3-2-2021

Body Force Modeling of Axial Turbomachines Without Calibration

Syamak Pazireh
University of Windsor

Recommended Citation

Pazireh, Syamak, "Body Force Modeling of Axial Turbomachines Without Calibration" (2021). *Electronic Theses and Dissertations*. 8530.
https://scholar.uwindsor.ca/etd/8530

Body Force Modeling of Axial Turbomachines Without Calibration

by

Syamak Pazireh

A Dissertation
Submitted to the Faculty of Graduate Studies
through the Department of Mechanical, Automotive and Materials Engineering
in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy
at the
University of Windsor

Windsor, Ontario, Canada
2020

Body Force Modeling of Axial Turbomachines Without Calibration

by

Syamak Pazireh

APPROVED BY:

X. Gao, External Examiner
Colorado State University

T. Bolisetti
Department of Civil and Environmental Engineering

G. Rankin
Department of Mechanical, Automotive and Materials Engineering

---

O. Jianu
Department of Mechanical, Automotive and Materials Engineering

---

J. Defoe, Advisor
Department of Mechanical, Automotive and Materials Engineering

17 December, 2020

Declaration of Originality

I hereby certify that I am the sole author of this dissertation and that no part of this dissertation has been published or submitted for publication.

I certify that, to the best of my knowledge, my dissertation does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my dissertation, published or otherwise, are fully acknowledged in accordance with the standard referencing practices.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

This dissertation proposes a new method of modelling turbomachinery blade boundary layer and shock losses using the body force method. Body force methods are used to model fan/compressor performance at a lower computational cost than unsteady Reynolds-Averaged Navier-Stokes (URANS) computations in non-uniform inflows. Most loss modelling approaches in the literature require calibration. Some recent work has shown the use of non-calibrated methods for entropy generation calculations. However, recent non-calibrated methods cannot estimate flow losses with boundary layer separation. In this dissertation, an artificial neural network has been developed and trained to analytically relate the blade geometry and flow regime to the boundary layer momentum thickness at the trailing edge. The trailing edge momentum thickness is used in a body force loss model that accounts for the relative total pressure drop. This model is capable of predicting the loss at off-design conditions. The accuracy of the model is over 90% in 2D cascades. The model is then applied to the NASA rotor 67 compressor blade row. The model captures the high entropy generation near the tip region for uniform and non-uniform inflows. For non-uniform inflow, it predicts the isentropic efficiency to within 1% compared to a URANS computation.

Acknowledgments

I would like to thank the following people, without whom I would not have been able to complete this research: My supervisor Dr. Jeff Defoe for his enthusiasm for the project, for his support, encouragement and patience. The meetings and conversations were vital in inspiring me to think outside the box. I am also thankful to the faculty of Engineering and all my committee members: Dr. Rankin, Dr. Jianu, and Dr. Bolisetti and my external examiner, Dr. Gao for all the considerate guidance. Further, I want to express my sincere gratitude to the Consortim for Aerospace Research and Innovation in Canada (CARIC), MITACS, and the Natural Sciences and Engineering Research Council of Canada (NSERC), Bombardier Aerospace, and Pratt & Whitney Canada for funding. I would also like to acknowledge researchers at the Whittle Laboratory, University of Cambridge for providing scripts used to execute MISES automatically. Lastly, I would like to thank my Spouse Elnaz for helping me to have an organized mind in the way of working on my thesis and my beautiful daughter Lena that her presence is the greatest hope and motivation for the all progress of my life.

Contents

List of Figures

List of Tables

Nomenclature

B number of blades in a row

$\boldsymbol{b}$ bias coefficient vector

c chord

$c_{\mathcal{D}}$ dissipation coefficient

C_D drag coefficient

c_f skin friction coefficient

c_τ shear coefficient

c_v specific heat at constant volume

d deviation

F neural network function

f body force per unit volume

H boundary layer shape parameter

H^* kinetic energy shape parameter

H_k kinetic shape parameter

h pitch - specific enthalpy

i incidence

M Mach number

$\hat{\boldsymbol{n}}$ camber surface normal vector

$\tilde{n}$ transition amplification ratio

p pressure

R specific gas constant

r radius

R Gas constant

Re_θ Reynolds number based on θ

$\boldsymbol{\mathcal{S}}$ blade loading model parameters

s entropy - spacing

$\dot{S}$ entropy flux per unit surface

T Temperature

$\hat{t}$ blade thickness normalized by chord

u velocity

V Absolute frame velocity

$\boldsymbol{V}$ Absolute frame velocity vector

w coefficient matrix

W Relative frame velocity

$\boldsymbol{X}$ input vector for artificial neural network

$\boldsymbol{y}$ output vector for artificial neural network

$\hat{z}$ blade camber normalized by chord

α flow angle from the axial direction

α^* angle of attack

δ boundary layer thickness

δ^* boundary layer displacement thickness

θ boundary layer momentum thickness

κ camber angle from the axial direction

ρ density

π total pressure ratio

σ solidity

ζ streamwise coordinate

ω loss coefficient

Ω Rotational speed

SUBSCRIPTS

0 free-stream quantity

e boundary layer edge

IA isolated airfoil

LE leading edge

max maximum

min minimum

n normal component

PS pressure side

p parallel component

SS suction side

t total

TE trailing edge

x x-component

y y-component

∞ ambient condition

SUPERSCRIPTS

X mixed-out average quantity

Chapter 1

Introduction

The existing applications of blade-geometry-included computational fluid dynamics (CFD) for axial turbomachinery, in general, require significant computational resources to model the flow behaviour in a full annulus non-uniform inflow condition. URANS (Unsteady Reynolds-Averaged Navier-Stokes) models are among the most-used approaches which take into account the blades geometry in the CFD solver in applications involving non-uniform/distorted flows. URANS simulations with the detailed geometry of the blade rows in the computational domain use sliding interfaces between rotating and stationary regions [1]. This approach can accurately predict the flow within the rows. This enables the complete details of the interactions to be modelled, including performance penalties for the blade rows due to the non-uniform inflows. However, such simulations are expensive, requiring on the order of tens to around one hundred million cells [1]. This complexity in modelling makes it difficult to use in the design process of new-generation turbofan engines when distorted inflows occur at the design conditions [2]. In addition, when the design of the aircraft comes into play, the aerodynamic interaction of the inlet flow of the engine and fan is particularly important. For this reason, the demand for more straightforward modelling with less computational cost that can accurately capture the relevant physics is increasing [3].

The simplest way of accounting for engine effects on the external aerodynamics around the wing, which is sometimes used by industry, is the "boundary condition" approach in which the solution domain sees the engine inlet and outlet as the global domain boundaries [4]. A schematic of a computational domain with the engine inlet and outlet as the zone boundaries is shown in Fig. 1-1. This approach simulates the engine/airframe interactions. This method does not focus on the flow inside the engine. Thus, the major disadvantage of this method is that in the case of non-uniform flow at the engine inlet, the effects on the external flow cannot be observed if the engine inlet is short enough to cause interaction between the fan and the inlet stream.

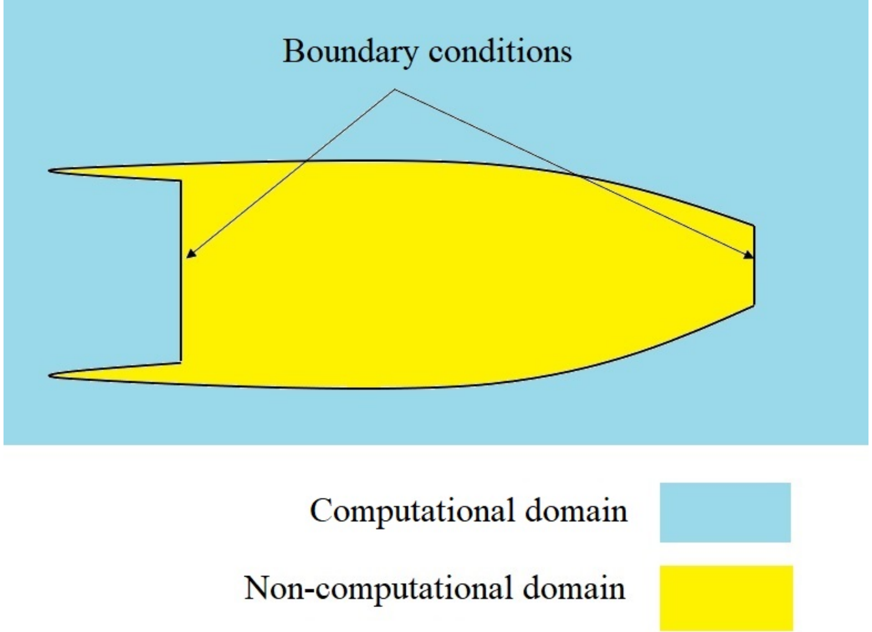


Figure 1-1: Boundary conditions for simple simulation for fan/airframe interaction

To get the fan/compressor response in terms of the fan/engine interactions, the "actuator disk model" is a simplified model that has received widespread use in simulating of fan/inlet-distorted problems [5–8]. This model, as a zero-dimensional strategy, is capable of capturing the responses to dynamic phenomena such as a local stall or distorted inlet flows [9]. However, this approach needs fan/compressor performance characteristic maps to accurately predict the pressure rise and flow turning in a thin disk within the engine [10]. This fact reduces the flexibility of the method in evaluating any new design, since in the design procedure, each new fan must utilize the RANS method in various conditions to get the required data for the actuator disk model [8]. Alternatively, if the radial equilibrium assumption is used, the tangential velocity at the trailing edge can be calculated for an axial compressor [11]. In the radial equilibrium approach, the radial pressure gradient is in a balance with the effect of centripetal acceleration due to the tangential component of velocity. However, this assumption would not be appropriate for circumferentially distorted inflows; it is only valid in axisymmetric flow. Figure 1-2 demonstrates different fan modelling approaches with the corresponding fidelity and computational costs.

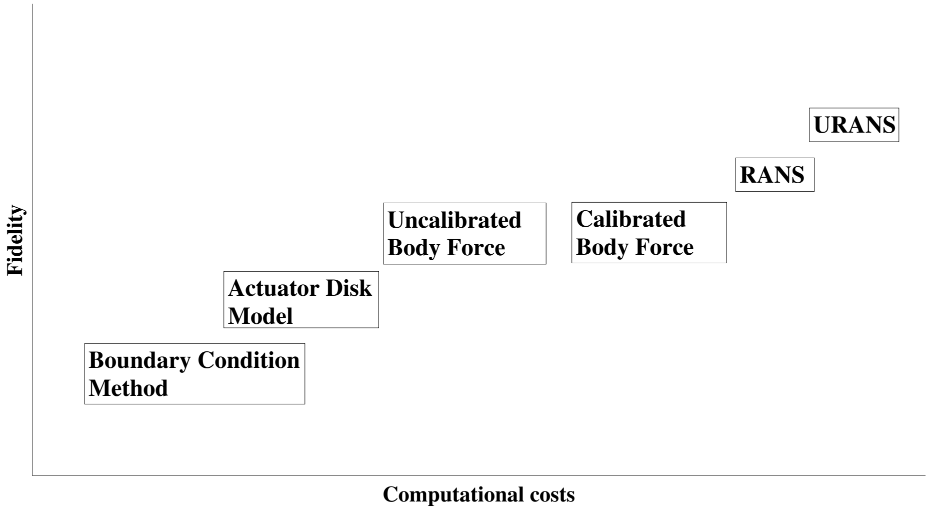


Figure 1-2: Fidelity vs. computational cost for fan modelling approaches

The body force method lies between the actuator disk model (ADM) and bladed-CFD methods in terms of both cost and fidelity. This method requires more geometrical details compared to the ADM, but it still has much of its simplicity [12]. In the body force model, flow within the fan is considered in the CFD solver, but the rotor/stator geometries are not added in the calculations. Instead, body forces are used as source terms in the governing equations in the rotor/stator swept volumes, which account for the forces representing the blade impacts as if they were present. Figure 1-3 shows the bladed CFD zone and the body force CFD zone. As shown on the right side of the figure, the body force consists of two main parts: the parallel force acting opposite to the local relative velocity direction accounting for entropy rise [13] and the part normal to the relative streamline which accounts for both flow turning and the total pressure rise [12].

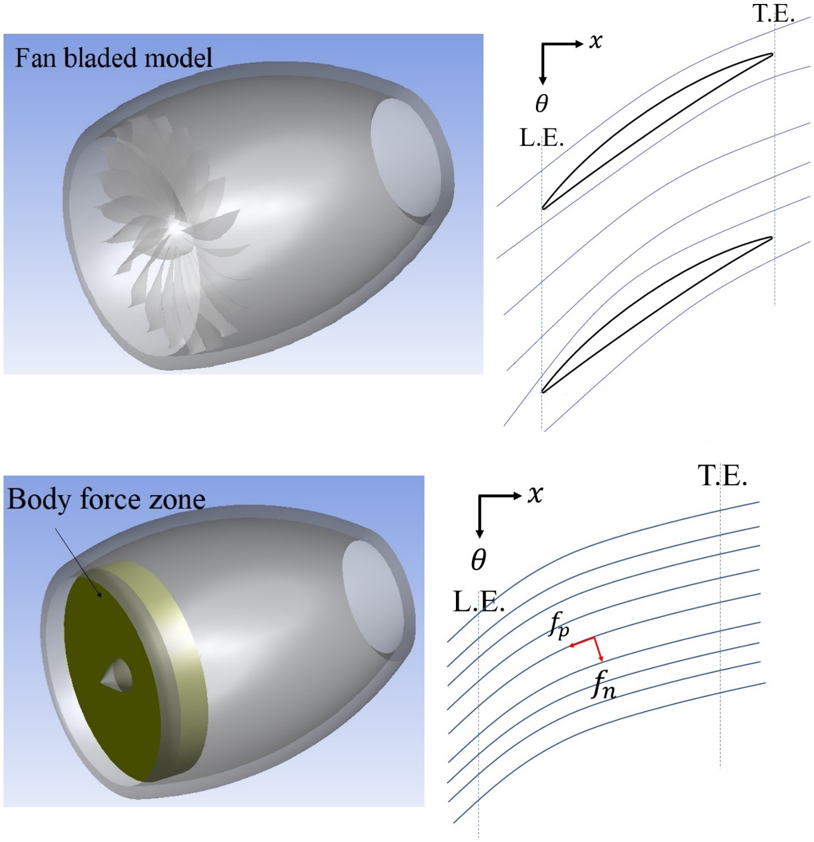


Figure 1-3: Flow field through bladed domain (upper) vs. through body force zone (lower)

Body force modelling is an appropriate alternative to bladed URANS for non-uniform inflow simulations as it does not require a time-resolved computation. This massively reduces computational costs, typically by about 2 orders of magnitude, since even with the access to large numbers of computing cores, all solvers have practical limits on parallelization: one can't allocate too few cells per core or else the computation becomes very inefficient. Thus, the body force seems a promising approach.

The first proposed model of the body force method by Gong [12] in compressors included calibrating the source term using coefficients that are generally derived from the design point conditions utilizing the RANS method. These suggested forces with separate terms met the requirements of overall pressure rise and flow turning and reasonable response to circumferential unsteadiness. Following the studies on body forces, research shifted to ways of eliminating calibration. Recently, a model for flow turning has been analytically developed by Hall [14] which does not need any calibration. Separately, a simple non-calibrated total pressure drop source term was introduced by Godard et al. [15]. In another model introduced by Guo et al. [16], simple non-calibrated viscous models have been introduced which lack accuracy off-design due to their inability to capture highly coupled viscous-inviscid interactions. A detailed literature review of the calibrated and non-calibrated fan models is found in Chapter 2. Nevertheless, there is still a need for non-calibrated viscous and shock wave induced body forces to act as parallel source terms which can accurately predict both on- and off-design performance.

1.1 Calibration of Body Forces

The early body force model in compressor simulations by Gong [12] employed momentum source terms accounting for flow turning based on angular momentum change. That model also included entropy generation in the form of a parallel force. However, flow turning and loss generation models in Gong's approach required RANS models to get calibration parameters from an operating point (usually the design point). In an improved model by Chima [17], deviation calibration parameters were proposed to be obtained from a near-stall operating point to yield better performance estimates in stall studies. The model details are explained in the literature review in Chapter 2.

This initial approach to the use of volumetric (body) forces in fan/compressor analysis still required the analysis of the RANS model at an operational point. This dependency on calibrating the off-design conditions is a weakness for the volumetric force model for purposes where low-cost computations are needed. In addition, in terms of the design process flexibility, any design process requires assessing the response of the multiple designs to the non-uniform inflows and then choosing the most efficient one. Assessment of the performance of any design requires calibration using the corresponding RANS simulations.

Xu [18] in 2003, introduced a new viscous force in body force simulations based on the drag coefficient. That work is a calibrated model as it needs RANS calculations to obtain the drag coefficients. Tucker [19], in 2011, introduced a review of models where large eddy simulations (LES) can be used for body force modelling calibration. The viscous model in the calibrated body force was developed by Peters [20] in 2014. His model shows an improvement in accuracy near stall and choke conditions compared to Gong's model. That model used the peak-efficiency operating point for calibration parameters. Hill and Defoe [21] in 2018 used Peters' viscous model and added off-peak efficiency for calibration parameters to capture choke condition losses. The details of the studies, as mentioned above, are discussed in Chapter 2. Consequently, over time, research has shifted to reducing the dependency on calibrating the model, so that for the volumetric force causing flow turning, an analytical model depending solely on the local deviation has been introduced by Hall et al. [14]. In terms of the loss model, a newer model has been presented by Thollet in 2017 [22] and used by Benichou et al. [23] and Godard et al. in 2019 [15]. This new loss model does not need RANS calibration. Figure 1-4 gives a hierarchy for the body force models in terms of required calibration.

Previous studies have focused less on conveying the details of the blade boundary layer and shock-induced losses in body forces without the need for calibration.

The non-calibrated loss model needs to account for the blade surface boundary layer's properties as it relates to the velocity distribution at each blade section. The boundary layer properties in the streamwise direction along a cross-section of the blade can be obtained by considering the boundary layer equations, which are in the form of ordinary differential equations. These equations provide local momentum and displacement thicknesses from leading to trailing edge. Entropy generation along the streamline due to the boundary layers may be calculated by having the aforementioned boundary layer properties. In addition, the displacement thickness can account for the boundary layer blockage added to the blade in the flow field. Details of the boundary layer governing equations are introduced in Chapter 2. The direct ways to compute

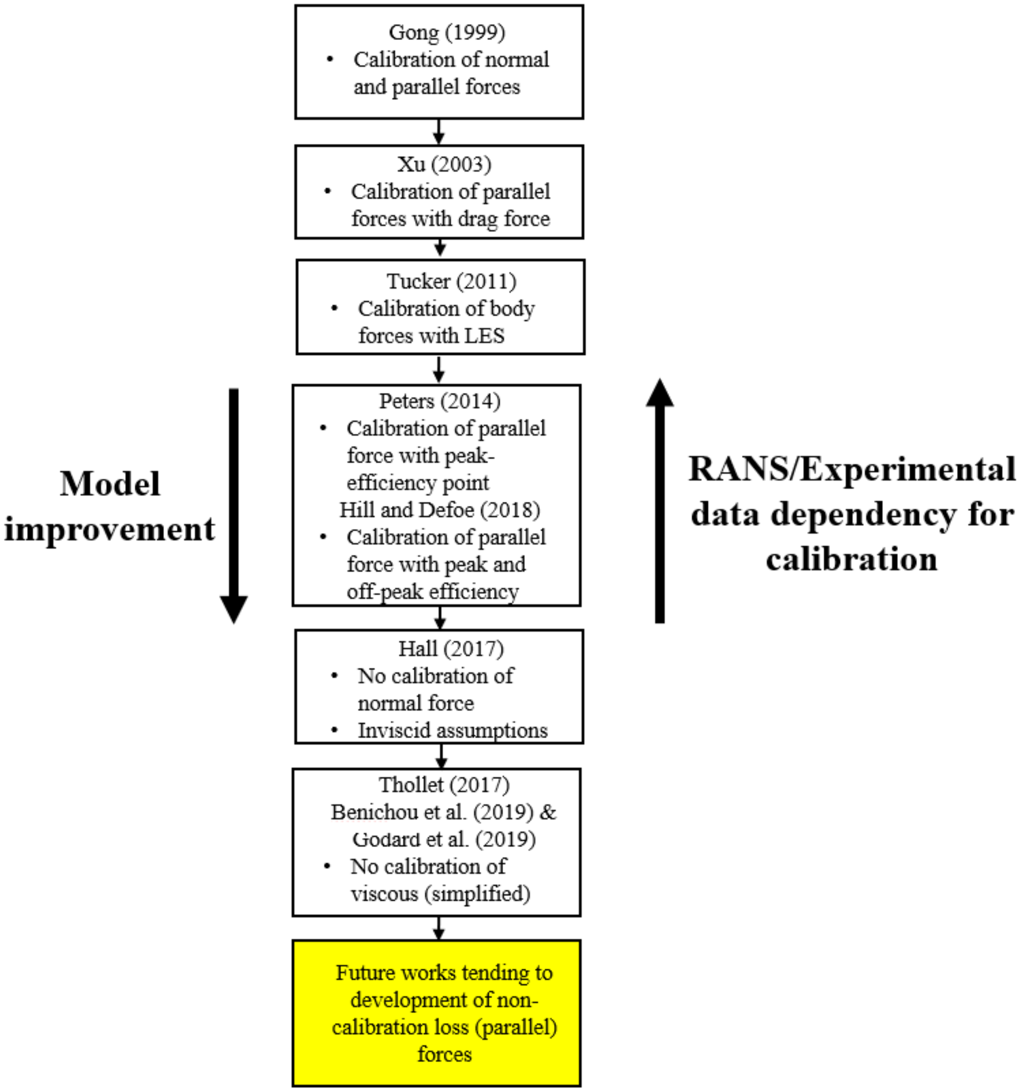


Figure 1-4: Hierarchy for improvement of calibration models in body force methods

the boundary layer equation use iterative procedures that are introduced in Chapter 2. The iterative procedures are not suitable for body force modelling approaches. These require chordwise velocity-distribution information on either side of the blade sections and are not local-based approaches. In addition, these iterative methods add to the cost of calculations, while the aim in non-calibrated body force models is to reduce the computational costs. Thus, there are challenges in calculations of boundary layer equations in non-calibrated body force simulations. These challenges are described in the next Section.

This study seeks a method that uses the boundary-layer-calculation approach to achieve a no-calibration loss body force model. In the no-calibration body force approach, the entropy generation is calculated directly based on the flow regime and real boundary layer properties along any streamline. It does not need RANS/experimental data to calibrate the body force in off-design conditions. The available approaches to solve the governing equations of the boundary layer and the challenges to use these solution methods in a body force modelling approach will be discussed in Chapter 2. A direct approach to obtaining the viscous losses in body force CFD is to solve the integral boundary layer equations along each streamline within each blade row, but this adds computational cost and becomes impractical for conditions at which the blade surface boundary layer(s) is/are separated. In Chapter 2, this is explained in detail. A less direct method is thus needed. So, an artificial neural network is considered to be an indirect replacement. Thus, the challenge for direct calculations of boundary layer equations in body force modelling will be explained in Section 1.2, and a remedy with artificial neural network will be introduced to be a tool to get rid of simultaneous boundary layer calculations but that still yields a boundary layer solution for non-calibrated models. Consequently, it will be described in Chapter 4 that an artificial neural network based on a large dataset of boundary layer and flow information can be a promising approach to provide the flow field boundary layer solutions with an analytical formula that uses local streamline properties to predict the viscous effects' entropy generation.

1.2 Challenges

To capture the local viscous entropy generation in a body force model, it is required to have the local boundary layer momentum thickness or dissipation coefficient. The details of this dependency are introduced in Chapter 2. The local loss needs both the precise velocity distribution around the blade and the simultaneous solution of the boundary layer differential equations. In body force modelling, due to the absence of the blade, these velocity and boundary layer properties for either side of the blade are fictitious and are estimated to yield the momentum loss as a source term in the

Navier-Stokes equations.

A no-calibration viscous model in body force computations seems to require solving the boundary layer equations along relative streamlines. Denton's viscous loss model [24] shows that local entropy generation is dependent on the cube of boundary layer edge velocity and dissipation coefficient. The detail of the model is introduced in Chapter 2. In addition, to obtain the local dissipation coefficient, local edge velocity is needed to be taken into account in the boundary layer equations. Thus, in body force computations, an accurate local normal force should represent the actual pressure difference between the suction and pressure sides of the blade. The pressure difference is related to the local edge velocities using Bernoulli's equation along a relative-frame streamline. The relationship between the local velocities on the suction and pressure sides and the normal force is introduced in Chapter 3. In addition, the viscous model predicts the local boundary layer displacement thickness, which alters the effective airfoil geometry and, consequently, velocity distribution. So, there is a coupling between the normal force and the viscous model. Figure 1-5 illustrates the coupling dependency between flow turning and loss body forces for a no-calibration method.

Estimation of boundary layer edge quantities in a body force calculation has several challenges. In terms of the viscous loss calculations, additional governing equations are needed to convect viscous information downstream along relative streamlines within rotor/stators. These add to the computational costs. Besides, extra variables are required to be stored in the computations. This leads to a need for extra memory. In addition, the implementation of such calculations in commercial software is complicated. Also, the calculations need iterative approaches to be robust. These approaches require "sub-iterative" calculations within the CFD solution iterations. This leads to significant extra computation costs. Furthermore, the coupling between blade surface slope and displacement thickness in normal force computations causes even more costs.

Another challenge in this regard is that the current state of the art normal force is based on local deviation angle. The airfoil total camber angle is not used as the

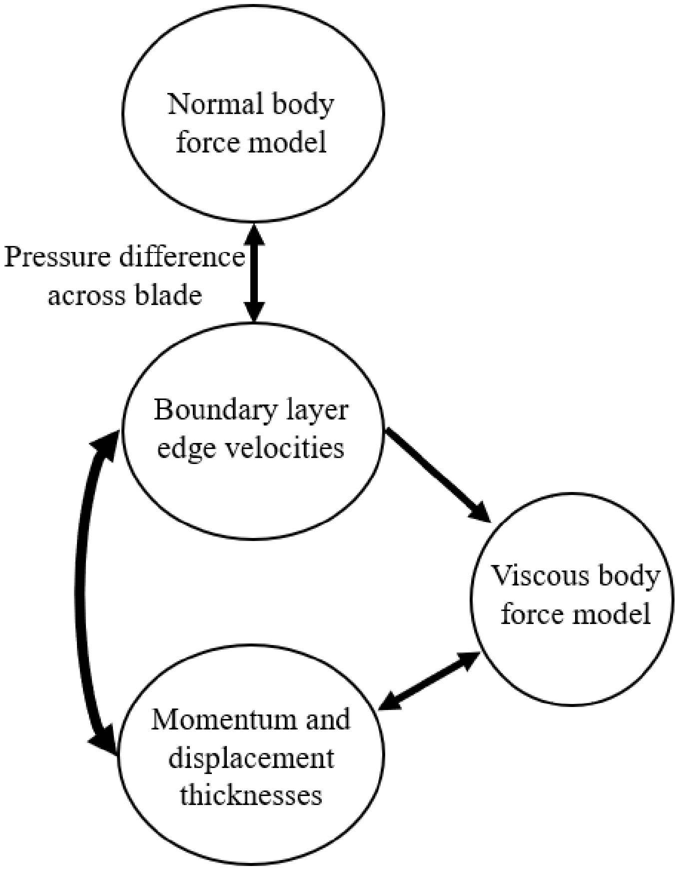


Figure 1-5: Mutual dependence of viscous body force model and blade loading

models only depend on the local camber surface shape. An alternative normal force modelling approach is potential flow theory which can capture the flow around the blade, but this requires airfoil section camber and thickness data. These models can successfully assess the velocity distribution in 2D for uniform inflows. However, in 3D cases, the specific blade shape a given relative streamline will see is not known a priori, so it is not possible to determine the required data for the velocity distribution along the chord. In addition, the accuracy of potential flow models reduces when the inflow is non-uniform.

1.3 High-Level Approach

Based on the explanations provided so far, it is clear that one must first eliminate the dependence on calibration for the viscous loss in body force models because this method requires CFD analysis with the presence of blade geometry for design conditions. Secondly, if precise calculations are required to produce the boundary layer's entropy generation, it seems necessary to calculate the coupled differential equations of the boundary layer. This method also requires the exact distribution of velocity around the blade, which current body force models cannot provide, and at the same time solving these equations reduces the speed of analysis of the body force method.

Chapter 2 shows that the momentum thickness of the boundary layer at the trailing edge can be related to the total pressure drop information and can be used to calculate the parallel force. As a result, there is a need for a fast method that can predict momentum thickness according to the inflow regime and some minimal geometric parameters. Artificial neural network as a surrogate modelling approach is a promising way to be able to accomplish this, as also introduced in Chapter 2.

In this dissertation a parallel force based on the highly-coupled velocity-boundary-layer physics is introduced that can be implemented at design and off-design conditions for fans/compressors. No calibration specific to the blade rows being modelled is required. Artificial neural network is an encouraging approach that paves the way for analysing coupled physics. For this purpose, an artificial neural network (ANN) is capable of learning the correlation between the physical and geometrical parameters and the boundary layer characteristics. Artificial neural networks train and learn the contributions of the inputs at producing the outputs. A big dataset accounting for real compressor operating conditions is generated, and an analytical equation for predicting local loss results. The artificial neural network can overcome the mathematical singularity problems in the boundary layer equatons when flow separation exists and avoids additional transport equations. This solves both the cost and complexity issues identified in the previous section.

For this study, a fully-simultaneous solver (MISES [25]) was used to generate boundary layer data for a wide range of compressor airfoils operating at different conditions to be utilized in an artificial neural network. The range of parameters considered in the artificial neural network are discussed in Chapter 4. MISES uses Euler CFD calculations for the flow outside the boundary layer, and the equations are coupled with viscous governing equations that are simultaneously solved using a Newton-Raphson solver. The equations are solved for all boundary nodes, and a finite-difference method is used for spatial marching along the airfoil.

The artificial neural network generates an analytical equation that relates the geometry and physical conditions to trailing edge momentum thickness. Consequently, a new parallel body force accounting for viscosity effects is provided so that it avoids dependence on the calibration. The approach is described in detail and validated in 2D in Chapter 4. It is shown to work in 3D in both uniform and non-uniform flow in Chapter 5.

1.4 Overview of Contributions

The contributions of this dissertation, which will be revisited in more detail in Chapter 6, are:

1. A new potential flow based analytical model has been proposed for 2D cascades to be used as a new flow turning body force. This model works with high accuracy for staggered airfoils in cascades. The model, however, is not suitable for 3D blade simulations.

2. Based on Youngren's relationship [26] between entropy generation and trailing edge momentum thickness, a blade profile viscous loss body force is formulated. The model accurately predicts loss given the correct trailing edge momentum thickness.

3. An artificial neural network is shown for the first time to be able to predict trailing edge momentum thickness when trained on a large dataset of 2D compressor cascade solutions. The dataset itself and the resulting neural network

are expected to be of use to the technical community. The model works well overall and captures entropy generation trends.

4. In the supersonic flow regime it was found that the use of a shock loss model has a high impact on entropy generation and spanwise mass flux predictions. A new shock loss model using Denton's shock entropy generation approach [24] is introduced. The loss prediction improvement is considerable at the design rotational speed for the NASA rotor 67 blades.

5. In non-uniform inflows, the new model over-predicts the efficiency penalty by 5%. Nevertheless, the new non-calibrated loss model is capable of recognizing azimuthal and spanwise entropy generation trends correctly, which is a promising improvement over previous non-calibrated loss models.

1.5 Dissertation Outline

Background and review of the relevant literature on non-uniform and distorted inlet flows, actuator disk models, body forces, viscous models, potential flow theory, and artificial neural networks in turbomachinery are provided in Chapter 2. Chapter 3 presents a new flow turning model for 2D staggered cascades. It examines the performance of Denton's loss model with constant dissipation coefficient. Chapter 4 introduces Youngren's momentum defect model as the total pressure loss in body force. It describes the architecture of the artificial neural network and input variables for calculating the momentum thickness. The assessment of the new model for capturing the trailing edge momentum thickness is also detailed in Chapter 4. Chapter 5 assesses the performance of the models in 3D rotor blades for both uniform and non-uniform inflows. Conclusions and recommendations for future work are given in Chapter 6.

Chapter 2

Literature Review

This chapter establishes the state of the art concerning the phenomena of non-uniform flows in fan/compressors, relevant modelling methodologies, and distinguishes the research gaps in the body force loss models that this work aims to fill.

2.1 Early Fan/Compressor Non-Uniform Inflow Studies

Jang et al. [27] have shown that flow distortion at the fan/compressor leading-edge vicinity reduces the performance and stability of the compression system. They considered the distortions that are created by a spinner nose and analyzed the flow with numerical simulations. In a study by Boldman et al. [28], it was shown that the blockage effect of the fan creates the lip separation and distortion at duct inlets. That research was conducted by experimental instruments on a low-subsonic fan test rig and the results showed that a similar stationary blockage which replaces the fan provides the same effects on the inlet distortions.

Plas et al. [29] worked on the boundary layer ingestion effects on turbofan engine fuel consumption. The distortion transfer across the fan was studied. They investigated the effect of inflow non-uniformity on fuel consumption.

Kemp and Sears [30] studied non-uniform flow at blade trailing edges that comes from the wakes. They investigated the effect of drag on the wake size and non-uniform parameters of the flow. Greitzer [31] studied the local distortions such as stall on the performance of compressors. He introduced a model that can capture the flow inlet distortions and its effects on stall.

There have been many studies on the prediction and simulation of non-uniform flows within compressor/fans to assess stability criteria. Mazzawy [32] developed a model based on the deviation from undistorted flows to analyse the stability. That model followed the response to the circumferential distortion using multiple parallel compressor segments. The work equation is implemented on two segments. The deficiency of this model is that it depends on the prescribed non-uniform static pressure and has no general applicability.

Later, Hah et al. [33] implemented URANS simulation for the transonic compressor with time-varying inlet conditions. The URANS simulations are beneficial in tracing the shock-boundary layer interaction.

Following the extensive research on non-uniform flows in turbomachines and revealing their importance in interaction with the fan/compressor, the need for simpler models was affirmed. In this respect, the initial thoughts were drawn to the use of averaging methods. Cumptsy and Horlock [34] showed that the features of the non-uniform inflows in turbomachinery could be used with an averaging method that preserves the non-uniform quantities but make the simulations simpler. They introduced formalized averaging methods to be used in non-uniform flows in turbomachinery. Following the push of studies towards more straightforward methods in turbomachines in non-uniform flow, actuator disk model methods were proposed.

2.2 Actuator Disk Model

An actual disk model (ADM) is a zero-thickness plane presenting the pressure rise and the swirl using radial equilibrium equations and the prescribed work. One of the original ideas to study non-uniform flow response with actuator disks is Yeh's [35] in 1959. That model was a mathematical treatment of flow upstream and downstream of the disk in which cascade calculations were implemented and an analytical solution for both circumferential and radial distortions investigated. The model considered stall propagation. This model uses the small perturbed velocity assumption. The study mentioned above was confined to a single row simulation. Nevertheless, the method yields qualitative insight into the response of blades rows to distortion.

This field of study was continued in 1962 by Hawthorne [11]. He developed the first actuator disk model considering 2D cylindrical coordinates and extracted an analytical model and applied it for compressible and incompressible flows. Following the proposed method by Hawthorne, in 1978 Hawthorne et al. [36] developed the non-uniform actuator disk model in which the model was based on vorticity analysis and the changes of vorticities across the disk were considered. This study showed that radius-based calculations are not enough to examine the compressor's response to non-uniform flow, and three-dimensional effects must be taken into account to capture the physics.

In recent years, the actuator disk model has been used in analysing fan flows. Van der Spuy et al. [37] in 2011 used CFD with an actuator disk model including source terms for blade forces to investigate low flow rate fans. At low flow rates, the radial flow distortions play important role in the performance. The authors extended an empirical actuator disk model with calibration for each blade row to take into account the radial direction variations. Thus, they concluded that the actuator disk model needs calibration to avoid the under-prediction issue within the modelling process.

Developments in the field of actuator disk modelling moved to fan-nacelle interactions. In 2011 Thouault et al. [3] studied the inlet lip separation and fan interactions using an actuator disk model. The study focused on the recognition of affecting physical and geometrical parameters on the boundary layer separation. This model is a combined form of "boundary condition" approach ADM in which the fan exit is the outer domain inlet boundary, and the inlet of the fan is considered the exit of the outer domain in CFD. However, based on the stator exit conditions where the fan outflow tends to be uniform, the fan stator exit (domain inlet) condition is prescribed from URANS simulations, and the simulations capture the distorted fan inflow. Although the model has good accuracy in tracking the fan inlet separation physics, it still has the limits of the “boundary condition” approach and depends on the URANS simulations or fan performance maps. In addition, it does not capture the flow characteristics within the rotor/stator.

Akturk and Camci [8] studied fan performance in response to inlet distortions using a radial equilibrium based actuator disk model. The model uses a prescribed pressure jump as the rotor response to the flow and adds it to momentum equations. The pressure rise is determined by the flow properties on the fan surface and the velocity triangle approach to account for the swirl. This model is confined to the incompressible flow, and there is no mechanism to take losses into account.

Other than implementing a total pressure rise coming from RANS or performance characteristic maps to predict the flow turning at the trailing edge by radial equilibrium methods in ADMs, plenum-throttling dynamic models have been proposed for stall and post-stall transients in compressors [5] and [38]. These models predict fan performance in non-uniform cases but cannot still capture fan-inlet interaction dynamics.

It seems that not only do ADMs not capture losses in most cases but also they have some constraints for non-uniform flows based on the assumptions. Other recent papers have looked at more sophisticated implementations of actuator disks, including hybrid approaches in which body forces are also used. However, all these approaches have some limitations, and a recent review paper by Godard [39] directly compares the ability of body force and actuator disk approaches to capture the fan aerodynamics in ultra-high bypass ratio (UHBR), short nacelle configurations. The authors conclude that body force models can capture inlet separations and fan-inlet coupling far more accurately than actuator disk methods, and with a two-order-of-magnitude computational cost reduction compared to full-wheel, unsteady RANS computations. In addition, the body force model appears more promising when the loss computations play a role in the predictions.

The approach of this dissertation could be used in an ADM. However, the ADM’s limitation with regards to capturing fan-distortion interaction limits the utility of such an implementation and thus the focus for the remainder of this dissertation is on body force methods.

2.3 Body Force Methods

From Godard's recent analysis it is clear that body force models are the better approach for capturing inlet-fan coupling in short nacelle configurations. Body force modelling deals with no-bladed CFD in which the fan/compressor is modelled with body forces acting on the fluid in the rotor/stator swept volumes.

A body force model captures the locally pitchwise-averaged performance of a fan by distributing momentum and energy (and sometimes mass) source terms throughout the swept volumes of the blade rows. This allows for the effects of radial flows, local responses to distortion, and other features of real turbomachine response to be captured so long as the distortion wavelengths are long compared to the blade pitch. The idea originates from Marble [40] and was later taken up by Gong [12] where it was used to study stall inception. In Gong's approach, the body force vector is broken into three parts, one turning the flow without work being done another solely producing total pressure rise and the third one accounting for total pressure loss:

$$\boldsymbol{f} = \boldsymbol{f_{turning}} + \boldsymbol{f_{\pi}} + \boldsymbol{f_{p}} \tag{2.1}$$

In Eq. 2.1 $\boldsymbol{f_p}$ is the total pressure loss term which acts in the streamline direction but opposite to the local relative velocity, $\boldsymbol{f_\pi}$ is the total pressure increase due to the work acting on the axial direction and $\boldsymbol{f_{turning}}$ acts normal to the local relative velocity vector $\boldsymbol{W}$ to turn the flow towards the camberline angle in the relative frame but has no effect on the total pressure. The circumferential component of this force is calculated as:

$$f_{turning,\theta} = CV_x(r\Omega + V_x tan(\kappa) - V_\theta) \tag{2.2}$$

where r is local radius, Ω is the rotational speed, x and θ are axial and circumferential directions, respectively. The constant C is calibrated from RANS simulations to ensure the correct deviation angle at the design flow coefficient and κ is the local blade metal angle. This shows that Gong's model needs bladed RANS results and geometrical data to calibrate the deviation. Furthermore, the parallel force in Gong's model is calculated as:

$$f_{p,i} = -\frac{K_p}{h} V_i W_i \tag{2.3}$$

where K_p is a constant coming from a RANS model. Subscript i is the *ith* component. V and W are absolute and relative frame velocities, respectively and h is the cascade pitch which is the spacing between two successive blades in a row.

The Euler equations with no mass sources are:

$$\nabla.(\rho \boldsymbol{V}) = 0 \tag{2.4}$$

$$\rho \boldsymbol{V}.\nabla(\boldsymbol{V}) + \nabla p = \boldsymbol{f_n} + \boldsymbol{f_p} \tag{2.5}$$

$$\rho \boldsymbol{V}.\nabla(h_t) = \rho r \Omega f_\theta - \boldsymbol{W}.\boldsymbol{f_p} \tag{2.6}$$

where $\boldsymbol{f_n}$ and $\boldsymbol{f_p}$ are the source terms accounting for normal and loss (viscous/shock) parallel forces, respectively, with the unit of force per volume (in SI $\frac{N}{m^3}$). Also, h_t is the specific total enthalpy and ρ is the fluid density. The first term on the right hand side of Eq. 2.6 refers to the work input by the rotor rotation and circumferential force and the second term corresponds to work done by viscous forces.

Later in 2003 Xu changed the viscous body force from Gong's model to a drag coefficient based approach [18]:

$$f_p = -C_D \frac{0.5 \rho W^2 S}{Vol} \tag{2.7}$$

where C_D is drag coefficient and there is no explicit equation to predict it (comes from bladed RANS simulations), S is the blade camber surface area and Vol is body force zone volume for a single passage. Xu showed that a simple drag coefficient model in a non-uniform flow is quite adequate to model time mean effects. However, for the cases when separations are present variations in the drag coefficient may become large and more detailed modelling may be required. The reason for this is that the drag coefficient is obtained directly by integrating the surface friction coefficient.

Defoe and Spakovszky have shown the capability of body force models for capturing the noise in boundary layer ingestion flows [41–43]. The results have shown that the body force approach is able to capture the rotor shock noise and inlet boundary layer ingestion interactions.

Studies of loss model advanced with the improvement of Gong's model by Peters [20]. Peters concluded that Gong's model showed little accuracy in conditions close to stall and choke. To remedy the limitations of Gong's model, Peters redesigned a new model in which the loss model depends on the local relative Mach number. His proposed model is:

$$f_p = \frac{K_{p1}}{h}[M_{rel}^2 + K_{p2}(M_{rel} - M_{ref})^2]W^2 \tag{2.8}$$

where M_{ref} is the average relative Mach number at the blade row at peak efficiency. K_{p1} and K_{p2} are calibration coefficients coming from the bladed RANS simulations at the peak efficiency condition. However, these coefficients need an iterative procedure in which the results of body force and the results of bladed RANS calculations are set to be compared to achieve a best fit. Peters et al. [44] used the calibrated approach to successfully capture the interaction of a ultra high bypass ratio (UHBR) fan with a short inlet nacelle, but this approach required detailed data of the fan for bladed RANS simulations which is generally unavailable to airframers in the parametric fan/nacelle design process.

Hall et al. [14] developed an analytical normal force model that requires no calibration. He used it to investigate fan stage design attributes for boundary-layer-ingesting turbofan engines. While only fan camber surface geometry data is needed, the model is lossless. This model uses the local deviation (d) angle using a linear lift coefficient for an isolated blade. That is to say, the lift coefficient in an isolated blade using the linear flow theory is correlated with local deviation as $C_l = 2\pi d$. Therefore, the local lift force appears to be:

$$F_L = C_l(0.5\rho W^2 S) \tag{2.9}$$

Accounting for finite-pitch effects, the volumetric normal force becomes:

$$f_n = C_l \frac{0.5\rho W^2}{|n_\theta| h} = \frac{(2\pi d)(0.5\rho W^2)}{|n_\theta|(\frac{2\pi r}{B})} \tag{2.10}$$

where n_θ is the circumferential component of the unit vector normal to the camber surface and B is the number of blades in the rotor/stator row. The normal force is exerted normal to the relative flow streamline. A schematic of a blade camber surface, the normal vector to the camber surface, the flow relative velocity, and normal/parallel force vectors are shown in Fig. 2-1.

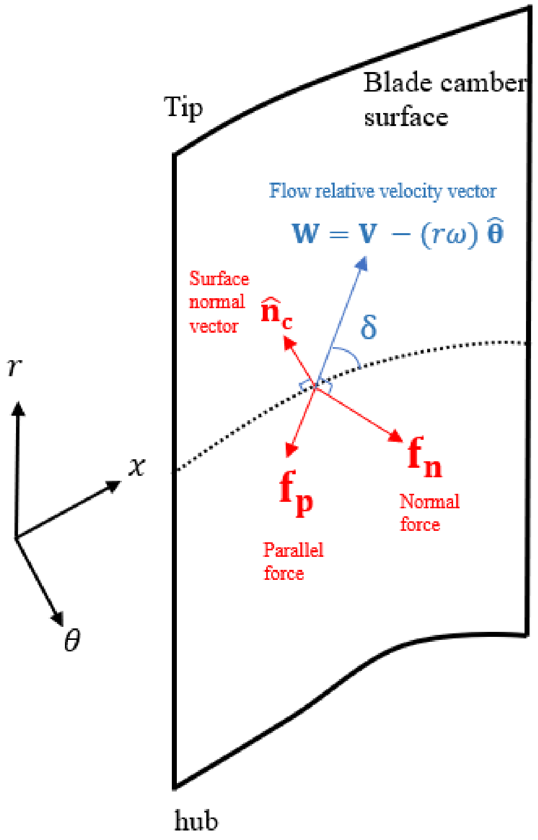


Figure 2-1: Geometric description of local blade camber surface normal unit vector $\hat{\boldsymbol{n}}_c$, relative velocity $\boldsymbol{W}$, and momentum normal source $\boldsymbol{f}$)

Hall's model (Eq. 2.10), has shown appropriate responses and acceptable predictions on non-uniform fan inflows under different non-uniform conditions. However, this model has no parameter that captures flow compressibility. Defoe et al. [45] showed that by scaling the response to weaker distortions, it is possible to make good predictions for more substantial distortions. They used an inviscid model, so the non-uniform effects of the flow on fan efficiency were not examined.

Continuing non-uniform flow studies on the fan/compressor, Hill and Defoe [21] developed the Hall analytical model by adding a compressibility parameter to capture the transonic shock wave impacts on flow turning. Then, they combined it with the Peters' calibration-based loss model. However, they concluded that Peters' method is not sufficient to model the efficiency of a transonic compressor in choke conditions, and therefore introduced an improved loss model. These innovated methods for both the normal/turning and loss models show an improvement in the compressor input work calculations' quality in the span-wise direction. However, both the normal and the loss force predictions rely on calibration.

