AERODYNAMIC SHAPE OPTIMIZATION OF 3D GAS TURBINE BLADE USING DIFFERENTIAL EVOLUTION METHOD

By

DARSHAK JOSHI

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

Of the Requirements for the Degree of

MASTER OF SCIENCE IN AEROSPACE ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2010

ACKNOWLEDGEMENTS

To start, I would like to thank My Professor Dr. Brian Dennis who helped me a lot to make this work successful .He guided me through the whole project and helped me to find my mistakes. This helped me to achieve the complete transition into Graduate Student. Another important person whom I liked to thank is James Ruiz. Though I have never met him personally, but his previous work on Turbomachinery cooling has helped me a great deal to find the solution of my problems during the Thesis.

I would also like to thank you the individuals who, in many form, have contributed to my work and success in UT-Arlington. To start with my lab mates: Kamal, Travis, Katie, Harsh and Monal. Both Monal and Harsh are my friends who have provided me assistance. Their help is always appreciated.

I also like to thank my family for their endless support and I like to dedicate this work to my late father.

Last but not the least I would like to extend my gratitude towards my committee members' for their co-operation. I once again thank Dr Dennis for his endless support.

April 16, 2010

ABSTRACT

AERODYNAMIC SHAPE OPTIMIZATION OF 3D GAS TURBINE BLADE USING DIFFERENTIAL EVOLUTION METHOD

Darshak Joshi, M.S

The University of Texas at Arlington, 2010

Supervising Professor: Brian H. Dennis

This paper outlines the aerodynamics shape optimization process of a 3D VKI gas turbine blade using Computational Fluid Dynamics (CFD) and differential evolution optimization method. The main objective of this work is to improve outlet total pressure for better aerodynamic performance. This benefit is achieved by optimizing the stacking line path of the turbine blade in 3D annular cascade for which higher exit total pressure is achieved. The optimization process starts with generating a 3D annular cascade with a boundary domain and meshing it using GAMBIT and is then solved in FLUENT using viscous flow analysis. The original stacking line is defined as a B –spline with its end points fixed and middle points free to move in translational direction. These middle points work as design variables whose position needs to be changed for improved exit total pressure which is a desired parameter. The turbine blade cascade is analyzed for only aerodynamic performance. The optimization analysis is an iterative process using a parallel computing system.

The shape optimization process runs iteratively with the CFD code Fluent and DE algorithm until the higher total pressure at outlet is achieved by optimizing the stacking line path the turbine blade in the annular cascade. Here the optimization process of the 3D turbine blade cascade is presented.

TABLE OF CONTENTS

APPENDIX

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

CHAPTER 1

INTRODUCTION

This report presents the aerodynamic shape optimization process of a 3D VKI turbine blade cascade by coupling a commercial CFD code with differential evolution algorithm. It relies on the interaction between the commercial CFD software FLUENT, a blade geometry generator GAMBIT and differential evolution (DE) algorithm with some design and optimization constraints. Here, most of the current 3D turbine blades are stacked in a perpendicular line from 2D geometry. This straight stack path is modified for improved exit total pressure .Optimization process runs iteratively with CFD code and DE optimization algorithm on parallel computing system.

The 3D turbine cascade with straight span is analyzed using FLUENT to measure total pressure difference across the cascade. Then, the turbine blade geometry is optimized using DE code for improved outlet total pressure. Optimization process also checks the usability of DE code in turbine blade problems. The reason to take this approach is to use simple optimization process for proof of concept of using DE method for shape optimization.

1.1 Thesis Contents

Chapter 2 gives information about the background of the related work done on the same topic. It also mentions some earlier shape optimization work using evolutionary strategies and specially evolution algorithm method. It also gives the information behind stacking line and secondary flow formation in turbine cascade.

Chapter 3 provides the governing equations of the CFD as were applied in the turbine blade analysis. It also mentions the FVM method and discretization techniques of this method. Basic information about flow analysis software FLUENT and Grid generation software GAMBIT is also reported.

Chapter 4 contains the detailed information about differential evolution code. It presents the theory behind the differential evolution code and their functions and constraints and how the DE optimization code is applied in this optimization analysis.

Chapter 5 outlines the methodology of analysis process starting from creating geometry using Gambit, viscous flow analysis using Fluent and final optimization process using DE code. It also links all the functions, solvers and intermediate processes required in the optimization sequence. The set up of the final analysis, the test model and all its test parameters are also content of chapter 5.

Chapter 6 discusses the results of the original annular turbine blade and optimized turbine blade geometry. It also explains the physical reason behind the results.

CHAPTER 2

SCIENTIFIC BACKGROUND AND MOTIVATION

Turbo machinery design is a field where designers find resurgence in the development and usage of optimization techniques in the scientific community. Previously blade design was manually optimized using trial and error approach by design experts relying on their expertise, experience and intuition. But such conventional through flow design method has nearly reached its limits. Nowadays, designers use a revolutionary approach which includes 3-D Navier-Stokes equations with efficient numerical design optimization method. Shape optimization method is one of the techniques, which is generally used to improve aerodynamic performance of the turbo machinery components such as turbine and compressor. In the last 15 years, through development and improvement in computational powers has provided an ideal platform for the optimization process to develop in this direction. Computational Fluid Dynamics (CFD) works as a core for optimization design process and covers all important basic fundamental aspects. The development of advanced optimization techniques handling large numbers of design parameters can be seen in the literature. Also, the application of these techniques on optimization of turbine blades and proposing new methods for shape optimization for gas turbine blades also can be seen in literature. The invention of splines in the late 60s along with introduction of advanced geometrical description systems generated from Beizer- and spline-curves in the 80s has started the way for non-convention profile forms for gas turbine blades and so optimization techniques for these blades have also started in 90s.This chapter will give literature review on shape optimization techniques and some basic information on stacking line and secondary flows.

2.1 Literature review on shape optimization techniques

The innovations in computational technology are playing an important role in aerodynamic design of turbine blades. The use of computational fluid dynamics techniques in design is mostly employed on its analysis phase. First a specific design configuration is evaluated to get its aerodynamic performance properties and then geometry is modified using shape optimization techniques to get an improved performance for that specific property. The detailed discussion of turbo machinery design system was given in Ref. 1. Recently there has been a lot of progress in aerodynamic shape optimization.

Shape optimization process using numerical techniques allows the designer to generate the exploration of design space to get the desired solution. It can improve shape of the geometry using one or many design objectives. For example, one possible design objective is to minimize flow losses in turbine cascade, which can be measured as reducing total pressure loss through proper reshaping of the blade cascade. This automated shape design is completed by coupling a CFD flow simulation code with numerical optimization method.

Numerical optimization techniques can optimize the geometry with some specified constraints. [2, 3, 4] The best known of these numerical methods are the genetic algorithms [5] and evolutionary algorithms [6, 7]. These methods help to optimize specific parameter by giving constraints to other properties related in the geometry. Lampart et al [8] used the steepest gradient method using sweep and straight lean for steam turbine optimization. They get 1.7 % stage efficiency gain which shows the suitability of selected parameters. Harrison [9] and Denton and Xu [10] have calculated the effects of various 3D airfoil blades stacking in turbine cascades and stages. Lotfi et al. [11] have parameterized the camber line of 5 profiles with Bezier curves; each has six control points which can add 30 design variables and sweep and lean were also parameterized by parabolic curve. Ahn et al. [12] have used a parabolic profile for staking line. These are the some previous work of shape optimization techniques to optimize the stack line for turbine blade cascades. For this work differential evolution algorithm is used for the shape optimization of annular turbine cascade. More information on DE is given in next section.

2.2 Differential Evolution in shape optimization problems

DE is a simple and effective evolutionary strategy (ES) that has became effective in providing solutions for difficult aerodynamic shape optimization problems. Shape optimization in turbo machineries is a problem with many possible constraints, but differential evolution method can handle all constraints and gives good accurate results for shape optimization problems. So, these differential evolution algorithms are widely applied in turbo machinery problems. [13, 14, 15] T. Rogalsky, R. W. Derksen and S. Kocabiyik [16] used differential evolution techniques for aerodynamic shape optimization. Ueno A. and Dennis B [14] also used DE to optimize flapping motion of the airfoil. Madavan N [17] used DE for design optimization of turbomachinery.

In this work, a practical shape optimization strategy for the turbine blade of annular cascade is presented. As, the aim of the work is to modify the stacking line of the 3D VKI airfoil for higher outlet total pressure, several numerical optimization methods are used for blade geometry. But for this problem differential evolution algorithm (DE) is used to optimize the stacking line path of the 3D turbine blade for higher aerodynamic performance. Application of DE in parameterization of stack line is given in chapter 4.

2.3 Introduction to gas turbine annular cascade

For high-performance gas turbine engines, compressor rotor/stator stages must provide high efficiency and compression ratio. Therefore compressors have been developed and optimized for many years to improve aerodynamic performance. Optimization of gas turbine stator blade in the annular cascade is considered for the work. High velocity required to obtain high pressure ratios in annular turbine cascade develops secondary flows and their interaction with boundary layers induce vortex in near wall region. It will cause total pressure losses in the cascade and so negatively influence overall efficiency of the cascade. This affects aerodynamic performance of the turbines in gas turbine engines. However cascade performance can be improved by optimizing how blades are stacked in radial direction. The application of sweep and lean on 3D airfoil blades is one of the important technical evolutions to improve the aerodynamic performance of the annular cascade [9]. Here also the improvement in aerodynamic performance is obtained by modifying the stacking path of turbine blade.

2.4 Importance of stack line in 3D turbine blade cascade

This work depends on the optimization of airfoil blade stack line in the annular cascade of gas turbine. The VKI airfoil is chosen as a base airfoil shape. Normally, airfoils are stacked from root to tip following path of the stack line. So, stacking line controls the shape of the 3D blade geometry.

Different 3D airfoil blade shapes are generated using different stack line paths. As stack line controls the shape of the 3D turbine stator blades and hence its position affects the pressure loss in the cascade. So, the aim is to find the best shape of the stack line for turbine blade that will produce an efficient 3D blade for the annular cascade. In this work, the 3D blade with lower total pressure loss across the cascade is considered as efficient blade. To reduce the pressure loss, exit total pressure is improved by mode of optimization. Figure 2.1 shows the airfoil blade with straight stacking line.

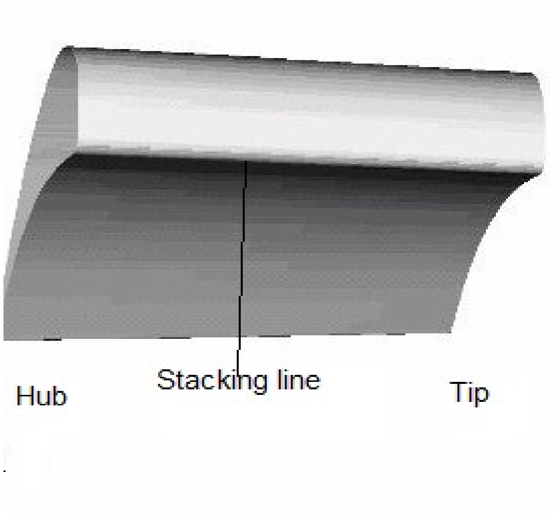


Figure 2.1 Turbine blades with Straight Stacking line

The blade geometry is usually designed by several two –dimensional blade sections with various radical locations joined by a straight stacking line in 3rd direction. Primary objective of this work is to minimize the adverse effects of three dimensional flows such as secondary flows and pressure losses across the turbine cascade. It also improves the performance by carefully redesigning the stacking line using DE with CFD analysis.

2.5 Secondary flows in aerodynamic performance

Secondary flow formation plays an important role in pressure losses in turbine cascade. This work is based on removal of secondary flows generated during flow analysis process in the turbine cascade for improvement in aerodynamic performance. So, it is important to understand the physical explanation behind the secondary flow formation and importance of its removal from the cascade to decrease pressure loss.

2.5.1 Secondary Flow

When the flow is turned by a blade row in turbine cascade, the flow far away from the end wall can be termed as two-dimensional. The boundary layer flows along the end walls has span wise velocity gradients. When this flow is turned, transverse velocity components will be produced. The 2D flow is known as primary flow and the 3D effect near the end walls is called as secondary flow. In real, secondary flow can be very large. Secondary flow development in the cascade will result drop in efficiency. Several experiments show that secondary vorticity in blade passages is concentrated in discrete vortex cores, which can result in the high shear and hence high total pressure loss. Moreover, annular cascade has end wall boundary layers due to high turning flow angle. These end wall boundary layers generated due to viscous inlet boundary layer and secondary flows, which significantly affect on the total pressure loss at the outlet.

2.5.2 Formation of secondary flows

In gas turbine cascades, slower moving flow in the boundary layers on the blade end walls and cross -passage pressure gradients produce secondary flows. Boundary layer has same pressure gradient as free stream region, but fluid is traveling more slowly in the boundary layer region, it is turned sharply by pressure gradient and so called ‘overturning’. When this happens, the flow turns into vortex which is generally formed in suction side. A simple schematic model of the secondary flows is shown in Figure 2.2 [Takeishi 1989, 18]. The mixing of secondary flows with free-stream flow leads to some pressure losses and hence reduce aerodynamic performance. Secondary flow is mostly a problem with turbine cascades because of large cross - shock waves and by tip leakage passage pressure gradients and larger flow turning angle. These flows also occur due to strong flow.

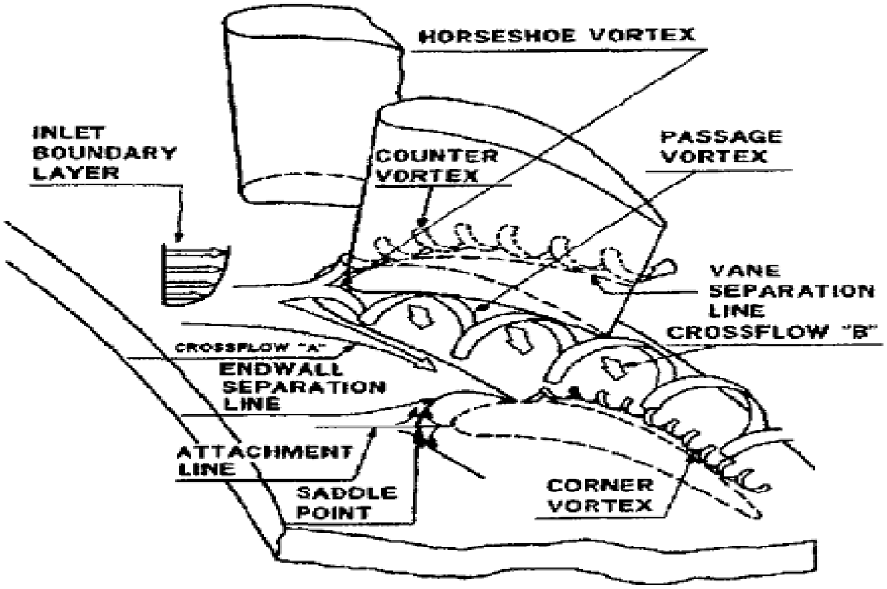


Figure 2.2 Secondary flows in Turbine Cascade [Takeishi 1989, 18]

Our aim is to reduce the strength of the secondary flows and vortex generated in the boundary layer region. This will help in smoothing the flow in the boundary layer region and reduces total pressure loss in the annular cascade. Modification of 3D airfoil blade from straight blade to lean or bow towards the tip side of the cascade will help in reduction in the formation of secondary flow and vortexes in the boundary layers. So modifying stacking line path of 3D turbine blade in annular cascade will give higher aerodynamic performance. This is achieved by optimizing the stack line of the blade by coupling between CFD code and differential evolution algorithm to get the improved exit total pressure.

