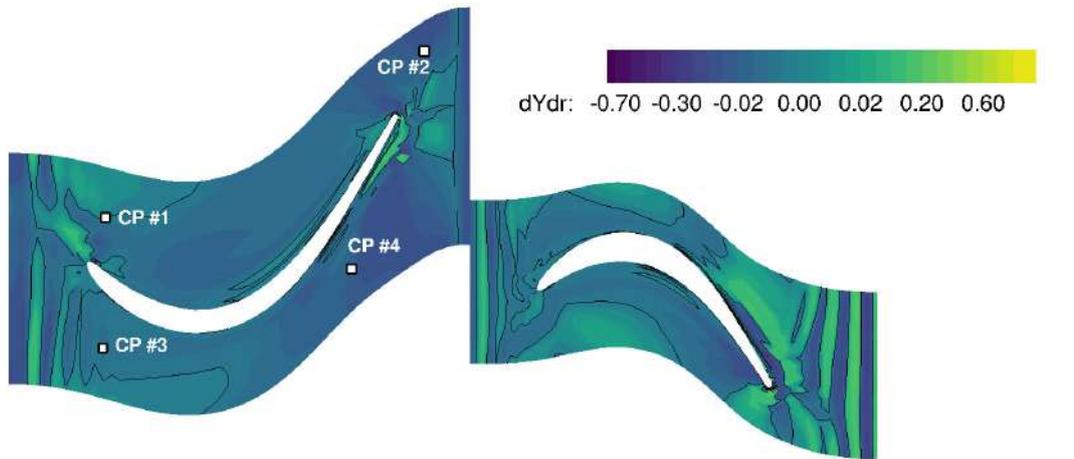


Figure 6.17: Adjoint-based sensitivity of rotor pressure ratio  $\pi_{\text{rotor}}$  to hub wall radial position  $r_{\text{hub}}$  (normalized values).

Recalling the definition of the sensitivity to radial position given by equation (6.8), which depends both on  $d/dx$  and  $d/dy$ , the results of the verification of the sensitivity of  $\pi_{\text{rotor}}$  to the hub wall x-coordinates are shown in figure 6.18, exhibiting again very good agreement with the FD approximation, with relative differences below 0.9%.

Figure 6.19 presents the normalized adjoint-based sensitivity of rotor pressure ratio  $\pi_{\text{rotor}}$  to the total enthalpy  $h_T^{\text{inlet}}$  imposed at the inlet of the stator. The first observation that can be taken from the figure is that all values are positive, meaning that increasing the total enthalpy at the entrance of the stage, leading to larger local temperatures, would increase the pressure ratio of the rotor. From the figure, we can see that the regions of

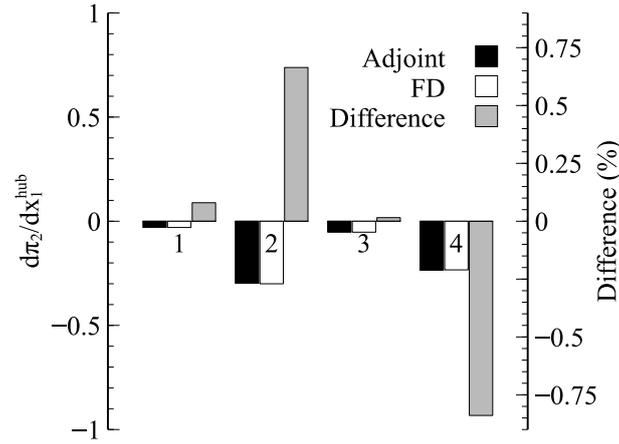


Figure 6.18: Verification of adjoint-based sensitivities of rotor pressure ratio  $\pi_{\text{rotor}}$  to hub wall grid x-coordinates using FD (normalized values).

higher sensitivity of  $\pi_{\text{rotor}}$  with respect to  $h_T^{\text{inlet}}$  are in the mid-span region and closer to the hub (left side of the figure). The results of the verification of the sensitivity presented

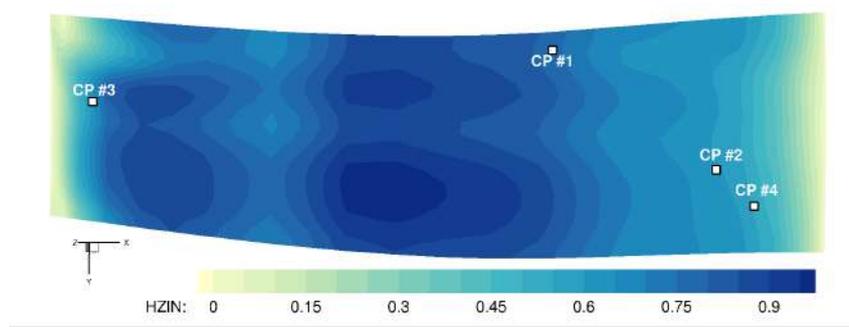


Figure 6.19: Adjoint-based sensitivity of rotor pressure ratio  $\pi_{\text{rotor}}$  inlet total enthalpy  $h_T^{\text{inlet}}$  boundary condition (normalized values).

in figure 6.19 are shown in figure 6.20, exhibiting very good agreement with the FD approximation, with relative differences below 1%.

### 6.6.3 Sensitivity of Stage Pressure Ratio

The normalized adjoint-based sensitivity of total pressure ratio of the whole stage  $\pi_{\text{stage}}$  to exit static pressure  $p_s^{\text{exit}}$  boundary condition is presented in figure 6.21. Contrary to the sensitivity of the pressure losses of the stator, the entirety of the exit face presents positive sensitivities, meaning that an increase in the static exit pressure would results in an increase in the pressure ratio of the whole stage. This is with agreement with the definition of the total pressure ratio and stator pressure losses. The higher sensitivities are, as for the case of stator pressure loss, located mid-span, aligned with the wake of the

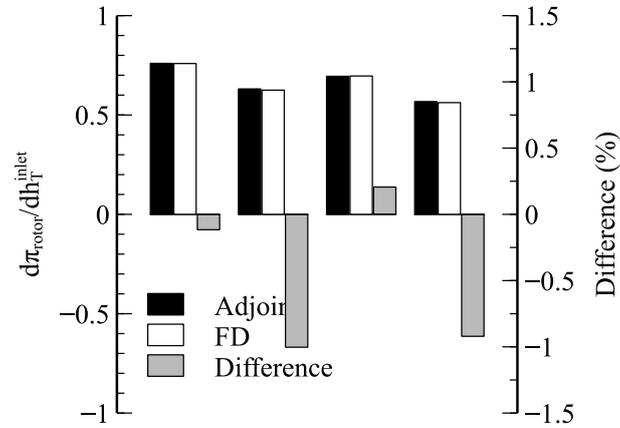


Figure 6.20: Verification of adjoint-based sensitivities of rotor pressure ratio  $\pi_{\text{rotor}}$  to inlet total enthalpy  $h_T^{\text{inlet}}$  boundary condition using FD (normalized values).

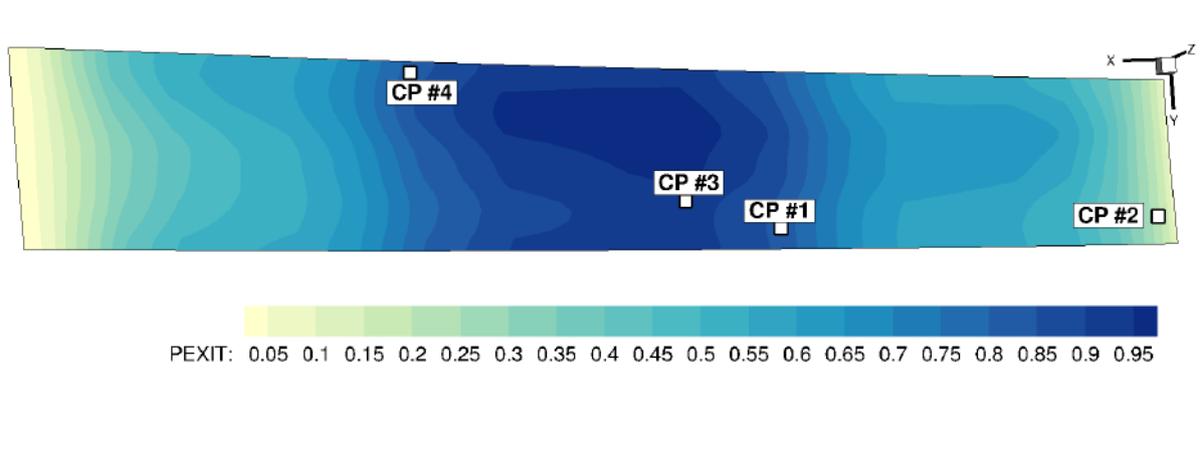


Figure 6.21: Adjoint-based sensitivity of stage total pressure ratio  $\pi_{\text{stage}}$  to exit static pressure  $p_s^{\text{exit}}$  boundary condition (normalized values).

rotor.

Figure 6.22 presents the comparison of the adjoint-based sensitivities of the total pressure ratio of the stage to rotor exit static pressure with finite-difference approximations. As for the other cases, the FD approximations and adjoint-based sensitivities present good agreement, with maximum relative differences slightly above 0.04%.

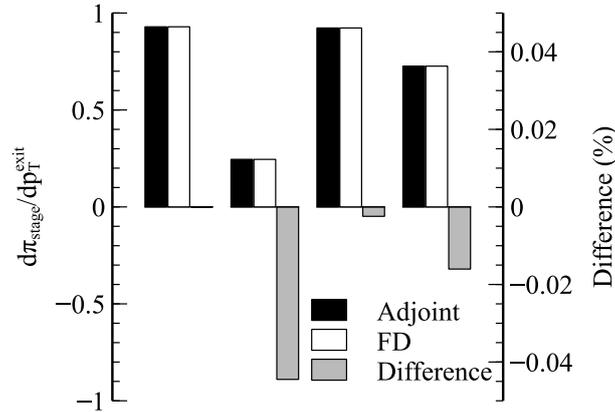


Figure 6.22: Verification of adjoint-based sensitivities of stage total pressure ratio  $\pi_{\text{stage}}$  to exit static pressure  $p_s^{\text{exit}}$  boundary condition using FD approximations (normalized values).

### 6.6.4 Sensitivity of Rotor Efficiency

Figure 6.23 presents the adjoint based sensitivity of the rotor efficiency  $\eta_{\text{rotor}}$  to the shape of the stator and rotor blades. In this case, the contour shown is the magnitude of the sensitivity vector projected onto the blade surface outer normal at each point, as defined by equation (6.7).

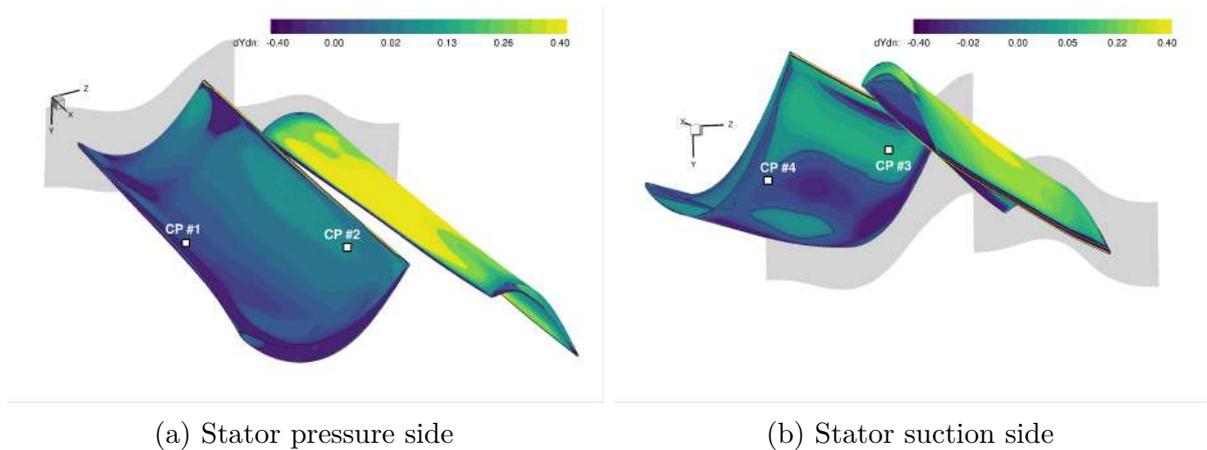


Figure 6.23: Adjoint-based sensitivity of rotor efficiency  $\eta_{\text{rotor}}$  to stator and rotor blade shapes in normal direction (normalized values).

Similarly to the hub contouring test case, this test case also demonstrates the coupling between rows by quantitatively showing the impact of the stator blade shape on the rotor efficiency. Analyzing figure 6.23, it can be seen that the rotor efficiency can be increased by moving the stator blade in the positive (negative) outer normal direction at the regions of positive (negative) derivatives.

The control nodes considered for the verification procedure are also identified in figure 6.23, two at the suction side and two at the pressure side of the blade. The derivatives of rotor efficiency with respect to the y-coordinate of the selected control nodes of the stator blade surface were compared to finite-differences and the results are shown in figure 6.24 (note that the values are normalized to the maximum value of the derivative at both row domains). Good agreement is again obtained, with a maximum relative difference of 1.1%. As with the hub control nodes, a good agreement with the FD approximation was also obtained for the other two coordinates, x- and y-coordinates.

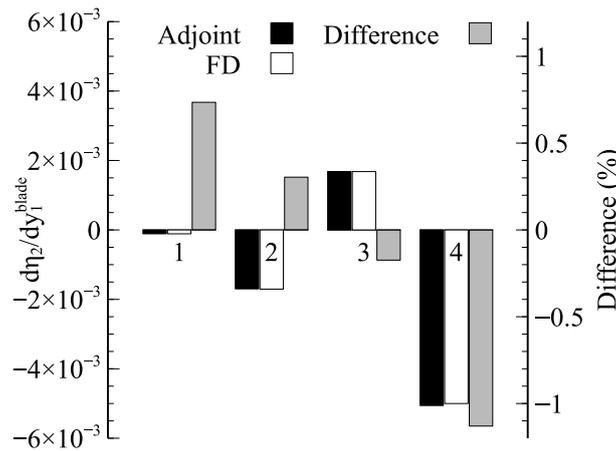


Figure 6.24: Verification of adjoint-based sensitivities of rotor efficiency to stator and rotor blade surface grid y-coordinates using FD (normalized values).

Another example of coupling is shown in figure 6.25 which shows the effect of inlet tangential velocity  $V_t$  on the stator efficiency  $\eta_{\text{stator}}$ , represented on the inlet plane. From the contour plot of figure 6.25, the stator efficiency sensitivity to the inlet tangential velocity varies considerably depending on both the radial and tangential location. This variation is particularly strong (positive) closer to the hub and at midspan (negative). Such rich information can be extremely useful with analyzing turbomachines at off-design conditions, such as when inflow distortion occurs [124].

Figure 6.26 presents the comparison of the adjoint-based sensitivities to FD approximations. In this case, the optimum step for the FD approximation was harder to obtain,

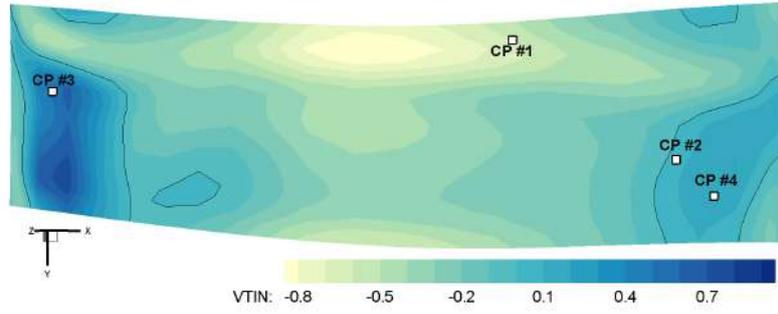


Figure 6.25: Adjoint-based sensitivity of rotor efficiency  $\eta_{\text{rotor}}$  to inlet tangential velocity  $V_t^{\text{inlet}}$  boundary condition (normalized values).

particularly in control node 2, where the minimum relative error we were able to obtain was approximately 3.5%. This was probably due to the differences of order of magnitude

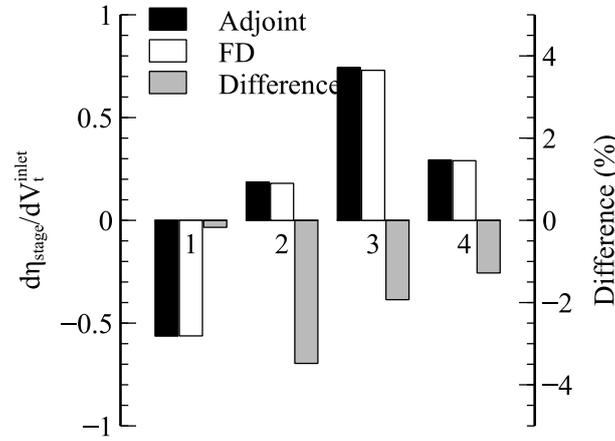


Figure 6.26: Verification of adjoint-based sensitivities of rotor efficiency  $\eta_{\text{rotor}}$  to inlet tangential velocity  $V_t^{\text{inlet}}$  boundary condition using FD (normalized values).

of the derivatives  $\mathcal{O}(10^{-6})$ , function of interest  $\mathcal{O}(10^1)$  and quantity to perturb  $\mathcal{O}(10^2)$ . The difference in magnitudes is not visible in the bar plots due to normalization. This difficulty highlights the advantages of the adjoint method over the FD method, as the adjoint-based sensitivities avoid the concept of perturbation step altogether.

### 6.6.5 Sensitivity of Stage Efficiency

We can also assess the efficiency of the whole stage  $\eta_{\text{stage}}$ , instead of only the rotor. Figure 6.27 presents the normalized adjoint-based sensitivity of the efficiency of the stage to inlet total pressure  $p_T^{\text{inlet}}$ . A positive high sensitivity region near the hub endwall is clear from the figure, indicating that increasing the total pressure through increasing velocity would lead to an increase of the overall efficiency of the stage, highlighting the importance

of the previously mentioned endwall contouring.

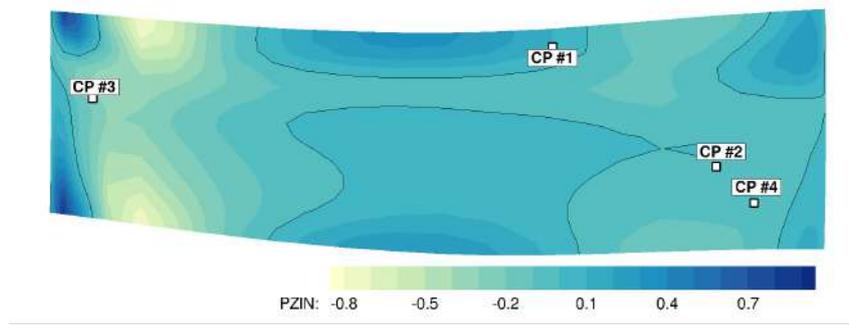


Figure 6.27: Adjoint-based sensitivity of stage efficiency  $\eta_{\text{stage}}$  to inlet total pressure  $p_T^{\text{inlet}}$  boundary condition (normalized values).

Figure 6.28 presents the comparison of the adjoint-based sensitivities at the control points highlighted in figure 6.27 with FD approximations. In this case, the relative differences are larger than in previous cases with a maximum relative difference slightly above 4%. This is due to not having found the optimal perturbation step and, albeit this slightly larger relative differences, the absolute differences are still very small with both values having the same signal.

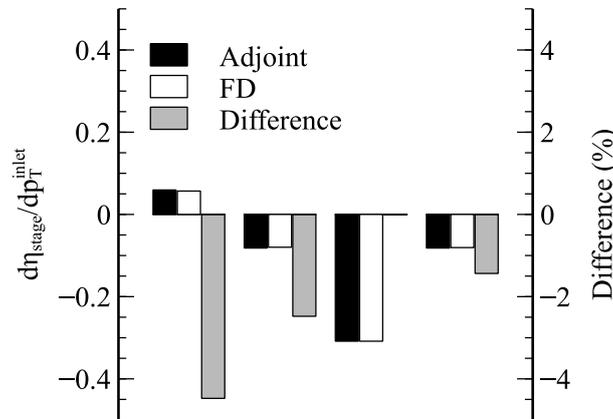


Figure 6.28: Verification of adjoint-based sensitivities of stage efficiency  $\eta_{\text{stage}}$  to inlet total pressure  $p_T^{\text{inlet}}$  boundary conditions using FD (normalized values).

