TIME-LINEARIZATION OF THE SPALART-ALLMARAS TURBULENCE MODEL

A Thesis

by

MICHAEL DANIEL POLEWSKI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Paul Cizmas
Committee Members,	Diego Donzis
	Doug Allaire
Head of Department,	Rodney Bowersox

May 2020

Major Subject: Aerospace Engineering

ABSTRACT

This work expands on a reduced-order flow solver used to predict harmonically unsteady flows, which utilizes the time-linearization method. The time-linearization method requires a steady-state flow field from a full-order model flow solver. The time-linearized flow solver then uses the steady-state flow field to predict the unsteady, complex amplitudes of perturbation in the frequency domain. Together, the full-order model and time-linearized flow solvers predict unsteady flows with small harmonic perturbations at a reduced computational cost than an unsteady full-order model simulation. This work explores the limitations of the time-linearized flow solver through various verification and validation tests, with the end goal of implementing a time-linearized Spalart-Allmaras turbulence model to provide closure to the time-linearized Reynolds-Averaged Navier-Stokes equations. The time-linearized flow solver was also used to perform rotordynamic stability analysis of a straight annular gas seal based off the seal used in the High-Pressure Oxidizer Turbo-Pump of the Space Shuttle Main Engine. New features such as a complex grid deformation tool and a complex force integrator were developed for the time-linearized flow solver to simulate complex geometries. Turbulence modeling is required to provide closure for the Reynolds-Averaged Navier-Stokes equations. This research compares two turbulence models for the full-order model flow solver: The Shear Stress Transport turbulence model and the newly implemented Spalart-Allmaras turbulence model. These turbulence models underwent verification and validation tests against cases from NASA's Turbulence Modeling Resource website along with a turbomachinery blade cascade.

DEDICATION

To my Parents and Wife for your unconditional love and support

ACKNOWLEDGMENTS

I would like to thank Dr. Paul Cizmas for convincing me to pursue a Master of Science degree, and for his support and guidance in the field of computational fluid dynamics. I would also like to thank Dr. Neil Matula, Dr. Forrest Carpenter, and Elizabeth Krath for your friendship, guidance and expertise.

Lastly, I would like to thank my cousin Lily Ricketts and her family for always being close by to provide delicious meals, laughter and support.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professors Paul Cizmas and Diego Donzis of the Department of Aerospace Engineering and Professor Doug Allaire of the Department of Mechanical Engineering.

All work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was funded by the Turbomachinery Research Consortium at Texas A&M University.

NOMENCLATURE

UNS3D	Unstructured, Unsteady, 3-Dimensional Flow Solver
SA	Spalart-Allmaras Turbulence Model
SST	Shear Stress Transport Turbulence Model
FOM	Full Order Model
TL	Time-Linearized
NASA	National Aeronautics and Space Administration
POD	Proper-Orthogonal Decomposition and Parallel version of UNS3D
LAPACK	Linear Algebra Package
RBF	Radial Basis Functions

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Computational fluid dynamics (CFD) envelops a large array of aerodynamic flow solvers, each with a different level of accuracy and computational expense. The flow solvers that are able to solve unsteady and turbulent flows with good accuracy include the Reynolds-Averaged Navier-Stokes (RANS), Large-Eddy Simulation (LES), and Direct Numerical Simulation (DNS) flow solvers. The most common CFD solver used for engineering application are the RANS flow solvers. This is primarily due to the large computational expense of LES and DNS flow solvers. RANS flow solvers involve decomposing the Navier-Stokes equations into a time-averaged and temporally fluctuating components, where only the time-averaged components are solved. The temporally fluctuating components produce extra terms known as the Reynolds stresses. The Reynolds stresses create a closure problem for the RANS equations, and thus require a form of turbulence modeling to close the problem. Turbulence models generally follow the Boussinesq eddy viscosity approximation to compute the Reynolds stresses. Since RANS flow solvers solve for a time-averaged solution, RANS flow solvers do not model turbulence nearly as well as LES or DNS flow solvers. Both LES and DNS are common tools to understanding the formation of turbulent structures, as they resolve the smaller turbulent scales than RANS flow solvers. The large computational expense of LES and DNS stems from solving the smaller turbulent scales, which are not as critical or feasible for general engineering applications. Even though RANS is considered to be the least expensive method for high accuracy CFD codes, simulating a turbulent and unsteady flow field can still be computationally expensive. Therefore, a method to calculate the unsteady and turbulent aerodynamics in an inexpensive manner is desired.

A reduced-order model to simulate harmonically-varying unsteady flows has been developed by Liliedahl (2017). This reduced order model is known as a time-linearized flow solver. The time-linearized flow solver is based off of RANS flow solvers, and currently lacks a time-linearized turbulence model to provide closure. The lack of a time-linearized turbulence model may or may not produce nonphysical solutions (Holmes and Lorence, 1998; Clark and Hall, 1999).

1.1 Background

For typical turbomachinery applications, one must typically obtain an unsteady, turbulent flow field that has reached a periodic state. It can be computationally expensive for a high accuracy flow solver, such as RANS and LES, to obtain such a state. For unsteady motion, one has to deform the computational mesh time-accurately to match the motion of the object. To model a turbulent flow, one must have a suitable mesh density to capture the turbulent and viscous effects of the flow. Flow solvers that use an explicit time-marching scheme require very small time steps in order to integrate accurately in time. Reaching a periodic solution may require many periods of oscillation to be simulated. It is then desired to have a method that is able to retain a high level of accuracy while reducing the computational cost.

In order to reduce the computational expense of acquiring a periodic solution, one must either eliminate the need for a time-accurate marching method, or to reduce the degrees of freedom of the problem. By eliminating the need for a time-accurate marching method, the computer does not need to compute the transient solution in order to obtain the periodic solution. By reducing the number of nodes of the problem, the computer has less computations to do. This section will now focus on the different methods in which the computational expense to simulate harmonically-varying, unsteady flows have been reduced compared to the expense of standard RANS flow solvers.

A method that eliminates the need for a time-accurate marching method is the known as the Quasi-Steady state method. The Quasi-Steady state method solves the RANS equations in a transformed coordinate system such that the unsteadiness can be modeled as a steady-state problem. This allows the unsteady solution to be solved in pseudo-time. For gas seals, the limitation to this method is that one needs to simulate the full annulus, the seal geometry must be axisymmetric, and the whirling orbit must be circular.

Attempts to further reduce the computational expense of harmonically-varying, unsteady aerodynamics have been made through the use of perturbation methods. With perturbation methods, the size of the computational domain may be reduced by solving the harmonic perturbations in the frequency domain. Examples of this domain reduction include simulating only one blade passage for blade cascades with multiple interblade phase angles, or reducing a full annulus of a seal to a small slice of the annulus. Since the unsteady, harmonically-varying perturbations are solved in the frequency domain, this eliminates the expense of calculating a transient solution to reach a periodic state. Several perturbation methods to model harmonically unsteady aerodynamics have been developed starting as early as the 1970's. Ni and Sisto (1976) were one of the first to apply a harmonic perturbation method for stall flutter in turbomachinery flows. The principle behind the perturbation method is that the unsteady flow field can be decomposed into a steady term and a unsteady perturbation. For turbomachinery flows, these unsteady perturbations are assumed to be harmonic. Since these perturbations are very small, the unsteady terms becomes a system of complex, linear equations. Ni and Sisto applied this method for a 2D flate plate cascade. Dietzen and Nordmann (1987) were among the first to perform rotordynamic analysis using this perturbation method. Dietzen and Nordmann used a quasi-3D finite difference solver, where both the laminar and eddy viscosity were assumed constant. In this method, the $\kappa - \epsilon_t$ turbulence model was dropped for the perturbation analysis. Arghir and Frene (1997) later expanded on the work done by Dietzen and Nordmann (1987), by including the effects of a linearized eddy viscosity term. These methods were limited in that they required axisymmetric domains, structured grids and its coordinate transformation of the computational grid. Along with these limitations, the assumptions applied were that the rotor had circular whirling orbit and the fluid was incompressible. The perturbation methods developed by Dietzen and Nordmann for rotordynamic stability analysis are referred to as the Whirling Rotor Methods.

The method used in this thesis is known as the Time Linearization method. The primary difference between the Whirling Rotor method and Time Linearization is how the coordinate transformations of the computational grid are obtained. Time-linearized coordinate transformations allow for any form of periodic motion to be used, as opposed to the whirling rotor methods which could only model circular orbits. Time Linearization was developed by Hall and Lorence (1993) to model cascaded airfoil geometries of inviscid and incompressible flows. Cizmas and Hall (1995) expanded on the Time Linearization method by including viscous effects in a interacting viscous-inviscid method. Clark and Hall (1999) and Holmes and Lorence (1998) applied the Time Linearization method to the two dimensional and three dimensional RANS equations, respectively, both including a time-linearized turbulence model. Both Clark and Hall and Holmes and Lorence applied their respective time-linearized RANS flow solvers for analyzing stall flutter and forced response on turbomachinery blade cascades. Both Clark and Hall and Holmes and Lorence found that the exclusion of the time-linearized turbulence model may produce physically incorrect solutions.

Time-linearized flow solvers may be used to simulate any harmonically-varying, unsteady flow field. Liliedahl (2017) applied the time-linearized flow solver, UNS3D TL, for turbomachinery applications, specifically rotordynamic stability analysis. Lilliedahl analyzed the rotordynamic stability of two gas seals: a straight annular seal (Nelson, 1985) and a stepped labyrinth seal (Wagner et al., 2009).

1.2 Proposed Research

Liliedahl (2017) has developed a time-linearized flow solver for the in-house CFD code UNS3D. UNS3D is a RANS flow solver which is three dimensional and unstructured. The time linearized code, UNS3D TL, was developed off of the existing structure of UNS3D. Having the same structure allows for updates and modifications between codes to be simple. Lilliedahl applied the time-linearized flow solver, UNS3D TL, for rotordynamic analysis, but did not implement a time-linearized turbulence model. Since the lack of a time-linearized turbulence model may or may not result in incorrect physical solutions, the proposed research is to develop an appropriate time-linearized turbulence model to improve the existing time-linearized flow solver UNS3D TL.

The turbulence model selected to be time-linearized is the Spalart-Allmaras one-equation turbulence model (Spalart and Allmaras, 1992). This turbulence model was chosen over the Shear Stress Transport (SST) two equation model (Menter, 1994) that already existed in UNS3D. This was due to the difficulties that would arise due to SST being a blend of the $\kappa - \epsilon_t$ and the $\kappa - \omega_t$ turbulence models. Spalart-Allmaras was chosen for its popularity in turbomachinery and internal flows. As a result, the proposed research also includes the implementation, validation and verification of the Spalart-Allmaras turbulence model for the time-marching and time-linearized versions of UNS3D.

For UNS3D's Spalart-Allmaras verification and validation, a collection of cases and grids provided by NASA's turbulence modeling resource website will be used (Rumsey et al., 2010). Each of the cases were selected to test a certain aspect of flow characteristics and boundary conditions common to turbomachinery flow. Three cases were selected from the turbulence modeling resource: Zero-Pressure Gradient Turbulent Flat Plate, 2D Airfoil Near Wake, and Backwards Facing Step. The Zero-Pressure Gradient Turbulent Flat Plate case is a canonical case to verify and validate turbulence models. For the flat plate case, one can compare the implemented turbulence model results to Cole's Law of the Wake (Coles, 1956). The Airfoil Near Wake is a case that simulates an experiment done by Nakayama (1985), for a supercritical airfoil. This numerical experiment measured the turbulence model's ability to predict the flow while under a non-zero pressure gradient. The Backwards Facing Step is another canonical case for turbulence models, where the turbulence model's ability to resolve separated flows is tested. This case simulated an experiment done by Driver and Seegmiller (1985). In addition to these NASA verification and validation cases, the turbulence model was compared against published experimental and numerical results for the 11th Standard Configuration, which is a set of experiments studying steady and unsteady turbine blade cascades (Fransson et al., 1999). This case tested the turbulence model's ability to simulate the general characteristics of turbomachinery flow: pressure gradients, separation and periodicity. The Shear Stress Transport turbulence model will also be undergoing the same test cases as Spalart-Allmaras to compare the accuracy and efficiency of each model. The implementation, verification and validation of the Spalart-Allmaras turbulence model will be done on both the single processor and multi-processor versions of UNS3D.

To ensure the time-linearized flow solver, UNS3D TL, is working properly, the verification and validation cases performed by Liliedahl (2017) will be repeated. Running these cases will provide an understanding of how the time-linearized code works and how best to implement the time-linearized turbulence model. These cases include Stokes' second problem, an inviscid channel with oscillating back pressure, and a straight annular seal modeled after the High Pressure Oxidizer Turbo-Pump of the Space Shuttle Main Engine (Nelson, 1985). Stokes' second problem is a 2D oscillating infinite flat plate where an analytical solution exists for the Navier-Stokes equations. In this experiment, the time-linearized flow solver's ability to simulate moving grids will be tested. For the channel with oscillating back pressure, the time-linearized flow solver, UNS3D TL, will test its ability to model oscillatory flow features only. The time-linearized flow solver, UNS3D TL, will compare its results with an unsteady solution obtained from the time-marching flow solver, UNS3D. The straight annular seal rotordynamic results will be compared to the results of Nelson's bulk flow model (Nelson, 1985).

Additional upgrades to UNS3D TL have also been implemented. The upgrades allow for a more automated process for simulating moving complex geometries such as the 11th Standard Configuration or a stepped labyrinth seal. One of the inputs to the time-linearized flow solver, is a mesh that describes the complex amplitudes of oscillatory motion of each grid point that the flow solver is simulating. Algebraic functions used to describe the motion of the computational grid may be defined by the user for cases with simple geometries, but developing algebraic functions for a complex geometry can be extremely difficult and require a lot of coding. Therefore, a complex grid deformation tool utilizing radial basis functions (de Boer et al., 2007) is also proposed. The method used by Liliedahl (2017) to determine the complex forces acting on the full annulus was limited to straight annular seals only. A new tool for calculating the complex forces on any moving wall boundary has been developed. This tool includes the option of replicating the force of a full annulus using a slice of the full domain.

In addition to the cases performed by Liliedahl (2017), the time-linearized flow solver, UNS3D TL, will be verified and validated with additional test cases. A translating rotor in inviscid flow will be used to compare the results obtained from time-linearized and time-marching flow solvers. The translating inviscid rotor case will test the time-linearized flow solver's performance when using a deforming mesh. The translating inviscid rotor will also test the accuracy of the force integration methods and the complex rotational periodic boundary conditions on the time-linearized flow solver. In addition to Stokes' second problem, the laminar viscous model will be tested using a flat plate case with oscillating back pressure. The results from the time-linearized flow solver will be compared against results obtained from the time-marching flow solver. To verify the time-linearized Spalart-Allmaras turbulence model, a turbulent flat plate with oscillating back pressure simulation will be done and compared to the results obtained from time-marching flow solver.

1.3 Novel Aspects of Research

This research project includes the implementation, verification and validation of the Spalart-Allmaras turbulence model into the UNS3D family of flow solvers. In addition to the verification and validation of the Spalart-Allmaras turbulence model, the verification and validation of the Shear Stress Transport turbulence model will be tested and compared to the Spalart-Allmaras turbulence model. For the time-linearized flow solver, the cases published by Liliedahl (2017) and additional cases will be simulated to verify how the time-linearized flow solver operates. In addition, two new features are added to the time-linearized flow solver: a complex force integrator and a grid deformation tool for the deformation of complex geometries.

2. PHYSICAL MODELS

The governing equations for the computational fluid dynamics flow solvers used in this research are the composed of the mass, momentum and energy conservation laws. The collection of these conservation laws for fluid dynamics is known as the Navier-Stokes equations. This section describes each of the conservation laws, it's general integral form and the formulation of the Reynolds-Averaged Navier-Stokes equations. This section will also present the turbulence models used and implemented in this research.

2.1 Governing Equations

The general formulation of a conservation law begins with observing an arbitrary conservative variable, ϕ, within a control volume. The control volume is an arbitrary finite volume, Ω, bounded by a closed surface, $\partial\Omega$. The closed surface consists of infinitesimal surface elements, dS, with a unit normal vector, $\hat{n}$, conventionally pointing outward from the control volume. A conservation law states that the time rate of change of a conserved quantity within a control volume is equal to the sum of the net flux through the surface of the control volume and the source terms of ϕ. The time rate of change within the control volume Ω is described as

$$\frac{\partial}{\partial t}\int_{\Omega}\phi d\Omega \tag{2.1}$$

For a fluid, the net flux is composed of the convective flux and the diffusive flux. The convective flux describes the amount of the conserved quantity, ϕ, going through the control surface due the the motion of the fluid. The convective flux, F_c is described as

$$F_c = -\oint_{\partial\Omega}\phi\left(\vec{v}\cdot\hat{n}\right)dS \tag{2.2}$$

where $\vec{v}$ is the velocity of the fluid.

The diffusive flux describes flux of ϕ entering the control volume Ω due to the diffusion of ϕ.

The diffusive flux is proportional to the gradient of ϕ, and is described as

$$\oint_{\partial\Omega} \kappa\rho \left(\nabla\phi \cdot \hat{n}\right) dS \tag{2.3}$$

where κ is a diffusivity coefficient, ρ is the density and $\nabla\phi$ is the gradient of ϕ.

The source terms can be acted both on the control volume through a vector $\vec{G}_v$, and on the surface through tensor $\mathbf{G}_s$.

$$\int_{\Omega} \vec{G}_v d\Omega + \oint_{\partial\Omega} \left(\mathbf{G}_s \cdot \hat{n}\right) dS \tag{2.4}$$

Combining equations 2.1 - 2.4, one can obtain the complete equation for a general conservation law for a fluid element.

$$\frac{\partial}{\partial t}\int_{\Omega} \phi d\Omega = -\oint_{\partial\Omega} \phi \left(\vec{v} \cdot \hat{n}\right) dS + \oint_{\partial\Omega} \kappa\rho \left(\nabla\phi \cdot \hat{n}\right) dS + \int_{\Omega} \vec{G}_v d\Omega + \oint_{\partial\Omega} \left(\mathbf{G}_s \cdot \hat{n}\right) dS \tag{2.5}$$

2.1.1 Conservation of Mass

The first conservation law of the Navier-Stokes equations is the conservation of mass. Here the conserved quantity is mass which may written in it's volume specific form as density, ρ, so it can be used in the generalized form of equation 2.5. There is no diffusive flux for the conservation of mass equation. Additionally, in the flow solvers UNS3D and UNS3D TL, the conservation of mass equations do not include any source terms, thus it will be neglected in this formulation. The conservation of mass for this paper is written as:

$$\frac{\partial}{\partial t}\int_{\Omega} \rho d\Omega + \oint_{\partial\Omega} \rho \left(\vec{v} \cdot \hat{n}\right) dS = 0 \tag{2.6}$$

2.1.2 Conservation of Momentum

The second conservation law of the Navier-Stokes equations is the conservation of momentum. The conserved quantity is momentum, $\vec{p}$, which is vector equal to mass, m, multiplied by the velocity vector $\vec{v}$ such that $\vec{p} = m\vec{v}$. For a control volume formulation, the volume specific form of momentum that will be used as the conservation variable will be $\rho\vec{v}$.

Following the same form for the general conservation law as equation 2.5, the conservation of momentum is written as

$$\frac{\partial}{\partial t}\int_{\Omega} \rho\vec{v} d\Omega + \oint_{d\Omega} \rho\vec{v}\left(\vec{v}\cdot\hat{n}\right) dS = \oint \left(-p\mathbf{I} + \boldsymbol{\tau}\right)\cdot\hat{n} dS \tag{2.7}$$

where the scalar p is the fluid pressure, $\mathbf{I}$ is the identity tensor and $\boldsymbol{\tau}$ is the viscous stress tensor. The viscous stress tensor is a symmetric tensor written as

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix} \tag{2.8}$$

with its components defined as

$$\begin{aligned}
\tau_{xx} &= \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z}\right) \\
\tau_{yy} &= \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} - \frac{\partial u}{\partial x}\right) \\
\tau_{zz} &= \frac{2}{3}\mu\left(2\frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right) \\
\tau_{xy} &= \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\
\tau_{xz} &= \tau_{zx} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\
\tau_{yz} &= \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)
\end{aligned}$$

where u, v, and w are the velocity components in the x, y, and z directions, respectively. The dynamic viscosity coefficient is reffered as μ.

2.1.3 Conservation of Energy

The last conservation law in the Navier-Stokes equations is the conservation of energy, also known as the 1st Law of Thermodynamics. Here, the total energy per unit volume is conserved. The total energy of the system is written as

$$\rho E = \rho \left(e + \frac{|\vec{v}|^2}{2} \right) \tag{2.9}$$

where e is the internal energy per unit mass and $\frac{|v|^2}{2}$ is the kinetic energy per unit mass. Within this paper, the fluid is assumed to be a calorically perfect gas therefore

$$e = c_v T \tag{2.10}$$

where c_v is the specific heat at constant volume and T is the temperature of the fluid. The flux term for the conservation of energy includes both a convective and diffusive flux term. The convective flux for the conservation of energy, $F_{E.conv}$, is written as

$$F_{E.conv.} = \oint_{\partial\Omega} -\rho E \left(\vec{v} \cdot \hat{n} \right) dS \tag{2.11}$$

where the diffusive flux for the conservation of energy, $F_{E.diff.}$, is written as

$$F_{E.diff.} = \oint_{\partial\Omega} \gamma \rho \kappa \left(\nabla e \cdot \hat{n} \right) dS \tag{2.12}$$

where κ is the thermal diffusivity coefficient and γ is the ratio of specific heat ($\gamma = \frac{c_p}{c_v}$). The diffusive flux may be rewritten as Fourier's law of heat transfer such that ∇T is used instead of ∇e.

$$F_{E.diff.} = \oint_{\partial\Omega} \gamma k \left(\nabla T \cdot \hat{n} \right) dS \tag{2.13}$$

where k is the thermal conductivity and T is the total temperature of the fluid. The scope of this paper does not include body sources for the conservation of energy, but does include the surface source terms. The surface source term for the conservation of energy is as follows

$$\vec{G}_s = -p\vec{v} + \boldsymbol{\tau} \cdot \vec{v}. \tag{2.14}$$

The complete equation for the conservation of energy is written as

$$\frac{\partial}{\partial t} \int_{\Omega} \rho E d\Omega + \oint_{\partial\Omega} \rho E \left(\vec{v} \cdot \hat{n}\right) dS = \oint_{\partial\Omega} k \left(\nabla T \cdot \hat{n}\right) - p \left(\vec{v} \cdot \hat{n}\right) + \left(\boldsymbol{\tau} \cdot \vec{v}\right) \hat{n} dS \tag{2.15}$$

The conservation of energy can also be written in terms of enthalpy, where total enthalpy, H, is defined as

$$H = \left(h + \frac{|\vec{v}|^2}{2}\right) = E + \frac{p}{\rho} \tag{2.16}$$

where h is the internal enthalpy and is defined as

$$h = c_p T \tag{2.17}$$

for a calorically perfect gas, where c_p is the specific heat at constant pressure.

The conservation of energy in terms of enthalpy, H, is written as

$$\frac{\partial}{\partial t} \int_{\Omega} \rho E d\Omega + \oint_{\partial\Omega} \rho H \left(\vec{v} \cdot \hat{n}\right) dS = \oint_{\partial\Omega} k \left(\nabla T \cdot \hat{n}\right) + \left(\boldsymbol{\tau} \cdot \vec{v}\right) \hat{n} dS \tag{2.18}$$

2.2 Turbulence Models

Turbulence modeling is essential for producing accurate results when using a Reynolds-Averaged Navier-Stokes (RANS) flow solver. This is due to how the RANS equations are formed. Reynolds averaging is the method in which the flow variables are decomposed into a mean and fluctuating component.

$$\phi = \bar{\phi} + \phi' \tag{2.19}$$

Equation 2.19 shows the Reynolds-averaging of an arbitrary flow variable ϕ, where $\bar{\phi}$ is the mean flow variable and ϕ' is the fluctuating component. There are three different methods to produce the mean components.

Time-averaging:

$$\bar{\phi} = \lim_{T \longrightarrow \infty} \frac{1}{T} \int_{t}^{t+T} \phi dt \tag{2.20}$$

Using time-averaging produces steady-state results for the mean flow variables. Time-averaging requires that the time interval T is much larger compared to the time scale of the turbulent fluctuations.

Spatial-averaging:

$$\bar{\phi} = \lim_{\Omega \longrightarrow \infty} \frac{1}{\Omega} \int_{\Omega} \phi d\Omega \tag{2.21}$$

Spatial-averaging averages across the control volume Ω. Here the flow variables are uniform in space but can vary in time.

Ensemble-averaging:

$$\bar{\phi} = \lim_{N \longrightarrow \infty} \frac{1}{N} \sum_{m=1}^{N} \phi \tag{2.22}$$

Ensemble averaging produces mean values that are a function of spatial coordinates and time.

For all methods of Reynolds-averaging, the fluctuating components have the following properties:

$$\overline{\phi'} = 0 \qquad \overline{\phi'_i \phi'_i} \neq 0$$

$$\overline{\phi'_i \phi'_j} \neq 0 \qquad \rightarrow \text{if both turbulent velocity components are correlated}$$

It is also typical to use what is known as Favre-averaging, which can be thought of as a density-weighted Reynolds averaging, where

$$\phi = \widehat{\phi} + \phi'' \tag{2.23}$$

and

$$\widehat{\phi} = \frac{1}{\rho} \lim_{T \longrightarrow \infty} \frac{1}{T} \int_{t}^{T+t} \rho \phi dt. \tag{2.24}$$

Similar to Reynolds averaging, the properties of the fluctuating terms for Favre-averaging are listed as follows

$$\widehat{\phi''} = 0 \qquad \widehat{\phi_i'' \phi_i''} \neq 0 \qquad \widehat{\phi_i \phi_j} \neq 0 \tag{2.25}$$

Mixing both Favre and Reynolds-averaging, the following properties are obtained.

$$\widehat{\rho \phi} = \overline{\rho} \widehat{\phi} \qquad \overline{\rho \phi''} = 0 \qquad \overline{\phi''} \neq 0 \tag{2.26}$$

Applying both Reynolds-averaging and Favre-averaging to the Navier-Stokes equations, one obtains the the Reynolds-Averaged Navier-Stokes equations. The mixing of the two averaging methods allows one to eliminate the some of the density fluctuation terms that would have appeared. Here, the Reynolds-averaging is applied to the density, ρ, and pressure, p, terms while the energy, enthalpy and velocity terms are Favre-averaged. Listed in differential form, the RANS equations are

$$\begin{aligned}
\frac{\partial \overline{\rho}}{\partial t} + \frac{\partial \overline{\rho} \widehat{v}_i}{\partial x_i} &= 0 \\
\frac{\partial \overline{\rho} \widehat{v}_i}{\partial t} + \frac{\partial}{\partial x_i} \left(\overline{\rho} \widehat{v}_j \widehat{v}_i \right) &= -\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial}{\partial x_i} \left(\widehat{\tau}_{ij} - \overline{\rho} \widehat{v_i'' v_j''} \right) \\
\frac{\partial \overline{\rho} \widehat{E}}{\partial t} + \frac{\partial}{\partial x_j} \left(\overline{\rho} \widehat{v}_j \widehat{H} \right) &= \frac{\partial}{\partial x_j} \left(k \frac{\partial \widehat{T}}{\partial x_j} - \overline{\rho} \widehat{v_j'' h''} + \widehat{\tau_{ij} \widehat{v}_i''} - \overline{\rho} \widehat{v_j'' K} \right) + \frac{\partial}{\partial x_j} \left[\widehat{v}_i \left(\widehat{\tau_{ij}} - \overline{\rho} \widehat{v_i'' v_j''} \right) \right]
\end{aligned}$$

The RANS equations are similar to the original Navier-Stokes equations except for a few additional fluctuation terms. These fluctuation terms represent the effects of turbulence in the RANS equations. In the momentum equations, there is an additional term known as the Favre-Averaged Reynolds stress tensor, τ_{ij}^F, which is

$$\tau_{ij}^F = -\overline{\rho}\widehat{v_i''v_j''}. \tag{2.27}$$

Most of the effects of turbulence can be seen in the equation for the conservation of energy. There is a new term, $\widehat{K}$, which is defined as the Favre-averaged turbulent kinetic energy and expressed as

$$\widehat{K} = \frac{1}{2}\widehat{v_i''v_j''}. \tag{2.28}$$

The effects of turbulence in the conservation of energy equation can be physically described as follows:

$$\frac{\partial}{\partial x_j}\left(\overline{\rho}\widehat{v_j''h''}\right) \rightarrow \text{ Turbulent Transport of Heat}$$
$$\frac{\partial}{\partial x_j}\left(\widehat{\tau_{ij}v_i''}\right) \rightarrow \text{Molecular Diffusion of } \widehat{K}$$
$$\frac{\partial}{\partial x_j}\left(\overline{\rho}\widehat{v_j''K}\right) \rightarrow \text{Turbulent Transport of } \widehat{K}$$
$$\frac{\partial}{\partial x_j}\left(\widehat{v}_i\tau_{ij}^F\right) \rightarrow \text{Work done by Favre-Average Stress Reynolds Stress.}$$

To solve the RANS equations, one must be able to calculate or approximate the Favre-averaged Reynolds stress tensor, τ_{ij}^F, which is composed of six unknowns, and the Reynolds heat flux $\overline{\rho}\widehat{h''v_j''}$, which has 3 unknowns. The Boussinesq eddy viscosity assumption is the most common method to provide closure to the RANS equations. This assumption observes that the Favre-average Reynolds stress tensor is similar to the mean stress tensor τ_{ij}. The Boussinesq eddy viscosity approximation assumes that the Reynolds stress tensor is proportional to the trace-less mean strain rate tensor such that

$$\tau_{ij}^F = -\overline{\rho}\widehat{v_i''v_j''} = \mu_T\left(\frac{\partial\widehat{v}_i}{\partial x_j} + \frac{\partial\widehat{v}_j}{\partial x_i} - \frac{2}{3}\frac{\partial\widehat{v}_k}{\partial x_k}\delta_{ij}\right) - \overline{\rho}\widehat{K}\delta_{ij} \tag{2.29}$$

and the Reynolds heat flux can be provided by the following equation

$$\overline{\rho}\widehat{h''v_j''} = \frac{c_p \mu_T}{Pr_T}\frac{\partial \widehat{T}}{\partial x_j} \tag{2.30}$$

where Pr_T is the turbulent Prandtl number and μ_T is known as the turbulent eddy viscosity. The value of the turbulent eddy viscosity μ_T is determined by a turbulence model. Most turbulence models solve one or more transport equations or approximates a value for the turbulent kinetic energy $\widehat{K}$, but it is important to note that not all turbulence models solve for $\widehat{K}$, and it may or may not be included in the Boussinesq eddy viscosity assumption. The following sections described the two turbulence models that are in UNS3D: The Shear Stress Transport turbulence model and the Spalart-Allmaras turbulence model that has been implemented in this research.

