Towards a Design Tool for Turbomachinery

by

Duane R. Epp

A thesis submitted in conformity with the requirements
for the degree of Masters of Applied Science
Graduate Department of Aerospace Engineering
University of Toronto

Abstract

Towards a Design Tool for Turbomachinery

Duane R. Epp
Masters of Applied Science
Graduate Department of Aerospace Engineering
University of Toronto
2010

A two-dimensional thin-layer Navier-Stokes cascade flow solver for turbomachinery is developed. A second-order finite-difference scheme and a second and fourth-difference dissipation scheme are used. Periodic and non-reflecting inlet and outlet boundary conditions are implemented into the approximate-factorization numerical method. Turbulence is modeled through the one-equation Spalart-Allmaras model. A two-dimensional turbomachinery cascade structured grid generator is developed to produce six-block H-type grids.

The validity of this work is tested in various ways. A grid convergence study is performed showing the effect of grid density. The non-reflecting inlet and outlet boundary conditions are tested for boundary placement influence. Comparisons of the flow solver numerical results are performed against experimental results. A Mach number sweep and angle of attack sweep are performed on two similar transonic turbine cascades.

Acknowledgements

I would like to thank Professor David W. Zingg for his financial assistance and for the opportunity to do this research. Thank you for your patience, gentle guidance and expertise throughout my time at UTIAS. I am truly grateful for your belief that I could finish this thesis. It has been a great privilege to have such an excellent supervisor.

To my friends and colleagues in the CFD lab I thank all of you. Markus, thank you for the help with boundary conditions. James, I would have been lost in my first year courses if it wasn't for your brilliance. Mo, not are you only a great friend but you were so helpful during my Approximate Factorization problems. You were there for me to bounce all of my ideas off of and vent my frustrations to. Scott, I can't believe anyone can have as much knowledge about everything as you do. Thank you for all of the help with HPACF and teaching me about computers and Linux in general. Marc, I'm grateful to have found such a great friend and mountain biking buddy! Pradeep, I am thankful for your help during my final days at UTIAS.

I would like to thank my family and friends for their support throughout this long journey. I would like to specifically mention my parents and thank them for their encouraging words and unwavering support. I don't think I could have finished this work without you.

Final and absolutely not the least I would like to thank my best friend and wife, Kate. Your loving support and understanding enabled me to make it through this long journey.

Duane R. Epp

University of Toronto Institute for Aerospace Studies
September 19, 2010

CONTENTS

List of Tables

List of Figures

List of Symbols

Alphanumeric Symbols

a	speed of sound
c	chord length of foil
c_x	axial chord length of foil
C_{ps}	static coefficient of pressure
d	distance function
e	total energy
E, F	inviscid flux functions in x and y directions
E_v, F_v	viscous flux functions in x and y directions
H	stagnation enthalpy
J	Jacobian matrix of coordinate transformation
M	Mach number
M_{is}	isentropic Mach number
P	pressure
Q	conserved flow variables
Q^n	flow variables at time step n
R	flow solver residual
R, Z	cascade coordinates
R^+1, R^-	Riemann invariants
$\mathcal{R}e$	Reynolds number
$\hat{S}$	viscous flux vector for thin-layer approximation
T	temperature
t	time
U, V	ξ and η contravariant velocity components

u, v	x and y components of velocity
x, y, X, Y	Cartesian coordinates

Greek Symbols

α	inlet flow angle
Δt	time discretization step
$\delta()$	dirac delta function
δ_{ij}	Kronecker delta
$\epsilon^{(2)}$	second order artificial dissipation coefficient
$\epsilon^{(4)}$	fourth order artificial dissipation coefficient
γ	specific heat ratio
μ	dynamic laminar viscosity
μ_t	turbulent eddy viscosity
Φ	Phi line
Ψ	Psi line
ρ	density
σ	spectral radius of the flux Jacobian
τ_{ij}	fluid stress dyad
θ	cascade coordinate
$\tilde{\nu}$	turbulence model working variable
Υ	pressure switch
ξ, η	curvilinear coordinates

Superscripts

$\bar{}$	time average
$\hat{}$	transformation to curvilinear space
$\tilde{}$	dimensional variable
^{o}	total value
^{s}	static value
^{T}	transpose

Subscripts

$_1$	inlet value
$_2$	outlet value

$_T$	associated with turbulence model
$_{ext}$	extrapolated value
$_{j,k}$	ξ and η direction nodal indices
$_{ref}$	reference value

Abbreviations

AF	Approximate Factorization
CFD	Computational Fluid Dynamics
UTIAS	University of Toronto Institute for Aerospace Studies

Chapter 1

Introduction

1.1 Motivation

Recent government and public campaigns to reduce greenhouse gas emissions have driven the aerospace industry to find new and innovative ways to reduce fuel burn. Pratt and Whitney is bringing to the market the Pure Power family of geared turbofan engines claiming double digit improvement in fuel burn. Bombardier and Alcoa have introduced new aluminum-lithium alloys for use in aircraft structures that will reduce the weight of Bombardier's upcoming C-Series. Within Prof. David Zingg's CFD group at UTIAS there are many tools to optimize the shapes of aircraft. Professor Zingg's group has shown that gradient based optimization tools have great potential for reducing aircraft drag and, consequently, aircraft fuel consumption. Aircraft engines could benefit from more robust design and optimization tools in order to become more fuel efficient while reducing both environmental noise and greenhouse emissions [19].

1.2 Background

The CFD group at University of Toronto Institute for Aerospace Studies (UTIAS), under Professor Zingg, has created many tools that can be applied to turbomachinery with some changes. The 2D grid generator AMBER2D was created by Nelson [33]. There are numerous flow solvers with varying capabilities including solving 2D and 3D flows. Both the Euler and Navier-Stokes flow equations can be solved using either an approximate factorization or Newton-Krylov numerical algorithms. In 2004, Nichols created TYPHOON a 3D Euler flow solver [37]. TORNADO is a predecessor of TYPHOON and

is a 2D Navier-Stokes flow solver [34].

Prof. Zingg's group's initial optimization tools were Optima2D and OptimaMB, which have gradient-based optimization capabilities for single-element (Optima2D) and multi-element (OptimaMB) 2D airfoils [35]. Driver's 2D optimization code with transition prediction has the capability to optimize the design of laminar flow wings [14]. Rumpfkeil's 2D unsteady turbulent flow optimization can optimize wing design for noise reduction [44]. Recently, Hicken has created a Euler-based aerodynamic optimizer that has the potential to discover novel aircraft configurations [24].

Other groups have developed various turbomachinery tools. He and Ning have developed a nonlinear harmonic methodology to calculate 2D unsteady viscous flows through turbomachinery blades [23]. Duta, Giles and Campobasso have developed a 3D adjoint method to minimize unsteady vibration of turbomachinery blades [15]. Florea and Hall have presented an adjoint method to increase cascade aeroelastic stability for turbomachinery while using the inviscid Euler equations [16]. Corral and Gisbert have shown that a gradient-based method can successfully minimize blade secondary losses while using an unstructured mesh Navier-Stokes solver [11]. Papadimitriou and Giannakoglou successfully implemented a continuous adjoint method to improve the aerodynamic performance of a 3D peripheral compressor blade cascade [39]. Their flow equations were the Navier-Stokes equations and the calculation of turbulence was with the Spalart-Allmaras model [47]. Wu, Liu and Tsai also used an adjoint method to reduce entropy generation on 3D turbine blades. They implemented the adjoint method with a multiblock inviscid and viscous flow solver [52].

Some groups have chosen not to use adjoint or gradient based optimization methods and use genetic or evolutionary algorithms. Samad and Kim have presented a multi-objective optimization of an axial compressor rotor blade using a genetic algorithm [45]. The genetic algorithm made the use of a Reynolds-averaged Navier-Stokes 3D flow solver. Oyama, Liou and Obayashi used an evolutionary algorithm to reduce entropy production in 3D steady state turbomachinery flow [38].

1.3 Objectives

The objective of this thesis is to create the initial building blocks for a turbomachinery design tool. A cascade grid generator was created using Amber2D as a basis. A two-dimensional flow solver for axial compressor and turbine blades was created using

TORNADO as a basis. This thesis will lay down the groundwork for a gradient-based optimization tool for axial flow turbomachinery.

Chapter 2

Governing Equations

2.1 Cascade Coordinate Transformation

Since TORNADO was originally created to solve external aerodynamic flows, a Cartesian coordinate system was used. Axial turbomachinery cascades are not usually mapped with x, y and z Cartesian coordinates; rather R, Z and θ cylindrical coordinates are used (Ainley and Mathieson [2] or Saravanamuttoo, Rogers and Cohen [46] provide good definitions of turbomachinery nomenclature). Usually, turbomachinery 2D problems are solved in the cascade plane [21]. The cascade plane is defined as a projection of a streamline's Z and θ coordinates along the blade axis. If the hub, casing and streamlines of the axial turbomachine are cylindrical in shape, then the cascade plane would be a simple cylinder. The cascade plane allows the use of the Cartesian coordinates of TORNADO to solve 2D flow approximations of axial turbomachinery, using the following elementary transformation:

$$X = Z, \quad Y = R\theta. \tag{2.1}$$

2.2 The Navier-Stokes Equations

The equations solved are the two-dimensional compressible Navier-Stokes equations. In Cartesian coordinates, the equations are as follows:

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = \frac{1}{\mathcal{R}e}\left(\frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y}\right). \tag{2.2}$$

The vector of conservative dependent flow variables is defined as:

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, \tag{2.3}$$

the inviscid fluxes are defined as:

$$E = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e+p)u \end{bmatrix} \quad \text{and} \quad F = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (e+p)v \end{bmatrix}, \tag{2.4}$$

while the viscous fluxes are defined as:

$$E_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \phi_1 \end{bmatrix} \quad \text{and} \quad F_v = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \phi_2 \end{bmatrix} \tag{2.5}$$

where

$$\begin{aligned} \tau_{xx} &= (\mu + \mu_t)(4u_x - 2v_y)/3, \\ \tau_{xy} &= (\mu + \mu_t)(u_y - v_x)/3, \\ \tau_{yy} &= (\mu + \mu_t)(-2u_x + 4v_y)/3, \\ \phi_1 &= u\tau_{xx} + v\tau_{xy} + (\mu \mathcal{P}r^{-1} + \mu_t \mathcal{P}r_t^{-1})(\gamma - 1)^{-1}\partial_x(a^2), \\ \phi_2 &= u\tau_{xy} + v\tau_{yy} + (\mu \mathcal{P}r^{-1} + \mu_t \mathcal{P}r_t^{-1})(\gamma - 1)^{-1}\partial_y(a^2). \end{aligned} \tag{2.6}$$

Using the equation of state for a perfect gas, the pressure (p) is related to the flow variables as follows:

$$p = (\gamma - 1)\left(e - \frac{1}{2}\rho(u^2 + v^2)\right), \tag{2.7}$$

where the definition of sound speed, a, is given by:

$$a = \sqrt{\gamma p/\rho} \quad , \tag{2.8}$$

and

$$\gamma = 1.4 \tag{2.9}$$

is assumed for air.

2.3 Curvilinear Thin Layer Navier-Stokes Equations

It is typical for turbomachinery flow solvers to neglect the viscous derivatives in the steamwise direction in order to reduce computational time. The assumption that allows this to be done is that the effects of viscosity are stronger near the foil surfaces and in the wake region. In these two regions, the gradients of the steamwise diffusion terms are small compared to the normal gradients.

The Navier-Stokes equations in Cartesian coordinates (2.2) are converted to curvilinear coordinates (ξ, η) by using J, the Jacobian of the transformation. In order to arrive at the thin-layer Navier-Stokes equations, the viscous derivatives in the ξ direction are neglected. The resulting curvilinear thin-layer Navier-Stokes equations are:

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} = \mathcal{R}e^{-1}\frac{\partial \hat{S}}{\partial \eta}, \tag{2.10}$$

where the vector of conservative dependent flow variables is defined as:

$$\hat{Q} = J^{-1}\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, \tag{2.11}$$

and the transformation Jacobian is defined as:

$$J^{-1} = (x_\xi y_\eta - x_\eta y_\xi). \tag{2.12}$$

The inviscid fluxes are defined as:

$$\hat{E} = J^{-1}\begin{bmatrix} \rho U \\ \rho U u + \xi_x p \\ \rho U v + \xi_y p \\ (e+p)U - \xi_t p \end{bmatrix} \text{ and } \quad \hat{F} = J^{-1}\begin{bmatrix} \rho V \\ \rho V u + \eta_x p \\ \rho V v + \eta_y p \\ (e+p)V - \eta_t p \end{bmatrix}, \tag{2.13}$$

where the contravariant velocities are defined as:

$$U = \xi_t + \xi_x u + \xi_y v, \quad V = \eta_t + \eta_x u + \eta_y v, \tag{2.14}$$

while the viscous flux is:

$$\hat{S} = J^{-1}\begin{bmatrix} 0 \\ \eta_x m_1 + \eta_y m_2 \\ \eta_x m_2 + \eta_y m_3 \\ \eta_x(u m_1 + v m_3 + m_4) + \eta_y(u m_2 + v m_3 + m_5) \end{bmatrix}, \tag{2.15}$$

where

$$\begin{aligned} m_1 &= (\mu+\mu_t)(4\eta_x u_\eta - 2\eta_y v_\eta)/3, \\ m_2 &= (\mu+\mu_t)(\eta_y u_\eta + \eta_x v_\eta), \\ m_3 &= (\mu+\mu_t)(-2\eta_x u_\eta + 4\eta_y v_\eta)/3, \\ m_4 &= (\mu\mathcal{P}r^{-1} + \mu_t\mathcal{P}r_t^{-1})(\gamma-1)^{-1}\eta_x\partial_\eta(a^2), \\ m_5 &= (\mu\mathcal{P}r^{-1} + \mu_t\mathcal{P}r_t^{-1})(\gamma-1)^{-1}\eta_y\partial_\eta(a^2). \end{aligned} \tag{2.16}$$

2.4 Non-Dimensionalized Variables in Turbomachinery

TORNADO was originally used for solving external flows and logically used free stream quantities for non-dimensionalization. Turbomachinery flows do not have the luxury of free stream values, therefore TORNADO needs a different approach for turbomachinery. Most turbomachinery codes employ non-dimensionalization with some combination of inlet total quantities [5]. Giles [17, 18] used the total density and total speed of sound, while Michelassi and Martelli [31] used total pressure and total temperature. Total pressure and total temperature have been chosen on the basis that most experiments set these variables at the inlet.

