

Unsteady Propeller Hydrodynamics

by

Dirk H Renick

B.S., Naval Architecture (1992)
United States Naval Academy
Naval Engineer's Degree (1999)
Massachusetts Institute of Technology
S.M., Mechanical Engineering (1999)
Massachusetts Institute of Technology

Submitted to the Department of Ocean Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Ocean Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

©2001 Dirk Renick. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part.

Author ...

.....
Department of Ocean Engineering
May 22, 2001

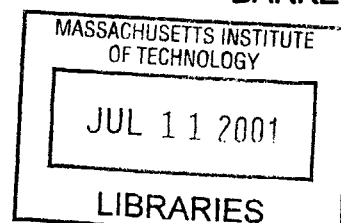
Certified by

Justin E. Kerwin
Professor of Naval Architecture
Thesis Supervisor

Accepted by

Henrik Schmidt
Professor of Ocean Engineering
Chairman, Department Committee on Graduate Studies

BARKER



Unsteady Propeller Hydrodynamics

by

Dirk H Renick

Submitted to the Department of Ocean Engineering
on May 22, 2001, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Ocean Engineering

Abstract

One of the main problem affecting modern propulsor design engineers is the ability to quantitatively predict unsteady propeller forces for modern, multi-blade row, ducted propulsors operating in highly contracting flowfields. Current algorithms provide valuable insight into qualitative trendlines for these modern designs. This thesis has focused on the more accurate quantitative force prediction by introducing more physical modeling into the numerical computations, using more accurate analytical representation of continuous physical phenomena, whilst not increasing the usage complexity for the desktop engineer.

This thesis developed several novel algorithms and techniques and applied them to build an evolutionary, general vortex-lattice lifting-surface propeller code. First, a general method to track the trajectory of individual wake singularity sheets and compute their influence velocities was evolved which reduces computation time, and dramatically increases the accuracy of the unsteady blade loading problem.

To improve the general coupling technique between potential-based propeller codes and volumetric Reynolds-Averaged Navier-Stokes codes, a general analytic method based upon an elliptic integral method for the velocity induced by a vortex ring of unsteady harmonic strength to compute of the time-averaged induced velocities in the swept volume of the propeller was introduced which is more accurate, as demonstrated in model problems, and more robust, as indicated by improved convergence and accuracy in a fully three dimensional propeller code. A discretized geometric technique was also created to internalize the coupling routines, making the code more robust, while decreasing the computation burden over current methods.

Finally, a higher order quadratic influence function technique was implemented within the wake to more accurately define the induction velocity at the trailing edge which has suffered in the past due to lack of discretization. These propeller program enhancements were fitted into a fully functional version of the Propeller Unsteady Forces (PUF)-series code, and coupled with a three dimensional RANS code.

Thesis Supervisor: Justin E. Kerwin
Title: Professor of Naval Architecture

Acknowledgments

There are an innumerable number of people to whom I am indebted beyond words for their patient counsel, guidance, encouragement and discouragement, inquisitive criticisms, and humbling intellect. It is their contributions which come out most prominently in this text. I serve merely as a scribe, translating their creative insights into numerical processes and implementation.

My wife Naoko has been my constant friend, counsellor, confidante, and check upon my sometimes overly tangential thoughts. Michael Griffin stepped in every time she wasn't there.

All of the MITHL Propnuts served as constant sounding boards and inspirations in the achievement of true propeller nirvana: Jake, Todd, Rich, Nick, Gerard, Chris, Mark, and Bill. And while all of my MIT professors showed me that nothing is that hard, Prof.'s Sclavounos, Drela, and Peraire, pushed me to achieve the really hard stuff.

Dr Chao-Ho Sung showed me what one man really can accomplish. CDR Mark Welsh illuminated the path, and LCDR(ret) Cliff Whitcomb modelled the research naval officer. All the Course 13-A CO's taught me a lot and filled my tool belt: CAPT(ret) Alan Brown, CAPT(ret) Dennis Mahoney, and CAPT Chip McCord. RMCS (SEAL) Foresman taught me to ask for what you want. Repeatedly.

The author is indebted to the Office of Naval Research for their indirect support of this thesis through grant N00014-97-1-0108, 'Computation of Propeller Induced Maneuvering Forces', Dr. Edwin Rood, Program Manager.

Contents

1	Thesis Overview and Motivation	17
1.1	Thesis Goals	17
1.2	Unsteady Propeller Forces	18
1.3	Historical Context	19
1.4	Summary of Computational Improvements	19
1.5	Summary of Results	21
2	Vortex-Lattice Lifting-Surface Formulation	23
2.1	Lifting Line Theory	23
2.2	Vortex-Lattice Theory	24
3	Unsteady Vortex-Lattice Lifting-Surface Formulation	27
3.1	Kutta Condition	28
3.2	Unsteady Modeling	29
3.3	Discrete Boundary Value Problem	30
3.4	Higher Order Wake Representation	34
3.4.1	Linear Variation in Wake Loop Strength	34
3.4.2	Quadratic Variation in Wake Loop Strength	35
4	Force Free Wake Alignment	39
4.1	Applications for Non-Uniform Wakes	40
4.2	Wake Modeling in a Propeller Code	42
4.3	Approach to Wake Alignment	42

4.4	Wake Alignment Model Problems	43
4.5	Wake Tracking Methodology	45
4.5.1	Wake Sheet Generator Line	46
4.5.2	Rotation of a Point about an Arbitrary Axis	46
4.6	Hub Image Placement	48
4.7	Aligned Wakes with Prescribed Effective Inflow	50
4.7.1	Once per Revolution Change in Axial Velocity	51
4.8	Wake Lattice Spacing	52
4.9	Higher Order Wakes	54
4.9.1	Higher Order Wake Loop Scheme	56
4.9.2	Convergence with Subdivided Wake Loops	56
4.9.3	Solution Validation with Subdivided Wake Loops	58
4.10	Validations of Aligned, Subdivided Wake Scheme	59
4.10.1	Convergence Test of Aligned, Subdivided Wake Scheme	60
4.10.2	Propeller 4119 in Non Uniform Axial Inflow	62
5	Coupled Algorithms	65
5.1	Time-averaged Induced Velocity	65
5.2	Velocity Induced by a Vortex Segment	67
5.2.1	Discrete Calculation of the Velocity Induced by a Conic Section	67
5.3	Velocity Induced by a Ring Vortex of Harmonically Varying Strength	68
5.3.1	Velocity Induced by Constant Strength Vortex Ring	68
5.3.2	Non-constant Strength Vortex Ring Induced Velocity	69
5.3.3	Conic Sections by Successively Integrating Vortex Rings	70
5.3.4	Discrete and Analytic Model Comparison	70
5.4	Induced Velocity Methodology Implementation	74
5.4.1	Numerical Validations	74
5.5	Grid Intersection Methodology	75
6	Coupled Validations	79
6.1	Propeller 4119 Validations	79

6.1.1	UNCLE-3D Coupling with Propeller 4119	80
6.1.2	IFLOW-41 Coupling with Propeller 4119	83
6.2	Propeller 4119 at 10 Degree Inclination	87
6.3	Propeller 4679 at 7.5 Degree Inclination	87
6.3.1	Nominal Flow Convergence	92
6.3.2	Propeller Solution Convergence	94
6.3.3	Aligned and Non-Aligned Wake Results	94
6.3.4	Blade Loading Numerical Comparison	95
6.3.5	Experimental Comparison	95
7	Conclusions	101
7.1	The Future	102
A	Validation Techniques for PUF Type Codes	103
A.1	Coupled Validation	103
A.1.1	Effective Inflow Equals 1.0 Test	103
A.1.2	V_θ Test	104
A.1.3	Blade Isolation Test	104
A.1.4	Coupled Problems	105
A.1.5	Coupling Code Non-Dimensionalizations	105
B	Potential - RANS Coupling	109
B.1	Effective Inflow	109
B.2	The Coupled Hull Flow Problem	110
B.3	Implementation in RANS Formulation	110
C	Propeller Numeric Routines	113
C.1	Romberg Integration	113
C.2	Tri-linear Interpolation	114
C.3	Elliptical Streamline Grids	115
C.3.1	Structured Grid Generation	117
C.3.2	Techniques of Structured Grid Generation	117

C.4 Poisson, or Elliptical, Grid Generation 118
C.4.1 Solution Techniques for the Elliptical Grid Equations 121
C.4.2 Forced Functions 122

D Propeller Terminology 125

D.1 Abbreviations 125
D.2 Propeller Terms 127

List of Figures

2-1	Lifting line representation of a finite wing.	24
2-2	A vortex-lattice lifting-surface propeller blade and wake model.	26
3-1	Chord-wise vortex lattice model	29
3-2	Timestepping and solution for unsteady blades.	30
3-3	Higher Order Wake Panel	35
4-1	Non-uniform inflow composed of aximuthal step	44
4-2	With the propeller operating in a 10 degree inclined inflow, the wake properly follows the inflow angle.	44
4-3	As the wake convects downstream aligned with the local flowfield, it forms a cylindrical tube.	45
4-4	Notation used in specifying how a point in the PUF reference frame is rotated about an arbitrary vector	46
4-5	Terminology used in determining the position of the wake hub image lattice point positions.	49
4-6	Velocity components of a 10 degree inclined flow.	50
4-7	Blade harmonic force phase and magnitude shifts with aligned wakes.	51
4-8	Blade harmonic force phase and magnitude shifts with aligned wakes.	52
4-9	4119 circulation at the seven-tenths radius for axially non-uniform inflow.	53
4-10	Convergence of the propeller blade solution using previous PUF-14.	55
4-11	Subdivision scheme for creating quasi-higher order wake panels via panel subdivision.	57

4-12	Less computational effort is required for the subdivided algorithm presented here with similar accuracy.	59
4-13	Validation of PUF14 using the wake subdivision scheme with PUF14 without the wake subdivision scheme in a uniform, axial inflow. . . .	60
4-14	Convergence rate of new wake scheme in non-uniform inflow.	61
4-15	Convergence test of PUF with fully aligned, subdivided wakes	62
4-16	Pitch angle of the wake sheets in non-uniform inflow.	63
5-1	Conic section made up of discrete vortex segments	68
5-2	Model problems to highlight the discretization required to reproduce analytic results.	71
5-3	Induced velocity upon a field point as it traverses through 360 degrees around a conical vortex sheet	72
5-4	Induced velocity upon a field point as it pierces a conical vortex sheet	73
5-5	Comparison of induced velocity calculated in steady propeller analysis and unsteady propeller analysis codes.	75
5-6	Comparison of induced velocity calculated in unsteady propeller analysis code via discrete and analytic methods.	76
5-7	Grid subdivision to transfer forces between PUF and RANS.	77
6-1	Coupling overview of codes	80
6-2	Views of DTMB Propeller 4119.	80
6-3	Numerical RANS Grid used in UNCLE-3D RANS code.	81
6-4	Differences in effective velocity prediction for analytic and discrete time-averaged induced velocity calculations.	82
6-5	Comparison of blade circulation from coupled PUF-RANS and stand alone PUF.	83
6-6	Infinite shaft grid used in IFLOW41	84
6-7	Pressure residual history for IFLOW41 coupled with PUF	85
6-8	4119 propeller lattice convergence with IFLOW-41.	86

6-9	Propeller 4119 at 10 degree inclined angle, showing the now inclined wakes.	88
6-10	Propeller 4119 at 10 Degree inclined angle showing the convection of swirl downstream of the propeller swept volume.	88
6-11	Magnitude of the mean C_P of Propeller 4119 at 10 Degree inclined angle.	89
6-12	Magnitude of the First Harmonic of Propeller 4119 at 10 Degree inclined angle C_P	90
6-13	Phase of the First Harmonic of Propeller 4119 at 10 Degree inclined angle C_P	91
6-14	Computational model of the DTMB downstream driven propeller shaft	92
6-15	Log of pressure residuals for nominal, axial flow around DTMB downstream driven shaft.	93
6-16	Nominal flow around DTMB downstream driven shaft, zero degree inclination.	93
6-17	Propeller 4679 blade lattice convergence study.	94
6-18	Propeller 4679 Steady C_P at $r/R = 0.5$	96
6-19	Propeller 4679 Steady C_P at $r/R = 0.5$	96
6-20	Propeller 4679 Steady C_P at $r/R = 0.7$	97
6-21	Propeller 4679 Steady C_P at $r/R = 0.9$	97
6-22	Propeller 4679 First harmonic C_P at $r/R = 0.5$	98
6-23	Propeller 4679 First harmonic phase of C_P at $r/R = 0.5$	98
6-24	Propeller 4679 First harmonic C_P at $r/R = 0.7$	99
6-25	Propeller 4679 First harmonic phase of C_P at $r/R = 0.7$	99
6-26	Propeller 4679 First harmonic C_P at $r/R = 0.9$	100
6-27	Propeller 4679 First harmonic phase of C_P at $r/R = 0.9$	100

List of Tables

4.1	Aligned and Non-aligned wake results for 10 degree angle of inclination	50
4.2	Unsteady blade forces for axially non-uniform effective inflow.	52
4.3	Affect of NTSR on subdivided wake scheme results.	60
6.1	Propeller 4119 coupled with IFLOW and UNCLE results	82
6.2	Predicted 1st Harmonic force results for P4679 at 7.5 degree inclined angle.	95

Chapter 1

Thesis Overview and Motivation

One of the main problem affecting modern propulsor design engineers is the ability to quantitatively predict unsteady propeller forces for modern, multi-blade row, ducted propulsors operating in highly contracting flowfields. Current algorithms provide valuable insight into qualitative trendlines for these modern designs. This thesis has focused on the more accurate quantitative force prediction by introducing more physical modeling into the numerical computations, using more accurate analytical representation of continuous physical phenomena, whilst not increasing the usage complexity for the desktop engineer.

1.1 Thesis Goals

The goal of this thesis is to present improved algorithms in production quality software that allows naval engineers to more accurately and more quickly predict the unsteady forces and moments on advanced propulsors. Accomplishing this goal will allow more flexibility in including advanced unsteady analysis as routine part of the exploration of the design space presented to an engineer at the outset of any project. In fact, the bounds upon the design space, at the outset, are more determined by computation setup, running complexity and computing time, rather than other inherent challenges.

The goal of improved algorithms led to the development of analytic routines to replace discretized routines which were shown to have slow convergence rates. In

other cases, unstable algorithms were replaced with more stable algorithms. And physical insights allowed for major speed-ups with no loss of accuracy.

The goal of production quality software means that internal assumptions must be clearly stated, converged and validated, or offered to the user to alter in an easy-to-use manner. This is accomplished in this thesis through the use of input files for controlling the code, vice requiring users to recompile the code frequently.

The final validation of these goals is in the comparison of the results to previous unsteady propeller codes, and experimental results.

1.2 Unsteady Propeller Forces

The accurate prediction of the steady and unsteady propulsor forces arising from operation of the propulsor in non-uniform wakes is an important aspect in the preliminary design or later stage analysis of any propulsor. By unsteady it is meant that the flow appears unsteady to a bug which sits upon the leading edge of the propeller. To an observer in the global reference frame, the flow will appear quite steady, but the bug will see large changes in angles of attack and other inflow phenomena traveling through 360 degrees.

In a practical sense, the predicted unsteady propeller blade forces lead to numerous other design considerations:

1. Shaft bearing and stern tube seal requirements.
2. Determination of acceptable vibration and radiated acoustic levels.
3. Steady off-axis maneuvering forces which must be counteracted by other control surfaces and drive seal and stern strength considerations.
4. Unsteady cavitation inception, which again affects noise and blade corrosion.

1.3 Historical Context

This thesis follows from and complements a long line of research conducted into the prediction of propeller forces arising during operation of a rotating propeller in an unsteady flowfield. While the propeller code is the heart of this thesis, the strong coupling between the propeller and the surrounding flow field must be accounted for by the use of an external flow solver. Thus, the issues surrounding accuracy, ease of use, and speed of use, associated with coupling the propeller and flow solvers are addressed.

The propeller code used as a starting point for this research is the Propeller Unsteady Forces (PUF) code developed at the Massachusetts Institute of Technology (MIT) originally by Kerwin and Lee [16]. The vortex-lattice lifting-surface theory which forms the basis for this research is based up the theory presented by Kerwin and Lee [16]. Recent work by Warren [30] highlighted the need for improved modeling of the wake near the trailing edge, which directly led to the work presented in Section 4.9. Work by Kinnas [17] showed that for contrived inflows wake alignment issues were important in unsteady blade force calculations.

The Reynold's-Averaged Navier-Stokes code used for this research is IFLOW developed at David Taylor Model Basin by Sung [27],[28]. The RANS code UNCLE, developed at Mississippi State University was also used to verify the validity of the approaches taken here.

1.4 Summary of Computational Improvements

This thesis has resulted in significant improvements to the unsteady analysis of propulsors by providing more accurate, more computationally efficient, and more robust computer analysis programs.

Propeller Unsteady Forces (PUF) Program

This Research	Previous Methodology
Propeller wakes fully aligned with the local inflow	Propeller wakes aligned with circumferential mean inflow and are not force free
New discretization of blade loops presented which allows for higher order wake loops downstream of the blade	Blade and wake loops separate in formulation
Wake loops discretized to guarantee wake grid independent solutions	Wake loop discretization tied to blade time step discretization requiring fine time discretizations to achieve wake grid independent solutions
Blade mean camber surface generated in elliptically solved-for flowfield	Blade mean camber surface generated along actual flow field stream surfaces, instabilities could result in blades outside flow domain
Blade forces output in format suitable for RANS code body force use	Suite of 3 post-processor programs used to transfer PUF forces to a RANS code

Propeller Unsteady Forces (PUF) Program and RANS Program Coupling

This Research	Previous Methodology
Analytic method for prediction of induced velocity via ring vortices of unsteady harmonic strength	Discrete algorithm via Biot-Savart Law
Fast geometric subdivision algorithm for transferring forces	Polygon overlap methodology
Trilinear interpolation used to transfer velocity field information between PUF and RANS domains	Cubic interpolation scheme, which is unstable in regions of high gradients

1.5 Summary of Results

This thesis has produced an improved vortex-lattice lifting-surface analysis code, PUF, as well as a revised RANS code, IFLOW, which, when used in tandem, give fast, accurate solutions to the most complex of geometries.

- The new PUF is a vortex-lattice lifting-surface propeller code which incorporates fully-three dimensional, independently aligned wakes.
- IFLOW is a three dimensional RANS code which internally couples with a propeller blade force solver.

Both of these codes can be run independently, or coupled with a single change to an input file. This flexibility removes several degrees of freedom from the analysis process and allows engineers to concentrate on the engineering design issues, rather than the analysis methodology issues.

Chapter 2

Vortex-Lattice Lifting-Surface Formulation

Vortex-lattice lifting-surface theory models a propeller as a singularity distribution in free space, and imposes the kinematic boundary condition at selected points on the foil surface requiring there to be no flow through the propeller blade at that point.

2.1 Lifting Line Theory

The roots of vortex-lattice lifting-surface theory lie actually in the development of the lifting line theory early in the 20th century [5]. In lifting line theory, a wing is represented as a two dimensional line of bound vorticity, as shown in Figure 2.1. Figure 2.1 shows a wing modeled as a lifting line constant strength vortex. Due to Kelvin's theorem, which briefly states that the circulation within a material volume is constant over time, the lifting line of vorticity which lies along the wing must extend downstream of the wing where there is a line of vorticity of equal and opposite magnitude downstream of the lifting line wing. Taking a material volume as any simple enclosed volume which surrounds the convex hull of the lifting line wing and its trailing vortex system (composed of two trailers and the starting vortex), we see that Kelvin's theorem is satisfied since the sum of all the circulation due to the lifting line is zero. A more realistic vortex-lattice lifting line representation of the wing which

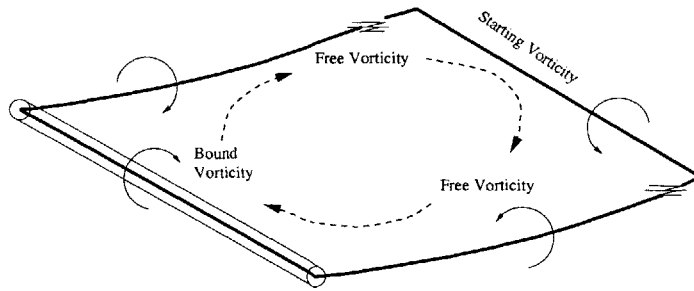


Figure 2-1: Lifting line representation of a finite wing.

has non constant strength along the span can be implemented by stacking individual vortex segments shown in Figure 2.1 end on end.

A mathematical requirement of the lifting line wing representation is that the circulation on the tips of the wings be zero. A completely rigorous mathematical derivation is presented in both Newman [22] and Kerwin [14]. Due to this singularity requirement, the circulation distribution over the span of a lifting line is usually represented as a sine series, as first demonstrated by Glauert [5].

The lifting line wing has been implemented as a vortex-lattice lifting-line wing by Kerwin [14]. The lifting line wing theory is directly transferable to propellers, and was implemented as the Propeller Lifting Line computer program by Coney [2],[3]. The problem with the lifting line representation of the blade, though, is that it can not represent swept wings (or skewed blades) due to the increasing singularity present in the Cauchy principal value integral equation. The lifting-surface theory removes this singularity by spreading the concentrated bound vortex over a chord.

2.2 Vortex-Lattice Theory

In a traditional propeller lifting surface propeller code, a grid lattice is placed on the blade mean camber surface, the hub and duct (if present) and the trailing wake system [16]. Each lattice segment is assigned a strength of vorticity. On the solid body surfaces, such as the blade and hub, a control point is placed near the center of the grid lattice. The strength of each vortex lattice segment is assigned by satisfying the

kinematic boundary condition that the flow must be tangent to the propeller, hub, and duct surfaces at every control point. A wake lattice is traditionally found by laying the lattice upon a helicoidal surface, where the pitch is set by the hydrodynamic pitch possibly with induced effect corrections.

Mathematically, the propeller problem involves a simple matrix equation. By attacking the geometry of the problem, an influence matrix is formulated which gives the velocity induced by a unit strength of vorticity along every vortex lattice upon every control point – an $[INF]$ luence matrix.

$$[INF] \left[\vec{\Gamma} \right] = \vec{V}_{induced} \quad (2.1)$$

The symmetry of the blades and wakes allows for great computational simplification through similarity principles. Next, to satisfy the kinematic boundary condition, the physics of the problem dictate that the component of the induced velocity normal to the blade, when added to the component of the effective inflow velocity normal to the blade must be zero.

$$\left[[INF] \left[\vec{\Gamma} \right] + V_{eff} \vec{v} \right] \cdot \hat{n} = 0 \quad (2.2)$$

Equation (2.2) is the heart of the propeller lifting surface code.

Equation (2.2) highlights the fact that a valid propeller calculation requires both the correct geometrical position of the lattice and the correct effective velocity everywhere on the blade surface.

The propeller forces resulting from the vorticity and source distributions are calculated from the Kutta-Joukowski and Lagally theorems, respectively. Unsteady forces arise from the unsteady pressure term in the unsteady Bernoulli equation. A Lighthill leading edge suction force correction is applied to these forces in a chordwise fashion, and the propeller’s sectional drag is calculated either based on strip wise two dimensional empirical drag coefficients, or a stripwise 2-D integral boundary layer calculation [9]. Cavitation effects can be included through the use of a pressure and velocity indication which finds the detachment and re-attachment points of the cavity

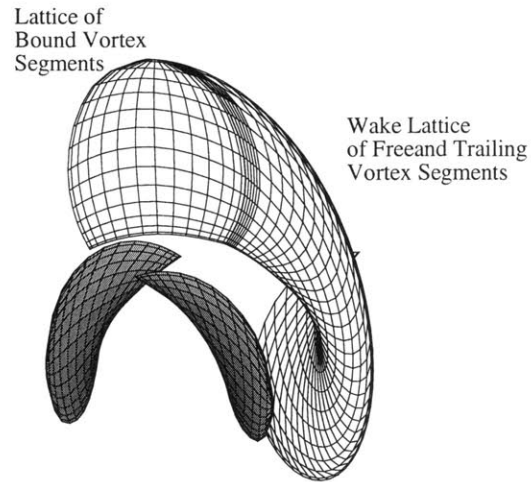


Figure 2-2: A vortex-lattice lifting-surface propeller blade and wake model.

sheet. The computer code developed in this thesis uses only empirical sectional drag coefficients and does not account for cavitation effects.

Chapter 3

Unsteady Vortex-Lattice Lifting-Surface Formulation

In an unsteady, vortex-lattice lifting-surface propeller code the requirement to satisfy both a Kutta condition and Kelvin's theorem drives scientists to devise ever new variations on a numerical theme to align theory with observation.

The main feature of the *lifting surface* formulation requires a representation of a lifting body.

- Camber effects are accounted for by modeling the lifting surface on the mean camber surface.
- Induced effects due to both bound and shed vorticity are calculated.
- Angle of attack effects are found by properly modeling the three dimensional surface rotating through the fluid.

The paper thin *vortex lattice* representing the blade is set upon the mean camber surface. Consequently provisions must be made for the two-dimensional effects of the foil:

- Thickness. Can be accounted for through the distribution of source singularities alongside the vortex singularities.