CHAPTER 3

COMPUTATIONAL FLUID DYNAMICS FOR TURBINE BLADE ANALYSIS

Computational Fluid Dynamics, preferably known as CFD predicts quantitatively, when fluids flow with the combinations of simultaneous flow of turbulence, heat, mass transfer, phase change, chemical reaction and mechanical movement and displacements of surrounding solids. CFD is one the branches of the fluid mechanics which has various numerical methods and algorithms for finding and analyzing solution of fluid related problems with the help of computers. They simulated the interaction of the fluids with different complex surfaces used in engineering with numerical methods. Even with simplified equations and high-speed computers, sometimes only approximate solutions can be achieved in many cases. Nowadays research is going on to improve the accuracy and speed of the simulation processes which helps to solve complex problems like transonic or turbulent flows with the help of high-speed supercomputers.

The most fundamental consideration in CFD is how continuous fluid is treated in a discretized fashion numerically. One common method used for this is to divide computational domain into small cells to make a volume mesh or grid and a suitable solver is applied to solve the equations. The basic fundamental equations behind all CFD problems are Navier-Stokes Equations, which govern almost every single-phase fluid flow. These equations are mainly partial differential equations which define mass, moment and energy flow conservation through a medium like fluid or solid. These governing equations are used to measure various flow properties of 2D area or 3D control volume of interest using finite area or finite volume method.

3.1 Governing equations for blade cascade model

The CFD governing equations shows the mathematical representations of the conservation laws of physics. It is derived in the numerical form for calculating the nature of fluid flow. For the flow analysis of a model, various equations are required depending on the type of flow conditions experienced and requirement of measurement of engineering quantities. For turbine blade model, velocity and pressure are evaluated with viscous effects of fluid flow. This will required flow and turbulence equations for fluid. As compressible flow model is selected for annular cascade, ideal gas law is selected for fluid in the volume, which requires solution of energy equation. Here are descriptions of the governing equations used in the numerical analysis. Derivation of each equation can be referenced in the references sections of the report. [19]

3.1.1 Conservation of mass equation

Mass conservation law states that neither mass can be created or nor mass can be destroyed, it's only transferred. By applying these laws to 3D control volume, we can get the following conservation equation for 3D, steady and compressible flow which is derived by Tu et al. [19].

$$\frac{\partial \rho}{\partial t}+\frac{\partial u\rho}{\partial x}+\frac{\partial v\rho}{\partial y}+\frac{\partial w\rho}{\partial z}=0 \qquad 3.1$$

3.1.2 Momentum equations

Newton's second law of motion states that sum of forces acting on a fluid volume must be same for both input and output. Momentum equations in CFD are also derived using the same principle. Here the momentum equations for a fluid volume show transfer of momentum for 3D, compressible, Newtonian field.

$$\frac{\partial \rho u}{\partial t}+\frac{\partial \rho uu}{\partial x}+\frac{\partial \rho vu}{\partial y}+\frac{\partial \rho wu}{\partial z}=-\frac{\partial p}{\partial x}+\mu\frac{\partial^2 u}{\partial x^2}+\mu\frac{\partial^2 u}{\partial y^2}+\mu\frac{\partial^2 u}{\partial z^2}+S_x \qquad 3.2$$

$$\frac{\partial \rho}{\partial t}+\frac{\partial \rho uv}{\partial x}+\frac{\partial \rho vv}{\partial y}+\frac{\partial \rho wv}{\partial z}=-\frac{\partial p}{\partial y}+\mu\frac{\partial^2 v}{\partial x^2}+\mu\frac{\partial^2 v}{\partial y^2}+\mu\frac{\partial^2 v}{\partial z^2}+S_y \quad 3.3$$

$$\frac{\partial \rho w}{\partial t}+\frac{\partial \rho uw}{\partial x}+\frac{\partial \rho vw}{\partial y}+\frac{\partial \rho ww}{\partial z}=-\frac{\partial p}{\partial z}+\mu\frac{\partial^2 w}{\partial x^2}+\mu\frac{\partial^2 w}{\partial y^2}+\mu\frac{\partial^2 w}{\partial z^2}+S_z \quad 3.4$$

Momentum equations contain dynamic viscosity (μ) which is defined for the fluid at the inlet.

3.1.3 Conservation of energy equation

Energy conservation laws are derived from the laws of thermodynamics. Conservation of energy equation states that the rate at which the total energy changes within a control volume is equal to the net rate of input energy in the system plus net rate of work done on the system by forces. As we consider only steady flow effects in turbine cascade, changes in total energy of the control volume due to only viscous forces and pressure forces are considered. Energy changes due to heat transfer are not related in this problem. By applying this law to control volume the 3D energy conservation equations defined by Tu et al [19]

$$\frac{\partial E}{\partial t}+\frac{\partial uE}{\partial x}+\frac{\partial vE}{\partial y}+\frac{\partial wE}{\partial z}=\frac{1}{\rho C_p}\frac{\partial p}{\partial t}+\frac{\Phi}{\rho C_p} \quad 3.5$$

The effects of viscous forces on energy are represented by the dissipation function (ϕ).This source is due to mechanical work done on the fluid that is converted into energy. But considering the effect of viscosity, the shear stresses on the control volume are converted into velocity gradients and so the dissipation function is expressed as:

$$\Phi=\left(\frac{\partial u\tau_{xx}}{\partial x}\right)+\left(\frac{\partial v\tau_{yx}}{\partial y}\right)+\left(\frac{\partial w\tau_{zx}}{\partial z}\right)+\left(\frac{\partial v\tau_{xy}}{\partial x}\right)+\left(\frac{\partial v\tau_{yy}}{\partial y}\right)+\left(\frac{\partial v\tau_{zx}}{\partial z}\right)+\left(\frac{\partial w\tau_{zx}}{\partial x}\right)+\left(\frac{\partial w\tau_{yz}}{\partial y}\right)+\left(\frac{\partial w\tau_{zz}}{\partial z}\right) \quad 3.6$$

3.1.4 Turbulence Modeling

Most of turbo machinery problems consider turbulence as one of the important engineering property. Turbulence influences the aerodynamic performance of the turbine cascades; therefore its consideration is critical in turbine flow analysis. Turbine is a flow behavior characterized by chaotic changes in the mostly viscous flow. Turbulence can be seen by rapid variation of pressure and velocity in space and time, due to various type of flow regime like low momentum diffusion and high viscous convection. Turbulence is the reason for the making of rotational motion of flow or “turbulence eddies” as commonly known on different size of length scales. Inlet flow with viscous boundary layer will cause vortexes and secondary flow at tip wall region that will decrease total pressure in the cascade. So, it is important to model turbulence for turbine blade flow model.

This analysis has two equations K-epsilon realizable model because of its proven compatibility and accuracy in turbo machineries for confined fluid flow. It also gives good results for flow involving boundary layers under strong pressure gradients, separation and recirculation. Turbulence equations for k-epsilon model are formulated using Navier –Stokes equation for annular turbine cascade.

3.1.4.1 Variables defining the Turbulence

Turbulent oscillations for any point in fluid is measured by time averaged component (ū) and the time oscillation component

(u (t)) = ū + u’ (t)

The same assumption can be applied to other variables

For Pressure p: (p (t) = p¯ + p’ (t) and density ρ (t) =ρ + ρ’ (t)

3.1.4.2 Equations defining the Turbulence

In Reynolds averaging, the solutions variables of Navier-Stokes equations are decomposed into mean and fluctuating components. These approximated components help in reconstructing conservation equations for time and fluctuating values.

$$\frac{\partial \rho \overline{uu}}{\partial x}+\frac{\partial \rho \overline{vu}}{\partial y}+\frac{\partial \rho \overline{wu}}{\partial z}=-\frac{\partial \bar{p}}{\partial x}+\frac{\partial}{\partial x}\left[v\frac{\partial \bar{u}}{\partial x}\right]+\frac{\partial}{\partial y}\left[v\frac{\partial \bar{u}}{\partial y}\right]+\frac{\partial}{\partial z}\left[v\frac{\partial \bar{u}}{\partial z}\right]+Stresses+S_x \qquad 3.7$$

$$\frac{\partial \rho \overline{uv}}{\partial x}+\frac{\partial \rho \overline{vv}}{\partial y}+\frac{\partial \rho \overline{wv}}{\partial z}=-\frac{\partial \bar{p}}{\partial y}+\frac{\partial}{\partial x}\left[v\frac{\partial \bar{v}}{\partial x}\right]+\frac{\partial}{\partial y}\left[v\frac{\partial \bar{v}}{\partial y}\right]+\frac{\partial}{\partial z}\left[v\frac{\partial \bar{v}}{\partial z}\right]+Stresses+S_y \qquad 3.8$$

$$\frac{\partial \rho \overline{uw}}{\partial x}+\frac{\partial \rho \overline{vw}}{\partial y}+\frac{\partial \rho \overline{ww}}{\partial z}=-\frac{\partial \overline{p}}{\partial z}+\frac{\partial}{\partial x}\left[v\frac{\partial \overline{w}}{\partial x}\right]+\frac{\partial}{\partial y}\left[v\frac{\partial \overline{w}}{\partial y}\right]+\frac{\partial}{\partial z}\left[v\frac{\partial \overline{w}}{\partial z}\right]+Stresses+S_z \qquad 3.9$$

These conservation equations are known as the Reynolds Averaged Navier Stokes Equations (RANS) .They are similar to laminar Navier Stokes equations with some additional terms for 3D momentum equations. These additional terms are generally known as Reynolds Stresses and are shown in equations 3.8, 3.9 and 3.10. It was formulated by Boussinesq (1868) and Reynolds stresses are formulated using average deformation rate as follows:

$$\text{Stresses} = \begin{aligned}
\overline{u'u'} &= -2\nu_T \frac{\partial \overline{u}}{\partial x} + \frac{2}{3}k \\
\overline{v'v'} &= -2\nu_T \frac{\partial \overline{v}}{\partial y} + \frac{2}{3}k \\
\overline{w'w'} &= -2\nu_T \frac{\partial \overline{w}}{\partial z} + \frac{2}{3}k \\
\overline{u'v'} &= -\nu_T\left(\frac{\partial \overline{v}}{\partial x} + \frac{\partial \overline{u}}{\partial y}\right) \\
\overline{u'w'} &= -\nu_T\left(\frac{\partial \overline{u}}{\partial z} + \frac{\partial \overline{w}}{\partial x}\right) \\
\overline{v'w'} &= -\nu_T\left(\frac{\partial \overline{v}}{\partial z} + \frac{\partial \overline{w}}{\partial y}\right)
\end{aligned} \qquad \text{for}\begin{Bmatrix} x \\ y \\ z \\ x,y \\ x,z \\ y,z \end{Bmatrix} \qquad 3.10$$

Reynolds Stresses also has two additional terms for the turbulent viscosity (μ) and turbulent kinetic energy (k). Turbulent kinetic energy is calculated using equation 3.12

$$k_{local}=\frac{1}{2}\left(u'^2+v'^2+w'^2\right) \qquad 3.11$$

Turbulent viscosity is calculated using local turbulent kinetic energy and turbulent dissipation (ε) values.

$$v_T = \frac{\mu_T}{\rho} = \frac{C_\mu k^2}{\varepsilon} \qquad 3.12$$

Realizable k-epsilon model (RKE) is selected mainly because of the flow regime which separates and recirculates the boundary layer flows which give better results than the Standard k-epsilon model (SKE). The Transport equations are different from standard k-epsilon model in the generation of turbulent kinetic energy (k) and dissipation rate (ε). The main numerical difference is in the value of C_μ term for calculating turbulent viscosity. In SKE model this term remains constant while in RKE model changes depend on the value of strain flow. For turbo-machinery problems, the RKE model has shown more accurate result in the production of k and ε.

Below turbulence transport equations are mentioned for 3D turbine cascade flow. G_k term is for production of the kinetic energy from Mean velocity gradients and G_b represents the kinetic energy production due to buoyancy. The Y_M term accounts for the effects of compressibility in the k-epsilon model.

$$\rho\frac{Dk}{Dt}=\frac{\partial}{\partial x}\left[\left(\mu+\frac{\mu_T}{\sigma_k}\right)\frac{\partial k}{\partial x}\right]+\frac{\partial}{\partial y}\left[\left(\mu+\frac{\mu_T}{\sigma_k}\right)\frac{\partial k}{\partial y}\right]+\frac{\partial}{\partial z}\left[\left(\mu+\frac{\mu_T}{\sigma_k}\right)\frac{\partial k}{\partial z}\right]+G_k+G_b-\rho\varepsilon-Y_M \qquad 3.13$$

$$\rho\frac{D\varepsilon}{Dt}=\frac{\partial}{\partial x}\left[\left(\mu+\frac{\mu_T}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x}\right]+\frac{\partial}{\partial y}\left[\left(\mu+\frac{\mu_T}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial y}\right]+\frac{\partial}{\partial z}\left[\left(\mu+\frac{\mu_T}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial z}\right]+\rho C_1 S\varepsilon-\rho C_2\frac{\varepsilon}{k+\sqrt{v\varepsilon}}+\frac{\varepsilon}{k}C_{\varepsilon 1}C_{\varepsilon 3}G_b \qquad 3.14$$

$$G_k=\mu_T S^2 \qquad 3.15$$

$$\eta=\frac{Sk}{\varepsilon} \qquad 3.16$$

$$C_1=\max\left[0.43,\frac{\eta}{\eta+5}\right] \qquad 3.17$$

$$Y_M=\frac{\rho\varepsilon 2k}{a^2} \qquad 3.18$$

$$S=\sqrt{S_{ij}S_{ij}} \qquad 3.19$$

These Turbulent equations has been formulated for compressible flow as given in reference by Tu et al [19].It also gives some value for turbulent coefficients for a wide range of turbulent Flows as shown Reference [21].

$$\sigma_k=1.0,\sigma_\varepsilon=1.2,C_{\varepsilon 1}=1.44,C_2=1.92$$

We do not consider gravity in this turbine blade flow problem so buoyancy term (G_b) in the turbulence term equation is also neglected. Moreover in the steady state analysis problem local acceleration terms are also neglected.

$$G_b = \frac{\partial k}{\partial t} = \frac{\partial \varepsilon}{\partial t} = 0 \qquad 3.20$$

3.2 Finite volume method

The Finite volume method is the most used method in computational fluid dynamics (CFD).The FVM is a method of evaluating partial differential equation based on control volume integration similarly as the finite difference method, but the FVM method works well with both structured and unstructured grid. The FVM uses gauss divergence theorem to convert the volume integrals into surface integrals and hence the value from cell faces into cell surfaces. These can be done by using 1st order scheme or 2nd order scheme. In first order upwind the values at cell faces are the averaged values with adjacent cell centers and in second order upwind the values are the averaged with next two cells. However FOU method produces numerical diffusion as the discretized equations are more diffused than the original equations and even more diffusion for large gradients. While the SOU method gives a smooth solution with less numerical diffusion with more time as it required more cells in numerical calculations. FOU method converges faster than the SOU method, but this analysis requires more accurate prediction of secondary flow and boundary layer flow structures, SOU method is used. This will help to predict correct flow properties in flow analysis for retrieving a solution. For more details about FVM method reference Tu, et al [19].