The reference length within turbomachinery could also be defined differently as there are many more reference lengths within turbines and compressors. Niazi suggested using the diameter of the rotor or the blade trailing edge [36], while Michelassi and Martelli used the stator pitch [31]. Chord length has been implemented in TORNADO for commonality with the original code, since reference lengths can be arbitrarily chosen.

These methods have been implemented into the TORNADO code as follows:

$$x = \frac{\tilde{x}}{c}, \quad y = \frac{\tilde{y}}{c}, \quad \rho = \frac{\tilde{\rho}}{\tilde{\rho}_1^o}, \quad u = \frac{\tilde{u}}{\tilde{a}_1^o}, \quad v = \frac{\tilde{v}}{\tilde{a}_1^o}, \quad e = \frac{\tilde{e}}{\tilde{\rho}_1^o \tilde{a}_1^{o\,2}}. \tag{2.17}$$

Here the tilde represents a dimensional value. Throughout this thesis, values are assumed to be non-dimensional unless noted by tilde. As Chima points out, the Reynolds number, $\mathcal{R}e$, must be specified in the new non-dimensional terms [9] and is defined as follows:

$$\mathcal{R}e_1 = \frac{\rho_1^o\, a_1^o\, c}{\mu_1^o}. \tag{2.18}$$

This is unlike the original TORNADO that uses the freestream values. In the turbomachinery version of TORNADO, the inlet total values are used to calculate the Reynolds number. The total inlet dynamic viscosity μ_1^o is calculated by Sutherland's law

$$\mu_1^o = \mu_{ref}\left(\frac{T_1^o}{T_{ref}}\right)^{2/3}\frac{T_{ref}+S}{T_1^o+S} \tag{2.19}$$

where

$$T_{ref} = 273.15^oK, \quad S = 110.4^oK \quad and \quad \mu_{ref} = 1.716\cdot 10^{-5}\frac{kg}{ms}. \tag{2.20}$$

2.5 Turbulence Model

Turbulence is modeled using the Spalart-Allmaras one equation turbulence model [47]. Wilcox [50] states that the Spalart-Allmaras predictions are attractive for airfoil and wing applications for which TORNADO was originally created. The turbulence model has not been changed in this thesis. The Spalart-Allmaras model remains an efficient turbulence model in terms of robustness versus computational time. The Spalart-Allmaras model is as follows:

$$\begin{aligned}\frac{D\tilde{\nu}}{Dt} = \underbrace{\frac{c_{b1}}{Re}[1-f_{t2}]\tilde{S}\tilde{\nu}}_{\text{Production}} \quad &+ \quad \underbrace{\frac{1+c_{b2}}{\tilde{\sigma}Re}\nabla\cdot[(\nu+\tilde{\nu})\nabla\tilde{\nu}] - \frac{c_{b2}}{\tilde{\sigma}Re}(\nu+\tilde{\nu})\nabla^2\tilde{\nu}}_{\text{Advection and Diffusion}} \\ &- \quad \underbrace{\frac{1}{Re}\left[c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right]\left(\frac{\tilde{\nu}}{d}\right)^2}_{\text{Destruction}} + \underbrace{Ref_{t1}\Delta U^2}_{\text{Trip}}.\end{aligned} \tag{2.21}$$

As shown, the first term is the production term; the second and third are the combined advection/diffusion terms; the fourth is the destruction term and the last is a laminar/turbulent transition trip term. The trip functions f_{t1} and f_{t2} are set to zero for this thesis. The kinematic eddy viscosity, ν_t=μ_t/ρ, is calculated by solving the following equation:

$$\nu_t = \tilde{\nu}f_{v1} \tag{2.22}$$

with

$$f_{v1} = \frac{\chi^3}{\chi^3+c_{v1}^3} \tag{2.23}$$

and

$$\chi = \frac{\tilde{\nu}}{\nu}, \tag{2.24}$$

where the kinematic laminar viscosity, ν, is calculated by:

$$\nu = \mu/\rho. \tag{2.25}$$

The above production and destruction terms contain a modified vorticity term, $\tilde{S}$, which is calculated using the following equation:

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}. \tag{2.26}$$

In the above equation, the von Karman constant, κ, is equal to 0.41; the distance fuction, d, is calculated as the distance to the closest solid wall node. The magnitude of the vorticity, S, is given as:

$$S = \left[\left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \right)^2 + \left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right)^2 \right]^{\frac{1}{2}} \tag{2.27}$$

and the viscous function, f_{v2}, is:

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}. \tag{2.28}$$

The destruction term, f_w, defined as:

$$f_w = g \left(\frac{1 + c_{w3}^3}{g^6 + c_{w3}^6} \right)^{\frac{1}{6}} \tag{2.29}$$

with

$$g = r + c_{w2}(r^6 - r) \tag{2.30}$$

and the nondimensional variable, r, is defined as:

$$r \equiv \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}. \tag{2.31}$$

The transition functions are:

$$f_{t1} = c_{t1} g_t \exp\left(-c_{t2} \frac{w_t^2}{\Delta U^2} \left[d^2 + g_t^2 d_t^2 \right] \right) \tag{2.32}$$

and

$$f_{t2} = c_{t3} exp\left(-c_{t4} \chi^2 \right) \tag{2.33}$$

where

$$g_t = \min\left(0.1, \frac{\Delta U}{w_t \Delta x_t} \right), \tag{2.34}$$

and ΔU is the velocity difference between a field point and the trip point, d_t is the distance to the trip point, w_t is the vorticity at the trip point and Δx_t is the grid spacing at the trip piont.

The other parameters of the Spalart-Allmaras model are set as follows:

$$\begin{aligned} c_{b1} &= 0.1355, & c_{b2} &= 0.622, \\ c_{w1} &= c_{b1}/\kappa^2 + (1 + c_{b2})/\tilde{\sigma}, & c_{w2} &= 0.3, \\ c_{w3} &= 0.2, & c_{v1} &= 7.1, \\ \tilde{\sigma} &= 2/3, & \kappa &= 0.41, \\ c_{t1} &= 5.0, & c_{t2} &= 2.0, \\ c_{t3} &= 1.2, & c_{t4} &= 0.5. \end{aligned}$$

Modifications to the vorticity-like term $\tilde{S}$ are made within TORNADO to ensure a non-negative $\tilde{S}$ value using the following relation:

$$\tilde{S} = S f_{v3} + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \tag{2.35}$$

where the functions f_{v2} and f_{v3} are defined as:

$$f_{v2} = \left(1 + \frac{\chi}{c_{v2}}\right)^{-3} \tag{2.36}$$

$$f_{v3} = \frac{(1 + \chi f_{v1})\,(1 - f_{v2})}{\chi} \tag{2.37}$$

and

$$c_{v2} = 5.0 \tag{2.38}$$

is used to improve the stability of the tubulence model.

The distance function, d, contained in the production term needs special attention in cascade flows. The periodic boundary condition is an imaginary boundary since, in a real tubine or compressor, there is more than one blade. Periodic boundaries can cause a calculation error in the turbulence production when a point in the computed domain is actually closer to the non-computed blade rather than to the blade within the computed domain. The domain decomposition has been conducted in a manner in order to reduce the effect of this calculation error (Section 3.1 gives a full description of the domain topology). The periodic boundary is positioned roughly halfway between the computed and non computed blades. This reduces the number of points at which this error can occur. It can be assumed that the effect of error in d is negligible near the periodic

boundary since d is large near the periodic boundary and, by definition, the production of turbulence is near solid walls. From equation (2.21) it can be shown that:

$$\frac{D\tilde{\nu}}{Dt} \propto \tilde{S} \tag{2.39}$$

and equation (2.35) shows that:

$$\tilde{S} \propto \frac{1}{d^2}. \tag{2.40}$$

Therefore, the effect of erroneous values d on turbulence production is inversely proportional to d^2, and the effects will decrease exponentially as d increases. An exact calculation of d could have been performed, but it would not provide any added accuracy to the solution. However, care will have to be taken to ensure that the periodic boundary does not move far from the mid-plane of the blades.

2.6 Boundary Conditions

Boundary conditions need to be treated differently in turbomachinery than in airfoil flow since the computational boundaries are much closer to the blade/foil. Since the far edge of the computational grid is close to the blade (about 0.5-2 chord lengths), any reflective or non physical nature of an imposed boundary condition will have an effect on the flow solution. Giles proposed nonreflecting boundary conditions [17] for turbomachinery that have been widely implemented into numerous flow solvers. As Giles has done, the theory of characteristics must be addressed first to ensure that the boundary conditons do not impose any reflections into the solution.

2.6.1 Method of Characteristics

The method of characteristics provides us with four basic characteristic variables for solving one-dimensional, isentropic, nonstationary flows of perfect gases. The first two variables are entropy and pitchwise velocity, which are related to vorticity. The two other variables are the upstream and downstream running pressure waves as shown below [18]:

$$R^+ = u + \frac{2a}{\gamma - 1} \tag{2.41}$$

and

$$R^- = u - \frac{2a}{\gamma - 1}. \tag{2.42}$$

2.6.2 Inflow Boundary

For the purpose of this thesis, all flows are considered subsonic and so the inflow boundary must take into account the possibility of R^- leaving through the inlet face. Most turbomachinery solvers set some combination of total values at the inlet as well as an inlet flow angle or inlet velocity ratio [4, 49, 7, 8, 9, 12, 22, 25, 30, 28, 29, 17, 31, 32, 36, 43].

In this thesis, total pressure, P_1^o, total temperature, T_1^o, as well as the inlet angle, α, are set at the inlet face to match the values given in the experimental validation work by Jouini [26]. The turbulence model working variable, $\tilde{\nu}$, is set at the inlet face to a value of 0.001 for all cases in this thesis.

In order to solve for the four conservative variables and to model the acoustic disturbances leaving the computational field through the inlet face, the downstream running Riemann invariant, R_{ext}^-, is extrapolated from the interior [36].

Holmes [25] has stated that simple zeroth-order extrapolation is sufficient for similar inlet boundary extrapolations. Since introducing a simple logic in the inlet is computationally inexpensive, zeroth-order and second-order extrapolation are used. Zeroth-order extrapolation is used in the first 25 iterations or while the solution residual is above 10^{-6} which ever is longer. This is done to ensure that the solution does not diverge. When the solution has become more stable, a second-order extrapolation is used.

Using the user specified P_1^o, T_1^o and α and the extrapolated R_{ext}^-, the axial velocity is calculated as follows:

$$u_1 = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \tag{2.43}$$

This poses a problem since TORNADO uses only x and y coordinates. where

$$A = 1 + 2\,(1 + (\tan(\alpha))^2)(\gamma - 1), \tag{2.44}$$

$$B = -2R_{ext}^- \tag{2.45}$$

and

$$C = (R_{ext}^-)^2 - 4\,\gamma\,T_1^o\,(\gamma - 1)^2. \tag{2.46}$$

The pitchwise velocity is calculated using the specified inlet angle, α,

$$v_1 = u_1\,\tan(\alpha). \tag{2.47}$$

Assuming an ideal gas, the inlet temperature can be calculated using the following relation:

$$T_1 = T_1^o - \frac{(u_1^2 + v_1^2)(\gamma - 1)}{2\,\gamma}. \tag{2.48}$$

This allows us to calculate the remaining inlet conservative variables as follows:

$$\rho_1 = \frac{P_1^o}{T_1 + \frac{\sqrt{v_1^2+u_1^2}}{2}} \tag{2.49}$$

and

$$e_1 = \rho\left(T_1(\gamma-1) + \frac{(v_1^2+u_1^2)}{2}\right). \tag{2.50}$$

2.6.3 Outflow Boundary

At the outlet boundary, there are three characteristics leaving the domain, so only the outlet static pressure, P_2, which corresponds to the desired isentropic Mach number, is prescribed by the user. This practice is used by many different turbomachinery codes and is also convenient for validation against experimental results [4, 49, 7, 8, 9, 12, 22, 25, 30, 28, 29, 17, 31, 32, 36, 43].

The three characteristic values entropy (H_{ext}), pitchwise velocity (v_{ext}) and upstream running Reiman variable (R^+_{ext}) are extrapolated using the same technique as the inlet boundary condition. The turbulence model working variable, $\tilde{\nu}$, is also extrapolated from the interior.

Using the specified pressure, extrapolated values and the following relation:

$$H = \frac{e+P}{\rho}, \tag{2.51}$$

it can be shown that

$$\rho_2 = \frac{1}{B}\left(A - \frac{C(\sqrt{C^2+4AB}-C)}{2B}\right) \tag{2.52}$$

where

$$A = P_2\frac{\gamma(\gamma+1)}{(\gamma-1)^2}, \tag{2.53}$$

$$B = H_{ext} - \frac{1}{2}\left((R^+_{ext})^2 - v^2_{ext}\right) \tag{2.54}$$

and

$$C = \frac{2\,R^+_{ext}\sqrt{\gamma p_2}}{(\gamma-1)}; \tag{2.55}$$

the exit axial velocity can then be calculated using

$$u_2 = R^+_{ext} - \frac{2\sqrt{\frac{\gamma * P_2}{\rho_2}}}{\gamma+1}. \tag{2.56}$$

Since the pitchwise velocity is extrapolated from the interior, the final conservative variable is calculated

$$e_2 = H_{ext}\rho_2 - P_2. \tag{2.57}$$

2.6.4 Block Interface Boundary

Block interface boundaries were originally handled in two different ways in TORNADO. There is an average boundary condition for wake cuts (ξ direction) and a transparent boundary condition for normal (η direction) interfaces. The wake cut boundary would average the common points and copy the two columns of halo points from the adjacent domain. The transparent boundary interface condition simply copied the two rows of values into the adjacent block's two halo rows.

In the case of turbomachinery, it has been said that errors introduced at the leading edge can be transported downstream on the blade surface, giving poor results [4]. In order to reduce any averaging errors, the transparent boundary conditions were introduced for the streamwise interface boundaries.

2.6.5 Periodic Boundary

Periodic boundary conditions were introduced for the first time into TORNADO in this thesis. Flow within turbomachinery could be calculated by using a number of blades in a row as most experiments for turbomachinery do, but introducing a periodic boundary condition reduces the computational cost considerably. Periodic boundary conditions simulate the physics of a cascade without solving the flow around more than one blade. Gostelow [21] defines a cascade as an infinite row of equidistant similar bodies.