- Leading edge suction. Foils not at their shock free entry angle produce leading edge suction. This can be handled through the use of an analytically correct 2-D solution, applied in three dimensions, as shown by Lan [19].
- Boundary layer (shear) drag. Any number, or combination of, empirical, semi-empirical, or 2-D/3-D analytic methods can be implemented [9].

3.1 Kutta Condition

In general, the Kutta condition can take many forms:

1. Zero pressure jump at the trailing edge.
2. Finite velocity at the trailing edge.
3. Smooth flow tangent to the blade at the trailing edge [22].
4. A wake singularity sheet extends downstream of the blade where the singularity strength is the difference in singularity strengths on the upper and lower surfaces of the blade directly upstream of the trailing edge

The numerical implementation of the Kutta condition then falls classically into either an explicit or an implicit form [16].

In the explicit formulation of the Kutta condition the strength of the last few bound vortices on the blade are manipulated to satisfy a condition imposed by the researcher. For steady flow, the condition that the bound vortex strength vanish at the trailing edge is imposed. Numerically, the imposition of this extra condition requires either another unknown, or that one of the constraint equations be dropped.

In the implicit formulation of the Kutta condition, judicious placement of the blade and wake vortices are assumed to satisfy the Kutta condition. The hypothesis can be assured by placing the final downstream control point along the trailing edge. Since the numerics of the discretized boundary value problem require there to be no fluid velocity normal to the blade at any control point, the Kutta condition is assured. Further derivations and supporting arguments have been shown in the literature.

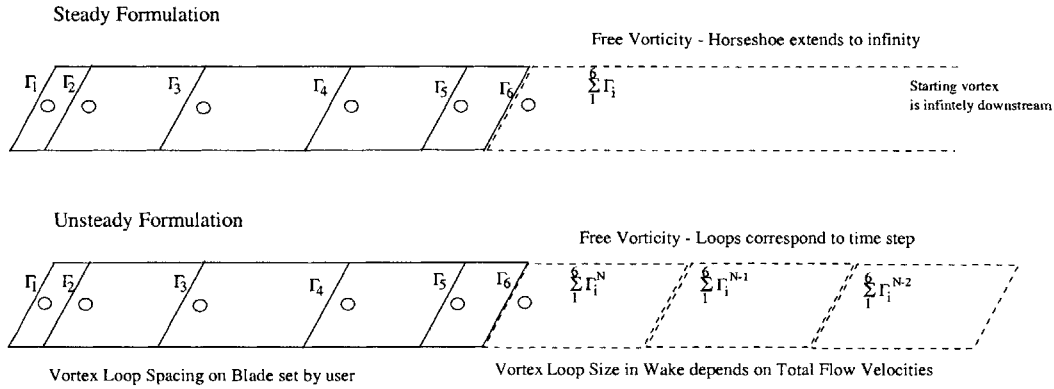


Figure 3-1: These two figures show the major difference between steady (upper figure) and unsteady (lower figure) vortex lattices formulations.

3.2 Unsteady Modeling

The major difference between a steady and unsteady formulation is the shedding of span-wise vortex segments of free vorticity which account for the changes in blade circulation as the propeller traverses three hundred and sixty degrees. This is shown in Figure 3-1. In a steady code the wake vortex is a single horseshoe extending to infinity. In an unsteady code the wake vortex loop strength changes due to the change in blade shed vorticity at each blade time step. So along a constant spanwise series of wake vortex loops, there are unequal amounts of vorticity. The difference in strength of the trailing vortex segments (*i.e.*, the vortex lattice segments in the streamwise direction) require that spanwise vortex segments, called shed vortex segments, be introduced which connect the trailers to satisfy Kelvin.

In the unsteady code, the unknowns include not only the vortex strength on the blades, but also the first wake loop. This is because as the blade advances through the fluid from time step N to time step $N + 1$ the amount of circulation shed into the wake to satisfy Kelvin is unknown. So the general solution procedure, as shown in Figure 3-2 is

1. Starting from a known solution, advance the wakes downstream by one loop. Note that we make the assumption that the loops are sized such that one loop is equivalent in size to the distance a fluid particle would travel in one time step.

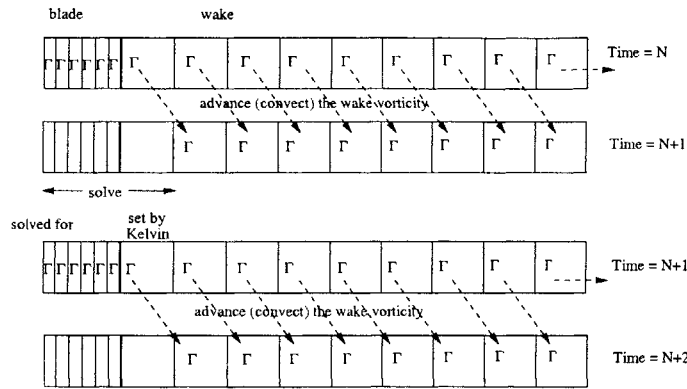


Figure 3-2: This figure shows a slice along the chordwise direction (constant span) for a blade, and the steps required to advance the blade one time step.

2. Solve the boundary value problem on the blade surface.
3. Place the correct circulation in the first wake loop based on the definition of Kelvin's theorem.
4. This then represents another known solution, and the process repeats itself.

One requirement to implement this scheme, is that the vortex lattice spacing in the wake is directly dependent upon the time step set by the user. The blade vortex spacing, though, is set in a different manner by the user since there is no intra-problem dependency between the number and relative size of the blade and wake vortex lattices. This gives rise to a sharp gradient in grid density at the trailing edge of the blade as the solution transitions from the blade to the wake lattice. This problem is further accentuated by the fact that the blade lattice is cosine spaced in the chordwise direction.

This thesis formulates an implicit Kutta condition which advantageously satisfies Kelvin's theorem. Start with the governing discretized boundary value problem. At all control points on the blade surface, a Neumann boundary condition is enforced which says that there can be no flow normal to the blade. The fluid velocity at any control point is the sum of the vortex induced flow due to the vorticity on the blade, the vortex induced flow due to the vorticity in the wake, and the effective inflow.

3.3 Discrete Boundary Value Problem

$$\sum_{j=1}^{N_b} (\Gamma_b)_j B_{i,j} + \sum_{k=1}^{N_w} (\Gamma_w)_k W_{i,k} = -V_i \quad (3.1)$$

where	N_b	number of blade vortex loops
	$(\Gamma_b)_j$	circulation strength of blade vortex loop j
	$B_{i,j}$	Influence of unit circulation strength in blade j on control point i
	N_w	number of wake vortex loops
	$(\Gamma_s)_k$	circulation strength of wake loop k
	$W_{i,k}$	Influence of unit circulation strength in wake loop k on control point i
	V_i	Effective velocity at control point i

In the unsteady formulation, a simple notation scheme is introduced modifying Equation 3.1 which states that the end state of the problem is to find the solution at the next time step, $N + 1$.

$$\sum_{j=1}^{N_b} (\Gamma_b)_j^{N+1} B_{i,j} + \sum_{k=1}^{N_w} (\Gamma_s)_k^{N+1} W_{i,k} = -V_i \quad (3.2)$$

From the physics of the problem, it is known that the vorticity present in the wake convects with the local velocity. By sizing the wake panels such that the size of one wake panel is the distance traveled by the blade in one time step, we can state with certainty that the vorticity present on the $k + 1$ panel at time $N + 1$ is the vorticity which existed on the k^{th} panel at time N .

$$(\Gamma_s)_{k+1}^{N+1} = (\Gamma_s)_k^N \quad \text{for } k > 0 \quad (3.3)$$

Since the shed vorticity present in the wake is known at time N , this means that the wake influence velocities are known due to all wake panels except the $k = 1$ panel.

$$\sum_{j=1}^{N_b} (\Gamma_b)_j^{N+1} B_{i,j} + (\Gamma_s)_1^{N+1} W_{i,1} = -V_i - \sum_{k=2}^{N_w} (\Gamma_s)_k^N W_{i,k-1} \quad (3.4)$$

To account for the vorticity present on the first wake panel, $(\Gamma_s)_1^{N+1}$, we make use of Kelvin's theorem. When applied to a material contour which encompasses the blade and the entire wake, we can see that any change in vorticity on the blade over a given time interval must be balanced by a change in vorticity in the wake.

$$\begin{aligned}\frac{\partial \Gamma}{\partial t} &= 0 \\ \frac{\partial \Gamma_b}{\partial t} + \frac{\partial \Gamma_s}{\partial t} &= 0\end{aligned}\tag{3.5}$$

The first observation to make concerning Equation 3.5 is that integrating all terms with respect to time t gives the absolute change in vorticity. That being said, the second observation to make is that the only place the vorticity in the wake changes within the material volume (which by the definition of a material volume convects with the flow) is on the first wake panel.

$$\int_N^{N+1} \frac{\partial \Gamma_b}{\partial t} dt + (\Gamma_s)_1^{N+1} = 0\tag{3.6}$$

Applying a very low order integration scheme, such as a one-sided trapezoidal rule for the first term in Equation 3.6

$$\begin{aligned}\int_N^{N+1} \frac{\partial \Gamma_b}{\partial t} dt &= (T_b)^N + \Delta T_b \\ T_b &= \sum \Gamma_b \\ \Delta T_b &= \sum (\Gamma_b)^{N+1} - \sum (\Gamma_b)^N \\ \int_N^{N+1} \frac{\partial \Gamma_b}{\partial t} dt &= \sum (\Gamma_b)^{N+1}\end{aligned}\tag{3.7}$$

Substituting Equation 3.7 into Equation 3.6 gives

$$\Gamma_{s1}^{N+1} = - \sum_{j=1}^{N_b} (\Gamma_b)_j^{N+1} \quad (3.8)$$

Substituting Equation 3.8 into Equation 3.4

$$\sum_{j=1}^{N_b} (\Gamma_b)_j^{N+1} B_{i,j} - \sum_{j=1}^{N_b} (\Gamma_b)_j^{N+1} W_{i,1} = -V_i - \sum_{k=2}^{N_w} (\Gamma_s)_k^N W_{i,k-1} \quad (3.9)$$

Equation 3.9 is the final discretized boundary value problem which can be solved numerically. The important things to note about Equation 3.9 is that all the unknowns on the left hand side are the Γ_b of interest at the next time step. Further, the low order integration means that interestingly enough that first wake panel does not influence the right hand side of the equation.

Combining the two terms on the left hand side of Equation 3.9 reduces it to the more usual form of $Ax = b$.

$$\sum_{j=1}^{N_b} (\Gamma_b)_j^{N+1} [B_{i,j} - W_{i,1}] = -V_i - \sum_{k=2}^{N_w} (\Gamma_s)_{k-1}^N W_{i,k} \quad (3.10)$$

Physically this means that the blade influence loops include the influence of the first wake panel. Note that the influence of the first wake panel is *not* found on the right hand side of Equation 3.10.

This formulation differs from previous unsteady formulations in the placement of the first shed vortex segment. Previous researchers have placed the first shed vortex segment one quarter downstream of the foil trailing edge, presumably in an attempt to avoid the trailing edge singularity. This thesis has shown that the correct placement of the first shed vortex is one complete time step downstream of the blade trailing edge.

3.4 Higher Order Wake Representation

One method to increase the accuracy for a given problem is to increase the order of the discretized representation of the physical, analog, system. For a vortex-lattice code, this strategy leads to a higher order representation of the singularity distribution over a given vortex loop since in its most basic form, a vortex loop is a constant strength dipole sheet. Practically speaking, this is implemented by subdividing a vortex loop and assigning a spatially varying singularity strength to the subdivided loops based on the singularity value at the boundary of the original loop.

For a lifting surface code, there is no requirement to represent the entire transition wake as higher order loops, because most of the loops are far downstream from the control points of interest. To save computational time, then, only the first wake panel downstream of the trailing edge is represented by a higher order singularity distribution. This solution technique assumes that the blade solution is affected most by the order of the first wake panel since the first wake panels adjacent to the trailing edge have the largest influence upon induced velocities at the boundary value problem control points.

The theory and results are presented below for a linear variation, and 2nd order variation in singularity strength.

3.4.1 Linear Variation in Wake Loop Strength

The first improvement upon the low order scheme presented in Equation 3.10 is to model the strength of vorticity across the first wake panel as linearly varying across the panel. This is accomplished by breaking the first shed wake panel into several smaller panels as shown in Figure 3-3. Higher order wake panels are modeled by subdividing the first wake panel. The singularity strength of each sub panel can then be forced to conform to a higher order scheme: linear, quadratic or higher. Notice that the strength across the panel then depends upon the blade circulation which is

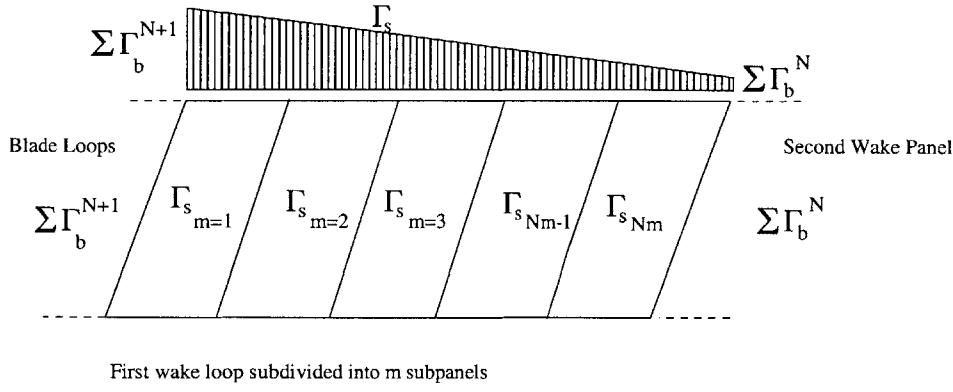


Figure 3-3: Higher order wake panel.

the unknown, and the previous timestep blade circulation, which is known.

$$\begin{aligned}\Gamma_{sm}^{N+1} &= (1 - m)\Gamma_{s0} + m\Gamma_{s1} \\ \Gamma_{sm}^{N+1} &= (1 - m) \sum_{j=1}^{N_b} \Gamma_{bj}^{N+1} + (m) \sum_{j=1}^{N_b} \Gamma_{bj}^N\end{aligned}\quad (3.11)$$

The variable m in Equation 3.11 represents the distance between 0 and 1 across the subdivided panel.

This changes the discretized boundary value problem as shown in Equation 3.12. Notice in Equation 3.12 that the influence of the first wake panel is accounted for on both the left hand side and right hand side of the equation. And that each subdivided panel influences each control point.

$$\sum_{j=1}^{N_b} (\Gamma_b)_j^{N+1} [B_{i,j} - \sum_{m=1}^{Nm} (1 - m)W_{m,1}] = -V_i - \sum_{k=2}^{N_w} (\Gamma_s)_k^N W_{i,k-1} - \sum_{m=1}^{Nm} \sum_{j=1}^{N_b} (\Gamma_b)_j^N (m)W_{m,1}\quad (3.12)$$

where $W_{m,1}$ is the influence of the m^{th} subpanel in the 1^{st} wake panel

3.4.2 Quadratic Variation in Wake Loop Strength

The second improvement upon the low order scheme presented in Equation 3.10 is to model the strength of vorticity across the first wake panel as quadratically varying.

This is accomplished by again breaking the first shed wake panel into several smaller panels as shown in Figure 3-3.

Examining a single chordwise strip of vortex loops extending from the blade and into the wake, this method states that the strength of vorticity in the wake panel varies quadratically in space downstream of the blade trailing edge. Therefore, the strength of the vorticity is represented as a quadratic equation as shown in Equation 3.13.

$$\Gamma_{sm} = as^2 + bs + c \quad (3.13)$$

where s = distance downstream of trailing edge
 a, b, c = polynomial coefficients

Because the wake panel spacing is constant, a 3×3 matrix can be formed to find the polynomial coefficients using the unknown Γ_{s0} at the trailing edge, and the known Γ_{s1} and Γ_{s2} on the second and third panels. When the coefficient matrix is inverted and the terms re-ordered to show the dependence on the various Γ_s 's, Equation 3.14 results. This is equivalent to solving the Lagrange polynomial solution.

$$\Gamma_{sm} = \left(\frac{1}{2}s^2 - \frac{3}{2}s + 1\right)\Gamma_{sTE} + (-s^2 + 2s)\Gamma_{s1} + \left(\frac{1}{2}s^2 - \frac{1}{2}s\right)\Gamma_{s2} \quad (3.14)$$

By following up with the usual representation of the shed vorticity in terms of the blade vorticity at a given time step

$$\begin{aligned} (\Gamma_s)_{TE} &= \sum_{j=1}^{N_b} (\Gamma_b)_j^{N+1} \\ \Gamma_{s1} &= \sum_{j=1}^{N_b} (\Gamma_b)_j^N \end{aligned}$$

$$\Gamma_{s2} = \sum_{j=1}^{N_b} (\Gamma_b)_j^{N-1}$$

Inserting these definitions into Equation 3.10 and migrating the unknowns to the left hand side and the known to the right hand side leads to the final discrete equation shown in Equation 3.15.

$$\begin{aligned} & \sum_{j=1}^{N_b} (\Gamma_b)_j^{N+1} [B_{i,j} - \sum_{m=1}^{Nm} (\frac{1}{2}s^2 - \frac{3}{2}s + 1)W_{m,1}] = \\ -V_i - & \sum_{k=2}^{N_w} (\Gamma_s)_k^N W_{i,k-1} - \sum_{m=1}^{Nm} (\sum_{j=1}^{N_b} (\Gamma_b)_j^N (-s^2 + 2s) + \sum_{j=1}^{N_b} (\Gamma_b)_j^{N-1} (\frac{1}{2}s^2 - \frac{1}{2}s))W_{m,1} \end{aligned} \quad (3.15)$$

where s is the non-dimensional distance of the subdivided wake panel across the total wake panel

Chapter 4

Force Free Wake Alignment

Force free wake alignment means that the wake vortex lattice structures are aligned with the local fluid velocity. The wake is then force free by the Kutta-Jukowski theorem, Equation 4.1, since the cross product of the velocity and vorticity is zero.

$$\vec{F} = \rho \vec{V} \times \vec{\Gamma} \quad (4.1)$$

This chapter presents the development, implementation, and validation of a novel technique to align the vortex-lattice wake with the local flowfield. The engineering applicability to this line of research is presented in Section 4.1. A broad overview to wake modeling in the context of a propeller code is presented in Section 4.2. Section 4.3 presents the approach taken to accomplish three dimensional, independent wake alignment. Two model problems are shown in Section 4.4. Section 4.5 contains the details on wake growing. Hub image placement is shown in Section 4.6. A very relevant adjunct to wake alignment which makes the computational burden bearable is wake subdivision, shown in Section 4.8. A novel method to improve computation accuracy at reduced computational expense through judicious application of higher order techniques is presented in Section 4.9, where a higher order wake panel to resolve the influence velocity more accurately near the trailing edge singularity is created. Results for these new algorithms as implemented in a fully functional propeller code are found in Section 4.10.

4.1 Applications for Non-Uniform Wakes

Real world applications of propeller codes highlight two cases of interest where wake alignment contributes to the accuracy of the solution:

1. Propellers operating on inclined shafts and the non-axial flow which follows the aft buttock lines.
2. Maneuvering vehicles present strong cross flow contributions to the inflow seen by the propeller in its rotating reference frame.
3. Propellers operating within unsteady flowfields due to the presence of upstream appendages such as control surfaces, sails, struts, or an inclined shaft.

These real world flow conditions lead to unsteady propeller blade forces. The naval engineer then has the task of designing the propeller such that the outwardly measurable effects of the generated unsteady blade forces can be accurately modeled, in magnitude and variation, to within the design specifications.

There are two approaches that can be taken to align the propeller vortex-lattice wake with the local fluid velocities:

1. Align the vortex lattice wake using the combined affects of the blade and wake singularity-induced velocity and a given effective inflow.
2. Align the vortex lattice wake with a prescribed total inflow. The total inflow can be either recorded in an experimental test facility or found with a numerical simulation technique such as the RANS used in this research.

The first approach has been undertaken by Shih [25] and Kinnas [23] among others. This approach brings with it the requirements to lump vorticity at the tip of the wake sheet as the tips roll-up and to de-singularize the influence kernel function. This approach also has the more serious shortcoming that it does not properly account for the fact that the vortex-vortex interaction between the vorticity present in the inflow and the blade vorticity alters the inflow velocity profile [24]. This is explained more fully in Appendix B.1.

The second approach, that of aligning the the vortex lattice wake with a prescribed velocity field is undertaken in this thesis. This research utilizes the basic model of coupling with a Reynolds-Averaged Navier-Stokes flow solver [15] to give directly the wake field. There are two main reasons as to the relevance of this approach:

1. The general fidelity, usability, and availability of RANS codes makes them accessible to design engineers.
2. A RANS code can properly model the vortex-vortex interactivity, redistributing the vorticity present in the inflow in the presence of the propeller.

One possible shortcoming, which is not explored further, is that the time-averaged nature of the RANS solution means that the fluid velocities are time-averaged (or equivalently spatially-averaged), while the lifting-surface formulation requires local velocities to correctly solve the boundary value problem [15]. The details are explained in Appendix B. This assumption has proven valid in practice for axisymmetric flow, where circumferential mean flow averaged over the entire propeller swept volume, is substituted for local blade flow.

There are two major advantages of the approach taken here:

1. **Computational efficiency:** The calculation of the blade and wake singularity influence on the wake sheets is an extremely time consuming process which scales with the square of the number of singularity elements. Memory can not be traded against the computation, since the influences change at each alignment step.
2. **Compatibility with empirical measurements:** Empirical measurements of flow around bodies in the plane of the propeller provide either nominal (velocity measurements in the absence of the propeller) or total (velocity measurements with the effects of the propeller) velocities. For use in numerical validation, if total velocities are measured these can be exactly used as boundary conditions. Nominal values measured in an empirical test can be used to calibrate, or set the boundary conditions, for a numerical solver. In both cases, if the measurements

are taken far enough upstream of the propeller to negate the influence of the propeller (usually 2 propeller diameters for open propellers) these measurements are equal and can be used as a boundary condition on a volume solver. Note, however, that this method assumes that the propeller code is coupled with an external code which solves for the effective inflow while maintaining the proper velocities downstream of the propeller to convect the wake.

4.2 Wake Modeling in a Propeller Code

The wake scheme used as a starting point, is that devised by Greeley and Kerwin [7] as implemented in both the PBD-14 code [21] and the original PUF code [13]. In this formulation, the wake is divided into both a transition and ultimate wake model.

The transition wake is a fully three dimensional wake vortex lattice. The lattice of the transition wake matches the spanwise number of blade lattice element, while the streamwise number of vortex loops depends upon the time step set for the particular problem. The transition wake extends downstream from the blade a user-specified distance.

The ultimate wake model is composed of trailing helical vortices extending to infinity. That is, at each spanwise lattice position, a helical vortex is grown, with the same pitch as the tipwise element. The closed form solutions due to Hough and Ordway [8] as implemented by Leibman [20] provide accurate influence velocities due to the ultimate wake helical vortices. The helical vortices at a given radius are averaged at each blade position, to produce a steady set of helical vortices.

4.3 Approach to Wake Alignment

Two pieces are required to solve the wake alignment problem:

- individual wake growing.
- faster numerical algorithms to compensate for the increased number of influence calculations required.

The first piece is the need to align each wake vortex-lattice sheet convecting downstream from each blade with the local flowfield. Because the symmetry of the axisymmetric wake generation scheme is destroyed, there is an increase in computational and memory requirements (since each wake must be individually grown and has individual influence calculations and associated storage requirements). This leads to the requirement to divorce wake discretization from problem size. This is accomplished through the implementation of a pseudo-higher level wake discretization scheme using sub-divided wake vortex loops.

4.4 Wake Alignment Model Problems

The first problem for the two cases presented in Section 4.1 involves the propeller acting in the non-uniform flowfield produced by upstream appendages. As is clearly shown in the extreme case of a step change in inflow velocity in Figure 4-1 the pitch angle of the wake is directly affected by the non-uniformity in axial velocity. To quantify these effects upon the resulting propeller higher harmonic forces, a prescribed inflow composed of known harmonic inflows can be imposed, and then the resulting harmonic forces predicted by the propeller program with aligned and non-aligned wakes can be measured and compared.

The second problem of wakes convecting downstream at angle, as in the case of a maneuvering vehicle or a propeller mounted at a fixed angle of inclination with respect to the inflow as shown in Figure 4-2 , again results in a change to the wake pitch angle. The once per revolution change in pitch directly affects the blade solution. This change results from the azimuthal component of the velocity field convecting the wake. Conceptually, if the cartesian representation of any given velocity is translated to a cylindrical form, and radial velocities would of course not affect the pitch, but axial and azimuthal velocity components would change the pitch.

