INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

**A Method for Designing Low-Speed Linear Turbine Cascades
for the Experimental Investigation of Viscous Effects in
Transonic Axial Turbines**

by

Kenneth James Zavitz, B. Eng. (Aerospace)

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of

**Master of Engineering
in Aerospace Engineering**

Ottawa-Carleton Institute for Mechanical and Aerospace Engineering

Department of Mechanical and Aerospace Engineering

Carleton University

Ottawa, Ontario

Canada

© 2000

Kenneth James Zavitz

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-52703-4

Canada

**The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis**

**A Method for Designing Low-Speed Linear Turbine Cascades for the Experimental
Investigation of Viscous Effects in Transonic Axial Turbines**

submitted by

Kenneth James Zavitz, B. Eng. (Aerospace)

**in partial fulfilment of the requirements for
the degree of Master of Engineering
in Aerospace Engineering**

**Professor S.A. Sjolander
Thesis Supervisor**

**Professor R. Bell
Chair, Dept. of Mechanical and Aerospace Engineering**

**Carleton University
May 2000**

Abstract

This thesis describes a systematic method for the design of low-speed linear turbine cascades that can be used for the experimental investigation of viscosity-based phenomena in transonic cascade flows. The design method involves the use of two optimization algorithms (a genetic algorithm and a gradient-based optimizer) and two direct flow solvers (a panel method solver and a Navier-Stokes solver). Seven tests of the method demonstrate its performance and reveal limitations. The method was found to work reasonably well for subsonic to transonic turbines with shock-free flows.

Acknowledgements

I would like to thank my thesis supervisor, Professor S.A. Sjolander, for his help in guiding and reviewing my work. I would like to thank my colleague, M.W. Benner, for providing me with a Navier-Stokes solver and a subroutine for importing CFD grids. I would also like to thank M.W. Benner for providing extensive technical support related to the Navier-Stokes solver. Thanks are also due to Professor R.J. Kind and Professor S.A. Sjolander for providing me with computer codes and support material that greatly facilitated my computer programming tasks.

Financial support for this study provided by the Natural Sciences and Engineering Research Council of Canada through a Research Grant (S.A. Sjolander) and a Postgraduate Scholarship (Ken Zavitz) is gratefully acknowledged.

Table of Contents

Acceptance Form	ii
Abstract	iii
Acknowledgements	iv
List of Tables	viii
List of Figures	x
Nomenclature	xii
Chapter 1: Introduction	1
1.1 Design Goal	1
1.2 Overview of Remainder of Thesis	4
Chapter 2: Literature Review	5
2.1 Use of Low-Speed Cascades to Investigate Viscous Effects in High-Speed Axial Turbomachinery	5
2.2 Inverse Design of Turbomachinery Airfoils based on Pressure Distribution	6
2.3 Design of Turbomachinery Airfoils using Optimization	8
2.4 Review of Optimization Algorithms	12
2.4.1 Methods of Optimization	12
2.4.2 Genetic Algorithms	16
Chapter 3 The Cascade Design Method	23
3.1 Introduction	23
3.2 The Design Method Algorithm	26
Chapter 4: Generation of the Cascade Geometry	29

Chapter 5: Flow Analysis Tools	32
5.1 General Remarks	32
5.2 Panel Method	33
5.2.1 Description of the Panel Method Code	33
5.2.2 Panel Generation	39
5.3 Navier-Stokes Code	42
5.3.1 Description of the Navier-Stokes Code	42
5.3.2 Description of the Grid Generator	46
Chapter 6: The Objective Function and Tools for Design Optimization	51
6.1 Introduction	51
6.2 The Objective Function	52
6.3 The Employed Genetic Algorithm: GAFDO	57
6.3.1 Description of GAFDO	57
6.3.2 Customization of GAFDO	62
6.3.2.1 General Remarks	62
6.3.2.2 Narrowing the Design Space	64
6.3.2.3 Population Sizes	68
6.3.2.4 Roulette Selection vs. Tournament Selection	70
6.3.2.5 Probability of Crossover and Mutation	72
6.3.3 Deciding When to Stop the Genetic Algorithm	73
6.4 The Employed Gradient-based Optimization Method	74

Chapter 7: Test Cases	76
7.1 Tests based on Low-Speed Cascades	76
7.1.1 General Remarks	76
7.1.2 Test Case: Cascade LS1	76
7.1.3 Test Case: Cascade LS3	81
7.2 Tests based on High-Speed Cascades	85
7.2.1 Overview of Tests	85
7.2.2 Test Case: Cascade HS1 (Target C_p s based on Navier-Stokes calculations)	86
7.2.3 Test Case: Cascade HS1 (Target C_p s based on Wind Tunnel Data)	91
7.2.4 Test Case: Cascade HS2 (based on exit Mach number of 0.8)	95
7.2.5 Test Case: Cascade HS2 (based on exit Mach number of 0.95)	102
7.2.6 Test Case: Cascade T106	106
Chapter 8: Conclusions and Potential Developments	111
8.1 Conclusions	111
8.2 Potential Modifications to the Cascade Design Package	112
References	115
Appendix A: The Cascade Generation Algorithm	119
Appendix B: User's Manual for Foil Design GAFDO	125

List of Tables

Table 6.1:	Standard and Suggested Settings for GAFDO for the Cascade Inverse Design Problem	60
Table 6.2:	Schedule for Constant that multiplies the Original Range and Range-Narrowing Constant to Determine the Minimum Allowable Range for Each Parameter	67
Table 6.3:	Parametric Study of Population Sizes and Design-Space Narrowing	69
Table 6.4:	Parametric Study of Roulette Selection vs. Tournament Selection	71
Table 7.1:	Wind Tunnel Conditions for LS1 Wind Tunnel Data	77
Table 7.2:	Settings for the Navier-Stokes Evaluations of the Inverse Design of LS1	77
Table 7.3:	Wind Tunnel Conditions for LS3 Wind Tunnel Data	81
Table 7.4:	Settings for the Navier-Stokes Evaluations of the Inverse Design of LS3	81
Table 7.5:	Settings for the Navier-Stokes Evaluation of the Original HS1	87
Table 7.6:	Settings for the Navier-Stokes Evaluations of the Low-Speed Design of HS1 (Target C_{ps} based on Navier-Stokes Target)	88
Table 7.7:	Wind Tunnel Conditions for HS1 Wind Tunnel Data	91
Table 7.8:	Settings for the Navier-Stokes Evaluations of the Low-Speed Design of HS1 (Target C_{ps} based on Wind Tunnel Data)	91
Table 7.9:	Wind Tunnel Conditions for HS2 Wind Tunnel Data (outlet Mach number: 0.8)	95
Table 7.10:	Settings for the Navier-Stokes Evaluations of the Low-Speed Designs of HS2	96
Table 7.11:	Wind Tunnel Conditions for HS2 Wind Tunnel Data (outlet Mach number: 0.95)	103

Table 7.12:	Wind Tunnel Conditions for T106 Wind Tunnel Data	106
Table 7.13:	Settings used for the Navier-Stokes Evaluations of the Low-Speed Design of T106	107

List of Figures

Figure 2.1:	A Multimodal Objective Function	13
Figure 2.2:	Crossover	18
Figure 2.3(a):	Fitness Scaling: Early Generations	21
Figure 2.3(b):	Fitness Scaling: Middle Generations	21
Figure 2.3(c):	Fitness Scaling: Late Generations	21
Figure 4.1:	Cascade Geometric Model	30
Figure 5.1:	Panel Method Prediction for an Aft-Loaded Low-Speed Cascade	35
Figure 5.2:	Panel Method Prediction for a Forward-Loaded Low-Speed Cascade	36
Figure 5.3:	Location of the Test Points used to Verify the Aft Stagnation Point Location	38
Figure 5.4:	Formation of a Panel Distribution	41
Figure 5.5(a):	CFD Grid in Trailing-Edge Region: Dense Grid	45
Figure 5.5(b):	CFD Grid in Trailing-Edge Region: Sparse Grid	45
Figure 5.6:	Representative Grid for Navier-Stokes Calculations	47
Figure 5.7:	Regions of the Grid	49
Figure 6.1(a):	Effect of Constant in Objective Function: $K = 3$	55
Figure 6.1(b):	Effect of Constant in Objective Function: $K = 5$	55
Figure 6.2:	Results of the Parametric Study of the Probability of Mutation and the Probability of Crossover.	73
Figure 7.1:	Pressure Distribution Comparison of LS1 and its Inverse Design	79
Figure 7.2:	Geometric Comparison of LS1 and its Inverse Design	80
Figure 7.3:	Pressure Distribution Comparison of LS3 and its Inverse Design	83
Figure 7.4:	Geometric Comparison of LS3 and its Inverse Design	84

Figure 7.5:	Comparison of Original HS1 and Inversely-Designed Low-Speed HS1 Pressure Distributions (Target C_p s based on Navier-Stokes Target)	89
Figure 7.6:	Comparison of Original HS1 and Inversely-Designed Low-Speed HS1 Geometries (Target C_p s based on Navier-Stokes Target)	90
Figure 7.7:	Comparison of Original HS1 and Inversely-Designed Low-Speed HS1 Pressure Distributions (Target C_p s based on Wind Tunnel Data)	93
Figure 7.8:	Comparison of Original HS1 and Inversely-Designed Low-Speed HS1 Geometries (Target C_p s based on Wind Tunnel Data)	94
Figure 7.9:	Comparison of Original HS2 and Inversely-Designed Low-Speed HS2 Pressure Distributions (based on exit Mach number of 0.8)	97
Figure 7.10:	Comparison of Original HS2 and Inversely-Designed Low-Speed HS2 Geometries (based on exit Mach number of 0.8)	99
Figure 7.11:	Pressure Distributions Based on Surface Distance and on Axial Distance	101
Figure 7.12:	Comparison of Original HS2 and Inversely-Designed Low-Speed HS2 Pressure Distributions (based on exit Mach number of 0.95)	104
Figure 7.13:	Comparison of Original HS2 and Inversely-Designed Low-Speed HS2 Geometries (based on exit Mach number of 0.95)	105
Figure 7.14:	Comparison of Original T106 and Inversely-Designed Low-Speed T106 Pressure Distributions	108
Figure 7.15:	Comparison of Original T106 and Inversely-Designed Low-Speed T106 Geometries	110

Nomenclature

Letters and Symbols

A	constant used for fitness scaling
B	constant used for fitness scaling
b	chord
b*	surface length for side of relevance: suction side or pressure side
C_p	pressure coefficient: $C_p = (P - P_{ref})/q_{ref}$
f	raw fitness value
f^*	scaled fitness value
f	the dependent variable of the objective function
g	a parameter used to define a cascade
i	an index used to number the target values of pressure coefficient
K	a constant appearing in the objective function
M	the number of parameters that are required to completely define a cascade
N	the number of influential target values in a given target pressure-coefficient distribution
P	static pressure
q	dynamic pressure = $\frac{1}{2}\rho V^2$
s	surface distance
V	velocity
x	axial co-ordinate
y	tangential coordinate

Greek Letters and Symbols

β	metal angle
β	Clauser's parameter
δ^*	boundary layer displacement thickness
ρ	density
τ_w	wall shear stress

Subscripts

2	outlet
avg	average
max	maximum
ref	reference value at an upstream, undisturbed flow reference location
x	axial

Chapter 1: Introduction

1.1 Design Goal

The detailed aerodynamic flow through a transonic turbine blade row is not easy to measure. Blockage effects from measurement probes, short run times, and small blade sizes make it expensive and difficult, if not impossible to gather flow-field data of the quality and resolution required for design improvements. Fortunately, there exists a technique by which a low-speed cascade may be used to gather experimental data related to the viscous effects of the flow through a transonic cascade of interest. Low-speed versions of cascades are much easier and less expensive to study experimentally because of their relatively large size and the convenience of using a low-speed rather than a high-speed wind tunnel. In order to be useful in a study of the viscous aspects of the aerodynamics of transonic axial turbine cascades, the low-speed cascades must have aerodynamic properties similar to those of their high-speed counterparts. The present thesis describes a method for designing such low-speed equivalents to high-speed turbine blade rows.

According to similarity principles, for compressible flow through a cascade, a geometrically similar blade shape will produce a similar flow if the Reynolds number, Mach number and pressure-coefficient distributions are matched. In addition, the free-stream turbulence intensity and surface roughness must be appropriately set to achieve flow similarity. For the purposes of this thesis, free-stream turbulence is assumed to be zero and the surfaces of the turbine blades are assumed to be hydraulically smooth.

Because a low-speed cascade will unavoidably have lower Mach numbers than a high-speed cascade, exact flow similarity is not achievable. The goal in designing a low-speed cascade must therefore be to reproduce the flow features that are of concern. For example, the designer might be interested in boundary layer transition, or the interaction between boundary layers and blade wakes. Boundary layer development and blade loading of the high-speed cascade are perhaps the most important flow features that should be matched by the low-speed cascade. These features are largely controlled by the surface pressure-coefficient distribution and the Reynolds number. The problem of designing a low-speed equivalent cascade is then one of inverse aerodynamic design based on the pressure-coefficient distribution of the given high-speed cascade. The low-speed design can be scaled to match the Reynolds number.

A common non-dimensional parameter used in the study of turbulent boundary layers is Clauser's parameter (White, 1991) given by:

$$\beta = (\delta^*/\tau_w)(dP/dx) \quad (1.1)$$

where δ^* is the boundary layer displacement thickness, τ_w is the wall shear stress. Clauser's parameter is related to the local pressure coefficient by:

$$\beta = ((\delta^*/b_x)/C_f)(d(C_f)/d(x/b_x)) \quad (1.2)$$

It can be argued that a low-speed cascade that has the same pressure-coefficient distribution as a given high-speed cascade (for the same Reynolds number) will have approximately the same distribution of Clauser's parameter provided that δ^*/b_x and C_f are approximately the same. The quantity δ^*/b_x is mainly a function of the local pressure gradient and the Reynolds number, and is therefore expected to be matched. C_f is mainly a function of

Reynolds number, surface roughness and Mach number. Of these parameters, only the Mach number cannot be matched between the flow cases. For Mach numbers less than unity, the effect of compressibility on C_f is small (e.g. see Fig. 7-24 from White, 1991).

As a result of the difference in compressibility effects resulting from unmatched Mach numbers, a low-speed equivalent cascade designed on the basis of pressure-coefficient distribution will, in general, have a different geometry than its high-speed counterpart. The present design method was developed for the purpose of designing such low-speed cascades. The method allows a designer to impose geometric constraints on the low-speed cascade design so that problematic designs can be avoided. For example, a design constraint may be added to require a minimum blade thickness.

A limitation to the method exists in that the pressure distribution of the high-speed cascade to be studied should not contain sharp gradients in pressure, such as those due to strong shock waves. This makes it difficult or even impossible for the method to yield a corresponding low-speed cascade geometry, especially when realistic geometric constraints are imposed. For this reason, the present design method is intended to apply only to subsonic and transonic cascades that are free of shocks.

1.2 Overview of Remainder of Thesis

The next chapter briefly reviews related work by others. It includes a discussion about the inverse design of turbomachinery airfoils based on prescribed pressure distributions. The present design method involves the use of an optimization strategy, hence a section is included that describes the design of turbomachinery airfoils using optimization methods. A subsection is devoted to the topic of genetic algorithms, since a genetic algorithm was used for optimization purposes in the implementation of the present design method.

Chapter 3 outlines the present method for designing low-speed equivalents of high-speed turbine cascades. Chapters 4, 5 and 6 discuss the three major components of the software implementation of the design method: a cascade geometry generator (Chapter 4), flow analysis tools (Chapter 5), and optimization tools (Chapter 6).

In Chapter 7, tests are performed to demonstrate the method. Two examples show how designing a low-speed cascade based on the pressure-coefficient distribution of a low-speed cascade results in an almost unchanged geometry, as it should. Another six examples of tests performed using the pressure-coefficient distributions of three different high-speed cascades show the geometric differences between the original high-speed cascades and their low-speed versions.

Chapter 2: Literature Review

2.1 Use of Low-Speed Cascades to Investigate Viscous Effects in High-Speed Axial Turbomachinery

The concept of using low-speed facilities to study the flow in a corresponding transonic turbomachinery blade row is not new. Wisler (1985) describes a large-scale low-speed facility with a rotating axial compressor rig. The facility is used to improve high-speed compressors in a variety of ways, such as increased efficiency and increased stall margin. The low-speed versions were first designed to have the same axial distributions of pressure coefficient as their high-speed counterparts. After using software to verify that boundary layer development should be similar (transition location and potential separation locations), the low-speed compressors were constructed and tested in the low-speed rotating rig. Changes to the low-speed designs were then proposed and tested experimentally until performance improvements were confirmed. Considering the probability of successfully scaling the changes to the high-speed case, the high-speed compressors were modified in a similar way. The modified high-speed compressors showed improved performance as well.

Lakshminarayana et al. (1992) describe a similar low-speed facility with a rotating axial turbine rig. The facility is used to gain insight into the aerodynamics of typical state-of-the-art high-speed turbines via low-speed model testing.

The design of low-speed versions of high-speed blade rows typically involves the specification of a blade surface distribution of pressure coefficient, Mach number or a similar

quantity. The next section provides a brief background to the problem of inversely designing turbomachinery (and other) airfoils from such surface distributions.

2.2 Inverse Design of Turbomachinery Airfoils based on Pressure Distribution

As stated in Section 1.1, the goal of the present work is to develop a method for designing low-speed cascades for use in the study of high-speed cascade aerodynamics. It was reasoned that, in order to promote similarity of viscous effects, such low-speed cascades should be designed to have the same pressure-coefficient distributions as their high-speed counterparts. Section 2.1 mentioned that low-speed axial compressors designed on this basis have already been used for research. This section provides a brief review of some of the methods by which turbomachinery airfoils have been inversely designed from such aerodynamic requirements.

The problem of designing a compressor or turbine cascade to meet aerodynamic requirements, such as having an acceptable pressure distribution, correct flow turning or minimal profile losses, is a familiar problem to engineers involved in the design of turbomachinery cascades. Such problems have commonly been handled by using so-called "direct" methods of design. These methods employ standard analysis tools to check cascade designs in an iterative design process, usually involving much human interaction and decision making. The efficiency of these methods often depends on the experience of the designer. In addition, the resulting design is not guaranteed to be optimal. An example of a direct design method that applies to turbomachinery blade design is provided by Ye (1984). The design package used to implement that method essentially consisted of a collection of analysis tools that were used in a standard or

manually chosen sequence to investigate aspects of cascade performance at various levels of detail, plus a blade profile generator that was manually controlled by the user to allow changes to the design.

In order to reduce the amount of effort required by designers, so-called “inverse” methods have been developed. These usually involve the use of a flow solver that is capable of solving for the geometry of an airfoil based on prescribed surface distributions of some quantity such as pressure-coefficient or Mach number. For example, Léonard and Van den Braembussche (1991, 1992) used an inverse Euler solver that solved for the shape of a compressor or turbine blade based on a specified surface Mach number or pressure distribution. Their 1992 paper dealt with strategies for imposing geometrical and mechanical restrictions to the inverse designs. Demeulenaere and Van den Braembussche (1996) describe a 3-D inverse Euler flow solver for use in designing turbomachinery blades. This inverse flow solver, capable of handling rotational inviscid flows having shocks, was used as part of a method that also involved direct analysis and refinement of the design, as for a typical direct method. Singh (1996) describes a quasi 3-D method for designing turbomachinery blades. Viscous effects were taken into account by combining an integral boundary layer prediction method with an inverse Euler solver.

In recent years, methods of inverse cascade design based on optimization have become popular. These methods are iterative in nature and, like conventional direct methods, employ direct analysis tools to assess the designs. Unlike conventional direct methods, however, these methods are more automated and less dependent on human expertise and heuristics in order to improve a design, and usually yield designs that are better optimized. The next section describes the application of optimization techniques to turbomachinery blade design.

2.3 Design of Turbomachinery Airfoils using Optimization

Optimization strategies for turbomachinery blade design may involve formulating the design problem, including any applicable design constraints, into an objective function that relates the independent design parameters to the design objective. For example, in the case of turbine cascade design optimization, the set of independent design parameters could include blade chord, blade spacing and the inlet metal angle. The set of independent design parameters must completely specify a turbine cascade design. The design objective is reflected in the value of the dependent variable of the objective function. In this case, the dependent variable could be assigned an index value that would reflect the merits of the design in light of the specified design requirements. For example, if the design requirement is to produce a given amount of flow turning, the index value could be formulated as the absolute value of the difference between the actual and desired flow turning angles.

In the context of function optimization, the *search space* is the space formed by using the independent variables of the objective function as dimensions. For engineering applications, a search space is sometimes called a *design space*. A maximizer is a point in the search space (a set of the independent parameters) that maximizes the value of the dependent variable of the objective function. Conversely, a minimizer is a point that minimizes the dependent variable. The problem of function optimization is then one of finding the maximizer or the minimizer, depending on the goal. Optimization of the objective function in the above example for a turbine cascade would be performed by searching for the minimizer. Once the minimizer is found, its coordinates in the search space would define the set of independent design parameters that define a turbine cascade design optimized to produce the desired amount of flow turning.

Numerous methods are available for function optimization. Among the optimization algorithms used in the design of turbomachinery airfoils are genetic algorithms (GAs) and line search methods. These and other methods are described in the next section. The remainder of this section surveys some of the applications of such optimization methods in problems of, and related to, turbomachinery blade design.

Cravero and Dawes (1997) describe a procedure for optimizing the design of an axial turbine stage. The method incorporates a streamline curvature throughflow code, loss correlations and an optimization algorithm. The objective was to improve stage efficiency by twisting and re-staggering the blades. To demonstrate the method, a high-pressure axial steam turbine stage was optimized. The initial and optimized stage designs were analyzed using a 3-D Navier-Stokes flow solver, confirming improved performance.

Goel et al. (1996) describe a general purpose turbine design and optimization method that uses three optimization algorithms to automate a direct design procedure: numerical optimization, a gradient-following method, and a genetic algorithm. The three algorithms each have their own advantages and disadvantages. In particular, the genetic algorithm has an excellent ability to locate the vicinity of the global optimum of the objective function, but lacks computational efficiency due to the requirement to perform direct analyses for a large number of candidate designs. The method involves switching between the available optimization algorithms in an attempt to exploit the best qualities of each. For example, when the genetic algorithm starts to show reduced gains in optimization with time, a gradient-following method is used to continue the optimization with greater efficiency. Design heuristics were also used to aid the optimization process.

Vicini and Quagliarella (1997) provide an insightful discussion of the role that genetic algorithms have in methods for designing airfoils (not turbomachinery airfoils, though). They describe the strengths and weaknesses of GAs and provide an understanding of how GA-based methods are implemented. They also describe how GA-based methods of design compare with others, such as conventional inverse methods and direct methods. The argument is made that GAs are useful for performing optimization up until the point where a nearly-optimal design is found; after that point, it would be more efficient to use another method such as one based on gradients of the objective function. Chan (1998) provides a similar discussion about the suitability of genetic algorithms for use in aerodynamic design in general.

Trigg et al. (1997) describe a method for designing steam turbine blades for minimum profile losses by using a genetic algorithm coupled with a 2-D Navier-Stokes solver. Geometric constraints were imposed by restricting the domain of the objective function to include only mechanically feasible designs. An example was given that showed how a blade designed using this method had 5% less profile losses than a blade that was considered to be typical of what would be produced by an expert human designer without the aid of an optimization tool.

In a similar paper by Dennis et al. (1999), the total pressure loss across a 2-D linear turbine cascade row was reduced by using a genetic algorithm coupled with a Navier-Stokes flow solver. The design method involved an elaborate set of design constraints by comparison to Trigg et al. (1997).