2.2.1 Shear Stress Transport Turbulence Model

The Shear Stress Transport turbulence Menter (1994) model is a two equation turbulence model that is a mixture of a high Reynolds number $\kappa - \epsilon_t$ turbulence model and the $\kappa - \omega_t$ turbulence model of Wilcox. The goal of merging the two models is to combine the best attributes of both models while fixing their weaknesses. For example, the $\kappa - \omega_t$ model is sensitive to the freestream values of ω_t, thus the SST model switches to the $\kappa - \epsilon_t$ model to resolve this issue. The Shear Stress Transport turbulence model was the only pre-existing turbulence model in the flow solver UNS3D. The Shear Stress Transport model is defined by the transport equations below:

$$\frac{\partial \rho\kappa}{\partial t} + \frac{\partial(\rho v_j \kappa)}{\partial x_j} = \frac{\partial}{\partial x_j}[(\mu_L + \sigma_\kappa \mu_T)\frac{\partial \kappa}{\partial x_j}] + \tau_{ij}^F S_{ij} - \beta^* \rho \omega_t \kappa \tag{2.31}$$

$$\frac{\partial \rho\omega_t}{\partial t} + \frac{\partial(\rho v_j \omega_t)}{\partial x_j} = \frac{\partial}{\partial x_j}[(\mu_L + \sigma_{\omega_t} \mu_T)\frac{\partial \omega_t}{\partial x_j}] + \tau_{ij}^F S_{ij}\frac{C_\omega \rho}{\mu_T} - \beta\rho\omega_t^2 + 2(1-f_1)\frac{\rho\sigma_{\omega t2}}{\omega_t}\frac{\partial \kappa}{\partial x_j}\frac{\partial \omega_t}{\partial x_j} \tag{2.32}$$

The first equation (2.31) is the transport equation for turbulent kinetic energy κ, and the latter is the transport equation for the specific dissipation ω_t. f_1 is the blending function between the $\kappa - \epsilon_t$ and the $\kappa - \omega_t$ turbulence models. The Favre-averaged turbulent shear stresses, $\boldsymbol{\tau}^F$, are given by

$$\boldsymbol{\tau}_{ij}^F = 2\mu_T \tilde{S_{ij}} - \left(\frac{2\mu_T}{3}\right)\frac{\partial \tilde{v}_k}{\partial x_k}\delta_{ij} - \frac{2}{3}\bar{\rho}\tilde{K}\delta_{ij} \tag{2.33}$$

The turbulent eddy viscosity, μ_T, is given by

$$\mu_T = \frac{a_1 \rho \kappa}{max\left(a_1 \omega_T, f_2 ||curl \vec{v}||_2\right)} \tag{2.34}$$

The blending function f_1 is defined as

$$f_1 = tanh\left(arg_1^4\right) \tag{2.35}$$

where arg_1 is defined as

$$arg_1 = min\left[max\left(\frac{\sqrt{\kappa}}{0.09\omega_T d}, \frac{500\mu_L}{\rho\omega_T d^2}\right), \frac{4\rho\sigma_{\omega 2}\kappa}{CD_{\kappa\omega}d^2}\right]. \tag{2.36}$$

The rest of the functions and variables are defined as follows:

$$CD_{\kappa\omega} = max\left(2\frac{\rho\sigma_{\omega 2}}{\omega_T}\frac{\partial\kappa}{x_j}\frac{\partial\omega_T}{\partial x_j}, 10^{-20}\right) \tag{2.37}$$

$$f_2 = tanh\left(arg_2^2\right) \tag{2.38}$$

$$arg_2 = max\left(\frac{2\sqrt{\kappa}}{0.09\omega_T d}, \frac{500\mu_L}{\rho\omega_T d^2}\right). \tag{2.39}$$

The variables that are constant across both the $\kappa - \omega_T$ and $\kappa - \epsilon_T$ turbulence models are

$$a_1 = 0.31 \qquad \beta^* = 0.09 \qquad \kappa^* = 0.41. \tag{2.40}$$

The coefficients $\beta, C_\omega, \sigma_\kappa, \sigma_\omega$ within the two transport equations are also blended using the following function where ϕ_1 is the value of one of the constants from the $\kappa - \omega_T$ turbulence model and ϕ_2 is the value of one of the constants from the $\kappa - \epsilon_T$ two-equation turbulence model.

$$\phi = f_1 \phi_1 + (1 - f_1)\,\phi_2 \tag{2.41}$$

The coefficients for the $\kappa - \omega_T$ turbulence model are:

$$\sigma_{\kappa 1} = 0.85 \qquad \sigma_{\omega 1} = 0.5 \qquad \beta_1 = 0.075 \qquad C_{\omega 1} = \frac{\beta_1}{\beta^*} - \frac{\sigma_{\omega 1} \kappa^{*2}}{\sqrt{\beta^*}} = 0.533 \tag{2.42}$$

The coefficients for the $\kappa - \epsilon_T$ turbulence model are:

$$\sigma_{\kappa 2} = 1.0 \qquad \sigma_{\omega 2} = 0.856 \qquad \beta_2 = 0.0828 \qquad C_{\omega 2} = \frac{\beta_2}{\beta^*} - \frac{\sigma_{\omega 2} \kappa^{*2}}{\sqrt{\beta^*}} = 0.440 \tag{2.43}$$

This blending allows the Shear Stress transport turbulence model to switch between the two turbulence models depending on distance from the wall and which region of the turbulent boundary layer is being analyzed.

2.2.2 Spalart-Allmaras Turbulence Model

The Spalart-Allmaras turbulence model Spalart and Allmaras (1992) is a one-equation turbulence model which solves the transport equation of a eddy-viscosity like variable, $\tilde{\nu}$. The Spalart-Allmaras turbulence model is a popular turbulence model for turbomachinery flows as it was designed for wall-bounded flows. Since the Spalart-Allmaras turbulence model only solves for one equation, it is also inherently cheaper than SST.

The Spalart-Allmaras is a one-equation turbulence model which is a transport equation using the viscosity-like variable, $\tilde{\nu}$, as it's working variable. P and D are the production and destruction source terms in the equation.

$$\frac{D\tilde{\nu}}{Dt} = P - D + \frac{1}{\sigma}\left[\nabla \cdot ((\nu_L + \tilde{\nu})\nabla\tilde{\nu}) + c_{b2}(\nabla\tilde{\nu})^2\right] \tag{2.44}$$

$$P = c_{b1}\left(1 - f_{t2}\right)\tilde{S}\tilde{\nu} \tag{2.45}$$

$$D = \left(c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{v2}\right)\left[\frac{\tilde{\nu}}{d}\right]^2 \tag{2.46}$$

ν_L is the laminar kinematic viscosity, and d is the distance to the wall. The turbulent eddy dynamic viscosity is calculated as:

$$\mu_T = \rho\tilde{\nu} f_{v1}$$

Where ρ is the density and the function f_{v1} is defined as:

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

The following functions for the Spalart-Allmaras equation are defined below.

$$\chi = \frac{\tilde{\nu}}{\nu_L} \qquad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \qquad f_w = g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^3}\right]^{\frac{1}{6}}$$

$$g = r + c_{w2}\left(r^6 - r\right) \qquad r = min\left[\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, 10\right] \qquad f_{t2} = c_{t3} e^{-c_{t4}\chi^2}$$

$$W_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)$$

The turbulence model has an additional trip term, f_{t1}, used to model transition into turbulence. Research has found that the two trip terms, f_{t1} and f_{t2}, are unnecessary to simulate fully turbulent flows (Allmaras et al., 2012). The implementation used in this research does not use f_{t1}. It is important to note that r is defined as a minimum function in the Spalart-Allmaras turbulence model. While it was stated that these functions make the Shear-Stress Transport difficult to linearize, it is not the case for the Spalart-Allmaras model, as r does not need to have a minimum value (Spalart and Allmaras, 1992), and because f_w reaches a maximum for any value of r greater than 10. The following constants for the Spalart-Allmaras equations are defined below.

$$c_{b1} = 0.1355 \qquad \sigma = \frac{2}{3} \qquad c_{b2} = 0.622 \qquad \kappa = 0.41 \qquad c_{w2} = 0.3$$
$$c_{w3} = 2 \qquad c_{v1} = 7.1 \qquad c_{t3} = 1.2 \qquad c_{t4} = 0.5 \qquad c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}$$

The version of the Spalart-Allmaras turbulence model being implemented in this research is the compressible version with the modifications to the modified vorticity term stated in Allmaras et al. (2012). So the modified vorticity term $\tilde{S}$ is:

$$\tilde{S} = \Omega + \bar{S} \qquad \text{if} \quad \bar{S} \geq -c_2\Omega$$
$$\tilde{S} = \Omega + \frac{\Omega(c_2^2\Omega + c_3\bar{S})}{(c_3 - 2c_2)\Omega - \bar{S}} \qquad \text{if} \quad \bar{S} < -c_2\Omega$$

With Ω being the magnitude of vorticity defined as $\Omega = \sqrt{W_{ij}W_{ij}}$, $\bar{S} = \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}$ and the constants $c_2 = 0.7$ and $c_3 = 0.9$.:

To form the compressible version of the turbulence model, the Spalart-Allmaras turbulence model is combined with the mass conservation equation, which turns out to be density multiplied by the turbulence model plus $\tilde{\nu}$ times the Conservation of Mass equation.

$$0 = \rho * SA + \tilde{\nu} * Mass \tag{2.47}$$

$$0 = \frac{\partial \rho\tilde{\nu}}{\partial t} + \nabla \cdot (\rho\vec{u}\tilde{\nu}) - \rho\,(P - D) - \frac{1}{\sigma}\nabla \cdot [\rho(\nu_L + \tilde{\nu})\nabla\tilde{\nu}] - \frac{c_{b2}}{\sigma}\rho(\nabla\tilde{\nu})^2 + \frac{1}{\sigma}(\nu_L + \tilde{\nu})\nabla\rho \cdot \nabla\tilde{\nu} \tag{2.48}$$

Following the same manner to non-dimensionalize the conservation equations, with the additional two nondimensionalized terms:

$$\tilde{\nu}^* = \tilde{\nu}\frac{\rho_\infty}{\nu_\infty} \qquad \tilde{S}^* = \tilde{S}\frac{L_\infty^2}{\nu_\infty} \tag{2.49}$$

The non-dimensionalized compressible turbulence model is of the form:

$$\begin{aligned}
&\frac{\partial \rho^* \tilde{\nu}^*}{\partial t^*} + \nabla^* \cdot \left(\rho^* \vec{u}^* \tilde{\nu}^*\right) \\
&= \frac{1}{Re}\left[\rho^*\left(P^* - D^*\right) + \frac{1}{\sigma}\nabla^* \cdot \left[\rho^*(\nu_L^* + \tilde{\nu}^*)\nabla^*\tilde{\nu}^*\right] + \frac{c_{b2}}{\sigma}\rho^*(\nabla^*\tilde{\nu}^*)^2 - \frac{1}{\sigma}(\nu_L^* + \tilde{\nu}^*)\nabla^*\rho^* \cdot \nabla^*\tilde{\nu}^*\right]
\end{aligned}$$

where Re is the Reynolds number.

3. TIME-LINEARIZATION

Time-linearization is a method that predicts unsteady, periodic flows at a greater computational efficiency than solving for an unsteady solution using a RANS solver. The time linearization method eliminates the need of solving the transient solution to obtain a periodic solution by solving the unsteady flow features in the frequency domain. By solving in the frequency domain, time-linearized flow solvers may also reduce the computational domain if the computational domain is axis-symmetric. The implementation of the time-linearized solver can solve the unsteady periodic solution of a flow field with harmonically moving walls or a flow field that has harmonically oscillating flow features such as an oscillating back pressure. The time-linearized flow solver solves for harmonically moving walls using a coordinate transformation described in the next section.

3.1 Coordinate Transformation

In the present analysis, the computational grid is allowed to undergo small rigid body and elastic deformations. The motion of the boundaries is assumed to be small and harmonic. Consider a coordinate system ξ, η, ζ that is nearly the same as the fixed in space coordinate system x, y, z, but which differs by a small perturbation so that the deforming grid can be considered a small perturbation about the fixed grid

$$\begin{aligned}
x(\xi,\eta,\zeta,\tau) &= \xi + f(\xi,\eta,\zeta,\tau) \\
y(\xi,\eta,\zeta,\tau) &= \eta + g(\xi,\eta,\zeta,\tau) \\
z(\xi,\eta,\zeta,\tau) &= \zeta + h(\xi,\eta,\zeta,\tau) \\
t(\xi,\eta,\zeta,\tau) &= \tau
\end{aligned} \tag{3.1}$$

The coordinate ξ, η and ζ are the coordinates fixed to the moving grid and to zeroth order are equal to x, y and z, respectively. The first-order perturbation functions f, g and h describe how the grid deforms and are specified so that the boundary of the grid conforms to the motion of the boundary, such as the shaft of a rotor.

3.2 Harmonically-Varying, Small Disturbance Approximation

The dependent variables of the governing equations are represented in a form that is similar to equation (3.1), as the sum between a mean and a perturbation, both of which are functions of the transformed coordinate system

$$\phi(\xi,\eta,\zeta,\tau) = \phi_0(\xi,\eta,\zeta) + \phi_1(\xi,\eta,\zeta,\tau) \tag{3.2}$$

where ϕ_0, the steady component is known from the solution of the steady flow, and ϕ_1 is the as yet unknown perturbation. Note that although ϕ_0 appears steady in the (ξ,η,ζ) coordinate system, in the (x,y,z) coordinate system ϕ_0 is unsteady (Hall, 1993). In the case of a Reynolds-averaged Navier–Stokes (RANS) equations model written as a function of primitive state vector, the representation (3.2) applies to all components of the state vector, that is, ρ, u, v, w and p. If the turbulent flow is modeled by the Spalart-Allmaras model, then representation (3.2) applies also to the dependent variables $\tilde{\nu}$, which is the viscosity-like variable used in the transport equation of the Spalart-Allmaras turbulence model.

A further simplification is obtained by conveniently assuming that the perturbation variables undergo simple harmonic motion. This assumption is supported by the fact that most unsteady flows in turbomachinery are temporally periodic. Consequently, the unsteady flow can be represented as a Fourier series in time. Using the fact that the governing small disturbance equations are linear, each Fourier mode can be analyzed individually; then all modes can be summed to form the complete solution. As a result, it is assumed, without loss of generality, that

$$\phi_1(\xi,\eta,\zeta,\tau) \rightarrow Re\left[\phi_1(\xi,\eta,\zeta)e^{j\Omega t}\right] = \frac{1}{2}\left[\phi_1(\xi,\eta,\zeta)e^{j\Omega t} + \overline{\phi}_1(\xi,\eta,\zeta)e^{-j\Omega t}\right] \tag{3.3}$$

where $\phi_1(\xi,\eta,\zeta)$ is the complex amplitude of the perturbation, $\overline{\phi}_1$ is the complex conjugate of $\phi_1(\xi,\eta,\zeta)$, and Ω is the rotor whirling frequency.

An alternative notation to (3.2) is

$$\phi(\xi, \eta, \zeta, \tau) = \phi_0(\xi, \eta, \zeta) + \epsilon\phi_1(\xi, \eta, \zeta)e^{j\Omega t} \tag{3.4}$$

where ϵ has a small value, and the magnitudes of ϕ_0 and ϕ_1 are of the same order. Note, however, that $\phi_0 \in \mathbb{R}$ while $\phi_1 \in \mathbb{C}$. The 0 and 1 subscripts denote zeroth- and first-order terms.

Further substitutions of (3.4) can be made by using a truncated Taylor series expansion. Since all terms in the Navier-Stokes equations are directly or indirectly defined by state variables, $\vec{Q}$, and the mesh node locations, $\vec{x}$, one can show through Taylor series expansion that equation (3.4) is equivalent to:

$$\phi = \phi_0 + \epsilon \left[\frac{\partial \phi}{\partial \vec{x}} \vec{x}_1 + \frac{\partial \phi}{\partial \vec{Q}} \vec{Q}_1 \right] e^{j\omega t} \tag{3.5}$$

By substituting (3.5) in the governing equations for a finite-volume formulation, and collecting the zeroth-order terms, one obtains the steady, nonlinear, governing equations. By collecting the first-order terms, one obtains the linear, complex, small disturbance equations. Consequently, in order to calculate the unsteady flow, one first solves the steady, nonlinear, governing equations, which yield the steady (or mean) solution. Then, the following small disturbance linear, complex equations are solved,

$$\left[j\omega\Omega_{i,0} - \frac{\partial \vec{R}_i}{\partial \vec{Q}} \right] \vec{Q}_1 = - \left[\Omega_i \frac{\partial \vec{Q}_i}{\partial \vec{x}} + \vec{Q}_{i,0} \frac{\partial \Omega_i}{\partial \vec{x}} - \frac{\partial \vec{R}_i}{\partial \vec{x}} \right] \vec{x}_1 \tag{3.6}$$

where $\vec{R}$ are the residuals of the governing equations, and Ω is the control volume of a cell. The unsteady solution is obtained by summing the mean and small perturbation solutions using (3.5).

As shown above, the time-linearized flow solver requires three inputs: a steady-state solution, Q_0, obtained by the steady, nonlinear governing equations, a prescribed grid deformation, $\vec{x}_1$, and a chosen harmonic frequency ω. The following section describes the method in which the grid deformations, $\vec{x}_1$, are calculated.

3.3 Grid Generator for Moving Meshes using Radial Basis Functions

Originally, the initial values for the perturbed grid were calculated using case-specific algebraic expressions which were created by the user, often a function of the geometry of the object to be simulated. This method works well with simple geometries but can be extremely difficult and time consuming if simulating complex geometries such as an impeller blade cascade. A general perturbed grid solver was needed to simulate such complex geometries, and to eliminate the hassle of creating case-specific grids.

Mesh deformation based on radial function interpolations (de Boer et al., 2007) was chosen to be the method in which the perturb grids were to be generated. This method was additionally chosen as UNS3D already had an existing mesh deformation tool using radial basis functions, which has proven to be successful.

The method used to deform a mesh involves creating an interpolation function, s, which describes the displacement of the nodes in the whole domain. The interpolation function is approximated by a sum of basis functions:

$$s(\mathbf{x}) = \sum_{j=1}^{n_b} \alpha_j \phi(||\mathbf{x} - \mathbf{x}_{b_j}||) + p(\mathbf{x}) \tag{3.7}$$

where $\mathbf{x} \in \mathbb{R}^3$, and is the position vector of the interior nodes, $\mathbf{x}_{b_j}$, is the position vector of the boundaries, ϕ is the chosen radial basis function with respect to the euclidean distance, and $p(\mathbf{x})$ is a polynomial. Polynomial $p(\mathbf{x})$ depends on which basis function was chosen, and can be determined along with α_j by the interpolation conditions:

$$s(\mathbf{x}_{b_j}) = \mathbf{d}_{b_j}, \tag{3.8}$$

$$\sum_{j=1}^{n_b} \alpha_j q(\mathbf{x}_{b_j}) = 0 \tag{3.9}$$

where $\mathbf{d}_{b_j} \in \mathbb{C}$, and are the known displacements of the boundaries. Since the time-linearized solver works in the complex domain, $\mathbf{d}_{b_j}$ and α_j will be complex, but the radial basis functions will not, as they are a function of the steady grid's euclidean distances. The coefficients α_j and the polynomial, p, can be obtained by solving the complex system:

$$\begin{bmatrix} \mathbf{d}_{b_j} \\ 0 \end{bmatrix} = \begin{bmatrix} M_{b,b} & P_b \\ P_b^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} \tag{3.10}$$

where β are the coefficients of the polynomial p. $M_{b,b}$ is an n_b x n_b matrix containing the evaluation of the basis function ϕ_{b_i,b_j}, and P_b is an n_b x 4 matrix with row j given by $\begin{bmatrix} 1 & x_{b_j} & y_{b_j} & z_{b_j} \end{bmatrix}$. The system is solved using complex solvers in LAPACK. Once the values for α and β are solved for, one can calculate the deformation of the interior points using :

$$s(\mathbf{x}_{in}) = \mathbf{d}_{in} \tag{3.11}$$

3.4 Time-Linearized Spalart-Allmaras Turbulence Model

The linearized turbulence model has been applied the same way as the full order turbulence model in our in-house code UNS3D, where the turbulent transport equations are solved after the Navier-Stokes equations, updating the turbulent eddy viscosity used for solving the Navier-Stokes equations in the next iteration. The Spalart-Allmaras turbulence model will be linearized using the same method as presented in equation (3.4). The time-linearized Spalart-Allmaras turbulence model is of the form:

$$\left(\frac{D\tilde{\nu}}{Dt}\right)_1 = [j\Omega\tilde{\nu}_1 + u_{j,0}\frac{\partial\tilde{\nu}_1}{\partial x_j} + u_{j,1}\frac{\partial\tilde{\nu}_0}{\partial x_j}] \tag{3.12}$$

$$P_1 = (\tilde{S}_0\tilde{\nu}_1 + \tilde{S}_1\tilde{\nu}_0) \tag{3.13}$$

$$D_1 = c_{w1}\left[f_{w,0}(\frac{2\tilde{\nu}_0\tilde{\nu}_1}{d_0^2} - \frac{2\tilde{\nu}_0^2 d_1}{d_0^3}) + f_{w,1}\frac{\tilde{\nu}_0^2}{d_0^2}\right] \tag{3.14}$$

$$\vec{F}_{v,1} = \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}(\nu_{L,0} + \tilde{\nu}_0)\frac{\partial\tilde{\nu}_1}{\partial x_j} + (\nu_{L,1} + \tilde{\nu}_1)\frac{\partial\tilde{\nu}_0}{\partial x_j} + 2c_{b1}\frac{\partial\tilde{\nu}_0}{\partial x_i}\frac{\partial\tilde{\nu}_1}{\partial x_i}\right] \tag{3.15}$$

$$\left(\frac{D\tilde{\nu}}{Dt}\right)_1 = P_1 - D_1 + \vec{F}_{v,1} \tag{3.16}$$

It is important to note that every variable and function in the Spalart-Allmaras turbulence model has been linearized in this implementation. In this implementation, there is only one term which is a conditional term and that is the modified vorticity, $\tilde{S}$. Specifically, the conditional term is the modification to prevent negative values of the modified vorticity presented in (Allmaras et al., 2012). For this conditional term, it was chosen to use the same functions to determine modified vorticity $\tilde{S}$ that the steady state run used.

Conditional terms such as if, min and max statements are problematic for time-linearization as the conditional terms are time dependent and cannot be assumed to be entirely harmonic. For example, the Shear Stress Transport turbulence model has many conditional terms which include the distance to wall terms. If there is an airfoil undergoing pitching or plunging motion the distance to wall value for a node will change with time. In time-linearization, only the complex amplitude of the distance to wall is considered. Thus, the time-linearized flow solver only takes into consideration the maximum amplitude of the distance to the wall, where there is only one value for the conditional term, whereas the conditional term may switch values during a time-dependent simulation. For this reason, the Shear Stress Transport turbulence model will not be used for the time-linearized flow solver, as it is highly dependent on conditional terms and will affect the switching between the $\kappa - \omega$ and the $\kappa - \epsilon$ turbulence models.

4. NUMERICAL MODELS

The Reynolds-Averaged Navier-Stokes (RANS) equations are solved using the in-house flow solver UNS3D. UNS3D stands for Unstructured, Unsteady Three-Dimensional flow solver. This section outlines the various numerical methods and discretization used in UNS3D. The majority of these methods are the same as the time-linearized flow solver UNS3D TL. The time-linearized flow solver requires the same numerical methods as the full-order solver, as the time-linearized flow solver requires a fully converged steady-state input from the full-order flow solver.

4.1 Spatial Discretization

The UNS3D flow solvers use the finite volume method for its spatial discretization. The finite volume method is a method in which the domain of interest is discretized into a set of non-overlapping control volumes where the integral form of the Navier-Stokes equations are solved for each individual control volume. The control volumes form a computational mesh or grid. The two most common methods of forming control volumes from computational grid is the cell-centered method and the cell-vertex method, also known as the dual-mesh method. The cell centered method uses the same cells provided by the computational grid, where information is stored within the center of the cells. The cell-vertex method, or the dual-mesh method, forms a secondary mesh where the nodes of the computational grid are the center of the control volumes for the secondary mesh. UNS3D uses the dual-mesh approach as it has better adaptability with mixed element grids and is computationally cheaper than the cell-centered method.

4.1.1 Convective Flux

As described in the previous chapter, the Navier-Stokes equations, written in integral form, are comprised of a surface integral and a volume integral. The surface integral requires that one accounts for the flux of a conservative variable, such as momentum, going through the surfaces of the control volume. The flux of a conserved variable is composed of two components: the convective flux and diffusive flux.