Thollet et al. [46] studied the blockage factor parameter in the body force model with Gong's calibrated normal force and Peters' calibrated loss force and reviewed those models with the blockage factor effect in the simulations. They concluded that the addition of specific source terms to capture metal blockage greatly enhances the accuracy of the flow prediction in transonic blade rows, both in terms of work coefficient and choking mass flow rate. In another attempt to study the fan engine performance under boundary layer ingestion (BLI) distortions, an interdisciplinary fan engine configuration was proposed by Vega et al. [47]. They used a calibrated aerodynamic body force model for 3D CFD calculations and integrated it with a zero-dimensional engine cycle program to assess the engine's fuel consumption. Even though the body force model is a reduced-order simulation that decreases the computational costs for fuel consumption multi-disciplinary platforms, that integrated model used Gong's calibrated viscous model that requires experimental data. The other simplified loss model has been used by Liu et al. to assess fan performance using the body force model [48]. They used a radial distribution of loss for case-specific simulations and implemented it in the body force model. That loss simulation may not be a general approach in body force models.

Dufour et al. [49] demonstrated that the body force approach is beneficial in predicting fan windmilling rotational speeds for high bypass ratio configurations. That model, however, requires in advance calibration coefficients from bladed CFD analysis to be used in the body force model with Gong's approach for normal and parallel forces.

In a different approach, Guo and Hu [50] have tried a loss model in the body force method using simple calculations of the blade loss coefficient and converting it to the entropy produced along the streamline. This method requires knowing the total temperature of the airfoil's trailing edge. However, the method is based on simple models.

Recently, some studies have been accomplished in body force modellings for capturing the stall/surge dynamics [51, 52]. The stall studies show that the empirical data are required for loss models. Thus, the simulations may be limited to a specified range of operating conditions.

New developments in parallel force model have been implemented for non-uniform inflows in fans by Godard et al. [15]. In this model, the local friction coefficient (c_f) is calculated using an empirical equation. The model is dependent on the local deviation angle (d) and the deviation (d_{ref}) at maximum efficiency from the body-force model. That model is:

$$f_p = \frac{\rho W^2}{|n_\theta|(\frac{2\pi r}{B})} \left(c_f + \pi \left(d - d_{ref}\right)^2\right) \tag{2.11}$$

The model requires no calibration with bladed computations, however the friction coefficient is a simple correlated model for turbulent flow over a flat plate. This model does not include the separation and displacement thickness losses in a detailed manner. The same equation has been used by Benichou et al. [23], as well. The results of [23] show that the loss prediction for a stator in non-uniform flow has a 37% of error in loss coefficient, meaning that an improvement in the loss body force is required.

Recently, Minaker and Defoe coupled an analytical, inviscid body force model with a turbofan primary design scheme to predict the crosswind separation speeds for fan and nacelle systems [53, 54]. In their research, geometry data are determined for the specified fan operating designs, and the body force simulations assess the crosswind separation speeds. However, the simulations do not include any loss models.

All body force approaches conceptually involve taking the local pressure difference across a blade and smear it out over a blade pitch to yield a body force per unit volume. Similarly, immersed boundary methods with smeared geometry (IBMSG) for fan modelling have been successfully applied by Cao et al. [55] and Ma et al. [56]. This modelling approach still requires some calibration of the forces to ensure that the correct deviation and loss are achieved.

In summary, it is recognized that the methods of loss prediction in body force models started with Gong's [12] calibration method and have been upgraded by Peters' model [20] over time. Recently, Thollet [22] offered a simple model without the need for calibration, but this model does not have the necessary quality in modelling non-uniform inflows [23]. Therefore, it is necessary to develop a model without the need for calibration, but in an improved way. In the next section, the theories of conventional loss and entropy production models in turbomachinery are reviewed.

2.4 Background Related to Loss Models

In the blade row in compressors/fans, the fluid boundary layer is strongly coupled to the non-viscous flow outside the boundary layer. Boundary layer flows are also associated with the dissipation of mechanical energy, which exhibits itself as a loss or inefficiency of the fluid motion. The flow in the boundary layer creates vorticities that conflict with the potential flow theory for velocity prediction, but it also changes the flow field due to the blockage it produces. The velocity changes around the airfoil stem from displacement effects that the boundary layers have on the flow. Denton [24] has discussed the different loss mechanisms in turbomachinery. He has illustrated that the boundary layer viscous shear work comes from integrating the shear work across the boundary layer. In conclusion, he introduced the entropy generation rate per unit surface of the boundary layer in the form of:

$$\dot{S} = \frac{\rho c_{\mathcal{D}} u_e^3}{T} \tag{2.12}$$

where $c_{\mathcal{D}}$ is the boundary layer local dissipation coefficient, u_e is boundary layer edge velocity, and T is the local static temperature. That method requires an accurate airfoil loading model, which delivers accurate local velocities, distributed along the chord, since it is dependent on the cube of velocity. In a simple way of correlating the dissipation coefficient based on the experimental data, Schlichting [57] has presented an equation:

$$c_{\mathcal{D}} = 0.0056 Re_\theta^{(-1/6)} \tag{2.13}$$

where Re_θ is local momentum thickness Reynolds number. This equation requires the local momentum thickness for which there is no available analytical equation without any need for iterative procedures.

Regarding the shock wave entropy generation, Denton has provided an equation for weak shocks in terms of the local relative Mach number as:

$$\Delta s \approx c_v \frac{2\gamma\left(\gamma-1\right)}{3\left(\gamma+1\right)^2}\left(M_{rel}^2-1\right)^3 \tag{2.14}$$

where c_v is specific heat at constant volume and γ is isentropic expansion factor (heat capacity ratio). Equation 2.14 constitutes the base for the shock loss model of the body force modelling in this dissertation as discussed in Chapter 4.

Youngren [26] has introduced a mass-average loss where the total pressure defect is calculated by the momentum thickness. In that model, assuming adiabatic walls and that the edge pressure is equal to the local pressure inside the boundary layer, the local total pressure defect is linearized to:

$$\delta p_t = \frac{p_{te}}{p_e} \rho_e u_e \delta u \tag{2.15}$$

where δu is velocity deviation from edge velocity ($\delta u = u_e - u$), p_{te} is the total pressure at the boundary layer edge and p_e is the static pressure at the boundary layer edge. A schematic of the boundary layer geometry for the loss calculations is shown in Fig. 2-2.

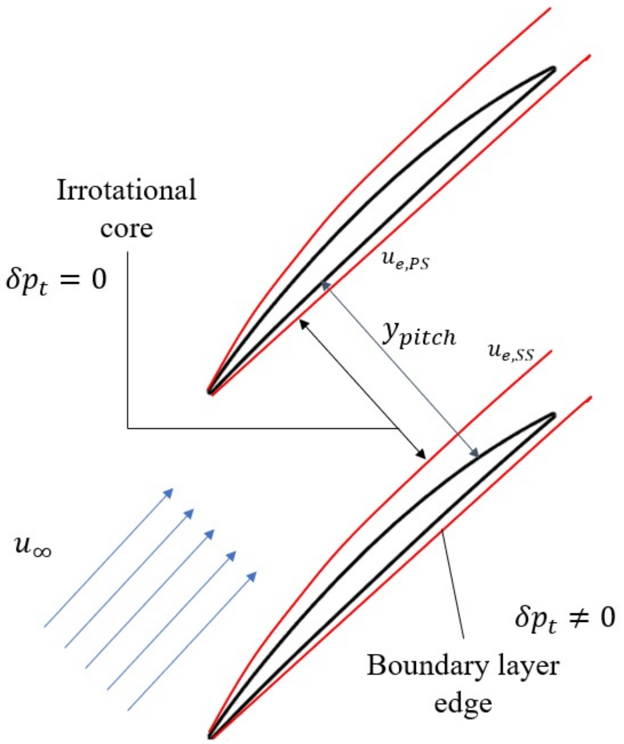


Figure 2-2: Schematic of geometry for boundary layer calculations

The mass-averaged total pressure defect across the boundary layer, used in MISES [25] is:

$$\Delta \bar{p}^M_{t,local} = \int_0^{y_{pitch}} \delta p_t \frac{d\dot{m}}{\dot{m}} = \frac{p_{te}}{p_e} \frac{\rho_e^2 u_e^3}{\dot{m}} b\theta \tag{2.16}$$

where b is the streamtube width, $\Delta \bar{p}^M_{t,local}$ is mass-averaged total pressure loss, θ is boundary layer momentum thickness, and $\dot{m}$ is the mass flow rate. Therefore, the total pressure drop of a cascade is the total pressure defect at the trailing edge:

$$\bar{\Delta p}_{t,loss}^{M} = p_{t,inlet} - p_{t,exit} = \left[\frac{p_{te}}{p_e}\frac{\rho_e^2 u_e^3}{\dot{m}} b\theta\right]_{exit} \tag{2.17}$$

For fan/compressor cases, Eq. 2.17 can be considered as the profile loss (neglecting the shock loss). Since the details of the momentum thickness at the exit are adequate to find the total pressure loss, this model forms the base of the viscous loss body force model in this dissertation.

Further studies have been conducted by Singh [58] in loss simulation of 3D test cases which capture shock-boundary layer interactions. In that research, a 3D time-marching CFD-based computer program for a transonic compressor was introduced. The model uses an inviscid solver but uses integral boundary layer equations on the blade surface to calculate the displacement thickness, and then the displacement thickness updates the 3D compressor geometry. The boundary layer equations are two-dimensional, and a simple treatment of shock-boundary-layer interaction is included. It was shown that even the 2D boundary layer approach on 3D geometry calculations is encouraging.

2.5 Boundary Layer Equations

The theory of boundary layers and the governing equations and some solutions have been presented by Schlichting in [57]. Assessing the boundary layer's physics and its governing equations is needed to achieve a reliable viscous entropy generation prediction. The accurate viscous entropy generation may provide a robust viscous parallel force in a body force approach. For the sake of precise prediction of viscous entropy generation, it is necessary to model the coupled physics of the boundary layer and the flow of the effectively inviscid region outside the boundary layer. To get the momentum defect in the boundary layer, the integral momentum equation needs to be solved:

$$\frac{d\theta}{d\xi} + (2 + H - M_e^2)\frac{\theta}{u_e}\frac{du_e}{d\xi} = \frac{c_f}{2} \tag{2.18}$$

where H is the boundary layer shape parameter and ξ is streamwise direction coordinate. One of the promising correlations for the skin friction coefficient has been introduced by Drela and Giles [59] for both the laminar and turbulent regimes. Their work suggests a fully-simultaneous mathematical approach for solving of the boundary layer equations coupled with an inviscid Euler equation solver. In addition to being dependent on the boundary layer edge Mach number and skin friction coefficient, the momentum thickness is reliant on the shape parameter as well, which is defined as:

$$H = \frac{\delta^*}{\theta} \tag{2.19}$$

where δ^* is the displacement thickness, defined as:

$$\delta^* = \int_0^{\delta} (1 - \frac{\rho u}{\rho_e u_e}) d\eta \tag{2.20}$$

where δ is boundary layer thickness and η is a dummy variable of integration across the boundary layer thickness. The displacement thickness accounts for the boundary layer blockage effects. In viscous-inviscid solvers (described in the next Section), the displacement thickness is added to the blade thickness to account for the effective blade reshaping by the boundary layer's presence.

At high incidence angles or for highly cambered airfoils, the velocity around the airfoil has high gradients, leading to high shape parameter values and consequently, flow separations in the boundary layers. Thus, a supplementary equation is required to capture the shape factor values.

One of the well-known equations for capturing shape factor parameter is Head's model [60]. This model is suitable for fully-turbulent boundary layers. In Head's model, the turbulent region of flow is assumed to be bounded by a flow that is non-turbulent and substantially irrotational. Interaction between the two flow regimes is generally introduced as entrainment (of the non-turbulent by the turbulent flow). In Head's model, a differential equation for the entrainment has been proposed that provides the shape parameter. However, the model is appropriate only for fully-turbulent boundary layers. The other approach is to use the kinetic energy integral boundary layer equation [59]. In this approach, the kinetic energy shape parameter (H^*) is calculated by an ordinary differential equation. Drela and Giles have focused on this equation in [59]:

$$\theta\frac{dH^*}{d\xi} + (2H^{**} + H^*(1-H))\frac{\theta}{u_e}\frac{du_e}{d\xi} = 2c_{\mathcal{D}} - H^*\frac{c_f}{2} \tag{2.21}$$

where H^* is boundary layer kinetic energy shape parameter and H^** is density shape parameter. An empirical equation accounting for the dissipation coefficient has been introduced for the laminar regime in [59]. However, for the turbulent regime, due to the presence of two layers in turbulent region in the boundary layer (wall layer and wake layer), the dissipation coefficient is expressed as a sum of two contributions [59]:

$$c_{\mathcal{D}} = \frac{c_f}{2}U_s + C_\tau(1-U_s) \tag{2.22}$$

where C_τ is shear coefficient which is a measure of stress in wake layer and U_s is an equivalent normalized wall slip velocity which is defined as:

$$U_s = \frac{H^*}{2}\left(1 - \frac{4}{3}\frac{H_k - 1}{H}\right) \tag{2.23}$$

where H_k is kinematic shape parameter which is defined as:

$$H_k = \frac{\int_0^\delta (1-\frac{u}{u_e})d\eta}{\int_0^\delta \frac{u}{u_e}(1-\frac{u}{u_e})d\eta} \tag{2.24}$$

In Drela's model, one extra ordinary differential equation is coupled with the other equations. A shear stress lag equation is [59]:

$$\frac{\delta}{C_\tau}\frac{dC_\tau}{d\xi} = 4.2(C_{\tau_{EQ}}^{0.5} - C_\tau^{0.5}) \tag{2.25}$$

where $C_{\tau_{EQ}}$ is the equilibrium shear stress coefficient which is expressed by an empirical equations as:

$$C_{\tau_{EQ}} = H^*\frac{0.015}{1-U_s}\frac{(H_k-1)^3}{H_k^2 H} \tag{2.26}$$

The last ordinary differential equation remaining to get the details of the boundary layer is to predict the position of transition from laminar to turbulent flow. Drela has included the spatial amplification rate with empirical relation:

$$\frac{d\hat{n}}{d\xi} = \frac{d\hat{n}}{dRe_\theta}\frac{1}{2}\left(\frac{\xi}{u_e}\frac{du_e}{d\xi} + 1\right)\frac{\rho_e u_e \theta^2}{u_e \xi}\frac{1}{\theta} \tag{2.27}$$

If separation of the flow does not occur within the boundary layer, solving the four equations 2.18, 2.21, 2.25 and 2.27 is sufficient to evaluate the momentum thickness.

Pazireh and Defoe [61] presented a body force model in which the boundary layer equations are introduced through transport equations, convecting momentum thickness, shape factor parameter, shear stress coefficient and the amplification ratio along the relative frame streamlines as the fictitious viscous characteristics for parallel force calculations in the body force model. However, when separation occurs, Eq. 2.21 is not solvable, and thus, Goldsten's singularity problem appears. This singularity means that shape parameter approaches infinity and the solution cannot be obtained. The issue originates from the high degree of coupling between the local boundary layer edge velocity and the displacement thickness so that the velocity changes in a way to prevent the displacement thickness from progressing to infinity. To avoid the singularity, iterative procedures have been proposed. These procedures start with an initial guess for local displacement thickness and velocity distributions and then are updated by a range of methods. The following section describes a brief discussion of the boundary layer iterative methods.

2.6 Iterative procedures for two-way coupling between viscous-inviscid interactions

In two-way viscous-inviscid coupling procedures, separate calculations of outer inviscid flows and the inner viscous flows are combined to produce a composite solution of the overall flow. These techniques require a non-linear inviscid flow solver and viscous differential equations with the boundary conditions on the matching edges with displacement thickness and boundary layer edge velocity as the interactive variables of the two solvers. Iterative procedures accomplish simultaneous solutions. Figure 2-3 is a schematic of four conventional iterative methods of inviscid-viscous solvers in turbomachinery applications.

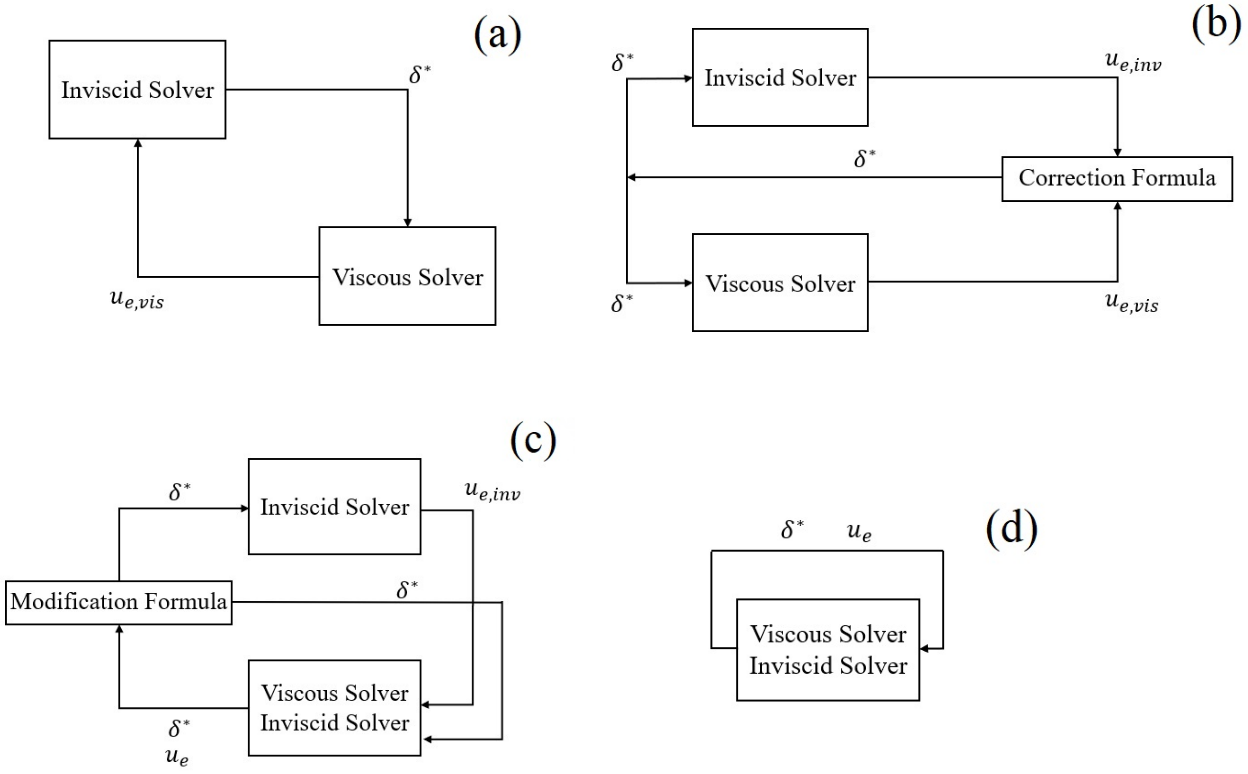


Figure 2-3: Iterative techniques for viscous-inviscid interaction calculations - (a) fully inverse method, (b) semi-inverse method, (c) semi-simultaneous method and (d) fully simultaneous method

In direct solution (one-way coupling), the velocity comes from an inviscid solver and is implemented into the viscous solver. Then, the displacement thickness as an output of viscous solver is implemented into the inviscid solver to update the geometry thickness and the procedure iterates until the difference in velocity or displacement with previous iterations matches the error criterion. This strategy works for the attached boundary layers. However, in the presence of separation, the calculation features should be modified. As shown in Fig. 2-3(a), the fully-inverse approach implies that the initial displacement is given to the viscous solver, and the velocity is determined using Eqs. 2.18 and 2.21. Then that velocity is returned to the inviscid solver, and with Newton-Raphson or any non-linear equations solver, the corresponding displacement is calculated for discrete points on the matching edge. This procedure continues until the error criterion is met. The low speed of this method is its disadvantage. The semi-inverse method (shown in Fig. 2-3(b)) was developed to speed-up the inverse technique convergence rate. In the semi-inverse approach, the input of both solvers is the displacement thickness. This reduces the complexity in the inviscid solver. The displacement thickness is then updated using the difference of velocity prediction in each iteration. This strategy still has divergence problems in some cases as it is sensitive to the relaxation factor. The semi-simultaneous approach (shown in Fig. 2-3(c)) uses the simultaneous attributes of the calculations, which helps reduce sensitivity to the relaxation factor, and adds a modification formula for avoiding the singularity problem. The reader is referred to refs. [62–64] for more mathematical procedure details of these methods.

2.7 Airfoil Loading Models

Hall's loading normal force model employs only the camber surface data and forces the flow to adjust to the camberline direction with a force dependent on the local deviation [14]. However, in reality blade loading depends on the camber and thickness distributions of the airfoil section along the streamline. The calculation of the velocity/pressure distributions around an airfoil in inviscid flow using an analytical model is one of the noteworthy problems in fan/compressor body force modelling approaches. Doing this accurately requires airfoil camber and thickness distributions. Potential flow theory is a promising approach that provides the induced velocities around an airfoil. Although there are several approaches to create airfoil loading in potential flow theory, distributed singularities along the chord of the airfoil is one of the favoured methods. Thickness effects are captured using source singularities distributed along the chord. The effects of the camber and any non-zero angle of attack, which produce the load on the airfoil, are captured with vortex distributions.

An accurate velocity distribution model can lead to a new normal body force model which can track the actual local load in body force simulations. Previous studies have been carried out on isolated blades by Powell [65]. He introduced an analytical model for calculating of velocity distribution around an isolated blade. Later, the potential flow around the airfoils has been extended for cascade by taking into account the finite-pitch effects by Baddoo and Ayton [66]. This section introduces the loading models for staggered cascades that can form a basis for body force models in future studies, especially for situations where the local viscosity effects in the calculations require knowledge of the edge velocities on both sides of the blade.

2.7.1 Isolated Airfoil Potential Flow

We begin with an isolated blade geometry definition. A schematic of the geometry nomenclature is illustrated in Fig. 2-4. Consider an isolated thin airfoil under the assumption of small disturbances in a two-dimensional, steady, incompressible flow. W_∞ is the free stream relative to blade velocity, η_{th} is the distance between camberline and blade surface, η_c is the distance between chordline and camberline, c is chord length, α^* is angle of attack.

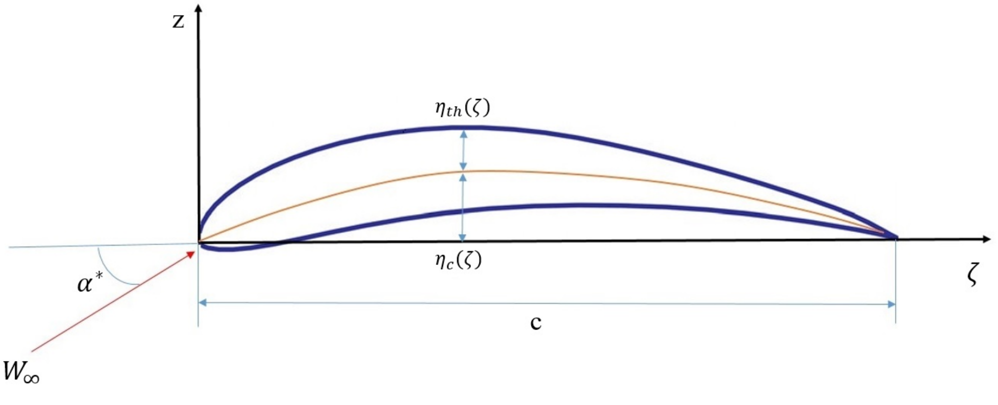


Figure 2-4: Airfoil geometry details for velocity distribution calculations in potential flow theory

We treat all lengths as being non-dimensionalized so that the semi-chord of the airfoil is 1. In any plane normal to the leading edge of the airfoil a system of rectangular co-ordinates ζ, z is used where the ζ-axis is along the chord with $\zeta = -1$ at the leading edge. Powell [65] has presented a model to calculate the velocity distribution around an isolated airfoil as:

$$W_{SS,PS}(\zeta) = \frac{\cos\alpha^{*}\big(1 + \frac{\mathcal{A}^{(1)}(\zeta)}{B_2} \pm \frac{\mathcal{A}^{(4)}(\zeta)}{B_1}\big) \pm \frac{\sin\alpha^{*}}{B_2}\left(1 + \frac{\mathcal{A}^{(3)}(\zeta)}{B_1}\right)\sqrt{\frac{1-\zeta}{\zeta}}}{\sqrt{1 + \big(\frac{\mathcal{A}^{(2)}(\zeta) \pm \mathcal{A}^{(5)}(\zeta)}{B_2}\big)^2}} W_{\infty} \tag{2.28}$$

where the subscripts SS and PS refer to suction side and pressure side, respectively. B_1 and B_2 are compressibility factors applied to the contributions to the velocity distribution due to camber and thickness, respectively. $\mathcal{A}^1$ to $\mathcal{A}^5$ are shape parameters. The shape parameters are defined as:

$$\mathcal{A}^{(1)}(\zeta) = \frac{1}{\pi}\int_{-1}^{1} \frac{d\eta_{th}}{dx_0}\frac{1}{\zeta - x_0}dx_0 \tag{2.29}$$

$$\mathcal{A}^{(2)}(\zeta) = \frac{d\eta_{th}}{d\zeta} \tag{2.30}$$

$$\mathcal{A}^{(3)}(\zeta) = \frac{1}{\pi}\int_{-1}^{1} \left[\frac{d\eta_{th}}{dx_0} - \frac{\eta_{th}}{2x_0\left(1 - x_0\right)}\right]\frac{1}{\zeta - x_0}dx_0 \tag{2.31}$$

$$\mathcal{A}^{(4)}(\zeta) = \frac{1}{\pi}\sqrt{\frac{1+\zeta}{1-\zeta}}\int_{-1}^{1} \frac{d\eta_c}{dx_0}\sqrt{\frac{1+x_0}{1-x_0}}\frac{1}{\zeta - x_0}dx_0 \tag{2.32}$$

$$\mathcal{A}^{(5)}(\zeta) = \frac{d\eta_c}{d\zeta} \tag{2.33}$$

The precise form of compressibility factors are presented in [65] as:

$$B_1 = \sqrt{1 - M^2} \tag{2.34}$$

$$B_2 = \sqrt{1 - M^2\big(1 - C_{p_i}\big)} \tag{2.35}$$

where:

$$C_{p_i} = 1 - \frac{\left(1 + \mathcal{A}^{(1)}\right)^2}{1 + \left(\mathcal{A}^{(2)}\right)^2} \tag{2.36}$$

The flow around an airfoil with a boundary layer can be represented by the inviscid flow about a suitable chosen displacement surface. This is a valid assumption for subsonic flow. Therefore, the surface thickness can be updated as:

$$\eta_{th,vis} = \eta_{th} + \delta^* \tag{2.37}$$

This approach is capable of capturing the velocity distribution on either side of the blade. A sample of the chordwise velocity distribution using Powell's model around the airfoil NACA 6412 is shown in Fig. 2-5. The maximum error on the suction side is 5% and on the pressure side is 3%.

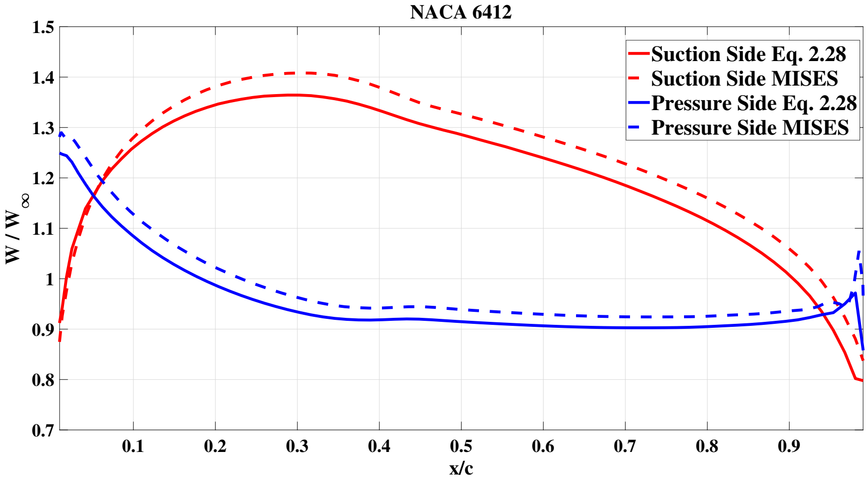


Figure 2-5: Normalized velocity distribution on either side of NACA6412 at $M_\infty = 0.3$

2.7.2 Cascade Airfoil Potential Flow

Baddoo and Ayton [66] have presented a model that employs potential flow theory for an infinite cascade. In that model, the perturbation size of the singularities on the airfoil is assumed to be comparable to the aspect ratio of the airfoils. That model uses Riemann–Hilbert's theory to convert the potential flow from a single airfoil to repeated airfoils, accounting for the cascade pitch in the calculations. The model calculates the airfoil surface tangent velocities:

$$\begin{aligned} W_{SS,PS}(\zeta) = \pm\alpha^{*} e^{-\pi/s} & \sqrt{\frac{\sinh\left(\pi\left(1-\zeta\right)/s\right)}{\sinh\left(\pi\left(1+\zeta\right)/s\right)}} - \\ & \frac{1}{s}\int_{-1}^{1} \frac{d\eta_{th}}{dx_0} \coth\left(\frac{\pi\left(x_0-\zeta\right)}{s}\right) dx_0 \\ & \mp \frac{1}{s}\sqrt{\frac{\sinh\left(\pi\left(1-\zeta\right)/s\right)}{\sinh\left(\pi\left(1+\zeta\right)/s\right)}} \times \\ & \int_{-1}^{1} \frac{d\eta_c}{dx_0} \sqrt{\frac{\sinh\left(\pi\left(1+x_0\right)/s\right)}{\sinh\left(\pi\left(1-x_0\right)/s\right)}} \left[\coth\left(\frac{\pi\left(x_0-\zeta\right)}{s}\right) - 1\right] dx_0 \end{aligned} \qquad (2.38)$$

where s is the non-dimensional pitch spacing that is $s = \frac{2h}{c}$. Since the normalized chord is 2, the normalized pitch is 2 times the actual pitch to chord ratio.

Powell's airfoil loading model [65] shown in Eq. 2.28 uses shape parameters which account for the source and vortex distributions on only isolated airfoils. In addition, Baddoo and Ayton's model [66] is limited to low-camber airfoils as the model assumed that source and vortex distributions on the chord are equivalent to placing them on the camberline. Furthermore, both models take leading-edge velocities to add to the local perturbations which account for thickness and camber effects. Moving the vorticity and source distributions from the chordline to the camberline in Baddoo and Ayton's method needs to be completed in order to obtain a model for velocity distributions in highly cambered cascades. In Chapter 3, such a modified model for cascade airfoil velocity distributions is introduced.

2.8 Artificial Neural Networks in Turbomachinery

An artificial neural network mimics the structure of the human neural network where there is a dense network comprising thousands to millions of neurons throughout the human brain. Neurons receive signals from an input system and pass the information to a main-frame in which the information is transferred to the output. A neuron is a switch with information input and output [67]. An artificial neural network consists of a few to tens of neurons in each layer and the neurons are connected to the neighbour layers with weighting coefficients. Figure. 2-6 shows an architecture of a neural network sample that includes five neurons in the hidden layer with four inputs and one output. The mathematical procedure is explained later. Mostly when referring to an n layer network it implies that the number of hidden layers is n. In this dissertation, a one-layer network refers to a network with a hidden layer.

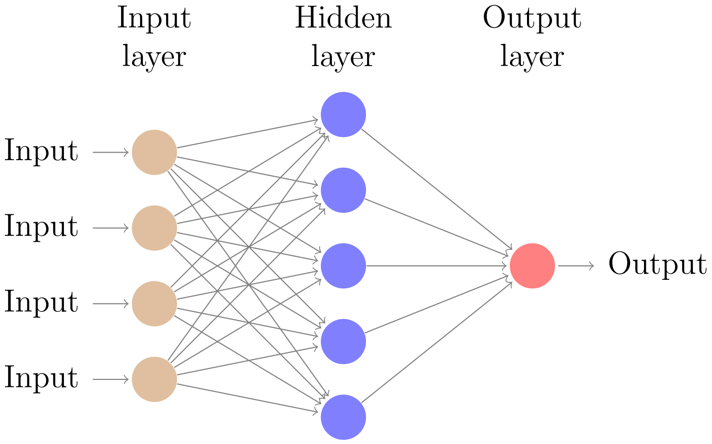


Figure 2-6: An architecture of a one hidden layer neural network

A neural network in a turbomachinery application is an example of supervised learning. Supervised learning implies that to train the network, data samples comprising input and output data are needed. More clearly speaking, supervised learning looks for an algorithm that relates the input to the output data based on the training dataset [68]. On the contrary, unsupervised learning refers to methods where there is no output label and the learning process is followed by the correlations between samples. The reader is referred to the work of Zhao and Liu [68] for more detailed information on the algorithms for supervised and unsupervised learning. Neural networks can be applied to both classification and regression problems. Dreiseitl and Ohno-Machado [69] have discussed artificial neural networks for both classification and regression and compared the performance with other machine learning methods in a review paper. Since the subject in this dissertation concentrates on turbomachinery physics where the variables are continuous (not discrete), we focus on the regression type of neural network.

From a data perspective, inputs and output are attributes of a sample in the dataset. An example in turbomachinery CFD can be that any local cell is a sample which has flow properties and turbulence information. Therefore the attributes for the cell can be fluid Mach number, Reynolds number, distance from the wall, turbulence kinetic energy, etc. Thus, if the turbulence kinetic energy is dependent on the three other attributes, then there would be three input variables and one output. The total number of cells times the number of CFD simulations with different conditions form the number of samples. Another example can be the global turbomachinery parameters considered for artificial neural network. For a specific compressor, mass flow rate and rotational speed provide the pressure ratio and isentropic efficiency information. Thus, in this example, there are four attributes: the mass flow rate and rotational speed are inputs and the isentropic efficiency and pressure ratio are outputs. Data from performance characteristics maps for different conditions would be data samples. After explaining the mathematical structure of a feed-forward back-propagation network in the coming paragraphs, a few state-of-the-art papers which apply ANNs to turbomachinery are described.

A neural network aims to provide an analytical equation that estimates the output based on any given input. There are multiple neural network architectures, but here we describe the feed-forward back-propagation approach, which is most often used for turbomachinery applications.

The calculations in the feed-forward method start with normalizing the input data. We assume a one-layer network has m inputs and n neurons in the hidden layer ending up with o outputs. $\boldsymbol{X}$ represents the input vector. $\boldsymbol{X_{min}}$ is considered to be a vector that includes the minimum of each input attribute on the whole dataset. $\boldsymbol{X_{max}}$ is a vector including the maximum of each input attribute on the whole dataset. The normalized input vector $\boldsymbol{F_1}$ is calculated as:

$$F_{1,i} = 2\frac{(X_i - X_{min,i})}{X_{max,i} - X_{min,i}} - 1 \qquad i = 1, ..., m \tag{2.39}$$

So that $-1 < F_{1,i} < 1$. The matrix w_1 is considered to be a weighting matrix between the normalized input vector and hidden layer neurons and thus has size $n \times m$. Thus, the arrays on any ith row of the matrix w_1 represent the contribution of the inputs on the ith neuron. The matrix w_2 is the weighting matrix between the hidden layer and output with a dimension of $1 \times n$. A bias vector $\boldsymbol{b_1}$ with the dimension of $n \times 1$ is used in the hidden layer. The non-normalized input to the hidden layer is thus the vector $w_1\boldsymbol{X} + \boldsymbol{b_1}$. This vector is then normalized to an interval of [0,1]. This is done by a transfer function. A well-known transfer (activation) function is a hyperbolic tangent sigmoid function, which is defined as:

$$tansig(a) = \frac{2}{1+e^{-2a}} - 1 \tag{2.40}$$

where a is any input variable. The variables assigned to every single neuron in the hidden layer should be normalized. Therefore, the vector $\boldsymbol{F_2}$ is the variables on the hidden layer (normalized of $w_1\boldsymbol{X} + \boldsymbol{b_1}$ with the hyperbolic sigmoid function) and is calculated as:

$$F_{2,j} = tansig\left(\sum_{i=1}^{m} w_{1,j,i} \times F_{1,i} + b_{1,j}\right) \qquad j = 1, ..., n \tag{2.41}$$

Similarly, the information from the hidden layer to the output layer is transferred using implementing of hyperbolic tangent sigmoid function on another bias vector $\boldsymbol{b_2}$ summed with the matrix multiplication of w_2 and $\boldsymbol{F_2}$ as:

$$F_{3,k} = tansig\left(\sum_{j=1}^{n} w_{2,k,j} \times F_{2,j} + b_{2,k}\right) \qquad k = 1, ..., o \tag{2.42}$$

F_3 is a normalized vector, which should be converted back to the original output value. With considering y_{max} as the maximum output in the whole dataset and y_{min} as the minimum output in the dataset, the predictive output F_4 is calculated as:

$$F_{4,k} = \frac{(F_{3,k}+1)}{2}(y_{max} - y_{min}) + y_{min} \qquad k = 1, ..., o \tag{2.43}$$

In the feed-forward procedure, the weighting matrices and bias vectors are initially randomly generated. All samples in the dataset are given to the network. Based on the inputs, outputs are predicted through Eqs. 2.41 to 2.43. The root mean square error is used for all $\mathcal{N}$ samples in the dataset as:

$$RMS = \sqrt{\frac{\sum_{d=1}^{\mathcal{N}}(y_{actual,d} - F_{4,d})^2}{\mathcal{N}}} \tag{2.44}$$

where y_{actual} is the actual output in dataset.

In the back-propagation algorithm, gradient optimization is used to minimize the RMS error. So, the derivative of the RMS error with respect to the weight matrices and bias vectors are calculated, and the matrices and vectors are updated in each iteration until the gradients and RMS error are minimized. The detailed mathematical steps for the derivative calculations have been discussed by Svozila et al. [70].

Advanced studies have been conducted by researchers in the field of ANN applications in turbomachinery in recent years. A creative idea in the development of turbulent closure terms using an ANN was proposed by Tracey et al. in 2015 [71]. They produced a large data using the Spalart-Allmaras turbulent model for different conditions and offered a machine-learned turbulent model that predicts the skin friction coefficients for the 2D plates and 3D wings. The turbulent kinematic viscosity and three model source terms were given to the ANN to generate a new advanced turbulence model. Any cell at any condition is taken as a sample (observation). However, in that work, the selection of predominant attributes has remained a challenge.

Weatheritt et al. in 2017 [72] used an ANN to come up with an algebraic equation for the Reynolds stress closure term that shows an improvement compared to the traditional RANS simulations. That model used the anisotropy tensor, scalar invariants for the anisotropy tensor, and the basis for the anisotropy tensor as the attributes. However, their model showed poor results in high vorticity regions. That year, Weatheritt et al. [73] applied the previous anisotropy tensor in a modified closure equation to a turbine wake case. Stress tensor predictions showed an improvement, however the model is not capable of accurately predicting the far-wake region.

Recently, an ANN has been used in fan body force modelling by Luis et al. in 2018 [74]. In that work, instead of the common body force equations, the body force values on each local position in the blade swept volume was replaced by the blade forces extracted from the bladed RANS simulations. Those extracted blade forces were given to the ANN. Forces were calculated based on the cell position using the algebraic equations that were generated by ANN. That model has a lack of accuracy in the prediction of the body forces in the tip region.

2.9 Concluding Remarks

Studies of the boundary layer loss literature show that the velocity distribution around the blade and the boundary layer properties are highly intertwined. Both Powell [65] and Baddoo and Ayton's models [66] need full blade geometry to determine the loading and this confines the methods to 2D cascades. On the other hand, even if there is a known velocity distribution around the blade, it will not be possible to calculate the boundary layer's properties under the conditions of separation of the flow within the boundary layer in body force models without significantly increasing the calculation cost. The equations for the velocity distribution calculations around the blade have other limitations that mean they are not currently feasible to implementation for 3D compressors. These equations are rooted in the potential flow theory method, which will lead to poor accuracy in non-uniform inflows. Also, calculations with this method require the geometry of the airfoil surface on different sections of the blade from the leading edge to the trailing edge, making it difficult to employ for three-dimensional flows.

Youngren's Eq. (2.17) shows that just by having the momentum thickness, the boundary layer loss can be predicted. This equation, along with some simplifying assumptions, which are mentioned in Chapter 4, forms a new model for the loss that is discussed in this dissertation. The new model does not need any edge velocity distribution estimates. Finally, after making sure that the loss depends only on trailing edge momentum thickness, artificial neural network is used using the neural network method to provide an analytical model to calculate the momentum thickness at low computational cost. The fourth chapter deals with this issue in detail.

Chapter 3

A New Body Force Model for 2D Cascades

This chapter introduces a no-calibration body force loss model based on Denton's viscous entropy generation model [24]. In addition, a potential flow based turning model which is a hybrid of Powell's isolated blade theory [65] and Baddoo and Ayton's finite pitch model [66] in a 2D cascade is introduced to be used to yield the required local boundary layer edge velocities in the loss model. The results of the blade loading model are assessed. In addition, the results of a loss model with constant dissipation coefficient in body force calculations are assessed.

3.1 Denton's Loss Model For No-Calibration Body Force

To compute a viscous body force, it is required that the properties of boundary layers on either side of the blade be determined. Denton [24] declared that the rate of entropy increase along a relative streamline is related to the viscous force per unit mass acting on the fluid in the direction of the streamline. Thus, the viscous body force (with units $\frac{N}{m^3}$) is related to the entropy generation by:

$$f_p = \rho T \frac{ds}{d\xi} \tag{3.1}$$

The entropy flux in the blade boundary layer along the streamline is calculated as:

$$\dot{S} = \rho W h |n_\theta| \frac{ds}{d\xi} \tag{3.2}$$

Utilizing Denton's relationship (Eq. 2.12) for the local entropy generation in a boundary layer [24] and combining with Eq. 3.2 yields:

$$\rho T \frac{ds}{d\xi} = \frac{c_{\mathcal{D}} \rho W_e^3}{W h |n_\theta|} \tag{3.3}$$

Combination of Eqs. 3.1 and 3.3 gives the loss body force as:

$$f_p = \frac{c_{\mathcal{D}} \rho W_e^3}{W h |n_\theta|} \tag{3.4}$$

The local loss body force is the summation of the suction and pressure side volumetric forces. The overall viscous loss force per unit volume is:

$$f_p = \frac{\rho}{W h |n_\theta|} \left(c_{\mathcal{D},SS} W_{SS}^3 + c_{\mathcal{D},PS} W_{PS}^3 \right) \tag{3.5}$$

It can thus be seen that the edge velocities from the loading model are required for the viscous losses to be determined. Section 3.2 introduces a new analytical flow turning model that captures the local velocities on either side of the blade in a 2D cascade.

3.2 A New Flow Turning Model In 2D Cascades

The author of this dissertation combined the two models of Powell [65] and Baddoo and Ayton [66] and developed a new analytical model which uses the local relative velocity in body force calculations. The integrals for the shape parameters are updated for finite-pitch cascade effects and the inlet velocity is replaced by the flow local relative velocity in a body force implementation. The local relative velocity represents the local potential flow with no perturbation to which the camber and thickness perturbations are added.

Baddoo and Ayton [66] introduced a modified Plemelj formula explaining that in a periodic singularity distribution, for any singular distribution of a function $f(t)$ the following relation is satisfied:

$$\frac{1}{\pi}\int_{-1}^{1}\frac{f(t_0)}{t_0-\zeta}dt_0=\frac{1}{s}\int_{-1}^{1}f(\tau)coth\left(\frac{\pi\left(\tau-\zeta\right)}{s}\right)d\tau \tag{3.6}$$

Thus, the shape parameter related to the thickness source distribution in Powell's model (Eq. 2.29) changes for a cascade calculation to:

$$\mathcal{S}^{(1)}(\zeta)=-\frac{1}{s}\int_{-1}^{1}\frac{d\eta_{th}}{dx_0}\coth\left(\frac{\pi\left(x_0-\zeta\right)}{s}\right)dx_0 \tag{3.7}$$

In addition, Baddoo and Ayton proposed that the vortex distribution for the periodic blades in a cascade may have a solution for the velocity perturbation of the form:

$$\mathcal{S}^{(3)}(\zeta)=e^{-\pi/s}\sqrt{\frac{\sinh\left(\pi\left(1-\zeta\right)/s\right)}{\sinh\left(\pi\left(1+\zeta\right)/s\right)}} \tag{3.8}$$

$$\begin{aligned}\mathcal{S}^{(4)}(x)=\mp\frac{1}{s}\sqrt{\frac{\sinh\left(\pi\left(1-\zeta\right)/s\right)}{\sinh\left(\pi\left(1+\zeta\right)/s\right)}}\times\\ \int_{-1}^{1}\frac{d\eta_c}{dx_0}\sqrt{\frac{\sinh\left(\pi\left(1+x_0\right)/s\right)}{\sinh\left(\pi\left(1-x_0\right)/s\right)}}\left[\coth\left(\frac{\pi\left(x_0-\zeta\right)}{s}\right)-1\right]dx_0\end{aligned} \tag{3.9}$$

The other two shape parameters are the same as in Powell's model:

$$\mathcal{S}^{(2)}(\zeta)=\frac{d\eta_{th}}{d\zeta} \tag{3.10}$$

$$\mathcal{S}^{(5)}(\zeta)=\frac{d\eta_c}{d\zeta} \tag{3.11}$$

Thus, an analytical model to calculate the velocity distribution around a finite pitch airfoil cascade in body force computations is:

$$W_{SS,PS}(\zeta) = \frac{\cos\alpha^*\big(1+\frac{\mathcal{S}^{(1)}(\zeta)}{B_2} \pm \frac{\mathcal{S}^{(4)}(\zeta)}{B_1}\big) \pm \frac{\sin\alpha^*\mathcal{S}^{(3)}(\zeta)}{B_1 B_2}}{\sqrt{1+\big(\frac{\mathcal{S}^{(2)}(\zeta)\pm\mathcal{S}^{(5)}(\zeta)}{B_2}\big)^2}} W \tag{3.12}$$

where W in the body force model is the local relative velocity.

Figure 3-1 shows the nomenclature used for the 2D cascade calculations. c is the chord length, h is the pitch, R_{LE} is the leading edge radius, t_{max} is the maximum thickness, $x_{t_{max}}$ is the position of maximum thickness, χ is the airfoil camber (difference between leading and trailing edge blade metal angles), $x_{c_{max}}$ is the position of maximum camber, σ is the chord to spacing ratio, φ_{TE} is the trailing edge boat-tail angle, i is the incidence angle, λ is the stagger angle, and κ_{LE} and κ_{TE} are the blade metal angles at the leading and trailing edges, respectively.