### 6.6.6 Sensitivity of Exit Mass Flow

The sensitivity of the mass flow at the exit of the rotor  $\dot{m}^{\text{out}}$  to the total pressure boundary condition imposed at the inlet  $p_T^{\text{inlet}}$  of the rotor is presented in figure 6.29. The values are normalized by the maximum absolute value of the derivative. The positive derivative, exhibited in almost all inlet section locations, reveals the expected increase of mass flow with the increase of inlet total pressure.

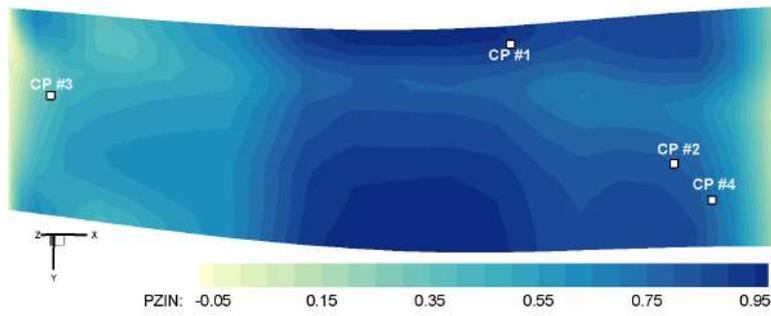


Figure 6.29: Adjoint-based sensitivity of outlet mass flow  $\dot{m}^{\text{out}}$  to inlet total pressure  $p_T^{\text{inlet}}$  boundary condition (normalized values).

The adjoint-based derivative values also show good agreement with the FD approximation, as seen in figure 6.30, where the results of the verification presented for the four control nodes exhibit differences smaller than 0.5%.

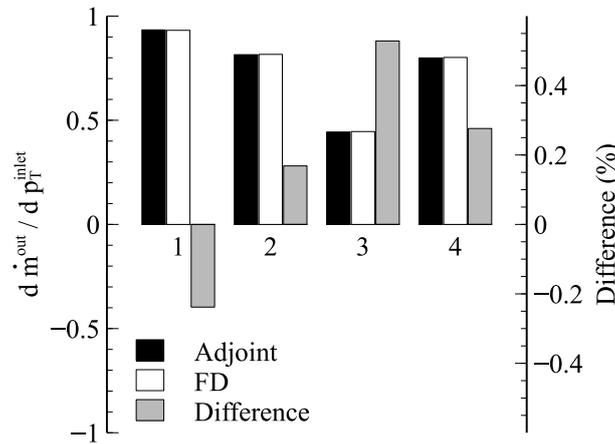


Figure 6.30: Verification of adjoint-based sensitivities of mass flow  $\dot{m}^{\text{out}}$  to inlet total pressure  $p_T^{\text{inlet}}$  boundary condition using FD (normalized values).

### 6.6.7 Sensitivity of Rotor Blade Averaged Total Temperature

The sensitivity of the area averaged total temperature at the surface of the rotor blade to stage inlet total pressure  $p_T^{\text{inlet}}$  is presented in figure 6.31, where the four control points used for verification with finite-differences are highlighted. Negative values across the majority of the inlet indicate that increasing total pressure in those regions would lead to a increase in the averaged total temperature of the rotor blade. At the hub and casing we observe from the figure that the opposite occurs. As from previous results highlighted, these regions are very prone to high sensitivities, with signal changes in close areas, indicating a possible region of complex flows.

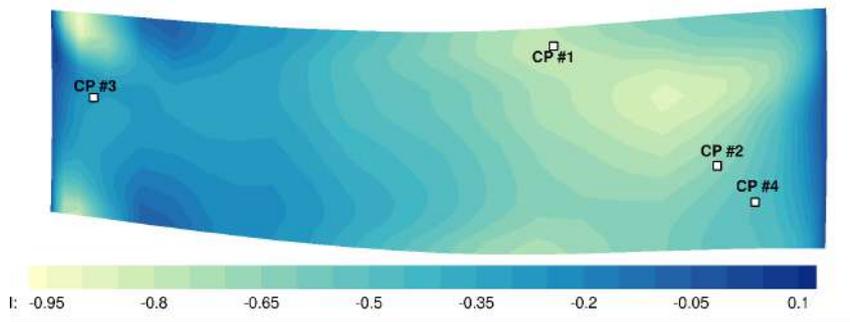


Figure 6.31: Adjoint-based sensitivity of averaged total temperature of the rotor blade  $(T_{Ta})_{\text{rotor}}$  to inlet total pressure  $p_T^{\text{inlet}}$  boundary condition (normalized values).

Figure 6.32 presents the comparison of the normalized adjoint-based sensitivities with finite-difference approximations. Very good agreement is visible from the figure, with a maximum relative difference close to 1%.

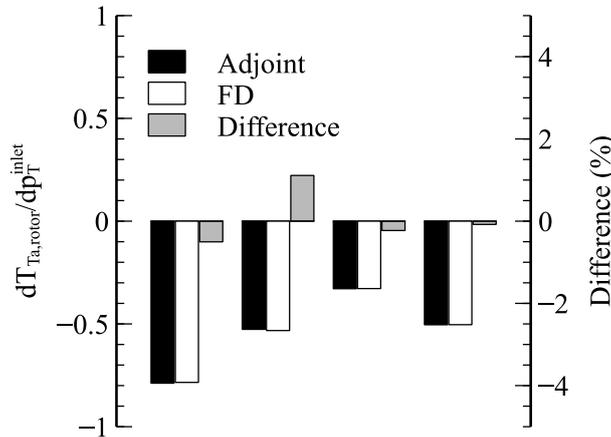


Figure 6.32: averaged total temperature of the rotor blade  $(T_{Ta})_{\text{rotor}}$  to inlet total pressure  $p_T^{\text{inlet}}$  boundary condition using FD (normalized values).

## 6.7 Final Remarks

This chapter presented the numerical verification of the correctness of the implementation of the adjoint mixing-plane interface, necessary to compute sensitivities involving multi-row domains. The adjoint-based sensitivity results were compared to FD approximations with which very good agreement was obtained. Throughout the verification process, the weaknesses of the FD method were highlighted, particularly their sensitivity to the perturbation step. While the adjoint method required one additional solver run for each function of interest, with a computational cost similar to flow direct run in terms of CPU time, the finite-difference approach required many direct solver runs to obtain a converged

value, thus emphasizing the benefits of using the adjoint method for sensitivity analysis, particularly when a large number of design variables is used. With the confidence on the correctness of the results given by the multi-row adjoint solver, the following chapter will present results of sensitivity analyses using this verified adjoint solver.

# Chapter 7

## Sensitivity Analysis of a Stator/Rotor Stage of a Low Pressure Turbine

This chapter presents a comprehensive analysis of various sensitivities, obtained of the stator/rotor stage introduced in chapter 6, using the adjoint multi-row solver.

The chapter is divided in three sections, 1) sensitivity to boundary conditions, 2) sensitivity to blade shape and hub geometry and 3) aero-thermal sensitivity analysis. All provide extremely important information to the engineer interested in designing a new turbomachine component or trying to improve an existing one.

The performance metrics  $\mathcal{I}$  presented in this chapter are the ones described in the previous chapter, namely rotor efficiency  $\eta_{\text{rotor}}$ , stage efficiency  $\eta_{\text{stage}}$ , rotor pressure ratio  $\pi_{\text{rotor}}$ , stage pressure ratio  $\pi_{\text{stage}}$ , and the area averaged total temperature  $T_{Ta}$ .

Similarly to the previous chapter, a summary of the results presented in the current chapter is displayed in table 7.1. The table is divided into three sections, corresponding to the three sections of this chapter.

This selection of results aims to highlight the coupling between the rows, by showing how various metrics are influenced by parameters defined across the individual row domains. The choice of presenting the same metrics for both stage and rotor will evidence the impact of considering the metric of the full stage in contrast to considering only one row (the rotor, in this case).

	$\frac{d\mathcal{I}}{d\alpha}$	$(T_{Ta})_{\text{rotor}}$	$\pi_{\text{rotor}}$	$\pi_{\text{stage}}$	$\eta_{\text{rotor}}$	$\eta_{\text{stage}}$
Boundary Conditions	$p_T^{\text{inlet}}$	-	✓	✓	✓	✓
	$V_t^{\text{inlet}}$	-	✓	✓	✓	✓
	$h_T^{\text{inlet}}$	-	✓	✓	✓	✓
	$\phi^{\text{inlet}}$	-	-	-	-	-
	$p_s^{\text{exit}}$	-	✓	✓	-	-
Blade Shape / Hub Geometry	$n_{\text{blade}}$	-	-	✓	-	✓
	$r_{\text{hub}}$	-	-	✓	-	✓
Aero-thermal Analysis	$p_T^{\text{inlet}}$	✓	-	-	-	-
	$h_T^{\text{inlet}}$	✓	-	-	-	-
	$V_t^{\text{inlet}}$	✓	-	-	-	-

Table 7.1: Selection of test cases included in verification of total derivatives.

## 7.1 Sensitivity to Boundary Conditions

Boundary conditions (BC) at inlet and exit surfaces are typically defined (as is the case of the current solver) as radial profiles. The adjoint solver not only provides the sensitivity to those radial 1D profiles, but also to all individual cells of the inlet/exit 2D surfaces. Both results have their uses, while the full 2D results provide a more detailed insight to the behavior of the performance metrics to changes in the BC quantities, the 1D profiles can be directly used in optimization environment or manual tweaking of the operating conditions of the component in analysis.

Recalling subsection 6.3.1, the quantities enforced at an inlet boundary are absolute total pressure  $p_T$ , absolute total enthalpy  $h_T$ , absolute tangential velocity  $V_t$ , and velocity direction cosines  $C_r$  and  $C_z$ . At an exit, the only imposed boundary condition is static pressure  $p_s$ . The following subsections present a selection of results of the analysis of how the various boundary conditions influence various performance metrics. The results are presented grouped by performance metric.

### 7.1.1 Rotor and Stage Efficiencies

Figure 7.1 presents the normalized adjoint-based sensitivities of rotor and stage efficiencies to inlet total pressure ( $d\eta/dp_T^{\text{inlet}}$ ) at the inlet of the stage (stator inlet). This represents the impact that upstream conditions (in this case represented by the boundary conditions) have

on components that are located downstream, such as is the case of rotor efficiency  $\eta_{\text{rotor}}$ . The comparison between the two efficiencies aims to provide insight of how considering the stage globally instead of focusing on one component may impact the direction on which the parameters must evolve to improve efficiency.

The contour plots show similar locations of the areas of high and low sensitivities. The main difference between the two cases is the magnitude of the sensitivity, which is higher (although in the same order of magnitude) for the case of stage efficiency. This comes as expected, as, since stage efficiency is computed from averaging on the inlet of the stage, where the boundary conditions are imposed for this particular case. The high positive sensitivity of the efficiency to inlet total pressure near the hub and the casing, particularly in the region between the blades, indicates that increasing the inlet total pressure in that region leads to an improvement of the efficiency of the stage, which might be due to the decrease of the viscous effects on the endwall. Right next to this positive high valued sensitivities near the hub, a large region of strong negative sensitivities is very clear. The hub and casing regions are crucial in a turbomachine component due to the secondary flow phenomena occurring in that region. As such, high sensitivities to boundary conditions in those regions are to be expected. In the mid-span region, between the blades another region of strong sensitivities is clear (albeit not as strong as in the hub). This indicates that increasing the velocity of the flow in that region, leading to an increase of total pressure, would also increase the efficiency of the stage.

The difference in magnitude of sensitivity of the two performance metrics is particularly visible near the hub and the casing, where the stage efficiency is much more sensitive to inlet total pressure, which can be easily seen from figure 7.2, where the radial profiles of sensitivity and total pressure are plotted. In this figure, a difference in magnitude in the region near the casing ( $\text{span} = 1$ ) is also noticeable, as well as the difference in magnitudes of the sensitivities. The sensitivity to inlet total pressure of rotor efficiency is clearly lower than of stage efficiency, with a difference in total (area averaged) sensitivity of 18%. This trend is expected as the diffusion of the flow in the stator tends to "dissipate" the influence of the inlet conditions in quantities computed at downstream locations.

In figure 7.3 we compare the sensitivity of the rotor efficiency to total pressure at the rotor inlet (top) against the sensitivity of the stage efficiency to the same quantity at the stage inlet (bottom). The values are normalized by the maximum value of the

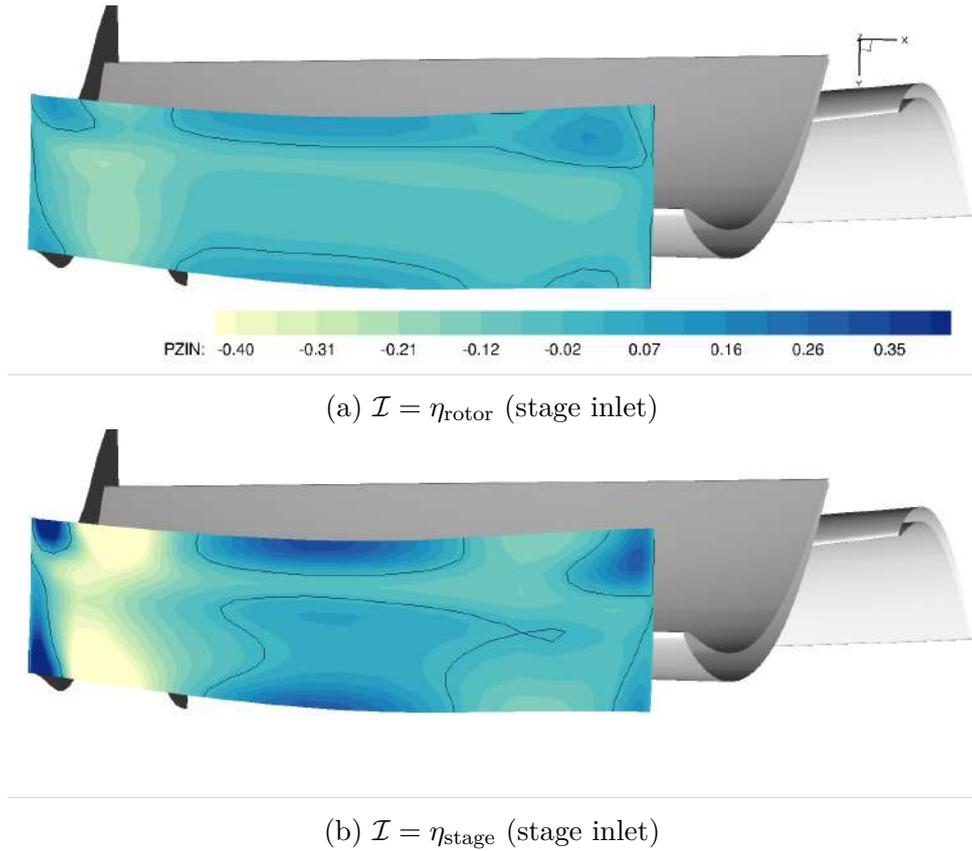


Figure 7.1: Adjoint-based sensitivity of rotor and stage efficiencies to stage inlet total pressure  $d\eta/dp_T^{\text{inlet}}$  (normalized values).

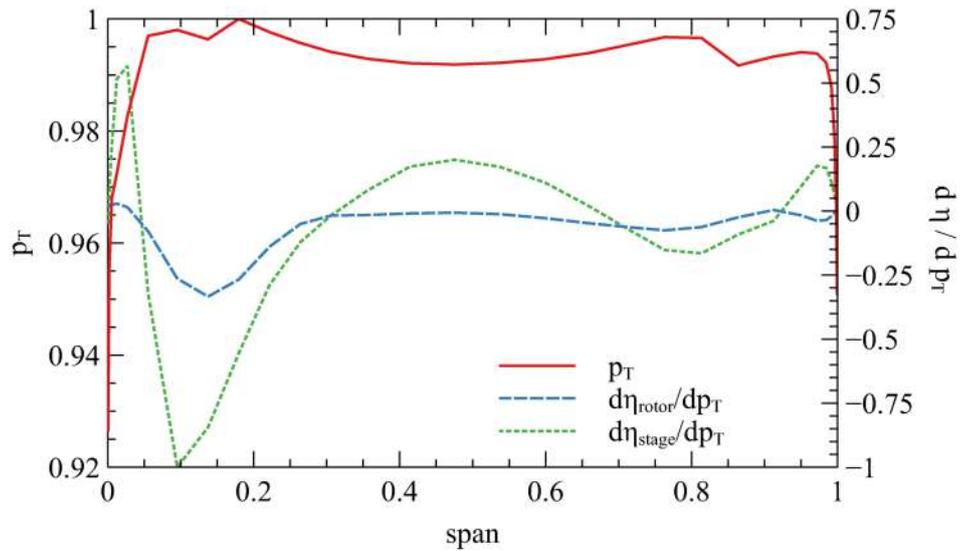


Figure 7.2: Radial distribution of stage inlet total pressure and adjoint-based sensitivities of rotor and stage efficiencies to stage inlet total pressure profiles (normalized values).

two solutions. In this comparison we see that, although the difference between the two is not as evident as in the previous case (see figure 7.1), the stage efficiency shows higher local sensitivity to total pressure, although maintaining the same qualitative shape of the contour.

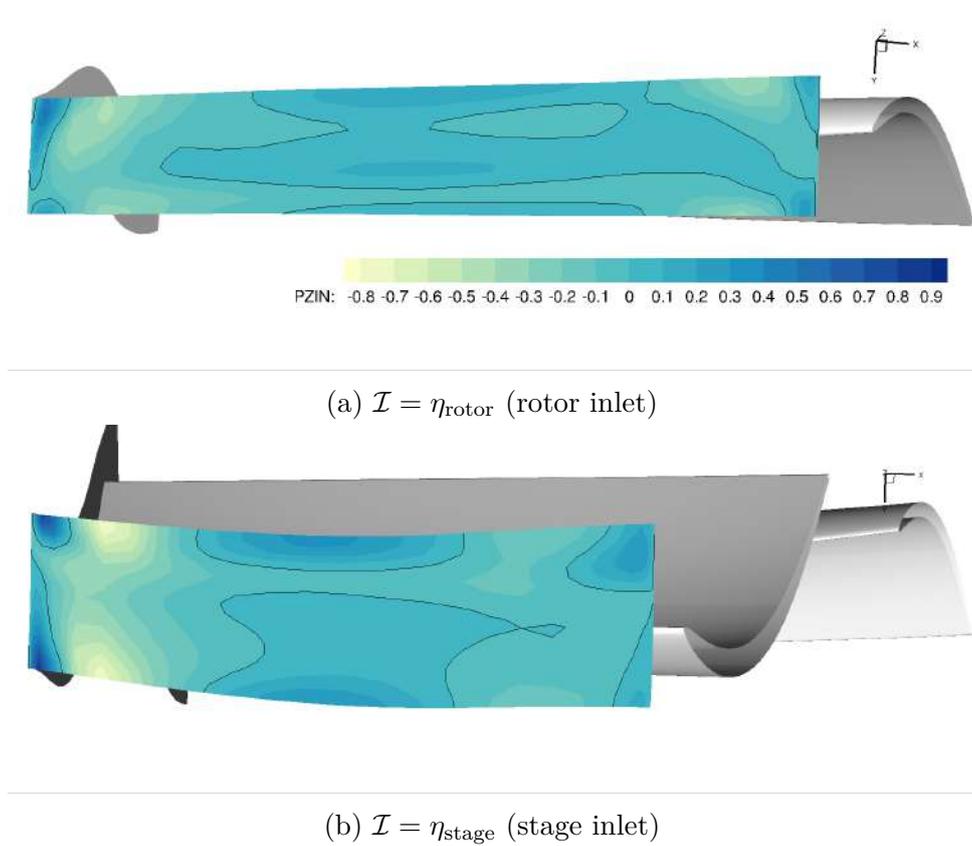


Figure 7.3: Adjoint-based sensitivity of rotor and stage efficiencies to respective inlet total pressure  $d\eta/dp_T^{\text{inlet}}$  (normalized values).

Inlet tangential velocity  $V_t$  can be related to flow distortions resulting from the turbomachines being operating at off-design conditions [124]. Its influence on the stage and rotor efficiencies,  $d\eta/dV_t^{\text{inlet}}$ , shown in figure 7.4, presents the same behavior as inlet total pressure, with the magnitude of the stage sensitivities being higher than the sensitivities across the stator. The contour plots show that reducing the tangential velocity in the area between the blades leads to an increase in efficiency, both when looking to rotor or to stage efficiency. The opposite occurs near the hub, where we observe positive sensitivity, against the overall tendency across the inlet surface. This region also presents values of higher magnitude than in the rest of the domain, highlighting once again the importance of the endwall region in the performance of the stage. Comparing the results of sensitivity to tangential velocity to inlet total pressure, particularly in the region between the blades

and near the hub reveals that the region of positive sensitivities to inlet total pressure presents negative sensitivities to tangential velocity.

