ABSTRACT

Title of Document:	MESH ADAPTION FOR TRACKING VORTEX STRUCTURES IN OVERTURNS SIMULATION OF THE S-76 ROTOR IN HOVER
	John Kaney Hayes, Master of Science, 2016
Directed by:	Dr. James D. Baeder, Department of Aerospace Engineering

The constant need to improve helicopter performance requires the optimization of existing and future rotor designs. A crucial indicator of rotor capability is hover performance, which depends on the near-body flow as well as the structure and strength of the tip vortices formed at the trailing edge of the blades. Computational Fluid Dynamics (CFD) solvers must balance computational expenses with preservation of the flow, and to limit computational expenses the mesh is often coarsened in the outer regions of the computational domain. This can lead to degradation of the vortex structures which compose the rotor wake. The current work conducts three-dimensional simulations using OVERTURNS, a three-dimensional structured grid solver that models the flow field using the Reynolds-Averaged Navier-Stokes equations. The S-76 rotor in hover was chosen as the test case for evaluating the OVERTURNS solver, focusing on methods to better preserve the rotor wake. Using the hover condition, various computational domains, spatial schemes, and boundary conditions were tested. Furthermore, a mesh adaption routine was implemented, allowing for the increased refinement of the mesh in areas of turbulent flow without the need to add points to the mesh. The adapted mesh was employed to conduct a sweep of collective pitch angles, comparing the resolved wake and integrated forces to existing computational and experimental results. The integrated thrust values saw very close agreement across all tested pitch angles, while the power was slightly over predicted, resulting in under prediction of the Figure of

Merit. Meanwhile, the tip vortices have been preserved for multiple blade passages, indicating an improvement in vortex preservation when compared with previous work. Finally, further results from a single collective pitch case were presented to provide a more complete picture of the solver results.

MESH ADAPTION FOR TRACKING VORTEX STRUCTURES IN OVERTURNS
SIMULATION OF THE S-76 ROTOR IN HOVER

By

John Kaney Hayes

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2016

Advisory Committee:
Dr. James D. Baeder, Chair/Advisor
Dr. Christopher Cadou
Dr. Inderjit Chopra

Acknowledgements

I would like to thank Dr. James Baeder for accepting me into his group and allowing me to conduct research I had not conceived of. I would also like to thank Dr. Christopher Cadou and Dr. Inderjit Chopra for taking the time to serve on my committee and assist with feedback along the way. Thank you to all the other members of Dr. Baeder's group as well as the other grad students in the cube farm during the fun times and those times we were buried in work. I would especially like to thank Taran, Bharath, and Bumseok for their direct contributions to the current research. Finally, thank you to the friends and family who supported me outside of the classroom and the lab.

Table of Contents

List of Tables

List of Figures

Nomenclature

a	Speed of sound
A	Area
$\tilde{A}$	Roe-averaged Jacobian matrix
c	Chord length
C_{do}	Profile drag coefficient
C_P	Coefficient of power
C_Q	Coefficient of torque
C_T	Coefficient of thrust
C_p	Specific heat of ideal gas at constant pressure
C_v	Specific heat of ideal gas at constant volume
d	Distance to nearest wall
e	Internal energy per unit mass
E	Total energy per unit volume
$\boldsymbol{F}_i, \boldsymbol{G}_i, \boldsymbol{H}_i$	Inviscid flux vectors
$\boldsymbol{F}_v, \boldsymbol{G}_v, \boldsymbol{H}_v$	Viscous flux vectors
$\boldsymbol{F}_L, \boldsymbol{F}_R$	Left and right states of a flux interface
j, k, l	Computational domain indices
L	Reference length
M	Mach number
M_{tip}	Mach number at the blade tip
n	Time step
p	Pressure
P_r	Prandtl number
q_x, q_y, q_z	Thermal conduction terms
$\boldsymbol{Q}$	Vector of conserved variables
R	Radius
R_e	Reynolds number
$Re_{\theta c}$	Critical Reynolds number
$\overline{Re}_\theta$	Reynolds number based on momentum thickness
$\overline{Re}_{\theta t}$	Reynolds number at transition onset
$\boldsymbol{S}$	Vector of body forces
t	Time
T	Thrust
Tu	Turbulence intensity
$T_\xi T_\xi^{-1}$	Matrices of left and right eigenvectors
u, v, w	Cartesian velocity components
v_i	Inflow velocity
V	magnitude of the velocity vector
x, y, z	Cartesian coordinate directions

Symbols

γ	Ratio of specific heats
λ	Eigenvalue
λ_i	Non-dimensionalized inflow ratio
Λ	Diagonal matrix of eigenvalues
μ	Coefficient of molecular viscosity
μ_t	Turbulent/Eddy viscosity
ν	Kinematic viscosity/ Intermittency
ξ, η, ζ	Computational coordinate directions
ρ	Density
σ	Rotor solidity
τ_{ij}	Viscous stress tensor for a Newtonian fluid
Ω	Angular velocity

Subscripts and Modifiers

$\bar{q}$	Mean value
q'	Fluctuating value
q_∞	Freestream flow value
q_o	Initial value
$\frac{\partial q}{\partial x}$	Partial derivative of q with respect to x
$q_i, q_j, q_k,$	Component of vector q along a computational domain index
$q_x, q_y, q_z,$	Component of vector q along a physical domain index
Δq	Change in value q between two adjacent indices

Abbreviations

AIAA	The American Institute of Aeronautics and Astronautics
AoA	Angle of Attack
APG	Adverse Pressure Gradient
BEMT	Blade Element Momentum Theory
BMTR	Basic Model Test Rig
BVI	Blade/Vortex Interactions
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
CRWENO	Compact-Reconstruction Weighted Essentially Non-Oscillatory
DADI	Diagonalized Alternate Direction Implicit
DDES	Delayed Detached Eddy Simulation
DNS	Direct Numerical Simulation
DoD	Department of Defense
FM	Figure of Merit
FVM	Free-Vortex Method
GT	Georgia Tech

IHC	Implicit Hole Cutting
LES	Large Eddy Simulation
LHS	Left Hand Side
LUSGS	Lower-Upper Symmetric Gauss-Seidel
MPI	Message Passing Interface
MUSCL	Monotone Upstream-centered Schemes for Conservation Laws
NASA	National Aeronautics and Space Administration
OGE	Out of Ground Effect
OVERTURNS	Overset Transonic Unsteady Reynolds-Averaged Navier-Stokes)
PDE	Partial Differential Equation
RANS	Reynolds-Averaged Navier-Stokes
RHS	Right Hand Side
RPM	Rotations Per Minute
STP	Standard Temperature and Pressure
TSDM	Transition and Stall Delay Model
UoT	University of Toledo
WENO	Weighted Essentially Non-Oscillatory

Chapter 1. Introduction

1.1   Motivation

The field of rotorcraft is constantly evolving and the next step in its development will necessitate improvements of current rotor designs. The rotor design affects numerous aspects of the flow field, including the behavior of the helicopter wake, a region formed by the tip vortices shed from the trailing edges of the rotor blades. These structures may encounter the advancing blade behind it, leading to unsteady airloads. Better understanding of these interactions will improve the analysis of rotor performance necessary to improve future rotor designs. Consistent and accurate prediction of hover performance has proven especially challenging, given the importance of the condition in the design of the final rotorcraft[1]. Computational fluid dynamics (CFD) provides an alternative to experimental testing, allowing quicker development and testing of potential rotor designs.

1.1.1 Overview of Rotor in Hover

Hover denotes a flight condition in which the helicopter is stationary, necessitating the balancing of the aerodynamic and gravitational forces acting on the vehicle. The hover condition presents a number of additional challenges when compared with a fixed wing aircraft in steady flight, namely the variation of velocity with rotor radius. In flow over a fixed wing the velocity is largely constant except at the edges of the wing; meanwhile the velocity changes over the rotor blades, rising from zero at the center of the hub to maximum value at the tip. Figure 1.1 illustrates this behavior while also showing that the velocity along the blade is azimuthally axisymmetric and varies linearly. Knowing that dynamic pressure is proportional to the square of the velocity, it results that the dynamic pressure is concentrated near the blade tips.
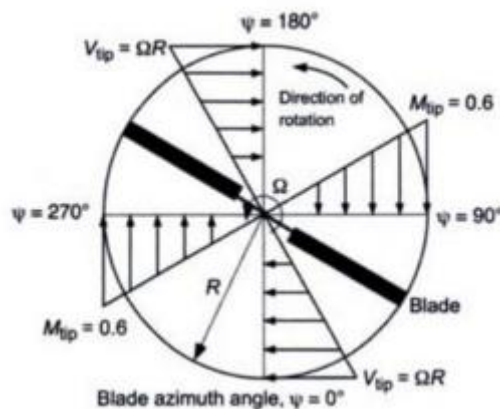
Figure 1.1 Radial velocity distribution over rotor in hover[2]

The high dynamic pressure near the blade tips leads to a concentration of aerodynamic forces in these regions. This imbalanced inflow, and the resulting aerodynamic forces, produces a strong wake in the tip region, creating tip vortices trailing from each blade. When analyzing a fixed wing, the wake will convect away from the surface and no longer interfere with the flow. However, the vortices produced in hover form a helical wake below the rotor, in which the trailing flow from one blade may alter the flow over the other surfaces. The slipstream velocity increases as the wake convects downward, causing contraction in accordance with the theory of conservation of fluid mass. Tracking these vortex structures has proven challenging due to the highly turbulent nature of the flow in the tip vortices. Therefore, models and experimental methods have been developed to predict these structures.

The rotor wake model describes the airflow coming off of the rotor during flight and includes a series of vortices shed by the blades. This flow is partly turbulent and therefore an exact understanding of vortices does not exist. However, sufficient modeling exists to allow the depiction of the flow coming off a blade in three main forms. These structures form primarily at 3 locations, the root of the blades, the blade tips, and the trailing edges of the blades. An example of a single blade vortex system can be found below in Figure 1.2. The vortex sheet emerging from the trailing edge of the blade is clearly visible along the length of the blade while the more localized tip vortex develops from the blade tip. Not shown here is the root vortex, which bears a close resemblance to the tip vortex.
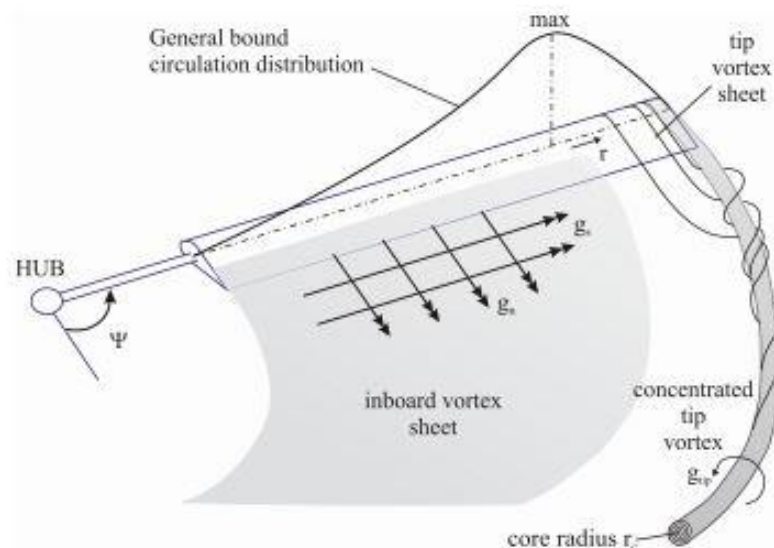
Figure 1.2 Model of vortex sheet and tip vortex shedding from a single blade[3]

In hover flight this wake is radially axisymmetric and may be visualized by successive views of a single blade at advancing azimuth angles. Figure 1.3 below demonstrates the behavior of the rotor vortex as a whole, highlighted by the tip vortices but with the vortex sheet visible just off of

the blades. The vortex sheet forms when the boundary layers over the upper and lower blade surfaces merge, creating a structure containing positive and negative vorticity. The tip vortices descend at the same rate, while contracting in accordance with the predicted behavior. These structures initially descend slowly but the rate increases past the first blade passage due to the downwash from the passing blade. Finally, notice in Figure 1.3 below that the separate tip vortices merge into a single vortex after a few blade passages.
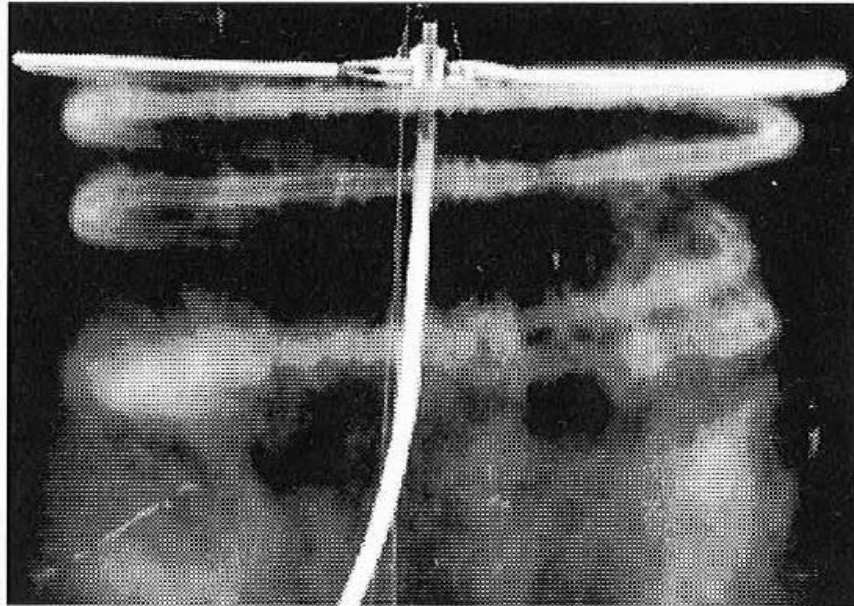
Figure 1.3 Merging of tip vortices from two-bladed rotor in hover[4]

Hover presents an especially critical case, as it is often one of the defining parameters of helicopter design and performance. Even a small amount of inaccuracy in estimating the hover performance can drastically reduce the performance or payload of the final vehicle. Therefore, a great deal of effort has been put into modeling and capturing the wake structure to ensure accurate prediction[4]. Accurate experimental capturing of the wake has proven challenging, requiring large facilities to minimize recirculation in the test chamber, while outdoor testing presents an uncontrolled environment where winds may disrupt the test. For example, work by Shinoda and Johnson demonstrates that testing results for the S-76 rotor have varied between testing facilities, including the NASA-Ames 80' x 120' wind tunnel[5]. This uncertainty necessitates large power margins in the design process to ensure sufficient power in the final vehicle. An overview of available experimental methods for rotorcraft is presented in the following section.

1.1.2 Overview of Experimental Methods

Leishman and Bagai have described some of the challenges facing analysis of helicopter performance in hover, along with the disturbances which can take place[6]. As illustrated below in Figure 1.4, a helicopter in hover sees an axial flow through the rotor, leading to a relatively uniform helical structure formed by the tip vortices. The vortices are visible for about two complete rotor revolutions, or 720° and the contraction of the wake is visible as the flow continues downward from the rotor plane. Figure 1.4 demonstrates the ability to observe the rotor wake using natural condensation in the air, one of the numerous ways to experimentally investigate the dynamics of vortices. This is but one of the methods available to obtain quantitative results concerning rotor wakes.

Knowing the importance of the rotor wake in helicopter analysis, one must understand the experimental methods available to capture and analyze it. Among the most notable techniques applicable to a helicopter in hover are the smoke and light sheet[7-10], natural condensation[11,12], schlieren[13,14], and shadowgraphy[15-20]. Each of these methods provides some level of visualization of the rotor wake, though not all provide readily available quantitative data.

The smoke and light sheet method injects a white smoke into the rotor wake and then illuminates a plane of the flow, allowing for an accurate photograph of the flow structure. Figure 1.4 below demonstrates the method in use on a rotor in hover, with three distinct vortices rendered in the smoke. This process requires a very strong light produced quickly and is quite time consuming in forward flight, though it does allow precision mapping of vortex structures. The method does produce clear images of vortex cores, as the centrifugal forces in the vortex core scatter the smoke particles, leaving a void within the flow.

The smoke particles must be large enough to appear clearly in the photos while remaining small enough to follow the flow. Thus particle size may interfere with the results of the test, for example the apparent vortex core size may varies with particle size due to the centrifugal force produced within the vortex core[6]. Furthermore, as seen in Figure 1.4, the smoke particles diffuse as the vortex ages, leading to decreased resolution of the vortices. Due to the dispersion of smoke particles and setup required, smoke flow visualization has been used primarily in subscale cases, only rarely being applied to full-scale rotors[2]. In addition to the smoke and light sheet method, natural condensation provides flow visualization of the rotor wake.

Figure 1.4 Flow visualization of a rotor plane using the smoke and light sheet method[6]

Figure 1.5 Flow visualization of tip vortices in hover due to natural condensation[6]

Natural condensation of water vapor allows observers to track the tip vortex location generated by the rotor blades[18]. As seen in Figure 1.5 above, the results are similar to those generated by the smoke and light sheet method. While only occurring under certain atmospheric conditions, usually in outdoor testing, this has the advantage of not introducing any foreign particulates into the test. Unfortunately, the quantitative results on tip vortex locations have proven difficult to measure, as the water vapor dissipates similarly to the smoke particles in the previous method[6]. Therefore visualization through natural condensation does not lend itself to precise measurements, but provides an unmodified view of the rotor wake. The last experimental methods of measuring rotor wake to be discussed here are the density gradient methods, including shadowgraphy and schlieren.

Both shadowgraphy and schlieren rely on a large density inhomogeneity in the flow, distorting the light in regions of high or low density[6]. However, not all vortical fields contain sufficient density gradients to allow for proper imaging. This is primarily an issue with subscale rotors, which must operate at a close to full-scale tip speed and relatively high thrust to generate vortices strong enough to capture. Figure 1.6.a below shows a shadowgraph of a curved vortex, with the view nearly parallel to the vortex axis. The bright ring surrounding the nucleus denotes the edge of the vortex as seen on the shadowgraph. Knowing the refractive index of the fluid, one can calculate the actual size of the vortex core based off of the size of the core within the shadowgraph. Figure 1.6.b shows a separate set of results, demonstrating the ability to track vortex positions through the use of shadowgraphy. One can even see the upward convection of the tip vortex at the front of the rotor, denoting an area in which blade/vortex interactions (BVI) are present. While such precise results are difficult to obtain, they demonstrate the capabilities of density imaging methods.
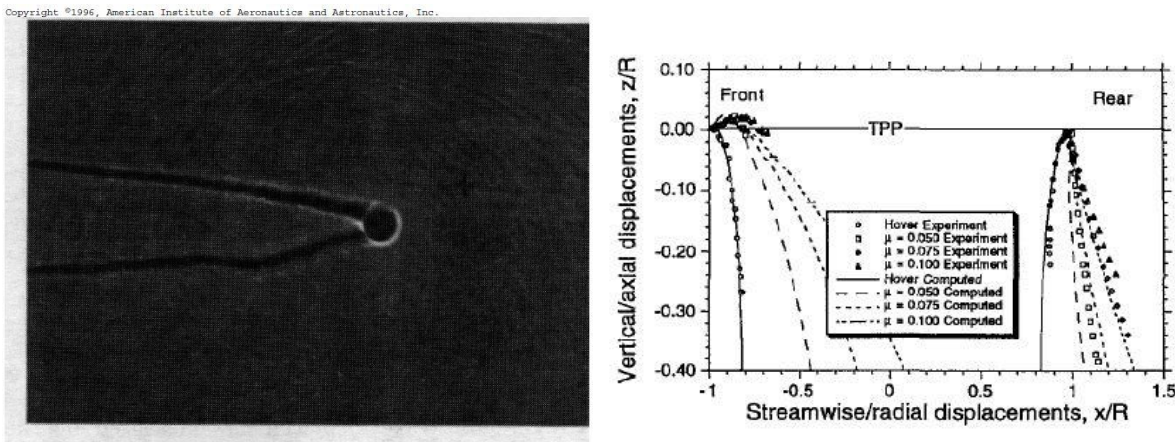
(a) Flow visualization of a tip vortex using shadowgraph technique

(b) Wake displacement positions obtained using shadowgraph technique

Figure 1.6 Results from Leishman and Bagai using shadowgraph techniques[6]

1.1.3 Overview of Common Mathematical Models

In addition to the experimental methods discussed in the previous section, there exist mathematical models which are used to predict performance and wake structure in various flow regimes. These methods can be split into two broad categories, those used to predict performance and those used to model the rotor wake.

Performance models allow for the quantification of thrust and power values for a given rotor and flight condition, enabling analysts to predict certain parameters on which to base a more detailed design. The two primary performance models in modern use are the Rankine-Froude momentum theory and the Blade Element Momentum Theory proposed by Gustafson and Gessow[2]. Momentum theory entails the derivation of a first-order model of the rotor thrust and power values using a quasi-one-dimensional integration of the control volume enclosing the rotor and its wake. Figure 1.7 below illustrates the flow around a rotor in hover as predicted by momentum theory, traveling through the plane of the rotor disk and contracting and speeding up as the air convects downward. Here the rotor is treated as an actuator disk, a surface over which there is a pressure difference. By making a few assumptions about the flow, namely that it is one-dimensional, quasi-steady, incompressible, and inviscid, one can employ the principles of conservation of momentum and energy to calculate the velocity at the rotor-disk and the far wake.

Further parameters such as induced velocity, disk loading, power required to hover, and coefficients of thrust and power may be calculated through manipulations to the initial conservation equations and knowledge of the flow. An induced power correction factor, κ is used to represent a number of physical effects within the flow which are not accounted for in the conservation equations. This correction factor is based on experimental results taken by various manufacturers and may vary depending on the source used but serves to more accurately represent the flight conditions. While this method provides basic performance aspects, it is limited and provides no information as to the blade loading or design of the rotor itself[2].
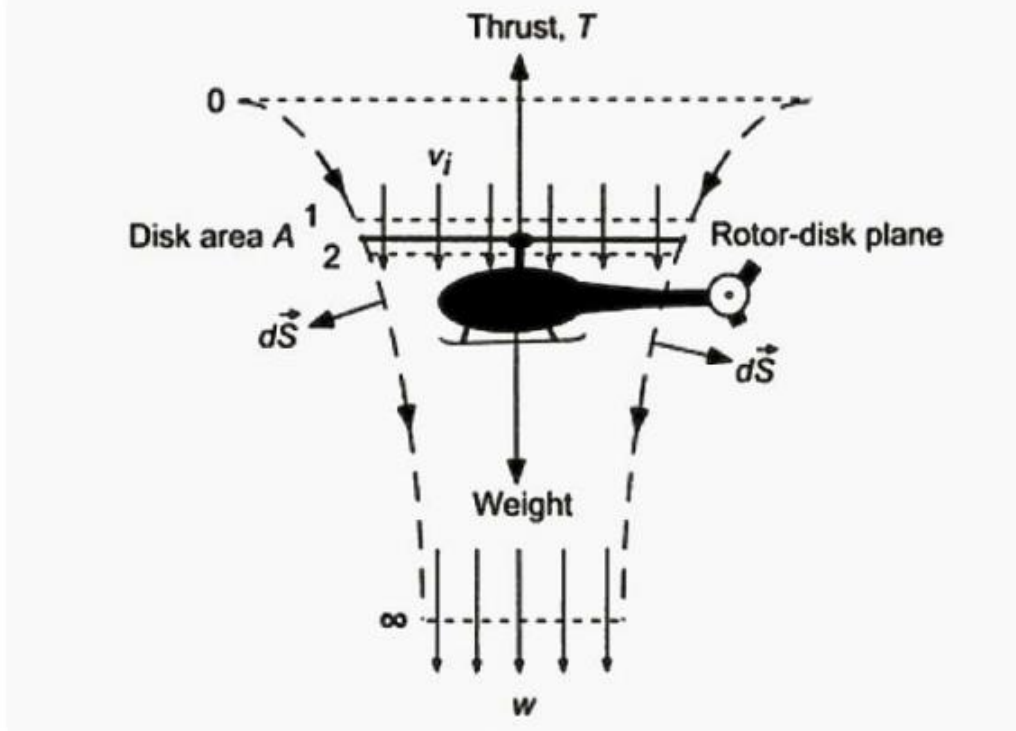
Figure 1.7 Flow model as predicted by momentum theory in hovering flight[2]

In addition to the wake structure, the other primary values for analyzing rotor performance in hover are the integrated thrust and power coefficients. Momentum theory gives the ideal values of the coefficient of thrust (C_T) and (C_P) as follows,

$$C_T = \frac{T}{\rho A \Omega^2 R^2} \tag{1.1}$$

$$C_P = \left(\frac{T}{\rho A \Omega^2 R^2}\right)\left(\frac{v_i}{\Omega R}\right) = C_T \lambda_i = \frac{C_T^{3/2}}{\sqrt{2}} \tag{1.2}$$

where

$$\lambda_i = \frac{v_i}{\Omega R}\sqrt{\frac{T}{2\rho A \Omega^2 R^2}} = \sqrt{\frac{C_T}{2}} \tag{1.3}$$

in which T is thrust, ρ is density, A is the rotor disk area, ΩR is the blade tip speed, v_i is the inflow velocity, and λ_i is the non-dimensionalized inflow ratio. These parameters are useful as non-dimensional values of thrust and power, ensuring that a wide range of cases may be easily compared with one another, independent of individual conditions and testing configurations. The above values represent idealized values and the computational coefficients of thrust and power values are calculated using the flow field values as discussed later in the current work. The final commonly used parameter when discussing helicopter performance is Figure of Merit (*FM*).

Figure of Merit denotes the ratio of the ideal power required to hover to the actual power required to hover. *FM* was introduced in the 1940s as a standard, non-dimensional measurement of hover efficiency[2]. Due to non-ideal effects such as tip loss, turbulence, and viscous losses the *FM* will always be less than one. By taking non-ideal effects such as induced power and profile power into account the *FM* can be written as follows,

$$FM = \frac{C_T^{3/2}}{\frac{\kappa C_T^{3/2}}{\sqrt{2}} + \frac{\sigma C_{do}}{8}}$$

(1.4)

where κ is the induced power factor, σ is the rotor solidity, and C_{do} is the profile drag coefficient. Here the induced power factor is a value accounting for numerous non-ideal effects in the flow, derived from experimental results and rotor measurements. Solidity represents the ratio of the blade area to the rotor disk area and the profile drag coefficient accounts for the drag over the blade.

Figure of Merit, along with the coefficients of thrust and power, provide non-dimensional values to compare the efficiency and performance of multiple rotorcraft. By performing collective sweeps for a range of test conditions, one can quickly compare these integrated values to determine relative performance. The current work employs these values in analyzing the hover condition, as presented later in this report.

Blade element momentum theory (BEMT) builds on the momentum theory presented above, analyzing the radial and azimuthal distribution of aerodynamic loading over the rotor. This is accomplished by treating each blade as a series of quasi-2D airfoils, integrating spanwise forces and moments to obtain the thrust and power estimates for the rotor. BEMT analyzes the inflow as it varies over the blade, accounting for such effects as tip loss, airfoil shape, blade twist, blade taper, and number of blades in the rotor. The effects of other blades and the vortical wake are included through an induced Angle of Attack (AoA) computed using the Biot-Savart law[2]. Numerical solutions employing these methods are accurate across a range of flight conditions and rotor designs. While BEMT does not provide a complete picture of the complicated flow caused by the tip vortices of the rotor, prescribed- and free-vortex methods (FVM) have been developed to provide more accurate predictions, though with a greater numerical cost.

Vortex wake models explicitly track the convection of tip vortices relative to the rotor while treating the convection and diffusion of these structures separately. There exists a number of wake models but all employ the Biot-Savart Law[2] to calculate the induced velocity at any point in the flow caused by a vortex element. Figure 1.8 below shows a number of representations of the vortex wake behind a rotor blade[2] illustrating the trailed and shed circulation behind a single blade of the rotor. Figure 1.8.a demonstrates a straight-line element model composed of trailed and shed

9

circulation while 1.8.b shows a vortex lattice generating a roll-up vortex. Figure 1.8.c traces lines of constant strength vorticity and 1.8.d uses points to denote paths of trailed and shed circulation. These methods require many individual elements to model the wake, usually on the order of thousands of points or segments depending on the method. In comparison to finite-difference and finite-volume CFD methods the vortex models are far less computationally expensive but require varying degrees of assumptions depending on the type of model used.
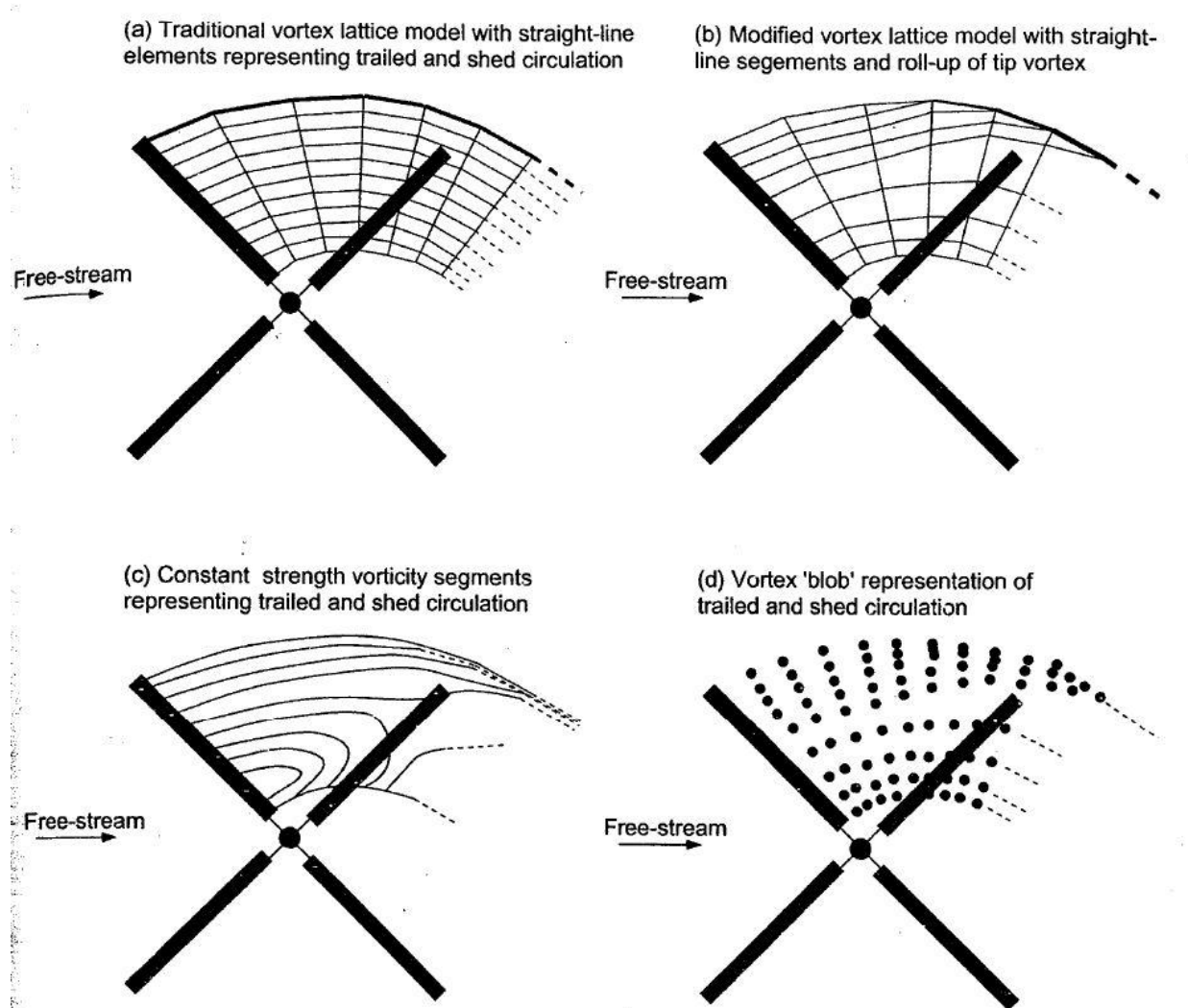
Figure 1.8 Visualizations of the vortex wake generated by a rotor blade[2]

1.1.4 Overview of CFD Methods

Computational Fluid Dynamics (CFD) methods employ first principles to simulate the entire flow within the rotor plane and wake, including compressibility effects, unsteady flow, separation, and other real-world issues which models do not capture. Older works employed simpler methods, such as irrotational, inviscid flow approximations but most modern solvers evaluate the full

Navier-Stokes equations[2]. Because the CFD method makes fewer assumptions about the flow it can generate more accurate results for any number of designs. However, this precision comes with the cost of massive computational requirements. A single run may use dozens or hundreds of processors within a cluster with a runtime of multiple days or even weeks. Modern CFD solutions can be run with tens or hundreds of millions of mesh points, allowing sufficient numerical resolution to capture the entirety of the flow.

There are two primary types of CFD models, those employing a finite-difference method and those employing a finite-volume method. Finite-difference methods estimate spatial derivatives using equations based on Taylor series approximations evaluated at each point within the computational grid. Meanwhile, finite-volume methods solve an integral form of the Navier-Stokes equations and operate at each cell rather than at nodes in the grid. The integral values of mass, momentum, and energy of each cell is divided by the cell volume to calculate average conserved variables and fluxes over cell boundaries. Both methods will produce the same results when properly implemented but finite-difference methods are restricted to use on structured meshes while finite-volume methods may operate across structured or unstructured meshes[2].

1.2   Literature Review

The current work is aimed at employing CFD in predicting the performance of the S-76 rotor in hover, as part of the concerted effort of the AIAA Invited Hover Session to evaluate and improve CFD methodology. To ensure each participant had the same baseline results for comparison, a test rotor based on the S-76 rotor as evaluated in the "Experimental Study of Main Rotor/Tail Rotor/Airframe Interactions in Hover" report composed by Balch et. al[21] was chosen as the baseline experimental case. Finally, results from a number of organizations have been presented at previous Scitech conventions, of which two were chosen to be reviewed here as an overview of the current state of various CFD solvers.

1.2.1 Invited Hover Session

In an effort to ensure accurate prediction of rotor capability the AIAA Applied Aerodynamics Technical Committee Rotorcraft Simulation Working Group[1] was formed. The stated goals of the committee are to coordinate participants from DoD, NASA, Industry, and Academia to evaluate the state of CFD across the group, determine future challenges, and direct the development of new capabilities. Due to the existence of workshops focusing on the forward flight condition, this working group elected to focus on the hover condition. To ensure accurate comparisons the S-76 benchmark case was chosen, as defined by the test rotor work undertaken by Balch et. al[21].

1.2.2 Experimental Study

Balch et. al[21] performed experiments on the S-76 test rotor which the Working Group uses as the baseline integrated thrust and power values results. The testing was performed using a four-bladed rotor in hover, a 1/4.71 scale model of the S-76 rotor. These test blades had a radius of 55.0 inches and chord of 3.1 inches with a linear 10 degree twist. A number of blade tips were tested, including a swept-tapered, straight rectangular, and swept-tapered with anhedral. Apart from this outer 5%, the blades tested were identical. The airfoil sections used were the SC1013-Rd, SC1095R8, and SC1095. The test operated at a tip Mach number of 0.65 and chord Reynolds number of 1.16×10^6. A range of collective pitches varying from 0° to 15° were experimented upon. The experiment focused primarily on measuring integrated loads, so other values such as distributed forces and tip vortex position are compared only to other calculations undertaken as part of the Working Group.