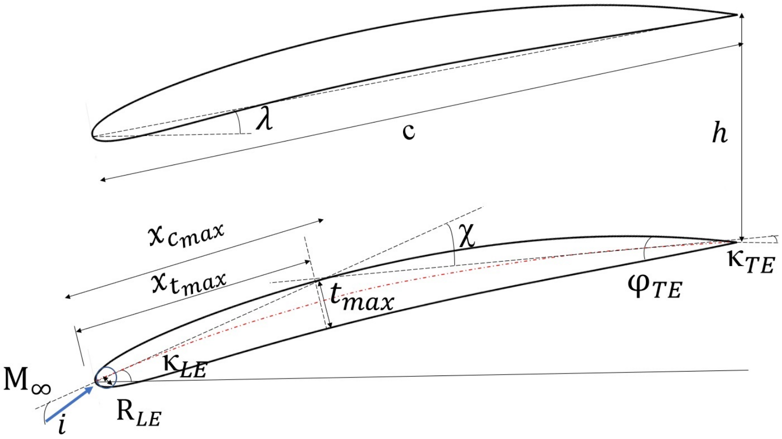


Figure 3-1: Schematic of geometry parameters for airfoil shape definition

Two compressor cascades were chosen to assess this new analytical blade loading model. Schematic views of these cascades are shown in Fig. 3-2.

The detailed data of the geometries are given in Table 3.1. The thickness and camber functions used to define the airfoil parameterizations are presented in a Python code in Appendix A. Cascade 1, with a blunt leading edge and 6% thickness, represents a compressor operating in nearly incompressible flow. However, cascade 2, with a sharper leading edge, represents a compressor operating in the high subsonic regime.

(a) Cascade 1

(b) Cascade 2

Figure 3-2: Studied cascades for the viscous model assessment

Table 3.1: Geometry data for cascades 1 and 2

parameter	cascade 1	cascade 2
$\frac{t_{max}}{c}$	0.06	0.05
$\frac{x_{t_{max}}}{c}$	0.4	0.5
χ(deg)	25	15
$\frac{x_{c_{max}}}{c}$	0.5	0.5
$\frac{R_{LE}}{c}$	0.005	0.001
φ_{TE} (deg)	10	10
λ (deg)	25	30

The loading is given by the pressure difference across the blade at a given location, $p_{PS} - p_{SS}$. This is smeared out over a blade pitch $h = 2\pi r/B$, where r is the local radius and B is the number of blades in a blade row, to yield a body force per unit volume f_n:

$$f_n = \frac{p_{PS} - p_{SS}}{h|n_\theta|} \tag{3.13}$$

Bernoulli's equation can be used to express the pressure difference in terms of the squared velocity difference:

$$p_{PS} - p_{SS} = 0.5\rho\left(W_{SS}^2 - W_{PS}^2\right) \tag{3.14}$$

It should be reiterated that suction side and pressure side velocities take into account the compressibility factors B_1 and B_2. The compressibility factors used in the velocities were presented in Eqs. 2.34 and 2.35.

Combining Eqs. 3.12, 3.13, and 3.14 leads to a new normal body force:

$$f_n = \frac{0.5\rho\,(W_{SS}^2 - W_{PS}^2)}{|n_\theta|(\frac{2\pi r}{B})} \tag{3.15}$$

MISES was used to carry out bladed CFD calculations to be the reference results for body force modelling comparisons. In MISES the inlet relative Mach number and relative flow angle are given to the solver. In the body force model a 2D Fluent solver [75] was used. User-defined-functions were used to implement the analytical model calculations in the source terms. The grid independence study for the 2D body force solutions is detailed in Chapter 4. The shape parameters used in Eqs. 3.7 to 3.11 are evaluated numerically. First order finite differences are used for derivatives and the rectangle rule is used for integrals. Appendix B provides the MATLAB code used for the shape parameters calculations.

The velocity distributions which result from the body force model for cascade 1 at an inlet Mach number of 0.3 for two incidence angles (0 and 6 degrees) are shown alongside MISES results in Figs. 3-3 and 3-4. At zero incidence angle, the maximum velocity error on the suction side is about 2% while the maximum velocity error for the pressure side is about 13%, which occurs at the trailing edge. One deficiency of the model is that it does not ensure zero loading at the trailing edge as required by the Kutta condition. However, very low loading after 90% chord does not affect the flow turning significantly. Good agreement for the pressure side velocity is shown at an incidence angle of 6 degrees. The suction side has a maximum error of 5%.

To assess the performance of the new loading model for producing flow turning, the flow turning angles have been calculated using both Hall's and the new analytical models. The flow angle differences from leading to trailing edge ($\beta_{LE} - \beta_{TE}$) for cascade 1 at an inlet Mach number of 0.3 and solidity of 1 are presented in Fig. 3-5. The new model shows a good consistency with MISES results. The maximum error for the new loading model appears at 5 degrees incidence angle and is 5%. Hall's model has an error of 7% at -6 degree incidence angle. Overall, performance of the new model is better than Hall's model.

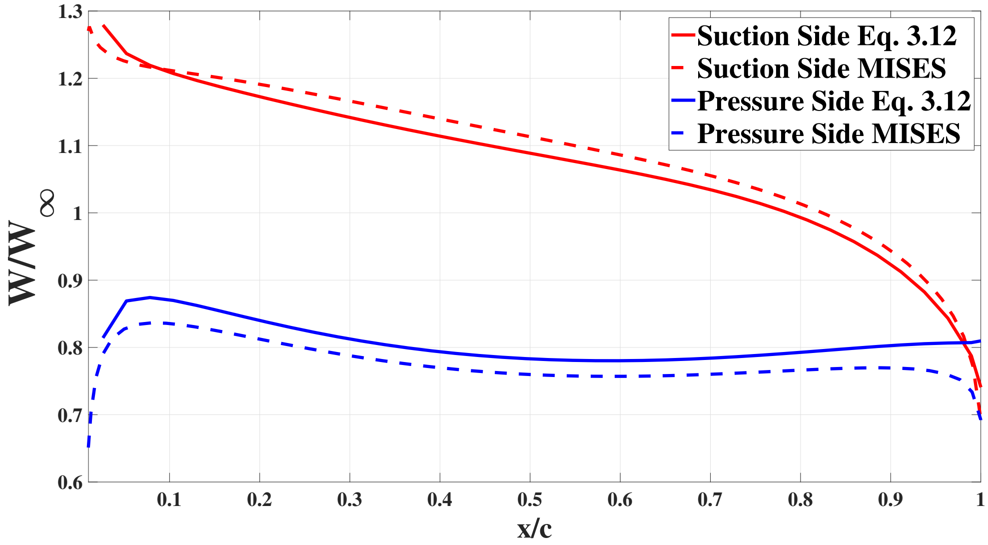


Figure 3-3: Comparison of analytical velocity distribution model with MISES CFD results for cascade 1 - $M_\infty = 0.3$, stagger angle = 25(deg), i = 0(deg), solidity = 1.0

Cascade 2 with the body force model has been simulated and the velocity distributions for a solidity of 1.2, inlet Mach number of 0.65, and incidence angle of 4 degrees are shown in Fig. 3-6. The inconsistency is clear in the first 10% chord, but otherwise agreement is very good. The higher loading prediction in the first 10% chord leads to over-prediction of flow turning. The flow turning for cascade 2 at an inlet Mach number of 0.65, and solidity of 1.2 is presented in Fig. 3-7. The flow turning body force by Hall's model uses a compressibility factor as described by Minaker and Defoe in [53]. The maximum error of flow turning with new body force model for the considered range of incidences, is at -6 degree where the error is 25%, which is a high discrepancy. However, this model has a better prediction of flow turning in positive incidence angles compared to Hall's model. Given that the model predicts the loading distribution accurately, it is a good way of obtaining W_{SS} and W_{PS} for a loss model.

In the next Section, the model is used with Denton's body force loss model, (Eq. 3.5) for which the local pressure side and suction side velocities are needed.

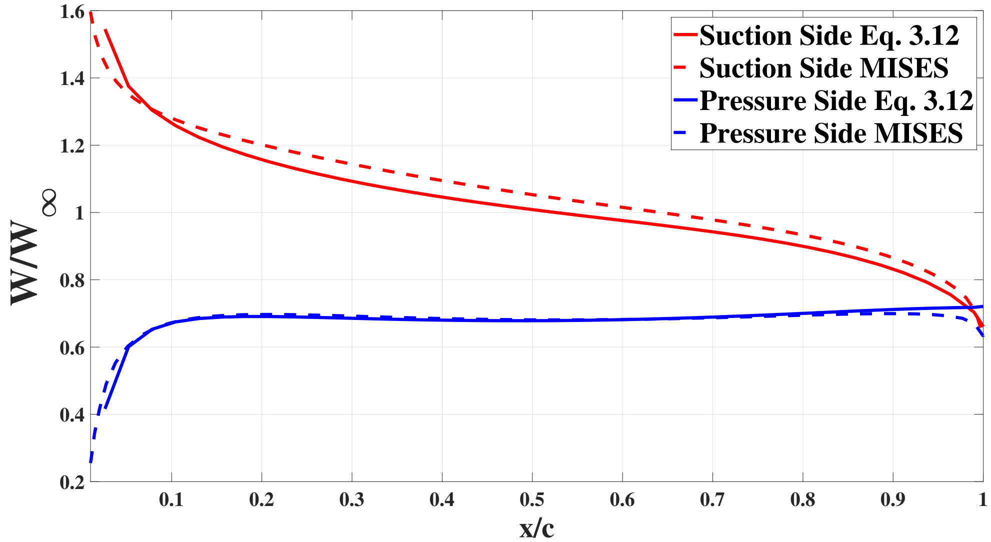


Figure 3-4: Comparison of analytical velocity distribution model with MISES CFD results for cascade 1 - $M_\infty = 0.3$, stagger angle = 25(deg), i = 6(deg), solidity = 1.0

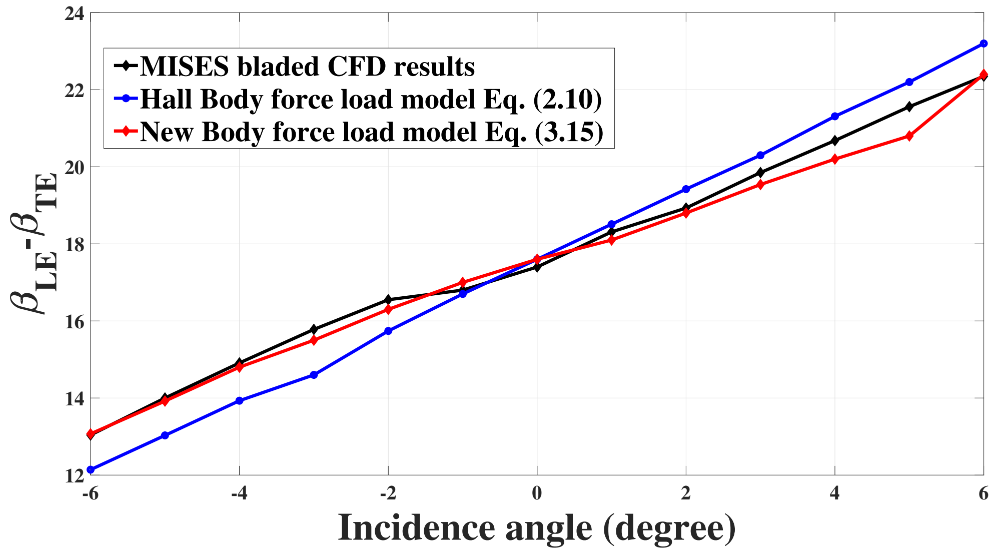


Figure 3-5: Comparison of flow turning of new loading model with MISES CFD results and Hall's model for cascade 1 - $M_\infty = 0.3$, stagger angle = 25(deg), solidity = 1.0

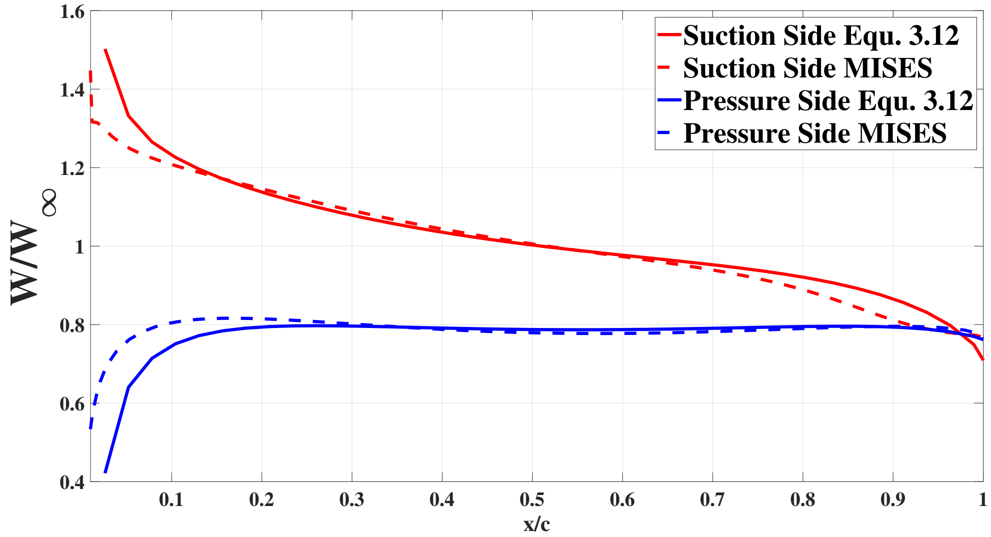


Figure 3-6: Comparison of analytical velocity distribution model with MISES CFD results for cascade 2 - $M_\infty = 0.65$, stagger angle = 30(deg), i = 4(deg), solidity = 1.2

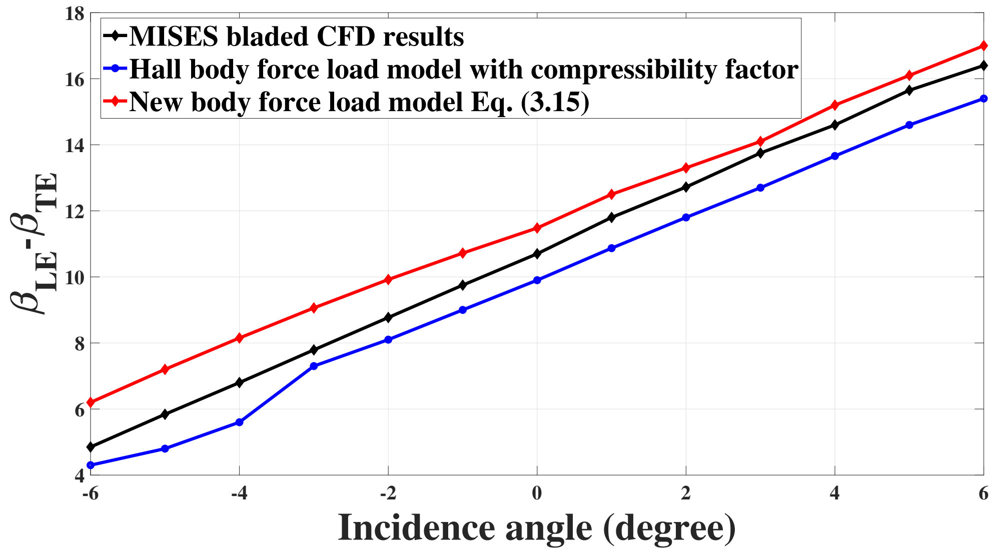


Figure 3-7: Comparison of flow turning of new loading model with MISES CFD results and Hall's model for cascade 2 - $M_\infty = 0.65$, stagger angle = 30(deg), solidity = 1.2

3.3 Denton's Loss Model with Constant Dissipation Coefficient

Since there is no direct method for body force boundary layer calculations, as described in previous Chapters, a constant dissipation coefficient of 0.002 is used in the loss model. This value comes from the Denton's article [24] in which the author of that work has concluded that the dissipation coefficient in turbomachinery applications is around 0.002.

For cascade 1 with the new analytical model, the viscous loss coefficient has been computed by Denton's body force loss model. The viscous loss coefficient is computed as:

$$\omega_{viscous} = \frac{p_{t,LE} - p_{t,TE}}{p_{t,LE} - p_{LE}} \tag{3.16}$$

Figure 3-8 shows the loss coefficient for a range of incidence angles between -6 to 6 degrees. Except the low incidence angles between -1 to 1 degrees, the model has a large discrepancy, generally under-predicting the loss coefficient. At higher incidence angles, the actual average dissipation coefficient values as computed by MISES are high and reach over 0.003 . The velocity distribution and loading model results (in Fig. 3-5) show that the main part of this discrepancy originates from the constant dissipation coefficient assumption, since the edge velocities are well-predicted. In the boundary layer calculations, the maximum dissipation coefficient is generated near the leading edge where the local velocities have higher values as well. To carry out more assessment of this modelling approach, the loss coefficient for cascade 2 at an inlet Mach number of 0.65 has been computed. The results of this simulation are presented in Fig. 3-9. The higher Reynolds number and higher inlet Mach number for this cascade shows that the constant dissipation coefficient assumption is not valid for loss modelling at all. Thus, the loss model should be improved to a more robust model in body force calculations.

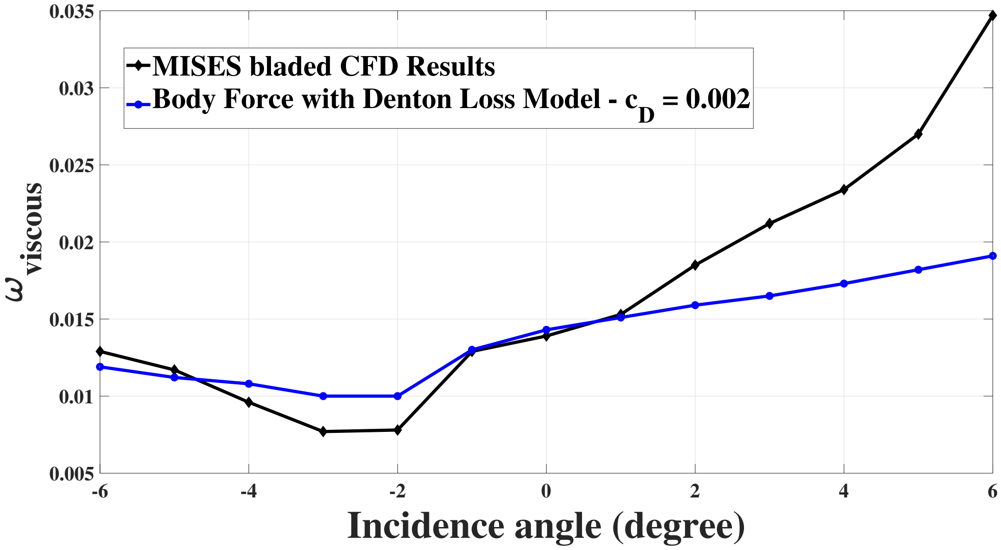


Figure 3-8: Comparison of viscous loss coefficients of new load model with MISES CFD results for cascade 1 - $M_\infty = 0.3$, solidity = 1.0, Re = 335000

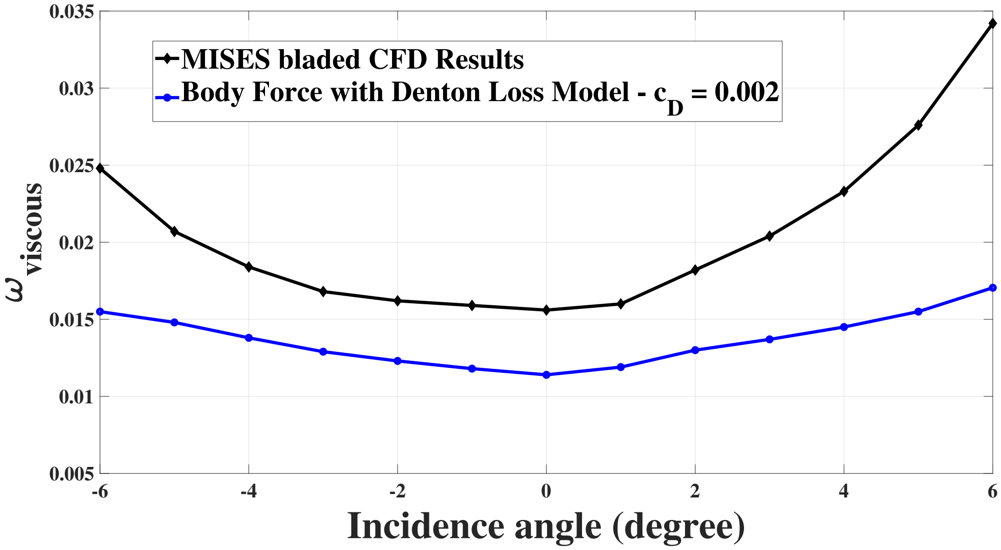


Figure 3-9: Comparison of viscous loss coefficients of new load model with MISES CFD results for cascade 2 - $M_\infty = 0.65$, solidity = 1.2, Re = 700000

3.4 Concluding Remarks

The new analytical loading model uses the chordwise geometry data to calculate the shape parameter integrals along the chord. In 3D blades, the section geometry data for the loading calculations are dependent on the relative flow streamlines, and the streamline chordwise geometry data are not know a priori. Thus, using this model in its current form in 3D cases is challenging. However, in a 2D approach, the new analytical model is a promising method that can capture the flow turning and provides the suction and pressure side local velocities. Furthermore, it was shown that even with the velocity distribution, Denton's loss model with a constant dissipation coefficient is not appropriate for entropy generation prediction in body force modelling. The reason is that this model depends on the cube of velocity so that a minor error in velocity prediction can have a considerable effect on the loss prediction. In addition, the model is dependent on the local boundary layer dissipation coefficient. The results show that a constant dissipation coefficient is not an acceptable idea. At higher incidence angles, higher dissipation coefficients exist. Besides, there is no direct method to calculate the local boundary layer quantities. In the next Chapter, it will be shown that Youngren's [26] loss model, which is the model used in MISES, can be a replacement for loss predictions. That model requires the trailing edge quantities and a novel neural network model along with some reasonable assumptions will be shown to yield a reliable body force loss model.

Chapter 4

A New Body-Force-Based Loss Model

This chapter introduces a new viscous body force model based on the total pressure defect model presented by Youngren [26] used in MISES by Drela and Youngren [25] A simple shock loss model based on the Denton's framework [24] is also presented. The assessments and validation of the viscous and shock models are discussed. The viscous (boundary layer) losses are related to blade geometry and flow parameters using an artificial neural network.

4.1 Viscous Loss Body Force Model

The base of the loss model in this chapter is Eq. 2.17 in which the boundary layer and flow quantities at the trailing edge determine the relative total pressure loss. Assumptions are made to simplify this model.

- The flow velocities on either side of the trailing edge are approximately equal (neglecting the wake area, the properties are uniform, thus the boundary layer edge velocities are the same as that of the outside flow).
- The density across the pitch at the trailing edge is approximately uniform

- Flow outside of the boundary layer is inviscid and isentropic. Shock losses are handled by a separate model.

- The deviation angle at the trailing edge is small ($\cos(\beta_{TE} - \kappa_{TE}) \approx 1$)

Figure 4-1 shows the uniform flow at the trailing edge based on these assumptions.

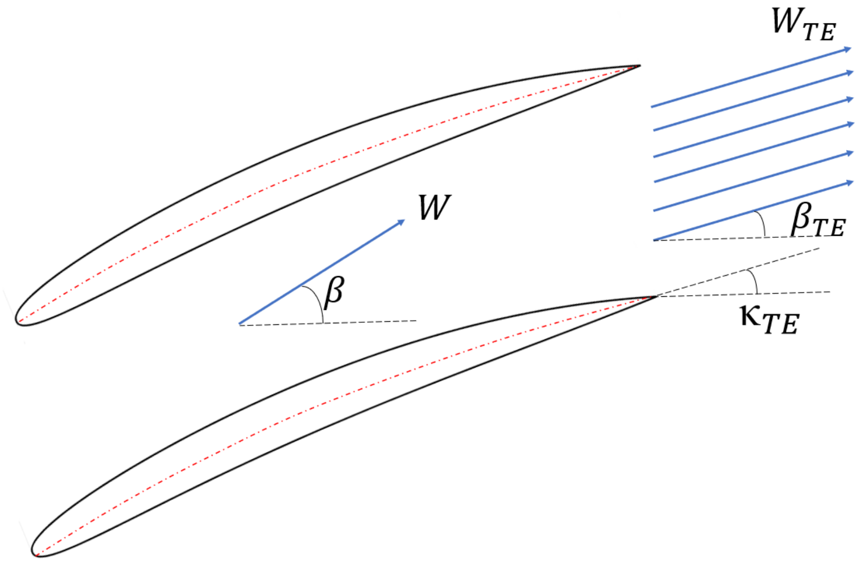


Figure 4-1: Schematic of uniform flow at trailing edge

Now, we return to Eq. 2.17:

$$\bar{\Delta p}^{M}_{t,loss} = p_{t,inlet} - p_{t,exit} = \left[\frac{p_{te}}{p_e}\frac{\rho_e^2 u_e^3}{\dot{m}} b\theta\right]_{exit} \tag{2.17}$$

By applying the isentropic flow assumption outside the boundary layer, we write $\frac{p_{te}}{p_e}$ in terms of the relative Mach number:

$$\frac{p_{te}}{p_e} = (1 + 0.2M^2_{rel,TE})^{\frac{\gamma}{\gamma-1}} \tag{4.1}$$

Applying the assumptions about the flow uniformity and direction at the trailing edge:

$$\dot{m} = \rho_{TE} W_{TE} |n_\theta|_{TE} \left(h - \delta^*_{TE}\right) b \tag{4.2}$$

Thus, the mass-averaged relative total pressure loss in Eq. 2.17 can be written as:

$$(p_{t,rel,LE} - p_{t,rel,TE})_{viscous} \approx \frac{(1 + 0.2M_{rel,TE}^2)^{\frac{\gamma}{\gamma-1}} \rho_{TE} W_{TE}^2}{(h - \delta^*)\, |n_\theta|_{TE}} \theta_{TE} \tag{4.3}$$

where $M_{rel,TE}$ is the relative Mach number, ρ_{TE} is the flow density and W_{TE} is the flow relative velocity at the trailing edge. Also, δ^* is the displacement thickness (sum of displacement thickness of suction and pressure sides $(\delta^*_{SS} + \delta^*_{PS})$) and $\theta_{TE} = \theta_{TE,SS} + \theta_{TE,PS}$.

Further assumptions are made below to interpret this loss in terms of a force per unit volume:

- Boundary layer blockage caused by the displacement thickness may be neglected. In the common compressor operations $\frac{\delta^*}{h} < 0.02$ [76]. This assumption is valid for attached flows. In the next chapter, it will be shown that the assumption is violated in some separated flows. Nevertheless, the aim is to have the most simplified and straightforward computations. Including this would require blade thickness effects be included and updated within the normal force model, which is outside the scope of this dissertation. Thus, the displacement thickness is skipped for simplification purposes.

- The compressor chord length and camber arc length are considered to be almost equal. For circular arc camber airfoils, the chord-length to camber-length ratio is $\frac{2\sin(\chi/2)}{\chi}$, where χ is the camber angle. For a camber of 45 degrees (a high camber angle for a compressor), the chord to camber length ratio is 0.975.

The local loss body force is calculated as:

$$f_p = \frac{dp_{t,rel}}{d\xi} \tag{4.4}$$

where ξ represents the relative streamline direction. Eq. 4.4 with the previous assumption that the chord and camber arc lengths are almost equal in compressor airfoils is developed to:

$$f_p = \frac{dp_{t,rel}}{d\xi} \approx \frac{p_{t,rel,LE} - p_{t,rel,TE}}{c} \tag{4.5}$$

Combining Eq. 4.3 with Eq. 4.5 leads to a loss body force:

$$f_{p,viscous} = \frac{(1+0.2M_{rel,TE}^2)^{3.5}\rho_{TE}W_{TE}^2}{\left(\frac{2\pi r}{B}\right)|n_\theta|_{TE}} \left(\frac{\theta_{TE,SS} + \theta_{TE,PS}}{c}\right) \tag{4.6}$$

where c is the chord length. In the next step, the trailing edge flow quantities need to be estimated at any position on the relative streamline based on the local flow quantities. Mass continuity in a body force with no blockage at the steady-state condition implies that:

$$\rho W \cos(\beta) = \rho W_x = \rho_{TE} W_{TE} |n_\theta|_{TE} \tag{4.7}$$

where β is the local relative velocity angle with respect to the axial axis and ρ and W are the local density and relative velocity at any position within the rotor/stator in body force simulation. This equation assumes that the contraction in passage has a low impact on the flow velocity and the radial velocity is negligible. In Chapter 5, it will be shown that in a rotor case, this assumption is valid for a wide range of span ratios except the hub region. In incompressible flow the axial velocity remains constant from the leading to trailing edge. Thus we have:

$$W_{TE} = \frac{W_x}{\cos(\kappa_{TE})} \tag{4.8}$$

Similarly, assuming that the speed of sound remains constant due to small changes in temperature (in NASA rotor 67 the static temperature at design speed changes from 285 to 295 K which is only 3% variation) we have:

$$M_{rel,TE} = \frac{M_x}{\cos(\kappa_{TE})} \tag{4.9}$$

Therefore, with assuming constant axial velocity and using mass continuity, Eq.

4.6 can be re-written as:

$$f_{p,viscous} = \frac{(1+0.2(\frac{M_x}{|n_\theta|_{TE}})^2)^{3.5}\rho W_x^2}{\left(\frac{2\pi r}{B}\right)(|n_\theta|_{TE})^3}\left(\frac{\theta_{TE,SS}+\theta_{TE,PS}}{c}\right) \tag{4.10}$$

This model accounts for the loss within the blade row and does not take into account the mixing losses. In addition, it does not include the boundary layer blockage effects.

This viscous body force model is a simplified method that requires only the trailing edge momentum thickness and trailing edge blade metal angle. The assessment of the body force model detailed in Section 4.3 shows that all the assumptions made in the model in a 2D cascade have over 90% accuracy. In 3D simulations, a discrete number of the trailing edge metal angles for specified span fractions can be given to the solver and the relevant trailing edge angle at any position in the body force domain can be calculated by interpolation for the local span fraction.

4.2 Shock Loss Body Force Model

Neglecting radius change effects, the relative total pressure is related to the entropy rise using Gibbs equation:

$$\Delta s = -R\frac{\Delta p_{rel,t}}{p_{t,rel}} \tag{4.11}$$

Assuming that normal shock waves appear with local supersonic relative flow, Denton's shock loss in Eq. 2.14 can be inserted in Eq. 4.11 to get the changes of relative total pressure:

$$\left(p_{t,rel,LE}-p_{t,rel,TE}\right)_{shock} = p_{t,rel}\frac{C_v}{R}\frac{2\gamma\left(\gamma-1\right)}{3\left(\gamma+1\right)^2}\left(M_{rel}^2-1\right)^3 \tag{4.12}$$

The shock loss prediction presented in Eq. 4.12 is appropriate for normal shocks [24]. Thus, it over-predicts the shock loss with the same relative Mach number compared to the oblique shock. A volumetric shock loss model is obtained by dividing the total pressure change by the staggered blade spacing:

$$f_{p,shock} = \begin{cases} \frac{p_{t,rel}\frac{C_v}{R}\frac{2\gamma(\gamma-1)}{3(\gamma+1)^2}\left(M_{rel}^2-1\right)^3}{\left(\frac{2\pi r}{B}\right)|n_\theta|} & M_{rel} > 1 \\ 0 & M_{rel} \leq 1 \end{cases} \tag{4.13}$$

Fig 4-2 shows the shock loss coefficient computations in the body force and MISES for cascade 2. The inlet flow angles are the same for all the simulations. However, based on the shock wave effects on the upstream flow, the incidence angles vary. As expected, the model over-predicts the shock losses for high Mach numbers. To have a correct shock calculation, the normal component of Mach on the shock wave should be used in the entropy generation equations [24]. However, in the body force modelling, the shock wave angle and its normal component Mach number cannot be determined. The model has an error of 25% at Mach number of 1.3. This error is acceptable as in a transonic rotor (as will be shown in the next Chapter), only in the outer 30% span does the relative Mach number become greater than one such that shock losses come into the computations.

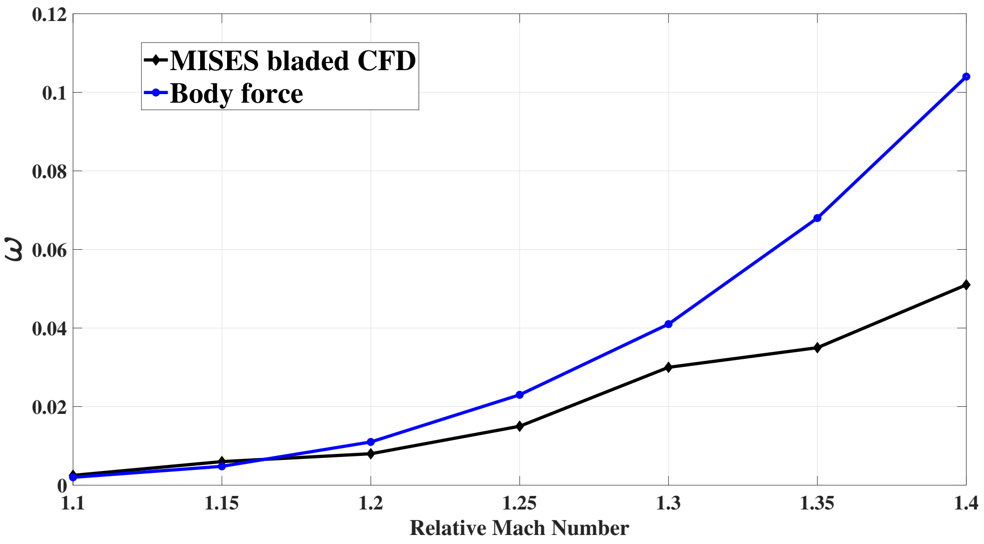


Figure 4-2: Assessment of shock loss for cascade 2

The total volumetric loss is:

$$f_p = f_{p,shock} + f_{p,viscous} \tag{4.14}$$

4.3 Assessment of Viscous Loss Body Force

To validate the viscous loss body force model presented in Eq. 4.6, the trailing edge momentum thicknesses from MISES are prescribed in 2D compressor cascades. In a 2D solver, with uniform inflow, the flow with the body force model is axisymmetric with periodic boundary conditions employed. Hall's loading model was used for flow turning. Grid independence was ensured by increasing the number of axial cells along the blade axial chord (with corresponding increases upstream and downstream) until the loss coefficient stopped changing.

This was done for the cascade 1. The results are shown in Table 4.1. 40 axial cells is sufficient while 20 is likely adequate. The results shown in this chapter are for 40 axial cells.

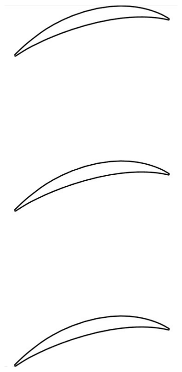
Table 4.1: Grid independence study for body force model in cascade 1 with $M = 0.3$. Zero incidence, $\sigma = 1.0$, Reynolds number based on chord 3.35×10^5.

Cells along chord	ω
10	0.0126
20	0.0135
40	0.0139
60	0.0139

To assess the body force loss model, three cascades are considered. These cascades are shown in Fig. 4-3. Cascade 1 and 2 are the same as those used for assessment in Chapter 3. The geometry information of these cascades are shown in Table 4.2. The viscous loss coefficient for cascade 1 for a range of incidence angles between -6 to 6 degrees is shown in Fig. 4-4. The flow regime is nearly incompressible as the inlet Mach number is 0.3. The maximum error is 7%. Cascade 2 is modelled at a Mach number of 0.65 to assess the model's performance for high subsonic Mach numbers. The viscous loss coefficient for cascade 2 for a range of incidence angles between -6 to 6 degrees are shown in Fig. 4-5. The maximum error is 9%, and for the range of incidence angles considered, the model over-predicts the loss coefficient. In this regime, the axial velocity is not constant and is reduced as the density increases. Higher axial velocities in some chordwise regions lead to over-prediction. As a highly cambered airfoil airfoil, cascade 3 is modelled to assess the impact of high deviation. The viscous loss coefficient for cascade 3 for a range of incidence angles between -6 to 6 degrees are shown in Fig. 4-6. Equ. 4.10, contains a term with the cube of the cosine of the blade metal angle at the trailing edge. Recall that it was assumed that the blade metal angle is equal to the flow angle at the trailing edge. In this cascade, a deviation of approximately 10 degrees occurs. The cube of the cosine of that 10-degree difference in the calculations can create an error of over 8% in loss coefficients. Thus, for highly cambered airfoils with high deviations at the trailing edge, the loss model tends to yield high errors. This error can be doubled if the flow is the high-subsonic regime.

(a) Cascade 1

(b) Cascade 2



(c) Cascade 3

Figure 4-3: Studied cascades for the viscous model assessment

Table 4.2: Geometry data of cascades 1 and 2

parameter	cascade 1	cascade 2	cascade 3
$\frac{t_{max}}{c}$	0.06	0.05	0.085
$\frac{x_{t_{max}}}{c}$	0.4	0.5	0.48
χ(deg)	25	15	50
$\frac{x_{c_{max}}}{c}$	0.5	0.5	0.57
$\frac{R_{LE}}{c}$	0.005	0.001	0.001
φ_{TE} (deg)	10	10	17
λ (deg)	25	30	13

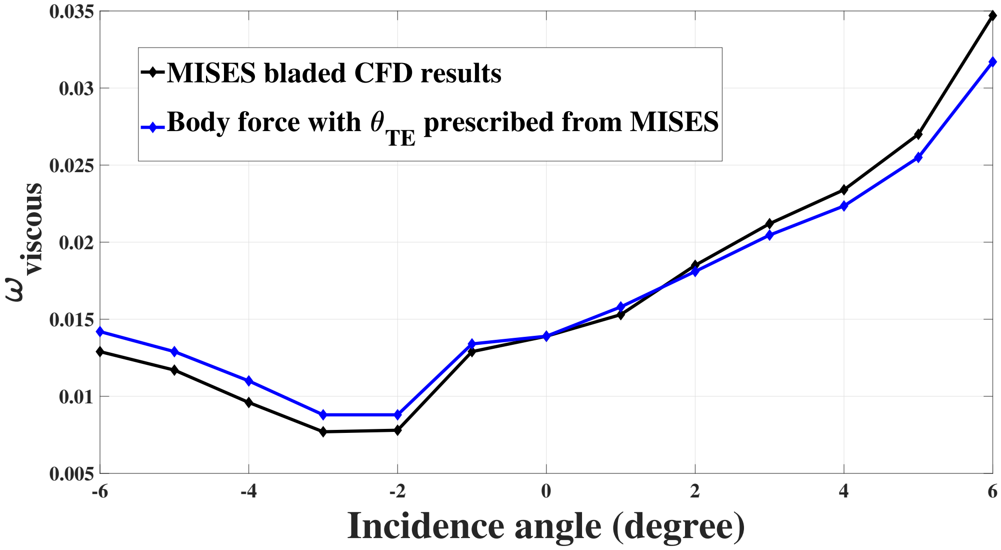


Figure 4-4: Validation of viscous loss model with prescribed trailing edge momentum thickness for cascade 1 ($\sigma = 1.0, M_\infty = 0.3, Re = 335000$)

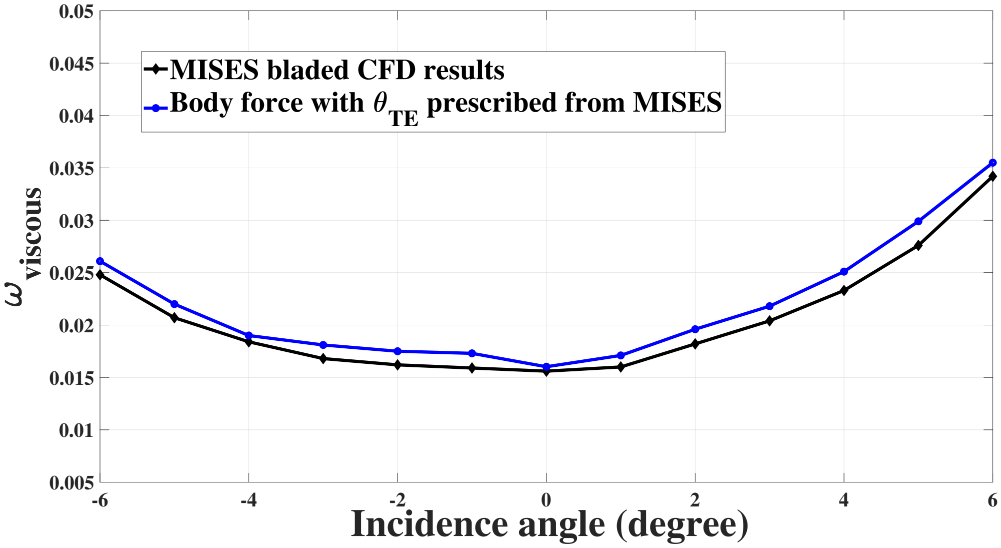


Figure 4-5: Validation of viscous loss model with prescribed trailing edge momentum thickness for cascade 2 ($\sigma = 1.2, M_\infty = 0.65, Re = 700000$)

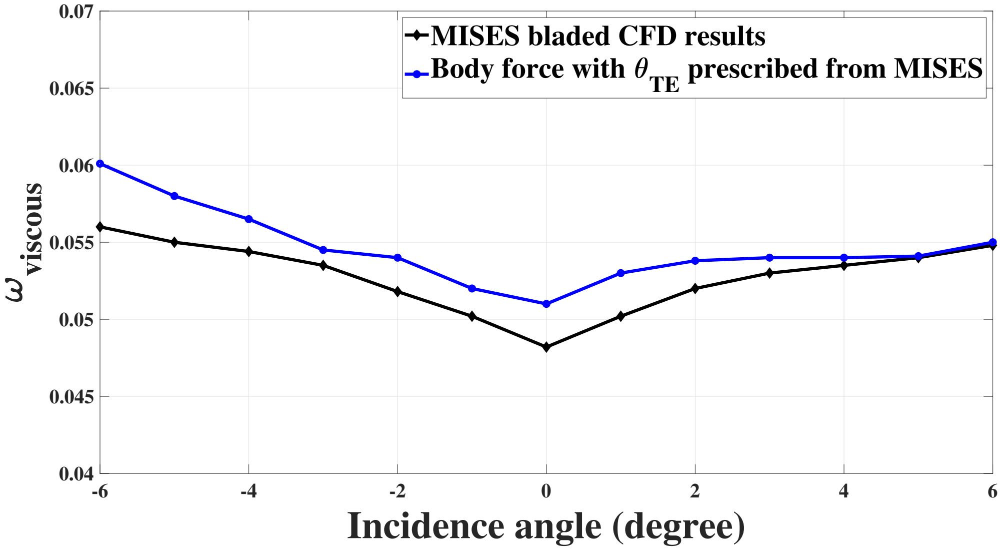


Figure 4-6: Validation of viscous loss model with prescribed trailing edge momentum thickness for cascade 3 ($\sigma = 2, M_\infty = 0.4, Re = 440000$)

4.4 Artificial Neural Network to Estimate Trailing Edge Momentum Thickness

An analytical relationship that can correctly predict the momentum thickness of the boundary layer at the trailing edge of the airfoil is needed for an uncalibrated body force model. Machine learning with artificial neural network tools was used by the author to train a network and provide an analytical model. In the process, a program was developed in the Python language, which builds blade geometry parametrically using the approach from work by Lu et al. [77]. In that paper, a method is introduced wherein the airfoil shape is defined by a set of geometric parameters. A Bezier curve is used to define the camber-line. The analytical equation defining the thickness distribution is:

$$t = t_1\zeta^{0.5} + t_2\zeta + t_3\zeta^2 + t_4\zeta^3 + t_5\zeta^4 \tag{4.15}$$

where ζ is the chordwise coordinate and t is the local thickness. There are five coefficients t_1 to t_5 which are obtained by solving five closure equations:

1) Maximum thickness value is known.

2) Chord fraction of maximum thickness position is known.

3) The slope of equation on the maximum thickness is zero.

4) Trailing edge boat-tail angle is known.

5) Leading edge radius (R_{LE}) is related to the first and second deravative of thickness by:

$$\left|\frac{t''}{(1+t'^2)^{1.5}}\right| = \frac{1}{R_{LE}} \tag{4.16}$$

A Python function for the geometry generation is available in Appendix A. The code transfers the blade geometry to MISES and the mesh generation and flow solution are then carried out. Ten physical and geometric variables are chosen as the main parameters specifying the boundary layer characteristics of any compressor cascade. The variables are shown in Table 4.3. Table 4.4 shows the ranges of the variables and the steps used to define the parameter space.

Table 4.3: Variables for neural network training

Parameter	Definition	
$\frac{t_{max}}{c}$	maximum thickness to chord ratio	input
$\frac{x_{t_{max}}}{c}$	position of maximum thickness to chord ratio	input
χ(deg)	camber angle	input
$\frac{x_{c_{max}}}{c}$	position of maximum camber to chord ratio	input
i (deg)	incidence	input
M_∞	incoming free stream relative Mach number	input
Re_c	chord-based Reynolds number	input
$\frac{c}{h}$	solidity	input
$\frac{R_{LE}}{c}$	leading edge radius to chord ratio	input
φ_{TE} (deg)	trailing edge boat-tail angle	input
$\frac{\theta_{SS}}{c}$	suction side momentum thickness to chord ratio	output
$\frac{\theta_{PS}}{c}$	pressure side momentum thickness to chord ratio	output

The boundary layer equations show that the flow Reynolds number and velocity distribution are needed to determine the boundary layer behaviour. The analytical blade loading model showed that the incoming relative Mach number, incoming flow angle and camber and thickness data are needed to provide the velocity distribution around the blade. Based on these observations, the variables shown in Table 4.3 were chosen as the complete set of parameters that determine the trailing edge momentum thickness. A large dataset from this data generation process was stored. The process was executed automatically until the results were provided for the defined ranges of the variables. Around 400,000 combinations of geometries and physical conditions have been generated and given to the neural network for training.

Taylor et al. [78] showed that one hidden-layer for compressor problems works well. The double-layered neural network structure is used with a hidden layer and an output layer with 40 neurons by the feed-forward back-projection method. Thus, the input vector contains 10 nodes, the hidden layer 40 neurons and the output layer 2 nodes in this research. There is no specific criteria for the optimum architecture in a neural network. However, from one to three layers with ten to forty neurons were tested to assess the optimum structure. The momentum thickness to chord ratio for either side of the blade are the two output variables in the study. It also should be

Table 4.4: Ranges for data generation from CFD used in artificial neural network and steps

Parameter	Range	Step
$\frac{t_{max}}{c}$	0.025-0.15	0.025
$\frac{x_{t_{max}}}{c}$	0.3-0.5	0.1
χ(deg)	10-40	15
$\frac{x_{c_{max}}}{c}$	0.4-0.6	0.1
i (deg)	(-6)-(6)	1
M_∞	0.2-1.6	0.2
Re_c	$1 \times 10^5 - 1.51 \times 10^6$	4.7×10^5
$\frac{c}{h}$	0.5-2	0.5
$\frac{R_{LE}}{c}$	0.001-0.021	0.005
φ_{TE} (deg)	0-10	5

mentioned that both linear (rectified linear unit activation) and non-linear (sigmoid tangent hyperbolic) functions were tested for the training, and due to the non-linear nature of the output quantities and continuous structure of the dependency of output to input, the sigmoid tangent hyperbolic functions performed best for this particular ANN. The difference between the artificial-neural-network-based viscous model and calibration in the body force approach is that the artificial neural network is a trained model operating for any types of blade row with no need for single-passage bladed RANS simulations. Calibrated models are case-specific.