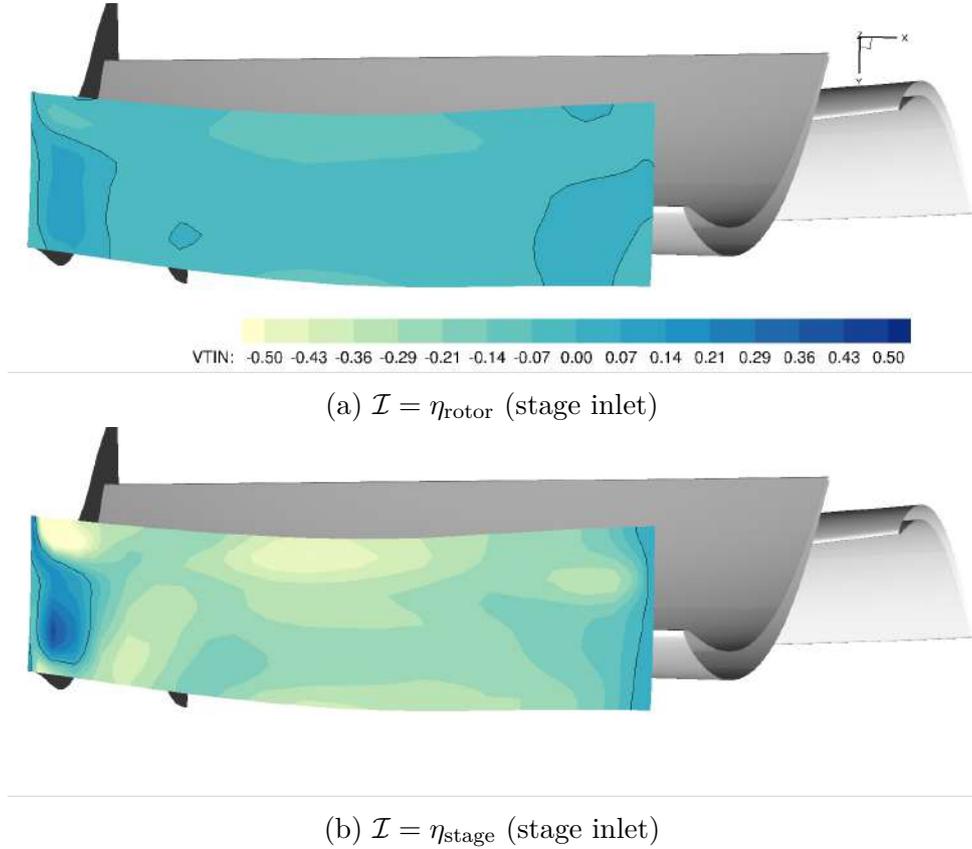


Figure 7.4: Adjoint-based sensitivity of rotor and stage efficiencies to stage inlet tangential velocity  $d\eta/dV_t^{\text{inlet}}$  (normalized values).

In figure 7.5, the adjoint-based sensitivity profiles of rotor and stage efficiencies to inlet tangential velocity are compared to each other. The values presented are normalized by the absolute maximum of the two profiles. Much like the case of inlet total pressure, although the two profiles present similar shapes, the magnitude of the sensitivity to rotor efficiency is considerably smaller than to stage efficiency.

The influence of inlet total enthalpy on the two efficiency metrics in analysis presents a different behavior than the previous quantities. In figure 7.6, higher local values of  $d\eta/dh_T^{\text{inlet}}$  are visible for the case of rotor efficiency than stage efficiency. The shape of the contour is also quite different, particularly in the mid-span region. The tendency for higher (negative) sensitivities near the hub and casing is present in both cases. In figure 7.7, the swap in difference of magnitudes between the two cases is even more noticeable.

The comparison of the influence of total enthalpy at the rotor and stage inlets on rotor and stage efficiencies, respectively, is presented in figure 7.8. These sensitivities

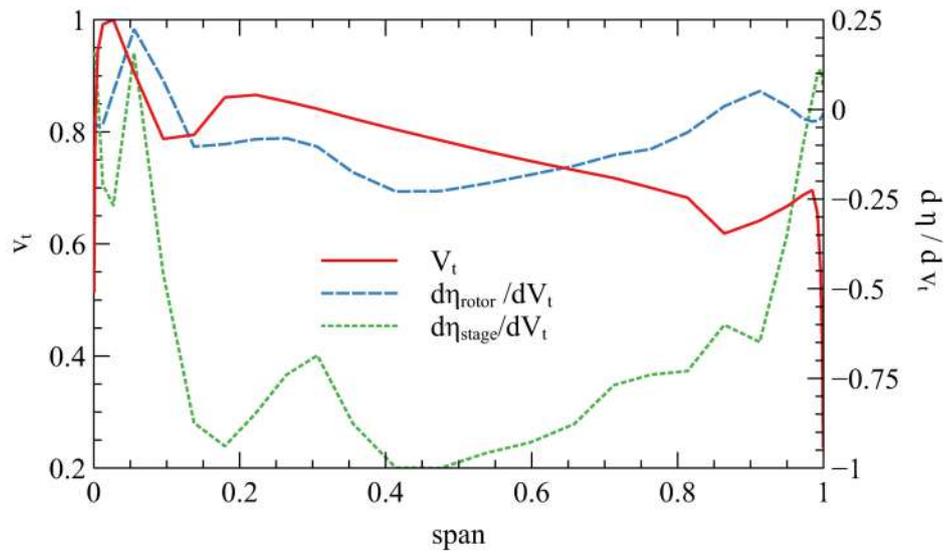


Figure 7.5: Comparison between adjoint-based sensitivity of stage and rotor efficiencies to inlet tangential velocity  $\partial\eta_*/\partial V_t^{\text{inlet}}$  (normalized values).

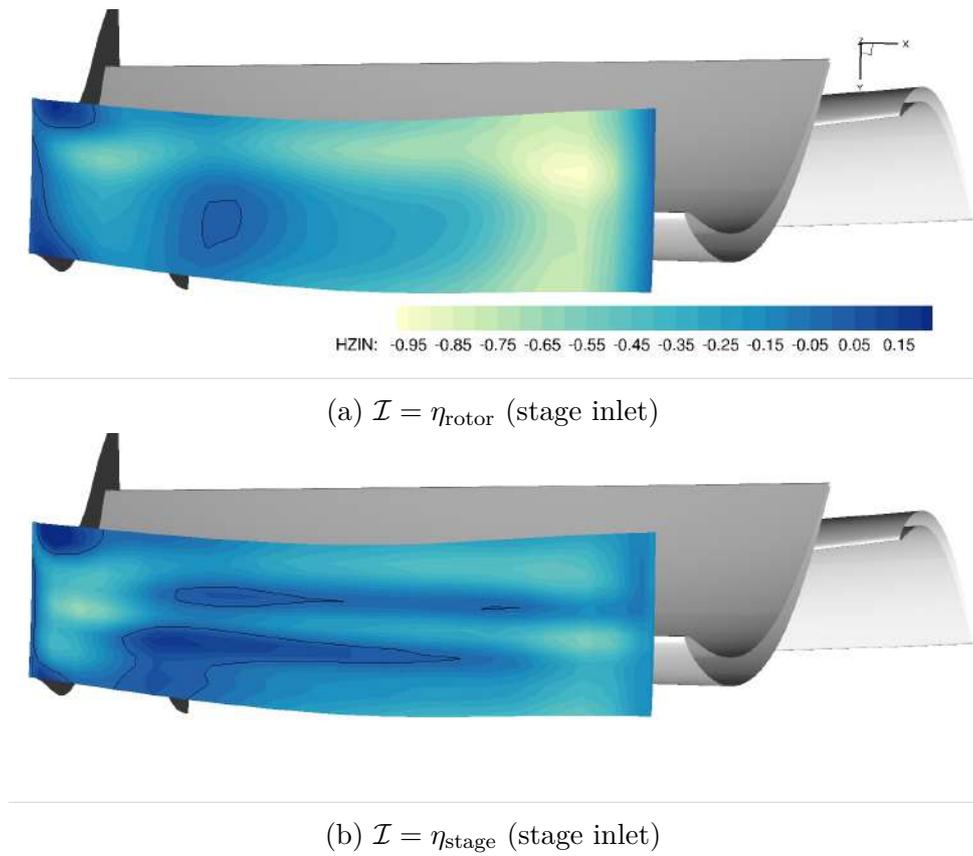


Figure 7.6: Adjoint-based sensitivity of rotor and stage efficiencies to stage inlet total enthalpy  $d\eta/dh_T^{\text{inlet}}$  (normalized values).

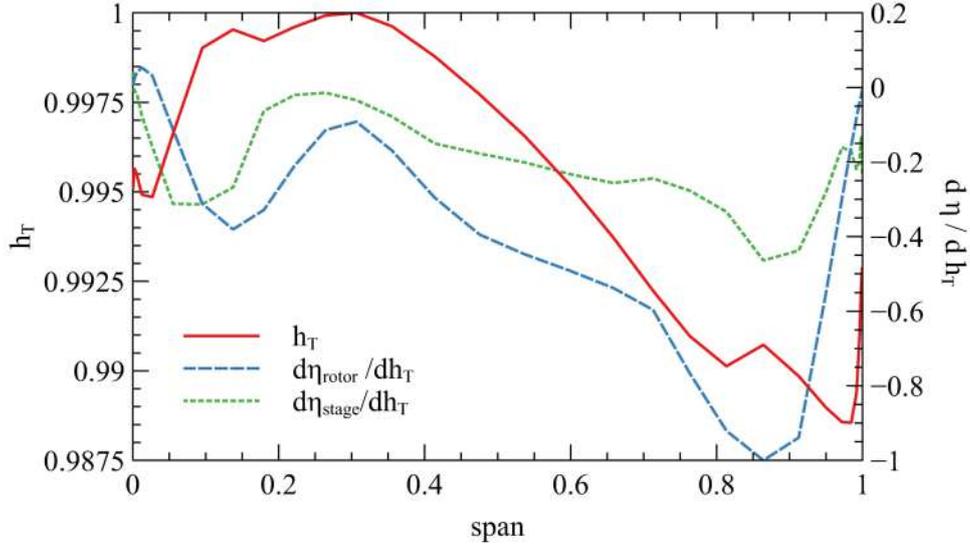
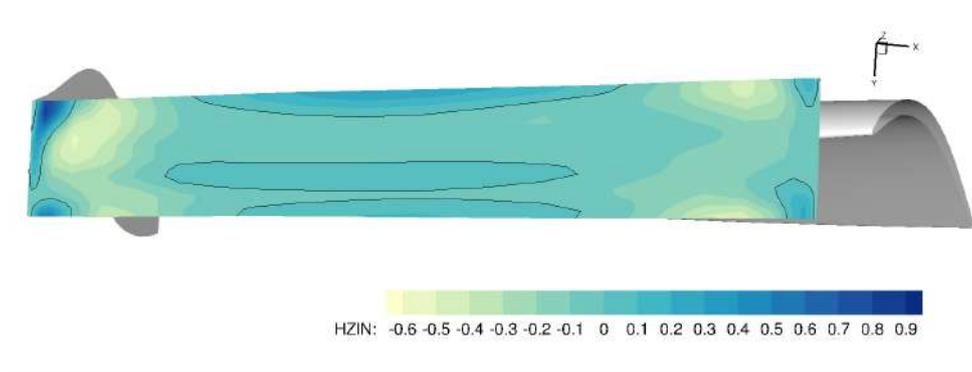


Figure 7.7: Comparison between adjoint-based sensitivity of stage and rotor efficiencies to inlet total enthalpy (normalized values).

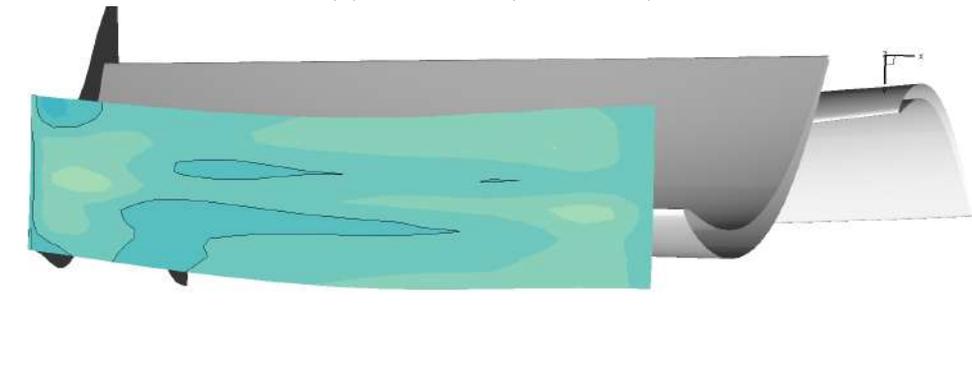
follow the same behavior as when assessing stage inlet boundary conditions only, with the rotor efficiency presenting higher local sensitivity to rotor inlet total enthalpy than stage efficiency is to stage inlet total enthalpy. It is also clear from this figure, the importance of the endwall regions, where the magnitude of the derivatives are much higher than on the rest of the inlet region.

Figure 7.9 shows the normalized averaged radial profile of the three sensitivities analyzed previously,  $d\eta_{\text{rotor}}/dh_T^{\text{stage inlet}}$ ,  $d\eta_{\text{rotor}}/dh_T^{\text{rotor inlet}}$  and  $d\eta_{\text{stage}}/dh_T^{\text{stage inlet}}$ . From the figure, we see the higher magnitude of sensitivity for the single-row case near the hub when compared to the multi-row sensitivities. Near the casing, we see that  $d\eta_{\text{rotor}}/dh_T^{\text{inlet}}$  presents higher values than the other two.

The influence of the stage exit static pressure on both efficiencies in analysis is presented in figure 7.10, where it is visible that the influence of exit static pressure is higher if we look only at the efficiency of the rotor than the whole stage. Both results show that an increase in the stage exit static pressure will result in an improvement of the efficiency. This is particularly evident in the region further away from the trailing edge, which indicates that the effect of reducing the flow speed in the region below the blades would result in an increase of efficiency (as it leads to an increase of static pressure).



(a)  $\mathcal{I} = \eta_{\text{rotor}}$  (rotor inlet)



(b)  $\mathcal{I} = \eta_{\text{stage}}$  (stage inlet)

Figure 7.8: Adjoint-based sensitivity of rotor and stage efficiencies to respective inlet total enthalpy  $d\eta/dh_T^{\text{inlet}}$  (normalized values).

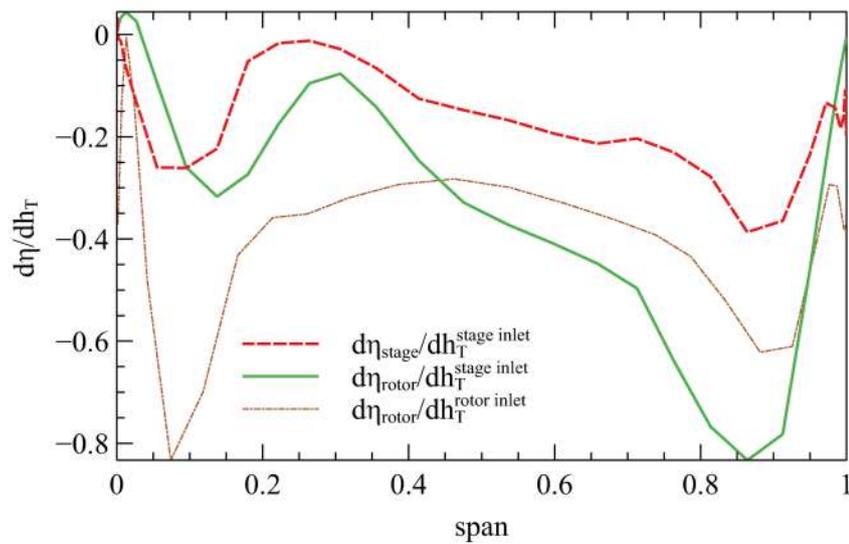
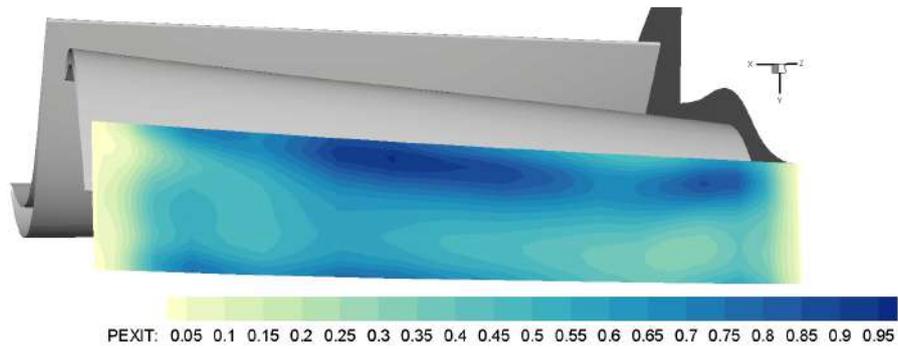
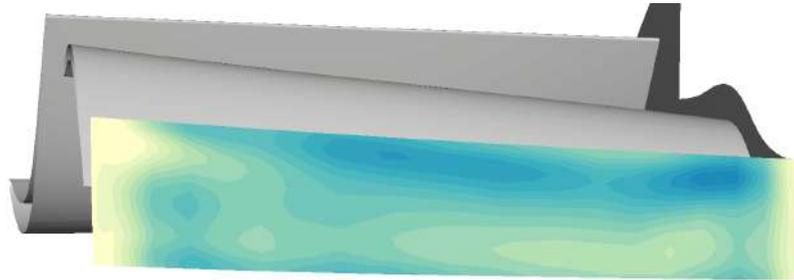


Figure 7.9: Comparison between (normalized) adjoint-based sensitivity of stage and rotor efficiencies to inlet total enthalpy.



(a)  $\mathcal{I} = \eta_{\text{rotor}}$  (stage exit)



(b)  $\mathcal{I} = \eta_{\text{stage}}$  (stage exit)

Figure 7.10: Adjoint-based sensitivities of stator and stage efficiencies ratio to stage exit static pressure,  $d\eta/dp^{\text{exit}}$ .

## 7.1.2 Rotor and Stage Pressure Ratio

In this subsection we look at how both rotor and stage pressure ratios are influenced by the various stage boundary conditions quantities.

Figure 7.11 presents the sensitivities obtained by selecting pressure ratios  $\pi_{\text{rotor}}$  and  $\pi_{\text{stage}}$  as functions of interest in the adjoint system of equations, for the case of inlet total pressure. Both cases (rotor pressure ratio and stage pressure ratio) present mainly negative values of  $d\pi/dp_T^{\text{inlet}}$  across the inlet, with the exception of two regions near the hub and casing. Comparing the two cases, a region of higher (negative) sensitivity close to the hub is observed in the case of  $\pi_{\text{stage}}$  and not on the other. The overall values tend to be higher in magnitude when considering the whole stage than considering only the pressure ratio at the rotor.

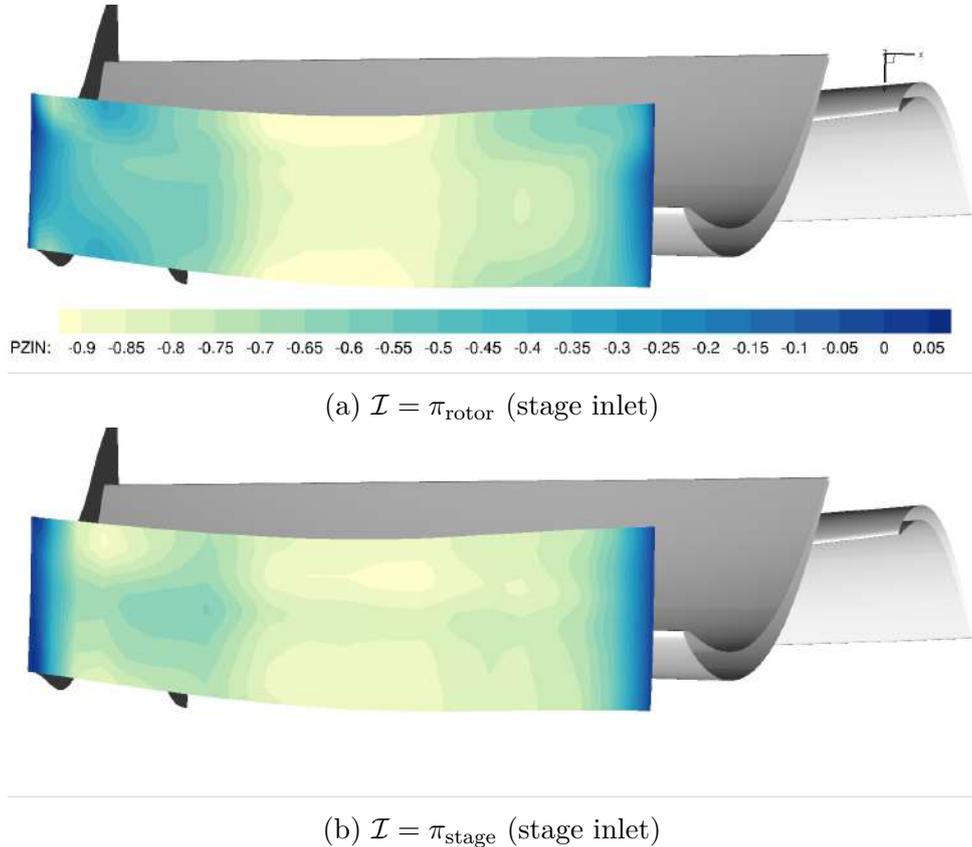


Figure 7.11: Adjoint-based sensitivity of rotor and stage pressure ratios to stage inlet total pressure  $d\pi/dp_T^{\text{inlet}}$  (normalized values).