Periodic boundaries have been implemented into TORNADO using a similar approach to transparent block boundary conditions. The halo row values are simply copied from the adjacent block's interior points. The one exception is that the adjacent block is at the opposite side of the control volume. Meaning that the halo points share the conservative variables with the corresponding points within the control volume, but have a different physical point in space.

Although this was simple in theory, numerous subroutines and functions needed to be altered in order to implement periodic boundary conditions.

2.6.6 Wall Boundary

For the purpose of this thesis, all flow test cases were run with viscous wall boundary conditions, since turbomachinery flows are highly dependent on viscous effects. The

viscous wall boundary conditions are implemented by first setting a no-slip condition

$$u = 0, \quad v = 0, \quad w = 0 \ . \tag{2.58}$$

The pressure is then extrapolated using

$$\left(\frac{\partial p}{\partial n}\right)_{wall} = 0. \tag{2.59}$$

The final conditions assume that the there is an adiabatic wall condition

$$\left(\frac{\partial T}{\partial n}\right)_{wall} = 0 \tag{2.60}$$

and a perfect gas

$$P = \rho RT; \tag{2.61}$$

and is implemented similar to the pressure extrapolation 2.59 as follows:

$$\left(\frac{\partial \rho}{\partial n}\right)_{wall} = 0. \tag{2.62}$$

For the Spalart-Allmaras turbulence model, the turbulent eddy viscosity is zero at the wall:

$$\tilde{\nu} = 0. \tag{2.63}$$

Chapter 3

Grid Generation

The UTIAS CFD lab has been using the Automated MultiBlock Elliptic gRid generator, AMBER2D, to create 2D multiblock structured grids for single and multi-element airfoils. This thesis has altered the original AMBER2D code in order to create a 2D multiblock cascade structured grid generator [33].

3.1 Grid Topology

Since rotor and stator blades are relatively close to each other in space, they must be modeled as such. The effects of a neighboring blade can either be modeled by including numerous blades in the flow domain or by using cascade grids. Cascade grids have the same geometry on the upper and lower surfaces. When periodic boundary conditions are used on these surfaces, the effects of the neighboring blades are taken into account. In order to reduce the domain size and, ultimately, the flow solution time, cascade grids with periodic boundaries will be used in this thesis.

There are several different types of structured grid topologies that can be used in modeling turbine and compressor blades. The simplest is an O-topology (see Figure 3.1 for a sample O-grid from Choi and Yoo). The grid lines are augmented from the blade's shape near the blade to an oval at the far field. The grid orthogonality at the leading and trailing edges is very desirable for viscous calculations [29]. While the skewness at the inlet, outlet and periodic boundaries causes poor flow results [1], Kiss et al. have also pointed out that the resolution upstream of the blade may be more dense than required when an increase in O-grid lines is needed to resolve the shocks in the wake region [27].

A C-topology is a widely used topology for single element airfoils (see Figure 3.2 for

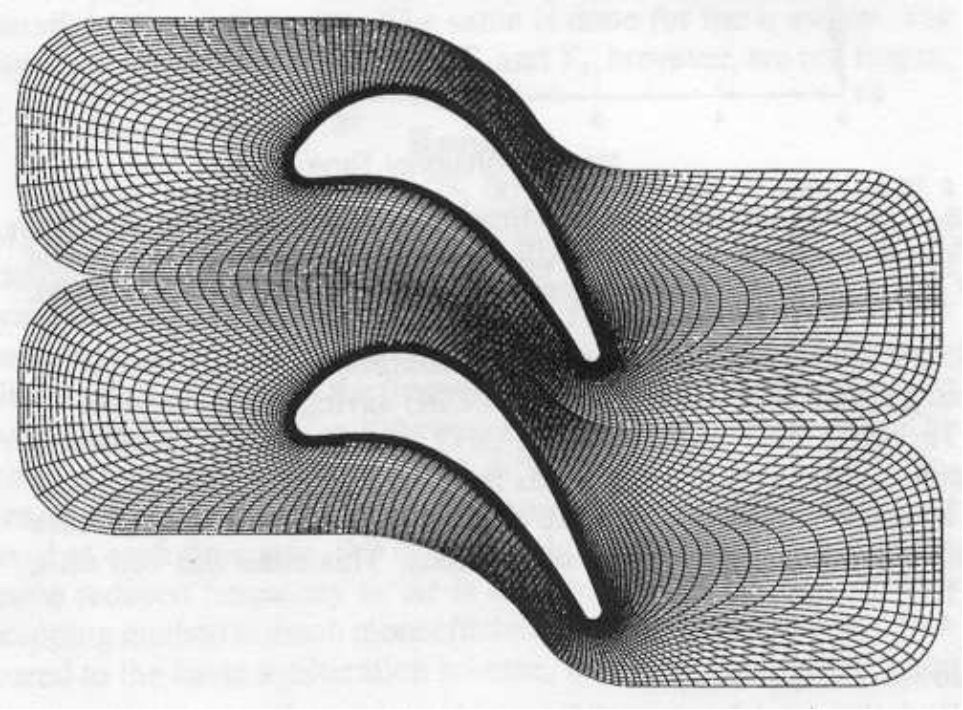

Figure 3.1: O-topology Grid [10]

a sample C-grid from Kiss, Schetz and Moses). The flow area is divided up into three blocks for sharp trailing edges and four blocks for blunt trailing edges. C-grids have good orthogonality at the leading edge and the outlet boundary, but the trailing edge, periodic and inlet boundary have lower orthogonality [1]. An advantage C-type grids have over the O-type grid is that the wake region can be made more dense to resolve shocks without increasing grid density in the low pressure gradient regions.

H-topology grids (see Figure 3.3) are common in turbomachinery applications [3]. They are relatively easy to construct and are easily applied to 3-D geometries [25]. H-grids have numerous advantages, including post processing ease and applicability to a wide range of blade shapes [28]. The numerous blocks of H-grids allow for optimized choices of local grid spacing for faster yet accurate flow calculations. H-grids also have no orthogonality problems like O- and C-type grids in the inlet and outlet boundaries. H-topology grids will be used in this thesis.

All of the mentioned grid types are structured. There are some flow solvers that use unstructured grids [20]. Structured grids may take more time to create, but save on flow solver processing time and are easily applied to multiprocessor computing [1]. This thesis will use structured grids.

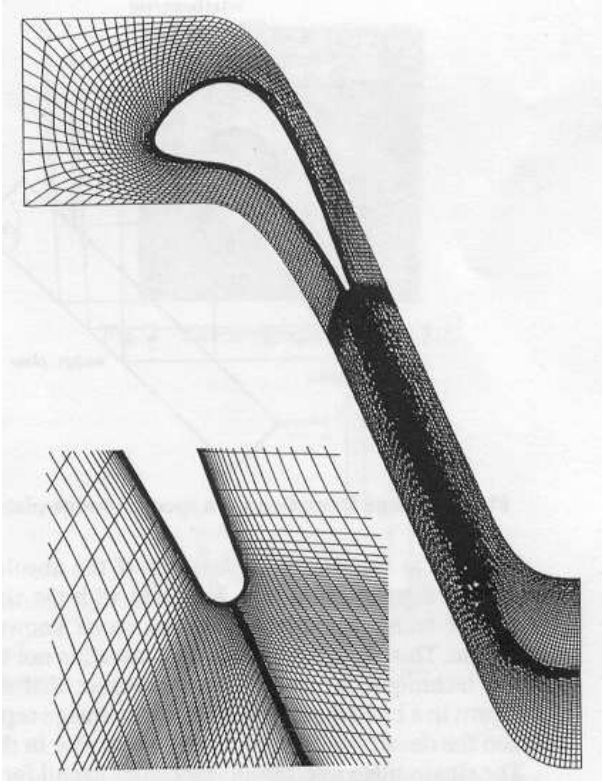

Figure 3.2: C-Topology Grid [27]

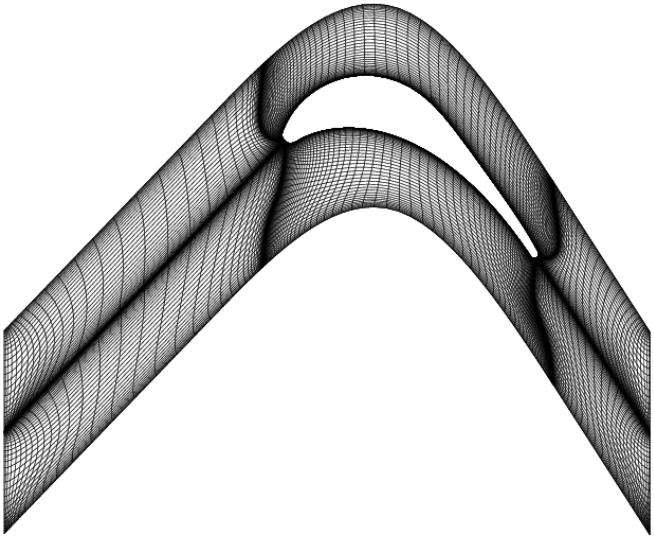

Figure 3.3: H-Topology, HS1A

Other hybrid [22] and zonal [32] meshing methods exist, but an elementary H-grid was chosen to enhance the robustness of the eventual turbomachinery tool, as these meshing techniques are harder to apply to a generic foil shape.

3.2 Block Boundaries

The first module of AMBER2D is kmS, which creates the grid block boundaries. kmS has been rewritten to create the eleven splines that define the cascade block boundaries.

The splines used in kmS are natural cubic splines. Cubic splines comprise I piecewise continuous third-order polynomial curves that connect $I+1$ control points:

$$X_I(t) = {A_I}^x + {B_I}^x t + {C_I}^x t^2 + {D_I}^x t^3 \tag{3.1}$$

and

$$Y_I(t) = {A_I}^y + {B_I}^y t + {C_I}^y t^2 + {D_I}^y t^3 \tag{3.2}$$

where

$$0 \leq t \geq 1. \tag{3.3}$$

In order to solve for the coefficients A_I, B_I, C_I and D_I, the coordinate value, first derivative and second derivative need to be equivalant for two spline sections at any interior control point. The end point coordinates, first derivative and second derivative have to be specified in order to solve fully for the A_I, B_I, C_I and D_I coefficients.

The following subsections describe how the $I+1$ control points for each of the eleven block boundary lines are formed, as shown in Figure 3.4.

3.2.1 Foil Line

The first line (Line 1) that is created in the kmS code is the foil line. It begins at at the trailing edge and traces the foil geometry clockwise through the stagnation point and ends at the trailing edge. This is the same line convention that was used in the original kmS.

3.2.2 Ψ Lines

Line 6 is the inlet Ψ line. It originates at the grid stagnation point and goes upstream to the grid boundary (user defined X value XLB) at the inlet angle ANG_OUT (user defined).

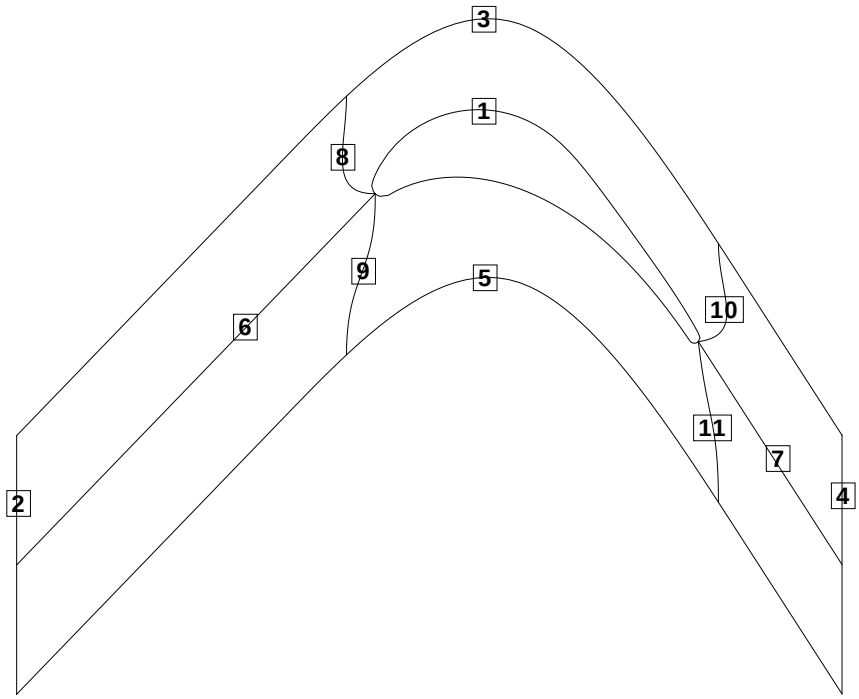

Figure 3.4: Block Boundary Spline Numbering

The grid stagnation point is determined by finding the point on the foil where the surface tangent is perpendicular to the user defined inlet angle (ANG_OUT). This is not the true stagnation point, but it is used by kmS to create the meshing stagnation point. Since the foil could have more then one point where the surface tangent is perpendicular to the inlet angle, the user can define the points at which the software searches for the stagnation point by setting $STAG\_MINS$ and $STAG\_MAXS$. These are the foil spline lengths (s=0 and s=1 for the wake cut point). The default values of $STAG\_MAXS = 0.66$ and $STAG\_MINS = 0.33$ work well for foils with low to medium curvature.

Line 7 is the outlet Ψ line. It originates from the wake cut point and goes downstream to the grid boundary (user defined X value XRB) at the outlet angle ANG_OUT (user defined). The wake cut point is then determined by the airfoil coordinate file structure (the file starts and ends with this point). If the user wishes to use a different point, the coordinate file can be manually altered.

3.2.3 Inlet and Outlet Faces

Line 2 and 4 are the inlet and outlet faces. These block boundaries are set by the user defined variables XLB and XRB. The center of the boundary is positioned at the end of the respective Ψ lines. The total height of the inlet and outlet faces are set using the $BLSP$ (blade spacing) variable.

3.2.4 Periodic Faces

Line 3 and 5 are the upper and lower periodic faces respectively. The two faces have the same geometry to ensure a seamless periodic boundary condition. The periodic boundaries are constructed from five different lines: inlet Ψ, inlet pseudocam, camber line, outlet pseudocam and outlet Ψ as described below. The resultant line is then translated up and down one half of the blade spacing, creating the upper and lower periodic faces.