The convergence of the training process for the current study is shown in Fig. 4-7. During an epoch the data are presented for training. In this work, 70% of the data were used for training. Fifteen percent of the dataset was used in the validation set in which the performance of the model is evaluated. Data may be interchanged between the training set and validation set during the iterative optimization process. A test set is not introduced during training and is only used for assessment. The process continues until the weighting matrices and bias vectors yield a minimum error between the actual outputs and the predicted outputs. The error is defined by the mean square error of the predicted and actual output data. The artificial neural network has gone through 19 epochs in this case. After the 14th epoch, there is no significant change in mean square error and the training is completed at the 19th epoch where the best validation performance which is the minimum mean square error within the training process is reached at 2.4×10^{-7}.

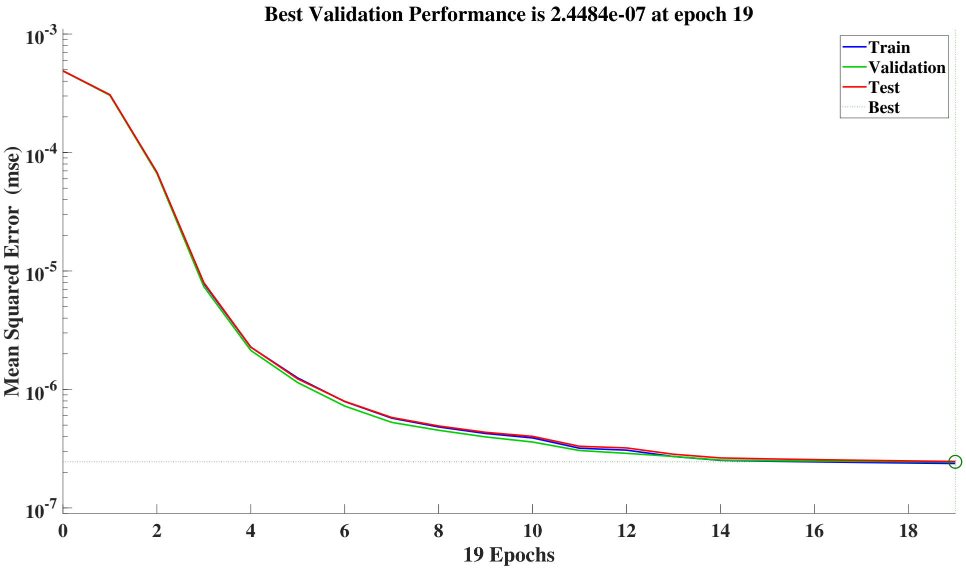


Figure 4-7: Training process error in artificial neural network for momentum thickness data training

The analytical functions to calculate the momentum thickness to chord ratio of either side of the blade at the trailing edge are presented through Eqs. 4.17 to 4.21. $\boldsymbol{X}$ is the input vector as shown in Eq. 4.17:

$$\boldsymbol{X} = \begin{bmatrix} \frac{t_{max}}{c} \\ \frac{x_{t_{max}}}{c} \\ \chi \\ \frac{x_{c_{max}}}{c} \\ i \\ M_\infty \\ Re_c \\ \frac{c}{h} \\ \frac{R_{LE}}{c} \\ \varphi_{TE} \end{bmatrix} \tag{4.17}$$

The model equations are the same as those presented in Chapter 2. Here, $\boldsymbol{X}_{min}$ and $\boldsymbol{X}_{max}$ are:

$$\boldsymbol{X}_{min} = \begin{bmatrix} 0.025 \\ 0.3 \\ 5 \\ 0.4 \\ -6 \\ 0.1 \\ 1 \times 10^5 \\ 0.5 \\ 0.001 \\ 0 \end{bmatrix} \tag{4.18}$$

$$\boldsymbol{X}_{max} = \begin{bmatrix} 0.15 \\ 0.5 \\ 40 \\ 0.6 \\ 6 \\ 1.6 \\ 1.51 \times 10^6 \\ 2 \\ 0.02 \\ 10 \end{bmatrix} \tag{4.19}$$

and the outputs are:

$$F_{4,k} = 2\frac{(F_{3,k}+1)}{2}\left(y_{max_k} - y_{min_k}\right) + y_{min_k} \qquad k = 1, 2 \tag{4.20}$$

where

$$\begin{bmatrix} \frac{\theta_{SS,TE}}{c} \\ \frac{\theta_{PS,TE}}{c} \end{bmatrix} = \begin{bmatrix} F_{4,1} \\ F_{4,2} \end{bmatrix} \tag{4.21}$$

and, y_{min} and y_{max} are:

$$y_{min} = \begin{bmatrix} 0.00099497 \\ 0.00015109 \end{bmatrix} \tag{4.22}$$

$$y_{max} = \begin{bmatrix} 0.037774 \\ 0.018823 \end{bmatrix} \tag{4.23}$$

The weighting coefficient matrices w_1 and w_2 and the bias vectors $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are presented in Appendix B.

The comparison of trailing edge momentum thickness between the ANN model and CFD solver (MISES) for 31 random samples in both the subsonic and supersonic regimes are shown in Figs. 4-8 and 4-9 for trained data. As was mentioned earlier, the test dataset is never trained. They only are used to assess the final performance of the model. The selection of test datasets is randomly made in the training process, and the user only determines what percentage of data be considered as a test set. Samples 10, 12, and 17 are among the predictions which have a large error for the suction side. These data could be from the test dataset, which is not trained. Another reason for some of the discrepancies is that at high incidence angles or in some highly cambered cases, the CFD solver may not have reached convergence so that the provided data for ANN for some variables may be incomplete. However, the average error for the suction side momentum thickness is 11% and for the pressure side it is 17%. The results show that the prediction works for subsonic and supersonic regimes. However, at some cases the error could be high.

The compressor cascades in Fig. 4-3 are considered again to assess the ANN model. None of those cascades are among the trained dataset for ANN. However, cascade 1 and cascade 2 have geometries which are in the ranges where the neural network has been trained. On the other hand, cascade 3 has a high camber and high trailing edge boat-tail angle, which are not within the defined range. Figure 4-10 shows the comparison of the ANN predicted momentum thickness and loss coefficients with data from MISES for incidence angles between -6 to 6 degrees for cascade 1. The maximum

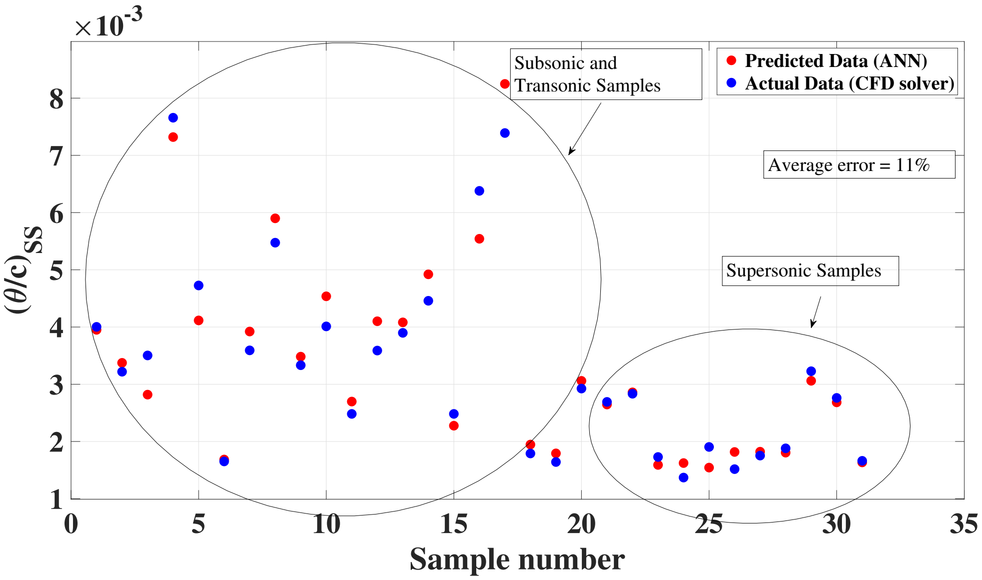


Figure 4-8: Comparison of non-dimensional momentum thickness of suction side for 31 random samples with predicted model of ANN and actual data

error of momentum thickness is at an incidence angle of -6 degrees, where the error is 26%. Figure 4-11 shows the some comparison for cascade 2. The ANN model under-predicts the momentum thickness. The maximum error is 23% at the incidence angle of -6 degrees. At positive incidence angles, the error is up to 8%. Figure 4-12 shows the some comparison again now, for cascade 3. Again, recall that the camber and trailing boat-tail angle of cascade 3 are not within the defined range for the ANN training data to assess how the model behaves beyond the defined geometry ranges. A high discrepancy is shown for this cascade for all incidence angles. This case's error has a minimum of 9% at the incidence of 6 degrees and a maximum of 27% at the incidence of -6 degrees. The current results show that at high negative incidence angles this model can have the accuracy issues.

To show how the ANN model performs beyond the defined range for the incidence angle, Fig. 4-13 illustrates the normalized trailing edge momentum thickness for a range of Mach number and the range of incidence angles from -10 to 10 degrees. All other variables are kept constant. As it is shown, the model predicts the increasing trend of the momentum thickness for the high incidence angles even though the training data did not include those incidence angles. This implies that the model will

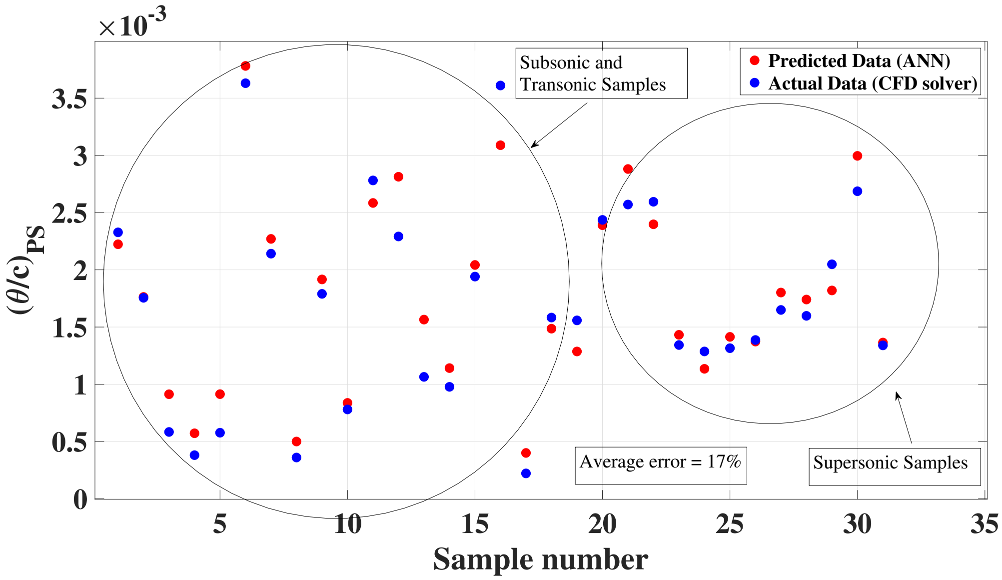


Figure 4-9: Comparison of non-dimensional momentum thickness of pressure side for 31 random samples with predicted model of ANN and actual data

not fail in real conditions where high incidence angles occur.

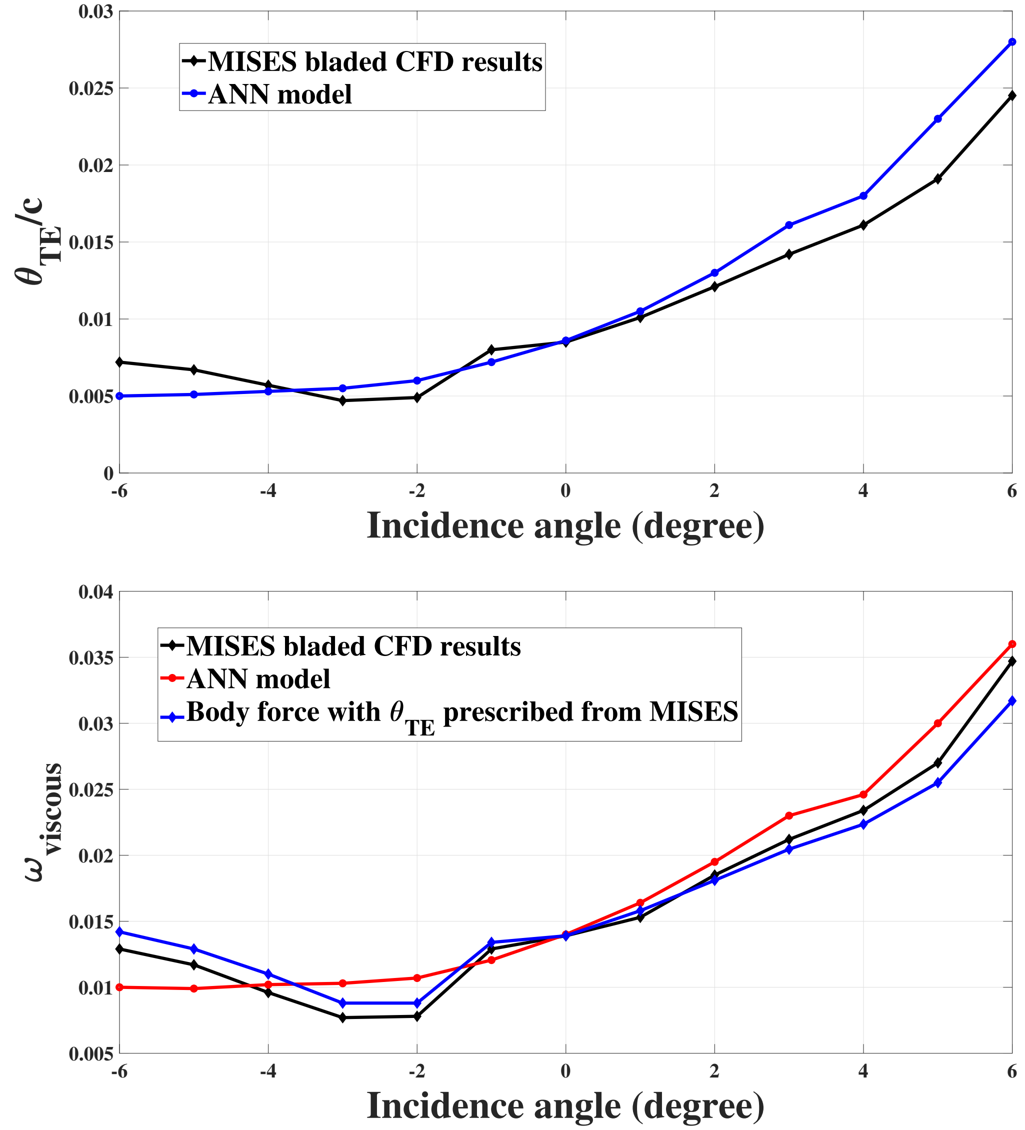


Figure 4-10: Comparison of trailing edge momentum thickness(upper) and loss coefficient (lower) from MISES and ANN model for cascade 1($\sigma = 1.0, M_\infty = 0.3, Re = 3.35 \times 10^5$)

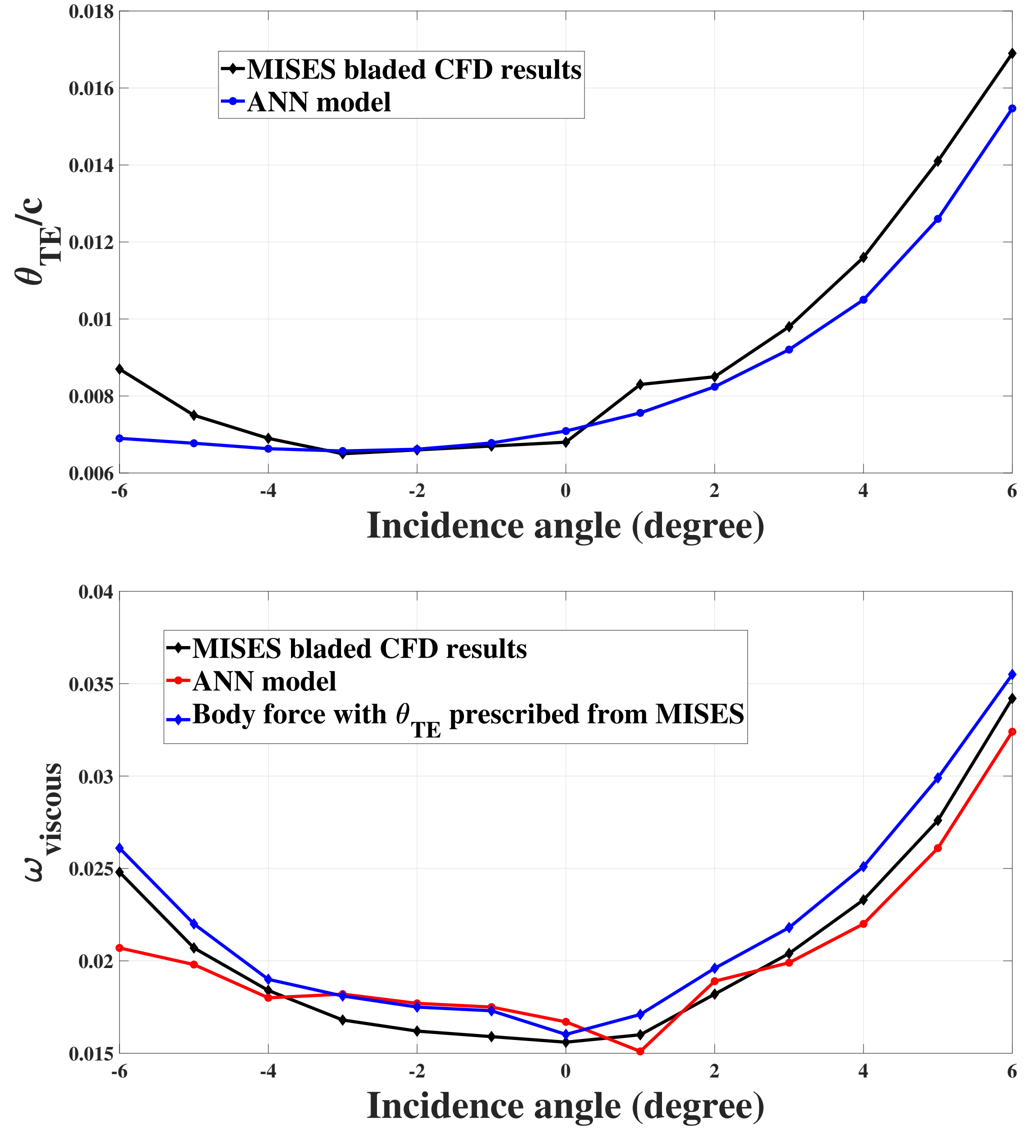


Figure 4-11: Comparison of trailing edge momentum thickness(upper) and loss coefficient (lower)from MISES and ANN model for cascade 2($\sigma = 1.2, M_\infty = 0.65, Re = 7 \times 10^5$)

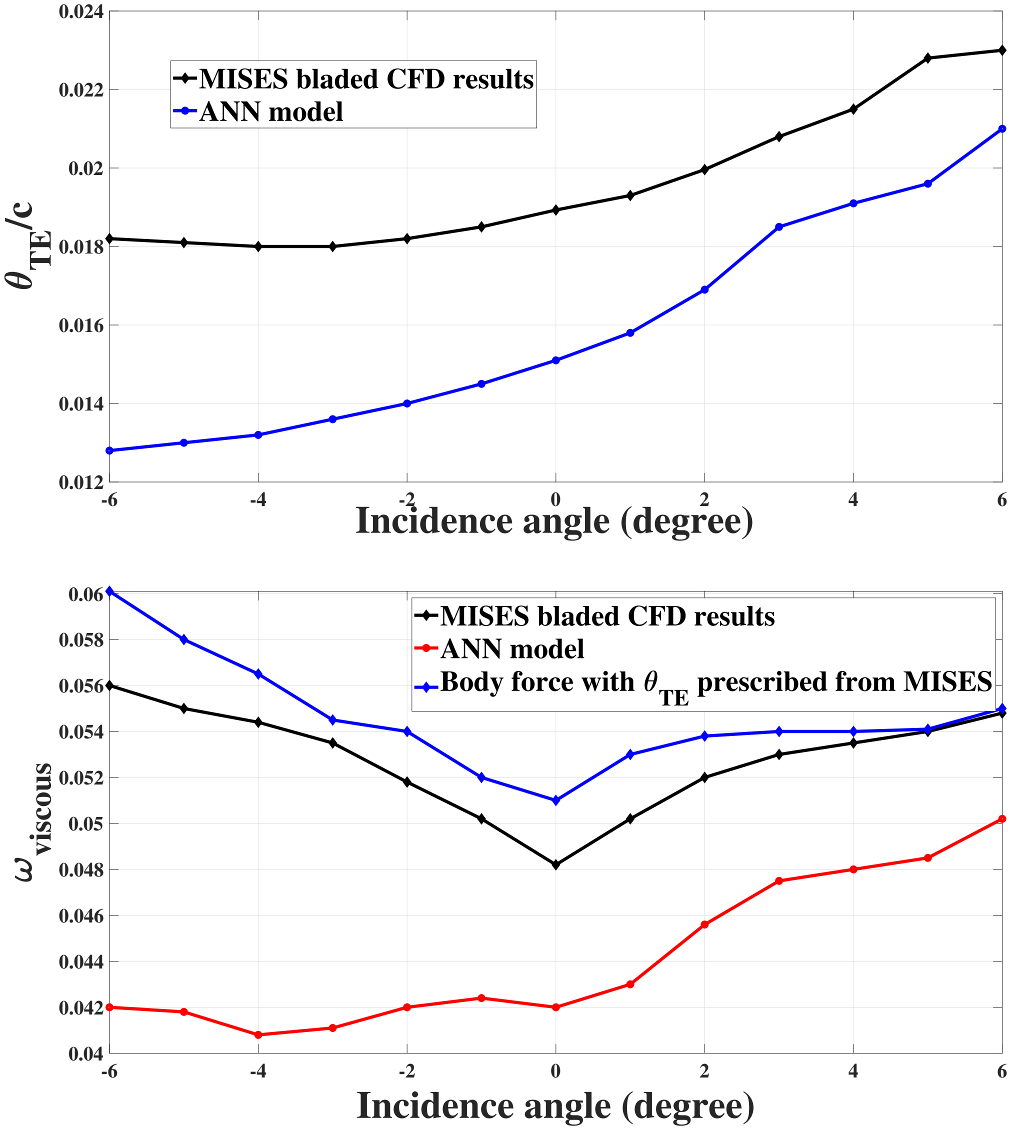


Figure 4-12: Comparison of trailing edge momentum thickness(upper) and loss coefficient (lower) from MISES and ANN model for cascade 3($\sigma = 2.0, M_\infty = 0.4, Re = 4.4 \times 10^5$)

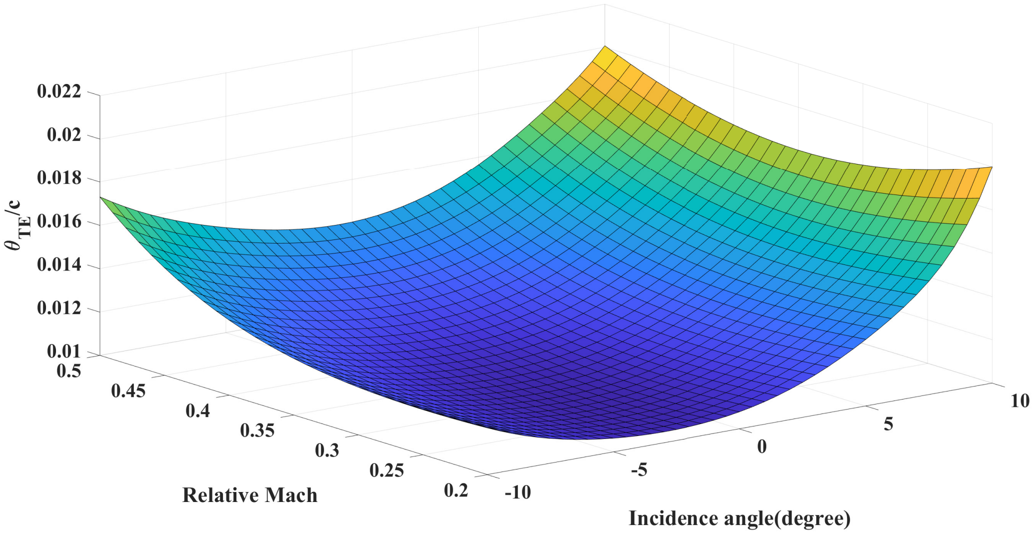


Figure 4-13: Momentum thickness prediction with ANN for a range of incidence angle beyond the defined range in training

4.5 Concluding Remarks

Drela and Youngren's loss model was the base of the new body force loss model in this chapter. The assumptions made in the new body force model can have an accuracy of over 90% in loss predictions with the accurate trailing edge momentum thickness prescription. The ANN model has poor predictions at high negative incidence angles. The prediction of this model for non-trained blades can have good accuracies, but it does not guarantee the accuracy at all conditions. In addition, a weak prediction is shown with around 27% error in non-trained geometries which are not within the range of defined variables used in the neural network training.

In next the chapter, the performance of this model on a 3D rotor case study is investigated.

Chapter 5

Body Force Model Assessment in a 3D Compressor Rotor

In this chapter, NASA rotor 67 [79] is selected as a case study to assess the new loss body force model. Mesh independence and the solver setup are discussed. The results of the simulations are presented in two parts. The first part examines the compressor performance for three different rotational speeds with uniform inlet conditions. The results are compared with experimental data and with single-passage bladed RANS simulations. The second part focuses on non-uniform inlet condition simulation results. The outcomes are compared with bladed URANS simulations.

5.1 Case Study and Simulation Setup

5.1.1 Case Study

The NASA rotor 67 rotor-only blade row has been examined to assess the new body force approach. This test case is a transonic compressor. Table 5.1 describes the geometry and design-speed operational data. The diameter, hub-to-tip radius ratio and aspect ratio are for leading edge positions. π is the mass-averaged total pressure ratio. Figure 5-1 illustrates the compressor rotor.

Given that T_{ref} (288 K) and p_{ref} (101.325 kPa) are the reference condition's

Table 5.1: NASA rotor 67 characteristics at design point

Corrected rotational speed (RPM)	16043
Diameter (m)	0.505
$\frac{r_h}{r_t}$	0.42
π	1.63
$\dot{m}_{cr}$ (kg/s)	32.2
Isentropic efficiency (%)	92
Number of Blades	22
Aspect ratio	1.56

temperature and pressure, the corrected mass flow rate is:

$$\dot{m}_{cr} = \dot{m}\frac{\sqrt{T_{t,in}/T_{ref}}}{p_{t,in}/p_{ref}} \tag{5.1}$$

The corrected rotational speed is:

$$\Omega_{cr} = \frac{\Omega}{\sqrt{T_{t,in}/T_{ref}}} \tag{5.2}$$

This compressor rotor at its design condition has flow in the low subsonic regime (M_{rel} around 0.5 at the hub) and flow in the high subsonic regime at mid-span, and flow in the transonic/supersonic regime at the tip (M_{rel} around 1.3). In the computational domain, the inlet and outlet ducts are eight times the length of the mid-span chord (three times the leading edge tip radius). In the following section, the RANS and URANS setup are introduced. After that, the mesh studies for both the bladed RANS and body force are detailed.

5.1.2 BLADED RANS and URANS Computational Setup

ANSYS CFX R19.1 [80] is used in the bladed RANS and URANS simulations since it is robust for turbomachinery simulations. A steady-state solver in CFX with ideal gas accounting for the flow compressibility is used for the single passage computations with uniform flow, and an unsteady solver with the full-annulus geometry is used for

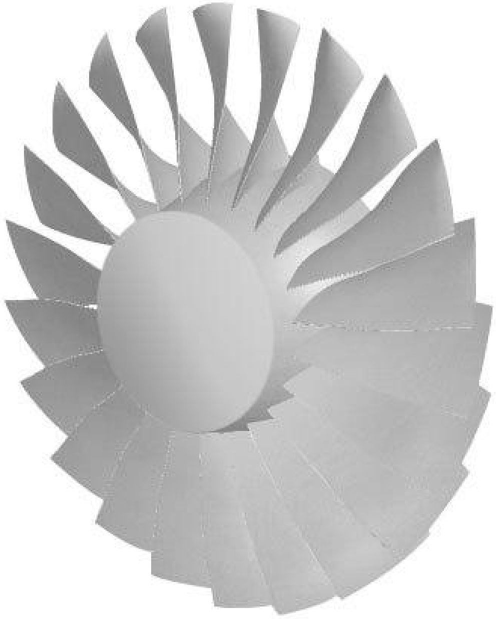


Figure 5-1: NASA rotor 67

non-uniform inflow. The turbulence model employed is the shear stress transport (SST) approach by Menter [81]. A first-order discretization is used for the turbulent transport equations and second-order discretization is used for all other transport equations. The interface between the inlet duct and rotating duct for a steady state is a mixing plane which is 8% of the rotor LE tip radius away from the hub leading edge. Figure 5-2 is a schematic of the single passage solution domain and interface position. One interface is used at the rotor-stationary duct mesh interface. The side surfaces are set as periodic boundary conditions. There is no gap between the rotor tip and the shroud.

The unsteady solver requires a sliding interface between rotating and stationary mesh regions. In the uniform inflow RANS simulations, the inlet flow is considered to have total pressure of 101325 kPa and static pressure is set at the outlet. The inlet total temperature is fixed at 288 K. In this study, to avoid having endwall losses, the wall condition on the shroud and hub have been selected as zero-shear stress conditions. This ensures that all the entropy generation only occurs within the rotor swept volume, due to the profile and shock losses, to assess the new loss model. The no-slip condition is applied on the rotor blade. The boundary conditions for non-uniform inflow with URANS simulations is described in Section 5.3. The results are compared to the experimental data provided by the previous research of Fidalgo et al [82].

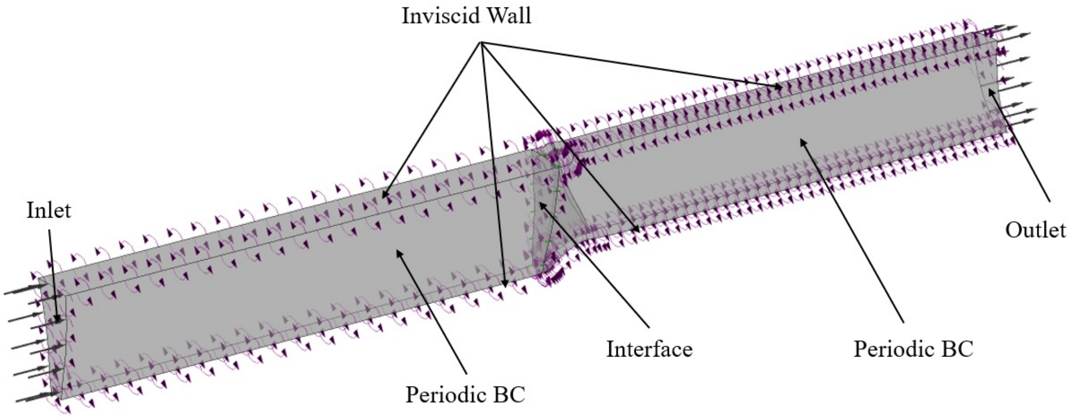


Figure 5-2: Schematic of single passage domain for uniform inflow computations

5.1.3 Body Force Computations Setup

Both uniform and non-uniform inflows in the body force model use a steady-state solver. Since there are no blades in the body force model, there is no unsteady interaction between the rotor blades and the non-uniform inflow, so a steady solution is sufficient. ANSYS Fluent R19.1 [75] was selected for the body force simulations. ANSYS Fluent has the capability to extend its functionality using user-defined functions in the C programming language. An axisymmetric-swirl 2D solver was used for uniform flow. Air with ideal gas properties is the operating fluid. An inviscid solver is used as there are no viscous walls. Figure 5-3 shows the solution domain for uniform inflow calculations with the 2D axisymmetric solver.

The body force parallel to the relative streamline in the body force domain accounts for entropy generation. Several user-defined functions in the C language were

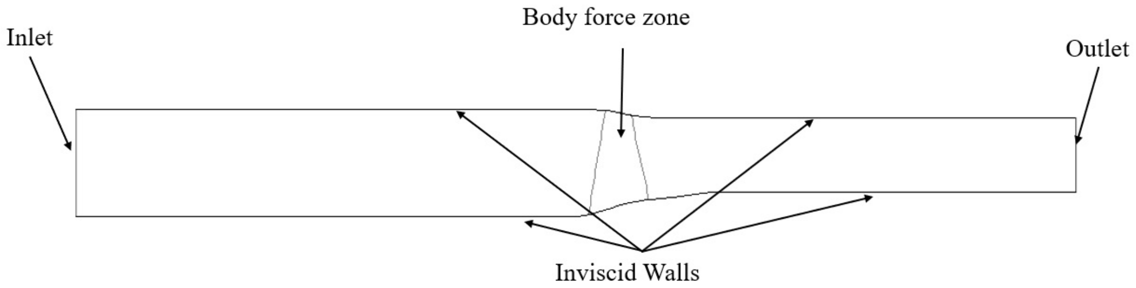


Figure 5-3: Schematic of 2D axisymmetric solution domain for uniform inflow computations

developed to implement the body force approach. The artificial neural network analytical coefficients were defined as matrices, and the viscous body force model was implemented in a “DEFINE ADJUST” macro which computes the source terms and stores them in memory. The calculated body forces are sent to a “DEFINE SOURCE” macro which adds the sources to the governing equations. Also, the loading model for the turning force is calculated within the cells in the body force domain using another “DEFINE ADJUST” macro and is used in every iteration within the solution to add the body forces to the Navier-Stokes and energy equations in “DEFINE SOURCE” macros. Additional transport equations with zero diffusivity and zero sources are solved along with the central governing equations: the analytical loss model requires leading-edge relative Mach number, incidence angle and chord-based Reynolds number. As shown in the viscous body force model discussion in Chapter 4, the local loss calculations in any cell needs to have the properties mentioned above. A preliminary investigation showed that the PISO solver for pressure-velocity coupling performs more robustly than SIMPLE or Coupled solvers. So, PISO coupling was used in the setup. Hall’s loading model with the new viscous model as well as the shock loss model form the base of the body force models employed in this chapter. This is consistent with the 2D setup used foe body force assessment in Chapter 4.

The governing equations with no mass sources used for the body force modelling are:

$$\nabla.(\rho \boldsymbol{V}) = 0 \tag{5.3}$$

$$\rho \boldsymbol{V}.\nabla(\boldsymbol{V}) + \nabla p = \boldsymbol{f_n} + \boldsymbol{f_p} \tag{5.4}$$

$$\rho \boldsymbol{V}.\nabla(h_t) = \rho r \Omega f_\theta - \boldsymbol{W}.\boldsymbol{f_p} \tag{5.5}$$

$$\nabla\left(\rho \boldsymbol{V} \phi\right) = 0 \tag{5.6}$$

where ϕ is any scalar for additional transport equations mentioned earlier.

For flow in a 3D blade row, the chord length needed for the loss model can be estimated using the distance from leading edge to trailing edge at constant span fractions. For the current study, 14 sections of the blade with constant span fraction are defined and the corresponding chord length are prescribed in the calculation. The chord length of the cells in rotor swept volume are interpolated using the local radius and linear interpolation with the available data. The local span fraction is calculated and the corresponding blade geometry is interpolated from the available data.

5.1.4 Mesh Independence Studies

Bladed Case

ANSYS Turbogrid [83] was used to generate the mesh around the blades for the (U)RANS simulations. The computational domain consists of a stationary inlet duct, a rotating rotor region, and a rotating outlet duct. A mesh independence study was carried out to ensure reliable results. The number of cells for a single passage ranged from 2.3×10^5 to 5×10^6 cells. The pressure ratio prediction for several grid cases are shown in Table 5.2. The results for a rotational speed of 9620 RPM, showed that 2×10^6 cells are sufficient. The reason that this speed was chosen is that it is the one used in the URANS simulations discussed later in this Chapter. Fig. 5-4 shows the spanwise pressure ratio for four mesh sizes. The spanwise pressure ratio trend is the same for all of them. However, the figure shows that from 1M cells to 2M cells there are very low changes in pressure ratio. The boundary layer mesh on the blade consists of 14 layers with expansion ratio varying from 1.06 to 1.4 and the y+ reaches to the maximum of 30 on the blade wall. These values demonstrate that the grid resolution is adequate to capture the flow separation within the boundary layer.

Table 5.2: Grid independence study at rotational speed of 9620 RPM and $\dot{m}_{cr} = 21$ (kg/s) for single passage RANS

Number of cells	230k	570k	1M	2M	5M
π	1.209	1.191	1.185	1.183	1.183

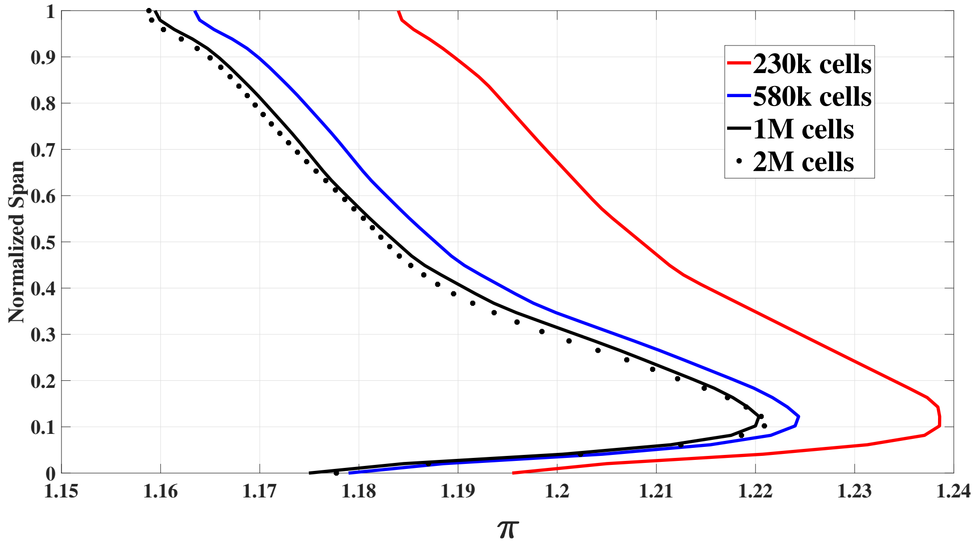
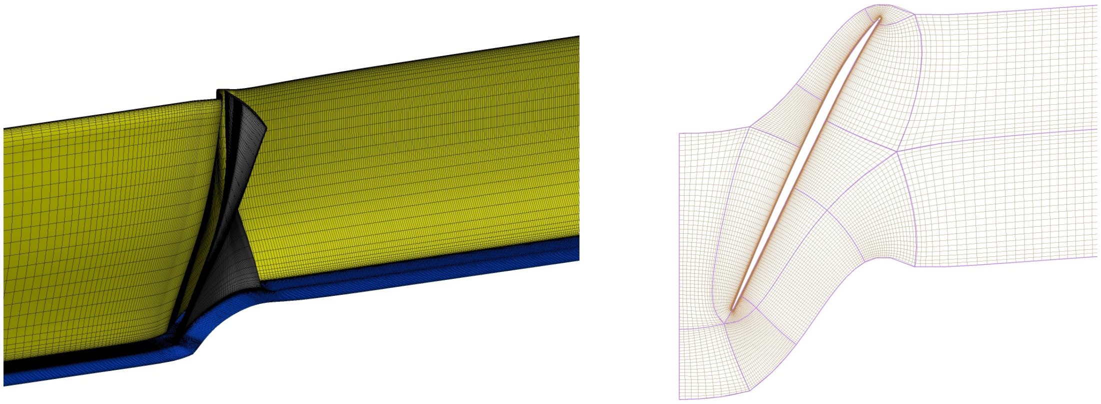


Figure 5-4: Spanwise pressure ratio for four different cells - Bladed RANS simulations

Figure 5-5 shows the single passage bladed mesh and tip section grid. Boundary layer refinements are generated around the blade to properly capture viscous effects. The blade tip has no gap with the shroud to avoid tip-leakage losses.

Body Force

Pointwise [84] software was used to generate a fully structured grid for the body force computations. For this purpose, a 2D grid was made to be implemented in Fluent's axisymetric-swirl solver to assess the uniform inflows and a full-annulus grid was generated to be implemented in Fluent for non-uniform inflow. A grid independence



(a) (b)

Figure 5-5: (a) Single passage bladed grid, (b) blade tip grid for 2M cells

study was carried out to ensure reliable results. The flow pressure and temperature gradients are less sensitive to the number of cells in the radial direction than the axial direction within the body force domain. The hub and shroud are considered to be inviscid walls and do not require dense boundary layer meshes. The studies show that changes in the number of cells from 20 to 30 in the radial direction do not modify the result accuracy. Thus, 20 cells in the radial direction are set for the radial direction. This resolution is sufficient for the non-uniform flow as well. As will be discussed later, the non-uniform simulation and comparison with the bladed URANS results show this grid resolution's validity. However, the results' accuracy is more sensitive to the number of cells in the axial direction. The axial cell size is gradually stretched moving away from the body force zone. This upstream grid resolution enhancement causes the solver to capture the flow suction velocity gradients before entering the rotor domain with reasonable accuracy. Table 5.3 displays the grid independence study results for the 2D axisymmetric body force simulations. The results show that 50 cells in the body force zone in the axial direction are sufficient. Figure 5-6 shows the body force grid structures for axisymmetric (uniform inlet) and Full-annulus (non-uniform inlet). In the full-annulus body-force grid, 120 cells are used in the circumferential direction.

In the next section, the results of the uniform inflow computations at three rotational speeds are presented and discussed.

Table 5.3: Grid independence study of body force at rotational speed of 9620 RPM and $\dot{m}_{cr} = 21$ (kg/s) for single passage

Chordwise cells	15	30	40	50	60
π	1.230	1.195	1.189	1.186	1.186

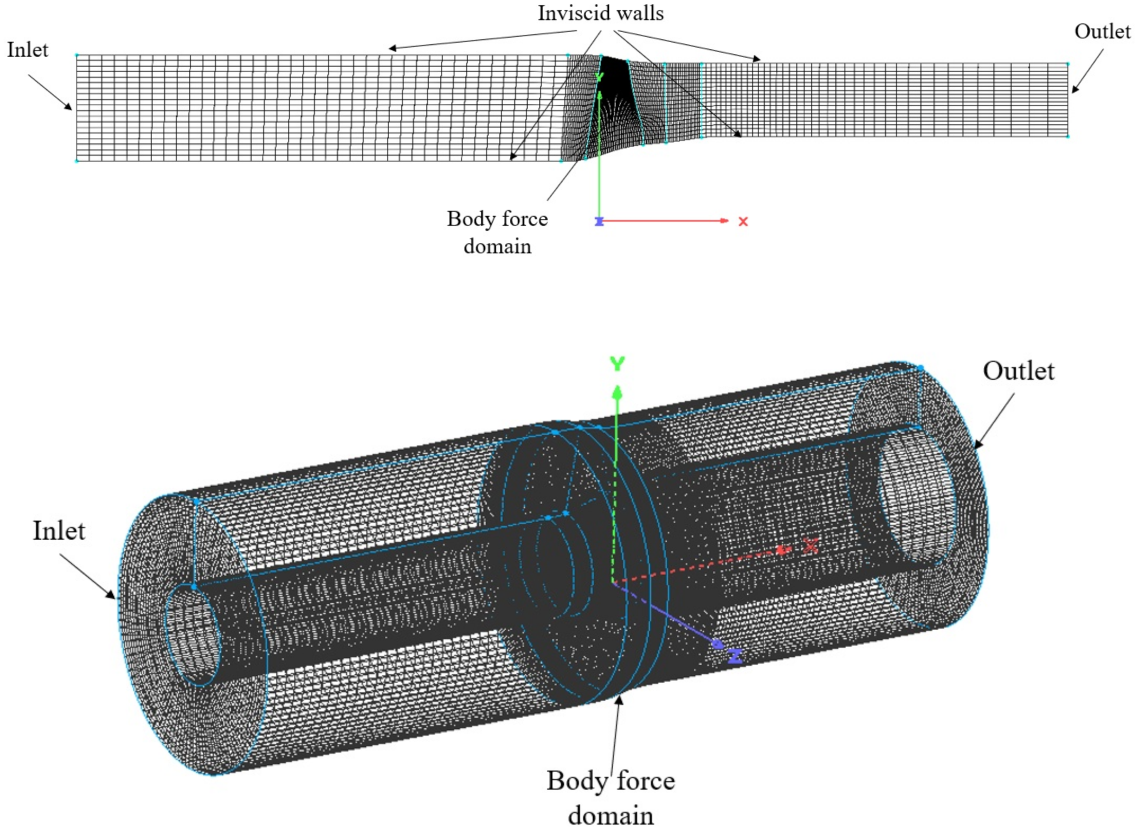


Figure 5-6: 2D axisymmetric grid for uniform flow study (upper), Full annulus grid for non-uniform flow study (lower)

5.2 Uniform Inflow Modelling

Figure 5-7 shows the performance map of pressure ratio vs corrected mass flow rate for NASA rotor 67 at three different rotational speeds. The results of the body force and bladed RANS computations are plotted along with the experimental results from Fidalgo et al. [82]. The experimental setup geometry includes a spinner nose while there is no spinner nose in the numerical bladed RANS simulations. Since the spinner nose affects the hub section velocity and we aim to have the flow field with more compatibility with the assumptions made for the loss model, the RANS simulations do not include a spinner nose in the computations. However, the experimental results are a measure to approximate the accuracy of the computations to see how accurate the model captures the physics. CFD pressure ratios are mass-averages and the averaging plane at the downstream is 4% of the rotor LE tip radius away from the hub after the rotor.

The pressure ratio results show that the turning model and the new viscous model capture the pressure ratio in good agreement with bladed RANS simulations at a variety of different conditions. With the definition of pressure ratio error as:

$$err(\%) = \frac{\pi_{BF} - \pi_{RANS}}{\pi_{RANS} - 1} \times 100 \tag{5.7}$$

the maximum error appears to be at 80% of design rotational speed and the corrected mass flow rate of 24.5 kg/s, where RANS predicts the pressure ratio as 1.42 while body force over-predicts it as 1.46, having 9% error. Comparing the body force model with the experimental data at 100 % of the rotational speed shows the over-prediction of the pressure ratio at lower mass flow rates. At lower mass flow rates where the high incidence angles appear, flow separations occur, and high values of displacement thickness are established. Consequently, the blade loading is altered. There is no means in the current loading model to take into account the boundary layer blockage at high incidence angles, and the flow deviation is underestimated with Hall's model. Below the mass flow rate of 33.5 kg/s, due to intense flow separations, the current RANS model could not converge.

