NOTE TO USERS

Aerodynamic Design and Optimization of Turbomachinery Blading

Temesgen Teklemariam Mengistu

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montréal, Québec, Canada

April 2005

Your file Votre référence
ISBN: 0-494-04062-9
Our file Notre référence
ISBN: 0-494-04062-9

ABSTRACT

Aerodynamic Design and Optimization of Turbomachinery Blading

Temesgen Teklemariam Mengistu, Ph.D.
Concordia Unviersity, 2005

Aerodynamic shape optimization of gas turbine blades is a very challenging task, given e.g. the flow complexity, the stringent performance requirements, the structural and manufacturing constraints, etc... This work addresses the challenge by automating the optimization process through the development, implementation and integration of state-of-the-art shape parametrization, numerical optimization methods, Computational Fluid Dynamics (CFD) algorithms and computer architectures. The resulting scheme is successfully applied to single and multi-point aerodynamic shape optimization of several cascades involving two-dimensional transonic and subsonic, viscous and inviscid flow in compressor and turbine cascades.

The optimization objective is to achieve a better aerodynamic performance, subject to aerodynamic and structural constraints, over the full operating range of gas turbine cascades by varying the blade profile. That profile is parameterized using a Non-Uniform Rational B-Splines (NURBS) representation, which is flexible accurate and capable of representing the blade profiles with a relatively small number of control points for a given tolerance. The NURBS parameters are then used as design variables in the optimization process.

The optimization objective is determined from simulating the flow using an in-house CFD code that solves the two-dimensional Reynolds-Averaged Navier-Stokes (or Euler) equations using a cell-vertex finite volume method on an unstructured triangular mesh and turbulence is modeled using the Baldwin-Lomax model.

To save computing time significantly, Artificial Neural Network (ANN) is used to build a low fidelity model that approximates the optimization objective and constraints. Moreover, to reduce the computing wall-clock time, the optimization scheme was parallelized on an SGI ALTIX 3700 machine using Message Passing Interface (MPI), resulting in a parallelization efficiency of almost 100%.

Different numerical optimization methods (genetic algorithm, simulated annealing and sequential quadratic programming) were developed, tested and implemented for the different parts of this work.

The present choice of objective function and optimization methodology results in a significant improvement in performance for all the cascades that were optimized, without violating the design constraints. The use of ANN results in a ten-fold speed-up of the design process and the scheme parallelization allows for further reduction of the wall-clock time.

ACKNOWLEDGEMENTS

I gratefully acknowledge Dr. Wahid Ghaly, my supervisor, for his continued support and guidance over the years. He spent many hours of his time to help and teach me. I thank him for the opportunity he gave me to study and for all the lessons and precious advice provided throughout the research. This work is made possible by him.

I would also like to thank my colleagues for having a good working environment and valuable source of assistance, encouragement and friendship. Special thanks goes to Kasra Daneshkhah for a fruitful discussion I had and steady helps regarding the CFD analysis, Tarek Mansour and Wassim Bacha for reading the first draft of the thesis.

Moreover, I would like to thank the entire MIE department faculty and staff for all the support and encouragement (and toleration) they have shown me throughout the years of my stay.

Special thanks must go to RQCHP, Réseau québécois de calcul de haute performance at University of Montréal, and the staff, Michel Béland and Richard Lefebvre, for their consistent support during the development and use of parallel computations at RQCHP.

My friends supported me in my research work. I want to thank them for all their help, support, encouragement and valuable hints. Especially I am obliged to Fantahun Defersha, Alebachew Yimer, and Aklilu Haile.

Finally, the ultimate acknowledgement be to my parents, brothers and sisters for their unending love, prayer and support all the time.

TABLE OF CONTENTS

LIST OF FIGURES ix

LIST OF TABLES xiii

LIST OF SYMBOLS xiv

1 INTRODUCTION **1**

1.1 Aerodynamic design approaches to turbomachine blading 2

1.2 Aerodynamic blade shape optimization 4

1.2.1 Optimization algorithms 5

1.2.2 Response surface approximation (RSA) 6

1.3 Previous work on optimization 7

1.4 Present work 10

2 NUMERICAL OPTIMIZATION **13**

2.1 Introduction 13

2.2 Numerical Optimization 14

2.2.1 Genetic Algorithm 15

2.2.2 Simulated Annealing 19

2.3 Flow Field Analysis 21

2.4 Response Surface Approximation 22

2.4.1 ANN Training 25

2.4.2 ANN Testing 27

3 BLADE SHAPE PARAMETRIZATION **36**

3.1 Introduction 36

3.2 Methodology 40

3.2.1 NURBS Functions 40

3.2.2 Optimization Objective 41
3.2.3 Optimization Scheme 42
3.2.4 Numerical Implementation 43
3.3 Method Validation 45
3.4 NURBS Representation of Generic Blade Cascades 46
3.5 NURBS Representation of Sanz Transonic Compressor Cascade . . . 52
3.6 NURBS Representation of DFVLR Subsonic Turbine Cascade T106 . 56

4 OPTIMIZATION OF GAS TURBINE CASCADES 60

4.1 Introduction 60
4.2 Design Methodology and Choice of Objective Function 61
4.3 Geometric Representation 64
4.4 Optimization Algorithm 65
4.5 Flow Field Analysis 66
4.6 Aerodynamic Optimization Process 66
4.7 Single-Point Optimization 69
4.7.1 Redesign of Transonic Impulse Turbine Cascade 70
4.7.2 Redesign of Onera Compressor Cascade 73
4.8 Multi-point Design of NACA Transonic Compressor Cascade 90
4.9 ANN-Based Multi-point Optimization of a NACA 65 Subsonic Compressor 98

5 CONCLUSION 114

5.1 Summary 114
5.2 Concluding remarks 116
5.3 Recommendations for future work 118

Bibliography **121**

Appendices **129**

A BLADE NOMENCLATURE **129**

B FLOW FIELD ANALYSIS **132**

B.1 The Governing Equations 132

B.2 Space Discretization 134

B.3 Artificial Viscosity 136

B.4 Integration to Steady State 137

B.5 Boundary Conditions 138

B.5.1 Inflow/Outflow Boundary Conditions 138

B.5.2 Periodic Boundary Conditions 139

B.5.3 Solid wall Conditions 139

C VALIDATION CASES **141**

C.1 Optimization Scheme Validation 141

C.1.1 The Rosenbrock Function 141

C.1.2 The Rastrigin Function 142

C.1.3 Concluding Remarks 144

C.2 ANN-based Optimization Scheme Validation 144

LIST OF FIGURES

1.1 Illustration of S1 and S2 surfaces,[Boyer,[1]]. 4

2.1 A typical nerve cell, [Guyton,[2]]. 33
2.2 A typical artificial neuron cell, [Guyton,[2]]. 33
2.3 A typical neural networks architecture. 34
2.4 The back propagation neural networks algorithm. 35

3.1 NURBS vs B-Splines accuracy for the generic compressor. 41
3.2 SQP convergence history for generic compressor with 9 and 11 control points. 43
3.3 NURBS representation of quarter circle and the control polygon. . . . 46
3.4 Convergence history for the generic compressor for different number of control points. 50
3.5 NURBS representation of generic compressor profile. 51
3.6 Generic compressor blade: Mach number distribution. 51
3.7 NURBS representation of generic turbine:IGV. 52
3.8 NURBS representation of SANZ compressor. 54
3.9 Sanz transonic compressor Mach number along the blade. 55
3.10 Sanz transonic compressor:Mesh. 55
3.11 NURBS representation of DFVLR-T106 subsonic turbine. 57
3.12 Sanz transonic compressor: Original vs NURBS curvature. 58
3.13 DFVLR-T106 subsonic turbine: Mach number distribution. 59

4.1 Blade geometry description. 65
4.2 The aerodynamic optimization implementation. 67

4.3 The ANN-based aerodynamic design optimization process. 68
4.4 GA convergence history for the impulse turbine cascade. 75
4.5 Isentropic Mach contours for unconstrained case. 76
4.6 Isentropic Mach contours for constrained case. 77
4.7 Isentropic Mach along the blade for unconstrained case. 78
4.8 The impulse turbine cascade. 79
4.9 Isentropic Mach along the blade for constrained case. 80
4.10 Isentropic Mach contours. 81
4.11 Isentropic Mach distribution. 82
4.12 Impulse turbine profile resulting from the SA algorithm. 83
4.13 Isentropic Mach distribution along the blade resulting from the SA algorithm. 84
4.14 Convergence history for the impulse turbine. 85
4.15 Pressure distribution on the blade for the initial, target and optimized design. 86
4.16 Isentropic Mach contours. 87
4.17 Isentropic Mach distribution. 88
4.18 Onera compressor cascade original and redesigned profiles. 89
4.19 The original and optimized NACA transonic compressor rotors. . . . 93
4.20 Efficiency vs. total pressure ratio for the original and optimized NACA transonic compressor rotors. 94
4.21 Efficiency vs. mass flow rate for the original and optimized NACA transonic compressor rotors. 95
4.22 Total pressure loss coefficient vs. mass flow rate for the original and optimized NACA transonic compressor rotors. 96

4.23 Isentropic Mach contours for the original and optimized NACA transonic compressor profile at a given mass flow rate. 97
4.24 Pressure distribution along the original and optimized NACA transonic compressor profile at a given mass flow rate. 98
4.25 The range of geometry explored for the design space. 102
4.26 The convergence history of ANN training/testing. 103
4.27 The ANN model validation. 104
4.28 Speed lines of blade profiles used in training/testing the ANN. 105
4.29 The original and optimized blade profiles. 106
4.30 Efficiency vs. mass flow rate for the NACA 65 subsonic compressor. . 107
4.31 Efficiency vs. pressure ratio for the NACA 65 subsonic compressor. . 108
4.32 Efficiency vs. mass flow rate for the NACA 65 subsonic compressor. . 109
4.33 Efficiency vs. pressure ratio for the NACA 65 subsonic compressor. . 110
4.34 Pressure ratio vs. mass flow rate for the NACA 65 subsonic compressor. 111
4.35 Mach contours for the improved and original NACA 65 subsonic compressor profiles at a given mass flow rate. 112
4.36 Pressure distribution on the blade surface for the NACA 65 subsonic compressor profile at a given mass flow rate. 113

A.1 Blade definition. 129

B.1 The cell vertex. 136
B.2 Cascade notation and boundary conditions. 140

C.1 Convergence history for the Rosenbrock test function with 40 design variables. 148

C.2 Convergence history for the Rastrigin test function with 40 design variables. 149
C.3 The surface plot for Rastrigin's function. 150
C.4 The Rosenbrock surface plot. 151
C.5 The surface plot of ANN-approximation for the Rosenbrock's function. 151
C.6 Comparison of convergence history for the optimization of Rosenbrock test function based on ANN approximation with the optimization based on the exact function. 152

LIST OF TABLES

C.2 The computation result of SA and GA for the Rastrigin function . . . 146

C.1 The SA result for the Rosenbrock function 146

C.3 Predicted value vs. actual values at the 21 test points 147

LIST OF SYMBOLS

C_k	the weight for the k term of the objective function
F_{obj}	objective function
m	mass flow rate
N	the basis function for the Non-Uniform Rational B-Splines
n	number of data points
$\overrightarrow{P}$	control point for the Non-Uniform Rational B-Splines
p	pressure
$S1$	blade to blade surface
$S2$	meridional plane
u	knot parameter for the Non-Uniform Rational B-Splines
W	weight for the Non-Uniform Rational B-Splines
X	design variables
x, y	axial and tangential directions/coordinates
β	flow angles
η	adiabatic efficiency
ω	total pressure loss coefficient, defined as $\frac{P_{02}-P_{01}}{P_{02}-P_2}$
	Subscripts
0	total or stagnation quantity of the flow
1,2	downstream and upstream of the blade, or static quantity of the flow
i, j, k	running index
p	degree of B-spline basis function
	Superscript
*	the target quantity
	Acronyms
2D	Two-Dimensional

3D	Three-Dimensional
ANN	Artificial Neural Networks
CFD	Computational Fluid Dynamics
DV	Design Variables
GA	Genetic Algorithm
LE	Blade Leading Edge
NURBS	Non-Uniform Rational B-Splines
MDO	Multi-Disciplinary Optimization
MPI	Message Passing Interface
PR	Pressure Ratio
PT	Penalty Term
RSA	Response Surface Approximation
SA	Simulated Annealing
SQP	Sequential Quadratic Programming
TE	Blade Trailing Edge

Chapter 1

INTRODUCTION

In the last two decades, the unprecedented progress in computer technology and the maturity of Computational Fluid Dynamics (CFD) have enabled industry to routinely use two-dimensional (2D) and three-dimensional (3D) simulations of the flow field in understanding the complex nature of the flow in e.g. turbines and compressors. Recently, CFD is used in the optimization in turbomachinery design of different components and configurations.

Continuous progress has been made in the performance, reliability, and efficiency of the gas turbine engines, however there is still a constant need for continuous improvement with regard to noise, cost, efficiency, power, safety, and weight. Moreover, there is always a constant drive to reduce design cycle time. Unfortunately, an improvement in any one of these areas can often lead to detrimental effects in another area unless it is properly accounted for. Therefore, different engine types have specific engine design criteria which are optimized for their particular application. Hence a great deal of research is currently devoted to CFD-based shape optimization so as to allow the designer to meet the challenging design criteria in the shortened design cycle time.

The stators and rotors that make up the numerous stages of a turbomachine are designed to operate at a certain optimal condition, usually referred to as the design point. However, during their actual application they are operated at off-design conditions as well. In these off-design ranges the flow enters the stator and rotor stages at varying incidence angles, and lower losses are even more difficult to achieve due to certain flow phenomena such as flow separation and shock waves. Moreover, optimization should also take into account some key flow features that control the operating range such as compressor surge and stall, and turbine choke.

For years, researchers have used cascade tests to conduct research and to push the envelope for running turbomachines at off-design conditions. Although these cascade tests are not a perfect model representation of an actual rotating turbomachine, cascade testing provides the blade designer with a more economical and experimentally simpler method of examining the aerodynamic performance under various operating conditions. Some of the cascades are used in the present work as optimization test cases.

1.1. Aerodynamic design approaches to turbomachine blading

The internal flow in turbomachines is three-dimensional, viscous, and unsteady. The computational cost of such flow field is so high that it is common to consider an axisymmetric flow during the design cycle. The fully 3D problem is replaced by a series of 2D problems on two mutually perpendicular planes, the meridional plane and the blade-to-blade plane, which gives rise to the so-called through-flow problem and blade-to-blade problem, as illustrated in Fig. 1.1.

The through-flow is solved in the meridional plane, $S2$ plane in Fig. 1.1, containing the axis of the machine and provides the radial variation of pressure, temperature, and velocity triangles [1]. The flow in the blade-to-blade plane, which is $S1$ plane in the Fig. 1.1, is then solved on each of these axisymmetric surfaces like the flow around the rotor and the stator. The present work deals with the design of blade sections on these so-called blade-to-blade surfaces.

There are several approaches to the design problem. The classical approach called 'direct approach' where the designer evaluates the performance of a given geometry, and modifies it manually as a function of the computational results, according to either empirical rules or his/her own experience. This approach can be very time consuming and inefficient in some cases; this prompted researchers to develop alternate approaches to the design such as inverse methods and optimization methods.

Inverse methods derive their name from the fact that they invert the flow analysis algorithm such that the target performance e.g., the blade loading (pressure difference across the blade surfaces) or the pressure distribution along the blade surfaces is specified and the geometry that satisfies it is part of the flow solution [3]. This approach is ideal when the designer knows how to prescribe a 'good' target performance. Inverse methods are very efficient computing-wise however they require an experienced designer.

In cases where the designer would like to specify a range of design objectives and constraints, the alternative approach is numerical optimization method where some of the difficulties of inverse methods are avoided, however it is computationally much more expensive than inverse methods. Optimization methods are more flexible than inverse method in that objectives and constraints of any type can be imposed.

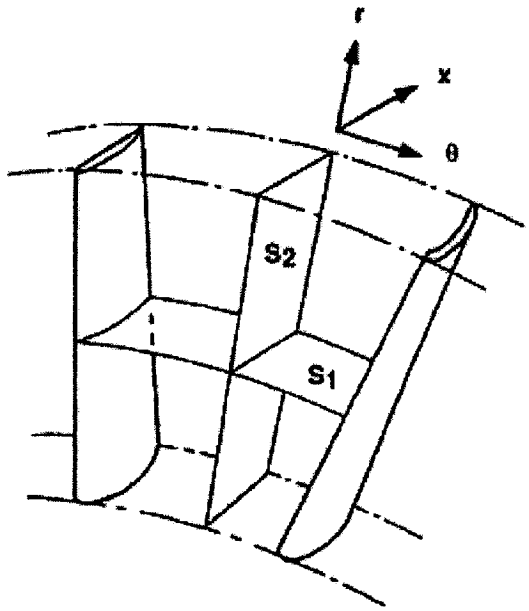


Figure 1.1: Illustration of S1 and S2 surfaces,[Boyer,[1]].

1.2. Aerodynamic blade shape optimization

Simulation-based shape optimization has three main components:

- geometric parametrization of the shape to be optimized
- flow simulation method to compute the optimization objective
- optimization algorithm to drive the optimization process

These three components are independent of each other and need to be integrated into one computation loop if this optimization approach is to be fully automated.

Although numerical optimization methods have been successfully used for a variety of design problems, its application to blade shape optimization still remains a formidable task. First of all the flow field inside a compressor or turbine blade is

generally unsteady, three-dimensional, turbulent, subsonic or transonic and the flow structure can be extremely complex. Secondly, the aerodynamic design optimization problem of a blade is severely constrained with e.g., structural and manufacturing constraints that result in a rather complex design space. Moreover the aerodynamic performance of a blade in transonic flow regime is very sensitive to the blade shape, which requires an accurate and flexible parametrization of that shape, and has to allow for local as well as global shape control. In addition, the objective function of an aerodynamic design optimization problem is often multi-modal and nonlinear as the flow field is governed by a system of nonlinear partial differential equations.

1.2.1 Optimization algorithms

Any numerical optimization method uses an optimization algorithm, the latter can use either direct or indirect search methods. The gradient-based search methods are well known algorithms of the indirect search type; they probe the optimum using the local gradient information. This method is efficient in searching the optimum, however it would frequently find and stop at the nearest local optimum value. This approach has been widely used in many problems including aerodynamic design of a wing [4] and some turbomachinery design optimization problems, some of which are mentioned later. However, the objective function for aerodynamic design problem of turbomachinery is usually multimodal, and thus one could only hope for reaching the optimum that is in the neighborhood of the initial design point using gradient-based methods. To find a global optimum, one must start the optimization process repeatedly from a number of initial points and check for consistency of the computed optima.

The second approach is the use of direct search methods. Global optimization method (e.g. Simulated Annealing (SA) and Genetic Algorithm (GA)) is an emerging optimization algorithm, which is recently applied in turbomachinery design problems [5; 6; 7]. These methods are global optimizers that can handle complex nonlinear search domains to provide global optimum design solution. Unlike the gradient-based methods, which usually require the objective function to be well behaved, these search techniques are able to tolerate noisy and discontinuous design spaces and objective functions. They are stochastic in nature, so they are able to search the entire design space with more chance of finding the global optimum when compared with gradient-based methods. They do not suffer of getting trapped in a local optimum and hence failing to find a global optimum. However they require significantly higher computation effort (in terms of the number of function evaluations) compared with the gradient-based methods.

1.2.2 Response surface approximation (RSA)

Compared with the gradient methods, direct methods require a relatively large number of function evaluations, which is found to be the most time consuming part of the optimization task. To reduce that computation load, it is possible to approximate the objective function over the design space using a Response Surface Approximation (RSA). Therefore, RSA is capturing the attention of researchers working on Multidisciplinary Optimization (MDO) problems in order to reduce the cost of optimization. RSA reduces significantly the number of objective function evaluations by using a surrogate model to approximate the objective function. It gives a low fidelity model, which must be validated later with the high fidelity model (the CFD simulation).

RSA is used to model the output of aerodynamic problem in aerodynamic design and optimization task. The RSA model is then used as an inexpensive model to approximate the objective function. The high fidelity evaluations of the objective function using the CFD simulations are reserved for initial model construction, design validation and model update.

1.3. Previous work on optimization

Over the past few years, there has been a significant research activity on aerodynamic optimization for turbomachines. Some representative works done on optimization in turbomachinery are given below.

Dennis [5] used a combination of GA and Sequential Quadratic Programming (SQP) algorithms for single point optimization of a two-dimensional turbine cascade. The objective function was penalized with aerodynamic, mechanical and geometric constraints. The optimization scheme required from 220 to 675 objective function evaluation, each requiring a call to the flow analysis code; this is quite a substantial amount of CFD computations. The blade geometry was modeled with a B-spline, and the genetic algorithm implemented in the optimization process required a binary representation of the design variables. The B-spline representation required a relatively large number of control points; in addition to its lack of flexibility in representing a larger family of blade profiles. This increases the load on the optimization process and narrows the design space. In addition, a binary representation of the real parameters of the design variables needs more memory compared with that of a real parameter representation.

Wang *et al.* [6] implemented the SA method on multiple processors for aerodynamic shape optimization. Their major focus was in the reduction of the computation associated with SA, the stochastic global optimization algorithm. It was indicated that SA was less powerful in exploring the design space, it also required large computational resources [8].

Dennis *et al.* [9] worked on design optimization of a two-dimensional cascade. The optimization objective was to minimize a weighted sum of total pressure loss, the negative of the aerodynamic loading (defined in terms of the force tangent to the cascade) and the number of airfoils subject to various aerodynamic and mechanical constraints, which were also included in the weighted sum. The shape of the blade was defined with physical parameters and B-spline representation. The optimization computations were reduced by combining a stochastic optimization with a response surface approach. The computation was carried out on 32 distributed memory parallel computers, consuming a wall-clock time of 50 hours. The design optimization process was found to consume a total of 5611 analysis calls to the 2D flow analysis code.

Oyama *et al.* [7] worked on the design point optimization of three-dimensional blades using genetic algorithm in a parallel computation environment. Although the optimization was carried out only at the design point, the researchers noted that the off-design performance had also improved.

Burguburu and Pape [10] worked on the design of turbomachinery bladings by numerical optimization using a gradient and genetic algorithm. The authors modified the suction side of the blade using Bézier curve with the goal of achieving maximum efficiency at some given operating point. The optimization approach used in the work is a combination of gradient and GA to make the search process fast and to find the global optimum. The result of the gradient method was verified by

GA and found to give the same optimum. This is most probably a local optimum solution which might be achieved under one or more of the following conditions: relatively small design space, insufficient exploration time and cycle for the GA, an initial solution that is very close to the found 'optimum' one. The authors noticed that the optimized blading shows a significant improvement both at the design and at off-design points, although the optimization was carried out at the design point. The Bézier curve used for this work is an effective representation for shape optimization of simple curves however complex curves require a high-degree Bézier curve. As the degree of Bézier curve increases so does the round-off error. Also it can be quite inefficient to compute a high-degree Bézier curve [11].

Multi-point optimization has been getting more attention recently since performance improvement in a turbine or compressor is required at both design and at off-design points. Köller *et al.* [12] designed a subsonic compressor for low losses over a wide operating range. The number of evaluations of the objective function, which involves a flow simulation using a CFD code, is significantly increased in this case as each blade profile is analyzed at all selected operating points.

It was shown that RSA can provide an inexpensive means to compute the objective function for a set of design variables [13]. It reduces the required number of CFD analysis and smoothes out the design space [14]. It provides a low fidelity model, which has to be validated using the high fidelity model, namely the CFD flow simulation. Artificial Neural Networks (ANN) is one type of RSA to approximate the design space. There are some attempts of using ANN for a single point aerodynamic design and optimization task in the area of turbomachinery design and optimization [13; 15; 16].

In summary, the examples of recently published research work done on single and multi-point aerodynamic shape optimization demonstrate the feasibility of using

numerical optimization methods such as GA and SA, in conjunction with Euler and Navier-Stokes codes. Application of some RSA models has also been published for a single-point aerodynamic optimization.

This review also shows that there is a need for: a) a more accurate and flexible representation for the blade shape, b) a more efficient numerical optimization methods and c) a more efficient optimization process. A typical multi-point aerodynamic design and optimization would be to improve the aerodynamic performance over the entire operating range including the design and off-design points using the minimum number of design parameters, the minimum number of CFD flow simulations and the minimum possible wall-clock time.

1.4. Present work

There are several motivations for this research. First, aerodynamic shape optimization in internal flow problems is relatively less investigated compared with optimization in external flow over wings. Research funding is definitely one reason. Another reason is that a typical wing is probably simpler than a typical turbine blade in terms of geometry as well as flow behavior. Therefore there is more work to be done in optimization in e.g. gas turbine blading owing to the flow as well as shape complexity, which also implies that the aerodynamic blade design requires a relatively large number of design parameters, making the task of blade design quite difficult and challenging.

The second motivation is the advancement and innovation of computer technology as well as the maturity of CFD and of optimization techniques, which have made simulation-based automated optimization a realizable task with a relatively modest computing cost.

The third motivation is the requirement that the design cycle time be reduced while the design constraints are made more stringent. This can be accomplished by automating the process of sweeping the design space hence allowing designers to focus on the resulting optimum blade shapes rather than spending their time in manually exploring the design space.

The optimization is achieved by devising a strategy for each of the main components of the optimization process listed in Sec. 1.2, namely a parametric representation of the blade shape, a flow simulation method to compute the objective function and an optimizer that decides on how to change that shape to achieve the optimization objective.

This research aims at developing a robust, flexible, and efficient global aerodynamic design and optimization method for turbomachines that can be used for single point as well as multi-point optimization problems. To this end, the following points, which also represent the contributions of this work, are developed:

1. A parametric representation of blade profiles in two-dimensional flow using Non-Uniform Rational B-Splines (NURBS) is developed, implemented and assessed for 2D turbine and compressor cascades

2. Genetic algorithm (GA) and simulated annealing (SA) are developed, validated and applied to 2D cascade optimization as well as to optimally represent the blade geometry using NURBS with the minimum possible number of parameters

3. A gradient-based scheme that uses sequential quadratic programming (SQP) is developed, validated and used in conjunction with GA/SA for the construction of ANN approximation model.

4. The choice of objective function for single point and multi-point optimization is developed

5. The methodology of using ANN, a low order approximation of the optimization objective is developed, validated and implemented in the optimization loop

6. Optimization of different compressor and turbine cascades are carried out for single and multi-point optimization, inverse method optimization, in transonic or subsonic, inviscid or viscous turbulent flow regimes

The thesis is organized as follows. Chapter 1 gives an introduction, motivation, scope and organization of the thesis. The numerical optimization algorithms, GA and SA, as well as the response surface approximation, ANN, are discussed in Chapter 2. In Chapter 3, a key element of the design, namely the parametric representation of the blade shape, is developed and applied to different turbine and compressor cascade profiles. Chapter 4 presents the development and implementation of aerodynamic design using GA and applies it to the design of turbine and compressor cascades. Both single and multi-point optimization of turbomachine cascades are studied and results are presented. In addition RSA based design scheme is presented and implemented for the case of a compressor rotor at the design point and at off-design points in the same chapter. The last chapter summarizes and concludes the thesis; it also points out some outstanding challenges and makes recommendations for future work.

Chapter 2

NUMERICAL OPTIMIZATION

2.1. Introduction

The major elements in turbomachinery blade design using optimization scheme are three: shape parametrization, numerical optimization and objective function computation. These three elements perform independently of each other and coupling of them gives the automated shape optimization procedure.

The computation time involved during the optimization process can be prohibitively large in particular when using the Navier-Stokes equations to simulate the flow field so as to compute the objective function. For this reason, Response Surface Approximation (RSA) is also an important element in the design process in order to reduce the optimization computing time by providing a good approximation of the objective function.

This chapter will focus on presenting the numerical optimization methods that were used namely, GA and SA, as well as the RSA method used to approximate the objective function, namely ANN. It will also outline the numerical scheme that was used to simulate the flow in a 2D cascade so as to compute an accurate value for

the objective function.

2.2. Numerical Optimization

Numerical optimization schemes are categorized into two broad classes: gradient based and non-gradient based optimization scheme.

Gradient-based optimization schemes are fast and need a relatively small number of function evaluations when compared with the non-gradient based optimizers, however they are local optimizers, they will probably stop at the first optimum obtained during the process. They are of limited use when searching for e.g. global optimum of multi-modal optimization problem, such as aerodynamic optimization. The most popular and common method of the gradient-based algorithm is the sequential quadratic programming (SQP). The SQP method works very well and is relatively fast for problems, which have not multi-modal extrema.

Hybridization of the gradient method with global stochastic search schemes is an idea found to perform well in some cases [14; 17].

Both the gradient and the non-gradient schemes were tested for different cases of multi-modal problems and for the aerodynamic optimization. The global optimizers GA and SA are found to perform best in most cases but at a relatively high computing cost. There is a way to work with these algorithms by reducing the computation load by using ANN to approximate the objective function, hence taking advantage of their global optimization behavior. The weakness of the gradient-based method and the strength of the global optimization method have been critically observed by several researchers in the field [18].

The interest of this research falls on the use of global optimization technique after performing a thorough investigation and tests on several kinds of optimization

scheme.

After carrying out tests of GA, SA and SQP on solving the current problem, it was decided to use GA and SA in the aerodynamic optimization and SA in NURBS parametrization and SQP combined with GA/SA in the process of building ANN model.

2.2.1 Genetic Algorithm

Genetic algorithms are increasingly used for aerodynamic and other optimization task [19; 5; 20; 21; 22; 23]. They provide robust strategy capable of finding the near global minimum in a problem containing many local minima.

Genetic algorithms are general-purpose search algorithms based upon the principles of evolution observed in nature. Genetic algorithms combine selection, crossover, mutation, and elitism operators with the goal of finding the best solution to a problem. It searches for this optimal solution until a specified termination criterion is met [24; 25].

The variables for the GA algorithm (the genes) can be either binary coded or real coded. A real coded genetic algorithm is used in this work. Using a real parameter for real variable problems instead of binary parameter for the design variables are common now in a problem where the design variables are real parameters [13].

The basic operations that make up the genetic algorithm developed are selection, crossover, mutation, and elitism.

Population Size

The population size is the number of candidate solutions in one generation. The larger the population size the more computationally intensive is the search. In

nature, the bigger the gene pool the more diverse is the genetic make up of the population with many individuals each with its own set of characteristics that enable it to survive. One advantage of this diversity is that there will be no dominant gene that, for instance, may be susceptible to a particular disease and result in the elimination of the whole species.

With large populations, it can be seen how the search for the global optimal solution can be a slow (if not a never-ending) process.

If the population size is small, then a strong individual quickly becomes dominant and the diversity of the gene pool is reduced. The result is that good individuals (local optima) are quickly created but the dominance of particular genes restricts the search space.

As new solutions are generated, it is common to keep the population size constant by replacing the old individuals with new ones. Each generation could be completely replaced by its offspring, or as a new offsprings is created, it could be accepted or rejected depending on its fitness, which is based on the value of the objective function. The advantage that computers have over nature is that good individuals do not have to die and can be retained for indefinite reproduction. The retention of certain fit individuals is known as elitism.

Selection

This is the operation of choosing candidates (parents) on which to perform crossover in order to create new solutions. The purpose is that the best individuals prevail the selection process.