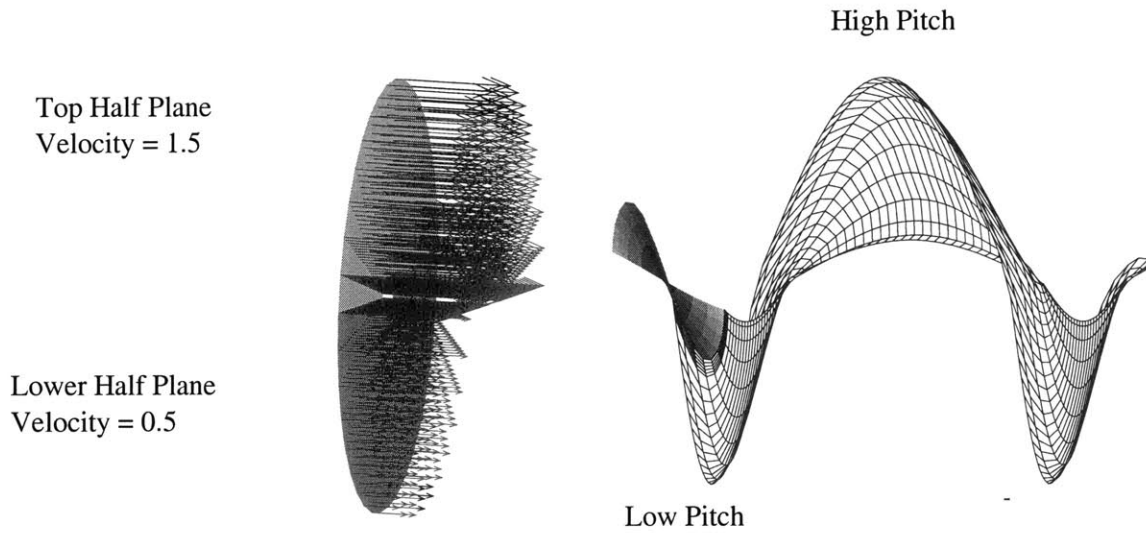


Figure 4-1: A non-uniform inflow directly affects the pitch angle of the wake sheets. Here the non-uniformity is produced by a step function change in the axial velocity.

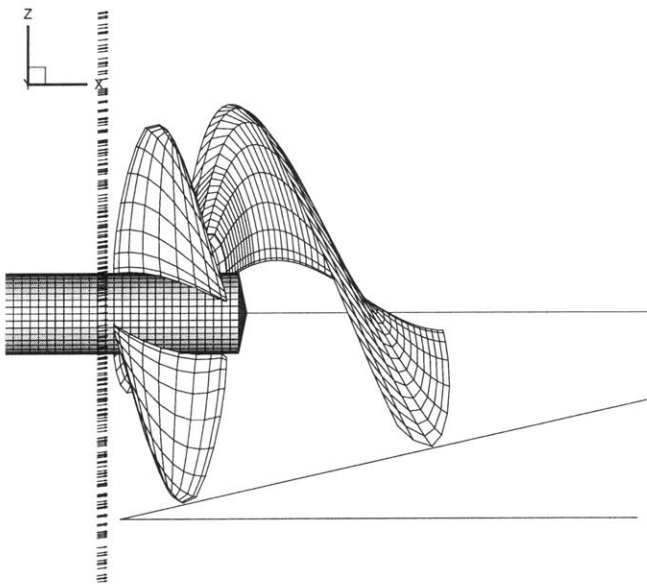


Figure 4-2: With the propeller operating in a 10 degree inclined inflow, the wake properly follows the inflow angle.

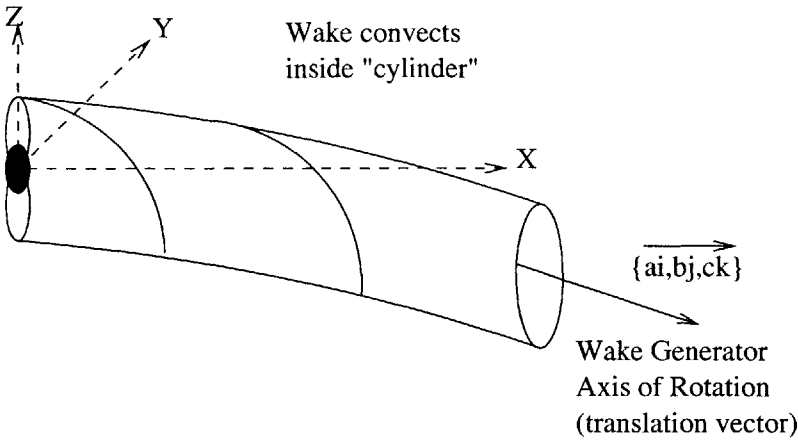


Figure 4-3: As the wake convects downstream aligned with the local flowfield, it forms a cylindrical tube.

4.5 Wake Tracking Methodology

The PUF reference frame is stationary, with the axis centered near the blade, at the centroid of blade rotation. Because the blades remain stationary a rotational component is added to the transition wake as it convects downstream from the blade trailing edge. The approach taken here is to convect the wake sheets with the local flowfield. As the wake sheets convect, a center of rotation is defined at each timestep. The wake sheets can then be rotated about that axis by the angular blade step.

If the wake sheets can be considered to convect downstream within a cylindrical tube that is aligned with the wakefield, then there must be a centerline axis along which the cylinder can be generated. This is shown in Figure 4-3. Conversely, if the centerline axis (generator line) of the cylindrical tube can be defined, then the wake sheets can be properly rotated about the generator line to account for the rotational component required within the PUF reference frame. Notice for a propeller in a uniform, straight ahead flow situation, like in a water tunnel, the wake generator line lies along the shaft axis. In real world cases where the wake shifts to align with an unsteady, non-uniform flow field, the wake generator line is arbitrarily oriented in space. This leads to the requirement to rotate a point in space a given angle about an arbitrarily-oriented axis.

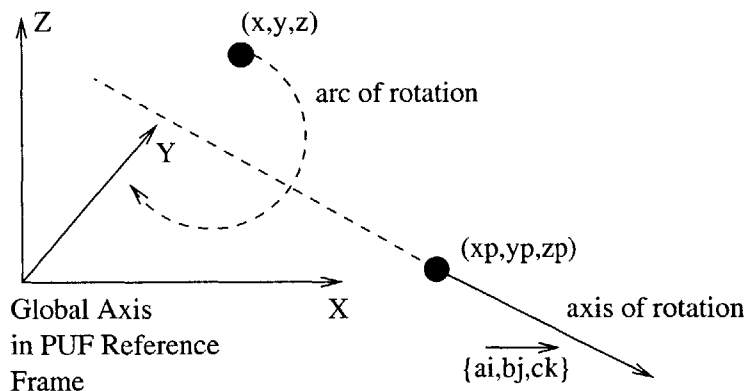


Figure 4-4: Notation used in specifying how a point in the PUF reference frame is rotated about an arbitrary vector

4.5.1 Wake Sheet Generator Line

The wake sheet generator line is created by tracking a cylindrical shell of fluid through space. The centerline of the cylindrical shell is the wake sheet generator line. The cylindrical shell is generated by marching a ring downstream from the seven-tenths radius of the propeller. The center of this ring is the centerline axis. The position of the generator line at each timestep is found by averaging the coordinates of the ring. The orientation of the generator line is found using a backwards Euler differencing scheme.

Once the wake sheet generator line is defined, the wake sheets are convected with the local flow and rotated by the propeller angular increment, which is dependent on the number of propeller time steps per revolution. This involves a three dimensional rotation of a point about an arbitrary axis.

4.5.2 Rotation of a Point about an Arbitrary Axis

The rotation of a point (x, y, z) about an arbitrary vector $\{\widehat{a}_i, \widehat{b}_j, \widehat{c}_k\}$ positioned in space at the point (x_p, y_p, z_p) by a desired angle α is found using a series of rotational and translational matrix manipulations. The basic series of steps, and associated transformation matrices, are shown below:

- Translate the coordinate center of the axis of rotation to the axes center.

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -XP & -YP & -ZP & 1 \end{bmatrix}$$

- Rotate the axis of rotation vector about the Y-axis to place the coordinate center of the vector in the YZ plane

$$R_\theta = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Rotate the axis vector about the X-axis to align the rotation axis vector with the Z-axis.

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) & 0 \\ 0 & -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- The rotation axes vector and the global Z axes are now colinear. The desired rotation is now a rotation of the point about the Z -axis by the desired angle α .

$$R_\alpha = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The remaining steps remove the transformations accomplished in steps 1-3.

- Undo the rotation about the X-axis

$$R_{-\phi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Undo the rotation about the Y-axis

$$R_{-\theta} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Translate the rotation vector base point (which really defines the local coordinate axes) from the global axes center to original position

$$T_- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ XP & YP & ZP & 1 \end{bmatrix}$$

These matrices can be combined to create a single transformation matrix.

$$[Transform] = [T] [R_\theta] [R_\phi] [R_\alpha] [R_{-\phi}] [R_{-\theta}] [T_-] \quad (4.2)$$

4.6 Hub Image Placement

The method of images is used for the hub and duct image placement – as is done for the blade vortex lattice hub and duct. Because there is no general guarantee that a shaft extends downstream of the propeller through the length of the wake, the hub images are more an accounting method to convect the hub image vortices which

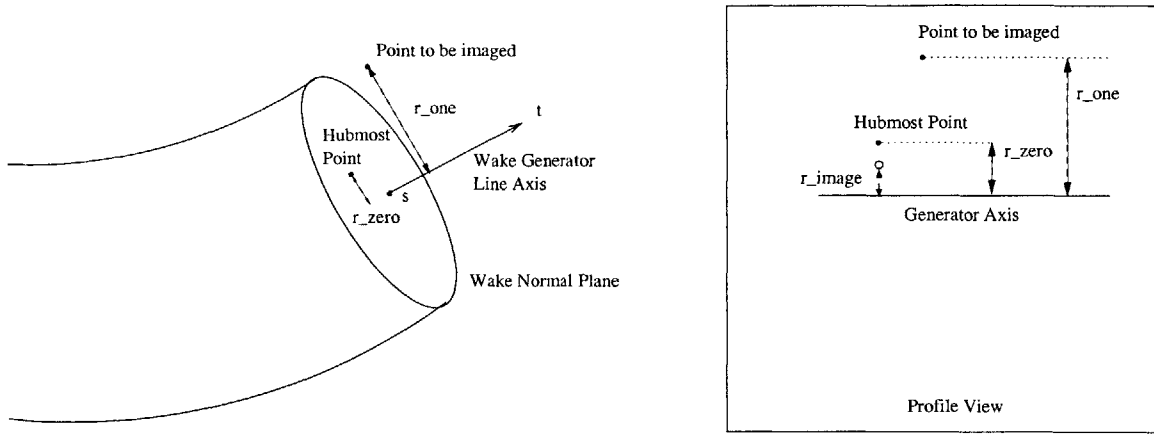


Figure 4-5: Terminology used in determining the position of the wake hub image lattice point positions.

are required for enforcement of the kinematic boundary condition on the hub in the presence of the blade vortices.

Due to the non-axisymmetric nature of the wake vortex loop position, it is not possible to use a common axis center to place the image vortices. Rather, the position of the wake hub images must follow from the position of the wake vortex lattice elements.

As normal, the radius of the hub image point is then given by Equation 4.3.

$$r_{image} = \frac{(r_0 - HGAP)^2}{r_1} \quad (4.3)$$

Here r_0 is the distance from hubmost wake lattice point to the generator axis and r_1 is the distance from the point to be imaged to generator axis. The tricky part is to now define the center axis from which r_0 and r_1 are measured. As luck would have it, though, the center axis was previously defined in convecting the wake in the first place.

The hub image location, as shown in Figure 4-5 is placed in a position coplanar with the innermost wake vortex lattice element to prevent undesirable lattice skew. The axial location of the image vortice is set to match the axial location of the hubmost lattice.

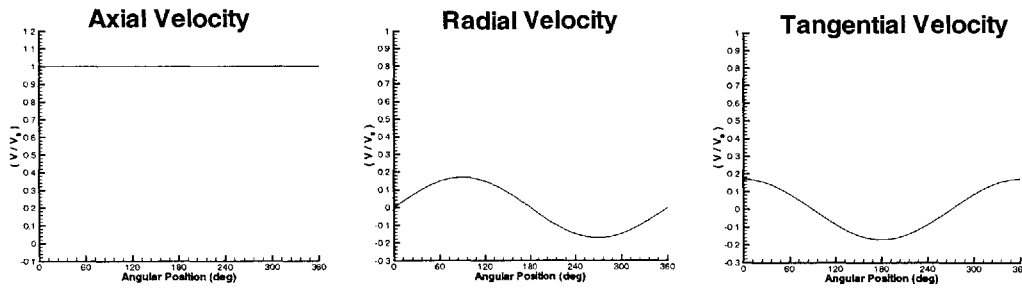


Figure 4-6: Velocity components of a 10 degree inclined flow as seen by a propeller. Plots of the axial, radial, and tangential component of the inflow at the propeller seven-tenths radius.

Force Component	Aligned Wake	Non-aligned Wake	$\Delta\%$
Steady K_T	0.15799	0.15740	-0.37
Steady $10K_Q$	0.29348	0.29443	0.32
1st Harmonic Axial Force	0.01849	0.01458	21
1st Harmonic Vertical Force	0.00158	0.00217	37
1st Harmonic Side Force	0.01055	0.00823	22

Table 4.1: Aligned and Non-aligned wake results for 10 degree angle of inclination.

4.7 Aligned Wakes with Prescribed Effective Inflow

The results presented here represent calculations made with a prescribed effective inflow inclined at 10 degrees. Fully coupled results are presented in Chapter 6. A standard three-bladed propeller, DTMB Propeller 4119, is used for all calculations.

The first result is to show the wake position when aligned with a ten degree angled inflow as shown in Figure 4-2. The axial, radial, and tangential components of the inflow are shown in Figure 4-6. The once per revolution variation in tangential velocity component will produce a wake with non-constant pitch angle. The resulting differences in predicted operating conditions between using the aligned wake model and the non-aligned wake model are shown in Table 4.1. The differences in lateral forces are mainly due to phase differences. Notice that while the mean forces are generally similar, the predicted first harmonic forces differ by an order of magnitude

Circulation at the 0.7 Radius in 10 Degree Inclined Inflow

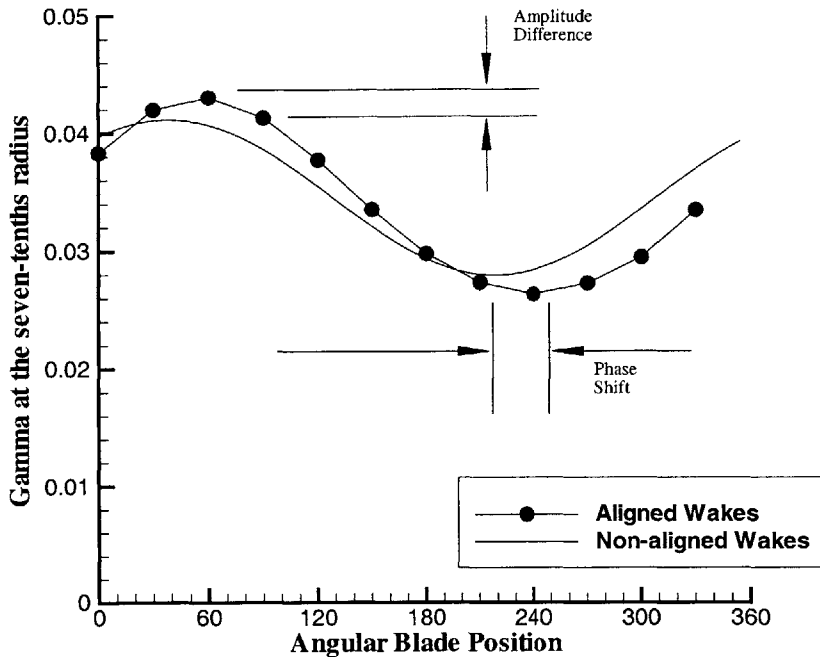


Figure 4-7: With the propeller operating in a 10 degree inclined inflow, this plot shows the circulation at the $\frac{7}{10}$ radius as the propeller rotates through 360 degrees.

Figure 4-7 highlights the difference in resulting blade solution for aligned and non-aligned wakes. Figure 4-7 shows the blade circulation at the $\frac{7}{10}^{th}$ radius as the propeller sweeps through 360 degrees in a 10 degree inclined flow. Note that the mean, or steady circulation, is nearly the same, but the prediction in the 1st harmonic shows phase and amplitude differences.

4.7.1 Once per Revolution Change in Axial Velocity

The second test is to vary the wake pitch angle by creating a stylized flowfield with varying axial velocity. The components of the flowfield are shown in Figure 4-8. This is a simplified model for the effects of upstream appendages. The variation in axial velocity will produce a varying wake pitch angle. Again, the propeller program is run with and without wake alignment.

The results of this analysis are shown in Figure 4-9. Because there is negligible

Force Component	Aligned Wake	Non-aligned Wake	$\Delta\%$
Steady F_X	-0.04874	-0.05068	4.00
Steady F_Y	0.00274	0.00270	1.50
Steady F_Z	0.02822	0.02891	2.50
1st Harmonic F_X	0.01555	0.01311	15.7
1st Harmonic F_Y	0.00214	0.00296	38.3
1st Harmonic F_Z	0.00752	0.00574	23.7

Table 4.2: Unsteady blade forces produced by a propeller in an axially non-uniform flow with and without wake alignment. There are large penalties associated with not aligning the wake.

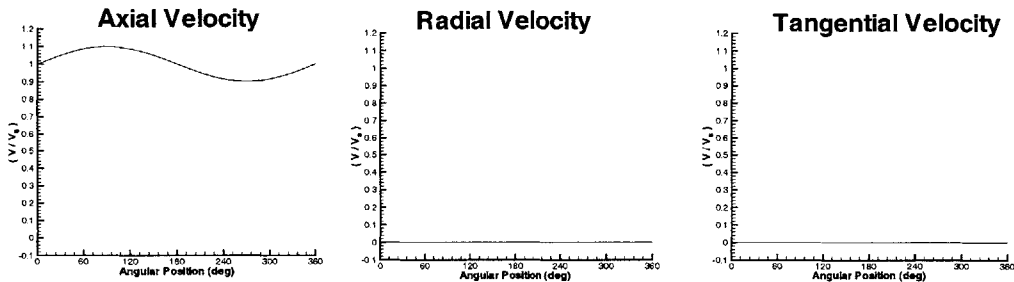


Figure 4-8: Velocity components for a once per revolution variation in the magnitude of the axial component of the inflow. Plots of the axial, radial, and tangential component of the inflow at the propeller seven-tenths radius.

steady offset, the mean predicted propeller K_T and $10K_Q$ differ insignificantly. The once per revolution variation, though, highlights the difference in unsteady performance. Table 4.7.1 shows the differences in 1st harmonic forces.

4.8 Wake Lattice Spacing

Due to the requirement that the vorticity present on a given wake panel convects downstream in the time interval between successive blade clicks, the size of the wake vortex lattice loops are directly determined by this time step (Number of Time Steps per revolution in PUF terminology). Because each shed vortex loop corresponds to the distance between angular blade positions, a large number of blade steps per revolution is required to keep a sufficiently small wake discretization. But an increased

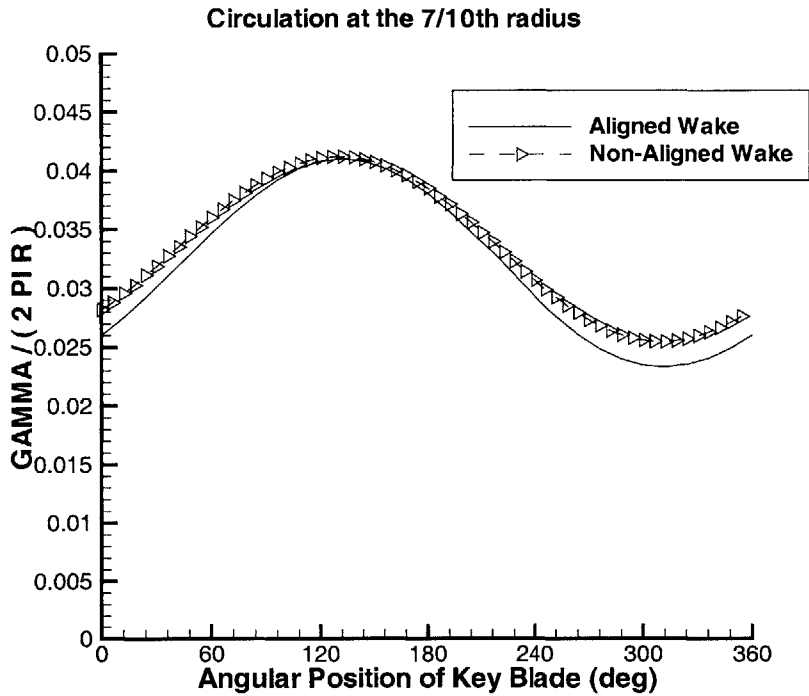


Figure 4-9: 4119 circulation at the seven-tenths radius for axially non-uniform inflow.

number of time steps requires longer solution times. This problem is highlighted by the required angular resolution which follows directly from the user-specified number of time steps per revolution (NTSR).

The propeller time step is given by:

$$\Delta T_{prop} = \frac{2J}{NTSR} \quad (4.4)$$

Which directly leads to the following angular increment between each successive blade

position

$$\begin{aligned}
 J &= \frac{V_S}{nD} \\
 n &= \frac{\Omega_{prop}}{2\pi} \\
 \theta_{prop} &= \Omega_{prop}\Delta T_{prop} \\
 &\text{in PUF, } V_S = 1.0 \text{ and } D = 2.0 \\
 \theta_{prop} &= \frac{\pi\Delta T_{prop}}{J} \\
 \theta_{prop} &= \frac{2\pi}{NTSR} \tag{4.5}
 \end{aligned}$$

Notice that the angular discretization is independent of advance coefficient, J , and depends only upon the user input number of blade positions per revolution. This means that if it is possible to create a wake discretization scheme independent of blade time steps it will also be problem independent. This would obviously be a more robust algorithm in general.

This discretization error is seen in the convergence rate of the blade forces with changes in pseudo time step (the NTSR variable using PUF terminology) shown in Figure 4-10. This test was conducted using a uniform effective inflow and maintaining constant lattice size on the blade. These results are similar to those presented by Warren [30]. Under the older wake growing scheme, where wake loop size is dependent upon time step discretization, the convergence of K_T and K_Q such that they are no longer dependent upon NTSR comes with a stiff penalty in the form of computational expense. Notice from Figure 4-10 that the wake growing scheme is unstable at low NTSR. A goal of this research is to remove the solution dependence on NTSR while maintaining low computation times.

4.9 Higher Order Wakes

There are two methods which could be used to remove the dependence between the propeller discrete time stepping and the wake loop discretization.

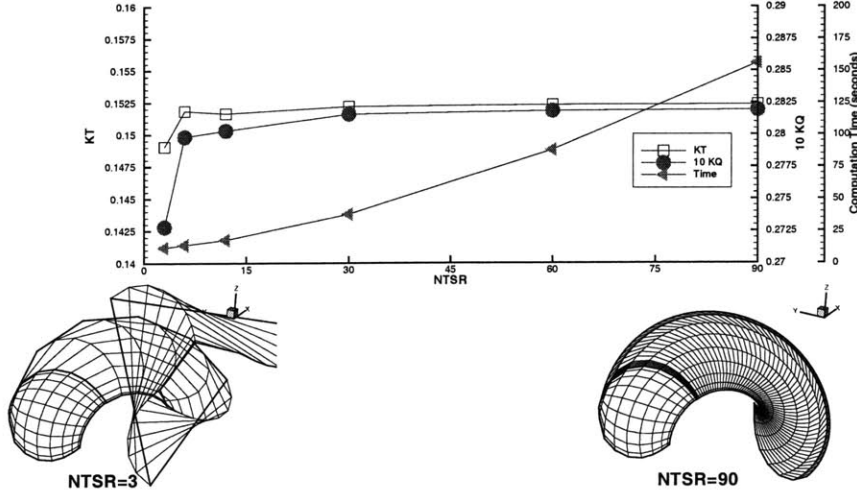


Figure 4-10: The convergence of the propeller solution is directly determined by the pseudo time stepping, which is controlled via the user-defined Number of Time Steps per Revolution (NTSR).

1. Higher order vortex strength representation. The coarse discretization could be overcome with higher order representation of the vortex loop strengths.
2. Higher order panels. If the shed vortex loops can be considered constant strength singularity panels, then a higher order panel would accurately represent the geometry, while minimizing the computational expense. The curvature effects are missed for very coarse panels.

These methods directly result from the fact the influence velocity of the wake is due to the product of the influence coefficient and the vortex loop strength

$$V_i^{inf} = \sum_{k=1}^{N_w} \Gamma_k [WIF]_{i,k} \quad (4.6)$$

The second method is the better choice, since a higher order panel will also solve the instability in the vortex loop placement scheme associated with the current lower order scheme.

4.9.1 Higher Order Wake Loop Scheme

A classical higher order loop would be created by defining the perimeter of the loop, and using a higher-order influence function to return the influence of that loop. The degree would depend, then, on the higher order derivatives at the wake loop edges. This is nearly equivalent to breaking the loop into smaller loops, calculating the influence of each smaller loop, and re-assembling these smaller influences into an overall influence coefficient for the total loop. Recall that global loop size is still dictated by the propeller discrete time stepping (since shed vorticity from the blades at each pseudo time step must be convected in the wake). The advantages of the loop subdivision scheme are,

- wake curvature effects are captured
- more stable than large stepping scheme currently used
- ability to easily integrate a streamwise strength variation scheme

Figure 4-11 shows pictorially the resulting wake loop discretization with and without subdivisions. The smaller subdivided panel better capture the real curvature effects of the wake, and reduce the mis-alignment between the wake panel tangency and the local fluid flow.