Camber Line

The center portion of a periodic face is the camber line from the stagnation point to the trailing edge, as defined by the foil coordinate file. The Y values of the camber line are calculated by averaging the upper and lower surfaces at a given X value.

Pseudocam Lines

The pseudocam lines are created to smooth the transition between the camber line and the Ψ lines. The pseudocam lines start from their respective ends of the camber line and smooth a portion of the Ψ line defined by the user by using $SPOINTS$. $SPOINTS$ is defined as half of the percentage of the Ψ line that the user wants to smooth. For example, if the user wants to smooth 30% of the Ψ line, they would set $SPOINTS$ to 15. The pseudocam X values are set as a number of equidistant points between the two defined end points. The smoothing of the Y values is performed by varying the gradient along the pseudocam using the following relation:

$$Y = Y_1'X + (3 - Y_1' - 2 * Y_2')X^2 + (Y_1' + Y_2' - 2)X^3 \tag{3.4}$$

where X varies from $0 - 1$ for the distance of the psudocam, Y_1' is gradient at the camber line and Y_2' is the gradient at the Ψ line.

3.2.5 Φ Lines

The Φ lines for the upper and lower stagnation point are lines 8 and 9 while the upper and lower trailing edge Φ lines are 10 and 11. The Φ lines are created using cubic Bezier curves. Bezier curves are chosen in order to provide a line that does not create an invalid grid (negative cell sizes). It also provides a smooth, continuous line that aids in flow solver convergence.

Any points on a cubic Bezier curve can be determined by using the following equations:

$$X(s) = (1-s)^3X_0 + 3s(1-s)^2X_1 + 1s^2(1-s)X_2 + s^3X_3 \tag{3.5}$$

and

$$Y(s) = (1-s)^3Y_0 + 3s(1-s)^2Y_1 + 1s^2(1-s)Y_2 + s^3Y_3 \tag{3.6}$$

where s varies from 0 to 1 for the length of the curve. The four control points are the starting point (X_1, Y_1), starting control point (X_2, Y_2), end control point (X_3, Y_3) and the end point (X_4, Y_4). The starting point (X_1, Y_1) for the stagnation point Φ line is the stagnation point while the trailing edge point is the starting point for the trailing edge Φ lines.

The end point (Y_3,X_3) can be influenced by the user by setting $Prcnt\_IN$ and $Prcnt\_OUT$ in the input file. The Φ line end points are set to be the point on the

respective periodic boundary where:

$$X_3 = X_1 + (X_1 - X_{bound})PRCNT \tag{3.7}$$

where X_{bound} is the X value of the respective inlet or outlet boundary, X_0 is the above Φ line starting point and $PRCNT$ is the user defined $Prcnt\_IN$ or $Prcnt\_OUT$.

For simplicity, consider the starting and end control points as vectors originating from their respective end points. The start control vector goes from (X_1, Y_1) and ends at (X_2, Y_2) and the end control vector starts at (X_4, Y_4) and ends at (X_3, Y_3).

In order to ensure grid continuity across the periodic boundary condition, the perpendicular grid lines at each periodic face must have the same slope. To achieve this, the gradient at the Φ end points are set to be vertical. Since the inlet and outlet faces are also vertical, all the perpendicular grid lines will have a vertical gradient along the periodic faces. This is achieved by having the end control vector vertical, $X_3 = X_4$.

The length of the end control vector has ramifications on the shape of the Φ line and the grid quality. If the end control vector is shorter, the Φ line will have more gradual bends. This would reduce the skewness of grid points near the Φ boundary, increasing the accuracy and speed of the flow solve. If the end control vector is longer, the end of the Φ line will have a longer vertical region; increasing the vertical points farther away from the periodic boundary. This increases orthogonality of the grid points near the periodic boundary condition, but also increases the accuracy of the periodic boundary condition in the flow solve. To balance these opposing benefits, the end control vector length is set to be half the Y distance between the starting and end control points, $Y_3 = (Y_1 + Y_4)/2$ for each Φ line.

The angle of the start control vector has been chosen to reduce the skewness of the grid near the trailing and stagnation points. The trailing edge Φ line start control vector bisects the angle between the respective trailing edge surface gradient and the outlet Ψ line starting gradient. The stagnation point Φ line start control vector bisects the angle between the stagnation point surface gradient and the inlet Ψ line starting gradient.

The length of the start control vector also has ramifications on the shape of the Φ line and the grid quality. If the start control vector is short, then the Φ line would have high curvature near the starting point. The high curvature would produce a skewed grid near the foil surface; either at the stagnation point or the trailing edge. Since these points have high pressure gradients, the grid quality at these points has a significant impact on the accuracy of the entire flow solution. If the start control vector is too long, the

resultant Φ line could have a loop in it. This would cause negative grid cells, causing an invalid grid that is insolvable. To balance these opposing problems, the starting control vector is set to half the Y distance between the starting and end control points as follows:

$$X_2 = X_1 + \frac{|\Delta X| \cos(X_1')}{2} \tag{3.8}$$

and

$$Y_2 = Y_1 + \frac{|\Delta Y| \sin(Y_1')}{2}, \tag{3.9}$$

where ΔX and ΔY are the X and Y distances between the starting and end control points and where X_1' and Y_1' are the X and Y start control vector gradients.

3.3 Grid Generation

After the block boundaries are formed, the grid points are created using Agrid, an algebraic grid generator. This code was developed by T. Nelson and A. Wilkinson in 1992. The original Agrid code has been successfully used to create cascade grids with no alterations. Agrid creates a rough grid by interpolating interior points using transfinite interpolation. For more information on the methods used in Agrid, see T. Nelson's PhD thesis [33].

3.4 Elliptical Grid Smoothing

Smooth continuous grids are essential for accurate flow solutions. The elliptical grid smoother in Megrid is implemented for cascades with no alterations. Megrid uses Soreson's method to control the smoothing and general appearance of the grid [51]. For more details on how the method is executed within Megrid, refer to T. Nelson's PhD thesis [33].

3.5 Periodic Boundary Treatment

In order to have accurate flow solutions, all of the periodic boundaries must have identical point distributions, unless an averaging method is implemented. Halama, Arts and Fort have shown that an averaging method can be implemented to overcome the nonsimilar grid point structures on joining periodic faces [22]. Since the aim of this thesis is to

work towards a design tool for turbomachinery flow, solve speed is critical. Halama et al.'s algorithm for averaging nonsimilar periodic boundaries would increase flow solution convergence time. Therefore, grids will be created with each periodic face having the same geometry, number of points and grid spacing.

In order to ensure that the grid points are sill equal, a post grid smoothing program is utilized similar to what Cravero and Satta discuss [12]. After creating a grid using the original AMBER2D, a user would use check_grid to ensure that the grid points on the interior block boundaries are equal to double precision. This program was altered to include a check of the periodic boundary points and renamed check_cas. An extra input file is needed to communicate the blade spacing to the program.

Chapter 4

Numerical Algorithm

The numerical methods of TORNADO were altered in order to solve for turbomachinery cases. TORNADO has two methods for solving the governing equations, Approximate Factorization (AF) and Newton-Krylov (NK). Even though the NK solver has been proven to be much faster at converging on a solution, only the AF solver will be altered for turbomachinery flows for this thesis. The current TORNADO code uses Approximate Factorization for start up, so in order to focus the scope of this thesis, only AF has been altered for turbomachinery flows as shown in the following sections.

4.1 Multi-Block Approximate Factorization

In order to solve the steady two-dimensional thin-layer Navier-Stokes equations, a first-order implicit Euler method is used within TORNADO [13]. When one applies the Euler time-marching scheme to the thin layer Navier-Stokes equations 2.10, the following relation is formed:

$$\Delta\hat{Q}^n + \Delta t \left(\partial_\xi \hat{E}^{n+1} + \partial_\eta \hat{F}^{n+1} - \mathcal{R}e^{-1} \partial_\eta \hat{S}^{n+1} \right) = 0 \tag{4.1}$$

where Δt is the time step and

$$\Delta\hat{Q} = \hat{Q}^{n+1} - \hat{Q}^n \tag{4.2}$$

when

$$\hat{Q}^n = \hat{Q}(n\Delta t). \tag{4.3}$$

The vectors $\hat{E}$, $\hat{F}$, and $\hat{S}$ are then locally linearized as follows:

$$\begin{aligned} \hat{E}^{n+1} &= \hat{E}^n + \hat{A}^n \Delta\hat{Q}^n + O(\Delta t^2) \\ \hat{F}^{n+1} &= \hat{F}^n + \hat{B}^n \Delta\hat{Q}^n + O(\Delta t^2) \\ \hat{S}^{n+1} &= \hat{S}^n + \hat{K}^n \Delta\hat{Q}^n + O(\Delta t^2), \end{aligned} \tag{4.4}$$

where the matrices $\hat{A}$, $\hat{B}$, and $\hat{K}$ are the flux Jacobians as defined below:

$$\hat{A} = \frac{\partial \hat{E}}{\partial \hat{Q}}, \quad \hat{B} = \frac{\partial \hat{F}}{\partial \hat{Q}}, \quad \text{and} \quad \hat{K} = \frac{\partial \hat{S}}{\partial \hat{Q}}.$$

When Equations 4.1 and 4.4 are combined, the following relation is arived at:

$$[I + \Delta t \partial_\xi \hat{A}^n + \Delta t \partial_\eta \hat{B}^n - \Delta t \mathcal{R}e^{-1} \partial_\eta \hat{K}^n] \Delta\hat{Q}^n = \hat{R}^n \tag{4.5}$$

where

$$\hat{R}^n = -\Delta t [\partial_\xi \hat{E}(\hat{Q}^n) + \partial_\eta \hat{F}(\hat{Q}^n) - \mathcal{R}e^{-1} \partial_\eta \hat{S}(\hat{Q}^n)].$$

Two additional approximations have been made to the left hand side (implicit operator) of equation 4.5 in order to reduce computational time. When the spacial differencing and approximate-factorization method of Beem and Warming [6] are applied to 4.5, the following relation is formed:

$$[I + \Delta t \delta_\xi \hat{A}^n][I + \Delta t \delta_\eta \hat{B}^n - \Delta t \mathcal{R}e^{-1} \delta_\eta \hat{K}^n] \Delta\hat{Q}^n = \hat{R}^n, \tag{4.6}$$

where

$$\hat{R}^n = -\Delta t [\delta_\xi \hat{E}(\hat{Q}^n) + \delta_\eta \hat{F}(\hat{Q}^n) - \mathcal{R}e^{-1} \delta_\eta \hat{S}(\hat{Q}^n)],$$

and the symbol δ is a central differenced spatial operator.

To further reduce the complexity of the implicit operator, the diagonal form of Pulliam and Chaussee [42] is utilized. The resultant diagonalized Jacobian matrices are:

$$\Lambda_\xi = T_\xi^{-1} \hat{A} T_\xi \tag{4.7}$$

and

$$\Lambda_\eta = T_\eta^{-1} \hat{B} T_\eta, \tag{4.8}$$

where the diagonal matrices Λ_ξ and Λ_η have elements that are the eigenvalues of the flux Jacobians. Since the flux Jacobian $\hat{B}$ and viscous flux Jacobian $\hat{K}$ cannot be simultaneously diagonalized, the viscous flux Jacobian, $\hat{K}$, has been dropped from the left-hand

side. In order to do this, a term approximating the viscous eigenvalues is added to Λ_η. Since the matrix T_ξ has the eigenvectors of $\hat{A}$ as columns and T_η has the eigenvectors of $\hat{B}$ as columns, the eigenvector matrices are factored out. The resulting system of equations can be shown as:

$$T_\xi[I + \Delta t\delta_\xi\Lambda_\xi]T_\xi^{-1}T_\eta[I + \Delta t\delta_\eta\Lambda_\eta - \Delta tI\delta_\eta(\lambda_\nu)]T_\eta^{-1}\Delta\hat{Q}^n = \hat{R}^n, \tag{4.9}$$

where,

$$\lambda_\nu = \frac{\gamma\mu}{\rho Pr\mathcal{R}e}(\eta_x^2 + \eta_y^2) \tag{4.10}$$

is used to approximate the viscous eigenvalues.

Variable time stepping is used within TORNADO to achieve an accelerated convergence rate. Since grids can have widely varying cell sizes, the Courant number would vary within the grid. In order to make the Courant number more uniform, it is scaled with the Jacobian as follows:

$$\Delta t = \frac{\Delta t_{ref}}{1 + \sqrt{J}}. \tag{4.11}$$

4.2 Numerical Dissipation

Artificial dissipation needs to be implemented in order to maintain stability of the flow solve. The nonlinear interactions in the convection terms of the momentum equations can cascade causing numerical instability in flows with shocks [41]. The matrix dissipation scheme of Swanson and Turkel [48] is used in TORNADO to provide numerical dissipation as follows:

$$\left(\frac{\partial\hat{E}}{\partial\xi}\right)_{j,k} = \delta_\xi\hat{E}_{j,k} - \Delta_\xi d_{j+\frac{1}{2},k} \tag{4.12}$$

with

$$\begin{aligned}
d_{j+\frac{1}{2},k} &= |\hat{A}|_{j+\frac{1}{2},k}\, J^{-1}_{j+\frac{1}{2},k}\left(\epsilon^{(2)}_{j+\frac{1}{2},k}\,\Delta_\xi\, J_{j,k}\,\hat{Q}_{j.k} - \epsilon^{(4)}_{j+\frac{1}{2},k}\,\Delta_\xi\nabla_\xi\Delta_\xi\, J_{j,k}\,\hat{Q}_{j,k}\right)\\
\epsilon^{(2)}_{j,k} &= \kappa_2\max(\Upsilon_{j+1,k}, \Upsilon_{j,k}, \Upsilon_{j-1,k})\\
\epsilon^{(4)}_{j,k} &= \max(0, \kappa_4 - \epsilon^{(2)}_{j,k})\\
\Upsilon_{j,k} &= \frac{|p_{j+1,k} - 2p_{j,k} + p_{j-1,k}|}{|p_{j+1,k} + 2p_{j,k} + p_{j-1,k}|},
\end{aligned}$$

where $\Upsilon_{j,k}$ is a pressure switch to control the use of first-order dissipation near shock waves, δ_ξ is the centered difference operator, Δ_ξ is the first-order forward-difference operator, and ∇_ξ is the backward difference operator. We use $\kappa_4 = 0.02$ and $\kappa_2 = 1.0$ since

this thesis deals with tubulent flows only. The matrix $|\hat{A}|$ is defined as:

$$|\hat{A}| = T_\xi \, |\Lambda_\xi| \, T_\xi^{-1}. \tag{4.13}$$

The term $|\hat{A}|_{j+\frac{1}{2},k}$ is evaluated using a simple average:

$$|\hat{A}|_{j+\frac{1}{2},k} = \frac{1}{2}[\,|\hat{A}|_{j,k} + |\hat{A}|_{j+1,k}\,]. \tag{4.14}$$

The Roe average is recommended for flows containing very strong shock waves. The matrix $|\Lambda_\xi|$ is comprised of the eigenvalues of the flux Jacobian matrix $\hat{A} = \frac{\partial \hat{E}}{\partial \xi}$, as shown here:

$$|\Lambda_\xi| = \begin{bmatrix} |\lambda_1| & 0 & 0 & 0 \\ 0 & |\lambda_2| & 0 & 0 \\ 0 & 0 & |\lambda_3| & 0 \\ 0 & 0 & 0 & |\lambda_4| \end{bmatrix} = \begin{bmatrix} |U| & 0 & 0 & 0 \\ 0 & |U| & 0 & 0 \\ 0 & 0 & |U + a\,\theta| & 0 \\ 0 & 0 & 0 & |U - a\,\theta| \end{bmatrix}, \tag{4.15}$$

where $\theta = \sqrt{\xi_x^2 + \xi_y^2}$, U is the contravariant velocity component in the ξ direction, ξ_x and ξ_y are the metrics of the curvilinear coordinate transformation and a is the speed of sound.