From figure 7.12 we can observe a reversion of the sign of the impact of tangential velocity at the stage inlet to rotor and stage pressure ratios. While increasing the tangential velocity at most of the inlet will lead to an increase of the pressure ratio of the stator, the

same change in tangential velocity would decrease the pressure ratio across the whole stage. This reversion of sign is made clearer in figure 7.13, where the averaged radial profiles of

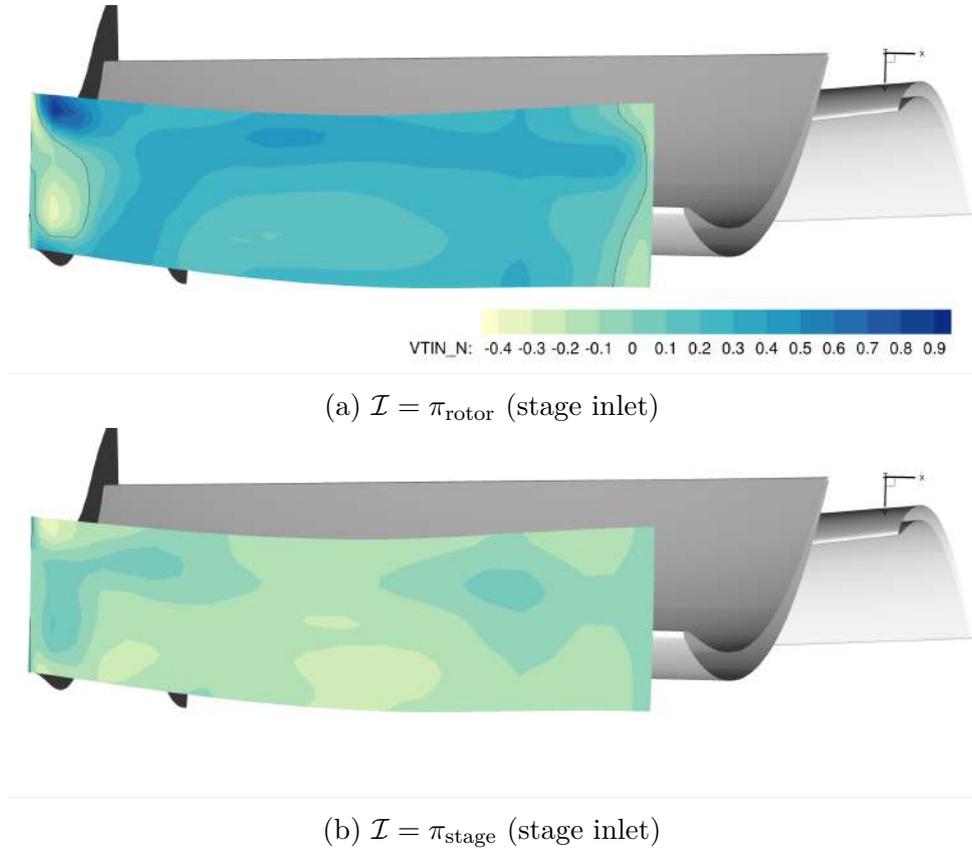


Figure 7.12: Adjoint-based sensitivity of rotor and stage pressure ratios to stage inlet tangential velocity  $d\pi/dV_t^{\text{inlet}}$  (normalized values).

the two quantities are presented. The value of  $d\pi_{\text{stage}}/dV_t^{\text{inlet}}$  remains negative for most of the span, with the exception of a region near the hub.

Looking at the sensitivity of the rotor and stage pressure ratios to stage exit static pressure, presented in figure 7.14, the differences between the two are much less pronounced than for the case of rotor and stage efficiencies. The whole boundary region has positive values of  $d\pi/dp^{\text{exit}}$ , as the exit pressure is in the numerator of the expression of the pressure ratio, and the higher values are midspan between the hub and the casing, aligned with the rotor wake. A possible interpretation of this result is that reducing the blockage effect of the blade would lead to an increase in the pressure ratio.

Both stage and rotor pressure ratios show similar sensitivity to the values of static pressure at the exit of the stage, as seen in figure 7.14, with very small difference both in magnitude and in the shape of the contour plot.

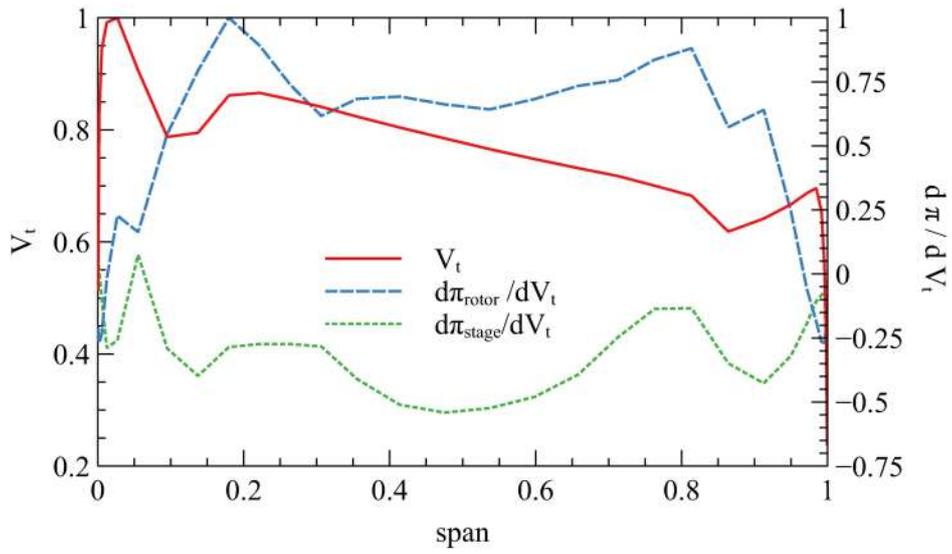
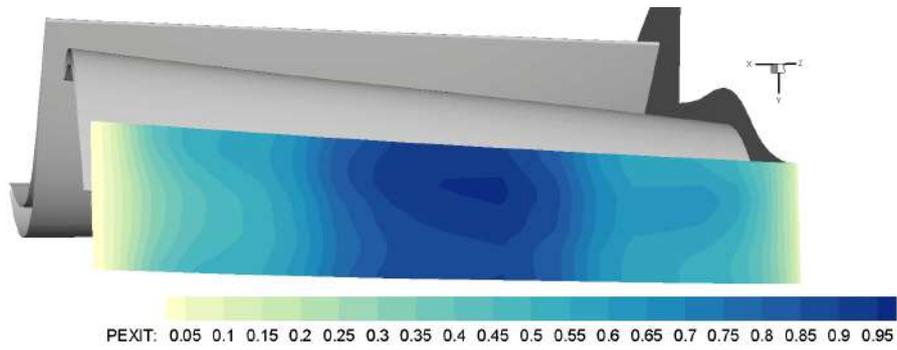
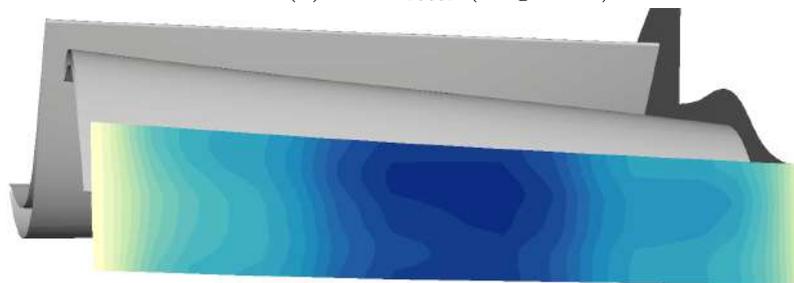


Figure 7.13: Comparison between (normalized) adjoint-based sensitivity of stage and rotor pressure ratios to stage inlet tangential velocity.



(a)  $\mathcal{I} = \pi_{rotor}$  (stage exit)



(b)  $\mathcal{I} = \pi_{stage}$  (stage exit)

Figure 7.14: Adjoint-based sensitivities of stator and stage pressure ratio to stage exit static pressure  $d\pi/dp^{exit}$ .

## 7.2 Sensitivity to Blade Shape and Hub Geometry

This section presents the adjoint-based sensitivity analysis of the previously mentioned performance metrics to the shape of the blades and hub of the low pressure turbine stator/rotor stage in study. The results are first sorted by sensitivity to blade shape and hub geometry and, for each, sorted by performance metric.

### 7.2.1 Blade Shape

The most obvious modification that one can think of performing in a turbomachine to improve its performance is the shape of its blades.

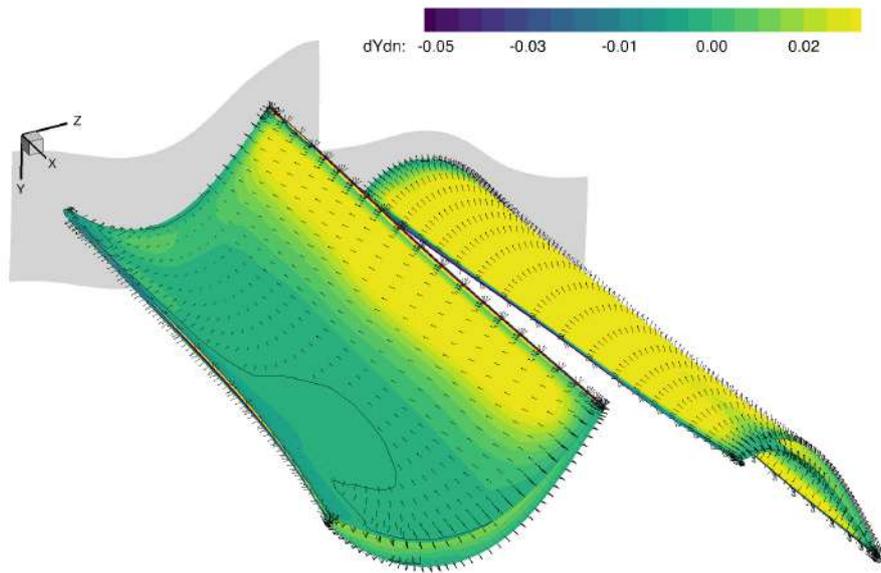
The adjoint-based sensitivity of stage efficiency and stage pressure ratio to the shape of the blade is presented in figures 7.15 and 7.16, respectively. Similarly to the results presented in subsection 6.6.4, the contour of these two figures is the magnitude of the sensitivity vector projected onto the blade surface outer normal, as defined by equation (6.7). The values are normalized by the maximum sensitivity of the two results (efficiency and pressure ratio), in order to allow the comparison of magnitude between the two.

It is clear that the efficiency is much more sensitive to the shape of the blade than the pressure ratio. Although for the stator blade pressure side the derivatives of the two metrics are qualitatively similar, with both metrics increasing with the movement of the surface towards the outer normal, the same is not visible for the stator blade suction side. In the suction side of the blade of the stator, a movement in the outer normal direction, increasing thickness, leads to an increase of the efficiency of the stage. Contrary to this, the sensitivity information for the pressure ratio shows that moving the blade surface inward would lead to an increase of total pressure ratio.

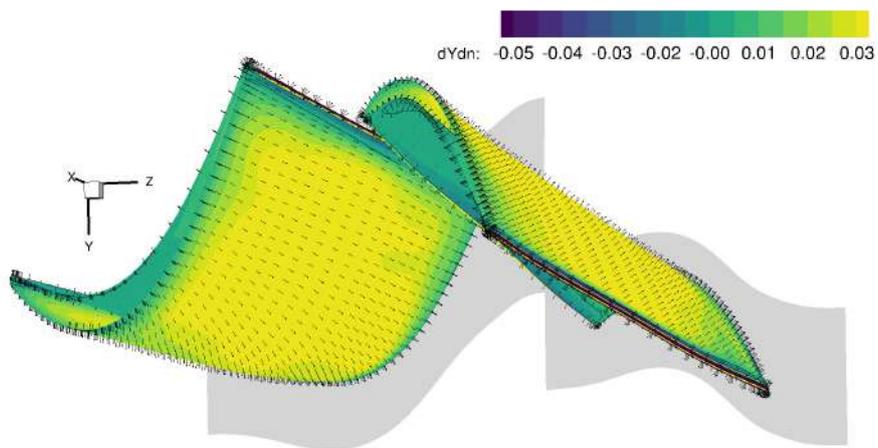
### 7.2.2 Hub Geometry

As previously mentioned, the modification of the hub and/or casing shape, often referred as *endwall contouring*, can significantly impact secondary flows, leading to significant performance improvement of the turbomachine.

The sensitivity of the rotor and stage efficiencies to the radial position of the mesh grid nodes of the hub is presented in figure 7.17. The sensitivity to radial position is obtained from the sensitivity to x- and y-coordinates with the transformation defined by

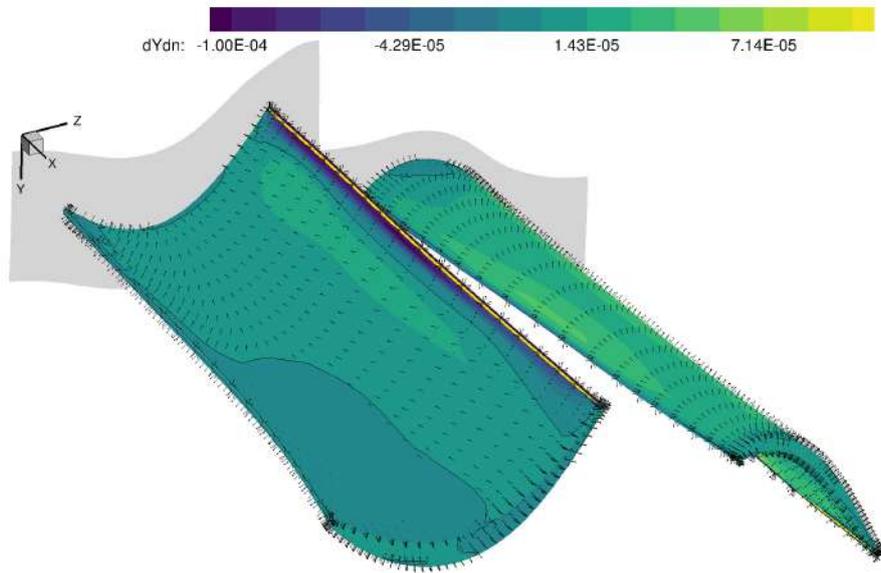


(a) Stator pressure side / Rotor suction side

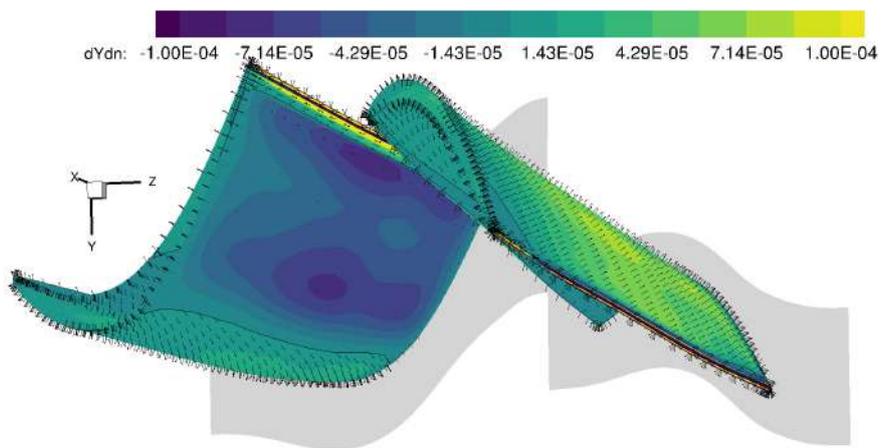


(b) Stator suction side / Rotor pressure side

Figure 7.15: Adjoint-based sensitivity of stage efficiency  $\eta_{\text{stage}}$  to blade shape in normal direction (normalized values).



(a) Stator pressure side / Rotor suction side



(b) Stator suction side / Rotor pressure side

Figure 7.16: Adjoint-based sensitivity of stage pressure ratio  $\pi_{\text{stage}}$  to blade shape in normal direction (normalized values).

equation (6.8), meaning that, for the presented figures, positive values of  $\partial\eta/\partial r_{\text{hub}}$  indicate that moving the surface nodes in the direction of the casing (or the viewer) would increase the efficiency. It should also be noted that the computational domains are repeated axially in these figures to better visualize the sensitivity results.

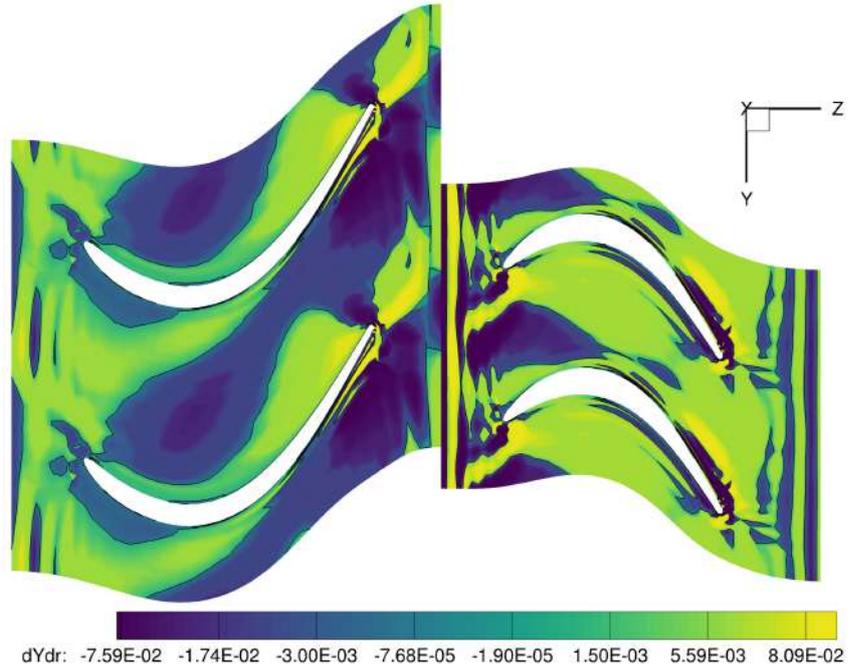
In figure 7.17a it is visible that creating a hump of the hub surface along the stator profile would lead to an increase of the rotor efficiency. Similarly, the negative sensitivity near the trailing edge of the suction side of the stator indicates the need to create a depression by moving the hub surface away from the casing. This particular information is contradicted if we look at the efficiency of the whole stage, in figure 7.17b, where the same area of negative values of  $\partial\eta_{\text{rotor}}/\partial r_{\text{hub}}$  now presents mainly positive values for  $\partial\eta_{\text{stage}}/\partial r_{\text{hub}}$ . The discrepancy between the two sensitivities highlights the importance of taking the coupling between blade rows into account. The sensitivity to the geometry of the hub of the rotor is very similar for the two cases, indicating that a depression (negative bump) between each blade would lead to an increase in efficiency.