Giannakoglou (1999) presents an inverse design method for designing turbomachinery blades based on a required surface pressure or velocity distribution. The method uses a genetic algorithm coupled with a direct finite-volume flow solver that uses unstructured grids. In order to

minimize the time required to analyze the numerous candidate designs involved in the optimization process, an Artificial Neural Network (ANN) is trained to estimate which designs are not worth investigating using the flow solver. The estimates are derived from a database of stored cascade results.

Pierret and Van den Braembussche (1998) describe another method for designing turbomachinery blades that makes use of an ANN, an optimization strategy and a 2-D Navier-Stokes flow solver. The optimization method was Simulated Annealing (SA): see Section 2.4.1. As with Giannakoglou (1999), the ANN is used to aid the optimization strategy by providing approximations for performance. Instead of using the ANN to decide which designs should be further analyzed using the Navier-Stokes flow solver, the ANN directly provides estimates of performance, replacing the flow solver. A Navier-Stokes evaluation of the design is made at the end of each global design iteration and is added to a database of stored results, helping to improve the accuracy of the ANN predictions with each design iteration.

Tong and Gregory (1992) describe a tool for the preliminary design of single or multistage turbines. The design tool uses a combination of expert systems, numerical optimization and machine learning in order to search the design space for the best design possible. The expert system emulates a human design expert by using customized, problem-specific heuristics and strategies to quickly find a reasonably good design. Subsequently, alternate use of the expert system and numerical optimization techniques finds a design closer to the nearest local optimum of the objective function. A machine learning strategy, namely a genetic algorithm, is then used to explore the design space further in an attempt to find a better design. If a promising design is found, further optimization is performed using a combination of the expert system and numerical optimization, as before. With the increase in computational

speed since then, expert systems may be less justifiable as a means to reduce computational time since they greatly increase the complexity of the design tool.

The remaining sections of this chapter provide background information regarding optimization strategies including the genetic algorithm.

2.4 Review of Optimization Algorithms

2.4.1 Methods of Optimization

Many problems involving engineering design optimization may be formulated as a problem of finding an extreme value (maximum or minimum) of an objective function. Section 2.3 introduced the concept of an objective function and some of the terminology associated with optimization. An objective function may be continuous or discontinuous and may have one or more local maximum and/or minimum values. The word “multimodal” is used to describe objective functions having more than one local maximum or minimum. In problems involving aerodynamic design, the objective function may easily be multimodal. An example of a hypothetical multimodal objective function is shown in Figure 2.1. Usually, problem constraints are used in order to restrict the extent of the search space, limiting the size of the space that must be searched. Probably the simplest way of limiting the search space is to define an allowable range for each independent parameter.

Figure 2.1: A Multimodal Objective Function

Many optimization techniques have been devised to search for local maximum or minimum values of the dependent quantity. Line search methods are traditional methods that have been widely used in practical optimization problems. These methods employ gradient information (of the dependent variable with respect to the independent variables) to decide on a direction of search, represented by a vector in the search space. Once the search direction is known, an offset in the search direction is applied to a starting point in the search space in an effort to locate another point closer to the maximizer (or minimizer). By iteratively following search vectors from the previous point, the maximizer (or minimizer) may be found (or closely

approximated). Such methods include the line search Newton's method (and related methods), conjugate direction method, and descent methods, all of which are described by Fletcher (1987). Many line search methods are based on a quadratic model of the objective function in which a quadratic function is used to approximate portions of the objective function. In the process of defining search vectors, these methods make use of the first and second derivatives of the dependent variable with respect to the independent variables. As a result, they are generally more efficient and yield more accurate results than linear methods, which ignore second derivative information. Whenever the gradient information required by quadratic methods is not available, it is sometimes reasonable to approximate the gradients by using finite difference techniques, such as the forward difference or central difference. For example, this has led to the development of quasi-Newton methods which use approximations for second derivative information.

The optimization techniques mentioned so far must be provided with a suitable starting point in the search space in order to find the global optimum. Such a starting point may be provided by a guess, by using heuristics, or by using a different kind of search technique that more widely explores the search space. These techniques are usually less efficient than gradient-based methods of search. However, they are often useful in providing a starting point for other optimization techniques when the objective function is multimodal. Perhaps the simplest of the techniques for finding a region of the search space near the global maximum/minimum value is an enumerative search. This method works by systematically searching the search space by visiting each point (using a sufficiently fine resolution) and then reporting the coordinates of the point having the overall maximum/minimum value of the dependent quantity. Of course, this method is extremely inefficient and therefore is not used for most engineering problems. In order to cover vast search spaces more efficiently, many techniques have been devised that start by examining several points widely distributed over the search space and then gradually search only

in regions showing the most promise of containing the global optimum. Some of these techniques, such as simulated annealing and genetic algorithms, employ stochastic mechanisms in order to direct the search. Other methods use heuristics in order to direct the search.

Simulated annealing (Ansari and Hou, 1997) is a method of optimization that borrows principles from the annealing process of metal in which, under slow cooling from a near-melting temperature, the metal atoms settle into lower-energy configurations. At higher temperatures, the metal atoms are more mobile, and are capable of moving to produce higher-energy as well as lower-energy configurations. As the temperature is reduced, less thermal energy is available to allow the atoms to move to higher-energy configurations, and changes that produce lower-energy configurations become more common. The annealing process is analogous to a search for the minimum value of the dependent variable in a search space by using simulated annealing: particular configurations of the metal atoms are analogous to points in the search space, atomic configurations of less total energy are analogous to points having a lower associated value of the dependent variable. In the case of simulated annealing, points in the search space are visited sequentially by moving short "distances" through the space in random directions. The "temperature" in simulated annealing governs the probability that a move to a higher value of the dependent variable will be accepted. As the "temperature" drops according to a cooling schedule, movements to higher values of the dependent variable are more likely to be rejected (more so for movements that produce larger increases of the dependent variable). Eventually, points having a minimal value of the dependent variable are reached. The advantage of simulated annealing over simple gradient-based methods is that, by allowing the occasional movement towards higher values of the dependent variable, there is protection against the path becoming trapped at a local minimum.

The genetic algorithm is another method of optimization that uses stochastic processes and is able to find global, rather than local, extremes in the dependent variable. GAs are becoming increasingly popular as design and optimization tools for a wide variety of engineering problems, as well as problems from other disciplines. The next section describes the workings of basic genetic algorithms.

2.4.2 Genetic Algorithms

The genetic algorithm (Goldberg, 1989, Holland, 1992, Ansari and Hou, 1997) is a popular method for function optimization. It is robust and relatively simple to implement. Many variations of genetic algorithms are possible. All are based on the principles of genetic evolution. This section describes the basics of GAs and explains some of the associated terminology.

The function to be optimized, called the objective function, or *fitness function* is a function relating one dependent parameter, *fitness*, to several independent design parameters. A given set of design parameters defines an *individual*, or *design*, the merits of which are ranked by the associated value of fitness (higher fitness values correspond to designs that better meet the design requirements). The GA searches the design space in an attempt to find the design (in the form of design parameters) that corresponds to the highest value of fitness.

The GA mimics natural evolution by letting a population of individuals (designs) evolve over many generations in a “survival of the fittest” scenario. The individuals of highest fitness have the greatest likelihood of becoming “parents”. Pairs of “parents” combine genetically to create new designs, forming the next generation. The average (and maximum) fitness of the members in a population generally increases over the generations as the genetic information of

the fittest individuals becomes more prevalent among the members. The set of parameter values representing the most-fit individual ever encountered is retained and becomes the final product of the optimization.

In order to mimic the mechanics of some of the genetic processes in nature, the design parameters are stored in a form called a chromosome. The parameters are represented by one or more packages of genetic information, called genes. A gene typically consists of a string of digits. For computer implementation purposes, a binary string of ones and zeros is often used. The set of genes belonging to a particular individual are concatenated to form a chromosome. Simple implementations of genetic algorithms store the information required to represent an individual using only a single chromosome. In nature, however, the genetic information of a living cell is usually contained in several chromosomes. The linear structure of a chromosome is conducive to the exchange of genetic information between parent designs. In particular, the process of *crossover* involves the exchange of corresponding segments of parent chromosomes to form child chromosomes.

The operators used to form the chromosomes of children designs are called *genetic operators*. One operator, already mentioned, is crossover. Without crossover, a child chromosome would be formed from a chromosome of only one of its parents. Crossover is implemented by swapping sections of one parental chromosome with corresponding sections from the chromosome of another parent. The result is a pair of child chromosomes, each containing a mix of genetic information from both of its parents. The locations along a child chromosome at which the original source of the genetic information switches from one parent to the other are randomly selected and are called *crossover points*. Figure 2.2 illustrates the process of crossover.

Figure 2.2: Crossover

Mutation is another genetic operator, used to help compensate for the occasional loss of “good” genetic material from the population. Mutation is the random alteration of the genetic information. One form of the mutation operator is the reversal of a gene digit (from a zero to a one or vice versa), applied with an equally low probability to every digit of a child’s chromosome. Increasing the rate of mutation can be beneficial to the evolutionary process. Too high a mutation rate, however, can disrupt the process of evolution by introducing too many random changes to the chromosomes.

In order to select pairs of parent individuals, a process called *selection* is used. Commonly used selection procedures are tournament selection and roulette wheel selection. In tournament selection, pairs of different individuals are selected at random (without bias) from the population and the fittest individual of the pair is selected to become a parent. In roulette wheel selection, a numerical version of a roulette wheel is spun to select an individual from the

population. Each individual in the population has a “slot” on the roulette wheel. The size of each slot on the wheel is directly related to the level of fitness of each associated individual. To improve the performance of the GA, roulette wheel selection should be accompanied by fitness scaling (described next).

Fitness scaling refers to the adjustment of the fitness values of the members of a population in order to enhance GA performance. After the fitness values have been initially determined, they are scaled so that they are brought closer to the population average in early generations and farther from the average in later generations. This helps prevent the loss of gene pool diversity that would occur if a few highly fit individuals dominated the early generations. It also helps amplify differences in fitness later in the evolution, when the progress of the GA is typically slower. Fitness scaling may be linear or non-linear, depending on the nature of the adjustments to the original fitness values. In particular, linear fitness scaling is a simple but effective method of fitness scaling.

Linear fitness scaling involves mapping the original, or raw, fitness values to scaled values using the formula:

$$f' = Af + (1 - A)f_{avg} \quad (2.1)$$

where: f is the raw fitness value, f' is the scaled fitness value; A is a constant; and the subscript “avg” refers to the population average. Note that the second term on the right-hand side is constant for a given population. Also note that a fitness value equal to the population average will be scaled to itself, i.e.:

$$f'_{avg} = f_{avg} \quad (2.2)$$

Figure 2.3(a) shows how linear scaling maps raw fitness values to scaled values for a typical early generation of a GA run. The line with slope A is used to map raw fitness values to scaled fitness values and is represented by Eqn. 2.1. A line having a slope of unity and passing through the origin is included to demonstrate that the average fitness of the population members remains unchanged during the scaling process. Constant A in Eqn. 2.1 directly adjusts the spread of the fitness distribution. The value of constant A is calculated so that the ratio of the maximum fitness to the average fitness is, after scaling, equal to a specified constant. In this work, the latter is called the *fitness scaling parameter*. Constant A is therefore calculated according to the equation:

$$A = (\text{fitness scaling parameter} - 1) / (f_{\max} / f_{\text{avg}} - 1) \quad (2.3)$$

A reasonable range for the fitness scaling parameter is 1.2 to 2.0 (Goldberg, 1989). In the examples presented in Figure 2.3, the fitness scaling parameter is 1.25.

As shown in Fig. 2.3(a), the scaled fitness values for a typical early generation have less spread than the raw fitness values. Figure 2.3(b) shows how the same type of scaling affects a typical generation of the GA run. Note how, due to the distribution of raw fitness values, the scaled fitness values have greater spread (as indicated by the fact that A is greater than unity). This is a consequence of maintaining the ratio of maximum to average scaled fitness values at 1.25 while the average raw fitness value has become relatively close to the maximum raw fitness value.

If, in later generations, a negative scaled fitness value would normally appear, the constant A is replaced by a different constant: B. The latter is set so that the minimum raw fitness value is scaled to zero, thus providing the maximum spread in the scaled fitness values that is

(a) Early Generations

(b) Middle Generations

(c) Late Generations

Figure 2.3: Fitness Scaling

possible without having any negative scaled fitness values. Figure 2.3(c) shows such scaling of fitness values for a typical late generation that contains at least one individual of relatively low fitness. The ratio of the maximum scaled fitness to the average scaled fitness is reduced from what it would have been had constant A been used.

Some GAs employ a strategy whereby the design space is narrowed to concentrate exploration in the most promising region. The borders of the design space are periodically adjusted according to the regions occupied by the fittest individuals of the population. Individuals that fall outside the new design space must then be replaced. Such a narrowing of the design space can profoundly increase the rate of progress of the GA with generation.

To try to further increase the computational efficiency of genetic algorithms, some researchers make use of so-called *micro-genetic algorithms* (Krishnakumar, 1989). These algorithms use smaller than normal population sizes to reduce the amount of computational time required to process a given number of generations. High mutation rates and periodic removal and replacement of individuals from a population are tactics that are used in micro-genetic algorithms to help maintain genetic diversity. In order to help avoid the accidental loss of the best chromosome, a likely event when the population is small, a tactic called *elitism* is employed. Elitism is the promotion of the most-fit individual of a population to the next generation. Elitism is highly beneficial when the population size is very small (e.g. a population of 10). When larger populations are used, elitism may have less of an influence on GA performance.

This section concludes the literature review. The next four chapters describe the present method for designing low-speed turbine cascades for the study of viscous effects in high-speed turbine blade rows.

Chapter 3 The Cascade Design Method

3.1 Introduction

Chapter 1 introduced the problem of designing a low-speed equivalent of a high-speed turbine cascade. It was argued that complete aerodynamic similarity was not possible, but that most of the important viscous effects could be closely reproduced by the low-speed version of the cascade if it had the same pressure-coefficient distribution as the original (high-speed) cascade. It was mentioned that the Reynolds number, free-stream turbulence intensity and surface roughness must also be matched, but these are not the main focus of the design process. The problem of designing a low-speed equivalent cascade was argued to be essentially one of inverse aerodynamic design based on the pressure-coefficient distribution of the high-speed cascade.

The present design method does not require the use of inverse design codes. Rather, ordinary direct analysis codes are used. In the present work, a 2-D panel method and a 2-D incompressible Navier-Stokes solver were used to perform the necessary flow analyses. The pressure-coefficient distributions from low-speed cascade candidate designs are determined from computed flow results and compared with a corresponding distribution from the high-speed cascade. The pressure-coefficient distribution, as used in the present design process, is expressed as a function of the non-dimensional axial coordinate, with pressure-coefficients defined by:

$$C_p = (P - P_{ref}) / q_{ref} \quad (3.1)$$

where the subscript “ref” indicates a value taken at the reference location: an upstream,

undisturbed flow location. Because the axial coordinate normalized by the axial chord length is the basis for pressure distribution, it is convenient to use the axial chord length as the basis for Reynolds number matching. This involves sizing the low-speed cascade so that, for the intended low-speed wind tunnel free-stream velocity, the Reynolds number will match that of the high-speed cascade. The difference between the pressure-coefficient distributions of the low-speed and high-speed cascades is minimized by the use of optimization algorithms that modify the geometric parameters of the low-speed cascade. Chapter 6 explains this process in detail.

The main optimization algorithm, chosen for its robustness, is the genetic algorithm (see Section 2.4.2). When the genetic algorithm has identified the region of the global optimum but starts to lose efficiency, a gradient-based search method continues the optimization. The latter method simply makes small changes to the current design to determine which further changes should be made to yield a better design. If the best design arrived at by the genetic algorithm is close to optimal, the optimum will be reached in a systematic way with greater efficiency than would be achieved using the genetic algorithm. If, however, the initial design is far from optimal, there is a risk of converging on a design that is a local optimum, rather than the global optimum. This is because the fitness function may be multimodal (see Fig. 2.1) depending on its formulation, which includes any imposed design constraints. Using a genetic algorithm followed by a gradient-based search method results in an optimization process that is both robust and efficient.

The main problem with using a genetic algorithm is the inherent need to perform a large number of fitness evaluations. This translates to a requirement to evaluate the pressure-coefficient distributions of a large number of trial cascade designs. The present design method

partially compensates for this problem by performing most fitness evaluations using the panel method flow solver, and very few (typically 3 or 4) evaluations using the Navier-Stokes flow solver. The panel method is very fast by comparison to the Navier-Stokes code. A panel method result using 56 panels per blade and 29 blades takes less than one second on a 300 MHz Pentium-based desktop computer. A high quality Navier-Stokes result (2500 iterations with a grid of 230 by 140 cells) takes roughly six hours to generate on the same computer. While the computational time is reduced by several orders of magnitude by using the panel method for the majority of flow evaluations, the level of accuracy of the resulting cascade geometry is the same as if the optimization codes had been using only the Navier-Stokes solver.

For the first global iteration of the present design method, the C_p distribution of the high-speed turbine blade row is used as the target. The optimization tools are used to inversely design a cascade based on the target, using only the panel method for flow predictions. At the end of the first global iteration, the resulting cascade geometry closely produces the target pressure distribution when tested by the panel method. The flow field for this geometry is then analyzed using the Navier-Stokes solver. Inevitably, the C_p results differ somewhat from the potential flow results; experience shows that the differences occur mainly towards the trailing edge on the suction side. The differences between the two pressure distributions are then used as an offset from the original target distribution to develop a modified target C_p distribution. This temporary target is intended to guide optimization via the panel method so that the resulting design will satisfy the original target when tested using the Navier-Stokes code. For example, if the geometry arrived at using the panel method gave less than the required suction in a particular region, as indicated by the Navier-Stokes analysis, the modified target distribution would call for more suction in that region. The complete procedure is then repeated for a second iteration. As in the first iteration, the final step is a Navier-Stokes analysis of the geometry arrived at using the

panel method. Experience shows that only three or four iterations are necessary to achieve a geometry which gives the true target C_p distribution, as predicted by the Navier-Stokes code, with the desired precision. As indicated, each global iteration involves only a single Navier-Stokes analysis.

The basic operational nature of the design method has been presented in this section. The next section outlines a step-by-step algorithm for the method.

3.2 The Design Method Algorithm

The design algorithm is iterative in nature and requires minor supervision and decision-making on the part of the designer. The steps of the algorithm follow:

Step 1. The pressure-coefficient distribution of the high-speed cascade is determined as accurately as possible (by whatever means) and becomes the main target for the inverse design.

Step 2. The genetic algorithm is used to design a cascade that closely produces the current target pressure-coefficient distribution when analyzed using the panel method. The evolution is allowed to proceed for 750 generations (see Section 6.3.3).

Step 3. The gradient-based search method (see Section 6.4) is used to refine the design (also using the panel method).

Step 4. The Navier-Stokes flow solver is used to estimate the pressure-coefficient distribution that would be obtained by the current low-speed design in a low-speed wind tunnel. This pressure-

coefficient distribution is then plotted along with the main target distribution (from Step 1) so that a visual comparison can be made. If the pressure distributions agree to a satisfactory degree, the design process is complete. If not, the design process continues with Step 5.

Step 5. The panel method is used to obtain the pressure-coefficient distribution for the current low-speed design.

Step 6. A modified target pressure-coefficient distribution is derived from the distributions of Steps 1, 4 and 5. The modified target distribution is formed for each point in the original target distribution by using equation 3.2. Every term in the equation represents a pressure coefficient value located at the same non-dimensional axial position and on the same side of the blade.

$$\text{modified target} = \text{target} + (\text{panel method: current design} - \text{Navier-Stokes: current design}) \quad (3.2)$$

Step 7. Repeat starting at Step 2 using the modified target as the current target. At the risk of having problems with progress in optimization, Step 2 may be skipped, and the process could repeat starting at Step 3 using the current low-speed design as a starting point for the gradient-based optimizer. Skipping Step 2 is less risky if the difference between the target and modified target is small.

The modified target is designed so that, when it is satisfied by a panel method evaluation of a design, that same design will satisfy the true target when evaluated by the Navier-Stokes code. The method of forming the modified target (Step 6) assumes that the difference in pressure-coefficient distribution between panel method and Navier-Stokes evaluations for the

latest design will not change greatly with design improvements. This assumption is often a good one because the initial design, obtained by using the panel method with the true target C_p distribution, tends to be geometrically close to the final design, which is intended to match the true target when tested with the Navier-Stokes code.

The design method was implemented so that its performance could be studied. The next three chapters discuss the main components of the implementation: the algorithm for constructing cascade geometry from a set of geometric parameters (Chapter 4), the flow solvers (Chapter 5), and the optimization algorithms (Chapter 6). Several test cases are presented in Chapter 7.

Chapter 4 Generation of the Cascade Geometry

An essential component of the design package is an algorithm that converts a set of independent design parameters into a cascade geometry. The algorithm provides blade profile coordinates along with blade spacing, to completely define a linear 2-D cascade geometry.

The cascade geometry is specified by 25 independent parameters. The geometry model is an extension of an 18 parameter model used by Islam & Sjolander (1999) to construct hypothetical blade rows as part of a study of deviation in axial turbines. The earlier model used a leading-edge ellipse and a trailing-edge circle. The suction side was defined by two Bezier cubic polynomials (Farin, 1988) which joined at the throat. Continuity of curvature was maintained at the join point. A single Bezier cubic defined the pressure side of the blade. In the present study, it was found that this earlier model gave insufficient control over the details of the geometry. Several modifications were therefore made.

Figure 4.1 summarizes the modified geometry model. The x-axis is aligned in the axial direction (i.e. axis of rotation in the case of a turbomachine). Join points indicate the connections between adjacent curves. They are labeled with a letter and a number. The letter is "S" for the suction side and "P" for the pressure side of the blade. The suction side is represented by three Bezier cubic polynomials. Neither of the join points surrounding the middle Bezier curve is required to lie at the throat. Two Bezier cubics define the pressure side. A pair of control points is used to control the end slopes and the general shape of each Bezier curve. The tangent line at the end of a Bezier cubic polynomial passes through the control point associated with that end. The distance of each control point from its associated end of the curve can be adjusted to alter the

SYMBOL DEFINITIONS

- JOIN POINT
- BEZIER CUBIC CONTROL POINT

Figure 4.1: Cascade Geometric Model

curvature distribution along the curve. Continuity of curvature is enforced at the join points between Bezier cubics. As the model is presently formulated, some iteration is needed in the calculations to enforce this continuity of curvature. However, the method has been found to be very robust and essentially no convergence problems have been encountered.

Several of the geometric parameters commonly used to define turbomachinery blade rows, such as tangential chord, trailing-edge wedge angle and spacing, appear in the present geometric specification. Other geometric parameters, such as the location of the control points for the Bezier cubics, are only meaningful in the context of this particular method of geometric construction. Furthermore, some important geometric parameters, such as the throat width and the unguided turning, are not used directly to specify the cascade geometry. The resulting values of these quantities are calculated after the geometry has been established. The independent parameters used in the present model have been chosen specifically to allow reliable, automatic generation of trial cascade geometries by the optimization methods. The model may be less suitable for use in a direct blade design system. For a detailed description of the algorithm for designing cascade geometries, see Appendix A.

Chapter 5: Flow Analysis Tools

5.1 General Remarks

A well-designed turbine blade row has thin boundary layers and minimal flow separation. The most important viscous effects are confined to the thin boundary layers along the blade surfaces, making an assumption of zero viscosity (inherent in potential flow) reasonable. Inviscid flow solutions for turbine cascades predict surface pressures reasonably well provided that the appropriate inviscid outflow angle is directly or indirectly enforced. A 2-D panel method potential flow solver was incorporated into the present design package. It is used to provide rapid estimates of pressure distribution with a level of accuracy sufficient to show trends in flow behavior with changes in cascade geometry.