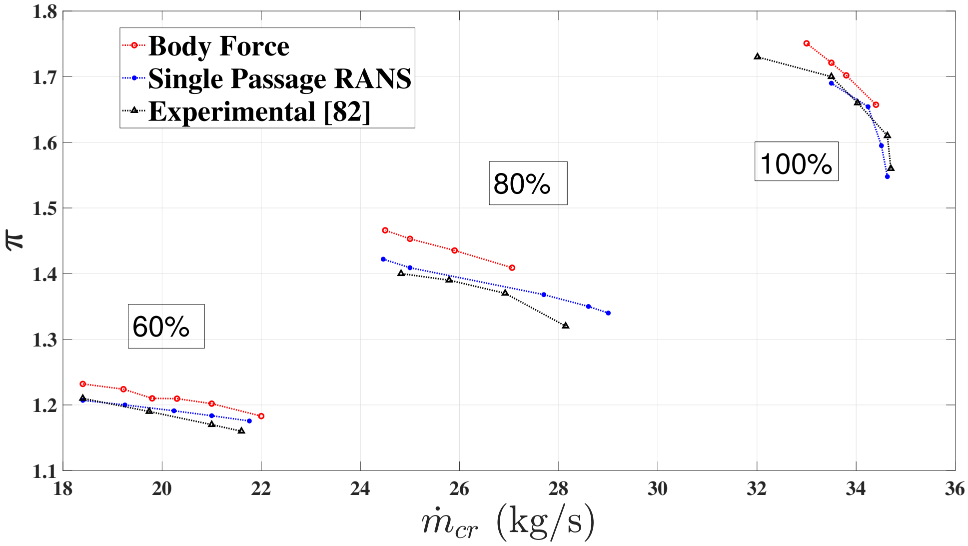


Figure 5-7: Pressure ratio vs corrected mass flow rates for three rotational speeds for NASA rotor 67 - experimental results from [82]

Fig. 5-8 shows the performance map of isentropic efficiency vs corrected mass flow rate for the rotor-only case of NASA rotor 67. The results of body force and bladed RANS simulations are plotted along with the experimental results from Fidalgo et al. [82].

Fig. 5-9 shows the performance map of temperature ratio vs corrected mass flow rate for the rotor-only case of NASA rotor 67. The results of body force temperature ratios demonstrate that the Hall's model over-predicts the input work. That is to say, for the worst case, at the 80% of design speed, a higher pressure ratio over-prediction suffers more from the input work in the body force than the entropy generation under-prediction. The details of the contribution of work input and loss on the total pressure ratio are described in the next Section.

Next, the detailed spanwise results from these computations for all three rotational speeds and two mass flow rates for each speed are presented and discussed. The spanwise results are circumferentially-averaged for the bladed computations.

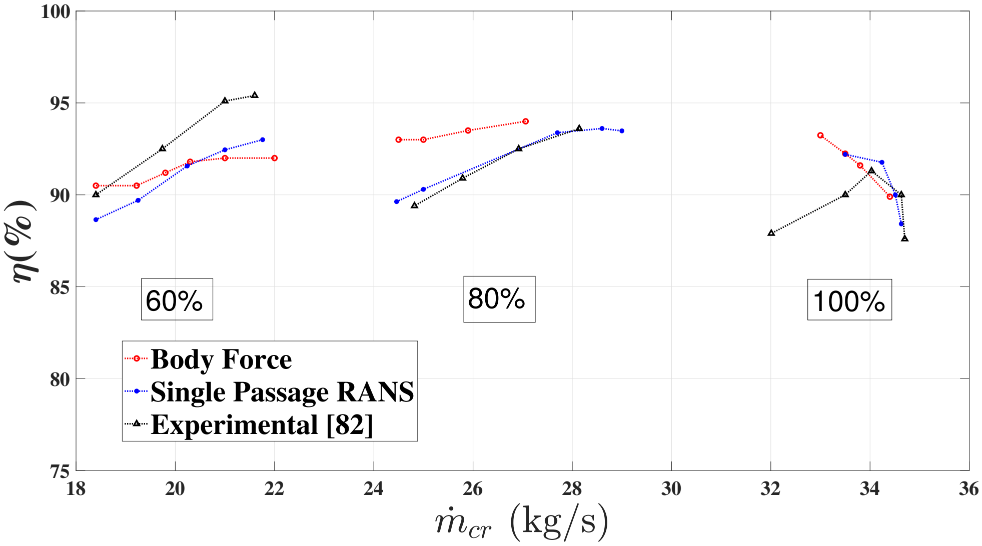


Figure 5-8: Efficiency vs corrected mass flow rates for three rotational speeds for NASA rotor 67 - experimental results from [82]

5.2.1 60% Speed Spanwise Results

At 60% rotational speed, the inlet relative Mach numbers range from 0.3 to 0.78 from hub to tip. There are no shock waves, and there are no endwall and tip gap losses in the bladed RANS simulations. The only loss sources are the blade boundary layers. This speed is a reference for the non-uniform simulation, which is discussed in Section 5.3. First, we consider Hall's model accuracy in terms of the blade loading and flow turning at two different mass flows. One of the flow coefficients is the design condition and the other is a lower flow coefficient which has high incidence angles across the span. The normalized flow coefficient is defined as:

$$\hat{\phi} = \frac{\bar{V}_x^M / U_{mid}}{(\bar{V}_x^M / U_{mid})_{des}} \tag{5.8}$$

where $\bar{V}_x^M$ is the mass-averaged axial velocity just upstream of the rotor, U_{mid} is the rotor blade speed at mid-span, and the subscript *des* refers to design condition quantities.

To compare the body force model with the bladed single passage RANS, two normalized flow coefficients, 0.93 and 1, have been selected. Figure 5-10 shows the

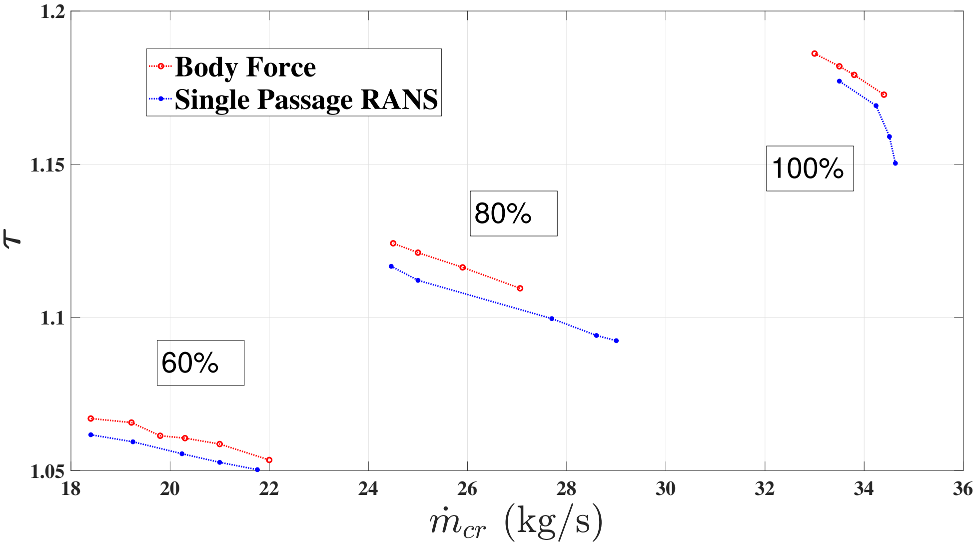


Figure 5-9: Temperature ratio vs corrected mass flow rates for three rotational speeds for NASA rotor 67

spanwise total pressure ratio for $\hat{\phi} = 0.93$. The results show that the body force model over-predicts the pressure ratio and the accuracy reduces as the span ratio increases from 30% to 100%. The overall pressure ratio using the body force model is 1.23 and the bladed single passage RANS predicts the overall pressure ratio to be 1.21. Thus, the error for the overall pressure ratio is 9.5% while the local error reaches a maximum of 21% at the tip.

Figure 5-11 demonstrates the spanwise total temperature ratio for $\hat{\phi} = 0.93$. The overall temperature ratio is over-predicted by 4%. However, unlike the pressure ratio, the temperature ratio is well-predicted at the tip by the body force but gets worse at 50% to 90% span ratio. To assess the contributions of loss and input work to the pressure rise over-prediction, spanwise isentropic efficiency is plotted in Fig 5-12. The results show that the prediction of entropy generation and relative total pressure drop is predicted with good accuracy by the body force model from 20% to 80% span. However, at the hub and near the tip, the isentropic efficiency is over-predicted. The velocity vector contours in these regions are provided to illustrate the physical conditions of the boundary layers.

Figure 5-13 presents the relative velocity vectors for the rotational speed of 60%

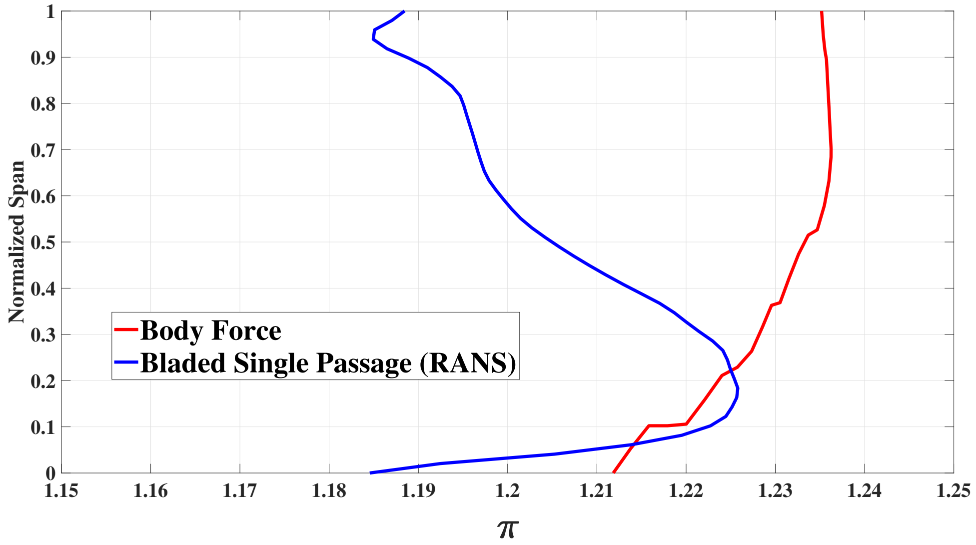


Figure 5-10: Spanwise pressure ratio at 60% speed for corrected mass flow rate of 18.8 kg/s ($\hat{\phi} = 0.93$)

with the normalized flow coefficient of $\hat{\phi} = 0.93$ at the tip. Figure 5-13a shows the leading edge zone. The streamlines on the suction side attest that flow separation occurs near the leading edge. This early separation leads to the higher displacement thickness, and Fig. 5-13b shows that the suction surface boundary layer is very thick at the trailing edge. Thus, a high deviation with a high blockage occurs. A couple of physical phenomena occur, which may cause the viscous model to fail at these conditions. First off, the model assumes that the flow at the trailing edge is parallel to the trailing edge blade metal angle. In the current condition, the velocity vectors at the suction side do not follow the aforementioned assumption. Secondly, velocity values at either side of the blade at the trailing edge are different. Thirdly, the displacement thickness is not negligible: the estimation of displacement to pitch ratios for the two normalized flow coefficients from the bladed single passage RANS simulations are shown in Fig. 5-14. More importantly, the radial velocity changes at hub and tip due to the contraction of the gas path leads to deviation from the assumption that the axial velocity remains constant along the streamline. The assumption of a constant axial velocity is further violated when the effects of blockage are also significant in the flow path.

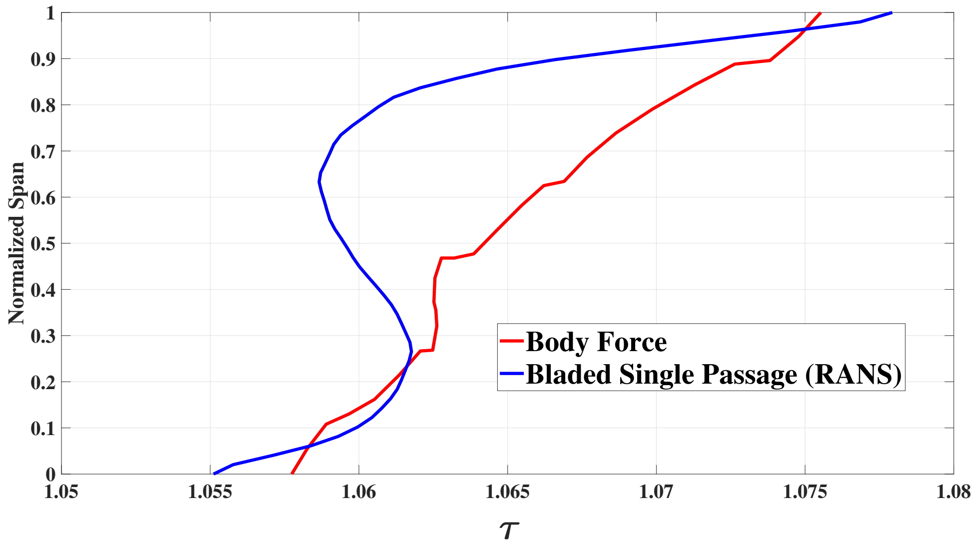


Figure 5-11: Spanwise temperature ratio at 60% speed for corrected mass flow rate of 18.8 kg/s ($\hat{\phi} = 0.93$)

Figure 5-15 presents the relative velocity vectors for the rotational speed of 60% with the normalized flow coefficient of $\hat{\phi} = 0.93$ at the hub. The low pitch along with a high separation zone with high blockage at the hub causes higher flow acceleration compared to the other radial sections. Axial velocity contours for both body force and bladed single passage simulations on the meridional plane at the rotational speed of 60% with normalized mass flow coefficient of $\hat{\phi} = 0.93$ are shown in Fig. 5-16. The blockage effects on the axial velocity in the near-hub region for the bladed case is visible. However, the story is different for the body force simulation since there is no blockage. Consequently, the flow acceleration is less compared to the bladed simulation at the hub region. Nevertheless, the assumption of constant axial velocity at mid-span for the body force simulations appears reasonable.

The impact of changing suction (pressure ratio) on axial velocity within the blade row can be considered by looking at the normalized spanwise mass fluxes at the leading and trailing edges. Two planes located 4% of the rotor LE tip radius from the hub's leading and trailing edges in the axial direction are used for the mass-flux plots. Figure 5-17 shows the spanwise mass flux at the leading edge. The body force model yields more suction at 50% span at the leading edge. However, the bladed

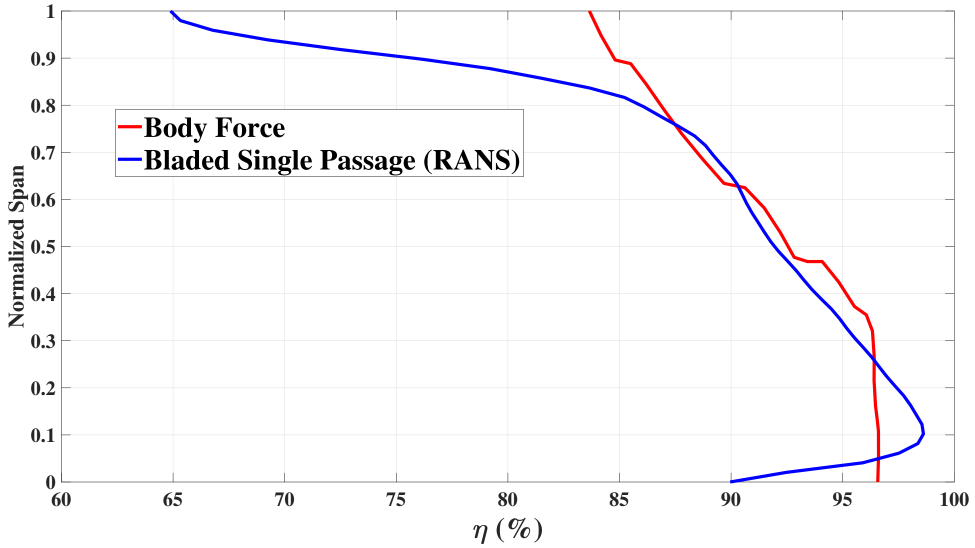
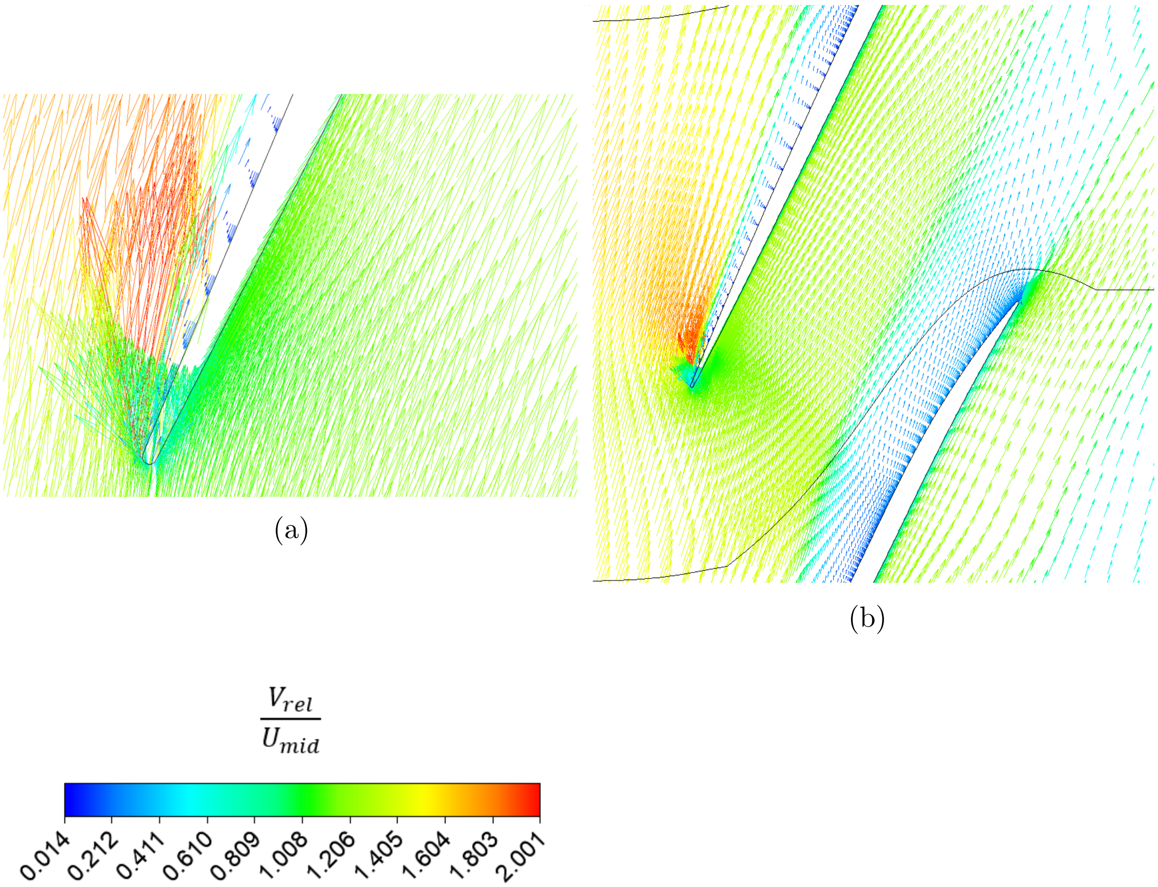


Figure 5-12: Spanwise isentropic efficiency at 60% speed for corrected mass flow rate of 18.8 kg/s ($\hat{\phi} = 0.93$)

RANS computations yield almost uniform flow at the leading edge. This comes from the blade blockage impact on the upstream flow. While there is no blade in the body force simulation, the mid-span streamlines see smaller stream-tubes compared to the hub where a radial change in the hub curve leads to a stream-tube with a larger area. Consequently, a lower axial flow is conducted through the hub region. This can be seen in Fig. 5-16 for the body force flow field. However, in the presence of the blades, the lower pitch at the hub accelerates the incoming flow and reduces the leading edge incoming axial velocity non-uniformity compared to the body force model. Figure 5-18 shows the spanwise mass flux at the trailing edge. Due to high blockage at low span fractions, the flow acceleration and mass flux from the leading to the trailing edge are higher compared to higher span fractions. Consequently, mass flux decreases as the span fraction increases for the bladed case. A loss acting in the opposite direction of the streamline affects the mass flux. In the tip region where a thick turbulent boundary layer is generated, high shear work acts on the flow which reduces the mass flux compared to lower span fractions. Figure 5-12 showed that a high entropy generated region appears at the tip region, and the blades experience fewer losses in the lower span. Thus, higher blade blockage and lower loss at lower


(a)

(b)

Figure 5-13: Streamline velocity vectors at the tip at 60% speed and corrected mass flow rate of 18.8 kg/s, (a) leading edge view, (b) a pitch spacing view

radii cause the higher accelerated axial velocity and mass fluxes in the first half span.

To further evaluate the loss body force model, the spanwise behaviour of the off-design flow coefficient ($\hat{\phi} = 0.93$) is compared with the design flow coefficient ($\hat{\phi} = 1$). The difference between the viscous effects for these two conditions is shown in Fig. 5-14, where it is clear that the design flow coefficient has a negligible displacement thickness compared to the off-design state. For this purpose, to investigate the performance of the rotor at the design flow coefficient, Figures 5-19 to 5-21 show the spanwise total pressure ratio, total temperature ratio and isentropic efficiency for the flow rate of 20.3 ($\hat{\phi} = 1$), respectively.

At the flow coefficient $\hat{\phi} = 1$ the rotor-exit relative total pressure at the tip is lower than at mid-span for both body force and bladed RANS computations. However,

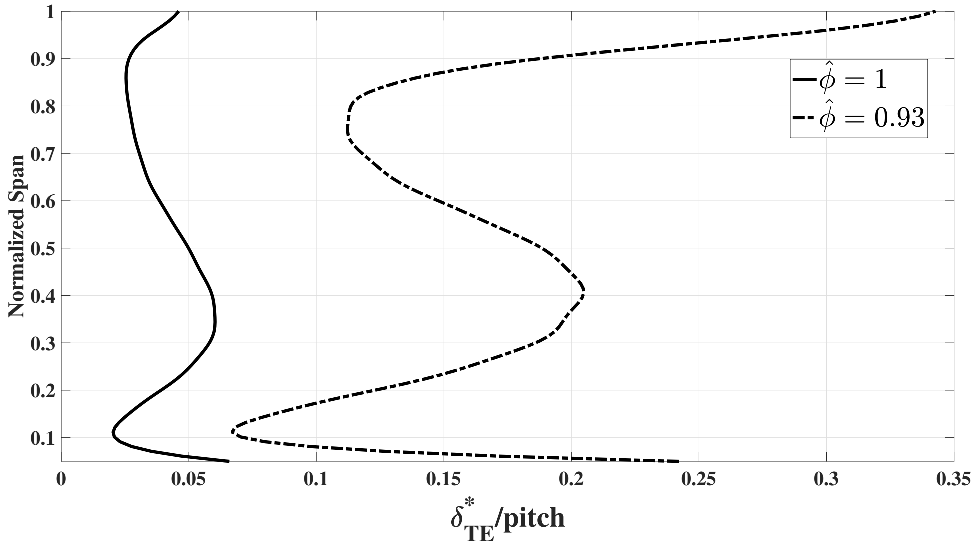


Figure 5-14: Spanwise boundary layer displacement to pitch ratio at 60% speed for corrected mass flow rates of 18.8 kg/s ($\hat{\phi} = 0.93$) and 20.2 kg/s ($\hat{\phi} = 1$)

unlike at $\hat{\phi} = 0.93$ where the temperature ratio in the body force at the tip was the same as the bladed RANS, the temperature ratio at the tip for $\hat{\phi} = 1$ is over-predicted. This implies that the main part of the over-predicted pressure ratio at the tip comes from the over-prediction of the total temperature ratio. Figure. 5-21 shows that the over-prediction of the isentropic efficiency at the tip is lower compared to what occurs at $\hat{\phi} = 0.93$. While the body force has a difference of 7% compared to the bladed RANS at the tip for $\hat{\phi} = 1$, it has a high difference of 19% for $\hat{\phi} = 0.93$. The boundary layer blockage is the principle reason for the different behaviours in the loss prediction. As was shown in Fig. 5-14, the design state encounters a negligible displacement thickness at the tip compared to $\hat{\phi} = 0.93$. A lower displacement thickness yields a less flow deviation angle at the trailing edge. Thus the flow is more aligned to the blade metal angle than for $\hat{\phi} = 0.93$. So at $\hat{\phi} = 1$, the flow better matches the assumptions in the viscous loss model. In addition, the lower displacement thickness has a reduced impact on the blade de-cambering. The body force loading model does not include the re-cambering in the computations. The comparison of Figs. 5-17 and 5-22 shows that the upstream mass flux is not influenced by the viscous effects. However, Fig. 5-23 indicates that the mass flux is more uniform at the trailing edge

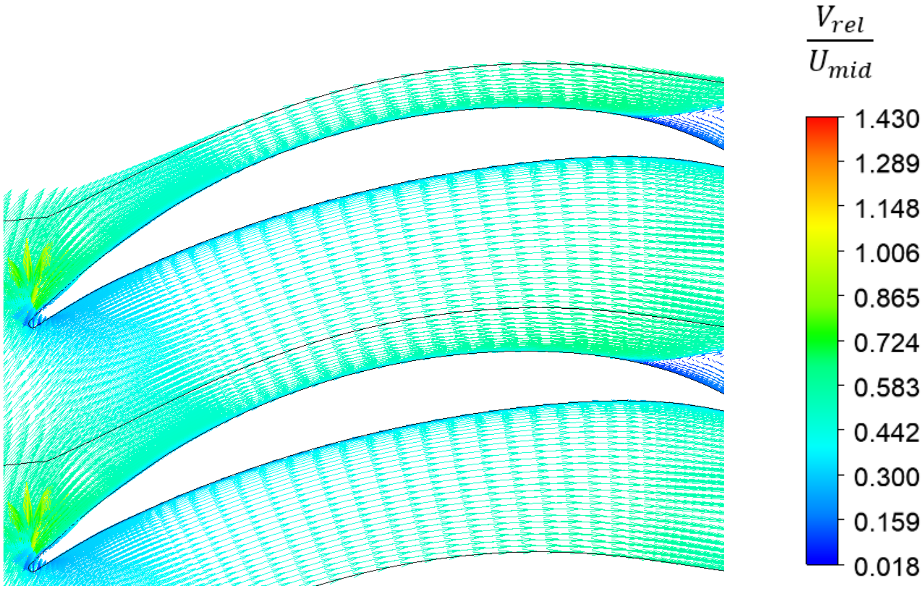


Figure 5-15: Relative velocity vectors at the hub at 60% speed and corrected mass flow rate of 18.8 kg/s ($\hat{\phi} = 0.93$)

for the body force computation since there is no blade blockage affecting the hub section. Overall, the new loss model has a good loss prediction. Both the body force and the bladed RANS estimate the total isentropic efficiency to be 92%.

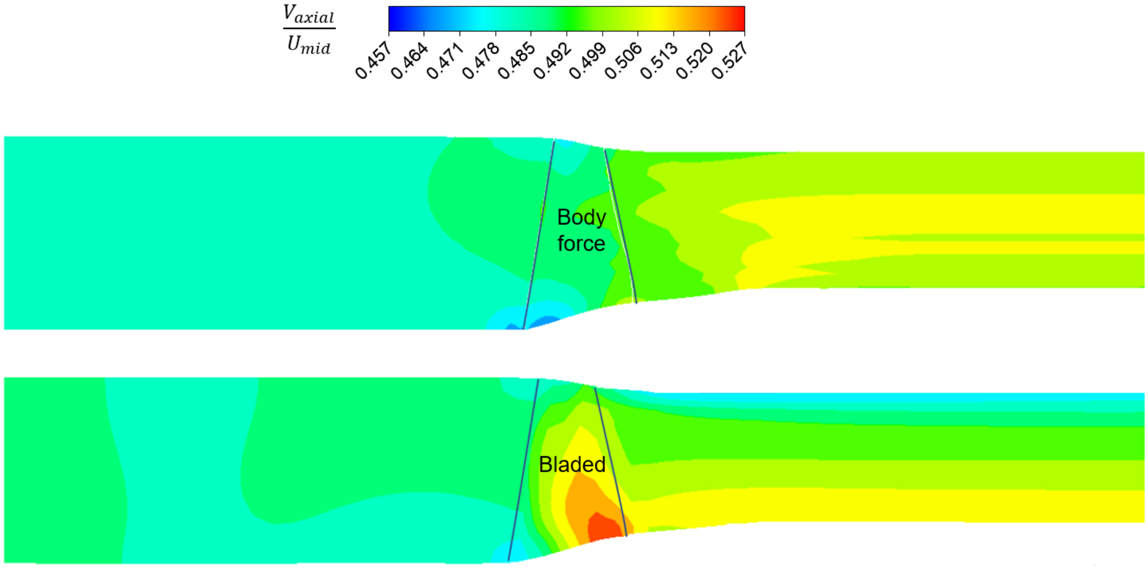


Figure 5-16: Normalized axial velocity contour on meridional plane for 60% speed with corrected mass flow rates of 18.8 kg/s ($\hat{\phi} = 0.93$)

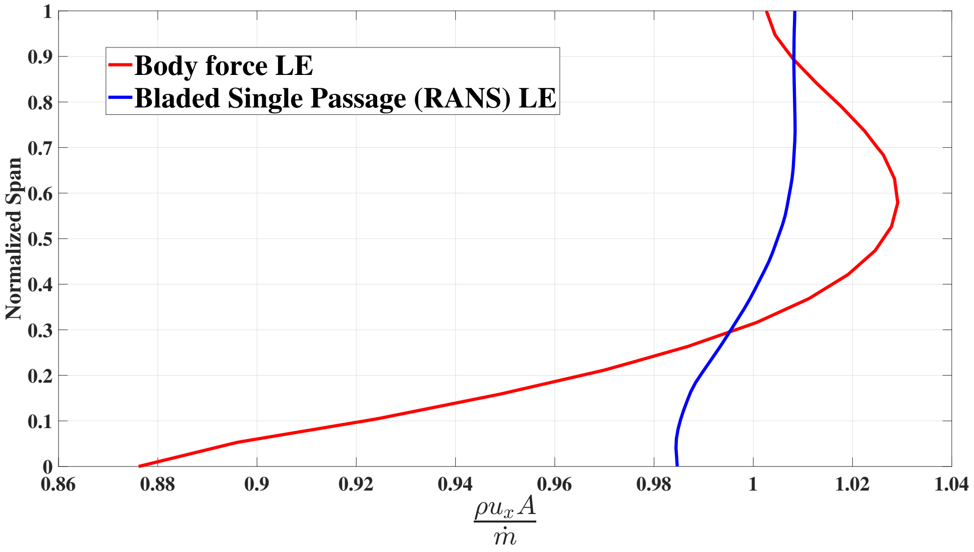


Figure 5-17: Spanwise non-dimensional mass flux at leading edge - 60% speed for corrected mass flow rate of 18.8 (kg/s)

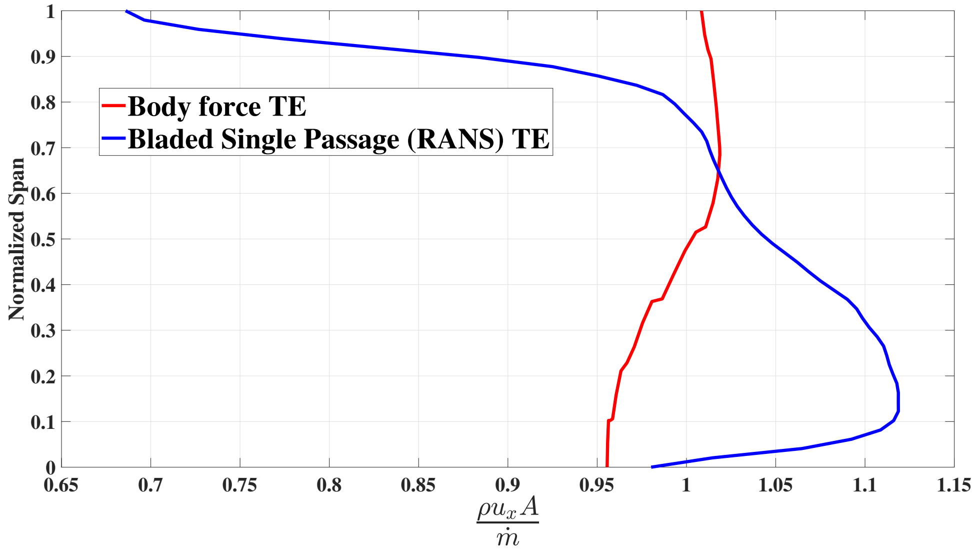


Figure 5-18: Spanwise non-dimensional mass flux at trailing edge edge - 60% speed for corrected mass flow rate of 18.8 (kg/s)

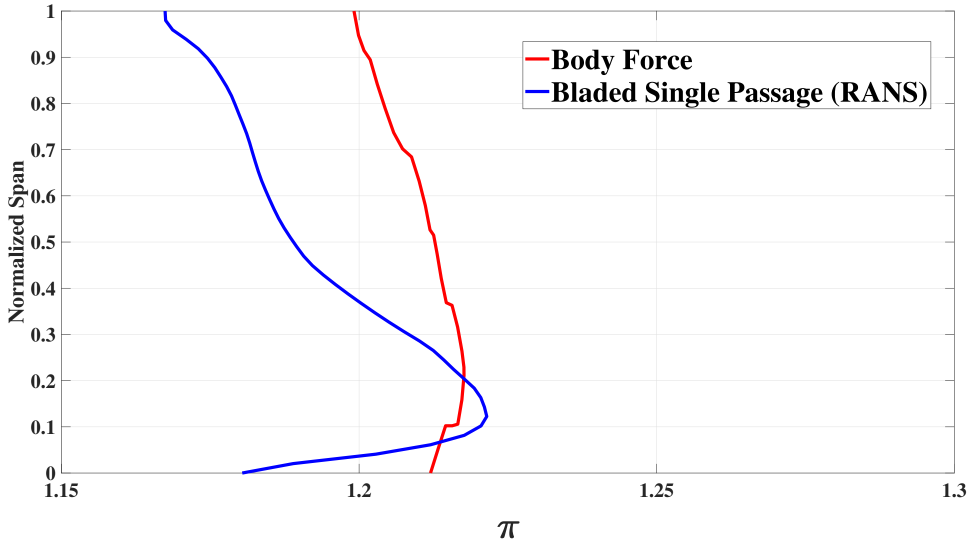


Figure 5-19: Spanwise pressure ratio at 60% speed for corrected mass flow rate of 20.2 kg/s ($\hat{\phi} = 1$)

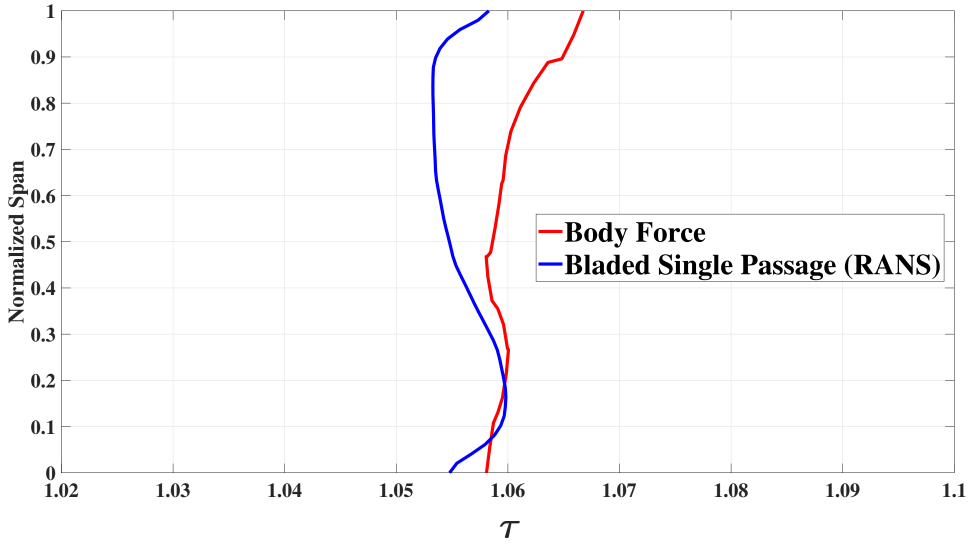


Figure 5-20: Spanwise temperature ratio at 60% speed for corrected mass flow rate of 20.3 kg/s ($\hat{\phi} = 1$)

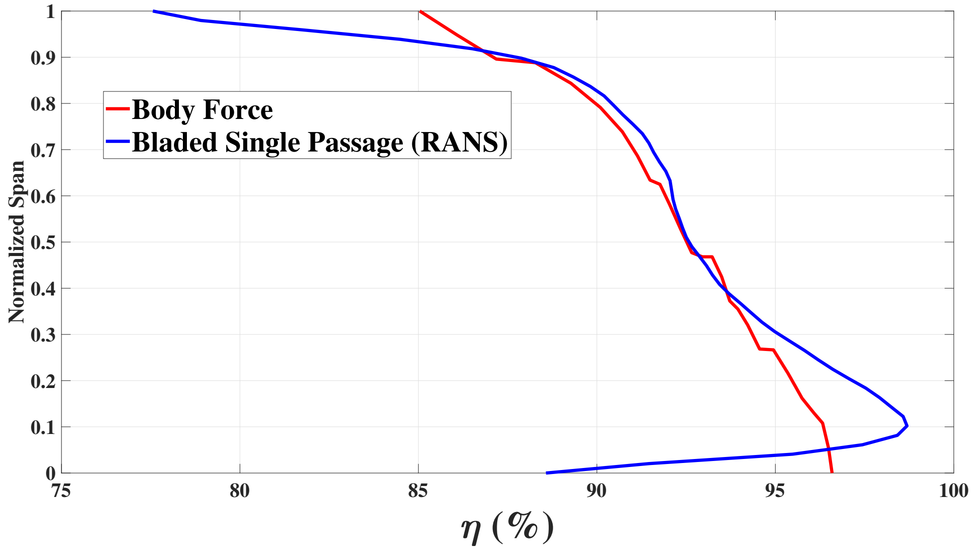


Figure 5-21: Spanwise isentropic efficiency at 60% speed for corrected mass flow rate of 20.3 kg/s ($\hat{\phi} = 1$)

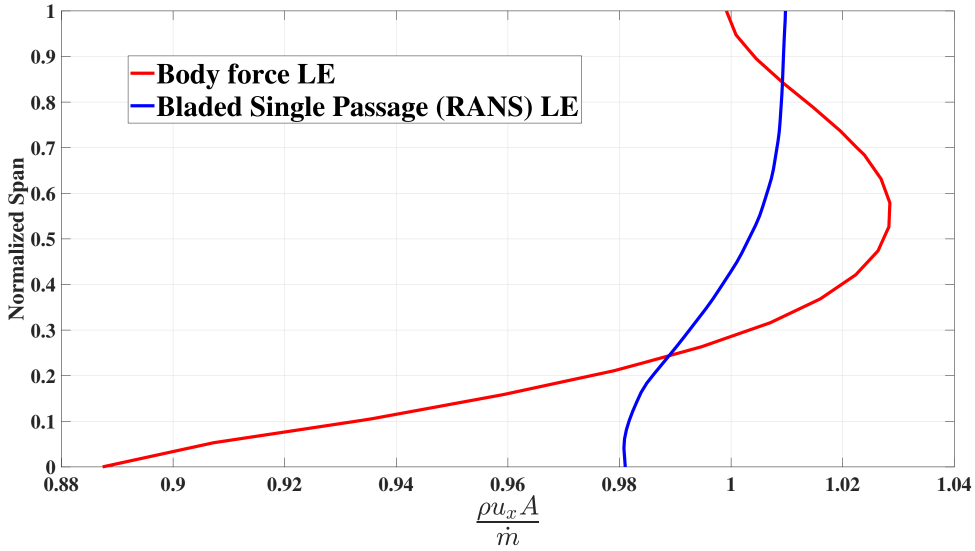


Figure 5-22: Spanwise non-dimensional mass flux at leading edge - 60% speed for corrected mass flow rate of 20.3 kg/s ($\hat{\phi} = 1$)

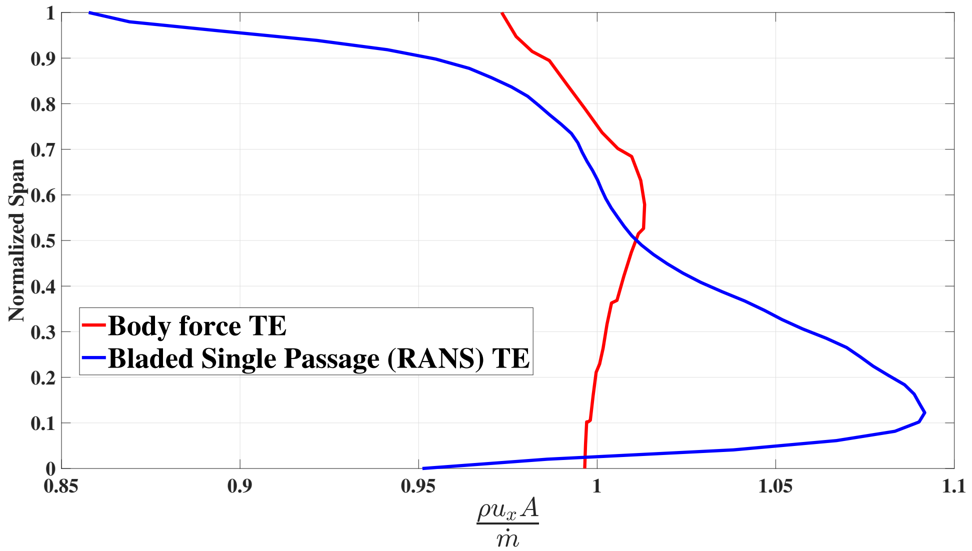


Figure 5-23: Spanwise non-dimensional mass flux at trailing edge edge - 60% speed for corrected mass flow rate of 20.3 kg/s ($\hat{\phi} = 1$)

5.2.2 80% Speed Spanwise Results

In this section the body force modelling at 80% rotational speed with normalized flow coefficient of $\hat{\phi} = 0.93$ (corrected mass flow rate of 24.5 kg/s) is discussed. In Fig. 5-8 it was shown that at this condition, a worst-case of isentropic efficiency is predicted by the body force. To assess the performance, the spanwise pressure ratio is shown in Fig 5-24. Over 40% span, over-prediction occurs and it becomes worse as the span fraction increases. However, the temperature ratio is well captured by the blade loading model as shown in Fig.5-25. Consequently, Fig 5-26. shows that the poor performance in the loss generation predictions occur at the top 20% span. To have a deep analysis of this discrepancy, the spanwise relative Mach number at the leading edge is shown in Fig 5-27. The top 20% span shows that the Mach number at the leading edge lies between 0.9 to 1.1. This is the main cause of the issue in the loss prediction. The reason is that when generating the ANN training data in MISES, at high incidence angles the software suffered from divergence problems for the transonic flow regime. Still, the decreasing trend of the isentropic efficiency from hub to tip is captured by the loss model. Fig. 5-28 shows the spanwise mass flux at the trailing edge for both the body force and the bladed RANS simulations. The high amount of under-prediction of loss in the tip region leads to a higher mass flow rate at that area in the body force compared to the bladed simulations. As it was discussed earlier, the loss affects the local mass flow rate.

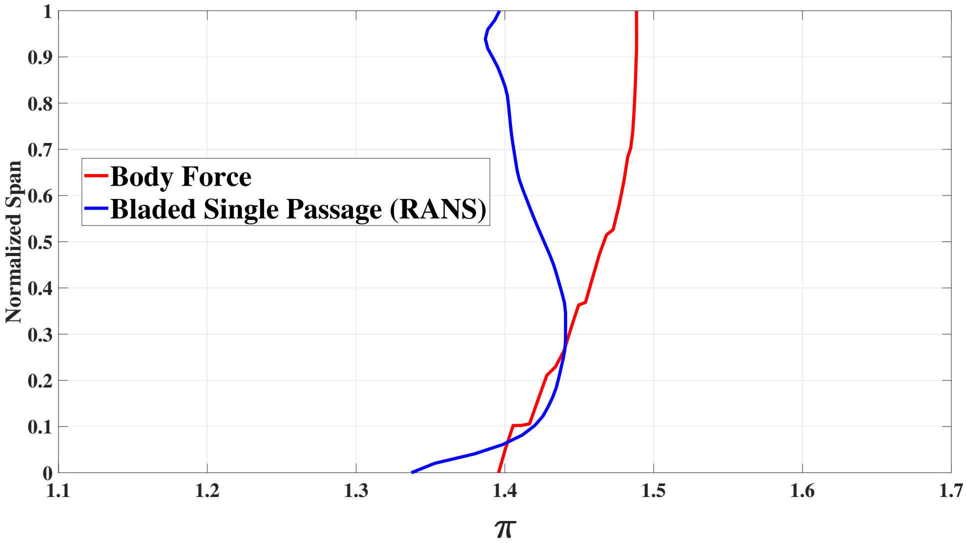


Figure 5-24: Spanwise pressure ratio at 80% speed for corrected mass flow rate of 24.5 kg/s ($\hat{\phi} = 0.93$)

5.2.3 100% Speed Spanwise Results

This section considers 100% rotational speed where the rotor incoming relative Mach number experiences the supersonic flow from 50% span to the tip of the blade. Thus, shock losses play an important role along with the viscous loss in the performance predictions by the body force model. To assess the shock loss and viscous loss, the simulations in the body force were carried out in two ways. Firstly the body force only included the viscous loss and in the second case, the body force included both the viscous and shock losses. Fig 5-29 shows the spanwise pressure ratio for the normalized flow coefficient of $\hat{\phi} = 1.03$ (corrected mass flow rate of 33.5 kg/s). This mass flow rate was the lowest flow rate for the bladed simulations where convergence was reached. The pressure ratio comparison with and without shock loss models shows that a considerable improvement in the pressure ratio prediction is made when

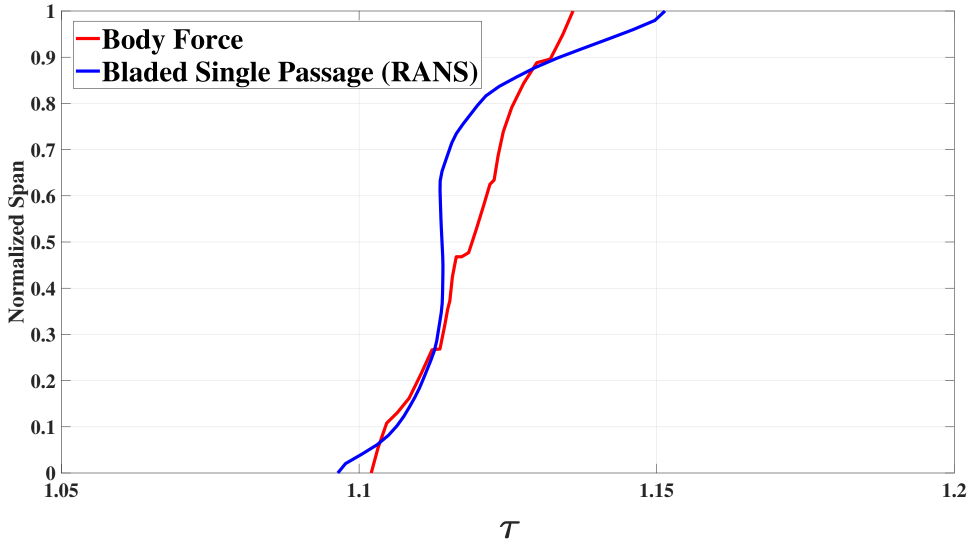


Figure 5-25: Spanwise temperature ratio at 80% speed for corrected mass flow rate of 24.5 kg/s ($\hat{\phi} = 0.93$)

a shock loss is added to the loss model in the body force simulations. It shows that the pressure ratio is well-predicted in the tip region. Figure 5-30 shows the spanwise total temperature ratio at 100% rotational speed. The temperature ratio over the top 30% span shows that a shock has a high effect on the total temperature added to the flow. This difference in the temperature ratio shows the difference of viscous work which adds to the total enthalpy. Nevertheless, the shock loss model has rendered the body force capable of capturing the temperature ratio trend accurately at the tip. Figure 5-31 shows the spanwise isentropic efficiency and highlights the shock loss effects in the entropy generation predictions. As shown, when taking into account the shock loss, the entropy generation is accurately predicted over the top 30% span. This implies that a good shock loss model in the supersonic regime plays an important role in the entropy generation. At the hub, the viscous loss is over-predicted. At mid-span where the flow behaviour is closer to those assumed in the loss model, the accuracy of the loss is good. Figures 5-32 and 5-33 show the spanwise normalized mass fluxes at the leading edge and trailing edge, respectively, for 100% rotational speed. The shock loss has no effect on the incoming mass flux to the rotor, however, at the trailing edge, it is affecting the mass flux and due to the shock loss, a lower mass

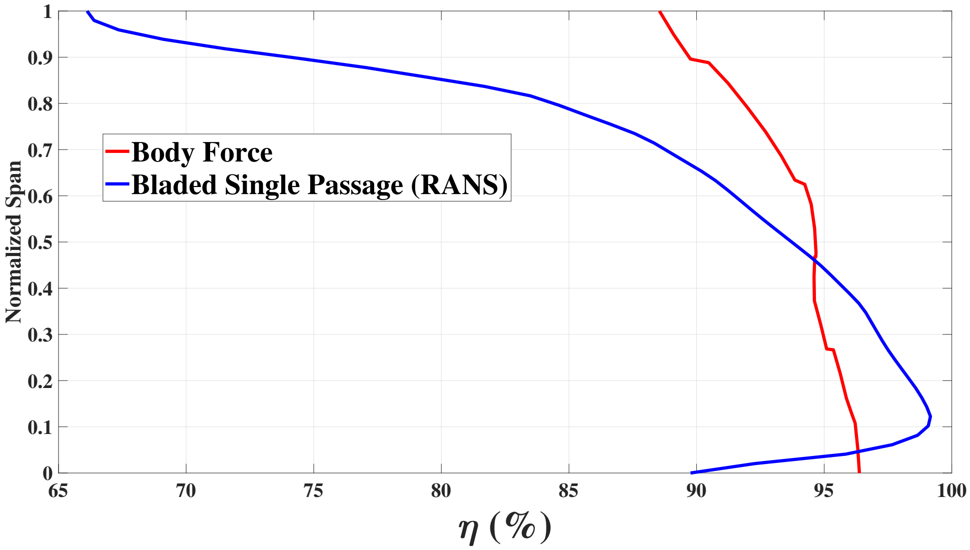


Figure 5-26: Spanwise isentropic efficiency at 80% speed for corrected mass flow rate of 24.5 kg/s ($\hat{\phi} = 0.93$)

flux is passed through the top span region compared to the no-shock model, resulting in good agreement with the bladed mass flux. It also should be mentioned that a viscous loss that reduces the local mass flux, alters the local velocity and consequently the local flow deviation angles are affected. At lower mass fluxes, higher deviation angle end up with the higher loading force by Hall's model and the total input work changes.