Regarding the pressure ratio of the rotor alone or the whole stage, their sensitivities to the hub geometry are presented in figure 7.18. The values in these figures are normalized by the absolute maximum of  $\partial\eta_{\text{rotor}}/\partial r_{\text{hub}}$ ,  $\partial\eta_{\text{stage}}/\partial r_{\text{hub}}$ ,  $\partial\pi_{\text{rotor}}/\partial r_{\text{hub}}$  and  $\partial\pi_{\text{stage}}/\partial r_{\text{hub}}$ , to better visualize the difference in magnitude between the sensitivity to the two types of performance metrics, efficiency and pressure ratio. The difference in magnitude is quite clear between the sensitivities of the two metrics, with the pressure ratio being much less sensitive to changes in the hub surface nodes.

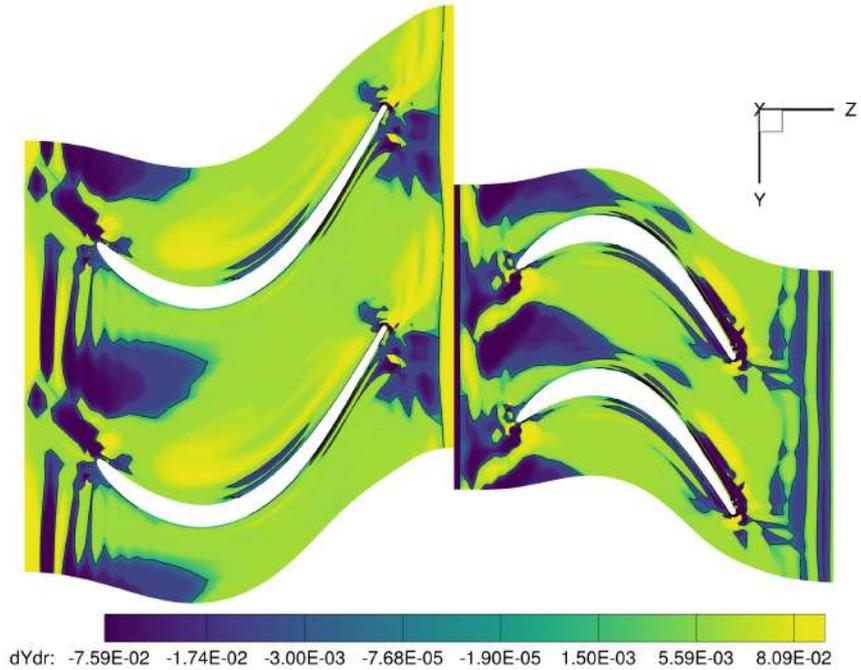
### 7.2.3 Improving Stage Efficiency with Endwall Contouring

Based on the findings presented in the previous subsection, an attempt to improve the stage efficiency by contouring the hub of the rotor through the application of two bumps on its surface is now presented. The locations of the bumps were selected from the analysis of the adjoint-based sensitivities to the hub geometry presented in figure 7.17b, and their height selected using a line search procedure performed manually.

Looking again at figure 7.17b, it shows that the hub region between the blades, at a streamwise location equal to the leading edge of the rotor blade presents high (negative) influence on the stage efficiency. The hub region close to the suction side of the rotor blade also presents relatively high (positive) influence. As such, these two regions were selected

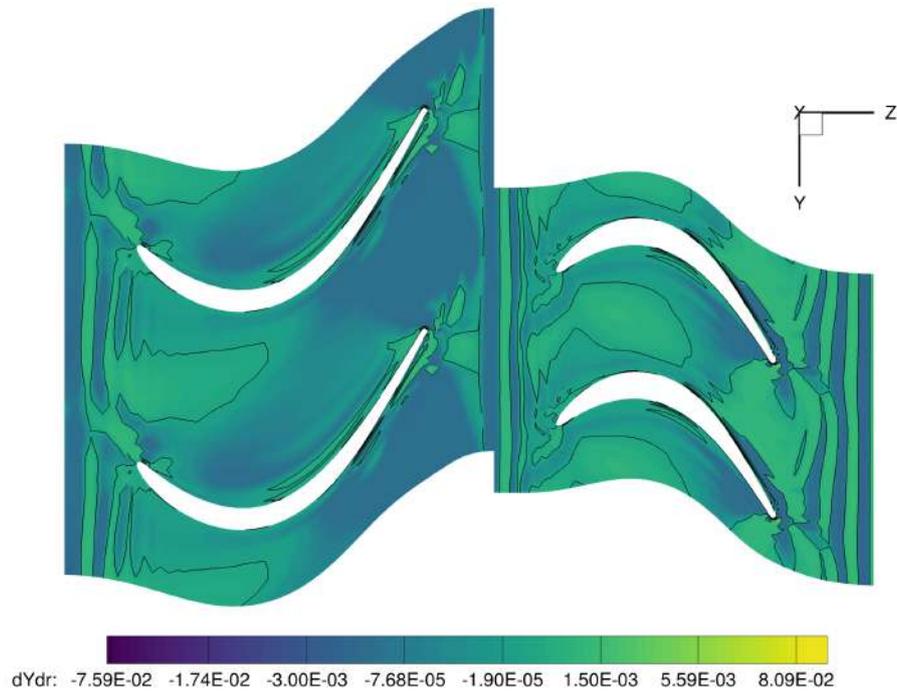


(a)  $\mathcal{I} = \eta_{\text{rotor}}$

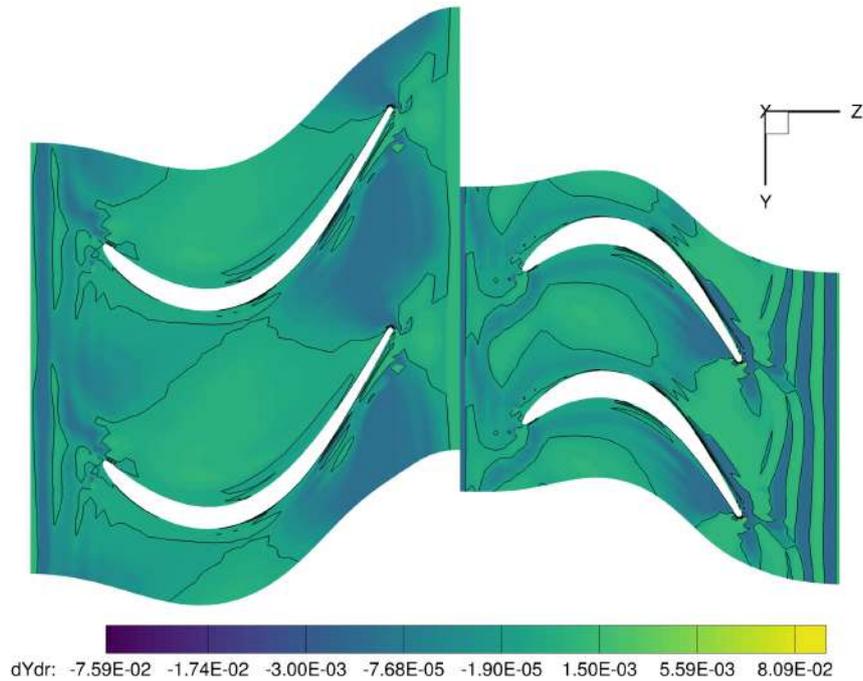


(b)  $\mathcal{I} = \eta_{\text{stage}}$

Figure 7.17: Sensitivity of rotor and stage efficiencies to radial perturbation of hub mesh nodes  $\partial\eta/\partial r_{\text{hub}}$  (normalized values).



(a)  $\mathcal{I} = \pi_{\text{rotor}}$



(b)  $\mathcal{I} = \pi_{\text{stage}}$

Figure 7.18: Sensitivity of rotor and stage pressure ratios to radial perturbation of hub mesh nodes  $\partial\pi/\partial r_{\text{hub}}$  (normalized values).

to impose the bumps, which are represented in figure 7.19. The bumps are imposed on

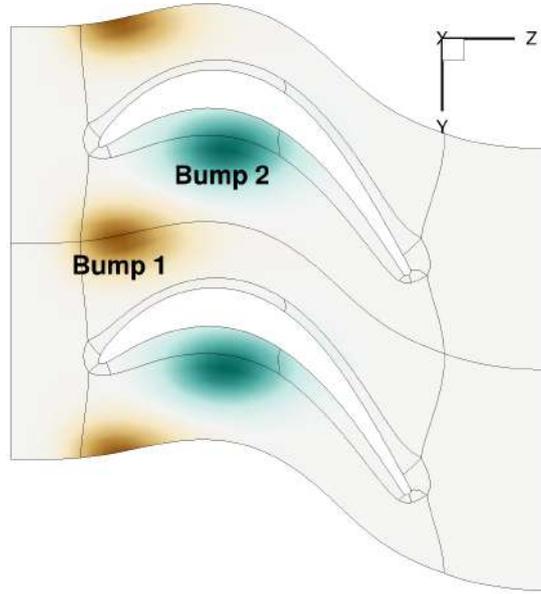


Figure 7.19: Hicks-Henne bumps imposed on the rotor hub.

the hub by perturbing its surface ( $\mathbf{X}_s$ ) in the radial direction by an amount given by the Hicks-Henne bump function [125], defined as

$$\Delta \mathbf{X}_{s,i} = \sum_{j=1}^{N_b} h_j \begin{bmatrix} \sin \left( \pi \hat{x}_{1,i}^{\frac{\log 0.5}{\log t_{c,1,j}}} \right) \Big]^{t_{w,1,j}} \\ \left[ \sin \left( \pi \hat{x}_{2,i}^{\frac{\log 0.5}{\log t_{c,2,j}}} \right) \right]^{t_{w,2,j}} \end{bmatrix}, \quad (7.1)$$

where  $N_b$  is the number of bumps;  $h_j$  is the amplitude (peak height) of bump  $j$ ;  $\hat{x}_{1,j}$  and  $\hat{x}_{2,j}$  are the normalized coordinates of vertex  $i$  in the two directions  $\theta$  and  $z$ , respectively;  $t_{c,1,j}$  and  $t_{c,2,j}$  define the location of the peak of bump  $j$ ; and  $t_{w,1,j}$  and  $t_{w,2,j}$  define how spread the bump is in each location (higher values lead to a less spread bump). The interior mesh nodes  $\mathbf{X}_v$  are modified using an inverse distance weight interpolation scheme [126] as

$$\Delta \mathbf{X}_{v,j} = \frac{\sum_{i=1}^{N_s} \Delta \mathbf{X}_{s,i} / r_{j,i}^p}{\sum_{i=1}^{N_s} 1 / r_{j,i}^p}, \quad (7.2)$$

where  $N_s$  is the number of surface nodes;  $r_{j,i}$  is the distance from the interior node  $j$  to the surface node  $i$  and  $p$  is the power parameter. The sensitivity of the mesh nodes to the Hicks-Henne bumps height parameter  $h_j$  is obtained from a first-order finite difference

approximation, computed using the original and modified grids, as

$$\frac{d\mathbf{X}}{dh_j} = \frac{\mathbf{X}^* - \mathbf{X}}{h_j}, \quad (7.3)$$

where  $\mathbf{X}$  is the original mesh and  $\mathbf{X}^*$  is the modified mesh from the imposition of a bump. This sensitivity is used, together with the adjoint-based sensitivity of the various performance metrics w.r.t. the mesh  $d\mathcal{I}/d\mathbf{X}$ , to compute the total sensitivity of the metrics w.r.t. the height parameters as

$$\frac{d\mathcal{I}}{d\mathbf{h}} = \left( \frac{\partial \mathcal{I}}{\partial \mathbf{X}} - \boldsymbol{\psi}^T \frac{\partial \mathcal{R}}{\partial \mathbf{X}} \right) \frac{d\mathbf{X}}{d\mathbf{h}}. \quad (7.4)$$

The resultant gradients are presented in figure 7.20. These gradients indicate that: i)

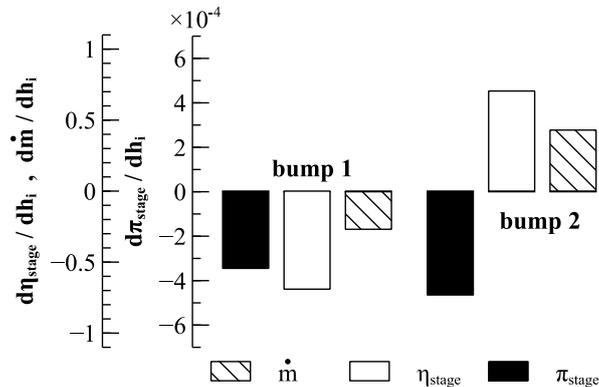


Figure 7.20: Sensitivity of stage performance metrics to bumps height.

introducing a negative bump (inwards) near the leading edge (bump 1) would translate into an increase of the stage efficiency, total pressure ratio and mass flow; ii) a positive displacement (outwards) of bump 2 would translate into an increase of the stage efficiency and mass flow while reducing the stage total pressure ratio.

Should this information be used by a gradient-based numerical optimization algorithm, such as steepest descent, the bump height parameters,  $h_1$  and  $h_2$ , would be perturbed along the search direction according to

$$\mathbf{h} = (h_1, h_2) = \beta \left( \frac{d\eta_{\text{stage}}}{dh_1}, \frac{d\eta_{\text{stage}}}{dh_2} \right), \quad (7.5)$$

where a line search procedure would be performed to find the optimal value of  $\beta$  that maximizes  $\eta_{\text{stage}}$ . The results of a procedure similar to the line search, performed manually,

are presented in figure 7.21, where the variation of  $\dot{m}^{\text{exit}}$ ,  $\eta_{\text{stage}}$  and  $\pi_{\text{stage}}$  relative to the baseline is plotted for different values of  $\beta$ . The behavior of the stage efficiency is as

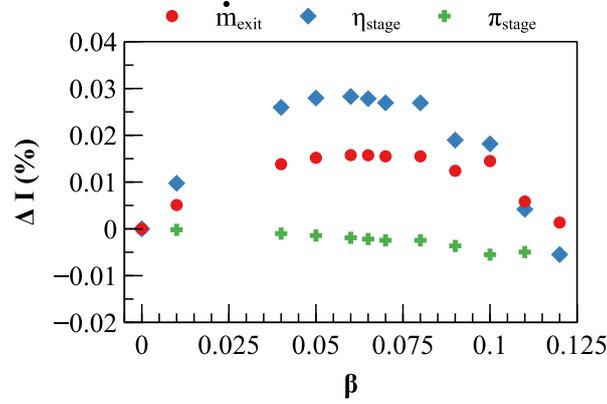


Figure 7.21: Variation of the stage performance metrics with the perturbation step parameter  $\beta$ .

one would expect, increasing with  $\beta$  up to a certain point (between 0.05 and 0.08 in the present case) and then decreasing. The mass flow at the exit of the stage presents a similar variation (in percentage) to the stage efficiency, although with smaller magnitude. On the other hand, the stage total pressure ratio shows a slight decrease as the bumps are perturbed following the search direction given by  $d\eta_{\text{stage}}/d\mathbf{h}$ .

A value of  $\beta = 0.06$  was selected from figure 7.21, which is roughly near the value that maximizes  $\eta_{\text{stage}}$ , that translates into an efficiency improvement of approximately 0.03% and a maximum bump height of approximately 0.8% of the rotor blade span, as shown in figure 7.22. The relatively small expected efficiency improvement can be explained by the already tuned low pressure turbine stage test case we used, but also it might indicate that the number of bumps should be extended to provide a larger design space.

The improvement of the stage efficiency with the application of the "optimal" bumps is the result of a modified flow field. An attempt to identify the main flow features responsible for the stage efficiency improvement is now presented.

Figure 7.23 presents the original and modified pressure fields (normalized values) due to the presence of the two bumps, in an XY plane located at the center of the first bump, as well as the relative difference between the two, computed as

$$\Delta p = \frac{(p)_{\text{bumps}} - (p)_{\text{base}}}{(p)_{\text{base}}} \times 100, \quad (7.6)$$

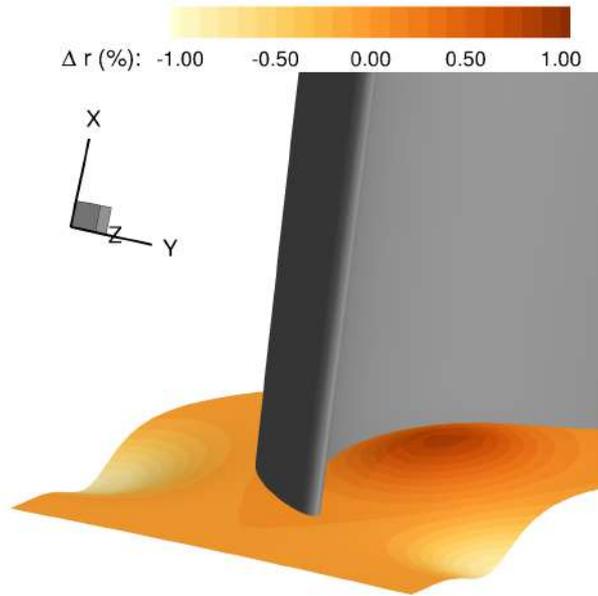


Figure 7.22: Radial displacement imposed on the rotor hub (in % of blade span).

meaning that positive difference values represent an increase from the baseline to the modified pressure flow field. The increase of pressure due to the negative bump (concave) is evident but there is also a reduction in pressure near the suction side of the blade.

Figure 7.24 presents the pressure field in an XY plane located at the center of the second bump. In this case, the presence of the bumps only reduces the pressure in that plane, particularly in the region near suction side of the blade.

Figure 7.25 shows the original radial velocity component as well as its variation due to the presence of the two bumps, in an XY plane located at 85% of the blade chord. Both the velocity and difference values are normalized by the maximum radial velocity in the rotor passage. The two bumps create a deficit in the velocity in the radial direction (figure 7.25b), reducing the mixture of the boundary layer with the flow. This region contains relatively high velocity in the radial direction, as also seen in figure 7.26a that illustrates a set of streamtraces along the rotor blade passage. The secondary flow created at the interface hub-blade is clearly visible in the streamtraces. The effect of the bumps is visible in figure 7.26b, where the streamtraces remain closer to the surface than in the original geometry, reducing its detrimental impact on the performance, thus increasing the efficiency.

The original radial velocity field in a XY plane located behind the rotor blade is presented in figure 7.27a, normalized by the maximum radial velocity in the rotor passage. The region of higher mixture of the boundary layer, highlighted in figure 7.26, is visible

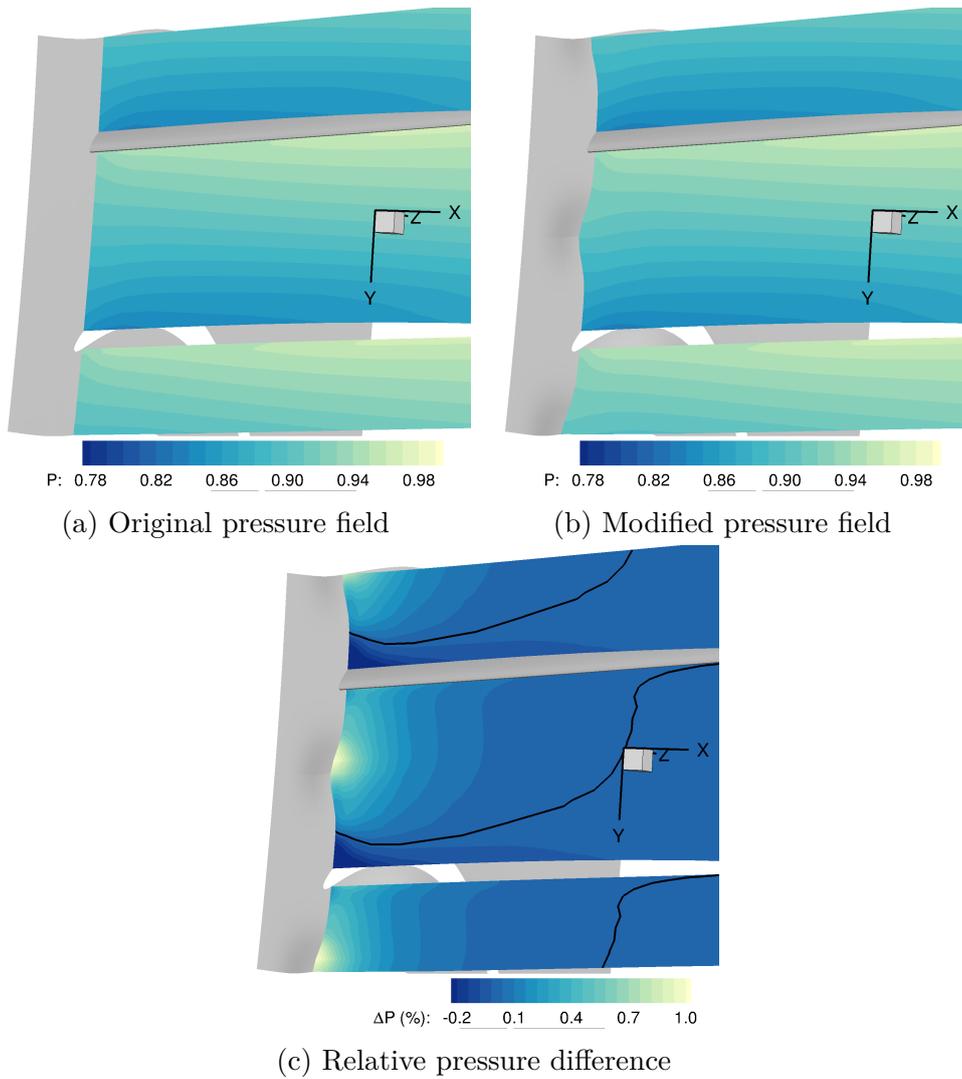


Figure 7.23: Pressure field change due to the two bumps on the rotor hub at an XY plane centered at bump 1.