For more accurate calculations of surface pressure, a 2-D incompressible viscous Navier-Stokes code is used. It is able to model viscous effects such as boundary layers and wakes. Because the Navier-Stokes evaluations are computationally expensive, they are used sparingly in the design process: a Navier-Stokes evaluation is made only at the end of each global iteration of the design process. The pressure-coefficient distribution calculated from the flow solution is used to confirm the cascade design or, if the design is not satisfactory, to correct the course of optimization for the next iteration (see Chapter 3 for a description of the design method).

The panel method and Navier-Stokes flow solvers used in the present work are discussed in more detail in the remaining sections of this chapter.

5.2 Panel Method

5.2.1 Description of the Panel Method Code

The two-dimensional potential flows through trial cascades are calculated using a panel method code. The code was developed for, and used earlier in, a study of the effects of surface roughness on the aerodynamics of turbine blades (Kind et al., 1998). The primary result of the code is the surface pressure-coefficient distribution around a blade. The code is also used to predict the velocity field away from the blades for use in initializing Navier-Stokes code runs.

The panel method is named for the way in which geometry is modeled. A blade profile is represented by flat panels consisting of two-dimensional vortex sheets of linearly-varying strength. The panels are concentrated around the leading and trailing edges (see Section 5.2.2 for a description of the method used to distribute panels). The GA uses 56 panels per blade as a standard setting. This was (optionally) increased to 80 panels to improve accuracy while using the gradient-based optimizer to fine-tune the solution of the GA.

The flow through an infinite cascade is approximated by the flow through a finite row of blades, each represented by the same distribution of vortex sheets. The desired flow is taken as that predicted for the middle blade in the row. A row of 29 blades was found to give a good compromise between speed and accuracy. The slight inaccuracy due to the assumption of a finite number of blades has no effect on the final cascade design because Navier-Stokes calculations, which use repeating boundary conditions to represent infinite cascades, are used to correct the design.

Figure 5.1 shows a comparison of the pressure-coefficient distribution predicted by the panel method with that obtained from wind tunnel data (Whitehouse, 1993) for a typical low-speed cascade. This cascade, designated “LS3” at Carleton University, is designed for low-speed wind tunnel studies and is representative of the mid-span section of a modern, low-speed turbine blade row. The cascade was modeled using 29 blades, each with the same distribution of 56 panels. Note that the pressure-coefficient distribution predicted by the panel method is similar to that from wind tunnel data, with the exception of the trailing-edge region. There, the combined effects of the potential flow assumption and a trailing-edge Kutta condition (described shortly) force the flow to remain attached to the blade until it reaches an aft stagnation point. This causes the surface pressure-coefficient to drop to unrealistic values as the flow accelerates around part of the highly curved trailing edge, then to rise abruptly to unity at the aft stagnation point. A real subsonic flow does not exhibit such a pressure drop or recovery as the flow leaves the trailing edge. For the present work, panel method results for the aft 4.5% of the axial chord were ignored to avoid highly unrealistic predictions of pressure coefficient.

Figure 5.2 shows a comparison of the pressure-coefficient distribution predicted by the panel method with that obtained from wind tunnel data (Amrud, 1985) for a forward-loaded low-speed cascade. This cascade, designated “LS1” at Carleton University, is the tip section from an actual low-pressure turbine and was used at Carleton to study the tip-leakage flow in turbines. The cascade was modeled using 29 blades, each with the same distribution of 56 panels. As in the case of LS3, the pressure-coefficients in the trailing-edge region are unrealistic. Note how the panel method solution shows a sharp change in pressure gradient near the peak suction region. This is a result of using a small number of panels to resolve the blade contour. A higher concentration of panels in the peak suction region would result in a smoother pressure-coefficient distribution.

Figure 5.1: Panel Method Prediction for an Aft-Loaded Low-Speed Cascade

Figure 5.2: Panel Method Prediction for a Forward-Loaded Low-Speed Cascade

The Kutta condition is used to give the potential flow problem a unique solution and is achieved by specifying the rear stagnation point. Panel method analyses for airfoils and turbomachinery blades frequently use a cusped trailing edge and locate the rear stagnation point at the sharp cusp. However, for turbine blades the cusped trailing edge results in considerable geometric distortion since these blades often have fairly thick trailing edges. In the present analysis, the true trailing-edge geometry is modeled. The panel method code places the stagnation point at the junction point between a specified pair of panels. The pressure distribution and outlet flow angle were found to be sensitive to the exact location of the specified stagnation point. For this reason, the first estimate of the stagnation point for the analysis of a given cascade was sometimes not satisfactory. Small changes to the stagnation point location were made by distorting the panels in the trailing-edge region. An iterative scheme was used to find a good stagnation point location. The stagnation point was considered to be reasonably well placed when the flow just off the suction and pressure surfaces met with equal velocities a short distance downstream of the trailing-edge. This condition was deemed to exist when the velocity difference between two suitably-chosen test points was less than 0.1% of the average velocity at the test points. The locations of the test points are shown in Figure 5.3. The test points, indicated by squares, are located at equal distances from the radial line of the trailing-edge circle that extends at the blade metal angle, β_2 . The method by which the test points were located is not critical because the design method does not depend heavily on the accuracy of the panel method results. However, the test point location must vary smoothly with changes in cascade geometry so that the panel method can correctly reveal trends in pressure-coefficient distribution with changes in geometry.

Figure 5.3: Location of the Test Points used to Verify the Aft Stagnation Point Location

5.2.2 Panel Generation

The accuracy of a panel method solution for a cascade having blades with blunt trailing edges is highly dependent on how the panels are distributed. The ratio of the length of a panel to that of a neighbouring panel should be as close to unity as possible (e.g. a practical maximum allowable ratio is 1.3:1). On the other hand, there is a need for clustering in regions of high curvature in order to minimize the total number of panels since computational time increases more than proportionally to the number of panels used.

As mentioned in the previous section, the accuracy of the panel method is of limited importance, while the ability of the code to show trends in pressure distribution with changes in cascade geometry is of great importance. It is therefore important that no large change in the distribution of panels should occur for any given small change in geometry. Additionally, the distribution of panels around a blade must vary smoothly with the location of the aft stagnation point (the junction point between a specified pair of panels) so that the flow field (hence the pressure-coefficient distribution) varies smoothly as a function of stagnation point location. This smooth trend is a requirement of the algorithm responsible for locating the aft stagnation point.

In the present work, the panel distribution was defined by the junction points between neighbouring panels. The process of setting the junction point positions begins by selecting two key points on the blade profile: the forward-most point (measured in the axial direction) and the point that lies on the radial of the trailing-edge circle extending at the blade outlet metal angle. For purposes of this work, these key points divide the blade profile into two sections: the “suction surface” and the “pressure surface”. A set of geometric series are used to define a finite array of first increasing and then decreasing numbers, which are then uniformly scaled so that the sum of

the array elements is equal to the suction surface length. Points are then mapped to the blade's suction surface, starting at the aft key point and ending at the forward key point, using the array elements to define surface distance intervals. The array is then re-scaled and the process is repeated for the pressure surface of the blade. The result is a preliminary panel distribution. Figure 5.4 shows the preliminary distribution of panel junction points (the upper blade profile apparent in the figure) for a typical cascade blade. For this distribution containing 56 panel junction points, the forward key point is point 29 and the aft key point (the stagnation point) is Point 1. The arrows in the figure point at panel junction points where the geometric ratio governing surface distance intervals (between the junction points) changes.

The aft key point is interpreted by the panel method code as the aft stagnation point. As mentioned in the previous section, it is important to be able to move the stagnation point so that a proper match of flow velocities from the suction and pressure sides is achieved just downstream of the trailing edge. It is also important not to adversely affect the smoothness of the preliminary panel distribution while moving the stagnation point. One obvious method of moving the stagnation point without adversely affecting the smoothness of the distribution would be to move every point by a small, equal distance along the surface of the blade profile, until the stagnation point is in the desired location. In order to save computational time, another method has been used whereby the vast majority of points are positioned independently of the choice of stagnation point location. This allows the reuse of many of the geometry-dependent quantities that are calculated during the first panel method evaluation, accelerating any additional panel method evaluations that may be required to find an acceptable stagnation point location. Whenever the preliminary panel distribution is modified by moving the aft key point (the stagnation point) from its original position, a small group of panels on each side of the aft key point is moved as well. The number of panels in each group is the same: one less than 50% (rounded down to the next

Figure 5.4: Formation of a Panel Distribution

lowest integer) of 31.25% (also rounded down to the next lowest integer) of the total number of panels. The number of panels in each distinct region of the blade profile (evident in Figure 5.4) was determined in a similar way: by using percentages of the total number of panels or of the number of panels in a larger region. The employed percentage values were judged based on the appearance of the resulting panel distribution. By moving neighbouring points along with the aft key point, a smooth variation of panel length around the blade profile is maintained. In Figure 5.4, the lower panel distribution has had its aft key point, the aft stagnation point (point number 1 in the figure), moved by a rotation of 20 degrees counterclockwise about the centre of the trailing-edge circle. Panel junction points 8 and 50 and all points forward of them have remained in their preliminary positions. Six points on each side of Point 1 have been moved along the blade surface such that their fractional distance to Point 1, measured from the nearest fixed boundary point, remains unchanged. For example, if the surface distance from the fixed boundary Point 8 to Point 3 is 80% of the distance from Point 8 to Point 1, then Point 3 will move along the surface 80% as far as point 1 moves.

The next section describes the Navier-Stokes flow solver that was used in the present work to obtain pressure-coefficient predictions of higher accuracy than those that can be obtained using the panel method.

5.3 Navier-Stokes Code

5.3.1 Description of the Navier-Stokes Code

A modified version of a Navier Stokes code by Dawes (Dawes, 1988) was used to obtain more accurate pressure distributions than could be obtained using the panel method. The Dawes

code solves the Reynolds-averaged Navier-Stokes equations on a structured, finite-volume H-mesh using time marching. The turbulence is modeled using the Baldwin & Lomax (1978) eddy viscosity/mixing length model. Several modifications to the code have been made by Benner (2000): a strictly two-dimensional version has been created; to allow incompressible flow calculations, the artificial compressibility model of Chorin (1967) has been implemented; and the use of blocked cells has been introduced to allow more flexibility in modeling complex geometries. For use in the present work, the code was run in 2-D mode with incompressible, viscous, turbulent flow using blocked cells to represent the blade geometry.

A variety of grid densities, with up to more than 200 cells in both the axial and blade-to-blade directions, have been tried in the present work. High grid densities were used wherever possible, so that boundary layer displacement effects would modify the surface pressure distribution as they would for a real cascade flow. A typical high-density grid would have 30 cell columns upstream of the blade, 100 cell columns along the blade, 100 cell columns downstream of the blade, 30 cell rows within a blade and 110 cell rows between blades (overall dimensions: 230 cells (axially) by 140 cells (pitchwise)). The layer of cells next to the blade surface typically has a thickness of 0.01% to 0.1% of the axial chord.

High grid densities are tolerable for two reasons. The first reason is that the Navier-Stokes code is run only a few times during the design process. Usually 3 or 4 iterations of the overall method, each accompanied by a Navier-Stokes evaluation, are adequate. The second reason why high grid densities are tolerable is that the Navier-Stokes solution for the design at the end of one design iteration may be used to initialize the solution for the design at the end of the next iteration, thus reducing the number of internal iterations needed for solution convergence. This is practical whenever the change in geometry between iterations is small, which is usually

the case. In order to reduce computational time for the first Navier-Stokes evaluation in the design process, the velocity and pressure fields may be initialized with panel method results.

It was found that if the grid around the trailing edge was too fine, the convergence of the time-marching solution was sometimes hampered by an apparent tendency for the location of the trailing-edge stagnation points to fluctuate in time. This is most likely due to the inherently unsteady nature of the trailing-edge flow, including the shedding of vortices, which sometimes exist in physical reality and can appear in time-resolved computations for turbine blades (e.g. Currie & Carscallen, 1998). Figure 5.5 (a) shows the trailing-edge region of a grid for which the density of cells is higher than that normally used in this work: there are 66 cell rows within the blade thickness. The trailing edge of the blade occupies the upper-left region of the figure. The suction surface is above the pressure surface. The direction of flow is generally down and right. High clustering along the sides of the blade helps to capture boundary layer development. The kinks in the clustered regions are formed where the highly-clustered cells leave the blade surface. A flow solution for this high-density grid exhibited unsteady flow in the trailing-edge region and an unsteady pressure coefficient distribution around the blade. For the present inverse-design purposes it appears that it may be preferable to use a lower cell density in the trailing-edge region. Meshes with a reduced density downstream of the trailing-edge are less able to resolve the details of the trailing edge flow while still giving good pressure predictions around the rest of the blade. A steady and much more converged flow solution was obtained when the density of cells behind the trailing edge was reduced from 66 to 20 cells within the blade thickness, as shown in Figure 5.5 (b).

The next section describes the grid generator that was used to produce CFD grids for use in the present work.

Figure 5.5(a): CFD Grid in Trailing-Edge Region: Dense Grid

Figure 5.5(b): CFD Grid in Trailing-Edge Region: Sparse Grid

5.3.2 Description of the Grid Generator

The purpose of this section is to describe briefly the grid generator that was developed as part of the present design package. Additional information concerning the grid generator and its use can be found in Appendix B.

The grid generator produces a structured H-mesh that uses blocked cells to define a single blade of a cascade. Figure 5.6 shows a representative example of a mesh (every other node is removed for clarity). In order to construct a mesh, the grid generator requires the cascade geometry and several settings provided by the user. The grid settings control the number of nodes, the extent of the grid in the streamwise direction, as well as numerous clustering parameters. They also affect the shape of the grid, affecting the orthogonality and skewness of the cells. At the first application of the Navier-Stokes code (ie. at the end of the first global iteration), the user defines a grid using default settings. The mesh is then inspected to make sure that the skewness and aspect ratios of the cells are acceptable. For example, in the grid shown in Figure 5.6, the orientation of the grid downstream of the trailing edge was adjusted to reduce skewness of the cells in the trailing-edge region (by default, the gridlines downstream of the blade will be set at the outlet metal angle). For subsequent global iterations of the design method, the grid parameters require few, if any, further adjustments.

Repeating boundary conditions are used by the Navier-Stokes solver to simulate an infinite cascade. For this reason, the top and bottom edges of the grid must have the exact same shape and node distribution. This was accomplished by forming the grid on one side of a blade, then moving part of the grid to the opposite side of the blade. This type of grid avoids having a

Figure 5.6: Representative Grid for Navier-Stokes Calculations

repeating boundary condition along a fluid/blade surface interface: a boundary condition not allowed by the employed Navier-Stokes solver.

The grid is constructed in three basic steps: boundaries are established, dividing the grid into 11 sub-regions; nodes are placed along these boundaries according to clustering parameters; and the sub-regions are filled with nodes by using various interpolation methods.

Figure 5.7 shows how the grid is divided into 11 sub-regions. The cut line shown in the figure is the middle row of interpolated nodes between blades. It is used to cut four large regions into eight smaller regions, as apparent in the figure. For example, Regions 7 and 1 were formed from a single region, as were Regions 2a and 8a. The four smaller regions above the cut line (namely Regions 1, 2a, 2b and 3) are then translated vertically downward by an amount equal to the blade spacing so that, in the final grid, the blade has regions of fluid on both sides of it. In Figure 5.7, the regions of the final grid are those that are labeled. The borders of the sub regions, other than those that form part of the blade profile or the cut line, are a mix of Bezier-cubic polynomials and straight line segments.

Nodes were clustered along the boundaries primarily by using geometric ratios to define distance intervals. Nodes were interpolated using linear interpolation in the regions upstream and downstream of the blade. The method of interpolation for nodes within the blade profile is unimportant because the cells in that region are blocked. The nodes in the four regions between blades (2a, 2b, 8a, 8b in Figure 5.7) are positioned using two steps:

Step 1: An initial node location is found by a simple form of interpolation.

Figure 5.7: Regions of the Grid

Step 2: A displacement vector, governed by the shapes of the boundaries and the initial node location, is then applied to the initial node location.

This chapter has reviewed the flow analysis tools used to evaluate low-speed cascade designs in the present work. As described in Chapter 3, the panel method is used to perform the majority of cascade analyses while the Navier-Stokes solver is used to analyze the resulting design at the end of each global design iteration. The next chapter describes the optimization algorithms that make use of the panel method to rapidly evaluate a large number of trial designs in the process of design optimization.

Chapter 6: The Objective Function and Tools for Design Optimization

6.1 Introduction

The present design method employs an optimization strategy that involves an objective function. The objective function serves as a means of evaluating candidate cascade designs with respect to the design objectives and the design constraints. The input to the function is a set of geometric parameters that represent a cascade design. The output is a single number that indicates how well the design objectives have been met.

An evaluation of the objective function for a given cascade begins by constructing the cascade's geometry according to a set of geometric parameters. This is accomplished by using the method of geometric construction outlined in Chapter 4. The next step is to predict the pressure-coefficient distribution around a blade of the cascade using the panel method solver (see Section 5.2). The predicted pressure-coefficient distribution is then further processed to yield a value of the objective function corresponding to the given cascade. The next section describes the objective function in detail.

In the present work, optimization tools are used to search within a design space for the optimal cascade design: that with the highest objective function value. The optimization tools are described in sections 6.3 and 6.4.

6.2 The Objective Function

The basic form of the objective function was defined, by trial and error, as follows:

$$f(g_1, g_2, \dots, g_M) = \max \left[K - \sqrt{\frac{\sum_{i=1}^N (\Delta C_p(g_1, g_2, \dots, g_M, i))^2}{N}}, 0 \right] \quad (6.1)$$

where: f is the dependent variable of the function; ΔC_p is the difference between the i^{th} target value and the corresponding predicted value of C_p on the same side of the blade at the same non-dimensional axial coordinate; $g_1, g_2, \dots, g_M$ are the M parameters that define the cascade geometry; N is the number of C_p values of the specified target distribution that are within the forward 95.5% of the cascade, measured axially; and K is a constant set to 3. In order to calculate ΔC_p , linear interpolation is used so that the predicted value of C_p will have the same non-dimensional axial coordinate as the corresponding target value. Target C_p values in the aft 4.5% of the cascade (measured axially) are ignored since the corresponding predicted values of C_p , derived from panel method results, are ignored in this region of the distribution (see Section 5.2.1).

Since the comparison of pressure distributions requires a negligible amount of computational time compared to the other computations involved in the design method, it is recommended that a uniform density of target points be used such that the details of the variation of the target pressure distribution are well resolved. In order to increase the influence of a particular region of the target distribution, more target points can be added to that region. In the

present work, N was typically no less than 50 so that the target distribution would be reasonably well defined.

To help avoid obtaining a cascade geometry that cannot be built for use in a low-speed wind tunnel, the present method enforces a minimum thickness requirement: the thickness of a blade (measured in the tangential direction) must be greater than 3% of the axial chord from $x/b_x = 0.15$ to $x/b_x = 0.85$ and must not be less than zero over the remaining parts of the blade. Any candidate cascade that does not meet this requirement is assigned an objective function value of zero instead of the value that would be given by Eqn. 6.1. This thickness requirement, although simplistic, has worked well for the purposes of this work. A more sophisticated thickness requirement might be applied before using the method to produce designs that will actually be built for testing. Such a requirement could be based on considerations such as structural rigidity and the allowance of space for instrumentation within blades.

The objective function (Eqn. 6.1) was used directly by the genetic algorithm as a fitness function. Accordingly, the value of the objective function was made to increase to a maximum, rather than to decrease to a minimum, for the optimal cascade design. Because the employed genetic algorithm requires non-negative values of fitness when roulette wheel selection is employed, the objective function was formulated so that its dependent variable is always non-negative. The constant K in the objective function was selected considering two processes of the genetic algorithm. These processes, discussed next, are (a) the random generation of individuals for the initial population and (b) fitness scaling:

(a) The Random Generation of Individuals for the Initial Population

During formation of the initial population, individuals (cascade candidate designs) are generated randomly, evaluated, and accepted or not into the initial population depending on whether or not they have a fitness greater than zero. The process of forming members and evaluating them continues until enough members of sufficient fitness have been found to form the initial population. To help save time, the rejection of a large proportion of the randomly-generated individuals was avoided by setting the constant K appearing in the objective function (Eqn. 6.1) sufficiently high.

(b) Fitness Scaling

Linear fitness scaling was used in this work. Fitness scaling was described in Section 2.4.2. During the scaling process, the fitness value of each member of a population is adjusted by performing a linear stretch of the fitness values with respect to the population average. The average fitness of a population will therefore remain constant before and after the scaling process. The linear stretch factor is set to a value that causes the scaled maximum fitness of a population to be equal to the population's average fitness multiplied by a specified fitness scaling parameter. As mentioned in Section 2.4.2, it is desirable for fitness scaling to bring the fitness values of the members of a population closer to the population average in early generations ("compression") and farther from the population average in later generations ("expansion"). This requires that the non-scaled maximum-to-average fitness ratio of a population is larger than the fitness scaling parameter in early generations and smaller in later generations. It is possible to meet this requirement with a constant value of the fitness scaling parameter because the maximum-to-average fitness ratio tends to decrease with successive generations. For example Figure 6.1(a) shows the average, maximum and scaled maximum fitness values for a typical GA run at every tenth generation. The fitness scaling parameter in this case was set to 1.2 (see Eqn. 2.3 with

(a) $K = 3$ (b) $K = 5$

Figure 6.1: Effect of Constant in Objective Function

reference to Eqn. 2.1). The graph shows “compression” for the first twenty to thirty generations, and “expansion” thereafter. The trend in the non-scaled maximum-to-average fitness ratio with generation is influenced by constant K in the objective function. For example, if K is increased by 2 units (i.e. from 3 to 5), all fitness values of a given population would increase by 2 units, lowering the population's maximum-to-average fitness ratio toward unity for all generations. This change would cause the transition from “compression” to “expansion” (i.e. the generation at which the scaled maximum fitness becomes larger than the maximum fitness) to occur at an earlier generation for the same value of the fitness scaling parameter. Using the same data as for the left-hand graph in Figure 6.1, the constant K was changed to “5” and the results were plotted on the right-hand graph in Figure 6.1. Although the run was not actually conducted with K set to “5”, this change serves to illustrate a point: for the exact same individuals in each population, the transition from “compression” to “expansion” occurs at an earlier generation: in this case, within the first ten generations. This would be expected to reduce performance in early generations since there would be a reduction in the number of generations for which “compression” would help to prevent a few relatively fit individuals from dominating. A reduction in the fitness scaling parameter below 1.2 would increase the number of generations for which the fitness values are “compressed” but would reduce the effectiveness of roulette wheel selection, since the probability of selection would approach the same level for all members.

This section has presented the objective function that was used in the present work for the inverse aerodynamic design of turbine cascades based on pressure-coefficient distribution. The next two sections discuss the optimization codes that were used for finding the global optimum (maximum) of this function within a given design space.

6.3 The Employed Genetic Algorithm: GAFDO

6.3.1 Description of GAFDO

A general description of genetic algorithms was presented in Section 2.4.2. This section describes features of the genetic algorithm used in this work: GAFDO (Genetic Algorithm For Design Optimization). The step-by-step details of the algorithm are not included in this thesis text. The interested person may obtain a copy of the computer code, which is written in the Fortran 77 standard. Many comments are included in the code to aid in understanding it. A user's manual for the software is included in Appendix B.