There are two commonly used selection procedures which are driven by fitness: roulette wheel and tournament selection [19]. In roulette wheel, each individual is assigned a slice of a wheel, the size of the slice being proportional to the fitness of

the individual. The wheel is then spun and the individual having better fitness has a better chance of being selected.

Crossover

Crossover is the operator that creates new candidate solutions. The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover occurs during evolution according to a user-definable crossover probability (Pc).

Two kinds of crossover operations are included in the real-coded GA developed in this work namely, arithmetic and heuristic crossover operators.

Arithmetic crossover operator combines linearly two parent chromosome vectors to produce two new offsprings given as:

$$\begin{cases} Child1 = \alpha Parent1 + (1-\alpha)Parent2 \\ \\ Child2 = (1-\alpha)Parent1 + \alpha Parent2 \end{cases} \tag{2.1}$$

where α is a random number between 0 and 1.

While heuristic crossover operator uses the fitness values of the best individual and the worst individual to determine the search direction and creates the new offspring. The offspring are created according to the following equations:

$$\begin{cases} Child1 = Bestparent + \alpha(Bestparent - Worstparent) \\ \\ Child2 = Bestparent \end{cases} \tag{2.2}$$

Where α is a random number between 0 and 1.

Mutation

Mutation is a genetic operator that alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With these new gene values, the genetic algorithm may be able to arrive at a solution better than was previously possible.

Mutation is an important part of the genetic search as it helps to prevent the population from stagnating at any local optimum. Mutation occurs during evolution according to a user-defined mutation probability (Pm). This probability should usually be set fairly low (the default value is set to 0.01). If its value is set too high, the search will turn into a primitive random search. Uniform type mutation is used for the algorithms in this work; it replaces the value of the chosen gene with a uniform random value selected between the user-specified upper and lower bounds for that gene.

Elitism

For a generational GA, elitism makes few identical copies (e.g. two) of the best performer in the old pool and places them in the new pool, thus ensuring that the fit chromosome survives. It is simply the guarantee that the fit solution found to date would remain within the population.

The calculation is stopped when anyone of the following three criteria is reached: When the best fitness in the current population becomes less than the specified fitness threshold during the minimization problem; when the specified maximum number of generation has been run; or when the elapsed evolution time exceeds the specified maximum evolution time.

In summary, GA is a non-gradient based optimization. It does not use gradient

information during the process. It is based on the function value. For this reason, it requires a large number of iterations (function calls) compared with SQP. Their great advantage is that they cannot be easily trapped in local minima or maxima. They can be used effectively for multi-modal optimization problems with several extreme.

2.2.2 Simulated Annealing

As its name implies, simulated annealing (SA) exploits analogy between the way in which a metal cools and freezes to form the minimum energy crystalline structure (the annealing process) and the search for a minimum in a general system. If a physical system is melted and then cooled slowly, the entire system can be made to produce the most stable (crystalline) arrangement which corresponds to the system global minimum energy level, and not get trapped in a local minimum.

The SA algorithm was first proposed by Metropolis *et al.* [26] as a means to find the equilibrium configuration of a collection of atoms at a given temperature. Kirkpatrick *et al.* [27] were the first to use the connection between this algorithm and mathematical minimization as the basis of an optimization technique for combinatorial (as well as other) problems.

SA's major advantage over other methods is its ability to avoid being trapped in local minima. The algorithm employs a random search, which not only accepts changes that decrease the objective function f, but also some changes that would increase it. The latter are accepted with a probability $P = e^{-\frac{\delta f}{T}}$ where δf is the increase in objective function f and T is a control parameter, which by analogy with the original application is known as the system "temperature" irrespective of the objective function involved.

Briefly SA works in the following way. Given a function to optimize, and some initial values for the variables, simulated annealing starts at a high artificial temperature. While cooling the temperature slowly, it repeatedly chooses a subset of the variables and changes them randomly in a certain neighborhood of the current point. If the objective function has a lower function value at the new iterate, the new value is chosen to be the initial value for the next iteration. If the objective function has a higher function value at the new iterate, then the new value is chosen to be the initial value for the next iteration with a certain probability, depending on the change in the value of the objective function and the temperature. The higher the temperature and the lower the change, the more probable the new value is chosen to be the initial variable for the next iteration.

Throughout this process, the temperature is decreased gradually, until eventually the values do not change anymore. Then, the function is presumably at its global minimum. The global minimum is obtained by choosing an appropriate "cooling schedule", which includes the temperature and its cooling rate. A cooling schedule describes the temperature parameter T, and gives rules for lowering it as the search progresses.

Unfortunately, there is no systematic way of determining the best annealing schedule for a given optimization problem. The one implemented in this work is that suggested by Corana *et al.* [28], in which T is decreased geometrically after a predetermined number of function evaluations. An initial temperature T_{in} is given based on Kirkpatrick's [27] suggestion, namely that a suitable initial temperature is one that results in an acceptance rate of about 80% for all moves and, after every m steps, T is multiplied by ϵ, a temperature reduction factor, that assumes a value between zero and one.

2.3. Flow Field Analysis

The flow field analysis method is used to simulate the flow in order to compute the aerodynamic performance of the given turbomachine blade. An accurate analysis method must be used to calculate e.g. the loss, efficiency, total pressure ratio, mass flow rate so that aerodynamic optimization can be achieved by a proper sweep of the design space. The accurate prediction of the flow field for the turbomachine cascade is crucial for optimization. If the flow simulation does not capture the flow physics then the optimizer will not capture it either, which may result in an untrue optimum. In other words, the optimum design is at best as good as the flow simulation tool.

The CFD simulation code that is used in this work, was originally developed and implemented by Ahmadi and Ghaly [3] for 2D inviscid flow and was further extended to solve the Reynolds-Averaged Navier Stokes (RANS) equations by Daneshkhah [29] where turbulence is modeled using the Baldwin-Lomax model. The two-dimensional inviscid/viscous flow equations are solved using a cell-vertex finite volume space discretization on an unstructured triangular mesh. The steady state solution is reached by pseudo-time marching the Euler/RANS equations using an explicit five-stage Runge-Kutta scheme. Local time stepping and implicit residual smoothing was used for convergence acceleration. The nonlinear blend of second and fourth order artificial viscosity was found to be successful in capturing shocks and eliminating pressure-velocity decoupling with minimal numerical diffusion. The method of characteristics was used to impose inflow and outflow boundary conditions. A short description of the CFD algorithm is given in Appendix B and more details of the method are given in [3; 29].

2.4. Response Surface Approximation

Response surface methodology is currently commonly used in engineering design optimization problems [24; 18]. It is a method of building a relatively inexpensive approximation model to compute the objective function instead of using the CFD simulation results to compute it. The approximation model can then be used inside an optimization loop to compute the objective function in place of the original expensive model, hence the number of such evaluations is no longer a critical parameter to be considered. In addition, as it is mentioned by Lai and Yuan [14], using approximation models can eliminate the computational noise which has a strong adverse effect on numerical optimization techniques by creating some non-physical local optima. The commonly known methods of approximations are Polynomial Approximations and Artificial Neural Networks (ANN) [30; 31].

In the Polynomial Approximation method, the response surface model is a polynomial of n^{th} degree whose coefficients are determined from a linear system of equations. The linear system is set up using least square minimization of the error between the polynomial and the actual model.

The selection of the data set for building the approximation is crucial and challenging. It affects the end result of the approximation as well as the optimization that is based on this approximation. For this, there is a task of Design of Experiments (DOE) before the creation of the model. The DOE gives a systematic and efficient means of analyzing a design space. It explores the high-dimensional design space and screen the most influential variables as design variables. Quadratic model is widely used in polynomial approximation scheme due to its flexibility and ease of use. The method has been used by several researchers in aerodynamic design [32; 30]. The effectiveness of the approximation method was studied by Markine and

Toropov [33] and was presented in comparison with the high fidelity model. The authors found that the low fidelity model was accurate enough with a good quality of approximation.

ANN-based approximation is a response surface approximation method based on the notion of artificial intelligence. The ANN-based approximation model is obtained by training it with some representative data and validating with data that was not used in the training. ANN is a very powerful interpolator that can be used to map functions with multiple inputs/otputs.

Researchers have used ANN-based approximation successfully for turbomachinery blade design [34; 35; 16; 36; 37; 23], and a comparison of ANN based and polynomial approximation has been reviewed and compiled in a paper by Shyy *et al.* [18]. Rai and Madavan [38] investigated the feasibility of applying neural networks to the design of turbomachinery airfoils. The ANN approach was found to be quite efficient for the task.

In this work the ANN is used as a low order Response Surface Approximation (RSA) of the objective function at a relatively low computing cost.

Artificial Neural Network is a mathematical model of the human brain. It is a network of multiple layers of simple processing elements called neurons. Each neuron is linked to some of its neighbors with varying coefficients of connectivity that represent the strengths of these connections. Learning is accomplished by adjusting these strengths to cause the overall network to output results for a certain set of inputs [39].

The most basic element of the human brain is a specific type of cell, which provides us with the abilities to remember, think, and apply previous experiences to each of our every actions. These cells are known as neurons, each of which can connect with up to 2×10^5 other neurons. Brain power is a function of the number

of these basic components and the multiple connections between them [40].

A biological neuron receives inputs from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then provides its own output as the final result. Figure 2.1 below shows a simplified biological neuron and the relationship of its four components.

The basis of ANN is to simulate the basic functions of natural neurons, however it is much simpler than the biological neuron. Figure 2.2 shows the basic operation of each artificial neuron in ANN.

The basic building blocks of ANN are the artificial neurons analogous to the natural ones. The various inputs to each neuron are multiplied by a connection weight and these products are summed up. It is then fed to a transfer function to generate the output. The following equation mimics the action of a net of neurons:

$$\begin{aligned} Y &= f(I) \\ I &= \sum_i (W_i X_i) \end{aligned} \tag{2.3}$$

Where the neuron output Y is a function of the weighted sum I of inputs/or the input layer coming from the previous layer of neurons.

As the brain basically learns from experience, ANN learn from the given data. It is sometimes called machine learning algorithms. Changing its connection weights (training) causes the network to learn the solution to a problem. The system learns new knowledge by adjusting these connection weights W_i using an optimization algorithm so as to minimize the error of its prediction [40].

The learning ability of a neural network is determined by its architecture and the algorithmic method chosen for training. There are generally two kinds of training scheme: unsupervised learning and supervised learning [41].

In unsupervised learning, the network learns from the given input data only. The sample output is not given in this case. It will find a way to organize or cluster the data without seeing the outputs [41]. In supervised learning, the network learns from the input and output sample data presented.

Back propagation algorithm is a method that is proven highly successful in training of multilayered neural nets using supervised learning.

It is believed based on a semi-theoretical proof that a feed-forward neural net with at least one hidden layer can approximate any continuous nonlinear function arbitrarily well, provided that sufficient number of hidden neurons are available [42].

A typical back propagation neural network has an input layer with several neurons, one or more hidden layers, and an output layer. Each of them are connected by adjustable weights which enable the network to compute complex associations between the input and output variables. Figure 2.3 shows a typical neural net with one input layer having four neurons or nodes, one hidden layer with several nodes and one output layer with two nodes.

The design of ANN involves two steps: a training step followed by a testing step.

2.4.1 ANN Training

The ANN training involves finding the right ANN model for a given problem, i.e. determining the type of ANN network, its architecture and choosing a training strategy. These choices depend on the function being approximated, like the presence of local minima, high dimensionality, disparity in input scales, etc.

The training algorithm in this work is a combination of GA and SA optimization for wider exploration of the design space and is followed by a gradient-based

scheme for a better exploitation. The weights are initialized randomly at the start and after a few iterations of the exploration algorithm, the weights are saved and are used for future experimentation.

The error is measured based on the maximum relative error or average error or percentage of the exact prediction out of the total cases under consideration at a certain predefined accuracy.

The steps in designing ANN model are:

- Choosing an appropriate structure: The multi-layer feed forward network is the most popular, it is the hierarchy of processing units, organized in a series of two or more mutually exclusive set of neurons or layers [41]. The first is the input layer used to accept input from the external. The last layer is the output layer which returns the output of the network. In between lies one or more hidden layers, where the computational process of the network is concentrated. The weights connect each unit in one layer to those in the next layer.

- Training strategy: The design of a reasonable training strategy is problem dependent. The following factors must be very well considered for the success of the training: order of training set, training algorithm convergence and divergence, trap at local minimum error, and measure of error.

- Setting and updating initial conditions for the weights: This step mainly depends on the characteristics of the error surface. If the error surface changes rapidly, the gradient calculated based on local information alone will give a poor indication of the "right path" [43], for this case, a smaller learning rate is desirable. If the surface is relatively smooth, a larger learning rate will speed convergence. However the shape of the error surface is rarely available, thus a general rule might be to use a larger learning rate that works and does not

cause oscillation. Proper initialization of the weight overcome local minimum and make the training more efficient [39].

- Choosing the number of hidden layers and units: The choice of the number of hidden layers and units requires engineering judgement [43]. Trade-offs between training time and network accuracy lead to iterative adjustment of the network using simulations. It is problem dependent.

2.4.2 ANN Testing

This is a measurability of the network generalization capability. Generalization is defined as the ability of the trained network to correctly predict the output for an input data not included in the training set. This is a major objective of the ANN design. The ANN design should produce a network that generalizes the behavior correctly to new as yet unforeseen inputs [43; 41].

The training algorithm keeps on reducing the error, but that is not an indication for its accuracy, because a behavior called memorization might be occurring. The units memorize the I/O mappings in the training set, without any capability to generalize. For this reason, there must be an optimum point where the training has to be stopped although it can still reduce the error. This point is determined by the test data that are not included in the training set. The training has to stop at the point where the error on the test set data is minimum and reasonably acceptable. This can measure the generalization capability of the network. A net that has been overtrained will usually have poor generalization, since the output space will follow the training data too closely.

The flow chart showing the steps involving the design steps is shown in Fig. 2.4.

The Back-propagation is one of the most commonly used ANN, structures coupled with supervised learning as the training strategy. In this training algorithm, the error between the results of the output neurons and the actual outputs is calculated and propagated backward through the network.

The algorithm is based on the Delta Rule which is based on the idea of continuously modifying the strengths of the input connections to reduce the difference (the delta) between the desired output value and the actual output of a neuron [41]. This rule changes the connection weights in the way that minimizes the mean squared error of the network. The error is back propagated into previous layers one layer at a time. The process of back-propagating the network errors continues until the first layer is reached. The network type called Feed forward, Back-propagation derives its name from this method of computing the error term.

Given the input/target data for the training given below as:

$$[X, T] = \{(x_1, t_1), (x_2, t_2), ..., (x_n, t_n)\} \tag{2.4}$$

where n is the number of training sets.

The basic steps involved in the design of an ANN of back propagation type is described as follows:

1. The number of nodes in the input and output layers is given in the problem of interest; the number of hidden layer(s) and the number of nodes in each hidden node, which can solve the problem are estimated.

2. The weights of each connection are randomly initialized.

3. The input vector is fed forward.

4. The output vector is calculated using Eq. 2.4 that mimics the neuron behavior by weighing each input, summing it and passing it to a transfer function.

5. The network error, defined as l_2-norm of the difference between the computed and the target outputs, is calculated using Eq. 2.8.

6. The network then attempts to minimize that error using an optimization approach.

For a general backpropagation ANN with input layer, hidden layer(s) and output layer, the input to each layer other than the first layer is given by the weighted sum of outputs coming from previous layers:

$$net_j = \sum_i (w_{ij} x_i) \tag{2.5}$$

Where i is the index for the node in the previous layer while j is in the current layer.

The output is obtained from the transfer function, f:

$$Y_j = f(net_j) = f(\sum_i (W_{ij} X_i)) \tag{2.6}$$

The transfer function, f, called activation function could be any function that can be used to convert the activation input into an output. Usually a continuous and analytic transfer function is used. Examples of activation functions used in ANN are *step function* which simulates a binary decision, and *sigmoid function* and *tan hyperbolic function* for nonlinear, continuous and differentiable replacement of the step function.

The sigmoid function is given by:

$$f(x) = \frac{1}{1+e^{-x}} \tag{2.7}$$

The output obtained from the output layer is compared with the target output given, and error is computed as follows:

$$E = \frac{1}{2}\sum_i (t_i - o_i)^2 \tag{2.8}$$

Where E is the sum of squared error, t is the target output and o the predicted output from the network.

The learning algorithm is adjusting the connection weights in order to minimize the error function. The learning rule is thus:

$$W_i \leftarrow W_i + \Delta W_i, \qquad \Delta W_i = -\eta \frac{\partial E}{\partial W_i} \tag{2.9}$$

Where the term $\frac{\partial E}{\partial W_i}$ is the gradient of the error function with respect to the connecting weight W_i, ΔW is the change in weight and η is the learning rate used to accelerate the training strategy, a value between 0 and 1 is assigned based on the nature of the problems, i.e the error surface. If the error surface changes rapidly, the gradient calculated based on local information only will give a poor indication of the "right path" [43], for this case, a smaller learning rate is desirable. If the surface is relatively smooth, a larger learning rate will speed convergence. However the shape of the error surface is rarely available, thus a general rule might be to use a larger learning rate that works and does not cause oscillation.

Another way to avoid oscillation when using large value of learning rate is to make the change in weight dependent of the past weight change by adding a momentum term [44]:

$$\Delta W_i(t+1) = -\eta \frac{\partial E}{\partial W_i} + \alpha \Delta W_i(t-1) \tag{2.10}$$

Where t denotes the presentation number and α is a constant which determines the effect of the previous weight change.

Gradient descent learning scheme can easily be trapped in local minima and its convergence rate is slow. In order to improve its performance and accelerate the convergence, advanced optimization scheme of second order training accurate can be used.

Approximating the function to be minimized, i.e. the error function in Eq. 2.8, based on 2^{nd} order Taylor series expansion:

$$F(X_{k+1}) = F(X_k) + \nabla F(X_k)^T \Delta X_k + \frac{1}{2} \Delta X_k^T \nabla^2 F(X_k) \Delta X_k \tag{2.11}$$

For quadratic function:

$$F(X) = \frac{1}{2} X^T A X + b^T X + c \tag{2.12}$$

where T denotes transpose, and A is the Hessian matrix which is defined below.

$$c = F(X) \qquad b = \nabla F(X) \qquad A = \nabla^2 F(X) \tag{2.13}$$

Setting the gradient of the Taylor series quadratic function to zero:

$$\begin{aligned} \nabla F(X_k) + \nabla^2 F(X_k) \Delta X_k &= 0 \\ \nabla F(X_k) + H_k \Delta X_k &= 0 \end{aligned} \tag{2.14}$$

Solving for ΔX_k:

$$\begin{aligned} \Delta X_k &= -H_k^{-1}\nabla F(X_k) \\ X_{k+1} &= X_k - H_k^{-1}\nabla F(X_k) \end{aligned} \tag{2.15}$$

Where H is the Hessian matrix defined as A in Eq. 2.13.

The evaluations of the Hessian matrix is very costly, hence the idea of SQP method is used to approximate the Hessian matrix. Different form of approximations exists. The two common approximation methods are Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS), in which an initial Hessian matrix H_o, is chosen and subsequently updated by an updated formula given in each methods [45; 46].

Although it requires n iterations to converge for a quadratic systems with n degree of freedom, the n directions have to be followed several times due to the fact that the system is not quadratic and the presence of round-off errors [44].

Although many practical issues are still unresolved, there are some useful observations:

Lack of success in application is usually due to faulty training, faulty architecture (number of hidden layers & units), and lack of functional relationship between input and output.

Boosting of the training strategy using GA and SA algorithms was found to improve the efficiency [47].

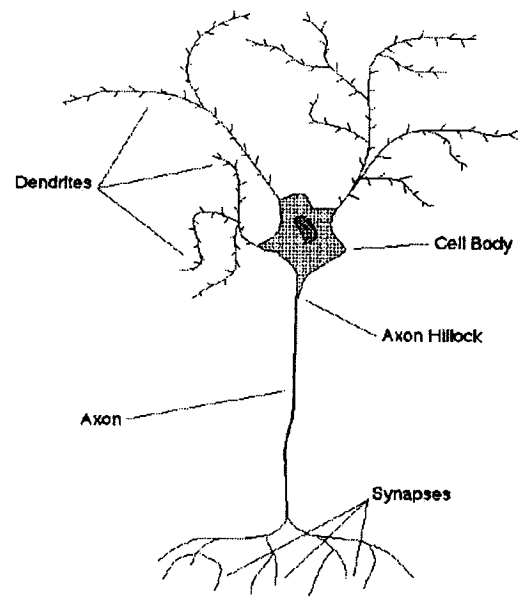


Figure 2.1: A typical nerve cell, [Guyton,[2]].

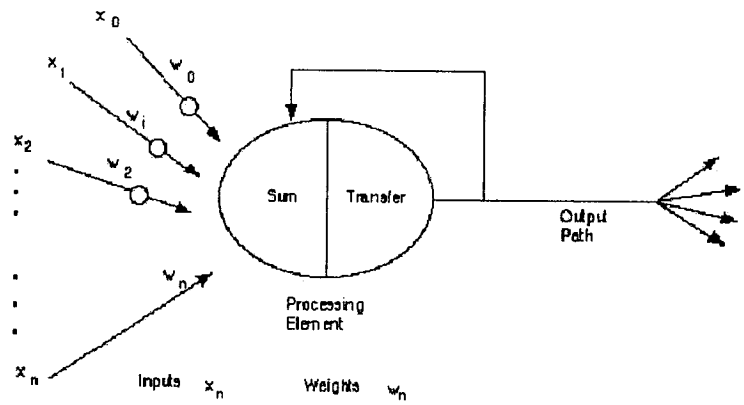


Figure 2.2: A typical artificial neuron cell, [Guyton,[2]].

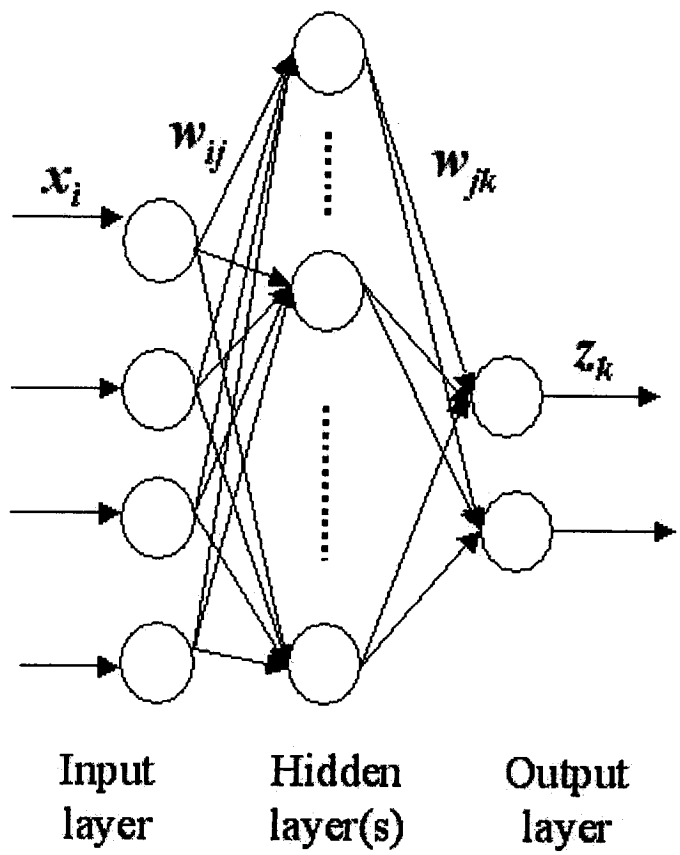


Figure 2.3: A typical neural networks architecture.

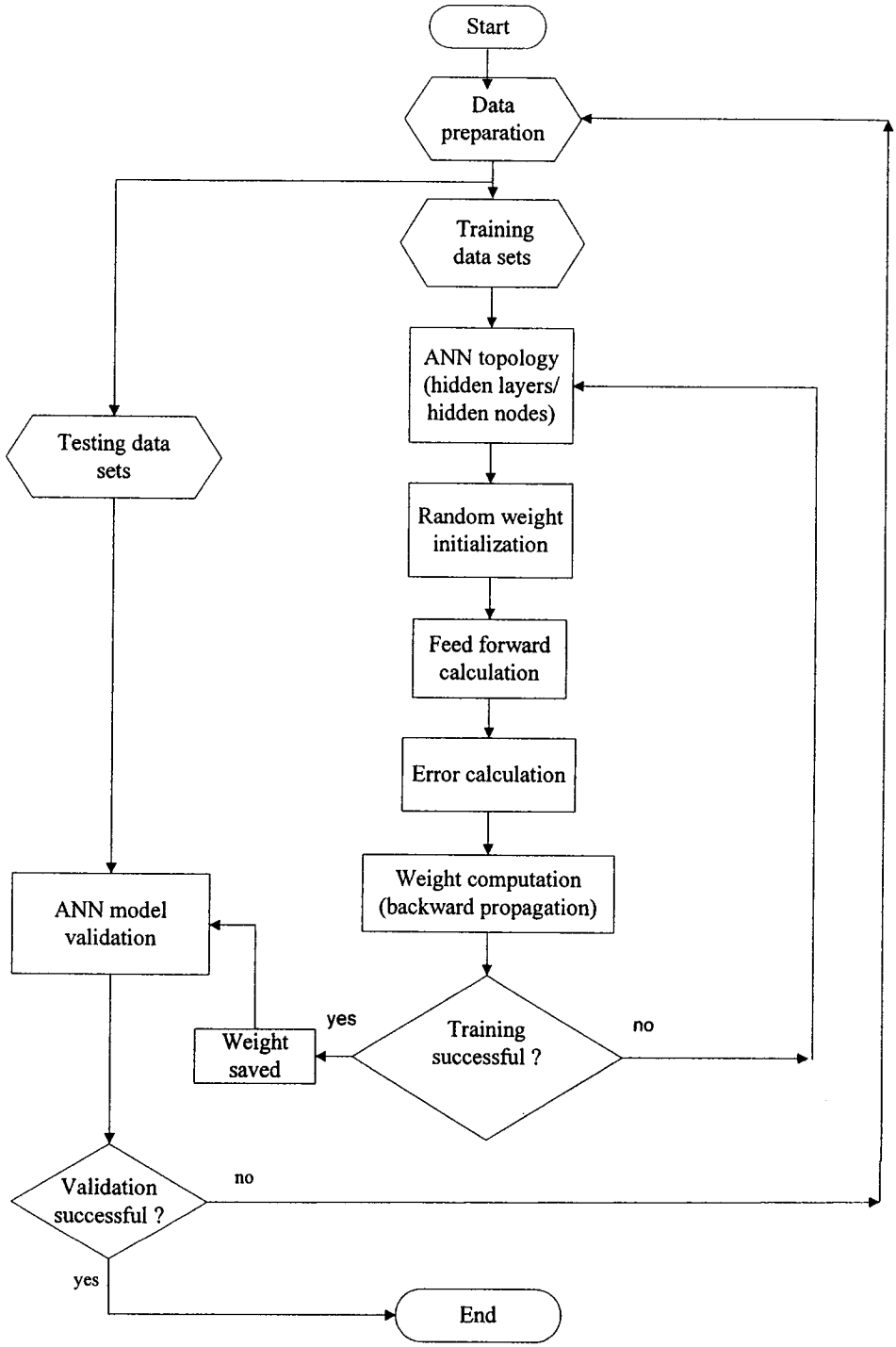


Figure 2.4: The back propagation neural networks algorithm.

Chapter 3

BLADE SHAPE PARAMETRIZATION

3.1. Introduction

In blade design optimization, one of the major challenges is how to choose the least number of design variables while maintaining the freedom and quality of the blade representations. The geometric representation of curves in two-dimensions, or surfaces in three-dimensions, is an important part of any shape optimization procedure. The geometry needs to be represented with a set of geometric parameters. These parameters must be chosen so that the variations of airfoil geometry can be obtained by smoothly varying them. It might also be necessary to impose some of the geometric constraints for structural reasons, aerodynamic reasons (e.g. to eliminate flow separations), etc. One would like to represent the shape with the least possible number of parameters for a given accuracy. Compressor blades of gas turbine engines are usually thin, low cambered and have a round leading edge (LE) and a sharp trailing edge (TE), whereas turbine blades are highly cambered and

have round leading and trailing edges. Therefore, the geometric representation of such blades with one function is quite a challenge.

Bézier curves [48] were first used in representing airfoil geometry, probably because of their ease of implementation. It has been used by several researchers in the field [49; 50; 51]. They used two Bézier curves, one for suction side and one for pressure side to define the blade section. Continuity of first, second or higher derivative is satisfied at the leading and trailing edge. They have two limitations, first they are global in nature, i.e. when a control point is moved the entire blade shape is modified, which results in less control over the local blade profile; second they cannot represent conics (e.g. leading and trailing edge circles) exactly.

To overcome the global nature of the Bézier polynomials, B-splines use the concept of control points introduced by Bézier, but with more complex interpolation functions that can capture local characteristics such that the displacement of a control point introduces a local modification of the curve near that point [9; 24] However, B-splines cannot represent conics exactly so that e.g. leading and trailing edge circles cannot be captured exactly.

Rai and Madavan [16] defined the thickness at specified control points and the thickness values at other points are obtained using B-spline method. Different airfoil shapes can be obtained by varying the thickness values at the control points. Similarly they defined control points on the mean camber line and splines are used to obtain other points on the camber line. Again the B-spline representation does not represent a wide family of blade shapes.

To alleviate the shortcomings of Bézier curves and B-splines, i.e. to allow for local control of the curve and represent conics exactly, NURBS can be used to represent the blade shape. Using a single NURBS function with at most thirteen control points and weights, Trépanier [4], were recently successful in representing,

rather accurately (up to manufacturing tolerance) and efficiently, the geometry of two-dimensional airfoils used in wing sections.

After making a search of the existing parameterizations schemes, NURBS functions are selected and developed to find the optimum parameterizations of blades used in gas turbine engines in this work. The NURBS curve is given by a sum over all control points, n, of a rational B-spline $N_{i,p}$, times the control point coordinates, $\overrightarrow{P}_i$, times a weight, W_i, so that the coordinates of the blade profile are determined once the control points and the corresponding weights are specified. The NURBS curve is defined as [52]:

$$\overrightarrow{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) W_i \overrightarrow{P}_i}{\sum_{j=0}^{n} N_{j,p}(u) W_j} \tag{3.1}$$

Where $\overrightarrow{C}(u)$ are the x- and y-coordinates of the point on the curve, W_i is the corresponding weight, $N_{i,p}$ is the p_{th} degree B-spline basis function, and $\overrightarrow{P}_i$ are the x- and y-coordinates of control point i on the curve, which corresponds to u_i, the i^{th} element of the knot vector. The latter is determined using the chord length method [52]. The basis functions $N_{i,p}$ vanish everywhere except in the vicinity of point i, where the size of this vicinity depends on the order p. The weight W_i provides control on the curve attraction towards control point i. The NURBS are defined on the non-uniform parameters called knots, so that some of the control points affect a larger region of the curve while others affect a smaller region depending on the knot vector distribution.