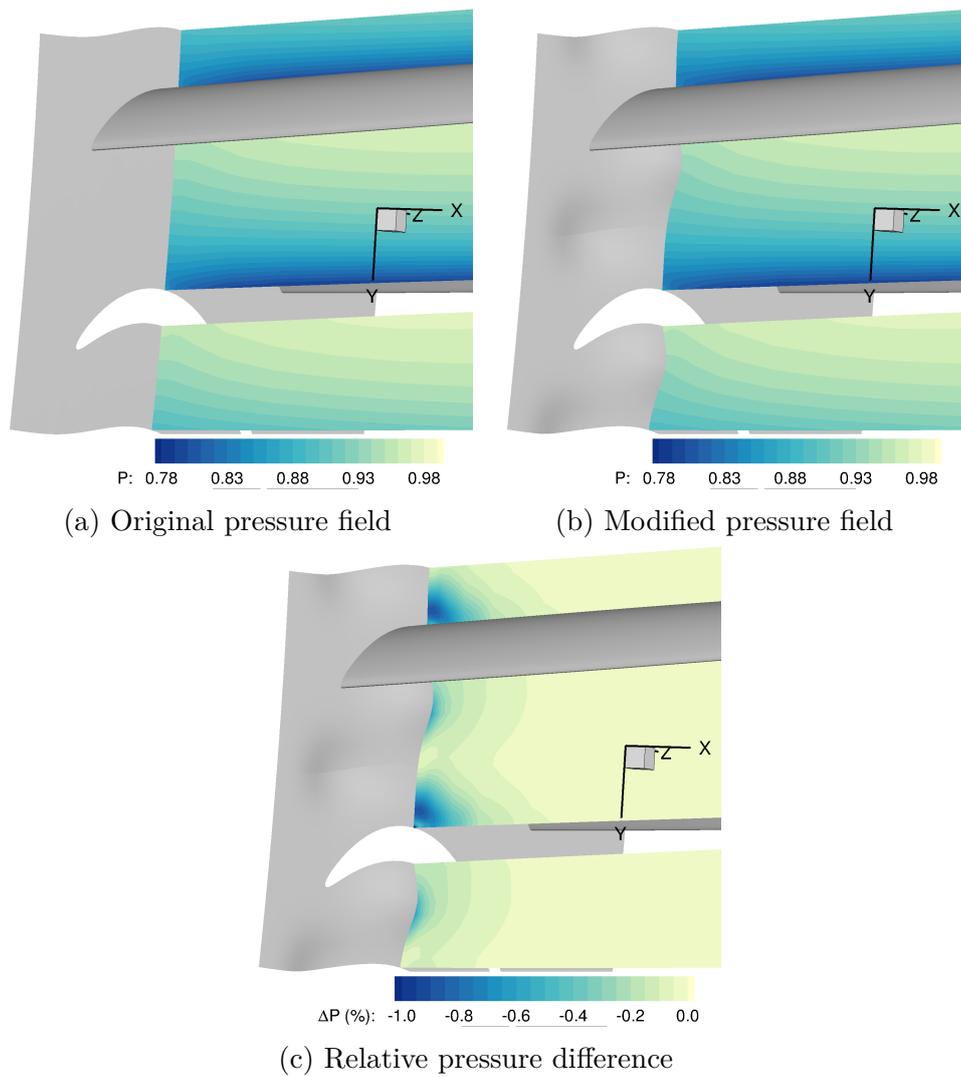


Figure 7.24: Pressure field change due to the two bumps on the rotor hub at an XY plane centered at bump 2.

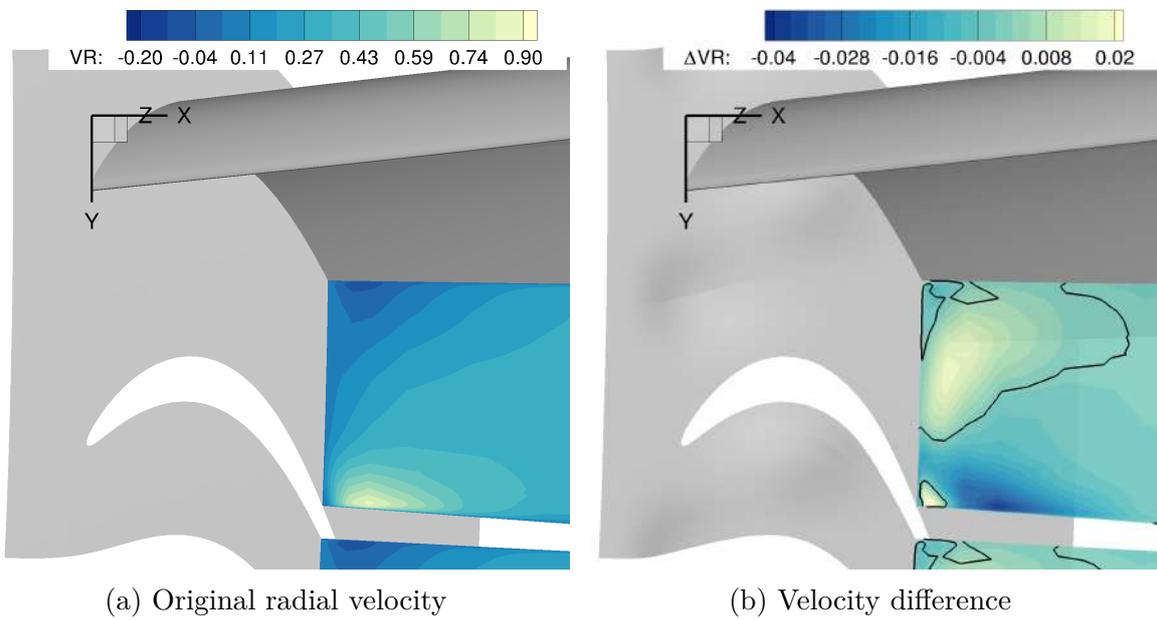


Figure 7.25: Radial velocity change due to the two bumps in the rotor hub at an XY plane at 85% chord (normalized values).

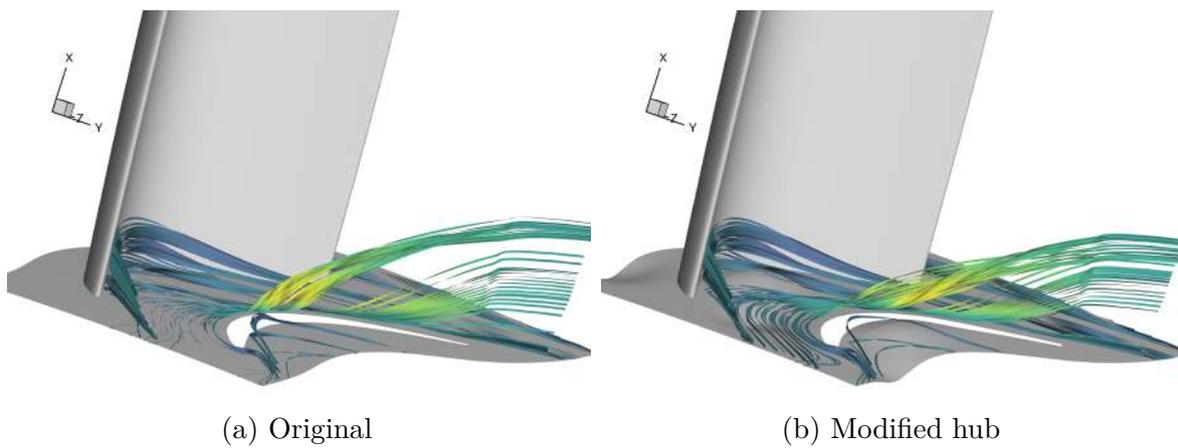


Figure 7.26: Streamtraces along the rotor blade passage.

from the higher values of radial velocity below the trailing edge. The presence of the two

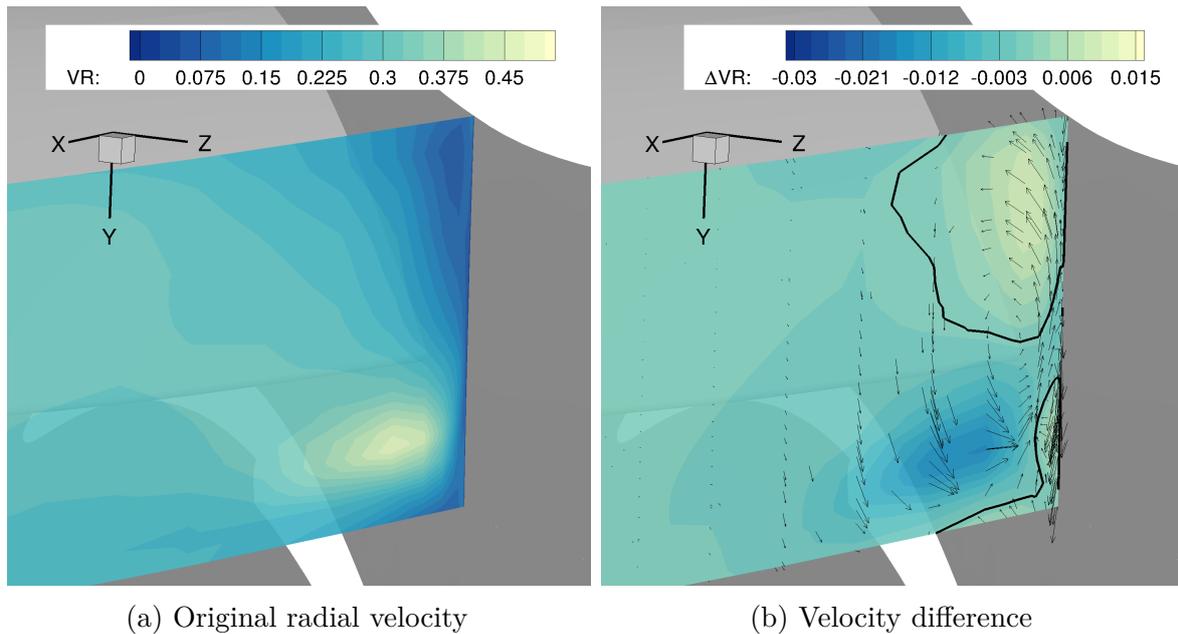


Figure 7.27: Radial velocity change due to the two bumps in the rotor hub at an XY plane behind blade (normalized values).

bumps mitigate this by trying to homogenize the radial velocity profile in the pitchwise direction, which is clear in figure 7.27b, where the variation in the radial velocity due to the bumps is presented, along with the vector field of the variation of velocity.

### 7.3 Thermal Sensitivity Analysis

So far, the sensitivity analyses presented have been relative to aerodynamic metrics of performance. While the designer might want to increase the efficiency or pressure ratio of a stage (or sequence of stages), he might be constrained to what can be achieved due to heat transfer to the blades. This is particularly important for high pressure turbines, where the temperatures can be very large leading to special materials, such as ceramic-matrix composites [127], and cooling of the blades through film or jet cooling [128] must be used to allow the turbine blades to withstand such conditions.

This section presents a series of results of sensitivity analysis of the area averaged total temperature at the surface of the blade of the rotor,  $(T_{Ta})_{rotor}$  to various inlet boundary conditions quantities.

The sensitivity of  $(T_{Ta})_{rotor}$  to stage inlet total pressure  $p_T^{inlet}$  is presented in figure 7.28. From this sensitivity information we can see that, apart from a region very close to the

hub and casing, the main trend is that an increase of inlet total pressure will result in a decrease of  $(T_{Ta})_{\text{rotor}}$ .

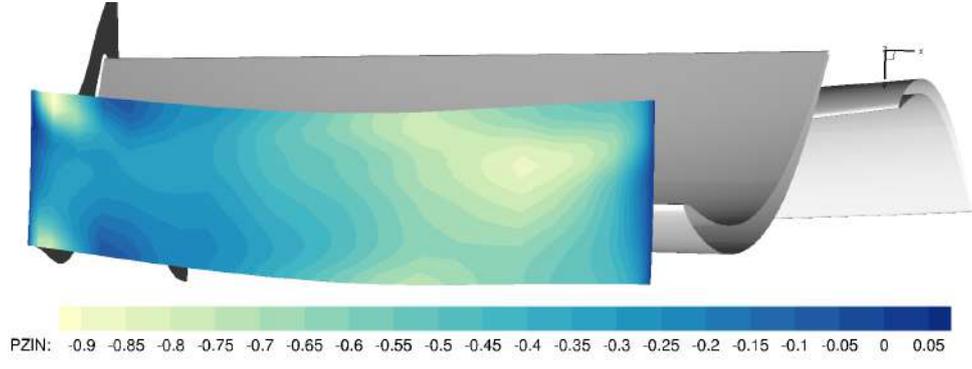


Figure 7.28: Adjoint-based sensitivity of rotor blade vane averaged total temperature  $(T_{Ta})_{\text{rotor}}$  to stage inlet total pressure  $p_T^{\text{inlet}}$  (normalized values).

Figure 7.29 presents the sensitivity of  $(T_{Ta})_{\text{rotor}}$  to stage inlet total enthalpy  $h_T^{\text{inlet}}$ . In this case, we have only positive sensitivities, which is expected, since enthalpy is directly related to temperature of the flow. Larger sensitivities are observed far away from the hub and casing and in the region between blades. Recalling figure 7.6, where the sensitivity of rotor and stage efficiencies to stage inlet total enthalpy is presented, we can infer that, as  $d\eta_{\text{rotor}}/dh_T^{\text{inlet}}$  and  $d\eta_{\text{stage}}/dh_T^{\text{inlet}}$  are mainly negative across the inlet, an increase in efficiency by tweaking the temperature of the flow would lead to a decrease in the average temperature of the blade of the rotor as well. Again, an expected result since efficiency is a measure of how close the expansion of the flow is to an isentropic process.

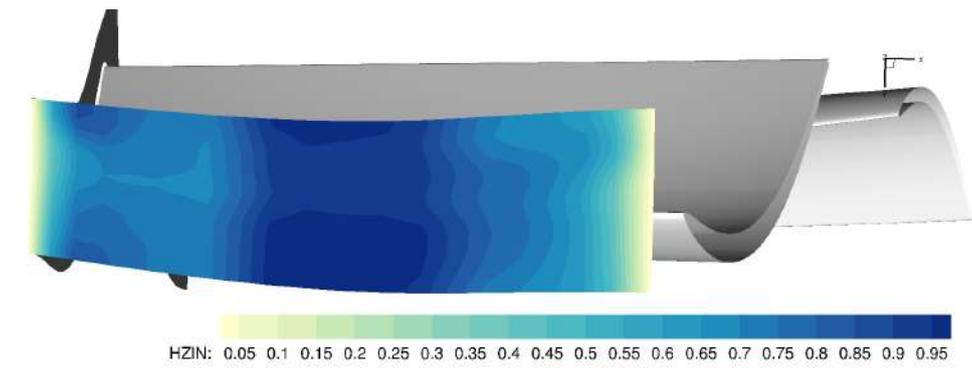


Figure 7.29: Adjoint-based sensitivity of rotor blade vane averaged total temperature  $(T_{Ta})_{\text{rotor}}$  to stage inlet total enthalpy  $h_T^{\text{inlet}}$  boundary condition (normalized values).

The last case presented in this section regarding boundary conditions is the sensitivity of  $(T_{Ta})_{\text{rotor}}$  to stage inlet tangential velocity  $V_t^{\text{inlet}}$ , shown in figure 7.30. In this case, we see two regions of negative and positive sensitivities close to the hub, a large region of

negative sensitivities midspan of the blade, and a region of positive sensitivities closer to the casing, in the region between blades. These sensitivities are, in a large region of the inlet, opposite to the sensitivities  $d\pi/dV_t^{\text{inlet}}$  presented in figure 7.12, indicating that, as in the case for stage inlet total enthalpy, an attempt to decrease the pressure ratio of the stage by tweaking the tangential velocity at the inlet might translate into an increase in the averaged total temperature at the rotor blade surface. These results

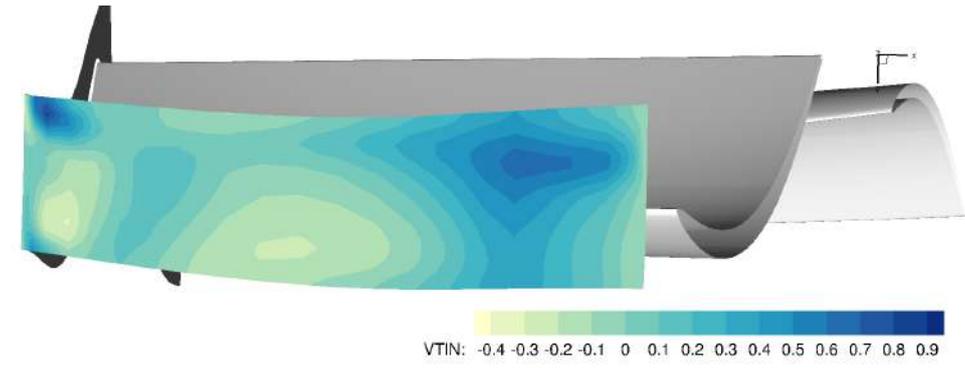


Figure 7.30: Adjoint-based sensitivity of rotor blade vane averaged total temperature  $(T_{Ta})_{\text{rotor}}$  to stage inlet tangential velocity  $V_t^{\text{inlet}}$  boundary condition (normalized values).

highlight the importance of assessing not only aerodynamic metrics or performance, but also aero-thermal metrics in the design or optimization of a turbomachine component.

## 7.4 Final Remarks

This chapter presented an application of the adjoint solver to the sensitivity analyses of a stator-rotor stage of a low pressure turbine. This was possible due to the adjoint mixing-plane whose formulation and implementation was the core of this work. The various sensitivity results presented in this chapter highlighted the coupling between the rows (stator and rotor) and served as an example of how the sensitivity analysis could be used by a designer to obtain insight on the influence of various parameters on certain performance metrics.

The results were also computed with a computational cost many orders of magnitude lower than that required by traditional methods such as finite-differences. The two row computational domain contained 540 cells at the inlet of the stage, and 570 at its exit. In terms of surface mesh grid nodes, each domain contained approximately 4,000 mesh grid nodes on the blade surface and 1,000 on the surface of the hub. Had the sensitivities

exhibited in this chapter been computed with finite-differences, at least more than 30,000 flow re-evaluations would have been required, which for the case of study at hand would have translated to more than a year of CPU time, as detailed in table 7.2. This is a conservative estimate given the assumption that an optimal step perturbation size was known *à priori*.

	Stator	Rotor	$N_\alpha$	Flow evaluations	CPU Time
Inlet	540	-	5	2,700	
Exit	-	570	1	570	
Hub	1,000	1,000	3	6,000	
Blade	4,000	4,000	3	24,000	
Time = 20 min ×				33,270	≈ 462 days

Table 7.2: Hypothetical time requirements for equivalent of adjoint-based sensitivity analysis with finite-differences.

# Chapter 8

## Conclusions

The work presented in this document comes from a growing necessity for the turbomachinery industry to have higher fidelity tools, particularly of efficient sensitivity analysis tools, which are of the utmost importance as numerical optimization is nowadays a desired tool for extracting the highest performance of already very highly tuned components. Turbomachines are typically composed of many stages of compressors and/or turbine stator-rotor rows to achieve higher ratios of compression and/or expansion and, as such, the consideration of the interaction between the various rows is absolutely required for the improvement of their performance. Therefore, the use of multi-row coupling in the numerical simulation of turbomachinery is vital for the optimization of these very complex machines, as considering single individual components becomes a limitation, albeit providing valuable information of certain phenomena.