4.9.2 Convergence with Subdivided Wake Loops

Figure 4-12 shows the results for varying the subdivisions of each wake vortex loop. Notice that KT and 10KQ converge after the wake loops are subdivided to an equivalent $NTSR=60$ level, and requires one-third of the computation time. Based on these results, the algorithm is set to subdivide the wake vortex loops to use an equivalent pseudo timestepping which would result if the user had specified $NTSR = 60$ in the old scheme. The pseudo time stepping in the wake, then, is not equivalent to Equation 4.4 but is given by Equation 4.7.

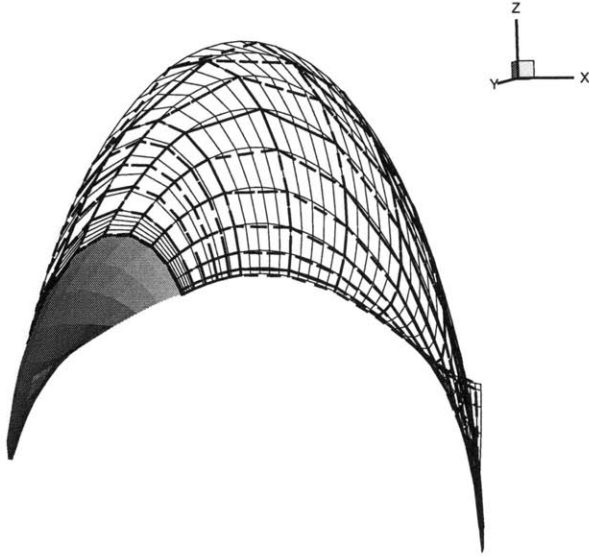


Figure 4-11: Subdividing the loop sizes required by the pseudo time step into smaller loops creates a higher order influence panel. The solid dashed lines are the actual wake loop sizes as dictated by the pseudo time step and the lighter solid lines are the subdivided wake loop locations. There are five subdivisions per time step.

$$\Delta T_{wake} = \frac{2J}{60} \quad (4.7)$$

And the equivalent rotational component added at each time step is shown in Equation 4.8.

$$\theta_{prop} = \frac{\pi}{30} \quad (4.8)$$

The results show a 70 percent reduction in computation time for equivalent accuracy and convergence. Computation time could be further reduced by implementing a higher order influence calculation since the computation burden is now evenly divided between the wake growing algorithm using small pseudo time steps and the calculation of the influences for each of these smaller loops. The time savings results from many factors:

- reduced number of angular positions requires fewer time steps to convect the starting vortex downstream
- the force calculation directory depends on the number of blade panel positions
- less memory required

The use of the user-defined parameter for the number of timesteps per revolution should now only be set higher than the number of blades to capture unsteady, higher harmonic effects. There are many calculations which do not require the calculation of higher harmonic unsteady forces, which will benefit from the 70 percent speed-up.

- Maneuvering vehicle calculations typically only require 1st and 2nd harmonic unsteady forces.
- When coupled with an external flow solver, such as a RANS code, it does make any sense at all to try and calculate harmonic forces higher than the azimuthal grid resolution. For instance, given a seven bladed propeller coupled with a RANS code with 36 grid cells in the azimuthal direction, the user should not expect to resolve any harmonics higher than 2.

4.9.3 Solution Validation with Subdivided Wake Loops

The numerical accuracy of the new wake subdivision scheme is found by comparing the results of PUF with and without the wake subdivision scheme, where there are a large number of wake loops in the non-subdivided scheme. This check is necessary due to the large scale changes in array indexing, solution advancement, and influence looping the wake subdivision scheme required.

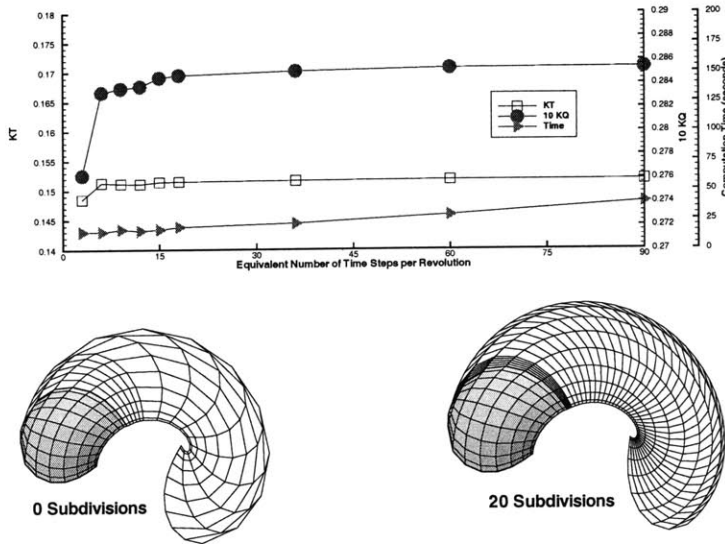


Figure 4-12: Using a pseudo higher order influence scheme through subdividing the wake loops results in less computation effort for convergence.

The numerical results for circulation distribution with the wake subdivision scheme is shown in Figure 4-13. The integrated results for K_T and $10K_Q$ are,

	$NTSR$	K_T	$10K_Q$
PUF14 ¹ with subdivided wakes	3	0.15173	0.28532
PUF14 ¹ without subdivided wakes	60	0.15174	0.28534
PUF14 ² without subdivided wakes	60	0.15231	0.28184

¹ Individual Wakes

² Imaged Wakes

It is hypothesized that this difference is due to round off errors present when rotating the imaged wakes to the correct angular positions.

4.10 Validations of Aligned, Subdivided Wake Scheme

To test the subdivision scheme, Propeller 4119 is given an unsteady effective inflow. While the blade lattice size is held fixed at 15X15, the number of discrete time steps is increased to the maximum memory of the machine. The results are presented in Table 4.3. Notice that the results do vary slightly. This is due to the different

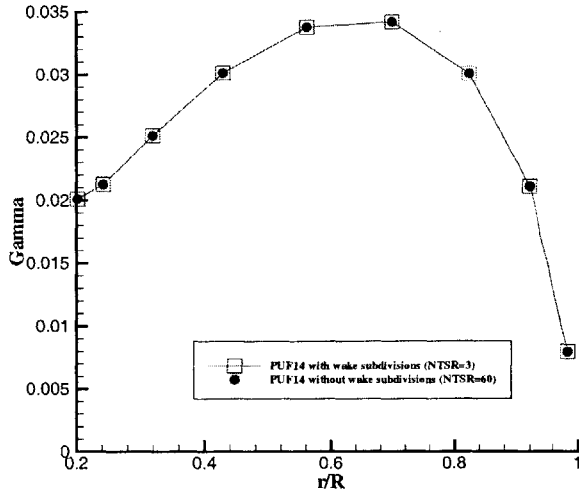


Figure 4-13: Validation of PUF14 using the wake subdivision scheme with PUF14 without the wake subdivision scheme in a uniform, axial inflow.

NTSR	K_T	$10K_Q$
15	0.15380	0.28503
30	0.15423	0.28643
45	0.15453	0.28731

Table 4.3: Affect of NTSR on subdivided wake scheme results. The difference of less than 1% is attributable to the variable spacing in the first wake panel, which does depend upon NTSR

subdivision sizes present in the first wake lattice.

4.10.1 Convergence Test of Aligned, Subdivided Wake Scheme

The first convergence test is to investigate if the propeller solution with the subdivided wakes requires a finer blade grid than the constant-sized wake panel solution. To make the investigation more interesting, the propeller code with subdivided wakes is tested against the propeller code with constant-sized wakes in a non-uniform inflow which is specified as the effective flowfield. The convergence rate with blade lattice size is shown in Figure 4-14. These results indicate that the users of the new code do not need to change their thought process on the correct settings for blade discretization.

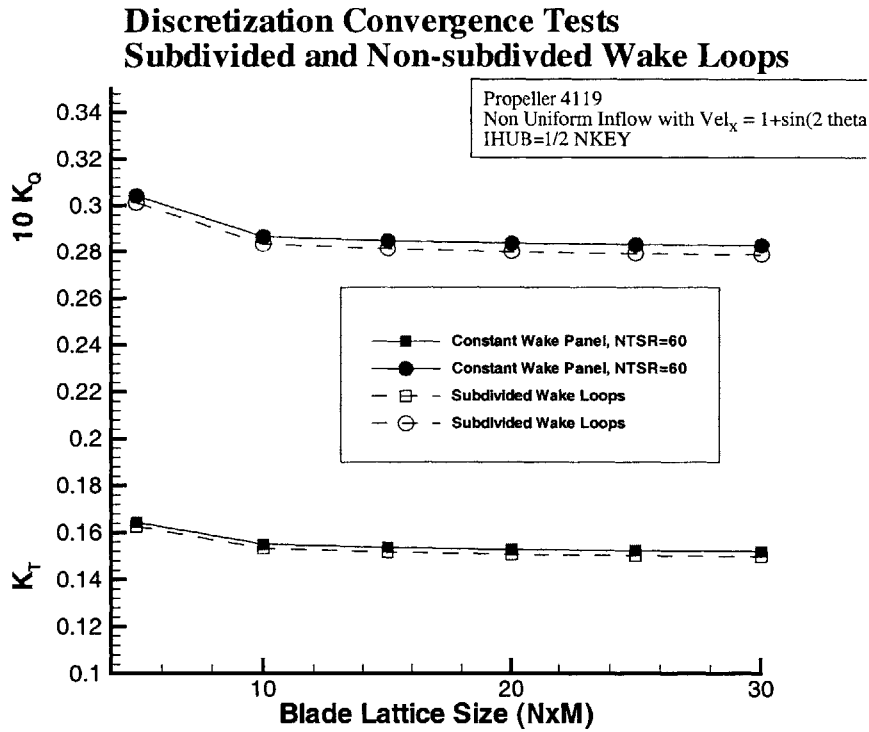


Figure 4-14: Convergence rate of the new wake discretization scheme in a non-uniform inflow. Compared to the previous scheme of large, non-subdivided wake panels, the convergence rate is similar with changes in blade lattice density.

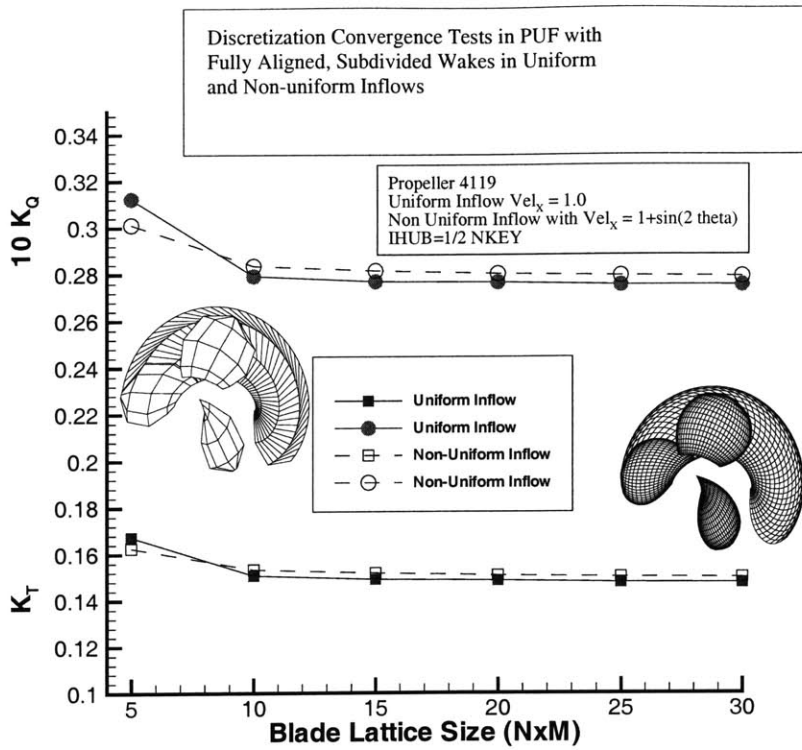


Figure 4-15: Mean K_T and $10K_Q$ results are shown for PUF using fully aligned, subdivided wakes in both uniform and non-uniform inflows. Grid convergence occurs at a lattice size of around 20.

Mean K_T and $10K_Q$ results are shown in Figure 4-15 for PUF using fully aligned, subdivided wakes in both uniform and non-uniform inflows. These results indicate that grid convergence occurs at a lattice size of around 20.

4.10.2 Propeller 4119 in Non Uniform Axial Inflow

The varying pitch angle of the wake as shown in Figure 4-1 is compared with the wake grown in previous methods in Figure 4-16. While there is some difference in the performance prediction between the aligned and non-aligned calculations, the largest difference shows up in the prediction of the blade harmonic forces.

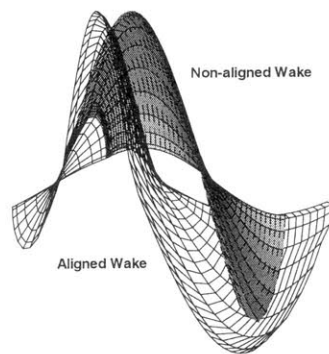


Figure 4-16: A non-uniform inflow directly affects the pitch angle of the wake sheets. Here the non-uniformity is produced by a step function change in the axial velocity. Aligning the wakes with the local flow produces radical changes in the wake pitch angle. The aligned wake is the same as shown in Figure 4-1.

Chapter 5

Coupled Algorithms

The major shortcoming of the lifting surface formulation described above is the inability to model the vorticity interaction. As previously mentioned, the propeller induction velocities cause a redistribution of the vorticity present in the inflow. This interaction is critical in the design of modern, full stern submarines, where the stern prismatic coefficient is so high that the propulsor actually inhibits separation of the boundary layer.

The second area of propulsor design affected by this shortcoming is in the design of multi-element propulsors. The problem is that at points where the trailing wake vorticity from the upstream blade rows intersect the control points of the downstream blades, the mathematics of the problem produces a singular solution. Multi-element propulsor design is important in the design of modern naval combatant propulsors, and in the design of waterjet pumps.

5.1 Time-averaged Induced Velocity

When coupling a potential-based lifting-surface blade solver with a RANS code, the need arises to correctly evaluate the effective velocity given the total velocity solved for by the RANS code.

In the propeller code, the velocity at any field point can be broken down as follow:

$$V^{\otimes} = V_{eff}^{\otimes} + V_{ind}^{\otimes} + \tilde{V}^{\otimes} \quad (5.1)$$

where: V_{eff}^{\otimes} is the effective inflow

V_{ind}^{\otimes} is the time-averaged induced velocity

\tilde{V}^{\otimes} is the fluctuating, local blade-to-blade induced velocity

Since the effective inflow is not measurable in the RANS code, the assumption is made that the mean of the local blade-to-blade flow is zero,

$$\overline{\tilde{V}^{\otimes}} = 0 \quad (5.2)$$

This then directly leads to the final statement that the time-averaged effective velocity is equivalent in the RANS code, as indicated by the V^{\odot} notation and the propeller code, such that the effective velocity is just the total velocity minus the time-averaged induced velocity.

$$V_{eff}^{\otimes} = V_{eff}^{\odot} = V^{\odot} - V_{ind}^{\otimes} \quad (5.3)$$

This statement ignores the effect of the fluctuating blade-to-blade component on the vorticity redistribution, explained in Section B.1, but is countered by the physical observation that the higher harmonically fluctuating blade to blade flows quickly decay away from the blades.

The computational burden then comes from the requirement to compute the time-averaged induced velocity at all boundary condition points. In its most elemental form, this calculation involves determining the velocity induced at a field point in space by a large set of three dimensional vortex segments rotated about an axis to form conic vortex sections, where the strength of the vorticity varies with respect to angular position on the conic.

5.2 Velocity Induced by a Vortex Segment

The velocity induced by a vortex segment on a point in space is known precisely from the Biot-Savart Law [14],

$$d\vec{v} = \frac{\Gamma}{4\pi} \frac{\vec{R} \times d\vec{s}}{R^3} \quad (5.4)$$

where: $d\vec{v}$ is the induced velocity vector

\vec{R} is the vector from the vortex element to the field point

$d\vec{s}$ is the unit vector along the vortex segment

R^3 is the cube of the distance vector

Integrating Equation 5.4 along the length of the segment gives the total induced velocity upon the field point. The numerical implementation of Equation 5.4 is described more fully by Greeley [7].

5.2.1 Discrete Calculation of the Velocity Induced by a Conic Section

Using the Biot-Savart law as implemented numerically by Greeley [7] naturally leads to representing the conic section as a series of vortex sticks swung about the x -axis. This is shown in Figure 5-1. Because the strength of the vortex segment varies with angular position, it follows that the induced velocity of the conic section can be found through integrating the effect of the discrete vortex segments,

$$\vec{v} = \int_0^{2\pi} \int_0^s \frac{\Gamma(\theta)}{4\pi} \frac{\vec{R} \times d\vec{s}}{R^3} d\theta \quad (5.5)$$

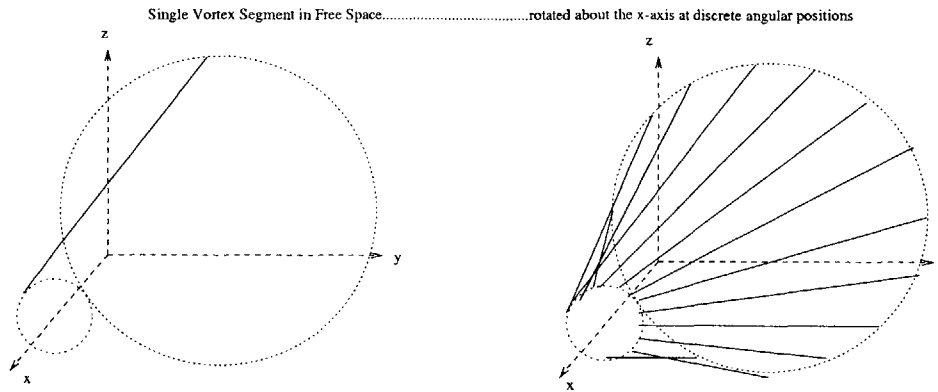


Figure 5-1: Conic section made up of discrete vortex segments. This illustration shows how discrete vortex segments can be combined to form a nearly continuous sheet.

5.3 Velocity Induced by a Ring Vortex of Harmonically Varying Strength

An alternative method, presented by Buchoux [1] for the uniform axisymmetric case, and refined by Greeley is to represent the conic section as rings of vorticity. The Biot-Savart equation is still the governing equation, but instead of integrating around 0 to 2π , vortex rings are integrated along the surface of the conic section. The derivation for the induced velocity due to a vortex ring of uniform strength is shown in Kucheman [18]. The final resulting equation for the velocity induced by a vortex ring of harmonically varying strength was shown first by Greeley [6] who implemented several computer codes to solve the model problem.

5.3.1 Velocity Induced by Constant Strength Vortex Ring

As shown in Kucheman [18], a constant strength vortex ring induces only an axial and radial component of velocity. The x component of velocity is given by integrating the ring around the circumference,

$$V_x(x, r) = -\frac{\Gamma}{4\pi r'} \int_0^{2\pi} \frac{r \cos(\phi - \phi') - 1}{\sqrt{x^2 + r^2 + 1 - 2r \cos(\phi - \phi')}} d\phi' \quad (5.6)$$

This integral is expressed in terms of elliptic integrals through,

$$V_x(x, r) = -\frac{\Gamma}{4\pi r'} \frac{1}{\sqrt{x^2 + (r+1)^2} [x^2 + (r-1)^2]} I_1$$

where

$$I_1 = \int_0^{2\pi} \frac{r \cos(\phi - \phi') - 1}{\sqrt{1 - \frac{2r[\cos(\phi - \phi') + 1]}{x^2 + (r+1)^2}} \left[1 + \frac{2r[1 - \cos(\phi - \phi')]}{x^2 + (r-1)^2} \right]} d\phi'$$

I_1 can be further simplified by letting,

$$k^2 = \frac{4r}{x^2 + (r+1)^2}$$

such that,

This gives the final result that,

$$v_x = \frac{\Gamma}{2\pi r} \frac{1}{\sqrt{x^2 + (r+1)^2}} \mathbf{K}(k) - \left[1 + \frac{2(r-1)}{x^2 + (r-1)^2} \right] \mathbf{E}(k) \quad (5.7)$$

A similar derivation can be followed to arrive at the radial velocity induced by a vortex ring. Due to the axial symmetry of the problem there can be no tangential component of the induced velocity

5.3.2 Non-constant Strength Vortex Ring Induced Velocity

Non-constant strength vortex rings can be modeled through replacing Γ in Equation 5.6 with $\Gamma_n \sin(n\phi)$. Modeling the various harmonic component of Γ , Equation 5.6 is modified to,

$$V_x(x, r) = -\frac{1}{4\pi r'} \int_0^{2\pi} \frac{\Gamma_n \sin(n\phi') (r \cos(\phi - \phi') - 1)}{\sqrt{x^2 + r^2 + 1 - 2r \cos(\phi - \phi')}^3} d\phi' \quad (5.8)$$

Equation 5.8 can again be reduced to a complete elliptical integral of the third kind. All three components of velocity are present, now, due to the loss of axial symmetry.

5.3.3 Conic Sections by Successively Integrating Vortex Rings

The conic section problem is then solved by successively integrating the vortex rings along the length of the conic, and properly accounting for the phase shift and amplitudes of the induced velocities. Due to the linear nature of the problem, the solution is sped up by finding the complex response phase and amplitude response at a point defined by its axial and radial coordinates. The response at other points along the same radial ring are phase shifted from this initial solution.

Adaptive Quadrature Method

An adaptive quadrature scheme is required due to the highly non-linear nature of the influence velocity calculation. The Romberg integration scheme [26] with the Richardson extrapolation method [26] was implemented. The basic methodology is outlined in Appendix C.1.

At each subdivision step, a trapezoidal integration is performed followed by the Richardson error extrapolation. Based on the convergence check, a further subdivision is performed unless the maximum subdivision level has been reached.

5.3.4 Discrete and Analytic Model Comparison

The necessity of the analytic model can be shown through comparing the discrete and analytic induced velocities calculated for simple model lamp shade problems. The lamp shade is formed by rotating a three dimensional element of vorticity about the x -axis as seen in Figure 5-1. To simulate the harmonic effects found in unsteady propeller forces, harmonically varying strengths are assigned to the elemental vortex as it traverses 360 degrees, where the strength of the segment is dependent upon the angular position of the element.

The question as to how fine a discretization level is required to produce results comparable to the analytic results must be answered. This is done through two model problems where a field point is moved relative to a conic section represented by sticks of vortex segments.

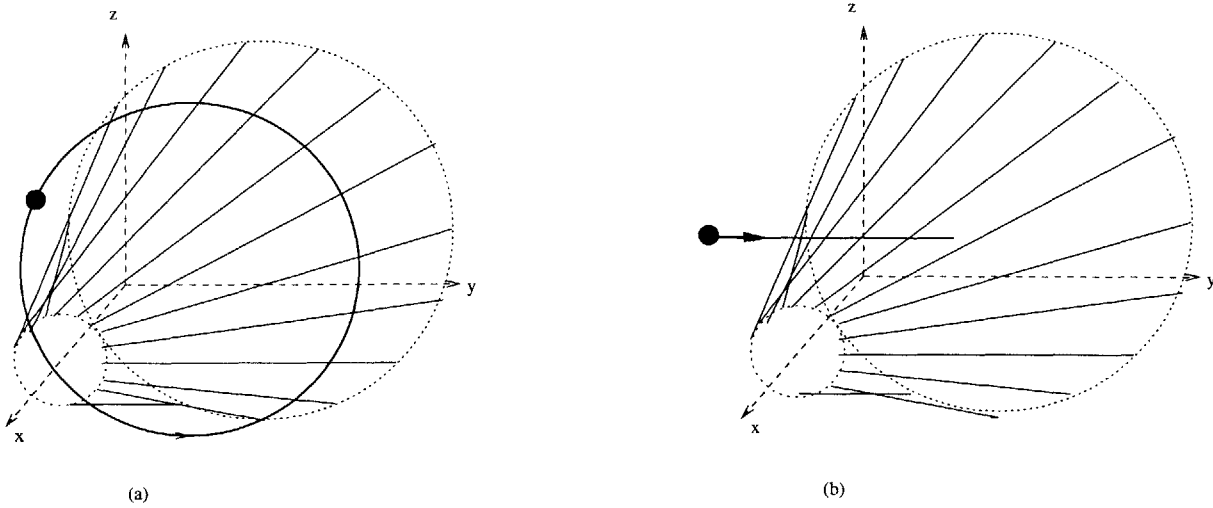


Figure 5-2: Two model problems to highlight the discretization required to reproduce analytic results.

The first model problem involves forming a conic section of straight vortex segments and rotating a field point around the outside of the conic section. Increasing the number of discrete segments makes the resulting induced velocity calculations converge to the analytic results as shown in Figure 5-3. The method of using discretized lateral elements does not converge until a high number of elements are used, while the method of successively integrating analytic rings converges quite quickly. Note the instability in the predicted induced velocities for a medium resolution level of discretization.

The second validation is to again calculate the influence of a conical section of vorticity upon a field point as the field point approaches, and passes through, the lamp shade. Again, the conical section has non-uniform, harmonically varying strength. The results shown in Figure 5-4 show that a high number of discretizations are needed for the discrete solution to approach the analytic solution. The method of using discretized lateral elements does not converge until a high number of elements are used, while the method of successively integrating analytic rings converges quite quickly.