The convective flux describes the amount of the conserved quantity, ϕ, going through the control surface due the the motion of the fluid. The convective flux, F_C described as

$$F_C = -\oint_{\partial\Omega} \phi\left(\vec{v} \cdot \tilde{n}\right) dS. \tag{4.1}$$

Since the values of the convective flux are stored within the edges of the control volume, a numerical scheme is required to determine the fluxes at the edges of the control volume from the nodal-based state vector. The simplest method to determine the edge based flux would be a central scheme, which is simply taking the average between two nodes. Central schemes are unstable, thus a modification or another scheme must be used to calculate the convective flux. Central schemes can be modified to add artificial dissipation, such as Jameson's scheme (Jameson et al., 1981), to stabilize the numerical solver. Another set of schemes known as upwind schemes have been developed to have a stable method of calculating the convective fluxes. Upwind schemes utilizes the information provided by the flow field characteristics to ensure the flow travels in the correct direction. To calculate the convective flux, this research utilizes an upwind scheme developed by Roe (1981) with the additional entropy correction developed by Harten (1983), known as the Roe's Flux-Difference Splitting scheme with the Harten Entropy Fix. An upwind scheme was chosen for it's stability and ability to capture shock waves and boundary layers accurately. Flux-difference schemes are upwind schemes that were developed to solve the Riemann shock problem, where discontinuities such as shocks create difficulties for standard numerical methods to solve for the convective flux of the conservation equations. The use of central schemes with artificial dissipation may have numerically smoothed out these flow features and is another reason why an upwind scheme was chosen.

4.1.2 Diffusive Flux

The diffusive flux describes flux of ϕ entering the control volume Ω due to the diffusion of ϕ. The diffusive flux is proportional to the gradient of ϕ, and is described as

$$\oint_{\partial\Omega} \kappa\rho \left(\nabla\phi \cdot \hat{n}\right) dS \tag{4.2}$$

where κ is a diffusivity coefficient, ρ is the density and $\nabla\phi$ is the gradient of ϕ. Since the gradients are stored at the nodes, a modified central scheme is used to calculate the edge-based gradients. An ordinary central scheme provides uncoupling between the local terms and edge-based gradients. The following modification has been added to determine the edge-based gradients

$$\nabla\phi_{ij} = \frac{1}{2}\left(\nabla\phi_i + \nabla\phi_j\right) - \left[\frac{1}{2}\left(\nabla\phi_i + \nabla\phi_j\right) \cdot \hat{e}_{ij}\right] \hat{e}_{ij} + \frac{\phi_j - \phi_i}{\vec{x}_j - \vec{x}_i}\hat{e}_{ij}. \tag{4.3}$$

To calculate the diffusive flux and for second-order spatial discretization, the gradients at the nodes must be calculated. Since UNS3D is an unstructured and cell-vertex/dual-mesh flow solver, there are specific numerical methods that can be used to calculate the gradients. The following gradient methods are available in UNS3D: Green-Gauss (Kim, 2003), least-squares (Kim, 2003), least squares with QR decomposition (Gargoloff, 2007) and the Weighted Essentially Non-Oscillatory method. This research utilizes the Least Squares with QR decomposition as the primary gradient calculation method.

4.1.3 Second-Order Spatial Discretization

So far, this chapter has described the spatial discretization used for first-order spatial accuracy. First-order schemes tend to have higher dissipation and is considered not accurate enough for the standards of today's flow solvers. It is now the standard to simulate flows using second-order spatial accuracy. For the finite-volume method, second-order spatial accuracy is achieved by assuming the state variables vary linearly across the control volume. The state variables and their gradients which are located at the nodes are required to define a piecewise, linearly-varying approximation for a continuous flow field (Barth and Jesperson, 1989). To achieve a linear variation, the nodal values across edge "ij" are reconstructed and defined as "left" and "right" state variables.

$$\vec{Q}_L = \vec{Q}_i + \frac{1}{2}\Psi_i \left(\nabla\vec{Q}_i \cdot (\vec{x}_j - \vec{x}_i)\right)$$
$$\vec{Q}_R = \vec{Q}_j - \frac{1}{2}\Psi_j \left(\nabla\vec{Q}_j \cdot (\vec{x}_j - \vec{x}_i)\right)$$

For second-order accuracy, both the convective and diffusive fluxes utilize the reconstruction presented above. The value of Ψ is a limiter function and is determined on which solution limiter is used. The value of Ψ ranges from 0 to 1, so that the limiter may revert the reconstruct solution to being first-order accurate if non-physical phenomena occurs. Solution limiters are functions that prevent non-physical oscillations and spurious solutions in the vicinity of large gradients, such as shocks, when the second-order linear reconstruction is used. The solution limiters accomplish this by enforcing a monotonicity preserving scheme. The primary solution limiter used in this work is the Venkatakrishnan (1995) and the Dervieux (1985) limiters.

4.2 Temporal Discretization

The Navier-Stokes equations can be decomposed into two parts: the spatial derivatives and the temporal derivatives. The spatial discretization section that was just discussed describes how the spatial derivatives of the Navier-Stokes equations are solved. The spatial derivatives of the Navier-Stokes equations are composed of the source terms along with the convective and diffusive flux terms. These terms are grouped together to form the residual, thus the Navier-Stokes equations can be described as

$$\frac{\partial}{\partial t}\left(\vec{Q}_i, \Omega_i\right) = \vec{R}_i \tag{4.4}$$

where $\vec{Q}_i$ is the state vector, Ω_i is the control volume and $\vec{R}_i$ is the residual located at node i.

Temporal discretization is used accurately integrate this ordinary differential equation in time. The two common methods to integrate through time for flow solvers are the explicit and implicit time integration. Implicit time integration is a method that is inherently stable and can utilize large time steps but has large memory requirements and is computationally expensive per time step. Implicit time integration is also complicated to implement. Explicit time-integration is easy to implement, is computationally inexpensive and has minimal memory requirements for each time step. The drawbacks to using explicit time-integration is that the scheme can be numerically unstable depending on the size of the maximum time step. For stability, the size of the maximum time step is limited by the Courant-Friedrichs-Lewy number, commonly known as the CFL number. The CFL condition states that the domain of dependence of the numerical scheme must include the domain of dependence of the governing equations. The CFL number relates the speed at which information is propagating throughout the discretized domain. Explicit time-integration is the primary time-integration method for UNS3D, and is solely used in this research due to it's simplicity. Explicit time-integration utilizes a first-order, forward finite difference approximation for the time derivative. Second-order accuracy may be achieved by using a four stage Runge-Kutta method. UNS3D utilizes the four stage Runge-Kutta integration and optimizes its explicit time-integration by using convergence acceleration techniques, implicit residual smoothing and local time-stepping. For unsteady simulations, UNS3D utilizes dual-time stepping. A detailed description of the temporal and spatial discretization techniques of UNS3D may be found in the works by Kim (2003), Gargoloff (2007) and Liliedahl (2017).

4.3 Boundary Conditions

The implementation of boundary conditions can greatly affect the accuracy and convergence of the final solution of the flow solver. The rest of this section describes the implementation of the various boundary conditions in both the full order and time-linearized versions of UNS3D. The boundary conditions in the UNS3D family of flow solvers have weak implementations. Weak implementation of boundary conditions refer to the boundary conditions not being directly applied to the nodal values, but to boundary nodes created by the dual mesh. These boundary nodes are used in determining the fluxes from the boundary, and it is in these fluxes that the boundary conditions are satisfied.

4.3.1 Full-Order Model Boundary Conditions

The first boundary conditions for discussion are the inlet and outlet boundary conditions. The inlet and outlet conditions are the locations in which the computational domain is artificially truncated, thus it is ideal that the inlet and outlet boundary conditions replicate the effect of having an infinite domain. This is accomplished by the use of the local characteristics of the flow. These characteristics determine the direction in which information is propagating throughout the flow field. The local characteristics are obtained by calculating the eigenvalues of the local flux Jacobian matrix. Positive eigenvalues indicate that information is going downstream while a negative eigenvalue indicates that information is propagating upstream. For supersonic flows, all information propagates downstream, as the flow velocity has reached or exceeded the speed of sound. Subsonic inlets and outlets both have four positive and one negative characteristics. This means that subsonic inlets can impose four conditions from the freestream or farfield flow, and one condition from the interior of the computational domain. For subsonic outlets, 4 conditions must be imposed from the interior of the domain, and one condition can be imposed from downstream of the computational domain. For the subsonic inlet, the four conditions imposed from the freestream are the two angles of attack, α and β, the total pressure, and the total temperature. The condition imposed from the interior of the domain is the outgoing Riemann invariant. For subsonic outlets, only one condition is imposed from downstream of the domain, which is a user-specified static pressure. The conditions enforced from the interior of the domain is the entropy, tangential velocity, and the incoming and outgoing Riemann invariants.

The wall boundaries are composed of inviscid and viscous wall boundaries. All wall boundaries enforce the no penetration condition, which means that the velocity normal to surface of the wall is zero. This is the only condition for inviscid walls. For viscous flows, the boundary conditions consist of the no penetration and the no slip condition. The no slip condition states that the velocity of the fluid is equal to the velocity of the wall surface. For viscous flows with stationary walls, the fluid velocity at the wall is zero.

Symmetry boundary conditions are used to truncate the computational domain whenever there is a plane of symmetry. Symmetry boundary conditions are also used to simulate a 2D flow field with a 3D flow solver, creating what is known as a quasi-3D simulation. There are two conditions that must be satisfied with symmetry boundaries: no-penetration and the zero-gradient state variables on the planes of symmetry. For this research, a strong implementation of the zero-gradient condition was enforced, that is both the boundary state vectors and state vectors on the symmetry boundary had the zero gradient condition applied.

4.3.2 Shear Stress Transport Boundary Conditions

The Shear Stress Transport turbulence model primarily utilizes Neumann boundary conditions. The boundary conditions used are the recommended values from the original reference for the Shear Stress Transport turbulence model (Menter, 1994). The boundary conditions are listed in Table 4.1.

Boundary	κ	ω
Wall	$\kappa = 0$	$\omega = 10\frac{6\nu}{\beta_1(\delta d_1)^2}$
Freestream	$\frac{10^{-5}U_\infty^2}{Re_L} < \kappa < \frac{0.1U_\infty^2}{Re_L}$	$\frac{U_\infty}{L} < \omega < \frac{10U_\infty}{L}$
Inlet	$\frac{10^{-5}U_\infty^2}{Re_L} < \kappa < \frac{0.1U_\infty^2}{Re_L}$	$\frac{U_\infty}{L} < \omega < \frac{10U_\infty}{L}$
Symmetry	$\frac{\partial\kappa}{\partial n} = 0.0$	$\frac{\partial\omega}{\partial n} = 0.0$

Table 4.1: Shear Stress Transport Boundary Conditions

The symmetry boundary condition for the Shear Stress Transport turbulence model follows the same structure as the governing equations, where the zero-gradient normal is applied for κ and ω. For the freestream and inlet boundaries, Dirichlet boundaries are enforced for a specific range. It is important to note that the L represents the length of the computational domain, not the reference length of an object of study. One must also be careful when trying to obtain the freestream values of

ω from turbulent intensities, as there are many ways to define the equation for turbulence intensity Blazek (2001); Menter (1994). For the outlet, the value of κ and ω is simply interpolated. Periodic boundaries are applied to κ and ω in the same manner as the state vectors in the full-order model.

4.3.3 Spalart-Allmaras Boundary Conditions

The Spalart-Allmaras turbulence model utilizes both Dirichlet and Neumann for its working variable $\tilde{\nu}$. Dirichlet boundary conditions are applied to the inlet/freestream and wall boundaries, while Neumann boundary conditions are applied for symmetry boundaries. Table 4.2 lists the boundary conditions for the Spalart-Allmaras turbulence model.

Boundary	Dirichlet	Neumann
Wall	-	$\tilde{\nu} = 0.0$
Freestream	-	$\frac{\tilde{\nu}}{\nu_L} = 3 - 5$ (Fully Turbulent)
Inlet	-	$\frac{\tilde{\nu}}{\nu_L} = 3 - 5$ (Fully Turbulent)
Symmetry	$\frac{\partial \tilde{\nu}}{\partial n}$ = 0.0	-

Table 4.2: Spalart Allmaras Boundary Conditions

The symmetry boundary condition for the Spalart-Allmaras turbulence model follows the same structure as the governing equations, where the zero-gradient normal is applied for $\tilde{\nu}$. For the freestream and inlet boundaries, a $\frac{\tilde{\nu}}{\nu_L}$ value between 3 and 5 is used to simulate fully turbulent flows. If tripping the Spalart-Allmaras turbulence model, this value would be less than 1. For the outlet, the value of $\tilde{\nu}$ is simply interpolated. Periodic boundaries are applied to $\tilde{\nu}$ in the same manner as the state vectors in the full-order model.

4.3.4 Time-Linearized Boundary Conditions

The time-linearized flow solver utilizes many of the same boundary conditions as the full order solver. Dirichlet, Neumann and zero-gradient boundary conditions are applied to the complex, first-order state vector as well as the real zeroth-order steady-state vector. The time-linerized flow solver does contain a few unique boundary conditions. The first set of boundary conditions unique to the time-linearized flow solver are the analytically non-reflecting inlet and outlet boundary conditions. These boundary conditions follow the same analysis as the full-order inlet and boundary conditions, where the characteristics of the flow are analyzed to observe the direction that information is travelling. The non-reflective boundary conditions are designed to cancel information from being reflected from the boundary into the interior of the domain. A detailed description of the method can be found in Liliedahl (2017).

The second set of unique boundary conditions are the complex, rotational periodic boundary conditions for circular whirling orbits. These boundary conditions is what allows the truncation of the computational domain for rotordynamic analysis of seals. These boundary conditions also give an insight in how the forces are obtained for rotordynamic analysis of seals. Rotational periodic boundaries for time-accurate, full-order model work by averaging the master and slave nodes for the values of pressure and density. For the velocity components, an average of the master and slave node velocity components are performed after the slave nodes velocity components are rotated by a rotation matrix to match the same unit normal of the master periodic boundary face. After the rotated slave velocities are averaged with the master velocities, the slave velocities are then rotated back into the initial unit normal of the slave periodic boundaries.

The time-linearized rotational periodic boundaries for circular whirling orbits follow the same principle as the full order model, except with an additional step. Similar to how the velocity components must be rotated in order to properly average the values, a phase shift must be applied in order to properly average all values in the state vector. This is due to the first-order perturbations being complex, where each value of the first-order state vector can be thought of as a vector in the complex plane. A description of the complex rotational periodic boundary conditions is as follows:

Step 1: Phase shift slave terms to master phase

$$\rho_{1s} = \rho_s \left(\cos(\theta) + j\sin(\theta)\right)$$

$$u_{1s} = u_s \left(\cos(\theta) + j\sin(\theta)\right)$$

$$v_{1s} = v_s \left(\cos(\theta) + j\sin(\theta)\right)$$

$$w_{1s} = w_s \left(\cos(\theta) + j\sin(\theta)\right)$$

$$p_{1s} = p_s \left(\cos(\theta) + j\sin(\theta)\right)$$

Step 2: Enforce average values on master face

$$\rho_{1m} = 0.5 \left(\rho_{1m} + \rho_{1s}\right)$$

$$u_{1m} = 0.5 \left(u_{1m} + u_{1s}\right)$$

$$v_{1m} = 0.5 \left(v_{1m} + v_{1s}\cos(\theta) - w_{1s}\sin(\theta)\right)$$

$$w_{1m} = 0.5 \left(v_{1m} + w_{1s}\cos(\theta) + v_{1s}\sin(\theta)\right)$$

$$p_{1m} = 0.5 \left(p_{1m} + p_{1s}\right)$$

Step 3: Phase shift average values to slave phase and enforce

$$\rho_{1s} = \rho_{1m} \left(\cos(\theta) - j\sin(\theta)\right)$$

$$u_{1s} = u_{1m} \left(\cos(\theta) - j\sin(\theta)\right)$$

$$v_{1s} = \left(v_{1m}\cos(\theta) + w_{1m}\sin(\theta)\right)\left(\cos(\theta) - j\sin(\theta)\right)$$

$$w_{1s} = \left(w_{1m}\cos(\theta) - v_{1m}\sin(\theta)\right)\left(\cos(\theta) - j\sin(\theta)\right)$$

$$p_{1s} = p_{1m} \left(\cos(\theta) - j\sin(\theta)\right)$$

where the value θ is the physical angle of rotation between the master and slave rotational periodic boundaries. For an axisymmetric domain with a circular whirling orbit, the phase angle between the master and slave boundaries is equivalent to the physical angle θ. The phase shifts enforces the unsteady solution to be axisymmetrically periodic. This phase shift only applies when an axisymmetric domain has a circular whirling orbit.

4.4 Rotordynamic Analysis

Rotordynamic stability coefficients are obtained through the equations of motion for a whirling rotor. These coefficients are comprised of the direct and cross-coupled inertial, damping and stiffness coefficients. For gas seals, it is common to neglect the inertial terms. The system of equations representing the equations of motion for a gas seal are shown below:

$$-\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} K & k \\ -k & K \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} C & c \\ -c & C \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} \tag{4.5}$$

where X and Y represent the displacement of the rotor, and $\dot{X}$ and $\dot{Y}$ are the velocity components of the rotor. For a circular whirling orbit, the displacements and velocities are defined as follows:

$$X = Re(\epsilon e^{j\omega t}) \qquad \dot{X} = Re(j\omega\epsilon e^{j\omega t}) \tag{4.6}$$

$$Y = Re(-j\epsilon e^{j\omega t}) \qquad \dot{Y} = Re(\omega\epsilon e^{j\omega t}) \tag{4.7}$$

where ϵ is the radius of the whirling orbit. One then analyzes the forces on the rotor when the rotor is displaced only in the x-direction, or when $t = 0$. Substituting these definitions at $t = 0$ into the system of equations 4.5 then gives the following equations.

$$-\frac{F_x}{\epsilon} = K + c\omega \tag{4.8}$$

$$-\frac{F_y}{\epsilon} = -k + C\omega \tag{4.9}$$

In order to get the forces in the x and y direction, we must integrate the pressure along the surface of the shaft.

$$
\begin{aligned}
F_x &= \int_0^L \int_0^{2\pi} Real\left[(p_0 + \epsilon\tilde{p}_1 e^{j\theta})\right] \cdot R\cos(\theta) d\theta dz \\
F_y &= \int_0^L \int_0^{2\pi} Real\left[(p_0 + \epsilon\tilde{p}_1 e^{j\theta})\right] \cdot R\sin(\theta) d\theta dz
\end{aligned}
$$

$$
\begin{aligned}
F_x &= \int_0^L \int_0^{2\pi} Real\left[(p_0 + \epsilon\left(\tilde{p}_{1r}\cos(\theta) - \tilde{p}_{1i}\sin(\theta)\right)\right] \cdot R\cos(\theta) d\theta dz \\
F_y &= \int_0^L \int_0^{2\pi} Real\left[(p_0 + \epsilon\left(\tilde{p}_{1r}\cos(\theta) - \tilde{p}_{1i}\sin(\theta)\right)\right] \cdot R\sin(\theta) d\theta dz
\end{aligned}
$$

Figure 4.1 shows the integration of pressure of a whirling rotor in cylindrical coordinates. The $R\cos(\theta)d\theta dz$ term represents the x-component of area while the $R\sin(\theta)d\theta dz$ term represents the y-component of area. R is the radius of the rotor. There is a new term $\tilde{p}_1$ which represents the value of $(p_{1r} + jp_{1i})\,e^{j\theta}$ at $\theta = 0$. With an axisymmetric domain and a circular whirling orbit, $\tilde{p}_1$ can be used to obtain the values of the complex pressure in the circumferential direction through the use of phase shifts. This is the same concept that was introduced in Section 4.3.4, which was the complex rotational periodic boundaries for axisymmetric domains with circular whirling orbits. When integrating over θ, the only terms left are the squared trigonometric terms. The force is obtained by:

$$
\frac{F_x}{\epsilon} = R\pi \int_0^L \tilde{p}_{1r} dz \tag{4.10}
$$

$$
\frac{F_y}{\epsilon} = -R\pi \int_0^L \tilde{p}_{1i} dz \tag{4.11}
$$

In order to obtain the rotordynamic coefficients, there must be a minimum of two time-linearized simulations at different whirling frequencies. When the forces for each frequency are obtained, one can create a system of equations using Equations 4.9 and solve for the rotordynamic coefficients.

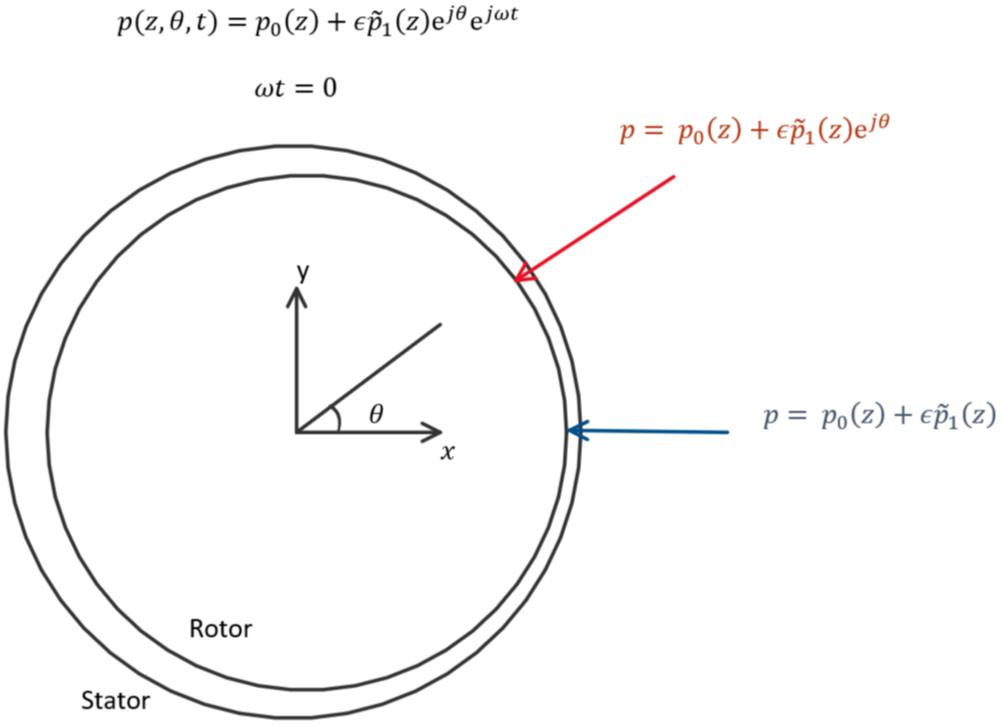

Figure 4.1: Complex Pressures for Circular Whirling Orbit with an Axisymmetric Domain.

Unfortunately, the method described of obtaining the forces for rotordynamic analysis only applies to smooth annular seals.

4.5 Time-Linearized Force Integration

The time-linearized flow solver lacked a generalized method to obtain the amplitudes of the force components. A generalized method would allow the time-linearized solver to obtain forces acting on complex geometries such as an airfoil, and allows the time-linearized flow solver to perform rotordynamic analysis on any seal geometry. The method implemented to obtain the forces was modeled after the full-order flow solver UNS3D, which simply involved integrating the pressure and shear stress on the wall boundaries. The new force integrator has the ability to distinguish which walls to calculate the forces on, and to reconstruct the forces acting on the full annulus when simulating a slice of the domain. The reconstruction is accomplished by applying a rotation to the wall surface areas for each cell, and a phase shift for the state vector to match the rotation.

5. RESULTS

5.1 Verification and Validation of UNS3D Turbulence Models

The Spalart-Allmaras turbulence must be verified and validated for the full-order flow solver before implementing a time-linearized flow solver into the time-linearized code. The cases used for verification and validation of the Spalart-Allmaras turbulence model come from NASA's turbulence modeling resource website. Both the parallel and sequential versions of UNS3D had the Spalart-Allmaras turbulence model implemented and underwent the same tests. The Shear-Stress Transport turbulence model was also verified and validated using the same test cases. The results from UNS3D were compared against experimental and numerical results from NASA's FUN3D and CFL3D. The results from the sequential version of UNS3D used a coarser grid than the parallel UNS3D and the NASA codes.

5.1.1 Turbulent Flat Plate

The first case used to verify and validate the turbulence models is the canonical turbulent flat plate case. The turbulent flat plate case allows one to validate a turbulence model against Cole's Law of the Wake Coles (1956), where the velocity profile and distance from the wall has been non-dimensionalized to highlight the different regions of the turbulent boundary layer such as the log-region sublayer. For verification, both the sequential and parallel versions of UNS3D will be compared against results from NASA's FUN3D and CFL3D published on the Turbulence Modeling Resource Website (NASA, 2019). These results include dimensionless turbulent viscosity μ_T, κ, and ω contours.

A description of the computational mesh and the boundary conditions for this case may be found in figure 5.1. Two levels of mesh refinement were used for this case, where the parallel code used the finest grid and the sequential code used the 3rd finest grid. Table 5.1 describes freestream boundary conditions for the various turbulence models and codes. The freestream turbulence boundary conditions that NASA's FUN3D SST and CFL3D SST used are above the

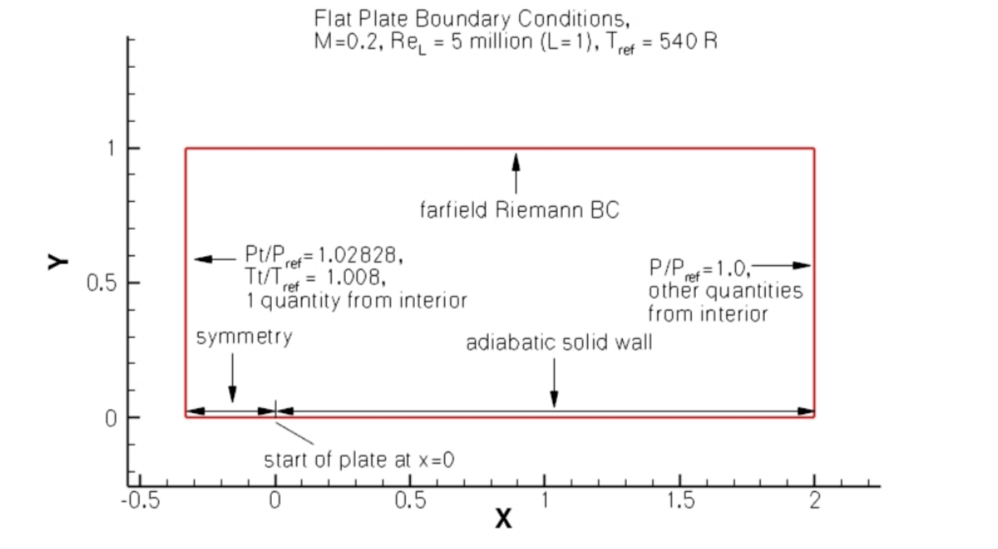

Figure 5.1: Flat Plate Boundary Conditions. Figure and Meshes obtained from NASA (2019).

Code	Mesh	$\frac{\tilde{\nu}_\infty}{\nu_L}$	$\omega_\infty \frac{\nu_L}{c_\infty^2}$	$\kappa_\infty \frac{1}{c_\infty^2}$
UNS3D SA	2 x 137 x 97	3.0	-	-
POD UNS3D SA	2 x 545 x 385	3.0	-	-
FUN3D SA	2 x 545 x 385	3.0	-	-
UNS3D SST	2 x 137 x 97	-	1 x 10^-6	9 x 10^-9
POD UNS3D SST	2 x 545 x 385	-	4 x 10^-8	4 x 10^-10
FUNS3D SST	2 x 545 x 385	-	1 x 10^-6	9 x 10^-9

Table 5.1: Turbulent Boundary Conditions for Flat Plate

recommended range of what Menter proposed Menter (1994). The recommended range for SST is

$$\frac{U_\infty}{L} < \omega_\infty < 10\frac{U_\infty}{L} \qquad\qquad 10^{-5}\frac{U_\infty^2}{Re_L} < \kappa_\infty < 0.1\frac{U_\infty^2}{Re_L}. \tag{5.1}$$

The values used by NASA FUN3D correspond to

$$\omega_\infty = 125\frac{U_\infty}{L} \qquad\qquad \kappa_\infty = 1.125\frac{U_\infty^2}{Re_L} \tag{5.2}$$

As shown, FUN3D uses turbulent freestream conditions that are about 2 orders of magnitude higher than the recommended range. POD UNS3D SST uses values of κ and ω which lie within the recommended range.

$$\omega_\infty = 5\frac{U_\infty}{L} \qquad \kappa_\infty = 0.05\frac{U_\infty^2}{Re_L} \tag{5.3}$$

POD UNS3D SST did not use the same freestream boundary conditions as the NASA codes, as POD UNS3D SST is sensitive to freestream values beyond the recommended range proposed by Menter. When POD UNS3D SST used the turbulence boundary conditions of the NASA codes, POD UNS3D SST produced κ and ω contours that were about 10 percent less in magnitude than the NASA results. The results presented for POD UNS3D SST are for the turbulent boundary conditions in the recommended range as stated in Table 5.1.

