

# Blade shape optimization of an axial turbine using the adjoint method

by

**Puja Priyadharshini Natarajan**

in partial fulfillment of the requirements for the degree of

**Master of Science**  
in Mechanical Engineering

at the Delft University of Technology,  
to be defended publicly on Friday September 21<sup>st</sup>, 2018 at 10:00 AM.

Student No:	4582691	
Supervisor(s):	Dr. ir. R. Pecnik,	TU Delft
	Ir. S.H.H.J Smit,	TU Delft
Thesis committee:	Dr. ir. C. A. Infante Ferreira,	TU Delft
	Prof. dr. ir. Sikke Klein,	TU Delft
	Ir. Pim Goedhart,	Sulzer turbo service

Cover image is available at

<https://www.udvavisk.com/services/>



# Acknowledgement

I would like to express my sincere gratitude to Dr. ir. Rene Pecnik for providing me with an opportunity to be a part of this project and, for the guidance during the course of the research. I would also like to thank my daily supervisor, Ir. Stephan Smit for his continuous support and enthusiasm. Their guidance helped me in all the time of research and writing of this thesis.

Besides my supervisors, I would like to thank Sjef Mattheij and Ir. Pim Geodhart, Sulzer turbo services for introducing me to the facilities at the company and for providing required information to proceed with the project.

My sincere thanks to my parents and siblings for their support and encouragement. I would like to thank my friends for helping me through all their means and for their timely pep talks which made me feel less away from my family.

Finally, I would like to thank the TU Delft, Energy and Process Department, all professors and fellow students for making thesis journey very pleasant and memorable.

*Puja Priyadharshini Natarajan  
Delft, September 2018*



# Abstract

The main objective of the thesis is to analyze the potential for improving the aerodynamic efficiency of Siemens SGT5-2000E turbine section by optimizing the blade shapes. An adjoint-based shape optimization is implemented at the mid-span of the axial turbine. The optimization is performed, for the stator and the rotor individually, to reduce the entropy generation (objective function) with mass flow rate as the constraint.

The design optimization methodology is implemented using *SU2*, an open-source computational fluid dynamics (CFD) tool coupled with the adjoint-based optimization technique. The *SU2* optimizer algorithm begins by computing the objective function of the existing design by using the flow solver. The flow simulation is performed by solving the RANS equations and SST turbulence model. The discrete adjoint solver utilizes the objective function and constraints to evaluate the gradients of the objective function with respect to the design variables. Each of the design variables is altered to improve the shape and the gradients are used to find an optimal search direction. The algorithm is structured to iterate until an optimal shape is determined.

The optimization methodology is implemented for the existing stator to reduce the entropy generation and an optimal shape is determined. Then, the rotor is optimized with the outlet conditions of the optimized stator as the inlet conditions. The optimized stator and rotor resulted in a significant decrease in entropy generation of about 16% and 24% respectively. Finally, with the optimized blades the stage simulation is performed which resulted in 1.4% increase in the total-to-total efficiency compared to the baseline stage.



# Contents

<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Symbols</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Need for the adjoint method . . . . .	1
1.2 Advancements of optimization techniques. . . . .	2
1.3 Stanford University Unstructured (SU2). . . . .	5
1.4 Research aim . . . . .	5
1.5 Outline of the report . . . . .	5
<b>2 Turbomachinery : Concepts and Principles</b>	<b>7</b>
2.1 Gas turbine. . . . .	7
2.2 Axial turbine stage. . . . .	8
2.2.1 Velocity triangle . . . . .	9
2.3 Two-dimensional losses in turbomachinery . . . . .	11
2.3.1 Mechanisms of entropy generation . . . . .	11
2.3.2 Blade boundary layer losses . . . . .	11
2.3.3 Trailing edge losses . . . . .	13
2.3.4 Effect of Mach number . . . . .	14
<b>3 Numerical Modelling</b>	<b>17</b>
3.1 Laws of conservation . . . . .	17
3.1.1 Mass conservation . . . . .	17
3.1.2 Momentum conservation . . . . .	17
3.1.3 Energy conservation. . . . .	18
3.1.4 Reynolds-Averaged Navier-Stokes (RANS) equation. . . . .	18
3.1.5 Turbulence model . . . . .	19
3.2 Numerical method . . . . .	19
3.2.1 Spatial discretization using FVM. . . . .	20
3.3 Time integration . . . . .	21
3.3.1 Steady simulation . . . . .	21
3.3.2 Linear solvers . . . . .	22
3.4 Boundary conditions . . . . .	22
3.4.1 Wall boundary conditions . . . . .	22
3.4.2 Periodic boundary condition . . . . .	23
3.4.3 Non-reflecting boundary condition . . . . .	23
3.4.4 Mixing plane approach . . . . .	23

<b>4</b>	<b>Principle of Adjoint Method</b>	<b>25</b>
4.1	Adjoint method . . . . .	25
4.1.1	Continuous Vs Discrete adjoint method. . . . .	25
4.2	Discrete adjoint solver . . . . .	26
4.3	Principle of Automatic Differentiation (AD) . . . . .	29
4.3.1	Forward Vs reverse automatic differentiation . . . . .	29
<b>5</b>	<b>Shape optimization methodology</b>	<b>35</b>
5.1	Surface parameterization . . . . .	35
5.2	Mesh deformation . . . . .	36
5.3	Flow solver . . . . .	37
5.4	Adjoint solver . . . . .	37
5.5	Objective function and constraints . . . . .	37
5.6	Optimization framework. . . . .	38
5.7	Alternative stage optimization methodology. . . . .	40
<b>6</b>	<b>Results and Discussions</b>	<b>41</b>
6.1	Computational domain . . . . .	41
6.1.1	Two-dimensional computational domain . . . . .	41
6.1.2	Mesh description . . . . .	42
6.1.3	Quality of the mesh . . . . .	43
6.2	Modelling assumptions . . . . .	44
6.3	Equation of state . . . . .	44
6.4	CFD result of baseline stage. . . . .	45
6.5	Results of the optimized blade shapes. . . . .	49
6.5.1	Optimization of the stator. . . . .	49
6.5.2	Optimization of the rotor . . . . .	54
6.6	Results of optimized blades - Stage simulation . . . . .	59
<b>7</b>	<b>Conclusions and Recommendations</b>	<b>63</b>
7.1	Recommendations . . . . .	64
	<b>Appendix</b>	<b>64</b>
<b>A</b>	<b>Adjoint method formulation</b>	<b>65</b>
<b>B</b>	<b>Mesh around leading and trailing edge.</b>	<b>67</b>
<b>C</b>	<b>Compressibility factor chart</b>	<b>69</b>
<b>D</b>	<b>Variation in the mass flow rate</b>	<b>71</b>
	<b>Bibliography</b>	<b>73</b>

# List of Figures

1.1	Blade parameterization by using non-uniform B-Spline [1]	3
1.2	Local FFD (left) and Global FFD (right) [2].	4
2.1	Sections of gas turbine (Siemens SGT5-2000E).	8
2.2	Mid-span region.	8
2.3	Flow field around the stator and the rotor. (SS: suction side; PS: Pressure side).	9
2.4	Velocity diagram for the turbine stage [3].	10
2.5	Laminar and turbulent boundary layer formation on the blade surface.	12
2.6	Formation of bubble separation [4].	12
2.7	Flow visualization on the suction side [4].	13
2.8	Trailing edge with separated boundary layer [5].	13
2.9	Variation of losses with Mach number for turbine cascade [6].	14
2.10	Variation of loss with Mach number with varying trailing edge thickness [7].	14
3.1	Schematic of the primal mesh and the control volume on a dual mesh (Taken from Palacios <i>et al.</i> , [8]).	20
4.1	Variable graph (refer to Equation (4.23) - (4.25))	31
5.1	Shape optimization methodology. ( $\alpha_g$ - vector of design variables; $J$ - Objective functions /constraints)	39
5.2	Alternative stage optimization methodology.	40
6.1	Geometry of the stator (left) and rotor (right).	42
6.2	Mesh of the stator and rotor.	43
6.3	Static pressure distribution in the stage.	46
6.4	Temperature distribution in the stage.	47
6.5	Mach number distribution in the stage.	47
6.6	Velocity distribution in the stage	48
6.7	Entropy distribution in the stage.	48
6.8	Reduction in the entropy generation for different mass flow rates (kg/s) as constraint ('m' indicate mass flow rate at the inlet and outlet).	49
6.9	Free Form Deformation (FFD) box with 25 CPs (design variables) in the stator (left) and the rotor (right); the fixed points are specified in yellow colour.	50
6.10	Baseline and optimal geometry of the stator.	51
6.11	Variation of mass flow rate of the optimal shape w.r.t baseline shape at each design iteration.	51
6.12	Mach number distribution in the baseline design.	52

6.13	Mach number distribution in the optimal design. . . . .	52
6.14	Pressure distribution around the baseline and optimal geometry. . . . .	53
6.15	Entropy distribution in the baseline (left) and optimal (right) stator. . .	53
6.16	Variation of entropy generation of the optimal design w.r.t baseline design at each design iteration. . . . .	54
6.17	Variation of mass flow rate of the optimal design w.r.t baseline at each design iteration. . . . .	55
6.18	Baseline and optimal geometry of the rotor . . . . .	55
6.19	Zoomed in view of the leading edge of the rotor - baseline (red) and optimal (blue). . . . .	56
6.20	Mach number distribution in the baseline geometry. . . . .	56
6.21	Mach number distribution in the optimal geometry. . . . .	57
6.22	Boundary layer separation in the baseline (left) and optimal (right) geometry. . . . .	57
6.23	Pressure distribution around the baseline and optimal geometry of the rotor. . . . .	58
6.24	Relative Mach number distribution in the baseline design. . . . .	58
6.25	Relative Mach number distribution in the optimal design. . . . .	59
6.26	Entropy distribution in the baseline (left) and optimal (right) shape of the rotor. . . . .	59
6.27	Static pressure distribution in the optimized stage. . . . .	60
6.28	Total temperature distribution in the optimized stage. . . . .	60
6.29	Absolute Mach number distribution in the optimized stage. . . . .	60
B.1	Zoomed in view of the leading edge of the stator (left) and rotor (right). . . . .	67
B.2	Zoomed in view of the trailing edge of the stator (left) and rotor (right). . . . .	67
C.1	Compressibility factor chart [9]. . . . .	69
D.1	Variation in the mass flow rate of the optimized stage w.r.t baseline stage. . . . .	71

# List of Tables

6.1	Mesh independence test. . . . .	42
6.2	Equiangle skewness Vs mesh quality. . . . .	43
6.3	Gas composition. . . . .	44
6.4	Equation of state parameters for the ideal gas. . . . .	45
6.5	Inlet and outlet boundary conditions. . . . .	46
6.6	The operating conditions of the stator and the rotor. . . . .	50
6.7	Thermodynamic properties of the baseline stage. . . . .	61
6.8	Thermodynamic properties of the optimal stage. . . . .	61
6.9	Comparison of the baseline and the optimized stage. . . . .	61

## List of Symbols

$\alpha$	Absolute flow angle	$^{\circ}$
$\alpha_g$	Geometric design variable	—
$\beta$	Relative flow angle	$^{\circ}$
$\eta_{ts}$	Total-to-static efficiency	%
$\eta_{tt}$	Total-to-total efficiency	%
$\kappa$	Thermal conductivity	$W/mK$
$\mu$	Dynamic viscosity	$N\ s/m^2$
$\rho$	Density	$kg/m^3$
$\phi$	Viscous dissipation	$N/m^2$
$\tau$	Shear stress	$N/m^2$
$A$	Area	$m^2$
$c$	Absolute velocity	$m/s$
$c_p$	Specific heat	$J/kg\ K$
$C_p$	Static pressure coefficient	—
$e_0$	Total internal energy	$J$
$\vec{F}^c$	Convective fluxes	—
$\vec{F}^v$	Viscous fluxes	—
$g$	Gravity	$m/s^2$
$G$	Flow solver iteration	—
$H$	Enthalpy	$J$
$J$	Objective function / constraints	—
$L$	Lagrangian function	—
$M$	Mesh deformation matrix	—
$N$	Rotational speed	$rpm$
$N'$	Shifted Lagrangian	—
$N_{CFL}$	Courant-Friedrichs-Lewy number	—
$\vec{n}_{i,j}$	Unit normal	—
$p$	pressure	$Pa$
$P'$	Production term	—
$Q$	Source term	—
$q$	Heat flux	$W/m^2$
$r_{mid}$	Mid radius	$m$
$R$	Residual	—
$R'$	Gas constant	$J/kmol\ K$
$S$	Entropy	$J/K$
$S_{gen}$	Entropy generation	%
$T$	Temperature	$K$
$t$	Time	$s$
$u$	Velocity	$m/s$
$U'$	Blade velocity	$m/s$
$U$	State variable vectors	—
$w$	Relative velocity	$m/s$
$X$	Grid points	—

## Subscripts

0	Stagnation point
1	Stator inlet section
2	Rotor inlet/interface section
3	Rotor outlet section
a	Axial
$\bar{a}$	Adiabatic index
b	Baseline
ct	Critical
in	Inlet
opt	Optimal
out	Outlet
s	Static
t	Tangential
tot	Total
r	Reduced
ref	Reference



# 1

## Introduction

Shape optimization is the predominant technique and it is essential in many fields including the field of turbomachinery. Designing aerodynamically efficient, turbine or compressor, blades have a greater value in the gas turbine market. Any improvement in the performance of the turbomachine translates directly to the reduction of losses. It is not only the aerodynamics that affects the efficiency, but also other factors like stability, structural analysis etc. The present work mainly focuses on optimizing the turbine blades to improve the aerodynamic efficiency. Advancements in the field of computational fluid dynamics (CFD), producing solutions at high accuracy accompanied by the increase in the computational power, provides room for combining the optimization algorithms which results in an automatic design framework. An optimal shape can be achieved with the help of this design framework by minimizing the objective functions (eg: profile loss, entropy generation, etc.). In certain cases, the proposed optimal shape should also satisfy either the geometric constraints (eg: blade thickness, axial chord length etc.) or the flow constraints (eg: mass flow rate, flow angle, etc.), or both. Usually, the optimized design will not be very different from the baseline design. A sensitivity analysis<sup>1</sup> is done to identify the consequences of shape change in the baseline design and then the optimization is executed to improve the performance. In this thesis work, a gradient-based method is used which has the required improvements in terms of computational power and design space. This method is capable of finding the optimal shape within a few design steps, provided an efficient gradient calculation method is used. As evaluating the gradients are often the expensive step in the optimization framework, using an efficient way that accurately determines the sensitivities is very important.

### 1.1. Need for the adjoint method

The gradient-based techniques are efficient at locating local minimum for high dimensional, non-linearly constrained and complex problems. It can attain an optimal solution in a few iterations. Nevertheless, this approach involves not only the evaluation of objective function but also requires an expensive numerical method for calculating

---

<sup>1</sup> The calculation of the derivatives of one or more quantities (objective functions) with respect to one or more independent variables (design variables)

the gradients of the objective function with respect to the design variables. In order to evaluate the gradients in an efficient way, the adjoint approach is implemented in the gradient-based method. The gradient evaluation, in the adjoint method, is independent of the number of design variables and, the cost of its evaluation is of the same order of magnitude of the objective function. For instance, if the problem requires a large number of design variables, it is ineffective to choose the conventional gradient-based methods like finite difference method or linearized method ([10], [11], [12], [13]), because the computation of the gradients is very costly in these methods. Therefore, the adjoint method is the best alternative for the gradient-based optimization with features of fast, exact gradient evaluation, and minimal computational resource requirements ([14], [15]).

An open-source optimization routine is used for this project. Therefore, the literature study is mainly focused on understanding the application of different techniques and which is more suitable for the successful implementation of the adjoint based shape optimization. The following questions are answered based on the literature study,

1. Either to use the discrete or continuous adjoint method?
2. How to parameterize the design space?
3. How to implement the geometric and flow constraints?

The answers to these questions will be discussed in the following passages.

## 1.2. Advancements of optimization techniques.

In aerodynamic design, the first set of the adjoint equations were derived and implemented by Jameson (1995) [16] for transonic flow problems. The author combined the adjoint method and the computational fluid dynamics (CFD) procedures and thereby developed an optimization design tool. He explained both discrete and continuous adjoint method but in his work he employed the latter because of its simplicity in implementation. In the continuous adjoint method, the nonlinear flow equations which are in partial differential equation form are linearized with respect to design variables and then from this linearized equations, the adjoint equation is obtained, followed by discretization. In the discrete adjoint method, the flow equations are discretized first and then they are linearized and the adjoint formulation is then obtained. Both these methods have their own pros and cons. The continuous method has a simple formulation and easy to implement but the discrete method is composed of a large system of coupled equations. The discrete approach sustains consistency between the solution of the flow equations and the gradients. The gradients evaluated by the discrete method can be equivalent to the one obtained from the finite difference method. Whereas, the gradients computed through the continuous approach will be inappropriate with the exact gradients evaluated by the finite difference method. This is because of the dissimilarity in the discretization of equations which might fail to converge when the solution is close to the local minimum. The implementation of the discrete adjoint method is complex because of the enormous requirement of the

computational power. Giles *et al.* (2000) [17] suggested methods to simplify the execution of the discrete approach by using the Automatic Differentiation (AD) (answer to the first question).

The solution to the second question depends on which type of parameterization technique is chosen. Jameson [18], in one of his initial works, defined that the position of each of the mesh nodes as a discrete design variable. The optimization execution would become tedious because of tens of thousands of design variables and it might be much more complicated for three-dimensional design. Later, he used the steepest descent method [16] which had the advantage of tolerating the errors in the gradient evaluation and supported stronger coupling of the flow solver, the adjoint solver and the design optimization to quickly converge the overall design process. The application of the individual surface mesh node as the design variable initiated high-frequency noises in the gradients and collapsed the steepest descent method. This issue was addressed by many researchers where the design surface was parameterized by employing the smooth functions that changed the initial geometry. The parameterization approach facilitates the use of lesser number of design variables (compared to the mesh nodes) and, removes the high-frequency noises without the necessity of smoothing the changes in the design. Hicks and Henne *et al.* (1978) [19] suggested analytic perturbation functions that were distributed over the design surface and provided a reasonable parameterization of the design span. Variation to the shape could be performed to a particular section by specific choice of design variables keeping remaining geometry left unperturbed [20]. The control points of the B-spline curves could also be specified as the design variables.

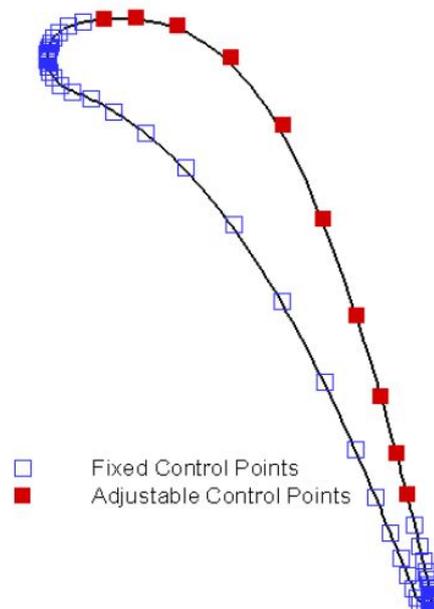


Figure 1.1: Blade parameterization by using non-uniform B-Spline [1]

B-spline works similar to the Hicks-Henne function as it mitigates the number of design variables required and provides local control for improving the shape with higher-order optimization methods like quasi-Newton's procedures [21]. Compared to Hicks-

Henne's approach, B-spline method provided more space for the movement of the given design variables and it required smoothing procedure to eliminate the high frequencies which could be encountered during the gradient calculation.

In addition to these two methods, Free Form Deformation (FFD) could also be used to specify the design variables. This technique employs a FFD box that completely encloses the geometry to be modified (refer Figure 1.2). The control points in the FFD box are chosen as the design variables [2]. By moving the control points, variation to the surface is generated and the perturbation is propagated through the mesh based on the linear elasticity theory [22]. FFD control points provided more flexibility as the span-wise spacing of the control points are sufficiently large and local control of the design variables are also possible. Compared to the B-spline, FFD approach is considered more efficient for complex geometries and it is well suited for the adaptive and the progressive geometry strategies [23]. The application of any of these parameterization techniques had shown significant advancement in optimization by accelerating the convergence and coupling the individual elements of the design process.

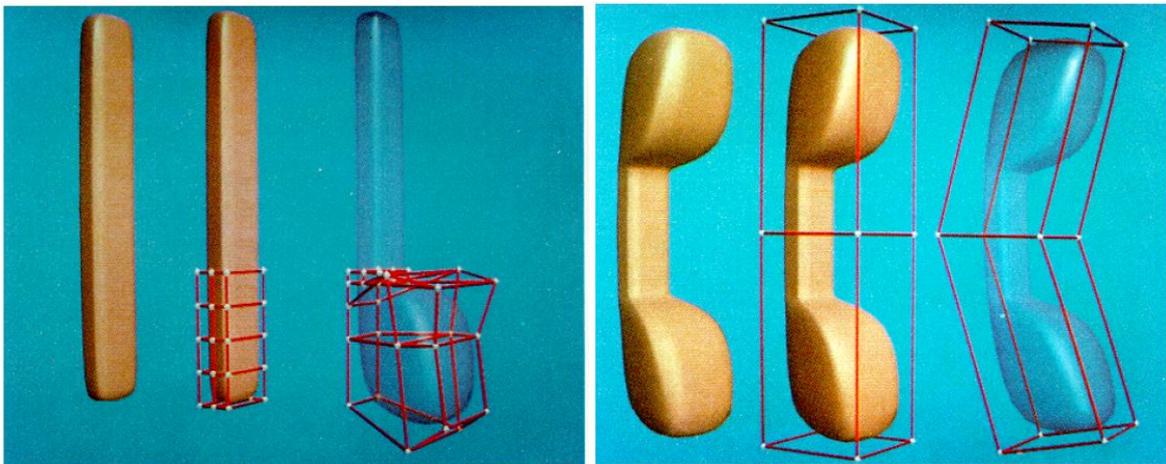


Figure 1.2: Local FFD (left) and Global FFD (right) [2].

The automated optimization design tool aims at attaining a target design using iterative modifications to the shape. Salvatore *et al.* (2017) [23] used an optimization platform to perform the design optimization for a transonic cascade to minimize the entropy generation rate by imposing a constraint on the flow outlet angle which eventually resulted in the reduction of the profile losses. Therefore, the geometric and flow constraints are mainly imposed based on the required result.

In the aerodynamic design optimization, usually the considered design space is smooth and the optimization algorithms are used to find the optimal shapes. The design procedure begins by computing the required parameters of the initial design. These values of the baseline design will be preserved for the gradient calculation in the future. The optimization techniques utilize the objective function and the constraints to evaluate the gradients. Each of the design variables is altered to improve the baseline shape and the gradients are used to decide the optimal shape search

direction. The algorithm will be structured to iterate until an optimized shape is found where there is no further room for improvement.