The Basic Model Test Rig (BMTR) was designed as a self-contained system which could test a range of rotor and fuselage systems under varying conditions. This rig was integrated into the Sikorsky model hover test facility as illustrated in Figure 1.9 below. The bay in which the experiments were conducted minimized wind conditions and ensured the quality of the acquired data regardless of ambient conditions. Thus, the authors were able to extract accurate measurements of the desired rotor forces, moments, and torque values for the scale rotor under specified control inputs. The S-76 blades employed in the study had a scale of 1/5 of the full rotor blade, while maintaining the linear twist, solidity, airfoils, and tip shapes of the original rotor. The out of ground effect (OGE) tests of the main rotor resulted in thrust, power, and FM trends which matched those of the full-scale equivalent. Figure 1.10 below presents the results acquired in the isolated rotor testing of the S-76.
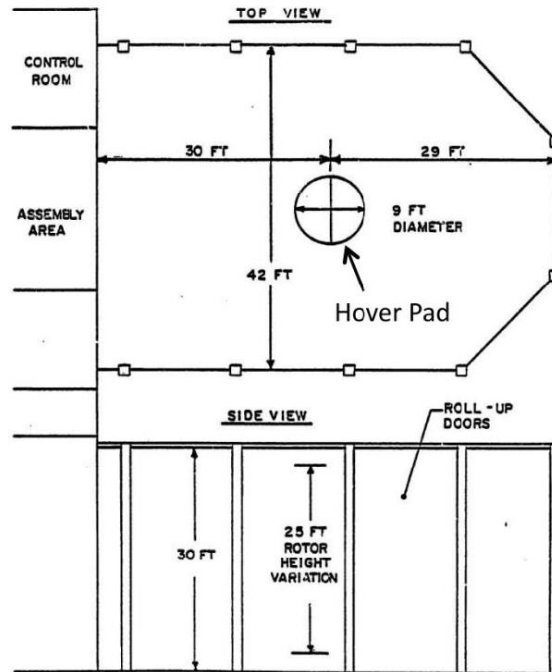
Figure 1.9 Schematic of test stand employed in the S-76 rotor tests by Balch et. al[2]
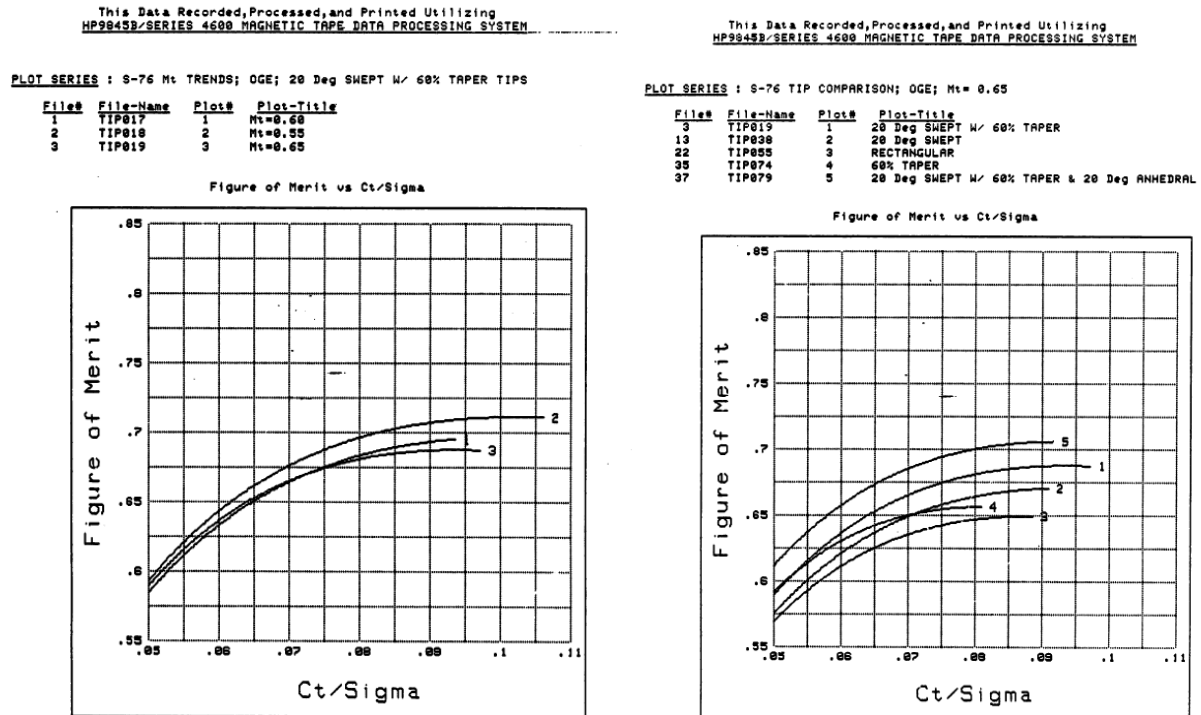
(a) Comparison of tip Mach numbers

(b) Comparison of tip shapes at M = 0.65

Figure 1.10 Integrated forces results attained by Balch et. al when testing the isolated S-76 rotor OGE[2]

Balch et. al conducted two sets of tests on the isolated main rotor of the S-76 in hover, OGE conditions. Figure 1.10.a presents the results of the sweep of tip Mach numbers while 1.10.b is the sweep of tip shapes for the S-76 blades. The sweep of tip Mach numbers above was conducted using the swept-tapered baseline blade for three values, M_{tip} = 0.55, 0.60, and 0.65. Meanwhile five tip shapes were evaluated, including the three which were ultimately chosen by the Invited Hover Session for testing using computational methods. The results presented in Figure 1.10 were digitized and used as comparison points for the CFD studies seen later in the current work. The work of Balch et. al discussed in the above section created a foundation for the computational studies undertaken by the current author and other institutes.

1.2.3 Previous CFD Results

This work is the continuation of the Invited Hover Session efforts and therefore it is important to review the current state of CFD solvers used to evaluate the S-76 rotor.

1.2.3.1. Assessment of Planform Effects on Rotor Hover Performance (GT)

Researchers from Georgia Tech[22], in conjunction with figures from Industry, have employed a GT-Hybrid solver in their evaluation of the S-76 rotor case. This method evaluates the full Navier-Stokes calculations near the blades, combined with a vortex wake model in the far-field domain. This represents a hybrid approach, ensuring full modeling of the viscous flow near the blades while modeling the wake in the more stable region away from the blade. This method shows good agreement with the integrated values of thrust and power taken from experimental results. That is due to the relative independence of these values from the wake behavior. However, the differences in tip vortex descent and contraction is more pronounced, as this method is not fully simulating the flow in those regions but instead predicting the flow using the vortex wake model.

Figure 1.11 below illustrates the computational domain employed in this work. The near-body domain solved the full Navier-Stokes equations while away from the rotor the wake was captured using a Lagrangean approach. From this captured region, the full wake was modeled, enabling the authors to avoid diffusion of the wake normally seen in numerical solvers. The effect of the vortices on trailing blades is computed based on the vortex-induced velocities at the far field boundary of the computational grid region, neglecting contributions captured within the CFD region itself.

14
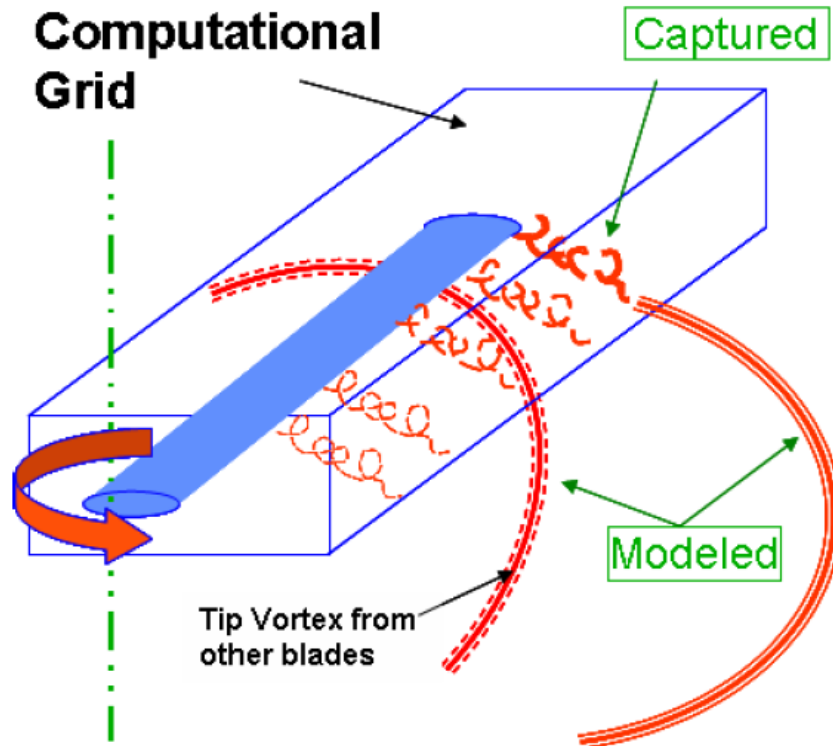
Figure 1.11 Computational domain employed with Georgia Tech's GT-Hybrid solver[22]

Figure 1.12 presents the integrated forces results attained by the authors, including comparisons to the experimental results and other CFD solvers. They showed good agreement with the thrust values while over predicting torque coefficient by a slight amount. Thus the torque coefficient for a given thrust value is over estimated, leading to an under prediction of the Figure of Merit. Figure 1.12.c indicates that GT-Hybrid is not the only CFD solver to under predict the FM, with other sources showing similar values.
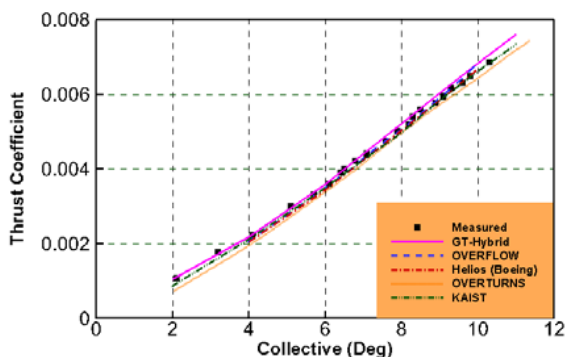
(a) Thrust coefficient vs. pitch
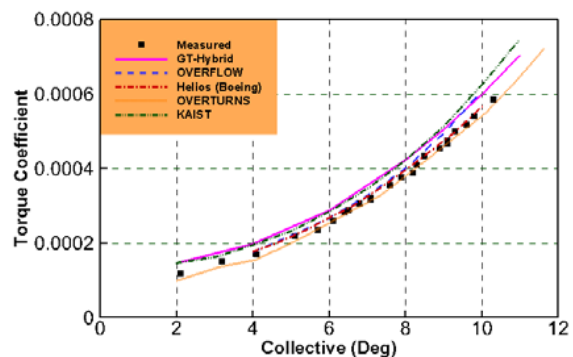
(b) Power coefficient vs. pitch
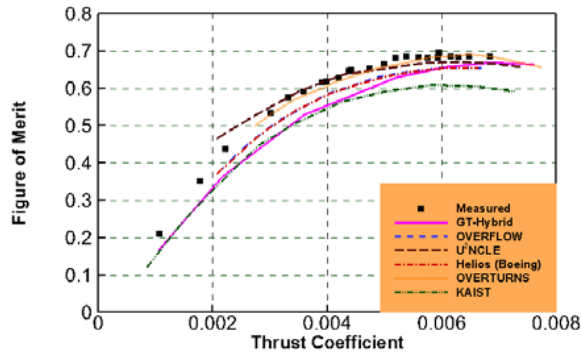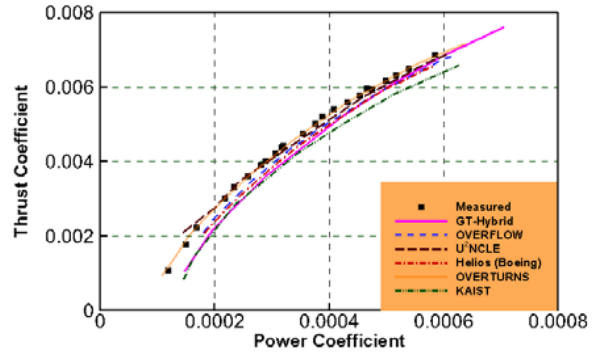
(c) Figure of Merit vs. thrust coefficient   (d) Thrust coefficient vs. power coefficient

Figure 1.12 Baseline blade forces results for a number of solvers, including Georgia Tech's GT-Hybrid solver[22]

The ideal CFD solution would produce a FM value within 1% of predicted values across all collective pitches. Current results show close agreement with predicted thrust coefficients but a larger spread of predicted power coefficients. This results from the difficulty in accurately predicting and modeling the transition and turbulent behavior near the blade surface. While the thrust is largely independent of these factors, the predicted power depends heavily on the accurate modeling of this behavior. These results present not only the potential of CFD solvers for predicting performance but also the need to further develop current solvers.

Figure 1.13 below shows the tracking of the tip vortex descent and contraction as captured by a number of solvers, including GT-Hybrid. Tip vortex trajectory influences the inflow seen across trailing blades, making the capture of these vortices essential to accurate hover performance analysis. The free-vortex method (FVM) used here achieves good correlation with full capture methods for the first rotor revolution, or 360 degrees. Past this the methods begin to diverge, and the FVM significantly under predicts the contraction rate of the vortex. Despite this, the method accurately captures the trends seen by other works while employing the less expensive Lagrangean methods avoid full modeling of the rotor wake.
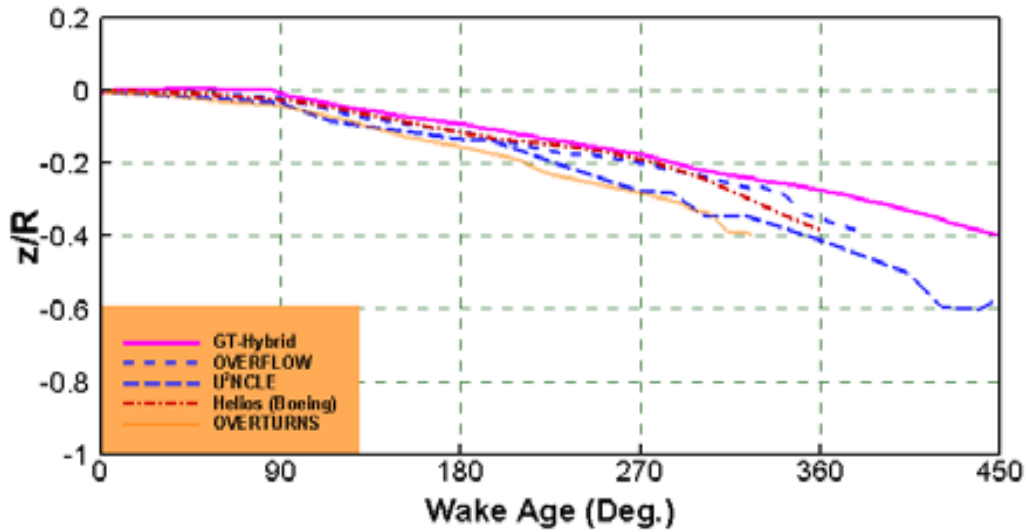
(a) Tip Vortex Descent Rate
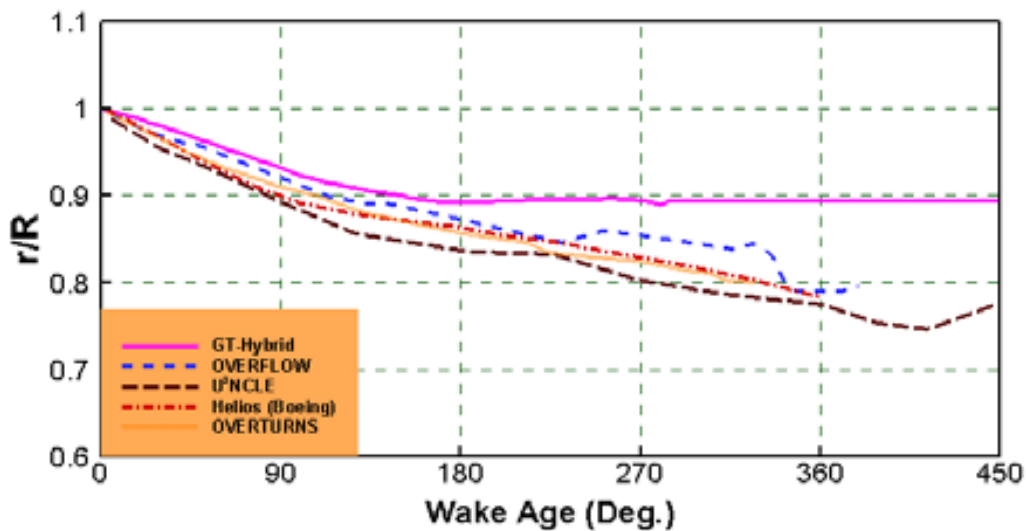
(b) Tip Vortex Contraction Rate

Figure 1.13 Tip vortex descent and contraction rates for baseline blade at trim condition[22]

1.2.3.2. S-76 Rotor Hover Predictions Using Advanced Turbulence Models (UoT)

The University of Toledo[23] has employed the unstructured CFD solver U^{2}NCLE to analyze the S-76 rotor. They employ a single domain, composed of an unstructured mesh which models a single blade as well as a model hub. Using periodic boundary conditions, the effects of the remaining three blades are modeled based on the results simulated within the computational domain. The authors tested various levels of mesh refinement, altering the resolution near the leading and trailing edges of the blade as well as near the tip region. Furthermore, the work involved testing a number of transition models, the process which governs the transition from

17

laminar to turbulent flow within the simulation. Studying a number of transition models, the participants developed a Transition and Stall Delay Model[23] (TSDM) which increased turbulence production in regions of separated flow, striking a balance between the point of flow separation while capturing the transition region at the blade.
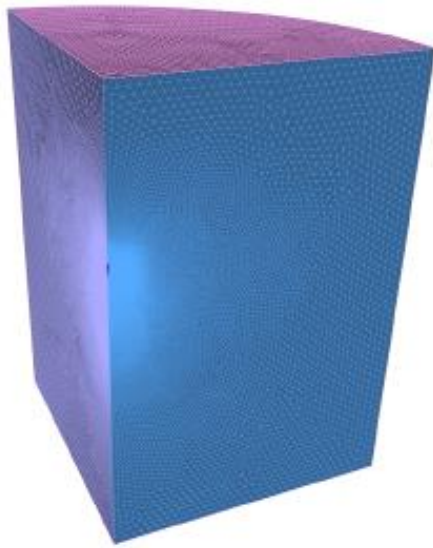
(a) Computational domain of one blade (b) Spanwise slice of unstructured grid

Figure 1.14 Computational domain and unstructured mesh employed with the U^{2}NCLE solver[23]

Figure 1.14 above displays the unstructured volume grid created for the S-76 rotor testing. Employing periodic boundary conditions, the authors were able to simulate a single blade, rather than the full 4-bladed domain. The mesh was refined such that the largest spacing near the blade was 1 to 2% of the tip chord, ensuring accurate prediction of the near-body flow. Employing the grid seen above, the authors were able to generate accurate results for sweep of hover cases as seen below in Figure 1.15.
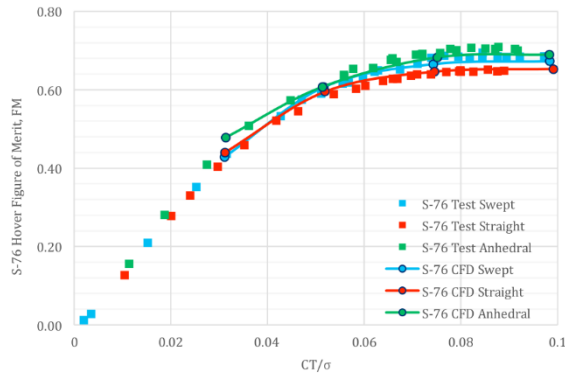
(a) Figure of Merit vs. thrust coefficient
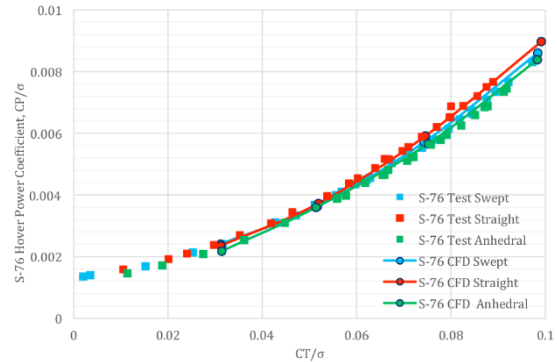
(b) Thrust coefficient vs. power coefficient

Figure 1.15 Sweep of collective pitch angles for a range of tip shapes conducted using the U^{2}NCLE solver[23]

Using their single bladed mesh, the authors were able to capture accurate thrust and power values across a range of pitch angles and blade tips. Furthermore, they captured the relative performance of each tip shape, showing that the anhedral shape required the least power while the straight tip required the most at the tested tip Mach number of 0.65. Additionally, these tests captured the effect of the tip vortices on rotor performance. The swept-tapered and anhedral tips both showed improved performance at higher thrust values as a result of the altered tip vortex strength. Figure 1.16 below illustrates this point, comparing the tip vortex strength, size, and location for each tip.
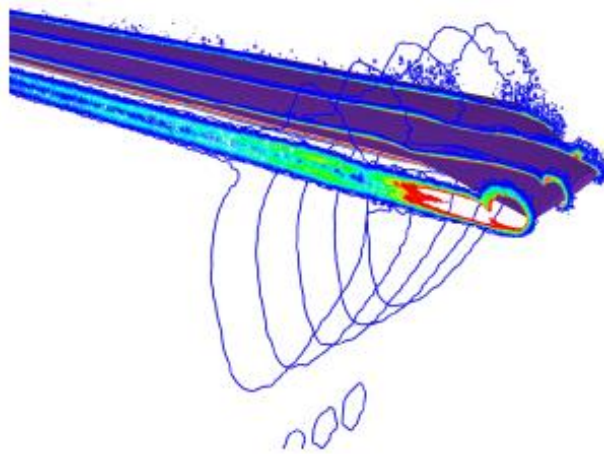
(a) Swept-tapered tip
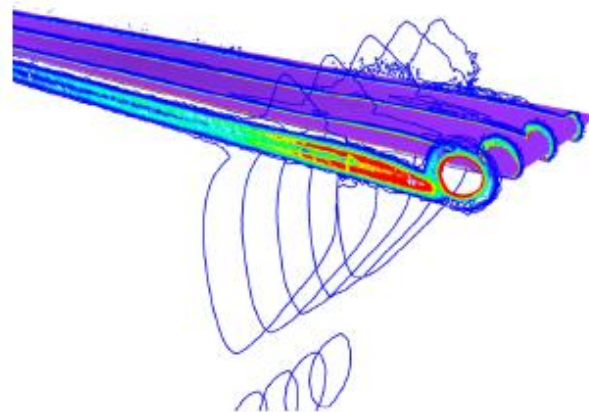
(b) Straight rectangular tip

19
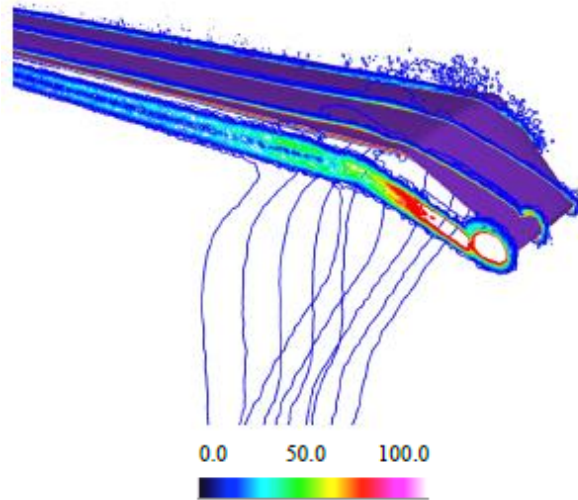
(c) Swept-tapered tip with anhedral of 20°

Figure 1.16 Tip vortex strengths across three tip shapes, as predicted with the U^{2}NCLE solver[23]

In addition to the integrated forces and tip vortices, the University of Toledo studied the spanwise sectional thrust distribution and chordwise pressure distributions for the three blade tips at a trimmed angle of $C_T/\sigma = 0.09$ with a tip Mach number of 0.65. Figure 1.17 below shows the coefficient of thrust distribution measured by U^{2}NCLE, normalized by the local velocity. Figure 1.18 shows the corresponding coefficient of torque distribution. It can be seen that the tip shape has some effect on the thrust distribution due to the different tip vortices produced and their subsequent effect on the flow. The sectional torque sees a smaller difference between the tip shapes, with the swept and anhedral results being nearly identical while the rectangular tip sees a slightly larger peak near the tip.
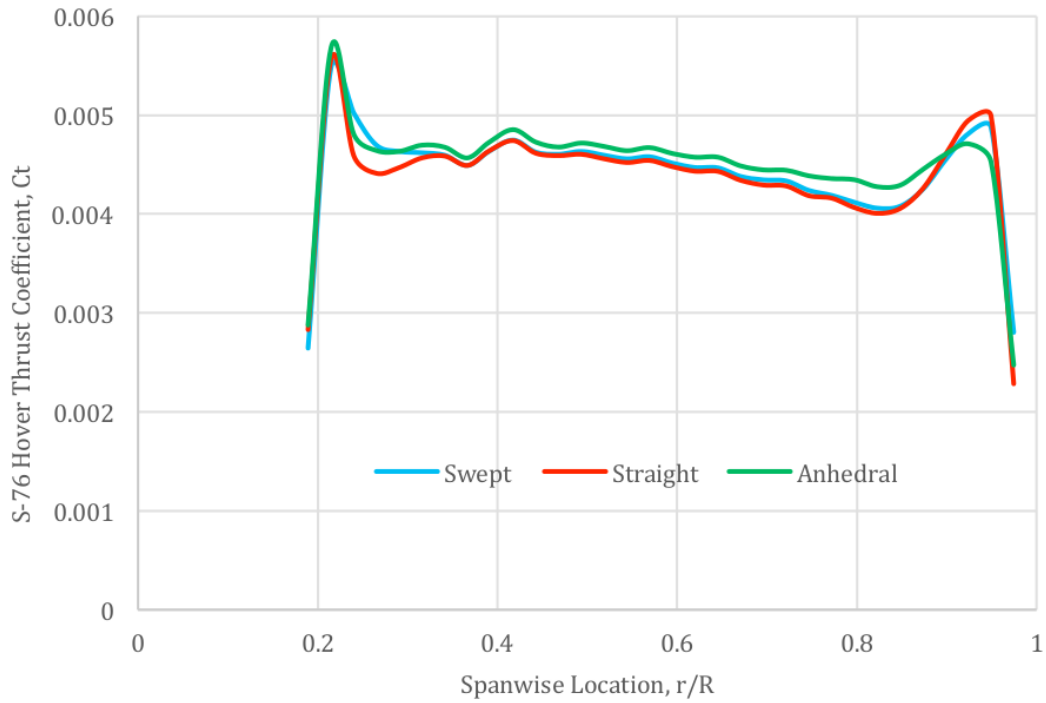
Figure 1.17 Sectional thrust coefficient at trim condition, as predicted by U^2NCLE^{23}
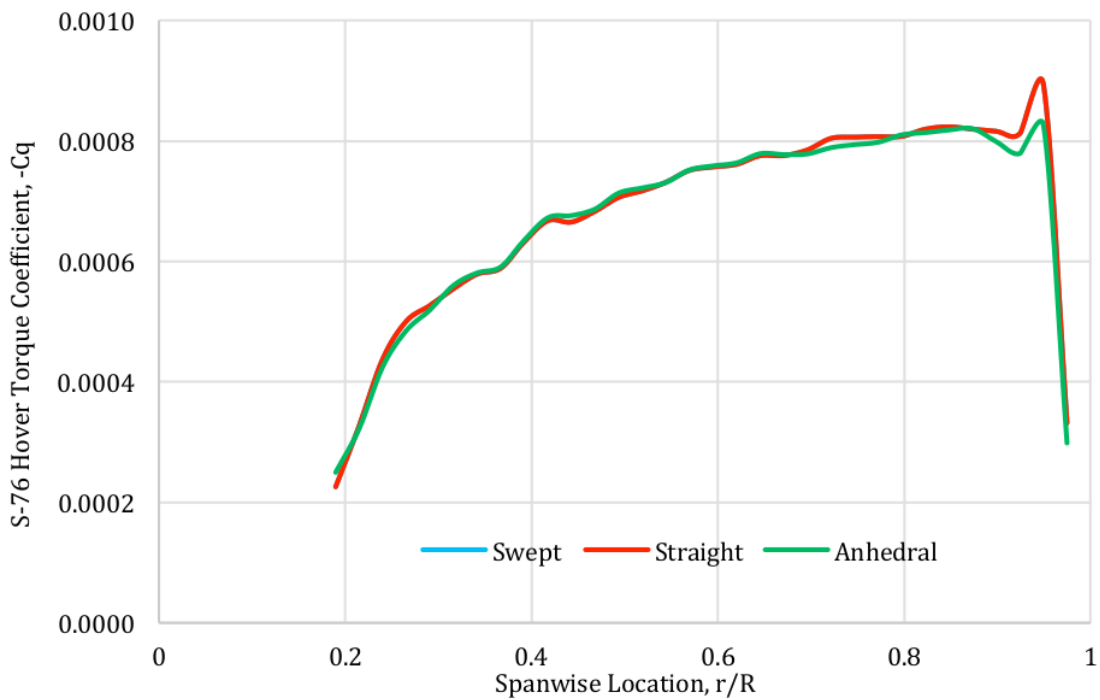
Figure 1.18 Sectional torque coefficient at trim condition, as predicted by U^2NCLE^{23}

Meanwhile, Figure 1.19 shows the pressure coefficient distribution over the chord at a number of spanwise locations. There is a large difference in performance at $r/R = 0.95$, where the tip shapes diverge, and elsewhere there is a reduced yet noticeable impact on the pressure distribution.

(a) r/R = 0.925

(b) r/R = 0.95

(c) r/R = 0.975

(d) r/R = 0.99
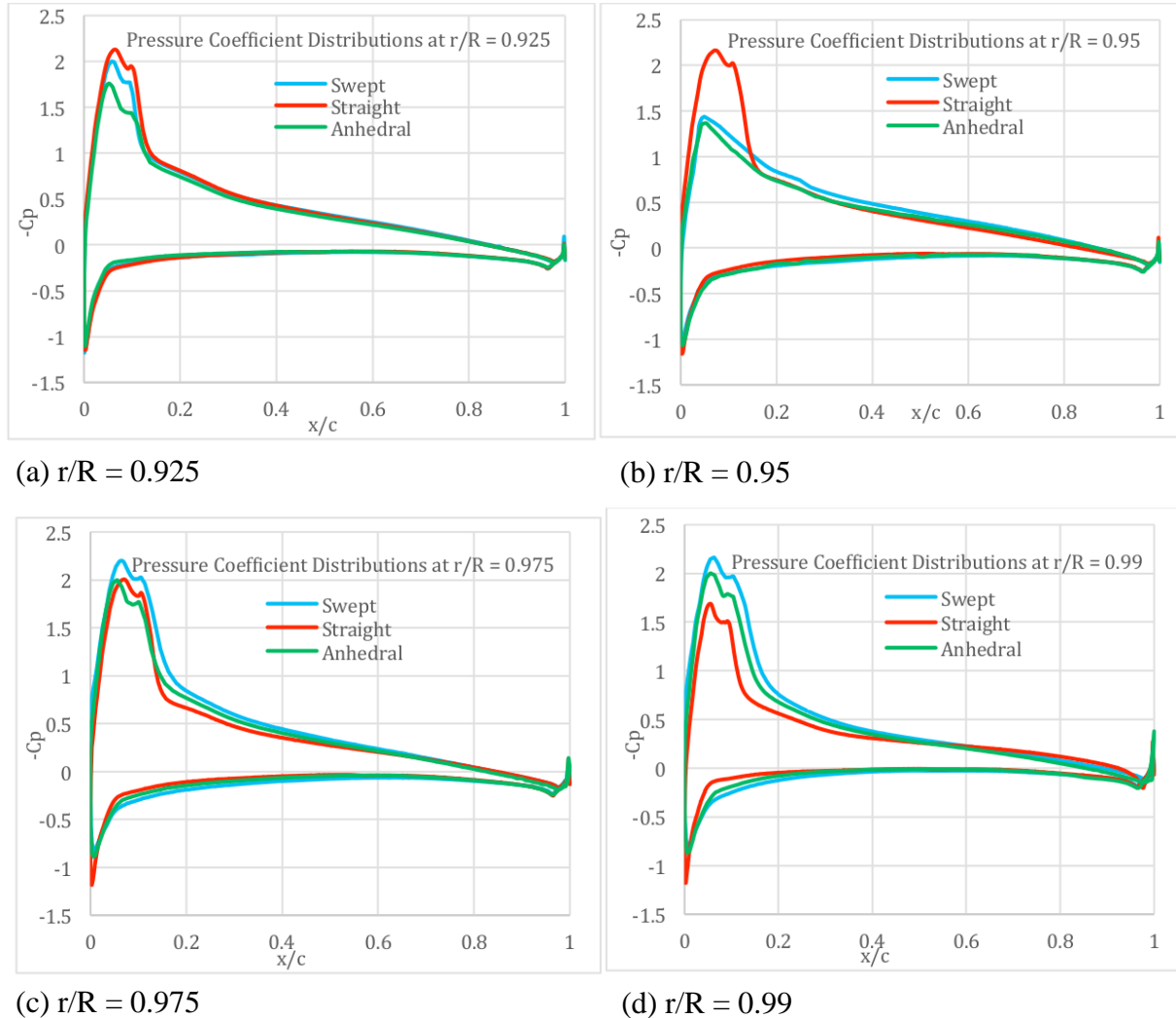
Figure 1.19 Chordwise pressure coefficient distribution at a range of spanwise locations at trim condition, as predicted by U^{2}NCLE[23]

Finally, the University of Toledo also studied the tip vortex trajectories generated by each blade type at the same trim and tip Mach number conditions seen above. Figure 1.20 below shows the vortex descent and contraction paths for all three tip shapes. The contraction rates start to deviate after half a revolution, with the straight tip shape ultimately contracting more slowly than the other two shapes. Meanwhile, the descent rates remain close until ¾ a rotor revolution, at which point the straight tip again diverges and settles at a lower descent speed than the other tips. Overall, the use of the U^{2}NCLE solver on a single blade mesh produced accurate results and demonstrated the viability of periodic boundary conditions as well as the need for sufficiently refined meshes.
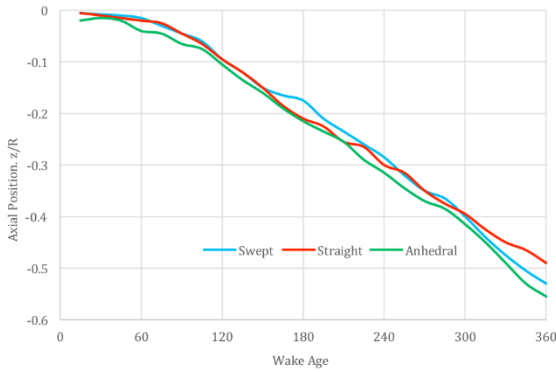
(a) Descent trajectories (b) Contraction trajectories
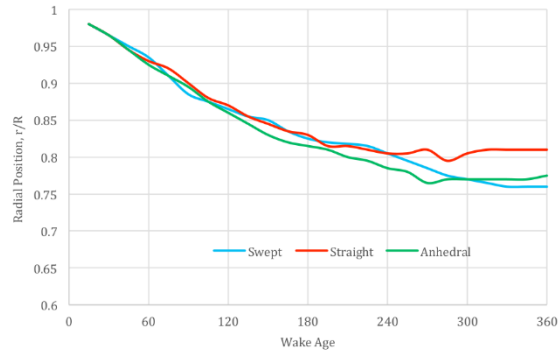
Figure 1.20 Tip vortex trajectories in both axial and radial directions at trim condition, as predicted by U^{2}NCLE[23]

1.3 Goals and Methods of the Current Work

1.3.1 Goals

The primary goals of the current work are as follows:

1. Maintain accurate prediction of performance of the S-76 rotor in hover.

2. Successfully incorporate mesh adaption to the current work to improve resolution without additional computational expense to the solver.

3. Better resolve the rotor wake for multiple blade passages out to convergence of the solution.

1.3.2 Methods

The stated goals have been pursued through the following approach:

1. Generation of a new computational domain on which to simulate the test case.

2. Testing of multiple solver configurations on a single set of conditions to compare performance and accuracy.

3. Following the conclusion of initial testing, a full sweep of collective pitch angles was conducted to determine accuracy across a range of conditions.

4. Finally further analysis of a single case was conducted to review the more detailed aspects of the flow as predicted by the solver.

1.4   Thesis Contributions

The advancements contributed by this research are as follows:

1. Development of a new blade mesh utilizing the O-O mesh design and successful generation of a usable mesh based on the S-76 rotor blade.

2. Addition of clustering and adaption routines to the background mesh in an effort to better capture the tip vortices of the blade.

3. Development of a periodic boundary applicable to a Cartesian background mesh, which allows reduction of the computational domain.

4. Implementation of the CRWENO spatial scheme to this version of OVERTURNS, to ensure higher spatial accuracy than previously attainable.

1.5   Scope of Thesis

This thesis focuses on the performance evaluation of a rotor in hover and improvements to the current CFD solver in pursuit of this goal. The following section shows the organization of the thesis.

The second chapter outlines the computational methodology employed in the current work. This includes both OVERTURNS and the separate codes used to generate and modify the meshes.

The third chapter provides an overview of the results of the S-76 hover case, including a sweep of collective pitches as well as detailed evaluations of one set of conditions.

The fourth chapter summarizes the conclusions of the current work as well as providing recommendations for further study.

Chapter 2. Methodology

The following chapter details the numerical methods employed in simulating the aerodynamics of a helicopter rotor in hover. The work was performed using the structured Reynolds-Averaged Navier-Stokes (RANS) solver developed at UMD, OVERTURNS (Overset Transonic Unsteady Reynolds-Averaged Navier-Stokes). OVERTURNS is a three-dimensional MPI-parallel, structured, finite-difference solver which solves the RANS equations at each point in the computational domain. However, before discussing the solver itself, one must review the governing equations of the flow.

The Navier-Stokes equations comprise a system of partial differential equations (PDE's) which governs unsteady, compressible flow in the Eulerian reference frame. This Eulerian reference frame denotes a frame in which the observer focuses on a specified location and observes as the flow passes through this point. The Navier-Stokes equations are applied to the three-dimensional Cartesian reference frame, so a series of transformations must be applied to the original equations. In order, these transformations are: non-dimensionalize the NS equations, apply Reynolds-Averaging, and transforming from the Cartesian coordinates to Curvilinear coordinates. The non-dimensionalization serves to decrease numerical inaccuracies, Reynolds-Averaging normalizes the turbulence scale, and curvilinear coordinates simplify the numerical algorithms within the solver. Once these transformed equations are obtained, they are discretized in both time and space, forming a coupled system of equations to be solved.