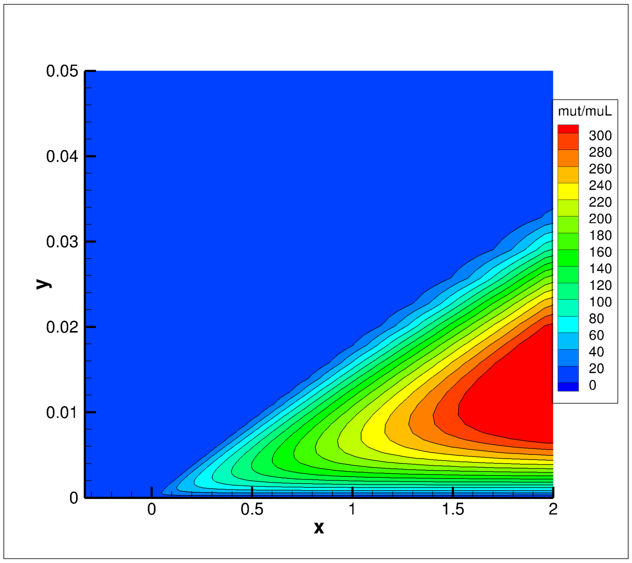

Figure 5.2: Flat Plate: SEQ UNS3D SA μ_t/μ Contours.

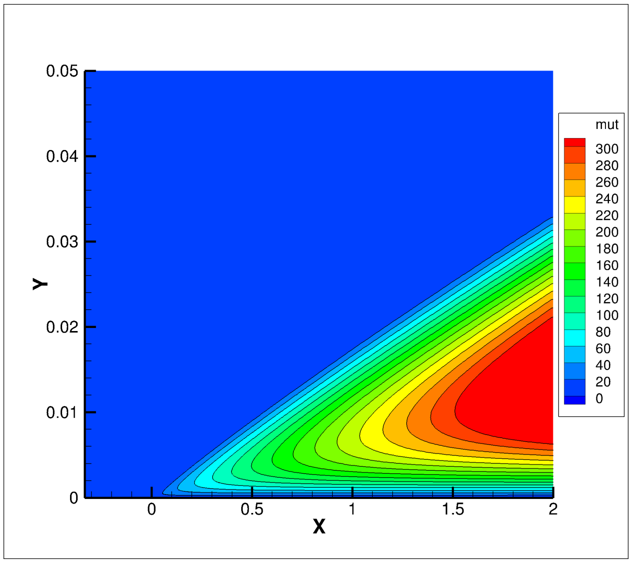

Figure 5.3: Flat Plate: FUN3D SA μ_t/μ Contours. Figure compiled using data from NASA (2019).

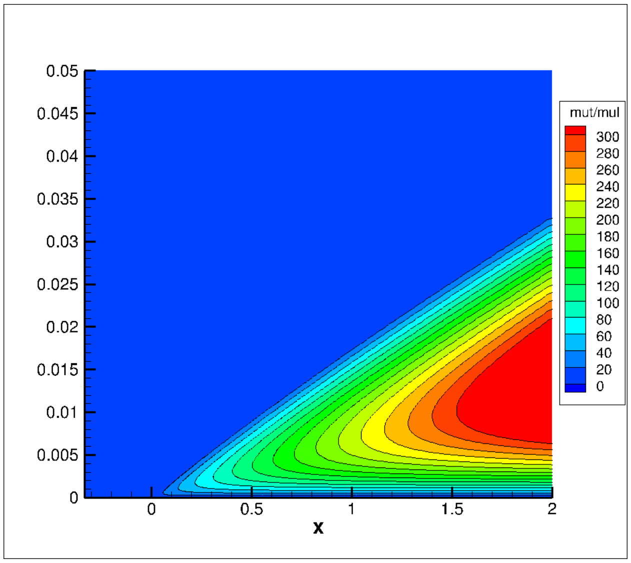

Figure 5.4: Flat Plate: POD UNS3D SA μ_t/μ Contours.

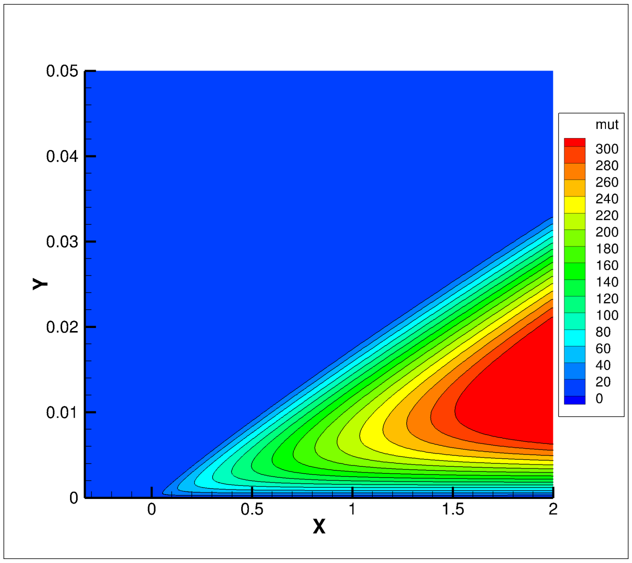

Figure 5.5: Flat Plate: FUN3D SA μ_t/μ Contours. Figure compiled using data from NASA (2019).

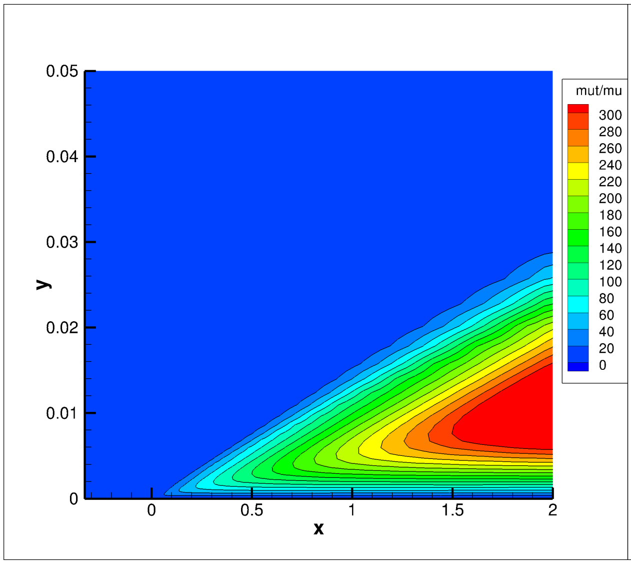

Figure 5.6: Flat Plate: SEQ UNS3D SST μ_t/μ Contours.

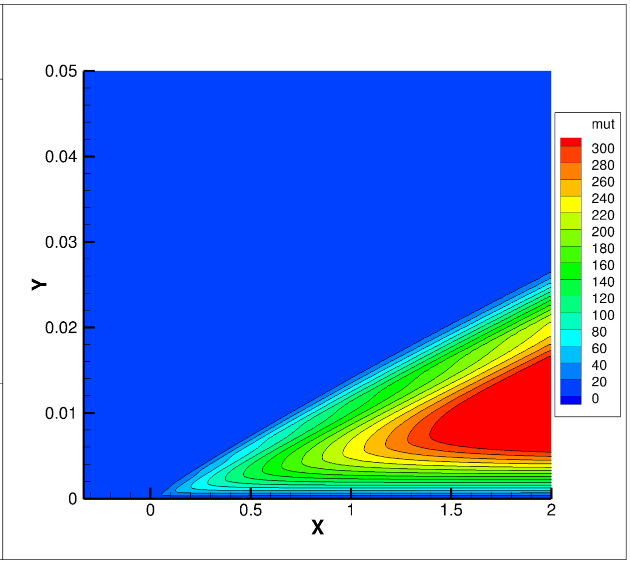

Figure 5.7: Flat Plate: FUN3D SST μ_t/μ Contours. Figure compiled using data from NASA (2019).

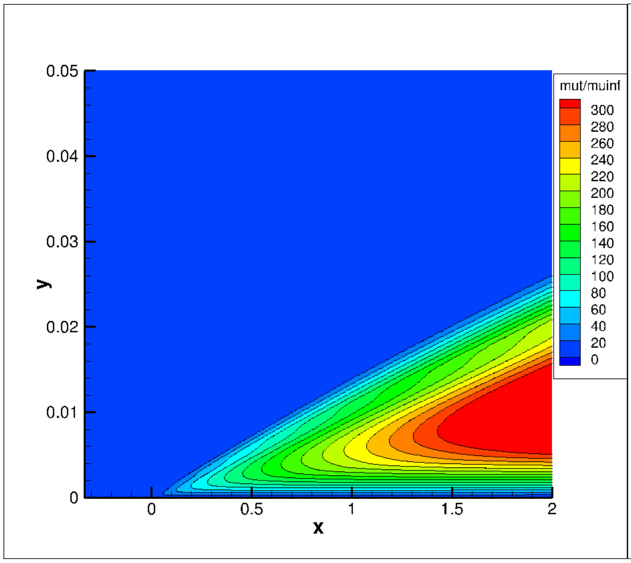

Figure 5.8: Flat Plate: POD UNS3D SST μ_t/μ Contours.

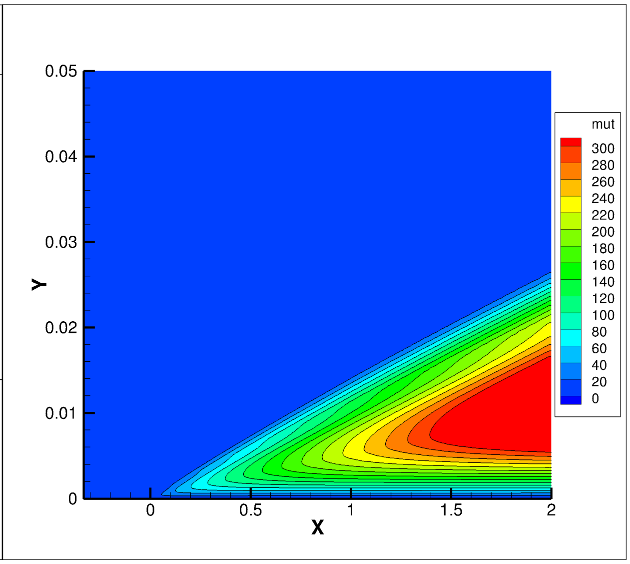

Figure 5.9: Flat Plate: FUN3D SST μ_t/μ Contours. Figure compiled using data from NASA (2019).

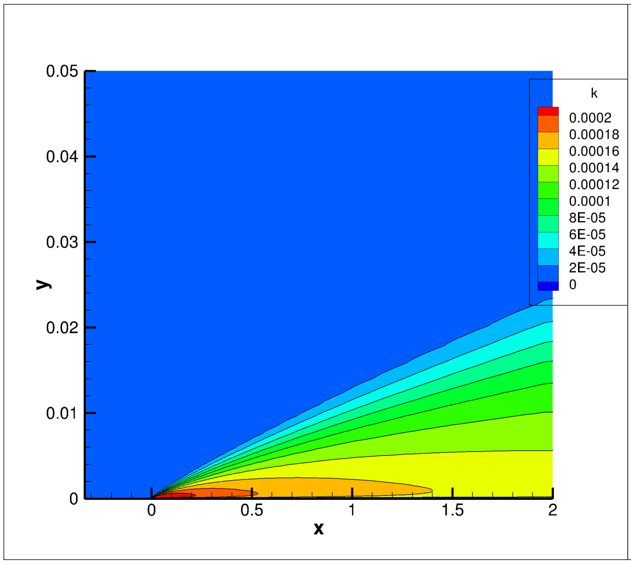

Figure 5.10: Flat Plate: SEQ UNS3D SST k Contours.

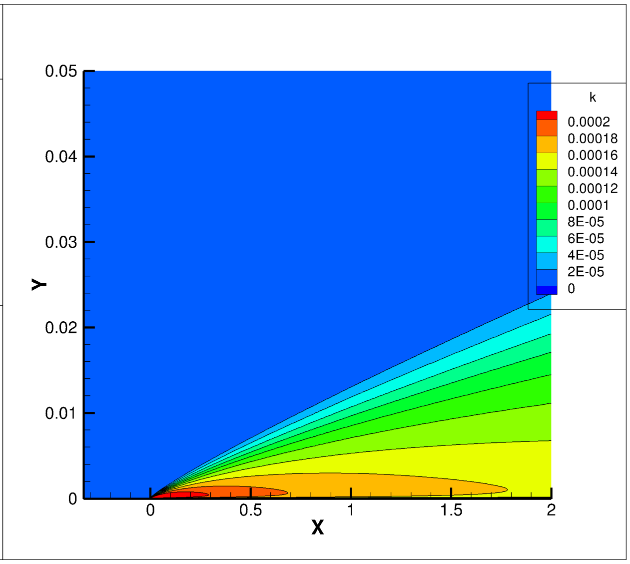

Figure 5.11: Flat Plate: FUN3D SST k Contours. Figure compiled using data from NASA (2019).

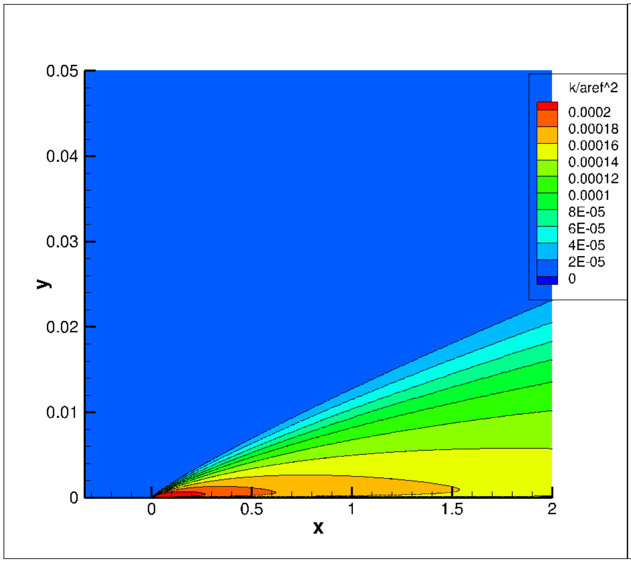

Figure 5.12: Flat Plate: POD UNS3D SST k Contours.

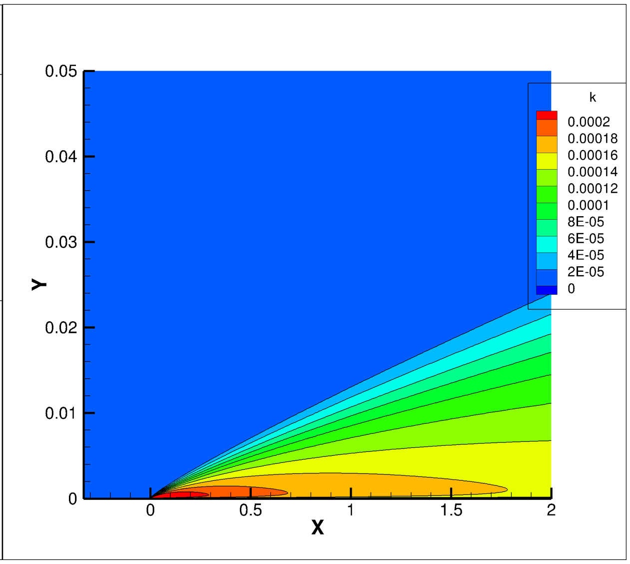

Figure 5.13: Flat Plate: FUN3D SST k Contours. Figure compiled using data from NASA (2019).

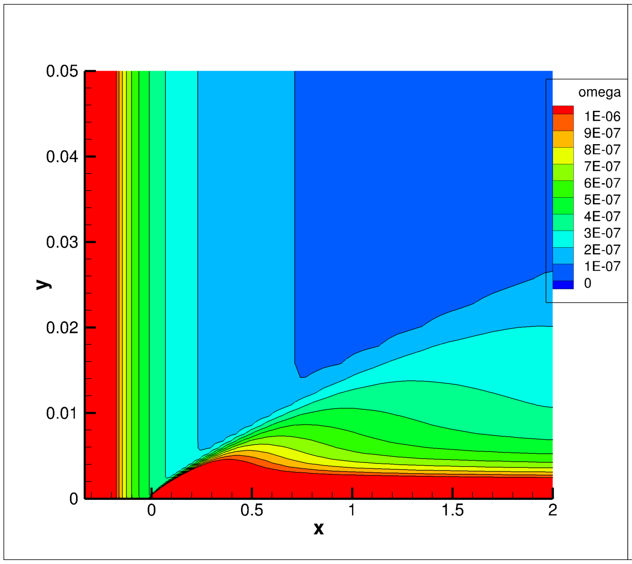

Figure 5.14: Flat Plate: SEQ UNS3D SST w Contours.

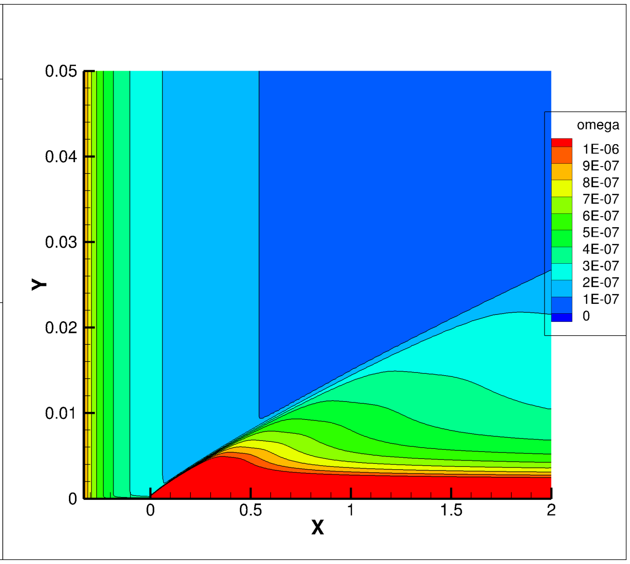

Figure 5.15: Flat Plate: FUN3D SST w Contours. Figure compiled using data from NASA (2019).

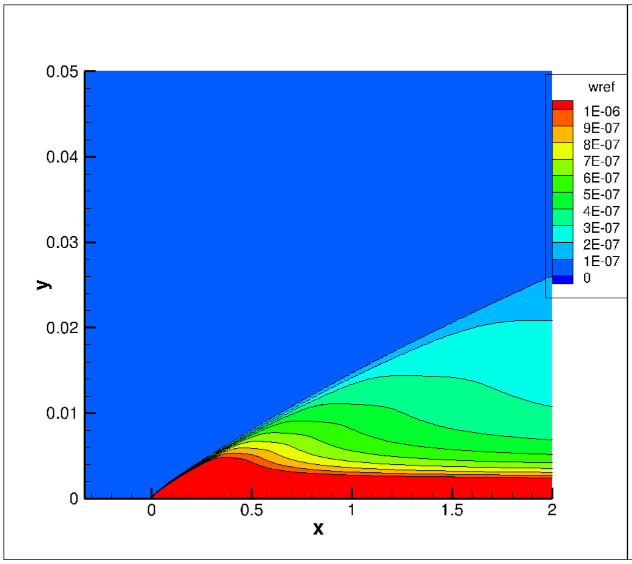

Figure 5.16: Flat Plate: POD UNS3D SST w Contours.

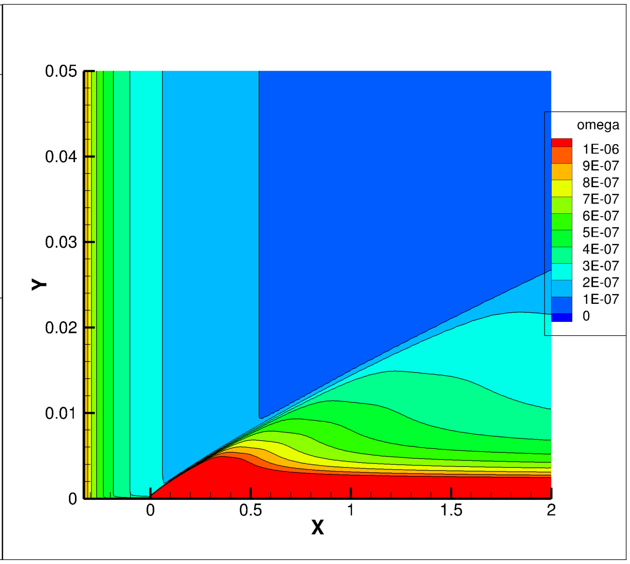

Figure 5.17: Flat Plate: FUN3D SST w Contours. Figure compiled using data from NASA (2019).

The primary differences between the POD UNS3D SST ω contours and the FUN3D SST ω

contours is the region near the inlet of the domain. FUN3D ω contours start at high values and dissipate as the flow goes out of the domain. This is to be expected as the turbulent boundary conditions for FUN3D are at least an order of 1 higher than the turbulent boundary conditions for POD UNS3D SST. Overall UNS3D is in good agreement with FUN3D in the region near the wall.

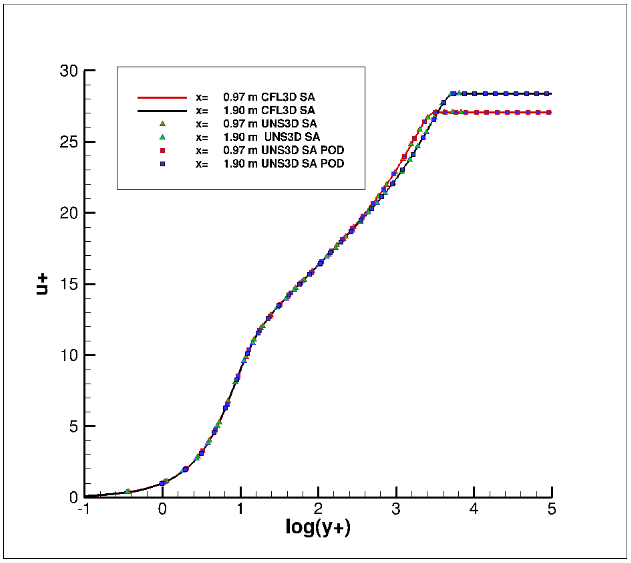

Figure 5.18: SA: Law of the Wake. Figure used data from NASA (2019).

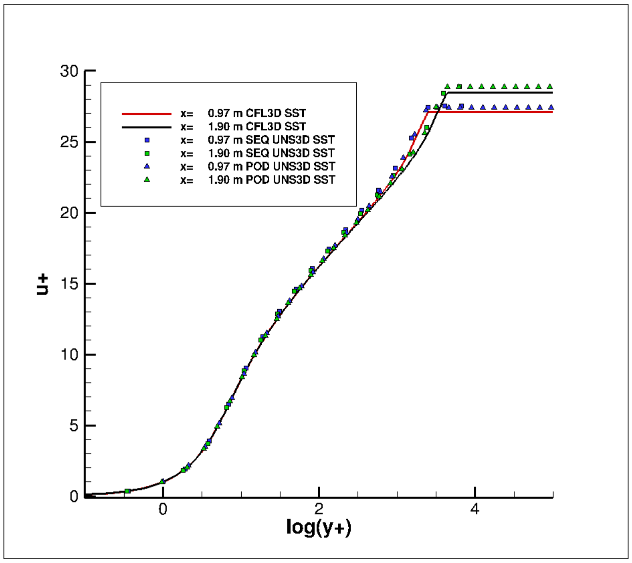

Figure 5.19: SST: Law of the Wake. Figure used data from NASA (2019).

Overall the turbulence models in both of the UNS3D codes are in good agreement with both theory and the results from NASA's CFD codes. The Spalart-Allmaras results from UNS3D seem to match the Spalart-Allamras results from the NASA codes, FUN3D and CFL3D, better than UNS3D's SST codes. The implementation of FUN3D and UNS3D's SST turbulence models are very different, as shown by the different boundary conditions required. The non-dimensionalization of UNS3D's SST and the NASA codes' SST is also different, where the order of magnitude between κ and ω is closer in the NASA code than UNS3D. In UNS3D, the κ and ω freestream conditions are about 12 orders of magnitude different, which could lead to some numerical errors.

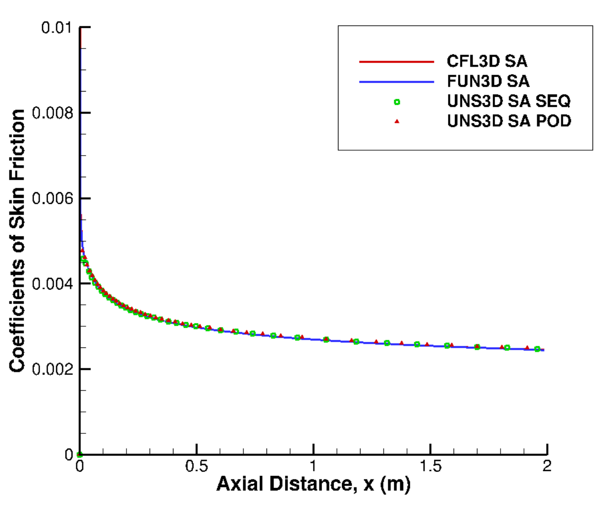

Figure 5.20: Turbulent Flat Plate: SA Skin Friction Coefficients. Figure used data from NASA (2019).

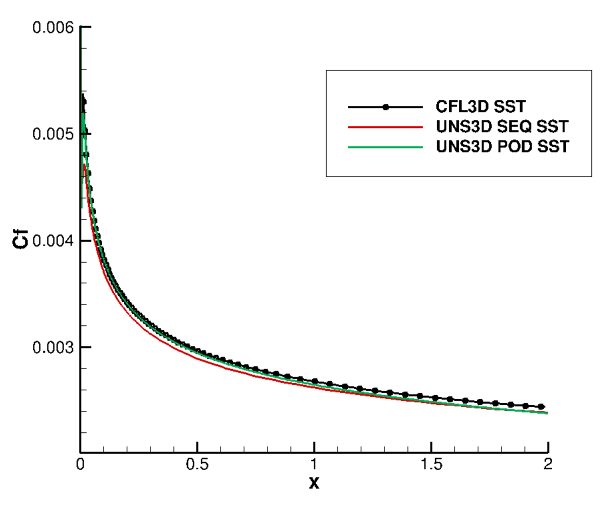

Figure 5.21: Turbulent Flat Plate: SST Skin Friction Coefficients. Figure used data from NASA (2019).

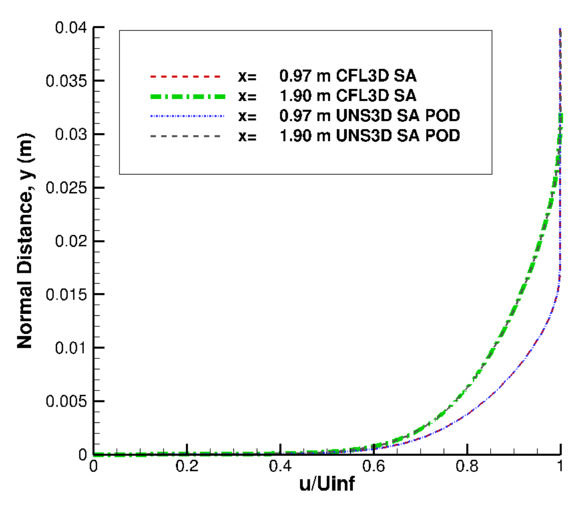

Figure 5.22: Turbulent Flat Plate: SA Velocity Profiles. Figure used data from NASA (2019).