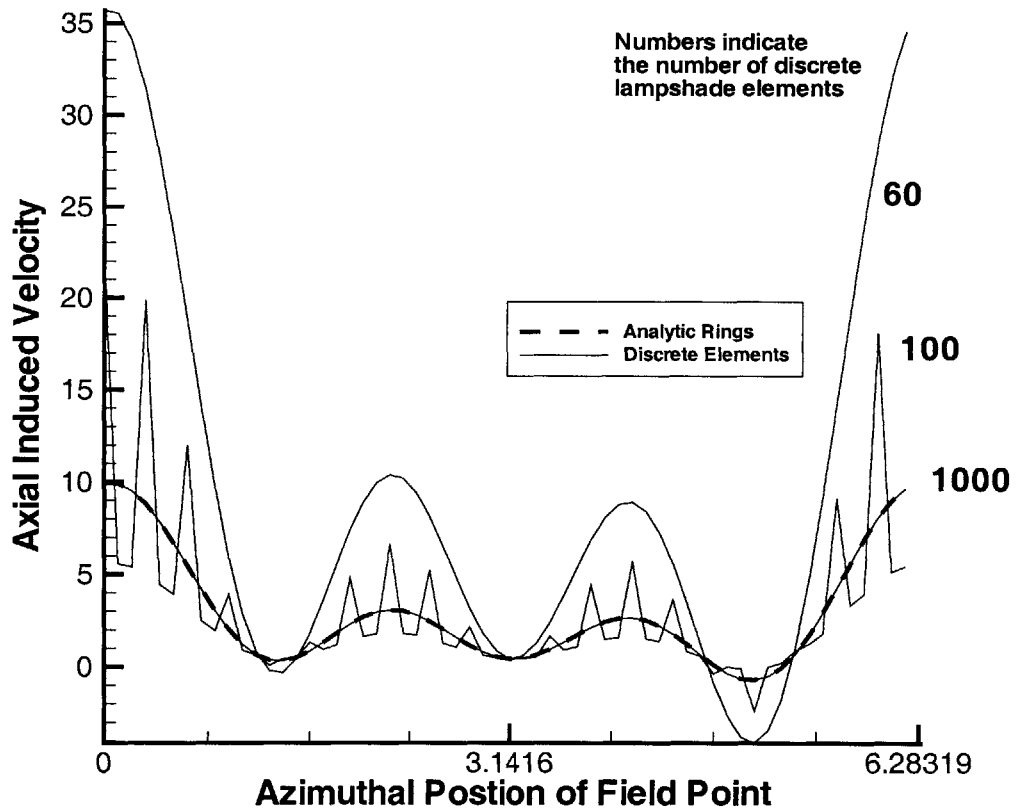


Figure 5-3: Induced velocity upon a field point as it traverses around a conical vortex sheet as predicted by both the analytic method and discretized stick elements.

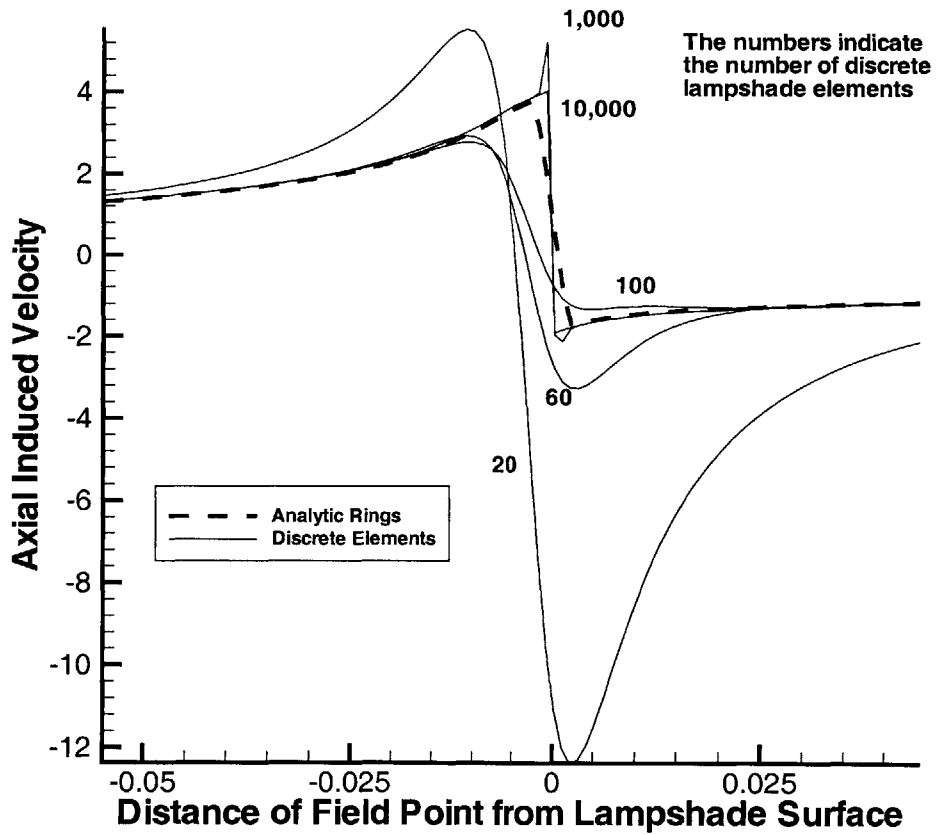


Figure 5-4: Induced velocity upon a field point as it pierces a conical vortex sheet as predicted by both the analytic method and discretized stick elements.

5.4 Induced Velocity Methodology Implementation

The analytic expression for the velocity induced by a non-uniform vortex ring was implemented as part of a new fully unsteady three dimensional vortex-lattice lifting-surface code, PUF-15. The basic procedure follows as,

1. At the end of the blade solution, calculate blade circulation harmonic amplitudes and phases.
2. For each blade loop, calculate the analytic induced velocity based on an adaptive Romberg iteration scheme which calls the harmonic vortex ring influence routines.
3. The wakes are handled as axisymmetric annuluses, similar to the blades, with the proper shed vortex strength.
4. The induced velocity is then the recombination of the influence and the circulation amplitude, offset by the phase difference between the response and the input.

5.4.1 Numerical Validations

Axisymmetric ring vortices of constant strength were shown by Buechoux [1] to have an analytic solution, and an induced velocity routine based on those derivations was implemented by Bechoux in the steady propeller analysis code, PBD-14. By analyzing the unsteady propeller in a steady flowfield the results from the two analysis techniques are demonstrable.

Figure 5-5 shows the comparison of the analytic routines implemented by Bechoux in the steady propeller code PBD and the analytic routines implemented in the unsteady propeller program PUF15 by this research. Both codes use an analytical representation of the vorticity as vortex rings and solve the problem via elliptic integrals. The good agreement and solution smoothness over the entire blade validates the new approach. Comparison between the two analytic codes shows very

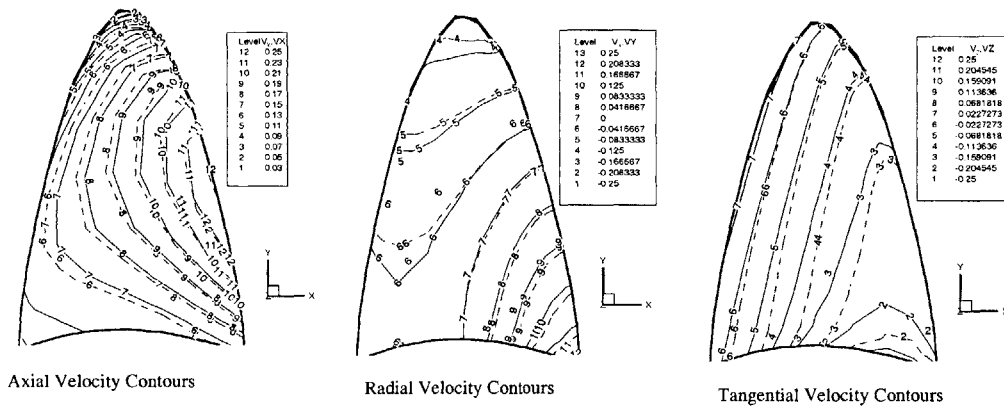


Figure 5-5: Comparison of induced velocity calculated in steady propeller analysis program, solid contour lines, and the unsteady propeller analysis program PUF in dashed lines.

good agreement. Within an unsteady code, a comparison can be made between the induced velocities produced by the discrete algorithm and the new analytic algorithm. Comparison between the discrete modeling algorithm implemented in PUF14 and the analytic algorithm implemented in PUF15 is shown in Figure 5-6. The largest discrepancies are found near the tip, where the blade vortex lattice segments are densely spaced. This discrepancy, however, is balanced by the fact that there is very little load at the tip, which minimizes the error introduced into the coupling.

5.5 Grid Intersection Methodology

The second major piece in coupling the vortex-lattice lifting-surface code and a RANS code is to interpolate the velocities from the RANS grid to the propeller grid, and to interpolate the propeller-generated forces within the swept volume of the propeller to the proper RANS cells.

The interpolation of continuous propeller forces to the RANS grid is the more complex of the two interpolations since volume interpolations, vice point interpolations, must be done. Previous coupling methods have used an exhaustive polyhedron overlap search algorithm which finds the percent overlap between each PUF cell and each RANS grid cell.

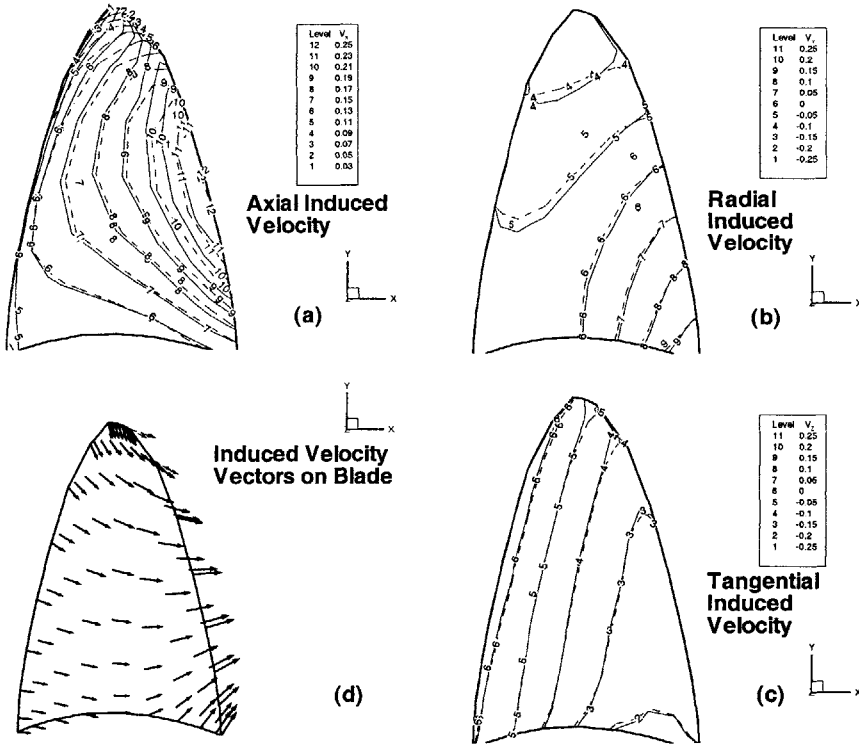


Figure 5-6: Comparison of induced velocity calculated in unsteady propeller analysis code via discrete (solid lines) and analytic (dashed lines) methods. The comparison of induced velocity vectors in (d) shows the large discrepancy at the tip.

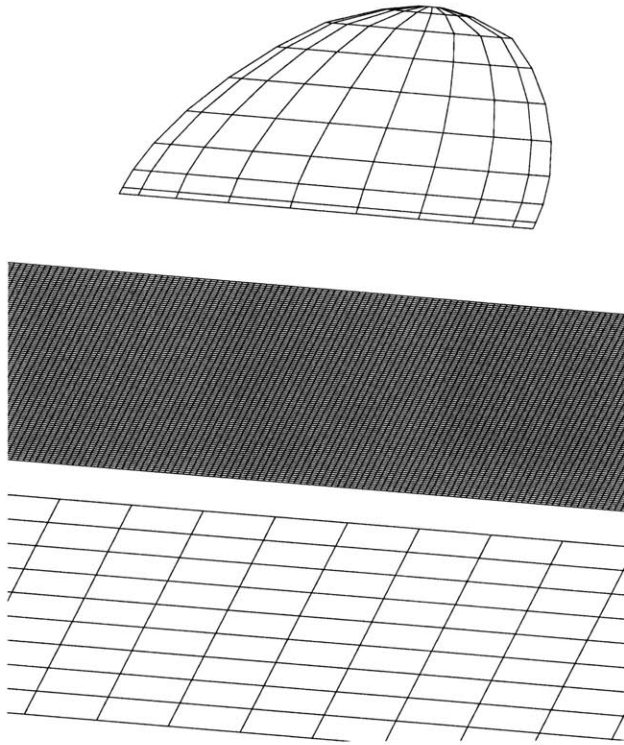


Figure 5-7: Grid subdivision to transfer forces between PUF and RANS. The upper figure is the circumferential PUF blade grid where body forces are known. The bottom grid is the actual RANS grid, and the middle grid is the subdivided RANS grid, which is the fine transfer mesh.

This thesis has implemented a new algorithm which uses a cell subdivision technique. The process is to subdivide the RANS cells into smaller cells. It is then a very fast search to determine if the center of the subdivided RANS cell lies within any of the PUF cells. If the center of the subdivided RANS cell is found to lie within the swept volume of the propeller, a linear interpolation is done from the edges of the PUF cell to place the correct force density in the RANS cell. Recall that RANS solvers introduce the propeller forces as source terms on the right hand side of the RANS equation [29]. This formulation also leads to the propeller forces being called body forces. A pictorial of this process is shown in Figure 5-7.

Numerical experiments show that subdividing the RANS cells by a factor of 10 along each side, such that there are 1,000 subdivided cells within each RANS cell, keeps the interpolation error well under 0.10%. And because the searching algorithm now involves the search for a point within a limited set of domains, vice polyhedron

overlapping volumes, the algorithm is actually faster.

Chapter 6

Coupled Validations

Coupling the propeller and RANS codes allows for numerical validation, as well as validation against experimental test results.

Section 6.1 presents results for coupling the propeller code PUF-15 with two separate RANS codes, IFLOW-41 and UNCLE-3D. Two RANS codes are used to assess the relative errors due to the discretizations and formulations of the propeller code, the coupling code, and the RANS code. Section 6.2 presents the results for the same propeller 4119 inclined at 10 degrees with respect to the horizontal. Section 6.3 presents coupled results for the DTMB Propeller 4679 driven by a downstream shaft inclined at 7.5 degrees. Experimental data recorded for propeller 4679 is compared with the predictions from this thesis.

Figure 6-1 gives an overview of the code interaction, and workflow followed to show validated results. Among the four coupled cases presented here, all validations were performed to the greatest extent possible. The blocks in the center of Figure 6-1 represent the codes used for the analysis, and the dashed line encompasses newly created analysis codes

6.1 Propeller 4119 Validations

The DTMB propeller 4119 shown in Figure 6-2 represents nearly an ideal propeller to validate against due to the large, benign blade sections, with no rake and skew

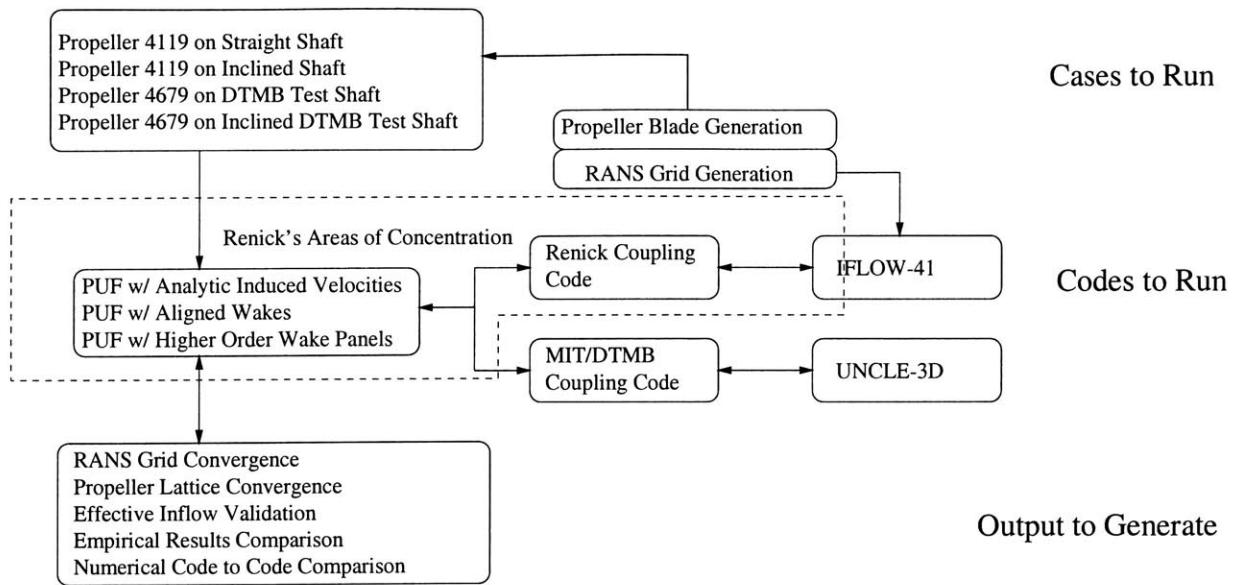


Figure 6-1: This schematic gives an overview of the coupled RANS cases run, and the types of analysis performed..

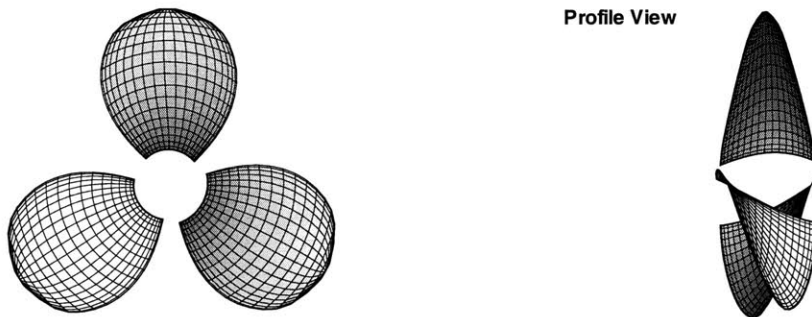


Figure 6-2: These plots show the geometric simplicity of the DTMB propeller 4119.

[4]. Experimental data is available for boundary layer profiles, pressure distributions, drag coefficients, and downstream wake survey [12]. This fact and the geometric simplicity make 4119 an ideal validation candidate for numerical propeller schemes.

6.1.1 UNCLE-3D Coupling with Propeller 4119

The UNCLE-3D RANS solver represents a modern day RANS solver. The computational RANS grid is shown in Figure 6-3. It is an O-grid topology of size 75X35X57. A grid convergence study was not completed since this validation work follows from

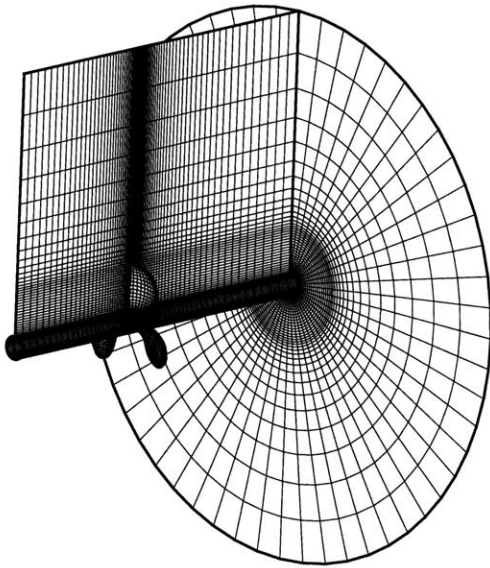


Figure 6-3: Numerical grid used in the UNCLE-3D RANS code

previous validation work [30]. Note the great number of cells across the chord of the propeller. Experience has shown that greater than 10 cells are required across the chord to accurately capture the changes which occur in the blade passage.

Induced Velocity Calculations

Coupling the propeller and RANS code in a uniform inflow, with a no-slip boundary condition on the shaft to simulate an inviscid flow, should give the result that the effective velocity everywhere on the blade is 1.0. Previous results for this test case [30], showed that this result was not attained.

The improved algorithm for the calculation of induced velocities presented in this thesis shows large improvements over previous results. Figure 6-4 shows contours of effective velocity for the new (analytic) and old (discrete) cases. The velocity should be 1.0 everywhere on the blade. The analytic algorithm presented in this thesis shows better results than the discrete scheme previously used. These improvements, while not perfect, carry through into the calculation of the propeller performance as compared to experiment, shown in Table 6.1.

A final way to view the positive trend is to examine the circulation distribution on the blade predicted by using the new and old induced velocity algorithms with

	PUF/UNCLE3D discrete	PUF/UNCLE3D analytic	Experiment
K_T	0.16412	0.15267	0.1455
$10K_Q$	0.30045	0.28191	0.2810

Table 6.1: Results predicted from UNCLE and IFLOW coupling of propeller 4119 in axial, uniform inflow. The discrete column is the previous method used to calculate time-averaged induced velocities, and the analytic column is using the method presented in this thesis.

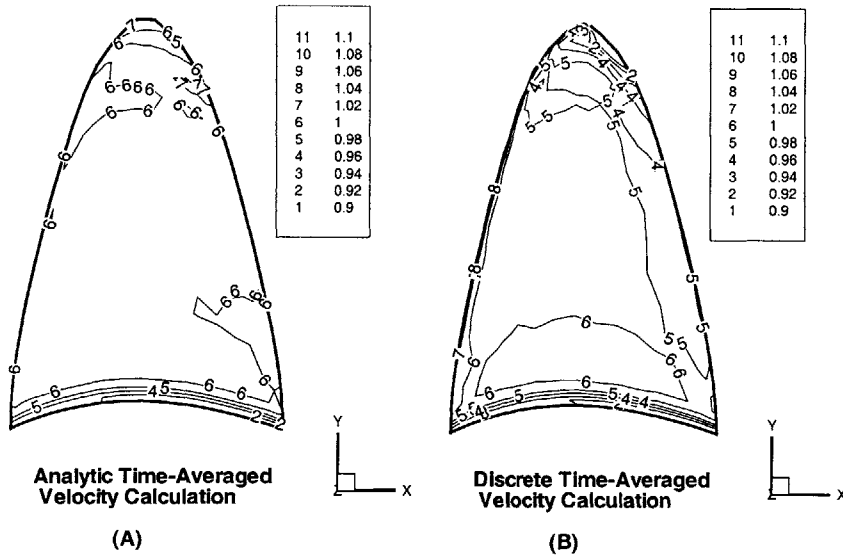


Figure 6-4: These two plots show the axial component of the effective velocity for Propeller 4119 in uniform inflow, coupled with the UNCLE flow solver.

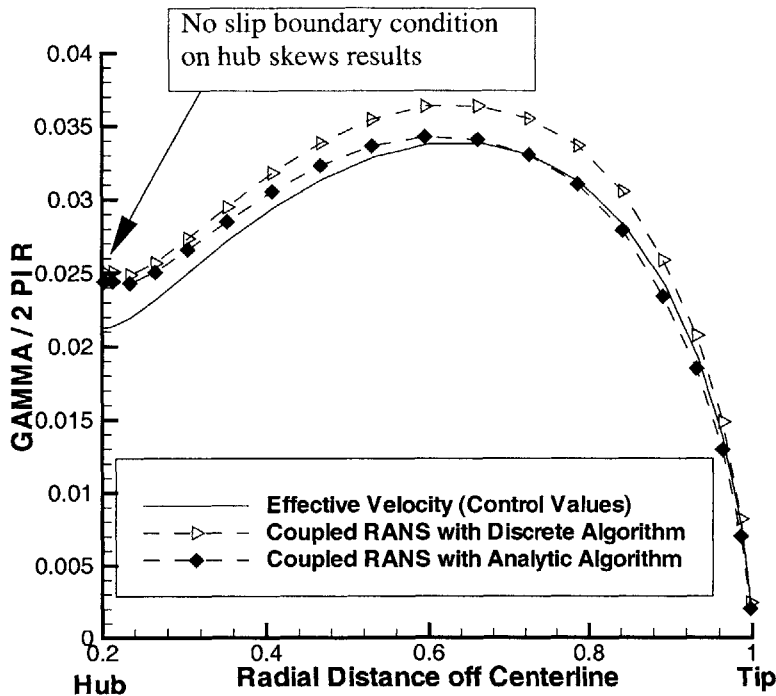


Figure 6-5: Blade circulations are shown for the old and new coupling algorithms, compared to the stand-alone calculation, which is the control in this experiment.

the circulation distribution predicted for a perfect effective inflow of uniform axial velocity. A plot of the predicted blade circulations is shown in Figure 6-5. Notice how much closer the analytic algorithm recovers the true solution. Both coupled algorithms are inaccurate at the hub, because the RANS code has grown a shaft boundary layer. This delta at the hub does highlight the effect of *not* properly accounting for a thin boundary layer, in that neglecting the presence of the shaft boundary layer will underpredict the true circulation at the hub.