The basis function, $N_{i,p}$, is given by the recurrence formula [52]. Letting $U = u_0, ..., u_m$ as a nondecreasing sequence of real numbers, defining u_i as knots and U as the knot vector, the i^{th} basis function of order $p+1$ (degree p), denoted by $N_{i,p}(u)$, is defined as:

$$N_{i,0} = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{3.3}$$

In order to compute the points on the NURBS curve, the control point as well as the basis function have to be defined. The control points can be set at any x- and y-locations, but fixing the basis function requires specification of the degree, p, and the knot vector, U.

The key feature of a NURBS curve is that its shape is determined/controlled by the set of control points and the corresponding weights. Moreover, placing and moving either one or more of the control points, the knots or the weights can accomplish either a local or a global change of the target shape. However, using the NURBS to model a desired shape is a very challenging task. A NURBS curve also represents exactly conics, e.g. circles, ellipses, hyperbolas, cylinders, cones. This implies that NURBS functions can represent a much wider family of curves compared with what B-splines or Bézier curves can represent, while simultaneously ensuring the profiles smoothness, confirmed by other researchers in the field as well [17], NURBS are becoming an industry standard tool for the representation and design of geometry; here are summary of the key reasons for the use of NURBS:

- They can represent exactly conics, e.g. circles, and provide the flexibility to design a large variety of shapes.
- They can be evaluated reasonably fast by numerically stable and accurate algorithms.
- They are invariant under affine as well as perspective transformations

- They are generalizations of B-splines and Bézier curves and surfaces.

In this section, geometric representation of existing blade with the least possible number of parameters for a given accuracy would be presented.

In the present research, NURBS functions have been selected and developed to represent several airfoil shapes used in blades of gas turbine engines. The objective of geometric parametrization is dual: in addition to parameterizing the geometry with a given accuracy, it was required to find the optimum parametrization that would minimize the number of geometric parameters to describe the blade profile.

This parametrization involves the solution of an inverse problem for the geometric parameters, where the error in the representation is minimized using an optimization algorithm- Simulated Annealing (SA), the latter gives the geometric parameters that would approximate a given blade shape up to a given tolerance.

3.2. Methodology

3.2.1 NURBS Functions

Consider the NURBS representation of a curve, as given by Eq. 3.1, with e.g. ten points; this representation will have 30 design variables, namely x, y, and W at each control point. This number of control points would correspond to 20 design variables, had we used B-splines or a Bézier polynomial. The reduction in the number of design variables in this case seems in favor of using B-splines however, Fig. 3.1 shows that with the same number of control points, a NURBS function represents a given curve more accurately and smoothly than a B-spline representation of the same curve. This result concurs with a similar observation made by Trépanier *et al.* [4] for airfoils used in wing sections. Figure 3.1 also suggests that a B-spline representation with

the same number of design variables (rather than same number of control points) as a NURBS representation will have approximately the same level of accuracy but not the same smoothness.

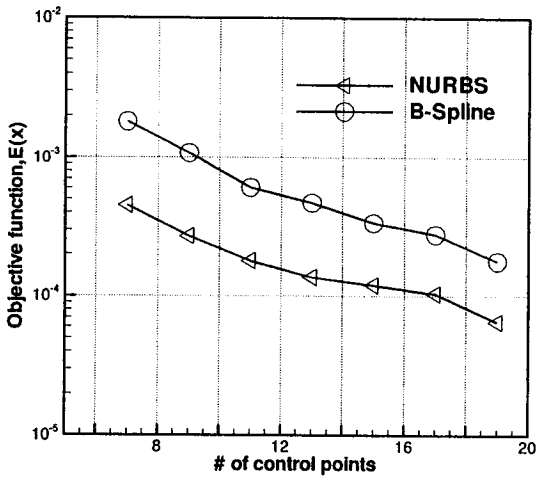


Figure 3.1: NURBS vs B-Splines accuracy for the generic compressor.

3.2.2 Optimization Objective

In this work, one would like to approximate the blade profile with the minimum number of NURBS parameters, given a certain approximation tolerance. This parameterizations involves the solution of an inverse problem for the control points and weights, where the error in the geometric representation has to be minimized. This error is taken to be the objective function $E(X)$ to be minimized, and can be defined in different forms. One way is the following:

$$E(X) = k\varepsilon_{ave} + \varepsilon_{max} \tag{3.4}$$

where

$$\varepsilon_{ave} = \frac{1}{m} \sum_{j=1}^{m} e_j$$

$$\varepsilon_{max} = MAX\ e_j \quad 1 \leq j \leq m$$

$E(X)$ is the objective function, X is the vector of design variables, which contains the control points and the corresponding weights, e_j is the distance between the original curve and its approximation computed at m locations, and k is a weight factor. The precision level required to represent a given shape is determined based on the manufacturing tolerances and/or the change in aerodynamic performance.

3.2.3 Optimization Scheme

The simulated annealing technique is implemented to solve the problem. It should be noted that the problem under consideration was found to be strongly nonlinear with local optima. When using a gradient-based optimization algorithm, the sequential quadratic programming (SQP) method, it was not possible to reach the allowable tolerance, which suggests that for the present problem, the SQP method was trapped in a local minimum.

Figure 3.2 shows the convergence history for the SQP method when applied to the generic compressor with 9 and 11 control points, where the objective function decreases by about one order of magnitude and levels off around 2.5×10^{-3}. On the other hand, when the SA method was used, a transition out of local minima into the neighborhood of the global minimum was possible.

In all cases tested, the SA gives approximately the same value for the objective function for different initial guesses of the design variables. However, these initial guesses are not arbitrary; they are based on the profiles curvature, as described below. This verifies that the solution is near the global optimal value.

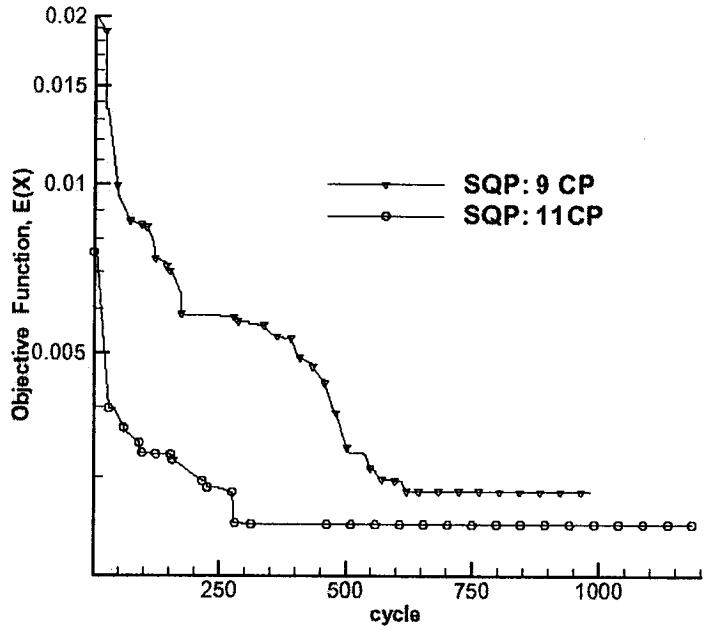


Figure 3.2: SQP convergence history for generic compressor with 9 and 11 control points.

3.2.4 Numerical Implementation

Given the blade shape as a list of x- and y-coordinates, the optimum NURBS function that would represent it is obtained as follows:

1. Some carefully selected points are chosen from the given data.

2. An initial guess is obtained for the NURBS control points assuming all the weights are set equal to one, this is simply a B-spline.

3. The position of the selected control points and the corresponding weights are optimized using SA so as to minimize the error between the representation and the original data.

The initial guess for the control points should be carefully selected since this problem is a strongly nonlinear one. It was found [4] that the best results were obtained when the selected data points are concentrated in areas of high curvature, typically in the round leading and trailing edge regions, where a high accuracy is also required. This initial guess is then obtained by fitting these selected data points with a non-uniform B-spline function, which is a NURBS but with weights equal to one. The control points can be found directly by solving a linear system of equations [52].

For any given blade shape, the optimum NURBS representation is obtained as follows. Using a relatively small number of control points, e.g. five or seven, the SA parameters controlling the cooling schedule are tuned such as to give the highest convergence rate for the optimizer. These SA parameters are then fixed and the SA optimizer is repeatedly run for gradually increased number of control points until the error in the objective function falls within the predetermined limit. The resulting set is the optimum one; it will be composed of the minimum number of control points (and corresponding weights) for which the objective function falls within the prescribed limit.

The precision level required to represent a given shape is determined based on the manufacturing tolerances and/or the change in aerodynamic performance. For a gas turbine blade, the manufacturing tolerance would be around 2×10^{-4} for a blade of unit chord.

The methodology, described in this section and the previous one, was programmed in C++ and the resulting program was used to calculate the results presented in the following section. The calculations were performed on an IBM Pentium 4 running at 1.6 GHz clock speed.

This implementation of the developed scheme starts with a validation of the methodology described above. Blade cascade profiles are then approximated with NURBS and are optimized with the proposed method. Generic as well as real blade shapes typical of turbine and compressor blades are represented and some of the results are presented.

3.3. Method Validation

The methodology and optimization scheme described above were first validated by representing a quarter circle. This case has an exact solution, where the quarter circle is exactly represented using NURBS function with three control points and weights. In principle, one should be able to drive the objective function to zero and recover the exact control points and weights. Since the first and last points are fixed, the design variables reduce to three, the x- and y -coordinates and the weight at the remaining point. The present methodology was able to approximate the quarter circle of radius one with a precision of $\varepsilon_{max} = 10^{-8}$.

Figure 3.3 shows the representation of the quarter circle with three control points. The maximum normal distance between the reference/original and approximated curve is 10^{-8} so that the two curves are seen as one on the figure.

The method is then used to represent different shapes of interest. In all cases tested, it was possible to represent the blade shapes with a tolerance of 2×10^{-4} with nine to thirteen control points for the generic blade shapes and eleven to nineteen for the actual blade shapes.

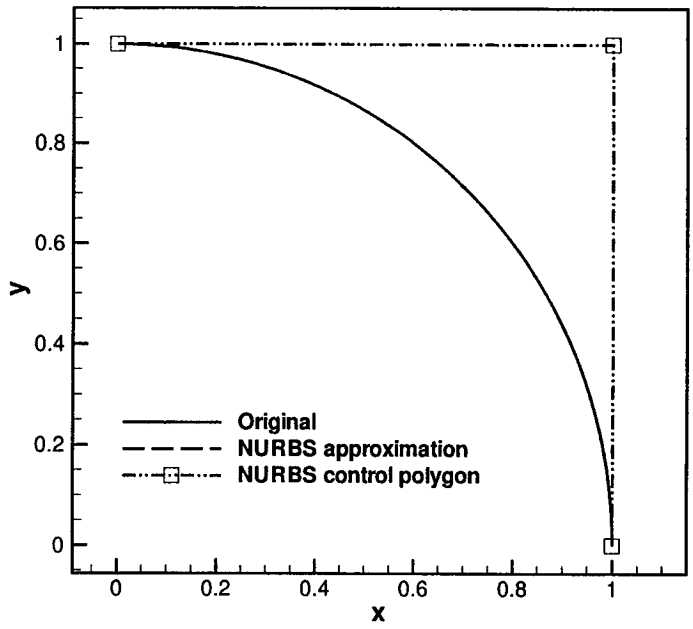


Figure 3.3: NURBS representation of quarter circle and the control polygon.

3.4. NURBS Representation of Generic Blade Cascades

In this section, generic shapes of gas turbine engine blades are represented by NURBS and are tested with the proposed method. These generic shapes are given by analytical expressions for the camber line and the blade thickness and represent typical blade features such as the inlet and exit blade angles, the maximum blade thickness, the LE and TE shapes (round or sharp). Two blade shapes have been tested, one typical of turbine blades and one typical of compressor blades. The

generic turbine blade is an inlet guide vane (IGV) that has a 25% maximum thickness with round leading and trailing edges, the inlet and exit blade angles are 0^o and 50^o, respectively and the spacing-to-chord ratio is 0.5. The generic compressor blade has 12% maximum thickness with a round LE and a sharp TE, the inlet and exit blade angles of -20^o and 20^o, respectively and the spacing-to-chord ratio is 0.5. The flow incidence angle in both cases is zero. These shapes are represented analytically with the following profile. Refer the Appendix A for the nomenclature of the blading.

$$y^{\pm}(x) = f(x) \pm \frac{1}{2}T(x) \tag{3.5}$$

where

$$f(x) = \tfrac{1}{2}(tan(\beta_2) - tan(\beta_1))x^2 + xtan(\beta_1)$$

and

$$T(x) = 2T_{max}(\sqrt{x(1-x)}) \qquad for\ \ round\ \ LE\ \ and\ \ TE$$

or

$$T(x) = \tfrac{3\sqrt{3}}{2}T_{max}\sqrt{x}(1-x) \qquad for\ \ round\ \ LE\ \ and\ \ sharp\ \ TE$$

The blade shape on the upper and lower blade surfaces is given by $y\pm$, f is the camber line, and T is the thickness distribution, T_{max} is the maximum thickness on the blade, x is the distance along the chord ,β_1 and β_2 are the blade angles at LE and TE respectively.

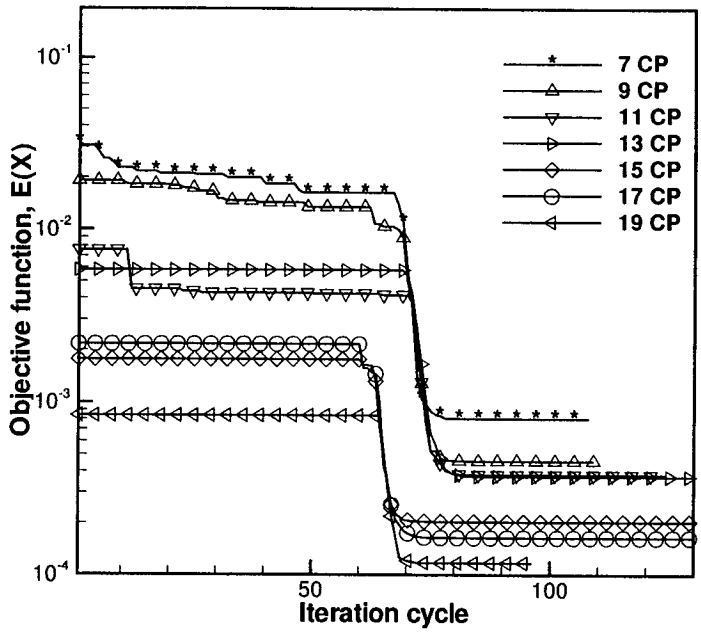
The generic compressor cascade, described by the analytic profile given above, was first used in the evaluation of the NURBS parametric representation. Different numbers of control points, as well as initial guesses, were used to obtain an optimum

representation of the blade shape. The optimization convergence history is given in Figs. 3.4 a and b, for different number of control points. As we increased the number of control points from 7 to 19, the objective function $E(X)$ decreased from 8.1×10^{-4} to 1.2×10^{-4}, at the cost of an increased computation time. The optimal representation for this case, i.e. the minimum number of control points for the given tolerance is obtained with nine control points. Figure 3.5 provides the initial and final blade profiles, and the control polygons (joining the control points) for the optimum compressor blade representation.

The accuracy of the representation can also be measured by the aerodynamic performance of the approximate shape as compared with the original one. The flow in the cascade was simulated using the JST scheme [53]. The latter is a cell-centered finite volume time marching scheme with a blend of nonlinear second and fourth order artificial viscosity. Similar H-grids were used for both approximate and original blade profiles. Each H-mesh has 151x51 grid points; in the streamwise direction there are 30 points upstream, 90 in the blade region and 30 downstream, and 51 grid points in the blade-to-blade direction. The boundary conditions used in the CFD simulations are: total pressure, total temperature, and inlet flow angle far upstream and the static pressure far downstream; both inlet and exit flows are subsonic. At present, the Mach number distribution along the blade surfaces was compared for the approximate and the original profiles.

Figure 3.6 shows that the Mach number distribution corresponding to the exact blade shape and the optimal representation of the same blade almost coalesce. This result suggests that the manufacturing tolerance restriction for this case is more stringent than the error in the aerodynamic performance associated with the approximation of the profile using NURBS.

The generic turbine IGV cascade, described by the analytical profile described above, was also used in the evaluation of the NURBS parametric representation. Figure 3.7 gives the final NURBS profile and the control polygon for the turbine IGV. The optimal number of control points for this cascade was found to be thirteen. Note that the turbine IGV has a round edge at both ends, which requires relatively more control points to resolve the high curvatures at both edges.



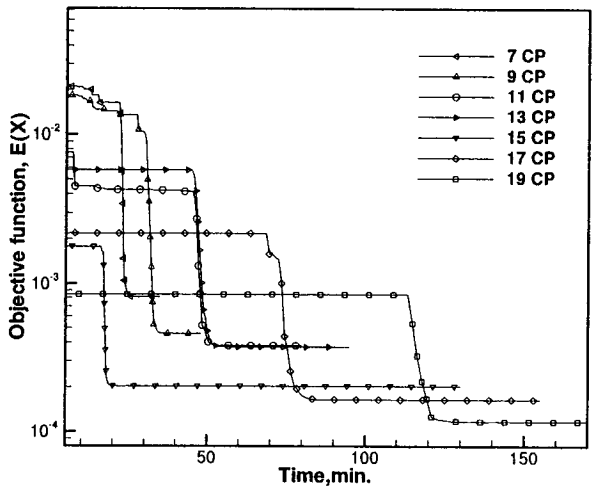


Figure 3.4: Convergence history for the generic compressor for different number of control points.

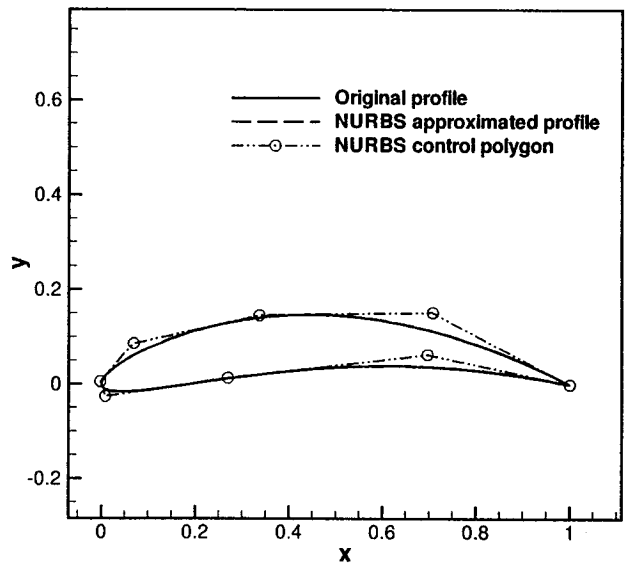


Figure 3.5: NURBS representation of generic compressor profile.

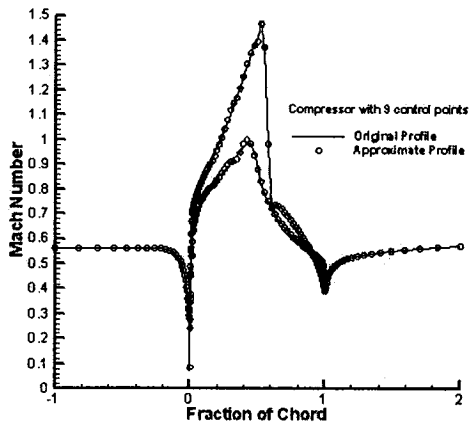


Figure 3.6: Generic compressor blade: Mach number distribution.

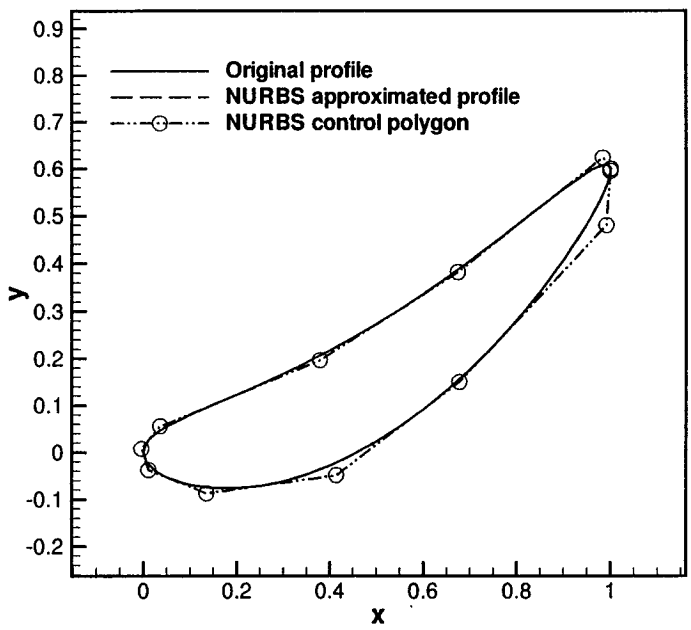


Figure 3.7: NURBS representation of generic turbine:IGV.

3.5. NURBS Representation of Sanz Transonic Compressor Cascade

The Sanz cascade is a transonic compressor cascade that is given in Fottner [54] as a test case for CFD code validation. It is given as a set of x-y pair of data values and is characterized by a round LE and a sharp TE. The spacing to chord ratio is 1.034, the inlet and exit flow angles are 30.81^o and -0.35^o, respectively, and the exit Mach number is 0.544.

The optimization method is repeated for this cascade and the optimal number of control points for this geometry was found to be eleven. Figure 3.8 shows the

final NURBS profile and the control polygon. The H-mesh is shown in Fig. 3.10 for two blade passages. Figure 3.9, where the isentropic Mach number along the approximate and original profiles is given, shows excellent agreement except in the shock-free supercritical flow region. This suggests that, for such shock-free supercritical regions, matching the aerodynamic performance may be more stringent than matching the manufacturing tolerance.

This blade, although similar to the generic compressor blade, required more control points to reach the same approximation tolerance. This is due to the fact that the Sanz profile is less smooth than the compressor generic profile (which is a smooth analytic profile). This smoothness can be measured by the profile curvature, which is given in Fig. 3.12 for the original and the NURBS approximated profiles. Note that the noise in the original data (probably due to including the displacement thickness in the profile) is not present in the NURBS approximation. This lack of smoothness possibly explains the need for eleven (rather than nine) control points to adequately represent this cascade.

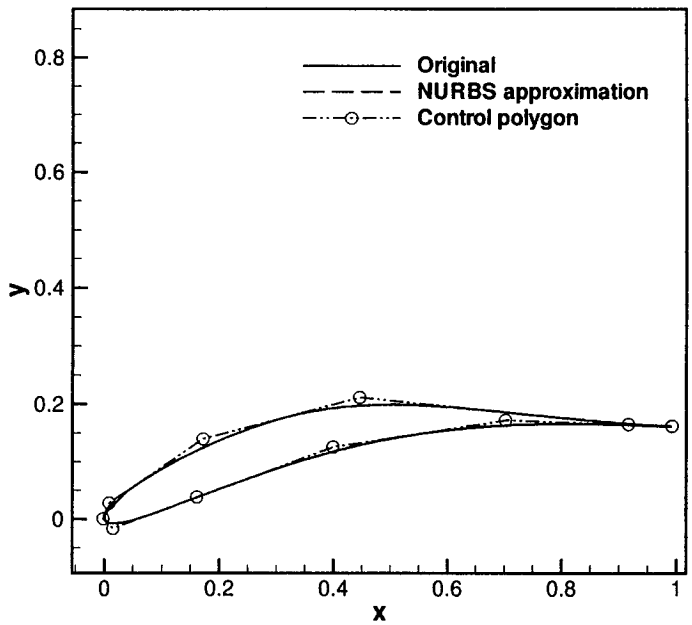


Figure 3.8: NURBS representation of SANZ compressor.

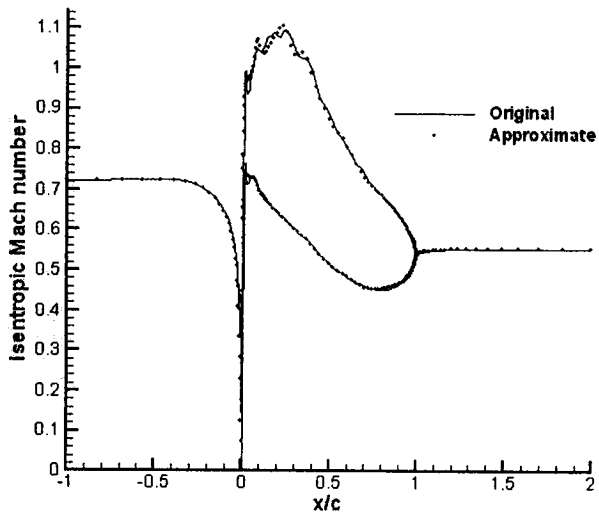


Figure 3.9: Sanz transonic compressor Mach number along the blade.

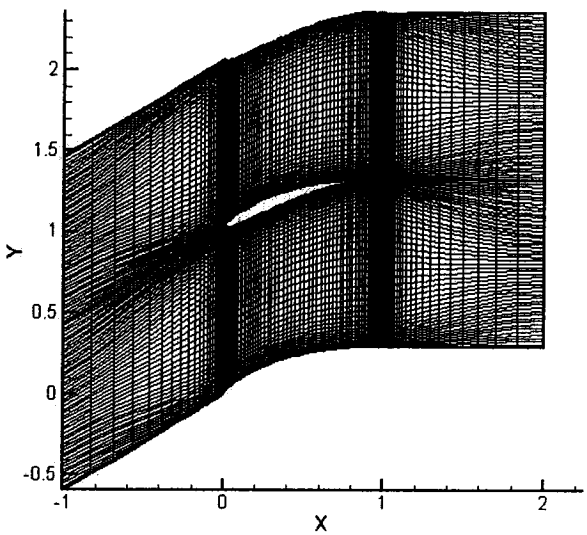


Figure 3.10: Sanz transonic compressor:Mesh.

3.6. NURBS Representation of DFVLR Subsonic Turbine Cascade T106

The DFVLR-T106 cascade is a subsonic turbine cascade that is also given in Fottner [54] as a test case for CFD code validation. It is given as a set of x-y pair of coordinates and is characterized by round leading and trailing edges. The spacing to chord ratio is 0.6869, the inlet and exit flow angles are 37.7^o and -63.2^o, respectively, and the exit Mach number is 0.59. This case was simulated as a laminar flow case where Reynolds number is equal to 500.

The optimal number of control points for this cascade was found to be nineteen. Figure 3.11 shows the final NURBS profile and the control polygon. Again the lack of smoothness of this profile compared with the generic turbine profile explains the need for nineteen (rather than thirteen) control points to adequately represent this cascade. Figure 3.13 shows the excellent agreement between the isentropic Mach number along the original and the NURBS-approximated blade profiles.

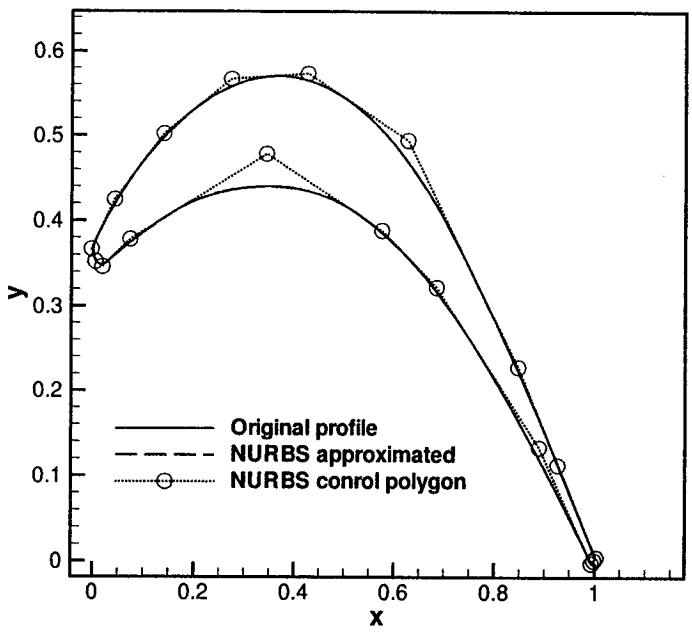


Figure 3.11: NURBS representation of DFVLR-T106 subsonic turbine.

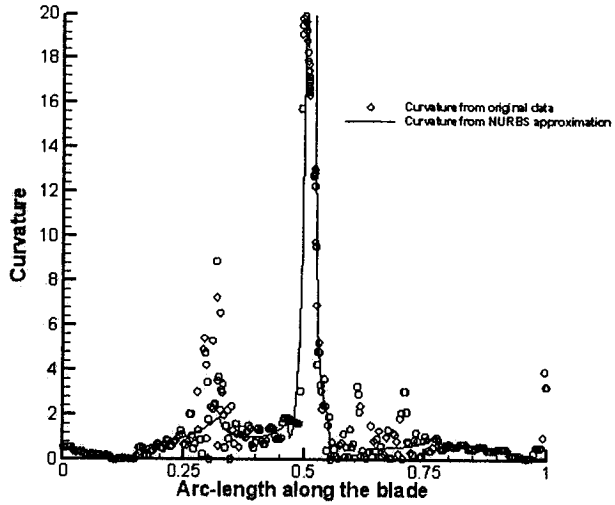


Figure 3.12: Sanz transonic compressor: Original vs NURBS curvature.

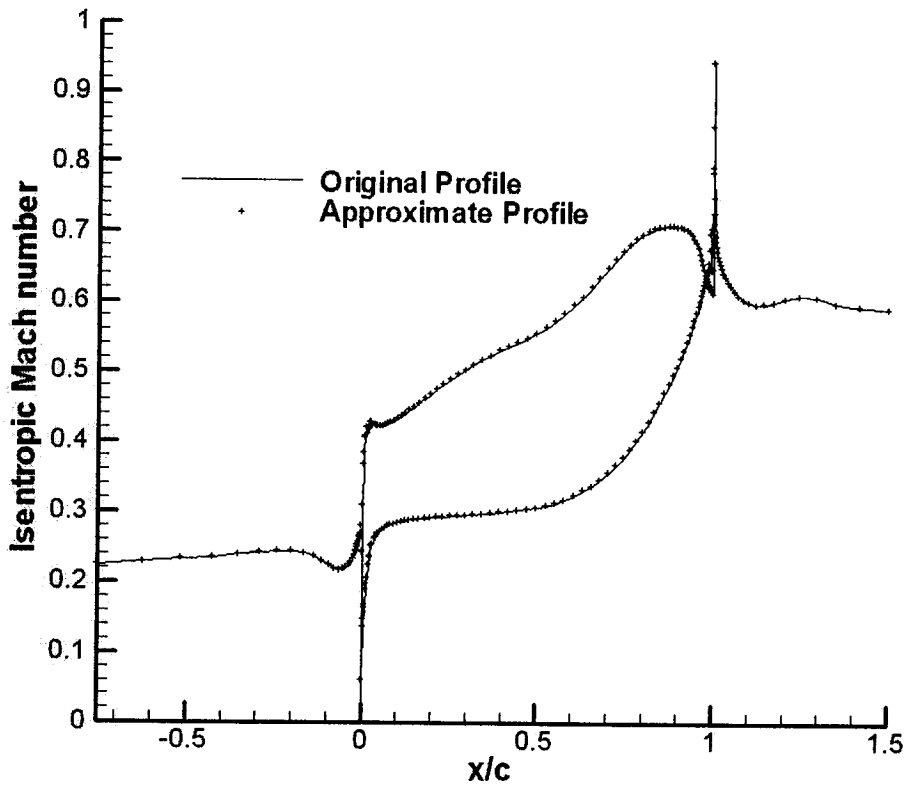


Figure 3.13: DFVLR-T106 subsonic turbine: Mach number distribution.