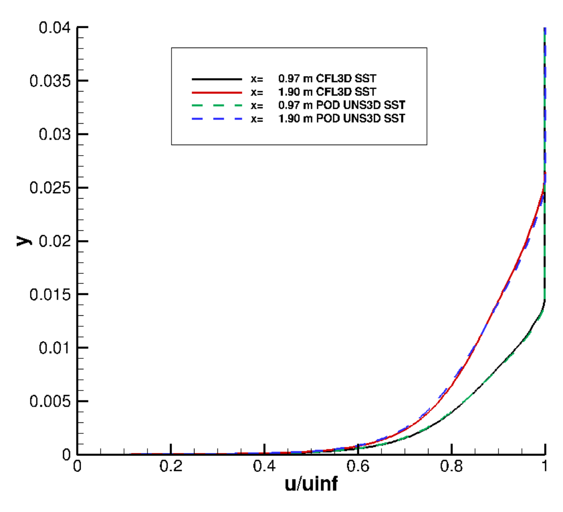

Figure 5.23: Turbulent Flat Plate: SST Velocity Profiles. Figure used data from NASA (2019).

5.1.2 2D Airfoil Near-wake

This test case is based off of an experiment performed by Nakayama (1985). The conventional airfoil from Nakayama (1985) was simulated at a low Mach number and zero angle of attack. This case allows the turbulence model to be tested on flows with pressure gradients, such as impellers and blade cascades. Since the sequential code is being tested, the grid used was the mesh where the inlet and outlet were located 20 chords away instead of the 500 chords, in an effort to reduce computation time required for convergence. According to the Turbulence Modeling Resource, the main difference between the two grids with dissimilar boundaries were the results for the lift coefficient, C_L, depending on the way the farfield boundary conditions were implemented. The input parameters are listed in table 5.2.

P_{ref} [Pa]	T_{ref} [K]	M_{inlet} [-]	Re [-]	μ_{ref} [Pa s]	$\frac{\tilde{\nu}}{\nu}$ [-]	$\omega_\infty \frac{\nu_L}{c_\infty^2}$ [-]	$\kappa_\infty \frac{1}{c_\infty^2}$ [-]
62,000	300	0.088	1,200,000	1.838e-5	3	4 x 10^-8	4 x 10^-10

Table 5.2: Input parameters for Nakayama airfoil.

The boundary conditions for the Nakayama test case stated by the Turbulence modeling resource page is shown in Fig. 5.24. Modifications were made to the boundary conditions to avoid issues of having the inlet right next to the outlet. The boundary conditions enforced for the test case were: Riemann inlet on the C-curve, Riemann outlet on the right side, and freestream boundary conditions were enforced on the top and bottom horizontals. As in the flat plate case, the same procedures were followed to make the 3D flow solver simulate a 2D case.

As with the turbulent flat plate case, the freestream values of κ and ω for POD UNS3D SST are different than the values used for the FUN3D simulation.

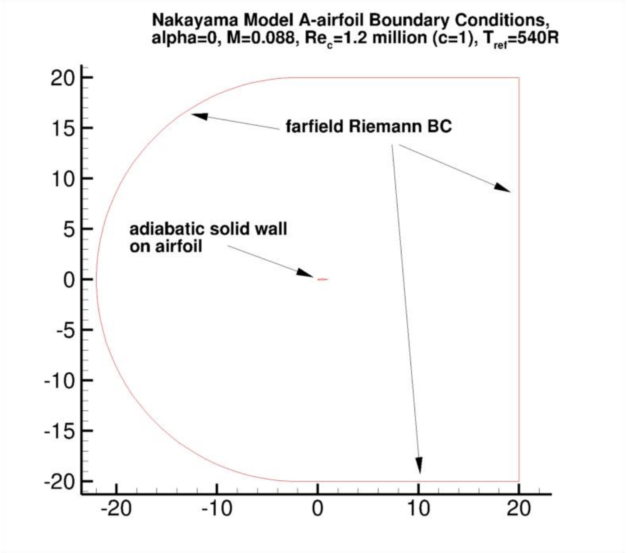

Figure 5.24: Nakayama boundary conditions. Figure from NASA (2019).

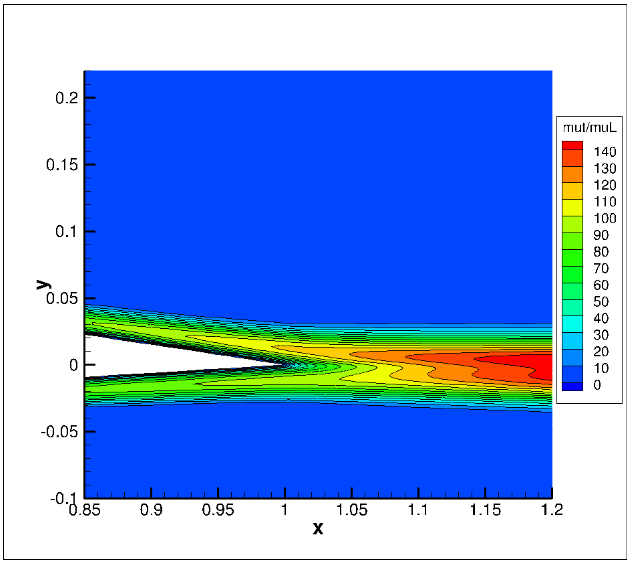

Figure 5.25: Nakayama: SEQ UNS3D SA μ_t/μ Contours.

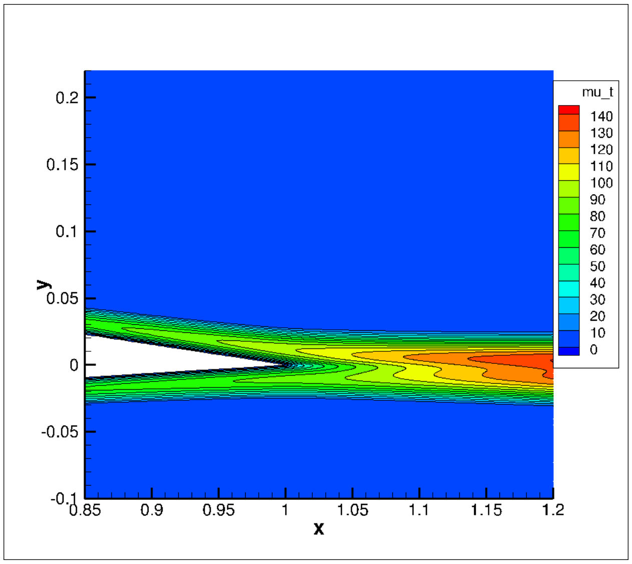

Figure 5.26: Nakayama: FUN3D SA μ_t/μ Contours. Figure compiled using data from NASA (2019).

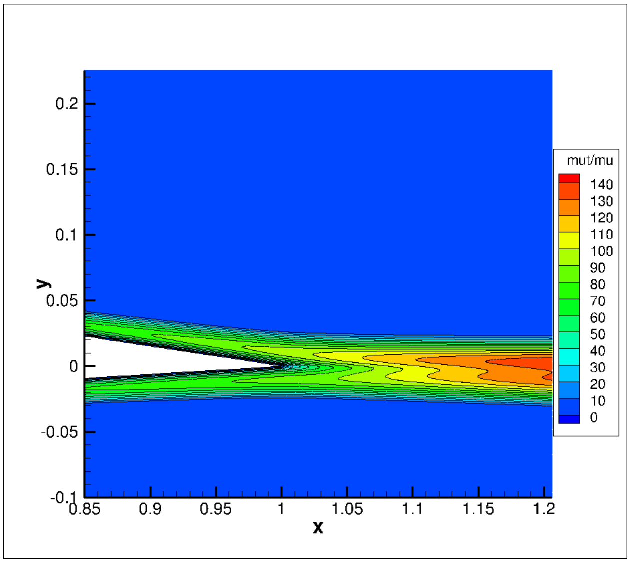

Figure 5.27: Nakayama: POD UNS3D SA μ_t/μ Contours.

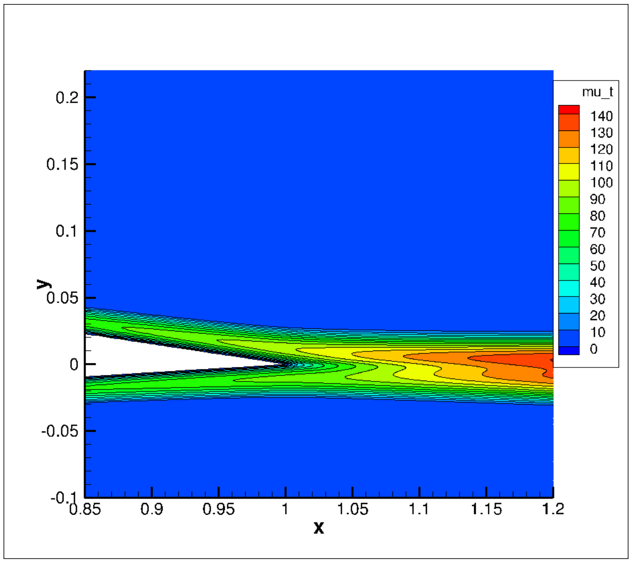

Figure 5.28: Nakayama: FUN3D SA μ_t/μ Contours. Figure compiled using data from NASA (2019).

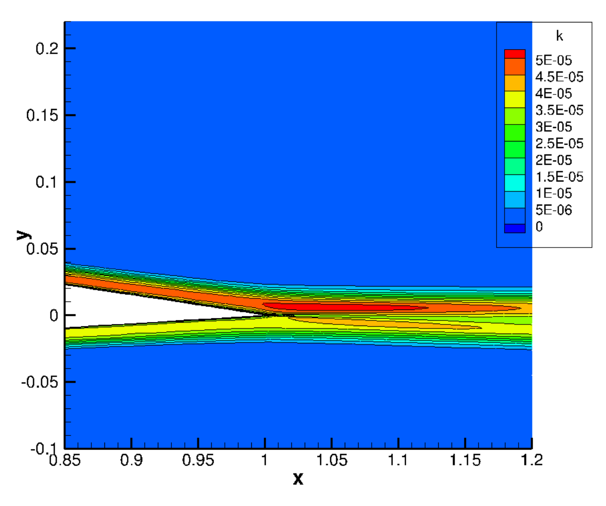

Figure 5.29: Nakayama: SEQ UNS3D SST κ Contours.

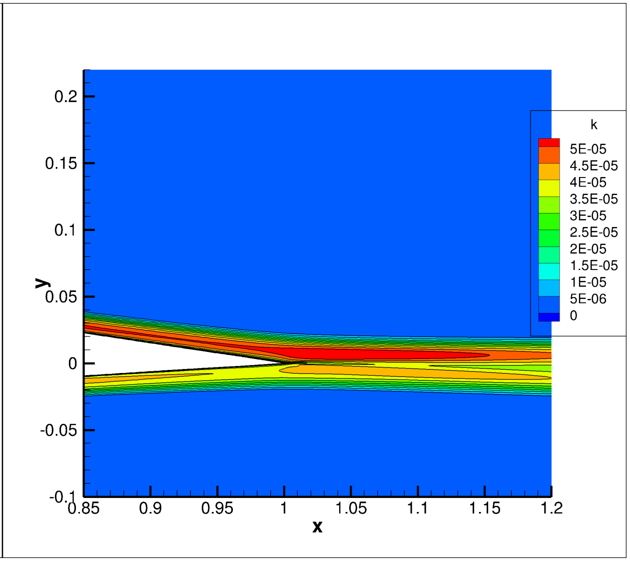

Figure 5.30: Nakayama: FUN3D SST κ Contours. Figure compiled using data from NASA (2019).

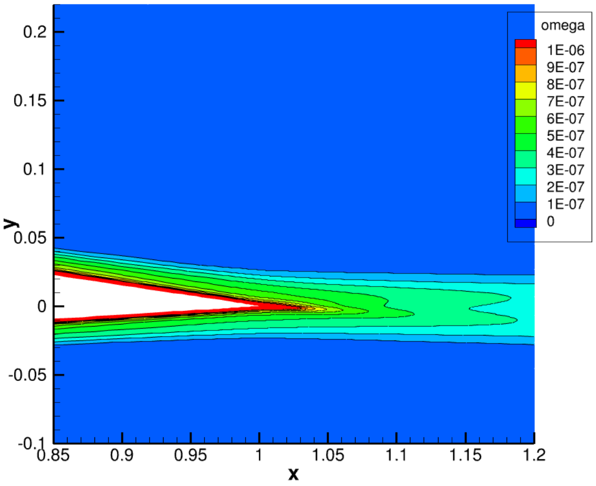

Figure 5.31: Nakayama: SEQ UNS3D SST ω Contours.

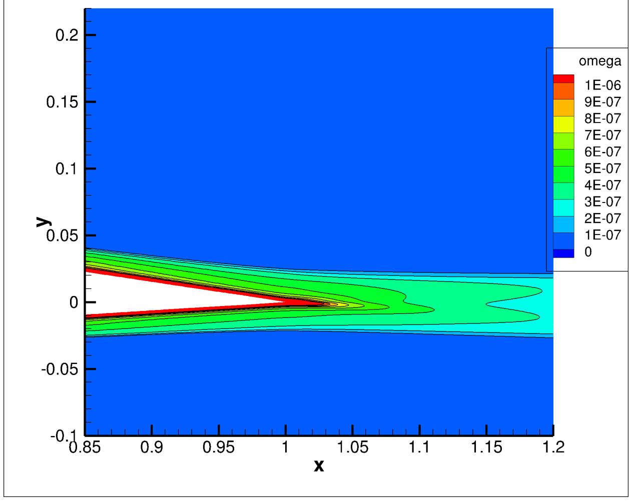

Figure 5.32: Nakayama: FUN3D SST ω Contours. Figure compiled using data from NASA (2019).

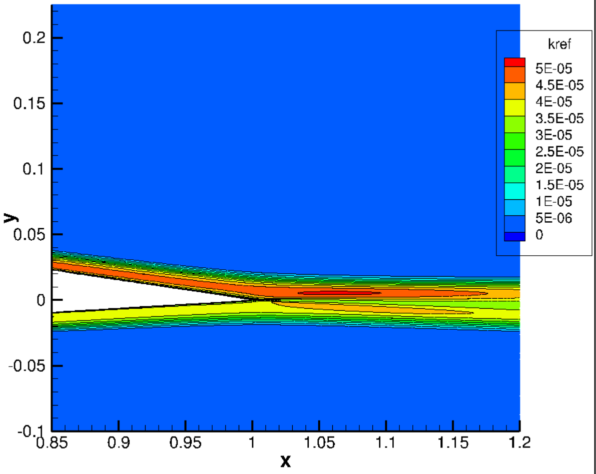

Figure 5.33: Nakayama: POD UNS3D SST k Contours.

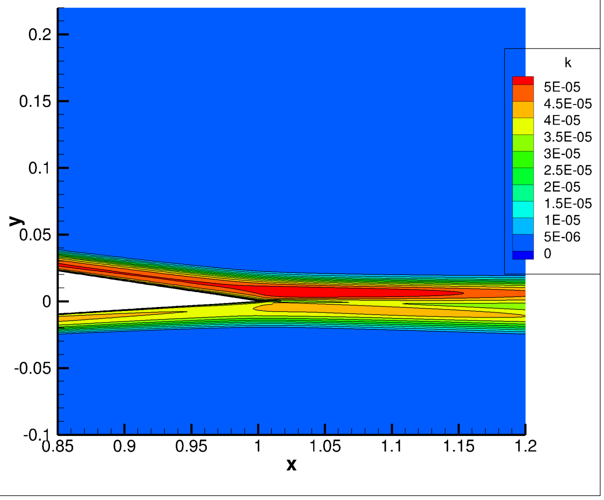

Figure 5.34: Figure compiled using data from NASA (2019).

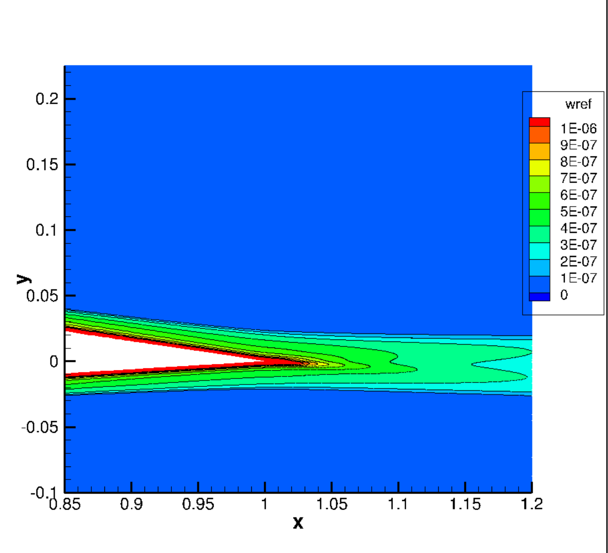

Figure 5.35: Nakayama: POD UNS3D SST ω Contours.

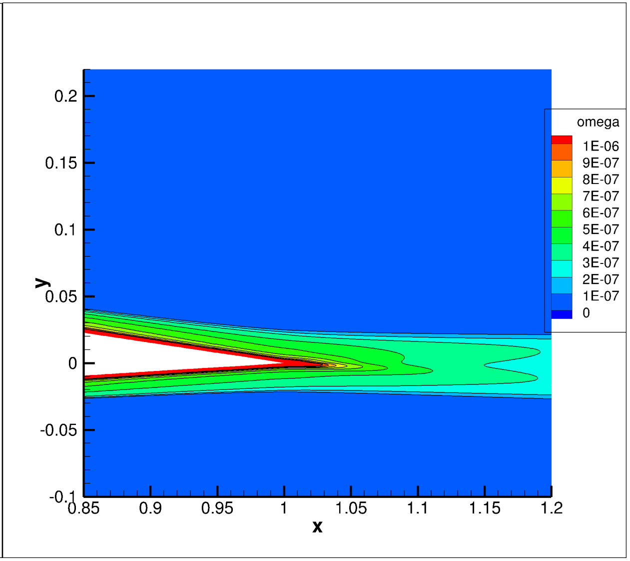

Figure 5.36: Nakayama: FUN3D SST ω Contours. Figure compiled using data from NASA (2019).

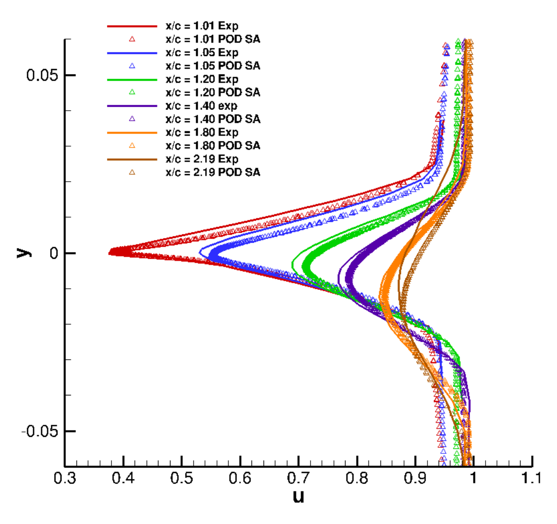

Figure 5.37: Nakayama: POD UNS3D SA Velocity Profiles. Figure used data from NASA (2019).

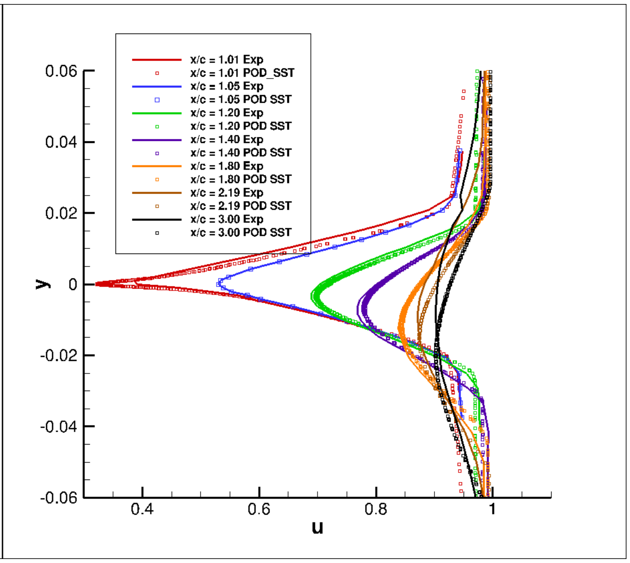

Figure 5.38: Nakayama: POD UNS3D SST Velocity Profiles. Figure used data from NASA (2019).

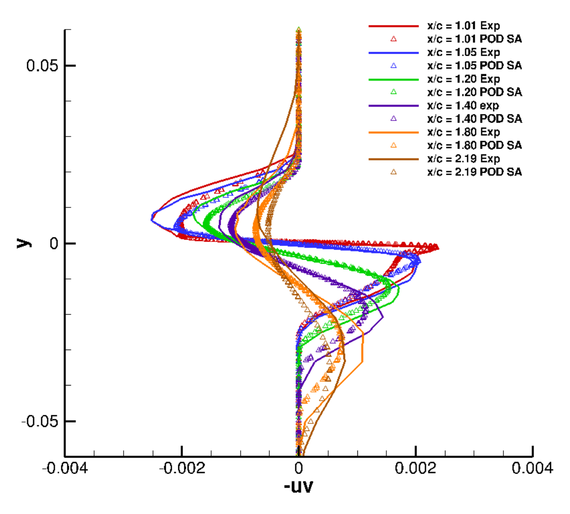

Figure 5.39: Nakayama: POD UNS3D SA Turbulent Velocity Profiles. Figure used data from NASA (2019).

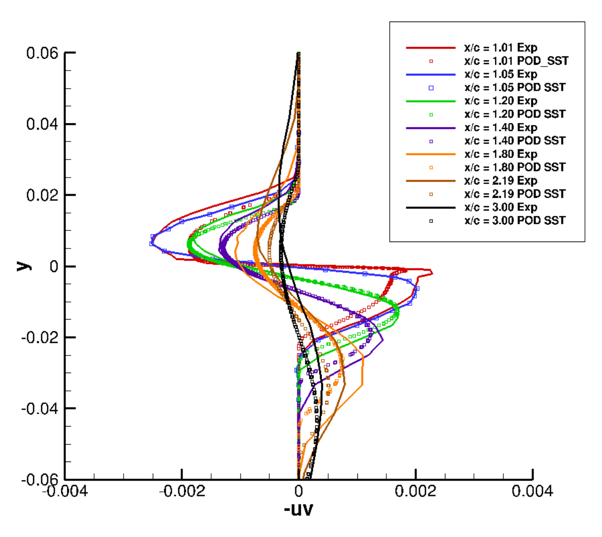

Figure 5.40: Nakayama: POD UNS3D SST Turbulent Velocity Profiles. Figure used data from NASA (2019).

Both turbulence models show comparable results to the FUN3D turbulence models. When comparing the velocity profiles of each turbulence models against experimental results, the Shear Stress transport matched the experiment better than the Spalart-Allmaras model. The Shear Stress Transport also performed better in matching the experimental results for the turbulent velocity fluctuations. Both the Spalart-Allmaras and Shear Stress Transport models agree with experimental results quite well.

5.1.3 2D Backwards Facing Step

The backwards facing step is a standard case to test how well the turbulence model simulates separated flows. This case is precursor to the separated flows that are common in turbomachinery seals. The grids used for this test case was the second coarsest and finest grid provided by the NASA Turbulence Modeling NASA (2019). The sequential and parallel UNS3D results were compared against NASA FUN3D and CLF3D results which used the finest grid. The Plot3D files provided by NASA had the grid separated into 4 different blocks. This test case was proposed in Driver and Seegmiller (1985). The boundary conditions are shown in Fig. 5.41.

The verification and validation data for the turbulence models include velocity profiles, skin friction coefficients, and reattachment points.

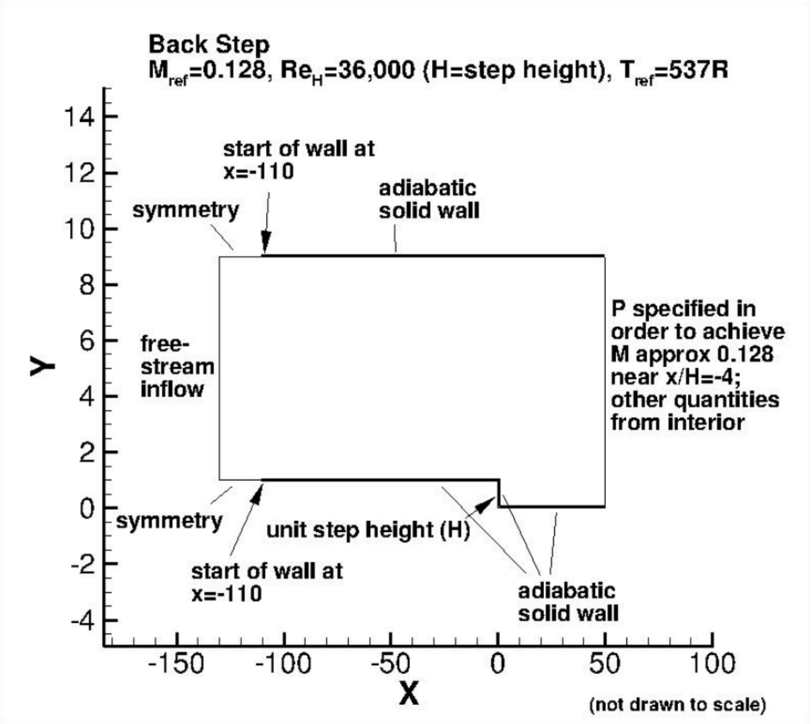

Figure 5.41: Back Step Boundary Conditions. Figure obtained from NASA (2019).

The sequential UNS3D and CFL3D velocity profile generated using the S-A model agree quite

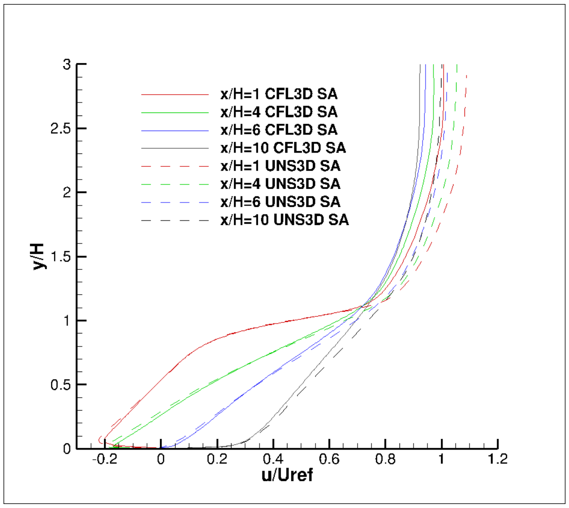

Figure 5.42: S-A Velocity Profiles for a Backwards Facing Step. Figure used data from NASA (2019).

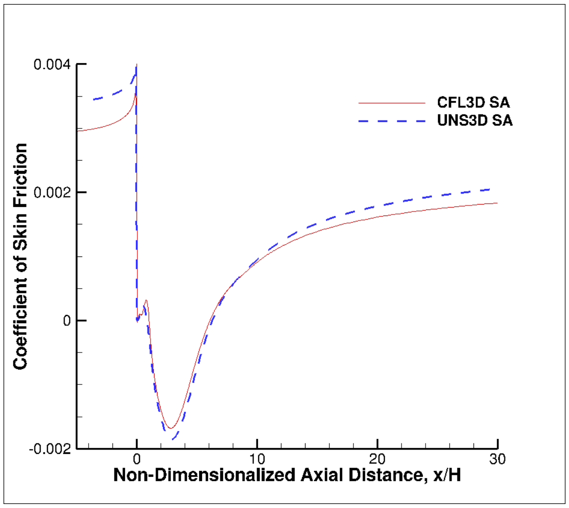

Figure 5.43: [S-A Coefficient of Friction Profile for a Backwards Facing Step. Figure used data from NASA (2019).

well, as shown in Fig. 5.42. Figure 5.43 shows that the skin friction coefficients corresponding to UNS3D and CFL3D agree rather well, in spite of the fact that sequential UNS3D used a coarser grid.