### 1.3. Stanford University Unstructured (SU2)

*SU2* is an open-source platform written in C++ and Python for the analysis of partial differential equations (PDEs) and constrained optimization problems with state-of-the-art numerical methods. These high-level programming languages in *SU2* have provided the ability to ensure code reuse and flexibility to modify the existing tool for the new and different purpose. It is an industry standard tool with a state-of-the-art adjoint based optimization technology. The performance of this software has been validated by many researchers [23], [24], [25]. For this project, the design and optimization environment of *SU2* is used. The tools for performing the flow simulation, determining the gradients by projecting sensitivities on to the design space, mesh deformation and an optimizer with search algorithm are available in the *SU2* suite [26]. With these coupled capacities, accurate gradients can be calculated to find the optimal design for specific objective function and constraints. To gain a deep understanding of how *SU2* works, refer Palacios *et al.* (2013) [8] .

### 1.4. Research aim

With the aforementioned knowledge, the adjoint-based shape optimization is implemented. The study is focused on the turbine section of the Siemens STG-2000 (V94.2(3)) series gas turbine. This gas turbine is taken as a reference for feasible study to improve the aerodynamic efficiency of the turbine. During the course of the thesis work, a detailed two-dimensional computational fluid dynamic analysis is done at the mid-section of the turbine to analyze the factors affecting the performance of the stage. The shape optimization for the stator and the rotor blades are implemented to reduce the entropy generation.

The aim of this thesis is to reduce the entropy generation of the axial turbine at mid-span by performing the adjoint-based blade shape optimization. This eventually improves the aerodynamic efficiency of the turbine. The shape optimization for the stator and the rotor are done individually as stage optimization features in *SU2* are not yet developed.

### 1.5. Outline of the report

In the second chapter, the concepts of the turbomachinery, the flow pattern around blades and the reasons for the entropy generation in a transonic cascade are explained. In the third chapter, the numerical modelling of the flow equations, turbulence models and boundary conditions implemented in the flow solver are described. In the fourth chapter, the adjoint method is analytically formulated and the working of the automatic differentiation is presented. In the fifth chapter, coupling of the flow solver, the adjoint solver along with the optimization tool is explained. The results and discussions section report the flow characteristics of the optimized blades and the performance of the baseline stage configuration is compared with the optimized stage. Conclusions and recommendations are presented in the last chapter.



# 2

## Turbomachinery : Concepts and Principles

The main purpose of this chapter is to highlight the components of a turbine, their functions and performance. The explanation is mainly focused on the fluid flow pattern around the turbine blades. As the objective of this thesis is to reduce the entropy, the causes of its generation is also outlined in this chapter.

### 2.1. Gas turbine

The open cycle gas turbine is an internal combustion engine that extracts the chemical energy from the fuel and converts it into mechanical energy to generate power. It is composed of four sections: Inlet, compressor, combustion chamber and turbine (Figure 2.1).

- **Inlet section:** It is the section which sucks the air from the atmosphere and properly directs the air on to the compressor section.
- **Compressor section:** In this section, the air is pressurized and then it is fed to the combustion chamber.
- **Combustion section:** The combustion chamber consist of fuel injectors that inject atomized fuel and mixes with the pressurized air. The air-fuel mixture is burned to generate the high temperature and high pressure gas stream and it is then expanded through the turbine section.
- **Turbine section:** The turbine is composed of alternate stationary and rotating airfoil-shaped blades. The high pressure and high temperature gas from the combustion chamber expands through the stationary blades and whirls the rotating blades. The rotating blade drives the compressor to draw more pressurized air into the combustion chamber, and they operate the generator to produce power.

The turbine section of the gas turbine model, Siemens SGT5-2000E is considered for analysis (Figure 2.1). It is an axial turbine composed of four stages. In this thesis, the mid-span section of the first stage of the turbine is focused.

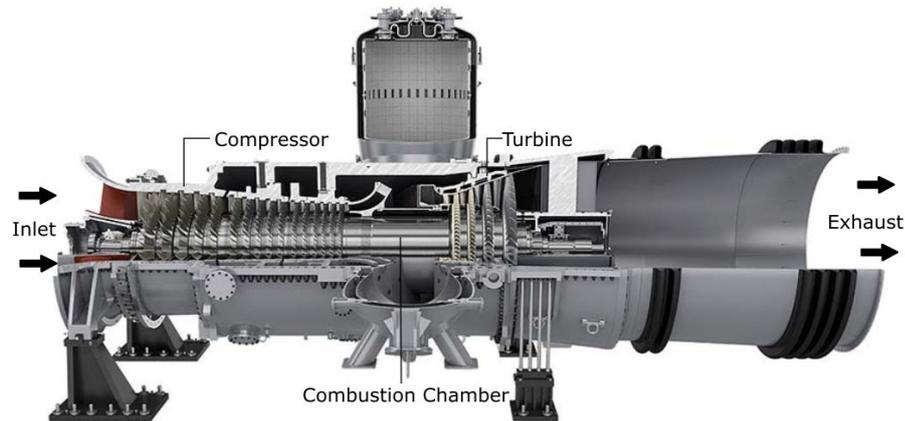


Figure 2.1: Sections of gas turbine (Siemens SGT5-2000E).

## 2.2. Axial turbine stage

The turbine extracts the energy by expanding the high temperature and high pressure gas to low temperature and low pressure gas. Each stage of the turbine is composed of two rows of blades. The first row is composed of stationary blades which are referred either stator or nozzle. The stator form a converging duct that accelerates the hot gas. It directs the high velocity gas to flow at an optimum angle over the rotor and subsequently rotates the rotor. Thus, the velocity of the gas is increased in the stator and the kinetic energy of the fluid is converted to mechanical energy to rotate the rotor. The pressure and temperature of the gas drop gradually through the turbine stages. To generate the required shaft power, the fluid properties (temperature, pressure and velocity) are traded off.

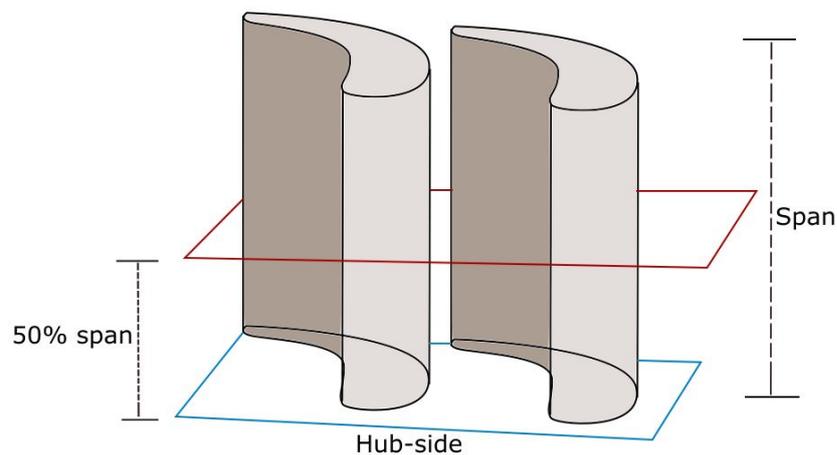


Figure 2.2: Mid-span region.

Figure 2.2 shows the mid-span section of the blade in a stage. The flow enters the inlet at zero degree angle of incidence and the flow splits at the leading edge (stagnation point) and moves along the suction and pressure side of the blade (Figure 2.3). The flow passing over the convex side also referred to as the suction side, is

accelerated. On the concave surface (pressure side), the flow is decelerated. The static pressure coefficient  $C_p$  is defined as the difference between the blade surface pressure and reference pressure at the inlet normalized by the inlet dynamic pressure. The value of  $C_p$  is lowest on the suction side at the location of the throat area where the velocity is higher. Likewise,  $C_p$  is higher at the stagnation point on the blade section. On the suction side, the velocity gradually increases from the stagnation point to the throat area and reduces when it experiences the adverse pressure gradient downstream of the throat in a subsonic flow. The same flow pattern can be observed in both the stator and the rotor.

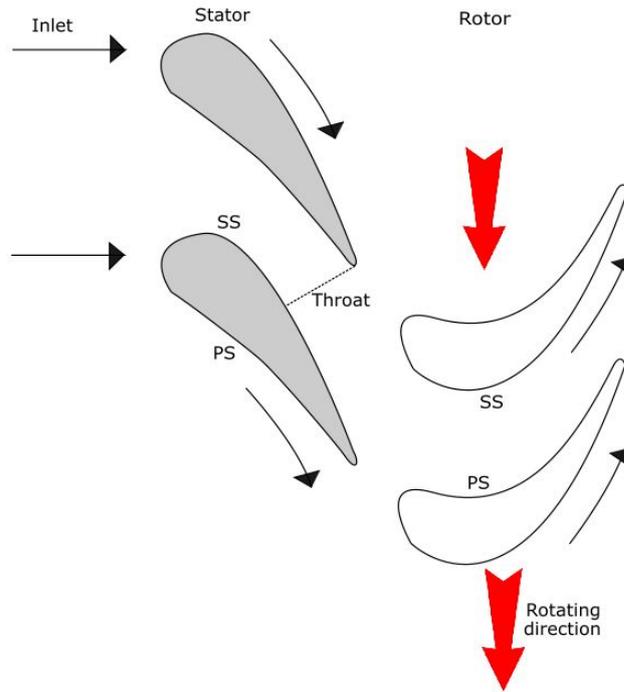


Figure 2.3: Flow field around the stator and the rotor. (SS: suction side; PS: Pressure side).

### 2.2.1. Velocity triangle

The fluid at the inlet strikes the stator at an absolute angle  $\alpha_1$  with velocity  $c_1$ . The fluid velocity and the angle at the inlet and outlet of the turbine can be decomposed into two components : axial and tangential components. As in the case of the axial turbine, the inlet flow angle is parallel to the positive x-direction and the fluid is accelerated in the stator. The increased absolute velocity  $c_2$  at the outlet of the stator turns the fluid in an optimum angle and strikes the rotor and stimulate the blades to rotate. The relative frame of reference gives a better understanding of the flow field than the absolute frame of reference as the rotor is a rotating body. The entire fluid properties can be obtained analytically by assuming the flow to be isentropic. The blade velocity of the rotor depends on the rotational speed ( $N$ ) and the mid-plane radius ( $r_{mid}$ ).

$$\text{Blade velocity, } U' = \frac{2\pi N r_{mid}}{60} \quad (2.1)$$

The rotor inlet and outlet relative velocities are obtained by vectorially solving the

velocity triangle (Figure 2.4) by using the formula specified below.

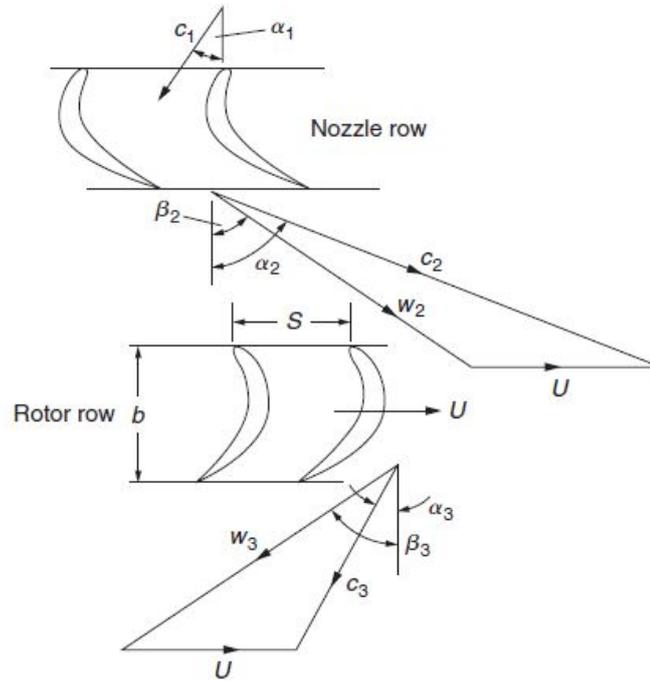


Figure 2.4: Velocity diagram for the turbine stage [3].

#### Stator inlet

$$c_1 = \sqrt{c_{a1}^2 + c_{t1}^2}$$

$$\alpha = 0 \text{ (assumed)}$$

#### Stator Outlet and rotor inlet

$$c_2 = \sqrt{c_{a2}^2 + c_{t2}^2}$$

$$w_{t2} = c_{t2} - U'$$

$$w_{a2} = c_{a2}$$

$$\alpha_2 = \tan^{-1}(c_{t2}/c_{a2})$$

$$\beta_2 = \tan^{-1}((c_{t2} - U')/c_{a2})$$

#### Rotor outlet

$$w_{t3} = c_{t3} + U'$$

$$w_{a3} = c_{a3}$$

$$w_3 = \sqrt{w_{a3}^2 + w_{t3}^2}$$

$$\alpha_3 = \tan^{-1}(c_{t3}/c_{a3})$$

$$\beta_3 = \tan^{-1}(w_{t3}/c_{a3})$$

The suffix 'a' and 't' represents the axial and tangential components, 'c' represents the absolute velocity and 'w' represents the relative velocity. After determining the flow velocities at the inlet, outlet and interface of the stator and rotor, the absolute and relative flow angles can be computed. The thermodynamic properties at each section can also be evaluated based on the expression mentioned below,

#### Stator inlet

$$T_1 = T_{01} - c_1^2/2c_p$$

$$m = \rho * A * c$$

$$p_1 = p_{01} * (T_1/T_{01})^{\bar{a}}$$

$$\rho_1 = p_1/(R'T_1)$$

#### Stator outlet and rotor inlet

$$T_{01} = T_{02}$$

$$T_2 = T_{02} - c_2^2/2c_p$$

$$\rho_2 = p_2/(R'T_2)$$

$$p_{02} = p_2 * (T_{02}/T_2)^{\bar{a}}$$

#### Rotor outlet

$$T_3 = T_{03} - c_3^2/2c_p$$

$$\rho_3 = p_3/(R'T_3)$$

$$p_{03} = p_3 * (T_{03}/T_3)^{\bar{a}}$$

where  $\bar{a}$  is the adiabatic index and it is expressed as  $\bar{a} = \left(\frac{\gamma}{\gamma-1}\right)$ ,  $\gamma$  is the specific heat ratio,  $R'$  is gas constant,  $T$  is the temperature and  $p$  is the pressure. The suffix '1',

'2', '3' indicates the static conditions at inlet, interface and outlet respectively and '0' represents the total conditions. Through this calculation, the flow properties can be calculated even before performing the CFD simulations. In the following section, the factors that affect the efficiency of the turbine is discussed.

## 2.3. Two-dimensional losses in turbomachinery

Enormous upgrades have been done to improve the efficiency of the turbomachines in the past years. This has made these machines to operate at around 90% of total-to-total efficiency. To acquire further improvements is very challenging and imposes the necessity to understand the fluid dynamics and thermodynamics of the flow and, the factors that affect the efficiency. In most cases, the loss is defined as any flow property that affects the efficiency of a turbomachine. The mechanism of losses such as profile loss, trailing edge, etc. are generated due to the combination of various factors and they are hardly independent. According to Denton (1993) [5], in the two-dimensional case, the factors that contribute to the losses are narrowed down to thermodynamic irreversibilities. Any irreversible flow process in a domain generates entropy which definitely reduces the efficiency. The measure of the entropy is independent of the frame of reference. The entropy values remain constant irrespective of whether it is measured in the stationary or the rotating blade row. Entropy is a function of temperature and pressure and the change in entropy is expressed as,

$$\Delta s = s - s_{ref} = c_p \ln(T/T_{ref}) - R' \ln(p/p_{ref}) \quad (2.2)$$

$$= c_v \ln(T/T_{ref}) - R' \ln(\rho/\rho_{ref}). \quad (2.3)$$

The temperature ( $T$ ), pressure ( $p$ ), density ( $\rho$ ) mentioned in the above equation can be substituted either all static or all stagnation values. In the stator, if the stagnation flow properties are considered to determine the entropy change, then the Equation (2.2) reduces to (the stagnation temperature remains constant),

$$\Delta s = -R' \ln(p_{02}/p_{01}). \quad (2.4)$$

The entropy is continuously generated in the boundary layer of the blade surface and in regions where shock waves are produced. It cannot be destroyed rather it is diffused into the surrounding flow of the machine. The loss in efficiency is proportional to the entropy generation.

### 2.3.1. Mechanisms of entropy generation

As the entropy generation by irreversibilities affects the efficiency, it is necessary to understand how the entropy is generated. Usually, the entropy is generated due to the friction either in the boundary layer or free shear layer and, also due to the non-equilibrium processes such as shock waves or sudden expansion of the flow.

### 2.3.2. Blade boundary layer losses

When the fluid passes the blade surface, the fluid closer to the blade tends to stick and slows down the flow. This forms the boundary layer which influences the velocity to change from zero near the blade surface to free stream value away from the blade

surface. The boundary layer can be either laminar or turbulent depending on the Reynolds number of the fluid flow.

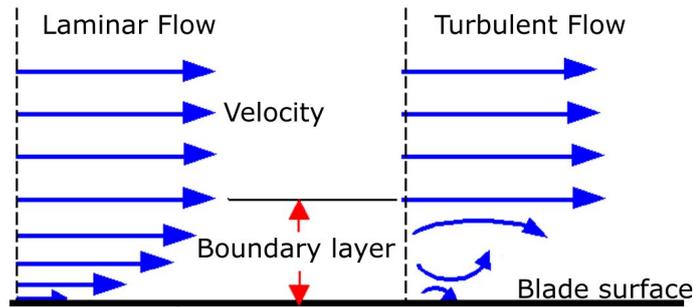


Figure 2.5: Laminar and turbulent boundary layer formation on the blade surface.

The laminar boundary layer is formed at lower Reynolds number, where the stream-wise velocity change is uniform, close to the leading edge. The laminar boundary layer which is formed at the leading edge undergoes a high acceleration on the suction side. According to Hosdson *et al.*, (1987) [27] the over acceleration in the boundary layer leads to a separation bubble close to the meeting point of the leading edge and the suction side of the blade. After the reattachment behind the bubble on the suction side (refer Figure 2.6), the laminar boundary layer continues to grow and separates at the lowest suction pressure region because of the adverse pressure gradient and forms another separation bubble. Due to the adverse pressure gradient, the probability of the boundary layer separation is more at the rear end of the suction side.

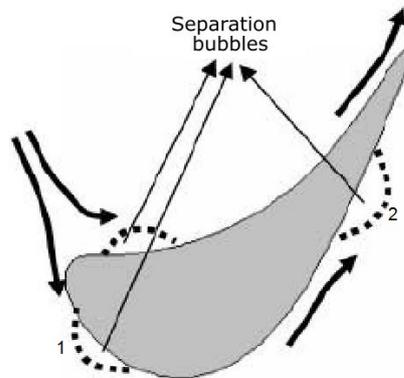


Figure 2.6: Formation of bubble separation [4].

The boundary layer formed experiences transformation from the transition to turbulent and it re-attaches behind the second bubble separation on the suction side. The turbulent boundary layer continues to thicken and may detach again due to the adverse pressure gradient near the trailing edge forming the trailing edge wake. It should be noted that the location of the separation bubble depends on the inlet flow angle and Reynolds number of the flow at the inlet. According to Denton (1993) [5], the major amount of entropy is generated in the turbulent boundary layer and it is highly focused in the inner part of the boundary layer. The formation of the boundary layer at the trailing edge also forms a wake region which gives rise to the local drop

in the stagnation pressure. The entropy generated behind the trailing edge is a direct consequence of the mixing of the boundary layers. This is because of the viscous dissipation in the wake region which is an inescapable effect of the boundary layer formation on the blade surface. Hence, the thickness of the boundary layer is proportional to the quantity of loss on the blade surface. The turbulent boundary layer generally occurs in regions of high velocity gradient which is most likely on the suction surface, both in the stator and the rotor blade surfaces. The entropy created due to this factor can be minimized by maintaining the boundary layer laminar as long as possible.

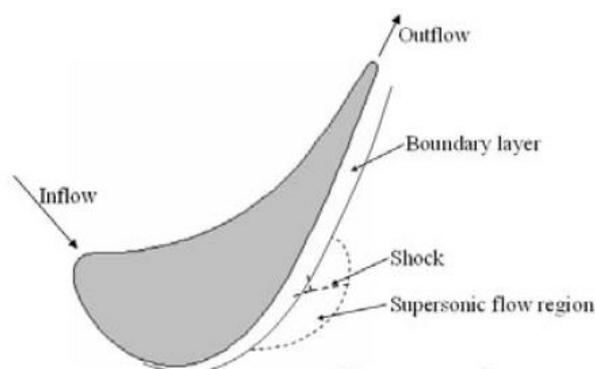


Figure 2.7: Flow visualization on the suction side [4].

### 2.3.3. Trailing edge losses

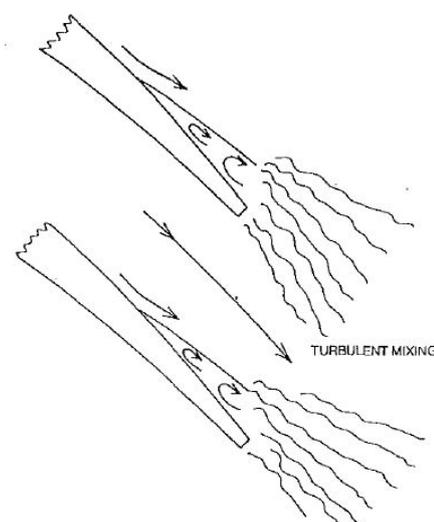


Figure 2.8: Trailing edge with separated boundary layer [5].

The next major part of the two-dimensional losses originates from the trailing edge of the blade. Mee *et al.*, (1992) [6] depicted that one-third of the total loss (Figure 2.9) was due to the mixing losses for a blade which had 6.3% of the blockage at trailing edge in subsonic flow. According to Denton and Xu (1990) [28], in the case of the choked transonic turbine blades with nominal trailing edge thickness, the influence of

the boundary layer losses were less compared to the trailing edge shock losses. This analysis was made by assuming the flow to be inviscid.

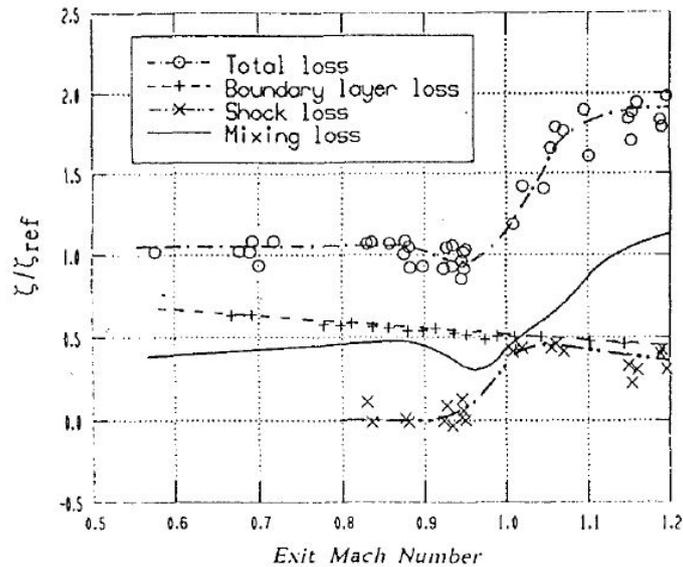


Figure 2.9: Variation of losses with Mach number for turbine cascade [6].