The matrix T_ξ also contains the right eigenvectors of $\hat{A}$. The elements of $|\Lambda|_\xi$ are modified to avoid zero eigenvalues as follows:

$$\begin{aligned} \tilde{\lambda}_1, \tilde{\lambda}_2 &= \max(\lambda_{1,2}, V_l\,\sigma) \\ \tilde{\lambda}_3 &= \max(\lambda_3, V_n\,\sigma) \\ \tilde{\lambda}_4 &= \max(\lambda_4, V_n\,\sigma), \end{aligned} \tag{4.16}$$

where σ is the spectral radius of the flux Jacobian, $V_l = 0.025$ and $V_n = 0.25$.

4.3 Numerical Method Changes

In order to solve turbomachinery cases the AF algorithm of TORNADO was altered in two different ways: non-dimensionalization changes and block boundary updates. The non-dimensionalization changes, as described in Section 2.4, were implemented throughout the code wherever references were made to the Reynolds number or free stream variables. The periodic and interface boundary conditions, as described in Sections 2.6.4 and 2.6.5, are updated explicitly at the end of each approximate factorization iteration.

Chapter 5

Results and Validation

All results in this section were computed on the University of Torontos High Performance Aerospace Computing Facility (HPACF), using the Hewlett-Packard ES45 AlphaServers and 1000MHz EV68CB Alpha Processors.

5.1 Turbomachinery Performance Validation

In order to validate the results of the modified TORNADO flow solver, turbomachinery specific performance coefficients have been implemented. Isentropic Mach Number and the Coefficient of Static Pressure are standard turbomachinery variables that were also used by Jouini [26] to communicate experimental results.

5.1.1 Coefficient of Static Pressure

The Coefficient of Static Pressure is like the turbomachinery equivalent of the coefficient of pressure used in aerodynamics, the difference being that the reference pressure is the upstream static pressure instead of free stream pressure. The coefficient of static pressure, C_{ps}, is calculated as follows:

$$C_{ps} = \frac{P^s - P_1}{q_1}, \tag{5.1}$$

where P^s is the static pressure where the coefficient of static pressure is being calculated, P_1 is the static pressure at the upstream boundary, and q_1 is the dynamic pressure at the upstream boundary.

Mesh	Number of Nodes	Off-Wall Spacing ($10^{-6}c_x$)	Far-Field Spacing ($10^{-3}c_x$)	LE Spacing ($10^{-4}c_x$)	TE Spacing ($10^{-4}c_x$)	Nodes to Far-Field	Nodes on Foil Surface
Coarse	5670	4.00	4.00	4.00	4.00	27	101
Medium	21942	2.00	2.00	2.00	2.00	53	201
Fine	86310	1.00	1.00	1.00	1.00	105	401

Table 5.1: Grid Convergence: Mesh Parameters

5.1.2 Isentropic Mach Number

The isentropic Mach number is the Mach number that would be expected if the flow was considered reversible, i.e. experienced no losses. This is widely used in turbomachinery to depict the flow characteristics around a turbomachinery blade. Isentropic Mach Number, M_{is}, is calculated as follows:

$$M_{is} = \sqrt{\left(\frac{2}{\gamma - 1}\right)\left(\left(\frac{P_1^0}{P^s}\right)^{\frac{\gamma-1}{\gamma}} - 1\right)} \tag{5.2}$$

where P^s is the static pressure where the Isentropic Mach Number is being calculated, P_1^0 is the total pressure at the upstream boundary and γ is the ratio for specific heats. $\gamma = 1.4$ is used for air.

5.2 Grid Convergence Study

A grid convergence study was conducted. A test case from Jouini [26] was used as the experimental comparison. The HS1A Ma=0.71 angle of incidence of 0° was used. This test case was chosen because of the low Mach number and little separated flow. The fine grid, as defined in Table 5.1, was created first. The medium and coarse meshes were created by removing every other node. The resulting grids are depicted in Figures 5.1, 5.2 and 5.3.

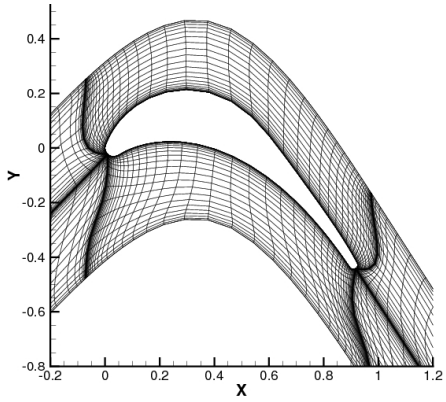

Figure 5.1: Grid Convergence: Coarse Mesh

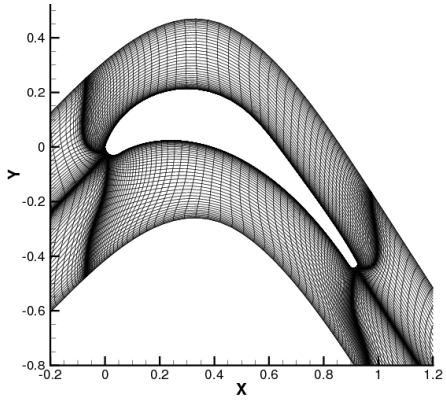

Figure 5.2: Grid Convergence: Medium Mesh

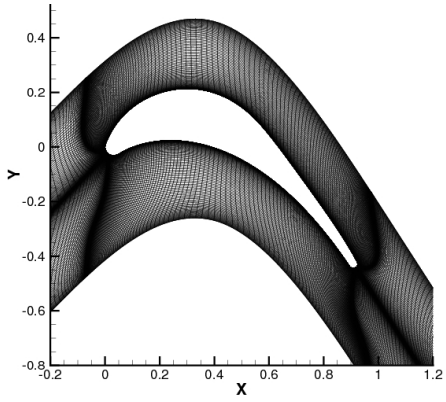

Figure 5.3: Grid Convergence: Fine Mesh

The convergence time and number of iterations are shown in Figures 5.4, 5.5 and 5.6. The convergence time of the fine grid was so long that a restart was necessary due to HPACF process limitations. It is noted that the fine mesh took 290,000 seconds or 80.5 hours and 4.5 million iterations to converge. The medium mesh took 21,600 seconds or 6 hours and 350,000 iterations. The coarse mesh took 413 seconds or 7 minutes and 30,000 iterations.

Figure 5.8 and 5.9 show the coefficient of static pressure and isentropic Mach number plots against the experimental results of the grid convergence study. It can be seen that the resolution of the pressure gradient is improved as the grid resolution increases. There is little difference between the fine and medium grid results. The medium grid solution gives satisfactory results while keeping the solution time reasonable. The rest of this thesis will use the resolution of the medium grid, 21,942 nodes, for all other investigations.

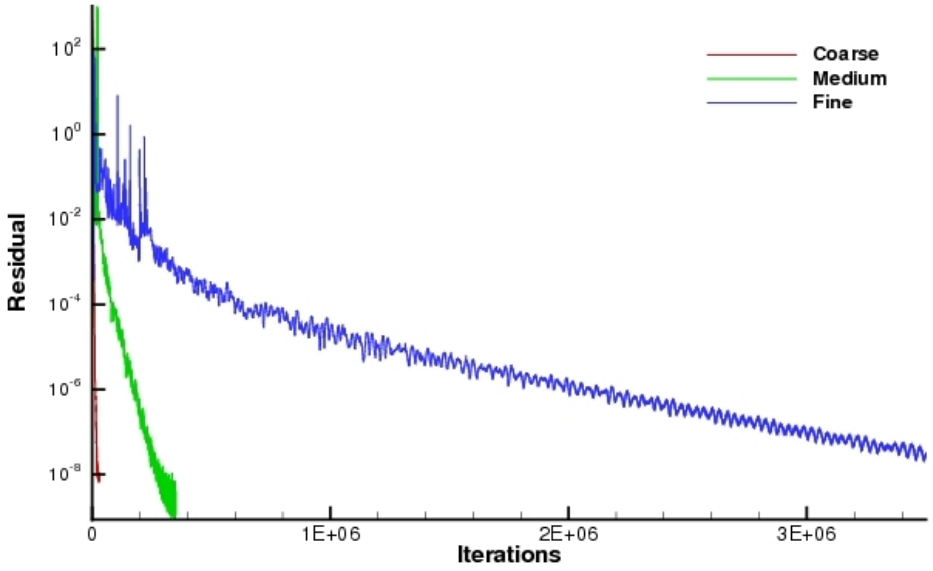

Figure 5.4: Convergence Iterations: (HS1A, $\alpha = 0°$, M=0.71)

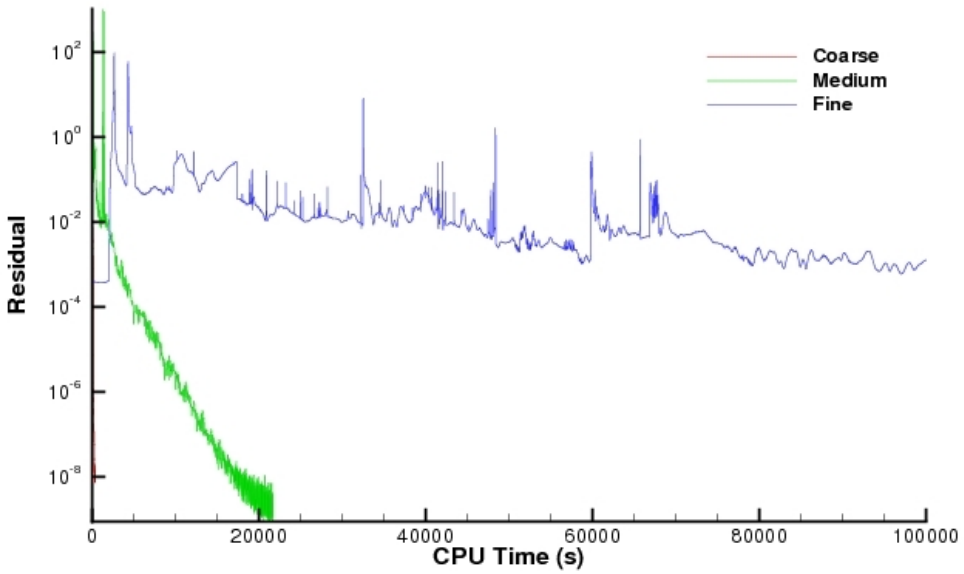

Figure 5.5: Full Convergence Time: (HS1A, $\alpha = 0°$, M=0.71)

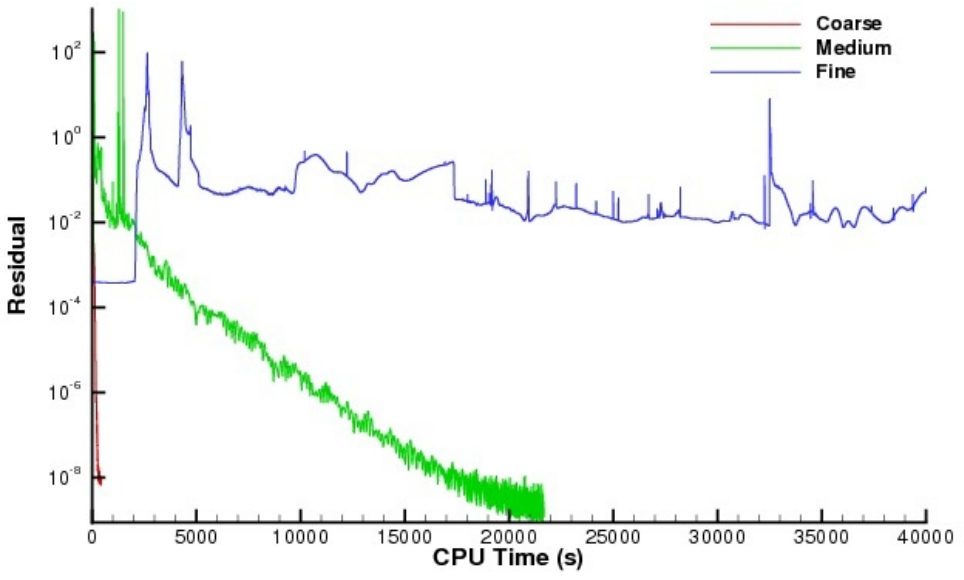

Figure 5.6: Partial Convergence Time: (HS1A, $\alpha = 0°$, M=0.71)