The order of topics in this section is as follows: (a) the method of encoding genes, (b) design parameter limits, (c) population sizes, (d) the method of selection, (e) the genetic operators: crossover and mutation, (f) elitism and (g) the use of a two-part fitness function. Section 6.3.2 documents the parametric studies that were conducted to set some of the controlling parameters of the genetic algorithm.

(a) The Method of Encoding Genes

Cascade design parameters are coded into a binary chromosome by using one gene for each parameter. One chromosome is used to contain all of the genes of an individual. The genes always lie in the same order along the chromosome. A parameter is encoded in three steps:

Step 1. The parameter is normalized to a number between zero and one by subtracting the lower parameter limit, then dividing by the difference between the limits.

Step 2. The normalized value is then multiplied by $2^n - 1$, where n is the number of binary digits in the gene.

Step 3 The resulting number (less any fractional component) is converted to binary format.

For example: if the parameter limits are 4 and 10 and the parameter value is 6.5, then, for a 6-digit gene, the parameter would be encoded as follows:

Step 1. $(6.5 - 4) / (10 - 4) = 0.416666$

Step 2. $(0.416666)(2^6 - 1) = 26.25$ (This is rounded down to 26 for the next step.)

Step 3. $26 \text{ (base 10)} = 011010$ (This is the binary form of the gene.)

The number of binary digits per gene was set high so that the design space would be finely resolved. A typical gene contains from 8 to 10 binary digits, allowing the representation of 256 to 1024 discrete values.

(b) Design Parameter Limits

At the beginning of a GA run, the parameter limits are set to encompass a wide range of cascade shapes so that the cascade design of interest will likely be included in the design space. Periodically during the run, the parameter limits are modified in order to exclude the least promising regions of the design space. This strategy, described more thoroughly in Section 6.3.2.2, has resulted in accelerating the rate of progress of the GA compared to a run that does not involve narrowing the design space.

(c) Population Sizes

In order to begin a genetic algorithm run with a diverse gene pool, the initial population is formed by randomly generating individuals, keeping only those individuals that meet a minimum-fitness requirement. A typical initial population size for the present application is 20 to 50 individuals (see Table 6.1). To reduce computational time, the size of the continuing populations is set smaller than the initial population. Section 6.3.2.3 provides experimental evidence that the continuing population size has little effect on the rate of progress of the GA with generation. This means that the time required by the GA to attain a certain level of fitness is approximately proportional to the continuing population size, ignoring the time required to evaluate the initial population. The continuing population size should therefore be as small as possible without causing adverse effects on performance, such as reduced reliability in locating the vicinity of the global optimum. After the members of the initial generation have been formed, the population size is typically reduced to between 10 and 20 individuals while retaining the individuals of highest fitness (see Table 6.1). Section 6.3.2.3 discusses how reasonable values of the initial population size and continuing population size were determined.

An alternative strategy for reducing the population size is to allow a generation to be created before reducing the size of the population. That way, the best genes from the initial population are given a chance to collect into a smaller number of individuals so that many are not lost when the population size is reduced. Preliminary tests have been conducted to determine whether this alternative strategy would yield improved performance, but no noticeable changes in GA performance versus generation were observed.

Name of Controlling Parameter	Standard Value	Suggested Range or Setting
Initial Population Size	40	20 to 500
Population Size for third and subsequent generations	20	14 to 300
Last Generation	750	400 to 2000
Probability of Crossover	12%	8% to 20%
Probability of Mutation	0.9%	0.4% to 1.2%
Fitness Scaling	on	on
Fitness Scaling Factor	1.2	1.2 to 2.0
Selection Method	Roulette Wheel	Roulette Wheel
Elitism	on	on
Multiple A (non-zero value for design-space narrowing)	20	5 to 30
Multiple B	0 (off)	0 (off) to 10 times Multiple A
Fraction of the population to keep during design-space narrowing	80%	60% to 90%

Table 6.1: Standard and Suggested Settings for GAFDO for the Cascade Inverse Design Problem

(d) The Method of Selection

Selection of parents is accomplished by either roulette wheel selection or tournament selection. In the present optimization problem, both methods seem to produce comparable levels of GA performance provided that roulette wheel selection is accompanied by proper fitness scaling. Section 6.3.2.4 compares the experimental results obtained with each method.

(e) The Genetic Operators: Crossover and Mutation

In addition to selection, the genetic operators of crossover and mutation are used. Child chromosomes are formed in pairs: one based on one parent and the other based on the other parent. A copy of each of the parental chromosomes (one per parent in this case) forms a tentative child chromosome. Crossover is then used to exchange some genetic information between these tentative child chromosomes so that each child contains genetic information from

both parents. Mutation is then applied to complete the formation of each of the two child chromosomes.

Each gene along the chromosome has one potential crossover location. The probability that crossover will be applied to a given gene is a specified constant. If a gene is chosen for crossover, the crossover point is selected at random without bias.

Every digit in every chromosome has the same probability of being mutated from its original, parental, value. Mutations are implemented by toggling the value of a bit between its two possible values: zero and one.

Reasonable values for the probabilities of crossover and mutation were determined experimentally, as outlined in Section 6.3.2.5.

(f) Elitism

Elitism was implemented by forming the next generation then replacing one of the members (the first one formed) by the most-fit individual of the previous generation. Elitism was implemented to help avoid an accidental loss of all of the highly fit individuals from one generation to the next.

(g) The Use of a Two-Part Fitness Function

To reduce computational time, the fitness function is run using the panel method with a relatively small number of blades in cascade (15) in the first 200 generations (from 0 to 199) and a larger number (29) in generations thereafter. This change to the number of blades used for representation of a cascade causes a change in the fitness function, slightly altering the optimal

cascade geometry. If the genetic algorithm is allowed to run well beyond 200 generations, it will approach the same design as if the larger number of blades had been used for all fitness evaluations. It does not matter if the genetic algorithm is not allowed to run long enough to converge on the optimum using the larger number of blades in cascade because the gradient-based optimization method is used to complete the optimization process. The gradient-based optimizer used in the present work is described in Section 6.4.

This section has reviewed the main features of GAFDO that collectively define it apart from other genetic algorithms in general. The next section discusses how most of the controlling parameters of the GA were determined.

6.3.2 Customization of GAFDO

6.3.2.1 General Remarks

In order to set the controlling parameters of the GA (such as probabilities of crossover and mutation, population sizes, etc.) several parametric studies were conducted. These studies were conducted because, without correctly setting the controlling parameters, the performance of the GA would be adversely affected. From the earliest usage of GAFDO to the most recent usage, the time taken by the code to solve the same problem has decreased by more than one order of magnitude through improvement of the choice of parameters.

Of course, many choices of the settings that control the GA are feasible and work reasonably well. In fact, some of the settings have optimal values that are partially a function of the fitness function itself, which changes with each new design problem. In order to reduce the

computational time by several orders of magnitude, thereby making parametric studies practical, the fitness function was altered to become much less computationally expensive. Instead of using a comparison of C_p distributions to calculate the fitness of a cascade, a direct geometric comparison was made between a single blade of the candidate cascade and a single blade of the target cascade, with the axially-furthest point of each blade at the same location. It was hoped that the objective function based on geometry would be similar enough to that based on pressure coefficient that the parametric studies would yield results applicable to the best parameter settings for the intended use of the GA. Using tests involving the geometry-based objective function, a satisfactory set of GA-controlling settings has been found and acceptable ranges for some of the settings have been identified. These settings and suggested ranges are listed in Table 6.1 (see Section 6.3.1).

The fitness function used for geometric comparison is as follows:

$$f(g_1, g_2, \dots, g_M) = \left[K - \sqrt[4]{\frac{\sum_{i=1}^N (\Delta y / b_x(g_1, g_2, \dots, g_M, i))^2}{N}}, 0 \right] \quad (6.2)$$

where: f is the dependent variable of the function; $\Delta y / b_x$ is the normalized y-coordinate difference between the i^{th} defining point of the target blade profile and the point of the candidate blade profile on the same side of the blade at the same non-dimensional axial coordinate; $g_1, g_2, \dots, g_M$ are the M parameters that define the candidate cascade geometry; N is the number of defining points given for the specified target blade profile; and K is a constant set to 3.

The use of the fourth root, rather than the square root, was intended to amplify fitness differences for highly fit designs. The terms of the sum apply to all of the given points of the

target geometry. When necessary, linear interpolation was used to find a point on the design having the same axial coordinate as the corresponding point on the target geometry.

The remaining parts of Section 6.3.2 discuss how appropriate settings for the GA-controlling parameters were determined. Section 6.3.2.2 discusses how certain parameters were set to control the process of narrowing the design space. Section 6.3.2.3 discusses the selection of population sizes. Section 6.3.2.4 discusses the issue of whether roulette wheel selection or tournament selection should be used. Section 6.3.2.5 discusses how the probabilities of crossover and mutation were set.

6.3.2.2 Narrowing the Design Space

The concept of narrowing the design space was introduced in Section 2.4.2. The parameters that were used in the present work for controlling the narrowing process are (a) “Multiple A”, (b) “Multiple B”, (c) the percentage of the population retained during design-space narrowing, and (d) the minimum-range constants. These parameters are described next:

(a) “Multiple A”

“Multiple A” (an informally named variable) is a controlling parameter of the GA used to set the generation interval between successive adjustments to the borders of the design space. More specifically, the design-space narrowing process is applied only to generations that are a multiple of “Multiple A”. Experience with the design method indicates that a value of 5 for “Multiple A” works well for small populations and a value of 20 works well for a wide range of population sizes. If “Multiple A” becomes too large (more than 40), the design-space will not be sufficiently narrowed and the rate of progress of the GA will be diminished. The objective is to

let the evolution proceed for enough generations so that the population has had time to respond to the changes in the extent of the design space. A large population (300) may require more generations to respond than a small population (20) because the “good” genes of the individuals require several generations to be brought together, by probability, into common individuals.

(b) “Multiple B”

“Multiple B” is of secondary importance to the operation of the GA. It is a number, the multiples of which indicate the generations at which the design space is fully widened to the original size. This feature was added as a mechanism to prevent good regions of the design space from being excluded from the search, as often happens when the design space is narrowed. The exclusion of good regions of the design space is more likely to occur when the population size is very small (e.g. 10). Unfortunately, it is also when the population size is small that the GA is less robust and will poorly explore regions that have been made available by periodic widening of the search space. “Multiple B” is typically set to zero to disable this feature (see Table 6.1).

(c) The Percentage of the Population Retained During Design-Space Narrowing

As mentioned in Section 2.4.2, narrowing the design space generally involves the loss of a portion of the population since not every member is guaranteed to be included in the updated design space. In the present work, the design space is narrowed about the fittest individuals in the population. The group of fittest individuals is determined by selecting the top $n\%$ of the population, where n is a user-defined amount. The extent of each dimension of the design space is made slightly greater than that required to exactly hold the fittest individuals. The remaining individuals are eliminated regardless of whether they could be represented in the new space. They are replaced by offspring from the individuals that were preserved. The population formed

by the combination of the preserved members and their offspring is considered to constitute a new generation: the first generation to be defined in the newly-formed design space.

(d) The Minimum-Range Constants

The minimum-range constants are a reference to multiplication factors used during design-space narrowing. The appropriate values for these constants are affected by the fitness function and the original design parameter limits. Before explaining the role of these constants, the method of adjusting the parameter limits during the narrowing process must be understood, and is described next.

The range of each design parameter is calculated at the start of a GA run by subtracting the specified minimum limit from the specified maximum limit. These values are stored and not altered for the rest of the run. When the design space is narrowed, the range of each parameter is set to be 10% wider than that occupied by the fittest $n\%$ of the population, where n is specified. This means that, strictly speaking, the “narrowing process” may actually result in expanding the design space, or at least in some of its dimensions, but not beyond the initial size. The “narrowing process” is referred to as such because, after the process has been applied several times, the design-space usually tends to be smaller than its initial size. When the limits of a particular design parameter are expanded, the expansion occurs equally in both directions, stopping at an original limit if one (or both) is (are) reached. The expansion helps to prevent possible good regions of the design space from being removed. Despite this precaution, it has been found that optimal values of the design parameters are often beyond the extent of the newly

From Generation	To Generation	Minimum-Range Constant: Percentage of Original Range Size
0	99	70
100	199	50
200	299	30
300	399	20
400	last	10

Table 6.2: Schedule for Reducing the Minimum-Range Constants

formed borders of the design space. For this reason, a mechanism to fully widen the search space (on a schedule controlled by “Multiple B”) was added (see part (b) above). In practice, however, satisfactory results have been obtained without ever widening the search space fully.

With an understanding of how the parameters are adjusted while the design-space is narrowed, the role of the minimum-range constants may be explained. These constants multiply the original range of each parameter (stored at the beginning of a GA run) to determine the minimum allowable range for each parameter. The constants are set so that greater narrowing is allowed for parameters where the initial range is very large and near-optimal values are expected to appear in early generations (i.e. so that as much narrowing as possible can occur while making it unlikely that good regions of the design space are removed). According to a schedule based on generation number (see Table 6.2), the minimum-range constants are reduced. This allows the range of each parameter to become progressively smaller. If any given parameter has a sufficiently wide spread in values among the top members of a population, its new range will be larger than the minimum allowed.

The next section discusses how the initial and continuing population sizes were selected. The influence of population size on the effectiveness of narrowing the design space is also discussed.

6.3.2.3 Population Sizes

Because design-space narrowing and population size both affect the rate and extent of fitness gain with generation, this section will explore the effects of each on test cases. Using the geometry-based fitness function (as for all of the other parametric studies), the tests listed in Table 6.3 were conducted 10 times each, and the fitness results were algebraically averaged.

The effect of parameter space narrowing on the performance of the GA is apparent for small population sizes. Tests 1 and 2 have a large continuing population size of 300 and show no appreciable difference in performance even though one of the cases makes use of design-space narrowing. Test pairs 4, 5 and 7, 8 have smaller continuing population sizes and show an improvement in performance resulting from the use of design-space narrowing (although the amount of improvement for test pair 4, 5 is not substantial). A likely reason for the stronger effect on smaller population sizes (than 300) is that the smaller spread in parameter values of the fewer individuals that must be retained during design-space narrowing leads to a greater amount of narrowing.

Test Name	Initial Population Size.	Continuing Population Size	Design-Space Narrowing	Best Fitness in the Initial Population	Best Fitness by Generation 50	Best Fitness by Generation 100	Best Fitness by Generation 500	Best Fitness by Generation 1000
Test 1	1000	300	no	2.7899	2.8984	2.9121	2.9308	2.9361
Test 2	1000	300	yes*	2.7915	2.8909	2.9072	2.9330	2.9384
Test 3	1000	10	yes*	2.7886	2.8684	2.8972	2.9374	2.9481
Test 4	1000	20*	yes*	2.7940	2.8827	2.8981	2.9353	2.9462
Test 5	1000	20*	no	2.8085	2.8940	2.9094	2.9302	2.9349
Test 6	1000	40	yes*	2.7964	2.8942	2.9152	2.9483	2.9556
Test 7	1000	80	yes*	2.7858	2.8921	2.9118	2.9478	2.9572
Test 8	1000	80	no	2.7894	2.8836	2.9012	2.9224	2.9264
Test 9	1000	160	yes*	2.8070	2.8835	2.9057	2.9409	2.9496
Test 10	100	20*	yes*	2.7276	2.8684	2.8926	2.9367	2.9446
Test 11	40*	20*	yes*	2.7025	2.8637	2.8860	2.9311	2.9401
Note: Standard settings were used (see Table 6.1) unless otherwise indicated.								
* standard								

Table 6.3: Parametric Study of Population Sizes and Design-Space Narrowing

The initial population sizes of tests 11, 10 and 4 are 40, 100 and 1000 respectively, with all other parameters set to standard values. They all exhibit roughly the same level of performance, indicating that an initial population size of 40 is just as good as an initial population size of 100 or 1000. It would be unwise to try to save time by reducing the initial population further since little time would be saved and because the progress of a run depends to some degree on an initial population that adequately covers the design space. A simple test was conducted involving the standard GA settings on a two-parameter fitness function having 6 local optima of roughly the same magnitude. It was discovered that the GA would correctly find the overall optimum when the initial population size was 50 or larger, but would sometimes miss it when the initial population size was reduced to 20. Tests involving cascade design performed with an

initial population size of 20 have always worked well but it was decided that a value of 40 should be used as a standard to ensure reliability.

The continuing population sizes were studied using test cases 3, 4, 6, 7, 9 and 2, which had, respectively, continuing populations of 10, 20, 40, 80, 160 and 300 members. The fitness reached by generation 1000 was highest in test 7, with a continuing population size of 80. Considering that all of the tested population sizes produced results good enough for use of the gradient-based optimizer, and that computational time varies almost directly with the continuing-population size, a population size of 10 seems to be the best setting. A continuing-population size of 20 was selected as standard before these results were available and has been used to conduct the test cases presented in Chapter 7. Smaller continuing-population sizes than 10 have not been tested.

6.3.2.4 Roulette Selection vs. Tournament Selection

A parametric study was conducted to compare the relative performance of the GA when using roulette selection versus that when using tournament selection. Population sizes were varied to see if the type of selection that works best is governed by population size. The other settings were set to standard values (see Table 6.1). Ten trials were performed for each test case and the results were averaged across the trials. The results are summarized in Table 6.4. The performance of roulette wheel selection may increase depending on the value of the fitness scaling parameter used.

Test Name	Initial Population Size.	Continuing Population Size	Type of Selection: Roulette (R) or Tournament (T)	Best Fitness in the Initial Population	Best Fitness by Generation 50	Best Fitness by Generation 100	Best Fitness by Generation 500	Best Fitness by Generation 1000
Test 11	40*	20*	R*	2.7025	2.8637	2.8860	2.9311	2.9401
Test 12	40*	20*	T	2.6411	2.8432	2.8810	2.9450	2.9585
Test 13	40*	10	R*	2.6725	2.8189	2.8606	2.9244	2.9387
Test 14	40*	10	T	2.7074	2.8432	2.8764	2.9328	2.9439
Test 7	1000	80	R*	2.7858	2.8921	2.9118	2.9478	2.9572
Test 15	1000	80	T	2.7955	2.9168	2.9348	2.9616	2.9682
Note: Standard settings were used (see Table 6.1) unless otherwise indicated.								
* standard								

Table 6.4: Parametric Study of Roulette Selection vs. Tournament Selection

Tournament selection outperformed roulette wheel selection in all of the cases presented in Table 6.4. In particular, the standard settings for GAFDO (see Table 6.1), represented by Test 11 were outperformed when only the method of selection was changed. The standard settings (listed in Table 6.1) were set before tournament selection had been explored and consequently do not reflect the best known settings. The goal of the genetic algorithm is to find a nearly optimal design so that the gradient-based optimizer has a good starting point in the design space. All of the tests presented in Table 6.4 meet this requirement.

6.3.2.5 Probability of Crossover and Mutation

A parametric study was conducted early during code development to determine appropriate values of the probabilities of crossover and mutation. The fitness function used an earlier form of the cascade geometry in which a blade profile had 2 Bezier cubic polynomials for the suction surface and one for the pressure surface. As for the other parametric studies, geometric comparison was used for the fitness function rather than pressure-distribution comparison. The population sizes were set conservatively at 1000 for the initial population and 300 for continuing populations. Design-space narrowing was not performed.

The study covered probabilities of crossover from 0.05 to 0.70 and probabilities of mutation from 0.001 to 0.050. Figure 6.2 shows the results graphically. The gain in best fitness over the first 100 generations was taken 10 times and averaged for each node in the rectangular mesh shown. A contour map was generated to help visualize the region of highest gain, indicated by the dark area. The optimum combination was found to be a probability of crossover of 12% and a probability of mutation of 0.9%. These values have been adopted as standard (see Table 6.1).

Figure 6.2: Results of the Parametric Study of the Probability of Mutation and the Probability of Crossover

6.3.3 Deciding When to Stop the Genetic Algorithm

Schemes for deciding when to stop the GA could be devised based on the number of generations processed without a substantial improvement in fitness (e.g. 250 generations without a 10% reduction of the difference between ideal fitness and the attained fitness), or on the progress of the average fitness of a population. It has been observed that if the population average fitness for the most recent generation increases significantly beyond the population average fitness for all earlier generations, another gain in fitness is likely to occur within the next few hundred generations. Conversely, if the population average drops to very low levels due to the propagation of “bad” genes (this happens more often with small population sizes), it will likely take many hundreds of generations to achieve the next increase in fitness. The computer

implementation of the present design method continuously outputs the following GA progress indicators that may be examined without interrupting a run: maximum unscaled fitness (both for the current population and for all completed generations), average fitness, minimum fitness. The user can use this information in conjunction with a set of criteria to decide when to stop a GA run.

It is interesting to note how, without narrowing the design space, the performance versus generation changes negligibly with continuing-population size. In Section 6.3.2.3, tests 5, 8 and 1 have continuing population sizes of 20, 80 and 300 respectively, yet all three cases show roughly the same level of fitness for a given number of generations. With design space narrowing, the number of generations processed is still a good indication of the level of fitness likely attained, provided that the GA settings are reasonably well chosen. Typically after 750 generations the fitness will be high enough to begin using the gradient-based optimizer. It is suggested that, for simplicity, the GA be run for 750 generations before switching over to the gradient-based optimizer (described next).

6.4 The Employed Gradient-based Optimization Method

Although the genetic algorithm is a highly robust search method, it is not always the most efficient means of reaching the optimal point in the design space once a point in the vicinity of the optimum is found. In order to improve the overall performance of the design method, a gradient-based optimization method (in this work, called “the GTOM”) is used to compliment the genetic algorithm. This tool optimizes design fitness by using gradient information to move step-by-step through the design space. This method should start from the best known design point, as provided by the GA, to help guard against “homing in” on a local optimum rather than the overall optimum.

The original version of the GTOM simply evaluated the gradient and followed it until a maximum fitness was reached, then repeated the process. The gradient vector is formed so that each of its component vectors (one for each of the design parameters) is proportional to the partial derivative of fitness with respect to the corresponding design parameter. The partial derivatives were approximated by using a central difference calculation involving a pair of test points surrounding the current design point. It was noted that, after the GTOM had stopped finding better designs by searching along the gradient, that the test points used in the central difference approximation would occasionally have record high fitness. The GTOM was modified to take advantage of this fact: after a test point with a record high fitness level was encountered, the design point would then be moved to that test point's location. Although the gradient calculation was less accurate due to the fact that the partial derivatives for a particular gradient calculation were obtained from sometimes different test points, a large improvement in overall GTOM performance was observed.

During development of the GTOM a problem with the performance was encountered: the design solution would often proceed a very small "distance" through the design space and then stop where no apparent local optimum existed. This problem may be due to numerical effects that affect the accuracy of fitness evaluations such as round-off error. To help counteract this problem, the test points used in the central difference approximation for partial gradients were selected at random distances from the central design point. If, on one particular pass through the gradient evaluation, a test point gave an inaccurate fitness value, the associated partial derivative calculation would be inaccurate. On the next pass through the gradient-evaluation stage of the algorithm, the same test point would likely not be chosen, and there would be a chance that the next test point would yield a more accurate value of fitness.

Chapter 7: Test Cases

7.1 Tests based on Low-Speed Cascades

7.1.1 General Remarks

In order to test the accuracy of the designs produced by the present design method, two different low-speed cascades were inversely designed from corresponding target pressure-coefficient distributions. In each case, the target distribution was derived from low-speed wind tunnel data obtained at Carleton University from a low-speed cascade model. Because compressibility effects are negligible for low-speed cascades, the inverse designs are expected to be almost identical to their corresponding original designs. To help determine if the cascade design method has problems associated with forward-loaded or aft-loaded blade designs, the two test cascades used to generate the target pressure-coefficient distributions were chosen so that both types of loading would be tested: one of the cascades, designated "LS1", has forward-loaded blades while the other cascade, designated "LS3", has aft-loaded blades. The tests show that the inverse designs are very similar to their originals, as expected. The most notable exception is the difference in blade spacing between cascade LS1 and its inverse design.