Chapter 4

OPTIMIZATION OF GAS TURBINE CASCADES

4.1. Introduction

In this part of the work, an optimization method is developed and tested for two-dimensional cascades of compressor and turbine blades with the goal of improving their aerodynamic performance either at the design point or over the full range of operation by properly reshaping their profile, which is parameterized using a NURBS approximation. The performance is measured in terms of the cascade adiabatic efficiency, total pressure ratio, and/or total pressure loss. The present work uses Genetic Algorithm (GA) as the optimization scheme and computes the objective function and constraints from the flow CFD simulations. Parallel processing is also implemented to reduce the computation turn-around time. The validity and effectiveness of the developed optimization method is demonstrated (see Appendix C) and is applied to the redesign of different turbine and compressor cascades.

4.2. Design Methodology and Choice of Objective Function

A common aerodynamic shape design problem is to achieve a better performance of gas turbine cascade with the minimum number of blades over the entire operating range. The objectives can then be defined as e.g. maximizing the adiabatic efficiency, minimizing the total pressure loss and/or minimizing the blade count for a given set of flow conditions, aerodynamic and structural constraints. One way to accomplish these objectives is to vary the blade spacing and the blade shape (which is described by the blade camber line and thickness distribution) while fixing the inlet and exit flow boundary conditions as well as constraining the minimum blade thickness and cross-sectional area for structural reasons.

In any optimization problem, the choice of the objective function affects the optimization process as well as the results. Thus, a careful and well-studied identification and formulation of that function is crucial. For example, the overall aerodynamic performance of a compressor rotor is measured by its adiabatic efficiency and total pressure rise at design and at off-design conditions. Therefore, one can choose to maximize the efficiency or the pressure ratio or both; this can be done either at the design point only or on the full operating range. With this design strategy in mind, the objective function is constructed such that it can serve for a single point or multi-point optimization case, and can be defined as follows.

$$F_{obj}(X) = Min\left[C_1\sum_{i=1}^{n}(1-\eta_i) + C_2\sum_{i=1}^{n}\sum_{j=1}^{n}|\eta_j-\eta_i| + C_3\sum_{i=1}^{n}\left(1-\frac{PR}{PR^o}\right) + C_4\sum_{i=1}^{n}\omega_i + C_5\sum_{i=1}^{n}\sum_{j=1}^{n}|\omega_j-\omega_i| + PT\right] \quad (4.1)$$

Where X is the vector of design variables, which includes the shape parameters that control the blade profile as well as the backpressure. Varying the backpressure in the pre-determined range from the choke to the stall limits while fixing the rotor speed allows for tracing a speed line, it allows for design and off-design calculations that correspond to different mass flow rates with varying efficiency and pressure ratio.

The first, third and fourth terms in the objective function, Eq. 4.1, attempt to maximize the efficiency (η), the pressure ratio(PR) and minimize the total pressure loss coefficient(ω) at the design and off-design points,respectively. The second term eliminate large difference in efficiency and the fifth term eliminates large difference in total pressure loss at the design and off-design points which would tend to keep them constant over the entire operating range. The pressure ratio PR is divided by the maximum pressure ratio PR^o so that it would be scaled between 0 and 1. The last term in the objective function is a penalty term, PT that accounts for the mechanical, geometrical and aerodynamic constraints imposed on the optimization process. The constraints could include the exit flow angle deviation, the spacing to chord ratio, and the stall margin.

The summation is carried out over n pre-selected operating points. In this work, these pre-selected points are the design point and three off-design points, i.e. $n = 4$. The weights C_k, $k = 1 - 5$, are prescribed by the designer, they are determined such that the different components of the objective function have the desired influence on the optimization process.

Note that the current choice of the objective function, given in Eq. 4.1, allows for the implementation of different design options depending on the values given to the C_k coefficients. Some of the options are given below:

1. Single point design: $n = 1$, $C_2 = C_5 = 0$

2. Multi-point design: $n > 1$

3. Multi-point maximum efficiency design: $n > 1$ and $C_k = 0$ for $k = 3, 4, 5$

4. Multi-point maximum pressure ratio design: $n > 1$ and $C_k = 0$ for $k = 1, 2, 4, 5$

5. Multi-point minimum total pressure loss coefficient design: $n > 1$ and $C_k = 0$ for $k = 1, 2, 3$

The objective of the present task of aerodynamic shape design optimization is to design and optimize a turbine or a compressor cascade for improving the aerodynamic performance at the design point and/or at off-design conditions.

In the case of redesign or retrofit of an existing blade, this can be achieved by minimizing the total pressure loss across a two-dimensional linear cascade with the modification of the blade camber line and thickness distribution while fixing the mass flow rate and the flow turning. The redesign of an existing blade requires the enforcement of a number of specific constraints. The original and the new blades should have the same axial chord so that it may fit into the existing turbomachine (see Appendix A for the blade nomenclature). The inlet and exit flow angles should be the same in the redesigned blade as in the original one otherwise the velocity triangles will not match with the neighboring blade rows. The mass flow rate through the redesigned blade row also must be the same as in the original blade row otherwise the turbomachine will perform at an off-design mass flow rate which can lead to a serious drop in efficiency and create performance problems. The spacing between the blades must be kept fixed so that the total number of blades in the turbomachine is maintained. The thickness distribution for the redesigned airfoil is chosen to be the same as the original airfoil to avoid an unacceptably thin blade and to fix the section area so that it will be able to sustain the expected loads without performing

a detailed elasticity analysis of the blade geometry. Other constraints that were directly implemented in the CFD simulation include fixing the inlet flow angle, the inlet total pressure and temperature, the exit static pressure, as well as the blades axial chord and spacing. A numerical optimization method using a Genetic Algorithm (GA) was implemented and the objective was calculated based on the flow field analysis solver. The blade camber line and thickness distribution were parameterized with non-uniform rational B-splines (NURBS) so as to keep the number of design variables, namely the NURBS control points, relatively small while achieving high flexibility and robustness.

4.3. Geometric Representation

In this work, the blade profile is defined using the mean camber line and the tangential thickness distribution as shown in Fig. 4.1. The camber line strongly influences the flow turning hence the blade loading distribution while the thickness distribution allows for a direct implementation of the structural constraints such as minimum blade thickness. Either the camber line or thickness distribution or both have to be modified to achieve the objective. Both profiles are parameterized using NURBS function described in the previous chapter. The camber line and the thickness distribution were represented with 7 to 9 control points. The y-coordinates and/or the weights of these control points were used as design variables.

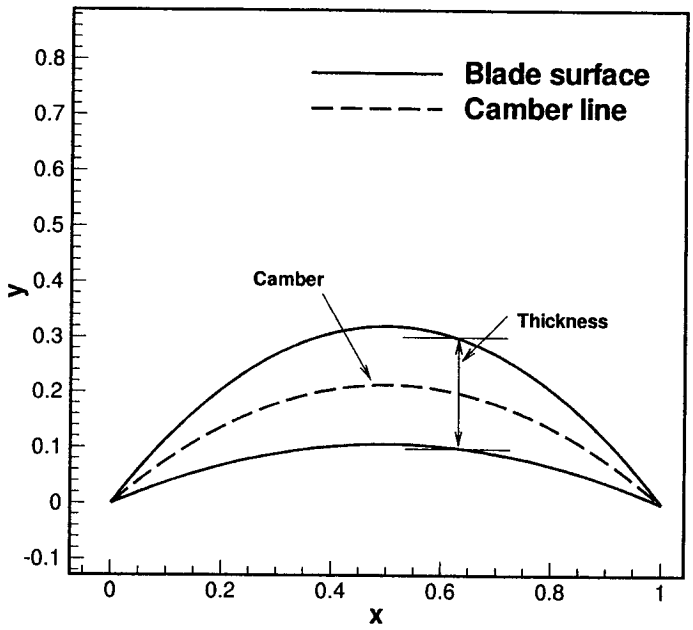


Figure 4.1: Blade geometry description.

4.4. Optimization Algorithm

A global optimization technique was used to design the airfoil shape by varying the geometric representation parameters described in the previous section. The objective of the optimization given by Eq. 4.1 is to determine the airfoil shape that satisfies the chosen objectives while conforming to the specified constraints. An implementation of the GA was used in this work as described in Ch. 2. The following parameters are used as default values in the GA algorithm: 32 individuals in each generation, the crossover probability is 0.75, the mutation probability is 0.06, the number of elitism is 2, and the maximum number of generations is 100.

These values have been selected after performing numerical experimentation on a typical multi-modal optimization test problem (see Appendix C), as the aerodynamic optimization problem is expected to show multi-modality behavior.

4.5. Flow Field Analysis

The CFD algorithm described in Sec. 2.3 was used to simulate the flow field. The flow at the inlet and exit planes, which are placed one chord upstream and downstream of the cascade, is always subsonic for the case presented. At the exit plane, the exit static to inlet total pressure is specified. The boundary conditions at inlet that can be imposed in this CFD code are total pressure, total temperature and inlet flow angle, or reduced mass flow rate, total temperature and inlet flow angle. The first set of inlet boundary conditions is used in the constrained optimization where the objective function is penalized by delta mass flow rate, while the second set is used in the optimization with no constraints, since all geometry and flow requirements are explicitly specified.

4.6. Aerodynamic Optimization Process

The aerodynamic optimization process starts with the geometric parametrization of the blade geometry using NURBS functions. The coefficients of this representation, which are used as design variables are varied by the numerical optimization method, GA, to generate a generation of blades for which the objective function is computed based on either a CFD simulation of the flow or ANN. This objective function is then used in the GA calculations to generate the next generation of geometries and so on until the optimization process converges. Figure 4.2 shows a flow chart of the

aerodynamic optimization process.

When the objective function is evaluated from ANN then the optimization process is slightly modified such that an initial pool of candidates that are feasible and cover the design space is generated and then the objective function is calculated for these candidates and is used in the training and testing of the ANN. Figure 4.3 shows a flow chart of constructing the ANN model as well as using the generated model in the aerodynamic optimization process.

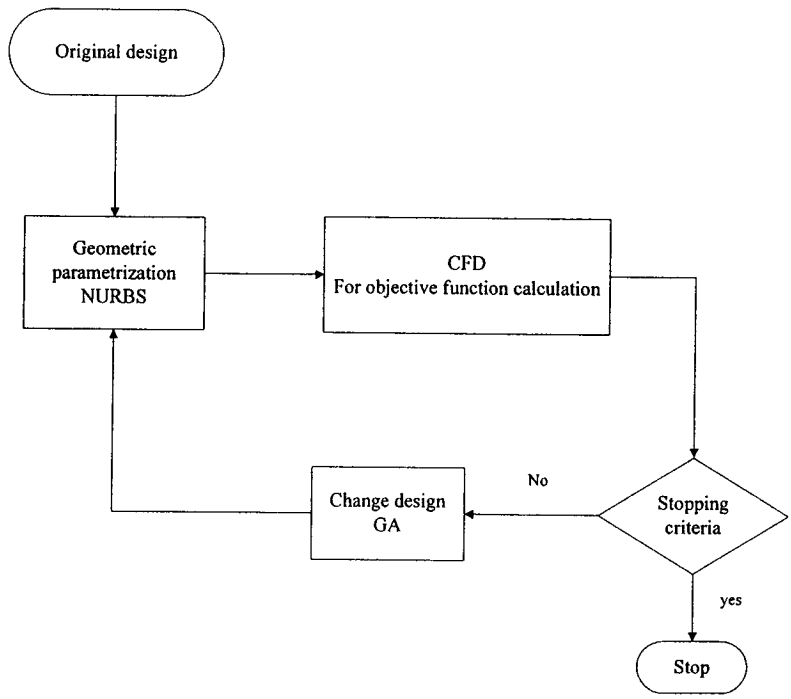


Figure 4.2: The aerodynamic optimization implementation.

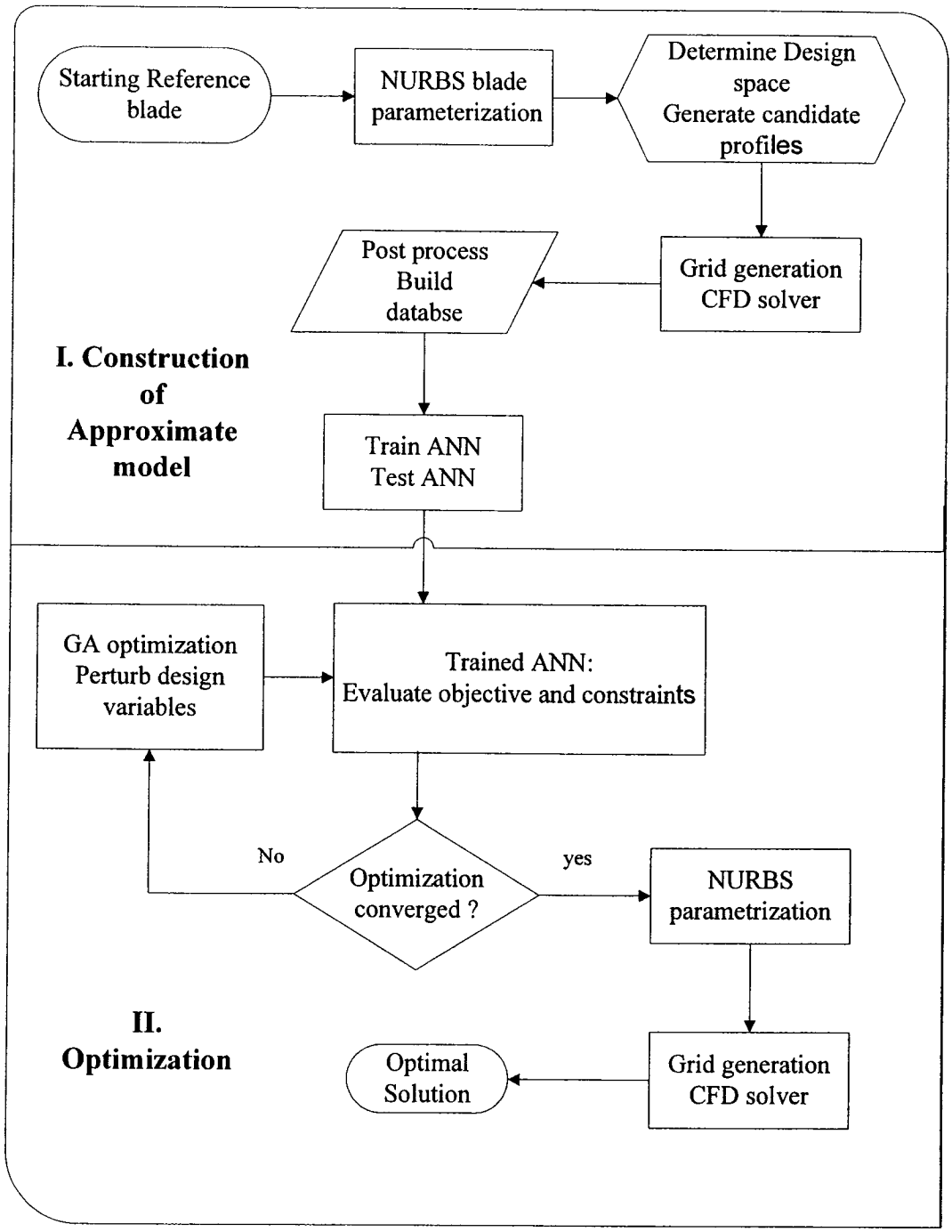


Figure 4.3: The ANN-based aerodynamic design optimization process.

4.7. Single-Point Optimization

The developed algorithm was used to design an impulse turbine cascade given in [3] and the Onera compressor cascade described in [54]. Both cascades are transonic and they were redesigned for minimum design point loss.

One important issue in the design process is how to choose the design variables and reduce their number to a minimum while maintaining the freedom and quality of the blade shape representation. The design variables are the y-coordinates or/and weights of the NURBS control points that are used in the blade geometric parametrization. This parametrization ensures good continuity of the blade profile, and the parameters defining the geometry have intuitive meaning that facilitates imposing constraints on their variation so as to restrict the design space. The cost function is the design point total pressure loss subject to several aerodynamic and mechanical constraints, which include fixed mass flow rate, inlet and exit flow angles, exit static pressure, chord and spacing, and thickness distribution. The constants defined in Eq. 4.1 are zero except for C_4, and the objective function is penalized with delta mass flow rate and delta exit flow angle (to enforce a given mass flow rate and flow turning). The other constraints are directly enforced through the prescribed inflow boundary conditions imposed in the CFD code, which are the total pressure, total temperature and inlet flow angle as well as the static pressure at the outflow boundary. The objective function for this case is given by:

$$F_{obj}(X) = Min[\omega + K_1(\Delta\dot{m}) + K_2(\Delta\beta)] \tag{4.2}$$

Where $\Delta\dot{m}$ is the difference between computed and target mass flow rates and $\Delta\beta$ is the difference between the computed and the target exit flow angles (in degree). The weights K_1 and K_2 are user specified penalty coefficients; they are chosen so as

to have equal penalizing effect on the objective function. In this case, they take the following values:

$$K_1 = 1000 \quad when \ |\Delta\dot{m}| > 0.01, \ and \ 0 \quad otherwise \tag{4.3}$$
$$K_2 = 10 \quad when \ |\Delta\beta| > 1.0, \ and \ 0 \quad otherwise$$

Note that a fixed mass flow rate can be imposed as an inlet condition to the CFD flow simulation; in that case K_1 is set to zero.

In order to redesign a given blade, the following approach is used. Initially we have a transonic blade which has a shock formed on the blade suction side. The original blade profile, which is described by its camber line and tangential thickness distribution, is taken as the initial design. The camber line is represented by NURBS using between 7 to 11 control points whereas the thickness distribution remains fixed.

The optimization algorithm generates a new blade profile by changing the NURBS control points for the camber line. The unstructured grid is generated for this new blade geometry, Euler equations CFD solver is then used to simulate the flow and compute the total pressure loss. These steps are repeated until the total pressure loss is reduced to a minimum (Theoretical limit is zero, however due to the presence of artificial viscosity in the CFD code, this limit cannot be achieved exactly.), which implies that the shock is either eliminated or at least weakened.

The transonic impulse turbine and Onera compressor cascade are redesigned using the current optimization procedure, and results are presented below.

4.7.1 Redesign of Transonic Impulse Turbine Cascade

This case is optimized using two different objectives. The first objective is minimization of the total pressure loss while the second objective is to match a given

pressure distribution on the blade surfaces.

Optimization for minimum loss

In this subsection, the impulse turbine is redesigned to minimize the total pressure loss so that the objective function takes the form discussed in Sec. 4.7.

For this cascade, the spacing to chord ratio is 0.526 and the thickness distribution assumes a parabolic profile with maximum thickness to chord ratio of 21.45% occurring at mid-chord. The inlet flow angle is 40.63° and the ratio of exit static to inlet total pressure is 0.833.

The optimization history is given in Fig. 4.4 where the best candidate in each GA generation is given. The total pressure loss is reduced from 0.0043 to 0.0019 in three generations for the unconstrained optimization, and from 0.0043 to 0.0026 in five generations for the constrained optimization; each generation consists of 32 individuals. A single flow field analysis took approximately two minutes of CPU time (when starting from a converged solution obtained for a given candidate of the previous generation) with a CFL number of 4 on a mesh with 2800 points.

Figure 4.5 shows that the flow over the original cascade exhibits a shock located around mid-chord; this shock has been eliminated in the optimized cascade. Figures 4.5 and 4.7 give the results for an unconstrained optimization, whereas Figs. 4.6 and 4.9 show the results for a constrained optimization case. Note that the Mach number level for the latter case is the same in both original and optimized cascades. Figure 4.8 shows that the changes in shape between the original and the redesigned blades are relatively small, which would be difficult to achieve by manually changing the blade shape. Note also that both designs have a reversed curvature particularly along the suction side to allow the flow to compress reversibly. Other researchers [3; 55] have also observed similar behavior.

In order to assess the flexibility of the developed design method, the thickness distribution was allowed to vary by few percents from the original one. A very close but smoother and better design has been achieved with insignificant change in the computing time. The optimized Mach contour and distribution along the blade surfaces are shown in Figs. 4.10 and 4.11.

This cascade was also optimized using the SA algorithm. The SA algorithm used about 25% less function evaluations to eliminate the shock by modifying the blade geometry and the resulting blade shape has a reversed curvature along the blade suction side so that the flow can be compressed reversibly. The blade shape and corresponding isentropic Mach distribution along the blade resulting from the SA algorithm, which are given in Figs. 4.12 and 4.13, clearly show that the blade shape obtained using SA is not as smooth as the one obtained with GA; moreover the mach distribution resulting from SA although transonic and shock free, is far from being smooth.

This case reflects the fact that GA is more powerful in exploring the design region and is best suited for large scale problems where the number of design variables and the size of the design space are relatively large, while the SA method seems to be more accurate and quick for relatively low number of design variables [8]. Thus GA was adopted as the preferred optimization scheme in the cases reported in this Chapter.

Inverse Design Using Optimization

In this section, the optimization problem is formulated such that the optimization objective is to reach a target pressure distribution along the blade so as to mimic the inverse design problem. The objective function is the least-square sum of differences in pressure between the target and the current design at a fixed number of points

on the pressure and suction sides as shown below:

$$F_{obj}(X) = \sum_{i=1}^{n} (p - p^*)_i^2 \tag{4.4}$$

The design variables X are the y-coordinates and weights of the NURBS control points, p and p^* are the calculated and target pressures, respectively, at a preselected set of n points on the blade surfaces.

The convergence history of the optimization by GA is shown in Fig. 4.14. It took about 1000 generations each of which consists of 32 individuals to reach the optimum. The computation was done in parallel using 16 processors so that the wall time for each generation is about 3 minutes. The optimum and target pressure distributions along the blade surfaces are shown in Fig. 4.15. The developed design algorithm seems to be flexible enough to handle different optimization objectives, though the problem was relatively stiff and took a large number of GA generations to converge.

4.7.2 Redesign of Onera Compressor Cascade

This cascade is a highly loaded transonic compressor cascade tested at ONERA [54] with a detached strong shock that impinges on the blade suction side. The blade is optimized for minimum total pressure loss with fixed mass flow rate and exit flow angle as constraints, as given in Eq. 4.2 and discussed in Sec. 4.7.

The inlet flow angle is 39^o, and the exit flow angle is -24^o. The exit static pressure is 0.77 times the inlet total pressure.

Figures 4.16 and 4.17 show the Mach number field where a shock that occurs in the original design was removed in the optimized one. The total pressure loss was reduced from 0.0045 to 0.0017 in five generations of the GA optimization with a

population of 32 in each generation. The modified and original blade profiles, shown in Fig. 4.18, suggest that the change in blade geometry is concentrated in the first quarter chord where the shock occurs.

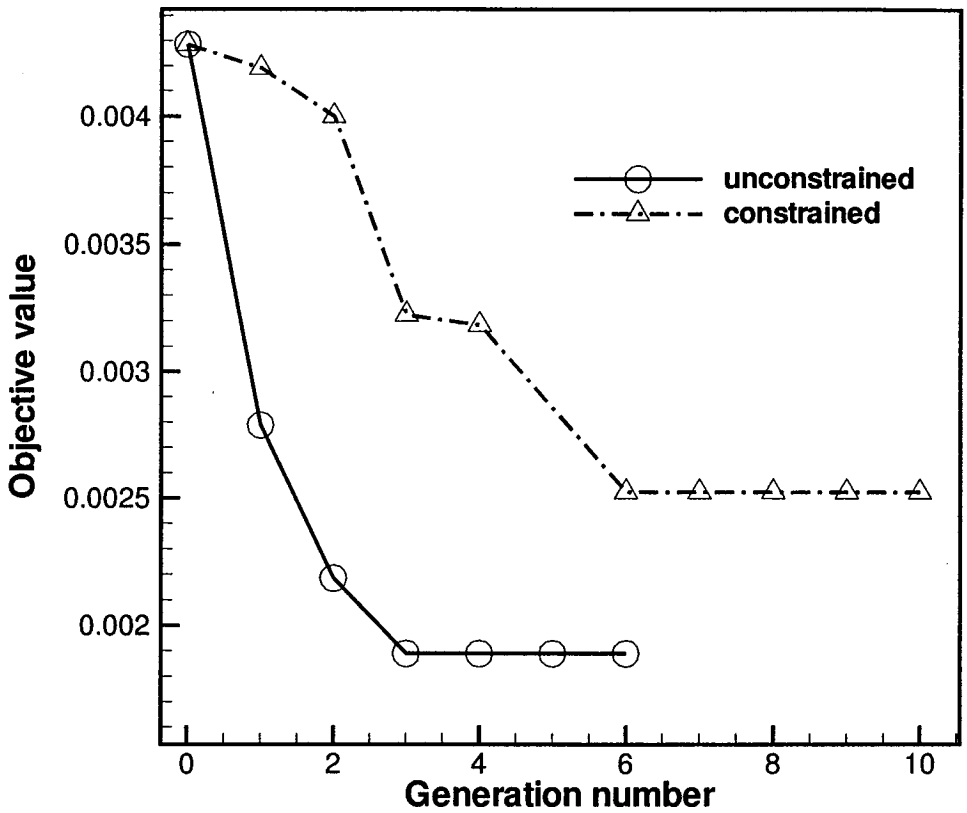


Figure 4.4: GA convergence history for the impulse turbine cascade.

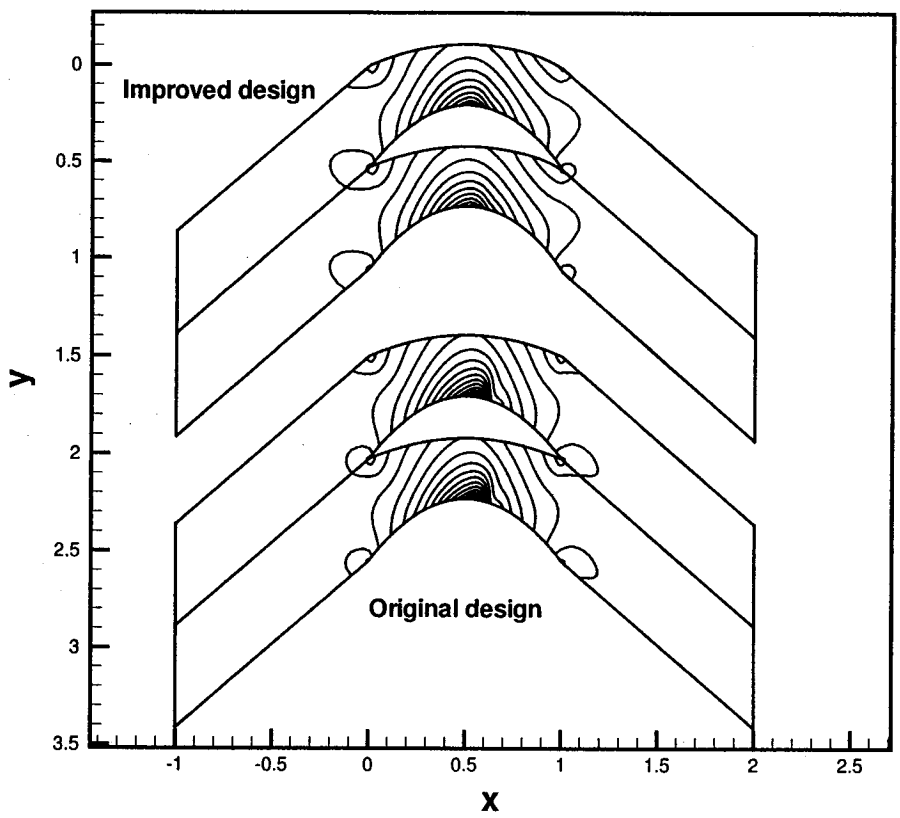


Figure 4.5: Isentropic Mach contours for unconstrained case.

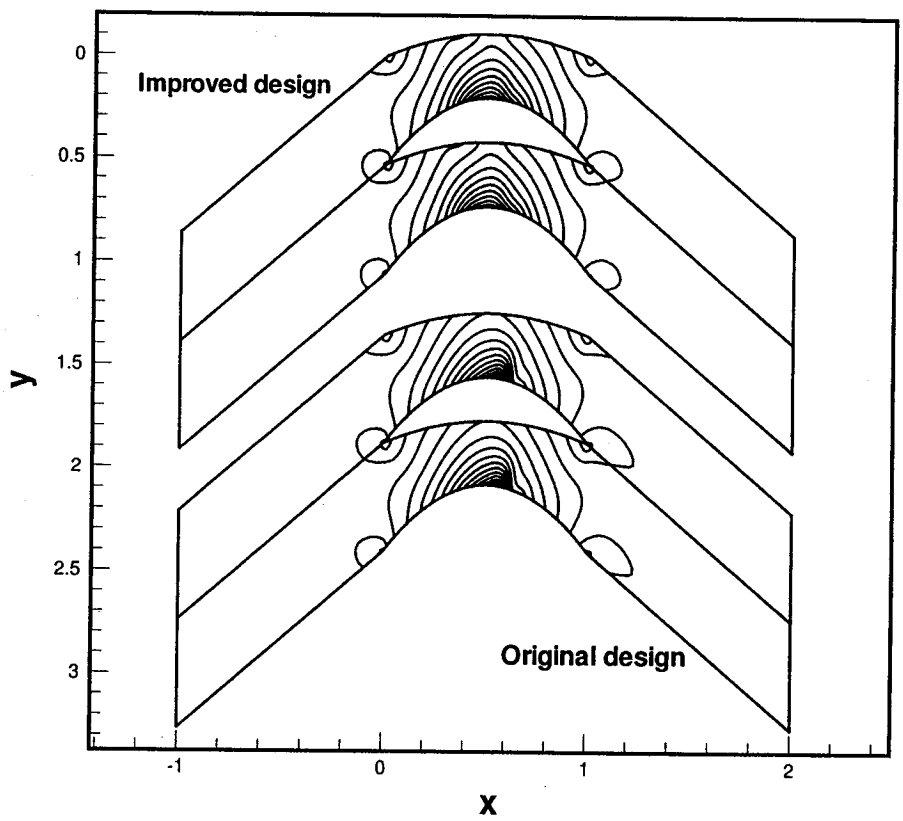


Figure 4.6: Isentropic Mach contours for constrained case.