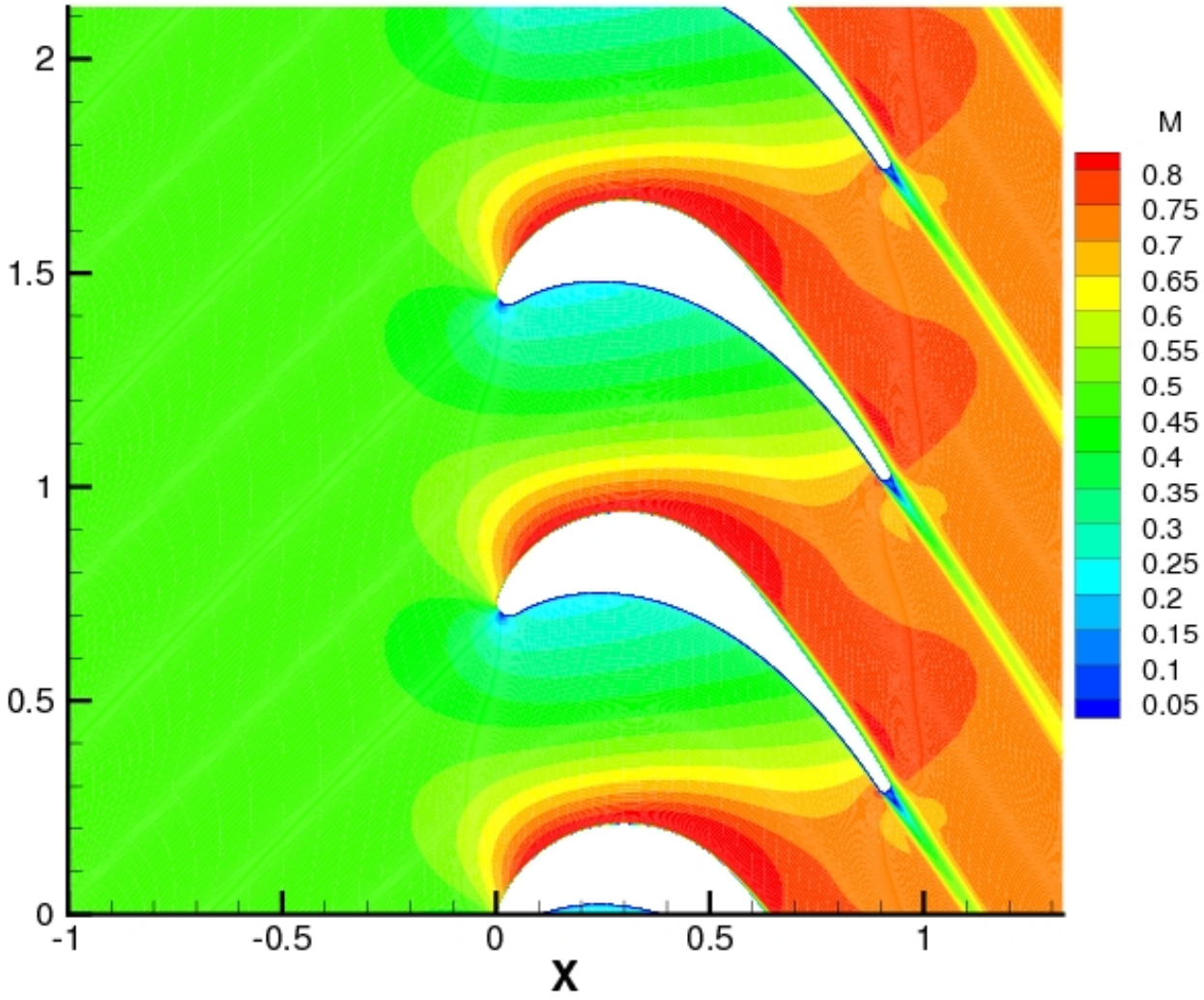

Figure 5.7: Converged Mach Contours: (HS1A, $\alpha = 0°$, M=0.71)

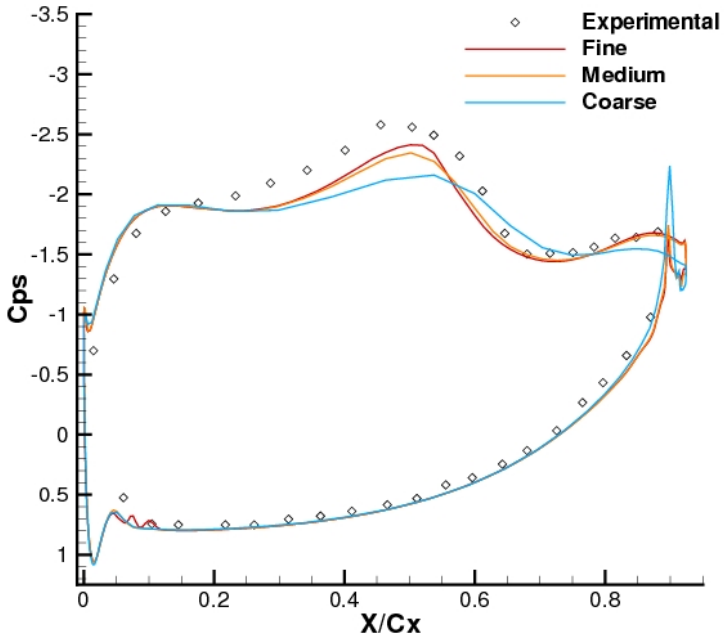

Figure 5.8: Converged C_{ps}: (HS1A, $\alpha = 0°$, M=0.71)

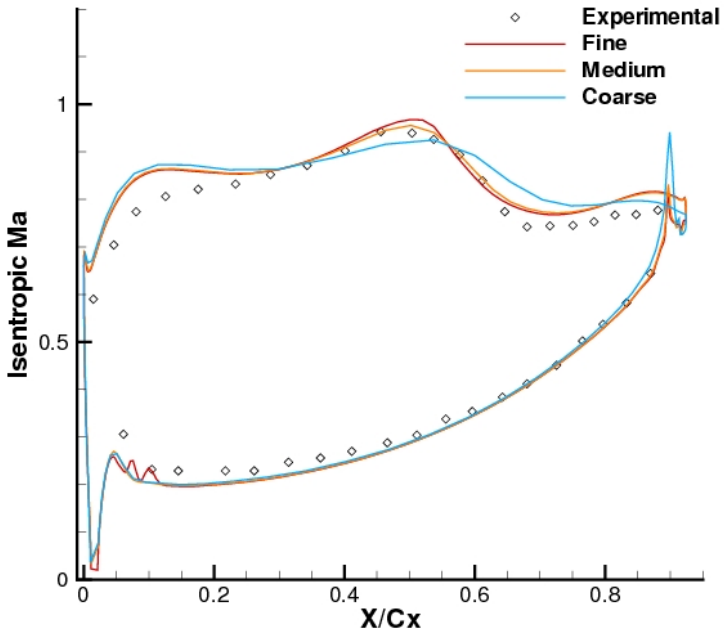

Figure 5.9: Converged Isentropic Mach Number: (HS1A, $\alpha = 0°$, M=0.71)

5.3 Domain Length Influence

To test the effectiveness of the non reflective exit boundary conditions, a grid domain length study was conducted. The medium grid used in the grid convergence study was used as a baseline. A longer second grid was created using the same number of grid points, 21942 nodes. The outlet position was placed 1.0 chord length farther downstream. The two grids are depicted in Figures 5.11 and 5.12.

The domain length influence study used the same test case as the grid convergence study, HS1A, $\alpha = 0°$, M=0.71. The results of the study can be found in Figure 5.10. It can be seen that there is no significant difference between the two results.

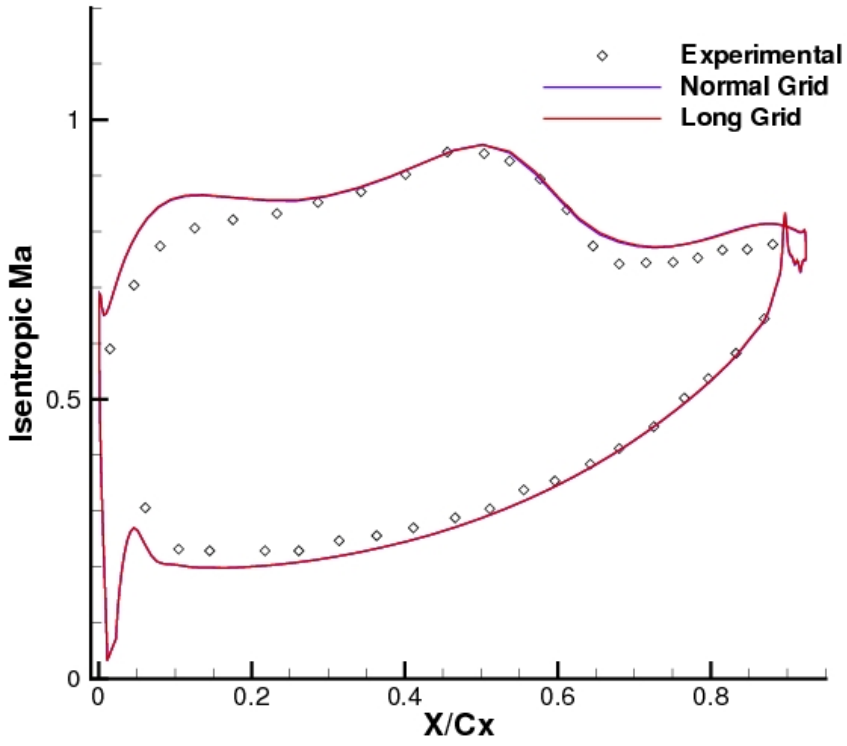

Figure 5.10: Isentropic Mach Number: (HS1A, $\alpha = 0°$, M=0.71)

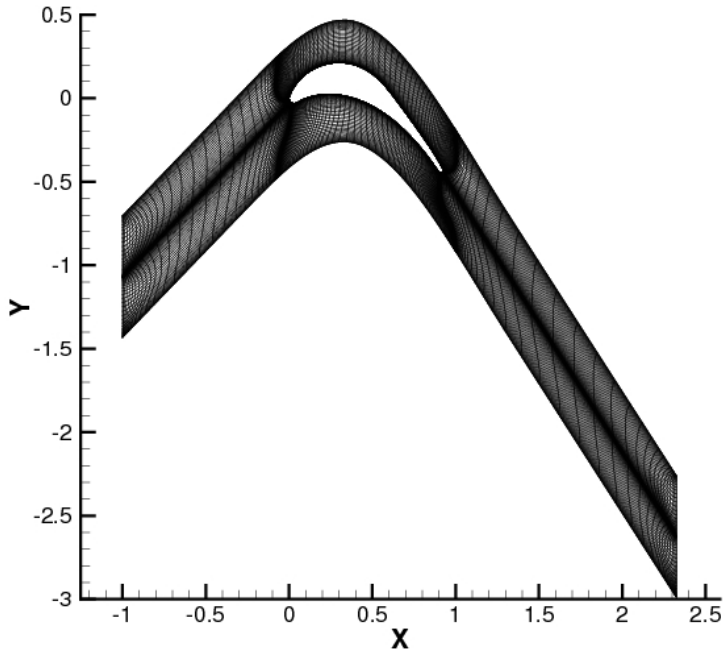

Figure 5.11: Long Grid - HS1A

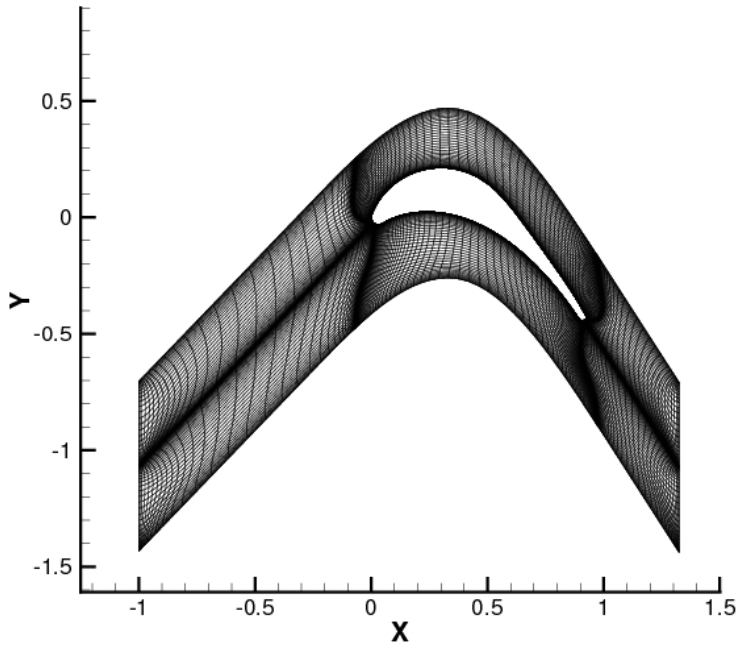

Figure 5.12: Short Grid - HS1A

5.4 Comparison with Experimental Test Cases

In order to demonstrate that the numerical results of the newly developed code are valid, comparisons to experimental results of Jouini [26] are presented in the following subsections. Since there remains the limitation of subsonic freestream Mach number flows on the current flow solver, only cases with freestream Mach numbers below 1.0 are presented.

Original flow solutions assumed that the flow was fully turbulent. These flow solutions did not give satisfactory results. When a transition point was specified, the flow solutions showed the type of results as seen in Section 5.2. A transition point study was preformed and showed no influence on the final C_{ps} or Isentropic Mach number distributions. It is not known why the solution is dependent on the transition terms within the SA turbulence model but not the transition point location. Throughout this thesis a transition point is specified near the trailing edge (0.8 X/c_x) on the lower surface of the blade, and the transition point is specified at the beginning of the adverse pressure gradient on the upper surface of the blade (0.60 X/c_x).

5.4.1 Outlet Mach Number Sweep

A Mach number sweep was performed using four cases that match experimental results by Jouini [26]. Four subsonic design incidence cases were selected and are listed in Table 5.2. Note that the blade used in this analysis is the HS1B blade and not the HS1A blade that was used for the previous numerical analysis and all four cases have the same angle of attack, 0°.