7.1.2 Test Case: Cascade LS1

The wind tunnel data for LS1 is taken from Amrud (1985). Table 7.1 summarizes the most relevant wind tunnel conditions.

Item	Description	Value
inlet flow angle	relative to axial	0.0 deg.
inlet flow velocity		30 m/s
Reynolds Number	based on axial chord and upstream reference	396 000

Table 7.1: Wind Tunnel Conditions for LS1 Wind Tunnel Data

The settings for the Navier-Stokes calculations used in the evaluation of the inverse design are listed in Table 7.2.

Item	Description	Value
inlet flow angle	relative to axial	0.0 deg.
inlet flow velocity		30 m/s
reference density	the density used for incompressible flow	1.199 kg/m ³
Reynolds Number	based on axial chord and upstream reference	396 000

Table 7.2: Settings for the Navier-Stokes Evaluations of the Inverse Design of LS1

Two global iterations of the design method were applied in order to design the inverse version of LS1. The pressure-coefficient distribution of LS1 and that of the inverse design match reasonably well, as shown in Figure 7.1. Recall (from Section 3.1, Equation 3.1) that the pressure coefficient is defined by:

$$C_p = (P - P_{\text{ref}}) / q_{\text{ref}} \quad (7.1)$$

where P is static pressure, q is dynamic pressure and the subscript “ref” indicates a value taken at the reference location: an upstream, undisturbed flow location.

The blade geometry given by Amrud (1985) was not normalized and was given for a staggered position at a different stagger angle than that corresponding to the wind tunnel data. The coordinates given by Yaras and Sjolander (1988) were more convenient to use because they were normalized and given for a stagger angle of zero. The geometry of the actual LS1 cascade was recreated by taking the blade geometry from Yaras and Sjolander (1988) and then adjusting the stagger angle according to Amrud (1985) so that it corresponds with the wind tunnel data.

Figure 7.2 shows the original cascade geometry overlaid with the inversely-designed geometry. As evident in Figure 7.2, there is a substantial difference in the ratio of blade spacing to axial chord between the original version and the inverse version. This could be an indication that the flow is relatively insensitive to blade spacing for this type of geometry. Compressibility effects, being negligible in this case, did not contribute to the difference in blade spacing. The blade geometry and stagger angle are very similar between the original and inverse designs.

Figure 7.1: Pressure Distribution Comparison of LS1 and its Inverse Design

Figure 7.2 Geometric Comparison of LS1 and its Inverse Design

7.1.3 Test Case: Cascade LS3

The cascade geometry and wind tunnel data for LS3 are taken from Whitehouse (1993).

Table 7.3 summarizes the most relevant wind tunnel conditions.

Item	Description	Value
axial velocity ratio	outlet/inlet axial velocity	1.003
turbulence intensity	low	
inlet flow angle	relative to axial	28.35 deg.
inlet flow velocity		30 m/s
Reynolds Number	based on axial chord and upstream reference	294 000

Table 7.3: Wind Tunnel Conditions for LS3 Wind Tunnel Data

The settings for the Navier-Stokes calculations used in the evaluation of the inverse design are listed in Table 7.4.

Item	Description	Value
inlet flow angle	relative to axial	28.35 deg.
inlet flow velocity		30 m/s
reference density	the density used for incompressible flow	1.199 kg/m ³
Reynolds Number	based on axial chord and upstream reference	294 000

Table 7.4: Settings for the Navier-Stokes Evaluations of the Inverse Design of LS3

Three global iterations of the design method were applied in order to design the inverse version of LS3. The pressure-coefficient distribution of LS3 and that of the inverse design are

shown in Figure 7.3. The distribution of pressure coefficients for the inverse design agrees well with the target distribution in most areas. The slight difference between the geometries of the pressure surfaces near the leading edge is apparently responsible for the difference in pressure distributions in that region. It is possible that the cascade geometry model does not contain enough degrees of freedom to model a more suitable geometry. Another possibility is that the most suitable geometry was not found by the optimization strategies. To improve the pressure distribution agreement, a denser distribution of target points might be used in the leading-edge region of the pressure surface. Fortunately, the details of flow in that region are generally less important (from a designer's perspective) than the details of the flow over the suction surface.

Figure 7.4 shows the original cascade geometry overlaid with the re-designed geometry, which is very similar.

Figure 7.3: Pressure Distribution Comparison of LS3 and its Inverse Design

Figure 7.4 Geometric Comparison of LS3 and its Inverse Design

7.2 Tests based on High-Speed Cascades

7.2.1 Overview of Tests

A total of five test cases were examined using high-speed cascade pressure distributions as input:

(1) Two tests were performed using data from a high-pressure turbine cascade which was tested at Carleton University as is designated "HS1". It has a pressure distribution showing evidence of shock wave effects over the aft portion of the suction surface. The source of data for the target pressure distribution was a Navier-Stokes solution for one of the tests and wind tunnel data for the other test. In both tests, part of the suction-side pressure distribution of the inverse design did not correspond to the target, indicating limitations to the computer implementation of the design method, or perhaps the method itself.

(2) Two tests were performed using data from a cascade designated "VKI-1". This cascade was used to compare the performance of four European cascade wind tunnels (Kiock et al., 1986) and has also been tested at Carleton University. The cascade is known as "HS2" at Carleton University. Like HS1, the pressure distribution shows evidence of shock wave effects over the aft portion of the suction surface. The strength of the shock wave effects is much less pronounced than for the cases involving HS1. In both tests, part of the suction-side pressure distribution did not correspond to the target. It is thought that, in these cases, geometric limitations of the cascade generation algorithm are the cause.

(3) One test was performed using data from a cascade designated “T106” (Hoheisel, 1990).

This low-pressure turbine cascade has a smoother pressure distribution and a larger region of diffusion on the suction side than for the other high-speed test cases. The region of diffusion is probably the most critical part of the flow field to match if the low-speed cascade is intended to investigate viscous effects in the high-speed counterpart. The low-speed inverse design was successful in producing approximately the same pressure gradients as that produced by the original cascade.

It should be pointed out that although geometric limitations appear to cause problems with meeting design objectives, some of these limitations should be maintained. Trailing-edge radius restrictions and minimum thickness restrictions should not be violated even if such restrictions cause mismatches between the target and design pressure-coefficient distributions, otherwise the cascade will be structurally unsuitable for use in wind tunnel tests. The present design procedure automatically compromises between the imposed geometric constraints and the aerodynamic requirements according to the definition of the objective function.

A discussion concerning the basis for matching the inverse design and target pressure distributions (axial chord versus surface distance) is included in Section 7.2.4.

7.2.2 Test Case: Cascade HS1 (Target C_p s based on Navier-Stokes calculations)

HS1 is a high-speed cascade representing the midspan section of the high-pressure turbine rotor from a small gas turbine engine of recent design. This cascade is also being tested in the Carleton University High-Speed Wind Tunnel (Jouini, 1999). The design outlet Mach is about 1.05 and the peak Mach number on the suction side is about 1.25. The target pressure

Item	Description	Value
number of airfoils	mounted radially	7000
LE section radius		530.700 in.
cone angle		0.2 deg.
meridional chord		0.590 in.
TE pitch		0.476 in.
stagger angle		25.10 in.
true chord		0.654 in.
inlet stagnation pressure		22.99 psia
inlet stagnation temperature		2098.13 deg. R
inlet Mach number		0.53
outlet Mach number		1.04
inlet flow angle	relative to axial	46.00 deg.
outlet flow angle	relative to axial	-58.49 deg
channel convergence ratio		1.138
specific heat ratio		1.319
Reynolds Number	based on axial chord and upstream reference	126 000

Table 7.5: Settings for the Navier-Stokes Evaluation of the Original HS1

distribution was obtained from Navier-Stokes calculations performed by the designers of the blade during the original design process (c.f. Section 7.2.3 showing test results obtained using wind tunnel data from Jouini (1999).). Table 7.5 summarizes the relevant settings for the Navier-Stokes calculations that were used to create the target distribution.

The settings for the Navier-Stokes calculations used in the evaluation of the low-speed design are listed in Table 7.6.

Item	Description	Value
inlet flow angle	relative to axial	46.00 deg.
inlet flow velocity		30 m/s
reference density	the density used for incompressible flow	1.199 kg/m ³
Reynolds Number	based on axial chord and upstream reference	126 000

Table 7.6: Settings for the Navier-Stokes Evaluations of the Low-Speed Design of HS1 (Target C_p s based on Navier-Stokes Target)

Five global iterations of the design method were applied in order to design this low-speed version of HS1. The designs at the ends of global iterations 2 and 3 were analyzed with poorly-converged Navier-Stokes solutions, possibly adding to the total number of iterations by introducing error into the modified targets for the panel-method calculations for iterations 3 and 4 respectively.

The midspan pressure-coefficient distribution of HS1 and the pressure-coefficient distribution of the low-speed design are shown in Figure 7.5. The target distribution has a region of diffusion beginning at about 65% b_x . This is primarily the result of an incident shock wave. The inverse design reproduces the degree of diffusion and its spatial extent reasonably well. The diffusion is followed by a region of strong expansion. In that region (the aft 15% of the suction surface), there is poor matching between the design and target distributions. This is thought to be due to insufficient geometric flexibility of the cascade model. The use of just three Bezier cubics to define the suction side of the blade does not allow for a blade profile with the rapid variations in surface curvature needed to match the target pressure variations. On the high-speed blade, these variations are of course the result of compressibility effects rather than blade surface curvature.

Figure 7.5: Comparison of Original HS1 and Inversely-Designed Low-Speed HS1 Pressure Distributions (Target C_p s based on Navier-Stokes Target)

Figure 7.6 Comparison of Original HS1 and Inversely-Designed Low-Speed HS1 Geometries (Target C_p s based on Navier-Stokes Target)

Figure 7.6 shows the original cascade geometry overlaid with the inversely-designed low-speed geometry. The original geometry, used only for comparison purposes, was taken from Jouini (1999). Compressibility effects played a role in determining the pressure distribution in the high-speed cascade. As a result, there are considerable differences between the low- and high-speed geometries.

7.2.3 Test Case: Cascade HS1 (Target C_p s based on Wind Tunnel Data)

The cascade geometry and wind tunnel data for the second HS1 test case were provided by Jouini (1999). The wind tunnel data, used to derive the target pressure-coefficient distribution, was taken for design conditions. Table 7.7 summarizes the most relevant wind tunnel conditions.

Item	Description	Value
inlet flow angle	relative to axial	46.0 deg.
inlet Mach number		0.486
outlet Mach number		
inlet stagnation pressure		1.562 bar
inlet static pressure		1.329 bar
inlet static temperature		300 K
inlet flow velocity		168.7 m/s
inlet density		
Reynolds Number	based on axial chord and upstream reference	521 000

Table 7.7: Wind Tunnel Conditions for HS1 Wind Tunnel Data

The settings for the Navier-Stokes calculations used in the evaluation of the low-speed design are listed in Table 7.8.

Item	Description	Value
inlet flow angle	relative to axial	46.00 deg.
inlet flow velocity		30 m/s
reference density	the density used for incompressible flow	1.199 kg/m ³
Reynolds Number	based on axial chord and upstream reference	521 000

Table 7.8: Settings for the Navier-Stokes Evaluations of the Low-Speed Design of HS1 (Target C_p s based on Wind Tunnel Data)

Two global iterations of the design method were applied in order to design this low-speed version of HS1. The pressure distribution of HS1 and that of the low-speed design do not match well, as shown in Figure 7.7.

Figure 7.8 shows the original cascade geometry overlaid with the re-designed geometry. There are considerable differences between the low- and high-speed geometries. As for the HS1 test case in Section 7.2.2, compressibility effects played a major role in determining the target distribution in the high-speed cascade, contributing to the difference in geometry between the high- and low-speed cascades.

It is doubtful, even with better geometric flexibility of the cascade model, that a low-speed cascade could produce the given target pressure distribution. The target pressure distribution in the present test case is strongly affected by compressibility effects. The present design method was developed primarily to design low-speed equivalents of shock-free, low-pressure turbine blades. Such blades generally have much smoother pressure distributions on the suction side than that of the present test case. Sections 7.2.4 and 7.2.6 describe test cases involving cascades with shock-free flows.

Figure 7.7: Comparison of Original HS1 and Inversely-Designed Low-Speed HS1 Pressure Distributions (Target C_p s based on Wind Tunnel Data)

Figure 7.8: Comparison of Original HS1 and Inversely-Designed Low-Speed HS1 Geometries (Target C_p s based on Wind Tunnel Data)

7.2.4 Test Case: Cascade HS2 (based on exit Mach number of 0.8)

VKI-1 (designated HS2 at Carleton University) was designed as a reference cascade and has been used to benchmark four European transonic cascade wind tunnels (Kiock et al., 1986). It was used in the same way during the commissioning of the Carleton University High-Speed Cascade Wind Tunnel (Jeffries, 2000). The target pressure-coefficient distribution for the design of a low-speed equivalent of HS2 was taken from wind tunnel data for design incidence with an inlet stagnation pressure of 1.3 bars, provided by Jeffries (2000). Table 7.9 summarizes the most relevant wind tunnel conditions. The outlet Mach number was about 0.8 and the peak Mach number on the suction side was about 0.94. Thus, flow in HS2 with an inlet stagnation pressure of 1.3 bars is free of shocks.

Item	Description	Value
inlet flow angle	relative to axial	30.0 deg.
inlet Mach number		0.27
outlet Mach number		0.8
inlet stagnation pressure		1.3 bar
inlet static pressure		1.24 bar
inlet static temperature		300 K
inlet flow velocity		93.7 m/s
inlet density		1.44 kg/m ³
Reynolds Number	based on axial chord and upstream reference	245 000

Table 7.9: Wind Tunnel Conditions for HS2 Wind Tunnel Data (outlet Mach number: 0.8)

The settings for the Navier-Stokes calculations used in the evaluation of the low-speed design are listed in Table 7.10. The Reynolds number could have been matched exactly to that

Item	Description	Value
inlet flow angle	relative to axial	30.00 deg.
inlet flow velocity		30 m/s
reference density	the density used for incompressible flow	1.199 kg/m ³
Reynolds Number	based on axial chord and upstream reference	234 000

Table 7.10: Settings for the Navier-Stokes Evaluations of the Low-Speed Designs of HS2

for the original (high-speed) HS2 cascade, but was set to a different value due to an inaccurate calculation for the size of the low-speed cascade. The difference in Reynolds numbers is not expected to significantly affect the results. The same minor error applies to the other test case for HS2 in the next section.

Three global iterations of the design method were applied in order to design this low-speed version of HS2. The pressure distribution of HS2 and that of the low-speed inverse design are shown in Figure 7.9. The panel method results for the low-speed design are also shown in this figure, along with the target for the panel method (the target used in the formulation of the objective function). The figure shows that the target C_p distribution for the panel method has become modified from the true target distribution during the course of the global iteration cycles. As evident, the modifications were predominantly needed on the suction side near the trailing edge. This can be expected since boundary layer displacement effects are often greatest in this region, causing inaccuracy for the panel method, which does not model the boundary layer. Two other significant contributing factors to the difference between the targets are the use of a low number of panels per blade (56) and a low number of blades per cascade (29) used in the panel-

Figure 7.9: Comparison of Original HS2 and Inversely-Designed Low-Speed HS2 Pressure Distributions (based on exit Mach number of 0.8)

method calculations. These settings for the panel method were chosen as a compromise between speed and accuracy.

The Navier-Stokes calculations show satisfactory agreement with the (main) target C_p distribution, except on the suction side very close to the trailing edge. The trailing-edge radius of the low-speed design was nearly the minimum value allowed. If the minimum allowable radius was smaller, there may have been better agreement between the pressure-coefficient distributions.

Figure 7.10 shows the original cascade geometry overlaid with the re-designed geometry. The original cascade geometry, used only for comparison purposes, was taken from Kiock et al (1986). As expected, the low-speed cascade requires a larger outlet metal angle to produce the same expansion ΔC_p between the inlet and outlet as the high-speed counterpart. Changes in other geometric parameters, including the solidity (spacing/chord length) and the stagger angle, are also apparent. However, because of experience gained from the results for LS1, no conclusions should be drawn from the observed difference in blade spacing.

If boundary layer development is considered the most important matching criterion, it would be better to use surface distance as a basis for pressure-coefficient distributions; boundary-layer development depends more on streamwise pressure-coefficient gradients than on pressure-coefficient gradients in the axial direction. The remainder of this section discusses the practical considerations of using a surface-based coordinate system and illustrates that the inversely-designed cascade is a function of the coordinate system used.

In order to facilitate a comparison of the surface-based pressure-coefficient distribution of a candidate low-speed design and that of the original design, it is perhaps best to normalize the

Figure 7.10: Comparison of Original HS2 and Inversely-Designed Low-Speed HS2 Geometries (based on exit Mach number of 0.8)

surface coordinate so that it spans from zero to unity. Each cascade would then require two different normalization lengths: one for each side of the blade. For example, the normalized surface distance could be defined as the surface distance from the leading-edge stagnation point divided by an appropriate normalization length. In this work, the normalization length is given the symbol b^* and is defined as the surface distance from the leading-edge stagnation point to the point where the true chord line exits the profile at the trailing edge, following the suction or pressure side of the blade profile; whichever corresponds to the side of interest. A practical difficulty with this approach is that the leading-edge stagnation point must be located before the surface coordinates can be defined.

Once the normalized surface-distance-based pressure-coefficient distributions of the high-speed and low-speed designs are matched, the pressure-coefficient gradients, taken with respect to a single characteristic length per cascade, will generally not be matched for both the suction and pressure surfaces. This is because the ratio of pressure-surface length to suction-surface length will generally be different for the high-speed and low-speed cascades due, for example, to differences in camber, thickness ratio and stagger angle. Because the flow along the suction surface is more critical to viscous effects than the flow along the pressure surface, the suction surface length should perhaps be selected as the characteristic length.

Figure 7.11 shows how pressure-coefficient distributions for the high-speed cascade (target values) and low-speed cascade (Navier-Stokes results) compare with each other when they are both plotted based on a normalized axial coordinate and when they are both plotted based on a normalized surface coordinate.

Figure 7.11: Pressure Distributions Based on Surface Distance and on Axial Distance

In Figure 7.11, the correspondence between the high-speed and low-speed cascades is better for the plots based on axial chord than for those based on surface distance. This is to be expected since the low-speed design was designed according to an objective function that uses an axial basis for comparing pressure-coefficient distributions. If the objective function instead compared distributions based on surface distance, the geometry of the low-speed design would be slightly different from the one obtained and the plots based on surface distance would be expected to match more closely than those based on axial coordinate.

An axial basis for pressure-coefficient distributions was used in this work because it is convenient: the coordinates are easier to normalize because the leading-edge stagnation point need not be detected; because there is only one normalization length per cascade, pressure-coefficient gradients with respect to the characteristic length (used for Reynolds number matching) may be matched for both sides of the blade. The axial basis also has the benefit of matching the tangential force coefficients (the Zweifel coefficients). As shown in Fig. 7.11, the gradients of pressure-coefficient with respect to normalized axial coordinate are also well matched, although their locations are not.

7.2.5 Test Case: Cascade HS2 (based on exit Mach number of 0.95)

The target pressure-coefficient distribution for this design of a low-speed equivalent of HS2 is taken from wind tunnel data for design incidence with an inlet stagnation pressure of 1.5 bars, provided by Jeffries (2000). Table 7.11 summarizes the most relevant wind tunnel conditions. The outlet Mach number was about 0.95 and the peak Mach number on the suction side was about 1.15. Impingement of a weak shock wave causes a fairly abrupt diffusion near the trailing edge.

Item	Description	Value
inlet flow angle	relative to axial	30.0 deg.
inlet Mach number		0.28
outlet Mach number		0.95
inlet stagnation pressure		1.5 bar
inlet static pressure		1.42 bar
inlet static temperature		300 K
inlet flow velocity		97.2 m/s
inlet density		1.65 kg/m ³
Reynolds Number	based on axial chord and upstream reference	291 000

Table 7.11: Wind Tunnel Conditions for HS2 Wind Tunnel Data

(outlet Mach number: 0.95)

The settings for the Navier-Stokes calculations used in the evaluation of the low-speed design are the same as for the previous case, and are listed in Table 7.10.

Two global iterations of the design method were applied in order to design this low-speed version of HS2. The pressure distribution of HS2 and that of the low-speed design are shown in Figure 7.12. The pressure distribution of the design matches fairly well with the target in general. The match could have been better in the aft quarter of the suction surface. It is suspected that limitations of the flexibility of the cascade geometry model are to blame, as in the test cases for HS1.

Figure 7.12: Comparison of Original HS2 and Inversely-Designed Low-Speed HS2 Pressure Distributions (based on exit Mach number of 0.95)

Figure 7.13 shows the original cascade geometry overlaid with the inversely-designed low-speed geometry. The original cascade geometry, used only for comparison purposes, was taken from Kiock et al (1986). As expected, the low-speed cascade again requires a larger outlet metal angle to produce the same expansion ΔC_p between the inlet and outlet as the high-speed counterpart. This effect is greater in this test case than in the previous case, where the same high-speed cascade was used, but with a lower exit Mach number.

Figure 7.13: Comparison of Original HS2 and Inversely-Designed Low-Speed HS2 Geometries (based on exit Mach number of 0.95)

7.2.6 Test Case: Cascade T106

The cascade geometry and wind tunnel data for T106 were taken from Hoheisel (1990).

Table 7.12 summarizes the most relevant wind tunnel conditions. The outlet Mach number was 0.590 and the peak Mach number on the suction side was about 0.696, indicating that the flow is subsonic.

Item	Description	Value
inlet flow angle	relative to axial	37.7 deg.
inlet Mach number		0.279
outlet Mach number		0.590
Reynolds Number	based on stagger-line chord and outlet reference	500 000
Reynolds Number	based on axial chord and downstream reference	429 000
Reynolds Number	based on axial chord and upstream reference	237 000

Table 7.12: Wind Tunnel Conditions for T106 Wind Tunnel Data

The settings for the Navier-Stokes calculations used in the evaluation of the low-speed design are listed in Table 7.13.

Item	Description	Value
inlet flow angle	relative to axial	37.7 deg.
inlet flow velocity		30 m/s
reference density	the density used for incompressible flow	1.199 kg/m ³
Reynolds Number	based on axial chord and downstream reference	427 000
Reynolds Number	based on axial chord and upstream reference	237 000

Table 7.13: Settings used for the Navier-Stokes Evaluations of the Low-Speed Design of T106

Three global iterations of the design method were applied in order to design this low speed version of T106. The pressure-coefficient distribution for T106 and that of the low-speed design are shown in Figure 7.14. As can be seen, the pressure-coefficient distributions are well matched. The regions of poorest match are near the leading and trailing edges. In the leading-edge region on the suction side, the rapid acceleration of the flow around a leading-edge radius (seen as an upward “spike” in the pressure distribution) was not reproduced. This also occurred for the test case for LS3 (Fig 7.3). In another test case, cascade HS2 with an exit Mach number of 0.95, the “spike” existed for the low-speed design although it was not represented in the target distribution. The lack of agreement of pressure distributions in the leading-edge region could be a result of the way in which the objective function was formulated since, in this case, only one target point is not well matched using the current definition of the fitness function. In the trailing-edge regions of T106 and its low-speed equivalent, the mismatch of the pressure distributions is a result of the method: the aft 4.5% (measured axially) of pressure distributions obtained by the panel method are ignored due to the unrealistic values of pressure-coefficient that result from the absence of flow separation in potential flow.