To sum up, at all the operating conditions, the body force shows higher suction at mid-span fractions because of the lack of blade metal blockage. The total pressure rise depends on both the flow turning and loss models. At design flow coefficients the viscous loss model is accurate compared to at lower flow coefficients. Violations of the assumptions made for the loss model occur at the hub and tip. The shock model improves the loss prediction with a high impact in the body force.

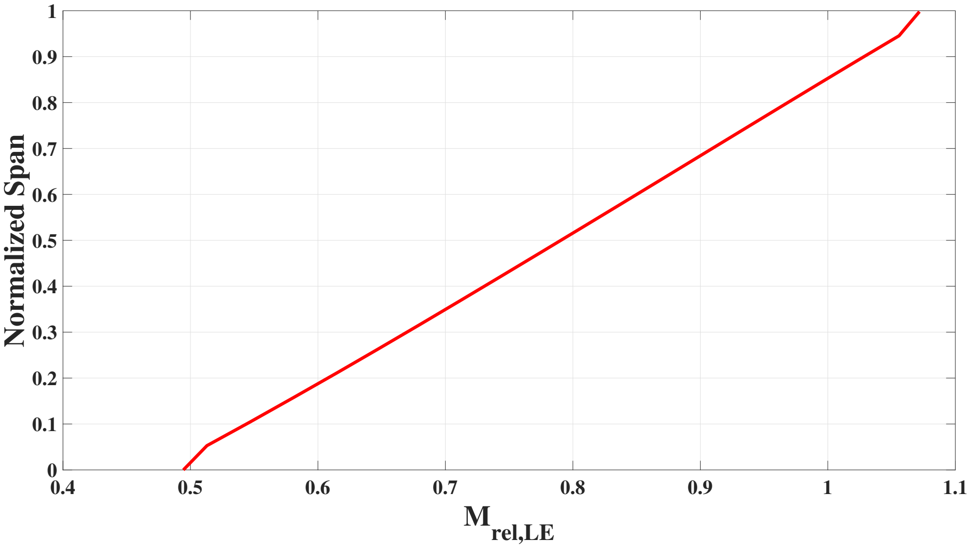


Figure 5-27: Spanwise leading edge relative Mah number at 80% speed for corrected mass flow rate of 24.5 kg/s ($\hat{\phi} = 0.93$) - from body force computation

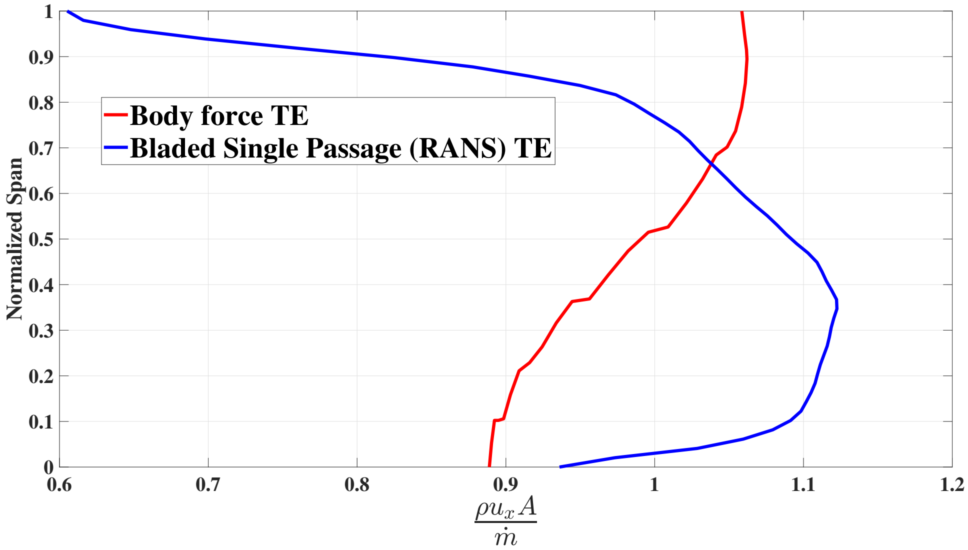


Figure 5-28: Spanwise non-dimensional mass flux at trailing edge edge - 80% speed for corrected mass flow rate of 24.5 kg/s ($\hat{\phi} = 0.93$)

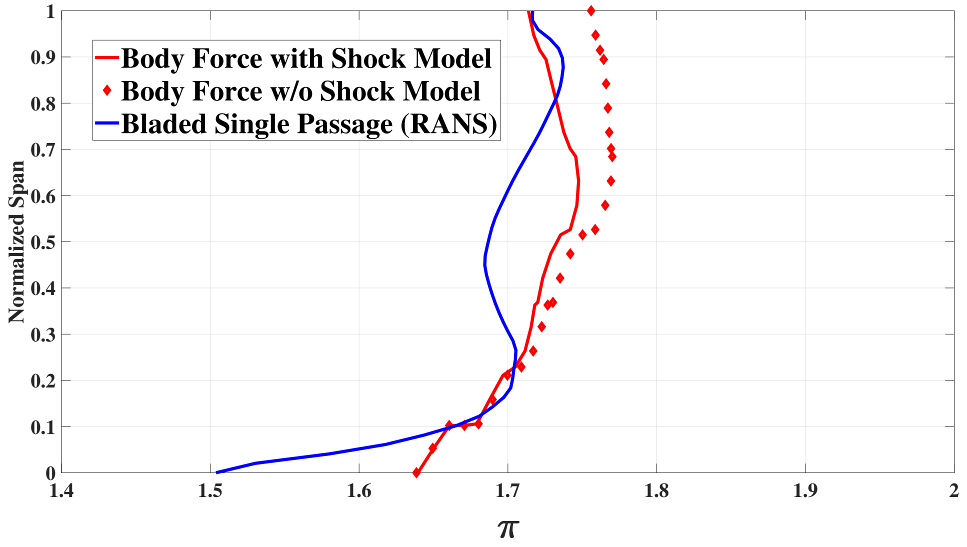


Figure 5-29: Pressure ratio in spanwise at 100% speed for corrected mass flow rate of 33.5 kg/s ($\hat{\phi} = 1.03$)

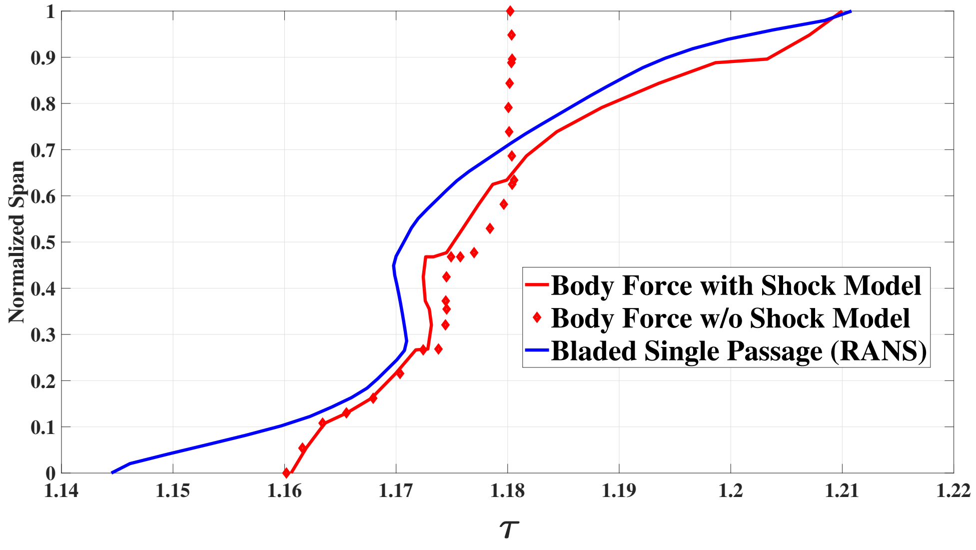


Figure 5-30: Spanwise temperature ratio at 100% speed for corrected mass flow rate of 33.5 kg/s ($\hat{\phi} = 1.03$)

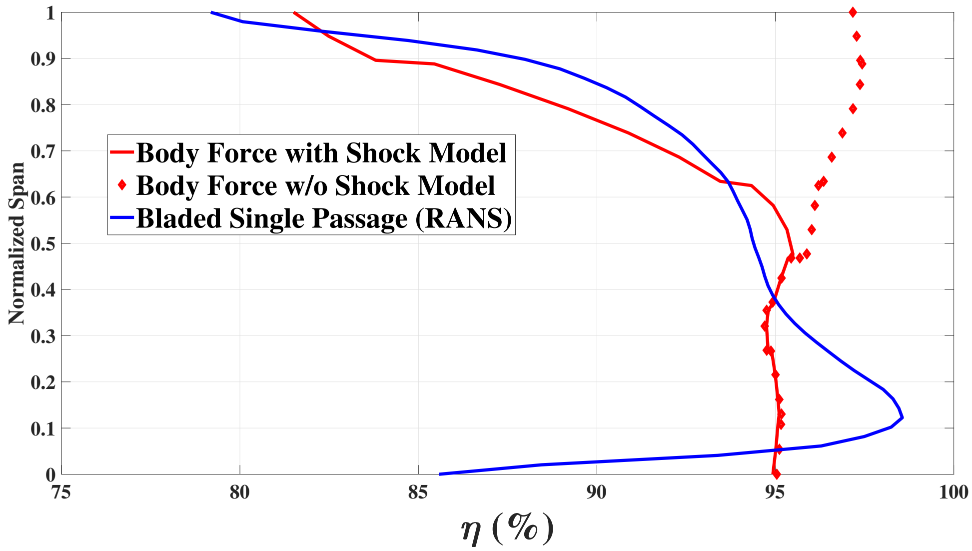


Figure 5-31: Spanwise isentropic efficiency at 100% speed for corrected mass flow rate of 24.5 kg/s ($\hat{\phi} = 0.93$)

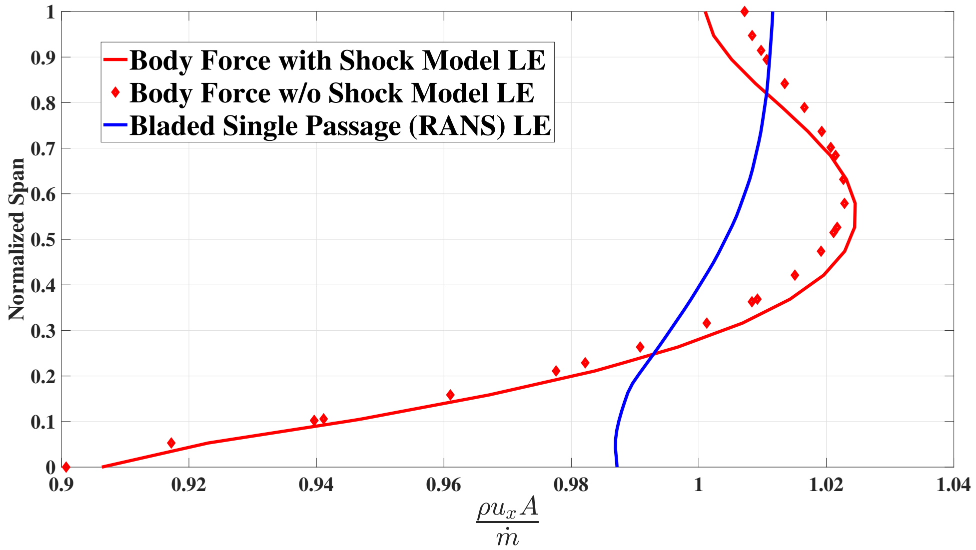


Figure 5-32: Spanwise non-dimensional mass flux at leading edge edge - 100% speed for corrected mass flow rate of 33.5 kg/s ($\hat{\phi} = 1.03$)

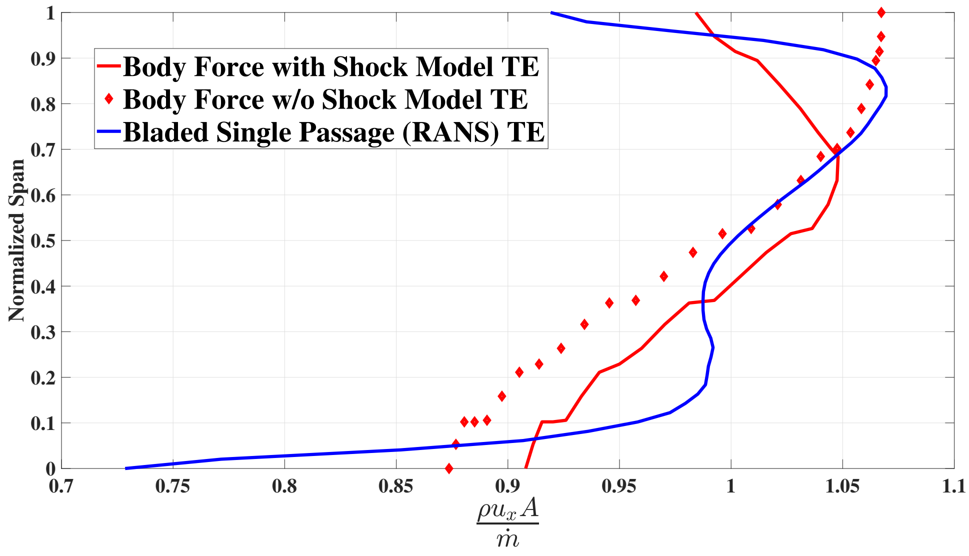


Figure 5-33: Spanwise non-dimensional mass flux at trailing edge edge - 100% speed for corrected mass flow rate of 33.5 kg/s ($\hat{\phi} = 1.03$)

5.3 Non-Uniform Inflow Modelling at 60% Speed

A boundary layer ingestion type of inlet boundary condition has been analyzed for non-uniform flow simulations. The depth of the inlet boundary layer is considered to be half of the inlet diameter, and the velocity profile is quadratic within the boundary layer where $V_{wall} = 0.5V_{max}$. This boundary layer represents a low-speed turbulent boundary layer [45]. Previous work by Defoe et al. [45] shows that Hall's flow turning model works well for such vertically stratified distortions. 60% rotational speed is chosen for which the inlet maximum axial Mach number is around 0.3 at the corrected mass flow rate of 20.5 (kg/s). Due to the low Mach number we assume incompressible flow and calculate the total pressure from the velocity distribution. The body force and bladed computation are carried out at the same corrected mass flow rate, which is 20.5 (kg/s) with flow coefficient of 0.5 ($\hat{\phi} = 1$), close to the design point to prevent operating in unstable conditions. The profile of total pressure and the related contour plot are shown in Fig. 5-34.

U_{mid} is the mid-span blade speed. At the uniform inlet condition a pressure ratio

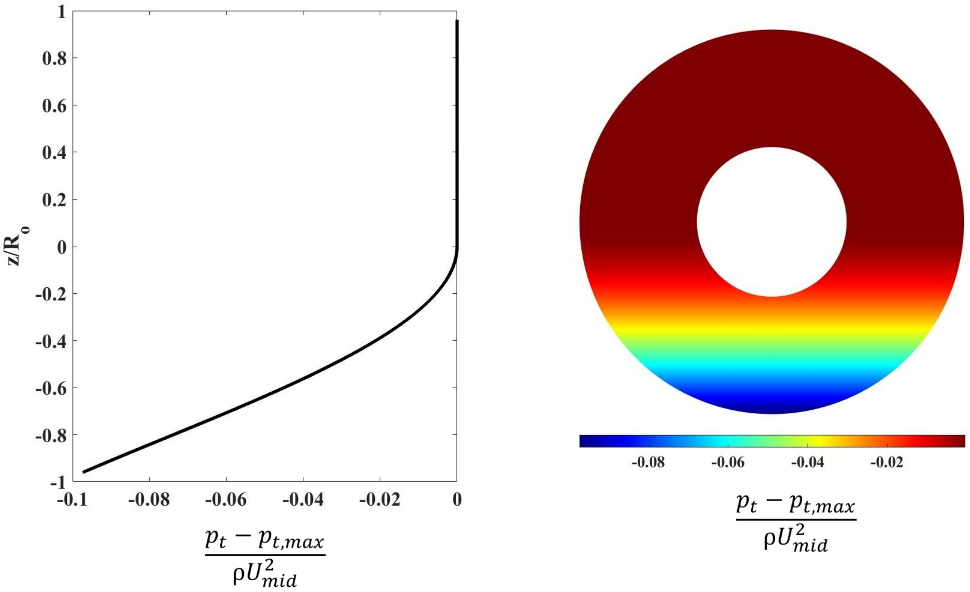


Figure 5-34: Non-uniform inlet boundary condition of total pressure

of 1.21 with the body force and 1.19 with the bladed RANS simulations are predicted. Both the bladed and body force simulations predict the isentropic efficency to be 92%.

The time step for the bladed URANS with non-uniform inflow was $9.45 \times 10^{-6}(s)$. This ensures that in one time step, the rotor rotates one degree and there are thus 16.36 steps per blade passing period. It is worth mentioning that a better resolution occurs if the time step is chosen in a way that it ensures only one cell in circumferential direction is passed in one time step but due to the limited computation resources, we selected one degree instead of one cell. The run time was 15 revolutions. The domain geometry is the same as for the uniform flow cases. The mid-span normalized mass flux upstream of the rotor half a mid-span chord from the rotor leading edge for the body force and bladed URANS computations are shown in Fig. 5-35. The body force shows more suction than URANS at mid-span since the overall mass flux in the undistorted region, is higher.

The contours of the trailing edge total pressure to ambient static pressure ratio $(\frac{(p_t)}{p_\infty})$ $(p_\infty = 101325$ (kPa)) at the exit of the rotor for both body force and bladed URANS are illustrated in Fig. 5-36. The body force captures the trailing edge total pressure well compared to the bladed URANS. The contours of non-dimensional total

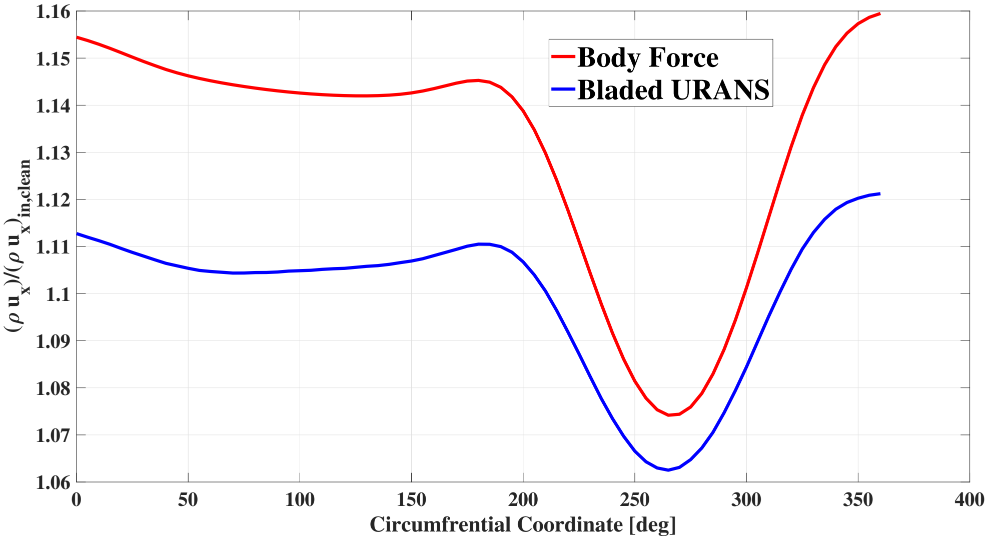


Figure 5-35: Normalized mass fluxes at mid-span upstream of the rotor modelled by body force and bladed URANS

temperature ($\frac{T_t}{T_\infty}$) and at the outlet section of the rotor for both body force and bladed URANS time-averaged results are illustrated in Fig. 5-37. The highest temperature ratio in the body force model is 0.05 more than the bladed case. Hall's loading model does not take into account the displacement thickness effects on the loading and since there are high incidence angles in the incoming flow to the rotor due to the non-uniform inflow, this leads the model to over-predict the work input. Figure 5-38 shows the incidence angles for the body force and the bladed URANS simulations. The body force computation predicts the incidence to be around 4 degrees at the hub while the URANS shows that an incidence of around 6 degrees appears. Both simulations have the same maximum incidence angles in the tip region. The lower incidence angles for the body force at the hub is the main reason that the body force is under-predicting the entropy in that region. The mid-span region experiences the minimum incidence angles.

The contours of non-dimensional entropy ($\frac{(s-s_\infty)T_\infty}{U_{mid}^2}$) at the outlet section of the rotor for both body force and bladed URANS time-averaged results are illustrated in Fig. 5-39. The high entropy region near the tip is qualitatively captured by the body force. However, the current loss model poorly captures the entropy rise near the hub. As it was shown in the uniform simulations, the varying axial velocity at the hub region violates the assumptions made for the viscous loss model. In addition, the displacement thickness variation in the non-uniform flow affects the local mass flux and consequently the local loading. This leads to more error in the predictions of the body force model compared to the bladed URANS. The performance characteristic maps including non-uniform operations are shown in Figs. 5-40 and 5-41 for pressure ratio and isentropic efficiency, respectively. The overall results of the body force and bladed URANS simulations are shown in Table 5.4 and are compared with uniform inflow at the same corrected mass flow rate. The results show that at the same mass flow, the body force model estimates the isentropic efficiency to be 90% for non-uniform flow and 92% for clean inflow. Thus, the body force model predicts that the non-uniform inflow yields a 2% of drop in isentropic efficiency. The bladed RANS yields an isentropic efficiency for uniform flow of 92%, and the URANS shows that the efficiency is 89%, so there is 3% decline in isentropic efficiency for non-uniform flow.

A 2% efficiency penalty due to a BLI distortion is typical of what has been found in the literature [85,86], so the 3% penalty from the URANS is in the agreement with the literature.

The new loss model recognizes the high and low entropy generation regions and predicts a lower efficiency in non-uniform inflow conditions. It performs well in the tip regions, but the hub entropy generation prediction is poor. Blockage effects and different mass flux predictions in addition to the assumptions made for the loss model contribute to the entropy generation shift in the exit plane.

5.4 Concluding Remarks

In this chapter, the loss model was assessed in a 3D rotor case study. There are discrepancies in the spanwise local entropy generation but the increased loss at mid-span is well predicted by the loss model except at 80% speed in the transonic regime. In addition, the shock loss improves the loss predictions significantly. Furthermore, at

60% speed where there are no shock loss, the entropy generation and loss predictions are in good agreement at design. The model captures a reduced pressure ratio in non-uniform inflows compared to the uniform inflow and only a 1% difference in predicted efficiency penalty shows that the model performs greatly in a quantative manner in non-uniform inflow. It is worth noting that the 2% efficiency penalty predicted by the body force is in line with those found for other boundary-layer ingestion fans.

Table 5.4: Comparison of body force and bladed CFD for the same mass corrected flow rate (20.5 kg/s) for uniform and non-uniform inflows

	π	$\eta(\%)$
Body Force (uniform)	1.21	92
Body Force (non-uniform)	1.20	90
RANS (uniform)	1.19	92
URANS (non-uniform)	1.18	89

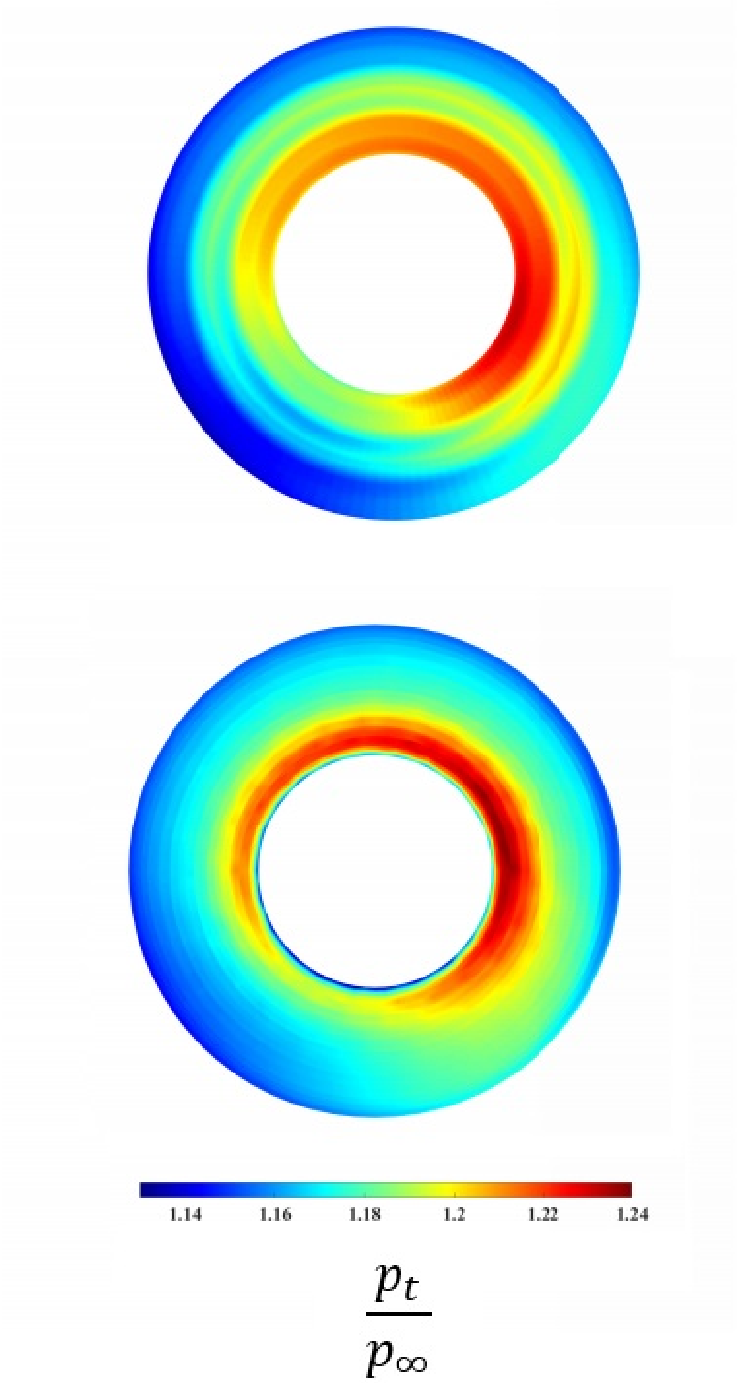


Figure 5-36: Contours of normalized total pressure for body force (upper) and time-averaged URANS (lower) at the rotor trailing edge

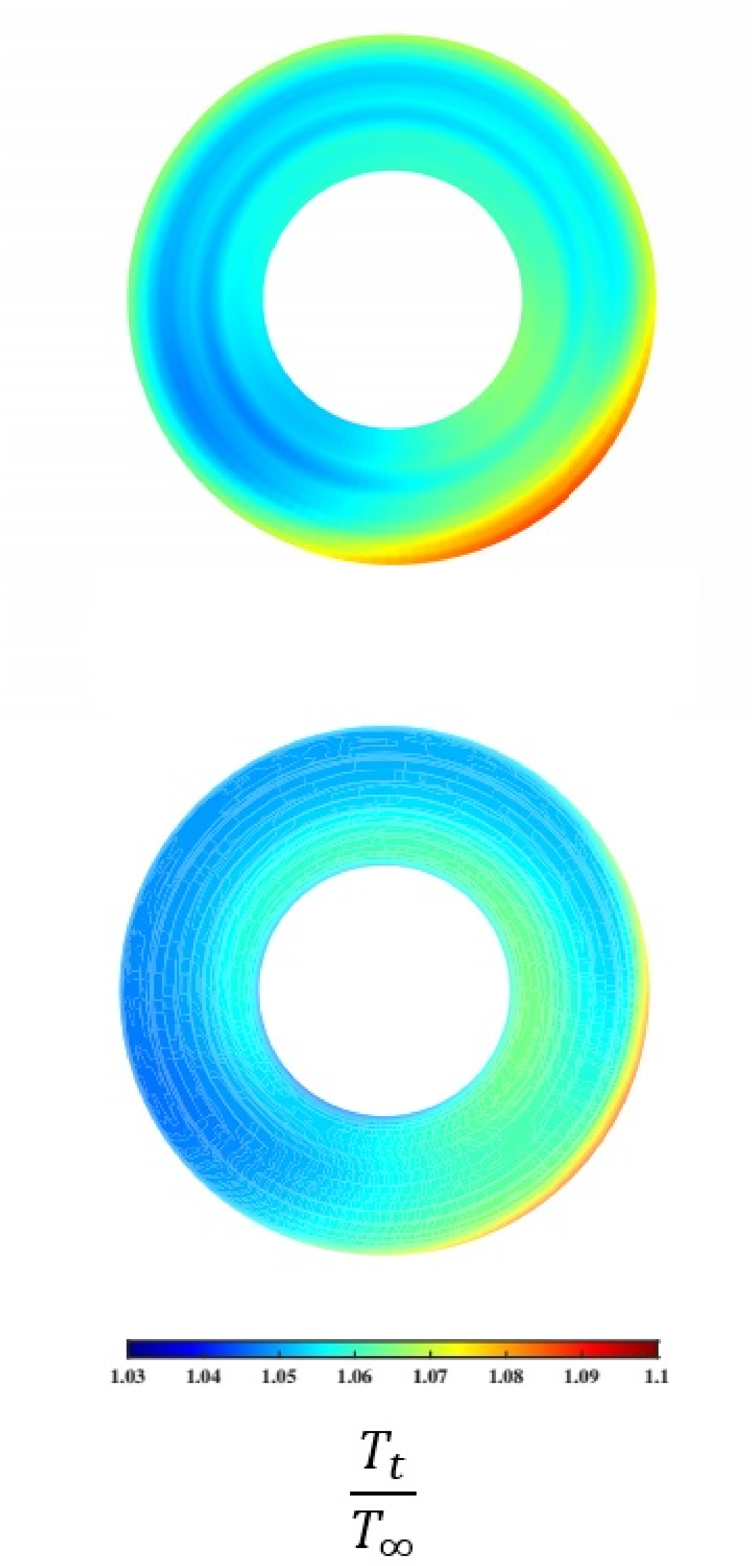


Figure 5-37: Contours of normalized total temperature for body force (upper) and time-averaged URANS (lower) at the trailing edge

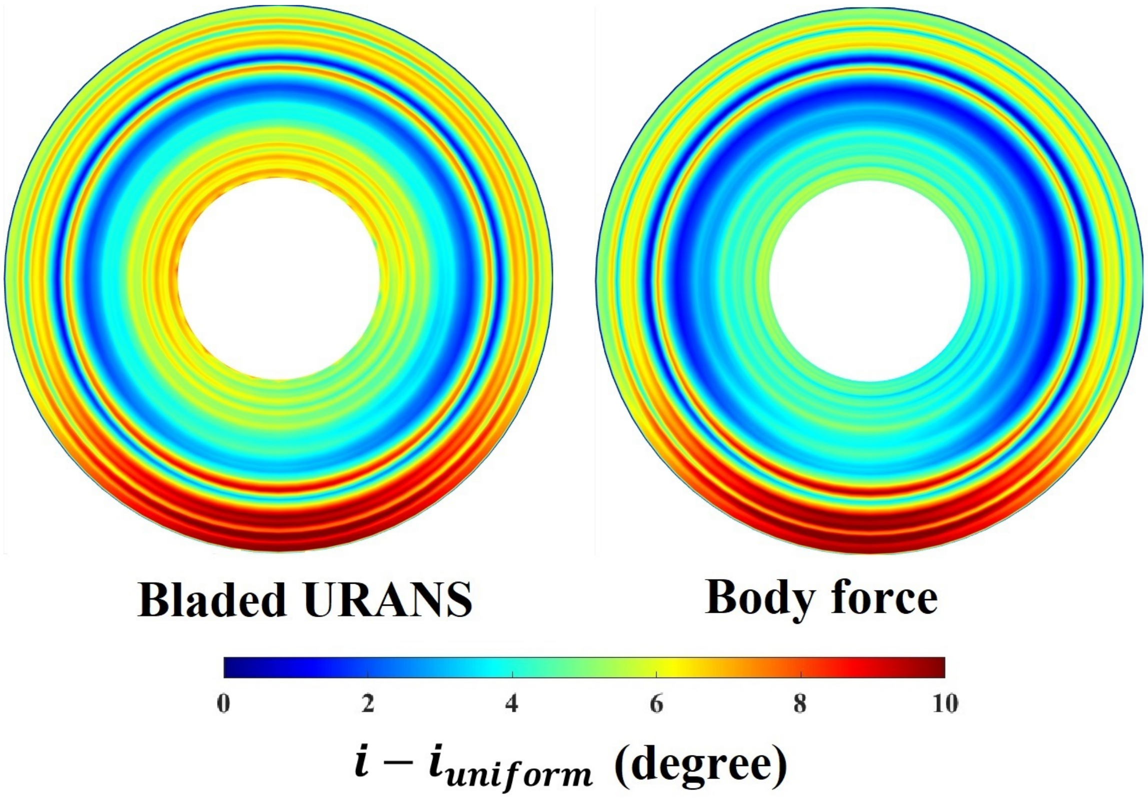


Figure 5-38: Incidence angles for non-uniform inflow in body force and bladed URANS simulations

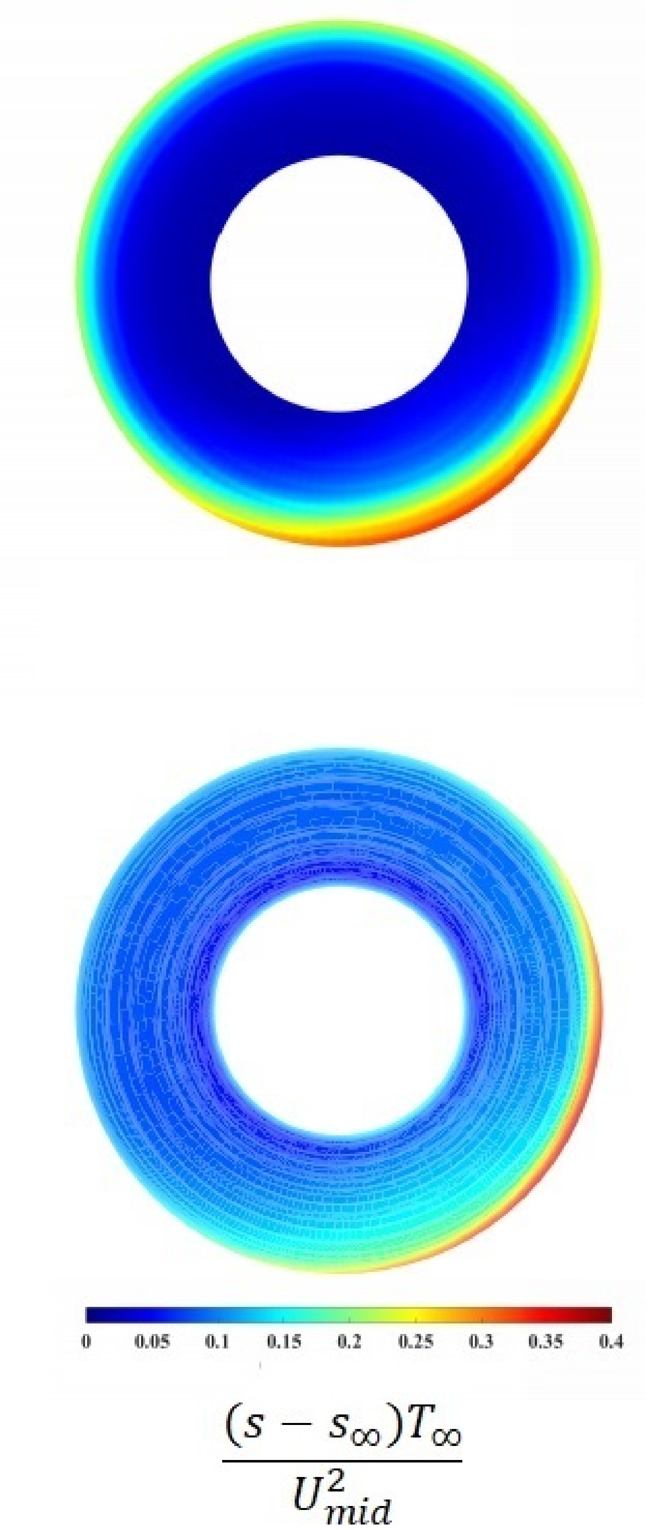


Figure 5-39: Contours of normalized entropy for body force (upper) and time-averaged URANS (lower) at the trailing edge

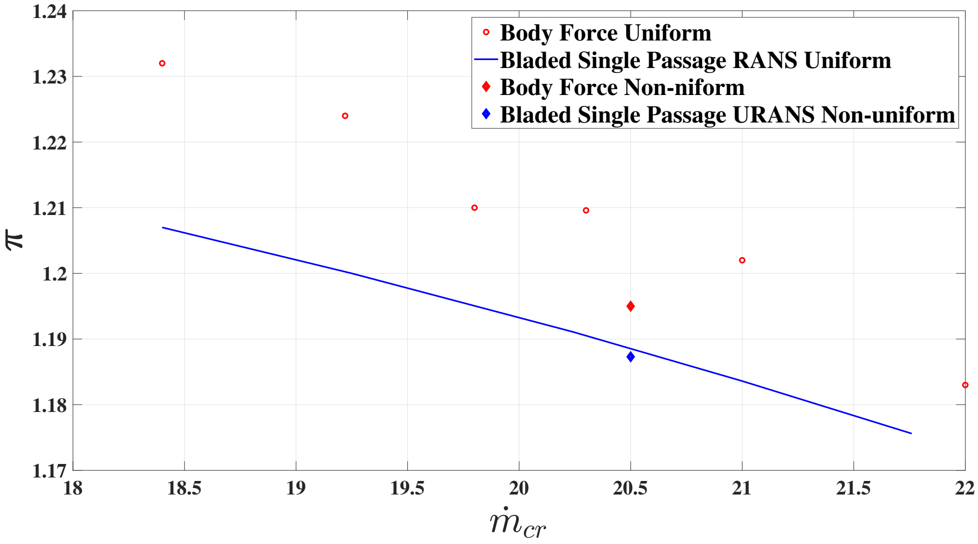


Figure 5-40: Pressure ratio vs corrected mass flow rates at 60% mass flow rate for uniform and non-uniform inflows

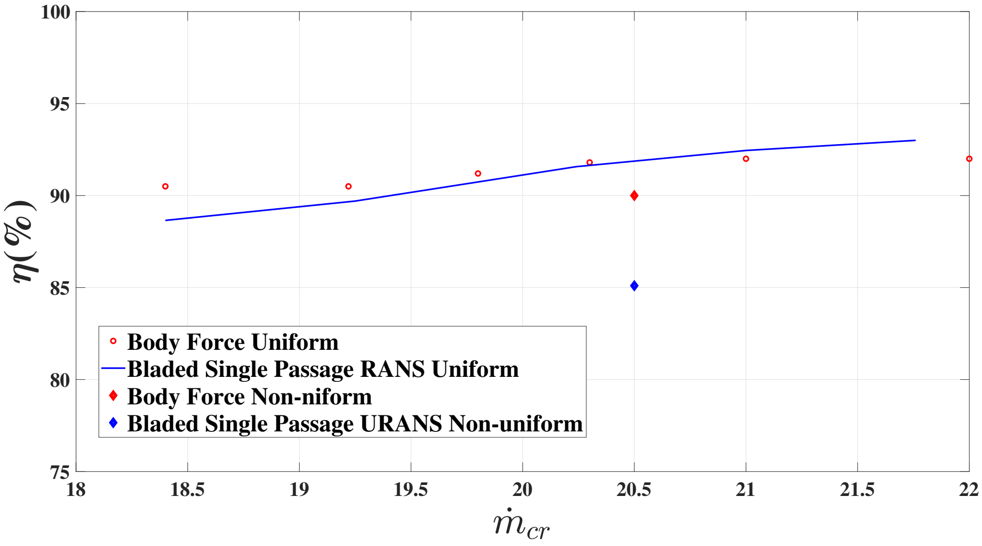


Figure 5-41: Efficiency vs corrected mass flow rates at 60% mass flow rate for uniform and non-uniform inflows

Chapter 6

Summary, Contributions, and Future Work

This dissertation aimed to develop a parallel force model for the body force method without the need for calibration using RANS simulations. Previous loss models either use simplifications without calibration and are not suitable for the separated flow on the blade surface (Benichou et al. [23] showed that loss prediction error in the stator for non-uniform inflow is 37% with the latest no calibration loss model) or require calibration, in which case the computational costs increase. The current study uses the total pressure defect model of Drela and Youngren as the viscous base model and Denton's shock model as the base for shock effects. An artificial-neural-network-based equation yields an analytical model for viscous loss calculations which is implemented in parallel force computations. This chapter gives a summary of the accomplishments in this study, then goes through describing the dissertation contributions. In the end, recommendations for future work are presented.

6.1 Summary

The body force model is a simplified simulation approach that is a promising method for studying new turbofan engine designs facing non-uniform conditions. Body force models can capture the fan/inlet interactions at low computational cost. An accurate no-calibration approach requires the solution of the boundary layer equations along each relative streamline using the local flow quantities. It was shown that the boundary layer solution requires the velocity distribution around the blade. The direct solution of the boundary layer uses an iterative procedure that adds to the cost of the computations. Therefore, a viscous profile loss body force, which depends on the local flow and trailing edge momentum thickness, has been introduced. It was also shown that an indirect way of carrying out boundary layer calculations to obtain the momentum thickness at the trailing edge is an artificial-neural-network-based model. Thus, a new ANN model has been presented using the blade geometry and the flow parameters to predict the trailing edge momentum thickness. In addition, a simple shock loss model was introduced. The model was assessed in 2D cascades and a 3D rotor case study.

6.2 Contributions

The contributions arising from the dissertation are:

1. A new loading model for 2D cascades is introduced that predicts the velocity distribution and flow turning in the body force computations. This model can be used in Denton's based loss generation models where the local velocities for the suction and pressure sides are used along with the dissipation coefficient. It was also shown, however, that the assumption of constant dissipation coefficient can lead to significant under-prediction of loss coefficient even when good agreement of the edge velocities exists. This finding supports the idea that a different approach to computing loss in a body force model is needed, one that does not rely on dissipation coefficient.

2. A new body force model for viscous loss, based on trailing edge momentum thickness, is presented. It is based on Drela and Youngren's approach for loss determination. It was shown that when the trailing edge momentum thickness for 2D cascades is prescribed from MISES, the model does an excellent job of

predicting the loss coefficient.

3. A machine-learning-based model for the trailing edge momentum thickness based on seven geometric and three flow variables was developed and trained on a large dataset of approximately 400,000 cascade computations from MISES. For cascades at constant radius and constant streamtube area, it was shown to be able to reliably predict trailing edge momentum thickness. When applied to a 3D compressor rotor, the model is able to predict loss with reasonable accuracy between 25% and 85% span where streamtube contraction has a low impact on the axial velocities. In the transonic regime the data-driven approach lacks accuracy since the provided data is insufficient. Overall, these results suggest that the machine-learning-based approach is promising, but that training data which includes radius and streamtube area changes would be beneficial to include.

4. A new shock loss model is introduced to the body force simulations, based on Denton's shock loss formulation. Inclusion of this model massively increases the accuracy of the efficiency prediction in regions of supersonic relative flow.