### 2.3.4. Effect of Mach number

The losses in the turbine increases suddenly when the flow approaches the sonic condition. Xu and Denton (1988) [7] made a detail study to understand the variation of the losses with Mach number for varying trailing edge thickness for turbine blades (Figure 2.10) which was in accordance with the results proposed by Mee *et al.*, (1992) [6].

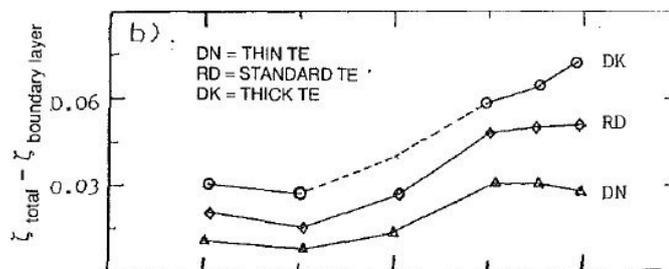


Figure 2.10: Variation of loss with Mach number with varying trailing edge thickness [7].

A correlation was proposed by Chen (1987) [29] to predict the variation of the turbine profile loss with respect to the Mach number. He proposed that the losses increase suddenly as Mach number reaches unity, decrease gradually between Mach numbers 1 and 1.2 and again increases at Mach number more than 1.2. The reason for the decrease in losses for a higher Mach number was explained by Denton and Xu (1990) [28]. They stated that it is due to the expansion of flow from the sonic condition at the throat region to the supersonic condition far downstream is similar to the increase in the flow area at the trailing edge. It is also evident from the result

produced by Mee *et al.*, (1992) [6], that the shock losses were more for higher Mach numbers. Most of the shock appears at downstream of the axial plane and it increases the entropy at the exit of the cascade.



# 3

## Numerical Modelling

In this chapter, the numerical modelling of the problem will be discussed with a specific focus on the fluid dynamic solver.

### 3.1. Laws of conservation

Numerical modelling is considered as the backbone for any computational fluid dynamics (CFD) problem. A control volume is considered as an arbitrary domain ( $\Omega$ ) through which the fluid flows. The specified domain can either be stationary, moving, rotating or deform during flow. Irrespective of the flow behaviour the problem is computed by solving the laws of conservation of mass, momentum, and energy. The differential form of these equations are given below.

#### 3.1.1. Mass conservation

The mass of the system has to remain constant, as mass can neither be created nor be destroyed. The law of conservation of mass states that the total quantity of mass flow entering the domain ( $\Omega$ ) per unit time is equal to the total quantity of mass flow leaving per unit time. The differential form of the above-mentioned statement is,

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0. \quad (3.1)$$

where  $\rho$  is the density of the fluid,  $u$  is the velocity of the fluid particle and,  $i$  is the index which depends on the two or three-dimensional flow field.

#### 3.1.2. Momentum conservation

The conservation of momentum is described by Newton's second law of motion. It states that the rate of change of momentum of the fluid within a domain ( $\Omega$ ) is due to the total flow of the momentum into the domain and the external forces (surface and body forces) acting on the fluid within the domain. The differential form of the momentum equation for the compressible fluid is expressed as,

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}[\rho u_i u_j + p \delta_{ij} - \tau_{ji}] = 0. \quad (3.2)$$

where  $\delta_{ij}$  is the Kronecker delta function, the viscous stress is defined as,  $\tau_{ij} = 2\mu S_{ij}^*$  and the viscous strain-rate  $S_{ij}^*$  is defined as,

$$S_{ij}^* = \frac{1}{2} \left[ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij}.$$

### 3.1.3. Energy conservation

The conservation of energy can be defined in simple terms as the energy can neither be created nor be destroyed but it can be transformed from one form to another. The energy equation can be expressed as,

$$\frac{\partial}{\partial t} (\rho e_0) + \frac{\partial}{\partial x_j} [\rho u_j e_0 + u_j p + q_j - u_i \tau_{ij}] = 0. \quad (3.3)$$

where  $e_0$  is the total internal energy and it is expressed as  $e_0 = e + \frac{1}{2} u_k^2$  and the heat flux ( $q_i$ ) can be written in terms of temperature gradient based on Fouriers Law,

$$q_j = - \left( \kappa \frac{\partial T}{\partial x_j} \right). \quad (3.4)$$

where  $\kappa$  is the thermal conductivity.

### 3.1.4. Reynolds-Averaged Navier-Stokes (RANS) equation

In order to obtain an average form of the governing equations, the continuity, momentum and energy equations are time-averaged (Equation (3.1) - (3.3)). A density time average decomposition (Favre averaging) is implemented for  $u_i$  and  $e_0$ , and a standard time average decomposition (Reynolds averaging) is introduced for  $\rho$  and  $p$ ,

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i} [\bar{\rho} \tilde{u}_i] = 0, \quad (3.5)$$

$$\frac{\partial}{\partial t} (\bar{\rho} \tilde{u}_i) + \frac{\partial}{\partial x_j} [\bar{\rho} \tilde{u}_i \tilde{u}_j + \bar{p} \delta_{ij} + \overline{\rho u_i'' u_j''} - \bar{\tau}_{ji}] = 0, \quad (3.6)$$

$$\frac{\partial}{\partial t} (\bar{\rho} \tilde{e}_0) + \frac{\partial}{\partial x_j} [\bar{\rho} \tilde{u}_j \tilde{e}_0 + \tilde{u}_j \bar{p} + \overline{u_j'' p} + \overline{\rho u_j'' e_0''} + \bar{q}_j - \bar{u}_i \tau_{ij}] = 0. \quad (3.7)$$

The above-mentioned expressions are referred to as Reynolds-Averaged Navier-Stokes (RANS) equation and are expressed in the differential form. As *SU2* is solving these equations using a finite volume scheme, these equations are now written in integral form as given below,

$$\int_{\Omega_i} \frac{\partial U}{\partial t} d\Omega + \sum_{j \in N_i} (F_{c_{ij}} + F_{v_{ij}}) \Delta S_{ij} = Q |\Omega_i|. \quad (3.8)$$

The vectors of the state variables,  $U = (\rho, \rho u_1, \rho u_2, \rho e_0)^T$  where  $\rho$  is the density,  $e_0$  is the total energy per unit mass and  $u$  is the velocity in the two-dimensional flow field.

$F_c$  and  $F_v$  are the convective and viscous fluxes respectively,  $\Delta S_{ij}$  is the surface area of the face related with edge  $ij$ ,  $\Omega_i$  is the volume of the control volume and  $N_i$  is the neighbouring nodes to node  $i$ . The convective and the viscous fluxes are defined as,

$$\vec{F}_i^c = \begin{pmatrix} \rho u_i \\ \rho u_i u_1 + p \delta_{i1} \\ \rho u_i u_2 + p \delta_{i2} \\ \rho u_i H \end{pmatrix} \vec{F}_i^v = \begin{pmatrix} - \\ \tau_{i1} \\ \tau_{i2} \\ u_i \tau_{ij} + \mu_{tot}^* c_p \partial_i T \end{pmatrix} \quad (3.9)$$

where  $i = 1, 2$  is the index,  $p$  is the static pressure,  $H$  is the fluid enthalpy,  $\delta_{ij}$  is the Kronecker delta function and  $T$  is the temperature. The heat-flux ( $q_j$ ) can also be expressed as,

$$q_j = -\kappa \frac{\partial T}{\partial x_j} \equiv -c_p \frac{\mu_{tot}^*}{Pr} \frac{\partial T}{\partial x_j}. \quad (3.10)$$

In order to enclose the viscosity model in the system of equations, the dynamic viscosity ( $\mu_{dyn}$ ) is assumed to satisfy Sutherland's law and the turbulent viscosity is estimated through the turbulence model,

$$\mu_{tot} = \mu_{turb} + \mu_{dym} ; \mu_{tot}^* = \frac{\mu_{dyn}}{Pr_d} + \frac{\mu_{turb}}{Pr_t}. \quad (3.11)$$

where  $Pr_d$  and  $Pr_t$  are the dynamic and turbulent Prandtl numbers, respectively [8].

### 3.1.5. Turbulence model

The turbulence model is implemented to close the RANS equations. There are many methods to carry out the closure model, these are algebraic models with either one or two equations. One of the most successful turbulence model used in the aerodynamic performance analysis is the  $k - \omega$  model because of its accuracy and robustness. It is a two-equation model - the first equation solves for the turbulent kinetic energy ( $k$ ) and the second equation for the specific turbulent dissipation rate ( $\omega$ ). The advantages of this model are that it performs better under adverse pressure gradient than the  $k - \epsilon$  model, the formulation in the viscous sub-layer is simple and has significant numerical stability. However,  $k - \omega$  can over-predict the shear stresses of the adverse pressure gradient, it is very sensitive to boundary conditions at the inlet and has difficulty in convergence compared to  $k - \epsilon$  model.

Shear Stress Transport (SST) model is an improvement made to the  $k - \omega$  model. It addresses the particular imperfection of the standard  $k - \omega$  model. SST can predict the flow separation and reattachment close to the wall, it can be imposed to the viscous affected regions without any modification and it has superior performance in simulating boundary layer with adverse pressure gradients. The detail explanation of the SST model and its implementation in the RANS equation is provided in the paper by Menter (1992) [30].

## 3.2. Numerical method

The partial differential equations (PDEs) are discretized by using a finite volume method or finite element method in *SU2*. It is implemented in an edge-based struc-

ture on a dual grid with control volumes built employing a median-dual, vertex-based scheme (Figure 3.1). The median-dual control volumes are created by attaching the centroids, faces and edge-midpoints of all cells sharing the specific node.

### 3.2.1. Spatial discretization using FVM

The semi-discretized integral form of a typical PDE is given by,

$$\int_{\Omega_i} \frac{\partial U}{\partial t} d\Omega + \sum_{j \in N_i} (F_{c_{ij}} + F_{v_{ij}}) \Delta S_{ij} - Q |\Omega_i| = \int_{\Omega_i} \frac{\partial U}{\partial t} d\Omega + R_i(U) = 0. \quad (3.12)$$

where  $R_i$  represents residual,  $F_c$  and  $F_v$  is the numerical approximation of convective and viscous fluxes respectively.

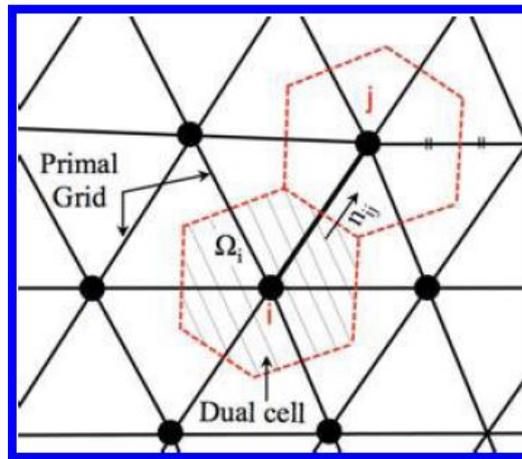


Figure 3.1: Schematic of the primal mesh and the control volume on a dual mesh (Taken from Palacios *et al.*, [8]).

The fluxes are calculated at the midpoint of each edge. The fluxes are evaluated by the numerical solvers which compute the fluxes at all the edges in the primal mesh. These fluxes are combined to evaluate the residual ( $R_i$ ) at each node.

#### Integration of convective fluxes

The discretization of convective fluxes can be performed either by the central or up-wind methods. Several numerical schemes such as JST, Lax-Friedrich, Roe, AUSM, HLLC, Roe-Turkel are available in  $SU2$ , but this section will focus on the Roe scheme. The flux-difference-splitting scheme proposed by Roe [31] calculates the convective fluxes from flow which is reconstructed on both sides of the face from the values at the neighbouring nodes,

$$F_{c_{ij}} = F(U_i, U_j) = \left[ \frac{\vec{F}_c + \vec{F}_v}{2} \right] \vec{n}_{i,j} + P |\Lambda| P^{-1} (U_i - U_j), \quad (3.13)$$

where  $\vec{n}_{i,j}$  is the unit normal pointing outwards related with the face between nodes  $i$  and  $j$ ,  $U_i$  is the vector of the conserved variables at the point  $i$  and  $F_c$  is the convective

flux at node  $i$ .  $P$  is the matrix of eigenvectors of the flux Jacobian matrix, constructed using the Roe averaged variables and projected in the  $\vec{n}_{ij}$  direction, and  $|\Lambda|$  is a diagonal matrix with entries corresponding to the absolute value of the eigenvalues of the flux Jacobian matrix. Second-order accuracy is achieved through reconstruction of variables on the cell interfaces by using a Monotone Upstream-centered Schemes for Conservation Laws (MUSCL) approach [32] with gradient limitation.

### Integration of viscous fluxes

The viscous fluxes are evaluated by employing the finite volume method for which flow quantities and their first derivatives are required at the faces of the control volumes. In *SU2*, the quantities of flow variables, the velocity components, the dynamic viscosity and the heat conduction coefficient are averaged at the cell faces. The weighted least-squares method is used to calculate the gradients of the flow variables at the nodes and then they are averaged at cell faces. The truncation errors of the scheme are mitigated by including the correction equations.

### Source term integration

Source terms ( $Q$ ) are approximated by using piece-wise constant reconstruction within each of the finite volume/finite element cells. The source term plays a basic role in the formulation of the turbulence models.

## 3.3. Time integration

### 3.3.1. Steady simulation

Equation (3.12) must be valid for the whole time interval and  $R_i(U)$  can be evaluated either with explicit methods (time,  $t^n$ ) or implicit methods (time,  $t^{n+1}$ ). *SU2* has both Euler and Runge-Kutta explicit method and, Euler implicit method. Focusing on the implicit Euler scheme to discretize the system which is expressed as,

$$\int_{\Omega_i} \frac{\partial U}{\partial t} d\Omega + R_i(U) \approx |\Omega_i| \frac{dU_i}{dt} + R_i(U) = 0. \quad (3.14)$$

The above equation can be further simplified as,

$$\frac{|\Omega_i^n|}{\Delta t_i^n} \Delta U_i^n = -R_i(U^{n+1}); \text{ where } \Delta U_i^n = U_i^{n+1} - U_i^n. \quad (3.15)$$

Yet the residuals at the time  $n+1$  are unknown, and linearization about  $t^n$  is required,

$$R_i(U^{n+1}) = R_i(U^n) + \frac{\partial R_i(U^n)}{\partial t} \Delta t_i^n + O(\Delta t^2), \quad (3.16)$$

$$= R_i(U^n) + \sum_{j \in N(i)} \frac{\partial R_i(U^n)}{\partial U_j} \Delta U_j^n + O(\Delta t^2). \quad (3.17)$$

Eventually, the following linear system has to be solved to determine the solution update ( $\Delta U_i^n$ ),

$$\left( \frac{|\Omega_i|}{\Delta t_i^n} \delta_{ij} + \frac{\partial R_i(U^n)}{\partial U_j} \right) \Delta U_j^n = -R_i(U^n). \quad (3.18)$$

Even though the implicit scheme is unconditionally stable, a certain value of  $\Delta t_i^n$  is necessary to relax the problem. *SU2* employs the local-time-stepping technique which permits each cell in the mesh to move forward at a different time step and ultimately accelerate the convergence to steady-state. The estimation of the time step needs the value of the eigenvalues and 1<sup>st</sup> order approximations to the Jacobian at each node  $i$  as per the following equation,

$$\Delta t_i = N_{CFL} \min\left(\frac{|\Omega_i|}{\lambda_i^c}, \frac{|\Omega_i|}{\lambda_i^v}\right), \quad (3.19)$$

where  $N_{CFL}$  is the Courant-Friedrichs-Lewy (CFL) number,  $\lambda_i^c$  and  $\lambda_i^v$  are the convective and viscous spectral radius respectively.

### 3.3.2. Linear solvers

The *SU2* suite has many linear solvers for computing the Equation (3.18), which includes Flexible Generalized Minimal Residual (FGMRES) and Biconjugate Gradient Stabilized (BiCGSTAB) methods. Focusing on FGMRES, which is an iterative method for the system of linear equations. This method approximates the solution by the vectors in a Krylov subspace with minimal residual where the vector is determined by Arnoldi iteration. The Lower-Upper Symmetric Gauss-Seidel (LU-SGS) scheme is used as a pre-conditioner to enhance the efficiency of the FGMRES. The preconditioning is the implementation of a transformation to the original system that makes acceptable for the numerical solution.

## 3.4. Boundary conditions

The accuracy of the flow simulation mainly depends, in spite of an appropriate turbulence model (closure model) for RANS equation and other properties of flow equation, on the boundary conditions used. By considering many criteria, different boundary conditions are used at different sections in the computational domain.

### 3.4.1. Wall boundary conditions

#### Adiabatic boundary condition

The adiabatic wall temperature refers to zero thermal conductivity of the blade wall material and this leads to the absence of heat transfer due to conduction. Therefore, the heat flux across the blade surfaces is considered to be zero which implicitly indicates there is no heat loss on the blade wall.

#### No-slip boundary condition

By providing the no-slip condition for the viscous fluid, it is assumed that at a structural boundary (blade), the fluid will have zero velocity relative to the wall boundary. The velocity of the fluid at all the fluid–structure boundaries is equal to that of the solid boundary. The normal velocity components are considered zero and the tangential velocity components are considered to be equal to wall velocity. Therefore, the fluid is assumed to be stuck to the blade surface and it moves with the same velocity as the surface (if the blade is considered moving).

### 3.4.2. Periodic boundary condition

The periodic boundary condition is utilized in the domains which have repeated pattern in the flow distribution. For instance, let us consider a turbine with 'n' number of stator blades and 'n+m' number of rotor blades. By imposing this boundary condition it is self-understood that the flow phenomenon around one blade row considers the influence of the remaining blades in the row. This reduces the computational cost and time.

### 3.4.3. Non-reflecting boundary condition

In the aerodynamic problems, the flow around an isolated airfoil/wing is simulated with a far-field boundary condition by assuming a distance of many chords away from the surface. But in the turbomachines, the far-field boundary is more or less one chord distance away from the blade surface. If the far-field boundary condition is implemented in the turbomachinery blades, it reflects several different unwanted information and which propagates indefinite shock waves. In order to overcome this difficulty, Giles (1990) [33] proposed the non-reflecting boundary condition. The purpose of this boundary condition is to bring the far-field boundary position closer to the blades without affecting the flow field in the adjacent blades. This boundary condition makes the small computational domain more efficient and accurate for calculation without non-physical boundary reflection.

At the inlet boundary, there will be a certain number of incoming modes and at the outlet certain amount of outgoing modes. The average changes in the characteristic of the incoming modes are evaluated to satisfy the user-specified average quantities. When the non-reflecting boundary condition is implemented at the inlet, the average characteristics of the incoming modes are calculated by taking the difference between the computed average quantities and the user-specified quantities to ensure the solution is matching. Similarly, at the outlet condition, static pressure is specified. In order to verify the outlet boundary condition, the characteristic of the average outgoing quantities is evaluated in such a way that the exit static pressure is in accordance with the user-specified value at convergence. The spatial harmonics of the incoming characteristics are imposed by the non-reflecting boundary condition (NRBC) theory based on the amplitudes of the closely similar spatial harmonics of the outgoing characteristics. The characteristic changes at the interface nodes are summed to the equivalent average values, with which the flow equations are linearized at the boundary. The averaging at the interface is done for averaging the flow quantities consistently by using the mixing-out approach which is explained in the following section. The updated values are employed to calculate the conservative variable at the boundary and which is further employed to calculate the convective and the viscous numerical fluxes in Equation (3.8).

### 3.4.4. Mixing plane approach

The mixing plane approach is implemented to average the fluid properties between the stator outlet and rotor inlet (mixing interface). In the mixing plane approach [20], the fluid domains are considered to be at steady-state even though the turbine is composed of one row of stationary blades (stator) and another row of rotating blades

(rotor). The information from one flow field is passed as the boundary condition to the adjacent zones. The fluxes of mass, momentum and energy are averaged and conserved across the interface. The mixing plane approach at the interface has great significance in nullifying the sudden variations in fluid properties like shock waves, wakes and flow separations that can arise due to the difference in the zone to zone flow fields.

# 4

## Principle of Adjoint Method

The purpose of using the adjoint method for the shape optimization is explained in the first chapter. The types of adjoint method, derivation of discrete adjoint method and its implementation in *SU2* is focused in this chapter.

### 4.1. Adjoint method

In the gradient-based shape optimization method, the aim is to minimize a required objective function such as entropy generation, lift or drag, with respect to a set of design variables. Minimization is done through an iterative process which needs the computation of the gradients or sensitivity derivatives of the objective function with respect to the design variables. The calculation of the sensitivity derivatives can be efficiently and cost-effectively performed by the adjoint method. The adjoint method is of two types - continuous and discrete method. In the continuous method, the adjoint equations are obtained from the governing partial differential equations and then eventually discretized. In the discrete approach, the adjoint equations are directly acquired from the discretized governing equations.

#### 4.1.1. Continuous Vs Discrete adjoint method

The continuous and discrete methods have different execution technique even though they are used for the same purpose. Because of this, many users were held up with a doubt - which method is more suitable for performing the design optimization. The researchers have addressed this issue by comparing the performance of these two methods for different scenarios. The following points to be noted:

- The discrete approach provides the exact values for the gradients of the objective function by comparing with the finite difference method. While the continuous approach provides a rough estimation of gradients based on approximate discretization. Therefore, the discrete adjoint method is the more efficient method. The gradients evaluated by the discrete approach are beneficial for any optimizer as the gradient information will be consistent with the evaluation of the objective function.
- The code for the continuous approach is usually straightforward than the discrete method in terms of computational time, memory storage and implementation.

Nevertheless, the discrete method can be made more efficient by structuring the code without explicit storage.

- It is very much complex to derive the continuous adjoint method for the turbulence model with additional source term but the discrete adjoint method performs well in this case.

The early works on aerodynamic shape optimization were based on the continuous methods, while the development of Automatic Differential (AD) tool has provided the sophistication of employing the discrete adjoint approach. Therefore, the discrete adjoint-based shape optimization has been implemented to determine the gradient of the objective function.

## 4.2. Discrete adjoint solver

Let ' $J$ ' be the objective function and the design variables is chosen based on the Free Form Deformation (FFD) method, in which control points are assigned as the geometric design points ( $\alpha_g$ ). A detail explanation of the FFD method formulation will be discussed in the next chapter. Based on the movement of the design variables mapped on to the surface, the mesh deformation happens through linear elasticity method which generates a new mesh ' $X$ '. Then the flow solver evaluates the flow field  $U$  and the objective function  $J$ . (The derivation given below is obtained from Albring *et al*, (2015,2016) [25],[24]). The design optimization problem combining a steady-state constraint could be expressed as,

$$\min_{\alpha_g} J(U(\alpha_g), X(\alpha_g)). \quad (4.1)$$

$$\text{subject to : } R(U(\alpha_g), X(\alpha_g)) = 0. \quad (4.2)$$

Note that  $R(U)$  does not only include the residuals of flow but also the residuals of the turbulence model. Because of this  $U$  will also contain variables of the additional equation.

$$\int_{\Omega_i} \frac{\partial U}{\partial t} d\Omega + \sum_{j \in N_i} (F_{c_{ij}} + F_{v_{ij}}) \Delta S_{ij} - Q|\Omega_i| = \int_{\Omega_i} \frac{\partial U}{\partial t} d\Omega + R_i(U) = 0. \quad (4.3)$$

The compressible RANS equations and the turbulence model are discretized as mention in the previous chapter. The time integration is performed with the implicit Euler scheme to obtain a linear system to solve for  $n$  iterations (refer section 3.3.1),

$$\left( D_{ij}^n + \frac{\partial R(U^n)}{\partial U^n} \right) \Delta U^n = -R(U^n). \quad (4.4)$$

where  $R(U^n)$  is the residual obtained from the spatial integration. The space integration is expressed as,

$$\Delta U_i^n = U_i^{n+1} - U_i^n, \quad (4.5)$$

$$D_{ij}^n := \frac{|\Omega_i|}{\Delta t_i^n} \delta_{ij}; \left( \frac{\partial R(U^n)}{\partial U^n} \right)_{ij} := \frac{\partial R_i(U^n)}{\partial U_j^n}. \quad (4.6)$$

where  $\Omega_i$  indicates the domain of the cells  $i$  and  $\Delta t_i^n$  is the time step which varies in each cell due to time-stepping technique.