Figure 7.14: Comparison of Original T106 and Inversely-Designed Low-Speed T106 Pressure Distributions

Figure 7.15 shows the original cascade geometry overlaid with the inversely-designed low-speed geometry. The most noticeable differences between the blade profiles occur over the region where x/b_x ranges from 0.0 to 0.4. The inversely-designed profile has a much smaller blade thickness than the high-speed original for x/b_x from 0.0 to 0.2. A similar modification to the low-speed geometry is evident in Figures 7.6 and 7.8. The low-speed profile also has a higher blade thickness in the aft quarter of the profile. The inversely-designed cascade has a smaller stagger angle and only slightly higher outlet flow angle (-63.4 deg. for the high-speed cascade, -64.20 deg. for the low-speed cascade) compared with the original cascade. The similarity in outflow angle is a result of the fact that the Mach numbers of the "high-speed" original cascade are low enough that flow expansion due to a drop in density is small compared to that due to flow turning.

Figure 7.15: Comparison of Original T106 and Inversely-Designed Low-Speed T106 Geometries

Chapter 8: Conclusions and Potential Developments

8.1 Conclusions

The present work has presented a method for the design of low-speed linear turbine cascades for use in modeling the viscous effects associated with flow through low-pressure turbines. The method provides a systematic means by which a cascade may be inversely designed to produce a given pressure-coefficient distribution while satisfying constraints on the cascade's geometric characteristics. The method involves the use of optimization routines and standard direct flow solvers in an iterative design procedure. The optimization routines, namely a genetic algorithm and a gradient-based method, are used in a complementary way to achieve robust and efficient optimization. The direct flow solvers, namely a panel method solver and a Navier-Stokes solver, are used in a way that exploits the accuracy of the Navier-Stokes solver and the speed of the panel method solver.

Several test cases were presented, offering insight into the abilities and limitations of the method. Two tests based on low-speed cascades demonstrated that the geometry of the resulting design is similar to the original design, as expected. Two tests based on the high-speed cascade HS1 reveal that the method can work poorly when shock waves in the high-speed cascade flow complicate the target pressure-coefficient distribution. A pair of test cases using the high-speed cascade HS2 with two different flow speeds demonstrates that the outlet flow angle of the low-speed cascade increases with increased compressibility effects in the high-speed flow. A test case involving cascade T106 demonstrates a successful application of the method to a typical subsonic turbine blade row.

In general, insufficient flexibility of the geometric model used by the method was suspected of adversely affecting the quality of some of the test results.

8.2 Potential Modifications to the Cascade Design Package

This section describes several ideas (given in order of priority) that may lead to better performance of the cascade design package. Alternate design packages, based on the present one, are then proposed.

It would have perhaps been better to design low-speed cascades that match the surface-based distribution rather than the axial-based distribution of pressure coefficients with their high-speed counterparts since boundary layer development depends more on streamwise pressure gradients than on axial pressure gradients. Complications of using a surface-based distribution include the problem of finding the leading-edge stagnation point and the fact that pressure-coefficient gradients with respect to the characteristic length cannot be matched for both sides of the blade.

In order to obtain improved cascade designs, the present design method could benefit from the use of a more flexible geometric model. The present geometric model makes use of third-order polynomials: Bezier cubic polynomials. Perhaps the use of more polynomials per blade surface would be beneficial. Another potential change would be to use higher-order polynomials. The geometric model should be designed so that a wide range of cascade designs can be represented. To facilitate the optimization processes that are part of the design method, the vast majority of cascades that can be represented should be realistic in nature, e.g. not have regions of negative thickness.

In order to improve performance of the gradient-based optimizer used by the present method, the method could be replaced by a method that uses quadratic modeling. Such a model makes use of first and second derivative information of the objective function with respect to the independent parameters in order to achieve better convergence and to improve efficiency. One major problem with implementing a quadratic model is the fact that derivative information is not readily available. Derivatives could be approximated, as for the present method (which requires only first derivatives), by using a finite difference or central difference method. The finite difference method makes use of a test point on one side of the point at the location where the gradient approximation applies. The central difference method, by comparison, uses one test point on each side of the point of interest. The central difference method is preferred in this work because the test points are located symmetrically about the point of interest. Both methods require two objective function evaluations per partial derivative.

An improvement could be made to the algorithm responsible for creating panel distributions, based on blade profiles. The algorithm lacks consideration of the local curvature of the blade profile in determining the panel distribution. By concentrating more panels where curvature is higher, it may be possible to achieve better panel method results for a given number of panels per blade. Any change to the panel distribution algorithm should be made in such a way that the panel distribution remains continuously varying with geometry and with the location of the trailing-edge stagnation point (the junction point between a particular pair of neighbouring panels). In order to save computational time, it is also necessary to ensure that the majority of panel junction points (those in the forward two-thirds of the blade) are positioned independently of the trailing-edge stagnation point location. Many of the mathematical quantities calculated by the panel method will then be independent of the trailing-edge stagnation point location and will

not need to be recalculated each time the trailing-edge stagnation point is moved toward an acceptable position.

The present design method can be modified to perform other tasks related to inverse design of turbomachinery cascades. For example, by using a compressible-flow Navier-Stokes solver instead of an incompressible solver, it is possible to design a high-speed cascade based on a given pressure distribution. In this case, it may be beneficial to use a panel method solver combined with a compressibility correction. The Prandtl-Glauert compressibility correction (Anderson, 1991) is a popular method for correcting pressure coefficients from incompressible results so that they approximate compressible flow results. By using three dimensional flow solvers, modified geometric constraints and a modified objective function, the cascade design method could be converted to a fully three dimensional version. This would facilitate studies of the viscous effects in high-speed cascade flows that involve substantial three dimensional flow effects such as those due to tip clearance or blade damage. A fully three dimensional version of the design method would also facilitate the design of cascades for low-speed wind tunnels with rotating-rigs.

References

- Amrud, K.K., 1985, "Tip Leakage in a Planar Cascade of Turbine Blades," Master's Thesis, Dept. of Mechanical & Aerospace Engineering, Carleton University, Ottawa, Canada.
- Anderson, J.D. Jr., 1991, "Fundamentals of Aerodynamics," Second Edition (International Edition), McGraw-Hill, ISBN 0-07-001679-8
- Ansari, N., and Hou, E., 1997, "Computational Intelligence for Optimization," Kluwer Academic Publishers, Boston/London/Dordrecht.
- Baldwin, B.S., and Lomax, H., 1978, "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257.
- Benner, M.W., 2000, "Profile and Secondary Losses in Axial Turbines at Off-Design Incidence," Ph.D. Thesis, Dept. of Mechanical & Aerospace Engineering, Carleton University, Ottawa, Canada.
- Chan, Y.Y. 1998, "Applications of Genetic Algorithms to Aerodynamic Designs," Canadian Aeronautics and Space Journal, Vol. 44, No. 3, Sept. 1998, pp. 182-187.
- Chorin, A.J., 1967, "A Numerical Method for Solving Incompressible Viscous Flow Problems," J. Comp. Physics, Vol. 2, pp. 12-26.
- Cravero, C., and Dawes, W.N., 1997, "Throughflow Design Using an Automatic Optimization Strategy," ASME paper 97-GT-294.
- Currie, T.C., and Carscallen, W.E., 1998, "Simulation of Trailing Edge Vortex Shedding in a Transonic Turbine Cascade," Trans. ASME, J. Turbomachinery, Vol. 120, No. 1, January 1998, pp. 10-19.
- Dawes, W.N., 1988, "Development of a 3D Navier Stokes Solver for Application to all Types of Turbomachinery," ASME Paper 88-GT-70.

- Demeulenaere, A., and Van den Braembussche, R.A., 1996, "Three-Dimensional Inverse Method for Turbomachinery Blading Design," ASME Paper 96-GT-39.
- 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 Algorithm." ASME Paper 99-GT-441.
- Farin, G., 1988, "Curves and Surfaces for Computer Aided Geometric Design," Academic Press, Inc., ISBN 0-12-249050-9.
- Fletcher, R., 1987, Practical Methods of Optimization. John Wiley & Sons. Chichester/New York/Brisbane/Toronto/Singapore.
- Giannakoglou, K.C., 1999, "Designing Turbomachinery Blades using Evolutionary Methods," ASME Paper 99-GT-181.
- Goel, S., Cofer IV, J.I., Singh, H., 1996, "Turbine Airfoil Design Optimization," ASME Paper 96-GT-158.
- Goldberg, D.E., 1989, "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley Publishing Company, Inc., ISBN 0-201-15767-5.
- Hoheisel, H., 1990, "Test Cases for Computation of Internal Flows in Aero Engine Components," AGARD-AR-275, July 1990, pp. 112-117.
- Holland, J.H., 1992, "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence," The MIT Press, ISBN 0-262-08213-6.
- Islam, A.M.T., and Sjolander, S.A., 1999, "Deviation in Axial Turbines at Subsonic Conditions," ASME Paper 99-GT-26, accepted for publication, ASME Transactions.
- Jeffries, M.S., 2000, "Initial Investigations of Axial Turbine Aerodynamics Using the Carleton University High Speed Wind Tunnel," Ph.D. Thesis, Dept. of Mechanical & Aerospace Engineering, Carleton University, Ottawa, Canada.

Jouini, D.B., 1999, "Influence of Leading-Edge Geometry on the Performance of Axial Turbine Cascades at Off-Design Conditions," Ph.D. Thesis, Dept. of Mechanical & Aerospace Engineering, Carleton University, Ottawa, Canada.

Kind, R.J., Serjak, P.J., and Abbott, M.W.P., 1998, "Measurements and Prediction of the Effects of Surface Roughness on Profile Losses and Deviation in a Turbine Cascade," ASME Trans., J. Turbomachinery, Vol. 120, pp. 20-27.

Kiock, R., Lehthaus, F., Baines, N.C., and Sieverding, C.H., 1986, "The Transonic Flow Through a Plane Turbine Cascade as Measured in Four European Wind Tunnels," ASME Trans., J. Eng. for Gas Turbines and Power, Vol. 108, April 1986, pp. 277-284.

Krishnakumar, K., 1989, "Micro-Genetic Algorithms for Stationary and Non-Stationary Function Optimization," Proc., SPIE, Intelligent Control and Adaptive Systems, Vol. 1196, 1989, pp. 289-296.

Lakshminarayana, B., Camci, C., Halliwell, I., and Zaccaria, M., 1992, "Investigation of Three Dimensional Flow Field in a Turbine Including Rotor/Stator Interaction. Part 1: Design, Development and Performance of the Turbine Research Facility," AIAA Paper 92-3325.

Léonard, O., and Van den Braembussche, R.A., 1991, "Design Method for Subsonic and Transonic Cascade with Prescribed Mach Number Distribution," ASME Paper 91-GT-18.

Léonard, O., and Van den Braembussche, R.A., 1992, "Inverse Design of Compressor and Turbine Blades at Transonic Flow Conditions," ASME Paper 92-GT-430.

Pierret, S. and Van den Braembussche, R.A., 1998, "Turbomachinery Blade Design using a Navier-Stokes Solver and Artificial Neural Network," ASME Paper 98-GT-4.

Singh, U.K., 1996, "A Prescribed Velocity Distribution Scheme for Turbomachine Blading," ASME Paper 96-GT-177.

Tong, S.S., and Gregory, B.A., 1992, "Turbine Preliminary Design Using Artificial Intelligence and Numerical Optimization Techniques," *Journal of Turbomachinery*, Jan. 1992, Vol. 114, pp. 1-7.

Trigg, M.A., Tubby, G.R., and Sheard, A.G., 1997, "Automatic Genetic Optimization Approach to 2D Blade Profile Design for Steam Turbines," ASME Paper 97-GT-392.

Vicini, A., and Quagliarella, D., 1997, "Inverse and Direct Airfoil Design Using a Multiobjective Genetic Algorithm," *AIAA Journal*, Vol. 35, No. 9, Sept. 1997, pp.1499-1505.

White, F.M., 1991, "Viscous Fluid Flow: Second Edition," McGraw-Hill, Inc., ISBN 0-07-069712-4.

Whitehouse, D.R., 1993, "The Effect of Axial Velocity Ratio, Turbulence Intensity, Incidence and Leading Edge Geometry on the Off-Design Performance of a Turbine Blade Cascade," Master's Thesis, Dept. of Mechanical & Aerospace Engineering, Carleton University, Ottawa, Canada.

Wisler, D.C., 1985, "Loss Reduction in Axial-Flow Compressors Through Low-Speed Model Testing," *Trans. ASME J. Eng. for Gas Turbines and Power*, Vol. 107, April 1985, pp. 354-363.

Yaras, M., and Sjolander, S.A., 1988, "Measurement of the Tip Clearance Flow in a Rectilinear Cascade of Turbine Blades: A Test Case for Computational Methods," Report No. M&AE 88-1, April 1988, Dept. of Mechanical & Aerospace Engineering, Carleton University, Ottawa, Canada.

Ye, Z.-Q., 1984, "A Systematic Computational Design System for Turbine Cascades, Airfoil Geometry, and Blade-to-Blade Analysis," *Trans. ASME J. Eng. for Gas Turbines and Power*, Vol. 106, July 1984, pp. 598-605.

Appendix A: The Cascade Generation Algorithm

Figure A.1 shows a typical cascade geometry revealing the seven curves used to construct it. The x-axis is aligned in the axial direction (i.e. axis of rotation in the case of turbomachinery). The join point between curves is labeled with a letter and a number. The letter is "S" for points on the suction side of the blade and "P" for points on the pressure side. The control points for the Bezier cubic polynomials are labeled starting with the letter and number of the associated junction point, followed by a letter: "W" (for West) for points nominally on the left-hand side of the join point and "E" (for East) for points nominally on the right-hand side. The cascade geometry is specified by a set of 25 independent parameters. In the following description of the method of geometric construction, each of the cascade parameters is associated with a number between 1 and 25 that was arbitrarily assigned for reference purposes during development of the algorithm. The same numbering system for the parameters is used in the user interface of the design package.

The geometry is constructed starting with the trailing-edge circle, such that the extreme right point on the circle passes through (1,0). The radius of this circle is set by parameter #10. The leading-edge ellipse is located such that its centre lies at a y-coordinate equal to the tangential chord, given by parameter #3. The semi-minor axis length and major-to-minor axis length ratio are given by parameters #4 and #5 respectively. The ellipse is formulated in its rotated position such that the major axis is aligned with the leading-edge metal angle and the leftmost point has an x-coordinate of zero. Note that the axial chord will always be unity. The leading-edge metal angle is the average of parameters #6 and #7, which are, respectively, the angles of the tangent lines at the suction and pressure surface leading-edge join points (measured

Figure A.1: Cascade Geometry Construction

counterclockwise from the positive x-axis). The leading-edge wedge angle is therefore equal to parameter #6 minus parameter #7. Parameters #6 and #7 might have alternatively been used to specify the leading-edge metal angle and the leading-edge wedge angle. Knowing the position of the ellipse, the leading-edge wedge angle and the leading-edge metal angle, the join points S1 and P1 are located.

Next, the join points and control points for the Bezier cubic polynomials are located. To locate point S2, a straight line is drawn from the centre of the leading-edge ellipse to point (1.0) at the trailing edge. At the known x-coordinate of point S2 (parameter #2) a point is located on the straight line. From this point, a vertical offset (parameter #8) is applied to locate point S2. Next, the slope at point S2 is determined. Since the curvature at point S2 is intended to be concave down, it is a requirement that points S1 and S3 lie “below” the line of tangency at S2. At this stage, point S3 has not yet been positioned, so a reasonable slope at S2 must be enforced so that point S3 may be placed below the tangency line at S2. The slope at S2 is specified indirectly by linearly interpolating between two slopes using a number between 0 and 1 (parameter #9). The two limiting slopes are: (1) the slope of the line connecting point S1 to point S2 and (2) the slope of the line connecting point S2 to point (1.0). Figure C1 indicates the range of slopes allowed at point S2 symbolically by a miniature arc bounded by straight line segments at the limiting slopes. Next, the “East” control point (the control point to the right) of point S2, point S2E, is determined. The distance from S2 to S2E (casually referred to as the “handle length” for point S2E) is specified by parameter #24. Since point S2 and its slope are known, it is a straightforward task to find point S2E.

In order to locate point S4, several steps are taken. First, the angle of a line from point S2E to the upper point of tangency with the trailing-edge circle is measured. This is used as a

limiting line of tangency for point S4. The other limiting tangency line is taken to be the vertical line of tangency from the right edge of the trailing-edge circle. An interpolation factor (parameter #12) is then used to linearly interpolate between these tangency limits (on the basis of angle, rather than slope) to find the tangency line belonging to point S4. Once the direction of tangency is established, it is a simple matter to find point S4. This method of placing S4 ensures that point S2E will not lie “above” the line of tangency at S4. In an earlier version of the algorithm, this would ensure that the curvature of the suction surface is concave down near the trailing edge. Since then an additional join point has been added (point S3) and downward concavity is no longer guaranteed at the trailing edge. Nothing would be lost by allowing the line of tangency at S4 to pass below point S2E. With point S4 and its direction of tangency known, the trailing-edge metal angle is calculated (for use in locating point P3 and for other purposes) using the trailing-edge wedge angle (parameter #11). As well, since the “handle length” for S4W is given (parameter #15), point S4W is known.

Point S3 is placed so that the suction surface will likely be concave down near the trailing edge. In order for this to be the case, point S3E must lie below the tangency line from point S4. Concavity downwards must be enforced at point S2. In order to achieve this, point S3W must lie below the line of tangency at point S2. Point S3's x-coordinate is found by using an interpolation fraction (parameter #23) that uses the x-coordinates of points S2 and S4 as limits. To achieve the required and the desired concavities at points S2 and S4 respectively, the y-coordinate of point S3 should be below the lines of tangency from these two points, since S3W and S3E must then also be below these lines. The lowest intersection of these tangent lines with a vertical line at S3's x-coordinate is used as an upper limit for the position of point S3. To choose a suitable range for S3, the upper limit, as specified above, and the middle of the range, as determined by the intersection of the line from S2E to S4W with the vertical line at S3's x-coordinate, are used to

locate the bottom of the suitable range. An interpolation fraction (parameter #18) is then used to set the y-coordinate of point S3. The direction of tangency at S3 is set by using another range and another interpolation fraction (parameter #19). This range varies from the slope from S3 to S2E to the slope from S3 to S4W. This method of specifying the direction of tangency at point S3 ensures that points S2E and S4W will lie either both above or both below the tangent line of S3. This ensures that the direction of concavity at S3 is the same from both sides. Concavity at point S3 will be concave up if point S3 lies below the line connecting S2E to S4W and vice versa.

Now that point S3 has been located, point S3W is located by using an interpolation fraction (parameter #14) and an acceptable range to ensure that it will lie below the tangent at S2 (to ensure that the curvature is concave down at S2). The acceptable range for point S3W lies along the line of tangency at S3 and extends west from point S3 by the minimum of the following distances: (1) the distance from S3 to the point of intersection of tangent lines of S2 and S3, (2) the distance from point S3 to S2E. Because the Bezier curve from point S2 to point S3 is now completely defined, the curvature at point S3 is calculated and the “handle length” of point S3E is calculated (by an iterative method) to match the magnitude of curvature at S3. At this point, the Bezier curve from point S3 to point S4 is now also completely determined. Because the handle length for point S3E is determined after the line of tangency at S4 is set, it is possible that point S3E will lie above this line of tangency, in which case curvature will be concave up at the trailing edge.

The forward Bezier curve of the suction surface is completed by specifying points S1E and S2W. Point S1E must lie below the line of tangency at point S2. It is specified by an interpolation fraction that finds a point between point S1 and the point of intersection between the tangents of S1 and S2. Point S2W is then located (by an iterative scheme) to match curvature at

point S2. If the position of point S2W happens to lie above the tangent at S1, the curvature at S1 will be concave up. Fortunately, this condition is rare (if at all possible given typical parameter limits) and has never been encountered in practice.

Point P3 is located knowing the trailing-edge wedge angle (parameter #11) and the trailing-edge metal angle (calculated after finding point S4). Point P2 is located by a specified x-coordinate (parameter #22) and specified vertical offset (parameter #20) from point S2. Point P1E is found knowing the location of P1, the direction of the tangent there, and by using parameter #16 to set the handle length. Similarly, point P3W is located using a handle length specified by parameter #17. The slope at point P2 is specified by an interpolation fraction (parameter #21) that is used to interpolate between the following two slopes: (1) the slope of the line passing through P1E and P2, and (2) the slope of the line passing through P2 and P3W. As for point S3, this ensures that the direction of concavity will be the same from both sides of P2 by causing points P1E and P3W to lie on the same side of the tangent of P2. The magnitude of curvature at P2 is specified (parameter #25). The handle lengths for points P2W and P2E are found (using an iterative scheme) to meet this curvature requirement. Once the handle lengths are known, points P2W and P2E are then located using the slope at P2.

The spacing is used to help calculate the throat width. A specialized subroutine finds the point on a surface nearest to any given point. To find the throat point, the point given to this subroutine is point P3 translated vertically (so that it belongs to the next blade above), and the surface given to the subroutine is the suction surface of the original blade. To calculate the unguided turning, the slope of the tangent at the throat point is compared to the slope at point S4.

Appendix B: User's Manual for Foil Design GAFDO

Table of Contents

B.1 Introduction	127
B.2 Software Setup	128
B.3 The Menu System	131
B.4 Using the Genetic Algorithm Optimizer: GAFDO	132
B.4.1 Preparing a GA Run	132
B.4.2 Setting the Menu Options for a GA run	133
B.5 Using the Gradient-based Optimizer to Optimize a Turbine Cascade Design	136
B.6 Analyzing a Turbine Cascade	139
B.6.1 Using the Panel Method to Analyze a Turbine Cascade	139
B.6.2 Using the Navier-Stokes Solver to Analyze a Turbine Cascade	141
B.6.2.1 Overview	141
B.6.2.2 Generating a Grid	143
B.6.2.3 Setting the Flow Solver Parameters	148
B.6.2.4 Monitoring the Solution	152
B.6.2.5 Using the Results	153
B.7 Modifying the Target Pressure-Coefficient Distribution	154

B.8 Miscellaneous Features	126 156
B.8.1 Support for Tecplot Graphics	156
B.8.2 Automated Features	157
B.8.3 Support for Parallel Processing using a Computer Network	158
B.8.4 Navier-Stokes and Euler Solutions for General Problems	159
B.8.5 Inverse Design based on Geometry	161
B.8.6 Generic GAFDO	162
B.8.6.1 Compiling and Executing the Main Program	162
B.8.6.2 Compiling and Executing the Slave Program (used for Network operation)	163
B.8.6.3 Demonstration of Generic GAFDO	165

List of Tables

Table B.1 Explanation of Grid Menu Settings	144
Table B.2: Explanation of CFD Menu Settings	149
Table B.3: Files Compatible with Tecplot	156

List of Figures

Figure B.1: The Main Menu of GAFDO.	131
Figure B.2: Main Menu for the GA	135
Figure B.3: Figure for Explaining Grid Menu Settings	147

B.1 Introduction

GAFDO is a software package that is named after one of its main components: a genetic algorithm (Genetic Algorithm For Design Optimization). It is capable of inversely designing turbine cascades, based on pressure-coefficient distribution, to the level of accuracy afforded by a Navier-Stokes solver. The turbine cascade designs produced by this software are intended for use in low-speed wind tunnels. Other capabilities of the package include:

- editing features for cascade designs
- 2-D panel method analysis of cascades: pressure distribution, velocity field estimate
- 2-D and limited 3-D Euler and Navier-Stokes analyses for cascade and other flows
- grid generation: structured H-mesh, blocked flow, unblocked flow, import capability
- graphical output in Tecplot format for cascade geometry, curvature distribution, pressure distribution, grid geometry, CFD residual plots

In addition, users who obtain the generic version of GAFDO may customize the program to solve a very wide range of optimization problems.