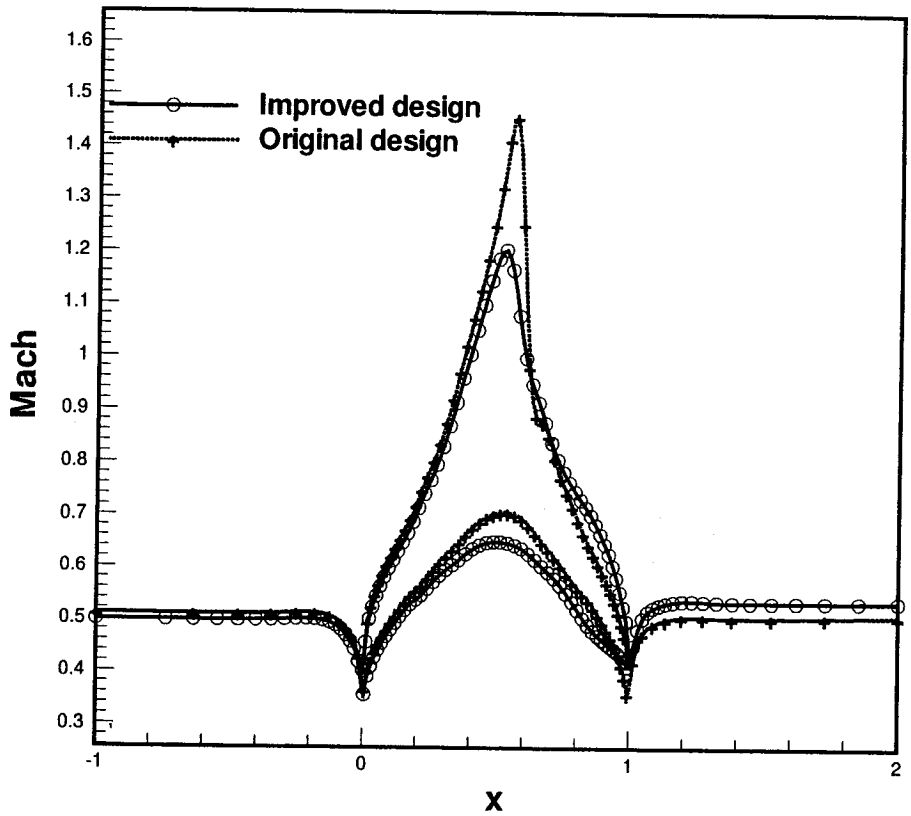


Figure 4.7: Isentropic Mach along the blade for unconstrained case.

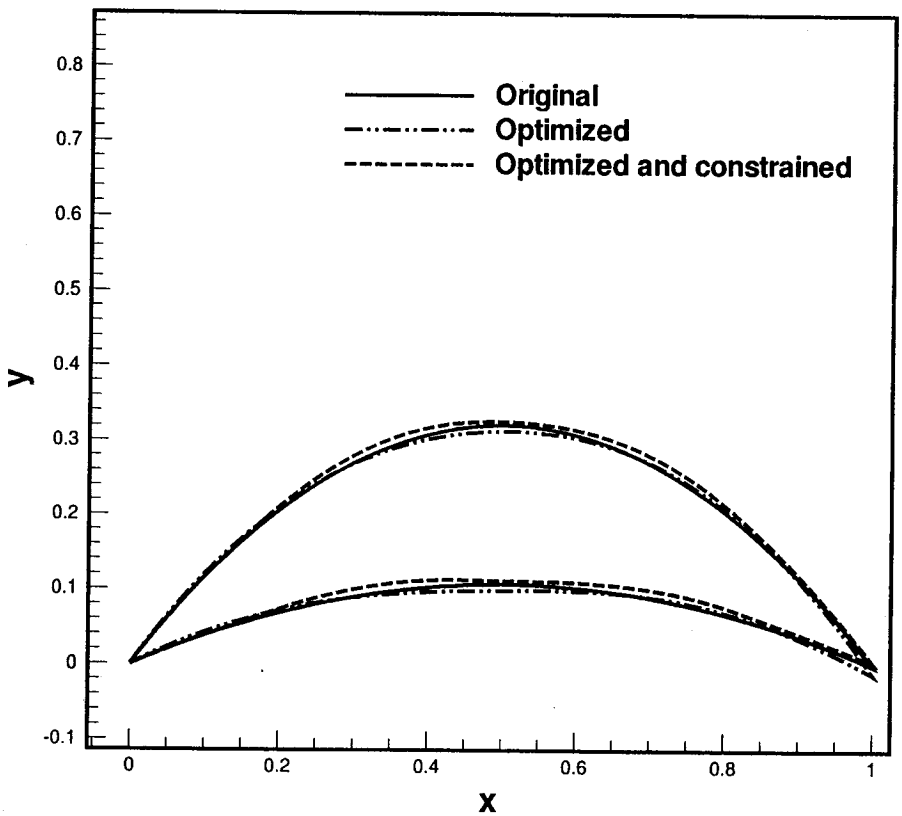


Figure 4.8: The impulse turbine cascade.

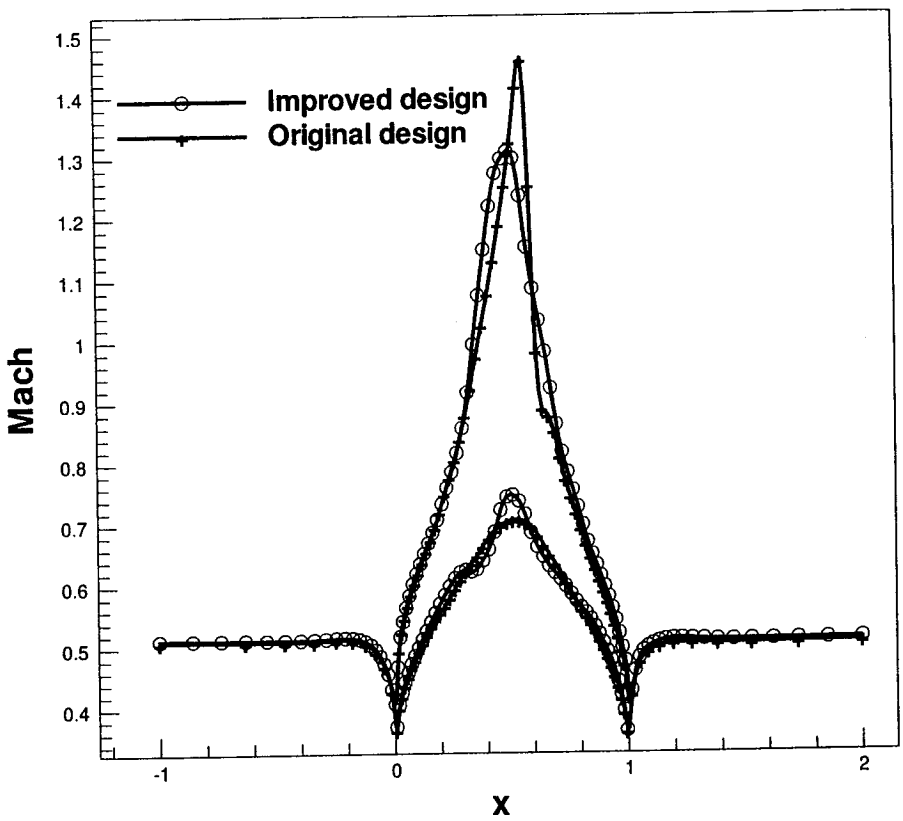


Figure 4.9: Isentropic Mach along the blade for constrained case.

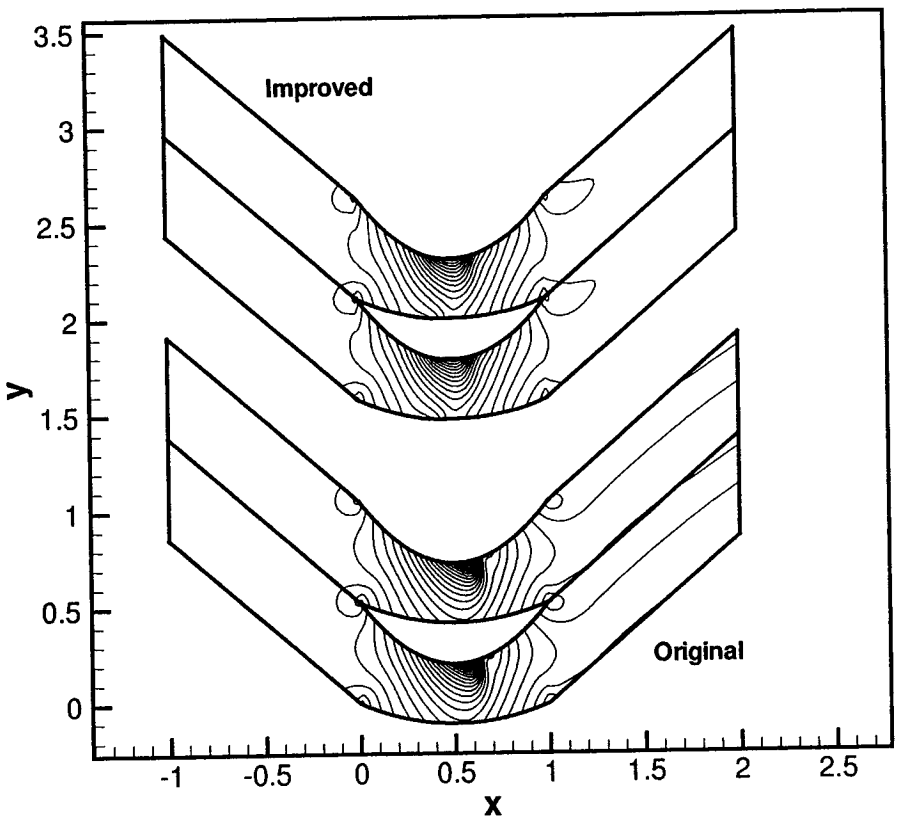


Figure 4.10: Isentropic Mach contours.

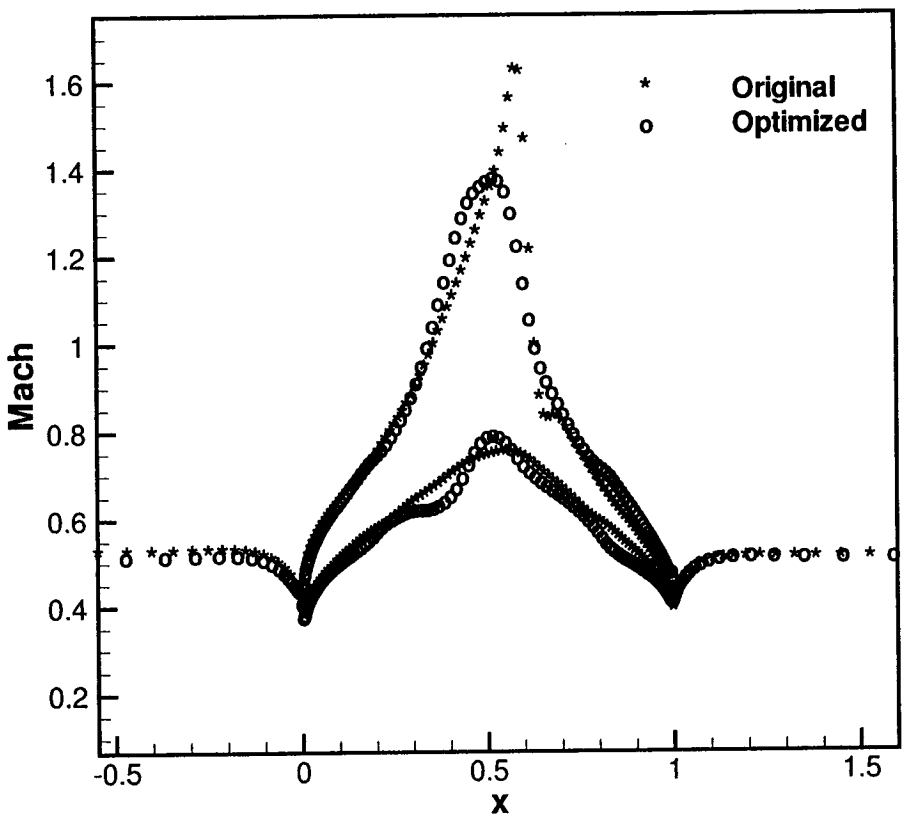


Figure 4.11: Isentropic Mach distribution.

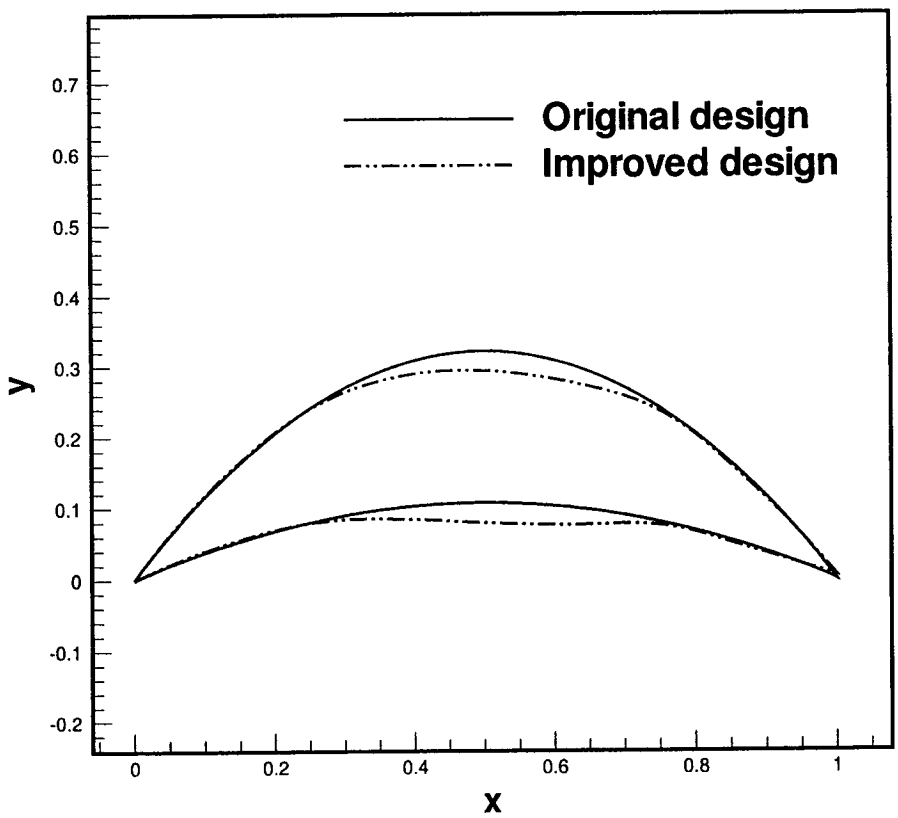


Figure 4.12: Impulse turbine profile resulting from the SA algorithm.

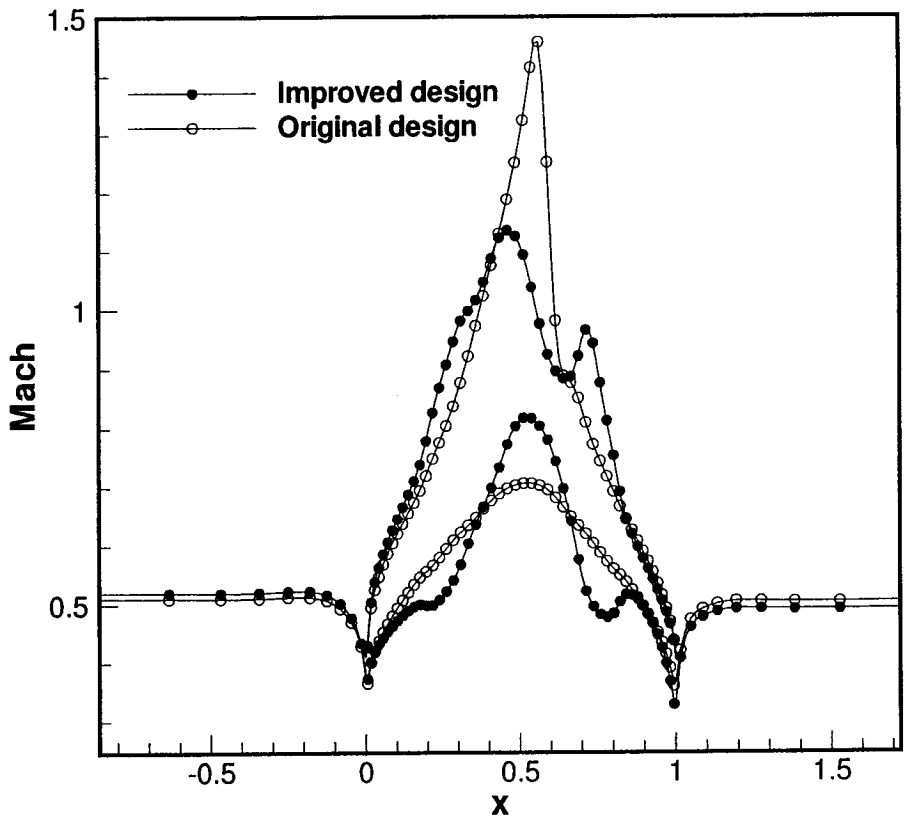


Figure 4.13: Isentropic Mach distribution along the blade resulting from the SA algorithm.

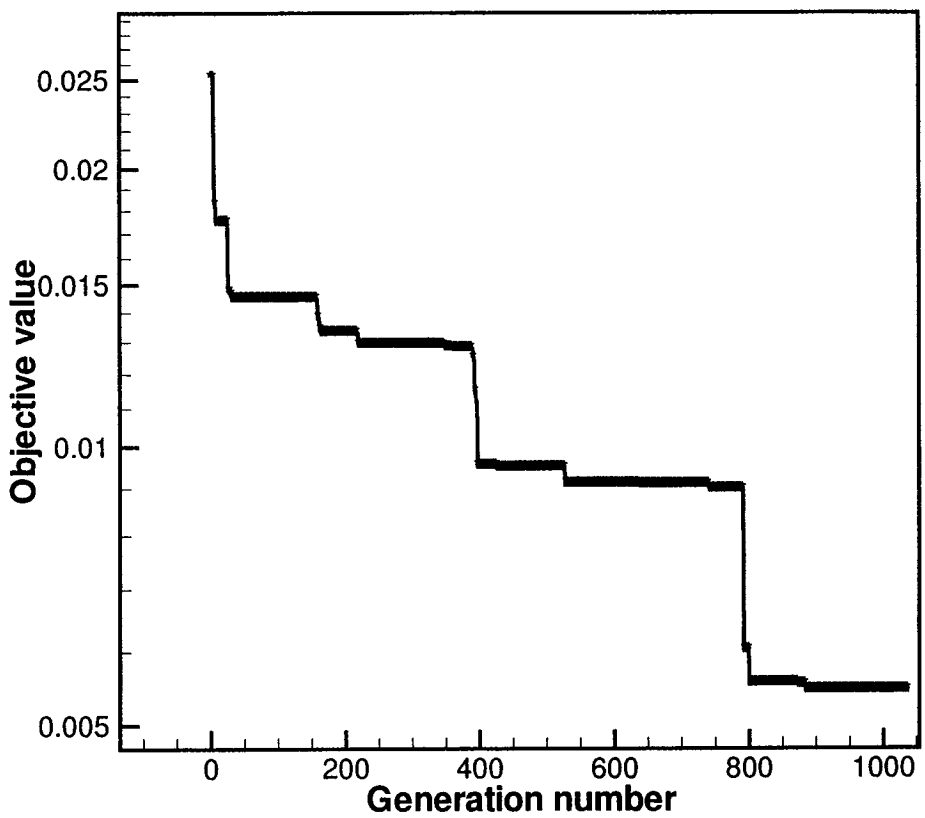


Figure 4.14: Convergence history for the impulse turbine.

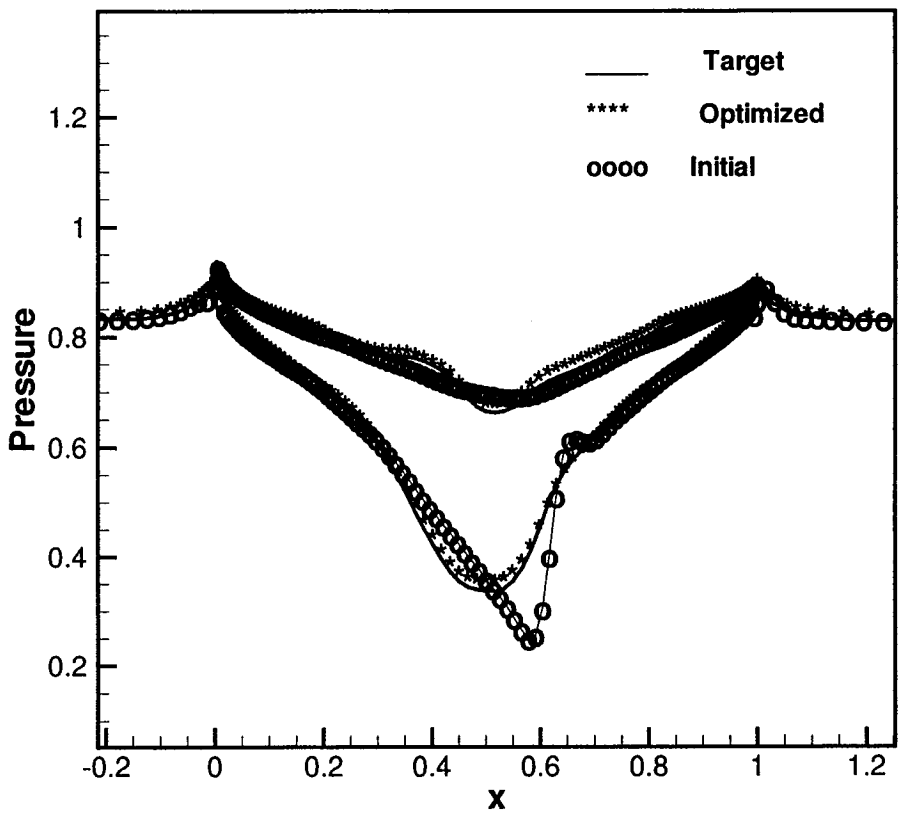


Figure 4.15: Pressure distribution on the blade for the initial, target and optimized design.

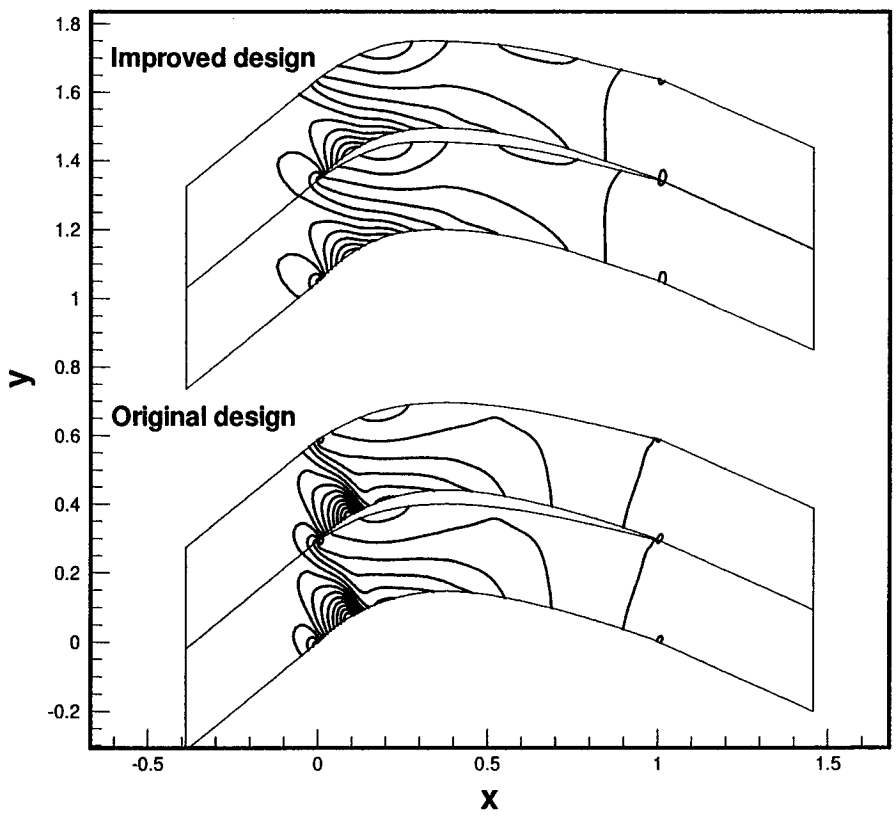


Figure 4.16: Isentropic Mach contours.

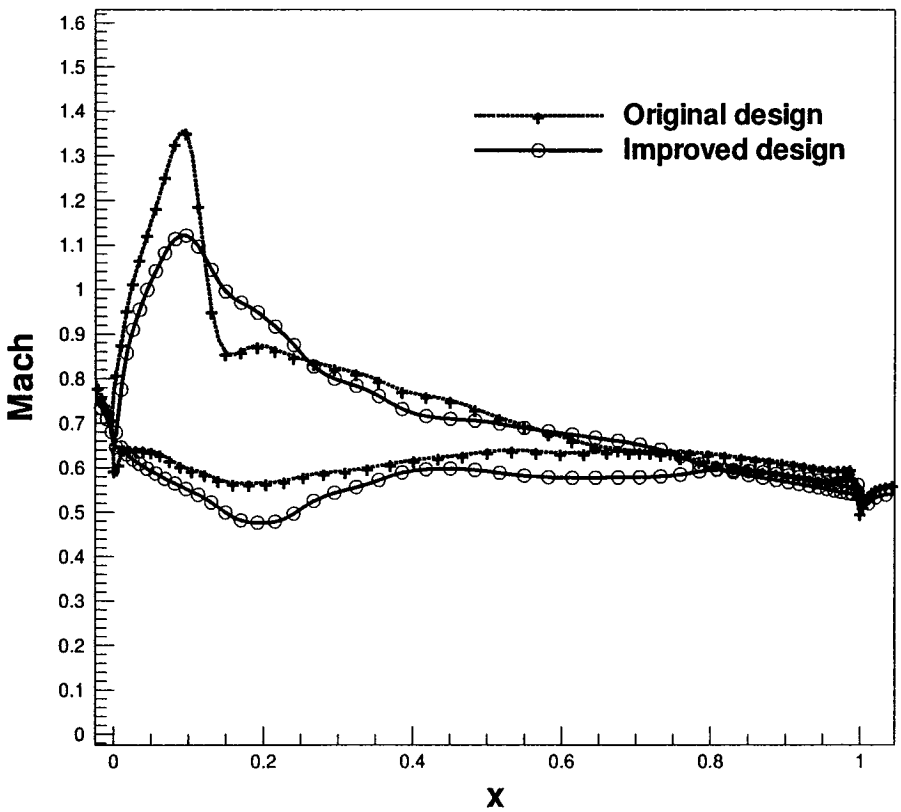


Figure 4.17: Isentropic Mach distribution.

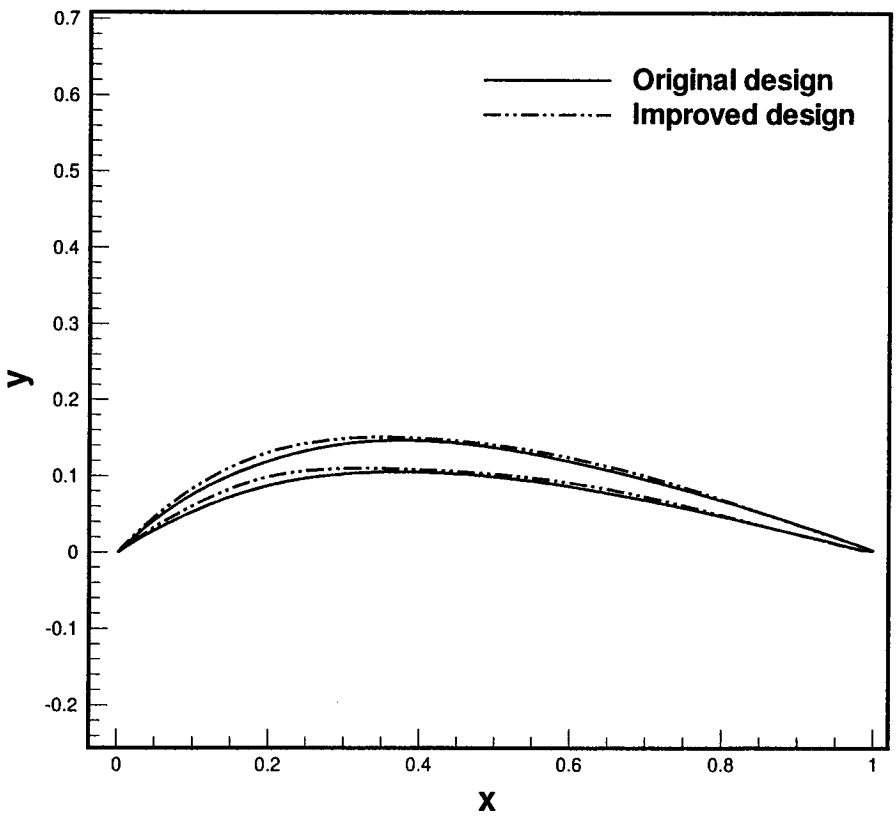


Figure 4.18: Onera compressor cascade original and redesigned profiles.

4.8. Multi-point Design of NACA Transonic Compressor Cascade

The near-tip section of a NACA transonic compressor rotor blade [56] is presently used as a test case for multi-point optimization where the target is to improve the rotor blade performance over the design speed line. (A speed line is obtained by varying the backpressure while keeping the rotor angular velocity fixed.) The performance of that particular blade was numerically investigated by Veress and Sánta [57]. The adiabatic efficiency of this cascade varies between 65% and 89% and the maximum pressure ratio is close to two.

The performance map (given by e.g. efficiency or pressure ratio vs. mass flow rate) for the original design was constructed and was used as the base line in the optimization process. The rotor performance is usually measured in terms of the adiabatic efficiency, pressure ratio and total pressure loss. For this test case, the objective function was defined as maximizing the efficiency and minimizing the total pressure loss coefficient at the design point and at three pre-selected off-design points so as to cover the entire speed line. Given the fact that the flow was assumed to be two-dimensional and inviscid, the CFD flow simulation is rather rapid, hence the objective function was computed from the CFD flow simulation using the 2D Euler finite volume method described earlier [58].

The design variables are the backpressure and the geometric parameters describing the blade profile. The latter is described by the blade camber line and a thickness distribution, which are represented by nine NURBS control points and weights, where only the y-coordinate of the control points is allowed to vary. By varying the backpressure in the pre-determined range from choke to stall limit, the rotor would perform at different mass flow rates with varying efficiency and pressure

ratio.

The objective function is a weighted sum of adiabatic efficiency and total pressure loss over the entire operating range, given by the multi-point maximum efficiency and minimum total pressure loss design, Eq. 4.1. It also includes the geometric and aerodynamic constraints as penalty terms.

The weights, C_1 to C_5 in Eq. 4.1, are determined through numerical experimentation and are taken as 1, 0.5, 0.0, 1 and 0.5, respectively so as to maximize the adiabatic efficiency and minimize the total pressure loss over the operating range. This choice of weights achieves the objectives and in addition, results in an increase in total pressure ratio although the latter was not included in the objective function. This fact was also observed by other researchers [7]. The penalty term (PT) in the objective function takes into account the mechanical and geometrical constraints imposed on the optimization process. The spacing to chord ratio is fixed, a minimum bound is set on the thickness distribution, and the exit flow angle is set to the value prevailing in the original design.

The first step in the optimization process is to properly define the design space. Taking the existing original blade profile as a baseline, the lower and upper limits for the above-mentioned design variables are determined such as to cover the target performance range. These limits correspond to an incidence angle varying between $\pm 11^o$, a flow turning between 8^o and 28^o, and an adiabatic efficiency between 65% and 89%.