2.1 Governing Equations

The Navier-Stokes equations are three-dimensional time dependent versions of the conservation of the mass, momentum, and energy of a fluid. Assuming a continuous flow, the system of equations can be expressed in Cartesian coordinates as follows,

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \boldsymbol{F_i}}{\partial x} + \frac{\partial \boldsymbol{G_i}}{\partial y} + \frac{\partial \boldsymbol{H_i}}{\partial z} = \frac{\partial \boldsymbol{F_v}}{\partial x} + \frac{\partial \boldsymbol{G_v}}{\partial x} + \frac{\partial \boldsymbol{H_v}}{\partial x} + \boldsymbol{S} \tag{2.1}$$

where $\boldsymbol{Q}$ is a vector of conserved variables, $\boldsymbol{F_i}$, $\boldsymbol{G_i}$, and $\boldsymbol{H_i}$ are inviscid flux vectors, and $\boldsymbol{F_v}$, $\boldsymbol{G_v}$, and $\boldsymbol{H_v}$ are viscous flux vectors. $\boldsymbol{S}$ denotes a vector of any body forces as well as changes due to transition from one reference frame to another, for example from inertial to rotational frames of reference. The vector of conserved variables, $\boldsymbol{Q}$, is defined as,

25

$$Q = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{Bmatrix} \tag{2.2}$$

for a given flow. Density is represented by ρ, u, v, and w denote the Cartesian velocity components, and E represents the total energy per unit volume given as,

$$E = \rho \left[e + \tfrac{1}{2}(u^2 + v^2 + w^2) \right] \tag{2.3}$$

where e is the internal energy per unit mass.

The inviscid flux vectors (F_i, G_i, H_i) are given as,

$$F_i = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E + p) \end{Bmatrix} \tag{2.4}$$

$$G_i = \begin{Bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ v(E + p) \end{Bmatrix} \tag{2.5}$$

$$H_i = \begin{Bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ w(E + p) \end{Bmatrix} \tag{2.6}$$

Meanwhile the viscous flux vectors (F_v, G_v, H_v) are given as,

$$F_v = \begin{Bmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{zx} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{Bmatrix} \tag{2.7}$$

$$G_v = \begin{Bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{zy} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y \end{Bmatrix} \tag{2.8}$$

$$H_v = \begin{Bmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z \end{Bmatrix} \tag{2.9}$$

where q_x, q_y, and q_z are thermal conduction terms computed as a function of temperature and pressure as follows,

$$q_i = -k \frac{\partial T}{\partial x_i} \tag{2.10}$$

Pressure, p, is given by the equation of state for a perfect gas as follows,

$$p = \rho R T \tag{2.11}$$

in which R is the specific gas constant. The current work assumes air which exists at the standard temperature and pressure (STP). With this assumption, the calorically perfect gas assumption is valid, meaning the specific heat values are constant. The specific heats for an ideal gas at constant volume (C_v) and constant pressure (C_p) are given as,

$$C_v = \frac{R}{\gamma-1}; \quad C_p = \frac{\gamma R}{\gamma-1} \tag{2.12}$$

Therefore the internal energy per mass, e, for a calorically perfect gas is,

$$e = C_v T \tag{2.13}$$

Applying Eq. (2.12) and (2.13) to the equation of state Eq. (2.11), we can rewrite it as,

$$p = (\gamma - 1)\rho e \tag{2.14}$$

27

Now apply the new equation of state, Eq. (2.14), to the total energy equation, Eq. (2.3), and rearranging,

$$p = (\gamma - 1)\left[E - \frac{1}{2}\rho(u^2 + v^2 + w^2)\right] \tag{2.15}$$

knowing that the ratio of specific heats (γ) is 1.4 for air at STP.

Furthermore, using Stokes' hypothesis, the viscous stress tensor for a Newtonian fluid, τ_{ij}, is given as,

$$\tau_{ij} = \mu\left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right] \tag{2.16}$$

Finally, the coefficient of molecular viscosity, μ, is given by Sutherland's formula[24] as follows,

$$\mu = C_1\frac{T^{\frac{3}{2}}}{T+C_2} \tag{2.17}$$

Where $C_1 = 1.4 \times 10^{-6}$ kg/(ms$\sqrt{K}$) and $C_2 = 110.4$ K for air at STP.

2.2   Non-Dimensional Form of Navier-Stokes Equations

If the Navier-Stokes equations were solved in their base form, numerical inaccuracies may arise due to combining values of multiple orders of magnitude. Therefore, the current work non-dimensionalizes all flow variables, normalizing them to unity, to minimize potential inaccuracies. These non-dimensional terms are given as follows (denoted by *),

$$x^* = \frac{x}{L}; \quad y^* = \frac{y}{L}; \quad z^* = \frac{z}{L}; \quad t^* = \frac{ta_\infty}{L} \tag{2.18}$$

$$u^* = \frac{u}{a_\infty}; \quad v^* = \frac{v}{a_\infty}; \quad w^* = \frac{w}{a_\infty}; \quad \mu^* = \frac{\mu}{a_\infty} \tag{2.19}$$

$$\rho^* = \frac{\rho}{\rho_\infty}; \quad p^* = \frac{p}{\rho_\infty a_\infty^2}; \quad T^* = \frac{T}{T_\infty} \tag{2.20}$$

where L is the reference length, here the chord length of the airfoil is chosen, a is the speed of sound, and ∞ denotes a freestream value. Further non-dimensional values are commonly used in aerodynamic analysis and are given below,

Mach Number: $\qquad M_\infty = \frac{V_\infty L}{a_\infty}$ $\qquad$ (2.21)

28

$$\text{Reynolds Number:} \qquad Re_\infty = \frac{\rho_\infty V_\infty L}{\mu_\infty} \qquad (2.22)$$

$$\text{Prandtl Number:} \qquad P_r = \frac{\mu C_p}{k} \qquad (2.23)$$

where V_∞ is the magnitude of the freestream velocity, equal to $\sqrt{u_\infty^2 + v_\infty^2 + w_\infty^2}$. Meanwhile, the known value of the Prandtl Number of air at STP is 0.72.

This non-dimensionalization creates a version of the Navier-Stokes equations identical to the original form with the exception of the viscous stress tensor and heat conduction terms. These altered equations are presented below, where Eq. (2.24) is the viscous stress tensor and Eq. (2.25) is the heat conduction term,

$$\tau_{ij} = \frac{\mu M_\infty}{Re_\infty}\left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right] \qquad (2.24)$$

$$q_i = -\frac{\mu M_\infty}{Re_\infty \Pr(\gamma-1)}\frac{\partial T}{\partial x_i} \qquad (2.25)$$

2.3   Reynolds-Averaged Navier-Stokes (RANS) Equations

If the current work dealt only in inviscid and laminar flows, the previous assumptions would be enough to now solve the governing equations. However, the flow near a helicopter blade is often turbulent, requiring further modification to the governing equations. The most accurate potential method involves using a Direct Numerical Simulation (DNS) to directly calculate all spatial and temporal turbulence scales. However, this level of precision is tremendously computationally expensive, requiring more computing resources than currently available. Therefore, the current work employs the RANS equations, which decompose the dependent variables of the governing equations, Eq. (2.1), into a mean and fluctuating component. These RANS equations are then averaged over a prescribed time period.

The time-averaged value (mean value) of a given variable *f* is defined as the following,

$$\bar{f} = \frac{1}{\Delta t}\int_{t_o}^{t_o+\Delta t} f \, dt \qquad (2.26)$$

where t_o is the current time and Δt is the time step size. For the fluctuating component, the time-averaged value is zero. Thus, any two fluctuating components, *f'* *and* *g'*, the following relations can be shown,

29

$$\overline{\bar{f}g'} = 0; \quad \overline{\bar{f}g} = \bar{f}\bar{g}; \quad \overline{f + g} = \bar{f} + \bar{g} \tag{2.27}$$

$$\overline{f'f'} \neq 0; \quad \overline{f'g'} \neq 0 \tag{2.28}$$

Now the dependent variables of the Navier-Stokes equations may be rewritten in terms of mean ($\bar{f}$) and fluctuating components (f') as follows,

$$u = \bar{u} + u'; \; v = \bar{v} + v'; \; w = \bar{w} + w'; \; \rho = \bar{\rho} + \rho'; \; p = \bar{p} + p'; \; T = \bar{T} + T' \tag{2.29}$$

With the mean and fluctuating terms defined, the turbulence intensity (Tu) is introduced as the ratio of the root-mean-square of the velocity fluctuations divided by the mean velocity,

$$Tu = \frac{V'}{\bar{V}} \tag{2.30}$$

$$V' = \sqrt{\frac{1}{3}[(u')^2 + (v')^2 + (w')^2]}; \quad \bar{V} = \sqrt{(\bar{u})^2 + (\bar{v})^2 + (\bar{w})^2} \tag{2.31}$$

Replacing the dependent variables in Eq. (2.1) with the decomposed versions from Eq. (2.29) and time-averaging gives the final RANS equations. This system of equations is identical to the original Navier-Stokes equations, with the addition of the terms representing fluctuating turbulent variables. These fluctuating terms behave like a viscous stress tensor, transmitting momentum due to turbulent fluctuations. Commonly known as the Reynolds stress tensor, these terms are represented as the following,

$$\left(\overline{\tau_{iJ}}\right)_{turb} = -\overline{\rho u_i' u_j'} \tag{2.32}$$

The Reynolds stress tensor takes the form of a symmetric 3x3 matrix, adding six new unknowns to the Reynolds-Averaged momentum equations. During computations, the Reynolds stress tensor is dependent on the mean flow quantities, employing a turbulence model to calculate these quantities and thus close the RANS equations. The turbulence model employed in the current work will be discussed further in Section 2.5.3.

2.4 Curvilinear Coordinate Transformation

While the Cartesian form of the Navier-Stokes equations is usable on any computational grid, the stencils employed for the numerical spatial derivatives assume a uniform grid spacing. The non-uniform body grid, which may include areas where the mesh stretches rapidly, is ill-suited for these schemes. Therefore the current work utilizes curvilinear coordinate transformation, in which the governing equations are mapped from the body conforming grid (x,y,z) onto a computational domain (ξ,η,ζ) with equal grid spacing. Applying chain rule differentiation to the governing

30

equations, Eq. (2.1), the Cartesian terms may be transformed to the computational domain as follows,

$$\frac{\partial \widetilde{Q}}{\partial t} + \frac{\partial \widetilde{F}}{\partial \xi} + \frac{\partial \widetilde{G}}{\partial \eta} + \frac{\partial \widetilde{H}}{\partial \zeta} = \widetilde{S} \tag{2.33}$$

where

$$\begin{aligned}
\widetilde{Q} &= \frac{1}{J} Q \\
\widetilde{F} &= \frac{1}{J} \big[\xi_t Q + \xi_x (F_i - F_v) + \xi_y (G_i - G_v) + \xi_z (H_i - H_v) \big] \\
\widetilde{G} &= \frac{1}{J} \big[\eta_t Q + \eta_x (F_i - F_v) + \eta_y (G_i - G_v) + \eta_z (H_i - H_v) \big] \qquad (2.34) \\
\widetilde{H} &= \frac{1}{J} \big[\zeta_t Q + \zeta_x (F_i - F_v) + \zeta_y (G_i - G_v) + \zeta_z (H_i - H_v) \big] \\
\widetilde{S} &= \frac{1}{J} S
\end{aligned}$$

here J represents the Jacobian of the coordinate transformation matrix, the determinant of the following 3x3 matrix,

$$\begin{bmatrix} \dfrac{\partial \xi}{\partial x} & \dfrac{\partial \xi}{\partial y} & \dfrac{\partial \xi}{\partial z} \\[2mm] \dfrac{\partial \eta}{\partial x} & \dfrac{\partial \eta}{\partial y} & \dfrac{\partial \eta}{\partial z} \\[2mm] \dfrac{\partial \zeta}{\partial x} & \dfrac{\partial \zeta}{\partial y} & \dfrac{\partial \zeta}{\partial z} \end{bmatrix} \tag{2.35}$$

Figure 2.1 below illustrates the transition from the physical body conforming domain to an equi-spaced computational domain. Figure 2.1 illustrates a two-dimensional transformation though this can easily be extended into three-dimensions, as in the current work.

(a) physical space   (b) computational space
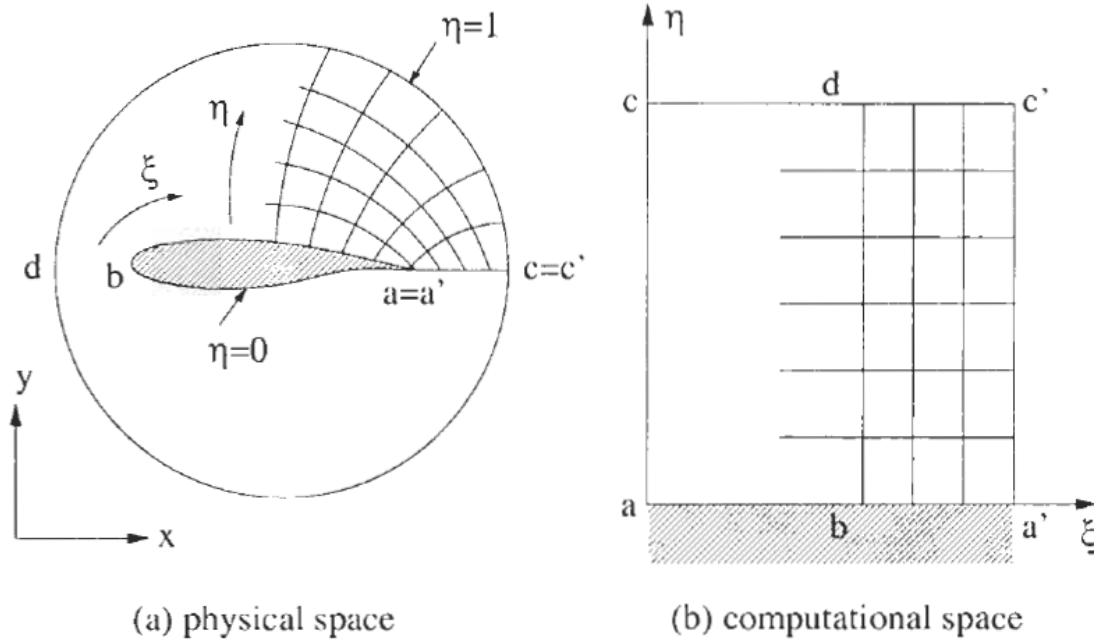
Figure 2.1 Mapping from the physical to computational domain using curvilinear coordinate transformation[25]

2.5   Numerical Algorithms

The following section details the numerical algorithms employed in discretizing the governing equations, turbulence modeling, and numerical boundary conditions employed in the OVERTURNS solver. A control volume is created around each point within the computational domain, with interfaces between these control volumes lying at the midpoint between adjacent points. By integrating over these cell faces the time rate of change of the conserved quantities can be evaluated at each time step. The semi-discrete form of the RANS equations in the curvilinear domain can be written as follows,

$$\frac{\partial \widetilde{Q}}{\partial t} = -\frac{\widetilde{F}_{j+\frac{1}{2}}-\widetilde{F}_{j-\frac{1}{2}}}{\Delta \xi} - \frac{\widetilde{G}_{k+\frac{1}{2}}-\widetilde{G}_{k-\frac{1}{2}}}{\Delta \eta} - \frac{\widetilde{H}_{l+\frac{1}{2}}-\widetilde{H}_{l-\frac{1}{2}}}{\Delta \zeta} + \widetilde{S}_{j,k,l} \qquad (2.36)$$

where (j,k,l) are the indices of the computational domain, corresponding to the (ξ,η,ζ) directions. Meanwhile $(j \pm \frac{1}{2}, k \pm \frac{1}{2}, l \pm \frac{1}{2})$ denote the interfaces between computational cells. The following section discusses the inviscid and viscous flux contributions seen at these cell interfaces.

32

2.5.1 Inviscid Fluxes

OVERTURNS evaluates the inviscid fluxes in two steps:

1. Reconstructing the conserved variables from Eq. (2.4-2.6) at the cell faces

2. Evaluating the fluxes at the cell faces using these reconstructed variables

These inviscid fluxes represent the convection of the flow field between cells in the computational domain. Figure 2.2 below demonstrates calculations of two flow states, to the left and right of the cell interface $j + \frac{1}{2}$, based on the conservative variables at cells j and $j+1$.
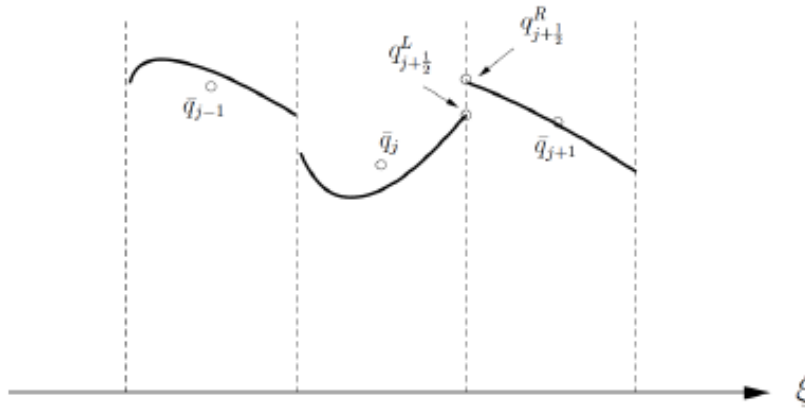
Figure 2.2 Illustration of a one-dimensional piecewise reconstruction[26]

The current version of OVERTURNS contains three available reconstruction schemes to calculate the flow at the left and right cell faces:

1. MUSCL – Monotone Upstream-Centered Scheme for Conservation Laws[27]

2. WENO – Weighted Essentially Non-Oscillatory[28]

3. CRWENO – Compact-Reconstruction Weighted Essentially Non-Oscillatory[29]

In all cases the left and right states are first used to define a local Riemann problem and the interface flux is obtained using Roe flux difference splitting[30] as follows,

$$\boldsymbol{F}(q^L, q^R) = \frac{\boldsymbol{F}(q^L) + \boldsymbol{F}(q^R)}{2} - \left| \tilde{A}(q^L, q^R) \right| \frac{q^R - q^L}{2} \tag{2.37}$$

Where $\mathbf{F}_L$ and $\mathbf{F}_R$ are the left and right state fluxes respectively and $\tilde{A}$ is the Roe-averaged Jacobian matrix. The second term in Eq. (2.37) accounts for numerical dissipation, using Harten's entropy correction to the eigenvalues of the flux Jacobian as given by,

$$|\lambda| = \begin{cases} |\lambda|, & if|\lambda| > \delta \\ \frac{\lambda^2+\delta^2}{2\delta}, & if|\lambda| \le \delta \end{cases} \tag{2.38}$$

In which $\delta = \max[0, (\bar{\lambda}_{i+1/2} - \bar{\lambda}_i), (\bar{\lambda}_{i+1} - \bar{\lambda}_{i+1/2})]$.

2.5.1.1   MUSCL Scheme with Koren's Limiter

The third-order Monotone Upstream-Centered Scheme for Conservation Laws[27] (MUSCL), employing Koren's differentiable limiter[31], is designed as a finite-volume method which can provide accurate numerical solutions even in the presence of shocks or other discontinuities. Koren's limiter serves to limit higher order reconstruction in regions of high gradients, thus the final scheme is third order accurate in regions of low gradients while decreasing to first order in the presence of discontinuities (as evidenced by a high gradient). Using the averaged cell values of $\bar{q}_{j+1}, \bar{q}_j, \bar{q}_{j-1}$, the interface values $q_{j-1/2}^L$ and $q_{j+1/2}^R$ are given through cell reconstruction as follows,

$$q_{j-1/2}^L = \bar{q} + \phi_i\left[\frac{1}{3}(\bar{q}_{j+1} - \bar{q}_j) + \frac{1}{6}(\bar{q}_j - \bar{q}_{j-1})\right] \tag{2.39}$$

$$q_{j-1/2}^R = \bar{q} - \phi_i\left[\frac{1}{3}(\bar{q}_{j+1} - \bar{q}_j) + \frac{1}{6}(\bar{q}_j - \bar{q}_{j-1})\right] \tag{2.40}$$

where ϕ is Koren's differentiable limiter, given as,

$$\phi_j = \frac{3\Delta\bar{q}_j\nabla\bar{q}_j+\epsilon}{2(\Delta q_j - \nabla q_j)^2 + 3\Delta q_j\nabla q_j + \epsilon} \tag{2.41}$$

in which ϵ is a small constant to ensure division by zero does not occur. Δ and ∇ denote forward and backward operators as defined by $\Delta\bar{q}_j = (q_{j+1} - q_j)$ and $\nabla\bar{q}_j = (q_j - q_{j-1})$.

2.5.1.2   WENO Scheme

The following section outlines the fifth-order Weighted Essentially Non-Oscillatory[28] (WENO5) scheme as defined by Henrick et. al. This finite-volume scheme employs a cell-averaged approach, calculating the spatial derivative through a convex combination of fluxes at each stencil. The contribution of each flux is weighted based on the smoothness of the solution, or

magnitude of the flow gradient, over each stencil. In a region of high gradient, the smoothest stencil is the most heavily weighted, resulting in a third order reconstruction over the five point stencil. Meanwhile, in smooth regions fifth order accuracy is achieved. The interface parameters are calculated through the following,

$$q_{j+1/2}^L = \sum_{r=0}^{k} w_r v_r^L \tag{2.42}$$

in which w_r are the weight terms and v_r^L represent the interpolations from the given stencils. Here q represents the interpolated flux computed based on the discrete values of the flux represented below by $\bar{q}$. The weight term can be further broken down as follows,

$$w_r = \frac{\alpha_r}{\sum_{s=0}^{k} \alpha_s} \tag{2.43}$$

$$\alpha_r = \frac{d_r}{(\beta_r + \epsilon)^2} \tag{2.44}$$

where d_r are optimal weight coefficients, β_r are smoothness indicators, and $\epsilon = 10^{-6}$. The values of v_r^L, d_r, and β_r are presented as follows,

$$v_0^L = \frac{1}{3}\bar{q}_{j-3} - \frac{7}{6}\bar{q}_{j-1} + \frac{11}{6}\bar{q}_j$$

$$v_1^L = -\frac{1}{6}\bar{q}_{j-1} + \frac{5}{6}\bar{q}_j + \frac{1}{3}\bar{q}_{j+1}$$

$$v_2^L = \frac{1}{3}\bar{q}_j + \frac{5}{6}\bar{q}_{j+1} - \frac{1}{6}\bar{q}_{j+2}$$

$$d_0 = \frac{1}{10}$$

$$d_1 = \frac{3}{5}$$

$$d_2 = \frac{3}{10}$$

$$\beta_0 = \frac{13}{12}\left(\bar{q}_{j-2} - 2\bar{q}_{j-1} + \bar{q}_j\right)^2 + \frac{1}{4}\left(\bar{q}_{j-2} - 4\bar{q}_{j-1} + 3\bar{q}_j\right)^2$$

$$\beta_1 = \frac{13}{12}\left(\bar{q}_{j-1} - 2\bar{q}_j + \bar{q}_{j+1}\right)^2 + \frac{1}{4}\left(\bar{q}_{j+1} - \bar{q}_{j-1}\right)^2$$

$$\beta_2 = \frac{13}{12}\left(\bar{q}_j - 2\bar{q}_{j+1} + \bar{q}_{j+2}\right)^2 + \frac{1}{4}\left(3\bar{q}_j - 4\bar{q}_{j+1} + \bar{q}_{j+2}\right)^2 \tag{2.45}$$

2.5.1.3   CRWENO Scheme

This section provides an overview of the fifth-order Compact-Reconstruction Weighted Essentially Non-Oscillatory[29] (CRWENO) scheme as presented by Ghosh et. al. Based on the existing WENO algorithm, this method calculates lower order compact stencils at each interface before computing the optimal weights to result in a higher-order scheme. As with the WENO scheme, the weights are calculated based on the smoothness of the stencil. The final result of these modifications is a scheme which provides a high-order compact scheme in smooth regions and lower-order biased compact scheme near discontinuities. The primary advantage of this scheme is not that the accuracy is improved beyond fifth-order but that the absolute error is significantly reduced. The interface parameters are given in the same general form as Eq. (2.42) from the WENO scheme while the general form of the weight term is found in Eq. (2.43-2.44). The third-order interpolations at $j+1/2$ ($\hat{f}^n_{j\pm1/2}$), optimal weights (c_n), and smoothness indicators (β_n) are given below,

$$
\begin{aligned}
&\frac{2}{3}q^0_{j-1/2} + \frac{1}{3}q^0_{j+1/2} = \frac{1}{6}\left(\bar{q}_{j-1} + 5\bar{q}_j\right) \\
&\frac{1}{3}q^1_{j-1/2} + \frac{2}{3}q^1_{j+1/2} = \frac{1}{6}\left(5\bar{q}_j + \bar{q}_{j+1}\right) \\
&\frac{2}{3}q^2_{j+1/2} + \frac{1}{3}q^2_{j+3/2} = \frac{1}{6}\left(\bar{q}_j + 5\bar{q}_{j+1}\right) \\
&c_0 = \frac{1}{5} \\
&c_1 = \frac{1}{2} \\
&c_2 = \frac{3}{10} \\
&\beta_0 = \frac{13}{12}\left(\bar{q}_{j-2} - 2\bar{q}_{j-1} + \bar{q}_j\right)^2 + \frac{1}{4}\left(\bar{q}_{j-2} - 4\bar{q}_{j-1} + 3\bar{q}_j\right)^2 \\
&\beta_1 = \frac{13}{12}\left(\bar{q}_{j-1} - 2\bar{q}_j + \bar{q}_{j+1}\right)^2 + \frac{1}{4}\left(\bar{q}_{j-1} - \bar{q}_{j+1}\right)^2 \\
&\beta_2 = \frac{13}{12}\left(\bar{q}_j - 2\bar{q}_{j+1} + \bar{q}_{j+2}\right)^2 + \frac{1}{4}\left(3\bar{q}_j - 4\bar{q}_{j+1} + \bar{q}_{j+2}\right)^2
\end{aligned}
\tag{2.46}
$$

2.5.2 Viscous Fluxes

The viscous fluxes of the governing equations take the following form, in terms of curvilinear coordinates,

$$
\frac{\partial}{\partial\xi}\left(\alpha\frac{\partial\beta}{\partial\eta}\right)
\tag{2.47}
$$

which the current work evaluates using the following second-order central differencing scheme,

$$\frac{1}{\Delta\xi}\left(\left[\alpha_{j+1/2,k}\left(\frac{\beta_{j+\frac{1}{2},k+1}-\beta_{j+\frac{1}{2},k}}{\Delta\eta}\right)\right]-\left[\alpha_{j-1/2,k}\left(\frac{\beta_{j-\frac{1}{2},k+1}-\beta_{j-\frac{1}{2},k}}{\Delta\eta}\right)\right]\right) \tag{2.48}$$

where $\delta_{j+1/2,k}=\frac{\delta_{j,k}+\delta_{j+1,k}}{2}$ with $\delta=(\alpha,\beta)$.

2.5.3 Turbulence Modeling

As mentioned previously, the RANS equations split the state variables into mean and fluctuating components to avoid modeling all turbulence effects. Instead, the fluctuating terms introduce new terms called the Reynolds stress tensor which consists of six transport equations as well as an equation of the energy dissipation rate. These models are known as the Reynolds Stress Models[32,33], and though physically realistic, they are also computationally expensive and experience problems of stability and convergence. For these reasons, the more commonly used method is relating the Reynolds stress tensor to the mean strain rate through the Boussinesq eddy viscosity as follows,

$$\left(\bar{\tau}_{ij}\right)_{turb}=-\bar{\rho}\overline{u_i'u_j'}=\frac{2}{3}\bar{\rho}k\delta_{ij}-\mu_t\left[\left(\frac{\partial\bar{u}_i}{\partial x_j}+\frac{\partial\bar{u}_j}{\partial x_i}\right)-\frac{2}{3}\frac{\partial\bar{u}_k}{\partial x_k}\delta_{ij}\right] \tag{2.49}$$

Where *k* represents turbulent kinetic energy,

$$k=\frac{1}{2}[(\bar{u}_1')^2+(\bar{v}_1')^2+(\bar{w}_1')^2] \tag{2.50}$$

and μ_t represents turbulent (eddy) viscosity. From Eq. (2.49) the total viscous stress tensor can be written as,

$$\left(\bar{\tau}_{ij}\right)_{turb}=\frac{2}{3}\bar{\rho}k\delta_{ij}-(\mu+\mu_t)\left[\left(\frac{\partial\bar{u}_i}{\partial x_j}+\frac{\partial\bar{u}_j}{\partial x_i}\right)-\frac{2}{3}\frac{\partial\bar{u}_k}{\partial x_k}\delta_{ij}\right] \tag{2.51}$$

Researchers have developed a number of models to calculate turbulent kinetic energy and eddy viscosity as a function of the mean quantities of the flow. The majority of these models are created based on experimental results and theoretical predictions, resulting in modern turbulence modeling being driven by empirical data. One of these methods is the one-equation Spalart-Allmaras[34] (SA) model, which is widely used in the aerospace field. This model is employed in all cases within the current work. The SA model solves a transport equation for the eddy viscosity, and is detailed in the following section.

2.5.4 Spalart-Allmaras (SA) Turbulence Model

The SA turbulence model is a one-equation turbulence model commonly employed in the aerospace industry and used in all cases of the current work. It relates the Reynolds stresses and mean strain as follows,

$$\overline{u'_i u'_j} = -2\mu_t S_{ij} \tag{2.52}$$

where the eddy viscosity in the isotropic equation above is related to the turbulence variable $\tilde{v}$ as the following,

$$\mu_t = v_t = \rho \tilde{v} f_{v1} \tag{2.53}$$

in which f_{v1} is a function of the turbulence variable $\tilde{v}$ and molecular viscosity v as shown below,

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \tag{2.54}$$

where $\chi = \tilde{v}/v$ and $c_{v1} = 7.1$. Meanwhile, the turbulence field variable $\tilde{v}$ may be obtained through the following PDE,

$$\frac{\partial \tilde{v}}{\partial t} + \mu_j \frac{\partial \tilde{v}}{\partial x_j} = \frac{1}{\sigma} \left[\frac{\partial}{\partial x_j} \left((v + \tilde{v}) \frac{\partial \tilde{v}}{\partial x_j} \right) + c_{b2} \frac{\partial \tilde{v}}{\partial x_j} \frac{\partial \tilde{v}}{\partial x_j} \right] + c_{b1} \tilde{S} \tilde{v} - c_{w1} f_w \left(\frac{\tilde{v}}{d} \right)^2 \tag{2.55}$$

and where d is the distance from the current point of the flow to the nearest wall. Additionally,

$$\tilde{S} = max \left[\Omega + \frac{\tilde{v}}{\kappa^2 d^2} f_{v2} , 0.3\Omega \right] \tag{2.56}$$

$$f_w = g \left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right]^{1/6} \tag{2.57}$$

in which Ω is vorticity magnitude and f_{v2} and g are as follows,

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \tag{2.58}$$

$$g = r + c_{w2}(r^6 - r) \tag{2.59}$$

$$r = min \left[\frac{\tilde{v}}{\tilde{S} \kappa^2 d^2} , 10 \right] \tag{2.60}$$

where the constants introduced above are given as follows,

$$c_{b1} = 0.1355 \quad c_{b2} = 0.622 \quad \kappa = 0.41 \quad \sigma = \frac{2}{3}$$
$$c_{v1} = 7.1 \quad c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1+c_{b2}}{\sigma} \quad c_{w2} = 0.3 \quad c_{w3} = 2 \tag{2.61}$$

The left hand side (LHS) of Eq. (2.55) above represents the convection of the turbulence field variable ($\tilde{v}$) at mean flow velocity (u). Meanwhile, on the right hand side (RHS) of Eq. (2.55) the first term indicates diffusion of $\tilde{v}$, second term represents production of $\tilde{v}$, and the final term measures the destruction of $\tilde{v}$.

2.5.4.1 γ - Re_θ SA Transition Model

Transition modeling serves to improve flow predictions in cases with significant regions of laminar flow. The current work employs this γ - Re_θ SA Transition Model developed by Medida et. al[35] as an addition to the base SA Turbulence Model. This method employs the local intermittency value, γ, to govern boundary layer transition by controlling the amount of turbulent kinetic energy generated. This intermittency field is computed through the following transport equation,

$$\frac{D(\rho\gamma)}{Dt} = P_\gamma - D_\gamma + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_f}\right)\frac{\partial\gamma}{\partial x_j}\right] \tag{2.62}$$

in which,

$$P_\gamma = \rho F_{onset} G_{onset} \max\left(\frac{\Omega}{F_{length}}, \frac{1.0}{F_{length,min}}\right), \text{If } \gamma > 1.0, P_\gamma = (1 - \gamma)P_\gamma \tag{2.63}$$

$$D_\gamma = \rho\Omega\gamma(1.0 - G_{onset}) \tag{2.64}$$

$$G_{onset} = \begin{cases} 1.0, \text{if } \max(F_{onset}) > 1.0 \text{ at a given point} \\ 0.0, \text{otherwise} \end{cases} \tag{2.65}$$

$$F_{onset} = \max(F_{onset2} - F_{onset3}, 0) \tag{2.66}$$

$$F_{onset1} = \frac{Re_v}{2.193Re_{\theta c}} \tag{2.67}$$

$$F_{onset2} = \min(\max(F_{onset1}, F_{onset1}^4), 4.0) \tag{2.68}$$

$$F_{onset3} = \max(2 - (0.25R_T)^3, 0) \tag{2.69}$$

$$Re_v = \frac{\rho d^2 S}{\mu}, \quad Re_{\theta c} = 0.62\overline{Re}_{\theta t}, \quad R_T = \frac{\mu}{\mu_t} \tag{2.70}$$

$$F_{length} = 40.0, \quad F_{length,min} = 2.5 \tag{2.71}$$

In the above equations, $Re_{\theta c}$, denotes the critical Reynolds number and governs the transition onset location. At this location the intermittency increases within the boundary layer. The length of this transition region depends on the F_{length} value. Both the critical Reynolds number and length of transition region depend on the local momentum thickness, $\overline{Re}_{\theta t}$. The $\overline{Re}_{\theta t}$ field is dependent on a new transition equation as follows,

$$\frac{D(\rho \overline{Re}_{\theta t})}{Dt} = P_{\theta t} + \frac{\partial}{\partial x_j}\left[\sigma_{\theta t}(\mu + \mu_t)\frac{\partial \overline{Re}_{\theta t}}{\partial x_j}\right] \text{ where } \overline{Re}_{\theta t} = \frac{\rho_\infty U_\infty \theta_t}{\mu_\infty} \tag{2.72}$$

where

$$P_{\theta t} = c_{\theta t}\frac{\rho}{t}(Re_{\theta t} - \overline{Re}_{\theta t})(1.0 - F_{\theta t}) \tag{2.73}$$

$$F_{\theta t} = min\left(e^{-\left(\frac{d}{\delta}\right)^4}, 1.0\right) \tag{2.74}$$

$$\theta_{BL} = \frac{\overline{Re}_{\theta t}\mu}{\rho U}, \quad \delta_{BL} = 7.5\theta_{BL}, \quad \delta = \frac{50\Omega d}{U}\delta_{BL} \tag{2.75}$$

The local momentum, $\overline{Re}_{\theta t}$, is determined using experimental results while the value of $\overline{Re}_{\theta t\infty}$ results from a piecewise interpolation of the freestream turbulence values given below in Table 2.1[36]. Solving for $\overline{Re}_{\theta t}$ in Eq. (2.72) is then done by applying the values from Table 2.1 to the following,

$$Re_{\theta t} = Re_{\theta t\infty}F(\lambda_\theta) \tag{2.76}$$

$$F(\lambda_\theta) = \begin{cases} -(-12.986\lambda_\theta - 123.66\lambda_\theta^2 - 405.689\lambda_\theta^3)e^{-\left(\frac{Tu}{1.5}\right)^{1.5}}, \text{If } \lambda_\theta \leq 0 \\ 1 + 0.275(1 - e^{-3.5\lambda_\theta})e^{-\frac{Tu}{0.5}}, \qquad\qquad\qquad \text{If } \lambda_\theta > 0 \end{cases} \tag{2.77}$$

$$\lambda_\theta = \frac{\rho\theta^2}{\mu}\frac{dU}{ds} \tag{2.78}$$

$$\sigma_f = 1.0, \quad c_{\theta t} = 0.03, \quad \sigma_{\theta t} = 2.0 \tag{2.79}$$

$Tu\%$	$Re_{\theta t\infty}$
0.01	1800.0
0.03	1135.0
0.51	894.0
1.33	392.0
2.00	252.0
5.25	165.0
6.50	100.0

Table 2.1 Piecewise linear correlations between turbulence intensity (*Tu%*) and freestream turbulence values ($Re_{\theta t\infty}$) (Reproduced from Reference 36)

2.5.4.2   Delayed Detached Eddy Simulation (DDES)

The Delayed Detached Eddy Simulation (DDES) is a hybrid RANS-LES (Large Eddy Simulation) method created by Spalart et. al[37-39]. This method is based on DES, a modification to the SA model which separates the turbulence model into two types of behavior, RANS behavior within the boundary layer (BL) and a subgrid model outside of the BL. To do so, DDES replaces the original length scale, d in Eq. (2.55), with a new scale, $\tilde{d}$, as follows,

$$\tilde{d} = \min(d, C_{DES}\Delta) \tag{2.80}$$

in which *C_{DES}* is a constant, normally taking the following form,

$$\Delta = \max(\Delta x, \Delta y, \Delta z) \tag{2.78}$$

To further improve the turbulence model, Scotti et. al[41] developed an anisotropic grid correction which alters the definition of Δ to the following,

$$\Delta = \cosh\left(\sqrt{\frac{4}{27}\left[\ln(a_1)^2 - \ln(a_1)\ln(a_2) + \ln(a_2)^2\right]}\right) \times (\Delta x \times \Delta y \times \Delta z)^{\frac{1}{3}} \tag{2.79}$$

From Eq. (2.77-79) we can see that $\tilde{d}$ is dependent only on grid parameters, which may cause a problem known as modeled stress depletion (MSD) in regions of thick BL's or shallow separation. MSD denotes an issue where the DDES method has switched to use the LES model but the grid is too coarse to accurately simulate the resulting effects. This may lead to a reduction in the Reynolds stress modeled, causing premature separation in an effect called grid induced separation. Therefore, this method further modifies $\tilde{d}$ to take the eddy viscosity into account as given below,

41

$$\tilde{d} = d - f_d \max(0, d - D_{DES}\Delta) \tag{2.80}$$

$$f_d = 1 - \tanh([8r_d]^3) \tag{2.81}$$

$$r_d = \frac{\nu_t + \nu}{\sqrt{U_{i,j}U_{i,j}}\kappa^2 d^2} \tag{2.82}$$

The addition of eddy viscosity prevents the issue of MSD, by ensure that the DES limiter is not active within the boundary layer. Meanwhile, the LES is still employed in separated regions.