The main computer program of the design package is called "Master.exe", named for the role it plays when communicating with a second program, called "Slave.exe". The latter, not essential to the design package; is used to accelerate genetic algorithm runs by dividing work amongst two or more computers connected by a network. The program "Master.exe" contains all of the components of the design package except for the Navier-Stokes flow solver.

This user's manual begins by explaining how to prepare the software for use on a computer. Each component of the package is then explained in an order that reflects the steps required to carry out a design exercise. It is assumed that the user is familiar with the design

procedure. The last sections of this manual describe more advanced topics such as graphics software support, automated features, parallel processing via a network, features extraneous to the design method and a generic version of GAFDO.

B.2 Software Setup

The main program, called "Master.exe", is the only executable program required to use the GAFDO package on a single computer. If more than one computer will be used to perform genetic algorithm runs, then the file "Slave.exe" is also required. In order to prepare a run of the design method, it is recommended that the following method be used to organize the required files and folders:

Step 1: Create a directory (folder) for the run. This directory should preferably be located near the root directory of the applicable disk drive so that excessively long path names are avoided. For example, if the turbine blade corresponding to the target pressure-coefficient distribution is called "turbine1", then the directory could be called "turbine1 run".

Step 2: Place a copy of the target distribution in this directory. This will make the target file readily available when it is needed.

Step 3: Create a directory in the main directory for the run. This directory will contain the files generated during the first iteration of the method. It will contain, among other things, the

files involved in the first use of the genetic algorithm. As an example, this directory could be called "1.0 first GA run".

Step 4: In the directory just created, add a copy of the main program "Master.exe" (and a copy of "Slave.exe" if desired). Also copy the main target file into this directory (the main target file must end with the extension ".udf").

Step 5: In the directory last created, make a folder that will be used to store the files involved in the first use of the gradient-based optimizer. For example, this folder could be named "1.1 first pass hillclimb".

Step 6: In the directory last created, add a copy of "Master.exe" and the file containing the target distribution. Create a directory that will hold the files involved in the Navier-Stokes evaluation of the first-pass cascade design. For example, this folder could be named "1.2 first pass NS".

Step 7: In the directory last created, add a copy of "Master.exe" and "NSFlow.exe" (the latter is the Navier-Stokes flow solver).

Step 8: In the directory last created, add a folder that will hold the revised temporary target file. For example, this folder could be called "1.3 udf for pass 2".

Step 9: Add a copy of the target distribution to the last directory created and rename the target file so that its extension is changed from ".udf" to ".mta". Also add a copy of "Master.exe" to the directory last created. "Master.exe" will be used to modify the

original target distribution (the “.mta” file) using the resulting panel method and Navier-Stokes results (“thefoil.gqp” and “thefoil.hqp” respectively) from the first-pass of the method.

At this point, all of the directories required for the first pass of the method have been created and all of the required files have been placed in these directories. After the first pass of the method has been completed, another corresponding set of directories must be created for each subsequent pass. In order to conserve disk space, the executable files may be deleted after they have been used.

MAIN MENU

1. load/save/form parameter limits
2. set pressure-coefficient reference location: currently upstream
3. GAFDO: pressure-based fitness
4. automatic GAFDO (pressure-based fitness)
using "default .plm" and "gafdoset.txt"
5. gradient-based optimizer
6. set panel method settings for fitness evaluation (currently code 2)
7. load/save/edit cascade parameters, generate cascade geometry
8. generate input files for the Dawes code (if not using disk,
first run profile generator for ability to make initial velocity estimate)
9. evaluate last cascade generated (in "thefoil.def")
10. calculate pressure-coefficient distribution from Dawes code results
11. target modification menu (see help, #16)
12. menu driven components for fitness evaluation via panel method
including profile generator (writes to memory or disk)
13. automatically generate Dawes code input files using "gridset.txt"
and "CFDset.txt" for settings
14. set disk usage option: currently the disk is used
15. GAFDO: geometry-based fitness
16. view help message
17. quit

Figure B.1: The Main Menu of GAFDO

B.3 The Menu System

The main menu of "Master.exe" appears when the program is started, and is shown in Figure B.1. In this case, each menu option is listed starting with a number followed by a period. If a menu option is currently not applicable (due to the state of other menu settings), its number appears in parentheses.

Menus with many options are listed one segment at a time. Pressing ENTER causes the next segment to be displayed. If the user enters a real or negative number when an integer is expected, the program may crash. When a menu is listed one segment at a time, only the menu options of the last segment displayed are available. When the last (or only) segment of a menu has been displayed, pressing ENTER will display the same menu, starting from the top.

B.4 Using the Genetic Algorithm Optimizer: GAFDO

B.4.1 Preparing a GA Run

In order to prepare the target pressure distribution, an ASCII text file editor must be used to prepare a file called "thefoil.udf" (The extension "udf" stands for "User Defined File"). This file contains the definition of the target pressure-coefficient distribution that is used by the optimization routines when evaluating cascade fitness. The first line of the file should contain an integer equal to the number of values of C_p that are listed in the remainder of the file. The remaining lines of the file define the pressure-coefficient distribution. Each line consists of a real number defining an axial coordinate (normalized by axial chord) followed by a real number defining the associated pressure-coefficient value. The file must list target distribution points in order, starting at the axially-aftmost location, working toward the leading edge via the suction surface, then back along the pressure surface, ending at the same point.

The pressure coefficients contained in the target distribution file must be defined as:

$$C_p = (P - P_{ref}) / q_{ref}$$

where C_p is the pressure coefficient, P is the static pressure, P_{ref} is the reference static pressure and q_{ref} is the reference dynamic pressure. The available reference locations are upstream and downstream of the cascade. The precise locations of these reference locations are given by the axial extents of the Navier-Stokes code solution domain (by default: 1/2 axial chord upstream of the cascade in the axial direction and 1 axial chord downstream of the cascade in the axial

direction), since the final cascade design is designed so that its Navier-Stokes (not panel method) solution agrees with the target distribution.

After the target pressure distribution file has been created, the user should copy it into the directory where the GA will operate. A copy of the file should be made and named "thefoil.mta" (The extension "mta" stands for "Master TArget"). This file will be needed to retain the original target information if a design exercise is carried out, in which case the user-defined file ("thefoil.udf") will undergo modification.

B.4.2 Setting the Menu Options for a GA run

This section describes how to start a GA run. First, ensure that the file "thefoil.udf" exists in the working directory, along with the main program: "Master.exe". Run the main program. The main menu will appear.

The cascade parameter limits are automatically set to default values unless the user specifies them. In order to set customized parameter limits, select menu option #1. A menu entitled "Parameter Limits Menu" appears. This menu allows parameter limits to be loaded, set to default, saved, viewed and formed. In order to edit parameter limits, the limits must first be saved to a file, then edited (with a text editor), then loaded back into memory. The menu options are self-explanatory except for option #4: "Form parameter limits based on the parameter values in memory". This option forms a new set of parameter limits that are centred about the current parameter values in memory and that are within the default limits. In order to use this option correctly, a cascade known to be similar in geometry to the cascade about to be designed must be loaded into memory (e.g. from menu option #7 of the main menu). It is important that the

parameter limits surround the parameter values of any cascade design of interest. To be conservative, it is recommended that the default limits are used instead. After custom parameter limits have been set, it is best to save them (preferably using the default file name "default_.plm").

If the GA is to be used to inversely design based on pressure-coefficient data (rather than geometry), main menu option #2 should be set for the appropriate reference location used in the definition of pressure coefficient. This must correspond to the reference location used in the definition of the target distribution (see Section B.4.1).

The remaining steps involved in starting a GA run involve selecting options from the main menu of the GA. The remainder of this section discusses how this is done.

From the main menu, select option #3 to display the main menu of the GA, entitled "GAFDO Main Menu" (see Figure B.2). At this point, if the parameter limits have been set previously, the user will be asked whether they should be used, or whether default values should be set. If parameter limits have not been previously set, default values will be set automatically. When default values are set, the user must enter the inlet flow angle. If the user enters the incorrect inlet flow angle, the value may be corrected by going back to the main menu, then reselecting option #3.

In the main menu for the GA, a title for the run may be specified (option #1). An eight-letter name for use in naming the output files must be entered as the case name (option #2). The remaining menu options may be left at their default values. Menu option 77 should be selected to

GAFDO Main Menu

1. title: "title goes here"
 2. case name:
 3. starting population size (generation 0): 40
 4. population size starting with generation 1: 20
 5. last generation: 750
 6. crossover probability (default 0.12): 0.119999997
 7. mutation probability (default 0.009): 0.00899999961
 8. fitness scaling will be used.
 9. fitness scaling factor (suggested between 1.2 and 2.0, default 1.2): 1.20000005
 10. selection by roulette wheel (not by tournament)
 11. elitism will be implemented
 12. Multiple A, the multiples of which indicate the generations for which the parameter limits should be adjusted: 20
 13. Multiple B, the multiples of which indicate the generations for which the parameter limits should be set to original values: 0
 14. fraction of the population to keep during parameter limit adjustment: must be from 0.6 to 0.9 (default = 0.8): 0.0800000012
 15. sparse disk output
 16. network capability is disabled
 - (17) number of slaves to scan: 2
 - (18) number of blades a slave processes at once: 10
 - (19) file scanning will proceed quickly.
 20. load defaults
 21. load setting file
 22. finished (begin the run)
- Special:
23. quit to main menu
 77. save setting file

Figure B.2: Main Menu for the GA

save the settings (to a text file called "gafdoset.txt"). To start the GA run, select menu option #22.

Menu options #2 to #14 are self-explanatory and control the GA via parameters and other settings. Refer to Chapter 6 for information about the parameters and settings related to the GA.

Menu option #15 controls the amount of output written to disk. This affects the amount of disk space used during a GA run, as well as the time required for the run. It is recommended that little output is written to disk. Setting option #15 to "sparse disk output" results in disk

output for every 10th generation. Setting option #15 to "frequent disk output" results in output for every generation. Setting option #15 to "full disk output" results in the same output given by "frequent disk output" plus (raw and scaled) fitness information for every member of every generation in the file "[casename].out".

Menu options 16 to 19 deal with the use of more than one computer on a network to share in fitness evaluations. These options are described in Section B.8.3.

Menu option #20 resets the menu settings to default values. Menu option #21 sets the menu settings according to data stored in the previously saved file "gafdoset.txt".

B.5 Using the Gradient-based Optimizer to Optimize a Turbine Cascade Design

When the GA run is finished, the final product is contained in a file called "[case name].chr". Before beginning the gradient-based optimizer, the user may want to check the quality of the result by plotting the target pressure-coefficient distribution and the distribution from the cascade design produced by the GA (See Section B.6.1 for instructions on how to perform a panel method analysis. See Section B.8.1 for instructions on how to view the data.). A less preferred, but much easier, method of checking the quality of the design produced by the GA is to load the design parameters into memory from the file "[case name].chr" (menu option #7 from the main menu), then perform a fitness evaluation (menu option #9), noting the fitness value displayed on the screen. A high value of fitness indicates close agreement between the pressure-coefficient distribution of the cascade design and the target distribution.

If the user is satisfied with the quality of the result of the GA, the gradient-based optimizer should be used to further the optimization (unless the GA has optimized to a satisfactory degree). The gradient-based optimizer is run by selecting option #5 from the main menu. Before doing this, the user should ensure that a copy of the target pressure-coefficient distribution file ("thefoil.udf"), and the file containing the starting-point cascade design (e.g. "[case name].chr") are in the working directory.

The gradient-based optimizer begins by asking the user whether the fitness function should be based on pressure distribution or geometry. It then asks whether the output file "hill.out" should be cleared and offers the user a chance to opt out of using the gradient-based optimizer. The file "hill.out" contains a record of the best cascade designs encountered during a run of the gradient-based optimizer. The only reason why the file "hill.out" should be cleared is if it contains unwanted data and the file size is excessive. In order to reduce the size of this file without losing the latest cascade design recorded, the file may be edited by removing all but the last 28 numbers.

The gradient-based optimizer then asks for the number of gradient evaluations to perform. This is the only method by which the program is set to stop operation. However, it is recommended that the user specify an extra large number of iterations, then simply interrupt the program when a sufficient level of fitness has been attained, as judged by the user. The user may open another copy of the software from the same directory in order to check the result by loading and then analyzing the latest cascade design in the file "hill.out".

After the number of gradient evaluations have been entered, the gradient-based optimizer asks the user for the source of cascade parameters that will be used as a starting point in the optimization. The user may opt to use the parameter values in memory, in which case the user must have previously loaded the values (e.g. from "[case name].chr" or even from a previous version of "hill.out"). The user may also opt to load a set of cascade parameters at this point, in which case the user is given access to the same, self-explanatory, loading facility available under main menu option #7).

After the optimizer has identified the starting cascade parameters, these parameters are listed to the screen. Note the value of the last parameter, which is the inlet flow angle (degrees from axial).

The parameter limits are then checked to see if a definition has been set since the program was started. If a set of parameter limits exist in memory, the user is given the option of using them. The limits set for the inlet flow angle will be ignored. The inlet flow angle will be fixed according to the starting cascade design. If no cascade limits have been previously set, default values will automatically be set.

The gradient-based optimizer then assesses whether the starting cascade parameters fall within their respective limits. If not, they are raised or lowered to reach the minimum or maximum limit (respectively), and a notice of the change is displayed. This affects the starting design.

The optimizer then informs the user that it is about to evaluate the fitness of the starting cascade design. It gives the user a last chance to exit to the main menu. If the user chooses to

continue, the optimizer will prompt the user for panel method settings: the number of panels per blade and the number of blades surrounding the central blade in the finite cascade. Default values are recommended. Whatever values are chosen, the user must remain consistent from one global iteration of the design method to the next. To help the user remember the settings, a file called "hcnote.txt" is created (if it does not already exist) in the working directory and a line stating the settings is appended to the file.

The gradient-based optimizer will then proceed automatically from this point. Remember to check the result before it has finished and to interrupt the code (e.g. by pressing Ctrl and C) when the result is satisfactory.

B.6 Analyzing a Turbine Cascade

B.6.1 Using the Panel Method to Analyze a Turbine Cascade

In order to analyze a cascade using the panel method, the cascade geometry must exist in either memory or on disk. A cascade geometry may be produced in either memory or on disk only after cascade parameters have been loaded (or otherwise set) in memory. To load cascade parameters and define the cascade geometry, select option #7 from the main menu.

After the cascade geometry has been defined, the panel method analysis may be performed by following these steps:

Step 1: Set the appropriate reference location for the definition of pressure coefficient using main menu option #2.

Step 2: Set the panel method settings using main menu option #6.

Step 3: Perform a fitness evaluation based on pressure using main menu option #9 followed by option #1.

After the panel method evaluation is complete, the computer attempts to calculate a fitness value using the target pressure-coefficient stored in "thefoil.udf". If this file does not exist, an error message will be displayed. This does not affect the panel method result, and should be disregarded (unless a fitness evaluation is desired).

It is recommended that the disk usage option be set to use the disk (rather than memory) for output. That way, the output files are written to disk and can be examined further. The only thing that can be done with a panel method result in memory is it can be compared with a target pressure-coefficient distribution as part of a fitness evaluation (main menu option #12 followed by option #6).

The panel method result, when stored to disk, is stored in a file called "thefoil.cas" (Incidentally, the analogous result using the Navier-Stokes solver has the same file name.). The panel method version of the file "thefoil.cas" contains the coordinates of the panel junction points and the corresponding pressure-coefficient data. It also contains a statement of the following relevant items: inlet flow angle, outlet flow angle, number of blades surrounding the central

blade, the blade spacing to axial chord ratio, the reference location used in the definition of pressure coefficient.

B.6.2 Using the Navier-Stokes Solver to Analyze a Turbine Cascade

B.6.2.1 Overview

The computer program used to implement the Navier-Stokes solver (which is also capable of solving Euler flows) is called "NSflow.exe". This section and the following four sections describe how to use "Master.exe", in combination with "NSflow.exe", to analyze a turbine cascade in 2-D flow.

This program has no user interface; when it is run, it automatically loads input files produced by "Master.exe" then begins calculations. The process of preparing the input files has two main parts. First, the user must create a grid. The grid generator included in "Master.exe" was developed specifically for 2-D turbine cascades, and is menu-driven. Section B.6.2.2 explains how to use the grid generator. Second, the user must specify the flow solver parameters that complete the problem specification. Section B.6.2.3 explains how to set these parameters. After these two steps are complete, "Master.exe" generates the input files required by "NSflow.exe".

Up to three input files are produced by "Master.exe" for any given flow problem: "gridoutp.grd", "gridoutp.daw" and, optionally, "initvels.txt". The grid node coordinates are contained in a file called "gridoutp.grd". For turbulent flow, "gridoutp.grd" will also contain data required for turbulence modeling (distances to the nearest two surfaces for each cell centre).

The flow solver settings and information about the approximate curvature of a blade are contained in a file called "gridoutp.daw". This file is thoroughly labeled and can be modified using a text editor. In this way, changes to CFD settings can be made without having to use "Master.exe" (One change to gridoutp.daw that cannot be made is a change from an Euler run or a laminar Navier-Stokes run into a turbulent Navier-Stokes run, because the file "gridoutp.grd" would not have the data required for turbulence modeling.). An optional input file is "initvels.txt". This file contains the coordinates of grid cell centres and the corresponding values of pressure-coefficient (based on either an upstream or downstream reference) and velocity components. It is produced using information from a 2-D panel method analysis, and may be used by the flow solver to reduce computational time by initializing the velocity and pressure fields with reasonable estimates.

As "NSflow.exe" runs, it writes information about residual flow quantities to a file called "Fg_ch.2". By examining this file, the user may monitor a run in progress. Section B.6.2.4 discusses how to examine and interpret this file.

The output of a run is written to files "output1F" and "turbul.dat" at iterations specified by the user, and after the last iteration. The file "output1F" is the main output file. It contains, among other flow quantities, the information required to calculate the pressure-coefficient distribution for a blade. Instructions for automatically calculating the pressure-coefficient distribution appear in Section B.6.2.5. The file "turbul.dat" is only needed in the event that the user wishes to resume calculations. There are two reasons why a user would want to resume a calculation: (1) to allow convergence to proceed further, (2) to begin a new run with a slightly different geometry or a slightly different value for a parameter, but the same number of cells in each part of the grid.

B.6.2.2 Generating a Grid

Before beginning to prepare the grid, a cascade geometry must exist either in the file "thefoil.def", or in memory. If a cascade geometry does not exist in "thefoil.def", or is not the latest geometry created in memory (since "Master.exe" was started), the user should use the facilities under main menu option #7 to prepare a cascade geometry. It is recommended that the cascade geometry be output to disk rather than memory so that it is readily accessible.

The grid menu is accessed through main menu option #8. The grid menu displays values for most of the grid-controlling parameters. A description of each grid menu option is given in Table B.1. Figure B.3 serves to help illustrate some of the descriptions provided in Table B.1. As a time-saving feature, the menu settings can be stored and retrieved from a file called "gridset.txt" by selecting menu options 66 and 43 respectively.

After the grid settings have been set (and saved, if desired), the user should select grid menu option #44 to continue. At this point, the program offers the user a chance to load a grid from a file (see section B.8.4). To generate the grid specified by the settings of the grid menu, the user should choose not to load a grid. The program will then generate a grid in memory and offers to write a copy of the grid to a file called "gridoutp.tec" for the purpose of viewing the grid with graphics software. After the user has accepted or rejected the offer, the menu for setting flow solver parameters is presented. Section B.6.2.3 describes that menu and how to set the parameters.

Menu Option	Description
1	source of cascade geometry: either the file "thefoil.def" or a cascade generated in memory
2	type of grid: Style 1 is for a 2-D cascade that uses blocked cells to represent a blade. Style 2 is for a 2-D cascade that does not use blocked cells to represent a blade. Style 3 is for the 3-D version of Style 1. Capable of modeling blades having tip clearance.. Style 4 is for general 2-D grids Style 5 is for general 3-D grids that consist of identical planes of nodes that are stacked along a common normal axis.
3	number of cell columns axially spanning the upstream portion of the grid
4	number of cell columns axially spanning the suction surface of a blade
5	number of cell columns axially spanning the downstream portion of the grid
6	number of cell rows tangentially spanning the region between blades
7	number of cell rows tangentially spanning a blade thickness
8	number of cell layers along the span of a blade
9	number of cell layers in the tip gap
10	blade span (in same units as cascade geometry definition (e.g. axial chord))
11	ratio of cell height (spanwise) at blade end to adjacent (fluid) cell height
12	tip clearance as a percentage of the blade span
13	ratio of (axial) width of a cell at the centre of the suction surface to the width of a cell at the leading-edge end of the suction surface
14	stretch factor applied to the width of the cell at the aft end of the pressure surface
15	approximate height of cells next to the blade at the leading-edge ends of the suction surface and pressure surface given as a percent of the axial chord length
16	same as for option 15 but at downstream end
17	ratio of cell heights: (b-a)/a in Figure B.3
18	analogous to 17 but taken at leading edge of blade
19	analogous to 17 but taken at trailing edge of blade
20	analogous to 17 but taken at extreme downstream edge of grid

Table B.1: Explanation of Grid Menu Settings (part 1 of 3)

Menu Option	Description
21	<p>angle in degrees (about approximate leading-edge circle centre) separating leading-edge key points (key points 2 and 3)</p> <p>Figure B.3 shows the location of key points 2 and 3.</p> <p>The smaller this value is, the closer the leading-edge key points will be. The smaller the radius of curvature at the axially foremost position, the closer the leading-edge key points will be. The leading-edge key points are at roughly equal blade surface distances from the point at the leading edge where the tangency is perpendicular to the grid inlet angle (not the inlet flow angle). The centre of a circle representing a leading-edge circle is found by moving axially aft by the radius of curvature at the axially foremost point, from the axially foremost point.</p>
22	analogous to 21 but it applies to the trailing edge
23	<p>surface distance aft of leading-edge key points (key point 1 and key point 2) at which suction and pressure surface key points are placed</p> <p>Figure B.3 shows the location of these key points.</p>
24	grid extent axially ahead of the cascade (units of axial chord)
25	grid extent axially behind the cascade (units of axial chord)
26	grid inlet angle in positive degrees
27	grid outlet angle in positive degrees
28	outlet flow angle estimate in positive degrees
29	<p>number of cell rows next to the blade surface that are set perfectly orthogonal to the blade surface (excluding the leading-edge and trailing-edge regions)</p> <p>It is recommended that this parameter be set to zero to avoid problems with grid quality.</p>
30	<p>key point 1 angular offset for grid lines radiating from this location of the blade</p> <p>See Figure B.3 for the key point locations</p>
31	<p>distance that grid lines tend to radiate before turning, in the region of key point 1</p> <p>See Figure B.3 for the key point locations</p>
32	same as 30 but for suction side key point

Table B.1: Explanation of Grid Menu Settings (part 2 of 3)

Menu Option	Description
33	same as 31 but for suction side key point
34	same as 30 but for key point 2
35	same as 31 but for key point 2
36	same as 30 but for key point 3
37	same as 31 but for key point 3
38	same as 30 but for pressure side key point
39	same as 31 but for pressure side key point
40	same as 30 but for key point 4
41	same as 31 but for key point 4
42	load the default grid menu settings
43	load grid menu settings from the file "gridset.txt"
44	continue with creating the input files for the flow solver
45	go to the main menu The settings of the grid menu will not change.
46	specify the blade definition point number for some or all of the key points (from 1 to 4)
47	This menu option allows the trailing-edge region of all blades in the cascade to be truncated squarely. This may help an Euler solution by forcing separation points to occur at roughly the correct locations. A coarse mesh is recommended when solving inviscid flows.
48	This option controls the angle of the cut line when the trailing-edges of the blades are truncated (see menu option 47)
49	This option controls the amount of the trailing-edge region removed when the trailing-edges of the blades are truncated (see menu option 47)
66	save the current grid menu settings to the file "gridset.txt"

Table B.1: Explanation of Grid Menu Settings (part 3 of 3)

Figure B.3: Figure for Explaining Grid Menu Settings

B.6.2.3 Setting the Flow Solver Parameters

This section describes how the flow solver parameters are set for the flow solver: "NSflow.exe". It is assumed that the reader is already familiar with the basics of using Computational Fluid Dynamics (CFD) flow solvers. The previous section describes how the user may acquire a grid by accessing the "Grid Menu" of the design package's main program "Master.exe". To get to the CFD menu, the user must select menu option #8 from the main menu, then continue beyond the grid menu, the offer to load a grid, and the offer to output a grid file for viewing purposes ("gridoutp.tec"). The CFD menu displays all of the flow solver parameters and allows the user to modify them from default values

Table B.2 describes the options of the CFD menu. Menu options 1 to 5 lead to specialized submenus that contain settings for blocked regions, walls, inlets, outlets and periodic boundary conditions. Upon entering the CFD menu for the first time (since "Master.exe" began execution) all of the flow solver settings are set to default values (including the menus under options 1 to 5). The settings of the submenus (i.e. the settings that identify the location of blocked regions, walls, inlets, outlets and periodic boundary conditions) are automatically set based on the grid (that was previously acquired before entering the CFD menu). These settings are not adjusted automatically after they have been set unless the grid is reset (menu option #37). If a subsequent change is made to the grid, it is the user's responsibility to ensure that the appropriate submenu settings are adjusted, if required. One way of doing this is to save the menu options to a file (option #55), reset the menu (option #37), then load only the flow solver menu settings without loading the settings for the sub-menus (option #38).