It is important to note that if  $\Delta t_i^n \rightarrow \infty$  then  $D^n$  will become zero. If so Equation (4.4) would become identical to a step of Newton-Raphson's method if the non-linear system  $R(U)$  is considered zero. Nevertheless, as the implicit Euler method is used for discretization, a damped Newton's method is obtained. Once the solution is converged, the resulting solution ( $U^*$ ) depends on the residual  $R(U)$  (right side of Equation (4.4)). The flow Jacobian ( $\partial R/\partial U$ ) is the approximation of the left hand side of Equation (4.4). The flow equation Equation (4.4) can be further transformed into a fixed point equation  $U = G(U)$  and the feasible solution is evaluated by the iteration process.

$$U^{n+1} = G(U^n) := U^n - P(U^n)R(U^n). \quad (4.7)$$

Equation (4.7) is the further simplification of the flow equation Equation (4.4) which is expressed in Newton's method. ' $P$ ' refers to the preconditioner and it is defined as,

$$P(U) := \left[ D + \frac{\partial \tilde{R}(U)}{\partial U} \right]^{-1}. \quad (4.8)$$

The tilde over the residual represents that it might be an approximation of the exact Jacobian. The flow solver iteration ' $G$ ' is assumed to be stationary only at the feasible points that are,

$$R(U^*) = 0 \Leftrightarrow U^* = G(U^*). \quad (4.9)$$

By implementing the Banach fixed-point theorem, Equation (4.7) converges only if  $\|\frac{\partial G}{\partial U}\| < 1$ . In  $SU2$  many approximations are made to mitigate the complexity of solving.

- Application of first-order approximation of the implicit terms, even though higher spatial discretization is implemented for solving the residuals ( $R(U)$ ).
- Approximation of linear system (4.4) is only obtained.
- The linearized treatment of the boundary condition is neglected.

These approximations facilitate the efficient evaluation of the discrete adjoint solver. But then these approximations are not applicable if a conventional implementation of the adjoint method is done as mentioned in the section A. This is because, while implementing in the traditional way, the resulting linear system contains the exact flow Jacobian ( $\partial R/\partial U$ ) to be solved for the adjoint variables. Korivi *et al.* (1992) [34] suggested a method for solving the adjoint method which was similar to the iterative flow solver and allows the use of approximate Jacobian. Albring *et al.* (2015) [24] followed Korivi's approach and combined it with the automatic differentiation for the efficient evaluation of the gradients.

When a design optimization is implemented, it is obvious that the computational mesh is subjected to deformation. Based on the movement of the design variables mapped

on to the surface, the mesh deformation happens through linear elasticity method. Nielsen and Park (2006) [35] proposed a method for deriving discrete formulation of the sensitivity analysis that eliminates the necessity of explicit linearization of the mesh movements with respect to geometric design variables (grid sensitivities). The strategy introduced by them had a great impact on the mitigating the cost in linearizing the mesh sensitivities. Albring *et al.* (2015) [24] also combined a similar methodology to the discrete adjoint solver in *SU2*. The mesh deformation routine in *SU2* creates a new mesh with the new grid points ( $X$ ). The change in the computational mesh domain predominantly depends on  $X$ . Because of this, an extra constraint is added to the original optimization problem,  $M(\alpha_g) = X$ . Therefore, the optimization problem is finally reframed as,

$$\min_{\alpha_g} J(U(\alpha_g), X(\alpha_g)). \quad (4.10)$$

$$\text{subject to : } U(\alpha_g) = G(U(\alpha_g), X(\alpha_g)), \quad (4.11)$$

$$X(\alpha_g) = M(\alpha_g). \quad (4.12)$$

where  $J$  represents the objective function to be minimized,  $G$  is the flow solver iteration and  $M$  is surface and mesh deformation matrix.

The Lagrangian function (L) related to the problem is,

$$L(\alpha_g, U, X, \bar{U}, \bar{X}) = J(U, X) + [G(U, X) - U]^T \bar{U} + [M(\alpha_g) - X]^T \bar{X}, \quad (4.13)$$

$$= N'(U, \bar{U}, X) - U^T \bar{U} + [M(\alpha_g) - X]^T \bar{X}. \quad (4.14)$$

where  $N'$  represents the shifted Lagrangian,  $\bar{U}$  and  $\bar{X}$  indicates flow adjoint vector variable and mesh adjoint variable vector respectively.

$$N'(U, \bar{U}, X) := J(U, X) + G^T(U, X) \bar{U}. \quad (4.15)$$

The Lagrangian function (L) is differentiated with respect to geometric design variable by employing the chain rule. The adjoint variables  $\bar{X}$  and  $\bar{U}$  are obtained by eliminating the terms  $\partial U / \partial \alpha_g$  and  $\partial X / \partial \alpha_g$ . The adjoint variables are defined as follows,

$$\bar{U} = \frac{\partial}{\partial U} N'(U, \bar{U}, X) = \frac{\partial}{\partial U} J^T(U, X) + \frac{\partial}{\partial U} G^T(U, X) \bar{U}. \quad (4.16)$$

$$\bar{X} = \frac{\partial}{\partial X} N'(U, \bar{U}, X) = \frac{\partial}{\partial X} J^T(U, X) + \frac{\partial}{\partial X} G^T(U, X) \bar{U}. \quad (4.17)$$

The flow vector adjoint equation (Equation (4.16)), is evaluated using the fixed point iteration approach,

$$\bar{U}^{n+1} = \frac{\partial}{\partial U} N(U^*, \bar{U}^n, X). \quad (4.18)$$

where  $U^*$  is the numerical solution of the flow equations. After computing the adjoint solution of  $\bar{U}$ , sensitivity of the mesh node is computed by solving the Equation (4.17) and the total derivative of the objective function ( $J$ ) with respect to the geometric design variable ( $\alpha_g$ ) which is also equivalent to the total derivative of the Lagrangian and it is expressed as,

$$\frac{dJ^T}{d\alpha_g} = \frac{dL^T}{d\alpha_g} = \frac{d}{d\alpha_g} M^T(\alpha_g) \bar{X}. \quad (4.19)$$

The derivation of the adjoint method is explained in this section clearly. The sensitivity equation Equation (4.17) and Equation (4.16) can be effectively computed by using the automatic differentiation (AD). The significance, types of AD and how it is implemented in *SU2* will be explained in the following section.

### 4.3. Principle of Automatic Differentiation (AD)

Automatic differentiation also known as algorithmic differentiation (AD) is a recent technique to analytically solve the derivative of a function particularized by the computer program. The main objective of using these techniques is to solve complicated equations (like PDEs) faster and without any errors. AD is considered to perform more efficiently than the finite difference method and symbolic method. The finite difference method produces the truncation error while discretization and cancellation. Whereas the symbolic method is easily prone to error and it is quite difficult to convert the computer code into a single equation. If both these methods are employed for higher order derivatives, they are susceptible to increase the complexity and errors. Also, these conventional methods become less efficient if it is used for the partial differential equation with a large number of the inputs. AD works based on the principle that the computer program executes the numerical problem by decomposing it into a sequence of elementary arithmetic operators (addition, multiplication, etc.) and elementary functions (like sin, cos, log, etc.). The derivatives are automatically evaluated by repeatedly implementing the chain rule. In the case of the gradient-based optimization, AD efficiently evaluates the partial derivative of the function with respect to many variations. AD is software dependent for solving these problems. In this thesis, a open source AD tool CoDipack [36] which had been already implemented in the *SU2* [8] is used. Through this open source platform, AD tool could be easily applied for non-linear equations like new turbulence models, transitions models, fluid models or objective functions.

The main rule to implement AD is to decompose the differentials by using the chain rule. A simple expression for better understanding is given as follows,

$$y = f(g(h(x))) = f(g(h(z_0))) = f(g(z_1)) = f(z_2). \quad (4.20)$$

Equation (4.20) can be further rewritten by using chain rule as follows,

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial z_2} \frac{\partial z_2}{\partial z_1} \frac{\partial z_1}{\partial x}. \quad (4.21)$$

AD can be implemented by two methods namely *forward/tangent* and *backward/reverse* automatic differentiation. The forward mode first computes the  $\frac{\partial z_1}{\partial x}$ , then  $\frac{\partial z_2}{\partial z_1}$  and lastly  $\frac{\partial y}{\partial z_2}$ . As name indicates, the reserve mode computes the differential equation in a reverse order.

#### 4.3.1. Forward Vs reverse automatic differentiation

The main difference between these methods is the propagating direction to perform differentiation. Thus, in this section, how a program is decomposed into an elementary sequence and initialization of derivatives based on chain rule is explained.

### Forward Mode

It computes the gradient through the forward propagation. In order to understand the forward mode automatic differentiation, a simple example is considered.

$$z = xy + \sin(x). \quad (4.22)$$

In the above mentioned equation,  $x$  and  $y$  are the variables to be defined by the user. The equation can be further decomposed into,

$$a = xy, \quad (4.23)$$

$$b = \sin(x), \quad (4.24)$$

$$z = a + b. \quad (4.25)$$

Let us assume a variable ' $t$ ' as yet-to-be given variable. The chain rule is expressed as,

$$\frac{\partial w}{\partial t} = \sum \frac{\partial w}{\partial u_i} \frac{\partial u_i}{\partial t}, \quad (4.26)$$

$$= \frac{\partial w}{\partial u_1} \frac{\partial u_1}{\partial t} + \frac{\partial w}{\partial u_2} \frac{\partial u_2}{\partial t} + \dots \quad (4.27)$$

where  $w$  is the output variable and  $u$  represents the input variable that depends on the output variable  $w$ . The decomposed equations are differentiated with respect to yet-to-be given variable ' $t$ ' using the chain rule.

$$\frac{\partial x}{\partial t} = ?, \quad (4.28)$$

$$\frac{\partial y}{\partial t} = ?, \quad (4.29)$$

$$\frac{\partial a}{\partial t} = \frac{\partial x}{\partial t} y + \frac{\partial y}{\partial t} x, \quad (4.30)$$

$$\frac{\partial b}{\partial t} = \cos(x) \frac{\partial x}{\partial t}, \quad (4.31)$$

$$\frac{\partial z}{\partial t} = \frac{\partial a}{\partial t} + \frac{\partial b}{\partial t}. \quad (4.32)$$

The question marks for Equations (4.28) and (4.29) are supplied because the values are unknown. By considering  $t = x$  the entire algorithm will solve for  $\partial z / \partial x$ .

$$\frac{\partial x}{\partial x} = 1, \quad (4.33)$$

$$\frac{\partial y}{\partial x} = 0, \quad (4.34)$$

$$\frac{\partial a}{\partial x} = y, \quad (4.35)$$

$$\frac{\partial b}{\partial t} = \cos(x), \quad (4.36)$$

$$\frac{\partial z}{\partial x} = \frac{\partial a}{\partial x} + \frac{\partial b}{\partial x}. \quad (4.37)$$

Similarly,  $t = y$  is considered while solving for  $\partial z/\partial y$ . Equations (4.28) and (4.29) are referred to as the seed values because they are required to initialize the algorithm. The simplicity of forward method has a disadvantage of running the entire algorithm twice with different initial values. The forward mode performs more efficiently if it is implemented for function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  with  $m \gg n$  where  $n$  is the number of input variables and  $m$  is number of output variable. Therefore, the cost of the forward mode depends linearly on the order of input variable  $O(n)$ .

### Reverse Mode

In order to avoid running the entire the algorithm twice, it is ideal to frame a sequence of the algorithm to compute the output derivative at one run. Let us recall the chain rule employed in the forward mode, where two substitutions were required (  $t = x$  and  $t = y$  ) to calculate the gradients. Since the chain rule is symmetric, let's write the chain rule by reversing the derivatives upside down. By doing so, the role of the input and output is inverted.

$$\frac{\partial s}{\partial u} = \sum \frac{\partial w}{\partial u_i} \frac{\partial u_i}{\partial s}, \tag{4.38}$$

$$= \frac{\partial w}{\partial u_1} \frac{\partial u_1}{\partial s} + \frac{\partial w}{\partial u_2} \frac{\partial u_2}{\partial s} + \dots \tag{4.39}$$

The same naming convention is used where  $u$  and  $w$  are the input and output variables. The yet-to-be given variable  $t$  is replaced with  $s$  in this section to avoid confusion. The chain rule should be implemented repeatedly to the input variables (Equations (4.23) - (4.25) ) in the backward direction to frame a program in which by substituting some value for  $s$ , the derivative has to be determined in one run.

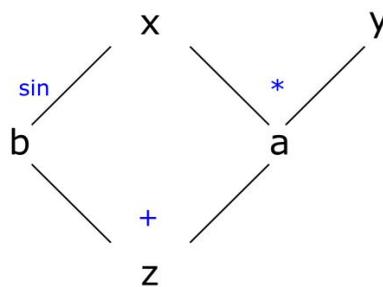


Figure 4.1: Variable graph (refer to Equation (4.23) - (4.25))

The differentiation has to be done by questioning which output variable will affect the given input variable. This is the basis for the reverse mode where the chain rule is applied in the backward direction. From the Figure 4.1, it is easy to say that variable  $a$  depends on  $x$  and  $y$  or directly affects  $z$ ,  $b$  depends only on  $x$  or directly affects  $z$ ,  $z$  depends on  $a$  and  $b$ , variable  $x$  can directly affect both  $a$  and  $b$  and, variable  $y$  can directly affect  $a$ . By implementing this verbal strategy to the Equations (4.23) -

(4.25), the following expressions are obtained.

$$\frac{\partial s}{\partial z} = ?, \quad (4.40)$$

$$\frac{\partial s}{\partial b} = \frac{\partial s}{\partial z}, \quad (4.41)$$

$$\frac{\partial s}{\partial a} = \frac{\partial s}{\partial z}, \quad (4.42)$$

$$\frac{\partial s}{\partial y} = x \frac{\partial s}{\partial a}, \quad (4.43)$$

$$\frac{\partial s}{\partial x} = y \frac{\partial s}{\partial a} + \cos(x) \frac{\partial s}{\partial b}, \quad (4.44)$$

Just to give an idea on how the above equations are obtained. Consider the Equation (4.44), where the variable  $x$  is affected by both  $a$  and  $b$ , the chain rule is expressed as  $\partial s / \partial x = (\partial s / \partial a)(\partial a / \partial x) + (\partial s / \partial b)(\partial b / \partial x)$ . If  $s = z$  is substituted in the above equations, the gradients can be obtained from the last two equations. This is equivalent to initializing the entire program by assuming  $(\partial s / \partial z) = 1$ . Thus, by performing a repeated differentiation for input variables rather than the output variable, we can obtain gradients in one run. If the gradient of different output variables has to be calculated, then the program has to be re-run with different seed values. The cost of the reverse mode depends on the order of the number of output variables.

#### Implementation in *SU2*

*SU2* is an open source integrated design tool for calculating the versatile problems governed by the PDEs on the unstructured meshes written in C++. It is based on object-oriented structure, in which the discrete adjoint solver can be implemented using the Operator overloading approach along with the Expression Templates. This approach provides flexible implementation and requires only the input or output variables of the solver. The data structure and optimization methods established through the discrete adjoint solver is smoothly integrated with the existing framework of the *SU2*. This implementation can generate the gradients automatically without any manual modification during one evaluation of the flow solver iteration  $G$  (refer Equation (4.7)). The extended iteration can be expressed as,

$$\begin{pmatrix} V \\ W \end{pmatrix} = \begin{pmatrix} G(U, X) \\ J(U, X) \end{pmatrix} \quad (4.45)$$

$V$  and  $W$  are the seed values. The above equation indicates one iteration of the flow solver and the immediate execution of the objective function at an arbitrary flow field  $U$ . By applying the reverse AD to Equation (4.45),

$$\begin{pmatrix} \bar{U} \\ \bar{X} \end{pmatrix} = \begin{pmatrix} \frac{\partial G(U, X)^T}{\partial U} & \frac{\partial J(U, X)^T}{\partial U} \\ \frac{\partial G(U, X)^T}{\partial X} & \frac{\partial J(U, X)^T}{\partial X} \end{pmatrix} \begin{pmatrix} \bar{V} \\ \bar{W} \end{pmatrix} \quad (4.46)$$

By solving the above equation, the adjoint variables  $\bar{U}$  and  $\bar{X}$  can be determined, provided an arbitrary choice is made for  $\bar{V}$  and  $\bar{W}$ . As the discrete adjoint implementation

in  $SU2$  is based on Albring *et al.*, [24] work, similar values for the assumption is chosen in this section. Hence,  $\overline{W}$  is assumed as  $\overline{W} \equiv 1$ .

$$\begin{pmatrix} \overline{U} \\ \overline{X} \end{pmatrix} = \begin{pmatrix} \frac{\partial G(U, X)^T}{\partial U} & \frac{\partial J(U, X)^T}{\partial U} \\ \frac{\partial G(U, X)^T}{\partial X} & \frac{\partial J(U, X)^T}{\partial X} \end{pmatrix} \begin{pmatrix} \overline{V} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{\partial G(U, X)^T}{\partial U} \overline{V} & \frac{\partial J(U, X)^T}{\partial U} \\ \frac{\partial G(U, X)^T}{\partial X} \overline{V} & \frac{\partial J(U, X)^T}{\partial X} \end{pmatrix} \quad (4.47)$$

In Equation (4.47), matrix on the right hand represents the shifted Lagrangian (N) (refer Equation (4.15)) determined at  $(U, \overline{V}, X)$ . The information can be easily stored by performing iteration of  $G(U^*, X)$  starting from  $n = 0$  using the Expression template approach. The succeeding iteration is initialized with some assumption ( $\overline{U}^0$ ) and then the stored information can be evaluated with the  $\overline{V}$  set to already existing adjoint solution  $\overline{U}^n$  and  $\overline{W} \equiv 1$  in the Equation (4.46) to evaluate  $\frac{\partial N}{\partial U}(U^*, \overline{U}^n, X)$ . The iteration is proceeded until  $|\overline{U}^n|$  is sufficiently small. The mesh sensitivity equation (Equation (4.19)) is solved for the mesh deformation routine by applying the reverse automatic differentiation.



# 5

## Shape optimization methodology

Up to now the framework of the flow and discrete adjoint solvers were explained in detail. In this chapter, the approach for the geometric and mesh deformation will be discussed. A detailed explanation of the fully automated shape optimization algorithm in *SU2* is presented.

### 5.1. Surface parameterization

The variation of the objective function ( $J$ ) with respect to the shape deformations in the direction normal to the surface is computed by applying the discrete adjoint method. The choice of design variable has to be made in such a way that it will not affect the computation time and other complexities. It is even possible to select each of the surface nodes in the computational mesh as a design variable, but this methodology cannot be implemented for three-dimensional structures because the final count of design variable will be hundreds of thousands. A more practical approach is to determine the surface sensitivities at each of the mesh nodes on the surface to be designed. This information is further projected on to the design space composed of design variables. The surface sensitivities are evaluated in a repeated fashion by following the above-mentioned procedure in order to march the design towards an optimum shape through gradient-based optimization method.

The parameterization of the geometry is done by the Free-Form Deformation (FFD) approach [2]. This technique has origins in the computer graphics industry. The FFD box encapsulates the body (blade) that has to be redesigned and it is parameterized as a Bezier solid. The FFD box bounding around the surface will create a mapping between the mesh nodes and the control points (the intersection of the lines on the outer box). The control points in the box control the shape of the geometry by performing a smooth deformation. The variation in the surface due to the perturbation of the FFD control points propagate through the mesh. Rather than regenerating the entire mesh for each design variation, automatic mesh re-generator is in-built in the mesh deformation routine. Because of automatic mesh generation, the initial information of the geometry is preserved and it is an efficient and cost-effective method. Therefore, the deformation of the mesh is propagated through the linear elasticity method [22]. The group of control points which are defined on the surface of the

box, depends on the Bernstein polynomials. The box is parameterized by following the expression below,

$$X(u, v) = \sum_{i=0}^l \sum_{j=0}^m P_{i,j} B_i^l(u) B_j^m(v) \quad (5.1)$$

where  $l, m$  are the degree of the FFD function,  $u, v$  are the parametric coordinates,  $P_{i,j}$  are the coordinates of the control point  $(i, j)$  and  $B_i^l(u) B_j^m(v)$  are the Bernstein polynomials. The Cartesian coordinates of the points on the surface of the object are then changed to parametric coordinates within the Bezier box. The FFD boxes provide more flexibility in deforming the shape and in order to avoid unfeasible shape, the control points mapped on the computational domain can be locally controlled by immobilizing.

## 5.2. Mesh deformation

The potential to deform the mesh simultaneously with the change in the geometry is a prime tool in CFD. It is very essential in a shape optimization procedure. After each design process the shape of the geometry will be deformed and the volume of the mesh should also be altered accordingly to avoid negative volumes. The simplest way to deal with moving boundaries is to completely regenerate a new mesh after each design iteration. This method is very robust because the new mesh does not have to take into account the properties of the previous mesh and therefore can handle very large deformations. However, regeneration also has some serious drawbacks like loss of physical conservation laws, because the physical quantities have to be interpolated from the old to the new mesh in some way. Moreover, re-meshing is computationally burdensome for complex geometries. Thus, an automatic mesh generation technique is required to project the solution from the previous mesh to the new mesh. *SU2* uses the linear elasticity method for automatic mesh regeneration and update. The linear elasticity method is also referred to as the solid body elasticity approach assumes the volume mesh to act as a solid body with linear elasticity [22]. The equation of linear elasticity governs small displacements,  $V = (u_1, u_2, u_3)^T$ , of an elastic solid body subject to body force and surface friction.

$$\frac{\partial^2 V}{\partial t^2} - \Delta \sigma = f \text{ in } \Omega, \quad t > 0, \quad (5.2)$$

where  $f$  represents the body force and  $\sigma$  is the stress tensor given in terms of the strain tensor ( $\epsilon$ ) and it is written as follows based on the constitutive relation,

$$\sigma = \lambda \text{Tr}(\epsilon) I + 2\mu \epsilon, \quad (5.3)$$

where  $\text{Tr}$  is the trace,  $\lambda$  and  $\mu$  are the Lamé constants,  $E$  is the Young's modulus, and  $\nu$  is Poisson's ratio.  $\lambda$  and  $\mu$  are expressed in terms of Young's modulus  $E$  and Poisson's ratio  $\nu$  as,

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)}, \quad \mu = \frac{E}{2(1 + \nu)}, \quad \epsilon = \frac{1}{2}(\Delta u + \Delta u^T). \quad (5.4)$$

Young's modulus greater than zero ( $E > 0$ ) indicates the stiffness of the material, where a large value of  $E$  indicates rigidity. Poisson's ratio ( $\nu$ ), is a measure of how much the material shrinks in the lateral direction as it extends in the axial direction. The advantage of this method is that it produces high-quality grids.