2.5.5 Time Integration

While the RHS of Eq. (2.36) governs the inviscid and viscous fluxes in the flow, the LHS evaluates the conserved variables (Q) in time. There are two time marching methods available, explicit and implicit marching. Explicit methods employ only information from the previous time step when calculating the values at the current step. However, implicit methods indirectly employ data from the current time step when calculating the conserved variables. Implicit methods do require the inversion and evaluation of large matrices to accomplish this but they demonstrate greater stability and convergence performance compared to explicit methods. The sizes of explicit methods are also limited by mesh size and local flow quantities. For these reasons, the current work employs a second-order accurate backwards in time (BDF2) method to integrate Eq. (2.36) as shown below,

$$\frac{\partial \widetilde{Q}^{n+1}}{\partial t} = -\frac{\widetilde{F}^{n+1}_{j+\frac{1}{2}} - \widetilde{F}^{n+1}_{j-\frac{1}{2}}}{\Delta\xi} - \frac{\widetilde{G}^{n+1}_{k+\frac{1}{2}} - \widetilde{G}^{n+1}_{k+\frac{1}{2}}}{\Delta\eta} - \frac{\widetilde{H}^{n+1}_{l+\frac{1}{2}} - \widetilde{H}^{n+1}_{l+\frac{1}{2}}}{\Delta\zeta} + \widetilde{S}^{n+1}_{j,k,l} \tag{2.83}$$

where

$$\frac{\partial \widetilde{Q}^{n+1}}{\partial t} = \frac{3\widetilde{Q}^{n+1} - 4\widetilde{Q}^{n+1} + \widetilde{Q}^{n+1}}{2\Delta t} \tag{2.84}$$

The fluxes at time step, $n+1$, in Eq. (2.83) are unknown, therefore they must be linearized and expressed as a combination of conserved variables and fluxes at the previous time step, n. Using a Taylor series expansion, the terms are linearized about $\widetilde{Q}^n$ as seen below,

$$\widetilde{F}^{n+1} = \widetilde{F}^n + \tilde{A}\Delta\widetilde{Q} + O(\Delta t^2) \tag{2.85}$$

$$\widetilde{G}^{n+1} = \widetilde{G}^n + \tilde{B}\Delta\widetilde{Q} + O(\Delta t^2) \tag{2.86}$$

$$\widetilde{H}^{n+1} = \widetilde{H}^n + \tilde{C}\Delta\widetilde{Q} + O(\Delta t^2) \tag{2.87}$$

42

in which $\Delta \widetilde{Q} = \widetilde{Q}^{n+1} - \widetilde{Q}^n$ represents the difference between old and linearized solutions, while A, B, and C are the flux Jacobians given by $\frac{\partial \widetilde{F}}{\partial \widetilde{Q}}, \frac{\partial \widetilde{G}}{\partial \widetilde{Q}}, \frac{\partial \widetilde{H}}{\partial \widetilde{Q}}$. Linearization is also applied to the source term, S. This linearization is second-order and will not degrade the accuracy of the also second-order BDF2 scheme. The final linearized form of Eq. (2.83) is given below in terms of $\Delta \widetilde{Q}$, the solution update,

$$\left[I + \Delta t \left(\delta_\xi \tilde{A}^n + \delta_\eta \tilde{B}^n + \delta_\zeta \tilde{C}^n\right)\right]\widetilde{Q} = -\Delta t \left(\delta_\xi \widetilde{F}^n + \delta_\eta \widetilde{G}^n + \delta_\zeta \widetilde{H}^n - \widetilde{S}^n\right) \quad (2.88)$$

The RHS of the above equation includes the physical flow solution while the LHS represents the stability and convergence of the numerical scheme. This implicit algorithm results in a large banded system of algebraic equations. This matrix, while sparse, would prove expensive to solve directly to generate a solution for $\Delta \widetilde{Q}^n$. Therefore, approximations to the LHS are employed to simplify the matrix inversion, though at the cost of convergence speed.

OVERTURNS has two available methods for the inversion of the system equations,

1. LUSGS – Lower-Upper Symmetric Gauss-Seidel[40]

2. DADI – Diagonalized Alternate Direction Implicit[41]

The current work employs the LUSGS method, though both are described below.

2.5.5.1   Lower-Upper Symmetric Gauss-Seidel (LUSGS) Algorithm

The LUSGS algorithm denotes an approximate factorization method which evaluates the LHS of the linearized form of Eq. (2.88), the semi-discrete RANS equations[40]. The factorization separates the terms from the RANS equations into three distinct groups, a lower diagonal L, an upper diagonal U, and a main diagonal D. These groups take the following form,

$$[L + D + U]\Delta \widetilde{Q}^n \approx [D + L]D^{-1}[D + U]\Delta \widetilde{Q}^n = -\Delta t \left(\delta_\xi \widetilde{F}^n + \delta_\eta \widetilde{G}^n + \delta_\zeta \widetilde{H}^n - \widetilde{S}^n\right) \quad (2.89)$$

in which,

$$L = \Delta t \left(-\tilde{A}_{j-1,k,l}^+ - \tilde{B}_{j,k-1,l}^+ - \tilde{C}_{j,k,l-1}^+\right) \quad (2.90)$$

$$U = \Delta t \left(\tilde{A}_{j+1,k,l}^- + \tilde{B}_{j,k+1,l}^- + \tilde{C}_{j,k,l+1}^-\right) \quad (2.91)$$

$$D = I + \Delta t \left(\tilde{A}_{j,k,l}^+ - \tilde{A}_{j,k,l}^- + \tilde{B}_{j,k,l}^+ - \tilde{B}_{j,k,l}^- - \tilde{C}_{j,k,l}^+ - \tilde{C}_{j,k,l}^-\right) \quad (2.92)$$

The change in the solution, $\Delta\widetilde{\boldsymbol{Q}}^n$, can then be found by solving the system of equations defined by Eq. (2.89) and Eq. (2.90-92) as seen below,

$$[D + L]\Delta\overline{\boldsymbol{Q}} = -\Delta t\big(\delta_\xi\widetilde{\boldsymbol{F}}^n + \delta_\eta\widetilde{\boldsymbol{G}}^n + \delta_\zeta\widetilde{\boldsymbol{H}}^n - \widetilde{\boldsymbol{S}}^n\big) \tag{2.93}$$

$$[D + U]\Delta\widetilde{\boldsymbol{Q}} = D\Delta\overline{\boldsymbol{Q}} \tag{2.94}$$

Here, L, D, and U represent 5 x 5 block matrices of the three dimensional Navier-Stokes equations. Due to the size of these matrices, it would prove computationally expensive to invert the sum terms ($D + L$ and $D + U$). Therefore, a spectral radius approximation is employed for the flux Jacobian matrices $(\tilde{A}, \tilde{B}, \tilde{C})$ to transform the main diagonal, D, into a diagonal matrix. The spectral radius approximation is as follows,

$$\begin{aligned}
\tilde{A}^+ &= \tfrac{1}{2}\big(\tilde{A} + \sigma_\xi\big), & \tilde{A}^- &= \tfrac{1}{2}\big(\tilde{A} - \sigma_\xi\big) \\
\tilde{B}^+ &= \tfrac{1}{2}\big(\tilde{B} + \sigma_\eta\big), & \tilde{B}^- &= \tfrac{1}{2}\big(\tilde{B} - \sigma_\eta\big) \\
\tilde{C}^+ &= \tfrac{1}{2}\big(\tilde{C} + \sigma_\zeta\big), & \tilde{C}^- &= \tfrac{1}{2}\big(\tilde{C} - \sigma_\zeta\big)
\end{aligned} \tag{2.95}$$

in which $\tilde{A}^{+-}, \tilde{B}^{+-}$, and $\tilde{C}^{+-}$ represent the right and left Jacobians. Meanwhile,

$$\begin{aligned}
\sigma_\xi &= \big|U_\xi\big| + c + \frac{2\mu(\xi_x^2 + \xi_z^2 + \xi_z^2)}{\rho} \\
\sigma_\eta &= \big|U_\eta\big| + c + \frac{2\mu(\eta_x^2 + \eta_z^2 + \eta_z^2)}{\rho} \\
\sigma_\zeta &= \big|U_\zeta\big| + c + \frac{2\mu(\zeta_x^2 + \zeta_z^2 + \zeta_z^2)}{\rho}
\end{aligned} \tag{2.96}$$

where U_k is the contravariant velocity in the k-direction. The approximate factorization leads to errors when compared with the exact solution which are reduced through a dual time stepping method employing Newton-like sub-iterations. The dual time stepping method is described fully in Section 2.5.5.3.

2.5.5.2   Diagonalized Alternating Direction Implicit (DADI) Algorithm

The DADI algorithm inverts the LHS of the linearized form of the semi-discrete RANS equations, Eq. (2.36). Developed by Pulliam and Chaussee[41], the method splits the LHS of Eq. (2.36) into three factors as below,

$$\big[I + \Delta t\big(\delta_\xi\tilde{A} + \delta_\eta\tilde{B} + \delta_\zeta\tilde{C}\big)\big]\Delta\widetilde{\boldsymbol{Q}} \approx \big[I + \Delta t\delta_\xi\tilde{A}\big]\big[I + \Delta t\delta_\eta\tilde{B}\big]\big[I + \Delta t\delta_\zeta\tilde{C}\big]$$

44

$$= -\Delta t \left(\delta_\xi \widetilde{\boldsymbol{F}}^n + \delta_\eta \widetilde{\boldsymbol{G}}^n + \delta_\zeta \widetilde{\boldsymbol{H}}^n - \widetilde{\boldsymbol{S}}^n \right) \tag{2.97}$$

These factors can be further simplified by diagonalizing the inviscid components of the flux Jacobians, producing the following,

$$
\begin{aligned}
\tilde{A} &= T_\xi \Lambda_\xi T_\xi^{-1} \\
\tilde{B} &= T_\eta \Lambda_\eta T_\eta^{-1} \\
\tilde{C} &= T_\zeta \Lambda_\zeta T_\zeta^{-1}
\end{aligned}
\tag{2.98}
$$

in which T_ξ is a matrix containing the set of left eigenvectors of matrix $\tilde{A}$ while T_ξ^{-1} is the set of right eigenvectors of $\tilde{A}$. Meanwhile T_η and T_ζ and their inverses correspond to $\tilde{B}$ and $\tilde{C}$ respectively. Additionally, $\tilde{A}$ gives the set of eigenvalues of matrix Λ_ξ, with similar relations between $\tilde{B}$ and $\tilde{C}$ with Λ_η and Λ_ζ.

Now the diagonalized flux Jacobians, given by Eq. (2.98), can be substituted into the LHS of Eq. (2.97) as follows,

$$
\begin{aligned}
&\left[T_\xi T_\xi^{-1} \left(I + \Delta t \delta_\xi T_\xi \Lambda_\xi T_\xi^{-1} \right) \right] \left[T_\eta T_\eta^{-1} \left(I + \Delta t \delta_\eta T_\eta \Lambda_\eta T_\eta^{-1} \right) \right] \\
&\left[T_\zeta T_\zeta^{-1} \left(I + \Delta t \delta_\zeta T_\zeta \Lambda_\zeta T_\zeta^{-1} \right) \right] \Delta \widetilde{\boldsymbol{Q}} = \Delta t \left(\delta_\xi \widetilde{\boldsymbol{F}}^n + \delta_\eta \widetilde{\boldsymbol{G}}^n + \delta_\zeta \widetilde{\boldsymbol{H}}^n - \widetilde{\boldsymbol{S}}^n \right)
\end{aligned}
\tag{2.99}
$$

Assuming the inviscid flux Jacobian eigenvectors are constant in the local area gives the following form of Eq. (2.99),

$$
\begin{aligned}
&\left[T_\xi \left(I + \Delta t \delta_\xi \Lambda_\xi \right) T_\xi^{-1} \right] \left[T_\eta \left(I + \Delta t \delta_\eta \Lambda_\eta \right) T_\eta^{-1} \right] \left[T_\zeta \left(I + \Delta t \delta_\zeta \Lambda_\zeta \right) T_\zeta^{-1} \right] \Delta \widetilde{\boldsymbol{Q}} = \\
&\Delta t \left(\delta_\xi \widetilde{\boldsymbol{F}}^n + \delta_\eta \widetilde{\boldsymbol{G}}^n + \delta_\zeta \widetilde{\boldsymbol{H}}^n - \widetilde{\boldsymbol{S}}^n \right)
\end{aligned}
\tag{2.100}
$$

Now the diagonal algorithm has reduced the system to a series of 5 x 5 matrix multiplications and scalar tridiagonal inversions. The inversion process is shown in the following steps,

$$
\begin{aligned}
S_1 &= T_\xi^{-1} \Delta t \left(\delta_\xi \widetilde{\boldsymbol{F}}^n + \delta_\eta \widetilde{\boldsymbol{G}}^n + \delta_\zeta \widetilde{\boldsymbol{H}}^n - \widetilde{\boldsymbol{S}}^n \right) \\
S_2 &= \left(I + \Delta t \delta_\xi \Lambda_\xi \right)^{-1} S_1 \\
S_3 &= \left(T_\xi^{-1} T_\eta \right)^{-1} S_2
\end{aligned}
$$

$$S_4 = \left(I + \Delta t \delta_\eta \Lambda_\eta\right)^{-1} S_3 \tag{2.101}$$

$$S_5 = \left(T_\eta^{-1} T_\zeta\right)^{-1} S_4$$

$$S_6 = \left(I + \Delta t \delta_\zeta \Lambda_\zeta\right)^{-1} S_5$$

$$\Delta \widetilde{Q} = T_\zeta S_6$$

This algorithm is only rigorously valid for the Euler equations. This results from the inability to diagonalize the viscous and inviscid flux Jacobians simultaneously. Therefore, the viscous flux Jacobian eigenvalues are approximated using the following,

$$\Lambda_v(\xi) = \overline{\mu J^{-1}\left(\xi_x^2 + \xi_y^2 + \xi_z^2\right)} J \rho^{-1}$$

$$\Lambda_v(\eta) = \overline{\mu J^{-1}\left(\eta_x^2 + \eta_y^2 + \eta_z^2\right)} J \rho^{-1} \tag{2.102}$$

$$\Lambda_v(\zeta) = \overline{\mu J^{-1}\left(\zeta_x^2 + \zeta_y^2 + \zeta_z^2\right)} J \rho^{-1}$$

which results in the new form of the diagonal algorithm as given by,

$$\left[T_\xi \left(I + \Delta t\left(\delta_\xi \Lambda_\xi - \delta_{\xi\xi} \Lambda_v(\xi)\right)\right) T_\xi^{-1}\right]\left[T_\eta \left(I + \Delta t\left(\delta_\eta \Lambda_\eta - \delta_{\eta\eta} \Lambda_v(\eta)\right)\right) T_\eta^{-1}\right]\left[T_\zeta \left(I + \right.\right.$$
$$\left.\left.\Delta t\left(\delta_\zeta \Lambda_\zeta - \delta_{\zeta\zeta} \Lambda_v(\zeta)\right)\right) T_\zeta^{-1}\right] \Delta \widetilde{Q} = \Delta t\left(\delta_\xi \widetilde{F}^n + \delta_\eta \widetilde{G}^n + \delta_\zeta \widetilde{H}^n - \widetilde{S}^n\right) \tag{2.103}$$

The second derivatives in Eq. (2.103) are solved through a second-order central differencing scheme. The errors caused by the approximation of the viscous terms are reduced through a dual time stepping method in conjunction with Newton-like sub-iterations, as described in Section 2.5.5.3.

2.5.5.3   Dual Time Stepping

Both time stepping methods employed in the current work necessitate approximating certain terms on the LHS of the semi-discrete form of the RANS equations. This leads to factorization errors, which must be reduced through dual time stepping, performing sub-iterations at each physical time step in the solution. A pseudo time, τ, is introduced to represent the sub-iterations as follows,

$$\frac{\partial \widetilde{Q}}{\partial \tau} + \frac{\partial \widetilde{Q}}{\partial t} + \frac{\partial \widetilde{F}}{\partial \xi} + \frac{\partial \widetilde{G}}{\partial \eta} + \frac{\partial \widetilde{H}}{\partial \zeta} = \widetilde{S} \tag{2.104}$$

Eq. (2.104) can be discretized as,

$$\frac{\widetilde{Q}^{p+1}-\widetilde{Q}^p}{\Delta\tau} + \frac{\widetilde{Q}^{p+1}-\widetilde{Q}^n}{\Delta t} - \frac{\widetilde{F}^{p+1}_{j+\frac{1}{2}}-\widetilde{F}^{p+1}_{j-\frac{1}{2}}}{\Delta\xi} - \frac{\widetilde{G}^{p+1}_{k+\frac{1}{2}}-\widetilde{G}^{p+1}_{k-\frac{1}{2}}}{\Delta\eta} - \frac{\widetilde{H}^{p+1}_{l+\frac{1}{2}}-\widetilde{H}^{p+1}_{l-\frac{1}{2}}}{\Delta\zeta} + \widetilde{S}^{p+1}_{j,k,l} \tag{2.105}$$

in which n denotes the physical time scale and p denotes the sub-iteration time scale. The first step of the sub-iterations, $\widetilde{Q}^p$, is set to $\widetilde{Q}^n$. Linearizing Eq. (2.105) around time scale n results in the following,

$$\left[\frac{1}{\Delta\tau} + \frac{1}{\Delta t} + \left(\delta_\xi\tilde{A}^p + \delta_\eta\tilde{B}^p + \delta_\zeta\tilde{C}^p\right)\right]\Delta\widetilde{Q} = -\left(\delta_\xi\widetilde{F}^p + \delta_\eta\widetilde{G}^p + \delta_\zeta\widetilde{H}^p - \widetilde{S}^p + \frac{\widetilde{Q}^p-\widetilde{Q}^n}{\Delta t}\right) \tag{2.106}$$

Introducing $h = \frac{\Delta t}{1+\frac{\Delta t}{\Delta\tau}}$ to Eq. (2.106) allows it to be rewritten as,

$$\left[I + h\left(\delta_\xi\tilde{A}^p + \delta_\eta\tilde{B}^p + \delta_\zeta\tilde{C}^p\right)\right]\Delta\widetilde{Q} = -h\left(\delta_\xi\widetilde{F}^p + \delta_\eta\widetilde{G}^p + \delta_\zeta\widetilde{H}^p - \widetilde{S}^p + \frac{\widetilde{Q}^p-\widetilde{Q}^n}{\Delta t}\right) \tag{2.107}$$

The previous equation resembles the final linearized equation as given by Eq. (2.88) and both the LUSGS or DADI methods can be employed to solve it. The unsteady residual for each time step follows as,

$$\delta_\xi\widetilde{F}^p + \delta_\eta\widetilde{G}^p + \delta_\zeta\widetilde{H}^p - \widetilde{S}^p + \frac{\widetilde{Q}^p-\widetilde{Q}^n}{\Delta t} \tag{2.108}$$

This unsteady residual should decrease towards zero over the sub-iterations to preserve accuracy. Normally, a drop in residual of one to two orders of magnitude is expected, ensuring the factorization error is smaller than the other discretization errors. If no sub-iterations are performed, Eq. (2.88) is recovered.

2.5.6 Initial and Boundary Conditions

The unsteady Navier-Stokes equations take the form of an Initial Boundary Value Problem, which entails that the development of the flow solution depends on the initial and boundary conditions prescribed to the simulation. Therefore, the primitive variables (ρ,u,v,w,p) must be assigned a starting value at every point in the computational domain. In the current study, freestream values are assigned to each point in the flow. Due to the non-dimensionalization of the RANS equations, $\rho_\infty = 1.0$, $p_\infty = \frac{1}{\gamma}$, and the velocity values (u,v,w) vary depending on the freestream Mach number. This accounts for the initial conditions prescribed to the flow, however the boundaries of the computational domain must also be studied.

Boundary conditions govern the behavior of the boundary cells within the computational domain, those cells which interact with a physical, numerical, or artificial boundary. Physical boundaries take the form of a wall where the no-slip condition must be enforced. Numerical

boundaries represent the periodicity used to shrink the computational domain from four blades to only one. Artificial boundaries are necessary due to the finite nature of the computational domain, representing the farfield conditions and imposed flow such as a sink condition. Other conditions, such as the wake cut numerical boundary used in a C-topology blade mesh, are present in OVERTURNS but not employed in the current work. Figure 2.3 below shows a 2-D cross-cut of the blade mesh, an overhead view of the background mesh, and an azimuthal view of the background mesh; detailing the boundary conditions imposed in the current work.

(a) 2-D cross-cut of blade mesh at mid-span   (b) Top-down view of computational domain
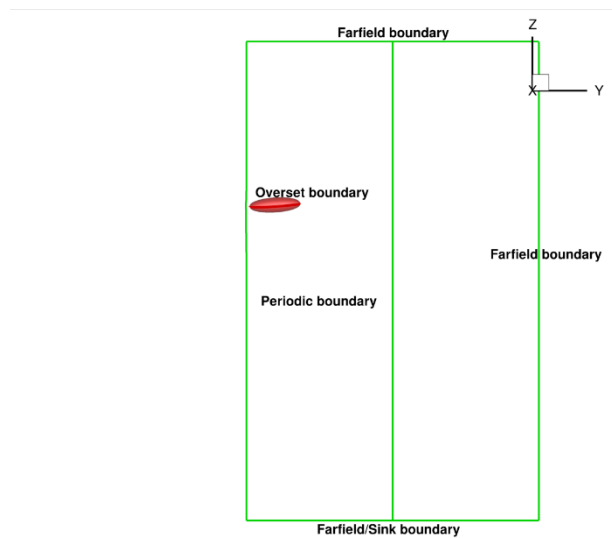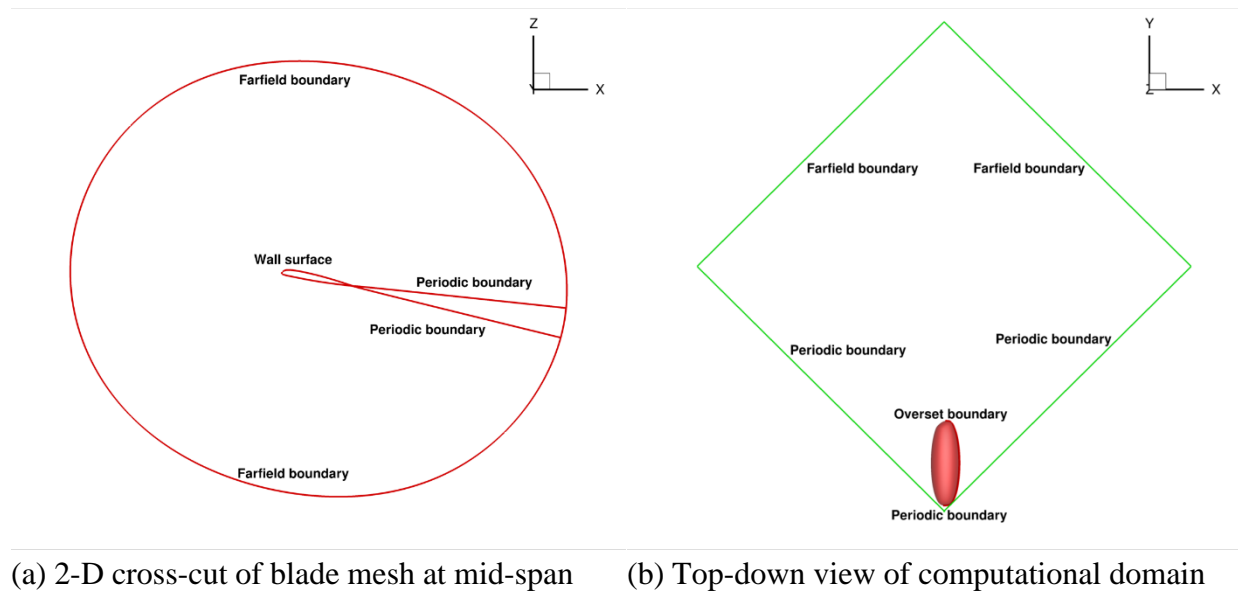
(c) Side view of blade and background meshes

Figure 2.3 An example of the blade and background meshes comprising the computational domain and the respective boundary conditions

2.5.6.1 Farfield Boundary Condition

Farfield boundaries are artificial boundaries necessary because the computational domain has a finite size. The boundaries are placed far enough from solid bodies within the flow such that no outgoing behavior is reflected back into the domain to interfere with the flow. Previous works have shown that placing the boundaries 20-30 chord lengths or more from any body surfaces ensures that no non-physical behavior is created[26]. Non-reflecting boundary conditions are achieved using Riemann invariants to propagate ingoing or outgoing waves. In regions where the flow is outgoing, the Riemann invariants are extrapolated from interior cells; while in regions of ingoing flow freestream values are extrapolated from. Due to the distance from the excited flow, numerical dissipation causes strong gradients to diminish before reaching the domain boundary. Furthermore, the mesh is stretched in the outer regions of the domain, assisting with the dissipation of strong gradients. This dissipation has the potential to negatively impact the solution unless care is taken to create a large enough computational domain to separate the majority of the flow and the outer boundaries.

2.5.6.2 Sink Boundary Condition

The sink boundary condition is a variation on the farfield boundary presented above. The primary change occurs at the bottom surface of the background mesh. Under this condition, a region of the surface is assigned a separate velocity than the freestream value employed by the farfield condition. This region sets the flow at the boundary as an outflow, pulling the surrounding flow towards the outer boundary and away from the rest of the flow. The sink is positioned directly below the rotor as the goal of the region is to simulate the movement of the starting vortex out of the computational domain, ensuring that it does not interfere with the flow. The boundary divides the lower surface of the computational domain into three regions as follows,

- The inner region of the sink, in which the prescribed velocity is constant
- The outer region of the sink, in which the prescribed velocity decreases quadratically to zero
- The region outside of the sink, where the boundary reverts to the farfield conditions

The strength of the flow due to the sink is based on the induced velocity defined as follows,

$$v_{induced} = -0.5w_\infty + \sqrt{0.25w_\infty^2 + \left(M_{tip}\sqrt{\frac{c_T}{2}}\right)^2} \tag{2.109}$$

where w_∞ is the vertical freestream velocity, M_{tip} is the tip Mach number, and C_T is a prescribed coefficient of thrust, based on the predicted thrust value of the rotor. The sink region is divided into two sections, the inner and outer region, which are given as,

$$Inner: \qquad 0 < R_{local} < \frac{\sqrt{2}}{2} R_{blade} \qquad\qquad (2.110)$$

$$Outer: \quad \frac{\sqrt{2}}{2} R_{blade} < R_{local} \leq R_{blade} \qquad\qquad (2.111)$$

in which R_{local} is the distance of the current node from the center of rotation and R_{blade} is the radius of the rotor blade. Within these regions, the prescribed velocity created by the sink boundary takes the following form,

$$Inner: \qquad v_{boundary} = 1.533 v_{induced} \qquad\qquad (2.112)$$

$$Outer: \quad v_{boundary} = \frac{2}{(3-2\sqrt{2})}\left(\left(\frac{R_{local}}{R_{blade}}\right)^2 - 2\left(\frac{R_{local}}{R_{blade}}\right) + 1\right) + 1.533 v_{induced} \qquad (2.113)$$

This prescribed velocity is added to the local velocity and propagates from the boundary inward into the computational domain.

Figure 2.4 below illustrates the effect of the sink condition on the surrounding flow. The flow field below was extracted after only 40 iterations, or ten degrees of rotation, to allow the conditions at the boundary to propagate inward but not allow the solution near the blade to develop very far. Both images display a spanwise slice of the flow at the bottom of the background mesh, and the blade is not visible in the current image. Through observing the vertical momentum, one can see that the sink condition, Figure 2.4.a, has created a strong downwash in the neighboring region. As the flow convects downward it will contact this region and be drawn downward through the lower boundary and smooth the rotor wake.

(a) Sink boundary condition
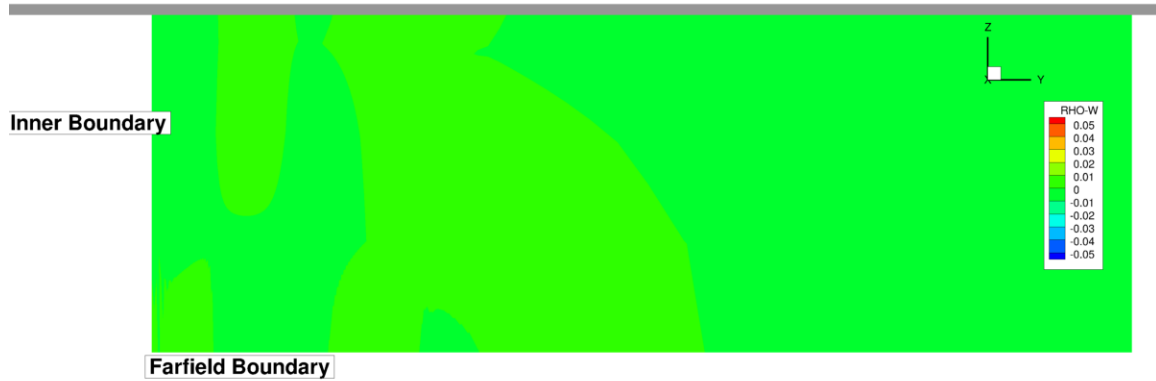
(b) Farfield boundary condition

Figure 2.4 Lower boundary of the computational domain with and without the sink boundary condition enabled, visualized using contour levels of vertical momentum

2.5.6.3   Wall Boundary Condition

At a solid wall, such as the body of the rotor blade, density is extrapolated from the interior points of the computational domain. Viscous conditions within the flow dictate that at the wall the velocity of the flow relative to the wall is zero, fulfilling the no-slip condition. With this restriction in mind, the pressure (p) is given by the normal momentum equation as,

$$p_\xi \left(\frac{\partial \xi}{\partial x_j} \frac{\partial \zeta}{\partial x_j} \right) + p_\eta \left(\frac{\partial \eta}{\partial x_j} \frac{\partial \zeta}{\partial x_j} \right) + p_\zeta \left(\frac{\partial \zeta}{\partial x_j} \frac{\partial \zeta}{\partial x_j} \right) = -\rho U \left(\frac{\partial \eta}{\partial x_j} \frac{\partial Q}{\partial \xi} \right) - \rho V \left(\frac{\partial \zeta}{\partial x_j} \frac{\partial Q}{\partial \eta} \right) - \rho W \left(\frac{\partial \zeta}{\partial x_j} \frac{\partial Q}{\partial \zeta} \right) \quad (2.114)$$

If the flow near the wall is inviscid, the velocity components of the flow (U,V,W) are extrapolated from the computational domain to the surface. Furthermore, the no-penetration condition must be enforced, ensuring that there is no flow through the solid boundary. This is accomplished by setting the contravariant component of velocity in the wall normal direction equal to the surface velocity.

2.5.6.4   Periodic Boundary Condition

Periodicity employs ghost cells to carry density, pressure, and velocity values over from the boundary of the mesh to the corresponding physical points at the far boundary of the mesh. The current work employs periodicity in two cases, at the trailing edge of the blade mesh and the inner facing boundaries of the background mesh, as illustrated above in Figure 2.5. The blade mesh contains ghost cells at the beginning and end of the mesh, allowing the third- and fifth-order spatial schemes to be employed at the edges of the physical mesh by using information calculated at the corresponding physical cell to the ghost cell.

Meanwhile, the current work assumes that the wake of a rotor in hover is periodic, allowing only one blade to be modeled before using periodicity to simulate the effects of the remaining blades. For a four bladed rotor, such as the current case, the azimuthal period is 90°. Only one blade and the corresponding background area are modeled in the computational domain, with the remaining three blades accounted for through periodic boundaries at the inner edges of the background mesh. As with the blade mesh, the outermost points in these regions are treated as ghost cells, in which the flow quantities are prescribed by transforming the data from corresponding physical cells at the other boundary.

2.6   Overset Mesh Connectivity

The computational domain employed in the current work is composed of two separate structured meshes, a blade mesh and background mesh. To ensure that data is communicated properly between these two meshes, OVETURNS contains an overset mesh capability. Overset mesh systems, also known as chimera mesh systems, denote a system of independent meshes used to model domain which requires fine flow resolution as well as a large domain where a coarser grid is preferred. The most common uses of the overset system are modeling complex geometries and capturing the rotor wake in addition to the near-body flow. The alignment and geometries of the meshes do not have to correspond, allowing a wide range of meshes to be combined in an overset system. A key component of an overset system is the data transfer connectivity between the meshes.

To transfer data between meshes, OVERTURNS employs an Implicit Hole Cutting (IHC) method developed by Lee[42] and further developed by Lakshminarayan[43] to add further boundary conditions and donor search methods. These search methods employ a stencil walk procedure developed by Gupta[44]. The IHC method takes the computed values from grid cells with a small cell size and interpolates these values to coarser cells on the other mesh. Employing the search algorithm discussed above, the method checks each cell in the computational domain, marking the smallest cell in regions of overlap. This search algorithm is discussed in detail in the works of Jose[45]. The presence of solid bodies is detected through the steadily decreasing cell sizes as a body-

52

surface is approached. A hole is cut around this solid body to ensure that flow is not modeled within these cells. Figure 2.5 below demonstrates the IHC method when observing a circular body overset with a rectangular background mesh.
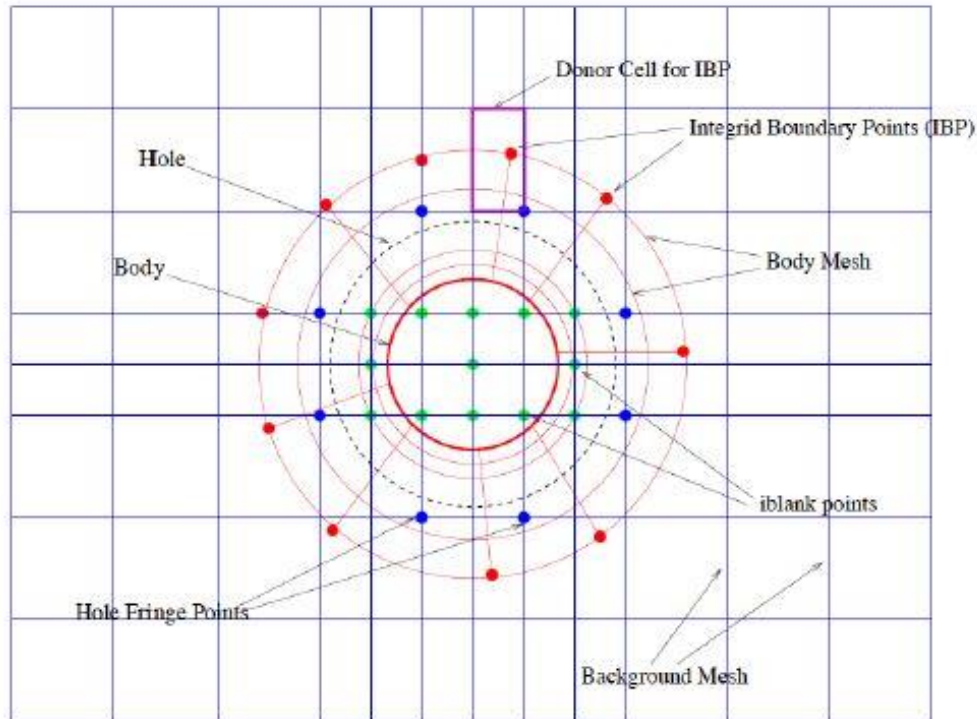
Figure 2.5 Example of IHC method and terms associated with hole cutting. Reproduced from Jose et. al[45]

The IHC methodology consists of three primary steps: 1) hole cutting, 2) marking of fringe and receiver points (integrid boundary points), 3) identification of donor cells. Figure 2.5 above illustrates this process, beginning with step one. The green points denote hole points, cut out of the background mesh by the body mesh where no flow solution is solved for. Step two determines the hole fringe (chimera) points, which receives information from the other grid. These points are denoted by the blue points in Figure 2.5, and along with the red Integrid Boundary Points, serve as receiver points which have solution values interpolated from donor cells. The donor cells, identified in step three, compute the weight of solution values from each point of the donor cell, average the solution values from each point based on the weights, and donate the information to the receiver cells. An example of a donor cell for a given boundary point is highlighted in purple in Figure 2.5. Any point which does not fall in one of these categories is referred to as a field point.