The goal of the optimization is to achieve maximum efficiency at all points on the design speed line. The optimized blade, whose profile is given in Fig. 4.19, shows that this goal is accomplished rather well. Figures 4.20, 4.21 and 4.22 show the improvement in efficiency and reduction in total pressure loss. The efficiency increases by about 1.7% and the total pressure loss coefficient was reduced by an

average of 15%. At the same blade speed and backpressure, the improved blade was found to have an average of 1.4% higher mass flow rate.

The higher mass flow rate achieved at a reduced total pressure loss suggests that a smaller number of the optimized blades may give an acceptable performance; this will reduce the weight and the manufacturing cost.

In addition the range of the improved blade is wider than that of the original blade. The optimum blade shape results in a weaker passage shock, which was prevailing in the original design and is now weakened in the improved design.

The four-point optimization task was performed at the design point plus three off-design points, where the values of the backpressure are 0.65, 0.70, 0.75, and 0.78. Note that Figs. 4.20 to 4.22 include not only those points used in the optimization but they also include points that were not implied in the optimization process and, as can be seen from these figures, the improved solution shows a continuous variation of blade performance for all points including those that were not included in the optimization function.

The Mach contours and Mach distribution along the original and the optimized blades at a given mass flow rate, are given in Figs. 4.23 and 4.24. This mass flow rate does not correspond to one of the four values of back pressure that were used in the optimization function; in other words, the CFD simulation of the flow over the optimized blade at that mass flow rate is not implied in the optimization process. The Mach contours, given in Fig. 4.23, indicate that the passage shock in the optimized design is spilled out of the passage, so that the pressure side is shock free and the shock on the suction side is weakened, as can be depicted in Fig. 4.24. This can explain the performance improvement of the optimized blade at that mass flow rate.

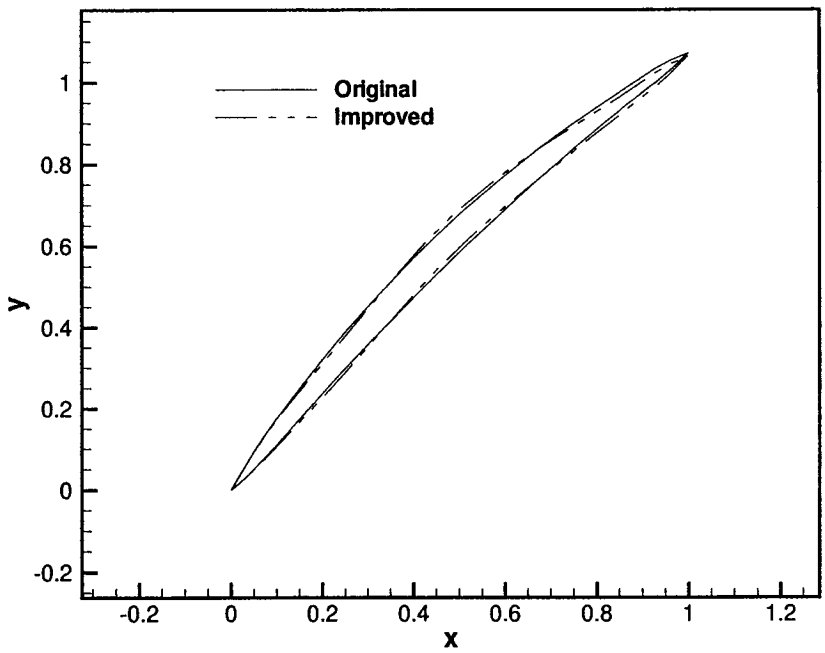


Figure 4.19: The original and optimized NACA transonic compressor rotors.

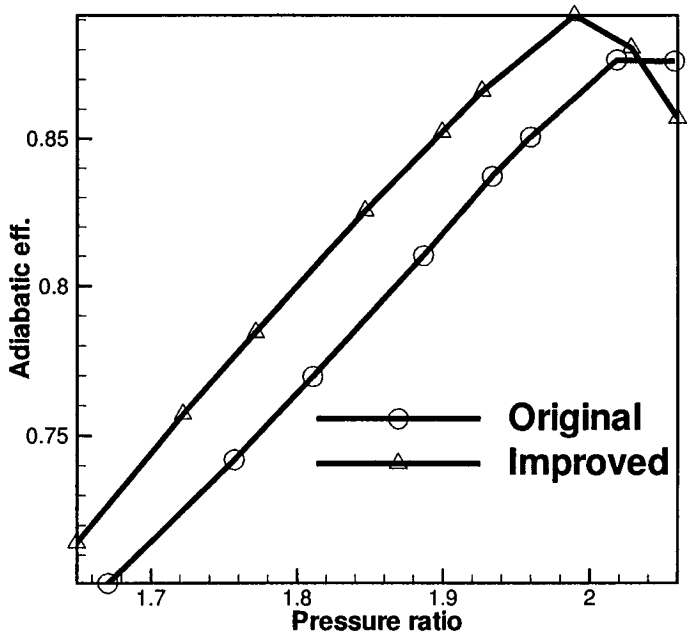


Figure 4.20: Efficiency vs. total pressure ratio for the original and optimized NACA transonic compressor rotors.

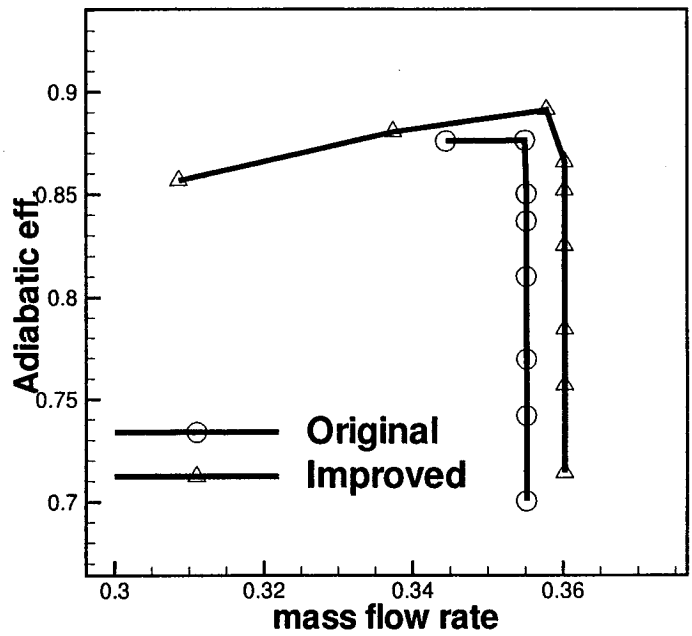


Figure 4.21: Efficiency vs. mass flow rate for the original and optimized NACA transonic compressor rotors.

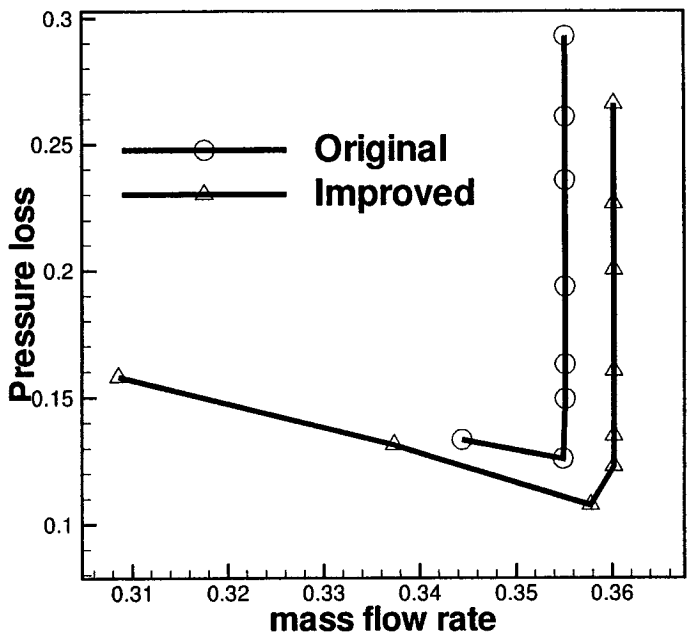


Figure 4.22: Total pressure loss coefficient vs. mass flow rate for the original and optimized NACA transonic compressor rotors.

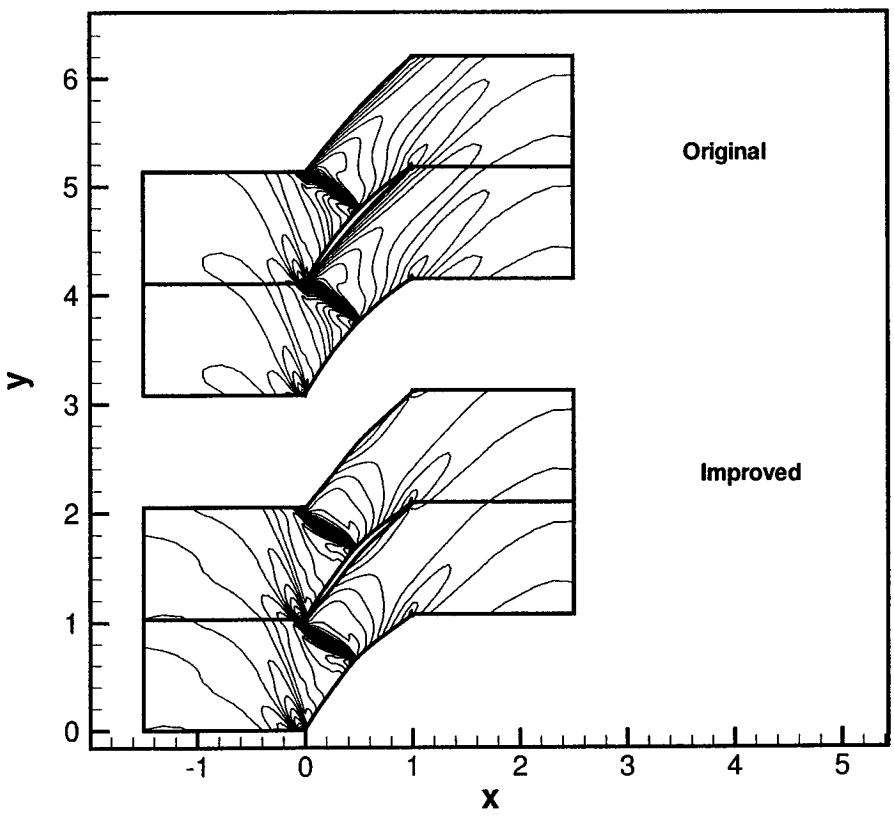


Figure 4.23: Isentropic Mach contours for the original and optimized NACA transonic compressor profile at a given mass flow rate.

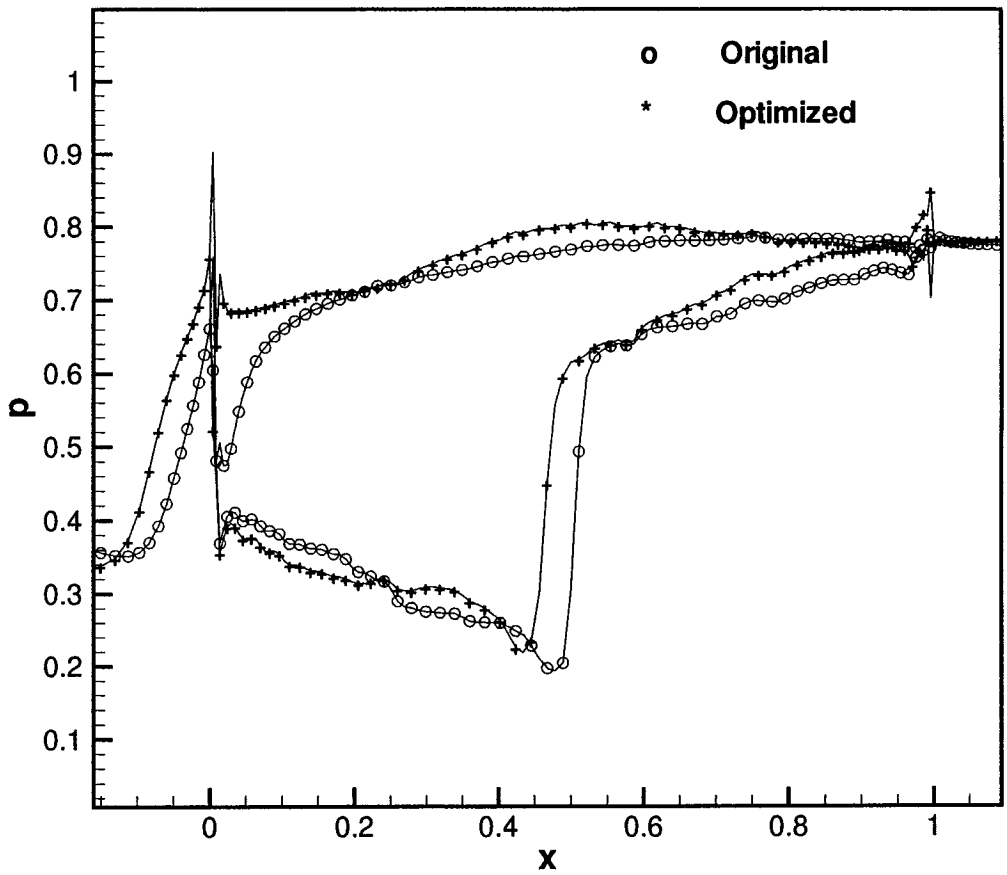


Figure 4.24: Pressure distribution along the original and optimized NACA transonic compressor profile at a given mass flow rate.

4.9. ANN-Based Multi-point Optimization of a NACA 65 Subsonic Compressor

This test case deals with optimizing the performance of a subsonic compressor rotor where the flow is assumed to be turbulent and is governed by the RANS equations;

the Reynolds number is 2.45×10^5. ANN is used so as to reduce the computational effort and hence accelerate the design process, and the optimization objective is to maximize the adiabatic efficiency over the rotor operating range. Four values for the backpressure were taken to cover the operating range.

The optimization approach is similar to the previous case, (i.e. the transonic NACA compressor cascade) except for the use of ANN in this case to obtain an approximate value of the objective function. More explicitly, the NURBS representation, the GA numerical optimization, the CFD boundary condition implementation used in the previous section are also used in this section. The performance is measured in terms of adiabatic efficiency and the objective function is given in Eq. 4.1. The ANN model is trained and tested using a data base of blade profiles for which the flow is simulated using CFD. The accuracy and effectiveness of the developed optimization method is demonstrated (see Appendices C) in the process of redesign of this NACA subsonic compressor cascade.

The base line geometry is given by a NACA 65 airfoil, for which there are extensive low speed cascade experimental data [59], the blade shape is defined in terms of a mean camber line and a thickness distribution. Both the camber line and thickness distribution have been parameterized using NURBS with 11 and 9 control points, respectively. The family of airfoils used in training and testing the ANN was obtained by perturbing the NURBS control points in a preset range in order to build an approximation model using ANN. This approximate model is then used in the GA optimization process to compute the objective function for a given set of design parameters.

ANN was used to construct the response surface using relatively few CFD flow simulations. A family of fifty blade profiles has been generated in the design space and, for each blade profile, the flow was simulated at the design point and at three

off-design points; thus the total number of CFD runs amounts to 200 simulations that cover a reasonably large range of the design space, see Fig. 4.25. The camber and thickness is allowed to vary from 7% to 10% while the back pressure is varied incrementally from the choke to stall limit of the original design.

The CFD flow simulations were carried out in parallel (one CFD simulation per node) on four nodes of an ALTIX 3700 server using Message Passing Interface (MPI), which required about 60 hours of wall-clock time.

The ANN was constructed with one hidden layer containing 41 nodes and was trained using 35 of the 50 candidates; it was then tested with the remaining 15 candidates. The success of ANN depends on its performance on the test cases, as the ANN training does not include these cases. The ANN training took about five hours on a Pentium IV running at 2.2GHz clock speed and the average error was found to be less than $\pm 5\%$, as seen in Fig. 4.27.

The stopping criterion of ANN training is such that it is determined based on the error incurred in ANN testing. As the ANN weights optimization is continued, the error on the training keeps decreasing until it goes down to machine accuracy (which implies overtraining). Unlike the error on the training, the error on testing ANN reaches a certain minimum and then it starts to rise again, when the overtraining degrades its generalization capability and it starts to study details on the data that leads to memorization rather than generalization. The ANN weights optimization is stopped at the point where the error on the testing reaches its minimum, as shown in Fig. 4.26.

The optimization algorithm that employs the GA uses the trained ANN to compute the objective function as well as the constraints. The result obtained from ANN-based numerical optimization is then verified by simulating the flow over the optimum shape using CFD.

This test case is a multi-point maximum efficiency optimization where the design point and three off-design points are included in the optimization function, Eq. 4.1. The weights, $C_1 ... C_5$ in Eq. 4.1 are taken as 1, 0.5, 0, 0 and 0, respectively. The selected choice was found to achieve the objectives and it also resulted in an increase in total pressure ratio; this behavior was also observed by Oyama *et al.*[7].

Figure 4.28 shows the performance map for the candidates used in building the ANN model as well as the improved blade; the latter has shown more than 4% gain in adiabatic efficiency.

The optimized profile has shown a significant change near the trailing edge as shown in Fig. 4.29. It gives a better performance in terms of adiabatic efficiency as well as pressure ratio (although the latter was not part of the optimization function) over the full range of operation. Figures 4.30 and 4.31, which plot the efficiency vs. mass flow rate and vs. pressure ratio, show a 7% improvement in efficiency and about 1% increase in total pressure ratio for the same blade speed and mass flow rate.

The performance of the improved rotor blade is tested at an operating point different from those taken in the optimization, a smooth and continuous trend similar to the first test case is observed at all points, as seen in Figs. 4.32, 4.33, and 4.34

As mentioned earlier, the optimized profile has shown a significant change near the trailing edge as shown in Fig. 4.29 and the maximum thickness of the optimized profile has increased by 3.7% and has slightly moved downstream. The high camber near the trailing edge resulted in a higher flow turning at the cost of an increase in profile loss. Figures 4.35 and 4.36 show a comparison of the Mach contour and pressure distribution of the improved design with the original design at one operating point.

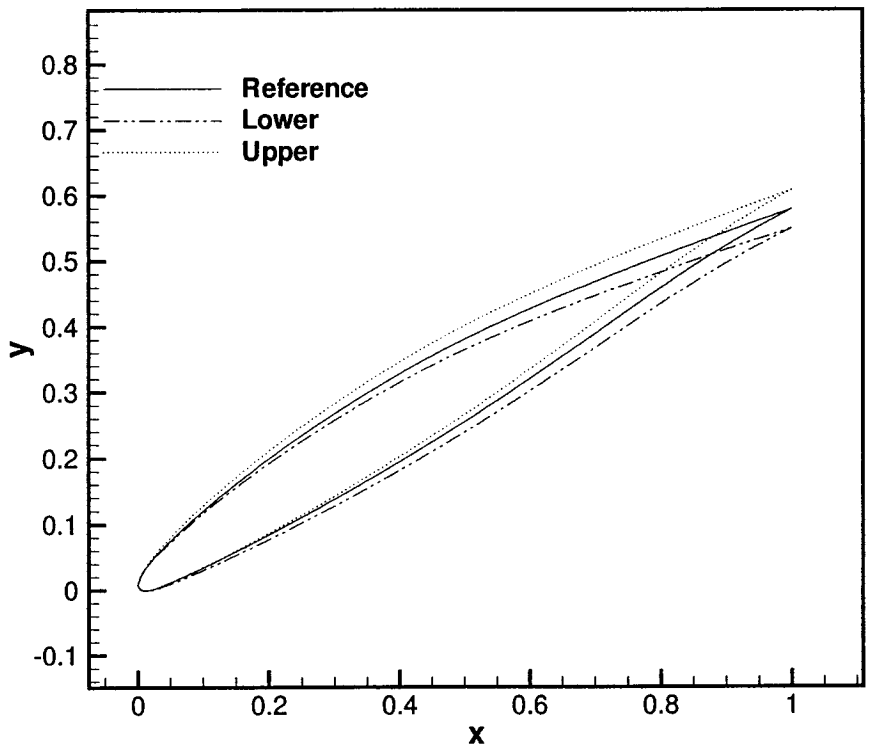


Figure 4.25: The range of geometry explored for the design space.

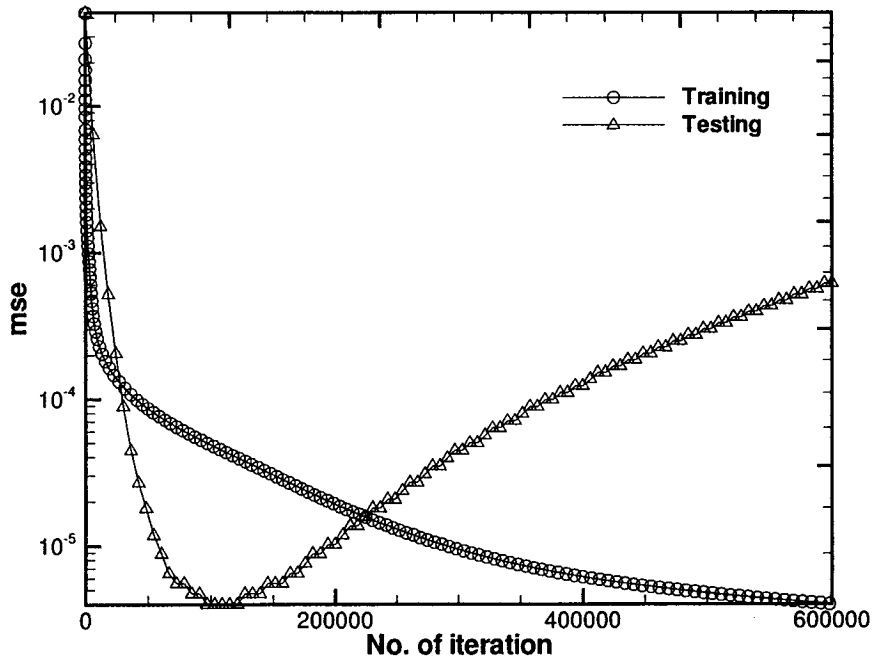


Figure 4.26: The convergence history of ANN training/testing.

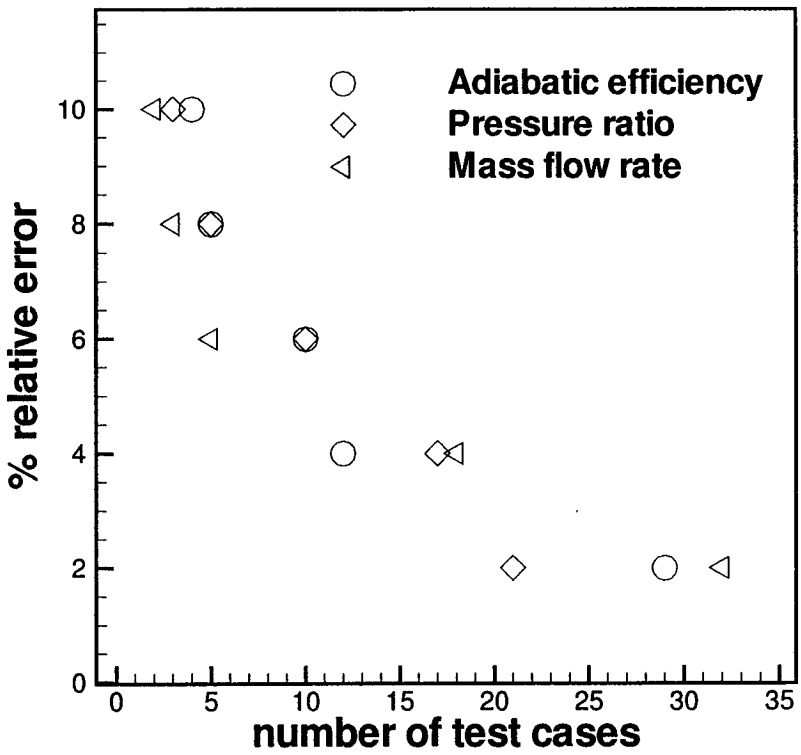


Figure 4.27: The ANN model validation.

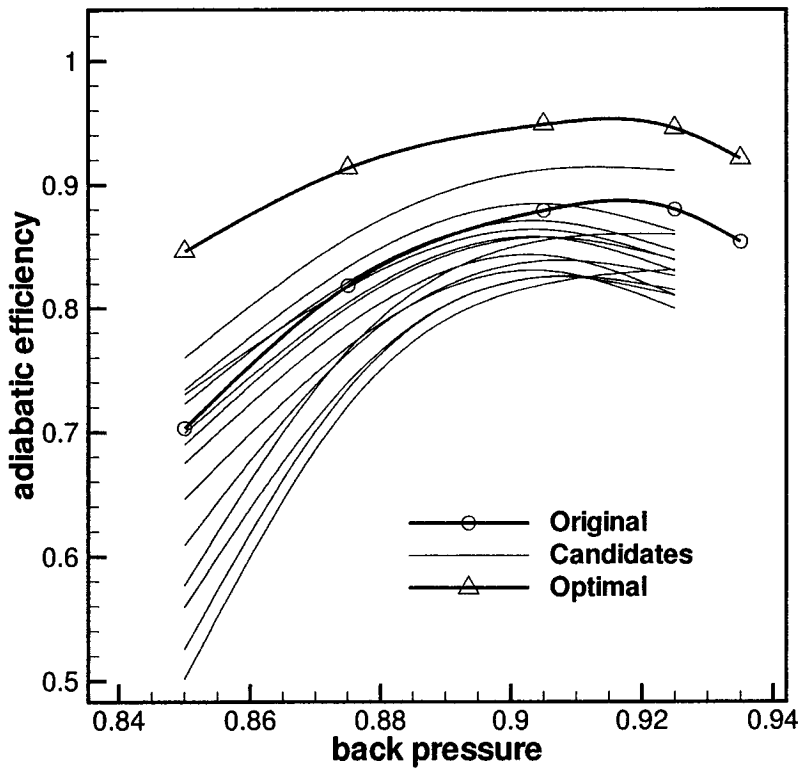


Figure 4.28: Speed lines of blade profiles used in training/testing the ANN.

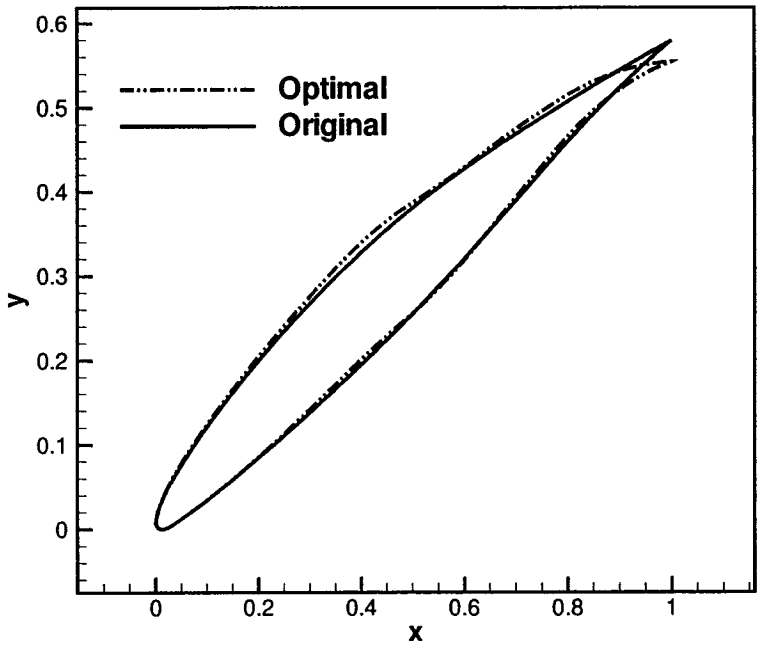


Figure 4.29: The original and optimized blade profiles.

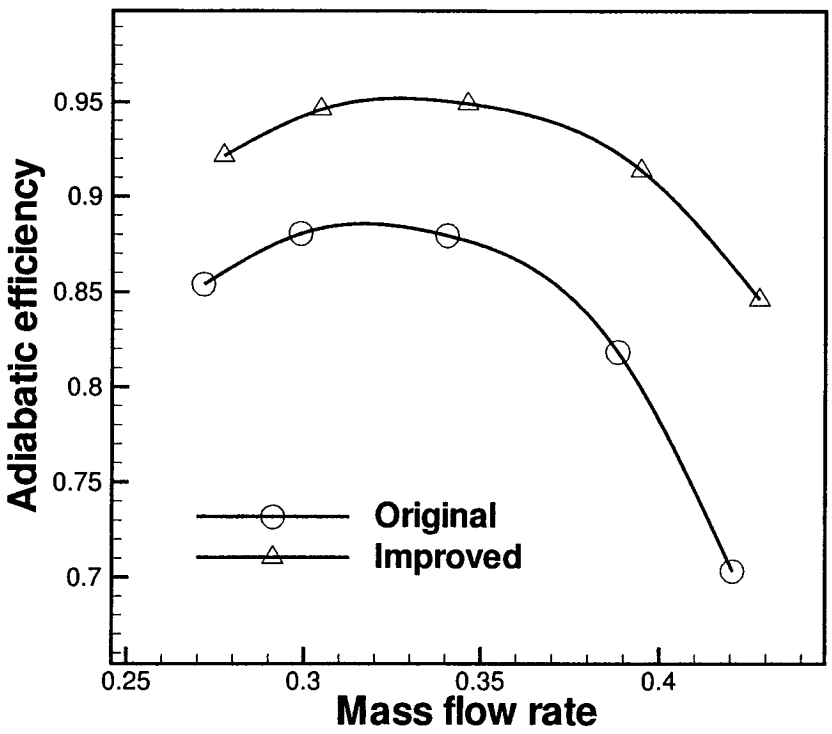


Figure 4.30: Efficiency vs. mass flow rate for the NACA 65 subsonic compressor.

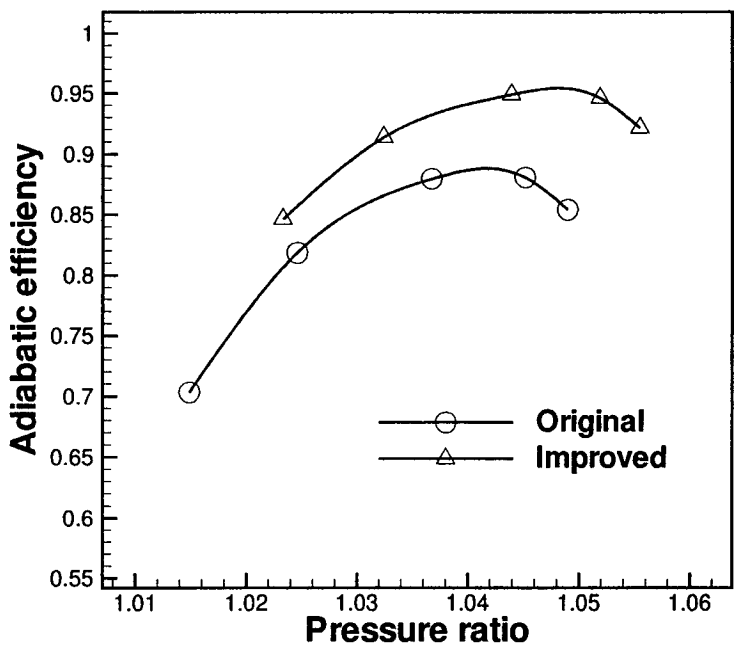


Figure 4.31: Efficiency vs. pressure ratio for the NACA 65 subsonic compressor.