3.3 Gambit Software

Gambit (v2.2.30) is mesh generation software used for modeling and meshing the complex objects. Gambit also has a separate functionality to model turbine blades determining a pitch and hub data. Gambit is used to design and mesh the turbine blade model. For this blade cascade model, control volume around the turbine blade represents the flow domain. Gambit can discretize the flow model into smaller volumes known as cells for flow analysis. Gambit also has the functionality of discretizing faces into cells using various meshing techniques. Gambit also has the ability to export discretized file into various formats.

3.4 Fluent Software

Normally commercial CFD codes are used with numerical optimization methods to perform optimization process. Most of these CFD codes are validated by their own companies for various applications. For this problem, Fluent, (v6.3.30) the commercial CFD software is used for the aerodynamic shape optimization of turbine blade cascade. Generally for turbo machinery shape optimization problems, FLUENT is extensively used with various numerical optimization techniques for various types of blade profiles. Fluent also previously used with DE algorithm for optimization problem and has given very good results. [14, 15, 17]

Fluent uses the discretized domain and mesh created in Gambit to solve governing fluid and turbulence equations over the entire control volume. Fluent cannot create model and meshes and therefore it must be utilized with Gambit or other mesh generation software for flow analysis. But Fluent can modify boundary conditions and zone type of the mesh to use it for the solution. Then fluent case file is generated and used with DE code for the optimization process. More information regarding FLUENT analysis process is given in Chapter 5.

CHAPTER 4

DIFFERENTIAL EVOLUTION ALGORITHM FOR BLADE OPTIMIZATION

This chapter mentions the design optimization methodology using numerical optimization algorithm coupled with CFD analysis solver for increasing the value of exit total pressure. This will accomplish by using differential evolution method [22] with FLUENT solver on parallel computers to find best stacking path of turbine blade in the annular cascade. The analysis of each candidate stacking path design is computed using a CFD solver FLUENT with DE algorithm. A parallel computer system is used to solve many candidate designs simultaneously. It is also important to define constraints and parameterize the stack line for DE. These will mention in latter part of the chapter. First Overview of differential evolution method is given.

4.1 Introduction to Differential Evolution method

Differential Evolution is an evolution strategy (ES)-based method developed by Storn and Price in 1995 for single-objective optimization in search spaces. It is theoretically simple process with good convergence properties and is widely applied in variety of applications [8, 9]. DE is an evolutionary algorithm which depends on design populations and some basic operations like selection, cross-over and mutation by selecting randomly initial parameters like other evolutionary strategies such as GAs to get the required solution. But in DE each separate parameter works as a floating point numbers. It interprets the objective function value as a measure of that points' fitness as an optimum. Therefore, DE is a good choice for the minimization of the functions that works on continuous design variables. As a typical evolutionary algorithm, DE uses survival of the fittest principle. An initial population of vectors is converted into a solution vector through repeated cycles of selection, recombination and mutation.

The approach uses a population P^Q that has N K-dimensional real valued parameter vectors in generation Q, where K is the number of decision variables:

$$P^Q = P_i^Q = P^Q_{i,k} \quad i=1...N; \quad k= 1.....K \qquad 4.1$$

Population is initialized randomly and its size N remains constant for the whole optimization process. To generate new parameter vector for next generation, a scaled perturbations is created using weighted differences between randomly selected two or more parameters from current population. These perturbations modify another parameter vector selected from same population and generate new trial parameter vector using following mutation scheme equation

$$P_{i,k}^{Q+1} = p_{i,k}^{Q} + F.(p_{b,,k}^{Q} – p_{c,k}^{Q}), \quad i=1....N; \quad k=1.....K \qquad 4.2$$

Where a, b, c belongs to sign {1.....N};

Where a, b, c is the random parameters from the population that are naturally different from each other. The mutation parameter F ranges from 0.5 to 1. It is a real constant parameter that controls the amplification process.

DE is similar to other ES approaches as it also uses discrete cross-over. The strategy used in Differential Evolution is to modify trial parameter vector P_i^{Q+1} as

$$P_{i,k}^{Q+1} = p_{i,k}^{Q+1} \ \text{if} \ r_{i,k} \leq C_r \ \text{or} \qquad 4.3$$

$$P_{i,k}^{Q+1} = p_{i,k}^{Q} \ \text{otherwise}$$

Where $r \in [0, 1]$ is a constant random number, $C_r \in [0, 1]$ is a cross-over parameter.

Mutation and cross-over operations are performed for producing new vectors for continuation of optimization process. Some vectors t may fall outside the boundaries of variables. Therefore, new vectors are accepted only if they fall inside the variables boundary constraints during mutation and cross-over operations. The evaluation of each parameter can be done separately using parallel computers.

The original DE algorithm invented [22] was mainly intended for unconstrained optimization process, but this optimization analysis requires constraints on the design parameters for stability of the optimization process. If the parameters go beyond constrained boundaries, invalid geometries may occur. Therefore constraints are applied in an existing DE code and trial is performed to check the bound violation. If any variable breaks the predetermined boundary, it is changed with the nearest boundary value and change the design into correct design. This is mentioned using following fragment of C code.

```
if   (inibound_h<inibound_l)
 {
 printf ("\nError!inibound_h=%f<inibound_l=%f\n", inibound_hinibound_l);
 exit (1);
  }
```

Here inibound_h and inibound_i are the higher and lower limits of the boundary variables. This code corrects the design and puts it back into its original population.

4.2 Turbine blade shape parameterization

The ultimate goal of the optimization problem is to minimize the pressure loss by reducing secondary flow and vortex near the end wall, so turbine blade is optimized to decrease pressure loss in the annular cascade. Improvement in aerodynamic performance is achieved by decreasing outlet total pressure loss for the annular cascade. Therefore, total pressure loss is measured in terms of percentage loss using the difference between inlet and outlet total pressure and its value is 24.9 % for the straight blade. So, our aim is to optimize the stack line of the straight blade to reduce total pressure loss in the annular cascade.

The turbine blade stack line is parameterized using control points for DE method. 2D turbine blade geometry follows one control line called ‘stacking line ‘to make 3D turbine blade. The shape of the turbine blade is determined by the location of points of the stack line from hub to tip in annular cascade. The location of stack line points was defined by its co-ordinates. Considering the translation of the blade from hub to tip of the cascade, the stacking line path can be interpreted as a closed line with its ends is fixed in Cartesian space. This stack line path is parameterized using a B-spline with 9 control points. As the 3D turbine blade geometry has fixed ends in z direction, constraints for the ends of stack line has to be applied, so the two end locations of B-spline are considered as fixed. The middle 7 control points can move in vertical directions with the objective of maximizing total pressure. So, the y positions of 7 control points were selected as design variables.

This vertical movement of control points has to be constrained for the optimized stacking line path. The upper constraint for vertical movement is selected as 0.001 and lower constraint for vertical movement is selected as -0.001. These 7 design variables are able to move between these bounds to form the stacking path for maximizing the exit total pressure.

The stacking path is selected at the origin of the control volume, for which location of first control point is at z=0. As the span length is 30 mm, the other end of the stack line is fixed at 30 mm. Then, 7 control points become design variables known as y_1, y_2, y_3, y_4, y_5, y_6, y_7. Seven control points described the stack line motion during the optimization process. Figure 4.1 shows the example stacking line defined by its control points. The geometrical constraints applied on turbine blade are mentioned in Table 4.2

Table 4.2 Design Constraints for Geometry

parameter	**Lower Constraint**	**Upper Constraint**
Position (y_i)	-0.001 mm	0.001 mm
Fixed Control Point(y_f)	0mm (z=0)	30 mm (z=-0.03)
Minimum Allowable distance between vertical movement	2.00 mm	-----------

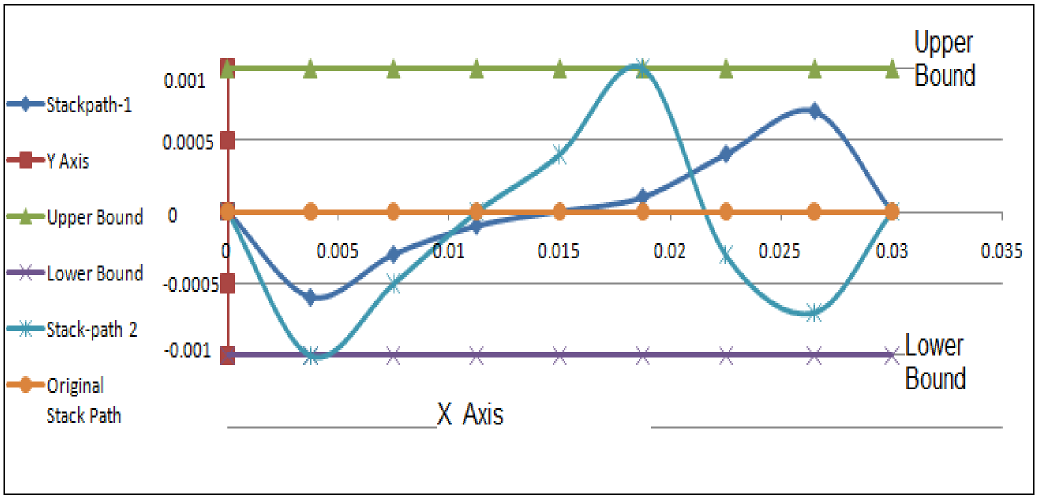


Figure 4.1 Example of stacking line translation motion analysis

4.3 Optimization process using DE Algorithm

The DE optimization algorithm needs some initial parameters to start the process. First Strategy is decided for DE optimization with maximum number of generations and number of parameters for the iteration process. The cross over factor, weight factor and random seed are also required before start of the initialization process. Here 7 design parameters are used in DE optimization code. The population size for the selecting design variables taken as 27 in the optimization.

Present differential evolution algorithm uses floating-point representation to represent design parameters so if design candidates where an individual is characterized by a vector of real numbers. Because of a typical EA, DE has a random initial population which is improved by applying selection, mutation and cross-over operations. All of the input variables are need to be checked for their constraints. Upper and lower bound for control points are selected as 0.001 and -0.001 for the translational motion and these factors are easily implemented in existing DE code using a simple C code as shown in above section.

CR and F remain constant during the whole algorithm process. Cross-over parameter controls the cross over operation and shows the probability that an element for the trial vector is selected from the linear combination of randomly chosen vectors in place of old vectors. CR controls the rotational invariant of search and its value is 0.5. F is a control parameter which controls the speed and robustness of search. It's value typically between 0 and 1. Here it is taken as 0.8. All initial parameters are used in DE algorithm during the optimization process.

CHAPTER 5

TURBINE BLADE ANALYSIS

5.1 Outline of analysis procedure

Turbine blade viscous flow model obtained by CFD analysis is optimized using differential evolution algorithm. First 3D airfoil mesh is created using GAMBIT mesh generation software with boundary layers at the root of the 3D blade for better flow estimation in that region. This mesh model is imported in FLUENT for flow analysis. FLUENT has given very good results for turbine blades I in the past. Optimization procedure is set up to run using FLUENT in parallel with DE optimization code to get improved outlet total pressure. Linux parallel computing system, which usually runs on the batch mode, is used to run the autonomous analysis sequence. Batch sequence read mesh points using neutral file and DE optimized the stack line of blade from hub to tip for higher exit total pressure. Parallel process can be run until sufficient improvement in total pressure was obtained. The more information on parallel system has been given in the parallel computing section 5.7.

The figure 5.1 shows the flow chart mentioning the optimization analysis sequences for the turbine blade. The black box has all the procedures which will run iteratively along the optimization process and will be mentioned in detail later in the chapter.

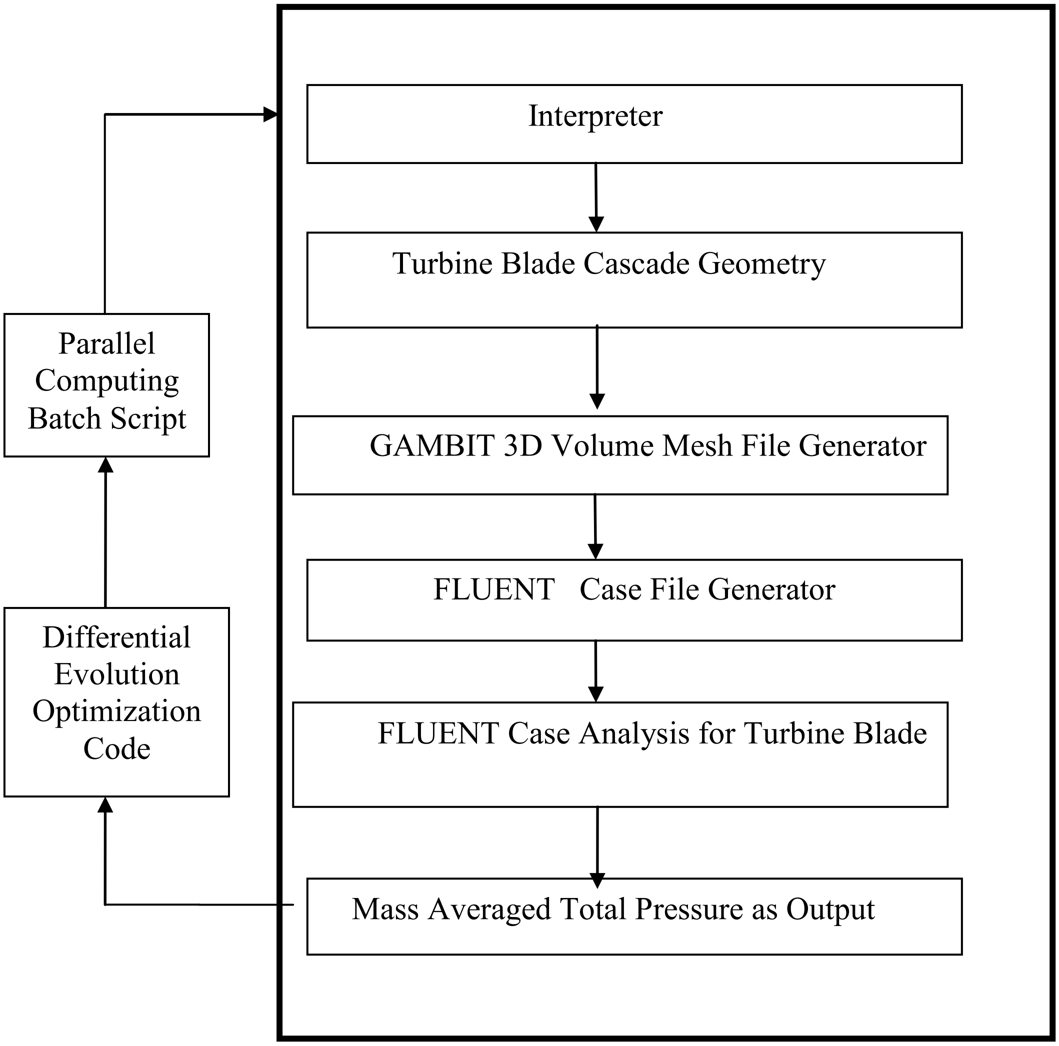

Interpreter
Turbine Blade Cascade Geometry
GAMBIT 3D Volume Mesh File Generator
FLUENT Case File Generator
FLUENT Case Analysis for Turbine Blade
Mass Averaged Total Pressure as Output
Parallel Computing Batch Script
Differential Evolution Optimization Code

Figure 5.1 Flow Chart for optimization sequence

5.2 Interpreter

The DE code exports the stacking line path points which control the shape of the stacking line path by variable text file (vars.txt). The file defines the stacking line with non dimensional variables. The interpreter is a c code which converts non-dimensional stacking line points from vars.txt file into Cartesian coordinates and output the data into a file labeled stacking_line.dat.