The connectivity information is stored using an Iblank array. An Iblank value of 0 corresponds to a hole point while 1 denotes a field point. This modifies the solver, causing the governing equations to not be solved at the hole points, only at the field points. An Iblank value of -1 denotes

53

a fringe or chimera point during implicit inversion, but is reverted to a value of 1 when calculating fluxes. During the time inversion, these values are blanked by multiplying the time step by the function, max(*Iblank*, 0). Multiplying the LHS of Eq. (2.107) by this value and the RHS by the function, abs(*Iblank*), ensuring that the hole points are blanked out on both sides of the equation while the fringe and chimera points are blanked only on the LHS. This prevents any inaccuracies within the hole points from adversely affecting the flow solution. Meanwhile, donor cells still transfer information to the receiver cells at the boundaries between the meshes.

2.7   Parallelization

As CFD has developed and simulations have become more complex the computational expense of the programs has also increased. Therefore, to ensure that runs are completed in a reasonable amount of time, parallel computing has been utilized to ensure that these large meshes are still solvable. The computational domain is divided into smaller areas, or sub-domains, and the solution is solved independently across multiple processors. The process of splitting the overset meshes into sub-domains is known as domain decomposition, and once split the sub-domains must be able to communicate information about the flow to one another. OVERTURNS employs the message passing interface (MPI) to support parallel computing and communication between individual processors.

The governing equations at a given time step are solved at each sub-domain, allowing full use of each available processor. Once a time step is completed at every sub-domain, solution data is passed between processors containing a common boundary. Figure 2.6 below demonstrates the domain decomposition of the blade mesh into five independent meshes. Note that the divisions occur only in the spanwise direction, as splits in the normal and chordwise directions may degrade solution accuracy. This loss in accuracy is observed when there is a strong gradient across a sub-division boundary, for example in the boundary layer where strong wall-normal gradients may be found.
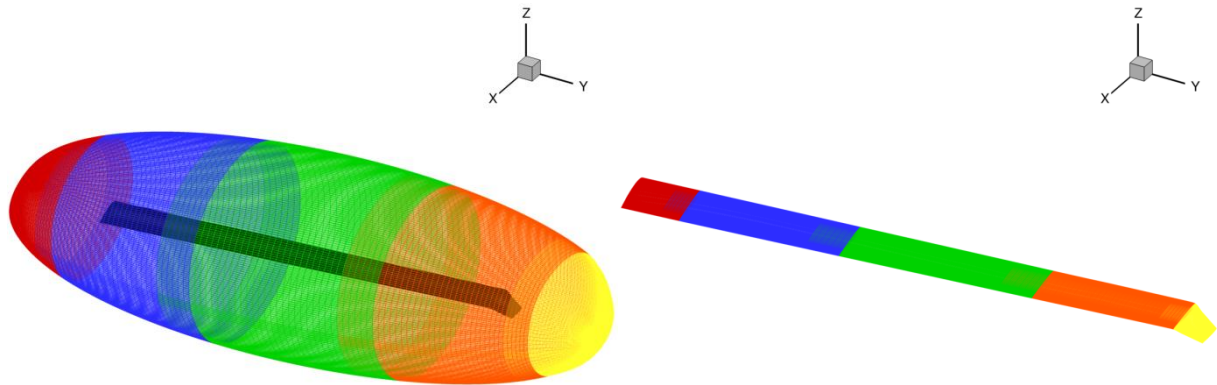
(a) Outer boundaries of the blade sub-
domains with the blade surface outlined
within

(b) Zoomed-in view of the blade surface,
highlighting each decomposition along the
span

Figure 2.6 Spanwise domain decomposition of the O-O S-76 blade mesh, where individual sub-
domains are separated by color

The mesh size of each sub-division is nearly equal, ensuring that no processor bears an abnormally large load and slows down the rest of the solution. While Figure 2.6 shows a domain divided into five partitions, the number of sub-domains can be adjusted based on the memory requirements and number of processors available. The current work employs twenty sub-divisions for the blade mesh, each of which is approximately 9 Megabytes in size. Figure 2.6.a also shows the overlap between adjacent sub-domains necessary to maintain the spatial accuracy of the solution.

2.8   Mesh Adaption

Cases involving viscous effects may require a very high density of points to accurately capture the near-body flow as well as other regions of turbulent flow. Simply increasing the number of points in the computational domain would make the computations very expensive and render the solution impossible to perform with available resources. Therefore, in addition to the overset mesh system employed, a clustering method was developed to increase grid resolution in regions of interest. Based off the work by Amiraux[46], the 3D elliptic grid generation algorithm uses a 3D Poisson equation with source terms, in which the source terms mark the regions to be refined. By prescribing a radius, skew angle, and amount of refinement, the user may alter the mesh to a certain geometry or flow pattern. As Figure 2.7 below shows, the adaption occurs first in the computational domain. This baseline curvilinear mesh is then adapted to the physical domain using tri-cubic interpolation. The elliptic equations are Laplacian in nature, ensuring a smooth refinement. Poisson equations can be computationally expensive so the current work limits itself

to clustering points around prescribed sources. Figure 2.8 below demonstrates this process, assigning two source points in a Cartesian mesh and clustering the surrounding points.
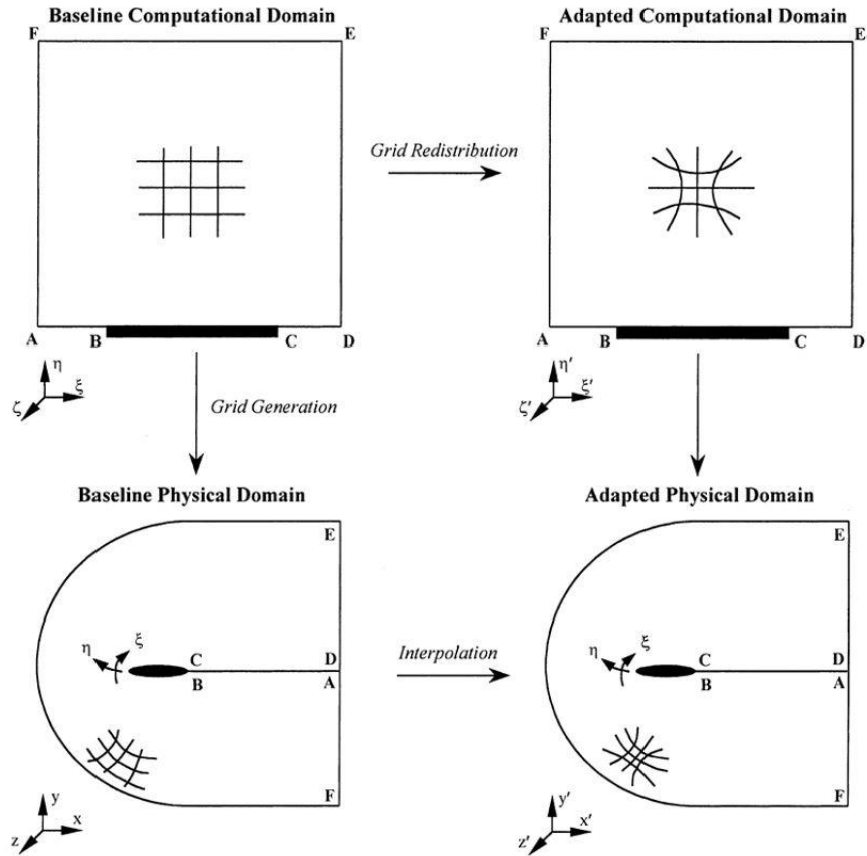

Figure 2.7 Mesh adaption process transitioning from the computational to physical domain[46]
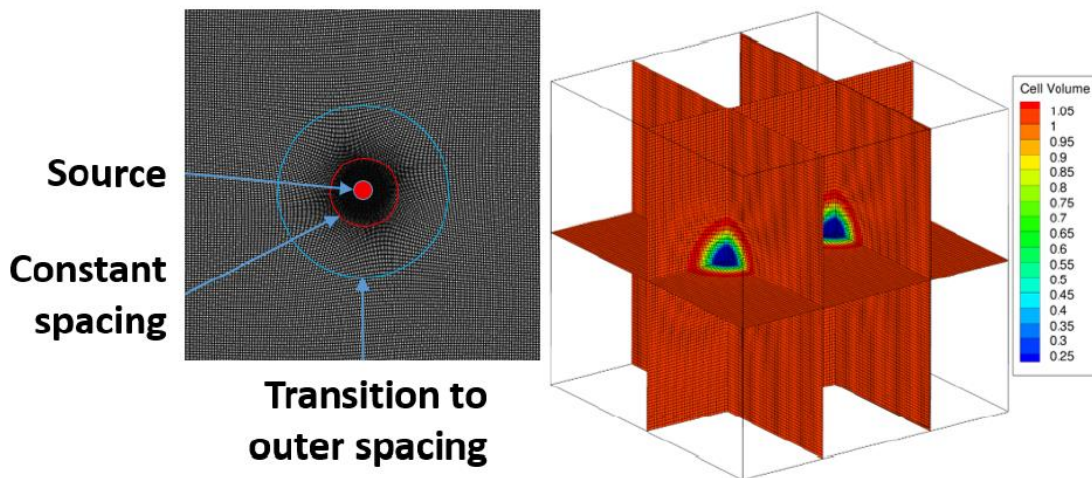

Figure 2.8 Clustering around a source term and corresponding change in mesh[46]

The general form of the numerical implementation of the 3D elliptic grid generation is presented below, and further details may be found in the work of Amiraux[46]. First the transformations between the physical and computational domains are given as the Jacobian J and Inverse Jacobian J^{-1} matrices,

$$J = \begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix} \tag{2.115}$$

$$J^{-1} = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} \tag{2.116}$$

where the inverse terms may be found through row expansion. Meanwhile the determinant gives the cell volume as,

$$\det(J) = j^{-1} = x_\xi y_\eta z_\zeta + x_\zeta y_\xi z_\eta + x_\eta y_\zeta z_\xi - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi \tag{2.117}$$

The Metric Tensor and Inverse Metric Tensors take the following form,

$$G = J^T J = g_{ij} = x_{\xi i} x_{\xi j} \tag{2.118}$$

$$G^{-1} = g^{ij} = \nabla_x \xi_i \cdot \nabla_x \xi_j \tag{2.119}$$

With these coordinate transformations the Laplace Equation in the computational coordinates (ξ, η, ζ) become,

$$\xi_{xx} + \xi_{yy} + \xi_{zz} = 0 \tag{2.120}$$

$$\eta_{xx} + \eta_{yy} + \eta_{zz} = 0 \tag{2.121}$$

$$\zeta_{xx} + \zeta_{yy} + \zeta_{zz} = 0 \tag{2.122}$$

and through inverting these equations and adding the source terms $[P\ Q\ R] = [\nabla_\xi S]C$ to produce the following Poisson Equation,

$$g^{11} x_{\xi\xi} + g^{22} x_{\eta\eta} + g^{33} x_{\zeta\zeta} + 2g^{12} x_{\xi\eta} + 2g^{13} x_{\xi\zeta} + 2g^{23} x_{\eta\zeta}$$
$$+ g^{11} P x_\xi + g^{22} Q x_\eta + g^{33} R x_\zeta = 0 \tag{2.123}$$

in which the source terms $P, Q,$ and R are composed of source terms in the computational plane multiplied with the transformation matrix as follows,

$$P = (S_{11})_\xi \xi_x + (S_{12})_\xi \xi_y + (S_{13})_\xi \xi_z + (S_{11})_\eta \eta_x + (S_{12})_\eta \eta_y + (S_{13})_\eta \eta_z$$
$$+(S_{11})_\zeta \zeta_x + (S_{12})_\zeta \zeta_y + (S_{13})_\zeta \zeta_z \qquad (2.124)$$

$$Q = (S_{21})_\xi \xi_x + (S_{22})_\xi \xi_y + (S_{23})_\xi \xi_z + (S_{21})_\eta \eta_x + (S_{22})_\eta \eta_y + (S_{23})_\eta \eta_z$$
$$+(S_{21})_\zeta \zeta_x + (S_{22})_\zeta \zeta_y + (S_{23})_\zeta \zeta_z \qquad (2.125)$$

$$R = (S_{31})_\xi \xi_x + (S_{32})_\xi \xi_y + (S_{33})_\xi \xi_z + (S_{31})_\eta \eta_x + (S_{32})_\eta \eta_y + (S_{33})_\eta \eta_z$$
$$+(S_{31})_\zeta \zeta_x + (S_{32})_\zeta \zeta_y + (S_{33})_\zeta \zeta_z \qquad (2.126)$$

Eq. (2.118-2.120) may be substituted into Eq. (2.117) to arrive at the final form of the Poisson Equation employed by this method.

The adaption is run independently from OVERTURNS, at one of two stages in the analysis:

1. In preprocessing regions where high vorticity is expected are chosen, locations in the mesh are chosen as source terms, and the mesh is clustered about these points.

2. In post processing regions of high vorticity are identified and input as source terms, around which the mesh is adapted.

In both cases the adaption process is run over a number of iterations, normally a few hundred, to ensure that the mesh remains smooth. Between iterations the virtual coordinates defining the location and strength of the source terms must be re-evaluated, as the adaption process deforms the computational domain, altering the source locations. The change in location between iterations is tracked as an error value and once the change is sufficiently small the adaption is halted. In contrast to methods such as Automated Mesh Refinement (AMR) or Vortex Tracking Grids (VTG), the current method provides refinement in regions of interest without increasing the number of mesh points and computational cost of the system. In the current work a mesh which has not been adapted using this process will be referred to as a "baseline" or "unadapted" mesh while one on which this process has been performed is a "clustered" or "adapted" mesh.

Figure 2.9 below demonstrates the usage of the adaption routine as described in the first stage above. Based on the location of the blade mesh within the baseline Cartesian mesh, the regions where the root and tip vortices are expected to form are identified and altered. No skew angle has been added in this region of the mesh, only the nodal positions have been altered to decrease the spacing in the x-, y-, and z- directions by half. Figure 2.10 displays the same level of refinement, again with no skew induced, but applied only in regions of high vorticity. Using an existing solution, a cutoff value of vorticity was chosen and regions of high vorticity were identified. These
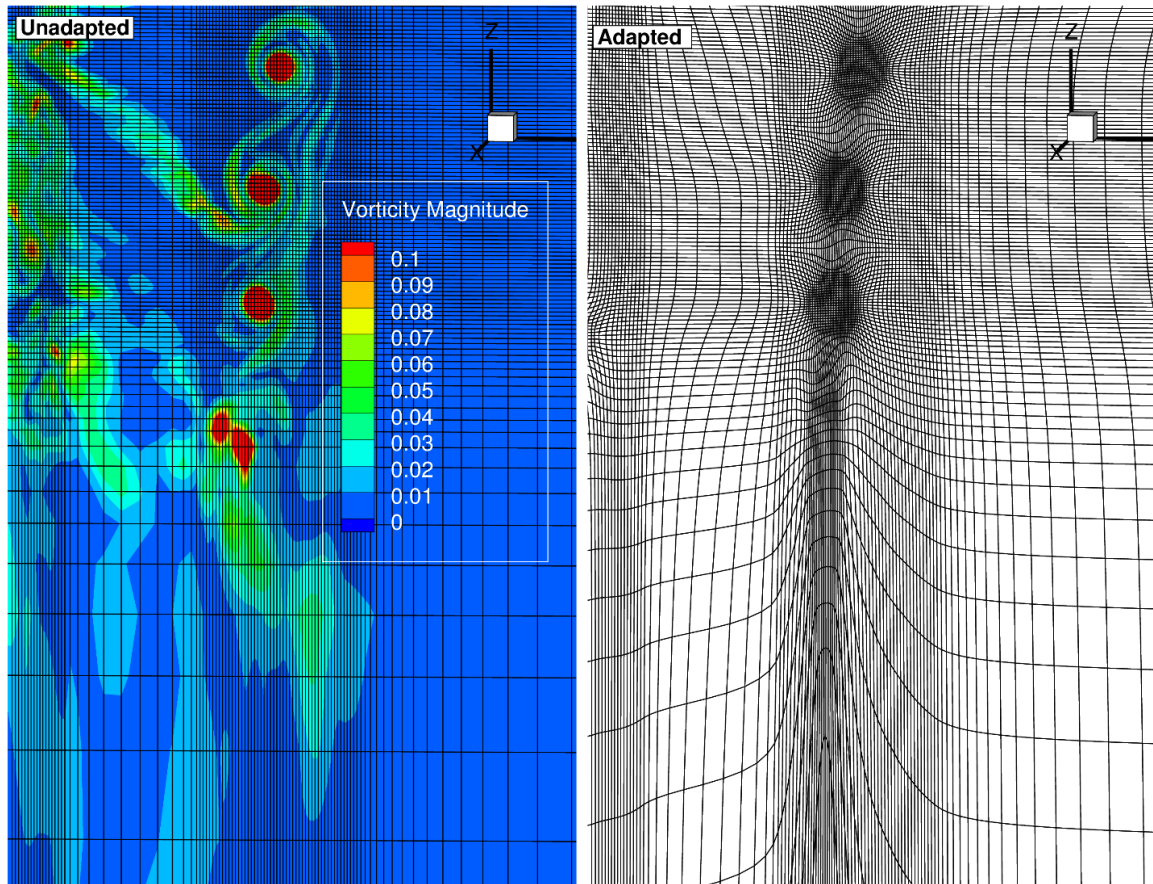
are used as the source terms for the Poisson equations, and the mesh is adapted accordingly to refine the cell spacing near these vortices.
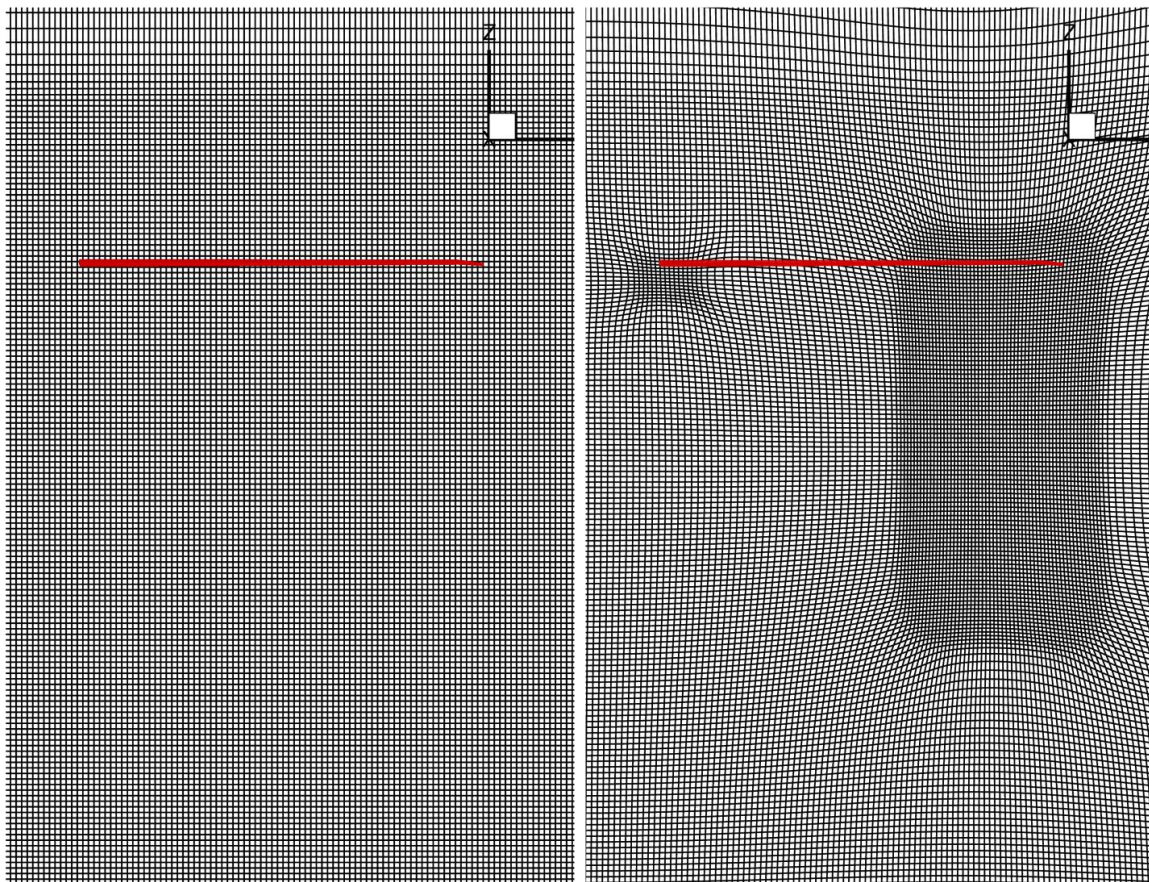
(a) Unadapted background mesh showing near blade region

(b) Background mesh with clustered regions near root and tip

Figure 2.9 Comparison of baseline background mesh and an adapted version showing the position of the blade

(a) Unadapted background mesh showing location of tip vortices

(b) Adapted mesh showing the location of clustering in tip vortex regions

Figure 2.10 Comparison of baseline and adapted meshes created using the tip vortices as sources

Comparing the adapted regions in Figures 2.9 and 2.10, one can see the potential of both approaches. The first method provides a more controlled but less precise refinement, spreading the clustering over a larger zone but with less focus on the regions of interest. The second method results in smaller regions of refinement, centered on vorticity magnitude in this example, though other values may be used. This does result in a rougher transition between the clustered and unclustered regions due to the less structured boundaries of the refined areas. This can be alleviated to an extent by restricting the refinement to the more refined regions of the baseline mesh, reducing the skew produced by stretching further spaced cells near the boundary of the computational domain. While both methods were tested in the preliminary investigation, the first method, that of a region based adaption, was ultimately chosen for the majority of the cases tested. The nature of the vorticity based method requires fully developed solutions for each individual case in order to create the adapted mesh, followed by the re-running of the simulation with the new mesh. This represents a significant investment in computational resources and did not prove feasible for the number of cases examined in the subsequent chapter.

2.9   Summary

The current chapter discussed the governing equations and numerical methods employed in OVERTURNS. Section 2.1 reviewed the governing equations of the flow, the Navier-Stokes equations, in their original form before non-dimensionalizing them in Section 2.2. The next section discussed the RANS equations, the modifications to the Navier-Stokes equations employed in the current work to model the flow solution. After mapping the physical coordinates to the grid coordinates, the spatial schemes used to reconstruct the flow variables at cell boundaries were discussed. The MUSCL, WENO, and CRWENO schemes were discussed before reviewing the second order method used to evaluate viscous fluxes. The Spalart-Allmaras model, along with the correction terms, was discussed in Section 2.5.4. This covered the turbulence modeling employed by OVERTURNS, as well as the DDES modeling employed to prevent excess turbulence from being generated. Meanwhile, the time stepping method, along with the sub-iterations employed to improve convergence are reviewed. Initial conditions were employed to begin the simulation, while boundary conditions ensure that the finite computational domain successfully mimics the expected flow. The implicit hole cutting (IHC) technique was employed to successfully transfer information between the overset meshes of the computational domain. MPI parallelization is employed to distribute the computations across multiple processors and sub-domains, ensuring that the computational load does not exceed the available resources. Finally, the mesh adaption routine employed to improve resolution in areas of active flow without additional computational expense is presented.

Chapter 3. S-76 Rotor Computational Simulations

In the following chapter, the OVERTURNS results of the S-76 rotor blade in hover are given. Multiple collective pitches have been investigated for the baseline blade and flow conditions. Additionally, the case of 10° collective pitch was chosen to inspect the effects of an altered tip shape as well as altered tip Mach numbers. Furthermore, the effects of alterations to the simulation, such as different spatial schemes and adapting the mesh, shall be presented. Results are compared to the experimental results of Balch et. al[21] as well as computational results from the Invited Hover Session at Scitech.

3.1. Computational Domain

The simulations employed an overset mesh system composed of an O-O topology blade mesh and a Cartesian wake mesh, as shown below in Figure 3.1. The implicit hole cutting method described in Section 2.6 serves to pass information between the meshes composing the computational domain. Due to the periodicity of a flow in hover, computational expenses were reduced by modeling a single blade and one quarter of the wake rather than modeling the whole four bladed system and corresponding wake. Periodic boundary conditions were employed to simulate the effects of the remaining blades.
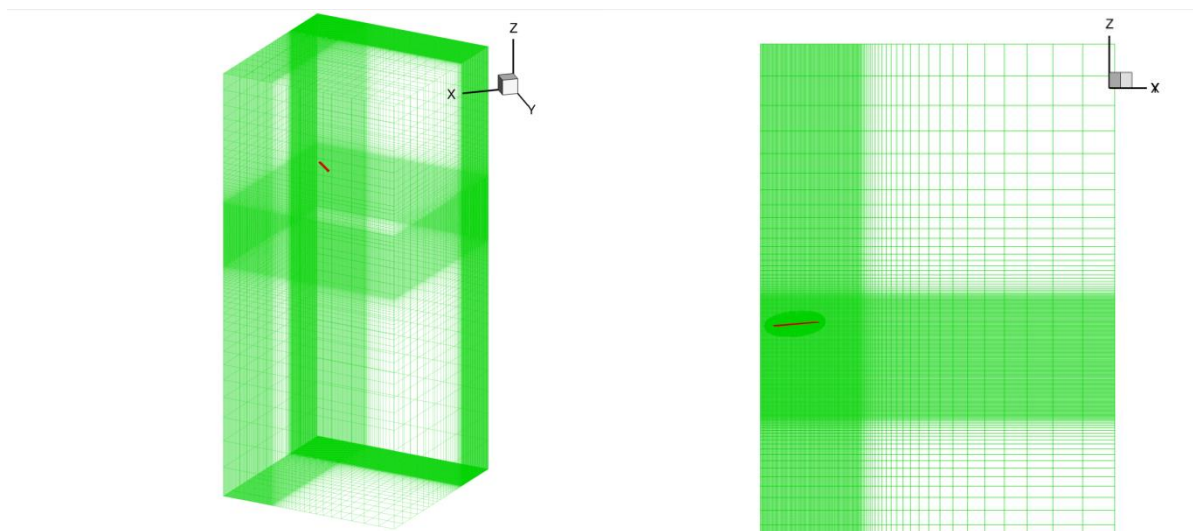
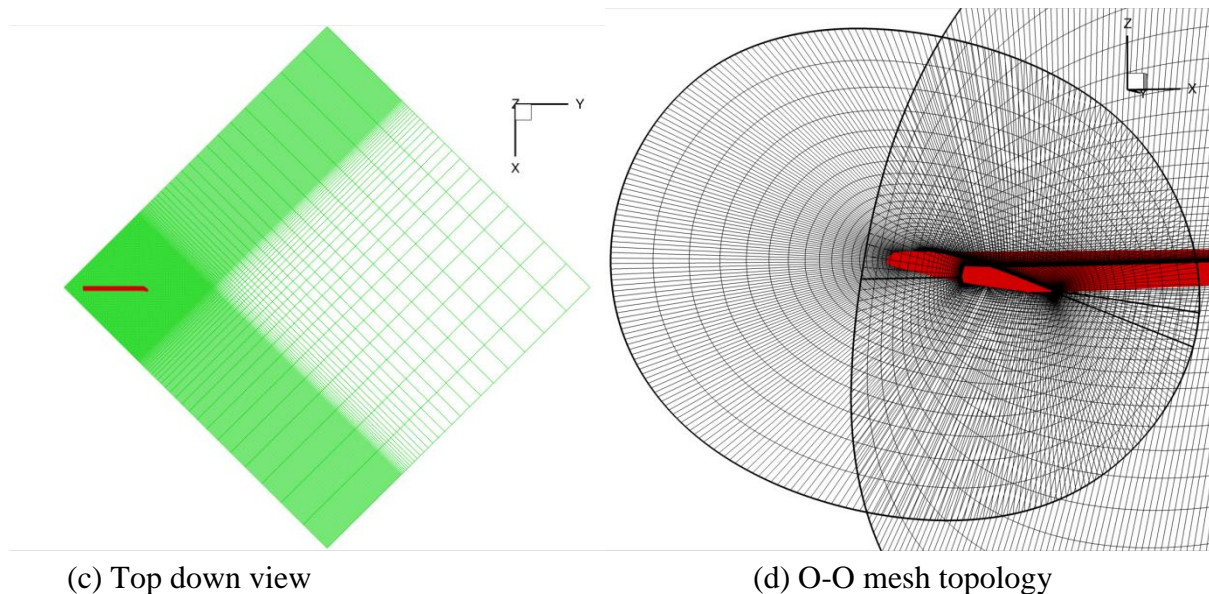(a) Computational Domain   (b) View from behind blade

(c) Top down view (d) O-O mesh topology

Figure 3.1 S-76 computational domain composed of an overset blade mesh on a Cartesian background mesh. The blade is highlighted in red and the background mesh in green

As Figure 3.1 above demonstrates, X- Y- axes are aligned with the blade mesh rather than the background mesh. This was accomplished by simply rotating the background mesh by 45 degrees, ensuring that the blade is centered within the domain. Figure 3.1.a shows an isometric view of the blade and background mesh, with b-c presenting the vertical and horizontal placements of the blade. Figure 3.1.d illustrates the blade mesh topology, as seen from behind and outboard of the root. The chordwise planes wrap around the entire blade, overlapping at the trailing edge. Meanwhile the spanwise planes extend across the upper and lower surfaces of the blade, meeting at the root and tip.

To reduce computational expense, the simulation of the 4-bladed S-76 was restricted to a single blade, employing the rotational periodicity present in the hover condition. This resulted in the quarter domain background mesh seen above, covering 90º and a single blade of the total rotor. Using the periodic boundary condition described in Section 2.5.7.3, the current work simulated the effects of the remaining blades.

The Cartesian background mesh is composed of 146 x 146 x 204 points, totaling about 3.4 million points. It extends from -0.04R (rotor radii) to 6.55R in the radial direction, along the Y-axis of the computational domain. In the chordwise direction, along the X-axis of the domain, the mesh extends to 3.3R at the corners both in front of and behind the blade. Vertically, the background mesh reaches 3.7R above the blade and -7.2R below the blade. There is no root cut-out region as would appear in most cylindrical meshes, as it is not necessary to prevent the mesh converging to a single point. The mesh is composed of 146 x 146 x 204 points in the X-,Y-, and Z- directions respectively. The finest spacing, 0.1c, occurs in a region near the blade, extending

from the edge of the domain horizontally to $1.3R$ and vertically from $0.35R$ to -1.3R. Outside of this region the cell spacing increases with distance from the blade, leading to a relatively coarse spacing of $7.0c$ near the outer boundaries.

The blade mesh consists of 293 x 161 x 76 points with an O-O topology, totaling approximately 3.6 million points. This O-O topology results in a mesh which wraps around the chordwise and spanwise directions of the blade surface, extending outwards in an ovoid shape by about $3c$ at the maximum boundaries. The finest spacing along the surface is $0.0001c$, occurring at the leading and trailing edges of the surface. At the middle of the chord, the spacing increases to a maximum of $0.01c$ at the blade surface. The largest spacing is $0.25c$ in the normal direction, near the outer boundary of the mesh. Figure 3.1.d shows a single chordwise plane overlaid with two spanwise planes, one each on the upper and lower surface. The chordwise planes collapse at the root and tip to close off the mesh, creating a fully enclosed near-body system.

In addition to the sweep of collective pitch angles tested, the effects of coning were tested at certain collective pitch angles. Once the blade was rotated to the given collective pitch, it was rotated around a just outboard of the center of rotation by an angle of 3.5º. It was then translated downward to ensure that the tip of the coned blade matched the position of the pre-coned surface. Figure 3.2 below shows the coned blade, green, compared with the original blade, shown in red.
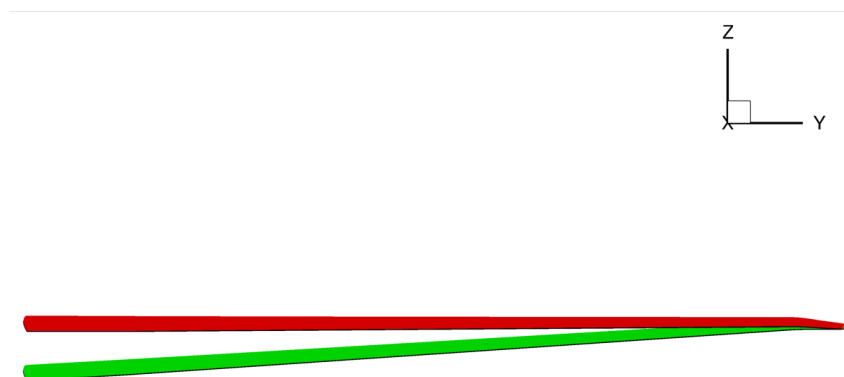

Figure 3.2 Blades at 10º collective pitch, with and without coning

3.2. Hover Condition Simulations

These simulations were performed on the swept-tapered S-76 blades. The baseline blade starts to taper at 95% of the rotor radius. This taper includes a 35° degree leading edge sweep and taper such that the tip chord is 60% of the root chord and varies linearly along the outer 5% of the blade. Figure 3.3 below shows the tip regions of the blade, illustrating the region where the blade transitions from a straight to tapered configuration. The blade contains a linear 10° twist which persists along the length of the blade.

The flow conditions for each run were identical, only changing the collective pitch angle between runs. The runs were conducted using the following flow conditions:

- Standard atmosphere and temperature
- Tip Mach Number = 0.65
- Reynolds Number = 1.1697 x 10^6
- Rotor Rotation = 1483.9 RPM

The step size employed was a 1/4 ^{o}/step, resulting in 1440 iterations per rotor revolution. All runs employed the Spalart-Allmaras turbulence model discussed in Section 2.5.4. All cases were run until 10 rotor revolutions, at which point the integrated flow values had steadied. At this point in the solution, the starting vortex had convected downstream and the rotor wake had fully developed.
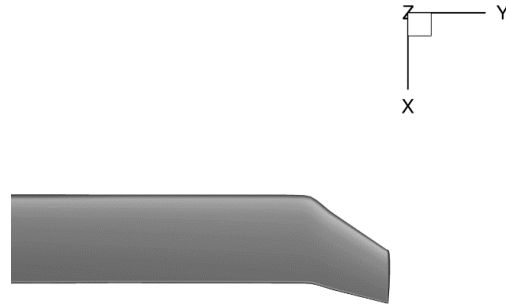
Figure 3.3 Planform of swept-tapered tip shape for S-76 rotor

3.2.1. 10° Collective Pitch Testing

Prior to running the full sweep of collective pitches, the author determined the optimal set of run conditions and OVERTURNS parameters for the current data set. To accomplish this, the swept-tapered blade at a 10° collective pitch setting was chosen as a starting point. This data set was chosen because there are many results from other solvers, as well as the experimental data, for this setting to verify and compare the results of OVERTURNS to. The current section will provide an overview of the initial testing to determine the most appropriate run conditions followed by a more detailed analysis of the results for the 10° case using the final configuration.