### 5.3. Flow solver

The governing equations (steady-state RANS equations) are spatially discretized using the finite volume method. The implicit Euler time integration scheme is implemented to accelerate the local time-stepping and to obtain convergence to a steady-state solution. The convective fluxes are computed by using Roe's scheme with second order accuracy and the oscillation near shocks are controlled by using van-Albada type slope limiter. The viscous fluxes are evaluated by the weighted least square method. Shear Stress Turbulence (SST) model is implemented for the turbulence closure problem. Boundary conditions are implemented weakly by using the ghost cell concept. The adiabatic and the no-slip boundary conditions are imposed for the blade walls. Non reflecting boundary conditions are implemented for the subsonic inlet and outlet boundary conditions. At the inlet of the domain, the stagnation temperature and pressure is specified along with static pressure at the outlet. Further to mitigate the computational cost, the periodic boundary conditions are applied to the circumferential lateral walls from the suction and pressure side of the blade. The solver provides the fully converged flow solution which is directly inherited by the adjoint solver.

### 5.4. Adjoint solver

The discrete adjoint method in the *SU2* constructs the required Jacobian matrix by mirroring the solution of the direct flow problem. The flow residual is evaluated by initially looping over each of the edges to identify the contribution from up-wind discretization and then looping over the boundary nodes to add the appropriate boundary condition contributions. Hence, in the discrete adjoint, the contributions to the Jacobian matrix from the fluxes across edges are first estimated, followed by the contributions from the boundary nodes. The derivatives of the fluxes, which is employed in both edges and boundary conditions, are calculated by applying Automatic Differentiation. The adjoint solver solves the PDEs to determine the surface sensitivities which are the measure of change in objective function at each node on the surface due to the variation of the design variable in local normal direction. The surface sensitivity value is evaluated at each node of the numerical grid with negligible computational cost in *SU2*.

### 5.5. Objective function and constraints

The main objective of this entire study is to increase the efficiency of the turbine at mid-span by reducing the losses. The factors that affect the efficiency are discussed already in chapter 2. Therefore, the entropy generation is considered as the objective function. In addition to this, constraints are assigned to the optimization routine to meet the design requirements and to avoid unfeasible designs. Therefore, the design loop is structured to find the optimal shape with minimized objective function by sat-

isfying the constraint. The constraint is imposed on the mass flow rate at the inlet and outlet of the domain. An inequality constraint for mass flow rate is imposed by assuming a higher value than the reference mass flow rate at the inlet and outlet section.

The optimization problem is formulated as,

*Minimize*

$$\text{Objective function} \equiv \text{Entropy generation } (S) \equiv \frac{S_{out} - S_{in}}{S_{in}}$$

*Subject to*

$$\text{Constraint}_1 \equiv \dot{m}_{in, \text{ estimated at each design step}} < \dot{m}_{in, \text{ assumed}}$$

$$\text{Constraint}_2 \equiv \dot{m}_{out, \text{ estimated at each design step}} < \dot{m}_{out, \text{ assumed}}$$

The constraints 1 and 2 are implemented at the inlet and outlet of the domain respectively. With these constraints, the optimizer has the freedom to alter the shape by reducing the entropy generation and maintain the mass flow rate within the tolerance limit of +/- 0.5%.

## 5.6. Optimization framework

The procedure that is used to perform design optimization in *SU2* is showed in the form of flow chart in Figure 5.1. In order to understand the design optimization procedure, the tools required to perform this design loop is explained. The design methodology discussed below is based on the progression of the flow chart.

- To begin with optimization procedure, two input data are required - 1. The meshed geometry for which the aerodynamic shape optimization has to be implemented. 2. The configuration file with boundary conditions, objective function, constraints and design variables.
- The design process is automated by an optimization algorithm. In *SU2*, Sequential Least Square programming (SLSQP) algorithm is used. It is a *minimization* routine of the python code (*scipy.optimize*). At the beginning of each design iteration, the optimizer receives the flow variables and gradients of the objective function with respect to the design variables from the flow solver and the adjoint solver respectively. Based on the sensitivity analysis, the optimal shape search direction is determined by the optimizer and it computes a new set of design variables.
- The new design variables are given as the input for the shape parameterization module and it generates the displacements. The deformation of the geometry is performed by the Free-Form Deformation (FFD) approach. The FFD boxes encapsulate the geometry and create a mapping between the FFD control points and mesh surface nodes. These control points are the design variables which propagates the deformation smoothly as it varies.

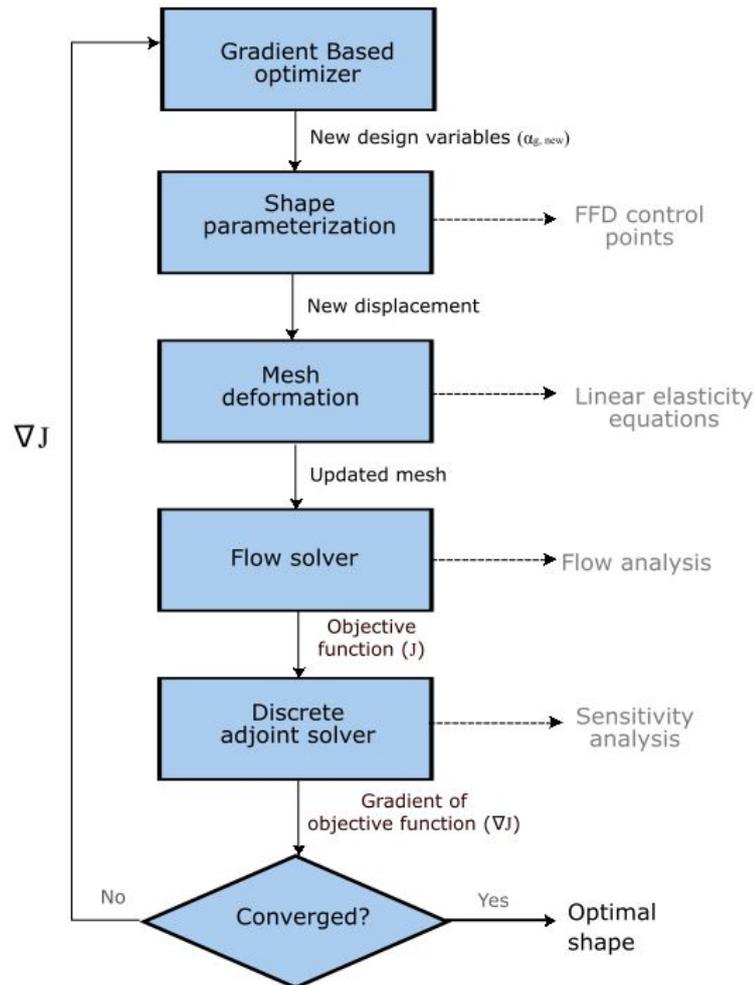


Figure 5.1: Shape optimization methodology. ( $\alpha_g$  - vector of design variables;  $J$  - Objective functions /constraints )

- As the shape of the surface changes, the volume of the mesh should also be deformed simultaneously to avoid the negative volumes and prevent the optimization process from failing. *SU2* treats the mesh as an elastic solid and assigns the non-uniform stiffnesses to all the elements in the mesh. The linear elasticity equations are solved on the mesh to calculate the nodal displacements by using the movement of the boundaries as input.
- With this updated geometry and mesh, flow simulation is done to analyze the performance . The analysis is done by using the flow solver by the finite volume method to evaluate the objective/constraint function(s).
- The sensitivity analysis is done by using the adjoint solver. The gradients of the objective function with respect to the new set of design variables (FFD control points) are determined.

The design optimization procedure is summarized as follows. Once the python script is initiated, the gradient-based optimizer will coordinate the shape optimization cycle consisting of the flow solver, adjoint solver and mesh deformation tools available

in *SU2* platform. The optimizer continues to change the shape of the surface, deform the mesh, compute the performance and the gradients until it converges to an optimal value of the objective function ( $J$ ). The algorithm is structured to run until it reaches the optimization criteria or a maximum number of iterations.

### 5.7. Alternative stage optimization methodology.

The main objective of this thesis is to perform shape optimization for the stage. As the stage optimization feature is still not available in *SU2*, an alternative method is used.

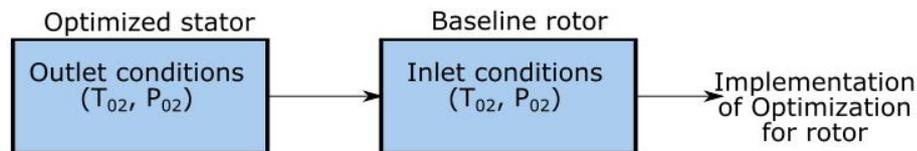


Figure 5.2: Alternative stage optimization methodology.

The optimization methodology for the stator is performed as mentioned in Figure 5.1. The outlet conditions (total temperature and total pressure) of the optimized stator is used as the inlet operating conditions for the baseline rotor. The mixing-plane approach is implemented for both the stator and the rotor. In this way, the two individual optimized shapes are expected to result in the optimal shape as a stage.

# 6

## Results and Discussions

The results of the baseline design and the optimized design will be presented in this chapter. Firstly, the computational domain considered for optimization will be presented, followed by the simulation results of the baseline design. Furthermore, the results of the optimized case will be discussed by comparing with the baseline design.

### 6.1. Computational domain

The stator and the rotor geometry of the first stage of the Siemens SGT5-2000E are analyzed to understand the flow physics around the blades. The main objective is to reduce the entropy generation by optimizing the shape of the blades and eventually increasing the stage efficiency.

#### 6.1.1. Two-dimensional computational domain

A mid-plane analysis is done for the stator and the rotor in this assignment. The computational domain for the blades are constructed separately. The blade coordinate points are extracted at mid-span from the 3D geometry and connected through spline to form a surface. The inlet of the stator and the axial distance between stator and rotor are the exact measure of the given 3D geometry. The periodic boundaries (Wall 1 and Wall 2 in Figure 6.2) for the blades are constructed from the midpoint of the blade, by translating one half of the pitch distance up and down. Then periodic boundaries are connected smoothly with the inlet and outlet of the blade.

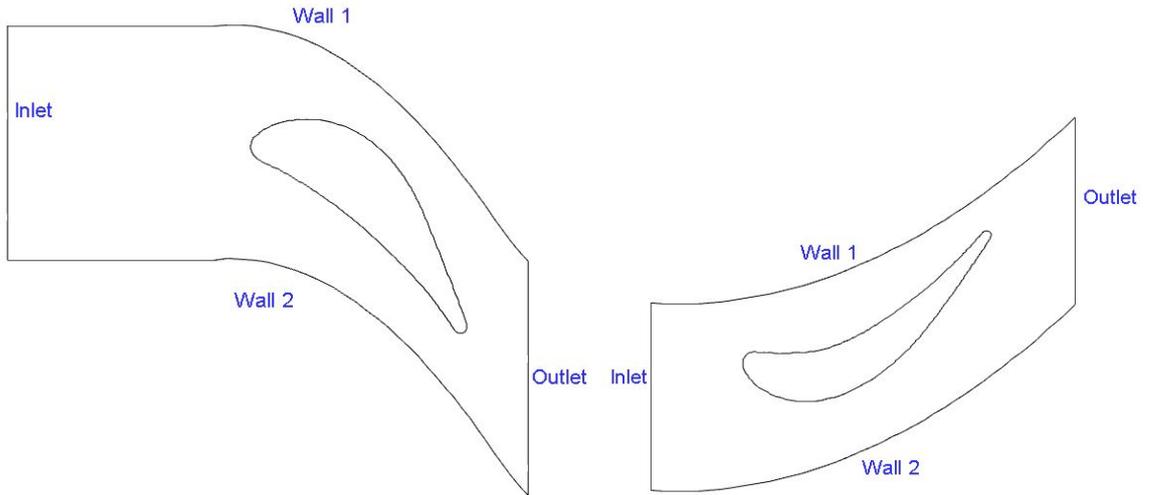


Figure 6.1: Geometry of the stator (left) and rotor (right).

### 6.1.2. Mesh description

The computational flow domain is divided into two regions: (i) structured quadrilateral O-grid mesh body-fitted around the blade surface, (ii) unstructured triangular mesh in the far field. The O-grid mesh encloses the surface close to the blade thereby forming a structured boundary layer mesh. The structured mesh is then extended into unstructured mesh till the far-field boundaries. The semi-unstructured mesh generally allows more flexibility than the standard mesh (only quadrilateral or triangular mesh) in terms of skewness minimization and convergence rate [37].

The boundary layer regime is discretized in the direction normal to the blade, in such a way that the first wall distance from the blade is estimated based on the  $y^+$ . The  $y^+$  value is maintained less than unity to capture the wake as accurately as possible and it implicitly predicts the shock and entropy generation regions. The structured mesh is mapped on to the unstructured mesh for smooth propagation of the flow information. Mesh independence test is carried out to check the accuracy of the CFD solution. The simulation is performed for three different number of grid elements with the same boundary conditions. The variation of the total-to-total efficiency is checked for different meshes and the simulation is conducted in 12 cores.

Table 6.1: Mesh independence test.

Category	Number of elements in			Computational time in 12 cores (s)	$\eta_{tt}$ (%)
	Stator	Rotor	Total		
Coarse	9964	9128	19092	231.534	93.01
Medium	18366	17630	35996	527.16	93.18
Fine	22688	22642	45330	1132.59	93.21

The above table depicts the characteristics of three different meshes and it is evident that the computational time depends on the number of grids. Therefore, the mesh

with medium number of grids is considered for further simulation as the variation in the number of elements does not have a greater impact on the total-to-total efficiency.

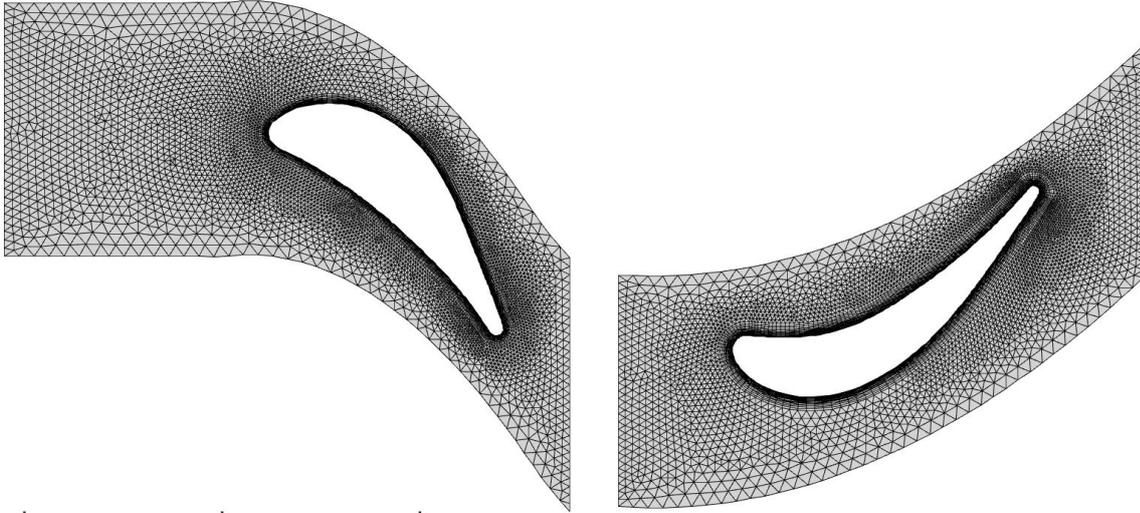


Figure 6.2: Mesh of the stator and rotor.

Note the mesh distribution at the leading and trailing of blades are shown in the Figure B.1 and Figure B.2 respectively.

### 6.1.3. Quality of the mesh

A high quality mesh is of higher-priority to trust the CFD results obtained. The kind of mesh generated is prone to higher skewness than the aspect ratio. Hence, the quality of the mesh is examined in GAMBIT and it is estimated based on the 'EquiAngle Skew ( $Q_{EAS}$ )' function. The EquiAngle Skew function is defined as,

$$Q_{EAS} = \max \left[ \frac{\theta_{max} - \theta_{eq}}{180 - \theta_{eq}}, \frac{\theta_{eq} - \theta_{min}}{\theta_{eq}} \right] \quad (6.1)$$

where  $\theta_{max}$  and  $\theta_{min}$  are the minimum and the maximum angles between the edge of the elements and  $\theta_{eq}$  represents the characteristic angle corresponding to an equilateral cell of similar form. As the mesh grids consist of quadrilateral and triangular shapes, the characteristic angle ( $\theta_{eq}$ ) also varies. For the triangular grid,  $\theta_{eq}$  value considered in GAMBIT is 60 degrees and for the quadrilateral grid,  $\theta_{eq}$  value is 90 degrees. EquiAngle Skew values ranges between 0 and 1. If  $Q_{EAS} = 0$  then it indicates that elements in the mesh are equilateral and  $Q_{EAS} = 1$  indicates completely degenerated mesh. The relationship between the EquiAngle Skew and the mesh quality is shown in the table below,

Table 6.2: Equiangle skewness Vs mesh quality.

$Q_{EAS}$	0	0-0.25	0.25 - 0.5	0.5-0.75	0.75-0.9	0.9-1	1
<b>Quality</b>	Equilateral	Excellent	Good	Fair	Poor	Very poor	Degenrate

In the stator, 95.87% of the elements have the EquiAngle Skew value in the range of 0-0.1. Similarly, in the rotor, 96.02% of the elements have the EquiAngle Skew value

in the range of 0-0.1. Based on the above mentioned Table 6.2, the stator and the rotor mesh are of high quality [38]. Ideally, in a mesh, the maximum change in grid spacing should not be greater than 20%. The change in grid size should be smooth to avoid unreliability in the simulation results. In the stator and rotor mesh, around 88.40% and 89.25% of cells are having grid spacing less than 20%.

## 6.2. Modelling assumptions

In this section, the assumptions made for building the model are explained. The entire simulation is performed for the first stage of the turbine section, of which the analysis is focused only at the 50% span of the turbine blades. Further assumptions are listed below:

- Fluid flow is modeled by using the steady-state RANS equation which implicitly states that the fluid is viscous and turbulent.
- Turbulence is modeled based on the Shear Stress Turbulence (SST) model.
- The entire simulation is performed on a two-dimensional domain.
- The working fluid in the turbine is considered as the ideal gas. The explanation for concluding with the ideal gas as the working medium is discussed in the following section.
- The stator and the rotor blades are considered to have no heat loss (adiabatic).

## 6.3. Equation of state

The working fluid in the turbine enters at high temperature and pressure and it consists of different gases. The mass fraction of each of the gases is given in the following table.

Table 6.3: Gas composition.

Gas Mixture	Mass fraction (%)
Argon	0.02
Carbon dioxide	0.085
Water	0.075
Nitrogen	0.715
Oxygen	0.105

From the mass fraction of given gases (Argon, carbon-di-oxide, water, nitrogen and oxygen), the number of moles of each gas is calculated as follows,

$$\text{Number of moles of each gas, } n_i = \frac{m_i}{M_i} \quad (6.2)$$

where  $m_i$  is mass (kg) and  $M_i$  is the molecular mass (kg/kmol) of the  $i^{th}$  gas component.

In order to estimate the total molecular mass ( $M_t$ ), the mole fraction ( $x$ ) is determined. The mole fraction is a ratio of the number of moles of  $i^{th}$  gas and the total number of the moles. The total molecular mass is calculated by summation of the mole fraction and molecular mass of each component. From this, the gas constant of the mixture ( $R$ ) is estimated by using the following formula,

$$R' = \frac{R_u}{M_t} = 294.34 \text{ J/kg.K} \quad (6.3)$$

where  $R_u$  is the Universal gas constant and  $M_t$  is the total molecular mass.

From the specified gas composition, the specific heat capacity ( $c_p$ ) of the mixture is estimated based on the NASA format [39]. As  $c_p$  is a function of temperature, the inlet temperature of the stage is considered as the temperature limit.

$$c_p = R' (a_1 + a_2T + a_3T^2 + a_4T^3 + a_5T^4) = \frac{\gamma R'}{\gamma - 1} \quad (6.4)$$

From the above mentioned equation,  $c_p$  for each component is determined and summed to estimate the specific heat capacity of the mixture ( $c_p = 1301.932 \text{ J/kg.K}$ ). The specific heat ratio ( $\gamma$ ) is calculated to be 1.293 by using the formula in Equation 6.4. The critical temperature of the mixture ( $T_{ct}$ ) is calculated by adding the product of mole fraction ( $x_i$ ) and critical temperature ( $T_c$ ) of each component. Similar procedure is also followed for estimating the critical pressure ( $p_{ct}$ ). The reduced temperature ( $T_r$ ) and pressure ( $p_r$ ) are calculated as specified in the equation to identify the fluid model from the compressibility chart.

$$T_r = \frac{T}{T_{ct}} = 6.65; \quad p_r = \frac{p}{p_{ct}} = 0.22 \quad (6.5)$$

where  $T$  and  $p$  are the inlet temperature and pressure respectively.

If the value of compressibility factor ( $Z$ ) is equal to 1, the fluid is considered to behave like an ideal gas. Based on the estimated values of  $T_c$  and  $p_{ct}$ , looking into the compressibility chart (Figure C.1), it is clear that compressibility factor ( $Z$ ) is greater than 1. This brings to the assumption that given gas mixture in the turbine behaves like an ideal gas.

Table 6.4: Equation of state parameters for the ideal gas.

Parameters	Values	Units
$R'$	294.34	J/kg K
$\gamma$	1.293	—
$T_{ct}$	200.1774	K
$p_{ct}$	5.984978	MPa

## 6.4. CFD result of baseline stage

The main specification of the axial turbine is : The flow solver in *SU2* is used to solve the steady-state Reynolds Averaged Navier-Stokes equations. The flow simulation for

Table 6.5: Inlet and outlet boundary conditions.

Parameters	Values	Units
Inlet total pressure, $p_{tot,in}$	1.1	MPa
Inlet total temperature, $T_{tot,in}$	1333.15	K
Outlet static pressure, $p_{s,out}$	0.54	MPa
Working fluid	Ideal gas	-

the stage is marched in time using an implicit Euler scheme with the CFL number equal to 10. The simulation is carried out until it satisfies the residual convergence criteria of fifth order of magnitude ( $10^{-5}$ ). The results from the stage simulation are discussed in the following passage.

The pressure distribution in the stage (Figure 6.3) shows that the high pressure fluid enters the stator in the axial direction along the positive x-axis at zero degree angle of incidence. The streamline of the fluid divides at the stagnation point which corresponds to the leading edge of the blade with part of the fluid moving along the suction side and remaining along the pressure side. The flow accelerates along the suction side of the stator and it progresses through the rotor. A low pressure region is encountered because of the high velocity gradient on the suction side in both the blades.



Figure 6.3: Static pressure distribution in the stage.