5.3 Generation of 3D airfoil mesh using Gambit

Turbine blade analysis requires blade mesh file with all the boundary conditions which will be used in optimization process. This file is a combination of VKI turbine airfoil with its boundaries and 3D volume mesh generated using Gambit mesh generation software. Neutral file has all three combinations and is exported from gambit for use in flow analysis using FLUENT. Figure 5.2 shows the simple process for making 3D airfoil neutral file.

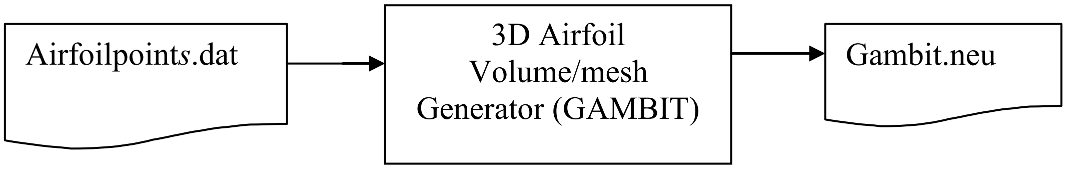


Figure 5.2 Airfoil neutral file process flow

5.3.1 Airfoil Geometry

The geometry of a stator blade row from gas turbine engines was used for the experiment. The blade row is characterized by relatively higher turning angle and a low aspect ratio. The 2D airfoil blade is stacked in z direction to form 3D annular cascade. VKI airfoil is selected as a base airfoil. The stator profile is untwisted and all cross sections have a constant inner and outer diameter. Its co-ordinates are mentioned in Appendix A. The sketch of the blade cascade data is shown in Fig 5.3 .The geometrical data of the 2D turbine stator cascade used in the test-case are given in the Table 5.1

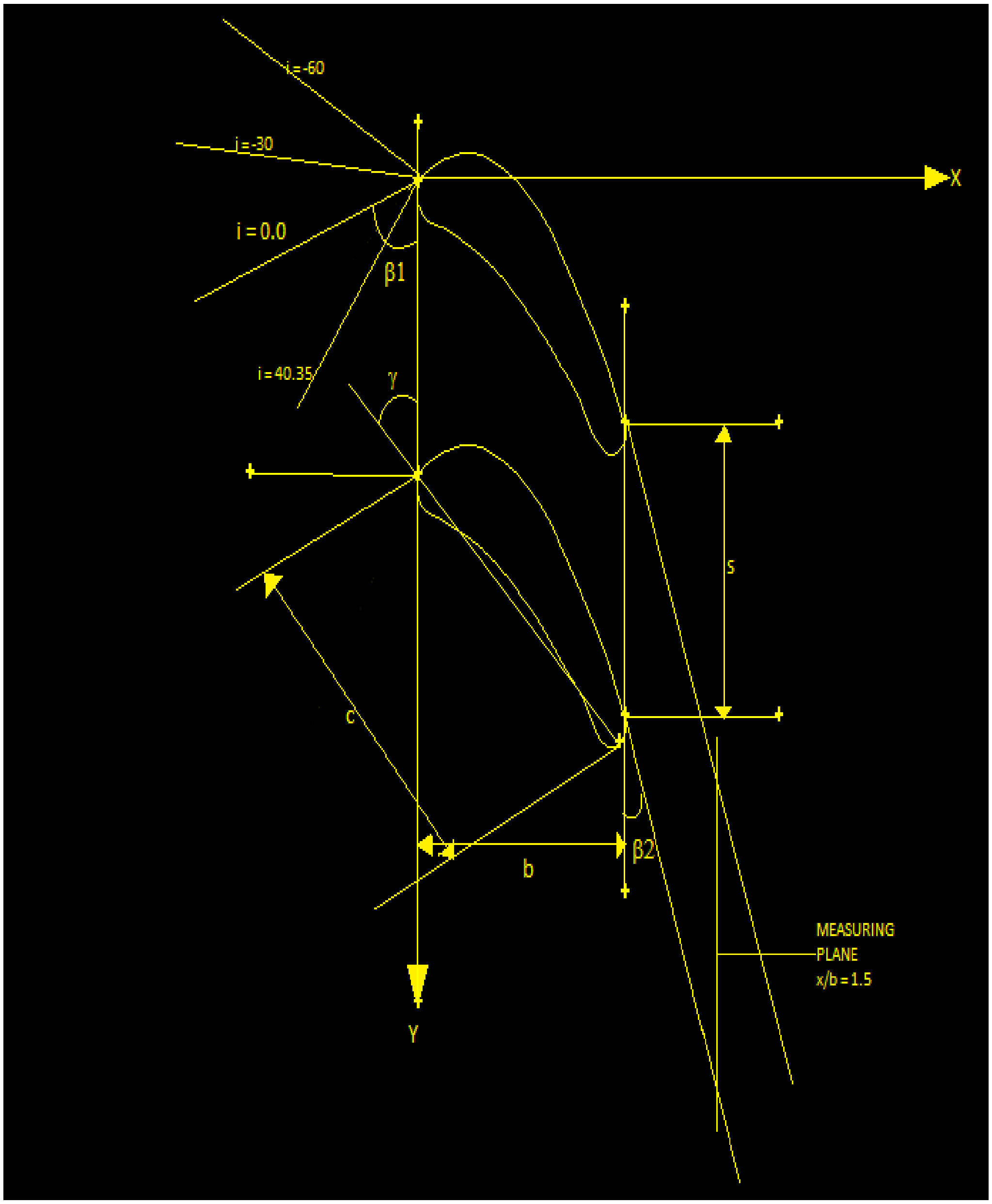


Figure 5.3 2D cascade geometry data

Table 5.1 Cascade Geometry Data

Parameter	**Value with Unit**
Chord Length, c	64.795 mm
Axial Chord Length ,b	45.06 mm
Inlet Blade Angle,β_1	40.35^o
Pitch S	45 mm
Span, h	30 mm

In the annular cascade of gas turbine engine, the blades are stacked using hub to tip ratios, whose assign value is 0.8. First the blade number in the annular cascade is assumed to calculated blade to tip ratio using prior turbines. But the blade number of an annular cascade is chosen in such a way that the azimuthal distance of two near blades at a mean radius has the value of the pitch of the cascade:

$$n = \frac{2\pi r_m}{s} \qquad 5.1$$

The value of n was rounded to integer value and the mean radius can be calculated using above equation. Mean radius can calculate hub and tip radius. This resulting hub to tip ratio has almost identical value with our initial guess value. The geometrical data for the annular turbine cascade are given in Table 5.2.

Table 5.2 Annular Cascade geometry data

Cascade Notation	Value
Hub to tip ratio r_h/r_t	0.8
Blade Number ,n	20
Hub Radius r_h	127.38 mm
Mean Radius ,r_m	112.38 mm

The annular cascade is simplified into linear cascade for the ease of computational modeling. It reduces the complex geometry and makes modeling easy because of linear cascade. It is achieved by considering straight span instead of circular span between hub and tip in the cascade model. The problem deals with aerodynamic losses such as secondary flow and boundary layer vortexes. These effects will be better experienced between the two adjacent airfoil blades in the cascade. So it is appropriate to mesh the cascade between the two adjacent airfoil blades for flow analysis. Several researchers tried this approach successfully for measuring required parameters using blade to blade mesh. [23, 24, 25] Therefore, the grid generation simplifies and the number of the cells is reduced for this type of mesh. The sketch of the 2D computational domain is shown in Figure 5.4.

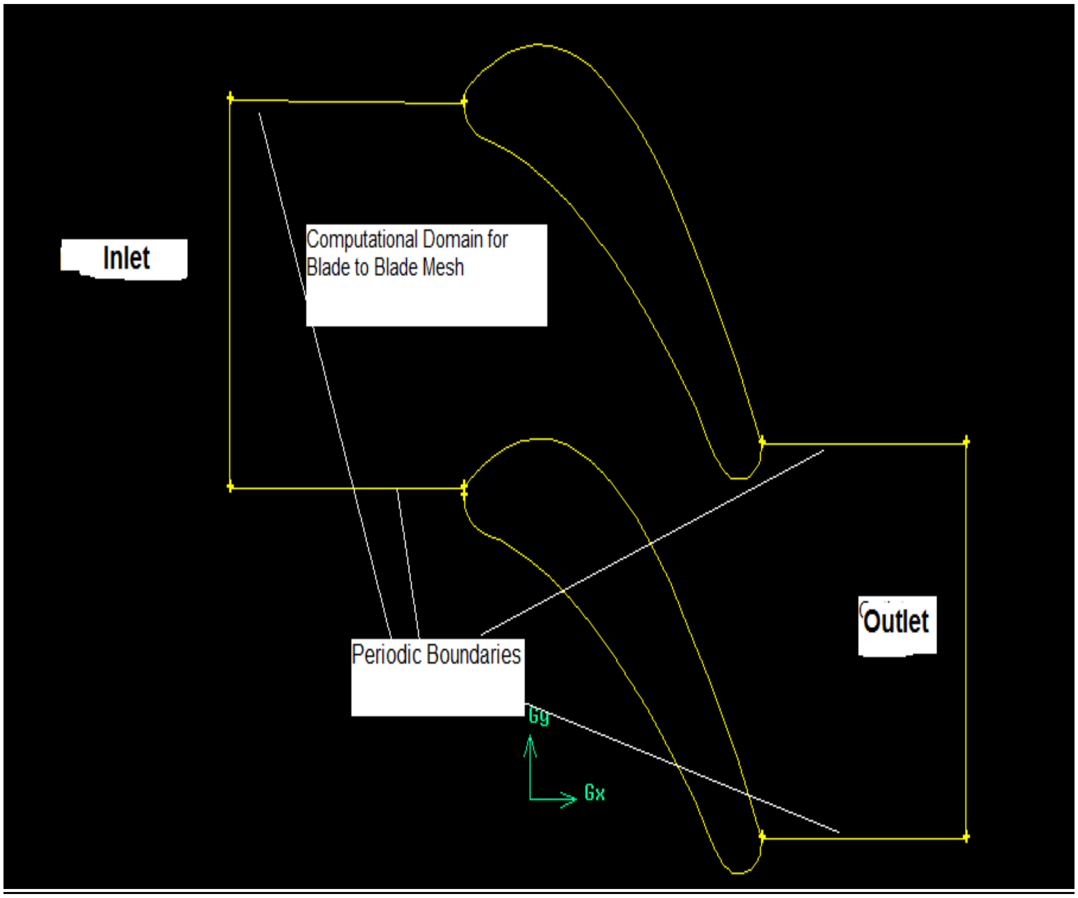


Figure 5.4 Computational domain for Blade to Blade mesh model

5.3.2 Mesh for 2D cascade

First 2D geometry was meshed for the purpose of checking the correctness of the mid span flow and adjusts distance between node points on the front and rear walls. It was also used as the pattern for 3D volume mesh. Airfoil data points were imported in Gambit using set of co-ordinate points, which is found in appendix A. Then another airfoil is created using pitch distance between two airfoils. The imported points are connected using NURBS lines and edges are created. The inlet and outlet of the computational domain were moved away with 0.35 mm distance from the leading edge and trailing edge respectively.

As the VKI blade has thick trailing radius, it is separated using one connecting point to make leading edge and trailing edge for blade to blade domain. This is completed by using shape preservation rule to move saddle point for edge separation [26]. Therefore, the computational domain of the flow is divided into the region from trailing edge of top airfoil to the leading edge of bottom airfoil as shown in Fig 5.4. At start, edges were meshed in such a manner where size of the elements was reduced in the areas where high pressure and velocity gradients were expected. Around the airfoil blade more cell points are used to control the cell height and create sufficient boundary nodes in the vicinity of the walls. The fluid region has structured mesh with triangular elements with clustering nodes in the airfoil blade region. The final 2D face mesh has 24 +56+24 cells in the top/bottom edges. Figure 5.5 shows the final 2D mesh. The other edge mesh parameters are given in Table 5.3.

Table 5.3 2D edge mesh parameters

Edge mesh points	**Dimension**
Front/Rear Edge Mesh Points	24
Inlet Edge mesh Points	28
Outlet Edge mesh points	32
Airfoil Mesh Points	112

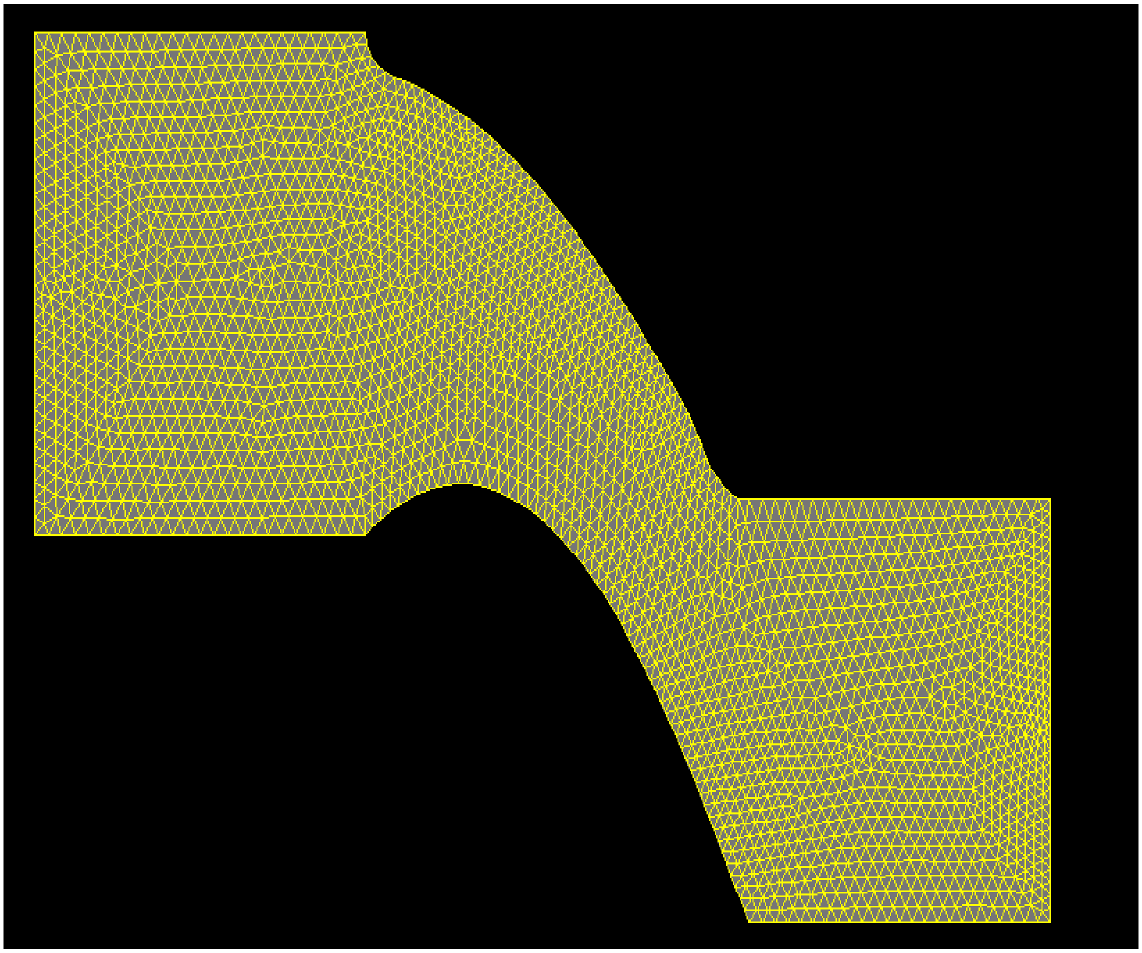


Figure 5.5 2D Airfoil mesh

The 2D mesh is formed of 4780 triangular elements. A high mesh quality is essential for good results of a CFD simulation. So, mesh quality is checked for skewness of the elements and almost 80 % cells have" Equiangle Skew "value lower than 0.1. According to methods of quality judgment, the mesh has good quality. The co-ordinate system was oriented such a way that the X-axis shows the flow upstream direction and Y axis is in agreement with azimuthal component of the flow downstream of the cascade.