The following list details the various run and solver settings which were tested for the current work:

- Cylindrical versus Cartesian background mesh
- 4-bladed simulation versus 1-bladed simulation using periodic boundary conditions
- Unadapted Cartesian background mesh versus pre-adaption of the background mesh in regions of expected high vorticity
- Comparison of three spatial schemes, the 3rd order MUSCL scheme, 5th order WENO scheme, and 5th order CRWENO scheme

- Farfield boundary conditions at the edges of the computational domain versus sink boundary conditions at the edges of the computational domain

All of these tests were run using the same flow conditions and options in OVERTURNS apart from where noted. Each case was run on the Deepthought 2 high performance computing cluster run by the University of Maryland, College Park. A few of the different run environments produced significant changes in run time, which was measured by wall clock and not total computing time across all nodes.
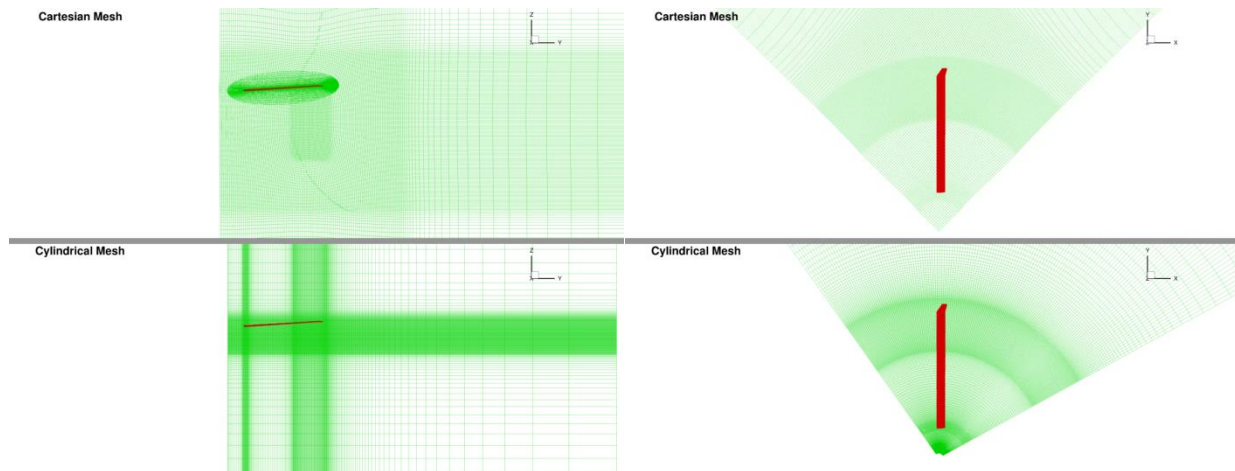
3.2.1.1. Cylindrical and Cartesian Background Mesh

The first priority in the current work was to decide on the type of background mesh to employ. The two options reviewed by the author were a cylindrical background mesh, as shown below in Figure 3.4, or a Cartesian style mesh, as shown previously in Figure 3.1 and below. The Cartesian mesh contains 146 x 146 x 208 points for a total of approximately 3.4 million mesh points while the cylindrical mesh consists of 140 x 235 x 220 mesh points totaling 7.2 million nodes. As discussed in the previous section, the Cartesian mesh extends a total of $6.55R$ in the radial direction, $3.3R$ in front of and behind the blade, $3.7R$ above the blade, and $-7.2R$ below the blade. Meanwhile, the cylindrical mesh covers an arc of 90° with the blade at the center, extending in the radial direction out to $4.0R$, $3.0R$ above the rotor plane, and $-4.5R$ below the rotor plane. The Cartesian mesh offers a finest spacing of $0.1c$ in the clustered regions near and below the root and tip. The cylindrical mesh is also refined near the root and tip regions to a spacing of $0.02c$, though this fine spacing does not extend as far from the blade as the finest region of the Cartesian mesh.

While the cylindrical mesh provides a more refined mesh in the finest region, it presented a number of issues compared with the Cartesian topology. The cylindrical mesh covers a smaller area than the Cartesian mesh while containing more than double the number of nodes. The computational domain could be increased without sacrificing resolution by adding more points to the mesh, but this has the potential to increase the computational load beyond what the available resources can support. Alternatively, resolution could be sacrificed to expand the computational domain without increasing the number of points, though this leads to issues with cell size. As one moves away from the refined region, the size of the cells increases rapidly, with the largest cells occurring at the edges of the domain. Figure 3.4.b shows this in the vertical direction, where the cylindrical mesh sees a more rapid increase in spacing than the corresponding region of the Cartesian mesh. A decrease in resolution may exacerbate this problem, threatening the stability of the solution.
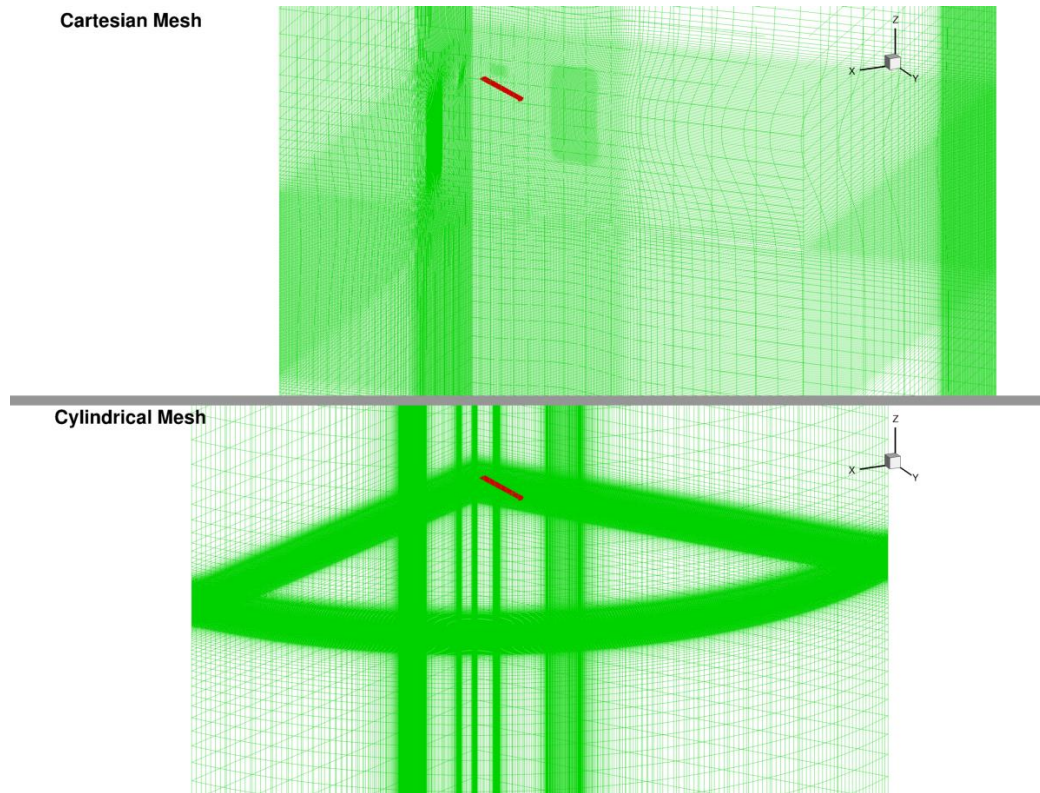
(a) Outboard view of Cartesian (top) and cylindrical
(bottom) meshes

(b) View from behind the blade, showing a   (c) View from above the blade, showing a horizontal
spanwise plane of the background mesh   plane of the background mesh

Figure 3.4 Comparison of the tested Cartesian (top images) and cylindrical (bottom images) background meshes, showing the position of the blade mesh in each domain

Cases were run on both meshes, using the same run conditions apart from the background meshes and corresponding changes to connectivity and boundary conditions. The results after 10

rotor revolutions are presented as follows. Table 3.1 below compares the integrated forces results for the same rotor conditions after 10 revolutions.

	Cylindrical Background Mesh	Cartesian Background Mesh	Experimental Values ($\theta = 9.8°$)	Largest Percent Difference (%)
C_T	0.0061558	0.0065121	0.00649	5.15
C_Q	0.00062325	0.00064072	0.00054	18.7
FM	0.54796	0.57996	0.68467	20.0

Table 3.1 Comparison of integrated forces values between cylindrical and Cartesian background cases and measured values

Table 3.1 shows the predicted integrated thrust and power values, as well as the corresponding Figure of Merit of the current case, compared to the nearest available experimental values. The rightmost column compares the largest difference between the computational and experimental results, not necessarily the difference between the computational setup that was chosen for further testing and the experimental results. The computed integrated forces values are close to one another, with the largest difference resulting from the thrust value and carrying over to the FM measurement. These baseline results demonstrate that both background meshes produce similar results when looking at the performance metrics of the rotor and stability of the solution. Both solutions over predict the power requirements when compared with experimental values, leading to an under prediction of FM. The cylindrical mesh predicts a lower value, but the Cartesian solution has still under predicted FM. This under prediction is discussed further in the collective sweep and conclusion section of the current work. Now one may examine the flow fields seen in each case to determine which background mesh is suitable for the current work.

Figure 3.5 below shows a comparison of the flow fields, visualized using vorticity magnitude, after 10 rotor revolutions. The spanwise slice shows a range of vorticity magnitudes from 0 to 0.1 while the iso-surfaces below trace the areas with a vorticity magnitude of 0.1. Comparing the two results, one can see that the cylindrical mesh has not preserved the vortices as well as the Cartesian mesh. Figure 3.5.a shows that two rotor passages have been clearly preserved, with the third passage being smeared out. Meanwhile, Figure 3.5.b shows five passages, each distinct from the others and more coherent than those seen in the cylindrical results. This is a result of the more rapid loss of resolution in the cylindrical mesh when compared with the Cartesian mesh. The vortices remain stable in the fine region, yet as they propagate downward the mesh becomes too coarse to maintain these structures and the vortices collapse as a result.
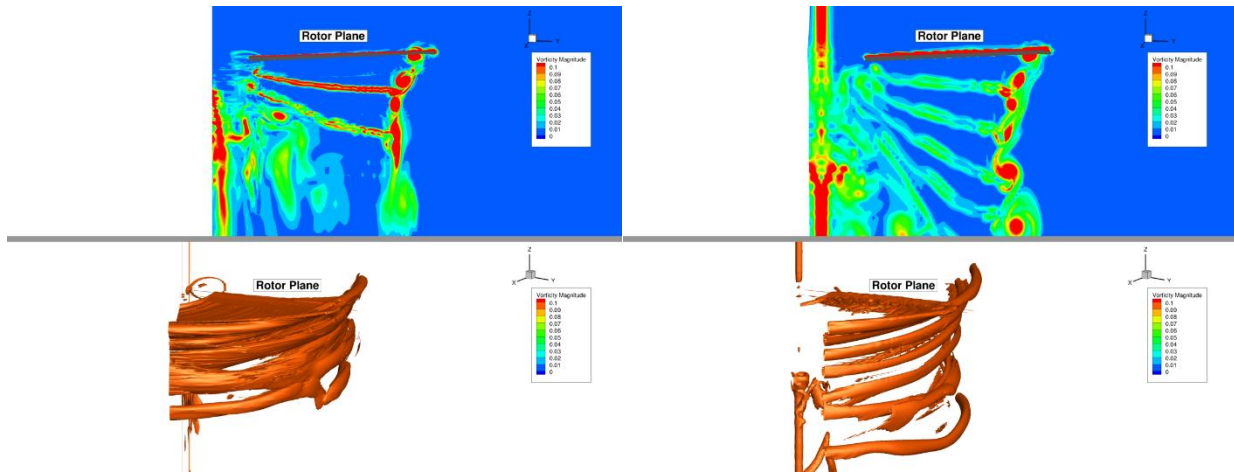
(a) Cylindrical background mesh   (b) Cartesian background mesh

Figure 3.5 10 rotor revolution flow fields for the swept-tapered blade at 10° collective pitch

Comparing the two potential background meshes, the current author determined that the Cartesian mesh was better suited to the current work. Both meshes produced comparable integrated forces results as well as converging to the same order of magnitude, indicating similar stability. However, the two meshes deviated in the flow fields they produced. The cylindrical mesh did not capture as many rotor passages as the Cartesian mesh, nor did it maintain the same stable vortices. Despite these differences, the blade forces were similar in each solution because the flow in the rotor wake does not affect the near-body flow of the blade to a large extent. Because one of the stated goals of the current work is to improve vortex capturing, the Cartesian mesh was chosen for use in further testing.

3.2.1.2.   4-Bladed Simulation versus 1-Bladed Simulation

Following the choice of the Cartesian mesh for the current work, the decision had to be made whether to run a full 4-bladed simulation with no periodic boundary conditions or a 1-bladed simulation using periodic boundary conditions to simulate the remaining blades. The full background mesh, suitable for all 4 blades, was created first and the 1-bladed background mesh was created by partitioning the full background mesh into quarters. Meanwhile, the same blade mesh was used to create all 4 blades, simply rotating the mesh as needed to properly position the blades. Figure 3.6 below shows the original 4-bladed domain in red with the blades positioned near the center at 90° to each other. The smaller, green region denotes the domain employed when only simulating a single blade. In this case, the quarter background mesh was rotated by 45° to align the blade with the center of the background mesh. The reason for rotating the background mesh rather than the blade mesh was to keep the blade mesh aligned with the Y-axis for ease of reference.
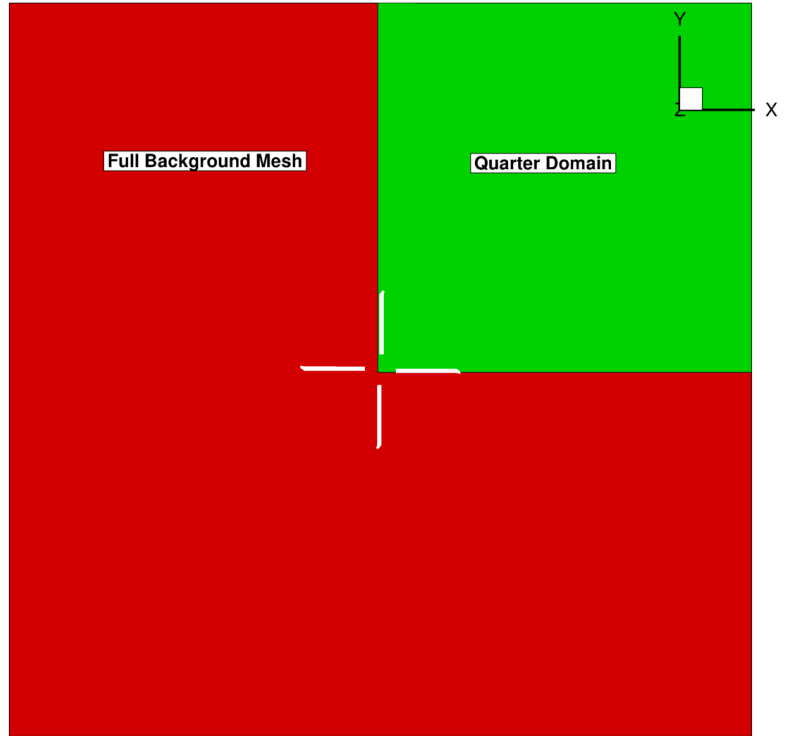
70

www.manaraa.com

Figure 3.6 Overlay of full computational domain as well as the quarter domain used for simulating only one blade

The 4-bladed simulation presents a far greater computational requirement than the corresponding 1-bladed simulation employing periodic boundary conditions. As there are three additional blades being simulated, the number of points from just the blade meshes has quadrupled from 3.6 million points to 14.2 million points, larger than the total domain for the 1-bladed simulation which totals 7 million points. Meanwhile, the size of the background mesh has nearly doubled in both horizontal directions, comprising 285 x 285 x 204 nodes for a total of 16.6 million mesh points. Therefore, the computational domain for a 4-bladed simulation totals 30.8 million points, more than four times the size of the 1-bladed working space.

The larger computational domain requires a corresponding increase in computational resources. The quarter domain cases were run across 40 cores on the Deepthought 2 cluster, with both the blade and background meshes split into 20 partitions. To achieve subdomains of comparable size to those of the quarter domain, the full domain was split into 100 partitions, with the blades split into 12 larger partitions per blade and the background mesh employing the remaining 52 nodes. The HPC cluster contains 20 cores per node and allocates the entirety of a node to a job, hence the desire to restrict the subdivisions to multiples of twenty. The subdomains of both the quarter and full background meshes were similar in number of points, meaning that there was not a large difference in computational time between the two cases. However, the full domain used five times the number of CPU hours that the quarter domain employs, which presented a problem given the

71

finite resources available to the current author. Thus, the quarter domain was chosen to conduct further runs by ensuring that the maximum number of tests could be performed.

3.2.1.3.   Unadapted Cartesian Background Mesh versus Pre-adapted Mesh

Once the 1-bladed, Cartesian background mesh was chosen for testing, the final parameters of the mesh had to be decided upon. The outer boundaries of the mesh had already been determined to ensure that the farfield boundaries would not interfere with the simulation. To accomplish this without an unreasonable number of nodes the mesh becomes less refined as the boundary is approached, with the greatest refinement found near the rotor blade. This is in keeping with CFD conventions and is not unexpected when working with structured meshes. It was the near-body region, the finer regions visible in Figure 3.1, where the majority of the turbulent flow occurs which presented an opportunity for improvement.

To improve the quality of the solution, the current author decided to cluster the mesh in the regions of expected high vorticity, creating a pre-adapted mesh on which to compute the solution. As stated earlier in the work, the goal of the alteration was to improve mesh resolution in certain regions without adding additional points to the computational domain. To achieve this, the mesh adaption routine discussed in Section 2.8 was altered to modify a chosen region, rather than identifying and altering regions of high vorticity. Figure 3.7 below shows a comparison of the unadapted and adapted Cartesian meshes, with the two further refined regions visible near the root and tip of the blade. The left-hand images illustrate the change in vertical spacing of the background mesh while the right-hand images show the radial change.
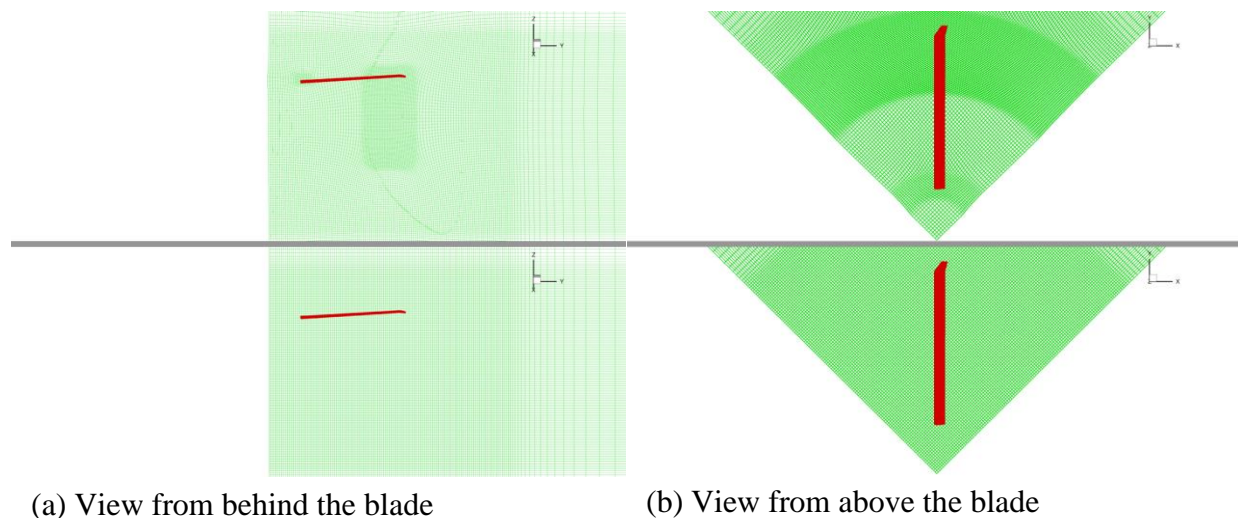

(a) View from behind the blade   (b) View from above the blade

Figure 3.7 Comparison of Cartesian background meshes before and after further clustering near the root and tip

The near tip region extends from $10c$ to $22c$ in the radial direction and from $4c$ above the blade to $16c$ below it, as seen above. The smaller region near the root covers only $2c$ to $5c$ in the radial direction and $-1.6c$ to $1.6c$ in the vertical direction. In both regions the mesh is clustered to half the spacing of the unclustered region. The current work uses an adapted mesh created from the quarter domain shown in Figure 3.6 but the method is just as easily applied to the 4-bladed domain. Note that this version of the background mesh is identical to that presented above in Section 3.2.1.1 in addition to being the final choice when running the collective sweep of the rotor.

The advantage of the pre-adaption is that it does not require an existing solution to create a finer mesh, only a prediction as to where the majority of flow structures will appear. The finer resolution is accomplished by stretching the mesh in other locations, necessitating careful monitoring of the process to ensure the mesh is not rendered unusable elsewhere in the domain. Here the region based adaption has an advantage over the source based results. The large, continuous regions provide a smoother transition from the unclustered to clustered regions than the small, scattered regions such as the example in Figure 2.7. Hence, the mesh remains closer to its original layout, reducing the possibility of resulting errors in the solution.

The current author chose to first test this adapted mesh against the unadapted Cartesian mesh, using the same test conditions discussed previously. Table 3.2 below presents the integrated forces after 10 rotor revolutions. As seen when comparing the cylindrical and Cartesian meshes previously, there is not a large gap in predicted performance between the two meshes. The computational integrated thrust and power values differ by just under 5%. Again the computational results over predict the power requirement and will be discussed further in later sections.

	Unadapted Cartesian Background Mesh	Adapted Cartesian Background Mesh	Experimental Values ($\theta = 9.8^{\circ}$)	Largest Percent Difference (%)
C_T	0.0068380	0.0065121	0.00649	5.36
C_Q	0.00067129	0.00064072	0.00054	24.31
FM	0.59562	0.57996	0.68467	15.29

Table 3.2 Comparison of integrated forces values between original Cartesian and adapted Cartesian background cases and measured values

Figure 3.8 below shows the corresponding flow fields for this test. As with the previous cases the solution is presented below at 10 rotor revolutions, showing a spanwise slice with vorticity magnitude contours as well as an iso-surface tracing the regions with a strength of 0.1. Observing the spanwise slice, one can see that even the unadapted Cartesian mesh captures the wake better than the cylindrical mesh seen in Section 3.2.1.1. Both of these meshes have preserved multiple blade passages of the tip vortex and capture the structure of the wake. The root behavior shows a larger difference, with the unadapted mesh showing regions of higher vorticity than the clustered

73

case but lacking the high-vorticity regions near the center of rotation. While the flow near the root has changed, the flow near the blade tip illustrates the change between each case.
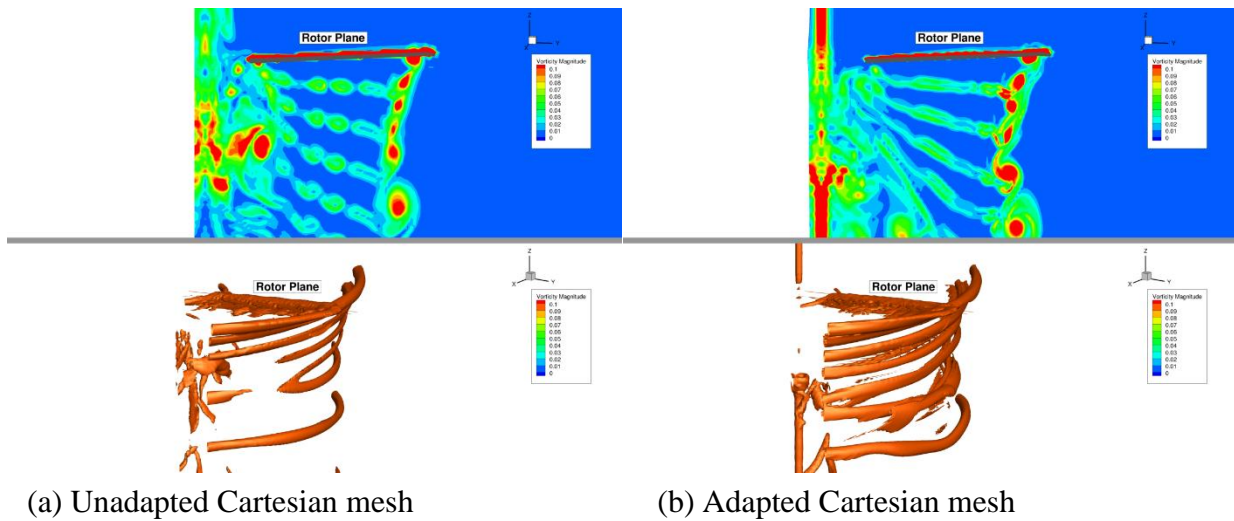

(a) Unadapted Cartesian mesh   (b) Adapted Cartesian mesh
Figure 3.8 Flow fields of the unadapted and adapted Cartesian meshes at 10 rotor revolutions

The flow near the tip provides a clearer difference between the two meshes. Viewing the spanwise slices, one can see that the vortices in the adapted background mesh are larger and have convected downwards slightly further than those in the unadapted mesh. Furthermore, the iso-surfaces show that the vortices in the adapted case are stronger than those in their unadapted counterpart. The contraction of the adapted case is also more pronounced than that of the unadapted mesh, as seen in the location of the vortex cores in the spanwise slices.

The two background meshes tested here produced comparable rotor performance values but differed in the preservation of the rotor wake. The unadapted mesh saw quicker deterioration of the wake, as evidenced by the reduced number of blade passages displayed by the iso-surface visualization. Furthermore, the adapted mesh better shows the contraction present in the rotor wake. The results of this section mirror those seen when comparing the cylindrical and Cartesian background meshes, in which both produce similar blade forces but differ in the preservation of the rotor wake. In accordance with the goals of the current work, the accurate modeling of the rotor wake is of utmost importance and so the adapted Cartesian mesh was chosen to be employed in further testing.

3.2.1.4. Comparison of Spatial Schemes

Having finalized the computational domain, the current author determined which of the available reconstruction schemes in OVERTURNS was best suited to the current work. These numerical schemes govern the convection of flow variables from cell to cell within the computational domain. OVERTURNS contained the MUSCL and WENO schemes previously and the current author added the CRWENO scheme to the current version of the program. Section 2.5.1 discussed each of these schemes in detail, so the current section will focus on the testing and performance differences between each method.

As with the previous tests, each scheme was tested on the computational domain discussed above and run to 10 rotor revolutions before comparing the results of each case. The MUSCL scheme, a three point scheme providing third order accuracy in regions of smooth flow and decreasing to first order in the presence of discontinuities, served as the starting scheme. By averaging cell values from each stencil point, the interface values are calculated. Meanwhile the WENO and CRWENO schemes are fifth order schemes, employing additional points and using a weighted approach when averaging the flux values. Both schemes provide fifth order accuracy in smooth regions which is reduced to third order accuracy in regions with discontinuities. The compact version employs compact, lower-order stencils to calculate the weight terms, whereas the non-compact method calculates the weights directly using higher order terms.

Table 3.3 below compares the results between each scheme, using identical testing conditions apart from the scheme employed and run to 10 rotor revolutions. As with the previous tests, the integrated forces do not change a great deal between runs. Each algorithm produces comparable thrust and power values, with the largest difference in computational results being slightly more than 2%. Each computational case over predicts power requirements, leading to an under prediction of FM as seen before. Finally, the run time of the solution using each method was compared, with the wall time increasing along with the complexity of the method. Given the additional calculations necessary for the higher order methods, this was an expected outcome and did not prohibitively increase the required resources.

	MUSCL	WENO	CRWENO	Experimental Values (θ = 9.8°)	Largest Percent Difference (%)
C_T	0.006599	0.0064550	0.0065121	0.00649	1.68
C_Q	0.00064263	0.00063218	0.00064072	0.00054	19.01
FM	0.58984	0.58008	0.57996	0.68467	15.29
Wall time/rev (hrs.)	4.25	5.00	5.5		

Table 3.3 Comparison of integrated forces values between three spatial schemes and measured values

Figure 3.9 below shows a spanwise slice and iso-surface trace of the vorticity magnitude at 10 rotor revolutions. Figure 3.9.a shows the result of the 3rd order MUSCL scheme, 3.9.b shows the 5th order WENO scheme flow, and finally 3.9.c visualizes the solution attained with the 5th order CRWENO scheme. There is a significant difference between the MUSCL scheme and the two 5th order schemes, as evidenced in 3.9.a. The spanwise slice shows that the vortex strength has quickly dissipated, maintaining very few blade passages yet maintaining a large yet dissipated vortex structure at the bottom of the wake. The iso-surface supports this, showing only two passages below the rotor plane which maintain a strength of 0.1 or greater. The two 5th order solutions, b and c, show much more agreement and more preservation of the vortex structures. The WENO solution shows significantly more behavior inboard of the root, which may slow the stabilization of the solution. Meanwhile, in the tip region both schemes show a clear wake structure, though the CRWENO case has convected further downward and shows slightly more curvature than the WENO result.

(a) MUSCL scheme   (b) WENO scheme
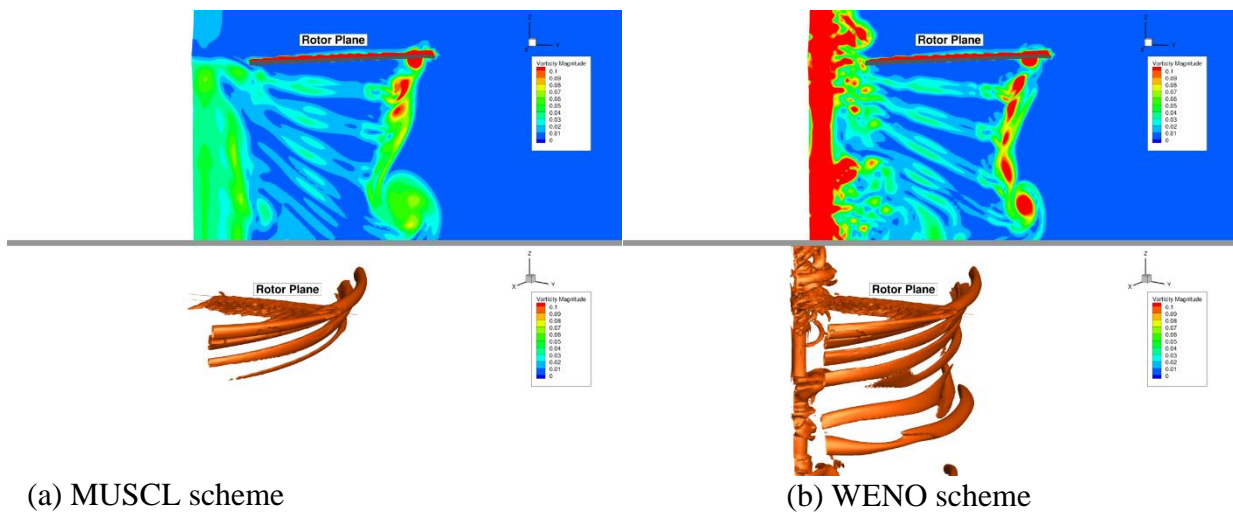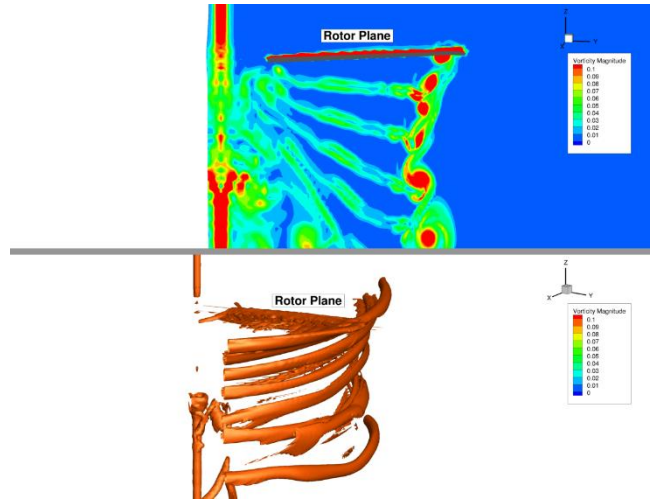
(c) CRWENO scheme

Figure 3.9 Flow fields of three separate numerical algorithms after 10 rotor revolutions, tested under identical conditions

Having viewed the numerical and flow field results of each scheme, one was chosen for use in further testing. All three showed good agreement of predicted blade forces, so these results did not affect the decision of which scheme to employ. The higher order methods necessitated an increase in run time but did not increase computational requirements enough to make using the WENO or CRWENO schemes unfeasible. The flow fields showed a significant gap in the preserved rotor wake between the third order MUSCL scheme and both fifth order schemes. As one of the primary goals of the current work is the resolution of the wake structure, the MUSCL scheme was discarded as an option for the current work. The more chaotic flow inboard of the root present in the WENO solution, as well as the slightly more smeared out vortex cores present in the rotor wake, resulted in the CRWENO algorithm being chosen to conduct the full range of testing with.

3.2.1.5. Comparison of Farfield and Sink Boundary Conditions

The final parameter to test before conducting a full sweep of pitches were the boundary conditions employed at the edges of the computational domain. Section 2.5.6 discusses the boundary conditions present in OVERTURNS and Figure 2.3 is presented again below to aid in the current discussion. All of the boundary conditions within the blade mesh are unchanging, with only a single correct condition for each boundary location. The blade surface itself is defined as a solid surface, the overlapping points at the trailing edge use periodic conditions, and the outermost boundary uses a farfield boundary, receiving information from the background mesh. Meanwhile, the background mesh also contains some boundaries with set conditions but also provides the option to alter parts of the outer boundary. The overset boundary connects the background mesh to the blade mesh while the periodic boundaries transmit the information from one inner boundary to the other and simulate the results of the remaining blades. The bottom boundary of the

77

computational domain, visible in 3.9.c, may be assigned as either a farfield boundary or a sink boundary.

(a) 2-D cross-cut of blade mesh at mid-span   (b) Top-down view of computational domain

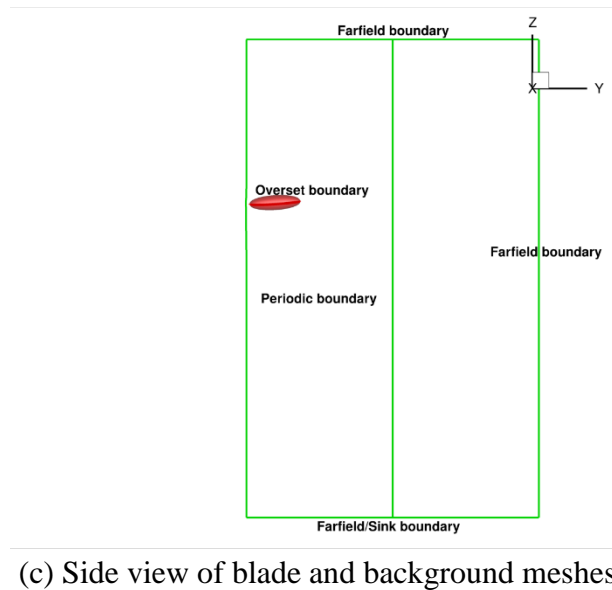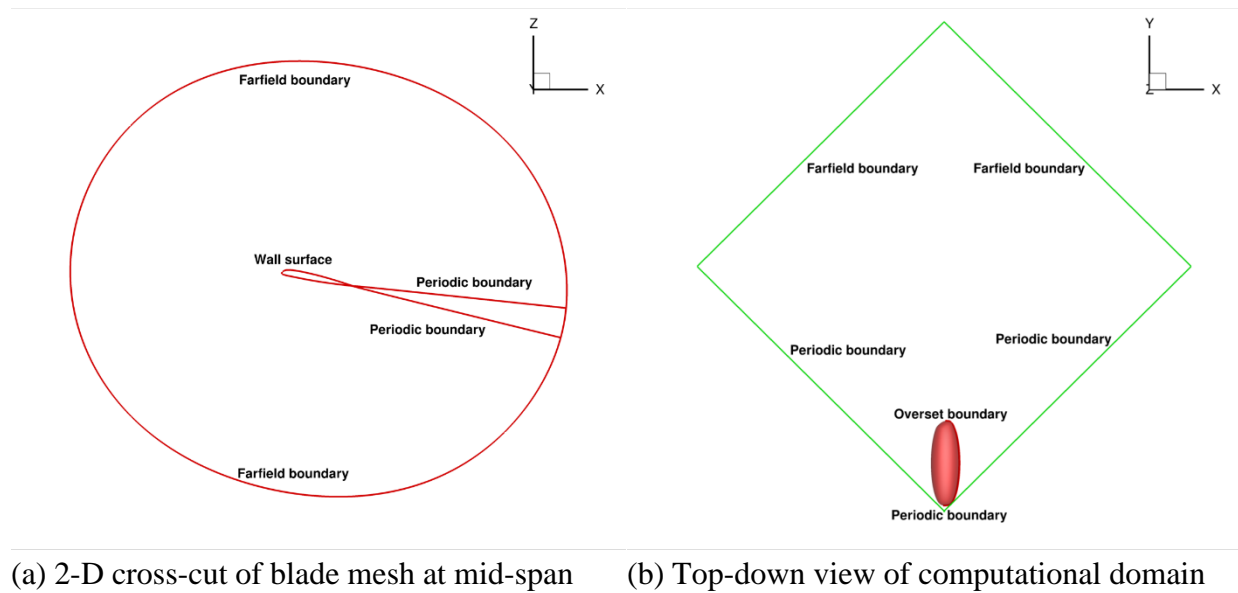(c) Side view of blade and background meshes

Figure 3.10   An example of the blade and background meshes comprising the computational domain and the respective boundary conditions (Duplicate of Figure 2.3)

As discussed in Section 2.5.6.2, the sink boundary condition is designed to smooth the flow by drawing some of the initial flow downwards out of the computational domain. The startup of the rotor creates a large starting vortex which propagates downward, and parts of which may roll up back into the plane of the rotor and disrupt the near-body flow. The strong turbulent flow from the start up vortex should merge with the region of downward flow created by the sink condition,

drawing it out of the computational domain. The sink condition requires a C_T value to gauge the strength of the boundary, so the value of 0.063 from the experimental data of Balch et. al[21] was used. Apart from the alteration to the lower boundary the parameters of each run were identical and the results after 10 rotor revolutions are presented below.