Menu Option	Description
1	<p>menu for cell blocking</p> <p>Up to 10 blocked regions of the flow may be specified. Each region has a unique number from 1 to 10 associated with it. The directions North, South, West, East, Up, Down refer to the faces of the blocked region that face in the positive y-direction, negative y-direction, negative x-direction, positive x-direction, positive z-direction and negative z-direction respectively. In addition to being used to specify a 3-D blocked region, this convention is also used to specify 2-D surfaces, such as walls, periodic boundaries, inlets and outlets. The extents of each region are set by specifying the node index for each of the six faces. The node index starts at 1 for a node at the edge of the grid, and increases in the positive direction of the corresponding axis. Blocked regions require a wall definition for each face exposed to the flow (see option #2).</p>
2	<p>menu for walls</p> <p>Up to 66 walls may be specified, each with its own unique number from 1 to 66. Walls are impervious to flow, and may be either frictional or non-frictional. In addition, the wall temperature may be specified (in Kelvin). The location of a wall is specified as for a blocked region (see option #1). Each wall must be given the nominal direction of the fluid from the wall.</p>
3	<p>menu for inlets</p> <p>Up to 10 inlets may be specified, each with its own unique number from 1 to 10. The position of an inlet is defined as for a wall (see option #2). The nominal direction in which the fluid flows must be specified.</p> <p>Additional settings for each inlet include: whether the flow is subsonic or supersonic, the static pressure estimate (applies when the static pressure is constant across the inlet), the order of the polynomial for inlet pressure extrapolation, the stagnation pressure estimate (applies when the stagnation pressure is constant across the inlet), the stagnation temperature (applies when the stagnation temperature is constant across the inlet, velocity components for the inlet flow (applicable to uniform inlet flow only), the inlet meridional flow angle (applicable to uniform inlet flow only), the inlet swirl flow angle (the inlet flow angle of main importance for cascades).</p> <p>All angles are in degrees, temperatures are in Kelvin and pressures are in Pascals.</p>

Table B.2: Explanation of CFD Menu Settings (part 1 of 4)

Menu Option	Description
4	<p>menu for outlets</p> <p>Up to 10 outlets may be specified, each with its own unique number from 1 to 10. The position of an outlet is defined as for a wall (see option #2). The nominal direction against which the fluid flows must be specified.</p> <p>Additional settings for each outlet include: whether the flow is subsonic or supersonic, the static pressure (applies when the static pressure is constant across the outlet).</p> <p>All pressures are in Pascals.</p>
5	<p>menu for periodic boundaries</p> <p>Up to 100 periodic boundaries may be specified, each with its own unique number from 1 to 100. The position of a periodic boundary is specified in the same manner as for a wall (see option #2). The nominal direction of the side against which the fluid flows must be specified.</p>
6	selects viscous or inviscid flow
7	selects turbulent or laminar flow
8	selects compressible or incompressible flow
9	<p>selects 2-D flow or 3-D flow</p> <p>This setting must correspond to the type of grid used.</p>
10	specifies the value of the artificial compressibility parameter
11	<p>specifies two things: the reference velocity and a scaling factor</p> <p>The reference velocity must be set to the inlet velocity in metres per second.</p> <p>The scaling factor multiplies each grid coordinate so that the entire grid is stretched by this factor. The stretched grid is considered to have units of metres. Since the cascade package always produces cascades having an axial chord of unity, this scaling factor is typically set to the actual axial chord in metres.</p>

Table B.2: Explanation of CFD Menu Settings (part 2 of 4)

Menu Option	Description
12	<p>specifies the reference pressure</p> <p>The reference pressure is used for converting pressures into gauge pressures. For incompressible analyses, the reference pressure is used to calculate the reference density (the density used for the entire flow field). The reference pressure is usually set to a value close to the outlet static pressure. Care should be taken to avoid negative gauge pressures.</p>
13	<p>specifies the reference temperature</p> <p>The reference temperature is used to calculate, using Sutherland's Law, the viscosity used in laminar, incompressible analyses. For incompressible analyses, the reference temperature is used to calculate the reference density.</p>
14	specifies the number of the time steps at which the program stops iterations and outputs to disk for the last time
15	<p>controls how the run begins</p> <p>The run may start:</p> <ul style="list-style-type: none"> - with a uniform velocity field (set to zero) and a uniform pressure field (set to the outlet static pressure minus the reference pressure: resulting value must be positive) - with the velocity and pressure fields initialized using a panel method evaluation of the same cascade - by resuming from a previous run (even if the geometry is slightly different, as long as the number of grid nodes in each direction and in each grid region is the same) <p>In order to resume a run, the files "output1F" and "turbul.dat" must be present.</p>
16	initializes the time-step number (used only when resuming a run)
17	when resuming a run, specifies the name of an output file that is equivalent to "output1F" (the user normally should not change this)
18	when resuming a run, specifies the name of an output file that is equivalent to "turbul.dat" (the user normally should not change this)
19	specifies the time steps (by number) at which full output should be written
20	do not change this setting
21	do not change this setting
22	do not change this setting
23	do not change this setting
24	specifies the specific heat at constant pressure in J/kgK
25	specific heat ratio
26	<p>specifies the reference length scale in metres</p> <p>The reference length scale is output to file "output1F". It is also used by the flow solver to obtain a maximum turbulent length scale. (see the next menu option)</p>
27	specifies the maximum turbulent length scale (as a percentage of the reference length scale which is set by menu option #26)

Table B.2: Explanation of CFD Menu Settings (part 3 of 4)

Menu Option	Description
28	specifies the CFL number (convective time step limiter number)
29	specifies the diffusive time step limiter: K_d
30	specifies an artificial viscosity parameter: second order viscosity constant
31	specifies an artificial viscosity parameter: fourth order viscosity constant
32	specifies the blending constant: AVA
33	specifies whether the implicit residual smoothing parameters will be fixed or variable
34	do not change this setting
35	do not change this setting
36	do not change this setting
37	loads default menu settings and sets frictional walls, blocked regions, etc. as if entering the flow solver menu for the first time
38	loads all menu (not submenu) settings from a file called "CFDset.txt"
39	loads all menu (and submenu) settings from a file called "CFDset.txt"
40	exits the CFD menu and starts to produce files for the flow solver ("NSflow.exe")
41	exits to the main menu
42	skips the first output file
43	skips the first two output files
55	saves all menu and submenu settings to a file called "CFDset.txt"

Table B.2: Explanation of CFD Menu Settings (part 4 of 4)

After the CFD settings have been set, the user should save the settings (if desired), then select the option to continue. When the order to continue is given, the files for the flow solver will be produced.

B.6.2.4 Monitoring the Solution

After the input file has been prepared, the user should move them to the directory where the flow solver will operate (the flow solver only operates out of its own directory). The flow solver should then be started. As the flow solver runs, it outputs information about residuals to a file called "Fg_ch.2". This file is written in ASCII text format and contains eight columns of numbers. The first column indicates the time step at which the residuals were output (to the same

line in the file). The next column indicates a weighted average residual. This value should drop several orders of magnitude in a well converged run. The last column indicates the mass flux residual. This quantity should be on the order of 0.1% or less when the solution is well converged.

While the flow solver operates, the progress may be monitored by plotting the data in "Fg_ch.2". It is recommended that the average residual be plotted on a logarithmic scale from 1 to 0.00001, and that the mass flux residual be plotted on a linear scale from 0 to 15.

After a full output dump, the files "output1F" and "Residh.dat" may also be examined via a plotting package to note problems with convergence. The files are written to conform to the requirements of a commercially available plotting package: Tecplot Version: 7.5-0-0 (by AMTEC Engineering, Inc.). If this or a compatible plotting package is not available, the user may consider using a different plotting package (which may entail file conversion via a user-made program).

B.6.2.5 Using the Results

The most important product of the Navier-Stokes solver, from the point of view of the inverse design method, is the pressure-coefficient distribution. To calculate the pressure-coefficient distribution using "Master.exe", first make sure that main menu option #2 is set to the correct reference position (upstream or downstream) for the definition of pressure-coefficient. With the reference location set correctly, select main menu option #10. The pressure-coefficient distribution will be calculated using the data stored in "output1F" (the main output file of the flow solver). After the pressure distribution has been calculated, a brief message is displayed to the screen, reporting on various properties of the flow: inlet velocity, inlet static pressure, outlet flow

angle, outlet static pressure, outlet velocity. At this point, the program waits for the user to press ENTER. After ENTER has been pressed, the program writes the pressure distribution (plus the previously displayed message about flow properties) to a file called "thefoil.cas" (over-writing any previous copy). The pressure-coefficient distribution contained in "thefoil.cas" is analogous to that produced using the panel method. In this case, the pressure-coefficients are given along with the coordinates at which they apply. The coordinates correspond to the cell centres of the layer of grid cells surrounding a blade (of the infinite cascade). The coordinates are normalized based on axial chord.

The pressure distribution contained in "thefoil.cas" may be used to modify that target pressure coefficient used for optimization purposes. The next section describes this procedure.

B.7 Modifying the Target Pressure-Coefficient Distribution

The main work involved in modifying the pressure distribution is in gathering the appropriate files into the working directory. It is recommended that, in order to avoid confusion, a separate directory be used to hold the files as they are gathered. There are three required files: "thefoil.mta", "thefoil.hqp" and "thefoil.gqp". In addition, a copy of the executable ("Master.exe") needs to exist in the directory, since it can only access files in its own directory.

The first file mentioned, "thefoil.mta", is a copy of the original target pressure-coefficient distribution file. This is equivalent to the file "thefoil.udf" used as a target for the first GA run of the design process.

The second file, "thefoil.hqp", is the pressure distribution as calculated using the output file ("output1F") of the Navier-Stokes flow solver ("NSflow.exe"). To generate "thefoil.hqp", the user must run "Master.exe" in the same directory as "output1F", set the pressure-coefficient reference location appropriately (main menu option #2), then select main menu option #11, followed by (submenu) option #2.

The third file, "thefoil.gqp", is the pressure distribution as calculated using a panel method evaluation. To generate "thefoil.gqp", the user must run "Master.exe", perform a panel-method evaluation (on disk, not memory) of the cascade (the same cascade as for "output1F") using the appropriate reference location, then select main menu option #11, followed by (submenu) option #3.

Finally, to produce the modified target pressure-coefficient distribution ("thefoil.udf"), select main menu option #11 followed by (submenu) option #1. The user must not let the various versions of the target distribution file ("thefoil.udf") get confused (from iteration to iteration in the design process).

B.8 Miscellaneous Features

B.8.1 Support for Tecplot Graphics

The design package (Master.exe and NSflow.exe) produce many files that can be plotted by graphics software. Most of these files contain several lines of text that are meant to be interpreted by the commercially available plotting package: "Tecplot" (Version: 7.5-0-0, by AMTEC Engineering, Inc.). The user may edit these ASCII text files to allow different software to interpret them.

The output files compatible with Tecplot are listed and described in Table B.3.

Name of File	Software that Produced the File	Description of the File
thefoil.tec	Master.exe	cascade geometry including two blades, throat line, key points (join points, centres of leading-edge ellipse and trailing-edge circle), Bezier control points, curvature distribution
thefoil.cas	Master.exe	When developed based on a panel-method evaluation, this file contains panel junction point coordinates and the corresponding values of pressure coefficient. When based on a CFD calculation, "thefoil.cas" contains the coordinates of the cell centres for all cells next to the blade surface (normalized by axial chord), and the corresponding values of pressure coefficient.
thefoil.pjp	Master.exe	contains the panel junction points used during the last panel method evaluation
gridoutp.tec	Master.exe	contains a set of all grid node coordinates
Fg_ch.2	NSflow.exe	contains a history of residuals from a CFD calculation
Residh.dat	NSflow.exe	contains the residuals and coordinates for each cell centre in the grid at the time of the last full data output
output1F	NSflow.exe	contains the flow quantities and coordinates for each cell centre in the grid at the time of the last full data output

Table B.3: Files Compatible with Tecplot

B.8.2 Automated Features

In order to save time, two of the processes of the design package have been automated:

(a) starting the genetic algorithm and (b) preparing input files for the CFD flow solver.

(a) Starting the Genetic Algorithm

The genetic algorithm can be started or re-started automatically by selecting main menu option #4. The settings for the GA must be made available by including the file "gafdoset.txt" in the working directory. Depending on the case, several other files may also be required in the working directory. If a run is to be resumed, the file "[case name].rec" must be included and the files "[case name].tec" and "[case name].chr" should be included (for complete output). If customized parameter limits are to be used, the file containing them, "default_.plm", must be included.

(b) Preparing Input Files for the CFD Flow Solver

Input files for the CFD flow solver ("NSflow.exe") may be generated by performing the procedure once manually (from main menu option #8), and saving the grid settings and flow solver settings to files "gridset.txt" and "CFDset.txt" respectively. When these files are included in the working directory, the user may automatically create a set of flow solver input files by selecting main menu option #13.

B.8.3 Support for Parallel Processing using a Computer Network

In order to use more than one computer to divide the task of performing fitness evaluations for the GA, the user must create a working directory that is available to more than one computer via a network (Essentially, a directory is required that is available to more than one CPU.). A copy of "Slave.exe" should be placed in the working directory. The GA must be started with network capability enabled, using GAFDO menu option #16. The number of computers (more properly, CPUs) that will be used to perform fitness evaluations must be entered as "the number of slaves to scan", by menu option #17. The next menu option (#18) must be set to the number of cascades that will be processed per transaction between a slave program and the master program. This number must go evenly into the population size. Menu option #19 affects the rate at which the files containing output from the slaves are scanned. It should be set to scan files quickly.

After the GA is started, multiple copies of "Slave.exe" should be started. Each slave executable will claim its unique identification number (by writing a file called "slave.###" to disk, where ### is the unique slave number), then ask the user for some of the same information that was provided to "Master.exe". If a slave executable is stopped, its slave identification number cannot be reclaimed until the file "slave.###" has been deleted. Slave identification numbers must range from 1 to the total number of slaves.

Information transfer between the executables is accomplished by reading and writing files to the working directory. Because of time delays in passing information, using multiple computers connected by a network is not recommended unless the population size is large and each slave is given a large number of cascades to process per transaction.

B.8.4 Navier-Stokes and Euler Solutions for General Problems

The CFD code "NSflow.exe" can solve flow problems using general 2-D H-mesh grids. In order to use a generalized mesh, a 2-D mesh must be created using a separate application. The GAFDO package is capable of loading files of the following formats: (1) a table of xyz coordinates and (2) an output file from a commercial package: CFD 2000 version 3.3 by Adaptive Research.

A file containing xyz coordinates that may be loaded by "Master.exe" must be an ASCII text file. The grid node points must be listed in x, y, z format, with each coordinate separated by spaces and/or tabs. The order of nodes listed in the file must be such that the index corresponding to the x-coordinate cycles the fastest and the index corresponding to the z-coordinate cycles the slowest. A 2-D grid must have two planes of nodes specified: the first with $z = 0$ and the second with $z = 1$. Although the planes of nodes should differ only by the z-coordinate, it is the second plane (with $z = 1$) that defines the x and y coordinates for the 2-D grid. 3-D grids may also be imported by having more than two identical planes of nodes in the file. As for the 2-D case, the x and y coordinates are taken from the last plane listed in the file. The present version of "Master.exe" is unable to produce general 3-D grids (i.e. with non-identical planes of nodes, or with non-planar node distributions).

Grids from CFD 2000 may be 2-D or 3-D, but in the case of 3-D grids, only identical node planes may be used.

Before loading a grid from a file, the appropriate style of grid must be set in the grid menu (The grid menu is accessed by menu option #8 from the main menu). The style controls how cell blocking, walls, periodic boundary conditions, inlet and outlet conditions are set by default when either entering the flow solver menu for the first time or resetting the flow solver menu options (flow solver menu option #37). The style of the grid also affects whether the grid is 2-D or 3-D.

It is necessary to set the number of cells in the grid prior to loading a grid from a file. Grid menu options 3 to 9 are used to set the number of cells in the grid. If the grid does not apply to a turbine cascade, only the sum of the number of cells in each direction is important. For example, if the grid has a total width of 50 and a total height of 40, the user may specify the following:

(for the nominal x-direction)

50 cell columns upstream of blade (even though there is no blade)

0 cell columns along the blade

0 cell columns downstream of blade

(for the nominal y-direction)

40 cell rows between blades

0 cell rows within a blade

Only even numbers are allowed for the number of cell columns along a blade, the number of cell rows between blades, and the number of cell rows within a blade.

B.8.5 Inverse Design based on Geometry

In order to inversely design a cascade based on geometry (with the exception of the blade spacing parameter), select main menu option #15. The program asks the user for the name of a file that will serve as the target geometry. This file must have the same format as "thefoil.def" (To obtain a copy of "thefoil.def", run the cascade generator with the disk usage option set to use the disk.). The GAFDO menu then appears. The fitness function will be one for geometric comparison, and is defined as follows:

$$f(g_1, g_2, \dots, g_M) = \left[K - \sqrt[4]{\frac{\sum_{i=1}^N (\Delta y / b_x(g_1, g_2, \dots, g_M, i))^2}{N}}, 0 \right]$$

where: f is the dependent variable;

$\Delta y / b_x$ is the difference between the i^{th} target value and the corresponding actual value of y / b_x ;

$g_1, g_2, \dots, g_M$ are the M parameters of the cascade;

N is the number of target points around the given blade profile;

K is a constant set to 3.

The terms of the sum apply to all of the points of the target geometry. When necessary, linear interpolation is used to find a point on the design having the same axial coordinate as the corresponding point on the target geometry.

B.8.6 Generic GAFDO

B.8.6.1 Compiling and Executing the Main Program

(i) The required Fortran files of the generic version of GAFDO are:

`plainrunall.for` - the main program module and supporting subroutines

- creates the menu system and calls all subroutines

including the genetic algorithm and hillclimber

`plaingafdo.for` - the genetic algorithm subroutine and supporting subroutines

`plaingafdosettings.for` - a supporting subroutine for the genetic algorithm

(it stores/retrieves the GA settings to/from disk

and contains a menu system for displaying/editing

the settings)

`examplefit.for` - subroutines related to the fitness function

`plainhillclimb.for` - the gradient-based search method subroutine

Using the G77 Fortran compiler:

(ii) type "`g77 -c -O3 *.for`" at the prompt to compile each of the Fortran files.

-c means compile (to produce object files (*.o))

-O3 means optimize (NOT "-03")

"*.for" selects all Fortran files to be compiled.

(alternatively, if only one Fortran file has been modified since the last executable was produced, only that file needs to be compiled. For example, if "plaingafdo.for" was the only file modified, then "g77 -c -O3 plaingafdo.for" would be used to compile just the file "plaingafdo.for")

(iii) type "g77 -o master.exe *.o" to link the object files into an executable called "master.exe"

(The files, as they are provided, have been compiled and linked to form "master.exe".)

(iv) To run the code, copy "master.exe" to a suitable folder and execute it. The menu system should be self-explanatory.

B.8.6.2 Compiling and Executing the Slave Program (used for Network operation)

In order to use more than one computer to process the fitness evaluations, a set of computers connected by a network connection is required. Each machine should be able to work with files in the same, common, working directory. One of the computers should run "master.exe" while the remaining computers should run a copy of "slave.exe". Network operation of the program should be used when a typical fitness evaluation takes more than 1 second to perform, and the population size is greater than 100. Note that the initial population

will be performed by the main executable, but for the remaining generations, all of the fitness evaluations are performed by the slave executables.

To produce "slave.exe", the file "slave.for" and any other files containing subroutines involved in the fitness evaluation, must be compiled. The resulting object files should then be linked. First, make sure that the fitness-function-related subroutines are identical to those of the main executable ("master.exe"). Also, make sure that the main subroutine from "master.exe" is not included in any of the files (to be compiled and linked).

The following steps are used to produce the slave executable:

Step 1: Go to the folder "plain slave".

Step 2: Compile and link the Fortran files in that folder by typing:

```
g77 -c -O3 *.for", then "g77 -o slave.exe *.o
```

After the executable "slave.exe" has been produced, place it in the same folder as "master.exe" before performing a run. It is best to start "master.exe" first, then "slave.exe" to avoid confusion with the questions posed by "slave.exe".

Note: When "slave.exe" runs, it writes a unique file (slave.###) to the directory. This is its way of claiming a slave number (the "###" extension). The slave files should be erased manually before starting a run. If a slave is started and an error is made while entering the required information, close the application and erase the corresponding slave.### file, then open another copy of slave.exe.

B.8.6.3 Demonstration of Generic GAFDO

The fitness function in the demonstration has two independent parameters. The first two genes (out of 26) are used to encode these two parameters.

The peak value was found by the test GA run in the folder "test ga run". In that folder:

"gafdoset.txt" is the file containing the settings for the GA.

"testcase.chr" is the file containing the record of best members.

"testcase.out" is output showing general information about the run.

"testcase.rec" stores the memory state so that the GA can be resumed.

"testcase.tec" produces a TecPlot file showing progress.

To view the fitness function within the pre-set domain limits, follow these steps:

Step 1: At the MS-DOS prompt, type "qbasic".

Step 2: Open the file "showfit.bas" and run it (type: "Alt" then "R" then "S"). This

BASIC program displays the fitness landscape along with the point of interest (represented by a circle).

Compare the BASIC code with the file "testcase.chr". It can be seen that the latest fitness improvement occurred in generation 47 with parameter#1 = 0.954838693 and parameter#2 = 1.2630496. The same numbers appear in the BASIC code for the point indicated by a circle.

Because the setting file "gafdoset.txt" is provided, the menu settings that controlled this test case can be observed by loading this file from within the menu system of the code (menu option 12, then menu option 21).