6.1.2 IFLOW-41 Coupling with Propeller 4119

The RANS code IFLOW-41 was coupled with PUF to perform several analysis. The grid is a simple O-type grid, with 48 cells axially, 32 cells radially, and 32 cells azimuthally. Because the purpose of this study wasn't to capture boundary layer effects, the maximum Y^+ is allowed to be 19. The grid is shown in Figure 6-6 .

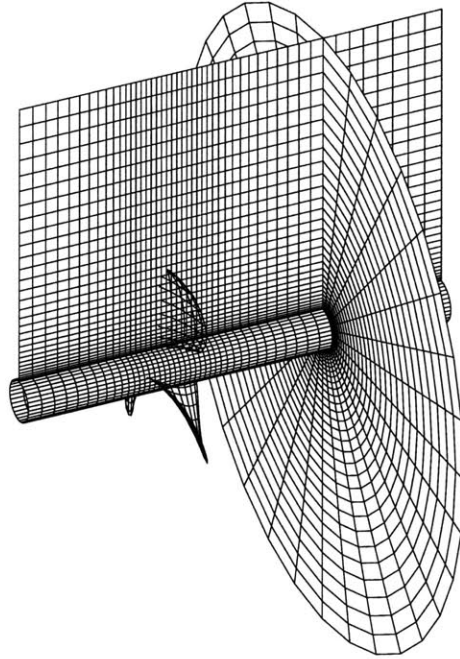


Figure 6-6: Infinite shaft grid used in IFLOW41 of size 48x32x32.

The pressure residual history for two different blade lattice sizes is shown in Figure 6-7. This figure shows that the coupling with IFLOW41 is quickly convergent. A spike in the pressure residual indicates that a new PUF solution for body forces was injected into the RANS domain at that iteration. The fact that the convergence is not dependent on the blade lattice size indicates that the body force coupling routine, which is now internal to PUF, does not introduce spurious results.

Propeller Lattice Convergence Study

A propeller lattice convergence was conducted to find the minimum necessary propeller lattice size required to produce converged answers. The results are presented in Figure 6-8. These indicate that good propeller convergence is achieved with a 15X15 blade lattice size in PUF. Recall that the real key here is that the number of discrete blade positions is 6. This is an order of magnitude reduction over previous techniques.

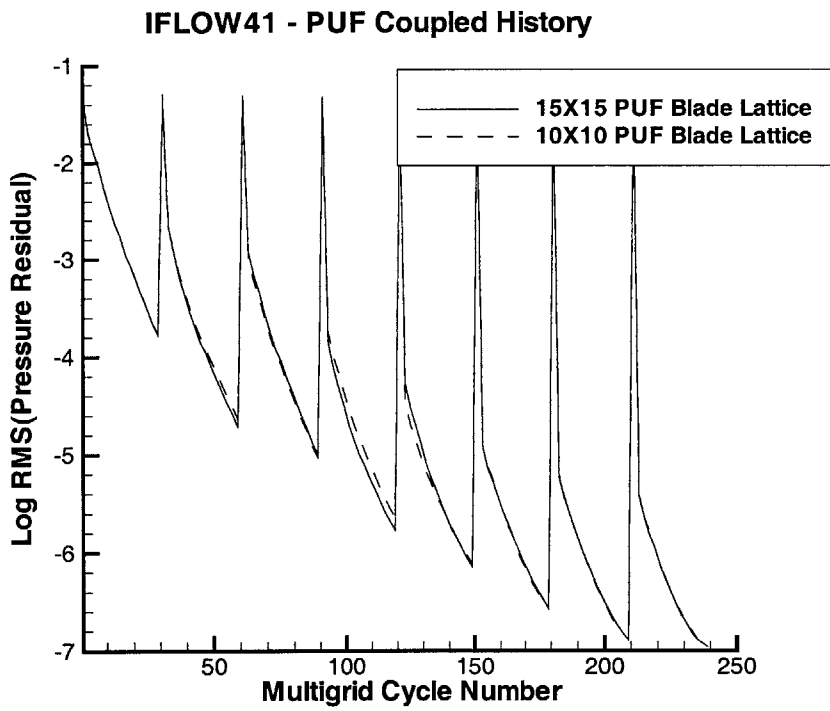


Figure 6-7: Pressure residual history for IFLOW41 coupled with PUF. The solid and dashed line are for two different blade grid densities used on the propeller code.

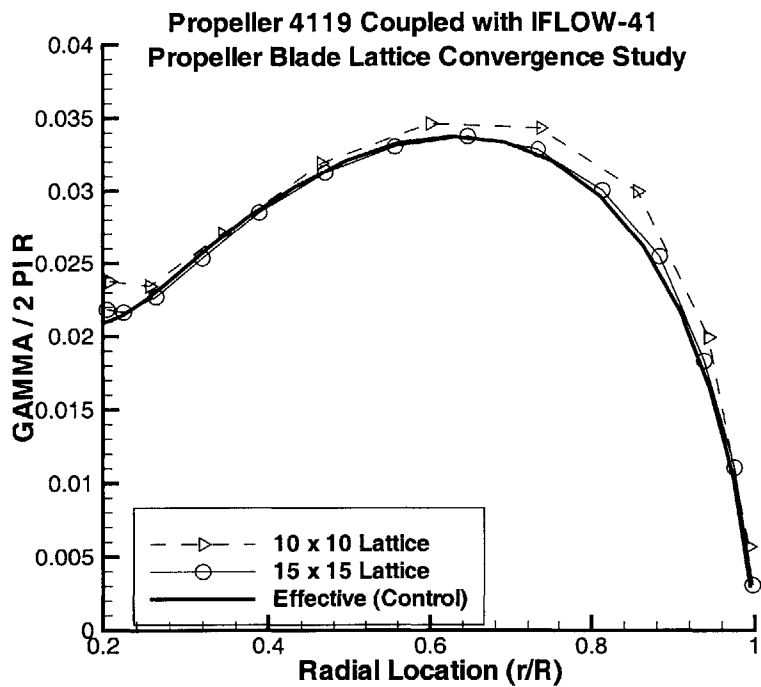


Figure 6-8: Increasing the propeller lattice size when coupled with IFLOW-41 produces results that are closer to the theoretical results. This study shows that use of a 15X15 blade lattice is sufficient to converge a simple propeller geometry, such as Propeller 4119.

6.2 Propeller 4119 at 10 Degree Inclination

A preliminary test of the full wake alignment and higher-order wake panel algorithms is presented here. The stability of the new method is shown in that the solutions converge, and produce results that follow intuition. Since there are no experimental results to compare with, this coupled RANS run shows that the new propeller program PUF-15 works well with the RANS code UNCLE-3D.

Figure 6-9 highlights new physical phenomena which are modeled when a realistic wake model is introduced: the wake detaches from the lee side of the shaft, while effects of radial contraction due to flow acceleration are still captured. Conversely, on the windward side of the shaft, the modeling is not so accurate. Realistically, the wake piles up on itself, while the model here allows the wake to pierce the shaft if the wake stepping algorithm oversteps the innermost streamline.

A pictorial showing the propeller-induced high axial velocities being convected downstream at an angle to the horizontal is shown in Figure 6-10.

The effects of the wake alignment scheme are clearly shown in the next set of figures. Figure 6-11 shows the magnitude of the first harmonic of C_P at the 0.5, 0.7, and 0.9 radii. Figure 6-12 shows the magnitude of the first harmonic of C_P at the 0.5, 0.7, and 0.9 radii. Figure 6-13 shows the phase of the first harmonic of C_P at the 0.5, 0.7, and 0.9 radii.

6.3 Propeller 4679 at 7.5 Degree Inclination

DTMB Propeller 4679 was tested at the DTMB High Speed Basin [10]. The three bladed propeller was driven from downstream with a strut/pod openwater drive system, which was inclined at 7.5 degrees with respect to the horizontal [11]. A computational rendering of the physical model is shown in Figure 6-14.

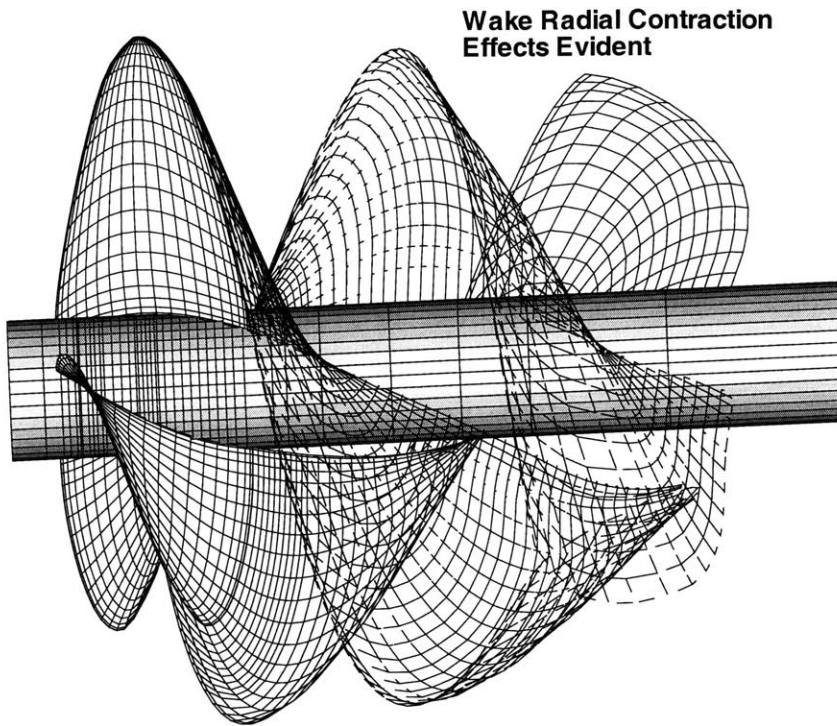


Figure 6-9: Using the flowfield produced by Propeller 4119 on a shaft inclined at 10 degrees, and coupled with the UNCLE-3D RANS code.

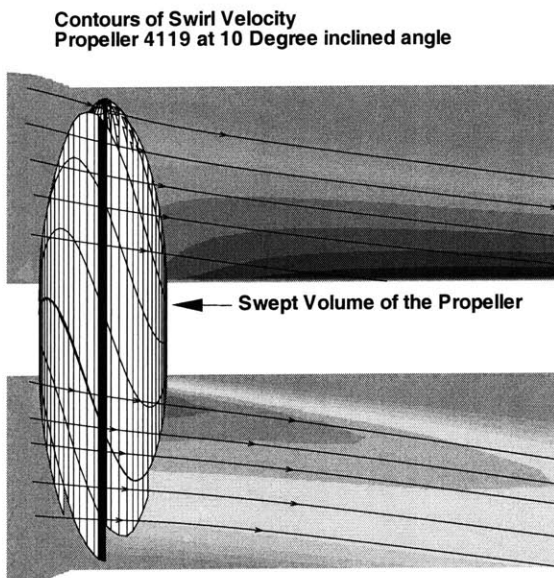


Figure 6-10: Propeller 4119 at 10 Degree inclined angle showing the convection of swirl downstream of the propeller swept volume.

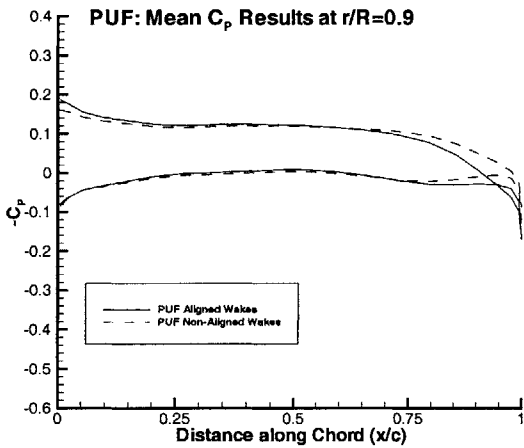
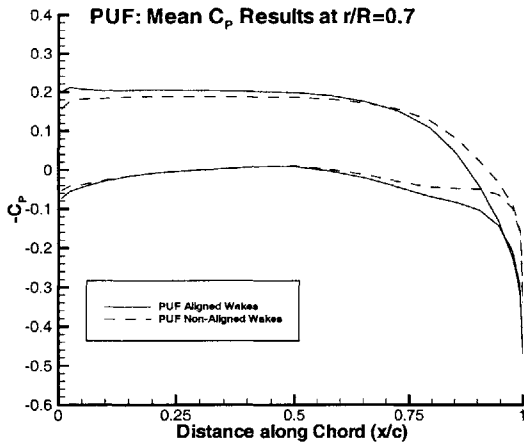
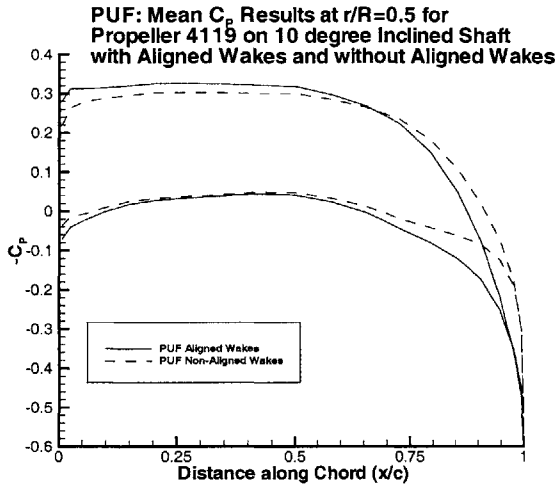


Figure 6-11: Magnitude of the Mean C_p of Propeller 4119 at 10 Degree inclined angle.

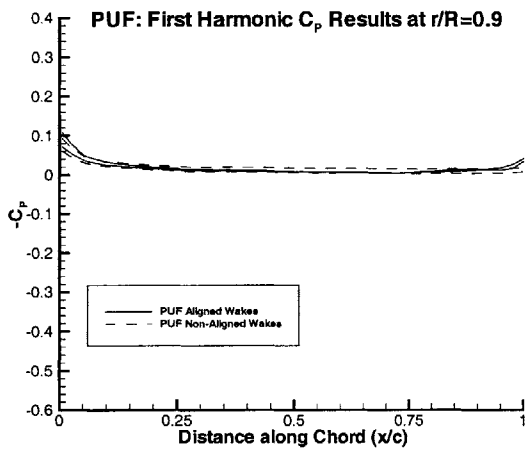
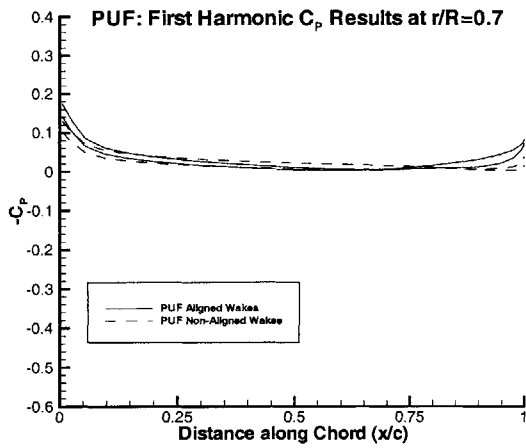
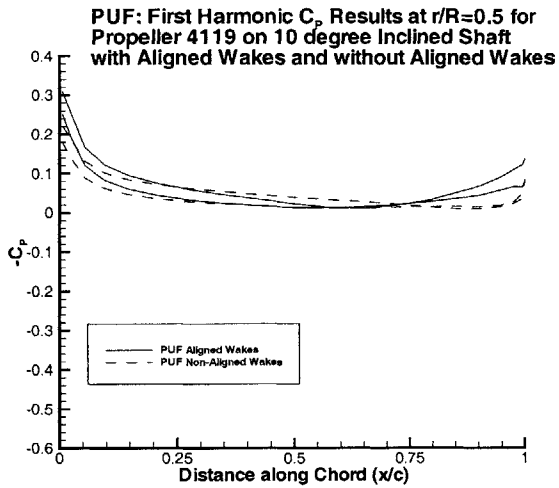


Figure 6-12: Magnitude of the First Harmonic of Propeller 4119 at 10 Degree inclined angle C_p .

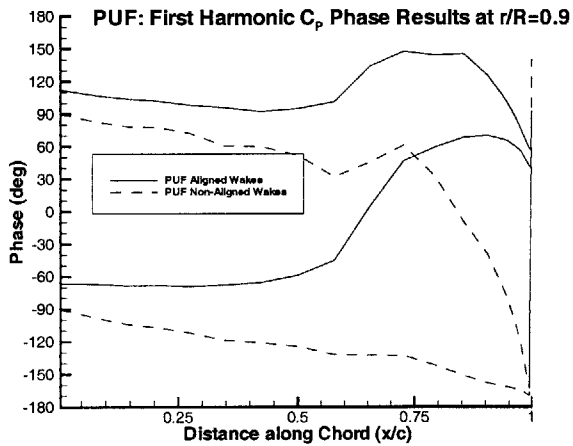
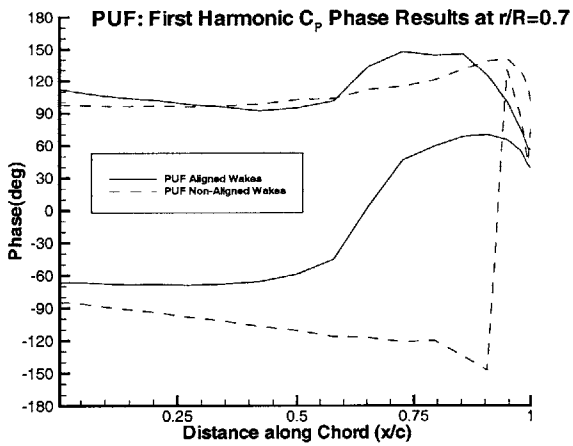
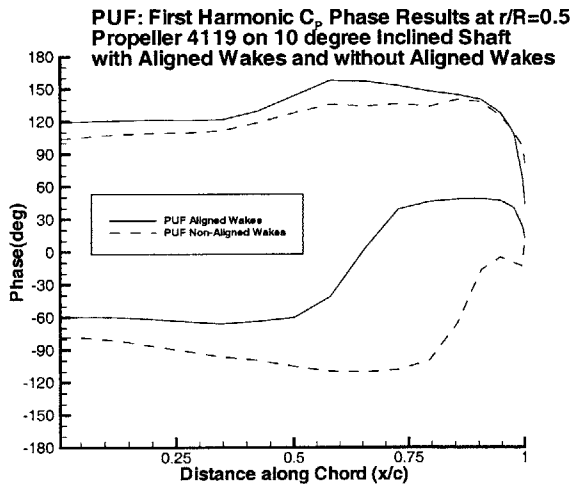


Figure 6-13: Phase of the First Harmonic of Propeller 4119 at 10 Degree inclined angle C_p .

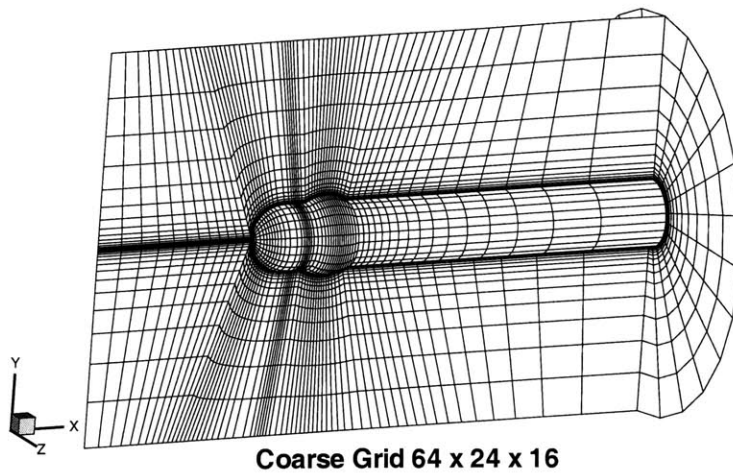


Figure 6-14: Computational model of the DTMB downstream driven propeller shaft. The propeller sits on the rounded ball near the nose.

Propeller	DTMB 4679
Inclined Angle	7.5 degrees
Advance Coeff.	1.078
Shaft Configuration	Downstream driven podded shaft

6.3.1 Nominal Flow Convergence

A grid convergence study was conducted on three grids of increasing fineness to ensure the solution was grid independent. Figure 6-15 shows the log of the pressure residual. Because no measurements were presented for the pressure coefficient along the shaft or boundary layer profiles, it is not possible to ascertain the validity of the RANS model.

It is interesting to examine the contours of axial velocity for the nominal flow case shown in Figure 6-16. Note the high velocities over the ball on the shaft where the propeller sits. The large radial gradients here show that the full detail of the shaft must be modeled to accurately reproduce the empirical propeller measurements.. The axial velocity contours shown clearly indicate that modeling the inflow as purely uniform, axial flow will produce erroneous computational predictions.

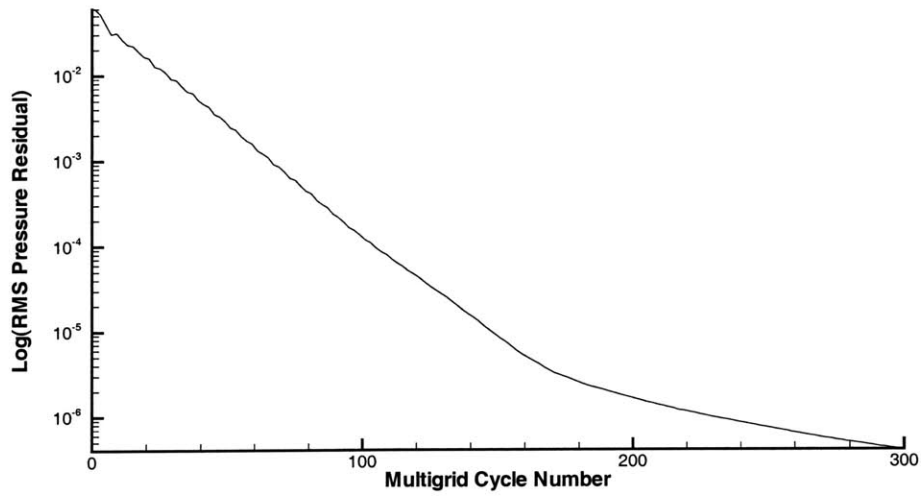


Figure 6-15: Log of pressure residuals for nominal, axial flow around DTMB downstream driven shaft.

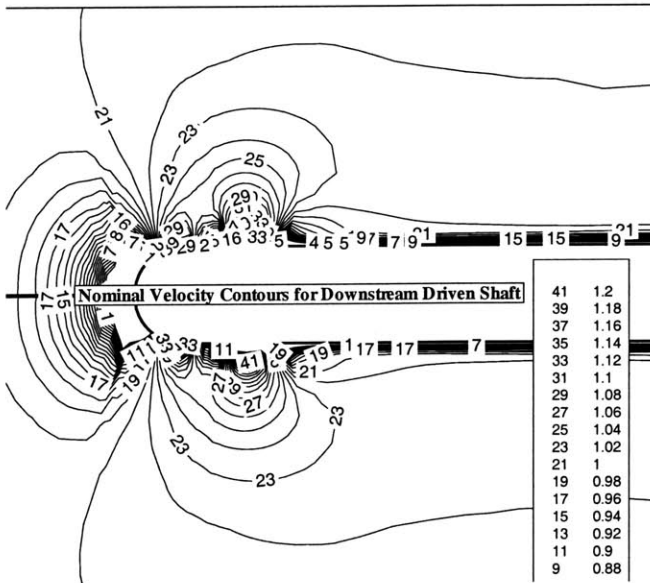


Figure 6-16: The nominal flow around the DTMB downstream driven test shaft.

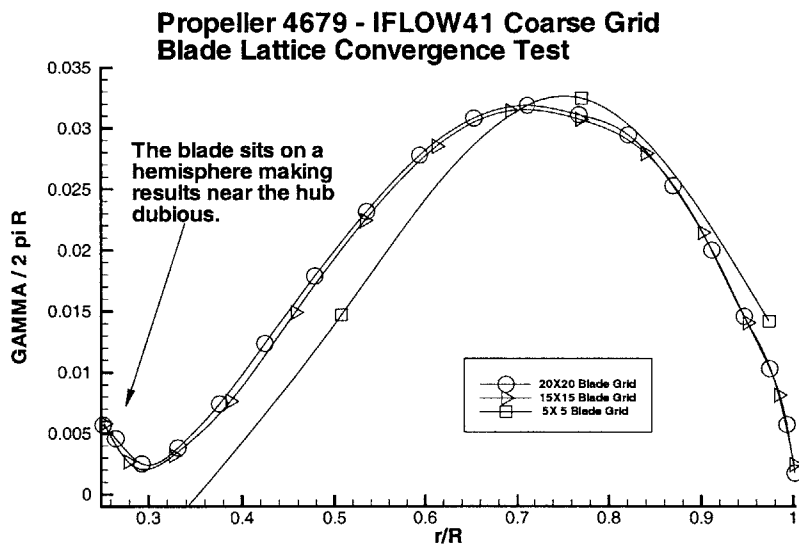


Figure 6-17: Propeller 4679 blade lattice convergence study. The 20X20 lattice shows nearly the same results as the 15X15 lattice, indicating that the solution has achieved grid independence.