Table 3.4 below presents the integrated forces and attained from the current cases. The computational thrust and power coefficients are very close to one another, differing by less than one percent. As a result, the predicted Figure of Merit values are also very similar, indicating that the sink has not affected the blade forces by any significant amount. Meanwhile the relation to the experimental values match the previous over prediction of power requirements. Overall, these results demonstrate that the sink condition has not greatly affected the flow near the blade, as the altered flow from the boundary has not propagated far upwards into the flow.

	Farfield Boundary	Sink Boundary	Experimental Values (θ = 9.8º)	Largest Percent Difference (%)
C_T	0.0065121	0.0065689	0.00649	1.22
C_Q	0.00064072	0.00064338	0.00054	19.14
FM	0.57996	0.58513	0.68467	15.29

Table 3.4 Comparison of integrated forces values between standard farfield boundary conditions and the sink boundary condition at the bottom of the computational domain and measured values

Figure 3.11 below shows the flow field under both boundary conditions. Both the spanwise slice and the iso-surface traces show a similar pattern. The tip vortices of each are distinct and follow the contraction expected from a rotor wake. The main difference is the vertical convection of the vortices, with the spanwise slice showing that the coherent structure of the farfield case has moved further downward than that of the sink case. In both instances, the starting vortex has moved beyond the current frame, as with the previous cases. To study the difference between these results further, a view of the domain nearer to the lower boundary is necessary.
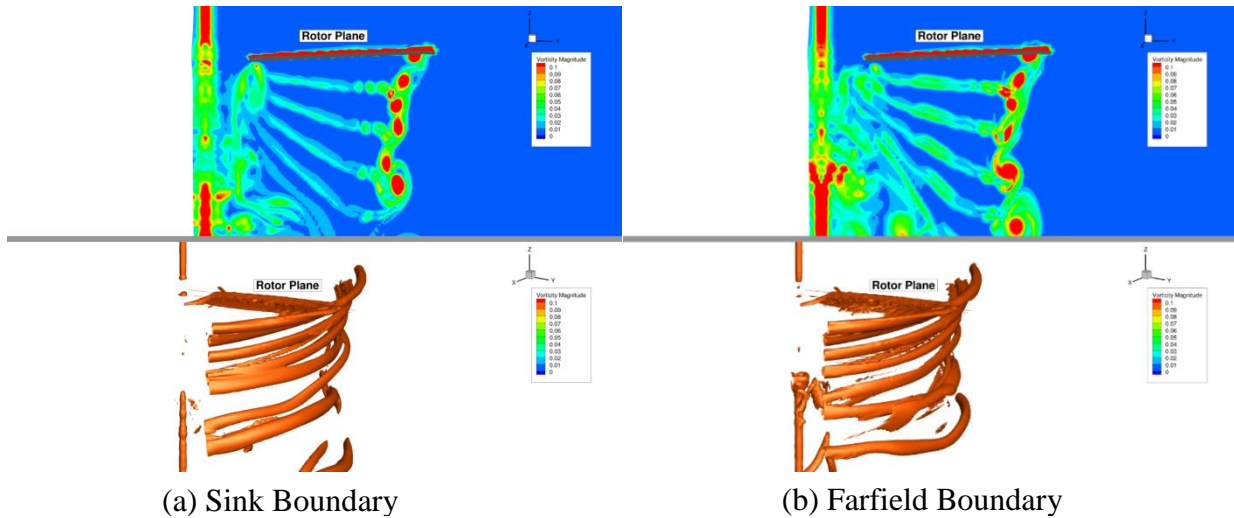
(a) Sink Boundary   (b) Farfield Boundary

Figure 3.11 Flow under sink and farfield conditions at the lower boundary of the computational domain

Figure 3.12 below shows a spanwise slice of the computational domain from the rotor blade to the lower boundary. The rotor blade is located just under the label, with the rotor vortex visible below it. Again it can be seen that there is not a significant difference in the flow near the blade and the resulting wake. Meanwhile near the lower boundary there is a thin region of turbulent flow at the edge of the sink condition. This region results from the downward flow caused by the sink boundary and the surrounding flow which is largely stationary. This is illustrated further in Figure 3.13.
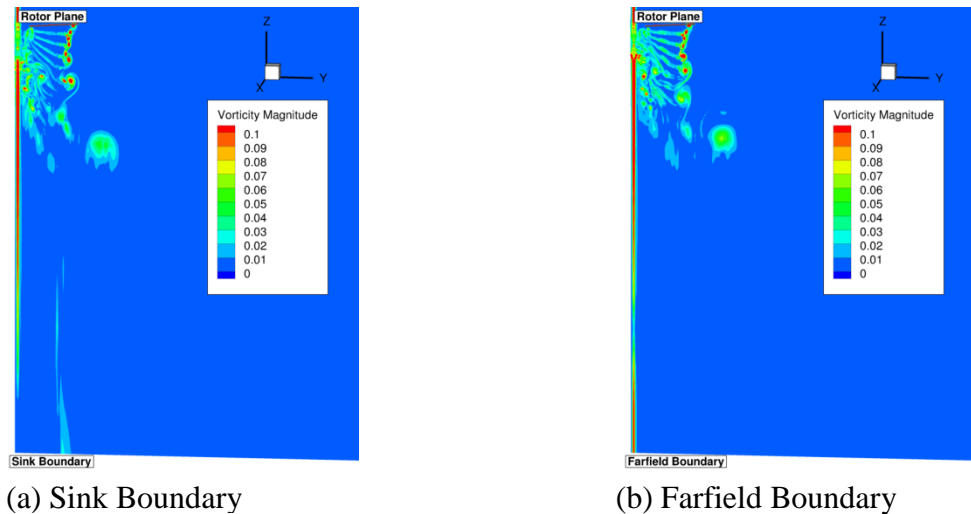

(a) Sink Boundary   (b) Farfield Boundary

Figure 3.12 Zoomed out view of a spanwise slice illustrated with vorticity magnitude contours, showing the lower boundary of the domain

Figure 3.13 highlights the same region as the previous image but using a vertical momentum contour plot rather than the vorticity magnitude employed above. The wake is visible as a series

80

of stronger flows located in the region of the downwash of the rotor. The starting vortex is visible as a slight upward flow located outboard of the lowest point in the rotor wake. Again there is not a large difference between the cases, with the exception appearing at the bottom of the domain. Here the extent of the sink condition, as well as the effect on the region above it, is clearly visible. Also apparent is the border of the downwash region created by the sink and the surrounding stationary fluid, resulting in the thin region of vorticity above.
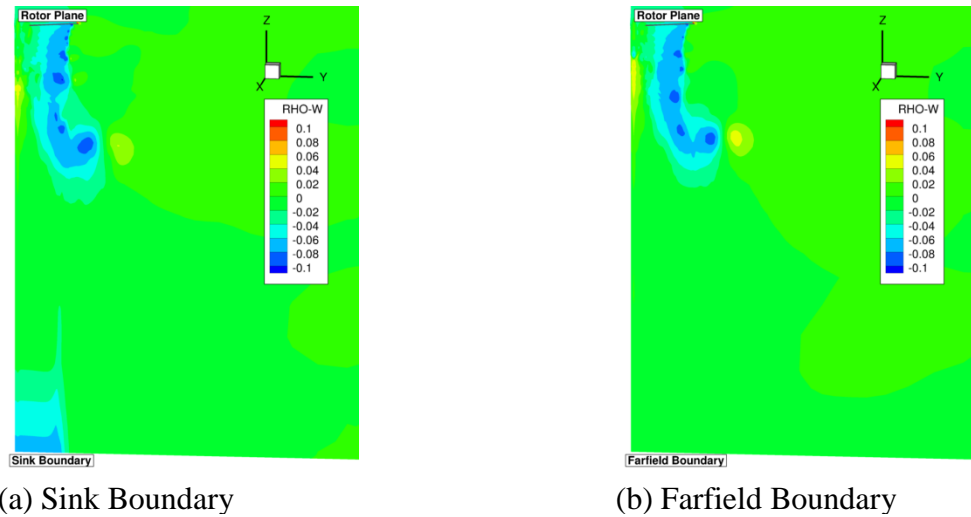

| (a) Sink Boundary | (b) Farfield Boundary |

Figure 3.13 The same view and cross-section as above, but visualized using vertical momentum

The integrated forces as well as the flow fields presented above indicate that the sink boundary condition did not have a significant impact on the flow near the rotor and the wake. The rotor wake does show varying levels of vertical convection between the cases, but these differences are not large enough to meaningfully affect the flow. Furthermore, the main portion of the wake does not come into direct contact with the region affected by the sink, as evidenced by Figures 3.12 and 3.13. This may be altered by reducing the size of the computational domain, bringing the sink closer to the wake, but care would have to be taken to not warp the wake itself and create a non-physical condition in the flow. As the current iteration of the sink boundary has not produced significant alterations in the solution, the current author chose to use the farfield boundary conditions when conducting further tests.

3.2.2. Sweep of Collective Pitches Using Swept-Tapered Tip Blade

The experimental results of Balch et. al[21] are presented here as a comparison to the computational results of the current work. Balch et. al primarily studied the effects of rotor/fuselage and main rotor/tail rotor interference using model scale testing at the Sikorsky Aircraft Model Rotor Hover Facility. The isolated rotor tests were conducted, both in and out of ground effect, to provide a baseline for their measurements. It is the out of ground effect (OGE) tests of the isolated S-76 rotor that the Invited Hover Session and the current work has chosen as

81

a point of reference. The experimental study positioned the rotor at a height of 3.0 Z/R to simulate OGE while the current work ensured the farfield boundaries of the computational domain were far enough from the rotor to ensure no interference.

The full sweep of collective pitches was run following testing using the baseline 10° collective case. As discussed in the previous section, a number of features and alterations to OVERTURNS were tested using this case before running the sweep of collective pitches with the final configuration of the program. All runs employed coned blade meshes using an O-O topology and Cartesian background mesh with clustering near the root and tip as shown in Figure 2.8. The solver employs the 5th order CRWENO spatial scheme and 2nd order BDF2 time method with an LUSGS approximation method. SA Turbulence modeling, with the addition of the DDES method developed by Spalart et. al[37-39], ensured accurate turbulent modeling. Farfield conditions are employed at the outer boundaries of the computational domain as described in Section 2.5.7.1.

The results of the collective sweep are presented in the following pages. Each case was run to 10 rotor revolutions, at which point the integrated forces and flow fields were extracted. The wall time was approximately 5 ½ hours per rotor revolution, resulting in total run times of just over two days. First the integrated thrust and power results, along with Figure of Merit, are compared with the experimental results reproduced from Balch et. al[21]. Next an example of the flow fields, visualized using the vorticity magnitudes, is presented.

Figure 3.14 below shows the integrated coefficient of thrust values generated by OVERTURNS as they compare to the experimental results of Balch et. al[21]. There is a high degree of correlation between the swept-tapered tip results, with the computational and measured values with the largest difference less than 5% at the lowest tested collective pitch while most values differ by only 1-2%. In all subsequent figures the experimental data is denoted by blue x's while the computational data is represented using red o's connected with a solid line.
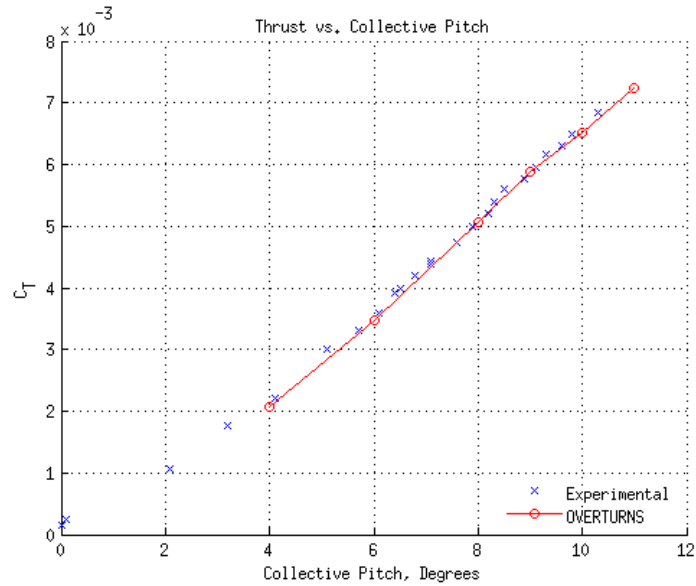
Figure 3.14 Integrated thrust values over a range of collective pitch angles for swept-tapered tip

Figure 3.15 below presents the corresponding integrated coefficient of power values over the same range of collective pitches. Here there is a larger divergence between computational and experimental results, with OVERTURNS over predicting the power requirements for the given sweep of pitches by as much as 18%. The swept-tapered tip case sees consistent over prediction, matching the trend of the experimental data. The cause of the consistent over estimation of the power requirement is still under investigation and meanwhile the shared trends of each set of results is explored further. It is likely that the modeling of the near blade transition and turbulent regions is the cause of the problem, as this would explain the good agreement in thrust but over prediction of power requirements.
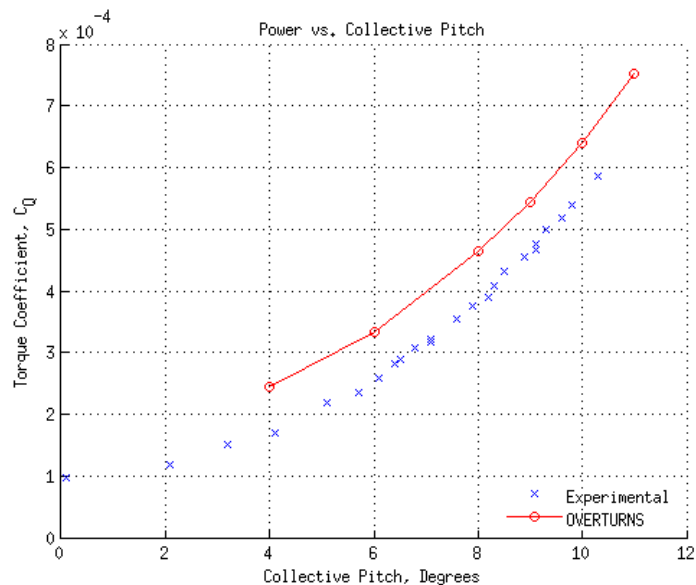

Figure 3.15 Integrated power values over a range of collective pitch angles for swept-tapered tip

While the trend of the computed and experimental power values match the actual values are offset from one another. Therefore, to better illustrate the shared trends between the data sets, a modification was added to the computational power requirements. First, a quadratic regression of both data sets was taken, to acquire a best fit function for each data set. These functions are presented below as,

$$Experimental\ Data: \quad C_Q = 0.0001(0.0481\theta^2 - 0.0266\theta + 0.9974) \tag{3.1}$$
$$Computational\ Data: \quad C_Q = 0.001(0.0058\theta^2 - 0.0154\theta + 0.2133) \tag{3.2}$$

Knowing the best fit functions for both data sets, the computational data was re-plotted using the offset present in the experimental equation, Eq. (3.1). Figure 3.16 below shows this modified comparison. This image makes the shared trend between each data set more apparent, with the modified computational and experimental data showing very similar behavior.
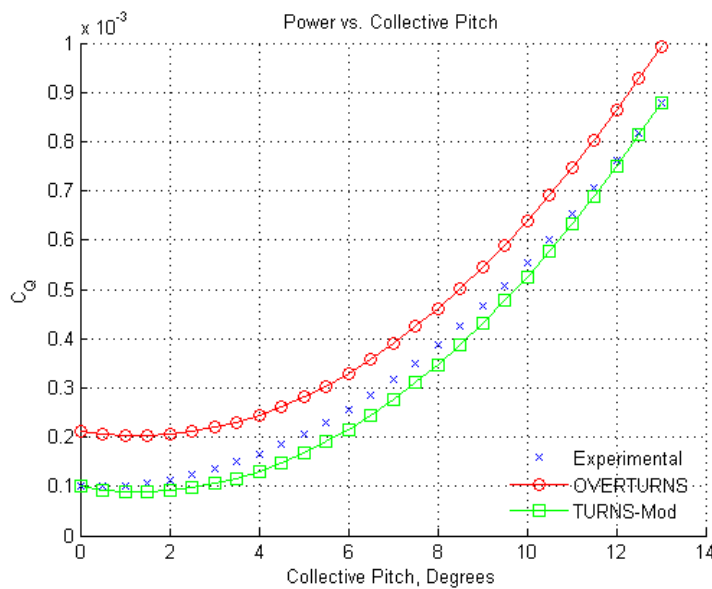

Figure 3.16 Modified integrated power values, with the best fit computational data displayed in green

Figure 3.17 below shows the Figure of Merit, or lifting efficiency of the rotor, for the sweep of collective pitch angles. Due to the over prediction of power requirements shown above, the figure of merit is consistently under predicted by OVERTURNS. Due to the accurate thrust values, and consistency of the predicted power, the correct trends are captured despite the decreased performance calculations. Below this, Figure 3.18 presents the modified FM, based on the modified coefficient of power presented above. Due to the lower power requirements seen above the FM is higher than that of the experimental predictions, trend predicted by the best fit function remains close to that of the experimental data.
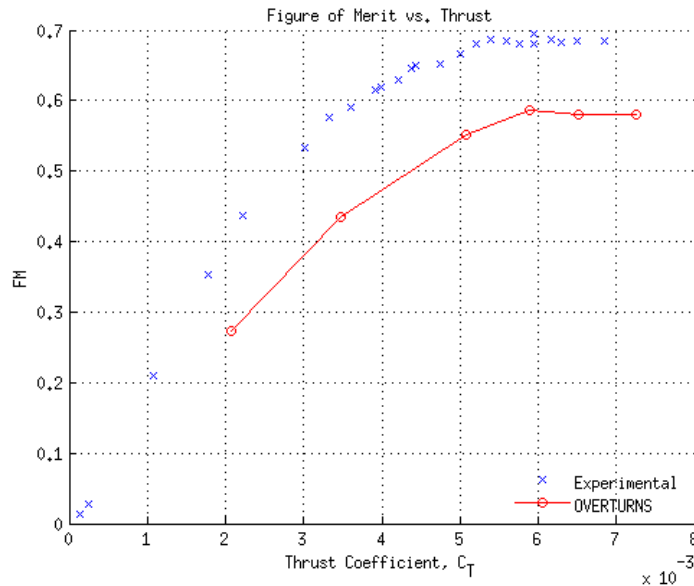
Figure 3.17 Figure of Merit values over a range of integrated thrust results for swept-tapered tip
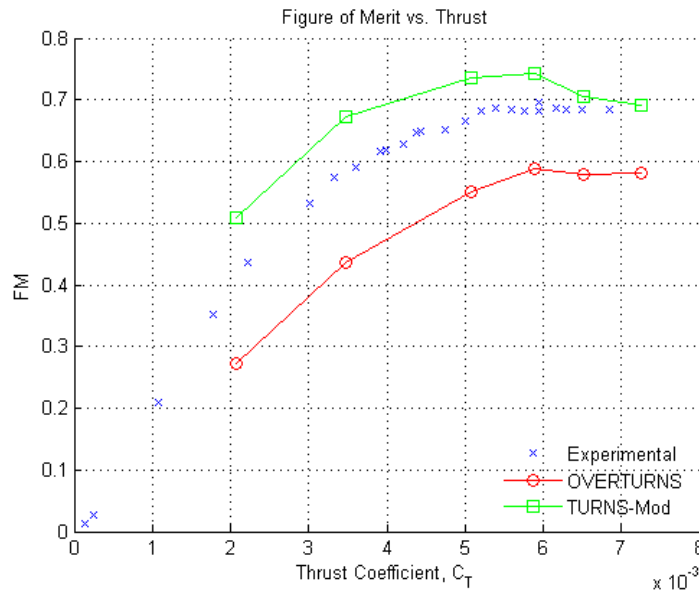

Figure 3.18 Figure of Merit values for collective sweep including the best fit data

Figure 3.19 below compares the integrated thrust and power coefficients predicted by OVERTURNS to the experimental values. The power requirements for a given level of thrust are overestimated, due to the inflated values seen in Figure 3.15. As when observing the thrust and power coefficients independently, the computational data matches the experimental trend despite higher magnitudes for the power requirements. Meanwhile, Figure 3.20 adds the modified data set, and the overlap with the experimental trend is evident.
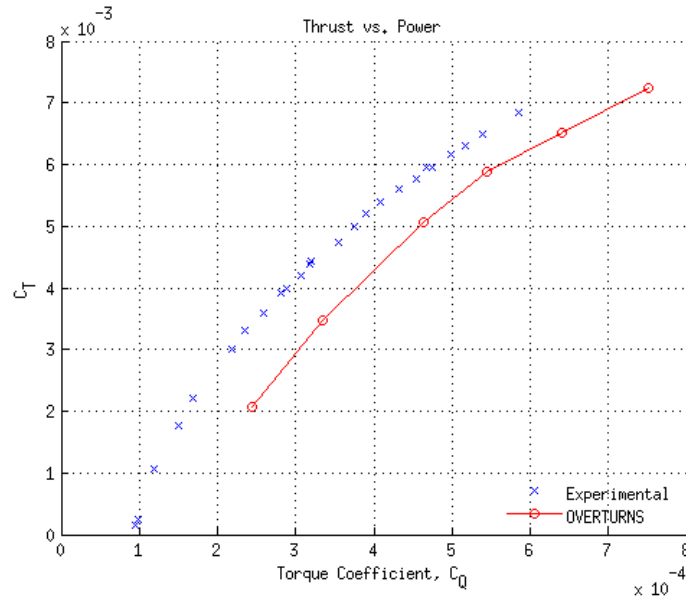
Figure 3.19 Integrated thrust values over a range of integrated power results for swept-tapered tip
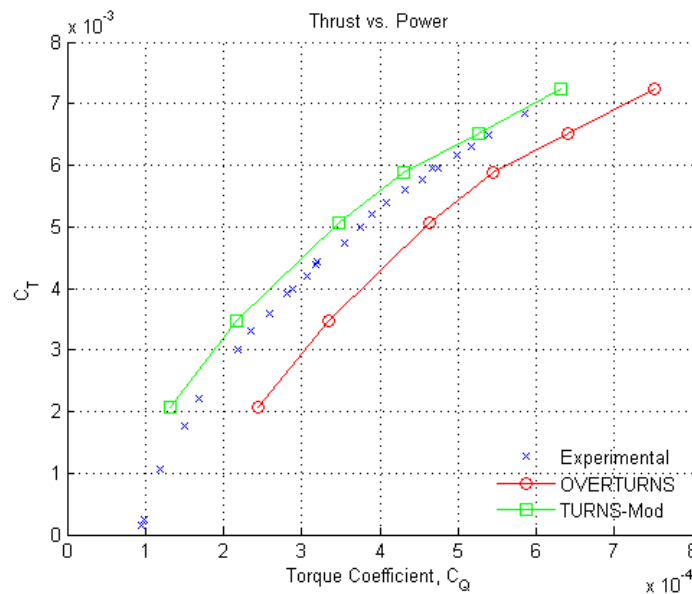

Figure 3.20 Integrated thrust values over a range of integrated power values with the addition of the modified best fit values

The following section displays the flow fields after 10 rotor revolutions visualized using Tecplot. All figures consist of two images, with the top displaying a slice located at the mid-chord of the rotor, extending spanwise past the tip to the edge of the computational domain. This slice shows the vorticity magnitude of the flow, on a scale from 0 to 0.1. The bottom image shows an iso-surface visualization of a vorticity magnitude of 0.1. In the subsequent images, the starting vortex of the rotor has convected well below the plane of the rotor and out of the frame of view.

Figure 3.21 below shows the flow at a collective pitch of 4°. Here the vortices below the plane of the rotor are not very powerful. While maintaining the helical shape and contraction expected of a rotor in hover, the strength of the vortices is low when compared to higher angles of attack.
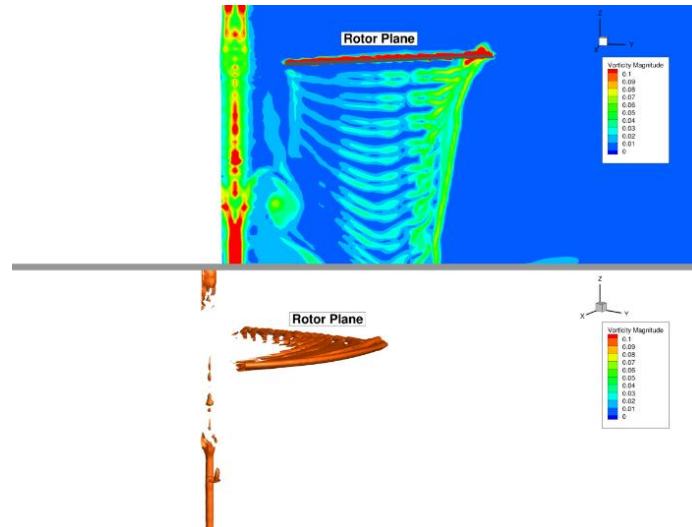

Figure 3.21 Flow field of 4° collective pitch results. Top image: Vorticity magnitude in the spanwise plane of the rotor; Bottom image: Iso-surface at a vorticity magnitude of 0.1

Figure 3.22 visualizes the flow at 6°. As seen in the spanwise slice, the vortices have increased in strength. While not yet strong enough to record multiple passes of the vortex in the iso-surface visualization, traces of the first blade passages have started to appear.
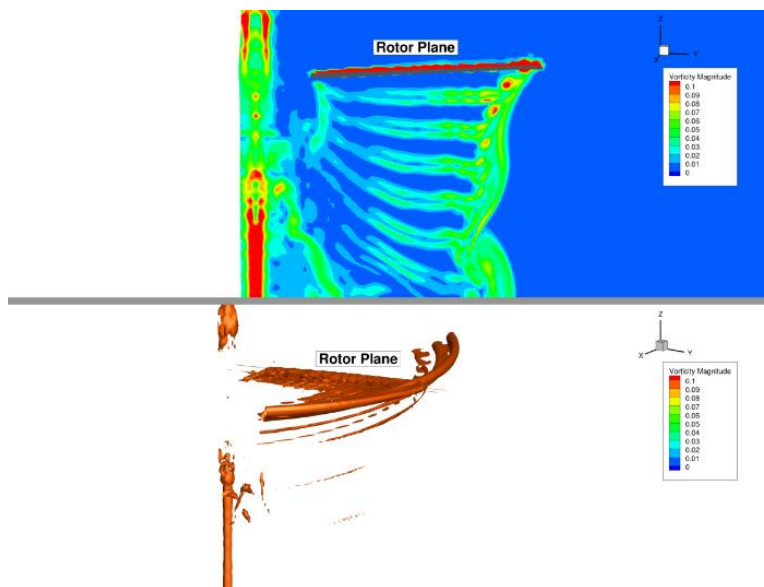

Figure 3.22 Flow field of 6° collective pitch results. Top image: Vorticity magnitude in the spanwise plane of the rotor; Bottom image: Iso-surface at a vorticity magnitude of 0.1

Figure 3.23 presents the results at 8°. Consistent with the previous results, the vortices have strengthened further, now clearly visible as iso-surfaces with a magnitude of 0.1. The first passages of the rotor are well preserved, and the full structure of the vortex is visible. As expected, the vorticity near the root is far weaker than at the tip, though traces are visible below the root.
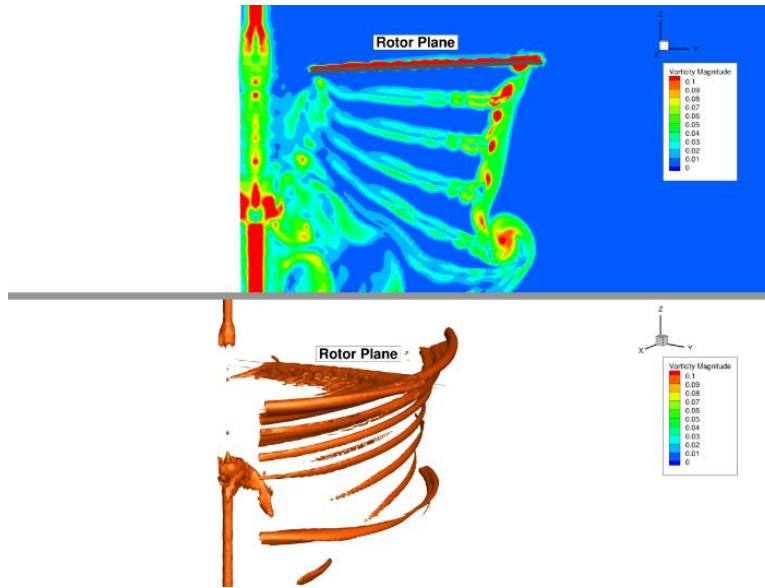

Figure 3.23 Flow field of 8° collective pitch results. Top image: Vorticity magnitude in the spanwise plane of the rotor; Bottom image: Iso-surface at a vorticity magnitude of 0.1

The 9° collective pitch flow field is presented below in Figure 3.24. As the spanwise slice shows, the vortices have grown in size, though the difference between the 8° and 9° wakes is less pronounced than that between 8° and 6° cases. Meanwhile, the iso-surfaces have thickened, demonstrating not only that the strength of the vortices has increased, but that the regions of high vorticity have increased. The behavior at the root remains comparatively weak and does not show the same clear structure that the tip vortices form.
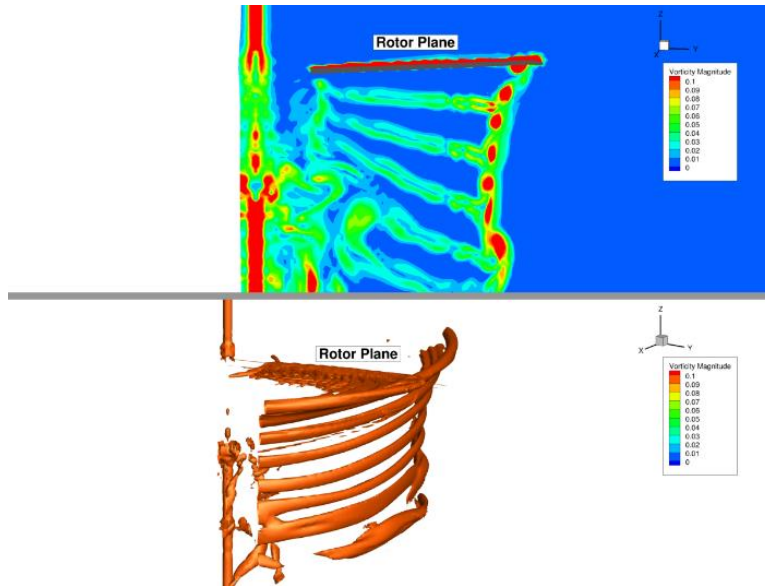
Figure 3.24 Flow field of 9° collective pitch results. Top image: Vorticity magnitude in the spanwise plane of the rotor; Bottom image: Iso-surface at a vorticity magnitude of 0.1

Figure 3.25 shows the continued trend of strong tip vortices at 10° highlighted by the iso-surfaces below. The vortices further downstream have grown slightly while those further upstream have changed slightly from the previous results but again there is not a significant difference induced by the 1° difference with the previous case. The root behavior remains chaotic and weaker than the flow near the tip.
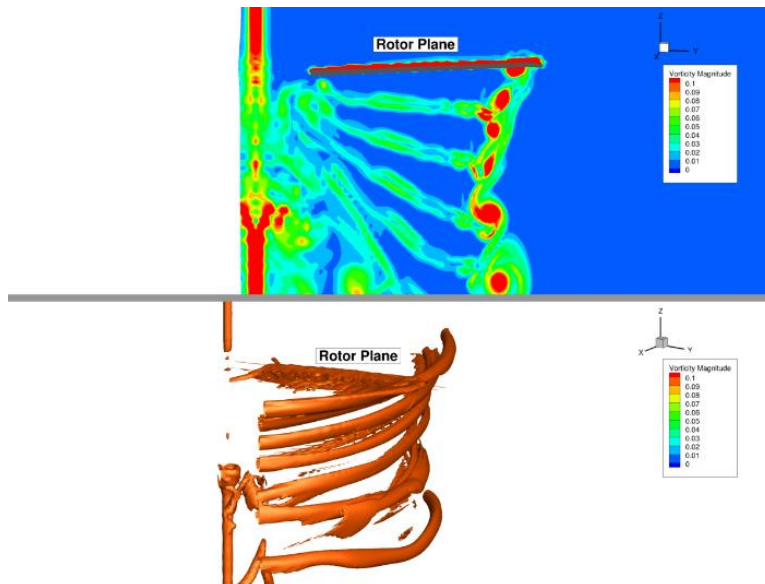

Figure 3.25 Flow field of 10° collective pitch results. Top image: Vorticity magnitude in the spanwise plane of the rotor; Bottom image: Iso-surface at a vorticity magnitude of 0.1

89

The last case, an 11° collective pitch, is presented below in Figure 3.26. In this flow region the airfoil is approaching stall and beginning to lose efficiency, as seen in Figure 3.16. Meanwhile the wake retains its helical shape but has started to lose coherency, with the vortex contraction showing a more erratic path. The vorticity magnitude remains high, indicating that regions of turbulent flow remain into this higher angle of attack.
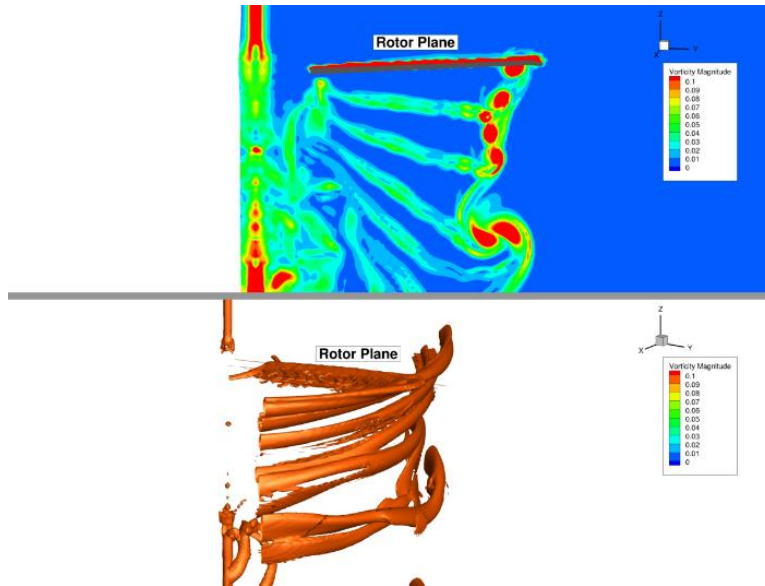

Figure 3.26 Flow field of 11° collective pitch results. Top images: Vorticity magnitude in the spanwise plane of the rotor; Bottom images: Iso-surface at a vorticity magnitude of 0.1

Having reviewed the flow fields at the full range of collective sweeps, a few conclusions may be reached. Firstly, the strength of the tip vortices steadily increased as the angle of attack increased, indicated by the spanwise slices and the increase in high vorticity regions traced by the iso-surfaces. While the Figure of Merit does level out at the higher collective pitches, the rotor has not yet entered the fully stalled region. This is evidenced by the collective thrust and power values seen in Figures 3.14 and 3.15. Furthermore it is clear that OVERTURNS has preserved the rotor wake across all collective pitches, allowing for observation of the changes in the rotor wake as a result of the altered pitch. Having reviewed the sweep of collective pitches, the 10° case was chosen for greater analysis in the following section.

3.2.3. Detailed Results From Selected Collective Pitch

The previous sections examined the integrated forces on the blades as well as the flow fields for a range of collective pitches. The current section expands on the results presented beforehand, focusing on a single collective pitch. As when testing various parameters of the simulation to attain the optimal configuration, the 10° collective pitch case was chosen for further analysis. By

restricting this section to a single case, more detailed results such as chordwise pressure distribution, sectional loading, and near-body vortices may be examined.