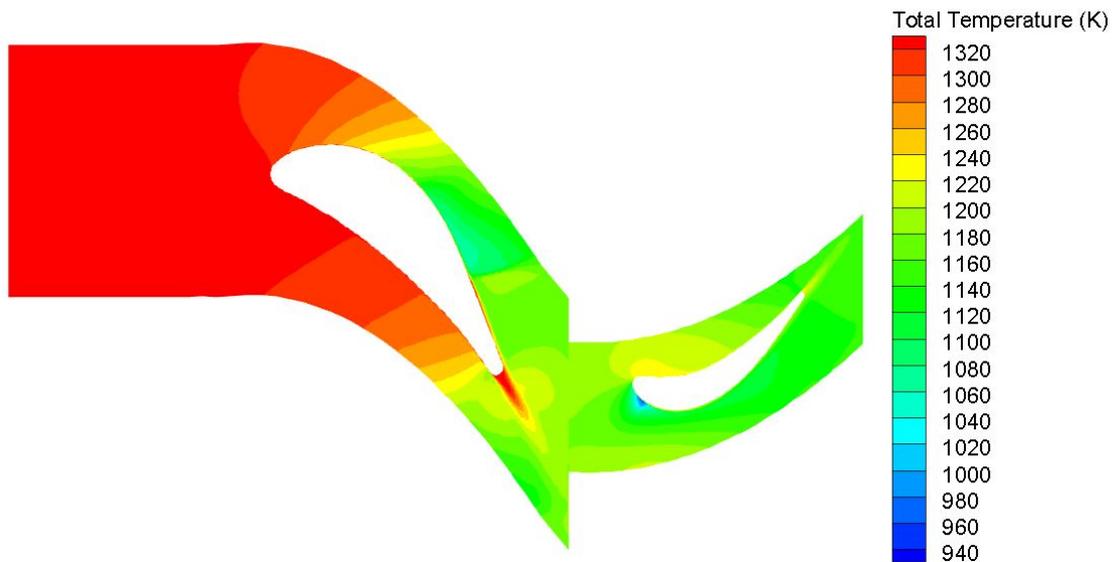


Figure 6.4: Temperature distribution in the stage.

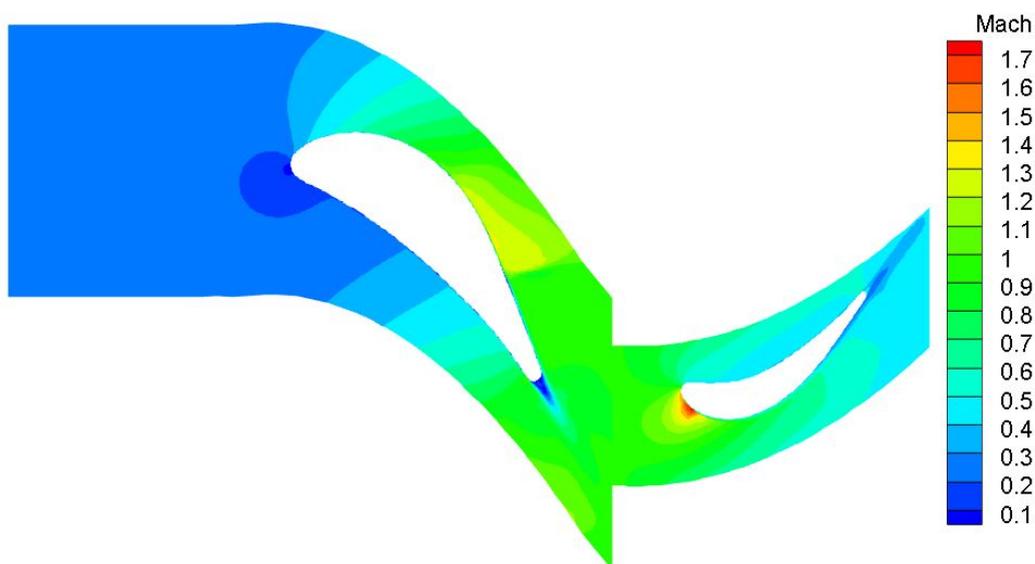


Figure 6.5: Mach number distribution in the stage.

The temperature distribution in the stage is shown in the Figure 6.4. In the stator, there is a sudden increase in the temperature on the suction side which implicitly represents strong shock wave formation. Figure 6.5 shows the absolute Mach number contours of the stator and rotor. The flow is subsonic at the inlet of the stator and at the outlet of the rotor. The absolute Mach number on the suction surface of the stator goes up to a maximum of 1.276 whereas in the rotor it reaches up to 1.704. As the rotor is the moving body, the relative Mach number distribution is shown in the Figure 6.24. The velocity distribution around blades is shown in the Figure 6.6. The absolute velocity is calculated for the stationary stator and the relative velocity is estimated for the rotor to account for the translation motion.



Figure 6.6: Velocity distribution in the stage

Figure 6.6 depicts that flow in the region of the leading edge is subjected to stagnation. The laminar boundary layer, proceeding from the point of stagnation over the leading edge, undergoes separation just downstream of the point of minimum pressure. A small distance downstream of the separation point where the transition to turbulent flow occurs in the free stream layer. The flow then reattaches to the blade surface by forming a turbulent boundary layer that extends from the reattachment point to the trailing edge. Depending on the thickness of boundary-layer, velocity increases with the change in all flow properties and generates the entropy. Whenever the fluid is subjected to the high rate of shear strain, entropy is generated because of the viscous shear. A relatively high rate of shear occurs in the wake region and at the edges of flow separation regions (Figure 6.7).

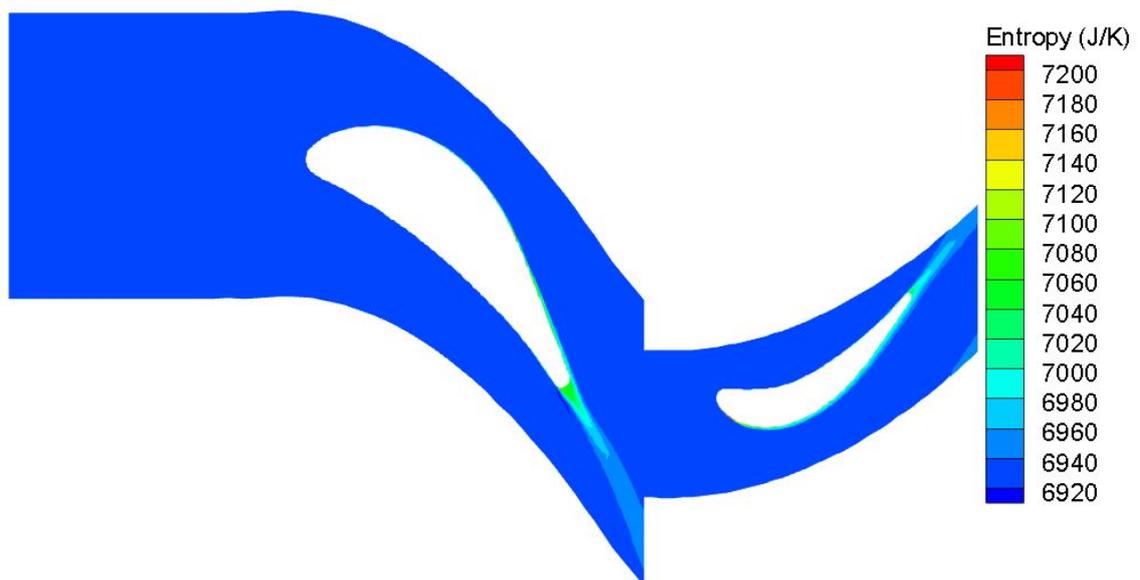


Figure 6.7: Entropy distribution in the stage.

According to Denton (1993) [5], the entropy generation is dominant on the suction side and the regions of higher velocity contribute to high amount of losses. This is evident in the Figure 6.7, where the entropy generation is more on the suction side of the blades. As the main aim of this thesis is to reduce the entropy generation, the optimization is carried out to minimize the entropy generation in these regions.

## 6.5. Results of the optimized blade shapes

The optimization for the stator and the rotor are performed separately as *SU2* is yet to develop the stage optimization feature. The optimization methodology is already illustrated in the Figure 5.1. The same procedure is followed for both the blades. The results of optimized blades are discussed in this section.

### 6.5.1. Optimization of the stator

The operating conditions for the stator are mentioned in the Table 6.6. The flow simulation around the stator shows (Figure 6.12) that the flow reaches a maximum Mach number of 1.276 at around 80% of the axial chord length on the suction side. After this region, a shock wave is generated causing a sudden increase in the boundary-layer thickness at the rear end of the suction side, which consequently results in wake formation. Therefore, to improve the performance of the stator, optimization is implemented to reduce the entropy generation (objective function) with the mass flow rate as the constraint. The calculated baseline mass flow rates at the inlet and outlet of the stator are 439.86 kg/s and 439.86 kg/s respectively. As inequality constraint of the mass flow rate is imposed at both the inlet and outlet of the stator, the optimization is implemented for different constraints to find an optimal shape with a maximum reduced entropy generation.

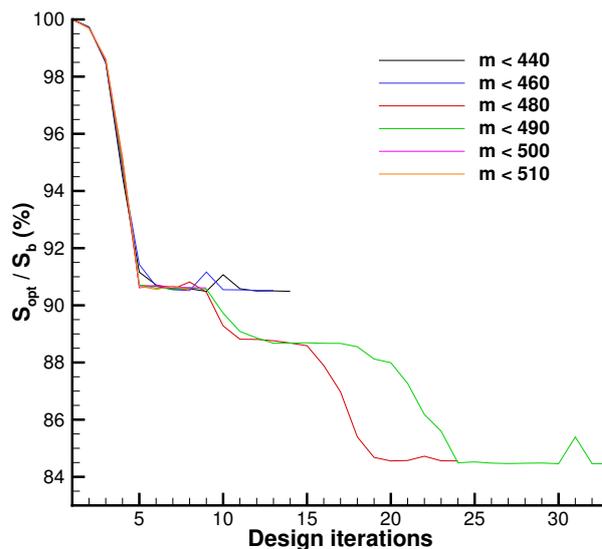


Figure 6.8: Reduction in the entropy generation for different mass flow rates (kg/s) as constraint ('m' indicate mass flow rate at the inlet and outlet).

Table 6.6: The operating conditions of the stator and the rotor.

	$p_{tot,in}(MPa)$	$T_{tot,in}(K)$	$p_{s,out}(MPa)$
<b>Stator</b>	1.1	1333.15	0.63
<b>Rotor</b>	1.075	1332.7	0.54

It is evident from the Figure 6.8 that with the mass flow rate less than 490 kg/s, the entropy generation is much reduced compared to the other cases. This particular optimization result is considered for further analysis. The design optimization is performed with 25 FFD control points (Figure 6.9) and the control points are mapped on the blade surface. Of the 25 control points (CPs), only 20 of them are considered as the design variables. The remaining five points are fixed close to the trailing edge to prevent an unfeasible shape. The added advantage of fixing these points ensure acceptable blade thickness without imposing a geometrical constraint [23].

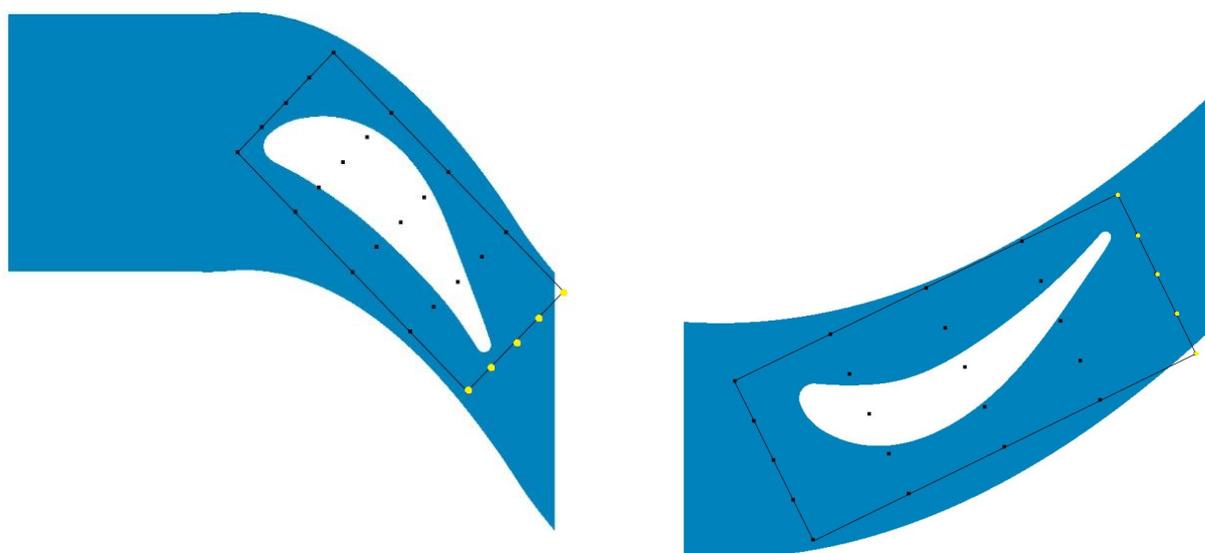


Figure 6.9: Free Form Deformation (FFD) box with 25 CPs (design variables) in the stator (left) and the rotor (right); the fixed points are specified in yellow colour.

Figure 6.8 shows that the design optimization converges within 33 iterations with around 16% decrease in the entropy generation with respect to the baseline value. The change in mass flow rate at the inlet and outlet of the stator is around 0.2% increase with respect to the baseline design and hence, the inequality constraint is satisfied.

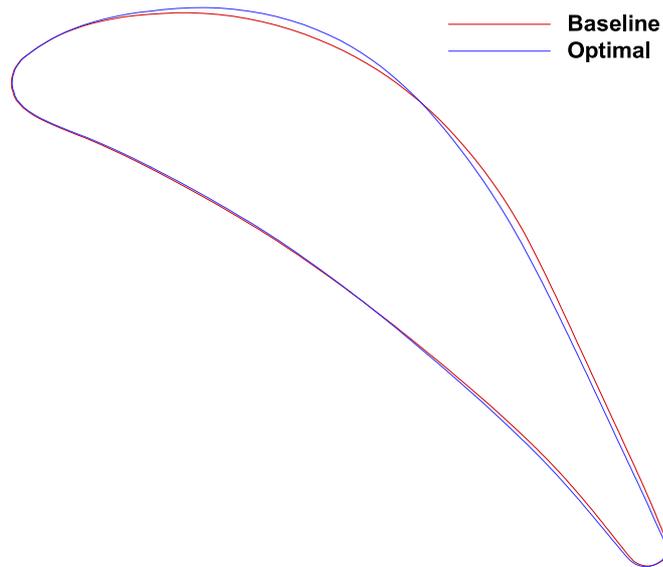


Figure 6.10: Baseline and optimal geometry of the stator.

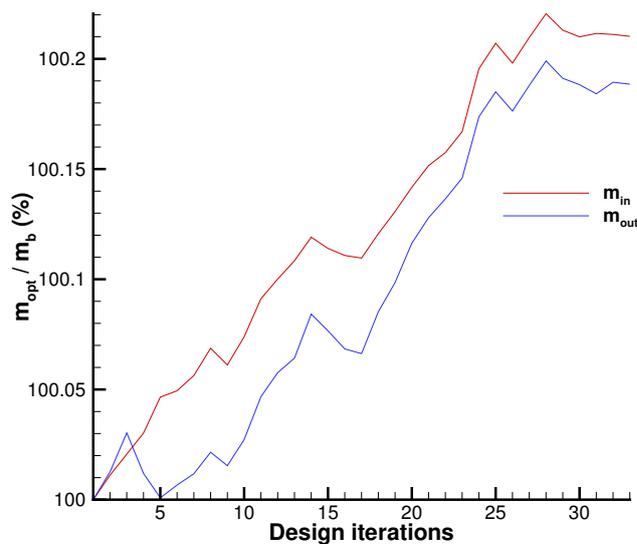


Figure 6.11: Variation of mass flow rate of the optimal shape w.r.t baseline shape at each design iteration.

The optimized blade shape is compared with the baseline shape in the Figure 6.10. The point where the baseline and optimal blade shape coincides is the throat area which determines the maximum mass flow rate. In this case, the assumed mass flow rate is 12% more than the initial value but the optimized case results in just 0.2% increase in the mass flow rate with respect to the initial value (Figure 6.11). This is because, the optimizer has maintained the throat area constant to control the mass flow rate and satisfy the constraint.

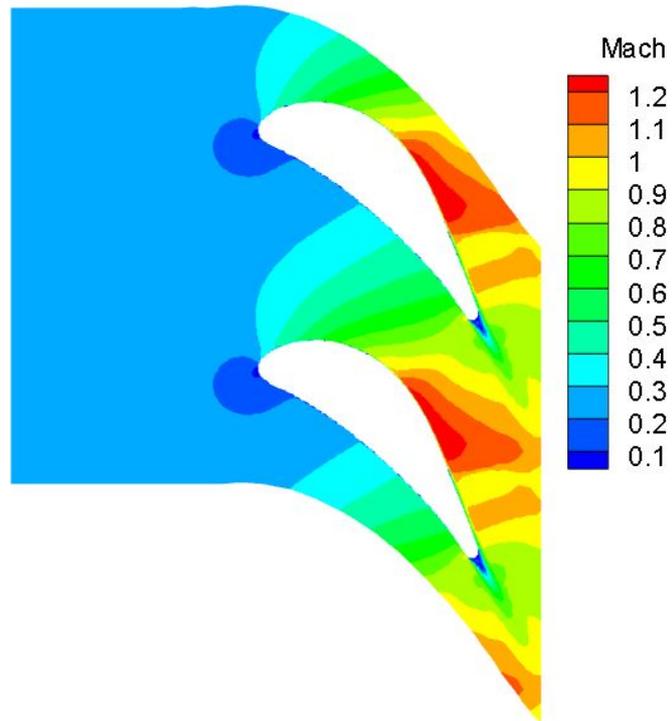


Figure 6.12: Mach number distribution in the baseline design.

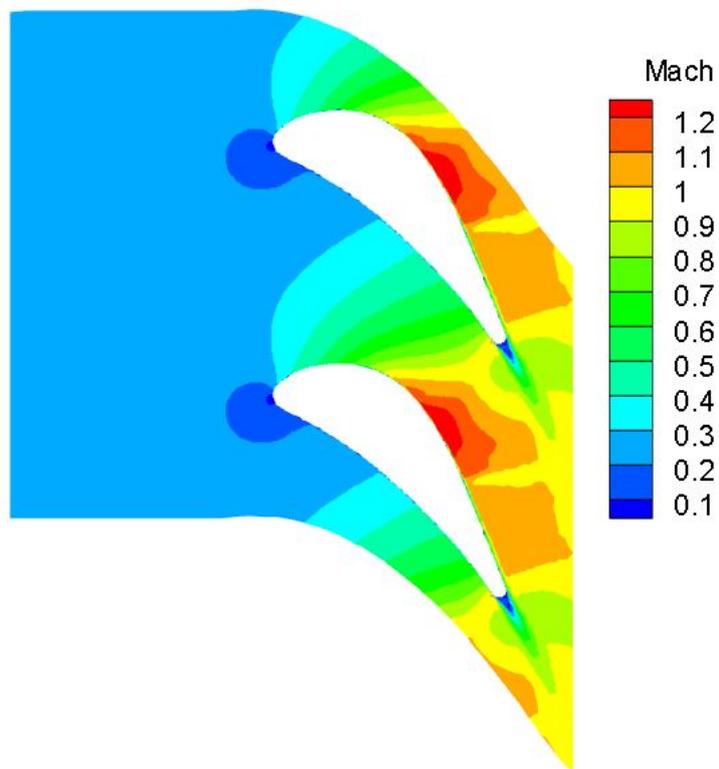


Figure 6.13: Mach number distribution in the optimal design.

In the Mach number distribution (Figure 6.13) of the optimal geometry, the onset of the supersonic flow is advanced compared to the baseline geometry. By taking a closer look in the Figure 6.10, the increase in blade thickness in the optimized shape after the leading edge has decreased the flow passage area which eventually makes the flow to accelerate upstream. The decrease in the blade thickness after the throat area leads to smoother flow deceleration by weakening the strength of the shock.

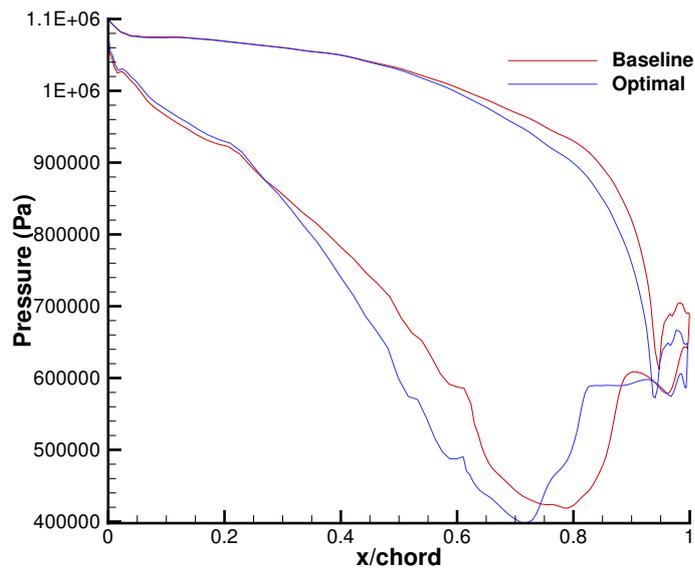


Figure 6.14: Pressure distribution around the baseline and optimal geometry.

The pressure distribution around the baseline and optimal shape (Figure 6.14) shows that the velocity peak on the suction side of the optimal design appears upstream with the gradual expansion of flow at rear-end of the suction side (after 80% of the axial chord length of the stator). On comparing the entropy distribution in the baseline and optimal shape, there is no significant change visible but the entropy generation is reduced.

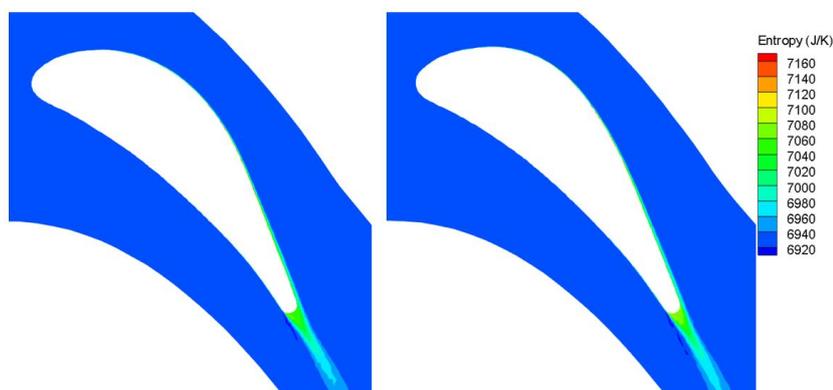


Figure 6.15: Entropy distribution in the baseline (left) and optimal (right) stator.