An adjoint solver had been previously developed for the sensitivity analysis of turbomachinery, based on a legacy CFD solver, using a hybrid approach called ADjoint which merged the discrete adjoint method with the use of Automatic Differentiation (AD) tools to obtain routines for the computation of the partial derivatives required to assemble the adjoint system of equations and the adjoint-based sensitivity formula.

The adjoint method is an efficient and accurate approach for the estimation of sensitivity information when the problem in analysis involves a large number of design variables, which far exceed the number of functions of interest. Even when the governing equations are computationally cheap and allow finite-difference approximations, these last introduce numerical uncertainty in the sense that they may be very sensitive to the perturbation step.

The discrete adjoint approach provides a well defined procedure to derive the adjoint equations which is nearly independent of the complexity of the governing equations. The treatment of arbitrary functions of interest with the discrete adjoint approach, when compared to the continuous approach, is much simpler as it is not limited to specific form of integral functions. The gradients obtained with the discrete approach are also consistent with the flow solver, which is highly beneficial for numerical optimization.

However, this legacy adjoint solver lacked the capacity of handling multi-row simulations and, therefore, it could not provide the so important information of the coupling between the various rows of the turbomachine components. Implementing the capability of handling such simulations on the adjoint was therefore extremely important.

The formulation of the adjoint counterpart of the mixing-plane of the CFD solver was developed by first dividing the mixing-plane algorithm in various steps and applying the chain rule to the derivative of each step, thus obtaining the linearization of the complete procedure.

The implementation of the theoretical formulation into the single-row adjoint solver was performed using the same hybrid approach used to develop the legacy adjoint solver, by rewriting the original routines corresponding to each of the steps considered in the formulation of the problem and applying an AD tool to obtain differentiated routines. The use of AD tools allowed the linearization of the mixing-plane algorithm while maintaining the consistency with the original implementation in the direct flow solver, but required special treatment of certain features of the original code, such as dynamic allocatable arrays inside of custom data structures, or MPI communications, which could not be correctly differentiated using the AD tool. These differentiated routines were then hand-assembled to obtain complete differentiation of the mixing-plane procedure. This hand-assembly came as a requirement to maintain consistency with the original flow solver.

A numerical verification of the implementation was performed and some selected results were presented for a test case comprising of an axial turbine stator-rotor stage. Various functions of interest were considered and their sensitivity to inlet boundary conditions or blade and hub geometries were presented and the adjoint-based sensitivities compared with first-order finite-difference approximations. The relative differences between the adjoint-based and FD-based methods revealed very good agreement of both methods, with differences below 1% for the majority of the presented cases, which attested both the

correct formulation and implementation of the multi-row capability.

The verification of the implementation also highlighted the benefits of using an adjoint solver for the computation of sensitivities versus the traditional finite-difference method. While the adjoint method required one additional solver run for each function of interest, with a computational cost similar to flow direct run in terms of CPU time, the finite-difference approach required many direct solver runs to obtain a converged value, just to obtain a small fraction of the results given by the adjoint solver.

A more in depth analysis of the sensitivity results of the test case was presented in the final chapter, with a selection of cases that intended to show not only the plethora of information that the adjoint solver is able to provide but also to highlight the physical coupling between the two rows, and thus, the importance of coupled multi-row analysis and design of turbomachinery. A prediction of the time necessary to obtain the same sensitivity information given by the adjoint solver with finite-differences indicated that more than one year of CPU time would have been necessary to evaluate all the necessary variable perturbations.

By using the adjoint solver with handling of the multistage interface between adjacent rows, it is possible to efficiently and accurately quantify the impact of: i) boundary inlet conditions to downstream blade rows performance; ii) boundary exit conditions to upstream rows performance; iii) upstream blade or hub/casing shapes to downstream row performance; and iv) downstream blade or hub/casing shapes to upstream row performance.

The proposed inclusion of adjoint multistage handling in a gradient-based multistage turbomachinery design framework is thus paramount to achieve the best overall results, both in terms of computational cost but mainly in terms of optimal design outcome.

## **8.1 Achievements**

The main goal of the present work was to implement the capacity to handle multi-row turbomachinery problems into a legacy adjoint solver. A formulation for the differentiation of the mixing-plane algorithm was presented and its correctness proved by the verification of its numerical implementation. From the close agreement with the sensitivity information obtained with the improved adjoint solver with finite-difference approximations, the main

goal can be considered achieved.

Not only can the improved adjoint solver handle multi-row problems to single row metrics of performance, but the computation of new multi-row performance metrics was also implemented, namely the multi-stage efficiency and total pressure ratio.

The formulation here presented can guide future users in the developing of adjoint counterparts of similar features in numerical solvers.

With the finalization of the work presented in this document, the adjoint solver is now capable of handling steady simulations of arbitrary number of blade rows, thus providing the designer a very large set of information.

## 8.2 Future Work

With the adjoint mixing-plane interface implemented, the clearer path to future work is its application in the sensitivity analysis and design of better performing multi-row turbomachinery components, through the incorporation of the sensitivity information given by the adjoint solver into a numerical gradient-based optimization framework.

The application of the solver to larger problems comprising various blade rows would surely provide very interesting information, and serve as another proof of capability of the implemented adjoint-solver. This will come naturally as the developed tool is made available to the sponsor jet engine manufacturer.

Finally, as the available computing power is growing, unsteady multi-stage simulations are starting to be performed in the analysis and design of turbomachinery to better replicate the flow physics. As such, extension of the adjoint solver to handle unsteady simulations should be pursued, either by using explicit time integration or harmonic balance techniques.

# Bibliography

- [1] IATA *IATA 20-Year Air Passenger Forecast*. 2017.
- [2] Comission, E., ed. *Flightpath 2050: Europe's Vision for Aviation ; Maintaining Global Leadership and Serving Society's Needs ; Report of the High-Level Group on Aviation Research*. Policy / European Commission. OCLC: 930887434. Publ. Off. of the Europ. Union, Luxembourg, 2011. 25 pp. ISBN: 978-92-79-19724-6.
- [3] IATA *Economic Performance of the Airline Industry*. 2018.
- [4] Chao, C.-C. Assessment of Carbon Emission Costs for Air Cargo Transportation. *Transportation Research Part D: Transport and Environment* 33 (Dec. 1, 2014), 186–195. ISSN: 1361-9209. DOI: [10/f6rrbm](https://doi.org/10/f6rrbm).
- [5] EUturbines *Roadmap on Turbomachinery Research*. EUturbines, 2014.
- [6] Škorpík, J. *Introduction to Turbomachinery*. Aug. 2009.
- [7] Moran, M. J., Shapiro, H. N., Boettner, D. D., and Bailey, M. B. *Fundamentals of Engineering Thermodynamics*. John Wiley & Sons, Dec. 7, 2010. 1025 pp. ISBN: 978-0-470-49590-2.
- [8] *Brayton Cycle*. Page Version ID: 850911824. URL: [https://en.wikipedia.org/w/index.php?title=Brayton\\_cycle&oldid=850911824](https://en.wikipedia.org/w/index.php?title=Brayton_cycle&oldid=850911824).
- [9] *Turbofan*. Page Version ID: 861971077. URL: <https://en.wikipedia.org/w/index.php?title=Turbofan&oldid=861971077>.
- [10] Beselt, C., Eck, M., and Peitsch, D. Three-Dimensional Flow Field in Highly Loaded Compressor Cascade. *Journal of Turbomachinery* 136, 10 (Aug. 8, 2014), 101007. ISSN: 0889-504X. DOI: [10.1115/1.4028083](https://doi.org/10.1115/1.4028083).

- [11] Schneider, C. M., Schrack, D., Kuerner, M., Rose, M. G., Staudacher, S., Guendogdu, Y., and Freygang, U. On the Unsteady Formation of Secondary Flow Inside a Rotating Turbine Blade Passage. *Journal of Turbomachinery* 136, 6 (Nov. 8, 2013), 061004. ISSN: 0889-504X. DOI: [10.1115/1.4025582](https://doi.org/10.1115/1.4025582).
- [12] Li, Z., and Zheng, X. Review of Design Optimization Methods for Turbomachinery Aerodynamics. *Progress in Aerospace Sciences* 93 (Aug. 2017), 1–23. ISSN: 03760421. DOI: [10.1016/j.paerosci.2017.05.003](https://doi.org/10.1016/j.paerosci.2017.05.003).
- [13] Nocedal, J., and Wright, S. J. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, 1999. 636 pp. ISBN: 978-0-387-98793-4.
- [14] Melanie, M. An Introduction to Genetic Algorithms (1999), 162.
- [15] Dorigo, M., and Gambardella, L. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation* 1, 1 (Apr. 1997), 53–66. ISSN: 1089778X. DOI: [10.1109/4235.585892](https://doi.org/10.1109/4235.585892).
- [16] Storn, R. Differential Evolution - A Simple and EFFICIENT Heuristic for Global Optimization over Continuous Spaces. *DIFFERENTIAL EVOLUTION* (1997), 19.
- [17] Kennedy, J., and Eberhart, R. Particle Swarm Optimization. In: *Proceedings of ICNN 95 - International Conference on Neural Networks*. Proceedings of ICNN 95 - International Conference on Neural Networks. Vol. 4. Nov. 1995, 1942–1948 vol.4. DOI: [10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).
- [18] Murty, K. *Linear Programming*. John Wiley & Sons, 1983.
- [19] Kiefer, J. Sequential Minimax Search for a Maximum. *Proceedings of the American Mathematical Society* 4, 3 (June 1953), 502. ISSN: 00029939. DOI: [10.2307/2032161](https://doi.org/10.2307/2032161).
- [20] Hooke, R., and Jeeves, T. A. "Direct Search" Solution of Numerical and Statistical Problems. *Journal of the ACM* 8, 2 (Apr. 1, 1961), 212–229. ISSN: 00045411. DOI: [10.1145/321062.321069](https://doi.org/10.1145/321062.321069).
- [21] Cauchy, M. A. Méthode Générale Pour La Résolution Des Systèmes d'équations Simultanées. *Compte Rendu des Séances de L'Académie des Sciences XXV* (1847), 3.

- [22] Fletcher, R., and Reeves, C. M. Function Minimization by Conjugate Gradients. *The Computer Journal* 7, 2 (Feb. 1, 1964), 149–154. ISSN: 0010-4620, 1460-2067. DOI: [10.1093/comjnl/7.2.149](https://doi.org/10.1093/comjnl/7.2.149).
- [23] Davidon, W. C. *Variable Metric Method for Minimization*. Technical Report Report ANL-5990. AEC Research and Development, 1959, 17.
- [24] Fletcher, R., and Powell, M. J. D. A Rapidly Convergent Descent Method for Minimization. *The Computer Journal* 6, 2 (Aug. 1, 1963), 163–168. ISSN: 0010-4620, 1460-2067. DOI: [10.1093/comjnl/6.2.163](https://doi.org/10.1093/comjnl/6.2.163).
- [25] Zoutendijk, G. *Methods of Feasible Directions;: A Study in Linear and Nonlinear Programming*. Elsevier Pub. Co, 1960. 126 pp.
- [26] Wolfe, P. Methods of Nonlinear Programming. In: *Recent Advances in Mathematical Programming*. New York, 1967.
- [27] Spivak, M. *Calculus*. Cambridge University Press, June 8, 2006. 683 pp. ISBN: 978-0-521-86744-3.
- [28] Squire, W., and Trapp, G. Using Complex Variables to Estimate Derivatives of Real Functions. *SIAM Review* 40, 1 (Jan. 1998), 110–112. ISSN: 0036-1445, 1095-7200. DOI: [10.1137/S003614459631241X](https://doi.org/10.1137/S003614459631241X).
- [29] Anderson, W. K., Newman, J. C., Whitfield, D. L., and Nielsen, E. J. Sensitivity Analysis for Navier-Stokes Equations on Unstructured Meshes Using Complex Variables. *AIAA Journal* 39, 1 (Jan. 2001), 56–63. ISSN: 0001-1452, 1533-385X. DOI: [10.2514/2.1270](https://doi.org/10.2514/2.1270).
- [30] Martins, J. R. R. A., Sturdza, P., and Alonso, J. J. The Complex-Step Derivative Approximation. *ACM Transactions on Mathematical Software* 29, 3 (Sept. 1, 2003), 245–262. ISSN: 00983500. DOI: [10.1145/838250.838251](https://doi.org/10.1145/838250.838251).
- [31] Lions, J. L. *Optimal Control of Systems Governed by Partial Differential Equations*. Grundlehren Der Mathematischen Wissenschaften. Springer-Verlag, Berlin Heidelberg, 1971. ISBN: 978-3-642-65026-0.
- [32] Mader, C. A. ADjoint: An Approach for the Rapid Development of Discrete Adjoint Solvers. University of Toronto, 2007.

- [33] Marta, A. C., Mader, C. A., Martins, J. R. R. A., Van der Weide, E., and Alonso, J. J. A Methodology for the Development of Discrete Adjoint Solvers Using Automatic Differentiation Tools. *International Journal of Computational Fluid Dynamics* 21, 9-10 (Oct. 2007), 307–327. ISSN: 1061-8562, 1029-0257. DOI: [10.1080/10618560701678647](https://doi.org/10.1080/10618560701678647).
- [34] McNamara, A., Treuille, A., Popović, Z., and Stam, J. Fluid Control Using the Adjoint Method. In: *ACM SIGGRAPH 2004 Papers*. SIGGRAPH '04. ACM, New York, NY, USA, 2004, 449–456. DOI: [10.1145/1186562.1015744](https://doi.org/10.1145/1186562.1015744).
- [35] Pironneau, O. On Optimum Design in Fluid Mechanics. *Journal of Fluid Mechanics* 64, 01 (1974), 97–110. DOI: [10.1017/S0022112074002023](https://doi.org/10.1017/S0022112074002023).
- [36] Jameson, A. Aerodynamic Design via Control Theory. *Journal of Scientific Computing* 3, 3 (1988), 233–260.
- [37] Jameson, A., Martinelli, L., and Pierce, N. A. Optimum Aerodynamic Design Using the Navier–Stokes Equations. *Theoretical and Computational Fluid Dynamics* 10, 1-4 (1998), 213–237.
- [38] Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. J., and Saunders, D. Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 1. *Journal of Aircraft* 36, 1 (1999), 51–60.
- [39] Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. J., and Saunders, D. Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 2. *Journal of Aircraft* 36, 1 (1999), 61–74.
- [40] Martins, J. R., Alonso, J. J., and Reuther, J. J. High-Fidelity Aerostructural Design Optimization of a Supersonic Business Jet. *Journal of Aircraft* 41, 3 (2004), 523–530. ISSN: 0021-8669. DOI: [10.2514/1.11478](https://doi.org/10.2514/1.11478).
- [41] Kenway, G., Kennedy, G., and Martins, J. Aerostructural Optimization of the Common Research Model Configuration. In: American Institute of Aeronautics and Astronautics, June 16, 2014. ISBN: 978-1-62410-283-7. DOI: [10.2514/6.2014-3274](https://doi.org/10.2514/6.2014-3274).

- [42] Marta, A. C., and Alonso, J. J. Toward Optimally Seeded Airflow on Hypersonic Vehicles Using Control Theory. *Computers & Fluids* 39, 9 (Oct. 2010), 1562–1574. ISSN: 0045-7930. DOI: [10.1016/j.compfluid.2010.05.009](https://doi.org/10.1016/j.compfluid.2010.05.009).
- [43] Marta, A. C., and Shankaran, S. Exploring the Use of Adjoint Methods for Detailed Sensitivity Analysis on Turbomachinery. In: International Conference on Engineering Optimization. Rio de Janeiro, Brasil, 2012.
- [44] Koehler, C., Durscher, R., Beran, P., and Bhagat, N. Adjoint-Enhanced Flow Visualization. *Journal of Visualization* (Apr. 13, 2018). ISSN: 1343-8875, 1875-8975. DOI: [10.1007/s12650-018-0490-6](https://doi.org/10.1007/s12650-018-0490-6).
- [45] Yang, S., and Liu, F. Optimum Aerodynamic Design of Cascades by Using an Adjoint Equation Method. In: *41st Aerospace Sciences Meeting and Exhibit*. 41st Aerospace Sciences Meeting and Exhibit. American Institute of Aeronautics and Astronautics, Reno, Nevada, Jan. 6, 2003. ISBN: 978-1-62410-099-4. DOI: [10.2514/6.2003-1068](https://doi.org/10.2514/6.2003-1068).
- [46] Chung, J., Lee, K., and Martin, G. Aerodynamic Design of 3D Compressor Blade Using an Adjoint Method. In: *42nd AIAA Aerospace Sciences Meeting and Exhibit*. 42nd AIAA Aerospace Sciences Meeting and Exhibit. American Institute of Aeronautics and Astronautics, Reno, Nevada, Jan. 5, 2004. ISBN: 978-1-62410-078-9. DOI: [10.2514/6.2004-27](https://doi.org/10.2514/6.2004-27).
- [47] Wu, H.-Y., Liu, F., and Tsai, H.-M. Aerodynamic Design of Turbine Blades Using an Adjoint Equation Method. In: *43rd AIAA Aerospace Sciences Meeting and Exhibit*. Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics, Jan. 10, 2005. DOI: [10.2514/6.2005-1006](https://doi.org/10.2514/6.2005-1006).
- [48] Arens, K., Rentrop, P., Stoll, S., and Wever, U. An Adjoint Approach to Optimal Design of Turbine Blades. *Applied Numerical Mathematics* 53, 2-4 (May 2005), 93–105. ISSN: 01689274. DOI: [10.1016/j.apnum.2004.11.003](https://doi.org/10.1016/j.apnum.2004.11.003).
- [49] Li, Y., Yang, D., and Feng, Z. Inverse Problem in Aerodynamic Shape Design of Turbomachinery Blades. In: *Volume 6: Turbomachinery, Parts A and B*. ASME Turbo Expo 2006: Power for Land, Sea, and Air. Vol. 2006. ASME, Barcelona, Spain, 2006, 1535–1543. ISBN: 978-0-7918-4241-6. DOI: [10.1115/GT2006-91135](https://doi.org/10.1115/GT2006-91135).

- [50] Papadimitriou, D., and Giannakoglou, K. A Continuous Adjoint Method for the Minimization of Losses in Cascade Viscous Flows. In: *44th AIAA Aerospace Sciences Meeting and Exhibit*. 44th AIAA Aerospace Sciences Meeting and Exhibit. American Institute of Aeronautics and Astronautics, Reno, Nevada, Jan. 9, 2006. ISBN: 978-1-62410-039-0. DOI: [10.2514/6.2006-49](https://doi.org/10.2514/6.2006-49).
- [51] Papadimitriou, D., and Giannakoglou, K. A Continuous Adjoint Method with Objective Function Derivatives Based on Boundary Integrals, for Inviscid and Viscous Flows. *Computers & Fluids* 36, 2 (Feb. 2007), 325–341. ISSN: 00457930. DOI: [10.1016/j.compfluid.2005.11.006](https://doi.org/10.1016/j.compfluid.2005.11.006).
- [52] Papadimitriou, D. I., and Giannakoglou, K. C. Total Pressure Loss Minimization in Turbomachinery Cascades Using a New Continuous Adjoint Formulation. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy* 221, 6 (Jan. 2007), 865–872. ISSN: 0957-6509, 2041-2967. DOI: [10.1243/09576509JPE463](https://doi.org/10.1243/09576509JPE463).
- [53] Luo, J., Xiong, J., Liu, F., and McBean, I. Secondary Flow Reduction by Blade Redesign and Endwall Contouring Using an Adjoint Optimization Method (Oct. 10, 2010), 547–562. DOI: [10.1115/GT2010-22061](https://doi.org/10.1115/GT2010-22061).
- [54] Marta, A. C., Shankaran, S., and Stein, A. Blade Shape Optimization Using a RANS Discrete Adjoint Solver. In: *Proceedings of the 2nd International Conference on Engineering Optimization, Number ENGOPT2010-1410, Lisbon, Portugal*. 2010.
- [55] Marta, A. C., Shankaran, S., Wang, Q., and Venugopal, P. Interpretation of Adjoint Solutions for Turbomachinery Flows. *AIAA Journal* 51, 7 (July 2013), 1733–1744. ISSN: 0001-1452, 1533-385X. DOI: [10.2514/1.J052177](https://doi.org/10.2514/1.J052177).
- [56] Marta, A. C., and Shankaran, S. Assessing Turbomachinery Performance Sensitivity to Boundary Conditions Using Control Theory. *Journal of Propulsion and Power* 30, 5 (2014), 1–14. ISSN: 0748-4658. DOI: [10.2514/1.B35087](https://doi.org/10.2514/1.B35087).
- [57] Mueller, L., and Verstraete, T. CAD Integrated Multipoint Adjoint-Based Optimization of a Turbocharger Radial Turbine. *International Journal of Turbomachinery, Propulsion and Power* 2, 3 (Sept. 12, 2017), 14. ISSN: 2504-186X. DOI: [10.3390/ijtp2030014](https://doi.org/10.3390/ijtp2030014).