5.3.3 Geometry and mesh for 3D annular cascade

To make geometry of annular cascade another face is created using hub to tip ratio of 0.8 and span distance of 30 mm in negative z direction from original 2D face. The face has lower aspect ratio compared to original 2D face. Hub and casing faces are connected using edges to generate other faces like inlet and outlet in the GAMBIT. Then 3D cascade volume is created by joining all the faces in GAMBIT.

The 3D volume mesh is constructed by extruding 2D mesh in z direction. The spacing between the airfoil blades varies along the blade height of annular cascades, the cells of the mesh close to the hub and shroud (tip), are clustered in 3D volume mesh because of high pressure gradients are expected near the wall region. The boundary layer is also created for the turbulent boundary layer problem as secondary flow and vortex boundary layers are expected near the end wall. It is shown in Figure 5.6.

As 2D mesh was used as pattern of sweep for the 3D volume mesh cascade, it will increase the number of cells which contain the high skewness ratio reaching the value of 0.85. They could decrease the convergence rate or deteriorate the solution. With the wall functions, number of cells in a mesh can be reduced and skewness of the cells is avoided. For highly viscous flow model, the dimensionless wall distance y+ on the blade height edge is applied in radial direction in order with boundary layer near the shroud wall. After several trial and error approaches acceptable range for y + distance is between 60 and 150. After several trials, the final 3D volume mesh with boundary layer is generated using Gambit. The final 3D mesh has around 173000 hexahedral/wedge volume elements. The final 3D Volume mesh has shown in figure 5.7.

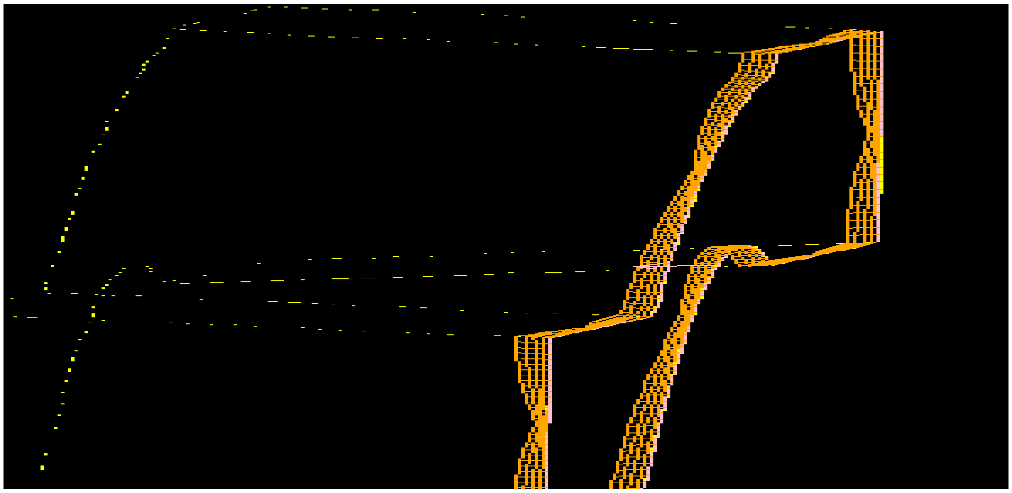


Figure 5.6 Boundary layers near the end wall region

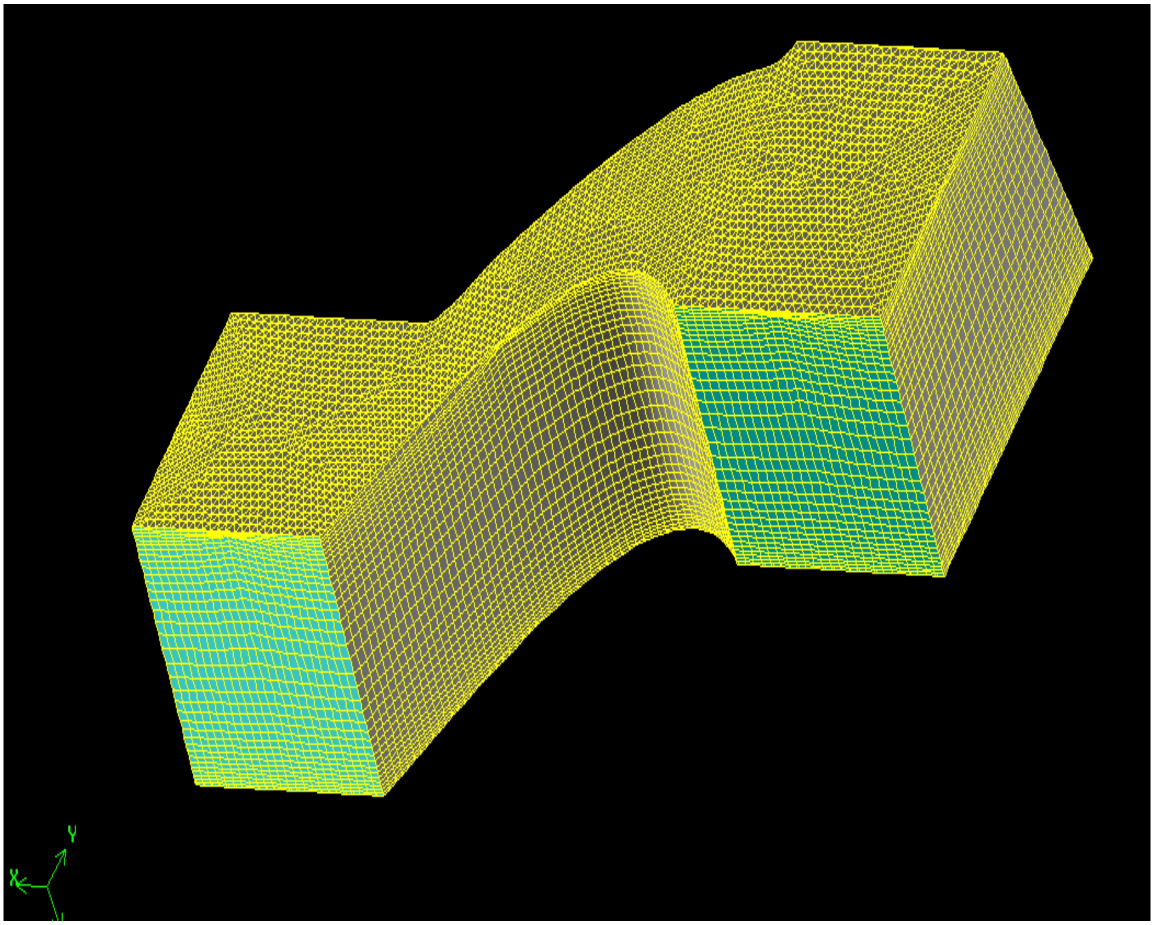


Figure 5.7 3D Volume mesh

5.4 Turbine blade model using Fluent

Final turbine blade flow model required for analysis is created using FLUENT by applying necessary boundary conditions. Script file read fluent commands for reading mesh data, setting up the boundary conditions, select solver conditions and also iterate blade model until solution reaches convergence. The script file that reads FLUENT text commands and make fluent case file is given in APPENDIX A and labeled as ‘flinput”. The fluent case file is used with differential evolution algorithm to get the results for optimized blade. The schematic diagram for this process is shown in Fig 5.8.

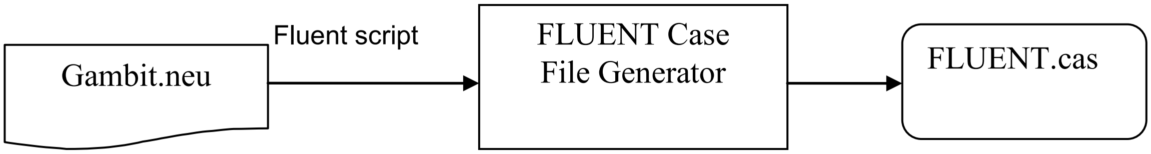


Figure 5.8 Fluent Case File Generator Process Flow

The remaining parts of this section will give the details about fluent conditions used to get the turbine blade flow model. The parameters with no discussion in this section have the FLUENT default value.

5.4.1 Solver conditions for FLUENT

FLUENT is software that uses finite volume method used for the solution. In FVM, computational domain is divided into control volumes and then discretized and integrated using conservation of mass, momentum and energy equations. Compressible flow is considered for the air due to inlet velocity is in the high subsonic region for the annular cascade. It requires density based (coupled) solver for solving turbine blade model. Density based solver is always applicable where high interaction and dependence between density, energy and momentum occurs. The solution process is repeated using different boundary conditions until convergence is checked. The flow was simulated as steady flow.

5.4.2 Turbulence Modeling

As turbulence affects the flow behavior of the turbo machines, its consideration is critical in turbine blade analysis. Turbulence also plays an important role for calculating viscous flow effects, which causes high pressure gradients in near wall region. These high pressure gradients are responsible for secondary flows and boundary layer flows which cause total pressure losses in annular turbine cascade. Therefore it is critical to consider appropriate turbulence model for the annular cascade.

Turbine blade model use k-epsilon turbulence model with realizable option. The k-epsilon model is mainly used for confined flows which usually occurred in turbine blade cascades and for high Reynolds number flows such as this problem. The turbulent equations for k-epsilon model are formulated in chapter 3.1.4. The realizable K-epsilon model gives more accurate results than any other turbulence model for the annular cascade. It also correctly predicts the boundary layer flows near the tip wall region in the cascade.

5.4.2.1 Wall Functions for K-epsilon turbulence model

Walls affect greatly the behavior of the flow. In the vicinity of the tip wall, the velocity fluctuations are damped and viscous effects become more important. High pressure and velocity gradients occur near the tip wall region in the annular cascade. So, non-equilibrium wall functions are used with K-epsilon turbulence model. . Non-equilibrium wall functions proved very effective in complex flows such as turbo machinery flows involving separation, reattachment and impingement where the turbulence and mean flow are experienced high pressure gradients and change in velocity [21]. The non-equilibrium wall functions predict better secondary flow effects in near the wall regions for high viscous flows.

5.4.3 Fluid Properties

Fluid passing through the annular cascade is air and modeled as ideal gas with constant specific heat capacities. Fluid properties for air with temperature of 300 ‘K and 1 atm pressure is given in Table 5.4

Table 5.4 Fluid Properties of the air

Property	**Value**
Density (kg/m^3)	Ideal-Gas
C_p (j/kg-k)	1006.43
Thermal Conductivity(w/m-K)	0.0242
Viscosity(kg/m-s)	1.7894e-05
Molecular-Weight(kg/kgmol)	28.966

5.4.4 Boundary Conditions

Boundary conditions have played an important role in determining flow properties in the solution. In the beginning, the whole analysis process runs several times varying mesh size and changing boundary conditions inputs for different velocity combination. The purpose of this process is to reduce computational time for the optimization process. Finally converged solution for turbine blade model with sufficient mesh parameters as shown in Figure 5.2 is used for optimization process. Rotational periodic boundary conditions are applied for the top and bottom faces in the volume as all nodes and flow conditions must be same for both faces for the turbine blade cascade as given in most of turbine blade flow problems. [20].

5.4.4.1 Inlet

As pressure difference between inlet and exit boundary across the fluid flow is required, pressure inlet and pressure outlet boundary conditions are used. Generally, compressible flow problems use pressure inlet condition as it gives the best opportunity to measure total pressure across the fluid in turbine cascade.

Pressure inlet condition sets the gauge total pressure of inlet stream. Here the value is taken as 0.1 for ease of measurement of the total pressure loss between inlet and outlet total pressure. Then, Flow is initialized using initial gauge pressure condition. It is used to compute initial values for pressure and velocity for pressure inlet boundary condition. For this cascade problem, initial gauge pressure is taken as 0.07 to initialize the flow stream into the cascade. Cylindrical co-ordinate system is used to define the flow direction using direction vector method.

Turbulence plays an important role for determining inlet velocity distribution as values for turbulence are defined at the inlet. It will help in good composition of inlet boundary layers, which influence the secondary flows as its composition and further expansion in the cascade depends on inlet velocity distribution. Turbulent intensity and turbulent viscosity are selected as the method of turbulence specification.

Turbulent Intensity is described as the ratio of the average velocity fluctuations in respect to the main flow velocity. Normally turbine cascade problems have turbulent intensity in the range of 1- 6 %. This flow model has 5 % turbulent intensity. High turbulent viscosity ratio is essential for producing high pressure gradients in near the wall region. This flow problem has turbulent viscosity ratio as 300.The all values for inlet boundary condition are given in table 5.5.

Table 5.5 Properties at Inlet

Total Pressure at the inlet	0.1 atm
Initial pressure at inlet	0.07 atm
Turbulent Intensity	5 %
Turbulent Viscosity Ratio	300

5.4.4.2 Outlet

As Total pressure loss between inlet and exit boundary is measured for the calculation of aerodynamic performance of the turbine cascade, pressure outlet condition is selected for the outlet of the cascade. Non-reflecting boundary condition is used for the exit of the turbine cascade. To get the mass averaged total pressure at the outlet, average boundary pressure option is enabled in the FLUENT. This condition helps to get the outlet total pressure, which will be the objective function in the DE Optimization process.

5.4.4.3 Walls

For the annular turbine, hub and tip of the cascade are specified using wall boundary functions. The no-slip condition is applied to oppose tangential velocity for the fluid at the walls. There is neither heat flux across the walls. The non-equilibrium wall functions for the wall boundaries are used in the turbulence equations.