6.3.2 Propeller Solution Convergence

A propeller convergence test was conducted by varying the lattice size of the propeller while holding the RANS grid constant. The resulting propeller circulation distribution plots are shown in Figure 6-17. Propeller lattice convergence is shown at a 20X20 lattice. The expected flattening of the circulation distribution is not seen near the hub due to the axial curvature of the hub.

6.3.3 Aligned and Non-Aligned Wake Results

An experiment was conducted with a propeller code with non-aligned wakes and a propeller code with aligned wakes, both coupled with the RANS codes for Propeller 4679 at 7.5 degree inclined angle. This test quantifies the effects upon the unsteady forces of aligning the wake. The coupled results, presented in Table 6.3.3 show similar trends presented earlier for stand-alone PUF. The large difference in the Z component of the unsteady force is expected as the propeller is inclined with respect to the Y axis, thus the Z directed force is the unsteady component of the side force.

Component	Wake Aligned	Non-aligned Wake
F_X	0.01581	0.01602
Phase F_X	183	159
F_y	0.00718	0.00726
Phase F_Y	16	36
F_Z	0.01313	0.01138
Phase F_Z	26	8

Table 6.2: Predicted 1st Harmonic force results for P4679 at 7.5 degree inclined angle for aligned and non-aligned wakes.

6.3.4 Blade Loading Numerical Comparison

The plots of the blade loading for each discrete propeller position are shown in Figure 6-18. From the figure, it is clear that there are magnitude and phase differences between the two propeller programs.

6.3.5 Experimental Comparison

The results from the experiments conducted at DTMB contain C_P data at constant radial positions along the propeller, for both the suction and pressure side. The data are both amplitude and phase information. This data allows a key comparison to be made with the fully-aligned wake method, since results presented in previous sections have shown that the wake alignments procedure alters both the magnitude and phase of the resulting blade forces.

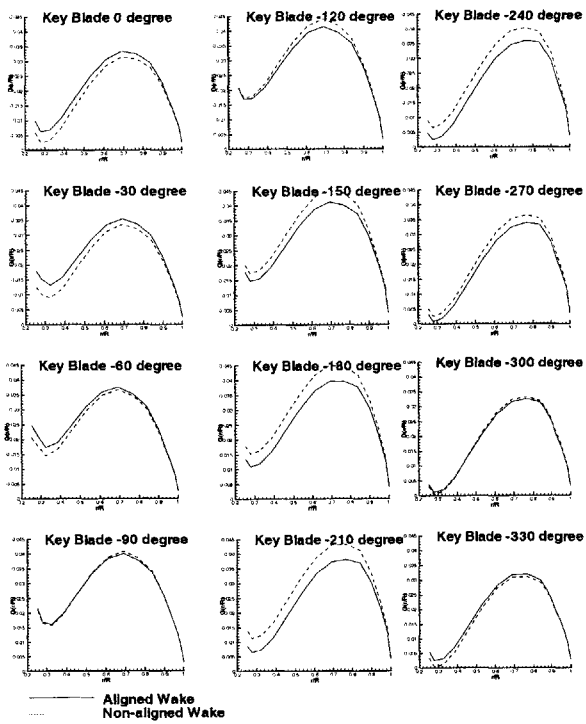


Figure 6-18: Propeller 4679 Steady C_p at $r/R = 0.5$

**Propeller 4679 at 7.5 Degree Inclined Angle
Steady C_p at $r/R=0.5$**

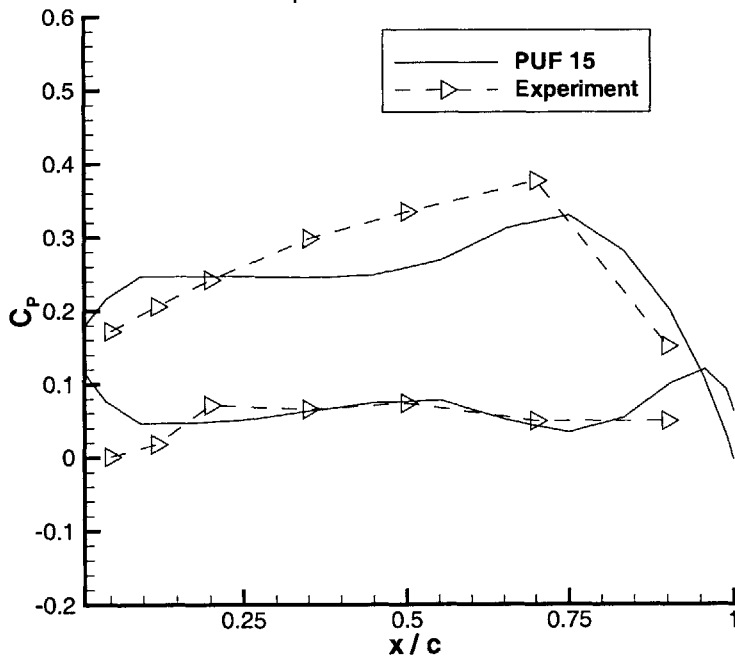


Figure 6-19: Propeller 4679 Steady C_p at $r/R = 0.5$

**Propeller 4679 at 7.5 Degree Inclined Angle
Steady C_p at $r/R=0.7$**

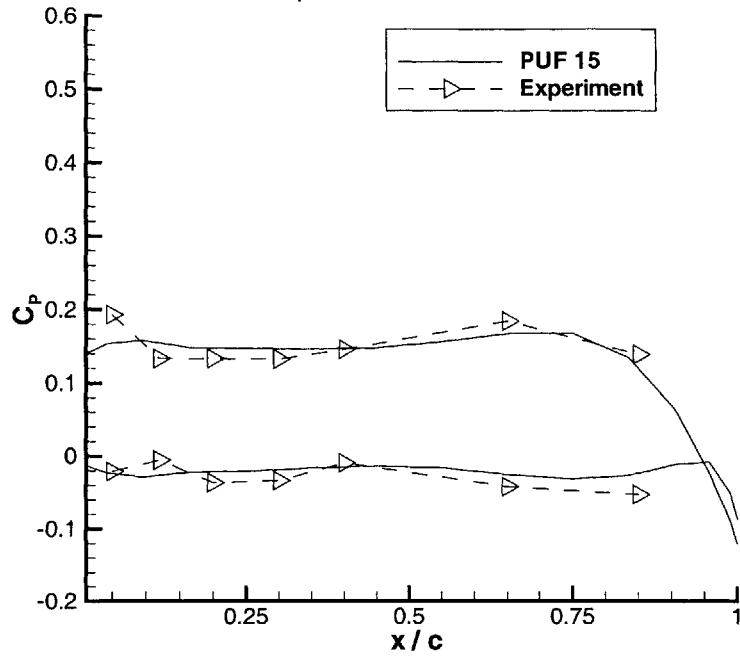


Figure 6-20: Propeller 4679 Steady C_p at $r/R = 0.7$

**Propeller 4679 at 7.5 Degree Inclined Angle
Steady C_p at $r/R=0.9$**

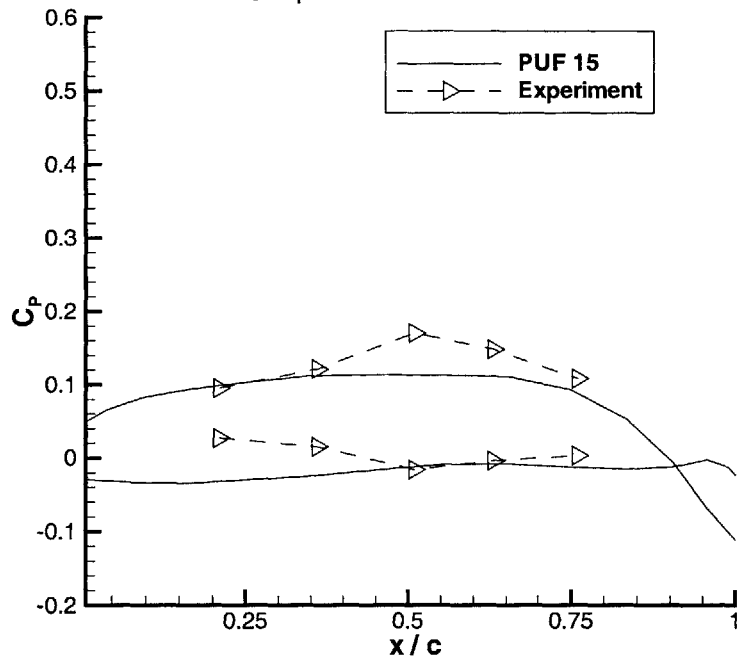


Figure 6-21: Propeller 4679 Steady C_p at $r/R = 0.9$

**Propeller 4679 at 7.5 Degree Inclined Angle
1st Harmonic C_p at $r/R=0.5$**

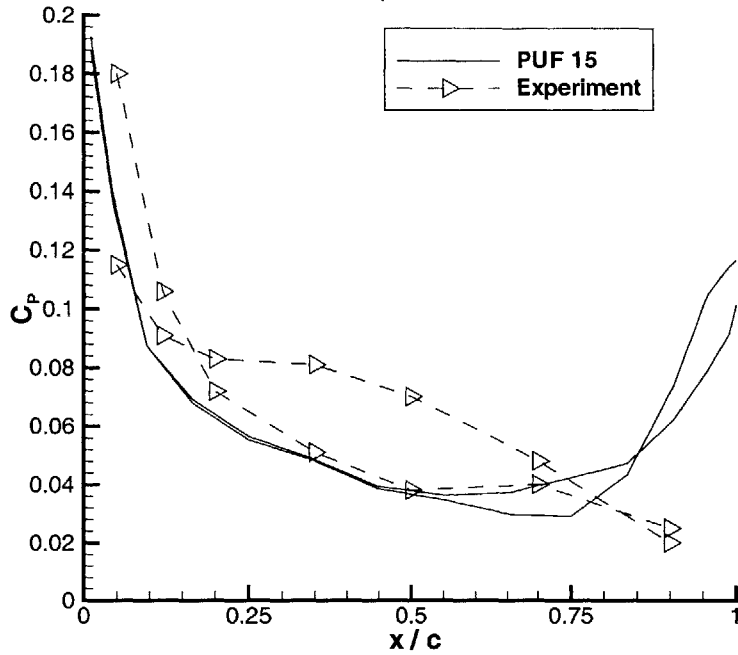


Figure 6-22: Propeller 4679 first harmonic C_p at $r/R = 0.5$

First Harmonic Phase Comparison

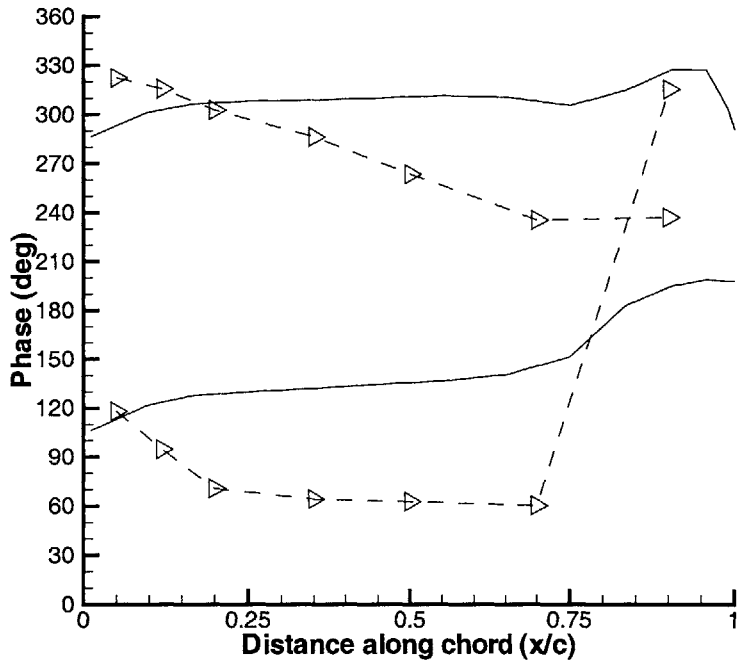


Figure 6-23: Propeller 4679 first harmonic phase of C_p at $r/R = 0.5$

**Propeller 4679 at 7.5 Degree Inclined Angle
1st Harmonic C_p at $r/R=0.7$**

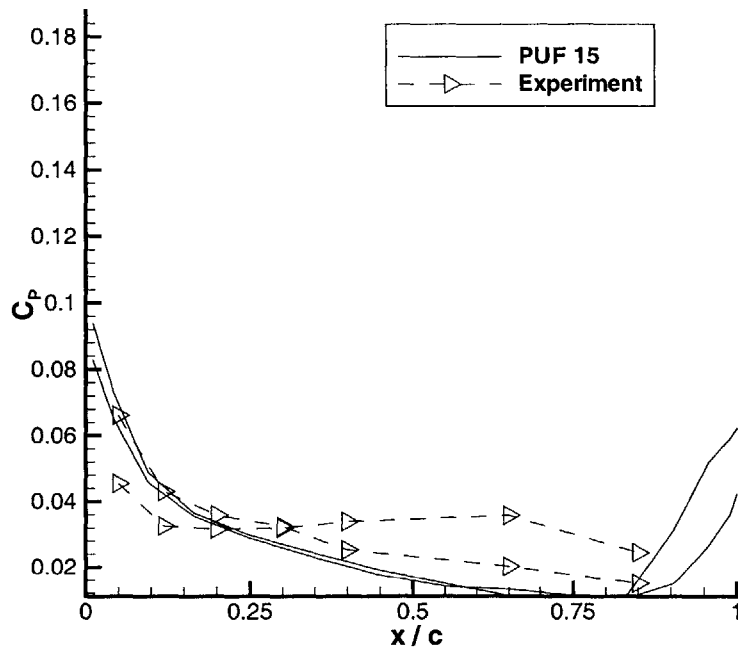


Figure 6-24: Propeller 4679 first harmonic C_p at $r/R = 0.7$

First Harmonic Phase Comparison

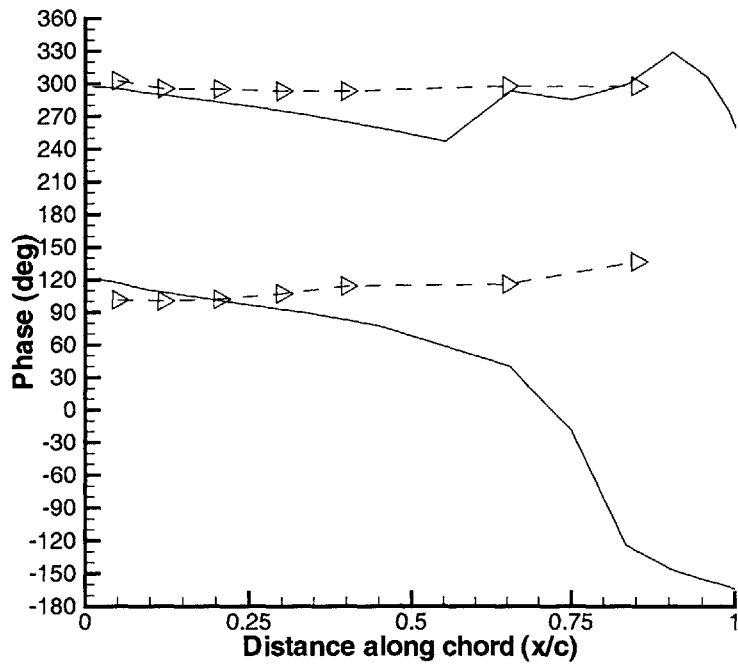


Figure 6-25: Propeller 4679 first harmonic phase of C_p at $r/R = 0.7$

**Propeller 4679 at 7.5 Degree Inclined Angle
1st Harmonic C_p at $r/R=0.9$**

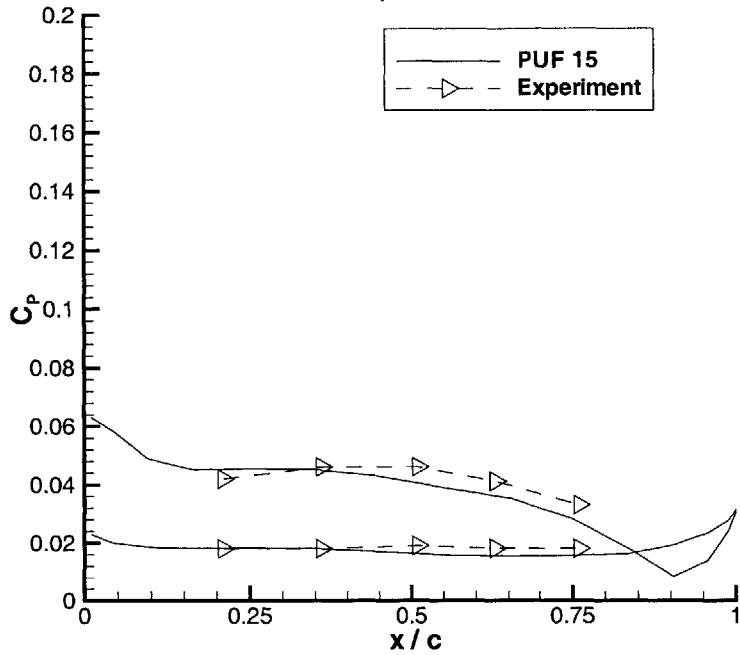


Figure 6-26: Propeller 4679 first harmonic C_p at $r/R = 0.9$

First Harmonic Phase Comparison

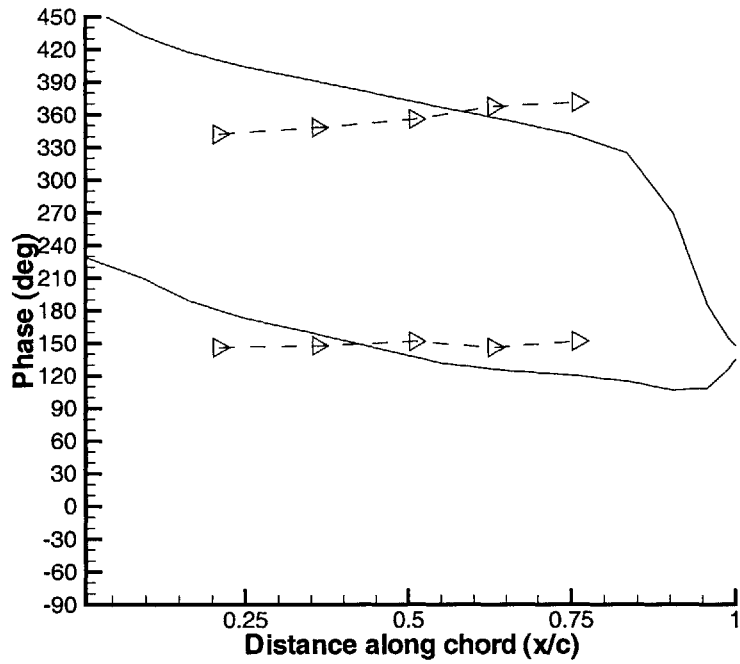


Figure 6-27: Propeller 4679 first harmonic phase of C_p at $r/R = 0.9$

Chapter 7

Conclusions

This thesis has focused on design challenges facing propulsion engineers. By improving upon the state of the art techniques in vortex-lattice lifting-surface methods and coupled RANS methods as specifically applied to complex propulsor design and analysis, this thesis has advanced the frontier of knowledge, and will allow practicing engineers to more fully explore their design space.

There are still a great number of challenges facing the industry and the scientific research community which supports it. The largest challenge is the promise of more fully populated flow simulation codes and empirical testing techniques to enhance our understanding of the physics of the problem, and then apply these new insights into rigorous design methodologies. This thesis has shown how localized applications of physics-based modeling and sleight of mind paradigm shifts in discretization thoughts have led to noticeable gains in efficiency and accuracy. And as far as pushing the state of the art, this is a practical course to achieving evolutionary progress. But as basic science advances upon revolutionary insights based on ever refined length scales, time scales, and densities, there must be at some point order imposed to feed back into the engineer that builds a product.

There are, also, many specific challenges facing future researchers in the state of the art as envisioned today,

1. **Unsteady drag force modeling:** While the state of the art is pushing for

steady drag modeling on a vortex-lattice lifting-surface model, the unsteady blade drag, especially in the highly loaded internal propulsion units gaining popularity, remains an open question. Experimental evidence shows that even the steady numerical models do not perform as well as was hoped.

2. **Thickness modeling:** This thesis ignored the thickness of the blade (usually modeled as source singularities) upon the blade solution. This was purposeful in that if the unsteady thickness load coupling effects are modeled, then the unsteady viscous load coupling effects must too be modeled. Ignoring both in tandem is hypothesized to create no undue logical errors.
3. **Entropy generation in RANS:** The fact that blade drag appears as an entropy source in the RANS code could be modeled through another tweaking of the right hand side of the RANS equation. However, the effects upon firmly-established turbulent boundary layer models would certainly lead to an active debate.

7.1 The Future

The future is bright for today's engineers. Ever faster computational resources with ever expanding memory make problems solvable today that were untenable twenty years ago. This thesis has provided a few new tools to examine some of the more interesting aspects of the design space that the digital and hardware revolution brings forth.

Appendix A

Validation Techniques for PUF Type Codes

This appendix seeks to layout validation techniques which have been used with success at the MIT Marine Hydrodynamics Laboratory. Along with validation to code improvements, sections are also provided which list best practice trouble shooting methods.

Note that the best validation is with experimental data, but in the absence of good data for all cases, there are some numerical techniques to verify the accuracy of a coupled technique.

A.1 Coupled Validation

The best validation between a vortex-lattice lifting-surface code and a volume solver (RANS, Euler) is the effective inflow equals 1.0 test.

A.1.1 Effective Inflow Equals 1.0 Test

Running the volume solver in inviscid mode turns the expensive volume solver into a Laplace Solver.

1. Set farfield upstream fluid velocity to 1.0

2. Couple PUF and the volume solver

At the end of the coupling, the effective velocity everywhere on the blade should be 1.0 since

$$V_{eff} = V_{RANS} - V_{induced} \quad (\text{A.1})$$

and there is no vorticity in the inflow which interacts with the blade vorticity to cause a radial redistribution of the propeller inflow profile.

This procedure also results in a completely aligned wake, and so can be used to validate wake self-alignment techniques, too.

Plots of interest would show

- Contour plot of the effective velocity V_X on the blade surface
- Wake lattice pitch at various spanwise positions

A.1.2 V_θ Test

Between the propeller code and the RANS code, the tangential induced velocity (or swirl velocity) must be the same. An easy check, then, is to extract the swirl velocity from the RANS domain at the blade trailing edge, and compare this with the swirl velocity output by the propeller code. Differences are easily attributable to force non-dimensionalization and scaling issues, and should be immediately addressed.

A.1.3 Blade Isolation Test

A good test to test the coupling suite is to make an inflow that has a pie slice of a lower velocity. Using this contrived flowfield as an input to the propeller code, the output induced velocities from the propeller code should show a spike in the same blade position.

Another similar worthwhile test is to couple the RANS and the propeller code where the RANS inflow has harmonic content. Once again, an overlay of the induced

velocities output by the propeller program and the inflow, should show contour levels which are in phase.

A.1.4 Coupled Problems

In generating a new coupling, most of the problems lie in the non-dimensionalization of forces.

A.1.5 Coupling Code Non-Dimensionalizations

Length Scale

The length scales are independently scaled between PUF and the volume solver (labelled as RANS in the following equations). The ratio of the length scales is given as λ .

$$\frac{L_{PUF}}{L_{RANS}} = \lambda \quad (\text{A.2})$$

The L in both codes is taken as the same measure, usually of propeller diameter. Therefore, the volume ratio is given as

$$\frac{\Delta_{PUF}}{\Delta_{RANS}} = \frac{\lambda^3 \Delta_{RANS}}{\Delta_{RANS}} = \lambda^3 \quad (\text{A.3})$$

Rotation Speed

By standard definition, the advance ratio, J , is given as

$$J = \frac{V}{nD} \quad (\text{A.4})$$

Within PUF, V is always V_{ship} which is assumed to be 1.0 and the propeller diameter D is 2.0. Thus, the rotation speed in PUF is given as

$$n_{PUF} = \frac{1}{2J} \quad (\text{A.5})$$

This translates into an Ω given as

$$\Omega_{PUF} = \frac{\pi}{J} \quad (\text{A.6})$$

Force

Start from the fact that by similarity considerations, the non-dimensional thrust coefficient must match between the two codes.

$$K_{T_{RANS}} = K_{T_{PUF}} \quad (\text{A.7})$$

By definition, then, the force in the RANS domain can be shown to be

$$\begin{aligned} F_{RANS} &= K_T \rho_{RANS} n_{RANS}^2 D_{RANS}^4 \\ &= K_T \rho_{RANS} \left(\frac{V_s}{J D_{RANS}} \right)^2 D_{RANS}^4 \\ &= \frac{1}{J^2} K_T \rho_{RANS} V_s^2 D_{RANS}^2 \end{aligned} \quad (\text{A.8})$$

Substitute in the definition for K_T which is known from PUF.

$$\begin{aligned} F_{RANS} &= \left[\frac{T_{PUF}}{\rho D_{PUF}^4 / 4J^2} \right] \frac{\rho_{RANS} V_s^2 D_{RANS}^2}{J^2} \\ &= \frac{4T_{PUF}}{D_{PUF}^4} D_{RANS}^2 \rho_{RANS} V_s^2 \\ &= \frac{4T_{PUF}}{D_{PUF}^2} \frac{\rho_{RANS} V_s^2}{\lambda^2} \\ &= \frac{1}{\lambda^2} T_{PUF} \rho_{RANS} V_s^2 \end{aligned} \quad (\text{A.9})$$

If in the RANS code, $\rho = 1.0$ and $V_S = 1.0$ then

$$\frac{F_{RANS}}{F_{PUF}} = \frac{1}{\lambda^2} \quad (\text{A.10})$$

Force Density

Since most RANS codes use a force potential in the formulation of the Navier-Stokes equations, it is useful to see how a force density in the PUF coordinate system is transformed to a force density in the RANS coordinate system.

$$FRCDENSRANS = \frac{F_{RANS}}{L_{RANS}^3}$$

$$\frac{FRCDENSRANS}{FRCDENSPUF} = \lambda \quad (\text{A.11})$$

Appendix B

Potential - RANS Coupling

B.1 Effective Inflow

All propeller design and analysis takes place in the effective inflow. The calculation of the effective inflow is the starting point for any propeller design. An accurate knowledge of the effective inflow is critical to the success of the propeller design. The effective inflow is obscured by the fact that the vorticity present on the propeller blades and the vorticity within the boundary layer where the propeller operates interacts in a highly non-linear fashion.