The reattachment point for the FUN3D S-A and SST models are x = 6.1 and x = 6.5, and the reattachment point for the experiment is 6.26 ± 0.01. For the UNS3D SA and SST models, the reattachment point was located at x = 6.065 and x = 6.8 respectively. The resolution in the x direction is about 0.1, so grid refinement is needed to properly locate the reattachment point and for more accurate results. The SST model took quite few more iterations to converge to a density residual of 10^{-11}, compared to the SA model.

The same case was ran on the finest grid provided by NASA turbulence modeling resource using the parallel version of UNS3D, POD UNS3D. Both the Spalart-Allmaras and Shear Stress Tranpsort models were compared against NASA's CFL3D flow solver along with experimental results. The simulation using UNS3D SST had difficulty converging, as did CFL3D SST. The CFL3D SST simulation resorted to running the simulation time-accurately until a steady-state solution was achieved. The simulation for POD UNS3D SST did not run time-accurately.

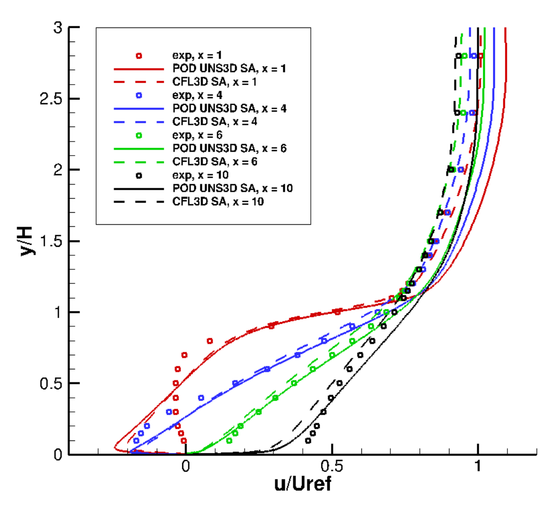

Figure 5.44: Backstep: Spalart-Allmaras Velocity Profiles. Figure used data from NASA (2019).

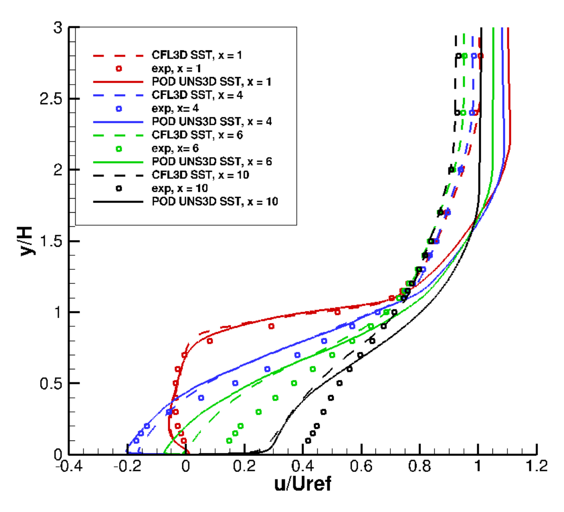

Figure 5.45: Backstep: Shear Stress Transport Velocity Profiles. Figure used data from NASA (2019).

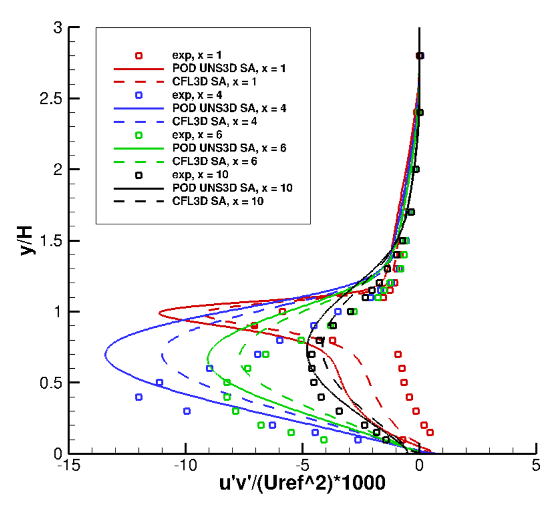

Figure 5.46: Backstep: Spalart-Allmaras Turbulent Velocity Profiles. Figure used data from NASA (2019).

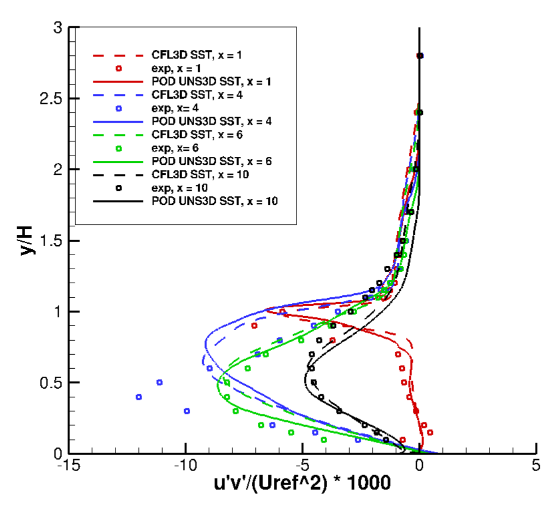

Figure 5.47: Backstep: Shear Stress Transport Turbulent Velocity Profiles. Figure used data from NASA (2019).

The backwards facing step case is a very interesting problem as turbulence models generally have difficulty modeling separated flows. This also allows comparisons for how well each turbulence model and their flow solvers simulate separated flows. When comparing the velocity profiles in Figures 5.44 and 5.44, one can observe that the Shear Stress Transport turbulence model matches the experimental velocity profile better than the Spalart-Allmaras velocity profile near the back step (x=0). After x = 4, the Spalart-Allmaras turbulence model generally performs better than the Shear Stress model. Both the CFL3D and UNS3D flow solvers have similar velocity profiles for each turbulence model, but the CFL3D flow solver seems to match experimental data as y/H increases.

For the turbulent fluctuations in Figures 5.46 and 5.47, the UNS3D Spalart-Allmaras turbulence model had significantly different results compared to CFL3D. UNS3D SST and CFL3D SST had comparable results to each other, and matched the experimental data better than the Spalart-Allmaras turbulence models in general.

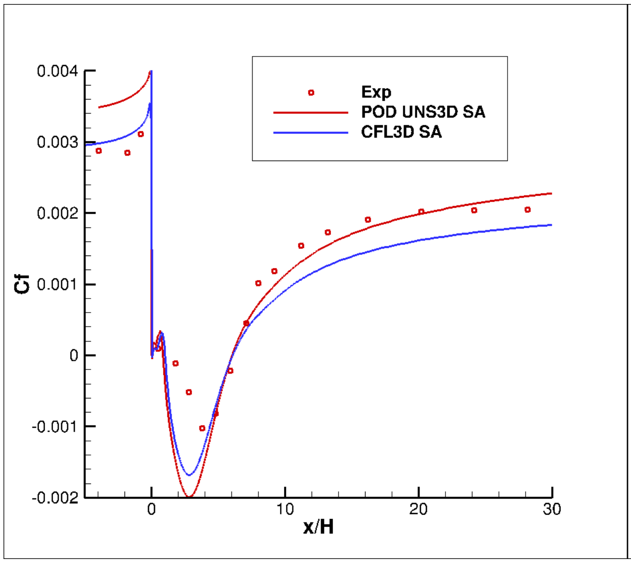

Figure 5.48: Backstep: Spalart-Allmaras Skin Friction Coefficient. Figure used data from NASA (2019).

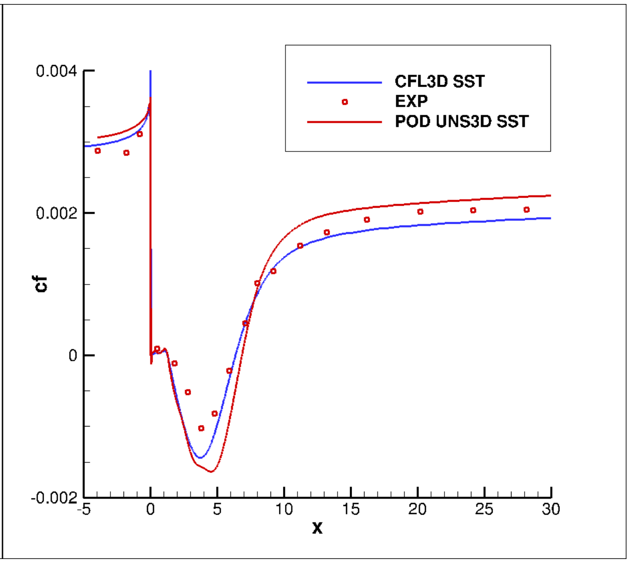

Figure 5.49: Backstep: Shear Stress Transport Skin Friction Coefficient. Figure used data from NASA (2019).

Comparisons for the skin friction coefficients are found in Figures 5.48 and 5.49. The reattachment point in the flow field for each flow solver and turbulence model is determined by locating where the skin friction coefficient, Cf, transitions from a negative value to a positive value. Experimental data shows a reattachment point occurring near a x/H value of 6.26. The reported reattachment points for CFL3D's SA and SST turbulence models are $x/H = 6.1$ and $x/H = 6.5$, respectively. The reattachment points for UNS3D SA and SST models are $x/H = 6.013$ and $x/H = 6.832$. Both CFL3D turbulence models performed better than UNS3D in prediciting the reattachment point. The results from POD UNS3D SST might be improved if the simulation was ran time-accurately until a steady-state was achieved. Both flow solvers show that the Spalart-Allmaras turbulence model under predicts the reattachment point while the Shear Stress model over predicts.

5.1.4 11th Standard Configuration

The 11th Standard Configuration represents a set of published experiments by Fransson et al. (1999) done on a turbine blade cascade geometry. These experiments include steady, unsteady, subsonic and off-design transonic cases. The cases with unsteady flow obtain their unsteadiness from the forced plunging motion of the airfoil cascades. Only one steady-state case with stationary airfoils were simulated. This test case was selected to test the turbulence model in turbomachinery flows, test the periodicity of the turbulence model, and to provide a steady-state solution to eventually test the time-linearized turbulence model.

The simulation was done on one airfoil and utilized periodic boundary conditions to simulate the rest of the cascade. So far, only the steady, subsonic case has been simulated with the parallel Spalart-Allmaras and Shear Stress Transport turbulence model. Below is the plot of isentropic Mach number vs position along the chord of the 11th Standard Configuration case 100 (steady-state, subsonic), where the published experimental and numerical results of various solvers from Fransson et al. (1999) can be compared to UNS3D with the Spalart-Allmaras Turbulence model. The flow conditions for STCF 11 case 100 are listed in Table 5.3. The chord length in the experiment was 65 mm and was chosen to be the reference length for determining the Reynolds number.

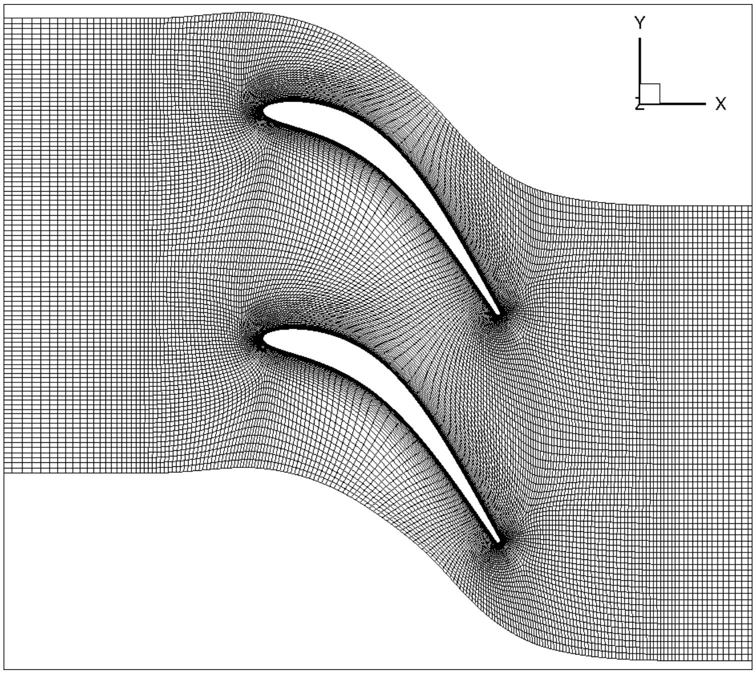

Figure 5.50: 11th Standard Configuration Mesh.

P_{tot} [Pa]	T_{tot} [K]	M_{inlet} [-]	Re_L [-]	μ_{ref} [Pa s]	$\frac{\tilde{\nu}}{\nu}$ [-]	$\omega_\infty \frac{\nu_L}{c_\infty^2}$ [-]	$\kappa_\infty \frac{1}{c_\infty^2}$ [-]
124,600	330	0.31	446,000	1.846e-5	3	4 x 10^-8	4 x 10^-10

Table 5.3: Input parameters for STCF 11 case 100.

Here we see that both the Spalart-Allmaras and Shear Stress Transport turbulence models of UNS3D are in good agreement with the experimental results and outperforms the published non-linear viscous code results. Both turbulence models produce nearly identical results. A grid independence study was performed with a mesh that was twice as fine and the results were the same as the coarser mesh results presented in Figure 5.51.

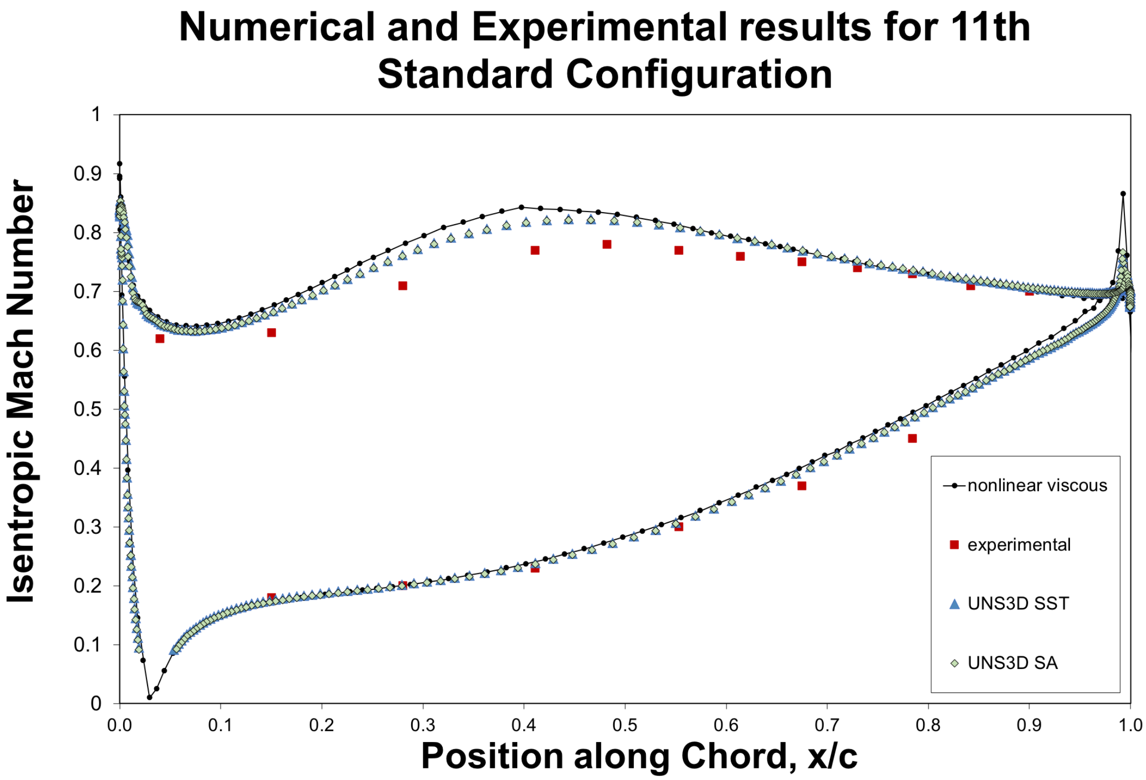

Figure 5.51: Experimental and non-linear viscous data from Fransson et al. (1999).

5.2 Verification & Validation of the Time-Linearized Flow Solver

The verification and validation of the time-linearized flow solver began with the task of reproducing the results published by Liliedahl (2017), using his codes, meshes and input files. The cases to be reproduced include an inviscid channel with oscillating back pressure, Stokes' second problem, and a straight annular gas seal. The ultimate goal for the time-linearized flow solver was to be able to predict rotordynamic stability coefficients for gas seals. The gas seal simulated was based off of the gas seal used in the high pressure oxidizer turbopump used in the space shuttle main engine. The rotordynamic stability coefficients were compared against results from Nelson's bulk flow method (Nelson, 1985).

5.2.1 1.0 m Inviscid Channel

An inviscid channel with oscillating back pressure was chosen as a numerical experiment to verify that the time-linearized flow solver may match the results obtained from a full-order flow solver such as UNS3D. The experiment is ideal for eliminating the effects of a moving computational grid, and to focus on the time-linearized flow solver's ability to calculate the unsteady flow caused by oscillatory flow variables only. A 1 m x 0.02 m x 0.02 m inviscid channel was used for this experiment. The back pressure oscillated with an amplitude of 5 kPa at a frequency of 700 Hz. A description of the mesh and boundary conditions may be found in Figure 5.52. The mesh consisted of 8 x 8 x 100 equally spaced nodes.

The results of the UNS3D and UNS3D TL flow solvers are compared against each other. Figure 5.54 shows the comparisons of the pressure amplitudes along the length of the channel between the time-linearized results and the full-order model. Figure 5.54 shows that the time-linearized results are in good agreement with the full-order model.

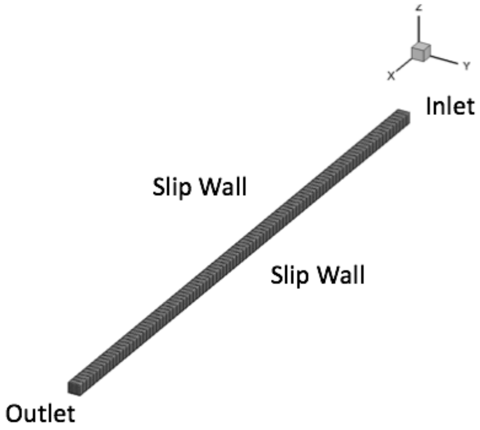

Figure 5.52: Invisicid Channel: 1-meter mesh.

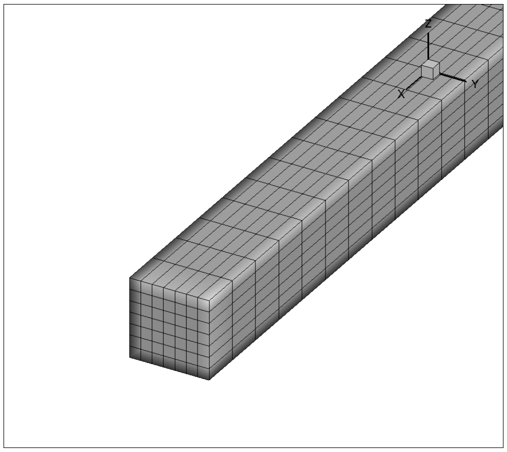

Figure 5.53: Inviscid Channel: 1-meter mesh zoomed

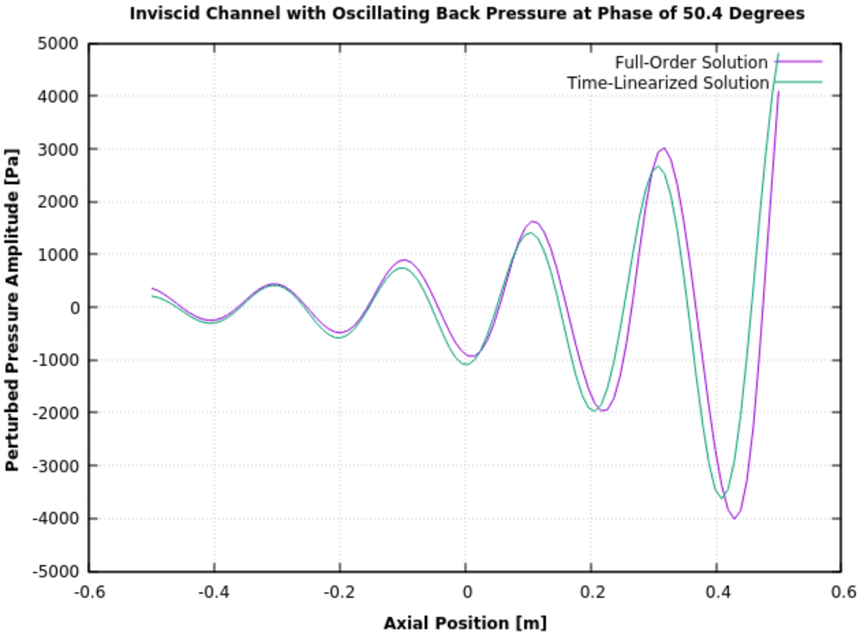

Figure 5.54: Inviscid Channel: 1-meter Pressure Amplitude Comparisons.

5.2.2 Oscillating Infinite Plate: Stokes' Second Problem

To verify that the time-linearized solver captures the effects of the deformation of the grid, Stokes' Second Problem was simulated as there is a solution to Navier-Stokes that can be compared to. Stokes' problem is an excellent case for a time-linearized solver as the solution can be compared to the analytical solution, and the flow is solely dominated by viscosity and the motion of the plate. This provides an excellent opportunity to test the effects of the moving mesh, and to verify that the moving mesh effects on the flux calculations are working correctly. A description of the mesh and boundary conditions is shown in Figure 5.55.

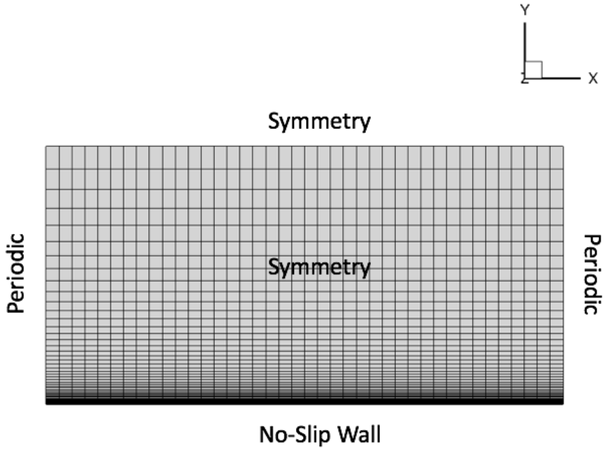

Figure 5.55: Stokes' 2nd Problem: Mesh and Boundary Conditions

Stokes' Problem was also used to verify that the time-linearized flow solver works well with the perturbed grid produced by the new grid deformation tool which utilizes radial basis functions. An observation made during this case is that the perturbed mesh must be nondimensionalized, otherwise the solution may diverge. As shown below, the grid produced by radial basis functions works quite well with the time-linearized flow solver. The graphs in Figure 5.56 show the time-linearized velocity profile versus the analytical solution at a frequency of 550 Hz.

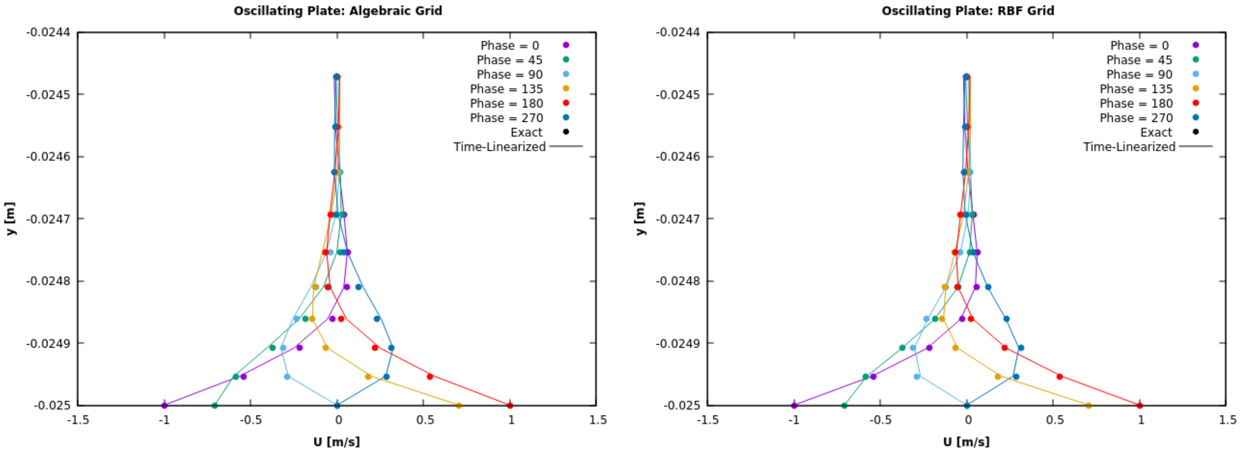

Figure 5.56: Stokes' 2nd problem: comparison of flow profile at various phase angles between exact solution and time-linearized solution, using an algebraic grid (left) and a grid developed with the Complex Grid Deformation Tool.

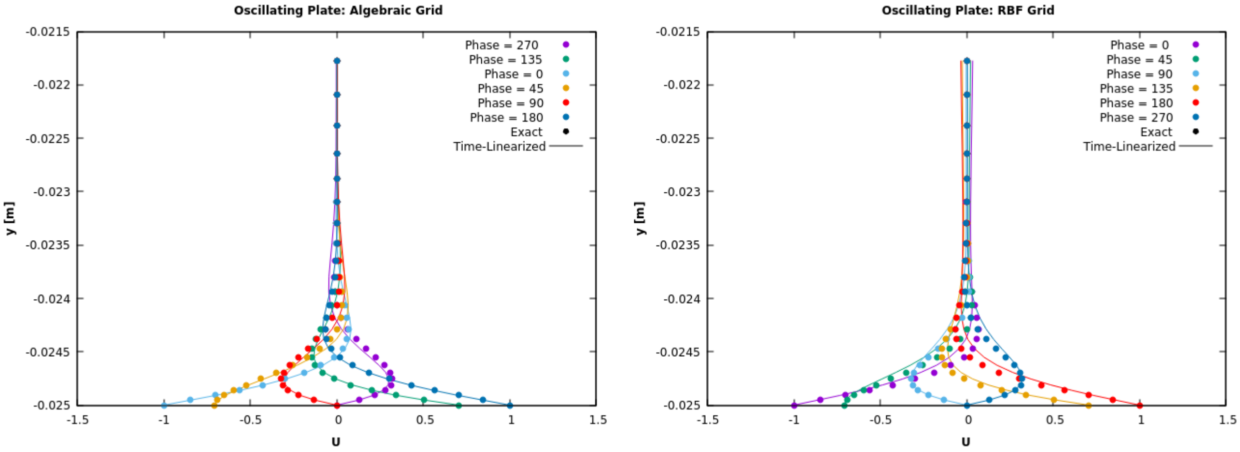

Figure 5.57: Stokes' 2nd problem: comparison of flow profile at various phase angles between exact solution and time-linearized solution, using an algebraic grid (left) and a grid developed with the Complex Grid Deformation Tool at 275 Hz.

This case was also ran at a lower frequency of 275 Hz, where both the algebraic and radial basis function grids were compared against each other. It was found that both perturbed grids did not match the analytical solution for Stokes' second problem and that each grid type produced different results. Of the two grid types, the linear deformation of the algebraic grid produced much better results. This is due to the fact that the radial basis function deformation performs better when deforming interior grid points only.