Mach Number	Inlet Total Temperature (K)	Inlet Total Pressure (Pa)	Outlet Static Pressure (Pa)	Reference Length (m)	Inlet Angle
0.50	295.370165	121937.4248	101330.0	0.04	46.0°
0.70	311.7268612	142718.3099	101330.0	0.04	46.0°
0.84	328.2412301	163699.5153	101330.0	0.04	46.0°
0.95	346.5462572	186955.7196	101330.0	0.04	46.0°

Table 5.2: Outlet Mach Number Sweep: Test Cases

The results of the Mach number sweep with the corresponding experimental results

are depicted in Figures 5.13, 5.14, 5.15 and 5.16. The trends that are shown in the experimental results have been replicated in the numerical results. There are very good trends shown in the rear half upper section of the foil. The numerical results do show some smoothing of the pressure gradient, but that is expected with the Spalart-Allmaras turbulence model within adverse pressure gradients [50]. There is some overprediction of the upper surface leading edge isentropic Mach number. This could be due to the selection of turbulence model or the block boundary influences at the leading edge. The actual accuracy at the leading and trailing edges cannot be determined because the experimental results did not have data for these zones.

5.4.2 Angle of Attack Sweep

An angle of attack sweep was performed using four cases that match experimental results by Jouini [26]. Four cases with similar subsonic Mach numbers were selected and are listed in Table 5.3. Although there are numerous cases provided by Jouini, an angle of attack sweep with the four cases of the same Mach number was not possible. The four cases presented here have Mach numbers around 0.7.

Test Case	Inlet Total Temperature (K)	Inlet Total Pressure (Pa)	Outlet Static Pressure (Pa)	Reference Length (m)	Inlet Angle	Mach Number
$-10°$	217.2025216	149675.0369	101330.0	0.04	36.0°	0.75
0°	323.1574575	143934.6591	101330.0	0.04	46.0°	0.71
10°	310.0130275	140540.9154	101330.0	0.04	46.0°	0.68
14.5°	309.8598142	140346.2604	101330.0	0.04	46.0°	0.67

Table 5.3: Angle of Attack Sweep: Test Cases

The results of the angle sweep with the corresponding experimental results are depicted in Figure 5.17, 5.18, 5.19 and 5.20. Once again there is smoothing of the pressure gradient in the rear half upper section of the foil. The trends of the experimental results are still captured relatively well. The upper leading edge section shows an increase in the isentropic Mach number as the angle of incidence increases. The isentropic Mach number overprediction of the upper surface leading edge is also present in the angle of attack sweep. What can be seen from the angle of attack sweep is that the numerical solution becomes more accurate as the angle of attack decreases.

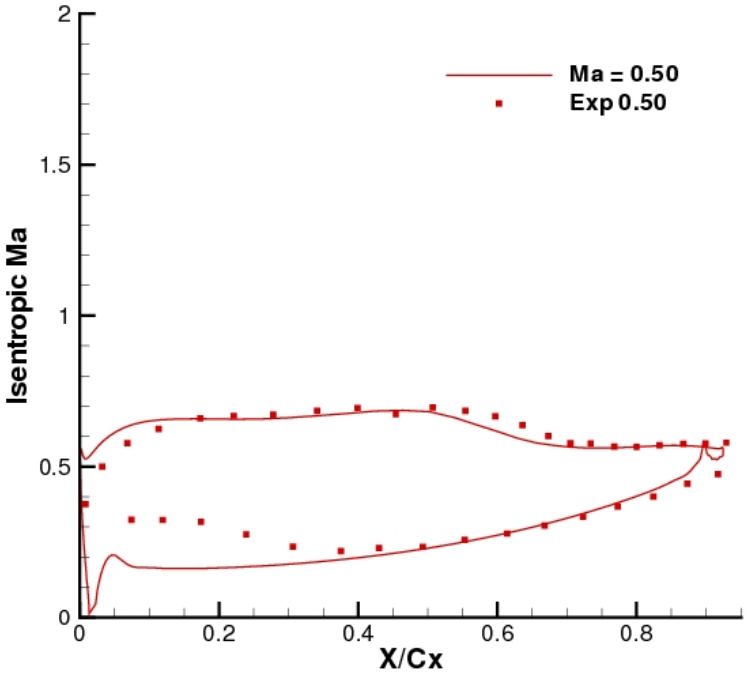

Figure 5.13: HS1B Mach Sweep: M=0.50

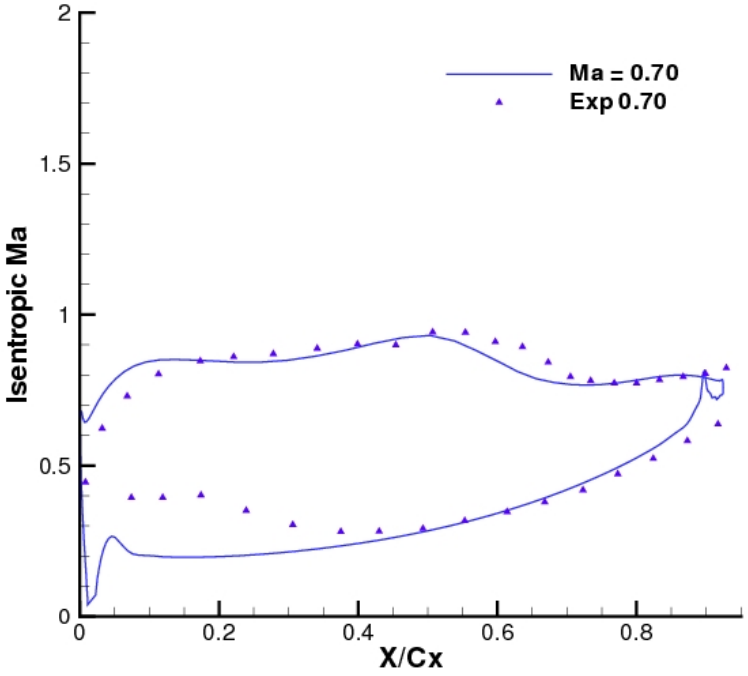

Figure 5.14: HS1B Mach Sweep: M=0.70

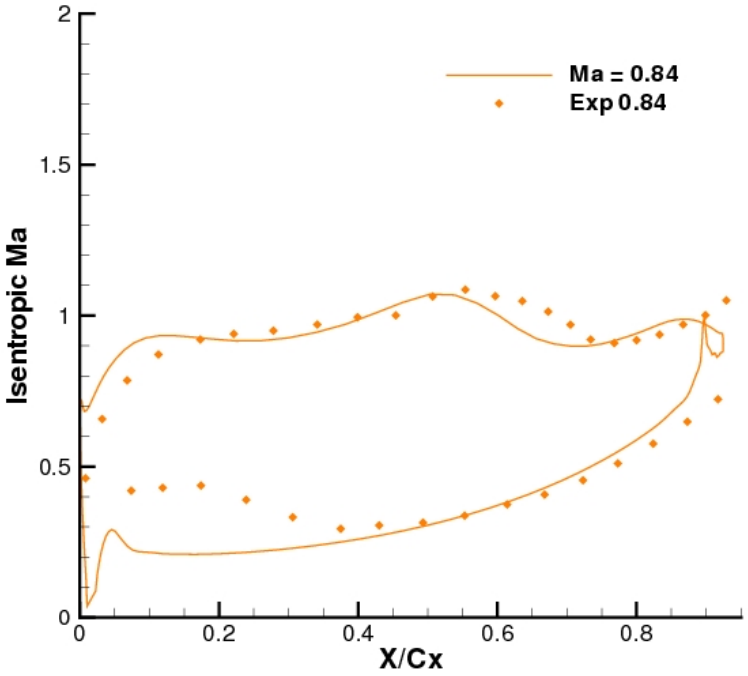

Figure 5.15: HS1B Mach Sweep: M=0.84

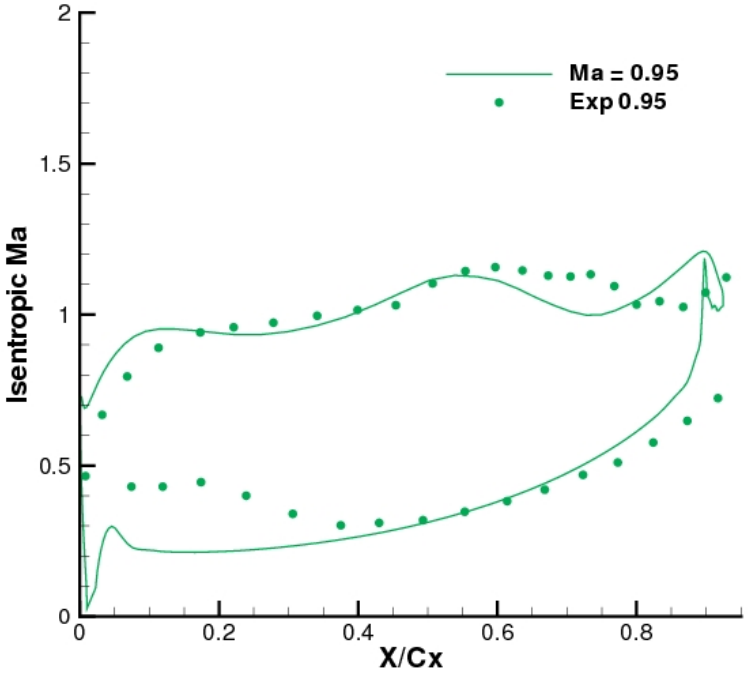

Figure 5.16: HS1B Mach Sweep: M=0.95

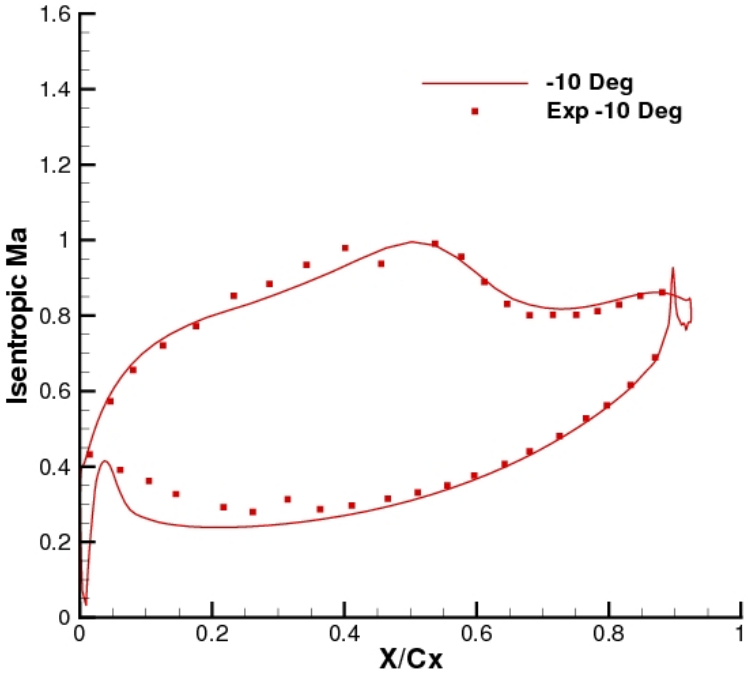

Figure 5.17: HS1A Angle Sweep: $\alpha = -10°$

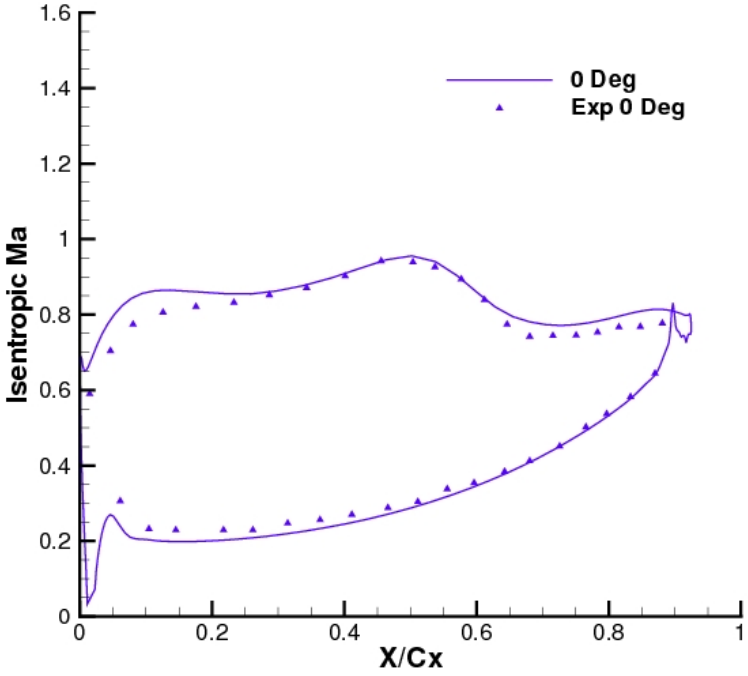

Figure 5.18: HS1A Angle Sweep: $\alpha = 0°$

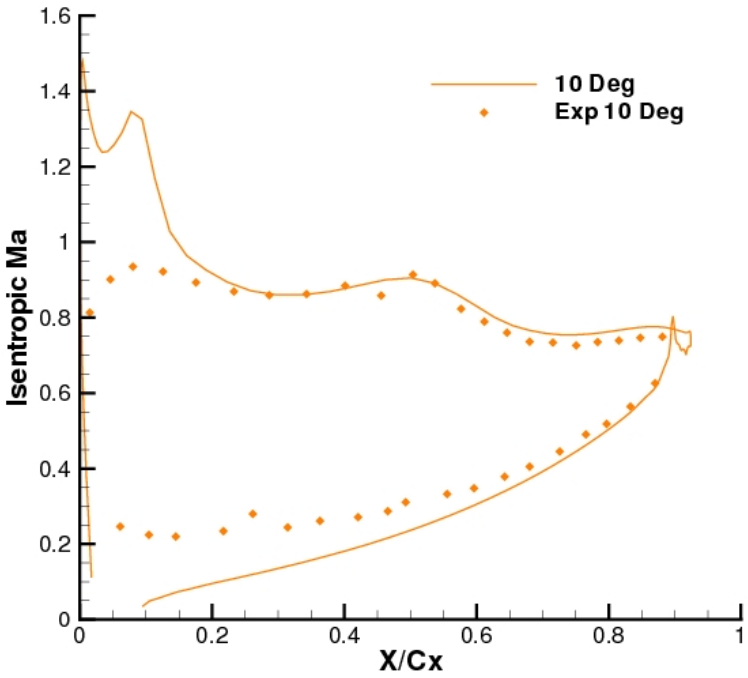

Figure 5.19: HS1A Angle Sweep: $\alpha = 10°$

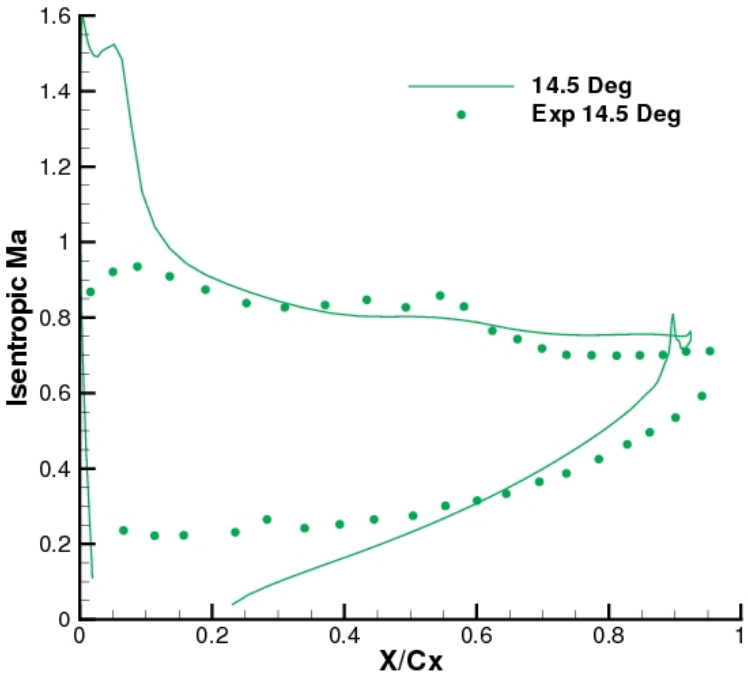

Figure 5.20: HS1A Angle Sweep: $\alpha = 14.5°$

Chapter 6

Conclusions

A 2D cascade structured grid generator was developed using Amber2D as a basis. The tool creates H-type grids for turbomachinery blades and stators. The grids can be customized by the user to provide the desired concentration and number of nodes.