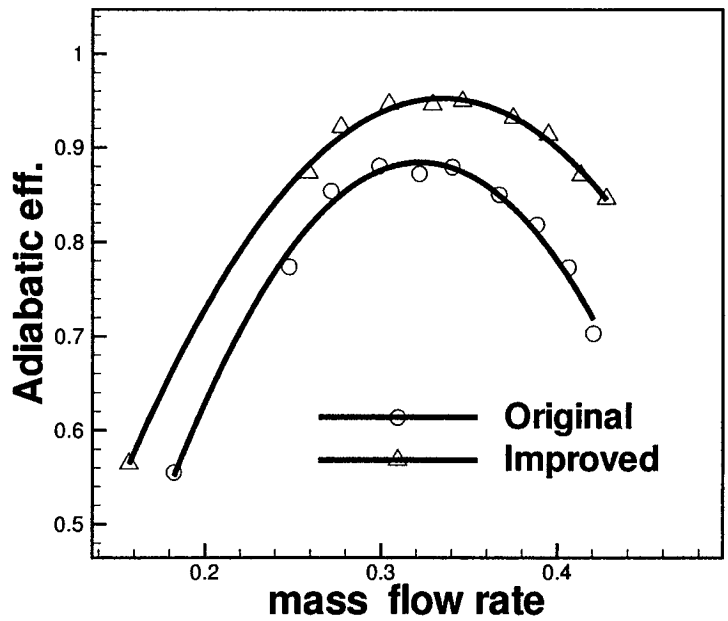


Figure 4.32: Efficiency vs. mass flow rate for the NACA 65 subsonic compressor.

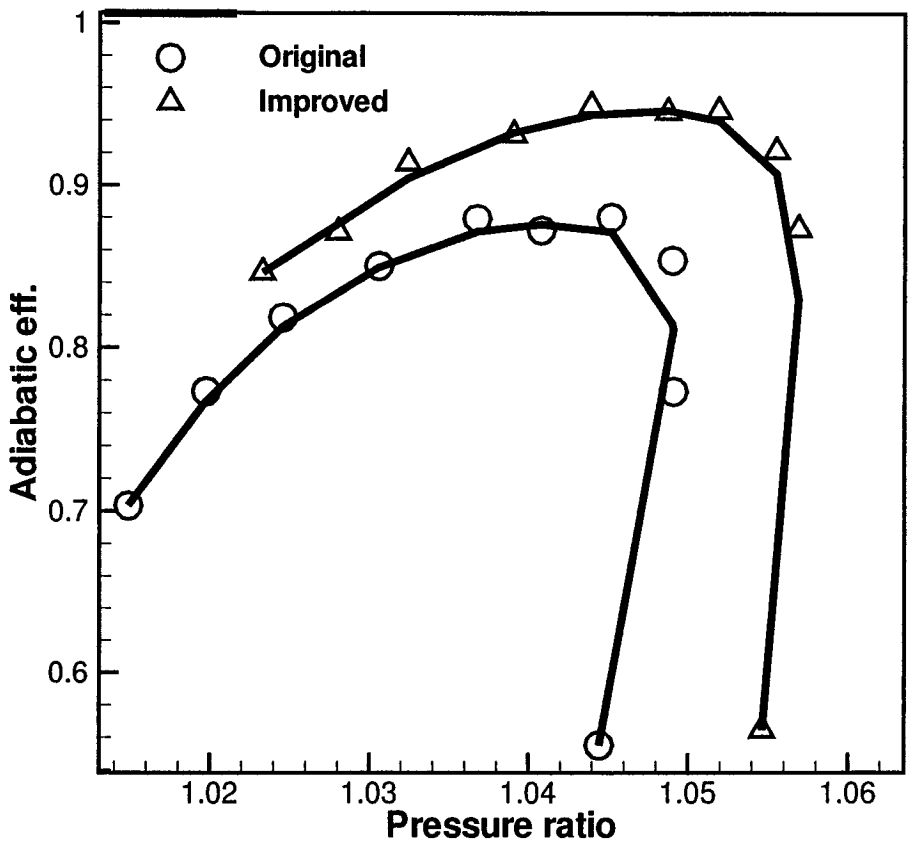


Figure 4.33: Efficiency vs. pressure ratio for the NACA 65 subsonic compressor.

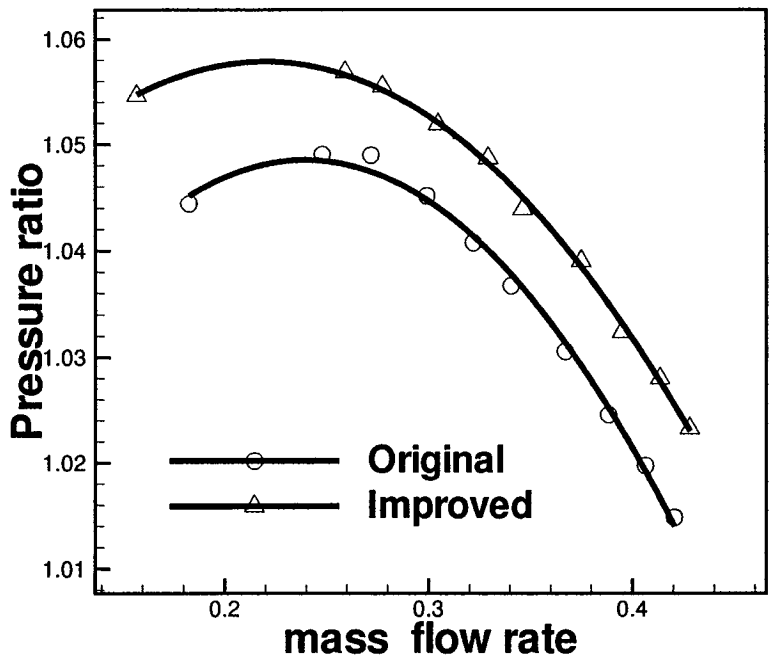


Figure 4.34: Pressure ratio vs. mass flow rate for the NACA 65 subsonic compressor.

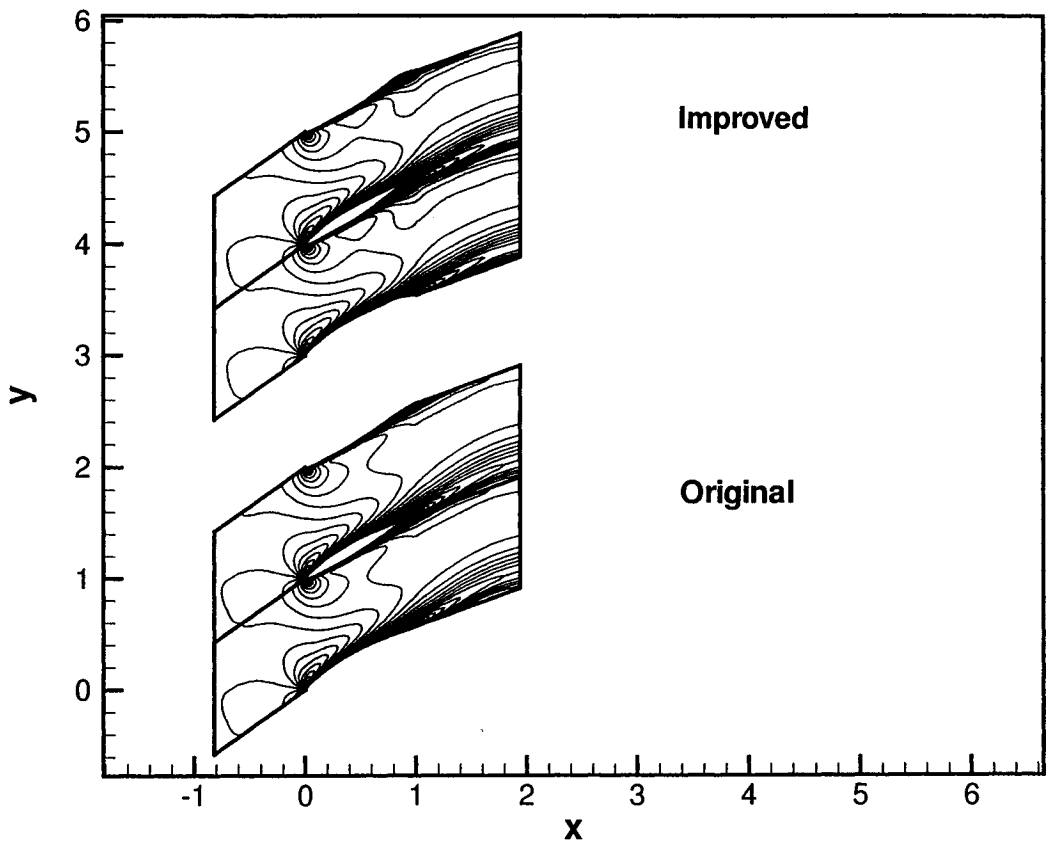


Figure 4.35: Mach contours for the improved and original NACA 65 subsonic compressor profiles at a given mass flow rate.

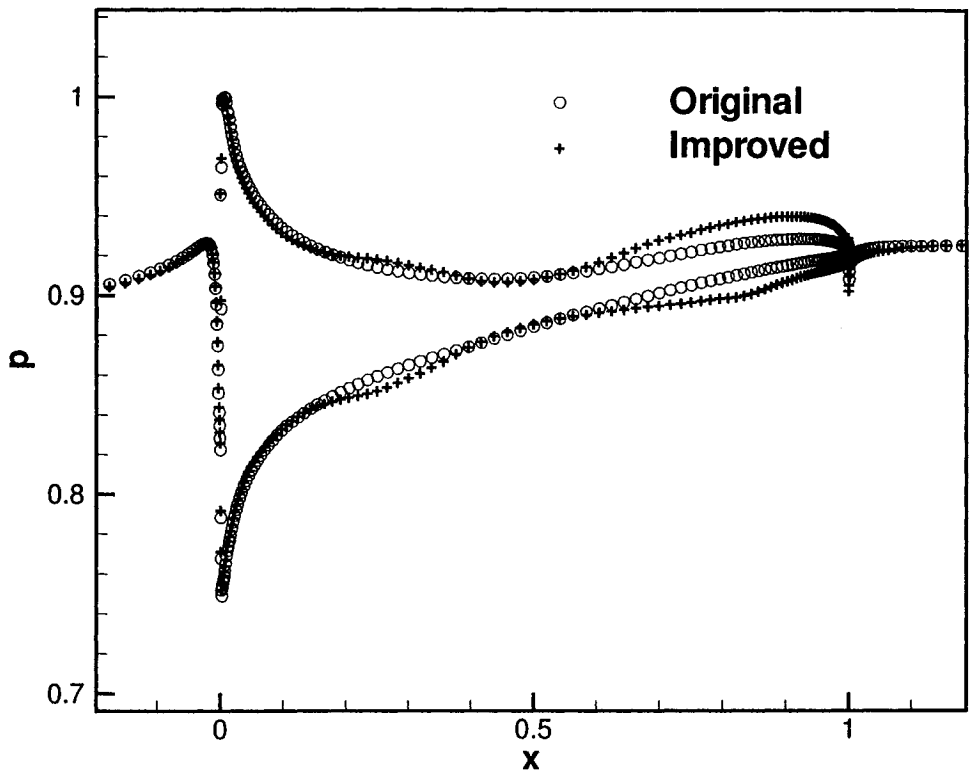


Figure 4.36: Pressure distribution on the blade surface for the NACA 65 subsonic compressor profile at a given mass flow rate.

Chapter 5

CONCLUSION

5.1. Summary

Aerodynamic shape design of compressor and turbine blades is quite challenging given the complexity of the design space and the inherent multidisciplinary nature of such design. An attempt to automate this design process is explored in this work, by developing an automated optimization process that integrates shape parametrization functions, used to approximate the blades profile, with CFD, to compute the optimization objective from the flow simulation, with numerical optimization methods that drive the optimization process. Each component of this automatic optimization process (except for the CFD method) is developed and validated, the goodness of the automated process is then demonstrated in the redesign of two-dimensional flow in turbine and compressor cascades where the flow is transonic, subsonic, viscous or inviscid, and where the optimization objective is a weighted sum of single design point or multi-point, single or multi-objectives; the weighted sum is augmented by the design constraints.

The main objective of this work is to develop and implement methods for

automated aerodynamic shape optimization of turbomachinery blades. The first step in this process is to parameterize the blade profile. At present, Non-Uniform Rational B-Splines (NURBS), which are used by modern Computer Aided Design (CAD) tools such as CATIA, were chosen because of their excellent mathematical properties for shape parametrization in terms of accuracy, generality and flexibility, and in terms of their ability to adequately represent the blade shape with as few design parameters as possible. A separate optimization algorithm that uses SA has thus been integrated in the design cycle to enable the NURBS parametrization of a given blade with a relatively small number of parameters while keeping the quality and smoothness of the original blade profile.

The numerical optimization schemes that were developed, tested, and used in this work are GA and SA which are direct global optimization methods; and SQP which is a gradient-based method that is effective for detecting a local optimum. It was found necessary in the course of this work to develop these schemes to address the different optimization tasks. The GA was chosen for the aerodynamic optimization part, SA was found to work best with the NURBS control point and weight calculation, and a hybrid scheme of GA, SA and SQP was needed to find the weights of the ANN algorithm.

By using a response surface-based optimization objective instead of a CFD-based objective, the computational effort in achieving an improved design can be dramatically reduced. Although this low order approximation is not as accurate as the CFD-based value, the result from this method can be verified with the CFD-based calculation and it provides with a more economical and computationally simpler method of achieving the design task in a short period of time.

An Artificial Neural Network (ANN) is chosen as the low order response surface approximation of the objective function. The construction of the response surface

approximation with ANN is based on a database generated using CFD simulations, which are parallelized using MPI on a cluster of parallel computers.

The objective function formulation is made so general that the designer have an option of choosing single or multi-point optimization and combine different aerodynamic performance measures like adiabatic efficiency, total pressure loss, pressure rise etc... The developed methodology and the formulation of the objective function resulted in a significant and consistent aerodynamic design improvement, and the use of ANN resulted in a ten-fold speed-up of the design process.

5.2. Concluding remarks

A fast, flexible, and robust single as well as multi-point global shape optimization method for the aerodynamic performance of gas turbine blade cascades was developed and was successfully applied to the design of transonic, subsonic, inviscid, or viscous two-dimensional flow in compressor and turbine cascades.

The choice of objective function was found to be crucial for the success of the optimization process; it was constructed so as to achieve a better aerodynamic performance over the full operating range (including the design point) by reshaping the blade profile. The objective function is formulated as a weighted sum of the performance parameters such as adiabatic efficiency, total pressure loss (or entropy rise) and pressure ratio. It is also augmented with the constraints such as the minimum blade thickness, the stall margin, and the mass flow rate.

The objective function is obtained from either a high fidelity or a low fidelity model. The high fidelity model is based on solving the Euler/Navier-Stokes equations for the flow field, which is then used in computing the objective function.

The low fidelity model uses an ANN with a back propagation algorithm to approximate the optimization objectives and constraints. The ANN is developed and implemented for multi-point optimization of viscous turbulent flow in cascades. The ANN is found to reduce the computing time by approximately one order of magnitude. The turn-around computation time was further reduced by parallelizing the optimization process; this parallelization achieved almost 100% efficiency.

The NURBS functions used to parameterize the profile are found to be flexible, smooth and accurate. They also allow for representing a given blade shape with a relatively small number of parameters. Finding these parameters involves an optimization process, where SA was implemented to search for the NURBS parameters that would best represent the initial turbomachinery shape.

In the course of this work, it was found necessary to develop different numerical optimization methods to address different optimization tasks within the global optimization process. A real-coded Genetic Algorithm (GA) is used in the aerodynamic optimization. A Simulated Annealing (SA) scheme is used to find the initial NURBS representation of the blade profile. A Sequential Quadratic Programming (SQP) is used in conjunction with GA and SA to determine the weights of the ANN.

The test cases that are presented show that the developed optimization process is general and flexible, and it can handle various aerodynamic design objectives and constraints. They also showed that the present methodology is able to reach the optimization objective in increasing the adiabatic efficiency and/or increasing the pressure ratio and/or decreasing the total pressure loss in a sustained manner over the full operating range.

The ANN approximation of the aerodynamic objective function can perform well for a rather modest size of the database, which has a significant impact on the accuracy and effectiveness of the resulting ANN model. The database was chosen to

adequately cover the design space and was composed of feasible blade shapes. This was found to save on the computation time for generating the CFD simulations, and it also improved the generalization capability of the ANN model by reducing the overtraining or memorization problem.

In an industrial environment, the database already exits, at least partially, therefore the use of ANN can be even more attractive since the aerodynamic optimization effort will be significantly reduced.

5.3. Recommendations for future work

The work that was carried out in this thesis has pointed out some research directions such as further improvement and refinement of the optimization process, the geometric parametrization, the ANN approximation, and their coupling are needed.

- The geometric parametrization is crucial for the success of the design optimization and generation of new blade profiles. Flexibility of the parametrization and smoothness of the blade profile eases the optimization task. As the order of the NURBS curve increases, the blade becomes smoother, however the number of NURBS parameters increases. The former eases the optimization task while the latter tends to rise the optimization cost. Therefore a high order NURBS using curvature information as constraints and a small number of parameters needs to be developed.

- The numerical optimization in this work is global optimization scheme in the hope of finding a near global optimum. There are many factors that determine the success of the optimization process. The formulation of the objective function and constraints, the selection of design variables and their range are

some of the factors that affect the optimization results. A sensitivity analysis is needed to study the effect of the the design variables as well as the different formulations of the objectives and constraints on the optimization results.

- The results of the optimization are as good as the analysis tool used therein. Hence a careful consideration of the fluid flow simulation tool is important in generating physically correct designs. Issues such as discretization methods, turbulence models, convergence, boundary conditions, mesh quality should be carefully considered.

- A different and improved method for estimating the aerodynamic performance is needed. The total pressure loss, the adiabatic efficiency, and the pressure ratio are performance measures that have different requirements. Some of them even have contradicting goals, e.g. minimizing the total pressure loss in a rotor does not guarantee maximizing its adiabatic efficiency, and vice-versa. The same argument holds for the pressure rise. Therefore the problem of aerodynamic optimization would fall into multi-objective optimization even for a single point design. Other ways of measuring the aerodynamic performance that would account for all the necessary aerodynamic performance measures would simplify the optimization task.

- To account for three-dimensional flow effects on the optimum design, 3D flow simulation and blade parametrization are needed. The methods presented in this work can readily be extended to the optimization of 3D blade rows.

- Multi-Disciplinary Analysis and Optimization (MDO) including various disciplines, such as aerodynamics, thermodynamics, structural dynamics, aeroelasticity and manufacturing affect the design space.

- ANN has a lot of unexplored potential that needs further work, e.g. data selection for the training and testing, ANN architecture in terms of number of hidden layer(s) and nodes in each layer, the optimization method for the training algorithm, the type of transfer function. Moreover, work is needed to optimize the size of the database needed for the ANN training and testing so as to generate an accurate model with relatively little effort. The scaling rule for the number of neurons, the size of the database, for a given set of input/output variables is an area of active research in the ANN research community. Some rules of thumb exist but still need more work.

Bibliography

[1] Boyer, K. M. (2001) *An Improved Streamline Curvature Approach for Off-Design Analysis of Transonic Compression Systems*. Ph.D. thesis, Virginia Polytechnic Institute and State University, Mechanical Engineering.

[2] Guyton, A. C. (1997) *Basic Neuroscience*. W.B. Saunders Co.

[3] Ahmadi, M. and Ghaly, W. (1998) Aerodynamic inverse design of turbomachinery cascades using a finite volume method of unstructured meshes. *Inverse Problems in Engineering*, **6**, 281–298.

[4] Trépanier, J.-Y., Lépine, J., and Pépin, F. (2000) An optimized geometric representation for wing profiles using nurbs. *CASI*, **46**, 12–19.

[5] Dennis, B. H., Dulikravich, G. S., and Han, Z.-X. (1999) Constrained shape optimization of airfoil cascades using a navier-stokes solver and a genetic/sqp algorthm. *ASME paper 99-GT-441*.

[6] Wang, X. and Damodaran, M. (2000) Aerodynamic shape optimization using computational fluid dynamics and parallel simulated annealing algorithms. *AIAA paper 2000-4847*.

[7] Oyama, A., Liou, M.-S., and Obayashi, S. (2002) Transonic axial-flow blade

shape optimization using evolutionary algorithm and three dimensional navier-stokes solver. *AIAA paper 2002-5642*.

[8] Mengistu, T. T. and Ghaly, W. (2003) On the use of global optimization methods in the aerodynamic optimization of transonic cascades. *11th annual CFD Society of Canada*, **1**, 238–247.

[9] Dennis, B. H., Egorov, I. N., Han, Z.-X., Dulikravich, G. S., and Poloni, C. (2000) Multi-objective optimization of turbomachinery cascades for minimum loss, maximum loading, and maximum gap-to-chord ratio. *AIAA paper 2000-4876*.

[10] Burguburu, S. and le Pape, A. (2003) Improved aerodynamic of turbomachinery bladings by numerical optimization. *Aerospace Science and Technology*, **7**, 277–287.

[11] Samareh, J. A. (1999) A survey of shape parametrization techniques. *CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics, NASA Langley Research Center, Hampton, VA 23681.*.

[12] Köller, U., Mönig, R., Küsters, B., and Schreiber, H.-A. (2000) Development of advanced compressor airfoils for heavy-duty gas turbines-part i: Design and optimization. *Journal of Turbomachinery*, **122**, 397–404.

[13] Pierret, S., Demeulenaere, A., Gouverneur, B., Hirsch, C., and den Braembussche, R. V. (2000) Designing turbomachinery blades with the function approximation concept and the navier-stokes equation. *AIAA paper 2000-4879, Propulsion and Energetics Panel.*

[14] yang Lai, Y. and Yuan, X. (2002) Blade design with three-dimensional viscous analysis and hybrid optimization approach. *AIAA paper 2002-5658*.

[15] Yamaguchi, Y. and Arima, T. (2000) Multi-objective optimization for the transonic compressor stator blade. *AIAA paper 2000-4909*.

[16] Rai, M. M. and Madavan, N. K. (1998) Aerodynamic design using neural networks. *AIAA paper 1998-4928*.

[17] Xing, X. and Damodaran, M. (2004) Design of three-dimensional nozzle shape using nurbs, cfd, and hybrid optimization strategies. *AIAA paper 2004-4368*.

[18] Shyy, W., Papila, N., Vaidyanathan, R., and Tucker, K. (2001) Global design optimization for aerodynamics and rocket propulsion components. *Progress in Aerospace Sciences*, **37**, 59–118.

[19] Gen, M. and Cheng, R. (1997) *Genetic Algorithm and Engineering Design*. John Wiley & Sons.

[20] Trigg, M. A., Tubby, G. R., and Sheard, A. G. (1997) Automatic genetic optimization to 2d blade profile design for steam turbines. *ASME paper 97-GT-382*.

[21] Nagata, Y. and Chu, K. H. (2003) Optimization of fermentation medium using neural networks and genetic algorithm. *Biotechnolgical letters*, **25**, 1837–1842.

[22] Zhu, Z. W. and Chan, Y. Y. (1998) An engineering study of genetic algorithms oriented to geometric design applications. *AIAA paper 1998-4722*.

[23] Duvigneau, R. and Visonneau, M. (2002) Hybrid genetic algorithms and neural networks for fast cfd-based design. *AIAA paper 2002-5465*.

[24] Bock, H. G., Egartner, W., Kappis, W., and Schulz, V. (2002) Practical shape optimization for turbine and compressor blades using prsqp method. *Optimization and Engineering*, **3**, 395–414.

[25] Goldberg, D. E. (1985) *Genetic Algorithms in Search, Optimization and Machine learning*. Addison-Wesley.

[26] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953) Equation of state calculations by fast computing machines. *Chem. Phys.*, **21**, 1087–1092.

[27] Kirkpatrick, S., Gelatt, J., D., C., and Vecchi, M. P. (1983) Optimization by simulated annealing. *Science*, **220**, 671–680.

[28] Corana, A., Marchesi, M., Martini, C., , and Ridella, S. (1987) Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. *ACM Transaction on Mathematical Software*, **13**, 262–280.

[29] Daneshkhah, K. (2004) Reynolds-averaged navier stokes solution method. *Private communication, Mechanical and Industrial Engineering Department, Concordia University*.

[30] Goel, T., Vaidyanathan, R., Haftka, R. T., and Shyy, W. (2004) Response surface approximation of pareto optimal fron in multi-objective optimization. *AIAA paper 2004-4501*.

[31] Dornberge, R., Buche, D., and Stoll, P. (11-14 Sep 2000) Multidisciplinary optimization in turbomachinery design. *European Congress on Computational Methods in Applied Sciences and Engineering, Barcellona, ECCOMAS 2000*.

[32] Lian, Y. and Liou, M. (2004) Multiobjective optimization using coupled response surface model and evolutionary algorithm. *AIAA paper 2004-4323*.

[33] Markine, V. L. and Toropov, V. V. (2002) Use of high- and low-fidelity models in approximations for design optimization. *AIAA paper 2002-5651*.

[34] Pierret, S. and Braembussche, R. A. V. D. (1998) Turbomachinery blade design using a navier-stokes solver and artificial neural networks. *ASME paper 98-GT-4*.

[35] Papila, N., Shyy, W., Fitz-Coy, N., and Haftka, R. T. (1999) Assesement of neural net and polynomial based techinques for aerodynamic applications. *AIAA paper 1999-3167*.

[36] Vaidyanathan, R., Papila, N., Shyy, W., Tucker, P. K., Griffin, L. W., Haftka, R. T., and Fitz-coy, N. (2000) Neural network and response surface methodology for rocket engine component optimization. *AIAA paper 2000-4880*.

[37] Mengistu, T. and Ghaly, W. (2004) Aerodynamic design of gas turbine cascades using global optimizers and artificial neural networks. *ICCFD3*.

[38] Rai, M. M. and Madavan, N. K. (2000) Application of artificial neural networks to the design of turbomachinery airfoils. *AIAA paper 2000-0169*.

[39] Abhijit, S. P. and Robert, B. M. (1996) *Pattern Recognition with Neural Networks in C++*. CRC press.

[40] Hassoun, M. H. (1995) *Fundamentals of Artificial Neural Networks*. MIT Press.

[41] Haykin, S. (1999) *Neural Networks, a comprehensive foundation*. Prentice Hall, 2nd edn.

[42] Cybenko, G. (1989) Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems*, **2**, 303–314.

[43] Schalkoff, R. J. (1997) *Artificial Neural Networks*. McGraw-Hill.

[44] Kröse, B. and der Smagt, P. V. (1996) *An Introduction to Neural Networks*. University of Amsterdam, 8th edn.

[45] Vanderplaats, G. (1984) *Numerical Optimization Techniques for Engineering Design*. McGraw-Hill.

[46] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992) *Numerical Recipes in Fortran, The art of scientific computing*. Cambridge University Press, 2nd edn.

[47] Mengistu, T. and Ghaly, W. (2004) Single and multipoint shape optimization of gas turbine blade cascades. *AIAA paper 2004-4446*.

[48] Farin, G. (1993) *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, 2nd edn.

[49] Talya, S. S., Chattopadhyay, A., and Rajadas, J. N. (2000) Multidisciplinary analysis and design optimization procedure for cooled gas turbine blades. *AIAA paper 2000-4877*.

[50] Pierret, S. and Braembussche, R. V. D. (1999) Three-dimensional turbine blade design using a navier-stokes solver and artifical neural network. *ASME paper GT C557/154/99*.

[51] Goel, S., Cofer, J. I., and Singh, H. (1996) Turbine airfoil design optimization. *Acta. Math.*, **96-GT-158**.

[52] Piegl, L. and Tiller, W. (1995) *The NURBS Book*. Springer.

[53] Jameson, A., Schmidt, W., and Turkel, E. (1981) Numerical solution of the euler equations by finite volume methods using runge-kutta time-stepping schemes. *AIAA paper 1981-1259*.

[54] Fottner, L. (1990) Test cases for computation of internal flows in aero engine components. *AGARD-AR-275*, propulsion and Energetics Panel.

[55] Dang, T. (1995) Inverse methods for turbomachine blades using shock-capturing techniques. *AIAA paper 1995-2465*.

[56] George W. Lewis, J. (1952) *Experimental Investigation of Axial-Flow Compressor Inlet Stage Operating at Transonic Relative Inlet Mach numbers*. NACA, nACA Research Memorandum, NACA RM E52C27.

[57] Árpád Veress and Sánta, I. (2002) A 2d mathematical model on transonic axial compressor rotor flow. *Periodica Polytechnica Ser Transp. Eng.*, **30**, 53–67.

[58] Ahamdi, M. and Ghaly, W. (1998) A finite volume method for inviscid transonic cascade flow with solution adaptation on unstructured meshes. *CASI*, **44**, 175–181.

[59] Emery, J. C., Herrig, L. J., Erwin, J. R., and Felix, A. R. (1958) *Systematic Two-Dimensional Tests of NACA 65-series Compressor Blades at Low Speeds*. NACA, NACA Report 1368.

[60] Wilson, D. G. (1991) *The Design of High-Efficiency Turbomachinery and Gas Turbines*. MIT Press, 5th edn.

[61] Baldwin, B. and Lomax, H. (1978) Thin layer approximation and algebraic model for separated turbulent flows. *AIAA paper 78-257*.

[62] Holmes, D. and Connel, S. (1989) Solution of 2-d navier-stokes equations on unstructured adaptive meshes. *AIAA paper 81-1259*.

[63] Giles, M. (1990) Nonreflecting boundary conditions for euler equation calculations. *AIAA Journal*, **28(12)**, 2050–2058.

[64] Deb. K, A., Anand and Joshi, D. (2002) *A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization*. KanGAL, kanGAL Report No. 2002003.

[65] Fogel, D. B. and Beyer, H.-G. (1995) A note on the empirical evaluation of intermediate recombination. *Evolutionary Computation Journal*, pp. 491–495.

[66] McKay, M., Bechman, R., and Conover, W. (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometics*, **21**, 239–245.

Appendix A

BLADE NOMENCLATURE

Consider the schematic figure for a typical blade cascade shown in Fig. A.1, This figure is taken from the David Gordon [60].

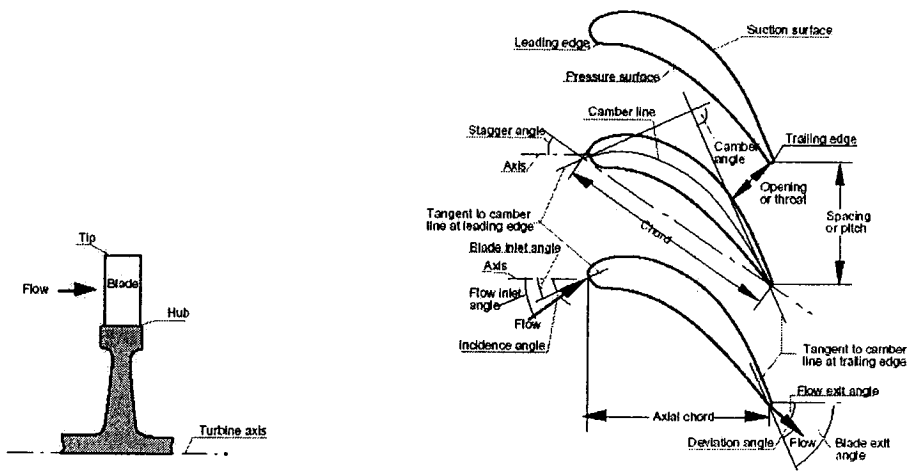


Figure A.1: Blade definition.