### 6.5.2. Optimization of the rotor

The same shape optimization methodology is followed for the rotor, except the steady state translation is considered to account for the rotation. The outlet conditions from the optimized stator blade (total temperature, total pressure and inlet flow angle) is considered as the inlet for the baseline rotor. With these operating conditions (refer Table 6.6), flow simulation is conducted for the rotor. It is encountered that the flow reaches a maximum Mach number of 1.704 after the leading edge on the suction side (Figure 6.20). A sudden shock appears in this region causing an increase in the boundary layer thickness which eventually leads to the flow separation. In order to improve the performance of the rotor, optimization is performed to minimize the entropy generation with the mass flow rate less than 490 kg/s as the constraint. The same mass flow rate constraints are implemented for the stator and rotor to maintain consistency in the solution. The evaluated baseline mass flow rates at the inlet and outlet section are 439.28 kg/s and 439.99 kg/s respectively. The optimization is performed with the FFD control points as the design variables (Figure 6.9). The same pattern of the design variables, as explained in the stator optimization, are chosen for this case.

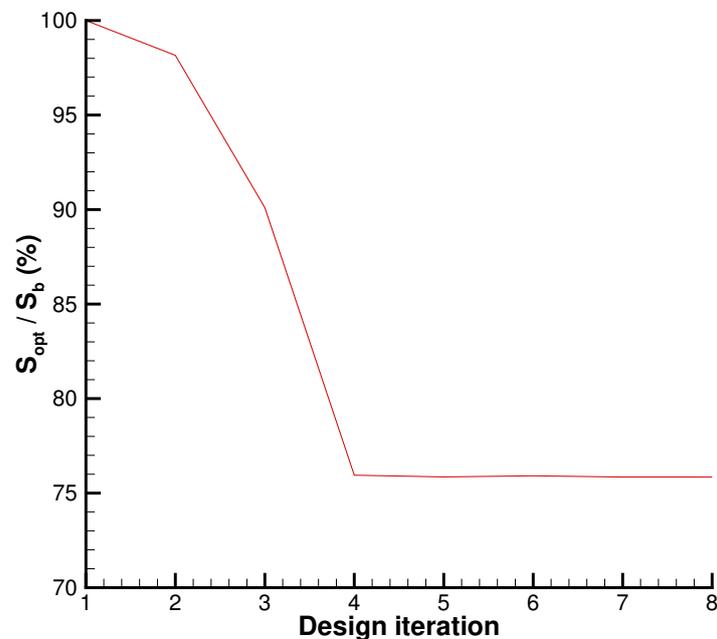


Figure 6.16: Variation of entropy generation of the optimal design w.r.t baseline design at each design iteration.

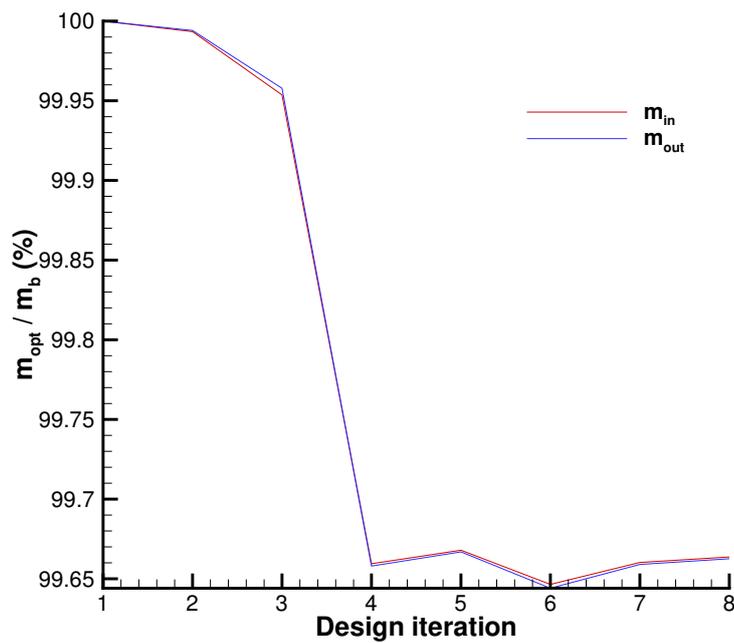


Figure 6.17: Variation of mass flow rate of the optimal design w.r.t baseline at each design iteration.

The optimization process converges within 4 design iterations with approximately 24% decrease in the entropy generation (Figure 6.16). Furthermore, for the given inequality constraints, there is about 0.4% decrease in the mass flow rate at the inlet and outlet with respect to the baseline design. Thus, the imposed constraint is satisfied.

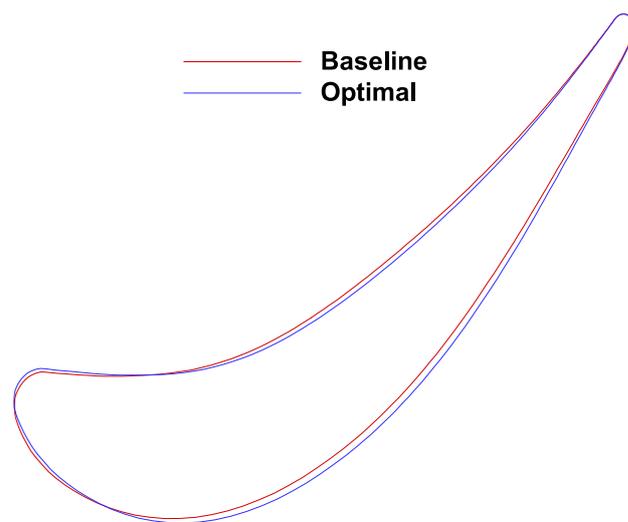


Figure 6.18: Baseline and optimal geometry of the rotor

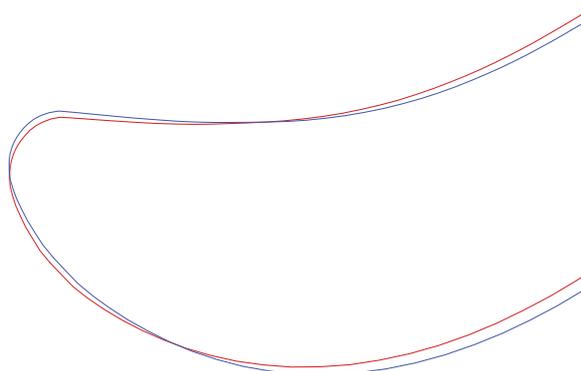


Figure 6.19: Zoomed in view of the leading edge of the rotor - baseline (red) and optimal (blue).

The throat area of the optimal and baseline design of the rotor is maintained constant to fulfill the constraints. In the baseline design (Figure 6.20), the flow simulation shows that the flow is over accelerated immediately after the leading edge which leads to strong shock formation and eventually leads to boundary layer separation.

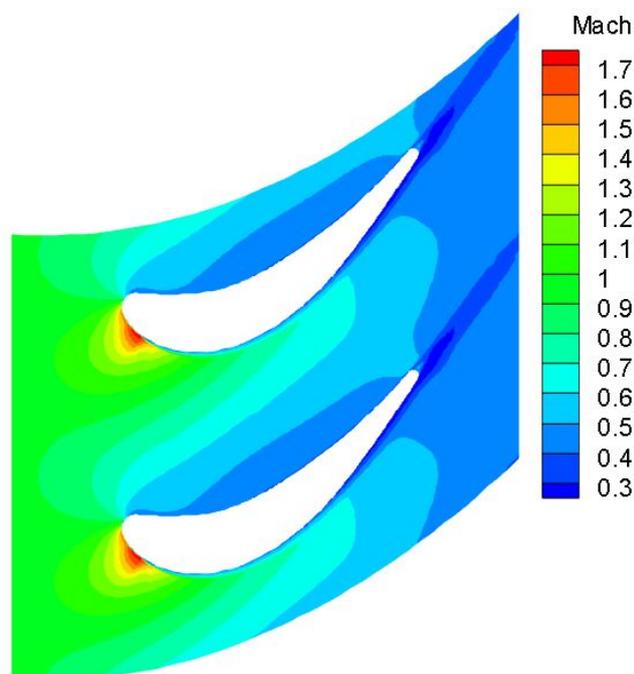


Figure 6.20: Mach number distribution in the baseline geometry.

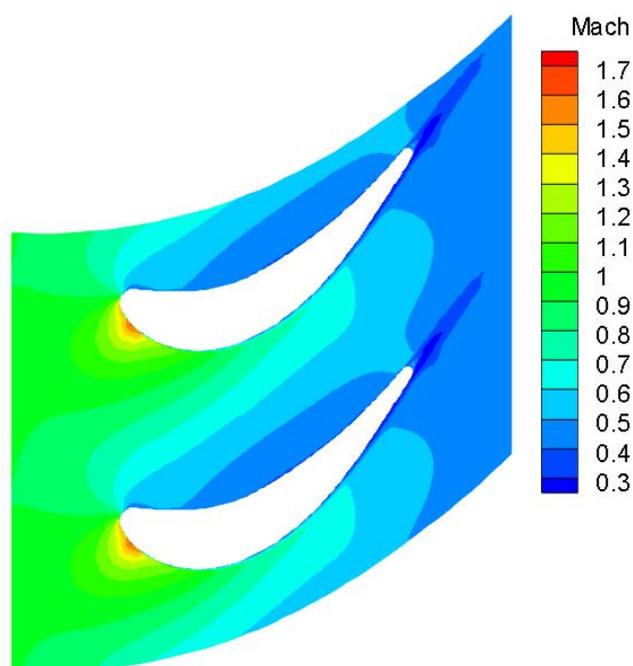


Figure 6.21: Mach number distribution in the optimal geometry.

From the Figure 6.21, it is evident that the shock wave is reduced on the suction side of the re-designed shape. The optimized shape has significantly reduced the flow separation by decreasing the boundary layer thickness (Figure 6.22). Both the baseline and the optimal configuration show similar flow behavior on pressure side of the blade, this ensures the fact that most of the losses in the transonic cascade is caused due to the adverse pressure gradient on the rear end of the suction side.

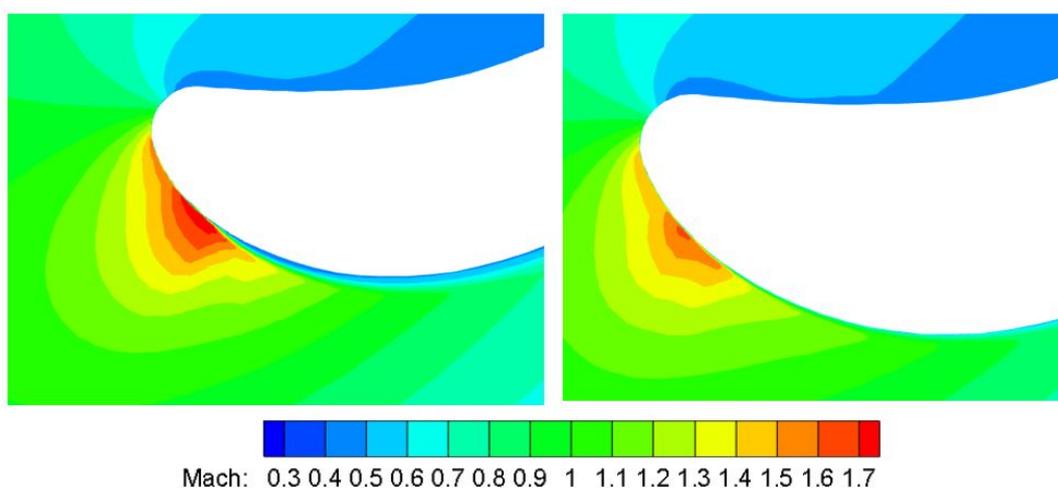


Figure 6.22: Boundary layer separation in the baseline (left) and optimal (right) geometry.

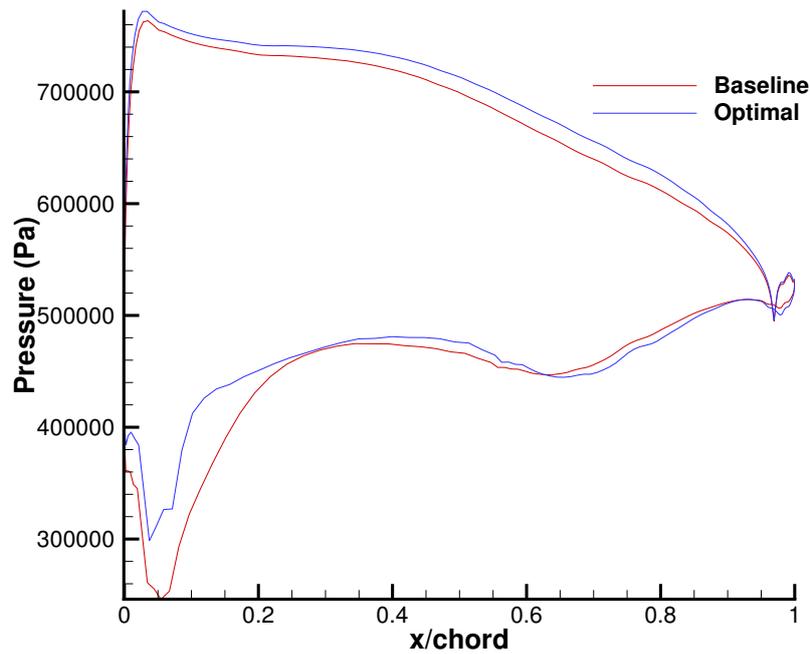


Figure 6.23: Pressure distribution around the baseline and optimal geometry of the rotor.

The pressure distribution around the blades show that the velocity peak is reduced in the optimal shape which leads to the decrease in the over acceleration after the leading edge. The optimized shape provides much more uniform flow at the outlet.

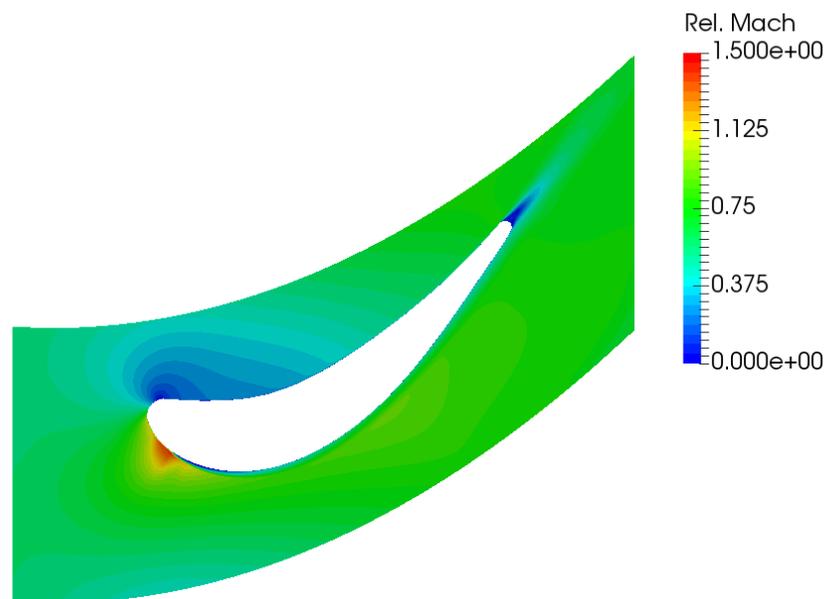


Figure 6.24: Relative Mach number distribution in the baseline design.

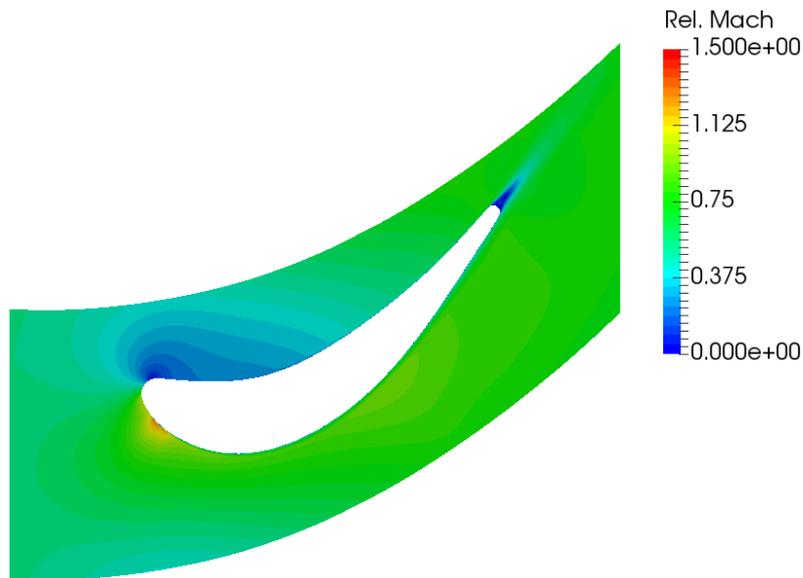


Figure 6.25: Relative Mach number distribution in the optimal design.

The relative Mach number distribution in the optimized blade shows decrease in the shock wave on the suction side and also displays uniform flow by reducing the boundary layer separation. The distribution of entropy in the baseline and optimized shape of the rotor are shown in the Figure 6.26. The optimized rotor shows notable decrease in the entropy on the suction side.

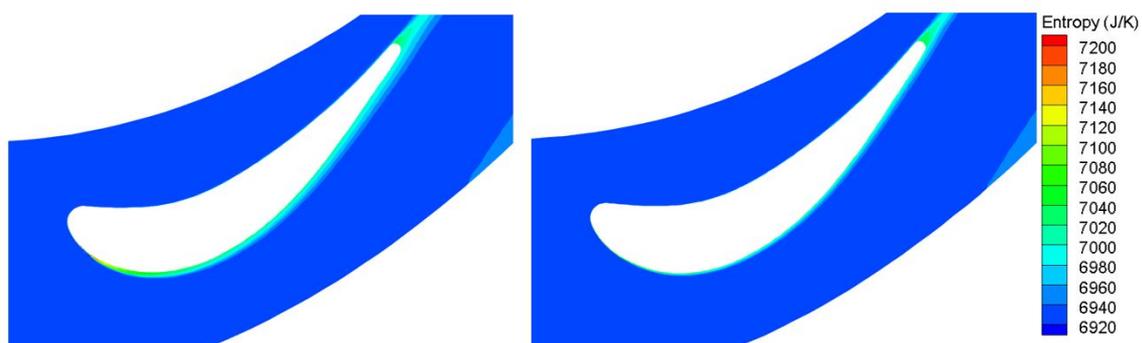


Figure 6.26: Entropy distribution in the baseline (left) and optimal (right) shape of the rotor.

## 6.6. Results of optimized blades - Stage simulation

The optimized stator and rotor are simulated together as a stage to analyze the overall performance. The flow simulation for the optimized case is performed with the same operating conditions specified in Table 6.5.

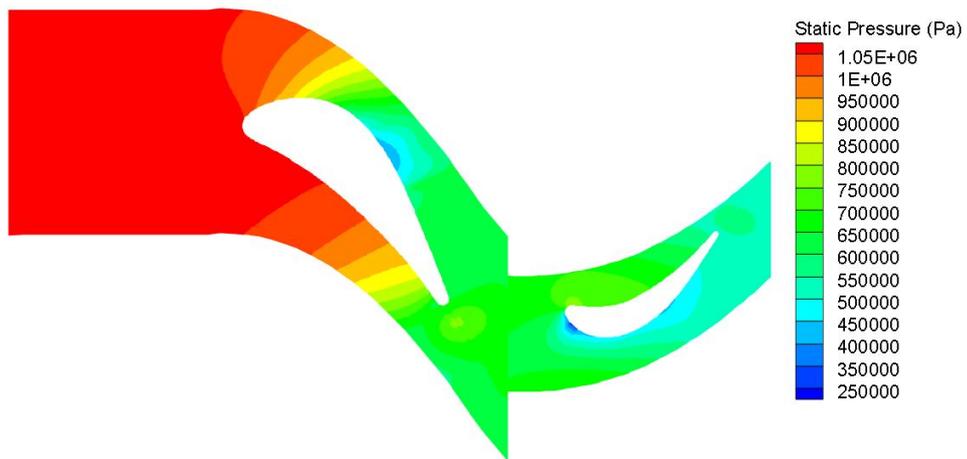


Figure 6.27: Static pressure distribution in the optimized stage.

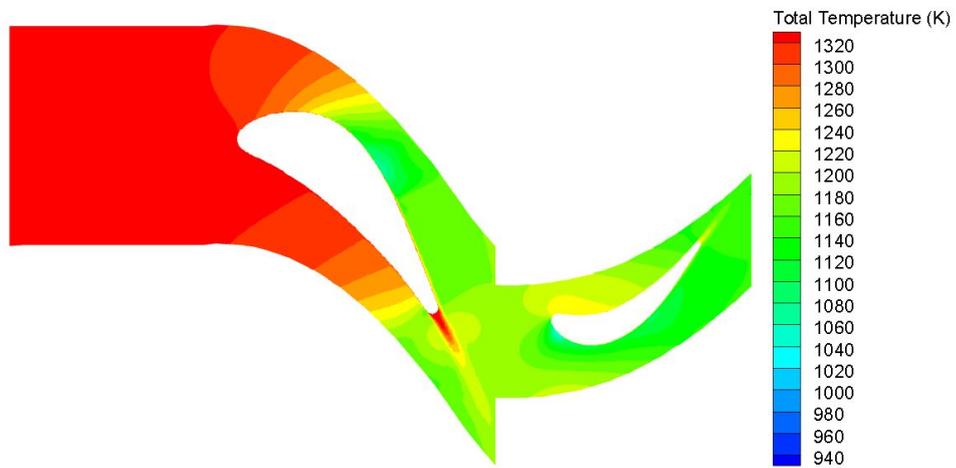


Figure 6.28: Total temperature distribution in the optimized stage.

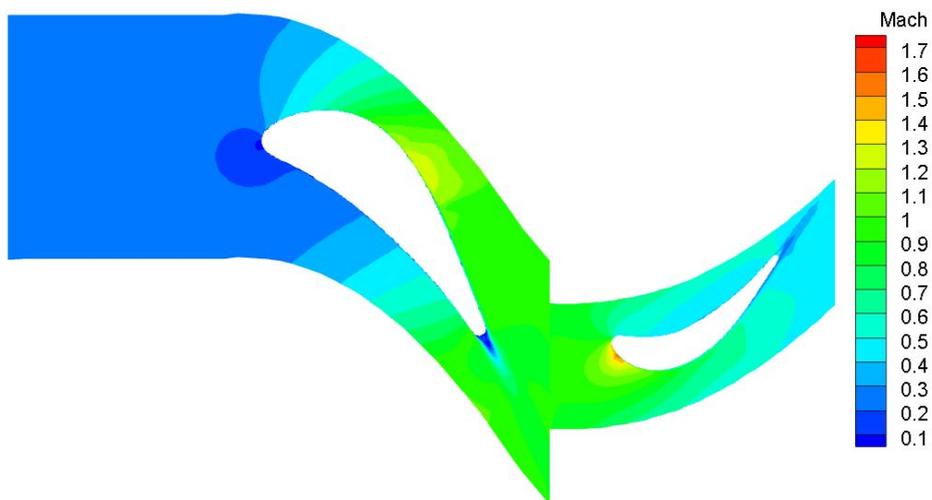


Figure 6.29: Absolute Mach number distribution in the optimized stage.