5.2.3 Straight Annular Gas Seal: Nelson Seal

Rotordynamics is the study of the dynamics of spinning rotors. Due to mass imbalances and perturbations, a rotor never truly spins about the center axis, but instead spins about the center axis in what is known as a whirling orbit. For turbomachinery, the orbit radius must be kept to a minimum in order to prevent catastrophic failure from the blades hitting the hub. Gas seals are used primarily to prevent flow leakage, but also to help stabilize the whirling rotor. To predict whether a gas seal stabilizes a rotor, the rotordynamic stability coefficients are determined. The stability coefficients are calculated by obtaining the aerodynamic forces acting on the shaft, and determining the direct and cross-coupled stiffness and damping values from the equations of motion of the rotor. The common methods for prediciting rotordynamic coefficients are the bulk flow methods, a full-order CFD simulation, and experiments. Bulk flow methods often require experimental results in order to finely tune each method. Time-linearization is a state of the art tool for predicting rotordynamic stability coefficients, as it reduces the problem size of a full-order CFD simulation by truncating the computational domain and solving for unsteady flows in the frequency domain.

The straight annular gas seal that was simulated followed the dimensions and parameters described in Nelson (1985) and listed in Table 5.4.

Total Inlet Pressure [MPa]	Back Pressure [MPa]	Length [mm]	Radius [cm]	Clearance [mm]	Shaft Rotational Velocity [rpm]	Shaft Whirl Velocity [rpm]
1.52	0.65	6.5	3.25	0.086	30,400	30,400

Table 5.4: Input parameters for Annular Seal (HPOTP).

For this research, only the results for a seal with a length-to-diameter ratio of 0.1 are shown, as there were difficulties reproducing the results from Liliedahl (2017). Many numerical experiments were performed in order to obtain results that were close to the original results published by Liliedahl (2017), where parameters such as inlet and outlet boundaries, rotating wall boundary conditions, and post-processing methods were examined.

The initial simulation using the same input, mesh and post-processing codes from one of Liliedahl's previous simulations produced the following results, where the results from Nelson (1985) and Liliedahl (2017) are also listed in Table 5.5 for comparison. The results listed are for a mesh that truncated the full annulus into a quarter of a degree slice. Liliedahl used 2nd order spatial accuracy along with non-reflecting boundary conditions at the inlet and outlet.

Origin	Inlet/Outlet BC	K	k	C	c
Nelson (1985)	-	588,000	-1,030	17.8	0.011
Liliedahl (2017)	Non-reflecting	580,000	6,400	60	7
Present Research	Standard	447,606	-25,275	60.28	5.5
Present Research	Non-reflecting	170,117	-25,607	54	4.6

Table 5.5: Quarter-degree slice: Rotordynamic Results

As shown, the non-reflective boundary conditions have a significant impact on the accuracy of the results. The present research results using the same conditions that Liliedahl had used produced significantly different results for the direct and cross-coupled stiffness coefficients, where as the direct and cross-coupled damping coefficients match the results from Liliedahl (2017) quite well.

Due to the significant differences from the present research and Liliedahl (2017), an investigation into the complex rotational periodic boundary conditions, non-reflecting boundary conditions, size of degree slice, and post-processing methods were performed. Among the experiments performed was how accurate the force acquisition was for predicting rotordynamic coefficients. The results from Table 5.6 were obtained by simulating a two-degree slice of the full annulus. Table 5.6 also shows the comparisons between the reference speeds, or how the frequency is non-dimensionalized, and the method for obtaining the complex forces. The Axisymmetric force method refers to the method described in 4.4, where solely the complex pressures located at the points closest to the rotor hub at time, t = 0, are used to obtain the complex forces. The axisymmetric method was used to obtain the results from Table 5.5. The Force Integrator method refers to the newly implemented method described in Section 4.5, where the complex pressures are integrated along the wall surfaces of each cell. The Force Integrator also rebuilds the full annulus using rotations and phase shifts of the surface areas and complex pressures, respectively.

Force Method	Inlet BC	Outlet BC	Reference Speed	K	k	C	c
Axisymmetric	3	1	347	529,546	149	91	3.69
Axisymmetric	3	1	511	529,546	149	62	2.5
Force Integrator	3	1	347	536,531	-18,767	91.24	0.53
Force Integrator	3	1	511	536,531	-18,767	62	0.36

Table 5.6: Two-degree slice: Rotordynamics Results

The reference speed is the reference speed of sound in which the harmonic frequency is non-dimensionalized by. The results from Table 5.6 were simulated at a dimensionless value of $\omega_{hrm} = 9$, which is in the vicinity of 30,400 RPM if using 347 m/s as the reference speed of sound. The reference speed of 511 m/s is the reference speed of sound from the reference temperature for the Nelson seal, and would suggest the frequency is 44,000 RPM. Table 5.6 shows that the reference speed only affects the values of the damping coefficients.

Table 5.6 shows some disparity between the results obtained in Table 5.5. This suggests that there is a limit on how much one is able to truncate the computational domain, as the two-degree slice results are favorable to the results from Liliedahl (2017); Nelson (1985). Table 5.6 shows differences between the results obtained from each force acquisition method, primarily in the cross-coupled stiffness terms. The axisymmetric method also reported that the complex forces were about 50x larger in magnitude compared to the force integrator method.

While these results are not far off from the results published by Liliedahl (2017), the cross-coupled stiffnesses and disparity between the complex forces suggested that the time-linearized solver should undergo additional testing before simulating additional seals and testing the time-linearized Spalart-Allmaras turbulence model.

5.2.4 0.1 m Inviscid Channel

The purpose of this experiment was to observe how well the time-linearized code matches the full-order model when the inlet and outlet were spatially closer to each other. What typically happens is that the inlet and outlet reflect acoustic waves that tend to disrupt the accuracy and convergence of the final solution.

The channel used for this case was a 0.1-meter channel with a square cross section of 0.04 meters squared. The mesh used was 10 x 10 x 50 equally spaced nodes. A description of the mesh and its boundary conditions may be found in Figure 5.58.

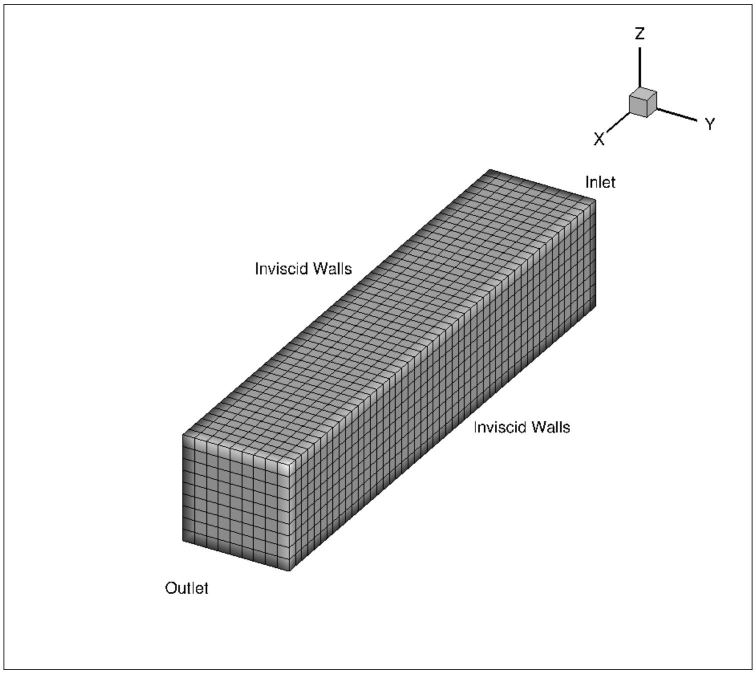

Figure 5.58: Invisicid Channel: 0.1-meter mesh.

The channel had an back pressure which was oscillating at an amplitude of 5 kPA with a frequency of 700 Hz. It was found that for the time-linearized code, the solution would not converge unless a non-reflective inlet boundary condition was used. The full-order model had no difficulty with convergence as the acoustic waves were dissipated in real-time.

Figure 5.59 - 5.61 shows that after 10 periods of oscillating the back pressure, the full-order solution has reached a state of periodicity. Figure 5.62 shows the amplitudes of pressure along the center line of the channel at a point in time. The lines in Figure 5.62 represents the amplitudes of pressure from the time-linearized flow solver and the full-order model. The difference between amplitudes is most likely due to the time-linearized flow solver's inability to capture all nonlinear effects.

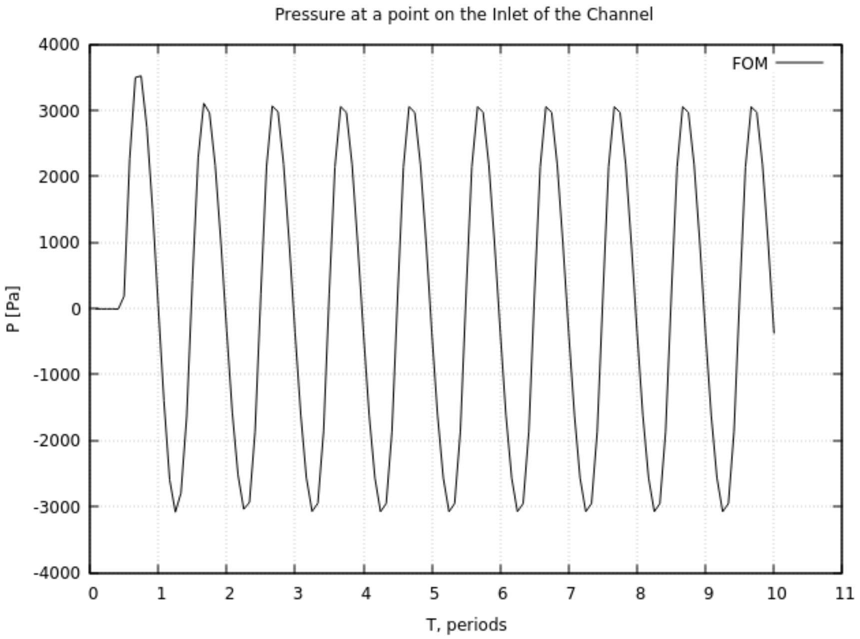

Figure 5.59: Amplitude of Pressure at inlet of 0.1-meter Channel.

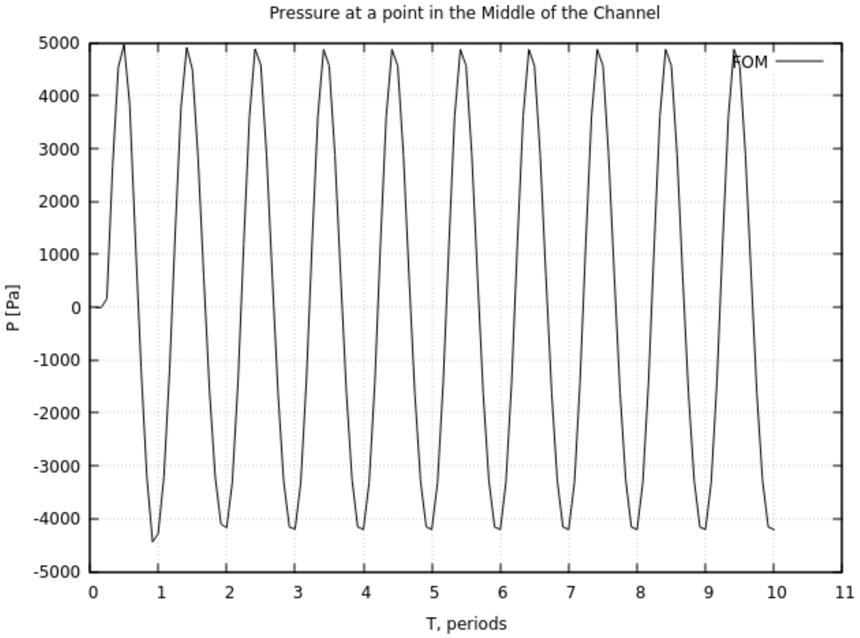

Figure 5.60: Amplitude of Pressure in the Middle of 0.1-meter Channel.

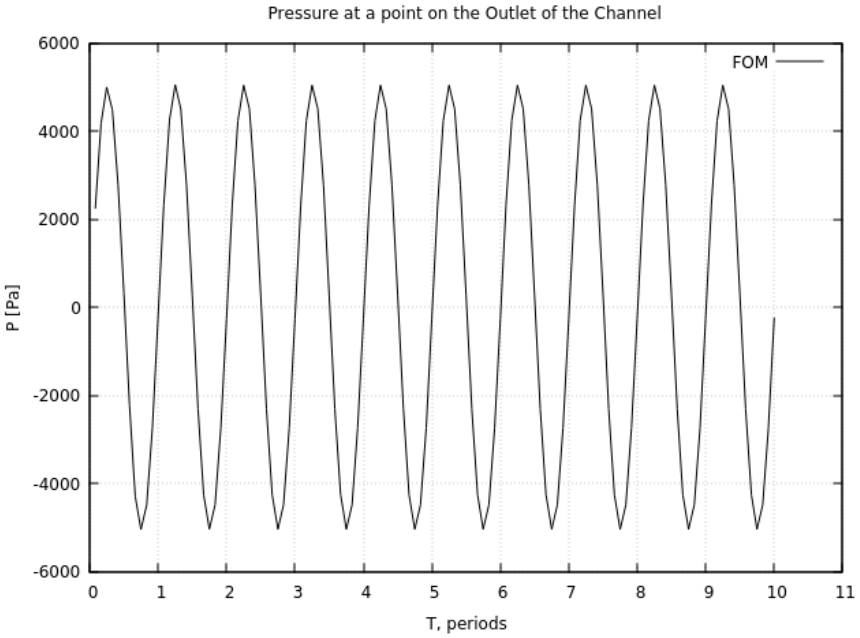

Figure 5.61: Amplitude of Pressure at outlet of 0.1-meter Channel.

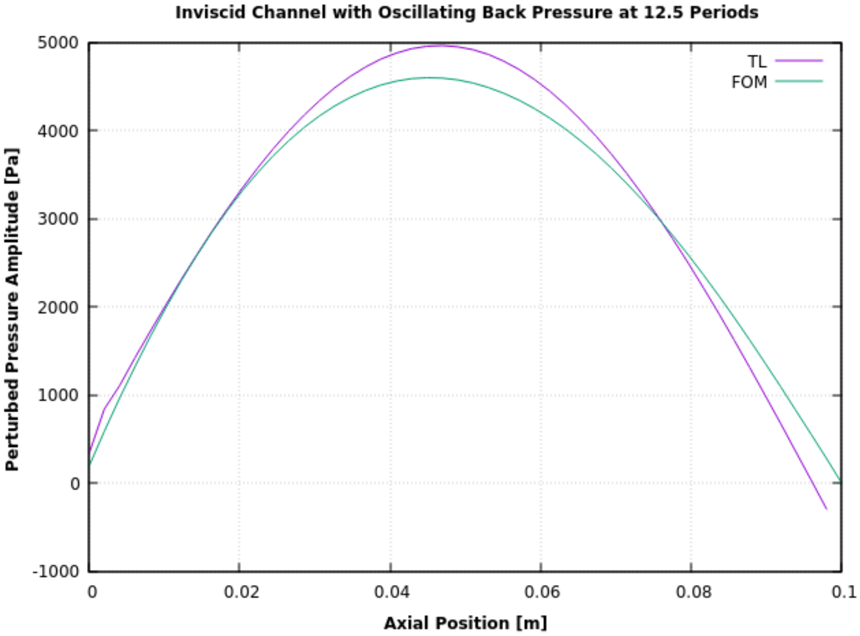

Figure 5.62: Inviscid Channel: 0.1-meter Pressure Comparisons.

5.2.5 Inviscid Translating Rotor

An axisymmetric smooth seal with zero pressure gradient was simulated with inviscid flow. The rotor translated in only one direction. Both the full-order model and time-linearized model were used to simulate this case. This case was used to verify multiple aspects of the time-linearized flow solver against the results of the full-order flow solver. The ability to capture the effects of a moving computational grid was compared to the full order model. The time-linearized flow solvers' new force integration method was also tested against the results of the full-order model. A grid independence study was also performed with the time-linearized flow solver, where each grid density was also compared against the results of the full-order model. This case was also used to understand how the complex rotational periodic boundaries were enforced, and how the time-linearized flow solver is able to reduce the size of the computational domain.

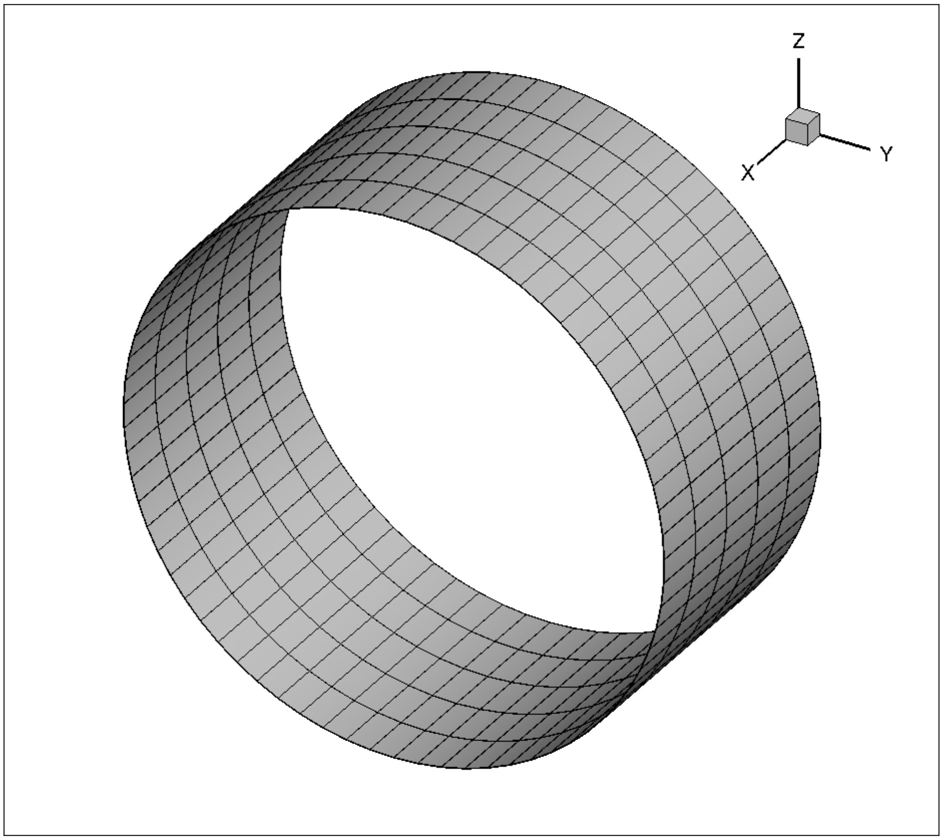

Figure 5.63: Inviscid Translating Rotor.

Three grids were used for the grid independence study with each grid doubling the amount of nodes in each direction. The forces in the translational direction, the direction of motion, were used to compare the time-linearized flow solver to the full-order model. For the time-linearized flow solver, the first-order terms were observed to see whether the real and imaginary parts of the complex amplitudes would converge towards a specific real part and a specific imaginary part.

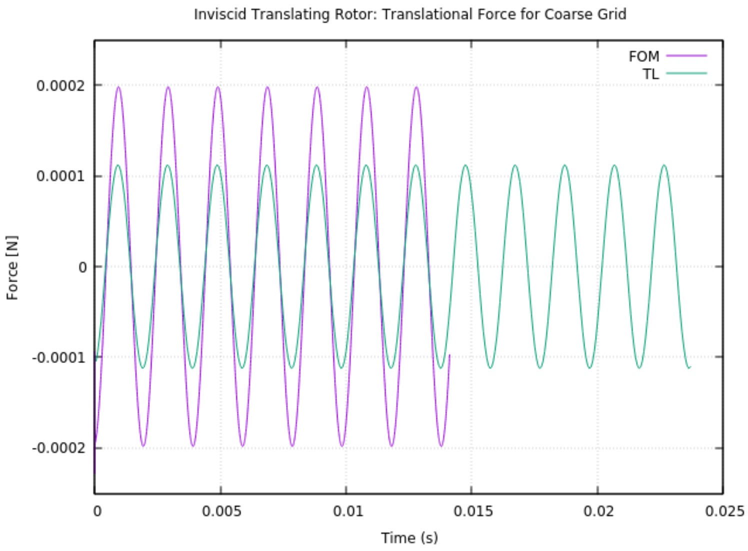

Figure 5.64: Inviscid Translating Rotor: Coarse Mesh Forces.

For the coarse mesh results shown in Figure 5.64, the full-order model immediately reaches a state of periodicity. The time-linearized results has an amplitude about 50 percent smaller than the full-order model.

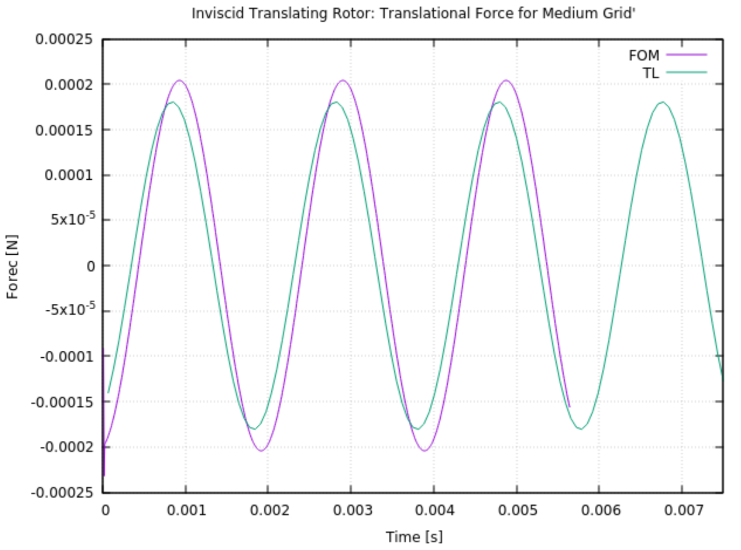

Figure 5.65: Inviscid Translating Rotor: Medium Mesh Forces.

The medium mesh results are presented in Figure 5.65. The full-order model quickly reaches a state of periodicity, and reaches an amplitude of oscillation of about 0.0002 Newtons. The time-linearized flow solver is out of phase with the full-order flow solver, most likely due to small transient effects, and has a amplitude which is about 10 percent smaller than the full-order model.

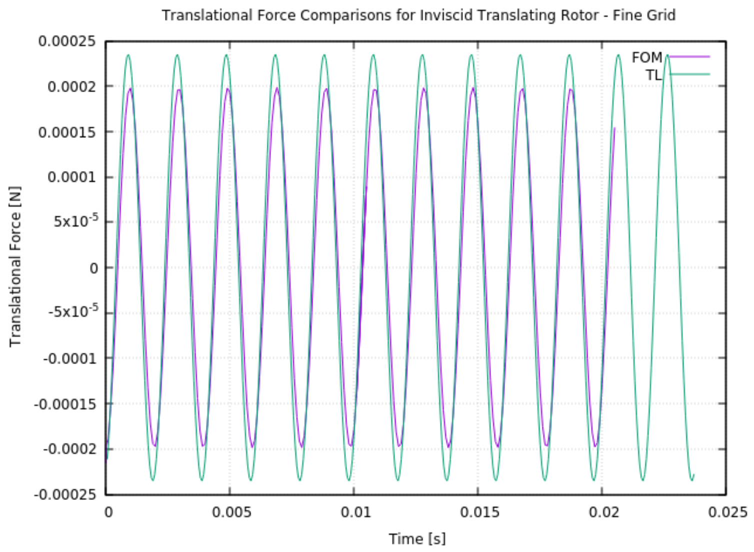

Figure 5.66: Inviscid Translating Rotor: Fine Mesh Forces.

The fine mesh results are presented in Figure 5.66. Here it is observed that the time-linearized model has a larger amplitude than the full order model results.

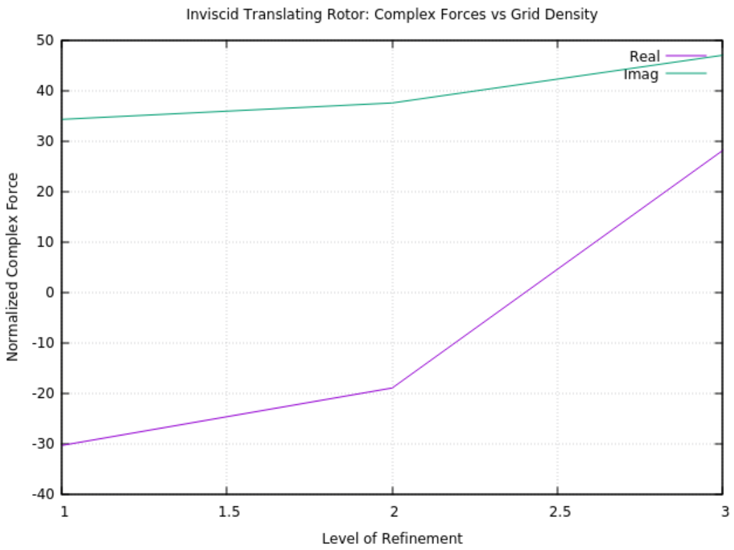

Figure 5.67: Inviscid Translating Rotor: Complex Forces vs Grid Density.

The real and imaginary parts of the complex forces for each level of grid density are shown in Figure 5.67. Figure 5.67 shows that the real and imaginary forces may be converging to a value but finer grids are required to determine whether or not grid independence can be reached. From Figures 5.64 - 5.66, it is shown that the forces obtained from the full-order model remain relatively constant as grid refinement varies. The same cannot be said for the forces obtained from the time-linearized flow solver. It was found that the complex rotational periodic boundaries work as intended from this case, as the complex state vector from $\theta = 0°$ with a phase shift of 90 degrees matched the complex state vector at $\theta = 90°$.

Table 5.7 show the run times for each of the meshes used. The full-order model (FOM) results were obtained using Texas A&M's Terra super computer, which utilizes Intel Xeon E5-2680 v4 2.40 GHz 14-core processors. The time-linearized results were obtained using a single core from an Intel Core i7-6700 3.40GHz processor. One can compare the computational run times of the full-order model (FOM) and the Time-Linearized model (TL) against each other. It can be observed that the time-linearized model is computationally cheaper compared to the full order model.

Mesh	FOM CPU	FOM Run Time(per processor) (seconds)	Total FOM time (seconds)	TL CPU	TL Run Time (seconds)
Coarse	20	54,736 (11 periods)	1,094,720	1	6,121
Medium	4	72,000 (3 periods)	288,000	1	132,000
Fine	28	144,000 (10.5 periods)	4,032,000	1	1,958,520

Table 5.7: Inviscid Translating Rotor: Computational Run Times

The primary expense of the full order model stems from obtaining a periodic state. While this case shows that a periodic state is quickly reached for the full-order model, one must still simulate more than one period of oscillation to ensure periodicity has been reached. The time-linearized solver shows that it can obtain a periodic solution an order of magnitude or higher faster than the full order model. The time linearized flow solver required 150,000, 800,000 and 2,000,000 pseudo time-steps to match the full-order model for the coarse, medium and fine grids, respectively.

5.2.6 Viscous Flat Plate with Oscillating Back Pressure

UNS3D TL needs to verify that its laminar modeling is working properly. To verify that the model is working correctly, UNS3D TL will be compared against the viscous results that is obtained from UNS3D. The case selected was a 2D viscous flat plate with oscillating back pressure. Ideally, one would like to use the a channel mesh similar to the one used in Section **??**. A channel was not selected due to the level of mesh refinement required to properly model the boundary layers of the channel. Thus, a 2D flat plate requires viscous spacing on one wall, in only two dimensions. Figure 5.68 describes the mesh and boundary conditions used. The mesh featured a flat plate that was 0.1 meter long. A laminar flow model was used for this simulation, where the back pressure oscillated at an amplitude of 5 kPa at 700 Hz. The time-linearized flow solver utilized non-reflecting inlet boundary conditions. The following flow parameters for this simulation is listed in Table 5.8.