5.4.4.4 Periodicity

Periodicity is mathematically simple boundary condition. It enables all variables and also all coefficients have the same value at both ends of the computational domain. Because of annular cascade, rotational periodic boundary condition is used to reduce computational domain. To make periodic boundary from hub to tip of the cascade, a rotational axis and direction is defined for fluid in the control volume [21]. This will ensure an annular cascade by creating the mesh at equal angles from the origin in radial direction.

5.4.5 Solution Controls

Solution controls are used to control the flow properties of the solvers in turbine blade analysis. Solver parameters like under relaxation factors, discretization schemes and courant number are changed in FLUENT to provide stable solution and faster convergence rate for blade analysis. Following section will give details about these parameters.

5.4.5.1 Courant number

The set of governing equations is discretized with time for both unsteady and steady analysis for flow analysis. But in only steady stated analysis the solution proceeds with time until steady state solution with convergence is reached. Therefore courant number represents a relaxation factor in steady state analysis. Here Courant Number defined in Equation 5.2 gives the largest velocity component in a given cell (u), time step (Δt) and cell size (Δx) in the direction of velocity. When flow equations has more advection is than diffusion, then courant number automatically decreases less than 1 to decrease oscillations in the residuals. It also helps to decrease numerical diffusion and improve accuracy of the solution.

$$CourantNumber = u\frac{\Delta t}{\Delta x} \quad 5.2$$

For blade analysis, courant number is set to low value of 0.8 for initializing the solution for high velocity flow for first 1000 iterations. Then it will be increased to 0.9 for the rest of 1000 iterations. Higher value than this number will cause stability problems.

5.4.5.2 Under-relaxation factors

FLUENT solver has the under relaxation factors to control the effect of previous iteration on current iterations. Following equation shows the relationship between the under-relaxation factor (α) and the scalar value (ϕ) and resulted change in scalar value (Δϕ).Here the value of turbulent viscosity and solid is taken as 0.8 because of getting convergence. It will help in reducing oscillations in energy and turbulence equations and more smooth solution will be achieved. Table 5.6 gives list of default values of under-relaxation factors used in this analysis.

$$\phi_{new} = \phi_{old} + \alpha\Delta\phi \quad 5.3$$

Table 5.6 Fluent values for under relaxation factors

Parameter	**Dimension**
Turbulent Kinetic Energy	0.8
Turbulent Dissipation Rate	0.8
Turbulent Viscosity	0.8
Solid	0.8

5.4.5.3 Discretization schemes for flow analysis

This turbine blade analysis problem uses 2nd order upwind scheme to calculate fluid flow properties. Moreover the 2nd order upwind scheme is also used to evaluate turbulent values of kinetic energy and dissipation rate. In first order upwind technique, scalar properties at current cell faces are derived from the upstream cell values from the previous iteration. Upwinding Scheme is a technique which uses upstream values to calculate the current cell values. Similarly second order upwinding method uses the cell values of two adjacent cells to compute the face value of the current cell. Table 5.7 gives the discretization scheme used for flow analysis problem.

Table 5.7 Discretization Scheme

Parameter	**Discretization Scheme**
Fluid Flow	Second Order Upwind Scheme
Turbulent Kinetic Energy	Second Order Upwind Scheme
Turbulent Dissipation Rate	Second Order Upwind Scheme

5.5 Fluent analysis

Using solver commands mentioned in previous steps, flow model with required results is achieved which can be iterated with DE using a parallel computing system. The input file named as ‘flinput’ was used to control the commands of fluent and other iteration process performed by fluent. This file can be seen in Appendix A. The flinput file reads the current case file, initializes the solution and assigns the number of iterations. Then it runs the solution for number of maximum iterations assigned by the flinput file. This process can be repeated using parallel computing system until we got the required solution for turbine cascade matching all necessary fluent conditions. More details of the parallel computing system are given in next section. Figure 5.9 shows the flow chart of parallel computing analysis process using FLUENT.

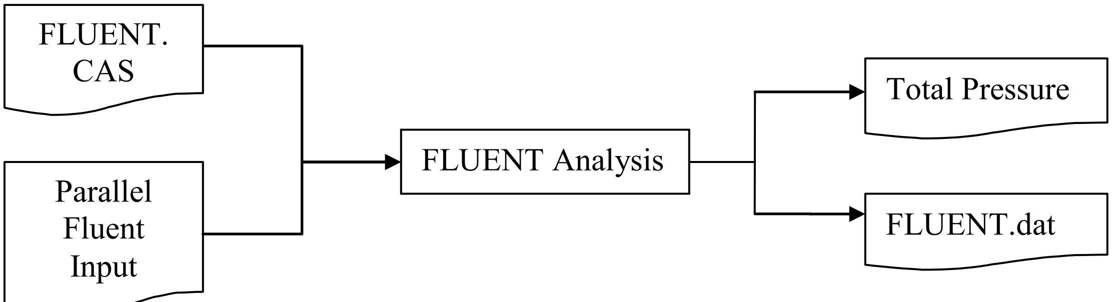


Figure 5.9 Parallel computing process using FLUENT

The resulting analysis runs 2105 FLUENT iterations to obtain required results for annular turbine blade cascade. The fluent analysis creates a data file at the end of the iterations which is named as FLUENT.dat file and other file for giving difference between total pressure across the cascade named as ‘OUTLET.srp’, that helps to measure total pressure loss across the cascade of the turbine blade. Optimization process uses the exit total pressure value as design objective to improve design performance. More turbine blade results are discussed in Chapter 6.

5.6 Parallel computing system

The whole optimization process is run on parallel computing system with computer resources provided by CFD lab at University of Texas at Arlington. The parallel computing processor is a Linux based operating system with 9 nodes. Each node is made of 2 processors which contains 2.8 GHz processor with 2 GB RAM. Each different node allows different computational processes to iterate simultaneously on same system. Every step in optimization process from the flinput to the output file generator is scripted for running in “batch mode” on parallel computing system. Batch mode is a script file which controls the analysis process and essentially connects all input, optimization codes and output files. Batch mode also allows the optimization process to iterate without human interaction. Batch script file also uses to give the name of the jobs and the number of nodes required to run the computational process.

CHAPTER 6

TURBINE BLADE OPTIMIZATION ANALYSIS RESULTS

Shape optimization process is performed to get the improved turbine blade for the annular cascade. The flow analysis results of blade cascade with original stacking path are presented and compared with the turbine blade cascade of optimized stacking path. The results are obtained by setting up the FLUENT optimization analysis with DE algorithm through parallel computer processing. First, the total pressure for original turbine blade geometry and improved turbine blade geometry are mentioned. Then, secondary flow profiles for original turbine blade in the annular cascade are shown and how it is removed by modifying stacking line path are mentioned.

These results are obtained by running 10 generations of iteration. Each generation has 27 population sizes for the final optimization analysis.

6.1 Improvement in total Pressure at outlet

DE optimization analysis improves the annular cascade performance by reducing total pressure loss. It is achieved by measuring improvement in total pressure for the optimized blade at the turbine outlet. After 10 generations, the value of total pressure at the outlet is 0.08048 atm for the best result. The original turbine blade in annular cascade has outlet total pressure value is 0.0763 atm. So, DE optimization analysis shows around 4% increment in total pressure value for the annular cascade.

6.2 Improvement in aerodynamic performance

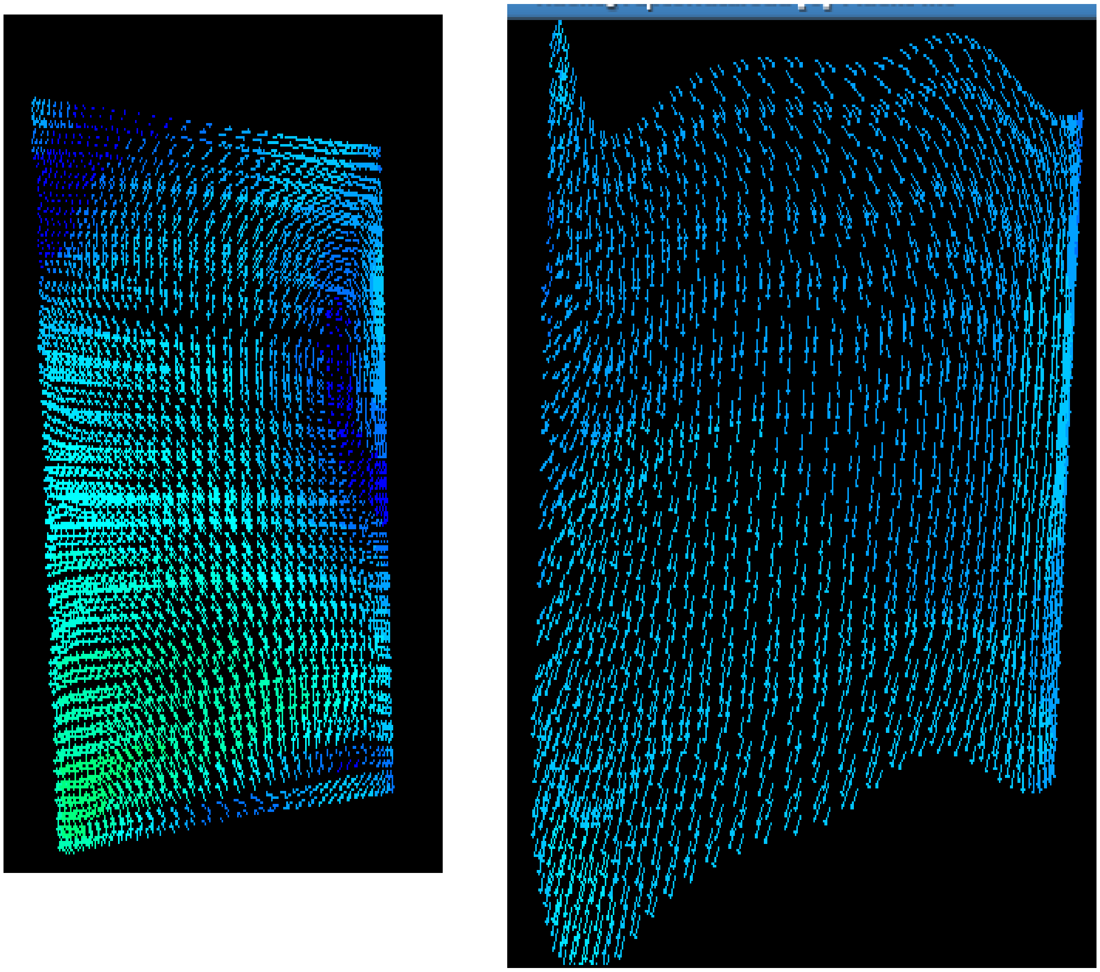
For gas turbine engine cascades, the performance is measured by the total pressure recovery through the cascade. Total pressure p_t changes in the cascade during flow passage through inlet to outlet because of various flow effects. Most of aerodynamicists measure the turbine cascade performance by total pressure recovery from inlet to outlet. It measures the amount of free stream conditions that are recovered at the exit of the cascade. Pressure losses due to secondary flow, boundary layers on the near wall region and flow separation in the cascade will affect the total pressure recovery at the exit. Here, turbine blade with straight stacking line has pressure recovery coefficient of 76.3, while the best design of turbine blade achieved after 10 generation has a total pressure recovery co-efficient of 80.48. Table 6.1 shows the comparison between the straight blade and optimized blade.

Table 6.1 Computed Performance for original and optimized design

	Pressure Ratio	Pressure recovery co-efficient
VKI turbine blade	1.31578	76.3
Optimized blade	1. 2425	80.4

6.3 Effects of Secondary Flows on aerodynamic performance.

Turbine blade with straight span produces large vortexes' and secondary flows along the hub wall region in the annular cascade. These vortices near the wall region and secondary flow velocities are the main reason behind the total pressure loss of annular cascades. Figure 6.1 shows the secondary velocity vectors in the turbine cascade near the wall region at middle of the airfoil. However the secondary flow and vortex along the wall are not seen in the optimized blade. The optimization process successfully removes vortexes' and secondary velocities in the cascade by modifying the stacking line path. The removal of secondary velocity vectors and vortexes' gives a notable improvement in the turbine blade cascade efficiency. So, the outlet total pressure for optimized blade is increased.



(A) (B)

Figure 6.1 Velocity Vectors across the span at the middle of the cascade for (A) Straight Stack line (B) Optimized Stack Line

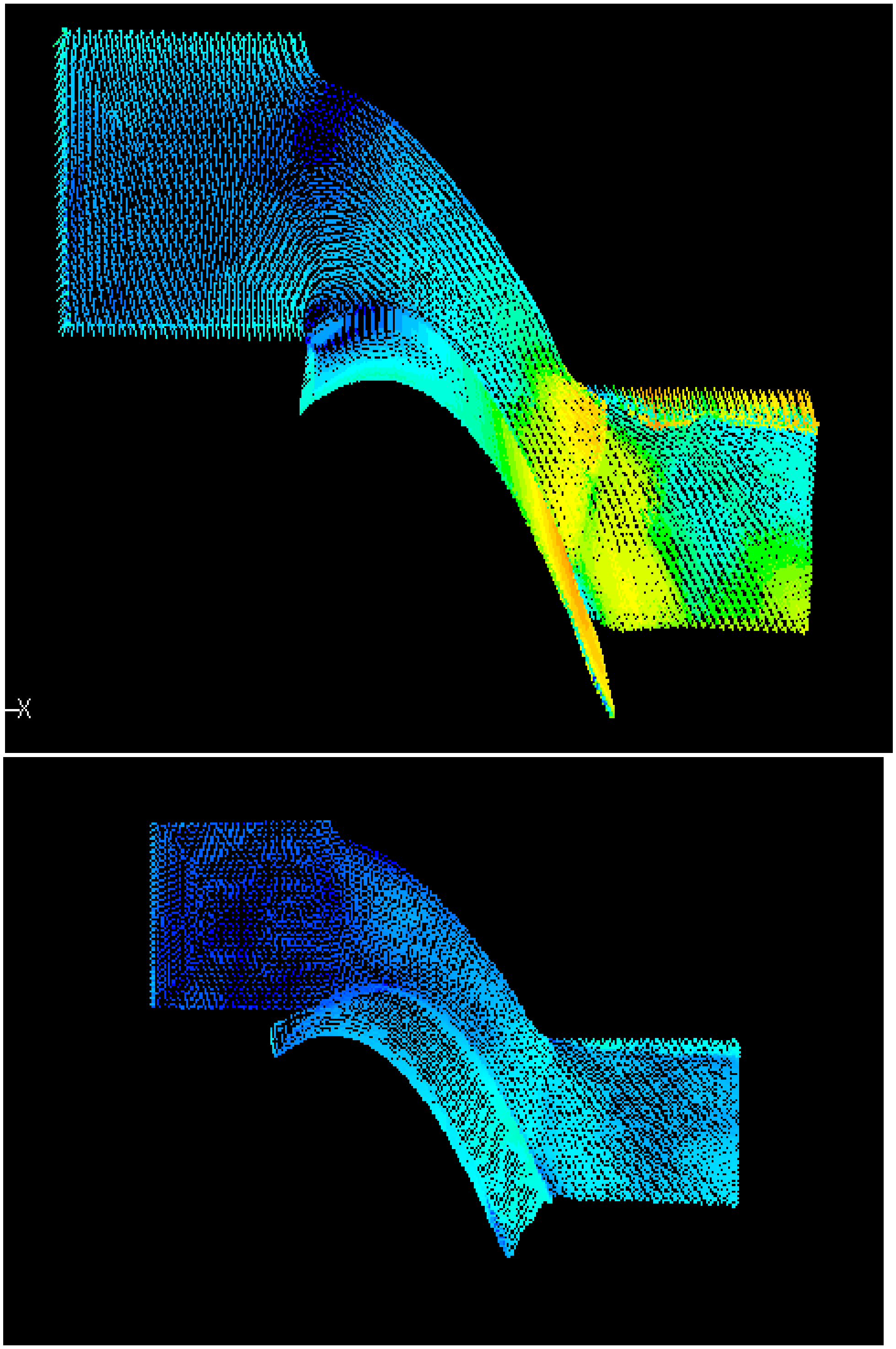


Figure 6.2 .Velocity Vectors for the Original and Optimized blade for the Rear wall

6.4 Total Pressure loss as a function of Iteration.

This optimization process runs 10 generations of iterations and every generation has 27 populations size for the objective function. The result is improved with each generation during the analysis process. Fig 6.3 shows the graph of % pressure loss with number of generations for optimization analysis. It shows that pressure loss percentage is decreased with increasing in order of generation during optimization.