5. The new loss model captures local entropy generation trends correctly for a non-uniform inflow representative of boundary layer ingestion. The model still lacks accuracy at the hub in the non-uniform case due to axial velocity variations which are neglected in the ANN model and also due to the mass flux predition in the body force model that is different with the bladed URANS, but the overall efficiency (and efficiciency penalty) prediction in non-uniform flow for the non-calibrated model can be considered an advancement of the state of the art.

6.3 Future Work

The following additional work is recommended based on the findings of this dissertation:

1. The neural network in this study has used ten independent variables, whereas the real physics has more parameters which affect the loss. The range of incidence angles in the available data is between -6 to 6 degrees, while at off-design conditions, the incidence angles were shown to exceed these bounds. Future studies with a broader range of variables may help upgrade the current analytical model.

2. A sensitivity analysis on the boundary layer calculations is needed to preserve the most dominant variables for ANN input and skip the less sensitive parameters to reduce the complexity of the ANN training while increasing the accuracy.

3. The studies of loss generation around the airfoil showed that the leading edge radius plays a vital role in determining the amplification ratio progress. The sharper leading edges are more susceptible to higher shape factors and separations than blunt leading edges. So, it seems that a sensitivity analysis on the leading edge radius is required to improve the loss model.

Bibliography

[1] Arne Stuermer. DLR TAU-Code uRANS Turbofan Modeling for Aircraft Aerodynamics Investigations. *Aerospace*, 6(11), 2019.

[2] Jakob P. Haug and Reinhard Niehuis. Full annulus simulations of a transonic axial compressor stage with distorted inflow at transonic and subsonic blade tip speed. *International Journal of Turbomachinery, Propulsion and Power*, 3(1), 2018.

[3] N. Thouault, C. Breitsamter, and N.A. Adams. Numerical investigation of inlet distortion on a wing-embedded lift fan. *Journal of Propulsion and Power*, 27(1):16–28, 2011.

[4] Gang Yu, Dong Li, Yue Shu, and Zeyu Zhang. Numerical simulation for engine/airframe interaction effects of the bwb300 on aerodynamic performances. *International Journal of Aerospace Engineering*, 2019, 2019.

[5] E. M. Greitzer. Surge and rotating stall in axial flow compressors part i: Theoretical compression system model. *Journal of Engineering for Gas Turbines and Power*, 98(2):199–211, 1976. ASME.

[6] A. H. Stenning. Inlet distortion effects in axial compressors. *Journal of Fluids Engineering, ASME*, 102(7), March 1980.

[7] E. M. Greitzer F. K. Moore. A theory of post-stall transientsin axial compression systems:part 1—development of equations. *Journal of Engineering for Gas Turbines and Power, ASME*, 108(1):68–76, January 1986.

[8] A. Akturk and C. Camci. Experimental and computational assessment of a ducted-fan rotor flow model. *Journal of Aircraft*, 49(3), June 2012. AIAA.

[9] A.J Al-Daini and J. W. Railly. An actuator-disc model for the prediction of abrupt stall in an axial compressor rotor. *International Journal of Heat and Fluid Flow*, 1(3):99–106, 1979.

[10] W. G. Joo and T. P. Hynes. The simulation ofturbomachinery blade rows in asymmetric flow using actuator disks. *Journal of Turbomachinery*, 119:723–732, October 1997. ASME.

[11] W. R. Hawthorne and J. H. Horlock. Actuator disc theory of the incompressible flow in axial compressors. *Thermodynamics and Fluid Mechanics Group*, 176(1):789–814, 1962.

[12] Y. Gong, C. S. Tan, K. A. Gordon, and E. M. Greitzer. A computational model for short-wavelength stall inception and development in multistage compressors. *Journal of Turbomachinery*, 121(4), October 1999. ASME.

[13] Sho Sato, Nathan Spotts, and Xinfeng Gao. Validation of fan source term model constructed without blade geometry. In *AIAA SciTech Forum*, San Diego, California, January 2019. AIAA.

[14] D. K. Hall, E. M. Greitzer, and C. S. Tan. Analysis of fan stage conceptual design attributes for boundary layer ingestion. *Journal of Turbomachinery*, 139(7), 2017.

[15] Benjamin Godard, Edouard De Jaeghere, and Nicolas Gourdain. Efficient design investigation of a turbofan in distorted inletconditions. In *Proceedings of ASME Turbo Expo 2019*, Phoenix, Arizona, USA, June 2019. ASME. Proceedings of ASME Turbo Expo 2019.

[16] Jin Guo, Baofeng Tu Jun Hu, and Zhiqiang Wang. A mixed-fidelity computational model of aero engine for inlet distortion. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 223(14), 2019.

[17] R. V. Chima. A three-dimensional unsteady cfd model of compressor stability. In *ASME Turbo Expo 2006: Power for Land, Sea and Air*, number GT2006-90040, pages 1157–1168, Barcelona, Spain, May 2006. ASME.

[18] L. Xu. Assessing viscous body forces for unsteady calculations. *Journal of Turbomachinery*, 125(3), 2003. ASME.

[19] P.G. Tucker. Computation of unsteady turbomachinery flows: Part 1—progress and challenges. *Progress in Aerospace Sciences*, 47(7):522–545, October 2011.

[20] Andreas Peters. *Ultra-Short Nacelles for Low-FPR Propulsors.* phdthesis, Massachusetts Institute of Technology, Cambridge, MA, 2013.

[21] D. J. Hill and J. J. Defoe. Innovations in body force modeling of transonic compressor blade rows. *International Journal of Rotating Machinery*, July 2018. Hindawi.

[22] William Thollet. *Body force modeling of fan-airframe interactions.* phdthesis, University of Toulouse, 2017. PhD Thesis.

[23] E. Benichou, G. Dufour, Y. Bousquet, N. Binder, A. Ortolan, and X Carbonneau. Body force modeling of the aerodynamics of a low-speed fan under distorted inflow. *International Journal of Turbomachinery Propulsion and Power*, 4(3), 2019.

[24] Denton. Loss mechanisms in turbomachines. In *Turbo Expo: Power for Land, Sea, and Air*, volume Volume 2: Combustion and Fuels; Oil and Gas Applications; Cycle Innovations; Heat Transfer; Electric Power; Industrial and Cogeneration; Ceramics; Structures and Dynamics; Controls, Diagnostics and Instrumentation; IGTI Scholar Award, Cincinnati, Ohio, May 1993. ASME. V002T14A001.

[25] Mark Drela and Harold Youngren. A user's guide to mises 2.63. MIT Aerospace Computational Design Laboratory, February 2008.

[26] Harold Hayes Youngren. Analysis and design of transonic cascades with splitter vanes. mathesis, Massachusetts Institute of Technology, February 1991.

[27] C.M. Jang, S.M. Choi, and K.Y. Kim. Effects of inflow distortion due to hub cap's shape on the performance of axial flow fan. *Journal of Fluid Science and Technology*, 3(5):598–609, 2008.

[28] D.R. Boldman, D.P. Hwang, M. Larkin, and P. Schweiger. Effect of a rotating propeller on the separation angle of attack and distortion in ducted propeller inlets. In *31st Aerospace Sciences Meeting & Exhibit*. AIAA, 1993.

[29] A. P. Plas, M. A. Sargeant, V. Madani, D. Crichton, E. M. Greitzer, T. P. Hynes, and C. A. Hall. Performance of a boundary layer ingesting (bli) propulsion system. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2007. AIAA.

[30] N. H. Kemp and W. R. Sears. The unsteady forces due to viscous wakes in turbomachines. *Journal of Aeronautical Sciences*, 22(7), July 1955. AIAA.

[31] E. M. Greitzer. Review—axial compressor stall phenomena. *Journal of Fluids Engineering*, 102(2):134–151, 1980. ASME.

[32] R. S. Mazzawy. Multiple segment parallel compressor model for circumferential flow distortion. *Journal of Engineering for Gas Turbine Power*, 99(2):288–296, April 1977. ASME.

[33] C. Hah, D. C. Rabe, T. J. Sullivan, and A. R. Wadia. Effects of inlet distortion on the flow field in a transonic compressor rotor. *Journal of Turbomachinery*, 120(2):233–246, 1998. ASME.

[34] N. A. Cumpsty and J. H. Horlock. Averaging nonuniform flow for a purpose. *Journal of Turbomachinery*, 128(1):120–129, January 2006. ASME.

[35] H. Yeh. An actuator disc analysis of inlet distortion and rotating stall in axial flow turbomachines. *JOURNAL OF THE AEROSPACE SCIENCES*, 1959.

[36] W. R. Hawthorne, N. AMitchell, J.E. McCune, and C.S. Tan. Nonaxisymmetric flow through annular actuator disks: Inlet distortion problem. *Journal of Engineering for Power*, 100(4):603–617, 1978.

[37] S.J. Van Der Spuy, F.N. Le Roux, T.W. Von Backstom, and D.G. Kroger. The simulation of an axial flow fan performance curve at low flow rates. In *ASME Turbo Expo 2011: Turbine Technical Conference and Exposition*, volume 4, pages 425–434. ASME, 2011. GT2011-45709.

[38] F. K. Moore E. M. Greitzer. A theory of post-stall transients in axial compression systems: Part ii—application. *Journal of Engineering for Gas Turbines and Power*, 108(2), 1986. ASME.

[39] B. Godard, E. De Jaeghere, N. Ben Nasr, J. Marty, R. Barrier, and N. Gourdain. Methodologies for turbofan inlet aerodynamics prediction. In *35th AIAA Applied Aerodynamics Conference*. AIAA, 2017.

[40] F. Marble. *Aerodynamics of Turnines and Compressors*, chapter C, pages 83–166. Princeton University Press, 1964.

[41] J. Defoe, A. Narkaj, and Z. S. Spakovszky. A novel mpt noise prediction methodology for highly-integrated propulsion systems with inlet flow distortion. In *15th AIAA/CEAS Aeroacoustics Conference (30th AIAA Aeroacoustics Conference)*, Miami, Florida, May 2009. AIAA.

[42] J. Defoe, A. Narkaj, and Z. Spakovszky. A body-force based methodology for predicting multiple-pure-tone noise: Validation. In *16th AIAA/CEAS Aeroacoustics Conference*, Stockholm, Sweden, June 2010. AIAA.

[43] J. J. Defoe and Z. S. Spakovszky. Effects of boundary-layer ingestion on the aero-acoustics of transonic fan rotors. *Journal of Turbomachinery*, 135(5), 2013. ASME.

[44] A. Peters, Z. S. Spakovszky, W. K. Lord, and B. Rose. Ultrashort nacelles for low fan pressure ratio propulsors. *Journal of Turbomachinery*, 137(2), 2015.

[45] J. J. Defoe, M. Etemadi, and D. K. Hall. Fan performance scaling with inlet distortions. *Journal of Turbomachinery*, 140(7), July 2018. ASME.

[46] William Thollet, Guillaume Dufour, Xavier Carbonneau, and Florian Blanc. Body-force modeling for aerodynamic analysis of air intake – fan interactions. *International Journal of Numerical Methods for Heat and Fluid Flow*, 26(7):2048–2065, 2016.

[47] L. L. de Vega, G. Dufour., and N. G. Rosa. A fully coupled body force-engine performance methodology for boundary layer ingestion. In *AIAA Propulsion and Energy 2019 Forum*, Indianapolis, IN, August 2019. AIAA.

[48] Z.X Liu, H.Z. Diao, X.C. Zhu, and Z.H. Du. Numerical investigation of the axial compressor performance with inlet distortions. *Journal of Aerospace Engineering*, 2017. SAGE.

[49] G. Dufour and W. Thollet. Body force modeling of the aerodynamics of the fan of a turbofan at windmill. In *Proceedings of ASME Turbo Expo 2016: Turbomachinery Technical Conference and Exposition*, 2016.

[50] J. Guo and J. Hu. A three-dimensional computational model for inlet distortion in fan and compressor. *Journal of Power and Energy*, 232(2):144–156, June 2017. SAGE.

[51] J. P. Longley. Calculating stall and surge transients. In *ASME Turbo Expo 2007: Power for Land, Sea, and Air*, pages 125–136, 2007.

[52] M. Righi, V. Pachidis, L. Konozsy, and L. Pawsey. Three-dimensional through-flow modelling of axial flow compressor rotating stall and surge. *Aerospace Science and Technology*, 78:271–279, 2018. Elsevier.

[53] Q. J. Minaker and J. J. Defoe. Prediction of crosswind separation velocity for fan and nacelle systems using body force models: Part 1: Fan body force model generation without detailed stage geometry. *International Journal of Turbomachinery Propulsion and Power*, 2019.

[54] Q. J. Minaker and J. J. Defoe. Prediction of crosswind separation velocity for fan and nacelle systems using body force models: Part 2: Comparison of crosswind separation velocity with and without detailed fan stage geometry. *International Journal of Turbomachinery Propulsion and Power*, 2019.

[55] T. Cao, P. Hield, and P. G. Tucker. Hierarchical immersed boundary method with smeared geometry. *Journal of Propulsion and Power*, 2017. AIAA.

[56] Y. Ma, J. Cui, N. R. Vadlamani, and P. G. Tucker. A mixed-fidelity numerical study for fan-distortion interaction. In *Proceedings of ASME Turbo Expo 2018*, GT2018-75090. ASME, 2018.

[57] H. Schlichting and K. Gersten. *Boundary-Layer Theory*. Springer, 2017.

[58] U. K. Singh. A computation and comparison with measurements of transonic flow in an axial compressor stage with shock and boundary layer interaction. *Journal of Engineering for Gas Turbines and Power*, 104(2):510–515, 1982. ASME.

[59] M. Drela and M. B. Giles. Viscous-inviscid analysis of transonic and low reynolds number airfoils. *Journal of AIAA*, 25(10), October 1987.

[60] M. R. Head. Entrainment in the turbulent boundary layer. Technical Report 3152, Aeronautical research council reports and memoranda, LONDON: HER MAJESTY'S STATIONERY OFFICE - MINISTRY OF AVIATION, September 1958.

[61] S. Pazireh and J. J. Defoe. A no-calibration approach to modelling compressor blade rows with body forces employing artificial neural networks. In *ASME Turbo Expo 2019: Turbomachinery Technical Conference and Exposition*, Phoenix, Arizona, USA, June 2019. ASME.

[62] R. C. Lock and B. R. Williams. Viscous-inviscid interactions in external aerodynamics. *Progress in Aerospace Sciences*, 24(2):51–171, 1987.

[63] A. E. P. Veldman. A simple interaction law for viscous–inviscid interaction. *Journal of Engineering Mathematics*, 65:367–383, 2009.

[64] A. E. P. Veldman. Quasi-simultaneous viscous-inviscid interaction fortransonic airfoil flow. In *4th AIAA Theoretical Fluid Mechanics Meeting*, Toronto, Ontario Canada, 2005. AIAA.

[65] B. J. Powell. The calculation of the pressure distribution on a thick cambered aerofoil at subsonic speeds including the effects of the boundary layer. Technical report, AeronauticalL Research Council, London: Her Majestery'S Stationary Office- Ministry of Technology, June 1967.

[66] P. J. Baddoo and L. J. Ayton. Potential flow through a cascade of aerofoils: direct and inverse problems. *Mathematical, Physical and Engineering Sciences*, 474(2217), September 2018. The Royal Society A.

[67] David Kriesel. A brief introduction to neural networks, 2007.

[68] Zheng Zhao and Huan Liu. Spectral feature selection for supervised and unsupervised learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, page 1151–1157, June 2007.

[69] Stephan Dreiseitl and Lucila Ohno-Machado. Logistic regression and artificial neural network classificationmodels: a methodology review. *Journal of Biomedical Informatics*, 35:352–359, 2002.

[70] D Svozila, V Kvasnicka, and J Pospichal. Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems*, 39:43–62, 1997.

[71] Brendan Tracey, Karthik Duraisamy, and Juan J. Alonso. A machine learning strategy to assist turbulence model development. In *53rd AIAA Aerospace Sciences Meeting*, January 2015.

[72] Jack Weatheritt, R. Sandberg, J. Ling, G. Sáez, and J. Bodart. A comparative study of contrasting machine learning frame work applied to rans modelling of jets in cross flow. In *Proceedings of ASME Turbo Expo 2017: Turbomachinery Technical Conference and Exposition*, 2017.

[73] Jack Weatheritt, Richard Pichler, Richard D. Sandberg, Gregory Laskowski, and Vittorio Michelassi. Machine learning for turbulence model development using a high-fidelity hpt cascade simulation. In *Turbo Expo: Power for Land, Sea, and Air*, volume Volume 2B: Turbomachinery, 2017.

[74] López de Vega Luis, Guillaume Dufour, Florian Blanc, and William Thollet. A machine learning based body force model for analysis of fan-airframe aerodynamic interactions. In *Global Power and Propulsion Society Conference*, 2018.

[75] ANSYS. Ansys fluent- cfd software, release 19.1, 2019.

[76] Y. Dong and N. A. Cumptsy. Compressor blade boundary layers: Part 1—test facility and measurements with no incident wakes. *ASME, Journal of Turbomachinery*, 112(2):222–230, 1990.

[77] Lu Xiaoqiang, Huang Jun, Song Lei, and Li Jing. An improved geometric parameter airfoil parameterization method. *Aerospace Science and Technology*, 78:241–247, July 2018.

[78] J.V. Taylor, B. Conduit, A. Dickens, C. Hall, M. Hillel, and R.J. Miller. Predicting the operability of demanded compressors using machine learning. In *Proceedings of ASME Turbo Expo 2019:Turbomachinery Technical Conference and Exposition GT2019*, Phoenix, Arizona, USA, 2019. ASME.

[79] Anthony J. Strazisar, Jerry R. Wood, Michael D. Hathaway, and Kenneth L. Suder. Laser anemometer measurements in atransonic axial-flow fan rotor. Technical Report 2879, NASA National Aeronautics and Space Administration, 1989.

[80] ANSYS. Ansys cfx- cfd software, release 19.1, 2019.

[81] F Menter. Zonal two equation k-w turbulence models for aerodynamic flows. In *23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference*, Orlando, FL, U.S.A., July 1993. AIAA.

[82] V. Jerez Fidalgo, C. A. Hall, and Y. Colin. A study of fan-distortioninteraction within the nasarotor 67 transonic stage. *ASME Journal of Turbomachinery*, 134(5)(5), September 2012.

[83] ANSYS. Ansys turbogrid - cfd mesh generator software, release 19.1, 2019.

[84] PointWise. Pointwise - cfd mesh generator software, v18.3r1.

[85] E. J. Gunn and C. A. Hall. Aerodynamics of boundary layer ingesting fans. In *ASME Turbo Expo 2014: Turbine Technical Conference and Exposition*, volume Volume 1A: Aircraft Engine; Fans and Blowers of *Turbo Expo: Power for Land, Sea, and Air*, 2014.

[86] D. J. Hill and J. J. Defoe. Scaling of incidence variations with inlet distortion for a transonic axial compressor. *Journal of Turbomachinery*, 142(2), 2020.

Appendix A

Python Code Running MISES

```
"""Module containing function that calls the geometry data and creates
 geometry file as input for MISES
Then, it runs the MISES and creates the output as text files and subsequently
saves the text file in a folder with the name of CreatedDtata
Developed by Syamak Pazireh (Sept 2019)"""


import MisesFunctions as mf
import MISES_InputData as MID
import os
import os.path
import shutil
from numpy import arange
import os
import time


def file_generation (thickness,thickness_position, maximum_camber,
max_camber_position, incidence, Mach, Re, solidity,R_LE,TE_beta):
```

```
# ********************************** Calling MISES input files generator
***********************************************
#Inputs: thickness, maximum camber, max camber position
, incidence, Mach, Re, solidity
ext = MID.InputData(thickness,thickness_position, camber
, max_camber_position,incidence, Mach, Re, solidity,R_LE,TE_beta)
return ext

"""
try:
os.remove("ises.{0}".format(ext))
except:
pass
try:
os.remove("stream.{0}".format(ext))
except:
pass
try:
os.remove("blade.{0}".format(ext))
except:
pass
try:
os.remove("{0}".format(ext))
except:
pass
"""

incidence = 0
Mach = 0.1
```

```
Re = 1000000
thickness = 0.05
max_thickness_position = 0.5
camber = 20
max_camber_position = 0.5
solidity = 1.0
R_LE = 0.01
TE_angle = 5

start_time = time.time()
for incidence in arange(-6,7,1):
for Mach in arange(0.25,1.7,0.2):
for Re in arange(100000,1700000,470000):
for camber in arange(25,50,15):
for max_thickness_position in arange(0.3,0.8,0.1):
for max_camber_position in arange(0.4,0.7,0.1):
for thickness in arange(0.02,0.16,0.025):
for solidity in arange(0.5,2.5,0.5):
for R_LE in arange(0.015,0.021,0.005):
for TE_angle in arange(10,15,5):

#Inputs: thickness,thickness_position, maximum camber, max camber position,
 incidence, Mach, Re, solidity,R_LE,TE_beta
ext = file_generation (thickness,max_thickness_position, camber,
 max_camber_position, incidence, Mach, Re, solidity,R_LE,TE_angle)

# Run ISET to create the mesh
command = 'iset'
try:
    mf.RunIset(command, ext, cwd_in=None, dump=False)
```

```
except:
    pass
    # Run ISES to run the simulations
command = 'ises'
try:
    a = mf.RunIses(command, ext, Nrun=50, cwd_in=None, dump=False)
    # Remove idat file and do not run iplot (if no convergence reached)
    # Otherwise, run iplot and save the output
    if a == 0:
        os.remove("idat.{0}".format(ext))
    else:
        # Run IPLOT to create the output file
        command ='iplot'
        try:
            mf.RunIplot(command, ext, cwd_in=None, dump=False)
            print('iplot terminated')
        except:
            pass
except:
    pass

try:
    # Extract the data from iplot output file
    data = open('{0}.txt'.format(ext), 'r')
    lines = data.readlines()

    SS = []
    PS = []
    XS = []
    ZetaS = []
```

```
UeS = []
ThetaS = []
HS = []
Mach_S = []
XP = []
ZetaP = []
UeP = []
ThetaP = []
HP = []
Mach_P = []

"""
The data are read from MISES outputfile and written to the text file as input
 of artificial neural network platform
"""
# In this section the code splits the suction and pressure side data and puts
 them in arrays
i = 0
SS.append(lines[i+2].split())  # Splilit the data in each line of iplot text
 file to 12 parts
XS.append(SS[i][0])            # Assign first column as x/c of suction side
ZetaS.append(SS[i][2])         # Assign 3rd column as s/c of suction side
 (streamline)
UeS.append(SS[i][4])           # Assign 5th column as Ue/a0 of suction side
 (velocity distribution)
ThetaS.append(SS[i][6])        # Assign 7th column as momentum_thickness/chord
 of suction side
HS.append(SS[i][7])            # Assign 8th column as shape factor parameter of
 suction side
Mach_S.append(SS[i][11])        # Assign 12th column as Mach number of suction
```

```
 side
index = float(XS[i])
while index<1:
    i += 1
    # Save suction side data in lists
    SS.append(lines[i+2].split())  # Splilit the data in each line of iplot

     text file to 12 parts
    XS.append(SS[i][0])            # Assign first column as x/c of suction side
    ZetaS.append(SS[i][2])         # Assign 3rd column as s/c of suction side
     (streamline)
    UeS.append(SS[i][4])           # Assign 5th column as Ue/a0 of suction side
     (velocity distribution)
    ThetaS.append(SS[i][6])        # Assign 7th column as momentum_thickness
    /chord of suction side
    HS.append(SS[i][7])            # Assign 8th column as shape factor
     parameter of suction side
    Mach_S.append(SS[i][11])        # Assign 12th column as Mach number of
     suction side
    index = float(XS[i])

index_s = i+1
i = 2+(i+1)+2-1
j = 0
PS.append(lines[i].split())  # Splilit the data in each line of iplot text file
 to 12 parts
XP.append(PS[j][0])            # Assign first column as x/c of suction side

ZetaP.append(PS[j][2])         # Assign 3rd column as s/c of suction side
 (streamline)
```

```
UeP.append(PS[j][4])           # Assign 5th column as Ue/a0 of suction side
 (velocity distribution)
ThetaP.append(PS[j][6])        # Assign 7th column as momentum_thickness/chord
 of suction side
HP.append(PS[j][7])            # Assign 8th column as shape factor parameter of
 suction side
Mach_P.append(PS[j][11])        # Assign 12th column as Mach number of suction
 side
index = float(XP[j])
while index<1:
    j += 1
    i += 1
    # Save suction side data in lists
    PS.append(lines[i].split())  # Splilit the data in each line of iplot text
     file to 12 parts
    XP.append(PS[j][0])            # Assign first column as x/c of suction side

    ZetaP.append(PS[j][2])         # Assign 3rd column as s/c of suction side
     (streamline)
    UeP.append(PS[j][4])           # Assign 5th column as Ue/a0 of suction side
     (velocity distribution)
    ThetaP.append(PS[j][6])        # Assign 7th column as momentum_thickness
    /chord of suction side
    HP.append(PS[j][7])            # Assign 8th column as shape factor
     parameter of suction side
    Mach_P.append(PS[j][11])        # Assign 12th column as Mach number of
     suction side
    index = float(XP[j])
index_p = j+1
```

```
FS = []
for k in range(0,index_s-1):
    num = (((1 + 0.2*float(Mach_S[k+1])**2)**3.5 * float(UeS[
    k+1])**3*float(ThetaS[k+1])) - \
        ((1 + 0.2*float(Mach_S[k])**2)**3.5 *
         float(UeS[k])**3*float(ThetaS[k]))) / (float(ZetaS[k+1])
         -float(ZetaS[k]))
    FS.append(num)
FS.append(0)

FP = []
for k in range(0,index_p-1):
    num = (((1 + 0.2*float(Mach_P[k+1])**2)**3.5 * float(UeP[
    k+1])**3*float(ThetaP[k+1])) - \
        ((1 + 0.2*float(Mach_P[k])**2)**3.5 *
         float(UeP[k])**3*float(ThetaP[k]))) / (float(ZetaP[k+1])
         -float(ZetaP[k]))
    FP.append(num)
FP.append(0)

# In this section, a text file is oppend and the all nessasary data
are written in it
# thickness,max_thickness_position, camber, max_camber_position, incidence,

 Mach, Re, solidity,R_LE,TE_angle
fid = open('{0}.txt'.format(ext), 'w')
fid.write('%s%12s%15s%7s%7s%10s%9s%9s%9s%13s%3s%4s%6s%14s%12s%12s%30s%10s\n'
%('x','s','Ue/a0','theta','H','Mach'\
        ,'t_max','X_t_max','camber','X_camer_max','i','M0','Re','solidity','L.E.
        radius','T.E.angle','d/ds((1+0.2Me^2)*Ue^3*theta)','[17]/h'))
```

```
fid.write(’\n’)
fid.write(’%95s\n’%(’"Suction Side"’))
fid.write(’\n’)
for i in range(0,index_s):
    fid.write(’%11.8f%11.8f%7.4f%11.8f%7.4f%8.5f%7.4f%10.7f%5d%7.1f%11d%6.2f%9
    d%4.1f%12.3f%9d%25.13f%25.13f\n’\
            %(float(XS[i]),float(ZetaS[i]),float(UeS[i])
            ,float(ThetaS[i]),float(HS[i])
            ,\
            float(Mach_S[i]),thickness,max_thickness_position,camber,
            max_camber_position,incidence
            , Mach, Re, solidity,R_LE,TE_angle,FS[i],FS[i]*solidity))
fid.write(’\n’)
fid.write(’%95s\n’%(’"Pressure Side"’))
fid.write(’\n’)
for i in range(0,index_p):
    fid.write(’%11.8f%11.8f%7.4f%11.8f%7.4f%8.5f%7.4f%10.7f%5d%7.1f%11d%6.2f%9d
    %4.1f%12.3f%9d%25.13f%25.13f\n’\
            %(float(XP[i]),float(ZetaP[i]),float(UeP[i])
            ,float(ThetaP[i]),float(HP[i])
            ,\
            float(Mach_P[i]),thickness,max_thickness_position,camber,
            max_camber_position,incidence,
             Mach, Re, solidity,R_LE,TE_angle,FP[i],FP[i]*solidity))

fid.close()


# Finally move the txt file to the saving folder (all files are saved there)
cwd = os.getcwd()
```

```
    current = "{}".format(cwd) + "/" + ext + ".txt"
    destination = "{}".format(cwd) + "/CreatedData/" + ext + ".txt"
    try:
        shutil.move(current,destination)
    except:
        pass
    current = "{}".format(cwd) + "/" + ’idat.’ + ext
    destination = "{}".format(cwd) + "/DatFiles/" + ’idat.’ + ext
    try:
        shutil.move(current,destination)
    except:
        pass

except:
    pass

try:
    os.remove("ises.{0}".format(ext))
except:
    pass
try:
    os.remove("stream.{0}".format(ext))
except:
    pass
try:
    os.remove("blade.{0}".format(ext))
except:
    pass
try:
    os.remove("{0}".format(ext))
```

```
except:
    pass




execution_time =time.time() - start_time

# path joining version for other paths
cwd = os.getcwd()
DIR = "{}".format(cwd) + "/CreatedData/"
number_of_files = len([name for name in os.listdir(DIR) if
 os.path.isfile(os.path.join(DIR, name))])

# Generates execution time filename
time_filename = 'time for execution of {0} files'.format(number_of_files)

fid = open(time_filename, "w")
fid.write('%s\n'%('execution time'))
fid.write('%15.2f\n'%(execution_time))
fid.write('%s\n'%('number of generated files'))
fid.write('%d'%(number_of_files))
fid.close();

import matplotlib.pyplot as plt
import os
from numpy import arange
import numpy as np
import math
from math import pi, atan, sin, cos, sqrt, pow, tan, asin
```

```
def
 airfoil(max_thickness,max_thickness_position,
 camber,max_camber_position,R,TE_angle):

    """
    The geometry of space parameterized airfoil is generated
    This airfoil section is used in compressor and Specifications
    Developed by Syamak Pazireh (Oct 2019)
    """

    t_max = max_thickness                           # t_max/chord
    t_max_position = max_thickness_position         # t_max_position/chord
    p = max_camber_position
    # max_camber_position/chord
    R_LE = R                                        # Fraction of LE radius over chord
    beta = TE_angle                                 # Trailing edge boat-tail angle
    #m = max_camber                                  # max_camber/chord
    d = tan(camber*pi/180)
    m = (sqrt(1-4*d**2*(p-1)*p)-1)/(4*d)

    y_t = []
    y_c = []
    THETA = []
    dyc_dx = []
    CLx = []
    CLy = []

    NumberOfNodes = 200
```

```
x = arange(0,1+(1./NumberOfNodes),(1./NumberOfNodes))
CLx = x

i = 0
# Calculation of local y_c (camberline) NACA 4 digit camberline
for X in x:
    if X<p:
        y_c.append(m/p**2*(2*p*X-X**2))
        # equation of camber line
        dyc_dx.append(2*m/p**2*(p-X))
        # slope of camberline
    else:
        y_c.append(m/(1-p)**2*((1-2*p)+2*p*X-X**2))
        # equation of camber line
        dyc_dx.append(2*m/(1-p)**2*(p-X))
        # slope of camberline
    THETA.append(atan(dyc_dx[i]))
    i += 1

CLy = y_c

XT = t_max_position

# ref paper is : "An improved geometry parameter airfoil parameterization
 method" Lu Xiaoqiang et. al. 2018 Journal of Aerospace science and technology

C = np.array([[XT**0.5,XT**1,XT**2,XT**3,XT**4],[0.5*XT**-0.5,1,2*XT
,3*XT**2,4*XT**3],\
             [0.25,0.5,1,1.5,2],[1,0,0,0,0],[1,1,1,1,1]])
B = np.array([t_max/2,0,-tan(beta/2*pi/180),sqrt(2*R_LE),0])
```

```
    A = np.dot(np.linalg.inv(C),B)



    y_t = []
    xtop = []
    ytop = []
    xbot = []
    ybot = []

    for i in range(0,len(CLx)):
        y_t.append(A[0]*CLx[i]**0.5+A[1]*CLx[i]**1+A[2]*CLx[i]**2+
        A[3]*CLx[i]**3+A[4]*CLx[i]**4)
        xtop.append(CLx[i] + (y_c[i])*sin(THETA[i]))
        ytop.append(CLy[i] + (y_t[i])*cos(THETA[i]))
        xbot.append(CLx[i] - (y_c[i])*sin(THETA[i]))
        ybot.append(CLy[i] - (y_t[i])*cos(THETA[i]))

    return x,y_c,ybot,ytop,y_t,xtop,xbot,dyc_dx



"""
(Xc,Yc,YL,Yu,Y,Xu,XL,dyc_dx) = airfoil(0.1,0.4,0.1,0.4,0.01,15)
 #(max_thickness,max_thickness_position,camber,max_camber_position,R,,TE_angle)
plt.plot(Xc, Yc,'r',Xu,Yu,'b',XL,YL,'m')
plt.xlabel('x/c')
plt.ylabel('y/c')
plt.title('airfoil')
plt.xlim(0, 1)
plt.ylim(-0.5, 0.5)
plt.grid(True)
```

```
plt.show()
"""

"""Module containing function definitions for generating
input geometry and solver
conditions for MISES.
Developed by Syamak Pazireh (Sept 2019)"""

import os
import os.path
import math
from math import pi, atan, sin, cos, sqrt, pow, tan
import airfoil_geometry_generator as agg
import matplotlib.pyplot as plt
import numpy as np


def InputData(thickness, thickness_position, camber, max_camber_position,
 incidence, Mach, Re, solidity,LE_Radius,TE_beta):

    print ('Generating input data for MISES')
    # **************** NACA 4 digit camber with C4 profile compressor
     ******************************
    t = thickness              # Maximum thickness (percentage of the chord)

    # Set Flow Angles For Analysis

    zeroincidance = 0 # degrees
    minangle = 0      # degrees
    maxangle = 0      # degrees
```

```
intervangle = 0   # delta degrees



# ----------------------Stream Tube Specifications
-----------------------

Xstreamstart = -20 # X coordinate of the streamtube at start.
Xstreamend = 20    # X coordinate of the streamtube at the end.
radiusstart = 1    #  m' -Radius at X coordinate of the streamtube at start.
radiusend = 1      # m' -Radius at X coordinate of the streamtube at the end.
Thicknessstart = 1 # Thickness at X coordinate of the streamtube at start.
Thicknessend = 1   # Thickness at X coordinate of the streamtube at the end.
r=(radiusstart+radiusend)/2

# Set ------------------------------Geometry
 Parameters------------------------------------
chord = 1 # m'
Lref = 1



# --------------------------------Set Up Grid
 Parameters---------------------------------

Xinletgrid = 2 * chord
# Distance from the leading edge. Usually 2 times the chord (Lref/rmean) is ok.
Xoutletgrid = 2 * chord
# Distance from the trealing edge. Usually 2 times the chord (Lref/rmean) is ok.

# ---------------------------SET UP AIRFOIL DATA NAME
 -----------------------------------
```

```
name = 'airfoil'


# ++++++++++++++++++++++++ calling data generator
+++++++++++++++++++++++++++++++++++++++++++++++++++++
# ================================== Define blade - Shape Space
 =======================================

(Xc,Yc,YL,Yu,Y,Xu,XL,dyc_dx) =
 agg.airfoil(thickness,thickness_position,camber,max_camber_position
 ,LE_Radius,TE_beta)
# Camber Line

CLx = Xc
CLy = Yc

xtop = []
ytop = []
xbot = []
ybot = []

for i in range(1,len(CLx)):
    xtop.append(Xc[i])
    ytop.append(Yu[i])

for i in range(0,len(CLx)):
    xbot.append(Xc[i])
    ybot.append(YL[i])
```

```
# Assemble data required for MISES blade file
X = xtop[::-1] + xbot
Y = ytop[::-1] + ybot
XCFX = xbot[::-1] + xtop
YCFX = ybot[::-1] + ytop
ZCFX = XCFX
ThetaCFX = YCFX
CFX = ZCFX + ThetaCFX

Xdim = X
Ydim = Y

# round off trailing edge
Xdim1=Xdim*10;
Ydim1=Ydim*10;

# Plot Blade
"""
plt.plot(Xc, Yc,’r’,Xc,YL,’b’,Xc,Yu,’b’)
plt.xlabel(’x/c’)
plt.ylabel(’y/c’)
plt.title(’airfoil’)
plt.xlim(0, 1)
plt.ylim(-0.5, 0.5)
plt.grid(True)
plt.show()
"""

#   ============================== INPUT DATA FOR EXECUTABALE MISES FILES
 ================================
```

```
# Blade File

LE_CamberLine_angle = atan(dyc_dx[0])
LE_CamberLine_angle = LE_CamberLine_angle * 180/pi

sinl = dyc_dx[0]
sout = dyc_dx[-1]
"""sinl = 0
sout = 0"""
chinl = Xinletgrid
chout = Xoutletgrid
pitch = 1/solidity  # non-dimensinal pitch with respect to chord which is
 pitch/chord

AOA = incidence + LE_CamberLine_angle
AOA = AOA*pi/180

# Case name
name_of_file = "i-{0}-M-{1}-Re-{2}-mt-{3}-mtp-{4}-c-{5}-mcp-{6}-s-{7}-
R_LE-{8}-TE-{9}".format(\
incidence,Mach,Re/1000000.0,thickness,thickness_position,camber,
 max_camber_position,solidity,LE_Radius,TE_beta)

blade_filename = 'blade.' + name_of_file



# opens BLADE.* file
fid = open(blade_filename, "w")
```

```
#writes title/ header and flow angles, blade angles and pitch
and current blade
 geometry coordinates
fid.write('%s\n'%(name_of_file))
fid.write('%12.8f%12.8f%12.8f%12.8f%12.8f\n'%(sinl,sout,chinl,chout,pitch))
z = [Xdim,Ydim]
zp = np.transpose(z)

for i in range(0,len(Xdim)):
    fid.write("%12.8f %12.8f\n" %(zp[i][0],zp[i][1]))

# Closes BLADE.* file
fid.close()

# ============================ Stream File
 ==============================================

Wheelspeed = -0
streamstart = Xstreamstart
streamend = Xstreamend
nondimradiusstart =radiusstart
nondimradiusend = radiusend
bstart = Thicknessstart
bend = Thicknessend



# Write out stream file

# Generates full output filename
stream_filename = 'stream.' + name_of_file
```

```
# Opens STREAM.* file
fid = open(stream_filename, "w")

fid.write('%12.8f\n'%Wheelspeed)
fid.write('%12.8f %12.8f %12.8f\n'%(streamstart,nondimradiusstart,bstart))
fid.write('%12.8f %12.8f %12.8f\n'%(streamend,nondimradiusend,bend))

# Closes STREAM.* file
fid.close();

# ======================== ISES File
 ===========================================
Mach_in = Mach
Mach_out = Mach
p2_p01 = 1/(1 + 0.2 * Mach_out**2)**3.5
p1_p01 = 1/(1 + 0.2 * Mach_in**2)**3.5
Globvars = [1,2,5,15,6]     # | grid exit slope | LE stag. pt.
Constraits = [1,4,3,16,6]   # | set LE Kutta | set TE Kutta
"""Globvars = [1,2,5]     # | grid exit slope | LE stag. pt.
Constraits = [1,4,3]  # | set LE Kutta | set TE Kutta """
p1divpo1 = p1_p01      # ignored
s1 = tan(AOA)          # inlet flow slope/direction
mp1 = -1.0 * chord     # of Lrefs (=chord) uptsream of LE that BC is imposed
Mach_out = 0.0         # ignored
p2divpo1 = p2_p01      # outlet pressure
s2 = 0                 # ignored
mp2 = 1.5 * chord      # of Lrefs (=chord) downstream of TE that BC is imposed
msplitfrac = 0.0       # ignored
hwalldivhoa = 0.0      # adiabatic
```

```
turbpercent = 9       # negative tells MISES it’s a %turb and not Ncrit value
ISMOM = 4             # type of isentropy/momentum handling
 (generally do not change)

if Mach<1:
    Mcrit = 0.98
else:
    Mcrit = 0.85



if Mach<0.6:
    tranloc1 = 1.02       # imposed turbulence transition on side
    1 (>1 means none)
    tranloc2 = 1.02       # imposed turbulence transition on side
    2 (>1 means none)
    Coeff_artificialdissipation = 1.0
else:
    tranloc1 = 0.04       # imposed turbulence transition on
    side 1 (>1 means none)
    tranloc2 = 0.04       # imposed turbulence transition on
    side 2 (>1 means none)
    Coeff_artificialdissipation = -1.0 # negative value disables the second
     order dissipation




# Write out ISES case definition file
# Generates full output filename

ises_filename = ’ises.’ + name_of_file
```

```
# Opens ises.* file
fid = open(ises_filename, "w")

fid.write('%d %d %d %d %d\n'
%(Globvars[0],Globvars[1],Globvars[2],Globvars[3],Globvars[4]))
fid.write('%d %d %d %d %d\n'
%(Constraits[0],Constraits[1],Constraits[2],Constraits[3],Constraits[4]))
fid.write('%12.8f %12.8f %12.8f %12.8f\n'%(0,p1divpo1,s1,mp1))
fid.write('%12.8f %12.8f %12.8f %12.8f\n'%(Mach_out,p2divpo1,s2,mp2))
fid.write('%12.8f %12.8f\n'%(msplitfrac,hwalldivhoa))
fid.write('%12.8f %12.8f\n'%(Re,turbpercent))
fid.write('%12.8f %12.8f\n'%(tranloc1,tranloc2))
fid.write('%d %12.8f %12.8f\n'%(ISMOM,Mcrit,Coeff_artificialdissipation))



"""
fid.write('%d %d %d\n'%(Globvars[0],Globvars[1],Globvars[2]))
fid.write('%d %d %d\n'%(Constraits[0],Constraits[1],Constraits[2]))
fid.write('%12.8f %12.8f %12.8f %12.8f\n'%(Mach_in,p1divpo1,s1,mp1))
fid.write('%12.8f %12.8f %12.8f %12.8f\n'%(Mach_out,p2divpo1,s2,mp2))
fid.write('%12.8f %12.8f\n'%(msplitfrac,hwalldivhoa))
fid.write('%12.8f %12.8f\n'%(Re,turbpercent))
fid.write('%12.8f %12.8f\n'%(tranloc1,tranloc2))
fid.write('%d %12.8f %12.8f\n'%(ISMOM,Mcrit,Coeff_artificialdissipation))
"""

# Closes ISES.* file
fid.close();
```

```
    fid = open(name_of_file, "w")
    for i in range(0,len(CFX)):
        fid.write("%f," %(CFX[i]))
    fid.close()

    return name_of_file

"""Module containing function definitions for running Mises
To use, add "import MisesFunctions as mf" to your python script.
To call, use (for example) mf.RunIset(...)

Originally by Tim Houghton
Modified/updated by Nishad Sohoni (Oct 2012 to Feb 2013)
Developed for automated big data generation by Syamak Pazireh (Sept 2019)"""

import subprocess
import os
ospj = os.path.join
import warnings
import shutil

class Error(Exception):
    """Base class for MisesFunctions exceptions."""
    pass

class IsetError(Error):
    """Exception raised in RunIset."""
```

```
    pass


class IsesError(Error):
    """Exception raised in RunIses."""
    pass



def _dump_func(prog, prog_out):
    """Internal function to write dump file."""
    with open('{0}_dump.txt'.format(prog), 'w') as fout:
        fout.write(prog_out)


def RunIset(command, ext, cwd_in=None, dump=False):
    """Function to run iset. Inputs:
    command:    Command to run ISET.
                Useful if the location of 'iset' is not in the
                system path, or if you have named it something
                other than 'iset'.
    ext:        File extention for MISES imputs.
    cwd_in:     Directory in which the subprocess runs.
                Use None as the default value.
    dump:       Will write ISET stdout to a dump file."""

    print ("Running ISET...")
    # Run ISET in a subprocess.
    iset_proc = subprocess.Popen([command, ext], cwd=cwd_in,
                            stdin=subprocess.PIPE, stdout=subprocess.PIPE)
    # Supply sequence of parameters to ISET
    # "\n" is the same as hitting "Return"
    iset_out = iset_proc.communicate("2\n\n3\n4\n0\n")[0].splitlines()
```

```
    #iset_out = iset_proc.communicate("1\n\n2\n\n3\n4\n0\n")[0]
    # Check output
    for line in iset_out:
        if "Grid not initialized" in line or "grid not generated" in line:
            if dump:
                # Write output to dump file and exit
                _dump_func("_".join(["iset",ext]), iset_out)
            raise IsetError("{0}: Grid not initialized!".format(ext))
    else:
        for line in iset_out:
            if "Number of streamlines" in line:
                print ("{0}: Grid check probably ok.".format(ext))
    return

def RunIses(command, ext, Nrun, cwd_in=None, dump=False):
    """Function to run ises. Inputs:
    command:    Command to run ISES.
                Useful if the location of ’ises’ is not in the
                system path, or if you have named it something
                other than ’ises’
    ext:        File extention for MISES imputs
    Nrun:       Number of iterations.
                This script will run ISES for Nrun iterations,
                then quit, whether or not ISES has converged.
    cwd_in:     Directory in which the subprocess runs.
                Use None as the default value.
    dump:       Will write ISES stdout to a dump file."""

    print (’Running case {0} in ISES’.format(ext))
    # Run ISES in a subprocess.
```

```
    ises_proc = subprocess.Popen([command, ext], cwd=cwd_in,
                                stdin=subprocess.PIPE, stdout=subprocess.PIPE)
    # Supply sequence of parameters to ISES
    # '\n' is the same as hitting 'Return'
    ises_out = ises_proc.communicate('{0}\n0\n'.format(Nrun))[0]
    # Check output
    if "Converged on tolerance" in ises_out.splitlines()[-3]:
        print ('Convergence check ok.')
        a = 1
    else:
        print ('No convergence reached.')
        a = 0
        """
        if dump:
            # Write output to dump file and exit
            _dump_func('_'.join(['ises',ext]), ises_out)
        raise IsesError("Convergence check failed!")
        """
    return a


def RunIplot(command, ext, cwd_in=None, dump=False):
    """Function to save data from iplot to a text file.
    Inputs:
    command:    Command to run Iplot.
    ext:        File extention for MISES imputs
    cwd_in:     Directory in which the subprocess runs.
                Use None as the default value.
    dump:       Will write Iplot stdout to a dump file."""
```

```
print ('Running iplot')
# Run IPLOT in a subprocess.
iplot_proc = subprocess.Popen([command, ext], cwd=cwd_in,
                        stdin=subprocess.PIPE, stdout=subprocess.PIPE)
# Supply sequence of parameters to IPLOT
# '\n' is the same as hitting 'Return'
iplot_out = iplot_proc.communicate
("8\n{0}.txt\n0".format(ext))[0].splitlines()
return
```