- [58] Denton, J. D. The Calculation of Three-Dimensional Viscous Flow Through Multistage Turbomachines. *Journal of Turbomachinery* 114, 1 (Jan. 1, 1992), 18–26. ISSN: 0889-504X. DOI: [10.1115/1.2927983](https://doi.org/10.1115/1.2927983).
- [59] Giles, M. B. *UNSFLO: A Numerical Method for Unsteady Inviscid Flow in Turbomachinery*. Gas Turbine Laboratory, Massachusetts Institute of Technology, 1988.
- [60] Frey, C., Kersken, H.-P., and Nurnberger, D. The Discrete Adjoint of a Turbomachinery RANS Solver. In: *ASME Turbo Expo 2009: Power for Land, Sea, and Air*. American Society of Mechanical Engineers, 2009, 345–354.
- [61] Wang, D. X., and Li, Y. S. 3D Direct and Inverse Design Using NS Equations and the Adjoint Method for Turbine Blades. In: Oct. 10, 2010, 537–545. DOI: [10.1115/GT2010-22049](https://doi.org/10.1115/GT2010-22049).
- [62] Wang, D. X., He, L., Li, Y. S., and Wells, R. G. Adjoint Aerodynamic Design Optimization for Blades in Multistage Turbomachines - Part II: Validation and Application. *Journal of Turbomachinery* 132, 2 (2010), 021012. ISSN: 0889504X. DOI: [10.1115/1.3103928](https://doi.org/10.1115/1.3103928).
- [63] Walther, B., and Nadarajah, S. Constrained Adjoint-Based Aerodynamic Shape Optimization of a Single-Stage Transonic Compressor. *Journal of Turbomachinery* 135, 021017 (2013), 021017.
- [64] Walther, B., and Nadarajah, S. An Adjoint-Based Optimization Method for Constrained Aerodynamic Shape Design of Three-Dimensional Blades in Multi-Row Turbomachinery Configurations (June 16, 2014), V02BT39A031. DOI: [10.1115/GT2014-26604](https://doi.org/10.1115/GT2014-26604).
- [65] Giles, M. B. Nonreflecting Boundary Conditions for Euler Equation Calculations. *AIAA journal* 28, 12 (1990), 2050–2058.
- [66] Wang, D. X., and He, L. Adjoint Aerodynamic Design Optimization for Blades in Multistage Turbomachines - Part I: Methodology and Verification. *Journal of Turbomachinery* 132, 2 (2010), 021011. ISSN: 0889504X. DOI: [10.1115/1.3072498](https://doi.org/10.1115/1.3072498).

- [67] Backhaus, J., Engels-Putzka, A., and Frey, C. A Code-Coupling Approach to the Implementation of Discrete Adjoint Solvers Based on Automatic Differentiation. In: Institute of Structural Analysis and Antiseismic Research School of Civil Engineering National Technical University of Athens (NTUA) Greece, 2016, 3828–3842. ISBN: 978-618-82844-0-1. DOI: [10.7712/100016.2076.6428](https://doi.org/10.7712/100016.2076.6428).
- [68] Navier, C. Mémoire Sur Les Lois Du Mouvement Des Fluides. *Mémoires de l'Académie Royale des Sciences de l'Institut de France* 6, 1823 (1823), 389–440.
- [69] Stokes, G. G. *On the Theories of the Internal Friction of Fluids in Motion, and of the Equilibrium and Motion of Elastic Solids*. 1849.
- [70] Reynolds, O. On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion. *Philosophical Transactions of the Royal Society of London. A* 18 (1895), 123–164. JSTOR: [90643](https://www.jstor.org/stable/90643).
- [71] Spalart, P., and Allmaras, S. A One-Equation Turbulence Model for Aerodynamic Flows. In: *30th Aerospace Sciences Meeting and Exhibit*. 30th Aerospace Sciences Meeting and Exhibit. American Institute of Aeronautics and Astronautics, Reno,NV,U.S.A., Jan. 6, 1992. DOI: [10.2514/6.1992-439](https://doi.org/10.2514/6.1992-439).
- [72] Wilcox, D. C. Multiscale Model for Turbulent Flows. *AIAA journal* 26, 11 (1988), 1311–1320.
- [73] Wilcox, D. C. *Turbulence Modeling for CFD*. DCW Industries, 2006. 522 pp. ISBN: 978-1-928729-08-2.
- [74] Menter, F. R. Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications. *AIAA Journal* 32, 8 (Aug. 1994), 1598–1605. ISSN: 0001-1452, 1533-385X. DOI: [10.2514/3.12149](https://doi.org/10.2514/3.12149).
- [75] Wilcox, D. C. Reassessment of the Scale-Determining Equation for Advanced Turbulence Models. *AIAA Journal* 26, 11 (1988), 1299–1310. ISSN: 0001-1452. DOI: [10.2514/3.10041](https://doi.org/10.2514/3.10041).
- [76] Blazek, J. *Computational Fluid Dynamics: Principles and Applications*. Third edition. OCLC: ocn909314295. Butterworth-Heinemann, Amsterdam, 2015. 447 pp. ISBN: 978-0-08-099995-1.

- [77] Jameson, A., Schmidt, W., and Turkel, E. Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. *AIAA paper* 1259 (1981), 1981.
- [78] Brandt, A. Multi-Level Adaptive Solutions to Boundary-Value Problems. *Mathematics of Computation* 31, 138 (1977), 333–390. ISSN: 0025-5718. DOI: [10.2307/2006422](https://doi.org/10.2307/2006422).
- [79] Jameson, A. Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method. *Applied Mathematics and Computation* 13, 3 (Jan. 1, 1983), 327–355. ISSN: 0096-3003. DOI: [10.1016/0096-3003\(83\)90019-X](https://doi.org/10.1016/0096-3003(83)90019-X).
- [80] Aghaei tog, R., Mesgharpoor Tousi, A., and Soltani, M. Design and CFD Analysis of Centrifugal Compressor for a Microgasturbine. *Aircraft Engineering and Aerospace Technology* 79, 2 (Jan. 30, 2007), 137–143. ISSN: 0002-2667. DOI: [10.1108/00022660710732680](https://doi.org/10.1108/00022660710732680).
- [81] Roclawski, H. Rotor-Stator-Interaction of a Radial Centrifugal Pump Stage with Minimum Stage Diameter (2006), 8.
- [82] Karnahl, J., von Wolfersdorf, J., Tham, K.-M., Wilson, M., and Lock, G. CFD Simulations of Flow and Heat Transfer in a Pre-Swirl System: Influence of Rotating-Stationary Domain Interface. In: *Volume 5: Heat Transfer, Parts A and B*. ASME 2011 Turbo Expo: Turbine Technical Conference and Exposition. ASME, Vancouver, British Columbia, Canada, 2011, 665–677. ISBN: 978-0-7918-5465-5. DOI: [10.1115/GT2011-45085](https://doi.org/10.1115/GT2011-45085).
- [83] Bothe, F., Friebe, C., Heinrich, M., and Schwarze, R. CFD Simulation of Incompressible Turbomachinery — A Comparison of Results From ANSYS Fluent and OpenFOAM. In: *Volume 2B: Turbomachinery*. ASME Turbo Expo 2014: Turbine Technical Conference and Exposition. ASME, Düsseldorf, Germany, June 16, 2014, V02BT39A025. ISBN: 978-0-7918-4561-5. DOI: [10.1115/GT2014-26338](https://doi.org/10.1115/GT2014-26338).
- [84] Tonello, N., Eude, Y., de Laage de Meux, B., and Ferrand, M. Frozen Rotor and Sliding Mesh Models Applied to the 3D Simulation of the Francis-99 Tokke Turbine with Code\_Saturne. *Journal of Physics: Conference Series* 782 (Jan. 2017), 012009. ISSN: 1742-6588, 1742-6596. DOI: [10.1088/1742-6596/782/1/012009](https://doi.org/10.1088/1742-6596/782/1/012009).

- [85] Denton, J. D. *Time Marching Methods for Turbomachinery Flow Calculation*. 1979. 47 pp.
- [86] Holmes, D. G. Mixing Planes Revisited: A Steady Mixing Plane Approach Designed to Combine High Levels of Conservation and Robustness. In: *Proceedings of ASME Turbo Expo 2008: Power for Land, Sea and Air*. Berlin, Germany, Jan. 1, 2008, 2649–2658. DOI: [10.1115/GT2008-51296](https://doi.org/10.1115/GT2008-51296).
- [87] Arnone, A., and Pacciani, R. Rotor-Stator Interaction Analysis Using the Navier-Stokes Equations and a Multigrid Method (June 5, 1995), V001T01A041. DOI: [10.1115/95-GT-177](https://doi.org/10.1115/95-GT-177).
- [88] Gundy-Burlet, K. L., and Dorney, D. J. Effects of Radial Location on the Migration of Hot Streaks in a Turbine. *Journal of Propulsion and Power* 16, 3 (May 2000), 377–387. ISSN: 0748-4658, 1533-3876. DOI: [10.2514/2.5589](https://doi.org/10.2514/2.5589).
- [89] Sondak, D. L., and Dorney, D. J. Simulation of Coupled Unsteady Flow and Heat Conduction in Turbine Stage. *Journal of Propulsion and Power* 16, 6 (Nov. 2000), 1141–1148. ISSN: 0748-4658, 1533-3876. DOI: [10.2514/2.5689](https://doi.org/10.2514/2.5689).
- [90] Sondak, D. L., Gupta, V., Orkwis, P. D., and Dorney, D. J. Effects of Blade Count on Linearized and Nonlinear Hot Streak Clocking Simulations. *Journal of Propulsion and Power* 18, 6 (Nov. 2002), 1273–1279. ISSN: 0748-4658, 1533-3876. DOI: [10.2514/2.6063](https://doi.org/10.2514/2.6063).
- [91] Yao, J., Jameson, A., Alonso, J. J., and Liu, F. Development and Validation of a Massively Parallel Flow Solver for Turbomachinery Flows. *Journal of Propulsion and Power* 17, 3 (May 2001), 659–668. ISSN: 0748-4658, 1533-3876. DOI: [10.2514/2.5793](https://doi.org/10.2514/2.5793).
- [92] Rai, M. M. A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations. *Journal of Computational Physics* (1986), 32.
- [93] Rai, M. M. An Implicit, Conservative, Zonal-Boundary Scheme for Euler Equation Calculations. *Computers & Fluids* 14, 3 (Jan. 1986), 295–319. ISSN: 00457930. DOI: [10.1016/0045-7930\(86\)90027-7](https://doi.org/10.1016/0045-7930(86)90027-7).

- [94] Felten, F. N., Kapetanovic, S., Holmes, D. G., and Ostrowski, M. Gas Turbine Temperature Prediction Using Unsteady CFD and Realistic Non-Uniform 2D Combustor Exit Properties. In: ASME, 2008, 1725–1734. ISBN: 978-0-7918-4314-7. DOI: [10.1115/GT2008-50275](https://doi.org/10.1115/GT2008-50275).
- [95] Mbaye, M., Soize, C., Ousty, J.-P., and Capiez-Lernout, E. Robust Analysis of Design in Vibration of Turbomachines. *ASME Journal of Turbomachinery* 135, 2 (2012), DOI: [10.1115/1.4007442](https://doi.org/10.1115/1.4007442).
- [96] Kapetanovic, S., Felten, F. N., Holmes, D. G., and Ostrowski, M. Impact of Combustor Exit Circumferential Flow Gradients for Gas Turbine Temperature Prediction. In: 5 Th International Conference on Computational Heat and Mass Transfer. 2007, 14.
- [97] Taylor, B. *Methodus incrementorum directa & inversa*. Impensis Gulielmi Innys, 1717. 134 pp.
- [98] Hazewinkel, M., ed. *Encyclopaedia of Mathematics: An Updated and Annotated Translation of the Soviet "Mathematical Encyclopaedia"*. Reidel, Dordrecht ; Boston : Norwell, MA, U.S.A, 1988. 10 pp. ISBN: 978-1-55608-010-4.
- [99] Gill, P. E. *Practical Optimization*. Academic Press, 1981. 401 pp.
- [100] Griewank, A. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, Philadelphia, PA, 2008. ISBN: 978-0-89871-659-7 0-89871-659-4.
- [101] Bischof, C., Carle, A., Corliss, G., Griewank, A., and Hovland, P. ADIFOR - Generating Derivative Codes from Fortran Programs. *Scientific Programming* 1, 1 (1992), 11–29. ISSN: 1058-9244. DOI: [10.1155/1992/717832](https://doi.org/10.1155/1992/717832).
- [102] Giering, R., Kaminski, T., and Slawig, T. Generating Efficient Derivative Code with TAF. *Future Generation Computer Systems* 21, 8 (Oct. 2005), 1345–1355. ISSN: 0167739X. DOI: [10.1016/j.future.2004.11.003](https://doi.org/10.1016/j.future.2004.11.003).
- [103] Utke, J., Naumann, U., Fagan, M., Tallent, N., Strout, M., Heimbach, P., Hill, C., and Wunsch, C. OpenAD/F: A Modular Open-Source Tool for Automatic Differentiation of Fortran Codes. *ACM Transactions on Mathematical Software* 34, 4 (July 1, 2008), 1–36. ISSN: 00983500. DOI: [10.1145/1377596.1377598](https://doi.org/10.1145/1377596.1377598).

- [104] Hascoët, L., and Pascual, V. The Tapenade Automatic Differentiation Tool: Principles, Model, and Specification. *ACM Transactions On Mathematical Software* 39, 3 (2013).
- [105] Walther, A. Getting Started with ADOL-C (2009), 10.
- [106] Hogan, R. J. Fast Reverse-Mode Automatic Differentiation Using Expression Templates in C++. *ACM Transactions on Mathematical Software* 40, 4 (July 8, 2014), 1–16. ISSN: 00983500. DOI: [10.1145/2560359](https://doi.org/10.1145/2560359).
- [107] Sagebaum, M., Albring, T., and Gauger, N. R. High-Performance Derivative Computations Using CoDiPack (Sept. 21, 2017). arXiv: [1709.07229 \[cs\]](https://arxiv.org/abs/1709.07229).
- [108] Giles, M. B., and Pierce, N. A. An Introduction to the Adjoint Approach to Design. *Flow, turbulence and combustion* 65, 3-4 (2000), 393–415.
- [109] Nadarajah, S., and Jameson, A. A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization. *AIAA paper* 667 (2000), 2000.
- [110] Nadarajah, S., and Jameson, A. Studies of the Continuous and Discrete Adjoint Approaches to Viscous Automatic Aerodynamic Shape Optimization. *AIAA paper* 2530 (2001), 2001.
- [111] Lewis, R. M. Numerical Computation of Sensitivities and the Adjoint Approach. In: *Computational Methods for Optimal Design and Control*. Ed. by Borggaard, J., Burns, J., Cliff, E., and Schreck, S. Progress in Systems and Control Theory 24. Birkhäuser Boston, Jan. 1, 1998, 285–302. ISBN: 978-1-4612-7279-3 978-1-4612-1780-0.
- [112] Mader, C. A., R. A. Martins, J. R., Alonso, J. J., and Der Weide, E. V. ADjoint: An Approach for the Rapid Development of Discrete Adjoint Solvers. *AIAA Journal* 46, 4 (Apr. 2008), 863–873. ISSN: 0001-1452, 1533-385X. DOI: [10.2514/1.29123](https://doi.org/10.2514/1.29123).
- [113] Marta, A. C., Shankaran, S., Holmes, D. G., and Stein, A. Development of Adjoint Solvers for Engineering Gradient-Based Turbomachinery Design Applications. In: ASME, 2009, 385–397. ISBN: 978-0-7918-4888-3. DOI: [10.1115/GT2009-59297](https://doi.org/10.1115/GT2009-59297).

- [114] Martins, J., Alonso, J., and van der Weide, E. An Automated Approach for Developing Discrete Adjoint Solvers. In: *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, Newport, Rhode Island, May 2006. ISBN: 978-1-62410-040-6. DOI: [10.2514/6.2006-1608](https://doi.org/10.2514/6.2006-1608).
- [115] Nadarajah, S. The Discrete Adjoint Approach to Aerodynamic Shape Optimization. Stanford University, 2003.
- [116] Walther, B. Adjoint-Based Constrained Aerodynamic Shape Optimization for Multistage Turbomachines. McGill University, 2014.
- [117] Launder, B. E., and Spalding, D. B. *Mathematical Models of Turbulence*. Academic Press, London, 1972. 169 pp.
- [118] Bisson, F., and Nadarajah, S. Adjoint-Based Aerodynamic Optimization Framework. In: *52nd Aerospace Sciences Meeting*. 52nd Aerospace Sciences Meeting. American Institute of Aeronautics and Astronautics, National Harbor, Maryland, Jan. 13, 2014. ISBN: 978-1-62410-256-1. DOI: [10.2514/6.2014-0412](https://doi.org/10.2514/6.2014-0412).
- [119] Bertsimas, D., Nohadani, O., and Teo, K. M. Robust Optimization for Unconstrained Simulation-Based Problems. *Operations Research* 58, 1 (Feb. 2010), 161–178. ISSN: 0030-364X, 1526-5463. DOI: [10.1287/opre.1090.0715](https://doi.org/10.1287/opre.1090.0715).
- [120] Luo, J., Xiong, J., Liu, F., and McBean, I. Three-Dimensional Aerodynamic Design Optimization of a Turbine Blade by Using an Adjoint Method. *Journal of Turbomachinery* 133, 1 (2011), 011026. ISSN: 0889504X. DOI: [10.1115/1.4001166](https://doi.org/10.1115/1.4001166).
- [121] Luo, J., Liu, F., and McBean, I. Turbine Blade Row Optimization Through Endwall Contouring by an Adjoint Method. *Journal of Propulsion and Power* 31, 2 (Mar. 2015), 505–518. ISSN: 0748-4658, 1533-3876. DOI: [10.2514/1.B35152](https://doi.org/10.2514/1.B35152).
- [122] Lynch, S. P., and Thole, K. A. Comparison of the Three-Dimensional Boundary Layer on Flat Versus Contoured Turbine Endwalls. *Journal of Turbomachinery* 138, 4 (Jan. 5, 2016), 041008. ISSN: 0889-504X. DOI: [10.1115/1.4032165](https://doi.org/10.1115/1.4032165).
- [123] Pini, M., Persico, G., Pasquale, D., and Rebay, S. Adjoint Method for Shape Optimization in Real-Gas Flow Applications. *Journal of Engineering for Gas Turbines and Power* 137, 3 (Oct. 14, 2014), 032604. ISSN: 0742-4795. DOI: [10.1115/1.4028495](https://doi.org/10.1115/1.4028495).

- [124] Xie, H., Wu, Y., Wang, A., and Ouyang, H. Numerical Investigation of Inlet Distortion for Different Rear Mounted Engine Installations at Taking-Off Conditions. In: ASME, June 15, 2015, V001T01A009. ISBN: 978-0-7918-5662-8. DOI: [10.1115/GT2015-42350](https://doi.org/10.1115/GT2015-42350).
- [125] Hicks, R. M., and Henne, P. A. Wing Design by Numerical Optimization. *Journal of Aircraft* 15, 7 (July 1978), 407–412. ISSN: 0021-8669, 1533-3868. DOI: [10.2514/3.58379](https://doi.org/10.2514/3.58379).
- [126] Shepard, D. A Two-Dimensional Interpolation Function for Irregularly-Spaced Data. In: *Proceedings of the 1968 23rd ACM National Conference On -*. The 1968 23rd ACM National Conference. ACM Press, Not Known, 1968, 517–524. DOI: [10/d7jpkv](https://doi.org/10/d7jpkv).
- [127] Min, J., Harris, D., and Ting, J. Advances in Ceramic Matrix Composite Blade Damping Characteristics for Aerospace Turbomachinery Applications. In: *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. 52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. American Institute of Aeronautics and Astronautics, Denver, Colorado, Apr. 4, 2011. ISBN: 978-1-60086-951-8. DOI: [10.2514/6.2011-1784](https://doi.org/10.2514/6.2011-1784).
- [128] Bogard, D. G., and Thole, K. A. Gas Turbine Film Cooling. *Journal of Propulsion and Power* 22, 2 (Mar. 2006), 249–270. ISSN: 0748-4658, 1533-3876. DOI: [10.2514/1.18034](https://doi.org/10.2514/1.18034).