While the nominal inflow is the inflow field when there is no propeller present, it is not equal to the effective inflow. This is due to the effects of the propeller upon the vorticity in the inflow. This can be seen from the definition of the vorticity vector.

$$\vec{\omega} = \vec{\nabla} \times \vec{V} \quad (\text{B.1})$$

A non-radially-uniform meridional inflow suggests the presence of a tangential component of vorticity.

$$\vec{\omega}_\theta = \frac{\partial \vec{V}_r}{\partial x} - \frac{\partial \vec{V}_x}{\partial r} \quad (\text{B.2})$$

The tangential vorticity can be thought of as a ring which circumscribes the propeller

shaft (Newman, 1977). The propeller induced velocity causes a contraction of the tangential vorticity ring. As this ring contracts, Kelvin's theorem requires that the strength of the vorticity remain constant. Therefore, from equation (B.2), it is evident there is a change in $\frac{\partial V_x}{\partial r}$ to counteract the change in $\frac{\partial V_r}{\partial x}$. If there were no tangential vorticity present in the inflow, though, the nominal and effective inflows are exactly equal. This fact is a standard technique used to validate the any coupling presented between a potential-based propeller code and an RANS code.

B.2 The Coupled Hull Flow Problem

To accurately design a propeller requires an accurate knowledge of the ship's resistance. The increased fluid velocity along the stern of the vessel due to the presence of the operating propeller causes an increase in drag since there is more incomplete pressure recovery. Historically, the values for the resistance of the ship with and without a propeller operating are related through the thrust deduction coefficient.

$$R_T = (1 - t) T \tag{B.3}$$

Modern computation techniques which model the interaction of the propeller and the fluid flow around the hull solve both the effective inflow and thrust deduction problems by nearly exactly solving for the fluid flow over the hull with the propeller operating. In this method, the entire submarine and duct, if present, is modelled in the computer. The shear flow is exactly computed along the body, and the correct propeller interaction effects are also modelled.

B.3 Implementation in RANS Formulation

The effects of the propeller are introduced into the RANS domain as body forces, which appear as source terms on the right hand side of the RANS equation, Equation B.4

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\frac{1}{Re} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] + F_i \quad (\text{B.4})$$

The body force term F_i must be properly non-dimensionalized for the RANS code at hand. Usually, the non-dimensionalization is,

$$F_i = \frac{f_i}{\rho U^2 / L} \quad (\text{B.5})$$

Where f_i is the force per unit volume from the propeller code.

Appendix C

Propeller Numeric Routines

C.1 Romberg Integration

There are several advantages to the use of the Romberg adaptive quadrature scheme with the Richardson extrapolation [26]:

- Improved accuracy. The truncation error associated with the normal trapezoidal scheme is $O(h^2)$ while with the Romberg scheme the order of the error is reduced to $O(h^4)$.
- Calculation reuse. Previously calculated terms are used in the extrapolation step reducing computational expense.
- Controlled accuracy. The stepping scheme which steps the solution to finer levels can be halted once a desired convergence level is reached.
- Fast convergence. Because the Romberg scheme contains an extrapolated error estimate, fewer steps are needed to achieve convergence.
- The Romberg integration scheme has been validated to perform very well when the higher derivatives of the integrand are large, which is the case for vortex influence functions.

The basic formula of the trapezoidal integration is given in Equation C.1.

$$T_{1,k} = \frac{h}{2} [f(a) + f(b) + 2 \sum_{j=1}^l f(a + jh)] \quad (\text{C.1})$$

where

$$h = \frac{b-a}{2^{k-1}} l = 2^{k-1} - 1$$

The prediction via application of the trapezoidal scheme via successive spacing refinement can be accelerated through extrapolated estimates of the coarser approximations. Forming these successive extrapolations and solution into a matrix gives the elements of the classic Romberg matrix,

$$T_{l,k} = \frac{1}{4^{l-1} - 1} (4^{l-1} T_{l-1,k+2} - T_{l-1,k}) \quad (\text{C.2})$$

The first column of the Romberg matrix contains the approximation of the integral and the extrapolation for each step is along the diagonal. The relative truncation error at the end of any refinement step is given by,

$$\epsilon = \left| \frac{T_{1,k} - T_{1,k-1}}{T_{1,k}} \right| \quad (\text{C.3})$$

C.2 Tri-linear Interpolation

The art of data interpolation is a careful balance of the computation time, accuracy, and stability of the method. Many schemes were investigated before the tri-linear scheme was selected:

- tricubic spline interpolation (too unpredictable in high gradient regions)
- constant strength (too inaccurate for coarse RANS grids)

- corner-weighted block interpolation (inaccurate)

In the PUF-RANS coupled scheme, the major interpolations involve interpolating the velocities defined on cell centers to the PUF control points (Neumann boundary condition points), which are usually not coincidental. Originally, PUF utilized a three dimensional cubic spline interpolation. This scheme proved unstable due to the presence of large, higher order, gradients in the flowfield, especially in the boundary layer region. This thesis implemented a tri-linear interpolation scheme.

Given the position and velocities on the eight node points for the block where the field point is located,

$$X_i, Y_i, Z_i, VX_i, VY_i, VZ_i \quad \text{where } i = 1..8$$

The linear assumption assumes that the velocity in any direction at any field point in the block is given by,

$$VX(x, y, z) = Ax + By + Cz + Dxy + Exz + Fyz + Gxyz + H$$

$$VY(x, y, z) = Ax + By + Cz + Dxy + Exz + Fyz + Gxyz + H$$

$$VZ(x, y, z) = Ax + By + Cz + Dxy + Exz + Fyz + Gxyz + H$$

where the weight coefficients A-H differ for each velocity direction

The weight coefficients are found for each of the three directions independently by solving the 8x8 coefficient matrix. Standard LAPACK LU decomposition and back substitution routines were used for this thesis.

C.3 Elliptical Streamline Grids

One of the first problems that is encounter by the propeller program is to find a circumferential mean flowfield from the RANS total flowfield, so that the propeller

blade lattice can be placed along these streamlines. Previous schemes took the circumferential mean velocity through the RANS domain. The problem came that in regions of high velocity, the propeller blade growing algorithm would take too large of a step and go through a solid boundary. This produced propellers that extended into hubs, and into ducts.

The solution which this thesis incorporated was to use the 2-D outer boundary of RANS domain as the edges of an inviscid flowfield, and solve Laplace's equation internal to the domain to give beautiful streamlines which follow the boundaries, and will keep the propeller blades in the fluid domain.

The goal of any grid generation routine is to logically organize a set of (x, y, z) points within the fluid domain of interest so that the conservation laws can be applied on these discrete points.

Within the realm of numerical grid generation routines, there are several types of grids. The generation of a particular grid type is by dictated the later use of the grid. A particular type of grid is only suited to one, or a few, of the many numerical implementations of the governing fluid dynamic equations. On a broad level, there are two types of grids: structured and unstructured.

Another classification of grid generation routines is between those that are fixed and those that are adaptive. In a fixed grid, the gridpoints are fixed. In an adaptive scheme, the grid points are altered as the flow solution evolves to cluster gridpoints in regions of high flow gradients, near shocks for instance. With adaptive grids, the quality of the flow solution is increased due to the fineness of the grid in the regions that require it. However, this increase in accuracy comes with a computational, and programming, cost.

There are a wide variety of grid generation techniques available to the fluid dynamicist, each of which embodies a unique set of trade-offs. And it is not merely hearsay that the quality of the mesh grid greatly affects the quality of the numerical flow solution. This paper seeks to explain the rudimentary differences, and offer simple algorithms for their implementations.

C.3.1 Structured Grid Generation

The goal of a structured grid generation routine is to map gridpoints distributed in the x, y physical fluid domain to a ξ, η computational domain. In the computation domain $\Delta\xi = \Delta\eta = 1$ to allow a finite difference discretization of the differential form of the flow equations. As the finite difference equations must also be transformed from the physical to the computational domain, the grid and flow solution are intimately coupled by the metrics of the transformation. The requirements for a structured grid are that

1. There be a one-to-one mapping such that gridlines do not cross each other.

Also, for numerical accuracy of the solution to the flow equations we desire that,

1. The distribution of gridpoints is smooth.
2. There is a minimum of grid line skew.
3. Grid line orthogonality or near orthogonality is maintained.
4. Grid points are concentrated in regions of high flow gradients.

C.3.2 Techniques of Structured Grid Generation

There are three historically-used techniques for generating structured grids: conformal mapping, algebraic mapping, and the elliptic partial differential equation.

Conformal Mapping involves the use of complex variables to map the physical domain to the complex domain. While the metrics are guaranteed to be good, conformal mapping is applicable only in two dimensional regions. Even in two dimensional regions, very few geometries are known for which conformal mapping functions can be generated. Conformal grids are no longer used in industry.

Algebraic Mapping uses an algebraic transformation to map between the physical and computational domain. These routines are very fast and easy to implement.

However, control of grid skewness and smoothness is difficult, and discontinuities at the boundary are deadly when later solving boundary value partial differential equations.

Elliptic PDE grids are generated by solving Poisson's equation with a specified boundary geometry and either Dirichlet or Neumann boundary conditions (recall that with Dirichlet boundary conditions the points on the boundary are specified, and with Neumann boundary conditions the gradient, or normal, at the boundary is specified). PDE grid generation routines are also called boundary fitted, elliptical, or Laplacian grids. These routines are by far the most flexible and widely used structured grid method.

The following two sections will discuss the theory of elliptic grid generation, and how algebraic grids found a proper place.

C.4 Poisson, or Elliptical, Grid Generation

The beauty of Laplace's equation is its minimization principle. It can be shown that given a function ξ which satisfies Laplace's equation,

$$\xi_{xx} + \xi_{yy} = 0 \tag{C.4}$$

This function ξ also minimizes the quantity I , where I is given by

$$I = \int_{\Omega} \nabla \xi \cdot \nabla \xi d\Omega \tag{C.5}$$

Note that $\nabla \xi$ mathematically expresses the length of the grid spacing in the $\xi\eta$ domain. Hence, a solution to Laplace's equation leads to a minimization of grid density. The very minimum is of course equal spacing !

So, now all we have to do is solve Laplace's equation in the computational domain to minimize ξ and η . Laplace's equation in two dimensions for the (ξ, η) coordinates

is

$$\xi_{xx} + \xi_{yy} = 0 \quad (\text{C.6})$$

$$\eta_{xx} + \eta_{yy} = 0 \quad (\text{C.7})$$

In equations (C.6) and (C.7) x and y are the independent variables and ξ and η are the dependent variables. This implies that we are given a distribution of gridpoints in the (x, y) domain and alter the corresponding (ξ, η) gridpoints until equations (C.6) and (C.7) are satisfied. In reality, though, we exactly know the (ξ, η) position of every grid point in the $\xi\eta$ computational domain. So the trick is to interchange the role of the independent and dependent variables in equations (C.6) and (C.7) such that we are now solving for the (x, y) position of every (ξ, η) gridpoint.

It may be helpful to follow the algebraic steps necessary to derive the new version of Poisson's equation. We started by making the assertion that

$$\xi = \xi(x, y)$$

$$\eta = \eta(x, y)$$

and that

$$\begin{pmatrix} d\xi \\ d\eta \end{pmatrix} = \begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} \quad (\text{C.8})$$

Obviously we can state that a relationship exists for the inverse relationship.

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{pmatrix} \begin{pmatrix} d\xi \\ d\eta \end{pmatrix} \quad (\text{C.9})$$

By inspection we can relate the two transformation matrices and assert that

$$\begin{aligned} \begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix} &= \begin{pmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{pmatrix}^{-1} \\ &= \frac{1}{J} \begin{pmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{pmatrix} \end{aligned} \tag{C.10}$$

where J is the Jacobian of the transformation between (ξ, η) and (x, y) defined as

$$J = x_\xi y_\eta - x_\eta y_\xi \tag{C.11}$$

Now we are ready to perform the algebraic steps necessary to show how ξ_{xx} equals a function where x and y are differentiated with respect to ξ and η . We start with the [1,1] element of equation (C.10).

$$\begin{aligned} \xi_{xx} &= \left(\frac{y_\eta}{J} \right)_x \\ &\text{use the chain rule of differentiation} \\ &= \xi_x \left(\frac{y_\eta}{J} \right)_\xi + \eta_x \left(\frac{y_\eta}{J} \right)_\eta \\ &= \left(\frac{y_\eta}{J} \right) \left(\frac{y_\eta}{J} \right)_\xi - \left(\frac{y_\xi}{J} \right) \left(\frac{y_\eta}{J} \right)_\eta \end{aligned}$$

When this process is carried out again to find $\xi_{yy}, \eta_{xx}, \eta_{yy}$ and after several algebraic groupings are made, the following final expressions for x and y in terms of ξ and η result.

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0 \tag{C.12}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0 \tag{C.13}$$

where:

$$\alpha = x_{\eta}^2 + y_{\eta}^2$$

$$\beta = x_{\xi}x_{\eta} + y_{\xi}y_{\eta}$$

$$\gamma = x_{\xi}^2 + y_{\xi}^2$$

Equations (C.12) and (C.13) are sometimes referred to as the Thompson equations after Joe Thompson at Mississippi State University who first discovered their properties.

C.4.1 Solution Techniques for the Elliptical Grid Equations

We can solve equations (C.12) and (C.13) for x and y , respectively using standard iterative numerical methods for solving elliptical PDE's such as Gauss Siedel, SLOR, multigrid, etc . . . The non-linearities can be assumed away by treating them as linear. A simplified solution technique follows the following logic:

1. Provide an initial guess for the $(x, y)_{j,k}$ values of the gridpoints.
2. Calculate $(\alpha, \beta, \gamma)_{j,k}$ at each gridpoint.
3. Keeping $(\alpha, \beta, \gamma)_{j,k}$ fixed, perform a set number of Gauss-Siedel iterations on the x coordinate.
4. Keeping $(\alpha, \beta, \gamma)_{j,k}$ fixed, perform a set number of Gauss-Siedel iterations on the y coordinate.
5. Update boundary conditions, be they Dirichlet or Neumann.
6. Check for convergence through a suitable technique – for instance, calculate the log of the L_2 norm of the residual solution vector.
7. Repeat steps 2-6 until the solution is converged to the desired level of convergence.

Note that the non-linearities present in equations (C.12) and (C.13) slow the convergence rate for elementary iterative techniques. More sophisticated non-linear solution methods such as Newton-Rhapon can provide a faster convergence.

Although no forcing functions were used in this implementation, it would be an easy manner to add them in to handle more exotic domain geometries.

C.4.2 Forced Functions

Solving equations (C.12) and (C.13) guarantees a smooth, one to one mapping. However, an orthogonal grid is not guaranteed and there is no ability to cluster gridpoints in the desired region. Also, the smoothness of the Laplacian solution leads to pinched grids near convex boundaries, and stretched grids near concave boundaries. These deficiencies can be overcome through the use of forcing functions on the right hand side of the PDE's, leading exactly to Poisson's equation. Equations (C.12) and (C.13) with forcing functions are given by:

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2(Px_{\xi} + Qx_{\eta}) \quad (C.14)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2(Py_{\xi} + Qy_{\eta}) \quad (C.15)$$

The forcing functions P and Q allow for clustering near lines of constant ξ_i and η_i and near individual points (ξ_i, η_i) . The expressions for P and Q are:

$$P(\xi, \eta) = - \sum_i a_i \text{sign}(\xi - \xi_i) e^{c_i|\xi - \xi_i|} - \sum_i b_i \text{sign}(\xi - \xi_i) e^{-d_i \sqrt{(\xi - \xi_i)^2 + (\eta - \eta_i)^2}} \quad (C.16)$$

$$Q(\xi, \eta) = - \sum_i a_i \text{sign}(\eta - \eta_i) e^{c_i|\eta - \eta_i|} - \sum_i b_i \text{sign}(\eta - \eta_i) e^{-d_i \sqrt{(\xi - \xi_i)^2 + (\eta - \eta_i)^2}} \quad (C.17)$$

where

a_i, b_i are amplification factors

$a_i, b_i > 0$ attracts lines to ξ_i

$a_i, b_i < 0$ repel lines from ξ_i

c_i, d_i are decay factors.

The amplification factors cause gridlines to cluster near the gridline of interest. The decay factors set the degree to which the forcing decays as you get away from the gridline of interest. There is no way to specify *a priori* values for a_i, b_i, c_i, d_i .

With the introduction of the forcing functions, one to one gridpoint mapping with non-overlapping gridlines between the $\xi\eta$ and xy planes are no longer guaranteed. The grid must be carefully checked before it is used in a flow solver.

Orthogonality at the boundary can be artificially introduced by requiring that the grid points on the body, which is itself a line of constant η (i.e., η_0), be orthogonal to the gridpoints above the body surface at η_1 . The Dirichlet boundary condition mathematically can be expressed as:

$$x_\xi x_\eta + y_\xi y_\eta = 0 \tag{C.18}$$

Obviously this condition can be applied at any other boundary, too. Of course, at every boundary, only one type of boundary condition can be applied.

Appendix D

Propeller Terminology

D.1 Abbreviations

$[B]$	Propeller Blade-on-Blade influence matrix
D	Propeller diameter (= 2.0 in PUF)
$DTMB$	David Taylor Model Basin (now Naval Sea Systems Command, Carderock Division)
$HGAP$	Hub gap, distance between the hub and the propeller, only useful for tip driven propeller
J	advance coefficient, $= \frac{V_S}{nD}$
K_T	$= \frac{T}{\rho n^2 D^4}$
K_Q	$= \frac{Q}{\rho n^2 D^5}$
$NTSR$	Number of Time Steps per Revolution, PUF input parameter
n	propeller revolution rate in rps
\hat{n}	Surface normal vector
P	Pressure
Q	Propeller torque
r	Local radius
R	Propeller max radius (= 1.0 in PUF)

R_T	Total resistance
T	Barehull resistance
T	Propeller thrust
V^\otimes	Velocity in PUF
V^\odot	Velocity in RANS
\vec{V}_{eff}	Effective Inflow Velocity
$\vec{V}_{induced}$	Induced Inflow Velocity
$\vec{V}_{nominal}$	Nominal Inflow Velocity
\vec{V}_{total}	Total Inflow Velocity
V_S	Ship velocity (=1.0 in PUF)
$[W]$	Propeller wake on blade influence matrix
W_T	Taylor wake fraction
W_N	Nominal volumetric wake fraction
Z	Number of propeller blades
∇	Gradient Operator
ρ	density
λ	distance scaling factor between PUF and RANS
$\vec{\Gamma}$	Circulation
Γ_B	Bound circulation
Γ_S	Shed circulation, dumped into the wake
ω	Rotation rate

D.2 Propeller Terms

Effective Wake	Non-physical fluid velocity that is the total fluid velocity in the region of the propeller minus the induced velocities due to the propeller. Most potential-based propeller solvers solve the propeller problem in an effective wake.
Nominal Wake	Fluid flow in the region of the propeller at the design vehicle speed without the presence of the propeller
Total Wake	Fluid velocities in the region of the propeller with the effects of the propeller.

Bibliography

- [1] F. Buechoux. Improved algorithms for the computation of induced velocities in propeller design. Master's thesis, Massachusetts Institute of Technology, 1995.
- [2] William Coney. *A Method for the Design of a Class of Optimum Marine Propellers*. PhD thesis, Massachusetts Institute of Technology, Dept. of Ocean Engineering, 1989.
- [3] William B. Coney. *MIT-PLL User's Manual*. Dept. of Ocean Engineering, M.I.T., 1988.
- [4] S. Denny. Cavitation and open water performance tests fo a series of propellers designed by lifting surface methods. Technical Report Technical Report 2878, DTNSRDC, 1968.
- [5] H. Glauert. *The Elements of Aerofoil and Airscrew Theory*. Cambridge University Press, New York, 1948. reissued by Dover.
- [6] D. Greeley. personal correspondance. private communitaction relaying several years worth of research involving the numerical computations of ring vortex induced velocity, August 1999.
- [7] David S. Greeley and Justin E. Kerwin. Numerical methods for propeller design and analysis in steady flow. *SNAME Transactions*, 90:415–453, 1982.
- [8] G.R. Hough and D.E. Ordway. The generalized actuator disk. Technical Report TAR-TR 6401, Therm Advanced Research, 1964.

- [9] Gary S. Hufford, Mark Drela, and Justin E. Kerwin. Viscous flow around marine propellers using boundary layer strip theory. *Journal of Ship Research*, 38(1), March 1994.
- [10] S. D. Jessup. Measurements of the pressure distribution on two model propellers. Technical Report DTNSRDC-82/035, David Taylor Model Basin, July 1982.
- [11] Stuart Jessup. Experimental data for unsteady panel calculations and comparisons dtmb 4679. In Gindroz, Hishino, and Pylkkaned, editors, *22nd ITTC Propulsion Committee Propeller RANS/Panel Method Workshop*, 5-6 April, 1998.
- [12] Stuart D. Jessup. *An Experimental Investigation of Viscous Aspects of Propeller Blade Flow*. PhD thesis, The Catholic University of America, June 1989.
- [13] J. E. Kerwin. Propeller unsteady performance analysis program: Mit-puf-2 documentation. Technical report, MIT Department of Ocean Engineering, January 1985.
- [14] J. E. Kerwin. 13.04: Hydrofoils and propellers class notes. Technical report, M.I.T., 1997.
- [15] J.E. Kerwin, D.P. Keenan, S.D. Black, and J.G. Diggs. A coupled viscous/potential flow design method for wake-adapted, multi-stage, ducted propulsors using generalized geometry. *SNAME Transactions*, 102:23–56, 1994.
- [16] Justin E. Kerwin and Chang-Sup Lee. Prediction of steady and unsteady marine propeller performance by numerical lifting-surface theory. *SNAME Transactions*, 86:218–253, 1978.
- [17] Spyros Kinnas and Sangwoo Pyo. Cavitating propeller analysis including the effects of wake alignment. *Journal of Ship Research*, March 1999.
- [18] D. Kucheman and J. Weber. *Aerodynamics of Propulsion*. McGraw-Hill Book Company, Inc., 1953.

- [19] C.E. Lan. A quasi-vortex lattice method in thin wing theory. *J. Aircraft*, 11(9), September 1974.
- [20] L. Leibman. An enhanced vortex lattice propeller design method. Master's thesis, Massachusetts Institute of Technology, 1991. Dept. of Ocean Engineering.
- [21] G. P. McHugh, T. E. Taylor, W. M. Milewski, and J.E. Kerwin. Pbd-14.4: A coupled lifting -surface design/analysis code for marine propulsors. Technical report, Depart of Ocean Engineering, Massachusetts Institute of Technology, August 1998.
- [22] J.N. Newman. *Marine Hydrodynamics*. MIT Press, 1989.
- [23] Sangwoo Pyo and Spyros A. Kinnas. Propeller wake sheet roll-up modeling in three dimensions. *Journal of Ship Research*, 41(2), 1997.
- [24] Wei-Zen Shih. Effective wake calculations by solving the euler equation. Department of Ocean Engineering Report No. 88-2, Massachusetts Institute of Technology, 1988.
- [25] Wei-Zhen Shih. *Propeller Steady Flow Analysis Programs: MIT-PSF-2 User's Manual*. Dept. of Ocean Engineering, M.I.T., 1997.
- [26] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. New York: Springer-Verlag, 1980.
- [27] C.-H. Sung and M.J. Griffin. Improvements in incompressible turbulent horse-shoe vortex juncture flow calculations. In *29th Aerospace Sciences Meeting*. AIAA 91-0022, Jan 7-10 1991.
- [28] C.-H. Sung, M.J. Griffin, and R.M. Coleman. Numerical evaluation of vortex flow control devices. In *AIAA 22nd Fluid Dynamics, Plasma Dynamics and Lasers Conference*. AIAA 91-1825, June 24-26 1991.

- [29] D. T. Valentine. Reynolds-averaged navier-stokes codes and marine propulsor analysis. Technical Report CRDKNSWC/HD-1262-06, Carderock Division, Naval Surface Warfare Center, October 1993.
- [30] C. L. Warren. *Prediction of Propulsor-Induced Maneuvering Forces Using a Coupled Viscous/Potential-Flow Method for Integrated Propulsors*. PhD thesis, Massachusetts Institute of Technology, June 1999.