Appendix B

MATLAB Code for Generating Shape Parameters

```
function [] = main2()

    clear all
    clc
    format long

    global CLx CLy kapa s1 s2 s3 s4 s5 xtop ytop xbot ybot T NumberOfNodes

    a = importdata('ag17.txt');
    [max_thickness,max_thickness_position,camber,max_camber_position,R,TE_angle]
    =  airfoil_parameters_specifying( a ) ;

    % Physical parameters
    % A sample
    max_thickness = 0.055 ;
    max_thickness_position = 0.5;
```

```
camber = 15;

max_camber_position = 0.5 ;

R = 0.001 ;

TE_angle = 10 ;

Stagger = 0;



Re = 300000;



x = [max_thickness   max_thickness_position   camber   max_camber_position
 incidence  M0   115000   s_c   R TE_angle] ;



NumberOfNodes = 80 ;


[Xc,Yc,Ybot,Ytop,y_t,Xtop,Xbot,dyc_dx] = ...
airfoil_geom(max_thickness,max_thickness_position,camber,max_camber_position
,R,TE_angle,Stagger) ;


AOA = atand(dyc_dx(1)) + incidence ;


CLx = Xc ;

CLy = Yc ;

xtop =  Xtop;

ytop = Ytop ;

xbot = Xbot ;

ybot = Ybot ;

T = y_t ;

kapa = atand(dyc_dx) ;
```

```
[Ue_S_pitch,Ue_P_pitch] = velocity_distribution (M0,AOA,s_c) ;



for i = 1:NumberOfNodes
    d1 = sprintf('%s%d%s%1.40d%s','s1[',i-1,'] = ',s1(i),';') ;
    d2 = sprintf('%s%d%s%1.40d%s','s2[',i-1,'] = ',s2(i),';') ;
    d3 = sprintf('%s%d%s%1.40d%s','s3[',i-1,'] = ',s3(i),';') ;
    d4 = sprintf('%s%d%s%1.40d%s','s4[',i-1,'] = ',s4(i),';') ;
    d5 = sprintf('%s%d%s%1.40d%s','s5[',i-1,'] = ',s5(i),';') ;
    d6 = sprintf('%s%d%s%1.40d%s','x_c[',i-1,'] = ',CLx(i),';') ;
    d7 = sprintf('%s%d%s%1.40d%s','kapa[',i-1,'] = ',kapa(i),';') ;
    F1(i) = cellstr(d1) ;
    F2(i) = cellstr(d2) ;
    F3(i) = cellstr(d3) ;
    F4(i) = cellstr(d4) ;
    F5(i) = cellstr(d5) ;
    F6(i) = cellstr(d6) ;
    F7(i) = cellstr(d7) ;
end

F01 = mat2dataset(F1') ;
F02 = mat2dataset(F2') ;
F03 = mat2dataset(F3') ;
F04 = mat2dataset(F4') ;
F05 = mat2dataset(F5') ;
F06 = mat2dataset(F6') ;
F07 = mat2dataset(F7') ;
F9 = [F01;F02;F03;F04;F05;F06;F07];
```

```
    export(F9,'file','data.txt')

end

function [Xc,Yc,Ybot,Ytop,y_t,Xtop,Xbot,dyc_dx] = ...
    airfoil_geom(max_thickness,max_thickness_position,camber
    ,max_camber_position,R,TE_angle,Stagger)

    global NumberOfNodes

%     """
%     The geometry of space parameterized airfoil is generated
%     This airfoil section is used in compressor and Specifications
%     Developed by Syamak Pazireh (Oct 2019)
%     """

    t_max = max_thickness ;                     %t_max/chord
    t_max_position = max_thickness_position ;       % t_max_position/chord
    p = max_camber_position ;
    % max_camber_position/chord
    R_LE = R  ;                               % Fraction of LE radius over chord
    beta = TE_angle ;                           % Trailing edge boat-tail angle
    d = tan(camber*pi/180) ;
    m = (sqrt(1-4*d^2*(p-1)*p)-1)/(4*d) ;


    x = linspace(0,1,NumberOfNodes) ;
    CLx = x ;

    i = 0 ;
```

```
% Calculation of local y_c (camberline)
for i=1:NumberOfNodes
    X = x(i) ;
    if (X<p)
        y_c(i) = (m/p^2*(2*p*X-X^2)) ;
        % equation of camber line
        dyc_dx(i) = (2*m/p^2*(p-X))   ;
        % slope of camberline
    else
        y_c(i) = (m/(1-p)^2*((1-2*p)+2*p*X-X^2)) ;
        % equation of camber line
        dyc_dx(i) = (2*m/(1-p)^2*(p-X)) ;
        % slope of camberline
    end
    THETA(i) = atan(dyc_dx(i)) ;
end

CLy = y_c ;

XT = t_max_position ;

% ref paper is : "An improved geometry parameter airfoil parameterization
 method" Lu Xiaoqiang et. al. 2018
% Journal of Aerospace science and technology

C = [XT^0.5          XT^1   XT^2    XT^3      XT^4 ;
       0.5*XT^-0.5    1         2*XT   3*XT^2  4*XT^3;
       0.25                0.5      1         1.5          2;
       1                    0         0         0              0;
       1                    1        1         1              1] ;
```

```
B = [t_max/2   ;    0  ;     -tan(beta/2*pi/180)  ; sqrt(2*R_LE)  ; 0] ;
A = inv(C)*B ;

for i =1:NumberOfNodes
    y_t(i) =
     (A(1)*CLx(i)^0.5+A(2)*CLx(i)^1+A(3)*CLx(i)^2+A(4)*CLx(i)^3+A(5)*CLx(i)^4 ;
    xtop(i) = (CLx(i) + (y_c(i))*sin(THETA(i))) ;
    ytop(i) = (CLy(i) + (y_t(i))*cos(THETA(i))) ;
    xbot(i) = (CLx(i) - (y_c(i))*sin(THETA(i))) ;
    ybot(i) = (CLy(i) - (y_t(i))*cos(THETA(i))) ;
end

for i=1:NumberOfNodes
    if (xtop(i)== 0 && ytop(i) == 0)
        alpha_top(i) = (0) ;
    else
        alpha_top(i) = (asin(ytop(i)/(sqrt(xtop(i)^2+ytop(i)^2)))) ;
    end
    if (xbot(i)== 0 && ybot(i) == 0)
        alpha_bot(i) = (0) ;
    else
        alpha_bot(i) = (asin(ybot(i)/(sqrt(xbot(i)^2+ybot(i)^2)))) ;
    end
    if (x(i)== 0 && y_c(i) == 0)
        alpha_camber(i) = (0) ;
    else
        alpha_camber(i) = (asin(y_c(i)/(sqrt(x(i)^2+y_c(i)^2)))) ;
    end

    Xtop(i) = cos(Stagger*pi/180 + alpha_top(i))*(sqrt(xtop(i)^2+ytop(i)^2)) ;
```

```
        Ytop(i) = sin(Stagger*pi/180 + alpha_top(i))*(sqrt(xtop(i)^2+ytop(i)^2)) ;
        Xbot(i) = cos(Stagger*pi/180 + alpha_bot(i))*(sqrt(xbot(i)^2+ybot(i)^2)) ;
        Ybot(i) = sin(Stagger*pi/180 + alpha_bot(i))*(sqrt(xbot(i)^2+ybot(i)^2)) ;
        Xc(i)    = cos(Stagger*pi/180 + alpha_camber(i))*(sqrt(x(i)^2+y_c(i)^2)) ;
        Yc(i)    = sin(Stagger*pi/180 + alpha_camber(i))*(sqrt(x(i)^2+y_c(i)^2)) ;

    end
end

function [Ue_SS,Ue_PP] = velocity_distribution (M0,AOA,s_c)

    global CLx CLy s1 s2 s3 s4 s5 xtop ytop xbot ybot T NumberOfNodes

    X = CLx.*2 - 1 ;
    TT = T.*2 ;
    s = s_c.*2 ;
    Y = CLy.*2 ;

    k = NumberOfNodes ;

% % ********************** Ue Calculation
 **********************************************



    for j = 1:k-1
        sum = 0 ;
        for l = 1:k-1
            if (j ~= l)
                sum = (TT(l+1)-TT(l))*coth(pi*(X(l)-X(j))/s) +sum ;
            end
```

```
        end
        s1(j) = -1/s*sum ;

        s2(j) = (T(j+1)-T(j))/(xtop(j+1)-xtop(j)) ;
        s5(j) =  (CLy(j+1)-CLy(j))/(CLx(j+1)-CLx(j)) ;

        sum = 0 ;
        for l = 1:k-1
            if (j ~= l)
                sum = (Y(l+1)-Y(l))*sqrt(sinh(pi*(1+X(l))/s)/sinh(pi*(1-X(l))
                /s))*(coth(pi*(X(l)-X(j))/s) - 1)+sum ;
            end
        end

        s4(j) = -1/s*sqrt(sinh(pi*(1-X(j))/s)/sinh(pi*(1+X(j))/s))*sum ;
        s3(j) = exp(-pi/s)*sqrt(sinh(pi*(1-X(j))/s)/sinh(pi*(1+X(j))/s)) ;
    end
%
    s1(k) = s1(k-1) ;
    s2(k) = s2(k-1) ;
    s3(k) = s3(k-1) ;
    s4(k) = s4(k-1) ;
    s5(k) = s5(k-1) ;

% % ********************** Compressibility factors *************************
    Cpi = 1 - ((1+s1).^2)./(1+s2.^2) ;
    if (M0<1)
        beta = sqrt(1-M0^2) ;
        B = sqrt(1 - M0^2 *(1 - M0*Cpi)) ;
    else
```

```
        beta = pi*sqrt(M0^2-1)/2 ;
        B = sqrt(M0^2 *(1-M0*Cpi)-1) ;
    end
    if(beta<1/3)
        beta = 1/3;
    end


% % *****************************************************************


    Ue_SS = sqrt(abs(((cosd(AOA).*(1+s1./B+s4./beta)+sind(AOA)./beta.*(s3.
    /B)).^2)./((1+((s2+s5)./B).^2))))*M0;
    Ue_PP = sqrt(abs(((cosd(AOA).*(1+s1./B-s4./beta)-sind(AOA)./beta.*(s3.
    /B)).^2)./((1+((s2-s5)./B).^2))))*M0;

%     CpS = (1- (Ue_SS/M0).^2)./sqrt(1-M0^2) ;
%     CpP = (1- (Ue_PP/M0).^2)./sqrt(1-M0^2) ;
%     Ue_SS = sqrt(1- ((CpS.*0.7.*M0.^2 + 1).^(1/3.5) - 1 )./(0.2*M0^2)) ;
%     Ue_PP = sqrt(1- ((CpP.*0.7.*M0.^2 + 1).^(1/3.5) - 1 )./(0.2*M0^2)) ;
end
```

Appendix C

Artificial Neural Network Model Coefficients

```
xmax_ANN[0] = 0.1499999999999999944488848768742172978818;
xmin_ANN[0] = 0.0249999999999999979183318288278314867057;
xmax_ANN[1] = 0.5000000000000000000000000000000000000000;
xmin_ANN[1] = 0.2999999999999999888977697537484345957637;
xmax_ANN[2] = 40.0000000000000000000000000000000000000000;
xmin_ANN[2] = 5.0000000000000000000000000000000000000000;
xmax_ANN[3] = 0.5999999999999999777955395074968691915274;
xmin_ANN[3] = 0.3999999999999999666933092612453037872910;
xmax_ANN[4] = 6.0000000000000000000000000000000000000000;
xmin_ANN[4] = -6.0000000000000000000000000000000000000000;
xmax_ANN[5] = 1.5999999999999998667732370449812151491642;
xmin_ANN[5] = 0.0999999999999999916733273153113259468228;
xmax_ANN[6] = 1510000.0000000000000000000000000000000000000000;
xmin_ANN[6] = 100000.0000000000000000000000000000000000000000;
xmax_ANN[7] = 2.0000000000000000000000000000000000000000;
```

```
xmin_ANN[7] = 0.5000000000000000000000000000000000000000;
xmax_ANN[8] = 0.0200000000000000004163336342344337026589;
xmin_ANN[8] = 0.0010000000000000000208166817117216851329;
xmax_ANN[9] = 10.0000000000000000000000000000000000000000;
xmin_ANN[9] = 0.0000000000000000000000000000000000000000;
ymax[0] = 0.0377740000000000002019717726398084778338 7;
ymin[0] = 0.0009949699999999999690258878359827576787;
ymax[1] = 0.0188229999999999999454214361094273044727 7;
ymin[1] = 0.0001510900000000000187750509583750613274;
NumberofLayers = 1;
bias1[0] = -2.0808878071177629998089742002775892615318;
bias1[1] = -3.8814874884315888614594314276473596692085;
bias1[2] = 2.9625971323225201992102029180387035012245;
bias1[3] = -3.8383376846925059133752711204579100012779;
bias1[4] = -1.2288064735553749784457977511920034885406;
bias1[5] = 1.0861111533118279481868739821948111057281;
bias1[6] = 0.9895276073723939802562199474778026342392;
bias1[7] = -0.3167583379988440084318312983668874949217;
bias1[8] = -2.2896977548494681364843472692882642149925;
bias1[9] = -0.8891694461031469653988779100473038852215;
bias1[10] = 2.5065931110113019997243100078776478767395;
bias1[11] = -2.1657518235405519568814725062111392617226;
bias1[12] = -0.9982629374741469607812405229196883738041;
bias1[13] = 0.9160841864987530058783704589586704969406;
bias1[14] = 1.4731706467550500505581112520303577184677;
bias1[15] = 0.4235736924366679811981839520740322768688;
bias1[16] = 0.8409251279728250372258457900898065418005;
bias1[17] = 0.0073209636660619996015308963421830412699;
bias1[18] = -0.1892560666526330037662262384401401504874;
bias1[19] = 0.7858193223318139519761871270020492374897;
```

```
bias1[20] = 0.9622743799120370145061542643816210329533;
bias1[21] = 0.6071742256314790475713039086258504539728;
bias1[22] = 0.8829438475999540347771699089207686483860;
bias1[23] = -0.7275370053164890160601885327196214348078;
bias1[24] = 2.5162756884517820665791987266857177019119;
bias1[25] = 1.0515379986534540979192797749419696629047;
bias1[26] = 1.3469350002577500369227436749497428536415;
bias1[27] = -1.2796345226386300897303271995042450726032;
bias1[28] = 1.3931853948822059408030327176675200462341;
bias1[29] = -0.9240189752880729745498911142931319773197;
bias1[30] = 1.9654000485611780035810625122394412755966;
bias1[31] = -0.9316950668131960267004387787892483174801;
bias1[32] = -1.4214953333730520057542889844626188278198;
bias1[33] = -1.6543294709189639846158570435363799333572;
bias1[34] = -1.9912952911394921073906516539864242076874;
bias1[35] = 1.6785003374125440078756810180493630468845;
bias1[36] = 1.9784358043314440589455216468195430934429;
bias1[37] = -1.7090268323451121013079045951599255204201;
bias1[38] = 1.5040604780271469298469355635461397469044;
bias1[39] = -4.9991752039636754290086173568852245807648;
bias2[0] = 0.3866316084141199760360052550822729244828;
bias2[1] = -0.3956789390961200081342497014702530577779;
W1[0][0] = -0.2142917395701590121959156931552570313215;
W1[0][1] = 0.1712283378008589984631271363468840718269;
W1[0][2] = 0.6418763595404549660017323731153737753630;
W1[0][3] = -0.1493634233156619917082252868567593395710;
W1[0][4] = 0.7779678743958630127153242028725799173117;
W1[0][5] = -1.3485474188076660428947661785059608519077;
W1[0][6] = -0.1130563856633139963037137931678444147110;
W1[0][7] = -1.3439590768506870510634598758770152926445;
```

```
W1[0][8] = -0.0859306692332140048540267684984428342432;
W1[0][9] = -0.0436735520475309973997823931313178036362;
W1[1][0] = 1.4598802488438940017090317269321531057358;
W1[1][1] = -0.1489542250578400062188677566155092790723;
W1[1][2] = -0.4321252257244679872982828783278819173574;
W1[1][3] = 0.1978751863209680106336918470333330333233;
W1[1][4] = -0.2378351969521380060434978531702654436231;
W1[1][5] = 0.4833523395731080252168965216696960851550;
W1[1][6] = -3.1338805311005240028521257045213133096695;
W1[1][7] = 0.6408198160345599525555826403433457016945;
W1[1][8] = -0.0972610763857120058650806981859204825014;
W1[1][9] = -0.5635502187209630031716756093373987823725;
W1[2][0] = -0.7847720597017270316442250077670905739069;
W1[2][1] = 0.8362928992235310055036734411260113120079;
W1[2][2] = 0.4181991101347189987436081537452992051840;
W1[2][3] = -0.2422088727127340024303236987179843708873;
W1[2][4] = 0.1346979295223729999886330688241287134588;
W1[2][5] = -0.0572209264680480014786390086101164342836;
W1[2][6] = 1.0322523062020310113950927188852801918983;
W1[2][7] = -0.1802552795373119909339010291660088114440;
W1[2][8] = 0.0312151766441680009189774125388794345781;
W1[2][9] = 0.2994751742202430011552394262253073975444;
W1[3][0] = 0.2976835495152029831267270765238208696246;
W1[3][1] = 0.0508158491612199988973763709054765058681;
W1[3][2] = 0.2763429907240960159420239961036713793874;
W1[3][3] = -0.1229524061880010038505517400153621565551;
W1[3][4] = 0.1319484164116890012596883252626867033541;
W1[3][5] = -0.5992754593015590103277645539492368698120;
W1[3][6] = -2.2408806965022991519731476728338748216629;
W1[3][7] = -0.1912011105899220075876598912145709618926;
```

```
W1[3][8] = 0.14641725856951198769806410382443573325587;
W1[3][9] = -0.14574697522249399628258004213421372696766;
W1[4][0] = -0.02794722093017200023612467418843152699999;
W1[4][1] = -0.64529324416140199094371610044618137180811;
W1[4][2] = 0.55887735973153096491472524576238356530677;
W1[4][3] = 0.24998897706822201247867099027644144371155;
W1[4][4] = -1.38246169372845995049203793314518406987199;
W1[4][5] = -0.43317178419240698339365280844504013657577;
W1[4][6] = 0.70640168800018798833662003744393587112433;
W1[4][7] = 0.92984186226259801788529557597939856350422;
W1[4][8] = -0.31930806293583102739574997031013481318955;
W1[4][9] = -0.54323365968413495075139962864341214299200;
W1[5][0] = -0.65264553830792604305344184467685408890255;
W1[5][1] = 0.01449334802450799944417703102317318553100;
W1[5][2] = -0.66262046406646801788298262181342579424388;
W1[5][3] = 0.24135811786626901187524651959392940625555;
W1[5][4] = -0.41920882852662100237495224064332433044911;
W1[5][5] = 2.14946748340141802202651888364925980567933;
W1[5][6] = 0.04628583198603099940759975083892641123388;
W1[5][7] = 0.41354677753213497926054742492851801216600;
W1[5][8] = -0.17822724340017601085683907058410113677388;
W1[5][9] = 0.12920100264773198839307610796822700649500;
W1[6][0] = -0.74129951771543001193265354231698438525200;
W1[6][1] = -0.48983800013340800250105644408904481679200;
W1[6][2] = 0.64763967311041004126792586248484440147888;
W1[6][3] = -1.27606098842804605020262442849343642592433;
W1[6][4] = 0.35004618946055698902597441701800562441355;
W1[6][5] = 0.63901213523716005049379873526049777865411;
W1[6][6] = 0.66412173832599397105269645180669613182544;
W1[6][7] = -0.40881273333615997245260587078519165515900;
```

```
W1[6][8] = -0.3809880904634599940727923694794299080968;
W1[6][9] = -0.0687436269896630064435427698299463372678;
W1[7][0] = 0.0187349799093310016850910670882512931712;
W1[7][1] = -0.2110327911815879919910088347023702226579;
W1[7][2] = 0.1498572724534890099334916158113628625870;
W1[7][3] = -0.0026645528720130001558430432595514503191;
W1[7][4] = 0.3873356109768180122188141467631794512272;
W1[7][5] = 0.0826111793258119952998441704039578326046;
W1[7][6] = 0.2865417711562749936149430141085758805275;
W1[7][7] = 0.0762129212048780063648578675383760128170;
W1[7][8] = -0.0623919363176859975705923488931148312986;
W1[7][9] = 0.0539372351529539967307513848027156200260;
W1[8][0] = 1.4333484145065509274274972995044663548470;
W1[8][1] = -0.1845366455159850038736379929105169139802;
W1[8][2] = -0.3833043170549880196418257582990918308496;
W1[8][3] = 0.0454812763631270017272711925215844530612;
W1[8][4] = 0.1148235929775670016717015187168726697564;
W1[8][5] = -0.2837579931317690262737585271679563447833;
W1[8][6] = -0.3478782580523249734483215434011071920395;
W1[8][7] = 0.8072000103686629568500165987643413245678;
W1[8][8] = -0.1079178617821630048467795859323814511299;
W1[8][9] = 0.1598082740726040062995139123813714832067;
W1[9][0] = 0.5255535858896469569856435555266216397285;
W1[9][1] = -0.1321023768200089987256973245166591368616;
W1[9][2] = -0.1655215084446750029023576189501909539104;
W1[9][3] = 0.0718773075408259964902413230447564274073;
W1[9][4] = -0.7015423544763319529948830677312798798084;
W1[9][5] = -1.0199872927635209585162101575406268239021;
W1[9][6] = -0.9185467496141519472274694635416381061077;
W1[9][7] = 0.5703810151116519566727447454468347132206;
```

```
W1[9][8] = 0.0813237755761039987634219983192451763898;
W1[9][9] = 0.1371586921210030118878364646661793813109;
W1[10][0] = -1.1612479955550600863034560461528599262238;
W1[10][1] = -0.2545148857953810073162514981959247961640;
W1[10][2] = -0.2946963130894200144638261917862109839916;
W1[10][3] = -0.1327375927701530122782003218162572011352;
W1[10][4] = -0.0935480270313220063638226520197349600494;
W1[10][5] = -0.8446777318038289905643978272564709186554;
W1[10][6] = 1.3067678998214560248669613429228775203228;
W1[10][7] = -0.0046686768179909996639986857758231053594;
W1[10][8] = -0.0395598837111749979111863240177626721561;
W1[10][9] = 0.3150037689988470224200511893286602571607;
W1[11][0] = 0.2136627640467489985809379504644311964512;
W1[11][1] = 0.0376469140676789978949834392096818191931;
W1[11][2] = -1.1726050112267940583876679738750681281090;
W1[11][3] = -0.1734444408606030108010997992096235975623;
W1[11][4] = -0.2423177429510789959721250852453522384167;
W1[11][5] = -0.2632351026161889873122845528996549546719;
W1[11][6] = 0.0215966380328590015769130161515931831673;
W1[11][7] = -0.7593442049854539632747219002339988946915;
W1[11][8] = -0.0545780293644569994282456093515065731481;
W1[11][9] = -0.0945814387147539958888842193118762224913;
W1[12][0] = 0.6893978162231579887020416208542883396149;
W1[12][1] = 0.6211830132589180042401721948408521711826;
W1[12][2] = -0.0928075800643920062471536880366329569370;
W1[12][3] = -0.0461450410508549985211246280414343345910;
W1[12][4] = 0.1423903014642000086631412614224245771766;
W1[12][5] = -0.1178462384946890040282596601173281669617;
W1[12][6] = -0.2628690090004550117264159325713990256190;
W1[12][7] = 0.4489700642842229805218323690496617928147;
```

```
W1[12][8] = -0.0452843415553220024483138672621862497181;
W1[12][9] = -0.3679591005682290028033776252414099872112;
W1[13][0] = 0.1766650483134679927577082025891286320984;
W1[13][1] = -0.5874240990787139571338570931402500718832;
W1[13][2] = 0.5466011750655529466413895534060429781675;
W1[13][3] = -0.6746511254996629958213816280476748943329;
W1[13][4] = -0.8603246085520840358284999638271983712912;
W1[13][5] = -0.2742705652093320178863677938352338969707;
W1[13][6] = -0.2530932287264109903368591858452418819070;
W1[13][7] = -1.0451407055190289341339848760981112718582;
W1[13][8] = 0.9123345108686820026377972681075334548950;
W1[13][9] = 0.1601230359557520022129040171421365812421;
W1[14][0] = 0.9740670393110310021356212928367312997580;
W1[14][1] = -0.5232008722601080297209819036652334034443;
W1[14][2] = -0.5302723648788919552643505994637962430716;
W1[14][3] = 0.2655572564938060153849619382526725530624;
W1[14][4] = -1.4929819958039489691259404935408383607864;
W1[14][5] = 1.1592397255804220979058527518645860254765;
W1[14][6] = -1.0038879763094550323643261435790918767452;
W1[14][7] = 1.4686103735146720516979712556349113583565;
W1[14][8] = 0.2248242529281070101188078069753828458488;
W1[14][9] = 0.2458470677047759900268886212870711460710;
W1[15][0] = -0.7466584966404530288031082818633876740932;
W1[15][1] = -0.1517986142882550015809073329364764504135;
W1[15][2] = -0.5155047395570909740669662824075203388929;
W1[15][3] = -0.1858541814487640053688011221311171539128;
W1[15][4] = -0.0925814412313469975446267312690906692296;
W1[15][5] = -0.0856485634765130066092808647226775065064;
W1[15][6] = 0.3982446432860550222798678987601306289434;
W1[15][7] = 0.0264840024271300010505481026257257326506;
```

```
W1[15][8] = -0.1043281936516699953676479140085575636476;
W1[15][9] = -0.0994089255375709968109987357820500619709;
W1[16][0] = 0.2280145218118440064714036452642176300287;
W1[16][1] = -0.0905963233319929950315696487450622953475;
W1[16][2] = -0.1236305966010440016145111030709813348949;
W1[16][3] = 0.0258005151705349988766613478219369426370;
W1[16][4] = -0.6565507007314830278943418306880630552769;
W1[16][5] = -0.2264555406300469975189315618990804068744;
W1[16][6] = 0.3827797182900459782572966105362866073847;
W1[16][7] = 0.6484207832986480513071114728518296033144;
W1[16][8] = -0.1970885436867199913546500056327204219997;
W1[16][9] = -0.1766120758836549897718981583238928578794;
W1[17][0] = 0.8458863895526900211763177139800973236561;
W1[17][1] = -0.0534369155219689973002061833540210500360;
W1[17][2] = 0.1553640590533900134317946140072308480740;
W1[17][3] = 0.0015953447811310000024886956992986597470;
W1[17][4] = -0.5151947730902279865006221371004357933998;
W1[17][5] = 0.3277564102409359914069852948159677907825;
W1[17][6] = -0.2712209326897069838757658999384148046374;
W1[17][7] = 0.2183425355708630000162173701028223149478;
W1[17][8] = -0.2095215389256989968469468976763891987503;
W1[17][9] = 0.0001998453451219999973172830598500127053;
W1[18][0] = -0.1498650091388289973259873022470856085420;
W1[18][1] = -0.2914321466073619970593711059336783364415;
W1[18][2] = -0.6481360250840779713144002016633749008179;
W1[18][3] = -0.0013538989407330000429990901622545607097;
W1[18][4] = -0.5515686688856880071085697636590339243412;
W1[18][5] = -0.4598334378647879749912874558503972366452;
W1[18][6] = -0.0735744205833119974125011708565580192953;
W1[18][7] = 0.8203678652618650035677205778483767062426;
```

```
W1[18][8] = 0.1520651739195600093967897237234865315259;
W1[18][9] = 0.1095352386915739995032481601811014115810;
W1[19][0] = 0.7692634409434599973565127584151923656464;
W1[19][1] = -0.1869857635656609895491442330239806324244;
W1[19][2] = -0.0191979496981759985285709291247258079238;
W1[19][3] = 0.0517324261860380024691252742741198744625;
W1[19][4] = -0.4088294080793309914767519330780487507582;
W1[19][5] = 1.2270751396140429090308998638647608458996;
W1[19][6] = 0.3367755838627429776366284386313054710627;
W1[19][7] = 0.4447376541833670260395194873126456514001;
W1[19][8] = -0.0990621451323450002979242867695575114340;
W1[19][9] = 0.0079257244657880004945482710354554001242;
W1[20][0] = 0.5068279743481559629358912388852331787348;
W1[20][1] = -0.0150467019214229995316411958583557861857;
W1[20][2] = -0.4159061999505149764821965163719141855836;
W1[20][3] = 0.1604778019683049949950515156160690821707;
W1[20][4] = -0.6005825082241800227933481437503360211849;
W1[20][5] = 0.9282402836813119506587099749594926834106;
W1[20][6] = -0.4271828481884509809418659642687998712063;
W1[20][7] = 0.3414429725412059801747943765803938731551;
W1[20][8] = 0.9238727166407529889369243392138741910458;
W1[20][9] = 0.0056858790049819996417768130925196601311;
W1[21][0] = -0.3943841832228209831612275593215599656105;
W1[21][1] = -0.0087163674450990002301464798506458464544;
W1[21][2] = -0.1719724103094489864318461513903457671404;
W1[21][3] = -0.0060296735952679999734349358675444818800;
W1[21][4] = 0.1299227477116310092153383948243572376668;
W1[21][5] = 2.4373834136593779398083370324457064270973;
W1[21][6] = 0.1068690843063089951359145857168186921626;
W1[21][7] = -0.0246631884418729990793295314688293728977;
```

```
W1[21][8] = -0.0119716539959360008388289386971337080467;
W1[21][9] = 0.0479970237625549980831074492471088888124;
W1[22][0] = -0.1327673746634679963207759101351257413626;
W1[22][1] = 0.1475213931550579948837054189425543881953;
W1[22][2] = 0.2396791047152599940517347931745462119579;
W1[22][3] = 0.0410901563834010011277975138455076375976;
W1[22][4] = 0.3652210932749869742863779720210004597902;
W1[22][5] = 1.0302443034117929610005148788332007825375;
W1[22][6] = 0.6435411906355960054426645911007653921843;
W1[22][7] = -0.5181479230063119612026412141858600080013;
W1[22][8] = -0.1126484005779739977493036917621793691069;
W1[22][9] = -0.2227881951768919954748326972548966296017;
W1[23][0] = 0.3751737516933210114800090195785742253065;
W1[23][1] = -0.0416005743503760003210167894849291769788;
W1[23][2] = 0.0096416101979459999798960367911604407709;
W1[23][3] = 0.0069220092001170000287069150601837463910;
W1[23][4] = -0.1568635409747149989101444589323364198208;
W1[23][5] = -2.8249610347480538230513502639951184391975;
W1[23][6] = 0.4933543409664319923102482334797969087958;
W1[23][7] = 0.0850118308031909963862204904216923750937;
W1[23][8] = -0.0024871598994319998703061980194206626038;
W1[23][9] = -0.0299016292631240007682258408294728724286;
W1[24][0] = 0.8234016888852530025033615856955293565989;
W1[24][1] = -0.1636392172818849910509442224793019704521;
W1[24][2] = 0.0709726138942290002420421046736009884626;
W1[24][3] = -0.1404034189074480076797613037342671304941;
W1[24][4] = 0.6844783927470730056086267723003402352333;
W1[24][5] = 1.7746347643874988975198903062846511602402;
W1[24][6] = -0.2316612452573419966839196604269091039896;
W1[24][7] = -0.1156040086333850053224026055431750137359;
```

```
W1[24][8] = -0.29026896717546601944803796868654899299914;
W1[24][9] = -0.0169617563099680006277747423837354290299;
W1[25][0] = 0.6586269262733660534436808120517525821924;
W1[25][1] = 0.2637570589214500160757381763687590137124;
W1[25][2] = 0.1347291698105499979654098297032760456204;
W1[25][3] = 0.2892707016652200091577640250761760398746;
W1[25][4] = 0.1598587112912830121569385255497763864696;
W1[25][5] = 1.1997821031115820655799097949056886136532;
W1[25][6] = -0.2081884797671980091848809024668298661709;
W1[25][7] = -0.0144622412209769995311559043216220743489;
W1[25][8] = -0.0118323152576060002666435977403125434648;
W1[25][9] = 0.7351966395607449911153707944322377443314;
W1[26][0] = 1.0047832700823209339802133399643935263157;
W1[26][1] = -0.4410181944009909926762702525593340396881;
W1[26][2] = -0.4355249770984139789931077757501043379307;
W1[26][3] = 0.2416303063693640007514318313042167574167;
W1[26][4] = -1.3279985715874480511899946577614173293114;
W1[26][5] = 0.7916368113345190193896883101842831820250;
W1[26][6] = -1.0558745558000499631390312060830183327198;
W1[26][7] = 1.2949391332677180344035150483250617980957;
W1[26][8] = 0.1955039305866859977722782559794723056257;
W1[26][9] = 0.1719317914361210009044356183949275873601;
W1[27][0] = 0.3121404542743980159436034682585159316659;
W1[27][1] = -0.0397585368109540024383896650306269293651;
W1[27][2] = -0.0777851095531350011924587306566536426544;
W1[27][3] = -0.0215078324704669997247119539451887249015;
W1[27][4] = -0.1656245912572620049285632148894364945590;
W1[27][5] = -3.6960692147235230109458825609181076288223;
W1[27][6] = 0.8235364401306369819266706144844647496939;
W1[27][7] = 0.2106435776369499912608063141306047327816;
```

```
W1[27][8] = 0.0103985192016520008179147893656590895262;
W1[27][9] = 0.0164815611044310007804458706459627137519;
W1[28][0] = 0.9435622172440439880958251706033479422331;
W1[28][1] = -0.3528733004223780223007622680597705766559;
W1[28][2] = 0.1731581394334990031680376887379679828882;
W1[28][3] = 0.0839479553119820004480899910959124099463;
W1[28][4] = 1.2854635102685600944738553153001703321934;
W1[28][5] = -0.4647972418682599915484843222657218575478;
W1[28][6] = -0.8770798571543709787334819338866509497166;
W1[28][7] = -0.1625147677942209989421229465733631514013;
W1[28][8] = -0.7896956481189739696091578480263706296682;
W1[28][9] = -0.0676433087704080004964879435647162608802;
W1[29][0] = -0.6727970827183400448490147027769125998020;
W1[29][1] = 0.1146855936223930005901650019950466230512;
W1[29][2] = 0.6370511323556079519292438817501533776522;
W1[29][3] = -0.1344682189011250084131887660987558774650;
W1[29][4] = 0.5390208276945860355056083790259435772896;
W1[29][5] = 0.3401972747395339835563277119945269078016;
W1[29][6] = 0.3477981477762310258405875629250658676028;
W1[29][7] = -1.2857897358176579860611354888533242046833;
W1[29][8] = 0.1177645712032869967922366072343720588833;
W1[29][9] = 0.0125603886066739996868468765001125575509;
W1[30][0] = -0.6071954429525869834094464749796316027641;
W1[30][1] = 0.3533568891691239999275353511620778590441;
W1[30][2] = -0.6027242309487980120863426236610393971205;
W1[30][3] = 0.4336606040874820267028155740263173356652;
W1[30][4] = 0.0414357256543999991071913768792001064867;
W1[30][5] = -1.5682834733680028893587632410344667732716;
W1[30][6] = -0.1288066485609420108726652642872068099678;
W1[30][7] = 1.1459414965716820944408027571626007556915;
```

```
W1[30][8] = -0.1685799493129070125707613669874262996018;
W1[30][9] = 0.0350870116230520001643355954001890495420;
W1[31][0] = -0.8664337593802170500723036639101337641478;
W1[31][1] = -0.0500046678972220007275417685832508141175;
W1[31][2] = -0.0120267853474550004411192105635564075783;
W1[31][3] = -0.1080980177071020065238116103500942699611;
W1[31][4] = 0.3769900719530310251315086134127341210842;
W1[31][5] = -3.8745104135065369099777399242157116532326;
W1[31][6] = 0.7926130593190710005302435092744417488575;
W1[31][7] = -0.3268951401990859984003634508553659543395;
W1[31][8] = -0.1160166226346750067976998366248153615743;
W1[31][9] = 0.0182545838918510006099182874095276929438;
W1[32][0] = -0.7106958187489189615604345817700959742069;
W1[32][1] = 0.2840726483345459851825864916463615372777;
W1[32][2] = -0.1322661890987139876596501153471763245761;
W1[32][3] = 0.3635296400609859812647073340485803782940;
W1[32][4] = -0.2525958340058599982391740468301577493548;
W1[32][5] = -0.5176246984679380158667072464595548808575;
W1[32][6] = 0.1696749090228230061327963085204828530550;
W1[32][7] = 0.8171882459156439848513286960951518267393;
W1[32][8] = 0.1746310907858250049962123284785775467753;
W1[32][9] = 0.2440537844731699990141748912719776853919;
W1[33][0] = -0.5925106102751329650857314845779910683632;
W1[33][1] = 0.0617492958719629975172793479032407049090;
W1[33][2] = 0.0139312594256039996160057015117672563065;
W1[33][3] = -0.0468007328564550018068501913148793391883;
W1[33][4] = 0.1728356675158579935924763049115426838398;
W1[33][5] = 2.0409615485138679069621048256522044539452;
W1[33][6] = 0.0176345336810399995475240331188615527935;
W1[33][7] = -0.1662606553235840001203627025461173616350;
```

```
W1[33][8] = -0.4791770640782270218593907884496729820967;
W1[33][9] = -0.3046018597717489728715634100808529183269;
W1[34][0] = -1.0930335392802079663709946544258855283260;
W1[34][1] = 0.1198076290013500022624270968663040548563;
W1[34][2] = -0.5010331827836479456905749430006835609674;
W1[34][3] = 0.2558921869480639821325951288599753752351;
W1[34][4] = -0.2318057331600349990630149932258063927293;
W1[34][5] = -0.4768542710990930078018834592512575909495;
W1[34][6] = -0.8342227828210629914096330139727797359228;
W1[34][7] = 0.1091505820090930001242668367922306060791;
W1[34][8] = -0.1153211048470449973102347485109930858016;
W1[34][9] = 0.2223840336282270069290234459913335740566;
W1[35][0] = 1.0167898259421439544070153715438209474087;
W1[35][1] = -0.0073568421638230000261793328775183908874;
W1[35][2] = 0.0827897118477659982316296805038291495293;
W1[35][3] = 0.0285096620800629994840758030250071897171;
W1[35][4] = -0.2487931606991820032703799370210617780685;
W1[35][5] = -0.4505183356690850016157412483153166249394;
W1[35][6] = 0.5261449252087819772327748069074004888535;
W1[35][7] = 0.4251232819837910126281599332287441939116;
W1[35][8] = -0.0179487181264310000083916918356408132240;
W1[35][9] = -0.0452344592065680034531638398220820818096;
W1[36][0] = 0.9959750650498530033516431103635113686323;
W1[36][1] = -0.1775201693204130071190860462593263946474;
W1[36][2] = 0.3188128479307359985917003086797194555402;
W1[36][3] = -0.7739189099129030191548395123390946537256;
W1[36][4] = 0.2706511419435699838942355199833400547504;
W1[36][5] = 0.6344359806610849750185820994374807924032;
W1[36][6] = 1.8617584223773380180944059247849509119987;
W1[36][7] = -0.2161142630122049879126677751628449186683;
```

```
W1[36][8] = -0.0290701549735969989873840546579231158830;
W1[36][9] = -0.2668641184712259861733230081881629303098;
W1[37][0] = -0.3185166607804459926178708428778918460011;
W1[37][1] = -0.9275802905496609573887667465896811336279;
W1[37][2] = 1.2325358916860280622529444372048601508141;
W1[37][3] = 1.2806671750278020294189218475366942584515;
W1[37][4] = 0.0058187306234630002330088238693406310631;
W1[37][5] = 0.4350659443455410224643742367334198206663;
W1[37][6] = 0.0925551034494749985670480896260414738208;
W1[37][7] = 0.2718594352280799819610024314897600561380;
W1[37][8] = -0.3738378817519160124405175338324625045061;
W1[37][9] = 0.0386058533238970019718649950846156571060;
W1[38][0] = 0.1520992120143550008215527213906170800328;
W1[38][1] = 0.0438500481741600015017645830539549933746;
W1[38][2] = -0.5044714746989189535497644101269543170929;
W1[38][3] = 0.2241300613703730015835446920391405001283;
W1[38][4] = 0.2480594251879749967404364952017203904688;
W1[38][5] = 0.4293099091856160121061236623063450679183;
W1[38][6] = -0.1137066137563459977677382539695827290416;
W1[38][7] = -0.1069635819228989953577624305580684449524;
W1[38][8] = 1.5606374742807789246512584213633090257645;
W1[38][9] = 0.0191502345036009997969461693401171942241;
W1[39][0] = -0.2909132938899580134162192734947893768549;
W1[39][1] = 0.3848827522840920223323735172016313299537;
W1[39][2] = 0.7429732624235110272081783477915450930595;
W1[39][3] = -0.2098626294478060094750304642730043269694;
W1[39][4] = 1.2451240913367769280739594250917434692383;
W1[39][5] = -1.0733741315319440534636896700249053537846;
W1[39][6] = -0.2156330180369669946571775653865188360214;
W1[39][7] = -1.9958639538811209224178355725598521530628;
```

```
W1[39][8] = -0.7092422375050929650441844387387391179800;
W1[39][9] = -0.2786838441928189968876949933473952114582;
W2[0][0] = 0.4926015134086529911883189924992620944977;
W2[1][0] = -0.1832277278086359983966247000353178009391;
W2[0][1] = 0.3605724275022549885072464803670300170779;
W2[1][1] = 0.4615227871805809845007217973034130409360;
W2[0][2] = 0.4059842274783519999736824956926284357905;
W2[1][2] = 0.1563588496888060064282655048373271711171;
W2[0][3] = 0.9091175133930800011938799798372201621532;
W2[1][3] = 0.6834839257097470310853282171592582017183;
W2[0][4] = 0.0279374546738400016121506297395171714015;
W2[1][4] = 0.0140899855016639995869409673900918278378;
W2[0][5] = -0.3521675878011280014234785085136536508799;
W2[1][5] = 0.0183433993905850016570280303085382911377;
W2[0][6] = -0.1001283162370049939493554802538710646331;
W2[1][6] = -0.0202599968668029997709290057628095382825;
W2[0][7] = 0.5051805100498449885293439365341328084469;
W2[1][7] = 0.1841369391050090009720463513076538220048;
W2[0][8] = 0.0494126277969359969932661158509290544316;
W2[1][8] = 0.2160314695833039921879503708623815327883;
W2[0][9] = -0.4434342621355260027904421349376207217574;
W2[1][9] = 0.2660373349657100261467235213785897940397;
W2[0][10] = 0.6408971954259840009271442795579787343740;
W2[1][10] = 1.1923076408702419914931169842020608484745;
W2[0][11] = -0.2555465371927799766993416596960742026567;
W2[1][11] = 0.3397862640045030135915737901086686179042;
W2[0][12] = 0.1202193051843220006347223716147709637880;
W2[1][12] = 0.2379970685179360057048825183301232755184;
W2[0][13] = -0.0245119829219639998041380124504939885810;
W2[1][13] = 0.0088994265553359996195892378523240040522;
```

```
W2[0][14] = 0.7920382729873569616074746591039001941681;

W2[1][14] = -0.1805437395665880073725162446862668730319;

W2[0][15] = 0.3106433121467250257907721788797061890364;

W2[1][15] = 0.0810611484186089942260977636578900273889;

W2[0][16] = 0.3866825084480379759455104249354917556047;

W2[1][16] = -0.0857608790727920067276102145115146413445;

W2[0][17] = 0.6838936841556699786082162972888909280300;

W2[1][17] = -0.0688927398083050007704386530349438544363;

W2[0][18] = -0.3432761195444160029488500640582060441375;

W2[1][18] = -0.0536726561309179978076500105999002698809;

W2[0][19] = -0.5833756107236669485871516371844336390495;

W2[1][19] = 0.0903736302946240022127000202090130187571;

W2[0][20] = 0.3188100869498760037146212198422290384769;

W2[1][20] = 0.0160715527410499985083713880840150522999;

W2[0][21] = 1.2583223301536869964678544420166872441769;

W2[1][21] = 0.4138489150150649886761300422222120687366;

W2[0][22] = -0.6989595952688669733277038176311179995537;

W2[1][22] = 0.0437638946217270008443023243671632371843;

W2[0][23] = 1.5766405255141280328246011777082458138466;

W2[1][23] = -0.2114775206555239983607918929919833317399;

W2[0][24] = 0.1333024555215990003542714248396805487573;

W2[1][24] = -0.5002604713507039502218276538769714534283;

W2[0][25] = 0.0963863217932720006686508895654696971178;

W2[1][25] = 0.1221585157203410043624813852147781290114;

W2[0][26] = -0.9697096436021650500691748675308190286160;

W2[1][26] = 0.1944978879448290010056865639853640459478;

W2[0][27] = -0.8520040556914529883414388677920214831829;

W2[1][27] = 0.1908225951026789879438894104168866761029;

W2[0][28] = -0.0764786588576859965860066381537762936205;

W2[1][28] = -0.0882425565138710010781863957163295708597;
```

```
W2[0][29] = -0.2748718525191010120245493908441858366132;
W2[1][29] = -0.2425730903103479885185578268647077493370;
```

Appendix D

Post-processing MATLAB Code for Time-Averaging

```
function [ ] = contour_CFD_time_averaging_PostProcessing( )

    clear all
    clc
    format long

    % Pre-allocate the variables that will save ALL of the data
    % Preallocating_variables_save_all_data;
    % Pressure contour post procesing for time-averaging
    P = 'E:\PhD thesis';
    S = dir(fullfile(P,'*.csv'));
    M = 0 ;
    O = 0 ;
    for l=1:numel(S)
        %------------------------------INPUTS---------------------------------%
        tic
        F = fullfile(P,S(l).name);
```

```
data = xlsread(F);
y = data(:,2) ;
z = data(:,3) ;
c = data(:,4) ;
C(:,:,1) = c ;

s = size(c);
j = 1 ;
for i = 1:1:s(1)
    r(j,:) = sqrt(y(i)^2+z(i)^2) ;
    j = j + 1 ;
end

Ri = min(r) ;
Ro = max(r) ;
i = 1;
for theta = 0:5:360
    j = 1;
    for r = Ri:0.002:Ro
        Z(i,j) = r*sind(theta) ;
        Y(i,j) = r*cosd(theta) ;
        for k = 1:s(1)
            distance(k) = sqrt((Z(i,j)-z(k))^2 + (Y(i,j)-y(k))^2);
        end
        [d,I] = min(distance) ;
        O(i,j,1) = c(I) ;
        j = j+1;
    end
    i = i+1;
end
```

```
        M = O(:,:,l) + M ;

        toc

    end

    O2 = ((M./numel(S))+101325)./(101325) ;
    figure(1) ;
    [~, hC] = contourf(Y,Z,O2,200) ;
    caxis([1.13 1.24])

    h = colorbar('peer',gca,'SouthOutside','fontsize',20,'FontName',
     'Times','FontWeight', 'bold');
    h.Label.String = ' p_t/ p_{t,in,clean}';
    h.Label.String = '$ \frac{p_t}{(p_{t,clean})_{inlet}}$';
    h.Label.Interpreter = 'latex';
    h.Label.FontSize = 35;
    shading flat
    colormap jet
    set(hC,'LineStyle','none')
    set(gca,'fontsize',25,'FontName', 'Times','FontWeight', 'bold','XColor',
     'none','YColor','none')
    axis equal

end
```

Vita Auctoris

NAME: Syamak Pazireh
PLACE OF BIRTH: Tabriz, Iran
YEAR OF BIRTH: 1989
EDUCATION: Urmia University, B.Sc., Urmia, Iran, 2011
Iran University of Sci. and Tech., M.Sc., Tehran, Iran, 2013