On comparing the outcomes of the baseline stage with the optimal stage (refer Section 6.4), it is evident, from the temperature (Figure 6.28) and Mach number (Figure 6.29) distribution contours of the optimized stage, that there is significant decrease in the intensity of shock and boundary layer separation.

The thermodynamic properties of the baseline and optimal stage at the inlet, outlet and interface are obtained from the flow simulation results (refer Table 6.7 and Table 6.8). The inlet and outlet thermodynamic properties are almost constant because these are the specified operating conditions in the flow solver. The blade shape variations have changed the interface condition compared to the baseline stage properties. This variation has apparently improved the efficiency of the stage.

Table 6.7: Thermodynamic properties of the baseline stage.

	$p_{tot}$ (MPa)	$T_{tot}$ (K)	$p_s$ (MPa)	$H$ (MJ)
<b>Inlet</b>	1.10	1333.09	1.06	1.72
<b>Interface</b>	1.07	1332.86	0.64	1.54
<b>Outlet</b>	0.61	1178.82	0.54	1.49

Table 6.8: Thermodynamic properties of the optimal stage.

	$p_{tot}$ (MPa)	$T_{tot}$ (K)	$p_s$ (MPa)	$H$ (MJ)
<b>Inlet</b>	1.10	1333.09	1.06	1.72
<b>Interface</b>	1.08	1332.85	0.65	1.55
<b>Outlet</b>	0.62	1177.80	0.54	1.49

Table 6.9: Comparison of the baseline and the optimized stage.

	$\eta_{tt}$ (%)	$\eta_{ts}$ (%)	$S_{gen}$ (%)	DOR
<b>Baseline</b>	93.18	77.72	3.33	0.22
<b>Optimal</b>	94.49	78.20	2.67	0.26

By comparing the flow simulation results of the optimized stage with the baseline stage in Table 6.9, it is obvious that the proposed design has reduced the losses and eventually leads to the reduction of the entropy generation of about 20%. The optimized blades have resulted in about 1.4% increase in the total-to-total efficiency ( $\eta_{tt}$ ) and consequently, the degree of reaction (DOR) is increased from 0.22 to 0.26. The degree of reaction is defined as the ratio between the enthalpy drop in the rotor and the enthalpy drop in the stage. For the stage simulation, there is around 0.1% increase in the mass flow rate of the optimized stage compared to the baseline stage (Figure D.1). Generally, the stator determines the mass flow rate of the stage. It is anticipated that there should be 0.2% increase in the mass flow rate of the stage as per the results shown in the Figure 6.11. On the contrary, there is only 0.1% increase in stage. This is due to the variation in the interface condition of the optimized stage compared to the baseline stage simulation. This discrepancy in the variation of flow properties can be avoided, if the stator and rotor are optimized simultaneously.



# 7

## Conclusions and Recommendations

The present work reports the adjoint-based aerodynamic shape optimization for an axial turbine. The scope of the thesis is to optimize the Siemens SGT5-2000E turbine section at the mid-span to improve the aerodynamic efficiency. This is done by performing shape optimization to reduce the entropy generation with mass flow rate as the constraint.

The gradient-based optimization method is used as it is efficient in terms of the computational power and design space availability. This method is capable of determining the optimal shape in a few design iterations, provided an effective method for gradient evaluation is used. Normally, the aerodynamic design in the turbomachinery involves a large number of design variables with an objective function to be minimized. For these cases, the adjoint method is more suitable for evaluating the gradients of the objective function with respect to design variables. The cost of the gradient evaluation is of the same order of magnitude of the objective function and it is independent of the number of design variables.

The design optimization is implemented in an open-source CFD software, Stanford University Unstructured (*SU2*), with the state-of-the-art adjoint-based optimization technique. The tools for performing the flow simulation, gradient evaluation, surface parameterization, mesh deformation and an optimizer with search algorithm are available in *SU2*. The optimization procedure starts by evaluating the objective function (entropy generation) of the baseline design by using the flow solver. The adjoint solver employs the objective function and the constraints to compute the gradients with respect to the design variables. The Free-Form Deformation (FFD) approach is used to parameterize the blade geometry. Each of the control points in the FFD box is varied to improve the baseline shape and the gradients are utilized to determine an optimal search direction. To avoid re-meshing at each design step, the mesh deformation tool with the linear elasticity method is used to propagate the surface deformation to the entire mesh. The algorithm is structured to iterate until an optimized shape with reduced entropy generation is found.

### Concluding remarks

The adjoint-based optimization for the stator and the rotor is implemented successfully. The above-mentioned optimization procedure is implemented for the stator. The same optimization procedure is implemented for the rotor, except outlet total conditions of the optimized stator are used as the inlet operating conditions of the rotor. On comparing the results of the baseline and optimized design, it can be inferred that:

- In the optimized stator, the intensity of shock waves are reduced and there is a gradual expansion of flow on the suction side. The reduction in the entropy generation in the stator is approximately 16%.
- In the optimized rotor, the boundary layer separation on the suction side is reduced and shock losses are minimized. The reduction in the entropy generation of the rotor is approximately 24%.
- In the optimized stage, the entropy generation is reduced by 20%. This eventually leads to the increase in the efficiency of about 1.4%. Also, there is an increase in the degree of reaction as a consequence of blade shape optimization.

## 7.1. Recommendations

The present work is still at the initial phase. The scope of the work can be further improved by considering the following recommendations.

- It is suggested to perform the shape optimization for the stator and the rotor simultaneously to maintain the consistency in the flow properties at the interface. The stage optimization can be performed with two Free-Form Deformation boxes on the stator and the rotor as mentioned in Albring *et al.* [40]. This can be performed in *SU2* itself, when the stage optimization feature is updated.
- The shape optimization can be further implemented by increasing the number of design variables for smoother deformation.



## Adjoint method formulation

The analytical derivation of the discrete adjoint method and the grid deformation are explained briefly in this section. The computation of the sensitive derivative of the objective function will be solved on the unstructured mesh. The objective function to be minimized is represented as  $J$ . The objective function depends on the flow variables  $U$ , grid points  $X$  and the physical ( $\alpha_p$ ) and geometrical ( $\alpha_g$ ) design variables. The flow variables ( $U$ ) depends on both the physical and geometrical variables but the grid points ( $X$ ) depends only on the geometrical design variables. The objective function  $J$  is expressed as,

$$J = J[\alpha_p, U(\alpha_p, \alpha_g), X(\alpha_g)] \quad (\text{A.1})$$

The relationship between the flow variables and the design variables can be computed by the nonlinear, steady-state governing equation and it is represented as follows,

$$R[\alpha_p, U(\alpha_p, \alpha_g), X(\alpha_g)] = 0 \quad (\text{A.2})$$

where the  $R$  is the residual of the discretized flow equations.

In the design process, usually the physical design variables ( $\alpha_p$ ) are not considered and hence, they are neglected. The gradient of objective function with respect to geometrical design variable defined as,

$$\frac{dJ}{d\alpha_g} = \frac{\partial J}{\partial U} \frac{\partial U}{\partial \alpha_g} + \frac{\partial J}{\partial X} \frac{\partial X}{\partial \alpha_g} \quad (\text{A.3})$$

Similarly, the total derivative of the residual of the discretized flow equations with respect to the geometrical design variables ( $\alpha_g$ ) is,

$$\frac{dR}{d\alpha_g} = \frac{\partial R}{\partial U} \frac{\partial U}{\partial \alpha_g} + \frac{\partial R}{\partial X} \frac{\partial X}{\partial \alpha_g} = 0 \quad (\text{A.4})$$

The sensitivity derivative  $\frac{\partial U}{\partial \alpha_g}$  in the Equation (A.3) is obtained by further simplifying Equation (A.4) and it is rewritten as,

$$\frac{\partial U}{\partial \alpha_g} = -\left(\frac{\partial R}{\partial U}\right)^{-1} \left[\frac{\partial R}{\partial X} \frac{\partial X}{\partial \alpha_g}\right] \quad (\text{A.5})$$

Equation (A.4) implies that each design variable needs the linearized flow equation to be solved every time. This makes it not advisable to use the finite difference as it will consume lot of memory and computational time. The adjoint method are especially architected to deal with a large number of design variables as the influence of the design variables on an objective function by the flow variables are decoupled. It implicitly explains that the adjoint method is independent of the number of design variables. The dependency of the objective function sensitivity on the flow variable sensitivity ( $\partial U/\partial \alpha_g$ ) is eliminated by substituting the Equation (A.5) in Equation (A.3). This is expressed as,

$$\frac{dJ}{d\alpha_g} = -\frac{\partial J}{\partial U} \left( \frac{\partial R}{\partial U} \right)^{-1} \left[ \frac{\partial R}{\partial X} \frac{\partial X}{\partial \alpha_g} \right] + \left[ \frac{\partial J}{\partial X} \frac{\partial X}{\partial \alpha_g} \right] \quad (\text{A.6})$$

In the above equation, the term  $-\frac{\partial J}{\partial U} \left( \frac{\partial R}{\partial U} \right)^{-1}$  represents the adjoint variable vector.

The adjoint variables ( $\lambda$ ) are consequently calculated from the solution of the adjoint system and the equation is written as,

$$\left( \frac{\partial R}{\partial U} \right)^T \lambda = -\left( \frac{\partial J}{\partial U} \right)^T \quad (\text{A.7})$$

Equation (A.7) illustrates the discrete form of adjoint equation and it can be rewritten as,

$$\lambda^T = -\left( \frac{\partial J}{\partial U} \right) \left( \frac{\partial R}{\partial U} \right)^{-1} \quad (\text{A.8})$$

Further above equation is substituted into Equation (A.6), the gradient of the objective function with respect to the design variable is,

$$\frac{dJ}{d\alpha_g} = \frac{dX}{d\alpha_g} \left[ \frac{\partial J}{\partial X} + \lambda^T \frac{\partial R}{\partial X} \right] \quad (\text{A.9})$$

# B

## Mesh around leading and trailing edge.

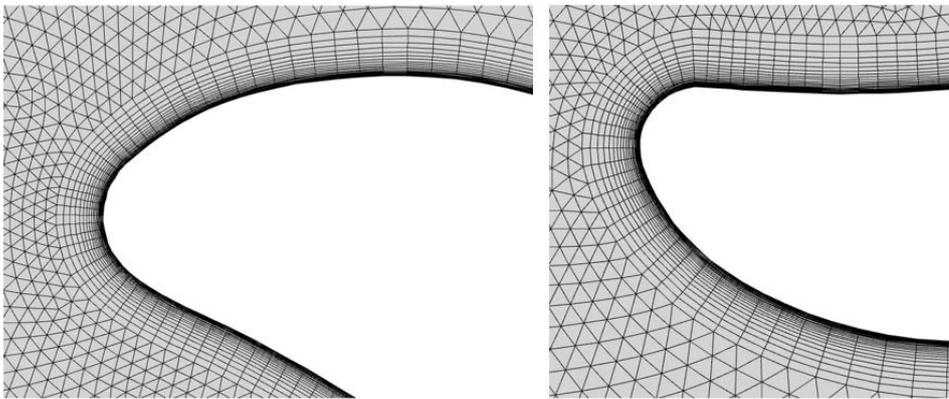


Figure B.1: Zoomed in view of the leading edge of the stator (left) and rotor (right).

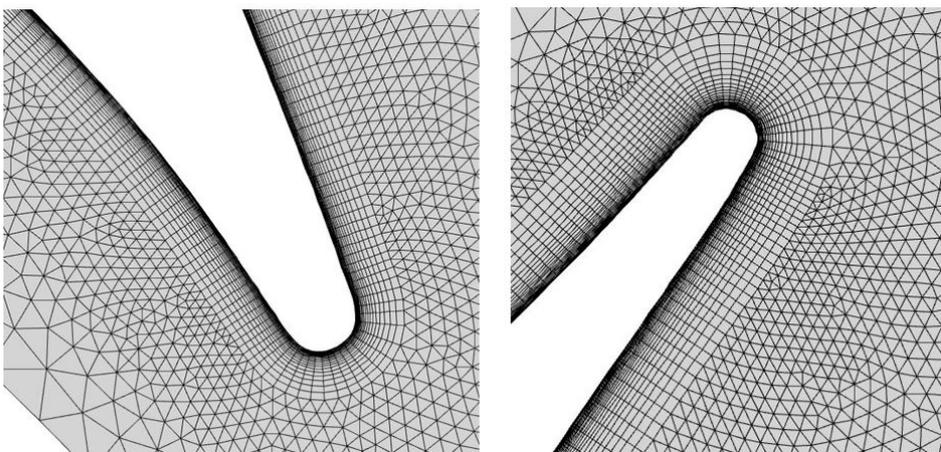


Figure B.2: Zoomed in view of the trailing edge of the stator (left) and rotor (right).



# C

## Compressibility factor chart

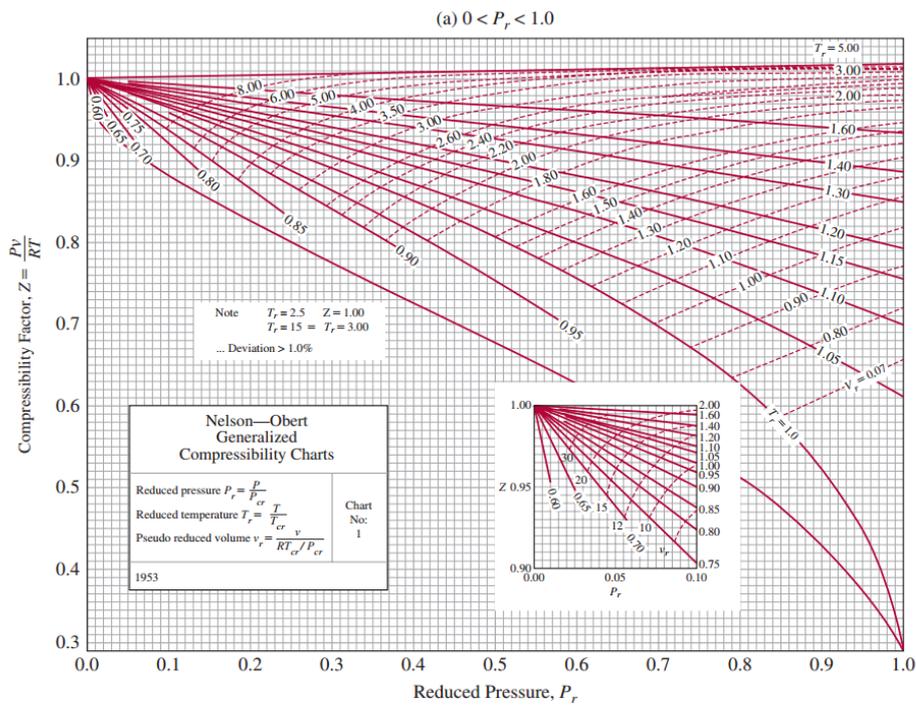


Figure C.1: Compressibility factor chart [9].



# D

## Variation in the mass flow rate

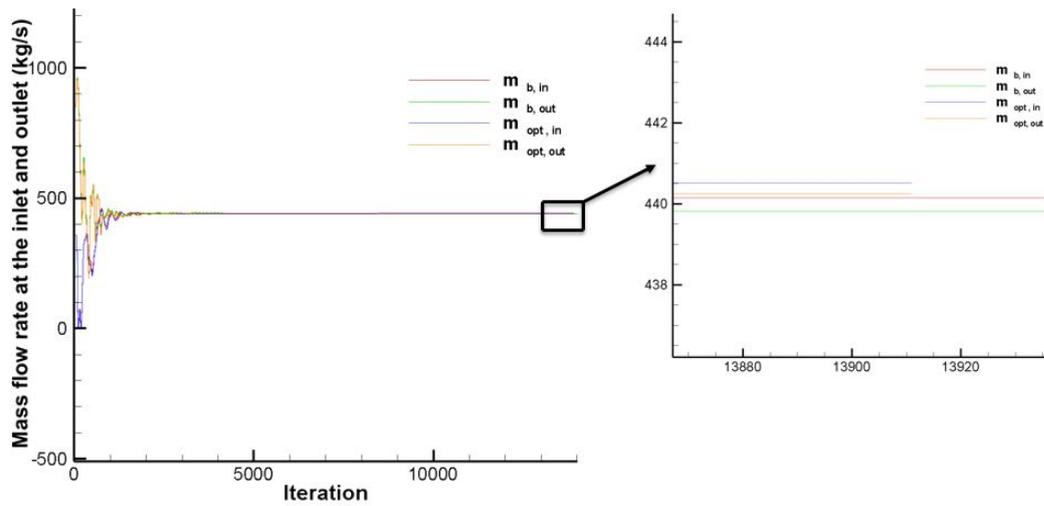


Figure D.1: Variation in the mass flow rate of the optimized stage w.r.t baseline stage.



# Bibliography

- [1] C. Zhang and Z. Feng, *Aerodynamic shape design optimization for turbomachinery cascade based on discrete adjoint method*, *Proceedings of ASME Turbo Expo 2011 GT2011*, (2011).
- [2] T. W. Sederberg and S. R. Parry, *Free-form deformation of solid geometric models*, *ACM SIGGRAPH computer graphics* (1986).
- [3] S. L. Dixon and C. Hall, *Fluid mechanics and thermodynamics of turbomachinery* (2013).
- [4] S. Acharya and G. Mahmood, *Turbine Blade Aerodynamics* (2001).
- [5] J. D. Denton, *Loss mechanisms in turbomachines*. ASME 1993 International Gas Turbine and Aeroengine Congress and Exposition, Cincinnati, Ohio, 24–27 (1993).
- [6] D. Mee, N. Baines, M. Oldfield, and T. Dickens, *An examination of the contributions to loss on a transonic turbine blade in cascade*, *ASME 1990 International Gas Turbine and Aeroengine Congress and Exposition*, , V005T16A012 (1990).
- [7] L. Xu and J. D. Denton, *The base pressure and loss of a family of four turbine blades*, *Journal of turbomachinery* (1988).
- [8] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso, *Su2: an open-source suite for multiphysics simulation and design*, *AIAA Journal* (2015).
- [9] S. K. Abbas, *Production of synthesis gas by steam reforming of methane*, (2014).
- [10] A. Jameson, *Aerodynamic design via control theory*, *ICAESE Technical Report* (1988).
- [11] J. Reuther and A. Jameson, *Control based airfoil design using euler equation*, *Paper No. AIAA-94-4272-CP* (1994).
- [12] J. Reuther and A. Jameson, *Control theory based airfoil design using for potential flow and a finite volume discretization*, *AIAA Paper No. AIAA-94-0499* (1994).
- [13] A. Jameson and N. A. Pierce, *Optimum aerodynamic design using the navier-stokes equations*, *Proceedings of the AIAA 35<sup>th</sup> Aerospace Science Meeting and Exhibit, Reno, NV* (1997).
- [14] J. C. Newman, A. C. Taylor, R. Barnwell, P. A. Newman, and G. J. W. Hou, *Overview of sensitivity analysis and shape optimization for complex aerodynamic configurations*, *J. Aircr.*, 36(1) , pp. 87 (1999).

- [15] S. Yang, H. Wu, and F. Liu, *Aerodynamic design of cascades by using an adjoint equation method*, AIAA Paper No. AIAA-2003-1068 (2003).
- [16] A. Jameson, *Optimum aerodynamic design using cfd and control theory*, 12th Computational Fluid Dynamics Conference (1995).
- [17] M. B. Giles and N. A. Pierce, *An introduction to the adjoint approach to design*, Flow, turbulence and combustion (2000).
- [18] A. Jameson, *Automatic design of transonic airfoils to reduce the shock induced pressure drag*, Proceedings of the 31st Israel annual conference on aviation and aeronautics, Tel Aviv (1990).
- [19] R. M. Hicks and P. A. Henne, *Wing design by numerical optimization*, Journal of Aircraft (1978).
- [20] D. Wang and L. He, *Adjoint aerodynamic design optimization for blades in multi-stage turbomachines: Part 1 methodology and verification*, Journal of Turbomachinery-transactions of The Asme - J TURBOMACH-T ASME (2010).
- [21] M. Pini, G. Persico, and V. Dossena, *Robust adjoint-based shape optimization of supersonic turbomachinery cascades*, ASME Turbo Expo 2014: Turbine Technical Conference and Exposition GT2014 (2014).
- [22] R. P. Dwight, *Robust mesh deformation using the linear elasticity equations*, Computational fluid dynamics (2009).
- [23] S. Vitale, T. A. Albring, M. Pini, N. R. Gauger, P. Colonna, et al., *Fully turbulent discrete adjoint solver for non-ideal compressible flow applications*, Journal of the Global Power and Propulsion Society (2017).
- [24] T. A. Albring, M. Sagebaum, and N. R. Gauger, *Development of a consistent discrete adjoint solver in an evolving aerodynamic design framework*, 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2015).
- [25] T. A. Albring, M. Sagebaum, and N. R. Gauger, *Efficient aerodynamic design using the discrete adjoint method in su2*, 17th AIAA/ISSMO multidisciplinary analysis and optimization conference (2016).
- [26] M. Colonno, K. Naik, K. Duraisamy, and J. Alonso, *An adjoint-based multidisciplinary optimization framework for rotorcraft systems*, 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2012).
- [27] H. Hodson and R. Dominy, *Three-dimensional flow in a low-pressure turbine cascade at its design condition*, Journal of turbomachinery (1987).
- [28] J. D. Denton and L. Xu, *The trailing edge loss of transonic turbine blades*, ASME 1989 International Gas Turbine and Aeroengine Congress and Exposition (1989).

- [29] S. Chen, *A loss model for the transonic flow low-pressure steam turbine blades*, Institute of Mech. Engrs. C (1987).
- [30] F. R. Menter, *Improved two-equation  $k - \varepsilon$  turbulence models for aerodynamic flows*, NASA Technical Memorandum (1992).
- [31] P. L. Roe, *Characteristic based schemes for the euler equations*. Ann. Rev. Fluid Mechanics (1981).
- [32] J. M. Weiss, J. P. Maruszewski, and A. S. Wayne, *Implicit solution of the Navier-Stokes equation on unstructured meshes*, AIAA Journal (1997).
- [33] M. Giles, *Unsflo: A numerical method for the calculation of unsteady flow in turbomachinery*, Cambridge, Gas Turbine Laboratory, Massachusetts Institute of Technology (1991).
- [34] V. M. Korivi, A. C. Taylor III, P. A. Newman, G. W. Hou, and H. E. Jones, *An incremental strategy for calculating consistent discrete cfd sensitivity derivatives*, (1992).
- [35] E. J. Nielsen and M. A. Park, *Using an adjoint approach to eliminate mesh sensitivities in computational design*, AIAA journal (2006).
- [36] M. Sagebaum, T. Albring, and R. Gauger N, *Codipack code differentiation package — scientific computing*, (2017).
- [37] L. Sbardella, A. Sayma, and M. Imregun, *Semi-unstructured mesh generator for flow calculations in axial turbomachinery blading*, Unsteady Aerodynamics and Aeroelasticity of Turbomachines (1998).
- [38] *Gambit user's guide: 3.graphical user interface*, (2011).
- [39] <https://www.sharcnet.ca/software/ansys/17.0/enus/help/cfxmod/i1299598.html>, (2011).
- [40] T. Albring, N. Gauger, S. Vitale, and M. Pini, *Adjoint-based optimization of multi-stage turbomachines using su2*, (2017).