Figure 3.27 below shows the chordwise pressure distribution at a range of spanwise locations. Ranging from $r/R = 0.5$ to 0.99, the growing pressure loading as one moves outward along the blade is apparent, along with the dip in loading due to tip loss seen in 3.27.c. The largest distribution is seen in 3.27.f, at $r/R = 0.975$ before falling slightly at the very tip of the blade, corresponding with the expected behavior of a flow over a rotor blade.
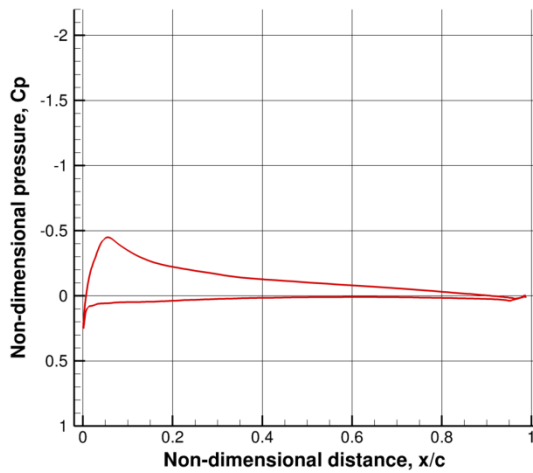
(a) r/R = 0.5
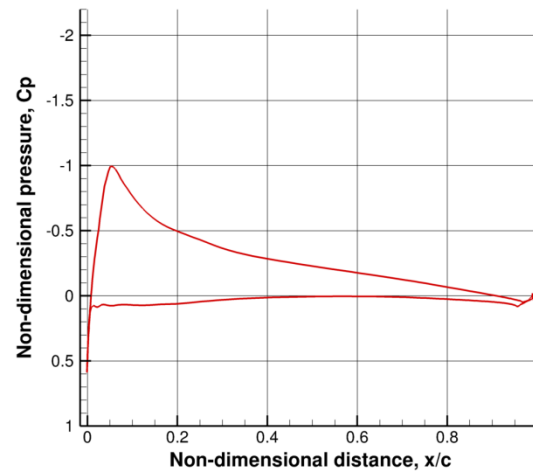
(b) r/R = 0.75

(c) r/R = 0.85

(d) r/R = 0.90

91
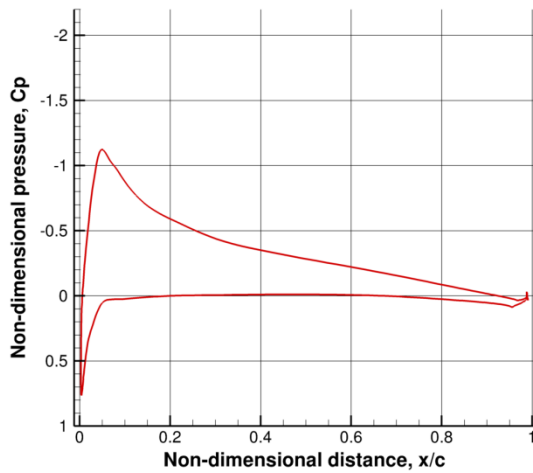
(e) r/R = 0.95
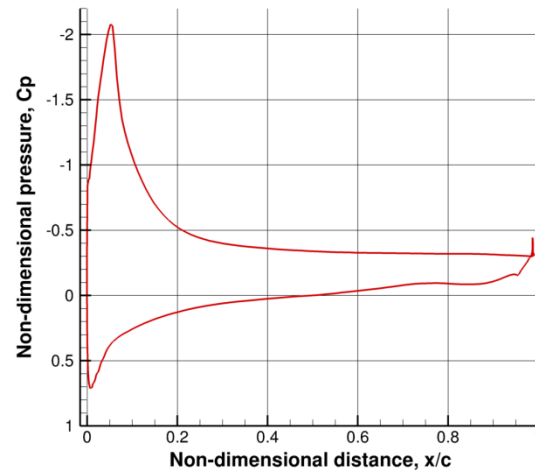
(f) r/R = 0.975
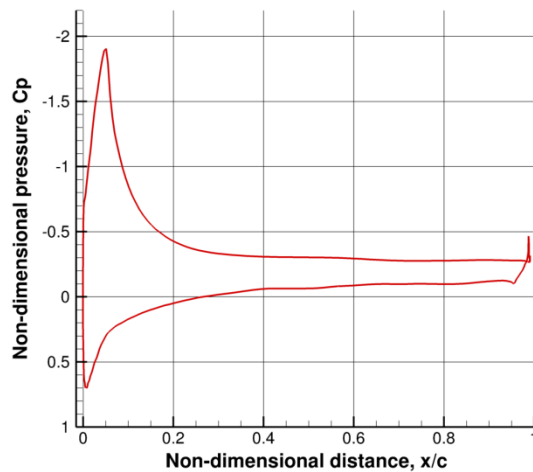
(g) r/R = 0.99

Figure 3.27 Non-dimensional Coefficient of Pressure distribution at selected spanwise locations

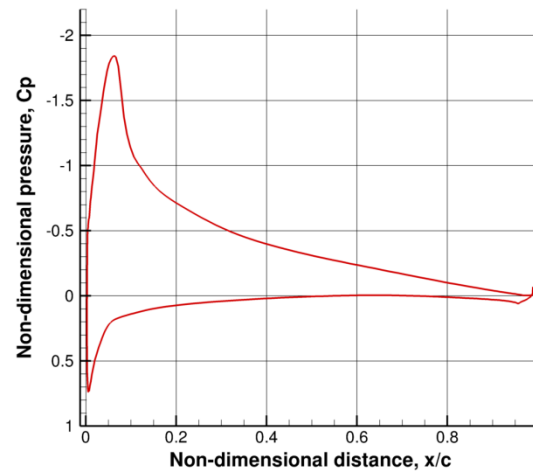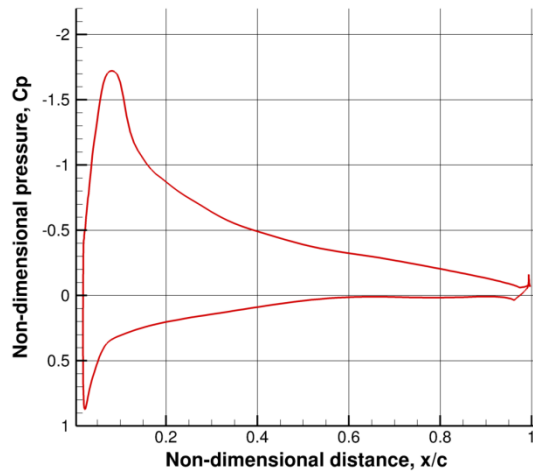Figure 3.28 below shows the corresponding sectional thrust distribution over the blade. Again the distribution behaves as expected, strengthening as the outer portion of the blade is approached before dipping, peaking, and falling off at the edge of the blade. This corresponds with the pressure distributions seen above, and provides further insight into the predicted behavior near the surface of the rotor blade.

Figure 3.29 presents the corresponding sectional torque coefficient for the 10° case. As expected the torque steadily increases from the root through the mid-span of the blade, before changing drastically along the outer 10% of the blade. As with the sectional thrust, there is a small trough before a large peak, though in this case the peaks and troughs are sharper than those of the thrust values. The discontinuity just past *r/R* = 1 results from the rounded tip used to close the blade

92

mesh, in which the adjacent mesh points are very close to one another, and   does   not   adversely affect the rest of the span.

Figure 3.28 Sectional thrust coefficient as a function of spanwise location
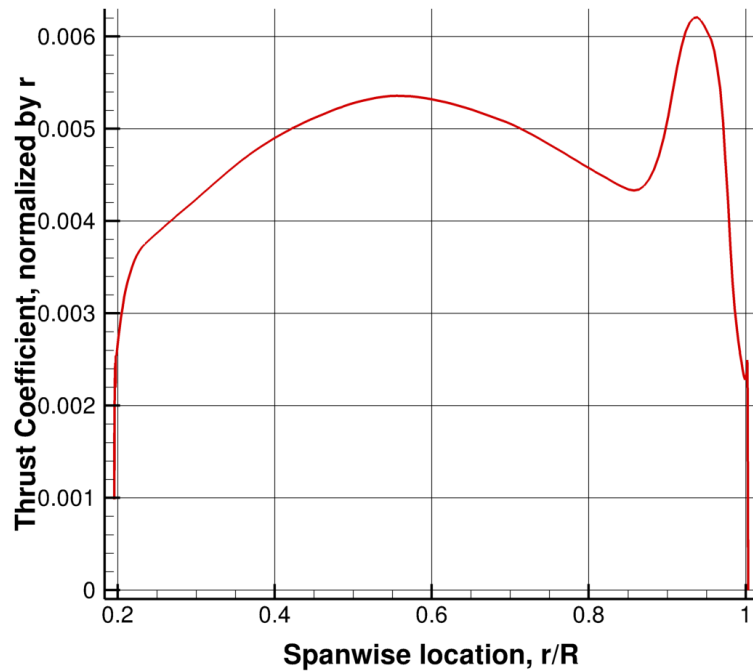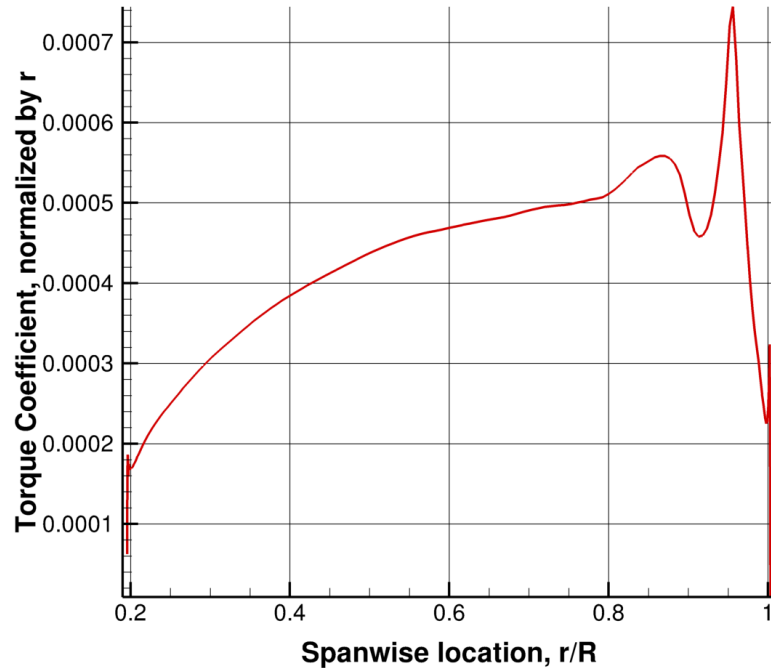
Figure 3.29 Sectional torque coefficient as a function of spanwise location

The previous discussion focused on the integrated forces over the blade, specifically the coefficient of pressure distribution over the chord and the sectional thrust and torque coefficients along the span of the blade. In addition to reviewing the forces on the blade, it is useful to visualize the near-body flow. These results correspond directly with the data seen above and drives the rest of the flow field, primarily the wake created by the rotor blades. The following paragraphs present the near-body flow as well as a discussion of the images presented in Figure 3.30.

Figure 3.30 below shows the vorticity contours near the body of the rotor blade. The images show a series of 8 slices, beginning at the leading edge of the blade and spaced 0.2 c apart. The magnitude of the vorticity values displayed here is greater than in previous images to ensure a clear image, with values smaller than 1 being blanked out. Figure 3.30.a shows the view from behind the blade, slightly outboard of the tip to allow clear images of the growing vorticity contours. Meanwhile, 3.30.b provides a spanwise view of the blade, demonstrating the strengthening vortex sheet along and behind the mid-span of the blade.
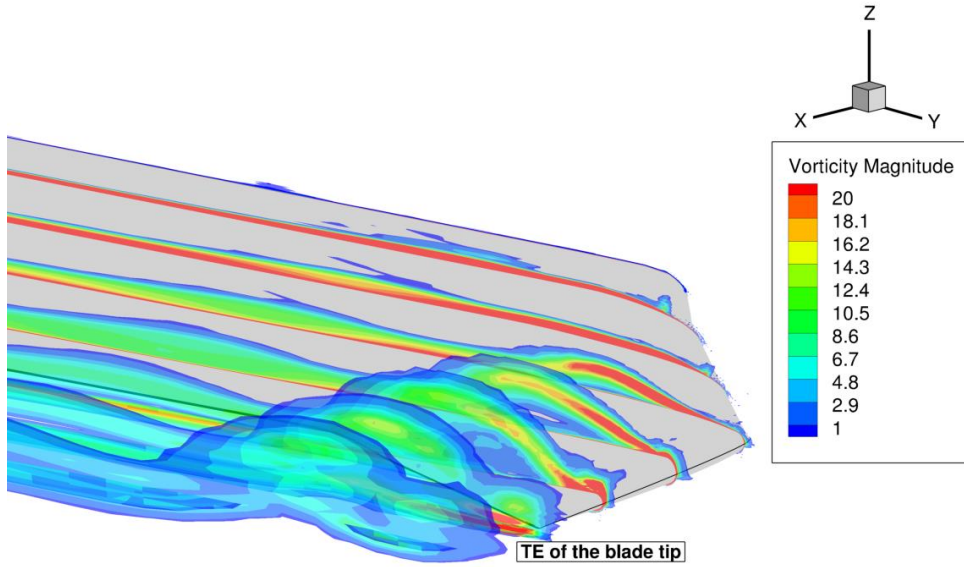
(a) View from behind the TE
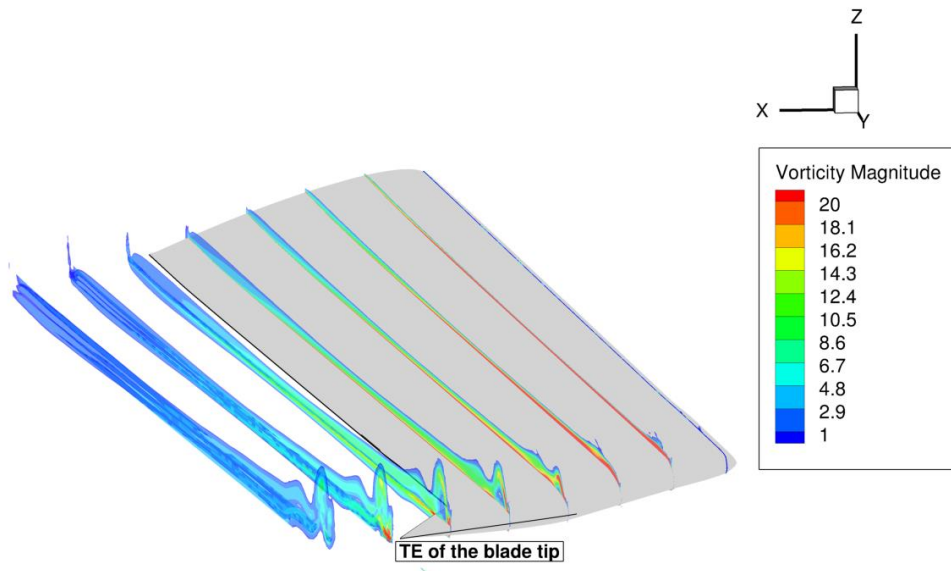
(b) View from outboard of the blade tip

Figure 3.30 Contour plots of vorticity magnitude near the blade

95

Both images demonstrate the large region of higher vorticity flow present at the transition from the rectangular cross-section of the inner 95% of the blade to the swept-tapered blade tip. Near the LE of the blade there is no apparent jump but closer to the TE and behind the blade it is clear that the transition has produced a significant vortex structure. In addition to the chaotic near-body flow seen above, the tip vortices which define the rotor wake represent another region which has a large effect on the flow field.

The tip vortices consist of regions of turbulent flow generated at the tips of the rotor blades, propagating out behind the blade and downward to form the helical rotor wake seen in hover. The current work tracked the wake trajectory for the 10° collective pitch case and the contraction and descent rates are presented below. However, first a brief discussion of the tracking method used is presented.

The tracking method employed in the current work calculates the vorticity of each point in the domain as seen below in Figure 3.31. As expected the shape of the wake and regions of higher vorticity follow the same pattern as the vorticity magnitude and iso-surfaces presented in Figure 3.25. Choosing the TE of the blade tip as the starting point, the method evaluates the vorticity and azimuth angle of the neighboring cells before stepping to the cell that fulfills two conditions:

1. First the neighboring cells are sorted by vorticity and checked against the vorticity of the current cell.

2. The neighbor with the highest vorticity and a higher azimuth angle, to ensure stepping in the correct direction, becomes the new center point and the process is repeated.

Using this method, the path of the vortex is tracked and the radial and vertical positions of each point are stored. Adjusting the azimuth angle such that the starting point has a wake age of 0 degrees, the contraction and descent rates measured are presented in Figure 3.32 below.
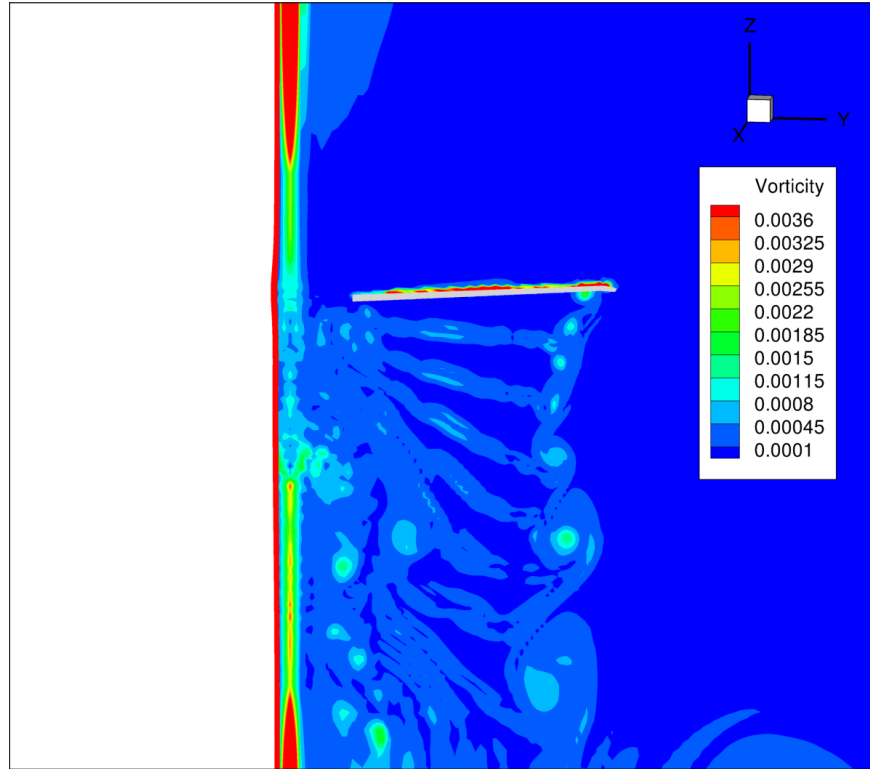
Figure 3.31 Vorticity as seen by the vortex tracking method
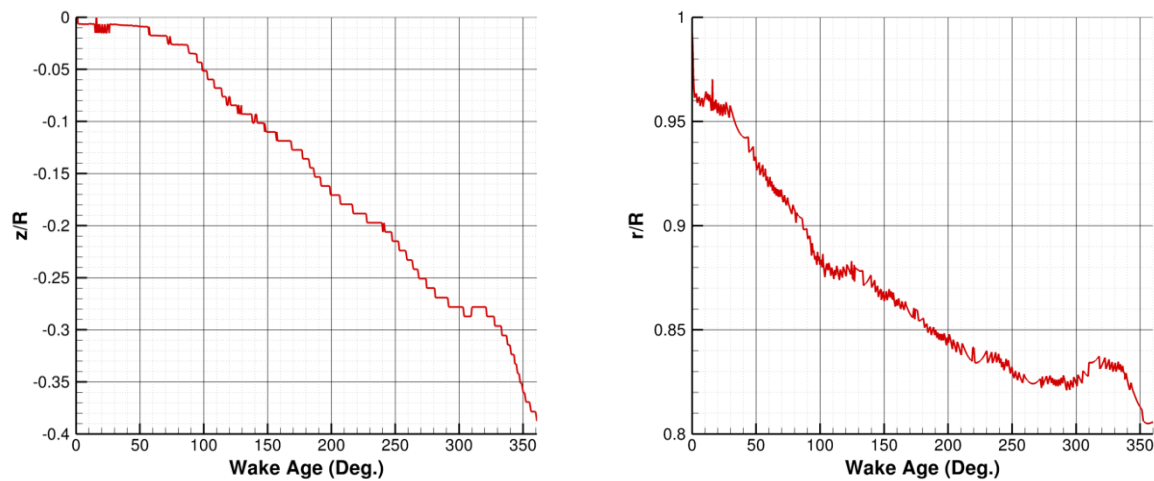
(a) Tip vortex descent rate　　　　　　　(b) Tip vortex contraction rate

Figure 3.32 Measured tip vortex contraction and descent rates at 10° collective pitch

Figure 3.32 shows that in the first blade passage the wake has shrunk by approximately 20% of the rotor radius while descending by 40%. While experimental values are not available, the computational results of Kim et. al[22], presented in Figure 1.13, and Sheng et. al[23], presented in Figure 1.20, provide opportunities for comparison. Note that the results of Kim and Sheng are

taken at the trim condition of $CT/\sigma = 0.09$ (collective pitch = 9.3º) while the current work's results were taken at a collective pitch of 10º, so a slight divergence is expected. The current work shows a slower descent than that of Sheng et. al by approximately 1/3 and a comparable descent rate to that of Kim et. al. Comparing the contraction rates, OVERTURNS shows less contraction than Sheng et. al, a difference of 5% of the rotor radius, but both predict more contraction than that of Kim et. al, which predicts a shrinkage of 10% of the rotor radius. Figure 1.13 shows that Kim et. al included results from a number of other solvers, including a previous version of OVERTURNS. The current work shows good agreement with these predictions in both descent and contraction rate.

3.3. Summary

In this chapter the sweep of collective pitches of the S-76 rotor in hover was evaluated to ensure the accuracy and functionality of the OVERTURNS solver. Initially the 10º collective case was used to test various operating conditions and arrive at the optimal configuration of the solver. Using the quarter-domain adapted Cartesian mesh and the 5th order CRWENO spatial scheme, the full range of collective pitches was tested. The thrust values showed excellent agreement with the experimental results while the power requirements were over-predicted while maintaining the expected trend. The rotor wake was captured for multiple blade revolutions at all pitch values, capturing the increased strength of the tip vortices at higher pitch angles. Following the collective sweep, a few more detailed results from the 10º case were reviewed, including pressure distribution, spanwise loading, and vortex behavior.

Chapter 4. Conclusion

4.1. Summary

The field of rotor craft is progressing at a rapid rate, requiring new developments in rotor design and the technology employed to test these designs. Simplified aerodynamic models provide quick evaluations of rotor performance but are not valid across all cases and cannot capture all behaviors of a turbulent flow. Three-dimensional computational fluid dynamics simulations provide an additional method of testing, allowing designers to study aspects such as the near-body flow, helicopter wake, and rotor efficiency under fully modeled operating conditions.

The present work evaluates the S-76 rotor in hover using the three-dimensional solver, OVERTURNS. Employing a structured mesh built around a single blade, periodic boundary conditions were implemented to simulate the remaining blades and conserve computational resources. The baseline S-76 blade was used to test a number of options within the solver to discover the optimal configuration. Following this, a sweep of collective pitch angles was performed, as well as a more detailed analysis of the 10° case. These results were compared with prior computational results from the AIAA Invited Hover Session as well as the experimental results available.

4.2. Observations and Conclusions

The current section summarizes the primary observations and conclusions from the current work.

4.2.1. OVERTURNS Conditions Testing

1. Both the tested cylindrical and Cartesian background meshes predicted integrated performance values within 5.5% of one another. However, the cylindrical mesh did not preserve the tip vortices past the first rotor revolution due to deteriorating mesh resolution. Therefore the Cartesian mesh was employed for further testing.

2. The full 4-bladed simulation would have proved too computationally expensive for available resources. By introducing periodic boundaries, the computational domain could be reduced to a single blade, while the effects of the remaining blades are simulated due to the periodic boundaries.

3. By altering the mesh adaption routine to cluster prescribed regions of the computational domain, resolution could be improved without adding nodes to the mesh. The additional

resolution in areas of high vorticity aided the preservation of the wake structure without sacrificing accuracy in other areas. Both meshes gave similar performance predictions, predicting thrust and power coefficients within 5% of one another.

4. The 5th order CRWENO reconstruction scheme proved superior to the 3rd order MUSCL and 5th order WENO schemes which were previously implemented in OVERTURNS. Again each method predicted thrust and power within 2% of one another but differed in the wake preservation. The MUSCL scheme showed the worst preservation, while the two 5th order schemes were closer in performance but ultimately the CRWENO scheme produced superior results.

5. Ultimately both the farfield and sink boundaries predicted the integrated forces within 1% of one another. There was a slight difference in the descent speed between the two cases but the sink boundary itself did not propagate far enough into the computational domain to meaningfully affect the solution.

4.2.2. Sweep of Collective Pitches

1. Thrust coefficients saw excellent agreement with experimental values across all tested pitch angles, with a maximum difference of under 5% at the lowest collective pitch and close to 1-2% at most values.

2. The computational power coefficients were over predicted by as much as 20%, though shared the same trend as the experimental values. This is likely a result of the differences in the measured and modeled transition and turbulent flow in the near-body region. By performing a quadratic regression on both data sets, the shared trend was verified and discussed.

3. As a result of the power over prediction, the Figure of Merit was under predicted across tested collective pitches, also by as much as 20% at most and closer to 15% at most collective pitches. As with the power coefficient results, the shared trend was examined and considered.

4. The flow field results show a gradual strengthening of the tip vortices as collective pitch increased. Stronger tip vortices began to appear at 8° and the rotor wake was captured for multiple blade passages.

4.2.3. Analysis of 10º Case

1. The chordwise pressure distributions showed the growth in pressure loading from the root to tip of the blade, including the trough and peak seen near the tip of the blade.

2. The sectional thrust and torque distributions were consistent with expected results. A slight discontinuity occurred at the very tip of the blade but this can be attributed to the rounded tip used to close the blade mesh and the very small spacing of the mesh in this region.

3. The near-body vorticity contours showed a large region of vorticity at the transition point between the swept-tapered tip and the remainder of the blade. The chordwise slices showed the gradual strengthening of the vorticity moving from the LE to the TE of the blade.

4. The tracking method successfully captured the tip vortex trajectory and change in the rotor wake through the first blade passage.

5. Both descent and contraction rates showed good agreement with previous computational studies.

4.0. Contributions of the Current Work

This section summarizes the contributions from the current work:

1. Creation of an O-O blade mesh and refinement of the mesh generator, culminating in the usable S-76 rotor blade employed in the current work.

2. Creation of the quarter domain Cartesian background mesh, as well as the periodic boundary conditions necessary for a Cartesian topology.

3. Implementation of the mesh adaption program, resulting in a clustered mesh to better capture the tip vortices without sacrificing solution accuracy.

4. Implementation of the CRWENO numerical reconstruction scheme, a 5th order spatial scheme resulting in better preservation of the tip vortices.

5. Full testing of the S-76 baseline blade in hover, to validate the additions to OVERTURNS through comparison with other computational results as well as the experimental data.

4.1. Future Work

The final section presents recommendations and possibilities for further research:

1. The solver currently consistently over predicted the coefficient of power values for the sweep of collective pitches. Additional investigation into this issue would lend further validation to predictions made with the current solver.

2. In addition to the baseline S-76 blade, alternate blade tips exist and provide further opportunities for testing. The creation of usable blade meshes and validation of these blades presents test case of particular interest.

3. Currently the mesh adaption program is run separately from the OVERTURNS solver. Integration of this into the run environment of OVERTURNS would allow for adaption in tandem with the running of the solution. This would reduce the user input required, negating the need to manually output a solution for use with the mesh adaption routine, before restarting OVERTURNS with the newly altered mesh.

4. The vortex based mesh adaption method represents a significant set up and computational cost, especially when studying a large number of cases. Further work, limited to a small number of tests, may provide further insight into streamlining and optimizing the method.

5. The current work assumes a rigid blade, with no blade deformation capability built into the current routine. Coupling of aerodynamic and structural predictors would result in more realistic predictions, furthering the capabilities of the current software.

6. The current work focused on a tip Mach number of 0.65 and did not produce any trans-sonic or super-sonic regions of flow. Testing a trans-sonic case using the additions to OVERTURNS presented here would ensure that the spatial schemes perform as expected and ensure the additional refinement could handle discontinuities present in trans-sonic flows.

References

[1] Hariharan N., Egolf A., Narducci R., and Sankar L. "Helicopter Rotor Aerodynamic Modeling in Hover: AIAA Standardized Hover Evaluations." AIAA Scitech 2015, Kissimmee, FL January 2015.

[2] Leishman, J. G., "Principles of Helicopter Aerodynamics," New York: Cambridge University Press, 2006.

[3] Zioutis, Christos K., Apostolos I. Spyropoulos, Anastasios P. Fragias, Dionissios P. Margaris, and Dimitrios G. Papanikas. "Influence of Helicopter Rotor Wake Modeling on Blade Airload Predictions." *International Journal of Engineering* 3.6 (2010), pp. 521-537.

[4] Caradonna, F. "Performance Measurement and Wake Characteristics of a Model Rotor in Axial Flight." *Journal of the American Helicopter Society* (1998), pp. 101-108.

[5] Shinoda, P., Johnson, W., "Performance Results from a Test of an S-76 Rotor in the NASA Ames 80- by 120- Foot Wind Tunnel," AIAA 93-3414, AIAA Applied Aerodynamics Conference, August 1993, Monterey, CA.

[6] Leishman, G., Bagai, A., "Challenges in Understanding the Vortex Dynamics of Helicopter Rotor Wakes," AIAA 27th Fluid Dynamics Conference. New Orleans, LA, June 1996.

[7] Werle, H. and Armand, C., "Measures et Visualisation Instationnaires sur les Rotor," ONERA T.P. No. 777, 1969.

[8] Scully, M. P. and Sullivan, J. P., "Helicopter Rotor Wake Geometry and Airloads and Development of Laser Doppler Velocimeter for Use in Helicopter Rotor Wakes," Massachusetts Institute of Technology Aerophysics Laboratory Technical Report 179, MIT DSR No. 73032, August 1972.

[9] Muller, R. H. G., "Winglets on Rotor Blades in Forward Flight - A Theoretical and Experimental Investigation," *Vertica,* Vol. 14, No. 1, 1990, pp. 31-46.

[10] Ghee, T. A., and Elliott, J. W., "The Wake of a Small-Scale Rotor Model in Forward Flight Using Flow Visualization," *Journal of the American Helicopter Society,* Vol. 40, No. 3, July 1995, pp. 52-65.

[11] Felker, F. F., Maisel, M. D., and Betzina, M. D., "Full-Scale Tilt-Rotor Hover Performance," *Journal of the American Helicopter Society,* Vol. 31, No. 2, 1986, pp. 10-18.

103

[12] Dadone, L., "Unpublished photographs from model tests performed in the 20-by-20 foot tunnel at Boeing Helicopters," Philadelphia, 1996.

[13] Tangier, J. L., "Schlieren and Noise Studies of Rotors in Forward Flight," Proceedings of the American Helicopter Society 33rd Annual Forum, Washington, DC., May 1977.

[14] Walters, R. E., and Skujins, O., "A Schlieren Technique Applied to Rotor Wake Studies," American Helicopter Society Mid-East Regional Symposium, Essington, PA, October 26-28, 1996.

[15] Parthasarthy, S. P., Cho, Y. L, and Black, L. H., "Wide-Field Shadowgraph Flow Visualization of Tip Vortices Generated by a Helicopter Rotor," Paper 85-1557, AIAA 18th Fluid Dynamics and Plasmadynamics and Lasers Conference, Cincinnati, OH, 1985.

[16] Light, J. S., "Tip Vortex Geometry of a Hovering Helicopter Rotor in Ground Effect," Proceedings of the 45th American Helicopter Society, Boston, MA, 1989.

[17] Leishman, J. G. and Bagai, A., "Rotor Wake Visualization in Low Speed Forward Flight," Paper 91-3232-CP, AIAA 9th Applied Aerodynamics Conference, Baltimore MD, June 1991.

[18] Bagai, A. and Leishman, J. G., "Improved Wide-Field Shadowgraph Set-up for Rotor Wake Visualization," *Journal of the American Helicopter Society,* Vol. 37, No. 3, 1992, pp. 86-92.

[19] Bagai, A., and Leishman, J. G., "A Study of Rotor Wake Development and Wake/Body Interactions in Hover," *Journal of the American Helicopter Society,* Vol. 37, No. 4, Oct. 1992, pp. 48-57.

[20] Swanson, A. A., "Application of the Shadowgraph Flow Visualization Technique to a Full-Scale Helicopter Rotor in Hover and Forward Flight," Paper 93-3411-CP, AIAA 11th Applied Aerodynamics Conference, Monterey, CA, June 1993.

[21]Balch, D.T., Sccullo, A., and Sheehy, T.W., "Experimental Study of Main/Tail Rotor/Airframe Interactions in Hover Volume I," *NASA Contractor Report* No. 166485.

[22] Kim, J. W., Sankar, L. N., Marpu, R., Egolf, T. A., and Hariharan, N., "Assessment of Planform Effects on Rotor Hover Performance," AIAA Scitech 2015, Kissimmee FL, January 2015.

[23] Sheng, C., Wang, J., and Zhao, Q., "S-76 Rotor Hover Predictions Using Advanced Turbulence Models," AIAA Scitech 2015, Kissimmee FL, January, 2015.

[24] Sutherland, W., "The viscosity of gases and molecular force," *Philosophical Magazine*, Series 5, Vol. 36, No. 223, pp. 507-531, 1893.

[25] Blazek, J., "Computational Fluid Dynamics: Principles and Applications," Second Edition, Elsevier Science, 2006.

[26] Kalra, T., "CFD Modeling and Analysis of Rotor Wake in Hover Interacting with a Ground Plane," Ph.D. Dissertation, Department of Aerospace Engineering, University of Maryland at College Park, 2014.

[27] Van Leer, B., "Towards the Ultimate Conservative Difference Scheme V. A Second-Order Sequel to Godunov's Method," *Journal of Computational Physics*, Vol. 135, No. 2, pp. 229-248, 1997.

[28] Henrick, A. K., Aslam, T. D., and Powers, J. M., "Mapped Weighted Essentially Non-Oscillatory Schemes: Achieving Optimal Order Near Critical Points," *Journal of Computational Physics*, Vol. 207, No. 2, pp. 542-567, 2005.

[29] Ghosh, D., Medida, S., and Baeder, J. D., "Compact-Reconstruction Weighted Essentially Non-Oscillatory Schemes for the Unsteady Navier-Stokes Equations," 42nd AIAA Fluid Dynamics Conference and Exhibit, New Orleans, LA, June 2012.

[30] Roe, P., "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, Vol. 135, No. 2, pp. 250-258, 1997.

[31] Koren, B., "Multigrid and Defect Correction for the Steady Navier-Stokes Equations," Proceedings of the 11th International Conference on Numerical Methods in Fluid Dynamics, Williamsburg, VA, June 1988.

[32] Launder, B. E., Reece, G. J., and Rode, W., "Progress in the Development of a Reynolds Stress Turbulent Closure," *Journal of Fluid Mechanics*, Vol. 68, No. 3, pp. 537-566, 1975.

[33] Durbin, P., "A Reynolds Stress Model for Near Wall Turbulence," *Journal of Fluid Mechanics*, Vol. 249, pp. 465-498, 1993.

[34] Spalart, P. R., and Allmaras, S. R., "A one-equation turbulence model for aerodynamic flows," *Recherche Aerospatiale*, No. 1, pp. 5-21, 1994.

[35] Medida, S., and Baeder, J. D., "Adverse pressure gradient modification to turbulence models for wall-bounded flows," *21st* AIAA Computational Fluid Dynamics Conference, AIAA Paper 2013-2426, 2013.

[36] Rinehart, T., "Contributions to the Understanding of Wind Turbine Aerodynamics Using a RANS Solver with Transition Modeling," MS Thesis, University of Maryland, 2014.

37 Spalart, P. R., and Shur, M., "On the Sensitization of Turbulence Models to Rotation and Curvature," *Aerospace Science and Technology*, Vol. 1, No. 5, pp. 297-302, 1997.

38 Spalart, P. R., Deck, S., Shur, M. L., Squires, K. D., Strelets, M. Kh., and Travin, A., "A new version of detached eddy simulation, resistant to ambiguous grid densities," *Theoretical Computational Fluid Dynamics*, Vol. 20, pp. 181-195, 2006.

39 Scotti, A., Meneveau, C., and Lilly, D. K., "Generalized Smagorinsky Model for Anisotropic Grids," *Physics of Fluids A: Fluid Dynamics*, Vol. 5, No. 9, pp. 259-276, 1993.

40 Jameson, A., and Yoon, S., "Lower-upper implicit schemes with multiple grids for Euler equations," *AIAA Journal*, Vol.25, No.7, pp. 929-935, 1987.

41 Pulliam, T. H., and Chaussee, D. S., "A Diagonal Form of an Implicit Approximate-Factorization Algorithm," *Journal of Computational Physics*, Vol.39, No. 2, pp. 347-363, 1981.

42 Lee, Y., "On Overset Grids Connectivity and Automated Vortex Tracking in Rotorcraft CFD," Ph.D. Dissertation, Department of Aerospace Engineering, University of Maryland at College Park, 2012.

43 Lakshminarayan, V. K., "Computational Investigation of Micro-Scale Coaxial Rotor Aerodynamics in Hover," Ph.D. Dissertation, Department of Aerospace Engineering, University of Maryland at College Park, 2009.

44 Gupta, V., "An Investigation of Quad Tilt Rotor Aerodynamics in Helicopter Mode," Ph.D. Dissertation, Department of Aerospace Engineering, University of Maryland at College Park, 2005.

45 Jose, A. I., "Investigation Into the Aerodynamics of Swashplateless Rotors Using CFD-CSD Analysis," Ph.D. Dissertation, Department of Aerospace Engineering, University of Maryland at College Park, 2012.

46 Amiraux, Mathieu, "Numerical Simulation and Validation of Helicopter Blade-Vortex Interaction Using Coupled CFD/CSD and Three Levels of Aerodynamic Modeling," Ph.D. Dissertation, Department of Aerospace Engineering, University of Maryland at College Park, 2014.