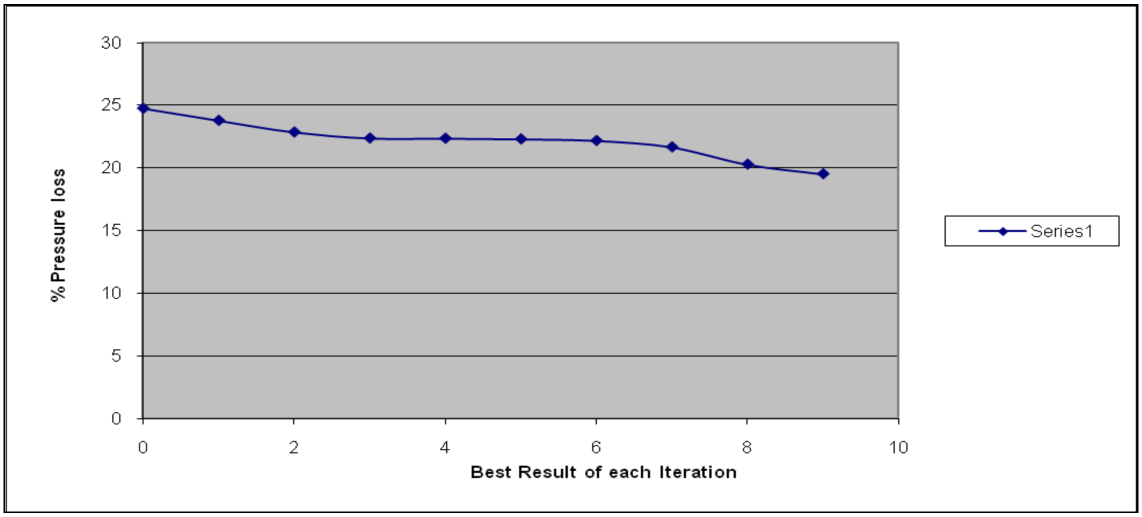


Figure 6.3 Pressure Loss in percentage versus best result of each generation

6.5 Static Pressure Contours for Original and Optimized blade

Figure 6.4 shows the contour plots for original and optimized blade. By seeing the figure carefully, we can observe the strength of secondary flow is decreased for optimized blade. Static pressure contours for original blade has more profiles near the top corner of the hub wall, while static pressure contours for optimized blade has fewer profiles at the same place. So, we conclude that strength of the secondary vectors is reduced near the top of the tip wall.

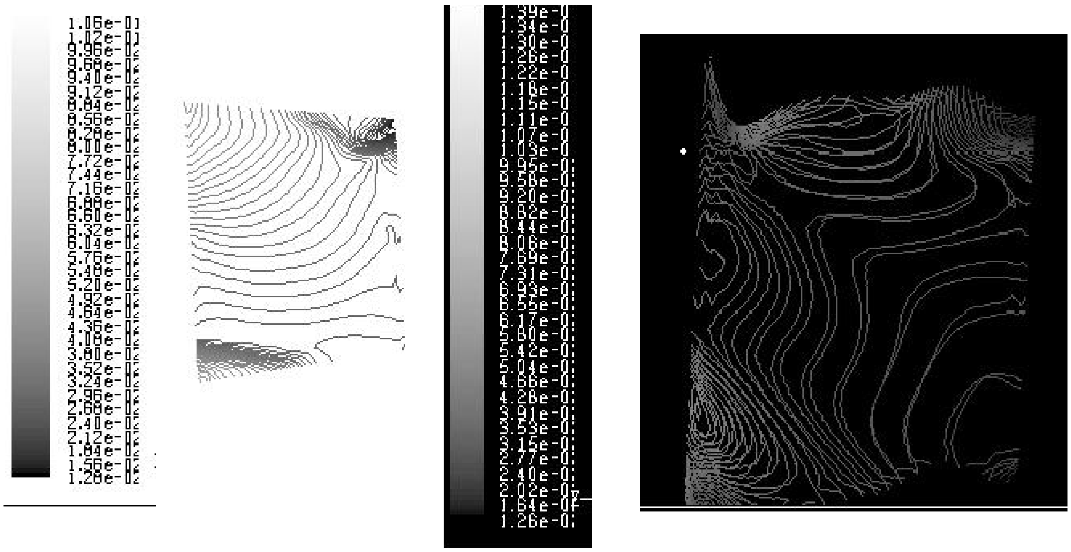


Figure 6.4 Static pressure contours for original and optimized Blade

6.6 Original and Optimized Turbine Blade

Figure 6.5 shows the sketch of the leading and trailing edges of the turbine blades with straight stack line and optimized stack line. As seen in the figure, optimized turbine blade has bowing of stack line on its hub side.

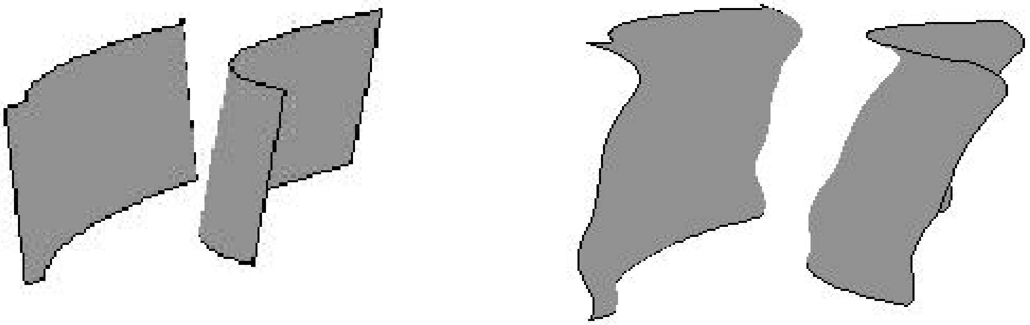


Figure 6.5 Leading and trailing edges of the original and optimized blade turbine blades

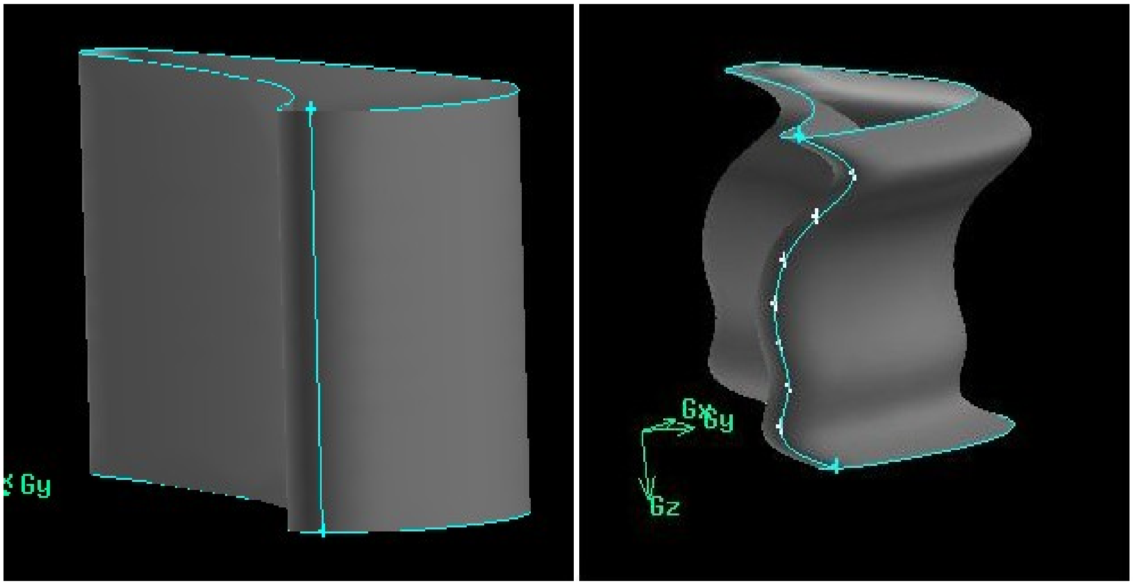


Figure 6.6 Original and Optimized turbine blade

Fig 6.6 shows the original 3D turbine blade and optimized 3D turbine blade for the annular cascade. Optimized turbine blade has bowed stack line near the hub wall. This optimized turbine blade shape is achieved after 10 generations of optimization iteration process. As we used 7 control points in DE code to get the optimized blade, it has a little unconventional shape. More conventional optimized blade shape can be achieved by reducing lower number of control points for the DE algorithm. Higher number of generations for optimization process also helps to get more conventional shapes of turbine blade. This figure shows the turbine blade shape for the best result achieved after 10 generations of iterations.

Improvement in aerodynamic efficiency for annular cascade is achieved by optimizing the stack line of the turbine blade. Result shows that stack path is little bowed near the hub region, which decreases pressure loss for the cascade. As blade with straight stack tends to produces secondary flows and vortex formation near hub region, but bowing the blade will help in stopping the formation of vortex and hence secondary flow. It will also help in formation of better free stream flow. Ultimately, it will reduce pressure loss across the cascade.

Thus, improvement in aerodynamic performance is obtained by optimizing the stack line of the 3D VKI turbine blade in the annular cascade. It also shows the use of differential evolution algorithm for the aerodynamic shape optimization process.

APPENDIX A

SUPPORTING DOCUMENTS

AIRFOIL POINTS

X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
-3.468551	4.580047	0	-0.581952	4.736988	0	1.874047	0.846226	0	-0.056769	2.286783	0
-3.469214	4.591521	0	-0.536900	4.691398	0	1.881580	0.833102	0	-0.104369	2.334317	0
-3.469948	4.603007	0	-0.492306	4.645096	0	1.888930	0.820290	0	-0.152394	2.381788	0
-3.470637	4.614538	0	-0.448143	4.598115	0	1.896085	0.807799	0	-0.200839	2.429143	0
-3.471134	4.626150	0	-0.404438	4.550512	0	1.903060	0.795582	0	-0.249673	2.476367	0
-3.471316	4.637835	0	-0.361191	4.502299	0	1.909866	0.783637	0	-0.298881	2.523421	0
-3.471110	4.649651	0	-0.318455	4.453488	0	1.916514	0.771948	0	-0.348449	2.570267	0
-3.470545	4.661585	0	-0.276242	4.404110	0	1.923008	0.760484	0	-0.398354	2.616875	0
-3.469642	4.673666	0	-0.234595	4.354180	0	1.929368	0.749238	0	-0.448584	2.663219	0
-3.468412	4.685881	0	-0.193516	4.303728	0	1.935588	0.738187	0	-0.499112	2.709249	0
-3.466889	4.698268	0	-0.153048	4.252769	0	1.941696	0.727326	0	-0.549936	2.754937	0
-3.465051	4.710819	0	-0.113202	4.201322	0	1.947604	0.716565	0	-0.601013	2.800256	0
-3.462869	4.723547	0	-0.073973	4.149428	0	1.952917	0.705711	0	-0.652330	2.845178	0
-3.460317	4.736475	0	-0.035347	4.097149	0	1.957649	0.694745	0	-0.703882	2.889663	0
-3.457352	4.749596	0	0.002729	4.044561	0	1.961808	0.683689	0	-0.755625	2.933666	0
-3.453936	4.762916	0	0.040308	3.991720	0	1.965412	0.672545	0	-0.807549	2.977187	0
-3.450068	4.776445	0	0.077398	3.938656	0	1.968470	0.661343	0	-0.859627	3.020169	0
-3.445711	4.790186	0	0.114012	3.885432	0	1.970993	0.650065	0	-0.911838	3.062594	0
-3.440841	4.804121	0	0.150168	3.832069	0	1.972978	0.638731	0	-0.964158	3.104434	0
-3.435420	4.818265	0	0.185871	3.778618	0	1.974437	0.627341	0	-1.016573	3.145651	0
-3.429425	4.832600	0	0.221137	3.725111	0	1.975464	0.614126	0	-1.069051	3.186217	0
-3.422806	4.847112	0	0.255983	3.671580	0	1.975798	0.600808	0	-1.121558	3.226107	0
-3.415520	4.861784	0	0.290406	3.618065	0	1.975415	0.587424	0	-1.174083	3.265302	0
-3.407547	4.876618	0	0.324441	3.564600	0	1.974316	0.573966	0	-1.226601	3.303763	0
-3.398831	4.891561	0	0.358066	3.511212	0	1.972468	0.560449	0	-1.279071	3.341485	0
-3.389330	4.906597	0	0.391308	3.457925	0	1.969840	0.546900	0	-1.331479	3.378419	0
-3.379022	4.921697	0	0.424163	3.404778	0	1.966420	0.533328	0	-1.383804	3.414568	0
-3.367864	4.936836	0	0.456630	3.351792	0	1.962157	0.519738	0	-1.436013	3.449894	0
-3.355820	4.951978	0	0.488734	3.298998	0	1.957038	0.506168	0	-1.488074	3.484391	0
-3.342799	4.967001	0	0.520454	3.246416	0	1.951007	0.492654	0	-1.539964	3.518021	0
-3.328721	4.981814	0	0.551817	3.194077	0	1.944023	0.479220	0	-1.591662	3.550787	0
-3.313640	4.996471	0	0.582811	3.141992	0	1.936048	0.465934	0	-1.643127	3.582672	0
-3.297723	5.011105	0	0.613431	3.090209	0	1.927018	0.452838	0	-1.694335	3.613649	0
-3.281128	5.025900	0	0.643699	3.038721	0	1.916902	0.440021	0	-1.745266	3.643710	0
-3.263867	5.040857	0	0.673609	2.987577	0	1.905663	0.427558	0	-1.795889	3.672848	0
-3.245921	5.055995	0	0.703161	2.936778	0	1.893240	0.415553	0	-1.846165	3.701067	0
-3.227290	5.071307	0	0.732353	2.886343	0	1.879611	0.404116	0	-1.896089	3.728338	0
-3.207954	5.086813	0	0.761191	2.836303	0	1.864752	0.393374	0	-1.945602	3.754667	0