The followings terms are defined for the blading shown in the figure.

1. aspect ratio: the ratio of the blade height to the chord.

2. axial chord: the length of the projection of the blade chord onto a line parallel to the the turbine axis

3. axial solidity: the ratio of the axial chord to the spacing.

4. blade height: the radius at the tip minus the radius at the hub.

5. blade inlet angle: the angle between the tangent to the camber line at the leading edge and the turbine axial direction.

6. blade exit angle: the angle between the tangent to the camber line at the trailing edge and the turbine axial direction.

7. camber angle: the external angle formed by the intersection of the tangents to the camber line at the leading and trailing edges. It is equal to the sum of the angles formed by the chord line and the camber-line tangents.

8. camber line: the mean line of the blade profile. It extends from the leading edge to the trailing edge, halfway between the pressure surface and the suction surface.

9. chord: the length of the perpendicular projection of the blade profile onto the chord line. It is approximately equal to the linear distance between the leading edge and the trailing edge.

10. chord line: if a two-dimensional blade section were laid convex side up on a flat surface, the chord line is the line between the points where the front and the rear of the blade section would touch the surface.

11. deflection: the total turning angle of the fluid. It is equal to the difference between the flow inlet angle and the flow exit angle.

12. deviation angle: the flow exit angle minus the blade exit angle.

13. flow exit angle: the angle between the fluid flow direction at the blade exit and the machine axial direction.

14. flow inlet angle: the angle between the fluid flow direction at the blade inlet and the machine axial direction.

15. hub: the portion of a turbomachine bounded by the inner surface of the flow annulus.

16. hub-tip ratio: the ratio of the hub radius to the tip radius.

17. incidence angle: the flow inlet angle minus the blade inlet angle.

18. leading edge: the front, or nose, of the blade and the tip.

19. meridional plane: a plane cutting a turbomachine through a diametral line and the (longitudinal) axis.

20. pitch: the distance in the direction of rotation between corresponding points on adjacent blades.

21. solidity: the ratio of the chord to the spacing.

22. spacing: same as pitch.

23. stagger angle: the angle between the chord line and the turbine axial direction (also known as the setting angle).

24. trailing edge: the rear, or tail, of the blade.

25. mean section: the blade section halfway between the hub.

Appendix B

FLOW FIELD ANALYSIS

B.1. The Governing Equations

The conservation of mass, momentum and energy for a two-dimensional, viscous compressible flow, loosely referred to as Navier-Stokes equations can be written in a Cartesian coordinate system (x, y) as:

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = \frac{\partial \mathbf{R}}{\partial x} + \frac{\partial \mathbf{S}}{\partial y} \tag{B.1}$$

where $\mathbf{w}$ is the vector of flow variables, $\mathbf{f}$ and $\mathbf{g}$ are the convective flux vectors, and $\mathbf{R}$ and $\mathbf{S}$ are the viscous flux vectors in each of the coordinate directions.

$$\mathbf{w} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \ \mathbf{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{bmatrix}, \ \mathbf{g} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho uH \end{bmatrix} \tag{B.2}$$

and

$$\mathbf{R} = \begin{bmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ u\sigma_{xx} + v\sigma_{xy} + q_x \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 0 \\ \sigma_{xy} \\ \sigma_{yy} \\ u\sigma_{xy} + v\sigma_{yy} + q_y \end{bmatrix} \tag{B.3}$$

the stress tensor components and the heat flux vector are given by

$$\begin{aligned} \sigma_{xx} &= 2\mu u_x - \frac{2}{3}\mu(u_y + v_x) \\ \sigma_{yy} &= 2\mu v_y - \frac{2}{3}\mu(u_y + v_x) \\ \sigma_{xy} &= \sigma_{yx} = \mu(u_x + v_y) \\ q_x &= -k\frac{\partial T}{\partial x} \\ q_y &= -k\frac{\partial T}{\partial y} \end{aligned} \tag{B.4}$$

Assuming that the fluid is an ideal gas thermally and calorically and given the definition of total enthalpy H

$$H = E + p/\rho \tag{B.5}$$

Also, for an ideal gas, the equation of state may be written as

$$p = (\gamma - 1)\rho\left[E - \frac{1}{2}(u^2 + v^2)\right] \tag{B.6}$$

where γ is the ratio of specific heats.

The effect of turbulence is taken into account by using the eddy-viscosity hypothesis. That is, the molecular viscosity μ and the molecular thermal conductivity k are replaced with

$$\begin{aligned} \mu &= \mu_l + \mu_t \\ k &= c_p\left[\left(\frac{\mu}{Pr}\right)_l + \left(\frac{\mu}{Pr}\right)_t\right] \end{aligned} \tag{B.7}$$

where c_p is the specific heat at constant pressure, Pr is the Prandtl number and the subscripts l and t refer to laminar and turbulent. The turbulence quantities μ_t and Pr_t are computed using the two-layer algebraic model of Baldwin and Lomax [61].

The Euler equations can be easily obtained by neglecting the viscous terms, namely the right-hand side of Eq. (B.1).

B.2. Space Discretization

The flow governing equations presented in the previous section are discretized in space using a cell-vertex finite volume method, on an unstructured triangular mesh. The computational domain is divided into triangles, fixed in time, and the flow variables are stored at their vertices. For any node, the control volume (surface in 2D) is taken as the union of all triangles with a vertex at that node, i.e. the control volume are overlapping (see Fig.B.1).

The governing equations, Eqs. (B.1), are then integrated over each control volume Ω (surface in 2D) which is bounded by the surface $\partial\Omega$ (curve in 2D), and using Gauss theorem (Green's theorem in 2D) one obtains

$$\frac{d}{dt}\iint_{\Omega} \mathbf{w}\, dx\, dy + \oint_{\partial\Omega} (\mathbf{f}\, dy - \mathbf{g}\, dx) = \oint_{\partial\Omega} (\mathbf{R}\, dy - \mathbf{S}\, dx) \tag{B.8}$$

when the cell-vertex discretization scheme is applied to Eq. B.8, the following set of coupled ordinary differential equations is obtained for each cell or control volume i surrounding node i :

$$\frac{\partial}{\partial t}(\Omega_i \mathbf{w}_i) + \sum_{e=1}^{n}(f_e\, n_{xe} + g_e\, n_{y_e}) = \sum_{e=1}^{n}(R_e\, n_{xe} + S_e\, n_{y_e}) \tag{B.9}$$

where the summation is taken over all edges of cell i, Ω_i is the cell area, f_e and g_e are the components of convective flux vector on edge e, R_e and $S - e$ are the

components of viscous flux vector, and n_x, n_y are the components of the outward normal to edge e.

The convective fluxes, f and g, along a particular edge of a control volume are numerically evaluated as the average of the nodal flux values at the ends of that edge, which assumes a linear variation and is second order accurate. In order to estimate the viscous fluxes, R and S, an auxiliary control volume is formed by connecting the cell centers. The discrete Gauss theorem is applied once on the computational cells to obtain a numerical approximation to the stress tensor and heat flux vector at the cell centers. The divergence of the stress tensor and of the heat flux vector is then obtained directly at the enclosed vertex by a second application of the theorem to the auxiliary control volume which is scaled for consistency, to the computational cell used in convective balance.

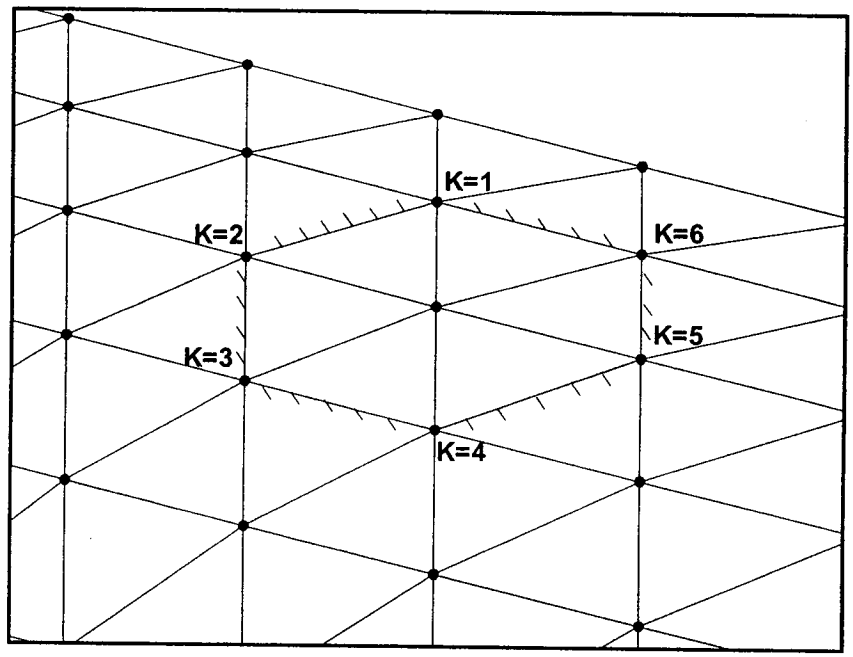


Figure B.1: The cell vertex.

B.3. Artificial Viscosity

In principle, the physical viscous terms of the Navier-Stokes equations are capable of providing the numerical scheme with the dissipative property necessary for stability and capturing discontinuities. However, for high Reynolds-number flows, this can only be achieved by resorting to extremely small mesh spacing throughout the domain. Thus, in practice, it is necessary to introduce artificial dissipative terms to maintain stability in the essentially inviscid portion of the flow field, and

to efficiently capture discontinuities.

In the present work, the nonlinear artificial viscosity formulation advanced by Jameson *et al.* [53], combined with the pseudo-Laplacian discretization, introduced by Holmes and Connell [62] have been used. Details of implementation can be found in [58].

B.4. Integration to Steady State

The discretization of the spatial derivatives transforms equation B.1 into the set of coupled ordinary differential equations

$$\Omega_i \frac{dw_i}{dt} + [Q(w_i) - D(w_i)] = 0 \quad i = 1, 2, 3, ..., n \tag{B.10}$$

where n is the number of mesh nodes. The residual $Q(w)$ represent the discrete approximation to the convective fluxes. $D(w)$ represents the dissipative terms, i.e. the discrete approximation to the viscous fluxes, as well as the artificial dissipation terms. These equations are integrated in pseudo-time using a five-stage hybrid time-stepping scheme given by

$$w^{(0)} = w^n \tag{B.11}$$

$$w^{(1)} = w^{(0)} - \alpha_1 \frac{\Delta t}{\Omega} \left[Q(w^{(0)}) - D_0 \right] \tag{B.12}$$

$$w^{(2)} = w^{(0)} - \alpha_2 \frac{\Delta t}{\Omega} \left[Q(w^{(1)}) - D_1 \right] \tag{B.13}$$

$$w^{(3)} = w^{(0)} - \alpha_4 \frac{\Delta t}{\Omega} \left[Q(w^{(2)}) - D_2 \right] \tag{B.14}$$

$$w^{(4)} = w^{(0)} - \alpha_4 \frac{\Delta t}{\Omega} \left[Q(w^{(3)}) - D_3 \right] \tag{B.15}$$

$$w^{(5)} = w^{(0)} - \alpha_5 \frac{\Delta t}{\Omega} \left[Q(w^{(4)}) - D_4 \right] \tag{B.16}$$

$$w^{n+1} = w^{(5)} \tag{B.17}$$

w^n represents the value of the solution vector at the n^{th} time step and $w^{(q)}$ represents the values at the q^{th} stage within a time step. The dissipative operator $D(w)$ is evaluated only at the first, third, and fifth stages of the scheme.

The coefficients of integration are:

$$\alpha_1 = 1/4 \quad \alpha_2 = 1/6 \quad \alpha_3 = 3/8 \quad \alpha_4 = 1/2 \quad \alpha_5 = 1 \tag{B.18}$$

B.5. Boundary Conditions

For a cascade problem, there are four types boundary condition, namely inflow/outflow boundary conditions, periodic boundary conditions and solid wall boundary conditions, which must be enforced along the boundaries of the computational domain, see Fig. B.2. Each boundary condition types is discussed below.

B.5.1 Inflow/Outflow Boundary Conditions

The computation of inflow and outflow boundary conditions is based upon a linearized characteristics method, described in Giles [63]. The change in the incoming characteristics are determined such as to satisfy specified boundary conditions.

For a subsonic inflow, there are three characteristics propagating into the domain and only one outgoing . The flow angles, stagnation enthalpy and stagnation pressure are specified at the inflow and the static pressure is extrapolated from inside the domain. In some cases the mass flow rate is specified instead of total pressure. There are four incoming waves for a supersonic inflow. The Mach number

is also specified for a supersonic inflow and subsequently the flow condition will be determined from these variables using isentropic relations.

For subsonic outflow, three outgoing waves are calculated from the numerical solution, while the incoming wave is fixed by specifying the static back pressure. There are four outgoing waves for a supersonic outflow, hence all the flow variables are extrapolated from inside the domain.

B.5.2 Periodic Boundary Conditions

For a linear cascade, the flow periodicity implies that all the variables are the same at periodic pairs of nodes. To avoid interpolation and accompanying inaccuracy, the grid generation provides the periodic nodes in pair, which are treated as interior nodes by adding all the contributions at one periodic node to its periodic counterpart.

B.5.3 Solid wall Conditions

For an inviscid flow along an impermeable wall, the flow tangency condition is imposed, which implies that all fluxes through the wall faces vanish except for the pressure contribution to the momentum equation. For viscous flow, the no slip boundary condition is imposed along the impermeable walls.

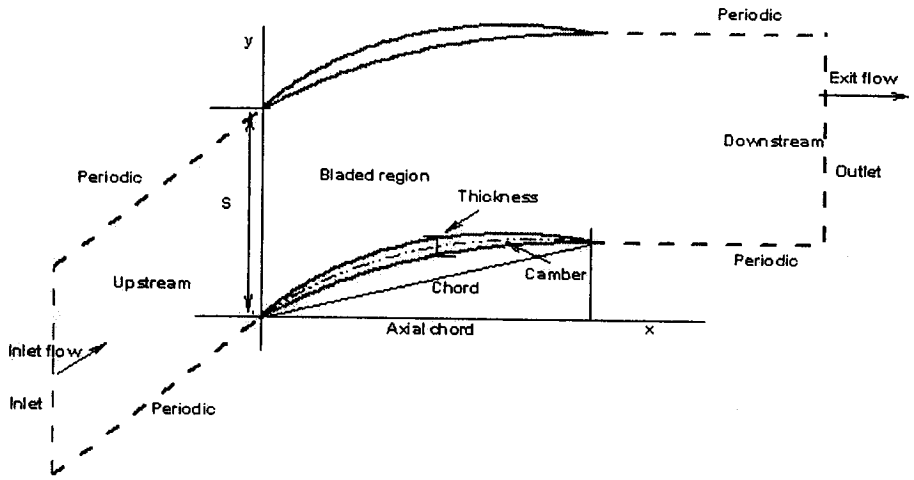


Figure B.2: Cascade notation and boundary conditions.

Appendix C

VALIDATION CASES

C.1. Optimization Scheme Validation

In order to test the SA and GA implementations, some standard optimization problems were chosen from the literature [64]. The problems size ranges from 2 to 10,000 design variables. The tests include the Rosenbrock banana function, the Rastrigin function and others. The test functions vary in difficulty, in number of local minima, and in number of design variables. They have a global extremum that is hidden among many local extrema. Therefore they are appropriate for testing different implementations of global search optimizers such as GA and SA. The search range that was chosen for each function includes several local minima. These problems are given below.

C.1.1 The Rosenbrock Function

The Rosenbrock test function is given by:

$$f(X) = \sum_{i=1}^{n-1} \left(100 \times (x_i^2 - x_{i+1})^2 \times (x_i - 1)^2\right) \tag{C.1}$$

It is a classic optimization problem, also known as the banana function. The global minimum is inside a long, narrow, parabolic shaped flat valley. The convergence to the global optimum is difficult and hence this problem has been repeatedly used to assess the performance of optimization algorithms. The function has a global minimum value of zero at the point $x_i = 1$ for all i, i.e. $f(1) = 0$ is the global optimum. Different ranges of the design variables were defined as ± 2.048, ± 10.048 and ± 60.048. In the following results, the range ± 10.048 was used. Table C.1, which compares the current SA results with those of Ref. [64], shows that the present implementation of SA is more efficient in terms of the number of function evaluations needed to reach the global minimum. Figure C.1 shows the convergence history for the problem of minimizing the Rosenbrock function with forty design variables, $n = 40$. Both GA and SA have shown a large improvement in the optimal solution relative to the initial one, but SA outperforms GA in getting very close to the global solution. SA was able to find the global optimum with an absolute error of 10^{-20} with 2,424,000 function evaluations. In all cases, the SA was able to find the global optimum point very accurately without any significant change in the computation time whereas the GA algorithm had difficulty in reducing the absolute error below 20, except for the case where n=2 (i.e., two variables) the GA did find the global minimum very accurately.

C.1.2 The Rastrigin Function

The Rastrigin function is given by the following expression

$$f(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) \right) \tag{C.2}$$

It is a widely used multivariable multimodal (i.e., with several local extrema) test function. The function global minimum value is 0 and it occurs at $x_i = 0$, i.e. the global minimum is $f(0) = 0$. In the optimization search, all the x_i's are defined in the range ± 5.12 and ± 10.12 and even larger.

Figure C.2 shows a comparison of the SA and GA algorithms applied to the Rastrigin function, with forty design variables, $n = 40$. Both SA and GA brought a large improvement in the optimal value. SA improved the optimal value by 99.6% relative to the initial value while GA improved it by 100%. The absolute error is less than 10^{-14} for GA and 4.97 for SA.

The range of the design variables was successively increased from ± 5.12 to ± 10.12, which increased the number of local optima from 11 to 21 as each integer corresponds to a local minimum of the Rastrigin function, see Fig. C.3. In all cases the GA was able to find the global minimum accurate to 10^{-14} without any significant change in the computation time. The values reported in Table C.2 correspond to an x_i range of ± 100.12.

Fogel and Beyer [65] reported results for the Rastrigin function with 30-variables; the best function value that they reported was larger than 10 after 200,000 function evaluations. In the present work, starting from the same initial range of design variables, a function value of 10^{-6} was achieved in 12,400 function evaluations, and a value less than 10^{-14} was achieved with 13,710 function evaluations. Deb *et al.* [64] reported results for the same function but with 20-variables and starting the optimization from an initial solution away from the global optimum, they were only able to reduce the function to a value between 10 and 20.

C.1.3 Concluding Remarks

Further tests were carried out with GA and SA and, in almost all cases tested, it was found out that SA and/or GA were able to track the global optimum in a reasonable computational effort however the algorithm parameters have to be well tuned.

The solution accuracy and convergence rate for both SA and GA algorithm depends on the respective parameters. For SA, the solution is most sensitive to the cooling schedule parameters i.e., the temperature (T) and its reduction factor (rt). A temperature of 5×10^6 and 10^6 are used for the Rosenbrock and Rastrigin test functions respectively, and a temperature reduction factor of 0.85 is used for both cases.

The value of the parameters used in testing the GA on the Rosenbrock and Rastrigin functions are as follows. The crossover probability is 0.65 and 0.85, the mutation probability is 0.01 and 0.05 and the elitism is 2, respectively.

The test functions that vary in difficulty, in number of local minima, and in number of design variables X, have been tested with the GA and SA routines developed by the authors. The GA algorithm had difficulty with the Rosenbrock function, which has a long flat valley near the global minimum; while the SA algorithm had difficulty with the Rastrigin function, which has numerous local minima. This suggests that any optimizer would not necessarily work for all cases.

C.2. ANN-based Optimization Scheme Validation

The purpose of this task is to validate the ANN-based optimization scheme. A test case of Rosenbrock's function has been selected and modeled by ANN back propagation with a 2-21-1 architecture of the network which contains 2 inputs nodes and 21 hidden nodes and 1 output node. The approximation was done using the

representative sample points selected by Latin-Hypercube experiments [66] over the specified domain for 100 sample data. A surface plot is generated and compared with the actual plot. The average relative error is about 23% while the maximum relative error is 58%. The 3D surface shown in Figs. C.4 and C.5 indicate that the ANN appears to better emulate the the underlying system. This result was obtained using only 100 sets of sample data, increasing the number of sample data set would improve the accuracy of the prediction without any significant additional load of computations. Table C.3 shows a comparison of the predictions by ANN with the exact value. The average relative error of each predictions show the good prediction power of ANN. In order to validate the use of ANN-based approximation for optimization process, the same test case has been optimized with the prediction of the objective function by the ANN-approximated model. As one can see from the Fig. C.6, the optimization process based on a low fidelity approximation gives a result with same trend as that of the exact high fidelity model. Thus it can be used for optimization process in order to simplify the computation task from the high fidelity model.

Note that this test case is a difficult function used repeatedly to assess the performance of optimization algorithms. As it can be seen from the Table C.3, the design space covers a large range and has a long, narrow, parabolic shaped flat valley. The convergence to the global optimum is difficult, hence the result obtained using only 100 data set for building the approximation by ANN is satisfactory.

Table C.2: The computation result of SA and GA for the Rastrigin function

	SA		*GA*	
Cases(no. of DV)	Optimal value	no. of functions	Optimal value	no. of functions
20	2.98	5,601	10^{-14}	7050
30	2.98	8,401	10^{-14}	13710
40	4.97	16,801	10^{-14}	14310

Table C.1: The SA result for the Rosenbrock function

 # OF FUNCTION EVALUATIONS

(Absolute error $< 10^{-20}$*)*

DV	Present work	Literature [64]
20	1,174,000	1,396,496
30	1,764,000	3,719,887
40	2,424,000	–

Table C.3: Predicted value vs. actual values at the 21 test points

Cases	Actual Value	Target value	% Error
1	2732.360	2768.150	1.310
2	1768.509	1826.403	3.274
3	1014.373	850.539	16.151
4	469.952	320.570	31.787
5	962.975	906.408	5.874
6	433.654	362.341	16.445
7	114.048	93.248	18.238
8	4.157	5.546	33.423
9	103.981	45.956	55.803
10	420.430	324.591	22.796
11	105.858	47.538	55.092
12	105.858	48.973	53.737
13	420.430	332.608	20.889
14	958.879	905.862	5.529
15	429.558	362.635	15.580
16	109.952	69.816	36.503
17	99.885	41.997	57.955
18	3897.734	3872.096	0.658
19	2724.168	2716.429	0.284
20	1760.317	1868.154	6.126
21	1006.181	821.830	18.322

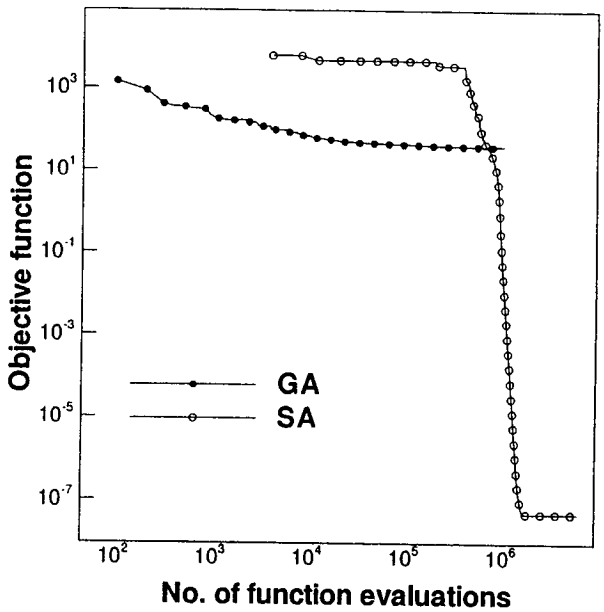


Figure C.1: Convergence history for the Rosenbrock test function with 40 design variables.

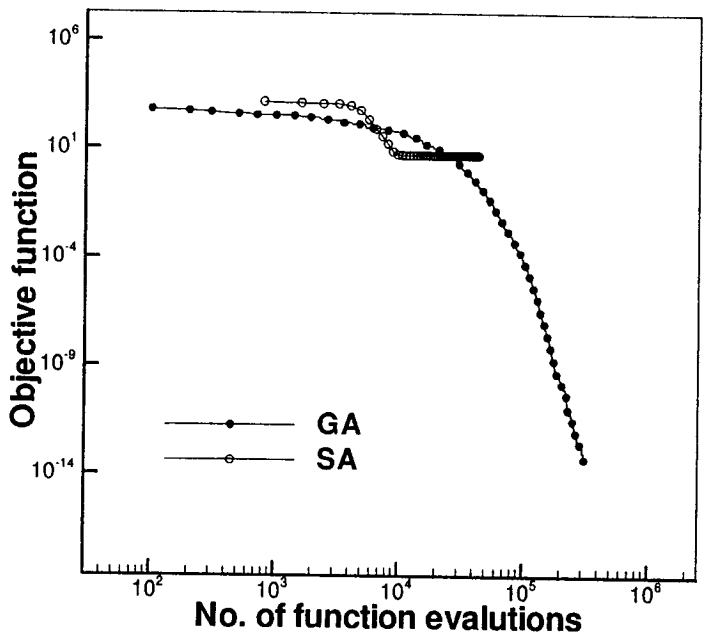


Figure C.2: Convergence history for the Rastrigin test function with 40 design variables.

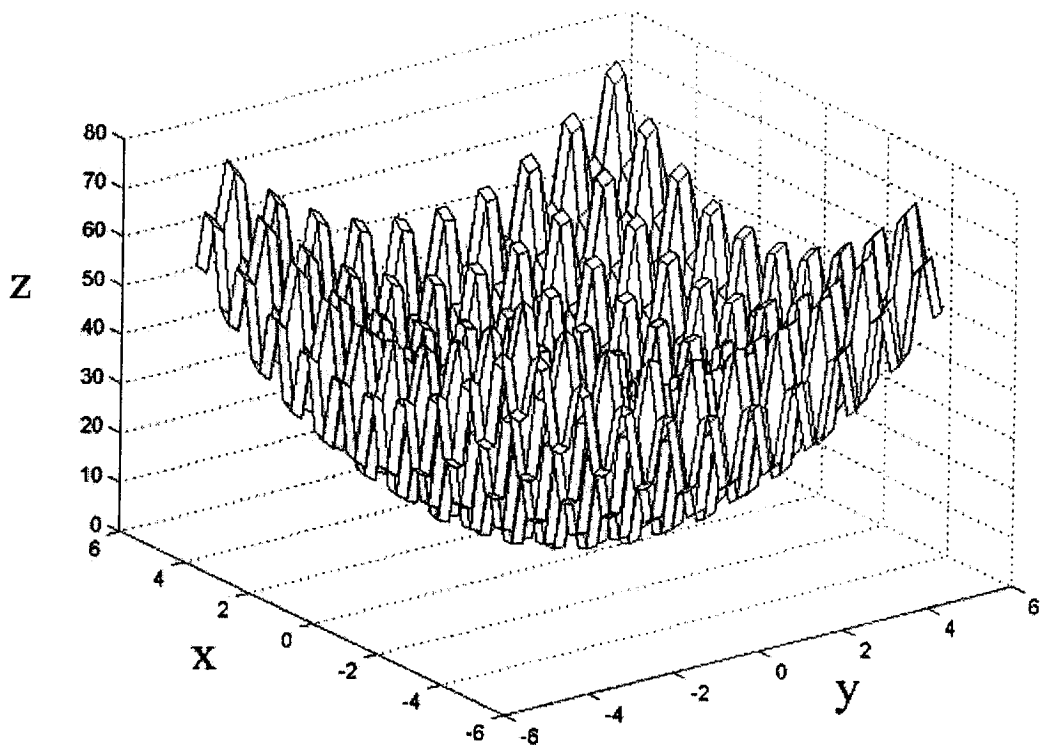


Figure C.3: The surface plot for Rastrigin's function.

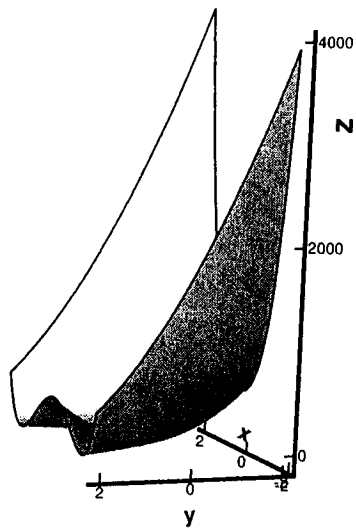


Figure C.4: The Rosenbrock surface plot.

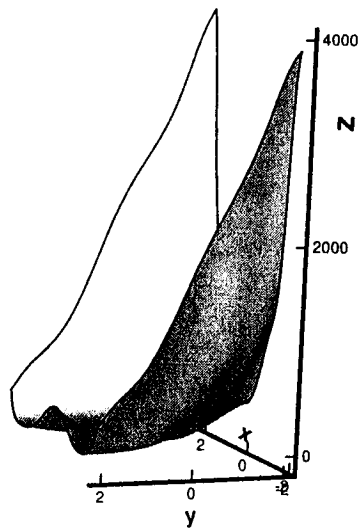


Figure C.5: The surface plot of ANN-approximation for the Rosenbrock's function.

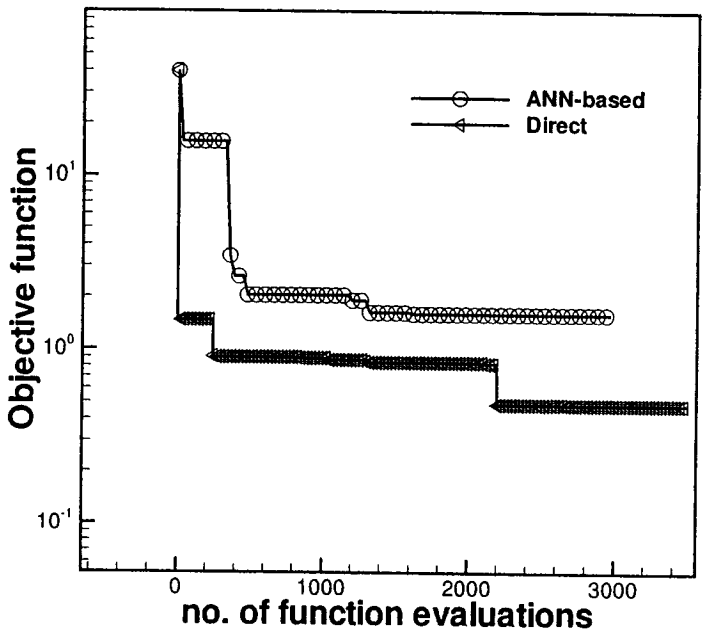


Figure C.6: Comparison of convergence history for the optimization of Rosenbrock test function based on ANN approximation with the optimization based on the exact function.