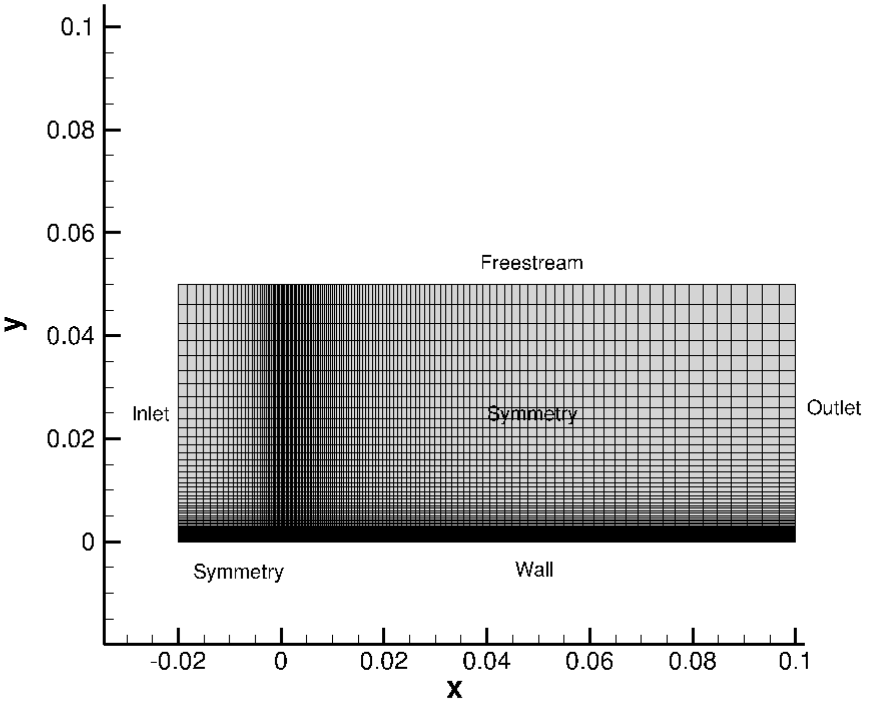

Figure 5.68: Viscous Plate Mesh.

Re	Mach	P_tot [Pa]	Pback [Pa]	T_tot [K]	Frequency [Hz]
1,328,000	0.6	129,420.42	101,325	321.6	700

Table 5.8: Viscous Oscillating Back Pressure: Parameters

The forces acting on the plate were used to compare the time-linearized flow solver to the full-order RANS solver. Figures 5.69 and 5.70 shows the shear and normal forces acting on the plate for both the full order model (FOM) and the time-linearized flow solver (TL).

Figure 5.69 shows that the full-order model mean force has shifted compared to the time-linearized shear forces. This shift in the mean would affect the time-linearized solver from calculating the correct perturbations. Figure 5.70 shows the normal forces acting on the plate, where

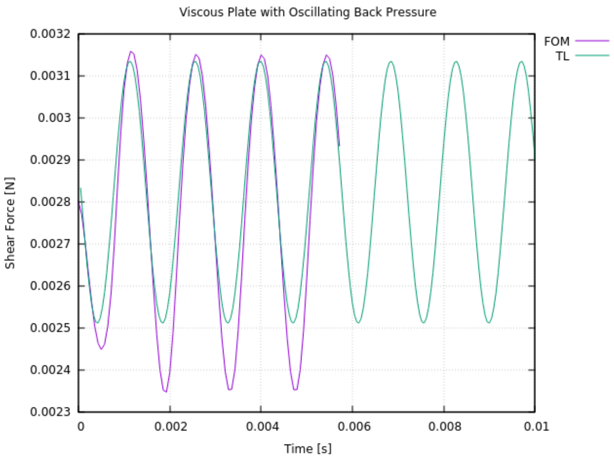

Figure 5.69: Viscous Oscillating Back Pressure: Shear Force.

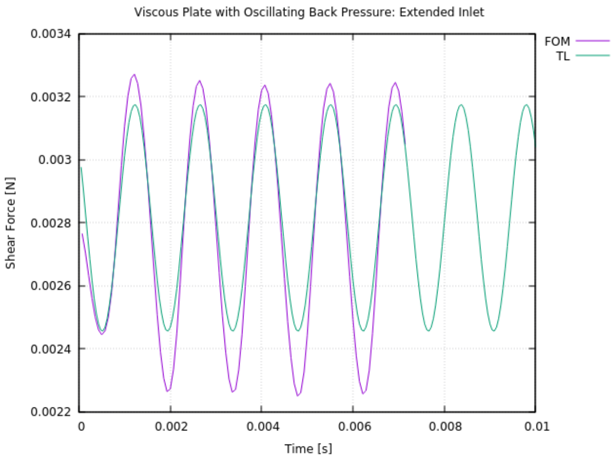

Figure 5.71: Extended Viscous Oscillating Back Pressure: Shear Force.

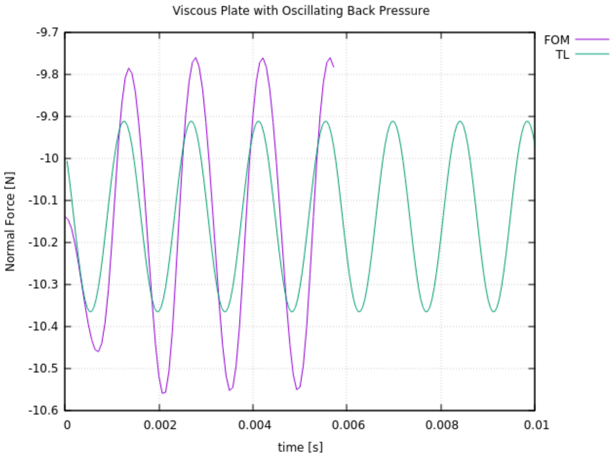

Figure 5.70: Viscous Oscillating Back Pressure: Normal Force.

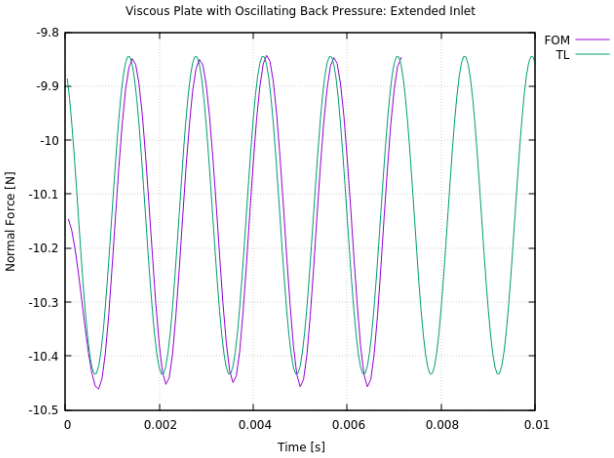

Figure 5.72: Extended Viscous Oscillating Back Pressure: Normal Force.

the time-linearized flow solver has a lower amplitude for the normal force obtained compared to the full-order model. The inlet being in close proximity to the wall and symmetry boundary was suspected to be the source of error for the time-linearized forces.

Another simulation was performed where the inlet was located 5 reference lengths away from the wall boundary. Figures 5.71 and 5.72 show the normal and shear forces obtained using the extended inlet mesh. Both the shear and normal forces are in better agreement to the full-order model. This suggests that there may be limitations to the time-linearized non-reflecting boundary conditions in regards to how much the distance from the the inlet to object of interest can be truncated.

Mesh	FOM CPU	FOM Run Time (per processor) (seconds)	Total FOM Time (seconds)	TL CPU	TL Run Time (seconds)
Original	20	33,892 (4 periods)	677,840	1	177,625 seconds
Extended	20	69,852 (5 periods)	1,397,040	1	178,080

Table 5.9: Viscous Plate with Oscillating Back Pressure: Computational Run Times

The full-order model (FOM) results were ran an Texas A&M's Ada supercomputer, which utilizes Intel Xeon E5-2670 v2 (Ivy bridge-EP), 10-core 2.50 GHz processors. The time-linearized simulation was ran on a single core from a Intel Core i7-6700 3.40GHz processor. The time-linearized solution required around 300,000 and 350,000 pseudo time-steps to match the full order model, for the extended inlet and original mesh respectively. The extended inlet run converged faster than the original mesh. Overall, the time-linearized flow solver has about an order of magnitude speed up compared to the full order model for this case.

5.2.7 Viscous Translating Rotor

The viscous translating rotor case was chosen to observe how well the time-linearized laminar model compares to the full-order model for a viscous moving mesh. For this comparison, a full annulus of a straight annular seal was simulated to avoid the influence of the rotational periodic boundary conditions. The parameters that define the flow are listed in Table 5.10.

The geometry is similar to the Nelson seal, except that the length of the seal is much shorter to reduce the computational time of this experiment. The length of the seal is 2 centimeters instead of 6.5 centimeters in the Nelson case. Figure 5.73 shows the geometry and mesh of the viscous translating rotor. This case was designed to be simpler than the Nelson seal, in order to observe where the time-linearized flow solver had difficulty. This case was ran to have first-order spatial

Re	Mach	P_tot [Pa]	Pback [Pa]	T_tot [K]	Frequency [Hz]
75,000	0.15	135,038	113,931	300	506.66

Table 5.10: Viscous Translating Rotor: Parameters

accuracy. Instead of a whirling orbit as the Nelson case has, the rotor was oscillated in only one direction. Rotating wall boundaries were not used. The force components were used to compare the full-order model and the time-linearized model.

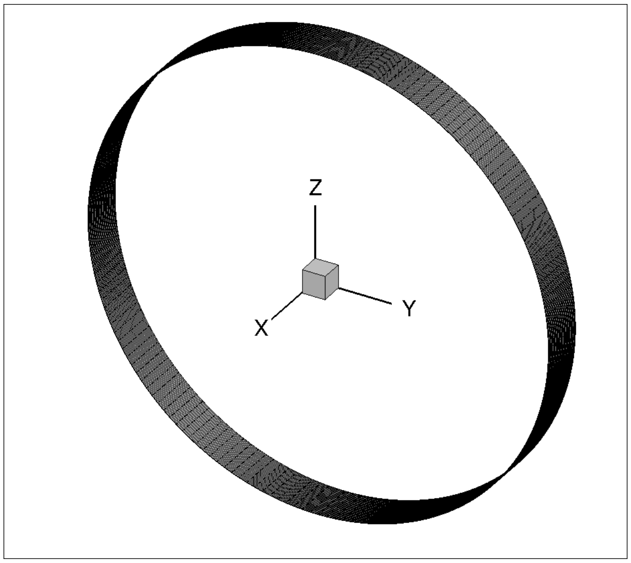

Figure 5.73: Viscous Translating Rotor Mesh.

The time-linearized translational forces match with the full order model quite well. The other forces have interesting results that shows some of the limitations for the time-linearized model. It can be observed in Figure 5.75 that the axial forces for the full order model have obtained a new steady-state value. A different steady-state value will lead to incorrect values in the time-linearized results, as the time-linearized results depend on the steady-state terms. Figure 5.76 shows another limitation of the time-linearized flow solver. Here it is observed that full-order model does not have

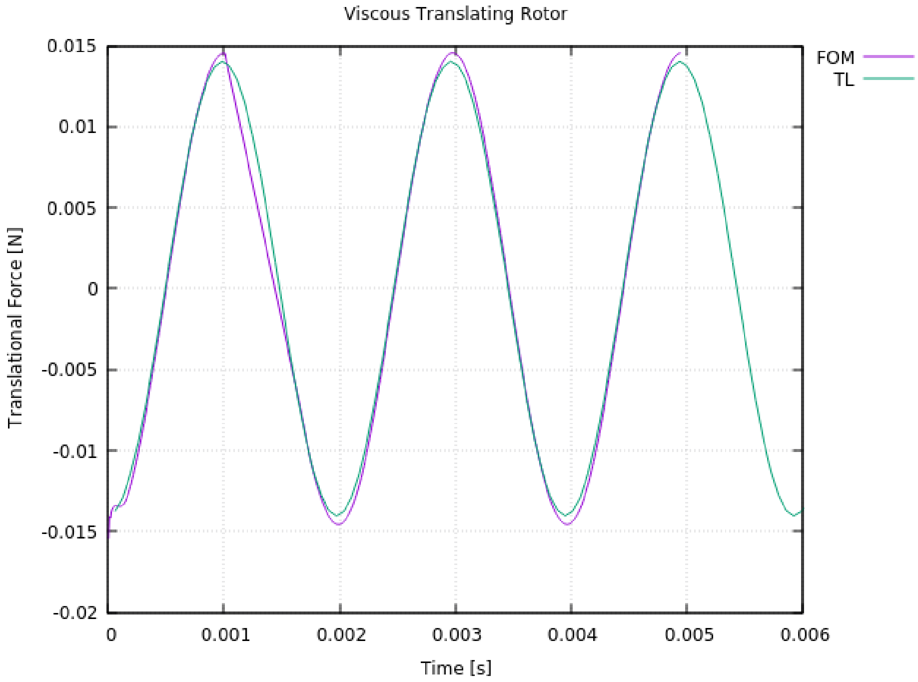

Figure 5.74: Viscous Translating Rotor: Translational Force.

any positive values for the force in the z-direction, while the time-linearized force does. This is due to the time-linearized solution being harmonic, thus there will be positive and negative values added to the mean steady-state term.

FOM CPU	FOM Run Time (per processor)	Total FOM TIME	TL CPU	TL Run Time
20	926,838 seconds (2.5 periods)	18,536,760 seconds	1	66,664 seconds

Table 5.11: Viscous Translating Rotor: Computational Run Times

Table 5.11 shows the computational run times of the viscous translating rotor. The full-order model (FOM) results were obtained using Texas A &M's Terra super computer, which utilizes Intel Xeon E5-2680 v4 2.40 GHz 14-core processors. The time-linearized results were obtained using a single core from an Intel Core i7-6700 3.40GHz processor. It can be observed that the time-linearized solution is significantly faster than the full order model. The time-linearized solver

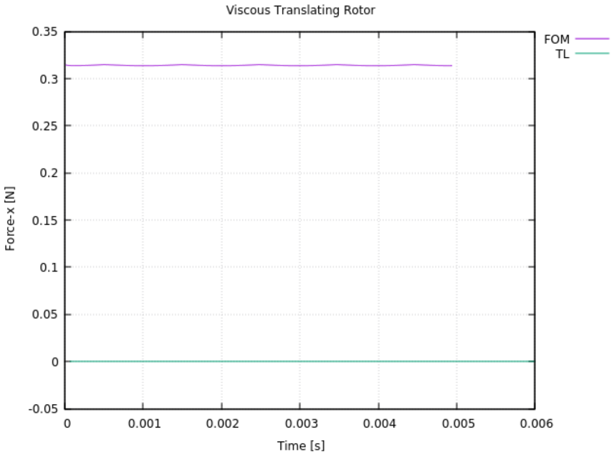

Figure 5.75: Viscous Translating Rotor: Axial Force.

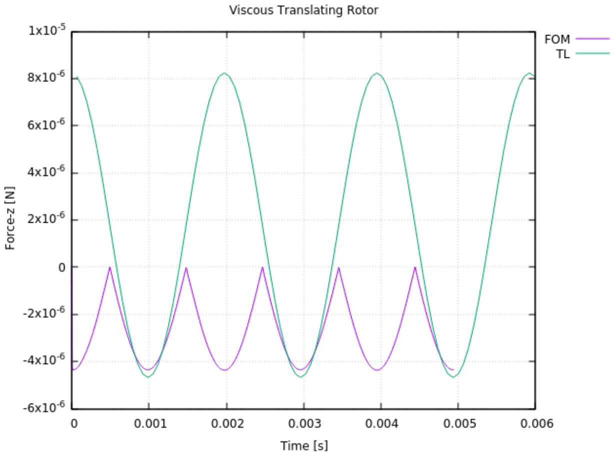

Figure 5.76: Viscous Translating Rotor: Force, z-component

required 30,000 pseudo time-steps to match the full-order model forces, which is considerably quicker than the other cases. For this case, the time-linearized flow solver was 280 times faster than the full-order model.

6. CONCLUSIONS

In comparison to NASA's FUN3D flow solver and experimental results, the newly implemented Spalart-Allmaras turbulence model has been successfully verified and validated. The Shear Stress Transport turbulence model shows great results as well, the only concern being that the POD UNS3D κ and ω contours do not match FUN3D's contours. At the same time, the NASA flow solver ran freestream boundary conditions of κ and ω that were not in the recommended range proposed by Menter (1994). Other than the κ and ω contours, POD UNS3D Shear Stress Transport turbulence model matched experimental results quite well. Both the Shear Stress Transport and Spalart-Allmaras turbulence models were able to test their accuracy when it came to separated flows on the backwards facing step case. When it comes to separated flows, the Spalart-Allmaras turbulence model did not perform as well as the Shear Stress Transport turbulence model against experimental results for the turbulent fluctuations and skin friction coefficient plots. Spalart-Allmaras performed better than the Shear Stress transport in the velocity profiles as the flow moved away from the step. Overall, the Shear Stress Transport tended to have better accuracy than the Spalart-Allmaras for the cases simulated. The Shear Stress Transport had more difficulty converging than the Spalart-Allmaras model, and showed signs of sensitivity towards farfield boundary conditions.

Case	Turbulence Model	# of CPU	Run time per CPU	Total run time
Flat Plate	SA	80	253,500 s	20,280,000 s
Flat Plate	SST	80	351,720 s	28,137,600 s
Nakayama	SA	80	946,880 s	75,750,400 s
Nakayama	SST	80	1,584,000 s	126,720,000 s

Table 6.1: Turbulence Model Run Times for Convergence .

Table 6.1 shows some of the run times for the NASA test cases. It is shown that the Spalart-

Allmaras turbulence model has a significantly faster run time than the Shear Stress Transport. This is primarily due to the Spalart-Allmaras turbulence model reaching convergence in fewer iterations than the Shear Stress Transport. Per iteration, the Spalart-Allmaras turbulence model is about 10 - 20 percent faster than Shear Stress Transport. It is also important to note that in some cases such as the Backwards facing step, the Shear Stress Transport did not converge for a steady state simulation, whereas the Spalart-Allmaras model did. The Spalart-Allmaras turbulence model does not match experimental results as well as the Shear Stress Transport model, but it is able to reach a converged state with a reduced computational expense.

For the time-linearized flow solver, it was evident that there are some issues that need to be resolved before continuing the implementation of the time-linearized Spalart-Allmaras turbulence model. The inviscid channels with oscillating back pressure was a very successful case for the time-linearized code. These cases showed that the time-linearized inviscid model works well. From the inviscid translating rotor case, it is observed that the time-linearized code may not be grid independent, or that it is more sensitive to grid refinement than the full-order model. The newly implemented Force Integrator was used to calculate the forces on the rotor, which showed good agreement with the medium mesh, and calculating the complex forces derived from the pressure amplitudes.

Stokes' second problem showed that the time-linearized viscous laminar model can accurately predict simple viscous flows. It also showed that the effects of a complex moving mesh were working properly for both the algebraically deformed grid and the grid deformed using the complex radial basis functions. This case unfortunately diverges due to the odd boundary condition combination of having no inlet and outlet, using periodic boundaries to simulate the infinite plate instead. This prevented the dissipation of reflections and growths in the flow field, and actually amplified them instead. When solving for different frequencies of oscillation, it was noted that the steady-state mesh must have enough points in the boundary layer that will be created by the deformation of the mesh to achieve an acceptable level of accuracy. While this is true for any full-order simulation, it shows that the meshes for time-linearized simulations must correspond to the appropriate level of refinement corresponding to the frequency of oscillation. A desirable characteristic of time-linearization is the concept of being able to efficiently simulate an object at multiple frequencies using only one steady state flow field. This is true as long as the steady state mesh is refined enough to simulate all levels of frequencies. For good practice, it may be beneficial to have more viscous refinement than would be typically needed for full-order simulations.

The results for the viscous plate with an extended inlet and oscillating back pressure showed that the time-linearized viscous model can replicate the shear forces with some accuracy. The Force Integrator was also used to determine the forces from the time-linearized model, and showed that both the pressure and viscous forces are in good agreement with the full order model.

The viscous translating rotor case showed the limitations of the time-linearized model in general. While the time-linearized model matched the translating force on the rotor quite well, the axial forces and forces in the z-direction showed the limitations of the reduced order model.

Comparisons between the unsteady full-order model simulations and the time-linearized model were made in Section 5.2. For these simulations, the forces and the computational run times were compared against a multi-processor full-order model versus a single processor time-linearized model. For all of the cases, the single processor time-linearized model obtained solutions in real time that was less or comparable to the run times of the multi-processor full-order model. The reduction in time was shown to be case dependent, but the time-linearized model was shown to achieve a speed up between 1-3 orders of magnitude compared to the full-order model.

6.1 Future Work

While the Spalart-Allmaras turbulence model has been successfully verified and validated, there are a few more aspects that need to be tested. The Spalart-Almaras turbulence model has very little experience simulating unsteady and fully three dimensional flows. Additional test cases should be simulated that cover these flow features.

For the time-linearized flow solver, there are many boundary conditions and tests that still need to be performed. The inviscid rotor test suggested that the time-linearized flow solver may be grid dependent or sensitive to the refinement of the computational grid. Additional grid independence studies should be performed. Regarding boundary conditions, tests need to be performed to ensure that the rotating wall, and both translational and rotational complex periodic boundaries are working properly along with their limitations. The time-linearized flow solver seemed to have difficulty converging whenever periodic boundary conditions were used. Experiments relating a full annulus and a slice of an annulus should be performed to test the complex rotational periodic boundary conditions and the force integrator. Once the boundary conditions are verified and validated, rotordynamic analysis can be performed where the full-order model is compared against the time-linearized model. With the force integrator and new complex grid deformation tool, complex seal geometries such as stepped labyrinth seals can under go rotordynamic analysis with ease.

Lastly, the time-linearized Spalart-Allmaras turbulence model has been implemented into the time-linearized flow solver, but has not performed any verification or validation tests. A turbulent flat plate with oscillating back pressure is a good place to start verifying the time-linearized turbulence model with the full-order model. Validation cases can include the unsteady 11th Standard Configurations cases. The 11th Standard Configuration can also be an opportunity to utilize the time-linearized flow solver's ability to simulate different interblade phase angles for blade cascades by adding a phase shift to the periodic boundaries.

REFERENCES

Allmaras, S. R., F. T. Johnson, and P. R. Spalart (2012, July). Modifications and clarifications for the implementation of the spalart-allmaras turbulence model. In *Seventh International Conference on Computational Fluid Dynamics*, Number ICCFD7-1902, Big Island, HI. 19, 20, 27

Arghir, M. and J. Frene (1997, July). Rotordynamic coefficients of circumferentially-grooved liquid seals using the averaged navier–stokes equations. *Journal of Tribology 119*, 556–567. 3

Barth, T. J. and D. C. Jesperson (1989). The design and application of upwind schemes on unstructured meshes. AIAA Paper 89-0366. 32

Blazek, J. (2001). *Computational Fluid Dynamics: Principles and Applications* (1st ed.). Elsevier. 36

Cizmas, P. G. A. and K. C. Hall (1995, August). Computation of steady and unsteady viscous flows using a simultaneously coupled inviscid-viscous interaction technique. *Journal of Fluid and Structures 9*(6), 639–657. 3

Clark, W. S. and K. C. Hall (1999, June). A time-linearization navier-stokes analysis of stall flutter. In *International Gas Turbine & Aeroengine Congress & Exhibition*, ASME Paper 99-GT-383, Indianapolis, Indiana. ASME. 1, 4

Coles, D. (1956, July). The law of the wake in the turbulent boundary layer. *Journal of Fluid Mechanic 1*(2), 191–226. 5, 43

de Boer, A., M. S. van der Schoot, and H. Bijl (2007, jun). Mesh deformation based on radial basis function interpolation. *Comput. Struct. 85*(11-14), 784–795. 6, 25

Dervieux, A. (1985, March). Steady Euler Simulations Using Unstructured Meshes. In J.-A. Essers (Ed.), *Computational Fluid Dynamics, VKI Lecture Series 1985-04*, Volume 1. Rhode Saint Genese, Belgium: von Karman Institute for Fluid Dynamics. 32

Dietzen, F. J. and R. Nordmann (1987, July). Calculating rotordynamic coefficients of seals by finite difference techniques. *Journal of Tribology 109*, 388–394. 3

Driver, D. M. and H. L. Seegmiller (1985). Features of a reattaching turbulent shear layer in divergent channel flow. *AIAA Journal 23*(2), 163–171. 5, 58

Fransson, T., M. Jöcker, A. Bölcs, and P. Ott (1999, Oct). Viscous and inviscid linear/nonlinear calculations versus quasi 3d experimental cascade data for a new aeroelastic turbine standard configuration. *Journal of Turbomachinery 121*(1), 717–725. 5, 62, 64

Gargoloff, J. I. (2007, May). *A Numerical Method for Fully Nonlinear Aeroelastic Analysis*. Ph. D. thesis, Texas A&M University, College Station, Texas. 31, 33

Hall, K. C. (1993, May). Deforming grid variational principle for unsteady small disturbance flows in cascades. *AIAA Journal 31*(5), 891–900. also presented at the 30th Aerospace Sciences Meeting & Exhibit, January 1992, Reno, as AIAA Paper 92-0665. 23

Hall, K. C. and C. B. Lorence (1993, October). Calculation of 3-dimensional unsteady flows in turbomachinery using the linearized harmonic Euler equations. *Transaction of the ASME: Journal of Turbomachinery 115*(4), 800–809. 3

Harten, A. (1983). Self adjusting grid methods for one dimensional hyperbolic conservation laws. *Journal of Computational Physics 50*, 235–269. 30

Holmes, D. G. and C. B. Lorence (1998). Three dimensional linearized navier-stokes calculations for flutter and forced response. In *Unsteady Aerodynamics and Aeroelasticity of Turbomachines*, pp. 211–224. Springer. 1, 4

Jameson, A., W. Schmidt, and E. Turkel (1981). Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. AIAA Paper 81-1259. 30

Kim, K. S. (2003, December). *Three-Dimensional Hybrid Grid Generator and Unstructured Flow Solver for Compressors and Turbines*. Ph. D. thesis, Texas A&M University, College Station, Texas. 31, 33

Liliedahl, D. N. (2017). *A Time-linearized and full-order Navier-Stokes solver for annular gas seal rotordynamic analysis*. Ph. D. thesis, Texas A&M, College Station, TX. 1, 4, 5, 6, 7, 33, 37, 65, 70, 71, 72

Menter, F. R. (1994). Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal 32*(8), 1598–1605. 4, 16, 35, 36, 44, 87

Nakayama, A. (1985). Characteristics of the flow around conventional and supercritical airfoils. *J. Fluid Mech. 160*, 155–179. 5, 52

NASA (2019). NASA turbulence modeling resource. `https://turbmodels.larc.nasa.gov`. Accessed: 2019-03-13. 43, 44, 45, 46, 47, 48, 49, 50, 51, 53, 54, 55, 56, 57, 58, 59, 60, 61

Nelson, C. C. (1985, July). Rotordynamic coefficients for compressible flow in tapered annual seals. *Journal of Tribology 107*, 318–325. 4, 6, 65, 69, 70, 71

Ni, R. H. and F. Sisto (1976, Apr.). Numerical computation of nonstationary aerodynamics of flat plate cascades in compressible flow. *Transactions of the ASME: Journal of Engineering for Power 98*(2), 165–170. 3

Roe, P. L. (1981, July). Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics 43*, 357–372. 30

Rumsey, C. L., B. Smith, and G. Huang (2010). Description of a website resource for turbulence models for reynolds-averaged navier-stokes applications. In *40th AIAA Fluid Dynamics Conference (Paper No. 2010-4742)*. 5

Spalart, P. R. and S. R. Allmaras (1992, January). A one-equatlon turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting & Exhibit*, Number AIAA-92-0439, Reno, NV. 4, 18, 19

Venkatakrishnan, V. (1995). Convergence to steady-state solutions of the Euler equations on unstructured grids with limiters. *Journal of Computational Physics 118*, 120–130. 32

Wagner, N. G., K. Steff, R. Gausmann, and M. Schmidt (2009). Investigations on the dynamic coefficients of impeller eye labyrinth seals. In *Proceedings of the 38th Turbomachinery Symposium, Turbomachinery Laboratory, Texas A&M University, College Station, TX, September*, pp. 14–17. 4