-3.187924	5.102493	0	0.789665	2.786677	0	1.848650	0.383466	0	-1.994704	3.780043	0
-3.167183	5.118398	0	0.817791	2.737478	0	1.831296	0.374554	0	-2.043344	3.804473	0
-3.145726	5.134479	0	0.845549	2.688732	0	1.812714	0.366809	0	-2.091522	3.827956	0
-3.123569	5.150784	0	0.872958	2.640433	0	1.792936	0.360396	0	-2.139179	3.850498	0
-3.100686	5.167266	0	0.900014	2.592619	0	1.772032	0.355515	0	-2.186315	3.872089	0
-3.077089	5.184025	0	0.926705	2.545310	0	1.750089	0.352384	0	-2.232899	3.892743	0
-3.052721	5.200914	0	0.953043	2.498496	0	1.727227	0.351208	0	-2.278893	3.912473	0
-3.027567	5.217886	0	0.979023	2.452209	0	1.703633	0.352207	0	-2.324278	3.931282	0
-3.001583	5.234996	0	1.004635	2.406465	0	1.679498	0.355583	0	-2.369035	3.949192	0
-2.974754	5.252094	0	1.029899	2.361258	0	1.655007	0.361485	0	-2.413134	3.966195	0
-2.947024	5.269237	0	1.054801	2.316627	0	1.630405	0.370060	0	-2.456558	3.982325	0
-2.918378	5.286275	0	1.079331	2.272559	0	1.606094	0.381522	0	-2.499289	3.997602	0
-2.888801	5.303259	0	1.127323	2.186183	0	1.582667	0.396153	0	-2.541287	4.012031	0
-2.858258	5.320044	0	1.150770	2.143912	0	1.560734	0.414159	0	-2.582565	4.025631	0
-2.826749	5.336628	0	1.173843	2.102236	0	1.540051	0.435012	0	-2.623083	4.038415	0
-2.794254	5.353015	0	1.196568	2.061186	0	1.519919	0.457701	0	-2.662843	4.050395	0
-2.760757	5.369054	0	1.218925	2.020772	0	1.499854	0.481575	0	-2.701809	4.061613	0
-2.726238	5.384748	0	1.240914	1.980992	0	1.479745	0.506530	0	-2.739978	4.072131	0
-2.690682	5.399948	0	1.262533	1.941858	0	1.459538	0.532520	0	-2.777333	4.081991	0
-2.654104	5.414702	0	1.283792	1.903369	0	1.439106	0.559502	0	-2.813858	4.091223	0
-2.616463	5.428814	0	1.304681	1.865535	0	1.418317	0.587382	0	-2.849556	4.099759	0
-2.577769	5.442282	0	1.325200	1.828356	0	1.397043	0.616065	0	-2.884451	4.107544	0
-2.538006	5.454958	0	1.345357	1.791843	0	1.375101	0.645452	0	-2.918528	4.114530	0
-2.497199	5.466789	0	1.365133	1.755994	0	1.352324	0.675416	0	-2.951794	4.120787	0
-2.455348	5.477673	0	1.384557	1.720812	0	1.328675	0.705931	0	-2.984187	4.126583	0
-2.412461	5.487510	0	1.403609	1.686294	0	1.304180	0.737031	0	-3.015707	4.132199	0
-2.368578	5.496194	0	1.422288	1.652461	0	1.278896	0.768742	0	-3.046306	4.137861	0
-2.323709	5.503625	0	1.440615	1.619295	0	1.252869	0.801118	0	-3.075939	4.143796	0
-2.277903	5.509797	0	1.458581	1.586795	0	1.226176	0.834189	0	-3.104463	4.150562	0
-2.231190	5.514506	0	1.476173	1.554980	0	1.198872	0.868011	0	-3.131708	4.158658	0
-2.183638	5.517744	0	1.493413	1.523831	0	1.170985	0.902585	0	-3.157600	4.168224	0
-2.135281	5.519357	0	1.510292	1.493349	0	1.142507	0.937902	0	-3.182143	4.178927	0
-2.086196	5.519388	0	1.526818	1.463543	0	1.113402	0.973927	0	-3.205361	4.190514	0
-2.036449	5.517629	0	1.542992	1.434404	0	1.083641	1.010647	0	-3.227269	4.202831	0
-1.986107	5.514122	0	1.558806	1.405921	0	1.053207	1.048021	0	-3.247875	4.215778	0
-1.935233	5.508810	0	1.574270	1.378095	0	1.022075	1.086015	0	-3.267202	4.229272	0
-1.883922	5.501634	0	1.589382	1.350926	0	0.990244	1.124639	0	-3.285297	4.243207	0
-1.832248	5.492536	0	1.604147	1.324393	0	0.957725	1.163865	0	-3.302192	4.257479	0
-1.780276	5.481558	0	1.618561	1.298508	0	0.924549	1.203705	0	-3.317935	4.271993	0
-1.728100	5.468742	0	1.632639	1.273252	0	0.890698	1.244128	0	-3.332580	4.286673	0
-1.675780	5.453980	0	1.646368	1.248622	0	0.856203	1.285137	0	-3.346170	4.301463	0

-1.623387	5.437366	0	1.659752	1.224609	0	0.821056	1.326701	0	-3.358760	4.316298	0
-1.570989	5.418892	0	1.672809	1.201206	0	0.785258	1.368810	0	-3.370415	4.331141	0
-1.518646	5.398652	0	1.685533	1.178401	0	0.748813	1.411434	0	-3.381171	4.345938	0
-1.466414	5.376591	0	1.697942	1.156196	0	0.711713	1.454552	0	-3.391074	4.360664	0
-1.414332	5.352805	0	1.710027	1.134581	0	0.673990	1.498137	0	-3.400173	4.375293	0
-1.362471	5.327386	0	1.721812	1.113538	0	0.635626	1.542179	0	-3.408506	4.389801	0
-1.310857	5.300293	0	1.733293	1.093076	0	0.596625	1.586645	0	-3.416150	4.404181	0
-1.259511	5.271643	0	1.744486	1.073167	0	0.557027	1.631522	0	-3.423156	4.418407	0
-1.208488	5.241481	0	1.755389	1.053810	0	0.516817	1.676767	0	-3.429582	4.432482	0
-1.157805	5.209856	0	1.766024	1.034989	0	0.475996	1.722380	0	-3.435490	4.446401	0
-1.107488	5.176835	0	1.776374	1.016690	0	0.434598	1.768304	0	-3.440880	4.460183	0
-1.057552	5.142466	0	1.786467	0.998918	0	0.392607	1.814528	0	-3.445742	4.473819	0
-1.008023	5.106808	0	1.796297	0.981641	0	0.350052	1.861035	0	-3.450050	4.487353	0
-0.958898	5.069921	0	1.805884	0.964851	0	0.306931	1.907773	0	-3.453815	4.500783	0
-0.910213	5.031861	0	1.815220	0.948548	0	0.263263	1.954726	0	-3.456998	4.514144	0
-0.861943	4.992682	0	1.824317	0.932692	0	0.219054	2.001863	0	-3.459685	4.527456	0
-0.814135	4.952439	0	1.833177	0.917273	0	0.174327	2.049147	0	-3.462031	4.540693	0
-0.766773	4.911184	0	1.841791	0.902282	0	0.129073	2.096556	0	-3.464191	4.553879	0
-0.719862	4.868956	0	1.850191	0.887700	0	0.083328	2.144055	0	-3.466313	4.566989	0
-0.673429	4.825823	0	1.858360	0.873506	0	0.037093	2.191612	0			
-0.62745	4.781817	0	1.866307	0.859691	0	-0.009606	2.23919	0			

Parallel Fluent Input File,’ flinput’

```
file/confirm-overwrite no
;read  gambit neutral file
file/import/gambit out.neu
/define/boundary-conditions/fluid fluid n n n y 0 0.065 -15 1 0 0 n n n
/grid/modify-zones/matching-tolerance 0.8
/grid/modify-zones/make-periodic 3 4  y y
/define/boundary-conditions/fluid fluid n n n y 0 0.023 -15 1 0 0 n n n
/grid/modify-zones/make-periodic 7 8 y y
/define/boundary-conditions/fluid fluid n n n y 0 0.0415 -0.15 1 0 0 n n n
/grid/check
/define/units pressure atm
/define/materials/change-create air air y ideal-gas n n n n n n n n n
/define/models/solver/density-based-implicit y
/define/models/viscous/ke-realizable? y
/define/models/viscous/near-wall-treatment/non-equilibrium-wall-fn?  y
/define/operating-conditions/operating-pressure 0
/define/boundary-conditions/modify-zone/zone-type 9 pressure-outlet
/define/boundary-conditions/modify-zone/zone-type 2 pressure-inlet
/define/boundary-conditions/pressure-inlet inlet n 0.1 n 0.07 n 300 y n y n 0.9 n -0.85 n 1 n n y 5 300
/define/boundary-conditions/pressure-outlet outlet n 0 n 300 n y y n 1 n 1 n n y n y
/solve/set/under-relaxation/solid 0.8
/solve/set/under-relaxation/turb-viscosity 0.8
/solve/set/discretization-scheme/epsilon 1
```

```
/solve/set/discretization-scheme/k 1
/solve/monitors/residual/convergence-criteria 0.001 0.003 0.0031 0.001 0.0015 0.001 0.001
;/solve/monitors/residual/plot? y
;/solve/initialize/initialize-flow
;read data file
/file/read-data/ mm.dat
/solve/set/courant-number 0.8
it 1000
;solve/set/courant-number  0.9
;it 1000
;Report Total Pressure on airfoil inlet and outlet
/report/surface-integrals/mass-weighted-avg 3 10 () total-pressure yes outlet.srp
;/file/write-data mm.dat
;Exit Fluent
exit
```

REFERENCES

[1] Jennions, I. K., "Elements of Modern Turbomachinery Design System," AGARD-LS- 195. May 1994.

[2] S. Eyi, “Transonic Turbomachinery Blade Design Using Optimization” Aeronautical Engineering Middle East Technical University Ankara, Turkey

[3] Sanz, J. M., "Design of Supercritical Cascades with High Solidity," AIAA Journal. Volume No. 9, September, pp. 1289-1293.

[4] Shelton, M. L., Gregory, B. A., Lamson, S. H., Moses, H. L., Doughty, T. L. and Kiss, T., "Optimization of a Transonic Turbine Airfoil Using Artificial Intelligence, CFD, and Cascade Testing," ASME Paper 93-GT-161, International Gas Turbine and Aero engine Congress and Exposition, May 1993

[5] Goldberg, D.E., Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley, 1989.

[6] Voigt, H. M., “Fuzzy Evolutionary Algorithms”, Technical Report TR-92-038 at ICSI ftp.icsi.berkeley.edu, 1992

[7] Storn R. and Price K.,” Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces”, TR-95-012 March 1995

[8] Lampart, P., Yershov, S., and Rusanov,A (2005),” Increasing Flow efficiency of hp and lp steam turbine stages from numerical optimization”, Of 3D Blading”, Engineering optimization, 37, 145–166

[9] Harrison, S. (1992),”The influences of blade lean on turbine losses”, ASME Journal of Turbomachinery, 114,184–190.

[10] Denton, J. and Xu, L. (1999),”The exploitation of three-dimensional flow in turbomachinery design” Proc Instn Mech Engrs, 213 Parct C.

[11] Lotfi, O., Teixeira, J., Ivey, P., and Kinghorn,I, I.(2006) ”Shape optimization of axial blade fan using genetic algorithms and a 3d Navier-stokes solver “ASME GT2006-90659

[12] Ahn, C. and Kim, K, “Aerodynamic design optimization of a compressor rotor with Navier-Stokes analysis”, A2 Volume 217 2003. Journal of Power and Energy

[13] T Mengistu and W Ghaly,” Aerodynamic optimization of turbomachinery blades using Evolutionary methods and ANN-based surrogate models”, 1 December 2007 Springer Science and business media, LLC 2007

[14] Ueno A. and Dennis B, “Optimization of Flapping Airfoil Motion with Computational Fluid Dynamics”, International Review of Aerospace Engineering, Vol. 2, No. 2 April 2009.

[15] A.Oyama, M.Liou and S Obayashi, “Transonic Axial-Flow Blade Shape Optimization using Evolutionary algorithm and three Dimensional Navier–Stokes Solver”, AIAA Paper 4-6 September 2002, Atlanta Georgia

[16] T. Rogalsky, R. W. Derksen, S. Kocabiyik,” Differential Evolution in Aerodynamic Optimization” University of Manitoba Canada

[17] Nateri. K. Madavan,” Turbomachinery Airfoil Design Optimization using Differential Evolution”, NASA Ames Research Center, Moffett Field, CA USA

[18] Takeishi 1989, Secondary flow schematic diagram

[19] J.Tu, G.H.Yeoh, C. Liu, “COMPUTATIONAL FLUID DYNAMICS: A Practical Approach,” Butterworth-Heinemann, Burlington, Massachusetts, 2008

[20] James D. Ruiz,” Thermal Design Optimization Of Multi-passage Internally Cooled Turbine blades “, University of Texas Arlington Thesis December 2008

[21] “Fluent 6.3 User’s Guide”, Fluent Inc., September 2006

[22] Rainer S, Price K,” Differential Evolution- A simple and efficient adaptive scheme for global optimization over continuous spaces “, TR-95-012 March 1995

[23] H.Watanbe and H.Harada,” Suppression of secondary flows in Turbine Nozzles with controlled stacking shape and Exit Circulation by Inverse Design Method “, ASME Paper 99-GT- 72

[24] Zangeneh et al, 1996, "Suppression of Secondary Flows in a Mixed-Flow Pump Impeller by Application of Three-Dimensional Inverse Design Method: Part 1 - Design and numerical Validation", ASME Journal of Turbomachinery, Vol. 118, pp. 536-543.

[25] Go to et al, 1996, "Suppression of Secondary Flows in a Mixed-Flow Pump Impeller by Application of Three-Dimensional Inverse Design Method: Part 2 - Experimental Validation", ASME Journal of Turbomachinery, Vol. 118, pp. 544-551

[26] Jani Keranen, “Comparison of Compound Lean Nozzles and Controlled Flow Nozzles at Off-Design “, Department of Energy and Environmental Technology, Tapolantie 262 FIN-47490 MANKALA

BIOGRAPHICAL INFORMATION

Darshak P Joshi has born in Gujarat India in 1986. He has started his student career in fall of 2003 as an Aeronautical Engineer in Sardar Vallabhbhai Patel Institute of Technology, Vasad in Gujarat India. He also took part in robot competitions and won 2nd prize in the national level technical competition. He also took educational training in Hindustan Aeronautics Limited in NASIK and KORAPUT division, which is India's largest Aircraft manufacturing company. Then, He received his undergraduate degree in Aeronautical Engineering with distinction in Gujarat University in May 2007.

Then, He moved to Arlington in USA to pursue his Masters in Aerospace Engineering in University of Texas Arlington. He joined Prof Dr. Brian Dennis for further studies in the Field of CFD. Darshak's primary focus is in the field of turbomachinery and shape optimization of turbine blade using differential evolution method. Later, He graduated in April 2010 with his Master of Science in Aerospace Engineering.