A 2D cascade thin-layer Navier-Stokes flow solver for turbomachinery was developed using TORNADO as a basis. The flow domain is defined using structured grids that decompose the flow domain into 6 blocks. The tool models turbulence through the one-equation Spalart-Allmaras model. A second-order finite-differencing and a second and fourth-difference dissipation scheme are used. Periodic and non-reflecting inlet and outlet boundary conditions are implemented to represent the physics of turbomachinery flow. An approximate-factorization numerical method was successfully implemented to resolve the resulting turbomachinery governing equations.

A grid convergence study was performed for an HS1A turbine blade at a transonic flow condition. The various grid sizes were compared against experimental results by Jouini [26]. A grid with off wall spacing of $2 \cdot 10^{-6} c_x$, 201 nodes on the foil surface and a total of 21,942 nodes showed the best result for a reasonable solution time.

A test of the non-reflective inlet and outlet boundary conditions was performed. The implemented boundary conditions produced the same results when the downstream boundary was moved by one chord length.

Comparisons of flow solver numerical results and experimental results were performed. A Mach number sweep was performed against a HS1B blade at four different subsonic Mach numbers (0.50, 0.70, 0.84, 0.95). Reasonable agreement with experimental results were shown over the selected range of Mach numbers. An angle of attack sweep of a HS1A blade was performed against experimental results at four different angles of attack

($-10.0°$, $0.0°$, $10.0°$, $14.5°$). Good agreement with experimental results were obtained for the $-10.0°$ and $0.0°$ cases. The $10.0°$ and $14.5°$ cases did not predict the leading edge upper surface well. Arnone, Liou and Povinelli explained that this could be due to truncation errors introduced by grid distortion at the grid stagnation point that introduce extra entropy that is transported downstream [3].

Chapter 7

Recommendations

The most obvious way to improve this design tool would be to reduce solution time. Implementing an efficient Newton-Krylov numerical method would speed up results shown by Pueyo and Zingg [40].

Solution accuracy could be increased by implementing a hybrid H-O grid. The elimination of the grid distortion at the leading edge could increase the solution accuracy. A hybrid H-O grid generation tool would be necessary.

A final step of implementing a gradient based optimization algorithm like that of Nemec and Zingg [35] could realize the goal of a full turbomachinery design tool. Incorporating: unsteady, multiple blade rows and 3 dimensional capabilities would develop the tool into becoming a cost saving design tool suitable for industry.

REFERENCES

[1] *CFD Validation for Propulsion System Components.* AGARD Advisory Report 355, 1998.

[2] D. G. Ainley and G. C. R. Mathieson, *A Method of Performance Estimation for Axial-Flow Turbines.* Aeronautical Research Council, 1957.

[3] A. Arnone, M.-S. Liou, and L. A. Povinelli, *Navier-Stokes Solution of Transonic Cascade Flows Using Nonperiodic C-Type Grids*, Journal of Propulsion and Power, Vol. 8, No. 2 (1992), pp. 410–417.

[4] A. Arnone and R. C. Swanson, *A Navier-Stokes Solver for Cascade Flows.* ICASE, 1988.

[5] T. Arts, *Three Dimensional Rotational Invisicid Flow Calculation in Axial Turbine Blades.* Von Karman Institute for Fluid Flow Technical Note 154, 1985.

[6] R. M. Beam and R. F. Warming, *An Implicit Factored Scheme for the Compressible Navier-Stokes Equations*, AIAA Journal, Vol. 26, no. 4 (1978), pp. 393–402.

[7] S. Burguburu, C. Toussaint, C. Bonhomme, and G. Leroy, *Numerical Optimization of Turbomachinery Bladings*, Journal of Turbomachinery, Vol. 126 (2004), pp. 91–100.

[8] M. S. Campobasso, *Effects of Flow Instabilities on the Linear Harmonic Analysis of Unsteady Flow in Turbomachinery.* Doctor of Philosophy Thesis, St Hugh's College, 2004.

[9] R. V. Chima, *Inviscid and Viscous Flows in Cascades with an Explicit Multiple-Grid Algorithm*, AIAA Journal, Vol. 23, No. 10 (1985), pp. 1556–1563.

[10] C. H. Choi and J. Y. Yoo, *Cascade Flow Calculations Using the κ - ω Turbulence Model with Explicit-Implicit Solver*, AIAA Journal, Vol. 35 (1997), pp. 1551–1552.

[11] R. Corral and F. Gisbert, *Profiled End Wall Design Using an Adjoint Navier-Stokes Solver*, ASME Journal of Turbomachinery, Vol. 130, pp. 021011–8.

[12] C. Cravero and A. Satta, *Comparison of Semi-Empirical Correlations and Navier-Stokes Method for the Overall Performance Assessment of Turbine Cascades.* Transactions of the ASME, 2003.

[13] S. De Rango, *Higher-Order Spatial Discretization for Turbulent Aerodynamic Flows*. PhD Thesis, University of Toronto, 2001.

[14] J. Driver and D. W. Zingg, *Numerical Aerodynamic Optimization Incorporating Laminar-Turbulent Transition Prediction*, AIAA Journal, Vol. 45, No. 8 (2007), pp. 1810–1816.

[15] M. Duta, M. Giles, and M. Campobasso, *The Harmonic Adjoint Approach to Unsteady Turbomachinery Design*, International Journal for Numerical Methods in Fluids, Vol. 40 (2002), pp. 323–332.

[16] R. Florea and K. C. Hall, *Sensitivity Analysis of Unsteady Inviscid Flow Through Turbomachinery Cascades*, AIAA Journal, Vol. 40, No. 6 (2001), pp. 1047–1056.

[17] M. B. Giles, *Non-Reflecting Boundary Conditions for the Euler Equations*, 1988.

[18] ——, *UNSFLO: A Numerical Method for Calculation of Unsteady Flow in Turbomachinery*. GTL, 1991.

[19] ——, *Some Thoughts on Exploiting CFD for Turbomachinery Design*. IMeche Symposium, Oxford University Computing Laboratory, 1998.

[20] ——, *Aerodynamic Design Optimization for Complex Geometries Using Unstructured Grids*. Report no. 97/08, Oxford University Computing Laboratory, 2000.

[21] J. P. Gostelow, *Cascade Aerodynamics*, Pergamon Press, 1984.

[22] J. Halama, T. Arts, and J. Fort, *Numerical Solution of Steady and Unsteady Transonic Flow in Turbin Cascades and Stages*, Computers and Fluids, Vol. 33 (2004), pp. 729–740.

[23] L. He and W. Ning, *Efficient Approach for Analysis of Unsteady Viscous Flows in Turbomachines*, AIAA Journal, Vol. 36, No. 11 (November 1998), pp. 2005–2012.

[24] J. E. Hicken and D. W. Zingg, *Aerodynamic Optimization Algorithm with Integrated Geometry Parameterization and Mesh Movement*, AIAA Journal, Vol. 48, No. 2 (2010), pp. 401–413.

[25] D. G. Holmes, *Numerical Methods for Flows in Turbomachinery*. Von Karman Institute for Fluid Dynamics Lecture Series, 1989.

[26] D. B. M. Jouini, *Experimental Investigation of Two Transonic Linear Turbine Cascades at Off-Design Conditions*. PhD Thesis, Carleton University, 2000.

[27] T. Kiss, J. A. Schetz, and H. L. Moses, *Experimental and Numerical Study of Transonic Turbine Cascade Flow*, AIAA Journal, Vol. 34 (1996), pp. 104–109.

[28] R. F. Kunz and B. Lakshminarayana, *Explicit Navier-Stokes Computation of Cascade Flows Using the κ-ϵ Turbulence Model*, AIAA Journal, Vol. 30, No. 1 (1992), pp. 13–22.

[29] F. Liu and X. Zheng, *Staggered Finite Volume Scheme for Solving Cascade Flow with a k-ω Turbulence Model*, AIAA Journal, Vol. 32, No. 8 (1994), pp. 1589–1597.

[30] J.-S. Liu, P. M. Sockol, and J. M. Prahl, *Navier-Stokes Cascade Analysis With a Stiff κ-ϵ Turbulence Solver*. NASA, 1988.

[31] V. Michelassi and F. Martelli, *Blade Row Interference Effects in Axial Turbomachinery Stages*. Von Karman Institute for Fluid Dynamics Lecture Series 1998-02, Numerical Simulation of Unsteady Stator-Rotor Interaction in Brite-Turbine Stage, 1998.

[32] K. Nakahashi, O. Nozaki, K. Kikuchi, and A. Tamura, *Navier-Stokes Computations of Two- and Three-Dimensional Cascade Flowfields*, Journal of Propulsion, Vol. 5, No. 3 (1989), pp. 320–326.

[33] T. E. Nelson, *Numerical Solution of the Navier-Stokes Equations for High Lift Airfoil Configurations*, PhD thesis, University of Toronto, 1994.

[34] T. E. Nelson and D. W. Zingg, *Compressible Navier-Stokes Computations of Multielement Airfoil Flows Using Multiblock Grids*, AIAA Journal, Vol. 32 (March 1994), pp. 506–511.

[35] M. Nemec, D. W. Zingg, and T. H. Pulliam, *Multipoint and Multi-Objective Aerodynamic Shape Optimization*, AIAA Journal, Vol. 42, No. 6 (2004), pp. 1057–1065.

[36] S. Niazi, *Numerical Simulation of Rotating Stall and Surge Alleviation in Axial Compressors*. PhD Thesis, Georgia Institute of Technology, 2000.

[37] J. Nichols, *A Three-Dimensional Multi-block Newton-Krylov Flow Sovler for the Euler Equations*. MASc Thesis, University of Toronto, 2004.

[38] A. Oyama, M.-S. Liou, and S. Obayashi, *Transonic Axial-Flow Blade Shape Optimization Using Evolutionary Algorithm and Three-Dimensional Navier-Stokes Solver*, AIAA, 2002-5642 (2002).

[39] D. I. Papadimitriou and K. C. Giannakoglou, *Compressor Blade Optimization Using A Continuous Adjoint Formulation*. Proceedings of GT2006, 2006 ASME TURBGO EXPO, May 8-11, Barcelona Spain, 2006.

[40] A. Pueyo and D. W. Zingg, *Efficient Newton-Krylov Solver for Aerodynamic Computations*, AIAA Journal, Vol. 36, No. 11 (1998), pp. 1991–1997.

[41] T. H. Pulliam, *Efficient Solution Methods for the Navier-Stokes Equations*, Lecture Notes for the Von Karman Institute For Fluid Dynamics Lecture Series, 1986.

[42] T. H. Pulliam and D. S. Chaussee, *A Diagonal Form of an Implicit Approximate Factorization Algorithm*, Journal of Computational Physics, vol. 39 (1981), pp. 347–363.

[43] R. Ramamurti, U. Ghia, and K. N. Ghia, *A Semi-Elliptic Analysis for 2-D Viscous Flows Through Cascade Configurations*, Computers Fluids, Vol. 20, No. 3 (1991), pp. 223–242.

[44] M. P. Rumpfkeil and D. W. Zingg, *A Hybrid Algorithm for Far-Field Noise Minimization*, Computers and Fluids, doi:10.1016/j.compfluid.2010.05.006 (2010).

[45] A. Samad and K.-Y. Kim, *Shape Optimization of an Compressor Blade by Multi-Objective Genetic Algorithm*, Journal of Power and Energy, Vol. 222, Part A (2008), pp. 599–611.

[46] H. I. H. Saravanamuttoo, G. F. C. Rogers, and H. Cohen, *Gas Turbine Theory*, Prentice Hall, England, 2001.

[47] P. R. Spalart and S. R. Allmaras, *A One-Equation Turbulence Model for Aerodynamic Flows*. AIAA, 92-0439, 1992.

[48] R. C. Swanson and E. Turkel, *On Central-Difference and Upwind Schemes*, Journal of Computational Physics, 101 (1992), pp. 292–306.

[49] Á. Veres and I. Sánta, *A Mathematical Model on Transonic Axial Compressor Rotor Flow*, Periodica Polytechnica Ser. Transp Eng., Vol 30, No. 1-2 (2002), pp. 53–67.

[50] D. C. Wilcox, *Turbulence Modeling for CFD Second Edition*, DCW Industries, Inc., 2002.

[51] A. R. Wilkinson and D. W. Zingg, *AMBER 2D Users's Guide*. UTIAS CFD LAB, 1993.

[52] H.-Y. Wu, F. Liu, and H.-M. Tsai, *Aerodynamic design of turbine bleades using an adjoint equation method*. 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno Nevada, January 2005.